# ccreative <br> <br> commons 

 <br> <br> commons}
$\begin{array}{lllllllllll}\text { C } & \mathrm{O} & \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{N} & \mathrm{S} & \mathrm{D} & \mathrm{E} & \mathrm{E} & \mathrm{D}\end{array}$

저작자표시-비영리-변경금지 2.0 대한민국
이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:


저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건 을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 숩게 요약한 것입니다.

$$
\text { Disclaimer } \square
$$

## c)Collection

# Numerical study of several self-assembly processes of polyelectrolytes: <br> DNA as scaffold to assemble gold nanoparticles and assembly process in PEDOT:PSS by addition of ionic liquids 

Ambroise de Izarra ( 암브로워즈 드 이잘라 )

Department of<br>Energy Science \& Engineering

DGIST

# Numerical study of several self-assembly processes of polyelectrolytes: <br> DNA as scaffold to assemble gold nanoparticles and assembly process in PEDOT:PSS by addition of ionic liquids 

Ambroise de Izarra ( 암브로워즈 드 이잘라 )

Department of<br>Energy Science \& Engineering

DGIST

# Numerical study of several self-assembly processes of polyelectrolytes: <br> DNA as scaffold to assemble gold nanoparticles and assembly process in PEDOT:PSS by addition of ionic liquids 

Advisor: Professor Yun Hee Jang<br>Co-advisor: Professor Yves Lansac

by
Ambroise de Izarra
Department of Energy Science \& Engineering
DGIST
A thesis submitted to the faculty of DGIST in partial fulfillment of the requirements for the degree of Doctor of Science in the Department of Energy Science \& Engineering. The study was conducted in accordance with Code of Research Ethics ${ }^{1}$
11. 20. 2020

Approved by

Professor Yun Hee Jang

(Advisor)
Professor Yves Lansac

(Co-advisor)

[^0]
# Numerical study of several self-assembly processes of polyelectrolytes: <br> DNA as scaffold to assemble gold nanoparticles and assembly process in PEDOT:PSS by addition of ionic liquids 

Ambroise de Izarra

Accepted in partial fulfillment of the requirements for the degree of Doctor of Science.
11. 20. 2020


Ph.D/ES<br>201734009<br>암브로워즈 드 이잘라 . Ambroise de Izarra. Numerical study of several self-assembly processes of polyelectrolytes: DNA as scaffold to assemble gold nanoparticles and assembly process in PEDOT:PSS by addition of ionic liquids. Department of Energy Science \& Engineering. 2021. 465p. Advisors Prof. Yun Hee Jang, Co-Advisors Prof. Yves Lansac


#### Abstract

Polyelectrolytes are polymers or macromolecules charged in solution which represents both a biological and an industrial interest. Their assembly properties are largely controlled by long range electrostatic interactions.

Despite a recent improvement in the understanding of the mechanisms involved, the prediction/characterization of the assembly processes remain problematic. This is due to the fact that the formation of complexes results from a delicate compromise between various interactions in addition to being sensitive to many experimental parameters such as the preparation conditions, the nature of the polyelectrolytes, the temperature and solvent effects...

It is the object of this thesis to better understand these assembly phenomena by an approach based on molecular simulation. In particular, we will focus on two systems of interest for future technological applications.

A first project will consist in modeling by Monte Carlo simulations the assemblies of functionalized gold nanoparticles in the presence of one or several DNAs. Then, it was reported that the properties of electrical transport of a complex made up of a semi-conducting polymer, the poly (3,4-ethylene dioxythiophene) (PEDOT) solubilized in water with its insulating counter-chain polystyrene sulfonate (PSS) have been improved by adding ionic liquid. The objective of this project will consist in understanding the mechanism of PEDOT:PSS conductivity enhancement by using the tools of molecular dynamics.


[^1]Abstract ..... i
List of Figures ..... viii
List of Tables ..... xiv
Introduction ..... 1
1 Nature and interactions of polyelectrolytes ..... 5
1.1 Generalities on polyelectrolytes ..... 6
1.1.1 Definition ..... 6
1.1.2 Natural polyelectrolytes ..... 7
1.1.2.1 Polysaccharides ..... 7
1.1.2.2 Polypeptides and proteins ..... 9
1.1.2.3 Polynucleotides ..... 10
1.1.3 Artificial polyelectrolytes ..... 10
1.1.3.1 Diblock copolymers ..... 12
1.1.3.2 Hydrophobic modified polyelectrolytes ..... 13
1.1.4 Conclusion ..... 13
1.2 Some physical properties of polyelectrolytes ..... 13
1.2.1 Polyelectrolytes in solution ..... 14
1.2.1.1 Strength of polyelectrolytes ..... 14
1.2.1.2 Dissociation of a ionomer in solution yields to polyelectrolyte ..... 15
1.2.2 Electrostatic interactions in aqueous solution ..... 18
1.2.2.1 Scale of electrostatic interactions in water ..... 18
1.2.2.2 Modeling electrostatic interactions in solution ..... 18
1.2.3 Physics of hydrophilic polyelectrolytes in solution ..... 21
1.2.3.1 Conformation of a neutral chain in solution ..... 21
1.2.3.2 Conformation of a weakly charged polyelectrolyte in solution ..... 25
1.2.3.3 Model of a highly charged polyelectrolyte: Manning-Oosawa theory ..... 28
1.2 .4 Conclusion ..... 32
1.3 Complexes of polyelectrolytes ..... 33
1.3.1 Interactions driving polyelectrolyte complexation ..... 33
1.3.2 Mechanism of complex formation ..... 34
1.3.2.1 Thermodynamics for complex formation ..... 34
1.3.2.2 Dynamic of complex formation ..... 36
1.3.3 Possible structures of polyelectrolytes ..... 36
1.3.4 Conclusion ..... 39
2 Studied polyelectrolyte systems ..... 41
2.1 Project 1: Polyelectrolyte complexation between DNA and nanoparticles ..... 43
2.1.1 Fundamental aspects of DNA compaction ..... 43
2.1.1.1 The coil-globule transition ..... 43
2.1.1.2 Compaction agents ..... 44
2.1.1.3 Condensation of cationic agents on DNA ..... 46
2.1.1.4 Mechanism of DNA compaction ..... 47
2.1.1.5 DNA overcharging effect ..... 49
2.1.1.6 Structures of compacted DNA : shape and stability ..... 49
2.1.1.7 Factors that modified DNA compaction/decompaction ..... 50
2.1.2 DNA compaction with functionalized gold nanoparticles ..... 51
2.1.2.1 Synthesis of functionalized gold nanoparticles. ..... 51
2.1.2.2 Applications of ligand-stabilized gold nanoparticles ..... 53
2.1.2.3 Self-assembly of ligand-stabilized gold nanoparticles on DNA tem- plates ..... 56
2.2 Project 2: Polyelectrolyte complexation between PEDOT:PSS and ionic liquids ..... 61
2.2.1 Review about conducting polymers ..... 61
2.2.1.1 Conjugation and $\pi$-bond ..... 61
2.2.1.2 Semi-conducting polymers ..... 62
2.2.1.3 Conduction of semi-conducting polymers ..... 64
2.2.1.4 Conduction mechanism in semi-conducting polymers ..... 66
2.2.2 The PEDOT semi-conducting polymer ..... 67
2.2.2.1 Synthesis of PEDOT:PSS and PEDOT:Tos ..... 67
2.2.2.2 Structure of PEDOT:PSS/PEDOT:Tos thin films ..... 69
2.2.2.3 Applications of PEDOT:PSS ..... 70
2.2.3 PEDOT:PSS conductivity enhancement by ionic liquids ..... 72
2.2.3.1 Definition and general properties of ionic liquids ..... 73
2.2.3.2 Controlling ordering in PEDOT:PSS solution with imidazolium- based ionic liquid for PEDOT:PSS conductivity enhancement ..... 74
2.3 Objectives of the thesis ..... 80
3 Simulation Methods ..... 83
3.1 Simulations at different scales ..... 84
3.2 Molecular Monte Carlo Simulations ..... 85
3.2.1 Motivations ..... 85
3.2.2 Monte Carlo simulations in various ensemble ..... 87
3.2.2.1 The canonical ensemble ..... 87
3.2.2.2 Isobaric-isothermal ensemble ..... 90
3.2.2.3 Grand Canonical ensemble ..... 92
3.2.3 Implemented Trial Moves ..... 94
3.2.3.1 Translational move ..... 94
3.2.3.2 Rotational move ..... 95
3.2.3.3 Collective move ..... 96
3.2.4 Interactions in the simulation box ..... 97
3.2.4.1 Periodic boundary conditions ..... 98
3.2.4.2 Cell lists for pairwise hard core potential ..... 100
3.2.4.3 Ewald summation method ..... 101
3.2.5 The Widom insertion method ..... 107
3.2.6 The Monte Carlo simulation package ..... 110
3.3 Molecular dynamics simulations ..... 111
3.3.1 General idea ..... 112
3.3.2 Integration of equations of motion ..... 113
3.3.3 Force Fields ..... 115
3.3.3.1 Bonded interactions ..... 115
3.3.3.2 Non-bonded interactions ..... 117
3.3.4 Regulation of Pressure and Temperature ..... 117
3.3.4.1 Thermostat ..... 117
3.3.4.2 Barostat ..... 120
3.3.5 Free energy calculation ..... 122
3.3.6 Molecular dynamics simulation packages ..... 126
3.3.7 How to perform a simulation with GROMACS?. ..... 127
4 Self-assembly of DNA mediated by cations or Nanoparticles ..... 131
4.1 Choice of molecular model ..... 133
4.1.1 Molecular model of DNA ..... 133
4.1.2 Molecular model of cationic gold nanoparticles and multivalent counterions ..... 135
4.2 Attraction and compaction of DNA molecules mediated by multivalent cations ..... 139
4.2.1 Adsorption of salt on a DNA molecule in presence of counterions ..... 139
4.2.1.1 Methods ..... 140
4.2.1.2 Results ..... 142
4.2.1.3 Conclusion ..... 144
4.2.2 Effective force between a pair of DNA molecules ..... 145
4.2.2.1 Methods ..... 146
4.2.2.2 Results ..... 148
4.2.2.3 Conclusion ..... 150
4.2.3 Osmotic Pressure calculation in hexagonal DNA bundle ..... 152
4.2.3.1 Notion of osmotic pressure ..... 153
4.2.3.2 Methods ..... 155
4.2.3.3 Influence of counterion charge on DNA lattice stability ..... 156
4.2.3.4 Influence of counterion radius on DNA lattice stability ..... 158
4.2.3.5 Conclusion ..... 158
4.3 Self-Assembly of DNA mediated by nanoparticles ..... 159
4.3.1 Aggregation of nanoparticles on a single fixed DNA ..... 159
4.3.1.1 Methods ..... 160
4.3.1.2 Adsorption of AuNPs on the DNA molecule ..... 164
4.3.1.3 Density of AuNPs and ions in the vicinity of the DNA ..... 166
4.3.1.4 DNA overcharging effect ..... 170
4.3.1.5 Distribution of AuNP on the DNA ..... 170
4.3.1.6 Conclusion ..... 173
4.3.2 Effect of counterions on the adsorption of AuNP on DNA molecule: PMF calculations ..... 174
4.3.2.1 Methods ..... 174
4.3.2.2 Result . ..... 175
4.3.2.3 Conclusion ..... 177
4.3.3 Redissolution of AuNPs from a single fixed DNA with excess of salt ..... 178
4.3.3.1 Methods ..... 178
4.3.3.2 Effect of the salt on the adsorption of AuNPs on DNA ..... 179
4.3.3.3 Density of cations in the vicinity of the DNA ..... 181
4.3.3.4 Conclusion ..... 181
4.3.4 Effective force between a pair of DNA ..... 182
4.3.4.1 Methods ..... 182
4.3.4.2 Results ..... 185
4.3.4.3 Conclusion ..... 188
4.3.5 Osmotic pressure calculation in hexagonal and square DNA phases ..... 189
4.3.5.1 Methods ..... 189
4.3.5.2 Results ..... 193
4.3.5.3 Conclusion ..... 195
5 Self-assembly in PEDOT:PSS solution by addition of ionic liquids for conduc- tivity enhancement. ..... 199
5.1 Choice of molecular model ..... 200
5.1.1 Atomistic models for PEDOT:PSS ..... 200
5.1.2 Atomistic models for ionic liquids ..... 201
5.2 Morphology of aqueous solution of PEDOT:PSS ..... 202
5.2.1 Methods ..... 203
5.2.2 Morphology of aqueous PEDOT:PSS complexes - small systems case ..... 205
5.2.3 Morphology of aqueous PEDOT:PSS complexes - larger systems case ..... 207
5.2.4 Conclusion ..... 209
5.3 Prediction of ion exchange between PEDOT:PSS/IL ..... 209
5.3.1 Methods ..... 211
5.3.2 Ion exchange free energy: PMF calculations ..... 212
5.3.3 Solvation of IL anions in US simulations. ..... 218
5.3.4 Conclusion ..... 221
5.4 Ionic exchange in mixed solutions: varying the IL anion ..... 221
5.4.1 Methods ..... 223
5.4.2 Morphology of mixed solution - small system cases ..... 224
5.4.2.1 Morphology of tri-EDOT:PTS solutions ..... 224
5.4.2.2 Morphology of tri-EDOT:16SS solution ..... 229
5.4.2.3 Morphology of 6EDOT:16SS solution ..... 233
5.4.3 Conclusion ..... 235
Conclusion and future work ..... 237
A Monte Carlo simulation package ..... 243
A. 1 Program to generate the coarse-grained system ..... 244
A.1.1 Initialization system.cpp ..... 244
A.1.2 Initialization system.h ..... 260
A.1.3 Makefile ..... 261
A. 2 Program to generate system topology ..... 262
A.2.1 Builiding topology.cpp ..... 262
A.2.2 Builiding topology.h ..... 270
A.2.3 Makefile ..... 272
A. 3 Program to perform a Monte Carlo simulation ..... 272
A.3.1 Main.cpp ..... 272
A.3.2 Move_MC.cpp ..... 275
A.3.3 Move_MC.h ..... 292
A.3.4 Potential_energy_calculation.cpp ..... 296
A.3.5 Potential_energy_calculation.h ..... 315
A.3.6 Cell_lists.cpp ..... 322
A.3.7 Cell lists.h ..... 329
A.3.8 MersenneTwister.h ..... 330
A.3.9 Makefile ..... 333
A. 4 Program to visualize a trajectory ..... 333
A.4.1 Main.cpp ..... 333
A.4.2 Visualization trajectory.cpp ..... 336
A.4.3 Visualization_trajectory.h ..... 340
A.4.4 Read_trajectory.cpp ..... 341
A.4.5 Read trajectory.h ..... 343
A.4.6 Makefite ..... 344
A. 5 Program to analyze a trajectory ..... 345
A.5.1 Main.cpp ..... 345
A.5.2 Analysis_trajectory.cpp ..... 347
A.5.3 Analysis_trajectory.h ..... 360
A.5.4 Makefile ..... 362
B Some tests for the Monte Carlo simulation package ..... 365
B. 1 Test of the Ewald summation technique ..... 365
B.1.1 Determination of Madelung constant for NaCl crystal ..... 365
B.1.2 Electrostatic interactions in water ..... 368
B. 2 Properties of the Kratky-Porod model ..... 369
B. 3 Test of the grand canonical MC scheme ..... 373
C Monte Carlo error analysis ..... 375
C. 1 Effective force between a pair of DNA with counterions ..... 377
C. 2 Osmotic pressure in a hexagonal bundle of DNA condensed with counterions ..... 380
C. 3 Effective force between a pair of DNA with AuNPs ..... 382
C. 4 Osmotic pressure in DNA lattices with AuNPs ..... 388
D Force-field parameters for PEDOT:PSS and ionic liquids ..... 395
D. 1 Force-field parameters for PEDOT:PSS ..... 395
D. 2 Force-field parameters for IL anions ..... 400
D. 3 Force-field parameters for IL cation ..... 405
E Morphology of mixed solution of PEDOT:PSS - large system case ..... 409
E. 1 Definiton of the complementary domain analysis ..... 409
E. 2 Morphology of tri-EDOT:PTS solution ..... 410
E. 3 Morphology of tri-EDOT:16SS solution ..... 416
E. 4 Morphology of 6EDOT:16SS solution ..... 422
E. 5 Morphology of 6EDOT:PTS ..... 428
F Supplemental material for the Umbrella Sampling calculation for tri-EDOT:Xand EMIM: $X$ ion pairs and ion exchange energy433
F. 1 Error on the PMF calculations for ion pairs and ion exchange energy ..... 433
F. 2 Umbrella sampling histograms. ..... 437
F. 3 PEDOT:X and EMIM:X complexes snapshots ..... 438
F. 4 Comparison between our PMFs/binding constant to data literature ..... 440

## List of Figures

1.1 Example of polyelectrolytes ..... 7
1.2 Chemical structure of algins ..... 8
1.3 Chemical structure of hyaluronate ..... 8
1.4 Chemical structure of chitosane ..... 9
1.5 General structure of an amino acid ..... 10
1.6 Detailed structure of the DNA compacted into chromosome ..... 11
1.7 Examples of diblock copolymers ..... 12
1.8 Scheme of an hydrophobic modified polyelectrolyte ..... 13
1.9 Example of a weak and a strong polyelectrolyte ..... 15
1.10 Water permanent dipole and ions ..... 16
1.11 The freely joined chain model ..... 22
1.12 Partition of Polyelectrolyte into electrostatic blobs ..... 26
1.13 Linear polyelectrolyte with the region of condensed counterions and region of free ..... 29
1.14 Plot of effective Manning parameter in function of the Manning parameter (Manning- Oosawa theory) ..... 32
1.15 Mixture of positive and negative polyelectrolytes ..... 34
1.16 Plot showing thermodynamics of polyelectrolyte complexation ..... 35
1.17 Schematic representation of the different stages of PECs formation ..... 37
1.18 Ladder-like structure adopted by water-soluble PECS ..... 38
1.19 Examples of structures of PECs with polyelectrolytes of different form and stiffness ..... 39
2.1 Three route for DNA compaction ..... 44
2.2 DNA compaction mode in fonction of the charge of compaction agent ..... 45
2.3 Scheme presenting correlations of compaction agent leading to DNA attraction ..... 48
2.4 Scheme presenting bridging of compaction agent leading to DNA attraction ..... 48
2.5 Plot showing the charge inversion of DNA ..... 49
2.6 Example of shapes that could be adopted after DNA compaction ..... 50
2.7 The Turkevich protocol to synthesize ligand-stabilized AuNPs ..... 52
2.8 The Brust protocol to synthesize ligand-stabilized AuNPs ..... 53
2.9 Scheme presenting AuNP as metal ion sensors ..... 54
2.10 AuNPs as building blocks in electronic devices ..... 55
2.11 Example of conductive wire made of AuNPs ..... 55
2.12 Curves of influence and time and AuNP concentration on AuNP adsorption on asingle DNA57
2.13 TEM images of close-packed assembly of gold nanoparticles on DNA ..... 58
2.14 Principle of DNA attachment process on a DNA origami by hybridization ..... 58
2.15 TEM images of different assemblies of AuNP/DNA through electrostatic interaction ..... 60
2.16 Crystal Assemblies of DNA origami and AuNPs ..... 60
2.17 Presentation of the carbon possible hybridization ..... 62
2.18 Example of some of the most used semi-conducting polymers ..... 63
2.19 Conductivity scale of various materials ..... 64
2.20 Bandgap diagram for insulators, semi-conductors and conductors ..... 65
2.21 Charge carriers along backbone of degenerated and non-degenerated systems ..... 66
2.22 Synthesis of PEDOT:Tos ..... 68
2.23 Chemical structure of PEDOT:PSS ..... 69
2.24 Structure of PEDOT:PSS complex ..... 69
2.25 Structure of PEDOT:Tos matrix ..... 70
2.26 Examples of applications of stretchable sensors for health monitoring ..... 72
2.27 Examples of common cations and anions that form ILs ..... 74
2.28 Structures of the ILs EMIMM:X used in the thesis ..... 75
2.29 Process of PEDOT:PSS film fabrication containing doping IL EMIM:X ..... 75
2.30 Electrical conductivity, Charge-carrier mobility and $\pi-\pi$ stacking distance of PE- DOT:PSS films without and after treatment with EMIIM:X. ..... 76
2.31 XPS spectra, 1D scattering profile in $\mathrm{q}_{z}$ direction and scheme for a possible reorga- nization of PEDOT:PSS matrix after insertion of ILs ..... 77
2.32 DFT-optimized geometries of PEDOT:X and EMIM:X complexes: binding free en- ergies and exchange free energies ..... 80
3.1 Application ranges of computational simulations for different scales ..... 84
3.2 Procedure of a canonical MC simulation ..... 89
3.3 Scheme of system in the isobaric-isothermal ensemble ..... 91
3.4 Scheme of system in the Grand-canonical ensemble ..... 92
3.5 Presentation of the PBCs in two dimensions ..... 99
3.6 Scheme explaining periodic boundary conditions ..... 101
3.7 Scheme explaining the principle of the Ewald summation technique ..... 102
3.8 Description of the Monte Carlo software ..... 110
3.9 Procedure of an MD simulation ..... 112
3.10 Schematic of the bonded interactions ..... 117
3.11 Scheme explaining PMF for system made of a guest and a host molecules ..... 122
3.12 Example of a free energy profile ..... 123
3.13 Example of reconstruction of free energy profile by the US technique ..... 126
3.14 General procedure to perform a simulation using GROMACS ..... 129
4.1 Representation of different CG DNA models ..... 134
4.2 Construction of the AuNP CG model ..... 136
4.3 Picture of the CG models of AuNP ..... 137
4.4 Snapshot of the simulation box with a MAM DNA its counterions and salt ions ..... 141
4.5 Picture of the three regions of DNA for which the density calculations of ions are performed ..... 142
4.6 Panoramic view of condensed monovalent salt cation density on the different DNA surface regions for $C_{s}=0.1 \mathrm{~mol} / \mathrm{l}$ ..... 143
4.7 Panoramic view of condensed monovalent salt cation density on the different DNA surface regions for $C_{s}=0.1-0.71-1.61 \mathrm{~mol} /$ ..... 144
4.8 Panoramic view of condensed monovalent, divalent and trivalent salt cation density145
4.9 Snapshot of a system made of a parallel pair of DNA with its monovalent counterions1 ..... 147
4.10 Plot of the effective force calculated between DNA with multivalent counterions. ..... 149
4.11 Projected densities of counterion in the $x O y$ plane of the simulation box of a pair of DNA with multivalent counterions ..... 151
4.12 Effective force calculated between a pair of DNA molecules with monovalent, diva- lent or trivalent counterions in presence of various amount of monovalent salt byAllahyarov and co-workers152
4.13 Scheme explaining the notio of osmotic pressure ..... 154
4.14 Snapshots of hexagonal DNA bundle compacte with multivalent counterions for osmotic pressure calculation ..... 156
4.15 Osmotic pressure calculated in hexagonal bundle of DNA compacted with counteri- ons of different valencies ..... 157
4.16 Osmotic pressure calculated in hexagonal bundle of DNA condensed by divalent counterions for different counterion diameters ..... 158
4.17 Definition of the coordinates used to calculate the helical projections ..... 163
4.18 Snapshots of the equilibrated systems containing one single DNA with AuNPs and ..... 165
4.19 Plot of adsorbed AuNPs on DNA for different $R_{+/-}$ ..... 166
4.20 Plot of adsorbed cations on DNA for different $R_{+/-}$ ..... 167
4.21 Coordination number of the different species for DNA/12-AuNP model ..... 168
4.22 Helical projection in the $x O y$ plane of the species density (ions and AuNPs) in the vicinity of the single DNA molecule ..... 169
4.23 Plot of the charge compensation $\theta(\mathrm{r})$ of the single DNA molecule by ions and AuNPs ..... 171
4.24 Histograms of the distance between adsorbed AuNPs on the single DNA ..... 172
4.25 Histogram of the biased distributions of the 30-AuNP along the reaction coordinate $O x$ defined between the DNA center ..... 175
4.26 PMF and internal energy variation between the 30-AuNP and the DNA decorated with 30-AuNPs ..... 176
4.27 Snapshot of the entropy-driven exchange upon adsorption of a 30-AuNP on the DNA molecule ..... 177
4.28 Plot of the desorption of AuNP and adsorption and salt for systems with a single DNA and excess of salt ..... 180
4.29 Plot of the helical projection of salt cations for systems with a single DNA and excess of salt ..... 181
4.30 Typical snapshots of a simulation with a pair of parallel DNA with AuNPs andrespective counterions184
4.31 Plot of the effective force between the parallel pair of DNA molecules with AuNPs and respective counterions ..... 185
4.32 Plot of the contribution of each force to the effective force between the parallel pair of DNA molecules with AuNPs and respective counterions ..... 187
4.33 Snapshots of hexagonal DNA bundle compacte with AuNPs for osmotic pressurecalculation192
4.34 Plot of the osmotic pressure in the DNA-AuNP bundles ..... 194
4.35 Projection of the densities of each species in the DNA bundle compacted with AuNPs1 ..... 196
5.1 Minimal model of PEDOT:PSS used in the MD simulations ..... 201
5.2 Larger-size model of PEDOT:PSS used in the MD simulation ..... 201
5.3 Imidazolium based ionic liquid EMIM: X used in the MD simulations ..... 202
5.4 Final snapshots of the equilibrated small aqueous PEDOT:PSS systems ..... 205
5.5 Cluster analysis for the small aqueous PEDOT:PSS systems ..... 206
5.6 Plot of $\mathrm{RDF}(\mathrm{C}-\mathrm{C})$ and $\mathrm{RDF}(\mathrm{C}-\mathrm{S})$ for the small aqueous PEDOT:PSS systems ..... 207
5.7 Final snapshots of the equilibrated large systems of PEDOT:PSS ..... 208
5.8 Plot of $\operatorname{RDF}(\mathrm{C}-\mathrm{C})$ and $\mathrm{RDF}(\mathrm{C}-\mathrm{S})$ for the small aqueous PEDOT:PSS systems ..... 209
5.9 Evolution of the number of clusters of PEDOT for the large aqueous PEDOT:PSS ..... 210
5.10 Snapshots of the pulling simulations in the case of tri-EDOT:TCB complexes ..... 211
5.11 PMF curves: tri-EDOT:X (left) and EMIM:X (right) ..... 212
5.12 Logarithm of the radial distribution function between the CM of the reference (tri- EDOT for Figure A or EMIM for Figure B) and the CM of the anion. ..... 216
5.13 Radial distribution function RDF (A-OW) calculated between atoms A of anion X ..... 218
5.14 Evolution of the average number of hydrogen bonds between water molecules adnthe most electronegative atoms of IL anions220
5.15 Radial distribution function RDF (A-OW) between the most electronegative atoms of IL anions at different CM-CM distance between anion and cation ..... 222
5.16 Final snapshots of small aqueous tri-EDOT:PTS systems treated with $\times 24$ pairs EMIM: X ..... 225
5.17 PEDOT Cluster analysis performed on small tri-EDOT:PTS systems treated with $\times 24$ pairs EMIIM:X ..... 226
5.18 Cluster composition analysis performed on small aqueous tri-EDOT:PTS systems treated with $\times 24$ pairs EMIM:ES or EMIM:TCB ..... 227
5.19 ..... 228
5.20 Plots of RDF(C-S) and CN(C-S) for small aqueous tri-EDOT:PTS systems treated with $\times 12$ or $\times 24$ pairs EMIM:X ..... 229
5.21 Final snapshots of small aqueous tri-EDOT:16SS systems treated with $\times 24$ pairs EMIM:X ..... 230
5.22 PEDOT Cluster analysis performed on small aqueous tri-EDOT:16SS systems treated with $\times 24$ pairs EMIM:X ..... 231
5.23 Cluster composition analysis performed on small aqueous tri-EDOT:16SS systemstreated with $\times 24$ pairs EMIM:ES or EMIM:TCB231
5.24 RDF(C-S) and RDF(C-C) for the small aqueous tri-EDOT:16SS system. ..... 232
5.25 Final snapshots of the small aqueous 6EDOT:16SS systems treated with $\times 64$ pairs ..... 233
5.26 Cluster composition analysis for the small aqueous 6EDOT:16SS system treated with $\times 64$ EMIM:ES or EMIM:TCB ..... 234
5.27 RDF(C-S) and RDF(C-C) for the small aqueous 6EDOT:16SS system. ..... 235
B. 1 NaCl crystal structure. ..... 366
B. 2 Energy contribution as a function of the splitting parameter $\alpha$ for $n_{c}=8$ ..... 367
B. 3 Madelung constant for NaCl crystal in function of the splitting parameter $\alpha$. ..... 368
B. 4 Scheme of the water molecule. ..... 369
B. 5 Illustration of the Kratky-Porod model ..... 370
B. 6 Angular correlation function of the bead chain model ..... 372
D. 1 Atomic names of tri-EDOT and PTS. ..... 396
D. 2 Atomic name description for IL anions ..... 400
D. 3 Atomic name description for EMIM ..... 405
E. 1 Final snapshots of large tri-EDOT:PTS aqueous systems treated with $\times 96$ pairs of EMIM:X. ..... 410
E. 2 Zoom-in of the final snapshots for large tri-EDOT:PTS aqueous systems treatedwith $\times 96$ pairs EMIM: X411
E. 3 Tri-EDOT cluster composition using the FoF alorithm for large tri-EDOT:PTS aque-
412 ous systems mixed with $\times 96$ pairs of IL
E. 4 Number of units in PEDOT domains in $\pi-\pi$ stacking using the PPS algorithm for large tri-EDOT:PTS aqueous systems without (red bars) and mixed with $\times 96$ pairs of IL (blue bars). ..... 414
E. 5 Plot of RDF(C-S) for large aqueous tri-EDOT:PTS systems mixed with $\times 96$ or $\times 192$pairs of IL EMIM:X415
E. 6 Plots of RDF(C-C) and CN(C-C) for large aqueous tri-EDOT:PTS systems mixed ..... 415
E. 7 Final snapshots for large tri-EDOT:16SS aqueous systems treated with $\times 96$ pairs EMIM:X. ..... 416
E. 8 Zoom-in of the final snapshots for large tri-EDOT:16SS aqueous systems treatedwith $\times 96$ pairs EMIM: X417
E. 9 Tri-EDOT cluster composition for tri-EDOT:16SS large aqueous systems mixed with $\times 96$ pairs of IL ..... 418
E. 10 Number of units in PEDOT domains in $\pi-\pi$ stacking using the PPS algorithm forlarge tri-EDOT:16SS aqueous systems without (red bars) and mixed with $\times 96$ pairsof IL (blue bars)419
E. 11 Plot of RDF(C-S) for tri-EDOT:16SS large systems mixed with $\times 96$ or $\times 192$ pairs420
E. 12 Plots of RDF(C-C) and CN(C-C) for tri-EDOT:16SS large systems mixed with $\times 96$ pairs of IL EMIM:X ..... 421
E. 13 Final snapshots of 6EDOT:16SS large aqueous systems treated with $\times 96$ pairs EMIM:X ..... 422
E. 14 Zoom-in of the final snapshots for large 6EDOT:16SS aqueous systems treated with$\times 96$ pairs EMIIM:X423
E. 15 6EDOT cluster composition for 6EDOT:16SS large aqueous systems mixed with $\times 96$ pairs of IL ..... 424
E. 16 Number of units in PEDOT domains in $\pi-\pi$ stacking using the PPS algorithm forlarge 6EDOT:16SS aqueous systems without (red bars) and mixed with $\times 96$ pairsof IL (blue bars)425
E. 17 Plot of RDF(C-S) for 6EDOT:16SS large aqueous systems mixed with $\times 96$ or $\times 192$426
E. 18 Plots of RDF(C-C) and CN(C-C) for 6EDOT:16SS large aqueous systems mixed with $\times 96$ pairs of IL EMIM:X. ..... 427
E. 19 Final snapshots of 6EDOT:PTS large aqueous systems treated with $\times 96$ pairs EMIM:X 42E. 20 Zoom-in of the final snapshots for large 6EDOT:PTS aqueous systems treated with$\times 96$ pairs EMIM:X429
E. 21 6EDOT cluster composition for 6EDOT:16SS large aqueous systems mixed with $\times 96$pairs of IL. Cluster analysis of big aqueous system of tri-EDOT:PTS in absence ofIL is displayed in red (6EDOT).430
E. 22 Number of units in PEDOT domains in $\pi-\pi$ stacking using the PPS algorithm forlarge 6EDOT:PTS aqueous systems without (red bars) and mixed with $\times 96$ pairsof IL (blue bars).431

E. 23 Plots of RDF(C-S) and RDF(C-C) for 6EDOT:PTS large aqueous systems mixed
with $\times 96$ pairs of IL EMIM:X

| F. 1 | Standard deviation of the potential of mean force calculated for tri-EDOT:X pairs. 435 |
| :--- | :--- | :--- |
| F.2 | St |

F. 2 Standard deviation of the potential of mean force calculated for EMIM:X pairs.436
F. 3 Histograms of the biased simulations for EMIM:ES ion pair. . . . . . . . . . . . . . 437
F. 4 Equilibrated complexes of tri-EDOT:X in our US simulations. . . . . . . . . . . . . 438
F. 5 Equilibrated complexes of EMIM-EDOT:X in our US simulations. . . . . . . . . . 439
F. 6 Comparison between PMF for EMIM:Cl and EMIM:ES calculated by us by Yee and
F. 7 Comparison between RDF and CN for EMIM:Cl and EMIM:ES calculated by us by Yee and co-workers . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 442
F. 8 CN calculated from RDF for EMIM:X and tri-EDOT:X in our work . . . . . . . . 443

## List of Tables

1.1 Table of amino acids ..... 9
1.2 Description of usual solvents and behavior of polyelectrolytes to be interacting with ..... 17
4.1 Table summarizing the number of ions and AuNPs in the system with a single fixed ..... 161
4.2 Systems for which the PMF is calculated between a $30-$ AuNP and the fixed DNA ..... 174
4.3 Amount of salt inserted in excess in system with a single fixed DNA and AuNPs . ..... 178
4.4 Amount of AuNPs and ions in the system containing 2 parallel DNA molecules ..... 183
4.5 Amount of salt in the DNA bundle compacted with AuNPs ..... 192
5.1 Concentration of PEDOT:PSS for small systems ..... 203
5.2 Concentration of PEDOT:PSS for large systems ..... 203
5.3 Ion binding and ion exchange free energies (kJ/mol) for PEDOT:PSS and IL EMIM:X213 ..... 213
5.4 Initial CM-CM distances, equilibrium CM-CM distances and association constantfor each complexes at $298 K$ for EMIM:X and tri-EDOT: X complexes.217
5.5 Average number of h-bonds between Water oxygen and the most electronegative atoms of IL anions ..... 220
5.6 Concentration of IL in small aqueous PEDOT:PSS systems ..... 223
5.7 Concentration of IL in large aqueous PEDOT:PSS systems ..... 224
B. 1 Calculation of the electrostatic interaction in water with the Ewald summation methodl ..... 369
B. 2 Calculation of persistence length of the bead chain model for different bending co-efficients.373
B. 3 Excess chemical potential of salt calculated with NVT MC simulation ..... 374
B. 4 Comparison between the number of salt pairs during NVT MC simulations and thenumber of salt pairs in GCMC simulations for which the chemical potential has beencalculated from the NVT MC simulations.374
C. 1 Effective force exerted between the DNA pair in presence of 120 monovalent coun-terions. The data are gathered during $4 \times 10^{5} \mathrm{MC}$ cycles and the force is printed ina file every MC cycle to calculate statistics.378
C. 2 Effective force exerted between the DNA pair in presence of 60 divalent counterions. The data are gathered during $4 \times 10^{5} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.
C. 3 Effective force exerted between the DNA pair in presence of 40 trivalent counterions. The data are gathered during $4 \times 10^{5} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.
C. 4 Effective force exerted between the DNA pair in presence of 30 tetravalent counterions. The data are gathered during $4 \times 10^{5} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.
C. 5 Osmotic pressure calculated in a hexagonal bundle of DNA with counterions of different valency $\mathrm{q}=+1,+2,+3$ or +4 e corresponding respectively to a number of $240,120,80$ and 60 counterions in the bundle. The diameter of the counterions is $\mathrm{d}_{c}=1 \AA$. The data are gathered during $4 \times 10^{5} \mathrm{MC}$ cycles to calculate statistics. 381
C. 6 Osmotic pressure calculated in a hexagonal bundle of DNA with divalent counterions ( $\times 120$ counterions in the bundle). The diameter of the counterions is varied from $\mathrm{d}_{c}=1,2,4$ and $6 \AA$. The data are gathered during $4 \times 10^{5} \mathrm{MC}$ cycles to calculate statistics.
C. 7 Effective force exerted between the DNA pair in presence of 6 -AuNPs at $R_{+/-}=$ 0.50 corresponding to $\times 20$ nanoparticles in the system. There is also $\times 240 \mathrm{DNA}$ positive monovalent counterions and $\times 120$ negative monovalent AuNP co-ions. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.
C. 8 Effective force exerted between the DNA pair in presence of 6 -AuNPs at $R_{+/-}=$ 1.00 corresponding to $\times 40$ nanoparticles in the system without small ions. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.
C. 9 Effective force exerted between the DNA pair in presence of 6 -AuNPs at $R_{+/-}=$ 1.00 corresponding to $\times 40$ nanoparticles in the system. There is also $\times 240$ DNA positive monovalent counterions and $\times 240$ negative monovalent AuNP co-ions. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.
C. 10 Effective force exerted between the DNA pair in presence of 6 -AuNPs at $R_{+/-}=$ 1.50 corresponding to $\times 60$ nanoparticles in the system. There is also $\times 240$ DNA positive monovalent counterions and $\times 360$ negative monovalent AuNP co-ions. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.
C. 11 Effective force exerted between the DNA pair in presence of 12 -AuNPs at $R_{+/-}=$ 0.50 corresponding to $\times 10$ nanoparticles in the system. There is also $\times 240$ DNA positive monovalent counterions and $\times 120$ negative monovalent AuNP co-ions. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.
C. 12 Effective force exerted between the DNA pair in presence of 12-AuNPs at $R_{+/-}=$ 1.00 corresponding to $\times 20$ nanoparticles in the system without small ions. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.
C. 13 Effective force exerted between the DNA pair in presence of 12 -AuNPs at $R_{+/-}=$ 1.00 corresponding to $\times 20$ nanoparticles in the system. There is also $\times 240$ DNA positive monovalent counterions and $\times 240$ negative monovalent AuNP co-ions. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.
C. 14 Effective force exerted between the DNA pair in presence of 12 -AuNPs at $R_{+/-}=$
1.50 corresponding to $\times 30$ nanoparticles in the system. There is also $\times 240$ DNA positive monovalent counterions and $\times 360$ negative monovalent AuNP co-ions. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics. ..... 387

|  |
| :--- |
|  |
|  |

C. 15 Osmotic pressure calculated in a square lattice of DNA condensed with $\times 406$ -AuNPs without salt $\left(R_{+/-}=1.00\right)$. The data are gathered during $\times 10^{6} \mathrm{MC}$ cyclesto calculate statistics.389
C. 16 Osmotic pressure calculated in a hexagonal lattice of DNA condensed with $\times 406$ -

AuNPs without salt $\left(R_{+/-}=1.00\right)$. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles | AuNPs without salt ( $R$ |
| :--- |
| to calculate statistics. | ..... 390

C. 17 Osmotic pressure calculated in a square lattice of DNA condensed with $\times 20$ 12- AuNPs without salt $\left(R_{+/-}=1.00\right)$. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles to calculate statistics. ..... 391
C. 18 Osmotic pressure calculated in a hexagonal lattice of DNA condensed with $\times 20$ 12- AuNPs without salt $\left(R_{+/-}=1.00\right)$. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles to calculate statistics. ..... 392
C. 19 Osmotic pressure calculated in a hexagonal lattice of DNA condensed with $\times 406$ - AuNPs $\left(R_{+/-}=1.00\right)$ with salt concentration of $\mathrm{c}_{\text {salt }}=30 \mathrm{mMol} / \mathrm{l}$. The data are gathered during $\times 10^{\circ} \mathrm{MC}$ cycles to calculate statistics. ..... 393
C. 20 Osmotic pressure calculated in a hexagonal lattice of DNA condensed with $\times 406$ - AuNPs $\left(R_{+/-}=1.00\right)$ with salt concentration of $\mathrm{c}_{\text {salt }}=60 \mathrm{mMol} / \mathrm{l}$. The data are gathered during $\times 10^{\circ}$ MC cycles to calculate statistics. ..... 394
C. 21 Osmotic pressure calculated in a hexagonal lattice of DNA condensed with $\times 406$ - AuNPs $\left(R_{+/-}=1.00\right)$ with salt concentration of $\mathrm{c}_{\text {salt }}=120 \mathrm{mMol} / \mathrm{l}$. The data are gathered during $\times 10^{\circ} \mathrm{MC}$ cycles to calculate statistics. ..... 394
D. 1 FF parameters: atomic charges taken from DFT calculations 201 , ..... 396
D. 2 FF parameters: Lennard-Jones vdW. ..... 397
D. 3 FF parameters: bond stretching. ..... 397
D. 4 FF parameters: angle bending. ..... 397
D. 5 FF parameters: Ryckaert-Bellemans dihedral torsion ..... 398
D. 6 FF parameters: Improper torsion ..... 399
D. 7 FF parameters: atomic charges taken from DFT calculations 201], ..... 401
D. 8 FF parameters: Lennard-Jones vdW. ..... 401
D. 9 FF parameters: bond stretching. ..... 401
D. 10 FF parameters: angle bending. ..... 402
D. 11 FF parameters: Ryckaert-Bellemans dihedral torsion ..... 403
D. 12 FF parameters: improper torsion. ..... 404
D. 13 FF parameters: atomic charges. ..... 405
D.14 FF parameters: Lennard-Jones vdW. ..... 405
D. 15 FF parameters: bond stretching. ..... 406
D. 16 FF parameters: angle bending. ..... 406
D. 17 FF parameters: Ryckaert-Bellemans dihedral torsion. ..... 407
D. 18 FF parameters: improper torsion. ..... 407
F. 1 Ion binding and ion exchange free energies ( $\mathrm{kJ} / \mathrm{mol}$ ). ..... 434
F. 2 Association constants for EMIM:Cl and EMIM:ES. Comparison between our results and results of Yee and Co-workers. ..... 444

Understanding the dynamics and self-assembly processes of polyelectrolytes in solution is currently an important issue, whether from a fundamental point of view or for industrial purposes [1] (2]. Polyelectrolytes constitute an universal class of objects grouping together any polymer or macromolecule comprising ionic sites in solution. If polyelectrolytes seem familiar to us because they are ubiquitous in everyday life (soap, food, drugs ...) and are also responsible for the transmission of genetic information through the DNA molecule, their description is no less very complicated [3].

This difficulty arises from a delicate compromise between a large number of physical interactions: there are not only the long-range electrostatic interactions exerted between the different ionic sites of the polyelectrolyte and between these sites and the free ions in solution, but also the short-range interactions between polyelectrolyte monomers (bonded interactions). Note also that the solvent plays a role on the conformation of the polyelectrolyte in solution [4]. All of these interactions make it difficult to construct theoretical models predicting the behavior of polyelectrolytes in solution and in extend their self-assembly [5]. However, it is admitted that spontaneous formation of polyelectrolyte complexes arises from electrostatic interactions between oppositively charged polyelectrolytes and this process is associated with the release of polyelectrolytes counterions from the surface of polyelectrolytes. The self-assembly processes are influenced by many parameters such as the intrinsic properties of polyelectrolytes (hydrophilic / hydrophobic character, molecular mass, rigidity, chemical nature of ionic sites as well as their distribution over the
polyelectrolytes ...) but also of the solvent ( pH , salt concentration ...). Due to all these parameters, the understanding of self-assembly processes and architecture of the resulted complexes is far from being understood.

However, the control of self-assembly processes open possibilities for the synthesis of new nanomaterials by a so-called bottom-up approach: it is a question of controlling the interactions governing the self-assembly of polyelectrolytes towards the design of assembled nanostructures. This approach makes it possible to overcome the limits of the so-called top-down approach failing at the molecular scale, when the nanostructures are elaborated by lithographic patterning techniques using short-wavelength sources [6] 7].

Although self-assembly phenomenon can be studied experimentally or using theoretical models, it is however possible to explore them with another complementary approach which is that of numerical simulation. It is the object of this thesis to study the stability or the formation of several polyelectrolyte complexes using numerical tools of statistical mechanics.

The first research project of this thesis consists in the study of the self-assembly of negatively charged DNA molecules with gold nanoparticles functionalized by ligands carrying a positive charge. Such assemblies make it possible the development of conducting nanowires or sensors [8]. Indeed, it is important to understand how the nanoparticles assemble on a single DNA molecule in order to probe the formation of more complicated structures containing nanoparticles and several DNA molecules.

The second project of this thesis is focused on the study of the self-assembly process of the conductive polymer poly(3,4-ethylenedioxithiophene) ("PEDOT") that adopts a polyelectrolyte behavior in presence of its counter-ionic negatively charged chain poly(styrene sulfonate) ("PSS"). The control of PEDOT:PSS morphology is an important prerequisite for its use in various applications in electronics (anode or hole transport in perovskite or organic solar cells) or bio-engineering (health or strain sensors) [9, 10]. Recently, it has been proven that the electronic transport properties of PEDOT:PSS can be improved by addition of imidazolium-based ionic liquids into PEDOT:PSS solutions [11, 12]. By using numerical simulation, we will compare the morphology of PEDOT:PSS aqueous systems before and after insertion of ionic liquids and we will explore the influence of
the nano-morphology of the resulted polyelectrolyte complex and their relation with the electrical transport properties observed experimentally at the macroscopic level.

This manuscript is organized in five chapters. The first chapter is a general presentation of polyelectrolytes. In particular, we introduce the different classes of polyelectrolytes (natural/artificial) and some of their relevant physical properties (charge fraction, condensation of counterions, hydrophobicity/hydrophilicty, role of the solvent). Finally, we provide a description of interactions and mechanisms that drive oppositively charged polyelectrolytes to form a complex (e.g. negative DNA in presence of cationic small spermine or spermidine biomolecules).

The second chapter summarizes the bibliography related to the two projects that constitutes this thesis. First, a review of DNA condensation is presented and we introduce some important applications based on gold nanoparticles. Then, we present key experiments that report how to control nanoparticles assemblies created on a single DNA and for complexes constituted of several DNA molecules or DNA superstructures like DNA origami. In a second part, we present generalities on conducting polymers as well as specific properties and applications of PEDOT conducting polymer. Lastly, after a short introduction to ionic liquids, relevant experiments on PEDOT:PSS conductivity enhancement in presence of different ionic liquids are presented.

In the third chapter, we present the molecular simulation tools used to sample the different configurations of the polyelectrolyte systems. First, we review the different molecular Monte Carlo simulation techniques that we have implemented in a home-made simulation package specially adapted for the first project. Second, we introduce a review of the molecular dynamics simulation technique used for the second project in order to probe dynamics and equilibrium properties of PEDOT:PSS solutions. Techniques introduced to compute interactions/forces between particles and to evaluate free energy are relevant to the two projects.

The fourth chapter presents results obtained from various Monte Carlo simulations. Before introducing the results obtained on DNA-nanoparticle systems, we rederive some results of the literature: the distribution of small ions around a DNA [13, 14], the interaction force between a pair of parallel DNA in presence of small multivalent ions [15, 16] and the stability of a DNA bundle as a function of the counterion valency [17. Then, we study the distribution of nanoparticles around a single DNA, estimate forces acting between parallel DNAs and determine bundle stability with nanoparticles of different charges.

The fifth chapter details the results obtained from molecular dynamics simulations of PEDOT:PSS solutions. In particular, we determine the morphology of un-treated PEDOT:PSS systems and compare it with morphologies obtain after injection of ionic liquids. We then deduce the design principles that an IL anion X must satisfy in order to exhibit high electrical transport properties by exploring the change in the nano-morphology of PEDOT:PSS.

## CHAPTER 1

## Nature and interactions of polyelectrolytes

## Contents

1.1 Generalities on polyelectrolytes6
1.1.1 Definition ..... 6
1.1.2 Natural polyelectrolytes ..... 7
1.1.3 Artificial polyelectrolytes ..... 10
1.1.4 Conclusion ..... 13
1.2 Some physical properties of polyelectrolytes ..... 13
1.2.1 Polyelectrolytes in solution ..... 14
1.2.2 Electrostatic interactions in aqueous solution ..... 18
1.2.3 Physics of hydrophilic polyelectrolytes in solution ..... 21
1.2 .4 Conclusion ..... 32
1.3 Complexes of polyelectrolytes ..... 33
1.3.1 Interactions driving polyelectrolyte complexation ..... 33
1.3.2 Mechanism of complex formation ..... 34
1.3.3 Possible structures of polyelectrolytes ..... 36
1.3.4 Conclusion ..... 39

Polyelectrolytes are polymers containing repeating unit blocks called monomers that carry electrostatic charges. Polyelectrolytes are very common in daily life. For instance, we are using polyelectrolytes to clean ourselves with soap and shampoo, prepare food in industry with gelatin or thickening agents, or use them to fabricate paper on which this thesis is printed. Also, various industries such as cosmetics, pharmacetics, water treatment are using polyelectrolytes. Beyond industrial uses, our heredity is stored in the DNA molecule which is also a polyelectrolyte.

There exists a throng of polyelectrolytes that scientists and industries classify depending of their origin (natural or synthetic), charge, flexibility and so on. What are the different polyelectrolytes that can be found in nature or be synthesized? What are their properties? How do polyelectrolyte complexes form in solution?

In this chapter, we first review some important natural and synthesized polyelectrolytes. Then we provide a physical description of polyelectrolytes such as their conformations in solution as well as some elements of electrostatics. Finally, we review the process of polyelectrolyte complex formation and the resulting structures that can emerge.

### 1.1 Generalities on polyelectrolytes

### 1.1.1 Definition

Let us consider a polyelectrolyte made of $N$ monomers containing only one ionizable group for simplification. In general, depending on the solvent in which the polyelectrolyte is introduced, only a fraction $f N$ of the groups will be ionized, leaving a fraction of $(1-f) N$ non-ionized monomers 4. If cations $\mathrm{C}^{+}$are released in solution, the resulting polyelectrolyte is qualified as anionic. On the contrary, if the anions $\mathrm{A}^{-}$are released, the polyelectrolyte is qualified as cationic. The two cases are exhibited in Figure 1.1 The ions that are solvated are called the counterions of the polyelectrolyte chain.

## Weakly and highly charged polyelectrolytes

Weakly charged polyelectrolytes are characterized by a dissociation degree $f \ll 1$. In such case, the number of charged groups along the polyelectrolyte is low compared to the total number of monomers. When all counterions have been solvated, the dissociation degree $f$ is high and the remaining polyelectrolyte is highly charged.
a) Cationic polyelectrolyte:


No dissociation
b) Anionic polyelectrolyte:


No dissociation


Total dissociation


Figure 1.1 - Example of polyelectrolytes represented by black backbone lines linked with cations and anions representing ionic sites. Red beads refer to cations $\mathrm{C}^{+}$while green beads refer to anions $\mathrm{A}^{-}$. If the polyelectrolyte remains in air or in apolar solvent, the polyelectrolyte is not dissociated. However in polar solvents, ions are solvated and result in full charged polymer chain: the polyelectrolyte.

### 1.1.2 Natural polyelectrolytes

### 1.1.2.1 Polysaccharides

Polysaccharides are long chains of carbohydrate molecules, more specifically made of repeating monosaccharide units that can have chemical formula $\mathrm{C}_{n}\left(\mathrm{H}_{2} \mathrm{O}\right)_{m}$.

Polysaccharides are organic molecules that intervene in life cycle of numerous species. For instance, glucose is essential for metabolism of the whole majority of organisms. Plants synthesize various types of polysaccharides from water, carbon dioxyde and solar energy (photosynthesis). Cellulose constitutes wall of cell and amidon is a supply energy reserve for various cultivated plants like seeds (wheat, barley or corn) or tubers (potatoes). Animals also have an energy supply
which is glycogen. Such polymers are neutrals but most of the existing polysaccharides are charged and we provide some examples below.

- Algins are anionic polysaccharides extracted from the cell walls of brown algae that form viscous gum when hydrated in presence of divalent cations. Algins are presented as linear copolymers composed of two saccharide units which are the $\alpha$-L-guluronate and the $\beta$-Dmannuronate shown in Figure 1.2


Figure 1.2 - Chemical structure of algins.

Algins are used in food industries (ice creams, dairy products) because of their gelling and thickening properties.

- Hyaluronate is a linear anionic polysaccharide composed of sequence of units that contain two saccharides which are glucuronate and N -acetyl-glucosamine.


Figure 1.3 - Chemical structure of hyaluronate.

The structure of hyaluronate is presented in Figure 1.3 This polysaccharide is a major compound of the extra-cellular matrix in mammals and is present in various tissues such as the synovial liquid in articulations. Hyaluronate has wide applications in medicine because of its healing properties, it is used as an aid in ophtalmic surgery such as cataract extraction or applied as injection to treat osteoarthrisis located in articulations

- An example of cationic polysaccharide is the chitosan which is very common in environment and is found mainly in crustacean shells. The positive charges come from the ammonium groups located in the $\beta$-glucosamine units (Figure 1.4).


Figure 1.4 - Chemical structure of chitosane.

### 1.1.2.2 Polypeptides and proteins

Table 1.1 - Table of the 20 amino acids that can be found in nature ${ }^{a}$.

| Amino acid Name | Symbols | Nature of group R | Nature of polypeptide |  |
| :---: | :---: | :---: | :---: | :---: |
| Aspartic acid | Asp | D | Acid | Anionic and hydrophilic |
| Glutamic acid | Glu | E | Acid | polyelectrolyte (pH $>$ pKa) |
| Cysteine | Cys | C | Acid |  |
| Arginine | Arg | R | Basic | Cationic and hydrophilic |
| Histidine | His | H | Basic | polyelectrolyte (pH $<$ pKa) |
| Lysine | Lys | K | Basic |  |
| Glutamine | Gln | Q | Non-charged polar |  |
| Asparagine | Asn | N | Non-charged polar |  |
| Proline | Pro | P | Hydrophobic | Neutral polymer, hydrophilic to |
| Serine | Ser | S | Non-charged polar | moderately hydrophilic |
| Threonine | Thr | T | Non-charged polar |  |
| Tyrosine | Tyr | Y | Non-charged polar |  |
| Tryptophane | Trp | W | Hyrophobic |  |
| Glycine | Gly | G | Non-charged polar | Amphiphilic and neutral |
| Alanine | Gln | A | Hydrophobic |  |
| Methionine | Asn | M | Hydrophobic | Neutral hydrophobic |
| Valine | Pro | V | Hydrophobic |  |
| Leucine | Ser | L | Hydrophobic | Hydrophobic |

${ }^{a}$ We give their symbols and their chemical characteristics.

There exists 20 different amino acids in nature and their structures differ only by the group $R$ (Figure 1.5) that gives to the amino acids their chemical properties.

An amino acid can establish a peptide bond by linking its carboxylic acid function to the amine function of another amino acid. A protein is defined as a linking of huge number of amino acids.


Figure 1.5 - General structure of an amino acid.

We give in Table 1.1 the list of the different amino acids and their characteristics.

Among the 20 amino acids, 6 are acid/basic which means that they carry a charge when they are solvated. Proteins that contain such amino acids are thus polyelectrolytes and are very important for a range of biological functions. For instance, these proteins can adsorb on opposite charged surface or can mediate interactions in the living cells. Examples are histones that are charged proteins condensing DNA in eucaryotes cells or protamines condensing DNA in sperm cells.

### 1.1.2.3 Polynucleotides

One of the most important polynucleotide is the DNA molecule which forms a double helix. The DNA is a highly charged molecule that contains 2 negative charges per base pair of height $3.4 \AA$ resulting in a high charge density of $0.58-|\mathrm{e}| / \AA$. The total length of the DNA molecule is about $\sim 1.8 \mathrm{~m}$ for the human species and it is unlikely that so highly charged and long chain can be packed directly in the nucleus of eucaryotic cells.

Fortunately, the nucleus contains histones that are an assembly of eight positively charged proteins (with large amount of lysine and arginine amino acids) carring a charge of $\sim+150|\mathrm{e}|$. The cationic histones wrap up the negative DNA in a very compact state in the cell nucleus: the chromosome (Figure 1.6).

### 1.1.3 Artificial polyelectrolytes

Although polyelectrolytes are widely present in environment, their extraction require a huge effort of purification. Natural polyelectrolytes involve often complex and irregular structures such as ramifications or polydispersity. Moreover, the diversity of the polyelectrolytes is limited in the environment. Consequently, most of the polyelectrolytes used in daily life are artificial. Synthesized


Figure 1.6 - Detailed structure of the DNA compacted into chromosome. At the end of the metaphase, complexation of DNA and histones takes place in a way that DNA wraps up around histones to form the chromatin. The chromatin is compacted in itself to form the chromosome. Reprinted from ref. [18] (Nature Education, 2008, 1(1), 26, Figure 1, Copyright 2008 with permission from Nature publishing group).
polyelectrolytes present often high regularity in monomer sequence and high degree of purity and can be synthesized in huge quantity. They are synthesized either by polymerization of ionisable monomers or by ionisation of a whole polymer chain (19).

Industrial processes aimed to design "clean" polyelectrolytes that can be dissolved in water rather than organic solvent. Nowadays, artificial polyelectrolytes are used for many applications:

- Gelling agent in food industries (ice-creams, candies...)
- Dispersant agent to separate cellulose from lignin in paper fabrication industries.
- flocculation agent in water treatment.

However, natural polyelectrolytes seem to be mandatory for specific applications because of their bio-compatibility and bio-degradation, they are used for medical applications such as implants or drug delivery systems [20].

A class of important artificial polyelectrolytes are the amphiphilic polyelectrolytes which present both hydrophobic and hydrophilic groups. We distinguish two main classes of amphiphilic polyelectrolytes that we provide a description below.

### 1.1.3.1 Diblock copolymers

Diblock copolymers are the most simple amphiphilic polyelectrolyte composed of two blocks: one block is a hydrophilic chain and the other one is a neutral hydrophobic chain.


Figure 1.7 - Diblock copolymers contain both a hydrophobic and a hydrophilic block. In solution, they can assemble into various objects such as micelles, vesicules or bilayers.

One of the eldest manmade amphiphilic polyelectrolyte is soap based on amphiphilic polyelectrolyte of formula $\mathrm{R}-\mathrm{COO}^{-}$with hydrophobic part trapping fat in complexes that are washed out with water thanks to its hydrophilic part. In solution, diblock copolymers can assemble into various objects depending of the nature of the solvent and intrinsic properties of the polyelectrolyte [21]. The most common structures are the micelles, the vesicules and the bilayers presented in Figure 1.7. The vesicule models properly the plasmic membrane of a cell. The double layer of amphiphilic polyelectrolyte isolates cytoplasm from the outer environment. Pharmaceutic scientists aim to use vesicules as a drug delivery system. The idea is to trap the active principle into a vesicule to target specific cells and deliver through the cells membrane antigenic molecules or antibodies [22]. Surfactants are amphiphilic molecules that diminish surface tension existing between a polar and an apolar mixture so that a resulting more homogeneous mixture emerges 23].

### 1.1.3.2 Hydrophobic modified polyelectrolytes

Hydrophobic modified polyelectrolytes (HMP) can be represented as a polyelectrolyte backbone on which are attached highly hydrophobic segments (Figure 1.8] [24].


Figure 1.8 - Scheme of an hydrophobic modified polyelectrolyte.

HMPs tend to associate in aqueous solution at high concentration yielding to a mixture of polyelectrolytes, with hydrophobic segments playing the role of bridges. Some applications of HMP are fabrication of gels to use in painting or cosmetics industries.

### 1.1.4 Conclusion

Life is made of polyelectrolytes. The charged nature of polyelectrolytes made them essential for life on Earth. Polyelectrolytes became very important in modern human society from synthesis of various chemicals to production of food or pharmaceutics. Industrial formulation of polyelectrolytes enable to synthesize a whole range of polyelectrolytes with specifically designed applications.

Unfortunately, from a physical point of view, polyelectrolytes are very difficult to describe and require simplifications to properly capture their behavior at the expense of their chemical properties. In the next section, we give some elements of the physics that govern polyelectrolyte behavior in solution.

### 1.2 Some physical properties of polyelectrolytes

Polyelectrolytes are polymers carrying charges and it is relevant to determine their physical properties in solution. Polyelectrolytes represent a huge range of molecules that can have very different physical properties. It is thus required to distinguish between different classes of poly-
electrolytes with different physical descriptions. Polyelectrolytes are subject to two major types of physical interactions that govern their behavior: the long-range electrostatic interactions between ionic sites - ionic sites and ionic sites - free ions in solution and the short-range interactions between monomers of the polyelectrolyte.

This section is organized as followed. First, we present the different classes of polyelectrolytes in solution, more precisely hydrophilic/hydrophobic and strong/weak polyelectrolytes. Then, the electrostatic description of aqueous solutions will be presented as well as the physical principles that govern the conformations of polyelectrolytes.

### 1.2.1 Polyelectrolytes in solution

### 1.2.1.1 Strength of polyelectrolytes

## Weak polyelectrolytes

Polyelectrolytes in water contain ionic sites that result from the dissociation of counterions in solvent. Polyelectrolytes that do not completely dissolve in solution have a fraction $f$ of sites that are ionized at equilibrium depending of the nature of the solvent. We qualify such polyelectrolytes as "weak" 4].

A model of weak polyelectrolyte is composed of a homopolymer chain that can release or adsorb proton $\mathrm{H}^{+}$and theory of $\mathrm{Br} ø$ nsted-Lowry for acids and bases can be applied.

- If the polyelectrolyte carries basic functions, it is mostly negatively charged and presents an affinity with protons $\mathrm{H}^{+}$. When pH decreases, $f$ decreases as well because the polyelectrolyte is neutralized.
- On the contrary, if the polyelectrolyte carries acid functions, the polyelectrolyte is cationic and neutralizes itself by releasing $\mathrm{H}^{+}$in solution with increasing pH .

The most common example of a weak polyelectrolyte is the poly(acrylic acid) noted PAAH (Figure 1.9). The carboxilic function (-COOH) has a $p K_{a} \sim 5$ and is mostly dissociated at basic pH .

## Strong polyelectrolytes

## Weak polyelectrolyte:



Poly(acrylic acid) - PAAH

Strong polyelectrolyte:



Poly(styrenesulfonate acid) - PSSH

Figure 1.9 - Example of a weak and a strong polyelectrolyte.

Strong polyelectrolytes are polyelectrolytes for which all monomers are ionized in aqueous solution $(f=1)$. Strong polyelectrolytes are typically monomers that carry strong acid groups that release $\mathrm{H}^{+}$in solution. The common example of a strong polyelectrolyte is the poly(styrenesulfonate acid) presented in Figure 1.9 for which every sulfonic group $\mathrm{SO}_{3} \mathrm{H}$ units are deprotonated into $\mathrm{SO}_{3}^{-}$ groups.

### 1.2.1.2 Dissociation of a ionomer in solution yields to polyelectrolyte

## Role of the solvent

Polyelectrolytes are studied or used for industrial purpose in solution. The solvent controls the polyelectrolyte behavior in solution and we focused on two important parameters:

- The relative permittivity $\epsilon_{s}$ measures how the solvent reduces the electrostatic interaction exerted between two charges denoted by $\mathrm{A}^{-}$and $\mathrm{C}^{+}$separated from a distance $\mathrm{r}_{A C}$ :

$$
\begin{equation*}
\mathcal{U}_{A^{-} \rightarrow C^{+}}^{\text {solvent }}=\frac{\mathcal{U}_{A^{-} \rightarrow C^{+}}^{\text {vacuum }}}{\epsilon_{s}}=\frac{e^{2} Z_{A^{+}} Z_{C^{-}}}{4 \pi \epsilon_{o} \epsilon_{s} r_{A C}} \tag{1.1}
\end{equation*}
$$

where $\epsilon_{o}=8.84 \times 10^{-12} \mathrm{C}^{2} \mathrm{~N}^{-1} \mathrm{~m}^{-2}$ and $e$ is the elementary electric charge $\left(e=1.6 \times 10^{-19}\right.$ C). Consequently, depending on the relative dielectric constant, the dissociation of ions will vary and ionizable polymers will be either totally ionized and adopt a polyelectrolyte behavior or keep its neutral monomers and remain uncharged to adopt an ionomer behavior. For
instance, water has a high relative permittivity of $\epsilon_{s}=78.5$ and thus dissociates properly ions pairs.

- In addition to regulate intensity of electrostatic interactions between charges, solvent have a dipole moment $\mu$.

Solvation of anion in water:


Solvation of cation in water:


Figure 1.10 - Existence of water permanent dipole moment allows good solvation of cations and anions through dipole-charge interactions.

Let us consider water as an example since it is one of the most important solvent in chemistry. The water molecule has a well-known structure with an oxygen atom surrounded by two lone pairs and linked with two hydrogen atoms forming tetrahedron with an angle H-O-H of $\phi=$ $109.5^{\circ}$. Each water molecule represents a dipole of norm $\mu_{O H}=1.5 D$ where $D=3.33 \times 10^{-30}$ C.m is the usual dipole moment unit in chemistry. The resulting dipole moment pointing from oxygen to hydrogen writes:

$$
\begin{equation*}
\mu=2 \times \mu_{O H} \times \cos (\alpha)=1.85 D \tag{1.2}
\end{equation*}
$$

Not only water dissociates most of polyelectrolytes (high dielectric constant) but solvates properly (high dipole moment) cationic or anionic species (Figure 1.10). Hence, polyelectrolyte at high charge fraction are soluble in water and present a total dissociation of charges (4].

## Quality of the solvent

We display the relative permittivity $\epsilon_{s}$ of several usual solvents as well as their dipole moment $\mu$ in Table 1.2. Their intrinsic properties and the behavior of ionisable polymers to be solvated is also provided.

Table 1.2 - Description of usual solvents and behavior of polyelectrolytes to be interacting with.

| Solvent | $\mu$ | $\epsilon_{s}$ | Nature of solvent | Nature of neutral Monomer | Behavior |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Toluene | 0 | 2.38 | apolar and aprotic | Solvable | Ionomer |
| Water | 1.85 | 78.5 | Polar, hydrogen bond | Hydrophilic | Hydrophilic PE |
|  |  |  | Hydrophobic | Hydrophobic PE |  |
| DMF | 3.82 | 37.8 | Polar, aprotic | Solvable | Solvophilic PE |
| Acetonitrile | 3.92 | 36.6 | Polar, aprotic | Non solvable | Solvophobic PE |

Ionisable polymers adopt a different behavior depending of the solvent they interact with. For each solvent, effective permittivity $\left(\epsilon_{s}\right)$ and dipole moment is given $(\mu)$ at 293 K . The table is adapted from ref. [4] (Phd Thesis, 2003, 35, Table 2-1, Copyright 2003 with permission from University Paris VI).

Most of the organic solvents are apolar or lightly polar and provide mediocre solvation for charged species. They also display very small relative permittivity as shown in the case of toluene reported in Table 1.2 In that case, monomers are not ionized and remain in the form of ion pairs (ionomer).

A polyelectrolyte is qualified of hydrophilic (respectively hydrophobic) if the interactions with water molecules are thermodynamically more favorable (respectively less favorable) than interactions with other hydrophobic (respectively hydrophilic) solvents.

Remark: The notion of hydrophilic/hydrophobic polyelectrolyte in water can be generalized in case of other solvents: the polyelectrolyte is qualified of solvaphilic/solvaphobic.

Some solvents display higher dipole moment than water but a reduced relative permittivity due to their aprotic nature (absence of hydrogen bonds) such as $\mathrm{N}, \mathrm{N}$-dimethylformamide (DMF) or acetonitrile. They are less efficient to dissociate ionomers into polyelectrolytes and counterions.

Soluble polyelectrolytes release a high amount of ions in solution. This effect motivates a suitable description of electrostatic interactions in solution that we provide in the following section.

### 1.2.2 Electrostatic interactions in aqueous solution

### 1.2.2.1 Scale of electrostatic interactions in water

We consider an aqueous solution at temperature $T$ and of relative dielectric constant $\epsilon_{s}$ that contains two opposite charges of value $+|\mathrm{e}|$ and $-|\mathrm{e}|$. The electrostatic potential between the two charges writes as a function of the separating distance $r$ :

$$
\begin{equation*}
\mathcal{U}(r)=\frac{e^{2}}{4 \pi \epsilon_{s} \epsilon_{o} r} \tag{1.3}
\end{equation*}
$$

We introduce the Bjerrum length [25] denoted by $l_{B}$, corresponding to the distance for which the energy interaction between charges is equal to the thermal agitation $k_{B} T$ :

$$
\begin{equation*}
l_{B}=\frac{e^{2}}{4 \pi \epsilon_{s} \epsilon_{o} k_{B} T} \tag{1.4}
\end{equation*}
$$

The electrostatic potential can be written in a different way:

$$
\begin{equation*}
\mathcal{U}(r)=\frac{l_{B}}{r} k_{B} T \tag{1.5}
\end{equation*}
$$

The value of $l_{B}$ is $7.1 \AA$ in water at room temperature. The electrostatic interaction dominates thermal fluctutations at length-scale smaller than $l_{B}$.

### 1.2.2.2 Modeling electrostatic interactions in solution

We studied in the previous section the role of water as a solvent that dissociates and solvates charges. In addition, water contains charges through autoprotolysis reaction so that pure water contains also charges. Hence, it is required to introduce a proper theory that describes well electrostatics of charges in solution.

Let us consider an aqueous solution represented by its relative dielectric constant $\epsilon_{s}$ that contains $n$ different species of charges $q_{i}(\mathrm{i}=1, \ldots, \mathrm{n})$. The density of charges in the system is linked
to the electric field through the first Maxwell equation:

$$
\begin{equation*}
\vec{\nabla} \cdot \mathbf{E}(\mathbf{r})=\frac{\rho(\mathbf{r})}{\epsilon_{s} \epsilon_{o}} \tag{1.6}
\end{equation*}
$$

By replacing the electric field by electric potential $(E(\mathbf{r})=-\vec{\nabla} \phi(\mathbf{r}))$, we obtain the Poisson equation:

$$
\begin{equation*}
\Delta \phi(\mathbf{r})=-\frac{\rho(\mathbf{r})}{\epsilon_{s} \epsilon_{o}} \tag{1.7}
\end{equation*}
$$

The total charge density of the system is the sum of the charge density for every species:

$$
\begin{equation*}
\rho(\mathbf{r})=\sum_{i=1}^{n} c_{i}(\mathbf{r}) q_{i} \tag{1.8}
\end{equation*}
$$

At equilibrium, the concentration of species $i$ of charge $q_{i}$ obeys the Boltzmann statistics:

$$
\begin{equation*}
c_{i}(\mathbf{r})=c_{i}^{o} \exp \left(-\frac{q_{i} \phi(\mathbf{r})}{k_{B} T}\right) \tag{1.9}
\end{equation*}
$$

The combination of equations 1.7 and 1.9 yields the Poisson-Boltzmann equation:

$$
\begin{equation*}
\Delta \phi(\mathbf{r})=-\sum_{i=1}^{n} \frac{c_{i}^{o} q_{i}}{\epsilon_{s} \epsilon_{o}} \exp \left(-\frac{q_{i} \phi(\mathbf{r})}{k_{B} T}\right) \tag{1.10}
\end{equation*}
$$

This equation is not linear and cannot be solved analytically excepted for few specific cases. However, if the electrostatic energy $q_{i} \phi(\mathbf{r})$ is small compared to thermal agitation energy $k_{B} T$, the Poisson-Boltzmann equation can be linearized at the first order of the exponential term. The obtained equation is the Debye-Hückel equation:

$$
\begin{equation*}
\Delta \phi(\mathbf{r})=\frac{1}{\lambda_{D}^{2}} \phi(\mathbf{r}) \tag{1.11}
\end{equation*}
$$

where a characteristic length $\lambda_{D}$ appears:

$$
\begin{equation*}
\lambda_{D}=\left(\frac{1}{\epsilon_{s} \epsilon_{o} k_{B} T} \sum_{i=1}^{n} c_{i}^{o} q_{i}^{2}\right)^{-\frac{1}{2}} \tag{1.12}
\end{equation*}
$$

The Debye length $\lambda_{D}$ gives the scope of the electrostatic interaction in a given solution.
The Debye-Hückel equation admits an analytical solution in spherical coordinates. In that case, the electrostatic potential created by a charge in space depends on the radial distance $r$ :

$$
\begin{equation*}
\frac{1}{r^{2}}\left(\frac{\partial}{\partial r}\left(r^{2} \frac{\partial \phi(r)}{\partial r}\right)\right)=\frac{1}{r} \frac{\partial^{2}(r \phi(r))}{\partial r^{2}}=\frac{1}{\lambda_{D}^{2}} \phi(r) \quad \Longrightarrow \quad \frac{\partial^{2}(r \phi(r))}{\partial r^{2}}=\frac{1}{\lambda_{D}^{2}} r \phi(r) \tag{1.13}
\end{equation*}
$$

The solution of the equation 1.13 is proportional to a decaying exponential term that model the screening of the charge by the surrounding charged species. The potential of a charge $q$ is written as:

$$
\begin{equation*}
\phi(r)=\frac{q}{4 \pi \epsilon_{s} \epsilon_{o} r} \exp \left(-\frac{r}{\lambda_{D}}\right) \tag{1.14}
\end{equation*}
$$

Remark: For an aqueous solution of rock salt, the Debye length is expressed simply as a function of salt concentration:

$$
\begin{equation*}
\lambda_{D}=\frac{3 \AA}{\sqrt{c_{s}(\mathrm{~mol} / L)}} \tag{1.15}
\end{equation*}
$$

An increase of ion concentration results in a shorter Debye length. In pure water, autoprotolysis involves presence of $\mathrm{H}_{3} \mathrm{O}^{+}$and $\mathrm{OH}^{-}$ions at a concentration of $\mathbf{c}=10^{-7}$
mol/l. The calculated Debye length is $\lambda_{D}=970 \mathrm{~nm}$.

### 1.2.3 Physics of hydrophilic polyelectrolytes in solution

We present the physics involved in the description of hydrophilic polyelectrolytes which constitute a reference class for the study of other classes such as hydrophobic polyelectrolytes [4]. The connectivity of the chain is taken into account because of the monomer-monomer interactions. The goal is not to provide an exhaustive review of theoretical or experimental researches but only to present the main results. As a starting point, we present the conformation of a neutral chain in solution and we introduce the concept of electrostatic blob to treat conformation of a weakly charged polyelectrolyte. Finally, the condensation of counterions on a polyelectrolyte is presented with the Manning-Oosawa theory.

### 1.2.3.1 Conformation of a neutral chain in solution

## The freely joined chain model

The simplest description of a neutral chain consists of the freely joined chain model [26]. The chain is ideal because it is not possible to distinguish monomer-monomer and monomer-solvent interaction. One qualifies the chain to be in a $\theta$ solvent (4].

A good measure of the size occupied by the polymer relies on the calculation of the end-to-end distance. As an illustration, if the polymer forms a entangled complex in solution, the end-to-end distance will give a proper estimation of the entangled complex size.

The polymer is represented as a continuous chain made of $N$ links of length a modeling monomers. Hence, every link points independently in every directions so that bond orientations are uncorrelated. If $\mathbf{r}_{i}$ is the vector associated to a link of index $i$, noncorrelation is implied by the following relations:

$$
\left\{\begin{array}{lll}
\left\langle\mathbf{r}_{i} \cdot \mathbf{r}_{j}\right\rangle=0 & \text { if } & i \neq j  \tag{1.16}\\
\left\langle\mathbf{r}_{i} \cdot \mathbf{r}_{j}\right\rangle=1 & \text { if } & i=j
\end{array}\right.
$$

It is straightforward to estimate the norm of the end-to-end vector for the ideal chain $R_{\text {ideal }}$ :

$$
\begin{align*}
R_{\text {ideal }}= & \left(\sum_{n, m=1}^{N}<\mathbf{r}_{n} \cdot \mathbf{r}_{m}>\right)^{\frac{1}{2}}  \tag{1.17}\\
R_{\text {ideal }}= & \left(\sum_{n=1}^{N}<\mathbf{r}_{n}^{2}>+2 \sum_{n>m}<\mathbf{r}_{n} \cdot \mathbf{r}_{m}>\right)^{\frac{1}{2}}  \tag{1.18}\\
R_{\text {ideal }}= & \left(N a^{2}\right)^{\frac{1}{2}}  \tag{1.19}\\
R_{\text {ideal }}= & a N^{\frac{1}{2}} \tag{1.20}
\end{align*}
$$

The end-to-end distance is $R_{\text {ideal }} \sim N^{\frac{1}{2}}$ for the model of ideal chain. Although the freely joined chain is the simplest model that describes a polymer, more sophisticated models exhibit also an end-to-end distance $R \sim N^{\frac{1}{2}}$. We provide an example below.

## The freely rotating chain

The nature of chemical bonds can impose a given angle between bonds of monomers: the freely rotating chain describes this situation [26]. The model is presented in Figure 1.11
a) The freely joined chain

b) The freely rotating chain


Figure 1.11 - a) Model of the freely joined chain. b) Model of the freely rotating chain where the angle $\alpha$ is imposed and constant. The projection of $\mathbf{r}_{n}$ on $\mathbf{r}_{n-1}$ gives $<\mathbf{r}_{n} \cdot \mathbf{r}_{n-1}>=\mathbf{r}_{n-1} \cos (\alpha)$.

It is also visible that for the freely rotating chain model (Figure 1.11), we obtain the following relationship:

$$
\begin{align*}
<\mathbf{r}_{n-1} \cdot \mathbf{r}_{n}> & =\cos (\alpha)<\mathbf{r}_{n-2} \cdot \mathbf{r}_{n}>  \tag{1.21}\\
& =(\cos (\alpha))^{2}<\mathbf{r}_{n-3} \cdot \mathbf{r}_{n}> \tag{1.22}
\end{align*}
$$

This recursion relation can be generalized with condition that $\left\langle\mathbf{r}_{m}^{2}\right\rangle=a^{2}$

$$
\begin{equation*}
<\mathbf{r}_{n} \cdot \mathbf{r}_{m}>=a^{2}(\cos (\alpha))^{|n-m|} \tag{1.23}
\end{equation*}
$$

Thanks to equation 1.23, we calculated the end-to-end distance $R_{\text {freely }}$ for a chain having a large number of links $N$ :

$$
\begin{array}{lc}
R_{\text {freely }}= & \left(\sum_{n=1}^{N} \sum_{m=1}^{N}<\mathbf{r}_{n} \cdot \mathbf{r}_{m}>\right)^{\frac{1}{2}} \\
R_{\text {freely }}= & \left(\sum_{n=1}^{N} \sum_{k=-n+1}^{N-n}<\mathbf{r}_{n} \cdot \mathbf{r}_{n+k}>\right)^{\frac{1}{2}} \\
R_{\text {freely }}= & \left(\sum_{n=1}^{N} \sum_{k=\infty}^{\infty}<\mathbf{r}_{n} \cdot \mathbf{r}_{n+k}>\right)^{\frac{1}{2}} \\
R_{\text {freely }}= & \left(\sum_{n=1}^{N} a^{2}\left(1+2 \sum_{k=1}^{\infty}(\cos (\alpha))^{k}\right)\right)^{\frac{1}{2}} \\
R_{\text {freely }}= & \left(\sum_{n=1}^{N} a^{2}\left(\frac{1+\cos (\alpha)}{1-\cos (\alpha)}\right)\right)^{\frac{1}{2}} \\
R_{\text {freely }}= & a\left(\frac{1+\cos (\alpha)}{1-\cos (\alpha)}\right) N^{\frac{1}{2}} \tag{1.29}
\end{array}
$$

The end-to-end distance for the freely rotating chain follows $R_{\text {freely }} \sim a N^{\frac{1}{2}}$ like the freely joined chain model.

Remark: For a chain that contains a large number of links $N$, the probability of a chain to have an end-to-end distance $R(N)$ is given by a gaussian probability:

$$
\begin{equation*}
\Omega(R, N) \propto P(R, N)=\left(\frac{3}{2 \pi R_{o}^{2}}\right)^{\frac{3}{2}} \exp \left(-\frac{3 R^{2}}{2 R_{o}^{2}}\right) \tag{1.30}
\end{equation*}
$$

where $R_{o}^{2}$ is the average of the square of end-to-end distance of the chain ( $R_{o} \sim a N^{\frac{1}{2}}$ for the above models.

## Effect of the excluded volume

The previous models do not take into account the excluded volume between segments that
represent non-interpenetration between monomers. In particular, the excluded volume effect aims to mimic the behavior of a polymer chain in a good solvent [4]: the monomer prefers to interact with the solvent instead of the other monomers. Consequently, the interaction free energy $F$ of the chain will be the sum of two contributions:

$$
\begin{equation*}
F=F_{e n t}+F_{e x c l} \tag{1.31}
\end{equation*}
$$

First, the entropic energy term $F_{\text {ent }}$ comes from the energy cost to stretch the polymer chain. Then, the excluded volume interaction term $F_{\text {excl }}$ takes into account the volume of the monomer.

The entropic contribution energy $F_{\text {ent }}$ for the chain having N monomer writes:

$$
\begin{equation*}
F=-T S(\Omega(R, N))=-T k_{B} \ln (\Omega(R, N)) \propto \frac{3}{2} \frac{R^{2}}{R_{o}^{2}} k_{B} T \tag{1.32}
\end{equation*}
$$

In the other hand, the excluded interaction volume represents an energy cost of $k_{B} T$ per monomer. The number of interactions is assumed to be of the order of $N$, the excluded volume is denoted by $v$ and the number density of monomers that occupied the pervaded volume is $\sim \frac{N}{R^{3}}$ so that $F_{\text {excl }}$ is expressed in the following way:

$$
\begin{equation*}
F_{e x c l}=\frac{N^{2} v}{R^{3}} k_{B} T \tag{1.33}
\end{equation*}
$$

If the constant term that does not depend on $R$ is omitted, we obtain the free energy of the polymer $F$ also called Flory energy $F_{\text {Flory }}$ [27]:

$$
\begin{equation*}
F_{F l o r y} \sim \frac{3}{2} \frac{R^{2}}{R_{o}^{2}} k_{B} T+\frac{N^{2} v}{R^{3}} k_{B} T \tag{1.34}
\end{equation*}
$$

The minimization of this energy in function of $R$ yields to the end-to-end equilibrium distance $R_{\text {Flory }}$ of the chain that takes into account excluded volume:

$$
\begin{equation*}
R_{F l o r y} \sim a N^{\frac{3}{5}} \tag{1.35}
\end{equation*}
$$

In conclusion, the end-to-end distance of a neutral polymer chain depends only on the number of monomers $N$ and the quality of the solvent through the parameter $\nu$ :

$$
\begin{equation*}
R \sim a N^{\nu} \tag{1.36}
\end{equation*}
$$

In $\theta$ solvent, the freely joined chain and the freely rotating chain models give $\nu=\frac{1}{2}$. In good solvent, (i.e. by taking account of excluded volume) the polymer occupies a higher volume than in $\theta$ solvent because the coefficent $\nu=\frac{3}{5}$ is higher.

### 1.2.3.2 Conformation of a weakly charged polyelectrolyte in solution

We consider a chain of $N$ monomers that contain a number of elementary charges $N f(0<f<$ 1) spread uniformly along the chain. The chain is assumed to be hydrophilic (or solvaphilic if the solvent is not water) and in solution with no added salt and no counterions.

The electrostatic interaction between charges distributed on the chain is repulsive because the scope of the electrostatic interaction in pure water ( $\lambda_{D}=970 \mathrm{~nm}$ ) is higher than the average distance separating charges on the polyelectrolyte. The chain would adopt a stretched conformation in order to minimize the electrostatic energy but at the expense of an important entropic cost. Not only, the configuration will be extended because of the charge repulsion of the chain but it will display local fluctuations below a characteristic size $\xi_{e}$ [28]. The local fluctuations are not sensitive to the electrostatic charges and the chain conformation at such scale resembles those of a neutral chain (section 1.2.3.1). The chain can be thus divided in "artificial" distincts units called electrostatic blobs [29] as illustrated in Figure 1.12.

A blob is assumed to be spherical with a radius $r_{e}=\xi_{e} / 2$ composed of $N_{e}$ monomers of size $a$ and carries a charge $q_{e}=N_{e} f|e|$ for which the electrostatic interaction compensates exactly thermal agitation.


Figure 1.12 - Partition of the polyelectrolyte of length $L$ into electrostatic blobs of diameter $\sim$ $\xi_{e}$. At large scale, the blobs are aligned because of the electrostatic repulsion. The length of the polyelectrolyte is thus $L=\left(N / N_{e}\right) \xi_{e}$.

The electrostatic interaction $U_{e}$ associated with an increase of the blob radius writes:

$$
\begin{equation*}
d U_{e}=\frac{4 \pi \rho_{e}^{2} r_{e}^{4}}{\epsilon_{o} \epsilon_{s}} d r_{e} \tag{1.37}
\end{equation*}
$$

where $\rho_{e}=\frac{N_{e} f|e|}{\frac{4}{3} \pi r_{e}^{3}}$ is the charge density of the blob. Integration of equation 1.37 results in the electrostatic energy of the blob:

$$
\begin{equation*}
U_{e}=\frac{4 \pi \rho_{e}^{2} r_{e}^{5}}{15 \epsilon_{o} \epsilon_{s}}=\frac{3}{20} \frac{\left(N_{e} f|e|\right)^{2}}{\pi \epsilon_{o} \epsilon_{s} r_{e}} \tag{1.38}
\end{equation*}
$$

By introducing the Bjerrum length, the electrostatic interaction is written as a function of $k_{B} T$ units for a blob of diameter $\xi_{e}$ :

$$
\begin{equation*}
U_{e}=\frac{5}{6}\left(N_{e} f\right)^{2} \frac{l_{B}}{\xi_{e}} k_{B} T \sim\left(N_{e} f\right)^{2} \frac{l_{B}}{\xi_{e}} k_{B} T \tag{1.39}
\end{equation*}
$$

From now on, we ignore the prefactor that may appear in calculations and only focus on dimensional quantity to establish scaling laws [28]. The existence of the blob relies on balance between the electrostatic and the thermal energy which provide a scaling law for the blob diameter:

$$
\begin{equation*}
\xi_{e} \sim\left(N_{e} f\right)^{2} l_{B} \tag{1.40}
\end{equation*}
$$

In the case of a blob where the electrostatic energy is not sufficient to compensate thermal energy, the chain behaves as if it was neutral and the size of the blob satisfies the relation 1.36

$$
\begin{equation*}
\xi_{e} \sim a N^{\nu} \tag{1.41}
\end{equation*}
$$

A combination of equations 1.40 and 1.41 yields to the determination of the number of monomers per blob $N_{e}$ and the size of the blob $\xi_{e}$ :

$$
\begin{align*}
& \xi_{e} \sim a\left(\frac{l_{B}}{a}\right)^{\frac{\nu}{\nu-2}} f^{\frac{2 \nu}{\nu-2}}  \tag{1.42}\\
& N_{e} \sim a\left(\frac{l_{B}}{a}\right)^{\frac{1}{\nu-2}} f^{\frac{2}{\nu-2}} \tag{1.43}
\end{align*}
$$

Inside the electrostatic blob thermic fluctuations dominate so that the chain behaves like a neutral chain in $\theta$ solvent or in good solvent depending of the value of $\nu$. At long range, the blobs repel each other and align themselves so that the length of the chain is:

$$
\begin{equation*}
L \sim \frac{N}{N_{e}} \xi_{e} \sim a N\left(\frac{l_{B}}{a}\right)^{\frac{\nu-1}{\nu-2}} f^{\frac{2 \nu-2}{\nu-2}} \tag{1.44}
\end{equation*}
$$

By using scaling laws, it is possible to characterize the polyelectrolyte with the concept of electrostatic blobs which takes account of local fluctuations at short range. The polyelectrolyte chains adopt the following characteristic lengths depending of the solvent quality.

- In $\theta$ solvent, we have $\nu=\frac{1}{2}$ :

$$
\begin{equation*}
\xi_{e} \sim a\left(\frac{l_{B}}{a}\right)^{-\frac{1}{3}} f^{-\frac{2}{3}} \quad \text { and } \quad L \sim a N\left(\frac{l_{B}}{a}\right)^{\frac{1}{3}} f^{\frac{2}{3}} \tag{1.45}
\end{equation*}
$$

- In good solvent, we have $\nu=\frac{3}{5}$ :

$$
\begin{equation*}
\xi_{e} \sim a\left(\frac{l_{B}}{a}\right)^{-\frac{3}{7}} f^{-\frac{6}{7}} \quad \text { and } \quad L \sim a N\left(\frac{l_{B}}{a}\right)^{\frac{2}{7}} f^{\frac{4}{7}} \tag{1.46}
\end{equation*}
$$

In all cases the length of the chain is proportional to the number of monomers:

$$
\begin{equation*}
L \sim a N \tag{1.47}
\end{equation*}
$$

In both cases, the chain adopts a more stretched configuration if its charge $f$ is increased. The result 1.45 is of first importance and has been derived first by Khun, Künzle and Katchalsky in 1948 [30].

### 1.2.3.3 Model of a highly charged polyelectrolyte: Manning-Oosawa theory

The extrapolation of the model of weakly charged polyelectrolyte to highly charged polyelectrolyte involves to identify an electrostatic blob to a monomer: the chain is completely stretched and can be represented by a charged cylinder of total charge $f N e$. Indeed a significant fraction of the monomers $f N$ will be ionized resulting in solvated counterions. However, for highly charged polyelectrolyte, a large fraction of counterions in solution are condensed on the polyelectrolyte resulting in an effective chemical charge $f_{e f f}$. This condensation is called the Manning-Oosawa phenomenon [31, 32].

In order to capture the strength of the electrostatic interaction between separated charges on a polyelectrolyte, the dimensionless Manning parameter is introduced [31, 33] and is defined as the ratio of the Bjerrum length over the distance $b$ separating two ionic sites on the polyelectrolyte:

$$
\begin{equation*}
\xi=\frac{l_{b}}{b}=\frac{e^{2}}{4 \pi \epsilon_{s} \epsilon_{o} k_{B} T b} \tag{1.48}
\end{equation*}
$$

Remark: for $\xi<1$, the electrostatic interactions acting between two ionic sites are
weaker than thermal fluctuations, the situation being reversed for $\xi>1$.

The Manning-Oosawa theory estimates the fraction of counterions condensed on the polyelectrolyte assimilated to a charged cylinder. The key idea is to divide the system in two cylindrical regions around the linear polyelectrolyte [34]: a region that contains the polyelectrolyte and condensed counterions within a cylinder of radius $r_{c}$ centered along the polyelectrolyte axis and another region that contains free counterions corresponding to a hollow cylinder such that $r_{c}<r<r_{f}$. The two regions are depicted in Figure 1.13 .


Figure 1.13 - Linear polyelectrolyte with the region of condensed counterions (full black beads) and region of free counterions (circle black beads).

If $\beta$ refers to the number of ions that are free, the effective Manning parameter is:

$$
\begin{equation*}
\xi_{e f f}=\beta \xi \tag{1.49}
\end{equation*}
$$

In other words, when more counterions are condensed on the polyelectrolyte, the value of the Manning parameter decays as counterions are screening electrostatic repulsions between polyelectrolyte ionic sites.

It is argued that the concentration of condensed and free counterions obeys a Poisson-Boltzmann
distribution. Hence, the concentration of free counterions and condensed counterions verifies the following relationship:

$$
\begin{equation*}
n_{c}=n_{f} \exp \left(\frac{-e \Delta \phi}{k_{B} T}\right) \tag{1.50}
\end{equation*}
$$

Let us derive the electrostatic potential induced by the polyelectrolyte cylinder. We recall that the polyelectrolyte and ions are in a solvent of relative permittivity $\epsilon_{s}$. By applying the Gauss theorem, we calculate the electric field $E(\mathbf{r})$ produced by the polyelectrolyte:

$$
\begin{equation*}
E(\mathbf{r})=\frac{e}{2 \pi \epsilon_{o} \epsilon_{s} b r} \mathbf{r} \tag{1.51}
\end{equation*}
$$

By applying the gradient to the electric field, we derive the expression of the electric potential in condition that the potential vanishes at $r_{f}$ :

$$
\begin{equation*}
\phi(r)=-\frac{e}{2 \pi \epsilon_{o} \epsilon_{s} b} \ln \left(\frac{r}{r_{f}}\right) \tag{1.52}
\end{equation*}
$$

The electric field can be expressed in function of the Manning parameter $\xi$ :

$$
\begin{equation*}
\phi(r)=-2 \xi\left(\frac{k_{B} T}{e}\right) \ln \left(\frac{r}{r_{f}}\right) \tag{1.53}
\end{equation*}
$$

The difference of electric field between $\mathrm{r}_{c}$ and $\mathrm{r}_{f}$ can be expressed in function of the effective Manning parameter $\xi_{\text {eff }}$ and subsequently in function of the volume:

$$
\begin{equation*}
\Delta \phi=-2 \xi_{e f f}\left(\frac{k_{B} T}{e}\right) \ln \left(\frac{r_{f}}{r_{c}}\right)=-\xi_{e f f}\left(\frac{k_{B} T}{e}\right) \ln \left(\frac{V_{f}+V_{c}}{V_{c}}\right) \tag{1.54}
\end{equation*}
$$

In order to remove the difference in electric potential $\Delta \phi$, we combine equations 1.50 and 1.54

$$
\begin{equation*}
\ln \left(\frac{n_{c}}{n_{f}}\right)=-\frac{e \Delta \phi}{k_{B} T}=\xi_{e f f} \ln \left(\frac{V_{f}+V_{c}}{V_{c}}\right)=\xi \beta \ln \left(\frac{V_{f}+V_{c}}{V_{c}}\right) \tag{1.55}
\end{equation*}
$$

In another way, the Manning parameter can be expressed in function of $n_{c}, V_{c}, n_{f}$ and $V_{f}$ :

$$
\begin{equation*}
\ln \left(\frac{1-\beta}{\beta}\right)=\ln \left(\frac{n_{c}}{n_{f}}\right)+\ln \left(\frac{V_{c}}{V_{f}}\right) \tag{1.56}
\end{equation*}
$$

Let us introduce the fraction of space that contains the condensed counterions as $\alpha=\frac{V_{c}}{V_{f}+V_{c}}$ so that equation 1.55 and 1.56 can be combined in the following way:

$$
\begin{equation*}
\ln \left(\frac{1-\beta}{\beta}\right)=\ln \left(\frac{\alpha}{1-\alpha}\right)-\beta \xi \ln \alpha \tag{1.57}
\end{equation*}
$$

Even though this equation seems complicated, we can extract several important information. Let us analyze the case $\phi \rightarrow 0$ such that the volume of the condensed region becomes small.

- Oosawa demonstrated that equation 1.57 admits two types of solutions. When $\xi<1$, a solution exits for $\beta=1$. Condensation of counterions does not occur when thermal fluctuations dominate electrostatic interactions and thus counterions are free. However, when $\xi>1$, a solution exists for $\beta=\frac{1}{\xi}$, sufficient amount of counterions condensate in order to maintain an effective charge density $\xi_{\text {eff }}$ inferior or close to the critical charge density $\xi_{c}=1$ corresponding to the onset of counterion condensation.
- The above conclusion holds for higher valency Z of the counterions but the onset of counterion condensation is reduced to $\xi_{c}=\frac{1}{|Z|}$ while for $\xi>\xi_{c}$, the fraction of condensed charge becomes $\beta=\frac{1}{|Z| \xi}$. We can summarize the evolution of the Manning parameter $\xi$ in function of the effective Manning parameter $\xi_{\text {eff }}$ on the plot represented in Figure 1.14 .
- If we consider the DNA molecule in presence of its natural monovalent counterions $\mathrm{Na}^{+}$, the Manning parameter is $\xi=4.2$ and condensation occurs for $\xi_{\text {eff }}=\xi \beta \sim 1$ and the fraction of the condensed monovalent ions on the DNA is $\frac{1-\beta}{\beta}=\frac{3.2}{4.2} \sim 0.75$.


Figure 1.14 - Plot that represents the effective Manning parameter $\xi_{\text {eff }}$ in function of the Manning parameter of the chain $\xi$ in the Manning-Oosawa condensation theory.

In summary, a highly charged polyelectrolyte can be assimilated to a weakly charged polyelectrolyte due to counterion condensation with an effective Manning parameter predicted to be $\xi_{e} f f=\xi$ if $\xi<\frac{1}{|Z|}$ and $\xi_{e} f f=\frac{1}{|Z|}$ if $\xi>\frac{1}{|Z|}$.

### 1.2.4 Conclusion

We provided some of the important physical aspects of polyelectrolytes in solution. First, we introduced the different classes of polyelectrolytes represented in particular by the weak/strong and hydrophilic/hydrophobic polyelectrolytes for which the solvent plays a major role.

Then, we introduced some important elements of electrostatics to describe charges in solution. The presence of free ions modifies strongly the electrostatic interactions in solution. The PoissonBoltzmann (PB) equation is non-linear and is widely used for analytical or numerical simulations of systems involving charged species in solution. When the thermal fluctuations dominate, the PB equation is linearized and leads to the Debye-Hückel equation for which the scale of the electrostatic interactions is given by the Debye length $\lambda_{D}$.

We presented physical concepts to describe hydrophilic polyelectrolytes in solution depending on their intrinsic charges. As a starting point, we introduced models that describe conformation of uncharged chains through the calculation of the end-of-end distance $R \sim a N^{\nu}$ where $\nu$ depends on the quality of the solvent. For a weakly charged polyelectrolyte, the chain adopts a linear conformation due to electrostatic repulsion at large scale and the typical length is found to be
proportional to the number of monomers $(R \sim a N)$. We recalled the concept of electrostatic blob that highlights local thermal fluctuations. For a highly charged polyelectrolyte, it is assumed that the chain has a stretched conformation and strong electrostatic interactions with dissociated counterions. Along this line, the Manning-Oosawa model has been presented. Above a critical charge of the polyelectrolyte given by the critical Manning parameter $\xi_{\text {crit }}=\frac{1}{|Z|}$, most of the counterions are condensed on the polyelectrolyte which can be assimilated to a weakly charged polyelectrolyte with an effective Manning parameter $\xi_{\text {eff }}=\frac{1}{|Z|}$.

### 1.3 Complexes of polyelectrolytes

Polyelectrolyte complexes (PECs) are complexes formed by interactions between oppositely charged polyelectrolytes. PECs drive attention of researchers since the end of the $19^{\text {th }}$ century with the work of Albrecht Kossel in 1896 who pointed out that electrostatic interactions drive precipitation in a system of oppositively charged proteins and carbohydrates [35]. We review some important aspects of polyelectrolyte complexes in the subsequent section. First, we focus on the interactions that drive complexation of polyelectrolytes. Then, we highlight mechanisms of complex formation and we provide in a last part, some examples of structures that can emerge from polyelectrolyte complexation.

### 1.3.1 Interactions driving polyelectrolyte complexation

Depending of the nature of the polyelectrolytes to be involved in the complex formation, complexation can be driven by different interactions. We review the most important interactions which are hydrogen bonds, electrostatic interactions and hydrophobic interactions [2].

- PECs can arise through hydrogen bonds between neutral hydrophilic polyelectrolytes that bear proton donating units and proton accepting units [36. Such complexes can form with neutral polymers that contain amine, alcohol, acid or ether groups [37]. In case of polyelectrolytes, hydrogen bonds still exist in presence of polyethylene glycol or polysaccharide backbone in addition to electrostatic interactions.
- It is admitted that the main interaction that governs complexation of polyelectrolytes are electrostatic interactions [1]. The parameters that influence complexation of polyelectrolytes
are in fact the ones modifying electrostatic interactions such as pH , ionic strength or charge carried by the polyelectrolyte.
- Although electrostatic interactions are the main driving force for polyelectrolyte complexation, it has been proven that hydrophobic interactions are also involved. For example, it is possible to determine hydrophobicity of polyelectrolytes by chromatography. In particular surface hydrophobicity is directly estimated and plays a role in polyelectrolyte complexation (38).


### 1.3.2 Mechanism of complex formation

The PECs are mostly driven by electrostatic interactions between oppositively charged polyelectrolytes. In solution, the polyelectrolytes are also associated with their respective low molecular weigth counterions that are released during the complex formation, resulting in an entropy gain of the system [39]. First, the role of counterions in complex formation is highlighted and we present subsequently the general kinetic of complex formation.

### 1.3.2.1 Thermodynamics for complex formation



Figure 1.15 - Mixture of positive and negative polyelectrolytes can result in complex formation and release of counterions. Reprinted from ref. [40] (Journal of colloid and interface science, 2011, $361(2)$, 407-422, Figure 1, Copyright 2011 with permission from Elsevier).

Let us consider a highly charged polyelectrolyte in solution with associated counterions. It is predicted from Manning-Oosawa theory (section 1.2.3.3) that such polyelectrolyte had a significant fraction of counterions that are condensed and form a charged layer. Not only, it is likely that
the fraction of condensed counterions lower the interaction energy of the system compared to the hypothetical situation of all free counterions.

When two opposite charged polyelectrolytes are mixed together to form a complex, the layers of counterions are destroyed and the counterions are realeased to form a like salt solution (Figure 1.15.

The complex formation induces a change in the energy (enthalpy $H$ ) and entropy $S$ of the system and each contribution depends on ionic strength if salt ions are present in solution in addition to the counterions of oppositively charged polyelectrolytes. The Figure 1.16 exhibits the variation of Gibbs free energy $\Delta G=\Delta H-T \Delta S$ as function of the salt concentration. In any cases, the variation of entropy $\Delta S$ is likely to be positive because of the release of counterions initially condensed on polyelectrolytes. The contribution would be larger at low ionic strength and decreases as ionic strength increases because in that case, the release of counterions represents a smaller fraction of free ions in solution that already contribute to the translational entropy. On the other hand, a negative (but smaller) contribution to the entropy can result from a loss of configurational and translational entropy of complexed polyelectrolytes, provided that the chains are long enough [41, 40]. However, complexation is usually favored entropically.


Figure 1.16 - Effect of ionic strength (salt concentration) on the variation of free energy $\Delta G$ of the system, the variation of entropy $\Delta S$ as well as the variation of enthalpy $\Delta H$. Depending on the salt concentration, the complexation can be exothermic or endothermic. Reprinted from ref. [40] (Journal of colloid and interface science, 2011, 361(2), 407-422, Figure 2, Copyright 2011 with permission from Elsevier).

The enthalpy of the system can be either positive or negative. At low ionic strength, the energy of the system decreases considerably upon polyelectrolyte complexion and the process is exother-
mic. At higher ionic strength, the decrease of energy associated to complexation of polyelectrolytes is compensated by an increase of energy because of release of counterions, so that the complexation is endothermic. Such trend has been confirmed by calorimetric measurements 42] and computer simulations 43. The crossover between endothermic and exothermic complexation depends of the nature of the polyelectrolytes. Nonetheless, a general trend is that for highly charged polyelectrolytes, the complexation is endothermic because a large fraction of counterions are adsorbed before complexation. On the contrary, complexation is thus likely to be exothermic for weakly charged polyelectrolytes [40].

We remind that complexation of polyelectrolytes not only involves electrostatic interactions but also depends on various other interactions due to the nature of polyelectrolytes (hydrogen bonds), or the solvent (hydrophobicity). A clear picture of each contribution in the complexation of polyelectrolytes has not been obtained so far.

### 1.3.2.2 Dynamic of complex formation

From a general perspective, the dynamic of complex formation can be summarized into three stages [2] that are shown in Figure 1.17

1. The mixing of polyelectrolytes results in the primary complex formation. This process is fast and takes place in several ms [2].
2. An intermediate complex emerges $1-2 \mathrm{~h}$ after mixing time, due to new bond formation and alteration of the polyelectrolyte chains.
3. Intercomplex aggregation result from the gathering of the intermediate complexes. This stage is influenced by various factors such as the conditions of complexation or the nature of polyelectrolytes [2].

The above mechanism can be influenced by many factors, we will review in detail most of them in the subsequent section.

### 1.3.3 Possible structures of polyelectrolytes

The intrinsic nature of polyelectrolytes such as the strength, the molecular weight [44, the charge distribution [45] and composition of the mixture influence the structure that can emerge


Figure 1.17 - Schematic representation of the different stages of PECs formation. Reprinted from ref. [2] (Artificial cells, nanomedicine, and biotechnology, 2011, 44(7), 1615-1625, Figure 1, Copyright 2011 with permission from Taylor \& Francis).
from complexation process. In addition, their conditions of preparation such as the solvent quality, ionic strength [46, 47], pH [48] as well as the mixing procedure (e.g. order of mixing [49, [50], time of mixing [51]) could result in various PEC morphologies.

PECs are classified into three categories of structures which are water-soluble, colloidal and dense PECs [2, 1].

- Studies on water-soluble PECs took their origin in the work of Kabanov [52, 53] and Tsuschida [54, 55]. PECs arise between polyelectrolytes with weak ionic groups and large difference in molar mass that are mixed in a non-stoichiometric ratio. The complex consists of long host polyelectrolytes on which are adsorbed short guest polyions of opposite charge in a ladder-like structure [52]. Polyelectrolyte complex highlights hydrophilic single-stranded and hydrophobic double-stranded units (Figure 1.18).
- PECs formation between strong polyelectrolytes results in macroscopic flocculated systems. However, in dilute solution, the complexation can be stopped at the colloidal level [56, 57. Although a $1: 1$ stoichiometry is found for strong polyelectrolytes, the general trend is that


Figure 1.18 - General aspect of the ladder-like structure adopted by water-soluble PECs.
the inner part of the particle of colloidal PECs constitutes a homogeneous and charge neutral entangled core where 1:1 stoichiometry prevails while the outer shell consists of polyelectrolyte layers that give the complex its charge. This outer shell stabilizes the particle from further aggregation [57].

- PECs can also form dense structures. PECs can separate from the solvent in a liquid-solid phase (precipitation) to form a solid phase called solute. In another case, the complexes remain soluble but two distinct phases emerge with one rich in PECs called coacervate. The precipitation occurs when the charge density of polyelectrolytes is high and ionic strength is weak such that the electrostatic interactions are maximum [58]. It is likely that electrostatic interactions play the major role in formation of dense structures given that an increase in ionic strength screens interactions between polyelectrolytes and reduces precipitation [36].

Depending on the stiffness, size and architecture of the polyelectrolytes, various complexes can be formed and some of them are presented in Figure 1.19

The structure and size of the PECs are investigated with various techniques. Dynamic light scattering is one of the most used techniques to estimate the size of PEC particles. The diameter can also be estimated in solution by using Stokes-Einstein equation [60]. The detailed structure of PECs is probed with small-angle X-ray diffraction and neutron scattering and has been used to investigate rod-like 61, 62], core-shell 63] or randomly branched PECs 64].


Figure 1.19 - Examples of structures of PECs with polyelectrolytes of different form and stiffness. A) Flexible linear + globular. B) Semiflexible linear + globular. C) Rodlike linear + globular. D) Flexible linear + flexible linear. E) Rodlike linear + flexible linear. F) Globular + globular. Reprinted from ref. [59 (Current opinion in colloid \& interface science, 2006, 11(5), 295-301, Figure 1, Copyright 2006 with permission from Elsevier).

### 1.3.4 Conclusion

Polyelectrolyte complexes represent an important class of materials which own their existence mainly from electrostatic interactions. The diversity of polyelectrolyte structures and the possible conditions of preparations result in a wide range of possible PECs. Consequently, PECs attract interest from the scientific community because of the various possible applications especially in pharmaceutics and medicine such as drug carrier systems for enzymes, DNA or drug where charged particles can be integrated to the complex [2]. Because of the diversity and the high number of existing and/or prospective PECs such that each system would represent a full study in itself, we decided to focus only on two systems of PECs that will be presented and discussed in the next chapter.

## CHAPTER 2

## Studied polyelectrolyte systems

## Contents

2.1 Project 1: Polyelectrolyte complexation between DNA and nanopar-
ticles . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
2.1.1 Fundamental aspects of DNA compaction ..... 43
2.1.2 DNA compaction with functionalized gold nanoparticles ..... 51
2.2 Project 2: Polyelectrolyte complexation between PEDOT:PSS and ionic liquids ..... 61
2.2.1 Review about conducting polymers ..... 61
2.2.2 The PEDOT semi-conducting polymer ..... 67
2.2.3 PEDOT:PSS conductivity enhancement by ionic liquids ..... 72
2.3 Objectives of the thesis ..... 80

The miniaturization of the transistor had a broad impact on the technology in the second part of the twentieth century and motivated a lot of researches in material science. By pursuing miniaturization until the nanoscale dimension, the standard "top-down" processing techniques that consist in using external tools to reduce the size of large structures into desired dimensions and patterns is increasingly difficult at the molecular size. In order to tackle this challenge, the "bottomup" approach uses the spontaneous self-assembly of small molecules by monitoring interactions at the molecular level in order to elaborate the material like a puzzle 65, 66]. Compared with topdown methods, such techniques represent the advantage of large chemical diversity as well as highly parallel synthesis [67]. An aqueous media made easier reorganization and reorientation of molecules through electrostatic interactions that leads to formation of self-assembled nanostructures called
polyelectrolyte complexes (PECs). Spontaneous assembly is widely spread in nature and a clear example is the condensation of DNA in eukaryotes cells and in certain viruses in presence of different small molecules or ions.

Despite better understanding of basic underlying mechanism, prediction and or/interpretation of various polyelectrolyte complexation leading to self-assembly are still unclear and problematic because of the different interactions and conditions of preparation that govern polyelectrolyte complex formation. However, the precise understanding of ordering at the nanoscale level is a prerequisite to the design of new materials with desired properties. The goal of this chapter is to present the polyelectrolyte systems for which characteristics and self-assembly process would be studied subsequently through numerical simulations.

The resulted aggregates can be either in thermodynamic equilibrium or in a metastable state 440. We will focus in this work on two systems for which interesting experimental results for prospective applications in various domains have been obtained.

- A first polyelectrolyte system involves double-strand DNA molecules negatively charged and positively charged gold nanoparticles (AuNPs). More precisely, the nanoparticles have a typical diameter comparable to that of the DNA ( $\sim 2 \mathrm{~nm}$ ) and are positively charged because of the functionalized ligands at the gold core surface. The self-assembly process of DNA and AuNPs yields to the stabilization of complexes made of a single DNA chain decorated by AuNPs, or complexes constituted by several DNA molecules and AuNPs [7] [68, 69].
- The second polyelectrolyte system is a solution of water-processable conducting polymers of Poly-3,4-ethylenedioxythiophene:polystyrenesulfonate (PEDOT:PSS) where PEDOT is positively charged and PSS negatively charged. It has been proven experimentally that PEDOT:PSS morphology can be controled by addition of ionic liquids, which are molten salts at room temperature [II].


### 2.1 Project 1: Polyelectrolyte complexation between DNA and nanoparticles

### 2.1.1 Fundamental aspects of DNA compaction

DNA compaction is a phase transition undergone by a DNA molecule from an elongated conformation to a very compact form. DNA compaction is ubiquitous in nature because it is required to package DNA into tiny space in nuclei of cells or in virus capsids. Also, in vitro experiments have been performed to understand factors that influence DNA compaction. It is a crucial step in order to use negatively charged DNA as a scaffold to further control assembly with various positively charged objects.

Remark: The compaction of a long DNA chain induces attraction between DNA base pairs which is locally similar to attraction of DNA base pairs between different DNA molecules in case of self-assembly.

### 2.1.1.1 The coil-globule transition

DNA is a negatively charged polyelectrolyte that adopts an elongated coil conformation in solution because of the strong electrostatic repulsion between the phosphate groups. Upon addition of compaction agent, a long DNA molecule (e.g. significantly longer than the persistence length of DNA of $l_{p}=50 \mathrm{~nm}$ ) undergoes a phase transition that results in its compaction 70, 71]. In vitro experiments performed in the group of Yoshikawa highlighted the different compaction pathway that can undergone a single DNA molecule [72, 73].

Three modes of compaction (figure 2.1) are usually encountered for DNA to change from an elongated state to a compact state and are summarized in the review of Estévez-Torres and Baigl [74.

- The first mode is an all-or-none compaction process where there is no intermediate steps: there is coexistence between elongated DNA and compacted DNA. This process is observed by addition of small multivalent counterions or acting on solution properties. For instance, a solution with a poor solvent such as ethanol that interacts unfavorably with DNA base pairs or addition of neutral polymers can favor DNA compaction [72, 73].


Figure 2.1 - Presentation of the three different routes for DNA compaction. Reprinted from ref. 744 (Soft Matter, 2011, 7(15), 6746-6756, Figure 1, Copyright 2011 with permission from the Royal Society of Chemistry).

