



Università degli Studi di Cagliari

---

Facoltà di Scienze

*Dipartimento di Scienze Chimiche e Geologiche*

Dottorato in Scienze e Tecnologie Chimiche

**Modelling in chimica computazionale:  
ottimizzazione di codici MD  
ed applicazioni allo studio di sistemi  
ad elevata carica**

Supervisore:

Dott.ssa Francesca Mocci

Coordinatore:

Prof. Mariano Casu

Candidato:

Massimiliano Porcu

XXVI Ciclo

*“Le ricerche di molti analisti  
hanno già gettato parecchio buio sull’argomento  
ed è probabile che, se continueranno,  
non sapremo più nulla di nulla al riguardo.”  
(Mark Twain)*

# Ringraziamenti

Desidero ringraziare tutti coloro che mi hanno aiutato nello svolgimento del lavoro di ricerca ed in particolare gli amici che hanno reso possibile il mio periodo di soggiorno a Stoccolma e la collaborazione con il gruppo di chimica computazionale della Stockholm University:

La Dott.ssa Francesca Mocci, sempre gentile e disponibile, per l'aiuto prestato durante lo svolgimento della ricerca e le numerose ore dedicate alla mia tesi e per il sostegno logistico a Stoccolma senza il quale, più di una volta, mi sarei trovato in grande difficoltà.

Il Prof. Aatto Laaksonen, per la gentilezza, presenza e disponibilità e per avermi offerto la possibilità di collaborare con il suo gruppo di ricerca nella bella Stoccolma.

Il Prof. Pino Saba, per il pronto sostegno materiale ed informatico nei momenti di necessità.

Il Prof. Alexander Lyubartsev, per la disponibilità e gli utili consigli.

# Indice

<b>Introduzione</b>	<b>1</b>
<b>1 MD e codici di simulazione</b>	<b>6</b>
1.1 Dinamica	7
1.1.1 Integrazione delle equazioni del moto	8
1.1.2 Condizioni periodiche al contorno	8
1.1.3 Modellazione delle interazioni	9
1.1.4 Interazioni tra atomi non legati	9
1.1.5 Interazioni elettrostatiche a lungo raggio	10
1.1.6 Controllo di temperatura e pressione	12
1.1.7 Inclusione del Solvente	12
1.1.8 Analisi delle traiettorie	13
1.2 Il codice M.DynaMix	13
1.2.1 Caratteristiche generali	14
1.2.2 L'algoritmo	15
1.2.3 Efficienza	18
1.2.4 Utenza	18
1.3 Il codice TRANAL	19
1.3.1 Strumenti di analisi	19
1.4 Il gradiente del campo elettrico	20
<b>2 Interazioni coulombiane a lungo raggio</b>	<b>21</b>
2.1 La somma di Ewald	21
2.1.1 Calcolo delle forze	27
2.1.2 Formule per l'energia e le forze nella somma di Ewald	28
2.2 Il metodo ENUF	31
2.2.1 Transformate di Fourier discrete per dati non equispaziati	31

2.2.2	Definizione del problema . . . . .	32
2.2.3	Fondamenti teorici . . . . .	32
2.2.4	Formulazione matrice-vettore . . . . .	32
2.2.5	Tipologie di prodotti matrice-vettore . . . . .	33
2.2.6	NDFT, FFT, NFFT . . . . .	33
2.2.7	Somma di Ewald veloce . . . . .	34
2.3	Il metodo SPME . . . . .	38
2.3.1	Interpolazione del fattore di struttura . . . . .	38
2.3.2	Sviluppo in B-splines . . . . .	39
2.3.3	Energia . . . . .	41
2.3.4	Forze . . . . .	42
2.3.5	Algoritmo . . . . .	42
<b>3</b>	<b>Evoluzioni di M.DynaMix e TRANAL</b>	<b>43</b>
3.1	Motivazioni . . . . .	44
3.2	Analisi preliminare . . . . .	44
3.3	Principali limitazioni di M.DynaMix e TRANAL . . . . .	45
3.3.1	Lo stato di TRANAL . . . . .	46
3.4	Migliorie . . . . .	46
3.4.1	Elettrostatica sulla griglia . . . . .	46
3.4.2	Allocazione dinamica della memoria . . . . .	46
3.4.3	Acceleratori computazionali . . . . .	47
3.5	Scansione temporale delle modifiche . . . . .	47
3.5.1	Fase 1: lungo raggio . . . . .	48
3.5.2	Fase 2: GPU e RAM dinamica . . . . .	48
3.5.3	Fase 3: EFG . . . . .	48
3.6	Implementazione del metodo SPME . . . . .	49
3.6.1	Linguaggio . . . . .	49
3.6.2	Algoritmo . . . . .	50
3.6.3	Dettagli implementativi . . . . .	51
3.6.4	Versione parallela . . . . .	52
3.6.5	Parametri di input . . . . .	52
3.6.6	Testing . . . . .	53
3.6.7	Benchmarking . . . . .	53

3.7	Implementazione del metodo ENUF . . . . .	54
3.7.1	Linguaggio . . . . .	55
3.7.2	Algoritmo . . . . .	55
3.7.3	Dettagli implementativi . . . . .	56
3.7.4	Versione parallela . . . . .	56
3.7.5	Parametri di input . . . . .	57
3.7.6	Testing . . . . .	58
3.7.7	Benchmarking . . . . .	58
3.7.8	Pubblicazioni . . . . .	59
3.8	TRilly . . . . .	59
3.8.1	Implementazione . . . . .	59
<b>4</b>	<b>Porting di M.DynaMix su GPU</b>	<b>61</b>
4.1	GPU e calcolo scientifico . . . . .	61
4.1.1	Stato dell'arte . . . . .	62
4.1.2	Strutturazione del mercato . . . . .	63
4.2	Architettura delle GPU . . . . .	63
4.2.1	Memoria . . . . .	64
4.3	Struttura di un programma CUDA . . . . .	65
4.3.1	GPU e MD . . . . .	66
4.4	Porting . . . . .	67
4.4.1	Approccio generale . . . . .	67
4.4.2	Common blocks e memoria dinamica . . . . .	68
4.4.3	CUDA kernels . . . . .	69
4.4.4	Linguaggio e interfacciamento . . . . .	70
4.4.5	Supporto multi-GPU . . . . .	70
4.4.6	Dettagli implementativi . . . . .	71
4.5	Parametri di input . . . . .	73
4.6	Testing . . . . .	74
4.7	Benchmarking . . . . .	74
<b>5</b>	<b>Interazioni tra DNA e controioni</b>	<b>76</b>
5.1	Il DNA . . . . .	76
5.1.1	Unità costitutive . . . . .	77

---

5.1.2	Struttura . . . . .	78
5.1.3	Dipendenza dalla sequenza . . . . .	81
5.2	MD e rilassamento quadrupolare . . . . .	81
5.2.1	Teoria . . . . .	82
5.3	La ricerca . . . . .	84
5.3.1	Motivazioni . . . . .	85
5.3.2	Sequenze nucleotidiche analizzate . . . . .	85
5.3.3	Set di parametri . . . . .	86
5.3.4	Protocollo di simulazione . . . . .	88
5.3.5	Interpretazione di RDF e RT . . . . .	89
5.3.6	EFG e funzione di autocorrelazione . . . . .	92
5.4	Conclusioni . . . . .	93
<b>6</b>	<b>Applicazioni ai liquidi ionici</b>	<b>95</b>
6.1	Studio e motivazioni . . . . .	95
6.2	Il DMIIm . . . . .	97
6.3	Force Fields . . . . .	98
6.4	Simulazione . . . . .	98
6.4.1	Comparazione dei set . . . . .	99
6.5	Studio del rilassamento quadrupolare . . . . .	102
6.5.1	Confronto con l'esperimento . . . . .	102
6.5.2	Moto dei singoli anioni . . . . .	104
6.6	Conclusioni . . . . .	108
	<b>Conclusioni</b>	<b>109</b>
	<b>Appendice 1: pubblicazioni</b>	<b>111</b>
	<b>Bibliografia</b>	<b>134</b>

# Introduzione

La Dinamica Molecolare Classica (MD) è una tecnica computazionale per lo studio dei sistemi fisici al livello atomico; strumento di investigazione ampiamente utilizzato, essa si basa essenzialmente sull'applicazione della meccanica Newtoniana ai sistemi microscopici. In questa metodica, le interazioni elementari sono parametrizzate classicamente, con lo scopo di semplificare l'approccio e rendere computazionalmente trattabili, sistemi con un elevato numero di particelle.

I codici di calcolo sono un fondamentale strumento di lavoro per coloro che si occupano di modelling computazionale; il risultato di uno studio o la tempistica con cui esso viene prodotto, possono essere influenzati notevolmente dalle caratteristiche del pacchetto di calcolo utilizzato; in questo senso, spesso il ricercatore è costretto a trovare un adeguato punto di accordo, tra esigenze in larga misura contrastanti.

Molte sono le tipologie di sistemi a cui la MD è usualmente applicata come strumento di indagine. Tra questi, in tempi recenti hanno riscosso interesse crescente gli *acidi nucleici* ed i *liquidi ionici*. In entrambe i sistemi, il controione della parte (bio-)organica è uno ione quadrupolare.

Il DNA è la molecola biologica per eccellenza, simbolo stesso della vita; la sua famosa *doppia elica* codifica l'informazione ereditaria, permettendo agli organismi di trasmettere le proprie caratteristiche alla discendenza. Dal punto di vista chimico, il DNA si presenta come un polianione altamente carico, le cui proprietà strutturali e funzionali sono fortemente condizionate dalle interazioni elettrostatiche.

In condizioni fisiologiche, la molecola si trova in soluzione acquosa, nella quale sono presenti ioni carichi positivamente; la presenza dell'acqua e dei controioni, riduce le forti repulsioni intramolecolari, consentendo al composto di acquistare stabilità e svolgere le proprie funzioni biologiche. Fare luce sulle tipologie di interazione tra DNA e controioni,



può aiutare a comprendere i meccanismi alla base di molti fenomeni e riveste per questo un elevato interesse teorico e applicativo.

Altri composti che, nell'ultimo decennio, sono stati oggetto di intensi studi sperimentali e computazionali, sono i liquidi ionici. Dal punto di vista applicativo, essi presentano grande interesse, avendo volatilità quasi nulla, elevata ampiezza in temperatura della fase liquida, grande stabilità termica, buona conduttività elettrica, caratteristiche che conferiscono loro proprietà peculiari, sia come solventi che come liquidi puri.

Le simulazioni basate su potenziali classici, hanno aiutato a comprendere più a fondo il legame tra la struttura chimica e le proprietà di questi composti. La verifica del grado di affidabilità dei potenziali intermolecolari per l'esecuzione di simulazioni MD su questi sistemi, è attualmente un obiettivo generale di primaria importanza.

Nello studio del DNA, come dei liquidi ionici, la MD permette di ottenere descrizioni al livello microscopico, che possono fornire un'utile chiave di lettura per la spiegazione di molte osservazioni sperimentali. Per contro, la MD è una tecnica nella quale le interazioni vengono parametrizzate e la natura stessa del modello gioca un ruolo fondamentale; inoltre, il confronto sistematico con l'esperimento è in molti casi difficoltoso. In questo scenario, è naturale chiedersi in che misura le conclusioni ottenute dalle simulazioni siano affidabili o costituiscano un artefatto prodotto dalla particolare forma usata per la descrizione del sistema.

Nel contesto descritto si inquadra il lavoro di ricerca presentato in questa tesi, che verte sulla messa a punto di opportuni strumenti informatici per l'esecuzione di simulazioni MD e per l'analisi dei dati prodotti e sulla successiva applicazione di tali strumenti allo studio di sistemi altamente carichi, quali acidi nucleici e liquidi ionici.

Per le sue possibilità computazionali, il programma di Dinamica Molecolare *M.DynaMix* è diffuso in un'ampia comunità di scienziati ed applicato allo studio di molte tipologie

di sistemi. In fase di pianificazione della ricerca di dottorato, è emerso che *M.DynaMix* sarebbe stato lo strumento per molti versi adatto all'esecuzione degli studi programmati. Nel pacchetto software erano però presenti alcune limitazioni, che lo rendevano non perfettamente idoneo.

Per ovviare agli inconvenienti, si è provveduto a realizzare le necessarie modifiche, arrivando a disporre di un programma adatto agli scopi prefissi. In particolare, tra i miglioramenti realizzati, spicca l'introduzione di due nuovi algoritmi per la determinazione delle interazioni elettrostatiche a lungo raggio, che hanno consentito, a parità di precisione, una riduzione dei tempi di calcolo di un ordine di grandezza rispetto alla versione originale. Inoltre, il codice è stato adattato all'esecuzione su *Graphics Processor Units* (GPU), la categoria di acceleratori computazionali attualmente più noti e di maggiore diffusione, incrementando ulteriormente l'efficienza, di un fattore rilevante. In ultimo, si è realizzata la creazione di un programma *ex novo*, per l'analisi dei dati delle dinamiche, capace di calcolare il *gradiente del campo elettrostatico*; ciò ha consentito di applicare una tecnica basata sulla determinazione dei *parametri del decadimento degli ioni quadrupolari*, che rende possibile il confronto sistematico con i dati delle misure NMR e può essere utilizzata per la validazione dei risultati delle simulazioni.

Gli strumenti informatici creati, sono stati successivamente utilizzati per l'analisi di alcuni selezionati sistemi chimici ad elevata carica, costituiti da *oligomeri di DNA* e *liquidi ionici*. Caratteristica comune ai sistemi investigati, era la presenza di ioni dotati di momento elettrico quadrupolare,  $^{23}\text{Na}$  per gli oligomeri e  $^{35}\text{Cl}$  per il liquido ionico.

Nello studio sugli oligomeri, sono state analizzate le interazioni tra DNA e controioni sodio, alla ricerca di proprietà esplicitamente dipendenti dalla sequenza di basi nucleotidiche. È noto infatti che, in alcune condizioni, i controioni sodio interagiscono preferenzialmente con specifiche sequenze denominate *A-tract*, coordinandosi con esse per periodi più lunghi. Purtroppo, per motivi che verranno discussi nel capitolo 5, dal punto di vista sperimentale non è possibile fornire una descrizione dettagliata al livello atomico di tali interazioni e le simulazioni rivestono un ruolo di grande importanza per il loro studio a tale livello.

Si ritiene che il rapporto tra *A-tract* e controioni possa avere un ruolo importante nella determinazione delle proprietà di tali sequenze, la cui presenza ha un ruolo chiave nel modulare i processi di deformazione complessiva della catena nucleotidica.

Nel contesto dei liquidi ionici, è stato realizzato uno studio del composto *1-decil-3-metil-imidazolio cloruro*, con l'intento iniziale di determinare un modello adeguato per la sua descrizione. L'applicabilità degli studi MD, dipende infatti dall'affidabilità del modello adottato per la parametrizzazione delle interazioni elementari; la necessità di rappresentare, contemporaneamente e in maniera adeguata, interazioni di Van der Waals, di debole entità, e interazioni Coulombiane, notevolmente più intense, rende la determinazione di un set di parametri valido per le simulazioni, un obiettivo non triviale.

Per tutti i sistemi studiati, è stato affrontato il problema della validazione dei risultati delle dinamiche.

Il calcolo dei parametri del decadimento quadrupolare dei controioni, ha consentito un confronto diretto con i dati sperimentali, permettendo di valutare la bontà relativa dei diversi modelli utilizzati per la descrizione delle interazioni. In presenza di buon accordo con l'esperimento, è apparso ragionevole riporre un elevato grado di fiducia nella descrizione della dinamica, al livello atomico, ottenuta dalla simulazione.

La parte rimanente di questo documento si articola come segue:

Nel **Capitolo 1**, è illustrata la tecnica computazionale della Dinamica Molecolare Classica, con particolare riferimento agli algoritmi più diffusamente applicati e alle caratteristiche generali dei codici di calcolo impiegati per le simulazioni e per l'analisi dei dati. Successivamente, è proposta una descrizione dettagliata del programma *M.DynaMix*, le cui migliorie ed evoluzioni sono state oggetto della presente ricerca.

Nel **Capitolo 2** è trattato, dal punto di vista teorico, il problema delle interazioni Coulombiane a lungo raggio nei sistemi periodici; sono quindi descritte in dettaglio le caratteristiche dei due nuovi algoritmi implementati in *M.DynaMix*, per il calcolo dell'energia elettrostatica dei sistemi di particelle.

Il **Capitolo 3** illustra la prima parte del lavoro di messa a punto di *M.DynaMix*,

che ha avuto come obiettivo principale l'implementazione degli efficienti algoritmi per il calcolo delle interazioni elettrostatiche a lungo raggio, descritti nel capitolo precedente.

La restante parte del lavoro di messa a punto dei codici, è descritta nel **Capitolo 4**; essa riguarda, in particolare, l'adattamento di *M.DynaMix* all'esecuzione su GPU.

Lo studio sulle interazioni tra oligomeri di DNA e controioni, è l'argomento illustrato nel **Capitolo 5**. Precedentemente alla discussione del problema della validazione delle simulazioni, è riportata la trattazione teorica dei meccanismi alla base del rilassamento quadrupolare.

Il **Capitolo 6** è relativo alle simulazioni del liquido ionico *1-decil-3-metil-imidazolio cloruro*.

In **Appendice**, infine, sono incluse le pubblicazioni scientifiche relative alle ricerche illustrate nel presente documento.

# 1 MD e codici di simulazione

La Dinamica Molecolare Classica (MD) è una tecnica computazionale per lo studio della dinamica di un sistema fisico al livello atomico o di gruppi di atomi. Potente strumento di investigazione, è largamente utilizzata per la comprensione dei comportamenti chimico fisici della materia al livello microscopico [1, 2, 3].

La tecnica si basa essenzialmente sull'applicazione della meccanica Newtoniana ai sistemi microscopici. Le interazioni elementari sono idealizzate, con lo scopo di semplificare l'approccio e rendere computazionalmente trattabili, complessi con un elevato numero di particelle.

Al contrario di ciò che avviene con i metodi quanto-meccanici, gli elettroni non sono trattati distintamente dai loro nuclei, ma ad essi accorpati nella medesima struttura. Il diametro delle strutture atomiche, così come l'entità delle cariche parziali, è ottenuto da misure sperimentali o ricavato da simulazioni da principi primi.

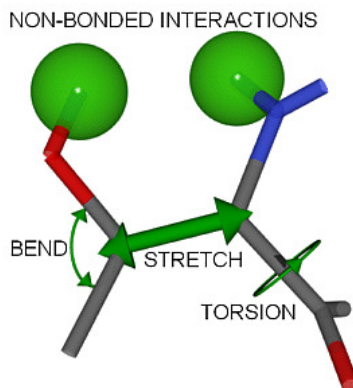


Figura 1.1: Modello dell'iterazione tra atomi legati in MD. I legami hanno la possibilità di allungarsi e accorciarsi (*stretching*), flettersi (*bending*) o torcersi (*torsion*).

Le interazioni tra particelle sono parametrizzate classicamente. La capacità dei legami atomici di accorciarsi e allungarsi (*stretching*) e flettersi (*bending*) è normalmente

rappresentata con funzioni armoniche; quella di ruotare (*torsion*), da sviluppi in serie di coseni.

I contributi dovuti ad atomi non legati, comprendono usualmente un potenziale di tipo Lennard-Jones per le interazioni di Van der Waals [6], a cui si aggiunge un contributo elettrostatico. Quest'ultimo è descritto attraverso un potenziale Coulombiano tra cariche puntiformi centrate sugli atomi; viene calcolato con tecniche basate su somme di Ewald o sue approssimazioni su griglie di punti [30, 35].

Per Force Field (FF) si intende la forma funzionale usata per la modellizzazione delle interazioni, unitamente all'insieme dei parametri adottati nel modello. Il FF è utilizzato per la determinazione dell'energia di interazione tra le particelle del sistema [4]; esso dipende strettamente dal campo di indagine considerato.

Nonostante le approssimazioni su cui la MD è basata, il confronto con l'esperimento indica come l'approccio posseda la propria validità; nel corso dell'ultimo trentennio, le tecniche di simulazioni hanno conseguito notevoli successi, sia come metodo di indagine di sistemi complessi, sia nella verifica dei limiti di validità di modelli teorici [4].

## 1.1 Dinamica

Il secondo principio della dinamica è alla base delle equazioni del moto di un sistema classico. Nella MD, si suppone di collocare un insieme di punti materiali in una cella di simulazione, la cui forma è di norma cubica, ottaedrica o esagonale. I punti materiali, di massa  $m$ , coincidono con gli atomi, di cui si trascurano i gradi di libertà elettronici. La legge  $\mathbf{F} = m\mathbf{a}$  fornisce un insieme di equazioni differenziali per le quali le posizioni  $\mathbf{r}$  e velocità  $\mathbf{v}$  iniziali sono note. L'integrazione delle equazioni è effettuata attraverso un opportuno algoritmo numerico.

Le equazione del moto del sistema sono quindi della forma

$$m_i \mathbf{a}_i = \mathbf{F}_i \quad (1.1)$$

dove  $\mathbf{F}_i$  è la forza che agisce sull' $i$ -mo atomo e  $\mathbf{a}_i$  la sua accelerazione. La forza è ottenuta come gradiente dell'energia potenziale di interazione  $V$ , ed è esprimibile come somma di contributi tra coppie di atomi

$$\mathbf{F}_i = -\nabla_{\mathbf{r}_i} V = -\sum_j \nabla_{\mathbf{r}_{ij}} V(\mathbf{r}_{ij}) = \sum_j \mathbf{F}_{ij}. \quad (1.2)$$

Nel caso dei potenziali di interazione comunemente utilizzati, continui con derivata continua, è possibile risolvere numericamente il sistema di equazioni differenziali, calcolando la soluzione a tempi discreti, separati da un passo di integrazione  $\Delta t$  (*time step*).

### 1.1.1 Integrazione delle equazioni del moto

La soluzione numerica delle equazioni del moto, si realizza trasformando la descrizione differenziale in relazioni alle differenze finite, che consentano di calcolare l'evoluzione di posizioni e velocità ad intervalli di tempo discreti. Se si riesce a mantenere un adeguato controllo sull'accuratezza, uno schema numerico che consente di passare dalle condizioni iniziali al tempo  $t_0$  alle posizioni e velocità al tempo  $t_0 + \Delta t$ , potrà essere reiterato un numero arbitrario di volte.

Molti algoritmi sono stati proposti per l'integrazione delle equazioni del moto; uno dei più diffusi, insieme alle sue varianti, è quello di *Verlet*, per il quale l'errore locale di troncamento sulle posizioni è dell'ordine di  $\Delta t^4$  [5].

Nella accuratezza complessiva della simulazione, la scelta del passo di integrazione ha un'importanza fondamentale; con piccoli valori di  $\Delta t$  si limita l'errore ma si rende più oneroso il calcolo; è quindi indispensabile trovare un compromesso tra precisione e costo. Il valore di  $\Delta t$  dipenderà concretamente dall'entità della forza di interazione e dalla massa delle particelle in gioco e dovrà essere scelto in modo da mantenere costante l'energia totale entro la precisione della macchina.

Nelle usuali simulazioni MD,  $\Delta t$  ha un valore comunemente compreso tra 0.1 *fs* e 10 *fs*.

### 1.1.2 Condizioni periodiche al contorno

Il tempo macchina necessario per una simulazione MD, in sistemi di grandi dimensioni, ha un andamento approssimativamente quadratico rispetto al numero  $N$  di particelle presenti. Le potenze di calcolo attualmente disponibili, limitano  $N$  ad un valore decisamente insufficiente a studiare il comportamento di una fase massiva (*bulk*). Per ovviare a questo problema, si suole introdurre le cosiddette condizioni periodiche al contorno (*Periodic Boundary Conditions*, PBC), immaginando di replicare la cella di simulazione nelle tre direzioni dello spazio, a costituire, da una piccola collezione di molecole, un

sistema periodico di estensione macroscopica, nel quale la frazione di particelle prossime alla superficie è irrisoria.

Nell'applicare le PBC, si da luogo a repliche della cella di simulazione che sono sue copie esatte: per ogni atomo nella cella originale è presente un corrispondente, con la medesima velocità, in ognuna delle celle circostanti. Vengono risolte le equazioni del moto per le sole molecole presenti nella cella principale, ottenendo la posizione delle molecole nelle altre celle per traslazione.

Un atomo che abbandoni la cella principale è sostituito da un altro proveniente dalla faccia opposta e avente la stessa velocità; in questo modo, il numero degli atomi nella cella di simulazione viene mantenuto costante.

### 1.1.3 Modellazione delle interazioni

Nel campo della chimica biologica, i campi di forza più comunemente utilizzati rappresentano l'energia potenziale del sistema di particelle come somma di due contributi, relativi ad interazioni tra atomi legati (*bounded*) e non legati (*unbounded*):

$$V(\mathbf{r}) = V_{bond}(\mathbf{r}) + V_{unbond}(\mathbf{r}). \quad (1.3)$$

L'interazione tra atomi legati, comprende comunemente termini di stretching, bending e torsion [4] ed assume una forma del tipo

$$\begin{aligned} V_{bond}(\mathbf{r}) &= V^{stretch}(\mathbf{r}) + V^{bend}(\mathbf{r}) + V^{tors}(\mathbf{r}) = \\ &= \sum_{i=1}^{N_{stretch}} k_i^{str} (r_i - r_{0_i})^2 + \sum_{i=1}^{N_{bend}} k_i^{ben} (\theta_i - \theta_{0_i})^2 + \sum_{i=1}^{N_{tors}} k_i^{tor} \left[ 1 + \cos(n_i \phi_i - \gamma_i) \right] \end{aligned} \quad (1.4)$$

dove le costanti  $r_{0_i}$  e  $\theta_{0_i}$  rappresentano distanze e angoli naturali del legame,  $k_i^{str}$  e  $k_i^{ben}$  le costanti di forza nei due casi,  $\phi_i$  è l'angolo diedro per la torsione  $i$ -esima mentre  $k_i^{tor}$ ,  $n_i$  e  $\gamma_i$  sono parametri il cui significato fisico dipende dagli atomi coinvolti. Le somme sono estese rispettivamente a tutti gli stretching, rotazioni e torsioni presenti nelle molecole in esame.

### 1.1.4 Interazioni tra atomi non legati

La mutua interazione tra particelle non direttamente legate, è usualmente espressa come somma di termini a coppia, dipendenti cioè dalla sola distanza tra gli atomi in-



teragenti. Una coppia di atomi è considerata non legata se le particelle fanno parte di molecole diverse o, se parte della stessa molecola, sono separate da più di due legami covalenti.

I termini unbonded sono normalmente espressi dalla somma di due contributi: un potenziale di tipo 6 – 12, il potenziale di Lennard-Jones, per le interazioni di Van der Waals; un termine elettrostatico, modellato attraverso un potenziale Coulombiano tra cariche puntiformi centrate sugli atomi.

$$V_{unbond}(\mathbf{r}) = V_{LJ}(\mathbf{r}) + V_{el}(\mathbf{r}) = \sum_{unbond} 4\epsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \sum_{unbond} \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \quad (1.5)$$

Il potenziale di Lennard-Jones [6] è dato a sua volta dalla somma di due contributi, uno attrattivo ed uno repulsivo; ciò determina un minimo dell'energia potenziale. Il termine repulsivo domina a breve distanza e garantisce che gli atomi non si compenetrino. I parametri  $\epsilon$  e  $\sigma$  nella (1.5) sono legati alla posizione del minimo e alla sua profondità; essi sono ottenuti, secondo diversi schemi di ricombinazione, da parametri analoghi definiti su base atomica [7].

Normalmente, le interazioni tra atomi non legati sono prese in considerazione solo quando la distanza tra coppie di particelle è contenuta entro un dato valore, detto raggio di *cut-off* ( $r_{cut}$ ). Il cut-off è in genere scelto in modo che una particella della cella di simulazione principale non interagisca con la sua immagine nelle celle replicate; tale condizione viene detta *convenzione della minima immagine*.

### 1.1.5 Interazioni elettrostatiche a lungo raggio

Il contributo elettrostatico a lungo raggio, presente nella relazione (1.5) come ultimo termine, è rappresentato da una sommatoria infinita estesa a tutte le coppie di atomi non legati del sistema, individuati tenendo conto delle condizioni periodiche al contorno.

La valutazione della somma attraverso un troncamento, basato sull'introduzione di una distanza di *cut-off*, si rivela non soddisfacente, producendo deviazioni sensibili dai dati sperimentali [39]. Una maggiore accuratezza è ottenibile con il metodo delle *somme di Ewald* [30], nel quale un sistema periodico neutro di  $N$  cariche puntiformi  $q_i$  centrate nelle posizioni  $r_i$ , è rimpiazzato da uno ad esso equivalente, tale che il cal-

colo delle interazioni elettrostatiche, possa essere realizzato tramite serie assolutamente convergenti.

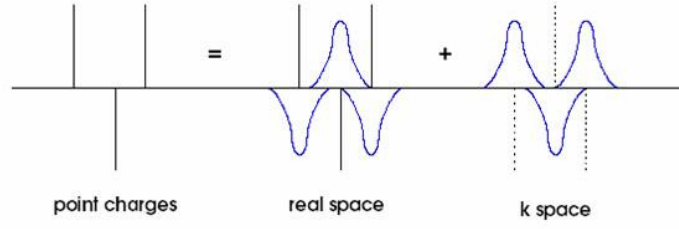


Figura 1.2: Modifica della distribuzione di cariche puntiformi secondo il meccanismo di Ewald. Un insieme di cariche puntiformi può essere interpretato come un insieme di cariche schermate, da cui viene sottratta la distribuzione di fondo che costituisce lo schermo.

L'idea alla base dell'approccio di Ewald è la sostituzione della distribuzione di cariche puntiformi  $q_i$  con una di più facile manipolazione. Il sistema originario, è modificato introducendo una serie di distribuzioni di carica diffuse, centrate attorno ad ogni atomo, tali da schermare le cariche collocate nelle posizioni  $r_i$ . Una seconda serie di distribuzioni diffuse, di segno opposto alle prime, è introdotta per riportare il sistema nella sua condizione di partenza. Il meccanismo è schematizzato in fig. 1.2.

Utilizzando la delta di Dirac per la descrizione della distribuzione di cariche puntiformi, si avrà

$$\begin{aligned} \rho_i(\mathbf{r}) &= q_i \delta(\mathbf{r} - r_i) = \rho_i^S(\mathbf{r}) + \rho_i^L(\mathbf{r}) = \\ &= \left[ q_i \delta(\mathbf{r} - r_i) - q_i G_\sigma(\mathbf{r} - r_i) \right] + q_i G_\sigma(\mathbf{r} - r_i) \end{aligned} \quad (1.6)$$

con le due distribuzioni, rispettivamente costituite dalle cariche originali addizionate con lo schermo e dal solo schermo di compensazione, date da

$$\begin{aligned} \rho_i^S(\mathbf{r}) &= q_i \delta(\mathbf{r} - r_i) - q_i G_\sigma(\mathbf{r} - r_i) \\ \rho_i^L(\mathbf{r}) &= q_i G_\sigma(\mathbf{r} - r_i). \end{aligned} \quad (1.7)$$

La distribuzione di schermo  $G_\sigma$  è usualmente di forma gaussiana

$$G_\sigma(\mathbf{r}) = \frac{1}{(2\pi\sigma^2)^{\frac{3}{2}}} \exp\left[-\frac{|\mathbf{r}|^2}{2\sigma^2}\right]. \quad (1.8)$$

Le distribuzioni  $\rho^S$  e  $\rho^L$  così ottenute, consentono una divisione della serie originaria in due termini, entrambi rapidamente convergenti. Il primo, dovuto a  $\rho^S$ , decade velocemente a grandi distanze ed è valutato in spazio reale. Il secondo, relativo a  $\rho^L$ , pur essendo di non rapida convergenza in spazio reale, è prodotto da una distribuzione di carica non più singolare ed è calcolabile in spazio reciproco con buona precisione, imponendo un opportuno cut-off.

Il metodo delle somme di Ewald appena descritto è computazionalmente gravoso, con un andamento del tempo di calcolo, quadratico rispetto al numero di particelle del sistema [31]. Per ovviare a questo problema, negli ultimi anni sono state proposte diverse varianti, basate su una mappatura del problema su griglie discrete di punti e l'utilizzo della trasformata di Fourier veloce (FFT). Tra questi, è da segnalare il metodo *Smooth Particle Mesh Ewald* (SPME) [35], presente nei più diffusi codici di simulazione MD [25, 26]. Di esso parleremo ampiamente nel corso del capitolo 2.

### 1.1.6 Controllo di temperatura e pressione

Le simulazioni MD possono essere effettuate in condizioni termodinamiche diverse. Ciò nasce anche dall'esigenza di confrontare i risultati di una simulazione con i dati sperimentali, ponendosi in condizioni il più possibile simili a quelle in cui questi dati sono stati ricavati.

Il caso più semplice è quello in cui il numero di particelle  $N$ , l'energia totale  $E$  e il volume  $V$  del sistema sono mantenuti costanti. Tale approccio si indica come *dinamica Hamiltoniana* e corrisponde al campionamento dell'insieme statistico *ensemble microcanonico* (NVE) [8].

In alternativa, mantenendo invariato il numero di particelle, è possibile eseguire la simulazione a volume e temperatura costanti (*ensemble canonico*, NVT) o a pressione e temperatura costanti (*ensemble isoterma-isobaro*, NPT). In quest'ottica sono stati sviluppati, nel corso degli anni, numerosi algoritmi che, alterando in modo opportuno le equazioni del moto, sono in grado di generare insiemi statistici isotermi (*termostati*) e isobari (*barostati*).

### 1.1.7 Inclusione del Solvente

L'acqua gioca un ruolo fondamentale nel determinare la struttura e le proprietà delle molecole biologiche. Le caratteristiche del modello utilizzato per descrivere il suo com-

portamento, hanno quindi un grande effetto sulla qualità ed attendibilità dei risultati prodotti dalle simulazioni MD.

Da un lato, vi sono modelli che tengono conto della natura discreta del solvente, quali ad esempio *TIP3P* e *SPC* [9, 10, 12, 14], trattando esplicitamente tutti gli atomi di cui esso è composto. Con questa descrizione dettagliata, è indispensabile utilizzare opportune condizioni periodiche al contorno, sia per prevenire la diffusione delle molecole d'acqua, che per consentire l'utilizzo di un numero limitato di molecole di solvente. L'approccio esplicito è in grado di riprodurre fedelmente vari comportamenti termodinamici e cinetici, ma la necessità di simulare la dinamica di un elevato numero di atomi, lo rende computazionalmente molto gravoso.

Alternativamente, sono in uso modelli *impliciti* [15], nei quali le molecole d'acqua non sono incluse nella simulazione ma rappresentate con un approccio di campo medio: attraverso una costante dielettrica efficace dipendente dalla distanza, si rende conto di uno dei più importanti effetti del solvente, quello di schermo per le interazioni elettrostatiche.

### 1.1.8 Analisi delle traiettorie

In una simulazione MD si produce una evoluzione deterministica che consiste tipicamente di una o più *traiettorie*, cioè l'insieme delle coordinate e velocità delle particelle del sistema in funzione del tempo. Tali traiettorie sono di norma analizzate *successivamente alla simulazione*, per estrarre le informazioni più importanti.

## 1.2 Il codice M.DynaMix

Il programma di Dinamica Molecolare M.DynaMix [65], è un codice largamente utilizzato, per la simulazione di insiemi di molecole in una cella periodica.

Per le sue possibilità computazionali, la struttura e l'organizzazione complessiva del codice, l'uso di M.DynaMix è diffuso in un'ampia comunità di scienziati, che lo applicano allo studio di molte tipologie di sistemi, quali acidi nucleici [21], bilayer lipidici [22], liquidi ionici [23] e altri [24]. L'importanza, da parte di molti soggetti, di disporre di uno strumento potente ed aggiornato, è stata una delle motivazioni che hanno condotto ad intraprendere un lavoro di revisione del codice. Il programma ha subito quindi miglioramenti su diversi fronti, attraverso l'introduzione di nuove funzionalità, l'aumento dell'efficienza di calcolo e l'ampliamento dello spettro di piattaforme hardware su

cui poterne realizzare l'esecuzione. Il nuovo strumento così prodotto è stato successivamente applicato allo studio di alcuni selezionati sistemi chimici ad elevata carica, che verrà descritto nei cap. 5 e 6.

Per comprendere appieno la linea su cui si è sviluppato il lavoro svolto, nei paragrafi seguenti sarà data una descrizione della situazione di partenza, illustrando le caratteristiche della versione originale del programma, precedentemente al processo di modifica. Sarà tracciato un quadro delle funzionalità generali del pacchetto, soffermandosi in particolare sui dettagli relativi all'implementazione del codice.

### 1.2.1 Caratteristiche generali

M.DynaMix è un codice polivalente, per la simulazione di insiemi di molecole interagenti con un Force Field di tipologia AMBER/CHARMM [25]; è stato proposto, nella sua prima versione parallela, alla fine degli anni 90, come evoluzione del programma MOLDYN [69].

La forma generale del FF implementato nel programma, rispecchia quanto descritto nelle sezioni (1.1.3) e (1.1.4) e contiene quindi termini di interazione bounded e unbounded. I primi, sono composti dai contributi quadratici di stretching e bending, e dal contributo di torsione espresso in serie di coseni. L'interazione tra atomi non legati comprende invece un termine di Lennard-Jones e un termine elettrostatico a lungo raggio, calcolato con l'algoritmo di somma di Ewald, delineato nella sezione (1.1.5). Il programma consente di effettuare simulazioni in diversi ensemble termodinamici, quali NVE, NVT, NPT e NPT anisotropico.

#### Linguaggio e versioni

Dal punto di vista dell'implementazione, il codice di M.DynaMix è scritto in linguaggio FORTRAN 77 (F77). Sono disponibili due versioni del programma, integrate nel medesimo pacchetto: una prima versione, seriale, è eseguibile su macchine sequenziali; una seconda è invece parallela e utilizza il protocollo di comunicazione MPI [70].

Strutturalmente, M.DynaMix ha la tipica organizzazione di un programma FORTRAN del suo periodo, con ampio uso di blocchi comuni di memoria (*common blocks*) e di variabili dichiarate implicitamente; la memoria è allocata staticamente.

Pur considerando le limitazioni intrinseche del linguaggio e lo stile di programmazione adottato, il codice possiede una sua modularità, ottenuta attraverso un'opportuna ri-

partizione delle procedure di calcolo (*subroutine*) in file, e la compilazione di file diversi a seconda delle esigenze. Questa sorta di embrionale *polimorfismo* [74] ad esempio, è il meccanismo utilizzato dagli autori del programma per differenziare la versione seriale da quella parallela: subroutine concettualmente corrispondenti, ma con un corpo di esecuzione diverso, sono create con il medesimo prototipo (stesso nome, stessa lista degli argomenti) e sono quindi, dal punto di vista formale, intercambiabili all'interno del programma. Tali routine sono poi collocate in file diversi, che vengono inclusi o meno, a seconda del caso, nella lista dei file di codice sorgente da compilare. La compilazione è realizzata attraverso opportuni *makefile*, distribuiti nel pacchetto fornito dagli autori, in cui la lista dei file di codice sorgente è definita senza ambiguità [69].

### Dati di input

Le informazioni sulla geometria e le caratteristiche chimiche del sistema da analizzare, sono comunicate al programma attraverso un insieme di file di input.

Ad ogni tipo molecolare usato nella simulazione è associato un file, in formato MOL [73, 69], contenente la posizione e la connettività degli atomi della molecola, nonché i parametri del relativo FF.

Le informazioni generali sulla simulazione, sono invece specificate in un ulteriore file di input [69]. Le più importanti tra esse comprendono: numero di molecole per ciascuna specie, forma e dimensioni della cella principale, durata del time step, lunghezza complessiva della simulazione, ensemble statistico di riferimento, valore del cut-off per le interazioni unbounded.

### 1.2.2 L'algoritmo

Schematicamente, l'algoritmo di M.DynaMix ha la forma che tipicamente ci si aspetta per un programma di Dinamica Molecolare Classica:

- Note le posizioni ad un istante  $t$ , tenendo conto di tutte le interazioni in gioco, viene calcolata la forza agente su ciascun atomo.
- Con uno schema numerico opportuno, a partire da posizioni e velocità al tempo  $t$ , si realizza l'integrazione delle equazioni del moto. In questa fase, è necessario rispettare gli eventuali vincoli presenti ad esempio invarianza di temperatura, pressione, lunghezze di legame o altro.

- Sono calcolate posizioni e le velocità di ciascun atomo al tempo  $t + \Delta t$ . Il ciclo può ricominciare.

Il calcolo delle forze su ciascun atomo prevede la valutazione della componente *bound-ed*, relativa agli atomi legati, e della componente *unbounded*, riguardante gli atomi non direttamente legati. La prima è valutata, tenendo conto di geometria e connettività delle molecole, tramite la parametrizzazione fornita dal FF. La seconda, nel suo contributo di Lennard-Jones, richiede che si individuino, per ciascuna particella, gli atomi all'interno del raggio di cut-off  $r_{cut}$ , distanza oltre la quale due particelle sono considerate interagire solo elettrostaticamente. Questa procedura viene svolta determinando, per ciascun atomo, una *lista di vicini* (o lista di *Verlet*), che elenchi tutti gli atomi all'interno del cut-off.

### Discretizzazione del tempo

In Dinamica Molecolare, la scelta del time step gioca un ruolo fondamentale. Piccoli valori di  $\Delta t$  limitano l'errore di discretizzazione ma aumentano il numero totale di passi necessari a coprire un dato intervallo di tempo e, di conseguenza, il costo computazionale complessivo. Valori di  $\Delta t$  troppo grandi rendono instabile la dinamica e inattendibili i risultati.

È noto empiricamente che, per avere una precisione accettabile,  $\Delta t$  debba essere scelto pari circa ad un decimo del tempo relativo al più rapido moto intrinseco del sistema simulato. Per le molecole biologiche, ciò corrisponde alla vibrazione dei legami dell'idrogeno, i cui tempi sono dell'ordine di alcuni fs; conseguentemente, il valore opportuno di  $\Delta t$  in questi casi è pari ad una frazione di fs.

Diverse tecniche sono state proposte per consentire l'utilizzo di time step più lunghi, evitando allo stesso tempo che la dinamica perda stabilità. Alcune di queste tecniche realizzano una integrazione vincolata delle equazioni del moto, mantenendo limitata l'oscillazione della lunghezza dei legami degli atomi più leggeri, tipicamente l'idrogeno; di questa famiglia fanno parte ad esempio gli algoritmi SHAKE e RATTLE [16, 17].

Una tecnica alternativa è quella del *doppio time step* [66].

### Doppio time step

Una caratteristica peculiare di M.DynaMix è l'utilizzo del doppio time step.

Con questa procedura, le interazioni unbounded a cui ciascun atomo è soggetto, sono valutate in momenti diversi a seconda della distanza della particella interagente.

Il metodo prevede l'introduzione di due time step, uno *short* ( $\Delta t_S$ ) e uno *long* ( $\Delta t_L$ ) e di due corrispondenti distanze di cut-off, *short range* ( $r_{cut}^S$ ) e *long range* ( $r_{cut}^L$ ). Il long time step è un multiplo intero dello short time step, si ha cioè  $\Delta t_L = n \cdot \Delta t_S$  con  $n \sim 10$ . Valori tipici per i cut-off sono invece  $r_{cut}^S \sim 5\text{\AA}$  e  $r_{cut}^L \sim 10\text{\AA}$ .

La dinamica delle particelle procede per avanzamenti discreti pari allo short time step. Ad ogni  $\Delta t_S$  si realizza un passo di dinamica, restringendo però le interazioni considerate, ai soli atomi collocati entro la distanza  $r_{cut}^S$ . Ogni long time step, cioè trascorsi  $n$  passi di dinamica, si procede al calcolo delle forze e all'aggiornamento delle posizioni tenendo in considerazione il cut-off più ampio,  $r_{cut}^L$ .

Questa tecnica consente di utilizzare time step brevi, ed avere quindi precisione elevata, senza perdite eccessive in termini di efficienza; il ridotto cut-off, infatti, limita il numero di particelle interagenti e di conseguenza il tempo di calcolo.

Per applicare la tecnica del doppio time step è necessaria la determinazione di due liste di Verlet per ciascun atomo, una relativa a  $r_{cut}^S$  e una a  $r_{cut}^L$ ; tali liste sono aggiornate tipicamente dopo un tempo dell'ordine di  $10 \cdot \Delta t^L$ .

## Parallelizzazione

La versione parallela di M.DynaMix fa uso del protocollo di comunicazione MPI [70]. Dal punto di vista delle informazioni, l'approccio adottato è quello della *replicazione dei dati*. Non vi è partizione geometrica del dominio: il parallelismo dei dati è parziale, inquanto riguarda le informazioni di output ma non quelle di input.

Come previsto dal modello *message passing*, il numero di processi creato è stabilito al caricamento del programma. I singoli eseguibili, relativi a ciascuna unità di calcolo, fanno riferimento a copie identiche delle informazioni, conservate nella memoria locale di ciascuna unità. A partire dagli stessi dati, sfruttando la conoscenza del proprio ID, ogni eseguibile è in grado di operare in modo diverso, calcolando solo la frazione di informazioni di sua competenza.

Al termine di ogni ciclo di dinamica, il processo con il ruolo di *master*, provvede a riallinere opportunamente le informazioni in possesso di ciascun nodo, in modo che ogni unità di calcolo disponga nuovamente di una copia identica dell'intero set di dati.



### 1.2.3 Efficienza

Come risultato da un approfondito profiling eseguito sul codice, il calcolo delle interazioni tra atomi non legati, è responsabile del contributo di gran lunga più rilevante al tempo totale di esecuzione del programma. Tale contributo può arrivare, a seconda del numero di atomi presenti nella cella, a superare il 95% della durata totale della simulazione.

Le procedure coinvolte in questa fase, che risultano quindi principalmente responsabili della resa di M.DynaMix in termini di efficienza, sono tre:

- Creazione delle liste di Verlet. L'assenza di una partizione geometrica del dominio obbliga a considerare tutte le coppie di atomi. Il contributo ha un andamento quadratico rispetto al numero  $N$  di particelle.
- Termine di Lennard-Jones. Il numero di coppie di atomi è limitato dai cut-off ma, considerando la complessità della funzione 6 – 12 e la necessità di ricombinare coefficienti noti su base atomica (vedi sezione 1.1.4), il tempo di calcolo complessivo è rilevante.
- Interazioni Coulombiane a lungo raggio. Calcolate con il meccanismo di Ewald, incidono con contributo quadratico rispetto a  $N$ .

### 1.2.4 Utenza

Nonostante la disponibilità di pacchetti più efficienti ed evoluti, anche di libero utilizzo [26], M.DynaMix è adottato da un'ampia comunità di scienziati ed è riuscito, nel corso degli anni, a mantenere un nutrito gruppo di utenti affezionati.

Il motivo di ciò è da ricercarsi principalmente nella sua diversità dagli standard. Da un lato, la precisione è tenuta in grande considerazione; tecniche come quella del doppio time step o della somma di Ewald vanno in questa direzione, anche se, in parte, a spese dell'efficienza. Dall'altro, nonostante sia sofisticato, il programma mantiene la sua semplicità, non solo di utilizzo, ma anche di comprensione e modifica. Il numero limitato di file e la linearità del codice, lo rendono un ottimo punto di partenza per coloro che desiderano implementare un nuovo algoritmo o realizzare una variante personale.

## 1.3 Il codice TRANAL

TRANAL (*TRAjectories ANALysis*) [69] è un pacchetto software che raggruppa un numero di strumenti per l'analisi delle traiettorie atomiche generate da programmi di Dinamica Molecolare.

Il programma è stato realizzato principalmente con l'obiettivo di elaborare i risultati delle simulazioni prodotte da M.DynaMix, ma è in grado di accettare traiettorie scritte in altri formati quali XMOL o PDB.

Dal punto di vista dell'implementazione, anche TRANAL è scritto in linguaggio F77. È disponibile unicamente una versione seriale del programma, eseguibile su macchine sequenziali.

### 1.3.1 Strumenti di analisi

Tra le possibilità offerte, il pacchetto consente il calcolo delle funzioni di distribuzione radiale (RDF), spaziale (SDF) e degli angoli di torsione; la determinazione dei tempi di residenza (RT), dei plot di Ramachandran e di diverse funzioni di correlazione temporale; la stima dei coefficienti di autodiffusione; il calcolo della permittività dielettrica media; il calcolo della densità di massa, della densità elettronica e del potenziale elettrostatico attraverso membrane.

Ricordiamo, per completezza, la definizione di alcune delle quantità appena citate, tra quelle che verranno utilizzate in seguito nel corso del presente lavoro.

#### RDF

In meccanica statistica la funzione di distribuzione radiale (RDF) è una funzione di correlazione di coppia di un sistema di particelle. Collocando l'origine del sistema di riferimento in corrispondenza della posizione di una delle particelle, la RDF è definita come la probabilità relativa di trovare un'altra particella ad una data distanza  $r$  dalla prima, rispetto alla medesima probabilità calcolata in un gas ideale con la stessa densità media.

L'informazione fornita dalle RDF, è rappresentabile sotto forma di grafico della probabilità relativa, in funzione della distanza dall'atomo considerato come riferimento.

## SDF

Una informazione più dettagliata è fornita dalla funzione di distribuzione spaziale (SDF), nella quale la probabilità relativa di trovare una particella, in una data collocazione rispetto ad un riferimento, è considerata dipendente anche dalle variabili angolari.

