Modélisation moléculaire de matrikines et protéines matricielles : Approches théoriques

THÈSE

présentée et soutenue publiquement le
21 décembre 2012

pour l'obtention du

Doctorat de l'Université de Reims Champagne-Ardenne
(Spécialité : Biophysique Moléculaire)

par

Thomas HASCHKA

Composition du jury

Président :	M.	Eric HENON	Professeur, Université de Reims Champagne-Ardenne
Rapporteurs :	M.	Othmar STEINHAUSER	Professeur, Université de Vienne (Autriche)
	M.	Jérôme GOLEBIOWSKI	Docteur, Université de Nice Sophia Antipolis (Nice)
Examinateurs :	Mme.	Monica ZOPPÈ	Professeur, Consiglio Nazionale delle Ricerche Pisa (Italie)
	Mme.	Catherine ETCHEBEST	Professeur, Université Paris Diderot (PARIS 7)
	Mme.	Sylvie RICARD-BLUM	Professeur, Université Claude Bernard (LYON 1)
Directeurs :	M.	Laurent MARTINY	Professeur, Université de Reims Champagne-Ardenne
	M.	Manuel DAUCHEZ	Professeur, Université de Reims Champagne-Ardenne
Invité :	M.	François Yves DUPRADEAU	Professeur, Université de Picardie Jules Verne (Amiens)

Contents

CONTENTS

1 | Résumé

1.1 Introduction

La recherche autour de la *Thrombospondine* (TSP) a été particulièrement intéressante ces dernières années. La protéine a été découverte en 1971 par Baenziger et al [1]. Elle est impliquée dans des processus physiologiques comme l'inflammation, l'angiogenèse, la synaptogenèse du système nerveux central, la guérison de blessures, la formation du cartilage, la prolifération et la néoplasie. Cinq différents types de TSP ont été identifiés dans les gènes humains. Ils forment la famille des TSPs. Ces protéines sont trimeriques ou pentameriques et sont situées dans la matrice extracellulaire. Elles ont une partie commune à leur C-terminal appelée le *Domaine de Signature* (DS). Cette partie est impliquée dans des fonctions diverses de la TSP. Elle est entre autre importante pour l'attachement aux cellules et pour la liaison au *Cluster of Differentiation 47* (CD-47) également connu sous le nom *Integrin Associated Protein* (IAP) dans la littérature. Environ 30 ions calcium par monomère peuvent se lier à la TSP. Il a été démontré qu'une partie ces ions sont détachables et échangeables sur cette protéine. Il a été retrouvé que la conformation et les fonctions de celle-ci sont dépendantes des concentrations en calcium. Jusqu'à aujourd'hui la compréhension de l'implication du calcium reste cependant mal connue.

Le travail décrit dans ce document a comme but de révéler le fonctionnement de la TSP DS. Pour cela des différentes méthodes ont été développées. Il a également fallu utiliser des approches différentes issues de domaines scientifiques divers. Dans ce travail, de nouvelles méthodes de communication scientifique ont été également développées et utilisées. Ce travail est divisé en quatre étapes qui nous ont permis d'aborder et de résoudre les problèmes posés sous différents angles montrant ainsi la pluri-

disciplinarité de ce travail. Les quatre étapes sont :

1. La biologie des thrombospondines :
   Cette partie résume les connaissances biologiques connues aujourd'hui. Elle pose des questions qui sont traitées dans ce travail. Un effort significatif a été fait pour condenser et récapituler la connaissance autour de ces protéines. Ceci inclut la description de la famille entière de ces protéines qui possèdent des fonctions très variées. Cette section traite de la génétique, des structures et de l'implication des TSP dans diverses maladies et leurs traitements connus à ce jour. Comme il existe des interactions importantes associées aux maladies qui sont encore mal établies en biologie moléculaire, et en général la dynamique des ces protéines n'étant pas connue. Nous avons fait appel à la dynamique moléculaire pour retrouver les changements conformationels et les fonctions de ces protéines associées.
2. La simulation par dynamique moléculaire classique :
   Dans cette partie le problème est traité sous l'angle de la mécanique classique à l'échelle atomique. Plusieurs simulations par les méthodes de la *dynamique moléculaire* (DM) classique ont été faites afin de mieux comprendre la dynamique de la DS et d'être capable de décrire le processus de déplétion des ions calcium. Cette partie introduit les éléments de base théoriques nécessaires et utiles pour créer et évaluer ces simulations. Elle traite des champs de forces et de leurs intégrations. Elle explique comment les ensembles statistiques sont générés pendant ces simulations, comment nous avons analysé et interprété les trajectoires obtenues par ces simulations. Finalement dans cette partie nous présentons les résultats obtenus par ces méthodes, comme l'identification des ions calcium échangeables de la TSP DS. Un problème majeur vient de la définition des champ de forces empiriques et particulièrement de la paramétrisation du calcium. Les champs de forces utilisés n'étant pas polarisables ne sont pas adaptables à traiter les interaction entre les protéines et les ions divalents. C'est pourquoi nous avons fait appel à la mécanique quantique.
3. La chimie quantique :
   Dans cette partie, nous traitons le problème d'un point de vue de la physique moderne. Nous descendons à l'échelle

sous- atomique, et mettons les propriétés des couches électroniques autour des parties des TSPs au coeur de la recherche. Ici nous décrivons comment on obtient une solution de l'équation de Schrödinger pour des molécules en utilisant la méthode du *champ Self- Consistent* (SCF) également connue sous le nom méthode de *Hartree Fock* (HF). Nous continuons par décrire comment cette méthode est intégrée d'une manière intelligente avec une autre la *Théorie de la Fonctionnelle de la Densité* (DFT) pour former les méthodes dites *Fonctionnelles Hybrides*, comme celle assez populaire *Becke, trois paramètres, Lee-Yang-Parr* (B3LYP) pour obtenir des résultats significatifs sur les molécules biologiques. Une fois la densité électronique obtenue par ces méthodes nous calculons les charges partielles afin de vérifier, et de réinjecter ces résultats dans les champs de forces de la DM classique. Pour être capables de calculer ces charges partielles nous avons déduit et implémenté une nouvelle méthode à partir de la *méthode par potentielle électrostatique* (ESP) que nous avons nommée *multicube*. Pour la création de ce logiciel, nous avons utilisé les méthodes contemporaines comme la programmation parallèle massive qui entre autre, nous permet de d'utiliser la puissance des *processeurs graphiques* (GPUs) qui font partie des machines de calculs modernes. Une telle approche sophistiquée est nécessaire dès que la DM classique ne prend pas en compte les processus physiques comme le transfert des charges ou la polarisation sur la surface moléculaire. Ces deux effets sont nécessaires pour décrire les liaisons des ions calcium correctement. De ces deux chapitre de simulations, il apparaît que les données de simulation obtenus sont à la fois très consèquentes en volume et difficiles à visualiser. Nous nous sommes alors attachés à produire des outils de visualisation nous permettant d'éffectuer ce démarches, tout en y associant une approche artistique.

4. La visualisation moléculaire : Ceci constitue la partie artistique de ce document. Ici, nous décrivons la visualisation des molécules biologiques et les calculs quantiques autour d'eux. La visualisation est importante puisqu'elle nous permet de mieux communiquer nos résultats avec nos partenaires. Ainsi nous progressons plus rapidement dans notre

recherche. Cette partie traite entre autre du développement d'un nouveau procédé afin de visualiser les résultats obtenus en chimie quantique grâce à l'utilisation du logiciel libre *Blender*. En outre, nous approfondissons nos créations de films sur la TSP DS autour de notre recherche. Ceux-ci ont été présentés à divers congrès internationaux et artistiques.

Pour terminer nous soulignerons les résultats de ce travail et proposerons de futurs travaux afin de progresser dans la recherche autour de la TSP DS. Les chapitres sont resumés un par un ci-dessous.

1.2 La biologie des thrombospondines

Les *Thrombospondines* (TSPs) forment une famille de protéines qui se trouvent dans la matrice extracellulaire. Ces molécules de grandes tailles sont impliquées dans une variété de fonctions biologiques. Chez l'humain cinq différents types de TSPs ont été identifiés, et plusieurs types différents existent également chez les animaux comme chez les insectes [2]. Les cinq TSPs chez l'humain peuvent être classés en formes trimériques et pentamériques. La TSP-1 et 2 sont des molécules trimériques et les TSP-3, 4 et 5 existent sous la forme pentamérique. La TSP-5 est aussi connue sous son nom *Cartilage Oligomeric Martrix Protein* (COMP). Ces protéines de grandes tailles sont composées par de nombreux domaines. La séquence d'une monomère d'une molécule de la famille des TSPs peut atteindre jusqu'au 1154 acides aminés. Toutes les TSPs ont un *Domaine de Signature* (DS) en commun. Dans ce domaine, la séquence est hautement conservée. La TSP-1 est la protéine de la famille la mieux connue. Elle a été découverte par Benzinger et al. [1]. Elle porte son nom parce qu'elle a été trouvé comme étant une protéine qui est relâchée par les platelets ou *thrombocytes* qui sont stimulés par la trombine. Depuis cette découverte, les TSPs ont trouvées d'être exprimées par des nombreuses cellules, entre autres par les cellules endothéliales, les fibroblastes, les adipocytes, les cellules musculaires lisses, les monocytes et les macrophages. Ici nous présentons quelques implications, sans être exhaustif, que nous avons trouvé intéressantes. Dans le reste du chapitre nous faisons la liaison de ces fonctions avec la structure de la TSP en rentrant plus dans les détails de la biologie moléculaire de la

protéine.

L'inflammation [3] [4]

L'inflammation est une réponse d'un tissu vivant à une blessure locale. En générale dans ce cas, la TSP-1 est exprimée plus fortement. Plusieurs récepteurs de la TSP-1 sont nécessaires pour la régulation de l'inflammation. La TSP-1 interagit avec les protagonistes variés pour activer les signaux qui dirigent l'inflammation. La TSP-1 a des effets anti et pro-inflammatoires, qui ont été observés *in vivo* dans les modèles d'animaux différents. Les souris en déficience de TSP-1 par exemple sont reconnues pour avoir la pneumonie et la colite aiguë et encore beaucoup d'autres anomalies qui sont liées à l'inflammation.

La prolifération [5] [6]

Les TSPs avec d'autre molécules jouent un rôle crucial pour la prolifération. La TSP-1 comme la TSP-2 ont été trouvées comme étant promoteurs de la prolifération de certains types de cellules. Ichii et al [5] ont démontré *in vitro* que la TSP-1 est impliquée dans la prolifération des cellules des muscles lisses. N. Lopez et al [6] ont retrouvé que la TSP-2 est par ailleurs un régulateur de la prolifération des cellules endothéliales.

La guérison des blessures [7] [8]

La guérison des blessures est un processus complexe qui prend en compte des étapes différentes qui se recouvrent partiellement. Ces étapes sont l'inflammation, la prolifération et la remodélisation du tissu. Comme décrit ci-dessus, les TSPs sont importantes dans les premières deux étapes. La TSP-1 et la TSP-2 ont été trouvés comme étant impliquées dans le processus totale de la guérison des blessures. La TSP-1 est présente lors des premières étapes et joue un rôle important pendant l'inflammation. La TSP-2 est exprimée plus fortement plus tard pendant la guérison d'une blessure. Les deux protéines sont donc impliquées dans la réparation d'une blessure. Les souris qui sont déficientes en TSP-2 montrent une guérison plus lente. Les blessures guéries de ces souris ont une concentration plus forte des vaisseaux sanguins et une matrice extracellulaire peu organisée.

L'angiogenèse [9] [10]

L'angiogenèse décrit le processus de la croissance des vaisseaux sanguins. Elle est régulée par les protéines différentes qui agissent comme un promoteur ou comme un répresseur de l'angiogenèse. Les deux, la TSP-1 comme la TSP-2 jouent un rôle actif comme promoteur ou represseur de l'angiogenèse. La TSP-1 a été par exemple trouvée à induire l'apoptose, d'inhiber la migration des cellules et la prolifération dans les cas des cellules endothéliales. Le contrôle de l'angiogenèse est un facteur crucial pour combattre le cancer. Plusieurs interventions thérapeutiques agissent sur le taux d'expression des TSPs. L'utilisation des peptides créés à partir des séquences peptidiques comme ABT-510 [11] de la TSP a été proposé pour contrôler l'angiogenèse.

L'apoptose [12] [13] [14]

L'apoptose décrit la mort programmée d'une cellule. L'induction d'apoptose joue un rôle important dans le renouvellement des tissus et dans le traitement des cancers. Certains médicaments utilisés lors de chimiothérapie sont entre autres crées pour introduire l'apoptose. La TSP a un double rôle dès lors qu'elle peut être un inhibiteur ou un promoteur de l'apoptose. TSP-1 peut par exemple inhiber l'apoptose introduite par le camptothécine ou le doxorubicine [12] dans les cellules carcinoïdes de la thyroïde. L'effet inverse a été retrouvé pour les cellules carcinoïdes du côlon [13] où la TSP peut introduire l'apoptose. Ceci a été également retrouvé pour les cellules endothéliales [14].

La chondrogenèse [15]

Il a été retrouvé que la TSP-5 a un rôle important dans la croissance des os. Les nombreux sites de mutations dans la séquence du TSP-5 ont été identifiés étant la cause pour la maladie héréditaire dite *Pseudoachondroplasia* (PSACH) et sa forme moins sévère *Multiple epiphyseal dysplasia* (MED). PSACH est mieux connue sous son nom commun *le nanisme*. Les deux maladies sont les causes d'une croissance irrégulière du cartilage qui fait que les os sont mal formés. La TSP-5 est impliquée en agissant ensemble avec d'autres protéines dans la formation du cartilage. Les protéines mutantes causent ces irrégularités dans la forma-

tion qui engendre ces maladies héréditaires.

La synaptogenèse du système nerveux central

La TSP-1 et TSP-2 sont sécrétées par les astrocytes. Il a été découvert que l'expression du TSP-1 et 2 est régulée dynamiquement pendent le développement du cerveau [16]. La concentration du TSP dans le cerveau embryonnaire est faible, elle est plus importante dans le cerveau postnatal et presque absente dans le cerveau adulte. K. S. Christopherson et al [17] ont démontré que la formation des synapses peut être engendrée par la TSP-1 dans un médium conditionné par des astrocytes et dans les cellules du ganglion rétinien des rats. Ils ont également démontré que même si les souris qui sont déficientes en TSP-1 ou en TSP-2 n'ont pas une plus faible quantité des synapses dans leur cerveau mais que la nombre des points synaptiques peut être 40% plus faible dans ces souris pendant certaines étapes du développement du cortex cérébrale que dans les souris mutantes qui n'expriment ni de la TSP-1 ni de la TSP-2.

L'obésité [18]

Une étude menée par Y. Li et al a démontré que le TSP-1 joue un rôle important dans l'obésité. Les souris transgéniques qui sont incapables d'exprimer la TSP-1 et qui étaient nourries avec un régime fortement gras développent l'obésité dans le même temps que les souris sauvages. La TSP-1 n'est donc pas impliquée dans le développement de l'obésité, mais les souris déficientes en TSP-1 ont montré qu'elles plus tolérantes au glucose et plus sensibles à l'insuline. Ces souris sont également plus faibles en accumulation des macrophages dans leur tissu adipeux. Y. Li et al proposent que la TSP-1 est responsable pour cette accumulation des macrophages dans ce tissu et qu'avec d'autres effets, TSP-1 induit l'inflammation du à l'obésité qui est la cause pour la résistance à l'insuline.

La Tumorogenèse.

De nombreux rôles complexes de la TSP-1 causent la formation des cellules tumorales et la métastase sont connus. Suivant le tissu et le type de cellule où se trouve la protéine, elle peut jouer soit le rôle d'un promoteur de tumeur, soit le rôle d'un ré-

presseur. Il a été par exemple retrouvé qu'il peut être un inhibiteur de l'angiogenèse tumorale. Encore que dans les cas précédents la TSP interagit avec une multitude des protéines de la matrice extracellulaire ou des protéines ancrées dans la membrane cellulaire où elle s'attache et induit la transduction des signaux. Les fonctions qui sont liées à la tumorogenèse sont particulièrement importantes dès que les nouveaux médicaments pour le traitement du cancer sont modélisés à partir des peptides du TSP-1. ABT-510 [19] [20], ABT-526 [21] [22] et ABT-898 [23] sont des molécules modélisées à partir des peptides du TSP qui se trouvent dans les étapes différentes d'essais cliniques pour une future utilisation comme pour le traitement du cancer.

Dans ce chapitre, nous faisons la liaison entre les fonctions et la structure de la TSP. La TSP est composée d'une multitude de domaines. Pour les différentes TSP qui se trouvent dans le corps humain cette composition est montrée en figure 3.1. Toutes les TSPs sont composées d'au moins trois répétitions de EGF-like, d'un domaine riche en calcium appelé le *Stalk* et d'un domaine globulaire au C-terminal appelé *Globe*. Ces parties qui sont en commun à toutes les TSPs s'appelle le *Domaine de Signature* (DS). Ce domaine est au coeur de la recherche effectuée dans ce travail. Elle contient des sites de fixations importantes, comme celle du CD-47 [24], ou aux diverses integrines [25] [26]. Ces interactions jouent des rôles importants et sont par exemple déterminant dans la tumorogenèse ou la préservation de la santé vasculaire en cas d'inflammation [27]. Ces processus sont encore mal compris dans sous l'angle de la biologie moléculaire. La TSP DS contient de nombreuses ions de calcium et la structure de ce domaine et sa fonction est dépendante de la concentration du calcium. Nous avons donc fait appel aux approches théoriques de dynamique moléculaire pour étudier ces processus.

1.3 La simulation en dynamique moléculaire classique

La *Dynamique Moléculaire* (DM) classique forme un ensemble de méthodes théoriques, qui peuvent être implémentées dans des logiciels, et qui permettent d'étudier les propriétés physiques d'un ensemble d'atomes et de molécules. Ces ensembles peuvent être des gaz, liquides ou des molécules biologiques comme dans

notre cas. L'origine de ces simulations se trouve dans les années 50 du 20ème siècle dans le domaine de la physique théorique [28]. Avec l'augmentation de la puissance des ordinateurs les intérêts dans la DM se sont multipliés et aujourd'hui des simulations en DM se font dans de nombreuses disciplines scientifiques. Aujourd'hui il existe même des machines spécialement construites pour la DM [29]. Dans notre cas nous avons utilisé la DM pour mieux comprendre le comportement dynamique de la *Domaine de Signature* (DS) de la *Thrombospondine* (TSP).

Comme dans toutes les simulations, qui sont par définition qu'une approximation de la réalité, ils existent plusieurs inconvénients. La description d'un système est limité par le temps du calcul alloué, et il faut trouver un modèle assez exact pour décrire les processus que l'on veut étudier sous contrainte de la puissance calculatrice disponible. Cette tâche est sûrement une des plus compliquées en effectuant ces simulations. Jusqu'à aujourd'hui l'exactitude des simulations de ce type fait débat.

La DM classique, comme on peut facilement le déduire de son nom, se limite aux principes de la physique classique. Elle prend en compte ni la mécanique quantique ni la relativité. Aujourd'hui, ces types de simulations contiennent de nombreux algorithmes qui touchent la théorie classique de l'électromagnétisme, des autres interactions interatomiques comme l'interaction de *van der Waals* et une grande partie de la mécanique statistique. Ces algorithmes sont aujourd'hui implémentés dans des nombreux logiciels comme *GROMACS* [30] où *NAMD* [31]. Dans ce chapitre nous décrivons la théorie que nous avons utilisée dans les simulations que nous avons réalisées.

Ayant décrit la théorie utilisée dans les logiciels que nous avons utilisé en effectuant nos simulations, nous décrivons comment ces simulations sont faites en pratique. Nous montrons les problèmes que nous avons rencontré en choisissant les bons paramètres, comme les champs de forces et les ensembles statistiques pour simuler un domaine de grande taille comme la TSP DS.

Une fois les simulations effectuées, leur analyse est un travail important et fondamental pour comprendre et interprêter le comportement moléculaire. Comme notre intérêt se concentre sur les effets dynamiques de la TSP DS, nous avons donc développé des modules et méthodes qui permettent de répondre à cette problématique. Un de ces modules est par exemple écrit

pour étudier comment la sphère de coordination d'un ion calcium change au cours d'une simulation.

Nous avons fait 66 simulations (c.f. tableaux 4.1 et 4.2) entièrement dont 64 des parties différentes de la TSP DS (3,5µs en total). Par ces simulations nous avons obtenu des résultats divers. Nous avons étudié la flexibilité de ce domaine et nous avons retrouvé que le *Stalk* et les répétitions EGF-like semblent très flexibles par rapport au *Globe*. Nous avons également pu montrer que la flexibilité des répétitions EGF-like est liée à la fixation d'un ion calcium. Dans les simulations où l'ion situé entre la première et la deuxième répétition EGF-like est solvaté, ces deux domaines montrent une augmentation dans leur flexibilité. Pour le *Stalk* et le *Globe* nous n'avons pas pu identifier une relation entre la solvatation des ions dans ces régions et la flexibilité. Le *Stalk* a semblé en générale assez flexible même si les ions ne sont pas partis. Sur le *Globe* qui est assez rigide par rapport aux autres parties de la DS nous avons pu identifier des boucles flexibles qui interagissent également avec des ions calcium. Le résultat le plus important est probablement l'identification des ions échangeables. Sur cette partie qui contient autour de 30 ions nous avons pu identifier le sites de 10 ions échangeables. Cela signifie qu'ils peuvent être facilement solvatés, pendant que les autres ions sont plus fixés à la protéine. Une identification des ions est donnée en figure 3.6. Les probabilités qu'un ion parte sont démontrés en figure 4.5 et 4.6.

De plus nous avons également fait des liaisons avec des modèles biologiques existants de structures qui contiennent une faible quantité d'ions calcium [32]. Nous avons montré qu'en créant ces modèles, on devrait différencier entre les différents membres de la famille des TSPs. Nous avons par exemple montré que les interactions entre le *Stalk* et *Globe* ont un comportement dynamique différent entre les différents TSP-1,2 et 5.

Dans ce chapitre nous montrons également la faiblesse des champs de forces. Nous montrons que les ions calcium ont des comportements différents selon les champs de forces. Nous constatons que les champs de forces classiques sont mal adaptés pour traiter les problèmes que nous rencontrons, dès qu'ils sont incapable de prendre des effets comme les transferts de charges ou la polarisation en compte. Pour faire face à ces problèmes nous avons fait appel aux méthodologies de la physique moderne, la mécanique quantique moléculaire.

1.4 La mécanique quantique moléculaire

Comme nous l'avons discuté dans le précédent paragraphe la *Dynamique Moléculaire* (DM) classique pose de nombreux problèmes si l'on veut l'utiliser pour effectuer des simulations des systèmes biologiques complexes comme le *Domaine de Signature* (DS) de la *Thrombospondine* (TSP). Le nombre important des ions calcium divalents et l'effet de polarisation engendré par ceux-ci sont mal décrits dans le contexte de la DM classique. C'est à cause de cette difficulté que nous avons choisi d'utiliser les méthodes de *Mécanique Quantique* (MQ) moléculaire. La MQ moléculaire a la capacité de calculer la probabilité de trouver un électron autour d'une molécule dans une certaine région d'espace. C'est-à-dire que nous descendons à l'échelle sous-atomique où les couches électroniques d'une molécule peuvent être explicitement calculées. La raison pour laquelle nous avons choisi cette démarche est que cela nous permet d'étudier en détail les effets de polarisation qui sont engendrés par des ions divalents comme le calcium à la surface de la TSP DS. Ceci nous permet d'établir une mesure d'erreurs dans les champs de forces et dans la DM classique en générale qui sont dues à la polarisation ou les transferts de charge. Dans nos études, nous avons utilisé deux méthodes différents, la méthode dits *Champ self-consistent* (SCF) également connu sous le nom *Hartree Fock* (HF) et la méthode de la *Théorie de la Fonctionnelle de la Densité* (DFT) et les méthodes dites *Fonctionnelles Hybrides* qui ont été développées à partir de ces deux-là. Ces méthodes numériques sont capables de donner une solution approximative pour l'équation de Schrödinger indépendante du temps. À partir de ces solutions, on peut ainsi déduire la probabilité de trouver un électron dans une certaine région d'espace, une information aussi appelée le nuage électronique. Comme la DM classique ne traite pas des électrons explicitement et attribut les charges monopolaires aux objets comme à des atomes ou à des gros grains dans les simulations nous avons du trouver une méthode pour mener les résultats de la QM moléculaire à l'échelle atomique de la DM classique. Nous avons alors essayé d'attribuer les charges monopolaires aux objets comme à des atomes ou à des grains grossiers qui sont utilisés dans la DM classique à partir du nuage électronique obtenu. Avec ces méthodes disponible pour faire ce type de

calculs comme la méthode dite *Electrostatic Potential* (ESP) [33] ou *Mulliken* [34], nous n'étions pas capables d'obtenir des résultats raisonnables. Pour faire face à ce problème nous avons développé et implémenté un nouvel algorithme basé sur la méthode ESP pour dériver les charges partielles. Nous avons pendant ce développement utilisé les nouvelles technologies de programmation parallèle. Le langage OpenCL nous a permis de paralléliser et d'implémenter notre algorithme sur les différents puces de calculs qui se trouvent dans un ordinateur actuel comme les unités centrales multicoeurs (CPU)s, les unités graphiques, (GPU)s ou des cartes accélératrices. En appliquant ce nouvel algorithme nous avons obtenu des résultats préliminaires sur une répétition du *Stalk* de la TSP DS. Nous avons également résolu plusieurs problèmes qui se posent autour du calcul des charges partielles et nous avons fait des propositions à ce sujet.

Nous nous avons posé la question comment peut-on visualiser les solutions obtenues par la MQ moléculaire ? Nous avons constaté que par des méthodes dédiés à la visualisation des volumes on devrait être capable de montrer ces résultats. Nous avons donc créé des procédés de visualisation décrits dans le prochain chapitre.

1.5 La visualisation moléculaire

La visualisation moléculaire est l'une des méthodes clés dans la compréhension d'un processus moléculaire comme ceux qui ont été décrits dans ce document. Dès que les premières structures ont été résolues par diffraction des rayons X, on s'est intéressé à leur visualisation. Le premier modèle d'une structure est attribué à J. C. Kendrew et al [35] et a été publié en 1958. Les premiers modèles étaient construits mécaniquement et la première visualisation sur ordinateur a été fait en 1966 [36]. Comme la structure d'une protéine ou d'autres molécules biologiques de grand taille est liée à la fonction de ceux-ci. Leur visualisation est une partie importante de la recherche. Plusieurs résultats présentés dans les chapitres 4 et 5 sont obtenus grâce à la visualisation. Les représentations graphiques font aujourd'hui partie de la recherche courante.

La visualisation n'aide pas seulement dans la compréhension de la fonction d'une molécule mais est aussi un outil performant

pour communiquer les problèmes et les solutions concernant ces molécules. Un film d'une molécule qui bouge peut résumer un changement conformationnel, la formations des liaisons aux ligands et d'autres fonctions. La visualisation a également un aspect esthétique. Elle est donc aussi un outil qui joue un rôle important dans la vulgarisation de la science.

Pendant nos travaux nous avons essayé de résoudre plusieurs problèmes par les méthodes de visualisation. Nous avons rapidement rencontré les limites des outils disponibles comme *Visual Molecular Dynamics* (VMD) [37] ou *PyMol* [38]. Pour franchir ces limites nous avons utilisé et développé de nouvelles approches que nous allons présenter dans ce chapitre.

Nous avons d'abord évaluer les méthodes pour créer des images, films *photoréalistes* à partir des structures et résultats obtenus en dynamique moléculaire. Nous avons introduit un outil qui permet d'integrér le *motion blur* la floutage d'une image à partir de la vitesse d'un objet qui est sur la scène d'un film. Nous avons écrit des scripts divers pour rendre les résultats obtenus en PCA accessibles pour le spectateur. Pour un rendu encore plus amélioré nous avons fait appel au logiciel *Blender* [39].

Mais la visualisation ne s'arrête pas aux structures de la TSP DS et celle des simulations faites par des méthodes de la DM. La visualisation peut aussi montrer des résultats de la MQ moléculaire. La MQ moléculaire donne accès au nuage électronique d'une molécule. Nous avons été particulièrement intéressés à visualiser ce nuage électronique. Cette tâche était compliquée dès que les molécules en question, les répétitions du *Stalk* de la TSP SD, sont pour ces approches des molécules relativement grandes qui contiennent autour de 200 d'atomes. Nous avons cependant avec *Blender* réussi à visualiser le nuage électronique de ces objets et nous montrons dans ce texte les premières images obtenues.

Nous avons utilisé les images avec succès lors des présentations aux séminaires internes et aux congrès internationaux. Nous pouvons présenter nos travaux non seulement aux congres scientifiques, mais aussi artistiques. Ceci présente la fin des nos travaux que nous allons conclure dans la section suivante.

1.6 Conclusion

Pendant ce travail nous avons abordé des disciplines différentes de la science. Nous avons même pu rapprocher un peu l'art de la science. Comme démontré précédemment nous avons commencé par la biologie de la *Thrombospondine* (TSP) et plus particulièrement son *Domaine de Signature* (DS). Nous avons continué en étudiant cette molécule par les méthodes de la *Dynamique Moléculaire* (DM). Etant parvenu aux limites de ce qui se peut faire par DM, à cause des spécificités de la TSP, notamment le nombre élevé des ions calcium qui se fixent à cette protéine, nous avons abordé la *Mécanique Quantique* (MQ) moléculaire. En utilisant la MQ moléculaire nous avons étudié la densité électronique d'une certaine région de la TSP DS qui a abouti à des résultats préliminaires. De futurs travaux seront nécessaires pour traiter correctement et analyser ces résultats. Nous avons également développé un nouvel algorithme pour dériver les charges partielles. Pendant ce développement nous avons utilisé les nouvelle technologies comme la programmation parallèle massive en utilisant la langage OpenCL. Ceci nous a permis de faire tourner notre algorithme pour la dérivation des charges partielles aux puces des calculs différents comme les *Processeurs Graphiques* (GPU)s. Mise à part toutes ces méthodes nous avons également travaillé sur la visualisation moléculaire. Les outils et les démarches que nous avons mis en place pour mieux visualiser les molécules nous ont permis de mieux comprendre leurs fonctions. Comme décrit, la visualisation joue aussi un rôle artistique et apporte une valeur esthétique. Nos travaux ont été non seulement sélectionnés pour des contenus scientifiques mais aussi pour des concepts artistiques, et nous avons pu les présenter lors de congrès scientifiques, comme aux congrès artistiques voir congrès mixtes. Ceci montre bien combien de domaines différents de la recherche, on doit traiter pour comprendre un objet minuscule, une sous unité d'une protéine. Beaucoup des choses sont dites sur les méthodes utilisées et les résultats obtenus et sont décrits et discutés dans les autres chapitres.

La question qui peut se poser maintenant est comment ce travail devrait continuer, comment le futur de la modélisation de la TSP et de sa SD sera-t-il ? Nous nous permettons de mettre quelques lignes concernant ce sujet, avant de finir ce travail en

remarquant quelques aspects originaux :

On devrait savoir que presque toutes les structures qui composent les domaines des TSPs sont connues aujourd'hui. Malheureusement quelques structures restent inconnues à cause de leur flexibilité. Nous croyons que l'on pourrait pas seulement descendre de l'échelle dynamique à l'échelle sous- atomique, on devrait également monter à l'échelle mécanique où les objets sont décrits par les méthodes de la mécanique des milieux continus pour décrire les protéines comme la TSP. Même si cela n'a pas été fait dans ce travail nous supposons que cette nouvelle façon de traiter ces problèmes pourrait aboutir à des résultats intéressants.

Cependant nous avons aussi remarqué l'ambiguïté de la liaison du TSP au *Cluster of Differentiation* (CD-47). Nous avons souligné que les peptides qui sont connus de se lier au CD-47 sont enfouis dans la structure de la TSP. Comme cette interaction est importante pour les fonctions biologiques diverses, nous supposons que des futurs travaux devront également traiter cette problématique.

De plus, on pourrait encore envisager beaucoup d'éventualités, mais nous nous arrêtons ici pour mettre un dernier point sur quelques résultats clés de ce travail.

Un des résultats très intéressant est probablement l'identification des ions *échangeables* sur le *Stalk* de la TSP DS qui font partie des premières connaissances de la dynamique de cette molécule. Ceci est décrit en détail dans le chapitre 4. Un autre résultat important de ce travail est probablement l'algorithme que nous avons développé pour obtenir les charges partielles. Cette nouvelle approche se différencie de celles existantes. Cet algorithme est détaillé au chapitre 5.

Pour finir nous espérons que ce travail sera une contribution à la recherche et que le lecteur aura trouvé plus d'aspects intéressants que ceux mentionnés ici. Nous sommes également intéressés de voir ce que les autres vont faire avec ces travaux qui ont démarré pendant cette thèse dans plusieurs domaines et où ces travaux nous emmènerons.

2 | Introduction

The research around *Thrombospondin* (TSP) proteins has been flourishing in the recent years. The protein was first discovered by Baenziger et al in 1971 [1], and is implicated in several different physiological processes such as inflammation, angiogenesis, central nervous system synaptogenesis, wound healing, cartilage formation, proliferation and neoplasia. In humans, five different types of TPSs on five different genes do exist. These protein molecules form the TSP family. TSPs are large trimeric or pentameric extracellular matrix proteins. Common to them is the so called *Thrombospondin Signature Domain* (TSP SD). This part of TSPs houses several core functions, and is essential to cell attachment as to the the binding of the *Cluster of Differentiation 47* (CD-47), also known as *Integrin Associated Protein* (IAP) in literature. The SD can bind around 30 Ca^{2+} ions per TSP monomer. Several of these Ca^{2+} ions have been found to be removable, and exchangeable on this protein. It was further found that depending on the concentration of Ca^{2+} the SD's conformation, and its functions are significantly altered. Nevertheless, these important processes are still not well understood today.

Hence the work presented herein has the goal to enlighten the inner workings of the TSP SD. In order to gain significant insights from this domain, several different methods had to be developed, different approaches had to be learned, and new ways of communicating these insights have been tried. The work, and thus this document, are divided into four different stages, which try to investigate the problem of understanding the TSP SD from different angles and shall point out the pluridisciplinity of this work. The four stages are:

1. The Biology of the Thrombospondins:
   This part treats essentially the biological aspects, and poses the questions to be resolved by this work. A significant ef-

fort was thus made to summarize the biology of TSPs as it is known today. This includes the description of the entire protein family, which has a large array of different functions. This section treats the genetics, the structures and the implications of TSP on diseases and state of art treatments of these.

2. Classical Molecular Dynamics Simulation:
   This part highlights the problem from a classical mechanical viewpoint treating the problem at an atomistic level. In order to understand the dynamics of the SD, and to be able to gain insights into the complex process of calcium depletion, several *Molecular Dynamics* (MD) simulations have been performed. This part deals with the elementary knowledge that is needed in order to make such simulations happen. It introduces one to MD *Force Fields* (FF), and their integration. It continues by explaining how statistical ensembles are generated during such simulations and shows how we analyzed and interpreted the trajectories generated by such simulations. Finally, we represent the results obtained by these simulations, such as the identification of exchangeable calcium ions within the TSP SD.
3. Molecular Quantum Mechanics:
   In this case we treat the problem from the viewpoint of modern physics descending to a subatomic level and making the properties of the electronic shell around parts of TSPs the core of the research. In this section, we describe how one obtains a solution of the Schrödinger equation for molecules using the *Self Consistent Field* also known as *Hartree Fock* (HF) method. Then we show how one intelligently joins this method with *Density Functional Theory* (DFT) in order form so called *Hybrid Functionals* such as the very popular *B3LYP* (Becke, three-parameter, Lee-Yang-Parr) in order to achieve decent results on bio-molecules. Once the electron density has been obtained we derived partial charges, which should in the end allow us to verify classical molecular dynamics force fields and to re-inject the results from such calculations into classical molecular dynamics force fields.

   In order to calculate these partial charges we have developed a new method derived from the *electrostatic potential* ESP method, that we named *multicube*. The program

uses contemporary methods such as massive parallel programming which allows to run *multicube* on the *Graphics Processing Unit* (GPU) of a modern computer. Such a sophisticated approach is necessary as classical molecular dynamics lacks the possibility to describe charge transfer, and polarization on the molecular surface, two effects essential in order to treat the binding of Ca^{2+} ions in a correct way.

4. Molecular Visualization:
   This part of the work deals with the visualization of large bio-molecules, and their quantum molecular calculations. Visualization is in itself important as it allows us to better understand and communicate our results to our coworkers, and thus to progress more rapidly in our work. This section further deals with the development of a process in order to visualize the results from quantum chemistry efficiently using the open source *Blender* software. Apart from this we detail the creation of the movies made around the research on the TSP SD outlined herein that were presented on international biological and artistic congresses.