- The second mode of compaction is a progressive transition from elongated coil to compacted state. This compaction happens when several consecutive DNA base pairs are attracted and form local complexation with positively charged compaction agents extending over more than 10 base pairs. The second compaction mode has been proven in presence of poly-L-lysine of different lengths [75].
- The last mode of compaction is an adsorption followed by wrapping of DNA around objects. We already presented this mode of compaction in section 1.1.2.3 in case of DNA wrapping around histones to form chromosome.


### 2.1.1.2 Compaction agents

Compaction agents are molecules that favor DNA compaction by inducing attractive interactions between the DNA base pairs or minimizing interaction of the DNA base pairs with the solvent. There exists various compaction agents for which effects on DNA compaction has been investigated in in vitro experiments.

- Small multivalent counterions with a valency $\mathrm{Z} \geq 3$ are known to induce DNA compaction. These can be natural polyamines like spermidine [76] $(\mathrm{q}=+3|\mathrm{e}|$ at $\mathrm{pH}=7)$ or spermine [77] $(\mathrm{q}=+4|\mathrm{e}|$ at $\mathrm{pH}=7)$ and inorganic cations like $\mathrm{Co}\left(\mathrm{NH}_{3}\right)_{6}^{3+}$ which are the
mostly used counterions. It has been also reported that trivalent metallic cations such as $\mathrm{Al}^{3+}, \mathrm{Ga}^{3+}$ [78], $\mathrm{Cr}^{3+}$ [79] or lanthanide ions [80] $\left(\mathrm{La}^{3+}, \mathrm{Eu}^{3+}, \mathrm{Tb}^{3+}\right)$ can induce DNA compaction. The mode of compaction that occurs in presence of such ions is the all-or-none compaction in most cases [74] and multivalent counterions are often in competition with monovalent salt present in solution. It is likely that an excess of counterions is required to compensate the effect of salt and induces DNA compaction and the transition concentration $\rho^{*}$ defined as the total charge of the compaction agent over charge carried by the DNA is greater than one (Figure 2.2) which means that for small Z, a higher number of compaction agent must be present in solution to induce DNA compaction [74].


Figure 2.2 - Scheme that presents the different DNA transition modes (all-or-none or progressive compaction) for the transition concentration $\rho^{*}$ necessary to induce DNA compaction, depending of the valency Z of the compaction agent. Reprinted from ref. [74] (Soft Matter, 2011, 7(15), 6746-6756, Figure 2, Copyright 2011 with permission from the Royal Society of Chemistry).

- Linear polycations are long polycations that carry a charge $Z>10$. Examples of such polycations are protamines [81] (compaction agent in sperm cell) or cationic polypeptides such as polylysine [82]. The compaction of DNA in presence of linear polycations is progressive and achieved at $\rho^{*} \sim 1$ (Figure 2.2 because each polycation can induce locally a DNA collapse [83]. Hence, the compaction of DNA is highly dependent of the valency Z of the compaction agent and for large value of Z the compaction occurs at $\rho^{*} \sim 1$.
- It is possible to compact DNA with highly charged, bulky, tridimensional polycationic
nanostructures such as cationic dendrimers [84, supramolecular assemblies 85] or nanoparticles [86, 87]. The case of compaction of DNA with nanoparticles will be discussed below. The size of the cationic nanostructures as well as the repartition of charge influence the compaction mode of DNA [74].
- Amphiphilic cationic species (surfactants) can also induce DNA compaction. Indeed, cationic surfactants that contain hydrophobic part adsorb on DNA because of electrostatic interactions but tend also to self-assemble in aqueous solution because of hydrophobicity which can result in cooperative effects that promote DNA compaction [74].
- In the previous examples, DNA compaction was performed by using cationic species. Another way to induce compaction consists of using neutral or anionic polymers such as respectively polyethyleneglycol [88] (PEG) or polyaspartate, polyglutamate or anionic polypeptides [89]. A large amount of polymers in solution would exert an osmotic pressure that results in DNA compaction [71], generally in an all-or-none mode for a single DNA molecule [74].


### 2.1.1.3 Condensation of cationic agents on DNA

We previously presented the different compaction agents that in most cases (we do not consider here neutral or anionic polymers) interact electrostatically with the DNA negative phosphate groups. In order that DNA compaction happens, it is required that a significant fraction of DNA charges to be neutralized by the compaction agent so that base pair - base pair interaction happens. It has been proven theoretically that DNA compaction occurs when the fraction of neutralized phosphates is $r \sim 0.89$ by modifying the Manning-Oosawa theory [77, 90]. Indeed, competition between two types of counterions of different valencies to adsorb on DNA is taken into account in the modified Manning-Oosawa theory and the fraction of neutralized phosphate groups $r$ is constant with $\mathrm{Na}^{+}$or $\mathrm{Mg}^{2+}$ as the lower valent cation and spermidine ${ }^{3+}$ or spermine ${ }^{4+}$ as the higher valent cation. Notice that the charge neutralization $r \sim 0.89$ has been verified experimentally by gel electrophoresis measurements of condensation of cations on DNA [91]. Hence, it is possible to apply the simple Manning-Oosawa theory to estimate the fraction of neutralized phosphates by counterions based on their valency Z. However, we remark that the spatial extension of the compaction agents is not taken into account so that the Manning-Oosawa theory would mostly apply to the case of small multivalent counterions [74].

We recall from section 1.2 .3 .3 that the fraction of condensed charge was defined as $\beta=\frac{1}{|Z| \xi}$ where $\xi=\frac{l_{B}}{b}$ was the Manning parameter. The fraction of neutralized DNA phospates base pair is defined as:

$$
\begin{equation*}
r=1-\beta=1-\frac{b}{|Z| l_{B}} \tag{2.1}
\end{equation*}
$$

Given that the value of $l_{B} \sim 7 \AA$ in pure water at $293 K$, the fraction of neutralized phosphate groups is $r=0.76,0.88,0.92,0.94$ respectively for counterions of valency $\mathrm{Z}=1,2,3$ and 4 . The Manning-Oosawa theory thus predicts that condensation happens if $\mathrm{Z} \geq 3$ and confirms that despite huge simplifications, the model explains well the counterion condensation effect on the DNA molecule.

### 2.1.1.4 Mechanism of DNA compaction

As soon as DNA is neutralized by cationic agents, different short range interactions play important roles in order to compact DNA. We distinguish two kind of short range effects that participate in DNA compaction. First, there are correlations between compacting agents adsorbed on DNA. Then, the spatial extension of the compacting agent may result in ion bridging between DNA phosphate groups.

## Correlations of cationic species on DNA

It has been admitted since the pioneering work of Kirkwood and Shumaker [92] that correlations between cationic species adsorbed on DNA can lead to base pair - base pair attraction. The Coulomb repulsion between condensed cationic species on DNA results in an alternation of positive and negative charges at the DNA surface [71]. Consequently, the attraction between like-charge surface induce short range attraction force [93, 94] as the complementary patterns adjust perfectly (Figure 2.3).

The correlation attraction is not predicted by the standard Poisson-Boltzmann theory and various approaches aim to predict existence of attractive interactions between like-charged objects such as integral equation theories 95, modified Poisson-Boltzmann theory 96 or density functional theory 97 .


Figure 2.3 - A) Correlations between ligands adsorbed on DNA induce short range attraction between two likely charged chains. B) The helical nature of DNA is taken into account explicitly but the principle is the same as in A). Reprinted from ref. [71] (Progress in biophysics and molecular biology, 2011, 105(3), 208-222, Figure 2, Copyright 2011 with permission from Elsevier).

## Compacting agents bridging DNA strands

The Manning-Oosawa theory is valid for point charge ions but most of the compaction agents are spatially extended. In particular, it has been proven that divalent rod-like counterions with two individual point-charges separated by a fixed distance can induce DNA compaction [98, 99].


Figure 2.4 - Bridging of divalent positive rod-like counterions to negatively charged surface that can be DNA surface. The configuration of counterions is either parallel or perpendicular to the strand, the latter case inducing the bridging effect. Reprinted from ref. 71] (Progress in biophysics and molecular biology, 2011, 105(3), 208-222, Figure 3, Copyright 2011 with permission from Elsevier).

In presence of parallel DNA molecules, small rodlike particle adopt a perpendicular configuration that favors base pair - base pair interaction (Figure 2.4 and thus DNA compaction. The bridging mechanism is relevant when the compacting agents are spatially extended and has been demonstrated not only for divalent ion but also for ions of higher valency such as polyamines [100].

### 2.1.1.5 DNA overcharging effect

It has been predicted theoretically that DNA charge can be reversed by counterion condensation [101] and experiments showed that polyamines may induce DNA charge reversal by measure of the mobility $(\mu)$ of DNA molecule in presence of various concentration of compacting agent [102, 103]. The change of the sign of $(\mu)$ indicates a change of sign of the DNA charge (Figure 2.5).


Figure 2.5 - Mobility of the compacted DNA ( $8 \mathrm{kbp}, 50 \mathrm{ng} / \mu \mathrm{l}$ ) as function of spermine concentration in buffer containing 1 mMol TRIS (red circles); 10 mMol TRIS (black sqaures) and 10 mMol TRIS and 50 mM KCl (blue triangles). Reprinted from ref. 102 (Nature Physics, 2007, 3(9), 641-644, Figure 1, Copyright 2007 with permission from Nature Publishing Group).

### 2.1.1.6 Structures of compacted DNA : shape and stability

Early TEM studies revealed various shapes of DNA compaction [104, 105] such as spherical globules, bundles, rods and toroids which is the most common shape observed (Figure 2.6).

The first in vitro observation of a DNA compacted into a toroid was reported in 1976 in the work of Gosule and Schellman using spermidine as compacting agent [76] and confirmed in detail decades later in the work of Hud and coworkers [106]. It has been reported that the toroids has a diameter about $\sim 100 \mathrm{~nm}$ with a hexagonal packing with an inter-chain distance of 2.6 nm .

From a physical approach, the compaction of DNA into well-ordered structures is not intuitive and results from the contribution of various factors that have been summarized by Blommfield's review [107]. On a first hand, there are unfavorable physical interactions that contribute to the free energy of DNA compaction: 1) Bending the DNA represents $+1 / 300 k_{B} T$ per base pair due to its intrinsic rigidity; 2) The demixing of DNA with solvent accounts for $+1 / 150 k_{B} T$ per base pair


Figure 2.6 - Example of shapes that could be adopted after DNA compaction (TES studies) 1) Toroid formed by DNA of T7 bacteriophage virus in presence of spermidine. Reprinted from ref. [76] (Nature, 1976, 259(5541), 333-335, Figure 3, Copyright 1976 with permission from Nature Publishing Group). 2) Toroid and spheroid of P4 DNA in presence of spermidine. 3) Rods and toroids of P4 DNA in presence of spermidine. Pictures 2) and 3) reprinted from ref. 105 (Journal of molecular biology, 1978, 121(3), 327-337, Figure 2-3, Copyright 1978 with permission from Elsevier).
[108; 3) the electrostatic repulsion between DNA phosphates using the Manning-Oosawa theory is about $+0.24 k_{B} T$ per base pair. On the other hand, the attractive correlations that induced DNA compaction is derived from modified Poisson-Boltzmann theory and is found to be $-0.3 k_{B} T$ per base pair [109]. These different repulsive and attractive contributions result in the negative free energy of $\sim-0.05 k_{B} T$ per base pair, which is compatible with a stable compacted state.

The DNA compacted structures have a limited size. For instance, the typical diameter of toroids was previously mentioned to be $\sim 100 \mathrm{~nm}$ and its limited size is explained by two contributions [110] which are summarized in the review of Estévez-Torres and Baigl [74]. The first reason is due to the winding of the DNA molecule into toroid that induces topological defects contributing to repulsion between DNA monomers. The second contribution comes from the energy barrier that needs to be overcome in order to bring two rod-like polymers in a parallel orientation.

### 2.1.1.7 Factors that modified DNA compaction/decompaction

We previously presented the compaction of DNA by various compaction agents as well as the compaction mechanism. We review some physico-chemical parameters that influence DNA compaction or decompaction.

## High dielectric solvent $\epsilon_{s}$ promote DNA decompaction

We introduced in section 1.2 .1 .2 the dielectric constant associated to a solvent $\epsilon_{s}$. Instead of describing exactly a solvent with molecular details, the effect of the solvent on electrostatic
interactions in solution are introduced through a factor $\epsilon_{s}$ which is the solvent dielectric constant. It is thus likely that the electrostatic interaction that contributes to DNA compaction is influenced by $\epsilon_{s}$ [11]. For instance, DNA compaction is favored when $\epsilon_{s}$ decreses using water-alcohol mixtures [77) 112].

## Effect of ionic strength on compaction/decompaction

DNA in presence of multivalent cations like polyamines can be decompacted by increasing the amount of monovalent or divalent cation [77]. In that case, the ions of lower valency replace polyamines adsorbed on DNA and from the Manning-Oosawa theory, it is predicted that compaction do not happen with monovalent and divalent ions. On the other hand, in presence of excess of neutral polymers such as PEG, the insertion of monovalent or divalent cations ( $\mathrm{Na}^{+}, \mathrm{Mg}^{2+}$ ) reduce electrostatic repulsion among DNA base pairs and DNA compaction is favored [88, 113].

### 2.1.2 DNA compaction with functionalized gold nanoparticles

We presented above compaction of DNA with various compacting agents. In this section, we focuse on the compaction of DNA with functionalized gold nanoparticles (AuNPs). First, we will present AuNP synthesis and their applications as building block on DNA scaffolds resulting in structured DNA-AuNP complexes.

### 2.1.2.1 Synthesis of functionalized gold nanoparticles

Functionalized gold nanoparticles are of high interest in the field of nanotechnology because of their potential applications in catalysis, biology, optics, nanoelectronics or medecine [114, (8). AuNPs can be functionalized with many agents such as polymers, surfactants, dendrimers or drugs which represents as many possible routes towards new applications. Many protocols have been established for the preparation of AuNPs with various sizes, shapes or ligand shell compositions. Because of the various control parameters that influence AuNP synthesis, the preparation of distinct types of gold nanoparticles remains challenging and a huge effort of research is dedicated to this topic [115, 116, 117, 118]. Unfortunately, the procedure yielding ligand-stabilized AuNPs always results in some dispersity of their size. We review some common techniques to synthesize ligandfunctionalized AuNPs. The principle of AuNP synthesis is based on the reduction of metal salts
through a reducing agent in presence of ligand molecules in order to self-assemble layers of ligands on the gold core surface to stabilize the nanoparticle [8]. The ligands are generally thiol chains with a terminal group such as amines, phosphines or thiolates [8]. The most famous synthesis protocol for functionalizing AuNPs has been proposed in 1951 by Turkevich [119] and is presented in Figure 2.7. In this protocol, sodium citrate is used both as stabilizer and reducing agent.


Figure 2.7 - The Turkevich protocol. Reprinted from ref. 120 (Comprehensive Analytical Chemistry, 2014, 66, 37-79, Figure 1, Copyright 2014 with permission from Elsevier).

More specifically, a solution of chloroauric acid is boiled and trisodium citrate dihydrate is added to the solution under vigorous stirring during few minutes. The resulted ligand-stabilized nanoparticles have a size of $\sim 20 \mathrm{~nm}$ [119]. Few decades later, Frens [121] proposed an improvement of the Turkevich method to synthesize AuNPs with broader size distribution (from 15 to 150 nm ) by tuning the ratio of trisodium citrate to chloroauric acid.

In 1994, Brust and co-workers [122] proposed another synthesis protocol for the elaboration of thiol-protected AuNPs (Figure 2.8. The method consists of transferring $\mathrm{AuCl}_{4}^{-}$anions from the aqueous solution to a toluene solution using tetraoctylammonium bromide as phase transfer agent. Then, $A u C l_{4}^{-}$is reduced with aqueous sodium borohydride in presence of alkane thiol. The resulting gold nanoparticles have a small size distribution ranging from 1 to 3 nm . The control of the particle size is done by varying the ratio of thiol ligand over $A u C l_{4}^{-}$anions or varying the amount of sodium borohydride [8].

The advantages of this protocol over the one proposed by Turkevish is the facile synthesis at ambient temperature, a relative high thermal and air stability of the resulted AuNPs, a better control of the AuNP size with a narrower dispersity and finally, a relatively easy functionalization and possible further ligand substitution depending of the desired use of the AuNPs [114].

Functionlized gold nanoparticles open routes to various possible applications which are discussed in the following section.


Figure 2.8 - The protocol proposed by Brust and co-workers. Reprinted from ref. [120] (Comprehensive Analytical Chemistry, 2014, 66, 37-79, Figure 1, Copyright 2014 with permission from Elsevier).

### 2.1.2.2 Applications of ligand-stabilized gold nanoparticles

Functionalized gold nanoparticles can be used for many applications due to their physical properties and their biocompatibility. We review some applications in the field of sensing, therapeutics and electronics [123].

## Gold nanoparticles as sensors

An important application of functionalized gold nanoparticles is their use as sensors to detect chemical or biological molecules. The gold nanoparticles must satisfy two criteria to be used as sensor. First, the ligand should selectively attach to the species of interest and it should reveal some change in physical or chemical properties to be detected [123]. For instance, gold nanoparticles can be used as a colorimetric sensor because of the absorption of AuNPs in the visible region. The AuNPs undergo a surface plasmon resonance (Figure 2.9). When they aggregates, interparticle surface plasmon coupling induces a shift from blue to red absorbed visible light [124]. This method allows to detect the presence of various ions in solution such as alkali or alkaline earth metal ions 125 or heavy metal ions such as $\mathrm{Pb}^{2+}, \mathrm{Cd}^{2+}$ or $\mathrm{Hg}^{2+}$. The principle has been extended to detect anions and small organic molecules and proteins 123 .

## Therapeutics applications

Functionalized gold nanoparticles can be used for a wide range of biomedical applications. Their strong resonant absorption as well as their scattering properties render AuNPs suitable for diagnostic techniques and potential applications in targeted therapeutics [128]. Upon laser illumination, excitation of AuNPs results in a local photothermal heating that could be exploited to induce cancer cell death. On the other hand, because AuNPs can absorb or scatter near-infrared


Figure $2.9-1$ ) Electromagnetic field induces surface plasmon coupling which is defined as a collective oscillation of the free conduction electrons of the AuNPs. Reprinted from ref. [126] (Environmental Chemistry Letters, 2020, 1-14, Figure 2, Copyright 2020 with permission from Springer). 2) When nanoparticles attach to the molecules to be detected (in this scheme metallic ions), surface plasmon coupling induces red shift absorption of light so that aggregated nanoparticles appear to be red. Reprinted from ref. [127] (Chemical reviews, 2012, 112, 2739-2779, Figure 4, Copyright 2012 with permission from ACS).
light (region of the spectrum ranging from $700-1100 \mathrm{~nm}$ ), they can be used as contrast agents in the diagnosis imaging of tumors [129]. Nanoparticles-based therapeutics and devices rely on their accumulation in the tumor that can be induced by modifying gold nanoparticles ligands with suitable compounds like PEG or conjugating ligands with antibodies or markers expressed in higher quantity in the tumor [123, 130].

## AuNPs as building blocks in nanoelectronics

Ligand-stabilized gold nanoparticles can be integrated in electronic devices such as singleelectron devices. In a macroscopic metallic conductor, the current is due to the motion of a high number of electrons. In an isolated piece of metal like nanoparticles, the number of electrons becomes countable and one has to take account of quantum physical effects. The simplest device consists of one or several ligand-stabilized gold nanoparticles placed between two electrodes with a current that can appear by tunneling effect [8]. Scanning tunneling microscope (STM) is based on this principle (Figure 2.10) and provides determination of structural and electronic characteristics of a sample.

If the gold nanoparticles are comprised between two electrodes and an additional gate electrode is introduced to externally control the current flow, the resulted system describes a single electron transistor [131].

However, the fabrication of nanoelectronic devices using nanoparticles is challenging and a suitable method to build them is based on the bottom-up approach. Indeed, in order to produce

2)


Figure $2.10-1$ ) Electronic device made of a scanning tunnelling miscroscope (STM) to investigate ligand-stabilized gold nanoparticles. 2) Single electron transistor (SET) with several assembled ligand-stabilized AuNPs. Pictures adapted from [8] (Philosophical Transactions of the Royal Society A, 2010, 368, 1405-1453, Figure 7 and Figure 15, Copyright 2010 with permission from The Royal Society publishing).
reliable and reproductible devices, it is a prerequisite to control the formation of such an assembly process in order to form one, two and three dimensional organized assembly of nanoparticles [ $\mathbb{\theta}$.


Figure $2.11-1$ ) AFM images of a chain made of positive packed ligand-stabilized AuNPs between two gold electrodes on ammonium persulfate (APS) coated Si surface. 2) I-V curve of the resulting chain. Reprinted from ref. [132] (Journal of Applied Physics, 2009, 105, 074302, Figure 3, Copyright 2009 with permission from AIP).

A proper way to achieve arrays of well-organized AuNPs at the nanometer scale consists of using a DNA molecule as a scaffold to build nanowires that display ohmic behavior at room temperature as shown in Figure 2.11 The control of electrostatic interactions between AuNPs and DNA enables to monitor the formation of AuNP-DNA assemblies and we discuss formation of such assemblies in the following section.

### 2.1.2.3 Self-assembly of ligand-stabilized gold nanoparticles on DNA templates

DNA molecule is the molecule that stores and transmits genetic information in biological systems. The field of nanotechnology uses this molecule out of its biological context as a building block to assemble various structures [6]. More specifically, the assembly of ligand-stabilized gold nanoparticles with DNA is a promising path in the spirit of the "bottom-up" approach for the design of new materials of well-controlled size and shape. Indeed, DNA molecule (or sometimes DNA superstructures like DNA origami [6, 133]) is used as a scaffold on which is assembled gold nanoparticles with positive ligands through electrostatic interactions.

In a general way, the compaction of DNA has been studied in detail by Zinchenko and coworkers [87] where the different modes of interaction between DNA and cationic nanoparticles are described depending of the nanoparticle size: 1) the DNA adsorb on nanoparticles larger than 40 nm; 2) the DNA wraps around nanoparticles of size $15 \mathrm{~nm} ; 3)$ smaller nanoparticles of size 10 nm adsorb on the DNA molecule [74] and we are mainly interested in the last situation since DNA is aimed to be used as a scaffold to assemble nanoparticles. The goal of this section is not to provide an exhaustive review of experiments, but only to mention key experiments. The review of Kumar and co-workers [134 as well as those of Julin and co-workers [135] present a wide range of possible structures that can be formed with DNA and metallic nanoparticles.

## Self-assembly of AuNPs on a single DNA template

Kim and co-workers fabricated linear arrays of electrostatically assembled positive gold nanoparticle on a stretched DNA bound to a ammonium persulfate (APS) coated Si surface [136, 132]. They show that the AuNP treatment time on DNA, diameter of nanoparticles or DNA concentration modified the spacing of aligned gold nanoparticles adsorbed on the DNA. The treatment time and DNA concentration can be tuned in order to have a close-packed assembly of nanoparticles adsorbed on the stretched DNA (Figure 2.12).

Similar studies have been performed previously by Woehrle, Warner and Hutchison 69] where the nano-assembly of DNA-gold nanoparticle was formed in ultrapure water during 5 minutes before deposition onto silicon-monoxide coated grids. The thickness of the positive ligand shell terminated by a positive quaternary ammonium group has a length of 0.7 nm (TMAT ligand), 1.0 nm (MEMA ligand) and 1.4 nm (PEGNME ligand). The TEM images of stretched DNA


Figure $2.12-1)$ AFM images of stretched DNA chains on which are adsorbed cationic gold nanoparticles. For a), b) and c), the DNA concentration is fixed to $3 \mathrm{ng} / \mu \mathrm{L}$ and only the treatment time is modified respectively from 10, 20 and 30 minutes. For d), the DNA concentration is $0.03 \mathrm{ng} / \mu \mathrm{L}$ and treatment time is 20 minutes. 2) Distance between gold nanoparticles as a function of a) the treatment time, b) the DNA concentration. Reprinted from ref. [132] (Journal of Applied Physics, 2009, 105, 074302, Figures 1-2, Copyright 2009 with permission from AIP).
with assembled nanoparticles of different ligand shell thicknesses is presented in Figure 2.13. The histograms present measured interparticle spacing distance gathered over hundreds of samples.

The calculated interparticle spacing is $D=2 L$ where L is the thickness of the ligand shell which reveals that a close-packed assembly of nanoparticles can be formed on a stretched DNA through a bottom-top approach.

Gold nanoparticles can be also attached to DNA structures such as DNA origami through hybridization [137, 138, 139]. The DNA molecule can be shaped into various objects (like a paper origami [140]) and the resulted structures, i.e. the DNA origami, can be functionalized with gold nanoparticles. The general procedure to assemble the gold nanoparticles on the DNA origami is described in Figure 2.14.

The key idea is to use complementarity of DNA base pair: the functionalized gold nanoparticles with single strand DNA are hybridized with complementary single strand DNA attached on the DNA origami. Given that it is possible to choose the position of single strand oligonucleotides


Figure 2.13 - Transmission electron microscopy (TEM) images of close-packed assembly of gold nanoparticles on DNA. The thickness of the ligand shell at the gold nanoparticle surface is $0.7,1.0$, 1.4 nm from right to left. The corresponding interparticle spacing distance histogram is $1.5 \pm 0.3 \mathrm{~nm}$ ( $\mathrm{N}=630$ samples ), $2.1 \pm 0.4 \mathrm{~nm}(\mathrm{~N}=549$ samples) and $2.8 \pm 0.4 \mathrm{~nm}(\mathrm{~N}=473$ samples). Pictures taken from ref. 69] (Langmuir, 2004, 20, 5982-5988, Figures 1-2, Copyright 2009 with permission from ACS).


Figure 2.14 - Principle of DNA attachment process on a DNA origami by hybridization. A) DNA origami template is assembled and single strand DNA oligonucleotides (in red) are attached at its surface. B) Gold nanoparticles are functionalized with thiolade oligonucleotides (which are also single strand DNA). C) The DNA origami template is deposed on a silicon oxide surface. D) Solution of single strand DNA functionalized gold nanoparticles is deposed on the DNA origami template and hybridization of complementary single strand DNA happens, resulting in attachment of nanoparticles on the DNA origami template. Most of the time, the hybridization is performed in presence of monovalent or divalent salt in order to screen electrostatic repulsion between single strand oligonucleotide to favor hybridization [138]. Pictures taken from ref. [139] (RCS Advances, 2014, 5, 8134-8141, Figures 2, Copyright 2014 with permission from RCS).
attached on the DNA origami, it enables control of AuNP alignement and spacing on the DNA origami.

## Self-assembly of AuNPs with multiple DNA templates: AuNP-DNA superstructures

Long double strand DNA molecules can be used as templates to assemble metal nanoparticles that results in AuNP-DNA superstructures. Such assemblies can be formed either by electrostatic interactions [141, 68, 142, 143] or by complementary binding [144, 145].

Warner and Hutchison used cationic-ligand gold nanoparticles of small diameter ( $\mathrm{d}<2 \mathrm{~nm}$ ) to be assembled with DNA molecules of length ranging from 42 nm ( $\sim 152$ base pairs) to $8 \mu \mathrm{~m}$ ( $\sim 23,130$ base pairs) in ultrapure water during an incubation time ranging from 5 minutes to 3 hours. The assemblies are formed in excess of AuNP over DNA (respectively $0.35 \mu \mathrm{~g} / \mu \mathrm{l}$ vs 0.05 $\mu \mathrm{g} / \mu \mathrm{l})$. Then, the solution is deposed on a silicon-monoxide-coated TEM grid surface and excess of water and AuNP were removed with filter paper before drying under ambient temperature. The structures are either linear AuNPs/DNA assemblies or 2D/3D structures such as ribbon and branched AuNP/DNA structures (Figure 2.15). A close-packed arrangement of AuNPs along DNA is found and structures have a length of $\sim 1 \mu \mathrm{~m}$, which is surprising because of the high charge carried by the nanoparticles ( $\sim 100$ ligands $=100$ positives charges) that would promote DNA bending and partial coverage of DNA. It is possible that the high degree of packing is induced by migration of nanoparticles along the DNA molecule, allowing them to consolidate into closely spaced assemblies on the DNA. On the other hand, it could be possible that the nanoparticles closespacing arises from a nucleation and growth mechanism similar to the assembly of poly-l-lysine on DNA [142.

In order to obtain structures with higher degree of organization, DNA origami can also be used to assemble nanoparticles into superlattice through electrostatic interactions. Because of the higher stiffness of DNA origamis compared to simple DNA chains, the resulted assemblies can adopt a crystalline structure. Julin and co-worker used helix bundle of DNA origami to construct tetragonal supperlattice in presence of cationic stabilized-ligand gold nanoparticles [146]. The DNA origami are constituted of 6 double helix DNA connected to each other (Figure 2.16 picture 1) along their long axis to form a hexagonal bundle. Positive gold nanoparticles were used with ligands


Figure 2.15 - Transmission electron microscopy images of different assemblies of AuNP/DNA through electrostatic interaction. Left) Linear assembly made up of a DNA with nanoparticles of diameter of $\mathrm{Y}=1.9 \pm 0.8 \mathrm{~nm}(\mathrm{~N}=790)$ with an inter-particle spacing $\mathrm{X}=1.4 \pm 0.5 \mathrm{~nm}(\mathrm{~N}=$ 130). Middle) Ribbon assembly. Right) Branched assembly. Pictures taken from ref. [142] (Nature Materials, 2003, 2, 272-277, Figures 1-2, Copyright 2003 with permission from Nature Publishing Group).


Figure $2.16-1)$ DNA origami used as self-assembly scaffold: 6-helix bundle (6HB). 2) Ligandstabilized gold nanoparticles. 3) Assemblies are formed upon decrease in ionic strength. 4) TEM picture of the assembly and schematic of the tetragonal structure determined by small angle X-ray scattering (SAXS). Pictures taken from ref. [146] (Nanoscale, 2019, 11, 4546-4551, Figures 1-4, Copyright 2019 with permission from RSC).
terminated by a quaternary ammonium group so that the nanoparticles are positively charged for a wide range of pH values (Figure 2.16 picture 2). Given that DNA and nanoparticles are highly charged, it is required to control the electrostatic interaction to avoid kinetically trapped structure (Figure 2.16 picture 3). At initial stage, components are mixed in $500-750 \mathrm{mMol} / \mathrm{l}$ of NaCl to screen electrostatic interaction and concentration of salt is progressively reduced by $50 \mathrm{mMol} / \mathrm{l}$ every 30 minutes until $0 \mathrm{mMol} / \mathrm{l}$. The assembly adopts a well-ordered tetragonal structure (Figure 2.16 picture 4).

We provided some examples of PECs structures than can be formed mostly with DNA and stabilized-ligand cationic gold nanoparticles. It is possible to control DNA condensation in presence of cationic AuNPs which play the role of compacting agents.

### 2.2 Project 2: Polyelectrolyte complexation between PEDOT:PSS and ionic liquids

### 2.2.1 Review about conducting polymers

### 2.2.1.1 Conjugation and $\pi$-bond

It is necessary to remind the conjugation of the carbon atom given that we will deal with organic polymers in the coming section.

The electronic ground state of the carbon atom is $1 s^{2} 2 s^{2} 2 p^{2}$ where the valence electrons are those localized in the outer atomic orbitals $2 \mathrm{~s}^{2}$ and $2 \mathrm{p}^{2}$. However, the configuration is predicted to be more stable either if all orbitals are half filled or completely filled. Consequently, the electronic structure writes $1 \mathrm{~s}^{2} 2 \mathrm{~s}^{1} 2 \mathrm{p}^{3}$ where each atomic orbital has one electron. Depending of the surrounding atoms that can create covalent bond with carbon, the orbitals 2 s and 2 p can combine to evolve into hybridized orbitals. The carbon atom can be either $\mathrm{sp}^{3}, \mathrm{sp}^{2}$ or sp hybridized.

- $\mathbf{s p}^{3}$ hybridization (Figure 2.17.1) happens when the carbon atom is surrounded by four atoms. The orbital 2 s and the three orbitals 2 p hybridize into 4 orbitals $\mathrm{sp}^{3}$ that can make $\sigma$ bond with other atoms. The carbon atom geometry is tetrahedral and the structure is saturated.

1) $\mathrm{sp}^{3}$ hybridization

2) 

$\mathrm{sp}^{2}$ hybridization

$\pi$-bond
Single bond


Double bond
3)
sp hybridization


Triple bond

Figure 2.17 - Presentation of the carbon possible hybridization. 1) $\mathrm{sp}^{3}$ hybridization. 2) $\mathrm{sp}^{2}$ hybridization. 3) sp hybridization. Picture taken from ref. [147] (Phd Thesis, 2017, 11, Figure I-1, Copyright 2017 with permission from University of Grenoble).

- $\mathbf{s p}^{2}$ hybridization (Figure 2.17-2) happens when the carbon atom is surrounded by three atoms. The orbital 2 s and two orbitals 2 p hybridize into 3 orbitals $\mathrm{sp}^{2}$ while the remaining 2 p orbital is perpendicular ( $2 \mathrm{p}_{z}$ orbital). The $\mathrm{sp}^{2}$ orbitals create a $\sigma$ bond but if two $\mathrm{sp}^{2}$ carbon atoms are closed enough, the $2 \mathrm{p}_{z}$ orbital overlap to create a $\pi$ bond. Hence, a conjugated bond is create by the $\sigma$ and the $\pi$ bond between the $\mathrm{sp}^{2}$ hybridized carbon atoms. The structure is planar and is qualified as unsaturated.
- sp hybridization (Figure 2.17-2) happens when the 2 s orbital is combined with a 2 p orbital resulting in two sp hybridized orbitals and 2 p orbitals. If two sp carbon atoms are close to each other, a triple bond appear due to a $\sigma$ bond and two $\pi$ bonds and the structure is linear and unsaturated


### 2.2.1.2 Semi-conducting polymers

It has been established that polymers can conduct electricity since the pioneering work of Weiss' team in 1963 that studied conductivity in polypyrroles [148]. The discovery of semi-conducting
polymers is attributed to Hideki Shirakawa, Alan J. Heeger and Alan G. MacDiarmid with the partial redox doping of a polyacetylene thin film that induces an increase of conductivity of several order of magnitude (up to $10^{5} \mathrm{~S} / \mathrm{cm}$ ) [149, 150, 15] and they have been awarded by a Nobel prize in 2000.



Polyacetylene Polythiophene


Polyaniline


Poly(3,4-ethylen dioxythiophene)

Figure 2.18 - Example of some of the most used semi-conducting polymers.

Semi-conducting polymers are based on an alternation of single and double chemical bonds so that $\mathrm{p}_{z}$ carbon orbitals result in a $\pi$-conjugated system that induces carbon electrons to be delocalized along the polymer backbone. The advantages of semi-conducting polymers are their low price and flexibility as well as their intrinsic conducting properties that can be exploited for many applications [152].

We present some of the most common semi-conducting polymers in Figure 2.18 Polyacetylene is one of the first synthesized conducting polymers and is studied in non-linear physics because propagation of charge can be described by soliton theory [153]. Polypyrrole is a water soluble and bio-compatible polymer with prospective applications for drug delivery or for artificial muscle component [154]. Polyaniline is a polymer which is simple to synthesize and can be used in circuit board manufacturing [155] and precursor for N-doped carbon materials [156]. Polythiophenes are easy to synthesize and have promising applications when added with substitutes that modify solubility or conductivity of the resulted thin films [157]. Poly(3,4-ethylendioxythiophene) also denoted as PEDOT has been synthesized first by Jonas and co-workers in 1988 [152]. Because of its poor water solubility, PEDOT is mixed with the water soluble negative polymer poly(styrenesulfonate)
to form a water soluble polyelectrolyte complex PEDOT:PSS.

### 2.2.1.3 Conduction of semi-conducting polymers

The electrical conductivity $\sigma$ usually expressed in $\mathrm{S} / \mathrm{cm}$ measures the ability of a material to carry an electric current. The electrical conductivity depends on the density of charges in the material $n$ and their mobility $\mu$ through the following relationship:

$$
\begin{equation*}
\sigma=q \times n \times \mu \tag{2.2}
\end{equation*}
$$

Materials are classified according to their conductivity into three categories: insulators, semiconductors and metals (Figure 2.19).


Figure 2.19 - Conductivity scale of various materials. Picture adapted from ref. [158] (Chemical Communications, 2015, 51, 16886-16899, Figure 2, Copyright 2015 with permission from RSC).

The electronic band structure theory explains the different physical behaviors between these three classes of materials. The electronic structure can be described by two types of bands: one type in which electrons can move freely and another type forbidden to electrons. More specifically, the available bands for electrons involve the conduction band $(\mathrm{CB})$ and the valence band $(\mathrm{CB})$. The conduction band corresponds to the band of lowest energy unoccupied by electrons while the valence band is the band of highest energy occupied by electrons. The valence and conduction bands are separated by a band of forbidden energy: the band gap. The conductivity of the material depends on the capacity of the electrons in the valence band to switch to the conduction band (Figure 2.20).

- For a bandgap of energy $E_{g}$ higher than 5 eV , the electrons cannot migrate from the valence band to the conduction band and the material is insulating.
- The material is a semiconductor if the bandgap is comprised between 1 and 5 eV . The electrons leave the valence band and migrate to the conduction band because of the thermal agitation. Hence, the contribution to the conductivity comes from the holes (positrons) in the valence band and the free electrons in the conduction band.
- On the contrary, when the conduction and valence bands are overlapping, electrons move freely from valence to conduction band and the material is a conductor.


Figure 2.20 - Bandgap diagram for insulators, semi-conductors and conductors. Picture adapted from ref. [159] (Phd Thesis, 2014, 12, Figure I-3, Copyright 2014 with permission from University of Franche-Comté).

Untreated semi-conducting polymers displays a low conductivity despite $\pi$-conjugated bonds $\left(\sigma \sim 10^{-15}-10^{-8}[160]\right)$. However, addition of charge carriers in the polymer by a doping redox reaction increases its conductivity by oxidation or reduction of the monomers of the polymer chain.

- In case of a reduction reaction, a type $n$ doping takes place: an excess of negative charge (electrons) occurs so that a supplemental level of energy appears under the conduction band. Hence, the energy associated to the bandgap is lower than for intrinsic semiconductors.
- On the contrary, for a type $p$ doping, conduction is performed through positive charges (holes)
because electron can jump to an energy level just above the valence band, leaving holes in the conduction band.


### 2.2.1.4 Conduction mechanism in semi-conducting polymers

When semi-conducting polymers are doped by introduction of electrons or holes, a local deformation of the conjugated chain happens in order to reduce its overall energy. The conduction mechanism in semi-conducting polymers arises from the coupling between the charge and the deformation. More precisely, the propagation of the charge along the conjugated chain is performed through an alternation of simple and double bonds. One has to distinguish between degenerated and non-degenerated systems (Figure 2.21).


The two solitons remain independent
b. Non-Degenerated system


Figure 2.21 - Charge carriers along backbone of degenerated and non-degenerated systems. Picture taken from ref. [147] (Phd Thesis, 2017, 15, Figure I-5, Copyright 2017 with permission from University of Grenoble).

An example of degenerated system is polyacetylene for which the defect propagates after temporary permutation of bonds along the backbone without altering the energy of the chain.

One the contrary, in a non-degenerated system such as polythiophene, the introduction of a charge induces a local deformation that changes the benzoid structure to quinoid structure associated to a change in energy and the defect is called polaron. If another charge is injected, it preferentially localized near the deformation to avoid an increase of the chain energy. In that case, this defect is called bipolaron. The nature of the defect can be determined through electron paramagnetic resonance spectroscopy (EPR) which detects polaron involving unpaired electrons of spin $1 / 2$ while bipolaron has no spin [11].

### 2.2.2 The PEDOT semi-conducting polymer

PEDOT is one of the most studied semi-conducting polymer with promising applications due to its excellent thermal and air stability, low cost and relatively simple synthesis. Indeed, PEDOT represents a suitable alternative to other conducting polymers that present several drawbacks. For instance, polyacetylene is very air-sensitive and is not suitable for industrial applications. The lack of stability of the conjugated chain in the doped state can be overcome by insertion of electron-donating atoms like nitrogen (N) or sulfur (S). Consequently, scientists focus on conducting polymers such as polyaniline, polypyrrole or polythiophene. Unfortunately, these polymers display several major problems. The aniline which is the precursor of polyaniline is a hazardous compound for health and environment. Polypyrrole displays toxicity and high vapor pressure which impede its industrial use. Polythiophenes are environmental friendly but are insoluble and unstable. Consequently, PEDOT represents a great alternative for relevant industrial applications and a lot of efforts are devoted to study its physical and chemical properties.

### 2.2.2.1 Synthesis of PEDOT:PSS and PEDOT:Tos

The PEDOT synthesis can be done by two types of polymerization reactions which are either an electrochemical or oxidative chemical polymerization of the EDOT-based monomers. PEDOT can be synthesized through oxidation of EDOT in an electrochemical cell that contains a working electrode, a reference electrode and a counter electrode immersed in a solution of EDOT monomers and poly(styrene sulfonate) sodium salt [161]. The EDOT monomers are oxidized at the metal
working electrode to form cations which combine together to make the PEDOT chains. The other route to synthesize PEDOT is through chemical oxidation by iron (III) complexes. The most used complex is $\mathrm{Fe}(\mathrm{Tos})_{3}$ and the synthesis route is described in Figure 2.22 which results in PEDOT:Tos where Tos is the monomer unit of poly(styrene sulfonate) (PSS). The polymerization mechanism is described in three steps:

- The complex $\mathrm{Fe}(\mathrm{Tos})_{3}$ oxidizes EDOT and is reduced to $\mathrm{Fe}(\mathrm{Tos})_{3}^{-}$.
- Two oxidized EDOT units combine to form a dimer that is deprotonated by water.
- The polymer chain is assembled after repetition of the above processes. Finally, the remaining $\mathrm{Fe}^{3+}$ ions doped the PEDOT chain and the Tos ${ }^{-}$ions play the role of counterions.


Figure 2.22 - Synthesis of PEDOT:Tos described by Mueller and co-workers. Picture taken from ref. [162] (Polymer, 2012, 53, 2146-2151, Figure 1, Copyright 2012 with permission from Elsevier).

It is also possible to synthesize PEDOT by oxidation of EDOT units in presence of $\mathrm{Na}_{2} \mathrm{~S}_{2} \mathrm{O}_{8}$ in a solution that contains PSS that stabilized the resulting PEDOT:PSS solution. The corresponding structure is presented in Figure 2.23 PSS plays an important role in the formation of a soluble PEC in water. First, PSS is always in excess and brings negative charges and serves as counterions of the PEDOT chains, stabilizing the PEC. Then, polymerization of EDOT into PEDOT generates positive charge every three or four EDOT units [163] and induces an electrostatic interaction between PEDOT and PSS.
1)
2)
EDOT


PSS

Figure 2.23 - Chemical structure of PEDOT:PSS.

### 2.2.2.2 Structure of PEDOT:PSS/PEDOT:Tos thin films

PEDOT:PSS is a disordered material made of polymers of different lengths, with defects non uniformly distributed along the chains as well as various effective distances separating delocalized electrons on the chains. The synthesis of PEDOT chains from polymerization results in chains composed of 6-18 EDOT units [164]. More precisely, the PEDOT:PSS is constituted of long PSS chains (molar mass $\mathrm{M}_{P S S} \sim 400000 \mathrm{~g} / \mathrm{mol}$ which is equivalent to 2200 styrene sulfonate units) on which are electrostatically bound smaller PEDOT chains (molar mass $\mathrm{M}_{\text {PEDOT }} \sim 1000-2500$ $\mathrm{g} / \mathrm{mol}$, which is indeed equivalent to 6-18 EDOT monomers).




Figure 2.24 - Structure of PEDOT:PSS complex. Picture adapted from ref. [165] (Smart Materials and Structures, 2014, 23, 074010, Figure 1, Copyright 2012 with permission from IOP Publishing).

The structure of thin films of PEDOT:PSS is organized in distinct domains of size ranging from 10-100 nm (Figure 2.24 [166, 165. A domain is presented as PEDOT units encapsulated by a PSS shell which inhibits the formation of large conducting domains resulting in poor transport properties of PEDOT:PSS untreated films [167, 168, 169. It is desired to form more ordered and extended PEDOT structures in order to increase charge mobility and thus transport properties.

Understanding the structure of PEDOT materials is thus a prerequisite for developing strategies to enhance its electrical transport properties.

The first experimental investigation of a structured PEDOT material was performed by Aasmundtveit and co-workers on PEDOT:Tos films 170 obtained by spin-coating. Analysis by grazing incidence wide angle X-ray scattering (GIWAXS) reveals that the structure of PEDOT:Tos was anisotropic with Tos anions forming alternating layers of $\pi$ stacked PEDOT chain (Figure 2.25).


Figure 2.25 - Structure of PEDOT:Tos matrix. Picture adapted from ref. [170] (Synthetic Metals, 1999, 101, 561-564, Figure 4, Copyright 1999 with permission from Elsevier).

The lattice parameters of the orthorombic structure is $a=14.0 \AA, b=6.8 \AA, c=7.8 \AA$. The $\pi$-stacking structure adopted by the PEDOT chains remains the same in synthesized PEDOT films and is not dependent of the counterions [163, 171, 172] but the lattice parameters and degree of crystallinity are highly dependent on the sample or deposition techniques. The morphology of PEDOT is an important asset in the quest for higher conductivity as illustrated by the work of Cho and co-workers where PEDOT single nanocrystals synthesized by vapor-phase polymerization display a conductivity as high as $8000 \mathrm{~S} / \mathrm{cm}$ with a low doping level of $10 \%$.

### 2.2.2.3 Applications of PEDOT:PSS

PEDOT is the most promising semi-conducting polymer in term of industrial applications. Indeed, PEDOT:PSS has been successfully used for energy conversion devices 9 and for biomedical applications [10].

## Applications in energy conversion devices

PEDOT:PSS can be used in organic solar cells (OSCs) to replace transparent electrodes made of indium tin oxide (ITO) that have serious drawbacks such as high temperature process, mechanical brittleness in addition to indium scarcity. These drawbacks prevent large-scale commercialization of flexible, wearable and printable devices. The first attempt to use PEDOT:PSS instead of ITO in organic solar cell has been performed in 2002 by Zhang and co-workers [173] with a power conversion effeciency (PCE) of $\sim 3.0 \%$ lower to the PCE of ITO-based OSCs $(\sim 5.4 \%)$. Moreover, PEDOT:PSS is used also as a hole transport layer in organic solar cells [174]. In addition, PEDOT:PSS has been also used as transparent electrode [175] and hole transport [176] in order to obtain ITO-free perovskite solar cells (PSCs). PEDOT:PSS is expected also to become an important component for the next-generation of stretchable electronics and will be present not only in flexible OSCs but also in organic light emission diodes (OLEDs) and may have also potential biomedical applications as strain sensors or organic thin film transistors (OTFTs) [177].

## Biomedical applications

Strain sensors are composed of a stretchable conductor (like PEDOT:PSS) and an elastomer as underlying substrate. When the stretchable conductors are deformed under external forces, the resistance of the device changes inducing a variation in the voltage of the output electric signal. Strain sensors are thus able to detect deformation of target objects and could be used as electronic skin or health monitor [178] for motion and pulse sensors, body temperature or electrocardiograms (Figure 2.26).

Organic thin film transistors (OFETs) are made of a source and a drain electrode connected with an organic semiconducting layer and a dielectric layer deposited between the semiconductor and the gate electode. If a voltage is applied at the gate, charge carriers are forming conducting channels between source and drain electrodes by accumulating between the semiconductor and the dielectric [177]. The use of PEDOT:PSS as electrode or interconnector results in stretchable OFETs that convert ionic signals into electric signals and can have applications for biomedical chemical sensors of metabolite level or disease biomarkers (Figure 2.26.


Figure 2.26 - Examples of applications of stretchable sensors for health monitoring. Picture adapted from ref. [178] (Journal of Materials Chemistry B, 2018, 28, 8625-8631, Figure 1, Copyright 2018 with permission from RSC).

### 2.2.3 PEDOT:PSS conductivity enhancement by ionic liquids

Because of the potential processibility, conductivity and film forming properties of PEDOT:PSS (and in extend PEDOT:Tos), a huge effort has been performed in the scientific community to find experimental route to the synthesis of high conductive films. In particular, researchers aim to improve electrical conductivity of PEDOT-based materials. Indeed, untreated thin films of PEDOT:PSS exhibit poor electrical transport properties (conductivity $\sigma<10 \mathrm{~S} / \mathrm{cm}$ and mobility $<10^{-3} \mathrm{~cm}^{2} \mathrm{~V}^{-1} \mathrm{~S}^{-1}$ ) [168, 179, 180]. The poor electrical transport properties are attributed to the structure of PEDOT:PSS thin films in which the hydrophilic (but insulating) PSS encapsulate the hydrophobic p-doped PEDOT cores. Conductivity enhancement of PEDOT:PSS has been achieved by pre-treatment that consists of adding chemicals in the PEDOT:PSS solution such as anionic surfactants [181], salts [182], organic solvents such as dimethyl sulfoxide (DMSO) [183], $\mathrm{N}, \mathrm{N}$-dimethylformamide (DMF) [184] or ethylene glycol [180]. Such treatments enhance electrical conductivity of PEDOT:PSS up to $2000 \mathrm{~S} / \mathrm{cm}$. The PEDOT:PSS conductivity enhancement mechanism is still under debate and various hypothesis have been proposed such as phase segregation,
chain conformation or doping. However, the most believed hypothesis is that small ions reduce electrostatic interaction to segregate PEDOT from PSS. Along this line, a significant enhancement (up to $2000 \mathrm{~S} / \mathrm{cm}$ ) of PEDOT:PSS conductivity has been achieved recently by addition of ionic liquids 11.

### 2.2.3.1 Definition and general properties of ionic liquids

Ionic liquids (ILs) are molten salts characterized by a fusion temperature inferior to $100^{\circ} \mathrm{C}$ at atmospheric pressure [185]. ILs are rather old materials that were discovered in the beginning of the $\mathrm{XX}^{t h}$ century with the work of Walden on IL $\left[\mathrm{EtNH}_{3}\right]^{+}\left[\mathrm{NO}_{3}\right]^{-}$that has a fusion temperature of $12^{\circ} \mathrm{C}$ [186]. However, research on ILs really took off in the 1990s and covers nowadays many applications such as electrolytes for batteries [187, 188], drugs in pharmaceutics [189], dispersants [190], gas storage agents [191] and so on.

Most of the time, an IL is constituted of an organic cation and an organic or inorganic anion. Because of the various existing anions and cations, there could be more than $10^{1} 2$ combinations of [cation-anion] [192] resulting for each case in a new IL with its intrinsic properties. In general, IL cations are asymmetrical and the most common cation contains an aromatic ring like the alkylpyrrolidiniums, alkylpyridiniums, imidazoliums and piperidiniums (Figure 2.27). IL anions are generally inorganic atomic anions such as chloride $\mathrm{Cl}^{-}$, iodide $\mathrm{I}^{-}$and bromide $\mathrm{Br}^{-}$or molecular inorganic anions such as tetrafluoroborate $\mathrm{BF}_{4}^{-}$or hexafluorophosphate $\mathrm{PF}_{6}^{-}$(Figure 2.27). However, IL anions can be organic molecules such as $\left(\mathrm{CF}_{3} \mathrm{SO}_{2}\right)_{2} \mathrm{~N}^{-},\left(\mathrm{CF}_{3}\right)_{2}, \mathrm{C}_{2} \mathrm{~N}_{3}{ }^{-}$or $\mathrm{CF}_{3} \mathrm{SO}_{3}^{-}$ (Figure 2.27)

ILs have interesting properties that make them suitable for various applications and we summarized several of them below.

- ILs are generally not flammable excepted those based on nitrates and picrates ions. They are known also to be stable and non volatile.
- They have a fusion temperature that depends on molecular composition and pyridinium as well as imidazolium based IL have a fusion temperature of $80^{\circ} \mathrm{C}$ 193, 194 and are stable up to $300^{\circ} \mathrm{C}$.
- ILs can be used as solvent for organic and inorganic compounds [195.


## IL cations

Imidazolium







| Quaternary phosphonium | Quaternary ammonium |
| :---: | :---: |
| $\mathrm{R}^{4},{ }^{1}$ | $\mathbf{R}^{4}{ }_{\prime \prime}+\mathrm{R}^{1}$ |

## IL anions


Bis(trifluoromethylsulfonyl)amide Bis(trifluoromethyl)amide




Figure 2.27 - Examples of common cations and anions that form ILs

- The length of the alkyl chain plays an important role for the solubility of IL in water or in organic solvent. The ILs can be classified into hydrophobic and hydrophilic ones depending of the type of IL anion [196]. For instance, point-like inorganic anions such as $\mathrm{Cl}^{-}, \mathrm{Br}^{-}$ will be easily solvated in water contrary to hydrophobic charge-dispersed anions such as hexafluorophospate $\mathrm{PF}_{6}^{-}$or tetracyanoborate $\mathrm{B}(\mathrm{CN})_{4}^{-}$.
- ILs display a typical density of $1.0-1.6 \mathrm{gcm}^{-3}$ and have a viscosity $10-100$ times higher than water or organic solvents [197]. The viscosity increases linearly with the length of the alkyl chain for a given anion. The density of ILs depends also of the nature of the anion and decreases with an increase in alkyl chain length 197.
- ILs have a large electrochemical capacity and can have tension of 5-6 V [198] which made them suitable as electrolyte for fuel cells [199].


### 2.2.3.2 Controlling ordering in PEDOT:PSS solution with imidazolium-based ionic liquid for PEDOT:PSS conductivity enhancement

The effect of four imidazolium-based ILs on the morphology and electrical transport properties of PEDOT:PSS has been investigated [11. For each IL, the cation is 1-ethyl-3-methylimidazolium

EMIM $^{+}$because of its low melting point and electrochemical stability while the IL anion $\mathrm{X}^{-}$can be the single atom $\mathrm{Cl}^{-}$, ethylene sulfonate $\mathrm{ES}^{-}$but also anions that contain carbonitrile electronwithdrawing groups which are tricyanomethanide $\mathrm{TCM}^{-}$or tetracyanoborate $\mathrm{TCB}^{-}$. The chemical structures of ILs are presented in Figure 2.28
CI anion ES anion TCM anion TCB anion
$\mathbf{X}=\quad \mathrm{Cl}^{-}$




Figure 2.28 - Structures of the ILs EMIM:X.

Conductivity enhancement of PEDOT:PSS thin films that contain IL has been performed in the following way [11] (Figure 2.29). Solution of 0.5 ml of $\mathrm{IL}(\mathrm{c}=0.15 \mathrm{~mol} / \mathrm{l})$ is mixed with 1 g of commercial solution of PEDOT:PSS solution (Clevios ${ }^{T M} \mathrm{PH} 1000$ with a mass ratio of PEDOT:PSS of $1: 2.5$ ) and stirred at room temperature during 10 minutes. Then, PEDOT:PSS + IL films are prepared by spin-coating the solution on a glass substrate and sonicated in deionized water, acetone and isopropyl alcohol during 10 minutes. Finally, the resulting films are dried at $130^{\circ} \mathrm{C}$ during 15 minutes.


Figure 2.29 - Process of PEDOT:PSS film fabrication containing doping IL. The IL remains in the PEDOT:PSS thin film after washing and removing of PSS excess.

The hypothesis for conductivity enhancement in presence of IL consists of an ion exchange between PEDOT:PSS and IL, described by the following equation:

$$
\begin{equation*}
P E D O T: P S S+E M I M: X \rightarrow P S S: E M I M+P E D O T: X \tag{2.3}
\end{equation*}
$$



Figure 2.30 - a) Electrical conductivity of untreated (pristine) and IL-treated PEDOT:PSS thin films. b) Charge-carrier mobility, c) $\pi-\pi$ stacking distance of untreated and IL-treated PEDOT:PSS films. d) Illustration of rearrangement of PEDOT:PSS without IL and with EMIM:TCB from the hypothesis of ion exchange between PEDOT:PSS and IL. Picture adapted from [11] (Advanced Materials, 2016, 28, 8625-8631, Figures 3-4, Copyright 2016 with permission from John Wiley and Sons).

Because of electrostatic interactions, it is believed that IL helps PSS to decouple from PEDOT which induces a subsequent self-assembly process leading to extended $\pi$-stacked PEDOT chains [11] as illustrated in Figure 2.30 c. An increase in conductivity of 5000 -fold has been reported in presence of EMIM:TCB (from 0.4 to $2103 \mathrm{~S}_{\mathrm{S}} \mathrm{cm}^{-1}$ ). Similar conductivity in presence of the same IL has been reported in a previous study [200]. More modest $\sim 3500, \sim 2000$ and $\sim 900$ fold enhancement (equivalent to 1405,840 and $359 \mathrm{S.cm}^{-1}$ ) has been achieved respectively in presence of EMIM:TCM, EMIM:ES and EMIM:Cl (Figure 2.30-a).

The conductivity does not display the same behavior than the carrier mobility (Figure 2.30-b) which is directly related to the degree of ordering of PEDOT. Indeed, transport of charge carriers would happen not only along but also across the $\pi$-stacked PEDOT domains by overlapping of $\pi$-orbitals [170. It appears that an additional factor plays a role in PEDOT:PSS conductivity: the number of charge carrier (i.e. the degree of $p$-doping). Indeed, non-monotonic variation of the
number of charge carriers has been reported along the series of anions $\mathrm{X}: 1.70 \times 10^{21}(\mathrm{X}=\mathrm{Cl})$, $1.50 \times 10^{21}(\mathrm{X}=\mathrm{ES}), 1.88 \times 10^{21}(\mathrm{X}=\mathrm{TCM})$ and $2.54 \times 10^{21}(\mathrm{X}=\mathrm{TCB})$. The amount of charge carriers suggests also that the IL anion X remains in the PEDOT phase [201, 11].


Figure 2.31 - a) XPS spectra of the $\mathrm{S}(2 \mathrm{p})$ of PEDOT:PSS films with and without ILs. b) 1D scattering profile in $\mathrm{q}_{z}$ direction. d) Schematic structure of stacking of untreated PEDOT:PSS (left), compact lamellar stacking of PEDOT:PSS in presence of EMIM:Cl or EMIM:ES (middle) and compact stacking of PEDOT:PSS in presence of TCB. d) TEM images of PEDOT:PSS films with and without ILs. Pictures taken from [11] (Advanced Materials, 2016, 28, 8625-8631, Figures 1-2, Copyright 2016 with permission from John Wiley and Sons).

Evidence from a change in morphology is depicted by X-ray photoelectron spectroscopy (XPS) measurements performed on thin films of PEDOT:PSS that display two signals correponding to the core 2 p electrons of sulfur $\mathrm{S}(2 \mathrm{p})$ with one signal ranging from $172-167 \mathrm{eV}$ (sulfonate group of PSS) and another signal ranging from 167-162 eV (thiophene group of PEDOT). The PSS phosphate peak decreases in the order of IL anion $\mathrm{Cl}>\mathrm{ES}>\mathrm{TCM}>\mathrm{TCB}$ due to the removal of PSS chains
decoupled from PEDOT in presence of IL after rinsing the films with water (Figure 2.31-a).
Morphology transformation of PEDOT:PSS induced by ion exchange in presence of ILs has been investigated by grazing incidence wide angle X-ray scattering (GIWAXS) and 1D scattering $\mathrm{q}_{z}$ profiles are obtained from 2D GIWAXS images (Figure 2.31.b). The untreated PEDOT:PSS sample displays two peaks at $\mathrm{q}_{z}=0.29 \AA^{-1}(\mathrm{~d}=21.7 \AA)$ and $\mathrm{q}_{z}=0.58 \AA^{-1}(\mathrm{~d}=10.8 \AA)$ attributed to the two lamellar stacking of PEDOT around PSS in (100) direction, respectively displayed in left, Figure 2.31-c and middle Figure 2.31. c. Another peak at $1.78 \AA^{-1}$ corresponds to the $\pi-\pi$ stacking of PEDOT in the (010) direction with a distance of $3.54 \AA$.

After addition of EMIM:Cl and EMIM:ES, the major peaks corresponding to the stacking in the (100) direction for untreated PEDOT:PSS ( $\mathrm{q}_{z}=0.29 \AA^{-1}$ ) shifts respectively to $\mathrm{q}_{z}=0.48$ $\AA^{-1}(\mathrm{~d}=13.1 \AA)$ and $\mathrm{q}_{z}=0.51 \AA^{-1}(\mathrm{~d}=12.3 \AA)$. This decrease is associated to a denser lamellar structure (middle, Figure 2.31-c) of PEDOT:PSS because of PSS partial removal after washing [11.

On contrary, in presence of EMIM:TCM and EMIM:TCB, the (100) peaks are much higher at $\mathrm{q}_{z}=0.33 \AA^{-1}(\mathrm{~d}=19.0 \AA)$ and $\mathrm{q}_{z}=0.27 \AA^{-1}(\mathrm{~d}=23.3 \AA)$ respectively. There are also shoulder peaks at $\mathrm{q}_{z}=0.55 \AA^{-1}(\mathrm{~d}=11.4 \AA)$ and $\mathrm{q}_{z}=0.49 \AA^{-1}(\mathrm{~d}=12.8 \AA)$ that resemble to (100) peaks displayed by samples containing EMIM:Cl and EMIM:ES. The first of the higher peaks can be assigned to the intercalation of TCM or TCB anions in the dense lamellar packing of PEDOT:PSS (left, Figure 2.31-c).

Moreover, the crystalline domains size in the (010) were estimated to be $22.3 \AA$ (untreated), 24.9 $\AA$ (with EMIM:Cl), $25.0 \AA$ (with EMIM:ES), $27.5 \AA$ (with EMIM:TCM), $31.0 \AA$ (with EMIM:TCB) which represent stacking of $6,7,7,8$ or 9 PEDOT units. This crystalline domain growth supports the picture of a self-assembly of PEDOT after dissociation by ILs. The transformation of PEDOT:PSS to a more ordered nano-structure is shown in Figure 2.31 d and the width of the nanofibrillar domain increases along the series of anion $\mathrm{X}=\mathrm{Cl}<\mathrm{ES}<\mathrm{TCM}<\mathrm{TCB}$.

The hypothesis of ion exchange between PEDOT:PSS and IL has been also investigated numerically using density functional theory (DFT) calculations performed in gas phase with a minimal model of PEDOT:PSS [201]. A trimer of EDOT (tri-EDOT) is chosen as a minimal model of PEDOT because only one positive unit charge is carried by at least three EDOT units and PTS is chosen as minimal model for PSS. Hence, the ion exchange between tri-EDOT:PTS and the IL

EMIM: X is determined through the calculation of the free energy of ion exchange $\Delta \Delta G_{x}^{0}$ from the standard Gibbs free energy $\left(\Delta G_{x}^{0}\right)$ performed on each ion pair before and after ion exchange:

$$
\begin{align*}
\Delta \Delta G_{x}^{0} & =\Delta G^{0}(E M I M: P T S)+\Delta G^{0}(\text { tri }-E D O T: X)  \tag{2.4}\\
& -\Delta G^{0}(\text { tri }-E D O T: P T S)-\Delta G^{0}(E M I M: X)
\end{align*}
$$

A more negative value of $\Delta \Delta G_{x}^{0}$ indicates a more favorable ion exchange process (in red, Figure 2.32. When PEDOT:PSS is mixed vigorously with EMIM:TCM ( $\Delta G^{0}=-248 \mathrm{~kJ} / \mathrm{mol}$ ) or EMIM:TCB $\left(\Delta G^{0}=-227 \mathrm{~kJ} / \mathrm{mol}\right)$, hydrophilic pairs of EMIM:PTS $\left(\Delta G^{0}=-317 \mathrm{~kJ} / \mathrm{mol}\right)$ would form spontaneously (black lines with squares, Figure 2.32, leaving behind hydrophobic PEDOT:TCM and PEDOT:TCB $\left(\Delta G^{0}=-200\right.$ and $-180 \mathrm{~kJ} / \mathrm{mol}$ respectively, black lines with triangles, Figure 2.32). Indeed, the ion exchange is favorable in presence of TCM and TCB anions $\left(\Delta \Delta G_{x}^{0} \sim-35 \mathrm{~kJ} / \mathrm{mol}\right)$.

On the other hand, ES and PTS have the same sulfonate group and will have similar binding energy between EMIM $\left(\Delta G^{0}(\right.$ EMIM:ES $)=-306 \mathrm{~kJ} / \mathrm{mol}$ and $\Delta G^{0}($ EMIM:PTS $\left.)=-317 \mathrm{~kJ} / \mathrm{mol}\right)$ and PEDOT $\left(\Delta G^{0}(\right.$ PEDOT:ES $)=-221 \mathrm{~kJ} / \mathrm{mol}, \Delta G^{0}($ PEDOT:PTS $\left.)=-235 \mathrm{~kJ} / \mathrm{mol}\right)$ leading to a negligible ion exchange $\left(\Delta \Delta G_{x}^{0} \sim 2 \mathrm{~kJ} / \mathrm{mol}\right)$. The point-charge-like Cl would prefer to bind to charge-localized EMIM ( $\Delta G_{x}^{0}$ (EMIM:Cl) $\sim-352 \mathrm{~kJ} / \mathrm{mol}$ ) than to charge-dispersed PEDOT ( $\Delta G_{x}^{0}$ (PEDOT: Cl$) \sim-258 \mathrm{~kJ} / \mathrm{mol}$ ) and therefore the ion exchange would not occur easily ( $\Delta \Delta G_{x}^{0} \sim 12$ $\mathrm{kJ} / \mathrm{mol}$ ).

Based on these results, design principles have been proposed for IL anion X in order to sustain high ion exchange. The anion X is required to be hydrophobic, bulky, soft and chargedispersed and a new hypothetical anion heptacyanocyclopentenide (HCCP) that satisfies this criteria has been previously proposed (displayed in left on Figure 2.32 for ion pairs EMIM:HCCP and PEDOT:HCCP). This anion displays the lowest ion pair binding energy when associated with EMIM $\left(\Delta G_{x}^{0}(\right.$ EMIM:HCCP $\left.) \sim-200 \mathrm{~kJ} / \mathrm{mol}\right)$ and the most negative ion exchange energy $\left(\Delta \Delta G_{x}^{0}\right.$ (EMIM:HCCP) $\sim-38 \mathrm{~kJ} / \mathrm{mol}$ ) among the series of considered anions.


Figure 2.32 - Examples of optimized geometries of EMIM:HCCP and PEDOT:HCCP ion pairs by DFT calculations. Gibbs free energy of ions pairs EMIM:X and PEDOT:X ( $\Delta \Delta G_{b}$, in black) and free energy of ion exchange between these ion pairs so that $\Delta \Delta G_{x}=\Delta G_{b}$ (EMIM:PTS) $+\Delta G_{b}$ (tri-EDOT:X) - $\Delta G_{b}$ (tri-EDOT:PTS) - $\Delta G_{b}$ (EMIM:X). The Gibbs free energy of tri-EDOT:PTS and EMIM:PTS ion pairs are respectively $-235 \mathrm{~kJ} / \mathrm{mol}$ and $-317 \mathrm{~kJ} / \mathrm{mol}$. Picture taken from ref. [201] (Journal of the American Chemical Society, 2018, 140, 5375-5384, Figure 2, Copyright 2018 with permission from ACS).

### 2.3 Objectives of the thesis

Based on the above experiments that describe different self-assembly processes, we will design numerical simulations to explore such processes. Considering the length scales and time scales of the phenomena at play, quantum degrees of freedom have to be ignored. Hence, we will rely on a classical description of the atomic systems simulated through molecular dynamics (MD) and Monte Carlo (MC) simulation techniques described in detail in the next chapter.

## Objectives of project 1

The first project will be investigated by using classical MC numerical simulations. We will only consider double strand DNA molecules and we will not focus on DNA superstructures like DNA origami for simplification.

First, the effective force between two DNA molecules and the stability of a hexagonal DNA bundle will be studied in presence of small counterions of different valencies in order to validate the simulation code we have constructed for this project.

Next, we will probe how gold nanoparticles self-assemble on a single DNA molecule. In par-
ticular, we will explore the influence of the nanoparticle charge on the assembly process and the effect of ionic strength on the assembled structures.

Then, instead than performing direct simulations to explore the self-assembly process of several DNA molecules and gold nanopaticles, we will use biased simulations to quantitatively estimate the attraction between two parallel DNA molecules through the calculation of the effective force as a function of their mutual separation and in presence of gold nanoparticles and small ions. Subsequently, we will probe the stability of a hexagonal or square-like DNA bundle in presence of gold nanoparticles and small ions by computing osmotic pressure as a function of the bundle lattice spacing.

## Objectives of project 2

The second project will be investigated with classical MD simulation techniques. First, we will simulate and analyze PEDOT:PSS aqueous systems where the influence of the length of PEDOT and PSS chains on the morphology will be characterized. Then, we will determine quantitatively the spontaneous ion exchange with free energy calculation performed on separate pairs of molecules. Finally, direct (i.e non biased) MD simulations will be performed in order to investigate the dynamic of PEDOT chains self-assembly in PEDOT:PSS mixtures in presence of EMIM: X ( $\mathrm{X}=$ Cl, ES, TCM, TCB and HCCP) IL. Influence of polyelectrolyte lengths and system size on the resulting PEDOT:PSS morphology will be investigated for the whole IL series. Our larger-scale MD simulations along the X anion series will permit to validate the design principles that the X anion must satisfy in order to obtain high ion exchange between PEDOT:PSS and the EMIM:X IL.

## CHAPTER 3

## Simulation Methods

## Contents

3.1 Simulations at different scales ..... 84
3.2 Molecular Monte Carlo Simulations ..... 85
3.2.1 Motivations ..... 85
3.2.2 Monte Carlo simulations in various ensemble ..... 87
3.2.3 Implemented Trial Moves ..... 94
3.2.4 Interactions in the simulation box ..... 97
3.2.5 The Widom insertion method ..... 107
3.2.6 The Monte Carlo simulation package ..... 110
3.3 Molecular dynamics simulations ..... 111
3.3.1 General idea ..... 112
3.3.2 Integration of equations of motion ..... 113
3.3.3 Force Fields ..... 115
3.3.4 Regulation of Pressure and Temperature ..... 117
3.3.5 Free energy calculation ..... 122
3.3.6 Molecular dynamics simulation packages ..... 126
3.3.7 How to perform a simulation with GROMACS? ..... 127

In this chapter, we review the two main computational techniques used to simulate the behavior and properties of the systems studied in this work. First, we introduce several algorithms used in classical Monte Carlo simulations. In a second part, we present the classical molecular dynamics technique.

### 3.1 Simulations at different scales

Numerical simulation involves the representation of a process using a model with parameters and variables properly chosen to reproduce the experimental system. The mathematical representation is based on empirical or theoretical models previously developed. Numerical simulations are used to probe complex phenomena involving large amount of data or lack of analytical solutions. Nowadays, the enormous available computational power allows numerical simulation to become an important complementary or substitute tool for various experiments.


Figure 3.1 - Application ranges of computational simulations for different scales. Picture adapted from ref. [202] (Chemical reviews, 2016, 116, 7898-7936, Figure 1, Copyright 2018 with permission from ACS).

Depending of the degree of reality that we want to model in a numerical simulation, we may consider different approaches depending of length scale and/or timescale of the phenomenon to be explored. The Figure 3.1 summarizes the different numerical approaches one may consider.

Quantum-mechanical approches aim to calculate the electronic structure and energy level but is limited to short length scale (usually a few nm ) and timescale (usually few ps ).

For larger systems with length scales going from nanometer to millimeter and timescale going from nanosecond to second (Figure 3.1), classical all-atom or coarse-grained (CG) molecular simulations are more adapted. CG models are idealized models derived from all-atom models where groups of atoms or molecules are replaced by CG particles. The use of CG models accelerates a simulation in two ways. First, it uses a fewer number of particles, decreasing the CPU cost to
calculate the interaction potentials. Second, the smoother CG potential both shorten the time for the system to reach equilibrium and decrease the energy barriers between local minima.

At a mesoscale level, one would consider evolution of systems with models that describe spatial and temporal variations of order parameters [203] like density, temperature or magnetization with partial derivative equations. Another route to probe evolution of systems at this scale is the kinetic Monte Carlo algorithm which simulates the dynamic from state to state transition with pre-chosen probabilities, suitable to study phenomenon like dislocation mobility, surface growth or surface diffusion [204].

For the works conducted in this thesis, we focus on length scale and timescale accessible to classical all-atom and CG simulations. The detailed molecular models will be presented at the beginning of chapters 4 and 5 .

### 3.2 Molecular Monte Carlo Simulations

Monte Carlo simulation (MC) techniques are a class of computational algorithms using randomness to solve deterministic problems. MC simulations take their origin from the Markov chain Monte Carlo method developed in the 1940's by Stanisław Marcin Ulam, John von Newmann and Nicholas Metropolis whose gave "Monte Carlo" as a name to the method, in reference to the famous place for gambling [205, [206]. Nowadays, Monte Carlo methods are indispensable techniques to solve various problems ranging from optimization problem to behavior of complex systems with many coupled degrees of freedom in various field such as physics, chemistry, biology, economy or social sciences.

### 3.2.1 Motivations

Macroscopic properties of a system of $N$ classical particules at temperature $T$ in a cubic simulation box of volume $V$ can be derived from the partition function $Q$ counting all the states available to the system:

$$
\begin{equation*}
Q(N, V, T)=c \int d \mathbf{p}^{N} d \mathbf{r}^{N} \exp \left[-\beta \mathcal{H}\left(\mathbf{p}^{N}, \mathbf{r}^{N}\right)\right] \tag{3.1}
\end{equation*}
$$

where $\mathbf{r}^{N}$ and $\mathbf{p}^{N}$ represent respectively the coordinates and the impulsions of the $N$ particles and $\beta=\frac{1}{k_{B} T}$ with $k_{B}$ the Boltzmann constant and $T$ the temperature. $\mathcal{H}\left(\mathbf{p}^{N}, \mathbf{r}^{N}\right)$ is the Hamiltonian of the system such that $\mathcal{H}=\mathcal{E}+\mathcal{U}$ where $\mathcal{E}$ is the kinetic energy and $\mathcal{U}$ the potential energy. The partition function is normalized according to $c=\frac{1}{N!h^{3 N}}$.

In the canonical ensemble, the average of an observable $A$ is expressed by statistical mechanics as

$$
\begin{equation*}
\langle A\rangle=\frac{\int d \mathbf{p}^{N} d \mathbf{r}^{N} A\left(\mathbf{p}^{N}, \mathbf{r}^{N}\right) \exp \left[-\beta \mathcal{H}\left(\mathbf{p}^{N}, \mathbf{r}^{N}\right)\right]}{\int d \mathbf{p}^{N} d \mathbf{r}^{N} \exp \left[-\beta \mathcal{H}\left(\mathbf{p}^{N}, \mathbf{r}^{N}\right)\right]} \tag{3.2}
\end{equation*}
$$

Given that $\mathcal{E}$ is a quadratic function of the momenta and that properties calculated does not depend explicitely of momenta $\left(A\left(\mathbf{p}^{N}, \mathbf{r}^{N}\right)=A\left(\mathbf{r}^{N}\right)\right)$, the resulting integration over momenta can be done analytically and equation 3.2 can be written in term of coordinates:

$$
\begin{equation*}
\langle A\rangle=\frac{\int d \mathbf{r}^{N} A\left(\mathbf{r}^{N}\right) \exp \left[-\beta \mathcal{U}\left(\mathbf{r}^{N}\right)\right]}{\int d \mathbf{r}^{N} \exp \left[-\beta \mathcal{U}\left(\mathbf{r}^{N}\right)\right]}=\frac{\int d \mathbf{r}^{N} A\left(\mathbf{r}^{N}\right) \exp \left[-\beta \mathcal{U}\left(\mathbf{r}^{N}\right)\right]}{Z} \tag{3.3}
\end{equation*}
$$

The calculation of the observable $A$ over positions (equation 3.3) should require the calculation of the configurational partition function $Z$, that depend only of coordinates.

The probability to find the system in a state $\left(\mathbf{r}^{N}\right)$ writes:

$$
\begin{equation*}
\mathcal{N}\left(\mathbf{r}^{N}\right)=\frac{\exp \left[-\beta \mathcal{H}\left(\mathbf{r}^{N}\right)\right]}{Z} \tag{3.4}
\end{equation*}
$$

For a very small system of 100 atoms, the corresponding partition function $Z$ already results in a 300-dimensional integral, most of the time not solvable analytically. Consequently, Monte Carlo simulation techniques are introduced.

Remark: All the investigations conducted in this work with Monte Carlo simulation techniques has been done with home-made simulation codes. Indeed, it is sometimes more interesting to write a full software specifically designed for a system one wants to probe, instead of using a third-party simulation package which is less flexible to
manage and/or to adapt.

### 3.2.2 Monte Carlo simulations in various ensemble

A thermodynamical ensemble characterizes the coupling of a physical system with its environment. Some of them are isolated, others are closed but can exchange heat with the surrounding or are open and can also exchange particles. We review the 4 most common thermodynamical ensembles used to perform molecular simulations.

- The microcanonical ensemble (NVE) represents the set of systems isolated from their environment and thus have constant energy.
- The canonical ensemble (NVT) refers to the systems with a constant volume, temperature and number of particles.
- The isothermal-isobaric (NPT) ensemble is widely use to mimic the experimental conditions encountered in real experiments where temperature and pressure are controlled.
- The collection of systems that can exchange particles with the environment at constant volume and temperature belong to the grand canonical ensemble $(\mu \mathrm{VT})$.

Given that experiments under constant $N, V$, and $E$ are very rare, it is not frequent to use the microcanonical ensemble in simulations. It is however important to notice that the microcanonical ensemble is the default ensemble for MD simulations where total energy is constant while the canonical ensemble is the default ensemble for MC simulations.

We review the different mathematical framework and Monte Carlo recipes for the most used thermodynamical ensembles.

### 3.2.2.1 The canonical ensemble

In the canonical ensemble, the configurational partition function $Z$ of a set of $N$ particles in a closed system of volume $V$ at temperature $T$ is defined as

$$
\begin{equation*}
Z=\frac{1}{\lambda^{3 N} N!} \int d \mathbf{r}^{N} \exp \left[-\beta \mathcal{U}\left(\mathbf{r}^{N}\right)\right] \tag{3.5}
\end{equation*}
$$

where $\lambda=\sqrt{\frac{h^{2}}{2 \pi m k_{B} T}}$ is the thermal de Broglie wavelength and $\mathcal{U}\left(\mathbf{r}^{N}\right)$ is the potential between particles.

The average observable $A$ writes:

$$
\begin{equation*}
\langle A\rangle=\frac{\int d \mathbf{r}^{N} A\left(\mathbf{r}^{N}\right) \exp \left[-\beta \mathcal{U}\left(\mathbf{r}^{N}\right)\right]}{Z} \tag{3.6}
\end{equation*}
$$

To compute the observable $A$ from 3.6 while avoiding the difficult estimation of $Z$, Metropolis and co-workers proposed an algorithm particularly suitable for the canonical ensemble [207].

During a simulation, the only known quantity is the Boltzmann factor $\exp \left[-\beta \mathcal{U}(\mathbf{r})^{N}\right]$. In other terms, we have access to the relative but not the absolute probability density, which depends explicitely on the partition function.

Let us consider a system in an old configuration $o$, whose Boltzmann factor is $\exp [-\beta \mathcal{U}(o)]$. The system reaches a new equilibrium state $n$ after a small perturbation, with the corresponding Boltzmann factor $\exp [-\beta \mathcal{U}(n)]$ and the configurational probability to find the system in the state $n$ writes:

$$
\begin{equation*}
\mathcal{N}(n)=\frac{\exp [-\beta \mathcal{U}(n)]}{Z} \tag{3.7}
\end{equation*}
$$

where $\mathcal{U}(n)$ is the potential energy of the system in the state $n$. The Metropolis algorithm aims to decide which state to accept or reject after this trial perturbation.

A system in equilibrium should satisfy the balance condition: the average number of accepted trial configuration from state $o$ to state $n$ must be equal to the average number of accepted trial configuration from state $n$ to state $o$. In practical, a stronger condition that fulfills the balance condition is the detailed balance condition: at equilibrium, the average number of accepted moves from $o$ to other state $n$ exactly cancels the reverse moves. This balanced condition is detailed as following:


Figure 3.2 - Procedure of a canonical MC simulation.

$$
\begin{equation*}
\mathcal{N}(o) \pi(o \rightarrow n)=\mathcal{N}(n) \pi(n \rightarrow o) \tag{3.8}
\end{equation*}
$$

The transition probability can be expressed as a product of the probability of accepting a trial move and the probability of performing a trial move:

$$
\begin{equation*}
\pi(o \rightarrow n)=\alpha(o \rightarrow n) \times \operatorname{acc}(o \rightarrow n) \tag{3.9}
\end{equation*}
$$

In the original Metropolis algorithm, $\alpha(o \rightarrow n)=\alpha(n \rightarrow o)$, so that equation 3.8 leads to the ratio:

$$
\begin{equation*}
\frac{\operatorname{acc}(o \rightarrow n)}{\operatorname{acc}(n \rightarrow o)}=\frac{\mathcal{N}(n)}{\mathcal{N}(o)}=\exp (-\beta[\mathcal{U}(n)-\mathcal{U}(o)]) \tag{3.10}
\end{equation*}
$$

Among many possibilities for $\operatorname{acc}(o \rightarrow n)$ that satisfy equation 3.10. Metropolis et al. chose the
following one:

$$
\operatorname{acc}(o \rightarrow n)= \begin{cases}\exp (-\beta[\mathcal{U}(n)-\mathcal{U}(o)]), & \text { if } \quad \mathcal{U}(n)>\mathcal{U}(o)  \tag{3.11}\\ 1, & \text { if } \quad \mathcal{U}(n) \leq \mathcal{U}(o)\end{cases}
$$

which can be rewritten in a more compact form:

$$
\begin{equation*}
\operatorname{acc}(o \rightarrow n)=\min (1, \exp (-\beta[\mathcal{U}(n)-\mathcal{U}(o)])) \tag{3.12}
\end{equation*}
$$

The Metropolis method allows a correct sampling of the state distribution of a system without computing the partition function.

We implemented the canonical Monte Carlo algorithm in a program written in C++ (Appendix A). We detailed the corresponding algorithm in Figure 3.2

### 3.2.2.2 Isobaric-isothermal ensemble

It is important to review this ensemble where the temperature $T$, the pressure $P$ and the number of particles $N$ remain constant, a situation encountered in many real systems which exchange heat but no particle with the environment.

This technique is particularly adapted to probe systems in the vicinity of a first order transition given that the system is able to transform into a state of minimal Gibbs free energy at constant pressure with fluctuating volume [208]. The isobaric-isothermal ensemble has been used by Woods [209] to simulate a two dimensional hard disk system. McDonald has formalised the method for continuous inter-molecular potentials (Lennard-Jones potential) and we give a description of this method below [210].

The partition function for the NPT ensemble is derived from the NVT ensemble. We consider a system of volume $V_{o}$ with $M$ particles and we divide this system into two subsystems of volume $V_{o}-V$ and $V$, each one containing respectively $M-N$ and $N$ particles.

The situation is described in Figure 3.3 and the partition function of the system is the product of the partition functions of the subsystems:


Figure 3.3 - The subsystem of volume $V_{o}-V$ is a reservoir of volume for the small system $V$.

$$
\begin{equation*}
Q\left(N, M, V, V_{o}, T\right)=\frac{V^{N}\left(V_{o}-V\right)^{M-N}}{\lambda^{3 M} N!(M-N)!} \int d \mathbf{s}^{M-N} \ldots \int d \mathbf{s}^{N} \exp \left[-\beta \mathcal{U}\left(\mathbf{s}^{N}, \mathbf{L}\right)\right] \tag{3.13}
\end{equation*}
$$

where the coordinates are scaled over the cubic box size $\mathbf{s}=\frac{\mathbf{r}}{\mathbf{L}}$.
Let us expand the volume of the outer system to infinity so that its pressure is not modified if the small system changes its volume $V$. In that case, the probability to find a microstate of volume $V$ is given by:

$$
\begin{equation*}
\mathcal{N}_{N, P, T}(V)=\frac{V^{N} \exp (-\beta P V) \int d \mathbf{s}^{N} \exp \left[-\beta \mathcal{U}\left(\mathbf{s}^{N}, \mathbf{L}\right)\right]}{\int_{0}^{V_{o}} d V^{\prime} V^{\prime N} \exp \left(-\beta P V^{\prime}\right) \int d \mathbf{s}^{N} \exp \left[-\beta \mathcal{U}\left(\mathbf{s}^{N}, \mathbf{L}^{\prime}\right)\right]} \tag{3.14}
\end{equation*}
$$

The probability of accepting a trial step after modification of the volume from $V$ to $V^{\prime}=V+\Delta V$ in a Monte Carlo simulation is:

$$
\begin{equation*}
\operatorname{acc}(o \rightarrow n)=\min \left(1, \exp \left(-\beta\left[\mathcal{U}\left(\mathbf{s}^{N}, V^{\prime}\right)-\mathcal{U}\left(\mathbf{s}^{N}, V\right)+P\left(V^{\prime}-V\right)-\frac{(N+1)}{\beta} \ln \left(\frac{V^{\prime}}{V}\right)\right]\right)\right) \tag{3.15}
\end{equation*}
$$

The rescaling of the system is carefully done when changing the volume. For a given molecule, only the center of mass has to be rescaled to preserve the bond distance between each atoms.

### 3.2.2.3 Grand Canonical ensemble

Grand canonical Monte Carlo ensemble is characterized by a constant volume $V$, constant temperature $T$ and chemical potential $\mu$. These conditions correspond to experimental situations when an adsorbent material is in contact with a reservoir of particles, for instance the adsorbent can be liquid water in contact with its gaseous phase. A first implementation of the Grand canonical Monte Carlo method was proposed by Norman and Filinov [211] and has been extended later by other groups.

The partition function of the Grand canonical Monte Carlo ensemble can be derived from the partition function of the canonical (NVT) ensemble in the same manner as the partition function for the isobaric-isothermal ensemble. In that case, it is not the volume but particles which are exchanged with a reservoir.


Figure 3.4 - The subsystem of volume $V_{o}-V$ is a reservoir of particles for the small system $V$.

As illustrated on Figure 3.4 small subsystem of volume $V$ can exchange particles with the biggest subsystem of volume $V_{o}-V$ without changing its chemical potential $\mu$.

Let us call $N$ the total number of particles in the system of volume $V$ and $(M-N)$ the number of particles in the system of volume $V_{o}-V$ so that the total number of particles is $M$. The function partition of such system reads:

$$
\begin{equation*}
Q\left(N, M, V, V_{o}, T\right)=\frac{V^{N}\left(V_{o}-V\right)^{M-N}}{\lambda^{3 M} N!(M-N)!} \int d \mathbf{s}^{M-N} \ldots \int d \mathbf{s}^{N} \exp \left(-\beta \mathcal{U}\left(\mathbf{s}^{N}\right)\right) \tag{3.16}
\end{equation*}
$$

If the two systems can exchange particles, the function partition sums all the possible distributions of the $M$ particles in the two subsystems:

$$
\begin{equation*}
Q\left(M, V, V_{o}, T\right)=\sum_{N=0}^{M} \frac{V^{N}\left(V_{o}-V\right)^{M-N}}{\lambda^{3 M} N!(M-N)!N!} \int d \mathbf{s}^{M-N} \ldots \int d \mathbf{s}^{N} \exp \left(-\beta \mathcal{U}\left(\mathbf{s}^{N}\right)\right) \tag{3.17}
\end{equation*}
$$

where $\mathbf{s}^{N}$ are the scaled coordinates of the system. When the volume of the reservoir $V_{o}-V$ and the number of particles $M$ tend to infinity, we introduce the density $\left(M /\left(V_{o}-V\right)\right) \rightarrow \rho$ linked to the chemical potential $\mu$ of the reservoir by

$$
\begin{equation*}
\mu=k_{B} T \ln \left(\lambda^{3} \rho\right) \tag{3.18}
\end{equation*}
$$

At a finite number of particles $N$ in the system of volume $V$, the partition function 3.17 is simplified in the limit of $(M / N) \rightarrow \infty$ and is expressed as following:

$$
\begin{equation*}
Q(\mu, V, T)=\sum_{N=0}^{M} \frac{\exp (\beta \mu N)}{\lambda^{3 N} N!} \int d \mathbf{s}^{N} \exp \left(-\beta \mathcal{U}(\mathbf{s})^{N}\right) \tag{3.19}
\end{equation*}
$$

We can derive the probability density to find a state with $N$ particles:

$$
\begin{equation*}
\mathcal{N}_{\mu, P, T}(V) \propto \frac{\exp (\beta \mu N) V^{N}}{\lambda^{3 N} N!} \exp \left(-\beta \mathcal{U}\left(\mathbf{s}^{N}\right)\right) \tag{3.20}
\end{equation*}
$$

A Grand canonical Monte Carlo simulation has two types of trial moves at each MC step in case of a system made of spherical particles, one for the displacement of a particle and another one to choose whether to insert or remove a particle from the system.

The acceptance probability for the particle displacement is:

$$
\begin{equation*}
\operatorname{acc}(o \rightarrow n)=\min \left(1, \exp \left(-\beta\left[\mathcal{U}\left(\mathbf{s}^{\prime}\right)^{N}-\mathcal{U}(\mathbf{s})^{N}\right]\right)\right) \tag{3.21}
\end{equation*}
$$

The acceptance probability for the particle insertion is written as:

$$
\begin{equation*}
\operatorname{acc}(N \rightarrow N+1)=\min \left(1, \frac{V}{\lambda^{3}(N+1)} \exp (\beta[\mu-\mathcal{U}(N+1)+\mathcal{U}(N)])\right) \tag{3.22}
\end{equation*}
$$

and the expression for the removal probability is:

$$
\begin{equation*}
\operatorname{acc}(N \rightarrow N-1)=\min \left(1, \frac{\lambda^{3} N}{V} \exp (-\beta[\mu+\mathcal{U}(N-1)-\mathcal{U}(N)])\right) \tag{3.23}
\end{equation*}
$$

We implemented and adapted the grand canonical Monte Carlo algorithm in a case of insertion/deletion of monovalent salt pairs at constant chemical potential represented by spherical beads and we provide some check tests in Appendix B. 3

### 3.2.3 Implemented Trial Moves

In this section, we review the Monte carlo trial moves that we implemented in the Monte Carlo simulation package. It is relatively easy to implement a new trial move in a $\mathrm{C}++$ program provided that the detailed balance condition 3.8 is respected.

The trial moves implemented can either change the position of the center of mass or the orientation of a rigid molecule. We present also a trial move to displace an ensemble of particles defined as a "cluster move".

### 3.2.3.1 Translational move

The translational move involves a displacement of the center of mass of a molecule placed at a center of an imaginary box of side $\Delta$ [208]. The trial displacement is created by choosing random numbers between $-\frac{\Delta}{2}$ and $\frac{\Delta}{2}$ to modify the $x, y$ and $z$ coordinates of the molecule center of mass. The translational move is applied as followed.

1. Pick randomly a molecule in the box.
2. Modify the coordinates of the molecule according to the following rules:

$$
\left\{\begin{array}{l}
x_{i}^{\prime}=x_{i}+\Delta(\text { Rand }-0.5)  \tag{3.24}\\
y_{i}^{\prime}=y_{i}+\Delta(\text { Rand }-0.5) \\
z_{i}^{\prime}=z_{i}+\Delta(\text { Rand }-0.5)
\end{array}\right.
$$

### 3.2.3.2 Rotational move

We need to introduce a rotational move to sample the orientational degree of freedom of a rigid molecule. We decided to use the version of the pivot algorithm presented by Steinhauser [212] and initially proposed by Lal in 1969 [213]. The pivot algorithm was first implemented to overcome the difficult sampling of internal degrees of freedom of a polymer chain. The algorithm is suited to probe the orientational degree of freedom for a rigid molecule around its mass center. If we consider N molecules in a simulation box, the steps to perform a rotational move is described as following:

1. Pick randomly a molecule of index $k \in[1 ; \mathrm{N}]$.
2. Choose randomly a direction uniformly distributed on a sphere of radius 1 associatd with an unit vector $\mathbf{n}$ whose origin coincides with the center of mass of the molecule $k$.

Remark: One need to sample uniformly the polar and azimutal Euler angles defined by $\theta \in[0 ; \pi]$ and $\phi \in[0 ; 2 \pi]$ by using the following relationships:

$$
\begin{cases}\phi & =2 \pi r  \tag{3.25}\\ \cos (\theta) & =2 w-1\end{cases}
$$

where $r$ and $w$ are randomly generated numbers $\in[0 ; 1]$.
3. Rotate the molecule of an angle $\alpha$ around the direction $\mathbf{n}$. The coordinates are transformed according to:

$$
\begin{equation*}
\mathbf{r}^{\prime}=\mathbf{r} \cos \alpha+(1-\cos \alpha)(\mathbf{r} . \mathbf{n}) \mathbf{n}+\sin \alpha(\mathbf{n} \times \mathbf{r}) \tag{3.26}
\end{equation*}
$$

We implemented the above pivot algorithm through a MC scheme in Appendix B. 2 by sampling the configurational degrees of freedom of the Kratky-Porod model. We ensure that the implementation is correct so that the algorithm can be used in the Monte Carlo simulation package.

### 3.2.3.3 Collective move

Collective Monte Carlo moves are developed to sample strongly attractive colloidal, molecular or atomistic systems. This kind of systems often form group of particles (called also "clusters") for which an attempt to displace one particle outside of the cluster results in an unfavorable variation of the interaction energy. Such Monte Carlo step would be rejected with the Metropolis algorithm preventing the system to relax properly and/or efficiently.

A flourishing literature proposed different algorithms to address such issues. The Virtual Move Monte Carlo Scheme (VMMC) proposed by Whitelam and Geissler [214] involves the construction of a cluster from a randomly chosen particle that executes a virtual trial move. All the particles moving in the same fashion have to be included in the cluster by calculating the interaction energy before and after the move. Finally, the positions of all particles in the cluster are updated. Although this algorithm was originally proposed for short-range interactions, Ružička and Allen have extended it to general pairwise interactions [215].

Another type of collective move particularly suited to sample fluid systems is the geometrical cluster algorithm (GCA) developed by Dress and Krauth [216] and generalized later by Liu and Luijten [217] where a molecular configuration is rotated around a pivot. The particles which overlap between the original and rotated configurations belong to the same cluster. Finally, the particles belonging to the defined cluster are exchanged between the original and the rotated configurations.

These algorithms are efficient for dense and complex systems but are not straightforward to implement in a software, especially with several species. We decided thus to use the partially
clothed pivot algorithm developed by Valleau and Gordon [218] which is easy to implement in a software and compatible to the pivot algorithm we presented in section 3.2.3.2

We proposed an adapted version of the algorithm to either rotate or translate a large particle and its surrounding cloud of small particles.

1. Pick randomly a large particle of index $k \in[1 ; \mathrm{N}]$.
2. Let us consider the original configuration $o$ and denote $\mathrm{m}_{o}$ the small particles to be chosen with a probability $\mathrm{p} \leq 1$ and falling within some predetermined volume defined by a certain radius $R$ around the center of mass of the large particle $k$ that has to be rotated with the pivot algorithm (section 3.2.3.2) or translated (section 3.2.3.1).
3. From the proposed move associated with the new state $n$, let us call $\mathrm{m}_{n}$ the total amount of small particles falling in the same volume after the move.
4. The acceptance probability for the step is:

$$
\begin{equation*}
\operatorname{acc}(o \rightarrow n)=\min \left(1,(1-p)^{m_{n}-m_{o}} \exp (-\beta[\mathcal{U}(n)-\mathcal{U}(o)])\right) \tag{3.27}
\end{equation*}
$$

where $\mathcal{U}(n)$ and $\mathcal{U}(o)$ are the energy of the new and former state respectively.

We selected $\mathrm{p}=1$ according to the work of Lobaskin and Linse [219]. Indeed, all moves including a new small particle between initial and final step will be rejected in order to respect the detailed balance condition. Also, in order to reject any rotation which will change the number of particles in the cluster, any choice of $\mathrm{p}<1$ should be avoided for efficiency as there might be overlap between the molecule and the ions not selected for the move inside the considered volume.

### 3.2.4 Interactions in the simulation box

We give a description of the interactions that we implemented for the Monte Carlo simulations. The ensemble of interactions is represented by a set of mathematical functions and parameters called force field, and we will give a complete presentation in section 3.3.3. The physical interactions are described with a minimal model including 3 important features.

- All molecules will be rigid.
- The solvent will be modeled by a dielectric constant $\epsilon$.
- The interactions are described with a combination of pairwise hard core potential (i.e. steric repulsion to prevent particle overlap) and pairwise Coulomb interactions reduced by the dielectric constant of the solvent $\epsilon$ :

$$
V_{i j}(r)= \begin{cases}\infty & r \leq\left(\sigma_{i}+\sigma_{j}\right) / 2  \tag{3.28}\\ \frac{q_{i} q_{j}}{4 \pi \epsilon_{o} \epsilon r} & r>\left(\sigma_{i}+\sigma_{j}\right) / 2\end{cases}
$$

where $\sigma_{i}$ and $\sigma_{j}$ are the particle diameters and r is the interparticle separation. One common numerical trick to compute the interactions between particles are the use of a set of boundary conditions called periodic boundary conditions (PBCs).

### 3.2.4.1 Periodic boundary conditions

The PBCs are chosen to remove (a part of) the boundary effects due to the simulation box in order to reproduce the behavior of macroscopic system. A description of the PBCs in 2 dimensions is presented in Figure 3.5

During a molecular simulation, particle's motion need to be restricted in the original simulation box. For instance, if the green particle depicted in Figure 3.5 escapes from one side of the simulation box in a given direction, it will re-enter in the box from the opposite side. The corresponding algorithm writes for the $x$ direction:

$$
\begin{cases}i f\left(x>L_{x}\right) & x=x-L_{x}  \tag{3.29}\\ i f\left(x<L_{x}\right) & x=x+L_{x} \\ i f\left(x \leq L_{x} \quad \& \& \quad x \geq L_{x}\right) & x=x\end{cases}
$$



Figure 3.5 - Presentation of the PBCs in two dimensions. The original cell contains the red particles with its eight nearest images cells filled with blue particles. The red particle centered on the dashed rectangle interacts with the closest periodic image of other particles depicted with arrows. The green particle in original cell is associated with its direct periodic images.

For a given particle in the simulation cell, the closest interacting neighbors within a centered cell on this particle is implemented with the following recipe. It is straightforward to switch from 2 to 3 dimensions.

- Particles in the computational box have only pairwise interactions. In order to take into account PBCs, the pair separation $R X, R Y$ and $R Z$ along the $x, y, z$ directions are computed for each pair of particle $i$ and $j$ as followed:

$$
\left\{\begin{array}{l}
R X=x_{i}-x_{j}  \tag{3.30}\\
R Y=y_{i}-y_{j} \\
R Z=z_{i}-z_{j}
\end{array}\right.
$$

- We apply the following tranformations (that hold for a cubic, a orthorhombic or a quadratic box) usually called "minimum image convention" [220] on $R X, R Y$ and $R Z$ which select the shortest distance between the particle i and either the particle $j$ or one of its images in the image cells:

$$
\left\{\begin{array}{l}
R X=R X-L_{x} \times \operatorname{round}\left(\frac{R X}{L_{x}}\right)  \tag{3.31}\\
R Y=R Y-L_{y} \times \operatorname{round}\left(\frac{R Y}{L_{y}}\right) \\
R Z=R Z-L_{z} \times \operatorname{round}\left(\frac{R Z}{L_{z}}\right)
\end{array}\right.
$$

where $L_{x}, L_{y}$ and $L_{z}$ are the size of the simulation box sides.

The minimum image convention will be used to calculate the electrostatic interactions in the simulation box.

### 3.2.4.2 Cell lists for pairwise hard core potential

Most of the computational time is spent calculating the interaction energy between particles in the box. Although the calculation of the electrostatic interactions relies on the minimum image convention, there is still a waste of time to evaluate the pairwise hard core potential at a given step. Indeed, when a particle is chosen to be displaced, a "brute force" approach requires to compute the interpenetration of the particle with all other particles in the box. We used the cell lists algorithm [221] that keeps track of the lists of particles in smaller boxes in the simulation box. We briefly present the algorithm given that we used it in the simulation code to evaluate the pairwise hard core potential.

The simulation box is divided into smaller boxes of side $r^{\prime}$. Then, we create lists in which we store the index of each particle contained in each small box. At a given MC step, the lists are modified in the following way (Figure 3.6):

- If the chosen particle (in green) is displaced so that the particle remain in the small simulation box, no update is required for the lists and the interpenetration of the particle with other particles in the small box and in adjacent small boxes (in red) is tested. If there is interpenetration or unfavorable change of energy, the new coordinates of the particle is rejected.
- If the particle is displaced in a new small box, the interpenetration of the particle is tested with the particles inside the new box and adjacent boxes of this new box. If the step is


Figure 3.6 - For this example in 2 dimensions, we divided the simulation box into 9 small boxes of side $r^{\prime}$. Supplemental boxes are added around the original box to take account of the PBCs.
rejected, the lists are not updated. If the step is accepted, we delete the index of the particle in the list of the former small box and add it to the list of the new small box.

The interpenetration is tested for adjacent small boxes given that a particle can be located in a small box but interpenetrates with another particle located in an adjacent small box. We divide the simulation box in 6 small boxes in the $x, y$ and $z$ direction resulting in 216 small boxes.

### 3.2.4.3 Ewald summation method

The Ewald summation method is used to compute the electrostatic interactions in a periodic system and cannot be evaluated between particles with a simple cut-off. In particular, the idea of the Ewald summation algorithm is to split the low converging electrostatic potential into two converging sub-potentials [222].

Let us suppose that the original set composed of point charges is modified by introducing gaussian charges ("clouds") of opposite charge such that the lattice becomes neutral as shown in Figure 3.7 lattice (B). In order to evaluate only the contribution of the point charges (lattice (A)), it is required to add a lattice of compensate diffuse charges of the same sign as the point charges (Figure 3.7 lattice (C)). The electrostatic potential produced by lattice (B) will decay thus faster


Figure 3.7 - (A) The original set of electrostatic charges (in blue) is separated into two sets of charges. (B) Each charge is associated with a diffuse opposite charge (in red). (C) The diffuse opposite charges are withdrawn so that $(A)=(B)+(C)$ in term of electrostatic charge.
and can be computed using a cutoff scheme. This contribution to the electrostatic potential is computed in the $\mathbf{r}$-space, while the contribution of the lattice $(\mathrm{C})$ is calculated in the $\mathbf{k}$-space.

We derive the Ewald summation algorithm to compute the electrostatic interaction in a cubic box. It is relatively straightforward to adapt the algorithm for an orthorhombic or a quadratic box. The gaussian charge distribution to compensate the initial point charges has a width of $\sqrt{\frac{2}{\alpha}}$ such that:

$$
\begin{equation*}
\rho(r)=q\left(\frac{\alpha}{\pi}\right)^{\frac{3}{2}} \exp \left(-\alpha r^{2}\right) \tag{3.32}
\end{equation*}
$$

We can write the gaussian distribution function in the Fourier space for a system of $N$ particles in the cubic box of volume $V$ :

$$
\begin{equation*}
\rho_{k}=\sum_{i=1}^{N} q_{i} \exp \left(-i \mathbf{k} \cdot \mathbf{r}_{i}\right) \tag{3.33}
\end{equation*}
$$

where $\mathbf{k}=\frac{\mathbf{n}}{L}$ with $\mathbf{n}=\left(n_{x}, n_{y}, n_{z}\right)$ denote the lattice vectors in the reciprocal lattice.
The insertion of $\rho_{k}$ in the Poisson equation leads to the electrostatic potential generated for the compensating charge (lattice (C) of Figure 3.7 corresponding to the long range part of the electrostatic potential) and is written as:

$$
\begin{equation*}
\phi^{L R}(r)=\sum_{\mathbf{k} \neq \mathbf{0}} \sum_{i=1}^{N} \frac{q_{i}}{4 \pi^{2} \epsilon_{o} k^{2} V} \exp \left(2 \pi i \mathbf{k} \cdot\left(\mathbf{r}-\mathbf{r}_{i}\right)\right) \exp \left[-\left(\frac{\pi k}{\alpha}\right)^{2}\right] \tag{3.34}
\end{equation*}
$$

The contribution to the vector $\mathbf{k}=\mathbf{0}$ is null since the cell must be neutral:

$$
\begin{equation*}
\sum_{i=1}^{N} q_{i}=0 \tag{3.35}
\end{equation*}
$$

The long range interaction energy is expressed as following:

$$
\begin{equation*}
\mathcal{U}^{L R}=\frac{1}{2 \pi V} \sum_{\mathbf{k} \neq \mathbf{0}} \frac{1}{4 \pi \epsilon_{o} k^{2}} \exp \left[-\left(\frac{\pi k}{\alpha}\right)^{2}\right] \cdot\left|\sum_{i=1}^{N} q_{i} \exp \left(2 \pi i \mathbf{k} \cdot \mathbf{r}_{i}\right)\right|^{2} \tag{3.36}
\end{equation*}
$$

with a supplemental term taking into account of the interaction of the gaussian distribution with itself:

$$
\begin{equation*}
\mathcal{U}^{\text {Self }}=-\frac{\alpha}{\sqrt{\pi}} \sum_{i=1}^{N} q_{i}^{2} \tag{3.37}
\end{equation*}
$$

The short range contribution is calculated up to a cut-off in the real space:

$$
\begin{equation*}
\mathcal{U}^{S R}=\frac{1}{2} \sum_{i \neq j}^{N} \frac{q_{i} q_{j}}{4 \pi \epsilon_{o}} \frac{\operatorname{erfc}\left(\alpha r_{i j}\right)}{r_{i j}} \Theta\left(r_{c u t}-r_{i j}\right) \tag{3.38}
\end{equation*}
$$

The total electrostatic interaction can thus be written as $\mathcal{U}^{E L}=\mathcal{U}^{S R}+\mathcal{U}^{L R}+\mathcal{U}^{\text {Self }}$. In order to use the Ewald method, 3 parameters need to be properly chosen

- The real cutoff parameter $r_{c u t}$,
- the cutoff lattice vector in the reciprocal space $n_{c}$,
- the splitting parameter $\alpha$.

We need to ensure that the short-range contribution is small at the cutoff $r_{c u t}$. In that case the short-range potential varies like:

$$
\begin{equation*}
\mathcal{U}^{S R} \propto \frac{\exp \left(-\alpha^{2} r_{c}^{2}\right)}{\left(\alpha r_{c}\right)^{2}} \tag{3.39}
\end{equation*}
$$

The long range potential also depends of a decaying exponentiel term:

$$
\begin{equation*}
\mathcal{U}^{L R} \propto \frac{\exp \left[-\left(\frac{\pi n_{c}}{\alpha L}\right)^{2}\right]}{\left(\left(\frac{\pi n_{c}}{\alpha L}\right)^{2}\right.} \tag{3.40}
\end{equation*}
$$

We impose that each contribution varies like $\frac{\exp \left(-x^{2}\right)}{x^{2}}$ [208]. Let us denote $s$ the value such that $\frac{\exp \left(-s^{2}\right)}{s^{2}}=\epsilon$. By identifying this with 3.39 and 3.40 we obtain the two following relations:

$$
\left\{\begin{array}{l}
\alpha=\frac{s}{r_{c u t}}  \tag{3.41}\\
n_{c}=\frac{s L \alpha}{\pi}
\end{array}\right.
$$

In our simulations, we have chosen $s=3$ which corresponds to $\epsilon \approx 10^{-5}$ and $r_{c u t}=\frac{\min \left(L_{x}, L_{y}, L_{z}\right)}{2}$. However, it appears that choosing such $r_{c u t}$ is not wise, because the real part of the electrostatic interaction scale as $N^{2}$ where $N$ is the number of charges of the system and most of the computational time is spent to calculate this part of the electrostatic interactions. It would have been more suitable to choose a smaller cutoff $r_{c u t}$ to achieve the optimal scaling of $\mathrm{N}^{\frac{3}{2}}$ [208].

However, we used a trick in order to save computational time in our software (Appendix A) for the calculation of the electrostatic energy by Ewald summation technique with our choice of cutoff. Instead of calculating the electrostatic energy from scratch at each MC step, we used the fact that only an ion, a nanoparticle or eventually a nanoparticle with its cloud of ions are displaced at each MC step while the other moities in the system remain fixed. Let us explain the principle for a system that contains only small ions that can move in the simulation box. The system in a state $(o)$ has an electrostatic energy of $E^{(o)}$. When an ion $i$ is displaced, we calculate the part of electrostatic energy associated with the new position of the ion $i$ (all other ions are fixed) denoted
by $e_{e l, i}^{(n)}$. Given that we store the trajectory of the previous positions in state (o) of all ions, we calculate the electrostatic energy of the old position of the ion $i$ and all other fixed ions in the simulation box, denoted by $e_{e l, i}^{(o)}$. Consequently, the electrostatic energy that corresponds to the new state $(n)$ of the system after the move of ion $i$ writes $E^{(n)}=E^{(o)}-e_{e l, i}^{(o)}+e_{e l, i}^{(n)}$. We can thus apply the MC scheme since we know $E^{(o)}$ and $E^{(n)}$ at each MC step. This trick allows to reduce the expensive calculation of the electrostatic energy at each MC step, especially for the real part for which the calculation scale from $N^{2}$ to $\sim N$ at each step. Hence, For this choice of parameters of $s$ and $r_{c u t}$, the damping parameter $\alpha$ and the number of wave number $n_{c}$ become for a cubic box of side $L$ :

$$
\left\{\begin{array}{l}
\alpha=\frac{6}{L}  \tag{3.42}\\
n_{c}=\frac{18}{\pi} \approx 6
\end{array}\right.
$$

Remark: In addition to the implementation of the Ewald summation technique in a MC simulation code, it is necessary to derive the electrostatic forces from the Ewald energy to perform further analysis.

The electrostatic force acting on particle $i$ from all other particles $j$ in the system (and all their images in the image cells taking into account PBCs) is expressed as:

$$
\begin{equation*}
\mathbf{F}_{i}^{E L}=-\vec{\nabla}_{\mathbf{r}_{i}} \mathcal{U}^{E L}=\mathbf{F}^{S R}+\mathbf{F}^{L R}=-\vec{\nabla}_{\mathbf{r}_{i}} \mathcal{U}^{S R}-\vec{\nabla}_{\mathbf{r}_{i}} \mathcal{U}^{L R} \tag{3.43}
\end{equation*}
$$

We notice that $\mathcal{U}^{\text {Self }}$ does not depend on coordinates and will vanish. The differentiation of the short-range electrostatic energy leads to the short-range electrostatic force:

$$
\begin{equation*}
\mathbf{F}^{S R}=-\vec{\nabla}_{\mathbf{r}_{i}} \mathcal{U}^{S R} \tag{3.44}
\end{equation*}
$$

$$
\begin{gather*}
\mathbf{F}^{S R}=-\vec{\nabla}_{\mathbf{r}_{i}}\left[\frac{1}{2} \sum_{i \neq j}^{N} \frac{q_{i} q_{j}}{4 \pi \epsilon_{o}} \frac{\operatorname{erfc}\left(\alpha r_{i j}\right)}{r_{i j}}\right]  \tag{3.45}\\
\mathbf{F}^{S R}=-\left[\frac{1}{2} \sum_{i \neq j}^{N} \frac{q_{i} q_{j}}{4 \pi \epsilon_{o}}\left[\vec{\nabla}_{\mathbf{r}_{i}}\left(\frac{1}{r_{i j}}\right) \operatorname{erfc}\left(\alpha r_{i j}\right)+\frac{1}{r_{i j}} \vec{\nabla}_{\mathbf{r}_{i}}\left(\operatorname{erfc}\left(\alpha r_{i j}\right)\right)\right]\right]  \tag{3.46}\\
\left.\mathbf{F}^{S R}=-\left[\frac{1}{2} \sum_{i \neq j}^{N} \frac{q_{i} q_{j}}{4 \pi \epsilon_{o}}\left[-\left(\frac{\mathbf{r}_{i j}}{r_{i j}^{3}}\right) \operatorname{erfc}\left(\alpha r_{i j}\right)+\frac{1}{r_{i j}}\left(-\frac{2 \alpha}{\sqrt{\pi}} \exp \left(-\left(\alpha r_{i j}^{2}\right)\right) \frac{\mathbf{r}_{i j}}{r_{i j}}\right)\right)\right]\right]  \tag{3.47}\\
\mathbf{F}^{S R}=q_{i} \sum_{i>j}^{N} q_{j}\left[\frac{2 \alpha}{\sqrt{\pi}} \exp \left(-\left(\alpha r_{i j}\right)^{2}\right)+\frac{\operatorname{erfc}\left(\alpha r_{i j}\right)}{r_{i j}}\right] \frac{\mathbf{r}_{i j}}{r_{i j}^{2}} \tag{3.48}
\end{gather*}
$$

In the same manner, let us derive the long-range part of the electrostatic interaction.
In order to be clearer, let us call:

$$
\begin{equation*}
\sum_{i=1}^{N} q_{i} \exp \left(2 \pi i \mathbf{k} \cdot \mathbf{r}_{i}\right)=\rho(\mathbf{k}) \tag{3.49}
\end{equation*}
$$

If we apply the gradient operator to $\rho(\mathbf{k})$, we obtain the useful relation:

$$
\begin{equation*}
\vec{\nabla}_{\mathbf{r}_{i}} \rho(\mathbf{k})=\delta_{i j} q_{j} 2 \pi i \mathbf{k} \exp \left(2 \pi i \mathbf{k} \cdot \mathbf{r}_{j}\right) \tag{3.50}
\end{equation*}
$$

Finally the long-range electrostatic interaction is calculating as following:

$$
\begin{equation*}
\mathbf{F}^{L R}=-\vec{\nabla}_{\mathbf{r}_{i}} \mathcal{U}^{L R} \tag{3.51}
\end{equation*}
$$

$$
\begin{gather*}
\mathbf{F}^{L R}=-\vec{\nabla}_{\mathbf{r}_{i}}\left[\frac{1}{2 \pi V} \sum_{\mathbf{k} \neq \mathbf{0}} \frac{1}{4 \pi \epsilon_{o} k^{2}} \exp \left[-\left(\frac{\pi k}{\alpha}\right)^{2}\right] \cdot|\rho(\mathbf{k})|^{2}\right]  \tag{3.52}\\
\mathbf{F}^{L R}=-\left[\frac{1}{2 \pi V} \sum_{\mathbf{k} \neq \mathbf{0}} \frac{1}{4 \pi \epsilon_{o} k^{2}} \exp \left[-\left(\frac{\pi k}{\alpha}\right)^{2}\right]\right] \cdot \vec{\nabla}_{\mathbf{r}_{i}}|\rho(\mathbf{k})|^{2}  \tag{3.53}\\
\mathbf{F}^{L R}=-\left[\frac{1}{2 \pi V} \sum_{\mathbf{k} \neq \mathbf{0}} \frac{1}{4 \pi \epsilon_{o} k^{2}} \exp \left[-\left(\frac{\pi k}{\alpha}\right)^{2}\right]\right] \cdot\left[2 \pi i q_{i} \mathbf{k}\left[\sum_{j=1}^{N} 2 \pi i q_{j} \sin \left(\mathbf{k} \cdot \mathbf{r}_{i j}\right)\right]\right]  \tag{3.54}\\
\mathbf{F}^{L R}=\frac{q_{i}}{4 \pi \epsilon_{o}} \sum_{j=1}^{N}\left[\frac{2 q_{j}}{V} \sum_{\mathbf{k} \neq \mathbf{0}}\left[\frac{\exp \left(-\left(\frac{\pi k}{\alpha}\right)^{2}\right)}{k^{2}} \sin \left(2 \pi \mathbf{k} \cdot \mathbf{r}_{i j}\right) \mathbf{k}\right]\right] \tag{3.55}
\end{gather*}
$$

The Ewald summation method is only suitable to compute the electrostatic interactions for relatively small systems of at most several hundreds charges, which will be our situation for the systems we will later consider. There exists some algorithms to improve the method like the multiple timestep method [309] that consists of evaluating less frequently the long range part of the electrostatic interaction.