Per visualizzare le SDF è possibile utilizzare una rappresentazione per isosuperfici, ottenute considerando tutti punti dello spazio aventi la medesima, fissata, probabilità di occupazione relativa.

## RT

Il tempo di residenza (RT) di una particella rispetto ad un dato riferimento è il rapporto tra il tempo da essa trascorso a *contatto diretto* con il riferimento ed il tempo trascorso a distanze diverse. Usualmente, un atomo si considera a contatto diretto con un riferimento, se la sua distanza da esso è compresa nell'intervallo  $(1.0 \pm 0.1)d$ , con  $2.5\text{\AA} \leq d \leq 3\text{\AA}$ .

## 1.4 Il gradiente del campo elettrico

Una quantità che in linea di principio è possibile valutare a partire dalle traiettorie di Dinamica Molecolare, è il gradiente del campo elettrico (*Electric Field Gradient*, EFG). Esso è un campo tensoriale di *rango* = 2, ottenuto come derivata delle componenti del campo elettrico, nelle tre direzioni dello spazio. Il tensore EFG può essere valutato numericamente in un insieme discreto di punti, che nella maggior parte dei casi si ha interesse a prendere coincidente con le posizioni occupate dai singoli nuclei atomici.

Il programma TRANAL, così come la maggior parte dei codici di analisi comunemente disponibili, non realizza il calcolo di EFG.

## 2 Interazioni coulombiane a lungo raggio

Il calcolo delle interazioni Coulombiane a lungo raggio riveste una grande importanza nelle simulazioni di Dinamica Molecolare. Da un lato, l'accurata valutazione dei contributi elettrostatici, è indispensabile per ottenere risultati affidabili. Dall'altro, esso è il termine computazionalmente più oneroso e rappresenta tipicamente il collo di bottiglia principale nell'esecuzione dei calcoli.

Una grande varietà di metodi sono stati proposti per la determinazione del termine di Coulomb in MD. Tra questi, approcci che si basano sulla introduzione di una distanza di cut-off [39, 40, 41], notoriamente, non assicurano accuratezza sufficiente e producono deviazioni significative dai dati sperimentali [29]. Viceversa, i metodi basati sul meccanismo di somma di Ewald [30, 31] sono considerati, al momento attuale, la scelta più affidabile. Diversi schemi di questo tipo sono stati proposti negli ultimi anni [32, 33, 34, 35, 36, 37, 38]; in questo contesto, larga applicazione trovano gli algoritmi che fanno uso di griglie di punti, che consentano di limitare l'entità delle risorse computazionali necessarie, mantenendo accettabile il livello di precisione.

### 2.1 La somma di Ewald

Si consideri un sistema di  $N$  cariche puntiformi  $q_i$ , complessivamente neutro, collocate nelle posizioni  $r_i$  in una cella di simulazione cubica di lato  $L$ , periodicamente ripetuta nelle tre direzioni dello spazio.

L'energia potenziale elettrostatica in questo caso è esprimibile come

$$\begin{aligned}
U_{qq} &= \sum_{i=1}^N \sum_{j=i+1}^N \frac{q_i q_j}{|\mathbf{r}_{ij}|} + \frac{1}{2} \sum_{\mathbf{n} \neq 0} \sum_{i=1}^N \sum_{j=1}^N \frac{q_i q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|} \\
&= \frac{1}{2} \sum_{\mathbf{n}}^{\dagger} \sum_{i=1}^N \sum_{j=1}^N \frac{q_i q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|}, \tag{2.1}
\end{aligned}$$

dove  $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ , e  $|\mathbf{r}|$  indica la norma del vettore  $\mathbf{r}$ . Vista la natura a lungo raggio dell'interazione elettrostatica,  $U_{qq}$  include i contributi degli atomi della cella principale e di tutte le repliche, ma esclude l'auto-interazione. La somma più esterna dell'eq. (2.1), corre su tutti i vettori interi  $\mathbf{n} = (n_1, n_2, n_3)$  con  $n_i \in \mathbb{Z}$ ; il simbolo ( $\dagger$ ) indica che il termine di auto-interazione non è incluso, cioè, quando  $\mathbf{n} = 0$ , i termini con  $i = j$  sono omessi.

La serie dell'eq. (2.1) non è assolutamente convergente ma solo condizionatamente convergente <sup>1</sup>.

Come conseguenza di ciò, l'ordine con cui la somma è effettuata condiziona il valore della serie [42]. Perchè la somma abbia significato fisico, si assume che la costruzione del sistema avvenga per aggiunte successive di paricelle poste a distanza crescente, in modo tale che la carica addizionata al sistema, come pure la carica complessiva, sia di volta in volta nulla.

La somma sul reticolo dell'eq. (2.1) può essere calcolata con un metodo sviluppato inizialmente nel 1921 da Ewald [30]. Diverse derivazioni equivalenti della somma sono successivamente state proposte [31, 27]. Nel seguito, ci riferiremo principalmente al lavoro di de Leeuw *et al.* [31, 43, 44].

In [31] de Leeuw *et al.* propongono una tecnica basata sull'uso di un fattore di convergenza, che trasforma la somma di una serie condizionatamente convergente in una serie con somma ben definita; ciò inquanto l'applicazione di uno specifico fattore di convergenza equivale a fissare un ordine per la somma.

Si consideri un sistema di particelle complessivamente neutro,  $\sum_i q_i = 0$ , e si sommino i termini dell'eq. (2.1), su tutti i vettori interi  $\mathbf{n}$  per sfere concentriche di raggio crescente; è possibile mostrare che, nel caso in cui si assuma che la cella principale sia circondata da un mezzo dielettrico uniforme con costante  $\epsilon_r$  e le distanze siano calcolate con la convenzione della minima immagine, l'energia potenziale elettrostatica assume la forma

<sup>1</sup>Una serie  $\sum a_i$  è detta assolutamente convergente se  $\sum |a_i|$  converge. Quando  $\sum a_i$  converge e  $\sum |a_i|$  diverge, la serie è chiamata condizionatamente convergente.

$$U_{qq}(\epsilon_r) = \sum_{1 \leq i < j \leq N} \frac{q_i q_j}{L} \Psi\left(\frac{r_{ij}}{L}\right) + \frac{\xi}{2L} \sum_{i=1}^N q_i^2 + \frac{2\pi}{(2\epsilon_r + 1)L^3} \left(\sum_i q_i \mathbf{r}_i\right)^2. \quad (2.2)$$

Quando il mezzo circostante è conduttore, cioè  $\epsilon_r = \infty$ , l'energia può essere scritta come

$$\begin{aligned} U_{qq}(\epsilon_r = \infty) &= \sum_{1 \leq i < j \leq N} \frac{q_i q_j}{L} \Psi\left(\frac{r_{ij}}{L}\right) + \frac{\xi}{2L} \sum_{i=1}^N q_i^2 \\ &= U_{qq}(\epsilon_r) - \frac{2\pi}{(2\epsilon_r + 1)L^3} \left(\sum_i q_i \mathbf{r}_i\right)^2. \end{aligned} \quad (2.3)$$

Dalla forma dell'eq. (2.3) si deduce che le condizioni periodiche al contorno, sia nel caso di mezzo circostante conduttore che isolante, influiscono sull'energia del sistema e possono condizionare quindi il risultato di una simulazione [45, 46].

La funzione  $\Psi$  è data da

$$\Psi(\mathbf{r}) = \sum_{\mathbf{n}} \frac{\operatorname{erfc}(\alpha|\mathbf{r} + \mathbf{n}|)}{|\mathbf{r} + \mathbf{n}|} + \frac{1}{\pi} \sum_{\mathbf{n} \neq 0} \frac{\exp(2\pi i \mathbf{n} \cdot \mathbf{r} - \pi^2 |\mathbf{n}|^2 / \alpha^2)}{|\mathbf{n}|^2}, \quad (2.4)$$

con le due funzioni di errore definite come  $\operatorname{erfc}(x) = 1 - \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt$  and  $\operatorname{erf}(x) = 1 - \operatorname{erfc}(x)$ . Il numero  $\xi$  usato nell'Eqs. (2.2) e (2.3) è definito come

$$\xi = \sum_{\mathbf{n} \neq 0} \frac{\operatorname{erfc}(\alpha|\mathbf{n}|)}{|\mathbf{n}|} + \frac{1}{\pi} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / \alpha^2}}{|\mathbf{n}|^2} - \frac{2\alpha}{\sqrt{\pi}}, \quad (2.5)$$

dove  $\alpha > 0$  è un parametro libero.

Le equazioni (2.2) e (2.3) non sono in forma appropriata per un calcolo numerico efficiente; inoltre, nel caso delle simulazioni MD, è necessario sviluppare una espressione per le forze.

Per arrivare ad una forma più adeguata dell'eq. (2.2), cominciamo con l'inserire  $\mathbf{r}_{ij}/L$  nell'eq. (2.4)

$$\begin{aligned} \Psi\left(\frac{\mathbf{r}_{ij}}{L}\right) &= \sum_{\mathbf{n}} \frac{\operatorname{erfc}\left(\alpha\left|\frac{\mathbf{r}_{ij}}{L} + \mathbf{n}\right|\right)}{\left|\frac{\mathbf{r}_{ij}}{L} + \mathbf{n}\right|} + \frac{1}{\pi} \sum_{\mathbf{n} \neq 0} \frac{\exp\left(2\pi i \mathbf{n} \cdot \frac{\mathbf{r}_{ij}}{L} - \pi^2 |\mathbf{n}|^2 / \alpha^2\right)}{|\mathbf{n}|^2} \\ &= L \sum_{\mathbf{n}} \frac{\operatorname{erfc}\left(\frac{\alpha}{L} |\mathbf{r}_{ij} + \mathbf{n}L|\right)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{\pi} \sum_{\mathbf{n} \neq 0} \frac{\exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij} - \pi^2 |\mathbf{n}|^2 / \alpha^2\right)}{|\mathbf{n}|^2}. \end{aligned} \quad (2.6)$$

Successivamente, riscaldiamo  $\alpha$  tramite la sostituzione  $\alpha \rightarrow \alpha L$  nelle Equazioni (2.5) e (2.6) che porta a

$$\Psi\left(\frac{\mathbf{r}_{ij}}{L}\right) = L \sum_{\mathbf{n}} \frac{\operatorname{erfc}(\alpha|\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{\pi} \sum_{\mathbf{n} \neq 0} \frac{\exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij} - \pi^2 |\mathbf{n}|^2 / (\alpha L)^2\right)}{|\mathbf{n}|^2} \quad (2.7)$$

e

$$\xi = \sum_{\mathbf{n} \neq 0} \frac{\operatorname{erfc}(\alpha|\mathbf{n}|L)}{|\mathbf{n}|} + \frac{1}{\pi} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} - \frac{2\alpha L}{\sqrt{\pi}}. \quad (2.8)$$

Inserendo le Equazioni (2.7) e (2.8) nella eq. (2.2) avremo

$$\begin{aligned} U_{qq}(\epsilon_r) &= \sum_{1 \leq i < j \leq N} q_i q_j \left\{ \sum_{\mathbf{n}} \frac{\operatorname{erfc}(\alpha|\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{\pi L} \sum_{\mathbf{n} \neq 0} \frac{\exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij} - \pi^2 |\mathbf{n}|^2 / (\alpha L)^2\right)}{|\mathbf{n}|^2} \right\} \\ &+ \frac{1}{2L} \sum_i q_i^2 \left[ \sum_{\mathbf{n} \neq 0} \frac{\operatorname{erfc}(\alpha|\mathbf{n}|L)}{|\mathbf{n}|} + \frac{1}{\pi} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} - \frac{2\alpha L}{\sqrt{\pi}} \right] \\ &+ \frac{2\pi}{(2\epsilon_r + 1)L^3} \left( \sum_i q_i \mathbf{r}_i \right)^2. \end{aligned} \quad (2.9)$$

Si noti che la prima sommatoria dell'eq. (2.9) è per fatta per  $i < j$ . Faremo ulteriori semplificazioni analizzando il termine a destra dell'eq. (2.9) per  $n = 0$  e  $n \neq 0$ .

Quando  $n = 0$  avremo i seguenti termini:

$$\begin{aligned} &\sum_{1 \leq i < j \leq N} q_i q_j \frac{\operatorname{erfc}(\alpha|\mathbf{r}_{ij}|)}{|\mathbf{r}_{ij}|} - \frac{\alpha}{\sqrt{\pi}} \sum_i q_i^2 + \frac{2\pi}{(2\epsilon_r + 1)L^3} \left( \sum_i q_i \mathbf{r}_i \right)^2 \\ &= \frac{1}{2} \sum_{i,j}^{\dagger} \frac{q_i q_j}{|\mathbf{r}_{ij}|} \operatorname{erfc}(\alpha|\mathbf{r}_{ij}|) - \frac{\alpha}{\sqrt{\pi}} \sum_i q_i^2 + \frac{2\pi}{(2\epsilon_r + 1)L^3} \left( \sum_i q_i \mathbf{r}_i \right)^2 \end{aligned} \quad (2.10)$$

dove sono stati inclusi i termini indipendenti da  $\mathbf{n}$ . Il simbolo  $(\dagger)$  indica che i termini  $i = j$  sono esclusi dalla somma.

Con  $n \neq 0$  l'espressione assume la forma

$$\begin{aligned}
& \sum_{1 \leq i < j \leq N} q_i q_j \left\{ \sum_{\mathbf{n} \neq 0} \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{\pi L} \sum_{\mathbf{n} \neq 0} \frac{\exp(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij} - \pi^2 |\mathbf{n}|^2 / (\alpha L)^2)}{|\mathbf{n}|^2} \right\} \\
& \quad + \frac{1}{2L} \sum_i q_i^2 \left\{ \sum_{\mathbf{n} \neq 0} \frac{\operatorname{erfc}(\alpha |\mathbf{n}|L)}{|\mathbf{n}|} + \frac{1}{\pi} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \right\} \\
& = \frac{1}{2} \sum_{i,j} q_i q_j \sum_{\mathbf{n} \neq 0} \left\{ \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{\pi L} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \exp(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}) \right\}. \quad (2.11)
\end{aligned}$$

Il fattore  $1/2$  nell'eq. (2.11) deriva dal cambiamento degli estremi della somma, da  $i < j$  a tutte le coppie  $i$  and  $j$ , tenendo conto della simmetria indotta da  $\mathbf{r}_{ij} = -\mathbf{r}_{ji}$  e  $\pm \mathbf{n}$ .

Combinando le Equazioni (2.10) e (2.11) possiamo identificare il termine in spazio reale  $U_{qq}^{real}$ , il termine in spazio reciproco  $U_{qq}^{recip}$ , il termine di auto-interazione  $U_{qq}^{self}$  e il termine dipendente dalle condizioni periodiche al contorno  $U_{qq}^{bc}$ . Il termine in spazio reale è dato da

$$\begin{aligned}
U_{qq}^{real} & = \sum_{1 \leq i < j \leq N} q_i q_j \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij}|)}{|\mathbf{r}_{ij}|} + \sum_{i < j} \sum_{\mathbf{n} \neq 0} \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{2} \sum_i q_i^2 \sum_{\mathbf{n} \neq 0} \frac{\operatorname{erfc}(\alpha |\mathbf{n}|L)}{|\mathbf{n}|L} \\
& = \frac{1}{2} \sum_{\mathbf{n}} \sum_{i,j} \frac{q_i q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|} \operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|), \quad (2.12)
\end{aligned}$$

mentre il termine in spazio reciproco è

$$\begin{aligned}
U_{qq}^{recip} & = \frac{1}{\pi L} \sum_{i < j} \sum_{\mathbf{n} \neq 0} q_i q_j \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \exp(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}) + \\
& \quad \frac{1}{2\pi L} \sum_i \sum_{\mathbf{n} \neq 0} q_i q_i \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2}. \quad (2.13)
\end{aligned}$$

Tuttavia, con la simmetria generata da  $\mathbf{r}_{ij} = -\mathbf{r}_{ji}$  e  $\pm \mathbf{n}$ , avremo

$$\begin{aligned}
U_{qq}^{recip} & = \frac{1}{2\pi L} \sum_{i \neq j} \sum_{\mathbf{n} \neq 0} q_i q_j \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \exp(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}) + \frac{1}{2\pi L} \sum_i \sum_{\mathbf{n} \neq 0} q_i q_i \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \\
& = \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \sum_{i,j} q_i q_j \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \exp(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}). \quad (2.14)
\end{aligned}$$

Inoltre, avremo



$$U_{qq}^{self} = \frac{\alpha}{\sqrt{\pi}} \sum_i q_i^2, \quad (2.15)$$

$$U_{qq}^{bc} = \frac{2\pi}{(2\epsilon_r + 1)L^3} \left( \sum_i q_i \mathbf{r}_i \right)^2, \quad (2.16)$$

ed in sintesi

$$U_{qq}(\epsilon_r) = U_{qq}^{real} + U_{qq}^{recip} - U_{qq}^{self} + U_{qq}^{bc}. \quad (2.17)$$

La somma in spazio reciproco della eq. (2.14), può essere espansa in due differenti forme. La prima, in termini del fattore di struttura  $S(\mathbf{n})$ ,

$$S(\mathbf{n}) = \sum_i q_i \exp\left(-\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_i\right), \quad (2.18)$$

è data da

$$\begin{aligned} U_{qq}^{recip} &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \sum_{i,j} q_i q_j \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot (\mathbf{r}_i - \mathbf{r}_j)\right) \\ &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \sum_i q_i \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \sum_j q_j \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot (-\mathbf{r}_j)\right) \\ &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} S(-\mathbf{n}) S(\mathbf{n}). \end{aligned} \quad (2.19)$$

Per un dato  $\mathbf{n}$ , il fattore di struttura  $S(\mathbf{n})$  è un numero complesso  $a$ , e il fatto che  $a\bar{a} = Re(a)^2 + Im(a)^2$ , fornisce una seconda espressione, reale, dell'eq. (2.14)

$$\begin{aligned} U_{qq}^{recip} &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \sum_{i,j} q_i q_j \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot (\mathbf{r}_i - \mathbf{r}_j)\right) \\ &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \left[ \left( \sum_i q_i \cos\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \right)^2 + \right. \\ &\quad \left. + \left( \sum_i q_i \sin\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \right)^2 \right]. \end{aligned} \quad (2.20)$$

L'ultima forma dell'eq. (2.20) è il più comune punto di partenza per l'implementazione della parte in spazio reciproco. L'espressione precedente invece (eq. (2.19)) è usata negli approcci per il calcolo della somma in spazio reciproco su griglie di punti.

### 2.1.1 Calcolo delle forze

Una volta ottenuta una forma per l'energia elettrostatica del sistema è possibile calcolare il contributo alle forze  $\mathbf{F}_i$  agente su ciascuna particella  $i$ . Tenendo conto dei diversi termini in cui è stata partizionata l'energia e usando l'eq. (2.17), possiamo ottenere la forza elettrostatica totale come gradiente dell'energia elettrostatica

$$\begin{aligned}
 F_i &= -\nabla_i U_{qq} \\
 &= -\nabla_i \left[ U_{qq}^{real} + U_{qq}^{recip} - U_{qq}^{self} + U_{qq}^{bc} \right] \\
 &= F_i^{real} + F_i^{recip} - 0 + F_i^{bc}.
 \end{aligned} \tag{2.21}$$

L'indice  $i$  nell'operatore  $\nabla$  indica che deve essere considerata la derivata parziale rispetto alla posizione della particella  $\mathbf{r}_i$ ; lo zero nell'eq. (2.21) deriva dal fatto che il termine di auto-interazione (eq. (2.15)) è indipendente da  $\mathbf{r}_i$ .

Nel seguito, riportiamo un riassunto contenente alcune delle regole di derivazione che saranno utilizzate per la realizzazione del calcolo.

$$\begin{aligned}
 \nabla_i \mathbf{n} \cdot \mathbf{r}_{ij} &= \nabla_i \mathbf{n} \cdot (\mathbf{r}_i - \mathbf{r}_j) = \mathbf{n} \\
 \nabla_i \frac{1}{|\mathbf{r}_{ij}|} &= -\frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^3} \\
 \nabla_i |\mathbf{r}_{ij}| &= \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} \\
 \nabla_i \operatorname{erfc}(\alpha |\mathbf{r}_{ij}|) &= -\frac{2\alpha}{\sqrt{\pi}} \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} \exp(-\alpha^2 |\mathbf{r}_{ij}|^2) \\
 \nabla_i \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ji}\right) &= -\mathbf{n} \frac{2\pi i}{L} \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ji}\right) \\
 \nabla_i \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}\right) &= +\mathbf{n} \frac{2\pi i}{L} \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}\right).
 \end{aligned} \tag{2.22}$$

Applicando le formule appena introdotte, è possibile calcolare le  $F_i$ . Il contributo del termine in spazio reale  $F_i^{real}$ , diventa

$$\begin{aligned}
F_i^{real} &= -\nabla_i U_{qq}^{real} \\
&= -\nabla_i \left[ \frac{1}{2} \sum_{\mathbf{n}} \sum_{i,j}^{\dagger} \frac{q_i q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|} \operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|) \right] \\
&= \sum_{\mathbf{n}}^{\dagger} \sum_j q_i q_j \frac{\mathbf{r}_{ij} + \mathbf{n}L}{|\mathbf{r}_{ij} + \mathbf{n}L|^2} \left[ \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{2\alpha}{\sqrt{\pi}} \exp(-\alpha^2 |\mathbf{r}_{ij} + \mathbf{n}L|^2) \right]. \quad (2.23)
\end{aligned}$$

In questa relazione, quando  $n \neq 0$ , tutti i termini  $j$  sono inclusi mentre per  $n = 0$  solo i termini tali che  $j \neq i$ . Per il calcolo del contributo proveniente dal termine in spazio reciproco, è conveniente l'utilizzo dell'eq. (2.20) perchè espressa in termini del vettore posizione  $\mathbf{r}_i$  invece che della posizione relativa  $\mathbf{r}_{ij}$ . Considerando ciò, il contributo alla forza proveniente dal termine in spazio reciproco è dato da

$$\begin{aligned}
F_i^{recip} &= -\nabla_i U_{qq}^{recip} \\
&= -\nabla_i \left[ \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \left[ \left( \sum_i q_i \cos\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \right)^2 + \left( \sum_i q_i \sin\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \right)^2 \right] \right] \\
&= \frac{2q_i}{L^2} \sum_{\mathbf{n} \neq 0} \mathbf{n} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \left[ \sin\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \operatorname{Re}(S(\mathbf{n})) + \cos\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \operatorname{Im}(S(\mathbf{n})) \right]. \quad (2.24)
\end{aligned}$$

In ultimo, il contributo che dipende dalle condizioni periodiche al contorno

$$\begin{aligned}
F_i^{bc} &= -\nabla_i \left[ \frac{2\pi}{(2\epsilon_r + 1)L^3} \left( \sum_i q_i \mathbf{r}_i \right)^2 \right] \\
&= -\frac{4\pi}{(2\epsilon_r + 1)L^3} q_i \sum_j q_j \mathbf{r}_j. \quad (2.25)
\end{aligned}$$

### 2.1.2 Formule per l'energia e le forze nella somma di Ewald

Consideriamo un sistema periodico, ottenuto replicando nello spazio una cella cubica di lato  $L$  contenente  $N$  cariche puntiformi  $q_i$ , collocate in posizioni  $\mathbf{r}_i$  e tali che  $\sum_i q_i = 0$ . Assumiamo che il mezzo che circonda il sistema oltre il margine della cella principale sia un dielettrico uniforme con costante dielettrica  $\epsilon_r$ . Le distanze siano valutate con la convenzione della minima immagine.

Dopo l'espansione ed il riordino dei termini dell'eq. (2.2), riscaldando  $\alpha \rightarrow \alpha L$  e usando la simmetria indotta da  $\mathbf{r}_{ij} = -\mathbf{r}_{ji}$  and  $\pm \mathbf{n}$ , l'energia elettrostatica totale del sistema può essere scritta come

$$U_{qq}(\epsilon_r) = U_{qq}^{real} + U_{qq}^{recip} - U_{qq}^{self} + U_{qq}^{bc}, \quad (2.26)$$

con i differenti contributi dati da

$$U_{qq}^{real} = \frac{1}{2} \sum_{\mathbf{n}}^{\dagger} \sum_{i,j} \frac{q_i q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|} \operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|), \quad (2.27)$$

$$\begin{aligned} U_{qq}^{recip} &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \sum_{i,j} q_i q_j \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}\right) \\ &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \left[ \left( \sum_i q_i \cos\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \right)^2 + \left( \sum_i q_i \sin\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \right)^2 \right] \\ &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} S(\mathbf{n}) S(-\mathbf{n}), \end{aligned} \quad (2.28)$$

$$U_{qq}^{self} = \frac{\alpha}{\sqrt{\pi}} \sum_i q_i^2, \quad (2.29)$$

$$U_{qq}^{bc} = \frac{2\pi}{(2\epsilon_r + 1)L^3} \left( \sum_i q_i \mathbf{r}_i \right)^2. \quad (2.30)$$

Si noti che  $\alpha > 0$  è un parametro libero. Il fattore di struttura  $S(\mathbf{n})$  è definito come

$$S(\mathbf{n}) = \sum_i q_i \exp\left(-\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_i\right). \quad (2.31)$$

La forza elettrostatica totale  $F_i$  su ogni particella è

$$F_i = F_i^{real} + F_i^{recip} + F_i^{bc}. \quad (2.32)$$

I contributi alla forza totale sono dati da

$$F_i^{real} = \sum_{\mathbf{n}}^{\dagger} \sum_j q_i q_j \frac{\mathbf{r}_{ij} + \mathbf{n}L}{|\mathbf{r}_{ij} + \mathbf{n}L|^2} \left[ \frac{\operatorname{erfc}(\alpha|\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{2\alpha}{\sqrt{\pi}} \exp(-\alpha^2|\mathbf{r}_{ij} + \mathbf{n}L|^2) \right], \quad (2.33)$$

$$F_i^{recip} = \frac{2q_i}{L^2} \sum_{\mathbf{n} \neq 0} \mathbf{n} \frac{e^{-\pi^2|\mathbf{n}|^2/(\alpha L)^2}}{|\mathbf{n}|^2} \left[ \sin\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \sum_j q_j \cos\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_j\right) - \cos\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \sum_j q_j \sin\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_j\right) \right], \quad (2.34)$$

$$F_i^{bc} = -\frac{4\pi}{(2\epsilon_r + 1)L^3} q_i \sum_j q_j \mathbf{r}_j. \quad (2.35)$$

Il numero positivo  $\alpha$  è il cosiddetto parametro di convergenza di Ewald; esso è scelto in modo da avere il massimo vantaggio computazionale. Si noti che  $\operatorname{erfc}(x) \approx e^{-x^2}$  per grandi valori di  $x$ . Scegliendo  $\alpha$  abbastanza grande, è possibile fare il modo che, nell'eq. (2.27), il solo termine che contribuisca alla somma in spazio reale sia quello con  $\mathbf{n} = 0$ . Nell'approssimare la somma in spazio reciproco invece, si considera normalmente che i soli termini con  $|\mathbf{n}| < n_{cut}$  debbano essere inclusi.

Adottando un cut-off sia in spazio reale che in spazio reciproco, si può quindi far sì che i termini trascurati in entrambe le somme siano dell'ordine di  $\delta$ , o di ordine inferiore. Il troncamento in spazio reale implica che un sufficiente numero di termini debba essere incluso nella somma in spazio reciproco dell'eq. (2.28).

Scelta una certa accuratezza  $\delta$ ,  $n_{cut}$  è determinato da

$$e^{-\pi^2 n_{cut}^2 / (\alpha L)^2} \leq \delta \Rightarrow n_{cut} \geq \frac{\alpha L}{\pi} \sqrt{-\log(\delta)}, \quad (2.36)$$

mentre  $r_{cut}$  fissato da

$$\operatorname{erfc}(\alpha r_{cut}) \approx e^{-\alpha^2 r_{cut}^2} \leq \delta \Rightarrow r_{cut} = \frac{\pi n_{cut}}{\alpha^2 L}. \quad (2.37)$$

Sono presenti due condizioni e quattro parametri. Per una data  $\delta$  è possibile individuare un opportuno valore di  $n_{cut}$  tale che, tramite le due relazioni precedenti,  $\alpha$  e  $r_{cut}$  restino fissati.

Adottando una scelta ottimale dei parametri, il costo computazionale del metodo di Ewald risulta pari a  $\mathcal{O}(N^{3/2})$  [43]. Ciò rappresenta un notevole passo avanti rispetto alla

complessità di ordine  $\mathcal{O}(N^2)$  relativa alla raggio di interazione “infinito” dei termini Coulombiani.

## 2.2 Il metodo ENUF

In questa sezione descriveremo un approccio alla tecnica di somma di Ewald basato sull'utilizzo delle trasformate FFT non uniformi (*NFFT*).

Il metodo [63], noto con l'acronimo ENUF (*Ewald summation using NonUniform fast Fourier transform*), combina la tradizionale somma di Ewald con le NFFT, per calcolare l'energia elettrostatica e le forze nelle simulazioni di Dinamica Molecolare. Energia e momento sono conservati entro la precisione floating point. Tramite una scelta opportuna dei parametri di convergenza, ENUF può riprodurre la medesima accuratezza della tradizionale somma di Ewald ma ad un costo computazionale di  $\mathcal{O}(N \log N)$ , con  $N$  numero di interazioni elettrostatiche nel sistema.

### 2.2.1 Transformate di Fourier discrete per dati non equispaziati

La transformate di Fourier discrete per insiemi non equispaziati di dati [47] è una generalizzazione della FFT [48]. Svariati approcci simili sono stati proposti; alcuni esempi sono [49, 50, 51, 52, 53, 54, 55, 56, 57] con comparazioni offerte in [58, 54, 59].

L'idea alla base della NFFT è di combinare le trasformate standard e le combinazioni lineari di funzioni a supporto compatto (*window functions*), che siano ben localizzate sia nel dominio spaziale che in quello delle frequenze.

Nelle FFT, l'introduzione di un cut-off nel dominio delle frequenze o un limitato numero di termini nel dominio spaziale, conducono rispettivamente ad un errore di campionamento (*aliasing*) ed un errore di troncamento. L'errore di aliasing è controllato tramite il fattore di sovracampionamento  $\sigma_s$ , e l'errore di troncamento error dal numero di termini,  $m$ , nell'approssimazione spazio-temporale. Per una certa tipologia di window functions (Gaussian, B-spline, Sinc-power, Kaiser-Bessel), è stato mostrato come, per un fissato fattore di sovracampionamento  $\sigma_s > 1$ , l'errore decade esponenzialmente con  $m$  [60].

### 2.2.2 Definizione del problema

Nel problema del calcolo della trasformata discreta di Fourier per dati non equispaziati (NDFT), si vuole calcolare, per un insieme dato di coefficienti di Fourier  $\hat{\mathbf{f}}_{\mathbf{k}} \in \mathcal{C}$  con  $\mathbf{k} \in I_M$ , il polinomio trigonometrico  $f(x) = \sum_{\mathbf{k} \in I_M} \hat{\mathbf{f}}_{\mathbf{k}} \exp(-2\pi i \mathbf{k} \mathbf{x})$  in ognuno dei punti non equispaziati di un fissato insieme  $(\mathbf{x}_j \in \mathcal{D}^d, j = 0, \dots, N-1)$ . In letteratura, i punti per i quali è realizzato il campionamento sono spesso detti *nodi*.

I dettagli nella esecuzione della NDFT dipendono in grande misura dalle definizioni date per l'insieme di nodi  $\mathcal{D}^d$  e lo spazio degli indici  $I_M$ . Nel seguito della descrizione faremo riferimento principalmente a quanto presentato nel lavoro [60, 61].

### 2.2.3 Fondamenti teorici

Consideriamo a dominio  $d$ -dimensionale  $\mathcal{D}^d$  nel quale l'insieme di nodi non equispaziati è collocato. Sia

$$\mathcal{D}^d := \{\mathbf{x} = (x_t)_{t=0, \dots, d-1} \in \mathcal{R}^d : -\frac{1}{2} \leq x_t < \frac{1}{2}, t = 0, \dots, d-1\}, \quad (2.38)$$

e  $N$  l'insieme di nodi ( $\mathcal{X} := \{\mathbf{x}_j \in \mathcal{D}^d : j = 0, \dots, N-1\}$ ). Per le applicazioni usuali,  $d$  sarà tipicamente 2 o 3. Sia  $\mathcal{F}$  uno spazio funzionale di polinomi trigonometrici di grado  $M_t$  ( $t = 0, \dots, d-1$ ) in dimensione  $t$ ; lo spazio funzionale  $\mathcal{F}$  può essere definito come

$$\mathcal{F} := \{f : \mathcal{D}^d \mapsto \mathcal{C} \text{ such that } f \in \text{span}(e^{-2\pi i \mathbf{k} \cdot} : \mathbf{k} \in I_M)\}. \quad (2.39)$$

La dimensione di questo spazio funzionale è  $\dim(\mathcal{F}) = M_{\Pi}$ , where  $M_{\Pi} = \prod_{t=0}^{d-1} M_t$ . Le frequenze  $\mathbf{k} \in I_M$  con  $I_M$  insieme degli indici sono tali che

$$I_M := \{\mathbf{k} = (k_t)_{t=0, \dots, d-1} \in \mathcal{Z}^d : -\frac{M_t}{2} \leq k_t < \frac{M_t}{2}, t = 0, \dots, d-1\}. \quad (2.40)$$

### 2.2.4 Formulazione matrice-vettore

Per un insieme dato di coefficienti di Fourier  $\hat{\mathbf{f}}_{\mathbf{k}} \in \mathcal{C}$  con  $\mathbf{k} \in I_M$  si vuole valutare il polinomio trigonometrico  $f(x) = \sum_{\mathbf{k} \in I_M} \hat{\mathbf{f}}_{\mathbf{k}} \exp(-2\pi i \mathbf{k} \mathbf{x})$  in ognuno dei dati nodi non equispaziati in  $\mathcal{X}$ . Il prodotto  $\mathbf{k} \mathbf{x}$  è l'usuale prodotto scalare tra vettori  $\mathbf{k}$  e  $\mathbf{x}$ , definito come  $\mathbf{k} \mathbf{x} := k_0 x_0 + \dots + k_{d-1} x_{d-1}$ . Conseguentemente, per ogni  $\mathbf{x}_j \in \mathcal{X}$ , possiamo valutare  $\mathbf{f}_j = f(\mathbf{x}_j) := \sum_{\mathbf{k} \in I_M} \hat{\mathbf{f}}_{\mathbf{k}} \exp(-2\pi i \mathbf{k} \mathbf{x}_j)$ .

Quanto detto può essere riformulato in notazione matrice-vettore ponendo

$$\mathbf{f} := (f_j)_{j=0,\dots,N-1}, \quad \mathbf{A} := (e^{-2\pi i \mathbf{k} \mathbf{x}_j})_{j=0,\dots,N-1; \mathbf{k} \in I_M}, \quad \hat{\mathbf{f}} := (\hat{f}_{\mathbf{k}})_{\mathbf{k} \in I_M} \quad (2.41)$$

e scrivendo  $\mathbf{f} = \mathbf{A} \hat{\mathbf{f}}$ .

### 2.2.5 Tipologie di prodotti matrice-vettore

Sia  $\overline{\mathbf{A}}$  la complessa coniugata della matrice  $\mathbf{A}$  e  $\mathbf{A}^H = \overline{\mathbf{A}}^T$  la trasposta complessa coniugata della matrice  $\mathbf{A}$ . Considerando queste convenzioni, le diverse tipologie di prodotti matrice-vettore utilizzate nel contesto della NDFT sono:

**Regolare:**

$$\mathbf{f} = \mathbf{A} \hat{\mathbf{f}}, \quad f_j = \sum_{\mathbf{k} \in I_M} \hat{f}_{\mathbf{k}} \exp(-2\pi i \mathbf{k} \mathbf{x}_j). \quad (2.42)$$

**Aggiunta:**

$$\hat{\mathbf{f}} = \mathbf{A}^H \mathbf{f}, \quad \hat{f}_{\mathbf{k}} = \sum_{j=0}^{N-1} f_j \exp(2\pi i \mathbf{k} \mathbf{x}_j). \quad (2.43)$$

**Coniugata:**

$$\mathbf{f} = \overline{\mathbf{A}} \hat{\mathbf{f}}, \quad f_j = \sum_{\mathbf{k} \in I_M} \hat{f}_{\mathbf{k}} \exp(2\pi i \mathbf{k} \mathbf{x}_j). \quad (2.44)$$

**Trasposta:**

$$\hat{\mathbf{f}} = \mathbf{A}^T \mathbf{f}, \quad \hat{f}_{\mathbf{k}} = \sum_{j=0}^{N-1} f_j \exp(-2\pi i \mathbf{k} \mathbf{x}_j). \quad (2.45)$$

### 2.2.6 NDFT, FFT, NFFT

Considerando che la matrice  $\mathbf{A}$  è di dimensioni  $N \times M_{\Pi}$ , con  $M_{\Pi} = \dim(\mathcal{F}) = \prod_{t=0}^{d-1} M_t$ , l'esecuzione delle tipologie di NDFT scritte in notazione matrice-vettore nelle eq. (2.42)–(2.45), richiede un numero di operazioni aritmetiche dell'ordine di  $\mathcal{O}(NM_{\Pi})$ .

Tuttavia, per il caso speciale di  $M_t = M_{(t=0,\dots,d-1)}$  e  $N = M^d$  nodi equispaziati  $\mathbf{x}_{\mathbf{k}}$  ( $\mathbf{k} \in I_M$ ), le  $f_{\mathbf{k}}$  possono essere calcolate dai coefficienti di Fourier  $\hat{f}_{\mathbf{k}}$  tramite una trasformata veloce (FFT), con un numero di operazioni aritmetiche dell'ordine di  $\mathcal{O}(N \log N)$ .



La trasformata veloce di Fourier per dati non equispaziati (NFFT) è una generalizzazione della FFT. L'idea di base è di combinare l'uso di window functions, ben localizzate sia nel dominio spaziale che in quello delle frequenze, con la FFT standard. Gli algoritmi risultanti sono approssimati ed è possibile mostrare che alcuni di essi hanno complessità computazionale  $\mathcal{O}[M_{\Pi} \log M_{\Pi} + \log(1/\epsilon)N]$ , dove  $\epsilon$  è l'accuratezza desiderata [60].

### 2.2.7 Somma di Ewald veloce

Con l'uso di un set di parametri ottimali nella somma di Ewald, i tempi di calcolo della parte reale e reciproca dell'espressione sono approssimativamente uguali. Nelle simulazioni MD si usa scegliere lo stesso cut-off per il calcolo delle interazioni a corto raggio e per la parte reale della somma; con questa scelta non ottimale, al crescere del numero di particelle del sistema, la parte reciproca della sviluppo di Ewald risulta la più onerosa da valutare [62]. In considerazione di ciò, nella discussione seguente ci concentreremo sul termine in spazio reciproco della somma di Ewald, mostrando come è possibile ridurne la complessità computazionale.

Basandoci sulle definizioni date nelle sezioni precedenti 2.2.1, riformuleremo l'espressione in termini di NDFT. Successivamente, mostreremo come la NFFT possa essere applicata, portando a un metodo di Ewald basato su trasformate di Fourier non uniformi.

#### NFFT e termine in spazio reciproco

Secondo lo sviluppo ricavato in precedenza, l'energia elettrostatica può essere espressa in base all'eq. (2.28)

$$U_{qq}^{recip} = \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} S(\mathbf{n}) S(-\mathbf{n}), \quad (2.46)$$

con il fattore di struttura  $S(\mathbf{n})$  definito come

$$S(\mathbf{n}) = \sum_j q_j \exp\left(-\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_j\right). \quad (2.47)$$

Confrontando la definizione della NDFT trasposta, data nell'eq. (2.45), con il fattore di struttura in eq. (2.47), si può notare che essi hanno la stessa forma generale. Dopo un

riassetto degli indici, anche i limiti delle somme possono essere fatti coincidere; infatti, sostituendo le posizioni normalizzate  $\mathbf{x}_j = \mathbf{r}_j/L$ , e ponendo  $\hat{\mathbf{f}}_j = q_j$ , si può vedere che l'eq. (2.47) si presenta come una istanza in 3 dimensioni dell'equazione eq. (2.45) con  $\hat{\mathbf{f}}_{\mathbf{n}} = S(\mathbf{n})$ . Inoltre, assumendo che nella simulazione MD la cella principale sia centrata nell'origine del sistema di riferimento, si può assumere che le posizioni normalizzate si siano collocate nel dominio  $\mathcal{D}^d$  come definito nell'eq. (2.38).

Di conseguenza, è possibile usare l'approccio NDFT per calcolare le componenti del fattore di struttura. Dal punto di vista computazionale ciò comporta che, per valutare  $S(\mathbf{n})$ , si possano anche utilizzare algoritmi basati su NFFT invece che realizzare la somma come si presenta nella espressione originale di Ewald.

Riscrivendo l'eq. (2.46) in termini delle componenti di Fourier, avremo

$$\begin{aligned} U_{qq}^{recip} &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} S(\mathbf{n}) S(-\mathbf{n}) \\ &= \frac{1}{\pi L} \sum_{\mathbf{n}_z > 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} |\hat{\mathbf{f}}_{\mathbf{n}}|^2. \end{aligned} \quad (2.48)$$

Per il calcolo dell'energia,  $U_{qq}^{recip}$ , l'uso dell'eq. (2.48) comporta che si debba:

1. calcolare tutti i coefficienti  $\hat{\mathbf{f}}_{\mathbf{n}}$  usando la NDFT trasposta
2. riscalare ciascuna delle  $|\hat{\mathbf{f}}_{\mathbf{n}}|^2$  con il fattore dato nell' eq. (2.48)
3. sommare tutte le componenti riscalate.

### Forze reciproche

Si può calcolare il contributo dallo spazio reciproco alle forze, usando un approccio a quello adottato per l'energia. Nella formula seguente,  $Re(\bullet)$  e  $Im(\bullet)$  indicano rispettivamente la parte reale ed immaginaria dell'argomento. dall'eq. (2.1.1) si ha che

$$\begin{aligned} F_i^{recip} &= \frac{2q_i}{L^2} \sum_{\mathbf{n} \neq 0} \mathbf{n} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \left[ \sin\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) Re(S(\mathbf{n})) + \right. \\ &\quad \left. \cos\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_j\right) Im(S(\mathbf{n})) \right]. \end{aligned} \quad (2.49)$$

Considerando che il fattore di struttura  $S(\mathbf{n})$  è un numero complesso, l'espressione tra parentesi nell'equazione eq. (2.49) può essere scritta come la parte immaginaria di un prodotto:

$$\sin\left(\frac{2\pi}{L}\mathbf{n} \cdot \mathbf{r}_i\right)Re(S(\mathbf{n})) + \cos\left(\frac{2\pi}{L}\mathbf{n} \cdot \mathbf{r}_i\right)Im(S(\mathbf{n})) = Im\left[\exp\left(\frac{2\pi i}{L}\mathbf{n} \cdot \mathbf{r}_i\right)S(\mathbf{n})\right]. \quad (2.50)$$

Inserendo l'eq. (2.50) nell'eq. (2.49) si ha

$$\begin{aligned} F_i^{recip} &= \frac{2q_i}{L^2} \sum_{\mathbf{n} \neq 0} \mathbf{n} \frac{e^{-\pi^2|\mathbf{n}|^2/(\alpha L)^2}}{|\mathbf{n}|^2} Im\left[\exp\left(\frac{2\pi i}{L}\mathbf{n} \cdot \mathbf{r}_i\right)S(\mathbf{n})\right] \\ &= Im\left[\frac{2q_i}{L^2} \sum_{\mathbf{n} \neq 0} \mathbf{n} \frac{e^{-\pi^2|\mathbf{n}|^2/(\alpha L)^2}}{|\mathbf{n}|^2} \exp\left(\frac{2\pi i}{L}\mathbf{n} \cdot \mathbf{r}_i\right)S(\mathbf{n})\right]. \end{aligned} \quad (2.51)$$

Si noti che la (2.51) è un'equazione vettoriale. Inoltre, ciascuna delle tre componenti dell'espressione ha la stessa struttura della trasformazione NDFFT coniugata dell'eq. (2.44). Facendo uso delle posizioni normalizzate  $\mathbf{x}_j = \mathbf{r}_j/L$ , ed i coefficienti  $\hat{\mathbf{f}}_j = q_j$ ,

$$\hat{\mathbf{g}}_{\mathbf{n}} = \mathbf{n} \frac{e^{-\pi^2|\mathbf{n}|^2/(\alpha L)^2}}{|\mathbf{n}|^2} S(\mathbf{n}) \text{ for } \mathbf{n} \neq 0, \quad (2.52)$$

Si può vedere, anche in questo caso, che ogni componente dell'eq. (2.51) è un'istanza in tre dimensioni dell'eq. (2.44). Assumendo che l'indice  $\mathbf{n}$  appartenga all'insieme  $I_M$  dell'eq. (2.40),  $\mathbf{n} \in I_M$ , e ponendo  $\hat{\mathbf{g}}_0 = 0$ , possiamo scrivere  $F_i^{recip}$  direttamente in termini di trasformata di Fourier

$$\begin{aligned} F_i^{recip} &= Im\left[\frac{2q_i}{L^2} \sum_{\mathbf{n} \in I_M} \hat{\mathbf{g}}_{\mathbf{n}} e^{2\pi i \mathbf{n} \cdot \mathbf{x}_i}\right] \\ &= \frac{2q_i}{L^2} Im(\mathbf{g}_i). \end{aligned} \quad (2.53)$$

Calcolare la componente di spazio reciproco della forza sulla  $i$ -sima particella,  $F_i^{recip}$  usando l'eq. (2.53), comporta che si debba:

1. partire dalle componenti del fattore di struttura  $S(\mathbf{n})$  già calcolate durante la valutazione dell'energia  $U_{qq}^{recip}$ ,

2. riscaldare ogni  $S(\mathbf{n})$  usando l'eq. (2.52),
3. ottenere un nuovo insieme di coefficienti di Fourier che sono antitrasformati in spazio reale tramite la NDFT coniugata dell'eq. (2.53),
4. calcolare il valore della forza sulla particella  $i$  attraverso l'eq. (2.53), considerando la parte immaginaria del coefficiente  $\mathbf{g}_i$  riscaldato ponendo  $\frac{2q_i}{L^2}$ .

Anche in questo caso, si possono utilizzare algoritmi basati su NFFT invece che realizzare il calcolo attraverso l'esecuzione pedissequa della somma.

### Somma di Ewald e NFFT

La riformulazione di  $U_{qq}^{recip}$  and  $F_i^{recip}$  presentata nelle eq. (2.48) e (2.53) mostra il ruolo centrale delle NDFT trasposta e coniugata nel calcolo dei termini di spazio reciproco di energia e forze. Partendo dalle posizioni delle particelle cariche, il fattore di struttura  $S(\mathbf{n})$  viene valutato attraverso una NDFT trasposta. Riscaldando i valori assoluti delle componenti di Fourier e sommando, si ottiene  $U_{qq}^{recip}$ .

Per determinare le forze  $F_i^{recip}$  si realizza una controtrasformata, dallo spazio reciproco a quello reale, prima calcolando le componenti di Fourier delle forze e quindi operando una NDFT coniugata.

L'implementazione concreta del meccanismo di somma di Ewald fa uso di un cut-off in spazio reciproco,  $n_{cut}$  e di un cut-off in spazio reale  $r_{cut}$ ; per  $\alpha$  sufficientemente grande, i cut-off fissano l'accuratezza  $\delta$ , nel senso che troncando le somme delle eq. (2.27) e (2.28) ai rispettivi cut-off si avrà :

$$e^{-\pi^2|\mathbf{n}|^2/(\alpha L)^2} \leq \delta \Rightarrow n_{cut} \geq \frac{\alpha L}{\pi} \sqrt{-\log(\delta)} \Rightarrow n_{cut} \propto L \propto N^{1/3}. \quad (2.54)$$

e

$$\text{erfc}(\alpha r_{cut}) \approx e^{-\alpha^2 r_{cut}^2} \leq \delta \Rightarrow r_{cut} = \frac{\pi n_{cut}}{\alpha^2 L}. \quad (2.55)$$

Nelle nuove espressioni per il calcolo di  $U_{qq}^{recip}$  e  $F_i^{recip}$  in termini di componenti di Fourier abbiamo posto  $M_t = 2\sigma_s n_{cut}$ , dove  $\sigma_s$  prende il nome di *fattore di oversampling*. Ciò fornisce  $M_{\Pi} = \prod_{t=0}^{d-1} M_t$ . In generale, la complessità computazionale del metodo

NFFT è  $\mathcal{O}(M_{\Pi} \log M_{\Pi} + \log(1/\epsilon)N)$ , dove  $\epsilon$  è l'accuratezza desiderata per la NFFT [60]. Usando l'eq. (2.54) e la definizione precedente di  $M_{\Pi}$ , la complessità computazionale assume la forma  $\mathcal{O}(N \log N + \log(1/\epsilon)N)$ . Si noti che  $\epsilon$  è funzione di  $m$  per un fissato fattore di oversampling. Vista la possibilità di controllare l'accuratezza nel calcolo del fattore di struttura tramite la NFFT, la complessità computazionale  $\mathcal{O}(N^{3/2})$  diventa  $\mathcal{O}(N \log N)$ .