At the end, we highlight the key results of this work and propose future works to further gain insights into the TSP SD.

3 | The Biology of Thrombospondins

3.1 Introduction

Thrombospondins (TSPs) form a family of extracellular matrix proteins. These large molecules are implicated in a variety of biological functions. In humans, five different types of TSPs do exist and various other forms have been found in animals such as in insects [2]. The five different TSPs found in humans can be classified into trimeric and pentameric forms. TSP-1 and 2 are forming trimeric molecules while TSP 3,4 and 5 are forming pentameric molecules. TSP 5 is further known as *Cartilage Oligomeric Matrix Protein* (COMP). These proteins exhibit a large size and contain a multitude of domains. The amino acid sequence length for a single monomer is between 1154 amino acids, in the case of TSP-2 and 737 amino acids in the case of TSP-5 long. All of the TSPs feature a highly conserved domain at their C-terminus, which is called the *Signature Domain* (SD). TSP-1, which was the first to be discovered in 1971 [1], is the most well studied protein of the TSP family. It bears its name from its discovery as it was found to be released from platelets or *thrombocytes* that were stimulated with thrombin. Since then TSPs have been found to be expressed by a variety of cells including endothelial cells, fibroblasts, adipocytes, smooth muscle cells, monocytes and macrophages. Below some physiological functions are presented in a non exhaustive manner. We point those out that we found to be interesting in order to show how critical TSPs are to a biological organism. In the rest of this chapter, we will then show how these critical functions are mapped to the proteins structure introducing the molecular biology of this protein.

Inflammation [3] [4]

Inflammation is the response of living tissue to local injury. TSP-1 is in general found to be up regulated in inflammatory processes. Several receptors of TSP-1 are necessary in order to regulate inflammation. TSP-1 is interacting with several other protagonists in order to activate different cellular pathways that control inflammation. Hence TSP-1 was found to have anti- and pro- inflammatory effects. These have been observed *in vivo* in different animal models. TSP-1 deficient mice for example show acute pneumonia and acute colitis besides several other anomalies in respect to inflammation.

Proliferation [5] [6]

Thrombospondins play a crucial role in proliferation through the interaction with other molecules. Both TSP-1 and TSP-2 were found to promote the proliferation of certain cell types. Ichii et al [5] were for example able to show *in vitro* that TSP-1 is involved in the platelet induced proliferation of smooth muscle cells. N. Lopez et al [6] further found that TSP-2 is for instance a regulator for endothelial cell proliferation.

Wound Healing [7] [8]

Wound healing is a complex process that involves different overlapping stages namely inflammation, proliferation and tissue remodelling. As already pointed out, TSPs do play important roles in the first two stages. Further TSP-1 and TSP-2 have been found to be implicated in the orchestration of the entire process of wound healing. It was shown that TSP-1 is present in early stages of injury, and plays as pointed out an important part in during inflammation. TSP-2 by contrast is up regulated in a later phases of wound healing and hence both proteins are found to participate in wound repair. Lack of TSP-2 in mice has in this case for example shown that their wounds heal slower, that they have a higher blood vessel concentration and feature a less organized extracellular matrix.

Angiogenesis [9] [10]

Angiogenesis describes the process of blood vessel growth and is regulated by several proteins acting either as angiogene-

sis inhibitors or promoters. Both TSP-1 and TSP-2 play an active role in angiogenesis. TSP-1 was for instance found to induce apoptosis, inhibit cell migration and proliferation in endothelial cells. Controlling angiogenetic activity is a major challenge in fighting cancer and therapeutic interventions to either up regulate or down regulate TSPs, as well as TSP mimic peptides such as ABT-510 [11], in order to control angiogenesis have been proposed.

Apoptosis [12] [13] [14]

Apoptosis describes the programmed cellular death. The induction of apoptosis plays an important role for instance in tissue renewal or in cancer treatment where drugs that are used in chemotherapy are designed to induce apoptosis. TSP was found to be either an apoptosis inhibitor or promoter. In the case of thyroid carcinoma cells, TSP-1 was found to inhibit apoptosis induced by camptothecin and doxorubicin [12]. The reverse effect has been found for colon carcinoma cells where TSPs in concert with other proteins can effectively induce apoptosis in such cells [13]. TSP-1 was further found to induce apoptosis in endothelial cells [14].

Chondrogenesis [15]

TSPs and in particular TSP-5 were found to be important in bone growth. Several mutation sites in TSP-5 are known to cause the hereditary disease *Pseudoachondroplasia* (PSACH) and its milder form *Multiple epiphyseal dysplasia* (MED). PSACH is colloquially better known under its name *dwarfism*. Both PSACH and MED cause bone malformations due to irregular cartilage formation. TSP-5 is found to orchestrate in connection with other proteins the growth cartilage. Mutant TSP-5 proteins have been found to fail to do so, causing these diseases.

Central Nervous System Synapsogenesis [16] [17]

TSP-1 and TSP-2 were found to be secreted by astrocytes. TSP-1,2 was further found to be dynamically regulated during brain development [16]. Concentration is found to be low in the late embryonic brain, then higher in the postnatal brain, and almost absent in the adult brain but are nevertheless expressed

by reactive astrocytes and microglia. Karen S. Christopherson et al [17] were able to show that synapses formation in astrocyte conditioned medium and in rat retinal ganglion cells can be induced by TSP-1. They were further able to show that neither TSP-1 or TSP-2 deficient mice show a decreased number of synapses in their brain. These mice however show a 40% decrease in synaptic puncta during certain stages of development in a TSP-1,2 double null cerebral cortex.

Obesity [18]

A study by Y. Li et al using mice has shown that TSP-1 plays a significant role in obesity. Transgenic TSP-1 knockout mice fed with a high fat diet are found to develop obesity just as fast as wild type mice. Hence TSP-1 is not found to be implicated in the process of obesity development itself. TSP-1 knockout mice however show significantly improved glucose tolerance and insulin sensitivity. It was further shown that such TSP-1 knockout mice had decreased macrophage accumulation in adipose tissue. Y. Li et al. are proposing that TSP-1 is responsible for the accumulation of macrophages in adipose tissue, and that together with other effects, provokes obesity induced inflammation which causes insulin resistance.

Tumorigenesis [40]

TSP was found to have numerous complex roles in the formation of tumor cells and in metastasis. Depending on tissue and cell type the protein was found either to be a tumor promoter or repressor. For instance, TSP is found to be an inhibitor for tumor angiogenesis. Again as in the previous cases, TSP orchestrates these functions in combination with a multitude of other extracellular matrix proteins or cell surface proteins where it attaches itself and induces signal transduction. Functions relating tumorgeneses, especially tumor angiogenesis repression are important as future cancer treatement drugs are modeled around peptide sequences from TSP-1. ABT-510 [19] [20], ABT-526 [21] [22] and ABT-898 [23] are mimic peptides built from sequences found in TSP-1. These molecules are currently at various stages of clinical trials as cancer treatment drugs.

3.2 Structural and Functional Overview

TSPs are composed of several different domains housing different functions. Human TSPs are divided into two groups, group A containing the trimeric TSPs: TSP-1 and 2, while group B contains the pentameric TSPs: TSP-3,4 and 5. Group A proteins are build up of a globular N-terminal domain, a coiled-coil oligomerization domain, a *von Willebrand Factor, Type C* (VWC) module, three *Thrombospondin Structural Repeats* (TSR), three *Epidermal Growth Factor* (EGF) like repeats, a wire like *Stalk* made up of calcium binding type N and type C modules in the sequence 1C-NCCNCCNCNCN-13C and a C-terminal *Globe* globular domain. In literature the three EGF-like repeats are also frequently connoted as three TSR2 repeats. Further is the *Stalk* as denoted as seven TSR3 repeats. The EGF-like repeats together with the *Stalk* and the *Globe* form the *Signature Domain* (SD) which is common to all TSPs and features high sequence identity between them. TSP-3,4 and 5 are structured differently. TSP-3 and 4 retain a globular N-terminal domain, while TSP-5 does not. TSR repeats as well as the vWC module are absent in group B members but they contain an additional fourth EGF-like repeat. Further to group A and B TSPs other TSPs were found in different species. Insect TSPs for examples are predicted to have more EGF-like repeats [2] [41]. Nevertheless all of them contain the TSP-SD which features high sequence identity. An overview of the domain composition of the five TSPs available in humans is shown in figure 3.1.

We now go further into detail and are describing every domain and its functions in the whole thrombospondin complex. Various crystal structures of TSP domains are today available and provide an insight of TSPs at the atomistic level. We shall begin at the N-terminal and work ourselves through to the TSP-SD which we will treat with a special regard as this domain is the one this work focuses on.

Globular N-terminal Domain

The N-terminal domain is one of the less conserved domains between thrombospondins, and is completely absent in TSP-5. It is famous for its ability to bind to various glycosaminoglycans in particular to heparin. The binding site of TSP to the *Low Den-*

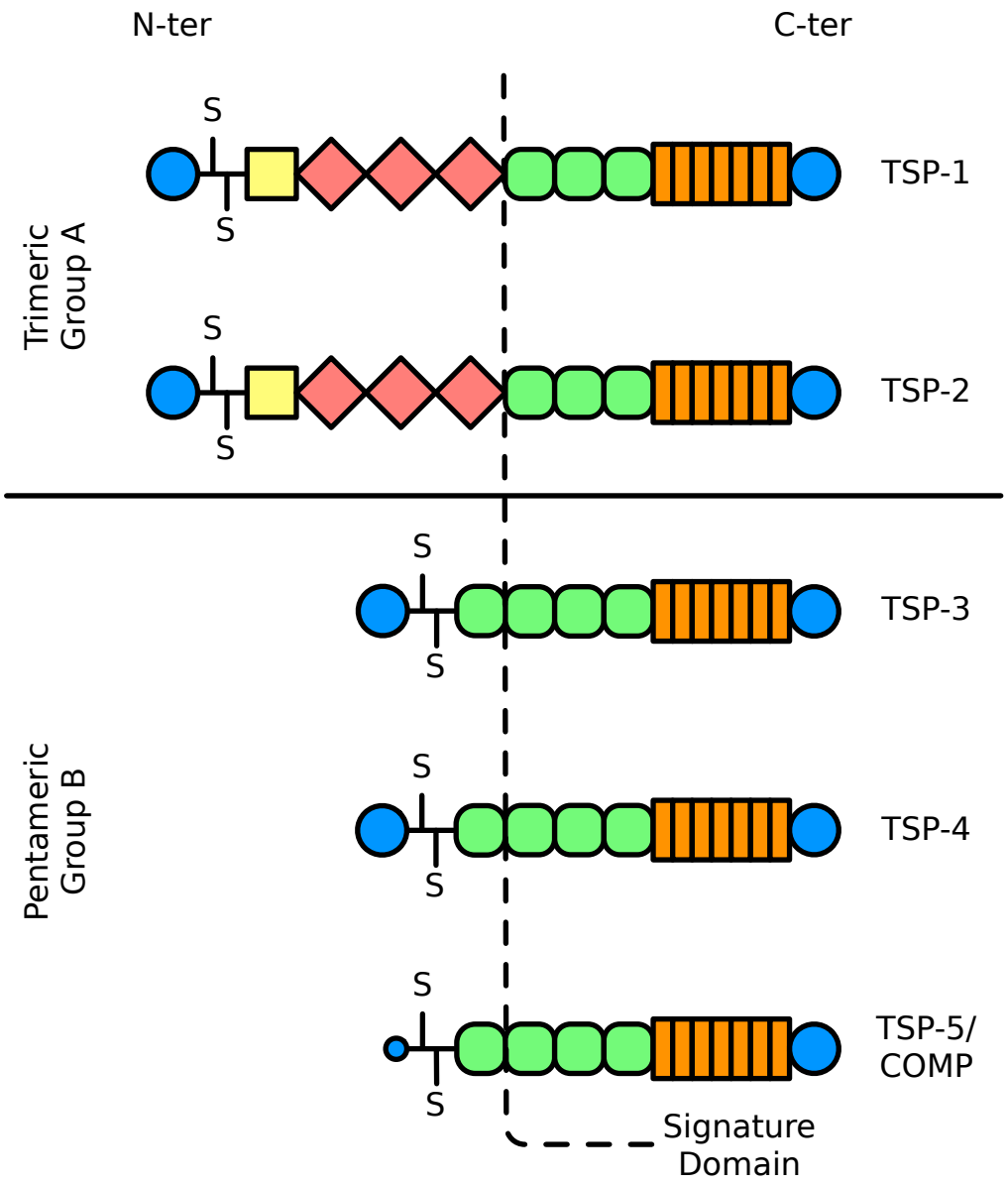

Figure 3.1: The TSP family: Globular N- and C-termini are shown in blue, the coiled-coil domain with disulphide bonds linking the monomers together is represented as a black line, the VWC module is shown in yellow, TSR repeats in red, EGF-like repeats in green and the *Stalk* in orange.

sity Lipoprotein Receptor-related Protein (LRP) is also found to be within the first 90 residues of the N-terminal domain. Both binding sites have been found to be important in that they regulate the cellular internalization of TSP-1 and 2. This process is involved for example in wound healing as heparin is released by injured tissue. Further the cellular internalization of TSP plays a role in *Matrix Metalloproteinase* (MMP) activity as these can bind to other domains of TSP but are internalized together with TSP. An another important binding site is that to cell surface calrecticulin within amino acids TSP-1 E35-D53. TSP-1 was found to influence, in interacting with calrecticulin, focal adhesion, cell migration and anoikis, a form of programmed cell death induced by the detachment of cells from extracellular matrix [42]. Several interactions with integrins have also been mapped to the N-terminal domain of TSP-1 and TSP-2. TSP-1 and TSP-2 was found to interact with integrins α4β1 and α6β1 [43] [44]. α4β1 is proposed to interact with residues TSP-1 A177ELDVP and α6β1 with residues TSP-1 L105-G114 [43]. α3β1 may only interact with TSP-1 as its binding site TSP-1 F208-F219 is not conserved in TSP-2 [45]. Integrin interactions at the N-terminal are mediating one further mechanism related to angiogenic activity and to modulate T cell behavior [46]. Interaction sites with fibrinogen were also reported to be found on the N-terminal domain of Thrombospondin. Those interactions have been mapped to residues TSP-1 I169-P182 which overlaps with the binding site to α4β1 integrin. [47] The structure of the N-terminal domain of TSP-1 has been resolved several times by the means of x-ray crystallography by K. Tan et al. [48] [49]. Those structures are deposited in the *Protein Data Bank* (PDB) [50] using the codes 1Z78, 1ZA4, 2ERF, 2ES3, 2OUH and 2OUJ. The N-terminal domain consists of β-sandwich structure, consisting of 13 anti-parallel β-strands and one irregular strand-like segment. Further the structure features six α-helices which are situated between the β-strands and at the C-terminal of the domain. Residues TSP-1 R47, K50, K60, R95, K98 and R189 were found to form a highly positively charged patch where glycosaminoglycans can bind. K. Tan et al. further found that two N-terminal domains can bind glycosaminoglycans cooperatively in forming a dimer, noting that dimerization of the N-terminal alone without glycosaminoglycans was not found to be possible [49]. The features of this domain are further highlighted in figure 3.2.

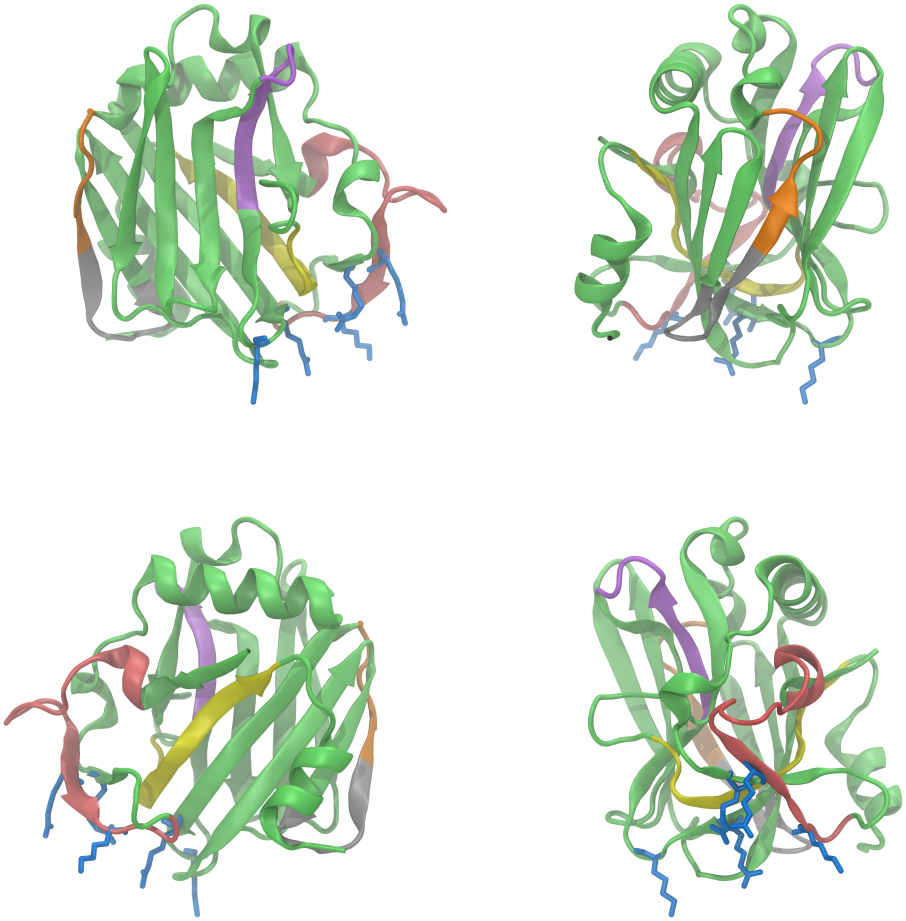

Figure 3.2: The TSP-1 N-terminal domain: This image shows the structure deposited in the PDB under the code 1ZA4, in four different rotations of 90° around the vertical axes. Highlighted are residues forming a highly positive patch where glycosaminoglycans can bind in licorice representation in blue, sequences proposed to be binding to integrin α4β1 in orange, integrin α6β1 in violet, integrin α3β1 in yellow, calrecticulin in red and fibrinogen in gray and orange.

Coiled-coil Oligomerization Domain

The structure of the coiled-coil oligomerization domain is very different depending whether the TSP family member is trimeric or pentameric. In both cases nevertheless one is presented with a coiled-coil structure rigidified by knobs in holes packing, and further stabilized by disulphide bonds. These disulphide bonds are close to the N-terminal side of this structure in trimeric TSPs, while they are close to the C-terminal side in pentameric TSPs. Each α-helix belongs to a different monomer of the TSP protein, and this rigid domain effectively holds the trimer or pentamer together. However, while the α-helices of trimeric TSPs are proposed to be tightly packed, the pentameric TSPs are probably featuring a hole and the coiled-coil structure is more tube-like than string-like as in trimeric TSPs. Insights into the trimeric oligomerization domain can be obtained from a structural homologue, the chicken cartilage matrix protein, also known as matrilin-1 [51]. Each α-helical coil contains two cysteins. It was shown that these cysteins are not necessary for the formation of the trimeric structure of TSP-1,2 but are stabilizing the trimer once formed. The same is true for matrilin-1. In contrast to the trimeric oligomerization domain, structures from X-ray crystallography have been resolved and are accessible in the PDB by the codes 1VDF [52] and 1FBM [53], for mouse TSP-5, and 1MZ9, for Norwegian rat TSP-5 [54]. The structures from mouse TSP-5 hold a chlorine ion inside them. Further the pentameric oligomerization domain reveals the capability to bind hydrophobic compounds within a pore that adapts to such ligands. The structure from Norwegian rat TSP-5 is resolved as a complex with vitamin D3. molecules, and the authors suggest that TSP-5 has storage and delivery functions for molecules such as vitamin D3 and all trans-retinol. A visualization of the pentameric coiled-coil oligomerization domain with such ligands is shown in figure 3.3. Even though two ligands are exposed in the structure the authors speculate that only the binding site closer to the N-terminal is valid if the protein is entirely assembled. Vitamin D3 might by metabolism be transformed into a steroid hormone that promotes skeletal development. [54]

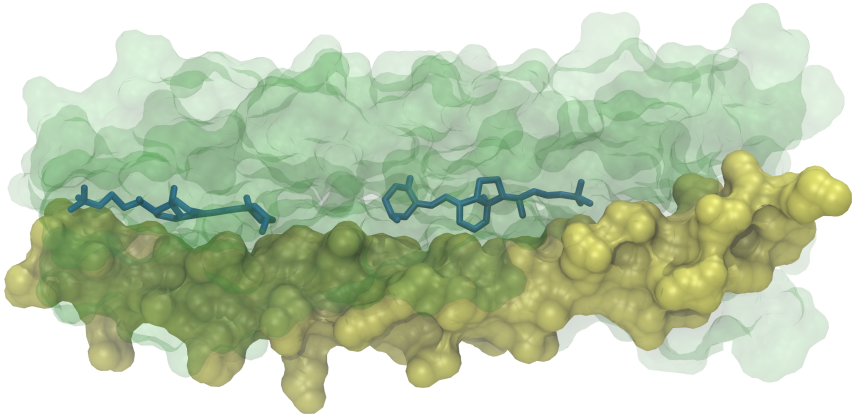

Figure 3.3: The TSP-5 coiled-coil oligomerization domain: Shown here is one coiled strand from one monomer in yellow, while the other four strands are transparent and green depicting the knobs in holes structure. In the interior of the coiled-coil complex one can find the two vitamin D3 molecules shown in blue. The figure was made using the structure referenced by 1MZ9 in the PDB. The N-terminal is shown on the right of this figure and the authors of this crystallographic structure speculate that *in vivo* only the rightmost binding site might be valid [54].

Von Willebrand Factor, Type C Module

TSP-1 and 2 do contain a VWC module linked to the C-terminal of oligomerization domain. The VWC module is also known by the names chordin-like cysteine-rich repeat or procollagen module. This domain is reported to be involved in anti-angiogenic activity, and has a *Transforming Growth Factor β* (TGF-β) binding site. This site has been mapped to N321GVQYRN, and fluorescence analysis indicate that this site is solvent accessible [55]. This module contains ten cysteine residues and all of them are involved in disulphide bonds. C. B. Carlson et al. [51] compared this module with the VWC module from procollagen-IIA for which a *Nuclear Magnetic Resonance* (NMR) structure is deposited in the PDB under the code 1U5M. They stated that the structure must be determined by the conserved cysteine residues which in this homologue form a 1-4, 2-8, 3-5, 6-9 and 7-10 pairing scheme and further proposed that this module plays a hinge like role due to its flexibility.

Thrombospondin Structural Repeats

Again unique to TSP-1 and 2 in the TSP family is that both of them contain three TSR repeats. TSR repeats have been found to play a major role in angiogenesis inhibition and derived disease treatments such as cancer. Several peptides that mimic the sequence G454VITRIR from the second TSR repeat, designated ABT-510 [19] [20], ABT-526 [21] [22] and ABT-898 [23], have been developed and are in various stages of clinical trials as cancer treatment drugs. Further 3TSR a recombinant protein containing all three TSR repeats and their effect on diseases such as inflammatory bowel disease have been studied [56]. TSR repeats have been found to bind glycosaminoglycans [57], to MMP2 and MMP9 [58], to pan-β1 integrins [59], to repress angiogenesis in binding to the *Cluster of Differentiation* (CD-36) which modulates the *Vascular Endothelial Growth Factor* (VEGF) [60] and to activate TGF-β. The interaction with CD-36 is targeted by the mimic peptides but has further also been attributed to the C447SVTCG [61] ad-joint to G454VITRIR in the second repeat and also available in the third repeat C504SVTCG. TGF-β activation has been mapped to the TSP-1 R431FK sequence which lies between the first and the second TSR repeat of TSP-1 and is not conserved on TSP-2. This activation plays an important

role in cancer research as active TGF-β can induce apoptosis in TGF-β sensitive tumor cells [13] [62]. It was further found that 3TSR can inhibit angiogenesis induced by *Nitric Oxide* (NO) in a mechanism which is also dependent on CD-36. The TSR structure of the second and third TSR in TSP-1 has been resolved by K. Tan et al [57] using X-ray crystallography and is available in the PDB by its code 1LSL. In this crystal, each of the two TSR repeats have folded into a spiraling, anti-parallel three stranded domain. Of the three strands, only two form to some extent a regular β-structure. The first strand (seen from the N-terminal) features a WXXWXXW motive. Each of these tryptophans bind to an arginin amino acid on the opposed strand, which make this fold rather unique. The structure reveals six cysteins in each repeat that are linked in the order 1-5, 2-6 and 3-4. The authors suggest that the link between the second and third TSR repeat is rather rigid while the link to the first TSR repeat which is four residues longer is probably more flexible. The structure reveals one fucose linked to the threonine in the CSVTCG sites found in the structure. Further C-mannosylation sites were indicated around the WXXW motives. TSP-1 W385, W438, W441 and W498 are for instance found to be ready for C-mannosylation [63]. Figure 3.4 shows the second and third TSR repeat from TSP-1 and points out several features described herein.

The Thrombospondin Signature Domain

Each member protein of the TSP family contains the SD at its C-termini. The SD is composed of three EGF-like repeats, a calcium rich wire like domain called the *Stalk* and a globular domain called the *Globe*.

The *Stalk* can be decomposed into several calcium binding repeats which are either of type N or C. The repeats occur in the following sequence 1C-NCCNCCNCNCN-13C (seen from N to C-terminal). Type N calcium repeats are composed of 13 amino acids with the exception of repeat 8N which is 15 amino acids long. Type C repeats contain 23 amino acids, and repeat 1C and 11C have inserts of up to 13 residues. In the structures of the SD up to 32 Ca^{2+} ions were found and up to 28 of these can be found in the calcium binding repeats of the *Stalk*. Around ten of these Ca^{2+} ions where found to be exchangeable on recombinant proteins built from the TSP-2 SD [64]. Several experiences have

Figure 3.4: TSP TSR repeats: Shown are the second and third TSR repeat of TSP-1. The unique tryptophan- arginine stacking linking the first strand to the second is represented licorice representation and highlighted in blue. The sequence from which mimic peptides ABT-510, ABT-526 and ABT-898 have been inspired is highlighted in red. Sequences found to be important for CD-36 binding are highlighted in violet, fucose sugars in gray and cysteins forming disulphide bonds in yellow.

shown that the structure and the function of TSPs is dependent on calcium concentration [65] [66] [25], hinting the importance of the *Stalk* and the aspects of calcium binding in the SD. Pentameric TSPs in humans contain four EGF-like domains, but by terminology the first EGF like domain is not considered to be part of the SD. TSPs in other species might have even more EGF like repeats [2] [41]. The TSP SD is further the part of TSPs that features the highest sequence and structural identity in structures that have been resolved so far. A *MUSCLE* alignment [67] of human TSP SD is shown in figure 3.7 for the *Stalk* and in figure 3.8 for the *Globe*. Several structures of different parts or of the entire signature domain have been resolved:

- A small structure of the TSP-1 double mutant C992S/N1067K which resolves the SD from *Stalk* repeat 7C to the C-terminal. PDB code: 1UX6 [68]
- A structure of the entire TSP-2 SD. PDB code: 1YO8 [69]
- A structure of the entire TSP-2 SD featuring the N702S mutation. This structure was resolved with the goal to gain insights into coronary heart disease as the equivalent mutation in TSP-1 is the cause for this hereditary disease. PDB code: 2RHP [70]
- A partial structure of the TSP-5 SD containing the C-terminal most EGF-like repeat, the entire *Stalk* and the *Globe* PDB code: 3FBY [71]

Images of the complete SD structure made from the information deposited in the PDB under the code 1YO8 is shown in figure 3.5.

The calcium replete structures show the EGF-like repeats linked to the *Stalk*, and interacting with repeat 12N and 13C from the *Stalk*. In the structures from TSP-2 a calcium ion has been found on the interface of the first two EGF-modules. Repeat 1C, 8N and 9C of the *Stalk* are interacting with the *Globe*, while repeats 2N to 7C are not directly linked to the *Globe* and form a 14Å by 40Å hole. The interactions between the *Globe* and the *Stalk* have been found to be different in the crystal structures [71]. The *Stalk* from TSP-5 was found not to interact with the surface of the globe in using repeat 8N. Subtile differences do exist in calcium placement between the different crystal structures. All structures, besides the TSP-1 C992S/N1067K double mutant structure, feature three ions bound to the *Globe*. The TSP-1 C992S/N1067K double mutant structure features four. Within

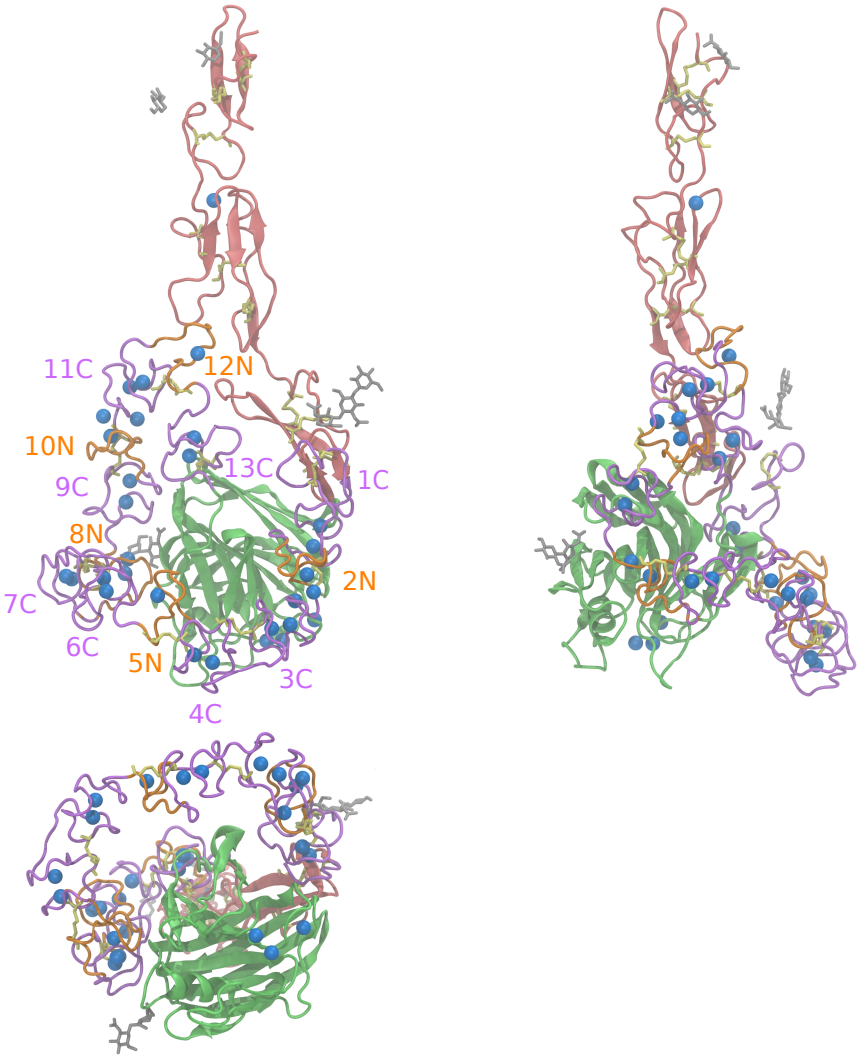

Figure 3.5: The Signature Domain: This visualization of the entire TSP-2 SD from has been made using the structural information found in the PDB [50] by the code 1YO8. Images are rotated by 90° around the vertical axes. The image in the second line is rotated 90° around the horizontal axes and the hole in the structure between the *Stalk* and the *Globe* is clearly visible. EGF-like repeats are colored in red, N type *Stalk* repeats in orange, C type *Stalk* repeats in violet, the *Globe* in green, cysteins in yellow, calcium ions in blue and glycans in gray.

the *Stalk* one finds 12 ions in the shortened TSP-1 double mutant structure (PDB code 1UX6) while the TSP-2 and TSP-5 structures (PDB code 1YO8 and 3FBY) reveal 26 ions in the *Stalk*. Although both TSP-2 and 5 have the same number of ions bound to the *Stalk* their placement differs. Besides the *Stalk* and *Globe* the TSP-2 structure features one ion situated between the first and second EGF-like repeat. In detail ion placement is highlighted in figure 3.6.

The *Globe* folds in solved TSP SD structures into a β-sandwich composed of 15 strands, and was found to be part of a larger family of lectin like folds [68]. In TSP-1,3,4 and 5 the *Globe* contains an unpaired cysteine. In the TSP-5 structure this cysteine is deeply buried within the *Globe* and should thus probably be inert in pentameric TSPs. The location is TSP-1 C992 for TSP-1 and not resolved in any structure, while for pentameric TSPs the location is given by TSP-5 C726 and homologues.

No explicit structural information exists of a TSP SD that is not repleted with Ca^{2+} ions. Structural information is only hinted by probing the Ca^{2+} depleted structure using monoclonal antibodies [32]. D. S. Annis et al. the authors of this studies suggested two models. In the first model, the interactions between 8N, 9C and the *Globe* are not maintained and residues between the cysteins in 12N and 13C are fully extended, while the structure of the other *Stalk* repeats should be maintained. In the second model, proposed interactions between 8N, 9C and the *Globe* are maintained resulting a shorter signature domain, but in coherence with the sizes measured for Ca^{2+} depleted structures [66] if all residues between 12N and 13C are fully extended. This model should allow for rapid change between a calcium replete, deplete form.

Hereditary diseases PSACH, MED and coronary heart disease are mapped to residue mutations in this domain. PSACH and MED severely alter the bone phenotype as PSACH may better be known under the name dwarfism. Residues found to be changed in PSACH and MED appear to be part of the calcium binding stalk of TSP-5 and are thought to modify the proteins ability to bind these ions which should affect the ability of TSP-5 to interact with matrilin-1,3 and 4 [72], type I,II [73] and type IX collagen [74]. All these interactions have been found to be cation dependent [72] [15]. TSP-5 was further found to engage type XII and XIV collagen in the skin extracellular matrix using the SD [75].

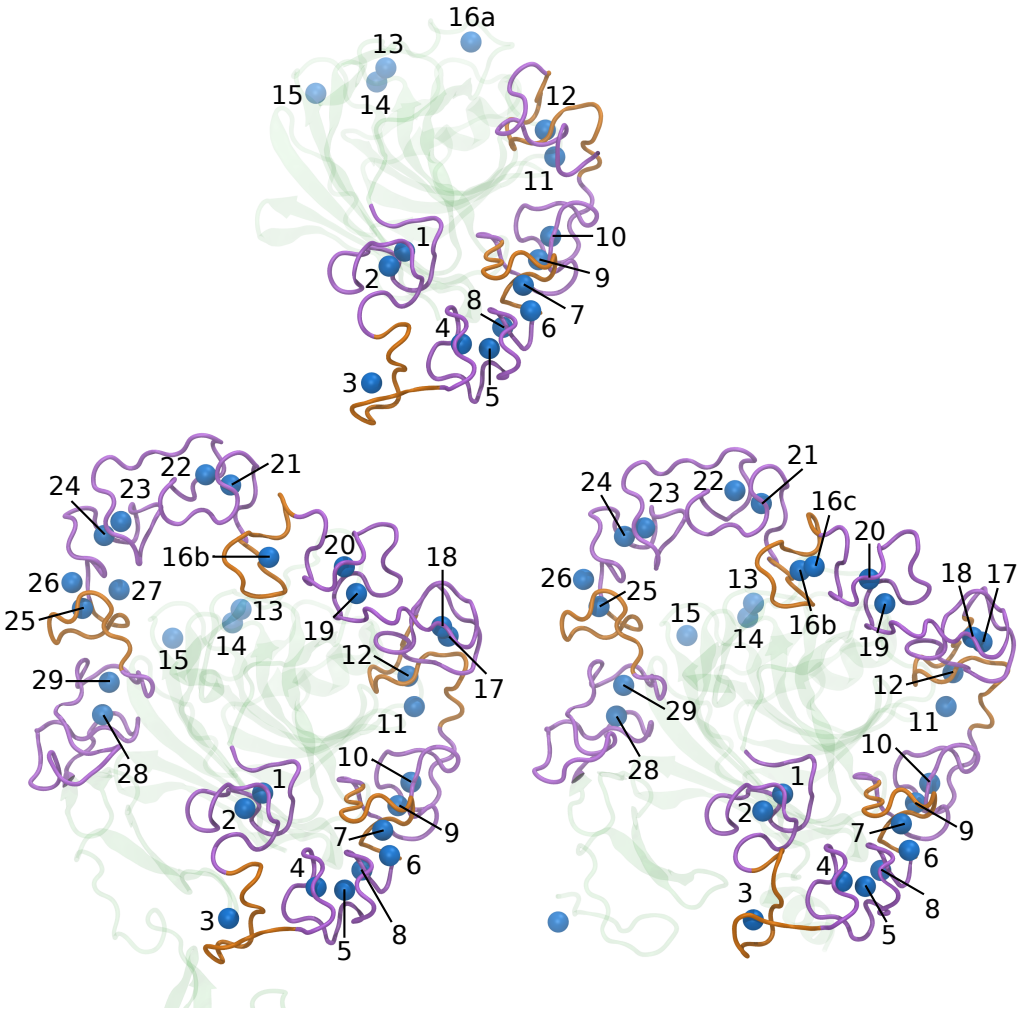

Figure 3.6: The *Stalk* of the SD: Ions are highlighted in blue, C type repeats in violet, N type repeats in orange, the rest of the SD structure available in transparent green. The TSP-1 SD structure from PDB code 1UX6 is shown on top, the TSP-2 structure from PDB code 1YO8 on the bottom left and the TSP-5 structure from PDB 3FBY on the bottom right.