Other techniques to compute the electrostatic interactions exists, like the particle mesh Ewald (PME) method [223, 224] that interpolates the charges on a grid and use the fast Fourier transform to calculate the long range part of the electrostatic interaction in order to achieve a scaling of the order of $\mathcal{O}(N \log N)$.

### 3.2.5 The Widom insertion method

Condition of coexistence between different phases requires the use of the chemical potential $\mu$. The Widom insertion technique [225] is a simple method to measure the chemical potential $\mu$ of a species in a system.

Let us consider the classical partition function of a system of $N$ particles in a box of volume $V=L^{3}$ written as

$$
\begin{equation*}
Q(N, V, T)=\frac{1}{\lambda^{3 N} N!} \int d \mathbf{r}^{N} \exp \left[-\beta \mathcal{U}\left(\mathbf{r}^{N}\right)\right]=\frac{V^{N}}{\lambda^{3 N} N!} \int_{0}^{1} \ldots \int_{0}^{1} d \mathbf{s}^{N} \exp \left[-\beta \mathcal{U}\left(\mathbf{s}^{N}\right)\right] \tag{3.56}
\end{equation*}
$$

where the scaled coordinates $\mathbf{s}^{N}=\left(\frac{\mathbf{r}}{L}\right)^{N}$ have been introduced. We can derive the Helmholtz free energy [208] of the system:

$$
\begin{align*}
F(N, V, T)= & -k_{B} T \ln Q  \tag{3.57}\\
& =  \tag{3.58}\\
& -k_{B} T \ln \frac{V^{N}}{\lambda^{3 N} N!}-k_{B} T \ln \left(\int d \mathbf{s}^{N}\right) \exp \left[-\beta \mathcal{U}\left(\mathbf{s}^{N}\right)\right]  \tag{3.59}\\
& =
\end{align*}
$$

The Helmholtz energy has a contribution from the ideal gaz $F_{i d}(N, V, T)$ and an excess part $F_{e x}(N, V, T)$. For a system containing a large number of particles $N$, the chemical potential describes the variation of free energy with a change in the number of particles. If a particle is added the chemical potential is expressed as

$$
\begin{equation*}
\mu=-k_{B} T \ln \frac{Q(N+1, V, T)}{Q(N, V, T)} \tag{3.60}
\end{equation*}
$$

By inserting the explicit form of $Q(N, V, T)$ in the chemical potential, we find a contribution from the perfect gas and an excess part:

$$
\begin{array}{cc}
\mu=-k_{B} T \ln \frac{V}{\lambda^{3}(N+1)}- & k_{B} T \ln \frac{\int d \mathbf{s}^{N+1} \exp \left[-\beta \mathcal{U}\left(\mathbf{s}^{N+1}\right)\right]}{\int d \mathbf{s}^{N} \exp \left[-\beta \mathcal{U}\left(\mathbf{s}^{N}\right)\right]} \\
\mu= & \mu_{i d}+\mu_{e x} \tag{3.62}
\end{array}
$$

The contribution from the perfect gas to the chemical potential $\mu_{i d}$ is calculated analytically. We present hereafter the computation of the excess part of the chemical potential $\mu_{e x}$. We can
rewrite the excess potential $\mu_{e x}$ in the following way:

$$
\begin{gather*}
\mu_{e x}=-k_{B} T \ln \frac{\int d \mathbf{s}^{N+1} \exp \left[-\beta \mathcal{U}\left(\mathbf{s}^{N+1}\right)\right]}{\int d \mathbf{s}^{N} \exp \left[-\beta \mathcal{U}\left(\mathbf{s}^{N}\right)\right]}  \tag{3.63}\\
\mu_{e x}=-k_{B} T \ln \frac{\int d \mathbf{s}^{N+1} \exp \left[-\beta \mathcal{U}\left(\mathbf{s}^{N}\right)\right] \exp [-\beta \Delta \mathcal{U}]}{\int d \mathbf{s}^{N} \exp \left[-\beta \mathcal{U}\left(\mathbf{s}^{N}\right)\right]}  \tag{3.64}\\
\mu_{e x}=-k_{B} T \ln \int d \mathbf{s}^{N+1} \mathcal{N}(N, V, T) \exp [-\beta \Delta \mathcal{U}]  \tag{3.65}\\
\mu_{e x}=-k_{B} T \ln \int d \mathbf{s}^{N+1}\langle\exp [-\beta \Delta \mathcal{U}]\rangle_{N} \tag{3.66}
\end{gather*}
$$

where the term $\langle\exp [-\beta \Delta \mathcal{U}]\rangle_{N}$ is the canonical average over configurations for the system with N particles and $\Delta \mathcal{U}$ is the difference between the potential energy of the system with $\mathrm{N}+1$ particles and the system with N particles.

The excess potential is thus easy to compute through a standard Metropolis Monte Carlo scheme. Practically, we carry out a NVT Monte Carlo simulation on a system of $N$ particles and at regular step we insert randomly a supplementary particle. The total energy of the system with the new particle is calculated and the inserted particle is removed. Hence, the difference of energy before and after the particle insertion is $\Delta \mathcal{U}$. By averaging $\exp [-\beta \Delta \mathcal{U}]$ over generated trial positions, we obtain the excess potential $\mu_{e x}$. We test the Widom insertion technique scheme in Appendix B. 3

Remark: The Widom insertion technique will be used to determine the chemical potential to be use for the grand canonical Monte Carlo simulation.

### 3.2.6 The Monte Carlo simulation package

In this section, we give a brief presentation of the Monte Carlo simulation package we have developed. Although the main core is the Monte Carlo program, we implemented additional programs to build the initial configuration of the systems, to associate the radius and charges for beads and to visualize and analyze the output trajectory. The description of the workflow between the different programs is presented in Figure 3.8


Figure 3.8 - Description of the software we developed and implemented to perform a complete Monte Carlo simulation, from the setup of an initial state to the analysis of an output trajectory.

We first construct the system using the "Initialization_system.cpp" program which creates a file "System_construct.txt" with the name of each moities (ions, atoms and molecules), their coordinates and the connectivity between each atom.

The "Building_topology.cpp" program reads inputs from the "System_construct.txt" file and
reads force field parameters (bead radius and charges) from "Radius_particles.txt" and "Electrostatic_charges.txt". The created output files contain the list of the atoms and their initial coordinates in "System_geometry.txt" and a list of each atom with their corresponding charge and radius in "System_topology.txt".

The core of the simulation package performing a Monte Carlo simulation is implemented in "Main_MC.cpp", the master program used to initialize the classes and parameters for the three other .cpp files.

The "Move__MC.cpp" is the program where the different Monte Carlo moves and the associated acceptance probabilities are implemented. The "Potential_energy_calculation.cpp" file computes the change of energy after a proposed MC move. Finally, "the Cell_lists.cpp" program manages the cell lists through the whole simulation.

Once we get the trajectory of the system during the MC simulation ("Traj_output.txt"), we can either visualize it with "Visualization_trajectory.cpp" or analyse it with "Analysis_trajectory. cpp". Most of the time, the trajectory files will have a size of the order of Gigaoctets, we thus we implemented the "Read_trajectory.cpp" program to read through such huge files. The full simulation package written in $C++$ is provided in Appendix A

### 3.3 Molecular dynamics simulations

Molecular dynamics simulation (MD) is a computational technique to study the dynamical and equilibrium properties of many-particle systems evolving according to classical or quantum mechanical laws. MD techniques became widely used to probe kinetics, structural or thermodynamical properties of physical or chemical systems. MD simulation inspired development of new sub-branches of techniques, such as coarse-grained (CG), ab initio or hybrid quantum mechanical approach. All investigations conducted in the thesis with MD simulations neglect the electronic degrees of freedom. Therefore, we will only review the classical MD simulation method describing the motion of nucleus. Unlike Monte Carlo simulation, MD simulation enables to monitor also the dynamical evolution of a system, which is important for different physical reactions.

### 3.3.1 General idea

In the same manner as molecular MC simulations, typical molecular dynamics simulations consist of large number of particles, whose average properties cannot be determined analytically. Hence, the basic flow to perform a MD simulation can be described as following.

- Prepare the system with the desired molecules and well-defined interaction potentials.
- At each iteration, move all the particles by solving Newton's equation of motion for sufficiently short timestep. The procedure is repeated until the system equilibriates.
- Compute the equilibirum properties of the system by using laws of statisctical physics to establish relationship between the state of the system and its macroscopic properties.

We detailed a procedure to perfom a MD simulation in Figure 3.9


Figure 3.9 - Procedure of an MD simulation.

### 3.3.2 Integration of equations of motion

A MD simulation requires to update particle coordinates, velocities and forces at each timestep by integrating Newton's equation of motion. A suitable integration algorithm should conserve energy and momentum, be computationally efficient and permit the use of a relatively long timestep for integration. The Verlet algorithm [226] satisfy these criteria and is implemented in many MD simulation packages. The Verlet algorithm is derived from the Taylor expansion of the equation of motion of a particle:

$$
\begin{align*}
& \mathbf{r}(t+\Delta t)=\mathbf{r}(t)+\mathbf{v}(t) \Delta t+\frac{\mathbf{f}(t)}{2 m} \Delta t^{2}+\frac{\Delta t^{3}}{3!} \dddot{\mathbf{r}}+O\left(\Delta t^{4}\right)  \tag{3.67}\\
& \mathbf{r}(t-\Delta t)=\mathbf{r}(t)-\mathbf{v}(t) \Delta t+\frac{\mathbf{f}(t)}{2 m} \Delta t^{2}-\frac{\Delta t^{3}}{3!} \dddot{\mathbf{r}}+O\left(\Delta t^{4}\right) \tag{3.68}
\end{align*}
$$

where $r, v, f$ and $\Delta t$ refer to position, velocity, force and timestep, respectively. A combination of these two equations leads to the Verlet algorithm for the position and velocity:

$$
\begin{cases}\mathbf{r}(t+\Delta t) & =2 \mathbf{r}(t)-\mathbf{r}(t-\Delta t)+\frac{\mathbf{f}(t)}{m} \Delta t^{2}+O\left(\Delta t^{4}\right)  \tag{3.69}\\ \mathbf{v}(t) & =\frac{\mathbf{r}(t+\Delta t)-\mathbf{r}(t-\Delta t)}{2 t}\end{cases}
$$

Subsequent integration scheme equivalent to the Verlet algorithm has been proposed. One of the most common is the leapfrog algorithm [221] which equations are given by:

$$
\begin{cases}\mathbf{r}(t+\Delta t) & =\mathbf{r}(t)+\mathbf{v}\left(t+\frac{\Delta t}{2}\right) \Delta t  \tag{3.71}\\ \mathbf{v}\left(t+\frac{\Delta t}{2}\right) & =\mathbf{v}\left(t-\frac{\Delta t}{2}\right)+\frac{\mathbf{f}(t)}{m} \Delta t\end{cases}
$$

Although the leapfrog produces the same trajectory as the Verlet algorithm, coordinates and velocities are not defined at the same timestep. Hence, kinetic and potential energies are calculated at different timesteps which make calculation of energy awkward and unsatisfactory. Another algorithm similar to the Verlet scheme and which does not have this drawback is the velocity-Verlet algorithm [227] which writes as following. It only uses a simple Taylor expansion of coordinates and velocities:

$$
\begin{cases}\mathbf{r}(t+\Delta t) & =\mathbf{r}(t)+\mathbf{v}(t) \Delta t+\frac{\mathbf{f}(t)}{2 m} \Delta t^{2}  \tag{3.73}\\ \mathbf{v}(t+\Delta t) & =\mathbf{v}(t)+\frac{\mathbf{f}(t+\Delta t)+\mathbf{f}(t)}{2 m} \Delta t\end{cases}
$$

The velocity-Verlet algorithm has the advantage to provide coordinates and velocities of particles at the same timestep. This algorithm only relies on quantities calculated at a previous step making it suitable for molecular simulations.

However, it is important to mention that for sufficient long time, the generated trajectory from an MD simulation may diverge from the true trajectory. Any integration algorithm cannot guarantee an accurate trajectory. In reality, a MD trajectory depends on the initial conditions and infinitesimally closed initial conditions ultimately lead to divergent trajectories denoting Lyapunov instability [208].

### 3.3.3 Force Fields

Forces fields are a set of parameters and mathematical functions that describe interactions between particless which aim to reproduce experimental macroscopic properties through a classical representation. Two types of interactions are described by a force field. The bonded interactions gather interactions existing within a molecule including the chemical bonds, bond angles and bond dihedrals. The derivation of the bonded interactions rely on a combination of experimental results (geometries of particles from X-ray diffraction) and density functional theory (DFT) calculations. The non-bonded interactions include the Coulomb potential and various functional forms describing the van der Waals interaction like the Lennard-Jones (LJ) or the Weeks-Chandler-Anderson (WCA) potentials. The parameters are still very often derived in order to reproduce some experimental data such as the density.

## Remark: In section 3.2.4, we already presented a simplified force field used for completely rigid molecules with our Monte Carlo simulations. We hereafter present a more versatile force field within the better-suited framework of MD simulation.

When a MD simulation is performed, it is important to choose the appropriate force field adapted to the class of molecules constituting the system from which macroscopic properties need to be evaluated. For example, the AMBER ff99SB force field has been developed for biological molecules and the MARTINI coarse-grained force field has been develop for large-scale simulations. In the following investigation performed by using MD simulations, we will use the OPLS-AA (Optimized Potentials for Liquid Simulations - All Atoms) force field adapted to simulate liquid systems.

### 3.3.3.1 Bonded interactions

The bonded interactions aimed to represent the molecular interactions between atoms within the same molecule. More precisely, the bond interaction simulates the vibration of a covalent bond, the angular interaction describes the angular stretching existing between two bonds and the proper and improper dihedrals interactions control the rotation and planarity of certain groups of atoms.

Although different mathematical representations of these interactions exist, we summarize briefly the most common implemented in molecular simulations packages.

The bond interaction and the angular interaction are usually modeled by quadratic functions that prevent bond breaking:

$$
\begin{equation*}
U_{i j}^{\text {bond }}=\frac{k_{b o n d}}{2}\left(\mathbf{r}_{i j}-\mathbf{r}_{e q}\right)^{2} \tag{3.75}
\end{equation*}
$$

$$
\begin{equation*}
U_{i j k}^{\text {angular }}=\frac{k_{\theta}}{2}\left(\theta_{i j k}-\theta_{e q}\right)^{2} \tag{3.76}
\end{equation*}
$$

where $k_{\text {bond }}$ and $k_{\theta}$ are harmonic spring constants of the bonded and angular interaction. The variable $r_{e q}$ represent the equilibrium bond length and $\theta_{e q}$ the equilibrium angle between two bonds.

The proper dihedral angle used to describe the molecular conformations is defined between 4 subsequent atoms (rotations around a bond) and has a more complicated expression:

$$
\begin{equation*}
U_{i j k l}^{\text {dihedral }}=\sum_{n=0}^{5} C_{n}\left(\cos \left(\psi_{i j k l}\right)\right)^{n} \tag{3.77}
\end{equation*}
$$

where $\psi_{i j k l}=\phi_{i j k l}-\pi$. Finally, improper dihedral is introduced to maintain planarity of molecules containing for example rings of atoms and to maintain geometry of tetrahedral group $\mathrm{CH}_{3}$. Its mathematical expression writes with an equilibrium angle $\phi_{s}$ :

$$
\begin{equation*}
U_{i j k l}^{\text {improper }}=k_{\phi}\left(1+\cos \left(2 \phi_{i j k l}-\phi_{s}\right)\right) \tag{3.78}
\end{equation*}
$$

The different bonded interactions are summarized in Figure 3.10
A)
B)




Figure 3.10 - Schematic of the bonded interactions. (A) Bond interaction. (B) Angular interaction. (C) Proper dihedral interaction. (D) Improper dihedral interaction.

### 3.3.3.2 Non-bonded interactions

The non-bonded interactions are divided into Coulomb and van der Waals interactions. We already discussed the treatment of the Coulomb interaction in section 3.2.4.3 The van der Waals interactions for OPLS-AA force field are represented by using the Lennard-Jones potential:

$$
\begin{equation*}
U_{i j}^{L J}=4 \epsilon_{i j}\left[\left(\frac{\sigma_{i j}}{r_{i j}}\right)^{12}-\left(\frac{\sigma_{i j}}{r_{i j}}\right)^{6}\right] \tag{3.79}
\end{equation*}
$$

where the Lorentz-Berthelot mixing rules $\epsilon_{i j}=\sqrt{\epsilon_{i} \epsilon_{j}}$ and $\sigma_{i j}=\left(\sigma_{i}+\sigma_{j}\right) / 2$ has been used.

### 3.3.4 Regulation of Pressure and Temperature

The default thermodynamical ensemble to perform a MD simulation is the microcanonical ensemble (NVE) given that forces derive from potentials which means that the total energy of the system is conserved if the number of particles and the volume of the system does not change.

Real experiments are performed under controlled temperature and/or pressure and we must add supplementary numerical ingredients to transform the NVE ensemble into NVT or NPT ensemble.

Along this line, thermostat and barostat algorithms are introduced.

### 3.3.4.1 Thermostat

The temperature is dependent of the velocities from a classical point of view through the equipartition theorem [228]. In other term, we must constraint the velocities at constant temperature. Different algorithms have been proposed in this way and we mention here only the thermostats we used in our simulations: the Berendsen thermostat and the Nosé-Hoover thermostat.

The Berendsen thermostat [229] involves a coupling with an external bath of temperature $T_{0}$. Let us recall the derivation of the temperature correction at each timestep.

The equation of motion that describes the coupling of a particle of mass $\mathrm{m}_{i}$ with a bath of temperature $T_{0}$ would read in one dimension:

$$
\begin{equation*}
m_{i} \dot{v}_{i}=F_{i}-m_{i} \gamma_{i} v_{i}+R_{i}(t) \tag{3.80}
\end{equation*}
$$

where $F_{i}$ is the conservative force exerted by the other particles, $-m_{i} \gamma_{i} v_{i}$ is the damping force with $\gamma_{i}$ refering to the strength of coupling with the bath. The third term is the random uncorrelated force which verify the following correlation function:

$$
\begin{equation*}
\left\langle R_{i}(t) R_{j}(t+\tau)\right\rangle=2 m_{i} \gamma_{i} k T_{0} \delta(\tau) \tag{3.81}
\end{equation*}
$$

The differential equation 3.80 describes the motion of a particle interacting with other particles which constituting a perfect gas of temperature $T_{0}$.

The next step is to determine how the temperature of the system evolves with the coupling to the heat bath. The rate of change of kinetic energy is given by the following expression:

$$
\begin{equation*}
\frac{d E_{k}}{d t}=\lim _{\Delta t \rightarrow 0}\left(\sum_{i=1}^{3 N} \frac{\frac{1}{2} m_{i} v_{i}^{2}(t+\Delta t)-\frac{1}{2} m_{i} v_{i}^{2}(t)}{\Delta t}\right) \tag{3.82}
\end{equation*}
$$

with $N$ the number of particles in the system. We express $v_{i}(t+\Delta t)$ by using the differential equation 3.80 such that

$$
\begin{equation*}
v_{i}(t+\Delta t)=v_{i}(t+)+\frac{1}{m_{i}} \int_{t}^{t+\Delta t}\left[F_{i}\left(t^{\prime}\right)-m_{i} \gamma_{i} v_{i}\left(t^{\prime}\right)+R_{i}\left(t^{\prime}\right)\right] d t^{\prime} \tag{3.83}
\end{equation*}
$$

By integrating equation 3.81 we obtain:

$$
\begin{equation*}
\sum_{i=1}^{3 N} \int_{t}^{t+\Delta t} d t^{\prime} \int_{t}^{t+\Delta t} d t^{\prime \prime}\left[R_{i}\left(t^{\prime}\right) R_{i}\left(t^{\prime \prime}\right)\right]=6 N m \gamma k T_{0} \Delta t \tag{3.84}
\end{equation*}
$$

If we combine equation 3.83 and equation 3.84 in equation 3.82 , we arrive at the following expression:

$$
\begin{equation*}
\frac{d E_{k}}{d t}=\sum_{i=1}^{3 N} v_{i} F_{i}+2 \gamma\left(\frac{3 N k}{2} T_{0}-E_{k}\right) \tag{3.85}
\end{equation*}
$$

The first term on the right hand side (RHS) of equation 3.85 is the derivative of the potential energy of the system. The other term in the RHS arises from the bath with which the system is coupled and can be rewritten in term of temperature:

$$
\begin{equation*}
\frac{d T}{d t}=2 \gamma\left(T_{0}-T\right) \tag{3.86}
\end{equation*}
$$

where we identified the coupling parameter to be $\tau_{T}=\frac{1}{2 \gamma}$. We can replace the stochastic term in equation 3.80 which leads to:

$$
\begin{equation*}
m_{i} \dot{v}_{i}=F_{i}+m_{i} \gamma\left(\frac{T_{0}}{T}-1\right) v_{i} \tag{3.87}
\end{equation*}
$$

At every MD timestep, the above equation rescales the velocities from $v$ to $\lambda v$, with $\lambda$ given by:

$$
\begin{equation*}
\lambda=1+\frac{\Delta t}{2 \tau_{T}}\left(\frac{T_{0}}{T}-1\right) \tag{3.88}
\end{equation*}
$$

If $\tau_{T} \rightarrow \infty, \lambda \approx 1$ and the Berendsen thermostat term vanishes and the run is sampling the microcanonical ensemble. The Berendsen algorithm is only suitable to equilibrate a system because it relaxes quickly to the desired temperature with small fluctuations. However, it is not
able to generate states in the canonical ensemble and it should therefore be avoided for the data production stage, when the system is equilibrated and after this stage when equilibrium quantities are calculated.

Once the system has reached equilibrium, it is more suitable to use the Nosé-Hoover thermostat in order to probe the canonical ensemble. The idea of this algorithm proposed at first by Nosé [230] and improved by Hoover [231] is to add extra contributions in the hamiltonian with coordinates, velocities and mass as well as a generalized variable to describe the coupling with the heat bath. The derivation of the Nosé-Hoover thermostat is complicated and more details can be found in [208].

### 3.3.4.2 Barostat

In a molecular simulation, the pressure is calculating by using the virial theorem:

$$
\begin{equation*}
P=\frac{2}{3 V}\left(E_{k}+\frac{1}{2} \sum_{i<j} \mathbf{r}_{i j} \mathbf{F}_{i j}\right) \tag{3.89}
\end{equation*}
$$

A molecular dynamics performed under NPT ensemble can be achieved by regulating the volume. In the simple case, the Berendsen algorithm can be extended to the control of the pressure by scaling the coordinates of particles and box vectors every MD timestep. An extra term is added to take into account the rescaling of the coordinates so that the equation of motion becomes for one direction:

$$
\begin{equation*}
\dot{x}=v+\alpha x \tag{3.90}
\end{equation*}
$$

The rate of change of the volume is thus given by:

$$
\begin{equation*}
\dot{V}=3 \alpha V \tag{3.91}
\end{equation*}
$$

In the same manner as for the Berendsen thermostat, the scaling variation of the pressure for
the Berendsen barostat is:

$$
\begin{equation*}
\frac{d P}{d t}=\frac{P_{0}-P}{\tau_{P}} \tag{3.92}
\end{equation*}
$$

The change of the volume can be expressed with the isothermal compressibility combined with equation 3.91

$$
\begin{equation*}
\frac{d P}{d t}=-\frac{1}{\beta_{T} V} \frac{d V}{d t}=-\frac{3 \alpha}{\beta_{T}} \tag{3.93}
\end{equation*}
$$

where $\beta_{T}$ is the isothermal compressibility. We identify the expression of $\alpha$ from equation 3.92 and 3.93 so that the rescaling velocity becomes:

$$
\begin{equation*}
\dot{x}=v-\frac{\beta_{T}\left(P_{0}-P\right)}{3 \tau_{P}} x \tag{3.94}
\end{equation*}
$$

The above equation represents a scaling of the coordinate $x$ to $\mu \mathrm{x}$ as well as a scaling of the box size $L$ to $\mu L$ with $\mu$ having the following expression:

$$
\begin{equation*}
\mu=1-\frac{\beta_{T} \Delta t}{3 \tau_{P}}\left(P_{0}-P\right) \tag{3.95}
\end{equation*}
$$

Altough the Berendsen barostat is adapted to equilibriate pressure in a system, it does not yield the exact NPT ensemble. We decided instead to use the Parrinello-Rahman extended ensemble algorithm [232] where some additional degrees of freedom are added in the Lagrangian of the system, like what is done for the Nosé-Hoover thermostat. In addition, the Parrinello-Rahman algorithm can handle anisotropic box deformation and pressure coupling in non-orthorombic box, and is thus widely used for MD simulations.

### 3.3.5 Free energy calculation

Free energies drive important molecular processes like aggregation reaction and are most of the time challenging to compute in a simulation. The free energy links the internal energy of the system $U$, (the sum of the kinetic and potential energy of all the particles in the system) the entropy $S$ and the temperature $T$ through the free energy state function:

$$
\begin{equation*}
F=U-T S \tag{3.96}
\end{equation*}
$$

Hence, we would like to restrain the free energy calculation along some reaction coordinate that characterizes the change of the system.


Figure 3.11 - The PMF along the $z$ direction would give the variation of free energy of the system associated with the binding of the guest molecule (in blue) to the host (in red). A) At initial stage, the host binds to the guest molecule. B) Unbinding stage.

The reaction coordinate can be:

- an inter or intramolecular distance,
- a bond or a torsion angle,
- the volume or the temperature of the system...

The calculation of free energy along a reaction coordinate $\xi$ is usually referred as the potential of mean force (PMF) noted $F(\xi)$. As illustred in Figure 3.11, let us consider a guest molecule and a host molecule that can bind to each other. The guest and host molecules can be either biomolecules or charged ions, methane molecules and so on.

We present an example of free energy profile $F(z)$ along the reaction coordinate $z$ in Figure 3.12 that resembles of a PMF describing the binding of two methane molecules in water [233].

The profile of the PMF (Figure 3.12 ) can give us some important information:


Figure 3.12 - Free energy profile $F(z)$ along the reaction coordinate $z$ for a guest-host (example: methane-methane) system. The "bump" characterizes the energy barrier that has been observed for PMF extracted from simulations that study the binding of two methane molecules or the binding of oppositively charged ions in water [234]. Such energy barrier has an entropic origin and represents the energy cost to reorganize the solvent around the host-guest system to promote the binding [233].

- The preferred equilibrium distance at which the host will bind to the guest molecule, i.e the $z$ value corresponding to the minimum of the free energy (binding energy).
- The free energy change upon binding, more precisely the difference in free energy between the minimum binding free energy and the free energy at large value of $z$ (unbinding state).

An important point to keep in mind is that the free energy profile represents more than a measure of the potential energy between the host and guest particle. It can include an entropy contribution induced by the solvent which is also contained in the PMF. More precisely, the PMF can include entropic contributions coming from the translational, rotational degrees of freedom of the host and guest molecules as well as from the reorganization of solvent around the molecules at the binding stage.

One of the most widely used technique to compute the PMF is the umbrella Sampling technique developed by Torrie and Valleau [235, 236]. The goal of the umbrella sampling algorithm is to provide a good estimation of the PMF along a chosen reaction coordinate. In other term, the quantity to be calculated is $F(z)$ :

$$
\begin{equation*}
F(z)=-k_{B} T \ln \mathcal{O}(z) \tag{3.97}
\end{equation*}
$$

where $\mathcal{O}(z)$ is the state distribution of the system along the coordinate $z$.
Without a proper sampling along the $z$ coordinate, some region would be poorly sampled. It is the case at large value of $z$ or region separated by a large barrier of potential energy, resulting in a poor estimation of $F(z)$.

To reach better sampling, we construct a biased distribution that allow us to sample all regions along the coordinate $z$. The umbrella sampling procedure is detailed in few points.

- Perform $K$ simulations of index $k$ of the same system where the only difference is the position $z$ of the guest molecule along the reaction coordinate.
- In each simulation, we constrain the system to sample a given portion of the coordinate $z$, around a centered $z_{k}$. The constraint can be added to the interaction energy through a supplementary term $\eta_{k}(z)$ to energetically penalize the system from visiting far positions $z$ from $z_{k}$. The biased interaction potential is expressed as:

$$
\begin{equation*}
\mathcal{U}_{k}^{b}\left(\mathbf{r}^{N}\right)=\mathcal{U}_{k}\left(\mathbf{r}^{N}\right)+\eta_{k}(z) \tag{3.98}
\end{equation*}
$$

- For each simulation a different target value $z_{k}$ is chosen to span the range of interest along the coordinate $z$.
- We store in a histogram the biased distribution $\mathcal{O}^{b}(z)$ for each simulation. Since it is a continuous variable, one has to discretize the coordinate $z$ into bins.
- We finally unweight the biased distributions and stitch them together to retrieve the unbiased free energy function $F(z)$. Hence, it is important that the biased distributions overlap to retrieve a continuous free energy profile.

We need to explicit the form of the biased potential $\eta_{k}(z)$. The most common choice is the harmonic potential centered at $z_{k}$ :

$$
\begin{equation*}
\eta_{k}(z)=\frac{C}{2}\left(z-z_{k}\right)^{2} \tag{3.99}
\end{equation*}
$$

The value of the force constant $C$ has to be properly chosen. A too small value results in insufficent biasing of the potential. Conversely, a too high value of $C$ will provide a too narrow distribution with poor overlap between the biased distributions, making impossible (or at least very time consuming) the determination of the free energy profile.

One can determine the unbiased distribution from the biased one:

$$
\begin{equation*}
\mathcal{O}\left(\mathbf{r}^{N}\right) \propto \mathcal{O}^{b}\left(\mathbf{r}^{N}\right) \exp \left(-\beta \eta_{k}(z)\right) \tag{3.100}
\end{equation*}
$$

If the distribution is integrated over all coordinates expect $z$, one gets:

$$
\begin{equation*}
\mathcal{O}(z) \propto \mathcal{O}^{b}(z) \exp \left(-\beta \eta_{k}(z)\right) \tag{3.101}
\end{equation*}
$$

If we take the algorithm of equation 3.101, we can in principle determined the free energy function for each simulation $k$ :

$$
\begin{equation*}
F(z)=-k_{B} T \ln \mathcal{O}^{b}(z)+k_{B} T \eta_{k}(z)+\text { const } \tag{3.102}
\end{equation*}
$$

The const variable must be adjusted for each simulation $k$ so that the profile of $F(z)$ can be properly reconstructed. An illustration of this principle is highlighted in Figure 3.13

Instead of shifting the constant to obtain good overlap, a more robust approach to derive the proper free energy function from the biased distribution is the weighted histogram analysis method (WHAM) [235, 237. The WHAM histogram analysis method is a set of dependent equations to be solved iteratively:


Figure 3.13 - A) From the distribution $\mathcal{O}\left(\mathbf{r}^{N}\right)$, we aim at reconstructing the profile of $F(z)$. B) The constant appearing in equation 3.102 needs to be properly chosen to reproduce the correct free energy function.

$$
\left\{\begin{array}{l}
\mathcal{O}(z)=\frac{\sum_{k=1}^{K} c_{k}(z)}{\sum_{k=1}^{K}\left(N_{k} \exp \left[\beta\left(F_{k}(z)-\mathcal{U}_{k}^{b}(z)\right)\right]\right)}  \tag{3.103}\\
F_{k}(z)=-\frac{1}{\beta} \ln \left(\sum_{z_{b i n s}} \mathcal{O}(z) \exp \left[-\beta \mathcal{U}_{k}^{b}(z)\right]\right)
\end{array}\right.
$$

where $K$ is the number of simulations, $c_{k}(z)$ is the number of counts in histrogram bins associated with $z$ and $N_{k}$ is the sum of $c_{k}(z)$ for one simulation. The inverse temperature is denoted by $\beta=\frac{1}{k_{B} T}$ and the biased potential is $\mathcal{U}_{k}^{b}(z)$. Finally the distribution of states along the reaction coordinate $\mathcal{O}(z)$ and the free energy function $F(z)$ are determined iteratively.

Alan Grossfield proposed a software package [238] to solve equations 3.103 and 3.104 but most of the molecular modeling software products contain some routines to solve the WHAM equation.

### 3.3.6 Molecular dynamics simulation packages

Even though it is possible to implement a home-made MD software for specific models, forcefield or specialized algorithms, it will be probably done at the expense of performance if the investigated systems are big/complex or if the code is not parallelized or well optimized.

Fortunately, most of the MD simulations are performed using existing simulation packages. The most famous are either free like GROMACS [239], NAMD [240] and LAMMPS [241] or are commercial like CHARMM [242] and AMBER [243].

The simulation packages must be chosen properly depending of the systems and interactions one wants to simulate. For instance, AMBER (Assisted Model Building with Energy Refinement) and CHARMM (Chemistry at Harvard Macromolecular Mechanics) are the mostly used simulation packages for biomolecular systems. These packages and their associated force fields have been developed since the early stage of MD simulation

NAMD (Nanoscale Molecular Dynamics) package developed by the theoretical biophysics group at University of Illinois at Urbana-Champaign, has been designed for efficient parallel computation either with CPU or GPU processors [244]. LAMMPS (Large Scale Atomic/Molecular Massively Parallel Simulator) is another efficient MD simulation package which can simulate not only biomolecules but inorganics molecules or metals.

The simulation package we used to perform our MD simulations is GROMACS (GROningen Machine for Chemical Simulations). It was first designed for simulations of proteins and has been extended for simulations of organic molecules like polymers or membranes. The optimized algorithms and processor optimizations in GROMACS result in 3-10 faster simulations compared to other programs. GROMACS includes several force-field such as GROMOS [245], CHARMM [246], AMBER [247] and OPLS-AA [248] allowing to simulate various types of atomistic systems. We use OPLS-AA force field since all our simulations will be focused on systems made of relatively small organic molecules in an aqueous environment.

### 3.3.7 How to perform a simulation with GROMACS?

A simulation procedure under GROMACS is controlled by specific command lines. The set of commands to be used in the proper order are displayed in Figure 3.14 First, we need to generate the initial structure of the system usually in the PDB (Protein Data Bank) format. Some softwares like Material Studio (commercial) or Avogadro (free) are suitable to prepare the initial files. The command $g m x$ pdb2gmx converts the pdb file to be readable by GROMACS under the .gro extension. The pdb2gmx command proposes also several force field and water models. The user can also choose his own force field parameters with the option -ff forcefield, where forcefield is a file containing the force-field parameters to be used for the simulation. The generated "system.gro" file contains the list of the atoms with their coordinates and "topol.top" file contains the force field used for each atom with their connectivity

The command gmx editconf generates the shape and the size of the simulation box (orthorhombic, monoclinic, octahedral or dodecahedral).

The solvation of the system is done with the gmx solvate command which prevents also overlap between solvent-solvent or solvent-solute molecules. The "topol.top" is also properly updated to include the solvent molecules.

The gmx insert-molecules command inserts the desired number of molecules in the box at random positions by replacing the solvent molecules. It is thus possible to insert small ions or more complicated molecules.

At this stage, we constructed a complete molecular system ("system_mol.gro") but an energy minimization needs to be carry out to release the constraints that appeared from the construction stage. Three algorithms are implemented in GROMACS to perform the minimization depending on the expected optimization quality: steepest descent, conjugate gradient, and L-BFGS [249].

The gmx mdrun command is the main command of GROMACS to perform various tasks: minimize the energy, perform a MD simulation or perform a particle test insertion. In general, the system is equilibrated through a short NVT followed by a NPT simulation. Then, a longer NPT simulation is peformed to collect the trajectory data (positions, velocity and forces) in .trr file format for further analysis.


Figure 3.14 - General procedure to perform a simulation using GROMACS.

## CHAPTER 4

## Self-assembly of DNA mediated by cations or Nanoparticles

## Contents

4.1 Choice of molecular model ..... 133
4.1 .1 Molecular model of DNA ..... 133
4.1.2 Molecular model of cationic gold nanoparticles and multivalent counterions 135
4.2 Attraction and compaction of DNA molecules mediated by multi- ..... 39
4.2.1 Adsorption of salt on a DNA molecule in presence of counterions ..... 139
4.2.2 Effective force between a pair of DNA molecules ..... 145
4.2.3 Osmotic Pressure calculation in hexagonal DNA bundle ..... 152
4.3 Self-Assembly of DNA mediated by nanoparticles ..... 159
4.3.1 Aggregation of nanoparticles on a single fixed DNA ..... 159
4.3.2 Effect of counterions on the adsorption of AuNP on DNA molecule: PMF ..... 174
4.3.3 Redissolution of AuNPs from a single fixed DNA with excess of salt ..... 178
4.3.4 Effective force between a pair of DNA ..... 182
4.3.5 Osmotic pressure calculation in hexagonal and square DNA phases ..... 189

In living organisms (i.e. in vivo), DNA is stored in a highly compacted phase governed by different compounds that bind, bend and assemble on DNA (chapter 2, section 2.1.1. In eukaryotes, DNA has a typical length of several centimeters (e.g. $\sim 12$ millions of b.p. for Saccharomyces cerevisiae, an unicellular budding yeast) to the meter (e.g. $\sim 2.9$ billions of b.p. for humans) and is compacted in the nucleus with histones which are proteins rich in basic amino acids such as arginine and lysine [250]. Moreover, small ligands like inorganic ions and polyamines may also
induce DNA compaction in prokaryotes within the small volume of the viral capsid or bacterial nucleoid. DNA condensation can be perform in vitro by inducing an attractive interaction between DNA molecules. There are two manners to induce such attraction. First, it has been shown that an excess of neutral polymers in presence of monovalent cations $\mathrm{Na}^{+}$can create entropic random collisions on DNA molecules which induce compaction [88. It is possible to favor DNA attraction in a second way through multivalent cationic species (compaction agents) [76]. However, the morphology of resulting aggregates depends on the solution properties, the charge and the structure of the compaction agents [107] as well as the DNA length (chapter 2] section 2.1.1]. The DNA molecule is negatively charged and cannot exist in solution without its counterions. Usually, physiological solutions contain various type of ions like divalent metal ions or spatially extended ions like polyamines such as spermidine ${ }^{+3}$ or spermine ${ }^{+4}$ that can be found in viruses [251].

DNA-DNA attraction can be explained through correlations between multivalent counterions. The role of interionic correlations leading to attraction has been first understood by Kirkwood and Shumaker [92]. The correlations induced by multivalent cations adsorbed on DNA create patterns of opposite charge creating a short range attraction between charge-like DNA 93]. Also, bridging of DNA between extended compaction agent like protamines [8] or nanoparticules can also induce DNA-DNA attraction [71].

Understanding DNA condensation might lead to the design of new strategies in the field of nanotechnology to control the shape, size and arrangement of nano-objects [146]. Bottom-up strategies rely on self-assembling building blocks like DNA to design new materials [252]. Consequently, such approach motivates researches that aim to assemble cationic gold nanoparticles (AuNPs) on DNA molecules [143, 7] or on DNA scaffolds like DNA origami [253] (chapter 2] section 2.1].

The chapter contains two parts and all the systems are studied with the Monte Carlo simulation package we designed (section 3.2) and implemented (Appendix A). The first part focuses on DNA interaction with small counterions. In particular, attraction of a DNA pair with multivalent cations and stability of a DNA hexagonal bundle will be explored. Although such calculations has been previously performed in the literature, [254, 13, 16, 255, 256] it represents a necessary step to study more complicated systems. In the second part, we probe the interaction of AuNPs on a single fixed DNA. Then, the effect of salt on the desorption of the AuNPs from a single DNA is investigated.

Finally, the attraction of a DNA pair in presence of AuNPs as well as the mechanical stability of a DNA bundle condensed with AuNPs will be explored.

### 4.1 Choice of molecular model

First, we need to use a proper model for DNA, AuNPs and ions. DNA is a complex molecule where each base pair contains around 70 atoms and AuNPs involve often hundreds of atoms. It is unlikely to model with such degree of realism since the computation of interactions will be very time consuming. Moreover the time-scale of the phenomena we want to investigate is much larger than what can be investigated with an all-atom model. Instead, we construct a rigid coarse-grained (CG) model for AuNP and use a rigid CG model for the DNA molecule. Even though a rigid DNA is not completely realistic - except maybe for very short DNA fragments - because DNA molecule has a persistence length of 50 nm (equivalent to 150 base pairs), it is a reasonable first approach for further realistic models.

### 4.1.1 Molecular model of DNA

The B-DNA molecule (the most common conformation of DNA in living cells) is represented by a double helix where each base contains one sugar group and one negative phosphate group. Several models of DNA have been proposed with various degree of realism in order to reproduce the helical distribution of negative charges. Figure 4.1 presents three CG rigid models often used to represent the DNA molecule.

- The cylindrical model (CM) shown in A) and developed by Kornyshev and Leikin [257] contains a cylindrical core of diameter $D=20.0 \AA$ and two strings of beads of diameter $d_{p}=1.0 \AA$ carrying a monovalent charge $q_{p}=-|e|$ to mimic the negative phosphate groups.
- A more refined model is the extended cylinder model (ECM) shown in B) and developed by Lyubartsev and Nordenskiöld [258] where the DNA molecule has a smaller cylindrical core than the CM model with a diameter of $D=17.8 \AA$ but a bigger phosphate group of radius diameter $d_{p}=4.2 \AA$.
- The Montoro-Abascal model represented in C) is a more detailed coarse-grained model of


Figure 4.1 - Representation of different CG DNA models. A) Cylindrical model (CM). B) Extended cylindrical model (ECM). C) Montoro-Abascal model (MAM). "M" design the major groove and " $\mathbf{m}$ " the minor groove. Picture adapted from ref. [13] (Physical Review E, 2003, 68, 061903, Figure 1, Copyright 2003 with permission from APS).

DNA [259]. In this model, the groove structure is better reproduced by adding a supplemental neutral bead corresponding to the sugar group, between the cylindrical core and the phosphate bead. The cylindrical core has a diameter of $D=7.8 \AA$ and the phophate and sugar beads a diameter $d_{p}=4.2 \AA$. Their coordinates along the cylindrical core obey the following relationships:

$$
\begin{array}{rlrl}
\rho_{i}^{p, S} & = & 8.9 \AA \\
\rho_{i}^{q, S} & = & 5.9 \AA \\
\phi_{i}^{S} & = & \phi_{0}^{S}+36^{\circ} \times i \\
z_{i}^{S} & = & z_{0}^{S}+3.40 \times i & \AA \tag{4.4}
\end{array}
$$

where $S=1,2$ refers to the nucleic acid strand and $i=0, \ldots, 9$ represents a full DNA turn. the numbers $\rho_{i}^{p, S}$ and $\rho_{i}^{q, S}$ are respectively the radial position of the phosphates and sugars from the center of the cylinder in $\AA, \phi_{i}^{S}$ is the angular position of a base (phosphate + sugar) in degrees and $z_{i}^{S}$ is the position of a base along the cylinder in $\AA$. For $S=1$ (first strand), $\phi_{0}^{S}$ and $z_{0}^{S}$ are both zero but for $S=2$ (second strand) $\phi_{0}^{S}=154^{\circ}$ and $z_{0}^{S}=0.78 \AA$.

Hence, the MAM describes the B-DNA where the pitch is $34.0 \AA$ per helix containing ten base pairs. Due to the overlapping between the phosphates, sugars and the cylinder, the MAM reproduces a more grooved DNA profile with a deeper cavity in the minor groove and we decided thus to use the MAM for all the Monte Carlo simulations performed in this chapter.

### 4.1.2 Molecular model of cationic gold nanoparticles and multivalent counterions

## Molecular model of cationic gold nanoparticles

The literature that describes the synthesis and applications of functionalized gold nanoparticles is very large, and we provided a short review about these topics in section 2.1.2 From a practical point of view, a typical functionalized AuNP is composed of a gold core on which are attached ligands of various nature depending on the synthesis protocal of the AuNP. In particular, we are interested in modeling AuNPs with a gold core functionalized with thiol ligands terminated with a positive ammonium group. A proper CG model of cationic gold nanoparticles (AuNP) is more difficult to establish than DNA due to the various possible shape, size and charge density of synthesized AuNPs [260]. However, from a numerical point of view, several CG models have been proposed to model AuNPs.

For instance, a detailed CG model with a gold core decorated with flexible ligands has been proposed by Lin and coworkers [261] and further improved by Gupta and Rai [262] in the context of interaction between a lipid bilayer and a single AuNP. Unfortunately, such CG model contains too much atoms and it is unlikely that such a detailed model could be use in simulations involving dozens of AuNP, which will be the case in our studies.


Figure 4.2 - Construction of the AuNP CG model.

On the other side, minimal CG model has been proposed by Komarov et al. [263] to study the aggregation of AuNPs on a single fix DNA at the center of a cubic box. Such model contains only one bead to model the gold core with one overlapping small bead at its surface carrying a charge $+|e|$ representing the positive ammonium thiol ligand chain. At a larger scale, Chiappini and co-workers [264] studied the aggregation of mixture of AuNPs and rigid, rod-like fd-viruses. At this scale, the AuNPs were modeled as a rigid sphere interacting with fd-viruses through a square-like short-range attractive potential.

We aim to construct a suitable CG model that reproduces properly the charged ligands and gold core of the AuNP without sacrificing too much computation time by managing a too large amount of beads and/ or charges in our systems.

Consequently, we described in Figure 4.2 the AuNP model used subsequently in our MC simulations. It is possible to synthesize small AuNPs with a narrow gold core distribution of average
diameter $\approx 14 \AA$ that contain typically $\sim 100$ gold atoms [265]. Consequently, the gold core will be modeled as a bead of diameter of $14 \AA$. In order to avoid steric hindrance that can prevent a proper equilibration of our systems, we decided to consider short ligands compared to the size of the gold core. Each ligand is composed of four overlapping beads of diameter of $3 \AA$ with a bead center - bead center distance of $2 \AA$. The overall AuNP has a diameter of $23 \AA$ if the ligands are attached to the gold core so that the ligand beads interpenetrates the surface of the core at their center. It is important to notice that each AuNP will carry a number of charges equal to the number of ligands. The monovalent charge carried by the ligand (positive ammonium group) is represented as a blue bead in frame B) of Figure 4.2

In order to study the influence of the AuNP charge distribution, we consider different variants of our model with different number and disposition of the ligands on the gold core such that AuNP will carry either 6,12 or 30 ligands for a charge density of $\approx 0.97,1.94$ and $4.87 \mathrm{C} / \mathrm{nm}^{2}$ respectively.


Gold nanoparticle with 6 ligands "6-AuNP"


Gold nanoparticle with 12 ligands "12-AuNP"


Gold nanoparticle with 30 ligands "30-AuNP"

Figure 4.3 - CG model of AuNP for different charge densities.

The Figure 4.3 presents the three different types of AuNPs with their ligands disposed on the gold core. The way to homogeneously distribute the ligands on the CG gold surface for the different models of AuNP is done in the following way.

- For the AuNP model with 6 ligands ( 6 -AuNP), positions are chosen according to the following Euler angles:

$$
\begin{array}{ccc}
\theta=0 & , & \phi=0 \\
\theta=\frac{\pi}{2} & , & \phi=\frac{\pi}{4} \times i \\
\theta=\pi & i=(0,1,2,3)  \tag{4.7}\\
\theta= & \phi=0
\end{array}
$$

- For the 12-AuNP model, the ligands are placed on the gold core according to:

$$
\begin{array}{cc}
\theta=\frac{\pi}{6} \\
\theta=\frac{\pi}{2} \\
\theta=\frac{5 \pi}{6} & , \tag{4.10}
\end{array} \quad \phi=\frac{2 \pi}{3} \times i \quad i=(0,1,2)
$$

- Finally, the rules to distribute the ligands on the 30 -AuNP model write as:

$$
\begin{array}{cc}
\theta=0 & \phi=0 \\
\theta=\frac{\pi}{6} & ,
\end{array}
$$

Even though our coarse-grained model is minimalistic, we tried to reproduce at best the structure of a typical AuNP. The main drawback of our AuNP CG model is the rather low charge density compared to synthesized gold nanoparticles which can have a higher charge densities. For
instance, Warner and Hutchison [] used AuNPs of diameter around $18 \AA$ carrying around 100 ligands for a charge density of $\approx 9|\mathrm{e}| / \mathrm{nm}^{2}$ or Wang and Murray used AuNPs of $44 \AA$ carrying around 910 ligands for a charge density of $15|\mathrm{e}| / \mathrm{nm}^{2}$.

## Molecular model of multivalent counterions

Monovalent or divalent ions will be assimilated as beads of a given radius. Furthermore we will also assimilate extended ions like spermidine ${ }^{3+}$ or spermine ${ }^{4+}$ as beads for simplification. The bead diameter is an important parameter for coarse-grained modeling because it influences the intensity of electrostatic interaction and thus the evolution of the system.

### 4.2 Attraction and compaction of DNA molecules mediated by multivalent cations

Given that interactions of DNA with multivalent cations and by extension DNA-DNA attraction induced by multivalent cations as well as bundle stability have been widely studied over the last decades, we decided to reproduce some results of the literature using our home-made Monte Carlo simulation package. This task has two main objectives. First, it represents a suitable control test for the simulation package to detect huge mistakes. Then, we want to compare the effect of the nature of the compaction agent (i.e. small ions vs. AuNP) on the formation and subsequent stability of an organized assembly of DNA molecules.

### 4.2.1 Adsorption of salt on a DNA molecule in presence of counterions

It is known that the charge distribution of salt on the DNA influences the interaction force and has consequences on the assembly formation of DNA caused by mutual attraction [266].

In this section, we reproduced some of charge adsorption patterns derived by Allahyarov and co-workers [13] in the case of the MAM DNA. In the case of added monovalent salt, the density of salt cations and salt anions is calculated in the minor groove, major groove and on the strand. Although the interacting potential used by Allahyarov and co-workers is the same as those used in our MC simulation, (a combination of electrostatics and steric interactions, section 3.2.4 they
perform MD simulations for equilibration and data collection.

### 4.2.1.1 Methods

In a cubic simulation box of side $L=102 \AA$, the DNA molecule composed of 3 helical pitches of $34 \AA$ each carrying $N_{p}=3 \times 20=60$ phosphate is placed at the center of the $x O y$ plane and parallel to the $z$ axis. Periodic boundary conditions are used mimicking an infinite DNA molecule. In order to neutralize the DNA phosphates, the system contains $N_{c}=120 / q_{c}$ counterions of valency $\mathrm{q}_{c}$ and of concentration $C_{c}=N_{c} / V^{\prime}$ where $V^{\prime}$ is the volume of the system substracted from the volume of the DNA (i.e. the volume accessible to the ions). We insert also $N=N_{+}=N_{-}$pairs of monovalent salt ions of diamater $d_{+/-}=3 \AA$ for a concentration $C_{s}=N_{+} / V^{\prime}$. The salt concentration varies from 0.1 to $1.61 \mathrm{~mol} / \mathrm{l}$, the latter case corresponding to 2000 added salt ions. Figure 4.4 presents a typical snapshot of the simulation. The system is held at 298 K and each run is performed through a NVT canonical MC scheme with a maximal ion displacement in a cubic box of size $0.10 \times L$ during each MC step. The DNA remains frozen during the simulation. We equilibriate the system during $5 \times 10^{3} \mathrm{MC}$ cycles, then gather statistics to calculate canonical averages during times $10^{5}$ MC cycles.

Remark: A Monte Carlo cycle involves $N$ trial moves of every free particles in the simulation box. It is equivalent to a standard MD step.

The key quantity to be calculated is the ion density in the minor and major groove as well as on the strand of phosphates. As written in ref. [13], the average charge density is defined as following:

$$
\begin{equation*}
\rho^{j}(\mathbf{r})=\left\langle\sum_{i=1}^{N_{j}} \delta\left(\mathbf{r}-\mathbf{r}_{i}^{j}\right)\right\rangle \tag{4.18}
\end{equation*}
$$

where for the present calculations, possible values for $j$ are + , - respectively for the salt cations and the salt anions. The total number of ions of species $j$ is defined by $N_{j}$ and $\mathbf{r}_{i}^{j}$ represents the position in space of the $i^{\text {th }}$ particle of species $j$.

The canonical average $\langle\ldots\rangle$ is calculated through a canonical NVT Monte Carlo simulation by adapting the equation 3.3.


Figure 4.4 - Snapshot of the simulation box. The DNA is composed by the black hard cylinder, yellow beads are the sugar groups and the red beads are the DNA phosphate groups. Free ions denoting salt are either green ( + ) or grey ( - ) and DNA counterions (c) are blue beads.

$$
\begin{align*}
\langle\mathcal{A}\rangle= & \frac{1}{Z}\left[\prod_{k=1}^{N_{c}} \int d^{3} \mathbf{r}_{k}^{c}\right]\left[\prod_{m=1}^{N_{+}} \int d^{3} \mathbf{r}_{m}^{+}\right]\left[\prod_{n=1}^{N_{-}} \int d^{3} \mathbf{r}_{n}^{-}\right] \\
& \times \mathcal{A} \exp \left(-\beta \sum_{j=c,+,-}\left[\sum_{j=c, p,+,-} V_{i j}\right]\right) \tag{4.19}
\end{align*}
$$

where $\beta=1 / k_{B} T$ is the inverse thermal energy and $V_{i j}$ is the interaction potential defined in section 3.2.4 The pairwise hard core potential excludes ion-ion as well as ion-DNA from interpenetration.

The density calculation of ions has to be properly defined in space. The width of the condensation shell is $2 \AA$ which corresponds to the Stern layer for the DNA molecule [13]. The three regions of interest are presented as dashed areas of width $\xi=3.4 \AA$ in Figure 4.5 and are located along the DNA helix in the grooves and on the phosphate strand. The volume of each region is bound
radially from the DNA surface by a distance of $\delta=\left(2+d_{+,-} / 2\right) \AA$ where $d_{+,-}$is the diameter of the free ions. Hence, we calculated the charge distribution in the dashed regions around the DNA. We will separate the calculation of charge density for salt cations and anions and consider an angular-resolved density profiles along the grooves and the phosphate strand, whose plot are called "panoramic view" in ref. [13]. We will follow the notation of Allahyarov and co-workers for the panoramic number densities of absorbed ions salt $\rho^{(j)}$ of species $j \in\{+,-\}$.


Figure 4.5 - The three regions of interest (dashed patterns) for which the density calculations are performed. Picture adapted from ref. [13] (Physical Review E, 2003, 68, 061903, Figure 3, Copyright 2003 with permission from APS).

### 4.2.1.2 Results

We calculated the panoramic angular distribution of salt cations for a salt concentration $C_{s}=$ $0.1 \mathrm{~mol} / \mathrm{l}$ in Figure. 4.6, left picture.

The cation density is well-structured in the minor groove and on the strand in contrary to the major groove case. The population density of cations is two times higher in the minor groove than on the strand, as noticed by Allahyarov and co-workers (Figure 4.6 right). In the MAM model of DNA we are using in the simulations, more cations bind in the DNA groove than on the strand compared to the cylindrical model and the enhanced cylindrical model (see in ref. [13). On the


Figure 4.6 - Left) Panoramic view of condensed salt cation density on the different DNA surface regions for $C_{s}=0.1 \mathrm{~mol} / \mathrm{l}$. The density unit is $\times 1000$ the real cation density for better clarity. The red line represents the cation distribution in the minor groove, the black line the cation distribution on the strand and blue line the cation distribution in the major groove. Right) Same quantities calculated by Allahyarov and co-workers, adapted from ref. [13] (Physical Review E, 2003, 68, 061903, Figure 6, Copyright 2003 with permission from APS).
one hand, it is arguing that a relocation of cations from the strand to the grooves brings an entropy gain for the salt ions. On the other hand, the preferred relocation in the minor groove may arise from the stronger electrostatic interactions between cations and phosphate groups contrary to the major groove where the phosphate are further away from each other resulting in lower electrostatic interactions with cations.

The dependence of adsorption pattern of adsorbed cations and anions with the salt concentration is plotted in Figure 4.7 The salt concentration is increased from $C_{s}=0.2 \mathrm{~mol} / \mathrm{l}$ to $C_{s}=1.61$ $\mathrm{mol} / \mathrm{l}$. As mentioned in ref. [13], the cations still adsorb mainly in the minor groove but the anions condensate more in the major groove when the salt concentration is increased. However, the total charge in the major groove calculated by the difference between the cation and anion densities in the major groove (denoted respectively by the dotted blue line and the full blue line in picture 4.7 ) is almost independent of the salt concentration, as show by the double arrow in Figure 4.7 The structure of the major groove and the electric field created by the phosphate strands preserve the charge whatever the salt concentration is by regulating the population of cations and anions. As the salt concentration increases, the charge in the minor groove is positive and increases (See the red full and the red dashed lines of Figure 4.7). Another important observation consists of the constant difference of densities of cations adsorbed in the minor groove and in the major groove
(single arrow in Figure 4.7). This trend shows that even at high salt concentration, the cations adsorbed more in the minor groove than in the major groove.


Figure 4.7 - Panoramic view of condensed salt cation and anion densities on the different DNA surface regions for concentrations of $C_{s}=0.2 \mathrm{~mol} / \mathrm{l}$ (left), $C_{s}=0.71 \mathrm{~mol} / \mathrm{l}$ (middle) and $C_{s}=$ $1.61 \mathrm{~mol} / \mathrm{l}$ (right). The density unit is $\times 1000$ the real density for better clarity. The red dashed line is the cation $(j="+")$ density in the minor groove. The blue dashed line refers to the cation density in the major groove. Conversely, the full blue line represents the anion density ( $j="-"$ ) in the major groove while the full red line corresponds to the anion density in the minor groove. The single arrow indicates that the difference between the density of cation adsorbed in the minor groove and the density of cation adsorbed in the major groove is constant. The double arrows display the constancy of charge in the major groove for added salt concentration.

We performed also supplemental calculations of ion number densities for multivalent counterions or multivalent salt ions. More precisely, we probe the effect of the counterion charge on the distribution of monovalent and divalent salt cations in the DNA grooves (Figure 4.8. When the charge of the counterion is increased, a larger fraction of counterion is adsorbed on the DNA so that the Coulomb correlations between the counterions and the phosphate groups are higher, [13] resulting in a lower amount of monovalent salt ions adsorbed in the DNA grooves (Figure 4.8 top pictures). We obtained similar results in presence of divalent salt, altough the effect of correlations between the counterions and the DNA phosphates on salt cation adsorption are less important compared to monovalent salt cations.

### 4.2.1.3 Conclusion

We reproduced properly the density profiles of the absorbed salt cations and salt anions for the Montoro-Abascal Model (MAM) of DNA with addition of salt ions. The presence of grooves is


Figure 4.8 - Top) Panoramic view of condensed salt cation density in the minor groove (left) and in the major groove (right) for monovalent salt of concentration $C_{s}=0.1 \mathrm{~mol} / \mathrm{l}$. The density unit is $\times 1000$ the real cation density for better clarity. The red line represents the cation density in presence of 120 monovalent counterions. The blue line represents the cation density in presence of 60 divalent counterions. The green line represents the cation distribution in presence of 40 trivalent counterions. Bottom) Panoramic view of condensed salt cation density in the minor groove (left) and in the major groove (right) for divalent salt of concentration $C_{s}=0.2 \mathrm{~mol} / \mathrm{l}$.
crucial for the ion distribution on DNA. An important remark is that the effect of the discreteness of DNA on the ions distribution vanishes a few angstroms away from the DNA surface [267, [16, 254].

### 4.2.2 Effective force between a pair of DNA molecules

The existence of attractive interaction between like-charged polyelectrolytes has been widely studied over the last decades. Such attractive interaction has been observed for DNA [268, 107] and other polyelectrolytes like colloidal rods [269] or charged microspheres [270]. The DNA molecule is very sensitive to the ionic surrounding and it is possible to overcome the electrostatic repulsion existing between DNA molecules in order to form bundle in presence of multivalent counterions. In this section, we probe the interaction between a pair of rigid and aligned DNA along the $z$ direction
in presence of counterions of various valency.
The determination of the total effective force between a pair of DNA molecules is important since the role of each moieties can be quantitatively specified to characterize the attraction/repulsion between the DNA pair. In aqueous solution, the free ions screen the naked electrostatic interaction between the two DNA molecules so that the resulting electrostatic interaction between the pair of DNA becomes weaker. For short distance between the two DNA a mutual attraction due to counterion overscreening can appear resulting in the aggregation of the DNA [255, [16, [254, 266] [271, 272].

### 4.2.2.1 Methods

We consider two parallel DNA molecules disposed along the $z$ axis and localized in the $x O y$ plane with coordinates $P_{1}\left(x_{1}, \frac{L}{2}\right)$ and $P_{2}\left(x_{2}, \frac{L}{2}\right)$ with $\ell=\left|x_{1}-x_{2}\right|$ denoting the distance between the DNA molecules. In our simulations, the relative orientation of the DNA molecules is chosen such that the minor groove of a DNA is exposed to the major groove of the other DNA (Figure 4.9). The double-helix structure of the DNA is an important parameter that influences greatly the effective interaction for a separation distance $\ell<25$ Åas pointed out in a previous work of Allahyarov and co-workers [254]. More precisely, the profile of the effective force has been calculated for the cylindrical model (CM) of the DNA for a separation distance of $\ell<22 \AA$. The force displays highly repulsive peaks when the two phosphate strands of one DNA are exactly facing the two other phosphate strands of the other DNA. There are also lower repulsive peaks when only one strand of a DNA is facing another strand of the other DNA.

The size of the simulation box is $L=102 \AA$ and corresponds to three turns of the DNA molecule. The two DNA are effectively infinite due to the use of periodic boundary conditions. The free ions present in the simulation box are the $N_{c}$ counterions that neutralize the DNA phosphate charges to ensure charge neutrality. Hence, we performed Monte Carlo simulations where counterions of different valencies neutralize the DNA phosphates. In particular, we will consider a number of $N_{c}=\frac{120}{q_{c}}$ with $q_{c}=+1,+2,+3$ or $+4|e|$ and of diameter $d_{c}=3 \AA$. The monovalent and divalent counterions can model $\mathrm{Na}^{+}$or $\mathrm{Mg}^{2+}$ while the trivalent and tetravalent counterions can model spermidine ${ }^{3+}$ or spermine ${ }^{4+}$. We still use a combination of hard sphere pairwise and electrostatic potential to describe the physical interactions as defined in section 3.2.4. A typical snapshot of a
simulation is shown in Figure 4.9. For each system, we performed a NVT Monte Carlo simulation at $T=298 \mathrm{~K}$ with an equilibration phase for $2 \times 10^{4}$ cycles followed by a phase of $4 \times 10^{5}$ cycles to gather statistics. The maximal counterion displacement is bounded by a cubic box of size $0.10 \times L$ during each MC step.


Figure 4.9 - Left) Example of a simulation snapshot with monovalent counterion. The counterions are in blue. Right) Disposition of the two DNA molecules in the $x O y$ plane.

The simulated quantity we are interested in is the effective force between the DNA molecules 16 , [254] along the $O x$ direction:

$$
\begin{equation*}
\mathbf{F}=\mathbf{F}_{1}+\mathbf{F}_{2} \tag{4.20}
\end{equation*}
$$

We will design by $\mathbf{r}_{k}^{i}$ the positions of the different charges in the box with the index $k$ denotes the $\mathrm{k}^{\text {th }}$ particle of species $i$ that could be a counterion $c$ or a phosphate $p$.

The first term $\mathbf{F}_{1}$ is the direct Coulomb force exerted by each phosphate groups $k$ of a DNA molecule on the other phosphate groups $n$ of the other DNA molecule:

$$
\begin{equation*}
\mathbf{F}_{1}=-\sum_{k=1}^{N_{p}}\left(\vec{\nabla}_{\mathbf{r}_{k}^{p}} \sum_{n=1}^{N_{p}} V_{p p}\left(\left|\mathbf{r}_{k}^{p}-\mathbf{r}_{n}^{p}\right|\right)\right) \tag{4.21}
\end{equation*}
$$

where $V_{i j}$ is the interaction potential as described in section 3.2.4 between species $i$ and $j$. Given that the DNA are fixed during the simulation, this term has to be computed only once.

The second term $\mathbf{F}_{2}$ represents the Coulomb interactions between the counterions $l$ in the box and phosphate groups $k$ :

$$
\begin{equation*}
\mathbf{F}_{2}=-\sum_{k=1}^{N_{p}}\left(\left\langle\sum_{l=1}^{N_{c}} \vec{\nabla}_{\mathbf{r}_{k}^{p}} V_{p c}\left(\left|\mathbf{r}_{k}^{p}-\mathbf{r}_{l}^{c}\right|\right)\right\rangle\right) \tag{4.22}
\end{equation*}
$$

Remark: The total force $F$ that takes into account the ions-ions and ions-DNA interactions include both the energetic and entropic contributions driving the evolution of the system. Given that the DNA molecule is completely rigid, the entropic contribution associated to the internal degrees of freedom of the DNA molecule is not taken into account in our simulations.

Instead of using one of the existing method [273] to calculate the PMF for which the reaction coordinate would be the DNA-DNA separation distance, we decided to extract the effective force F at discrete DNA-DNA distance [254, 16, 271]. For instance, the calculation of PMF with the umbrella sampling technique (see section 3.3.5 would require to immobilize one DNA and sample the other DNA position along the distance separating the DNA molecules. It is more difficult to implement the sampling routine since a special care need to be taken in order to move the DNA and the surrounding cloud of ions. Therein, in the current approach the 2 DNA molecules are immobile and we perform a Monte Carlo simulation for discrete separation distance $\ell$.

### 4.2.2.2 Results

We calculate the effective force $\mathbf{F}$ in a box that contains 2 DNA molecules placed in the same configuration as those described in Figure 4.9. The resulting force as a function of the separated DNA distance $\ell$ is presented in Figure 4.10 There is only repulsion between the DNA molecules
when the counterions are monovalent given that counterion condensation is not sufficient to induce DNA-DNA attraction [274].


Figure 4.10 - Effective force calculated for a distance $\ell$ varying between 24 to $48 \AA$ every $2 \AA$ corresponding to 13 separate MC simulations for each curve. We normalized the force according to $F_{o}=\frac{k_{B} T}{P}$ where $P=34 \AA$ is the length of a full DNA turn. A negative force means that there is repulsion between DNA whereas a positive force results in an attraction. The associated errors are given in Appendix C.1

When the counterions are multivalent, counterion condensation increases and correlations between condensed counterions on DNA molecules result in an attractive force with increase of magnitude when the valency of the counterions is increased. Such trend has been also observed in a previous study of Allahyarov and co-workers for the MAM [16]. When the DNA are closed, the mutual repulsion existing between the counterions should damp the total effective force between the pair of DNA molecules, but the presence of the grooves in the MAM allow more space for the counterions to remain in the inner region between the DNA molecules. Indeed, for a simpler DNA model like the cylindrical model (CM) which do not contain grooves, the interaction force decreases in presence of multivalent counterions at short DNA distance [254].

In order to probe the position of the counterions during the MC simulation, we plot the projected counterion density in the $x O y$ plane of the simulation box as shown in Figure 4.11. It is clear that the condensation of counterions at the DNA surface is higher when the valency increases [274] and the DNA molecules exchange fewer ions when they are far away from each other (Figure 4.11
left pictures). When the DNA approach from each other, a local higher density of counterions appears that induces a higher DNA-DNA attraction when the valency of the counterion increased as observed in Figure 4.11 middle and left pictures.

### 4.2.2.3 Conclusion

The calculation of the effective force was performed with MC simulations. In particular, the profiles of the effective forces calculated between the pair of DNA are in agreement with those obtained by Allahyarov and co-workers [16] with grand canonical MD simulations performed with the same model of DNA (MAM model) and counterions of the same radius as in our simulations 4.12 .

We notice that the choice of the DNA model influences greatly the profile of the effective force. For instance, the effective force calculated with the cylindrical model of DNA (CM, see picture 4.1) displays repulsion at short DNA-DNA distance because the CM model do not exhibit grooves such that the ions are depleted from the DNA [16]. On the contrary, the ions are relocated in the grooves of the MAM model at short DNA-DNA distance that do not repulse each other at short distance (Figure 4.12). On the other hand, the radius of the ion is also an important parameter that drives interactions between the DNA. In another work, Allahyarov and co-workers calculated the effective force between MAM DNA with counterions of diameter $\mathrm{d}_{c}=8$ and of different valencies [255]. They show that the force between the DNA become repulsive at short distance given that the big counterions cannot relocate in the grooves and are thus depleted from the DNA molecules.

However, we notice that the number of MC cycles was small compared to several millions of MD steps of similar studies [254, 255], which constitutes the main drawback in our MC simulations.

We confirm through our simulations that the pair of DNA can be aggregated spontaneously with multivalent cations. This is similar to previous MD simulations that determined the PMF between two atomistic DNA with divalent, (putrescine ${ }^{2+}$ ), trivalent (spermidine ${ }^{3+}$ ) and tetravalent counterions (spermine ${ }^{4+}$ ) [275]. The depth of the PMF increases when the valency of the counterions increases as well, which is in agreement with the profile of the calculated effective force in our MC simulations.
A) $\mathbf{q}_{\mathbf{c}}=+1|e|$

B) $q_{c}=+2|e|$

C) $\mathbf{q}_{\mathrm{c}}=+3|\mathrm{e}|$


Figure 4.11 - Projected densities of counterion in the $x O y$ plane of the simulation box for counterions of different valencies. The separation distance between the DNA is $\ell=48,36$ and $24 \AA$ respectively from the left, middle and right pictures. A) $q_{c}=+1|e|$. B) $q_{c}=+2|e|$. C) $q_{c}=+3|e|$. D) $q_{c}=+4|e|$. The helical projection is given in number of particles per $\AA^{2}$.


Figure 4.12 - Effective force calculated between a pair of DNA molecules with monovalent, divalent or trivalent counterions in presence of various amount of monovalent salt by Allahyarov and coworkers. The solid line in cyan corresponds to the calculations performed in this section ( $\mathrm{c}_{s}=$ $0 \mathrm{Mol} / \mathrm{l})$. Then, after insertion of increasing amount of monovalent salt from $\mathrm{c}_{s}=0.024 \mathrm{Mol} / \mathrm{l}$ (dashed green line), $0.097 \mathrm{Mol} / \mathrm{l}$ (dot-dashed blue line), $0.197 \mathrm{Mol} / \mathrm{l}$ (red solid line with symbols), $0.71 \mathrm{Mol} / \mathrm{l}$, (black dashed line with symbols) the force becomes repulsive between the DNA because of salt that screens electrostatic interactions. The force is normalized according to $F_{o}=\frac{k_{B} T}{P}$ where $P=34 \AA$ is the length of a full DNA turn. A negative force means that there is attraction between DNA whereas positive force results in a repulsion. Picture adapted from ref. [16] (Physical Review E, 2004, 69, 041904, Figures 4-5-8, Copyright 2004 with permission from APS).

### 4.2.3 Osmotic Pressure calculation in hexagonal DNA bundle

In the previous section, we quantitatively determined the interacting force acting between two rigid fixed DNA molecules in presence of multivalent counterions. Even though such approach gives insight about DNA aggregation, it is unlikely one can extrapolate the conclusions valid for a pair of DNA molecules to a bundle of DNA because of the different rearrangement of the counterions around the DNA molecules. Among possible ways to study bundle formation, we present two possible techniques.

- First, a "brute force" technique consists of inserting randomly the compaction agents and the DNA molecules in a simulation box and let the system equilibriates with a MC or MD scheme towards a configuration where a bundle of finite size can emerge. This technique has been used for instance to study the bundle formation of short DNAs with trivalent metallic ions in presence of monovalent salt [276, [277]. However, one has to be careful during the equilibration phase given that the system can easily reach a metastable equilibrium.
- On the other hand, one considers an infinite bundle of fixed DNA molecules (a DNA pre-
cipitate) in the simulation box in the desired configuration and we let the condensing agents evolve freely according to a MC or MD scheme and at equilibrium we determine quantitatively the mechanical stability of the resulting system.

In the following section, we will apply the second technique to study the stability of a hexagonal bundle of DNA molecules in presence of counterions. The quantity of interest is the osmotic pressure in the bundle. It is possible to use a MD scheme to perform such task but we decided to adapt our MC simulation package to perform it. This technique has been successfully used to study the stability of hexagonal DNA bundle with ions [278, 17] and has been applied to study the bundle stability of rodlike polyelectrolyte like negative actin protein in presence of positive lysozymes [279, 62, [280] or viruses in presence of small ions [281].

### 4.2.3.1 Notion of osmotic pressure

Let us consider a pure solvent $s$ and a solution of a solute $\sigma$ in the same solvent, separated by a semipermeable membrane as shown in Figure 4.13 so that the solvent can transfer from compartment A to compartment B while the solute remains only in compartment B. During the osmosis phase described in 1) of Figure 4.13 , the pure solvent migrates from compartment A to compartment B that contains the solution. When osmotic equilibrium is reached in 2 ), the same flow rate of solvent is exchanged between the compartments. The pressure exerted by the pure solvent in the compartment A on the compartment B is denoted by $P$. On the other side, the pressure exerted by the solution in compartment B on compartment A is $P+\Pi$.

The pressure $\Pi$ is directly related to the difference of height between the surface of liquid $\Delta h$ in the compartment and can be expressed with the fundamental law of hydrostatics

$$
\begin{equation*}
\Pi=\rho g \Delta h \tag{4.23}
\end{equation*}
$$

The term $\Pi$ is called osmotic pressure and Van't Hoff established a relationship between the osmotic pressure and the concentration of solute in the solution [282]:


Figure 4.13 - The compartment A contains a pure solvent $s$ and the compartment B contains a solution of a solute $\sigma$ in the same solvent. The compartments are separated with a semipermeable membrane indicated by the dashed region. 1) Because the flow of pure solvent through the membrane from left to right is greater than the flow of the solvent in the reverse direction (see black arrows), the level of liquid in compartment B rises. 2) When equilibrium is reached, the pressure differential equal to the osmotic pressure $\Pi$ equalizes the flow rate of the solvent in both compartments.

$$
\begin{equation*}
\Pi=\frac{n_{\sigma} R T}{V} \tag{4.24}
\end{equation*}
$$

where V is the volume of the solution in compartment $\mathrm{B}, n_{\sigma}$ the number of moles of solute present in the solution, $R$ the ideal gas constant and $T$ the temperature. The equation 4.24 can be written also as a function of the number of particles of solute $N_{\sigma}$ present in solution:

$$
\begin{equation*}
\Pi=\frac{N_{\sigma} k_{B} T}{V} \tag{4.25}
\end{equation*}
$$

Remark: The form of equation 4.25 is the same as the ideal gas law. The pressure exerted by a solute in a solution of volume $V$ is similar to the pressure exerted by an ideal gas constituted of the same number of particles in a volume $V$.

### 4.2.3.2 Methods

In an orthorhombic simulation box, let us consider 4 aligned DNA molecules made of three pitches $P$ disposed in a hexagonal fashion along the $z$ axis so that the lattice spacing equivalent to the distance between the DNA is denoted by $\ell$. The sides of the orthorhombic box depend on $\ell$ in the following way:

$$
\left\{\begin{array}{l}
L_{x}=2 \ell  \tag{4.26}\\
L_{y}=\sqrt{3} \ell \\
L_{z}=3 P
\end{array}\right.
$$

where P represents a DNA helical turn. It is clear that we reproduce a hexagonal lattice of infinitely long DNA with the orthorhombic cell matching the simulation box by using periodic boundary conditions. We could have alternatively used an hexagonal cell to describe the hexagonal lattice formed by the DNA [17]. The free ions present in the DNA bundle are the counterions $N_{c}$.

Remark: The bundle stability will be studied at the isoelectric point $R_{+/-}=1.00$ (where $R_{+/-}$refers to the ratio of the sum of the total charge of counterions to the total charge of the DNA) and we assume that the totality of the counterions remains in the DNA bundle to enforce charge neutrality (i.e no counterions in the supernatant solution).

The osmotic pressure will be the sum of two terms: a kinetic term $\Pi_{\text {ideal }}$ due to the motion of the counterions in the DNA bundle and a term coming from the electrostatic interactions of all charges present in the system $\Pi_{\text {electrostatics }}$. The osmotic pressure can be calculated with the virial theorem (equation 3.89).

$$
\begin{equation*}
\Pi=\Pi_{\text {ideal }}+\Pi_{\text {electrostatics }} \tag{4.29}
\end{equation*}
$$



Figure 4.14 - A) Snapshots of hexagonal DNA bundle viewed in the $O x$ direction. B) Snapshots of hexagonal DNA bundle viewed in the $O z$ direction. The lattice expansion corresponds to an increase of the DNA-DNA distance $\ell$ as well as of the simulation box.

$$
\begin{equation*}
\Pi=\frac{N k_{B} T}{V^{\prime}}+\frac{1}{3 V^{\prime}} \sum_{i<j} \mathbf{r}_{i j} \mathbf{F}_{i j} \tag{4.30}
\end{equation*}
$$

where $N$ denotes the number of free ions in the system and $\mathbf{F}_{\mathbf{i j}}$ denotes the electrostatic force between $i$ and $j$ and $V^{\prime}$ the accessible volume for the counterions in the box.

The osmotic pressure is calculated for increasing values of DNA-DNA distance separation $\ell$ as illustrated in Figure 4.14 The box size is changed accordingly with the DNA distance to simulate the bundle expansion at a large scale given the PBC conditions. For a given DNA-DNA distance $\ell$, we performed a standard NVT Monte Carlo simulation at $\mathrm{T}=298 \mathrm{~K}$ with an equilibration phase of $5 \times 10^{3} \mathrm{MC}$ cycles and then gather statistics to calculate osmotic pressure during $4 \times 10^{5}$ MC cycles. The counterions have a diameter $d_{c}=1 \AA$. The maximal counterion displacement is restrained in a cubic box of size $0.10 \times L$ during each MC step.

### 4.2.3.3 Influence of counterion charge on DNA lattice stability

We calculated the osmotic pressure for an hexagonal lattice of DNA where counterions of diameter $d_{c}=1 \AA$ are added such that $N_{c}=\frac{240}{q_{c}}$ with $q_{c}=+1,+2,+3$ or $+4|e|$. The osmotic pressures are plotted in Figure 4.15 and are in agreement with the previous calculated effective force
between a pair of DNA molecules with the MAM model. In presence of monovalent counterions, the DNA-DNA interaction is repulsive over the whole range of DNA separation as mentioned in previous studies performed with MC calculations [258, 283], which means that the DNA bundle is unstable with monovalent counterions. On the contrary, when the valency of the counterions is increased, the osmotic pressure is negative and indicate a bundle contraction.

We notice that the profile of osmotic pressures for multivalent counterions do not show a negative minimum associated with repulsion for short distance $\ell$ as displayed for osmotic pressure of DNA bundle calculated with divalent ions in the work of Lyubartsev and Nordenskiöld [258]. First, this fact may come from the pairwise hard core potential we used to model the steric interactions instead of a continuous repulsive potential as used in their studies. Then, the DNA model used in their simulation is closed to the extended cylindrical model (ECM) contrary to the more grooved MAM model we used in our simulations. The presence of the grooves allows more space for counterion motion in the DNA bundle at short DNA-DNA distance as already pointed out by Allahyarov and co-workers from calculation of effective force between a pair of MAM DNA molecules (Figure 4.10 of section 4.2.2.


Figure 4.15 - Osmotic pressure calculated in hexagonal bundle of DNA compacted with counterions of different valencies of diameter $\mathrm{d}_{c}=1 \AA . q_{c}=+1|e|$ (cyan). $q_{c}=+2|e|$ (blue). $q_{c}=+3|e|$ (red). $q_{c}=+4|e|$ (black). The associated errors are given in Appendix C. 2

### 4.2.3.4 Influence of counterion radius on DNA lattice stability

We probe now the effect of counterion radius on the osmotic pressure. We plot in Figure 4.16 the osmotic pressure for divalent counterions by varying the counterion diameter.


Figure 4.16 - Osmotic pressure calculated in hexagonal bundle of DNA condensed by divalent counterions for different counterion diameters. The associated errors are given in Appendix C. 2

When the radius of the counterion is increased, the osmotic pressure increases as well and the bundle becomes unstable for larger lattice step $\ell$. Such trend has been observed for osmotic pressure calculated by Lyubartsev and Nordenskiöld for the series of divalent counterions of the same radius as those used in our simulations. It is expected that counterions of larger radius result in lower electrostatic interactions between DNA-counterions as well as counterions-counterions. The pressure due to the ideal gas term remains constant and the resulting osmotic pressure is thus higher. It has been shown that $\mathrm{Mn}^{2+}$ induces a spontaneous DNA assembly into an ordered phase [284] while $\mathrm{Mg}^{2+}$ does not [285]. Such results indicate a possible link between bundle stability and ion size since the hydrated ionic radius of $\mathrm{Mg}^{2+}(r=2.99 \AA)$ is larger than the hydrated ionic radius of $\mathrm{Mn}^{2+}(r=2.86 \AA)$ [286].

### 4.2.3.5 Conclusion

We performed osmotic pressure calculations to determine the stability of a hexagonal bundle of DNA molecules in presence of counterions of various valency or radius. We show that an increase of
the counterion valency results in a more stable bundle, which is consistent to the trend observed for the effective force calculated in section 4.2.2. Then, we show that an increase in counterion radius destabilizes the DNA bundle. The bundle stability is explained through electrostatic interaction between the DNA molecules and the counterions [258]. Although we do not include water molecules, the inclusion of an hydrated radius for the counterions is a simple path to mimic the effect of water solvent into the minimalistic CG model. Lyubartsev and Nordenskiöld studied also the stability of the bundle in presence of divalent counterions and various salt concentration [258]. In particular, they study the influence of salt of monovalent anions and either monovalent and divalent cations that can be exchanged between the bundle and the supernatant phase by performing independent grand canonical MC simulations in the bundle and in bulk at constant chemical potential. A significant fraction of salt is released in the supernatant phase upon bundle formation, contributing to stabilize the bundle because of the resulting osmotic pressure exerted by salt on the DNA bundle.

### 4.3 Self-Assembly of DNA mediated by nanoparticles

The interaction of DNA with simple ions has been widely simulated in the past years, but it is generally more difficult to study DNA interactions with more complicated and/or charged compaction agents, especially if Monte Carlo simulations are used given that it requires the implementation of specialized moves (see section 3.2.3.2). We will use the simple rigid model of AuNP presented in section 4.1.2 First, we will study the interaction of AuNPs with a single fixed DNA. Then, we will calculate the effective force between a pair of DNA molecules mediated by various amount of AuNPs. Finally, the osmotic pressure of a DNA bundle compacted by AuNPs will be calculated. Despite the various phases that DNA-AunP bundle could form, we will consider only hexagonal and quadratic DNA lattices.

### 4.3.1 Aggregation of nanoparticles on a single fixed DNA

In this section, we perform a first set of Monte Carlo simulations to study the interaction of AuNPs with a single fixed DNA. For various $R_{+/-}$ranging from 0.50 to 1.50 , we characterize the repartition of the AuNPs on the DNA molecule. In most of the simulation, monovalent counterions of the DNA and of the AuNPs will be present in the simulation box. The assembly of AuNPs on

DNA has been studied by Komarov and co-workers [263] in a similar way but for a simpler model of AuNP. Altough simulations involving a single DNA appear to be minimalistic, it already reveals some important features about interactions and behavior between DNA and AuNPs.

### 4.3.1.1 Methods

We insert a single fixed DNA molecule at a center of the $x O y$ plane and parallel to the $z$ axis of a cubic box of side $L=510 \AA$ which represents 15 DNA helical turns (i.e. 150 bps ). The DNA is then neutralized with monovalent counterions (cations) of diameter $3 \AA$. The resulting system is equilibrated through a NVT run for $10^{4}$ Monte Carlo cycles.

Then, various amount of AuNPs with their associated negative counterions (anions) of diameter $3 \AA$ are randomly inserted in the box. For each system, a subsequent NVT MC simulation at $T=298 K$ is performed for $2 \times 10^{5} \mathrm{MC}$ cycles. At each MC step, we randomly pick a particle to be displaced. If an ion is picked, we apply a simple displacement in the simulation box (see section 3.2.3.1 , whereas a translation or a rotation can be performed with the same probability if a AuNP is chosen. In the latter case, a collective move is performed (see section 3.2.3.3) on the chosen AuNP and its surrounding ions located in a radius $R_{\text {Gold }}=15 \AA$ around the gold core center. The simulations are performed at constant density of DNA $\left(2.2 \times 10^{-6}\right.$ phosphates $\left./ \AA^{3}\right)$.

Due to the use of PBCs, we are simulating an infinitely long DNA molecule (i.e no end effecs are present). We summarize the simulations that we performed for different models of AuNP and different ratio $R+/-$ in Table 4.1.

Remark: For the case $R_{+/-}=1.00$ when the number of DNA phosphate charges is exactly the same as those carying by the AunPs present in the system, we perform also a simulation without the DNA and AuNP counterions.

We provide a description of the analysis to characterize the evolution and the equilibrium of each system.

- We set criteria to determine whether an ion or an AuNP is adsorbed on the DNA. We distinguish between adsorption on the strand and in the grooves to calculate the number of adsorbed species on the DNA. In the case of the AuNPs, we consider adsorption on the strand if the center of the gold core is comprised in a region delimited by a height $\xi=3.4$

Table 4.1 - Table summarizing the different species present in the simulation box. Each line represents a simulation with a given number of AuNPs, cations and anions.

| AuNP model | $R_{+/-}$ | AuNP | Cations | Anions |
| :--- | :---: | :---: | :---: | :---: |
| 6 ligands | 0.50 | 25 | 300 | 150 |
|  | 0.75 | 37 | 300 | 222 |
|  | 1.00 | 50 | - | - |
|  | 1.00 | 50 | 300 | 300 |
|  | 2.00 | 100 | 300 | 600 |
|  | 3.00 | 150 | 300 | 900 |
| 12 ligands | 0.50 | 12 | 300 | 144 |
|  | 0.75 | 19 | 300 | 228 |
|  | 1.00 | 25 | - | - |
|  | 1.00 | 25 | 300 | 300 |
|  | 2.00 | 50 | 300 | 600 |
|  | 3.00 | 75 | 300 | 900 |
| 30 ligands | 1.00 | 10 | - | - |
|  | 1.00 | 10 | 300 | 300 |
|  | 2.00 | 20 | 300 | 600 |
|  | 3.00 | 35 | 300 | 900 |

$\AA$ centered on the DNA phosphate groups and from a width to the phosphates surface of $\delta=2+d_{A_{N} P} / 2 \AA$ where $\mathrm{d}_{A_{N} P}=23 \AA$ is the value of the AuNP diameter that comprised the shell of ligands. An AuNP is adsorbed in the minor groove if the center of the AuNP falls in the minor groove of height $\xi=10.2 \AA$ and is separated from the DNA core cylinder of $\delta=2+d_{A_{N} P} / 2 \AA$. An AuNP is adsorbed in the major groove of height $\xi=17.0 \AA$ in the same manner. The adsorption of small ions on the DNA are calculated in the same way as those for AuNPs excepted that $\mathrm{d}_{A_{N} P}$ is replaced by $\mathrm{d}_{i o n}=1.5 \AA$.

- We calculate the coordination number function (CN) of the different species $i$. The CN is defined radially from the DNA axis. We project first the coordinates of each species in the $x O y$ plane and we calculate the radial distribution function defined as:

$$
\begin{equation*}
g_{i}(r)=\frac{1}{\rho_{s}} \sum_{j=0}^{N s} \frac{\delta(r)}{\pi r^{2}} \tag{4.31}
\end{equation*}
$$

where $r$ is the radial distance from the DNA center, $\rho_{s}$ is the density of species $s$. The coordination number is expressed by integrating the radial distribution function to find the cumulated number of particles of species $s$ from a radial distance r of the DNA axis:

$$
\begin{equation*}
C N_{i}(r)=\rho_{s} \int_{0}^{\frac{L}{2}} \pi r^{\prime 2} g_{i}\left(r^{\prime}\right) d r^{\prime} \tag{4.32}
\end{equation*}
$$

The radial distribution and the coordination number are calculated until $L / 2$ in order to avoid double counting of particle of species $s$ because of PBCs. The coordination number $\mathrm{CN}_{A u N P}(r)$ (and by extension $\left.\mathrm{g}_{A u N P}(r)\right)$ takes into account the center of mass of the nanoparticle.

- The charge compensation 13 is defined as the radial integrated charge over the simulation box. The term $q_{A u N P}$ denotes the charge carried the AuNPs.

$$
\begin{equation*}
\theta(r)=|e|\left[-C N_{\text {phos }}(r)+C N_{\text {cation }}(r)-C N_{\text {anion }}(r)+q_{A u N P} \times C N_{A u N P}(r)\right] \tag{4.33}
\end{equation*}
$$

- One way to calculate the species distribution around the DNA has been proposed by Abascal and Montoro [287] and is called "helical projection" in the $x O y$ plane. Each ion or AuNP is projected onto the nearest base pair in a spiral staircase fashion. The helical projection required 2 coordinates. The first one is the radial coordinates $\rho$ defined from the DNA center as shown in Figure 4.17. The projection on the DNA can be done either by moving along the $z$ axis from a distance of $\delta z$ or along the axial distance to the strand from a corresponding angle $\delta \phi$. We used the latter representation and we obtain the helical representation ( $\rho, \delta \phi$ ) from the former cylindrical coordinates ( $\rho, \phi, \mathrm{z}$ ) with the following relationship

$$
\begin{equation*}
\delta \phi=\frac{360}{L} z-\phi \tag{4.34}
\end{equation*}
$$

with $L$ the pitch of the DNA molecule. It is possible to express the helical projection in terms of cartesian coordinates

$$
\left\{\begin{array}{l}
x=\rho \cos (\delta \phi)  \tag{4.35}\\
y=\rho \sin (\delta \phi)
\end{array}\right.
$$



Figure 4.17 - Coordinates used to calculate the helical projection. For a particle $A$ of coordinates $\left(\rho_{A}, \phi_{A}, \mathrm{z}_{A}\right)$ in the cylindrical basis, the helical projection requires only two coordinates which are $\rho_{A}$ and $\delta \theta$ or $\delta$ z. Picture adapted from ref. [287] (The Journal of chemical physics, 1998, 109, 6200-6210, Figure 2, Copyright 1998 with permission from AIP Publishing).

- The characterization of the AuNP-AuNP distance is estimated with the following procedure performed at each MC step. When the system is in its equilibrium state, we consider only the AuNPs adsorbed on the DNA. For each AuNP adsorbed on the DNA molecule, we determine the closest other AuNP adsorbed on the DNA and we calculate the associated distance along DNA long-axis $z$ separating this two AuNPs corresponding to the index $k$ defined as:

$$
\begin{equation*}
k=\operatorname{int}\left(\frac{z}{\delta z}\right) \tag{4.37}
\end{equation*}
$$

where $\operatorname{int}(x)$ stands for the integer part of the real $x$ and $\delta z=2 \AA$.

The histogram is then normalized by the total number of counts and MC cycles. The highest probability corresponds to the most probable distance between the adsorbed AuNPs. The AuNPs adsorption on the fixed DNA will depend both of the AuNP charge and the amount of AuNPs present in the system.

### 4.3.1.2 Adsorption of AuNPs on the DNA molecule

After the insertion of the AuNPs and the associated counterions, the AuNPs progressively adsorbed on the DNA molecule. This process induces a change in the electrostatic energy of the system because of the adsorption of positive AuNPs on the negative DNA but also a variation of the entropy through the release of DNA positive counterions from the DNA surface that participate in the increase of translational entropy of the system.

We present snapshots of the several equilibrated systems (also without DNA and AuNPs counterions) for $R_{+/-}=1.00$ in Figure 4.18 where all the AuNPs are adsorbed on the DNA molecule. At a first sight, the aggregation is characterized by a regular distribution observed for every model of AuNP along the DNA molecule due to their electrostatic repulsion. In order to highlight the AuNP condensation associated with the release of the cations from the DNA, we calculated the number of absorbed species at the DNA surface through the simulation run for the systems presented in Table 4.1. First, we study the number of AuNPs adsorbed on the DNA as a function of $R_{+/-}$. The results are presented in Figure 4.19. For each model of AuNP, there is a complete aggregation of AuNPs on the DNA for $R_{+/-} \leq 1.00$, which means that at the isoelectric point, the DNA phosphate charges are totally neutralized by the AuNPs.

For higher number of AuNPs present in the system, it is possible to aggregate more AuNP on the DNA until a limit is reached. This phenomenon called overcharing may be explained by the electrostatic repulsion of the remaining AuNPs in the bulk that push AuNPs into available space on the DNA. Indeed, correlations between AuNPs make possible to aggregate around $\times 56-57$ or $\times 36$ 37 AuNPs if we consider respectively the 6 -AuNP and 12 -AuNP model, when $R_{+/-}$is increased up to 2.00 or 3.00 . Although similar conclusion emerges for the aggregation of the $30-\mathrm{AuNP}$, it seems that more particles can be aggregated on the DNA given that the adsorption curve did not reach a plateau at the end of the simulation. The acceptance rate for adsorbing a nanoparticle becomes very low and the system equilibration time becomes very large compared to the sampling


Figure 4.18 - Equilibrated systems containing either 6 -AuNPs, 12 -AuNPs or 30 -AuNPs at $R_{+/-}=$ 1.00. All the AuNPs has been adsorbed on the fixed DNA. The small cations are in green and the small anions are in grey.
time accessible with our direct MC simulations.

In the same way, we present the evolution of adsorbed cations on the DNA in Figure 4.20 When the number of AuNPs increases in the simulation, the number of cations adsorbed on the DNA decreases as well. The number of cations absorbed is the same whenever the 6 -AuNPs or the 12 -AuNPs are present in the system. However, for $R_{+/-}=1.00$, more cations are adsorbed for the 30-AuNP model (around 40 cations) due to the larger available space between adsorbed AuNPs on the DNA (Figure 4.18).

We probe the adsorption of the AuNPs and ions on the DNA and their overall distribution throughout the simulation box by calculating the coordination number (CN) defined from equations 4.32 and 4.31 The results are presented for systems containing the 12-AuNP nanoparticles in Figure 4.21 The CN function for cations (top) vanishes at short distance when $R_{+/-}$increases due to a higher number of 12 -AuNPs adsorbed on the DNA (bottom). For $R_{+/-}=2.00-3.00$, the flat profile from a radial distance of $20-100 \AA$ for the 12 -AuNP CN reveals that most of the AuNPs that remain in bulk are located after a distance of $100 \AA$ from the DNA core. Instead, this region of the simulation box is occupied by anions (up to $\times 150$ anions within a cylindrical region of radius $100 \AA$, middle picture). These features may reveal a possible emergence of successive layers of alternate charges, to be discussed in the following section.


Figure 4.19 - Adsorbed AuNPs on DNA for different $R_{+/-}$. Top) Systems with 6-AuNPs. Middle) Systems with 12-AuNPs. Bottom) Systems with 30-AuNPs.

### 4.3.1.3 Density of AuNPs and ions in the vicinity of the DNA

In this section, we probe more precisely the distribution of the different species around the DNA. We used equations 4.344 .35 and 4.36 to draw the helical projection of each species in the $x O y$ plane of the DNA. In figure 4.22 the helical projection is presented for mobile species in system characterized by a ratio $R_{+/-}=1.00$.


Figure 4.20 - Adsorbed cations on DNA for different $R_{+/-}$. Top) Systems with 6-AuNPs. Middle) Systems with 12-AuNPs. Bottom) Systems with 30-AuNPs.

We focus on the distribution of cations (top, Figure 4.22). We remark that the different density profiles depend implicitely on the adsorption of other species in the systems. The cation density is


Figure 4.21 - Coordination number (CN) of the different species for systems containing the 12AuNP model, for various $R_{+/-}$ratios. The minimal approach distance of the 12 -AuNPs center from the DNA axis is $\sim 12 \AA$.
higher in the minor groove in all cases and we distinguish clearly the shape of DNA [287]. However, in the major groove, the density profile of cations vanishes when more AuNPs are present in the system (Figure 4.18) which implies that most of the AuNPs are adsorbed in the major groove. Indeed, we calculated the density profile of AuNPs presented in the middle set of maps in Figure 4.22 and found that the distribution is the highest in the major groove. The spatial density distribution of nanoparticles in the major groove is narrower (in both radial and perpendicular directions to the


Figure 4.22 - Helical projection in the $x O y$ plane of the species density in the vicinity of the DNA molecule for systems with $R_{+/-}=1.00$. The index " $\mathbf{m}$ " indicates the minor groove, " $\mathbf{M}$ " the major groove and " $\mathbf{S}$ " the phosphate positions. Top) Helicial distribution of cations. Middle) Helical distribution of AuNP. Bottom) Helical distribution of anions. The helical projection is given in number of particles per $\AA^{2}$.

DNA surface) for the 30 -AuNPs than for 6 -AuNPs and 12 -AuNPs due to several reasons. From an energetic point of view, the 30-AuNP model contains more ligands that favor stronger electrostatic interactions with the close phosphates: a MC step aiming to displace the 30-AuNP along the DNA would be more likely rejected because of the unfavorable energy change. From a structural point of view, the $30-\mathrm{AuNPs}$ are bulkier than the 12 -AuNPs and the 6 -AuNPs because of the larger number of ligands that cover the gold core: there would be more interpenetration between the adsorbed $30-\mathrm{AuNPs}$ and the DNA if a MC step to attempt displacement or rotation of the $30-\mathrm{AuNPs}$ is performed. Finally, the anion distribution around the DNA molecule is shown at the bottom of Figure 4.22 The anion density remains low in the DNA vicinity. The presence of the 6 -AuNP does
not influence the anion distribution around the DNA due to its low charge density. When the charge of the AuNP is increased, an outer layer made of AuNP negative counterions emerges in the major groove.

### 4.3.1.4 DNA overcharging effect

We calculated the charge compensation using the equation 4.33 to probe the influence of the adsorbed AuNPs, cations and anions on the DNA effective charge experienced at a given radial distance, as shown in Figure 4.23 The charge compensation is positive at short distance at $\sim$ $8.5 \AA$ due to cations populating grooves and strand of the DNA molecule. After this maximum, a drop is observed due to the integrated density of DNA negative phosphate groups located at a radial distance $r=8.9 \AA$ from the DNA center. When $r$ is progressively increased, the effective charge increases and for excess of AuNPs $\left(R_{+/-}=2.00-3.00\right)$ not only the AuNPs compensate the DNA charge, but exceed it resulting in opposite values of integrated charge (charge inversion) at some distance and apparition of layers of opposite charges. It is important to remark that the onset of overcharging effect begins even for $R_{+/-}=1.00$, with some monovalent cations remaining adsorbed on DNA in addition to the totality of the AuNPs that compensate exactly the DNA charge (see $R_{+/-}=1.00$ without counterions). We notice that for systems containing the $30-$ AuNPs, there is a drop of the charge compensation near the DNA surface (blue curve for $R_{+/-}=$ 1.00 ), because anions tend to bind to the $30-\mathrm{AuNP}$ outer surface. It has been shown that rodlike polyelectrolytes like DNA can undergo a charge reversal in presence of an excess of condensed multivalent counterions both experimentally [102] or through numerical simulations [15, [13]. In our systems, this positive overcharging arises with the condensation of AuNPs while a second layer associated to the reversal charge around $r=150 \AA$ is due to the attracted anions that screen the first layer of AuNPs adsorbed on the DNA [288]. The correlations between AuNP-AuNP lead to depletion of cations from the DNA surface that induces the charge inversion [288].

### 4.3.1.5 Distribution of AuNP on the DNA

We estimated the distance along DNA long-axis between closest adsorbed AuNPs by calculating the corresponding normalized histogram of the consecutive AuNP distance. We performed such analysis only for the systems that contain 6 -AuNPs and 12 -AuNPs. The systems with the $30-$


Figure 4.23 - Left) Charge compensation $\theta(\mathrm{r})$ versus distance from the DNA core for systems with different model of AuNP at different $R_{+/-}$. Right) Zoom of the charge compensation $\theta(\mathrm{r})$.

AuNPs did not display satisfactory results due to two reasons. First, there is fewer 30-AuNPs than 12 -AuNPs or 6 -AuNPs at the same $R_{+/-}$ratio. Consequently, more MC steps are required to obtain good statistics. The other reason comes from the higher charge density of the $30-\mathrm{AuNP}$. As soon as the $30-\mathrm{AuNPs}$ are adsorbed on the DNA, the large unfavorable variation of electrostatic potential energy renders a MC move attempt to unbind a nanoparticle unlikely. The histograms are presented in Figure 4.24


Figure 4.24 - Normalized histograms of the distance along DNA long-axis between closest AuNPAuNP adsorbed on the DNA. Top) Systems that contain 6-AuNPs Bottom) Systems that contain 12-AuNPs.

Let us focus on the histogram of the distance along DNA long-axis between closest 6 -AuNPs adsorbed on the DNA in Figure 4.24 top picture. The peak for each histogram represents the most probable distance along DNA long-axis between closest adsorbed 6 -AuNPs. In the case $R_{+/-}=$ 1.00 , the presence of the DNA and AuNPs counterions does not modify the histogram profile and the most probable distance for both cases is $10 \AA$. At equilibrium, $\times 50$ AuNPs are adsorbed on the DNA of length $L=510 \AA$, for an AuNP-AuNP distance of $\frac{510}{50}=10.2 \AA$. It means on average that the AuNPs are evenly spaced along the DNA as shown also on the snapshot of the system on left, Figure 4.18 Such configuration is adopted by the 6-AuNP due to a minimization of electrostatic energy through correlation of AuNP-AuNP adsorbed on the DNA.

For $R_{+/-}=0.50-0.75$, the histograms present a broader distribution for the $z$-distance between consecutive AuNPs. Since fewer AuNPs are adsorbed on the DNA, they experienced a weaker
electrostatic repulsion. However, there is still a preferred distance between consecutive AuNPs of $14 \AA$ for $R_{+/-}=0.75$. This is in agreement with an evenly spaced disposition of the AuNPs along the DNA that would give an interspace distance of $\frac{510}{37} \approx 13.8 \AA$. At $\mathrm{R}_{+/-}=2.00-3.00$, a higher number of 6 -AuNPs are adsorbed on the DNA than in systems at $\mathrm{R}_{+/-}=0.50-0.75$ so that correlations between adsorbed 6 -AuNPs become more important and induce collective motion of 6 -AuNPs along the DNA molecule. For that reason, the distributions of interspacing distance between AuNPs adsorbed on DNA are narrower than in systems with lower amount of AuNPs with a most probable distance of $\sim 8 \AA$. Because of the weaker 6 -AuNP density on the DNA for systems at $\mathrm{R}_{+/-}=0.50-0.75$, the electrostatic repulsions between 6 -AuNPs are weaker and the motion of the 6 -AuNPs adsorbed on the DNA are less coupled with each other as shown from the broader distribution of $z$-distance between consecutive adsorbed AuNPs.

The results for systems containing the 12 -AuNPs are presented on the bottom picture in Figure 4.24 At the isoelectric point $\left(R_{+/-}=1.00\right)$, the histograms display a maximum at $\sim 20 \AA$ that matches the spacing distance of $\frac{510}{25}=20.4 \AA$ if the 12 -AuNPs are evenly spaced along the DNA.

For $R_{+/-}=0.50-0.75$, the distribution is broader. In addition to a fewer number of $12-$ AunPs adsorbed on DNA compared to the simulations involving the 6 -AuNP model, a layer of anions adsorb at the surface of the 12 -AuNPs due to their higher charge density and screen the AuNP-AuNP repulsion. Such effect can disturb the distribution of the 12-AuNP along the DNA.

The systems with an excess of 12 -AuNPs behave in a way similar than those with an excess of 6 -AuNPs. The histograms present the same type of pattern for $R_{+/-}=2.00-3.00$ due to a saturation of the DNA by 12 -AuNPs. For such cases, the most probable consecutive distance is $14 \AA$ which is equivalent to the distance of $\frac{510}{36} \approx 14.2 \AA$ separating the adsorbed $\sim 3612$-AuNPs (Middle picture of Figure 4.19).

### 4.3.1.6 Conclusion

We performed MC simulations on systems made of a single fixed DNA with different models of AuNP and in presence of their respective counterions. We confirm through our simulations that multivalent AuNPs adsorb on the DNA and induce the ejection of the positive DNA counterions. An excess of AuNPs leads to the overcharging of DNA surface and the onset of appearance of an effective charge forming layers of alternate sign as a function of the radial distance from the DNA
center. Consequently, only a finite amount of AuNPs can adsorb on the DNA and the AuNP-AuNP correlations result in an evenly spaced repartition of the AuNPs along the DNA. Such configuration has been observed experimentally in linear assembly of AuNPs electrostatically organized on DNA molecule by Warner and Hutchison [7] or DNA origami by Julin and co-workers [146]. The direct MC simulations highlight the spontaneous evolution of the systems, i.e. the AuNPs adsorption on DNA in presence of counterions but further analysis are required to characterize quantitatively this evolution as well as the role of counterions.

### 4.3.2 Effect of counterions on the adsorption of AuNP on DNA molecule: PMF calculations

The effect of counterions on the adsorption of AuNPs on the DNA is quantitatively determined in this section. In general, ions play essential roles in polyelectrolyte solutions being able to greatly modified the effective interactions between polyelectrolytes [107]. It has been shown that Coulomb repulsion between like-charge polyelectrolytes can be weakened by the screening of monovalent ions [16]. However, until recently [289], less attention has been paid to interaction between oppositively charged polyelectrolytes and this motivates calculation of the potential of mean force (PMF) between an AuNP and a DNA molecule in presence of various amount of salt. In particular, we want to highlight the effect of small ions on the adsorption of AuNPs on the DNA molecule.

### 4.3.2.1 Methods

The systems are prepared in the following way. We retrieve a final configuration corresponding to the equilibrated system that contains the $\times 1030$-AuNPs adsorbed on the 15 -turn DNA obtained in absence of their respective counterions (section 4.3.1).

Table 4.2 - Systems for which the PMF is calculated.

| AuNP model | $R_{+/-}$ | AuNP | Cations | Anions |
| :--- | :---: | :---: | :---: | :---: |
| 30 ligands | 1.00 | 10 | - | - |
|  | 1.00 | 10 | 75 | 75 |
|  | 1.00 | 10 | 150 | 150 |
|  | 1.00 | 10 | 300 | 300 |

The decorated DNA with the 30 -AuNPs is then placed in a new orthorhombic box of side 510 $\AA \times 102 \AA \times 510 \AA$ along the $z$ axis so that the DNA cylinder is at the position $(51 \AA, 51 \AA)$ in the
$x O y$ plane. Then, we select a $30-\mathrm{AuNP}$ adsorbed on the DNA molecule so that this $30-\mathrm{AuNP}$ can be pull out from the surface along the $x$ direction that defines the reaction coordinate. We apply the umbrella sampling method to obtain the PMF profile (in fact desorption) of the 30-AuNP on a DNA in presence of different amount of monovalent salt ions of diameter $3 \AA$ randomly inserted in the simulation box (Table 4.2. We extract the variation of the Helmholtz free energy along $O x$ direction defined as $\Delta F(x)$ from a series of biased MC simulations.


Figure 4.25 - Histogram of the biased distributions of the 30-AuNP along the reaction coordinate $O x$ defined between the DNA center and the 30-AuNP gold core.

For each PMF calculated, biased simulations are performed every $2 \AA$ along a separation distance between the DNA core and the $30-\mathrm{AuNP}$ core ranging from 16 to $96 \AA$. The value of the harmonic constant used for the biasing harmonic potential is $C=5 \mathrm{~kJ} / \mathrm{mol}$ which provides a good overlapping between the biased distributions of independent biased simulations (Figure 4.25). Biased simulations are performed during $2 \times 10^{5} \mathrm{MC}$ cycles at temperature $T=298 \mathrm{~K}$. If an ion is picked, we apply a simple displacement in the simulation box whereas a translation or a rotation can be performed with the same probability if an AuNP is chosen. A collective move is performed when a $30-\mathrm{AuNP}$ with its ionic cloud is either rotated or translated. The $30-\mathrm{AuNP}$ to be pulled out from the DNA can rotate but only translate along the reaction coordinate.

### 4.3.2.2 Result

The PMFs for the adsorption of a $30-\mathrm{AuNP}$ on the DNA already decorated with the other $\times 9$ AuNPs is calculated for various amount of monovalent salt.


Figure 4.26 - A) PMF as a function of the distance between the DNA core and the 30-AuNP core. Cation:anion refers to the number of salt ions in the system. B) Variation of the internal energy U of the system when the $30-\mathrm{AuNP}$ is pushed away from the DNA.

As plotted in Figure 4.26 picture A), the PMFs between the $30-A u N P$ and the DNA are always attractive and the attraction is weakened by increasing the number of ions in the box which induced a higher screening of the interaction between the DNA and the 30-AuNP. Similar trend has been found for calculated PMF with MC simulations and supported by Poisson-Boltzmann theory for a system composed of oppositively charged nanoparticles in presence of their counterions [289]. Simulations are performed at constant temperature $T=298 \mathrm{~K}$ and the PMFs give the variation of Helmhotlz free energy $\Delta F$ between the unbinding and binding state such that $\Delta F=\Delta U-T \Delta S$ where $\Delta U$ is the internal energy (sum of the electrostatic energy and kinetic energy that remain constant in our closed systems) of the system and $T \Delta S$ corresponds to the entropy contribution for the variation of free energy. We plot the variation of internal energy for each window of the different systems on picture B) of Figure 4.26. In absence of salt ions, the contribution to the free energy $\Delta F$ comes from the variation of the electrostatic energy of the system (= variation of the internal energy $U$ of the system, black line of picture B) associated with the displacement of the 30 -AuNP along the reaction coordinate and 30 -AuNPs adsorbed on DNA. When significant amount of salt is present in the system, the variation of internal energy is minor and we deduce that the change of free energy between the unbinding and binding state $\Delta F<0$ comes from the variation of entropy $(-T \Delta S<0 \Leftrightarrow \Delta S>0)$. Hence, when the 30 -AuNP nanoparticle moving along the reaction coordinate approaches the surface of the DNA, there is a release of salt anions adsorbed on the nanoparticle as well as salt cations adsorbed on DNA as shown in Figure 4.27

This ion depletion contributes to an increase of the translational entropy of the system and is the highest for the system with the lowest amount of salt ( $\times 75$ pairs of ion salt) because the fraction of ions released in the box is the highest. Consequently, the adsorption of the $30-\mathrm{AuNP}$ is driven by the increase of entropy in presence of salt ( $\Delta F \approx 280,200$ and $150 \mathrm{~kJ} / \mathrm{mol}$ respectively with 75 , 150 and 300 salt pairs in the system) while in absence of salt, the variation of electrostatic energy explains the adsorption of the $30-\mathrm{AuNP}$ on the DNA ( $\Delta F \approx 650 \mathrm{~kJ} / \mathrm{mol}$ ).