## 2.3 Il metodo SPME

L'algoritmo SPME è un metodo di calcolo delle interazioni Coulombiane ampiamente utilizzato in Dinamica Molecolare e le sue caratteristiche in termini di accuratezza ed efficienza sono ben note [35]. Esso consente di realizzare il calcolo del contributo in spazio reciproco, in modo efficiente e in condizioni di precisione controllata, tramite l'uso di una griglia discreta di punti e della FFT per la stima delle somme. Il costo computazionale del metodo è di ordine  $\mathcal{O}(N \log N)$ , con  $N$  numero di particelle interagenti.

### 2.3.1 Interpolazione del fattore di struttura

Per descrivere i dettagli matematici su cui il metodo si basa, estenderemo la simbologia utilizzata nel par. 2.1, così da fare riferimento alla notazione più diffusa, relativa al lavoro di Essmann *et al.* [35].

Consideriamo un sistema di  $N$  cariche puntiformi  $q_i$ , con  $\sum_i q_i = 0$ , collocate nelle posizioni  $\mathbf{r}_i$  in una cella di simulazione periodicamente ripetuta nelle tre direzioni dello spazio.

Rilassando la condizione che la cella di simulazione sia cubica, possiamo ritenerla individuata da una generica terna di vettori  $\mathbf{a}_{\alpha}$ , con  $\alpha = 1, 2, 3$  non necessariamente ortogonali. Alla terna  $\mathbf{a}_{\alpha}$  è possibile associare una terna reciproca  $\mathbf{a}_{\alpha}^*$ , tale che  $\mathbf{a}_{\alpha} \cdot \mathbf{a}_{\beta}^* = \delta_{\alpha\beta}$ , con  $\delta$  delta di Kronecker.

Considerando la simmetria indotta da  $\pm \mathbf{n}$ , è possibile riscrivere il fattore di struttura, definito in (2.18), come

$$S(\mathbf{m}) = \sum_j q_j \exp(2\pi i \mathbf{m} \cdot \mathbf{r}_j) = \sum_j q_j \exp\left[2\pi i(m_1 s_{1j} + m_2 s_{2j} + m_3 s_{3j})\right] \quad (2.56)$$

con  $\mathbf{m} = \sum_{\alpha} m_{\alpha} \mathbf{a}_{\alpha}^*$  vettore del reticolo reciproco e  $\mathbf{r}_j = \sum_{\alpha} s_{j\alpha} \mathbf{a}_{\alpha}$ .

Per approssimare il fattore di struttura, è necessario interpolare l'esponenziale complesso dell'eq. (2.56). L'interpolazione avverrà in pratica, utilizzando un insieme discreto di punti nella cella di simulazione (*nodi*) ed una famiglia di funzioni ben localizzate centrate su ogni nodo. Considerando che la cella di simulazione è ripetuta periodicamente nello spazio, l'energia elettrostatica data nell'eq. (2.30) risulterà ottenibile in termini di trasformate di Fourier discrete tridimensionali, calcolate sull'insieme di nodi introdotti.

Dati tre interi positivi  $K_1, K_2, K_3$  e un punto  $\mathbf{r}$  della cella principale, indichiamo con  $u_1, u_2, u_3$ , le sue *coordinate frazionarie*, definite come

$$u_{\alpha} = K_{\alpha} \mathbf{a}_{\alpha}^* \cdot \mathbf{r}, \quad (2.57)$$

con  $\alpha = 1, 2, 3$ .

Per le condizioni periodiche al contorno, possiamo assumere  $0 \leq u_{\alpha} \leq K_{\alpha}$ . Il generico termine della somma (2.56) può quindi essere riscritto come

$$\exp(2\pi i \mathbf{m} \cdot \mathbf{r}) = \exp\left(2\pi i \frac{m_1 u_1}{K_1}\right) \cdot \exp\left(2\pi i \frac{m_2 u_2}{K_2}\right) \cdot \exp\left(2\pi i \frac{m_3 u_3}{K_3}\right). \quad (2.58)$$

All'atto pratico,  $K_1, K_2, K_3$  rappresenteranno in numero di nodi della griglia discreta introdotta nella cella elementare di simulazione, in ciascuna direzione dello spazio.

### 2.3.2 Sviluppo in B-splines

Uno sviluppo dell'esponenziale complesso può essere fatto considerando la famiglia di funzioni a supporto compatto  $M_n(u)$ , nota come *Cardinal B-splines*. A partire dalla  $M_2(u)$ , che ha la forma

$$M_2(u) = \begin{cases} 1 - |u - 1|, & \text{se } 0 \leq u \leq 2 \\ 0, & \text{se } u < 0 \text{ o } u > 2 \end{cases} \quad (2.59)$$

è possibile definire le altre funzioni della famiglia in modo ricorsivo, secondo la relazione

$$M_n(u) = \frac{u}{n-1}M_{n-1}(u) + \frac{n-u}{n-1}M_{n-1}(u-1) \quad (2.60)$$

Come si può ricavare dalla definizione, le funzioni  $M_n(u)$  sono ben localizzate nel dominio spaziale; infatti, esse sono nulle all'esterno dell'intervallo  $0 \leq u \leq n$ . Il numero di  $n$  di splines utilizzate, detto *ordine* dello sviluppo, determina la precisione dell'interpolazione, più elevata al crescere di  $n$ .

Le Cardinal B-splines così definite sono differenziabili; per esse vale la relazione ricorsiva

$$\frac{d}{du}M_n(u) = M_{n-1}(u) - M_{n-1}(u-1). \quad (2.61)$$

Facendo uso delle  $M_n(u)$ , è possibile fornire un'approssimazione per l'esponenziale complesso che si può dimostrare avere la forma [35]

$$\exp(2\pi i \frac{m_i}{K_i} u_i) \approx b_i(m_i) \sum_{-\infty}^{\infty} M_n(u_i - k) \cdot \exp(2\pi i \frac{m_i}{K_i} k) \quad (2.62)$$

dove  $b_i(m_i)$  è dato da

$$b_i(m_i) = \exp \left[ 2\pi i (n-1) \frac{m_i}{K_i} \right] \cdot \left[ \sum_{k=0}^{n-2} M_n(k+1) \exp(2\pi i \frac{m_i}{K_i} k) \right]^{-1}. \quad (2.63)$$

Utilizzando l'approssimazione descritta nelle precedenti sezioni, è possibile riscrivere il fattore di struttura  $S(\mathbf{m})$ , definito dall'eq. (2.56), come

$$S(\mathbf{m}) = b_1(m_1)b_2(m_2)b_3(m_3)\mathcal{F}[Q](m_1, m_2, m_3), \quad (2.64)$$

dove la funzione  $Q$  è data da

$$Q(k_1, k_2, k_3) = \sum_{i=1}^N \sum_{n_1, n_2, n_3}^N q_i M_n(u_{1i} - k_1 - n_1 K_1) \cdot M_n(u_{2i} - k_2 - n_2 K_2) \cdot M_n(u_{3i} - k_3 - n_3 K_3). \quad (2.65)$$

### 2.3.3 Energia

Utilizzando l'approssimazione del fattore di struttura delineata in precedenza, è possibile riscrivere la parte reciproca dell'energia elettrostatica, data nell'eq. (2.30), come dipendente dalla trasformata di Fourier discreta tridimensionale della funzione  $Q(k_1, k_2, k_3)$ :

$$U_{qq}^{recip} = \frac{1}{2\pi V} \sum_{\mathbf{m} \neq 0} \frac{\exp(-\pi^2 \mathbf{m}^2 / \beta^2)}{\mathbf{m}^2} \cdot B(m_1, m_2, m_3) \cdot \mathcal{F}[Q](m_1, m_2, m_3) \cdot \mathcal{F}[Q](-m_1, -m_2, -m_3) \quad (2.66)$$

dove  $\mathcal{F}[Q](m_1, m_2, m_3)$  indica la trasformata discreta di Fourier di  $Q(k_1, k_2, k_3)$  ed il fattore  $B(m_1, m_2, m_3)$ , connesso all'interpolazione dell'esponenziale in termini di B-splines, è dato da

$$B(m_1, m_2, m_3) = |b_1(m_1)|^2 |b_2(m_2)|^2 |b_3(m_3)|^2. \quad (2.67)$$

Se si pone

$$C(m_1, m_2, m_3) = \begin{cases} \frac{1}{\pi V} \frac{\exp(-\pi^2 \mathbf{m}^2 / \beta^2)}{\mathbf{m}^2} & \text{se } \mathbf{m} \neq 0 \\ 0, & \text{se } \mathbf{m} = 0 \end{cases}, \quad (2.68)$$

la  $U_{qq}^{recip}$  potrà essere scritta come

$$U_{qq}^{recip} = \frac{1}{2} \sum_{m_1=0}^{K_1-1} \sum_{m_2=0}^{K_2-1} \sum_{m_3=0}^{K_3-1} Q(m_1, m_2, m_3) \cdot (\Theta_{rec} \star Q)(m_1, m_2, m_3) \quad (2.69)$$

dove

$$\Theta_{rec} = \mathcal{F}[B \cdot C] \quad (2.70)$$

ed il simbolo  $(\star)$  indica l'operazione di convoluzione discreta.



### 2.3.4 Forze

La tre componenti della forza reciproca  $F_{\alpha i}^{recip}$  sulla  $i$ -ma particella sono ottenute come derivata dell'energia rispetto alla posizione  $\mathbf{r}_i$ ; viste le relazioni precedenti e considerando che il termine  $\Theta_{rec}$  è indipendente dalle coordinate, avremo

$$F_{\alpha i}^{recip} = -\nabla_{\alpha i} U_{qq}^{recip} = \frac{dU_{qq}^{recip}}{d\mathbf{r}_{\alpha i}} = \sum_{m_1=0}^{K_1-1} \sum_{m_2=0}^{K_2-1} \sum_{m_3=0}^{K_3-1} \frac{dQ}{d\mathbf{r}_{\alpha i}}(m_1, m_2, m_3) \cdot (\Theta_{rec} \star Q)(m_1, m_2, m_3). \quad (2.71)$$

### 2.3.5 Algoritmo

L'applicazione dell'algoritmo SPME è schematicamente riassumibile nei seguenti passi:

- **Griglia e coordinate frazionarie.** Fissato il numero di punti  $(K_1, K_2, K_3)$  della griglia discreta nella cella di simulazione, e l'ordine  $n$  dello sviluppo in B-splines, si determinano le coordinate frazionarie  $u$  di ciascun atomo, in base all'eq. (2.57)
- **Approssimazione delle carica.** Tramite la valutazione delle  $M_n(u)$ , si determina l'interpolazione delle cariche  $Q(k_1, k_2, k_3)$  con, l'eq. (2.65).
- **FFT ed energia.** Si ottiene la FFT  $\mathcal{F}[Q](m_1, m_2, m_3)$  del vettore delle cariche e, tramite  $B(m_1, m_2, m_3)$ , si calcola l'energia con l'eq. (2.66).
- **FFT inversa e forze.** Si realizza il prodotto tra il vettore  $Q(m_1, m_2, m_3)$  ed i vettori  $B$  e  $C$  e conseguentemente si calcolano le forze tramite la (2.71); le  $dQ/d\mathbf{r}_{\alpha i}$  sono ottenute dalla conoscenza analitica delle derivate delle  $M_n(u)$ , espressa attraverso l'eq. (2.61).

### 3 Evoluzioni di M.DynaMix e TRANAL

I codici di calcolo sono un fondamentale strumento di lavoro per coloro che si occupano di modelling computazionale. Pacchetti di elevata complessità, richiedono usualmente un'adeguata fase di apprendimento prima di poter essere utilizzati e maneggiati con padronanza.

Nel campo delle simulazioni di Dinamica Molecolare classica, il risultato di uno studio computazionale o la tempistica con cui esso viene prodotto, possono essere influenzati notevolmente dalle caratteristiche del pacchetto di calcolo utilizzato; in questo senso, molto rilevanti sono la precisione e l'efficienza intrinseche del programma. La precisione è condizionata in larga misura dalla tipologia degli algoritmi adottati; la situazione è complessa perchè alcune tecniche di calcolo, che risultano adeguate per una data classe di sistemi, possono non esserlo per altri. Per quanto riguarda l'efficienza invece, occorre sottolineare che i tempi di calcolo delle simulazioni MD possono essere anche molto lunghi e riuscire a completare il proprio studio nel minor tempo possibile è un obiettivo di primaria importanza. Anche in questo caso, l'efficienza computazionale è legata principalmente al tipo di algoritmi utilizzati ed in particolare alle approssimazioni e semplificazioni presenti; si tratta quindi di un elemento strettamente connesso alla precisione. Ciò spesso costringe il ricercatore ad effettuare una scelta e trovare un adeguato punto di accordo, tra esigenze in larga misura contrastanti.

Nella scelta del programma di calcolo, ampia importanza riveste anche la tipologia di piattaforme su cui il pacchetto può essere eseguito. L'insieme delle risorse computazionali a cui si ha accesso, il budget dei tempi macchina, possono essere elementi da tenere in debita considerazione.

Nell'ambito degli studi MD, sono molti i pacchetti di simulazione disponibili. Nonostante la varietà cui il ricercatore ha accesso, è abbastanza comune il fatto che nessun pacchetto soddisfi completamente tutte le esigenze; la ricerca di un possibile compromesso tra i diversi elementi, è quindi parte frequente della pianificazione.

In alternativa, si può prendere in considerazione l'opportunità di intervenire sul pro-

gramma modificando il codice, così da adattarlo alle esigenze che si intendono soddisfare. Questa opportunità si verifica in un numero ridotto di casi, per via della complessità dei pacchetti e dell'ampio investimento di tempo e risorse che l'operazione potrebbe richiedere.

## 3.1 Motivazioni

In fase di pianificazione della ricerca di dottorato, è emerso che il codice M.DynaMix ed il pacchetto TRANAL sarebbero stati strumenti per molti versi adatti all'esecuzione degli studi computazionali programmati. La presenza del metodo di integrazione delle equazioni del moto basato sul doppio time step, è stata infatti ritenuta ottimale per la descrizione del comportamento di sistemi altamente carichi, oggetto finale delle applicazioni. Sebbene precisi e versatili, nei due pacchetti software erano però presenti alcune limitazioni, che li rendevano non perfettamente idonei.

Avendo le competenze tecniche necessarie e potendo pianificare un'attività a lungo termine, distribuita sull'intera lunghezza del dottorato, si è deciso di investire una parte del tempo nella realizzazione di modifiche ed evoluzioni del software, così da superare le limitazioni esistenti ed arrivare a disporre di un pacchetto pienamente adeguato per lo studio computazionale programmato.

Nella decisione presa, ha influito anche il riscontro di cui il programma gode nella comunità scientifica; M.DynaMix e TRANAL sono infatti diffusi in un'ampio gruppo di ricercatori ed utilizzati per lo studio di molte tipologie di sistemi [21, 22, 23, 24]. L'importanza, per molti soggetti, di disporre di strumenti all'avanguardia, è stata presa in debita considerazione ed ha rafforzato l'idea di intraprendere il lavoro di revisione dei codici.

I programmi hanno quindi subito miglioramenti su diversi fronti, attraverso l'introduzione di nuove funzionalità, l'aumento dell'efficienza di calcolo e l'ampliamento dello spettro di piattaforme hardware su cui poterne realizzare l'esecuzione.

## 3.2 Analisi preliminare

Nello stadio iniziale dell'attività, è stata effettuata un'analisi approfondita delle caratteristiche e delle limitazioni di M.DynaMix. In particolare, è stata eseguito un det-

tagliato studio dell'efficienza computazionale, sia complessiva che relativa alle diverse sottoprocedure.

In ingegneria del software, lo studio dell'efficienza computazionale di un programma viene detto *profiling*. Esso riguarda normalmente importanti proprietà, quali i tempi di esecuzione e la memoria occupata. Il profiling, eseguito con l'ausilio di opportuni programmi dedicati, permette di avere una misura oggettiva e precisa della situazione, anche al variare dei dati in ingresso. Nel nostro caso, il profiling iniziale ha consentito di individuare colli di bottiglia e debolezze e pianificare le opportune contromisure; ciò, anche in relazione alle dimensioni del sistema simulato che, come sempre accade nella MD, hanno un ruolo cardine.

### 3.3 Principali limitazioni di M.DynaMix e TRANAL

Uno degli inconvenienti principali riscontrati in M.DynaMix, è stata la non elevata efficienza computazionale; questo è apparso subito essere un serio limite, in grado di condizionare negativamente l'applicabilità del codice.

Il profiling preliminare, ha individuato la causa principale dello scarso rendimento, nel calcolo del termine Coulombiano a lungo raggio. Nel caso di M.DynaMix, il tempo di calcolo relativo a questo fattore è particolarmente elevato, visto l'utilizzo dell'algoritmo di Ewald; il metodo, notoriamente preciso quanto costoso in termini di risorse computazionali, è tanto più influente quanto maggiore è la dimensione del sistema, con un andamento proporzionale a  $N^{3/2}$  rispetto al numero  $N$  di particelle interagenti.

Un'altra debolezza riscontrata è connessa alla gestione della memoria. A causa dal meccanismo di allocazione statica, tipico del FORTRAN 77, il programma è incapace di adattarsi automaticamente alle dimensioni del sistema simulato. Può accadere quindi di allocare memoria RAM in modo sovrabbondante o insufficiente; nella prima situazione, si potrebbe avere uno spreco di risorse, anche elevato, con effetti negativi per il rendimento complessivo della piattaforma di calcolo utilizzata per l'esecuzione. Nella seconda situazione, l'utente sarebbe costretto a reimpostare a mano la dimensione delle principali strutture dati, per poi ricompilare, successivamente, il codice sorgente.

Come prospettiva generale infine, mentre l'esecuzione di M.DynaMix su macchine seriali o parallele di architettura standard, è immediata e non presenta particolari problemi, l'utilizzo di risorse di calcolo differenti, come ad esempio quelle basate su processori grafici (GPU) è completamente preclusa.

### 3.3.1 Lo stato di TRANAL

Di norma, nella fase di analisi delle traiettorie, i tempi di esecuzione e le efficienze di calcolo dei programmi non hanno un ruolo così rilevante come nella dinamica. La principale limitazione del pacchetto TRANAL consisteva quindi esclusivamente nella assenza di un meccanismo di calcolo del gradiente del campo elettrico (*Electric Field Gradient*, EFG), indispensabile per la parte della ricerca riguardante nuovi possibili meccanismi di validazione delle simulazioni MD.

## 3.4 Migliorie

Una volta in possesso dei risultati della fase preliminare di analisi, è stato possibile pianificare il lavoro successivo, definendo l'elenco di migliorie da apportare ai programmi ed il tempo stimato per ciascun intervento.

Di seguito, è schematizzato un elenco delle principali evoluzioni apportate sui codici.

### 3.4.1 Elettrostatica sulla griglia

Per diminuire il tempo di esecuzione complessivo e migliorare le performance, si è intervenuti sul meccanismo di calcolo del termine Coulombiano a lungo raggio, rendendo disponibili in M.DynaMix, oltre all'algoritmo di Ewald, due ulteriori algoritmi approssimati.

Le due efficienti procedure, SPME ed ENUF, descritte nel cap. 2, operano su una griglia discreta di punti utilizzando le trasformate di Fourier veloci e rendono possibile una valutazione più rapida del contributo elettrostatico, che può avvenire con accuratezza controllata, in un tempo dell'ordine di  $\mathcal{O}(N \log N)$ , con  $N$  numero di particelle del sistema.

### 3.4.2 Allocazione dinamica della memoria

Con il FORTRAN come strumento universalmente adottato, l'allocazione statica della memoria era una prassi obbligatoria nella chimica computazionale degli anni 90. Il linguaggio non offriva infatti nessuna possibilità di manipolazione dinamica ed il C era ancora lontano dal diffondersi tra gli (scettici) specialisti del settore. Ai nostri giorni le cose sono molto cambiate. In particolare, è noto che, con i moderni compilatori, moduli

di codice scritti in linguaggi diversi, quali ad esempio FORTRAN e C, possono facilmente coesistere [68].

Contando sulle nuove tecnologie disponibili, è stato messo a punto, in M.DynaMix, un meccanismo di gestione dinamica della memoria per le principali strutture di dati; il programma, è stato dotato della capacità di adattarsi automaticamente alle dimensioni del sistema simulato, utilizzando sempre il corretto ammontare di memoria RAM e liberando l'utente dall'onere di impostare a mano parametri e dimensioni.

### 3.4.3 Acceleratori computazionali

La possibilità di usare gli acceleratori computazionali basati su *Graphics Processor Units* (GPU), per incrementare le prestazioni delle applicazioni scientifiche, è oggi di estrema attualità. Le GPU si propongono come sistemi efficienti e a basso costo, con cui incrementare le prestazioni e ridurre i tempi di esecuzione dei codici. Nel settore della chimica computazionale, in particolare, si è assistito negli ultimi anni ad un grosso incremento di interesse, che si è concretizzato nell'adattamento di codici per questa nuova tipologia di hardware (*porting*), con relativa messa a punto di algoritmi ottimizzati.

In quest'ottica, è stata creata una versione di M.DynaMix che sfrutta le potenzialità dei processori grafici. Adattando gli algoritmi, le procedure più onerose dal punto di vista del tempo di calcolo, sono state re-implementate nella nuova architettura, consentendo un sensibile miglioramento delle performance. La nuova versione del programma è provvista anche del supporto *multi-GPU*, che consente sfruttare la presenza contemporanea più schede grafiche.

## 3.5 Scansione temporale delle modifiche

Vista la complessità del compito, si è preferito partizionare le attività di evoluzione del software in fasi autonome. In particolare, le modifiche del codice M.DynaMix elencate in precedenza, hanno avuto luogo in due passi separati; una ulteriore parte del lavoro, indipendente dalle prime due, ha avuto come oggetto l'implementazione del calcolo del gradiente del campo elettrico.

### 3.5.1 Fase 1: lungo raggio

Nella prima delle due fasi di modifica, partendo da una versione stabile del codice distribuita dagli autori (*versione 5.2.1 del settembre 2011*), sono stati introdotti in M.DynaMix gli algoritmi SPME e ENUF. I due metodi, autonomi e indipendenti, si sono affiancati al precedente esistente, la somma di Ewald, che è rimasto attivo.

In questa fase dell'implementazione si è scelto di seguire un approccio *conservativo*; l'architettura generale del programma non è stata modificata e le principali strutture dati sono rimaste invariate. Facendo leva anche sulle diffuse somiglianze tra i due algoritmi, le nuove funzionalità sono state introdotte nel modo meno invasivo possibile, rispettando le scelte stilistiche e di programmazione dei precedenti autori.

Successivamente a questa prima revisione, è stata realizzata un'ampia serie di test e verifiche, con l'obiettivo di evidenziare tutte le problematiche dovute a malfunzionamenti e banchi. Come risultato, si è ottenuta una versione stabile del codice, priva di problematiche evidenti, eventualmente disponibile su richiesta.

### 3.5.2 Fase 2: GPU e RAM dinamica

La seconda e più profonda fase di modifica di M.DynaMix, ha avuto come punto di partenza la versione stabile ottenuta dalla prima revisione. Il nuovo insieme di migliorie, ha riguardato la creazione di una versione del programma eseguibile su architetture GPU e la messa a punto del meccanismo di allocazione dinamica della memoria.

Per raggiungere gli obiettivi, è stato necessario rilassare gli stretti vincoli adottati nella prima fase e accettare di compiere una ristrutturazione più netta del codice, modificando l'architettura generale anche in modo significativo. In particolare, la volontà di interfacciarsi con gli acceleratori computazionali basati su GPU, ha richiesto la creazione di un opportuno insieme di dati aggiuntivi e di un gruppo di sottoprocedure *ad-hoc*.

Anche al termine di questa fase di modifiche, il codice è stato sottoposto ad una intensa procedura di test, che ne ha verificato l'integrità.

### 3.5.3 Fase 3: EFG

L'aggiunta del meccanismo di calcolo del gradiente del campo elettrico è stata realizzata su un codice a se stante battezzato TRILly. TRILly è stato creato in una fase indipendente e temporalmente autonoma dalle prime due.

## 3.6 Implementazione del metodo SPME

Come illustrato nella sez. 2.3, i metodi di calcolo delle interazioni elettrostatiche basati sul meccanismo di Ewald, esprimono l'energia come somma di diversi contributi. Tra questi, mentre la parte in spazio reale è rapidamente convergente e viene valutata con poco impegno di risorse, utilizzando un opportuno cut-off  $r_{cut}$ , il contributo in spazio reciproco è il termine più rilevante dal punto di vista computazionale.

Se calcolato con il metodo di Ewald assumendo di adottare il valore ottimale dei parametri (sez. 2.1.2), il contributo in spazio reciproco presenta una complessità proporzionale a  $N^{3/2}$ , con  $N$  numero di particelle interagenti. Gli approcci di Ewald approssimati, forniscono meccanismi alternativi per il calcolo del termine in spazio reciproco, consentendone una valutazione più rapida in condizioni di precisione controllata. SPME fa parte della categoria di metodi basati sull'introduzione di una griglia discreta di punti e l'uso della FFT per la stima delle somme e consente di ridurre il costo computazionale ad un ordine di  $\mathcal{O}(N \log N)$ .

### 3.6.1 Linguaggio

Nella struttura originale di M.DynaMix, era presente un modulo (insieme di file) contenenti le subroutine dedicate alla valutazione della parte in spazio reciproco della somma di Ewald. L'implementazione del metodo SPME è stata realizzata con la creazione di una nuova versione di questo modulo, da affiancare al primo già esistente. L'intercambiabilità dei due moduli, è stata assicurata facendo leva sulla modularità di cui M.DynaMix è dotato (sez. 1.2.1).

Per poter sfruttare al meglio le più avanzate tecniche di programmazione, si è scelto di usare il linguaggio C per la creazione di alcune delle nuove porzioni di programma. È noto che moduli di codice scritti in linguaggi diversi, quali ad esempio FORTRAN e C, possono facilmente coesistere nella stessa struttura, avendo l'accortezza di rispettare alcuni criteri non eccessivamente vincolanti [68]. Da parte sua, il linguaggio C possiede molti punti di forza rispetto al FORTRAN, collegati ad alcune sue caratteristiche intrinseche, quali l'allocazione dinamica della memoria, la manipolazione esplicita dei puntatori e la possibilità di dare luogo a strutture complesse di dati. Inoltre, l'uso del linguaggio C consente un più facile accesso ad un ampio serbatoio di librerie esterne; ciò è stato



particolarmente importante nel nostro caso, inquanto l'implementazione degli algoritmi ha richiesto l'utilizzo di routine di libreria per l'esecuzione delle FFT.

### 3.6.2 Algoritmo

Secondo quanto proposto nella sez. 2.3, l'applicazione dell'algoritmo SPME prevede l'esecuzione dei seguenti passi principali:

- Approssimazione delle cariche sulla griglia equispaziata (eq. (2.65)).
- Trasformazione delle cariche interpolate nello spazio reciproco (FFT inversa).
- Valutazione dell'energia (eq. (2.66)).
- Moltiplicazione della trasformata delle cariche per i fattori di rescaling  $B$  e  $C$  (eq. (2.67)) e ritorno in spazio reale (FFT diretta).
- Calcolo della forza su ciascun atomo (eq. (2.71)).

Nella implementazione realizzata in M.DynaMix, la creazione del set di B-splines utilizzato per l'interpolazione delle cariche, è realizzata in tempo reale attraverso un'opportuna subroutine. Il numero di funzioni da utilizzare è determinato dall'ordine dell'interpolazione, parametro specificato dall'utente tramite il file di input del programma.

Nel primo passo dell'algoritmo, le cariche  $q_i$ , collocate nelle posizioni  $\mathbf{r}_i$  della cella di simulazione, sono interpolate sulla griglia regolare di punti, attraverso l'opportuna combinazione di B-splines. I risultati, corrispondenti ai coefficienti della combinazione lineare, sono conservati nell'array FORTRAN unidimensionale  $Q$ . L'immagazzinamento è fatto in ordine consecutivo, secondo le tre direzioni dello spazio, salvando però le componenti in modo inverso,  $\{z, y, x\}$ , per ogni nodo della griglia discreta.

Al successivo passo dell'algoritmo, tramite un'opportuna funzione di libreria in linguaggio C, viene realizzata la trasformata tridimensionale delle cariche interpolate  $Q$ . La routine esterna accede all'array unidimensionale  $Q$  come se fosse tridimensionale. L'ordine inverso adottato per l'immagazzinamento delle componenti di  $Q$ , è connesso al differente modo, tra i due linguaggi, di collocare gli array multidimensionali in memoria: per colonne nel FORTRAN, per righe nel C. Le cariche trasformate sono conservate in un ulteriore array  $\hat{Q}^T$ ; anch'esse sono disposte in ordine inverso, inquanto prodotte da una routine scritta in linguaggio C.

Il calcolo dell'energia totale è realizzato a partire dai dati trasformati  $\hat{Q}^T$ , tenendo conto dell'ordine inverso con cui sono rappresentati.

L'array  $\hat{Q}^T$  è poi moltiplicato per gli opportuni coefficienti  $B$  e  $C$ , precedentemente calcolati e conservati in opportuni array unidimensionali. Una volta effettuato il prodotto, l'array  $\hat{Q}^T$  è sottoposto a FFT e utilizzato per valutare le forze.

### 3.6.3 Dettagli implementativi

Uno schema del modulo SPME, come implementato in M.DynaMix, è presentato in tabella 3.1; esso è costituito da un gruppo di file, parte scritti in linguaggio FORTRAN e parte in C; ogni file contiene un insieme di subroutine collegate all'esecuzione dei diversi passi dell'algoritmo. La tabella mostra, per ognuno dei passi dell'algoritmo, il linguaggio di programmazione usato per la realizzazione del corrispondente file. Alcuni file richiamano funzioni collocate in librerie esterne, collegate alla esecuzione della trasformata di Fourier veloce e della sua inversa.

Per l'esecuzione delle FFT, tra le diverse risorse disponibili, è stata selezionata la libreria FFTW [64]. La FFTW è una libreria largamente utilizzata, scritta in linguaggio C, contenente funzioni per l'esecuzione della trasformata discreta di Fourier in una o più dimensioni, per dati reali o complessi.

Step	Language	Lib Calls
1: Interpolazione delle cariche	F77	No
2: FFT Inversa	C	Yes
3: Valutazione dell'energia	F77	No
4: Rescaling	F77	No
5: FFT diretta	C	Yes
6: Calcolo delle forze	F77	No

Tabella 3.1: Schema di SPME: per ogni passo dell'algoritmo è indicato il linguaggio utilizzato per la scrittura della corrispondente routine e la presenza di chiamate a librerie esterne

### 3.6.4 Versione parallela

In M.DynaMix, l'algoritmo SPME è disponibile sia per la versione seriale che per quella parallela.

Per lo scambio dei dati e la sincronizzazione dei processi, l'implementazione parallela di SPME si basa sul protocollo di comunicazione MPI, su cui la versione originale del programma era già fondata. Il calcolo complessivo è stato partizionato dividendo il carico tra i processi, che sono in grado di operare sfruttando la conoscenza del proprio ID. Per la trasformazione da e verso lo spazio di Fourier, si è invece fatto uso di una versione parallela della FFT, disponibile nella libreria FFTW.

### 3.6.5 Parametri di input

Per usare SPME come metodo per il calcolo delle interazioni a lungo raggio, l'utente di M.DynaMix dovrà selezionare la corretta chiave nel file di input [69]; in particolare, oltre all'originale opzione `Ewald`, relativa al metodo inizialmente implementato nel programma, è ora presente un'opzione ulteriore, denominata `PME`, utilizzabile per selezionare il nuovo algoritmo.

Una volta optato per SPME, sulla stessa riga del file di input, l'utente ha la possibilità di specificare i parametri del calcolo, che influenzeranno il risultato prodotto, sia in termini di precisione che di costo computazionale.

I parametri specificabili dall'utente sono:

- Il fattore di convergenza di Ewald ( $\alpha$ ), che determina la ripartizione relativa tra i contributi in spazio reale e reciproco. Valori opportuni di  $\alpha$  sono collegati al cut-off in spazio reale ( $r_{cut}$ ) e possono essere fissati con il medesimo criterio valido per la somma di Ewald [43].
- l'ordine delle B-splines utilizzato nell'interpolazione della carica ( $p$ ), che condiziona l'accuratezza della rappresentazione; valori compresi tra 4 e 6 sono, di norma, sufficienti.
- il numero di punti della griglia FFT in ciascuna direzione dello spazio ( $n_x, n_y, n_z$ ), da cui dipende la precisione nella stima delle somme. La griglia discreta dovrebbe essere scelta in maniera dipendente dalla dimensione della cella di simulazione, in modo che il passo di campionamento non sia superiore a  $1.5\text{\AA}$  [35].

### 3.6.6 Testing

Il problema della correttezza di codici di grandi dimensioni è delicato e complesso. Per poter utilizzare un programma con profitto, riponendo fiducia nei risultati, occorre essere ragionevolmente certi che non siano presenti errori. In questo senso, è di primaria importanza condurre le procedure di test con il massimo scrupolo, cosa che può rivelarsi anche molto ardua. Nel caso dell'implementazione di SPME, si è fatto uso di un approccio molto efficace, noto come *cross-check testing* [71, 72].

Uno dei modi in cui tipicamente avviene l'evoluzione dei software scientifici, è la introduzione di nuovi algoritmi per calcolare, in modo più accurato o più efficiente, quantità che nel codice sono già valutate in altra maniera.

Se, nel corso di questa procedura, le funzionalità iniziali del programma non vengono intaccate, ci sarà una coesistenza di rami di esecuzione diversi, che portano al medesimo punto di arrivo. In questa condizione, è possibile realizzare un testing approfondito, utilizzando il ramo di esecuzione originale come termine di riferimento. Il procedimento può essere automatizzato consentendo, in modo naturale, la verifica capillare dei risultati in tutti i punti in cui i due rami di esecuzione si sovrappongono, cioè calcolando la medesima quantità intermedia o finale.

Nel caso di M.DynaMix, un metodo per il calcolo delle interazioni elettrostatiche a lungo raggio era già presente nella versione di partenza. Con la tecnica del cross-check, è stato possibile verificare in ogni sua parte il funzionamento del nuovo algoritmo ed ottenere una versione stabile e corretta del codice.

Dopo l'implementazione, lo stesso meccanismo di cross-check è stato usato per analizzare in modo sistematico la precisione e l'efficienza del programma, al variare dei parametri di input.

### 3.6.7 Benchmarking

L'algoritmo SPME è un metodo di calcolo delle interazioni Coulombiane ampiamente utilizzato in Dinamica Molecolare e le sue caratteristiche in termini di accuratezza ed efficienza sono ben note [35, 38]. L'analisi del comportamento di M.DynaMix, condotta comparando le prestazioni dell'algoritmo di Ewald con SPME, ha prodotto risultati in accordo con quanto atteso [35].

Il guadagno relativo del nuovo metodo, in termini di efficienza, è stato realizzato facendo riferimento ad un caso concreto connesso con la presente ricerca, quello del liquido

ionico *1-decil-3-metil-imidazolio cloruro* (DMIm). Le proprietà ed il comportamento dei liquidi ionici sono fortemente condizionati dalle interazioni elettrostatiche; la valutazione dell'efficienza realizzata tramite il DMIm, costituisce quindi un banco di prova attendibile sulle reali potenzialità della nuova implementazione di M.DynaMix. Le caratteristiche del DMIm, investigate anche grazie ai nuovi programmi sviluppati, sono l'oggetto del cap. 6.

La tabella 3.2 riassume le caratteristiche della stazione di calcolo utilizzata per il test ed il guadagno relativo di prestazioni, tra le simulazioni condotte con gli algoritmi SPME ed Ewald. I test fanno riferimento a due diverse numerosità del campione, pari rispettivamente a 125 molecole (5500 atomi) e 240 molecole (10560 atomi) di DMIm. Il guadagno è stato valutato considerando la lunghezza delle simulazioni ottenibili con i due algoritmi, a parità di tempo macchina impiegato. Si può osservare come il guadagno relativo di SPME rispetto ad Ewald sia funzione del numero  $N$  di particelle del sistema e tenda a diventare più marcato al crescere di  $N$ , passando da 4.6 volte per  $N = 5500$  a oltre 7 volte per  $N \sim 11000$ . Nel caso del sistema più grande, è ora possibile calcolare in meno di un giorno ciò che prima richiedeva una settimana di esecuzione.

Workstation	$N$	Speed-up
CPU Intel® Core™ i7 3930 K 3.20 GHz, 6-cores, 24 Gb Ram	5500	460%
	10560	740%

Tabella 3.2: Guadagno relativo nell'esecuzione di M.DynaMix con SPME, rispetto all'algoritmo di Ewald. La prima colonna riporta le caratteristiche della workstation utilizzata per il test; nelle successive, il guadagno relativo di prestazioni per due diverse numerosità del campione.

### 3.7 Implementazione del metodo ENUF

Come illustrato nella sez. 2.3, anche ENUF fa parte della categoria di metodi basati sull'uso di una griglia discreta di punti e della FFT. Il metodo fornisce un meccanismo alternativo per il calcolo del termine in spazio reciproco, permettendo una sua valutazione efficiente in condizioni di precisione controllata, con un costo computazionale di ordine  $\mathcal{O}(N \log N)$ .

### 3.7.1 Linguaggio

L'implementazione del metodo ENUF [67] è stata realizzata con la creazione di un'ulteriore versione del modulo di M.DynaMix, dedicato alla valutazione della parte in spazio reciproco della somma di Ewald. Il modulo ENUF, si aggiunge a quello originale e a SPME.

Anche in questo caso, per poter avere il massimo vantaggio dalle tecniche di programmazione più recenti, si è scelto di usare il linguaggio C per la creazione di alcune delle nuove porzioni di programma e per l'interfacciamento alle librerie esterne, necessarie per all'esecuzione della trasformata di Fourier su dati non uniformi.

### 3.7.2 Algoritmo

Riassumendo schematicamente quanto proposto nella sez. 2.2.7, l'algoritmo ENUF prevede l'esecuzione dei seguenti passi principali:

- Calcolo delle posizioni atomiche normalizzate (sez. 2.2.7).
- Esecuzione della NDFT *trasposta* per il calcolo dei coefficienti  $\hat{\mathbf{f}}_{\mathbf{n}}$  (eq. (2.45)).
- Valutazione dell'energia (eq. (2.48)).
- Scaling del fattore  $S(\mathbf{n})$  (eq. (2.52)).
- Esecuzione della NDFT *coniugata* e calcolo della forza (eq. (2.53)).

Nel primo passo dell'algoritmo, le coordinate sono normalizzate considerando le dimensioni della cella di simulazione, in maniera da ottenere l'opportuno insieme di nodi non equispaziati, richiesto nell'eq. (2.38). I nodi sono conservati nell'array FORTRAN unidimensionale  $X$ , in ordine consecutivo, secondo le tre direzioni dello spazio, salvando però le componenti in modo inverso,  $\{z, y, x\}$ , per ogni nodo della griglia discreta. Le cariche delle particelle sono invece collocate nell'array unidimensionale  $Q$ .

Al passo successivo, una funzione di libreria esterna, scritta in linguaggio C, realizza la trasformata tridimensionale *trasposta* delle cariche  $Q$ , collocate nei nodi  $X$ . La routine esterna, accede agli array unidimensionali  $Q$  ed  $X$  come se fossero tridimensionali. L'ordine inverso adottato per l'immagazzinamento delle componenti di  $Q$  è connesso, anche in questo caso, al differente modo di collocare gli array multidimensionali in memoria

da parte dei due linguaggi. I dati trasformati sono conservati in un ulteriore array di numeri complessi  $\tilde{S}$ .

Successivamente, l'energia è calcolata come somma di termini, utilizzando l'eq. (2.48). L'array  $\tilde{S}$  viene quindi riscaldato secondo l'eq. (2.51); a partire da  $\tilde{S}$ , sono calcolati tre diversi array complessi,  $\tilde{S}_{sx}$ ,  $\tilde{S}_{sy}$  e  $\tilde{S}_{sz}$ , uno per ogni direzione dello spazio reciproco.

Una funzione di libreria esterna, scritta in linguaggio C, trasforma  $\tilde{S}_{sx}$ ,  $\tilde{S}_{sy}$ ,  $\tilde{S}_{sz}$ , in relazione ai nodi  $X$ , tramite una trasformata tridimensionale *coniugata*. In questo passo dell'algoritmo, sono quindi eseguite tre trasformate di Fourier indipendenti, che danno luogo ai tre array complessi  $F_x$ ,  $F_y$ ,  $F_z$ . Il contributo alle forze su ogni particella, è ottenuto dall'eq. (2.53), attraverso le parti immaginarie delle tre trasformate.

### 3.7.3 Dettagli implementativi

Per l'esecuzione delle FFT su dati non equispaziati, l'attuale implementazione di M.DynaMix fa uso di una libreria esterna di funzioni. Tra le diverse risorse disponibili, è stata selezionata la libreria NFFT [61]. La NFFT è una libreria di ampia diffusione, scritta in linguaggio C, contenente funzioni per l'esecuzione della FFT non uniforme su dati discreti non equispaziati, sia reali che complessi. La NFFT si basa sull'impiego della FFTW.

Uno schema del modulo ENUF implementato in M.DynaMix è presentato in tabella 3.3; il modulo è costituito da un gruppo di file, parte scritti in linguaggio FORTRAN e parte in C; ogni file contiene un insieme di subroutine collegate all'esecuzione dei diversi passi dell'algoritmo. La tabella mostra, per ognuno dei passi dell'algoritmo, il linguaggio di programmazione usato per la realizzazione del corrispondente file. Alcuni file richiamano funzioni collocate in librerie esterne, collegate alla esecuzione di diversi tipi di trasformata di Fourier non uniforme.

### 3.7.4 Versione parallela

In M.DynaMix, l'algoritmo ENUF è al momento disponibile solo per la versione seriale del codice. La ragione è da ricercarsi nell'assenza di una adeguata libreria di funzioni per l'esecuzione parallela della trasformata di Fourier non uniforme. Mentre le usuali trasformate equispaziate rappresentano operazioni di routine e possono contare su meccanismi implementativi ormai consolidati, il campo delle trasformate non uniformi è relativa-

Step	Language	Lib Calls
1: Normalizzazione delle coordinate	F77	No
2: FFT <i>Trasposta</i>	C	Yes
3: Valutazione dell'energia	F77	No
4: Rescaling di $S(\mathbf{n})$	F77	No
5: FFT <i>Coniugata</i>	C	Yes
6: Calcolo della forza	F77	No

Tabella 3.3: Schema di ENUF: per ogni passo dell'algoritmo è indicato il linguaggio utilizzato per la scrittura della corrispondente routine e la presenza di chiamate a librerie esterne.

mente giovane e la produzione del software è ancora in una fase embrionale; il numero di risorse disponibili è scarso ed i pacchetti cui appoggiarsi sono spesso incompleti.

Si può ragionevolmente supporre che, nell'immediato futuro, si potrà disporre di una maggiore scelta e individuare più facilmente una risorsa software che consenta l'implementazione dell'algoritmo ENUF anche per l'esecuzione parallela.

### 3.7.5 Parametri di input

Per utilizzare ENUF come metodo di calcolo per le interazioni Coulombiane, l'utente di M.DynaMix dovrà selezionare la corretta chiave nel file di input [69]; in particolare, oltre alle opzioni precedentemente descritte, **Ewald** e **PME**, relative alla somma di Ewald e all'algoritmo SPME, è stata aggiunta l'opzione **ENUF**, utilizzabile per selezionare il nuovo algoritmo.

Una volta optato per ENUF, sulla stessa riga del file di input l'utente ha la possibilità di definire i parametri con cui effettuare il calcolo; essi influenzeranno il risultato, sia in termini di precisione che di costo computazionale.

I parametri del metodo, specificabili dall'utente, sono:

- Il fattore di convergenza di Ewald ( $\alpha$ ).
- Il numero di punti della griglia FFT in ciascuna direzione dello spazio ( $n_x, n_y, n_z$ ).

A partire dai parametri specificati dall'utente, il programma definisce automaticamente i valori ottimali del fattore di sovra-campionamento  $\sigma_s$  e del parametro di approssimazione  $p$ .



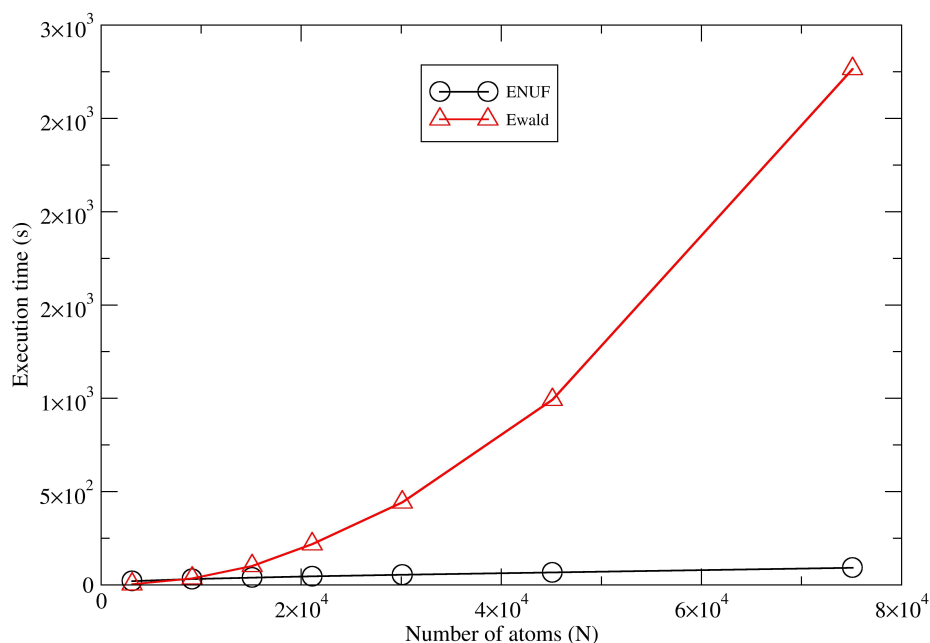


Figura 3.1: Tempo di esecuzione rispetto al numero di particelle  $N$ , per Ewald (rosso) ed ENUF (nero), per una simulazione di  $10^6$  iterazioni. Il sistema è composto da 50 ioni  $\text{Na}^+$  e 50 ioni  $\text{Cl}^-$  in soluzione in  $\text{H}_2\text{O}$  in una cella cubica. Il numero di molecole di  $\text{H}_2\text{O}$  è incrementato da  $10^3$  a  $25 \cdot 10^3$  in 7 passi discreti, mantenendo costante la densità di  $1.02 \text{ g/cm}^3$ .

### 3.7.6 Testing

Come nel caso di SPME, anche per l'implementazione dell'algoritmo ENUF è stato adottata la tecnica di validazione del cross-check. In particolare, ciò è avvenuto sia confrontato ENUF con il meccanismo di somma di Ewald che con l'approccio SPME. Ciò ha consentito anche di realizzare una analisi comparativa dei due algoritmi approssimati, mettendo in luce i rispettivi punti di forza e le corrispondenti debolezze, sia in termini di accuratezza che di efficienza.

### 3.7.7 Benchmarking

La figura 3.1 mostra il tempo di esecuzione nel calcolo del termine Coulombiano a lungo raggio, al variare del numero  $N$  di particelle del sistema, per i metodi Ewald e

ENUF [67]. Nella simulazione, di  $10^6$  iterazioni, il sistema è composto da 50 ioni  $Na^+$  e 50 ioni  $Cl^-$  in soluzione acquosa in una cella cubica. Il numero di molecole di solvente è incrementato da  $10^3$  a  $25 \cdot 10^3$  in 7 passi discreti, mantenendo costante la densità di  $1.02 g/cm^3$ .

La simulazione è stata effettuata su una workstation Intel Xeon, 1.86GHz, con 8 Gb di memoria RAM.

### 3.7.8 Pubblicazioni

L'implementazione del metodo ENUF descritta in questa sezione, è stata oggetto dell'articolo *Non-Uniform FFT and Its Applications in Particle Simulations*, pubblicato sulla rivista *Applied Mathematics*, incluso in appendice al presente documento.

## 3.8 TRIlly

Uno degli obiettivi di ricerca del lavoro descritto in questa tesi, è la messa a punto di una possibile tecnica di validazione, per le simulazioni di Dinamica Molecolare classica.

La tecnica di validazione considerata, che sarà descritta nei cap. 5 e 6, fa uso della del gradiente del campo elettrico (EFG), ottenuto come derivata delle componenti del campo elettrico, nelle tre direzioni dello spazio.

Il pacchetto di analisi delle traiettorie TRANAL non dispone di un meccanismo di calcolo di EFG.

### 3.8.1 Implementazione

Per superare le limitazioni di TRANAL e realizzare il calcolo del gradiente del campo elettrico a partire dai risultati delle dinamiche, è stato creato un codice a se stante, battezzato TRIlly, ottenuto dalla modifica e successiva unione di precedenti versioni di M.DynaMix e TRANAL.

TRIlly realizza il calcolo di EFG a partire dalla conoscenza del campo elettrico, valutato con la tecnica della somma di Ewald. Le derivate parziali sono determinate analiticamente e il programma produce il valore del campo tensoriale in corrispondenza delle posizioni occupate dai singoli nuclei atomici.

TRIlly è stato realizzato principalmente con l'obiettivo di elaborare i risultati delle

---

simulazioni prodotte da M.DynaMix, ma è in grado di accettare traiettorie scritte in altri formati, quali ad esempio quelli prodotti dal codice di simulazione AMBER [25].