Coronary heart disease has been linked to the mutation TSP-1 N700S [76] to be found on repeat 1C of the *Stalk*. NO signaling is also regulated by this domain as it interacts with CD-47 in the case of TSP-1. NO was found to regulate the blood flow, to be a limiting factor in platelet activation and preserves vascular health in case of inflammation. NO signaling is also related to a radiosensitivty in tissues [27]. Transgenic TSP-1 or CD-47 knockout mice have found to feature an impressing resistance to irradiation [77]. The importance of the signaling pathway might further be pointed out by the fact that NO signaling affects angiogenesis in tumor and would healing environments [27]. TSP-1 also was found to inhibit T cell activity using the CD-47 receptor [46]. Further the inhibition of thyroid carcinoma cells apoptosis was linked to the interaction between TSP-1 and CD-47 [12]. The interaction of TSP with CD-47 was linked to the sequences TSP-1 R1034FYVVMWK and I1121RVVM in peptide studies [24]. Both seem to be buried in the structures resolved of the SD [68] and are to be found within the *Globe*. While the RFYVVMWK motive seems to be conserved across the TSP family the IRVVM is not. It was found that CD-47 binds to TSP-1 selectively in studies involving TSP-1,2 and 4 [78]. The signature domain further features the integrin binding motive RGD in the *Stalk* of TSP-1,2 and 5. However while the location of this motive is conserved between TSP-1 and 2 (TSP-1 R926GD, TSP-2 R928GD) and found on *Stalk* repeat 12N, the site is found at a different location in TSP-5 (R367GD) where it is part of repeat 5N. Another difference is that the TSP-1,2 RGD site has found to be buried by the authors of available crystallographic structures [68] [69] while the TSP-5 RGD binding site was found to be at least partially available [71]. The SD has been found to bind to αvβ3 integrin using this site in a calcium dependent manner [25] [26].

As shown here the calcium deplete structure might have an important role. We further outline that the structures obtained for the SD have all been obtained by crystallization in a calcium rich medium, and major binding sites such as the integrin [70] and CD-47 binding [68] sites are found to be buried in these structures. We have thus tried to use the possibilities of computational simulation to gain further insights into the SD, in focusing on the effects and the actions caused either by calcium depletion or repletion. These are outlined in the following chapters, dealing with classical molecular dynamics and molecular

quantum mechanics.

```
          Stalk
          1C                                  2N

TSP-1 691 EDTDLDGWPNENLVCVANATYHCKKDNCPNLPNSGQ EDYDKDGIGDACD
TSP-2 693 EDSDLDGWPNLNLVCATNATYHCIKDNCPHLPNSGQ EDFDKDGIGDACD
TSP-3 457 TDTDIDGYPDQALPCMDNNK-HCKQDNCLLTPNSGQ EDADNDGVGDQCD
TSP-4 463 KDVDIDSYPDEELPCSAR---NCKKDNCKYVPNSGQ EDADRDGIGDACD
TSP-5 268 RDTDLDGFPDEKLRCPER---QCRKDNCVTVPNSGQ EDVDRDGIGDACD


          3C                     4C                     5N

TSP-1 740 DDDDNDKIPDDRDNCPFHYNPAQ YDYDRDDVGDRCDNCPYNHNPDQ ADTDNNGEGDACA
TSP-2 742 DDDDNDGVTDEKDNCQLLFNPRQ ADYDKDEVGDRCDNCPYVHNPAQ IDTDNNGEGDACS
TSP-3 505 DDADGDGIKNVEDNCRLFPNKDQ QNSDTDSFGDACDNCPNVPNNDQ KDTDGNGEGDACD
TSP-4 509 EDADGDGILNEQDNCVLIHNVDQ RNSDKDIFGDACDNCLSVLNNDQ KDTDGDGRGDACD
TSP-5 314 PDADGDGVPNEKDNCPLVRNPDQ RNTDEDKWGDACDNCRSQKNDDQ KDTDQDGRGDACD


          6C                     7C                     8N

TSP-1 799 ADIDGDGILNERDNCQYVYNVDQ RDTDMDGVGDQCDNCPLEHNPDQ LDSDSDRIGDTCDNN
TSP-2 801 VDIDGDDVFNERDNCPYVYNTDQ RDTDGDGVGDHCDNCPLVHNPDQ TDVDNDLVGDQCDNN
TSP-3 564 NDVDGDGIPNGLDNCPKVPNPLQ TDRDEDGVGDACDSCPEMSNPTQ TDADSDLVGDVCDTN
TSP-4 568 DDMDGDGIKNILDNCPKFPNRDQ RDKDGDGVGDACDSCPDVSNPNQ SDVDNDLVGDSCDTN
TSP-5 373 DDIDGDRIRNQADNCPRVPNSDQ KDSDGDGIGDACDNCPQKSNPDQ ADVDHDFVGDACDSD


          9C                     10N           11C

TSP-1 860 QDIDEDGHQNNLDNCPYVPNANQ ADHDKDGKGDACD HDDDNDGIPD----DKDNCRLVPNPDQ
TSP-2 862 EDIDDDGHQNNQDNCPYISNANQ ADHDRDGQGDACD PDDDNDGVPD----DRDNCRLVFNPDQ
TSP-3 625 EDSDGDGHQDTKDNCPQLPNSSQ LDSDNDGLGDECD GDDDNDGIPDYVPPGPDNCRLVPNPNQ
TSP-4 629 QDSDGDGHQDSTDNCPTVINSAQ LDTDKDGIGDECD DDDDNDGIPDLVPPGPDNCRLVPNPAQ
TSP-5 434 QDQDGDGHQDSRDNCPTVPNSAQ EDSDHDGQGDACD DDDDNDGVPD----SRDNCRLVPNPGQ


          12 N          13C

TSP-1 919 KDSDGDGRGDACK DDFDHDSVPDIDDICPENVDISE
TSP-2 921 EDLDGDGRGDICK DDFDNDNIPDIDDVCPENNAISE
TSP-3 688 KDSDGNGVGDVCE DDFDNDAVVDPLDVCPESAEVTL
TSP-4 692 EDSNSDGVGDICE SDFDQDQVIDRIDVCPENAEVTL
TSP-5 493 EDADRDGVGDVCQ DDFDADKVVDKIDVCPENAEVTL
```

Figure 3.7: Sequence alignment of the *Stalk* from human TSPs: Integrin binding sites are highlighted in blue, residues interacting with the globe are highlighted in red, residues interacting with EGF-like domains are highlighted in orange and the site of the N700S mutation causing conary heart desease is highlighted in gray. Insights into interactions highlighted here are found in section 4.8.

```
           Globe

TSP-1  955 TDFRRFQMIPLDPKGTSQNDPNWVVRHQGKELVQTVNCDPGLAVGYDEFNAVDFSGTFFI
TSP-2  957 TDFRNFQMVPLDPKGTTQIDPNWVIRHQGKELVQTANSDPGIAVGFDEFGSVDFSGTFYV
TSP-3  724 TDFRAYQTVVLDPEGDAQIDPNWVVLNQGMEIVQTMNSDPGLAVGYTAFNGVDFEGTFHV
TSP-4  728 TDFRAYQTVVLDPEGDAQIDPNWVVLNQGMEIVQTMNSDPGLAVGYTAFNGVDFEGTFHV
TSP-5  529 TDFRAFQTVVLDPEGDAQIDPNWVVLNQGREIVQTMNSDPGLAVGYTAFNGVDFEGTFHV


TSP-1 1015 NTERDDDYAGFVFGYQSSSRFYVVMWKQVTQSYWDTNPTRAQGYSGLSVKVVNSTTGPGE
TSP-2 1017 NTDRDDDYAGFVFGYQSSSRFYVVMWKQVTQTYWEDQPTRAYGYSGVSLKVVNSTTGTGE
TSP-3  784 NTVTDDDYAGFLFSYQDSGRFYVVMWKQTEQTYWQATPFRAVAQPGLQLKAVTSVSGPGE
TSP-4  788 NTQTDDDYAGFIFGYQDSSSFYVVMWKQTEQTYWQATPFRAVAEPGIQLKAVKSKTGPGE
TSP-5  589 NTVTDDDYAGFIFGYQDSSSFYVVMWKQMEQTYWQANPFRAVAEPGIQLKAVKSSTGPGE


TSP-1 1075 HLRNALWHTGNTPGQVRTLWHDPRHIGWKDFTAYRWRLSHRPKTGFIRVVMYEGKKIMAD
TSP-2 1077 HLRNALWHTGNTPGQVRTLWHDPRNIGWKDYTAYRWHLTHRPKTGYIRVLVHEGKQVMAD
TSP-3  844 HLRNALWHTGHTPDQVRLLWTDPRNVGWRDKTSYRWQLLHRPQVGYIRVKLYEGPQLVAD
TSP-4  848 HLRNSLWHTGDTSDQVRLLWKDSRNVGWKDKVSYRWFLQHRPQVGYIRVRFYEGSELVAD
TSP-5  649 QLRNALWHTGDTESQVRLLWKDPRNVGWKDKKSYRWFLQHRPQVGYIRVRFYEGPELVAD


TSP-1 1135 SGPIYDKTYAGGRLGLFVFSQEMVFFSDLKYECRDP------------------
TSP-2 1137 SGPIYDQTYAGGRLGLFVFSQEMVYFSDLKYECRDI------------------
TSP-3  904 SGVIIDTSMRGGRLGVFCFSQENIIWSNLQYRCNDTVPEDFEPFRRQLLQGRV-
TSP-4  908 SGVTIDTTMRGGRLGVFCFSQENIIWSNLKYRCNDTIPEDFQEFQTQNFDRFDN
TSP-5  709 SNVVLDTTMRGGRLGVFCFSQENIIWANLRYRCNDTIPEDYETHQLRQA-----
```

Figure 3.8: Sequence alignement of the *Globe* from human TSPs: Unconfirmed CD-47 binding sites found in peptide studies are highlighted in blue, regions found to be flexible using molecular dynamics simulations are highlited in green, residues interacting with the 8N stalk repeat are highlighted in red and residues muted in the TSP-1 C992S/N1067K double mutant structure are highlighted in violet. Insights into regions depicted to be flexible and interactions are found in section 4.8.

4 | Classical Molecular Dynamics Simulation

4.1 Introduction

Classical *Molecular Dynamics* (MD) simulations are a theoretical framework which if implemented on computers allow one to investigate physical properties of an ensemble of atoms and molecules. These might be gases, liquids, or biological molecules as in our case. Such simulations date back to the 1950's and were originally developed in the field of theoretical physics [28]. As computational power is rising, the interest in MD simulation is growing and extends today to a variety of fields. Important to us, it is nowadays also used in biology to investigate biological functions. This goes so far that even highly specialized machines have been built in order to perform such calculations [29]. In our case, we use MD simulations in order to understand the dynamics of the *Signature Domain* (SD) which is an integral part of the proteins belonging to the *Thrombospondin* (TSP) family.

As in all simulations, which are by definition, only an approximation of reality, there are major drawbacks in MD simulations. The description of the system is inherently limited by the computational time available, and one has thus to carefully choose a model where the accuracy is not limiting the description of the process to be investigated and where the simulation is still feasible under the constraints of the computational power and time available. This task is probably is one of the most complicated in performing such simulations and the accuracy of such simulations remains highly debated.

Classical MD, as is hinted by its name limits itself to classical physics. It neither takes quantum mechanics nor relativistics into account and is build on the principals of classical mechan-

ics. This type of simulation features today a set of algorithms that are touching the fields of electromagnetism, other modeled atom-atom interactions such as *van der Waals* (VDW) forces and a large set of statistical mechanics. These algorithms are today available through sophisticated software packages such as *GROMACS* [30] or *NAMD* [31]. In here we will describe such theory to the level of the MD simulations that we have performed.

After describing the theoretical framework used by the software we used to perform our simulations, we will show how such simulations are done in practice in simulating the SD of TSPs. We further will highlight the problems we faced, in choosing the right parameters, such as force fields and statistical ensembles in order to simulate this large complex protein domain.

Once such MD simulations have been performed, one still has to put a great effort into the analyses of the data and trajectories obtained. As our interest focuses on the SD of TSPs and on effects such as calcium depletion, we developed some modules and methods, as for instance a module to investigate how the coordination sphere of a Ca^{2+} ion is altered during a simulation. These developments will be presented in detail later in this chapter.

Finally, we outline the results that we obtained from such classical simulations, and thus the insights that we gained into the TSP SD domain such as the identification of the positions of *exchangeable* ions, and the accessibility to binding sites. We then discuss the biological implications of these results obtained.

4.2 Force Fields

A force field is the essence of every MD simulation. It defines the potential and thus the interactions between the objects that are in so called all atom force fields, the atoms, or in coarse grained simulations between the so- called beads. The interactions are modeled to reasonably recreate empirically observed molecular properties. This does not mean that they are necessarily physical. A typical potential defining a force field has several terms to model these interactions. In general, we talk about N-body models with configurational interactions. Hence, a potential of the form:

$$V(q) = V(q_1, q_2, \ldots, q_N), \tag{4.1}$$

where $q_i \in \mathbb{R}^3$ is the position of particle i in an ensemble of N particles. This potential is then in general in classical MD composed of bonded and non-bonded components. The bonded components which in most cases hold the molecule together are modeled by harmonic potentials around a minimized optimal distance r^0 or optimal angle φ^0 and dihedral ϑ^0. The non-bonded components in such force fields are in general approached by a Lennard Jones interaction which models the VDW interaction and a Coulomb potential which models the interactions due to the charge of the particles. Let us develop this:

$$V = V^{\text{bond}} + V^{\text{nonbond}}, \tag{4.2}$$

$$V = \sum_{\text{Bonds}} V^{\text{lb}} + \sum_{\text{Angles}} V^{\text{ab}} + \sum_{\text{Dihedrals}} V^{\text{dh}} + \sum_{i \neq j} V^{\text{VDW}} + \sum_{i \neq j} V^{\text{C}}, \tag{4.3}$$

where V^{lb} is the harmonic linear bond potential which takes the form:

$$V_{ij}^{\text{lb}}(q_i, q_j) = k_{ij}^{\text{lb}}(r_{ij} - r_{ij}^0)^2, \tag{4.4}$$

where $r_{ij} = |q_i - q_j|$. V^{ab} is taking care of the angles between the bonds, containing the three particle angle potential:

$$V_{ijk}^{\text{ab}}(q_i, q_j, q_k) = k_{ijk}^{\text{ab}}(\varphi_{ijk} - \varphi_{ijk}^0)^2, \tag{4.5}$$

with

$$\varphi_{ijk} = \cos^{-1} \frac{(q_i - q_j) \cdot (q_j - q_k)}{r_{ij} r_{jk}}. \tag{4.6}$$

The dihedral interaction is then finally modeled by a four particle cosine potential, which looks like:

$$V_{ijkl}^{\text{dh}}(q_i, q_j, q_k, q_l) = k_{i,j,k,l}^{\text{dh}}(\vartheta_{ijkl} - \vartheta_{ijkl}^0)^2, \tag{4.7}$$

where

$$\vartheta_{ijkl} = \cos^{-1} \frac{(d_{ij} \times d_{jk}) \cdot (d_{jk} \times d_{kl})}{|d_{ij} \times d_{jk}||d_{jk} \times d_{kl}|}, \tag{4.8}$$

and $d_{ij} = q_i - q_j$. We like to point out that several different models of the dihedral potential exists, and that the harmonic potential is just one specific example of such a potential. Hence, we write it down in a more general form:

$$V_{ijkl}^{\text{dh}}(q_i, q_j, q_k, q_l) = f(\vartheta_{ijkl}, \vartheta_{ijkl}^0), \tag{4.9}$$

where f is a function of ϑ_{ijkl} which has minima at ϑ^0_{ijkl}. Now that we have all the bonded parts of our potential that will drive our MD simulations, we still have to take care of the non-bonded parts. The Lennard Jones term is given by:

$$V_{ij}^{\text{VDW}}(q_i, q_j) = \epsilon_{ij}\left(\left(\frac{r_{ij}^m}{r_{ij}}\right)^{12} - 2\left(\frac{r_{ij}^m}{r_{ij}}\right)^{6}\right). \tag{4.10}$$

Hence, at distance r^m the potential reaches it's minimum ϵ. r^m is thus in general the sum of the VDW radii of particle i and particle j. The only long range term and thus in most cases the most computationally intensive term in MD simulations is the Coulomb interaction:

$$V_{ij}^{\text{C}}(q_i, q_j) = k_{ij}^{\text{C}}\frac{Q_i Q_j}{r_{ij}}, \tag{4.11}$$

where Q_i is the charge of particle i. For the different non bonded equations, one usually uses cutoff ranges in order to save large amounts of computational time. Such cutoffs can be implemented in the form presented here in ϵ_{ij} in (4.10) and in k_{ij} in (4.11). Finding pairs i, j is also a nontrivial task due to periodic boundary conditions used in almost all MD simulations. The coulomb interaction is further frequently modified depending on the type of electrostatics one uses, and is calculated in several different ways in order to speed up calculations. We explicitly outline electrostatics in section 4.4 of this chapter. We now have summarized the essence of a classical MD simulation potential which is called the force field. As we have all the necessary tools available we can now go into detail describing the force fields we used.

In the simulations that we made around the TSP SD we used two popular force fields:

1. The *Optimized Potentials for Liquid Simulations All Atom* (OPLS-AA) force field. [79] [80]
2. The *Groningen Molecular Simulation* (GROMOS) force field in its 56a3 [81] and a modified 56a3 parameter set [82].

While OPLS-AA is an all atom force field where each atom is simulated, GROMOS is a so called unified atom force field where several hydrogens, in general non polar hydrogens are unified into one bead with the atom they are linked to.

We now sketch the functional forms of these force fields:

In OPLS the functional form of the linear bonds and angles adhere exactly to what has been shown in equation (4.4) and (4.5). The dihedrals are modeled by a four cosine term potential of the form:

$$\begin{aligned} V_{ijkl}^{\text{dh}-\text{OPLS}} &= \frac{1}{2}[K_1(1+\cos(\vartheta-\vartheta_0)) + K_2(1-\cos(2(\vartheta-\vartheta_0))) \\ &+ K_3(1+\cos(3(\vartheta-\vartheta_0))) \\ &+ K_4(1-\cos(4(\vartheta-\vartheta_0)))], \end{aligned} \tag{4.12}$$

where ϑ, ϑ_0 and constants K are depending on the atoms i, j, k, l. Non-bonded interactions generally adhere to the scheme presented in equations (4.10) and (4.11) and are expressed by the term:

$$V_{ij}^{\text{nb}-\text{OPLS}} = f_{ij}^{\text{OPLS}} \left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{C_{ij}}{r_{ij}^{6}} + k^C \frac{Q_i Q_j}{r_{ij}} \right), \tag{4.13}$$

where for parametrization the combining rules apply $A_{ij} = \sqrt{A_{ii}A_{jj}}$ and $C_{ij} = \sqrt{C_{ii}C_{jj}}$. f_{ij}^{OPLS} is a prefactor that scales intermolecular long range interactions, such that $f_{ij}^{\text{OPLS}} = 0.5$ for interactions that are 4 bonds away. $f_{ij}^{\text{OPLS}} = 0$ for intramolecular interactions that are less then four bonds away, and $f_{ij}^{\text{OPLS}} = 1$ otherwise. k^C is a constant, dependent on the units used.

The GROMOS force field functional representation adheres to general scheme but makes several assumptions that shall improve the computational performance. Linear bond and angle interactions become:

$$U_{ij}^{\text{lb}-\text{GROMOS}}(q_i, q_j) = k_{ij}^{\text{lb}}(r_{ij}^2 - (r_{ij}^0)^2)^2, \tag{4.14}$$

$$V_{ijk}^{\text{ab}-\text{GROMOS}}(q_i, q_j, q_k) = k_{ijk}^{\text{ab}}(\cos\varphi_{ijk} - \cos\varphi_{ijk}^0)^2, \tag{4.15}$$

avoiding either expensive square root and arc cosine calculations. To model dihedral interactions the GROMOS force field distinguishes proper and improper dihedrals. Proper dihedrals are used to keep the dihedral angle in a certain configuration, such as keeping atoms in a plane. This interaction is modeled using the harmonic form presented in equation (4.7). Proper dihedrals are modeled by a multi-cosine potential of the form:

$$V_{ijkl}^{\text{dhp}-\text{GROMOS}}(q_i, q_j, q_k, q_l) = \sum_{n=1}^{N_\vartheta} k_{ijkl;\vartheta_n}^{dhp}(1+\cos\xi_n \cos m_n\vartheta_n), \tag{4.16}$$

where the sum from $n = 1$ to N_ϑ takes into account up to 6 multiplicities of m_n of ϑ_n as well as the phase shifts ξ_n that are limited to 0 and π. Non-bonded interactions are modeled in the same way as in the OPLS force field (4.13) with the exception that $f_{ij}^{\text{GROMOS}} = 0$ for intramolecular interactions less then four bonds in away and otherwise is set to $f_{ij}^{\text{GROMOS}} = 1$.

4.3 Integration

Now that we have a force field, we would like see how a system might evolve over time using such a force field. In order to do that one uses Newtonian mechanics, such that the force F can be derived from the potential V defined by the force field, and hence:

$$F = -\nabla V. \tag{4.17}$$

The second law of motion in Newtonian systems states that:

$$m_i \frac{d^2 q_i}{dt^2} = F_i, \tag{4.18}$$

where m_i is the mass of particle i, q_i its position and F_i the force acting onto it, and thus using an initial set of coordinates q_i and momenta p_i which are given by the initial velocities $p_i = m_i \frac{dq_i}{dt}$, in order to deduce the trajectories of our particles in time. Numerically this is done in the case of the simulations that we have performed using the so- called *Leap Frog* algorithm [30]:

$$p_i(t + \frac{\Delta t}{2}) = p_i(t - \frac{\Delta t}{2}) + F_i \Delta t, \tag{4.19}$$

$$q_i(t + \Delta t) = q_i(t) + \frac{p_i}{m_i}(t + \frac{\Delta t}{2})\Delta t. \tag{4.20}$$

This algorithm is time reversible and of third order, while efficient enough to drive MD simulations, as most of the errors in MD simulations do come from the choice of the force field and its parameters hence the initial model. Simple as it might seem, one has to add constraints in order to freeze certain degrees of freedom. These are needed, for instance, to keep bond lengths or angles within a molecule in acceptable ranges during a single interation step and hence to cercumvent certain instabilities that

might occur during a simulation. Such constraints can be derived using the Euler-Lagrange formalism which is derived from the principle of least action:

$$S = \int_{t_1}^{t_2} L(t)dt, \tag{4.21}$$

where S is known to be the action and L the Lagrange function:

$$L = T - V. \tag{4.22}$$

In this function T can be identified to be the kinetic and V to be the potential energy. By principal of least action one can derive the Lagrange formalism which is a different, but equivalent way of handling Newtonian mechanics. For details the reader shall be referred to literature on classical mechanics and variational calculus such as [83] [84] [85]. These references have further helped to show what follows in here. In the variation of trajectories from one point q_{t_1} to another q_{t_2} in a classical field the Hamiltonian principle states that the action remains zero:

$$\delta S = \int_{t_1}^{t_2} dt \left(L(q + \delta q, \dot{q} + \delta\dot{q}, t) - L(q, \dot{q}, t)\right) = 0, \tag{4.23}$$

with $\dot{q} = \frac{dq}{dt}$. An arbitrary differential of a function might be approximated in first order like:

$$df = f(x + dx, y + dy) - f(x, y) = \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy. \tag{4.24}$$

Hence in first order the variation integral (4.23) becomes:

$$\delta S = \int_{t_1}^{t_2} dt \left(\frac{\partial L}{\partial q}\delta q + \frac{\partial L}{\partial \dot{q}}\delta\dot{q}\right), \tag{4.25}$$

$$\delta S = \int_{t_1}^{t_2} dt \left(\frac{\partial L}{\partial q}\delta q + \frac{\partial L}{\partial \dot{q}}\frac{d}{dt}\delta q\right). \tag{4.26}$$

Integration by parts of the last term yields:

$$\int_{t_1}^{t_2} dt \left(\frac{\partial L}{\partial \dot{q}}\frac{d}{dt}\delta q\right) = \left[\frac{\partial L}{\partial \dot{q}}\delta q\right]_{t_1}^{t_2} - \int_{t_1}^{t_2} dt \left(\delta q \frac{d}{dt}\frac{\partial L}{\partial \dot{q}}\right), \tag{4.27}$$

where using the boundary condition that $\delta q(t_1) = \delta q(t_2) = 0$ which makes the border terms disappear, one obtains that:

$$\delta S = \int_{t_1}^{t_2} dt \left(-\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} + \frac{\partial L}{\partial q}\right)\delta q = 0. \tag{4.28}$$

Since δq can be chosen arbitrarily the term in the brackets has to disappear allowing one to state:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} = \frac{\partial L}{\partial q_i}. \tag{4.29}$$

These equation are called the Euler-Lagrange equations and are sufficient to describe a classical mechanical problem. So far so good, but right now we still have to add constraints. For descriptive purposes let us suppose we have a system of two coordinates. We can now imagine a constraint of the form:

$$C(q_1, q_2, t) = 0. \tag{4.30}$$

Variation of C leads;

$$\delta C = \frac{\partial C}{\partial q_1}\delta q_1 + \frac{\partial C}{\partial q_2}\delta q_2 = 0. \tag{4.31}$$

Rewriting equation (4.28) for such a system yields:

$$\delta S = \int_{t_1}^{t_2} dt \left(\left(-\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} + \frac{\partial L}{\partial q_1} \right) \delta q_1 + \left(-\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_2} + \frac{\partial L}{\partial q_2} \right) \delta q_2 \right) = 0. \tag{4.32}$$

Eliminating δq_2 by inserting equation (4.31) into (4.32) one obtains:

$$\begin{aligned} \delta S &= \int_{t_1}^{t_2} dt \left(\left(-\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} + \frac{\partial L}{\partial q_1} \right) \frac{1}{\partial C/\partial q_1} \right. \\ &- \left. \left(-\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_2} + \frac{\partial L}{\partial q_2} \right) \frac{1}{\partial C/\partial q_2} \right) \delta q_1 = 0. \end{aligned} \tag{4.33}$$

Since again equation (4.33) has to hold for arbitrary δq_1 we can state that the inner part has to be 0 yielding:

$$\left(-\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} + \frac{\partial L}{\partial q_1} \right) \frac{1}{\partial C/\partial q_1} = \left(-\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_2} + \frac{\partial L}{\partial q_2} \right) \frac{1}{\partial C/\partial q_2}, \tag{4.34}$$

which can be decoupled by a Lagrange multiplier $\lambda(t)$:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} - \frac{\partial L}{\partial q_1} - \lambda(t)\frac{\partial C}{\partial q_1} = 0, \tag{4.35}$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_2} - \frac{\partial L}{\partial q_2} - \lambda(t)\frac{\partial C}{\partial q_2} = 0. \tag{4.36}$$

The above equations (4.30), (4.35), (4.36) can in general be solved and what has been sketched here can be generalized to:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \sum_k \lambda_k(t)\frac{\partial C_k}{\partial q_i}, \tag{4.37}$$

for k constraints and i generalized coordinates. Now that we have obtained equation (4.37) which allows us to treat mechanical systems under constraints, we still need to apply this on MD simulations and on our *Leap Frog* integrator (4.19), (4.20). Let us rewrite L from equation (4.22) a bit more specific:

$$\begin{aligned} L &= \frac{p^2}{2m} - V(q) \\ &= \frac{1}{2}m\dot{q}^2 - V(q). \end{aligned} \tag{4.38}$$

Then it is straightforward to show by inserting (4.38) into (4.37) and $F_i = -\nabla V = -\frac{\partial V}{\partial q_i}$ that

$$m_i\frac{d^2 q_i}{dt^2} = \frac{\partial}{\partial q_i}\left(\sum_k \lambda_k C_k - V\right), \tag{4.39}$$

or keeping the force term:

$$m_i\frac{d^2 q_i}{dt^2} = F_i + \frac{\partial}{\partial q_i}\left(\sum_k \lambda_k C_k\right). \tag{4.40}$$

and the constraints can thus be interpreted as a correcting force, and hence we shall be able to use the *Leap Frog* integrator in the following from:

$$p_i(t + \frac{\Delta t}{2}) = p_i(t - \frac{\Delta t}{2}) + (F_i + \sum_k \lambda_k \frac{\partial C_k}{\partial q_i})\Delta t, \tag{4.41}$$

$$q_i(t + \Delta t) = q_i(t) + \frac{p_i}{m_i}(t + \frac{\Delta t}{2})\Delta t. \tag{4.42}$$

In practice one has to calculate λ_k to find the correct positions of the particles under constraints. This can be quite cumbersome in terms of computational efficiency, but several good approaches such as *Shake* [86] and *Lincs* [87] do exist to tackle this problem and are implemented in *GROMACS* [30].

4.4 Electrostatics

Classical electrodynamics theory is built upon the Maxwell equations:

$$\nabla \cdot E = 4\pi k_1 \rho, \tag{4.43}$$
$$\nabla \cdot B = 0, \tag{4.44}$$
$$\nabla \times B = 4\pi k_2 j + \frac{k_2}{k_1}\frac{\partial E}{\partial t}, \tag{4.45}$$
$$\nabla \times E = -k_3 \frac{\partial B}{\partial t}, \tag{4.46}$$

where E is the electric field, ρ is some charge density, B is the magnetic field, j are charge currents and k_1, k_2 and k_3 are constants changing according to the unit system used and are linked by the relation $\frac{k_1}{k_2 k_3} = c^2$ where c is known to be the speed of light. One might note that j and ρ are further linked by continuity equation:

$$\nabla \cdot j = -\frac{\partial \rho}{\partial t}. \tag{4.47}$$

In most MD simulations treating molecular systems, and those that we performed, all the above is reduced to the first Maxwell equation (4.43), while the influence of the others (4.44), (4.45) and (4.46) is completely neglected. Thus no fields are generated by moving charges as they would according to classical electromagnetic theory. Equation (4.43) leads to the Poisson equation:

$$\nabla \cdot (-\nabla \Phi) = 4\pi k_1 \rho, \tag{4.48}$$
$$\nabla^2 \Phi = -4\pi k_1 \rho, \tag{4.49}$$

whose solutions are know to result in the coulomb law:

$$\Phi = k_1 \frac{\rho}{r_{ij}}, \tag{4.50}$$
$$V = Q\Phi, \tag{4.51}$$

where Q is a charge, on which the *electrostatic potential* Φ acts. The interested reader who wants to know how to arrive from the Poisson equation at Coulombs law shall be referred to books on classical electrodynamics [88] [89]. All such interactions can thus be described using the Coulomb interaction which is part of the non bonded force field term (4.13):

$$V^{\mathrm{C}} = k^C \frac{Q_i Q_j}{r_{ij}}. \tag{4.52}$$

Since the coulomb interaction is a long range interaction setting a simple cutoff on r_{ij} is not convenient. Things get even more complicated with periodic boundary conditions as particles across box boundaries can impact the system. During our simulations we used two different approaches that are implemented in *GROMACS* [30] to overcome this problem. The *Particle Mash Ewald* (PME) [90] and the *Generalized Reaction Field* (GRF) [91] algorithm.

Ewald Summation

Ewald summation has been developed before the times of computer simulations and was initially used to calculate electric potentials in crystallography [92]. It is based on the idea that one can split off the short range term from the long range term:

$$V^{\mathrm{C}} = V^{\mathrm{C-sr}} + V^{\mathrm{C-lr}}. \tag{4.53}$$

In most cases this is done using the error, and complementary error function:

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt, \tag{4.54}$$

$$\mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt = 1 - \mathrm{erf}(x). \tag{4.55}$$

Hence one can write:

$$\frac{1}{r} = \frac{\mathrm{erf}(r\sqrt{\eta}/2)}{r} + \frac{\mathrm{erfc}(r\sqrt{\eta}/2)}{r}. \tag{4.56}$$

In these terms we can rewrite the coulomb interaction to:

$$\sum_{\mathrm{i}\neq\mathrm{j}} k^C \frac{Q_i Q_j}{r_{ij}} = k^C \sum_{\mathrm{i}\neq\mathrm{j}} \left(\frac{Q_i Q_j \mathrm{erf}(r_{ij}\sqrt{\eta}/2)}{r_{ij}} + \frac{Q_i Q_j \mathrm{erfc}(r_{ij}\sqrt{\eta}/2)}{r_{ij}} \right). \tag{4.57}$$

The advantage here is that $\mathrm{erfc}(r)/r$ converges rather fast, while $\mathrm{erf}(r)/r$ does not show an improvement over $1/r$ in terms of convergence. We call this part the long range part. η can here be used to shift the importance either to left or to the right term of the equation. The basic idea is then to solve the part that converges very slow in reciprocal, or Fourier space. We will

briefly sketch how this is done in general here. Assuming periodic boundary conditions we define the lattice function:

$$M(r) = \sum_n \delta(r - n), \tag{4.58}$$

using the Dirac delta function δ and n lattice transformations. Using the Fourier transformation of the Dirac delta function and under the condition that the lattice is infinite in size the lattice function (4.58) can be written like:

$$M(r) = \frac{1}{\Omega} \sum_k e^{ik \cdot r} \tag{4.59}$$

To obtain the energy of the whole lattice one has now to fold the lattice function with the single unit cell term:

$$V^{\text{lattice}} = M(r) * \left(\sum_{i \neq j} k^C \frac{Q_i Q_j}{r_{ij}} \right) \tag{4.60}$$

Using the form stated in (4.57) we can now fold the term containing the complementary error function erfc, which converges fast for growing interparticle distances r_{ij} with the lattice function summing in real space (4.58), while we fold the term containing the error function erf and which does not converge for growing r_{ij} in with the lattice function summing in reciprocal space (4.59).

Applying what is written above the term from equation (4.57) containing the error function erf becomes:

$$\frac{k^C}{\Omega} \int d^3 r \sum_k e^{ik \cdot r} \sum_{\text{i} \neq \text{j}} \frac{Q_i Q_j \text{erf}(|r_{ij} + r| \sqrt{\eta}/2)}{|r_{ij} + r|} \tag{4.61}$$

Where Ω is the volume of a unit cell. Here the integral can be solved, being finite only for systems with zero total charge. A compensation term, as we sum over the initial unit cell too, has further to be added. This is due to the transformation of equation (4.58) to equation (4.59) as it does not hold otherwise (not shown here).

Ewald summation can be even solved faster generating meshes in reciprocal space and using highly efficient Fourier transform algorithms to sum over the wave vectors k. This is the magic in the PME [90] algorithm.