Our results are in qualitative agreement with experiments where adsorption of multivalent cations such as spermidine ${ }^{+3}$ at the DNA surface is associated by the the realease of small monovalent DNA counterions increasing the translational entropy of the system [290].


Figure 4.27 - Entropy-driven exchange between AuNP and ions upon 30-AuNP adsorption.

### 4.3.2.3 Conclusion

We calculated the PMFs associated to the adsorption of a $30-$ AuNP on a DNA decorated with $\times 9$ other 30 -AuNPs (isoelectric point, $R_{+/-}=1.00$ ) with various amount of monovalent salt ions. The calculated PMFs are in qualitative agreement with the PMF between oppositively charged nanoparticles in presence of monovalent ions calculated in a previous work through MC simulations [289]. We demonstrated also that the adsorption of the 30-AuNP on the DNA may be driven by an entropic gain through release of small salt ions. However, we observed that the 30 -AuNP moving along the reaction coordinate did not rotate as soon as it was adsorbed on the DNA. This trend
could be associated to a loss of rotational entropy which would be minor compared to the increase of translational entropy associated with the release of salt ions.

### 4.3.3 Redissolution of AuNPs from a single fixed DNA with excess of salt

In the previous section, we confirmed that the spontaneous adsorption of AuNPs on the DNA is weakened when the amount of monovalent ions is increased due to a lower fraction of released ions that increase the entropy of the system. For a high amount of ions in the box, the free energy associated with the adsorption of AuNP would decrease so that thermal fluctuations dominate the spontaneous adsorption. In other term, an excess of ions my help in desorbing the AuNPs from the DNA. The influence of salt on the interaction of DNA with AuNPs is studied for different salt cation valency and at different concentrations.

### 4.3.3.1 Methods

From the previous set of simulations, we selected a final equilibrated configuration from a simulation performed at $R_{+/-}=1.00$ in presence of the 6 -AuNPs but without the counterions of the DNA and nanoparticles (section 4.3.1). For that case, all the $\times 506$-AuNPs in the system are adsorbed on the DNA. We inserted randomly in the same simulation box containing the DNA and the 6 -AuNPs a salt $\mathrm{XA}_{n}$ composed by a cation and an anion denoted respectively by X and A . The subscript $n$ refer to the number of anions and the valency of the cation.

Table 4.3 - Summary of the different type and amount of salt inserted in the DNA-AuNP systems.

| Salt | Cations | Anions | $C_{s}(\mathrm{mMol} / \mathrm{l})$ |
| :--- | :---: | :---: | :---: |
| XA | 500 | 500 | 6.25 |
|  | 1000 | 1000 | 12.50 |
|  | 1500 | 1500 | 18.75 |
| $\mathrm{XA}_{2}$ | 500 | 1000 | 6.25 |
|  | 1000 | 2000 | 12.50 |
|  | 1500 | 3000 | 18.75 |
| $\mathrm{XA}_{3}$ | 500 | 1500 | 6.25 |
|  | 1000 | 3000 | 12.50 |
|  | 1500 | 4500 | 18.75 |

The evolution of each system presented in Table 4.3 is probed through a NVT run at $T=298$ $K$ of $4 \times 10^{4} \mathrm{MC}$ cycles. The diameter of the inserted salt ions is $d_{+/-}=3 \AA$. The MC moves
consist of a translational move for simple ions and a combination of collective translational and rotational cluster move for an AuNP. We performed some of the analysis that has been detailed in the section 4.3.1 In particular, we will study the adsorption of the different species and their distribution in the vicinity of the DNA.

### 4.3.3.2 Effect of the salt on the adsorption of AuNPs on DNA

We study the effect of salt at different concentrations and different valency for the salt cation on the adsorption of the 6 -AuNP on the DNA. The amount of 6 -AuNPs and cations adsorbed on the DNA as a function of MC cycles are displayed in Figure 4.28. The monovalent salt has a limited effect on the adsorbed 6-AuNP on the DNA. The correlations between monovalent cations are weak at the DNA surface compared to cations of higher valency. From Figure 4.28 top left, it is obvious that monovalent salt AX cannot compete to unbind the 6 -AuNPs from the DNA. Indeed, there is only a partial desorption of the 6 -AuNPs in the range of studied salt concentration. This observation is in agreement with PMF calculated between similar nanoparticles of opposite charge $(\mathrm{Z}=+|12| \mathrm{e})$ in presence of small monovalent cations and anions. Even at $C_{s}=0.3 \mathrm{Mol} / \mathrm{l}$ of ions in the box, the PMF predicts attraction between the two nanoparticles with a variation of Helmholtz free energy of $\approx 4 \mathrm{k}_{B} \mathrm{~T}$ [289]. When the valency of the cation is increased, the number of 6 -AuNPs adsorbed on the DNA progressively drops. For a salt of divalent cation and monovalent anion $\mathrm{XA}_{2}$, around half of the adsorbed 6-AuNP are detached from the DNA molecule as seen in Figure 4.28 middle left. If the valency of the cation is increased to +3 , all the 6 -AuNPs are ejected to the bulk. (Figure 4.28 bottom left). When the ion valency increases, more salt cations are adsorbed on the DNA as observed in Figure 4.28, right column. Indeed, it is shown thanks to PMF calculations that for a given valency of opposite charged nanoparticles $(Z=+|12| e)$, a concentration of 10 $\mathrm{mMol} / \mathrm{l}$ of divalent cations and divalent anions have the same effect as $300 \mathrm{mMol} / \mathrm{l}$ of monovalent cations and anions on the PMF [289].

Along this line, it is expected in our systems that an increase in the cation charge or the salt concentration will weaken the spontaneous adsorption of AuNPs resulting in a competion between AuNP and cations to bind on the DNA. Competition between cations of different valency to bind on DNA has been widely studied. It has been possible to identify by electronic microscopy the ionic species in the vicinity of a DNA fragments [291]. Comparison has been made between the
competitive association of $\mathrm{Mg}^{2+}$ and $\mathrm{Na}^{+}$on a single 24 base-pairs DNA. It has been shown that a much higher concentration of $\mathrm{Na}^{+}$is required to replace half of the $\mathrm{Mg}^{2+}$ atmosphere around the DNA ( $88 \mathrm{mMol} / \mathrm{l}$ of $\mathrm{Na}^{+}$vs $5 \mathrm{mMol} / \mathrm{l}$ of $\mathrm{Mg}^{2+}$ ). Conversely, a concentration of $0.5 \mathrm{mMol} / \mathrm{l}$ of $\mathrm{Mg}^{2+}$ is sufficient to replace half of the $\mathrm{Na}^{+}$atmosphere at concentration of $20 \mathrm{mMol} / \mathrm{l}$. Such results have been later confirmed with MD simulations [292] performed with an atomistic model of DNA.


Figure 4.28 - Left) Adsorbed 6-AuNPs through NVT MC simulations. Right) Adsorbed salt cations through NVT MC simulations.

### 4.3.3.3 Density of cations in the vicinity of the DNA

We performed supplemental analysis to probe the distribution of the species in the vicinity of the DNA when the salt is added in the system and compete with the 6 -AuNPs for adsorption. We used the equations 4.344 .35 and 4.36 to construct the helical projection of the cations on the DNA plane for concentration of salt $C_{s}=18.75 \mathrm{mMol} / \mathrm{l}$ equivalent to 1500 cations in the system. The results are presented in Figure 4.29


Figure 4.29 - Helical projection of the cations on the $x O y$ plane. Left) Helical projection for monovalent cations for the salt AX. Middle) Helical projection for divalent cations for the salt $\mathrm{AX}_{2}$. Right) Helical projection for trivalent cations for the salt $\mathrm{AX}_{3}$. The helical projection is given in number of particles per $\AA^{2}$.

When the valency of the cation is increased the density of cations in the vicinity of the DNA grows as well. When the cation is monovalent, the minor groove is more populated as it was previously mentioned in section 4.2.1.2 We notice that it is also unlikely that the cations will populate the major groove since this region is already occupied by the 6 -AuNPs. When the valency of the cation is increased, the minor groove and the strands are mostly occupied.

### 4.3.3.4 Conclusion

The desorption of 6-AuNPs from a rigid fixed DNA molecule has been studied in presence of various amount of salt of different valencies. We have shown that a high salt concentration as well as high cation valency promote the desorption of the 6 -AuNPs from the DNA. Salt weaken the spontaneous attraction of 6-AuNP to DNA as supported by the series of PMFs calculated in section 4.3 .2 by allowing thermal fluctuations to detach the 6 -AuNPs from the DNA major groove. In addition, correlations induce the adsorption of multivalent cations in the minor groove and on the strands. This effect might also contribute to repel the 6 -AuNPs from the DNA.

### 4.3.4 Effective force between a pair of DNA

We considered in the previous section the interaction of AuNPs with a single DNA. Although we studied the distribution/interaction of AuNPs of different charges with a fixed MAM DNA model, from now we are interested in the effect of AuNPs on the interaction between two DNA molecules.

We study the effective force between two DNA molecules in presence of AuNPs in the same manner as the forces calculated in section 4.2.2 The effective force will be calculated for AuNPs of various charges and different amount of small monovalent ions and will give insight about a possible condensation of DNA by the AuNPs.

### 4.3.4.1 Methods

We consider two DNA in the same configuration as in section 4.2.2.1 The only difference is that the box size is switched from $L=102 \AA$ to $L=204 \AA$. Despite the higher charge carried by the AuNPs, we still want to have a significant amount of AuNPs in the box to perform proper sampling of the system within an affordable computation time. Consequently, each DNA in the simulation box contains only 60 basis pairs, which is equivalent to 6 helical turns. In addition to the fixed DNA placed in the box at positions $P_{1}\left(x_{1}, \frac{L}{2}\right)$ and $P_{2}\left(x_{2}, \frac{L}{2}\right)$ corresponding to a distance between the DNA molecule of $\ell=\left|x_{1}-x_{2}\right|$, we randomly insert monovalent cations, anions and AuNPs. We will consider only AuNPs with 6 or 12 ligands. The table 4.4 describes the different investigated systems.

Remark: We will not consider systems with 30-AuNPs because for the range of considered $R_{+/-}=0.50-1.50$, only $\times 4$-12 30-AuNPs are present in the system and we could not obtain a proper sampling for the configurations of 30 -AuNPs around the DNA in a reasonable time with our direct MC simulations.

The ions have a diameter of $3 \AA$ and each system is equilibrated during a $N V T$ run of $5 \times 10^{4}$ Monte Carlo cycles. Then, we perform an additional production run of $10^{6}$ cycles to extract the effective force acting between the pair of DNA. At each MC step a trial translational move is performed if an ion is randomly chosen while a translational or a rotational collective move is performed to displace an AuNP and its surrounding cloud of ions within a radius of $R_{\text {Gold }}=15$

Table 4.4 - Composition of simulated systems. Each line represents a simulation with a given number of AuNPs and monovalent ions.

| AuNP model | $R_{+/-}$ | AuNP | Cations $\left(N_{c_{+}}\right)$ | Anions $\left(N_{c_{-}}\right)$ |
| :--- | :---: | :---: | :---: | :---: |
| 6 ligands | 0.50 | 20 | 240 | 120 |
|  | 1.00 | 40 | - | - |
|  | 1.00 | 40 | 240 | 240 |
|  | 1.50 | 60 | 240 | 360 |
| 12 ligands | 0.50 | 10 | 240 | 120 |
|  | 1.00 | 20 | - | - |
|  | 1.00 | 20 | 240 | 240 |
|  | 1.50 | 30 | 240 | 360 |

$\AA$ around the gold core. The translational move is characterized by a maximum displacement in a box of side $0.10 \times L$. We denote respectively by $c_{+}$and $c_{-}$the cations and the anions, and by $n_{+}$the positive charges carried by the ligands of the nanoparticles. The total number of cations is $N_{c_{+}}$and the total number of anions is $N_{c_{-}}$while $N_{n_{+}}$is the total number of ligands in the system. We recall that the effective force $\mathbf{F}$ acting on one DNA is the sum of four contributions:

$$
\begin{equation*}
\mathbf{F}=\mathbf{F}_{1}+\mathbf{F}_{2} \tag{4.38}
\end{equation*}
$$

The term $\mathbf{F}_{1}$ is the force exerted by the phosphate groups of one DNA on the phosphate groups of the other DNA given by equation 4.21 of section 4.2.2.1

The second term $\mathbf{F}_{2}$ represents the Coulomb interaction between the cations $l$ and the phosphate groups $k$ of the DNA:

$$
\begin{equation*}
\mathbf{F}_{2}=-\sum_{k=1}^{N_{p}}\left(\left\langle\sum_{l=1}^{N_{c_{+}}} \vec{\nabla}_{\mathbf{r}_{k}^{p}} V_{p c_{+}}\left(\left|\mathbf{r}_{k}^{p}-\mathbf{r}_{l}^{c_{+}}\right|\right)\right\rangle\right) \tag{4.39}
\end{equation*}
$$

where $\mathbf{r}_{j}^{i}$ is the position of the different charges in the box. The index $j$ denotes the $\mathrm{j}^{\text {th }}$ particle of species $i$.

The third term $\mathbf{F}_{3}$ represents the Coulomb interactions between the anions $l$ and the phosphate groups $k$ of the DNA:

$$
\begin{equation*}
\mathbf{F}_{3}=-\sum_{k=1}^{N_{p}}\left(\left\langle\sum_{l=1}^{N_{c_{-}}} \vec{\nabla}_{\mathbf{r}_{k}^{p}} V_{p c_{-}}\left(\left|\mathbf{r}_{k}^{p}-\mathbf{r}_{l}^{c_{-}}\right|\right)\right\rangle\right) \tag{4.40}
\end{equation*}
$$

The fourth term $\mathbf{F}_{4}$ represents the Coulomb interactions between the positive charges $l$ carried by AuNPs and the phosphate groups $k$ of the DNA:

$$
\begin{equation*}
\mathbf{F}_{4}=-\sum_{k=1}^{N_{p}}\left(\left\langle\sum_{l=1}^{N_{n_{+}}} \vec{\nabla}_{\mathbf{r}_{k}^{p}} V_{p n_{+}}\left(\left|\mathbf{r}_{k}^{p}-\mathbf{r}_{l}^{n_{+}}\right|\right)\right\rangle\right) \tag{4.41}
\end{equation*}
$$

The forces are projected along the $O x$ axis on which are disposed the DNA. We perform a MC simulation and calculate the effective force every $6 \AA$ for a separation distance $\ell$ between the two parallel DNA ranging from $24 \AA$ to $96 \AA$.


Figure 4.30 - Typical snapshots of a simulation for different DNA-DNA separation distance $\ell$. The cations are represented as green small beads while the anions are represented by grey small beads. Left) Separation distance of $42 \AA$. Middle) Separation distance of $30 \AA$. Right) Separation distance of $24 \AA$.

Remark: In practice, the effective force calculation is done "on the fly" for the 2 DNA molecules. This trick allows to sample two times the effective force for better statistics.

### 4.3.4.2 Results

The effective force exerted between the pair of DNA in presence of various amount of ions and AuNPs is plotted in Figure 4.31 A positive (respectively negative) force indicates attraction (respectively repulsion) between the pair of DNA. The magnitude of the effective force is $\times 4$ higher for systems containing 12 -AuNPs than those containing 6 -AuNPs because they carry a highest charge resulting in a highest attraction.


Figure 4.31 - Effective force between the DNA molecules calculated for the systems presented in Table 4.4 The force is normalized according to $F_{o}=\frac{k_{B} T}{P}$ where $P=34 \AA$ is the DNA pitch length. Top) Effective force calculated in presence of 6-AuNPs. Bottom) Effective force calculated in presence of $12-A u N P s$.

In order to further explain the profile of the effective forces presented in Figure 4.31 we split
the effective force into each species contribution: the electrostatic force induced by the other DNA phosphate groups, the electrostatic force induced by cations and anions as well as those induced by the AuNPs (Figure 4.32). The values of the effective force and each species contribution as well as error calculations are given in Appendix C.3

At a large distance between the DNAs $(\ell \sim 72-96 \AA)$, the effective force is negligeable due to the AuNPs and free ions that screen the mutual DNA-DNA electrostatic repulsion. When the DNA molecules are approaching each other, there is a repulsive force for a separation distance of $\ell=66-72 \AA$ to $\ell=42-48 \AA$ (excepted for the systems at $R_{+/-}=0.50$ ) where an unstable equilibrium position appears for systems characterized by $R_{+/-}=1.00-1.50$ (green circles at $\ell=$ 42-48 $\AA$ (Figure 4.31). The unstable equilibrium position corresponds to the onset of attraction between the pair of DNA.

For closer DNA-DNA distance, the effective force is positive and maximum for a separation distance of $\ell=30-36 \AA$. At such distance, the AuNPs preferentially re-arrange to intercalate themselves between the DNA molecules, resulting in creation of bridges and attraction between the 2 DNA molecules as illustrated in Figure 4.30, middle picture.

At shortest separation distance $(\ell=24 \AA)$ the effective force drops to negative values for all cases which corresponds to repulsion between the DNA molecules excepted for the effective force calculated with 12 -AuNPs $\left(R_{+/-}=0.50\right)$. In the latter case, fewers 12 -AuNPs are present in the box and more DNA monovalent counterions participate in the attraction between the DNAs (blue curve of picture A), Figure 4.32. The DNA are close enough to impede intercalation of AuNPs as shown in Figure 4.30, right picture. The AuNPs still bind with the two DNA molecules by intercaling in the strand formed when the DNA are very closed to each other.

There is a stable equilibrium position between the DNA molecules comprised between $\ell=24$ $\AA$ and $\ell=30 \AA$ which would corresponds to a close-packed arrangement of AuNPs intercalated between the DNA molecules (blue circles, Figure 4.31. Further effective force calculation are required to determine with precision this equilibrium distance. It is also likely that in a direct simulation, the DNA molecules decorated with the AuNPs would be free to approach each other and assemble so that the distance separating the DNA would correspond to the equilibrium distance. However, For a low number of AuNPs in the box $\left(R_{+/-}=0.50\right)$, the effective force is always positive which means that attraction between the decorated DNA with AuNPs happen at larger


Figure 4.32 - Contribution of every species to the effective force for systems containing 12-AuNPs. In red, we plotted the intensity of the electrostatic force between the DNA molecules for comparison. The force between the AuNPs and the DNA is represented in orange while the forces caused by the cations and the anions are represented respectively in blue and green.
distance than in system at $R_{+/-}=1.00-1.50$ where attraction between the DNA arises much $\operatorname{closer}(\ell=42-48 \AA)$.

The contribution of each species to the effective force are given in Figure 4.32 for systems with 12-AuNPs. The electrostatic force between the DNA and AuNPs is the main attractive contribution to the effective force and is maximum around $\ell=30 \AA$. For small separation distance, it is clear that the AuNPs cannot intercalate between the pair of DNA but instead bind in the strand made by the gathering of the DNA so that the force drops for all cases (orange curves). This allows the cations to populate more the region between the DNA molecules given that the interaction (blue curves) increases at this range.

The general trend observed for the effective force intensity around $\ell=30 \AA$ corresponds to an increase in the order $R+/-=1.50<R+/-=1.00$ (with ions) $<R+/-=0.50$, as observed in Figure 4.31 At this distance, the contribution of the AuNPs to the effective force is respectively $209.8 F_{o}, 206.2 F_{o}$ and $211.7 F_{o}$ for $R_{+/-}=0.50,1.00,1.50$. The contribution of the cations to the effective force is respectively $15.3 F_{o}, 3.3 F_{o}$ and $-3.7 F_{o}$ for $R_{+/-}=0.50,1.00,1.50$. The AuNPs segregate cations from region between the DNA molecules. We notice also that cations destabilize the bridges made by the AuNPs around $\ell=30 \AA$ since at $R_{+/-}=1.00$, the effective force induced by the AuNPs in presence of small ions is lower than in absence of small ions.

At higher separation distance $(\ell=42-72 \AA)$, the effective force is negative for $R_{+/-}=1.00$ - 1.50 (Figure 4.31. For systems characterized by $R_{+/-}=0.50$ and 1.00 , the cations compete with the AuNPs in the region between the 2 DNAs as shown in plots A and C of Figure 4.32 The cations populate the region between the DNAs at the expense of the AuNPs as observed by the fall of the electrostatic force (in orange) induced by the AuNPs and the growth of the force (in blue) induced by the cations. At $R_{+/-}=1.00$ and in absence of small ions, the force induced by the AuNPs is weaker than the repulsion between the DNA because the AuNPs spread on the DNAs due to their mutual repulsion resulting in a lower electrostatic attraction (plot B in Figure 4.32). In excess of AuNPs, e.g. at $R_{+/-}=1.50$, the presence of anions (green line) at the surface of the highly charged 12-AuNPs results in an overall negative force and thus repulsion between DNA molecules (plot D, Figure 4.32).

### 4.3.4.3 Conclusion

The effective DNA-DNA interaction has been calculated in presence of AuNPs and explicit ions. We have shown that AuNPs induce DNA aggregation through bridges since the effective force is
positive and maximum at short distance $(\ell=30 \AA)$ and is associated with existence of a stable equilibrium in the range of $\ell=24-30 \AA$. The DNA monovalent positive counterions participate to the DNA mutual attraction at $R_{+/-}=0.50$. However, for $R_{+/-}=1.00$, a higher amount of AuNPs are present in the system and compete with cations to intercalate between the DNA molecules. At excess of AuNPs $\left(R_{+/-}=1.50\right)$ the negative co-ions (the counterions of AuNPs) destabilize the DNA pair formation by adsorbing on the AuNPs located in the region between the DNA molecules.

### 4.3.5 Osmotic pressure calculation in hexagonal and square DNA phases

In the previous section, we have calculated the effective force mediated by AuNPs between a pair of aligned DNA molecules and conclude that it is possible to induce attraction through bridges made by positive AuNPs. It has been shown that generic systems of oppositively charged "rod-like" and "sphere-like" objects are able to form bundle driven mostly by electrostatics interactions [62, 280]. For instance, positive multivalent lyzozymes can condensate F-actin by creating "bridging" sites. Along this line, we probe the possibility to form a DNA bundle mediated by AuNPs by calculating the net osmotic pressure. Given that the coarse-grained DNA model is completely rigid, we will study the mechanical stability of a bundle where the DNA molecules form a hexagonal or square lattice. The osmotic pressure calculation should give some hint about the preferred crystal structure adopted by the DNA molecules in presence of AuNPs. The effect of monovalent salt on the bundle stability will be also considered.

### 4.3.5.1 Methods

The DNA bundle is represented as an assembly of fixed DNA molecules in either square or hexagonal crystal packing. Each simulation box for both cases contains 4 DNA molecules packed with their long axis along the z-axis. Each DNA is effectively an infinite molecule due to application of PBCs and we thus simulate an infinite DNA bundle that model a phase separation between a DNA precipitate and the supernatant phase. If we denote by $\ell$ the distance separation between DNA molecules, the orthorhombic computational box sides are for a square lattice:

$$
\left\{\begin{array}{c}
L_{x}=2 \ell  \tag{4.42}\\
L_{y}=2 \ell \\
L_{z}=3 P
\end{array}\right.
$$

where P represents one DNA helical turn. In a similar way, each side of the orthorhombic computational box when DNA molecules are forming a hexagonal lattice (see section 4.2.3) is defined as:

$$
\left\{\begin{array}{c}
L_{x}=2 \ell  \tag{4.45}\\
L_{y}=\sqrt{3} \ell \\
L_{z}=3 P
\end{array}\right.
$$

The free moities that evolve around the fixed DNA molecules are the AuNPs and the monovalent positive or negative salt ions. We calculate the osmotic pressure for bundle of DNA in presence of 6 -AuNPs or 12 -AuNPs playing the role of compaction agents. The osmotic pressure is calculated for each lattice type at different DNA-DNA spacing $\ell$.

Remark: The bundle stability will be studied at the isoelectric point $R_{+/-}=1.00$ where we can make the reasonable assumption that all the AuNPs remain in the DNA bundle and do not escape in the bulk (supernatant phase). Such a hypothesis can not be made for the free ions.

In presence of free ions, the procedure to calculate the osmotic pressure is more complicated that for the case presented in section 4.2.3. In practice, the DNA bundle can exchange ions with the supernatant phase. This effect should be taken into account for the calculation of the osmotic pressure. We detail the procedure below.

- At a fixed salt concentration in the bundle and for a given DNA-DNA spacing $\ell$, we performed a standard NVT MC simulation with an equilibration phase of $5 \times 10^{3} \mathrm{MC}$ cycles and then gather statistics to calculate quantities of interest during $\times 10^{6} \mathrm{MC}$ cycles.
- We calculate the osmotic pressure in the bundle $\Pi_{\text {bundle }}$ and the chemical potential of the salt by using the Widom insertion technique as described in section 3.2.5
- Setting the chemical potential of salt in the bulk to its value obtained previously in ordered DNA phase, we perform a grand canonical Monte Carlo simulations to mimic the bulk containing salt in equilibrium with the ordered DNA phase [293] through $20 \times 10^{6} \mathrm{MC}$ steps. For each step, there is equal probability to displace a cation/anion in the box or insert/delete a couple of cation/anion to preserve electroneutrality
- The relative osmotic pressure [281 is defined by:

$$
\begin{equation*}
\Pi_{r e l}(\ell)=\Pi_{\text {bundle }}(\ell)-\Pi_{\text {bulk }}(\ell) \tag{4.48}
\end{equation*}
$$

A positive value of $\Pi_{r e l}$ indicates a mechanically unstable DNA bundle whereas a null or negative value of $\Pi_{r e l}$ corresponds to a mechanically stable/metastable or contracting DNA bundle.

Remark: In the case of a simulation without added ions, the bulk osmotic pressure vanishes $\left(\Pi_{b u l k}=0 \mathrm{~Pa}\right)$ and the bundle stability is given from the osmotic pressure in the bundle.

We summarize in Table 4.5 the different systems to be simulated. Snapshots of the system with 6-AuNPs without added ions are presented in Figure 4.33

Remark: We do not consider bundle stability in presence of 30-AuNPs. Only 8 nanoparticles are needed to neutralize the DNA phosphate charges, which is too low to have a reasonable statistics when osmotic pressure is calculated. It is possible to

Table 4.5 - Summary of the different species present in the DNA bundle for which the osmotic pressure is calculated at a given concentration of monovalent salt $C_{s}$.

| AuNP model | $R_{+/-}$ | AuNP | $C_{s}(\mathrm{mMol} / \mathrm{l})$ |
| :--- | :---: | :---: | :---: |
| 6 ligands | 1.00 | 40 | - |
|  | 1.00 | 40 | 30 |
|  | 1.00 | 40 | 60 |
|  | 1.00 | 40 | 120 |
| 12 ligands | 1.00 | 20 | - |
|  | 1.00 | 20 | 30 |
|  | 1.00 | 20 | 60 |
|  | 1.00 | 20 | 120 |

remove this constraint either by considering a bigger lattice with more DNA molecules or by extending the number of cycles of the MC simulation.

The salt ions have a diameter of $3 \AA$ and for each system presented in Table 4.5 a set of canonical NVT simulation in the bundle is performed for discrete values $\ell$ of DNA-DNA distance. At each MC step a trial translational move is performed if an ion is randomly chosen. A translational or rotational collective move is performed to displace an AuNP and its surrounding ions in a radius of $R_{\text {Gold }}=15 \AA$ around the gold core.


Figure 4.33 - A) Snapshots of hexagonal DNA lattice with a view in the $O y$ (left) and $O z$ (right) directions. B) Snapshots of square DNA lattice with a view in the $O y$ (left) and $O z$ (right) directions.

### 4.3.5.2 Results

The relative osmotic pressure $\Pi_{r e f}$ for each system of Table 4.5 is plotted in Figure 4.34 The values of the osmotic pressure and error calculations are given in Appendix C.4 The calculation of the osmotic pressure is done from a minimum DNA-DNA distance of $\ell=32 \AA$. For smaller values of $\ell$, it was impossible to insert properly or displace the AuNPs in the DNA lattice. We focused first on the systems without added salt. The relative osmotic pressure for the DNA hexagonal or square bundle containing 6 -AuNPs is always negative, indicating that the bundle is always contracting. At large distance $\ell$, the DNA molecules are far away from each other and their electrostatic interaction also vanishes. When the distance $\ell$ is reduced, the osmotic pressure increased as the AuNPs make bridge between the DNA molecules. The same conclusions hold for lattices of DNA which contain the 12 -AuNPs without added salt, at the exception that at a distance $\ell \approx 57 \AA$, the osmotic pressure is constant but slightly positive. This effect is due to the mutual repulsion of the highly charged 12-AuNP located on the same DNA and resulting in higher fluctuations of the osmotic pressure. Our observations are in qualitative agreement with osmotic pressure calculation reported in the literature for hexagonal systems of negatively-charged rodlike F-actin filaments condensed by lyzozymes proteins of charge $+5|\mathrm{e}|$ or $+9|\mathrm{e}|$ [279], where a negative osmotic pressure indicates a bundle formation of actin-Lyzozyme. However, the pairwise hard core potential used in our MC simulations prevents the investigation of bundle stability under a certain step of the DNA bundle because of interpenetration between DNA and gold nanoparticles. In ref. [279], a Lennard-Jones potential is used to model short range interaction instead of hard core pairwise potential such that at short lattice step, there is a strong repulsion between F-actin filaments and lyzozymes. It results in an increase of the osmotic pressure allowing the determination of a stable equilibrium that verifies $\Pi_{r e f}=0$ and $\frac{\partial \Pi_{r e f}}{\partial V}<0$.

When salt is added into the bundle, the osmotic pressure is shifted towards positive values. Addition of monovalent salt destabilizes the bundle and we notice that the salt bulk osmotic pressure $\Pi_{\text {bulk }}$ is not sufficient to compensate the effect of the salt present in the DNA lattice, resulting in a higher osmotic pressure $\Pi_{\text {bundle }}$ even for a concentration of $C_{s}=30 \mathrm{mMol} / \mathrm{l}$. For a high lattice step, the relative osmotic pressure remain positive but decreases because of the large amount of salt ion in the bundle. In presence of salt, we notice that there is an equilibrium


Figure 4.34 - Difference of osmotic pressure $\Pi_{r e l}$ for systems presented in Table 4.5 where DNA molecules can adopt either a square or hexagonal lattice.
position $\ell_{e q}$ when $\Pi_{r e l}=0$ atm., but this position is unstable because $\frac{\partial \Pi_{r e f}}{\partial V}>0$. In other words, an infinitesimal increase of the DNA-DNA distance $\ell$ would result in an increase of the relative osmotic pressure as well and the bundle will become unstable. The unstable equilibrium position separate two regions; for $\ell<\ell_{e q}$, the osmotic pressure is negative and there is spontaneous assembly of DNA and AuNPs while for $\ell>\ell_{e q}$, the higher amount of salt impede the formation of the bundle. The higher the salt concentration is in the bundle, the lower $\ell_{e q}$ is.

It is instructive to link the repartition of the AuNPs and the salt in the bundle as a function of the DNA-DNA separation distance $\ell$. Along this line, we calculated the ionic densities in the hexagonal DNA bundle (Figure 4.35 condensed by 12 -AuNPs for $\ell=32 \AA$ ( $\Pi_{r e l}=-6.78 \mathrm{~atm}$.) and $\ell=40 \AA\left(\Pi_{r e l}=0.93 \mathrm{~atm}.\right)$. The 12 -AuNPs adopt a threefold coordination with DNA (top picture, left of Figure 4.35 when the bundle is contracted at its minimal DNA interdistance $\ell=32 \AA$. In a similar way, a stable hexagonal F-actin bundle condensed with lyzozymes of charge $+9|\mathrm{e}|$ adopting a three-fold conformation with the F-actin has been investigated with MD simulations [279]. When the lattice expands, 12 -AuNPs adopt a twofold coordination with the DNA (top picture, right
of Figure 4.35 which sustain that the structure is dominated by electrostatic interactions, as previously mentioned through MD simulations performed on actin-lyzozyme hexagonal bundle 62.

Further analysis of the salt repartition in the bundle give insight about the influence of salt ions on the bundle stability. The cation distribution reveals that cations are mostly localized around the DNA molecules (middle picture, left of Figure 4.35) and when the bundle expands, the number of cations is increasing in the simulation box because the concentration $\mathrm{c}_{\text {salt }}$ is fixed. The cations rearrange also around the DNAs as observed by the higher intensity of the cation distribution (middle picture, right of Figure 4.35). The destabilization of the bundle at the DNADNA distance $\ell=40 \AA\left(\Pi_{r e l}=0.93 \mathrm{~atm}\right.$.) can be explained by the cations that induce screening of the electrostatic interactions between DNA and the 12-AuNPs. On the other way, the anion distribution is rather homogeneous at shortest DNA-DNA distance (top picture, right of Figure 4.35 but when the bundle expands, the anions are relocalized at the positions where 12 -AuNPs formed three-fold coordination with the DNA molecules.

### 4.3.5.3 Conclusion

The osmotic pressure in hexagonal and square lattice of aligned DNA in presence of AuNPs has been calculated for different concentrations of monovalent salt. Without added salt, it has been shown that a self-assembly of DNA-gold nanoparticles can appear in the frame of our coarsegrained model of rigid DNA and gold nanoparticles. Such self-assembly is electrostatically driven and negative rodlike DNA will condense with positively charged gold nanoparticles in order to minimize the overall electrostatic interaction.

When salt is added, the bundle is found to be less stable irrespective of the charge of the nanoparticles and the DNA lattice geometry. For large values of DNA-DNA interdistance $\ell$ and at any of the salt concentration investigated, the difference between the osmotic pressure in the bundle and in the bulk is positive and reveals that a self-assembly will not emerge spontaneously in presence of salt for DNA molecules. A possible reason for such a loss of stability may come from the cations which concentrate around the DNA molecules and weaken the overall electrostatic interaction by disturbing the brigdes formed between the DNA and the nanoparticles.


Figure 4.35 - Projection of the ionic densities in the $x O y$ mid-plane for the hexagonal bundle of DNA condensed with 12 -AuNPs. The distributions are calculated for a salt concentration of $\mathrm{c}_{\text {salt }}$ $=120 \mathrm{mMol} / \mathrm{l}$. Left) Ionic densities in the hexagonal DNA bundle for a DNA interdistance $\ell=$ $32 \AA$. Right) Ionic densities in the hexagonal DNA bundle for a DNA interdistance $\ell=40 \AA$. We display the salt anion, salt cation and AuNP densities in particle $/ \AA^{2}$.

Our results are consistent with experimental observations for which well-ordered self-assembled structures made of cationic gold nanoparticles and different rodlike rigid polymers like DNAorigami [146] or tobacco mosaic virus [294] can form through a progressive decrease of monovalent salt $(\mathrm{NaCl})$ initially present in excess in the solution.

Unfortunately, we cannot determine with our calculations whether a square or hexagonal DNA bundle would emerge from the self-assembly process of DNA and AuNP as we found similar osmotic pressure profile for both cases. It would be required to perform direct simulations with free DNAs molecules allowed to translate or rotate in the simulation box to investigate the structure that
can result from self-assembly process. However, One has to remember that DNA is a flexible polyelectrolyte chain so that in experiments involving self-assembly processes of DNA and AuNPs [7] [143, the bundle do not display crystallic arrangement. Our model with rigid DNA is not realistic but allows to derive important results regarding possible self-assembly process between DNA and AuNPs.

## CHAPTER 5

## Self-assembly in PEDOT:PSS solution by addition of ionic liquids for conductivity enhancement.

## Contents

5.1 Choice of molecular model ..... 200
5.1.1 Atomistic models for PEDOT:PSS ..... 200
5.1.2 Atomistic models for ionic liquids. ..... 201
5.2 Morphology of aqueous solution of PEDOT:PSS ..... 202
5.2.1 Methods ..... 203
5.2.2 Morphology of aqueous PEDOT:PSS complexes - small systems case ..... 205
5.2.3 Morphology of aqueous PEDOT:PSS complexes - larger systems case ..... 207
5.2.4 Conclusion ..... 209
5.3 Prediction of ion exchange between PEDOT:PSS/IL ..... 209
5.3 .1 Methods ..... 211
5.3.2 Ion exchange free energy: PMF calculations ..... 212
5.3.3 Solvation of IL anions in US simulations. ..... 218
5.3.4 Conclusion ..... 221
5.4 Ionic exchange in mixed solutions: varying the IL anion ..... 221
5.4.1 Methods ..... 223
5.4.2 Morphology of mixed solution - small system cases ..... 224
5.4.3 Conclusion ..... 235

In this chapter, we aim to probe the ionic exchange between PEDOT:PSS and ionic liquid (IL) EMIM:X in explicit water through molecular dynamics (MD) simulations performed for different atomistic models of PEDOT:PSS. First, we explore the morphology of aqueous PEDOT:PSS solutions. Then, ion exchange between PEDOT:PSS and EMIM:X is predicted through free energy
calculations using the umbrella sampling technique and we confirm design principles established through DFT calculations [201] that the IL anion X must satisfy in order to perform the most efficient ionic exchange. Finally, we confirm such ionic exchange at a larger scale by performing direct (i.e. non biased) MD simulations by inserting the IL and relaxing aqueous PEDOT:PSS solutions [11].

The electronic degrees of freedom are not taken into account in the latter approach and thus the electronic transport properties of PEDOT:PSS cannot be determined directly through classical MD simulations. However, we will link PEDOT:PSS morphology obtained from MD simulations at the nanoscale level to the change of morphology associated with improvement of electric transport properties of PEDOT:PSS at the macroscopic scale (see chapter 2 section 2.2.3.2.

### 5.1 Choice of molecular model

In experiments (see chapter 2 section 2.2.3.2, the aqueous solutions of PEDOT:PSS to be treated with IL [11, 12] contain longer PSS chains than PEDOT chains (section 2.2.2.2. In general, it is challenging to simulate long polymers at large scale with non biased MD simulations due to slow relaxation of polymer conformations [295]. We introduced different atomistic models of PEDOT and PSS to be used in subsequent simulations that avoid this drawback, at the expense of a fully realistic representation of the studied systems.

### 5.1.1 Atomistic models for PEDOT:PSS

The minimal model chosen to represent the PEDOT polymer is a trimer of EDOT (tri-EDOT) according to previous reports [296, 163] that only one positive charge is carried over at least three EDOT units of p-doped PEDOT while the minimal model for PSS is p-toluenesulfonate (PTS, the monomer unit of PSS). A representation of the minimal model of PEDOT:PSS denoted by tri-EDOT:PTS is shown in Figure 5.1

We present the OPLS-AA force field parameters describing atomic interactions for the triEDOT:PTS model in Appendix D. 1 This model is not very realistic but is a suitable first step model to capture at a larger scale the ion exchange predicted by DFT calculations using the same minimal representation of PEDOT:PSS [201, 11]. Larger-size PEDOT:PSS models are proposed


Figure 5.1 - Minimal model of PEDOT:PSS used in the MD simulations. Top) Tri-EDOT cation carrying a charge $+|\mathrm{e}|$. Bottom) PTS anion carrying a charge $-|\mathrm{e}|$. Color code for atoms: Hydrogen(white), carbon(cyan), sulfur(yellow) and oxygen(red).
(Figure 5.2) to probe the influence and consequences of longer chains on the ionic exchange by using a 6 -unit EDOT oligomer (6EDOT) and substituting the PTS monomer by a 16 -unit SS oligomer (16SS).


Figure 5.2 - Larger-size model of PEDOT:PSS used in the MD simulations. Top) 6EDOT cation carrying a charge $+2|\mathrm{e}|$. Bottom) PTS anion carrying a charge $-16|\mathrm{e}|$. Color code for atoms: Hydrogen(white), carbon(cyan), sulfur(yellow) and oxygen(red).

We perform MD simulations for different combinations of PEDOT or PSS models:

- A first set of simulations involving the tri-EDOT:PTS model,
- a second set of simulations involving the tri-EDOT:16SS model,
- a last set of simulations involving the 6EDOT:16SS model.


### 5.1.2 Atomistic models for ionic liquids

We will discuss the effect of EMIM:X IL on the morphology of PEDOT:PSS with MD simulations by using an atomistic model for IL (Figure 5.3). The IL anion designed by X and used
in experiments can be chloride ion $(\mathrm{Cl})$, ethylene sulfonate (ES), tricyanomethanide ( TCM ) or tetracyanoborate (TCB). An additional hypothetical anion to be involved in our simulations and predicted to be a promising candidate for conductivity enhancement is heptacyanocyclopentanide (HCCP, see chapter 2 section 2.2.3.2). The force-field parameters for IL are given in Appendix D. 2 and D. 3


Figure 5.3 - Imidazolium based ionic liquid EMIM:X.

### 5.2 Morphology of aqueous solution of PEDOT:PSS

We first probe morphology of aqueous solutions of PEDOT:PSS without added ionic liquids. It has been recognized that the morphology of PEDOT:PSS is one of the major factor that affect its transport properties [297, 298, 299]. A structural characterization of PEDOT:PSS films using grazing incidence wide angle X-ray scattering (GIWAXS) combined with transmission electron microscopy reveal highly crystalline domains of 3-10 stacked chains surrounded by amorphous matrix [300, 167. However, these characterizations cannot reveal an exact microscopic picture of the PEDOT:PSS morphology such as position of the counterions or local structure at the nano-scale level. As a consequence, MD simulations of aqueous PEDOT:PSS solutions aimed to provide possible answers given that conjectures about the material morphology are based on indirect evidences based on optical and transport measurements (see chapter 2 section 2.2.3.2).

Based on the atomistic MD simulations of PEDOT:PTS previously performed by FrancoGonzalez and Zozoulenko [301], we aim to simulate aqueous solutions containing different models of PEDOT:PSS (section 5.1.1. MD simulations of aqueous PEDOT:PSS solutions represent a required step to further study the influence of IL on the equilibrated PEDOT:PSS systems.

### 5.2.1 Methods

Each system is prepared by inserting PEDOT and PSS or (PTS) at random positions and orientations in a cubic simulation box of 6 nm side length corresponding to a size larger than each oligomer size, followed by solvation in water. Each water molecule is described with the SPC/E model [302. We apply the steepest descent algorithm on each system containing 20,000 atoms to avoid steric clashes and equilibriate systems for 30 -ns NVT run at 293 K using the Berendsen thermostat. Then, $30-\mathrm{ns}$ NPT run is performed at 1 atm using the Parrinello-Rahman barostat and at $293 K$ using the Nose-Hoover thermostat to achieve the equilibration and perform analysis on the resulted morphology. Most of the simulation details and OPLS-AA FF parameters (Appendix D.1) are taken from the previous work of Franco-Gonzalez and Zozoulenko [301] excepted the atomic charge parameters obtained from electrostatic-potential-fitted (ESP) charges through DFT calculations performed on isolated moieties [201]. Long-range electrostatic interactions are handled with the particle mesh Ewald method as implemented in GROMACS MD package.

Table 5.1 - Different models of PEDOT:PSS in solution with the number of each moieties inserted in a cubic simulation box of 6 nm sidelength.

| Model of PEDOT:PSS | $\mathrm{c}($ PEDOT:PSS in mol/l) |
| :--- | :---: |
| tri-EDOT:PTS | $12: 12(0.10)$ |
| tri-EDOT:16SS | $16: 1(0.13)$ |
| 6EDOT:16SS | $16: 2(0.13)$ |

The different models of PEDOT:PSS and the number of PEDOT and PSS units inserted in the simulation box are displayed in Table 5.1. We perform similar simulations for a larger cubic box of 12 nm in side length involving the same models of PEDOT:PSS (Table 5.2). In these cases $60-\mathrm{ns}$ NVT and $60-\mathrm{ns}$ NPT runs are performed to equilibrate the systems.

Table 5.2 - Different models of PEDOT:PSS in solution with the number of each moieties inserted in a cubic simulation box of 12 nm sidelength.

| Model of PEDOT:PSS | PEDOT:PSS $(\mathrm{c}$ in $\mathrm{mol} / \mathrm{l})$ |
| :--- | :---: |
| tri-EDOT:PTS | $96: 96(0.10)$ |
| tri-EDOT:16SS | $96: 6(0.10)$ |
| 6EDOT:16SS | $48: 6(0.05)$ |

Remark: Commercial aqueous solution of PEDOT:PSS with a ratio PEDOT/PSS of 1:2.5 has a mass fraction of $1.3 \%$ wt equivalent to concentration of $\approx 10^{-2} \mathrm{Mol} / \mathrm{l}$.

It is not possible to reproduce such concentration for our PEDOT:PSS systems given that system size will be rather large and equilibration will be difficult to achieve. Therefore, we decide to work instead at concentration of $0.05-0.13 \mathrm{Mol} / \mathrm{l}$ with the assumption that the increase in density will not have a very significant influence on the resulting self-assembly process.

In order to probe the morphology of the PEDOT:PSS complexes, we will perform the following analysis:

- Cluster analysis for the small PEDOT:PSS systems is performed with a friends-of-friends algorithm finding all the molecules within a distance $r_{c}$ of $4.0 \AA$. We calculate the evolution of the number of PEDOT aggregates and their composition (number of PEDOT units and PTS units).
- In order to probe the PEDOT complex morphology, a specific radial distribution function denoted by $\operatorname{RDF}(\mathrm{C}-\mathrm{C})$ is introduced. For each carbon atom on a given PEDOT backbone, the RDF is calculated so that only the closest backbone carbon atoms of other PEDOT units are taken into account:

$$
\begin{equation*}
R D F(C-C)=\frac{1}{<\rho_{c}>} \frac{1}{\left.<N_{c}\right\rangle} \sum_{i}^{M_{c}} \sum_{j}^{N_{c}} \frac{\delta\left(r_{i j}-r\right)}{4 \pi r^{2}} \tag{5.1}
\end{equation*}
$$

where $<\rho_{c}>$ is the local density of carbon atoms on PEDOT backbone, $N_{c}$ is the number of carbon atoms of PEDOT backbone. For each atom $i$ of carbon backbone, we define $M_{c}$ as the closest backbone carbon atom of other PEDOT and this number is equal to $M_{c}=$ (PEDOT units) - 1. For each backbone carbon atom, we defined a radius r and we denote by $\mathrm{r}_{i j}$ the distance between a backbone carbon $i$ and the closest carbon atom $j$ of another PEDOT unit. The range of $r$ is considered from 0 to half of the box size.

- We estimate the distribution of PTS (or PSS) around PEDOT by calculating the radial distribution function $\operatorname{RDF}(\mathrm{C}-\mathrm{S})$ between the sulfonate group $\mathrm{SO}_{3}$ of PTS (or PSS) and the PEDOT backbone carbon atoms.


### 5.2.2 Morphology of aqueous PEDOT:PSS complexes - small systems case



Figure 5.4 - Final snapshots of the equilibrated small aqueous PEDOT:PSS systems. A) Aqueous solution of tri-EDOT:PTS. B) Aqueous solution of tri-EDOT:16SS. C) Aqueous solution of 6EDOT:16SS. PEDOT chain are represented in blue while PSS/PTS are represented in green.

After random insertion of PEDOT and PSS/PTS chains in the simulation box, equilibration is reached for each system and we observe the local organization of PEDOT chains. Figure 5.4 shows snapshots of the systems at stable (or metastable) equilibrium. The evolution of each system leads to several domains of $\pi-\pi$ stacked PEDOT chains. It reveals that crystal nucleation takes place spontaneously in aqueous solution as it is known for similar polymeric systems [303]. For systems containing short PEDOT chains (tri-EDOT model), there is no connection between the small PEDOT aggregates contrary to the system with extended 6-EDOT chains. Nonetheless, it has been proven through MD simulations that the morphology of PEDOT aggregates is independent of the chain length [301].

In order to quantify the number of PEDOT aggregates through time, we determine the composition of each PEDOT aggregates as defined in section 5.2.1 Cluster analysis performed for the tri-EDOT:PTS system indicates that each cluster is composed of $\times 2$ or $\times 5$ tri-EDOT units and $\times$ 1-4 PTS units (Figure 5.5-A). The clusters appear to be stable (or metastable/dynamically arrested) within our simulation timescale.

At first sight, it appears that PTS small ions disturb the formation of a single tri-EDOT aggregate and this trend is confirmed when PTS is replaced by 16 SS chain leading to several clusters of $\sim \times 9,5,4$ and 2 tri-EDOT units coexisting during the NPT equilibration phase (Figure 5.5-B). However, cluster analysis performed on the 6EDOT:16SS system shows a single large cluster with


Figure 5.5 - Cluster analysis: composition of PEDOT clusters. A) Tri-EDOT:PTS model. B) Tri-EDOT:16SS model. C) 6EDOT:16SS model.
$\times 12$-15 6EDOT chains because longer PEDOT chains interact through $\pi-\pi$ stacking with several other chains simultaneously (Figure 5.5-C). In that case, the resulting morphology is rather a network of PEDOT:PSS than distinct PEDOT clusters aggregated on the PSS chain.

We characterize the local structure adopted by PEDOT chains by calculating the modified RDF(C-C) as shown on Figure 5.6 top. First, the RDF(C-C) curves exhibit sharp peaks at integer values so that $1 \leq d / d_{\pi-\pi} \leq 4$ corresponds to the formation of $\pi-\pi$ stacked crystallites involving at most 4 layers of PEDOT chains. The length of the PEDOT or PSS chains do not have any influence on the $\pi-\pi$ stacking distance $d_{\pi-\pi} \approx 3.65 \AA$ which is close to the value $d_{\pi-\pi} \approx 3.45 \AA$ found by Franco-Gonzalez and Zozoulenko [301] for PEDOT:PTS aqueous systems.

Figure 5.6. bottom plot exhibits spatial distribution of sulfonate group of PTS counterions or PSS chains around the PEDOT aggregates estimated by calculating RDF(C-S). In systems containing the 16 SS chains, the $\mathrm{RDF}(\mathrm{C}-\mathrm{S})$ curves present a peak at the same position ( $\mathrm{r} \approx 9 \AA$ ) but with a higher intensity than the peak obtained for systems containing the PTS anions. As pointed out by Franco-Gonzalez and Zozoulenko [301], this peak is attributed to a location of PTS or PSS unit on the side of the PEDOT crystallites domains. A lower intensity peak located around $\mathrm{r} \approx 4 \AA$ appears in the tri-EDOT:PTS solution corresponds to the preferential binding of free PTS
anions at the top of PEDOT cystallites (face to the tiophene groups of PEDOT).


Figure 5.6 - Top) Radial distribution $\operatorname{RDF}(\mathrm{C}-\mathrm{C})$ between closest PEDOT backbone carbon atoms as defined in section 5.2.1 and calculated for the small aqueous solutions of PEDOT:PSS. Bottom) Radial distribution RDF (C-S) between PEDOT backbone carbon atoms and sulfonate groups $\mathrm{SO}_{3}$ of PTS/PSS calculated for small aqueous solutions of PEDOT:PSS.

Our simulations of small systems of PEDOT:PSS solutions reveal the same behavior than the one reported on PEDOT:PTS systems [301]. In order to validate our observations, we analyze larger systems constituted of similar PEDOT:PSS aqueous solutions.

### 5.2.3 Morphology of aqueous PEDOT:PSS complexes - larger systems case

We probe the effect of system size on the resulting morphology characterics of aqueous PEDOT:PSS solutions. The systems to be discussed in this section are presented in Table 5.2 Snapshots of final tri-EDOT:PTS and tri-EDOT:16SS aqueous systems are presented in Figure 5.7.

Contrary to the tri-EDOT:16SS large system, the tri-EDOT:PTS large system exhibits several extended crystallite aggregates of tri-EDOT. This resulting morphology is in agreement with experiments showing highly-ordered PEDOT:PTS films produced by in situ vapor phase [304, 171]. A PEDOT:PTS film spin-casted in presence of polar solvent has also been reported to exhibits


Figure 5.7 - Final snapshots of the equilibrated large systems of PEDOT:PSS. A) Aqueous solution of tri-EDOT:PTS. B) Aqueous solution of tri-EDOT:16SS.
crystalline morphology [170]. Indeed, small PTS anions would easily distribute around PEDOT chains to neutralize them without interfering in the crystal self-assembly [163].

We calculate again the radial distribution functions $\operatorname{RDF}(\mathrm{C}-\mathrm{C})$ and $\operatorname{RDF}(\mathrm{C}-\mathrm{S})$ to characterize respectively the local morphology of the PEDOT aggregates and the distribution of PTS/PSS around PEDOT. The results are plotted in Figure 5.8

The $\operatorname{RDF}(\mathrm{C}-\mathrm{C})$ plots show the presence of sharp peaks over integer values of $d / d_{\pi-\pi}=8$ corresponding to the existence of extended tri-EDOT domains. On the contrary, $\mathrm{RDF}(\mathrm{C}-\mathrm{C})$ displaying only two and three sharp peaks reveals existence of shorter PEDOT crystallites domains. The RDF(C-S) curves confirm the distribution of PTS/PSS around the PEDOT chains observed previously for small systems of PEDOT:PSS. The only difference involves tri-EDOT:PTS system that shows more significant finite-size effects with the local peak at $r \approx 4 \AA$ that vanishes because most tri-EDOT units form large domains in $\pi-\pi$ stacking and PTS anions have no other choice than interacting with the sides of the tri-EDOT units (left, Figure 5.7). In order to probe the influence of PTS anions or 16SS chains on the formation of PEDOT aggregates, we retrieve the number of PEDOT clusters through the simulations as showns in Figure 5.9

It is clear that several PEDOT clusters are present after equilibration. The systems reach configurations with $\times 4-8$ and $\times 8-16$ tri-EDOT clusters respectively for the tri-EDOT:PTS and the tri-EDOT:16SS systems. The 16SS chains prevent the small PEDOT units to assemble into large domains as shown in right, Figure 5.7.


Figure 5.8 - Top) Radial distribution RDF(C-C) between closest PEDOT backbone carbon atoms as defined in section 5.2.1 and calculated for the large aqueous solutions of PEDOT:PSS. Bottom) Radial distribution RDF(C-S) between PEDOT backbone carbon atoms and sulfonate groups $\mathrm{SO}_{3}$ of PTS/PSS calculated for large aqueous solutions of PEDOT:PSS.

### 5.2.4 Conclusion

Morphology of different PEDOT:PSS solutions has been studied using atomistic molecular dynamics simulations. We confirmed through our simulations the findings of the previous work of Franco-Gonzalez and Zozoulenko [301]. We highlighted that PEDOT chains are aggregated in crystalline domains of $\pi-\pi$ stacked units for which $\pi-\pi$ distance are independent of the PEDOT chain length. PTS counterions are located mostly on the side of PEDOT units. The aqueous simulations of PEDOT:PSS constitute the starting point for further numerical experiments in presence of ILs.

### 5.3 Prediction of ion exchange between PEDOT:PSS/IL

The proposed exchange mechanism between $\mathrm{PEDOT}^{+}: \mathrm{PSS}^{-}$and the imidazolium based IL $\mathrm{EMIM}^{+}: \mathrm{X}^{-}$would result in the screening of electrostatic interactions inducing separation between PEDOT and PSS. This would lead to the emergence of ordered networks of $\mathrm{PEDOT}^{+}: \mathrm{X}^{-}$sur-


Figure 5.9 - Evolution of the number of clusters. Top) Aqueous solution of tri-EDOT:PTS. Middle) Aqueous solution of tri-EDOT:16SS system. Bottom) Aqueous solution of 6EDOT:16SS.
rounded by IL anions.
In this section, we quantitatively evaluate the hypothesis of an ion exchange between PEDOT:PSS and EMIM: X where X refers to the series of anions $\mathrm{X}=\mathrm{Cl}, \mathrm{ES}$, TCM and TCB. A suitable IL anion X in EMIM: X should disrupt properly the PEDOT:PSS attraction and bring the PEDOT units together in order to increase both the mobility $\mu$ through extended PEDOT domains and the charge carrier density $n$ of PEDOT units through X anions. Previous DFT calculations performed in vacuum (gas-phase) on minimal model of PEDOT:PSS and EMIM:X where $\mathrm{X}=\mathrm{Cl}$, ES, TCM, TCB highlighted the design principles that an IL anion should satisfy for proper ion exchange [201]. A bulky, polarizable, hydrophobic but water soluble anion with multiple electronwithdrawing groups would be required. Indeed, based on such principles, the hypothetical anion HCCP has been previously proposed.

In order to confirm the design principles that the anion X should satisfy for both aspects of conductivity enhancement, we determine the free energy of ion exchange $\Delta \Delta G_{x}$ through binding free energy $\Delta G_{b}$ calculated for each separated ion pair using PMF obtained from umbrella sampling
method (US). This approach allows to take into account explicit water solvation effects in contrary to previous gas phase DFT calculations [201, 11]. These simulations take also into account the entropic contributions from the translational, rotational and vibrational degrees of freedom of the ion pair for which free energy is determined along the reaction coordinate. Binding free energies $\Delta G_{b}$ and ion exchange free energies $\Delta \Delta G_{x}$ estimated from this series of PMF calculations, not only confirm the ion exchange free energy calculated through DFT study on the minimal model of PEDOT:PSS but clearly demonstrate the importance of hydrophobicity of the anion X

### 5.3.1 Methods

We perform umbrella sampling simulations to compute the PMF to retrieve the binding free energy $\Delta G_{b}$ of ion pairs. A total of 12 ion pairs are considered as in the previous DFT study corresponding to $\mathrm{EMIM}^{+}: \mathrm{X}^{-}$and tri-EDOT: $\mathrm{X}^{-}(\mathrm{X}=\mathrm{PTS}, \mathrm{Cl}, \mathrm{ES}, \mathrm{TCM}, \mathrm{TCB}$ and HCCP $)$. The bonded and non-bonded OPLS-AA parameters as well as atomic-charge parameters used to compute the electrostatic interactions are given in Appendix $D$


Figure 5.10 - Snapshots of the pulling simulations in the case of tri-EDOT:TCB complexes. Left) Initial configuration. Right) Final configuration.

We retrieve the DFT-optimized geometry of each ion pair and solvate the pair in a tetragonal simulation box ( $40 \AA \times 40 \AA \times 120 \AA$ ); 6442-6452 waters; density $\rho \approx 1.0 \mathrm{~g} / \mathrm{cm}^{3}$, concentration in solute $c=8.6 \mathrm{mMol} / \mathrm{l})$. Then, steepest-descent minimization is performed followed by equilibration during 100 ps constant-pressure (1 bar) and constant-temperature ( 293 K ) NPT MD simulation using the Berendsen weak coupling bath while the heavy atoms of the complex EMIM ${ }^{+}: \mathrm{X}^{-}$or tri-EDOT: $\mathrm{X}^{-}$are harmonically restrained around their equilibrium positions with a force constant of $1,000 \mathrm{~kJ} . \mathrm{mol}^{-1} . \mathrm{nm}^{-2}$.

After equilibration, we remove such constraints and we use the center-of-mass (CM) of EMIM (or tri-EDOT) as a fixed reference for pulling anion X away along the $z$ coordinate at a pulling rate
of $0.1 \AA . \mathrm{ps}^{-1}$ over 500 ps until the final CM-CM distance of EMIM:X (or tri-EDOT:X) reaches 50 A. A typical snapshot of the simulation is given in Figure 5.10

For each ion pair, we performed a series of 200 US simulations over 200 windows properly selected out of the pulling simulation so that successive windows corresponds to a $0.25 \AA$ increment in the distance between CMs of X and the reference cation. The biased simulation for each window consists of 100 ps equilibration and 10 ns production NPT run, which employs the ParrinelloRahman barostat and the Nose-Hoover thermostat and uses leap-frog integration scheme to solve Newton's second law of motion with a timestep of 2 fs . The biased harmonic potential is applied to the CM of X with a force constant of $1,000 \mathrm{~kJ} . \mathrm{mol}^{-1} . \mathrm{nm}^{-2}$.

The total production run to determine binding free energy $\Delta G_{b}$ for an ion pair is thus 2,000 ns and the WHAM method is used to construct the corresponding PMF and to estimate the corresponding standard deviation (see Appendix F.1 from 200 generated PMFs using the Bayesian bootstrap technique described by Hub et al. [305] and implemented in the version 5.1.4 of GROMACS.

### 5.3.2 Ion exchange free energy: PMF calculations

The profiles of the calculated PMFs for tri-EDOT:X and EMIM:X complexes are presented in Figure 5.11 They display local minima when the CM-to-CM separation between the pair is $\approx 4-5$ $\AA$, excepted for EMIM:PTS whose PMF shows a local minimum around $8 \AA$. The snapshots of the complexes taken at the COM-COM distance between anion and cation $\mathrm{R}_{e q}$ corresponding to the minimum of the PMF are displayed in Apendix F.3


Figure 5.11 - PMF curves: tri-EDOT:X (left) and EMIM:X (right).

The difference in height from the plateau (at $54 \AA$; set to zero) to the minimum gives the binding free energy $\Delta G_{b}$ of each ion pair (Table 5.3 in $\mathrm{kJ} / \mathrm{mol}$ ). More negative $\Delta G_{b}$ value implies a more favorable binding. The binding to EMIM is more favorable for TCM than other anions because van der Waals (vdW) short-range interactions are maximum when TCM binds with its planar structure facing EMIM aromatic cycle. The binding energy to the hydrophilic EMIM is in the decreasing order $\operatorname{TCM}(-7.1)>\operatorname{HCCP}(-3.2) \sim \operatorname{ES}(-2.8) \sim \operatorname{PTS}(-2.6) \sim \operatorname{TCB}(-2.6)>\operatorname{Cl}(-1.9)$.

In the case of interaction with the hydrophobic tri-EDOT, the most favorable binding occurs for the charge-dispersed and hydrophobic $\operatorname{TCM}(-18.3)$ because short range vdW interactions between PEDOT and this planar anion are maximum at short distance. The binding free energy of the rest of the series varies in the following order: $\operatorname{HCCP}(-17.3)>\operatorname{TCB}(-11.2)>\operatorname{PTS}(-2.8) \sim \operatorname{ES}(-3.0)>$ $\mathrm{Cl}(-1.8)$. The binding to cations (PEDOT, in particular) in water is less favorable for the hard anions (Cl, ES and PTS) than for the soft, bulky and hydrophobic anions (TCM, TCB and HCCP).

Table 5.3 - Ion binding and ion exchange free energies ( $\mathrm{kJ} / \mathrm{mol}$ ).

| X | $\Delta G_{b}$ (EMIM:X) | $\Delta G_{b}$ (tri-EDOT:X) | $\Delta \Delta G_{x}^{a}$ |
| :---: | :---: | :---: | :---: |
| PTS | $-2.6 \pm 0.6(-317)^{\text {b }}$ | $-2.8 \pm 0.6(-235)^{b}$ | - |
|  | $-\overline{1} .9 \overline{9} \overline{0} . \overline{4} \overline{(-352})^{\text {b }}$ | $-\overline{1.8} \overline{8} \overline{ \pm} \overline{0} . \overline{4}(-\overline{2} 5 \overline{8})^{\text {b }}$ | $\overline{0} . \overline{3} \overline{ \pm} \overline{2} . \overline{0} \overline{(12})^{6}{ }^{-}$ |
| ES | $-2.8 \pm 0.5(-306)^{b}$ | $-3.0 \pm 0.6(-221)^{b}$ | $0.0 \pm 2.3(2)^{b}$ |
| TCM | $-7.1 \pm 0.5(-248)^{b}$ | $-18.3 \pm 0.5(-200)^{b}$ | $-11.0 \pm 2.2(-35)^{b}$ |
| TCB | $-2.6 \pm 0.4(-227)^{b}$ | $-11.2 \pm 0.5(-180)^{b}$ | $-8.4 \pm 2.1(-35)^{b}$ |
| HCCP | $-3.2 \pm 0.5(-199)^{b}$ | $-17.3 \pm 1.3(-154)^{b}$ | $-13.9 \pm 3.0(-38)^{b}$ |

${ }^{a} \Delta G_{b}$ (EMIM:PTS) $+\Delta G_{b}$ (tri-EDOT:X) $-\Delta G_{b}$ (tri-EDOT:PTS) - $\Delta G_{b}$ (EMIM:X)
${ }^{b}$ Values in parentheses are taken from DFT calculations in gas phase (ref. [201]).

The free energy of ion exchange $\Delta \Delta G_{x}$ between tri-EDOT:PTS and EMIM:X is estimated by combining the binding free energy $\Delta G_{b}$ in the following way:

$$
\begin{align*}
\Delta \Delta G_{x} & =\Delta G_{b}(E M I M: P T S)+\Delta G_{b}(\operatorname{tri}-E D O T: X)  \tag{5.2}\\
& -\Delta G_{b}(\operatorname{tri}-E D O T: P T S)-\Delta G_{b}(E M I M: X)
\end{align*}
$$

A more negative value of $\Delta \Delta G_{x}$ indicates a more favorable ion exchange between tri-EDOT:PTS and EMIM:X. Hence, when PEDOT:PSS and EMIM:TCM or EMIM:TCB are mixed and sonicated in water, we conclude from our calculations that new hydrophobic pairs of tri-EDOT:TCM $\left(\Delta G_{b}\right.$ $\sim-18.3 \mathrm{~kJ} / \mathrm{mol})$ or tri-EDOT:TCB $\left(\Delta G_{b} \sim-11.2 \mathrm{~kJ} / \mathrm{mol}\right)$ form spontaneously leaving hydrophilic

EMIM:PTS pair behind ( $\Delta G_{b} \sim-2.6 \mathrm{~kJ} / \mathrm{mol}$ ). This behavior is in contradiction to the trend observed for gas phase DFT calculations where exchange was driven by the formation of EMIM:PTS pair $\left(\Delta G_{b} \sim-317 \mathrm{~kJ} / \mathrm{mol}\right)$ rather than tri-EDOT:TCM $\left(\Delta G_{b} \sim-200 \mathrm{~kJ} / \mathrm{mol}\right)$ or tri-DOT:TCB ( $\Delta G_{b} \sim-180 \mathrm{~kJ} / \mathrm{mol}$ ) pairs.

Indeed, negative free exchange energy is predicted for EMIM:TCM ( $-11.0 \mathrm{~kJ} / \mathrm{mol}$ ), EMIM:TCB $(-8.4 \mathrm{~kJ} / \mathrm{mol})$ and in presence of the newly designed anion IL EMIM:HCCP ( $-13.9 \mathrm{~kJ} / \mathrm{mol}$ ). On the contrary, PSS and ES display the same sulfonate groups $\mathrm{SO}_{3}$ and display similar binding energies with PEDOT and PSS. Therefore, exchange between PEDOT:PSS and EMIM:ES is limited as shown by the null binding exchange free energy ( $\sim 0.0 \mathrm{~kJ} / \mathrm{mol}$ ). Also localized and hydrophilic point-charge-like Cl anion displays a similar binding to PEDOT and EMIM because of its high solvation in water, resulting in a slightly positive free exchange energy $(\sim 0.3 \mathrm{~kJ} / \mathrm{mol})$.

Another way to compare each ion pair interaction involves the calculation of the association constant $K_{A}$ defined by:

$$
\begin{equation*}
\left[C^{+}\right]+\left[A^{-}\right]=[C A] \tag{5.3}
\end{equation*}
$$

where $\left[\mathrm{A}^{-}\right],\left[\mathrm{C}^{+}\right]$and $[\mathrm{CA}]$ are respectively the concentration of the anion, cation and of the complex. Consequently, the association constant writes:

$$
\begin{equation*}
K_{A}=\frac{[C A]}{\left[C^{+}\right]\left[A^{-}\right]} \tag{5.4}
\end{equation*}
$$

We calculated the association constant by following the protocol established by Dill and coworkers [306] and later by Yee, Shah and Maginn [307] using PMF curves. It is assumed that the complex is dissociated as soon as the CM-CM distance between the anion and the cation is greater than the distance corresponding to the PMF global minimum $R_{e q}$.

If $\alpha$ is the degree of dissociation and $c$ the concentration of solute ( $c=8.6 \mathrm{mMol} / \mathrm{l}$ ), the concentration of each species in solution writes:

$$
\begin{equation*}
[C A]=(1-\alpha) c ; \quad\left[A^{-}\right]=\left[C^{+}\right]=\alpha c \tag{5.5}
\end{equation*}
$$

The association constant can be written in term of the degree of dissociation:

$$
\begin{equation*}
K_{A}=\frac{(1-\alpha)}{\alpha^{2} c} \tag{5.6}
\end{equation*}
$$

By following the work of Yee and co-workers [307], the degree of association is estimated by calculating the ratio between the integrated radial distribution function $g(r)$ up to the equilibrium distance $R_{e q}$ and the total number of available free ions:

$$
\begin{equation*}
1-\alpha=\frac{\int_{0}^{R_{e q}} 4 \pi r^{2} \rho g(r) d r}{\int_{0}^{\infty} 4 \pi r^{2} \rho g(r) d r} \tag{5.7}
\end{equation*}
$$

where $\rho$ is the density solute in the box. In particular, there is one ion in the a simulation box of volume $\mathrm{V}=4 \mathrm{~nm} \times 4 \mathrm{~nm} \times 12 \mathrm{~nm}=192 \mathrm{~nm}^{3}$ corresponding to a density of $\rho=0.005 \mathrm{ion} / \mathrm{nm}^{3}$.

The RDF is also estimated from the PMF calculation with $r$ refering to the distance between the two molecules so that:

$$
\begin{equation*}
\Delta G_{b}(r)=-k_{B} T \ln g(r) \tag{5.8}
\end{equation*}
$$

The PMF are calculated for a single ion pair so that the degree of association writes in term of the coordination number of the anion:

$$
\begin{equation*}
1-\alpha=\int_{0}^{r_{e q}} 4 \pi r^{2} \rho g(r) d r=C N\left(R_{e q}\right) \tag{5.9}
\end{equation*}
$$

We displayed in Figure 5.12 the RDF derived from the binding free energy $\Delta G_{b}$ by using equation 5.8 A higher RDF peak corresponds to a higher interaction between the cation and the
anion.


Figure 5.12 - Logarithm of the radial distribution function between the CM of the reference (triEDOT for Figure A or EMIM for Figure B) and the CM of the anion.

We calculated the coordination number by integrating the radial distribution function in Figure 5.12 Hence, the association constant writes in term of coordination number as following:

$$
\begin{equation*}
K_{A}=\frac{C N\left(R_{e q}\right)}{\left(1-C N\left(R_{e q}\right)\right)^{2} c} \tag{5.10}
\end{equation*}
$$

The table 5.4 displays the association constant of each ion pair, initial CM-CM distances $R_{i}$ between cation and anion and equilibrium CM-CM distances $R_{e q}$ between cation and anion. We notice that the initial and equilibrium CM-CM distances between tri-EDOT and PTS ( $R_{i}=6.5 \AA$ and $R_{e q}=6.3 \AA$ ) are higher than the corresponding CM-CM distances between tri-EDOT and ES $\left(R_{i}=4.8 \AA R_{e q}=4.3 \AA\right)$. This trend is also observed for EMIM:PTS $\left(R_{i}=6.1 \AA R_{e q}=8.0 \AA\right)$ and EMIM:ES complexes $\left(R_{i}=5.4 \AA R_{e q}=4.0 \AA\right)$. This fact would be explained because of PTS has a CM more shifted due to the presence of a carbon cycle as shown by the initial EMIM:ES and EMIM:PTS complexes (see Appendix F.3).

We calculated the association constant for each ion pair (Table 5.4). The tri-EDOT unit binds lightly to the hard anions Cl, ES and PTS because their association constant are lower than 1. On the other hand, the softer anions TCM, TCB and HCCP display higher value of $K_{A}$, respectively of $95.30,8.34$ and 382.23 , indicating that ion pair would emerge between these ions and tri-EDOT. Association constant values are in agreement with those calculated with the same approach through PMF calculations [307] for EMIM: $\mathrm{Cl}\left(K_{A}=0.48\right.$ at 300 K$)$ but differs slightly for EMIM:ES $\left(K_{A}\right.$
$=1.33$ at 300 K ). We provide in Appendix F.4 a step-by-step comparison between our results and the results found by Yee and co-workers [307].

Also, experimental determination of association constants can lead to variation of several orders of magnitude and therefore we do not expect a perfect agreement with values calculated from numerical simulations. For instance, the experimental binding association constant of EMIM:ES at 298 K is $K_{A}=0.008$, three order of magnitude lower than the value computed by MD simulations.

The association constants of soft and charge-dispersed anions TCM, TCB and HCCP vary by several orders of magnitude whether they bind to tri-EDOT or EMIM. The association constant of TCB rises by one order of magnitude from 0.31 with EMIM to 8.34 with tri-EDOT. HCCP and TCM display the highest variations of binding coefficient, respectively from 0.62 and 0.92 with EMIM to 382.23 and 95.30 with tri-EDOT. These changes indicate that beyond interaction with the solvent, the interaction with the cation plays also an important role given that tri-EDOT is more hydrophobic than EMIM and consequently TCM, TCB and HCCP would bind more strongly to tri-EDOT than EMIM.

Table 5.4 - Initial CM-CM distances, equilibrium CM-CM distances and association constant for each complexes at $298 K$ for EMIM:X and tri-EDOT:X complexes.

| Complex | $R_{i}^{a} \AA$ | $R_{e q}^{b} \AA$ | $R_{C}^{c} \AA$ | $K_{A}^{d}$ from $R_{e q}$ | $K_{A}^{d}$ from $R_{C}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| [tri-EDOT][PTS] | 6.5 | 6.3 | 7.5 | 0.31 | 1.67 |
| [ $\mathrm{tri} \overline{\mathrm{E}} \overline{\mathrm{E}} \overline{\mathrm{D}} \overline{\mathrm{O}} \overline{\mathrm{T}}][\overline{\mathrm{Cl}}]$ | $\overline{4} \cdot \overline{6}$ | $\overline{4} . \overline{3}$ | $\overline{6} . \overline{0}$ | $\overline{0} . \overline{2} 8$ | $\overline{1} . \overline{0} 1$ |
| [tri-EDOT][ES] | 4.8 | 4.3 | 9.0 | 0.42 | 3.8 |
| [tri-EDOT][TCM] | 3.4 | 3.5 | 6.1 | 95.30 | 9780.04 |
| [tri-EDOT][TCB] | 4.4 | 4.2 | 7.2 | 8.34 | 20.28 |
| [tri-EDOT][HCCP] | 4.6 | 4.3 | 7.0 | 382.23 | 3706894.66 |
| [EMIM][PTS] | 6.1 | 8.0 | 7.9 | 3.44 | 3.31 |
| $[\mathrm{EMIM} \overline{\mathrm{M}}][\mathrm{Cl}]$ | $\overline{4} . \overline{6}$ | $\overline{4} . \overline{0}$ | $\overline{6} . \overline{6}$ | $\overline{0} . \overline{1} 8$ | $\overline{1} .31$ |
| [EMIM][ES] | 5.4 | 4.0 | 6.6 | 0.50 | 1.69 |
| [EMIM][TCM] | 3.7 | 3.7 | 5.6 | 0.92 | 3.29 |
| [EMIM][TCB] | 4.1 | 4.3 | 6.6 | 0.31 | 1.56 |
| [EMIM] $[\mathrm{HCCP}]$ | 5.1 | 5.8 | 8.1 | 0.62 | 2.89 |

[^2]At first sight, results obtained from PMFs and association constant calculations seem in contradiction with the trend exhibited from previous gas-phase DFT calculations where the anions bind more strongly to EMIM than to PEDOT (Table 5.3). This difference emphasizes the important
role played by water solvation on ion pair formation.

### 5.3.3 Solvation of IL anions in US simulations.

In order to achieve a better understanding of the effect of anion hydration on the complex formation, we analyze equilibrium configurations obtained from the previous biased simulations and corresponding to CM-CM distances $\zeta$ between the ions pairs. From these configurations, we extract the radial distribution functions RDFs and coordination numbers CNs between various atomic sites on the anion and oxygen atoms of water molecules.


Figure 5.13 - Left) Radial distribution function RDF[A-OW] calculated between atoms A of anion X and oxygen atoms (OW) of water. Atoms A can be: chloride ions Cl , oxygen atoms O 1 of sulfonate group of ES, oxygen atoms O2 of sulfonate group of PTS, nitrogen atoms N1, N2 and N3 respectively for TCM, TCB and HCCP. Right) Coordination number of water oxygen atoms OW resulting from the integrated radial distribution functions.

Figure 5.13 displays the RDF and CN for different anions calculated over 7 US windows selected for $\zeta$ comprised between 40 and $50 \AA$ which is much greater than the cutoff distance for non-bonded interactions ( $14 \AA$ ) to analyze water distribution around each anion without interference of the cation. It reveals that water molecules interact strongly with $\mathrm{Cl}^{-}$as can be seen from the first peak of height $\sim 3.6$ at $3.2 \AA$. The strong hydration of Cl anion ( $\mathrm{CN}=7.05$ of water oxygens after integration of the RDF up to $3.8 \AA$ corresponding to the first solvation shell (blue curve)) is a consequence of its spherical shape and small size inducing a high charge density. It is important to notice that RDFs in Figure 5.13 do not provide direct informations about hydrogen bonds between water and anions, because the presence of hydrogen bonds is detected through distance/angle calculation between the donor and hydrogen acceptor to be discussed below [308].

The RDF[O-OW] is similar between ES and PTS and the first peak at $2.8 \AA$ is associated to
a coordination number respectively $\mathrm{CN}(2.42)$ and $\mathrm{CN}(2.71)$ after integration up to $3.2 \AA$ (green and black curves). However, the peaks in RDF[O1-OW] and RDF[O2-OW] are more spaced and damped compared to those in $\mathrm{RDF}[\mathrm{Cl}-\mathrm{OW}]$. This observation may result from the bulky nature of the anions and the fact that the negative charge is dispersed over several number of sites ( $\left|\mathrm{q}_{O 1, O 2}\right|$ $=0.76<\left|\mathrm{q}_{C l}\right|=1.00$ ) leading to a lower charge density and a lower number of water molecules in the first solvation shell. Hence, the second solvation shell would be more disordered in that case.

In the case of bulky and charge-dispersed anions like TCM, TCB and HCCP the radial distribution functions between nitrogen (respectively N1, N2 and N3) atoms and water oxygen have their strongest peaks as low as $\sim 1$ at $3.0 \AA$. The depletion of water around these anions (CN of 1.94 for TCM, 2.01 for TCB and 1.20 for HCCP in the first solvation shell)) is explained by their delocalized charge: each Nitrogen carried a charge $\left|\mathrm{q}_{N 1, N 2, N 3}\right| \sim 0.50$.

The size of the first solvation shell differs between anions due to the difference in van der Waals radius of the considered atoms. The first solvation shell radius is the largest for $\mathrm{Cl}\left(r_{v d W}=4.41\right.$ $\AA$ ), followed by those for nitrogen atoms in carbonitrile groups of TCM, TCB and HCCP ( $r_{v d W}$ $=3.25 \AA$ ) and lastly for oxygen atoms of ES and PTS ( $r_{v d W}=2.96 \AA$ ).
$\operatorname{RDF}[\mathrm{A}-\mathrm{OW}]$ is used therein to evaluate the number of hydrogen bonds (H-bonds) that are forming between the most electronegative atoms of the anions (Oxygens of sulfonate groupes and Nitrogens of cyano groups) and water molecules. Our quantitative evaluation of hydrophobicity is set up as follows: a more hydrophobic anion will have fewer hydrogen bonds forming with the most electronegative atoms. In the case of Cl , we also compute hydrogen bonds only for comparison with the other anions, but one has to remember that Cl is large and its lone electron is in a diffuse orbital so that its charge density is low to act as an efficient bond acceptor. The procedure to calculate the H -bonds is described as follows. We consider first that an H -bond forms only for the water molecules that are located in the first solvation shell of Cl, sulfonate oxygens of PTS or ES and nitrogens of TCM, TCB and HCCP, so that the cutoff $\mathrm{r}_{c}$ is indeed set up at the first minimum after the solvation shell by looking at the $\mathrm{RDF}(\mathrm{X}-\mathrm{Ow})$ curve (Figure 5.13). Then, an extra criterion to consider H -bond formation is determined by the angle $\alpha<30^{\circ}$ made by Hydrogen-OxygenAcceptor ( Cl ion, O or N atoms). The results for the cutoff used for the H -bond calculation and the average number of H -bonds established for Cl , sulfonate oxygen atom O 1 of $\mathrm{ES}, \mathrm{O} 2$ of PTS as well as nitrogen atom $\mathrm{N} 1, \mathrm{~N} 2$ and N 3 respectively for $\mathrm{TCM}, \mathrm{TCB}$ and HCCP are given in Table
5.5. We consider the above 7 US windows trajectories in order to have proper statistics and the number of hydrogen bonds through time is plotted in Figure 5.14


Figure 5.14 - Evolution of the average number of hydrogen bonds between water molecules and chloride ion Cl , oxygen atom O 1 of sulfonate group of ES , oxygen atom O 2 of sulfonate group of PTS, nitrogen atom N1, N2 and N3 respectively for TCM, TCB and HCCP.

Notice that in case of several Acceptors (like the 3 oxygens O2 of PTS sulfonate group for instance), the resulting hydrogen bonds in Table 5.5 is the average number of hydrogen bonds forming around one Oxygen atom, by dividing the total number of hydrogen bonds by 3 (i.e the number of oxygen atoms O2).

Table 5.5 - Cutoff for H-bond calculations $\mathrm{r}_{c}$ and average number of H-bonds around the most electronegative atoms.

| Acceptor | $r_{c}(\AA)$ | H-Bonds |
| :--- | :---: | :---: |
| Cl | 3.8 | $5.96 \pm 0.46$ |
| O1(ES) | 3.2 | $2.06 \pm 0.14$ |
| O2(PTS) | 3.2 | $2.34 \pm 0.17$ |
| N1(TCM) | 3.4 | $1.56 \pm 0.18$ |
| N2(TCB) | 3.4 | $1.60 \pm 0.14$ |
| N3(HCCP) | 4.0 | $0.99 \pm 0.10$ |

We classify the anions in the increasing order of hydrophobicity based on the average number of hydrogen bonds in the following order $\mathrm{Cl}(5.96)<\operatorname{PTS}(2.34) \sim \mathrm{ES}(2.06)<\mathrm{TCB}(1.60) \sim$ $\operatorname{TCM}(1.56)<\operatorname{HCCP}(0.99)$. We note the great importance of anion hydrophobicity with anions that tend to display poor hydration like TCM, TCB and HCCP being suitable candidates for proper PEDOT:PSS treatment.

Finally, we determine again the radial distribution of water oxygens OW around the most
electronegative atoms of the anions but for different US windows corresponding to different CMCM between the cation and the anion (Figure 5.15).

We notice that the water solvation does not change when the hydrophilic anions Cl and ES bind to EMIM or tri-EDOT cations. On the contrary, the solvation state of TCM and TCB and HCCP change when they bind to the hydrophobic tri-EDOT as shown by the red arrows (Figure 5.15). The fall of RDF[A-OW] is associated to a depletion of water upon binding of TCM, TCB or HCCP with tri-EDOT. These depletion of water may be associated to an increase of translational entropy of the system that favor binding of the complex. Such fact would explain the higher binding energy $\Delta G_{b}$ for the corresponding complex.

### 5.3.4 Conclusion

We have determined quantitatively the ion exchange between minimal models of PEDOT:PSS and ILs through calculation of PMFs using umbrella sampling method. We found the same trend as for previous DFT calculations. A spontaneous exchange is predicted to happen between PEDOT:PSS and EMIM:X for bulky and charge-dispersed hydrophobic anions $\mathrm{X}=\mathrm{TCM}, \mathrm{TCB}$ and for the hypothetical anion HCCP. On the contrary, limited exchange is predicted in presence of the charge-localized and more hydrophilic Cl and ES anions. However, PMF calculations predict strongest binding between pairs of PEDOT:X than between pairs of EMIM:X which is contrary to the binding energies derived from gas phase DFT calculations that did not take into account solvation effects. Along this line, we probe solvation of key atoms of anions in water, and confirm that the most charge-dispersed and hydrophobic anions TCM, TCB and HCCP were poorly hydrated compared to more hydrophilic Cl, ES and PTS anions. As a result, we emphasize that the hydrophobic and charge dispersed TCM, TCB and HCCP would bind to hydrophobic and charge-dispersed PEDOT while the more hydrophilic EMIM and PSS would form weakly bound pairs or remain in solution.

### 5.4 Ionic exchange in mixed solutions: varying the IL anion

In order to validate the ion exchange predicted by PMF and DFT calculations and probe at a larger length-scale its consequence on the resulted morphology of PEDOT, we perform a series of


Figure 5.15 - Radial distribution function RDF[A-OW] where A can be oxygen atoms O1 and O2 of the sulfonate group of ES and PTS, respectively and the nitrogen atoms N1, N2 and N3 of the cyano groups of TCM, TCB and HCCP, for different CM-CM distances.

MD simulations for various mixed solutions. In particular, we study the influence of IL anion on the evolution of system morphology.

### 5.4.1 Methods

The initial state of the simulation is built from the final state of our previous equilibrated NPT simulation of aqueous solutions of PEDOT:PSS (section 5.2). Water molecules are randomly chosen and replaced by the desired amount of IL EMIM:X. The force field parameters are the same as those used to study aqueous solutions of PEDOT:PSS in section 5.2 and described in Appendix D From the resulting configurations, steepest descent minimization is applied in order to remove strong overlaps between inserted ILs and water or PEDOT:PSS. The procedure to equilibrate the resulted small (and large) systems is performed as follows. We first perform a subsequent 30ns ( $60-\mathrm{ns}$ for large systems) NVT equilibration run. In order to mimic the vigorous mixing and sonication of the PEDOT:PSS + IL solution carried out in experiments, we perform a subsequent $30-\mathrm{ns}$ NPT run (60-ns for large systems) where annealing is applied. We consider a linear increase of temperature from $293 K$ to $363 K$ between 0 and 10 ns followed by a linear decrease until 293 $K$. We finally perform a $30-\mathrm{ns}$ ( $60-\mathrm{ns}$ for large systems) NPT run at fixed temperature ( 293 K ) to ensure well-equilibrated systems.

The different systems to be explored by addition of EMIM: X for the different anions $\mathrm{X}=\mathrm{Cl}$, ES, TCM, TCB or HCCP are summarized in Table 5.6 In the case of small systems, various concentrations of ILs will be considered.

Table 5.6 - Different models for PEDOT:PSS together with the number of PEDOT and PSS moities as well as IL cations and anions in simulation box of $60 \AA$ sidelength.

| Model of PEDOT:PSS | PEDOT:PSS $(\mathrm{c}$ in mol/l) | EMIM:X (c in mol/l) |
| :--- | :---: | :---: |
| tri-EDOT:PTS | $12: 12(0.10)$ | $12: 12(0.10) / 24: 24(0.20)$ |
| tri-EDOT:16SS | $16: 1(0.13)$ | $24: 24(0.20) / 64: 64(0.53)$ |
| 6EDOT:16SS | $16: 2(0.13)$ | $64: 64(0.53) / 128: 128(1.06)$ |

In order to study the influence of the system size on ion exchange between PEDOT:PSS and ILs, large aqueous systems of PEDOT:PSS are considered and summarized in Table 5.7.

We probe the morphology of the resulting solutions of PEDOT:PSS in presence of IL EMIM:X by performing similar analysis than those performed previously on the aqueous systems of PEDOT:PSS in section 5.2 Cluster analysis is performed with a friends-of-friends algorithm finding

Table 5.7 - Different models of PEDOT:PSS together with the number of PEDOT and PSS moities as well as ILs cations and anions in simulation box of $120 \AA$ sidelength.

| Model of PEDOT:PSS | PEDOT:PSS $(\mathrm{c}$ in mol/L) | EMIM:X (c in mol/L) |
| :--- | :---: | :---: |
| tri-EDOT:PTS | $96: 96(0.10)$ | $96: 96(0.10) / 192: 192(0.20)$ |
| tri-EDOT:16SS | $96: 6(0.10)$ | $96: 96(0.10) / 192: 192(0.20)$ |
| 6EDOT:16SS | $48: 6(0.05)$ | $96: 96(0.10) / 192: 192(0.20)$ |

all the molecules within a distance $r_{c}$ of $4.0 \AA . \operatorname{RDF}(\mathrm{C}-\mathrm{C})$ is calculated to probe the structure of the PEDOT aggregates. $\mathrm{RDF}(\mathrm{C}-\mathrm{S})$ is calculated to probe the distribution of PTS/PSS around PEDOT units. Comparisons are made with the untreated aqueous PEDOT:PSS solutions to probe the change of morphology. All the analysis are performed over the last NPT equilibration stage at constant temperature.

We present the results for ion exchange and morphology change in relatively small system of PEDOT:PSS (box sidelength $=60 \AA$ ) in the following section, results obtained with larger systems (box sidelength $=120 \AA$ ) being found in Appendix E

### 5.4.2 Morphology of mixed solution - small system cases

The change in morphology induced by addition of ILs is studied for the small size PEDOT:PSS systems. The series of simulations performed in this section constitute more a proof-of-concept purpose to test the hypothesis of ion exchange mechanism postulated through previous PMFs and DFT calculations.

### 5.4.2.1 Morphology of tri-EDOT:PTS solutions

We explore the morphology of tri-EDOT:PTS systems after addition of EMIM:X where X can be Cl, ES, TCM, TCB or HCCP. First, it is visible from the final snapshots (Figure 5.16) that insertion of IL has a clear effect on PEDOT:PSS organization. As predicted by PMF calculations of section 5.3 mixed solutions of tri-EDOT:PTS in presence of EMIM:Cl or EMIM:ES reveal that hydrophilic and charge-localized anions Cl and ES tightly bind to EMIM or PEDOT and are mostly hydrated. It is unlikely that Cl and ES would form strong pairs with EMIM or tri-EDOT due to thermal agitation, the magnitude of binding energy between these ions being of the same magnitude that $K T=2.4 \mathrm{~kJ} / \mathrm{mol}$. On the opposite, most of the TCM, TCB and HCCP condense on tri-EDOT units. More planar TCM and HCCP anions appear to have a better ability than
bulky TCB to bind on tri-EDOT units as it was predicted by the binding energies lower between tri-EDOT and these anions. IL anion tends to act as a compaction agent to aggregate the tri-EDOT units. Therefore, we perform cluster analysis to determine the number of tri-EDOT cluster and their molecular composition (Figure 5.17). After insertion of ILs, the three clusters of tri-EDOT (2 clusters of $\times 5$ tri-EDOT units and 1 cluster of $\times 2$ tri-EDOT units) gather to form either 2 clusters of $\sim 6$ tri-EDOT units in presence of Cl or ES anions or one cluster of $\times 12$ tri-EDOT units in presence of TCM, TCB or HCCP. These hydrophobic anions bind to the hydrophobic tri-EDOTs units and play the role of a anionic glue gathering tri-EDOT units into a single aggregate. Our simulations show that anions with carbonitrile electron-withdrawing groups bind to the tri-EDOT units, which is necessary to sustain a degree of positive doping and thus a high degree of charge carrier density $(n)$. On the other hand, the emergence of larger domains of $\pi-\pi$ stacked PEDOT chains enhance charge carrier mobility $(\mu)$.


Figure 5.16 - Final snapshots of aqueous tri-EDOT:PTS systems treated with $\times 24$ pairs EMIM:X. Untreated aqueous PEDOT:PSS is also shown for comparison purpose. Color code: tri-EDOT (blue), PTS (green), X (red) and EMIM (yellow).


Figure 5.17 - Probability to find at least one cluster made of $n$ tri-EDOT units during the NPT equilibration stage at constant temperature.

Figure 5.18 displays cluster composition in systems mixed with EMIM:ES or EMIM:TCB ILs. Cluster analysis reveals that on average $\sim 3$ ES anions bind to the 2 clusters of tri-EDOT that contain respectively $\times 5-7$ tri-EDOT units which means that most of the ES anion either remain in solution or associate with EMIM (Figure 5.16). The emergence of one large cluster of $\times 12$ triEDOT units is associated to the binding of $\sim \times 14-15 \mathrm{TCB}$ anions to the aggregate. On another hand, each tri-EDOT cluster contains a similar number of PTS anions ( $\sim \times 3-4$ ) but a higher number of EMIM bind to the tri-EDOT clusters in presence of TCB anions. Indeed, bound TCB anions form a layer of negative charges that induce the subsequent binding of EMIM cations.

The minimal model of tri-EDOT and anion X used in previous DFT geometry optimization should describe properly the local electrostatic interactions between different moieties involved in the PEDOT:PSS separation phase triggered by ILs. DFT geometry optimization performed on PEDOT:PSS minimal model provides a clear picture of conformation/configuration of cation-anion complexes. However, emergence of stacked PEDOT:X films can lead to improper binding geometry (and thus improper binding energy) between PEDOT and the IL anion X. Therefore, we probe the most likely ditribution of TCB anions around the resulted tri-EDOT cluster for the tri-EDOT:PTS system mixed with 12 pairs of EMIM:TCB. A representative snapshots of tri-EDOT aggregate with


Figure 5.18 - Composition of tri-EDOT clusters for aqueous tri-EDOT:PTS systems mixed with $\times 24$ pairs of ILs. Top) Cluster composition in presence of EMIM:ES. Bottom) Cluster composition in presence of EMIM:TCB. For comparison, cluster analysis of aqueous tri-EDOT:PTS systems in absence of ILs is displayed in red (tri-EDOT) and cyan (PTS).
its surface bound TCB anions (Figure 5.19 right) is compared to the configuration/conformation expected from DFT-optimized geometry (Figure 5.19 left). The TCB anions bind on the side of the tri-EDOT units which is consistent with the DFT-optimized geometry for the complex triEDOT:TCB. However, geometries obtained by DFT calculations for more hydrophilic species like PTS or ES upon binding to PEDOT (e.g. negatively charged sulfonate group $\mathrm{SO}_{3}$ oriented towards positively tri-EDOT sulfur) are not displayed by direct MD simulations because of solvation effects.

In order to estimate phase segregation between tri-EDOT and PTS anions, we calculated RDF(C-S) between PEDOT backbone carbon atoms and sulfate groups of PTS in Figure 5.20 . A) as well as the coordination number of sulfonate groups $\mathrm{CN}(\mathrm{C}-\mathrm{S})$ as presented in Figure 5.20.B). First, it is expected that a higher amount of ILs favors nanophase segregation between PEDOT and PSS as confirmed by damped RDFs and shifted CNs curves at short distance in presence of 24 pairs of IL (in red) compared to systems containing only 12 IL pairs (in blue).

For systems containing EMIM:Cl or EMIM:ES, $\mathrm{RDF}(\mathrm{C}-\mathrm{S})$ show a limited decrease and CN(CS) present a limited shift toward greater values compared to the curves corresponding to systems containing EMIM:TCM, EMIM:TCB or EMIM:HCCP. These observations confirm the prediction that EMIM:Cl and EMIM:ES has a limited effect on the PEDOT:PSS complex and thus do not


Figure 5.19 - ]
[Structure of the PEDOT cluster obtained by MD vs binding geometries of PEDOT:X]Left) Cluster made of 12 tri-EDOT units (blue) decorated by TCB anions (red) corresponding to the last snapshot of simulated tri-EDOT:PTS mixed with 12 couples of EMIM:TCB. For clarity, only the relevant species are shown. Tri-EDOT:TCB binding geometries (green boxes) are closed to the optimized geometries obtain from DFT calculation (shown on the right). Binding geometries for tri-EDOT:ES and tri-EDOT:PTS pairs are also displayed.
trigger a proper tri-EDOT:PTS phase segregation. We observe a deeper fall of the RDF(C-S) and shift of the CN(C-S) towards greater values when $\mathrm{X}=\mathrm{TCM}$, TCB or HCCP revealing phase separation between tri-EDOT units and PTS anions. The majority of these anions encapsulate the tri-EDOT units gathered into a single cluster (see snapshots, Figure 5.16) and prevent free PTS anions to properly bind to tri-EDOT units. The $\operatorname{RDF}(\mathrm{C}-\mathrm{S})$ and $\mathrm{CN}(\mathrm{C}-\mathrm{S})$ curves are also displayed for untreated system in black for comparison.

MD simulations carried out on minimal model of PEDOT:PSS with different IL seem to validate the ion exchange mechanism predicted by DFT and PMF calculations. In order to support our results, we need to provide evidence of ion exchange for more realistic model of PEDOT:PSS.


Figure 5.20 - Radial distribution $\operatorname{RDF}(\mathrm{C}-\mathrm{S})$ between PEDOT backbone carbon atoms and sulfonate group of PTS and associated coordination number CN(C-S).

### 5.4.2.2 Morphology of tri-EDOT:16SS solution

Hence, we perform simulations to probe the morphology of equilibrated tri-EDOT:16SS systems after injection of different ILs. We display final snapshots of the equilibrated systems in presence of 24 pairs of IL in Figure 5.21 The initial $\times 3-4$ clusters of tri-EDOT units aggregate into a single and large cluster surrounded by IL anions when $\mathrm{X}=\mathrm{TCM}, \mathrm{TBC}$ or HCCP while several clusters decorating the 16 SS chains coexist in presence of Cl or ES anions.

In order to analyze quantitatively the change in morphology for tri-EDOT:16SS systems as highlighted by the snapshots, we perform cluster analysis to determine the probability to find a cluster made of $n$ units of tri-EDOTs (Figure 5.22). There are only slight variations in the number of clusters for systems mixed and equilibrated in presence of hydrophilic anions Cl and ES (blue histograms) compared to untreated aqueous systems of tri-EDOT:16SS (black histogram) containing clusters made of $\times 2,4-5$ or 9 tri-EDOT units.

The hydrophobic anion brings small tri-EDOTs units together into a single large cluster of $\times 15-16$ tri-EDOT units separated from the 16 SS chain (yellow histograms, Figure 5.22 bottom). Coexistence seen between a single large cluster and several small clusters occurs because IL can


Figure 5.21 - Final snapshots of aqueous tri-EDOT:16SS systems treated with $\times 24$ pairs EMIM:X. We show untreated aqueous solution of PEDOT:PSS for comparison. Color code: tri-EDOT (blue), 16SS (green), X (red) and EMIM (yellow).
disturb ordering inside the large cluster, leading to transient unpacking of some of tri-EDOT units. The ionic exchange followed by a nanophase segregation still happens when the PTS anion is replaced by a 16 SS chain.

Additional analysis are performed to determine the composition of tri-EDOT clusters in presence of 24 EMIM:ES and EMIM:TCB pairs (Figure 5.23). Cluster composition for the triEDOT:16SS system is similar to composition found for tri-EDOT:PTS system. More TCB anions ( $\times 12-17$, in yellow) than ES anions ( $\times 0-2$ and $\times 4-6$ depending of the tri-EDOT cluster size) bind to tri-EDOT clusters. As pointed out for the previous tri-EDOT:PTS systems equilibrated with ILs, the higher number of EMIM cations counted as part of the cluster in presence of TCB ( $\times 1-7$, in black) results from TCB forming a layer of negative charges at the surface of the tri-EDOT aggregate.

RDF(C-S) between PEDOT backbone carbon atoms and sulfonate groups of PSS presented in


Figure 5.22 - Probability to find at least one cluster made of $n$ tri-EDOT units during the NPT equilibration stage at constant temperature.


Figure 5.23 - Tri-EDOT cluster composition for aqueous tri-EDOT:16SS systems mixed with 24 pairs of ILs. Top) Composition in presence of EMIM:ES. Bottom) Composition in presence of EMIM:TCB. For comparison, cluster analysis of aqueous system of tri-EDOT:16SS in absence of IL is displayed in red (tri-EDOT).

Figure 5.24 A ) exhibit a deeper fall for systems containing TCM, TCB and HCCP compared to systems with Cl and ES anions. This observations confirm the trend observed for mixed solutions of tri-EDOT:PTS where charge-dispersed and hydrophobic anions are more efficient at inducing


Figure 5.24 - A) Radial distribution $\operatorname{RDF}(\mathrm{C}-\mathrm{S})$ between PEDOT backbone carbon atoms and sulfonate group of PTS and associated coordination number CN(C-S). B) Radial distribution RDF(CC) between closest PEDOT backbone carbon atoms as defined in section 5.2.1 and calculated for the small aqueous solutions of tri-EDOT:16SS with ILs.
phase separation between PEDOT and PSS. We also calculated the $\operatorname{RDF}(\mathrm{C}-\mathrm{C})$ between closest PEDOT backbone carbon atoms for each system in Figure 5.24-B). First, the presence of ILs does not modify the $\pi-\pi$ distance of stacked tri-EDOT units. The higher peaks observed for systems with TCM, TCB and HCCP anions indicate that the $\pi$-stacking order increases compared to systems with Cl and ES supporting the emergence of an extended domain of ordered tri-EDOT units.

However, a higher amount of EMIM:X destabilizes the formation of an extended tri-EDOT aggregate as shown by the $\operatorname{RDF}(\mathrm{C}-\mathrm{C})$ damped peaks when 64 pairs of EMIM:TCM, EMIM:TCB or EMIM:HCCP are present. Both TCM and HCCP anions are predicted to have the strongest binding energy with tri-EDOT units resulting in the totality of these anions incorporated into the
tri-EDOT cluster 5.21 Such high local density of charges results also in a higher number of EMIM cations bound to anions which can partially disturb $\pi-\pi$ stacking.

Results obtained for tri-EDOT:16SS system are promising and prove that ion exchange predicted for a minimal model of PEDOT:PSS by DFT and PMF calculations may be valid also for a PSS chain.

### 5.4.2.3 Morphology of 6EDOT:16SS solution

A more realistic PEDOT:PSS model is constituted by 6EDOT:16SS system. This improvement in realism comes at the expense of a more challenging and longer equilibration process due to more frequent chain entanglements.


Figure 5.25 - Final snapshots of aqueous 6 EDOT:16SS systems treated with $\times 64$ pairs of EMIM:X. Untreated aqueous 6EDOT:16SS is shown for comparison. Color code: tri-EDOT (blue), 16SS (green), X (red) and EMIM (yellow). Zoom-in snapshots display local structure of PEDOT:PSS aggregates.

By examining the final snapshots in Figure 5.25 we observe a change of the structure in the 6EDOT:16SS complex after indroduction of ILs. We only present the equilibrated systems for EMIM:ES and EMIM:TCB because they are more relevant to draw conclusions. At first sight, it appears that more 6EDOT chains are forming $\pi-\pi$ stacking in presence of TCB anions than in presence of ES anions, which is supported by the zoom-in picture in red frame. The TCB anions stitch together the 6EDOT units organized in an extended $\pi$ - $\pi$ stacking domain. Although ES anion was predicted to be an unsuitable candidate to perform ionic exchange, the resulted system displays a higher stacking of 6EDOT compared to the untreated system.


Figure $5.26-6 \mathrm{EDOT}$ clusters composition for aqueous 6EDOT:16SS systems mixed with 64 pairs of ILs. Top) Composition in presence of EMIM:ES. Bottom) Composition in presence of EMIM:TCB. For comparison, cluster analysis of aqueous 6EDOT:16SS system in absence of IL is displayed in red (6EDOT).

Cluster analysis performed on systems containing $\times 64$ EMIM:ES and EMIM:TCB confirm the trend observed in previous tri-EDOT:PTS and tri-EDOT:16SS systems. More TCB anions ( $\sim 40-$ 48) than ES anions ( $\sim 20-28$ ) bind to the 6EDOT units (Figure 5.26). The radial distribution function $\operatorname{RDF}(\mathrm{C}-\mathrm{S})$ does not change by addition of EMIM: Cl when compared to the $\mathrm{RDF}(\mathrm{C}-\mathrm{S})$ for untreated system (black curve, Figure 5.27.A, right)) even at a high IL concentration (128 pairs $\sim 1 \mathrm{~mol} / \mathrm{l}$ ). A small decrease in $\operatorname{RDF}(\mathrm{C}-\mathrm{S})$ occurs at short distance with addition of EMIM:ES. These ILs have a modest effect on the 6EDOT:16SS complex morphology.

On the other hand, a strongest decrease in $\mathrm{RDF}(\mathrm{C}-\mathrm{S})$ is occuring by addition of EMIM: X where X is TCM, TCB or HCCP. When 64 pairs of ILs are present in the system, $\operatorname{RDF}(\mathrm{C}-\mathrm{S})$ profiles are similar for each IL (blue curves, Figure 5.27,A, left)) but an increase in IL concentration result in a sharper decrease of the $\operatorname{RDF}(\mathrm{C}-\mathrm{S})$ in the order $\mathrm{HCCP}>\mathrm{TCM}>\mathrm{TCB}$ along the series of anions.

The morphology of PEDOT:PSS complexes is investigated by calculating again RDF(C-C) that shows a $\pi-\pi$ stacking of at most $\sim 36$ EDOT chains when $\times 64$ or $\times 128$ pairs of EMIM: Cl (in blue) or EMIM:ES (in green) are present in the system (Figure 5.27.B, top). These RDF(C-C) are not different from the $\mathrm{RDF}(\mathrm{C}-\mathrm{C})$ calculated for untreated 6EDOT:16SS revealing that there is no significant morphology change in PEDOT:PSS matrix by addition of EMIM: Cl or EMIM:ES.

However, after insertion of $\times 64$ EMIM:X pairs, a crystalline domain made of up to 6 EDOT units arises for $\mathrm{X}=\mathrm{TCB}$ as depicted by the sharp peaks in Figure 5.27-B, bottom in red, while


Figure 5.27 - A) Radial distribution $\operatorname{RDF}(\mathrm{C}-\mathrm{S})$ between PEDOT backbone carbon atoms and sulfonate group of PTS and associated coordination number CN(C-S). B) Radial distribution RDF(CC) between closest PEDOT backbone carbon atoms as defined in section 5.2.1 and calculated for the small aqueous solutions of 6EDOT:16SS with ILs.
damped peaks are present for $\mathrm{X}=\mathrm{TCM}$ and no difference with respect to untreated 6EDOT:16SS is observed for $\mathrm{X}=\mathrm{HCCP}$. For higher amount of ILs, the extended crystalline domain vanishes because high IL concentration saturates the system and prevents formation of well ordered $\pi$ stacked PEDOT domains.

As pointed out in previous direct MD simulations, TCB appears to be the best anion to provide both ion exchange between PEDOT and PSS and well ordered $\pi$-stacked PEDOT domains.

### 5.4.3 Conclusion

The ion exchange predicted by PMF and DFT calculations has been investigated with direct (i.e. non biased) MD simulations performed on relatively small systems of mixed PEDOT:PSS with

IL EMIM:X. First, we investigated the same minimal model for PEDOT:PSS (tri-EDOT:PTS) than the one used in PMF and DFT calculations, probing the change in morphology for the different systems by varying the anion X . Our simulations in presence of EMIM: X ILs confirmed that $\mathrm{X}=$ TCM, TCB and HCCP anions are suitable candidates in contrary to $\mathrm{X}=\mathrm{Cl}$ and ES to achieve proper ion exchange followed by nano-segregation between PEDOT and PSS. IL anions play the role of a compaction agent and aggregate tri-EDOT units into a $\pi$-stacked domain surrounded by excess anions in agreement with experimental results (see chapter 2 section 2.2.3.2.

More realistic models for PEDOT:PSS constituted by longer chains, namely tri-EDOT:16SS and 6EDOT:16SS are considered and the same conclusions are drawn as in the case of tri-EDOT:PTS system. However, TCB is predicted to be the most efficient in the series because this anion not only achieves a satisfying ion exchange by segregating properly PEDOT from PSS but also induce well-ordered $\pi-\pi$ stacked domains compared to TCM and HCCP. Indeed, these latter anions were predicted to have a stronger binding energy with PEDOT and direct simulations have shown that this binding induce in turn a layer of negative charges that attract positive EMIM resulting in disordering the packing of the PEDOT units. Non-planar TCB anion prevents such problem.

Our results are consistent with the trends exhibited by experimental conductivity and charge carrier mobility measured on PEDOT:PSS films treated with ILs (see chapter 2 section 2.2.3.2. The conductivity is enhanced both by the increase in the charge carrier mobility $\mu$ resulting from the formation of well ordered PEDOT domains with size increasing for the series of anions in the order $\mathrm{Cl}<\mathrm{ES}<\mathrm{TCM} \sim \mathrm{HCCP}<\mathrm{TCB}$ and by the higher charge carrier density $n$ induced by TCM, TCB and HCCP binding to tri-EDOT domains.

Our simulations confirm also the design principles established from DFT calculations, i.e. IL anions are required to have electron-withdrawing groups to sustain p-doping of PEDOT chains (modeled in MD as negative carbonitrile groups), to be hydrophobic, charge-dispersed, watersoluble and bulky such as TCB. The proposed HCCP anion that carries more carbonitrile groups and was predicted from DFT calculations to be more efficient, displays a similar behavior than TCM in our simulations. Hence, most of EMIM cations bind to the negatively-charged layer formed by TCM or HCCP anions adsorbed on tri-EDOT units, in turn preventing a proper phase segregation between PEDOT and PSS as well as the formation of well-ordered PEDOT domains.

## Conclusion and future work

In this thesis, we studied the stability/dynamic of several self-assembly processes of polyelectrolyte mixtures by using numerical simulations. From a technical point of view, the first project of the thesis involving DNA and gold nanoparticles has been conducted with classical molecular Monte Carlo simulations while the second project involving PEDOT and PSS with and without IL was investigated with classical molecular dynamics simulations. We close this work by presenting the main conclusions and possible future research regarding the two projects that constitute the core of this thesis.

## Project 1: self-assembly processes of gold nanoparticles with DNA

- In order to verify the home-made simulation code, we first reproduce several important results of the literature. The effective force has been calculated between a parallel pair of coarse-grained DNA with counterions of different valencies. As predicted by Manning-Oosawa theory and former simulations and experiments, the MC simulations display repulsive force in presence of monovalent counterions between the parallel DNA while onset of attraction occurs in presence of divalent counterions. Because of correlations induced by multivalent counterions of higher valency condensed on the DNA molecules (trivalent and tetravalent counterions), there is a net attraction with the magnitude of the force increasing with the valency.

Then, we determined the stability of a hexagonal bundle of DNA in presence of counterions
of different valencies by calculating osmotic pressure as a function of the distance between the parallel DNA molecules. The osmotic pressure was positive in presence of monovalent counterions which indicates an unstable bundle while in presence of multivalent counterions a negative osmotic pressure indicates bundle contraction. The results obtained for osmotic pressure calculation follow the same trend as the effective force calculation between the two parallel DNA molecules.

- We probe the adsorption of cationic functionalized gold nanoparticles on a fixed DNA molecule. The adsorption of multivalent gold nanoparticles carrying 6,12 and 30 positive elementary charges is associated to the release of small monovalent DNA counterions which increase the translational entropy of the system

A maximum number of nanoparticles can be adsorbed on the DNA and this number depends on the electrostatic repulsion between nanoparticles adsorbed on the DNA as well as steric interactions to intercalate nanoparticles on the DNA

The maximum number of nanoparticles adsorbed represents a total positive charge that exceeds DNA negative charge. Consequently, we observe the DNA overcharging effect that occurs when multivalent cations are adsorbed on DNA. By integrating the radial charge from the fixed DNA, we found alternation of layers of positive and negative charges that appear to compensate overcharging effect of gold nanoparticles adsorbed on the DNA.

We also characterized the spatial distribution of adsorbed nanoparticles on the DNA. Analysis highlighted that nanoparticles are mostly adsorbed in the major groove and are distributed regularly on the DNA molecules for a sufficient number of nanoparticles present in the system $\left(R_{+/-} \geq 1.00\right)$.

Finally, we show that an excess of salt inserted in the system of DNA with adsorbed nanoparticles (with 6 ligands) results in a competition between the salt cation and the nanoparticles to adsorb on the DNA. We keep monovalent anion and subsequenly increase the valency of the cation and observe desorption of nanoparticles from the major groove and adsorption of salt cation on the strand and in the minor groove.

We calculated also the potential of mean force between a highly charged nanoparticle (30 ligands) and a DNA decorated with the same type of nanoparticles. The nanoparticles
present in the system totally neutralized the DNA phosphate and we only vary the amount of monovalent salt. We show that a larger amount of salt induces a weaker effective attraction between the nanoparticle and the DNA. Either salt around the nanoparticle and the decorated DNA are screening not only the repulsive but also attractive interactions, or a large amount of monovalent salt is adsorbed on the DNA and the "incoming" nanoparticle cannot compete with them for adsorption.