## 4 Porting di M.DynaMix su GPU

Nell'ambito del calcolo scientifico, è molto sentita la necessità di disporre di piattaforme computazionali potenti ed efficienti, che consentano la soluzione rapida di problemi complessi. Ciò è vero anche nel campo della chimica computazionale ed in particolare nelle simulazioni dei sistemi biologici.

Con la prospettiva di migliorare le performance, è stata messa a punto una versione di M.DynaMix eseguibile su GPU. Ciò ha richiesto un adattamento degli algoritmi per la nuova tipologia di hardware e la riscrittura integrale delle procedure più onerose dal punto di vista computazionale.

La nuova versione del programma è provvista anche del supporto *multi-GPU*, che consente di sfruttare la presenza contemporanea più schede grafiche.

### 4.1 GPU e calcolo scientifico

Nella ricerca di prestazioni sempre più elevate, il settore del *High Performance Computing* (HPC) si è ormai stabilmente indirizzato verso la *parallelizzazione*, tecnica che consiste nella divisione di un problema complesso in un gran numero di sottoproblemi, la cui soluzione è effettuata in maniera concorrente.

La preferenza verso la parallelizzazione, è principalmente dovuta alle difficoltà che si incontrano quando si cerca di aumentare la potenza di un singolo processore, rispetto alla immediatezza di incrementare il numero di processori disponibili. Conseguentemente, le macchine contenenti una singola unità di elaborazione sono state quasi completamente soppiantate da sistemi multi processore, in grado di assicurare performance notevolmente più elevate.



Figura 4.1: Le GPU (Graphics Processor Units) sono gli acceleratori computazionali di maggiore diffusione.

### 4.1.1 Stato dell'arte

Le GPU (Graphics Processor Units) rappresentano la categoria di acceleratori computazionali più noti e di maggiore diffusione. Nate per l'elaborazione di immagini, le GPU hanno avuto, nel corso degli anni, un costante incremento delle prestazioni, trainato dal mercato dei videogiochi e dell'animazione tridimensionale cinematografica, settori in cui l'alta resa qualitativa e la possibilità di eseguire applicazioni in tempo reale, sono di importanza fondamentale.

Utilizzate per la visualizzazione delle immagini e presenti oramai in ogni computer, le GPU iniziarono, dalla fine degli anni 90, ad essere programmabili dall'utente in modo sempre più ampio. Sebbene l'elasticità fosse inizialmente ristretta alle sole applicazioni grafiche, si sviluppò ben presto la possibilità di impiegare tali unità per scopi diversi, ed in particolare come coprocessori dedicati all'esecuzione di calcoli di tipo generale (*General Purpose Computing*).

Le GPU sono sistemi efficienti e di basso costo, con cui incrementare le prestazioni computazionali e ridurre i tempi di esecuzione dei programmi. Nel settore della chimica computazionale, in particolare, si è assistito negli ultimi anni ad un grosso incremento di interesse, che si è concretizzato nell'adattamento di codici e nella messa a punto di algoritmi ottimizzati, per questa nuova tipologia di hardware (*porting*).

### 4.1.2 Strutturazione del mercato

Mentre il mercato delle piattaforme CPU (*Central Processing Units*) è variegato e presenta diversi produttori di hardware e software, nel contesto del GPGPU (*General Purpose computing on GPU*) vi è una grossa concentrazione delle risorse; è infatti presente un'azienda leader, la *NVidia Corporation*, che commercializza la quasi totalità dell'hardware e del software presente sul mercato. Nella sua posizione di soggetto principale, la NVidia ha fissato gli standard di riferimento, attraverso la creazione dell'architettura hardware per l'elaborazione parallela denominata CUDA (*Compute Unified Device Architecture*).

L'ambiente di sviluppo per CUDA annovera diversi linguaggi, quali FORTRAN, C, C++ e altri, che si presentano come *estensioni* dei corrispondenti linguaggi tradizionali e che fanno delle GPU architetture aperte, programmabili a piacimento dall'utente. La nomenclatura e le metodologie di programmazione proposte da NVidia, sono diventate riferimento per tutto il bacino d'utenza degli sviluppatori.

## 4.2 Architettura delle GPU

Le GPU sono circuiti integrati, inizialmente progettati con lo specifico obiettivo di processare dati con finalità grafiche. La loro architettura è profondamente diversa da quella di una usuale unità CPU e per questo richiedono linguaggi e tecniche di programmazione espressamente dedicate.

Le GPU non sono unità di calcolo autonome; per operare necessitano di un computer tradizionale, controllato da una CPU e provvisto degli usuali componenti di appoggio quali memoria RAM e hard disk. Per questo motivo, le GPU sono più propriamente definite *coprocessori*, svolgendo la loro funzione di acceleratori computazionali, in sinergia con l'unità principale.

Generalmente, la GPU è parte di una scheda (*Graphics Board*), in cui risulta alloggiata anche la memoria grafica globale e la logica di controllo; la connessione della scheda con la stazione su cui è collocata, avviene attraverso il *bus di sistema*, utilizzato, tra le altre cose, per il trasferimento di dati tra la memoria RAM della macchina a quella grafica.

Le GPU posseggono una architettura interna altamente parallela; esse sono costituite da un elevato numero di unità di elaborazione indipendenti (*core*), che può arrivare anche a diverse centinaia. Questo numero è molto più elevato di quanto normalmente disponi-

bile sulle usuali CPU che, al momento attuale, nella maggior parte dei casi è inferiore a 16. Ciascun core GPU è progettato per eseguire un gran numero di sottoprocessi contemporanei, detti *thread*, che a differenza delle CPU, realizzano operazioni molto semplici e sono concepiti per essere avviati a migliaia.

Il parallelismo messo in atto dalle GPU è allo stesso tempo un *parallelismo di thread* e un *parallelismo di dati*. L'architettura che prevede l'esecuzione della medesima istruzione su porzioni di dati differenti, viene indicata con l'acronimo SIMD (*Single Instruction Multiple Data*). Nel caso delle GPU, il SIMD si realizza partizionando un compito complesso in un grande numero di thread, che eseguono la medesima istruzione su dati diversi; tale approccio viene detto SIMT (*Single Instruction Multiple Thread*). La produttività della GPU è massima nel caso in cui i thread mantengano perfetto sincronismo di esecuzione.

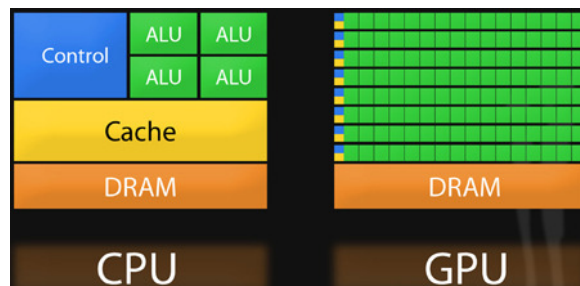


Figura 4.2: Le GPU posseggono una architettura interna altamente parallela; esse sono costituite da un elevato numero core indipendenti, anche diverse centinaia, molto maggiore di quanto disponibile sulle usuali CPU.

### 4.2.1 Memoria

Le GPU sono macchine sofisticate capaci di prestazioni notevoli, ma la scrittura di programmi efficienti per questo tipo di hardware non è immediata; se, nel progettare gli algoritmi, non si tiene conto delle caratteristiche intrinseche del coprocessore, il guadagno reale in termini di tempi di esecuzione può essere ben al di sotto delle aspettative. In questo senso, un ruolo fondamentale è giocato dalle differenze, tra GPU e macchine standard, nel modo in cui è organizzato e gestito l'accesso alla memoria.

Con il termine *latenza* di una memoria, si indica il ritardo tra l'inoltro della richiesta

di accesso ad un dato e l'istante in cui il dato è realmente disponibile. La memoria delle macchine CPU è usualmente organizzata in modo da minimizzare la latenza complessiva; a questo scopo, si fa uso di una ripartizione della memoria in livelli gerarchici (*cache*), con latenze (e purtroppo anche dimensioni) via via inferiori; gli algoritmi di controllo del sistema di cache, sono organizzati in modo tale che, in un dato frangente dell'esecuzione del programma, i dati che hanno alta probabilità di essere utilizzati, abbiano anche alta probabilità di essere collocati nella memoria con minore latenza.

Diversamente dalle CPU, le macchine GPU dispongono di varie tipologie di memoria, con caratteristiche e potenzialità diverse. Particolare importanza hanno la memoria globale e le memorie locali (*registri*, *memoria di thread*, *memoria di blocco*). Per ragioni connesse all'utilizzo principale, l'elaborazione di informazioni grafiche, l'ottimizzazione della memoria globale nelle GPU ha in generale lo scopo di massimizzare la mole di dati accessibile con una singola istruzione di lettura (*throughput*). Le memorie locali hanno velocità superiori alla memoria globale ma ampiezza molto limitata.

Tipicamente, la latenza della memoria principale in una macchina CPU è dell'ordine di 100 ~ 200 cicli di clock; questo limite è in parte *mascherato* dall'utilizzo degli opportuni meccanismi di caching. In una GPU, la latenza della memoria globale può facilmente superare i 1000 cicli di clock, ed è compensata dal caching solo marginalmente [103]. I lunghi tempi di attesa per la disponibilità dei dati, sono la principale causa di decremento di prestazioni, nell'esecuzione di programmi su macchine GPU.

Sintetizzando quanto detto, per creare programmi efficienti è in generale necessario progettare gli algoritmi tenendo conto delle peculiarità dell'hardware; in particolare per le GPU, l'obiettivo è minimizzare il numero di accessi alla memoria globale, rendendo massima la quantità di informazioni lette ad ogni accesso. Per mitigare gli effetti della latenza, è indispensabile fare più elaborazioni possibile, quindi eseguire quanti più thread possibile, sui dati già disponibili.

## 4.3 Struttura di un programma CUDA

Le GPU operano come coprocessori, svolgendo la loro funzione di acceleratori computazionali, in connessione all'unità principale. Nella terminologia corrente, il computer di appoggio su cui la GPU è alloggiata viene detto *host* mentre la scheda grafica stessa viene detta *device*.

Un programma CUDA ha una struttura ibrida: una parte del codice, relativo alla



CPU, è scritto con linguaggio e modalità standard; una seconda porzione, da indirizzare su GPU, è invece contenuta in specifiche procedure dette *kernel*. La parte del codice relativo all'host, ha il compito di avviare l'esecuzione dei kernel sul device; questa operazione prevede una inizializzazione preliminare dell'ambiente CUDA; successivamente, è possibile invocare l'esecuzione vera e propria dei kernel sul device, al termine della quale avverrà la deinizializzazione dell'ambiente. Nella fase di inizializzazione, l'host provvede ad allocare la memoria sul device e a operare il trasferimento dei dati dalla memoria host a quella del device. Successivamente all'esecuzione del kernel, i risultati del calcolo realizzato su GPU, vengono trasferiti dalla memoria del device a quella dell'host, avendo cura di liberare la prima.

Durante l'esecuzione dei kernel, la GPU procede in modo completamente autonomo; in questo frangente, la CPU è quindi libera di eseguire altri compiti in modo indipendente; le due unità di calcolo possono eventualmente essere sincronizzate, ricorrendo ad opportune istruzioni. Il programmatore dispone perciò di una ulteriore possibilità di parallelizzazione, potendo decidere quali porzioni dell'algoritmo indirizzare su ciascuna delle due unità di calcolo.

### 4.3.1 GPU e MD

Come discusso in precedenza, per progettare un efficiente algoritmo GPU, occorre tenere conto delle caratteristiche dell'hardware su cui andrà eseguito. Le linee guida generali cui il programmatore cercherà di attenersi saranno:

- scomporre la procedura complessiva in un gran numero thread elementari, il più possibile indipendenti;
- concepire i thread in modo che realizzino la medesima serie di istruzioni su porzioni diverse di dati, mantenendo sincronizzata l'esecuzione;
- limitare il numero di accessi in memoria, massimizzando la quantità di dati letta in ciascun accesso e il riutilizzo di dati già disponibili.

Il rispetto di queste direttive non dipende solo dalla corretta progettazione dell'algoritmo ma anche, in maniera fondamentale, dalla natura stessa del problema da risolvere. Alcune situazioni si prestano naturalmente ad essere descritte con approcci di tipologia

SIMD ed in esse l'algoritmo complessivo può facilmente essere partizionato in sottoprocessi elementari indipendenti. Altri problemi presenteranno difficoltà intrinseche e sarà virtualmente impossibile rappresentarli in forma puramente parallela.

Come l'esperienza di questi ultimi anni ha dimostrato, gli algoritmi tipici della Dinamica Molecolare non si prestano facilmente ad essere posti forma ottimale per l'esecuzione su GPU e per essi sarà indispensabile accettare compromessi o trovare delle soluzioni *ad-hoc* [26].

## 4.4 Porting

Come illustrato nella sez. 3.5, in una prima revisione di M.DynaMix, sono stati introdotti due efficienti algoritmi per il calcolo delle interazioni elettrostatiche a lungo raggio. Successivamente a tali modifiche, una volta in possesso di una versione stabile e verificata del programma, si è intrapresa la realizzazione della ulteriore serie di migliorie, che hanno condotto alla messa a punto di un codice eseguibile su GPU. In questa fase, sono state anche realizzate alcune ulteriori evoluzioni di minore entità.

Per via delle sostanziali differenze tra la struttura hardware delle GPU e quella dei processori standard, il porting di M.DynaMix su GPU ha richiesto cambiamenti profondi nell'architettura del programma. In particolare, è stato indispensabile modificare radicalmente un elevato numero di importanti strutture di dati. Si è perciò colta l'occasione per realizzare miglioramenti nell'organizzazione complessiva del codice, eliminando la maggior parte dei common blocks e introducendo l'allocazione dinamica della memoria, per molte tra le principali strutture dati.

Le modifiche realizzate sono state profonde e complesse. Oltre alle strutture dati necessarie per il funzionamento dei moduli eseguiti su CPU, è stato necessario introdurre opportune strutture aggiuntive, utilizzate per la comunicazione CPU $\leftrightarrow$ GPU e per l'esecuzione degli algoritmi sulla scheda grafica.

### 4.4.1 Approccio generale

La scelta metodologica fatta in fase di progetto, è stata quella di adattare all'esecuzione su GPU solo alcune routine del programma originale, mantenendo il più possibile invariata la restante porzione di codice. In particolare, si è optato per la conversione delle procedure più onerose dal punto di vista computazionale.

L'iniziale fase di profiling cui M.DynaMix è stato sottoposto, ha chiarito che il calcolo delle interazioni tra atomi non legati è responsabile del contributo di gran lunga più rilevante, al tempo totale di esecuzione del programma. Tale contributo può arrivare, a seconda del numero di atomi presenti nella cella, a superare il 95% della durata totale della simulazione.

Le procedure di maggior costo computazionale, principalmente responsabili della resa di M.DynaMix in termini di efficienza, sono risultate:

- creazione delle liste di Verlet (contributo  $\mathcal{O}(N^2)$ );
- calcolo delle forze tra atomi non legati, costituite da:
  - termine di Lennard-Jones;
  - interazioni Coulombiane a lungo raggio (Ewald, SPME, ENUF).

Si è proceduto quindi ad adattare all'esecuzione su GPU, due delle tre voci dell'elenco precedente, la creazione delle liste di Verlet e il termine di Lennard-Jones nella forza unbounded. Il calcolo delle interazioni Coulombiane a lungo raggio, nelle sue implementazioni con gli algoritmi Ewald, SPME ed ENUF, è stato invece mantenuto nella versione FORTRAN originale. Ciò è motivato dal fatto che, essendo l'esecuzione dei processi su GPU indipendente rispetto a quella su CPU, il termine Coulombiano può essere calcolato, su quest'ultima, parallelamente agli altri due, senza ulteriore peggioramento delle prestazioni. Inoltre, gli algoritmi approssimati SPME ed ENUF vengono eseguiti su CPU in un tempo dello stesso ordine di quello necessario alla GPU per completare i restanti due; in questo modo, il *carico* dei processi sulle unità di calcolo è ben distribuito, nel senso che nessuno dei due processori rimane a lungo inattivo in attesa che l'altro termini il suo compito.

#### 4.4.2 Common blocks e memoria dinamica

La versione di M.DynaMix usata come punto di partenza per la presente modifica, sebbene dotata di nuovi algoritmi per il calcolo delle interazioni Coulombiane, presentava la stessa organizzazione strutturale del codice iniziale, con ampio uso di common blocks e variabili dichiarate implicitamente; ciò costituiva un ostacolo alla messa a punto delle nuove funzionalità.

Moduli di un programma, che usano variabili collocate in blocchi comuni di memoria, non sono indipendenti; d'altro canto, procedure la cui esecuzione è indirizzata su coprocessori della CPU, devono necessariamente operare come entità autonome, capaci di comunicare con il contesto esterno, solo attraverso la lista degli argomenti. In considerazione di ciò, in una prima fase di revisione, è stata soppressa la maggior parte dei common blocks, sostituiti da un modello di comunicazione tra subroutines basato sulla lista degli argomenti. Inoltre, in questa fase, si è eliminato l'uso della dichiarazione implicita, preferendo richiedere al compilatore FORTRAN di controllare l'adeguatezza di ogni variabile utilizzata.

La soppressione dei common blocks, ha successivamente consentito di introdurre l'allocazione dinamica della memoria, per la quasi totalità degli array presenti nel codice. Per realizzare questa modifica, si è fatto uso della nota funzione di libreria `malloc` che, sebbene nativa del linguaggio C, con i più recenti compilatori è direttamente invocabile anche in porzioni di codice scritte in altri linguaggi.

L'allocazione dinamica è stata introdotta utilizzando *direttive condizionali di preprocessore*; ciò consente eventualmente di disabilitarla con una opzione di compilazione e tornare alla usuale allocazione statica, nel caso di compilatori che non consentono l'uso della `malloc`.

### 4.4.3 CUDA kernels

Per realizzare, su GPU, l'esecuzione delle parti di programma previste, sono stati creati opportuni CUDA kernel, scritti *ex novo*. Essi incapsulano le intere procedure di creazione delle liste di Verlet e di determinazione del termine di Lennard-Jones nella forza unbounded, sia a corto raggio che a lungo raggio (si veda la sez. 1.2.2).

L'utilizzo dei CUDA kernel in alternativa alle originali subroutine FORTRAN, può essere selezionato in fase di compilazione del programma, se la piattaforma di calcolo scelta per l'esecuzione possiede le idonee schede grafiche, attraverso l'utilizzo di una specifica opzione.

L'effettuazione delle modifiche discusse, ha richiesto l'introduzione di un elevato numero di nuove strutture dati, necessarie per la corretta comunicazione tra le diverse parti del programma e per l'esecuzione delle procedure indirizzate sulla scheda grafica.

Seguendo un approccio metodologico noto come tecnica del *wrapping*, l'interfacciamento tra la parte FORTRAN di M.DynaMix ed i CUDA kernel non è stato realizzato in

modo diretto ma attraverso moduli software intermedi, espressamente progettati come tramite tra le procedure.

#### 4.4.4 Linguaggio e interfacciamento

Come fatto in precedenza, anche in questo caso è stato usato il linguaggio C per la creazione delle nuove porzioni di codice. L'uso è più massivo rispetto alle evoluzioni descritte nelle sez. 3.6.1 e 3.7.1; in particolare, oltre ai i CUDA kernel, redatti completamente in C, sono scritti in questo linguaggio anche i moduli di wrapping.

L'utilizzo del wrapping è stato adottato per creare, dal punto di vista strutturale, un disaccoppiamento tra la parte FORTRAN del codice e le sezioni relative alla scheda grafica. Contando sulla facile possibilità di coesistenza di FORTRAN e C, si opera in modo che i moduli di interfacciamento, scritti in C, invochino i CUDA kernel e siano a loro volta invocati delle originali subroutine FORTRAN. Per incrementare il disaccoppiamento, la maggior parte delle nuove strutture dati, necessarie per il funzionamento del programma, è stata inserita nella parte C del codice.

Il mondo del GPGPU è giovane e la sua tecnologia è in rapida evoluzione; gli standard e le caratteristiche dell'hardware sono ancora in fase di assestamento e cambiamenti in questo senso sono probabili. L'indipendenza tra le parti FORTRAN e C del codice, potrà consentire di adattarsi alle nuove situazioni, circoscrivendo le modifiche a poche limitate sezioni e salvaguardando più possibile il nucleo centrale del programma, scritto in FORTRAN, racchiudente la maggior parte del *know-how* fisico chimico.

#### 4.4.5 Supporto multi-GPU

La versione di M.DynaMix eseguibile su GPU è stata sviluppata in modo da poter sfruttare la presenza contemporanea, sulla stessa stazione di calcolo, di più schede grafiche. Le strutture dati del programma sono progettate in modo da potersi adattare, in tempo reale, al numero di device utilizzati per la simulazione. Per amministrare le esecuzioni su più schede, il programma è stato dotato di un modulo avente la funzione di gestore dei processi, capace di dividere il carico di lavoro globale sui diversi coprocessori, avviare l'esecuzione dei kernel e riassemblare opportunamente i risultati.

Il numero di GPU utilizzate per l'esecuzione può essere specificato dall'utente con un'opportuna chiave nel file di input del programma.

### 4.4.6 Dettagli implementativi

La creazione delle liste di Verlet è uno dei moduli di M.DynaMix di cui è stato effettuato il porting su GPU. Il CUDA kernel realizzato con questo scopo, riceve in input le posizioni delle particelle e produce in output la lista di tutte le coppie interagenti (non legate) entro un dato cut-off  $r_{cut}$ .

#### Verlet non ridondante

Consideriamo gli atomi nella cella di simulazione numerati con un indice. La lista di Verlet del generico atomo  $i$  è l'elenco di tutti gli atomi  $j$  interagenti con  $i$ . Nelle versioni FORTRAN di M.DynaMix, tali elenchi sono calcolati e conservati in forma non ridondante. Ciò significa che, se la generica coppia ( $i \Leftrightarrow j$ ),  $i < j$  è presente nella lista di Verlet di  $i$ , viene omessa nella lista di  $j$ ; con questa tecnica, ciascuna coppia interagente viene annotata solo una volta; il numero di interazioni non replicate per particella ha quindi un andamento decrescente, come ad esempio quello della fig 4.3.

La rappresentazione in forma non ridondante, è una scelta efficiente se il calcolo delle liste di Verlet e delle forze unbounded è eseguito da macchine CPU. In questo caso infatti, le elaborazioni relative a particelle diverse possono essere svolte in modo indipendente; le forze di interazione, che costituiscono coppie azione-reazione, possono essere calcolate una sola volta, cosa che assicura la loro perfetta cancellazione entro la precisione floating-point della macchina; i calcoli sono svolti in modo efficiente, data la possibilità di massimizzare l'uso della cache [103].

#### Mappatura del problema

Nella creazione di un programma parallelo, uno degli oneri del programmatore è definire in che modo il problema debba essere ripartito sulle unità di calcolo; per le GPU, ciò significa indicare concretamente quale compito è assegnato a ciascun thread.

La scelta adottata per la versione GPU di M.DynaMix, è quella di associare a ciascun atomo del sistema un thread, cioè far sì che il calcolo della lista dei vicini e della forza unbounded del  $i$ -esimo atomo sia completamente realizzato dal  $t$ -esimo thread.

Il numero massimo di thread  $n_{th}$  generabili in una GPU è un parametro fisso<sup>1</sup>, dipen-

---

<sup>1</sup>Ciò è vero per le GPU per le quali M.DynaMix è stato sviluppato, con *compute capability*  $\leq 3.0$ . Compute capabilities successive, superano questa limitazione grazie all'adozione del *parallelismo dinamico*.

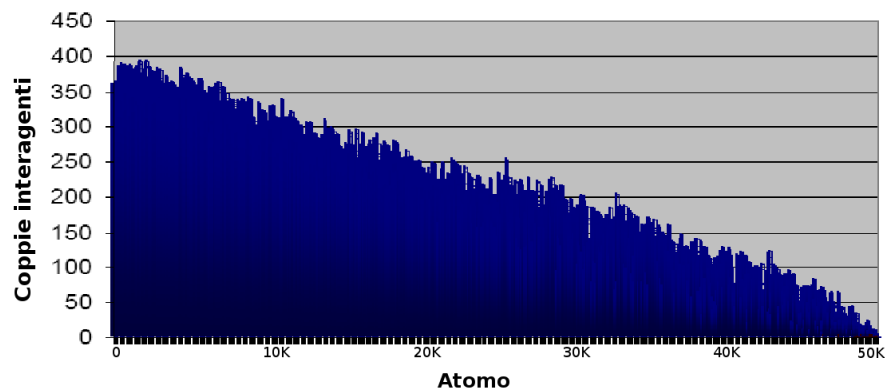


Figura 4.3: Numero di coppie interagenti per atomo, nel caso di liste di Verlet non ridondanti. La simulazione è relativa ad un sistema di  $50 \cdot 10^3$  particelle. Gli atomi indicati dal loro indice  $i$  con:  $0 \leq i < 50 \cdot 10^3$ . Le coppie sono annotate una sola volta quindi il loro numero decresce al crescere di  $i$ .

dente dalla tipologia e dalla versione del processore, dell'ordine di qualche migliaio. I sistemi oggetto di simulazioni MD, sono spesso costituiti da un numero  $N$  di atomi tale che  $n_{th} \ll N$ . Nella mappatura del problema su GPU, si è quindi optato per l'utilizzo di un'associazione *uno a molti* tra thread e particelle; con questo schema, ogni thread realizza il calcolo delle quantità relative a uno o più atomi del sistema.

### Ridondanza

Il piccolo numero di accessi alla memoria globale, gravata da elevata latenza, è la chiave per la creazione di efficienti programmi GPU. La forma non ridondante delle liste di Verlet pone problemi isormontabili da questo punto di vista, ed è stata in M.DynaMix abbandonata a favore della più adatta versione ridondante.

Consideriamo il thread  $t$ , associato alla particella  $i$ ;  $t$  ha l'onere di calcolare la forza complessiva sull'atomo  $i$ , tenendo conto di tutti gli atomi interagenti con esso e sommando i contributi calcolati (*accumulazione*). Per realizzare ciò,  $t$  scorre la lista di Verlet di  $i$  e, per ogni coppia, determina la corrispondente interazione. Il calcolo per la generica coppia ( $i \Leftrightarrow j$ ), nel caso di liste di Verlet non ridondanti, ha in realtà una duplice interpretazione e segue due percorsi di accumulazione differenti. La forza che  $j$  esercita su  $i$ ,  $F_{ji}$ , è ciò che  $t$  è deputato a calcolare; può essere accumulata nella memoria privata di  $t$ . La forza che  $i$  esercita su  $j$ ,  $F_{ij} = -F_{ji}$ , è invece al di fuori della giurisdizione di  $t$  e deve essere comunicata al thread associato a  $j$ , che è incaricato di gestirla. Nelle GPU,

i thread non hanno possibilità di comunicazione diretta; l'unica opzione in questo senso è usare la memoria globale, accessibile ad ogni thread, scrivendo o accumulando il valore in una specifica locazione. L'uso della lista non ridondante produce quindi un accesso *a pioggia* alla memoria globale, con un numero di indirizzamenti di  $\mathcal{O}(c^2)$  con  $c$  numero di coppie interagenti. L'algoritmo che scaturisce da questo approccio ha perciò efficienza molto bassa, con performance complessive peggiori di quelle ottenibili con una CPU.

Se, nella compilazione delle liste di Verlet, si annotano *tutte* le coppie senza omissioni, il numero di interazioni per particella è approssimativamente costante e presenta un andamento come quello esemplificato nella fig 4.4. In questa situazione, ciascun thread può calcolare la forza agente sulla particella assegnata, in modo completamente autonomo, conservando i risultati intermedi nelle veloci memorie locali, senza accedere alla memoria globale se non per annotare il risultato finale.

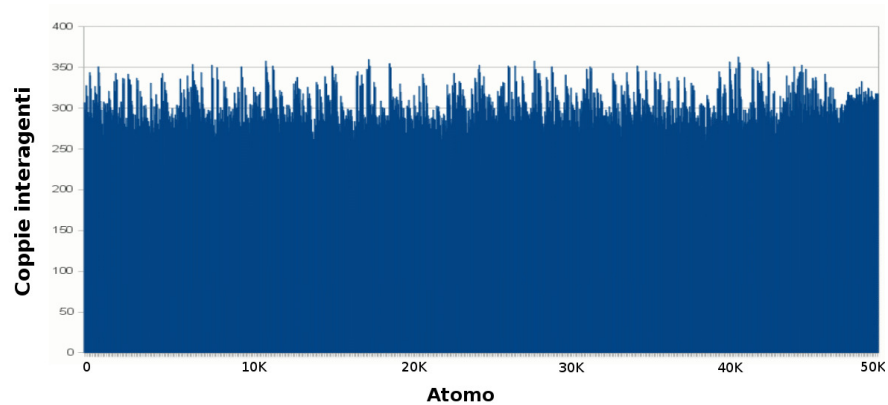


Figura 4.4: Numero di coppie interagenti per atomo, nel caso di liste di Verlet ridondanti. La simulazione è relativa ad un sistema di  $50 \cdot 10^3$  particelle. Gli atomi indicati dal loro indice  $i$  con:  $0 \leq i < 50 \cdot 10^3$ . Le coppie sono annotate due volte quindi il loro numero è approssimativamente costante al variare di  $i$ .

## 4.5 Parametri di input

L'esecuzione di M.DynaMix su scheda grafica può essere gestita dall'utente tramite la chiave GPU nel file di input [69]; in particolare, come ulteriori parametri opzionali, è possibile specificare quante e quali schede utilizzare per l'esecuzione del programma.



Le schede sono individuate secondo lo standard CUDA, con un intero  $s$ ,  $0 \leq s \leq S - 1$ , con  $S$  numero di schede presenti.

## 4.6 Testing

Come nei casi illustrati nella sez. 3.6.6 e 3.7.6, anche per la realizzazione della versione GPU di M.DynaMix, è stata adottata la tecnica di validazione del cross-check. In particolare, ciò è avvenuto assumendo come termine di confronto la versione CPU utilizzata come base di partenza per le modifiche (sez. 3.5.2). La tecnica del cross-check, ha consentito anche di realizzare un'analisi comparativa dei tempi di esecuzione, ottenendo una stima accurata dei guadagni di efficienza.

## 4.7 Benchmarking

Il guadagno in termini di efficienza della versione GPU di M.DynaMix, è stato valutato attraverso un gruppo di test, differenti nelle caratteristiche dei sistemi simulati e nella tipologia della scheda grafica utilizzata per l'esecuzione.

Una prima serie di risultati è stata ottenuta facendo riferimento ad un caso concreto connesso con la presente ricerca, quello del liquido ionico *1-decil-3-metil-imidazolio cloruro* (DMIm). Le caratteristiche del DMIm, investigate anche grazie ai nuovi programmi sviluppati, sono l'oggetto del cap. 6.

Workstation	GPU	$N$	vs SPME	vs Ewald
Intel® Core™ i7	NVidia GeForce	5500	220%	1010%
3.20 GHz, 24 Gb Ram	GTX 680	10560	300%	2220%

Tabella 4.1: Guadagno relativo nell'esecuzione di M.DynaMix GPU-SPME, rispetto alle versioni CPU con algoritmi SPME ed Ewald. Nelle prime due colonne sono riportate le caratteristiche della workstation e della scheda grafica impiegate per il test; nelle successive, il guadagno relativo di prestazioni con due diverse numerosità del campione, calcolato in riferimento alla versione CPU rispettivamente con algoritmi SPME ed Ewald. L'esecuzione del programma su CPU è realizzata sfruttando un solo core.

La tabella 4.1 riassume le caratteristiche della stazione di calcolo e della scheda grafica impiegate per il test ed il guadagno relativo di prestazioni, tra le simulazioni condotte con

la versione GPU e la versione CPU di M.DynaMix. Nella versione CPU, si sono valutate le interazioni elettrostatiche sia con l'algoritmo SPME che con l'algoritmo Ewald. I test fanno riferimento a due diverse numerosità del campione, pari rispettivamente a 125 molecole (5500 atomi) e 240 molecole (10560 atomi) di DMIm. Il guadagno è stato valutato considerando la lunghezza delle simulazioni ottenibili con le varie versioni, a parità di tempo macchina impiegato.

Si può osservare come il guadagno relativo della versione GPU rispetto ai due casi CPU, sia funzione del numero  $N$  di particelle e tenda a diventare più marcato al crescere di  $N$ . In particolare, con  $\sim 5000 \leq N \leq \sim 11000$ , l'accelerazione rispetto al caso CPU-SPME, appare compresa tra 2 e 3. Tenedo conto della velocizzazione introdotta dall'algoritmo SPME rispetto ad Ewald (tabella 3.2), il guadagno complessivo cresce molto più velocemente, assestandosi tra le 10 e le 20 volte.

Un ulteriore test di prestazioni è stato condotto considerando un sistema di  $NaCl$  in soluzione acquosa, costituito complessivamente da  $50 \cdot 10^3$  atomi. La tabella 4.2 riassume le caratteristiche dell'hardware impiegato ed il guadagno relativo di prestazioni. In questo caso, con un numero di atomi maggiore, il guadagno relativo della versione GPU rispetto alla CPU-SPME è pari a circa 9 volte.

Workstation	GPU			
Intel® Core™ i7 3.07 GHz, 16 Gb Ram	NVidia GeForce GTX 480	<b><math>N</math></b>	<b>vs SPME</b>	<b>vs Ewald</b>
		$50 \cdot 10^3$	920%	2500 %

Tabella 4.2: Guadagno relativo nell'esecuzione di M.DynaMix GPU, rispetto alle versioni CPU con algoritmi SPME ed Ewald, per un sistema di  $NaCl$  in soluzione acquosa, costituito complessivamente da  $50 \cdot 10^3$  atomi. L'esecuzione del programma su CPU è realizzata sfruttando un solo core.

## 5 Interazioni tra DNA e controioni

Il DNA è la molecola biologica per eccellenza, simbolo stesso della vita; la sua famosa *doppia elica* codifica l'informazione ereditaria, permettendo agli organismi di trasmettere le proprie caratteristiche alla discendenza e contiene le informazioni necessarie per la costruzione e il corretto funzionamento della cellula. Dal punto di vista chimico, esso si presenta come un polianione altamente carico, le cui proprietà strutturali e funzionali sono fortemente condizionate dalle interazioni elettrostatiche.

Il DNA è oggetto costante di intenso studio, sia teorico che sperimentale. Tra le metodiche comunemente utilizzate, hanno larga diffusione le tecniche di indagine computazionali. Vista la dimensione della molecola, che rende complessa l'applicazione di metodi *ab initio*, la Dinamica Molecolare è, tra tali tecniche, un approccio ampiamente utilizzato.

Nello studio del DNA, come di altre molecole, la MD permette di ottenere descrizioni al livello microscopico, che possono fornire un'utile chiave di lettura per la spiegazione di molte osservazioni sperimentali. Per contro, la MD è una tecnica nella quale le interazioni vengono parametrizzate e la natura stessa del modello gioca un ruolo fondamentale; inoltre, il confronto sistematico con l'esperimento è in molti casi difficoltoso.

In questo scenario, è naturale chiedersi in che misura le conclusioni ottenute dalle simulazioni siano affidabili o costituiscano un artefatto prodotto dalla particolare forma usata per la descrizione del sistema.

### 5.1 II DNA

Gli acidi nucleici sono polimeri di grandi dimensioni, utilizzati dagli esseri viventi per la conservazione e la trasmissione delle informazioni genetiche, necessarie allo sviluppo e al funzionamento degli organismi. Il DNA (acido desossiribonucleico) codifica l'informazione ereditaria, che viene trasmessa da una generazione alla successiva attraverso la

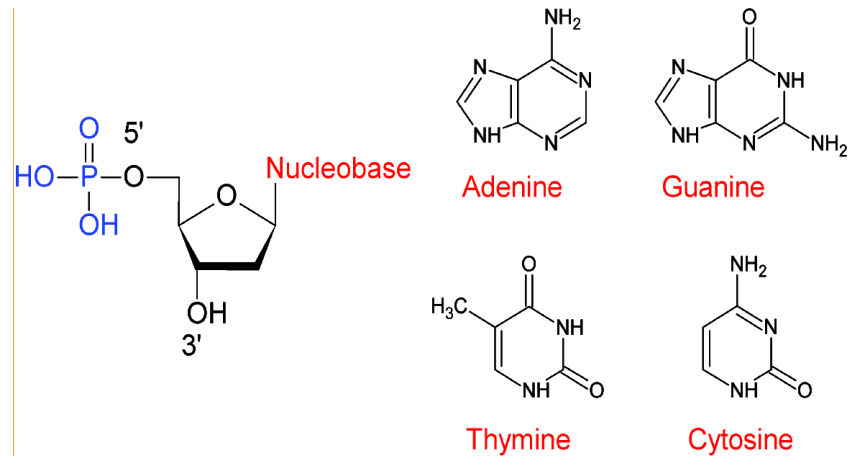


Figura 5.1: Le quattro basi azotate del DNA: adenina, guanina, citosina e timina.

sua *replicazione*. Il DNA conserva inoltre le istruzioni utilizzate per la sintesi di specifici componenti cellulari, quali le proteine, e dell'RNA (acido ribonucleico).

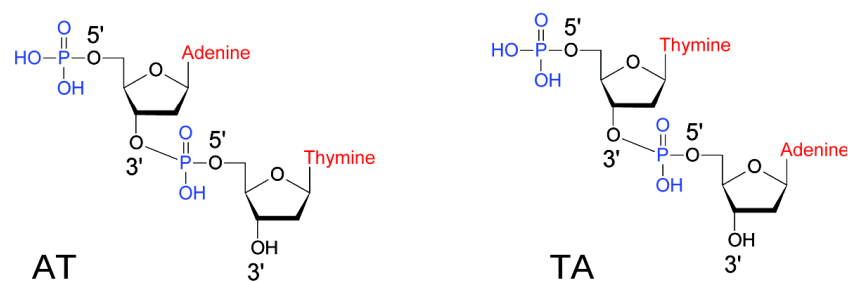


Figura 5.2: Legami nel DNA: il gruppo fosfato fa da ponte tra il carbonio 3' dello zucchero di un nucleotide e quello 5' dello zucchero del nucleotide successivo. Le catene vengono elencate a partire dagli elementi che si trovano all'estremità 5'; il filamento a sinistra viene quindi indicato come sequenza AT, mentre quello a destra come TA.

### 5.1.1 Unità costitutive

Le unità di base che compongono il DNA sono dette nucleotidi. Ogni nucleotide è formato da una base azotata, da un unità di desossiribosio e da un gruppo fosfato. Nel DNA, il fosfato di ciascun gruppo è legato allo zucchero di quello adiacente. Le basi azotate presenti nel DNA sono quattro: adenina (*A*), guanina (*G*), citosina (*C*) e timina

(*T*). Adenina e guanina sono molecole aromatiche eterocicliche chiamate purine; citosina e timina sono anelli pirimidinici.

La fig. 5.1 rappresenta le quattro basi nucleotidiche del DNA e i loro legami. La sequenza di nucleotidi che compone la catena viene elencata a partire dagli elementi che si trovano all'estremità 5'; un esempio è mostrato in fig. 5.2, dove il filamento a sinistra viene indicato come sequenza *AT*, mentre il filamento a destra come *TA*.

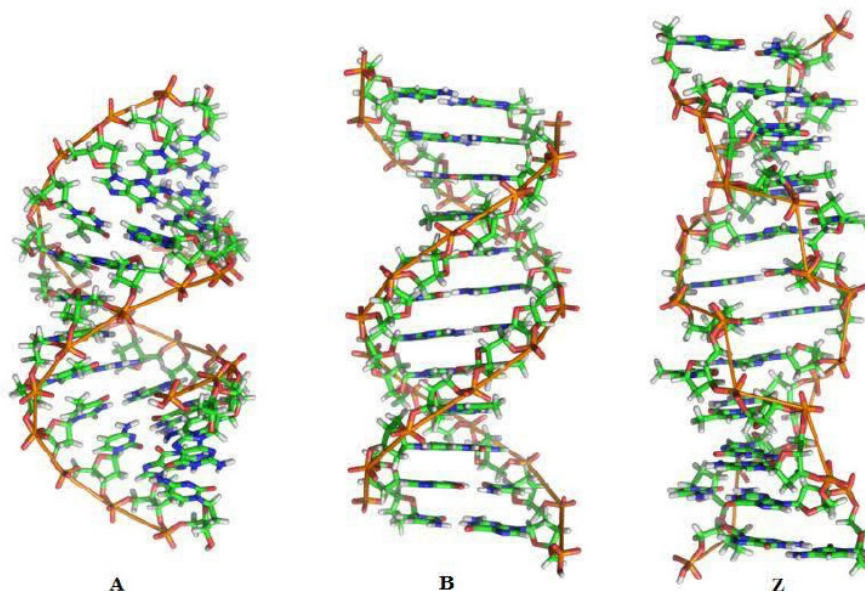


Figura 5.3: A-DNA, B-DNA e Z-DNA. Si differenziano tra loro per elementi strutturali quali la direzione di rotazione dell'elica, il numero di basi per passo, l'ampiezza del minor e del major groove.

### 5.1.2 Struttura

Le informazioni sperimentali a risoluzione atomica, ottenute da cristallizzazioni di piccole porzioni di molecola, mostrano come la struttura locale del DNA possa assumere varie forme, usualmente rappresentate con lettere maiuscole che vanno dalla A alla Z; le più comuni tra queste sono indicate come A-DNA, B-DNA e Z-DNA, schematizzate in fig 5.3; in condizioni fisiologiche, il DNA è quasi tutto nella forma B. Le diverse conformazioni si distinguono tra loro per elementi quali la direzione di rotazione dell'elica, il numero di basi per passo, l'ampiezza del minor e del major groove, l'inclinazione delle coppie di basi rispetto all'asse dell'elica e vari altri elementi strutturali [79].

La conformazione assunta dipende da diversi fattori, come la sequenza nucleotidica e grado d'idratazione e, come mostrato dagli esperimenti, è fortemente condizionata dall'intorno chimico [76, 77].

## Duplex

La forma del DNA che si ritiene più abbondante in vivo, il B-DNA, si presenta come costituito da una coppia di filamenti (*duplex*), uniti tra loro, a formare la famosa *doppia elica*. La doppia elica del B-DNA è la struttura biomolecolare più nota al pubblico non specializzato.

I due filamenti sono saldamente associati tra loro, attraverso legami a idrogeno che coinvolgono coppie di basi *complementari*; è noto infatti che l'adenina si lega solo con la timina e la citosina solo con la guanina.

La posizione reciproca con cui i filamenti sono avvolti, conduce alla formazione di due solchi, chiamati *major groove* e *minor groove*. Lo scheletro laterale, formato da un gruppo fosfato a cui è legato uno zucchero, che permette il legame tra nucleotidi adiacenti, viene detto *backbone*.

Mentre la coppia citosina-guanina dà luogo a tre legami a idrogeno, quella adenina-timina ne produce solo due. La stabilità della doppia elica è influenzata dalla natura della sequenza ed in particolare cresce con il contenuto di coppie *G – C* [91].

Nel B-DNA, i filamenti complementari hanno direzioni opposte e sono detti antiparalleli. Gli apici di un filamento sono individuate come estremità 5' ed estremità 3'.

Negli organismi eucarioti, la coppia di filamenti complementari costituenti il B-DNA, è contenuta per la maggior parte nel nucleo cellulare sotto forma di cromatina, una struttura filamentosa costituita in associazione con molecole proteiche di varia natura. Per la maggior parte della durata del ciclo cellulare, la cromatina mantiene lo stato di aggregazione filamentoso; quando il nucleo è in procinto di dividersi, essa tende a condensarsi e a formare un numero definito di strutture compatte, riconoscibili al microscopio ottico, chiamate cromosomi.

L'elica del B-DNA ha un diametro medio di  $20 \text{ \AA}$ ; per ogni passo dell'elica si contano approssimativamente 10 basi, con un angolo di torsione di  $36^\circ$  e una separazione spaziale tra due nucleotidi successivi di  $3.4 \text{ \AA}$ . Il backbone forma localmente un angolo quasi retto con il piano delle basi, che è perpendicolare all'asse dell'elica. Il numero totale di basi è stimato pari a 3.2 miliardi, per una lunghezza complessiva della molecola dell'ordine

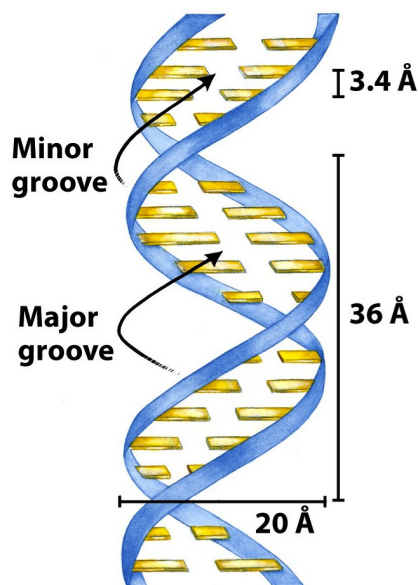


Figura 5.4: Struttura del B-DNA: la catena ha un diametro di 20 Å, con 10 basi per ogni passo dell'elica e una separazione spaziale tra due nucleotidi successivi di 3.4 Å.

di 1.5 m. Questi parametri, riassunti nella fig. 5.4, si riferiscono ad una conformazione media ideale, ottenuta dall'interpretazione dei dati di diffrazione a raggi X su fibre, assumendo l'esistenza di un solo motivo strutturale ripetuto identicamente per l'intera lunghezza dell'elica, indipendentemente dalla natura delle basi.

### Solvente e controioni

Il DNA è un polianione altamente carico e le sue proprietà strutturali e funzionali sono fortemente condizionate dalle interazioni elettrostatiche. In condizioni fisiologiche, la molecola si trova in soluzione acquosa, nella quale sono presenti ioni carichi positivamente (*controioni*), tra i quali  $K^+$ ,  $Na^+$ ,  $Mg^{2+}$  e  $Ca^{2+}$  sono i più abbondanti. La presenza dell'acqua e dei controioni riduce le forti repulsioni intramolecolari, consentendo al complesso di acquistare stabilità e svolgere le proprie funzioni biologiche.

Le interazioni tra DNA e controioni hanno un ruolo importante nel determinare la conformazione e le proprietà della molecola; il comportamento degli acidi nucleici non può essere compreso senza considerare l'atmosfera ionica circostante, al punto che l'acqua e i controioni sono considerati parte integrante della loro struttura [81, 80].

### 5.1.3 Dipendenza dalla sequenza

Studi teorici e sperimentali indicano come le interazioni tra DNA e controioni abbiano luogo in maniera dipendente dalla sequenza di basi [83, 82].

Ricerche basate su simulazioni di Dinamica Molecolare [85], hanno sollevato la discussione sul possibile ruolo strutturale delle interazioni tra cationi alcalini e DNA; gli studi hanno messo in evidenza come i controioni sodio interagiscano preferenzialmente nel minor groove con specifiche sequenze, coordinandosi con esse per periodi più lunghi [88]. In particolare, sono risultate avere importanza le sequenze denominate *A-tract* [87, 131], costituite da porzioni di DNA contenenti almeno quattro adenine o timine consecutive, senza tratti  $T - A$ <sup>1</sup>. È noto infatti che gli *A-tract* hanno un ruolo chiave nel modulare i processi di deformazione, in particolare di curvatura complessiva della catena e di restringimento dell'ampiezza del minor groove, del B-DNA [99]. In questo contesto, si è investigato sul fatto che le peculiarità strutturali degli *A-tract* potessero dipendere dalle caratteristiche delle loro interazioni con i controioni.

Dal punto di vista sperimentale, misure di elettroforesi e Risonanza Magnetica Nucleare (NMR) in soluzione [89, 90], hanno fornito importanti conferme sul fatto che alcuni ioni monovalenti prediligano coordinarsi con sequenze di basi ricche in *A-tract*. In generale tuttavia, per ioni quali  $K^+$  ed  $Na^+$  informazioni dettagliate al livello atomico sono problematiche da ottenere sperimentalmente; nella cristallografia a raggi X ad esempio, ioni leggeri con elevata mobilità, che fluttuano senza una precisa coordinazione, sono difficili da osservare; inoltre, essendo  $Na^+$  e  $H_2O$  isoelettronici, non sono facilmente distinguibili. Anche in soluzione, utilizzando tecniche NMR standard, non è possibile ottenere informazioni dettagliate [86].

## 5.2 MD e rilassamento quadrupolare

Il rilassamento NMR di nuclei con spin  $I > 1/2$ , avviene principalmente attraverso l'accoppiamento del momento elettrico di quadrupolo nucleare con il gradiente del campo elettrico generato dalla distribuzione di carica circostante. Per gli ioni quadrupolari in soluzione, come ad esempio  $^{23}Na$ , che ha  $I = 3/2$ , l'efficacia del rilassamento dipende dalla simmetria della distribuzione di carica e dalle sue fluttuazioni. La misura del

---

<sup>1</sup>Cioè, soltanto tratti  $A - A$ ,  $A - T$ ,  $T - T$ .



tempo di rilassamento dello spin nucleare può quindi fornire utili informazioni al livello atomistico sulla dinamica dei nuclei [92, 93, 94].

Per queste ragioni, il  $^{23}\text{Na}$ , controione fisiologico del B-DNA con abbondanza isotopica naturale del 100%, è ampiamente impiegato negli studi NMR di soluzioni acquose, per investigare le caratteristiche delle interazioni tra DNA e controioni. In particolare, questa tecnica è stata utilizzata per studiare la specificità del legame tra DNA e controioni e la sua dipendenza dalla sequenza nucleotidica [90].