Generalized Reaction Field

A second possible solution in order to tackle long range interactions is to use the GRF method. Again this method is based on the idea to decouple long range interactions from short range interactions, but in this case the implementation is completely different. In the case of this method the coulomb term is transformed into [81]:

$$\begin{aligned} V^{\mathrm{C-Total}} &= k^C \sum_{i \neq j} \frac{Q_i Q_j}{r_{ij}} \\ &+ k^C \sum_{l \neq m} - \frac{2 Q_l Q_m r_{lm} C_{\mathrm{RF}}}{R_{\mathrm{RF}}^3} \\ &+ k^C \sum_{l \neq m} - \frac{Q_l Q_l (1 - \frac{1}{2} C_{\mathrm{RF}})}{R_{\mathrm{RF}}} \end{aligned} \tag{4.62}$$

Here C_{RF} is the reaction field constant, R_{RF} defines a cutoff radius. The first term in this equation calculates near field interactions below the cutoff radius R_{RF}. The second term adds the force due to a reaction of dielectric outside the cutoff radius R_{RF}. The third term ensures that the electrostatic energy is zero at the cutoff distance R_{RF} and is constant. One might also note the different indices in (4.62). This is due to that the second and third term in these equations might include atom pairs that are not present in the first term. The reaction field constant is shown to be:

$$C_{\mathrm{RF}} = \frac{2(1-\epsilon)(1+\kappa+R_{\mathrm{RF}}) - \epsilon(\kappa+R_{\mathrm{RF}})^2}{(1+2\epsilon)(1+\kappa+R_{\mathrm{RF}}) + \epsilon(\kappa+R_{\mathrm{RF}})^2} \tag{4.63}$$

where ϵ is known to be the relative permittivity and κ the inverse Debye length. The derivation of the solution written down here was detailed by I. G. Tironi et al. [91]. We sketch some of the details here for completeness and a better general understanding. The first Maxwell equation (4.43) in matter becomes:

$$\epsilon E = 4\pi k_1 \rho. \tag{4.64}$$

Let us define a charge density like:

$$\rho(r) = \sum_j Q_j c_j^0 e^{-\frac{Q_j \Phi(r)}{k_b T}}, \tag{4.65}$$

where c_j^0 is the mean concentration and Q_j the charge of some particle of type j. k_b is the Boltzmann constant and $\Phi(r)$ the electrostatic potential. We then have a Boltzmann distributed charge density, imagining that our charges of type j are at thermodynamic equilibrium. Using the charge density defined in (4.65) and the Maxwell equation in matter, one can find the Poisson Boltzmann equation:

$$\nabla^2\Phi(r) = -\frac{k_1}{\epsilon}\sum_j Q_j c_j^0 e^{-\frac{Q_j\Phi(r)}{k_bT}}, \tag{4.66}$$

where with the development of the exponential factor:

$$e^{-\frac{Q_j\Phi(r)}{k_bT}} \approx 1 - \frac{Q_j\Phi(r)}{k_bT}, \tag{4.67}$$

one might write down equation (4.66) in linearized form:

$$\nabla^2\Phi(r) = \frac{k_1}{\epsilon}\sum_j \frac{c_j^0 Q_j^2\Phi(r)}{k_bT} - \frac{k_1}{\epsilon}\sum_j c_j^0 Q_j^2. \tag{4.68}$$

Here the second term equals zero for systems of zero total charge. Using the Debye length:

$$\lambda_D = \frac{1}{\kappa} = \sqrt{\frac{\epsilon k_bT}{k_1\sum_j c_j^0 Q_j^2}}, \tag{4.69}$$

the linearized Poisson Boltzmann equation can now be written in the convenient form

$$\nabla^2\Phi(r) = \kappa^2\Phi(r). \tag{4.70}$$

One should further note that $\sum_j c_j^0 Q_j^2$ can be identified to be the ionic strength in an solution to be simulated by MD. The electrostatic potentials can now be split into two terms where:

$$\Phi(r) = \begin{cases} \Phi_1(r) & r < R_{\mathrm{RF}} \\ \Phi_2(r) & r > R_{\mathrm{RF}} \end{cases} \tag{4.71}$$

Then $\Phi_1(r)$ can be solved using the Poisson equation (4.49), while $\Phi_2(r)$ can be solved by the linearized Poisson- Boltzmann (4.70) equation, using the boundary conditions:

- The potential shall be continuous:

$$\Phi_1(R_{\mathrm{RF}}) = \Phi_2(R_{\mathrm{RF}}), \tag{4.72}$$

- The normal component of the dielectric displacement is continuous for a boundary that is not charged.

$$\frac{\partial \Phi_1}{\partial r} = \epsilon \frac{\partial \Phi_2}{\partial r}, \tag{4.73}$$

- The potential vanishes for $r \to \infty$:

$$\Phi_2(\infty) = 0. \tag{4.74}$$

one can find the solution shown above (4.62) [91].

We briefly summarize what was shown here: In the case of the GRF method a cutoff radius R_{RF} is used. The total system is then approximated to a system where for each particle, its electrostatic interaction with the other particles is approximated by calculating the direct interaction between them if their interparticle distance is smaller then the radius R_{RF} and by an interaction with a polarizable medium if their interparticle distance is larger then the radius R_{RF}.

4.5 Molecular Dynamics and Statistical Ensembles

MD and statistical ensembles play an important role for each other. Force field parameters for example are derived from statistical observations such as the partitioning of particles in different fluids. Thermodynamic values such as the entropy of a system are values to be derived by MD simulations. However serious limitations do exist:

- Molecular dynamics simulations are small in respect to the assumptions made in the framework of statistical physics. The number of particles simulated is inherently limited and the assumption $N \to \infty$ is important in the derivation of many principals of statistical ensembles.
- Ergodic hypotheses which states that the density of the states in the ensemble of particles is the same as the density of states explored by one particle over time. Otherwise stated, the mean in the ensemble should be the same as the mean in time, can in most cases not be achieved, as the system to be simulated will not explore all states that are possible to be found.

- Equilibrium will not be achieved, which relates to the previous point. Philosophically, as we are simulating biological systems this is further questionable as being alive is by definition out of equilibrium and only a dead object shall be in thermodynamical equilibrium.

In the theory of statistical mechanics, three ensembles are widely used:

1. The microcanonical ensemble: conserving energy, volume and the number of particles (E, V, N),
2. The canonical ensemble: conserving temperature, volume and the number of particles (T, V, N).
3. The grand canonical ensemble: conserving fugacity, volume and temperature (z, V, T)

For the molecular dynamics simulations that we performed and that are outlined later in this chapter however the so called isothermic- isobaric ensemble was used. This ensemble conserves temperature, pressure and the number of particles, but allows the volume and energy to fluctuate (T, P, N).

Isothermal- Isobaric Ensemble

The derivation for this ensemble presented here is analogous to the derivation of other statistical ensembles, such as the canonical or grand canonical and is as those derived from a small ensemble embedded into a large microcanonical ensemble. Such derivations, which have been used as a guide to what is presented here are shown in [93] [94]. Pieces from [95] have further aided the following derivation.

Let us start with Euler's equation for extensive natural variables of thermodynamics that states:

$$E(S, V, N) = \frac{\partial E}{\partial S} S + \frac{\partial E}{\partial V} V + \frac{\partial E}{\partial N} N \tag{4.75}$$

This can be rewritten as:

$$E(S, V, N) = TS - PV + \mu N \tag{4.76}$$

Here E is the energy of a thermodynamic system, S the entropy, P the pressure, V the volume, μ the chemical potential and N the number of particles inside the system. Let us transform this now

to different set of natural variables (Legendre transformation): For the natural variables (T, V, N), we obtain the Helmholtz free energy:

$$F(T, V, N) = E - TS = -PV + \mu N, \tag{4.77}$$

and it follows:

$$dF = dE - SdT - TdS = -SdT - PdV + \mu dN. \tag{4.78}$$

continuing to the Gibbs free energy:

$$G(T, P, N) = E - TS + PV = F + PV = \mu N, \tag{4.79}$$

and

$$dG = dF + PdV + VdP = -SdT + VdP + \mu dN. \tag{4.80}$$

which is the thermodynamic potential describing this isothermal-isobaric system. From here immediately follows:

$$V = \left[\frac{\partial G}{\partial P}\right]_{N,T} \quad S = -\left[\frac{\partial G}{\partial T}\right]_{N,P} \quad \mu = \left[\frac{\partial G}{\partial N}\right]_{p,T}. \tag{4.81}$$

To show some further properties let us go back to the entirely closed system, the microcanonical ensemble. The microcanonical ensemble, as energy is fixed, states that the number of possible states in phase space is limited by the energy such that:

$$\Omega(E, V, N) = \int \dots \int_{H<E} d^{3N}q d^{3N}p, \tag{4.82}$$

where Ω is the phase space volume with the entropy:

$$S = k_b \ln(\Omega). \tag{4.83}$$

Here k_b is known to be the Boltzmann constant. The entropy is known to be the generating function of the microcanonical ensemble. Using

$$dS(E, V, N) = \frac{\partial S}{\partial E}dE + \frac{\partial S}{\partial V}dV + \frac{\partial S}{\partial N}dN, \tag{4.84}$$

and comparison with

$$dE(S, V, N) = TdS - PdV + \mu dN, \tag{4.85}$$

yields:

$$\left[\frac{\partial S}{\partial E}\right]_{V,N} = \frac{1}{T}, \quad \left[\frac{\partial S}{\partial V}\right]_{E,N} = \frac{P}{T}, \quad \left[\frac{\partial S}{\partial N}\right]_{E,V} = -\frac{\mu}{T}. \tag{4.86}$$

Let us now imagine a very small system, labeled S, embedded into a large system, labeled L. We shall now allow the small system to exchange energy with the large system. Further the small system might change it's volume effectively taking up the volume of the large system. We thus might formulate:

$$E = E_S + E_L \quad E_S \ll E_L, \tag{4.87}$$
$$V = V_S + V_L \quad V_S \ll V_L, \tag{4.88}$$

and for the phase spaces:

$$\Omega(E, V, N) = \Omega_S(E_S, V_S, N_S)\Omega_L(E - E_S, V - V_S, N_L). \tag{4.89}$$

Developing the entropy for the large system we obtain with (4.86):

$$S_L(E - E_S) \approx S_L(E) - E_S\frac{\partial S_L(E)}{\partial E} \approx S_L(E) - \frac{E_S}{T_L}, \tag{4.90}$$
$$S_L(V - V_S) \approx S_L(V) - V_S\frac{\partial S_L(V)}{\partial V} \approx S_L(V) - \frac{V_S P_L}{T_L}. \tag{4.91}$$

Now getting back to phase space, from these developments for the large system with $S_L = k_b \ln \Omega$ follows:

$$\Omega_L(E_L) \approx \Omega_L(E)e^{-E_S/(k_b T_L)}, \tag{4.92}$$
$$\Omega_L(V_L) \approx \Omega_L(V)e^{-V_S P_L/T_L}, \tag{4.93}$$

which we can combine to:

$$\Omega_L(E_L, V_L) = \Omega_L(E, V)e^{(-E_s - V_S P_L)/(k_b T_L)}. \tag{4.94}$$

Pressure and temperature for the small system are imposed by the large system. Looking only at the small system, we can thus right now state that the density of states for our isothermal- isobaric ensemble behaves like:

$$\rho_S \propto e^{S_L(E,V)/k_b} e^{-(E+VP)/k_b T}. \tag{4.95}$$

In this term $e^{S_L(E,V)/k_B}$ is constant and falls out in case of normalization. Hence we can write down the sum of all states, the partition function of this ensemble:

$$Z^{\text{ii}} = C^{\text{ii}} \int \dots \int e^{-(E+Vp)/k_b T} d^{3N}q d^{3N}p, \tag{4.96}$$

where C^{ii} is constant. The expectation value of a thermodynamic observable A in this system can now be calculated by:

$$\langle A \rangle = \frac{\int \dots \int A(q,p) e^{-(E+VP)/k_b T} d^{3N} q d^{3N} p}{\int \dots \int e^{-(E+Vp)/k_b T} d^{3N} q d^{3N} p}. \tag{4.97}$$

From the partition function directly follows that the Gibbs free energy can be found using:

$$G = k_b T \ln Z^{\text{ii}}. \tag{4.98}$$

Further follows volume:

$$V = -k_b T \left[\frac{\partial \ln Z^{\text{ii}}}{\partial P} \right]_{N,T}, \tag{4.99}$$

enthalpy:

$$H = -\frac{\partial}{\partial \beta} \ln Z^{\text{ii}}, \quad \text{with} \quad \beta = \frac{1}{k_b T} \tag{4.100}$$

and entropy:

$$S = -\left[\frac{\partial G}{\partial T} \right]_{N,P} = k_b \ln Z^{\text{ii}} + \frac{H}{T}. \tag{4.101}$$

Integration itself does not generate reliable statistical ensembles in MD simulations. The simulated systems do by integration alone not keep themselves at constant temperature and constant pressure, neither is energy conserved. In order to do so, thermostat and barostat algorithms have been developed.

Berendsen Thermostat and Barostat [96]

The Berendsen thermostat and barostat are modeled after a system that is linked to its environment by an external bath. This algorithm sets the change of temperature or pressure proportional to the temperature or pressure difference of this external bath, and hence:

$$\frac{dT}{dt} = \frac{1}{\tau_T}(T_{\text{BATH}} - T), \tag{4.102}$$

$$\frac{dP}{dt} = \frac{1}{\tau_P}(P_{\text{BATH}} - P). \tag{4.103}$$

From these equations the Berendsen thermostat now either rescales the velocities in the ensemble or changes the box volume so that

they exponentially converge against the temperature of the bath. Stating that the kinetic energy is proportional to the systems temperature we can write:

$$\Delta T = \frac{\Delta t}{\tau_T}(T_{\text{BATH}} - T), \tag{4.104}$$

$$v_2^2 = C\frac{t_2 - t_1}{\tau_T}(T_{\text{BATH}} - T) + v_1^2. \tag{4.105}$$

Here C shall be a constant that can be taken into τ_T as $v^2 \propto \frac{1}{2}mv^2 \propto T$. From this together with $v_2^2 = \lambda_v^2 v_1^2$ and $v_1^2 \propto T$ one obtains:

$$\lambda_v = \sqrt{1 + \frac{\Delta t}{\tau_T}\left(\frac{T_{\text{BATH}}}{T} - 1\right)} \tag{4.106}$$

where λ_v is the rescaling factor applied to the velocities after each integration step.

If one wants to rescale the volume of a simulation box all coordinates have to be scaled accordingly. Like for the velocities we will search for a factor to be multiplied onto the lengths of the system. We begin stating that the change in volume should behave exponentially like the following:

$$\frac{dV}{dt} = \alpha^3 V. \tag{4.107}$$

A pressure change is related to the isothermal compressibility which is defined to be:

$$\beta = -\frac{1}{V}\frac{\partial V}{\partial P}. \tag{4.108}$$

Symbolically we can write that:

$$\frac{dP}{dt} = \frac{\partial P}{\partial V}\frac{dV}{dt} = -\frac{1}{\beta V}\frac{dV}{dt} = -\frac{\alpha^3}{\beta}. \tag{4.109}$$

resolving for α by inserting (4.109) into (4.103) yields:

$$\alpha^3 = \frac{-\beta(P_{\text{BATH}} - P)}{\tau_P}, \tag{4.110}$$

where β can as a constant be put into τ_P and we obtain:

$$\frac{dV}{dt} = -\frac{(P_{\text{BATH}} - P)}{\tau_P}V, \tag{4.111}$$

$$V_2 = -\frac{\Delta t(P_{\text{BATH}} - P)}{\tau_P}V_1 + V_1, \tag{4.112}$$

from which follows by stating that $V_2 = \mu^3 V_1$, that rescaling of coordinates from q to μq and simulation box length from l to μl is done like:

$$\mu = \left[1 - \frac{\Delta t}{\tau_P}(P_{\text{BATH}} - P)\right]^{1/3}. \tag{4.113}$$

We shall note here that the Berendsen algorithm was used during our simulations, especially during the equilibration phase. It is widely used as it is rather simple to implement. The main disadvantage is that this thermostat suppresses fluctuations of the kinetic energy. Properties such as heat capacity that are derived from the fluctuations are thus not correct. The following thermostat was created to overcome such problems.

Canonical Sampling and Velocity Rescaling [97]

In order to obtain the ability to have a thermostat which allows for canonical sampling, albeit that velocities are modeled after the canonical equilibrium (Boltzmann) distribution:

$$P(E_{\text{kin}})dE_{\text{kin}} \propto E_{\text{kin}}^{(\frac{3N}{2}-1)} E^{-\frac{E_{\text{kin}}}{k_B T}} dE_{\text{kin}}, \tag{4.114}$$

where $P(E_{\text{kin}})$ is the probability to find the kinetic energy E_{kin}. Bussi et al. [97] have then shown that such a distribution can be introduced by adding a stochastic term to the Berendsen thermostat. Hence one writes:

$$dE_{\text{kin}} = \frac{dt}{\tau}(E_{\text{kin}}^{\text{bath}} - E_{\text{kin}}) + 2\sqrt{\frac{E_{\text{kin}}^{\text{bath}} E_{\text{kin}}}{3N\tau}} dW. \tag{4.115}$$

Here dW is a Wiener process. For the derivation of the stochastic term the reader shall be referred to [97]. In equation (4.115) one can rewrite the kinetic energy into terms of squared velocities from which follows that the rescaling factor that is calculated for each particle individually yields:

$$\lambda_v = \sqrt{1 + \frac{\Delta t}{\tau}\left(\frac{E_{\text{kin}}^{\text{bath}}}{E_{\text{kin}}} - 1\right) + 2\sqrt{\frac{E_{\text{kin}}^{\text{bath}}}{E_{\text{kin}} 3N\tau}} dW}. \tag{4.116}$$

The authors stress that for their implementation an exact solution of equation (4.115) is necessary and more complete solution for the rescaling factor λ_v is presented in [97]. Contrary to the

classical Berendsen algorithm fluctuations in kinetic energy are conserved using this algorithm which has been used in several of our simulations.

Parinello- Raman Pressure Coupling [98]

Parinello et al. [98] have developed a barostat that conserves ensemble properties and was also used in our MD simulations. This pressure coupling algorithm introduces a deformable unit cell in the form:

$$\Omega = \det(h) = a \cdot (b \times c), \tag{4.117}$$

where Ω is the unit cell volume and a, b and c are the vectors spanning the unit cell and h is a matrix composed of the vectors $\{a, b, c\}$. In applying this notation the position of an object in the unit cell is defined by:

$$q_i = hs_i = \xi_i a + \eta_i b + \zeta_i c. \tag{4.118}$$

Using the metric tensor $G = h^T h$ the distance between two points is defined by the inner product:

$$r_{ij}^2 = (s_i - s_j)^T G (s_i - s_j). \tag{4.119}$$

The trajectories for such pressure coupling systems are afterwards derived from the Lagrange function:

$$L = \frac{1}{2}\sum_i m_i \dot{s}_i^T G \dot{s}_i - \sum_i \sum_{j>i} V(r_{ij}) + \frac{1}{2} W \mathrm{Tr}(\dot{h}^T \dot{h}) - P\Omega, \tag{4.120}$$

where $\frac{1}{2}W\mathrm{Tr}(\dot{h}^T\dot{h})$ is the kinetic term and $p\Omega$ the potential term for the box deformation. W is a parameter with the units of a mass. Equations of motions to be integrated can then be derived using equation (4.29). The resulting equations are written as:

$$\frac{d^2 h}{dt^2} = \Omega (h^T)^{-1} W^{-1} (P - P_{\mathrm{BATH}}), \tag{4.121}$$

$$\frac{d^2 q_i}{dt^2} = \frac{F_i}{m_i} - h^{-1}\left(h\frac{dh^T}{dt} + \frac{dh}{dt}h^T\right)(h^T)^{-1}. \tag{4.122}$$

These equations can then be integrated by the leap frog scheme presented in section 4.3 of this chapter. Far away from equilibrium equation (4.121) causes large oscillations that lead to strange results and hence this algorithm has to be avoided during equilibration runs. From equation (4.121) it further follows that W controls the strength of thermodynamic coupling between the barostat and the simulated system.

4.6 Application of Classical Molecular Dynamics

In the work presented in here, we have performed 66 MD simulations of various systems containing various parts of the TSP family member proteins. The simulations were made using different force fields, different baro- and thermostats. A complete overview of methods used is shown in the tables 4.1 and 4.2. This section shall describe the choices made for these simulations. All simulations were performed using version 4.0.7 of the *GROMACS* software package [30].

Prerequisites

Before a MD simulation can be started, one has to generate the simulation box. In order to generate the simulation box one has to have the molecule to be simulated at hand. For all simulations a PDB file [50] containing the molecule to be simulated was prepared. This represents the first step in the following protocol that we used in order to prepare our MD simulations:

1. Creation of a PDB file to be used as input structure: PDB files downloaded from the PDB [50] have to be modified in many cases to be usable for MD simulations. Hence we made several changes to our input structures. In the TSP-1 SD structure with PDB code 1UX6 we added the missing residues TSP-1 R980HQGK in to the input structure for simulations 39, 40, 41, 42 as numbered in table 4.2. We did this to verify if they significantly impact the simulations and whether the other simulations without these residues are correct. This was done by moving the part containing these missing residues from the TSP-2 structure with PDB code 1YO8 to the TSP-1 structure and placing it to the right position using *VMD*. After minimization (later described in this chapter), we verified whether this way we produced a correct structure, which we did. From the TSP-2 structure (PDB code 1YO8) sugars have been stripped prior usage. Further artificial selenomethionine residues used for crystallization were changed to methionine residues by replacing the selenium atom by a sulfur atom in the TSP-2 structure. Monomers from TSP-5 where separated into differ-

ent PDB files, indicated by *mol 1,2,3* in tables 4.1 and 4.2. In several other simulations only certain parts of the PDB files containing TSP SD structures have been used. Such structures are indicated in tables 4.1 and 4.2 by the residue numbers that have been simulated.

2. Converting the PDB file into a *GROMACS* structure: This is done using the *pdb2gmx* utility of the *GROMACS* suite. During this step missing hydrogens and proper C and N-termini are generated if they are missing in PDB file. *GROMACS* files containing the particles according to the force field, unified atoms in case of the GROMOS force field and all explicit atoms in case of the OPLS-AA force field are obtained from the *pdb2gmx* utility in this step.

3. Defining the simulation box structure: In all simulations we used a simulation box of dodecahedral shape with a minimal distance to from the box surface to an atom of the molecule simulated of 1.2nm. The shape of such a simulation box is shown in figure 4.1. This step has been performed using the *editconf* utility which is part of the *GROMACS* suite.

4. Solvation of the simulation box: In this step the empty space in the newly created simulation box is filled with water molecules. All our simulations are so called *explicit solvent* simulations, and thus water is simulated by all its atoms present in the simulation box. In case of the GROMOS force field the simulation box has been solvated using the SPC three point water model [99], while for simulations that have been prepared for the *OPLS-AA* force field the TIP-4P [100] four point water model has been used. This step is performed using the *genbox* utility shipped with *GROMACS*

5. Addition of ions to the solvent: This step is needed either to reach a charge neutral box, which as we have seen in section 4.4 is necessary for all methods used to evaluate electrostatic forces or to change the ionic strength of the solvent. Adding ions to the solvent is done using the *genion* utility from the *GROMACS* software suite. Ionic strength has been varied during different simulations in order to investigate effects linked to it. Most simulations however were performed using the ionic strength of human interstitial fluid stated in [101].

6. Tweaking of force field parameters: In case where the GROMOS force field has been used with a modified 53a6 parameter set taken from [82] these parameters had to be changed at hand. In order to do so, we copied all files beginning with ffG53a6 from the *GROMACS* force field database into the directory where the simulation has been performed. The file names were then changed to contain the mod- prefix. The resulting mod-ff53a6nb.itp file containing non-bonded interactions is then edited in order to change the C^6 and the C^{12} parameters of the van der Waals interaction highlighted in equation (4.13) between the OM type atom and the CA2+ type atom. Values were taken from [82] and set to $C^6 = 2.0 \cdot 10^{-3} \frac{\text{kJ}}{\text{mol}} \text{nm}^6$ and $C^{12} = 1.67 \cdot 10^{-6} \frac{\text{kJ}}{\text{mol}} \text{nm}^{12}$. Then the mod-ffG53a6.itp file is edited to contain the new mod-ffG53a6nb.itp file. Finally the topology file created, normally called topol.top has to be updated to reference to mod-ffG53a6.itp and thus our modified force field.
7. Generating a run input file: Together with a so called mdp file in *GROMACS* the run input file is generated using the *grompp* utility that is shipped with *GROMACS*. In this process all the files generated until now are unified in one single file, that can then be easily transferred to a *High Performance Computing* (HPC) center where the minimization, equilibration or MD simulation is run.

After we have all this, we can begin to minimize the forces in our box.

Minimization

Since structures in our box have almost always strange artifacts, such as hydrogens from an amino acid pointing towards a calcium ions or inappropriate water molecule placement, and also because the force field is not necessarily reproducing the reality of a crystal from which the initial structures have been built, one has to optimize the placement of the atoms in the box in order to achieve minimal forces prior being able to start with the force field integration process. In any other case, extreme large forces due to steric clashes for instance would yield strange behavior of the molecule. In the case of our simulations we used a steepest descent minimization in a primary minimization run and a *Limited memory Broyden Fletcher Goldfarb Shanno* (L-BFGS) [102]

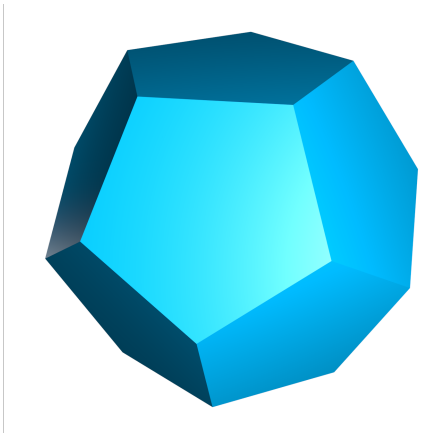

Figure 4.1: A dodecahedron: A shape like this represented the unit cell in all our simulations

algorithm in order to reach a minimized state. If the minimization did not yield in acceptable small forces, steepest descent and L-BFGS runs were run in alternating order until a small enough value has been obtained. In general, we targeted forces smaller than $10\frac{\text{kJ}}{\text{mol}\cdot\text{nm}}$ but values that were smaller then $200\frac{\text{kJ}}{\text{mol}\cdot\text{nm}}$ have been accepted. The minimum force occurring in a simulation box after minimization and prior to equilibration is highlighted for each simulation in the tables 4.1 and 4.2.

Equilibration

After minimization velocities have to be added to heat up the box. Motions in the box have then to be *equilibrated*, until one can run the MD simulation. Velocities are initially added to the solvent using a Boltzmann distribution. In order to increase statistical sampling several simulations were performed with the same initial conditions differencing only in the seed value used to generate these velocities. Afterwards, the general MD integration process takes places generating trajectories of all the molecules in the simulation box where the molecule to be observed, for instance the protein and ligands such as calcium ions rest position restrained. This kind of equilibration simulations that are performed prior the actual MD simulation have generated trajectories with a simulated time between 0.5 and 4ns. Per simulation values are highlighted in the tables 4.1 and 4.2. Dur-

ing the equilibration phase the Berendsen thermo- and barostats (see section 4.5 of this chapter) have always been used in order to prevent large oscillations of the system.

Simulation

The equilibrated system containing velocities is in the next step taken further to the simulation phase. In this phase no position restraints do exist on the molecule to be observed and its ligands. The velocity rescaling thermostat and Parinello- Raman barostat described in section 4.5 of this chapter have been used in most simulations during this phase even though some pure Berendsen thermo- and barostat simulations have been performed. The choice of the advanced thermo- and barostats has been made in order to have more accurate statistics in the simulation box. Trajectories of either 50ns or 200ns of simulated time have been generated, with integration steps between 0.5 and 2fs (Δt in equations (4.41) and (4.42)). The actual choices are shown in the tables 4.1 and 4.2. The simulations have been run either on HPC equipment installed at the ROMEO computational center of the University of Reims Champagne-Ardenne or at the *Centre Informatique National de l'Enseignement Superieur* (CINES).

Sim #	Initial Structure	N(Particles)	Volume	Field	Coulomb	T Bath	P Bath	F max	t eq	t sim	Δt	C(Na)	C(Cl)	C(Ca)
1	1Z78 (TSP-1)	30 407	323	FF56a3	GRF	Ber	Ber	9.84	2	50	2	0	5.14	0
2	1Z78 (TSP-1)	39 029	318	OPLS-AA	PME-s	Vre	P-R	9.93	1	50	2	146.21	151.44	0
3	1UX6 (TSP-1)	95 314	738	OPLS-AA	PME-s	Vre	P-R	8.23	1	50	2	144.01	146.26	36
4	1YO8 (TSP-2) without sugars	376 818	2885	OPLS-AA	PME-s	Vre	P-R	7.56	0.5	50	2	145.05	136.42	17.28
5	1UX6 (TSP-1) res 937 to 1152	28 399	301	FF56a3	GRF	Ber	Ber	8.95	4	50	2	0	5.51	0
6	1UX6 (TSP-1) res 937 to 1152	28 381	301	FF56a3	GRF	Ber	Ber	1.0	4	50	0.5	0	49.65	22.07
7	1UX6 (TSP-1)	95 365	723	OPLS-AA	PME-s	Vre	P-R	159.62	4	50	0.5	55.12	153.89	84.98
8	1UX6 (TSP-1) res 937 to 1152	28 137	297	FF56a3	GRF	Ber	Ber	7.82	1	50	0.5	139.78	458.48	156.55
9	1UX6 (TSP-1)	71 927	754	FF56a3	GRF	Ber	Ber	53.79	1	50	0.5	52.86	147.55	81.49
10	sim #6 @ 1.25ns	43 314	334	OPLS-AA	PME-s	Vre	P-R	8.22	1	50	2	0	44.75	19.89
11	1UX6 (TSP-1)	95 314	738	OPLS-AA	PME-s	Vre	P-R	8.23	1	50	2	144.01	146.26	36
12	1UX6 (TSP-1)	95 314	738	OPLS-AA	PME-s	Vre	P-R	8.23	1	50	2	144.01	146.26	36
13	1UX6 (TSP-1)	95 314	738	OPLS-AA	PME-s	Vre	P-R	8.23	1	50	2	144.01	146.26	36
14	1UX6 (TSP-1) res 827 to 841	8395	65.8	OPLS-AA	PME-s	Vre	P-R	8.76	1	200	2	126.18	126.18	50.47
15	1UX6 (TSP-1) res 842 to 864	10 402	80.0	OPLS-AA	PME-s	Vre	P-R	85.0	1	200	2	145.30	124.54	41.51
16	1UX6 (TSP-1)	71 897	738	FF56a3	PME-s	Vre	P-R	9.91	1	50	2	141.76	144.01	36
17	1UX6 (TSP-1)	71 897	738	FF56a3	PME-s	Vre	P-R	9.91	1	50	2	141.76	144.01	36
18	1UX6 (TSP-1)	71 897	738	FF56a3	PME-s	Vre	P-R	9.91	1	50	2	141.76	144.01	36
19	1UX6 (TSP-1)	71 897	738	FF56a3	PME-s	Vre	P-R	9.91	1	50	2	141.76	144.01	36
20	1UX6 (TSP-1) res 827 to 841	6008	62.0	FF56a3	PME-s	Vre	P-R	9.11	1	200	2	133.92	133.92	53.58
21	1UX6 (TSP-1) res 842 to 864	7779	80.0	FF53a6	PME-s	Vre	P-R	27.3	1	200	2	145.30	124.54	41.51
22	1UX6 (TSP-1) res 827 to 841	6008	62.0	FF56a3	GRF	Vre	P-R	9.81	1	200	2	133.92	133.92	53.58
23	1UX6 (TSP-1) res 842 to 864	7779	80.0	FF53a6	GRF	Vre	P-R	9.28	1	200	2	145.30	124.54	41.51
24	1UX6 (TSP-1)	71 897	738	FF53a6	GRF	Vre	P-R	9.0	1	50	0.5	141.72	143.97	35.99
25	1UX6 (TSP-1)	71 897	738	FF53a6	GRF	Vre	P-R	9.0	1	50	0.5	141.72	143.97	35.99
26	1UX6 (TSP-1)	71 897	738	FF53a6	GRF	Vre	P-R	9.0	1	50	0.5	141.72	143.97	35.99
27	1UX6 (TSP-1)	71 897	738	FF53a6	GRF	Vre	P-R	9.0	1	50	0.5	141.72	143.97	35.99
28	Ca^{2+} Ion	8883	64	OPLS-AA	PME-s	Vre	P-R	8.5	1	200	2	51.89	103.79	25.95
29	Ca^{2+} Ion	6517	64	FF53a6	PME-s	Vre	P-R	8.2	1	200	2	51.89	103.79	25.95
30	1UX6 (TSP-1) res 827 to 841	15 410	156	FF53a6	GRF	Vre	P-R	8.5	1	200	2	149.03	149.03	21.29
31	1YO8 (TSP-2)	280 910	2850	FF53a6	GRF	Vre	P-R	9.7	1	50	1	142.75	134.01	17.48
32	1YO8 (TSP-2)	280 910	2850	FF53a6	GRF	Vre	P-R	9.7	1	50	1	142.75	134.01	17.48
33	1YO8 (TSP-2)	280 910	2850	FF53a6	GRF	Vre	P-R	9.7	1	50	1	142.75	134.01	17.48
34	1YO8 (TSP-2)	280 910	2850	FF53a6	GRF	Vre	P-R	9.7	1	50	1	142.75	134.01	17.48

Table 4.1: MD simulations 1 to 34: Shown are from left to right, the simulation number, the initial structure used as an input to the simulation (PDB code [50] and modifications), the number of particles, the volume of the simulation box [nm^3], the force field type, the method used to evaluate the coulomb interaction, the thermostat type where *Ber* represents as Berendsen thermostat and *Ver* a velocity rescaling type thermostat with a stochastic term, the barostat type where Ber represents a Berendsen barostat and P-R a Parinello Raman barostat, the maximum force acting on a particle in the box after minimisation in $\left[\frac{\text{kJ}}{\text{mol}\cdot\text{nm}}\right]$, the time the system has spent in equilibrisation, the total runtime of the simulation, the time of a single integration step, the concentration of sodium ions in [mmol], of chlorine ions in [mmol] and of calcium ions.

Sim #	Initial Structure	N(Particles)	Volume	Field	Coulomb	T Bath	P Bath	F max	t eq	t sim	Δt	C(Na)	C(Cl)	C(Ca)
35	1UX6 (TSP-1)	71 897	738	FF53a6-mod	GRF	Vre	P-R	8.9	1	50	1	141.76	144.01	36.00
36	1UX6 (TSP-1)	71 897	738	FF53a6-mod	GRF	Vre	P-R	8.9	1	50	1	141.76	144.01	36.00
37	1UX6 (TSP-1)	71 897	738	FF53a6-mod	GRF	Vre	P-R	8.9	1	50	1	141.76	144.01	36.00
38	1UX6 (TSP-1)	71 897	738	FF53a6-mod	GRF	Vre	P-R	8.9	1	50	1	141.76	144.01	36.00
39	1UX6 (TSP-1) +missing	74 126	762	FF53a6-mod	GRF	Vre	P-R	71.7	1	50	1	143.79	150.32	34.86
40	1UX6 (TSP-1) +missing	74 126	762	FF53a6-mod	GRF	Vre	P-R	71.7	1	50	1	143.79	150.32	34.86
41	1UX6 (TSP-1) +missing	74 126	762	FF53a6-mod	GRF	Vre	P-R	71.7	1	50	1	143.79	150.32	34.86
42	1UX6 (TSP-1) +missing	74 126	762	FF53a6-mod	GRF	Vre	P-R	71.7	1	50	1	143.79	150.32	34.86
43	1YO8 (TSP-2) without sugars	280 910	2850	FF53a6-mod	GRF	Vre	P-R	9.3	1	50	1	142.75	134.01	17.48
44	1YO8 (TSP-2) without sugars	280 910	2850	FF53a6-mod	GRF	Vre	P-R	9.3	1	50	1	142.75	134.01	17.48
45	1YO8 (TSP-2) without sugars	280 910	2850	FF53a6-mod	GRF	Vre	P-R	9.3	1	50	1	142.75	134.01	17.48
46	1YO8 (TSP-2) without sugars	280 910	2850	FF53a6-mod	GRF	Vre	P-R	9.3	1	50	1	142.75	134.01	17.48
47	3FBY (TSP-5) mol 1	78 975	821.10	FF53a6-mod	GRF	Vre	P-R	55.5	1	50	1	147.63	139.54	60.67
48	3FBY (TSP-5) mol 1	78 975	821.10	FF53a6-mod	GRF	Vre	P-R	55.5	1	50	1	147.63	139.54	60.67
49	3FBY (TSP-5) mol 1	78 975	821.10	FF53a6-mod	GRF	Vre	P-R	55.5	1	50	1	147.63	139.54	60.67
50	3FBY (TSP-5) mol 1	79 975	821.10	FF53a6-mod	GRF	Vre	P-R	55.5	1	50	1	147.63	139.54	60.67
51	3FBY (TSP-5) mol 2	80 237	830.07	FF53a6-mod	GRF	Vre	P-R	68.3	1	50	1	148.05	136.04	58.02
52	3FBY (TSP-5) mol 2	80 237	830.07	FF53a6-mod	GRF	Vre	P-R	68.3	1	50	1	148.05	136.04	58.02
53	3FBY (TSP-5) mol 2	80 237	830.07	FF53a6-mod	GRF	Vre	P-R	68.3	1	50	1	148.05	136.04	58.02
54	3FBY (TSP-5) mol 2	80 237	830.07	FF53a6-mod	GRF	Vre	P-R	68.3	1	50	1	148.05	136.04	58.02
55	3FBY (TSP-5) mol 3	76 995	801.92	FF53a6-mod	GRF	Vre	P-R	9.8	1	50	1	151.34	138.90	60.12
56	3FBY (TSP-5) mol 3	76 995	801.92	FF53a6-mod	GRF	Vre	P-R	9.8	1	50	1	151.34	138.90	60.12
57	3FBY (TSP-5) mol 3	76 995	801.92	FF53a6-mod	GRF	Vre	P-R	9.8	1	50	1	151.34	138.90	60.12
58	3FBY (TSP-5) mol 3	76 995	801.92	FF53a6-mod	GRF	Vre	P-R	9.8	1	50	1	151.34	138.90	60.12
59	1YO8 (TSP-2) res 847 to 1171	72 502	743.21	FF53a6-mod	GRF	Vre	P-R	9.5	1	50	1	149.70	138.53	33.52
60	1YO8 (TSP-2) res 847 to 1171	72 502	743.21	FF53a6-mod	GRF	Vre	P-R	9.5	1	50	1	149.70	138.53	33.52
61	1YO8 (TSP-2) res 847 to 1171	72 502	743.21	FF53a6-mod	GRF	Vre	P-R	9.5	1	50	1	149.70	138.53	33.52
62	1YO8 (TSP-2) res 847 to 1171	72 502	743.21	FF53a6-mod	GRF	Vre	P-R	9.5	1	50	1	149.70	138.53	33.52
63	1YO8 (TSP-2) res 847 to 1171	72 502	743.21	FF53a6	GRF	Vre	P-R	9.5	1	50	1	149.70	138.53	33.52
64	1YO8 (TSP-2) res 847 to 1171	72 502	743.21	FF53a6	GRF	Vre	P-R	9.5	1	50	1	149.70	138.53	33.52
65	1YO8 (TSP-2) res 847 to 1171	72 502	743.21	FF53a6	GRF	Vre	P-R	9.5	1	50	1	149.70	138.53	33.52
66	1YO8 (TSP-2) res 847 to 1171	72 502	743.21	FF53a6	GRF	Vre	P-R	9.5	1	50	1	149.70	138.53	33.52

Table 4.2: MD simulations 35 to 66: Shown are from left to right, the simulation number, the initial structure used as an input to the simulation (PDB code [50] and modifications), the number of particles, the volume of the simulation box [nm^3], the force field type, the method used to evaluate the coulomb interaction, the thermostat type where *Ber* represents as Berendsen thermostat and *Ver* a velocity rescaling type thermostat with a stochastic term, the barostat type where Ber represents a Berendsen barostat and P-R a Parinello Raman barostat, the maximum force acting on a particle in the box after minimisation in $\left[\frac{\text{kJ}}{\text{mol}\cdot\text{nm}}\right]$, the time the system has spent in equilibrisation, the total runtime of the simulation, the time of a single integration step, the concentration of sodium ions in [mmol], of chlorine ions in [mmol] and of calcium ions.