- The effective force between two parallel DNA has been calculated for fixed DNA separation distances in presence of gold nanoparticles carrying different charge distributions and with/without small ions. The force was found maximum at short distance ( $\ell=30 \AA$ ) when the nanoparticles bridge the DNA molecules attracting each other until a stable equilibrium distance ( $\ell<30 \AA$ ). The force was higher for nanoparticles carrying 12 ligands compared to nanoparticles with 6 ligands. A detailed analysis to identify the different contributions to the effective force was performed for the system containing gold nanoparticles with 12 ligands. At low amount of nanoparticles $\left(\mathrm{R}_{+/-}=0.50\right)$, the attractive part of the effective force comes from nanoparticles and positive monovalent ions (the counterion of DNA). When the number of nanoparticles is increased, there is a competition between cations and nanoparticles to attract DNA molecules $\left(\mathrm{R}_{+/-}=1.00\right)$. When nanoparticles are in excess $\left(\mathrm{R}_{+/-}=1.50\right)$, the negative monovalent counterions of nanoparticles destabilize DNA pair formation by adsorbing on the nanoparticles. In that case, the total effective force is still attractive but lower than in systems with fewer nanoparticles.
- By calculating the osmotic pressure, we probe the stability of bundles made of aligned DNA molecules containing a number of nanoparticles that neutralize exactly DNA phosphates $\left(\mathrm{R}_{+/-}=1.00\right)$. We investigate the stability of two lattices that can be formed by DNA (hexagonal or square) with various amount of monovalent ion pairs. The calculated osmotic pressure follows the same trends for both kind of structure lattices. In absence of small ions, the osmotic pressure tends to zero when DNA are far apart and decreases when the bundle is forming (e.g. at small DNA-DNA distances). This observation may suggest a possible spontaneous self-assembly process in absence of ions. Conversely, when various amount of ion pairs are present in the bundle (concentration of 30,60 and $120 \mathrm{mMol} / \mathrm{l}$ ), the osmotic pressure is negative for very compact bundle (i.e. small DNA-DNA distances) but
displays a smaller magnitude values compared to systems without ions, indicating that small ions destabilize the bundle. However, for larger DNA lattice spacing, the osmotic pressure increases and becomes positive indicating bundle spontaneous expansion for such DNA-DNA distance separation. Consequently, small ions prevent spontaneous self-assembly of DNA with gold nanoparticles.
- The first drawback of our simulations remains in the use of a fully rigid model for DNA and nanoparticles. It would be desired to introduce some degrees of flexibility in our models of DNA and ligands attached to the gold nanoparticles. In that case, future simulations with a single DNA would display a change of DNA conformation and we could obtain new structures of DNA-nanoparticles depending of the charge/size of the nanoparticles (e.g. wrapping of DNA around nanoparticles like DNA around histones). The second drawback comes from the numerical aspects of the project. The simulation code has been designed to handle a relatively low number of charges and a supplemental effort would be devoted to parallelize routines that calculate interactions between beads. It would be also required to use some methods in order to speed up the calculation of the electrostatic energy with the Ewald summation method. In particular, the multiple timestep method [309] is adapted in our simulations and consists of evaluating less frequently the long range part of the electrostatic interaction that varies slowly compared to the short range part of the electrostatic interaction. In order to validate the osmotic pressure calculation at a larger scale, it would be desired to perform direct simulations to probe self-assembly dynamics in presence of larger number of DNA and gold nanoparticles that can both translate and rotate in the simulation box.

Project 2: Self-assembly process in PEDOT:PSS with ILs results in conductivity enhancement

- We determined first the morphology of PEDOT:PSS aqueous solutions. Various models of PEDOT:PSS have been employed. We considered a short tri-EDOT or a longer 6EDOT unit as PEDOT chain and PSS was assimilated as its monomer unit PTS or a 16 SS oligomer chain. Small aqueous system of PEDOT:PTS displays several clusters of $\pi$-stacked tri-EDOT units decorated by PTS while at a larger scale, the 6EDOT units form long chains of $\pi$-stacked
units. When PTS is replaced by more realistic 16 SS chain, tri-EDOT units form separated assemblies that bind on the 16 SS chain. At a larger scale, the tri-EDOT:16SS aqueous system forms an amorphous aggregate.
- The ion exchange between PEDOT:PSS and the IL EMIM:X is investigated for the series of IL anions $\mathrm{X}=\mathrm{Cl}, \mathrm{ES}, \mathrm{TCM}, \mathrm{TCB}$ and hypothetical HCCP. The free energy $\Delta \Delta G_{x}$ for ion exchange was quantitatively determined by umbrella sampling calculations performed on separate pairs of molecules by using the tri-EDOT:PTS model. More negative $\Delta \Delta G_{x}$ indicates a more efficient ion exchange between PEDOT:PSS and EMIM:X. The exchange free energy is negative in presence of bulky, charge-dispersed and hydrophobic IL anions (X $=\mathrm{TCM}, \mathrm{TCB}$ and HCCP) and positive in presence of charge-localized and hydrophilic soft IL anions ( $\mathrm{X}=\mathrm{Cl}$ and ES ), which is supported by supplemental analysis of the coordination number of water molecules around key atoms of each IL anion.
- In order to validate the ion exchange at a larger scale, we performed MD simulation on aqueous PEDOT:PSS systems equilibrated in presence of ILs. We showed that at small scale, bulky, charge-dispersed and hydrophobic anion (TCM, TCB and HCCP) gather triEDOT/6EDOT units into an extended $\pi$-stacked cluster surrounded by IL anions followed by a nano-segregation between the tri-EDOT cluster and the PTS units/16SS chains. On the contrary, the hydrophilic Cl and ES anions remain in solution and have a limited effect on the morphology of PEDOT:PSS. At a larger scale, we observe a collapse of the long domain of $\pi$-stacked tri-EDOT units into a compact cluster where IL anions play the role of compaction agent. We observe that Cl and ES anions do not modify the tri-EDOT:PTS complex organization. However, replacing PTS small anions by 16SS chains results in the formation of dense tri-EDOT:16SS and 6EDOT:16SS complexes: if the addition of IL cannot induce a complete segregation of 16SS chains in solution, analysis reveal that it can however, depending of its nature, modify the complex morphology. Indeed, 16SS chains are more segregated from tri-EDOT/6EDOT units in presence of TCM, TCB and HCCP anions while the complex morphology is mostly unaffected by Cl and ES anions.
- The change in morphology of PEDOT:PSS supports the hypothesis of an ion exchange between PEDOT:PSS and ILs. The electrical conductivity enhancement may be correlated
to the change in morphology observed in the MD simulations: larger domains made of $\pi$ stacked PEDOT units favor a higher charge carrier mobility $(\mu)$ while anions decorating the aggregates induce $p$-doping of the PEDOT units resulting in a higher charge carrier density $(n)$. The MD simulations confirm that bulky, hydrophobic but water-soluble IL anions with multiple electron-withdrawing groups would be required to achieve satisfactory conductivity enhancement.

Although we probed the influence of IL anions on the resulted morphology, future investigations will be devoted to find suitable IL cations to further improve ion exchange. In particular, a promising direction would be to replace the aromatic EMIM cation by a protic IL cation that can form hydrogen bonds with PSS favoring decoupling with PEDOT.

Another direction would be to improve the model of PEDOT:PSS by increasing the length of the polymer chains, at the expense of longer equilibration time due to existence of entanglements between PEDOT and PSS chains. The parallel tempering method would be useful in order to overcome the resulting slow dynamic: the idea consists in simulating replica of the system at different temperatures and regularly wrapping temperature of the replicas through a MC scheme [310].

The atomistic description of our systems prevents exploration of larger scale morphologies due to high computational cost to estimate interactions and to long timescale equilibration process. Hence, coarse-grained description of PEDOT:PSS can be used in order to investigate ion exchange at the micrometer scale. Several coarse-grained descriptions of PEDOT:PSS have been proposed in the literature [311, 312].

In addition to provide a high electrical conductivity, the ionic liquid EMIM:TCB has been employed as a plasticizer for PEDOT:PSS in order to produce highly stretchable, semitransparent and solution-processed electronic devices [12]. In order to understand the lower stress that PEDOT:PSS thin films undergo in presence of EMIM:TCB compared to untreated PEDOT:PSS samples, non-equilibrium MD simulation of PEDOT:PSS is required to link the change in PEDOT:PSS morphology with its mechanical properties.

## appendix $A$

## Monte Carlo simulation package

## Contents

A. 1 Program to generate the coarse-grained system ..... 244
A.1.1 Initialization_system.cpp ..... 244
A.1.2 Initialization_system.h ..... 260
A.1.3 Makefile ..... 261
A. 2 Program to generate system topology ..... 262
A.2.1 Builiding topology.cpp ..... 262
A.2.2 Builiding topology.h ..... 270
A.2.3 Makefile ..... 272
A. 3 Program to perform a Monte Carlo simulation ..... 272
A.3.1 Main.cpp ..... 272
A.3.2 Move_MC.cpp ..... 275
A.3.3 Move MC.h ..... 292
A.3.4 Potential_energy_calculation.cpp ..... 296
A.3.5 Potential energy calculation.h ..... 315
A.3.6 Cell_lists.cpp ..... 322
A.3.7 Cell lists.h ..... 329
A.3.8 MersenneTwister. $h$ ..... 330
A. 3.9 Makefile ..... 333
A. 4 Program to visualize a trajectory ..... 333
A.4.1 Main.cpp ..... 333
A.4.2 Visualization_trajectory.cpp ..... 336
A.4.3 Visualization_trajectory.h ..... 340
A.4.4 Read_trajectory.cpp ..... 341
A.4.5 Read_trajectory.h ..... 343
A.4.6 Makefile ..... 344
A. 5 Program to analyze a trajectory ..... 345
A.5.1 Main.cpp ..... 345
A.5.2 Analysis_trajectory.cpp ..... 347
A.5.3 Analysis trajectory.h ..... 360
A.5.4 Makefile ..... 362

In this appendix, we present the "home-made" Monte Carlo simulation package. Altough various modifications have been introduced to perform the different simulations, the following simulation package is adapted to perform an osmotic pressure calculation in a hexagonal DNA lattice with 6 -AuNPs without salt.

## A. 1 Program to generate the coarse-grained system

The program to generate the system is 'Initialization_system.cpp' and corresponding header file is also given as 'Initialization_system.h'.

## A.1.1 Initialization_system.cpp

Listing A. 1 - Initialization_system.cpp

```
    frame
#include <iostream>
#include <vector>
#include <cmath>
#include <gsl/gsl_rng.h>
#include <chrono>
#include <ctime>
#include <iomanip>
#include <fstream>
#include <sstream>
#include "Initialization_System.h"
#define PI 3.14159265
using namespace std;
Init_System ::Init_System()
{}
Init_System::~Init_System()
{}
Init_System::Init_System(double box_x, double box_y, double box_z)
{
    this ->box_x = box_x;
    this }->\mathrm{ box_y = box_y;
    this }->\mathrm{ box_z = box_z;
}
void Init_System::Init__polymer(int N_basis, double z_strand,
    string Residue, double x_init, double y_init, double z_init)
{
    double i;
```

```
    vector<double> buffer(3,0.0);
    double theta = (PI/180.0)*((180-154.4)/2);
    double alpha = (PI/180.0)*(154.4);
    double phi = (PI/180.0)*(36);
// Now, let's construct the rest of the polymer chain
for (i = 1; i <= N__basis; i++)
{
// We initialize the central bead.
buffer.at(0) = x_init;
buffer.at(1) = y_init;
buffer.at(2) = z_init + (i-1) * z_strand;
buffer.at(0) = Check_PBC_Box(buffer.at(0), box_x);
buffer.at(1) = Check_PBC_Box(buffer.at(1), box_y);
buffer.at(2) = Check_PBC_Box(buffer.at(2), box_z);
coordinates.push_back(buffer);
index_atom.push_back(index_atom.size() + 1);
atomic_number.push_back(100);
name_res.push_back(Residue);
charge.push_back(0);
radius_DNA.push_back (3.9);
// Initialisation of the inner balls
buffer.at(0) = x_init + 5.9 * cos(-theta -alpha +((i-1) * phi));
buffer.at(1)= y_init + 5.9 * sin(-theta -alpha +((i-1) * phi));
buffer.at(2) = z_init + ((i-1) * z_strand);
buffer.at(0)= Check_PBC_Box(buffer.at(0), box_x);
buffer.at(1) = Check_PBC_Box(buffer.at(1), box_y);
buffer.at(2)= Check_PBC_Box(buffer.at(2), box_z);
coordinates.push_back(buffer);
index_atom.push_back(index_atom.size() + 1);
atomic_number.push_back (16);
name_res.push_back(Residue);
charge.push_back(0);
radius_DNA.push_back (2.1);
buffer.at(0) = x_init + 5.9 * cos(-theta +((i-1) * phi));
buffer.at(1) = y_init + 5.9 * sin(-theta +((i-1) * phi));
buffer.at(2) = z_init + ((i -1) * z_strand) + 0.78;
buffer.at(0)= Check_PBC_Box(buffer.at(0), box_x);
buffer.at(1) = Check_PBC_Box(buffer.at(1), box_y);
buffer.at(2) = Check_PBC_Box(buffer.at(2), box_z);
coordinates.push_back(buffer);
index_atom.push_back(index_atom.size() + 1);
atomic_number.push_back (16);
```

```
    name__res.push__back(Residue);
    charge.push__back (0);
    radius__DNA.push__back(2.1);
    // Initialisation of the outer balls (charged)
    buffer.at(0) = x__init + 8.9* cos(-theta -alpha + ((i-1) * phi ));
    buffer.at(1) = y__init + 8.9 * sin(-theta -alpha +((i-1) * phi));
    buffer.at(2) = z__init + ((i-1) * z__strand );
    buffer.at(0) = Check_PBC__Box(buffer.at(0), box__x);
    buffer.at(1) = Check_PBC__Box(buffer.at(1), box_y);
    buffer.at(2) = Check_PBC__Box(buffer.at(2), box__z);
    coordinates.push__back(buffer)
    index__atom.push__back(index__atom.size() + 1);
    atomic__number.push__back (8);
    name__res.push__back(Residue);
    charge.push__back(-1);
    radius__DNA.push__back(2.1);
    buffer.at(0)= x_init + 8.9 * cos(-theta +((i-1)* phi));
    buffer.at(1) = y_init + 8.9 * sin(-theta +((i-1)* phi));
    buffer.at(2) = z_init + ((i - ) * z_strand ) + 0.78;
    buffer.at(0)= Check_PBC__Box(buffer.at(0), box__x);
    buffer.at(1) = Check_PBC__Box(buffer.at(1), box_y);
    buffer.at(2) = Check_PBC__Box(buffer.at(2), box_z);
    coordinates.push__back(buffer)
    index__atom.push__back(index__atom.size() + 1);
    atomic_number.push__back (8);
    name__res.push__back(Residue);
    charge.push__back(-1);
    radius__DNA.push__back(2.1);
}
// In a separate way, we update the connectivity of the chain.
vector<int> buffer__connectivity;
for (i = 1; i <= N__basis; i++ )
{
    // We derive the connectivity of the inner neutral balls
    buffer__connectivity.clear();
    buffer__connectivity.push__back(index__atom.at (4*(i - 1)));
    buffer__connectivity.push__back(index__atom.at (4*(i - 1)) +3);
    connectivity.push__back(buffer__connectivity);
    buffer__connectivity.clear();
    buffer__connectivity.push__back(index__atom.at (4*(i - 1)) +1);
    buffer__connectivity.push__back(index__atom.at (4*(i - 1))+2);
    connectivity.push__back(buffer__connectivity);
        // We derive the connectivity of each phosphate
```

```
if( i == 1 )
{
    buffer_connectivity.clear();
    buffer_connectivity.push_back(index_atom.at (4*(i - 1))+2);
    buffer_connectivity.push_back(index_atom.at (4*(i-1))+1);
    buffer_connectivity.push_back(index_atom.at(4*(i))+2);
    connectivity.push_back(buffer_connectivity);
    buffer_connectivity.clear();
    buffer_connectivity.push_back(index_atom.at (4*(i - 1))+3);
    buffer_connectivity.push_back(index_atom.at(4*(i - 1)));
    buffer_connectivity.push_back(index_atom.at (4*(i))+3);
    connectivity.push_back(buffer_connectivity);
}
else
    {
        if( i = N__basis )
        {
            buffer_connectivity.clear();
            buffer_connectivity.push_back(index_atom.at (4*(i-1))+2);
            buffer_connectivity.push_back(index_atom.at (4*(i - 1))+1);
            buffer_connectivity.push_back(index_atom.at (4*(i-2))+2);
            connectivity.push_back(buffer_connectivity);
            buffer_connectivity.clear();
            buffer_connectivity.push_back(index_atom.at (4*(i - 1))+3);
            buffer_connectivity.push_back(index_atom.at (4*(i - 1)));
            buffer_connectivity.push_back(index_atom.at (4*(i - 2))+3);
            connectivity.push_back(buffer_connectivity);
        }
        else
        {
            buffer_connectivity.clear();
                buffer_connectivity.push_back(index_atom.at (4*(i-1))+2);
                buffer_connectivity.push_back(index_atom.at (4*(i-1))+1);
                buffer_connectivity.push_back(index_atom.at (4*(i - 2))+2);
                buffer_connectivity.push_back(index_atom.at (4*(i))+2);
                connectivity.push_back(buffer_connectivity);
                buffer_connectivity.clear();
                buffer_connectivity.push_back(index_atom.at (4*(i-1))+3);
                buffer_connectivity.push_back(index_atom.at (4*(i - 1)));
                buffer_connectivity.push_back(index_atom.at (4*(i - 2))+3);
                buffer_connectivity.push_back(index_atom.at (4*(i))+3);
                connectivity.push_back(buffer_connectivity);
            }
    }
```

\}

195
196
197
\}
bool Init_System: Add__Free__Ion(double radius, string Residue,
double q, int atomic__n, double x_ion, double y__ion, double z_ion)
\{
bool IsSetUpProperly $=$ true;
vector<int> buffer__connectivity;
vector <double> buffer $(3,0)$;
int atoms;
double r_square $=0.0$;
double r_m_square $=5 * 5$;
// Test if the atom is inserted properly in the box
if $\left(x \_i o n<(\right.$ box_x $\left.-(1.5)) \& \& x \_i o n>(0+(1.5))\right)$
\{
buffer.at $(0)=x \_i o n ;$
\}
else
\{
IsSetUpProperly $=$ false;
return IsSetUpProperly;
\}
if $\left(y \_\right.$ion $<($box_y $-(1.5)) \& \&$ y__ion $\left.>(0+(1.5))\right)$
\{
buffer.at (1) = y_ion;
\}
else\{
IsSetUpProperly $=$ false;
return IsSetUpProperly;
\}
if $\left(\mathrm{z} \_\right.$ion $<($box_z $-(1.5))$ \& $\mathrm{z} \_$ion $\left.>(0+(1.5))\right)$
\{
buffer.at (2) = z_ion;
\}
else\{
IsSetUpProperly $=$ false;
return IsSetUpProperly;
\}
// Test if there is interpenetration between the DNAs
// and the inserted ions.
for ( atoms $=0 ;$ atoms $<$ coordinates.size () ; atoms ++ )
\{
$r_{\text {_square }}=$ pow (coordinates.at (atoms) $\left.[0]-\operatorname{buffer} . \operatorname{at}(0), 2\right)+$
pow (coordinates.at (atoms) [1] - buffer.at (1), 2) +
pow (coordinates.at (atoms) [2] - buffer.at (2), 2);
if (r_square $<(($ radius_DNA. at $($ atoms $)+1.5) *($ radius_DNA. at (atoms $)+1.5)))$
\{
return false;
\}
\}
// Test if there is interpenetration between the inserted ion
// and the gold nanoparticles.
for $($ atoms $=0 ;$ atoms $<$ coordinates_gold. size (); atoms ++ )
\{
r_square $=$ pow (coordinates_gold.at (atoms) $[0]-$ buffer.at (0), 2) +
pow (coordinates_gold.at (atoms) [1] - buffer.at (1), 2) +
pow (coordinates_gold.at (atoms) [2] - buffer.at (2), 2);
if $\left(\mathrm{r} \_\right.$square $\left.<(15 * 15)\right)$
\{
return false;
\}
\}
// Test if there is interpenetration between the previous ions and
// the current one.
for $($ atoms $=0 ;$ atoms $<$ coordinates_free__ions.size (); atoms ++ )
\{
$r_{\text {_ square }}=$ pow (coordinates_free_ions.at (atoms) [0] - buffer.at (0), 2) +
pow (coordinates_free_ions.at (atoms) [1] - buffer.at (1), 2) +
pow (coordinates_free_ions.at (atoms) [2] - buffer.at (2), 2);
if $\left(r \_\right.$square $\left.<(4 * 4)\right)$
\{
return false;
$\}$
\}
// We update the coordinates with the new atom
coordinates_free__ions.push__back(buffer);
// We update the attributes of the object
index__atom_free__ions.push__back(index__atom_free__ions.size () + 1);
atomic__number_free_ions.push_back (atomic_n );
name_res_free__ions.push__back (Residue) ;
charge_free__ions.push__back(q);
buffer_connectivity.push_back(index_atom_free_ions.size ()) ;
connectivity_free__ions.push__back(buffer_connectivity);
return IsSetUpProperly;
\}
bool Init__System: Add__Gold_Nano(double radius__gold, double radius__surf,
string Residue, double $q, i \bar{n} t$ atomic__gold, int atomic_branch,
int atomic__ammonium, double x_ion, double y__ion, double z_ion,
double cos_theta, double sin_theta, double phi, int ligands,
int count__success_insertion__gold)
\{
bool IsSetUpProperly $=$ true;
vector $<$ int $>$ buffer_connectivity;
vector <double> buffer_gold (3,0);

```
vector <double> buffer_surf(3,0);
int atoms;
double r_square = 0.0;
double total_radius_gold_nano = 11.5;
```



```
/********************************************************************
/******* Insert gold core in the box ********/
/****************************************************************/
// Test if the atom is inserted properly in the box
if(x_ion < (box_x - total_radius_gold__nano) &&
x_ion > (0 + total_radius_gold__nano))
{
    buffer_gold.at(0) = x_ion;
}
else
{
    IsSetUpProperly = false;
    return IsSetUpProperly;
}
if(y_ion < (box_y - total_radius_gold__nano) &&
y_ion > (0 + total_radius_gold_nano))
{
    buffer_gold.at(1) = y_ion;
}
else{
    IsSetUpProperly = false;
    return IsSetUpProperly;
}
if(z_ion < (box_z - total_radius_gold_nano) &&
z_ion > (0 + total_radius_gold__nano))
{
    buffer_gold.at(2) = z_ion;
}
else{
    IsSetUpProperly = false;
    return IsSetUpProperly;
}
// Test if there is interpenetration between previous inserted atoms
// and the current ones.
for( atoms = 0; atoms < coordinates.size(); atoms ++ )
{
    r_square = pow(coordinates.at(atoms)[0] - buffer_gold.at(0),2) +
                        pow(coordinates.at(atoms)[1] - buffer_gold.at(1),2) +
                        pow(coordinates.at(atoms)[2] - buffer_gold.at(2),2);
    if(r_square < ((radius_DNA.at(atoms)+11.5)*(radius_DNA.at(atoms)+11.5)))
    {
        return false;
    }
```

```
}
// Test if there is interpenetration between previous inserted atoms
// and the current ones
for( atoms = 0; atoms < coordinates_gold.size(); atoms = atoms + 25 )
{
    r_square = pow(coordinates_gold.at(atoms)[0] - buffer_gold.at(0),2) +
                    pow(coordinates_gold.at(atoms)[1] - buffer_gold.at(1),2) +
                            pow(coordinates_gold.at(atoms)[2] - buffer_gold.at(2),2);
        if(r_square < (23*23))
        {
        return false;
        }
}
// Test if there is interpenetration between the inserted
// ion and the previous ions
for( atoms = 0; atoms < coordinates_free_ions.size(); atoms ++ )
{
        r_square = pow(coordinates_free_ions.at(atoms)[0] - buffer_gold.at(0),2) +
                pow(coordinates_free_ions.at(atoms)[1] - buffer_gold.at(1),2) +
                pow(coordinates_free_ions.at(atoms)[2] - buffer_gold.at(2),2);
        if(r_square < (13*13))
    {
        return false;
    }
}
// We update the coordinates with gold nanoparticle core.
coordinates_gold.push_back(buffer_gold);
// We update the attributes of the object for the gold nanoparticle
index_atom_gold.push_back(index_atom_gold.size() + 1);
atomic_number_gold.push_back(atomic_gold);
name_res_gold.push_back(Residue);
charge_gold.push_back (0.0);
/****************************************************************/
/******** Insert ligands to AuG ********/
/**************************************************************/
// call random number generator.
double cos_theta_on_gold;
double sin_theta_on_gold;
double phi_on_gold;
int neutral_atoms = 3;
// Construction of the ligands around one gold nanoparticles.
// construct serie of 4 for theta = PI/2 and PHI = PI/2
    for (int i_ligands = 0; i_ligands < 4; i_ligands++)
    {
```

```
    double theta = PI/2;
```

    double theta = PI/2;
    double phi = PI/2;
    double phi = PI/2;
    // construct the branch with the neutral atoms
    // construct the branch with the neutral atoms
    for(int j = 0; j < neutral__atoms; j++)
    for(int j = 0; j < neutral__atoms; j++)
    {
    {
        // Select a surface atom
        // Select a surface atom
        buffer_surf.at(0) = buffer_gold.at(0) +
        buffer_surf.at(0) = buffer_gold.at(0) +
        (radius_gold + double(j)*2.0/3
        (radius_gold + double(j)*2.0/3
        *radius_surf) * sin(theta) * cos(i_ligands * phi);
        *radius_surf) * sin(theta) * cos(i_ligands * phi);
        buffer_surf.at(1) = buffer_gold.at(1) +
        buffer_surf.at(1) = buffer_gold.at(1) +
        (radius_gold + double(j)*2.0/3
        (radius_gold + double(j)*2.0/3
        *radius_surf) * sin(theta) * sin(i__ligands * phi);
        *radius_surf) * sin(theta) * sin(i__ligands * phi);
        buffer_surf.at(2) = buffer_gold.at(2) +
        buffer_surf.at(2) = buffer_gold.at(2) +
        (radius_gold + double(j)*2.0/3
        (radius_gold + double(j)*2.0/3
        *radius_surf) * cos(theta);
        *radius_surf) * cos(theta);
            // We update the coordinates with the surf containing the charge.
            // We update the coordinates with the surf containing the charge.
            coordinates_gold.push_back(buffer__surf);
            coordinates_gold.push_back(buffer__surf);
            // We update the attributes of the object for the surface
            // We update the attributes of the object for the surface
            index_atom_gold.push_back(index_atom_gold.size() + 1);
            index_atom_gold.push_back(index_atom_gold.size() + 1);
            atomic_number_gold.push_back(atomic__branch );
            atomic_number_gold.push_back(atomic__branch );
            name_res_gold.push_back(Residue);
            name_res_gold.push_back(Residue);
            charge_gold.push_back(0.0);
            charge_gold.push_back(0.0);
        }
        }
    // construct the final atom
    // construct the final atom
    // Select a surface atom
    // Select a surface atom
    buffer_surf.at(0) = buffer_gold.at(0) +
    buffer_surf.at(0) = buffer_gold.at(0) +
    (radius_gold + double(neutral_atoms) * 2/3*radius_surf) *
    (radius_gold + double(neutral_atoms) * 2/3*radius_surf) *
        sin(theta) * cos(i_ligands * phi);
        sin(theta) * cos(i_ligands * phi);
    buffer_surf.at(1) = buffer_gold.at(1) +
    buffer_surf.at(1) = buffer_gold.at(1) +
        (radius_gold + double(neutral_atoms) * 2/3*radius_surf) *
        (radius_gold + double(neutral_atoms) * 2/3*radius_surf) *
        sin(theta) * sin(i_ligands * phi);
        sin(theta) * sin(i_ligands * phi);
    buffer_surf.at(2) = buffer_gold.at(2) +
    buffer_surf.at(2) = buffer_gold.at(2) +
        (radius_gold + double(neutral_atoms) * 2/3*radius_surf) * cos(theta);
        (radius_gold + double(neutral_atoms) * 2/3*radius_surf) * cos(theta);
    // We update the coordinates with the surf containing the charge.
    // We update the coordinates with the surf containing the charge.
    coordinates_gold.push_back(buffer__surf);
    coordinates_gold.push_back(buffer__surf);
    // We update the attributes of the object for the surface
    // We update the attributes of the object for the surface
    index_atom_gold.push_back(index_atom_gold.size() + 1);
    index_atom_gold.push_back(index_atom_gold.size() + 1);
    atomic_number_gold.push_back(atomic_ammonium);
    atomic_number_gold.push_back(atomic_ammonium);
    name_res_gold.push__back(Residue);
    name_res_gold.push__back(Residue);
    charge_gold.push_back(q);
    charge_gold.push_back(q);
    }
    }
    // construct serie of 2 for theta = (0 or 1)PI.
// construct serie of 2 for theta = (0 or 1)PI.
for(int k = 0; k <= 1; k++)
for(int k = 0; k <= 1; k++)
{
{
for (int i_ligands = 0; i_ligands < 1; i_ligands++)
for (int i_ligands = 0; i_ligands < 1; i_ligands++)
{

```
    {
```

```
double theta = 0 + k*PI;
double phi = PI/2;
    // construct the branch with the neutral atoms
    for(int j = 0; j < neutral_atoms; j++)
    {
    // Select a surface atom
        buffer_surf.at(0) = buffer_gold.at(0) +
        (radius_gold + double(j)*2/3*radius_surf) *
        sin(theta) * cos(i_ligands * phi );
        buffer_surf.at(1) = buffer_gold.at(1) +
        (radius_gold + double(j)*2/3*radius_surf) *
        sin(theta) * sin(i_ligands * phi );
        buffer_surf.at(2) = buffer_gold.at(2) +
        (radius_gold + double(j)*2/3*radius_surf) * cos(theta);
        // We update the coordinates with the surf containing the charge.
        coordinates_gold.push_back(buffer_surf);
        // We update the attributes of the object for the surface
        index_atom_gold.push_back(index_atom_gold.size() + 1);
        atomic_number_gold.push_back(atomic_branch );
        name_res_gold.push_back(Residue);
        charge_gold.push_back(0.0);
    }
// construct the final atom
// Select a surface atom
buffer__surf.at(0) = buffer__gold.at(0) +
(radius__gold + double(neutral__atoms) * 2/3*radius__surf) *
sin(theta) * cos(i__ligands * phi );
buffer__surf.at(1) = buffer__gold.at(1) +
(radius__gold + double(neutral__atoms) * 2/3*radius__surf) *
sin(theta) * sin(i__ligands * phi );
buffer_surf.at(2) = buffer_gold.at(2) +
(radius_gold + double(neutral_atoms) * 2/3*radius_surf) *
cos(theta);
// We update the coordinates with the surf containing the charge.
coordinates__gold.push__back(buffer__surf);
// We update the attributes of the object for the surface
index__atom__gold.push__back(index__atom__gold.size() + 1);
atomic__number__gold.push__back(atomic__ammonium);
name__res__gold.push__back(Residue);
charge__gold.push__back(q);
}
```

\}

```
// Adaptation of the connectivity.
//We construct the connectivity of the core AuG.
buffer_connectivity.push_back(index_atom_gold.at(0) +
(count_success_insertion_gold * (1 + ligands * 4)));
for(int i = 0; i < ligands; i++)
{
        buffer_connectivity.push_back(index_atom_gold.at(1 + i*4) +
        (count_success_insertion_gold * (1 + ligands * 4)));
}
connectivity_gold.push_back(buffer_connectivity);
buffer_connectivity.clear();
//We construct the connectivity of the ligand.
for(int i = 0; i < ligands; i++)
{
        buffer_connectivity.push_back(index_atom_gold.at(1 + i*4) +
        (count_success_insertion_gold * (1 + ligands * 4)));
        buffer_connectivity.push_back(index_atom_gold.at(0) +
        (count_success_insertion_gold * (1 + ligands * 4)));
        buffer_connectivity.push_back(index_atom_gold.at(2 + i*4) +
        (count_success_insertion_gold * (1 + ligands * 4)));
        connectivity_gold.push_back(buffer_connectivity);
        buffer_connectivity.clear();
        buffer_connectivity.push_back(index_atom_gold.at(2 + i*4) +
        (count_success_insertion_gold * (1 + ligands * 4)));
        buffer_connectivity.push_back(index_atom_gold.at(1 + i*4) +
        (count_success_insertion_gold * (1 + ligands * 4)));
        buffer_connectivity.push_back(index_atom_gold.at(3 + i*4) +
        (count_success_insertion_gold * (1 + ligands * 4)));
        connectivity_gold.push_back(buffer_connectivity);
        buffer_connectivity.clear();
        buffer_connectivity.push_back(index_atom_gold.at(3 + i*4) +
        (count_success_insertion_gold * (1 + ligands * 4)));
        buffer_connectivity.push_back(index_atom_gold.at(2 + i*4) +
        (count_success_insertion_gold * (1 + ligands * 4)));
        buffer_connectivity.push_back(index_atom_gold.at(4 + i*4) +
        (count_success_insertion_gold * (1 + ligands * 4)));
        connectivity_gold.push__back(buffer_connectivity);
```

```
    buffer_connectivity.clear();
        buffer_connectivity.push_back(index_atom_gold.at(4 + i*4) +
        (count_success_insertion_gold * (1 + ligands * 4)));
        buffer_connectivity.push_back(index_atom_gold.at(3 + i*4) +
        (count_success_insertion_gold * (1 + ligands * 4)));
        connectivity_gold.push_back(buffer__connectivity);
        buffer_connectivity.clear();
    }
    // Clear the vector
    buffer_gold.clear();
    buffer_surf.clear();
    return IsSetUpProperly;
}
void Init_System:: Reorganize_order()
{
double initial_coordinates_size = coordinates.size();
// Reorganize vector of the free_ions added to total coordinates.
for(int i = 0; i < coordinates_free_ions.size(); i++)
{
        coordinates.push_back(coordinates_free_ions[i]);
        index_atom.push_back(index_atom_free_ions.at(i) + initial_coordinates_size);
        atomic__number.push_back(atomic_number_free_ions.at(i));
        name_res.push_back(name_res_free_ions.at(i));
        charge.push_back(charge_free_ions.at(i));
}
for(int i = 0; i < connectivity_free_ions.size(); i++)
{
    for(int j = 0; j < connectivity_free_ions[i].size(); j++)
    {
            connectivity_free_ions.at(i)[j] += initial_coordinates_size;
        }
        connectivity.push_back(connectivity_free_ions[i]);
    }
    // Update the initial coordinate size
initial_coordinates_size = coordinates.size();
// Reorganize vector of the gold nanoparticle added to total coordinates.
for(int i = 0; i < coordinates_gold.size(); i++)
{
    coordinates.push_back(coordinates_gold[i]);
    index_atom.push_back(index_atom_gold.at(i) + initial_coordinates__size);
```

627 628 629

```
        atomic_number.push_back(atomic_number_gold.at(i));
```

        atomic_number.push_back(atomic_number_gold.at(i));
        name_res.push_back(name_res_gold.at(i));
        name_res.push_back(name_res_gold.at(i));
        charge.push_back(charge_gold.at(i));
        charge.push_back(charge_gold.at(i));
    }
    }
    for(int i = 0; i < connectivity_gold.size(); i++)
    for(int i = 0; i < connectivity_gold.size(); i++)
    {
    {
        for(int j = 0; j < connectivity_gold[i].size(); j++)
        for(int j = 0; j < connectivity_gold[i].size(); j++)
        {
        {
            connectivity_gold.at(i)[j] += initial_coordinates_size;
            connectivity_gold.at(i)[j] += initial_coordinates_size;
        }
        }
        connectivity.push_back(connectivity_gold[i]);
        connectivity.push_back(connectivity_gold[i]);
    }
    }
    }
void Init_System::Output_System(string output_system)
void Init_System::Output_System(string output_system)
{
ofstream oFS;
ofstream oFS;
oFS.open(output_system.c__str());
oFS.open(output_system.c__str());
if (oFS)
if (oFS)
{
{
oFS << "SYSTEM_CONSTRUCTION" << endl;
oFS << "SYSTEM_CONSTRUCTION" << endl;
oFS << coordinates.size() << endl;
oFS << coordinates.size() << endl;
oFS << fixed << setprecision(2) <<box_x <<" "<< box_y <<" "<< box_z << endl;
oFS << fixed << setprecision(2) <<box_x <<" "<< box_y <<" "<< box_z << endl;
for( int site = 0; site < index_atom.size(); site++ )
for( int site = 0; site < index_atom.size(); site++ )
{
{
oFS << index_atom.at(site) <<" "<< atomic_number.at(site) <<" ";
oFS << index_atom.at(site) <<" "<< atomic_number.at(site) <<" ";
oFS << fixed << setprecision(20) << coordinates.at(site)[0] <<"'
oFS << fixed << setprecision(20) << coordinates.at(site)[0] <<"'
"<< coordinates.at(site)[1] <<" "<< coordinates.at(site)[2] <<" ";
"<< coordinates.at(site)[1] <<" "<< coordinates.at(site)[2] <<" ";
oFS << fixed << setprecision(2) << charge.at(site) <<" ";
oFS << fixed << setprecision(2) << charge.at(site) <<" ";
oFS << name_res.at(site) << endl;
oFS << name_res.at(site) << endl;
}
}
oFS << "CONNECTIVITY_IN_THE_SYSTEM" << endl;
oFS << "CONNECTIVITY_IN_THE_SYSTEM" << endl;
for( int connection = 0; connection < connectivity.size(); connection++ )
for( int connection = 0; connection < connectivity.size(); connection++ )
{
{
for (int sub_connection = 0; sub_connection <
for (int sub_connection = 0; sub_connection <
connectivity[connection].size(); sub_connection++ )
connectivity[connection].size(); sub_connection++ )
{
{
oFS << connectivity.at(connection)[sub_connection] <<" ";
oFS << connectivity.at(connection)[sub_connection] <<" ";
}
}
oFS << endl;
oFS << endl;
}
}
}
}
else
else
{
{
cout << "ERROR: Cannot open "<< output_system <<" file" << endl;

```
        cout << "ERROR: Cannot open "<< output_system <<" file" << endl;
```

```
    }
    oFS.close();
    oFS.clear();
}
void Init_System::Output_Residue(string output__residue)
{
    cout << "writing residue list to " << output_residue << "..." << endl;
    int counter = 0;
    ofstream oFS;
    oFS.open(output_residue.c_str());
    vector <string> list_residue;
    vector <int> atom_in_residue;
    // we count the different numbe of residue
    int count = 1;
    for( int i = 0; i < name_res.size()-1; i++)
    {
        if (name_res.at(i) != name_res.at(i+1)) count++;
    }
    //We separate the case where there is one residue from the case
    // where there are several.
    if( count == 1 )
    {
        list_residue.push_back(name_res.at (0));
        atom_in_residue.push_back(name_res.size());
    }
    else
    {
        int counter = 0;
        for( int atom = 0; atom < index_atom.size()-1 ; atom++ )
        {
            counter++;
            if (name_res.at(atom) != name_res.at(atom+1))
            {
            list_residue.push_back(name_res.at(atom));
            atom_in_residue.push_back(counter);
            counter = 0;
                }
        }
        //For the last residue.
        counter = 0;
        for(int i = 0; i < index__atom.size(); i++)
        {
            if (name_res.at(i) = name_res.at(index_atom.size() - 1))
            {
                counter++;
```

```
                }
            }
        list_residue.push_back(name_res.at(index_atom.size() - 1));
        atom_in_residue.push_back(counter);
    }
    if (oFS)
    {
        oFS << "RESIDUE_LIST" << endl;
        for( int j = 0; j < list__residue.size(); j++ )
        {
                oFS << atom_in_residue.at(j) <<" "<< list_residue.at(j) << endl;
        }
    }
    else
    {
        cout << "ERROR: Cannot open "<< output_residue <<" file" << endl;
    }
    oFS.close();
    oFS.clear();
}
void Init_System::Output_Connectivity(string output_connectivity)
{
    cout << " writing connectivity to " << output_connectivity << "..." << endl;
    ofstream oFS;
    oFS.open(output_connectivity.c__str());
    if (oFS)
    {
        for( int connection = 0; connection < connectivity.size(); connection++ )
        {
            for (int sub_connection = 0; sub_connection <
            connectivity[connection].size(); sub_connection++ )
            {
                oFS << connectivity.at(connection)[sub_connection] <<" ";
            }
            oFS << endl;
        }
    }
    else
    {
        cout << "ERROR: Cannot open "<< output_connectivity <<" file" << endl;
    }
    oFS.close();
    oFS.clear();
}
```

```
789
7 9 0
7 9 1
792
793
7 9 4
795
796
797
798
799
800
801
802
803
804
805
806
```

vector <vector<double> > * Init_System::getTraj()

```
vector <vector<double> > * Init_System::getTraj()
{
{
        return &(coordinates);
        return &(coordinates);
}
}
vector <int> * Init_System::getIndexAtoms()
vector <int> * Init_System::getIndexAtoms()
{
{
        return &(index_atom);
        return &(index_atom);
}
}
vector <int> * Init_System::getTypeAtoms()
vector <int> * Init_System::getTypeAtoms()
{
{
        return &(atomic_number);
        return &(atomic_number);
}
}
vector <string> * Init_System::getNameResidue()
vector <string> * Init_System::getNameResidue()
{
{
        return &(name_res);
        return &(name_res);
}
}
vector <double> * Init_System::getCharge()
vector <double> * Init_System::getCharge()
{
{
        return &(charge);
        return &(charge);
}
}
vector <vector<int> > * Init_System::getConnectivity()
vector <vector<int> > * Init_System::getConnectivity()
{
{
        return &(connectivity);
        return &(connectivity);
}
}
double Init_System::Check_PBC_Box(double position, double box_direction)
double Init_System::Check_PBC_Box(double position, double box_direction)
{
{
        double output_position;
        double output_position;
        if ( position > box_direction )
        if ( position > box_direction )
        {
        {
            output_position = position - box_direction;
            output_position = position - box_direction;
        }
        }
        else
        else
        {
        {
            if ( position < 0 )
            if ( position < 0 )
            {
            {
                output_position = position + box_direction;
                output_position = position + box_direction;
            }
            }
            else
            else
            {
            {
                output_position = position;
                output_position = position;
            }
            }
        }
        }
        return output_position;
        return output_position;
}
```

}

```

\section*{A.1.2 Initialization_system.h}

\section*{Listing A. 2 - Initialization_system.h}
```

    frame
    /*
*

* Initialisation of the system - declaration
* 

*/
\#ifndef INIT_SYSTEM_H
\#define INIT_SYSTEM_H
\#include <vector>
using namespace std;
class Init_System
{
private:
/* size x box */
double box_x;
/* size y box */
double box_y;
/* size z box */
double box_z;
/********* vector containing the elements of our system. **********/
/* Vector containing coordinate of the N_chain */
vector <vector<double> > coordinates;
/* Vector containing the index of each atom */
vector <int> index_atom;
/* Vector containing the name of each atom */
vector <int> atomic_number;
/* Vector containing the name of residue */
vector <string> name_res;
/* Vector containing the name of residue */
vector <double> charge;
/* Vector containing connectivity of each atom */
vector <vector<int>> connectivity;
/******** buffer for the free_ions *********/
vector <vector<double> > coordinates_free_ions;
vector <int> index_atom_free_ions;
vector <int> atomic__number_free_ions;

```
```

    vector <string> name__res_free__ions;
    vector <double> charge__free__ions;
    vector <vector<int>> connectivity_free__ions;
    /******** buffer for the gold nanoparticles *********/
    vector <vector <double > > coordinates_gold;
    vector <int> index__atom__gold;
    vector <int> atomic__number__gold;
    vector <string> name__res__gold;
    vector <double> charge__gold;
    vector <vector <int> > connectivity_gold;
    vector<double> radius__DNA;
    vector < double > radius__ions;
    vector<double> radius__AuNPS;
    public:
    Init__System();
    ~Init_System();
    Init__System(double box_x, double box_y, double box__z);
    void Init__polymer(int N__basis, double z_strand, string Residue,
        double x__init, double y__init, double z__init);
    bool Add__Free_Ion(double radius, string Residue, double charge,
        int q, double x_ion, double y_ion, double z_ion);
    bool Add__Gold_Nano(double radius__gold, double radius__surf,
            string Residue, double q, int atomic__gold, int atomic__branch,
            int atomic__ammonium, double x__ion, double y__ion, double z__ion,
            double cos_theta, double sin__theta, double phi,
            int ligands, int count__success__insertion__gold);
    void Reorganize__order();
void Output__System(string output__system);
void Output__Residue(string output__residue);
void Output Connectivity(string output_connectivity);
vector <vector <double> > * getTraj();
vector <int> * getIndexAtoms();
vector <int> * getTypeAtoms();
vector <string> * getNameResidue();
vector <double> * getCharge()
vector <vector<int> > * getConnectivity ();
double Check_PBC_Box(double position, double box__direction);
};
\#endif

```

\section*{A.1.3 Makefile}

Listing A. 3 - Makefile
frame
exe: main.o Initialization_System.o
g+ -o exe main.o Initialization_System.o -O3 - L/home/ambroise/gsl/lib
-lgsl - lgslcblas -lm -static-libstdc++
```

main.o: main.cpp Initialization__System.h
g++ -o main.o -c main.cpp -std=c}++1
Initialization__System.o: Initialization__System.cpp Initialization__System.h
g++ -o Initialization__System.o -c Initialization__System.cpp -std=c + +11
clean:
rm -rf *.o

```

\section*{A. 2 Program to generate system topology}

The program to generate the system topology is 'Builiding_topology.cpp' and the corresponding header file is also given as 'Building_topology.h'.

\section*{A.2.1 Builiding_topology.cpp}

Listing A. 4 - Builiding_topology.cpp
```

    frame
    \#include "building_topology.h"
\#include <cmath>
Topology:: Topology(string input__system, string init__gro__file,
string topol__system)
{
this }->\mathrm{ -input_system = input_system;
this ->init_gro__file = init__gro__file;
this }->\mathrm{ topol__system = topol__system;
read__Bonded__File("interaction__table/bonded.txt");
read__nonBonded__File("interaction__table/non__bonded.txt");
}
Topology::~ Topology()
{}
void Topology::read__Input_System()
{
ifstream iFS;
iFS.open(input_system.c_str());
string title__init__system = " ";
string title__connection = " ";
string line;
int atom__names__buffer = 0;
int atomic__number__buffer = 0;
vector <double> coordinates__buffer ( 3,0.0);
double charge__buffer = 0.0;
string residue__buffer = " ";
vector <int> connection__buffer;
if(iFS)
{
iFS >> title__init__system;

```
```

        iFS >> atoms;
        iFS >> box_x >> box_y >> box_z;
        for ( int i = 0; i < atoms; i++ )
        {
        iFS >> atom__names__buffer >> atomic__number__buffer >>
        coordinates__buffer.at (0) >> coordinates__buffer.at (1) >>
        coordinates__buffer.at(2) >> charge__buffer >> residue__buffer;
        index__atom.push__back(atom__names__buffer);
        atomic__number.push__back(atomic__number__buffer);
        coordinates.push__back(coordinates__buffer);
        charge.push__back(charge__buffer);
        name__res.push__back(residue__buffer);
        }
        iFS >> title__connection;
        vector <string> read__line;
        int count = 0 ;
        while (getline(iFS,line))
        {
            if(count >= 1)
            {
            read__line = explode(line, '\t');
            for( int i = 0; i < read__line.size(); i++)
            {
                connection__buffer.push__back(stoi(read__line.at(i)));
            }
                    connection.push__back(connection__buffer);
            read_line.clear();
            connection__buffer.clear();
        }
        count++;
        }
    }
    else
    {
        cout << " Cannot open "<< input__system << endl;
    }
    iFS.clear ();
iFS.close();
}
void Topology::output__Gro__File()
{
ofstream oFS;
oFS.open(init__gro_file.c__str());
if (oFS)
{
oFS << atoms << endl;
oFS << box_x <<" "<< box_y <<" "<< box__z << endl;

```
```

        for (int i = 0; i < atoms; i++)
        {
            oFS << atomic_number.at(i) <<" ";
            oFS << fixed << setprecision(20) << coordinates.at(i)[0] <<"
            oFS << fixed << setprecision(20) << coordinates.at(i)[1] <<"
            oFS << fixed << setprecision(20) << coordinates.at(i)[2] <<"
            oFS << name__res.at(i) << endl;
        }
    }
    else
    {
        cout << " Cannot open "<< init__gro__file << endl;
    }
    oFS.close ();
oFS.clear();
}
void Topology::output__Topology__File()
{
ofstream oFS;
oFS.open(topol_system.c_str());
if(oFS)
{
//__W_Write te charge in the output _____///
oFS << "[charge]" << endl;
oFS << ";i q"<< endl;
for( int i = 0; i < atoms; i++ )
{
oFS << index_atom.at(i) <<" "<< charge.at(i) << endl;
}
//__ Write the bond in the output _-____l//
// Retrieve all the bonds from the connections.
vector<vector<int>> enumerate__bond;
vector<int> enumerate__bond__buffer (2,0);
for( int i = 0; i < connection.size(); i++ )
{
enumerate__bond__buffer.at(0) = connection.at(i) [0];
for (int j = 1; j < connection[i].size(); j++)
{
if( connection[i].size() > 1)
{
enumerate__bond__buffer.at(1) = connection.at(i)[j];

```
```

                enumerate__bond.push__back(enumerate__bond__buffer);
            }
        }
    }
// Erase the bonds which appear several times.
for( int i = 0; i < enumerate__bond.size(); i++ )
{
for( int j = i +1; j< enumerate__bond.size(); j++)
{
if( (enumerate__bond.at (i) [0] = enumerate__bond.at (j) [0] \&\&
enumerate__bond.at(i ) [1] = enumerate_bond.at(j)[1] ) ||
(enumerate__bond.at(i) [0] = enumerate__bond.at (j) [1] \&\&
enumerate__bond.at(i)[1] =enumerate__bond.at(j)[0] ) )
{
enumerate__bond.erase(enumerate__bond.begin()+j);
}
}
}
// We construct associate the index of atom to the atomic mass
// and construct the enumerate__bond filled with atomic
// mass instead of index
vector<vector <int > > associate__index__to__atomic_n
(enumerate__bond.size(), vector< <int> (2,0));
associate__index__to__atomic__n.at (0) [0] = atomic__number.at (0);
for( int i = 0; i < enumerate__bond.size(); i++ )
{
for ( int k = 0; k < index__atom.size(); k++ )
{
if( enumerate_bond.at(i)[0]= index__atom.at(k) )
{
associate__index__to__atomic__n.at(i ) [0] = atomic__number.at (k);
}
if( enumerate__bond.at(i)[1]= index__atom.at(k) )
{
associate__index__to__atomic__n.at(i) [1] = atomic__number.at (k);
}
}
}
// We retrieve the energy of each bond with the above vector
associate_index_to__atomic__n and write it in the output file.
oFS << "[bonded]" << endl;
oFS << "; i j K ro" << endl;
for( int i = 0; i < associate__index__to__atomic__n.size(); i++ )
{
for( int j = 0; j < bonded__index.size(); j++)
{
if( (associate_index_to__atomic__n.at(i) [0] = bonded__index.at (j) [0])

```

197 198
            \(\& \&(\) associate__index_to__atomic_n.at (i) \([1]=\) bonded__index.at (j)[1]) )
            \{
            oFS \(\ll\) enumerate__bond.at (i) [0] \(\ll " \quad " \ll\) enumerate__bond.at (i) [1]
            \(\ll\) " " \(\ll\) bonded__values.at (j) [0] \(\ll " \quad " \ll\) bonded__values.at (j) [1]
            \(\ll\) endl;
        \(\}\)
        \}
\(\}\)
//_ Write the angle in the output -_ // // // \(/\)
// Retrieve all the angles from the connections.
vector \(<\) vector \(<\) int \(\gg\) enumerate__angle;
vector \(<\) int \(>\) enumerate__angle__buffer (3,0);
for (int i \(=0 ; \quad \mathrm{i}<\) connection. size () ; i++ )
\{
enumerate_angle__buffer.at (1) = connection.at (i) [0];
    for (int \(\mathrm{j}=1 ; \mathrm{j}<\) connection \([\mathrm{i}] . \operatorname{size}()-1 ; \mathrm{j}++\) )
    \{
        if ( connection [i].size() > 1 )
        \{
            for ( int \(k=j+1 ; k<\operatorname{connection}[i] . \operatorname{size}() ; k++)\)
            \{
            enumerate__angle__buffer.at \((0)=\) connection.at \((i)[j]\);
            enumerate_angle_buffer.at (2) = connection.at (i) [k];
            enumerate__angle.push__back(enumerate__angle__buffer);
            \}
        \(\}\)
    \(\}\)
\}
    Erase the angles which appear several times
for ( int \(\mathrm{i}=0 ; \mathrm{i}<\) enumerate_angle.size (); i++ )
\(\{\)
    for ( int \(j=i+1 ; j<\) enumerate__angle.size () ; \(j++\) )
    \{
        if ( (enumerate_angle.at \((\mathrm{i})[0]=\) enumerate_angle.at (j) [0] \&\&
            enumerate_angle.at \((\mathrm{i})[1]=\) enumerate__angle.at \((\mathrm{j})[1]\) \&\&
            enumerate_angle.at (i) [2] =enumerate_angle.at (j)[2] ) ||
            (enumerate__angle.at (i) \([0]=\) enumerate__angle.at (j) [2] \&\&
            enumerate__angle.at (i) [1] =enumerate__angle.at (j) [1] \&\&
            enumerate__angle.at (i) [2] =enumerate__angle.at (j) [0] ) )
            \{
            enumerate__angle.erase(enumerate__angle.begin()+j);
            \}
    \}
\(\}\)
```

// We construct associate the index of atom to the atomic mass and
// construct the enumerate_angle filled with atomic
// mass instead of index.
associate_index_to__atomic_n.clear();
associate_index_to_atomic_n.resize(enumerate__angle.size(), vector<int> (3,0));
for( int i = 0; i < enumerate_angle.size(); i++ )
{
for ( int k = 0; k < index_atom.size(); k++ )
{
if( enumerate_angle.at(i)[0] = index_atom.at(k) )
{
associate_index_to__atomic_n.at(i)[0] = atomic_number.at(k);
}
if( enumerate_angle.at(i)[1] = index_atom.at(k) )
{
associate_index_to_atomic_n.at(i)[1] = atomic_number.at(k);
}
if( enumerate_angle.at(i)[2] = index_atom.at(k) )
{
associate_index_to__atomic_n.at(i)[2] = atomic_number.at(k);
}
}
}
// We retrieve the energy of each angle with the above vector
// associate_index__to_atomic_n and write it in the output file
oFS << "[angular]" << endl;
oFS << ";i j k K theta "<< endl;
for( int i = 0; i < associate_index_to_atomic_n.size(); i++ )
{
for( int j = 0; j < angular_index.size(); j++)
{
if( (associate_index_to_atomic_n.at(i)[0] = angular_index.at(j)[0])
\&\& (associate_index_to_atomic_n.at(i)[1] == angular_index.at(j)[1])
\&\& (associate_index_to_atomic_n.at(i)[2] == angular_index.at(j)[2]) )
{
oFS << enumerate_angle.at(i)[0] <<" "<< enumerate_angle.at(i)[1]
<<" "<< enumerate_angle.at(i)[2] <<" "<<
angular_values.at(j)[0] <<" "<< angular_values.at(j)[1] << endl;
}
}
}

```

```

// We retrieve the energy of each angle with the above vector
// associate_index_to__atomic_n and write it in the output file
oFS << " [non-bonded] " << endl;

```
```

        oFS << ";i epsilon sigma"<< endl;
        for( int i = 0; i < atomic_number.size(); i++ )
        {
            for( int j = 0; j < non__bonded_index.size(); j++)
            {
                if( (atomic_number.at(i) = non__bonded_index.at(j)) )
                    {
                oFS << index_atom.at(i) <<" "<< non__bonded_values.at(j)[0]
                <<" "<< non_bonded_values.at(j)[1] << endl;
                    }
        }
        }
    }
    else
{
cout << "Cannot open "<< topol_system << endl;
}
oFS.close();
oFS.clear();
void Topology::read_Bonded_File(string bonded_interaction)
ifstream iFS;
iFS.open(bonded_interaction.c_str());
string line;
vector<string> split_line;
vector <int> buffer__bonded_index (2,0);
vector <double> buffer_bonded__values (2,0.0);
vector <int> buffer_angular_index (3,0);
vector <double> buffer_angular_values(2,0.0);
if(iFS)
{
while(getline(iFS, line))
{
if(line[0] != '[' \&\& line[0] != ';')
{
split_line = explode(line, '\t');
if(split_line.size() = 4)
{
buffer_bonded_index.at(0) = stoi(split_line.at(0));
buffer_bonded_index.at(1) = stoi(split_line.at(1))
buffer_bonded__values.at(0) = stod(split_line.at(2));
buffer_bonded_values.at(1) = stod(split_line.at(3));
bonded_index.push_back(buffer__bonded__index);

```
\}
\{
```

                bonded__values.push_back(buffer__bonded__values);
                    }
                else
                {
                if(split_line.size() = 5)
                {
                    buffer_angular_index.at(0) = stoi(split_line.at(0));
                    buffer_angular_index.at(1) = stoi(split_line.at(1));
                    buffer_angular_index.at(2) = stoi(split_line.at(2));
                    buffer_angular_values.at(0) = stod(split_line.at(3));
                    buffer_angular_values.at(1) = stod(split_line.at(4));
                    angular_index.push_back(buffer_angular_index);
                    angular_values.push_back(buffer_angular_values);
                }
            }
            split_line.clear();
        }
        }
    }
    else
    {
        cout << "Cannot open " << bonded_interaction << endl;
    }
    iFS.close();
    iFS.clear();
    }
void Topology::read_nonBonded_File(string nonbonded_interaction)
{
ifstream iFS;
iFS.open(nonbonded_interaction.c_str());
string line;
vector<string> split_line;
int buffer_non__bonded_index = 0;
vector <double> buffer_non_bonded_values (2,0.0);
if (iFS)
{
while(getline(iFS,line))
{
if(line[0] != '['\&\& line[0] != ';')
{
split_line = explode(line, '\t');
if(split_line.size() = 3)
{
buffer_non__bonded_index = stoi(split_line.at(0));
buffer_non_bonded_values.at(0) = stod(split_line.at (1));
buffer_non_bonded_values.at(1) = stod(split_line.at(2));
non_bonded_index.push_back(buffer__non_bonded_index);

```
```

                    }
            }
        }
    }
    else
    {
                cout << "Cannot open" << nonbonded_interaction << endl;
    }
    iFS.close();
    iFS.clear();
    }
vector<string> Topology::explode(const string\& s, const char\& c)
{
string buff{""};
vector<string> v;
for(auto n:s)
{
if(n != c) buff+=n; else
if(n=c \&\& buff != "") { v.push_back(buff); buff = " "; }
}
if(buff != " ") v.push_back(buff);
return v;
}

```

\section*{A.2.2 Builiding_topology.h}

Listing A. 5 - Builiding_topology.h
```

    frame
    /*

* Initialisation of the topology - declaration
* 

*/
\#ifndef TOPOLOGY_H
\#define TOPOLOGY_H
\#include <iostream>
\#include <string>
\#include <vector>
\#include <fstream>
\#include <iomanip>
\#include <sstream>
using namespace std;
class Topology{
private:

```
```

//___ Information File _-__-_-_-_-_-_
/* Input: File containing the initial information of the system file */
string input_system;
/* Output: File containing the initial trajectory of the system */
string init_gro_file;
/* Output: File describing the Topology of the System */
string topol_system;
//-_ Information system --_-_-------
/* Number of Atoms in the system */
int atoms;
/* size x box */
double box_x;
/* size y box */
double box_y;
/* size z box */
double box_z;
/* Vector containing coordinate of the N_chain */
vector <vector<double> > coordinates;
/* Vector containing the index of each atom */
vector <int> index_atom;
/* Vector containing the name of each atom */
vector <int> atomic_number;
/* Vector containing the name of residue */
vector <string> name_res;
/* Vector containing the name of residue */
vector <double> charge;
/* Vector containing connectivity of each atom */
vector <vector<int>> connection;
//-_ Information interaction ---------
/* Vector containing the bonded interaction (index) */
vector<vector<int>> bonded_index;
/* Vector containing the value of bonded interaction (ro and K) */
vector<vector<double> > bonded_values;
/* Vector containing the angular interaction (index) */
vector<vector<int>> angular_index;

```
```

    /* Vector containing the value of angular interaction (ro and K') */
    vector <vector < double > > angular__values;
    /* Vector containing the index for non bonded-interaction */
    vector<int> non__bonded__index;
    /* Vector containing the value for non-bonded interaction */
    vector <vector <double > > non__bonded__values;
    public:
    Topology(string input__system, string init__gro__file, string topol__system);
    ~Topology ();
    void read__Input__System();
    void output__Gro__File();
    void output__Topology__File();
    void read__Bonded__File(string bonded__interaction);
    void read__nonBonded__File(string nonbonded__interaction);
    vector<string> explode(const string& s, const char& c);
    };
\#endif

```

\section*{A.2.3 Makefile}

\section*{Listing A. 6 - Makefile}
frame
build_topology: main.o building_topology.o
g++ -o exe main.o building_topology.o -O3 -L/home/ambroise/gsl/lib
    -lgsl -lgslcblas -lm -static-libstdc++
main.o: main.cpp building_topology.h
\(\mathrm{g}++\)-o main.o -c main.cpp \(-\mathrm{std}=\mathrm{c}++11\)
building_topology.o: building_topology.cpp building_topology.h
g++ -o building_topology.o -c building_topology.cpp -std=c++11

\section*{A. 3 Program to perform a Monte Carlo simulation}

The main Monte carlo program is composed of 'Main.cpp', 'Move_MC.cpp', 'Potential_energy calculation.cpp' and 'Cell_lists.cpp'.

For each file excepted 'Main.cpp' and 'MersenneTwister.h', there is a corresponding header file denoted by 'Move_MC.h', 'Potential_energy _calculation.h' and 'Cell_lists.h'.

\section*{A.3.1 Main.cpp}

Listing A. 7 - Main.cpp
frame
1
```


#### Input file to the main file: 'conf.in'

40000000
../ building_topology_file/system_gro_1.txt
../ building_topology_file/system_topol.txt
traj_output_1.txt
energy_1.txt
\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#
\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#\#
\#include <iostream>
\#include <cmath>
\#include <vector>
\#include <fstream>
\#include <gsl/gsl_rng.h>
\#include <chrono>
\#include <ctime>
\#include <iomanip>
\#include "system_MC.h"
\#include "MersenneTwister.h"
using namespace std;
int main (int argc, char * argv[]){
if(argc != 2)
{
cout << "Usage: " << endl;
cout << argv[0] <<" configfile." << endl;
return -1;
}
// unsigned long int graine = stoi(argv[2]);
// Declaration of configfile \& output flux.
ifstream iFS;
string configfile;
ofstream oFS;
configfile = argv[1];
// Arguments of the configfile
int total_MC_step;
string input_gro;
string input_topol;
string traj_output;
string energy_output;
// Read the configfile
cout << endl;
cout << "***********************************" << endl;

```
```

cout <<"Reading " << configfile << "... **"<<endl;
iFS.open(configfile.c__str());
iFS >> total_MC__step;
iFS >> input__gro;
iFS >> input__topol;
iFS >> traj__output;
iFS >> energy__output;
cout <<"... Ok. **" << endl;
cout << "*********************************" << endl;
cout << endl;
cout << endl;
// Variable use for the generator number.
int pick__atom;
int pick_nano;
double random__metropolis;
// Initialise random number generator.
MersenneTwister rng;
// Reinitialize the output for trajectory file
oFS.open(traj__output.c__str());
oFS.close();
oFS.clear();
// Reinitialize the output for energy file.
oFS.open(energy__output.c__str());
oFS.close ();
oFS.clear();
// Define the differents atoms
int number_dna = 4;
int atom__per_dna = 150;
int number_dna__atom = atom__per__dna*number_dna;
int number__gold__nano = 40;
int free__ions = 0;
int atom_per_Nano = 25;
int total__N = free__ions + number_gold__nano - 1;
// Initialize time when running program.
clock__t tic = clock();
// Initialization of the system object.
System__MC system(total_MC__step, input__gro, input__topol, traj__output,
energy__output, number__dna__atom, atom__per_dna, number__dna,
number_gold_nano, atom__per__Nano, free__ions);
// Do the MC simulation
for ( int step = 0; step < total_MC_step; step++ )

```
```

    {
        pick_atom = 1+int(round ((total__N)*rng()));
        if( pick_atom > free__ions)
        {
            pick__nano = free__ions + atom__per_Nano
            *(pick__atom - free__ions) - (atom__per_Nano - 1);
                if(\operatorname{rng}()<=0.25)
                {
                system.rotate__MC((number_dna__atom-1) + pick__nano);
            }
            else
            {
                system.translate__MC((number_dna__atom-1) + pick__nano );
            }
            }
            else
            {
            system.translate__MC((number_dna_atom-1) + pick__atom);
        }
    }
    // Display the accepted steps.
    system.Display__accepted__step ();
    // Retrieve the total time of the program.
    clock_t toc = clock();
    cout << "Running time: %f seconds: " <<
        (double)(toc - tic) / CLOCKS_PER_SEC << endl;
    return 0;
    ```
\}

\section*{A.3.2 Move_MC.cpp}

Listing A. 8 - Move_MC.cpp
frame
```

\#include <iostream>
\#include <vector>
\#include <cmath>
\#include < gsl/gsl__rng.h>
\#include <chrono>
\#include <ctime>
\#include <iomanip>
\#include < fstream>
\#include <sstream>
\#include "system__MC.h"
\#include "List.h"
using namespace std;

```
```

System_MC::System_MC()
{}
System_MC::~System_MC()
{}
System_MC::System_MC(int total_MC_step, string input_gro, string
input_topol, string traj_output, string energy_output, int atom_DNA,
int atom_per_dna, int number_dna, int gold_nano, int
atom_per_gold__nano, int free_ions)
: energy__system(input_topol, input_gro, atom_DNA, atom_per_dna,
number_dna, gold_nano, atom_per_gold_nano, free_ions),
list (atom_DNA, gold_nano, free__ions, atom_per_gold_nano)
{
// We must initialize the counter to zero at each step.
this }->\mathrm{ compteur = 0;
this }->\mathrm{ compteur_rotate = 0;
this }->\mathrm{ compteur_translate_ion = 0;
this }->\mathrm{ compteur_translate_AuGP = 0;
this ->total_rotate = 0;
this }->\mathrm{ total_translate_ion = 0;
this }->\mathrm{ total_translate_AuGP = 0;
// Initialization of the species in the system.
this }->\mathrm{ atom_DNA = atom_DNA;
this ->atom_per_dna = atom_per_dna;
this }->\mathrm{ number_dna = number_dna;
this }->\mathrm{ gold_nano = gold_nano;
this }->\mathrm{ free_ions = free_ions;
this ->atom_per_gold_nano = atom_per_gold_nano;
// Initialization of files
this }->\mathrm{ total_MC_step = total_MC_step;
this }->\mathrm{ -input_gro = input_gro;
this ->input_topol = input_topol;
this ->traj_output = traj_output;
this->energy_output = energy_output;
// Read .gro file
readInputFile();
//We fill the trajectory with the initial coordinates.
SetBoxOutput(traj_output);
fillTrajectory(traj_output, coordinates);
// Initialization of the trajectory of the system
energy_system.InitTrajFirstStep(coordinates);
// Initialization of the neighbour list of AuGP.
list.Set_box(box_x, box_y, box_z);
list.Initialization_cell_list(coordinates);

```
```

    // We need also the box cell size of the list
    this }->>\mathrm{ box__cell__size = list.get__box__cell_list ();
    // Initialization of the energy of the system
    energy = energy__system.getEnergy ();
    // Initialization of the PBC for cell list in the energy class
    energy__system.Set__PBC__cell__list(list.Get__PBC_list ());
    energy__system.Set__PBC__vector__shift(list.get__box__shift__PBC());
    // Choose amplitude of move
    this }->\mathrm{ >amplitude__move = 1.0; // in Angstrom
    }
void System_MC::translate__MC(int pick__atom)
{
// We retrieve the step i with compteur.
compteur++;
// Vector to store trajectories of moved entity.
vector <vector <double > > feed__traj;
// Vector to store the neighbor of the AuGP
vector<int> Chosen__index__AuGP__list;
// Random numbers for displacement.
double random_x = rng();
double random_y = rng();
double random_z = rng();
vector<vector<int>> index_cell_cchosen__particles__after__move;
if( pick_atom <= ((atom_DNA-1) + free__ions) )
{
// We count the attemp to translate
total__translate__ion++;
Chosen__index__AuGP_list.push__back(pick__atom);
coordinates_try_MC.at(pick__atom)[0] = coordinates.at(pick__atom)[0]
+ (this ->amplitude__move * (2*random__x - 1.0));
coordinates_try__MC.at(pick__atom)[1] = coordinates.at(pick__atom )[1]
+ (this ->amplitude__move * (2*random__y - 1.0));
coordinates_try__MC.at(pick__atom)[2] = coordinates.at(pick__atom )[2]
+ (this ->amplitude__move * (2*random__z - 1.0));
// We fill all the other position with the former coordinates.
for( int i = 0; i < coordinates.size(); i++ )
{
if(i != pick__atom)
{
coordinates__try__MC.at(i)[0] = coordinates.at(i)[0];
coordinates_try__MC.at(i)[1] = coordinates.at(i)[1];

```
```

            coordinates_try__MC.at(i)[2] = coordinates.at(i)[2];
        }
    }
// Make PBC of the atom__trajectory
Check_PBC__Box ();
// Trajectory to give to energy class.
feed__traj.resize(1, vector <double> (3,0));
// Complete the feed_traj vector before sending it to energy class
feed__traj.at (0)[0] = coordinates__try__MC__pbc__version.at(pick_atom ) [0];
feed__traj.at(0)[1] = coordinates__try__MC__pbc__version.at(pick__atom )[1];
feed__traj.at (0)[2] = coordinates__try__MC__pbc__version.at(pick__atom )[2];
// Vector to store the index of cell of the chosen particles,
// before and after move.
index__cell_chosen__particles__after__move.resize(1, vector<int> (3,0));
// Complete the index Chosen cell vector
// retrieve the index cell after the move.
index__cell__chosen__particles__after__move.at (0)[0] = 1 +
floor(feed__traj.at (0)[0]/box__cell__size[0]);
index__cell__chosen__particles__after__move.at (0)[1] = 1 +
floor(feed__traj.at (0)[1]/box_cell_size[1]);
index__cell__chosen__particles__after__move.at(0)[2] = 1 +
floor(feed__traj.at (0)[2]/box_cell_size[2]);
// Test of the energy of the polymer,
// if the new energy is lower, we reject it.
double probability_transition;
double energy__after;
bool Is__accepted;
// We test first the interpenetration between molecules and cylinder.
energy__system.ChangeTraj(feed__traj , Chosen__index__AuGP__list,
list.Get__cell__list__without__chosen__atoms(Chosen__index__AuGP__list),
index__cell__chosen__particles__after__move);
if(energy__system.Test__Exclusion__volume() == false)
{
energy__system.Call_Calculate__Energy();
energy__after = energy__system.getEnergy();
probability_transition = exp(-(energy__after - (this ->energy))/KT);
// Acceptance or rejection of the montecarlo step.
Is_accepted = Metropolis__scheme(probability__transition,
energy__after, rng());
energy__system.AcceptMoveUpdate(Is__accepted, this ->energy);
if(Is__accepted = true)

```
```

            {
                compteur__translate__ion++;
                // Update the cell list.
                list.Update__cell__list(feed__traj, Chosen__index__AuGP__list);
            }
    }
    else
    {
        // We do not touch the trajectory.
    }
    // Copy the trajectory in output file
if(compteur % 4000=0) fillTrajectory(traj__output, coordinates);
// Copy the energy in a file.
doEnergy(this ->energy);
}
else
{
// Count all the move attempts
total_translate__AuGP++;
vector<int> particle__AuGP__list;
int mi = 0;
int mj = 0;
// We need to update the list for the former step
particle__AuGP__list = list.Update_lists__AuGP__ion(coordinates,
box_x, box_y, box_z, pick__atom);
// We need to store the index of the actual AuGP
// before its surrounding ions.
for(int i = 0; i < atom__per__gold__nano; i++)
{
Chosen_index_AuGP_list.push__back(pick__atom + i );
}
// Store the index of the actual surrounding ions.
for(int i = 0; i < particle__AuGP__list.size(); i++)
{
Chosen_index__AuGP__list.push__back(particle__AuGP__list.at (i));
}
mi = particle_AuGP__list.size ();
// Keep the DNA coordinates constant.
for(int i = 0; i <= (atom_DNA-1); i++ )
{
coordinates_try_MC.at(i)[0]= coordinates.at(i)[0];
coordinates_try__MC.at(i)[1]= coordinates.at(i)[1];
coordinates_try_MC.at(i)[2]= coordinates.at(i)[2];
}

```
// Modify the ions coordinates to Choose the ions to be moved with the AuGP.
for (int \(\mathrm{j}=\) atom_per_gold_nano; \(\mathrm{j}<\) Chosen_index_AuGP_list.size (); j++)
\{
    coordinates_try_MC.at (Chosen_index_AuGP_list.at(j))[0] =
    coordinates.at (Chosen_index_AuGP_list.at(j))[0] +
    (this \(->\) amplitude_move \(*(2 *\) random_x -1.0\()\) );
    coordinates_try_MC.at (Chosen_index_AuGP_list.at(j))[1] =
    coordinates.at(Chosen_index_AuGP_list.at(j))[1] +
    (this \(\rightarrow\) amplitude_move * \((2 *\) random_y -1.0\())\);
    coordinates_try_MC.at (Chosen_index_AuGP_list.at(j))[2] =
    coordinates.at(Chosen_index_AuGP_list.at(j))[2] +
    (this \(\rightarrow\) amplitude_move \(*(2 *\) random_z -1.0\()\) );
\}
bool boolean \(=\) false;
// Keep the other ions constant.
for (int \(\mathrm{i}=\) atom_DNA; \(\mathrm{i}<=\) atom_DNA + free_ions \(-1 ; \mathrm{i}++\) )
\{
    boolean \(=\) false;
    for (int \(\mathrm{j}=\) atom_per_gold_nano; \(\mathrm{j}<\) Chosen_index_AuGP_list.size () ; j++)
    \{
        if \((\mathrm{i}=\) Chosen_index_AuGP_list.at \((\mathrm{j})\) )
        \{
            boolean = true;
            break;
        \}
    \}
    if (boolean \(=\) false \()\)
    \{
        coordinates_try_MC.at(i) \([0]=\) coordinates.at(i) \([0]\);
        coordinates_try_MC.at(i) \([1]=\) coordinates.at(i) \([1]\);
        coordinates_try_MC.at(i)[2]= coordinates.at(i)[2];
    \}
\}
// Keep the gold nanoparticule before the chosen one
for (int \(\mathrm{i}=\) atom_DNA + free_ions ; \(\mathrm{i}<=\) pick_atom -1 ; \(\mathrm{i}++\) )
\{
    coordinates_try_MC.at(i) \([0]=\) coordinates.at(i) \([0]\);
    coordinates_try_MC.at(i) \([1]=\) coordinates.at(i) \([1]\);
    coordinates_try_MC.at(i)[2] = coordinates.at(i) [2];
\}
// Displacement of the gold nanoparticle
for (int i = pick_atom; i < pick_atom + atom_per_gold_nano; i++ )
\{
    coordinates_try_MC.at(i)[0] \(=\) coordinates.at(i)[0] +
    (this \(->\) amplitude_move * ( \(2 *\) random_x -1.0 ) ;
    coordinates_try_MC.at(i)[1] = coordinates.at(i)[1] +
```

    (this ->amplitude__move * (2*random_y - 1.0));
    coordinates_try_MC.at(i)[2] = coordinates.at(i)[2] +
    (this ->amplitude__move * (2*random_z - 1.0));
    }
// Keep the other coordinates constant.
for(int i = pick__atom + atom__per_gold_nano; i < coordinates.size() ; i++ )
{
coordinates_try_MC.at(i)[0]= coordinates.at(i)[0];
coordinates_try_MC.at(i)[1]= coordinates.at(i)[1];
coordinates_try_MC.at(i)[2]= coordinates.at(i)[2];
}
// Make PBC of the atom__trajectory
Check_PBC__Box();
// Trajectory to give to energy class.
feed__traj.resize(atom__per__gold__nano + mi, vector<double> (3,0));
// Copy in the buffer the coordinates of gold nanoparticle.
// (size = atom__per_gold__nano)
for( int atom = 0; atom < atom__per_gold__nano; atom++ )
{
feed_traj.at(atom)[0] =
coordinates_try__MC__pbc__version.at(Chosen__index__AuGP_list.at (atom )) [0];
feed__traj.at(atom)[1]=
coordinates_try__MC__pbc__version.at (Chosen__index_AuGGP_list.at (atom ) ) [1];
feed__traj.at(atom)[2]=
coordinates_try__MC__pbc__version.at(Chosen__index__AuGP__list.at(atom )) [2];
}
//Copy in the buffer the coordinates of the cloud of selected ions.
//(size = mi)
for( int atom = atom__per__gold__nano; atom < feed__traj.size(); atom++ )
{
feed_traj.at(atom)[0] =
coordinates_try__MC_pbc__version.at(Chosen__index__AuGP_llist.at(atom )) [0];
feed_traj.at(atom)[1] =
coordinates_try__MC_pbc__version.at(Chosen__index__AuGP_llist.at(atom )) [1];
feed_traj.at(atom)[2] =
coordinates_try__MC__pbc__version.at(Chosen__index_AuGP_list.at(atom )) [2];
}
// Vector to store the index of cell of the chosen particles, after move.
index__cell__chosen__particles__after__move.resize(atom__per__gold__nano + mi
, vector<int> (3,0));
// Complete the index Chosen cell vector
for( int atom = 0; atom < Chosen__index__AuGP__list.size(); atom++)

```
\{
index_cell_chosen_particles_after_move.at(atom) \([0]=1+\) floor (feed_traj.at (atom) [0]/box_cell_size[0]); index_cell_chosen_particles_after_move.at(atom)[1] = \(1+\) floor (feed_traj.at (atom) [1]/box_cell_size[1]);
index_cell_chosen_particles_after_move.at(atom) \([2]=1+\) floor (feed_traj.at(atom)[2]/box_cell_size[2]);
    \}
    // Test of the energy of the polymer,
    // if the new energy is lower, we reject it.
    double probability_transition;
    double energy_after;
    bool Is_accepted;
    // We test first the interpenetration between molecules and cylinder.
    energy_system. ChangeTraj (feed_traj, Chosen_index_AuGP_list,
    list. Get_cell_list_without_chosen_atoms (Chosen_index_AuGP_list),
    index_cell_chosen_particles_after_move);
    if (energy_system.Test_Exclusion_volume () =e false)
    \{
        energy_system. Call_Calculate_Energy ();
        energy_after \(=\) energy_system.getEnergy ();
        \(\mathrm{mj}=\) list. Count_ion_around_AuGP (coordinates_try_MC_pbc_version,
        pick_atom, box_x, box_y, box_z);
        probability_transition \(=\) pow \((0, \mathrm{mj}-\mathrm{mi}) *\)
        \(\exp (-(\) energy_after \(-(\) this \(\rightarrow\) energy \()) / \mathrm{KT})\);
        // Acceptance or rejection of the montecarlo step.
        Is_accepted = Metropolis_scheme(probability_transition,
        energy_after, rng());
        energy_system. AcceptMoveUpdate(Is_accepted, this \(\rightarrow\) energy );
        if (Is_accepted = true)
        \{
            compteur_translate_AuGP++;
                // Update the cell list.
        list. Update_cell_list(feed_traj, Chosen_index_AuGP_list);
        \}
    \}
        else
        \{
        // We do not touch the trajectory.
    \}
    // Copy the trajectory in output file
    if (compteur \(\% 4000=0)\) fillTrajectory (traj_output, coordinates);
    // Copy the energy in a file.
    doEnergy (this \(\rightarrow\) energy) ;
    \}
\}

394
395 396 397 398
void System_MC::rotate_MC(int pick_gold__nano)
\{
// We retrieve the step i with compteur.
compteur++;
// We count the attemp to translate.
total_rotate++;
// Vector to store the neighbor of the AuGP.
vector<int> particle_AuGP_list;
vector<int> Chosen_index_AuGP_list;
int mi \(=0\);
int \(\mathrm{mj}=0\);
// For the pivot algorithm.
double RX;
double RY;
double RZ;
double random_alpha \(=\) rng ();
double random_theta \(=\operatorname{rng}()\);
double random_phi \(=\) rng ();
```

double cos_alpha = cos(2 * PI *(2*random__alpha - 1) );
double sin_alpha = cos(PI/2 - (2 * PI *(2*random_alpha - 1)));
double cos_theta = 2 * random_theta - 1;
double sin_theta = cos(PI/2 - acos(cos_theta));
double cos_phi = cos(2 * PI * random_phi);
double sin_phi = cos(PI/2 - (2 * PI * random_phi));
// We need to update the list for the former step.
particle_AuGP_list = list.Update_lists_AuGP_ion(coordinates,
box_x, box_y, box_z, pick_gold_nano);
// We need to store the index of the actual AuGP
// before its surrounding ions.
for(int i = 0; i < atom__per_gold_nano; i++)
{
Chosen_index_AuGP_list.push_back(pick_gold_nano + i );
}
// We choose all the atoms in the vicinity of the AuGP
for(int i = 0; i < particle_AuGP_list.size(); i++)
{
Chosen_index_AuGP_list.push_back(particle_AuGP_list.at(i));
}
mi = particle_AuGP_list.size();

```
```

// Keep the DNA coordinates constant
for(int i = 0; i <= (atom_DNA-1); i++ )
{
coordinates_try__MC.at(i)[0] = coordinates.at(i)[0];
coordinates_try_MC.at(i)[1]= coordinates.at(i)[1];
coordinates_try__MC.at(i)[2]= coordinates.at(i)[2];
}
// Modify the ions coordinates to Choose the ions to be moved with the AuGP
for(int j = atom__per__gold__nano ; j < Chosen__index__AuGP_llist.size (); j++)
{
// We have to be careful if there is the PBC condition when we turn
// the gold nanoparticle
double RX = coordinates.at(Chosen__index__AuGP__list.at(j))[0]
- coordinates.at(pick_gold__nano)[0];
double RY = coordinates.at(Chosen__index__AuGP__list.at(j))[1]
- coordinates.at(pick__gold__nano)[1];
double RZ = coordinates.at(Chosen__index__AuGP__list.at(j))[2]
- coordinates.at(pick__gold__nano)[2];
RX = RX - box_x * round (RX / box_x );
RY = RY - box_y * round (RY / box_y );
RZ = RZ - box z * round(RZ / box z);
coordinates_try__MC.at(Chosen__index__AuGP__list.at(j))[0] =
coordinates.at(pick_gold__nano)[0] + (RX*cos__alpha) +
((1-cos_alpha) *(sin__theta * cos__phi*RX + sin__theta*sin__phi*RY +
(cos_theta)}*\textrm{RZ})*(\operatorname{sin__theta}*\operatorname{cos__phi}))
(sin__alpha*(( sin__theta *sin__phi*RZ) - (( cos__theta) *RY)));
coordinates_try__MC.at(Chosen__index__AuGP__list.at(j))[1] =
coordinates.at(pick_gold__nano)[1] + (RY*cos__alpha) +
((1-cos_alpha)})*(\operatorname{sin_theta}*\operatorname{cos__phi}*RX + sin__theta*sin__phi*RY +
(cos__theta)}*\textrm{RZ})*(\operatorname{sin__theta}*\operatorname{sin__phi}))
(sin__alpha*(( cos__theta *RX) - (sin__theta* cos__phi*RZ)));
coordinates_try__MC.at(Chosen__index__AuGP_list.at(j ) [ 2] =
coordinates.at(pick_gold__nano)[2] + (RZ*cos__alpha) +
((1-cos_alpha)}*(\operatorname{sin_theta}*\operatorname{cos__phi}*RX + sin__theta*sin__phi*RY +
(cos_theta)*RZ)*(cos__theta))}
(sin__alpha*((sin__theta*cos__phi*RY) - (sin__theta*sin__phi*RX)));
}
bool boolean = false;
// Keep the other ions constant
for(int i = atom_DNA; i <= atom_DNA + free__ions - 1; i++ )
{
boolean = false;
for(int j = atom__per__gold__nano;
j < Chosen__index__AuGP__list.size(); j++)
{
if(i = Chosen__index__AuGP__list.at (j))

```
```

        {
                boolean = true;
                break;
        }
    }
    if(boolean == false)
    {
        coordinates_try_MC.at(i)[0]= coordinates.at(i)[0];
        coordinates_try_MC.at(i)[1]= coordinates.at(i)[1];
        coordinates_try_MC.at(i)[2] = coordinates.at(i)[2];
        }
    }
// Keep the gold nanoparticule before the chosen one.
for(int i = (atom_DNA) + free__ions; i <= pick_gold_nano; i++ )
{
coordinates_try_MC.at(i)[0]= coordinates.at(i)[0];
coordinates_try_MC.at(i)[1] = coordinates.at(i)[1];
coordinates_try_MC.at(i)[2]= coordinates.at(i)[2];
}
// Rotation of the gold nanoparticle
for (int i = pick__gold_nano + 1; i < pick_gold__nano + atom__per_gold_nano; i++ )
{
// We have to be careful if there is the PBC condition
when we turn the gold nanoparticle.
double RX = coordinates.at(i)[0] - coordinates.at(pick_gold_nano)[0];
double RY = coordinates.at(i)[1] - coordinates.at(pick_gold_nano)[1];
double RZ = coordinates.at(i)[2] - coordinates.at(pick_gold_nano)[2];
RX = RX - box_x * round(RX / box_x);
RY = RY - box_y * round(RY / box_y);
RZ = RZ - box_z * round (RZ / box_z);
coordinates_try_MC.at(i)[0] =
coordinates.at(pick_gold_nano)[0] + (RX*cos_alpha) +
((1-cos_alpha)*(sin_theta*cos__phi *RX + sin_theta*sin_phi*RY +
(cos_theta)*RZ)*(sin_theta*cos_phi)) +
(sin_alpha*((sin_theta*sin_phi*RZ) - (( cos_theta) *RY)));
coordinates_try_MC.at(i)[1] =
coordinates.at(pick_gold_nano)[1] + (RY*cos_alpha) +
((1-cos_alpha)*(sin_theta*cos__phi*RX + sin_theta*sin_phi*RY +
(cos_theta)*RZ)*(sin_theta*sin_phi)) - +
(sin_alpha*((cos_theta *RX) - (sin_theta*\operatorname{cos_phi*RZ)));}
coordinates_try_MC.at(i)[2]=
coordinates.at(pick_gold_nano)[2] + (RZ*cos_alpha) +
((1-cos_alpha)*(sin_theta*cos__phi*RX + sin_theta*sin_phi*RY +
(cos_theta)*RZ)*(cos_theta)) +
(sin_alpha *((sin_theta*\operatorname{cos_phi*RY) - (sin_theta *sin__phi*RX)));}
}

```
```

// Keep the other coordinates constant
for( int i = pick_gold_nano + atom_per_gold_nano; i < coordinates.size(); i++)
{
coordinates_try_MC.at(i)[0] = coordinates.at(i)[0];
coordinates_try_MC.at(i)[1]= coordinates.at(i)[1];
coordinates_try_MC.at(i)[2]= coordinates.at(i)[2];
}
// Make PBC of the atom_trajectory
Check_PBC_Box();
// Vector to store trajectories of moved entity.
vector<vector<double> > feed_traj(atom__per_gold__nano + mi,
vector<double> (3,0));
// Copy in the buffer the AuGP;
for( int atom = 0; atom < atom_per_gold_nano; atom++ )
{
feed_traj.at(atom)[0] =
coordinates_try_MC_pbc_version.at(Chosen_index_AuGP_list.at(atom))[0];
feed_traj.at(atom)[1] =
coordinates_try_MC_pbc_version.at(Chosen_index_AuGP_list.at(atom ) ) [1];
feed_traj.at(atom)[2] =
coordinates_try_MC_pbc_version.at(Chosen_index_AuGP_list.at(atom ) ) [2];
}
//Copy in the buffer the coordinates of the cloud of selected ions.
for( int atom = atom_per_gold_nano; atom < feed_traj.size(); atom++ )
{
feed_traj.at(atom)[0] =
coordinates_try_MC_pbc_version.at(Chosen_index_AuGP_list.at(atom))[0];
feed_traj.at(atom)[1] =
coordinates_try_MC_pbc_version.at(Chosen_index_AuGP_list.at(atom))[1];
feed_traj.at(atom)[2] =
coordinates_try_MC_pbc_version.at(Chosen_index_AuGP_list.at(atom))[2];
}
// Vector to store the index of cell of the chosen particles, before move
vector<vector<int> > index_cell_chosen_particles__after_move
(atom_per_gold_nano + mi, vector<int> (3,0));
// Complete the index Chosen cell vector
for( int atom = 0; atom < Chosen_index_AuGP_list.size(); atom++)
{
index_cell_chosen_particles_after_move.at(atom)[0] = 1 +
floor(feed_traj.at(atom)[0]/ box_cell__size[0]);
index_cell_chosen_particles_after_move.at(atom)[1] = 1 +
floor(feed_traj.at(atom)[1]/ box_cell_size[1]);

```

610
\{
```

        index_cell_chosen_particles_after_move.at(atom)[2] = 1 +
        floor(feed_traj.at(atom)[2]/ box_cell_size[2]);
    }
    ```
    // Test of the energy of the polymer, if the new energy is lower,
    // we reject it.
    double probability_transition;
    double energy_after;
    bool Is_accepted;
    // We test first the interpenetration between molecules and cylinder
    energy_system. ChangeTraj (feed_traj, Chosen_index_AuGP_list,
    list. Get_cell_list_without_chosen_atoms (Chosen_index_AuGP_list),
    index_cell_chosen_particles_after_move);
    if (energy_system.Test_Exclusion_volume () = false)
    \{
        energy_system. Call_Calculate_Energy ();
        energy_after = energy_system.getEnergy ();
        // Calculation of the probability ratio.
        \(\mathrm{mj}=\) list. Count_ion_around_AuGP (coordinates_try_MC_pbc_version,
        pick_gold_nano, box_x, box_y, box_z);
        probability_transition \(=\operatorname{pow}(0, \mathrm{mj}-\mathrm{mi}) *\)
        \(\exp (-(\) energy_after \(-(\) this \(->e n e r g y)) / K T) ;\)
        // Acceptance or rejection of the montecarlo step.
        Is_accepted = Metropolis_scheme(probability_transition,
        energy_after, rng());
        energy_system.AcceptMoveUpdate(Is_accepted, this \(\rightarrow\) energy);
        if (Is_accepted = true)
        \{
            compteur_rotate++;
        // Update the cell list.
        list. Update_cell_list(feed_traj, Chosen_index_AuGP_list);
        \}
    \}
        else
        \{
        // Copy the trajectory in output file
        if (compteur \(\% 4000=0)\) fillTrajectory (traj_output, coordinates);
        // Copy the energy in the file.
        doEnergy (this \(\rightarrow\) energy) ;
    \}
bool System_MC:: Metropolis_scheme(double probability_transition,
double energy_after, double random_metropolis)

664
665
```

    if(probability_transition >= 1.0)
    {
        this ->energy = energy_after;
        for (int i = 0; i < coordinates.size(); i++ )
        {
            coordinates.at(i) [0] = coordinates_try_MC_pbc_version.at(i)[0];
            coordinates.at(i)[1] = coordinates_try_MC_pbc_version.at(i)[1];
            coordinates.at(i)[2] = coordinates_try_MC_pbc_version.at(i)[2];
        }
        return true;
    }
    else
    {
        if( random_metropolis <= probability_transition)
    {
        this->energy = energy_after;
        for (int i = 0; i < coordinates.size(); i++ )
        {
                coordinates.at(i)[0] = coordinates_try_MC_pbc_version.at(i)[0];
                coordinates.at(i)[1] = coordinates_try_MC__pbc_version.at(i)[1];
                coordinates.at(i)[2] = coordinates_try_MC_pbc_version.at(i)[2];
        }
        return true;
    }
    else
    {
        // We do not touch trajectories.
        return false;
    }
    }
    void System_MC::readInputFile()
ifstream iFS;
iFS.open(input_gro.c_sstr());
int atoms;
int buffer_atomic_number;
vector <double> coordinates__buffer(3,0.0);
string name_residue_buffer = " ";
if(iFS)
{
iFS >> atoms;
iFS >> this ->box_x >> this ->box_y >> this ->box_z;

```
\}
\{
```

        for ( int i = 0; i < atoms; i++ )
        {
            iFS >> buffer__atomic_number >> coordinates_buffer.at(0) >>
            coordinates_buffer.at(1) >> coordinates_buffer.at(2) >>
            name_residue_buffer;
            this ->atomic_number.push_back(buffer__atomic_number);
            this }->\mathrm{ coordinates.push_back(coordinates__buffer);
    ```

```

        }
    }
    else
    {
        cout << "Cannot open file." << endl;
    }
    iFS.clear();
    iFS.close();
        coordinates_try_MC.resize(coordinates.size(), vector<double> (3,0));
        coordinates_try_MC_pbc_version.resize(coordinates.size(),
        vector<double> (3,0));
    void System_MC::Check_PBC_Box()
for( int atom_count = 0; atom_count < coordinates.size(); atom_count++ )
{
// Check the x direction
if ( coordinates_try_MC.at(atom_count)[0] > box_x )
{
coordinates_try_MC_pbc_version.at(atom_count)[0]=
coordinates_try_MC.at(atom_count)[0] - box_x;
}
else
{
if ( coordinates_try_MC.at(atom_count)[0] < 0 )
{
coordinates_try_MC_pbc_version.at(atom_count)[0] =
coordinates_try_MC.at(atom_count)[0] + box_x;
}
else
{
coordinates_try_MC_pbc_version.at(atom_count)[0] =
coordinates_try_MC.at(atom_count )[0];
}
}
// Check the y direction.
if ( coordinates_try_MC.at(atom_count)[1] > box_y )
{
coordinates_try_MC_pbc_version.at(atom_count)[1] =
coordinates_try_MC.at(atom_count)[1] - box_y;
}

```
\}
\{
```

    else
    {
        if ( coordinates_try_MC.at(atom_count)[1]< 0 )
        {
            coordinates_try_MC_pbc_version.at(atom_count)[1] =
            coordinates_try_MC.at(atom_count)[1] + box_y;
        }
        else
        {
            coordinates_try_MC_pbc_version.at(atom_count)[1] =
            coordinates_try_MC.at(atom_count)[1];
        }
    }
    // Check the z direction
    if ( coordinates_try_MC.at(atom_count)[2] > box_z )
    {
        coordinates_try_MC_pbc_version.at(atom_count)[2] =
        coordinates_try_MC.at(atom_count)[2] - box_z;
        }
        else
        {
            if ( coordinates_try_MC.at(atom_count)[2] < 0 )
            coordinates_try_MC_pbc_version.at(atom_count)[2] =
            coordinates_try_MC.at(atom_count)[2] + box_z;
            }
            else
            {
                coordinates_try__MC_pbc_version.at(atom_count)[2] =
            coordinates_try_MC.at(atom_count)[2];
            }
        }
    }
    }
void System_MC:: fillTrajectory(string traj_output,
vector<vector<double> > \&input_coordinates)
{
ofstream oFS;
oFS.open(traj_output.c_str(),ios::app);
if (oFS)
{
oFS << "@" << endl;
for(int site = 0; site < input_coordinates.size(); site++ )
{
oFS << atomic_number.at(site) <<" ";
oFS << fixed << setprecision(14) << input_coordinates.at(site)[0] <<" "<<
input_coordinates.at(site)[1] <<" "<< input_coordinates.at(site)[2] <<" ";
oFS << name_residue.at(site) << endl;
}
}
else

```
```

    {
    ```
    {
                cout << "ERROR: Cannot open the file" << endl;
                cout << "ERROR: Cannot open the file" << endl;
    }
    }
    oFS.close();
    oFS.close();
    oFS.clear();
    oFS.clear();
}
}
void System_MC::SetBoxOutput(string traj_output)
void System_MC::SetBoxOutput(string traj_output)
{
{
    ofstream oFS;
    ofstream oFS;
    oFS.open(traj_output.c_str(), ios::app);
    oFS.open(traj_output.c_str(), ios::app);
    if (oFS)
    if (oFS)
    {
    {
        // We past the characteristic of the trajectory
        // We past the characteristic of the trajectory
        oFS << "[trajfile]" << endl;
        oFS << "[trajfile]" << endl;
        oFS << total_MC_step+1 <<" "<< coordinates.size() << endl;
        oFS << total_MC_step+1 <<" "<< coordinates.size() << endl;
        oFS << "box "<< box_x <<" "<< box_y <<" "<< box_z << endl;
        oFS << "box "<< box_x <<" "<< box_y <<" "<< box_z << endl;
    }
    }
    else
    else
    {
    {
        cout << "ERROR: Cannot open the file" << endl;
        cout << "ERROR: Cannot open the file" << endl;
        }
        }
    oFS.close();
    oFS.close();
    oFS.clear();
    oFS.clear();
}
}
void System_MC::doEnergy(double E)
void System_MC::doEnergy(double E)
{
{
    ofstream oFS;
    ofstream oFS;
    oFS.open(energy_output.c_str(),ios::app );
    oFS.open(energy_output.c_str(),ios::app );
        if (oFS)
        if (oFS)
        {
        {
            if((compteur % 10)=0)
            if((compteur % 10)=0)
            {
            {
                oFS << compteur << " "<< E << endl;
                oFS << compteur << " "<< E << endl;
            }
            }
        }
        }
        else
        else
        {
        {
            cout << "ERROR: Cannot open the file" << endl;
            cout << "ERROR: Cannot open the file" << endl;
        }
        }
        oFS.close();
        oFS.close();
        oFS.clear();
        oFS.clear();
}
```

}

```
```

void System_MC:: Display_accepted_step()
{
cout << "_______<< endl;
cout << " compteur translate ion: " << compteur_translate_ion <<
" ... over " << total_translate_ion << " try." << endl;
cout << " compteur_translate_AuGP: " << compteur_translate_AuGP <<
" ... over " << total_translate__AuGP << " try." << endl;
cout << "compteur_rotate_success : " << compteur_rotate <<
" ... over " << total_rotate << " try." << endl;
cout <<
" << endl;

```

\section*{A.3.3 Move MC.h}

Listing A. 9 - Move_MC.h
frame
/*
*
* Initialisation of the MC moves - declaration
*
*/
\#ifndef SYSTEM_H
\#define SYSTEM_H
\#include <iostream>
\#include <vector>
\#include "Energy.h"
\#include "List.h"
\#include "MersenneTwister.h"
using namespace std;
\#define PI \(\operatorname{acos}(-1.0) / /\) definition of PI
\#define KT 2.47770902 // in KJ/mol
class System_MC \{
    private:
    /* Counter of step, type of step and corresponding step */
    int compteur;
    int compteur_rotate;
    int compteur_translate_ion;
    int compteur_translate_AuGP;
    int total_rotate;
    int total_translate_ion;
    int total_translate_AuGP;
    /* Number of MC step */
    int total_MC_step;
```

/* box_size x */
double box_x;
/* box_size y */
double box_y;
/* box_size z */
double box_z;
/* box_cell_list */
vector<double> box_cell_size;
/* Vector containing coordinate of the N_chain */
vector<vector <double>> coordinates;
/* Vector containing coordinate of the site after the pivot */
vector<vector<double> > coordinates_try_MC;
/* Vector containing coordinate of the site after the pivot */
vector<vector<double> > coordinates_try_MC_pbc_version;
/* Vector containing the atomic number of each atom */
vector<int> atomic_number;
/* vector containing the name of each atom */
vector<string > name_residue;
/* File containing the system_construct.txt file */
string input_gro;
/* File containing the system_construct.txt file */
string input_topol;
/* File containing the output traj file */
string traj_output;
/* File containing the output of energy */
string energy_output;
/* Energy of the system */
double energy;
/* Energy of the system */
double energy_expand;
/* Initialize the energy object to compute energy */
Energy energy_system;
// amplitude move and rotation
double amplitude_move;
//-- Species present in the system (ions, DNA, gold nanoparticles) --//
/* Number of atom in the DNA molecule */

```
```

int atom_DNA;
/* Number of dna */
int number__dna;
/* Number of atoms per dna */
int atom__per__dna;
/* Number of gold nanoparticle */
int gold_nano;
/* Number of free ions */
int free__ions;
/* Number of atom per gold nanoparticle */
int atom__per_gold_nano;
//_ Declaration of random generator _____//
MersenneTwister rng;
//- List object --_-_-_//
List list;
public:
System__MC ( );
~System_MC ();
/**

* @brief Constructor where dimension of
box and size of chain is read.
* @details Constructor of system object,
with generation of the lattice on the flow.
* 
* @param int number for x and y 2D lattice,
number N of beads of the chain (N small "boxes")
*/
System_MC (int total__MC_step, string input__gro, string input__topol,
string traj__output, string energy__output, int atom_DNA, int atom_per__dna,
int number_dna, int gold__nano, int atom__per__gold__nano, int free__ions);
/**
* @brief Function which picks a random
site to apply pivot algorithm.
* 

*@details Function which picks and

```
```

retrieve the site position where pivot has to be applied.
*
*@param
*/
void translate__MC(int pick__atom);
/**

* @brief Function which turn a gold nanoparticle (en particulier)
* @details Function which turn a gold nanoparticle (en particulier)
* @param
*/
void rotate__MC(int pick__gold__nano);
/**
* @brief Function which retrieves the energy according to
metropolis algorithm
* 
* @details Function which retrieves the energy according to
metropolis algorithm
* 
* @param no parameter.
*/
bool Metropolis__scheme(double probability__transition,
double energy__after, double random__metropolis);
/**
* @brief Function which take into account pdb box
during MC simulation.
* @details Function which take into account pdb box
during MC simulation.
* @param At a MC step
*/
void readInputFile();
/**
* @brief Function which take into account pdb box during MC simulation.
* @details Function which take into account pdb box during MC simulation
* @param At a MC_step.
*/
void Check_PBC_Box();

```
    vector \(<\) vector \(<\) double \(\gg\) Check_PBC_Box (vector \(<\) vector \(<\) double \(\ggg\) input_traj,
    double box_x__dir, double box_y__dir, double box_z__dir);
    /**
    * @brief Function that output the trajectory
    * @details Function that output the trajectory
    * @param
    */
    void fillTrajectory (string traj_output,
    vector \(<\) vector \(<\) double \(\rangle>\) \&input_coordinates ) ;
    /**
    * @brief Function that output the trajectory
    * @details Function that output the trajectory
    * @param
    */
    void SetBoxOutput (string traj_output);
    /**
    * @brief Function that give the energy through the time step
    * @details Function that give the energy throught the time step;
    * @param energy and step.
    */
    void doEnergy (double \(E)\);
    /**
    * @brief Function that display the accepted step
    * @details Function that display the accepted step
    * @param
    */
    void Display__accepted_step ();
\};
\#endif

\section*{A.3.4 Potential_energy_calculation.cpp}

Listing A. 10 - Potential_energy_calculation.cpp
frame
```

\#include "Energy.h"
using namespace std;
Energy:: Energy()
{}
Energy::Energy(string input_topol, string input_gro, int atom_DNA,
int atom_per_dna, int number_dna, int gold_nano,
int atom_per_gold_nano, int free_ions)
{
// Initialization of the species in the system.
this }->\mathrm{ atom_DNA = atom_DNA;
this }->\mathrm{ atom_per_dna = atom_per_dna;
this ->number_dna = number_dna;
this }->\mathrm{ gold__nano = gold_nano;
this }->\mathrm{ free_ions = free_ions;
this ->atom_per_gold_nano = atom_per_gold_nano;
this }->\mathrm{ >input_topol = input_topol;
this ->input_gro = input_gro;
// Let's initialize the box size.
ReadBoxSize();
this ->alpha =6/(min_element(box_x, box_y, box_z));
this }->\mathrm{ rcut_sq = (min_element(box_x, box_y, box_z)/2)*
(min_element(box_x, box_y, box_z)/2);
this }->\mathrm{ Kmax = 6;
this }->V=\quad=box_x*box_y*box_z;
// Let's read the topology.
ReadTopology ();
//We initialize the vector with the complex sum used in the Im_Ewald
SUM.resize(2*Kmax +1, vector <vector <complex<double> > >
(2*Kmax +1, vector <complex<double> > (2*Kmax +1,0)));
SUM_buffer.resize( }2*\mathrm{ Kmax +1, vector <vector <complex<double > > >
(2*Kmax +1, vector<complex<double> > (2*Kmax +1,0)));
}
Energy::~Energy()
{}
void Energy::ReadTopology()
{
//clock_t tic = clock();
ifstream iFS;
iFS.open(input_topol.c_str());
string line;

```
```

vector<string> split_line;
int buffer_charge_index;
double buffer_charge;
vector<int> buffer_bond__index (2,0);
vector<double> buffer_bond (2,0.0);
vector<int> buffer_angle_index (3,0);
vector<double> buffer_angle(2,0.0);
int buffer_vdw__index;
vector<double> buffer_vdw(2,0.0);
if (iFS)
{
while(getline(iFS, line))
{
if(line[0] != '['\&\& line [0] !=';')
{
split_line = explode(line, '\t');
switch(split_line.size())
{
case 2:
buffer_charge_index = stoi(split_line.at(0));
buffer_charge = stod(split_line.at(1));
charge_index.push_back(buffer_charge_index);
charge.push_back(buffer_charge);
break;
case 4:
buffer_bond_index.at(0) = stoi(split_line.at(0));
buffer_bond_index.at(1) = stoi(split_line.at(1));
buffer_bond.at(0) = stod(split_line.at(2));
buffer_bond.at(1) = stod(split_line.at(3));
bond__index.push_back(buffer__bond_index);
bond.push_back(buffer_bond);
break;
case 5:
buffer__angle_index.at(0) = stoi(split_line.at(0));
buffer__angle_index.at(1) = stoi(split_line.at(1));
buffer_angle_index.at(2) = stoi(split_line.at(2));
buffer_angle.at(0) = stod(split_line.at(3));
buffer_angle.at(1) = stod(split_line.at(4));

```

109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
```

            angle__index.push__back(buffer__angle__index);
            angle.push__back(buffer__angle);
                    break;
                    case 3:
                        buffer__vdw__index = stoi(split__line.at (0));
            buffer__vdw.at(0) = stod(split_line.at (1));
            buffer__vdw.at(1) = stod(split__line.at(2));
            vdw__index.push__back(buffer__vdw__index );
            vdw.push__back(buffer__vdw );
                                    break;
                    }
            }
        }
    }
    else
{
cout << " Cannot open " << input__topol << endl;
}
}
bool Energy::Test_Exclusion__volume()
{
double r_x = 0;
double r__y = 0;
double r__z = 0;
double RX = 0;
double RY = 0;
double RZ = 0;
double r__sq_x = 0;
double r__sq__y = 0;
double r__sq__z = 0;
double r_sq = 0;
double r__vdw = = ;
double r__vdw__sq = 0;
double vdw__pick__atom__radius;
// Index for the pbc box
int new__x;
int new_y;
int new__z;
for( int atom = 0; atom < coordinates__atom__picked.size(); atom++ )
{
vdw__pick_atom_radius = vdw.at(index_pbc__list.at(atom))[1];

```
// Test interpenetration with the other atoms through the cell lists.
for (int i \(=\) index_cell_chosen_particles_after.at (atom \()[0]-1\);
\(\mathrm{i}<=\) index__cell__chosen_particles__after.at (atom) \([0]+1 ; i++\) )
\{
    for ( int \(j=\) index__cell__chosen__particles__after.at (atom \()[1]-1\);
    \(j<=\) index_cell__chosen_particles__after.at (atom) \([1]+1 ; j++\) )
    \{
        for (int \(k=\) index_cell__chosen_particles__after.at (atom) \([2]-1\);
        \(\mathrm{k}<=\) index__cell__chosen_particles__after.at (atom) \([2]+1 ; \mathrm{k}++\) )
        \{
            new_x \(=\quad\) i;
            if \((\mathrm{i}=0) \quad\) new_x \(+=\) box_shift_PBC;
            if \((\mathrm{i}=(\) box_shift__PBC +1\())\) new__x \(-=\) box__shift__PBC;
            new_y \(=\mathrm{j}\);
            if \((\overrightarrow{\mathrm{j}}=0) \quad\) new_y \(+=\) box_shift__PBC;
            if \((\mathrm{j}=(\) box__shift_PBC +1\())\) new_y \(-=\) box__shift__PBC;
            new_z \(=\mathrm{k}\);
            if \((\mathrm{k}=0) \quad\) new_z \(+=\) box__shift__PBC;
            if \((\mathrm{k}=(\) box__shift_PBC +1\())\) new_z \(\quad-=\) box__shift__PBC;
            for (int atom \(2=0 ;\) atom2
            \(<\) particle__cell__without_chose__atom[new_x][new_y][new_z].size ();
            atom \(2++\) )
            \{
                \(\mathrm{RX}=\)
                    (coordinates.at (particle__cell__without__chose__atom
                    [new_x][new_y][new_z][atom2])[0] + PBC_cell[i][j][k][0] * box_x)
                    - coordinates__atom_picked.at (atom) [0];
                    \(R Y=\)
                    (coordinates.at (particle__cell__without_chose_atom
                    [new_x][new_y][new_z][atom2])[1] + PBC_cell[i][j][k][1] * box_y)
                    - coordinates_atom_picked.at (atom) [1];
                    \(\mathrm{RZ}=\)
                    (coordinates.at (particle__cell__without__chose__atom
                    [new_x][new_y][new_z][atom2])[2] +PBC_cell[i][j][k][2]* box_z)
                    - coordinates_atom_picked.at (atom) [2];
                    \(r \_s q=R X * R X+R Y * R Y+R Z * R Z ;\)
                    r__vdw =
                    \(0.5 *\) (vdw. at (particle__cell__without__chose__atom
                    [new_x][new_y][new_z][atom2])[1] + vdw__pick__atom_radius );
                    \(r_{\text {__vdw__sq }}=\quad\) r_sdw \(*\) r_cvdw ;
                    if \(\left(r_{\text {__ }} s q<=r \_v d w \_s q\right)\)
                    \{
                return true;
```

                                    }
                    }
                }
            }
        }
    }
    return false;
    double Energy::Calculate_Real_Ewald_change()
// Calculation of the real part
double local_energy__before = 0;
double local_energy_after = 0;
double RX;
double RY;
double RZ;
double r_sq = 0;
double r = 0;
double delta_Real_change = 0;
// Withdraw the energy of the picked atom from total_energy with the former position
for( int atom = 0; atom < coordinates_atom_picked.size(); atom++ )
{
for( int i = 0; i < coordinates_without_pick_index.size(); i++ )
{
if(charge.at(coordinates_without_pick_index.at(i)) == 0 ||
coordinates_without_pick_index.at(i) == index__pbc_list.at(atom))
{
continue;
}
RX =
coordinates.at(coordinates_without_pick_index.at(i))[0] -
coordinates.at(index_pbc_list.at(atom))[0];
RY =
coordinates.at(coordinates_without_pick_index.at(i))[1] -
coordinates.at(index_pbc_list.at(atom))[1];
RZ =
coordinates.at(coordinates_without__pick_index.at(i))[2] -
coordinates.at(index_pbc_list.at(atom))[2];
// Other version for PBC condition testing //
RX = RX - box_x * round(RX / box_x);

```
\}
```