Nonostante il vasto numero di studi sulle interazioni in soluzione acquosa tra  $^{23}\text{Na}$  e DNA, diversi aspetti microscopici della dinamica devono ancora essere chiariti del tutto. In particolare, poco è noto sull'influenza delle differenti caratteristiche del legame dello ione sodio sul rilassamento quadrupolare. Dato che le costanti di tempo che caratterizzano il decadimento della magnetizzazione del  $^{23}\text{Na}$  in tali contesti coprono una scala di tempi che va dalla decina di fs ai ns, l'uso delle simulazioni MD si appare un metodo di indagine potenzialmente utile per investigare questi fenomeni.

L'uso delle simulazioni MD ha in diversi casi permesso di avere un utile punto di vista microscopico sulla natura delle interazioni tra controioni fisiologici e DNA in soluzione, consentendo il calcolo di osservabili fisiche in buon accordo con l'esperimento [95, 100].

### 5.2.1 Teoria

Il ritorno all'equilibrio termico della magnetizzazione longitudinale e trasversa del  $^{23}\text{Na}$  è noto essere bi esponenziale [96]. Le quattro velocità di rilassamento sono connesse al gradiente del campo elettrico (EFG) nelle posizioni nucleari, attraverso la funzione di densità spettrale  $J(\omega)$ . In soluzione isotropica, esse sono date dalle relazioni

$$\begin{aligned}
 R_{1f} &= 2 \left( \frac{eQ}{\hbar} \right)^2 J_{00}(\omega_0) \\
 R_{1s} &= 2 \left( \frac{eQ}{\hbar} \right)^2 J_{00}(2\omega_0) \\
 R_{2f} &= \left( \frac{eQ}{\hbar} \right)^2 [J_{00}(0) + J_{00}(\omega_0)] \\
 R_{2s} &= \left( \frac{eQ}{\hbar} \right)^2 [J_{00}(\omega_0) + J_{00}(2\omega_0)]
 \end{aligned} \tag{5.1}$$

dove  $e$  è la carica dell'elettrone,  $\hbar$  la costante di Plank ridotta,  $\omega_0$  e  $Q$  sono la frequenza di Larmor e il momento di quadrupolo. La  $J_{00}(\omega)$  è proporzionale alla trasformata di

Fourier della componente principale del tensore EFG dipendente dal tempo,  $V(t)$ . In un sistema isotropico, essa può essere derivata anche dall'intero tensore EFG [97]:

$$J(\omega) = \frac{1}{60} \langle V(0) : V(0) \rangle \int_{-\infty}^{\infty} \cos(\omega t) G(t) dt \quad (5.2)$$

dove il simbolo  $\langle : \rangle$  indica il prodotto tra componenti omologhe e  $G(t)$  la funzione di autocorrelazione temporale normalizzata del tensore EFG

$$G(t) = \frac{\langle V(t) : V(0) \rangle}{\langle V(0) : V(0) \rangle} \quad (5.3)$$

È prassi raggruppare le costanti dell'eq. 5.1 e 5.2 in una singola espressione, la costante di accoppiamento quadrupolare  $\chi$  definita come

$$\chi^2 = \frac{2}{3} \left( \frac{eQ}{h} \right)^2 \langle V(0) : V(0) \rangle \quad (5.4)$$

Per l'interpretazione di dati sperimentali ottenuti da macromolecole in soluzione, usualmente si assume che la funzione di autocorrelazione temporale  $G(t)$  decada come somma di funzioni esponenziali, relative a scale di tempo distinte

$$G(t) = \sum_{i=1}^N a_i e^{-t/\tau_i} \quad (5.5)$$

Se i tempi di correlazione  $\tau_i$  caratterizzanti  $G(t)$  sono piccoli comparati con la frequenza di Larmor, cioè se  $\omega_0 \tau \ll 1$ , tutte le velocità di rilassamento nell'eq. 5.1 diventano uguali e pari a

$$R = \frac{2}{5} \pi^2 \chi^2 \int_0^{\infty} G(t) dt \quad (5.6)$$

La condizione appena descritta è indicata come *extreme narrowing regime*.

Assumendo invece che il decadimento di  $G(t)$  sia descritto da un solo esponenziale e perciò caratterizzato da un solo tempo di correlazione, la velocità di rilassamento può essere espressa come

$$R = \frac{3\pi^2}{50} \chi^2 \frac{2I + 3}{I^2(2I - 1)} \left[ \frac{\tau_c}{1 + \omega^2 \tau_c^2} + \frac{4\tau_c}{1 + 4\omega^2 \tau_c^2} \right] \quad (5.7)$$

Dal punto di vista computazionale, quando il gradiente del campo elettrico è valutato con metodi non quantistici, come nel caso delle simulazioni MD, gli effetti della nube elettronica che circonda i nuclei sono presi in considerazione con l'introduzione di un coefficiente di polarizzazione detto *fattore di Sternheimer*,  $(1 + \gamma_\infty)$ , che agisce come amplificatore del gradiente EFG generato esternamente,  $V_{ext}$ :

$$V = (1 + \gamma_\infty)V_{ext} \quad (5.8)$$

Il confronto dei risultati mostra che, per ioni monovalenti quali  $^{23}\text{Na}$  o di massa maggiore, l'utilizzo della *approssimazione di Sternheimer* permette di ottenere risultati qualitativamente in accordo con approcci da primi principi [95, 97], anche se, dal punto di vista quantitativo, sono stati recentemente messi in evidenza alcuni limiti[101].

## 5.3 La ricerca

Le tecniche computazionali, in particolare la Dinamica Molecolare, sono state ampiamente utilizzate per gli studi, al livello atomistico, delle interazioni tra DNA, acqua e controioni; esse hanno fornito un'utile chiave di lettura per la spiegazione di molte osservazioni sperimentali, o permesso di ottenere informazioni non accessibili empiricamente [98, 90].

Dal punto di vista sperimentale, non è facile ottenere descrizioni al livello atomistico, del comportamento di controioni monovalenti quali  $\text{Na}^+$  e  $\text{K}^+$  e spesso i dati raccolti lasciano spazio a diverse interpretazioni. Informazioni molto dettagliate sulle interazioni, possono essere ottenute dalle simulazioni di Dinamica Molecolare.

Occorre ricordare tuttavia che la MD è una tecnica nella quale le interazioni elementari vengono modellizzate; è naturale chiedersi se i parametri utilizzati, o la natura stessa del modello, possa influire significativamente. L'impossibilità di un confronto sistematico con l'esperimento, lascia in molti casi aperto un ampio ventaglio di opzioni.

Sapere fino a che punto le informazioni ottenute siano attendibili, o frutto delle parametrizzazioni utilizzate, deve ancora essere ben compreso.

### 5.3.1 Motivazioni

La fase della ricerca descritta nel presente capitolo corre su due binari paralleli strettamente correlati.

Da un lato, ci si propone di investigare le modalità con cui le interazioni tra il DNA in soluzione ed i relativi controioni fisiologici, sono connesse alla sequenza di basi nucleotidiche. Dall'altro, occorre capire in che misura i risultati ottenuti dalle simulazioni siano affidabili o costituiscano un artefatto prodotto dalla particolare descrizione del sistema. In questo senso, il problema generale della validazione delle simulazioni MD, riveste una grande importanza materiale.

Con in mente questi obiettivi, è stata simulata la dinamica di tre differenti oligomeri di DNA, contenenti sequenze di A-tract di lunghezza diversa. La simulazione della dinamica delle suddette sequenze è stata fatta in soluzione acquosa diluita; la carica in eccesso è stata compensata dall'introduzione di un opportuno numero di atomi di controione fisiologico  $Na^+$ . Sequenze di DNA e condizioni di simulazione sono stati scelti in modo corrispondente a quanto fatto sperimentalmente nello studio [90].

Il calcolo del gradiente del campo elettrico, realizzato attraverso il programma TRILLY, ha consentito la stima dei parametri del decadimento quadrupolare dei controioni  $Na^+$ ; il successivo confronto con i dati sperimentali disponibili in letteratura, ha permesso di verificare il grado di affidabilità delle conclusioni ottenute dalle diverse, articolate, condizioni di simulazione.

### 5.3.2 Sequenze nucleotidiche analizzate

L'analisi realizzata ha avuto come oggetto tre dodecameri aventi diversa sequenza nucleotidica:

- l'oligomero di sequenza *CGAAAATTTTCG* (762 atomi) denominato *A4T4*: è caratterizzato da un A-tract composto da otto coppie di basi.
- L'oligomero di sequenza *CGCGAATTTCGCG* (758 atomi) denominato *DDD*: è caratterizzato da un A-tract composto da quattro coppie di basi.
- L'oligomero di sequenza *CGCTCTAGAGCG* (758 atomi) denominato *TA*: è com-

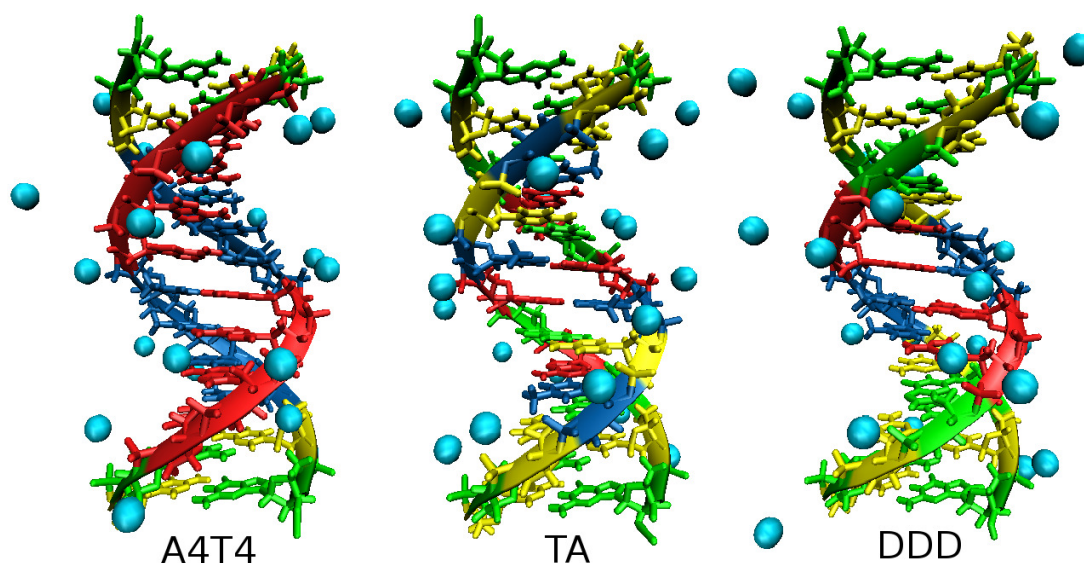


Figura 5.5: I tre dodecameri oggetto delle simulazioni: A4T4 (CGAAAATTTTCG), TA (CGCTCTAGAGCG) e DDD (CGCGAATTCGCG); i colori evidenziano le basi delle sequenze. Rosso: adenina, blu: timina, verde: guanina, giallo: citosina.

posto dalle stesse basi del DDD, disposte con diversa sequenza. Non possiede A-tract.

### 5.3.3 Set di parametri

La dinamica di ciascun oligomero, è stata simulata ricorrendo a tre diversi set di parametri, che si distinguono sia per la descrizione dello ione  $Na^+$  che per il modello utilizzato per il solvente.

La tabella 5.1 schematizza le caratteristiche principali dei set utilizzati, indicati con le single (a), (b) e (c), riportando, nell'ultima colonna, il tempo totale di simulazione nei diversi casi.

Il set (a) fa uso dei parametri di Smith e Dang [75] per la simulazione dello ione  $Na^+$  e del modello Flexible SPC (Simple Point Charge) di Toukan e Rahman per il solvente [10, 12]. I parametri di Smith e Dang sono utilizzati molto comunemente nella simulazione del sodio in soluzione acquosa. Flexible SPC è un modello a tre siti non polarizzabile con legami flessibili; la flessibilità del legame  $O-H$ , che compensa in parte la non polarizzabilità, gli conferisce elevata accuratezza [14]. L'oligomero, come anche

negli altri due set, è descritto da un Force Field ParmBSC0, utilizzato tipicamente per la modellazione di acidi nucleici [11].

Il set (b) utilizza i parametri di Åqvist (adattamento Amber) per lo ione  $Na^+$  che, secondo una opinione diffusa, sono stati i parametri più utilizzati nella simulazione di acidi nucleici per la descrizione di cationi monovalenti e divalenti [13]. Il modello del solvente è TIP3P, provvisto di 3 siti di interazione e di un potenziale contenente termini elettrostatici e contributi della forma  $6 - 12$  [9].

Il set (c) si differenzia dal (b) solo nell'utilizzo dei parametri di Joung e Cheatham per la descrizione dello ione [102].

Set	$Na^+$ Param.	$H_2O$ Param.	DNA FF	Time
(a)	Smith-Dang	Flex SPC	Amber Parmbsc0	400 ns
(b)	Åqvist (Amber adapt.)	TIP3P	Amber Parmbsc0	120 ns
(c)	Joung-Cheatham	TIP3P	Amber Parmbsc0	500 ns

Tabella 5.1: Set di parametri utilizzati e durata totale delle simulazioni.

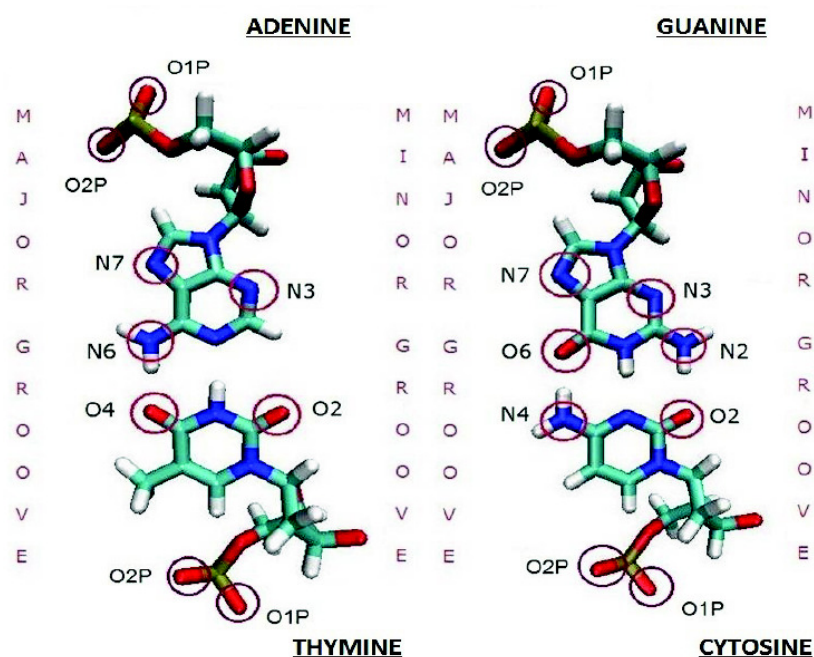


Figura 5.6: Le basi dei tre oligomeri rispetto alle quali sono state calcolate le RDF riportate in fig 5.7.

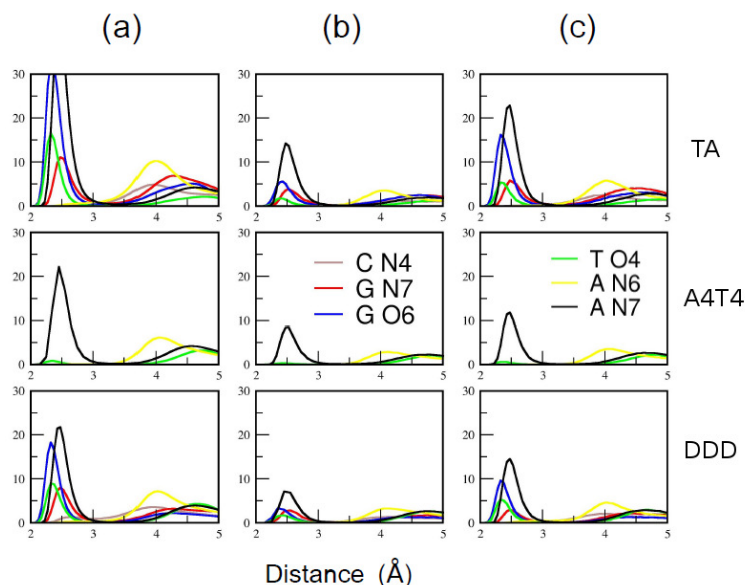


Figura 5.7: RDF degli ioni  $^{23}\text{Na}^+$  nel major groove, per i tre oligomeri (in orizzontale) e per i tre set di parametri considerati (in verticale). I siti di riferimento rispetto a cui le RDF sono calcolate sono indicati con colori differenti, il cui significato è riportato nella legenda.

### 5.3.4 Protocollo di simulazione

Sulla base dei dati sperimentali disponibili, la struttura iniziale scelta per i tre oligomeri è stata di tipo B-DNA. Collocati in una cella di simulazione ottaedrica, contenente 4500 molecole di  $\text{H}_2\text{O}$ , i sistemi sono stati neutralizzati con l'aggiunta di 22 controioni  $^{23}\text{Na}^+$ , inizialmente posizionati nei punti a minore potenziale elettrostatico. La dinamica si è svolta in condizioni isoterme e isobariche, con  $T = 300\text{ K}$  e  $P = 1\text{ bar}$ . Precedentemente all'esecuzione della dinamica vera e propria, il sistema è stato sottoposto ad una fase di *equilibratura*, necessaria per condurlo dalla configurazione iniziale ad una configurazione di equilibrio, analoga a quella descritta in [86].

Le dinamiche relative al set di parametri (a) sono state realizzate con il programma M.DynaMix; quelle dei set (b) e (c) con il programma Amber [25]. La scelta di M.DynaMix è legata principalmente alla presenza, nel programma, del meccanismo del doppio time step (si veda la sez. 1.2.2), tecnica ottimale per la produzione delle dinamiche nel caso di modelli dell'acqua flessibili, quali SPC adottato nel set (a).

### 5.3.5 Interpretazione di RDF e RT

Dalle simulazioni prodotte nelle diverse condizioni, è stato ricavato un primo insieme di informazioni, sotto forma di funzioni di distribuzione radiale (RDF) e tempi di residenza (RT) rispetto ad alcuni specifici atomi dei tre oligomeri.

La fig. 5.7, mostra l'andamento delle RDF dei controioni  $Na^+$ , per i tre oligomeri (in orizzontale) e per i tre set di parametri utilizzati (in verticale). Le RDF sono calcolate rispetto agli atomi del major groove indicati nella fig. 5.6 e riportate nei grafici con colori diversi.

La fig. 5.8 ha lo stesso significato generale, ma è relativa ad atomi degli oligomeri collocati nel backbone.

Le fig. 5.9 e 5.10 sono invece relative al comportamento medio, sull'insieme di ioni, dei tempi di residenza, calcolati in riferimento agli stessi atomi per i quali sono state calcolate le RDF, rispettivamente collocati nel major groove e nel backbone.

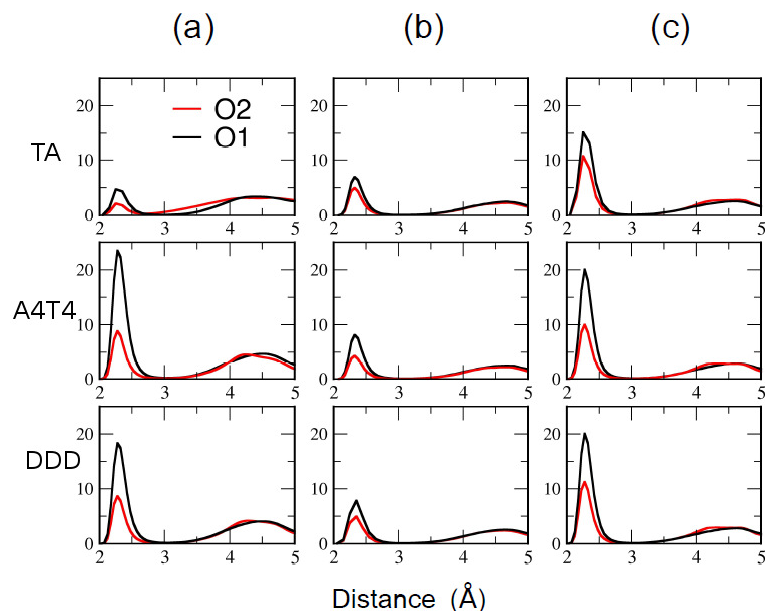


Figura 5.8: RDF degli ioni  $^{23}Na^+$ , rispetto agli ossigeni O1 e O2 del backbone, per i tre oligomeri (in orizzontale) e per i tre set di parametri considerati (in verticale).



### Schemi di lettura

Vista la presenza di diverse condizioni di calcolo, le RDF e i RT proposti nei diversi casi, possono essere considerati:

- secondo una **lettura verticale**: a parità di set di parametri, ricercando comportamenti differenti tra i tre oligomeri considerati, cioè dipendenti dalla sequenza di basi;
- secondo una **lettura orizzontale**: a parità di oligomero, evidenziando proprietà collegate ai set di parametri utilizzati per realizzare le simulazioni;
- tramite una **lettura incrociata**: combinando le due precedenti letture, per comprendere se i diversi set di parametri, concordano nelle previsioni generali relative al comportamento dei vari oligomeri.

La comparazione dei grafici, inoltre, sarà volta a mettere in evidenza:

- **differenze quantitative**: la forma generale delle curve è la medesima ma i valori assoluti, ad esempio l'altezza dei picchi, differiscono; ciò indica medesima *tipologia* ma diversa *intensità* delle interazioni stimate.
- **differenze qualitative**: la forma generale delle curve non coincide, sintomo di una diversa *tipologia* nello schema delle interazioni.

### Interpretazione

Ad una prima analisi generale, le RDF di fig. 5.7, e 5.8 presentano importanti differenze quantitative, sia secondo una lettura orizzontale che verticale. Ciò mette in luce un primo risultato complessivo delle simulazioni prodotte, la forte dipendenza delle caratteristiche del legame diretto  $Na^+$  - DNA dalla sequenza nucleotidica; tale dipendenza, presente in tutte le condizioni di simulazione, riguarda sia il major groove che il backbone, anche se nel set (b) di quest'ultimo appare in maniera decisamente più ridotta.

Per quanto riguarda i RT, in generale essi evidenziano la presenza di interazioni di tipo simile nelle due catene di ciascun oligomero; considerando la palindromicità dei duplex, ciò è indice della quasi raggiunta convergenza nella riproduzione delle interazioni, nel tempo di simulazione considerato,

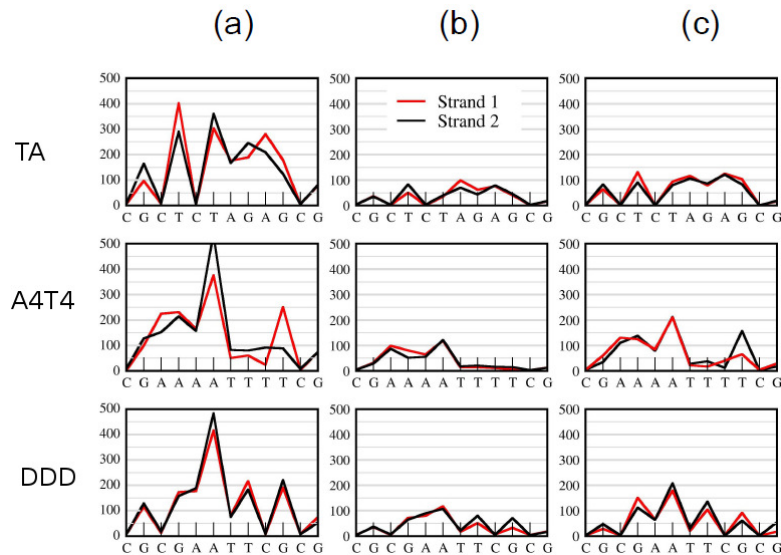


Figura 5.9: RT degli ioni  $^{23}\text{Na}^+$  nel major groove, rispetto alle basi della sequenza (in ascisse), per i tre oligomeri (in orizzontale) e per i tre set di parametri considerati (in verticale).

Più nello specifico, la lettura orizzontale delle RDF e dei RT, evidenzia proprietà del legame  $\text{Na}^+$  - DNA su cui le previsioni dei diversi set di parametri sono concordanti ed aspetti su cui sono discordanti. In particolare, i tre modelli sono in linea nel predire:

- che il contatto diretto  $\text{Na}^+$  - DNA sia meno probabile nel major groove degli oligomeri contenenti A-tract e che ad A-tract più lunghi corrisponda minore probabilità di trovare lo ione legato ad A o T.
- Che, indipendentemente dalla sequenza, il sito N7 dell'adenina sia il favorito per le interazioni di  $\text{Na}^+$  nel major groove.
- Che il tempo di residenza nel major groove sia fortemente dipendente dalla sequenza cui il nucleotide appartiene; in particolare, nella parte centrale degli A-tract, i RT appaiono essere più lunghi per l'adenina e più brevi per la timina, che a sua volta ha un tempo di residenza maggiore in regioni prive di A-tract. Per la guanina e specialmente per la citosina, i RT hanno basso valore in tutti i casi.

Viceversa, i tre set di parametri non concordano nel predire proprietà quali:

- probabilità di legame. Nel major groove le RDF hanno lo stesso andamento qualitativo al variare dei parametri, mentre nel backbone ciò non è vero; infatti, la

marcata dipendenza dalla sequenza del legame del sodio nel backbone, mostrata con il set di parametri (a), non è riprodotta dai set (b) e (c).

- Tempi di residenza, che variano con il modello utilizzato non soltanto quantitativamente ma anche in modo qualitativo.

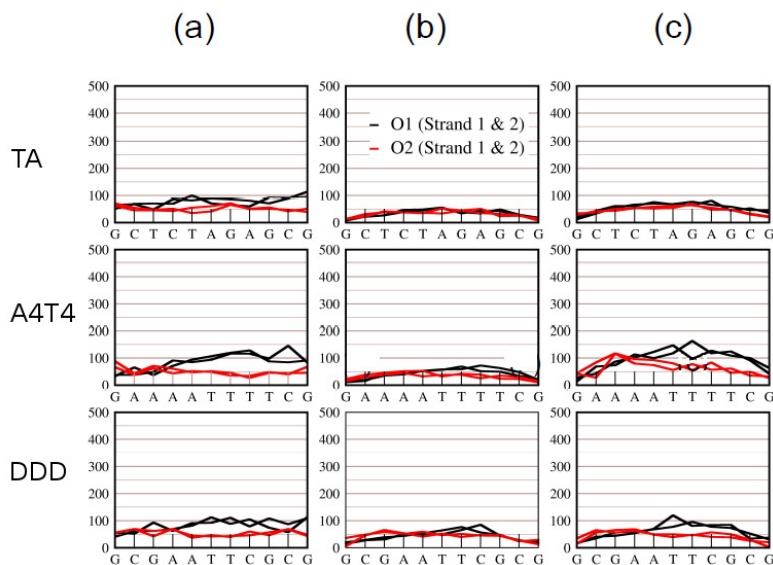


Figura 5.10:  $RT$  degli ioni  $^{23}\text{Na}^+$  nel backbone, rispetto alle basi della sequenza (in ascisse), per i tre oligomeri (in orizzontale) e per i tre set di parametri considerati (in verticale).

### 5.3.6 EFG e funzione di autocorrelazione

A partire dalle traiettorie atomiche generate dalle dinamiche, tramite il programma TRILly, si è realizzato il calcolo del gradiente del campo elettrico  $V(t)$  nelle posizioni occupate dai controioni  $\text{Na}^+$ . Dalla conoscenza di  $V(t)$ , tramite una estensione del programma di analisi TRANAL, si è determinata la funzione di autocorrelazione temporale  $G(t)$ , data dalla eq. (5.3). I grafici delle funzioni di autocorrelazione  $G(t)$ , per i diversi set di parametri (a), (b) e (c) e per i diversi oligomeri, sono riportati in fig. 5.11.

Come detto in precedenza, le sequenze di DNA e le condizioni di simulazione della presente ricerca, sono stati scelti in modo corrispondente a quanto fatto sperimentalmente nello studio [90]; lo stesso studio, fornisce la base per il confronto qualitativo diretto, tra gli andamenti delle funzioni di correlazione temporale  $G(t)$  e l'esperimento. La misura delle velocità di rilassamento  $R$  in condizioni di extreme narrowing per i

tre oligomeri (eq. (5.6)), evidenzia valori maggiori per sequenze con A-tract più lunghi,  $R_{AAT4} > R_{DDD} > R_{TA}$ . Dello stesso tipo è anche l'andamento atteso delle  $G(t)$ , visto il legame matematico tra le due quantità (eq. (5.3)):  $G_{AAT4} > G_{DDD} > G_{TA}$ . Dei valori prodotti con i tre set dei parametri, soltanto le simulazioni realizzate con il set (c) riproducono il comportamento atteso.

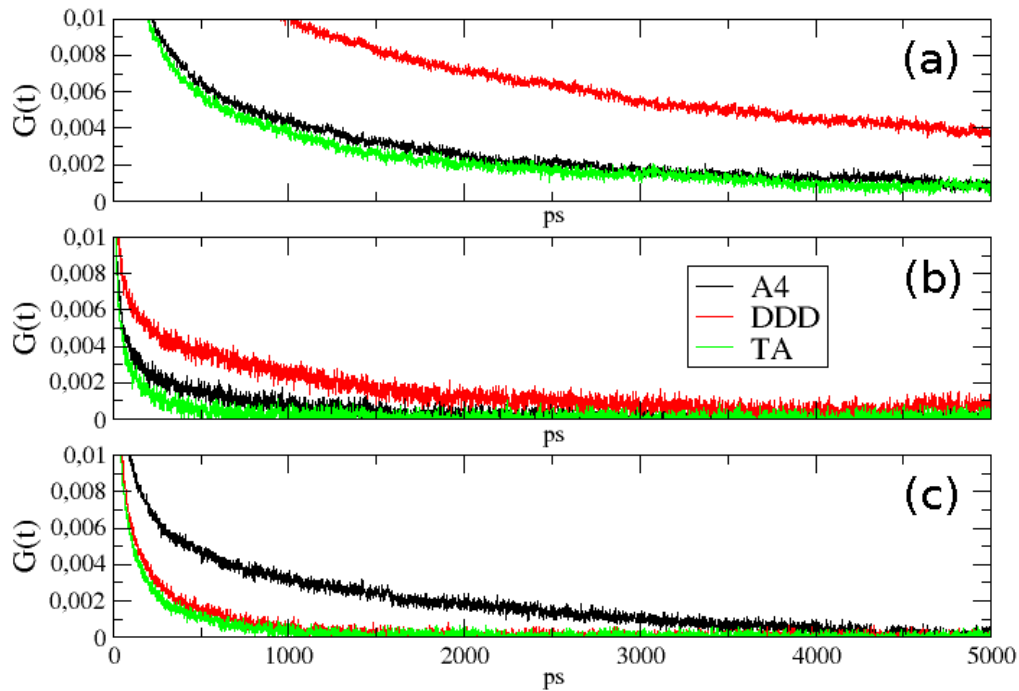


Figura 5.11: Andamento delle funzioni di autocorrelazione temporale  $G(t)$ , per i diversi set di parametri (a), (b) e (c) e per i diversi oligomeri. Soltanto le simulazioni realizzate con il set (c) riproducono il comportamento atteso.

## 5.4 Conclusioni

Le dinamiche svolte sui tre oligomeri di DNA mostrano che i set di parametri utilizzati nelle simulazioni, possono produrre differenze quantitative, e in alcuni casi qualitative, nei risultati relativi alla dipendenza, dalla sequenza di basi, delle interazioni tra  $Na^+$

e DNA. Poichè i dati sperimentali non permettono di avere informazioni al livello atomico da comparare alle simulazioni, è difficile valutare la bontà dei risultati sulla base delle RDF e dei RT. Il calcolo del gradiente del campo elettrico  $e$ , da questo, della funzione di autocorrelazione temporale, consente invece un confronto qualitativo diretto con i dati sperimentali, e permette di evidenziare come le simulazioni effettuate con il set di parametri ( $c$ ), sia l'unica in accordo con i dati sperimentali.

## 6 Applicazioni ai liquidi ionici

I liquidi ionici (IL) costituiscono un'ampia classe di composti, formati esclusivamente da ioni, la cui temperatura di fusione è inferiore o prossima ai 100 °C [105]. Dal punto di vista chimico, i liquidi ionici sono solitamente ottenuti dall'associazione di cationi organici e anioni inorganici [106].

Le caratteristiche di questi inusuali composti coprono un ampio intervallo di variabilità, dovuto all'elevato numero di possibilità nella scelta dell'anione e del catione [108]. In generale, rispetto ai sali inorganici, essi risultano più facili da maneggiare, avendo volatilità quasi nulla, e presentano elevata ampiezza in temperatura della fase liquida, grande stabilità termica, buona conduttività elettrica [104, 110, 111], caratteristiche che conferiscono loro proprietà peculiari sia come solventi che come liquidi puri. Per questo, dal punto di vista applicativo, i liquidi ionici presentano grande interesse: sono ottimi solventi sia per composti organici che inorganici, le cui caratteristiche sono praticamente modulabili a piacimento, grazie all'opportuna scelta di anione e catione.

Nell'ultimo decennio, i liquidi ionici sono stati oggetto di intensi studi teorici condotti, tra le altre, con tecniche computazionali come la Dinamica Molecolare [112, 113, 115, 106].

Le simulazioni basate su potenziali classici, hanno aiutato a comprendere più a fondo il legame tra la struttura chimica e le proprietà di questi composti, arrivando a predire fenomeni inaspettati, successivamente confermati dall'esperimento [112]. Lo sviluppo di potenziali intermolecolari affidabili, per l'esecuzione di simulazioni MD o Monte Carlo, è quindi attualmente un obiettivo generale di primaria importanza [118, 127].

### 6.1 Studio e motivazioni

Nel contesto appena delineato, si inquadra lo studio descritto in questo capitolo, svolto in collaborazione con il prof. Aatto Laaksonen e il dott. Yong Lei Wang del Dipartimento

di Materiali e Chimica Ambientale dell'Università di Stoccolma e con il dott. Kestutis Aidas, del Dipartimento di Fisica Generale e Spettroscopia dell'Università di Vilnius.

L'alta funzionalità dei liquidi ionici in un ampio insieme di applicazioni, che vanno dalla chimica all'ingegneria, è in gran parte il risultato del ruolo complementare giocato dalle forze di Van der Waals, di debole entità, e dalle più intense interazioni elettrostatiche. Lo scenario altamente complesso che tali composti offrono, rende impegnativo il loro studio, sia dal punto di vista teorico che sperimentale. La combinazione di tecniche sperimentali e computazionali, è quindi di grande importanza, sia per ottenere adeguati modelli computazionali, sia come ausilio per l'interpretazione, al livello microscopico, dei dati sperimentali.

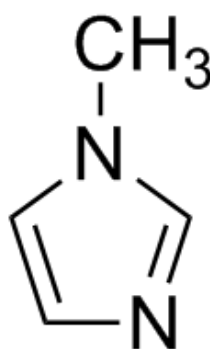


Figura 6.1: Il *N*-metilimidazolio, costituito da un anello imidazolico modificato per sostituzione dell'idrogeno legato all'azoto.

Tra le diverse tecniche computazionali, la Dinamica Molecolare si propone come valido strumento di indagine. L'applicabilità degli studi MD, dipende dall'affidabilità del modello adottato per la parametrizzazione delle interazioni elementari. La necessità di rappresentare, contemporaneamente e in maniera adeguata, interazioni di Van der Waals e Coulomb, rende la determinazione di un set di parametri valido per le simulazioni, un obiettivo non triviale.

Tra gli anioni dei liquidi ionici sono stati ampiamente impiegati gli alogeni, in particolare il  $Cl^-$  e il  $Br^-$ , che sono provvisti di momento di quadrupolo elettrico. Il segnale NMR di tali anioni, è stato utilizzato per investigare l'organizzazione mesoscopica di liquidi ionici a temperatura ambiente, ottenendo importanti informazioni sulla eterogeneità al livello microscopico dei sistemi analizzati [119]. Con anioni quali il  $Cl^-$  e il  $Br^-$ , è possibile una combinazione tra teoria ed esperimento attraverso l'applicazione

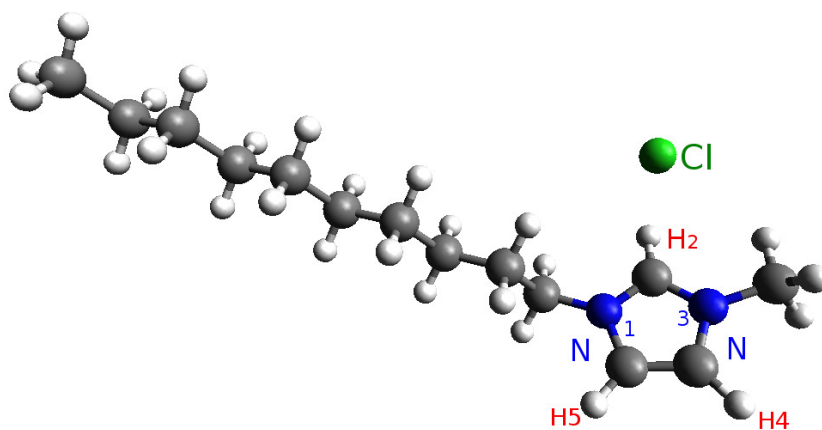
della tecnica illustrata nel cap.5, consistente nel calcolo, a partire dalle simulazioni MD, dei tempi di rilassamento di ioni dotati di momento elettrico quadrupolare; ciò rende possibile il confronto sistematico con i dati delle misure NMR e può essere utilizzato per avere una stima dell'adeguatezza dei modelli computazionali adottati.

Con questo intento, si è realizzata una serie di simulazioni MD di un particolare liquido ionico, il *1-decil-3-metil-imidazolio cloruro* (DMIm). Le simulazioni sono state condotte con diversi set di parametri; valutando la densità e i parametri del rilassamento quadrupolare dell'anione  $^{35}\text{Cl}$ , si è potuta valutare la loro bontà .

Il programma TRIlly è stato utilizzato per calcolare il gradiente del campo elettrico nelle posizioni occupate dagli ioni, in istanti diversi nel corso della dinamica. La versione GPU di M.DynaMix è stata invece utilizzata per la generazione di una parte delle traiettorie analizzate.

## 6.2 Il DMIm

Il 1-decil-3-metil-imidazolio cloruro è un liquido ionico costituito da un anello imidazolico, a cui sono legati in posizione 1 un gruppo decilico e in posizione 3 un gruppo metilico. La molecola, la cui formula grezza è  $\text{C}_{14}\text{H}_{27}\text{N}_2$ , contiene in totale 43 atomi.



*Figura 6.2: Il DMIm, costituito da un anello metil-imidazolico, cui è legato in posizione 1 un gruppo decilico. Nel composto, che contiene complessivamente 43 atomi, il catione imidazolico è bilanciato dal controanione cloro.*



## 6.3 Force Fields

Diversi FF per liquidi ionici contenenti imidazolo sono stati proposti [127, 118]. Un limite spesso riscontrato nei campi di forze per liquidi ionici, è la cattiva riproduzione delle proprietà dinamiche [118]. Sono stati proposti alcuni possibili correttivi per ovviare a questo limite, quali la riduzione della carica di anione e catione [122]. La riduzione di carica richiede però una riparametrizzazione completa del campo di forze; inoltre, i risultati ottenuti nel gruppo di ricerca del Prof. Laaksonen, in collaborazione con il quale è stato svolto questo lavoro, hanno mostrato che in alcuni casi la dinamica può essere bene riprodotta anche senza tale riduzione [115].

Sulla base di queste considerazioni, si è optato per la scelta di un Force Field per il catione proposto da Liu [114], esplicitamente sviluppato per la famiglia di liquidi ionici  $C_n\text{MIm}$  contenenti anelli imidazolici e non facente uso di cariche ridotte. Non essendo presente, nel suddetto FF di Liu, una parametrizzazione per il cloro, sono stati presi inizialmente in considerazione tre ulteriori set di parametri, che nel seguito saranno indicati come  $(AP)$  [116],  $(LU)$  [117] e  $(SD)$  [75],

## 6.4 Simulazione

Con lo scopo di selezionare il set di parametri più idoneo alla descrizione del del  $Cl$  nel DMIm, sono state inizialmente condotte tre distinte simulazioni, ciascuna relativa a uno dei set. Tali simulazioni, sono state divise in una fase di *equilibratura* e in una fase di *pre-produzione*. Successivamente, utilizzando solo il set di parametri che, sulla base della selezione, è stato ritenuto più affidabile, si è dato luogo ad una fase di *produzione* vera e propria.

I sistemi presi in considerazione nelle diverse fasi, di identica composizione, erano costituiti da 125 molecole di DMIm, neutralizzate da 125 controanioni  $Cl^-$ , per un totale di 5500 atomi, collocati in una cella di simulazione cubica.

Ciascuna delle equilibrature, necessarie per condurre i sistemi dalla configurazione iniziale ad una configurazione di equilibrio, ha avuto una durata di  $8\text{ ns}$  ed è stata realizzata con un protocollo analogo a quello descritto in [115].

Nelle tre fasi di pre-produzione, di durata pari a  $\sim 10\text{ ns}$ , sistemi sono stati mantenuti in condizioni isotermitiche e isobariche, con  $T = 350\text{ K}$  e  $P = 1\text{ bar}$ . I risultati di tali fasi hanno fornito elementi per la comparazione dei tre set secondo diversi criteri, quali

la stima della densità del liquido, l'andamento delle funzioni di densità radiale (RDF) e delle funzioni di distribuzione spaziale (SDF).

La fase di produzione, condotta anch'essa in condizioni NPT con  $T = 350 K$  e  $P = 1 bar$ . ha avuto la durata di alcune centinaia di  $ns$ ; l'attendibilità dei risultati di quest'ultima fase, è stata verificata attraverso il calcolo dei tempi di rilassamento quadrupolare del cloro, secondo lo schema delineato nel cap. 5.

Le simulazioni sono state effettuate con l'utilizzo del programma M.DynaMix; si è fatto uso della tecnica del doppio time step, con passo di integrazione lungo pari a  $2fs$  e un numero di 10 passi brevi all'interno di un passo lungo.

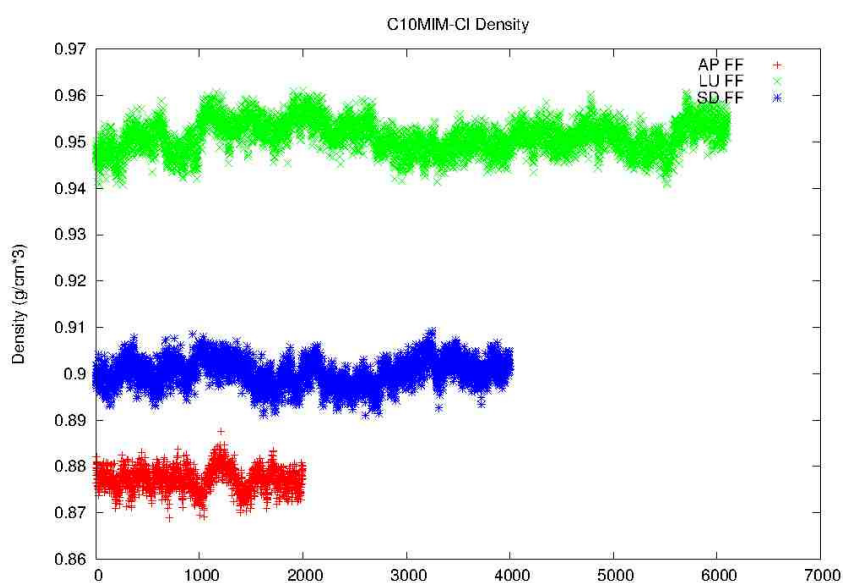


Figura 6.3: Densità del liquido ionico DMIm al variare del tempo ottenuta dalle fasi di pre-produzione, per i set di parametri Rosso: (AP) +, blu: (SD) \*, verde: (LU) \*.

### 6.4.1 Comparazione dei set

La fig 6.3 rappresenta il valore della densità del liquido al variare del tempo nel corso delle tre fasi di pre-produzione. La differenza tra le densità nei diversi casi, che copre un intervallo di ampiezza complessivo pari a oltre il 10%, indica come la scelta dei

parametri per l'anione sia di fondamentale importanza nel determinare questo parametro strutturale.

Considerando che il valore sperimentale della densità è  $0.98 \text{ g/cm}^3$  [128], si può osservare come il set ( $LU$ ), riproduca le proprietà strutturali decisamente meglio degli altri due. Ciò costituisce un punto a favore del set ( $LU$ ).

L'assenza di simulazioni sul DMIm-Cl, ha indirizzato verso la consultazione di ricerche su liquidi ionici simili, differenti unicamente per la lunghezza della catena alchilica legata ad  $N1$ . Risultati ottenuti per il *1-etil-3-metil-imidazolio cloruro* (C2MIm) con controanione cloro [119], hanno sottolineato importanti differenze tra i risultati delle simulazioni MD e dinamiche da principi primi. Una prima distinzione si osserva comparando le RDF ottenute con i due approcci; risulta che il primo picco della RDF calcolata tra il protone  $H2$  (fig. 6.2) e il  $Cl^-$ , appare spostato verso distanze più grandi nella trattazione classica, con un incremento di  $0.22\text{Å}$ .

Nella fig. 6.4 sono rappresentate le funzioni di densità radiale normalizzate, tra il protone  $H2$  e il  $Cl^-$ , calcolate dalle traiettorie MD realizzate con i tre diversi set di parametri. Il grafico mostra come la posizione del massimo sia dipendente dal set di parametri utilizzato; in particolare, la funzione ottenuta con il set ( $LU$ ), presenta il picco ad una distanza inferiore rispetto agli altri due, risultando in migliore accordo con il valore di  $2.4\text{Å}$  ottenuto negli studi precedentemente indicati.

Una ulteriore distinzione tra risultati di dinamiche classiche e da primi principi, riguarda le SDF del cloro nell'intorno dell'anello imidazolico ed in particolare in vicinanza dell'idrogeno  $H2$ . Mentre l'approccio quantistico prevede che l'anione tenda a concentrarsi lungo la direzione  $C2 - H2$ , quello classico mostra, nella stessa orientazione, un minimo locale [119].

La fig. 6.5 riporta alcune superfici di livello della SDF del  $Cl^-$  nell'intorno dell'imidazolio, per i tre set di parametri considerati. Sebbene un confronto diretto con i valori della letteratura non possa essere fatto per la diversa lunghezza delle catene alchiliche, si può osservare che il set ( $SD$ ) prevede un minimo pronunciato in corrispondenza dell'asse  $C2 - H2$ ; per la stessa superficie di isodensità, anche gli altri due set di parametri indicano la presenza di tale minimo, che però risulta molto meno accentuato; ( $AP$ ) ed ( $LU$ ) presentano dunque un comportamento più vicino a quello quantistico [126, 124, 125]. È interessante notare che i tre set di parametri adottati nel presente studio, conducono a distribuzioni ioniche in ogni caso più vicine a quelle quantistiche, rispetto ad altri modelli classici [127] che sistematicamente generano densità tra i legami  $C - H$ .

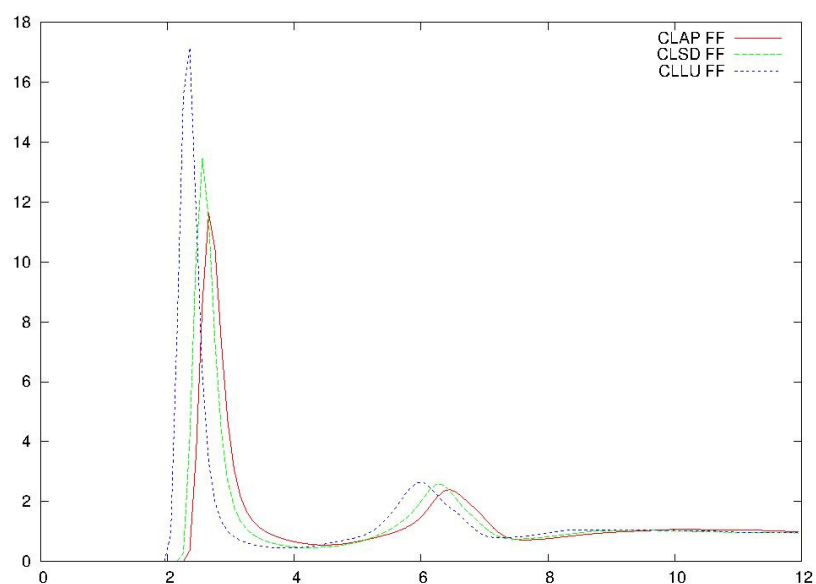


Figura 6.4: Funzioni di densità radiale normalizzate tra il protone H2 e il  $Cl^-$ , calcolate dalle traiettorie MD delle fasi di pre-produzione, realizzate con i set di parametri Rosso: (AP), blu: (SD), verde: (LU).