4.7 Analysis of Classical Simulations

Several different tools have been used to analyze the resulting trajectories of the molecular dynamics simulations. Many of them have been shipped with *GROMACS*, others tools especially *VMD* [37] but to lesser extend also *PyMOL* [38] have been used to investigate the trajectories. Further, several in house developed modules and scripts were used during the analyzation process. The *Wolfram Mathematica* software version 7 [103] also proved to be efficient in several tasks analyzing the outcome of our MD simulations. This section here describes how we analyzed our trajectories and how we obtained the results presented in section 4.8.

Basic Verification and Trajectory Conversion

The first thing that has been done after a MD simulation finished was to verify if the minimal distance between the molecule to be observed and its mirror images created by periodic boundary conditions did not get to close to each other. This has been verified using the *g_mindist* utility from the *GROMACS* suite. The distances did not descent below a minimum distance of one nanometer between the molecule and its next image. Afterwards the trajectory was modified so that the molecule to be observed always stays in the center of the box and the solvent is moving around the molecule, instead of having the molecule diffuse in the box. This proved to be problematic as we simulated a protein with ions as bound to the protein and several ions were not indicated to be in the same box as the protein during trajectory conversion. The most efficient way in converting our trajectories using the *trjconv* utility shipped with *GROMACS* proved to be the following two commands:

```
trjconv -f traj.xtc -o centered-atomic.xtc -s topol.tpr -pbc atom \
   -ur compact -center
trjconv -f centered-atomic.xtc -o centered-final.xtc -s topol.tpr -pbc whole
```

Listing 4.1: Trajectory alignment

where the first line puts the index group to be selected into the center of the box, while the second line makes broken molecules at the edge of the box whole. In the resulting trajectory the molecule is still rotating in the center of the box. We found that any further alignment option used with *trjconv* destroyed our en-

semble in the box. In order to circumvent this problem in cases where we wanted to align the trajectory and hence prevent the molecule from rotating in the box and have the solvent around instead rotate around the molecule, we used *VMD* to do so.

Global Conformational Change and Fluctuations

Basic insights into flexibility and local fluctuations of a simulated molecule have been investigated in calculating the *Root Mean Square Deviation* (RMSD) and the *Root Mean Square Fluctuation* RMSF. Both can be obtained by either *g_rms* for the RMSD and *g_rmsf* for the RMSF from utilities integrated in *GROMACS* package. The RMSD is calculated as:

$$\mathrm{RMSD}(t_0, t) = \sqrt{\frac{1}{M}\sum_{i=1}^{N} m_i \|r_i(t_0) - r_i(t)\|^2}, \tag{4.123}$$

where $M = \sum_{i=1}^{N} m_i$, m_i is the mass of particle i, and r_i its location. This formula gives for time t a value for the global deformation against a reference time t_0. First this to be efficient the structure at t has to be aligned against the structure of t_0, which is automatically done by *g_rmsd*. The RMSF formula by contrast is given by:

$$\mathrm{RMSF}_i = \sqrt{\frac{1}{T}\int_{t_0}^{T} \|r_i(t) - \bar{r}_i\|^2}, \tag{4.124}$$

which effectively becomes as one evaluates only discrete timesteps (frames):

$$\mathrm{RMSF}_i = \sqrt{\frac{1}{T}\sum_{t_{j=0}}^{T} \|r_i(t_j) - \bar{r}_i\|^2}. \tag{4.125}$$

Here T is the time duration of the total simulation. The RMSF returns the fluctuation of a particle i around its mean location r_i and hints the flexibility of certain regions of the structure. Both formulas were in general applied to all simulations that we made.

Sampling Interatomic Distances

Interatomic distances have been sampled using the *g_dist* utility from *GROMACS*. The distances have, for instance, been sampled between the oxygen atoms of the aspartic acid heads and

the calcium ions in order to gain insights into the deformation of a C- or N-type *Stalk* from a TSP SD holding an ion and further to investigate the process of releasing an ion from such a repeat.

Sampling Changes in the Radius of Gyration

The radius of gyration roughly estimates the compactness of a molecule. It is calculated by the formula:

$$R_{\mathrm{g}} = \sqrt{\frac{\sum_{i=1}^{N} \|r_i - r_c\|^2 m_i}{\sum_{i=1}^{N} m_i}}, \tag{4.126}$$

where N is the number of particles in the molecule to be observed, r_i and m_i are the radius and the mass of particle i and r_c is the center of mass of N particles. Changes in the radius of gyration are a good indicator for conformational changes, without the need of visual inspection on the part to be investigated with a graphical program such as *VMD*. We sampled the radius of gyration for the TSP SD *Stalk* and *Globe* for instance in order to investigate changes in conformation, and flexibility.

Surface Accessibility Sampling

Solvent surface accessibility is an important property in that it hints binding availability of amino acid residues. Surface accessibility has been sampled from a trajectory using the secondary structure assignment tool *DSSP* [104] in combination with the *GROMACS* utility *do_dssp*. This utility can generate an image file depicting the surface accessibility's in different shades of gray. This image file has then been imported into *Mathematica* where we generated surface accessibility graphs from this data.

Principal Component Analysis

Principal component analysis (PCA) is a statistical method to reduce the dimension of a system. We briefly sketch this method herein. The interested reader wanting to know more on this topic shall be referred to books in the field of statistics like [105]. We start with the covariance:

$$\sigma_{XY} = \frac{1}{N} \sum_{i=1}^{N} (x_t - \bar{x})(y_t - \bar{y}). \tag{4.127}$$

where X is a set of coordinates of particle x_t in time t and Y is a set of coordinates in time t for an other particle. $\bar{x}$ and $\bar{y}$ denote the mean values of these sets of coordinates. If and only if the fluctuations for particle x and particle y are completely uncorrelated in time it follows that $\sigma_{XY} = 0$. For a many particle system such as a protein one can now define the correlation matrix:

$$C_{ij} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1N} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{N1} & \sigma_{N2} & \cdots & \sigma_{NN} \end{bmatrix}, \tag{4.128}$$

which holds the standard deviations in the diagonal and the covariances in the off diagonal. This matrix is one way to locate fluctuating atoms (high values in the diagonal) and to find collective motions (high values in the off diagonal). PCA involves right now solving the eigenvalue problem:

$$Cv_i = \lambda_i v_i, \tag{4.129}$$

where the eigenvector v_i is called a principal component with the corresponding eigenvalue λ_i. Since the matrix C is self-adjoint vectors v_i form a complete bases set in which one could express the motions of the trajectory. However, it is convenient that one can sort these principal components by their eigenvalues λ_i and can neglect the components with small eigenvalues and hence reduce the number of coordinates of the system. The first principal components (those with the highest associated eigenvalues) thus contain most of the molecules motions. When performing PCA on a MD trajectory one has to make sure to avoid diffusion. This is achieved by centering the molecule and aligning the structures of the different frames on each other. We have used PCA in order to evaluate the force fields against each other and in order to detect flexible parts and major motions of the proteins in several simulations. In our case, we always performed the PCA only on the α-carbons of a protein. Further one should make sure, as this is a statistical method, that sampling is adequate. In order to do so, we always concatenated four equivalent simulations that only differ by the seed that was used in generating the velocities during equilibration phase. PCA can be performed using the utilities shipped with *GROMACS* namely *g_covar* and *g_anaeig*. Afterwards various post-processing scripts have been written.

First, a very simple script called *split-ev-com* has been written in order to split of the components for the eigencomp.xvg file generated with *g_anaeig* into separate files so that they are easier to manipulate. Running this script is necessary to run a second script that we created called *gen-v-1-8*. This utility creates a vector from the first eight principal components by applying the formula:

$$v_{1-8} = \sum_{i=1}^{8} \lambda_i v_i, \tag{4.130}$$

which can be used to investigate the cumulated motions from these components. Scripts have further been written in order to visualize principal components using *VMD*. This is outlined in section 6.3.

Ion Coordination Sphere Sampling

The TSP SD contains a multitude of calcium ions, as shown in figure 3.6. We were interested whether these ions do leave the protein and are solvated or not. In order to better understand this process we generated graphs depicting the members of the first solvation sphere around the calcium ions. Calcium ions are known to have around eight atoms in their first hydration sphere [106] [107]. In order to sample the atoms that surround the calcium ion, we have written a *GROMACS* plugin and a secondary filtering script that allows us to highlight atoms belonging to specific molecules. The *GROMACS* plugin that we called *g_ioncoordination* generates a file for each ion part of a *GROMACS* index group to be selected. A *GROMACS* index is a database where one can regroup objects in a simulation box in order to faciliate selection of these. For futher information the reader is referred to the documentation shipped with *GROMACS* [30]. To compile such a *GROMACS* plugin, we refer to the template shipped with *GROMACS* and installed by default in:

```
GROMACS_PREFIX/share/gromacs/template
```

Listing 4.2: *GROMACS* plugin template path

The compiled plugin can then be run by issuing a command like the following:

```
g_ioncoordination -s topol.tpr -n index.ndx -f traj.xtc -cutoff 0.28
```

Listing 4.3: Running *g_ioncoordination*

Here one has to choose a *GROMACS* index group containing the ions. 0.28 was in this case the cutoff radius in nanometers. The files generated by *g_ioncoordination* for each ion in such an index group then look like:

```
 [   0.000]
- OD2 - ASP - 104
- OD2 - ASP - 106
- OD1 - ASP - 113
- OD2 - ASP - 113
- O - CYS2 - 115
- O - ASN - 118
- O - VAL - 119
- O - ILE - 121
- CA - CA - 335
[   1.240]
- OD2 - ASP - 104
- OD2 - ASP - 106
- OD1 - ASP - 113
- OD2 - ASP - 113
- O - CYS2 - 115
- O - ASN - 118
- O - VAL - 119
- O - ILE - 121
- CA - CA - 335
```

Listing 4.4: Typical output file from *g_ioncoordination*

In this file the time of the current simulation frame is outlined in brackets. The atoms within the cutoff radius around the ion are listed in the following lines. Each line contains the type of atom as defined by the force field. The residue the atom belongs to, and the residue number in *GROMACS* convention (the residue numbers start at 1 for each protein, peptide at the N-terminal). Once we obtained these files, we wrote a filtering script that we called *coordination-analyze*. Running the *coordination-analyze* script like

```
coordination-analyze input output
```

Listing 4.5: Running *coordination-analyze*

where `input` is one of the output files generated by *g_ioncoordination* and `output` is the output file to be written. *coordination-analyze* then produces an output file that looks like the following:

```
0.000000 0 4 4 0
1.240000 0 4 4 0
2.480000 0 4 4 0
3.720000 0 4 4 0
4.960000 0 4 4 0
6.200000 0 4 4 0
7.440000 0 4 4 0
8.680000 0 4 4 0
```

```
9.920000 0 4 3 0
```

Listing 4.6: A typical output file from *coordination-analyze*

The first row of such a file presents the time of the current frame, the second the number of oxygen atoms from water molecules surrounding the molecule, the third the number of oxygens being part of a protein's backbone, the fourth the number of oxygens being part of the head of an aspartic amino acid and the fifth other oxygen atoms within the cutoff sphere selected when triggering *g_ioncoordination* in listing 4.3. Such output files can then be easily visualized by tools like *gnuplot* [108]. We used *Mathematica* [103] in order to generate graphs from which we decided whether an ion got fully solvated (all coordinated oxygens are part of a water molecule) or not.

Visual Inspection

Several results have been obtained by visual inspection of trajectories generated by the simulations made. The *VMD* [37] and *PyMol* [38] have been used in this task. Especially interactions between the the *Globe*, *Stalk* and EGF like domains have been investigated in using *VMD*.

4.8 Results from Classical Simulations

In this section, we highlight the results that we have obtained from our simulations, which are both of technical, for example results on force field properties, and of biological interest as we simulated the TSP SD. As shown in tables 4.1 and 4.2, simulations were made using structures of the TSP-1,2 and 5 SD and we obtained several interesting insights on these that we shall present in the following. Two simulations of the N-terminal, without further investigations have been made to get comfortable with our MD tools.

Insights into Force Fields

During our simulations, we observed that the GROMOS [81] [82] force field leads to increased flexibility on protein structures when it is compared to the OPLS-AA [79] [80] force field. We further noticed that changes in conformation seem to happen faster in the *GROMOS* force field than in the OPLS-AA force

field. We suppose that this is due to the reduced degrees of freedom as GROMOS is a unified atom force field. These results have been observed either by calculating the RMSD using equation (4.123), the RMSF using equation (4.125) as well as by PCA shown in section 4.7 as the eigenvalues obtained in solving equation (4.129) are smaller in simulations made using the OPLS-AA force field than in simulations performed with the GROMOS force field. Such eigenvalues are shown in table 4.3. A comparison of the RMSF values for α-carbons is shown in figure 4.2.

Further we found that during simulations made using the OPLS force field the ions always stayed in their positions and never got solvated. We thus got more and more interested by the GROMOS force field, in which according to E. Project et al [82] the bonds between the calcium ions and aspartic amino acids are to weak. We thus integrated the modifications suggested.

Several other properties around the ions seem to be different. In simulations made with a single calcium ion and with two chlorine counter ions in a water box we found significant differences in ion binding. In the case of OPLS a $Ca^{+}Cl$ complex existed for longer periods during the simulation. In simulations using the GROMOS force field in its 56a3 parametrization the complex formed several times but dissolved itself rather fast. From this we calculated the difference in Gibbs free energy between the bound and the unbound state. According to (4.98) this can simply be done by:

$$\Delta G = k_b T \ln\left(\frac{P(\text{bound})}{P(\text{unbound})}\right), \tag{4.131}$$

where $P(\text{state})$ shall be the probability to be in the referred state. For simulation 28, the OPLS case we obtained $\Delta G = -4.79 k_b T$, while for simulation 29 the GROMOS case we obtained $\Delta G = -0.21 k_b T$. In order to obtain the numbers for the probabilities the *g_ioncoordination* script plus a simple counting script proved helpful.

Finding that ions did not solvate in the OPLS [79] [80] force field, we changed to the GROMOS force field in its 53a6 parametrization [81] and later incorporated the modifications suggested by E. Project et al [82]. Regarding the coulomb parts of the force field, we did not find any significant differences in ion solvation and conformational flexibility between PME or GRF calculations.

EV	OPLS	53a6
1	1.680	8.513
2	0.960	6.047
3	0.566	3.920
4	0.417	2.111

Table 4.3: The first four eigenvalues from PCA of simulations made using different force fields. OPLS [79] [80] eigenvalues are created from a concatenated α-carbon trajectory from simulations 3, 11, 12 and 13. GROMOS 53a6 [81] parametrization eigenvalues are made from simulations 16,17,18 and 19. Values are in [nm^2]

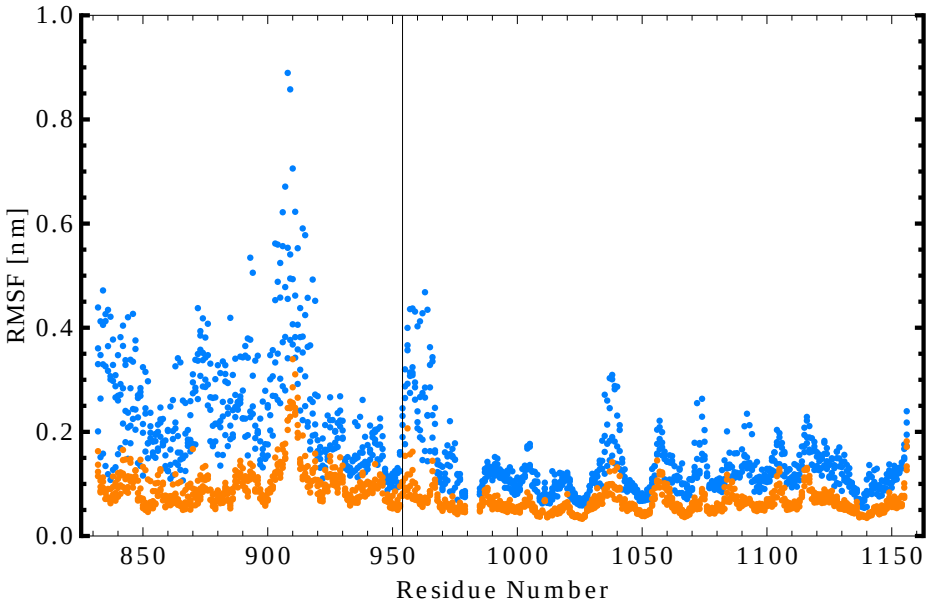

Figure 4.2: RMSF values for the α-carbons from the structure designated with PDB code 1UX6: In orange values obtained from simulations using the OPLS-AA [79] [80] force field and in blue values from simulations using the GROMOS force field in its 53a6 parametrization [81]. Values shown are taken from simulations 3, 11, 12, 13, 16, 17, 18 and 19 as referenced in table 4.1. The black vertical line between residue 954 and 955 highlights the limit between the *Stalk* and the *Globe*.

Interactions Between the *Globe* and the 8N *Stalk* Repeat in the Thrombospondin Signature Domain

Significant differences do exist in the interaction between repeat 8N from the TSP SD *Stalk* and the *Globe*. TSP-1 and 2 SDs have found to have such an interaction, while TSP-5 SD does not. Further differences do exist in the type of interaction between TSP-1 and TSP-2 SDs. Hydrogen bond interactions involving residues TSP-1 D848, S849, Y1029 and S1032 are conserved in TSP-2 by corresponding residues TSP-2 D850, N851, Y1031 and S1034. However, in TSP-2 a sidechain-sidechain interaction exists between residue TSP-2 N851 from 8N and TSP-2 S1034. This interaction is missing in TSP-1 as the corresponding residue to the aspargine TSP-2 N851 is an aspartic acid TSP-1 D850 [68] [69] [71]. Using our simulations, we found that these interactions modify the dynamic behavior of the protein. In TSP-1 we found that the hydrogen bond interaction between the residues from 8N, TSP-1 D848, S849 and the residues from the *Globe* TSP-1 Y1029, S1032 are maintained in 10 out of 16 simulations that have been made using the structure referenced by the PDB code 1UX6. In the six cases where the interactions are not maintained they are either be completely lost or different types of interactions between the *Globe* and the 8N repeat have been formed. In two cases we found that TSP-1 R1142 undergoes sidechain-sidechain interactions with either residue TSP-1 D848 or S849. This may hint a role of TSP-1 R1142 in the *Globe Stalk* interaction.

The situation turns out to be a bit different for TSP-2. In this case the interactions between repeat 8N and the *Globe* are only lost 2 out of 12 simulations that have been performed using the entire protein structure and calcium ions found in the PDB by the code 1YO8. As this structure contains, contrary to the TSP-1 structure, the entire *Stalk* and three EGF like repeats, we further cut the TSP-2 structure to the size of the TSP-1 structure. In the eight simulations that have been performed using this structure we found that the interactions between the 8N repeat of the *Stalk* have been retained completely in three simulations, in four at least partially and were completely lost in only one simulation.

Hence we speculate from these results that the interaction between the 8N repeat of the *Stalk* and the *Globe* is stronger in TSP-2 than in TSP-1. The observations regarding this interaction

were made by visually inspecting the first frames and the last frames from the performed simulations.

For the residues outlined in this paragraph we referre the reader to the sequence alighment shown in figures 3.7 and 3.8.

EGF-Stalk Interactions in the Thrombospondin Signature Domain

Published structural information, regarding the interactions between the EGF like repeats and the *Stalk* of the TSP SDs, only exists for TSP-2 and TSP-5 SDs. The published structures of TSP-2 contain all three EGF like repeats, while the TSP-5 structure only contains one EGF like repeat.

In the crystal structure found by the PDB code 1YO8 residue TSP-2 D673 from the EGF like domain is situated right next to residues TSP-2 D943, K933, D934 and R928. A shared hydrogen bond links residues TSP-2 D673, D943 and R928. This is depicted in figure 4.3. As these residues are conserved in TSP-1 we suppose that this interaction might exist in TSP-1 as well. Further residue TSP-2 F671 and R928 form a cation-π interaction. In TSP-1 the corresponding residue to TSP-2 F671 is TSP-1 Y659 and due to similarity in these residues and we suggest that this cation-π interaction is conserved as well, as is, we suppose, a lateral hydrogen bond between TSP-2 R625 and D926, as these residues are again conserved in TSP-1. Eight simulations have been performed of the TSP-2 signature domain containing all three EGF like repeats. All eight simulations have retained the interaction between TSP-2 F671 and R378, the other interactions have however been lost.

In the structure of the TSP-5 SD referenced by its PDB code 3FBY which contains only one EGF like repeat we identified a hydrogen bond interaction between TSP-5 E246 and K516. This hydrogen bond interaction is lost at some point in all our simulations.

Results regarding the *EGF-Stalk* interactions have here been obtained in visually inspecting the first and last frames of the MD simulations using *VMD*.

For the residues outlined in this paragraph we referre the reader to the sequence alighment shown in figure 3.7.

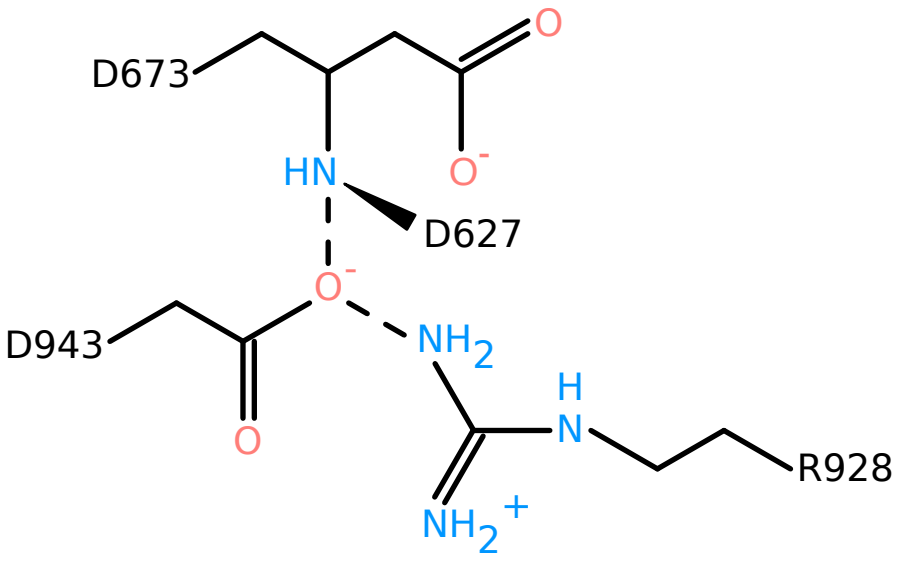

Figure 4.3: Shared hydrogen bond between the *Stalk* and the EGF like repeat of TSP-2

Depletion of Calcium Ions from the Thrombospondin Signature Domain

The TSP SD's functions and conformations have been found to be dependent on the state of the protein [32] [66] [25] [26]. State here refers to either being in calcium replete or calcium deplete form.

At time of writing no structure of any TSP in calcium deplete form has been deposited, which we suppose is probably due to a lack of organization and increased flexibility. Further details have been noted in section 3.2. Trying to get insights into calcium depletion using MD simulations is a challenging task. Several problems do exist, such as that current standard classical molecular dynamics force fields do not take the effect of polarization into account, which in reality will surely take place when a divalent ion approaches a protein's surface. The other problem to overcome is that ions seem to be well parametrized as ligands in chelation structures or in solution, but not in the process of hydration and dehydration [82]. Even though we found further differences regarding force fields that were discussed earlier in

this chapter. Aware of all these problems, we still tried to see whether we could enlighten the process of depletion, and the question on the structure of the ion depleted form of the TSP SDs. When investigating the first coordination sphere of a calcium ion, we defined a calcium ion to be solvated if all atoms in this coordination sphere have been found to be a part of a water molecule. A typical result of such an investigation can be found in figure 4.4. Such a figure has been generated using *g_ioncoordination* described in section 4.7 for each ion in every simulation. Using these figures, it proved to be straightforward to tell whether an ion depleted into the solvent or not. From these results we calculated the probability that an ion gets solvated by taking the number of simulations in which this ion is completely hydrated in its first coordination sphere dividing it by the number of simulations in which this ion exists as part of the structural input. Two probabilities have been calculated, one only taking simulations of the GROMOS force field in it's 53a6 parametrization [81] and one only taking simulations with the modified 53a6 parametrization set [82] into account. The probabilities are shown for the 53a6 parametrization set in figure 4.5, and for the modified 53a6 parametrization set in figure 4.6. In [64] T. M. Misenheimer et al stated that among the calcium ions in the TSP-2 SD approximately 10 are exchangeable. Using MD simulations and the dissociation probabilities shown in figures 4.5 and 4.6, we can hint the location of these *exchangeable* ions in the TSP SD structure. Interestingly we found about 9 ions, which perfectly corresponds to the observations made by T. M. Misenheimer et al, that were hydrated at least once in the simulations made using the modified 56a3 parametrization set.

Besides indicating the location of ions that are more loosely bound than others to the protein, we were interested in the way that these ions are bound to the protein, and how the process of depletion is happening. In particular, we investigated the process of depletion in N and C type repeats of the *Stalk* of the TSP SD. As shown in section 3.2 in figure 3.7 such repeats have high identity in the the amino acid pattern XDXDXDXXXXXD. We thus used basic distance measurements between the heads of the aspartic amino acids and the calcium ions to further investigate this process. Using those distance measurements we were able to distinguish four different processes:

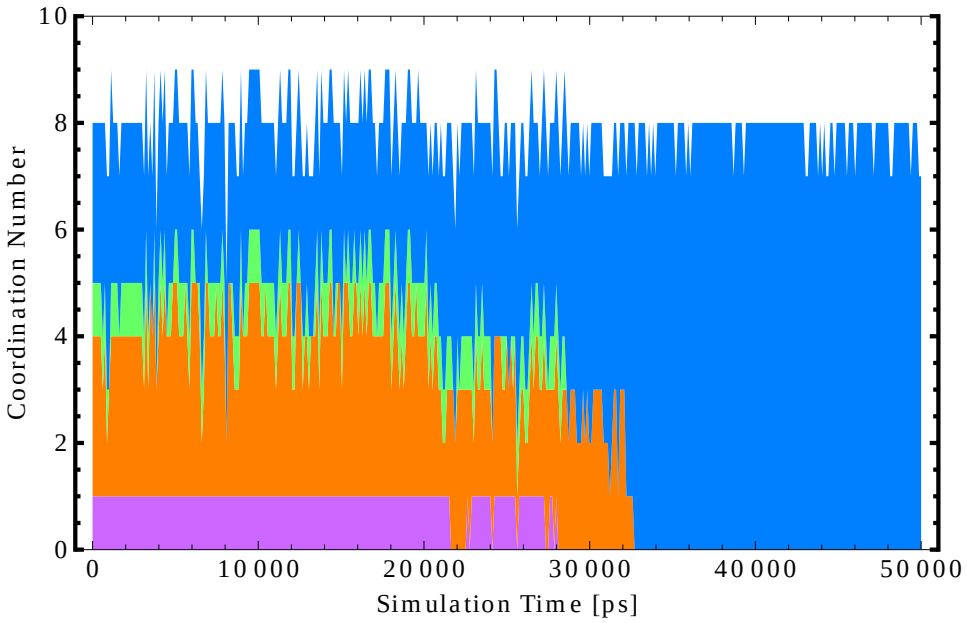

Figure 4.4: Coordination sphere during a molecular dynamics simulation: Shown here is the coordination sphere of ion number 6 as shown in figure 3.6 from simulation 31 as indicated in table 4.1. Oxygens from water molecules are shown in blue, from aspartic acid heads in orange, from the backbone in green and other oxygen atoms in violet

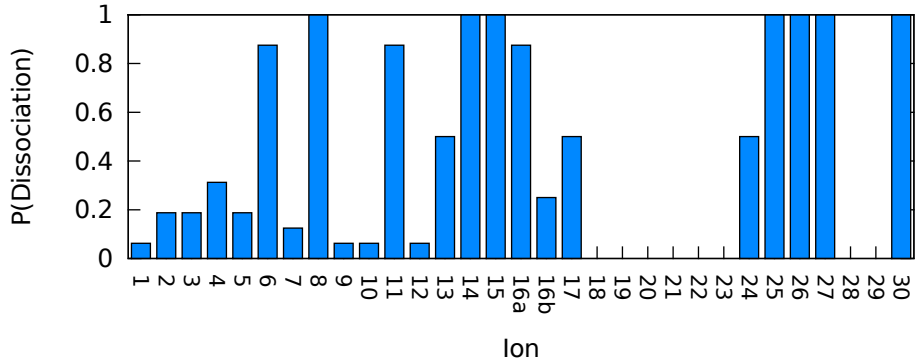

Figure 4.5: Dissociation probability of TSP SD ions in the GROMOS force field using the 53a6 parametrization set [81]: Ions are numbered like in figure 3.6. Ion number 30 represents the ion situated between the first and second EGF-like domain.

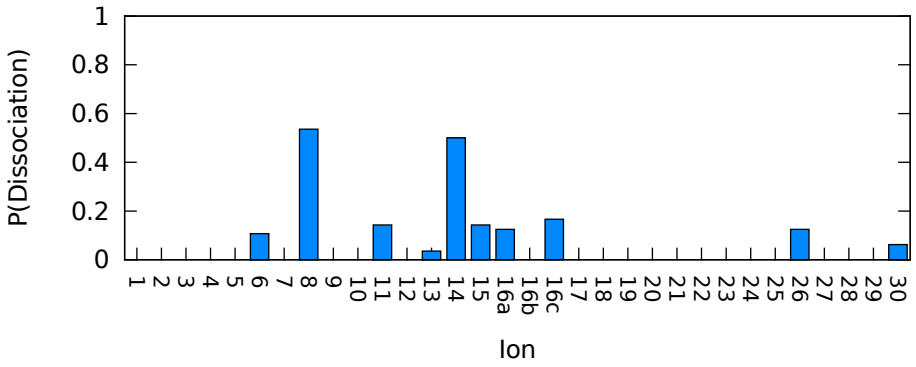

Figure 4.6: Dissociation probability of TSP SD ions in the GROMOS force field using the modified 53a6 parametrization set [82]: Ions are numbered like in figure 3.6. Ion number 30 represents the ion situated between the first and second EGF-like domain.

1. The four conserved aspartic amino acids rest in place, the ion pocket rest stable, and the ion does not leave the pocket.
2. The aspartic amino acid at position 13 of the repeat to be investigated opens the pocket.
3. The ion pocket is flexible, the ion departs from the pocket during the simulation.
4. The ion quits the pocket almost instantaneously at the beginning of the simulation.

The different behaviors of a stalk repeat are highlighted in figure 4.7. We further found that in the second case the aspartic amino acid was capable to take one of the ions from the pocket with it effectively exposing it to the solvent. One should also notice that several of the N and C type repeats have additional aspartic amino acid residues as highlighted by the sequences of the *Stalk* in figure 3.7. The ion was found to change its binding partners during the simulations, we made in such repeats, which might hint additional flexibility.

Structural Flexibility of the Thrombospondin Signature Domain

Insights into the TSP SD motions have been obtained either by RMSF, PCA, changes in radius of gyration or by visual investigation of the MD simulation trajectories. All of the methods

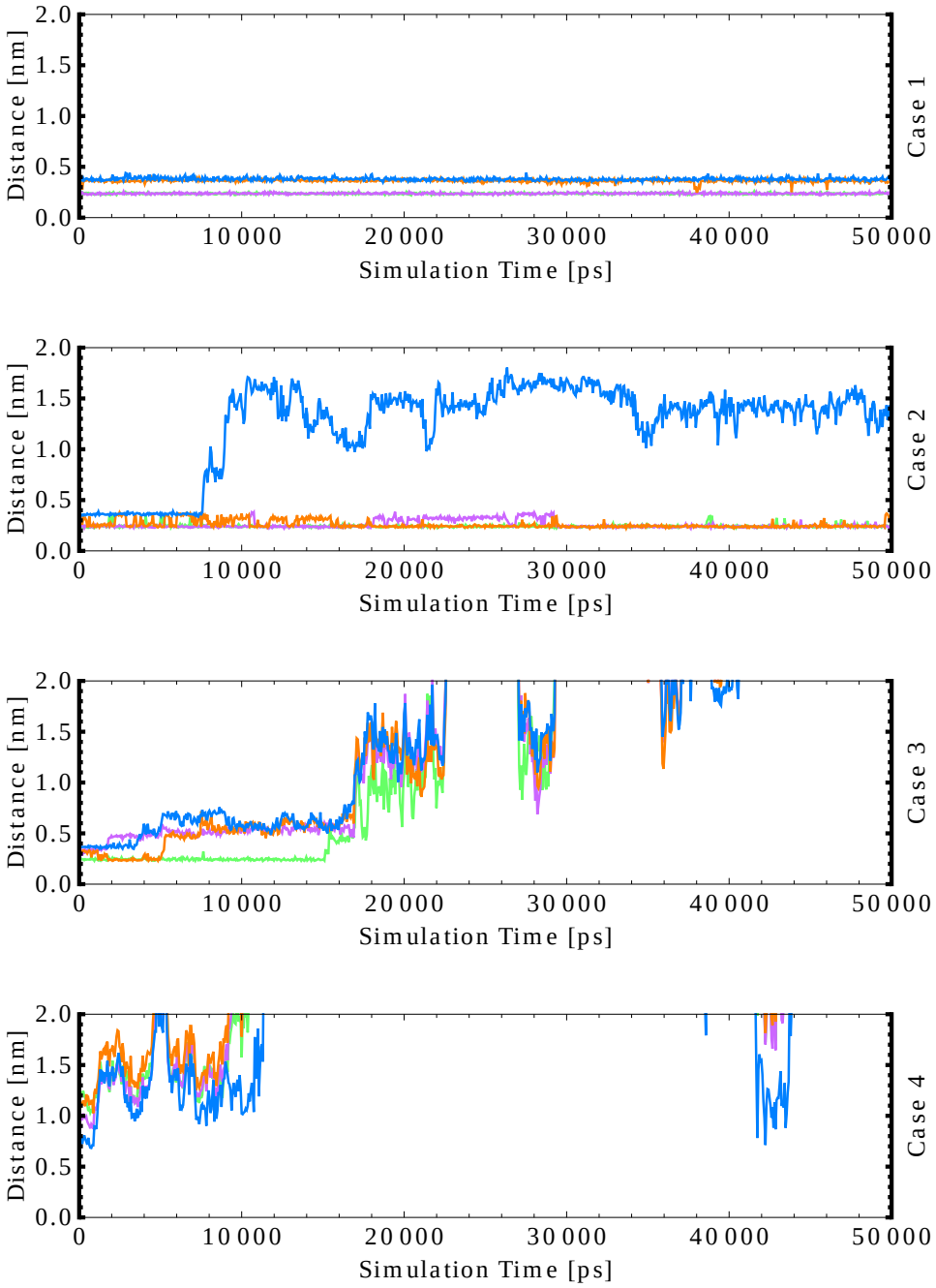

Figure 4.7: Different behaviors of a *Stalk* repeat: Shown are the distances from the calcium ion to the aspartic amino acid head oxygens. Distances are colored such that the aspartic aminio acids correspond to the repeats sequence like XDXDXDXXXXXXD.

indicated that the *Globe* is very rigid compared to the *Stalk* and the EGF like domains. We shall describe the three structural units of the SD and their flexibility here, beginning with the EGF like domains:

In the crystal structure obtained from the PDB by the code 1YO8 one ion can be found between the first and second EGF like repeat. Using RMSF calculations we tried to investigate the impact of this ion on the structures flexibility. We did this by calculating the mean of the RMSF values for each α-carbon from simulations where the ion was solvated and where it was not. We found that the loop containing residue TSP-2 P615 is significantly more flexible, noting that it is located right next to the calcium ion. However residues farer away like TSP-2 P607 and R608 seem to gain flexibility too. Results from this comparison are shown in figure 4.8. During all simulations, we further noticed, either by visual inspection or PCA that the EGF-like domains had a tendency to retract themselves to the *Stalk*, *Globe* ensemble.