        RY = RY - box_y * round (RY / box_y);
            RZ = RZ - box_z * round (RZ / box_z);
            r_sq = RX*RX + RY*RY + RZ*RZ;
            if( r_sq > rcut_sq)
            {
        continue;
            }
                r = sqrt(r_sq);
                local_energy_before = local_energy__before +
                charge.at(coordinates_without__pick__index.at(i)) *
                charge.at(index_pbc_list.at(atom))* erfc(alpha*r)/r;
    }
    }
// Add the energy of the picked atom from total_energy with the new position.
for( int atom = 0; atom < coordinates_atom_picked.size(); atom++ )
{
for( int i = 0; i < coordinates_without_pick_index.size(); i++ )
{
if(charge.at(coordinates_without__pick_index.at(i)) == 0 ||
coordinates_without_pick__index.at(i) == index__pbc_list.at(atom))
{
continue;
}
RX =
coordinates.at(coordinates_without__pick_index.at(i))[0] -
coordinates_atom_picked.at(atom)[0];
RY =
coordinates.at(coordinates_without__pick_index.at(i))[1] -
coordinates_atom_picked.at(atom)[1];
RZ =
coordinates.at(coordinates_without__pick__index.at(i))[2] -
coordinates_atom_picked.at(atom)[2];
RX = RX - box_x * round (RX / box_x);
RY = RY - box_y * round (RY / box_y);
RZ = RZ - box_z * round(RZ / box_z);
r_sq = RX*RX + RY*RY + RZ*RZ;
if( r_sq > rcut_sq)
{
continue;
}
else
{

```
                    r sqrt(r_sq);
                    local__energy__after \(=\) local__energy__after +
                    charge.at (coordinates__without__pick__index.at (i)) *
                            charge.at (index_pbc_list.at (atom)) \(* \operatorname{erfc}(\operatorname{alpha} * r) / r\);
                    \}
        \}
    \(\}\)
    delta__Real_change \(=-\operatorname{local\_ }\) energy__before + local__energy__after;
    return delta__Real__change;
\(\}\)
double Energy:: Calculate__Real_Ewald__Init ()
\{
    double RX;
    double RY;
    double RZ;
    double r_sq \(=0\);
    double \(r=0\);
    // Calculation of the real part
    double Ereal \(=0\);
    int counter__index_nano;
    int counter__index__dna;
    for ( int \(i=0 ; i<\) coordinates.size() \(-1 ; i++\) )
    \{
    if \(((\) charge.at \((i)=0))\)
    \{
        continue;
    \}
    // Index for the dna molecules.
    if ( i \(>=0 \& \&\) i \(<\) atom_DNA)
    \{
        counter_index__dna \(=(\) int \()\) floor \((i /\) atom__per__dna \() ;\)
    \(\}\)
    // Index for the Gold nanoparticles.
    if \((\) i \(>(\) atom_DNA + free__ions -1\())\)
    \{
        counter__index__nano =
        (int)floor \(\left(\left(\mathrm{i}-\left(\right.\right.\right.\) atom_DNA \(+\mathrm{free} \_\)ions \(\left.)\right) /\)atom__per_gold__nano \()\);
    \(\}\)
    for ( int \(j=i+1 ; j<\) coordinates.size () ; \(j++\) )
    \{
        if \(((\) charge.at \((j)=0))\)
```

{
continue;
}
RX = coordinates.at (i) [0] - coordinates.at (j) [0];
RY = coordinates.at (i) [1] - coordinates.at (j) [1];
RZ = coordinates.at (i) [2] - coordinates.at (j) [2];
// Rejection intermolecular interaction in each dna.
if( i >= (counter__index__dna*(atom__per_dna)) \&\&
i <= (counter__index__dna*(atom__per_dna) + (atom_per__dna - 1)) \&\&
j >= (counter_index__dna*(atom__per_dna)) \&\&
j <= (counter_index__dna*(atom__per_dna) + (atom__per__dna - 1))
)
continue;
}
// Rejection intermolecular interaction in each of the gold nanoparticles
if( i >= (atom_DNA + free__ions + counter__index__nano*(atom__per__gold__nano)
i}<=(\mathrm{ atom_DNA + free__ions + counter__index__nano*(atom__per__gold__nano) +
(atom__per__gold__nano - 1)) \&\&
j >= (atom_DNA + free_ions + counter_index_nano*(atom_per_gold_nano)
j <= (atom_DNA + free__ions + counter__index__nano*(atom_per_gold_nano) +
(atom_per_gold_nano - 1))
)
{
continue;
}
/* Other version for PBC condition testing */
RX = RX - box_x * round(RX / box_x );
RY = RY - box_y * round(RY / box_y );
RZ = RZ - box__z * round(RZ / box_z );
r__sq}=RX*RX + RY*RY + RZ*RZ
if( r__sq > rcut_sq)
{
}
else
{
r = sqrt(r__sq);
Ereal = Ereal + charge.at(i) * charge.at (j)* erfc(alpha*r)/r ;
}

```
        \}
\}

431
```

    return Ereal;
    }

```
double Energy:: Calculate__Im__Ewald__change()
\{
double delta_Im_change \(=0.0\);
double RL,RM,RN,KK,KSQ,AK;
double Eim_before \(=0\);
double Eim_after \(=0\);
int \(\mathrm{n}, \mathrm{l}, \mathrm{m}\);
int K;
// Declaration of \(\exp (i k r)\) and exponential factors;
complex<double \(>\) expikr__before;
complex<double> expikr_after;
vector \(<\) vector \(<\) complex \(<\) double \(\ggg>\)
complex__x_before (coordinates__atom__picked.size () ,
vector \(<\) complex \(<\) double \(\gg(2 * \operatorname{Kmax}+1,0))\);
vector \(<\) vector \(<\) complex \(<\) double \(\ggg>\)
complex_y_before (coordinates__atom_picked.size (),
vector \(<\) complex \(<\) double \(\gg(2 * \operatorname{Kmax}+1,0))\);
vector \(<\) vector \(<\) complex \(<\) double \(\ggg\)
complex_z__before (coordinates__atom__picked.size () ,
vector \(<\) complex \(<\) double \(\gg(2 * \operatorname{Kmax}+1,0))\);
vector \(<\) vector \(<\) complex \(<\) double \(\ggg\)
complex_x_after (coordinates__atom_picked.size (),
vector \(<\) complex \(<\) double \(\gg(2 * \operatorname{Kmax}+1,0))\);
vector \(<\) vector \(<\) complex \(<\) double \(\ggg\)
complex_y__after (coordinates__atom__picked.size () ,
vector \(<\) complex \(<\) double \(\gg(2 * \operatorname{Kmax}+1,0))\);
vector \(<\) vector \(<\) complex \(<\) double \(\ggg\)
complex_z__after (coordinates__atom__picked.size () ,
vector \(<\) complex \(<\) double \(\gg(2 * \operatorname{Kmax}+1,0))\);
for \((\) int atom \(=0 ;\) atom \(<\) coordinates_atom_picked.size (); atom++ )
\{
    for ( \(\mathrm{K}=-\mathrm{Kmax} ; \mathrm{K}<=\mathrm{Kmax} ; \mathrm{K}++\) )
    \{
            complex_x_before.at (atom) \([\mathrm{K}+\mathrm{Kmax}]=\)
            complex <double>(cos (double (K) \(* 2 * \mathrm{PI} *\)
            coordinates.at (index_pbc_list.at (atom) ) [0]/box_x),
            \(\sin (\) double \((\mathrm{K}) * 2 * \mathrm{PI} *\) coordinates.at (index_pbc_list.at (atom) ) [0]/box_x));
```

                complex_y_before.at(atom)[K+Kmax ] =
                complex <double>(cos(double (K) *2*PI*
                coordinates.at(index_pbc_list.at(atom))[1]/ box_y),
                sin(double(K)*2*PI*coordinates.at(index_pbc_list.at(atom))[1]/box_y));
                complex_z_before.at(atom)[K+Kmax] =
                complex <double>(cos(double(K) *2*PI*
                coordinates.at(index_pbc_list.at(atom))[2]/ box__z),
                sin(double(K)*2*PI*coordinates.at(index_pbc_list.at(atom))[2]/box_z));
            complex_x_after.at(atom)[K+Kmax] =
            complex <double>(cos(double (K) *2*PI*
            coordinates_atom_picked.at(atom)[0]/box_x),
            sin(double(K)*2*\overline{PI}*\mathrm{ coordinates_atom_picked.at(atom)[0]/box_x));}
            complex_y__after.at(atom)[K+Kmax] =
            complex <double>(cos(double(K) *2*PI*
            coordinates_atom_picked.at(atom)[1]/ box_y),
            sin(double(K)*2*PI*coordinates_atom_picked.at(atom)[1]/box_y));
                complex_z_after.at(atom)[K+Kmax ] =
                complex <double>(cos (double (K)*2*PI*
                coordinates_atom_picked.at(atom)[2]/box_z),
                sin(double(K)*2*PI*coordinates_atom_picked.at(atom)[2]/box_z));
            }
    }
// Withdraw the energy of the picked atom from
// total_energy with the former position.
for ( l = -Kmax; l <= Kmax; l++ )
{
RL = double(1)/box_x;
for ( m = -Kmax; m <= Kmax; m++ )
{
RM = double(m)/box_y;
for ( n = -Kmax; n <= Kmax; n++ )
{
RN = double(n)/box_z;
// test of magnitude of K vector
KK = l*l + m*m + n*n;
if( KK >= Kmax*Kmax+2)
{
continue;
}
if( l = 0 \&\& m=0 \&\& n=0)

```

539 540 541
```

        {
    ```
        {
        continue;
        continue;
            }
            }
                KSQ = RL*RL + RM*RM + RN*RN;
                KSQ = RL*RL + RM*RM + RN*RN;
                AK}=(1/(2*PI*V))*\operatorname{exp}(-(PI*PI*KSQ)/(alpha*alpha))/KSQ
                AK}=(1/(2*PI*V))*\operatorname{exp}(-(PI*PI*KSQ)/(alpha*alpha))/KSQ
                Eim__before = Eim__before +
                Eim__before = Eim__before +
                AK*( conj ((this }->\mathrm{ SUM__buffer.at (l+Kmax ) [m+Kmax ][n+Kmax]))
                AK*( conj ((this }->\mathrm{ SUM__buffer.at (l+Kmax ) [m+Kmax ][n+Kmax]))
                                    *(this }->>\mathrm{ SUM__buffer.at (l+Kmax ) [m+Kmax ] [n+Kmax ])).real ();
                                    *(this }->>\mathrm{ SUM__buffer.at (l+Kmax ) [m+Kmax ] [n+Kmax ])).real ();
                                    for(int atom = 0; atom < coordinates_atom_picked.size (); atom++)
                                    for(int atom = 0; atom < coordinates_atom_picked.size (); atom++)
                                    {
                                    {
    if(charge.at(index__pbc__list.at (atom)) =0)
    if(charge.at(index__pbc__list.at (atom)) =0)
    {
    {
        continue;
        continue;
        }
        }
    // Form cos and sin for the picked particle__before
    // Form cos and sin for the picked particle__before
    expikr__before = complex_x__before.at(atom)[l+Kmax]*
    expikr__before = complex_x__before.at(atom)[l+Kmax]*
    complex__y__before.at(atom)[m+Kmax]*
    complex__y__before.at(atom)[m+Kmax]*
    complex__z__before.at(atom)[n+Kmax];
    complex__z__before.at(atom)[n+Kmax];
    // Form cos and sin for the picked particle__after
    // Form cos and sin for the picked particle__after
    expikr__after = complex__x__after.at(atom)[l+Kmax]*
    expikr__after = complex__x__after.at(atom)[l+Kmax]*
    complex_y__after.at(atom) [m+Kmax]*
    complex_y__after.at(atom) [m+Kmax]*
    complex_z__after.at(atom)[n+Kmax];
    complex_z__after.at(atom)[n+Kmax];
    this }->>\mathrm{ SUM__buffer.at (l+Kmax) [m+Kmax ][n+Kmax ] =
    this }->>\mathrm{ SUM__buffer.at (l+Kmax) [m+Kmax ][n+Kmax ] =
    this }->>\mathrm{ SUM__buffer.at (l+Kmax) [m+Kmax ][n+Kmax ] +
    this }->>\mathrm{ SUM__buffer.at (l+Kmax) [m+Kmax ][n+Kmax ] +
    charge.at(index__pbc__list.at(atom))*
    charge.at(index__pbc__list.at(atom))*
            (-expikr__before + expikr_after);
            (-expikr__before + expikr_after);
                }
                }
                Eim__after = Eim__after +
                Eim__after = Eim__after +
                AK*( conj ((this }->\mathrm{ SUM_buffer.at (l+Kmax ) [m+Kmax ] [n+Kmax ]))*
                AK*( conj ((this }->\mathrm{ SUM_buffer.at (l+Kmax ) [m+Kmax ] [n+Kmax ]))*
                                ( this }->>SUM__buffer.at(l+Kmax) [m+Kmax][n+Kmax])).real()
                                ( this }->>SUM__buffer.at(l+Kmax) [m+Kmax][n+Kmax])).real()
                }
                }
        }
        }
    }
    }
    delta_Im__change = - Eim__before }\quad+\mathrm{ Eim__after;
    delta_Im__change = - Eim__before }\quad+\mathrm{ Eim__after;
    // Retrieve the total time of the program.
    // Retrieve the total time of the program.
    return delta__Im__change;
    return delta__Im__change;
}
}
double Energy:: Calculate__Im__Ewald__Init()
double Energy:: Calculate__Im__Ewald__Init()
{
```

{

```
```

double Eim = 0.0;
double RL,RM,RN,KK,KSQ,AK;
int n, l,m;
int K;
// Declaration of exp(ikr) and store exponential factors;
vector<complex<double> > expikr(coordinates.size());
vector<vector <complex<double> > >
complex_x(coordinates.size(), vector <complex<double> > (2*Kmax +1,0));
vector<vector<complex<double> > >
complex_y(coordinates.size(), vector <complex<double>> (2*Kmax +1,0));
vector<vector<complex<double> > >
complex_z(coordinates.size(), vector <complex<double>> (2*Kmax +1,0));
for( n = 0; n <coordinates.size(); n++)
{
for( K = -Kmax; K <= Kmax; K++ )
{
complex_x.at(n)[K+Kmax] = complex <double>(cos(double (K)*2*PI*
coordinates.at(n)[0]/box_x),
sin(double(K)*2*PI*coordinates.at(n)[0]/ box_x));
complex_y.at(n)[K+Kmax ] = complex <double>(cos(double (K)*2*PI*
coordinates.at(n)[1]/ box_y),
sin(double(K)*2*PI*coordinates.at (n)[1]/ box_y));
complex_z.at(n)[K+Kmax ] = complex <double>(cos(double (K)*2*PI*
coordinates.at(n)[2]/box_z),
sin(double(K)*2*PI*coordinates.at (n)[2]/ box__z));
}
}
// Start loop over wave vectors (L,M,N) and number of atoms.
for ( l = -Kmax; l <= Kmax; l++ )
{
RL = double(1)/box_x;
for ( m = -Kmax; m <= Kmax; m++ )
{
RM = double(m)/box_y;
for ( n = -Kmax; n <= Kmax; n++ )
{
RN = double(n)/box_z;
// test of magnitude of K vector
KK = l*l + m*m + n*n;
if( KK >= Kmax*Kmax+2)

```
```

6 4 7
648
6 4 9
650
6 5 1
6 5 2
6 5 3
6 5 4
6 5 5
656
6 5 7
6 5 8
6 5 9
660
6 6 1
6 6 2
63
664
6 6 5
666
667

```
    {
```

    {
                continue;
                continue;
            }
            }
            if( l = 0 && m=0 && n=0)
            if( l = 0 && m=0 && n=0)
            {
            {
                continue;
                continue;
            }
            }
                KSQ = RL*RL + RM*RM + RN*RN;
                KSQ = RL*RL + RM*RM + RN*RN;
                AK = (1/(2*PI*V))*exp(-(PI*PI*KSQ)/(alpha*alpha))}/\textrm{KSQ}
                AK = (1/(2*PI*V))*exp(-(PI*PI*KSQ)/(alpha*alpha))}/\textrm{KSQ}
                // Form cos and sin for each particle
                // Form cos and sin for each particle
                for (int j = 0; j < coordinates.size(); j++ )
                for (int j = 0; j < coordinates.size(); j++ )
            {
            {
                expikr.at(j) =
                expikr.at(j) =
                complex_x.at(j)[l+Kmax]*complex_y.at (j) [m+Kmax]*
                complex_x.at(j)[l+Kmax]*complex_y.at (j) [m+Kmax]*
                complex_z.at(j)[n+Kmax];
                complex_z.at(j)[n+Kmax];
            }
            }
                // Form sum for each species
                // Form sum for each species
                for (int j = 0; j < coordinates.size(); j++ )
                for (int j = 0; j < coordinates.size(); j++ )
            {
            {
                // We do not count interaction between phosphate in the DNA
                // We do not count interaction between phosphate in the DNA
                if((charge.at(j) = 0))
                if((charge.at(j) = 0))
            {
            {
                continue;
                continue;
            }
            }
                this }->\mathrm{ SUM. at (l+Kmax) [m+Kmax ][n+Kmax ] =
                this }->\mathrm{ SUM. at (l+Kmax) [m+Kmax ][n+Kmax ] =
                this }->\mathrm{ SUM.at(l+Kmax) [m+Kmax][n+Kmax] + charge.at(j)*expikr.at(j);
                this }->\mathrm{ SUM.at(l+Kmax) [m+Kmax][n+Kmax] + charge.at(j)*expikr.at(j);
            }
            }
                this }->\mathrm{ SUM_buffer.at(l+Kmax)[m+Kmax][n+Kmax] =
                this }->\mathrm{ SUM_buffer.at(l+Kmax)[m+Kmax][n+Kmax] =
                this }->\mathrm{ SUM. at (l+Kmax)[m+Kmax][n+Kmax];
                this }->\mathrm{ SUM. at (l+Kmax)[m+Kmax][n+Kmax];
                // Accumulate K-space potential energy
                // Accumulate K-space potential energy
                    Eim = Eim +
                    Eim = Eim +
                AK*(conj ((this }->\mathrm{ SUM. at (l +Kmax) [m+Kmax ][n+Kmax ]))*
                AK*(conj ((this }->\mathrm{ SUM. at (l +Kmax) [m+Kmax ][n+Kmax ]))*
                (this }->\mathrm{ SUM. at (l+Kmax)[m+Kmax][n+Kmax])).real();
                (this }->\mathrm{ SUM. at (l+Kmax)[m+Kmax][n+Kmax])).real();
            }
            }
        }
        }
    }
    }
    return Eim;
    return Eim;
    }
double Energy:: Calculate_self_Ewald()
double Energy:: Calculate_self_Ewald()
{
// Constant term in the self interaction contribution
// Constant term in the self interaction contribution
double Spe = 0.0;
double Spe = 0.0;
double sum = 0;

```
double sum = 0;
```

```
    for ( int i = 0; i < charge.size(); i++)
    {
        // We do not count interaction between phosphate in the DNA
        if((charge.at(i)=0))
        {
            continue;
        }
        sum = sum + charge.at(i)*charge.at(i);
    }
    Spe = -(alpha/sqrt(PI))*sum;
    return Spe;
double Energy::Calculate_intermolecular_Ewald()
    double EinterMol = 0.0;
    double r_sq = 0;
    double r = 0;
    double RX;
    double RY;
    double RZ;
    double r_sq__expand = 0;
    double r_expand = 0;
    double RX_expand;
    double RY_expand;
    double RZ_expand;
    // Intermolecular term for the DNA molecule.
    if(number_dna > 0)
    {
        for( int i = 0; i < atom_per_dna; i++ )
        {
            if(charge.at(i) = 0)
            {
                continue;
            }
            for( int j = i+1; j < atom_per_dna; j++ )
            {
                RX = coordinates.at(i)[0]-coordinates.at (j) [0];
                    RY = coordinates.at(i)[1]-coordinates.at(j)[1];
                RZ = coordinates.at(i)[2]-coordinates.at(j ) [2];
                    // Other version for PBC condition testing
                    RX = RX - box_x * round(RX / box_x);
                    RY = RY - box_y * round(RY / box_y);
```

\}
\{
$\mathrm{RZ}=\mathrm{RZ}-$ box_z $*$ round ( $\mathrm{RZ} /$ box_z);
$r_{-} s q=R X * R X+R Y * R Y+R Z * R Z ;$
$\mathrm{r} \quad=\operatorname{sqrt}\left(\mathrm{r} \_\right.$sq);
EinterMol = EinterMol
+ number_dna*(charge.at(i)*
charge.at $(j)) *(\operatorname{erf}(\operatorname{alpha} * r) / r)$;
\}
\}
\}
// Intermolecular term for the gold nanoparticles
if (gold_nano > 0)
\{
for ( int i $=$ atom_DNA + free_ions;
i < atom_DNA + free_ions + atom_per_gold_nano; i++ )
\{
if (charge. at (i) $=0$ )
\{
continue;
\}
for (int $\mathrm{j}=\mathrm{i}+1 \quad ; \mathrm{j}<$ atom_DNA + free_ions + atom_per_gold_nano $; \mathrm{j}++$ )
\{
$R X=$ coordinates.at(i)[0]-coordinates.at(j)[0];
$R Y=$ coordinates.at (i) $[1]-$ coordinates.at (j $)[1]$;
$R Z=$ coordinates.at(i)[2]-coordinates.at(j)[2];
// Other version for PBC condition testing
$\mathrm{RX}=\mathrm{RX}-$ box_x $*$ round $(\mathrm{RX} /$ box_x $)$;
$R Y=R Y-b o x \_y * \operatorname{round}\left(R Y / b o x \_y\right) ;$
$\mathrm{RZ}=\mathrm{RZ}-\mathrm{box}_{\mathrm{z}} \mathrm{z}$ * round ( $\mathrm{RZ} /$ box_z);
$r \_s q=R X * R X+R Y * R Y+R Z * R Z ;$
$\mathrm{r} \quad=\operatorname{sqrt}\left(\mathrm{r} \_\right.$sq);
EinterMol $=$ EinterMol + gold_nano $*($ charge.at $(i) *$
charge.at (j) ) * (erf (alpha*r)/r);
\}
\}
\}
return EinterMol;
\}
vector<string> Energy: explode (const string\& s, const char\& c)
\{
string buff $\{$ " " $\}$;
vector<string $>\mathrm{v}$;

809

```
    for(auto n:s)
    {
            if(n != c) buff+=n; else
            if(n=c && buff != "") { v.push_back(buff); buff = " "; }
    }
    if(buff != " ") v.push_back(buff);
    return v;
}
double Energy::getEnergy()
{
    return energy;
}
void Energy::InitTrajFirstStep(vector<vector<double> > &input_pbc)
{
    this }->\mathrm{ coordinates = input_pbc;
    // Since the intramolecular and the self energy does not depend on position,
    // we decide to calculate them only one time.
    // We initialize the energy of the system.
    energy = Eo*(Calculate_self_Ewald() + Calculate_intermolecular_Ewald() +
    Calculate_Real_Ewald_Init() + Calculate_Im_Ewald_Init ());
}
void Energy:: ChangeTraj(vector<vector<double> > &coordinates_atom_picked, vector<int> &index
vector<vector<vector<vector<int>>>> &particle_cell_without_chose_atom,
vector<vector<int>> &index_cell_chosen_particles_after)
{
    this }->\mathrm{ coordinates_atom_picked = coordinates_atom_picked;
    this ->index_pbc_list = index_pbc_list;
    this }->\mathrm{ particle_cell_without_chose_atom = particle_cell_without_chose_atom;
    this }->\mathrm{ index_cell_chosen_particles_after = index_cell_chosen__particles_after;
    coordinates_without_pick_index.clear();
    bool boolean = false;
    for(int i = 0; i < coordinates.size(); i++)
    {
            boolean = false;
            for (int j = 0; j < index_pbc_list.size(); j++)
            {
                if(i= index_pbc_list.at(j))
                {
                boolean = true;
```

```
                    break;
                }
        }
        if(boolean = false)
        {
        coordinates__without_pick_index.push__back(i);
        }
    }
}
void Energy::Call_Calculate_Energy()
    energy = energy + Eo*(Calculate_Real_Ewald_change() +
    Calculate_Im_Ewald_change());
}
void Energy::AcceptMoveUpdate(bool IsAccepted, double E)
{
    if(IsAccepted = true)
    {
        // If accepted, then the buffer and coordinates are updated with the same
        coordinates of the chosen atoms.
        for(int atom = 0; atom < coordinates_atom_picked.size(); atom++)
        {
            this }->\mathrm{ coordinates.at(index_pbc_list.at(atom))[0] =
            coordinates_atom_picked.at(atom)[0];
                this }->\mathrm{ coordinates.at(index_pbc_list.at(atom))[1] =
                coordinates_atom_picked.at(atom) [1];
                this }->\mathrm{ coordinates.at(index_pbc_list.at(atom))[2] =
                coordinates_atom_picked.at(atom)[2];
            }
            // For the imaginary part, we update
            for (int l = -Kmax; l <= Kmax; l++ )
            {
                for (int m = -Kmax; m <= Kmax; m++ )
                {
                    for (int n = -Kmax; n <= Kmax; n++ )
                    {
                                    this ->SUM. at (1+Kmax)[m+Kmax ][n+Kmax ] =
                                    this }->>SUM_buffer.at(l+Kmax)[m+Kmax][n+Kmax]
                    }
                }
            }
}
    else
    {
```

\{

```
        // If refused, then we don't touch the coordinates.
        this ->energy = E;
        // For the imaginary part, we update with the former SUM.
        for (int l = -Kmax; l <= Kmax; l++ )
        {
            for (int m = -Kmax; m <= Kmax; m++ )
            {
            for (int n = -Kmax; n <= Kmax; n++ )
            {
                this }->\mathrm{ SSUM_buffer.at (l+Kmax)[m+Kmax][n+Kmax] =
                this }->\mathrm{ SUMM. at ( l+Kmax)[m+Kmax][n+Kmax];
            }
        }
        }
    }
}
void Energy ::ReadBoxSize()
{
    ifstream iFS;
    iFS.open(input_gro.c_str());
    int atoms;
    if (iFS)
    {
        iFS >> atoms;
        iFS >> this }->\mathrm{ box_x >> this }->\mathrm{ boox_y >> this }->\mathrm{ >box_z;
    }
    else
    {
        cout << "Cannot open file " << input_gro << endl;
    }
}
void Energy::Set_PBC_cell_list(vector<vector<vector<vector<int> > > > PBC_cell)
{
    this }->\mathrm{ PBC__cell = PBC_cell;
}
void Energy::Set_PBC_vector_shift(int box_shift_PBC)
{
    this }->\mathrm{ box_shift_PBC = box_shift__PBC;
}
double Energy::min_element(double dir_x, double dir_y, double dir_z)
{
    double minimum = dir_x;
    if (minimum >= dir_y) minimum = dir_y;
```


## A.3.5 Potential_energy_calculation.h

Listing A. 11 - Move__MC.h

frame
/*
*

* Class to calculate energy of our system at a given frame
* 

*/
\#ifndef ENERGY_H
\#define ENERGY_H
\#include <iostream>
\#include <vector>
\#include <string>
\#include <fstream $>$
\#include <cmath>
\#include <complex>
\#include <iomanip>
\#include <bits/stdc++.h>
\#define PI $\operatorname{acos}(-1.0)$
\#define eps0 8.854188*pow $(10,-12)$
\#define q $1.6021765 * \operatorname{pow}(10,-19)$
\#define Angstrom pow $(10,-10)$
\#define Na $6.02214129 *$ pow $(10,23)$
\#define eps__water 78
\#define $\operatorname{Eo}\left((q * q) /\left(4 * \mathrm{PI} * \mathrm{eps} 0 * e p s \_\right.\right.$water $*$ Angstrom $\left.)\right) *(\mathrm{Na} / 1000)$
using namespace std;
class Energy\{
private:
//_ General features $\quad \ldots$ /
$/ *$ Vector containing coordinate of the system (take account of PBC) */
vector $<$ vector $<$ double $\rangle>$ coordinates ;
/* List of index of the moved entities and input quantites at each step */
vector $<$ vector $<$ double $\gg$ coordinates__atom_picked;
vector $<$ int $>$ index_pbc__list;

```
vector<int> coordinates__without__pick__index;
vector<vector<vector <vector <int > > > > particle__cell__without__chose__atom;
vector <vector <int > > index__cell_chosen__particles__before;
vector<vector<int> > index__cell__chosen__particles__after;
/* File containing the topology */
string input__topol;
/* File containing the topology */
string input__gro;
/* Vector containing the index of charge in the topology file */
vector <int> charge__index;
/* Vector containing the value of charge in the topology file */
vector < double> charge;
/* Vector containing the index of bond in the topology file */
vector < vector <int > > bond__index;
/* Vector containing the value of bond in the topology file */
vector <vector <double }>>>\mathrm{ bond;
/* Vector containing the index of angle in the topology file */
vector<vector<int> > angle__index;
/* Vector containing the value of angle in the topology file */
vector <vector < double > > angle;
/* Vector containing the index of vdw in the topology file */
vector<int> vdw__index;
/* Vector containing the value of vdw in the topology file */
vector <vector < double > > vdw;
/* Vector containing the index for non bonded-interaction */
double energy;
// !!! For the entropic part, we will use further analysis in a separate code.
/* we define the SUM appering in the imaginary part of ewald */
vector <vector <vector < complex <double >>>> SUM;
/* we define the SUM__buffer appering in the imaginary part of ewald */
vector <vector <vector < complex <double > > > > SUM__buffer;
/* Vector to stock the name of residue*/
vector<string> name__residue;
//__ For the electrostatic part _-_-_-_-_//
```

```
/* Splitting parameter */
double alpha;
/* box_size x */
double box_x;
/* box_size y */
double box_y;
/* box_size z */
double box_z;
/* Declaration of volume */
double V;
/* box_size z */
double Kmax;
/* box_size z */
double rcut_sq;
/* define self electrostat */
vector<double> self_elec;
/* define self electrostat */
vector<double> intra_elec;
//-_ Species present in the system (ions, DNA, gold nanoparticles) ---//
/* Number of atom in the DNA molecules */
int atom_DNA;
/* Number of dna */
int number_dna;
/* Number of atoms per dna */
int atom__per_dna;
/* Number of gold nanoparticle */
int gold_nano;
/* Number of free ions */
int free_ions;
/* Number of atom per gold nanoparticle */
int atom_per_gold_nano;
//_ PBC index for the cell list. --_-_-_-_-_-_
vector<vector<vector<vector<int>>>> PBC_cell;
int box_shift_PBC;
```

152
153
154

```
public:
/**
* @brief Constructor of the energy
* @details Constructor of the energy
* @param Topology file
*/
Energy ();
/**
* @brief Constructor of the energy
* @details Constructor of the energy
*
*@param Topology file
*/
Energy(string input_topol, string input_gro, int atom_DNA, int atom_per_dna,
int number_dna, int gold_nano, int atom_per_gold_nano, int free_ions);
/**
* @brief Constructor of the energy
* @details Constructor of the energy
* @param Topology file
*/
~Energy ();
/**
* @brief Function to read the topology file
* @details Function to read the topology file
* @param No parameter
*/
void ReadTopology();
/**
* @brief Function which test the interpenetration between atoms and cylinder
* @details Function which test the interpenetration between atoms and cylinder
* @param attribute: take the atom to be displaced.
*/
bool Test_Exclusion_volume();
```

206
207
208
209
210
211
212
213
214
215
216

```
* @brief Function which calculate the Real part of
electrostatics energy change after moving one atom
* @details Function which calculate the Real part of
electrostatics energy change after moving one atom
*
*@param attribute (selected atom)
*/
double Calculate__Real__Ewald__change();
/**
* @brief Function which calculate the Real part of electrostatics
* @details Function which calculate the Real part of electrostatics
*
*@param attribute (default)
*/
double Calculate__Real__Ewald__Init();
/**
* @brief Function which calculate the Fourier part of
electrostatics energy change after moving one atom
* @details Function which calculate the Fourier part of
electrostatics energy change after moving one atom
*
*@param attribute (selected atom)
*/
double Calculate__Im__Ewald_change();
/**
*@brief Function which calculate the Fourier part of electrostatics
* @details Function which calculate the Fourier part of electrostatics
* @param attribute (default)
*/
double Calculate__Im__Ewald__Init ();
/**
* @brief Function which calculate the Fourier part of electrostatics
* @details Function which calculate the Fourier part of electrostatics
*
*@param attribute (default)
*/
double Calculate_self_Ewald();
/**
*
```

```
* @brief Function which calculate the Fourier part of electrostatics
* @details Function which calculate the Fourier part of electrostatics
*
*@param attribute (default)
*/
double Calculate__intermolecular_Ewald ();
/**
*
*@brief Function to split a line into each component
* @details Function to split a line into each component
*-
* @param string and separating character
*/
vector<string> explode(const string& s, const char& c);
/**
*@brief Function that retrieve energy
* @details Function that retrieve energy
*
* @param no parameter
*/
double getEnergy();
/**
* @brief Function that Initialize the trajectory
* @brief Function that Initialize the trajectory
* @details Function that Initialize the trajectory
* @param no parameter
*/
void InitTrajFirstStep(vector <vector < double > > &input__pbc);
/**
*@brief Function that Initialize the trajectory
* @details Function that Initialize the trajectory
*@param no parameter
*/
void ChangeTraj(vector <vector < double > > &coordinates__atom__picked,
vector<int> &index__pbc_llist,
vector <vector <vector <vector <int>>>>> &particle__cell__without_chose_atom,
vector<vector<int> > &index_cell_chosen_particles_after);
/**
    *@brief Function that call energy calculation.
```

314
315 * @details Function that call energy calculation

```
*@details Function that call energy calculation
* @param no
*/
void Call__Calculate_Energy();
/**
* @brief Function that allow to update the coordinate in energy
    according to acceptance or rejection of MC step.
    *
    * @details Function that allow to update the coordinate in energy
    according to acceptance or rejection of MC step.
    *
    * @param boolean.
    */
    void AcceptMoveUpdate(bool IsAccepted, double E);
/**
* @brief Function that Initialize the box size
* @details Function that Initialize the box size
* @param no parameter
*/
void ReadBoxSize();
/**
* @brief Function that get the PBC for cell lists.
* @details Function that get the PBC for cell lists.
* @param no parameter
*/
void Set_PBC_cell_list(vector<vector <vector <vector <int>>>>>PBC_cell);
/**
* @brief Function that get the PBC for cell lists.
* @details Function that get the PBC for cell lists.
* @param no parameter
*/
```

```
    void Set__PBC__vector__shift(int box__shift__PBC);
    /**
    * @brief Function that give the minimum box size
    * @details Function that give the minimum box size
    * @param direction of the box
    */
    double min__element(double dir_x, double dir__y, double dir_z);
};
#endif
```


## A.3.6 Cell_lists.cpp

## Listing A. 12 - Cell_lists.cpp

frame
\#include "List.h"
/*
*

* Class to initialize cell lists
* 

*/
List: : List ()
\{
\}
List: : List (int atom_DNA, int gold_nano, int free__ions, int atom__per__gold__nano)
\{
this $\rightarrow$ atom_DNA $=$ atom_DNA;
this $\rightarrow$ gold_nano $\quad=$ gold__nano;
this $\rightarrow$ free__ions $\quad=$ free__ions;
this $\rightarrow$ atom__per_gold_nano $=$ atom__per_gold_nano;
particle_cell.resize (8, vector $<$ vector $<$ vector $<$ int $\ggg$
$(8$, vector $<$ vector $<$ int $\gg(8$, vector $<$ int $>(0,0))))$;
PBC_cell. resize (8, vector $<$ vector $<$ vector $<$ int $\ggg$
$(8$, vector $<$ vector $<$ int $\gg(8$, vector $<$ int $>(3,0)))$;
index_cell__per_particle.resize (atom_DNA + free__ions +
(gold__nano * atom__per__gold__nano), vector $<\operatorname{int}>(3,0))$;
$\}$
List: : ~List ()
\{\}

```
void List::Set_box(double box_x, double box_y, double box_z)
{
    this ->box_x = box_x;
    this }->\mathrm{ box_y = box_y;
    this ->box_z = box_z;
    this ->>cell_size_x = box_x/6;
    this }->\mathrm{ cell_size_y = box_y/6;
    this }->\mathrm{ cell_size__z = box_z/6;
    Initialization_PBC_list();
}
void List:: Initialization_cell_list(vector <vector < double> > &xyz)
{
    int i, j, k;
    // We insert the index of each ion in the cells. (direct cells)
    for(int index_atom = 0; index_atom < xyz.size(); index_atom++)
    {
        i = 1 + floor(xyz.at(index_atom)[0]/cell_size_x);
        j = 1 + floor(xyz.at(index_atom)[1]/ cell_size_y);
        k = 1 + floor(xyz.at(index_atom)[2]/cell_size_z )
        index_cell__per_particle.at(index_atom)[0] = i;
        index_cell_per_particle.at(index_atom)[1] = j;
        index_cell_per_particle.at(index_atom)[2] = k;
        particle_cell[i][j][k].push_back(index_atom);
    }
}
void List::Initialization_PBC_list()
{
    int pbc_x;
    int pbc_y;
    int pbc_z;
    for(int i = 0; i <= 7; i++)
    {
        for(int j = 0; j <= 7; j++)
        {
            for(int k = 0; k <= 7; k++)
            {
                //PBC for x direction.
                if(i= 0)
                {
                pbc_x = -1;
            }
            if(i>= 1 && i <= 6)
            {
                pbc_x = 0;
            }
```

88 89 90 91 92 93 94 95 96 97 98 99 100 101 102

```
    if \((\mathrm{i}=7)\)
```

    if \((\mathrm{i}=7)\)
    \{
    \{
        pbc_x \(=1\);
        pbc_x \(=1\);
    \}
    \}
    //PBC for y direction.
    //PBC for y direction.
    if \((\mathrm{j}=0)\)
    if \((\mathrm{j}=0)\)
    \{
    \{
    pbc_y \(=-1\);
    pbc_y \(=-1\);
    \}
    \}
    if \((\mathrm{j}>=1 \& \& \mathrm{j}<=6)\)
    if \((\mathrm{j}>=1 \& \& \mathrm{j}<=6)\)
    \{
    \{
    pbc_y \(=0\);
    pbc_y \(=0\);
    \}
    \}
    if \((\mathrm{j}=7)\)
    if \((\mathrm{j}=7)\)
    \{
    \{
    pbc_y \(=1\);
    pbc_y \(=1\);
    \}
    \}
    //PBC for z direction.
    //PBC for z direction.
    if \((\mathrm{k}=0)\)
    if \((\mathrm{k}=0)\)
    \{
    \{
    pbc_z \(=-1\);
    pbc_z \(=-1\);
    \}
    \}
    if \((\mathrm{k}>=1 \quad \& \& \mathrm{k}<=6)\)
    if \((\mathrm{k}>=1 \quad \& \& \mathrm{k}<=6)\)
    \{
    \{
    pbc_z \(=0\);
    pbc_z \(=0\);
    \}
    \}
    if \((\mathrm{k}=7)\)
    if \((\mathrm{k}=7)\)
    pbc_z \(=1\);
    pbc_z \(=1\);
    \}
    \}
    if \((\mathrm{i}>=1 \& \& \mathrm{i}<=6 \& \& \mathrm{j}>=1 \& \& \mathrm{j}<=6 \& \& \mathrm{k}>=1 \& \& \mathrm{k}<=6)\)
    if \((\mathrm{i}>=1 \& \& \mathrm{i}<=6 \& \& \mathrm{j}>=1 \& \& \mathrm{j}<=6 \& \& \mathrm{k}>=1 \& \& \mathrm{k}<=6)\)
    \{
    \{
        pbc_x \(=0 ;\)
        pbc_x \(=0 ;\)
        pbc_y \(=0\);
        pbc_y \(=0\);
        pbc_z \(=0\);
        pbc_z \(=0\);
            \}
            \}
        PBC_cell[i][j][k][0] \(=\) pbc_x;
        PBC_cell[i][j][k][0] \(=\) pbc_x;
        PBC_cell[i][j][k][1] \(=\) pbc_y;
        PBC_cell[i][j][k][1] \(=\) pbc_y;
        PBC_cell[i][j][k][2] \(\quad=\quad\) pbc_z;
        PBC_cell[i][j][k][2] \(\quad=\quad\) pbc_z;
        \}
        \}
    \}
    \}

```
```

142
143
144
145
146
147
148
149
150
151
152

```
vector<vector<vector<vector<int> > > > List::Get_PBC_list()
```

vector<vector<vector<vector<int> > > > List::Get_PBC_list()
{
{
return PBC_cell;
return PBC_cell;
}
}
vector<int> List::Get_cell_list_per_atom(int atom_index)
vector<int> List::Get_cell_list_per_atom(int atom_index)
{
{
return index_cell_per_particle[atom_index];
return index_cell_per_particle[atom_index];
}
}
vector<vector<vector<vector<int>>>>>\&
vector<vector<vector<vector<int>>>>>\&
List::Get_cell_list_without_chosen_atoms(vector<int> \&index__pbc_list)
List::Get_cell_list_without_chosen_atoms(vector<int> \&index__pbc_list)
{
{
particle_cell_buffer = particle_cell;
particle_cell_buffer = particle_cell;
int i, j, k;
int i, j, k;
int rank_atom_to_supress;
int rank_atom_to_supress;
for(int atom = 0; atom < index_pbc_list.size(); atom++)
for(int atom = 0; atom < index_pbc_list.size(); atom++)
{
{
i = index_cell_per_particle.at(index_pbc_list.at(atom))[0];
i = index_cell_per_particle.at(index_pbc_list.at(atom))[0];
j = index_cell_per_particle.at(index_pbc_list.at(atom))[1];
j = index_cell_per_particle.at(index_pbc_list.at(atom))[1];
k = index_cell_per_particle.at(index_pbc_list.at(atom))[2];
k = index_cell_per_particle.at(index_pbc_list.at(atom))[2];
for(int atom2 = 0; atom2 < particle_cell_buffer[i][j][k].size(); atom2++)
for(int atom2 = 0; atom2 < particle_cell_buffer[i][j][k].size(); atom2++)
{
{
if( index_pbc_list.at(atom) = particle_cell_buffer[i][j][k][atom2] )
if( index_pbc_list.at(atom) = particle_cell_buffer[i][j][k][atom2] )
{
{
rank_atom_to_supress = atom2;
rank_atom_to_supress = atom2;
// When we found the molecule in the list break the loop
// When we found the molecule in the list break the loop
break;
break;
}
}
}
}
// We withdraw the molecule that has been moved from the former
// We withdraw the molecule that has been moved from the former
// cell with index i,j,k
// cell with index i,j,k
particle_cell_buffer[i][j][k].erase (particle_cell_buffer[i][j][k].begin()
particle_cell_buffer[i][j][k].erase (particle_cell_buffer[i][j][k].begin()
+ rank_atom_to_supress);
+ rank_atom_to_supress);
}
}
return particle_cell_buffer;
return particle_cell_buffer;
}
}
void List::Update_cell_list(vector<vector<double> > \&coordinates_atom__picked,
void List::Update_cell_list(vector<vector<double> > \&coordinates_atom__picked,
vector<int> \&index_pbc_list)
vector<int> \&index_pbc_list)
{
{
int i, j, k;
int i, j, k;
int rank_atom_to_supress;

```
    int rank_atom_to_supress;
```

```
    for(int atom = 0; atom < index__pbc_list.size(); atom++)
    {
    /* We delete atom in former cell*/
    // We associate the former cell index if the current molecule
    // that has been moved.
    i = index__cell__per__particle.at(index__pbc__list.at (atom ) ) [0];
    j = index__cell__per__particle.at(index__pbc__list.at(atom))[1];
    k = index__cell__per__particle.at(index__pbc__list.at(atom ) ) [2];
    for(int atom2 = 0; atom2< particle__cell[i][j][k].size(); atom2++)
    {
        if( index__pbc_list.at (atom) = particle__cell[i][j][k][atom2] )
        {
            rank__atom__to__supress = atom2;
            // When we found the molecule in the list break the loop
            break;
        }
    }
    // We withdraw the molecule that has been moved from the former
    // cell with index i,j,k
    particle__cell[i][j][k].erase (particle__cell[i][j][k].begin()
    + rank__atom__to_supress);
    /* We associate the new cell to the index__cell__per__particle. */
    index__cell__per__particle.at(index_pbc__list.at(atom))[0] =
    1+ floor(coordinates__atom__picked.at(atom)[0]/cell__size__x );
    index__cell__per__particle.at(index__pbc__list.at(atom))[1] =
    1 + floor(coordinates__atom__picked.at(atom)[1]/cell__size__y);
    index__cell__per__particle.at(index_pbc__list.at(atom))[2] =
    1+ floor(coordinates__atom__picked.at(atom)[2]/cell__size__z);
    // New index cell of the corresponding particle
    i = index__cell__per__particle.at(index__pbc__list.at(atom))[0];
    j = index__cell__per__particle.at(index_pbc__list.at(atom))[1];
    k = index_cell_per__particle.at(index_pbc__list.at(atom))[2];
        // We associate the index of the moved particle with its new cell
        particle__cell[i][j][k].push__back(index__pbc__list.at(atom));
    }
}
vector<double> List::get__box__cell__list()
{
    vector<double> cell__size(3,0.0);
```

```
    cell__size[0] = cell__size__x;
    cell__size[1] = cell_size__y;
    cell__size[2] = cell__size__z;
    return cell_size;
}
int List:: get__box_shift__PBC()
{
    return box_x/cell__size__x;
}
// / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / /
/* Lists for the cluster displacements */
||||||||||||||||||||||||||||||||||
vector<int> List:: Update__lists__AuGP__ion(vector < vector < double > > &xyz,
double box_x, double box_y, double box_z, double pick__atom)
{
    double radius__in__ions__sq = 15*15;
    double RX = 0;
    double RY = 0;
    double RZ = 0;
    double x__gold;
    double y_gold;
    double z__gold;
    double r__sq;
    double x__ion;
    double y__ion;
    double z__ion;
    /*** Initialization of ion around gold AuGP ***/
    particle__ion__around__AuGP__list.clear ();
    x_gold = xyz.at(pick__atom)[0];
    y_gold = xyz.at(pick_atom)[1];
    z_gold = xyz.at(pick_atom)[2];
    for( int ions = atom_DNA; ions <= atom_DNA + free__ions - 1; ions ++ )
    {
        x_ion = xyz.at(ions)[0];
        y__ion = xyz.at(ions)[1];
        z_ion = xyz.at(ions)[2];
        RX = x__ion - x__gold;
        RY = y__ion - y__gold;
        RZ = z__ion - z__gold;
        // Other version for PBC condition testing //
```

```
    RX = RX - box_x * round(RX / box_x);
    RY = RY - box_y * round(RY / box_y);
    RZ = RZ - box_z * round(RZ / box_z);
        r_sq = RX*RX + RY*RY + RZ*RZ;
        if(r_sq < radius_in_ions_sq)
        {
        particle_ion_around_AuGP_list.push_back(ions);
        }
        else
        {
            continue;
        }
    }
    return particle_ion__around__AuGP_list;
}
int List::Count_ion_around_AuGP(vector<vector<double> > &xyz,
int index_gold, double box_x, double box_y, double box_z)
{
int count = 0;
double RX = 0
double RY = 0;
double RZ = 0;
double radius_in_sq = 15*15;
double x_gold = xyz.at(index_gold)[0];
double y_gold = xyz.at(index_gold)[1];
double z_gold = xyz.at(index_gold)[2];
double x ion;
double y_ion;
double z_ion;
double r_sq
for( int ions = atom_DNA; ions <= atom_DNA + free_ions - 1; ions ++ )
{
            x_ion = xyz.at(ions)[0];
            y_ion = xyz.at(ions)[1];
    z_ion = xyz.at(ions)[2];
    RX = x_ion - x_gold;
    RY = y_ion - y_gold;
    RZ = z_ion - z_gold;
    // Other version for PBC condition testing //
    RX = RX - box_x * round(RX / box_x);
```

```
            RY = RY - box_y * round(RY / box_y);
            RZ = RZ - box_z * round (RZ / box_z);
            r_sq = RX*RX + RY*RY + RZ*RZ;
            if(r_sq < radius_in_sq)
            {
            count++;
            }
            else
            {
                continue;
            }
    }
return count;
```

\}

## A.3.7 Cell_lists.h

Listing A. 13 - Cell_lists.h
frame
/*
*

* Initialisation of the cell lists - declaration
* 

*/
\#ifndef LIST_H
\#define LIST_H
\#include <vector>
\#include <iostream>
\#include <cmath>
using namespace std;
class List
\{
private:
// Initialization of the species present in the system
int atom_DNA;
int gold_nano;
int free__ions;
int atom_per_gold_nano;
double box_x;
double box_y;
double box_z;
double cell__size__x;
double cell__size_y

```
    double cell__size__z;
    // Vector for the verlet list
    vector<vector <vector <vector <int > > > > particle__cell;
    vector<vector<vector <vector <int>>>>> particle__cell__buffer;
    vector <vector <vector <vector <int > > > > PBC__cell;
    vector<vector<int>> index__cell__per__particle;
    // Vector for the AuGP-ion list
    vector<int> particle__ion__around__AuGP__list;
    public:
    List();
    ~List();
    List(int atom__DNA, int gold__nano, int free__ions, int atom__per__gold__nano);
    void Set_box(double box__x, double box__y, double box__z);
    void Initialization__cell__list(vector <vector < double > > &xyz);
    void Initialization__PBC__list();
    vector <vector <vector <vector <int > > > > 隹 __PBC_list ();
    vector<int> Get__cell_list__per__atom(int atom__index);
    vector <vector <vector <vector <int>>>>>&
    Get__cell__list__without__chosen__atoms(vector<int> &index__pbc__list );
    void Update__cell_list
    (vector <vector < double > > &coordinates__atom__picked, vector<int > &index__pbc__list);
    vector<double> get__box_ccell_list();
    int get__box__shift__PBC();
    // Functions for AuGPs cutoff with ions for clusters
    vector<int> Update_lists__AuGP__ion
    (vector<vector <double> > &xyz, double box_x, double box_y,
    double box__z, double pick__atom);
    int Count_ion__around__AuGP
    (vector <vector < double > > &xyz, int index_ggold, double box__x,
    double box_y, double box_z);
};
#endif
```


## A.3.8 MersenneTwister.h

Listing A. 14 - MersenneTwister.h
frame

```
/*
*
*
/*
```

* Mersenne Twister generator number

```
#ifndef _MERSENNETWISTER_H
```

\#define _MERSENNETWISTER_H
\#include <random>
$/ *!~ \ f i l e ~ M e r s e n n e T w i s t e r . h$
$\backslash$ brief A C ++11 implementation of a Mersenne-Twister
random number generator class.
*/
//! Mersenne-Twister class.
class MersenneTwister
\{
public:
//! Constructor.
MersenneTwister ()
\{
generator.seed (seed);
\}
//! Overloaded () operator.

double operator ()()
\{
return default_uniform_real_distribution(generator);
\}
//! Generate a random integer between min and max (inclusive).
/*! \param min
The minium of the range.
\param max
The maxium of the range.
$\backslash$ return
The uniform random integer.
*/
int integer (int min, int max)
\{
return std:: uniform_int_distribution<int $>\{\min , \max \}($ generator $) ;$
\}
//! Generate a random number from a normal distribution with
/*! zero mean and unit standard deviation.
$\backslash$ return
A random number drawn from the normal distribution.
*/
double normal()
\{
return default_normal_distribution(generator);
\}

```
//! Generate a random number from a normal distribution.
/*! \param mean
The mean of the the normal distribution.
\param stdDev
The standard deviation of the normal distribution.
\return
A random number drawn from the normal distribution.
*/
double normal(double mean, double stdDev)
{
return std:: normal__distribution<double >{mean, stdDev}(generator);
}
//! Get the random number generator seed.
/*! \return seed
The generator seed.
*/
unsigned int getSeed()
{
return seed;
}
//! Seed the random number generator.
/*! \param seed__
The new seed.
*/
void setSeed(unsigned int seed__)
{
seed = seed__;
generator.seed (seed);
}
private:
/// The Mersenne-Twister generator.
std::mt19937 generator ;
/// Default uniform__real distribution [0-1].
std::uniform_real__distribution<double> default__uniform__real__distribution {0.0, 1.0};
/// Default normal distribution with zero mean and unit standard deviation.
std:: normal__distribution<double> default__normal__distribution {0.0, 1.0};
/// The random number seed. We keep the same seed to check our code.
unsigned int seed = 1235;
};
#endif /* _MERSENNETWISTER H */
#endif
```


## A.3.9 Makefile

Listing A. 15 - Makefile

```
    frame
exe: main.o system_MC.o Energy.o List.o
g++ -o exe main.o system__MC.o Energy.o List.o -O3
-L/home/ambroise/gsl/lib -lgsl -lgslcblas -lm
-static-libstdc++
main.o: main.cpp
g++ -o main.o -c main.cpp -std=c++11 -I/home/ambroise/gsl/include
system_MC.o: system__MC.cpp
g++ -o system__MC.o -c system__MC.cpp -std=c++11 -I/home/ambroise/gsl/include
Energy.o: Energy.cpp
g++ -o Energy.o -c Energy.cpp -std=c++11
List.o: List.cpp
g++ -o List.o -c List.cpp
clean:
rm -rf *.o
```


## A. 4 Program to visualize a trajectory

The program to visualize a trajectory is 'Visualization_trajectory.cpp' with the corresponding header file 'Visualization_trajectory.h'. It requires also an adapted 'Main.cpp' and a tool to read a trajectory: 'Read_trajectory.cpp' and 'Read_trajectory.h'.

## A.4.1 Main.cpp

Listing A. 16 - Main.cpp

```
        frame
#include <SDL/SDL.h>
#include <GL/gl.h>
#include <GL/glu.h>
#include <cstdlib>
#include "camera.h"
#include "read_trajectory.h"
int main(int argc, char * argv[])
{
    // Read Trajectory
    ReadTrajectory traj("../traj_output_1.txt","system_connectivity.txt");
    int number_frame = traj.getNumberFrame();
    int number_atoms = traj.getNumberAtom();
    double x = traj.getBoxSize('x');
```

```
double y = traj.getBoxSize('y');
double z = traj.getBoxSize(' z');
// We initialize these vectors
vector <vector <double>> xyz = traj.ReadStep();
vector <int> type_Atoms = traj.gettypeAtoms();;
vector <string> name_Residue = traj.getNameResidue();
vector <vector<int>> connectivity = traj.getConnectivity();
// Initialisation Camera
SDL_Init(SDL_INIT_VIDEO); // Use SDL interface
atexit(SDL_Quit); // return SDL
SDL_Event event; //declaration struct event
int time_delay = 100;
bool pause = false;
CameraView * camera;
camera = new CameraView(x, y, z);
SDL_WM_SetCaption("Vizualisation_trajectory", NULL); // name of the window
SDL_SetVideoMode(800,800,32,SDL_OPENGL);
glClearColor (1.0,1.0,1.0,1.0);
glMatrixMode( GL_PROJECTION );
glLoadIdentity();
gluPerspective(70,(double)(800/800),1.0,10*(x));
glEnable(GL_DEPTH_TEST);
bool continuation = 1;
int frame_count = 1;
SDL_EnableKeyRepeat(10, 10);
float move = 0.0;
while(continuation)
{
    SDL_PollEvent(&event);
    switch(event.type)
    {
    case SDL_QUIT:
    exit(0);
    break;
            case SDL_KEYDOWN:
            switch (event.key.keysym.sym)
            {
                case SDLK_h:
```

```
            camera->Reset_Scene_View(event.key);
            break;
            case SDLK a:
            camera->Zoom_Scene_Plus();
            break;
            case SDLK_z:
            camera->Zoom_Scene_Minus();
            break;
                case SDLK_r:
                frame_count = 1;
                traj.Reset();
                break;
                case SDLK_p:
                pause = true;
                break;
            case SDLK_m:
            pause = false;
            break;
            case SDLK_v:
            for(int i = 0; i < 100; i++)
            {
            xyz = traj.ReadStep()
            }
            frame_count = frame_count + 100;
            break;
                case SDLK ESCAPE:
                exit(0);
                default:
                camera->Move_Scene(event.key);
    }
}
camera->DrawGL(connectivity, xyz, type_Atoms, name_Residue, number_atoms);
move += 0.01;
if(pause = false)
{
            if(frame_count <= number_frame)
            {
                xyz = traj.ReadStep();
                SDL_Delay (1);
                cout << frame_count << endl;
                frame_count++;
            }
}
else
{
```

```
                SDL__Delay (1);
            }
    }
    delete camera;
    SDL__Quit ();
    return 0;
}
```


## A.4.2 Visualization_trajectory.cpp

Listing A. 17 - Visualization_trajectory.cpp
frame
\#include <iostream>
\#include <SDL/SDL.h>
\#include <GL/gl.h>
\#include <GL/glu.h>
\#include "camera.h"
CameraView: : CameraView(double x, double y, double z)
\{
this $\rightarrow$ _x_box $=x$;
this $\rightarrow$ _y_box $=y$;
this $\rightarrow$ _z_zox $=\mathrm{z}$;
this $\rightarrow$ _distance $=1.5 * x$;
this $\rightarrow$ _angleY $=0.0$;
this $\rightarrow$ _angleZ $=0.0$;
\}
CameraView : :~CameraView ()
\{
\}
void CameraView:: Move_Scene(const SDL_KeyboardEvent \& event)
\{
if ((event.type = SDL_KEYDOWN)\&\&(event.keysym.sym $=$ SDLK_LEFT) $)$
\{
_angleZ += 1;
\}
else \{
if ((event.type $=$ SDL_KEYDOWN)\&\&(event.keysym.sym $=$ SDLK_RIGHT) )
\{
__angleZ -= 1;
\}
else\{
if $(($ event.type $=$ SDL_KEYDOWN $) \& \&($ event. keysym.sym $=$ SDLK_UP $))$
\{
__angleY $+=1$;
\}
else\{
if $(($ event.type $=$ SDL_KEYDOWN $) \& \&($ event.keysym.sym $=$ SDLK_DOWN $))$

```
                {
                    _angleY -= 1;
                }
                }
        }
    }
    if (__angleY > 90)
        _angleY = 90;
    else if (__angleY < -90)
        _angleY = -90;
}
void CameraView::Reset_Scene_View(const SDL_KeyboardEvent & event)
{
    _distance = 1.5*(_x_box);
    _angleY = 0.0;
    _angleZ = 0.0;
}
void CameraView::Zoom_Scene_Plus()
{
    _distance += 0.1*(_x_box);
    if(_distance > 3*(_x_box))
        _distance = 3*(_x_box);
}
void CameraView::Zoom_Scene_Minus()
{
    _distance -= 0.1*(_x_box);
    if(__distance < 0.5*(_x_box))
        _distance = 0.5*(_x_box);
}
void CameraView::Look()
{
        gluLookAt(__distance, 0,0,0,0,0,0,0,1);
        glRotated(__angleY,0,1,0);
        glRotated(__angleZ , 0,0,1);
}
void CameraView::D DrawGL(vector<vector<int>> &connectivity,
vector<vector<double> > &xyz, vector<int> &atom_types,
vector<string> &res_name, int number_atoms)
{
    glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
    glMatrixMode ( GL_MODELVIEW );
    glLoadIdentity();
    Look ();
```

```
// draw the box
```

```
glLineWidth(2.0);
```

glBegin (GL_LINE_LOOP) ;
glColor3ub ( $0,0,0$ );
glVertex3d(-(_x_box) $/ 2, \quad\left(\_y \_b o x\right) / 2, \quad(\quad$ z_box $\left.) / 2\right)$;
glVertex3d $\left(-\left(\_x \_b o x\right) / 2, \quad\left(\_y \_b o x\right) / 2,-\left(\_\right.\right.$z_box $\left.) / 2\right)$;
glVertex3d $\left(-\left(\_x \_b o x\right) / 2,-\left(\_y \_b o x\right) / 2,-\left(\_\right.\right.$z_box $\left.) / 2\right)$;
glVertex $3 \mathrm{~d}\left(-\left(\_x \_b o x\right) / 2,-\left(\_y \_b o x\right) / 2, \quad\left(\_\right.\right.$z_box $\left.) / 2\right)$;
glEnd () ;
glBegin(GL_LINE_LOOP) ;
glColor3ub ( $0,0,0$ );
glVertex3d ((_x_box)/2, (_y_box)/2, (_z_box)/2);
glVertex3d $\left(\left(\ldots x \_b o x\right) / 2, \quad\left(\_y \_b o x\right) / 2,-\left(\_\right.\right.$z_box $\left.) / 2\right)$;
glVertex3d $\left(\left(\ldots x \_b o x\right) / 2,-\left(\_y \_b o x\right) / 2,-\left(\_\right.\right.$z_box $\left.) / 2\right)$;
glVertex3d ((_x_box) $/ 2, \quad-\left(\_y \_b o x\right) / 2,\left(\_\right.$z_box $\left.) / 2\right)$;
glEnd () ;
glBegin(GL_LINE_LOOP) ;
glColor3ub ( $0,0,0$ );
glVertex3d( (_x_box)/2, -(_y_box)/2, (_z_box)/2);
glVertex3d( (_x_box)/2, -(_y_box)/2,-(_z_box)/2);
glVertex3d(-(_x_box) $/ 2,-\left(\_y \_b o x\right) / 2,-\left(\_\right.$z_box) $\left./ 2\right)$;
glVertex3d(-(_x_box)/2, -(_y_box)/2, (_z_box)/2);
glEnd () ;
glBegin(GL_LINE_LOOP);
glColor3ub ( $0,0,0$ );
glVertex3d( (_x_box) $/ 2, \quad\left(\quad y \_b o x\right) / 2, \quad\left(\_\right.$z_box) $\left./ 2\right)$;
glVertex3d( (_x_box) $/ 2, \quad\left(\_y \_b o x\right) / 2,-\left(\_\right.$z_box $\left.) / 2\right)$;
glVertex3d $\left(-\left(\_x \_b o x\right) / 2, \quad\left(\_y \_b o x\right) / 2,-\left(\_\right.\right.$z_box $\left.) / 2\right)$;
glVertex3d $\left(-\left(\_x \_b o x\right) / 2, \quad\left(\_y \_b o x\right) / 2, \quad\left(\_\right.\right.$z_box) $\left./ 2\right)$;
glEnd () ;
glBegin (GL_LINE_LOOP) ;
glColor3ub ( $0,0,0$ );
glVertex3d( (_x_box)/2, (_y_box)/2, -(_z_box)/2);
glVertex3d ( (_x_box)/2, -(_y_box)/2, -(_z_box)/2);
glVertex3d(-(_x_box)/2,-(_y_box)/2, -(_z_box)/2);
glVertex3d(-(_x_box)/2, (_y_box)/2, -(_z_box)/2);
glEnd () ;
glBegin (GL_LINE_LOOP) ;
glColor3ub ( $0,0,0$ );
glVertex3d( (_x_box)/2, (_y_box)/2, (_z_box)/2);
glVertex3d ( (_x_box)/2,-(_y_box)/2, (_z_box)/2);
glVertex $3 \mathrm{~d}\left(-\left(-\mathrm{x}_{\mathrm{x}}\right.\right.$ box $) / 2,-\left(-\mathrm{y} \_\right.$box $) / 2, \quad\left(\_\right.$z_box $\left.) / 2\right)$;
glVertex3d $\left(-\left(\_x \_b o x\right) / 2, \quad\left(\_y \_b o x\right) / 2, \quad\left(\_\right.\right.$z_box $\left.) / 2\right) ;$
glEnd () ;
// Draw the system at a given frame

```
    for ( int atom_count \(=0\); atom_count \(<\) number_atoms; atom_count++ )
    \{
        glPushMatrix ();
        GLUquadricObj *quadric;
        Color_Assignment (atom_types [atom_count]) ;
        glTranslatef(xyz[atom_count][0] -(_x_box)/2,
        xyz[atom_count][1]-(_y_box)/2, xyz[atom_count][2]-(_z_box)/2);
        quadric \(=\) gluNewQuadric ();
        gluQuadricDrawStyle(quadric, GLU_FILL);
        if (res_name[atom_count] = "DNA1" || res_name[atom_count] = "DNA2" ||
        res_name[atom_count] = "DNA3" || res_name[atom_count] = "DNA4")
        \{
        if (atom_count \(\% 5=0)\)
        \{
            gluSphere( quadric , 3.9 , 20 , 20 );
        \}
        else
        \{
            gluSphere( quadric , \(2.1,20,20\) );
        \}
        \}
        else
        \{
            if (res_name[atom_count].substr \((0,4)=\) "ionQ" \()\)
            \{
            gluSphere( quadric , 4 , 20 , 20 );
            \}
            else
            \{
            if (res name[atom count]. substr \((0,4)=\) "AUGP" \&\&
            (atom_count-600) \% \(25=0\) )
            \{
                gluSphere( quadric , 7 , 20 , 20 );
            \}
            else
            \{
                gluSphere( quadric , \(1.5,20,20\) );
            \}
            \}
        \}
        gluDeleteQuadric(quadric)
        glPopMatrix ()
    \}
    glFlush();
    SDL_GL_SwapBuffers () ;
void CameraView:: Color_Assignment(int atom_types) const
```

\}

```
{
    if(atom_types = 6) glColor4f(0.0,0.0,0.0,1); // Black
    if(atom_types = 8) glColor4f(1.0,0.0,0.0,1); // Red
    if(atom_types = 20) glColor4f(0.0,0.0,1.0,1); // Blue
    if(atom_types=7) glColor4f(0.90,0.90,0.90,1); // Grey
    if(atom_types = 16) glColor4f(1.0,1.0,0.0,1); // Yellow
    if(atom_types = 12) glColor4f(1.0,0.0,1.0,1); // Purple
    if(atom_types = 79) glColor4f(1.0,0.5,0.0,1); // Orange
    if(atom_types = 1) glColor4f (0.0,1.0,0.0,1); // Green
    if(atom_types = 21) glColor4f(0.658824,0.658824,0.658824,1); // light grey
}
```


## A.4.3 Visualization_trajectory.h

Listing A. 18 - Visualization_trajectory.h frame
\#ifndef CAMERAVIEW_H \#define CAMERAVIEW_H
\#include <iostream>
\#include <SDL/SDL.h>
\#include <GL/gl.h>
\#include <GL/glu.h>
\#include <vector>
using namespace std;
class CameraView
\{
private:
double _x_box;
double _y_box;
double _z_box;
double _distance;
double _angleY;
double __angleZ;
public:
CameraView (double $x$, double $y$, double $z$ );
$\sim$ CameraView ();
void Move_Scene(const SDL_KeyboardEvent \& event);
void Reset_Scene_View (const SDL_KeyboardEvent \& event);
void Zoom_Scene_Plus ();
void Zoom_Scene_Minus();
void Look();
void DrawGL(vector<vector<int>> \&connectivity,
vector<vector<double>> \&xyz, vector<int> \&atom_types,
vector $<$ string $>$ \&res_name, int number_atoms);
void Color_Assignment (int atom_types) const;
\};
\#endif

## A.4.4 Read_trajectory.cpp

Listing A. 19 - Read_trajectory.cpp

```
    frame
#include <iostream>
#include <string>
#include <iomanip>
#include "read_trajectory.h"
using namespace std;
ReadTrajectory::ReadTrajectory()
{
}
ReadTrajectory::ReadTrajectory(string trajfilename,
string connectivity_file) : iFS(trajfilename.c_str())
{
    this }->\mathrm{ _traj_file = trajfilename;
    this }->\mathrm{ __connect_file = connectivity_file;
    ReadHeaderTraj();
    ReadConnectivity();
}
ReadTrajectory::~ReadTrajectory()
{
    iFS.close();
}
void ReadTrajectory::ReadHeaderTraj()
{
    string header = " ";
    string box = " ";
    // We read in the file (header).
    iFS >> header;
    iFS >> _number_frame >> _number_atoms;
    iFS >> box >> _box_size_x >> _box_size_y >> _box_size_z;
    // We precise the size of each tab.
    this }->>_traj.resize(_number_atoms, vector<double> (3,0.0))
    this >>_type_atoms.resize(__number_atoms,0);
    this ->_residue_names.resize(__number_atoms, " " );
}
void ReadTrajectory:: ReadConnectivity()
{
    ifstream iFS;
    iFS.open(_connect_file.c_str());
    vector<int> connectivity_buffer;
    vector <string> read_line;
    string line;
```

```
    if(iFS)
    {
        while (getline(iFS,line))
        {
                read_line = explode(line, '\t');
                for( int i = 0; i < read_line.size(); i++)
                {
                    connectivity_buffer.push__back(stoi(read_line.at(i)));
                }
                _connectivity.push_back(connectivity_buffer);
                read_line.clear();
                connectivity_buffer.clear();
        }
    }
    else
    {
        cout << " cannot read " << __connect_file << endl;
    }
}
double ReadTrajectory::getBoxSize(char direction) const
{
    double dir = 0;
    switch(direction)
    {
        case 'x':
        dir = _box_size_x;
        break;
            case 'y':
            dir = _box_size_y;
            break;
            case 'z':
            dir = _box__size_z;
            break;
    }
    return dir;
}
vector <int> ReadTrajectory::gettypeAtoms()
{
    return _type_atoms;
}
vector<vector<double> > ReadTrajectory::ReadStep()
{
    string arobase = "";
    iFS >> arobase;
    for( int j = 0; j < __number_atoms; j++)
```

```
    {
                iFS >> _type_atoms.at(j) >> __traj.at(j)[0] >> _traj.at(j)[1] >>
                __traj.at(j)[2] >> __residue__names.at(j );
    }
    return __traj;
}
vector <string> ReadTrajectory::getNameResidue()
{
    return _residue__names;
}
int ReadTrajectory::getNumberFrame() const
{
    return __number_frame;
}
int ReadTrajectory::getNumberAtom() const
{
    return __number_atoms;
}
vector<vector<int>> ReadTrajectory::getConnectivity ()
{
        return _connectivity;
}
void ReadTrajectory:: Reset()
{
        iFS.close();
        iFS.clear ();
        iFS.open(__traj__file.c__str());
        ReadHeaderTraj();
}
vector<string> ReadTrajectory: explode(const string& s, const char& c)
{
    string buff{" " };
    vector < string> v;
    for(auto n:s)
    {
            if(n != c) buff+=n; else
            if(n=c&& buff != "") { v.push_back(buff); buff= " "; }
    }
    if(buff != " ") v.push__back(buff);
    return v;
}
```


## A.4.5 Read_trajectory.h

Listing A. 20 - Read_trajectory.h

```
    frame
#include <fstream>
#include <string>
#include <vector>
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <iomanip>
#include <sstream>
#include <stdexcept>
#include <fstream>
#ifndef READTRAJ_H
#define READTRAJ_H
using namespace std;
class ReadTrajectory{
    private:
    ifstream iFS;
    vector <vector<double>> _traj;
    vector <vector<int>> __connectivity;
    vector <int> _type__atoms;
    vector <string > __residue__names;
    double __number_frame;
    double __number__atoms;
    double __box__size__x;
    double __box__size__y;
    double __box_size__z;
    string __traj_file;
    string __connect_file;
    public:
    ReadTrajectory();
    ReadTrajectory(string trajfilename, string connectivity_file);
    ~ReadTrajectory ();
    void ReadHeaderTraj();
    void ReadConnectivity();
    double getBoxSize(char direction) const;
    vector <int> gettypeAtoms();
    vector <vector <double > > ReadStep ();
    vector <string> getNameResidue();
    int getNumberFrame() const;
    int getNumberAtom() const;
    vector <vector <int>> getConnectivity ();
    void Reset();
    vector<string> explode(const string& s, const char& c);
};
#endif //READTRAJ_H
```


## A.4.6 Makefile

Listing A. 21 - Makefile

```
    frame
CFLAGS = -O3
HDFLAGS = - I/usr/include
CPPFLAG = -std=c}++1
#LDFLAGS = -L./usr/lib/x86_64-linux -gnu -lSDL $(HDFLAGS)
exe: main.o camera.o read_trajectory.o
# g++ -o exe main.o $(CFLAGS) $(LDFLAGS) - Wall
g++ -o exe main.o camera.o read_trajectory.o -lSDL -lGL -lGLU
-ldrm -lX11 -lXext -lXmu
main.o: main.cpp camera.h
g++ -o main.o -c main.cpp
camera.o: camera.cpp
g++ -o camera.o -c camera.cpp
read__trajectory.o: read__trajectory.cpp
g++ -o read__trajectory.o -c read__trajectory.cpp -std=c++11
clean:
rm -rf *.o
```


## A. 5 Program to analyze a trajectory

The program to analyze a trajectory is 'Analysis_trajectory.cpp' with the corresponding header file 'Analysis_trajectory.h'. It requires also an adapted 'Main.cpp'. The same tool to read a trajectory is used to perform an analysis.

## A.5.1 Main.cpp

$$
\text { Listing A. } 22 \text { - Main.cpp }
$$

frame

```
#### Inut file to the main file: 'conf.in' ####
../../../MC_1/traj__output__1.txt
../../../building_topology__file/system__topol.txt
Pressure__1__electrostat__step.txt
Pressure__1__electrostat__total.txt
#############################################################
########################################################
#include "read__trajectory.h"
#include "Analysis.h"
```

```
#include <chrono>
#include <thread>
int main(int argc, char * argv[])
{
    if(argc != 2)
{
    cout << "Usage: " << endl;
    cout << argv[0] <<" configfile." << endl;
    return -1;
}
ifstream iFS;
string configfile;
ofstream oFS;
configfile = argv[1];
// Arguments of the configfile.
string traj_input;
string input_topol;
string Force_step;
string Force_total;
// Read the configfile
cout << endl;
cout << "***********************************" << endl;
cout <<"Reading " << configfile << "... **"<<endl;
iFS.open(configfile.c_str());
iFS >> traj_input;
iFS >> input_topol;
iFS >> Force_step;
iFS >> Force_total;
cout <<"... Ok. **" << endl;
cout << "**********************************" << endl;
// Read Trajectory
ReadTrajectory traj(traj_input);
int number_frame = traj.getNumberFrame();
int number_atoms = traj.getNumberAtom();
double box_x = traj.getBoxSize('x');
double box_y = traj.getBoxSize('y');
double box_z = traj.getBoxSize('z');
// We initialize these vectors
vector <vector<double> > xyz = traj.ReadStep();
int MC_step = 10000;
int step_normalization = 0;
// Parameter for the system.
```

```
int atom_DNA =
int gold_nano = 40;
int free_ions = 0;
int atom_per_gold_nano = 25;
int Qions = 0;
int cation = 0;
int anion = 0;
// Analysis tool for system with 50 Aunp (charge=6).
Analysis analysis(MC_step, atom_DNA, gold_nano,
free_ions, atom_per_gold_nano, Qions, cation, anion,
box_x, box_y, box_z, input_topol);
analysis.gather_Pressure_DNA_DNA(xyz, step_normalization);
analysis.Output_each_step_init(Force_step);
for (int step = 0 ; step < MC_step; step++)
{
    if(step >= 0.1*MC_step)
    {
            analysis.gather_Pressure_DNA_ions(xyz, step_normalization);
            analysis.gather_Pressure_ions_ions(xyz, step_normalization);
            analysis.Output_each_step(step_normalization, Force_step);
            xyz = traj.ReadStep();
            step_normalization++;
        }
        else
        {
            xyz = traj.ReadStep();
        }
    }
// Finalize Force calculation DNA-DNA interaction
analysis.gather_Pressure_with_units();
analysis.Average_Pressure();
//Output of forces
analysis.Output_Pressure(Force_total);
return 0;
}
```


## A.5.2 Analysis_trajectory.cpp

Listing A. 23 - Analysis_trajectory.cpp
frame
\#include <iostream>
\#include <string>
\#include <fstream>
\#include <iomanip>

```
#include <algorithm>
#include "Analysis.h"
using namespace std;
Analysis::Analysis()
{}
Analysis::~Analysis()
{}
Analysis::Analysis(int total_MC_step, int atom_DNA, int gold__nano,
int free_ions, int atom_per_gold_nano, int Qion, int cation, int anion,
double box_x, double box_y, double box_z, string topolfile)
{
    // Let's read the topology.
    ReadTopology();
    Pressure_total.resize(0.9*total_MC_step,0.0);
    Pressure_DNA_ions.resize(0.9*total_MC_step,0.0);
    Pressure_ions_ions.resize(0.9*total_MC_step,0.0);
    Pressure__average = 0;
    Pressure_standard_deviation = 0;
}
void Analysis::gather_Pressure_DNA_DNA(vector<vector<double>> xyz,int step)
{
    // Calculation of the force create by one DNA on the other.
    // (phosphate on phosphate)
    double RX;
    double RY;
    double RZ;
    double r_sq;
    double r;
    double V = box_x*box_y*box_z;
    double RL,RM,RN,KK,KSQ,AK;
    int n, l,m;
    int K;
    double somme;
    int count_zero;
    double cos_m1,cos_m2,cos_m3,sin_m1,sin_m2,sin_m3;
    double term1,term2;
```

```
vector <double> Force_DNA_DNA_buffer(3,0.0);
// Sum of the phosphates of DNA1
for(int i = 0; i < atom_DNA - 1; i++)
{
        // Sum of the phosphates of DNA2
        for(int j = i + 1; j < atom_DNA; j++)
    {
                if(charge.at(i)=0 || charge.at(j)=0)
                {
            continue;
        }
        else
        {
            Force_DNA_DNA_buffer [0] = 0;
            Force_DNA_DNA_buffer [1] = 0;
            Force_DNA_DNA_buffer [2] = 0;
                // Difference of distance
                RX= xyz.at(i)[0] - xyz.at(j)[0];
                RY = xyz.at(i)[1] - xyz.at(j)[1];
                RZ = xyz.at(i)[2] - xyz.at(j)[2];
                // Other version for PBC condition testing.
                RX = RX - box_x * round (RX / box_x);
                RY = RY - box_y * round (RY / box_y);
                RZ = RZ - box_z * round(RZ / box_z );
                r_sq = RX*RX + RY*RY + RZ*RZ;
                if( r_sq > rcut_sqq)
                {
            }
            else
            {
            r = sqrt(r_sq);
            // Real part of the force calculated on each phosphate of DNA1.
            term1 = charge.at(i) * charge.at(j)*
            (2*alpha/sqrt(PI)*exp(-pow(alpha, 2)*r_sq) + erfc(alpha*r)/r );
            Force_DNA_DNA_buffer[0] = term1 * (RX/r_sq);
            Force_DNA_DNA_buffer[1] = term1 * (RY/r_sq);
            Force_DNA_DNA_buffer[2] = term1 * (RZ/r_sq);
            }
                for ( l = 0; l <= Kmax; l++ )
                    {
```

$$
\begin{aligned}
& \mathrm{RL}=\text { double }(\mathrm{l}) / \text { box_x; } \\
& \text { for }(\mathrm{m}=0 ; \mathrm{m}<=\text { Kmax } ; \mathrm{m}++) \\
& \{\quad \mathrm{RM}=\text { double }(\mathrm{m}) / \text { box_y; }
\end{aligned}
$$

$$
\text { for }(\mathrm{n}=0 ; \mathrm{n}<=\text { Kmax } ; \mathrm{n}++)
$$

\{

$$
\mathrm{RN}=\text { double }(\mathrm{n}) / \text { box_z; }
$$

$$
\text { count_zero }=0
$$

$$
\text { if }(1=0) \text { count_zero++; }
$$

$$
\text { if }(\mathrm{m}=0) \text { count_zero++; }
$$

$$
\text { if }(\mathrm{n}=0) \text { count_zero++; }
$$

$$
\text { // test of magnitude of } K \text { vector }
$$

$$
\mathrm{KK}=1 * \mathrm{l}+\mathrm{m} * \mathrm{~m}+\mathrm{n} * \mathrm{n} \text {; }
$$

$$
\text { if }(\mathrm{KK}>=\operatorname{Kmax} * \operatorname{Kmax}+2)
$$

                \{
                    continue;
            \}
                if \((\mathrm{l}=0\) \& \(\mathrm{m}=0 \& \& \mathrm{n}=0)\)
                \{
                continue;
                \}
                cos_m1 \(=\cos (2 * \mathrm{PI} * \mathrm{RL} * \mathrm{RX}) ;\)
                \(\cos \_\mathrm{m} 2=\cos (2 * \mathrm{PI} * \mathrm{RM} * \mathrm{RY}) ;\)
                cos_m3 \(=\cos (2 * \mathrm{PI} * \mathrm{RN} * \mathrm{RZ})\);
                \(\sin \_\mathrm{m} 1=\sin (2 * \mathrm{PI} * \mathrm{RL} * \mathrm{RX}) ;\)
                \(\sin \_\mathrm{m} 2=\sin (2 * \mathrm{PI} * \mathrm{RM} * \mathrm{RY}) ;\)
                \(\sin \_\mathrm{m} 3=\sin (2 * \mathrm{PI} * \mathrm{RN} * \mathrm{RZ}) ;\)
                \(\mathrm{KSQ}=\mathrm{RL} * \mathrm{RL}+\mathrm{RM} * \mathrm{RM}+\mathrm{RN} * \mathrm{RN} ;\)
                \(\mathrm{AK}=(2 / \mathrm{V}) * \exp (-(\mathrm{PI} * \mathrm{PI} * \mathrm{KSQ}) /(\) alpha \(*\) alpha \()) / \mathrm{KSQ}\);
                term \(2=A K *\) charge.at (i) * charge.at (j) * pow(2,3-count_zero);
                    Force_DNA_DNA_buffer [0] += term \(2 * \sin \_m 1 * \cos \_m 2 * \cos \_m 3 * R L\);
            Force_DNA_DNA_buffer [1] += term \(2 * \cos \_m 1 * \sin \_m 2 * \cos \_m 3 * R M\);
            Force_DNA_DNA_buffer [2] \(+=\) term \(2 * \cos \_\mathrm{m} 1 * \cos \_\mathrm{m} 2 * \sin \_\mathrm{m} 3 * R N\);
                \}
                \}
            \}
                pressure_DNA_DNA += Force_DNA_DNA_buffer [0] * RX +
                Force_DNA_DNA_buffer [1] * RY +
                Force_DNA_DNA_buffer [2] * RZ;
                \}
        \}
    \}
    \}

167 168 169 170 171 172 173 174
void Analysis: : gather_Pressure_DNA_ions (vector $<$ vector $<$ double $\gg$ xyz, int step)
\{
// Calculation of the force create by one DNA on the other.
// (phosphate on phosphate)
double RX;
double RY;
double RZ;
double r__sq;
double r;
double $V=$ box_x*box_y*box_z;
double RL,RM,RN,KK,KSQ,AK;
int $\mathrm{n}, \mathrm{l}, \mathrm{m}$;
int $K$;
double somme;
int count_zero;
double cos_m1, cos_m2, cos_m3, sin_m1, sin_m2, sin_m3;
double term1,term2;
vector <double> Force__DNA__ions__buffer (3, 0.0);
// Sum of the phosphates of DNA1
for (int $\mathrm{i}=0 ; \mathrm{i}<$ atom_DNA; $\mathrm{i}++$ )
\{
// Sum of the phosphates of DNA2
for (int $j=$ atom_DNA; $j<x y z . \operatorname{size}() ; j++)$
\{
if (charge.at $(\mathrm{i})=0 \quad \|$ charge.at $(j)=0)$
\{
continue;
\}
else
\{
Force_DNA_ions_buffer [0] $=0$;
Force__DNA_ions_buffer [1] $=0$;
Force__DNA__ions__buffer [2] $=0$;
// Difference of distance.
$R X=\quad x y z \cdot a t(i)[0]-x y z \cdot a t(j)[0] ;$
$R Y=\quad x y z . a t(i)[1]-x y z \cdot a t(j)[1] ;$
$R Z=x y z \cdot a t(i)[2]-x y z \cdot a t(j)[2] ;$

221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274

```
// Other version for PBC condition testing.
RX = RX - box_x * round (RX / box_x);
RY = RY - box_y * round (RY / box_y);
RZ = RZ - box_z * round(RZ / box_z);
r_sq = RX*RX + RY*RY + RZ*RZ;
if( r_sq > rcut_sq)
{
}
else
{
    r = sqrt(r_sq);
    // Real part of the force calculated on each phosphate of DNA1.
    term1 = charge.at(i) * charge.at(j)*
    (2*alpha/sqrt(PI)*exp(-pow(alpha,2)*r_sq) + erfc(alpha*r)/r );
    Force_DNA_ions_buffer [0] = term1 * (RX/r_sq);
    Force_DNA_ions_buffer[1] = term1 * (RY/r_sq);
    Force_DNA_ions_buffer[2] = term1 * (RZ/r_sq);
}
for ( l = 0; l <= Kmax; l++ )
{
    RL = double(l)/box_x;
    for ( m = 0; m <= Kmax; m++ )
        {
            RM = double(m)/box_y;
            for ( n = 0; n <= Kmax; n++ )
            {
                RN = double(n)/box_z;
                    count_zero = 0;
                    if(l= 0) count_zero++;
                            if (m=0) count_zero++;
                    if(n=0) count_zero++;
                    // test of magnitude of K vector
                KK = l *l + m*m + n*n;
                    if( KK >= Kmax*Kmax+2)
                {
                                    continue;
                }
                    if( l = 0 && m=0 && n=0)
                    {
                        continue;
            }
```

```
                    cos_m1 = cos(2*PI*RL*RX);
                    cos_m2 = cos(2*PI*RM*RY);
                    cos_m3 = cos(2*PI*RN*RZ);
                    sin_m1 = sin (2*PI*RL*RX);
                    sin_m2 = sin}(2*PI*RM*RY)
                    sin_m3 = sin (2*PI*RN*RZ);
                    KSQ = RL*RL + RM*RM + RN*RN;
                    AK = (2/V)*exp(-(PI*PI*KSQ)/(alpha*alpha))/KSQ;
                            term2 = AK * charge.at(i) * charge.at(j) * pow(2,3-count_zero);
                            Force_DNA_ions_buffer[0] += term2 * sin_m1*\operatorname{cos_m2*\operatorname{cos_m}3*RL;}
                    Force_DNA_ions_buffer[1] += term2 * cos_m1*sin_m2*cos_m3*RM;
                    Force_DNA_ions_buffer[2] += term2 * cos_m1*cos_m2*sin_m3*RN;
                }
                }
            }
            Pressure_DNA_ions.at(step) += Force_DNA_ions_buffer[0] * RX +
                Force_DNA_ions_buffer [1] * RY +
                Force_DNA_ions_buffer[2] * RZ;
                }
            }
    }
}
void Analysis::gather_Pressure_ions_ions(vector<vector<double> > xyz, int step)
{
    // Calculation of the force create by one DNA on the other.
    // (phosphate on phosphate)
    double RX;
    double RY;
    double RZ;
    double r_sq;
    double r;
    double V = box_x*box_y*box_z;
    double RL,RM,RN,KK,KSQ,AK;
int n, l,m;
int K;
double somme;
int count_zero;
double cos_m1,cos_m2,}\operatorname{cos_m3,\operatorname{sin}_m1,\operatorname{sin}_m2,\operatorname{sin}_m3;
double term1,term2;
```

329
330
331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366

```
vector <double> Force_ions_ions__buffer(3,0.0);
// Sum of the phosphates of DNA1
for(int i = atom_DNA; i < xyz.size()-1; i++)
{
    // Sum of the phosphates of DNA2
    for(int j = i+1; j < xyz.size(); j++)
    {
            if(charge.at(i)=0 || charge.at(j) == 0)
            {
            continue;
            }
            else
            {
                    Force_ions_ions_buffer [0] = 0;
                    Force_ions_ions_buffer[1] = 0;
                    Force_ions_ions_buffer[2] = 0;
                // Difference of distance
                RX = xyz.at(i)[0] - xyz.at(j)[0];
                RY = xyz.at(i)[1] - xyz.at(j)[1];
                RZ = xyz.at(i)[2] - xyz.at(j)[2];
            // Other version for PBC condition testing.
            RX = RX - box_x * round (RX / box_x);
            RY = RY - box_y * round (RY / box_y);
            RZ = RZ - box_z * round(RZ / box_z);
            r_sq = RX*RX + RY*RY + RZ*RZ;
            if( r_sq > rcut_sqq)
            {
            }
            else
            {
                r = sqrt(r_sq);
                    // Real part of the force calculated on each phosphate of DNA1.
                    term1 = charge.at(i) * charge.at(j)*
                    (2*alpha/sqrt(PI)*exp(-pow(alpha, 2)*r_sq) + erfc(alpha*r)/r );
                    Force__ions_ions_buffer[0] = term1 * (RX/r_sq);
                    Force_ions_ions_buffer[1] = term1 * (RY/r_sq);
                    Force__ions_ions_buffer[2] = term1 * (RZ/r_sq);
            }
            for ( l = 0; l <= Kmax; l++ )
            {
                    RL = double(1)/box_x;
                    for ( m = 0; m <= Kmax; m++ )
                    {
```

```
                RM = double (m)/box_y;
                for ( n = 0; n <= Kmax; n++ )
            {
        RN = double(n)/box_z;
        count_zero = 0;
        if(l= 0) count_zero++;
        if(m=0) count_zero++;
        if(n=0) count_zero++;
        // test of magnitude of K vector
        KK = l*l +m*m + n*n;
        if( KK >= Kmax*Kmax+2)
        {
            continue;
        }
        if( l=0 && m=0 && n=0)
        {
            continue;
        }
        cos_m1 = cos(2*PI*RL*RX);
        cos_m2 = cos(2*PI*RM*RY);
        cos_m3 = cos(2*PI*RN*RZ);
        sin_m1 = sin(2*PI*RL*RX);
        sin_m2 = sin}(2*PI*RM*RY)
        sin_m3 = sin}(2*PI*RN*RZ)
        KSQ = RL*RL + RM*RM + RN*RN;
        AK =(2/V)*exp(-(PI*PI*KSQ)/(alpha*alpha))/KSQ;
        term2 = AK * charge.at(i) * charge.at(j) * pow(2,3-count_zero);
        Force_ions_ions_buffer[0] += term2 *
        sin_m1*cos_m2*cos_m3*RL;
        Force_ions_ions_buffer[1] += term2 *
        cos_m1*sin_m2*cos_m3*RM;
        Force_ions_ions_buffer[2] += term2 *
        cos_m1*cos_m2*sin_m3*RN;
        }
            }
        }
            Pressure_ions_ions.at(step) += Force_ions_ions__buffer[0] * RX +
            Force_ions_ions_buffer[1] * RY +
            Force_ions_ions_buffer[2] * RZ;
        }
        }
    }
}
```

437

```
void Analysis::ReadTopology()
{
    ifstream iFS;
    iFS.open(_topol_file.c__str());
    string line;
    vector<string> split_line;
    int buffer_charge_index;
    double buffer_charge;
    vector<int> buffer_bond__index (2,0);
    vector<double> buffer__bond (2,0.0);
    vector<int> buffer_angle_index (3,0);
    vector<double> buffer_angle(2,0.0);
    int buffer_vdw__index;
    vector<double> buffer_vdw (2,0.0);
    if (iFS)
    {
        while(getline(iFS, line))
        {
        if(line[0] != '[' && line[0] !=';')
        {
            split_line = explode(line, '\t');
                switch(split_line.size())
                {
                case 2:
                buffer_charge_index = stoi(split_line.at(0));
                buffer_charge = stod(split_line.at(1));
                charge_index.push_back(buffer_charge_index);
                charge.push_back(buffer_charge);
                break;
                case 4:
                buffer_bond_index.at(0) = stoi(split_line.at(0));
                buffer_bond_index.at(1) = stoi(split_line.at(1));
                buffer_bond.at(0) = stod(split_line.at(2));
                buffer_bond.at(1) = stod(split_line.at(3));
                bond_index.push_back(buffer_bond__index);
                bond.push_back(buffer_bond);
                break;
                case 5:
```

```
                    buffer_angle__index.at(0) = stoi(split_line.at(0));
                    buffer__angle__index.at(2) = stoi(split__line.at(2));
                    buffer__angle.at(0) = stod(split_line.at (3));
                    buffer__angle.at(1) = stod(split__line.at(4));
                        angle__index.push__back(buffer__angle_index);
                    angle.push__back(buffer__angle);
                    break;
                    case 3:
                    buffer_vdw__index = stoi(split__line.at(0));
                    buffer__vdw.at(0) = stod(split__line.at(1));
                                    buffer__vdw.at(1) = stod(split_line.at (2));
                    vdw__index.push__back(buffer__vdw__index );
                    vdw.push__back(buffer__vdw );
                    break;
                    }
                }
            }
    }
    else
    {
    cout << " Cannot open " << __topol__file << endl;
    }
}
vector<string> Analysis:: explode(const string& s, const char& c)
{
    string buff{" " };
    vector<string> v;
    for(auto n:s)
    {
            if(n != c) buff+=n; else
            if(n=c && buff != " ") { v.push_back(buff); buff= " "; }
    }
    if(buff != " ") v.push__back(buff);
    return v;
}
void Analysis::gather__Pressure__with__units()
{
    double Fo = (q*q)/(4*PI*eps0*eps__water*Angstrom );
    for(int i = 0; i < Pressure__total.size(); i++)
    {
            // Total pressure is pressure of perfect__gaz +
```

```
            Pressure DNA DNA + Pressure DNA ions + Pressure ions ions.
            Pressure_total.at(i) = ((free_ions + gold_nano )*kb*T)/(V*pow (10, - 30)) +
            (1/(3*(V*pow (10, -30)))) * Fo
            *(pressure_DNA_DNA + Pressure_ions_ions.at(i) + Pressure_DNA_ions.at(i));
    }
}
void Analysis::Average_and_STD_Pressure(vector<double> &Pressure_total,
double &Pressure_average, double &Pressure_standard_deviation)
{
    double n = Pressure_total.size();
    // Calculation of the average of the component of force considered.
    for(int i = 0; i < n; i++)
    {
        Pressure_average += Pressure_total.at(i);
    }
    Pressure__average = Pressure_average/n;
    // Calculation of the standard mean deviation.
    for(int i = 0; i < n; i++)
    {
        Pressure_standard_deviation +=
        pow(Pressure_total.at(i) - Pressure_average,2);
    }
    Pressure_standard_deviation= sqrt((1/n)*Pressure_standard__deviation );
}
void Analysis::Average_Pressure()
{
    // Finalize average and STD of pressure.
    Average_and_STD_Pressure(Pressure_total, Pressure_average,
    Pressure_standard_deviation);
}
void Analysis::Output_each_step_init(string output_Pressure__step)
{
    // Initialization of flux to open files.
    ofstream oFS;
    // Reinitialization of the force calculation.
    oFS.open(output_Pressure_step.c_str());
    if (oFS)
    {
        oFS << "#step Pressure_total Pressure_perfect Pressure__elec" << endl;
    }
    else
    {
```

```
        cout << "We cannot open " << output__Pressure_step << endl;
```

        cout << "We cannot open " << output__Pressure_step << endl;
    }
    }
    oFS.close();
    oFS.close();
    oFS.clear();
    oFS.clear();
    }
void Analysis::Output_each_step(int step, string output_Pressure_step)
void Analysis::Output_each_step(int step, string output_Pressure_step)
{
double Fo = (q*q)/(4*PI*eps0*eps_water*Angstrom);
double Fo = (q*q)/(4*PI*eps0*eps_water*Angstrom);
// Initialization of flux to open files.
// Initialization of flux to open files.
ofstream oFS;
ofstream oFS;
// Reinitialization of the force calculation.
// Reinitialization of the force calculation.
oFS.open(output_Pressure_step.c_str(),ios::app);
oFS.open(output_Pressure_step.c_str(),ios::app);
if (oFS)
if (oFS)
{
{
oFS << step;
oFS << step;
oFS << " ";
oFS << " ";
oFS << fixed << setprecision(5) <<
oFS << fixed << setprecision(5) <<
((free_ions + gold_nano) * kb * T)/(V*pow (10, -30)) +
((free_ions + gold_nano) * kb * T)/(V*pow (10, -30)) +
(1/(3*(V*pow (10,-30)))) * Fo *
(1/(3*(V*pow (10,-30)))) * Fo *
(pressure_DNA_DNA + Pressure_ions_ions.at(step) + Pressure_DNA_ions.at(step));
(pressure_DNA_DNA + Pressure_ions_ions.at(step) + Pressure_DNA_ions.at(step));
oFS << " ";
oFS << " ";
oFS << fixed << setprecision(5) <<
oFS << fixed << setprecision(5) <<
((free_ions + gold_nano) * kb * T)/(V*pow (10,-30));
((free_ions + gold_nano) * kb * T)/(V*pow (10,-30));
oFS << " ";
oFS << " ";
oFS << fixed << setprecision(5) <<
oFS << fixed << setprecision(5) <<
(1/(3*(V*pow (10,-30)))) * Fo
(1/(3*(V*pow (10,-30)))) * Fo
*(pressure_DNA_DNA + Pressure_ions_ions.at(step) + Pressure_DNA_ions.at(step));
*(pressure_DNA_DNA + Pressure_ions_ions.at(step) + Pressure_DNA_ions.at(step));
oFS << endl;
oFS << endl;
}
}
else
else
{
{
cout << "We cannot open " << output_Pressure_step << endl;
cout << "We cannot open " << output_Pressure_step << endl;
}
}
oFS.close();
oFS.close();
oFS.clear();
oFS.clear();
}
void Analysis::Output_Pressure(string output_Pressure_final)
void Analysis::Output_Pressure(string output_Pressure_final)
{
{
// Initialization of flux to open files.
// Initialization of flux to open files.
ofstream oFS;
ofstream oFS;
// Reinitialization of the force calculation.
// Reinitialization of the force calculation.
oFS.open(output_Pressure_final.c_str());
oFS.open(output_Pressure_final.c_str());
oFS.close();
oFS.close();
oFS.clear();

```
    oFS.clear();
```

```
    // Output of the force calculation.
    oFS.open(output__Pressure__final.c__str());
    if (oFS)
    {
        oFS << "R(DNA-DNA) ";
        oFS << "Pressure ";
        oFS << "Pressure_STD ";
        oFS << " Pressure__perfect ";
        oFS << " Pressure__electro" << endl;
        oFS << fixed << setprecision (5) << box_x**0.5;
        oFS << " ";
        oFS << fixed << setprecision(5) << Pressure__average;
        oFS << " ";
        oFS << fixed << setprecision (5) << Pressure__standard__deviation;
        oFS << " ";
        oFS << fixed << setprecision(5) <<
        ((free__ions + gold__nano) * kb * T)/(V*pow (10, - 30));
        oFS << " ";
        oFS << fixed << setprecision(5) <<
        Pressure__average - ((free_ions + gold__nano) * kb * T)/(V*pow (10, - 30));
    }
    else
    {
        cout << "We cannot open " << output__Pressure_final << endl;
    }
    oFS.close();
    oFS.clear ();
```

\}

## A.5.3 Analysis_trajectory.h

Listing A. 24 - Analysis_trajectory.h
frame
/*
*

* Class to do the analysis
* 

*/
\#ifndef ANALYSIS_H
\#define ANALYSIS_H
\#include <vector>
\#include <string>
\#include <cmath>
\#define PI $\operatorname{acos}(-1.0)$
\#define eps0 $8.854188 * \operatorname{pow}(10,-12)$
\#define q $1.6021765 * \operatorname{pow}(10,-19)$
\#define Angstrom pow (10, - 10)
\#define Na 6.02214129*pow $(10,23)$

```
#define eps_water 78
#define KT 2477.7090204
#define kb 1.38064852*pow(10,-23)
#define T 298
using namespace std;
class Analysis{
    private:
    // Definition of total step;
    int total_MC__step;
    // Initialization of the species present in the system.
    int atom_DNA;
    int gold_nano;
    int free_ions;
    int atom__per_gold__nano;
    int Qion;
    int cation;
    int anion;
    // String for topology file.
    string _topol_file;
    double alpha;
    double Kmax;
    double rcut_sq;
    /* Vector containing the index of charge in the topology file */
    vector <int> charge_index;
    /* Vector containing the value of charge in the topology file */
    vector <double> charge;
    /* Vector containing the index of bond in the topology file */
    vector<vector<int> > bond__index;
    /* Vector containing the value of bond in the topology file */
    vector<vector<double>> bond;
    /* Vector containing the index of angle in the topology file */
    vector<vector<int> > angle_index;
    /* Vector containing the value of angle in the topology file */
    vector<vector<double> > angle;
    /* Vector containing the index of vdw in the topology file */
    vector<int> vdw_index;
    /* Vector containing the value of vdw in the topology file */
    vector<vector<double> > vdw;
```

```
    // Initialization of box size.
    double box_x;
    double box_y;
    double box_z;
    // Declaration of volume.
    double V;
    double pressure__DNA_DNA;
    // Pressure DNA_ions;
    vector<double> Pressure__DNA__ions;
    // Pressure ions_ion;
    vector<double> Pressure__ions__ions;
    // Vector to store the force through MC__step.
    vector<double> Pressure__total;
    // Declaration of variable for average pressure and std pressure
    double Pressure__average;
    double Pressure__standard__deviation;
    public:
    Analysis ();
    ~Analysis();
    Analysis(int total__MC__step, int atom_DNA, int gold__nano, int free__ions,
    int atom__per_gold_nano, int Qion, int cation, int anion,
    double box__x, double box_y, double box_z, string topolfile);
    void gather__Pressure__DNA__DNA(vector <vector <double> > xyz, int step);
    void gather__Pressure__DNA__ions(vector<vector<double>> xyz, int step );
    void gather__Pressure__ions__ions(vector<vector<double>> xyz, int step);
    void gather__Pressure__with__units();
    void Average__and__STD__Pressure(vector<double> &Pressure__total,
    double &Pressure__average, double &Pressure__standard__deviation );
    void Average__Pressure();
    void Output__each__step__init(string output__Pressure__step);
    void Output__each__step(int step, string output__Pressure__step);
    void Output_Pressure(string output_Pressure_final);
    void ReadTopology();
    vector<string> explode(const string& s, const char& c);
};
#endif
```


## A.5.4 Makefile

Listing A. 25 - Makefile
frame
1 exe: main.o read_trajectory.o Analysis.o