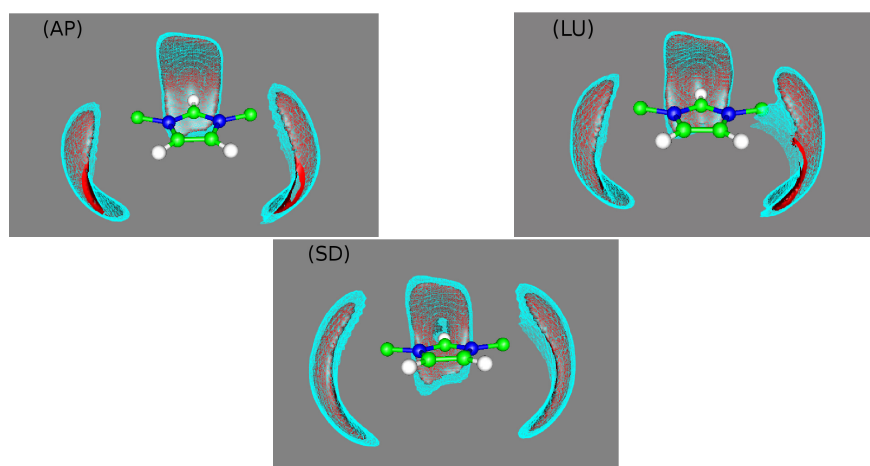


Figura 6.5: Funzioni di distribuzione spaziale di  $Cl^-$  in vicinanza dell'anello inidazolico dei DMIm. Le superfici colorate racchiudono regioni con una probabilità di occupazione 20 (ciano) e 30 (rosso) volte superiore quella del bulk, ottenute dalle simulazioni MD delle fasi di pre-produzione, effettuate con i set di parametri (AP), (LU) e (SD).

In considerazione dei risultati della pre-produzione, si è concluso che ( $LU$ ) sia il più affidabile tra i set di parametri considerati; esso è stato scelto per la realizzazione della fase di produzione.

## 6.5 Studio del rilassamento quadrupolare

In recenti studi NMR sono stati misurati i tempi di rilassamento quadrupolare del controanione  $Cl^-$  nel liquido ionico, il 1-decil-3-metil-imidazolio cloruro [129]. La determinazione di queste quantità dalle simulazioni ed il successivo confronto con l'esperimento, si presenta quindi come un potente strumento di indagine per la validazione dei risultati delle dinamiche.

Il calcolo dei parametri di rilassamento NMR in questo contesto, assume importanza rilevante anche per la comprensione della natura delle interazioni alla base dei tempi di rilassamento *brevissimi*, osservati di recente per gli alogenuri  $Cl$  e  $Br$  in liquidi ionici [129].

### 6.5.1 Confronto con l'esperimento

La teoria del rilassamento quadrupolare è stata descritta nella sez. 5.2, alla cui simbologia faremo riferimento. Inoltre, come descritto nella sez. 3.3.1, nella fase di sviluppo degli strumenti di informatici, è stato messo a punto il codice TRILly, per il calcolo del gradiente del campo elettrico in corrispondenza delle posizioni occupate dai nuclei degli anioni  $Cl^-$ .

A partire dalle traiettorie atomiche generate da M.DynaMix, tramite il programma TRILly, si è realizzato il calcolo del gradiente del campo elettrico  $V(t)$  nelle posizioni occupate dai nuclei dei controanioni  $Cl^-$ . Dalla conoscenza di  $V(t)$ , tramite il programma di analisi TRANAL, si è determinata la funzione di autocorrelazione temporale  $G(t)$ , data dalla eq. (5.3) e, successivamente, i tempi di rilassamento, definiti come gli inversi delle velocità di rilassamento, la cui espressione è data nella sez. 5.2.1.

La fig. 6.6 mostra l'andamento della  $G(t)$  in corrispondenza delle posizioni occupate dai nuclei dei controanioni  $Cl^-$ , ottenuta nella fase di produzione effettuata con il set di parametri ( $LU$ ); la durata della simulazione è di 120 ns.

In analogia a quanto fatto nell'interpretazione dei dati sperimentali su questo sistema [129], abbiamo analizzato il decadimento della funzione di autocorrelazione tempo-

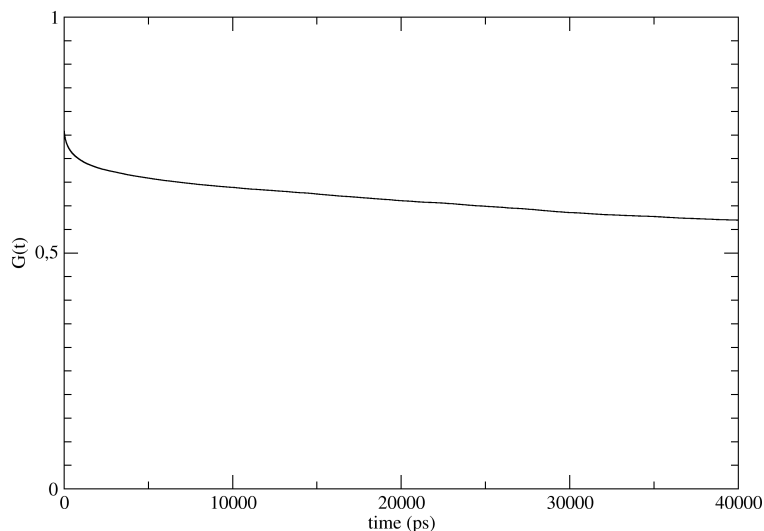


Figura 6.6: Funzione di autocorrelazione temporale del gradiente del campo elettrico in corrispondenza delle posizioni occupate dai nuclei dei controanioni  $Cl^-$ , nel liquido ionico  $DMIm-Cl$ , ottenuta dalla simulazione  $MD$ , della durata di  $120 ns$ , effettuata con il set di parametri ( $LU$ ).

rale, assumendo che fosse descritto da un solo esponenziale  $G(t) = ae^{-t/\tau_c}$ . In questo modo, è stato possibile ottenere una stima del tempo di correlazione  $\tau_c$  attraverso un best fit dei dati, ottenendo il valore  $\tau_c = 212 ns$ .

Il calcolo della costante di accoppiamento quadrupolare  $\chi$ , può essere eseguito con la relazione (5.4). Il valore di  $\chi$ , ottenuto come media sull'intero insieme di controanioni  $Cl^-$ , è  $\chi = 7 MHz$ . Tale valore risulta più alto dei valori sperimentali pari a  $1.5 MHz$  e  $0.7 MHz$ , relativi rispettivamente ai liquidi ionici  $[C4MIm][Cl]$  e  $[C2MIm][Cl]$  allo stato solido. Sebbene sovrastimato, il valore di  $\chi$  pare ancora di entità ragionevole per questa tipologia di liquidi ionici, soprattutto osservando come il valore sperimentale di  $\chi$  cresca all'aumentare della lunghezza della catena alchilica; è ragionevole ritenere che questo comportamento sia valido anche per catene più lunghe e che il valore calcolato possa avvicinarsi maggiormente ai dati reali.

La velocità di rilassamento, nel caso di  $G(t)$  monoesponenziale, è espressa dalla relazione (5.7) ed è funzione esplicita della frequenza  $\omega$ . Per poter comparare i risultati

della simulazione con lo studio sperimentale di riferimento [129], si è considerato lo stesso valore di frequenza, pari a  $\omega = 307.56 \text{ MHz}$ . Con queste assunzioni, il valore della velocità di decadimento è risultata pari a  $R_1 = 3.8 \cdot 10^3 \text{ Hz}$  e il suo inverso, il tempo di rilassamento teorico, pari a  $T_1 = 265 \mu\text{s}$ ; il valore di  $T_1$  è direttamente comparabile con la misura, il cui risultato è  $T_{1,exp} = 205.6 \mu\text{s}$ . La tabella 6.1 riassume le quantità stimate dalla simulazione ed il valore sperimentale del tempo di rilassamento.

$\tau_c$	$\chi$	$\omega$	$R_1$	$T_1$
212 ns	7 MHz	307.56 MHz	$3.8 \cdot 10^3 \text{ Hz}$	265 $\mu\text{s}$

Tabella 6.1: Parametri del rilassamento quadrupolare del  $Cl$ , stimati nella fase di produzione.

Il valore stimato del tempo di rilassamento  $T_1$ , appare in ottimo accordo con il dato sperimentale; tale corrispondenza è un forte punto a sostegno della validità della simulazione condotta.

I Force Field adottati, in particolare il set di parametri utilizzato per il controanione, paiono descrivere adeguatamente le interazioni elementari tra gli elementi del complesso; in questo quadro generale, è possibile riporre fiducia nella descrizione della dinamica, al livello atomico, ottenuta dalla simulazione ed usare tale descrizione come strumento di interpretazione per la comprensione dei meccanismi alla base di comportamenti e proprietà macroscopiche.

### 6.5.2 Moto dei singoli anioni

Il valore  $\chi = 7 \text{ MHz}$  determinato nella sezione precedente, è stato ottenuto come media valutata sull'intero insieme di ioni  $Cl^-$ . Una ulteriore analisi è stata condotta, per determinare il contributo apportato da ciascun controione alla media complessiva, cioè il valore della costante di accoppiamento  $\chi_i$  relativa al singolo controione.

I risultati sperimentali NMR [129], mostrano la presenza di un solo segnale per tutti gli ioni, che appaiono quindi equivalenti dal punto di vista delle interazioni microscopiche. Se la simulazione riproducesse adeguatamente questo comportamento, cioè se gli ioni fossero capaci di avere un sufficiente scambio tra tutti i tipi rappresentativi di coordinazione, i singoli valori di  $\chi_i$  varierebbero in un intervallo ristretto. Il risultato ottenuto

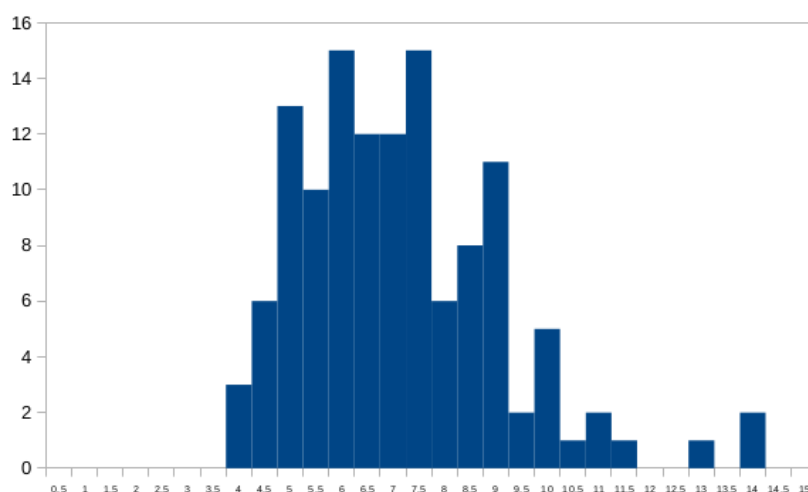


Figura 6.7: Distribuzione delle  $\chi_i$  per i 125 anioni  $\text{Cl}^-$  del sistema. In ascisse è riportato il valore di  $\chi$ , raggruppato in classi di ampiezza 0.5 MHz, in ordinate il numero di atomi in ciascuna classe.

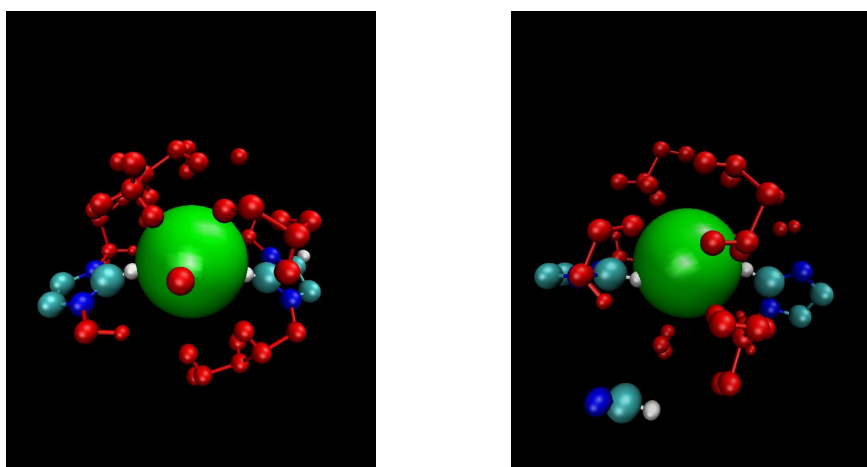


Figura 6.8: Rappresentazione di Van der Waals di due anioni ( $\text{Cl}_{36}$  e  $\text{Cl}_{34}$ ), caratterizzati da un elevato valore di  $\chi$  ( $\chi_{36} = 14 \text{ MHz}$ ,  $\chi_{34} = 13 \text{ MHz}$ ), insieme con il più vicino catione coordinato (bianco, blu e ciano, indicano rispettivamente H, N, e C dell'anello imidazolico, il rosso la catena laterale). In entrambe i casi, il cloro si trova lungo la direzione C – H2.



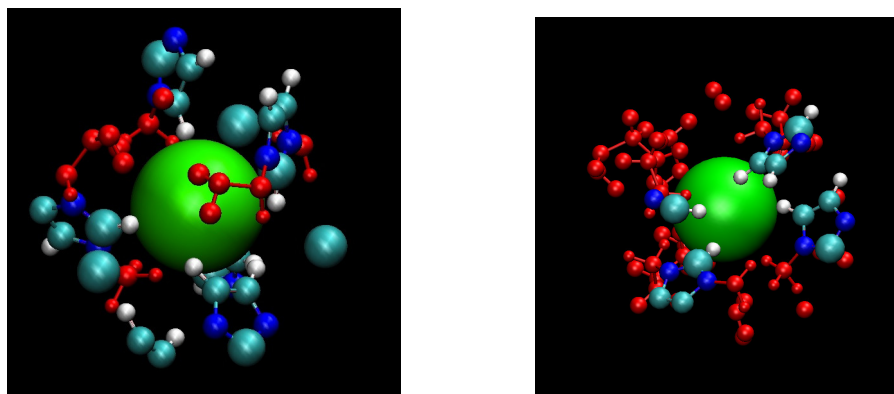


Figura 6.9: Esempio di coordinazione attorno agli ioni  $Cl_1$  e  $Cl_{70}$  aventi  $\chi_i = 6.2 MHz$ . La coordinazione avviene, in entrambe i casi, tramite gli idrogeni H2, H4, H5.

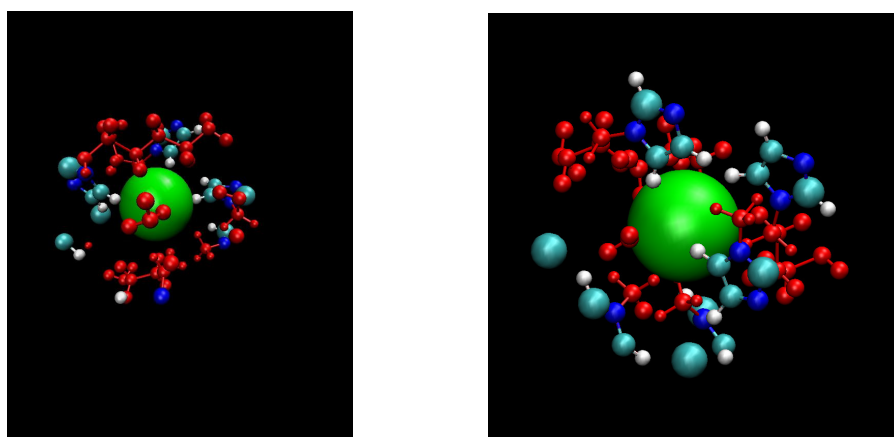


Figura 6.10: Gli anioni con basso valore di  $\chi_i$  (sinistra:  $\chi_{105} = 4.4 MHz$ , destra:  $\chi_{98} = 3.7 MHz$ ), sono coordinati con l'anello imidazolico attraverso gli atomi di idrogeno meno acidi.

fornisce invece una indicazione opposta, con valori di  $\chi_i$  compresi tra  $3.5 MHz$  e  $14 MHz$  al variare dallo ione considerato. La fig. 6.7 riporta l'istogramma della distribuzione delle  $\chi_i$ , ottenuto raggruppando i dati dei 125 ioni, in classi di ampiezza  $0.5 MHz$ .

L'alta variabilità delle  $\chi_i$  è una indicazione del fatto che la durata della simulazione è troppo limitata per poter fornire un risultato attendibile in questo senso; gli ioni  $Cl^-$ , impegnati in diversi legami con il DMIm, non hanno avuto tempo sufficiente per scambiare tra tutti i possibili tipi di coordinazione.

Un'analisi visuale delle traiettorie, ha permesso di verificare che anioni con valori simili di  $\chi_i$ , hanno anche coordinazioni simili, che rimangono stabili nel corso della dinamica.

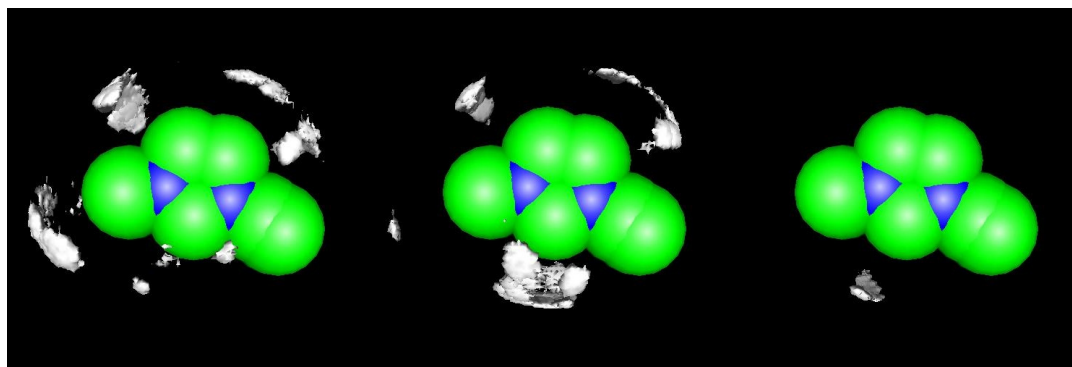


Figura 6.11: Rappresentazione delle SDF del  $Cl^-$  attorno all'anello imidazolico suddivise in base alla costante di accoppiamento quadrupolare. Sinistra: ioni con bassa  $\chi$  ( $< 4 MHz$ ); centro:  $\chi$  intermedia ( $= 7 \pm 0.5 MHz$ ); destra:  $\chi$  elevata ( $> 12 MHz$ ). Gli atomi di idrogeno sono omessi per chiarezza, così come gli atomi della catena alchilica successivi al primo. Come aiuto per la visualizzazione, il primo carbonio della catena decilica (a destra dell'imidazolio) è rappresentato diversamente dal carbonio metilico legato al C3 (a sinistra dell'imidazolio). Per ciascun ione cloro il livello di isodensità della rappresentazione della SDF è pari a metà del valore massimo.

Un esempio è fornito dalle figure 6.8, 6.9 e 6.10, relative rispettivamente a coppie di ioni con  $\chi_i$  elevata, media e bassa.

Il calcolo della  $\chi$  può rappresentare un metodo rapido per evidenziare differenze di coordinazione spesso non semplici da razionalizzare con un'analisi visiva o tramite le sole SDF.

Tuttavia, per avere un'analisi sistematica e non semplicemente aneddotica, è estremamente utile ricorrere alle funzioni di distribuzione spaziale. In figura 6.11, sono comparate le SDF del  $Cl^-$  attorno all'anello imidazolico, calcolate separatamente per ciascuno anione e raggruppate in tre gruppi in funzione del valore della costante di accoppiamento  $\chi$ : bassa ( $\chi < 4 MHz$ ), intermedia ( $\chi = 7 \pm 0.5 MHz$ ), elevata ( $\chi > 12 MHz$ ). Gli ioni con basso valore di  $\chi$ , coordinano prevalentemente con gli atomi di idrogeno legati al C metilico e con H4 e H5, mentre la prossimità ad H2 è molto limitata. La situazione cambia per  $\chi$  intermedie e la coordinazione all'idrogeno acido H2 è molto più significativa, mentre si riduce in proporzione quella agli altri idrogeni, in particolare a quelli metilici. A  $\chi$  elevate, si evidenzia una coordinazione esclusiva al protone più acido. L'ampia gamma di tipi di coordinazione messe in evidenza in questo studio, suggerisce che le analisi della coordinazione del cloro in liquidi ionici a base di imidazolio, debbano

essere effettuate su sistemi contenenti un ampio numero di coppie. La rappresentazione delle figure è stata realizzata con *gOpenMol* [130].

## 6.6 Conclusioni

Le dinamiche svolte sul liquido ionico DMIm, ed il calcolo dei parametri del decadimento quadrupolare del  $Cl$ , hanno consentito un confronto quantitativo diretto con i dati sperimentali. Il valore stimato del tempo di rilassamento, appare in ottimo accordo con il dato sperimentale; ciò suggerisce come i set di parametri utilizzati per la descrizione del catione e del controanione, rappresentino adeguatamente le interazioni elementari tra gli elementi del composto; in questa situazione appare ragionevole riporre un elevato grado di fiducia nella descrizione della dinamica, al livello atomico, ottenuta dalla simulazione. Inoltre, il calcolo della costante di accoppiamento quadrupolare, può rappresentare un metodo rapido per evidenziare differenze di comportamento tra gli elementi del complesso, spesso non semplici da determinare con un'analisi visiva delle traiettorie o tramite la rappresentazione delle funzioni di distribuzione spaziale dell'insieme di tutti gli ioni appartenenti al sistema.

# Conclusioni

La dinamica molecolare classica è una tecnica di simulazione sempre più utilizzata ed in fase di grande evoluzione, in seguito al rapido evolversi dei computer stessi. Rimanere al passo con i cambiamenti, richiede una periodica rivisitazione dei programmi di calcolo che, se non completamente ristrutturati, devono essere adeguati alle possibilità offerte dall'hardware moderno.

Durante lo svolgimento del lavoro di ricerca, è emersa l'opportunità di utilizzare le GPU nel campo della chimica computazionale. Questo ha condotto alla decisione di adattare all'architettura GPU il programma M.DynaMix, uno dei codici maggiormente utilizzati dal gruppo dell'Università di Stoccolma, presso il quale parte della presente ricerca è stata svolta.

M.DynaMix è stato reso più efficiente anche grazie all'implementazione di due nuovi algoritmi per il calcolo rapido della somma di Ewald, necessaria per il trattamento corretto dell'elettrostatica in sistemi altamente carichi.

Il notevole sforzo condotto su questa parte del lavoro di ricerca, ha portato ad un marcato miglioramento delle prestazioni del programma, la cui efficienza è aumentata di oltre un ordine di grandezza. I tempi di calcolo, sono uno degli aspetti più importanti di cui tenere conto nella progettazione di un lavoro di simulazione.

Altro importante fattore di cui occorre tenere conto nel realizzare uno studio basato su simulazioni, è l'attendibilità dei risultati prodotti. Purtroppo, il confronto diretto con i dati sperimentali non sempre è possibile e la validazione delle conclusioni teoriche può risultare difficoltosa.

In questo lavoro di ricerca è stato mostrato come, in sistemi ad elevata carica contenenti ioni monoatomici quadrupolari, si possa utilizzare il calcolo dei parametri del rilassamento della magnetizzazione, per valutare la qualità della simulazione.

Per realizzare il calcolo di tali parametri, è stato sviluppato un codice *ad-hoc* che permette la determinazione del gradiente di campo elettrico, a partire da traiettorie ottenute con un qualunque programma di simulazione MD.

Il calcolo dei parametri del rilassamento magnetico di ioni quadrupolari, è stato applicato a due tipi di sistemi ad elevata carica. In un caso (studio delle interazioni tra DNA e un suo controione fisiologico), è stato utilizzato per verificare quale, tra diversi set di simulazioni, con differenti parametri di campo di forze, fornisse risultati migliori. In un altro caso, è stato utilizzato per verificare la bontà di un campo di forze per un liquido ionico. In entrambe le situazioni, il calcolo di uno o più parametri del rilassamento magnetico degli ioni quadrupolari presenti nel sistema, ha dato importanti indicazioni sulla validità delle simulazioni.

## **Appendice 1: pubblicazioni**

# Non-Uniform FFT and Its Applications in Particle Simulations

Yong-Lei Wang<sup>1</sup>, Fredrik Hedman<sup>2</sup>, Massimiliano Porcu<sup>1,3</sup>, Francesca Mocci<sup>1,3</sup>, Aatto Laaksonen<sup>1</sup>

<sup>1</sup>Department of Materials and Environmental Chemistry, Arrhenius Laboratory, Stockholm University, Stockholm, Sweden

<sup>2</sup>Noruna AB, Stockholm, Sweden

<sup>3</sup>Dipartimento di Scienze Chimiche e Geologiche, Cagliari University, Cittadella Universitaria di Monserrato (CA), Monserrato, Italy  
Email: [yonglei.wang@mmk.su.se](mailto:yonglei.wang@mmk.su.se), [fredrik.hedman@noruna.se](mailto:fredrik.hedman@noruna.se), [massi@dsf.unica.it](mailto:massi@dsf.unica.it), [fmocci@unica.it](mailto:fmocci@unica.it), [aatto.laaksonen@mmk.su.se](mailto:aatto.laaksonen@mmk.su.se)

Received October 19, 2013; revised November 19, 2013; accepted November 26, 2013

Copyright © 2014 Yong-Lei Wang *et al.* This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. In accordance of the Creative Commons Attribution License all Copyrights © 2014 are reserved for SCIRP and the owner of the intellectual property Yong-Lei Wang *et al.* All Copyright © 2014 are guarded by law and by SCIRP as a guardian.

## ABSTRACT

**Ewald summation method, based on Non-Uniform FFTs (ENUF) to compute the electrostatic interactions and forces, is implemented in two different particle simulation schemes to model molecular and soft matter, in classical all-atom Molecular Dynamics and in Dissipative Particle Dynamics for coarse-grained particles. The method combines the traditional Ewald method with a non-uniform fast Fourier transform library (NFFT), making it highly efficient. It scales linearly with the number of particles as  $\mathcal{O}(N \log N)$ , while being both robust and accurate. It conserves both energy and the momentum to float point accuracy. As demonstrated here, it is straightforward to implement the method in existing computer simulation codes to treat the electrostatic interactions either between point-charges or charge distributions. It should be an attractive alternative to mesh-based Ewald methods.**

## KEYWORDS

**Ewald Summation Method; Non-Uniform Fast Fourier Transform; Dissipative Particle Dynamics**

## 1. Introduction

Computer modeling and simulation of molecular systems is an established discipline rapidly expanding and widely used from Materials Science to Biological systems. At finite temperatures and normal pressures, molecular compounds are in continuous rapid motion. In liquids and solutions, molecules collide and interact with each other at close distances. They also interact with external fields. In computer simulations of molecular systems, mechanistic ball and spring models are commonly employed to give a simple picture of atoms with specific sizes and masses bonded together with covalent bonds. Molecular equilibrium geometries and interactions are defined by so-called force fields where a somewhat arbitrary division between intramolecular and intermolecular interactions is made, because this partitioning is a simplification of the more detailed and fundamental understanding furnished by a quantum mechanical description.

Intramolecular interactions are normally described by bond stretching, angle bending and torsional angle motion terms. These interactions involve the closest bonded atoms described by two-, three-, and four-body terms, respectively. The bond and angle terms are normally given as harmonic wells while the torsion term is most often expressed as a Fourier sum.

Intermolecular interactions are those between separate molecules but also include all interactions within the same molecule beyond the bonded interactions. Non-bonded interactions are further divided between short-ranged and long-ranged interactions. The short-range interactions mimic the Van der Waals type of forces, while the long-range interactions are electrostatic interactions. The electron distributions around atoms are approximated by fixed point charges, and their interactions are treated by using Coulomb's law.

The effective range of short-range interactions is limited to a specified cut-off. By assuming a uniform beyond the cut-off, correction terms to the energy can be obtained by integrating over the non-zero part of the interactions [1-3]. Thus the fast-decaying short-range interactions can be accurately approximated by truncation at the cut-off distance in most calculations.

Artificially collecting and dividing the diffuse and fluctuating electron densities inside and around molecules on single atomic sites are a crude but conceptually simple and effective approximation, since Coulomb's law can be invoked. However, this simplification comes at a price since the interactions between point charges stretch over very long distances. Furthermore, these long-range interactions cannot be truncated without introducing simulation artifacts [4-7].

As the system size grows, calculating the electrostatic interactions becomes the major computational bottleneck. Methods based on Ewald summation [8,9] are still considered as the most reliable choice, and a large variety of schemes to compute them in computer simulations have been proposed [10-16]. There is also a multitude of alternative methods for representing electrostatic interactions. Examples are methods based on a cut-off [17-19], tree and multipole based methods [20-25], multigrid methods [26-28], reaction field methods [29-31], the particle mesh method [32,33] and the isotropic sum method [34].

In this paper, we describe an approach to Ewald summation based on the non-uniform fast Fourier transform technique. We use the acronym ENUF-Ewald summation using Non-Uniform fast Fourier transform (NFFT) technique. Our method combines the traditional Ewald summation technique with the NFFT to calculate electrostatic energies and forces in molecular computer simulations. In the paper, we show that ENUF is an easy-to-implement, practical, and efficient method for calculating electrostatic interactions. Energy and momentum are both conserved to float point accuracy. By a suitable choice of parameters, ENUF can be made to behave as traditional Ewald summation but at the same time gives a computational complexity of  $\mathcal{O}(N \log N)$ , where  $N$  is the number of electrostatic interaction sites in the system. Weighing all these properties together, we believe that ENUF should be an attractive alternative in simulations where the high accuracy of Ewald summation is desired.

In the next section, we summarize the basic methodology to apply Ewald summation to computing the electrostatic energies and forces within periodic boundary conditions. In Section 3, we introduce the Ewald method where the reciprocal space part is calculated based on non-uniform FFTs and discuss the underlying concepts and use of the libraries. We also give general guidelines to implement the method in existing simulation programs. In Section 4, we discuss its implementation in a general purpose atomistic Molecular Dynamics simulation package M.DynaMix giving its scaling characteristics in a standard desktop computer. In Section 5, we demonstrate its implementation in Dissipative Particle Dynamics package. This method is applied to simulating soft charged mesoscopic particles. It is necessary to use charge distributions in order to avoid non-physical aggregation of soft charged particles if point charges are used. This issue is discussed further in Section 5.

## 2. Electrostatic Interactions

We start by describing a model system of charged particles which captures the most salient features of electrostatic interactions in general MD systems. The electrostatic potential of a system with periodic boundary conditions (PBC) is first stated; we follow with the manipulation of the basic formulas to the form in which they are commonly written; this Section ends with a summary of expressions for both energy and forces.

### 2.1. Ewald Summation

Consider a cubic simulation box with edge length  $L$ , containing  $N$  charged particles, each with a charge  $q_i$ , located at  $\mathbf{r}_i$ . The boundary conditions in a system without cut-off is represented by replicating the simulation box in all directions. The total electrostatic potential energy of the charge-charge interactions is then given by

$$U_{qq} = \sum_{i=1}^N \sum_{j=i+1}^N \frac{q_i q_j}{|\mathbf{r}_{ij}|} + \frac{1}{2} \sum_{n \neq 0} \sum_{i=1}^N \sum_{j=1}^N \frac{q_i q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|} = \frac{1}{2} \sum_n \sum_{i=1}^N \sum_{j=1}^N \frac{q_i q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|}, \quad (1)$$

where  $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ , and  $|\mathbf{r}|$  denotes the length (2-norm) of the vector  $\mathbf{r}$ . Because of the long-range nature of the electrostatic interactions,  $U_{qq}$  includes contributions from all replicas, but exclude self-interactions, which is expressed in the triple sum in Equation (1). The outer sum is taken over all integer vectors



$\mathbf{n} = (n_1, n_2, n_3)$  such that each  $n_i \in \mathbb{Z}$ . The  $\dagger$  symbol on the first summation sign in Equation (1) indicates that the self-interaction terms should not be included, *i.e.*, when  $\mathbf{n} = 0$  then the  $i = j$  terms are omitted.

The sum in Equation (1) is not an absolutely convergent series, but rather conditionally convergent. As a consequence, the order of summation affects the value of the series. In fact, it was discovered by Riemann that any conditionally convergent series of real terms can be rearranged to yield a series which converges to any prescribed sum [35]. In a sense, this is a situation very similar to the case when a linear equation has an infinite number of solutions because it is under-determined; by adding a set of conditions a unique solution may be defined. For the specific case of Equation (1), a physically relevant summation order has to be prescribed and the boundary conditions of the surrounding media have to be specified.

The lattice sum of Equation (1) can be calculated by a method that was first developed in 1921 by Ewald [8]. He used it to calculate lattice potentials in solids. In the context of Molecular Dynamics, there are several different derivations of the Ewald summation method that are more recent; a small selection is given by [3,9]. In the following discussion we mainly follow the work of de Leeuw *et al.* [9,36,37].

In [9], de Leeuw *et al.* developed a technique using convergence factors that transforms the sum of a conditionally convergent series into a series with a well-defined sum. Furthermore, they showed that applying a specific convergence factor is equivalent to a certain summation order. Assuming an overall charge neutral system,  $\sum_i q_i = 0$ , and summing the terms in Equation (1) over all integer vectors  $\mathbf{n}$  in concentric spherical order, they showed that the electrostatic potential energy can be written as

$$U_{qq}(\epsilon_r) = \sum_{1 \leq i < j \leq N} \frac{q_i q_j}{L} \Psi\left(\frac{r_{ij}}{L}\right) + \frac{\xi}{2L} \sum_{i=1}^N q_i^2 + \frac{2\pi}{(2\epsilon_r + 1)L^3} \left(\sum_i q_i r_i\right)^2, \quad (2)$$

when the surrounding media of the periodically replicated cell is a uniform dielectric with dielectric constant  $\epsilon_r$  and distances are calculated with the minimum image convention. When the surrounding media is a conductor ( $\epsilon_r = \infty$ ), the energy can be written as

$$U_{qq}(\epsilon_r = \infty) = \sum_{1 \leq i < j \leq N} \frac{q_i q_j}{L} \Psi\left(\frac{r_{ij}}{L}\right) + \frac{\xi}{2L} \sum_{i=1}^N q_i^2 = U_{qq}(\epsilon_r) - \frac{2\pi}{(2\epsilon_r + 1)L^3} \left(\sum_i q_i r_i\right)^2. \quad (3)$$

From Equation (3) it is clear that the boundary conditions, vacuum or conductor, have an effect on the energy of the system. Depending on the simulated system and the properties of interest, the choice of boundary conditions can affect the obtained results [38,39].

The function  $\Psi$  is given by

$$\Psi(\mathbf{r}) = \sum_n \frac{\operatorname{erfc}(\alpha|\mathbf{r} + \mathbf{n}|)}{|\mathbf{r} + \mathbf{n}|} + \frac{1}{\pi} \sum_{\mathbf{n} \neq 0} \frac{\exp(2\pi i \mathbf{n} \cdot \mathbf{r} - \pi^2 |\mathbf{n}|^2 / \alpha^2)}{|\mathbf{n}|^2}, \quad (4)$$

with the two error functions defined as  $\operatorname{erfc}(x) = 1 - \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt$  and  $\operatorname{erf}(x) = 1 - \operatorname{erfc}(x)$ . The number  $\xi$  used in Equations (2) and (3) is defined as

$$\xi = \sum_{\mathbf{n} \neq 0} \frac{\operatorname{erfc}(\alpha|\mathbf{n}|)}{|\mathbf{n}|} + \frac{1}{\pi} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / \alpha^2}}{|\mathbf{n}|^2} - \frac{2\alpha}{\sqrt{\pi}}, \quad (5)$$

where  $\alpha > 0$  is a free parameter.

Equations (2) and (3) are not in a form that is appropriate for efficient numerical calculations and in the case of Molecular Dynamics simulation we also need expressions for the forces. To arrive at a more suitable form we make the necessary analysis for the electrostatic energy and forces in the following Sections.

## 2.2. Energies in Ewald Summation

To rearrange and expand Equation (2) we first insert  $r_{ij}/L$  in Equation (4)

$$\begin{aligned} \Psi\left(\frac{\mathbf{r}_{ij}}{L}\right) &= \sum_n \frac{\operatorname{erfc}\left(\alpha\left|\frac{\mathbf{r}_{ij}}{L} + \mathbf{n}\right|\right)}{\left|\frac{\mathbf{r}_{ij}}{L} + \mathbf{n}\right|} + \frac{1}{\pi} \sum_{n \neq 0} \frac{\exp\left(2\pi i \mathbf{n} \cdot \frac{\mathbf{r}_{ij}}{L} - \pi^2 |\mathbf{n}|^2 / \alpha^2\right)}{|\mathbf{n}|^2} \\ &= L \sum_n \frac{\operatorname{erfc}\left(\frac{\alpha}{L} |\mathbf{r}_{ij} + \mathbf{n}L|\right)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{\pi} \sum_{n \neq 0} \frac{\exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij} - \pi^2 |\mathbf{n}|^2 / \alpha^2\right)}{|\mathbf{n}|^2}. \end{aligned} \tag{6}$$

Next we rescale  $\alpha$  by making the substitution  $\alpha \rightarrow \alpha L$  in Equations (5) and (6) to get

$$\Psi\left(\frac{\mathbf{r}_{ij}}{L}\right) = L \sum_n \frac{\operatorname{erfc}\left(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|\right)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{\pi} \sum_{n \neq 0} \frac{\exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij} - \pi^2 |\mathbf{n}|^2 / (\alpha L)^2\right)}{|\mathbf{n}|^2} \tag{7}$$

and

$$\xi = \sum_{n \neq 0} \frac{\operatorname{erfc}(\alpha |\mathbf{n}|L)}{|\mathbf{n}|} + \frac{1}{\pi} \sum_{n \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} - \frac{2\alpha L}{\sqrt{\pi}}. \tag{8}$$

Inserting Equations (7) and (8) into Equation (2) we get

$$\begin{aligned} U_{qq}(\epsilon_r) &= \sum_{1 \leq i < j \leq N} q_i q_j \left\{ \sum_n \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{\pi L} \sum_{n \neq 0} \frac{\exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij} - \pi^2 |\mathbf{n}|^2 / (\alpha L)^2\right)}{|\mathbf{n}|^2} \right\} \\ &+ \frac{1}{2L} \sum_i q_i^2 \left[ \sum_{n \neq 0} \frac{\operatorname{erfc}(\alpha |\mathbf{n}|L)}{|\mathbf{n}|} + \frac{1}{\pi} \sum_{n \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} - \frac{2\alpha L}{\sqrt{\pi}} \right] + \frac{2\pi}{(2\epsilon_r + 1)L^2} \left( \sum_i q_i \mathbf{r}_i \right)^2. \end{aligned} \tag{9}$$

Note that the summation in Equation (9) is for  $i < j$  in the first sum. We make further simplifications by studying the terms on the right hand side of Equation (9) for  $n = 0$  and  $n \neq 0$ .

When  $n = 0$  we have the following terms

$$\begin{aligned} &\sum_{1 \leq i < j \leq N} q_i q_j \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij}|)}{|\mathbf{r}_{ij}|} - \frac{\alpha}{\sqrt{\pi}} \sum_i q_i^2 + \frac{2\pi}{(2\epsilon_r + 1)L^3} \left( \sum_i q_i \mathbf{r}_i \right)^2 \\ &= \frac{1}{2} \sum_{i,j}^\dagger \frac{q_i q_j}{|\mathbf{r}_{ij}|} \operatorname{erfc}(\alpha |\mathbf{r}_{ij}|) - \frac{\alpha}{\sqrt{\pi}} \sum_i q_i^2 + \frac{2\pi}{(2\epsilon_r + 1)L^3} \left( \sum_i q_i \mathbf{r}_i \right)^2 \end{aligned} \tag{10}$$

with the terms independent of  $\mathbf{n}$  included. The  $\dagger$  symbol indicates that the  $i = j$  terms are excluded from the daggered sum.

When  $n \neq 0$  we get

$$\begin{aligned} &\sum_{1 \leq i < j \leq N} q_i q_j \left\{ \sum_{n \neq 0} \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{\pi L} \sum_{n \neq 0} \frac{\exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij} - \pi^2 |\mathbf{n}|^2 / (\alpha L)^2\right)}{|\mathbf{n}|^2} \right\} \\ &+ \frac{1}{2L} \sum_i q_i^2 \left\{ \sum_{n \neq 0} \frac{\operatorname{erfc}(\alpha |\mathbf{n}|L)}{|\mathbf{n}|} + \frac{1}{\pi} \sum_{n \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \right\} \\ &= \frac{1}{2} \sum_{i,j} q_i q_j \sum_{n \neq 0} \left\{ \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{\pi L} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}\right) \right\}. \end{aligned} \tag{11}$$

The factor  $1/2$  in Equation (11) comes from changing the summation from  $i < j$  to all pairs  $i$  and  $j$ , and using the symmetry induced by  $\mathbf{r}_{ij} = -\mathbf{r}_{ji}$  and  $\pm \mathbf{n}$ .

By combining Equations (10) and (11) we identify the real-space term  $U_{qq}^{real}$ , the reciprocal-space term  $U_{qq}^{recip}$ , the self-interaction term  $U_{qq}^{self}$ , and the boundary-condition term  $U_{qq}^{bc}$ . The real-space term is given by

$$\begin{aligned} U_{qq}^{real} &= \sum_{1 \leq i < j \leq N} q_i q_j \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij}|)}{|\mathbf{r}_{ij}|} + \sum_{i < j, \mathbf{n} \neq 0} \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{1}{2} \sum_i q_i^2 \sum_{\mathbf{n} \neq 0} \frac{\operatorname{erfc}(\alpha |\mathbf{n}L|)}{|\mathbf{n}L|} \\ &= \frac{1}{2} \sum_{\mathbf{n}} \sum_{i, j} \frac{q_i q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|} \operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|), \end{aligned} \quad (12)$$

and the reciprocal-space term is

$$U_{qq}^{recip} = \frac{1}{\pi L} \sum_{i < j, \mathbf{n} \neq 0} q_i q_j \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}\right) + \frac{1}{2\pi L} \sum_i \sum_{\mathbf{n} \neq 0} q_i q_i \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2}. \quad (13)$$

However, with the symmetries generated by  $\mathbf{r}_{ij} = -\mathbf{r}_{ji}$  and  $\pm \mathbf{n}$ , we get

$$\begin{aligned} U_{qq}^{recip} &= \frac{1}{2\pi L} \sum_{i \neq j, \mathbf{n} \neq 0} q_i q_j \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}\right) + \frac{1}{2\pi L} \sum_i \sum_{\mathbf{n} \neq 0} q_i q_i \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \\ &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \sum_{i, j} q_i q_j \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}\right). \end{aligned} \quad (14)$$

Furthermore, we have

$$U_{qq}^{self} = \frac{\alpha}{\sqrt{\pi}} \sum_i q_i^2, \quad (15)$$

$$U_{qq}^{bc} = \frac{2\pi}{(2\epsilon_r + 1)L^3} \left( \sum_i q_i \mathbf{r}_i \right)^2, \quad (16)$$

and finally

$$U_{qq}(\epsilon_r) = U_{qq}^{real} + U_{qq}^{recip} - U_{qq}^{self} + U_{qq}^{bc}. \quad (17)$$

The reciprocal-space part, Equation (14), can be expanded in two different forms. The first form is in terms of the structure factor  $S(\mathbf{n})$ ,

$$S(\mathbf{n}) = \sum_i q_i \exp\left(-\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_i\right), \quad (18)$$

and is given by

$$\begin{aligned} U_{qq}^{recip} &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \sum_{i, j} q_i q_j \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot (\mathbf{r}_i - \mathbf{r}_j)\right) \\ &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \sum_i q_i \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \sum_j q_j \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot (-\mathbf{r}_j)\right) \\ &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} S(-\mathbf{n}) S(\mathbf{n}). \end{aligned} \quad (19)$$

Now for a fixed  $\mathbf{n}$ , the structure factor  $S(\mathbf{n})$  is just a complex number  $a$ , and the simple fact that  $a\bar{a} = \operatorname{Re}(a)^2 + \operatorname{Im}(a)^2$ , gives the real form of Equation (14)

$$\begin{aligned}
 U_{qq}^{recip} &= \frac{1}{2\pi L} \sum_{n \neq 0} \frac{e^{-\pi^2 |n|^2 / (\alpha L)^2}}{|n|^2} \sum_{i,j} q_i q_j \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot (\mathbf{r}_i - \mathbf{r}_j)\right) \\
 &= \frac{1}{2\pi L} \sum_{n \neq 0} \frac{e^{-\pi^2 |n|^2 / (\alpha L)^2}}{|n|^2} \left[ \left( \sum_i q_i \cos\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \right)^2 + \left( \sum_i q_i \sin\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \right)^2 \right].
 \end{aligned}
 \tag{20}$$

The first form in Equation (19) is used in our fast approach to calculating the reciprocal-space part. The last form in Equation (20) is the most common point-of-departure when implementing the reciprocal-space part.

### 2.3. Forces in Ewald Summation

Now that we have calculated the electrostatic energy of the system we can easily compute the electrostatic forces  $F_i$  that act on each particle  $i$ . Splitting the forces in the same way as we have split the energy and using Equation (17) we get the total electrostatic force by finding the negative of the gradient of the electrostatic energy

$$F_i = -\nabla_i U_{qq} = -\nabla_i \left[ U_{qq}^{real} + U_{qq}^{recip} - U_{qq}^{self} + U_{qq}^{bc} \right] = F_i^{real} + F_i^{recip} - 0 + F_i^{bc}. \tag{21}$$

The subscript  $i$  on the  $\nabla$  operator indicates that we take the partial derivatives with respect to the position of particle  $r_i$  and the 0 in Equation (21) comes from the self-interaction term in Equation (15) being independent of  $r_i$ . Before we do this calculation we note a couple basic, but helpful, formulas for calculating derivatives

$$\begin{aligned}
 \nabla_i \mathbf{n} \cdot \mathbf{r}_{ij} &= \nabla_i \mathbf{n} \cdot (\mathbf{r}_i - \mathbf{r}_j) = \mathbf{n} \\
 \nabla_i \frac{1}{|\mathbf{r}_{ij}|} &= -\frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^3} \\
 \nabla_i |\mathbf{r}_{ij}| &= \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} \\
 \nabla_i \operatorname{erfc}(\alpha |\mathbf{r}_{ij}|) &= -\frac{2\alpha}{\sqrt{\pi}} \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} \exp\left(-\alpha^2 |\mathbf{r}_{ij}|^2\right) \\
 \nabla_i \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}\right) &= -\mathbf{n} \frac{2\pi i}{L} \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}\right) \\
 \nabla_i \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}\right) &= +\mathbf{n} \frac{2\pi i}{L} \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij}\right)
 \end{aligned}
 \tag{22}$$

With above formulas, it is straightforward to find the different terms of  $F_i$ . The contribution from the real-space term,  $F_i^{real}$ , becomes

$$\begin{aligned}
 F_i^{real} &= -\nabla_i U_{qq}^{real} = -\nabla_i \left[ \frac{1}{2} \sum_n \sum_{i,j} \frac{q_i q_j}{|\mathbf{r}_{ij} + \mathbf{nL}|} \operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{nL}|) \right] \\
 &= \sum_n \sum_j q_i q_j \frac{\mathbf{r}_{ij} + \mathbf{nL}}{|\mathbf{r}_{ij} + \mathbf{nL}|^2} \left[ \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{nL}|)}{|\mathbf{r}_{ij} + \mathbf{nL}|} + \frac{2\alpha}{\sqrt{\pi}} \exp\left(-\alpha^2 |\mathbf{r}_{ij} + \mathbf{nL}|^2\right) \right],
 \end{aligned}
 \tag{23}$$

such that when  $n \neq 0$  include all  $j$  and otherwise only  $j \neq i$  - "the daggered saviour". Equation (20) is convenient to use when calculating the reciprocal-space contribution because it is expressed in terms of charge locations  $r_i$  rather than relative distances  $r_{ij}$ . Thus the reciprocal-space force is given by

$$\begin{aligned}
F_i^{recip} &= -\nabla_i U_{qq}^{recip} \\
&= -\nabla_i \left[ \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \left[ \left( \sum_i q_i \cos \left( \frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i \right) \right)^2 + \left( \sum_i q_i \sin \left( \frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i \right) \right)^2 \right] \right] \\
&= \frac{2q_i}{L^2} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \left[ \sin \left( \frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i \right) \operatorname{Re}(S(\mathbf{n})) + \cos \left( \frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i \right) \operatorname{Im}(S(\mathbf{n})) \right].
\end{aligned} \tag{24}$$

Finally, the contribution that depends on the boundary condition

$$F_i^{bc} = -\nabla_i \left[ \frac{2\pi}{(2\epsilon_r + 1)L^3} \left( \sum_i q_i \mathbf{r}_i \right)^2 \right] = -\frac{4\pi}{(2\epsilon_r + 1)L^3} q_i \sum_j q_j \mathbf{r}_j. \tag{25}$$

#### 2.4. Formulas for Energy and Forces in Ewald Summation

Consider a periodically replicated system, with the central box consisting of  $N$  point charges  $q_i$  and  $\sum_i q_i = 0$ . Assume that the surrounding media at the boundary of the periodically replicated system is a uniform dielectric with dielectric constant  $\epsilon_r$ . The cubic box has edge length  $L$ ; each charge  $q_i$  is located at  $\mathbf{r}_i$ , and distances are calculated with the minimum image convention. After expansion and rearrangements of Equation (2), rescaling  $\alpha \rightarrow \alpha L$  and using symmetries induced by  $\mathbf{r}_{ij} = -\mathbf{r}_{ji}$  and  $\pm \mathbf{n}$ , the total electrostatic energy of the system can be written as

$$U_{qq}(\epsilon_r) = U_{qq}^{real} + U_{qq}^{recip} - U_{qq}^{self} + U_{qq}^{bc} \tag{26}$$

with the different terms given by

$$U_{qq}^{real} = \frac{1}{2} \sum_{\mathbf{n}} \sum_{i,j} \frac{q_i q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|} \operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|), \tag{27}$$

$$\begin{aligned}
U_{qq}^{recip} &= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \sum_{i,j} q_i q_j \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \exp \left( \frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_{ij} \right) \\
&= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \left[ \left( \sum_i q_i \cos \left( \frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i \right) \right)^2 + \left( \sum_i q_i \sin \left( \frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i \right) \right)^2 \right] \\
&= \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} S(\mathbf{n}) S(-\mathbf{n}),
\end{aligned} \tag{28}$$

$$U_{qq}^{self} = \frac{\alpha}{\sqrt{\pi}} \sum_i q_i^2, \tag{29}$$

$$U_{qq}^{bc} = \frac{2\pi}{(2\epsilon_r + 1)L^3} \left( \sum_i q_i \mathbf{r}_i \right)^2. \tag{30}$$

Note that  $\alpha > 0$  is a free parameter. The structure factor  $S(\mathbf{n})$  is defined as

$$S(\mathbf{n}) = \sum_i q_i \exp \left( -\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_i \right). \tag{31}$$

The total electrostatic force,  $F_i$ , on each particle is

$$F_i = F_i^{real} + F_i^{recip} + F_i^{bc}. \tag{32}$$

Each of the force terms given by

$$F_i^{real} = \sum_n^\dagger \sum_j q_i q_j \frac{\mathbf{r}_{ij} + \mathbf{n}L}{|\mathbf{r}_{ij} + \mathbf{n}L|^2} \left[ \frac{\operatorname{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} + \frac{2\alpha}{\sqrt{\pi}} \exp(-\alpha^2 |\mathbf{r}_{ij} + \mathbf{n}L|^2) \right], \quad (33)$$

$$F_i^{recip} = \frac{2q_i}{L^2} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \left[ \sin\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \sum_j q_j \cos\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_j\right) - \cos\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \sum_j q_j \sin\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_j\right) \right], \quad (34)$$

$$F_i^{bf} = -\frac{4\pi}{(2\epsilon_r + 1)L^3} q_i \sum_j q_j \mathbf{r}_j. \quad (35)$$

The positive number  $\alpha$ , the so called Ewald convergence parameter, is chosen for computational convenience. Observe that  $\operatorname{erfc}(x) \approx e^{-x^2}$  for large values of  $x$ . By choosing  $\alpha$  large enough in Equation (27), we can ensure that the only terms that contribute in the real-space sum is when  $\mathbf{n} = 0$ . This may be expressed so that all terms with  $|\mathbf{n}| < n_{cut}$  should be included.