The *Stalk* as already pointed out was found to have significant flexibility. We were however unable to link motions of the *Stalk* to the depletion of ions, as the *Stalk* underwent large conformational changes even without loosing a single ion. RMSF calculations found that during our simulations repeat 12N tends to be very flexible in the stalk. This is further interesting as this repeat holds the integrin binding motive in TSP-1 and TSP-2 (TSP-1 R926GD, TSP-2 R928GD). Detailed insights into flexibility are provided in figure 4.9.

The *Globe* seems to be very rigid. However two loops have been found to have significant flexibility:

1. TSP-1 R959-V979 and corresponding residues in TSP-2 N961-I981 and TSP-5 W1024-S1061: This loop endorses calcium ion 16b as shown in figure 3.6. Flexibility however could not be linked to the presence of this ion as this loop is also flexible in our simulations if the ion stays attached to this loop.
2. TSP-1 W1040-S1059 and corresponding residues in TPS-2 W1042-S1061 and TSP-5 W614-P633: This loop is interacting with the calcium ions 13 and 14 as indicated in figure 3.6 using the first two glutamate residues at the beginning of this loop which has the sequence WKQXXQ… Again how-

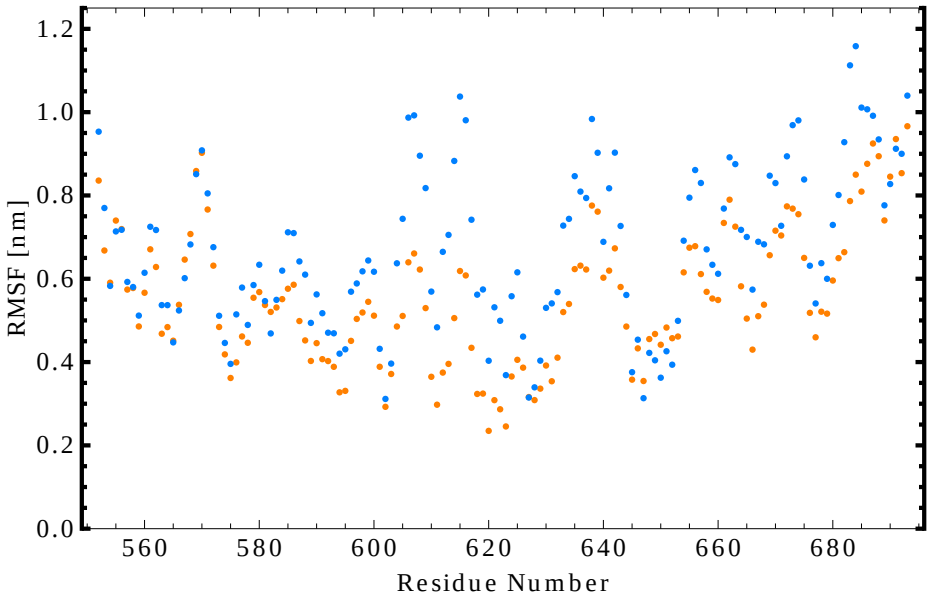

Figure 4.8: Comparison of RMSF fluctuations in the region of the EGF- like repeats: The mean of simulations 31, 32, 33, 34 and 45 where the ion stayed bound to the protein is shown in blue, while the mean of simulations 43, 44 and 46 in which the ion lefts its binding site is shown in orange. Simulation details can be found in tables 4.1 and 4.2.

ever, we were unable to link the flexibility of the loop to the departure of ions as the loop appears flexible in our simulations even with the ions attached. Further it is remarkable that this loop lies just next to a proposed binding site of CD-47 [24], one can thus speculate about an influence of the glutamic acid- calcium interaction on CD-47 binding.

Besides these two loops the helix at the C-terminus of TSP-5 shows extraordinary flexibility. This helix is not present in the other TSPs and at time of writing one can only speculate about its use. Further insights into the flexibility of the *Globe* are shown in figure 4.9.

Solvent Surface Accessibility of Binding Sites in the Thrombospondin Signature Domain

Three important binding sites have been identified in the SD. The SD houses the TSP-1 R926GD sequence which is an integrin binding motive [68]. This motive further exists in TSP-2 at the same position, *Stalk* repeat 12N, TSP-2 R928GD [69]. In TSP-5 however this pattern has moved to repeat 5N, TSP-5 R376GD [71]. Further the sequences proposed to be CD-47 binding are situated in this region: TSP-1 R1034FYVVMWK and I1121RVVM [24]. Using the *do_dssp* utility shipped with *GROMACS* and the trajectories of our simulations we investigated whether the integrin binding motive RGD in TSP-1 and TSP-2 as well the proposed CD-47 binding sites in TSP-1 are surface accessible. For the RGD site we found that all three residues are at least partially solvent accessible during the simulations. This accessibility does not relate to the loss of ion number 3 (c.f. figure 3.6), which is held in the RGD binding site and hence the ion site gets solvent accessible without losing the ion in repeat 12N. The surface accessibility for this site during simulation 16 as indicated in table 4.1 is shown in figure 4.10. We can only make speculations regarding the binding site to CD-47 as our results do not indicate clear solvent accessibility on the timescale of our simulations. Some minimal accessibility for the site TSP-1 I1121RVVM and no accessibility for the site TSP-1 R1034FYVVMWK was measured besides for the large arginine amino acid. Examples of such measurements are shown for the site TSP-1 I1121RVVM in figure 4.11 and for the site TSP-1 R1034FYVVMWK in figure 4.12.

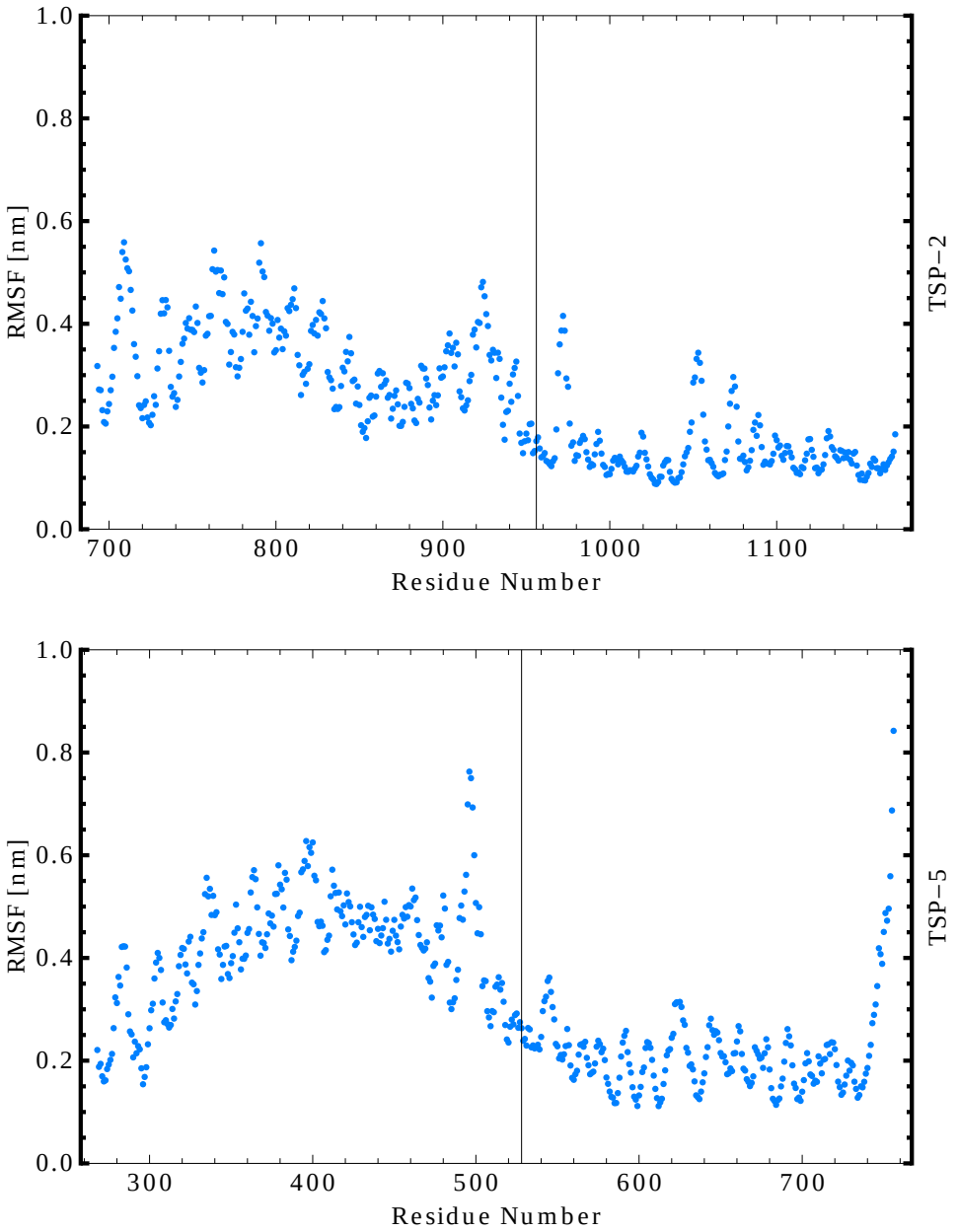

Figure 4.9: RMSF Calculations of the *Stalk, Globe* ensemble. The top figure is made from the mean of simulations 31 to 34 and 43 to 46 while the bottom figure is made from the mean of simulations 47 to 58 as indicated in the tables 4.1 and 4.2. The boundary between the *Stalk* (left) and the *Globe* (right) is indicated by a vertical line.

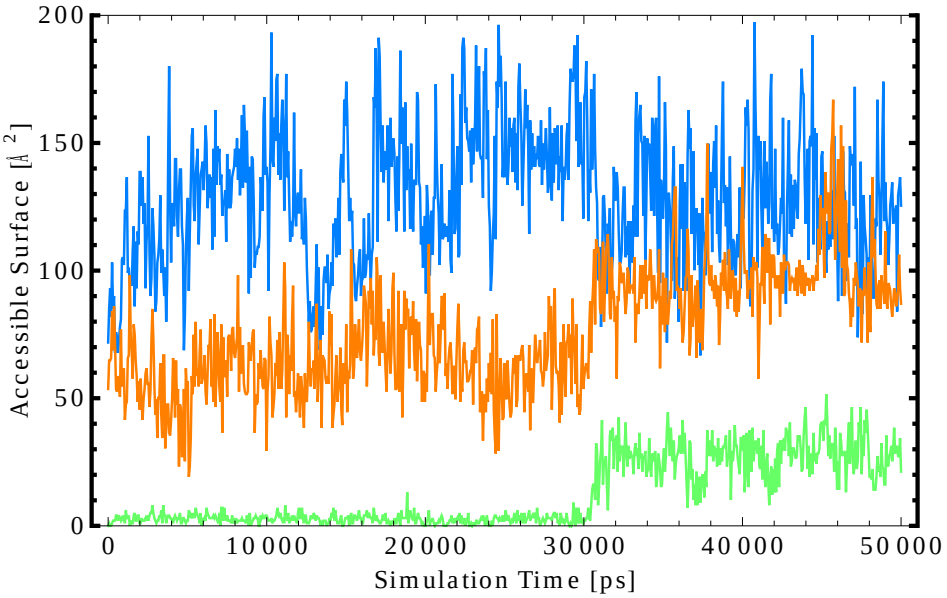

Figure 4.10: Surface accessibility of the proposed integrin binding site TSP-1 R926GD measured during simulation 16 as referenced in table 4.1: Accessibility is colored for the corresponding residue codes in the following manner: RGD

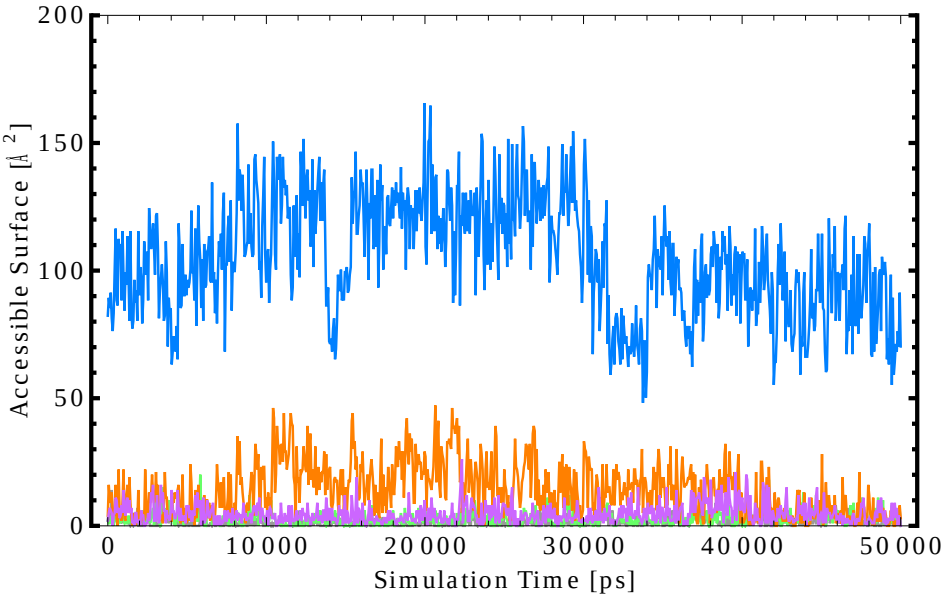

Figure 4.11: Surface accessibility of the proposed CD-47 binding site TSP-1 I1121RVVM measured during simulation 17 as referenced in table 4.1: Accessibility is colored for the corresponding residue codes in the following manner: IRVVM

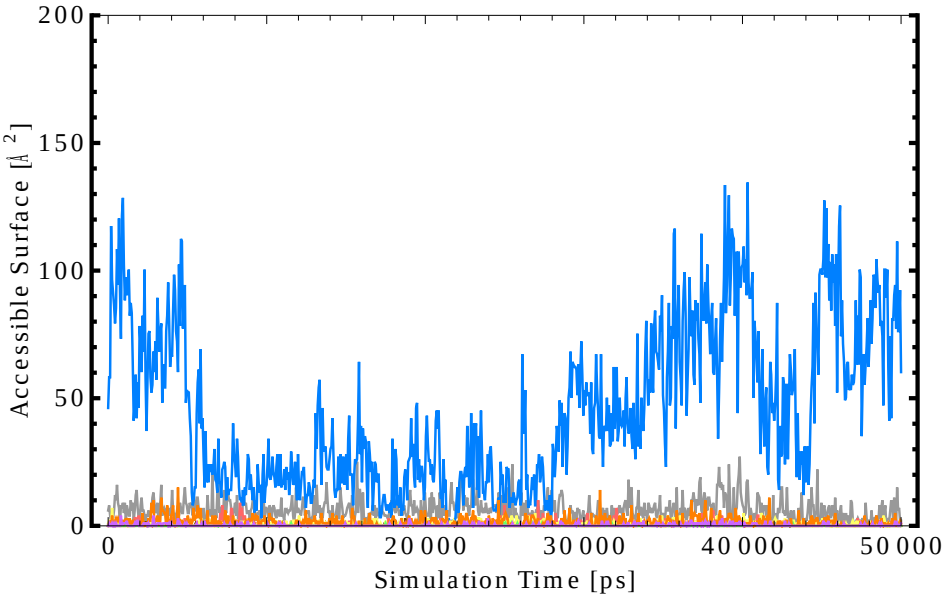

Figure 4.12: Surface accessibility of the proposed CD-47 binding site TSP-1 R1034FYVVMWK measured during simulation 17 as referenced in table 4.1: Accessibility is colored for the corresponding residue codes in the following manner: RFYVVMWK

4.9 Discussion on the Dynamics of Thrombospondin

We identified several features in the TSP SD from the MD simulations that we have made. We shall herein bring them into to the biological context of this protein domain.

Insights into Binding Sites

We start with the binding sites in this region. J. Lawler et al. have demonstrated in [25] that TSP does bind calcium selectively to αvβ3 integrin. In several simulations made, we found that this binding site is exposed in the calcium replete form during our simulations. One of these results is highlighted in 4.10. Our *in silico* model is thus coherent with this statement. The binding site to integrin in TSP-5 which lies on *Stalk* repeat 12N has not been investigated as it is thought to be already exposed in the structure to be found in the PDB by the code 3FBY and the binding site is extensively discussed in the article referring to this structure [71]. Regarding binding sites to CD-47 in TSP-1 the picture is however different. Binding sites to CD-47 have been discovered using peptide studies [24], and seem obscure knowing that they are buried in the crystallographic structure of TSP-1 that is referenced by the PDB code 1UX6. We further like to note that this structure is a C992S/N1067K double mutant, and might undergo disulphide bond isomerization in the wild type structure with effects on the structure and accessibility to CD-47 binding sites that one can only speculate about. In the MD simulations that we made we found that the binding site TSP-1 R1034FYVVMWK is completely buried, and that the binding site TSP-1 I1121RVVM is almost completely buried, not taking the arginine residue into account which is due to its large size solvent accessible. From the perspective of our MD simulations where these sites stay buried, we thus can not confirm these binding sites.

Insights into Structural Flexibility

After our statements on binding sites, we continue discussing structural flexibility, where we begin with the EGF like repeats, head over to the *Stalk* and finish with the *Globe*. RMSF cal-

culations have indicated that parts of the EGF like repeats get more flexible if the ion situated between the first and second EGF like repeat gets hydrated, which according to results shown in figures 4.5 and 4.6 is possible. The corresponding results are shown in figure 4.8. By visual trajectory inspection of the first and last frames of our MD simulations, as well as using PCA, we found that the EGF like repeats systematically tend to retract themselves to the *Globe*, *Stalk* ensemble. We can only speculate whether this happens *in vivo*, but are suggesting that such a movement should be taken into account in TSP models. Regarding the *Stalk* we find that it is overall very flexible in our simulations, highlighting flexibility in repeat 12N where the RGD binding site to integrins is located. We were however unable to correlate the flexibility of the *Stalk* to the depletion of the ions. In our simulations we found that the *Stalk* seems very flexible even without loosing any ions. This might however also be due to wrong parameters in the force field. We would like to remind that in the GROMOS force field with a modified 53a6 parametrization set [82] only the interaction between the OM and CA2+ atom types has been modified, and that interactions between atom types representing four instance backbone oxygens or other interaction partners with the calcium ion should be reparametrized as well. We think that this misparametrization could have caused additional flexibility in simulations with the GROMOS force field. The *Globe* according to our simulations is the most stable entity of the SD, although it highlights flexible loops. In regard to CD-47 binding this might be interesting as the loop TSP-1 W1040-S1059 lies directly next to the CD-47 binding site R1034FYVVMWK. This loop was further found to be interacting with the calcium ions found on the *Globe* using its glutamic acid residues. The flexibilities of the *Stalk* and *Globe* are highlighted in figure 4.9 made from RMSF calculations.

Identification of Exchangeable Ions

Having discussed the flexibility of the TSP SD, we continue in the identification of exchangeable ions. T. M. Misenheimer et al stated in [64] that a recombinant protein created from the TSP-2 SD sequence bound 22-27 calcium ions of which only ten are readily exchangeable. Using the results we obtained we can hint the location of these ions. Interestingly using the GROMOS force

field with the modifications proposed in [82] we found around 9 ions in our simulations that at some point are leaving their ion site. The probabilities of leaving a binding site in GROMOS force fields is highlighted in figure 4.5 and 4.6. In visually investigating simulation 8, which was made at high calcium ion concentrations as indicated in table 4.1, we have seen that calcium ions are exchanged at binding sites during such a simulation. Unsurprisingly the ions that have a certain probability to leave their binding site are located at locations where they act either as a third ion on the intersection of two *Stalk* repeats or where they are not available in an the crystal structure of an other TSP family member. Detailed calcium ion locations are highlighted in 3.6. Ion 16c for instance is located as a second ion in repeat 5N in the TSP-5 structure referenced by the PDB code 3FBY. This ion is found to leave this binding site in several MD simulations, and is not available in the crystal structure of TSP-2 referenced by PDB code 1YO8. The same can be stated for ion 16a which is only available in the structure of TSP-1 to be found using PDB code 1UX6. This ion is located in a loop attached to the *Globe*. Noting that the authors of these structures stated good reasons for finding the ions at these locations such as the different composition of the loop where calcium ion 16a is to be found between TSP-1 and TSP-2 [68] [69], we like to point out crystallization of these structures happened at calcium concentrations of up to 5mmol [68]. In the simulations that were used to create the probability graphs calcium concentrations listed in tables 4.1 and 4.2, are only due to initially bond calcium ions and their are no calcium ions in solution at the beginning of the simulation. Besides these ions the ion between the first and the second EGF like repeat also appears exchangeable in our simulations. Having identified these ions, we hope that the location of these might help further in the creation of dynamic models of the TSP SD.

Calcium Deplete Structures and interactions between EGF-like repeats, the *Stalk* and the *Globe*

D. S. Annis et al. have hypothesized two structures of the TSP SD from their results by immunochemical analysis [32]. In one of the proposed models the interactions between *Stalk* repeat 8N and 9C are maintained while in a second proposed model these interactions are lost. For both models the authors estimate an

elongation of the SD of 9nm. In order to review this hypothetical models we investigated the interaction between *Stalk* repeat 8N an the *Globe*. We found that according to our simulations there is a differentiation between the different members of the TSP family, and that effectively the 8N repeat from TSP-1 is more loosely bound to the *Globe* then the 8N repeat from TSP-2. We thus propose that in future models such a differentiation between the TSP family members should be included.

In the models proposed by D. S. Annis et al. [32] one further finds the interaction between the EGF-like repeats and the 12N, 13C and 1C are maintained. We found in all our simulations of the TSP-2 structure referenced by the PDB code 1YO8, that at least some interaction between the 12N *stalk* repeat and the third EGF like repeat remained maintained through a cation-π interaction formed by residues TSP-2 F671 and R928. For simulations made using the TSP-5 structure referenced by it's PDB code 3FBY we did not find any conserved interaction remarking that the EGF-like domain is placed in a different manner, which might be due to trimerization in the crystal structure.

4.10 Discussion about Classical Force Fields

In the case of the TSP, a careful reader will have several questions about what was presented in the results section of this chapter. Simulating the TSP SD has pushed everyday utilities on the edge of what is feasible. Nevertheless, we were able to obtain interesting results regarding the SD. We shall thus in this section discuss the limits of classical simulation and the results that we obtained regarding the TSP SD.

Limits of Classical Molecular Dynamics Simulation

In order to better understand classical MD one has to understand that classical MD is a model to resolve a problem, and that this model is far from perfect. Classical MD makes several assumption that were discussed in this chapter. Further force fields are parametrized to recreate empirical results. The modified 53a6 parametrization of the GROMOS force field was for instance obtained by statistical values [82] and the van der Waals force has been modified in order to recreate the statistical probability that a calcium ion gets hydrated in being released

from calmodulin. The van der Waals force, the coulomb interaction or bonded interactions in a force field hence do not have a forcibly physical meaning. They are parametrized to recreate experiences. In the TSP SD however we face a completely unknown object. The *Stalk* of the TSP SD has very unique structure, with around 25 calcium ions in it. Accountability of classical force fields on an object like this is thus questionable. Even worse we have shown that simple results regarding calcium ions like the free energy difference, as shown in section 4.8, between the bound state and unbound state of a calcium ion with a chlorine ion in the force fields used here are not reproducible between them. A further major drawback is that probably due to the high charge of a divalent calcium ion the charges on the surface of the protein should change. This process is called polarization and is not taken into account by the model of classical MD simulation chosen here. The problems described in here are further inherent in many MD simulations. Nevertheless, MD simulations have shown interesting results, and can help to think about and to better understand processes described by biologists in their experiences. We hence continued discussing our results, opinions that we derived from our MD simulations of the TSP SD. To overcome the obstacle of polarization, or at least to measure its impact we turned to molecular quantum mechanics, which shell be described in the following chapter.

5 | Molecular Quantum Mechanics

5.1 Introduction

As discussed in section 4.10 classical *molecular dynamics* (MD) suffers of significant drawbacks in performing simulations of systems such as the *Thrombospondin* (TSP) *Signature Domain* (SD). The large number of divalent calcium ions and the inherent effect of polarization are ill defined in terms of classical molecular simulations. Hence, we turned to molecular *quantum mechanics* (QM). Molecular QM has the ability to calculate the probability of finding an electron in a certain region of space. In effect we are descending from atomic or coarse grained simulations into the space of subatomic methods, where the electron shell of the atoms is explicitly calculated. One of the main ideas in doing this was to measure the effects of polarizability divalent ions are introducing on our TSP system, and to establish a first measure of errors that do happen in our classical MD simulations that are due to effects such as polarization or charge transfer. In order to do this, we used either the *Self Consistent Field* (SCF) also known as *Hartree Fock* (HF) method or so called hybrid methods that are coupling this method with *Density Functional Theory* (DFT). These are numerical methods that solve the time independent Schrödinger equation under approximations, and allow to derive a probability density that gives information about the probable locations of electrons in space, which we further may call the electron cloud. As the classical MD simulation approach does not know about electrons, and just attributes charges to objects such as atoms or beads, used in coarse grained simulations, we had to relate the results obtained from molecular QM to those obtained on the scale of classical MD. In order to do so,

we searched for an optimal way to attribute monopolar partial charges onto the particles, atoms or beads from coarse grained simulations, used in classical MD force fields. Using the methods at hand that allow to calculate so called Mulliken, or *Electrostatic Potential* (ESP) charges, we utterly failed to obtain meaningful results for problems of the size of the repeat from the *Stalk* of the TSP SD. To overcome this obstacle, we implemented our own algorithm to derive such partial charges. In this process we used new technologies and parallel programming techniques. The OpenCL language allowed us to parallelize our algorithm, and to implement it on the hardware to be found in contemporary compute devices such as multicore *Central Processing Unit*s (CPU)s, *Graphics Processing Unit*s (GPU)s or other accelerator boards and hence to leverage the compute power necessary to run this algorithm efficiently. Running it we obtained preliminary results on one *Stalk* repeat from the TSP SD, gained several insights and made several proposals on the derivation of partial charges.

5.2 Quantum Mechanics and Molecules

In 1900 Max Planck kicked of modern physics stating in his works on black body radiation [109] that energy is quantized. 26 years later motivated by the Einstein's photo effect, the hypothesis of de Broglie that particles behave like waves, and Debye's remark that if particles behave like waves they shall be governed by a wave like equation, Schrödinger published the equation named after him [110] [111]:

$$i\hbar\frac{\partial}{\partial t}\psi(r,t) = \left(\frac{-\hbar^2\nabla^2}{2m}\psi(r,t) + V(r,t)\right)\psi(r,t), \tag{5.1}$$

where r is a position in space, t shall be the time, $\hbar$ is the reduced Planck constant. ψ the so called wave function, m the mass of the wavelike particle to be described. In the stationary case, this equation reduces to what is called the time independent Schrödinger equation:

$$-\frac{\hbar^2\nabla^2}{2m}\psi(r) + V(r)\psi(r) = E\psi(r), \tag{5.2}$$

where E is the total energy of the particle. We shall note here that the Schödinger equation can be seen as an extension to clas-

sical mechanics as it can be shown that it is related to the Hamilton Jacobi equation in stating that action shall be quantized. In equation (5.2) using the normalization:

$$\int_{-\infty}^{\infty} |\psi(r)|^2 d^3r = 1, \tag{5.3}$$

$|\psi(r)|^2$ was identified to be the probability distribution of finding a particle in the field described by the potential $V(r)$. Further, in equation (5.2) $-\frac{\hbar^2\nabla^2}{2m}\psi(r)$ is identified to be the kinetic term and $V(r)\psi(r)$ to be the potential term. For a general system having an arbitrary number of particles, the Schrödinger equation is written like:

$$H\psi = E\psi, \tag{5.4}$$

where H is the Hamilton operator which describes the total energy of the system, an analogue to the Hamilton function from classical mechanics. We shall in the following use Dirac's Bra-Ket notation, where the inner product for eigenfunctions of a quantum mechanical and thus selfadjoint operator in a Hilbert space becomes:

$$\langle\psi|\psi\rangle = \int_{-\infty}^{\infty} \psi^\dagger\psi d^3r, \tag{5.5}$$

where $\psi^\dagger$ is the complex conjugate of ψ and from equation (5.3) it follows that:

$$\langle\psi|\psi\rangle = 1, \tag{5.6}$$

and further since $|\psi|^2$ was stated to be a probability density that the expected value for physical observable A is:

$$\langle A\rangle = \int_{-\infty}^{\infty} \psi^\dagger A\psi = \langle\psi|A|\psi\rangle\,, \tag{5.7}$$

from which we symbolically introduce the correspondence that A can be viewed as a matrix and $|\psi\rangle$ as a vector further noted as *ket*, where $\langle\psi|$ is the vector in the dualspace and shall be noted as *bra*. The time independent Schrödinger hence becomes:

$$H\,|\psi\rangle = E\,|\psi\rangle\,, \tag{5.8}$$

which can be seen as an eigenvalue problem where $|\psi\rangle$ are the eigenfunctions to the corresponding eigenvalue E. The probability to find an electron at a given point in space has however

analytically been solved from the Schrödinger equation only for the most basic atom, the hydrogen atom. In order to obtain the probability of finding an electron in a certain region in space around molecules, in our case around biological molecules, the repeats of the *Stalk* of the TSP SD, advanced numerical and approximative methods are needed in order to solve 5.4. We shall sketch the methods namely SCF, DFT and hybrid methods such as *Becke, three-parameter, Lee-Yang-Parr* B3LYP for completeness in the following text.

Hartree Fock, Self Consistent Field Method

The Self Consistent Field Method is an iterative method that from a guess solution finds a more accurate solution. As stated in our problem we would like to find the probability density for finding an electron in a certain region of space around a certain configuration of atoms. The derivation sketched here has been drawn from the derivations presented in the lecture *Quantentheorie II* held by Prof. Burgdörfer at the Vienna University of Technology in 2004 and from more or less detailed derivations with different aspects can be found in [112] [113] [114]. For the beginning, let us treat the electronic shell of a single atom. We shall then later on generalize this to molecular structures.