```
g++ -o exe main.o read_trajectory.o Analysis.o -lm -static-libstdc++
main.o: main.cpp
g++ -o main.o -c main.cpp -std=c++11
read_trajectory.o: read_trajectory.cpp
g++ -o read_trajectory.o -c read_trajectory.cpp -std=c++11
Analysis.o: Analysis.cpp
g++ -o Analysis.o -c Analysis.cpp -std=c++11
clean:
rm -rf *.o
```


## appendix B

## Some tests for the Monte Carlo simulation package

## Contents

B. 1 Test of the Ewald summation technique.
B.1.1 Determination of Madelung constant for NaCl crystal . . . . . . . . . . 365
B.1.2 Electrostatic interactions in water . . . . . . . . . . . . . . . . . . . . . 368
B. 2 Properties of the Kratky-Porod model . . . . . . . . . . . . . . . . . 369
B. 3 Test of the grand canonical MC scheme . . . . . . . . . . . . . . . . . 373

We perform numerical tests for several algorithms used in the Monte Carlo simulation package. In the first section, we check our implementation of the Ewald summation technique on two systems. First, we calculate the Coulomb energy of a rock salt $(\mathrm{NaCl})$ and calculate the Madelung constant. Then, the electrostatic interactions within a box filled with water molecules are calculated and compared with literature. Also, we implement the pivot algorithm and test it to calculate some properties of a semi-flexible polymer model: the Kratky-Porod model [313]. Finally, the grand canonical Monte carlo scheme (GCMC) is tested with an aqueous solution of monovalent salt.

## B. 1 Test of the Ewald summation technique

## B.1.1 Determination of Madelung constant for NaCl crystal

We test our implementation of the Ewald summation technique to calculate the Coulomb energy on a crystal lattice of NaCl to extract the Madelung constant.

The crystal structure of NaCl is displayed in Figure B.1 where the distance between ions is denoted by $d$.


Figure B. $1-\mathrm{NaCl}$ crystal structure.

The simple approach to calculate the ionic energy of an infinite crystal starts by considering an arbitrary ion in the crystal and calculating the energy contribution from the neighbors. If we consider the central black ion in picture B.1, the lattice energy writes as an infinite summation over neighbors. The black ion has 6 closest neighbors separated from $d, 12$ second nearest neighbors at a distance $d \sqrt{2}$ such that an infinite series can be constructed:

$$
\begin{equation*}
E=\frac{q_{+} q_{-}}{4 \pi \epsilon_{o} d}\left[6-\frac{12}{\sqrt{2}}+\frac{8}{\sqrt{3}}-\frac{6}{\sqrt{4}}+\frac{24}{\sqrt{5}}-\ldots\right] \tag{B.1}
\end{equation*}
$$

The infinite series is convergent and is denoted by $M$, the Madelung constant:

$$
\begin{equation*}
E=\frac{q_{+} q_{-}}{4 \pi \epsilon_{o} d} M \tag{B.2}
\end{equation*}
$$

We used some routines of the MC simulation package in order to calculate the Madelung constant using the Ewald summation method. The unit cell used to perform the calculations contains 8 ions shown in black in Figure B. 1

First, we remind that the energy calculated with the Ewald summation method (see section 3.2.4.3) is the sum of 3 terms:

- $\mathcal{U}^{S R}$ is the short range (called also "real space") contribution,
- $\mathcal{U}^{L R}$ is the long range (called also "reciprocal space") contribution,
- $\mathcal{U}^{\text {Self }}$ is the "correction" term contribution.

The cut-off for the reciprocal space is chosen to be $n_{c}=8$ and the different energy terms are plotted as a function of the splitting parameter $\alpha$ in Figure B. 2


Figure B. 2 - Energy contribution as a function of the splitting parameter $\alpha$ for $n_{c}=8$.

The total Coulombic energy curve reaches a plateau for $\alpha \in[5,11]$ with a value of -737.37 $\mathrm{kJ} / \mathrm{mol}$ which is consistent with the same calculation performed in the literature [314. Within this range, the contribution to the total Coulombic energy comes from the long range interaction $\mathcal{U}^{L R}$ and the correction contribution $\mathcal{U}^{\text {Self }}$. Due to the size of the simulation box that contains only 8 ions, $\mathcal{U}^{S R}$ is negligible over such range of splitting parameter $\alpha$. It is clear that the splitting parameter determines the relative importance of the short range and long range contributions to the total energy. When the parameter $\alpha$ increases, the long range contribution increases as well and one needs a larger cutoff $n_{c}$.

Along this line, we calculated the Madelung constant for NaCl and plotted it as a function of the splitting parameter in Figure B. 3 for different cutoffs in the reciprocal space. We obtained a Madelung constant of $M=1.74756$, which is the same value found for the same calculation


Figure B. 3 - Madelung constant for NaCl crystal in function of the splitting parameter $\alpha$.
performed in the work of Pratt [314]. A larger number of reciprocal vectors enables to estimate the Madelung constant for higher values of the splitting parameter alpha, although the execution time of the program is much longer.

Such calculations validate our implementation of the Ewald summation technique in the MC simulation package.

## B.1.2 Electrostatic interactions in water

We test our implementation of the Ewald summation technique to calculate the electrostatic interactions in water. We used the parameters for the Ewald summation technique and check the validity of our implementation by using data from NIST website [315.

The water model used to perform the calculation is characterized by an hydrogen charge $q_{H}=$ $+0.4238|\mathrm{e}|$ and an oxygen charge $q_{O}=-0.8476|\mathrm{e}|$, a bond length between the oxygen and hydrogen of $r_{O H}=1 \AA$ and an angle $\phi=109.47$ degrees between bonds as shown in Figure B. 4

The Ewald parameters are settled in the following way according to the NIST page 315:


Figure B. 4 - Scheme of the water molecule.

$$
\left\{\begin{array}{l}
\alpha=\frac{5.6}{\min \left(L_{x}, L_{y}, L_{z}\right)}  \tag{B.3}\\
n_{c}=5 \\
r_{c u t}=10 \AA
\end{array}\right.
$$

where $L_{x}, L_{y}$ and $L_{z}$ are the box sidelengths. The electrostatic interactions has been calculated for different box sizes. The results obtained with our implementation of the Ewald summation technique are given in Table B. 1 We remind that the electrostatic energy is the contribution of three terms $\mathcal{U}^{E L}=\mathcal{U}^{S R}+\mathcal{U}^{L R}+\mathcal{U}^{\text {Self }}$.

Table B. 1 - Calculation of the electrostatic interaction in water.

| system | 1 | 2 | 3 | 4 |
| :--- | :---: | :---: | :---: | :---: |
| Number of water molecules | 100 | 200 | 300 | 750 |
| $L_{x}=L_{y}=L_{z}(\AA)$ | 20 | 20 | 20 | 30 |
| $\mathcal{U}^{S R}(\mathrm{~kJ} / \mathrm{mol})$ | 18716.13 | 36806.42 | 53766.53 | 88971.80 |
| $\mathcal{U}^{L R}(\mathrm{~kJ} / \mathrm{mol})$ | 52.13 | 50.17 | 43.60 | 63.08 |
| $\mathcal{U}^{\text {Self }}(\mathrm{kJ} / \mathrm{mol})$ | -23651.51 | -47303.02 | -70954.53 | -118257.97 |
| $\mathcal{U}^{E L}(\mathrm{~kJ} / \mathrm{mol})$ | -4883.25 | -10446.43 | -17144.40 | -29223.09 |

We obtain the same results as those presented in the NIST website 315. By this way, we ensure that the electrostatic energy have been properly implemented in the MC simulation package.

## B. 2 Properties of the Kratky-Porod model

The worm like chain model has been first proposed by Kratky and Porod in 1949 and is called the Kratky-Porod model [316. This model is suitable to describe relatively stiff polyelectrolyte
under the following assumptions:

- Each angle defined by $\theta$ between bonds $i$ and $i+1$ are constrained with the following potential:

$$
\begin{equation*}
E=-C \cos \theta \tag{B.4}
\end{equation*}
$$

where $C$ is the bending coefficient.

- The bond length $l$ is small compared to the length $L$ of the chain.

A bond vector is a vector associated with a bond made between two consecutive beads of the chain as shown in Figure B. 5


Figure B. 5 - Illustration of the Kratky-Porod model. Each bond linking two consecutive beads is associated with a vector $\mathbf{r}_{i}$. The angle $\theta$ is made between two successive bonds.

Let us express the angular correlations between two bond vectors $\mathrm{r}_{i}$ and $\mathrm{r}_{j}$.

$$
\begin{array}{cc}
\left\langle\mathbf{r}_{i} \cdot \mathbf{r}_{j}\right\rangle & =\left\langle\frac{\left(\mathbf{r}_{i} \cdot \mathbf{r}_{i+1}\right) \mathbf{r}_{i+1}}{\left|\mathbf{r}_{i+1}\right|^{2}} \cdot \mathbf{r}_{j}\right\rangle \\
\left\langle\mathbf{r}_{i} \cdot \mathbf{r}_{j}\right\rangle & =\left\langle\frac{\left(\mathbf{r}_{i} \cdot \mathbf{r}_{i+1}\right) \mathbf{\mathbf { r } _ { i + 1 }}}{\left|\mathbf{r}_{i+1}\right|^{2}} \frac{\left(\mathbf{r}_{i+1} \cdot \mathbf{r}_{i+2}\right) \mathbf{r}_{i+2}}{\left|\mathbf{r}_{i+2}\right|^{2}} \cdot \mathbf{r}_{j}\right\rangle \\
\left\langle\mathbf{r}_{i} \cdot \mathbf{r}_{j}\right\rangle=\left\langle\frac{\left(\mathbf{r}_{i} \cdot \mathbf{r}_{i+1}\right) \mathbf{r}_{i+1}}{\left|\mathbf{r}_{i+1}\right|^{2}} \frac{\left(\mathbf{r}_{i+1} \cdot \mathbf{r}_{i+2}\right) \mathbf{r}_{i+2}}{\left|\mathbf{r}_{i+2}\right|^{2}} \ldots \frac{\left(\mathbf{r}_{j-1} \cdot \mathbf{r}_{j}\right) \mathbf{r}_{j}}{\left|\mathbf{r}_{j}\right|^{2}} \cdot \mathbf{r}_{j}\right\rangle \\
\left\langle\mathbf{r}_{i} \cdot \mathbf{r}_{j}\right\rangle & =l^{2}\langle\cos \theta\rangle^{j-i} \tag{B.8}
\end{array}
$$

We introduced the persistence length $l_{p}$ of the polymer chain [313] through the bond correlations:

$$
\begin{equation*}
\left\langle\mathbf{r}_{i} \cdot \mathbf{r}_{j}\right\rangle=l^{2} e^{\left(-\frac{(j-i) l}{l_{p}}\right)} \tag{B.9}
\end{equation*}
$$

The persistence length can be related to the bending coefficient $C$ through the following relationship [317]:

$$
\begin{equation*}
l_{p}=\frac{l C}{k_{b} T} \tag{B.10}
\end{equation*}
$$

We aim to calculate the persistence length of a Kratky-Porod model chain by using our MC simulation package. It would represent a proper test for the pivot algorithm implemented in section 3.2.3.2

In order to simplify calculations, we will set $k_{B} T=1 \mathrm{~kJ} / \mathrm{mol}$ and each bond length to be equal to $1 \AA$. Each simulation has been performed with 2000 MC steps which means that the pivot algorithm has been applied 2000 times for each bond. The chain contains 100 beads, resulting of 99 bonds.

We plotted the normalized angular correlation function and its logarithm for different values of the bending coefficient for the angular force acting between bonds in Figure B.6.

We obtain a decaying exponential function as predicted by formula B. 9


Figure B. 6 - Top) Normalized angular correlation function for chains characterized by different bending coefficients C. Bottom) Logarithm of the angular correlation function.

$$
\begin{equation*}
\frac{\left\langle\mathbf{r}_{i} \cdot \mathbf{r}_{j}\right\rangle}{l^{2}}=e^{\left(-\frac{(j-i) l}{l_{p}}\right)} \tag{B.11}
\end{equation*}
$$

It is thus straightforward to extract the persistence length and we summarize the results in

Table B. 2 - Calculation of persistence length of the bead chain model for different bending coefficients.

| Bending coefficient $C\left(\mathrm{~kJ} /\left(\mathrm{mol} \cdot \mathrm{rad}^{2}\right)\right)$ | Theoretical $l_{p}(\AA)$ | simulated $l_{p}(\AA)$ |
| :--- | :---: | :---: |
| 5 | 5 | 5.04 |
| 10 | 10 | 10.15 |
| 20 | 20 | 20.22 |
| 30 | 30 | 29.94 |

Table B. 2 We thus ensure that the pivot algorithm is well implemented in the MC simulation package.

## B. 3 Test of the grand canonical MC scheme

We modified the MC simulation package to perform grand canonical MC simulation (GCMC). In order to validate the GCMC scheme, we perform simulations on simple systems of solution of monovalent salt with an implicit water modeled using a dielectric constant $\epsilon=78$. The GCMC algorithm requires the chemical potential as input to perform a simulation. The validation of the GCMC algorithm is based on two steps:

- First, we extract the excess chemical potential with the Widom technique (see section 3.2.5) from a standard NVT MC simulation of an aqueous solution of monovalent salt.
- The chemical potential is used to perform a GCMC simulation of the same system. The average number of ion pairs is calculated and compared with the constant number of ion pairs of the previous NVT MC simulation.

In an orthorombic simulation box, we performed a series of NVT MC simulations for systems containing either $6,12,24$ or 48 pairs of monovalent salt ions of diameter $d_{+}=d_{-}=3 \AA$. The volume of the box is set to $V=(48 \AA) \times(41.57 \AA) \times(102 \AA)$ and the simulation is performed for $2 \times 10^{5} \mathrm{MC}$ cycles.

We remind that the chemical potential is expressed as:

$$
\begin{equation*}
\mu=\mu_{i d}+\mu_{e x} \tag{B.12}
\end{equation*}
$$

Table B. 3 - The number of ion pairs during NVT MC simulation is presented with $\mu_{e x}$ calculated using Widom insertion technique.

| salt pairs (NVT) | $\mu_{e x}(\mathrm{~kJ} / \mathrm{mol})$ |
| :--- | :---: |
| 6 | $-1.2988(+/-0.2445)$ |
| 12 | $-1.5826(+/-0.3157)$ |
| 24 | $-1.9705(+/-0.3984)$ |
| 48 | $-2.4364(+/-0.4833)$ |

The Widom insertion technique is used to determine $\mu_{e x}$ whereas $\mu_{i d}$ can be calculated analytically. The excess of energy is calculated every cycle to have a fair approximation of $\mu_{e x}$. The results are summarized in Table B. 3

We used the excess chemical potential $\mu_{e x}$ to calculate the total chemical potential $\mu$ and perform GCMC simulations for $2 \times 10^{5} \mathrm{MC}$ cycles. We kept the same orthorombic box and for each system, there is 48 salt pairs present at the beginning of the simulation.

We set the GCMC scheme such that there is the same probability to add/delete a salt pair or to displace an ion in the box.

After equilibration of the systems and insertion/deletion of ion pairs during the simulations, the average number of salt pairs in the box when the system is equilibrated is presented in Table

## B. 4

Table B. 4 - Comparison between the number of salt pairs during NVT MC simulations and the number of salt pairs in GCMC simulations for which the chemical potential has been calculated from the NVT MC simulations.

| salt pairs $(\mathrm{NVT})$ | salt pairs $(\mu \mathrm{VT})$ |
| :--- | :---: |
| 6 | $6.03(+/-2.61)$ |
| 12 | $11.97(+/-3.84)$ |
| 24 | $23.96(+/-4.46)$ |
| 48 | $48.31(+/-8.35)$ |

We found similar average number of salt pairs during the canonical and the grand canonical Monte Carlo simulations which validates our implementation of the Widom insertion technique to calculate the chemical potential.

## appendix $C$

## Monte Carlo error analysis

## Contents

C. 1 Effective force between a pair of DNA with counterions . . . . . . . 377
C. 2 Osmotic pressure in a hexagonal bundle of DNA condensed with counterions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 380
C. 3 Effective force between a pair of DNA with AuNPs. . . . . . . . . . 382
C. 4 Osmotic pressure in DNA lattices with AuNPs . . . . . . . . . . . . 388

We provide an estimation of the error associated to the calculation of the forces and the osmotic pressures presented in chapter 4 The Monte Carlo simulation method yields to approximate results and the accuracy depends on the number of values $N$ used to calculate the average of a given quantity $A$. The determination of the error is based on the calculation of the variance $\sigma^{2}$ defined as:

$$
\begin{equation*}
\sigma^{2}=\left\langle A^{2}\right\rangle-\langle A\rangle^{2} \tag{C.1}
\end{equation*}
$$

where

$$
\begin{equation*}
\langle A\rangle=\frac{1}{N} \sum_{i=1}^{N} A_{i} \tag{C.2}
\end{equation*}
$$

and

$$
\begin{equation*}
\left\langle A^{2}\right\rangle=\frac{1}{N} \sum_{i=1}^{N}\left(A_{i}\right)^{2} \tag{C.3}
\end{equation*}
$$

The standard deviation is given by the square root of the variance $\sqrt{\sigma^{2}}$. However, it is expected that the error decreases with the number of points $N$ which is not the case by using equation C.1 Instead, from a simulation where we extracted $N$ measures, we divide the set of measures into $L$ blocks containing $M=N / L$ measures and we compute the average for each block of size $M$ [208] so that:

$$
\begin{equation*}
\langle A\rangle_{k}=\frac{1}{M} \sum_{i=M *(k-1)+1}^{M * k} A_{i} \tag{C.4}
\end{equation*}
$$

where $k$ varies from 1 to $L$. We ensure that the size of the block are higher than the correlation length calculated by the autocorrelation function $\mathrm{c}(\mathrm{l})$ defined as:

$$
\begin{equation*}
c(l)=\frac{1}{N-l} \sum_{i=1}^{N-l}\left(A_{i}-\langle A\rangle\right)\left(A_{(i+l)}-\langle A\rangle\right) \tag{C.5}
\end{equation*}
$$

Finally the error $\Delta_{A}$ is calculated by computing the "standard deviation of the means" denoted by $\sigma_{M}$ :

$$
\begin{equation*}
\Delta_{A}=\sqrt{\sigma_{M}}=\sqrt{\left\langle I^{2}\right\rangle-\langle I\rangle^{2}} \tag{C.6}
\end{equation*}
$$

where

$$
\begin{equation*}
\langle I\rangle=\frac{1}{L} \sum_{k=1}^{L}\langle A\rangle_{k} \tag{C.7}
\end{equation*}
$$

and

$$
\begin{equation*}
\left\langle I^{2}\right\rangle=\frac{1}{L} \sum_{k=1}^{L}\langle A\rangle_{k}^{2} \tag{C.8}
\end{equation*}
$$

In the analysis, we have chosen $L=10$.

## C. 1 Effective force between a pair of DNA with counterions

We calculated the error associated with the calculation of effective force between a pair of parallel DNA molecules in presence of counterions of valency $\mathrm{q}=+1,+2,+3$ or $+4|\mathrm{e}|$ (see section 4.2 .2 .

The effective force writes as following:

$$
\begin{equation*}
\mathbf{F}=\mathbf{F}_{1}+\mathbf{F}_{2} \tag{C.9}
\end{equation*}
$$

The first term $\mathbf{F}_{1}$ is the direct Coulomb force exerted by each phosphate group of a DNA on the other phosphate groups of the other DNA:

$$
\begin{equation*}
\mathbf{F}_{1}=-\sum_{k=1}^{N_{p}}\left(\vec{\nabla}_{\mathbf{r}_{k}^{p}} \sum_{n=1}^{N_{p}} V_{p p}\left(\left|\mathbf{r}_{k}^{p}-\mathbf{r}_{n}^{p}\right|\right)\right) \tag{C.10}
\end{equation*}
$$

where $V_{i j}$ is the interaction potential as described in section 3.2.4. Given that the DNA are fixed during the simulation, this term has to be computed only once.

The second term $\mathbf{F}_{2}$ represents the Coulomb interactions between the counterions in the box and phosphate groups:

$$
\begin{equation*}
\mathbf{F}_{2}=-\sum_{k=1}^{N_{p}}\left(\left\langle\sum_{l=1}^{N_{c}} \vec{\nabla}_{\mathbf{r}_{k}^{p}} V_{p c}\left(\left|\mathbf{r}_{k}^{p}-\mathbf{r}_{l}^{c}\right|\right)\right\rangle\right) \tag{C.11}
\end{equation*}
$$

The force is calculated along the $O x$ axis of the simulation box:

$$
\begin{equation*}
F=\mathbf{F} \cdot \mathbf{e}_{x}=\left(\mathbf{F}_{1}+\mathbf{F}_{2}\right) \cdot \mathbf{e}_{x}=F_{1}+F_{2} \tag{C.12}
\end{equation*}
$$

The force are normalized by $F_{o}=k_{b} \mathrm{~T} / \mathrm{P}$ where P is the pitch of the DNA
Table C. 1 - Effective force exerted between the DNA pair in presence of 120 monovalent counterions. The data are gathered during $4 \times 10^{5} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.

| $\ell(\AA)$ | $\mathrm{F} / \mathrm{F}_{o}(\mathrm{q}=+1)$ | $\mathrm{F}_{1} / \mathrm{F}_{o}(\mathrm{q}=+1)$ | $\mathrm{F}_{2} / \mathrm{F}_{o}(\mathrm{q}=+1)$ |
| :---: | :---: | :---: | :---: |
| 24 | $-22.1 \pm 4.3$ | $-192.2 \pm 0.0$ | $170.1 \pm 4.3$ |
| 26 | $-14.8 \pm 3.8$ | $-172.8 \pm 0.0$ | $158.0 \pm 3.8$ |
| 28 | $-9.7 \pm 3.4$ | $-152.7 \pm 0.0$ | $143.0 \pm 3.4$ |
| 30 | $-7.3 \pm 3.2$ | $-134.7 \pm 0.0$ | $127.4 \pm 3.2$ |
| 32 | $-5.7 \pm 3.3$ | $-118.3 \pm 0.0$ | $112.6 \pm 3.3$ |
| 34 | $-4.5 \pm 2.9$ | $-103.2 \pm 0.0$ | $98.7 \pm 2.9$ |
| 36 | $-3.6 \pm 2.7$ | $-89.1 \pm 0.0$ | $85.5 \pm 2.7$ |
| 38 | $-2.8 \pm 2.8$ | $-75.8 \pm 0.0$ | $73.0 \pm 2.8$ |
| 40 | $-2.5 \pm 2.6$ | $-63.2 \pm 0.0$ | $60.7 \pm 2.6$ |
| 42 | $-1.9 \pm 2.7$ | $-51.1 \pm 0.0$ | $49.2 \pm 2.7$ |
| 44 | $-1.4 \pm 2.5$ | $-39.3 \pm 0.0$ | $37.9 \pm 2.5$ |
| 46 | $-0.9 \pm 2.4$ | $-27.9 \pm 0.0$ | $27.0 \pm 2.4$ |
| 48 | $-0.5 \pm 2.2$ | $-16.6 \pm 0.0$ | $16.1 \pm 2.2$ |

Table C. 2 - Effective force exerted between the DNA pair in presence of 60 divalent counterions. The data are gathered during $4 \times 10^{5} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.

| $\ell(\AA)$ | $\mathrm{F} / \mathrm{F}_{o}(\mathrm{q}=+2)$ | $\mathrm{F}_{1} / \mathrm{F}_{o}(\mathrm{q}=+2)$ | $\mathrm{F}_{2} / \mathrm{F}_{o}(\mathrm{q}=+2)$ |
| :---: | :---: | :---: | :---: |
| 24 | $6.4 \pm 5.2$ | $-192.2 \pm 0.0$ | $198.6 \pm 5.2$ |
| 26 | $6.2 \pm 4.4$ | $-172.8 \pm 0.0$ | $179.0 \pm 4.4$ |
| 28 | $3.4 \pm 4.1$ | $-152.7 \pm 0.0$ | $156.1 \pm 4.1$ |
| 30 | $1.5 \pm 3.6$ | $-134.7 \pm 0.0$ | $136.2 \pm 3.6$ |
| 32 | $0.9 \pm 3.5$ | $-118.3 \pm 0.0$ | $119.2 \pm 3.5$ |
| 34 | $0.6 \pm 3.7$ | $-103.2 \pm 0.0$ | $103.8 \pm 3.7$ |
| 36 | $0.4 \pm 3.2$ | $-89.1 \pm 0.0$ | $89.5 \pm 3.2$ |
| 38 | $0.3 \pm 2.9$ | $-75.8 \pm 0.0$ | $76.1 \pm 2.9$ |
| 40 | $0.1 \pm 3.0$ | $-63.2 \pm 0.0$ | $63.3 \pm 3.0$ |
| 42 | $0.0 \pm 2.8$ | $-51.1 \pm 0.0$ | $51.1 \pm 2.8$ |
| 44 | $0.1 \pm 2.6$ | $-39.3 \pm 0.0$ | $39.4 \pm 2.6$ |
| 46 | $0.0 \pm 2.7$ | $-27.9 \pm 0.0$ | $27.9 \pm 2.7$ |
| 48 | $-0.1 \pm 2.6$ | $-16.6 \pm 0.0$ | $16.5 \pm 2.6$ |

Table C. 3 - Effective force exerted between the DNA pair in presence of 40 trivalent counterions. The data are gathered during $4 \times 10^{5} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.

| $\ell(\AA)$ | $\mathrm{F} / \mathrm{F}_{o}(\mathrm{q}=+3)$ | $\mathrm{F}_{1} / \mathrm{F}_{o}(\mathrm{q}=+3)$ | $\mathrm{F}_{2} / \mathrm{F}_{o}(\mathrm{q}=+3)$ |
| :---: | :---: | :---: | :---: |
| 24 | $21.5 \pm 4.6$ | $-192.2 \pm 0.0$ | $213.7 \pm 2.4$ |
| 26 | $14.5 \pm 4.1$ | $-172.8 \pm 0.0$ | $187.3 \pm 4.6$ |
| 28 | $7.3 \pm 3.6$ | $-152.7 \pm 0.0$ | $160.0 \pm 4.1$ |
| 30 | $4.1 \pm 3.8$ | $-134.7 \pm 0.0$ | $138.8 \pm 3.6$ |
| 32 | $2.0 \pm 3.1$ | $-118.3 \pm 0.0$ | $120.3 \pm 3.8$ |
| 34 | $1.4 \pm 3.2$ | $-103.2 \pm 0.0$ | $104.6 \pm 3.1$ |
| 36 | $0.5 \pm 3.0$ | $-89.1 \pm 0.0$ | $89.6 \pm 3.2$ |
| 38 | $1.0 \pm 2.7$ | $-75.8 \pm 0.0$ | $76.8 \pm 3.0$ |
| 40 | $0.4 \pm 3.8$ | $-63.2 \pm 0.0$ | $63.6 \pm 2.7$ |
| 42 | $0.0 \pm 2.6$ | $-51.1 \pm 0.0$ | $51.1 \pm 3.8$ |
| 44 | $0.2 \pm 2.7$ | $-39.3 \pm 0.0$ | $39.5 \pm 2.6$ |
| 46 | $0.1 \pm 2.5$ | $-27.9 \pm 0.0$ | $28.0 \pm 2.7$ |
| 48 | $0.2 \pm 2.4$ | $-16.6 \pm 0.0$ | $16.8 \pm 2.5$ |

Table C. 4 - Effective force exerted between the DNA pair in presence of 30 tetravalent counterions. The data are gathered during $4 \times 10^{5} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.

| $\ell(\AA)$ | $\mathrm{F} / \mathrm{F}_{o}(\mathrm{q}=+4)$ | $\mathrm{F}_{1} / \mathrm{F}_{o}(\mathrm{q}=+4)$ | $\mathrm{F}_{2} / \mathrm{F}_{o}(\mathrm{q}=+4)$ |
| :---: | :---: | :---: | :---: |
| 24 | $44.3 \pm 5.7$ | $-192.2 \pm 0.0$ | $236.5 \pm 5.7$ |
| 26 | $25.2 \pm 5.4$ | $-172.8 \pm 0.0$ | $198.0 \pm 5.4$ |
| 28 | $10.7 \pm 5.6$ | $-152.7 \pm 0.0$ | $163.4 \pm 5.6$ |
| 30 | $5.7 \pm 5.2$ | $-134.7 \pm 0.0$ | $140.4 \pm 5.2$ |
| 32 | $3.5 \pm 4.9$ | $-118.3 \pm 0.0$ | $121.8 \pm 4.9$ |
| 34 | $1.2 \pm 4.7$ | $-103.2 \pm 0.0$ | $104.4 \pm 4.7$ |
| 36 | $1.2 \pm 4.8$ | $-89.1 \pm 0.0$ | $90.3 \pm 4.8$ |
| 38 | $0.1 \pm 4.2$ | $-75.8 \pm 0.0$ | $75.9 \pm 4.2$ |
| 40 | $0.9 \pm 4.3$ | $-63.2 \pm 0.0$ | $64.1 \pm 4.3$ |
| 42 | $0.5 \pm 3.9$ | $-51.1 \pm 0.0$ | $51.6 \pm 3.9$ |
| 44 | $0.0 \pm 3.7$ | $-39.3 \pm 0.0$ | $39.3 \pm 3.7$ |
| 46 | $0.3 \pm 3.4$ | $-27.9 \pm 0.0$ | $28.2 \pm 3.4$ |
| 48 | $0.0 \pm 3.3$ | $-16.6 \pm 0.0$ | $16.6 \pm 3.3$ |

## C. 2 Osmotic pressure in a hexagonal bundle of DNA condensed with counterions

We calculated the error associated with the calculation of osmotic pressure in a hexagonal bundle of DNA condensed with counterions of valency $\mathrm{q}=+1,+2,+3$ or $+4|\mathrm{e}|$ (see section 4.2.3).

The osmotic pressure writes:

$$
\begin{gather*}
\Pi=\Pi_{\text {ideal }}+\Pi_{\text {electrostatics }}  \tag{C.13}\\
\Pi=\frac{N k_{b} T}{V^{\prime}}+\frac{1}{3 V^{\prime}} \sum_{i<j} \mathbf{r}_{i j} \mathbf{F}_{i j} \tag{C.14}
\end{gather*}
$$

where $V^{\prime}$ is the accessible volume to the free ions.

Table C. 5 - Osmotic pressure calculated in a hexagonal bundle of DNA with counterions of different valency $\mathrm{q}=+1,+2,+3$ or $+4|\mathrm{e}|$ corresponding respectively to a number of $240,120,80$ and 60 counterions in the bundle. The diameter of the counterions is $\mathrm{d}_{c}=1 \AA$. The data are gathered during $4 \times 10^{5} \mathrm{MC}$ cycles to calculate statistics.

| $\ell(\AA)$ | $\Pi($ atm. $)(\mathrm{q}=+1)$ | $\Pi(\mathrm{atm}).(\mathrm{q}=+2)$ | $\Pi(\mathrm{atm}).(\mathrm{q}=+3)$ | $\Pi(\mathrm{atm}).(\mathrm{q}=+4)$ |
| :---: | :---: | :---: | :---: | :---: |
| 24 | $(1.909 \pm 0.072)$ | $(-2.226 \pm 0.101)$ | $(-4.582 \pm 0.114)$ | $(-6.205 \pm 0.113)$ |
| 26 | $(1.747 \pm 0.062)$ | $(-1.747 \pm 0.089)$ | $(-3.732 \pm 0.097)$ | $(-5.019 \pm 0.986)$ |
| 28 | $(1.595 \pm 0.054)$ | $(-1.437 \pm 0.080)$ | $(-3.176 \pm 0.085)$ | $(-4.274 \pm 0.851)$ |
| 30 | $(1.462 \pm 0.049)$ | $(-1.185 \pm 0.071)$ | $(-2.752 \pm 0.072)$ | $(-3.685 \pm 0.724)$ |
| 32 | $(1.339 \pm 0.042)$ | $(-1.020 \pm 0.065)$ | $(-2.401 \pm 0.066)$ | $(-3.226 \pm 0.593)$ |
| 34 | $(1.216 \pm 0.039)$ | $(-0.886 \pm 0.057)$ | $(-2.129 \pm 0.059)$ | $(-2.858 \pm 0.514)$ |
| 36 | $(1.121 \pm 0.034)$ | $(-0.775 \pm 0.051)$ | $(-1.896 \pm 0.051)$ | $(-2.544 \pm 0.467)$ |
| 38 | $(1.031 \pm 0.031)$ | $(-0.685 \pm 0.047)$ | $(-1.698 \pm 0.045)$ | $(-2.271 \pm 0.406)$ |
| 40 | $(0.949 \pm 0.028)$ | $(-0.604 \pm 0.042)$ | $(-1.531 \pm 0.042)$ | $(-2.058 \pm 0.355)$ |
| 42 | $(0.876 \pm 0.027)$ | $(-0.546 \pm 0.038)$ | $(-1.386 \pm 0.040)$ | $(-1.857 \pm 0.314)$ |
| 44 | $(0.809 \pm 0.024)$ | $(-0.495 \pm 0.035)$ | $(-1.263 \pm 0.035)$ | $(-1.689 \pm 0.294)$ |
| 46 | $(0.748 \pm 0.022)$ | $(-0.445 \pm 0.033)$ | $(-1.151 \pm 0.031)$ | $(-1.545 \pm 0.256)$ |
| 48 | $(0.699 \pm 0.021)$ | $(-0.408 \pm 0.030)$ | $(-1.058 \pm 0.029)$ | $(-1.419 \pm 0.240)$ |
| 50 | $(0.650 \pm 0.019)$ | $(-0.369 \pm 0.029)$ | $(-0.972 \pm 0.026)$ | $(-1.310 \pm 0.218)$ |
| 52 | $(0.607 \pm 0.017)$ | $(-0.338 \pm 0.025)$ | $(-0.900 \pm 0.024)$ | $(-1.209 \pm 0.202)$ |
| 54 | $(0.568 \pm 0.017)$ | $(-0.311 \pm 0.024)$ | $(-0.834 \pm 0.023)$ | $(-1.120 \pm 0.183)$ |
| 56 | $(0.533 \pm 0.015)$ | $(-0.288 \pm 0.022)$ | $(-0.775 \pm 0.022)$ | $(-1.042 \pm 0.171)$ |
| 58 | $(0.501 \pm 0.013)$ | $(-0.265 \pm 0.020)$ | $(-0.721 \pm 0.020)$ | $(-0.966 \pm 0.162)$ |
| 60 | $(0.472 \pm 0.014)$ | $(-0.248 \pm 0.019)$ | $(-0.672 \pm 0.018)$ | $(-0.910 \pm 0.144)$ |
| 62 | $(0.446 \pm 0.013)$ | $(-0.228 \pm 0.018)$ | $(-0.630 \pm 0.017)$ | $(-0.851 \pm 0.143)$ |
| 64 | $(0.421 \pm 0.012)$ | $(-0.214 \pm 0.017)$ | $(-0.592 \pm 0.016)$ | $(-0.799 \pm 0.129)$ |
| 66 | $(0.398 \pm 0.011)$ | $(-0.198 \pm 0.015)$ | $(-0.553 \pm 0.015)$ | $(-0.748 \pm 0.121)$ |
| 68 | $(0.378 \pm 0.011)$ | $(-0.185 \pm 0.015)$ | $(-0.524 \pm 0.014)$ | $(-0.705 \pm 0.115)$ |
| 70 | $(0.358 \pm 0.010)$ | $(-0.175 \pm 0.014)$ | $(-0.491 \pm 0.013)$ | $(-0.668 \pm 0.110)$ |
| 72 | $(0.341 \pm 0.009)$ | $(-0.167 \pm 0.013)$ | $(-0.467 \pm 0.013)$ | $(-0.626 \pm 0.106)$ |
| 74 | $(0.323 \pm 0.009)$ | $(-0.155 \pm 0.013)$ | $(-0.442 \pm 0.012)$ | $(-0.596 \pm 0.096)$ |

Table C. 6 - Osmotic pressure calculated in a hexagonal bundle of DNA with divalent counterions ( $\times 120$ counterions in the bundle). The diameter of the counterions is varied from $\mathrm{d}_{c}=1,2,4$ and $6 \AA$. The data are gathered during $4 \times 10^{5} \mathrm{MC}$ cycles to calculate statistics.

| $\ell(\AA)$ | $\Pi($ atm. $)\left(\mathrm{d}_{c}=0 \AA\right)$ | $\Pi(\mathrm{atm}).\left(\mathrm{d}_{c}=1 \AA\right)$ | $\Pi(\mathrm{atm}).\left(\mathrm{d}_{c}=4 \AA\right)$ | $\Pi(\mathrm{atm}).\left(\mathrm{d}_{c}=6 \AA\right)$ |
| :---: | :---: | :---: | :---: | :---: |
| 24 | $(-2.757 \pm 0.116)$ | $(-2.226 \pm 0.101)$ | $(-1.145 \pm 0.077)$ | $(-0.870 \pm 0.060)$ |
| 26 | $(-2.252 \pm 0.104)$ | $(-1.747 \pm 0.089)$ | $(-0.830 \pm 0.064)$ | $(-0.599 \pm 0.055)$ |
| 28 | $(-1.896 \pm 0.089)$ | $(-1.437 \pm 0.080)$ | $(-0.581 \pm 0.055)$ | $(-0.395 \pm 0.049)$ |
| 30 | $(-1.602 \pm 0.082)$ | $(-1.185 \pm 0.071)$ | $(-0.413 \pm 0.047)$ | $(-0.230 \pm 0.043)$ |
| 32 | $(-1.387 \pm 0.072)$ | $(-1.020 \pm 0.065)$ | $(-0.312 \pm 0.042)$ | $(-0.130 \pm 0.039)$ |
| 34 | $(-1.219 \pm 0.064)$ | $(-0.886 \pm 0.057)$ | $(-0.246 \pm 0.037)$ | $(-0.086 \pm 0.035)$ |
| 36 | $(-1.070 \pm 0.059)$ | $(-0.775 \pm 0.051)$ | $(-0.195 \pm 0.034)$ | $(-0.036 \pm 0.032)$ |
| 38 | $(-0.952 \pm 0.052)$ | $(-0.685 \pm 0.047)$ | $(-0.155 \pm 0.031)$ | $(-0.014 \pm 0.030)$ |
| 40 | $(-0.855 \pm 0.048)$ | $(-0.604 \pm 0.042)$ | $(-0.133 \pm 0.028)$ | $(0.003 \pm 0.026)$ |
| 42 | $(-0.766 \pm 0.044)$ | $(-0.546 \pm 0.038)$ | $(-0.113 \pm 0.026)$ | $(0.021 \pm 0.025)$ |
| 44 | $(-0.695 \pm 0.041)$ | $(-0.495 \pm 0.035)$ | $(-0.094 \pm 0.024)$ | $(0.016 \pm 0.023)$ |
| 46 | $(-0.624 \pm 0.036)$ | $(-0.445 \pm 0.033)$ | $(-0.084 \pm 0.021)$ | $(0.018 \pm 0.020)$ |
| 48 | $(-0.573 \pm 0.033)$ | $(-0.408 \pm 0.030)$ | $(-0.072 \pm 0.020)$ | $(0.025 \pm 0.019)$ |
| 50 | $(-0.529 \pm 0.030)$ | $(-0.369 \pm 0.029)$ | $(-0.062 \pm 0.018)$ | $(0.025 \pm 0.017)$ |
| 52 | $(-0.478 \pm 0.029)$ | $(-0.338 \pm 0.025)$ | $(-0.055 \pm 0.017)$ | $(0.026 \pm 0.017)$ |
| 54 | $(-0.446 \pm 0.029)$ | $(-0.311 \pm 0.024)$ | $(-0.046 \pm 0.016)$ | $(0.026 \pm 0.014)$ |
| 56 | $(-0.409 \pm 0.025)$ | $(-0.288 \pm 0.022)$ | $(-0.040 \pm 0.015)$ | $(0.030 \pm 0.014)$ |
| 58 | $(-0.379 \pm 0.023)$ | $(-0.265 \pm 0.020)$ | $(-0.036 \pm 0.014)$ | $(0.029 \pm 0.013)$ |
| 60 | $(-0.353 \pm 0.022)$ | $(-0.248 \pm 0.019)$ | $(-0.031 \pm 0.013)$ | $(0.028 \pm 0.012)$ |
| 62 | $(-0.329 \pm 0.021)$ | $(-0.228 \pm 0.018)$ | $(-0.028 \pm 0.012)$ | $(0.029 \pm 0.012)$ |
| 64 | $(-0.309 \pm 0.018)$ | $(-0.214 \pm 0.017)$ | $(-0.026 \pm 0.012)$ | $(0.029 \pm 0.011)$ |
| 66 | $(-0.287 \pm 0.018)$ | $(-0.198 \pm 0.015)$ | $(-0.023 \pm 0.011)$ | $(0.028 \pm 0.010)$ |
| 68 | $(-0.270 \pm 0.017)$ | $(-0.185 \pm 0.015)$ | $(-0.020 \pm 0.009)$ | $(0.028 \pm 0.010)$ |
| 70 | $(-0.255 \pm 0.016)$ | $(-0.175 \pm 0.014)$ | $(-0.018 \pm 0.009)$ | $(0.028 \pm 0.009)$ |
| 72 | $(-0.240 \pm 0.015)$ | $(-0.167 \pm 0.013)$ | $(-0.015 \pm 0.009)$ | $(0.028 \pm 0.009)$ |
| 74 | $(-0.226 \pm 0.013)$ | $(-0.155 \pm 0.013)$ | $(-0.013 \pm 0.008)$ | $(0.027 \pm 0.009)$ |

## C. 3 Effective force between a pair of DNA with AuNPs

We calculated the error associated with the calculation of effective forces between a pair of parallel DNA in presence of AuNPs (see section 4.3.4).

The effective force writes as following:

$$
\begin{equation*}
\mathbf{F}=\mathbf{F}_{1}+\mathbf{F}_{2}+\mathbf{F}_{3}+\mathbf{F}_{4} \tag{C.15}
\end{equation*}
$$

The first term $\mathbf{F}_{1}$ is the direct Coulomb force exerted by each phosphate group of a DNA on the other phosphate groups of the other DNA:

$$
\begin{equation*}
\mathbf{F}_{1}=-\sum_{k=1}^{N_{p}}\left(\vec{\nabla}_{\mathbf{r}_{k}^{p}} \sum_{n=1}^{N_{p}} V_{p p}\left(\left|\mathbf{r}_{k}^{p}-\mathbf{r}_{n}^{p}\right|\right)\right) \tag{C.16}
\end{equation*}
$$

where $V_{i j}$ is the interaction potential as described in section 3.2.4. Given that the DNA are fixed during the simulation, this term has to be computed only once.

The second term $\mathbf{F}_{2}$ represents the Coulomb interaction between the DNA phosphate groups and the positive DNA counterions:

$$
\begin{equation*}
\mathbf{F}_{2}=-\sum_{k=1}^{N_{p}}\left(\left\langle\sum_{l=1}^{N_{c_{+}}} \vec{\nabla}_{\mathbf{r}_{k}^{p}} V_{p c_{+}}\left(\left|\mathbf{r}_{k}^{p}-\mathbf{r}_{l}^{c_{+}}\right|\right)\right\rangle\right) \tag{C.17}
\end{equation*}
$$

The third term $\mathbf{F}_{3}$ represents the Coulomb interactions between the DNA phosphate groups and the negative AuNP co-ions:

$$
\begin{equation*}
\mathbf{F}_{3}=-\sum_{k=1}^{N_{p}}\left(\left\langle\sum_{l=1}^{N_{c_{-}}} \vec{\nabla}_{\mathbf{r}_{k}^{p}} V_{p c_{-}}\left(\left|\mathbf{r}_{k}^{p}-\mathbf{r}_{l}^{c_{-}}\right|\right)\right\rangle\right) \tag{C.18}
\end{equation*}
$$

The fourth term $\mathbf{F}_{4}$ represents the Coulomb interactions between the DNA phosphate groups and the charges carried by the AuNPs:

$$
\begin{equation*}
\mathbf{F}_{4}=-\sum_{k=1}^{N_{p}}\left(\left\langle\sum_{l=1}^{N_{n+}} \vec{\nabla}_{\mathbf{r}_{k}^{p}} V_{p n_{+}}\left(\left|\mathbf{r}_{k}^{p}-\mathbf{r}_{l}^{n_{+}}\right|\right)\right\rangle\right) \tag{C.19}
\end{equation*}
$$

The forces are calculated along the $O x$ axis of the simulation box in the following way:

$$
\begin{equation*}
F=\mathbf{F} \cdot \mathbf{e}_{x}=\left(\mathbf{F}_{1}+\mathbf{F}_{2}+\mathbf{F}_{3}+\mathbf{F}_{4}\right) \cdot \mathbf{e}_{x}=F_{1}+F_{2}+F_{3}+F_{4} \tag{C.20}
\end{equation*}
$$

The force is normalized by a force $F_{o}=k_{b} \mathrm{~T} / \mathrm{P}$ where P is the pitch of the DNA.

Table C. 7 - Effective force exerted between the DNA pair in presence of 6 -AuNPs at $R_{+/-}=0.50$ corresponding to $\times 20$ nanoparticles in the system. There is also $\times 240$ DNA positive monovalent counterions and $\times 120$ negative monovalent AuNP co-ions. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.

| $\ell(\AA)$ | $\mathrm{F} / \mathrm{F}_{o}$ | $\mathrm{~F}_{1} / \mathrm{F}_{o}$ | $\mathrm{~F}_{2} / \mathrm{F}_{o}$ | $\mathrm{~F}_{3} / \mathrm{F}_{o}$ | $\mathrm{~F}_{4} / \mathrm{F}_{o}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 24 | $-22.9 \pm 4.9$ | $-229.6 \pm 0.0$ | $99.2 \pm 23.6$ | $3.2 \pm 0.1$ | $104.1 \pm 48.5$ |
| 30 | $15.9 \pm 12.8$ | $-178.0 \pm 0.0$ | $37.0 \pm 26.8$ | $3.6 \pm 0.1$ | $153.2 \pm 77.7$ |
| 36 | $14.6 \pm 3.5$ | $-143.5 \pm 0.0$ | $35.1 \pm 10.4$ | $4.3 \pm 0.2$ | $118.6 \pm 24.9$ |
| 42 | $4.6 \pm 1.6$ | $-117.9 \pm 0.0$ | $39.6 \pm 4.7$ | $4.0 \pm 0.3$ | $78.7 \pm 11.1$ |
| 48 | $2.9 \pm 1.8$ | $-97.8 \pm 0.0$ | $35.5 \pm 7.6$ | $3.6 \pm 0.3$ | $61.4 \pm 14.6$ |
| 54 | $0.3 \pm 4.0$ | $-81.4 \pm 0.0$ | $32.5 \pm 9.8$ | $3.2 \pm 0.8$ | $45.8 \pm 22.0$ |
| 60 | $0.7 \pm 3.6$ | $-67.5 \pm 0.0$ | $27.5 \pm 6.0$ | $2.7 \pm 0.9$ | $37.9 \pm 21.2$ |
| 66 | $0.0 \pm 2.9$ | $-55.4 \pm 0.0$ | $24.6 \pm 9.0$ | $2.8 \pm 0.5$ | $27.8 \pm 22.1$ |
| 72 | $-0.7 \pm 5.5$ | $-44.6 \pm 0.0$ | $19.9 \pm 14.6$ | $1.8 \pm 1.2$ | $22.1 \pm 40.8$ |
| 78 | $0.2 \pm 3.4$ | $-34.7 \pm 0.0$ | $14.7 \pm 12.6$ | $1.9 \pm 1.8$ | $18.2 \pm 22.2$ |
| 84 | $1.2 \pm 3.2$ | $-25.5 \pm 0.0$ | $9.0 \pm 6.5$ | $0.7 \pm 1.7$ | $16.9 \pm 24.8$ |
| 90 | $0.6 \pm 3.8$ | $-16.8 \pm 0.0$ | $7.2 \pm 6.2$ | $0.5 \pm 0.9$ | $9.6 \pm 23.8$ |
| 96 | $-0.3 \pm 4.9$ | $-8.3 \pm 0.0$ | $2.9 \pm 16.8$ | $0.7 \pm 1.7$ | $4.4 \pm 45.4$ |

Table C. 8 - Effective force exerted between the DNA pair in presence of 6 -AuNPs at $R_{+/-}=$ 1.00 corresponding to $\times 40$ nanoparticles in the system without small ions. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.

| $\ell(\AA)$ | $\mathrm{F} / \mathrm{F}_{o}$ | $\mathrm{~F}_{1} / \mathrm{F}_{o}$ | $\mathrm{~F}_{2} / \mathrm{F}_{o}$ | $\mathrm{~F}_{3} / \mathrm{F}_{o}$ | $\mathrm{~F}_{4} / \mathrm{F}_{o}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 24 | $-62.7 \pm 1.0$ | $-229.6 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $166.9 \pm 1.0$ |
| 30 | $10.3 \pm 0.7$ | $-178.0 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $188.3 \pm 0.7$ |
| 36 | $10.5 \pm 0.2$ | $-143.5 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $154.0 \pm 0.2$ |
| 42 | $1.9 \pm 0.2$ | $-117.9 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $119.7 \pm 0.2$ |
| 48 | $0.2 \pm 0.6$ | $-97.8 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $98.0 \pm 0.6$ |
| 54 | $-0.3 \pm 0.2$ | $-81.4 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $81.1 \pm 0.2$ |
| 60 | $0.0 \pm 0.5$ | $-67.5 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $67.6 \pm 0.5$ |
| 66 | $-0.1 \pm 0.5$ | $-55.4 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $55.3 \pm 0.5$ |
| 72 | $-0.2 \pm 0.3$ | $-44.6 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $44.5 \pm 0.3$ |
| 78 | $0.0 \pm 0.1$ | $-34.7 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $34.8 \pm 0.1$ |
| 84 | $0.2 \pm 0.2$ | $-25.5 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $25.7 \pm 0.2$ |
| 90 | $0.1 \pm 0.1$ | $-16.8 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $16.9 \pm 0.1$ |
| 96 | $0.0 \pm 0.4$ | $-8.3 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $8.3 \pm 0.4$ |

Table C. 9 - Effective force exerted between the DNA pair in presence of 6 -AuNPs at $R_{+/-}=1.00$ corresponding to $\times 40$ nanoparticles in the system. There is also $\times 240$ DNA positive monovalent counterions and $\times 240$ negative monovalent AuNP co-ions. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.

| $\ell(\AA)$ | $\mathrm{F} / \mathrm{F}_{o}$ | $\mathrm{~F}_{1} / \mathrm{F}_{o}$ | $\mathrm{~F}_{2} / \mathrm{F}_{o}$ | $\mathrm{~F}_{3} / \mathrm{F}_{o}$ | $\mathrm{~F}_{4} / \mathrm{F}_{o}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 24 | $-38.9 \pm 1.7$ | $-229.6 \pm 0.0$ | $63.0 \pm 4.9$ | $5.4 \pm 0.1$ | $122.3 \pm 6.0$ |
| 30 | $9.1 \pm 0.5$ | $-178.0 \pm 0.0$ | $13.7 \pm 0.5$ | $5.5 \pm 0.2$ | $167.9 \pm 1.3$ |
| 36 | $8.4 \pm 0.4$ | $-143.5 \pm 0.0$ | $14.7 \pm 1.0$ | $3.8 \pm 0.2$ | $133.3 \pm 2.7$ |
| 42 | $0.5 \pm 0.6$ | $-117.9 \pm 0.0$ | $16.1 \pm 0.8$ | $1.4 \pm 0.3$ | $100.8 \pm 1.7$ |
| 48 | $-1.2 \pm 0.5$ | $-97.8 \pm 0.0$ | $14.2 \pm 1.4$ | $-0.3 \pm 0.3$ | $82.7 \pm 3.5$ |
| 54 | $-0.9 \pm 0.5$ | $-81.4 \pm 0.0$ | $11.4 \pm 0.7$ | $-1.0 \pm 0.9$ | $70.0 \pm 3.4$ |
| 60 | $-0.8 \pm 0.6$ | $-67.5 \pm 0.0$ | $9.9 \pm 1.3$ | $-1.4 \pm 0.8$ | $58.3 \pm 4.3$ |
| 66 | $-0.4 \pm 0.2$ | $-55.4 \pm 0.0$ | $6.6 \pm 0.6$ | $-1.5 \pm 0.5$ | $49.9 \pm 0.5$ |
| 72 | $-0.3 \pm 0.3$ | $-44.6 \pm 0.0$ | $6.1 \pm 1.6$ | $-1.5 \pm 1.4$ | $39.6 \pm 3.0$ |
| 78 | $-0.1 \pm 0.3$ | $-34.7 \pm 0.0$ | $4.3 \pm 0.9$ | $-0.9 \pm 0.3$ | $31.4 \pm 2.2$ |
| 84 | $-0.1 \pm 0.4$ | $-25.5 \pm 0.0$ | $2.9 \pm 1.7$ | $-0.8 \pm 0.8$ | $23.4 \pm 2.3$ |
| 90 | $0.0 \pm 0.3$ | $-16.8 \pm 0.0$ | $2.1 \pm 0.8$ | $-0.5 \pm 1.1$ | $15.3 \pm 2.7$ |
| 96 | $0.4 \pm 0.6$ | $-8.3 \pm 0.0$ | $-0.3 \pm 1.0$ | $-0.1 \pm 0.4$ | $9.1 \pm 2.7$ |

Table C. 10 - Effective force exerted between the DNA pair in presence of 6 -AuNPs at $R_{+/-}=1.50$ corresponding to $\times 60$ nanoparticles in the system. There is also $\times 240$ DNA positive monovalent counterions and $\times 360$ negative monovalent AuNP co-ions. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.

| $\ell(\AA)$ | $\mathrm{F} / \mathrm{F}_{o}$ | $\mathrm{~F}_{1} / \mathrm{F}_{o}$ | $\mathrm{~F}_{2} / \mathrm{F}_{o}$ | $\mathrm{~F}_{3} / \mathrm{F}_{o}$ | $\mathrm{~F}_{4} / \mathrm{F}_{o}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 24 | $-42.4 \pm 0.9$ | $-229.6 \pm 0.0$ | $54.3 \pm 3.2$ | $8.1 \pm 0.1$ | $124.8 \pm 1.4$ |
| 30 | $8.0 \pm 0.5$ | $-178.0 \pm 0.0$ | $10.1 \pm 1.0$ | $7.6 \pm 0.1$ | $168.3 \pm 1.6$ |
| 36 | $7.3 \pm 0.6$ | $-143.5 \pm 0.0$ | $10.2 \pm 0.8$ | $4.3 \pm 0.4$ | $136.3 \pm 2.2$ |
| 42 | $-0.6 \pm 0.3$ | $-117.9 \pm 0.0$ | $12.2 \pm 0.5$ | $1.1 \pm 0.2$ | $104.0 \pm 0.6$ |
| 48 | $-1.6 \pm 0.3$ | $-97.8 \pm 0.0$ | $9.8 \pm 1.1$ | $-1.3 \pm 0.4$ | $87.7 \pm 1.9$ |
| 54 | $-1.4 \pm 0.6$ | $-81.4 \pm 0.0$ | $8.1 \pm 1.0$ | $-3.4 \pm 0.5$ | $75.3 \pm 3.2$ |
| 60 | $-0.9 \pm 0.3$ | $-67.5 \pm 0.0$ | $6.0 \pm 0.4$ | $-2.8 \pm 0.9$ | $63.3 \pm 2.5$ |
| 66 | $-0.2 \pm 0.1$ | $-55.4 \pm 0.0$ | $4.7 \pm 0.7$ | $-3.6 \pm 1.4$ | $54.0 \pm 2.1$ |
| 72 | $-0.1 \pm 0.2$ | $-44.6 \pm 0.0$ | $2.5 \pm 1.3$ | $-2.6 \pm 1.8$ | $44.6 \pm 3.6$ |
| 78 | $0.0 \pm 0.7$ | $-34.7 \pm 0.0$ | $1.5 \pm 0.6$ | $-2.7 \pm 1.0$ | $35.9 \pm 5.0$ |
| 84 | $-0.1 \pm 0.5$ | $-25.5 \pm 0.0$ | $1.1 \pm 0.6$ | $-1.2 \pm 1.5$ | $25.6 \pm 1.8$ |
| 90 | $-0.1 \pm 0.4$ | $-16.8 \pm 0.0$ | $1.3 \pm 1.3$ | $-0.8 \pm 0.6$ | $16.2 \pm 2.2$ |
| 96 | $-0.2 \pm 0.1$ | $-8.3 \pm 0.0$ | $0.3 \pm 0.4$ | $-0.2 \pm 1.1$ | $8.0 \pm 0.2$ |

Table C. 11 - Effective force exerted between the DNA pair in presence of 12 -AuNPs at $R_{+/-}=0.50$ corresponding to $\times 10$ nanoparticles in the system. There is also $\times 240$ DNA positive monovalent counterions and $\times 120$ negative monovalent AuNP co-ions. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.

| $\ell(\AA)$ | $\mathrm{F} / \mathrm{F}_{o}$ | $\mathrm{~F}_{1} / \mathrm{F}_{o}$ | $\mathrm{~F}_{2} / \mathrm{F}_{o}$ | $\mathrm{~F}_{3} / \mathrm{F}_{o}$ | $\mathrm{~F}_{4} / \mathrm{F}_{o}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 24 | $8.7 \pm 63.4$ | $-229.6 \pm 0.0$ | $56.7 \pm 128.6$ | $1.9 \pm 0.4$ | $179.7 \pm 393.2$ |
| 30 | $52.1 \pm 114.9$ | $-178.0 \pm 0.0$ | $15.3 \pm 101.7$ | $3.5 \pm 0.3$ | $211.7 \pm 444.2$ |
| 36 | $35.6 \pm 112.1$ | $-143.5 \pm 0.0$ | $18.6 \pm 89.4$ | $4.2 \pm 0.6$ | $156.3 \pm 428.6$ |
| 42 | $19.8 \pm 27.3$ | $-117.9 \pm 0.0$ | $22.4 \pm 27.3$ | $2.4 \pm 0.7$ | $112.8 \pm 122.5$ |
| 48 | $11.2 \pm 16.1$ | $-97.8 \pm 0.0$ | $28.2 \pm 26.4$ | $1.5 \pm 0.7$ | $79.2 \pm 94.2$ |
| 54 | $4.9 \pm 13.8$ | $-81.4 \pm 0.0$ | $29.8 \pm 19.9$ | $1.7 \pm 0.3$ | $54.7 \pm 74.1$ |
| 60 | $3.7 \pm 19.9$ | $-67.5 \pm 0.0$ | $25.7 \pm 23.9$ | $1.3 \pm 1.0$ | $44.1 \pm 80.6$ |
| 66 | $2.3 \pm 17.2$ | $-55.4 \pm 0.0$ | $22.9 \pm 17.4$ | $1.3 \pm 1.0$ | $33.6 \pm 84.5$ |
| 72 | $0.8 \pm 34.8$ | $-44.6 \pm 0.0$ | $19.5 \pm 37.1$ | $1.5 \pm 0.9$ | $24.4 \pm 78.8$ |
| 78 | $2.9 \pm 45.6$ | $-34.7 \pm 0.0$ | $13.1 \pm 47.9$ | $0.5 \pm 2.1$ | $24.0 \pm 90.9$ |
| 84 | $-0.5 \pm 26.9$ | $-25.5 \pm 0.0$ | $12.4 \pm 31.0$ | $1.2 \pm 1.3$ | $11.4 \pm 84.3$ |
| 90 | $4.1 \pm 26.0$ | $-16.8 \pm 0.0$ | $3.4 \pm 29.4$ | $-0.3 \pm 1.9$ | $17.8 \pm 74.3$ |
| 96 | $-0.9 \pm 19.8$ | $-8.3 \pm 0.0$ | $4.6 \pm 20.7$ | $0.5 \pm 0.7$ | $2.3 \pm 76.8$ |

Table C. 12 - Effective force exerted between the DNA pair in presence of 12 -AuNPs at $R_{+/-}=$ 1.00 corresponding to $\times 20$ nanoparticles in the system without small ions. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.

| $\ell(\AA)$ | $\mathrm{F} / \mathrm{F}_{o}$ | $\mathrm{~F}_{1} / \mathrm{F}_{o}$ | $\mathrm{~F}_{2} / \mathrm{F}_{o}$ | $\mathrm{~F}_{3} / \mathrm{F}_{o}$ | $\mathrm{~F}_{4} / \mathrm{F}_{o}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 24 | $-15.1 \pm 18.0$ | $-229.6 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $214.5 \pm 18.0$ |
| 30 | $62.1 \pm 15.3$ | $-178.0 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $240.1 \pm 15.3$ |
| 36 | $7.0 \pm 17.4$ | $-143.5 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $150.5 \pm 17.4$ |
| 42 | $-3.3 \pm 30.1$ | $-117.9 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $114.6 \pm 30.1$ |
| 48 | $-3.5 \pm 50.4$ | $-97.8 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $94.3 \pm 50.4$ |
| 54 | $-5.5 \pm 30.6$ | $-81.4 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $75.9 \pm 30.6$ |
| 60 | $-4.4 \pm 28.4$ | $-67.5 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $63.1 \pm 28.4$ |
| 66 | $-1.5 \pm 32.0$ | $-55.4 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $53.9 \pm 32.0$ |
| 72 | $1.3 \pm 21.7$ | $-44.6 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $45.9 \pm 21.7$ |
| 78 | $0.9 \pm 11.1$ | $-34.7 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $35.7 \pm 11.1$ |
| 84 | $-1.0 \pm 13.5$ | $-25.5 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $24.5 \pm 13.5$ |
| 90 | $0.3 \pm 10.1$ | $-16.8 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $17.1 \pm 10.1$ |
| 96 | $-0.9 \pm 12.1$ | $-8.3 \pm 0.0$ | $0.0 \pm 0.0$ | $0.0 \pm 0.0$ | $7.4 \pm 12.1$ |

Table C. 13 - Effective force exerted between the DNA pair in presence of 12 -AuNPs at $R_{+/-}=1.00$ corresponding to $\times 20$ nanoparticles in the system. There is also $\times 240$ DNA positive monovalent counterions and $\times 240$ negative monovalent AuNP co-ions. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.

| $\ell(\AA)$ | $\mathrm{F} / \mathrm{F}_{o}$ | $\mathrm{~F}_{1} / \mathrm{F}_{o}$ | $\mathrm{~F}_{2} / \mathrm{F}_{o}$ | $\mathrm{~F}_{3} / \mathrm{F}_{o}$ | $\mathrm{~F}_{4} / \mathrm{F}_{o}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 24 | $-13.8 \pm 32.9$ | $-229.6 \pm 0.0$ | $18.0 \pm 9.4$ | $4.3 \pm 2.1$ | $193.4 \pm 98.5$ |
| 30 | $35.3 \pm 148.6$ | $-178.0 \pm 0.0$ | $3.3 \pm 37.4$ | $3.9 \pm 8.7$ | $206.2 \pm 113.2$ |
| 36 | $14.2 \pm 18.7$ | $-143.5 \pm 0.0$ | $9.8 \pm 7.4$ | $5.1 \pm 1.7$ | $142.8 \pm 66.2$ |
| 42 | $-4.0 \pm 4.4$ | $-117.9 \pm 0.0$ | $14.8 \pm 1.7$ | $-0.6 \pm 0.9$ | $99.6 \pm 16.0$ |
| 48 | $-4.5 \pm 3.5$ | $-97.8 \pm 0.0$ | $15.2 \pm 2.7$ | $-4.0 \pm 2.0$ | $82.1 \pm 20.2$ |
| 54 | $-4.3 \pm 4.2$ | $-81.4 \pm 0.0$ | $12.9 \pm 5.6$ | $-4.7 \pm 1.5$ | $68.9 \pm 26.3$ |
| 60 | $-1.9 \pm 4.3$ | $-67.5 \pm 0.0$ | $10.6 \pm 3.8$ | $-5.3 \pm 2.1$ | $60.2 \pm 23.9$ |
| 66 | $-1.4 \pm 6.6$ | $-55.4 \pm 0.0$ | $8.1 \pm 4.6$ | $-4.9 \pm 1.7$ | $50.9 \pm 30.1$ |
| 72 | $-1.7 \pm 5.0$ | $-44.6 \pm 0.0$ | $6.3 \pm 2.5$ | $-3.5 \pm 2.5$ | $40.1 \pm 18.8$ |
| 78 | $-1.6 \pm 8.9$ | $-34.7 \pm 0.0$ | $5.7 \pm 14.4$ | $-3.3 \pm 1.2$ | $30.7 \pm 25.2$ |
| 84 | $-1.4 \pm 6.6$ | $-25.5 \pm 0.0$ | $4.3 \pm 5.0$ | $-2.5 \pm 2.5$ | $22.3 \pm 30.6$ |
| 90 | $-1.1 \pm 6.0$ | $-16.8 \pm 0.0$ | $3.6 \pm 3.5$ | $-2.1 \pm 1.2$ | $14.2 \pm 23.2$ |
| 96 | $-1.2 \pm 7.2$ | $-8.3 \pm 0.0$ | $2.3 \pm 10.2$ | $-0.1 \pm 2.1$ | $4.9 \pm 21.7$ |

Table C. 14 - Effective force exerted between the DNA pair in presence of 12 -AuNPs at $R_{+/-}=1.50$ corresponding to $\times 30$ nanoparticles in the system. There is also $\times 240$ DNA positive monovalent counterions and $\times 360$ negative monovalent AuNP co-ions. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles and the force is printed in a file every MC cycle to calculate statistics.

| $\ell(\AA)$ | $\mathrm{F} / \mathrm{F}_{o}$ | $\mathrm{~F}_{1} / \mathrm{F}_{o}$ | $\mathrm{~F}_{2} / \mathrm{F}_{o}$ | $\mathrm{~F}_{3} / \mathrm{F}_{o}$ | $\mathrm{~F}_{4} / \mathrm{F}_{o}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 24 | $-19.7 \pm 8.9$ | $-229.6 \pm 0.0$ | $3.8 \pm 5.8$ | $-1.3 \pm 7.8$ | $207.4 \pm 31.3$ |
| 30 | $22.0 \pm 7.0$ | $-178.0 \pm 0.0$ | $-3.7 \pm 0.8$ | $-6.1 \pm 4.7$ | $209.8 \pm 25.3$ |
| 36 | $6.2 \pm 3.4$ | $-143.5 \pm 0.0$ | $-2.1 \pm 0.3$ | $-11.2 \pm 3.6$ | $163.1 \pm 13.8$ |
| 42 | $-8.4 \pm 2.7$ | $-117.9 \pm 0.0$ | $-0.6 \pm 0.3$ | $-22.2 \pm 2.6$ | $132.2 \pm 7.1$ |
| 48 | $-7.2 \pm 3.6$ | $-97.8 \pm 0.0$ | $-1.2 \pm 1.3$ | $-25.5 \pm 2.4$ | $117.3 \pm 8.0$ |
| 54 | $-5.0 \pm 2.2$ | $-81.4 \pm 0.0$ | $-2.0 \pm 0.5$ | $-25.3 \pm 2.1$ | $103.7 \pm 9.8$ |
| 60 | $-3.5 \pm 3.2$ | $-67.5 \pm 0.0$ | $-2.5 \pm 0.7$ | $-22.4 \pm 2.5$ | $88.9 \pm 8.8$ |
| 66 | $-2.5 \pm 1.9$ | $-55.4 \pm 0.0$ | $-2.6 \pm 0.2$ | $-19.3 \pm 0.6$ | $74.9 \pm 5.0$ |
| 72 | $-1.4 \pm 2.1$ | $-44.6 \pm 0.0$ | $-2.9 \pm 0.2$ | $-16.3 \pm 0.7$ | $62.4 \pm 6.4$ |
| 78 | $-0.9 \pm 1.5$ | $-34.7 \pm 0.0$ | $-2.6 \pm 0.3$ | $-12.9 \pm 1.2$ | $49.5 \pm 7.5$ |
| 84 | $0.0 \pm 3.3$ | $-25.5 \pm 0.0$ | $-2.2 \pm 0.7$ | $-9.8 \pm 1.4$ | $37.5 \pm 8.6$ |
| 90 | $-1.1 \pm 1.2$ | $-16.8 \pm 0.0$ | $-1.1 \pm 0.2$ | $-6.0 \pm 0.8$ | $22.8 \pm 5.1$ |
| 96 | $0.4 \pm 1.9$ | $-8.3 \pm 0.0$ | $-1.0 \pm 0.4$ | $-3.4 \pm 0.9$ | $13.1 \pm 7.3$ |

## C. 4 Osmotic pressure in DNA lattices with AuNPs

We calculated the error associated with the calculation of osmotic pressure in a hexagonal DNA bundle condensed with AuNPs (see section 4.3.5) that can exchange monovalent salt with supernatant phase.

The mechanical stability of the bundle is given by the difference of osmotic pressure in the bundle and in the supernatant phase:

$$
\begin{equation*}
\Pi_{\text {rel }}=\Pi_{\text {bundle }}-\Pi_{b u l k} \tag{C.21}
\end{equation*}
$$

Table C. 15 - Osmotic pressure calculated in a square lattice of DNA condensed with $\times 406$-AuNPs without salt $\left(R_{+/-}=1.00\right)$. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles to calculate statistics.

| $\ell(\AA)$ | $\Pi_{\text {rel }}(\mathrm{atm})$ | $\Pi_{\text {bundle }}(\mathrm{atm})$. | $\Pi_{\text {bulk }}(\mathrm{atm})$. |
| :---: | :---: | :---: | :---: |
| 32 | $(-7.888 \pm 0.225)$ | $(-7.888 \pm 0.225)$ | $(0.000 \pm 0.000)$ |
| 34 | $(-6.594 \pm 0.212)$ | $(-6.594 \pm 0.212)$ | $(0.000 \pm 0.000)$ |
| 36 | $(-5.386 \pm 0.189)$ | $(-5.386 \pm 0.189)$ | $(0.000 \pm 0.000)$ |
| 38 | $(-4.188 \pm 0.167)$ | $(-4.188 \pm 0.167)$ | $(0.000 \pm 0.000)$ |
| 40 | $(-3.188 \pm 0.153)$ | $(-3.188 \pm 0.153)$ | $(0.000 \pm 0.000)$ |
| 42 | $(-2.509 \pm 0.149)$ | $(-2.509 \pm 0.149)$ | $(0.000 \pm 0.000)$ |
| 44 | $(-2.043 \pm 0.145)$ | $(-2.043 \pm 0.145)$ | $(0.000 \pm 0.000)$ |
| 46 | $(-1.702 \pm 0.138)$ | $(-1.702 \pm 0.138)$ | $(0.000 \pm 0.000)$ |
| 48 | $(-1.440 \pm 0.131)$ | $(-1.440 \pm 0.131)$ | $(0.000 \pm 0.000)$ |
| 50 | $(-1.220 \pm 0.125)$ | $(-1.220 \pm 0.125)$ | $(0.000 \pm 0.000)$ |
| 52 | $(-1.045 \pm 0.117)$ | $(-1.045 \pm 0.117)$ | $(0.000 \pm 0.000)$ |
| 54 | $(-0.902 \pm 0.110)$ | $(-0.902 \pm 0.110)$ | $(0.000 \pm 0.000)$ |
| 56 | $(-0.779 \pm 0.104)$ | $(-0.779 \pm 0.104)$ | $(0.000 \pm 0.000)$ |
| 58 | $(-0.672 \pm 0.098)$ | $(-0.672 \pm 0.098)$ | $(0.000 \pm 0.000)$ |
| 60 | $(-0.581 \pm 0.093)$ | $(-0.581 \pm 0.093)$ | $(0.000 \pm 0.000)$ |
| 62 | $(-0.526 \pm 0.087)$ | $(-0.526 \pm 0.087)$ | $(0.000 \pm 0.000)$ |
| 64 | $(-0.449 \pm 0.085)$ | $(-0.449 \pm 0.085)$ | $(0.000 \pm 0.000)$ |
| 66 | $(-0.393 \pm 0.079)$ | $(-0.393 \pm 0.079)$ | $(0.000 \pm 0.000)$ |
| 68 | $(-0.349 \pm 0.075)$ | $(-0.349 \pm 0.075)$ | $(0.000 \pm 0.000)$ |
| 70 | $(-0.302 \pm 0.071)$ | $(-0.302 \pm 0.071)$ | $(0.000 \pm 0.000)$ |
| 72 | $(-0.254 \pm 0.070)$ | $(-0.254 \pm 0.070)$ | $(0.000 \pm 0.000)$ |
| 74 | $(-0.206 \pm 0.064)$ | $(-0.206 \pm 0.064)$ | $(0.000 \pm 0.000)$ |
| 76 | $(-0.186 \pm 0.063)$ | $(-0.186 \pm 0.063)$ | $(0.000 \pm 0.000)$ |
| 78 | $(-0.134 \pm 0.065)$ | $(-0.134 \pm 0.065)$ | $(0.000 \pm 0.000)$ |
| 80 | $(-0.125 \pm 0.055)$ | $(-0.125 \pm 0.055)$ | $(0.000 \pm 0.000)$ |
| 82 | $(-0.106 \pm 0.053)$ | $(-0.106 \pm 0.053)$ | $(0.000 \pm 0.000)$ |
| 84 | $(-0.089 \pm 0.051)$ | $(-0.089 \pm 0.051)$ | $(0.000 \pm 0.000)$ |
| 86 | $(-0.071 \pm 0.050)$ | $(-0.071 \pm 0.050)$ | $(0.000 \pm 0.000)$ |
| 88 | $(-0.055 \pm 0.047)$ | $(-0.055 \pm 0.047)$ | $(0.000 \pm 0.000)$ |
| 90 | $(-0.073 \pm 0.047)$ | $(-0.073 \pm 0.047)$ | $(0.000 \pm 0.000)$ |
| 92 | $(-0.039 \pm 0.049)$ | $(-0.039 \pm 0.049)$ | $(0.000 \pm 0.000)$ |
| 94 | $(-0.042 \pm 0.043)$ | $(-0.042 \pm 0.043)$ | $(0.000 \pm 0.000)$ |
| 96 | $(-0.023 \pm 0.041)$ | $(-0.023 \pm 0.041)$ | $(0.000 \pm 0.000)$ |
| 98 | $(-0.012 \pm 0.039)$ | $(-0.012 \pm 0.039)$ | $(0.000 \pm 0.000)$ |
| 100 | $(-0.025 \pm 0.038)$ | $(-0.025 \pm 0.038)$ | $(0.000 \pm 0.000)$ |
| 102 | $(-0.018 \pm 0.037)$ | $(-0.018 \pm 0.037)$ | $(0.000 \pm 0.000)$ |
|  |  |  |  |

Table C. 16 - Osmotic pressure calculated in a hexagonal lattice of DNA condensed with $\times 406$ AuNPs without salt $\left(R_{+/-}=1.00\right)$. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles to calculate statistics.

| $\ell(\AA)$ | $\Pi_{\text {rel }}($ atm. $)$ | $\Pi_{\text {bundle }}($ atm. $)$ | $\Pi_{\text {bulk }}(\mathrm{atm})$. |
| :---: | :---: | :---: | :---: |
| 32 | $(-7.118 \pm 0.278)$ | $(-7.118 \pm 0.278)$ | $(0.000 \pm 0.000)$ |
| 34 | $(-5.031 \pm 0.275)$ | $(-5.031 \pm 0.275)$ | $(0.000 \pm 0.000)$ |
| 36 | $(-3.347 \pm 0.272)$ | $(-3.347 \pm 0.272)$ | $(0.000 \pm 0.000)$ |
| 38 | $(-2.118 \pm 0.268)$ | $(-2.118 \pm 0.268)$ | $(0.000 \pm 0.000)$ |
| 40 | $(-1.339 \pm 0.273)$ | $(-1.339 \pm 0.273)$ | $(0.000 \pm 0.000)$ |
| 42 | $(-0.963 \pm 0.276)$ | $(-0.963 \pm 0.276)$ | $(0.000 \pm 0.000)$ |
| 44 | $(-0.777 \pm 0.267)$ | $(-0.777 \pm 0.267)$ | $(0.000 \pm 0.000)$ |
| 46 | $(-0.650 \pm 0.244)$ | $(-0.650 \pm 0.244)$ | $(0.000 \pm 0.000)$ |
| 48 | $(-0.554 \pm 0.224)$ | $(-0.554 \pm 0.224)$ | $(0.000 \pm 0.000)$ |
| 50 | $(-0.471 \pm 0.206)$ | $(-0.471 \pm 0.206)$ | $(0.000 \pm 0.000)$ |
| 52 | $(-0.403 \pm 0.190)$ | $(-0.403 \pm 0.190)$ | $(0.000 \pm 0.000)$ |
| 54 | $(-0.355 \pm 0.174)$ | $(-0.355 \pm 0.174)$ | $(0.000 \pm 0.000)$ |
| 56 | $(-0.320 \pm 0.153)$ | $(-0.320 \pm 0.153)$ | $(0.000 \pm 0.000)$ |
| 58 | $(-0.270 \pm 0.146)$ | $(-0.270 \pm 0.146)$ | $(0.000 \pm 0.000)$ |
| 60 | $(-0.256 \pm 0.133)$ | $(-0.256 \pm 0.133)$ | $(0.000 \pm 0.000)$ |
| 62 | $(-0.207 \pm 0.126)$ | $(-0.207 \pm 0.126)$ | $(0.000 \pm 0.000)$ |
| 64 | $(-0.197 \pm 0.118)$ | $(-0.197 \pm 0.118)$ | $(0.000 \pm 0.000)$ |
| 66 | $(-0.167 \pm 0.107)$ | $(-0.167 \pm 0.107)$ | $(0.000 \pm 0.000)$ |
| 68 | $(-0.152 \pm 0.099)$ | $(-0.152 \pm 0.099)$ | $(0.000 \pm 0.000)$ |
| 70 | $(-0.129 \pm 0.094)$ | $(-0.129 \pm 0.094)$ | $(0.000 \pm 0.000)$ |
| 72 | $(-0.090 \pm 0.086)$ | $(-0.090 \pm 0.086)$ | $(0.000 \pm 0.000)$ |
| 74 | $(-0.095 \pm 0.085)$ | $(-0.095 \pm 0.085)$ | $(0.000 \pm 0.000)$ |
| 76 | $(-0.049 \pm 0.078)$ | $(-0.049 \pm 0.078)$ | $(0.000 \pm 0.000)$ |
| 78 | $(-0.067 \pm 0.075)$ | $(-0.067 \pm 0.075)$ | $(0.000 \pm 0.000)$ |
| 80 | $(-0.016 \pm 0.071)$ | $(-0.016 \pm 0.071)$ | $(0.000 \pm 0.000)$ |
| 82 | $(-0.030 \pm 0.067)$ | $(-0.030 \pm 0.067)$ | $(0.000 \pm 0.000)$ |
| 84 | $(0.014 \pm 0.068)$ | $(0.014 \pm 0.068)$ | $(0.000 \pm 0.000)$ |
| 86 | $(-0.022 \pm 0.061)$ | $(-0.022 \pm 0.061)$ | $(0.000 \pm 0.000)$ |
| 88 | $(-0.007 \pm 0.058)$ | $(-0.007 \pm 0.058)$ | $(0.000 \pm 0.000)$ |
| 90 | $(-0.002 \pm 0.056)$ | $(-0.002 \pm 0.056)$ | $(0.000 \pm 0.000)$ |
| 92 | $(0.011 \pm 0.056)$ | $(0.011 \pm 0.056)$ | $(0.000 \pm 0.000)$ |
| 94 | $(0.037 \pm 0.052)$ | $(0.037 \pm 0.052)$ | $(0.000 \pm 0.000)$ |
| 96 | $(0.045 \pm 0.049)$ | $(0.045 \pm 0.049)$ | $(0.000 \pm 0.000)$ |
| 98 | $(0.051 \pm 0.048)$ | $(0.051 \pm 0.048)$ | $(0.000 \pm 0.000)$ |
| 100 | $(0.030 \pm 0.046)$ | $(0.030 \pm 0.046)$ | $(0.000 \pm 0.000)$ |
| 102 | $(0.071 \pm 0.046)$ | $(0.071 \pm 0.046)$ | $(0.000 \pm 0.000)$ |
|  |  |  |  |

Table C. 17 - Osmotic pressure calculated in a square lattice of DNA condensed with $\times 2012$-AuNPs without salt $\left(R_{+/-}=1.00\right)$. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles to calculate statistics.

| $\ell(\AA)$ | $\Pi_{\text {rel }}(\mathrm{atm})$ | $\Pi_{\text {bundle }}(\mathrm{atm})$. | $\Pi_{\text {bulk }}(\mathrm{atm})$. |
| :---: | :---: | :---: | :---: |
| 32 | $(-9.257 \pm 0.181)$ | $(-9.257 \pm 0.181)$ | $(0.000 \pm 0.000)$ |
| 34 | $(-7.102 \pm 0.205)$ | $(-7.102 \pm 0.205)$ | $(0.000 \pm 0.000)$ |
| 36 | $(-5.514 \pm 0.204)$ | $(-5.514 \pm 0.204)$ | $(0.000 \pm 0.000)$ |
| 38 | $(-4.007 \pm 0.164)$ | $(-4.007 \pm 0.164)$ | $(0.000 \pm 0.000)$ |
| 40 | $(-2.807 \pm 0.161)$ | $(-2.807 \pm 0.161)$ | $(0.000 \pm 0.000)$ |
| 42 | $(-1.904 \pm 0.155)$ | $(-1.904 \pm 0.155)$ | $(0.000 \pm 0.000)$ |
| 44 | $(-1.333 \pm 0.146)$ | $(-1.333 \pm 0.146)$ | $(0.000 \pm 0.000)$ |
| 46 | $(-1.040 \pm 0.210)$ | $(-1.040 \pm 0.210)$ | $(0.000 \pm 0.000)$ |
| 48 | $(-0.876 \pm 0.221)$ | $(-0.876 \pm 0.221)$ | $(0.000 \pm 0.000)$ |
| 50 | $(-0.458 \pm 0.130)$ | $(-0.458 \pm 0.130)$ | $(0.000 \pm 0.000)$ |
| 52 | $(-0.631 \pm 0.140)$ | $(-0.631 \pm 0.140)$ | $(0.000 \pm 0.000)$ |
| 54 | $(-0.382 \pm 0.167)$ | $(-0.382 \pm 0.167)$ | $(0.000 \pm 0.000)$ |
| 56 | $(-0.121 \pm 0.100)$ | $(-0.121 \pm 0.100)$ | $(0.000 \pm 0.000)$ |
| 58 | $(0.211 \pm 0.107)$ | $(0.211 \pm 0.107)$ | $(0.000 \pm 0.000)$ |
| 60 | $(0.159 \pm 0.085)$ | $(0.159 \pm 0.085)$ | $(0.000 \pm 0.000)$ |
| 62 | $(0.367 \pm 0.078)$ | $(0.367 \pm 0.078)$ | $(0.000 \pm 0.000)$ |
| 64 | $(0.113 \pm 0.076)$ | $(0.113 \pm 0.076)$ | $(0.000 \pm 0.000)$ |
| 66 | $(0.127 \pm 0.073)$ | $(0.127 \pm 0.073)$ | $(0.000 \pm 0.000)$ |
| 68 | $(0.139 \pm 0.069)$ | $(0.139 \pm 0.069)$ | $(0.000 \pm 0.000)$ |
| 70 | $(0.191 \pm 0.066)$ | $(0.191 \pm 0.066)$ | $(0.000 \pm 0.000)$ |
| 72 | $(0.458 \pm 0.075)$ | $(0.458 \pm 0.075)$ | $(0.000 \pm 0.000)$ |
| 74 | $(-0.014 \pm 0.059)$ | $(-0.014 \pm 0.059)$ | $(0.000 \pm 0.000)$ |
| 76 | $(0.002 \pm 0.054)$ | $(0.002 \pm 0.054)$ | $(0.000 \pm 0.000)$ |
| 78 | $(0.199 \pm 0.056)$ | $(0.199 \pm 0.056)$ | $(0.000 \pm 0.000)$ |
| 80 | $(0.206 \pm 0.052)$ | $(0.206 \pm 0.052)$ | $(0.000 \pm 0.000)$ |
| 82 | $(0.021 \pm 0.049)$ | $(0.021 \pm 0.049)$ | $(0.000 \pm 0.000)$ |
| 84 | $(0.053 \pm 0.044)$ | $(0.053 \pm 0.044)$ | $(0.000 \pm 0.000)$ |
| 86 | $(0.046 \pm 0.045)$ | $(0.046 \pm 0.045)$ | $(0.000 \pm 0.000)$ |
| 88 | $(0.021 \pm 0.043)$ | $(0.021 \pm 0.043)$ | $(0.000 \pm 0.000)$ |
| 90 | $(0.021 \pm 0.041)$ | $(0.021 \pm 0.041)$ | $(0.000 \pm 0.000)$ |
| 92 | $(0.021 \pm 0.040)$ | $(0.021 \pm 0.040)$ | $(0.000 \pm 0.000)$ |
| 94 | $(0.007 \pm 0.038)$ | $(0.007 \pm 0.038)$ | $(0.000 \pm 0.000)$ |
| 96 | $(0.039 \pm 0.038)$ | $(0.039 \pm 0.038)$ | $(0.000 \pm 0.000)$ |
| 98 | $(0.033 \pm 0.037)$ | $(0.033 \pm 0.037)$ | $(0.000 \pm 0.000)$ |
| 100 | $(0.084 \pm 0.033)$ | $(0.084 \pm 0.033)$ | $(0.000 \pm 0.000)$ |
| 102 | $(0.021 \pm 0.035)$ | $(0.021 \pm 0.035)$ | $(0.000 \pm 0.000)$ |
|  |  |  |  |

Table C. 18 - Osmotic pressure calculated in a hexagonal lattice of DNA condensed with $\times 2012$ AuNPs without salt $\left(R_{+/-}=1.00\right)$. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles to calculate statistics.

| $\ell(\AA)$ | $\Pi_{\text {rel }}(\mathrm{atm})$ | $\Pi_{\text {bundle }}(\mathrm{atm})$. | $\Pi_{\text {bulk }}(\mathrm{atm})$. |
| :---: | :---: | :---: | :---: |
| 32 | $(-9.257 \pm 0.313)$ | $(-9.257 \pm 0.313)$ | $(0.000 \pm 0.000)$ |
| 34 | $(-6.486 \pm 0.399)$ | $(-6.486 \pm 0.399)$ | $(0.000 \pm 0.000)$ |
| 36 | $(-4.379 \pm 0.388)$ | $(-4.379 \pm 0.388)$ | $(0.000 \pm 0.000)$ |
| 38 | $(-2.649 \pm 0.395)$ | $(-2.649 \pm 0.395)$ | $(0.000 \pm 0.000)$ |
| 40 | $(-1.544 \pm 0.385)$ | $(-1.544 \pm 0.385)$ | $(0.000 \pm 0.000)$ |
| 42 | $(-0.916 \pm 0.401)$ | $(-0.916 \pm 0.401)$ | $(0.000 \pm 0.000)$ |
| 44 | $(-0.499 \pm 0.364)$ | $(-0.499 \pm 0.364)$ | $(0.000 \pm 0.000)$ |
| 46 | $(-0.259 \pm 0.326)$ | $(-0.259 \pm 0.326)$ | $(0.000 \pm 0.000)$ |
| 48 | $(-0.211 \pm 0.311)$ | $(-0.211 \pm 0.311)$ | $(0.000 \pm 0.000)$ |
| 50 | $(-0.031 \pm 0.260)$ | $(-0.031 \pm 0.260)$ | $(0.000 \pm 0.000)$ |
| 52 | $(-0.084 \pm 0.214)$ | $(-0.084 \pm 0.214)$ | $(0.000 \pm 0.000)$ |
| 54 | $(0.014 \pm 0.197)$ | $(0.014 \pm 0.197)$ | $(0.000 \pm 0.000)$ |
| 56 | $(0.058 \pm 0.173)$ | $(0.058 \pm 0.173)$ | $(0.000 \pm 0.000)$ |
| 58 | $(0.061 \pm 0.149)$ | $(0.061 \pm 0.149)$ | $(0.000 \pm 0.000)$ |
| 60 | $(0.050 \pm 0.127)$ | $(0.050 \pm 0.127)$ | $(0.000 \pm 0.000)$ |
| 62 | $(0.284 \pm 0.145)$ | $(0.284 \pm 0.145)$ | $(0.000 \pm 0.000)$ |
| 64 | $(0.090 \pm 0.110)$ | $(0.090 \pm 0.110)$ | $(0.000 \pm 0.000)$ |
| 66 | $(0.091 \pm 0.175)$ | $(0.091 \pm 0.175)$ | $(0.000 \pm 0.000)$ |
| 68 | $(0.109 \pm 0.088)$ | $(0.109 \pm 0.088)$ | $(0.000 \pm 0.000)$ |
| 70 | $(0.525 \pm 0.217)$ | $(0.525 \pm 0.217)$ | $(0.000 \pm 0.000)$ |
| 72 | $(0.090 \pm 0.079)$ | $(0.090 \pm 0.079)$ | $(0.000 \pm 0.000)$ |
| 74 | $(0.621 \pm 0.075)$ | $(0.621 \pm 0.075)$ | $(0.000 \pm 0.000)$ |
| 76 | $(0.328 \pm 0.079)$ | $(0.328 \pm 0.079)$ | $(0.000 \pm 0.000)$ |
| 78 | $(0.329 \pm 0.078)$ | $(0.329 \pm 0.078)$ | $(0.000 \pm 0.000)$ |
| 80 | $(0.133 \pm 0.063)$ | $(0.133 \pm 0.063)$ | $(0.000 \pm 0.000)$ |
| 82 | $(0.340 \pm 0.063)$ | $(0.340 \pm 0.063)$ | $(0.000 \pm 0.000)$ |
| 84 | $(0.406 \pm 0.077)$ | $(0.406 \pm 0.077)$ | $(0.000 \pm 0.000)$ |
| 86 | $(0.565 \pm 0.057)$ | $(0.565 \pm 0.057)$ | $(0.000 \pm 0.000)$ |
| 88 | $(0.311 \pm 0.054)$ | $(0.311 \pm 0.054)$ | $(0.000 \pm 0.000)$ |
| 90 | $(0.146 \pm 0.049)$ | $(0.146 \pm 0.049)$ | $(0.000 \pm 0.000)$ |
| 92 | $(0.146 \pm 0.045)$ | $(0.146 \pm 0.045)$ | $(0.000 \pm 0.000)$ |
| 94 | $(0.299 \pm 0.049)$ | $(0.299 \pm 0.049)$ | $(0.000 \pm 0.000)$ |
| 96 | $(0.293 \pm 0.047)$ | $(0.293 \pm 0.047)$ | $(0.000 \pm 0.000)$ |
| 98 | $(0.284 \pm 0.045)$ | $(0.284 \pm 0.045)$ | $(0.000 \pm 0.000)$ |
| 100 | $(0.288 \pm 0.043)$ | $(0.288 \pm 0.043)$ | $(0.000 \pm 0.000)$ |
| 102 | $(0.284 \pm 0.041)$ | $(0.284 \pm 0.041)$ | $(0.000 \pm 0.000)$ |
|  |  |  |  |

Given that plenty of osmotic pressure calculations has been performed, we display only the error for osmotic pressure calculations in a hexagonal lattice of DNA condensed with $\times 406$-AuNP at constant salt concentration $\mathrm{c}_{\text {salt }}=30,60$ or $120 \mathrm{mMol} / \mathrm{l}$.

Table C. 19 - Osmotic pressure calculated in a hexagonal lattice of DNA condensed with $\times 406$ AuNPs $\left(R_{+/-}=1.00\right)$ with salt concentration of $\mathrm{c}_{\text {salt }}=30 \mathrm{mMol} / \mathrm{l}$. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles to calculate statistics.

| $\ell(\AA)$ | $\Pi_{\text {rel }}$ (atm.) | $\Pi_{\text {bundle }}(\mathrm{atm})$. | $\Pi_{\text {bulk }}(\mathrm{atm})$. |
| :---: | :---: | :---: | :---: |
| 32 | $(-6.721 \pm 0.302)$ | $(-6.157 \pm 0.029)$ | $(0.563 \pm 0.273)$ |
| 36 | $(-2.848 \pm 0.224)$ | $(-2.440 \pm 0.028)$ | $(0.408 \pm 0.196)$ |
| 40 | $(-1.056 \pm 0.177)$ | $(-0.745 \pm 0.028)$ | $(0.311 \pm 0.149)$ |
| 44 | $(-0.126 \pm 0.158)$ | $(0.148 \pm 0.026)$ | $(0.275 \pm 0.132)$ |
| 48 | $(0.119 \pm 0.148)$ | $(0.382 \pm 0.022)$ | $(0.262 \pm 0.126)$ |
| 52 | $(0.321 \pm 0.147)$ | $(0.592 \pm 0.019)$ | $(0.271 \pm 0.128)$ |
| 54 | $(0.368 \pm 0.144)$ | $(0.640 \pm 0.017)$ | $(0.271 \pm 0.127)$ |
| 58 | $(0.457 \pm 0.147)$ | $(0.752 \pm 0.015)$ | $(0.294 \pm 0.132)$ |
| 62 | $(0.520 \pm 0.146)$ | $(0.842 \pm 0.012)$ | $(0.321 \pm 0.134)$ |
| 66 | $(0.548 \pm 0.146)$ | $(0.902 \pm 0.011)$ | $(0.353 \pm 0.135)$ |
| 70 | $(0.572 \pm 0.149)$ | $(0.977 \pm 0.010)$ | $(0.405 \pm 0.139)$ |
| 74 | $(0.580 \pm 0.145)$ | $(1.019 \pm 0.008)$ | $(0.439 \pm 0.137)$ |
| 78 | $(0.578 \pm 0.146)$ | $(1.067 \pm 0.008)$ | $(0.489 \pm 0.138)$ |
| 82 | $(0.584 \pm 0.144)$ | $(1.114 \pm 0.007)$ | $(0.530 \pm 0.137)$ |
| 86 | $(0.590 \pm 0.142)$ | $(1.173 \pm 0.006)$ | $(0.582 \pm 0.136)$ |
| 90 | $(0.580 \pm 0.140)$ | $(1.194 \pm 0.006)$ | $(0.614 \pm 0.134)$ |
| 94 | $(0.552 \pm 0.140)$ | $(1.225 \pm 0.005)$ | $(0.673 \pm 0.135)$ |
| 98 | $(0.544 \pm 0.137)$ | $(1.249 \pm 0.005)$ | $(0.705 \pm 0.132)$ |
| 102 | $(0.531 \pm 0.135)$ | $(1.269 \pm 0.005)$ | $(0.737 \pm 0.130)$ |

Table C. 20 - Osmotic pressure calculated in a hexagonal lattice of DNA condensed with $\times 406$ AuNPs $\left(R_{+/-}=1.00\right)$ with salt concentration of $\mathrm{c}_{\text {salt }}=60 \mathrm{mMol} / \mathrm{l}$. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles to calculate statistics.

| $\ell(\AA)$ | $\Pi_{\text {rel }}$ (atm.) | $\Pi_{\text {bundle }}(\mathrm{atm})$. | $\Pi_{\text {bulk }}(\mathrm{atm})$. |
| :---: | :---: | :---: | :---: |
| 32 | $(-6.057 \pm 0.759)$ | $(-5.044 \pm 0.309)$ | $(1.012 \pm 0.450)$ |
| 36 | $(-2.178 \pm 0.605)$ | $(-1.509 \pm 0.295)$ | $(0.668 \pm 0.310)$ |
| 40 | $(-0.054 \pm 0.521)$ | $(0.451 \pm 0.284)$ | $(0.506 \pm 0.237)$ |
| 44 | $(0.624 \pm 0.474)$ | $(1.089 \pm 0.261)$ | $(0.464 \pm 0.213)$ |
| 48 | $(0.931 \pm 0.434)$ | $(1.423 \pm 0.223)$ | $(0.491 \pm 0.211)$ |
| 52 | $(1.066 \pm 0.407)$ | $(1.610 \pm 0.194)$ | $(0.543 \pm 0.213)$ |
| 54 | $(1.136 \pm 0.388)$ | $(1.714 \pm 0.175)$ | $(0.578 \pm 0.213)$ |
| 58 | $(1.155 \pm 0.362)$ | $(1.808 \pm 0.149)$ | $(0.653 \pm 0.213)$ |
| 62 | $(1.191 \pm 0.351)$ | $(1.954 \pm 0.134)$ | $(0.763 \pm 0.217)$ |
| 66 | $(1.191 \pm 0.333)$ | $(2.039 \pm 0.118)$ | $(0.848 \pm 0.215)$ |
| 70 | $(1.169 \pm 0.320)$ | $(2.126 \pm 0.104)$ | $(0.956 \pm 0.216)$ |
| 74 | $(1.138 \pm 0.307)$ | $(2.193 \pm 0.092)$ | $(1.054 \pm 0.215)$ |
| 78 | $(1.107 \pm 0.301)$ | $(2.267 \pm 0.086)$ | $(1.160 \pm 0.215)$ |
| 82 | $(1.067 \pm 0.290)$ | $(2.318 \pm 0.078)$ | $(1.250 \pm 0.212)$ |
| 86 | $(1.026 \pm 0.283)$ | $(2.380 \pm 0.072)$ | $(1.354 \pm 0.211)$ |
| 90 | $(0.985 \pm 0.275)$ | $(2.426 \pm 0.067)$ | $(1.440 \pm 0.208)$ |
| 94 | $(0.955 \pm 0.266)$ | $(2.453 \pm 0.063)$ | $(1.498 \pm 0.203)$ |
| 98 | $(0.914 \pm 0.257)$ | $(2.489 \pm 0.058)$ | $(1.575 \pm 0.199)$ |
| 102 | $(0.874 \pm 0.252)$ | $(2.513 \pm 0.055)$ | $(1.638 \pm 0.197)$ |

Table C. 21 - Osmotic pressure calculated in a hexagonal lattice of DNA condensed with $\times 406$ -$\operatorname{AuNPs}\left(R_{+/-}=1.00\right)$ with salt concentration of $\mathrm{c}_{\text {salt }}=120 \mathrm{mMol} / \mathrm{l}$. The data are gathered during $\times 10^{6} \mathrm{MC}$ cycles to calculate statistics.

| $\ell(\AA)$ | $\Pi_{\text {rel }}$ (atm.) | $\Pi_{\text {bundle }}(\mathrm{atm})$. | $\Pi_{\text {bulk }}(\mathrm{atm})$. |
| :---: | :---: | :---: | :---: |
| 32 | $(-5.020 \pm 1.027)$ | $(-2.997 \pm 0.339)$ | $(2.022 \pm 0.688)$ |
| 36 | $(-0.939 \pm 0.825)$ | $(0.500 \pm 0.312)$ | $(1.440 \pm 0.513)$ |
| 40 | $(1.218 \pm 0.688)$ | $(2.307 \pm 0.289)$ | $(1.089 \pm 0.399)$ |
| 44 | $(1.996 \pm 0.625)$ | $(3.082 \pm 0.262)$ | $(1.085 \pm 0.363)$ |
| 48 | $(2.253 \pm 0.573)$ | $(3.429 \pm 0.224)$ | $(1.176 \pm 0.349)$ |
| 52 | $(2.381 \pm 0.546)$ | $(3.747 \pm 0.197)$ | $(1.365 \pm 0.349)$ |
| 54 | $(2.385 \pm 0.526)$ | $(3.842 \pm 0.180)$ | $(1.456 \pm 0.346)$ |
| 58 | $(2.350 \pm 0.506)$ | $(4.013 \pm 0.159)$ | $(1.662 \pm 0.347)$ |
| 62 | $(2.298 \pm 0.489)$ | $(4.220 \pm 0.138)$ | $(1.922 \pm 0.351)$ |
| 66 | $(2.210 \pm 0.472)$ | $(4.351 \pm 0.125)$ | $(2.141 \pm 0.347)$ |
| 70 | $(2.091 \pm 0.456)$ | $(4.476 \pm 0.112)$ | $(2.384 \pm 0.344)$ |
| 74 | $(1.994 \pm 0.442)$ | $(4.554 \pm 0.101)$ | $(2.559 \pm 0.341)$ |
| 78 | $(1.921 \pm 0.429)$ | $(4.666 \pm 0.094)$ | $(2.745 \pm 0.335)$ |
| 82 | $(1.801 \pm 0.415)$ | $(4.731 \pm 0.086)$ | $(2.929 \pm 0.329)$ |
| 86 | $(1.764 \pm 0.400)$ | $(4.799 \pm 0.080)$ | $(3.034 \pm 0.320)$ |
| 90 | $(1.668 \pm 0.382)$ | $(4.854 \pm 0.073)$ | $(3.186 \pm 0.309)$ |
| 94 | $(1.549 \pm 0.375)$ | $(4.904 \pm 0.069)$ | $(3.355 \pm 0.306)$ |
| 98 | $(1.499 \pm 0.365)$ | $(4.957 \pm 0.065)$ | $(3.457 \pm 0.300)$ |
| 102 | $(1.419 \pm 0.353)$ | $(4.978 \pm 0.061)$ | $(3.559 \pm 0.292)$ |

## APPENDIX D

## Force-field parameters for PEDOT:PSS and ionic liquids

## Contents

D. 1 Force-field parameters for PEDOT:PSS395
D. 2 Force-field parameters for IL anions ..... 400
D. 3 Force-field parameters for IL cation ..... 405

A parameterization of the energy interactions based on the OPLS-AA force-field (see section 3.3.3.1 and 3.3.3.2 is used in MD simulations presented in chapter 5 Force-field (FF) parameters are given for the simplest model of PEDOT and PSS which are tri-EDOT and PTS molecules as well as for ionic liquids (ILs) used in the MD simulations.

## D. 1 Force-field parameters for PEDOT:PSS



Figure D. 1 - Atomic names of tri-EDOT and PTS.

Table D. 1 - FF parameters: atomic charges taken from DFT calculations [201].

| Name | C1 | C2 | C 3 | C 4 | C 5 | H 6 | H 7 | C 8 |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| FF type | CA | CB | CB | CA | CT | HC | HC | CT |
| q $(\|\mathbf{e}\|)$ | -0.275 | -0.003 | 0.005 | -0.183 | 0.090 | 0.186 | 0.153 | 0.200 |
| Name | H 9 | H 10 | C 11 | C 12 | C 13 | C 14 | C 15 | H 16 |
| FF type | HC | HC | CA | CB | CB | CA | CT | HC |
| q (\|e|) | 0.311 | 0.143 | 0.294 | -0.157 | 0.030 | 0.023 | 0.019 | -0.080 |
| Name | H 17 | C 18 | H 19 | H 20 | C 21 | C 22 | C 23 | C 24 |
| FF type | HC | CT | HC | HC | CA | CB | CB | CA |
| q (\|e|) | 0.111 | -0.072 | 0.021 | -0.071 | -0.126 | 0.014 | 0.012 | -0.110 |
| Name | C 25 | H 26 | H 27 | C 28 | H 29 | H 30 | O 31 | O 32 |
| FF type | CT | HC | HC | CT | HC | HC | OS | OS |
| q (\|e|) | 0.119 | -0.032 | -0.021 | -0.052 | 0.093 | 0.015 | 0.006 | -0.049 |
| Name | O 33 | O 34 | S 35 | O 36 | O 37 | S 38 | S 39 | C 40 |
| FF type | OS | OS | SA | OS | OS | SA | SA | CT |
| q (\|e|) | -0.055 | 0.028 | -0.093 | 0.012 | -0.029 | -0.009 | -0.087 | -0.002 |
| Name | H 41 | H 42 | H 43 | C 44 | H 45 | H 46 | H 47 | S 48 |
| FF type | HC | HC | HC | CT | HC | HC | HC | SY |
| q (\|e|) | 0.028 | 0.035 | 0.008 | -0.015 | 0.040 | 0.005 | 0.015 | 1.293 |
| Name | O 49 | O 50 | O 51 | C 52 | C 53 | C 54 | C 55 | C 56 |
| FF type | OY | OY | OY | CA | CA | CA | CA | CA |
| q (\|e|) | -0.766 | -0.766 | -0.766 | -0.115 | -0.115 | -0.115 | 0.007 | -0.115 |
| Name | C 57 | C 58 | H 59 | H 60 | H 61 | H 62 | H 63 | H 64 |
| FF type | CA | CT | HA | HA | HA | HA | HA | HA |
| q (\|e|) | -0.115 | -0.065 | 0.115 | 0.115 | 0.115 | 0.115 | 0.060 | 0.060 |
| Name | H 65 |  |  |  |  |  |  |  |
| FF type | HA |  |  |  |  |  |  |  |
| q (\|e|) | 0.060 |  |  |  |  |  |  |  |

Table D. 2 - FF parameters: Lennard-Jones vdW.

| FF type | CA | CB | CT | HC | OS | SY | OY | HA |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\sigma(\AA)$ | 3.55 | 3.55 | 3.50 | 2.50 | 2.90 | 3.55 | 2.96 | 2.42 |
| $\epsilon(\mathbf{k J} / \mathbf{m o l})$ | 0.292 | 0.292 | 0.276 | 0.125 | 0.586 | 1.046 | 0.711 | 1.255 |
| FF type | SA |  |  |  |  |  |  |  |
| $\sigma(\AA)$ | 3.55 |  |  |  |  |  |  |  |
| $\epsilon(\mathbf{k J} / \mathbf{m o l})$ | 1.653 |  |  |  |  |  |  |  |

Table D. 3 - FF parameters: bond stretching.

| Bond type | CA-CB | CA-CA | SA-CA | CB-CB | OS-CB | HC-CT |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: |
| $r_{e q}(\AA)$ | 1.370 | 1.441 | 1.745 | 1.425 | 1.345 | 1.090 |
| $k_{\text {bond }}\left(\mathbf{k J} \cdot \mathbf{m o l}^{-1} \cdot \AA^{-2}\right)$ | 4351.4 | 3096.2 | 2217.5 | 3347.2 | 4016.6 | 2217.5 |
| Bond type | CT-CT | OS-CT | CT-CA | CB-HC | CA-HC | CA-HA |
| $r_{e q}(\AA)$ | 1.529 | 1.410 | 1.489 | 1.075 | 1.080 | 1.080 |
| $k_{\text {bond }}\left(\mathbf{k J . m o l}^{-1} \cdot \AA^{-2}\right)$ | 2242.6 | 2677.7 | 2468.5 | 3347.2 | 3071.0 | 3096.2 |
| Bond type | SY-CA | SY-OY | SY-OS |  |  |  |
| $r_{e q}(\AA)$ | 1.779 | 1.440 | 1.682 |  |  |  |
| $k_{\text {bond }}\left(\mathbf{k J . m o l}^{-1} \cdot \AA^{-2}\right)$ | 2845.1 | 5857.6 | 2803.3 |  |  |  |

Table D. 4 - FF parameters: angle bending.

| Angle type | CA-CA-CB | CA-SY-OY | OY-SY-OY | CB-CB-CA | OS-CB-CA |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\theta_{e q}$ (degrees) | 129.85 | 170.20 | 119.00 | 112.39 | 125.19 |
| $k_{\theta}\left(\mathrm{kJ} . \mathrm{mol}^{-1} . \mathrm{rad}^{-2}\right)$ | 460.2 | 619.2 | 870.3 | 627.6 | 627.6 |
| Angle type | OS-CB-CB | HC-CT-HC | HC-CT-CT | OS-CT-HC | OS-CT-CT |
| $\theta_{\text {eq }}$ (degrees) | 126.70 | 1077.80 | 110.70 | 109.50 | 109.50 |
| $k_{\theta}\left(\mathrm{kJ} . \mathrm{mol}^{-1} . \mathrm{rad}^{-2}\right)$ | 627.6 | 276.2 | 313.8 | 292.9 | 418.4 |
| Angle type | CT-CA-CB | OY-SY-OS | CT-OS-CB | SY-OS-CT | HC-CT-CA |
| $\theta_{\text {eq }}$ (degrees) | 130.44 | 106.70 | 112.30 | 113.05 | 110.47 |
| $k_{\theta}\left(\mathrm{kJ} . \mathrm{mol}^{-1} . \mathrm{rad}^{-2}\right)$ | 460.2 | 1129.7 | 627.6 | 836.8 | 376.6 |
| Angle type | HC-CA-CA | CA-CA-CA | HA-CA-CA | CT-CA-CA | SY-CA-CA |
| $\theta_{\text {eq }}$ (degrees) | 130.70 | 120.00 | 120.00 | 120.00 | 119.40 |
| $k_{\theta}\left(\mathrm{kJ} . \mathrm{mol}^{-1} . \mathrm{rad}^{-2}\right)$ | 585.8 | 527.2 | 292.9 | 585.8 | 527.2 |
| Angle type | SA-CA-CB | SA-CA-CA | SA-CA-CT | CA-SA-CA | SY-CA-CA |
| $\theta_{\text {eq }}$ (degrees) | 119.40 | 122.57 | 123.52 | 90.41 | 119.40 |
| $k_{\theta}\left(\mathrm{kJ} . \mathrm{mol}^{-1} . \mathrm{rad}^{-2}\right)$ | 711.3 | 543.9 | 543.9 | 920.5 | 527.2 |

Table D. 5 - FF parameters: Ryckaert-Bellemans dihedral torsion.

| Dihedral type | HC-CT-CT-HC | HC-CT-CT-OS | HC-CT-OS-CB | CT-CT-OS-CB |
| :---: | :---: | :---: | :---: | :---: |
| $\mathrm{C}_{0}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.6276 | 1.0041 | 0.3347 | -2.9497 |
| $\mathrm{C}_{1}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 1.8828 | 2.9372 | 1.0041 | 5.6442 |
| $\mathrm{C}_{2}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 2.9957 |
| $\mathrm{C}_{3}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | -2.5104 | -3.9162 | -5.6735 | -5.6735 |
| $\mathrm{C}_{4}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $\mathrm{C}_{5}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Dihedral type | CT-OS-CB-CB | CT-OS-CB-CA | OS-CT-CT-OS | *-CB-CA-* |
| $\mathrm{C}_{0}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 9.1796 | 9.1796 | -3.61916 | 30.3340 |
| $\mathrm{C}_{1}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | -1.1296 | -1.1296 | -0.3891 | 0.0000 |
| $\mathrm{C}_{2}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | -8.0165 | -8.0165 | 9.7319 | -30.3340 |
| $\mathrm{C}_{3}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | -5.7237 | 0.0000 |
| $\mathrm{C}_{4}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $\mathrm{C}_{5}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Dihedral type | *-CB-CB-* | CB-CA-CA-SA | *-CA-CA-* | *-CA-SA-* |
| $\mathrm{C}_{0}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 30.3340 | 6.5270 | 30.3340 | 30.3340 |
| $\mathrm{C}_{1}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 1.7698 | 0.0000 | 0.0000 |
| $\mathrm{C}_{2}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | -30.330 | -5.1630 | -30.3340 | 30.3340 |
| $\mathrm{C}_{3}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | -3.1463 | 0.0000 | 0.0000 |
| $\mathrm{C}_{4}$ (kJ.mol ${ }^{-1}$ ) | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $\mathrm{C}_{5}$ (kJ.mol ${ }^{-1}$ ) | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Dihedral type | CB-CB-CA-SA | CB-CA-CT-HC | CA-CB-CB-CA | SA-CA-CA-SA |
| $\mathrm{C}_{0}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 6.5270 | 0.0000 | 4.7990 | -0.5732 |
| $\mathrm{C}_{1}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 1.7698 | 0.0000 | 1.7572 | 1.6694 |
| $\mathrm{C}_{2}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | -5.1630 | 0.0000 | -5.2969 | 4.5940 |
| $\mathrm{C}_{3}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | -3.1463 | 0.0000 | -1.2552 | -5.6735 |
| $\mathrm{C}_{4}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $\mathrm{C}_{5}$ (kJ.mol ${ }^{-1}$ ) | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Dihedral type | SA-CA-CT-HC | HC-CT-CA-CA | OY-SY-CA-CA | OY-SY-OS-CT |
| $\mathrm{C}_{0}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 1.8200 | 1.2468 |
| $\mathrm{C}_{1}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | -0.8158 | 3.4685 |
| $\mathrm{C}_{2}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | -0.9790 | 0.0000 |
| $\mathrm{C}_{3}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | -4.9873 |
| $\mathrm{C}_{4}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $\mathrm{C}_{5}$ (kJ.mol ${ }^{-1}$ ) | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Dihedral type | SY-OS-CT-HC | SY-OS-CT-CT | OS-CT-CT-HC |  |
| $\mathrm{C}_{0}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 1.5690 | 0.9790 |  |
| $\mathrm{C}_{1}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 4.7070 | 2.9371 |  |
| $\mathrm{C}_{2}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | -0.9790 |  |
| $\mathrm{C}_{3}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | -6.2760 | -3.9162 |  |
| $\mathrm{C}_{4}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 |  |
| $\mathrm{C}_{5}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 |  |

Table D. 6 - FF parameters: Improper torsion

| Torsion type | HC-N-CB-N | CA-HC-CA-N | CT-CB-N-CA |
| :--- | :---: | :---: | :---: |
| $f(\mathbf{k J} / \mathbf{m o l})$ | 4.602 | 4.602 | 4.602 |
| $\phi_{o}\left({ }^{\circ}\right)$ | 180.0 | 180.0 | 180.0 |

## D. 2 Force-field parameters for IL anions



Figure D. 2 - Atomic name description for tricyanomethanide (TCM), Heptacyanocyclopentenide (HCCP), tetracyanoborate (TCB) and ethylene sulfonate (ES).

Table D. 7 - FF parameters: atomic charges taken from DFT calculations [201].

| Name | C 1 | C 2 | C 3 | C 4 | C 5 | C 6 | C 7 | C 8 |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| FF type | CM | CZ | CZ | CZ | CA | CA | CB | CB |
| q $(\|\mathbf{e}\|)$ | -1.000 | 0.56 | 0.56 | 0.56 | -0.100 | -0.040 | -0.010 | -0.06 |
| Name | C 9 | C 10 | C 11 | C 12 | C 13 | C 14 | C 15 | C 16 |
| FF type | CB | CZ | CZ | CZ | CZ | CZ | CZ | CZ |
| q $(\|\mathbf{e}\|)$ | -0.06 | 0.400 | 0.400 | 0.400 | 0.030 | 0.030 | 0.030 | 0.030 |
| Name | C 17 | C 18 | C 19 | C 20 | C 21 | C 22 | N 1 | N 2 |
| FF type | CN | CN | CN | CN | CT | CT | NZ | NZ |
| q $(\|\mathbf{e}\|)$ | 0.430 | 0.430 | 0.430 | 0.430 | 0.051 | -0.180 | -0.560 | -0.560 |
| Name | N 3 | N 4 | N 5 | N 6 | N 7 | N 8 | N 9 | N 10 |
| FF type | NZ | NZ | NZ | NZ | NZ | NZ | NZ | NZ |
| q $(\|\mathbf{e}\|)$ | -0.560 | -0.500 | -0.400 | -0.400 | -0.400 | -0.400 | -0.500 | -0.440 |
| Name | N 11 | N 12 | N 13 | N 14 | B 1 | S 1 | O 1 | O 2 |
| FF type | NZ | NZ | NZ | NZ | BZ | SY | OY | OY |
| q $(\|\mathbf{e}\|)$ | -0.560 | -0.560 | -0.560 | -0.560 | -0.480 | 1.628 | -0.766 | -0.766 |
| Name | O 3 | O 4 | H 1 | H 2 | H 3 | H 4 | H 5 |  |
| FF type | OY | OS | HC | HC | HC | HC | HC |  |
| q $(\|\mathbf{e}\|)$ | -0.766 | -0.500 | 0.060 | 0.060 | 0.060 | 0.060 | 0.060 |  |

Table D. 8 - FF parameters: Lennard-Jones vdW.

| FF type | CM | CZ | CA | CB | CN | CT | NZ | BZ |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\sigma(\AA)$ | 3.30 | 3.30 | 3.55 | 3.55 | 3.30 | 3.50 | 3.28 | 3.60 |
| $\epsilon(\mathbf{k J} / \mathbf{m o l})$ | 0.271 | 0.271 | 0.292 | 0.292 | 0.272 | 0.276 | 0.711 | 0.210 |
| FF type | SY | OY | OS | HC |  |  |  |  |
| $\sigma(\AA)$ | 3.55 | 2.96 | 2.90 | 2.50 |  |  |  |  |
| $\epsilon(\mathbf{k J} / \mathbf{m o l})$ | 1,046 | 0.711 | 0.586 | 0.125 |  |  |  |  |

Table D. 9 - FF parameters: bond stretching.

| Bond type | CM-CZ | NZ-CZ | CZ-CA | CZ-CB | CA-CA | CB-CB |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $r_{e q}(\AA)$ | 1.430 | 1.160 | 1.450 | 1.450 | 1.370 | 1.430 |
| $k_{\text {bond }}\left(\mathrm{kJ} . \mathrm{mol}^{-1} . \AA^{-2}\right)$ | 3347.2 | 5439,2 | 3347.2 | 3347.2 | 4351.4 | 3347.2 |
| Bond type | CA-CB | OS-CT | HC-CT | CT-CT | SY-OY | SY-OS |
| $r_{e q}(\AA)$ | 1.440 | 1.410 | 1.090 | 1.529 | 1.440 | 1.682 |
| $k_{\text {bond }}\left(\mathrm{kJ} . \mathrm{mol}^{-1} . \AA^{-2}\right)$ | 3924.6 | 2677.7 | 2217.5 | 2242.6 | 5857.6 | 2803.3 |
| Bond type | SY-OY | SY-OS | BZ-CN |  |  |  |
| $r_{e q}(\AA)$ | 1.440 | 1.682 | 1.588 |  |  |  |
| $k_{\text {bond }}\left(\mathrm{kJ} . \mathrm{mol}^{-1} . \AA^{-2}\right)$ | 5857.6 | 2803.3 | 1589.9 |  |  |  |

Table D. 10 - FF parameters: angle bending.

| Angle type | CZ-CM-CZ | NZ-CZ-CM | NZ-CZ-C* | CB-CB-CA | CZ-CB-C* |
| :--- | :---: | :---: | :---: | :---: | :---: |
| $\theta_{e q}$ (degrees) | 125.52 | 179.20 | 180.00 | 120.00 | 120.00 |
| $k_{\theta}\left(\mathbf{k J . m o l}{ }^{-1} \cdot \mathbf{r a d}^{-2}\right)$ | 251.0 | 146.4 | 1255.2 | 527.2 | 585.8 |
| Angle type | CZ-CA-C | CA-CA-CB | CZ-CB-CZ | OY-SY-OY | SY-OS-CT |
| $\theta_{e q}$ (degrees) | 120.00 | 120.00 | 109.50 | 119.00 | 113.05 |
| $k_{\theta}\left(\mathbf{k J . m o l}^{-1} \cdot \mathbf{r a d}^{-2}\right)$ | 585.8 | 527.2 | 585.8 | 870.3 | 836.8 |
| Angle type | HC-CT-HC | HC-CT-CT | OS-CT-HC | OS-CT-CT | OY-SY-OS |
| $\theta_{e q}($ degrees $)$ | 1077.80 | 110.70 | 109.50 | 109.50 | 106.70 |
| $k_{\theta}\left(\mathbf{k J . m o l}^{-1} \cdot \mathbf{r a d}^{-2}\right)$ | 276.2 | 313.8 | 292.9 | 418.4 | 1129.7 |
| Angle type | SY-OS-CT | CN-BZ-CN | NZ-CN-BZ |  |  |
| $\theta_{e q}($ degrees $)$ | 113.05 | 111.90 | 177.61 |  |  |
| $k_{\theta}\left(\mathbf{k J . m o l}^{-1} \cdot \mathbf{r a d}^{-2}\right)$ | 836.8 | 502.1 | 502.1 |  |  |

Table D. 11 - FF parameters: Ryckaert-Bellemans dihedral torsion.

| Dihedral type | *-CZ-CB-* | *-CB-CB-* | CB-CB-CA-CA | *-CB-CA-* |
| :---: | :---: | :---: | :---: | :---: |
| $\mathrm{C}_{0}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 30.3340 | 4.7990 | 30.3340 |
| $\mathrm{C}_{1}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 1.7530 | 0.0000 |
| $\mathrm{C}_{2}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | -30.3340 | -5.2970 | -30.3340 |
| $\mathrm{C}_{3}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | -1.255 | 0.0000 |
| $\mathrm{C}_{4}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $\mathrm{C}_{5}$ (kJ.mol ${ }^{-1}$ ) | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Dihedral type | *-CA-CZ-* | CB-CB-CB-CA | CB-CA-CA-CB | *-CA-CA-* |
| $\mathrm{C}_{0}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 4.7990 | 4.7990 | 30.3340 |
| $\mathrm{C}_{1}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 1.7530 | 1.7530 | 0.0000 |
| $\mathrm{C}_{2}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | -5.2970 | -5.2970 | -30.3340 |
| $\mathrm{C}_{3}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | -1.255 | -1.255 | 0.0000 |
| $\mathrm{C}_{4}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $\mathrm{C}_{5}$ (kJ.mol ${ }^{-1}$ ) | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Dihedral type | HC-CT-CT-HC | HC-CT-CT-OS | HC-CT-OS-CB | CT-CT-OS-CB |
| $\mathrm{C}_{0}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.6276 | 1.0041 | 0.3347 | -2.9497 |
| $\mathrm{C}_{1}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 1.8828 | 2.9372 | 1.0041 | 5.6442 |
| $\mathrm{C}_{2}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 2.9957 |
| $\mathrm{C}_{3}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | -2.5104 | -3.9162 | -3.9330 | -5.6735 |
| $\mathrm{C}_{4}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $\mathrm{C}_{5}$ (kJ.mol ${ }^{-1}$ ) | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Dihedral type | CT-OS-CB-CB | CT-OS-CB-CA | OS-CT-CT-OS | CA-CB-CB-CA |
| $\mathrm{C}_{0}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 9.1796 | 9.1796 | -3.6191 | 4.7990 |
| $\mathrm{C}_{1}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | -1.1297 | -1.1297 | -0.3891 | 1.7572 |
| $\mathrm{C}_{2}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | -8.0165 | -8.0165 | 9.7319 | -5.2969 |
| $\mathrm{C}_{3}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 5.7237 | -1.2552 |
| $\mathrm{C}_{4}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $\mathrm{C}_{5}$ (kJ.mol ${ }^{-1}$ ) | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Dihedral type | NZ-CN-BZ-CN | OY-SY-OS-CT | SY-OS-CT-HC | SY-OS-CT-CT |
| $\mathrm{C}_{0}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 1.2468 | 0.0000 | 1.5690 |
| $\mathrm{C}_{1}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 3.4685 | 0.0000 | 4.7070 |
| $\mathrm{C}_{2}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $\mathrm{C}_{3}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | -4.9873 | 0.0000 | -6.2760 |
| $\mathrm{C}_{4}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $\mathrm{C}_{5}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Dihedral type | OS-CT-CT-HC |  |  |  |
| $\mathrm{C}_{0}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.9790 |  |  |  |
| $\mathrm{C}_{1}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 2.9371 |  |  |  |
| $\mathrm{C}_{2}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 |  |  |  |
| $\mathrm{C}_{3}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | -3.9162 |  |  |  |
| $\mathrm{C}_{4}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 |  |  |  |
| $\mathrm{C}_{5}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 |  |  |  |

Table D. 12 - FF parameters: improper torsion.

| Torsion type | CB-CB-CA-CZ | CB-CA-CA-CZ | CA-CA-CB-CZ |
| :--- | :---: | :---: | :---: |
| $\mathbf{k}_{\theta}(\mathrm{kJ} / \mathbf{m o l})$ | 4.602 | 4.602 | 4.602 |
| $\phi_{s}($ degrees $)$ | 180.0 | 180.0 | 180.0 |

## D. 3 Force-field parameters for IL cation



Figure D. 3 - Atomic name description for 1-Ethyl-3-methylimidazolium (EMIM).

Table D. 13 - FF parameters: atomic charges.

| Name | C52 | C53 | N54 | N55 | C56 | C57 | C58 | C59 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| FF type | CA | CA | N | N | CT | CB | CT | CT |
| q ( $\|\mathrm{e}\|$ ) | 0.056 | -0.174 | 0.071 | 0.025 | 0.002 | 0.100 | -0.058 | -0.180 |
| Name | H60 | H61 | H62 | H63 | H64 | H65 | H66 | H67 |
| FF type | HC | HC | HC | HC | HC | HC | HC | HC |
| q ( $\|\mathrm{e}\|$ ) | 0.258 | 0.182 | 0.060 | 0.060 | 0.238 | 0.060 | 0.060 | 0.060 |
| Name | H68 | H69 | H70 |  |  |  |  |  |
| FF type | HC | HC | HC |  |  |  |  |  |
| q (\|e|) | 0.060 | 0.060 | 0.060 |  |  |  |  |  |

Table D. 14 - FF parameters: Lennard-Jones vdW.

| FF type | N | CT | HC | CA | CB |
| :--- | :---: | :---: | :---: | :---: | :---: |
| $\sigma(\AA)$ | 3.25 | 3.50 | 2.50 | 3.55 | 3.55 |
| $\epsilon(\mathrm{~kJ} / \mathbf{m o l})$ | 0.711 | 0.276 | 0.125 | 0.293 | 0.293 |

Table D. 15 - FF parameters: bond stretching.

| Bond type | N-CT | CT-CT | CT-HC | N-HC | CA-CA | CA-HC |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: |
| $r_{e q}(\AA)$ | 1.471 | 1.529 | 1.090 | 1.010 | 1.441 | 1.080 |
| $k_{\text {bond }}\left(\mathbf{k J} . \mathrm{mol}^{-1} . \AA^{-2}\right)$ | 3071.1 | 2242,6 | 2845.1 | 3631.7 | 3096.2 | 3071.1 |
| Bond type | CA-N | CB-N | CB-HC |  |  |  |
| $r_{e q}(\AA)$ | 1.369 | 1.369 | 1.075 |  |  |  |
| $k_{\text {bond }}\left(\mathbf{k J} . \mathrm{mol}^{-1} . \AA^{-2}\right)$ | 3807.4 | 3807.4 | 2845.1 |  |  |  |

Table D. 16 - FF parameters: angle bending.

| Angle type | CT-CT-CT | CT-CT-HC | HC-CT-HC | N-CT-CT | CT-N-CT |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\theta_{\text {eq }}$ (degrees) | 112.70 | 110.70 | 107.80 | 109.50 | 107.20 |
| $k_{\theta}\left(\mathrm{kJ} . \mathrm{mol}^{-1} \cdot \mathrm{rad}^{-2}\right)$ | 489.5 | 313.8 | 276.1 | 468.6 | 435.1 |
| Angle type | HC-CT-N | CT-N-HC | HC-CT-HC | HC-CT-CT | CT-CT-N |
| $\theta_{e q}$ (degrees) | 109.50 | 120.00 | 107.80 | 110.70 | 111.20 |
| $k_{\theta}\left(\mathbf{k J} . \mathrm{mol}^{-1} \cdot \mathrm{rad}^{-2}\right)$ | 292.9 | 292.9 | 276.2 | 313.8 | 1255.2 |
| Angle type | CT-N-CB | CT-N-CA | N-CA-CA | HC-CA-CA | $\mathrm{N}-\mathrm{CA}-\mathrm{HC}$ |
| $\theta_{e q}$ (degrees) | 121.70 | 126.76 | 106.20 | 130.70 | 122.08 |
| $k_{\theta}\left(\mathbf{k J} . \mathrm{mol}^{-1} \cdot \mathrm{rad}^{-2}\right)$ | 1087.9 | 1004.2 | 1506.2 | 585.8 | 669.4 |
| Angle type | N-CB-HC | CB-N-CA | N-CT-HC | N-CB-N | N-CA-HC |
| $\theta_{e q}$ (degrees) | 125.70 | 110.36 | 107.70 | 107.19 | 122.08 |
| $k_{\theta}\left(\mathbf{k J} . \mathrm{mol}^{-1} \cdot \mathrm{rad}^{-2}\right)$ | 669.4 | 627.6 | 836.8 | 1673.6 | 669.4 |

Table D. 17 - FF parameters: Ryckaert-Bellemans dihedral torsion.

| Dihedral type | HC-CT-CT-HC | CT-CT-CT-HC | CT-CT-CT-CT | HC-CT-CT-N |
| :---: | :---: | :---: | :---: | :---: |
| $\mathrm{C}_{0}$ (kJ.mol ${ }^{-1}$ ) | 0.6282 | 0.6282 | 2.9316 | -4.1000 |
| $\mathrm{C}_{1}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 1.8846 | 1.8846 | -1.4657 | 5.0926 |
| $\mathrm{C}_{2}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.2094 | 2.9692 |
| $\mathrm{C}_{3}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | -2.5128 | -2.5128 | -1.6751 | -3.9618 |
| $\mathrm{C}_{4}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $\mathrm{C}_{5}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Dihedral type | CT-CT-CT-N | CT-N-CT-CT | HC-CT-N-CT | *-CA-CA-* |
| $\mathrm{C}_{0}$ (kJ.mol ${ }^{-1}$ ) | 3.3378 | 1.7903 | 1.1726 | 30.3340 |
| $\mathrm{C}_{1}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | -1.5537 | 3.4948 | 3.5179 | 0.0000 |
| $\mathrm{C}_{2}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 2.8227 | 0.5360 | 0.0000 | -30.3340 |
| $\mathrm{C}_{3}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | -4.6067 | -5.8213 | -1.6751 | 0.0000 |
| $\mathrm{C}_{4}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $\mathrm{C}_{5}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Dihedral type | *-CT-N_* | *-CA-N-* | *-CB-N-* | N-CA-CA-N |
| $\mathrm{C}_{0}$ (kJ.mol ${ }^{-1}$ ) | 0.0000 | 30.3340 | 30.3340 | 44.9780 |
| $\mathrm{C}_{1}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $\mathrm{C}_{2}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | -30.3340 | -30.3340 | -44.9780 |
| $\mathrm{C}_{3}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $\mathrm{C}_{4}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $\mathrm{C}_{5}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Dihedral type | CB-N-CT-CT | CA-N-CT-CT | *-CB-N-* | N-CA-CA-N |
| $\mathrm{C}_{0}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 6.3806 | 6.3806 | 30.3340 | 44.9780 |
| $\mathrm{C}_{1}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.2970 | 0.2970 | 0.0000 | 0.0000 |
| $\mathrm{C}_{2}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | -0.1338 | -0.1338 | -30.3340 | -44.9780 |
| $\mathrm{C}_{3}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | -6.5437 | -6.5437 | 0.0000 | 0.0000 |
| $\mathrm{C}_{4}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| $\mathrm{C}_{5}\left(\mathrm{~kJ} . \mathrm{mol}^{-1}\right)$ | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

Table D. 18 - FF parameters: improper torsion.

| Torsion type | HC-N-CB-N | CA-HC-CA-N | CT-CB-N-CA |
| :--- | :---: | :---: | :---: |
| $\mathbf{k}_{\theta}\left(\mathbf{k J . m o l}{ }^{-1}\right)$ | 4.602 | 4.602 | 4.602 |
| $\phi_{s}($ degrees $)$ | 180.0 | 180.0 | 180.0 |

## appendix E

## Morphology of mixed solution of PEDOT:PSS - large system case

## Contents



In order to support the simulations performed at relatively small scale (box sidelength of 60 $\AA$ ), we probe morphology of mixed PEDOT:PSS solutions with IL EMIM:X at a larger scale in a simulation box of $120 \AA$ side length.

## E. 1 Definiton of the complementary domain analysis

For larger systems, cluster analysis is performed with a friends-of-friends algorithm finding all the molecules within a distance $r_{c}$ of $4.0 \AA$ like for the small systems. This analysis is useful to determine the number of clusters in the box, but is not sufficient at large scale to characterize precisely the structure of these clusters. Indeed, in a large cluster there could be crystallite domains of PEDOT arranged in $\pi-\pi$ stacking and our complementary analysis aim to provide the probability to find the number of PEDOT in these domains. We explain the algorithm as follows. Let us consider the carbon atoms of PEDOT backbone and as an example let us consider that we want
to test if 2 PEDOT units form a domain. For each carbon atom backbone of a PEDOT, we search the closest carbon atom backbone located on the other PEDOT unit. If at least half of the carbon backbone atoms of a PEDOT have a closest carbon backbone neighbor located on the other PEDOT within a cutoff of $6.0 \AA$, we admit that the two PEDOT units are in $\pi$ - $\pi$ stacking and the two PEDOT units form a "domain".

In order to distinguish the two cluster analysis in the subsequent sections, the friends-of-friends algorithm will be mentioned by "FoF" while the new $\pi-\pi$ stacking algorithm will be mentioned by "PPS".

## E. 2 Morphology of tri-EDOT:PTS solution



Figure E. 1 - Final snapshots of large tri-EDOT:PTS aqueous systems treated with $\times 96$ pairs of EMIM:X. For comparison, untreated aqueous PEDOT:PSS system is also shown. Color code: tri-EDOT (blue), PTS (green), X (red) and EMIM (yellow).

Figure E. 1 displays last snasphots of equilibrated large tri-EDOT:PTS aqueous systems in


Figure E. 2 - Zoom-in of the final snapshots for large tri-EDOT:PTS aqueous systems treated with $\times 96$ pairs EMIM:X. For comparison, untreated aqueous PEDOT:PSS system is also shown. Color code: tri-EDOT (blue), PTS (green), X (red). We do not display EMIM cation and PTS anions in systems treated with EMIM:X for clarity.
presence of $\times 96$ pairs of ILs. EMIM:CL and EMIM:ES have limited effect on the PEDOT:PTS structure (blue frame) that is similar to structure obtained for untreated PEDOT:PTS system (black frame). On the contrary, in presence of hydrophobic anions $\mathrm{X}=\mathrm{TCM}, \mathrm{TCB}$ and HCCP, the tri-EDOT:PTS matrix collapses into one large domain of tri-EDOT units (orange frame). In Figure E.2, we present zoom-in snapshots of the large tri-EDOT:PTS in order to show the local structure of the tri-EDOT aggregates. It appears clearly that TCM, TCB and HCCP anions are stitching tri-EDOT domains together, playing the role of anionic glue.

The FoF Cluster analysis (Figure E.3) performed on systems mixed with $\times 96$ pairs of EMIM:Cl or EMIM:ES show that at most a single large cluster of $\sim 40$ tri-EDOT units is formed, an observation similar to the one made for untreated tri-EDOT:PTS system. This feature reinforces the fact that these ILs have limited effect on the PEDOT:PSS complexes since most EMIM cations or CL and ES anions remain in solution (see probability to find $\sim 0-8 \mathrm{Cl}$ or $\sim 0-16 \mathrm{ES}$ anions in tri-