Choose a cut-off in both real-space and reciprocal-space so that the neglected terms in the real-space and reciprocal-space parts are of the same order  $\delta$ , or less. The truncation in real-space implies that a sufficient number of terms must be included in the reciprocal-space sums, Equation (28).

Given a required accuracy  $\delta$ ,  $n_{cut}$  is fixed by

$$e^{-\pi^2 n_{cut}^2 / (\alpha L)^2} \leq \delta \Rightarrow n_{cut} \geq \frac{\alpha L}{\pi} \sqrt{-\log(\delta)}, \quad (36)$$

and  $r_{cut}$  is determined by

$$\operatorname{erfc}(\alpha r_{cut}) \approx e^{-\alpha^2 r_{cut}^2} \leq \delta \Rightarrow r_{cut} = \frac{\pi n_{cut}}{\alpha^2 L}. \quad (37)$$

We have two conditions and four parameters. With a required  $\delta$  we may just as well pick a suitable value for  $n_{cut}$  and let the above two equations determine  $\alpha$  and  $r_{cut}$ .

With an optimal choice of parameters the computational effort of the Ewald method becomes  $\mathcal{O}(N^{3/2})$  [36] giving a considerable improvement over the  $\mathcal{O}(N^2)$  computational complexity implied by the “infinite” reach of the Coulomb interactions.

### 3. ENUF: A Fast Method for Calculating Electrostatic Interactions

In Section 2 we summarized known results and prepared the ground for the development of a fast method for Ewald summation using the discrete nonuniform fast Fourier transform (NDFT).

#### 3.1. Discrete Fourier Transforms for Non-Equispaced Data

The fast Fourier transform for nonuniform data-points (NFFT) [40] is a generalization of the FFT [41]. Several similar approaches have been proposed; some examples are [42-50] with comparisons in [47,51,52].

The basic idea of NFFT is to combine the standard FFT and linear combinations of a window function that is well localized in both the spatial domain and the frequency domain. A controlled approximation using a cut-off in the frequency domain and a limited number of terms in the spatial domain results in an aliasing error and a truncation error, respectively. The aliasing errors is controlled by the oversampling factor  $\sigma_s$ , and the truncation error is controlled by the number of terms,  $m$ , in the spatial/time approximation. For a number of window functions (Gaussian, B-spline, Sinc-power, Kaiser-Bessel), it has been shown that for a fixed oversampling factor,  $\sigma_s > 1$ , the error decays exponentially with  $m$  [53].

##### 3.1.1. Problem Definition

We wish to calculate the discrete Fourier transform for nonequispaced data. The problem can be stated as follows. For a finite number of given Fourier coefficients  $\hat{f}_k \in \mathcal{C}$  with  $k \in I_M$  we want to evaluate the trigonometric polynomial  $f(x) = \sum_{k \in I_M} \hat{f}_k \exp(-2\pi i k x)$  at each of the given nonequispaced points  $(\mathbf{x}_j \in \mathcal{D}^d, j = 0, \dots, N - 1)$ . In the literature, *points* are often called *knots*. We use the two terms synonymously.

Obviously, the details of an NDFT depend on the definitions of a sampling set for knots,  $\mathcal{D}^d$ , and an index space  $I_M$ . More in-depth discussions and further details can be found in [53,54]. The presentation that follows is mainly drawn from these sources.

3.1.2. Underlying Concepts

Consider a  $d$ -dimensional domain  $\mathcal{D}^d$  in which the set of nonequispaced knots, or data points, are located. Let

$$\mathcal{D}^d := \left\{ \mathbf{x} = (x_t)_{t=0, \dots, d-1} \in \mathcal{R}^d : -\frac{1}{2} \leq x_t < \frac{1}{2}, t = 0, \dots, d-1 \right\}, \tag{38}$$

and the set of  $N$  data points  $(\mathcal{X} := \{ \mathbf{x}_j \in \mathcal{D}^d : j = 0, \dots, N-1 \})$ . For the application we have in mind  $d$  is usually 2 or 3. Let  $\mathcal{F}$  be a function space of trigonometric polynomials with degree  $M_t (t = 0, \dots, d-1)$  in dimension  $t$ ; the function space  $\mathcal{F}$  can be defined as

$$\mathcal{F} := \left\{ f : \mathcal{D}^d \mapsto \mathcal{C} \text{ such that } f \in \text{span} \left( e^{-2\pi i \mathbf{k} \cdot \mathbf{x}} : \mathbf{k} \in I_M \right) \right\}. \tag{39}$$

The dimension of this function space is  $\dim(\mathcal{F}) = M_{\Pi}$ , where  $M_{\Pi} = \prod_{t=0}^{d-1} M_t$ . The frequencies  $\mathbf{k} \in I_M$  with the index set  $I_M$  are such that

$$I_M := \left\{ \mathbf{k} = (k_t)_{t=0, \dots, d-1} \in \mathcal{Z}^d : -\frac{M_t}{2} \leq k_t < \frac{M_t}{2}, t = 0, \dots, d-1 \right\}. \tag{40}$$

3.1.3. Matrix-Vector Formulation

With these preliminary definitions we carry on with the problem of calculating the discrete Fourier transform for nonequispaced data. For a finite number of given Fourier coefficients  $\hat{f}_{\mathbf{k}} \in \mathcal{C}$  with  $\mathbf{k} \in I_M$  we want to evaluate the trigonometric polynomial  $f(\mathbf{x}) = \sum_{\mathbf{k} \in I_M} \hat{f}_{\mathbf{k}} \exp(-2\pi i \mathbf{k} \mathbf{x})$  at each of the given nonequispaced knots in  $\mathcal{X}$ , where the product  $\mathbf{k} \mathbf{x}$  is the usual scalar product of the two vectors  $\mathbf{k}$  and  $\mathbf{x}$  as  $\mathbf{k} \mathbf{x} := k_0 x_0 + \dots + k_{d-1} x_{d-1}$ . Consequently, for each  $\mathbf{x}_j \in \mathcal{X}$ , we evaluate  $f_j = f(\mathbf{x}_j) := \sum_{\mathbf{k} \in I_M} \hat{f}_{\mathbf{k}} \exp(-2\pi i \mathbf{k} \mathbf{x}_j)$ . This may be reformulated in matrix-vector notation by setting

$$\begin{aligned} f &:= (f_j)_{j=0, \dots, N-1}, \\ \mathbf{A} &:= \left( e^{-2\pi i \mathbf{k} \mathbf{x}_j} \right)_{j=0, \dots, N-1; \mathbf{k} \in I_M}, \\ \hat{\mathbf{f}} &:= (\hat{f}_{\mathbf{k}})_{\mathbf{k} \in I_M} \end{aligned} \tag{41}$$

and writing  $f = \mathbf{A} \hat{\mathbf{f}}$ .

3.1.4. Related Matrix-Vector Products

A number of related NDFT matrix-vector products can also be defined. To write them down we let  $\bar{A}$  be the complex conjugate of the elements of the matrix  $A$  and  $A^H = \bar{A}^T$  the transposed complex conjugate of the matrix  $A$ . Using these conventions we can name and summarize the related NDFT matrix-vector products and their component representation as

$$\text{Regular : } f = \mathbf{A} \hat{\mathbf{f}}, f_j = \sum_{\mathbf{k} \in I_M} \hat{f}_{\mathbf{k}} \exp(-2\pi i \mathbf{k} \mathbf{x}_j). \tag{42}$$

$$\text{Adjoint : } \hat{\mathbf{f}} = \mathbf{A}^H f, \hat{f}_{\mathbf{k}} = \sum_{j=0}^{N-1} \hat{f}_j \exp(2\pi i \mathbf{k} \mathbf{x}_j). \tag{43}$$

$$\text{Conjugated : } f = \bar{\mathbf{A}} \hat{\mathbf{f}}, f_j = \sum_{\mathbf{k} \in I_M} \hat{f}_{\mathbf{k}} \exp(2\pi i \mathbf{k} \mathbf{x}_j). \tag{44}$$

$$\text{Transposed : } \hat{\mathbf{f}} = \mathbf{A}^T f, \hat{f}_{\mathbf{k}} = \sum_{j=0}^{N-1} \hat{f}_j \exp(-2\pi i \mathbf{k} \mathbf{x}_j). \tag{45}$$

### 3.1.5. NDFT, FFT and NFFT

From the different NDFT products written in matrix-vector form, as in Equations (42)-(45), it is clear that it takes  $\mathcal{O}(NM_{\Pi})$  arithmetic operations to transform between the Fourier-samples and the Fourier-coefficients.

This is simply because the matrix  $A$  is  $N \times M_{\Pi}$ , with  $M_{\Pi} = \dim(\mathcal{F}) = \prod_{t=0}^{d-1} M_t$ .

However, for the special case of  $M_t = M_{(t=0, \dots, d-1)}$  and  $N = M^d$  equispaced knots  $\mathbf{x}_k (k \in I_M)$ , the Fourier-samples  $f_k$  can be calculated from the Fourier-coefficients  $\hat{f}_k$  by the fast Fourier transform (FFT) with  $\mathcal{O}(N \log N)$  arithmetic operations.

The fast Fourier transform for nonequispaced knots (NFFT) is a generalization of the FFT. The essential idea is that of combining a window function with the standard FFT. The window function is a well localized function in both the space domain and frequency domain. Several different window functions and similar approaches have been proposed. The resulting algorithms are approximate and some of them have been shown to have a computational complexity of  $\mathcal{O}(M_{\Pi} \log M_{\Pi} + \log(1/\epsilon)N)$ , where  $\epsilon$  is the desired accuracy [53].

## 3.2. Fast Ewald Summation

Using optimal parameters in the Ewald summation method implies that the time to calculate the real-space part and the reciprocal-space part are approximately equal. As the number of particles in the system grows we would like to combine the calculation of the short-range part of the potential with the real-space part. This implies that we need to choose a real-space cut-off about the same size as the short-range cut-off. With this nonoptimal choice, the reciprocal-space parts of the Ewald summation method become the most time-consuming to calculate [55].

To show how a fast Ewald summation approach may be obtained from the regular Ewald method, described in Section 2, we focus on the reciprocal-space parts. In Section 3.1, we gave the details of the discrete Fourier transform (DFT) for data that is nonuniformly spaced (NDFT). Based on these definitions we get a number of useful algorithmic primitives. First we reformulate the reciprocal-space part of the regular Ewald method in terms of the NDFT primitives. Then we show how the fast Fourier transform for nonequispaced (NFFT) can be applied, yielding an Ewald method based on the nonuniform fast Fourier transform.

### 3.2.1. Reciprocal Space Terms as DFT

We apply the generalized DFT, described in Section 3.1, to the calculation of the reciprocal-space energy and forces. This allows us to formulate the standard Ewald method for calculating the reciprocal energy and forces in terms of the NDFT primitives.

### 3.2.2. Reciprocal Energy

In the case of the electrostatic energy we have from Equation (28)

$$U_{qq}^{recip} = \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} S(\mathbf{n}) S(-\mathbf{n}), \quad (46)$$

with the structure factor  $S(\mathbf{n})$  defined as

$$S(\mathbf{n}) = \sum_j q_j \exp\left(-\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_j\right). \quad (47)$$

By comparing the definition of the transposed NDFT in Equation (45) and the structure factor in Equation (47) we note that they have the same structure; after a renumbering of the location indexes, the summation limits are also the same. In fact, by setting the normalized locations,  $\mathbf{x}_j = \mathbf{r}_j/L$ , and the samples,  $\hat{f}_j = q_j$ , we see by inspection that Equation (47) is a 3D instance of Equation (45) with  $\hat{f}_{\mathbf{n}} = S(\mathbf{n})$ . Furthermore, assuming that the MD simulation box is centered around the origin, the normalized locations can be assumed to be in the domain  $\mathcal{D}^d$  as defined in Equation (38).

Consequently, we can use the NDFT approach to calculate each of the components of the structure factor. From a computational point of view this means that we can also expect to utilize an NFFT based algorithm to calculate the components of the structure factor  $S(\mathbf{n})$ , rather than the straightforward summation normally used in the Ewald method.



Recasting Equation (46) in terms of Fourier-components

$$U_{qq}^{recip} = \frac{1}{2\pi L} \sum_{\mathbf{n} \neq 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} S(\mathbf{n}) S(-\mathbf{n}) = \frac{1}{\pi L} \sum_{n_z > 0} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} |\hat{\mathbf{f}}_{\mathbf{n}}|^2. \quad (48)$$

Calculating the energy,  $U_{qq}^{recip}$ , using Equation (48) means that we

- 1) calculate all  $\hat{\mathbf{f}}_{\mathbf{n}}$  using the transposed NDFT,
- 2) scale each  $|\hat{\mathbf{f}}_{\mathbf{n}}|^2$  with a factor given by Equation (48),
- 3) sum all the scaled components.

### 3.2.3. Reciprocal Forces

We calculate the contribution from the reciprocal-space forces using a similar approach as for the energy. In the formula below,  $\text{Re}(\cdot)$ , and  $\text{Im}(\cdot)$ , denote the real and imaginary part of the arguments, respectively. From Equation (24) we have that

$$F_i^{recip} = \frac{2q_i}{L^2} \sum_{\mathbf{n} \neq 0} \mathbf{n} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \left[ \sin\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \text{Re}(S(\mathbf{n})) + \cos\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \text{Im}(S(\mathbf{n})) \right]. \quad (49)$$

Now, the structure factor  $S(\mathbf{n})$  is just a complex number so the expression in the brackets in Equation (49) can be written as the imaginary part of a product

$$\sin\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \text{Re}(S(\mathbf{n})) + \cos\left(\frac{2\pi}{L} \mathbf{n} \cdot \mathbf{r}_i\right) \text{Im}(S(\mathbf{n})) = \text{Im} \left[ \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_i\right) S(\mathbf{n}) \right]. \quad (50)$$

Inserting Equation (50) this into Equation (49) gives

$$\begin{aligned} F_i^{recip} &= \frac{2q_i}{L^2} \sum_{\mathbf{n} \neq 0} \mathbf{n} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \text{Im} \left[ \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_i\right) S(\mathbf{n}) \right] \\ &= \text{Im} \left[ \frac{2q_i}{L^2} \sum_{\mathbf{n} \neq 0} \mathbf{n} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} \exp\left(\frac{2\pi i}{L} \mathbf{n} \cdot \mathbf{r}_i\right) S(\mathbf{n}) \right]. \end{aligned} \quad (51)$$

Note that Equation (51) is a vector equation. Furthermore, each of the three components has the same structure as the conjugated NDFT of Equation (44). By setting the normalized locations,  $\mathbf{x}_j = \mathbf{r}_j / L$ , and the samples,

$$\hat{\mathbf{g}}_{\mathbf{n}} = \mathbf{n} \frac{e^{-\pi^2 |\mathbf{n}|^2 / (\alpha L)^2}}{|\mathbf{n}|^2} S(\mathbf{n}) \quad \text{for } \mathbf{n} \neq 0, \quad (52)$$

we see, again, by inspection that each component of Equation (51) is a 3D instance of Equation (44). Assuming that  $\mathbf{n}$  is in the index set  $I_M$  of Equation (40),  $\mathbf{n} \in I_M$ , and setting  $\mathbf{g}_0 = 0$ , we can formulate  $F_i^{recip}$  directly in Fourier-terms

$$F_i^{recip} = \text{Im} \left[ \frac{2q_i}{L^2} \sum_{\mathbf{n} \in I_M} \hat{\mathbf{g}}_{\mathbf{n}} e^{2\pi i \mathbf{n} \cdot \mathbf{x}_i} \right] = \frac{2q_i}{L^2} \text{Im}(\mathbf{g}_i). \quad (53)$$

Calculating the reciprocal-space force  $F_i^{recip}$  on particle  $i$ , using Equation (53), means that we

- 1) start with the structure factor components,  $S(\mathbf{n})$ , already obtained when we calculated  $U_{qq}^{recip}$ ,
- 2) scale each  $S(\mathbf{n})$  using Equation (52),
- 3) giving a new set of Fourier-coefficients that are transformed back to real-space, via Equation (53), using the conjugated NDFT, and finally

- 4) with Equation (53), taking the imaginary part of coefficient  $g_i$  and scaling it with  $\frac{2q_i}{L^2}$  gives the

reciprocal force on particle  $i$ .

Thus we can use the NDFT approach to calculate each of the components of the reciprocal forces. Again, from a computational point of view this means we can expect to utilize an NFFT based algorithm to find the respective components.

### 3.2.4. Combining the Ewald Method and NFFT

The reformulation of  $U_{qq}^{recip}$  and  $F_i^{recip}$  as in Equations (48) and (53) shows the central role of the transposed and conjugated NDFT in calculating the reciprocal-space energy and forces. Starting with the location of the charged particles, the structure factor  $S(\mathbf{n})$  is calculated via a transposed NDFT. In the language of Ewald summation, we transform from real-space to reciprocal-space. Scaling the absolute value of the Fourier-components and summing gives  $U_{qq}^{recip}$ . To find  $F_i^{recip}$  we go back from the reciprocal-space to the real-space by first calculating the Fourier-components of the forces and then performing a conjugated NDFT.

An implementation of Ewald summation uses cut-offs, in reciprocal-space,  $n_{cut}$ , and real-space,  $r_{cut}$ ; with  $\alpha$  large enough and with a required accuracy,  $\delta$ , truncate the sums Equation (27) and Equation (28) at the respective cut-offs so that the last term added  $\leq \delta$ , in each of the sums. When  $n_{cut}$  is fixed by

$$e^{-\pi^2|\mathbf{n}|^2/(\alpha L)^2} \leq \delta \Rightarrow n_{cut} \geq \frac{\alpha L}{\pi} \sqrt{-\log(\delta)} \Rightarrow n_{cut} \propto L \propto N^{1/3}. \tag{54}$$

Then  $r_{cut}$  is determined by

$$\text{erfc}(\alpha r_{cut}) \approx e^{-\alpha^2 r_{cut}^2} \leq \delta \Rightarrow r_{cut} = \frac{\pi n_{cut}}{\alpha^2 L}. \tag{55}$$

We have recast  $U_{qq}^{recip}$  and  $F_i^{recip}$  in terms of Fourier-components and set  $M_t = 2\sigma_s n_{cut}$ , where  $\sigma_s$  is the oversampling factor. This gives  $M_{\Pi} = \prod_{t=0}^{d-1} M_t$ . In general the computational complexity of the NFFT method is  $\mathcal{O}(M_{\Pi} \log M_{\Pi} + \log(1/\epsilon)N)$ , where  $\epsilon$  is the desired accuracy in the approximation used within NFFT [53].

Using Equation (54) and the above definition of  $M_{\Pi}$ , we see that the complexity becomes  $\mathcal{O}(N \log N + \log(1/\epsilon)N)$ . Note that  $\epsilon$  is a function of  $m$ , for a fixed oversampling factor. With a controlled approximation of the structure factor via the use of nonuniform fast Fourier transform, the original computational complexity of  $\mathcal{O}(N^{3/2})$  becomes  $\mathcal{O}(N \log N)$ .

At this stage, the path to a fast Ewald method should now be clear. By specifying an accuracy  $\delta$ , we replace the transposed and conjugated NDFT with the corresponding operations using the NFFT algorithm. Thus Equations (48) and (53) become a concise procedure to calculate approximations of  $U_{qq}^{recip}$  and  $F_i^{recip}$ . Most of the mathematical details can be kept separate and hidden in a set of library routines and the remaining formulas pertain to the physics of the problem. Furthermore, with a library implementation based on a state-of-the-art FFT-library, we have good reason to expect it to be efficient.

### 3.2.5. Implementation and Results

In a first implementation [56,57], we used the libraries FFTW [58] and NFFT [54]. Details of the accuracy and scaling properties can be found in reference papers.

Basing the implementation on libraries has a number of advantages. It makes the implementation task easier and introduces a convenient division of labor in the program code: the mathematical aspects are mainly concentrated to the libraries while the physical aspects of the problem remain. Also, since the code becomes quite compact without becoming convoluted, it becomes easier to check, understand, and explain. Improvements and optimizations of the libraries can be easily included in the program, usually by just relinking the program. For example, the customization of the window function used in the NFFT algorithm---Gaussian functions, dilated cardinal B-splines, Sinc functions, or Kaiser-Bessel functions---is currently achieved by recompiling the NFFT library and relinking the application. Due to the comparatively small size of the NFFT library this is very quick. Furthermore, improvements in either theory or implementation of the used libraries will be easily accessible.

In summary, we claim that the ENUF method is

- efficient and concise, and
- has a clear separation of concerns between mathematical and physical details.

In a sense it can be said that we get the best of two worlds: a concise and efficient algorithm. The separation

of mathematical concerns is a bonus that has the potential to simplify implementation and further developments due to the fact that they may occur independently of each-other.

#### 4. ENUF Implemented in M.DynaMix

M.DynaMix is a highly modular general purpose parallel Molecular Dynamics code for simulations of arbitrary mixtures of either rigid or flexible molecules. It was released by Lyubartsev and Laaksonen in late 90's [59]. Most common force fields can be used in simulations with a variety of periodic boundary conditions (cubic, rectangular, hexagonal or a truncated octahedron). Quantum corrections to the atomic motion can be done using the Path Integral Molecular Dynamics approach. M.DynaMix has been used in applications from materials design to biological processes.

M.DynaMix deals with particle system interacting by a force field consisting of Lennard-Jones, electrostatic, covalent bonds, angles and torsion angles potentials as well as of some optional terms, in a periodic rectangular, hexagonal or truncated octahedron cell. Rigid bonds are constrained by the SHAKE algorithm [60]. In case of flexible molecular models the double time step algorithm is used [61]. Algorithms for NVE, NVT and NPT statistical ensembles are implemented, as well as Ewald sum approach for treatment of the electrostatic interactions. An option to calculate free energy by the expanded ensemble method with Wang-Landau optimization of the balancing factor is included in later versions. For its features and capabilities, M.DynaMix is diffused in a large modeling and simulation community. Written in FORTRAN 77, it can be run on a wide variety of hardware architectures both in sequential and parallel execution. The entire program source code consists of a number of FORTRAN files (modules), made of blocks of subroutines, performing different tasks or groups of tasks.

##### 4.1. Framework Overview

The situation outlined for M.DynaMix is very common in the field of computational science: a package created in late 90's that is still at work in spite of new innovations in computing platforms. In a situation like this, each upgrade requires a trade-off between the need to preserve the existing structure and the wish to obtain as much efficiency as possible.

To take the best advantage from latest programming techniques and tools, we decide to use C language to create some of the new code segments. On one hand, modules in FORTRAN and C can easily coexist in the same code, just taking account of a few mild guidelines [62]; on the other hand, C language provides several enhancements related to intrinsic features (dynamic memory allocation and direct pointers reference) and possibility to set up complex data structures; furthermore, it allows a more plain access to a large amount of external routine libraries written in C. From a general point of view, M.DynaMix code has been projected with a good modularity degree and has been possible to make the most part of upgrades just switching a block (subroutine) with a new one.

##### 4.2. Input Parameters

To use ENUF method for treating long range interactions, M.DynaMix user needs to set the proper key string in the input file [63]; parameters to specify are the Ewald convergence factor  $\alpha$  and the number of points for FFT grid in  $x$ ,  $y$  and  $z$  direction; starting from them, the program automatically sets proper values for over-sampling factor  $\sigma_x$  and approximation parameter  $p$ .

##### 4.3. Implementation

Ewald-like methods for computing electrostatic energies typically replace part of the summation in real space with an equivalent summation in Fourier space; among M.DynaMix modules, there is one devoted to this reciprocal space duty. In the starting setup, this module was present in M.DynaMix in a version related to the full Ewald algorithm; ENUF implementation involved the creation of a second instance of this module.

A scheme of the ENUF module as implemented in M.DynaMix is presented in **Table 1**; the module in this case is a *group* of files, part of them written in FORTRAN and part in C; each file contains routines related to the algorithm steps. The table displays for each step the programming language used to write corresponding file; Some files make calls to external libraries, typically used to perform non uniform Fourier transform and its inverse.

**Table 1.** ENUF algorithm scheme: for each step is listed the language of the related routines and the presence of calls to external libs.

Step	Language	Lib Calls
1: Coordinate scaling	F77	No
2: <i>Adjoint</i> FFT	C	Yes
3: Energy evaluation	F77	No
4: Factor rescaling	F77	No
5: <i>Regular</i> FFT	C	Yes
6: Forces evaluation	F77	No

#### 4.3.1. Coordinate Scaling and Adjoint Transform

In the first step of the algorithm, particle coordinates are rescaled respect to the simulation box, in order to obtain the proper set of nonequispaced knots as required in Equation (38). Knots are stored in the unidimensional FORTRAN array  $X$ , in consecutive way, saving coordinates in reverse component order  $\{z, y, x\}$  for each knot. Particle charges are located in unidimensional array  $Q$ .

At the next step, a C routine performs the 3-dim *adjoint* transform of charges  $Q$  located at knots  $X$ . C routine access to 1-dim FORTRAN vectors  $X$  and  $Q$  as three dimensional arrays; the reverse component order adopted in  $X$  vector filling is related to this and depends on the different way to arrange multi-dimensional arrays in memory: by columns for FORTRAN, by rows for C. Transformed data are stored in the *complex* array  $\tilde{S}$ .

#### 4.3.2. Energy, Regular Transform and Forces

In this stage, energy is evaluated by a sum, according to formula (48). Then, the transformed array  $\tilde{S}$  is rescaled using Equation (51); starting from  $\tilde{S}$ , three complex rescaled arrays are created,  $\tilde{S}_x$ ,  $\tilde{S}_y$  and  $\tilde{S}_z$ , one for each direction in the reciprocal space. A C routine back transform complex data set  $\tilde{S}_x$ ,  $\tilde{S}_y$  and  $\tilde{S}_z$  related to the same set of knots  $X$ . In this step, three independent back transformations are performed, producing complex arrays  $F_x$ ,  $F_y$  and  $F_z$ . Forces contributions for each particle are obtained according to Equation (53); components in  $x$ ,  $y$  and  $z$  directions are evaluated using the imaginary part of arrays  $F_x$ ,  $F_y$  and  $F_z$ .

#### 4.4. External Libraries

To perform Fourier transform on nonequispaced data, our current implementation make use of an external existing function library. Among the available resources, we select the library NFFT [54]. NFFT is a widely diffused C subroutine library for computing nonequispaced discrete Fourier transform in one or more dimensions, of arbitrary input size and of complex data; it is based on FFTW [58].

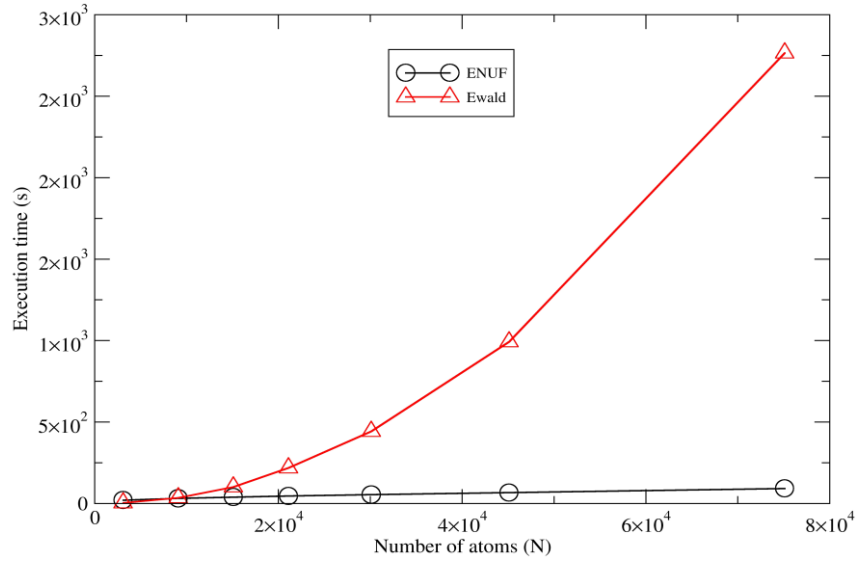
#### 4.5. Validation

In the original version of M.DynaMix the working method for the treatment of long range interactions is the full direct Ewald summation. Using a *cross-check* approach, namely comparing ENUF method to the stable one, it has been possible to debug every step of the algorithm implementation and check its correctness in deep way.

After the implementation, the same cross-check mechanism has been used to investigate ENUF precision and efficiency. **Figure 1** displays the execution time spent for long range interactions evaluation respect to the number of particles  $N$ , when full Ewald and ENUF are used. In the simulation, of 100 iterations, the system is composed by 50  $\text{Na}^+$  ions and 50  $\text{Cl}^-$  ions in water solution in a cubic cell. The number of water molecules is increased from  $10^3$  to  $25 \times 10^3$  in seven discrete steps, keeping constant the density to  $1.02 \text{ g/cm}^3$ . The simulation run on a Intel Xeon workstation, 1.86 GHz, with ram 8 Gb.

The full Ewald method has been taken as reference; for each system size, ENUF parameters have been tuned to obtain the same Ewald precision, evaluated in terms of the electrostatics energy contribution, tolerating a maximum divergence of 2%.

The plot shows that execution time for long range interactions that represents the efficiency bottleneck in the original M.DynaMix version, has been drastically reduced saving the same numerical precision. The plot also shows the different trend for execution time vs  $N$  in the two methods; the point set for the full Ewald scheme



**Figure 1.** Execution time for long range interactions vs. number of particles  $N$ , for Ewald and ENUF. System is composed by 50  $\text{Na}^+$  and 50  $\text{Cl}^-$  in  $\text{H}_2\text{O}$  solution in a periodic cubic cell. Number of water molecules is increased from  $10^3$  to  $25 \times 10^3$  keeping density to  $1.02 \text{ g/cm}^3$ .

is well fitted using the well-known theoretical behavior  $\mathcal{O}(N^2)$ ; data points related to ENUF correspond quite well to the expected  $\mathcal{O}(N \log N)$ .

### 5. Implementation of ENUF Method in Dissipative Particle Dynamics Scheme

The Dissipative Particle Dynamics (DPD) simulation method, originally proposed by Hoogerbrugge and Koelman, is a particle-based simulation method to simulate hydrodynamic phenomena at mesoscopic level [64, 65,66]. In DPD model, several atoms or molecules are grouped together to form coarse-grained particles. The interactions between any pair of DPD particles  $i$  and  $j$  are normally composed of three pairwise additive forces: the conservative force  $\mathbf{F}_{ij}^{C,DPD}$ , the dissipative force  $\mathbf{F}_{ij}^{D,DPD}$ , and the random force  $\mathbf{F}_{ij}^{R,DPD}$

$$\mathbf{f}_{ij} = \sum_{i \neq j} (\mathbf{F}_{ij}^{C,DPD} + \mathbf{F}_{ij}^{D,DPD} + \mathbf{F}_{ij}^{R,DPD}), \quad (56)$$

with

$$\mathbf{F}_{ij}^{C,DPD} = \alpha_{ij} \omega^C(r_{ij}) \hat{\mathbf{r}}_{ij}, \quad (57)$$

$$\mathbf{F}_{ij}^{D,DPD} = -\gamma \omega^D(r_{ij}) (\mathbf{v}_{ij} \cdot \hat{\mathbf{r}}_{ij}) \hat{\mathbf{r}}_{ij}, \quad (58)$$

$$\mathbf{F}_{ij}^{R,DPD} = \sigma \omega^R(r_{ij}) \theta_{ij} \hat{\mathbf{r}}_{ij}, \quad (59)$$

where  $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ ,  $r_{ij} = |\mathbf{r}_{ij}|$ ,  $\hat{\mathbf{r}}_{ij} = \mathbf{r}_{ij}/r_{ij}$ , and  $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ . The parameters  $\alpha_{ij}$ ,  $\gamma$ , and  $\sigma$  determine the strength of the conservative, dissipative, and random forces, respectively.  $\theta_{ij}$  is a randomly fluctuating variable, with zero mean and unit variance.

The pairwise conservative force is written in terms of a weight function  $\omega^C(r_{ij})$ , where  $\omega^C(r_{ij}) = 1 - r_{ij}/R_c$  is usually chosen for  $r_{ij} \leq R_c$  and  $\omega^C(r_{ij}) = 0$  for  $r_{ij} > R_c$  such that the conservative force is soft and repulsive. The unit of length  $R_c$  is related to the volume of DPD particles. Two weight functions  $\omega^D(r_{ij})$  and  $\omega^R(r_{ij})$  for dissipative and random forces, respectively, are coupled together through the fluctuation-dissipation theorem

$$\omega^D(r) = [\omega^R(r)]^2 \quad \text{and} \quad \sigma^2 = 2\gamma k_B T \quad (60)$$

to form a thermostat and generate naturally the canonical distribution (constant number of particles,  $N$ , volume,  $V$ , and temperature,  $T$ ) [67]. In most applications, the weight function  $\omega^D(r)$  adopts a simple form as [68]

$$\omega^D(r) = [\omega^R(r)]^2 = \begin{cases} \left(1 - \frac{r}{R_c}\right)^2 & (r \leq R_c) \\ 0 & (r > R_c) \end{cases} \tag{61}$$

In the original DPD model, one critical advantage is the soft repulsive nature of conservative potential, which enables us to integrate the equations of motion with large time step. However, such advantage restricts the direct incorporation of electrostatic interactions in DPD model. The main problem is that dissipative particles carrying opposite point charges tend to collapse onto each other, forming artificial ionic clusters due to the stronger electrostatic interactions than soft repulsive conservative interactions.

In order to avoid such unphysical phenomena, point charges at the center of dissipative particles are usually replaced by charge density distributions meshed around particles to remove the divergency of electrostatic interactions at  $r = 0$  [69,70]. In our implementation [71,72], a Slater-type charge density distribution is considered with the form of

$$\rho_e(r) = \frac{q}{\pi\lambda_e^3} e^{-\frac{2r}{\lambda_e}}, \tag{62}$$

in which  $\lambda_e$  is the decay length. The integration of Equation (62) over the whole space gives the total charge  $q$ .

The electrostatic potential  $\phi(r)$  generated by Slater-type charge distribution  $\rho_e(r)$  can be obtained by solving Poisson's equation

$$\nabla^2\phi(r) = -\frac{1}{\epsilon_0\epsilon_r} \rho_e(r), \tag{63}$$

in which the variables  $\epsilon_0$  and  $\epsilon_r$  are the permittivity of vacuum space and the dielectric constant of water at room temperature, respectively. In spherical coordinates, the Poisson's equation becomes

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial}{\partial r} \phi(r) \right) = -\frac{1}{\epsilon_0\epsilon_r} \rho_e(r). \tag{64}$$

By solving Poisson's equation, the electrostatic potential field  $\phi(r)$  can be analytically expressed by

$$\phi(r) = \frac{1}{4\pi\epsilon_0\epsilon_r} \frac{q}{r} \left( 1 - \left( 1 + \frac{r}{\lambda_e} \right) e^{-\frac{2r}{\lambda_e}} \right). \tag{65}$$

The electrostatic energy between two interacting charge density distributions  $i$  and  $j$  is the product of the charge density distribution  $i$  and the electrostatic potential generated by charge distribution  $j$  at position  $r_i$

$$U_{qq}^{E,DPD}(r_{ij}) = q_i\phi_j(r_i) = \frac{1}{4\pi\epsilon_0\epsilon_r} \frac{q_iq_j}{r_{ij}} \left( 1 - \left( 1 + \frac{r_{ij}}{\lambda_e} \right) e^{-\frac{2r_{ij}}{\lambda_e}} \right). \tag{66}$$

The electrostatic force on charge distribution  $i$  is the negative of the derivative of the potential energy  $U_{qq}^{E,DPD}$  respect to its position  $r_i$

$$\begin{aligned} F_i^{E,DPD}(r_{ij}) &= -\nabla_i U_{qq}^{E,DPD}(r_{ij}) \\ &= \frac{1}{4\pi\epsilon_0\epsilon_r} \frac{q_iq_j}{(r_{ij})^2} \left\{ 1 - e^{-\frac{2r_{ij}}{\lambda_e}} - \frac{2r_{ij}}{\lambda_e} e^{-\frac{2r_{ij}}{\lambda_e}} - \frac{2r_{ij}^2}{\lambda_e^2} e^{-\frac{2r_{ij}}{\lambda_e}} \right\}. \end{aligned} \tag{67}$$

By defining parameter  $r^* = r/R_c$  as the reduced center-to-center distance between two charge distributions and dimensionless parameter  $\beta = R_c/\lambda_e$ , respectively, the reduced electrostatic energy and force between two Slater-type charge distributions are given by

$$U_{qq}^{E,DPD}(r_{ij}^*) = \frac{1}{4\pi\epsilon_0\epsilon_r} \frac{q_i q_j}{(R_c r_{ij}^*)^1} \left\{ 1 - (1 + \beta r_{ij}^*) e^{-2\beta r_{ij}^*} \right\}, \quad (68)$$

$$\mathbf{F}_i^{E,DPD}(r_{ij}^*) = \frac{1}{4\pi\epsilon_0\epsilon_r} \frac{q_i q_j}{(R_c r_{ij}^*)^2} \left\{ 1 - (1 + 2\beta r_{ij}^* (1 + \beta r_{ij}^*)) e^{-2\beta r_{ij}^*} \right\}. \quad (69)$$

Comparing to the electrostatic energy and force between point charges in atomistic simulations in previous section, we can find that the electrostatic energy and force between two Slater-type charge distributions in DPD simulations are scaled with corresponding correction factors as

$$B_U = 1 - (1 + \beta r^*) e^{-2\beta r^*}, \quad (70)$$

$$B_F = 1 - (1 + 2\beta r^* (1 + \beta r^*)) e^{-2\beta r^*}. \quad (71)$$

The similarities between electrostatic energy and force between point charges and counterparts between charge density distributions imply that once we get electrostatic energy and force between point charges, from which the electrostatic energy and force between Slater-type charge density distributions in DPD simulations can be directly rescaled with corresponding correction factors.

For the electrostatic energy and force between point charges, we know that both of them do diverge when the relative distance between point charges is close to 0. While for the electrostatic energy and force between charge density distributions, in the limit of  $r_{ij}^* \rightarrow 0$ , the reduced electrostatic energy and force between charge density distributions are, respectively, described by

$$\lim_{r_{ij}^* \rightarrow 0} U_{qq}^{E,DPD}(r_{ij}^*) = \frac{1}{4\pi\epsilon_0\epsilon_r} \frac{q_i q_j}{R_c} \beta, \quad (72)$$

$$\lim_{r_{ij}^* \rightarrow 0} \mathbf{F}_i^{E,DPD}(r_{ij}^*) = 0. \quad (73)$$

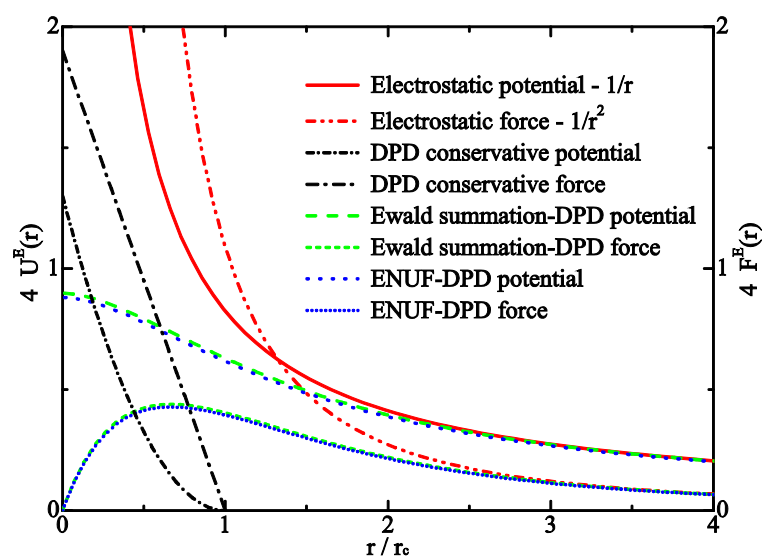
It is clear that the adoption of Slater-type charge distribution in DPD simulations removes the divergence of electrostatic interactions at  $r_{ij}^* = 0$ , which means that both electrostatic energy and force between charge distributions are characterized with finite quantities.

By matching the maximum electrostatic energy between charge distributions at  $r_{ij}^* = 0$  with Groot's previous work [69] gives  $\beta = 1.125$ . In our detailed implementations, we adopted a particular coarse-graining scheme [73] with  $N_m = 4$  and  $\rho = 4$ , in which the former parameter means 4 water molecules being coarse-grained into one DPD particle and the latter means there are 4 DPD particles in the volume of  $R_c^3$ . With this particular scheme, the length unit  $R_c$  is given as  $R_c = 3.107\sqrt[3]{\rho N_m} = 7.829\text{\AA}$ . From the relation of  $\beta = R_c/\lambda_e$ , we can get  $\lambda_e = 6.954\text{\AA}$ , which is consistent with the electrostatic smearing radii used in González-Melchor's work [70].

**Figure 2** shows the representation of reduced electrostatic energy and force with respect to the relative distance between Slater-type charge distributions. For a better comparison, we also include the typical soft conservative potential and force between dissipative particles, as well as the standard Coulombic potential and force between point charges, both of which do diverge at  $r=0$ . In short range length scale, both the electrostatic energy and force are characterized with finite quantities, which attribute to the adoption of Slater-type charge distributions instead of point charges. In long range length scale, both the electrostatic energy and force are consistent with counterparts between point charges, which implies that the ENUF-DPD method can capture essential characteristics of electrostatic interactions at mesoscopic level.

Combining electrostatic force  $\mathbf{F}_i^{E,DPD}$  and soft repulsive conservative force  $\mathbf{F}_i^{C,DPD}$  gives the total conservative force  $\mathbf{F}_i^{C*}$  on particles  $i$  in DPD simulations. The total conservative force  $\mathbf{F}_i^{C*}$ , together with dissipative force  $\mathbf{F}_i^{D,DPD}$  and random force  $\mathbf{F}_i^{R,DPD}$ , as well as the intramolecular bonding force  $\mathbf{F}_i^{S,DPD}$  for polymers and surfactants, act on dissipative particles and evolve the whole simulated system toward equilibrium conditions before taking statistical analysis.

As the number of charged DPD particles in the simulated system grows, the calculation of the reciprocal space electrostatic interactions will become the most time-consuming part. Using suitable parameters in



**Figure 2.** Electrostatic potential and force between two charge density distributions in DPD scheme calculated from the ENUF and Ewald summation methods with reference parameters. For a better comparison, the standard Coulombic potential and force, both of which diverge at  $r=0$ , and the typical conservative potential and force in standard DPD method are also included. Both the electrostatic potential and force expressions are plotted for two equal sign charge distributions.

ENUF-DPD method assures that the time to calculate the real space summations is approximately the same as the time to calculate the reciprocal space summations, thereby reducing the total computational time. Herein we try to explore the ENUF-DPD related parameters and get a set of suitable parameters for further applications.

As addressed in Section 3.2, the implementation of ENUF-DPD method uses the Ewald convergence parameter  $\alpha$ , required accuracy  $\delta (\ll 1)$ , and two cut-offs ( $r_{cut}$  for real space and  $n_{cut}$  for reciprocal space summations). These parameters are correlated with each other through two conditions shown in Equations (54) and (55). With required accuracy parameter  $\delta = 1.0 \times 10^{-4}$ , it is more convenient to pick a suitable value for  $n_{cut}$ . Then one can determine  $\alpha$  and  $r_{cut}$  directly from Equations (54). However, due to the fact that  $n_{cut}$  should be integer and  $r_{cut}$  should be a suitable value for the cell-link list update scheme in DPD simulations, we adopt another procedure to get suitable parameters.

During the calculation of real space summation and self-interaction parts of electrostatic energy and force between point charges, we can directly multiply corresponding correction factors to get the electrostatic energy and force between Slater-type charge density distributions. However, it is not accessible for the calculation of reciprocal space summation since we cannot rescale the NFFT transformation results. But if we choose suitable real space cutoff  $r_{cut}$ , beyond which two correction factors  $B_U$  and  $B_F$  converge to unit, one can directly adopt NFFT transformation result without any corrections. We find that electrostatic energy and force between charge density distributions are consistent with counterparts between point charges when the relative distance between two distributions is larger than  $3.0R_c$ . Hence in our simulations,  $r_{cut} = 3.0R_c$  is taken as the cutoff for real space summations of electrostatic interactions.

By evaluating the Madelung constant of a face-centered cubic (FCC) lattice, we adopt the Ewald convergence parameter with the value of  $\alpha = 0.20 \text{ \AA}^{-1}$ , which can generate accurate Madelung constant for FCC lattice structure and keep considerable accuracy. Then we perform coarse-grained simulations on bulk electrolyte system to explore suitable values for  $n_{cut}$ , and approximation parameter  $p$  in NFFT. It is specified that approximation parameter  $p = 2$  and cutoff  $n_{cut} = 7$  for reciprocal space summations can generate consistent electrostatic energy and force in comparison with those obtained from traditional Ewald summation method with reference parameters. Larger  $p$  values can further increase the accuracy of electrostatic interactions in ENUF-DPD method, but also the total computational time in treating electrostatic interactions increases. By compromising the accuracy and computational speed in the ENUF-DPD method, we adopt  $p = 2$  and  $n_{cut} = 7$  in following simulations.

With the set of explored optimized parameters, we address the computational complexity of ENUF-DPD method in treating electrostatic interactions. The computational complexity of ENUF-DPD method is appro-



ximately described as  $\mathcal{O}(N \log N)$ , which shows remarkably better computational efficiency than the traditional Ewald summation method with acceptable accuracy in treating long-range electrostatic interactions between charged particles at mesoscopic level.

The ENUF-DPD method is then validated by investigating the influence of charge fraction of polyelectrolyte on corresponding conformational properties [71]. With the increase of charge fraction on polyelectrolyte, both the intramolecular correlations between charged beads on polyelectrolyte chain and the intermolecular correlations between charged beads on polyelectrolyte and counterions are enhanced. The conformation transition of polyelectrolyte chain from collapsed state to fully extended conformation can be visualized from simulations. Meanwhile, the dependence of the conformations of fully ionized polyelectrolyte on charge valency and concentration of added salts are also studied in details. Counterions with larger valency show stronger condensations on polyelectrolyte chains. Such counterions can induce polyelectrolyte chains from extended conformation to compact state, and then to swollen conformation with the increase of counterion concentrations.

With the ENUF-DPD method, we further investigate the specific binding structures of dendrimers on amphiphilic bilayer membranes [74]. We construct mutually consistent coarse-grained models for dendrimers and lipid molecules, which can properly describe the conformation of charged dendrimers and the surface tension of amphiphilic membranes, respectively. Systematic simulations are performed and simulation results reveal that the permeability of dendrimers across membranes is enhanced upon increasing dendrimer sizes. The negative curvature of amphiphilic membrane formed in dendrimer-membrane complexes is related to dendrimer concentration. Higher dendrimer concentration together with the synergistic effect between charged dendrimers can also enhance the permeability of dendrimers across amphiphilic membranes.