We write the Hamilton operator for such a system in the form:

$$H = \sum_{i=1}^{N} H_0 + \frac{1}{2} \sum_{i \neq j} V(i,j), \tag{5.9}$$

where H_0 is the single particle Hamilton operator and $V(i,j)$ the two particle interaction operator. For instance:

$$H_0(i) = -\frac{\hbar^2 \nabla^2}{2m} + \frac{Ze^2}{|r_n - r_i|}, \tag{5.10}$$

where the first term is the kinetic energy term for the electron, with m being its mass. The second describes the coulomb interaction between the electron and the nucleus, with Z being the number of protons in the nucleus, e being the elementary charge and $|r_n - r_i|$ being the distance between the nucleus and electron i. $V(i,j)$ describes the particle interaction, as for this example the electron- electron interaction:

$$V(i,j) = \frac{e^2}{|r_i - r_j|}. \tag{5.11}$$

Having now a Hamilton operator for electrons at hand, we still need to think about how we can write down a wavefunction of identical particles. In the following we take the Pauli principle for granted, stating that electrons, like all fermions, of the same spin can not be in the same energetic state. A multi particle state for fermions can be described, because of the limited permutations available and the idea that operators that cause permutations shall commute, by the Slater determinant:

$$|\Psi\rangle = \frac{1}{\sqrt{N}} \begin{vmatrix} \psi_1(1) & \psi_1(2) & \cdots & \psi_1(N) \\ \psi_2(1) & \psi_2(2) & \cdots & \psi_2(N) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_N(1) & \psi_N(2) & \cdots & \psi_N(N) \end{vmatrix}, \tag{5.12}$$

as such a wavefunction is completely antisymmetric. For a derivation of this determinant the reader shall be referred to [93] or [112]. Using this Slater determinant we can now for a somewhat arbitrary wavefunction together with equation (5.7) define the energy functional:

$$E[\Psi] = \langle H \rangle = \langle \Psi | H | \Psi \rangle \,. \tag{5.13}$$

The idea behind the HF method is right now to solve the variational problem under constraints, using Lagrange multipliers. The constraints here are the orthonormality of the one particle wavefunctions. Hence the variational problem is written down like:

$$\frac{\delta}{\delta \langle \psi_i |} \left(E[\Psi] - \sum_k \langle \psi_k | \psi_k \rangle \lambda_k \right) = 0, \tag{5.14}$$

which in physical terms means that we want the energy to be stationary for variations of the single particle wavefunction ψ_i under the constraint that all single particle wavefunctions stay orthonormal. In order to solve equation (5.14) we first need to consider the energy functional presented in equation (5.13). Since we solve our problem for electrons, which are stated to be fermions, which follow the Pauli principle we can say that there exists a unique attribution between the particle index and the state index as everyone of the most lowest states is exactly once occupied. Hence, one can replace the summation over particles by a summation over states. For H_0 from the multiparticle Hamil-

ton operator in equation (5.9), it follows:

$$\langle\Psi|H_0|\Psi\rangle = \sum_{j=1}^{N} \langle\psi_j(1)|H_0(1)|\psi_j(1)\rangle\,, \tag{5.15}$$

where the particle number (1) in this case is completely arbitrary and chosen by us. In an analogue approach we can write for the interaction term with only two particle indices. $V(i,j)$:

$$\begin{aligned}\langle\Psi|V(i,j)|\Psi\rangle &= \frac{1}{2}\sum_{i\neq j}(\langle\psi_i(1)\psi_j(2)|V(1,2)|\psi_i(1)\psi_j(2)\rangle \\ &- \langle\psi_i(1)\psi_j(2)|V(1,2)|\psi_j(1)\psi_i(2)\rangle).\end{aligned} \tag{5.16}$$

From equation (5.15) and (5.16) we have obtained the energy functional written in equation (5.13) for our multielectron problem:

$$\begin{aligned}E[\Psi] &= \sum_k \langle\psi_k(1)|H(1)|\psi_k(1)\rangle \\ &+ \frac{1}{2}\sum_{i\neq j}(\langle\psi_i(1)\psi_j(2)|V(1,2)|\psi_i(1)\psi_j(2)\rangle \\ &- \langle\psi_i(1)\psi_j(2)|V(1,2)|\psi_j(1)\psi_i(2)\rangle).\end{aligned} \tag{5.17}$$

Now we can concentrate ourselves on solving the variational problem stated in equation (5.14). Using the formulas:

$$\frac{\delta}{\delta\langle\psi_i|}\langle\psi_k| = \delta_{ik}, \tag{5.18}$$

and

$$\sum_i a_i\delta_{ik} = a_k, \tag{5.19}$$

we solve for the second term in (5.14):

$$\frac{\delta}{\delta\langle\psi_i|}\left(\sum_k \langle\psi_k|\psi_k\rangle\,\lambda_k\right) = |\psi_i\rangle\,\lambda_i. \tag{5.20}$$

We continue with the first term of equation (5.14), the energy functional. We solve for the first term from equation (5.17):

$$\frac{\delta}{\delta\langle\psi_i|}\left(\sum_k \langle\psi_k(1)|H(1)|\psi_k(2)\rangle\right) = H(1)\,|\psi_i(1)\rangle\,, \tag{5.21}$$

for the second:

$$\begin{aligned}
&\frac{\delta}{\delta\langle\psi_i|}\left(\frac{1}{2}\sum_{i\neq j}\langle\psi_i(1)\psi_j(2)|V(1,2)|\psi_i(1)\psi_j(2)\rangle\right) = \\
= \quad &\frac{1}{2}\sum_{j,j\neq i}\langle\psi_j(2)|V(1,2)|\psi_j(2)\rangle\ |\psi_i(1)\rangle \\
&+\frac{1}{2}\sum_{k,k\neq i}\langle\psi_k(1)|V(1,2)|\psi_k(1)\rangle\ |\psi_i(2)\rangle\,, \qquad (5.22)
\end{aligned}$$

and the third:

$$\begin{aligned}
&\frac{\delta}{\delta\langle\psi_i|}\left(-\frac{1}{2}\sum_{k\neq j}\langle\psi_k(1)\psi_j(2)|V(1,2)|\psi_j(1)\psi_k(2)\rangle\right) = \\
= \quad &-\frac{1}{2}\sum_{j,j\neq i}\langle\psi_j(2)|V(1,2)|\psi_i(2)\rangle\ |\psi_j(1)\rangle \\
&-\frac{1}{2}\sum_{k,k\neq i}\langle\psi_k(1)|V(1,2)|\psi_i(1)\rangle\ |\psi_k(2)\rangle\,. \qquad (5.23)
\end{aligned}$$

Putting everything together and taking the symmetry into account stating that $V(1,2) = V(2,1)$, one obtains the HF equations:

$$\begin{aligned}
H_0\ |\psi_i(1)\rangle \quad &+ \quad \sum_{j=1}^{N}\langle\psi_j(2)|V(1,2)|\psi_j(2)\rangle\ |\psi_i(1)\rangle \\
&- \quad \sum_{j=1}^{N}\langle\psi_j(2)|V(1,2)|\psi_i(2)\rangle\ |\psi_j(1)\rangle = \lambda_i\ |\psi_i(1)\rangle\,. \qquad (5.24)
\end{aligned}$$

The second term in the HF equations is identified to be describing the coulomb force between the electrons, while the third term is, called the exchange term, can not be identified with anything one knows from classical physics. One further now defines the Fock potential in the n^{th} approximation :

$$U^n_{\text{HF}} = \sum_{j=1}^{N}\left\langle\psi^n_j(2)|V(1,2)|\psi^n_j(2)\right\rangle - \sum_{j=1}^{N}\left\langle\psi^n_j(2)|V(1,2)|\psi^n_i(2)\right\rangle. \qquad (5.25)$$

Hence, equation (5.24) can be resolved by iteration like:

$$(H_0(1) + U^{n-1}_i(1)\ |\psi^n_i\rangle = \lambda^n_i\ |\psi^n_i\rangle\,. \qquad (5.26)$$

λ_i turn out to be the so called HF energies. Solution can now be found in starting with guess wavefunctions $|\psi_i^0\rangle$ and iterating until convergence is reached. This way one can find approximate wave functions that return the probability to find an electron somewhere in the space around an atom. However, right now we still have not yet treated molecules. In order to do so we have to introduce the so called *Linear Combination of Atomic Orbitals* (LACO) Ansatz. In this method one supposes that the wavefunction of a molecule can be treated as a superposition of wavefunctions centered around an atom. Thus we shall write:

$$|\psi_m\rangle = \sum_{o=1}^{N_o} c_{om} \, |\chi_o\rangle \, . \tag{5.27}$$

Here ψ_m is to be the molecular wavefunction, χ_o the wavefunctions centered around the atoms, c_{om} the coefficients of describing our linear superposition, and N_o the number of such wavefunctions. For simplicity we shall introduce the operator:

$$F = H_0 + U, \tag{5.28}$$

and we can write down the equation:

$$F \sum_{o=1}^{N_o} c_{om} \, |\chi_o(1)\rangle = \lambda_m \sum_{o=1}^{N_o} c_{om} \, |\chi_o(1)\rangle \, . \tag{5.29}$$

Multiplying with $\langle\chi_l|$ we can further rewrite:

$$\sum_{o=1}^{N_o} c_{om} \, \langle\chi_l(1)|F|\chi_o(1)\rangle = \lambda_m \sum_{o=1}^{N_o} c_{om} \, \langle\chi_l(1)|\chi_o(1)\rangle \, , \tag{5.30}$$

which with further definition of the overlap matrix

$$S_{lm} = \langle\chi_l|\chi_m\rangle \, , \tag{5.31}$$

and the Fock matrix:

$$F_{lm} = \langle\chi_l|F|\chi_m\rangle \, , \tag{5.32}$$

can be written as the matrix equation:

$$Fc = \lambda S c, \tag{5.33}$$

which is called the Roothaan Hall equation [113] [115] [116]. This matrix equation is the basis of the HF-SCF method for molecules, and can be solved in an iterative way like the HF equation (5.24). We shall remind that even though (5.33) seems simple, that the Fock operator itself is dependent on the current set of orbitals and that every evaluation in an iterative process requires the calculation of several integrals in order to evaluate the matrix elements in (5.31) and (5.32). Having briefly sketched the basics around the HF-SCF method in order to obtain the localization of the electron cloud, the probability density in space to find an electron around a molecule, we shall introduce an other method that we have used:

Density Functional Theory and Hybrid Functionals

Density functional theory is based on the idea that the energy of a system of electrons can be written in terms of the electron probability density, and hence it exists a functional $E[\rho(r)]$ where in this case $\rho(r)$ shall be the probability density to find an electron at location r in space and hence, since we stated that the absolute squared of the wavefunction has this probability we note:

$$\rho(r) = \sum_{i=1}^{N} |\psi_i(r)|^2, \tag{5.34}$$

where ψ_i are our single particle wave functions. The existence of such a functional was described by Hohenberg and Kohn in 1964 [117]. The sketch of DFT we provide here is drawn from derivations in [113] [114]. Developing this energy functional a bit one can write it in terms like the following:

$$E[\rho(r)] = T[\rho(r)] + V_{\text{ee}}[\rho(r)] + V_{\text{ncl}}[\rho(r)]. \tag{5.35}$$

Here T is the kinetic energy, V_{ee} the potential energy due to electron, electron interaction, and V_{ncl} the potential describing the electron, nuclear interaction. The inherent problem with this equation is right now that the kinetic term can not be correctly represented in terms of a functional. One now defines new functional such as:

$$T_{\text{S}}[\rho(r)] = \sum_{i=1}^{N} \langle\psi_i| \frac{-\hbar^2\nabla^2}{2m} |\psi_i\rangle, \tag{5.36}$$

remembering that as previously stated $T_S[\rho(r)] \neq T[\rho(r)]$. One does the same for the coulomb interaction writing down:

$$J[\rho(r)] = \frac{e^2}{2} \int dr \int dr' \frac{\rho(r)\rho(r')}{|r - r'|}. \tag{5.37}$$

Again $J[\rho(r)] \neq V_{ee}[\rho(r)]$ which claims to be exact. In order to overcome those differences one introduces the so called exchange correlation functional that shall correct all the shortcomings:

$$E_{\mathrm{XC}}[\rho(r)] = (T[\rho(r)] - T_S[\rho(r)]) + (V_{\mathrm{ee}}[\rho(r)] - J[\rho(r)]). \tag{5.38}$$

Using these functionals equation (5.35) casts into:

$$E[\rho(r)] = T_{\mathrm{s}}[\rho(r)] + J[\rho(r)] + V_{\mathrm{ncl}}[\rho(r)] + E_{\mathrm{XC}}[\rho(r)]. \tag{5.39}$$

Applying the same variational approach that we have made in order to derive the HF equations, shown in equation (5.14), and the constraint that the wavefunctions shall form an orthonormal bases set, one can find the Kohn- Sham equations [118] (not shown):

$$\left(-\frac{\hbar^2\nabla^2}{2m} + V_{\mathrm{eff}}(r)\right) |\psi_i(r)\rangle = \epsilon_i \, |\psi_i(r)\rangle \,, \tag{5.40}$$

where the effective potential V_{eff} is found to be:

$$V_{\mathrm{eff}} = V_{\mathrm{ncl}} + \int \frac{e^2\rho(r')dr'}{|r - r'|} + \frac{\delta E_{\mathrm{XC}}(r)}{\delta\rho(r)}. \tag{5.41}$$

As stated, the derivation of the Kohn- Sham equations and DFT is in general analogue to the derivation of the HF method. The main difference is that the energy functional is written in more sophisticated terms. One can hence state that DFT is not a different method, but an extension to the HF method. The Kohn-Sham equations can further again, analogue to the HF method, solved iteratively. One starts with an initial set of wave functions ψ_i that are linked to $\rho(r)$ by equation (5.34) and calculates the V_{eff} which is used to obtain a new set of wave functions.

Until now we have nothing stated about the exchange correlation functional and until today its form remains mysterious. Several approaches to approximate this term do exist, and we like to state, that unsurprisingly, most of the errors in density functional theory arise from this exchange correlation term. We point out the following methods:

– *Local Density Approximation* (LDA): LDAs are the class of functionals that are only dependent on the electron density itself. In general this is written down like:

$$E_{\mathrm{XC}}^{\mathrm{LDA}}[\rho] = \int \gamma(\rho) dr, \tag{5.42}$$

where γ shall be a function depending on the electron density. A very simple form of a LDA type exchange correlation functional is for instance the Xα-method, [114]. In this method the functional is written like:

$$E_{\mathrm{XC}}^{\mathrm{LDA-X\alpha}}[\rho] = C \int \rho(r)^{4/3} dr, \tag{5.43}$$

with C being a constant. Many further LDA functionals do exist.

– *Generalized Gradient Approximation* (GGA): GGAs further depend on the gradient of the electron density and hence:

$$E_{\mathrm{XC}}^{\mathrm{GGA}}[\rho] = \int \gamma(\rho, \nabla\rho) dr, \tag{5.44}$$

where elaborated functions γ have been developed. Extensions of this functional include for example second derivatives, for such:

$$E_{\mathrm{XC}}^{\mathrm{GGA}}[\rho] = \int \gamma(\rho, \nabla\rho, \nabla^2\rho) dr. \tag{5.45}$$

– Hybrid functionals: Hybrid functionals use several different methods, further they may include the so called HF exchange functional, these are the types of functionals that we have used in our calculations in order to obtain the electron density around the molecules. We have used B3LYP [119] [120] [121] [122] and (*Becke, Half and Half, Lee-Yang-Parr* (BHandHLYP) [123] functionals. Here exchange and correlation are written as separate terms. This comes naturally, taking for example the HF equation that we derived. We stated that in (5.24), there exists a coulomb term and an exchange term. No correlation term exists in HF as it is a so called mean field method, and does not take instantaneous electron- electron interaction effects into account. Going back to the HF functional (5.17) we see that the existence

of the exchange term exists due to the antisymmetry imposed by the Slater determinant. For B3LYP one writes for instance:

$$\begin{aligned} E_{\mathrm{XC}}^{\mathrm{B3LYP}} &= E_{\mathrm{XC}}^{\mathrm{LDA}} + a_0(E_{\mathrm{X}}^{\mathrm{HF}} - E_{\mathrm{X}}^{\mathrm{LDA}}) \\ &+ a_{\mathrm{X}}(E_{\mathrm{X}}^{\mathrm{GGA}} - E_{\mathrm{X}}^{\mathrm{LDA}}) \\ &+ a_{\mathrm{C}}(E_{\mathrm{C}}^{\mathrm{GGA}} - E_{\mathrm{C}}^{\mathrm{LDA}}) \end{aligned} \tag{5.46}$$

without stating how LDA and GGA terms look like. a_0, a_{X} and a_{C} are constants chosen by the authors. For exact terms the reader is referred to the articles that assemble B3LYP [119] [120] [121] [122]. $E_{\mathrm{X}}^{\mathrm{HF}}$ here is the HF exchange term from equation (5.17).

Basis Functions [113]

During the calculations, the wavefunctions have been represented by *Contracted Gaussian Type Orbitals* (GTO)s. Approximation of atom centered wave functions by such GTOs is done as these functions are easy to integrate. A contracted GTO has the form:

$$|\chi(a,b,c)\rangle = N\sum_{i=1}^{n} c_i x^a y^b z^c e^{-\zeta_i r^2} \quad a,b,c \in \mathbb{N} \tag{5.47}$$

here N is the normalization constant and ζ controls the width of the orbital. In our calculations we used either the 6-31G* or 6-31G** basis set. Which means that one uses one contracted GTO with $n = 6$ in equation (5.47). One also speaks of six primitives for each orbital on an inner shell. A further GTO of $n = 3$, a single primitive called the diffuse primitive plus six d-type gaussians for each atom other than the hydrogen atom are also added in calculations where 6-31G* is used. In the case of 6-31G**, one adds to the above, three p-type gaussians for each hydrogen atom. p-type means that the sum of the indices is $a + b + c = 1$, while in the d-type the sum of the indices is $a + b + c = 2$. These functions, that approximate the wave functions, are then optimized by the methods described above. Several different other more or less complex basis functions for different types of calculations do exist, having here only introduced the ones we have used during our calculations. In this work we shall from now on denote a quantum mechanical calculation that results in an approximate

result of the electron density of a molecule by its method and its basis set. A typical calculation might thus be called HF/6-31G*.

Now that we have sketched the basics of the theory used our quest to derive the probability of finding an electron in space, we will in the continuing describe the calculations we have made and the programs we have created.

5.3 Obstacles in Deriving Monopolar Partial Charges

In order to obtain results that can be compared to classical MD force fields and to better understand the errors arising from polarization, we have sought for an effective way to parametrize partial charges on atoms or coarse grained beads that arise from MD force fields. We tried to do this using the ESP method implemented in *Gaussian* [124] [33]. To do so we prepared repeats 8N and 9C from the TSP SD. This was done in creating a *Protein Data Bank* (PDB) [50] file containing the atoms of these repeats, including water molecules and calcium ions that are found at these sites in the structure designated by its PDB code 1UX6. To obtain these small structures we have literally cut them out of the PDB file. Proper C- and N- termini have been added using the *pdb2gmx* utility from *GROMACS* [30]. The structures have then been energetically minimized using *Gaussian*, keeping the locations of backbone atoms frozen in order to prevent the peptide from refolding. The electron density calculations and minimizations have been performed in B3LYP/6-31** or BHandHLYP/6-31** theory. B3LYP was chosen due to its popularity while BHandHLYP was chosen as it was found to yield the best energy values for amino acids in a comparison of several different methods [125]. The ESP method [33] has then been tried in order to obtain partial charges from this structure. Not obtaining meaningful results and the problem that minimization often failed to converge after several days of computation, we tried to solve the problem in a new way that will be detailed later. Before that we shall talk about the implemented methods and the problems that arise in deriving partial charges from *ab inito* methods, hence by resolving the probability function to find an electron around a molecule.

Existing Methods to Derive Partial Charges

The popular methods to derive monopolar partial charges using *ab initio* calculations can be grouped in two methods:

1. Population Analysis: These type of methods put the charge of an object in direct relationship with the probability to find an electron around that object. Charges obtained from a method like this are colloquially known as Mulliken charges [34].
2. Electrostatic Potential Sampling or ESP methods: In this case the electrostatic potential is first established around the molecule from *ab inito* methods. Then this potential is sampled, and charges are attributed in order to recreate the *ab inito* potential. [33] [126] [127] [128] [129] [130] [131]

The first approach has to be found inaccurate in many situations as electrons are diffusing and allocation to atoms using only this technique is not clear, even though solutions have been suggested [132]. Further as we wanted to model a protein's interaction with its environment we focus on the second approach and may briefly sketch here how it is applied in general. Using *ab inito* methods such as the ones described in section 5.2 one obtains the electron density $\rho(r)$. One can using this density calculate a reference electrostatic potential:

$$\Gamma(r) = \int_R \frac{a\rho(r') + \sum_{i=1}^{\text{nAtoms}} q_i \delta(r' - x_i)}{|r - r'|} d^3r, \tag{5.48}$$

where a is a scale-factor that converts the electron density into a charge density. q_i is the nuclear charge of atom i at position x_i. R is a sufficient large volume to integrate the whole electron density in order that:

$$1 = \int_R \rho(r) dr. \tag{5.49}$$

Further one introduces the electrostatic potential that is generated in assigning partial charges to the objects, atoms or beads from coarse grained simulations:

$$\Phi(r) = \sum_{i=1}^{\text{nAtoms}} \frac{Q_i(x_i)}{r - x_i}. \tag{5.50}$$

Having both potentials at hand one defines a method to measure the difference between the two potentials:

$$F = \int_V D(\Gamma(r), \Phi(r)) d^3r, \tag{5.51}$$

where D is a measurement that describes the deviation of the potential obtained from *ab inito* methods and the potential obtained from the charges Q_i. V can in general be an arbitrary region in space. Monopolar charges are in the class of the ESP type methods, obtained in minimizing the function F by varying the charges Q_i. In most algorithms this is done using linear regression methods limiting the choice of the measurement function D.

Although the ESP approach seems to be straightforward, it suffers from significant drawbacks:

- Rotational dependence: In practice it was found that ESP charges are not independent on the rotation of a molecule. To elaborate this one has to take into account that one evaluates a finite number of testpoints and hence equation (5.51) has to be rewritten into:

$$F = \sum_{i=1}^{\text{nPoints}} D(\Gamma(p_i), \Phi(p_i)), \tag{5.52}$$

  where the testpoints are posed on a predefined geometry such as sphere, the Connolly surface or a homogeneous grid in space for instance. Rotating these testpoints around the molecule was found to yield different partial charges, and hence ESP charges are in general not rotation independent [129] [130].
- Conformational dependence: The electronic density $\rho(r)$ is dependent on the conformation of the molecule and consequently the charges, fit against an *ab inito* potential are not independent of the molecules conformation either. This poses significant problems if, for instance, one wants to obtain the monopolar charges in order to use them in a MD force field, where during a simulation the molecule might undergo several different conformations but the monopolar charges stay static.
- Shielding of buried charges: In buried charges one can observe the effect of shielding. To illustrate this imagine a

point charge that is surrounded by a homogeneous spherical charge around it. In this case an infinite number of solutions for the charges, the point charge and the charge surrounding the point charge, do exist if the potential outside both charges is all that is known. This problem is further augmented by a small number of testpoints. Methods do exist that try to tackle this problem by confining charges around zero during the minimization of F which nevertheless seems arbitrary [128].

Being confronted with these problems and finding ourselves unable to obtain meaningful partial charges, we developed our own algorithm in order to overcome, in increasing the number of sampling points, taking multiple rotations, conformations into account during the minimization procedure and using novel parallel programming techniques, the problems stated here.

5.4 The New Partial Charges Algorithm

The algorithm that we have developed is a direct minimization algorithm using the gradient descent approach. The algorithm is derived in the following way:

Gradient Descent under Constraints

Our algorithm works around the idea, like in other ESP methods, that the *ab initio* potential is already known. Such a potential can be obtained from *ab initio* software packages such as *Gaussian* [124], using the molecular QM methods that have been presented in section 5.2 of this chapter. We begin in writing down the constraints that we have to take into account in running the minimization algorithm. It is clear that we would like the charge to be conserved, this is written down like:

$$Q_{\text{tot}} = \sum_{i=1}^{\text{nAtoms}} Q_i. \tag{5.53}$$

Force fields further are parametrized with the idea in mind that there are several chemically equivalent atoms in a molecule, such as the hydrogen atoms of a methyl group. In order to derive partial charges that are comparable to the parameters of such force

fields, we rewrite the constraint from equation (5.53) into:

$$Q_{\text{tot}} = \sum_{i=1}^{k} c_i W_i \quad c \in \mathbb{N}^*, \tag{5.54}$$

where k is the number of different types of atoms representing the molecule, c_i the weight of elements which are chemically equivalent and W_i the charge of type i. One can identify that (5.54) defines a hyperplane in W_i-charge space. The normal on this hyperplane which we will need later on, can be found to be:

$$n = \nabla_{W_i} \left(Q_{tot} - \sum_{i=1}^{k} c_i W_i \right) = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix}. \tag{5.55}$$

Further, we shall express F from equation (5.51) as a function dependent on the charges W:

$$F(W) = \sum_{i=1}^{\text{nPoints}} D(\Gamma(p_i), \Phi(W, p_i)), \tag{5.56}$$

where we make sure that coordinates of testpoints p_i do not fall beneath the van der Waals radius of any atom of our molecule. In the numerical implementation this is done by suppressing such grid points. The minimization of F is done in the form of an iteration by changing the charges and hence:

$$W_i^N = W_i^{N-1} + \gamma^N dW_i^N, \tag{5.57}$$

where γ is the stepsize in the descent direction dW in charge space. In order for dW to be a valid descent direction, we consider

$$dW = \frac{\nabla_W F}{|\nabla_W F|} - \frac{\langle n | \nabla_W F \rangle}{|n|^2 |\nabla_W F|} n, \tag{5.58}$$

to be the gradient by W of F projected onto the forgoing plane (5.54) using the normal defined in (5.55). For as long as

$$\frac{\langle dW | \nabla_W F \rangle}{|dW||\nabla_W F|} \geq 0.5, \tag{5.59}$$

which is satisfied as dW is the projection into a plane, there exists a γ so that $F(W^N) < F(W^{N-1})$.

Now one still has to care about the stepsize γ whose optimization is a so called linesearch problem. From a numerical side speaking we further wanted our algorithm to run on various types of hardware and to run either in single (32 bit) or double (64 bit) floating pointing precision, further details are explained in section 5.5. In order to do so, we avoided second derivatives and hence our linesearch algorithm is very simplistic. We consider γ to be optimal if

$$\frac{\partial}{\partial\gamma}F(W^N + \gamma dW^N) = 0. \tag{5.60}$$

We evaluate the left side of (5.60) for $\gamma = n\gamma^{N-1}$ beginning with $n = 0$ and increasing n by 1 until the left side of (5.60) is greater then 0. Afterwards we refine the optimal stepsize by at least 5 bifurcation steps on the interval $\gamma \in ((n-1)\gamma^{N-1}, n\gamma^{N-1}]$. The lower boundary of the last bifurcation interval is chosen in order to avoid to large stepsizes. Bifurcation is continued until the lower boundary is greater than $(n-1)\gamma^{N-1}$ in order to avoid a lockup of the algorithm.

Measurement Function

One of the key advantages of the gradient descent approach is that we can use alternative measurement functions D. D is used to calculate the distance of the potential derived by *ab-initio* methods Γ, from the potential Φ created by the charges attributed to the particles. In the numerical implementation four different measurements in the algorithm do exist, and we would like to point out that further ones can be implemented using minimal effort. The measurements implemented are:

1. Least absolute deviations (L_1):

$$D = |\Gamma(p_i) - \Phi(p_i)|. \tag{5.61}$$

2. Least square deviations (L_2):

$$D = (\Gamma(p_i) - \Phi(p_i))^2. \tag{5.62}$$

3. Weighted least absolute deviations:

$$D = \frac{|\Gamma(p_i) - \Phi(p_i)|}{|\Gamma(p_i)| + \epsilon}. \tag{5.63}$$

4. Weighted least square deviations:

$$D = \frac{(\Gamma(p_i) - \Phi(p_i))^2}{\Gamma(p_i)^2 + \epsilon}. \tag{5.64}$$

where ϵ in equations (5.63) and (5.64) is to be a small number, in order to avoid divisions by zero. Measurements (5.63) and (5.64) have been implemented to augment the potential values far away from the molecule which should give them more weight in the sum of equation (5.56).

5.5 Numerical Implementation of the New Algorithm

The algorithm presented in the previous section 5.4 has been successfully implemented using parallel programming techniques. This is necessary as this type of algorithm imposes a significant overhead against traditional ESP methods due to its direct minimization procedure. The OpenCL *Application Programming Interface* API has been chosen due to its versatility to adapt to different hardware such as multicore CPUs, GPUs and other types of accelerator boards that are common in contemporary computers and HPC centers. One can see by investigating the algorithm presented in section 5.4 in detail, remarking that the *ab initio* potential Γ presented in equation (5.48) is already known, that the calculation of the potential Φ found in equation (5.50) and the difference F from equation (5.56) depending on such a Φ is the most resource consuming part of the algorithm. Hence we parallelized these parts of the calculation. We shall note that F for different Φ has to be evaluated several times in each step which render this parallelization further necessary. For instance, let us look at the gradient $\nabla_W F$ as found in equation (5.58). In order to evaluate this gradient with the numerical derivative:

$$f'(x) = \frac{f(x + \Delta) - f(x - \Delta)}{2\Delta}, \tag{5.65}$$

the potential Φ has to be evaluated two times the number of charges W_i often.

Our algorithm is based around the *Gaussian* [124] cube file, consequently it defines the grid that used for the calculation. In

order to obtain such file one can use the *cubegen* utility that is shipped with the *Gaussian* software package. The values in the gridpoints of the *ab initio* potential are stored in these files. For details on *Gaussian* cube files, the reader shall be referred the manuals that are shipped with the software. Our program allows for arbitrary types of grids as long as the axes are orthogonal. This should allow for great flexibility in creating the grids of testpoints. We further allow for multiple cubes in one calculation. One can for example generate cubes with different rotations of the molecule the partial charges are attributed for, and tackle the problem of rotational dependence this way. An other approach could be to take different conformations in different cubes into account, stating that Boltzmann weighting of cubes is not implemented in the algorithm but one should be able to implement it with minimal effort. This way one can tackle for instance the problem of conformational dependence of the charges, in minimizing grids with different conformations of the molecules in them at once.

The parallelization is done in the way that on an OpenCL compute device having multiple processing units (such as multiple cores on a CPU) the potential $\Phi(p_i)$ and its difference according to one of the measures D found in equations (5.61), (5.62), (5.63) and (5.64) are calculated in parallel. After this calculation a reduction step, hence calculating the sum in equation (5.56) is necessary. This reduction step is further parallelized, as the sum is split into partial sums. On a grid built of n_x by n_y by n_z grid points in directions x,y and z the sum is split into n_x by n_y partial sums summing over n_z values. The final sum of the partial sums is then performed on by the host program that controls the compute devices. If multiple cubes are used in one simulation, the calculation can further be split among multiple OpenCL compute devices in distributing the grids stored in the *Gaussian* cube files among those devices. In this case again the partial sums from all the grids on the different OpenCL compute devices are collected by the host program controlling them. The host program then performs the final reduction step in summing the results from all partial sums. The distribution of multiple grids on multiple OpenCL devices is highlighted in figure 5.1.

This reference implementation programmed in C using the OpenCL API is implemented in three files named *multicube.c*, *spot.cl* and *dpot.cl*, where *spot.cl* and *dpot.cl* contain either the

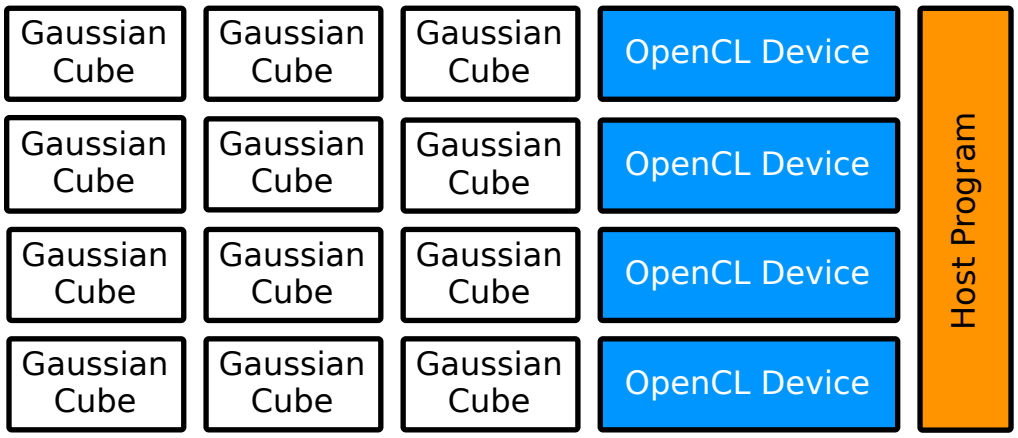

Figure 5.1: Distribution of grids defined by *Gaussian* [124] cube files across multiple OpenCL devices

single (32bit floating point) or double (64bit floating point) precision code to be run on the OpenCL device. The implementation can be compiled on Linux by the following command:

```
gcc -O2 multicube.c -lOpenCL -L/path/to/opencl -DD_DOUBLE
```

Listing 5.1: Compiling the *multicube* program on a Linux system.

where `/path/to/opencl` is the path to the OpenCL library on the system. The `-DD_DOUBLE` parameter creates a binary that runs the algorithm in double (64bit floating point) precision. The program can further be compiled on Mac OS X 10.6 or later using the command

```
gcc -O2 multicube.c -framework OpenCL -DD_DOUBLE
```

Listing 5.2: Compiling the *multicube* program on a Mac OS X system.

The *multicube* program then needs several input parameters. A file defining the van der Waals radii is needed, as well as a file containing the initial charges in order to start the minimization. These files are relatively simple. A three letter code in both files defines the atom type, the parameters are stated on the same line next to the three letter code. Even though chemical equivalence might exist between the types of atoms, a line has to be stated for each atom in the files defining the charges. This is due to the idea that one might take the charges from a previous run without chemical equivalency. If different charges are defined the mean

charge between chemical equivalent atoms is attributed to them before starting the algorithm. We show here two typical input files for a methane molecule with chemical equivalence for the hydrogen atoms:

```
C  0.0
H  0.0
H  0.0
H  0.0
```

Listing 5.3: Input file for the *multicube* program containing the initial charges

```
C  1.7
H  1.2
```

Listing 5.4: Input file for the *multicube* program containing the van der Waals radii.

In those input files, we have set the initial guess charge to zero, the van der Waals radii are set to 1.7Å for the carbon atom and to 1.2Å for the hydrogen atoms. Since the three letter code, in this case only one letter is used, does not differ for the three hydrogen atoms they are automatically supposed to be chemically equivalent. In the current version, we like to point out that one does have to make sure that the files do not end on a new line as this crashes the program. The program is then finally be called by a command like the following:

```
./multicube chargesfile vdwfile outputfile 1 1 1 0.00000001 0.00000001 10000 *cu
```

Listing 5.5: Calling the *multicube* program.

The parameters to the command are in order:

1. chargesfile: This parameter indicates the file containing the charges in atomic units as shown in listing 5.3,
2. vdwfile: Here the parameter indicates the file containing the van der Waals radii in ångström as shown for instance in listing 5.4.
3. outputfile: Indicates the file into which the charges obtained in the minimization procedure shall be written.
4. The fourth parameter defines the number of OpenCL devices to be used.
5. The fifth parameter indicates the measurement to be used where 1 is used for L_1 type measurements described by equation (5.61), 2 for L_2 type measurements described by equation (5.62), 3 for weighted L_1 type measurements described

by equation (5.63) and 4 for weighted L_2 type measurements described by equation (5.64).

6. The sixth parameter indicates the initial stepsize γ, in this case $\gamma = 1$.
7. The seventh parameter defines the Δ used in the numerical derivatives that are evaluated as shown in equation (5.65). The values used should be close to the square root of the machine epsilon. In IEEE-754 [133] 32bit single precision floating point numbers the machine epsilon is known to be around $\epsilon = 5.96 \cdot 10^{-16}$, while for 64bit double precision it is known to be around $\epsilon = 1.11 \cdot 10^{-16}$ and hence, the value chosen for the example of the 64bit calculation here is $\sqrt{\epsilon} \approx 10^{-8}$.
8. The eighth parameter is the convergence criterion. The algorithm stops if a stepsize γ smaller then this parameter is needed.
9. The ninth parameter is maximum number of iteration steps to be evaluated. The algorithm aborts in this case even if the convergence criterion is not reached.
10. After all parameters one has to put all the cube files defining the grids to be minimized in this minimization step at once. In this case all files ending with `cu` in the current working directory are selected to be taken into account.

Further one has to make sure in the current version that the files *dpot.cl* and *spot.cl* are in the current working directory in order for *multicube* to function.

5.6 Tests and Insights of the New Algorithm

Having derived, and programmed our own ESP like algorithm we tested it with a comprehensive benchmark set, in order to see whether it has achieved its objective, being accurate in deriving partial charges for the biological molecules that we treat in this work, hence be capable to assign partial charges to a N- or C-type repeat from the *Stalk* of the TSP SD. Further we wanted to see whether we are competitive with similar programs in the

field, and if the ideas that we have implemented really do result in partial charges of a higher quality. Hence we created a benchmark set which we will outline here:

The Benchmark Set

We have divided the benchmark set into three different types of molecules, but all our partial charge attributions share a common set of parameters.

In the initial state all charges of the non charged molecules have been set to zero. In case of the charged molecules a single charge has been set to the charge of the complete system, while all the other charges have been set to zero. In all calculations the van der Waals radii have been set to: H=1.2, Li=1.82, C=1.7, N=1.55, O=1.52, Al=1.84 and S=1.8Å [134] [135]. All the tests have been performed on either CPU or GPU devices. Only the L_1 (5.61) or the L_2 (5.62) type measurements have been examined. A stepsize of $\gamma < 10^{-8}$ was chosen to be the convergence criterion. 10^{-8} for reasons pointed out in section 5.5 has been chosen to be the Δ in numerical derivatives as defined by equation (5.65). Further values between the different types of molecules that are:

1. All natural amino acids in zwitterion conformation: These molecules have been chosen for the obvious reason that we deal with a biological molecule composed of them. The *Molden* [136] program has been used to create their initial structures. These structures have been minimized in HF/6-31* theory. Minima have been confirmed by vibrational analysis. From the electron density obtained in HF/6-31* theory *Gaussian* [124] cube files containing 80^3, 100^3 150^3 points with the corresponding *ab inito* potential have been created. Using these files multicube evaluated the atomic monopolar charges for all of them.

2. Polar molecules: This set of molecules contains, N-methylacetamide, acetone, *hydrogen cyanide* (HCN), *alanyl dipeptide* (DiAla), methanol, ethanol, formamide and *lithium aluminium hydride* (LiAlH4). In this set, initial structures have been modeled using the *Molden* program. They have been chosen to test the advanced features of our algorithm such as rotational stability, and further to compare against a testset that has been used in different methods. The *ab initio*

electrostatic potential was obtained by calculations in HF/6-31*. The molecules have been minimized and verified using vibrational analysis. *Gaussian* cube files containing the *ab initio* electrostatic potential in 80^3 points have been generated in 57 rotations (19 for each axes) for each molecule. In order to obtain these rotations a small script called *cuberotor* has been created, that automatically generates the appropriate input files for the *cubegen* utility from *Gaussian*. The resulting cubes, have then to be fixed, a bases transformation is necessary on the atom coordinates and the bases vector definition of the cube due to their rotated bases set that is not supported by our *multicube* program. The script written for this task is called *cubefix*. *Multicube* has then been run on all these cubes individually and once more, for each set of the 57 rotations, taking all the rotations in one single minimization into account.

3. The 8N repeat from the *Stalk* of the TSP-1 SD structure designated by its PDB code 1UX6: This molecule is the one to be investigated in the end by the *multicube* program that we have developed. In order to make a first test on molecules of this size, the 8N structure has been cut out of the 1UX6 PDB structure. The resulting structure contains the water molecules next to the 8N repeat and the two calcium ions. The structure has been protonated using the *pdb2gmx* utility that is shipped with *GROMACS* [30]. A single point calculation using the *Gaussian* software has been performed in BHandHLYP/6-31G** theory. We like to point out that this structure has significant drawbacks as no minimization has been performed. 48 *Gaussian* cube files (16 different rotations around each axes) containing 80^3 testpoints of the *ab inito* potential have been generated. One calculation minimizing all the cubes at once has been performed on a machine gratefully provided by *Transtec France*. This machine contained four GPUs on that were utilized by the *multicube* program in this minimization. This is the only structure where during our benchmark calculations chemical equivalency has been imposed. Atoms residing on the same type of amino acid that are of the same atom type according to the OPLS-AA force field [79] [80] have been defined to be equivalent. Note that this chemical equivalency is not the same

as in the OPLS-AA force field as this force field also contains equivalent atoms that belong to different kinds of amino acids. Our definition resulted in 102 individual charges that have been calculated for the 218 atoms in the system.

Results Obtained from the Benchmark Set

In order to have one single value that puts our charges into relation with the results of other algorithms, the reproducibility of the dipole moment has been chosen to compare our benchmark sets. Least squares have been used in order to evaluate the differences between the dipole moments:

$$\Delta|\mu|^2 = \left(|\mu|^{[\text{Ab Initio}]} - |\mu|^{[\text{Charges}]}\right)^2, \tag{5.66}$$

where μ is the dipole moment vector. Further the sum of the least squares has been evaluated for the different sets of molecules presented above in order to obtain statistical values from these benchmark sets. In order to calculate the dipole moments from charges obtained with *multicube* a script called *dipole* has been written. The script can be compiled using

```
gcc dipole.c -o dipole
```

Listing 5.6: compiling the *dipole* script

and can be executed using:

```
./dipole cubefile chargesfile
```

Listing 5.7: compiling the *dipole* script

where `cubefile` is a *Gaussian* [124] cube file from which the geometry of the atoms is read. The parameter `chargesfile` points to the charges output file from *multicube*. For the first set of molecules, the amino acids, we avoided taking the charged molecules, arginine, arpartic acid, glutamic acid and lysine into account. Using the single 80^3 points *Gaussian* cube file and deriving the charges using the L_1 (5.61) measurement function the sum of the least squares defined in equation (5.66) yields: $\sum_i \Delta|\mu_i| = 0.000497$. Using the L_2 (5.62) measurement function for the same setup yields: $\sum_i \Delta|\mu_i| = 0.005011$. The ESP method as implements in *Gaussian* in comparison yields $\sum_i \Delta|\mu_i| = 0.002526$. Detailed per molecule results for the amino acids are shown in table 5.1.