EDOT aggregates). In the other way, one large cluster gathering all the tri-EDOT units emerges and contains most of the hydrophobic IL anions ( $\sim 90$ TCM or $\sim 75-80$ TCB or $\sim 96$ HCCP).


Figure E. 3 - Tri-EDOT cluster composition using the FoF alorithm for large tri-EDOT:PTS aqueous systems mixed with $\times 96$ pairs of IL. Cluster analysis performed on large tri-EDOT:PTS aqueous system in absence of IL is displayed in red (tri-EDOT) and cyan (PTS).

The above analysis do not reveal a precise organization of the PEDOT units in term of $\pi-\pi$ stacking. The PPS analysis (Figure E.4) show that in absence of IL at most 10 tri-EDOT units
are in $\pi-\pi$ stacking (Figure E.4 red bars). This would not appear to be the case at first sight by looking at picture E.1 excepted that most of PTS anions are interacting and disturbing long $\pi$-stacked PEDOT chains. After insertion of EMIM:CL and EMIM:ES, longer chain in $\pi$ - $\pi$ stacking emerge (at most $\sim 35,38$ PEDOT respectively) given that PTS anions are more solvated in water du to its interaction with the ionic liquid. Finally after insertion of EMIM:TCM or EMIM:TCB, one single domain emerges and the PPS analysis show that most of the PEDOT units form a long chain of 80-88 PEDOT units in $\pi-\pi$ stacking. The conclusion is different after insertion of EMIM:HCCP where the large cluster of EMIM:HCCP is characterized by plenty of small domain of PEDOT in $\pi-\pi$ stacking (domains of at most $\sim 16$ PEDOT units). The bulkier HCCP anion seem to disturb the long $\pi$ stacked PEDOT chains.

The $\operatorname{RDF}(\mathrm{C}-\mathrm{S})$ curves displayed in Figure E. 5 confirm that the nano-segregation is stronger in presence of hydrophobic anions TCM, TCB or HCCP anions compared to hydrophilic Cl and ES anions as illustrated by the overall decrease exhibited by $\operatorname{RDF}(\mathrm{C}-\mathrm{S})$ in presence of these hydrophilic anions.

We calculated also the radial distribution $\operatorname{RDF}(\mathrm{C}-\mathrm{C})$ between closest PEDOT backbone carbons and corresponding coordination number $\mathrm{CN}(\mathrm{C}-\mathrm{C})$ for large tri-EDOT:PTS systems (Figure E.6.). The RDF(C-C) and CN(C-C) for larger systems in presence of EMIM:Cl and EMIM:ES have the same pattern as for the untreated larger tri-EDOT:PTS system. We see that the RDF(C-C) present a high bump over $\mathrm{d} / \mathrm{d}_{\pi}=15$ in case of untreated and treated systems with EMIM:Cl and EMIM:ES, because for a given PEDOT, most of the tri-EDOTs are located far away in the simulation box which is consistent with the snapshots of Figure E. 1 However, in presence of TCM TCB and HCCP the bump is displaced over $\mathrm{d} / \mathrm{d}_{\pi} \sim 3-6$ because most of the PEDOT are gathered into a single cluster which explains the higher peaks.

Simulations performed for large tri-EDOT:PTS systems seem to confirm the ion exchange mechanism between PEDOT:PSS and EMIM:X.


Figure E. 4 - Number of units in PEDOT domains in $\pi-\pi$ stacking using the PPS algorithm for large tri-EDOT:PTS aqueous systems without (red bars) and mixed with $\times 96$ pairs of IL (blue bars).


Figure E. 5 - Radial distribution function between carbon atoms of PEDOT backbone and sulfonate group of PTS (RDF(C-S)) for large tri-EDOT:PTS systems mixed with $\times 96$ or $\times 192$ pairs of IL EMIM:X.


Figure E. 6 - A) Radial distribution $\operatorname{RDF}(\mathrm{C}-\mathrm{C})$ between closest PEDOT backbone carbon for large tri-EDOT:PTS systems mixed with $\times 96$ pairs of IL EMIM:X. B) Corresponding coordination number CN(C-C).

## E. 3 Morphology of tri-EDOT:16SS solution



Figure E. 7 - Final snapshots for large tri-EDOT:16SS aqueous systems treated with $\times 96$ pairs EMIM:X. For comparison, untreated aqueous PEDOT:PSS system is also shown. Color code: tri-EDOT (blue), 16SS (green), X (red) and EMIM (yellow).

Simulations on large tri-EDOT:16SS aqueous systems are performed in a similar way. A change of morphology of the PEDOT:PSS mixture is less visible because 16SS chain entanglements prevent a complete nanosegregation to take place, at least within the timescale of our simulations.

In Figure E.8, we present zoom-in snapshots of the large tri-EDOT:16SS in order to show the local structure of the tri-EDOT aggregates. Also, for the tri-EDOT:16SS case, TCM, TCB and HCCP anions are stitching tri-EDOT domains together.

We also performed cluster analysis (Figure E.9) to determine the composition of PEDOT clusters. Untreated large tri-EDOT:16SS aqueous system and systems treated with EMIM: Cl and EMIM:ES present similar tri-EDOT clusters composed of $\sim 1-8$ units. This is consistent with the observation that only a limited number of IL anions bind to the tri-EDOT units as well. Despite the presence of 16 SS chains, IL anions $\mathrm{TCM}, \mathrm{TCB}$ and HCCP still have an effect on the resulting


Figure E. 8 - Zoom-in of the final snapshots for large tri-EDOT:16SS aqueous systems treated with $\times 96$ pairs EMIM:X. For comparison, untreated aqueous PEDOT:PSS system is also shown. Color code: tri-EDOT (blue), 16SS (green), X (red). We do not display EMIM cation and 16SS chains in systems treated with EMIM:X for clarity.

PEDOT:PSS complexes as fewer and larger tri-EDOT clusters surrounded by IL anions emerge.
The above analysis do not reveal the organization of the PEDOT units in term of $\pi-\pi$ stacking. The PPS analysis (Figure E.10) show that in absence of IL at most 10 tri-EDOT units are in $\pi-\pi$ stacking (Figure E.10 red bars) which is similar to the number of PEDOT units ( $\sim 10-16$ ) in $\pi-\pi$ stacking after insertion of EMIM:Cl or EMIM:ES. After insertion of EMIM:TCM, EMIM:TCB or EMIM:HCCP, there is emergence of large domains of PEDOT in $\pi-\pi$ stacking ( $\sim 40-56$ PEDOT units) which shows indeed that these IL gather the PEDOT units into large extended domain of PEDOT units.

RDF(C-S) calculated between the backbone carbon atoms of tri-EDOT units and sulfonate groups of 16 SS chains (Figure E.11 display also damped distributions for $\mathrm{X}=\mathrm{TCM}, \mathrm{TCB}$ and HCCP compared to the distribution obtained for untreated PEDOT:PSS systems. We conclude that these anions induce a segregation between PEDOT and PSS. On the contrary, RDF(C-S) for


Figure E. 9 - Tri-EDOT cluster composition for tri-EDOT:16SS large aqueous systems mixed with $\times 96$ pairs of IL. Cluster analysis performed on large aqueous system of tri-EDOT:PTS in absence of IL is displayed in red (tri-EDOT).
systems containing $\mathrm{X}=\mathrm{Cl}$ and ES do not display major change compared to $\mathrm{RDF}(\mathrm{C}-\mathrm{S})$ calculated for untreated PEDOT:PSS system, pointing towards a poorly segregating efficiency of these anions.

We calculated the radial distribution $\operatorname{RDF}(\mathrm{C}-\mathrm{C})$ between closest PEDOT backbone carbons and corresponding coordination number $\mathrm{CN}(\mathrm{C}-\mathrm{C})$ for large tri-EDOT:16SS systems (Figure E.12). The


Figure E. 10 - Number of units in PEDOT domains in $\pi-\pi$ stacking using the PPS algorithm for large tri-EDOT:16SS aqueous systems without (red bars) and mixed with $\times 96$ pairs of IL (blue bars).


Figure E. 11 - Radial distribution functions between carbon atoms of PEDOT backbone and sulfonate group of PTS ( $\operatorname{RDF}(\mathrm{C}-\mathrm{S})$ ) for tri-EDOT:16SS large systems mixed with $\times 96$ or $\times 192$ pairs of IL EMIM:X.

RDF(C-C) and CN(C-C) for larger systems in presence of EMIM:Cl and EMIM:ES have the same pattern as for the untreated larger tri-EDOT:PTS system. Conversely, the $\mathrm{RDF}(\mathrm{C}-\mathrm{C})$ in presence of TCM, TCB and HCCP present higher peaks, corresponding to shorter tri-EDOT domains stitched together by X anions. Therefore, the $\mathrm{CN}(\mathrm{C}-\mathrm{C})$ have higher values at short distance in presence of these ILs compared to system with EMIM:Cl, EMIM:ES or untreated system whose CN(C-C) is shifted.


Figure E. 12 - A) Radial distribution RDF(C-C) between closest PEDOT backbone carbon for large tri-EDOT:16SS systems mixed with $\times 96$ pairs of IL EMIM:X. B) Corresponding coordination number $\mathrm{CN}(\mathrm{C}-\mathrm{C})$.

## E. 4 Morphology of 6EDOT:16SS solution



Figure E. 13 - Final snapshots of 6EDOT:16SS large aqueous systems treated with $\times 96$ pairs EMIM:X. For comparison, untreated aqueous PEDOT:PSS system is also shown. Color code: 6EDOT (blue), 16SS (green), X (red) and EMIM (yellow).

Simulations are performed for 6EDOT:16SS large systems. Final snapshots are displayed in Figure E.13 as well as zoom-in snapshots of 6EDOT:16SS complexes in Figure E. 14 As already noticed for 6EDOT:16SS small aqueous systems, a precise identification of PEDOT clusters become more difficult due to existing interconnexion between longer PEDOT chains. However, analysis of cation and anion populations in PEDOT cluster are in agreement with previous set of smaller-size simulations.

The PPS analysis (Figure E.16) show that in absence or presence of IL, $\sim 106$ EDOT units are in $\pi-\pi$ stacking (Figure E.16, red bars), showing that entanglement between 6EDOT and 16SS chains kinetically trap the system and IL has a limited effect on the PEDOT:PSS morpology.

Entanglements between longer PEDOT chains seem to slow down, at least within our simulation timescale, nano-segregation between PEDOT and PSS chains. The decrease in RDF(C-S)


Figure E. 14 - Zoom-in of the final snapshots for large 6EDOT:16SS aqueous systems treated with $\times 96$ pairs EMIM:X. For comparison, untreated aqueous PEDOT:PSS system is also shown. Color code: tri-EDOT (blue), 16SS (green), X (red). We do not display EMIM cation in systems treated with EMIM: X for clarity.
distribution in Figure E.17 is moderate in presence of hydrophobic anions $\mathrm{X}=\mathrm{TCM}, \mathrm{TCB}$ and HCCP but still noticeable while $\mathrm{RDF}(\mathrm{C}-\mathrm{S})$ for $\mathrm{X}=\mathrm{Cl}$ and ES are very similar to distribution obtained for untreated PEDOT:PSS system. These observations suggest that there exists a partial (at our simulation timescale) nano-segregation between 6EDOT and 16PSS in presence of TCM, TCB and HCCP anions while PEDOT:PSS complexes remains unchanged in presence of Cl or ES anions.

We calculated the radial distribution $\operatorname{RDF}(\mathrm{C}-\mathrm{C})$ between closest PEDOT backbone carbons and corresponding coordination number $\mathrm{CN}(\mathrm{C}-\mathrm{C})$ for large 6EDOT:16SS systems (Figure E.18). The $\operatorname{RDF}(\mathrm{C}-\mathrm{C})$ for untreated system display only two peaks, corresponding to a situation where at most 3-4 PEDOT units are in pi-pi stacking simultaneously locally. In presence of EMIM:Cl and EMIM:ES, there are four peaks lower than those in presence of EMIM:TCM, EMIM:TCB or EMIM:HCCP. Consequently, the $\mathrm{CN}(\mathrm{C}-\mathrm{C})$ are shifted toward greater values for the untreated


Figure E. $15-6$ EDOT cluster composition for 6 EDOT: 16 SS large aqueous systems mixed with $\times 96$ pairs of IL. Cluster analysis of big aqueous system of tri-EDOT:PTS in absence of IL is displayed in red (6EDOT).
system and system treated with EMIM: Cl and EMIM:ES. These curves suggest that a minor change of the PEDOT:PSS morphology is induced by addition of IL: the PEDOT chains are more in $\pi-\pi$ stacking when IL is present and this effect is more important with EMIM:TCM, EMIM:TCB and EMIM:HCCP.


Figure E. 16 - Number of units in PEDOT domains in $\pi-\pi$ stacking using the PPS algorithm for large 6EDOT:16SS aqueous systems without (red bars) and mixed with $\times 96$ pairs of IL (blue bars).


Figure E. 17 - Radial distribution function between carbon atoms of PEDOT backbone and sulfonate group of PTS (RDF $(\mathrm{C}-\mathrm{S})$ ) for 6EDOT:16SS large aqueous systems mixed with $\times 96$ or $\times 192$ pairs of IL EMIM:X.


Figure E. 18 - A) Radial distribution RDF(C-C) between closest PEDOT backbone carbon for large 6EDOT:16SS systems mixed with $\times 96$ pairs of IL EMIM:X. B) Corresponding coordination number CN(C-C).

## E. 5 Morphology of 6EDOT:PTS



Figure E. 19 - Final snapshots of 6EDOT:PTS large aqueous systems treated with $\times 96$ pairs EMIM:X. For comparison, untreated aqueous PEDOT:PSS system is also shown. Color code: 6EDOT (blue), PTS (green), X (red) and EMIM (yellow).

Finally, simulations are performed for 6EDOT:PTS large systems. Final snapshots are displayed in Figure E.19 as well as zoom-in snapshots of 6EDOT:PTS complexes in Figure E.20 It is clear from the snapshots that there is a change of morphology of the 6EDOT:PTS systems. In presence of hydrophobic TCM, TCB and HCCP, the 6EDOT units gather into a single large cluster stitched by the anions (80-90 anions according to cluster analysis in Figure E.21). On the contrary, several clusters remains after treatment with EMIM:Cl or EMIM:ES like for the untreated systems.

The PPS analysis (Figure E.22) show that in absence or presence of EMIM:Cl or EMIM:ES, ~ 10 6EDOT units are in $\pi-\pi$ stacking (Figure E.22), while up to 16 and 40 units in $\pi-\pi$ stacking emerge respectively after insertion of EMIM:TCM and EMIM:TCB. We notice that the bulkier anion HCCP disturb the stacking of the PEDOT chain ( $\sim 106$ EDOT units are in $\pi-\pi$ ), although only one cluster of PEDOT units emerge (Figure E.19).


Figure E. 20 - Zoom-in of the final snapshots for large 6EDOT:PTS aqueous systems treated with $\times 96$ pairs EMIM:X. For comparison, untreated aqueous PEDOT:PTS system is also shown. Color code: tri-EDOT (blue), PTS (green), X (red). We do not display EMIM cation in systems treated with EMIM: X for clarity.

The decrease in $\operatorname{RDF}(\mathrm{C}-\mathrm{S})$ distribution in Figure E.23 right picture, is important in presence of hydrophobic anions $\mathrm{X}=\mathrm{TCM}, \mathrm{TCB}$ and HCCP but moderate $\mathrm{X}=\mathrm{Cl}$ and ES are very similar to distribution obtained for untreated PEDOT:PSS system. These observations suggest that there exists a nano-segregation between 6EDOT and PTS in presence of TCM, TCB and HCCP anions while PEDOT:PSS complexes remains unchanged in presence of Cl or ES anions.

We calculated the radial distribution $\operatorname{RDF}(\mathrm{C}-\mathrm{C})$ between closest PEDOT backbone carbons for 6EDOT:PTS systems (Figure E.23 left picture). The $\mathrm{RDF}(\mathrm{C}-\mathrm{C})$ for untreated system display only two peaks, corresponding to a situation where at most 5-6 PEDOT units are in pi-pi stacking simultaneously locally like in presence of EMIM:Cl and EMIM:ES. However, in presence of EMIM:TCM, EMIM:TCB or EMIM:HCCP, a higher number of peaks appears up to $\mathrm{d} / \mathrm{d}_{\pi}=8$ which is consistent with the presence of a single cluster made of extended domains of PEDOT in $\pi-\pi$ stacking.


Figure E. $21-6$ EDOT cluster composition for 6EDOT:16SS large aqueous systems mixed with $\times 96$ pairs of IL. Cluster analysis of big aqueous system of tri-EDOT:PTS in absence of IL is displayed in red (6EDOT).


Figure E. 22 - Number of units in PEDOT domains in $\pi-\pi$ stacking using the PPS algorithm for large 6EDOT:PTS aqueous systems without (red bars) and mixed with $\times 96$ pairs of IL (blue bars).


Figure E. 23 - Right) Radial distribution RDF(C-C) between closest PEDOT backbone carbon for large 6EDOT:PTS systems mixed with $\times 96$ pairs of IL EMIM:X. Left) Radial distribution function between carbon atoms of PEDOT backbone and sulfonate group of PTS (RDF(C-S)) for 6EDOT:PTS large aqueous systems mixed with $\times 96$ pairs of IL EMIM:X.

## appendix F

# Supplemental material for the Umbrella Sampling calculation for tri-EDOT: $X$ and EMIM: X ion pairs and ion exchange energy 

## Contents

## F. 1 Error on the PMF calculations for ion pairs and ion exchange energy

We estimate the statistical error of each PMF calculated between tri-EDOT:X and EMIM:X where X is $\mathrm{Cl}, \mathrm{ES}, \mathrm{TCM}, \mathrm{TCB}, \mathrm{HCCP}$ or PTS. Subsequently, we deduce the error for the ion exchange free energy $\Delta \Delta G_{x}$. The standard deviation of each PMF is computed from 200 PMFs generated using the bayesian bootstrap technique described by Hub et al. [305] and implemented in the version 5.1.4 of gromacs. In a few words, the bayesian bootstrap technique generate new PMFs by selecting randomly histograms among the 200 histograms generated provided that there is overlapping between the randomly chosen histograms. In Figure F.1 we plotted the standard
deviation and the PMFs respectively for 3EDOT:X while the standard deviation and the PMF for EMIM:X is shown in Figure F.2.

The standard deviation calculated from 200 PMFs using the bayesian bootstrap technique converges when the reaction coordinate $\xi$ increases (blue full curves). We recall that the ion exchange free energy is expressed as following as a function of the binding free energy for each ion pair:

$$
\begin{align*}
\Delta \Delta G_{x} & =\Delta G_{b}(E M I M: P T S)+\Delta G_{b}(\operatorname{tri}-E D O T: X) \\
& -\Delta G_{b}(\operatorname{tri}-E D O T: P T S)-\Delta G_{b}(E M I M: X) \tag{F.1}
\end{align*}
$$

The standard deviation $\Delta[\ldots]$ for the ion exchange free energy $\Delta\left[\Delta \Delta G_{x}\right]$ is calculated by summing the standard deviation of each binding energy:

$$
\begin{align*}
\Delta\left[\Delta \Delta G_{x}\right] & =\Delta\left[\Delta G_{b}(E M I M: P T S)\right]+\Delta\left[\Delta G_{b}(\operatorname{tri}-E D O T: X)\right] \\
& +\Delta\left[\Delta G_{b}(\operatorname{tri}-E D O T: P T S)\right]+\Delta\left[\Delta G_{b}(E M I M: X)\right] \tag{F.2}
\end{align*}
$$

The standard deviation are shown again in Table F. 1 with the values of the ion binding free energies $\Delta G_{b}$ and the ion exchange free energies $\Delta \Delta G_{x}$.

Table F. 1 - Ion binding and ion exchange free energies ( $\mathrm{kJ} / \mathrm{mol}$ ).

| X | $\Delta G_{b}$ (EMIM:X) | $\Delta G_{b}$ (tri-EDOT:X) | $\Delta \Delta G_{x}^{a}$ |
| :---: | :---: | :---: | :---: |
| PTS | $-2.6 \pm 0.6$ | $-2.8 \pm 0.6$ |  |
| $\overline{\mathrm{C}}$ | -1.9 ${ }^{\text {土 }} \overline{0} . \overline{4}$ | $-\overline{1} . \overline{8} \pm 0.4$ | $\overline{0} . \overline{3} \pm{ }^{-}{ }^{-1}{ }^{-}$ |
| ES | $-2.8 \pm 0.5$ | $-3.0 \pm 0.6$ | $0.0 \pm 2.3$ |
| TCM | $-7.1 \pm 0.5$ | $-18.3 \pm 0.5$ | $-11.0 \pm 2.2$ |
| TCB | $-2.6 \pm 0.4$ | $-11.2 \pm 0.5$ | $-8.4 \pm 2.1$ |
| HCCP | $-3.2 \pm 0.5$ | $-17.3 \pm 1.3$ | $-13.9 \pm 3.0$ |

By taking account of the error for the ion exchange free energy $\Delta\left[\Delta \Delta G_{x}\right]$, we ensure that the dispersion of $\Delta \Delta G_{x}$ around the mean in presence of EMIM: Cl and EMIM:ES yields only range of positive values for $\Delta \Delta G_{x} \pm \Delta\left[\Delta \Delta G_{x}\right]$, indicating a limited ion exchange between tri-EDOT:PTS
and these ILs (lines 2 and 3 of Table F.1. Conversely, the dispersion around the mean in presence of EMIM:TCM, EMIM:TCB and EMIM:HCCP leads to negative values for $\Delta \Delta G_{x} \pm \Delta\left[\Delta \Delta G_{x}\right]$ which supports ion exchange between these ILs and tri-EDOT:PTS (lines 4,5 and 6 of Table F.1).


Figure F. 1 - Standard deviation of the potential of mean force calculated for tri-EDOT:X pairs.


Figure F. 2 - Standard deviation of the potential of mean force calculated for EMIM:X pairs.

## F. 2 Umbrella sampling histograms

The histograms of the biased simulations showing the motion of the anion restrained by the harmonic potential along the reaction coordinate $\xi$ are presented in this section. In Figure F.3 we present a typical set of histograms (200 histograms) used to derive the PMF. In particular, the plot displays the histograms used to compute the PMF for EMIM and ES ion pair. There is sufficient overlapping between histograms so that the PMF can be reconstructed properly by applying the weight histogram analysis method (WHAM).


Figure F. 3 - Histograms of the biased simulations for EMIM:ES ion pair.

## F. 3 PEDOT:X and EMIM:X complexes snapshots

We also displayed the equilibrated geometries of tri-EDOT:X complexes after water solvation in Figure F. 4 as well as initial geometries of EMIM:X complexes after water solvation in Figure F. 5 It is also clear from the snapshots that the COM-COM distance along the reaction coordinate (long side of the box) between tri-EDOT and PTS is higher than for tri-EDOT and ES. Also, the COM-COM distance along the reaction coordinate between EMIM and PTS is higher than for EMIM and ES.


Figure F. 4 - Equilibrated complexes of tri-EDOT:X in our US simulations.


Figure F. 5 - Equilibrated complexes of EMIM-EDOT:X in our US simulations.

## F. 4 Comparison between our PMFs/binding constant to data literature

We highlight in this section a step-by-step comparison between our calculations and those perform in the literature, in particular from the work of Yee and co-workers [307]. The PMF calculations and association constants for 3EDOT:X are new, but some of them involving the imidazolium-based IL EMIM:X has been investigated in the litterature.

First of all the principle of the calculation is to derive the potential of mean force (PMF) as explained in section 5.3.2 We used the umbrella sampling (US) technique in our work which was different from those used by Yee and co-workers that consists of the adaptative biasing force method (ABF). Instead of applying an harmonic biased potential to the hamiltonian of the system like in the US technique 3.3.5, the ABF method consists of applying a mean force opposed to the force along the reaction coordinate in order to derive the average force $\langle F(\xi)\rangle$ along the reaction coordinate $\xi$.

We notice first that the systems in the work of Yee and co-workers contains only 3150 waters and one IL pair (anion:cation) molecules while ours contain $\sim 6450$ water and one IL pair for a volume box of $\mathrm{V}=4 \mathrm{~nm} \times 4 \mathrm{~nm} \times 12 \mathrm{~nm}=192 \mathrm{~nm}^{3}$. The cencontration of IL in the PMF box is $c$ $=8.6 \mathrm{mMol} / \mathrm{l}$ and the density of IL is $\rho=0.005 \mathrm{ion} / \mathrm{nm}^{3}$. We suppose that the volume occupied by the IL pairs is small compared to water molecules and we deduce that the volume of the box used in the work of Yee and co-workers is $\mathrm{V} \sim 94 \mathrm{~nm}^{3}$. The cencontration of IL in the PMF box is $c=17 \mathrm{mMol} / \mathrm{l}$ and the density of IL is $\rho=0.010 \mathrm{ion} / \mathrm{nm}^{3}$.

Also, the setup of the cutoff for the van der Waals interactions in our work is 14 and the PME grid size is $1.0 \AA$ which is similar to the work of Yee and co-workers.

It is important to notice that the box side length is not specified in their work but the reaction coordinate length for which is the free energy is calculated is set up between 2-12 $\AA$ between the ions, which is smaller than ours in general (2.5-55 $\AA$ ). It is understandable that their box side length is much smaller along the reaction coordinate and that this small range would garantee that there is influence of PBC while calculating the PMF.

Hence, we follow their protocol and calculated the RDF from the PMF calculation. By refering $r$ as the distance between the two ions we used the following equation, considering that thee binding
free energy is equivalent to the PMF along the reaction coordinate $r\left(\mathrm{PMF} \equiv \Delta G_{b}(r)\right)$ :

$$
\begin{equation*}
\Delta G_{b}(r)=-k_{B} T \ln g(r) \tag{F.3}
\end{equation*}
$$

We display in Figure F. 6 the PMF obtained for EMIM:Cl and EMIM:ES both in our work and in the work of Yee and co-workers. Given that force-field parameters and box size are completely different between our simulations and theirs we note however that there is a common trend: the binding energy is higher between EMIM and ES than EMIM and Cl. The binding energies in our work was indeed -1.9 and $-2.8 \mathrm{~kJ} / \mathrm{mol}$ for EMIM:Cl and EMIM:ES while the work of Yee and co-workers the binding energies was -1.2 and $-3.9 \mathrm{~kJ} / \mathrm{mol}$.


Figure F. 6 - A) PMF for EMIM:Cl and EMIM:ES calculated by us. B) PMF for EMIM: Cl and EMIM:ES calculated by Yee and co-workers. Picture adapted from ref. [307] (J. Phys. Chem. B, 2013, 117, 12556-12566, Figures 2, Copyright 2020 with permission from ACS).

From the PMF, we extract the radial distribution function (RDF) according to the equation F. 3 which are shown in Figure F.7. picture A. The RDF are thus higher at short distance for EMIM:ES than for EMIM: Cl for both cases (see picture B). The interaction between the cation and the anion vanished when the RDF tends to $\sim 1$. We display also the corresponding CN in picture C ) and D ). Our results are consistent with the work of Yee and co-workers given that our CN integrated up to $\sim 12 \AA$ give a $\mathrm{CN} \approx 0.06$ (picture C ) which displays a $\mathrm{CN} \approx 0.02$ for EMIM: Cl and a $\mathrm{CN} \approx 0.05$ for EMIM:ES.

An important remark is that the coordination number CN integrated over the box does not tend toward one in practice. In order to highlight this let us remind the equation of the CN number:

$$
\begin{equation*}
C N\left(R_{e q}=\int_{0}^{r_{e q}} 4 \pi r^{2} \rho g(r) d r\right) \tag{F.4}
\end{equation*}
$$

The density of IL pair $\rho$ is a fixed value and at large value of $r$ (separation distance between cation and anion) the radial distribution tend to $\sim 1$. Hence, it is obvious that the CN will not tend to a fixed values as the integration of $\mathrm{g}(\mathrm{r}) \sim 1$ will give $\mathrm{CN} \sim \frac{4}{3} \pi \mathrm{r}^{3}$.


Figure F. 7 - A) RDF derived from PMF for EMIM:Cl and EMIM:ES calculated by us. B) RDF derived from PMF for EMIM: Cl and EMIM:ES calculated by Yee and co-workers. Pictures adapted from ref. [307] (J. Phys. Chem. B, 2013, 117, 12556-12566, Figures 3 and 5, Copyright 2020 with permission from ACS).

This trend can be oberved on Figure F.8 where we plot the CN from the RDF (see Figure 5.12) calculated by using equation F.4. It can be seen that the CN for tri-EDOT:TCB, tri-EDOT:TCM and tri-EDOT:HCCP display sharp increase at small value of $r$ corresponding to the peaks of the
corresponding rdf at such values (see Figure 5.12).


Figure F. 8 - A) CN calculated from RDF for EMIM:X and tri-EDOT:X in our work (zoomed at short values of r). B) CN calculated from RDF for EMIM:X and tri-EDOT: X in our work.

Let us remind the concentration of each species at equilibrium if we denote by $\alpha$, the degree of dissociation.

$$
\begin{equation*}
[C A]=(1-\alpha) c ; \quad\left[A^{-}\right]=\left[C^{+}\right]=\alpha c \tag{F.5}
\end{equation*}
$$

where the cation concentration is $\left[\mathrm{C}^{+}\right]$, anion concentration is $\left[\mathrm{A}^{-}\right]$and complex concentration is also [AC]. The association constant can be written in term of the degree of dissociation and concentration $c$ of IL:

$$
\begin{equation*}
K_{A}=\frac{(1-\alpha)}{\alpha^{2} c} \tag{F.6}
\end{equation*}
$$

The degree of dissociation can be estimated by the ratio over the number of available free ions at equilibrium $\mathrm{R}_{e q}$ (which is the CN number integrated up to $\mathrm{R}_{e q}$ ) over the total number of available
free ions (which is the CN number integrated up to $R \sim \infty$ ):

$$
\begin{equation*}
\alpha=1-\frac{\int_{0}^{R} 4 \pi r^{2} \rho g(r) d r}{\int_{0}^{\infty} 4 \pi r^{2} \rho g(r) d r} \tag{F.7}
\end{equation*}
$$

One has to be very careful for the above equation F.7 because Yee and co-workers assume that $\int_{0}^{\infty} 4 \pi r^{2} \rho g(r) d r \sim 1$ to compute the association constant. By doing the calculations, we found that the CN at large values was increasing like $\mathrm{CN} \sim \frac{4}{3} \pi \mathrm{r}^{3}$ because we derive the CN from the PMF calculations. Their assumption was a way to get rid of the actual box size because whatever the box size is the CN number would be always one in principle. We decide to follow their method in order to compare their results and ours.

Hence, from the CN number we thus calculate the association constant $K_{A}$ :

$$
\begin{equation*}
K_{A}=\frac{[C A]}{\left[C^{+}\right]\left[A^{-}\right]} \tag{F.8}
\end{equation*}
$$

which can be rewritten as:

$$
\begin{equation*}
K_{A}=\frac{(1-\alpha)}{\alpha^{2} c}=\frac{C N\left(R_{e q}\right)}{\left(1-C N\left(R_{e q}\right)\right)^{2} c} \tag{F.9}
\end{equation*}
$$

We thus used our CN calculated to deduce the association constant that we compare with those found by Yee and co-workers. We summarize the association constant for EMIM:Cl and EMIM:ES in Table.

Table F. 2 - Association constants for EMIM:Cl and EMIM:ES. Comparison between our results and results of Yee and Co-workers.

| Complex | $K_{A}$ in dm ${ }^{3}$.mol (Our work) | $K_{A}$ in $\mathrm{dm}^{3} . \mathrm{mol}$ (ref. [307]) |
| :---: | :---: | :---: |
| EMIM:Cl | 0.18 | $0.48 \pm 0.6$ |
| $\overline{\mathrm{E}} \overline{\mathrm{M}} \overline{\mathrm{I}} \overline{\mathrm{M}}: \overline{\mathrm{E}} \overline{\mathrm{S}}$ | $\overline{0} .50$ | $\overline{1} . \overline{3} 3 \overline{ \pm} \overline{0} . \overline{6}$ |

Altough we do not obtain the same values for the association constant for both cases because the box size and force field parameters used are different, the association constant is higher for EMIM:ES complex than for EMIM: Cl for both cases.

## Bibliography

[1] Venkata S Meka, Manprit KG Sing, Mallikarjuna R Pichika, Srinivasa R Nali, Venkata RM Kolapalli, and Prashant Kesharwani. A comprehensive review on polyelectrolyte complexes. Drug discovery today, 22(11):1697-1706, 2017.
[2] Abhijeet D Kulkarni, Yogesh H Vanjari, Karan H Sancheti, Harun M Patel, Veena S Belgamwar, Sanjay J Surana, and Chandrakantsing V Pardeshi. Polyelectrolyte complexes: mechanisms, critical experimental aspects, and applications. Artificial cells, nanomedicine, and biotechnology, 44(7):1615-1625, 2016.
[3] M Muthukumar. 50th anniversary perspective: A perspective on polyelectrolyte solutions. Macromolecules, 50(24):9528-9560, 2017.
[4] Damien Baigl. Etude expérimentale de polyélectrolytes hydrophobes modeles. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2003.
[5] Kris T Delaney and Glenn H Fredrickson. Theory of polyelectrolyte complexation-complex coacervates are self-coacervates. The Journal of Chemical Physics, 146(22):224902, 2017.
[6] Nadrian C Seeman and Hanadi F Sleiman. Dna nanotechnology. Nature Reviews Materials, 3(1):1-23, 2017.
[7] Marvin G Warner and James E Hutchison. Linear assemblies of nanoparticles electrostatically organized on dna scaffolds. Nature materials, 2(4):272-277, 2003.
[8] Melanie Homberger and Ulrich Simon. On the application potential of gold nanoparticles in nanoelectronics and biomedicine. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 368(1915):1405-1453, 2010.
[9] Kuan Sun, Shupeng Zhang, Pengcheng Li, Yijie Xia, Xiang Zhang, Donghe Du, Furkan Halis Isikgor, and Jianyong Ouyang. Review on application of pedots and pedot: Pss in energy conversion and storage devices. Journal of Materials Science: Materials in Electronics, 26(7):4438-4462, 2015.
[10] Mohammed Elmahmoudy. Micro-and nano-patterning of conducting polymers for biomedical applications. PhD thesis, 2017.
[11] Seyoung Kee, Nara Kim, Bong Seong Kim, Seongjin Park, Yun Hee Jang, Seoung Ho Lee, Jehan Kim, Junghwan Kim, Sooncheol Kwon, and Kwanghee Lee. Controlling molecular ordering in aqueous conducting polymers using ionic liquids. Advanced Materials, 28(39):86258631, 2016.
[12] Mei Ying Teo, Nara Kim, Seyoung Kee, Bong Seong Kim, Geunjin Kim, Soonil Hong, Suhyun Jung, and Kwanghee Lee. Highly stretchable and highly conductive pedot: Pss/ionic liquid composite transparent electrodes for solution-processed stretchable electronics. ACS applied materials 8 interfaces, 9(1):819-826, 2017.
[13] E Allahyarov, H Löwen, and G Gompper. Adsorption of monovalent and multivalent cations and anions on dna molecules. Physical Review E, 68(6):061903, 2003.
[14] JosÃ® L. F. Abascal and Juan Carlos Gil Montoro. Ionic distribution around simple B-DNA models. III. The effect of ionic charge. J. Chem. Phys., 114(9):4277-4284, February 2001.
[15] E Allahyarov, H Löwen, and G Gompper. Dna condensation, redissolution and mesocrystals induced by tetravalent counterions. EPL (Europhysics Letters), 68(6):894, 2004.
[16] E Allahyarov, G Gompper, and H Löwen. Attraction between dna molecules mediated by multivalent ions. Physical Review E, 69(4):041904, 2004.
[17] Alexander P Lyubartsev and Lars Nordenskiöld. Monte carlo simulation study of ion distribution and osmotic pressure in hexagonally oriented dna. The Journal of Physical Chemistry, 99(25):10373-10382, 1995.
[18] A Annunziato. Dna packaging: nucleosomes and chromatin. Nature Education, 1(1):26, 2008.
[19] David A Mortimer. Synthetic polyelectrolytes-a review. Polymer International, 25(1):29-41, 1991.
[20] Alec B Scranton, B Rangarajan, and J Klier. Biomedical applications of polyelectrolytes. In Biopolymers Ii, pages 1-54. Springer, 1995.
[21] Domenico Lombardo, Mikhail A Kiselev, Salvatore Magazù, and Pietro Calandra. Amphiphiles self-assembly: basic concepts and future perspectives of supramolecular approaches. Advances in Condensed Matter Physics, 2015, 2015.
[22] Salomé dos Santos, Bruno Medronho, Tiago dos Santos, and Filipe E Antunes. Amphiphilic molecules in drug delivery systems. In Drug delivery systems: advanced technologies potentially applicable in personalised treatment, pages 35-85. Springer, 2013.
[23] Fredric M Menger, Ashley L Galloway, and Mary E Chlebowski. Surface tension of aqueous amphiphiles. Langmuir, 21(20):9010-9012, 2005.
[24] Yu A Shashkina, Yu D Zaroslov, VA Smirnov, OE Philippova, AR Khokhlov, TA Pryakhina, and NA Churochkina. Hydrophobic aggregation in aqueous solutions of hydrophobically modified polyacrylamide in the vicinity of overlap concentration. Polymer, 44(8):2289-2293, 2003.
[25] William Bailey Russel, WB Russel, Dudley A Saville, and William Raymond Schowalter. Colloidal dispersions. Cambridge university press, 1991.
[26] Masao Doi, Samuel Frederick Edwards, and Samuel Frederick Edwards. The theory of polymer dynamics, volume 73. oxford university press, 1988.
[27] Paul J Flory. Principles of polymer chemistry. Cornell University Press, 1953.
[28] Andrey V Dobrynin, Ralph H Colby, and Michael Rubinstein. Scaling theory of polyelectrolyte solutions. Macromolecules, 28(6):1859-1871, 1995.
[29] P-Gd De Gennes, P Pincus, RM Velasco, and F Brochard. Remarks on polyelectrolyte conformation. Journal de physique, 37(12):1461-1473, 1976.
[30] W Kuhn, O Künzle, and A Katchalsky. Verhalten polyvalenter fadenmolekelionen in lösung. Helvetica Chimica Acta, 31(7):1994-2037, 1948.
[31] Gerald S Manning. Limiting laws and counterion condensation in polyelectrolyte solutions i. colligative properties. The journal of chemical Physics, 51(3):924-933, 1969.
[32] Ben O'Shaughnessy and Qingbo Yang. Manning-oosawa counterion condensation. Physical review letters, 94(4):048302, 2005.
[33] Gerald S Manning. Limiting laws and counterion condensation in polyelectrolyte solutions ii. self-diffusion of the small ions. The Journal of Chemical Physics, 51(3):934-938, 1969.
[34] Fumio Oosawa. Polyelectrolytes. M. Dekker, 1971.
[35] A Kossel. Uber die basischen stoffen des zellkerns (the basic substances of the cell nucleus). Z. Phys. Chem, 22:178, 1896.
[36] Jérémie Gummel. Structures et mécanismes de formation de complexes polyélectrolyteprotéine. PhD thesis, 2006.
[37] Paul Dubin, J Bock, Richie Davis, Donald N Schulz, and Curt Thies. Macromolecular complexes in chemistry and biology. Springer Science \& Business Media, 2012.
[38] Tracy W Perkins, Derek S Mak, Thatcher W Root, and Edwin N Lightfoot. Protein retention in hydrophobic interaction chromatography: modeling variation with buffer ionic strength and column hydrophobicity. Journal of Chromatography A, 766(1-2):1-14, 1997.
[39] Alan S Michaels and Richard G Miekka. Polycation-polyanion complexes: Preparation and properties of poly-(vinylbenzyltrimethylammonium) poly-(styrenesulfonate). The Journal of Physical Chemistry, 65(10):1765-1773, 1961.
[40] Jasper Van der Gucht, Evan Spruijt, Marc Lemmers, and Martien A Cohen Stuart. Polyelectrolyte complexes: Bulk phases and colloidal systems. Journal of colloid and interface science, 361(2):407-422, 2011.
[41] Iuliia Konko. Aqueous solutions of complexes formed by model polyelectrolytes of opposite charges. PhD thesis, 2015.
[42] Nicolas Laugel, Cosette Betscha, Mathias Winterhalter, Jean-Claude Voegel, Pierre Schaaf, and Vincent Ball. Relationship between the growth regime of polyelectrolyte multilayers and the polyanion/polycation complexation enthalpy. The journal of physical chemistry B, 110(39):19443-19449, 2006.
[43] Zhaoyang Ou and M. Muthukumar. Entropy and enthalpy of polyelectrolyte complexation: Langevin dynamics simulations. J. Chem. Phys., 124(15):154902, April 2006.
[44] Yan Hu, Tianming Yang, and Xianming Hu. Novel polysaccharides-based nanoparticle carriers prepared by polyelectrolyte complexation for protein drug delivery. Polymer bulletin, 68(4):1183-1199, 2012.
[45] Fabien Brand, Herbert Dautzenberg, Werner Jaeger, and Mathias Hahn. Polyelectrolytes with various charge densities: Synthesis and characterization of diallyldimethylammonium chloride-acrylamide copolymers. Die Angewandte Makromolekulare Chemie: Applied Macromolecular Chemistry and Physics, 248(1):41-71, 1997.
[46] Rungsima Chollakup, John B Beck, Klaus Dirnberger, Matthew Tirrell, and Claus D Eisenbach. Polyelectrolyte molecular weight and salt effects on the phase behavior and coacervation of aqueous solutions of poly (acrylic acid) sodium salt and poly (allylamine) hydrochloride. Macromolecules, 46(6):2376-2390, 2013.
[47] Alexander Kudlay, Alexander V Ermoshkin, and Monica Olvera de La Cruz. Complexation of oppositely charged polyelectrolytes: effect of ion pair formation. Macromolecules, 37(24):9231-9241, 2004.
[48] Hiroshi Fukuda and Yasuo Kikuchi. Polyelectrolyte complexes of sodium dextran sulfate with chitosan, 2. Die Makromolekulare Chemie: Macromolecular Chemistry and Physics, 178(10):2895-2899, 1977.
[49] Mandy Mende, Heide-Marie Buchhammer, Simona Schwarz, Gudrun Petzold, and Werner Jaeger. The stability of polyelectrolyte complex systems of poly (diallydimethyl-ammonium chloride) with different polyanions. In Macromolecular Symposia, volume 211, pages 121-134. Wiley Online Library, 2004.
[50] Christophe Schatz, Jean-Michel Lucas, Christophe Viton, Alain Domard, Christian Pichot, and Thierry Delair. Formation and properties of positively charged colloids based on polyelectrolyte complexes of biopolymers. Langmuir, 20(18):7766-7778, 2004.
[51] Caroline Ankerfors, Sedat Ondaral, Lars Wågberg, and Lars Ödberg. Using jet mixing to prepare polyelectrolyte complexes: Complex properties and their interaction with silicon oxide surfaces. Journal of colloid and interface science, 351(1):88-95, 2010.
[52] Aleksandr B Zezin and Viktor A Kabanov. A new class of complex water-soluble polyelectrolytes. Russian Chemical Reviews, 51(9):833, 1982.
[53] VA Kabanov and AB Zezin. Soluble interpolymeric complexes as a new class of synthetic polyelectrolytes. Pure and applied chemistry, 56(3):343-354, 1984.
[54] Eishun Tsuchida, Yoshihito Osada, and Kei Sanada. Interaction of poly (styrene sulfonate) with polycations carrying charges in the chain backbone. Journal of Polymer Science Part A-1: Polymer Chemistry, 10(11):3397-3404, 1972.
[55] Eishun Tsuchida, Yoshihito Osada, and Hiroyuki Ohno. Formation of interpolymer complexes. Journal of Macromolecular Science, Part B: Physics, 17(4):683-714, 1980.
[56] Andreas F Thünemann, Martin Müller, Herbert Dautzenberg, Jean-François Joanny, and Hartmut Löwen. Polyelectrolyte complexes. In Polyelectrolytes with defined molecular architecture II, pages 113-171. Springer, 2004.
[57] Caroline Ankerfors. Polyelectrolyte complexes: Preparation, characterization, and use for control of wet and dry adhesion between surfaces. PhD thesis, KTH Royal Institute of Technology, 2012.
[58] PL Dubin, J Gao, and K Mattison. Protein purification by selective phase separation with polyelectrolytes. Separation and Purification Methods, 23(1):1-16, 1994.
[59] Renko de Vries and Martien Cohen Stuart. Theory and simulations of macroion complexation. Current opinion in colloid $\xi$ interface science, 11(5):295-301, 2006.
[60] Thierry Delair. Colloidal polyelectrolyte complexes of chitosan and dextran sulfate towards versatile nanocarriers of bioactive molecules. European journal of pharmaceutics and biopharmaceutics, 78(1):10-18, 2011.
[61] Isabelle Morfin, Eric Buhler, Fabrice Cousin, Isabelle Grillo, and François Boué. Rodlike complexes of a polyelectrolyte (hyaluronan) and a protein (lysozyme) observed by sans. Biomacromolecules, 12(4):859-870, 2011.
[62] Lori K Sanders, Camilo Guáqueta, Thomas E Angelini, Jae-Wook Lee, Scott C Slimmer, Erik Luijten, and Gerard CL Wong. Structure and stability of self-assembled actin-lysozyme complexes in salty water. Physical review letters, 95(10):108302, 2005.
[63] Masahiko Annaka. Salt effect on microscopic structure and stability of colloidal complex obtained from neutral/polyelectrolyte block copolymer and oppositely charged surfactant. Colloids and Surfaces B: Biointerfaces, 99:127-135, 2012.
[64] Li Shi, Florent Carn, François Boué, Gervaise Mosser, and Eric Buhler. Control over the electrostatic self-assembly of nanoparticle semiflexible biopolyelectrolyte complexes. Soft Matter, 9(20):5004-5015, 2013.
[65] William W Gerberich, JM Jungk, and WM Mook. The bottom-up approach to materials by design. In Nano and Microstructural Design of Advanced Materials, pages 211-220. Elsevier, 2003.
[66] Wei Lu and Charles M Lieber. Nanoelectronics from the bottom up. In Nanoscience And Technology: A Collection of Reviews from Nature Journals, pages 137-146. World Scientific, 2010.
[67] Nadrian C Seeman. Dna nanotechnology: From the pub to information-based chemistry. In DNA Nanotechnology, pages 1-9. Springer, 2018.
[68] R Andrew McMillan. Nanoparticles align. Nature materials, 2(4):214-215, 2003.
[69] Gerd H Woehrle, Marvin G Warner, and James E Hutchison. Molecular-level control of feature separation in one-dimensional nanostructure assemblies formed by biomolecular nanolithography. Langmuir, 20(14):5982-5988, 2004.
[70] Victor A Bloomfield. Dna condensation. Current opinion in structural biology, 6(3):334-341, 1996.
[71] Vladimir B Teif and Klemen Bohinc. Condensed dna: condensing the concepts. Progress in biophysics and molecular biology, 105(3):208-222, 2011.
[72] K Yoshikawa, M Takahashi, VV Vasilevskaya, and AR Khokhlov. Large discrete transition in a single dna molecule appears continuous in the ensemble. Physical review letters, 76(16):3029, 1996.
[73] M Takahashi, K Yoshikawa, VV Vasilevskaya, and AR Khokhlov. Discrete coil- globule transition of single duplex dnas induced by polyamines. The Journal of Physical Chemistry B, 101(45):9396-9401, 1997.
[74] André Estévez-Torres and Damien Baigl. Dna compaction: fundamentals and applications. Soft Matter, 7(15):6746-6756, 2011.
[75] Tatsuo Akitaya, Asako Seno, Tonau Nakai, Norio Hazemoto, Shizuaki Murata, and Kenichi Yoshikawa. Weak interaction induces an on/off switch, whereas strong interaction causes gradual change: folding transition of a long duplex dna chain by poly-l-lysine. Biomacromolecules, 8(1):273-278, 2007.
[76] Leonard C Gosule and John A Schellman. Compact form of dna induced by spermidine. Nature, 259(5541):333-335, 1976.
[77] Robert Wilfred Wilson and Victor A Bloomfield. Counterion-induced condensation of deoxyribonucleic acid. a light-scattering study. Biochemistry, 18(11):2192-2196, 1979.
[78] R Ahmad, M Naoui, JF Neault, S Diamantoglou, and HA Tajmir-Riahi. An ftir spectroscopic study of calf-thymus dna complexation with al (iii) and ga (iii) cations. Journal of Biomolecular Structure and Dynamics, 13(5):795-802, 1996.
[79] Hirohumi Arakawa, Rohana Ahmad, Mancef Naoui, and Heidar-Ali Tajmir-Riahi. A comparative study of calf thymus dna binding to cr (iii) and cr (vi) ions evidence for the guanine n-7-chromium-phosphate chelate formation. Journal of Biological Chemistry, 275(14):1015010153, 2000.
[80] Heidar-Ali Tajmir-Riahi, Rohana Ahmad, and Mancef Naoui. Interaction of calf-thymus dna with trivalent la, eu, and tb ions. metal ion binding, dna condensation and structural features. Journal of Biomolecular Structure and Dynamics, 10(5):865-877, 1993.
[81] Yves Lansac, Jeril Degrouard, Madalena Renouard, Adriana C. Toma, FranÃ§oise Livolant, and Eric Raspaud. A route to self-assemble suspended DNA nano-complexes. Scientific Reports, 6:21995, February 2016.
[82] UK Laemmli. Characterization of dna condensates induced by poly (ethylene oxide) and polylysine. Proceedings of the National Academy of Sciences, 72(11):4288-4292, 1975.
[83] RS Dias, AACC Pais, MG Miguel, and Björn Lindman. Modeling of dna compaction by polycations. The Journal of chemical physics, 119(15):8150-8157, 2003.
[84] Wei Chen, Nicholas J Turro, and Donald A Tomalia. Using ethidium bromide to probe the interactions between dna and dendrimers. Langmuir, 16(1):15-19, 2000.
[85] Chen-Feng Ke, Sen Hou, Heng-Yi Zhang, Yu Liu, Kun Yang, and Xi-Zeng Feng. Controllable dna condensation through cucurbit [6] uril in 2d pseudopolyrotaxanes. Chemical communications, (32):3374-3376, 2007.
[86] Anatoly A Zinchenko, Kenichi Yoshikawa, and Damien Baigl. Compaction of single-chain dna by histone-inspired nanoparticles. Physical review letters, 95(22):228101, 2005.
[87] Anatoly A Zinchenko, Takahiro Sakaue, Sumiko Araki, Kenichi Yoshikawa, and Damien Baigl. Single-chain compaction of long duplex dna by cationic nanoparticles: modes of interaction and comparison with chromatin. The Journal of Physical Chemistry B, 111(11):30193031, 2007.
[88] LS Lerman. A transition to a compact form of dna in polymer solutions. Proceedings of the National Academy of Sciences, 68(8):1886-1890, 1971.
[89] UK Laemmli, JR Paulson, and V Hitchins. Maturation of the head of bacteriophage t4. v. a possible dna packaging mechanism: in vitro cleavage of the head proteins and the structure of the core of the polyhead. Journal of supramolecular structure, 2(2-4):276-301, 1974.
[90] Gerald S. Manning The molecular theory of polyelectrolyte solutions with applications to the electrostatic properties of polynucleotides Quarterly reviews of biophysics, 11(2):179-246, 1978.
[91] C. Ma and V. A. Bloomfield. Gel electrophoresis measurement of counterion condensation on DNA. Biopolymers: Original Research on Biomolecules, 35(2):211-216, 1995.
[92] John G Kirkwood and John B Shumaker. Forces between protein molecules in solution arising from fluctuations in proton charge and configuration. Proceedings of the National Academy of Sciences of the United States of America, 38(10):863, 1952.
[93] A Yu Grosberg, TT Nguyen, and BI Shklovskii. Colloquium: the physics of charge inversion in chemical and biological systems. Reviews of modern physics, 74(2):329, 2002.
[94] Yan Levin. Electrostatic correlations: from plasma to biology. Reports on progress in physics, 65(11):1577, 2002.
[95] Roland Kjellander, T Åkesson, Bo Jönsson, and Stjepan Marcelja. Double layer interactions in mono-and divalent electrolytes: A comparison of the anisotropic hypernetted chain theory and monte carlo simulations. The Journal of chemical physics, 97(2):1424-1431, 1992.
[96] Jan Forsman. A simple correlation-corrected poisson- boltzmann theory. The Journal of Physical Chemistry B, 108(26):9236-9245, 2004.
[97] Alexandre Diehl, MN Tamashiro, Marcia C Barbosa, and Yan Levin. Density-functional theory for attraction between like-charged plates. Physica A: Statistical Mechanics and its Applications, 274(3-4):433-445, 1999.
[98] Klemen Bohinc, Aleš Iglič, and Sylvio May. Interaction between macroions mediated by divalent rod-like ions. EPL (Europhysics Letters), 68(4):494, 2004.
[99] Stefano Maset and Klemen Bohinc. Orientations of dipoles restricted by two oppositely charged walls. Journal of Physics A: Mathematical and Theoretical, 40(39):11815, 2007.
[100] Jejoong Yoo, Hajin Kim, Aleksei Aksimentiev, and Taekjip Ha. Direct evidence for sequencedependent attraction between double-stranded dna controlled by methylation. Nature communications, 7:11045, 2016.
[101] Boris I Shklovskii. Screening of a macroion by multivalent ions: correlation-induced inversion of charge. Physical Review E, 60(5):5802, 1999.
[102] KVEK Besteman, K Van Eijk, and SG Lemay. Charge inversion accompanies dna condensation by multivalent ions. Nature Physics, 3(9):641-644, 2007.
[103] Eric Raspaud, J Pelta, M De Frutos, and F Livolant. Solubility and charge inversion of complexes of dna and basic proteins. Physical review letters, 97(6):068103, 2006.
[104] Margaret Haynes, R Garrett, and W Gratzer. Structure of nucleic acid-poly base complexes. Biochemistry, 9(22):4410-4416, 1970.
[105] Dhruba K Chattoraj, Leonard C Gosule, and John A Schellman. Dna condensation with polyamines: Ii. electron microscopic studies. Journal of molecular biology, 121(3):327-337, 1978.
[106] Nicholas V Hud and Kenneth H Downing. Cryoelectron microscopy of $\lambda$ phage dna condensates in vitreous ice: the fine structure of dna toroids. Proceedings of the National Academy of Sciences, 98(26):14925-14930, 2001.
[107] Victor A Bloomfield. Dna condensation by multivalent cations. Biopolymers: Original Research on Biomolecules, 44(3):269-282, 1997.
[108] Steven C Riemer and Victor A Bloomfield. Packaging of dna in bacteriophage heads: some considerations on energetics. Biopolymers: Original Research on Biomolecules, 17(3):785794, 1978.
[109] R Marquet and C Houssier. Thermodynamics of cation-induced dna condensation. Journal of Biomolecular Structure and Dynamics, 9(1):159-167, 1991.
[110] Christian Holm, Patrick Kékicheff, and Rudolf Podgornik. Electrostatic effects in soft matter and biophysics, volume 46. Springer Science \& Business Media, 2001.
[111] Stephane Flock, Rudi Labarbe, and Claude Houssier. Dielectric constant and ionic strength effects on dna precipitation. Biophysical journal, 70(3):1456-1465, 1996.
[112] Masanori Ueda and Kenichi Yoshikawa. Phase transition and phase segregation in a single double-stranded dna molecule. Physical review letters, 77(10):2133, 1996.
[113] VV Vasilevskaya, AR Khokhlov, Y Matsuzawa, and K Yoshikawa. Collapse of single dna molecule in poly (ethylene glycol) solutions. The Journal of chemical physics, 102(16):65956602, 1995.
[114] Pengxiang Zhao, Na Li, and Didier Astruc. State of the art in gold nanoparticle synthesis. Coordination Chemistry Reviews, 257(3-4):638-665, 2013.
[115] Günter Schmid. Nanoparticles: from theory to application. John Wiley \& Sons, 2011.
[116] Clemens Burda, Xiaobo Chen, and R Narayanan andM. A. el-sayed, chemistry and properties of nanocrystals of different shapes. Chem. Rev, 105:1025-1102, 2005.
[117] Challa SSR Kumar, Josef Hormes, and Carola Leuschner. Nanofabrication towards biomedical applications: techniques, tools, applications, and impact. John Wiley \& Sons, 2006.
[118] Marek Grzelczak. J. pe rez-juste, p. mulvaney and l.-m. liz-marza n. Chem Soc. Rev, 37:17831791, 2008.
[119] J Turkevich, PC Stevenson, and J Hiller. Synthesis of gold nanoparticles turkevich method. Discuss. Faraday Soc, 11:55-75, 1951.
[120] Han-Wen Cheng, Zakiya R Skeete, Elizabeth R Crew, Shiyao Shan, Jin Luo, and Chuan-Jian Zhong. Synthesis of gold nanoparticles. In Comprehensive Analytical Chemistry, volume 66, pages 37-79. Elsevier, 2014.
[121] Gert Frens. Controlled nucleation for the regulation of the particle size in monodisperse gold suspensions. Nature physical science, 241(105):20-22, 1973.
[122] Mathias Brust, Merryl Walker, Donald Bethell, David J Schiffrin, and Robin Whyman. Synthesis of thiol-derivatised gold nanoparticles in a two-phase liquid-liquid system. Journal of the Chemical Society, Chemical Communications, (7):801-802, 1994.
[123] Saji Alex and Ashutosh Tiwari. Functionalized gold nanoparticles: synthesis, properties and applications-a review. Journal of nanoscience and nanotechnology, 15(3):1869-1894, 2015.
[124] Sudhanshu Srivastava, Benjamin L Frankamp, and Vincent M Rotello. Controlled plasmon resonance of gold nanoparticles self-assembled with pamam dendrimers. Chemistry of materials, 17(3):487-490, 2005.
[125] Shu-Yi Lin, Chun-hsien Chen, Meng-Chieh Lin, and Hsiu-Fu Hsu. A cooperative effect of bifunctionalized nanoparticles on recognition: sensing alkali ions by crown and carboxylate moieties in aqueous media. Analytical chemistry, 77(15):4821-4828, 2005.
[126] Kołątaj, Karol and Krajczewski, Jan and Kudelski, Andrzej Plasmonic nanoparticles for environmental analysis. Environmental Chemistry Letters, 1-14, 2020.
[127] Saha, Krishnendu and Agasti, Sarit S and Kim, Chaekyu and Li, Xiaoning and Rotello, Vincent M Gold nanoparticles in chemical and biological sensing. Chemical reviews, 112(5):27392779, 2012.
[128] Enrique Morales-Avila, Guillermina Ferro-Flores, Blanca E Ocampo-García, and Leobardo M Gómez-Oliván. Engineered multifunctional rgd-gold nanoparticles for the detection of tumour-specific $\alpha(\nu) \beta(3)$ expression: Chemical characterisation and ecotoxicological risk assessment. Journal of biomedical nanotechnology, 8(6):991-999, 2012.
[129] Ralph Weissleder. A clearer vision for in vivo imaging. Nature biotechnology, 19(4):316-317, 2001.
[130] Minakshi Das, Kyu Hwan Shim, Seong Soo A An, and Dong Kee Yi. Review on gold nanoparticles and their applications. Toxicology and Environmental Health Sciences, 3(4):193-205, 2011.
[131] Toshihiko Sato, Haroon Ahmed, David Brown, and Brian FG Johnson. Single electron transistor using a molecularly linked gold colloidal particle chain. Journal of Applied Physics, 82(2):696-701, 1997.
[132] Hyung Jin Kim, Yonghan Roh, Seong Kyu Kim, and Byungyou Hong. Fabrication and characterization of dna-templated conductive gold nanoparticle chains. Journal of Applied Physics, 105(7):074302, 2009.
[133] Shawn M Douglas, Hendrik Dietz, Tim Liedl, Björn Högberg, Franziska Graf, and William M Shih. Self-assembly of dna into nanoscale three-dimensional shapes. Nature, 459(7245):414418, 2009.
[134] Amit Kumar, Jae-Ho Hwang, Sumit Kumar, and Jwa-Min Nam. Tuning and assembling metal nanostructures with DNA. Chemical Communications, 49(26):2597-2609, 2013.
[135] Sofia Julin, Sami Nummelin, Mauri A Kostiainen, and Veikko Linko. Dna nanostructuredirected assembly of metal nanoparticle superlattices. Journal of Nanoparticle Research, 20(5):119, 2018.
[136] Hyung Jin Kim, Yonghan Roh, and Byungyou Hong. Controlled gold nanoparticle assembly on dna molecule as template for nanowire formation. Journal of Vacuum Science $\mathcal{B}$ Technology A: Vacuum, Surfaces, and Films, 24(4):1327-1331, 2006.
[137] Caroline Hartl, Kilian Frank, Heinz Amenitsch, Stefan Fischer, Tim Liedl, and Bert Nickel. Position Accuracy of Gold Nanoparticles on DNA Origami Structures Studied with SmallAngle X-ray Scattering. Nano Lett., 18(4):2609-2615, April 2018.
[138] Fatih N Gü̈, Friedrich W Schwarz, Jingjing Ye, Stefan Diez, and Thorsten L Schmidt. Toward self-assembled plasmonic devices: high-yield arrangement of gold nanoparticles on dna origami templates. ACS nano, 10(5):5374-5382, 2016.
[139] E. P. Gates, J. K. Jensen, J. N. Harb, and A. T. Woolley. Optimizing gold nanoparticle seeding density on DNA origami. RSC Adv., 5(11):8134-8141, January 2015.
[140] Paul WK Rothemund. Folding dna to create nanoscale shapes and patterns. Nature, 440(7082):297-302, 2006.
[141] Murali Sastry, Ashavani Kumar, Suvarna Datar, C. V. Dharmadhikari, and Krishna N. Ganesh. DNA-mediated electrostatic assembly of gold nanoparticles into linear arrays by a simple drop-coating procedure. Appl. Phys. Lett., 78(19):2943-2945, May 2001.
[142] Marvin G. Warner and James E. Hutchison. Linear assemblies of nanoparticles electrostatically organized on DNA scaffolds. Nature Materials, 2(4):272-277, April 2003.
[143] Gangli Wang and Royce W Murray. Controlled assembly of monolayer-protected gold clusters by dissolved dna. Nano Letters, 4(1):95-101, 2004.
[144] Zhaoxiang Deng, Ye Tian, Seung-Hyun Lee, Alexander E Ribbe, and Chengde Mao. Dnaencoded self-assembly of gold nanoparticles into one-dimensional arrays. Angewandte Chemie International Edition, 44(23):3582-3585, 2005.
[145] Stefan Beyer, Patrick Nickels, and Friedrich C Simmel. Periodic dna nanotemplates synthesized by rolling circle amplification. Nano letters, 5(4):719-722, 2005.
[146] Sofia Julin, Antti Korpi, Boxuan Shen, Ville Liljeström, Olli Ikkala, Adrian Keller, Veikko Linko, Mauri A Kostiainen, et al. Dna origami directed 3d nanoparticle superlattice via electrostatic assembly. Nanoscale, 11(10):4546-4551, 2019.
[147] Magatte Gueye. Propriétés électriques, optoélectroniques et thermoélectriques de matériaux à base de poly (3, 4-éthylènedioxythiophène) PEDOT. PhD thesis, 2017.
[148] R McNeill, R Siudak, JH Wardlaw, and DE Weiss. Electronic conduction in polymers. i. the chemical structure of polypyrrole. Australian Journal of Chemistry, 16(6):1056-1075, 1963.
[149] Hideki Shirakawa, Edwin J Louis, Alan G MacDiarmid, Chwan K Chiang, and Alan J Heeger. Synthesis of electrically conducting organic polymers: halogen derivatives of polyacetylene,(ch) x. Journal of the Chemical Society, Chemical Communications, (16):578-580, 1977.
[150] CK Chiang, CR Fincher Jr, YW Park, AJ Heeger, H Shirakawa, EJ Louis, SC Gau, and Alan G MacDiarmid. Electrical conductivity in doped polyacetylene. Physical Review Letters, 40(22):1472, 1978.
[151] CK Chiang, Young-Woo Park, AJ Heeger, Hideki Shirakawa, EJ Louis, and Alan G MacDiarmid. Conducting polymers: Halogen doped polyacetylene. The Journal of Chemical Physics, 69(11):5098-5104, 1978.
[152] Andreas Elschner, Stephan Kirchmeyer, Wilfried Lovenich, Udo Merker, and Knud Reuter. PEDOT: principles and applications of an intrinsically conductive polymer. CRC press, 2010.
[153] Siegmar Roth and Hartmut Bleier. Solitons in polyacetylene. Advances in Physics, 36(4):385462, 1987.
[154] Susumu Hara, Tetsuji Zama, Wataru Takashima, and Keiichi Kaneto. Artificial muscles based on polypyrrole actuators with large strain and stress induced electrically. Polymer journal, 36(2):151-161, 2004.
[155] S Bocchini, A Chiolerio, S Porro, D Accardo, N Garino, K Bejtka, D Perrone, and CF Pirri. Synthesis of polyaniline-based inks, doping thereof and test device printing towards electronic applications. Journal of Materials Chemistry C, 1(33):5101-5109, 2013.
[156] Xi Yin, Hoon T Chung, Ulises Martinez, Ling Lin, Kateryna Artyushkova, and Piotr Zelenay. Pgm-free orr catalysts designed by templating pani-type polymers containing functional groups with high affinity to iron. Journal of The Electrochemical Society, 166(7):F3240, 2019.
[157] Jean Roncali. Conjugated poly (thiophenes): synthesis, functionalization, and applications. Chemical Reviews, 92(4):711-738, 1992.
[158] Lee A Fielding, Jon K Hillier, Mark J Burchell, and Steven P Armes. Space science applications for conducting polymer particles: synthetic mimics for cosmic dust and micrometeorites. Chemical Communications, 51(95):16886-16899, 2015.
[159] Florian Jurin. Croissance et propriétés de films minces conducteurs par auto-assemblage de polyélectrolytes. PhD thesis, 2014.
[160] Thanh-Hai Le, Yukyung Kim, and Hyeonseok Yoon. Electrical and electrochemical properties of conducting polymers. Polymers, 9(4):150, 2017.
[161] Xinyan Cui and David C Martin. Electrochemical deposition and characterization of poly (3, 4-ethylenedioxythiophene) on neural microelectrode arrays. Sensors and Actuators B: Chemical, 89(1-2):92-102, 2003.
[162] Michael Mueller, Manrico Fabretto, Drew Evans, Pejman Hojati-Talemi, Cynthia Gruber, and Peter Murphy. Vacuum vapour phase polymerization of high conductivity pedot: Role of peg-ppg-peg, the origin of water, and choice of oxidant. Polymer, 53(11):2146-2151, 2012.
[163] G Zotti, S Zecchin, G Schiavon, F Louwet, L Groenendaal, Xavier Crispin, Wojciech Osikowicz, W Salaneck, and Mats Fahlman. Electrochemical and xps studies toward the role of monomeric and polymeric sulfonate counterions in the synthesis, composition, and properties of poly (3, 4-ethylenedioxythiophene). Macromolecules, 36(9):3337-3344, 2003.
[164] Oleg P Dimitriev, Yuri P Piryatinski, and Alexander A Pud. Evidence of the controlled interaction between pedot and pss in the pedot: Pss complex via concentration changes of the complex solution. The Journal of Physical Chemistry B, 115(6):1357-1362, 2011.
[165] Yuechen Li, Ryo Tanigawa, and Hidenori Okuzaki. Soft and flexible pedot/pss films for applications to soft actuators. Smart Materials and Structures, 23(7):074010, 2014.
[166] Hu Yan, Susumu Arima, Yusaku Mori, Tsubasa Kagata, Hiroshi Sato, and Hidenori Okuzaki. Poly (3, 4-ethylenedioxythiophene)/poly (4-styrenesulfonate): Correlation between colloidal particles and thin films. Thin Solid Films, 517(11):3299-3303, 2009.
[167] Claudia M Palumbiny, Johannes Schlipf, Alexander Hexemer, Cheng Wang, and Peter Müller-Buschbaum. The morphological power of soap: how surfactants lower the sheet resistance of pedot: Pss by strong impact on inner film structure and molecular interface orientation. Advanced Electronic Materials, 2(4):1500377, 2016.
[168] Nara Kim, Seyoung Kee, Seoung Ho Lee, Byoung Hoon Lee, Yung Ho Kahng, Yong-Ryun Jo, Bong-Joong Kim, and Kwanghee Lee. Highly conductive pedot: Pss nanofibrils induced by solution-processed crystallization. Advanced materials, 26(14):2268-2272, 2014.
[169] Udo Lang, Elisabeth Müller, Nicola Naujoks, and Jurg Dual. Microscopical investigations of pedot: Pss thin films. Advanced Functional Materials, 19(8):1215-1220, 2009.
[170] KE Aasmundtveit, EJ Samuelsen, LAA Pettersson, O Inganäs, T Johansson, and R Feidenhans. Structure of thin films of poly (3, 4-ethylenedioxythiophene). Synthetic Metals, 101(1-3):561-564, 1999.
[171] Olga Bubnova, Zia Ullah Khan, Hui Wang, Slawomir Braun, Drew R Evans, Manrico Fabretto, Pejman Hojati-Talemi, Daniel Dagnelund, Jean-Baptiste Arlin, Yves H Geerts, et al. Semi-metallic polymers. Nature materials, 13(2):190-194, 2014.
[172] Boram Cho, Kyung S Park, Jangmi Baek, Hyun S Oh, Yong-Eun Koo Lee, and Myung M Sung. Single-crystal poly (3, 4-ethylenedioxythiophene) nanowires with ultrahigh conductivity. Nano letters, 14(6):3321-3327, 2014.
[173] Fengling Zhang, Mikael Johansson, Mats R Andersson, Jan C Hummelen, and Olle Inganäs. Polymer photovoltaic cells with conducting polymer anodes. Advanced Materials, 14(9):662665, 2002.
[174] Zhike Liu, Jinhua Li, and Feng Yan. Package-free flexible organic solar cells with graphene top electrodes. Advanced Materials, 25(31):4296-4301, 2013.
[175] Bjorn Vaagensmith, Khan Mamun Reza, MD Nazmul Hasan, Hytham Elbohy, Nirmal Adhikari, Ashish Dubey, Nick Kantack, Eman Gaml, and Qiquan Qiao. Environmentally friendly plasma-treated pedot: Pss as electrodes for ito-free perovskite solar cells. ACS applied materials $\mathfrak{E}^{\text {interfaces, }} 9$ (41):35861-35870, 2017.
[176] Jungjin Yoon, Hyangki Sung, Gunhee Lee, Woohyung Cho, Namyoung Ahn, Hyun Suk Jung, and Mansoo Choi. Superflexible, high-efficiency perovskite solar cells utilizing graphene electrodes: towards future foldable power sources. Energy \& Environmental Science, 10(1):337345, 2017.
[177] Xi Fan, Wanyi Nie, Hsinhan Tsai, Naixiang Wang, Huihui Huang, Yajun Cheng, Rongjiang Wen, Liujia Ma, Feng Yan, and Yonggao Xia. Pedot: Pss for flexible and stretchable electronics: modifications, strategies, and applications. Advanced Science, 6(19):1900813, 2019.
[178] Zhong Ma, Sheng Li, Huiting Wang, Wen Cheng, Yun Li, Lijia Pan, and Yi Shi. Advanced electronic skin devices for healthcare applications. Journal of Materials Chemistry B, 7(2):173-197, 2019.
[179] Lambertus Groenendaal, Friedrich Jonas, Dieter Freitag, Harald Pielartzik, and John R Reynolds. Poly (3, 4-ethylenedioxythiophene) and its derivatives: past, present, and future. Advanced materials, 12(7):481-494, 2000.
[180] Jianyong Ouyang, Qianfei Xu, Chi-Wei Chu, Yang Yang, Gang Li, and Joseph Shinar. On the mechanism of conductivity enhancement in poly (3, 4-ethylenedioxythiophene): poly (styrene sulfonate) film through solvent treatment. Polymer, 45(25):8443-8450, 2004.
[181] Benhu Fan, Xiaoguang Mei, and Jianyong Ouyang. Significant conductivity enhancement of conductive poly (3, 4-ethylenedioxythiophene): poly (styrenesulfonate) films by adding anionic surfactants into polymer solution. Macromolecules, 41(16):5971-5973, 2008.
[182] Yijie Xia and Jianyong Ouyang. Salt-induced charge screening and significant conductivity enhancement of conducting poly (3, 4-ethylenedioxythiophene): poly (styrenesulfonate). Macromolecules, 42(12):4141-4147, 2009.
[183] Erol Yildirim, Gang Wu, Xue Yong, Teck Leong Tan, Qiang Zhu, Jianwei Xu, Jianyong Ouyang, Jian-Sheng Wang, and Shuo-Wang Yang. A theoretical mechanistic study on electrical conductivity enhancement of dmso treated pedot: Pss. Journal of Materials Chemistry C, 6(19):5122-5131, 2018.
[184] JY Kim, JH Jung, DE Lee, and Jinsoo Joo. Enhancement of electrical conductivity of poly (3, 4-ethylenedioxythiophene)/poly (4-styrenesulfonate) by a change of solvents. Synthetic Metals, 126(2-3):311-316, 2002.
[185] Khashayar Ghandi. A review of ionic liquids, their limits and applications. Green and sustainable chemistry, 2014, 2014.
[186] P Walden. Bull. acad. imper. sci.(st. petersburg). 1914.
[187] Jae-Kwang Kim, Aleksandar Matic, Jou-Hyeon Ahn, and Per Jacobsson. An imidazolium based ionic liquid electrolyte for lithium batteries. Journal of Power Sources, 195(22):76397643, 2010.
[188] Michel Armand, Frank Endres, Douglas R MacFarlane, Hiroyuki Ohno, and Bruno Scrosati. Ionic-liquid materials for the electrochemical challenges of the future. In Materials For Sustainable Energy: A Collection of Peer-Reviewed Research and Review Articles from Nature Publishing Group, pages 129-137. World Scientific, 2011.
[189] Ricardo Ferraz, Luís C Branco, Cristina Prudencio, Joao Paulo Noronha, and Željko Petrovski. Ionic liquids as active pharmaceutical ingredients. ChemMedChem, 6(6):975-985, 2011.
[190] Malika Boukherissa, Fabrice Mutelet, Ali Modarressi, Amadou Dicko, Driss Dafri, and Marek Rogalski. Ionic liquids as dispersants of petroleum asphaltenes. Energy \& Fuels, 23(5):25572564, 2009.
[191] Daniel J Tempel, Philip B Henderson, Jeffrey R Brzozowski, Ronald M Pearlstein, and Hansong Cheng. High gas storage capacities for ionic liquids through chemical complexation. Journal of the American Chemical Society, 130(2):400-401, 2008.
[192] Michel Picquet, Igor Tkatchenko, Immacolata Tommasi, Peter Wasserscheid, and Jörg Zimmermann. Ionic liquids, 3. synthesis and utilisation of protic imidazolium salts in homogeneous catalysis. Advanced Synthesis E Catalysis, 345(8):959-962, 2003.
[193] Jacob M Crosthwaite, Mark J Muldoon, JaNeille K Dixon, Jessica L Anderson, and Joan F Brennecke. Phase transition and decomposition temperatures, heat capacities and viscosities of pyridinium ionic liquids. The Journal of Chemical Thermodynamics, 37(6):559-568, 2005.
[194] John D Holbrey and Kenneth R Seddon. The phase behaviour of 1-alkyl-3-methylimidazolium tetrafluoroborates; ionic liquids and ionic liquid crystals. Journal of the Chemical Society, Dalton Transactions, (13):2133-2140, 1999.
[195] Cinzia Chiappe and Daniela Pieraccini. Ionic liquids: solvent properties and organic reactivity. Journal of physical organic chemistry, 18(4):275-297, 2005.
[196] Yun-Sheng Ye, John Rick, and Bing-Joe Hwang. Ionic liquid polymer electrolytes. Journal of Materials Chemistry A, 1(8):2719-2743, 2013.
[197] Frank Endres, Andrew Abbott, and Douglas R MacFarlane. Electrodeposition from ionic liquids. John Wiley \& Sons, 2017.
[198] Frank Endres, Mirko Bukowski, Rolf Hempelmann, and Harald Natter. Electrodeposition of nanocrystalline metals and alloys from ionic liquids. Angewandte Chemie International Edition, 42(29):3428-3430, 2003.
[199] Mariana Díaz, Alfredo Ortiz, and Inmaculada Ortiz. Progress in the use of ionic liquids as electrolyte membranes in fuel cells. Journal of Membrane Science, 469:379-396, 2014.
[200] Chantal Badre, Ludovic Marquant, Ahmed M Alsayed, and Lawrence A Hough. Highly conductive poly (3, 4-ethylenedioxythiophene): poly (styrenesulfonate) films using 1-ethyl-3-methylimidazolium tetracyanoborate ionic liquid. Advanced Functional Materials, 22(13):2723-2727, 2012.
[201] Ambroise de Izarra, Seongjin Park, Jinhee Lee, Yves Lansac, and Yun Hee Jang. Ionic Liquid Designed for PEDOT:PSS Conductivity Enhancement. J. Am. Chem. Soc., 140(16):53755384, April 2018.
[202] Sebastian Kmiecik, Dominik Gront, Michal Kolinski, Lukasz Wieteska, Aleksandra Elzbieta Dawid, and Andrzej Kolinski. Coarse-grained protein models and their applications. Chemical reviews, 116(14):7898-7936, 2016.
[203] Sharon C Glotzer and Wolfgang Paul. Molecular and mesoscale simulation methods for polymer materials. Annual Review of Materials Research, 32(1):401-436, 2002.
[204] Arthur F Voter. Introduction to the kinetic monte carlo method. In Radiation effects in solids, pages 1-23. Springer, 2007.
[205] N Metropolis. The beginning of the. Los Alamos Science, 15:125-30, 1987.
[206] Eckhardt Roger. Stan ulam, john von neumann and the monte carlo method. Argonne, USA, 1987.
[207] Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. Journal of the American statistical association, 44(247):335-341, 1949.
[208] Daan Frenkel and Berend Smit. Understanding molecular simulation: from algorithms to applications, volume 1. Elsevier, 2001.
[209] WW Wood. Monte carlo calculations for hard disks in the isothermal-isobaric ensemble. The Journal of Chemical Physics, 48(1):415-434, 1968.
[210] IR McDonald. Npt-ensemble monte carlo calculations for binary liquid mixtures. Molecular Physics, 23(1):41-58, 1972.
[211] GE Norman and VS Filinov. Investigations of phase transitions by a monte-carlo method. High Temperature, 7(2):216, 1969.
[212] Martin Oliver Steinhauser. A molecular dynamics study on universal properties of polymer chains in different solvent qualities. part i. a review of linear chain properties. The Journal of chemical physics, 122(9):094901, 2005.
[213] Moti Lal. Monte carlo computer simulation of chain molecules. i. Molecular physics, 17(1):5764, 1969.
[214] Stephen Whitelam and Phillip L Geissler. Avoiding unphysical kinetic traps in monte carlo simulations of strongly attractive particles. The Journal of chemical physics, 127(15):154101, 2007.
[215] Štěpán Ružička and Michael P Allen. Collective translational and rotational monte carlo cluster move for general pairwise interaction. Physical Review E, 90(3):033302, 2014.
[216] Christophe Dress and Werner Krauth. Cluster algorithm for hard spheres and related systems. Journal of Physics A: Mathematical and General, 28(23):L597, 1995.
[217] Jiwen Liu and Erik Luijten. Rejection-free geometric cluster algorithm for complex fluids. Physical review letters, 92(3):035504, 2004.
[218] Heather L Gordon and John P Valleau. A 'partially clothed' pivot algorithm for model polyelectrolyte solutions. Molecular Simulation, 14(6):361-379, 1995.
[219] Vladimir Lobaskin and Per Linse. Simulation of an asymmetric electrolyte with charge asymmetry 60: 1 using hard-sphere and soft-sphere models. The Journal of chemical physics, 111(9):4300-4309, 1999.
[220] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. The journal of chemical physics, 21(6):1087-1092, 1953.
[221] Roger W Hockney and James W Eastwood. Computer simulation using particles. crc Press, 1988.
[222] Michael P Allen and Dominic J Tildesley. Computer simulation of liquids. Oxford university press, 2017.
[223] Tom Darden, Darrin York, and Lee Pedersen. Particle mesh ewald: An n log (n) method for ewald sums in large systems. The Journal of chemical physics, 98(12):10089-10092, 1993.
[224] Ulrich Essmann, Lalith Perera, Max L Berkowitz, Tom Darden, Hsing Lee, and Lee G Pedersen. A smooth particle mesh ewald method. The Journal of chemical physics, 103(19):85778593, 1995.
[225] Ben Widom. Some topics in the theory of fluids. The Journal of Chemical Physics, 39(11):2808-2812, 1963.
[226] Loup Verlet. Computer" experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. Physical review, 159(1):98, 1967.
[227] William C Swope, Hans C Andersen, Peter H Berens, and Kent R Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. The Journal of Chemical Physics, 76(1):637-649, 1982.
[228] Bernard Diu, Bernard Roulet, Claudine Guthmann, and Danielle Lederer. Eléments de physique statistique. Hermann, 1989.
[229] Herman JC Berendsen, JPM van Postma, Wilfred F van Gunsteren, ARHJ DiNola, and Jan R Haak. Molecular dynamics with coupling to an external bath. The Journal of chemical physics, 81(8):3684-3690, 1984.
[230] Shuichi Nosé. A unified formulation of the constant temperature molecular dynamics methods. The Journal of chemical physics, 81(1):511-519, 1984.
[231] William G Hoover. Canonical dynamics: Equilibrium phase-space distributions. Physical review $A$, 31(3):1695, 1985.
[232] Michele Parrinello and Aneesur Rahman. Polymorphic transitions in single crystals: A new molecular dynamics method. Journal of Applied physics, 52(12):7182-7190, 1981.
[233] Vikas Dubey and Snehasis Daschakraborty. Influence of glycerol on the cooling effect of pair hydrophobicity in water: relevance to proteins' stabilization at low temperature. Physical Chemistry Chemical Physics, 21(2):800-812, 2019.
[234] Ludovic Garnier, Julien Devémy, Christine Bonal, and Patrice Malfreyt. Calculations of potential of mean force: application to ion-pairs and host-guest systems. Molecular Physics, 116(15-16):1998-2008, 2018.
[235] Glenn M Torrie and John P Valleau. Nonphysical sampling distributions in monte carlo freeenergy estimation: Umbrella sampling. Journal of Computational Physics, 23(2):187-199, 1977.
[236] Scott Shell. Biased sampling and related free energy techniques, ChE210D, 2019. https: //sites.engineering.ucsb.edu/~shell/che210d/Biased_sampling.pdf.
[237] Shankar Kumar, John M Rosenberg, Djamal Bouzida, Robert H Swendsen, and Peter A Kollman. The weighted histogram analysis method for free-energy calculations on biomolecules. i. the method. Journal of computational chemistry, 13(8):1011-1021, 1992.
[238] Alan Grossfield. Wham: the weighted histogram analysis method, version 2.0. 9. Available at membrane. urmc. rochester. edu/content/wham. Accessed November, 15:2013, 2013.
[239] David Van Der Spoel, Erik Lindahl, Berk Hess, Gerrit Groenhof, Alan E Mark, and Herman JC Berendsen. Gromacs: fast, flexible, and free. Journal of computational chemistry, 26(16):1701-1718, 2005.
[240] James C Phillips, Rosemary Braun, Wei Wang, James Gumbart, Emad Tajkhorshid, Elizabeth Villa, Christophe Chipot, Robert D Skeel, Laxmikant Kale, and Klaus Schulten. Scalable molecular dynamics with namd. Journal of computational chemistry, 26(16):1781-1802, 2005.
[241] Steve Plimpton, Paul Crozier, and Aidan Thompson. Lammps-large-scale atomic/molecular massively parallel simulator. Sandia National Laboratories, 18:43, 2007.
[242] Bernard R Brooks, Charles L Brooks III, Alexander D Mackerell Jr, Lennart Nilsson, Robert J Petrella, Benoît Roux, Youngdo Won, Georgios Archontis, Christian Bartels, Stefan Boresch, et al. Charmm: the biomolecular simulation program. Journal of computational chemistry, 30(10):1545-1614, 2009.
[243] DA Pearlman, DA Case, JW Caldwell, WS Ross, TE Cheatham, DM Ferguson, GL Seibel, UC Singh, PK Weiner, and PA Kollman. Amber 5.0. University of California, San Francisco, 1997.
[244] Wei Jiang, David J Hardy, James C Phillips, Alexander D MacKerell Jr, Klaus Schulten, and Benoît Roux. High-performance scalable molecular dynamics simulations of a polarizable force field based on classical drude oscillators in namd. The journal of physical chemistry letters, 2(2):87-92, 2011.
[245] Nathan Schmid, Andreas P Eichenberger, Alexandra Choutko, Sereina Riniker, Moritz Winger, Alan E Mark, and Wilfred F van Gunsteren. Definition and testing of the gromos force-field versions 54a7 and 54b7. European biophysics journal, 40(7):843, 2011.
[246] Alex D MacKerell Jr, Donald Bashford, MLDR Bellott, Roland Leslie Dunbrack Jr, Jeffrey D Evanseck, Martin J Field, Stefan Fischer, Jiali Gao, H Guo, Sookhee Ha, et al. All-atom empirical potential for molecular modeling and dynamics studies of proteins. The journal of physical chemistry B, 102(18):3586-3616, 1998.
[247] James A Maier, Carmenza Martinez, Koushik Kasavajhala, Lauren Wickstrom, Kevin E Hauser, and Carlos Simmerling. ff14sb: improving the accuracy of protein side chain and backbone parameters from ff99sb. Journal of chemical theory and computation, 11(8):36963713, 2015.
[248] William L Jorgensen, David S Maxwell, and Julian Tirado-Rives. Development and testing of the opls all-atom force field on conformational energetics and properties of organic liquids. Journal of the American Chemical Society, 118(45):11225-11236, 1996.
[249] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. SIAM Journal on scientific computing, 16(5):1190-1208, 1995.
[250] Kensal E Van Holde. Chromatin. Springer Science \& Business Media, 2012.
[251] JL Sikorav, J Pelta, and F Livolant. A liquid crystalline phase in spermidine-condensed dna. Biophysical journal, 67(4):1387-1392, 1994.
[252] Matthew R Jones, Nadrian C Seeman, and Chad A Mirkin. Programmable materials and the nature of the dna bond. Science, 347(6224):1260901, 2015.
[253] Sofia Julin, Antti Korpi, Nonappa, Boxuan Shen, Ville LiljestrÃๆm, Olli Ikkala, Adrian Keller, Veikko Linko, and Mauri A. Kostiainen. DNA origami directed 3d nanoparticle superlattice via electrostatic assembly. Nanoscale, 11(10):4546-4551, March 2019.
[254] E Allahyarov and H Löwen. Effective interaction between helical biomolecules. Physical Review E, 62(4):5542, 2000.
[255] Elshad Allahyarov, Gerhard Gompper, and Hartmut Löwen. Dna condensation and redissolution: interaction between overcharged dna molecules. Journal of Physics: Condensed Matter, 17(20):S1827, 2005.
[256] Alexander P. Lyubartsev and Lars Nordenskioeld. Monte Carlo Simulation Study of Ion Distribution and Osmotic Pressure in Hexagonally Oriented DNA. J. Phys. Chem., 99(25):10373-10382, June 1995.
[257] Alexei A Kornyshev and Sergey Leikin. Electrostatic interaction between helical macromolecules in dense aggregates: an impetus for dna poly-and meso-morphism. Proceedings of the National Academy of Sciences, 95(23):13579-13584, 1998.
[258] Alexander P Lyubartsev and Lars Nordenskiöld. Monte carlo simulation study of dna polyelectrolyte properties in the presence of multivalent polyamine ions. The Journal of Physical Chemistry B, 101(21):4335-4342, 1997.
[259] JC Gil Montoro and JLF Abascal. Ionic distribution around simple dna models. i. cylindrically averaged properties. The Journal of chemical physics, 103(18):8273-8284, 1995.
[260] Dhiraj Kumar, Brian J Meenan, Isha Mutreja, RAECHELLE D'SA, and Dorian Dixon. Controlling the size and size distribution of gold nanoparticles: a design of experiment study. International Journal of Nanoscience, 11(02):1250023, 2012.
[261] Jia-Qi Lin, Yong-Gang Zheng, Hong-Wu Zhang, and Zhen Chen. A simulation study on nanoscale holes generated by gold nanoparticles on negative lipid bilayers. Langmuir, 27(13):8323-8332, 2011.
[262] Rakesh Gupta and Beena Rai. Effect of size and surface charge of gold nanoparticles on their skin permeability: a molecular dynamics study. Scientific reports, 7(1):1-13, 2017.
[263] Pavel V Komarov, Lubov V Zherenkova, and Pavel G Khalatur. Computer simulation of the assembly of gold nanoparticles on dna fragments via electrostatic interaction. The Journal of chemical physics, 128(12):124909, 2008.
[264] Massimiliano Chiappini, Erika Eiser, and Francesco Sciortino. Phase behaviour in complementary DNA-coated gold nanoparticles and fd-viruses mixtures: a numerical study. Eur. Phys. J. E, 40(1):7, January 2017.
[265] Walter W Weare, Scott M Reed, Marvin G Warner, and James E Hutchison. Improved synthesis of small (d core $\approx 1.5 \mathrm{~nm}$ ) phosphine-stabilized gold nanoparticles. Journal of the American Chemical Society, 122(51):12890-12891, 2000.
[266] Markus Deserno, Axel Arnold, and Christian Holm. Attraction and ionic correlations between charged stiff polyelectrolytes. Macromolecules, 36(1):249-259, 2003.
[267] Joan Conrad, Mark Troll, and Bruno H Zimm. Ions around dna: Monte carlo estimates of distribution with improved electrostatic potentials. Biopolymers: Original Research on Biomolecules, 27(11):1711-1732, 1988.
[268] Juan Pelta, Françoise Livolant, and Jean-Louis Sikorav. Dna aggregation induced by polyamines and cobalthexamine. Journal of Biological Chemistry, 271(10):5656-5662, 1996.
[269] Anieke M Wierenga and Albert P Philipse. Low-shear viscosities of dilute dispersions of colloidal rodlike silica particles in cyclohexane. Journal of colloid and interface science, 180(2):360-370, 1996.
[270] Grace Martinelli Kepler and Seth Fraden. Attractive potential between confined colloids at low ionic strength. Physical review letters, 73(2):356, 1994.
[271] Niels Grønbech-Jensen, Robert J Mashl, Robijn F Bruinsma, and William M Gelbart. Counterion-induced attraction between rigid polyelectrolytes. Physical Review Letters, 78(12):2477, 1997.
[272] Kun-Chun Lee, Itamar Borukhov, William M Gelbart, Andrea J Liu, and Mark J Stevens. Effect of mono-and multivalent salts on angle-dependent attractions between charged rods. Physical review letters, 93(12):128101, 2004.
[273] Clara D. Christ, Alan E. Mark, and Wilfred F. van Gunsteren. Basic ingredients of free energy calculations: A review. Journal of Computational Chemistry, 31(8):1569-1582.
[274] Markus Deserno. Counterion condensation for rigid linear polyelectrolytes. PhD thesis, Universitätsbibliothek Mainz, 2000.
[275] Liang Dai, Yuguang Mu, Lars Nordenskiöld, and Johan RC van der Maarel. Molecular dynamics simulation of multivalent-ion mediated attraction between dna molecules. Physical review letters, 100(11):118301, 2008.
[276] Tiedong Sun, Alexander Mirzoev, Vishal Minhas, Nikolay Korolev, Alexander P Lyubartsev, and Lars Nordenskiöld. A multiscale analysis of dna phase separation: from atomistic to mesoscale level. Nucleic acids research, 47(11):5550-5562, 2019.
[277] Tiedong Sun, Alexander Mirzoev, Vishal Minhas, Nikolay Korolev, Alexander P Lyubartsev, and Lars Nordenskiöld. Hierarchical multiscale simulation of dna condensation. bioRxiv, page 375626, 2018.
[278] Lars Guldbrand, Lars G Nilsson, and Lars Nordenskiöld. A monte carlo simulation study of electrostatic forces between hexagonally packed dna double helices. The Journal of chemical physics, 85(11):6686-6698, 1986.
[279] Lori K Sanders, Wujing Xian, Camilo Guáqueta, Michael J Strohman, Chuck R Vrasich, Erik Luijten, and Gerard CL Wong. Control of electrostatic interactions between f-actin and genetically modified lysozyme in aqueous media. Proceedings of the National Academy of Sciences, 104(41):15994-15999, 2007.
[280] Camilo Guáqueta, Lori K Sanders, Gerard CL Wong, and Erik Luijten. The effect of salt on self-assembled actin-lysozyme complexes. Biophysical journal, 90(12):4630-4638, 2006.
[281] Alexander P Lyubartsev, Jay X Tang, Paul A Janmey, and Lars Nordenskiöld. Electrostatically induced polyelectrolyte association of rodlike virus particles. Physical review letters, 81(24):5465, 1998.
[282] Yves Rocard. Thermodynamique. 1967.
[283] Lars Guldbrand, Lars G. Nilsson, and Lars NordenskiÃ $\| l d$. A Monte Carlo simulation study of electrostatic forces between hexagonally packed DNA double helices. J. Chem. Phys., 85(11):6686-6698, December 1986.
[284] Sergey Leikin, V Adrian Parsegian, Donald C Rau, and R Peter Rand. Hydration forces. Annual Review of Physical Chemistry, 44(1):369-395, 1993.
[285] DC Rau, B Lee, and VA Parsegian. Measurement of the repulsive force between polyelectrolyte molecules in ionic solution: hydration forces between parallel dna double helices. Proceedings of the National Academy of Sciences, 81(9):2621-2625, 1984.
[286] Yizhak Marcus. Ionic radii in aqueous solutions. Chemical Reviews, 88(8):1475-1498, 1988.
[287] Juan Carlos Gil Montoro and José LF Abascal. Ionic distribution around simple b-dna models ii. deviations from cylindrical symmetry. The Journal of chemical physics, 109(14):6200-6210, 1998.
[288] Hans Greberg and Roland Kjellander. Charge inversion in electric double layers and effects of different sizes for counterions and coions. The Journal of chemical physics, 108(7):2940-2953, 1998.
[289] Jin-Si Zhang, Xi Zhang, Zhong-Liang Zhang, and Zhi-Jie Tan. Potential of mean force between oppositely charged nanoparticles: A comprehensive comparison between poissonboltzmann theory and monte carlo simulations. Scientific reports, 7(1):1-11, 2017.
[290] C Tongu, T Kenmotsu, Y Yoshikawa, AA Zinchenko, N Chen, and K Yoshikawa. Competitive effects of $2+$ and $3+$ cations on dna compaction. arXiv preprint arXiv:1601.01392, 2016.
[291] Yu Bai, Max Greenfeld, Kevin J Travers, Vincent B Chu, Jan Lipfert, Sebastian Doniach, and Daniel Herschlag. Quantitative and comprehensive decomposition of the ion atmosphere around nucleic acids. Journal of the American Chemical Society, 129(48):14981-14988, 2007.
[292] Jejoong Yoo and Aleksei Aksimentiev. Competitive binding of cations to duplex dna revealed through molecular dynamics simulations. The Journal of Physical Chemistry B, 116(43):12946-12954, 2012.
[293] Lori K. Sanders, Camilo GuÃiqueta, Thomas E. Angelini, Jae-Wook Lee, Scott C. Slimmer, Erik Luijten, and Gerard C. L. Wong. Structure and Stability of Self-Assembled ActinLysozyme Complexes in Salty Water. Phys. Rev. Lett., 95(10):108302, September 2005.
[294] Ville Liljeström, Ari Ora, Jukka Hassinen, Heikki T Rekola, Maria Heilala, Ville Hynninen, Jussi J Joensuu, Robin HA Ras, Päivi Törmä, Olli Ikkala, et al. Cooperative colloidal self-assembly of metal-protein superlattice wires. Nature communications, 8(1):1-10, 2017.
[295] Thomas E Gartner III and Arthi Jayaraman. Modeling and simulations of polymers: A roadmap. Macromolecules, 52(3):755-786, 2019.
[296] PK Kahol, JC Ho, YY Chen, CR Wang, S Neeleshwar, CB Tsai, and B Wessling. On metallic characteristics in some conducting polymers. Synthetic metals, 151(1):65-72, 2005.
[297] David C Martin, Jinghang Wu, Charles M Shaw, Zachary King, Sarah A Spanninga, Sarah Richardson-Burns, Jeffrey Hendricks, and Junyan Yang. The morphology of poly (3, 4ethylenedioxythiophene). Polymer Reviews, 50(3):340-384, 2010.
[298] Xavier Crispin, FLE Jakobsson, Annica Crispin, PCM Grim, Peter Andersson, A Volodin, Christian Van Haesendonck, Mark Van der Auweraer, William R Salaneck, and Magnus Berggren. The origin of the high conductivity of poly (3, 4-ethylenedioxythiophene)- poly (styrenesulfonate)(pedot- pss) plastic electrodes. Chemistry of Materials, 18(18):4354-4360, 2006.
[299] A Mantovani Nardes, Martijn Kemerink, René AJ Janssen, Jolanda AM Bastiaansen, Nicole MM Kiggen, Bea MW Langeveld, Albert JJM Van Breemen, and Margreet M De Kok. Microscopic understanding of the anisotropic conductivity of pedot: Pss thin films. Advanced Materials, 19(9):1196-1200, 2007.
[300] Takumi Takano, Hiroyasu Masunaga, Akihiko Fujiwara, Hidenori Okuzaki, and Takahiko Sasaki. Pedot nanocrystal in highly conductive pedot: Pss polymer films. Macromolecules, 45(9):3859-3865, 2012.
[301] Juan Felipe Franco-Gonzalez and Igor V. Zozoulenko. Molecular Dynamics Study of Morphology of Doped PEDOT: From Solution to Dry Phase. J. Phys. Chem. B, 121(16):4299-4307, April 2017.
[302] HJC Berendsen, JR Grigera, and TP Straatsma. The missing term in effective pair potentials. Journal of Physical Chemistry, 91(24):6269-6271, 1987.
[303] Koji Harano, Satoshi Okada, Shunsuke Furukawa, Hideyuki Tanaka, and Eiichi Nakamura. Formation of a polycrystalline film of donor material on pedot: Pss buffer induced by crystal nucleation. Journal of Polymer Science Part B: Polymer Physics, 52(12):833-841, 2014.
[304] Manrico V Fabretto, Drew R Evans, Michael Mueller, Kamil Zuber, Pejman Hojati-Talemi, Rob D Short, Gordon G Wallace, and Peter J Murphy. Polymeric material with metallike conductivity for next generation organic electronic devices. Chemistry of Materials, 24(20):3998-4003, 2012.
[305] Jochen S Hub, Bert L De Groot, and David Van Der Spoel. g_wham a free weighted histogram analysis implementation including robust error and autocorrelation estimates. Journal of chemical theory and computation, 6(12):3713-3720, 2010.
[306] Christopher J Fennell, Alan Bizjak, Vojko Vlachy, and Ken A Dill. Ion pairing in molecular simulations of aqueous alkali halide solutions. The Journal of Physical Chemistry B, 113(19):6782-6791, 2009.
[307] Patrick Yee, Jindal K. Shah, and Edward J. Maginn. State of Hydrophobic and Hydrophilic Ionic Liquids in Aqueous Solutions: Are the Ions Fully Dissociated? J. Phys. Chem. B, 117(41):12556-12566, October 2013.
[308] Keith E Gutowski and Edward J Maginn. Amine-functionalized task-specific ionic liquids: a mechanistic explanation for the dramatic increase in viscosity upon complexation with co2 from molecular simulation. Journal of the American Chemical Society, 130(44):14690-14704, 2008.
[309] Bernacki, Katarzyna and Hetényi, Balázs and Berne, BJ Multiple "time step" Monte Carlo simulations: Application to charged systems with Ewald summation. The Journal of chemical physics, 121(1):44-50, 2004.
[310] Robert H Swendsen and Jian-Sheng Wang. Replica monte carlo simulation of spin-glasses. Physical review letters, 57(21):2607, 1986.
[311] Mohsen Modarresi, Juan Felipe Franco-Gonzalez, and Igor Zozoulenko. Morphology and ion diffusion in pedot: Tos. a coarse grained molecular dynamics simulation. Physical Chemistry Chemical Physics, 20(25):17188-17198, 2018.
[312] Michael A Leaf. Conducting polyelectrolyte complexes: Assembly, structure, and transport. 2017.
[313] Jing-Zi Zhang, Xiang-Yao Peng, Shan Liu, Bang-Ping Jiang, Shi-Chen Ji, and Xing-Can Shen. The persistence length of semiflexible polymers in lattice monte carlo simulations. Polymers, 11(2):295, 2019.
[314] Ronald M Pratt. Madelung constants for ionic crystals using the ewald sum. Jurnal Kejuruteraan, 13:21, 2001.
[315] Daniel Siderius. Spc/e water reference calculations-9å cutoff. 2015.
[316] Otto Kratky and Günther Porod. Röntgenuntersuchung gelöster fadenmoleküle. Recueil des Travaux Chimiques des Pays-Bas, 68(12):1106-1122, 1949.
[317] Erwin Frey. Semiflexible polymer networks and persistence length.

## Ambroise de Izarra ( 암브로워즈 드 이잘라 )

## 고분자전해질 자기조립 전산모사 : DNA 상 금나노입자 조립 및 이온성액체에 의한 PEDOT:PSS 자기조립 조절

고분자전해질은 용액상에서 전하를 띄는 고분자로서, 생물학적, 산업적으로 중요하다 . 자기조립에 대한 이해가 향상되었음에도 불구하고, 자기조립 구조 및 특성의 예측은 물론 정확한 관찰조차 여전히 힘들다. 자기조립 특성이 주로 정전기적 상호작용에 의해 제어되기는 하나 여전히 수많은 원자 간 다양한 상호작용의 미세한 절충의 결과이고, 실험 조건, 전해질 특성, 온도, 용매 같은 많은 실험 변수에 민감하게 변하기 때문이다 . 본 논문의 목적은, 슈퍼컴퓨터를 활용한 분자 수준 전산모사, 즉 분자모델링을 통해 자기조립 현상을 분자 수준에서 관찰하고, 원천적으로 이해하고 예측하는 것이다. 산업적 활용이 가능한 두 자기조립 현상에 집중하여, DNA 상에 금 나노입자를 자기조립시켜 기능화하는 과정의 몬테카를로 모사와 PEDOT:PSS 수용성 전도성 고분자의 전기전도도가 이온성액체로 개선되는 기작의 분자동력학 모사를 수행한다 .

핵심어: 전해질, 전산모사, 몬테카를로 모사, 분자동력학


[^0]:    ${ }^{1}$ Declaration of Ethical Conduct in Research: I, as a graduate student of DGIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

[^1]:    Keywords: Polyelectrolytes, Monte-Carlo simulation, Molecular dynamics

[^2]:    ${ }^{a} R_{i}$ is the initial CM-CM distance between cation and anion for complexes obtained from previous DFT calculations (ref. [201]).
    ${ }^{b} R_{e q}$ is the CM-CM distance corresponding to the minimum in PMF curves.
    ${ }^{c} R_{C}$ is the CM-CM distance corresponding to the position after the endpoint.
    $\mathrm{K}_{A}$ is in $\mathrm{dm}^{3} \cdot \mathrm{~mol}^{-1}$.