With these two typical and representative applications, we can see that the newly implemented ENUF-DPD method can capture the essential characteristics of electrostatic interactions at mesoscopic level. This method has all capabilities of ordinary DPD method, but includes applications where electrostatic interactions are essential but previously inaccessible, hence can be used to study charged complex systems at mesoscopic level.

## 6. Summary and Conclusions

Treatment of electrostatic interactions based on Ewald summation techniques is reviewed. While Ewald-summation is still considered as the most accurate scheme to compute the long-ranged interactions, it is also the part slowing down simulations. The scaling for large systems makes the computations very time-consuming. In particular, the reciprocal part of Ewald becomes a bottle-neck. As an attractive alternative approach to mesh-based schemes which show a linear scaling, we introduce an Ewald method based on non-uniform fast Fourier transforms (ENUF) giving examples of two implementations in already existing software packages, for atomistic Molecular Dynamics and Dissipative Particle Dynamics. We demonstrate that the implementation becomes straight-forward as we rely on NFFT library. We discuss the optimization of convergence parameters and window functions.

The ENUF method scales linearly as  $\mathcal{O}(N \log N)$  and conserves both the energy and momentum to float point accuracy making it a very robust and accurate method.

## Acknowledgements

Y.-L. W. and A. L. acknowledge the KA Wallenberg Foundation for financial support. F. M. acknowledges the Wenner-Gren and the Carl Tryggers Foundations for funding for Visiting Professorship. Y.-L. W. and A. L. thank the Kavli Institute for Theoretical Physics China (KITPC) and Chinese Academy of Sciences (CAS) for support and scientific environment during their stay. This work is supported by SERC the Swedish e-Science Center and Swedish Science Council. Parts of the computations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at PDC, HPC2N, and NSC.

## REFERENCES

- [1] M. P. Allen and D. J. Tildesley, "Computer Simulation of Liquids," Oxford University Press, New York, 1989.
- [2] J. M. Haile, "Molecular Dynamics Simulation: Elementary Methods," John Wiley & Sons, New York, 1992.
- [3] D. Frenkel and B. Smit, "Understanding Molecular Simulation: From Algorithms to Applications," Academic Press, Inc., Orlando, 1996.
- [4] D. M. York, T. A. Darden and L. G. Pedersen, "The Effect of Long-Range Electrostatic Interactions in Simulations of Macro-

- molecular Crystals: A Comparison of the Ewald and Truncated List Methods,” *Journal of Chemical Physics*, Vol. 99, No. 10, 1993, p. 8345. <http://dx.doi.org/10.1063/1.465608>
- [5] M. Bergdorf, C. Peter and P. H. Hünenberger, “Influence of Cut-off Truncation and Artificial Periodicity of Electrostatic Interactions in Molecular Simulations of Solvated Ions: A Continuum Electrostatics Study,” *Journal of Chemical Physics*, Vol. 119, No. 17, 2003, p. 9129. <http://dx.doi.org/10.1063/1.1614202>
- [6] C. Anézo, A. H. de Vries, H. D. Höltje, D. P. Tieleman and S. J. Marrink, “Methodological Issues in Lipid Bilayer Simulations,” *Journal of Physical Chemistry B*, Vol. 107, No. 35, 2003, pp. 9424-9433. <http://dx.doi.org/10.1021/jp0348981>
- [7] M. Patra, M. Karttunen, M. T. Hyvönen, E. Falck, P. Lindqvist and I. Vattulainen, “Molecular Dynamics Simulations of Lipid Bilayers: Major Artifacts Due to Truncating Electrostatic Interactions,” *Biophysical Journal*, Vol. 84, No. 6, 2003, pp. 3636-3645. [http://dx.doi.org/10.1016/S0006-3495\(03\)75094-2](http://dx.doi.org/10.1016/S0006-3495(03)75094-2)
- [8] P. P. Ewald, “Die Berechnung Optischer und Elektrostatischer Gitterpotentiale,” *Annalen der Physik*, Vol. 369, No. 3, 1921, pp. 253-287. <http://dx.doi.org/10.1002/andp.19213690304>
- [9] S. W. de Leeuw, J. W. Perram and E. R. Smith, “Simulation of Electrostatic Systems in Periodic Boundary Conditions, I. Lattice Sums and Dielectric Constants,” *Proceedings of the Royal Society London, Series A*, Vol. 373, No. 1752, 1980, pp. 27-56. <http://dx.doi.org/10.1098/rspa.1980.0135>
- [10] D. M. Heyes, “Electrostatic Potentials and Fields in Infinite Point Charge Lattices,” *Journal of Chemical Physics*, Vol. 74, No. 3, 1981, pp. 1924-1929.
- [11] T. Darden, D. York and L. Pedersen, “Particle Mesh Ewald: An  $MlogN$  Method for Ewald Sums in Large Systems,” *Journal of Chemical Physics*, Vol. 98, No. 12, 1993, pp. 10089-10092. <http://dx.doi.org/10.1063/1.464397>
- [12] D. York and W. Yang, “The Fast Fourier Poisson Method for Calculating Ewald Sums,” *Journal of Chemical Physics*, Vol. 101, No. 4, 1994, pp. 3298-3300. <http://dx.doi.org/10.1063/1.467576>
- [13] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee and L. G. Pedersen, “A Smooth Particle Mesh Ewald Method,” *Journal of Chemical Physics*, Vol. 103, No. 19, 1995, pp. 8577-8593. <http://dx.doi.org/10.1063/1.470117>
- [14] P. F. Batcho and T. Schlick, “New Splitting Formulations for Lattice Summations,” *Journal of Chemical Physics*, Vol. 115, No. 18, 2001, p. 8312. <http://dx.doi.org/10.1063/1.1412247>
- [15] D. R. Wheeler and J. Newman, “A Less Expensive Ewald Lattice Sum,” *Chemical Physics Letters*, Vol. 366, No. 5-6, 2002, pp. 537-543. [http://dx.doi.org/10.1016/S0009-2614\(02\)01644-5](http://dx.doi.org/10.1016/S0009-2614(02)01644-5)
- [16] Y. Shan, J. L. Klepeis, M. P. Eastwood, R. O. Dror and D. E. Shaw, “Gaussian Split Ewald: A Fast Ewald Mesh Method for Molecular Simulation,” *Journal of Chemical Physics*, Vol. 122, No. 5, 2005, Article ID: 054101. <http://dx.doi.org/10.1063/1.1839571>
- [17] P. J. Steinbach and B. R. Brooks, “New Spherical-Cutoff Methods for Long-Range Forces in Macromolecular Simulation,” *Journal of Computational Chemistry*, Vol. 15, No. 7, 1994, pp. 667-683. <http://dx.doi.org/10.1002/jcc.540150702>
- [18] C. L. Brooks III, B. M. Pettitt and M. Karplus, “Structural and Energetic Effects of Truncating Long Ranged Interactions in Ionic and Polar Fluids,” *Journal of Chemical Physics*, Vol. 83, No. 11, 1985, p. 5897. <http://dx.doi.org/10.1063/1.449621>
- [19] K. S. Kim, “On Effective Methods to Treat Solvent Effects in Macromolecular Mechanics and Simulations,” *Chemical Physics Letters*, Vol. 156, No. 2-3, 1989, pp. 261-268. [http://dx.doi.org/10.1016/S0009-2614\(89\)87131-3](http://dx.doi.org/10.1016/S0009-2614(89)87131-3)
- [20] L. Greengard and V. Rokhlin, “A Fast Algorithm for Particle Simulations,” *Journal of Computational Physics*, Vol. 73, No. 2, 1987, pp. 325-348. [http://dx.doi.org/10.1016/0021-9991\(87\)90140-9](http://dx.doi.org/10.1016/0021-9991(87)90140-9)
- [21] J. E. Barnes and P. Hut, “A Hierarchical ( $MlogN$ ) Force-Calculation Algorithm,” *Nature*, Vol. 324, 1986, pp. 446-449. <http://dx.doi.org/10.1038/324446a0>
- [22] J. Pérez-Jordá and W. Yang, “A Simple  $O(MlogN)$  Algorithm for the Rapid Evaluation of Particle-Particle Interactions,” *Chemical Physics Letters*, Vol. 247, No. 4-6, 1995, pp. 484-490.
- [23] Z. H. Duan and R. Krasny, “An Adaptive Treecode for Computing Nonbonded Potential Energy in Classical Molecular Systems,” *Journal of Computational Chemistry*, Vol. 22, No. 2, 2001, pp. 184-195. [http://dx.doi.org/10.1002/1096-987X\(20010130\)22:2<184::AID-JCC6>3.0.CO;2-7](http://dx.doi.org/10.1002/1096-987X(20010130)22:2<184::AID-JCC6>3.0.CO;2-7)
- [24] I. Tsukerman, “Efficient Computation of Long-Range Electromagnetic Interactions without Fourier Transforms,” *IEEE Transactions on Magnetics*, Vol. 40, No. 4, 2004, pp. 2158-2160. <http://dx.doi.org/10.1109/TMAG.2004.829022>
- [25] E. T. Ong, K. M. Lim, K. H. Lee and H. P. Lee, “A Fast Algorithm for Three-Dimensional Potential Fields Calculation: Fast Fourier Transform on Multipoles,” *Journal of Computational Physics*, Vol. 192, No. 1, 2003, pp. 244-261. <http://dx.doi.org/10.1016/j.jcp.2003.07.004>
- [26] C. Sagui and T. Darden, “Multigrid Methods for Classical Molecular Dynamics Simulations of Biomolecules,” *Journal of Chemical Physics*, Vol. 114, No. 15, 2001, p. 6578. <http://dx.doi.org/10.1063/1.1352646>
- [27] R. D. Skeel, I. Tezcan and D. J. Hardy, “Multiple Grid Methods for Classical Molecular Dynamics,” *Journal of Computational Chemistry*, Vol. 23, No. 6, 2002, pp. 673-684. <http://dx.doi.org/10.1002/jcc.10072>
- [28] J. A. Izaguirre, S. S. Hampton and T. Matthey, “Parallel Multigrid Summation for the N-Body Problem,” *Journal of Parallel*

- and Distributed Computing, Vol. 65, No. 8, 2005, pp. 949-962. <http://dx.doi.org/10.1016/j.jpdc.2005.03.006>
- [29] J. A. Baker and R. O. Watts, "Monte Carlo Studies of the Dielectric Properties of Water-Like Models," *Molecular Physics*, Vol. 26, No. 3, 1973, pp. 789-792. <http://dx.doi.org/10.1080/00268977300102101>
- [30] I. G. Tironi, R. Sperb, P. E. Smith and W. F. van Gunsteren, "A Generalized Reaction Field Method for Molecular Dynamics Simulations," *Journal of Chemical Physics*, Vol. 102, No. 13, 1995, p. 5451. <http://dx.doi.org/10.1063/1.469273>
- [31] T. N. Heinz and P. H. Hünenberger, "Combining the Lattice-Sum and Reaction-Field Approaches for Evaluating Long-Range Electrostatic Interactions in Molecular Simulations," *Journal of Chemical Physics*, Vol. 123, No. 3, 2005, Article ID: 034107. <http://dx.doi.org/10.1063/1.1955525>
- [32] R. W. Hockney and J. W. Eastwood, "Computer Simulation Using Particles," McGraw-Hill, New York, 1981.
- [33] B. A. Luty, M. E. Davis, I. G. Tironi and W. F. van Gunsteren, "A Comparison of Particle-Particle, Particle-Mesh and Ewald Methods for Calculating Electrostatic Interactions in Periodic Molecular Systems," *Molecular Simulation*, Vol. 14, No. 1, 1994, pp. 11-20. <http://dx.doi.org/10.1080/08927029408022004>
- [34] X. Wu and B. R. Brooks, "Isotropic Periodic Sum: A Method for the Calculation of Long-Range Interactions," *Journal of Chemical Physics*, Vol. 122, No. 4, 2005, Article ID: 044107. <http://dx.doi.org/10.1063/1.1836733>
- [35] T. M. Apostol, "Mathematical Analysis," Addison-Wesley, Boston, 1979.
- [36] J. W. Perram, H. G. Petersen and S. W. de Leeuw, "An Algorithm for the Simulation of Condensed Matter Which Grows as the  $3/2$  Power of the Number of Particles," *Molecular Physics*, Vol. 65, No. 4, 1988, pp. 875-893. <http://dx.doi.org/10.1080/00268978800101471>
- [37] J. Kolafa and J. W. Perram, "Cutoff Errors in the Ewald Summation Formulae for Point Charge Systems," *Molecular Simulation*, Vol. 9, No. 5, 1992, pp. 351-368. <http://dx.doi.org/10.1080/08927029208049126>
- [38] S. W. de Leeuw, J. W. Perram and E. R. Smith, "Simulation of Electrostatic Systems in Periodic Boundary Conditions. II. Equivalence of Boundary Conditions," *Proceedings of the Royal Society London, Series A*, Vol. 373, No. 1752, 1980, pp. 57-66. <http://dx.doi.org/10.1098/rspa.1980.0136>
- [39] S. W. de Leeuw, J. W. Perram and E. R. Smith, "Simulation of Electrostatic Systems in Periodic Boundary Conditions. III. Further Theory and Applications," *Proceedings of the Royal Society London, Series A*, Vol. 388, No. 1794, 1983, pp. 177-193. <http://dx.doi.org/10.1098/rspa.1983.0077>
- [40] W. H. Press and G. B. Rybicki, "Fast Algorithm for Spectral Analysis of Unevenly Sampled Data," *Astrophysical Journal*, Vol. 338, 1989, pp. 277-280. <http://dx.doi.org/10.1086/167197>
- [41] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, Vol. 19, 1965, pp. 297-301. <http://dx.doi.org/10.1090/S0025-5718-1965-0178586-1>
- [42] A. Brandt, "Multilevel Computations of Integral Transforms and Particle Interactions with Oscillatory Kernels," *Computer Physics Communications*, Vol. 65, No. 1-3, 1991, pp. 24-38. [http://dx.doi.org/10.1016/0010-4655\(91\)90151-A](http://dx.doi.org/10.1016/0010-4655(91)90151-A)
- [43] M. Pippig and D. Potts, "Parallel Three-Dimensional Nonequispaced Fast Fourier Transforms and Their Application to Particle Simulation," *SIAM Journal on Scientific Computing*, Vol. 35, No. 4, 2013, pp. 411-437. <http://dx.doi.org/10.1137/120888478>
- [44] O. Ayala and L. P. Wang, "Parallel Implementation and Scalability Analysis of 3D Fast Fourier Transform Using 2D Domain Decomposition," *Parallel Computing*, Vol. 39, No. 1, 2013, pp. 58-77. <http://dx.doi.org/10.1016/j.parco.2012.12.002>
- [45] Q. Liu and N. Nguyen, "An Accurate Algorithm for Nonuniform Fast Fourier Transforms (NUFFT's)," *IEEE Microwave and Guided Wave Letters*, Vol. 8, No. 1, 1998, pp. 18-20. <http://dx.doi.org/10.1109/75.650975>
- [46] G. Steidl, "A Note on Fast Fourier Transforms for Nonequispaced Grids," *Advances in Computational Mathematics*, Vol. 9, No. 3-4, 1998, pp. 337-352. <http://dx.doi.org/10.1023/A:1018901926283>
- [47] J. A. Fessler and B. P. Sutton, "Nonuniform Fast Fourier Transforms Using Min-Max Interpolation," *IEEE Transactions on Signal Processing*, Vol. 51, No. 2, 2003, pp. 560-574. <http://dx.doi.org/10.1109/TSP.2002.807005>
- [48] C. Anderson and M. D. Dahleh, "Rapid Computation of the Discrete Fourier Transform," *SIAM Journal on Scientific Computing*, Vol. 17, No. 4, 1996, pp. 913-919. <http://dx.doi.org/10.1137/0917059>
- [49] A. Duijndam and M. Schonewille, "Nonuniform Fast Fourier Transform," *Geophysics*, Vol. 64, No. 2, 1999, p. 539. <http://dx.doi.org/10.1190/1.1444560>
- [50] J. P. Boyd, "A Fast Algorithm for Chebyshev, Fourier, and Sinc Interpolation onto an Irregular Grid," *Journal of Computational Physics*, Vol. 103, No. 2, 1992, pp. 243-257. [http://dx.doi.org/10.1016/0021-9991\(92\)90399-J](http://dx.doi.org/10.1016/0021-9991(92)90399-J)
- [51] A. F. Ware, "Fast Approximate Fourier Transforms for Irregularly Spaced Data," *SIAM Review*, Vol. 40, No. 4, 1998, pp. 838-856. <http://dx.doi.org/10.1137/S003614459731533X>
- [52] A. Nieslony and G. Steidl, "Approximate Factorizations of Fourier Matrices with Nonequispaced Knots," *Linear Algebra and its Applications*, Vol. 366, 2003, pp. 337-351. [http://dx.doi.org/10.1016/S0024-3795\(02\)00496-2](http://dx.doi.org/10.1016/S0024-3795(02)00496-2)
- [53] J. J. Benedetto and P. J. Ferreira, "Modern Sampling Theory: Mathematics and Applications," Springer, Berlin, 2001. <http://dx.doi.org/10.1007/978-1-4612-0143-4>

- [54] S. Kunis and D. Potts, "NFFT3.2.3," Institute of Mathematics, University of Lübeck, D-23560 Lübeck, 2013. <http://www.math.uni-luebeck.de/potts/nfft/>
- [55] F. Hedman and A. Laaksonen, "A Data-Parallel Molecular Dynamics Method for Liquids with Coulombic Interactions," *Molecular Simulation*, Vol. 14, No. 4-5, 1995, pp. 235-244. <http://dx.doi.org/10.1080/08927029508022020>
- [56] F. Hedman, "Algorithms for Molecular Dynamics Simulations," Ph.D. Thesis, Stockholm University, Stockholm, 2006. <http://www.diva-portal.org/smash/record.jsf?searchId=1&pid=diva2:178477>
- [57] F. Hedman and A. Laaksonen, "Ewald Summation Based on Nonuniform Fast Fourier Transform," *Chemical Physics Letters*, Vol. 425, No. 1-3, 2006, pp. 142-147. <http://dx.doi.org/10.1016/j.cplett.2006.04.106>
- [58] M. Frigo and S. G. Johnson, "FFTW-The Design and Implementation of FFTW3," *Proceedings of the IEEE*, Vol. 93, No. 2, 2005, pp. 216-231. <http://dx.doi.org/10.1109/JPROC.2004.840301>
- [59] A. P. Lyubartsev and A. Laaksonen, "M.DynaMix—A Scalable Portable Parallel MD Simulation Package for Arbitrary Molecular Mixtures," *Computer Physics Communications*, Vol. 128, No. 3, 2000, pp. 565-589. [http://dx.doi.org/10.1016/S0010-4655\(99\)00529-9](http://dx.doi.org/10.1016/S0010-4655(99)00529-9)
- [60] J. P. Ryckaert, G. Ciccotti and H. J. C. Berendsen, "Numerical Integration of the Cartesian Equations of Motion of a System with Constraints: Molecular Dynamics of n-Alkanes," *Journal of Computational Physics*, Vol. 23, No. 3, 1977, pp. 327-341. [http://dx.doi.org/10.1016/0021-9991\(77\)90098-5](http://dx.doi.org/10.1016/0021-9991(77)90098-5)
- [61] M. Tuckerman, B. J. Berne and G. J. Martyna, "Reversible Multiple Time Scale Molecular Dynamics," *Journal of Chemical Physics*, Vol. 97, 1992, pp. 1990-2001.
- [62] <http://www.yolinux.com/TUTORIALS/LinuxTutorialMixingFortranAndC.html>
- [63] A. P. Lyubartsev, M. DynaMix User Manual. [http://www.fos.su.se/sasha/md\\_prog.html](http://www.fos.su.se/sasha/md_prog.html)
- [64] P. J. Hoogerbrugge and V. A. Koelman, "Simulating Microscopic Hydrodynamic Phenomena with Dissipative Particle Dynamics," *Europhysics Letters*, Vol. 19, No. 3, 1992, p. 155. <http://dx.doi.org/10.1209/0295-5075/19/3/001>
- [65] V. A. Koelman and P. J. Hoogerbrugge, "Dynamic Simulations of Hard-Sphere Suspensions under Steady Shear," *Europhysics Letters*, Vol. 21, No. 3, 1993, p. 363. <http://dx.doi.org/10.1209/0295-5075/21/3/018>
- [66] Z. Y. Lu and Y. L. Wang, "An Introduction to Dissipative Particle Dynamics," In: L. Monticelli and E. Salonen, Eds., *Methods in Molecular Biology: Biomolecular Simulations*, Humana Press, New York, 2013, pp. 617-633. [http://link.springer.com/protocol/10.1007/978-1-62703-017-5\\_24](http://link.springer.com/protocol/10.1007/978-1-62703-017-5_24)
- [67] P. Español and P. Warren, "Statistical Mechanics of Dissipative Particle Dynamics," *Europhysics Letters*, Vol. 30, No. 4, 1995, p. 191. <http://dx.doi.org/10.1209/0295-5075/30/4/001>
- [68] R. D. Groot and P. B. Warren, "Dissipative Particle Dynamics: Bridging the Gap between Atomistic and Mesoscopic Simulation," *Journal of Chemical Physics*, Vol. 107, No. 11, 1997, p. 4423. <http://dx.doi.org/10.1063/1.474784>
- [69] R. D. Groot, "Electrostatic Interactions in Dissipative Particle Dynamics-Simulation of Polyelectrolytes and Anionic Surfactants," *Journal of Chemical Physics*, Vol. 118, No. 24, 2003, p. 11265. <http://dx.doi.org/10.1063/1.1574800>
- [70] M. González-Melchor, E. Mayoral, M. E. Velázquez and J. Alejandre, "Electrostatic Interactions in Dissipative Particle Dynamics Using the Ewald Sums," *Journal of Chemical Physics*, Vol. 125, No. 22, 2006, p. 224107. <http://dx.doi.org/10.1063/1.2400223>
- [71] Y. L. Wang, A. Laaksonen and Z. Y. Lu, "Implementation of Non-Uniform FFT Based Ewald Summation in Dissipative Particle Dynamics Method," *Journal of Computational Physics*, Vol. 235, 2013, pp. 666-682. <http://dx.doi.org/10.1016/j.jcp.2012.09.023>
- [72] Y. L. Wang, "Electrostatic Interactions in Coarse-Grained Simulations: Implementations and Applications," Ph.D. Thesis, Stockholm University, Stockholm, 2013. <http://www.diva-portal.org/smash/record.jsf?pid=diva2:641063>
- [73] A. AlSunaidi, W. K. den Otter and J. H. R. Clarke, "Liquid-Crystalline Ordering in Rod-Coil Diblock Copolymers Studied by Mesoscale Simulations," *Philosophical Transactions of the Royal Society London A*, Vol. 362, No. 1821, 2004, pp. 1773-1781. <http://dx.doi.org/10.1098/rsta.2004.1414>
- [74] Y. L. Wang, Z. Y. Lu and A. Laaksonen, "Specific Binding Structures of Dendrimers on Lipid Bilayer Membranes," *Physical Chemistry Chemical Physics*, Vol. 14, No. 23, 2012, pp. 8348-8359. <http://dx.doi.org/10.1039/c2cp40700k>

# Bibliografia

- [1] B.J. Alder, T.E. Wainwright, *Studies in Molecular Dynamics. I. General Method*, J. Chem. Phys, 31 (2): 459, 1959.
- [2] A. Rahman, *Correlations in the Motion of Atoms in Liquid Argon*, Physical Review, 136 (2A): A405-A411, 1964.
- [3] T. Schlick, *Pursuing Laplace's Vision on Modern Computers*, Mathematical Applications to Biomolecular Structure and Dynamics, IMA Volumes in Mathematics and Its Applications 82., Springer-Verlag. pp. 218-247, 1996.
- [4] D.W. Brenner, *The art and science of an analytical potential*, Phys. Stat. Sol. B, 217:23, 2000.
- [5] L. Verlet, *Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules*, Phys. Rev., 159: 98-103 , 1967.
- [6] J.E. Lennard-Jones, *On the Determination of Molecular Fields*, Proc. R. Soc. Lond. A,106 (738): 463-477 , 1924.
- [7] C.L. Kong, *Combining rules for intermolecular potential parameters. II. Rules for the Lennard-Jones (12-6) potential and the Morse potential*, J. Chem. Phys., 59, 2464-2467, 1973.
- [8] K. Huang, *Statistical Mechanics*, Wiley, 1987.
- [9] W.L. Jorgensen, J. Chandrasekhar, J.D. Madura, R.W. Impey, M.L. Klein, *Comparison of simple potential functions for simulating liquid water*, J. Chem. Phys., 79, 1983.
- [10] K. Toukan, A. Rahman, *Molecular-dynamics study of atomic motions in water*, Phys. Rev. B, 31: 2643-2648, 1985.

- [11] A. PÅ©rez, I. MarchÅ;n, D. Svozil, J. Sponer, T.E. Cheatham, C.A. Laughton, M. Orozco, *Refinement of the AMBER Force Field for Nucleic Acids: Improving the Description of  $\alpha/\gamma$  Conformers*, Biophysical Journal, 92, 11, 3817-3829, 2007.
- [12] H.J.C. Berendsen, J.R. Grigera, T.P. Straatsma, *The missing term in effective pair potentials*, J. Phys. Chem., 91: 6269-6271, 1987.
- [13] J. Aqvist, *Ion-water interaction potentials derived from free energy perturbation simulations*, J. Phys. Chem, 94, 8021, 1990.
- [14] M.W. Mahoney, W.L. Jorgensen, *A five-site model for liquid water and the reproduction of the density anomaly by rigid, nonpolarizable potential functions*, J. Chem. Phys., 112 8910-8922, 2000.
- [15] J. Kleinjung, F. Fraternali, *Design and application of implicit solvent models in biomolecular simulations*, Curr Opin Struct Biol., 25(100): 126-134, 2014.
- [16] J.P. Ryckaert, G. Ciccotti, H.J.C. Berendsen, *Numerical Integration of the Cartesian Equations of Motion of a System with Constraints: Molecular Dynamics of n-Alkanes*, J. Comp. Phy., 23 (3): 327-341, 1977.
- [17] T.R. Forester, W. Smith W, *SHAKE, Rattle, and Roll: Efficient Constraint Algorithms for Linked Rigid Bodies*, J. Comp. Chem., 19: 102-111, 1998.
- [18] M. P. Allen, D. J. Tildesley, *Computer simulation of liquids*, Oxford University Press, New York, 1989.
- [19] J. M. Haile, *Molecular Dynamics Simulation: Elementary methods*, John Wiley & Sons, New York, 1992.
- [20] T. Kuznetsova, B. Kvamme, *Thermodynamic properties and interfacial tension of a model water-carbon dioxide system*, Phys. Chem. Chem. Phys., 4 (6): 937-941, 2002.
- [21] Y. Cheng, N. Korolev, L. Nordenskiöld, *Similarities and differences in interaction of  $K^+$  and  $Na^+$  with condensed ordered DNA. A molecular dynamics computer simulation study*, Nucleic Acids Research, 34 (2): 686-696, 2006.
- [22] C.-J. Högberg, A.M. Nikitin and A.P. Lyubartsev, *Modification of the CHARMM force field for DMPC lipid bilayer*, J. Comp. Chem., 29 (14): 2359-2369, 2008.

- [23] G. Raabe and J. Köhler, *Thermodynamical and structural properties of imidazolium based ionic liquids from molecular simulation*, J. Chem. Phys, 128 (15): 154509, 2008.
- [24] E.S. Böes, E. Bernardi, H. Stassen, P.F.B. Gonçalves, *Solvation of monovalent anions in formamide and methanol: Parameterization of the IEF-PCM model*, Chem. Phys., 344: 101-113, 2008.
- [25] D.A. Case, D.A. Pearlman, J.W. Caldwell, T.E. Cheatham, W.S. Ross, C.L. Simmering, T.A. Darden, K.M. Merz, R.V. Stanton, A.L. Cheng, J.J. Vincent, M. Crowley, D.M. Ferguson, V. Tsui, R.J. Radmer, Y. Duan, J. Pitner, I. Massova, G.L. Seibel, U.C. Singh, P.K. Weiner, P.A. Kollman *AMBER 6.0*, University of California, San Francisco, CA., 1999.
- [26] B. Hess, C. Kutzner, D. van der Spoel, E. Lindahl, . *GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation*, J. Chem. Theory Comput, 4, 435-447, 2008.
- [27] D. Frenkel, B. Smit, *Understanding molecular simulation: from algorithms to applications*, Academic Press, Inc. Orlando, FL, USA, 1996.
- [28] D. M. York, T. A. Darden, L. G. Pedersen *The effect of long-range electrostatic interactions in simulations of macromolecular crystals: A comparison of the Ewald and truncated list methods*, J. Chem. Phys., 99, 8345, 1993.
- [29] M. Bergdorf, C. Peter, P. H. Hünenberger, *Influence of cut-off truncation and artificial periodicity of electrostatic interactions in molecular simulations of solvated ions: A continuum electrostatics study*, J. Chem. Phys., 119, 9129, 2003.
- [30] P. P. Ewald, *Die Berechnung optischer und elektrostatischer Gitterpotentiale*, Ann. Phys., 369, 253, 1921.
- [31] S. W. de Leeuw, J. W. Perram, E. R. Smith, *Simulation of electrostatic systems in periodic boundary conditions. I. Lattice sums and dielectric constants*, Proc. R. Soc. London, Ser. A, 373, 27, 1980.
- [32] D. M. Heyes, *Electrostatic potentials and fields in infinite point charge lattices*, J. Chem. Phys., 74, 1924, 1981.

- [33] T. Darden, D. York, L. Pedersen, *Particle mesh Ewald: An  $N \log N$  method for Ewald sums in large systems*, J. Chem. Phys., 98, 10089, 1993.
- [34] D. York, W. Yang, *The fast Fourier Poisson method for calculating Ewald sums*, J. Chem. Phys., 101, 3298, 1994.
- [35] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, L. G. Pedersen, *A smooth particle mesh Ewald method*, J. Chem. Phys., 103, 8577, 1995.
- [36] P. F. Batcho, T. Schlick, *New splitting formulations for lattice summations*, J. Chem. Phys., 115, 8312, 2001.
- [37] D. R. Wheeler, J. Newman, *A less expensive Ewald lattice sum*, Chem. Phys. Lett., 366, 537, 2002.
- [38] Y. Shan, J. L. Klepeis, M. P. Eastwood, R. O. Dror, D. E. Shaw, *Gaussian split Ewald: A fast Ewald mesh method for molecular simulation*, J. Chem. Phys., 122, 054101, 2005.
- [39] P. J. Steinbach, B. R. Brooks, *New spherical-cutoff methods for long-range forces in macromolecular simulation*, J. Comput. Chem., 15, 667, 1994.
- [40] C. L. Brooks III, B. M. Pettitt, M. Karplus, *Structural and energetic effects of truncating long ranged interactions in ionic and polar fluids*, J. Chem. Phys., 83, 5897, 1985.
- [41] K. S. Kim, *On effective methods to treat solvent effects in macromolecular mechanics and simulations*, Chem. Phys. Lett., 156, 261, 1989.
- [42] T. M. Apostol, *Mathematical Analysis*, Addison-Wesley, 1979.
- [43] J. W. Perram, H. G. Petersen, S. W. de Leeuw, *An algorithm for the simulation of condensed matter which grows as the  $3/2$  power of the number of particles*, Mol. Phys., 65, 875, 1988.
- [44] J. Kolafa, J. W. Perram, *Cutoff errors in the Ewald summation formulae for point charge systems*, Mol. Simul., 9, 351, 1992.



- [45] S. W. de Leeuw, J. W. Perram, E. R. Smith, *Simulation of electrostatic systems in periodic boundary conditions. II. Equivalence of boundary conditions*, Proc. R. Soc. London, Ser. A, 373, 57, 1980.
- [46] S. W. de Leeuw, J. W. Perram, E. R. Smith, *Simulation of electrostatic systems in periodic boundary conditions. III. Further theory and applications*, Proc. R. Soc. London, Ser. A, 388, 177, 1983.
- [47] W. H. Press, G. B. Rybicki, *Fast algorithm for spectral analysis of unevenly sampled data*. Astrophys. J., 338, 277, 1989.
- [48] J. W. Cooley, J. W. Tukey, *An algorithm for the machine calculation of complex Fourier series*, Math. Comp., 19, 297, 1965.
- [49] A. Brandt, *Multilevel computations of integral transforms and particle interactions with oscillatory kernels*, Comput. Phys. Commun., 65, 24, 1991.
- [50] M. Pippig, D. Potts, *Parallel three-dimensional nonequispaced fast Fourier transforms and their application to particle simulation*, SIAM J. Sci. Comput., 35, 411, 2013.
- [51] O. Ayala, L. P. Wang, *Parallel implementation and scalability analysis of 3D fast Fourier Transform using 2D domain decomposition*, Parallel Computing, 39, 58, 2013.
- [52] Q. Liu, N. Nguyen, *An accurate algorithm for nonuniform fast Fourier transforms (NUFFT's)*, IEEE Microwave and Guided Wave Lett., 8, 18, 1998.
- [53] G. Steidl, *A note on fast Fourier transforms for nonequispaced grids*, Adv. Comput. Math., 9, 337, 1998.
- [54] J. A. Fessler, B. P. Sutton, *Nonuniform fast Fourier transforms using min-max interpolation*, IEEE Trans. Sign. Proc., 51, 560, 2003.
- [55] C. Anderson, M. D. Dahleh, *Rapid computation of the discrete Fourier transform*, SIAM J. Sci. Comput., 17, 913, 1996.
- [56] A. Duijndam, M. Schonewille, *Nonuniform fast Fourier transform*, Geophys., 64, 539, 1999.

- [57] J. P. Boyd, *A fast algorithm for Chebyshev, Fourier, and sinc interpolation onto an irregular grid*, J. Comput. Phys., 103, 243, 1992.
- [58] A. F. Ware, *Fast approximate Fourier transforms for irregularly spaced data*, SIAM Rev., 40, 838, 1998.
- [59] A. Nieslony, G. Steidl, *Approximate factorizations of Fourier matrices with nonequispaced knots*, Lin. Alg. Appl., 366, 337, 2003.
- [60] J. J. Benedetto, P. J. Ferreira, *Modern sampling theory: mathematics and applications*, Springer, 2001.
- [61] S. Kunis, D. Potts, *NFFT3.2.3*, University of Lübeck, Institute of Mathematics, D-23560 Lübeck, Germany, 2013, <http://www.math.uni-luebeck.de/potts/nfft/>,
- [62] F. Hedman, A. Laaksonen, *A Data-parallel Molecular Dynamics Method for Liquids with Coulombic Interactions*, Mol. Simul., 14, 235, 1995.
- [63] F. Hedman, A. Laaksonen, *Ewald summation based on nonuniform fast Fourier transform*, Chem. Phys. Lett., 425, 142, 2006.
- [64] M. Frigo, S. G. Johnson, *FFTW—The design and implementation of FFTW3*, Proceedings of the IEEE, 93, 216, 2005.
- [65] A. P. Lyubartsev, A. Laaksonen, *M.DynaMix - A scalable portable parallel MD simulation package for arbitrary molecular mixtures*, Comput. Phys. Commun., 128, 565, 2000.
- [66] M. Tuckerman, B. J. Berne, G. J. Martyna, *Reversible multiple time scale molecular dynamics*, J. Chem. Phys., 97, 1990, 1992.
- [67] Y. Wang, F. Hedman, M. Porcu, F. Mocci, A. Laaksonen *Non-Uniform FFT and its applications in particle simulations*, App. Math., 5, 3, 2014.
- [68] See for instance: <http://www.yolinux.com/TUTORIALS/LinuxTutorialMixingFortranAndC.html>,
- [69] A. P. Lyubartsev, *M.DynaMix User Manual*, [http://www.fos.su.se/~sasha/md\\_prog.html](http://www.fos.su.se/~sasha/md_prog.html),

- [70] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, J. Dongarra, *MPI-The Complete Reference, Volume 1: The MPI Core*, MIT Press, Cambridge, MA, USA, 1998.
- [71] J.F. Jerger, D. Hayes, *The cross-check principle*, *Pediatric Audiometry*, 102:614-620, 1976.
- [72] R.G. Turner, *Double checking the cross-check principle*, *J. Am. Acad. Audiol.*, 14(5):269-77, 2003.
- [73] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, J. Dongarra, *MPI-The Complete Reference, Volume 1: The MPI Core*, MIT Press, Cambridge, MA, USA, 1998.
- [74] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, J. Dongarra, *MPI-The Complete Reference, Volume 1: The MPI Core*, MIT Press, Cambridge, MA, USA, 1998.
- [75] D.E. Smith, L.X. Dang, *Computer simulations of NaCl association in polarizable water*, *J. Chem. Phys.*, 100, 3757-3765, 1994.
- [76] H. Basu, B. Feuerstein, D. Zarling, R. Shafer, L. Marton, *Recognition of Z-RNA and Z-DNA determinants by polyamines in solution: experimental and theoretical studies*, *J. Biomol. Struct. Dyn.*, 6, (2): 299-309, 1988.
- [77] A.G. Leslie, S. Arnott, R. Chandrasekaran, R.L. Ratliff, *Polymorphism of DNA double helices*, *J. Mol. Biol.*, 143 (1): 49-72, 1980.
- [78] J. Watson, F. Crick, *The Molecular structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid*, *Nature*, 171, 737-738, 1953.
- [79] R. Sinden, *DNA structure and Function*, Academic Press, 1994.
- [80] B. Jayaram, D.L. Beveridge, *Modeling DNA in aqueous solutions: theoretical and computer simulation studies on the ion atmosphere of DNA*, *Annu. Rev. Biophys. Biomol. Struct.*, 25, 367-394, 1996.
- [81] E. Ennifar, P. Walter, P. Dumas, B.M. Pettitt, *A crystallographic study of the binding of 13 metal ions to two related RNA duplexes*, *Nucleic Acid Res.*, 31, 2671-2682, 2003.
- [82] N.V. Hud, M. Polak, *DNA-cation interactions: the major and the minor groove are flexible ionophores*, *Curr. Opin. Struct. Biol.*, 11, 293-301, 2001.

- [83] D. Beveridge, K. McConnel, *Nucleic Acids: Theory and computer simulation*, Y2K. Curr. Opin. Struct Biol, 10, 182196, 2000.
- [84] W.H. Braulin, *Nmr study of cation-binding environment on nucleic acids*, Advances in Biophysical Chemistry, 5, 187-217, 1995.
- [85] M.A. Young, B. Jayaram, D. Beveridge, *Intrusion of counterions into the spine of hydration in the minor groove of B-DNA: Fractional occupancy of electronegative pockets*, J. Am. Chem. Soc., 119, 59-69, 1997.
- [86] F. Mocci, A. Laaksonen, A. Lyubartsev, G. Saba, *Molecular dynamics investigation of  $^{23}\text{Na}$  NMR relaxation in oligomeric DNA aqueous solution*, J. Phys. Chem. B., 108, 16295-16302, 2004.
- [87] F. Mocci, G. Saba, *Molecular dynamics simulations of A-T Rich oligomers: sequence specific binding of  $\text{Na}^+$  in the minor groove of B-DNA*, Biopolymers, 68, 471-485, 2003.
- [88] D. Hamelberg, L. Williams, W. Wilson, *Influence of the dynamic position of cations on the structure of the DNA minor groove: Sequence-dependent effects*, J. Am. Chem. Soc., 123 (32), 7745-7755, 2001.
- [89] N.C. Stellwagen, S. Magnusdottir, C. Gelfi, P.G. Righetti, *Preferential counterion binding to A-tract DNA oligomers*, J. Mol. Biol., 305, 1025-1033, 2001.
- [90] P.V. Denisov, B. Halle, *Sequence specific binding of counterions to B-DNA*, PNAS, 97 (2), 629-633, 2000.
- [91] P. Yakovchuk, E. Protozanova, M.D. Kamenetskii, *Base-stacking and base-pairing contributions into thermal stability of the DNA double helix*, Nucleic Acids Res., 34 (2), 564-74, 2006.
- [92] L. Nordenskiöld, D. Chang, C. Anderson, M.J Record, *Sodium-23 NMR relaxation study of the effects of conformation and base composition on the interactions of counterions with double-helical DNA*, Biochemistry, 23, 4309-4317, 1984.
- [93] W.H. Braunlin, *NMR Studies of cation-binding environments on nucleic acids*, Adv. Biophys. Chem., 5, 89-139, 1995.

- [94] P. Cahen, M. Luhmer, C. Fontaine, C. Morat, J. Reisse, K. Bartik, *Study by (23)Na-NMR, (1)H-NMR, and ultraviolet spectroscopy of the thermal stability of an 11-basepair oligonucleotide*, Biophys., 78, 1059-1069, 2000.
- [95] M. Odelius, A. Laaksonen, *Combined MD simulation and NMR relaxation studies of molecular motion and intermolecular interactions. In Molecular Dynamics*, Theo. Comp. Chem., 7, 1999.
- Hubbard, P. S. J. Chem. Phys. 1970, 53, 985-987.
- [96] P.S. Hubbard, *Nonexponential Nuclear Magnetic Relaxation by Quadrupole Interactions*, J. Chem. Phys., 53, 985-987, 1970.
- [97] J. Schnitker, A.Z. Geiger, *NMR Quadrupole Relaxation of Xenon-131 in Water. A Molecular Dynamics Simulation Study*, Phys. Chem., 155, 29-54, 1987.
- [98] T.E. Cheatham, M. Young, *Molecular dynamics simulations of nucleic acids: successes, limitations and promise*, Biopolymers, 56(4), 232-256, 2001.
- [99] M.A. Young, J. Srinivasan, I. Goljer, S. Kumar, D.L. Beveridge, P.H. Bolton, *Structure determination and analysis of local bending in an A-tract DNA duplex: comparison of results from crystallography, nuclear magnetic resonance, and molecular dynamics simulation on d(CGCAAAAATGCG)*, Methods Enzymol., 261, 121-144, 1995.
- [100] F. Mocci, A. Laaksonen, *Insight into nucleic acid counterion interactions from inside molecular dynamics simulations is worth its salt*, Soft Matter, 8, 9268-9284, 2012.
- [101] K. Aidas, H. Agren, J. Kongsted, A. Laaksonen, F. Mocci, *A quantum mechanics/molecular dynamics study of electric field gradient fluctuations in the liquid phase. The case of Na<sup>+</sup> in aqueous solution*, Phys. Chem. Chem. Phys, 15, 1621, 2013.
- [102] M. Young, T.E. Cheatham, *Determination of Alkali and Halide Monovalent Ion Parameters for Use in Explicitly Solvated Biomolecular Simulations*, J. Phys. Chem. B, 112 (30), 9020-9041, 2008.

- [103] X. Mei, K. Zhao, C. Liu, X.W. Chu, *Benchmarking the Memory Hierarchy of Modern GPUs*, 11th IFIP International Conference on Network and Parallel Computing (NPC 2014), Taiwan, September 2014.
- [104] K.R. Seddon *Ionic liquids for clean technology*, J. Chem. Technol. Biotechnol, 68, 351-356, 1997.
- [105] K.Ghandi, *A Review of Ionic Liquids, Their Limits and Applications*, Green and Sustainable Chemistry, 4, 1, 44-53, 2014.
- [106] F. Mocci, A. Laaksonen, Y.L. Wang, G. Saba, A. Lai, F. Cesare Marincola, *CompChem and NMR Probing Ionic Liquids*, The Structure of Ionic Liquids, Soft and Biological Matter, Chap. 4, Springer International Publishing Switzerland, 2014.
- [107] T. Welton, *Room-Temperature Ionic Liquids. Solvents for Synthesis and Catalysis*, Chem. Rev., 99, 8, 2071-2084, 1999.
- [108] S. Ahrens, A. Peritz, T. Strassner, *Tunable Aryl Alkyl Ionic Liquids (TAAILs): The Next Generation of Ionic Liquids*, Angewandte Chemie International Edition, 48, 42, 7908-7910, 2009.
- [109] P. Wasserscheid, W. Keim, *Ionic Liquids New Solutions for Transition Metal Catalysis*, Angewandte Chemie International Edition, 39, 3772-3789, 2000.
- [110] D.M. Fox, W.H. Awad, J.W. Gilman, P.H. Maupin, H.C. De Long, P.C. Trulove, *Flammability, thermal stability, and phase change characteristics of several trialkylimidazolium salts*, Green Chemistry, 5, 724-727, 2003.
- [111] C.L. Hussey, *Room temperature haloaluminate ionic liquids. Novel solvents for transition metal solution chemistry*, Pure and Applied Chemistry, 60, 1763-1772, 1988.
- [112] E.J. Maginn, *Molecular simulation of ionic liquids: current status and future opportunities*, J. Phys. Condens. Matter, 21, 373101, 2009.
- [113] A.L. Sturlaugson, K.S. Fruchey, M.D. Fayer, *Orientational Dynamics of Room Temperature Ionic Liquid/Water Mixtures: Water-Induced Structure*, J. Phys. Chem. B, 116, 1777-1787, 2012.

- [114] Z. Liu, S. Huang, W. Wang, *A Refined Force Field for Molecular Simulation of Imidazolium-Based Ionic Liquids*, *J. Phys. Chem. B*, 108, 12978-12989, 2004.
- [115] Y.L. Wang, A. Lyubartsev, Z.Y. Lu, A. Laaksonen, *Multiscale coarse-grained simulations of ionic liquids: comparison of three approaches to derive effective potentials*, *Phys. Chem. Chem. Phys.*, 15, 7701-7712, 2013.
- [116] K. Heinzinger, *Computer simulations of aqueous electrolyte solutions*, *Physica*, 131B, 196, 1985.
- [117] F.G. Fumi, M.P. Tosi *Ionic sizes and born repulsive parameters in the NaCl-type alkali halides—I: The Huggins-Mayer and Pauling forms*, *J. Phys. Chem. Solid*, 25, 31, 1964.
- [118] F. Dommert, K. Wendler, R. Berger, L. Delle Site, C. Holm, *Force Fields for Studying the Structure and Dynamics of Ionic Liquids: A Critical Review of Recent Developments*, *ChemPhysChem*, 13, 1625-1637, 2012.
- [119] A. Bagno, F. D'Amico, G. Saielli, *Computing the  $^1\text{H}$  NMR Spectrum of a Bulk Ionic Liquid from Snapshots of Car-Parrinello Molecular Dynamics Simulations*, *ChemPhysChem*, 8, 873-881, 2007.
- [120] S. Badu, L. Truffandier, J. Autschbach, *Quadrupolar NMR Spin Relaxation Calculated Using Ab Initio Molecular Dynamics: Group 1 and Group 17 Ions in Aqueous Solution*, *J.Chem.TheoryComput.*, 9, 4074-4086, 2013.
- [121] I. Skarmoutsos, D. Dellis, R.P. Matthews, T. Welton, P.A. Hunt, *Hydrogen Bonding in 1-Butyl- and 1-Ethyl-3-methylimidazolium Chloride Ionic Liquids*, *J. Phys. Chem. B*, 116, 4921-4933, 2012.
- [122] Z. Liu, T. Chen, A. Bell, B. Smit, *Improved United-Atom Force Field for 1-Alkyl-3-methylimidazolium Chloride*, *J. Phys. Chem. B*, 114, 4572-4582, 2010.
- [123] R.C. Remsing, G. Hernandez, R.P. Swatloski, W.W. Masefski, R.D. Rogers, G. Moyna, *Solvation of Carbohydrates in N, N#-Dialkylimidazolium Ionic Liquids: A Multinuclear NMR Spectroscopy Study*, *J. Phys. Chem. B*, 112, 35, 11071-11078, 2008.

- [124] B.L. Bhargava, S. Balasubramanian, *Intermolecular structure and dynamics in an ionic liquid: A Car-Parrinello molecular dynamics simulation study of 1,3-dimethylimidazolium chloride*, Chem. Phys. Lett., 417 486-491 , 2006.
- [125] M.G. Del Popolo, R.M. Lynden-Bell, J. Kohanof, *Ab Initio Molecular Dynamics Simulation of a Room Temperature Ionic Liquid*, J. Phys. Chem. B, 109, 5895-5902, 2005.
- [126] M. Buhl, A. Chaumont, R. Schurhammer, G. Wipff, *Ab Initio Molecular Dynamics of Liquid 1,3-Dimethylimidazolium Chloride*, J. Phys. Chem. B, 09, 18591-18599, 2005.
- [127] C.G. Hanke, S.L. Price, R.M. Lynden-Bell, *Intermolecular potentials for simulations of liquid imidazolium salts*, Molecular Physics, 99, 10, 801-809, 2001.
- [128] <http://www.chemcas.com/material/cas/archive/171058-18-7.asp>.
- [129] V. Klimavicius, Z. Gdaniec, V. Balevicius, *Very short NMR relaxation times of anions in ionic liquids: New pulse sequence to eliminate the acoustic ringing*, Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy, 132, 879-883, 2014.
- [130] D.L. Bergman, L. Laaksonen, A. Laaksonen *Visualization of solvation structures in liquid mixtures*, Journal of Molecular Graphics and Modelling, 15, 5, 301-306, 1997.
- [131] A. Atzori, S. Liggi, A. Laaksonen, A. Lyubartsev, G. Saba, M. Porcu, F. Mocchi, *Molecular Dynamics Investigations on Base Sequence Specificity of Counterion Binding to DNA*, Biophysical Journal, 102, Issue 3, 637a, Published in issue: January 31, 2012.