For the second set of molecules, the polar molecules, a different approach has been chosen. As outlined above we have

created 57 *Gaussian* [124] cube files containing the molecule in different rotations. We were particularly interested if minimizing all the grids in one and the same calculation, hence F from equation (5.56) is evaluated by a sum over all the points p_i from the different cubes. This yields improved charges compared to evaluating each grid individually and taking the mean. We thus performed both types of calculations. For the sum of least squares across the molecules of the set of polar molecules, we obtained for calculations where the charges were evaluated for each grid individually and the mean was taken afterwards, using the L_1 type measurement function (5.61): $\sum_i \Delta|\mu_i| = 0.002506$. Further for the same using the L_2 (5.62) type measurement function $\sum_i \Delta|\mu_i| = 0.003818$.

In contrast for calculations where we minimized all the 57 grids arising from the different rotations at once, we obtained using L_1 type measurements: $\sum_i \Delta|\mu_i| = 0.001625$ and for L_2 type measurements: $\sum_i \Delta|\mu_i| = 0.003712$. The ESP method as implemented in *Gaussian* [124] [33] yields in comparison $\sum_i \Delta|\mu_i| = 0.002105$. Detailed per molecule results are highlighted in table 5.2. The results outlined here show that minimizations that used the L_1 measurement function (5.61) yield better results than those that used the L_2 measurement function (5.62). We were able to reproduce the dipole moment better using the L_1 measurement function then the ESP method that is implemented in *Gaussian* [33] [124] for the set containing the amino acids and, even if the difference is not as high as for the amino acids, for the polar set of molecules. Minimizations using our algorithm in combination with L_2 measurements however turned out to perform worse than the ESP method that is implemented in *Gaussian*. We further would like to point out that there are molecules, especially in the set of the polar molecules, where the ESP method from *Gaussian* yielded better results then our method in using either L_1 (5.61) or L_2 (5.62) type measurements. Regarding the idea to minimize all grid in one minimization we found that this approach yielded better results than minimizing the grids individually and taking the mean.

To further confirm these results, we also calculated the norm of the difference of the dipole vector that is obtained from the *ab initio* potential and the dipole vector that arises from the charges obtained by applying our algorithm. We did this in order to be sure that not only the norm of the dipole vector is preserved but

a-acid	*ab initio*	L_1 80^3	L_1 100^3	L_1 150^3	L_2 80^3	L_2 100^3	L_2 150^3	ESP
ala	3.9064	3.9099	3.9100	3.9100	3.9144	3.9142	3.9143	3.9123
asn	3.4354	3.4326	3.4327	3.4326	3.4478	3.4477	3.4478	3.4414
cys	3.4556	3.4650	3.4652	3.4651	3.4670	3.4672	3.4668	3.4590
gln	4.7365	4.7262	4.7261	4.7261	4.7466	4.7465	4.7465	4.7382
his	3.1152	3.1149	3.1151	3.1151	3.1382	3.1382	3.1383	3.1310
ile	3.7988	3.8018	3.8016	3.8017	3.8061	3.8061	3.8063	3.8081
leu	3.9161	3.9218	3.9217	3.9217	3.9302	3.9303	3.9305	3.9259
met	3.3939	3.3919	3.3917	3.3919	3.4396	3.4395	3.4399	3.4240
phe	4.1302	4.1353	4.1355	4.1356	4.1494	4.1492	4.1495	4.1424
pro	4.1949	4.1971	4.1972	4.1974	4.1952	4.1952	4.1953	4.1898
ser	3.4904	3.4999	3.5000	3.5001	3.4930	3.4930	3.4930	3.4928
thr	3.4956	3.5037	3.5039	3.5039	3.5106	3.5107	3.5108	3.5084
trp	4.6161	4.6213	4.6212	4.6210	4.6452	4.6454	4.6453	4.6391
tyr	4.0397	4.0410	4.0410	4.0411	4.0533	4.0533	4.0536	4.0514
val	3.7896	3.7942	3.7946	3.7944	3.7971	3.7974	3.7973	3.7992

Table 5.1: Dipole moments for non charged amino acids [Atomic Units]. Values indicated to the third power give the number of testpoints

		57 grids (1 min.)		57 grids mean (57 min.)		
Molecule	*ab-initio*	L_1 80^3	L_2 80^3	L_1 80^3	L_2 80^3	ESP
N-Methylacetamide	1.5681	1.5567	1.5520	1.5360	1.5499	1.5631
Acetone	1.2271	1.2369	1.2443	1.2373	1.2447	1.2348
HCN	1.2865	1.2595	1.2618	1.2595	1.2617	1.2750
DiAla	1.1197	1.1129	1.1094	1.1108	1.1061	1.1095
Ethanol	0.6840	0.6858	0.6721	0.6857	0.6721	0.6692
Formamide	1.6133	1.5966	1.6118	1.5966	1.6117	1.6139
LiAlH4	2.6308	2.6178	2.5819	2.6184	2.5818	2.5907
Methanol	0.7348	0.7496	0.7314	0.7492	0.7314	0.7270

Table 5.2: Dipole moments for a set of polar molecules [Atomic Units]. Values indicated to the third power give the number of testpoints

that the dipole vectors are also pointing into the same direction. Therefore we calculated the relative norm:

$$\delta\mu = \frac{\left|\mu^{[\text{Ab Initio}]} - \mu^{[\text{Charges}]}\right|}{\mu^{[\text{Ab Initio}]}} \tag{5.67}$$

The values for $\delta\mu$ are highlighted in tables 5.3 and 5.4. These results are in agreement with the statements made above.

From the second set of molecules, we where also able to investigate the rotational stability of our algorithm by calculating the mean and the standard deviation from the individual rotations. We found that in all cases we are in conformance with previous works that have optimized the location of the testpoints

a-acid	L_1 80^3	L_1 100^3	L_1 150^3	L_2 80^3	L_2 100^3	L_2 150^3	ESP
ala	2.262	2.315	2.309	4.014	4.009	4.028	3.689
asn	2.067	2.072	2.137	5.526	5.492	5.559	3.028
cys	4.107	4.157	4.123	13.775	13.866	13.783	6.895
gln	2.337	2.351	2.383	3.045	3.014	3.047	2.117
his	4.983	5.016	5.093	10.452	10.509	10.584	6.504
ile	0.890	0.854	0.880	4.340	4.327	4.379	3.765
leu	2.122	2.064	2.071	3.779	3.787	3.855	2.533
met	4.148	4.196	4.200	17.163	17.201	17.295	12.476
phe	3.182	3.398	3.368	5.459	5.471	5.541	3.476
pro	2.493	2.475	2.496	2.621	2.621	2.629	3.429
ser	4.387	4.324	4.435	4.304	4.274	4.330	4.850
thr	2.529	2.604	2.618	6.268	6.231	6.275	5.634
trp	1.349	1.275	1.196	7.000	6.998	6.961	5.463
tyr	2.630	2.504	2.560	6.674	6.628	6.721	4.717
val	1.495	1.591	1.534	4.419	4.479	4.451	3.221

Table 5.3: Norm of the difference between the *ab initio* dipole vector and the dipole vector calculated from derived charges for amino acids [Atomic Units]·10^3. Values indicated to the third power give the number of testpoints

	57 grids (1 min.)		57 grids mean (57 min.)		
Molecule	L_1 80^3	L_2 80^3	L_1 80^3	L_2 80^3	ESP
N-Methylacetamide	8.377	13.074	21.413	15.085	8.289
acetone	8.026	14.046	8.288	14.316	6.343
HCN	20.998	19.194	21.005	19.298	8.939
diAla	7.094	14.870	8.756	17.080	14.044
ethanol	8.685	26.487	8.344	26.515	23.885
formamide	11.160	4.727	11.014	4.717	9.758
liAlH4	4.927	18.591	4.704	18.638	16.683
methanol	21.587	18.447	20.840	18.466	12.859

Table 5.4: Norm of the difference between the *ab initio* dipole vector and the dipole vector calculated from derived charges for a set of polar molecules [Atomic Units]·10^3. Values indicated to the third power give the number of testpoints.

	L_1		L_2	
	mean(q)	σ	mean(q)	σ
C	0.333	0.013	0.313	0.002
H	0.024	0.003	0.027	0.001
H	−0.040	0.004	−0.035	0.001
H	−0.040	0.004	−0.035	0.001
O	−0.712	0.006	−0.693	0.002
H	0.435	0.002	0.423	0.001

Table 5.5: Rotational stability evaluated from 57 rotations. Shown are the mean values of the charges obtained and their standard deviation. [Atomic Units]

in the space around the molecule in order to achieve better rotational independence [129]. Further we can state that minimizations that were made using the L_1 (5.61) measurement function achieve less rotational stability then minimizations that were made using the L_2 (5.62) measurement function. Results from a calculation of the atomic monopolar charges of ethanol are shown in table 5.5.

The 8N repeat from the *Stalk* which forms the third group of our molecules has been used to do a first test on a large molecule. Due to the limited availability of machines equipped with GPUs and the limited time that we had access to a multi GPU machine at *Transtec*, we were only available to make a single testrun of this molecule. The first results obtained however looked promising. The minimization that took place using the L_2 measurement function in minimizing all 48 grids, the different rotations, at once yielded a value of 1.24e and 1.01e for the charge of the calcium ions. Since the structure has significant drawbacks we further neglected the obtained charge values and focused ourselves on the comparison of the potential function. In order to do so we first investigated the *ab initio* and the potential created by the monopolar charges obtained from this run using the *Visual Molecular Dynamics* (VMD) [37] program. We found that isosurfaces, surfaces where the values of the potential are the same, are almost indistinguishable. In order to get a better idea of the difference of the two potentials we took the mean of the absolute difference of all the testpoints on the grid defined by a 80^3 *Gaussian* [124] cube and found that a point of the potential generated by the monopolar charges obtained is on average $9{\cdot}10^{-4}$ Hartree away from the *ab initio* potential.

Runtime and Algorithmic Properties

The algorithm is, due to its iterative nature, highly computational intensive. In order to overcome this obstacle, a parallel implementation in OpenCL, as outlined in section 5.5, has been chosen. Running our benchmark set, we gained several insights into the different aspects of the algorithm. We found that even if the L_1 measurement function (5.61) seems to yield better charge models in terms of dipole reproducibility, it also introduces a significant higher computational effort compared to the L_2 measurement function (5.62). This is due to a higher number of steps that have to be evaluated until the convergence criterion is reached. For the acetone molecule for instance, we found that 8469 minimization steps had to be evaluated until convergence had been reached in case of a minimization where the L_1 measurement function (5.61) has been used. In the case where the L_2 measurement function (5.62)was used during minimization only 1018 steps had to be evaluated to reach convergence. The number of steps that had to be evaluated further seemed to decrease with the number of rotations, *Gaussian* cube files taken into account. Again in case of the acetone molecule for instance, we found that when minimizing a single 80^3 points grid, the algorithm took as stated in L_1 minimization 8469 steps. When minimizing all 57 grids containing 80^3 gridpoints, we found that the number of steps evaluated decreased to 3473 steps. We further changed the convergence criterion in defining several different minimal stepsizes. We found in our testruns on the acetone, methanol and ethanol molecule that charges did not change more then 10^{-3}e for stepsize limits of $\gamma < 10^{-6}$. Further interesting is that the stepsize is highly fluctuating and thus we suggest that at least some kind of convergence test is tried before limiting oneself to a certain convergence criterion. This effect as observed during a minimization using the L_2 measurement function (5.62) is highlighted in figure 5.2. This type of fluctuation was also observed for calculations where the L_1 type measurement function (5.61) has been used.

The runtime was of course effected by the number of steps that had to be evaluated, but further it was also effected by the type of hardware used.

Even though this algorithm has not been specifically optimized for a certain type of hardware, running the algorithm on CPU as

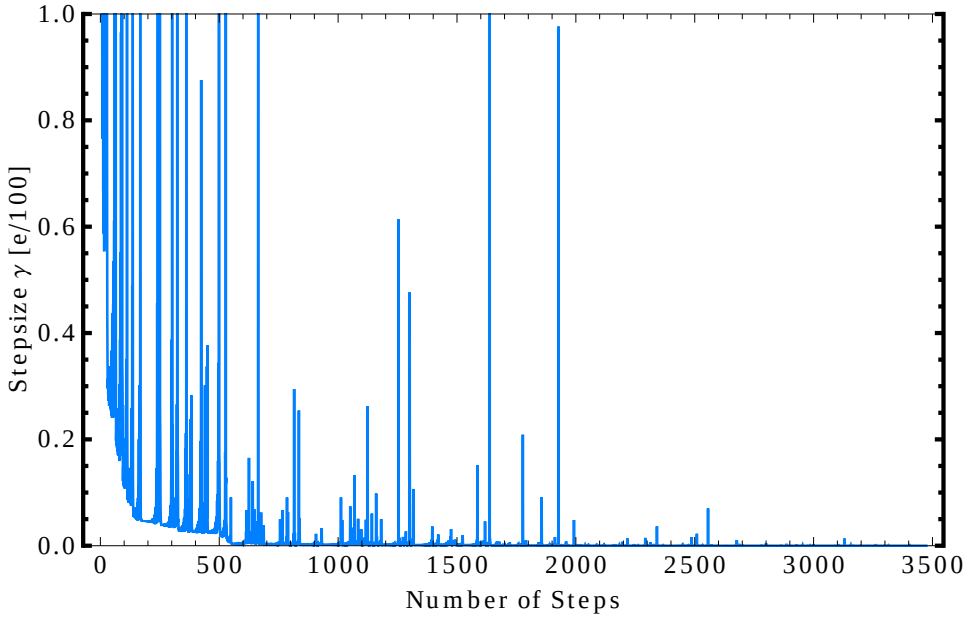

Figure 5.2: Stepsize fluctuation during a calculation of the monopolar atomic charges for the acetone molecule using the L_2 measurement function (5.62)

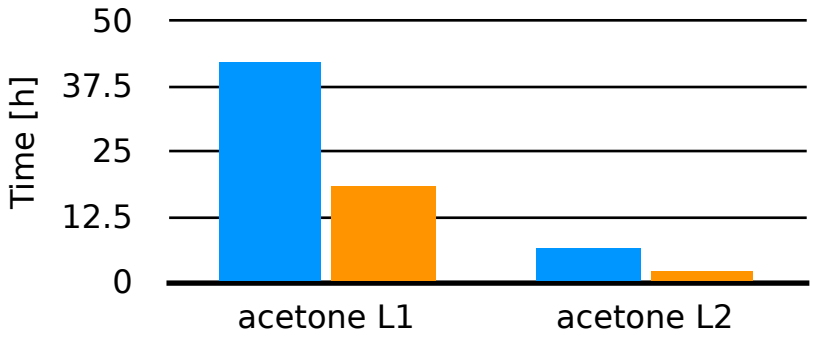

Figure 5.3: The runtime of the algorithm in order to obtain a charge model for the acetone molecule, using either the L_1 (5.61) or L_2 (5.62) measurement function: Times for calculations using two Intel Xeon X5650 hexacore CPUs on Mac OS X 10.6 with Apples OpenCL 1.0 implementation are highlighted in blue, while times for calculations using one single Nvidia Tesla GPU on Linux with Nvidia's OpenCL 1.1 implementation are shown in orange.

well as on GPU devices, we found that GPU devices outperformed the CPU independent of the measurement function that has been used. We noticed for instance that a single Nvidia Tesla M2050 GPU outperformed a two Intel Xeon X5650 hexacore processors using all the available cores when running the reference implementation of our algorithm. These results are detailed in figure 5.3.

5.7 Discussion on the New Algorithm

Even though the topic has been worked on for decades, we have shown that there is still room for improvement in the field of partial charges derived using an ESP like method. We shall discuss them in the following:

General Remarks

By increasing the number of testpoints, using iterative methods and parallel programming techniques, we have found promising results and are confident that we are heading in the right

direction. During the development and while testing of the algorithm, we have been intrigued by the impact of the measurement function and the different results the L_1 type function (5.61) finds in comparison to the L_2 type function (5.62). Even though that the development was led by the idea of being able to resolve the partial charges for the N- or C-type repeats belonging to the *Stalk* of the TSP SD, only the future will make clear whether this algorithm is capable to treat complex objects like these where the charges are buried, and many different charges have to be derived, an example where current state of the art methods fail to succeed. The preliminary results on the 8N repeat structure however are promising and we are optimistic in this respect. Even though the main goal, to resolve the problem of the polarization created by the calcium ions in the TSP SD was not achieved, due to constraints in computational resources, several interesting aspects in deriving partial charges have been revealed. We have successfully shown that our algorithm succeeds in working around the problems of rotational stability, and that the possibility to minimize multiple grids at once yields better results then taking the mean over individual minimizations. The strongest point of this algorithm is probably the possibility to easily exchange the measurement function D in equation (5.56) to almost any type of measurement that might be imagined. Having highlighted that the L_1 type function derives better charges in many cases we think that future work should be done in finding an optimal measurement function. We suggest the following enhancements of the algorithm presented herein.

Suggested Improvements

Several different aspects of this method can for sure be improved, we take the freedom to suggest some here:

1. Improvements around the measurement function: In the algorithm we implemented four different types of measurement functions shown in equations (5.61), (5.62), (5.63) and (5.64) and tested two of them. We believe that the algorithm can further be improved by different testfunction. One idea would for example be to run a minimization in order to obtain the charges, storing the fluctuations in the differences between the *ab initio* potential and the potential created by the charges, and run a second run where the testpoints are

in using the testfunction weighted by the fluctuations observed, equalizing the weight of the testpoints in the sum to be evaluated in order to obtain F from equation (5.56).

2. Taking more constraints into account: The reference implementation *multicube* is currently limited to the constraints of charge conservation and chemical equivalence. However we would like to point out that the algorithm presented herein would equally work for any type of linear constraints. To further detail this, one shall consider the following: Be

$$\sum_j A_{ij} x_j = b_i \tag{5.68}$$

a linear constraint. Here x_j holds the charges to be resolved, either Q for single atoms or W for different atom types. Vector b_i holds charge constraints, for instance the charge of the complete molecule to be conserved, or the charge for parts of the molecule to be conserved. Then from linear algebra we know that solutions for such a problem lie in the space:

$$x_i \in \{z_i + \ker(A_{ij})\} \tag{5.69}$$

where z_i is an initial solution. In our case this can be the initial charges starting the algorithm, and $\ker(A_{ij})$ is the kernel of the matrix A_{ij}. The point in here lies that (5.69) always represents a hyperplane in the space of the charges and hence, one always can find a projection into this subspace. From which follows that the algorithm presented in here should be expendable to such arbitrary constraints that in turn should prove to be helpful for example for the derivation of partial charges in the development of fragment libraries that can be used in MD simulations.

3. Runtime improvements: One shall be able to improve the runtime by at least a magnitude in porting this algorithm to a specific hardware. In here, a general approach using OpenCL was chosen in order to support a broad range of hardware while penalizing performance. An other point where significant runtime performance improvements shall be achievable, is in improving the linesearch problem of finding the right stepsize γ to be used in equation (5.57) which is very basic in this reference implementation.

Other Developments in the Field of Partial Charges

C. Chipot [131] et al have developed an algorithm to derive multipoles using statistical analysis. By choosing random points on a grid and evaluating the multipoles from these points they were able to obtain further parameters that describe the quality of such charges. The author points out that in several conformations least squares are ill defined and hence does try to tackle this problem with statistical analysis, while we are trying to solve this problem in increasing the number of test points as well as in changing the measurement function from L_2 (5.62) to presumably better L_1 (5.61).

Recent developments of B. Wang et al. [137] [138] featuring a method called *Full density screening* takes effects like charge penetration into account in fitting these charges. Even though the electrostatic potential is to a great extend modified, the author relies on the L_2 (5.62) measurement to fit the potential. It would be interesting to see what happens if the work B. Wang et al. would be extended to an algorithm like the one presented in here, and to see the impact of a change in the measurement function.

5.8 Conclusion on Molecular Quantum Mechanics

Many things have been said around the derivation of partial charges, but not only the partial charges, that in our case are important in order to link this level of theory to the level of classical MD simulation. However molecular QM, and its ability to model the electronic shell are further interesting not only in obtaining partial charges but in the general investigation of biological molecules. Advances in computing technology make calculations around large molecules common in the field such as the 8N repeat from the TSP SD's *Stalk* feasible as it was shown in here. One should further also state that also molecular QM, just as classical MD simulation has its weaknesses. A large number of different QM methods, and basis sets yield yet imperfect results that deviate from empirical measurements. One may point out for instantaneous the large differences in dipole moments obtained from *ab initio* calculations to be found in the *National*

Institute of Standards Computational Chemistry Comparison and Benchmark Database (CCCBDB) [139]. Regarding the wavefunction, and its ability to represent the probability density of finding an electron around a molecule, shorter the electron cloud we were not only interested in deriving partial charges from these methods, but also in visualizing such QM results. Accordingly we developed methods that allow us to investigate changes in the electron density, as we change the conformation of the molecule, which brings us right to the next chapter, molecular visualization.

6 | Molecular Visualization

6.1 Introduction

Molecular visualization is a key method in order to understand macromolecular processes such as those described herein. Visualization of macromolecules dates at least back to the first X-ray structures of such molecules. The first visual model of such a structure is attributed to J. C. Kendrew et al. [35] which has been published in 1958. These models were initially build mechanically, switching to computers as soon as 1966 [36]. As the structure of proteins or other large biological molecules is related to its function, visualizing it is an important factor in research. Several results that we presented in chapters 4 and 5 have only been possible due to the visualization of the results. Graphical representations of biological molecules have hence become standard.

Further graphical visualization helps not only in understanding the function of biological molecules but also in communicating problems and solutions around them. A movie of a moving molecule can summarize a conformational change, binding to different ligands and other dynamical functions. Besides these rather practical aspects, visualization also has an aesthetic aspect. The visualization of macromolecules can thus further play a role in art and also in vulgarization of the research undertaken.

As already stated we tried to resolve several problems by the methods of molecular visualization during the work presented herein and were rather quickly confronted with the limitations of available tools such as *Visual Molecular Dynamics* (VMD) [37] and *PyMol* [38]. In order to overcome these limitations we were using new approaches, developing new methods which we will present in the following. Presenting parts of our work, we were further particularly interested in creating high quality *photorealistic* movies to facilitate communications during internal sem-

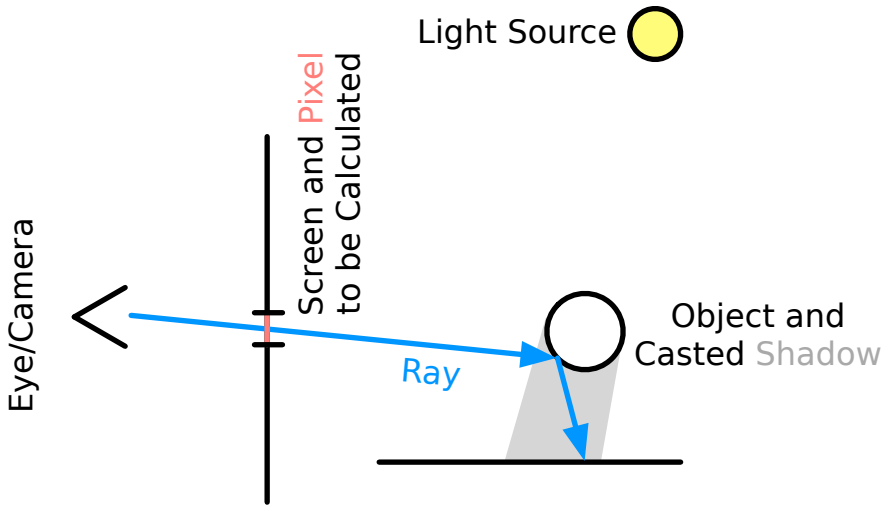

Figure 6.1: An illustration of a ray tracing algorithm

inars and presentations at international congresses.

6.2 Introduction to Computer Graphics and Ray Tracing

In this section we will briefly introduce the basic techniques that are used by the computer software to generate realistic images.

Ray tracing is the state of the art technique to generate *photo-realistic* images in using a computer. These algorithms are built around the idea that one literally traces the path of a ray from the observer through a pixel on a screen, hitting the objects in the scene, reflecting on them until the ray has traveled a certain distance or has underwent a certain number of reflections. In this process, the color value of the pixel on the screen is calculated. An illustration is shown in figure 6.1.

The physical model behind ray tracing described above is very flexible and image quality and accuracy are largely dependent on the accuracy of this model. We shall briefly sketch the surface on how this is done. The theory behind ray tracing was summarized

in the rendering equation [140]:

$$I(x,x') = g(x,x')\left(\epsilon(x,x') + \int_S \rho(x,x',x'')I(x',x'')dx''\right), \tag{6.1}$$

where $I(x,x')$ is the intensity of light passing from point x' to x, $g(x,x')$ is a geometry term ϵ shall be related to the intensity of emitted light and $\rho(x,x',x'')$ shall be related to the intensity of light scattered from x'' to x' by the surface at x'. S shall be all surfaces in the system which includes a background surface, in general a hemisphere that is large enough to contain the entire system of surfaces. Here one can see that the geometry term $g(x,x')$, the term to be related to emittance $\epsilon(x,x')$ and the term to be related to light scattering $\rho(x,x',x'')$ are very vague statements and these terms can be adapted to the level of theory used. The rendering equation further leaves out a lot to be found in a common ray tracer. Wavelength, polarization and as no phase is taken into account in equation (6.1). Diffraction is also ruled out. Nevertheless we shall state it here for the principles of ray tracing, which is one of the key methods implemented in the computer programs that we have used. We further shall think a bit about the solution of equation (6.1) for which several methods have been developed. Let us rewrite equation (6.1) so that:

$$I = g\epsilon + gMI, \tag{6.2}$$

where M is a linear operator, a matrix if you want, that describes the integral in equation (6.1). We can further rewrite this into:

$$(\mathbf{1} - gM)I = g\epsilon, \tag{6.3}$$

where $\mathbf{1}$ is the identity and hence, we can do the Neumann series development:

$$I = (\mathbf{1} - gM)^{-1}g\epsilon = g\epsilon + gMg\epsilon + gMgMg\epsilon + g(Mg)^3\epsilon\ldots \tag{6.4}$$

This can be physically interpreted as if I is the sum of a once scattered term, a twice scattered term, etc. Basic approximations can then be made like the following:

$$I = g\epsilon + gM\epsilon_0, \tag{6.5}$$

which reduces itself practically, as M operates directly on ϵ_o, to a sum over point light sources. In this approximation, as the integral over x'' is missing all visibility from emitting surfaces, there

are no shadows and no extended lighting surfaces. This probably is one of the most simple implementations of a rendering algorithm. Let us just go a bit further:

$$I = g\epsilon + gM_o g\epsilon_o + gM_o gM_o g\epsilon_o + \dots \tag{6.6}$$

In this case M_o shall be an operator that takes reflexion and refraction into account in summing in different ways over $g\epsilon_o$. A model like this can account for features like shadows, reflexion or refraction. This is pretty basic compared to what is implement in today's ray tracing render engines, but should give a basic idea about how ray tracing works and how ray tracing algorithms are developed.

6.3 Visualization and Classical Molecular Dynamics

As a large part of the work presented herein deals with classical *Molecular Dynamics* (MD). We dedicate this section to the visualization of the features of MD simulations and the trajectory resulting from such. In order to do so, we shall first think about basic visualization of a molecular structure such as a protein, and how to generate high quality images from them.

Protein Structure Rendering

During our work, we generated *photorealistic* renderings either with *VMD* [37] and *Tachyon* [141] or *PyMol* [38] in conjunction with *Blender* [39]. *Photorealistic* images of proteins have in general been created from a cartoon mesh representation with spheres for ions or licorice representation for ligands. We will at first describe how rendering in *VMD* is done, and then head over to the *PyMol*, *Blender* combination.

In *VMD*, an export filter for the *Tachyon* ray tracer exists. This export filter allows one to generate input files for *Tachyon*. In general one has to proceed in the following way in order to generate a high quality image:

One has to make sure that the structure is well represented in VMDs OpenGL Display. As the display further defines the size of the resulting image, the display size should be a multiple of the final resolution, so that the quotient resulting from dividing the

images height by its width is conserved. In order to resize the display to the right size the following command can be issued on the console within VMD:

```
display resize width height
```

Listing 6.1: Resizing the *VMD* OpenGL display

Here `width` and `height` are representing the number of pixels. Further, ambient occlusion and shadows are adding realism to the final image. These features can be turned on by the following commands in the console of *VMD*:

```
display shadows on
display ambientocclusion on
display aoambient 0.80
display direct 0.30
```

Listing 6.2: Enabling ambient occlusion and shadows

Enabling shadows needs no further explanation. Ambient occlusion is a method which calculates the accessibility of objects in a scene, to light, dust or other particles. In this case this allows buried objects, like holes in a molecule to appear darker than non buried objects. Further, we suggest that one verifies the colors and the materials used before rendering the image, in order to generate images that are visually and technically pleasing. Of special importance before rendering should be the choice whether one would like to create an orthographic or a perspective image. Orthographic images appear as a direct projection onto a two dimensional plane, while perspective images recreate a image as if captured by a camera lens or the human eye. Far away objects thus appear smaller in perspective images, while in orthographic images they appear of the same size as close objects. While perspective images are more natural, and are what we are used to looking at photos, for technical reasons orthographic images are sometimes preferable. A comparison between orthographic and perspective images is shown in figure 6.2. In *VMD* one can switch between the two types of projections using the following commands:

```
display projection Orthographic
```

Listing 6.3: Switching to the orthographic projection.

```
display projection Perspective
```

Listing 6.4: Switching to the perspective projection.

Generating a *Tachyon* input file can then be done by issuing the following command to the console of *VMD*:

```
render Tachyon inputfilename.dat
```

Listing 6.5: Generating a *Tachyon* input file.

Rendering can than be done externally by calling tachyon:

```
tachyon inputfilename.dat -aasamples 16 -res height width -format PNG result.png
```

Listing 6.6: Rendering an image with *Tachyon*.

In this case, 16 samples for antialiasing have been used which in general means that 16 images with a slightly displaced camera a generated and a smoothed image from the resulting image is generated in order to avoid rough edges. The `height` and the `width` give the number of pixels of the final image. For images to be printed on paper, one should know that the human eye is capable of distinguishing around 300 dots per inch and that most printers are able to print at resolutions of 600 dots per inch and upwards. One has thus to choose an adequate resolution for printed communications. Some readers might note that one can generate images using *Tachyon* directly from within *VMD*. The way of decoupling the final rendering from *VMD* shown here has the advantage that *Tachyon* input files can be moved to fast machines as they do exist in *High Performance Computing* (HPC) centers, which can be useful in the case of rendering movies, as shown later.

Using *PyMol* and *Blender* the process we have chosen is somewhat different. *PyMol* in this case is used to generate the scene, which is exported in a *Virtural Reality Modeling Language* (VRML) file. Importing such a file generated in *PyMol* in *Blender* poses several problems. In order to overcome these, two different versions (2.49 and 2.5 or higher) of *Blender* have been used. We started with *Blender* version 2.49 and imported the VRML file. In *Blender* an object to be rendered is in general described by *vertices*, which are the corners of an object. These *vertices* are then linked by *edges*, which can span *faces*. At the import, we have chosen a circle subdivision of 32, meaning that a circle is represented by 32 vertices. The import in *Blender* generates several objects. Objects in *Blender* are a data structure that contain the *vertices*, *edges*, *faces* and much more, such as material data like color or the index of refraction, of an object in the scene to be rendered. Objects have been created for instance for each ion

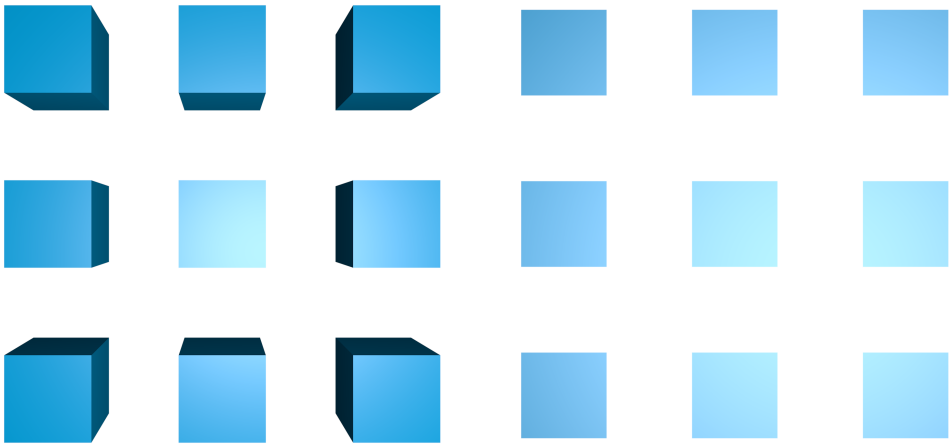

Figure 6.2: 9 cubes rendered using perspective projection on the left and in orthographic projection on the right.

and for the mesh highlighting the secondary structure of a protein, which we might call cartoon mesh in here. After the import in *Blender* version 2.49 the following protocol was applied:

1. Removing doubles: As the import of a VRML file from *PyMol* generates many vertices in the same location. One can remove them using the *remove doubles* button in *Blender*'s edit mode.
2. Converting Triangles to Quads: In calling the *Tris to Quads* function for all imported meshes one can make *Blender* try to create *faces* with four corners from *faces* with three corners. This effectively reduces the number of vertices and should make the rendered objects smoother.
3. Fixing normals: *Normals* are the normal vectors on the *faces*. These are important to exist and to be in proper direction, pointing to the outside of the mesh that represents the protein. In order to assure this, we run the *fix_all_normals.py Blender* Python script, which was kindly provided by Andrea Weikert from the Leibniz Rechenzentrum in Munich.
4. Generating materials from vertex colors: When the meshes are created by importing the VRML file generated by *PyMol*, the colors are added as vertex colors to the mesh. We

run the *mesh_mat_fromvcols.py* *Blender* Python script in order to generate materials from these vertex colors. Materials in *Blender* are data structures of an object that can be assigned to *vertices* and *faces* and contain typical material properties like the index of refraction, color etc. Vertex colors in the contrast are not of this structure and just define the color of a vertice independent from the type of material. The transformation of vertex colors to materials allows us afterwards for example to make a part of the rendered molecule transparent, if this part was highlighted in a special color in *PyMol*. This script was also kindly provided by Andrea Weikert.

5. Switching to *Blender* 2.5 or higher: In the last step the file is saved in *Blender* 2.49. A newer version of *Blender* is then opened. Using the Append command we then insert the objects from the saved file into the newer version of *Blender*. We do this way, as the interface is also saved in the files saved by *Blender* and we do not want to have the user interface of the new versions of *Blender* modified from what was saved in *Blender* 2.49.

From here on one can add proper lightning in *Blender*, change the materials of the molecule to be rendered and go ahead with rendering. We would like to note that right now sophisticated tools for importing molecules from files like *Protein Data Bank* (PDB) [50] files do exist. Such tools are implemented in integrated software packages such as the *Embedded Python Molecular Viewer* (ePMV) [142] or *Bioblender* [143]. The protocol here was used as we felt that we gained some flexibility, as contrary to *Bioblender*, which is shipped as a complete package or ePMV, which works only with certain versions of *Blender*, we were able to use the latest *Blender* version available and felt better using a stock *Blender* version than a modified one. An image of the *Thrombospondin* (TSP) *Signature Domain* (SD) rendered with *Blender* can be found in figure 6.3.

Movies from Classical Molecular Dynamics Trajectories

Now that we have shown how to render a single structure, we can go ahead to movie rendering, in particular to the rendering of classical molecular dynamics trajectories. Our simulations were all made with *GROMACS* [30]. Performing molecular dynamics

Figure 6.3: A *photorealistic* image of the TSP SD made with the help of *Blender*

is outlined in detail in chapter 4. Trajectories from simulations can easily opened in *VMD*. Even though the trajectories can be investigated in *VMD*, several steps had to be done to create high quality movies from such trajectories. A big limitation further is the large size of MD trajectories and hence, the large amount of *Random Access Memory* (RAM) required by *VMD*. Another obstacle is that we were not able using the tools shipped with *Gromacs* to prevent the molecule from turning around its axes in the simulation box. In the case of molecular visualization it is however preferable to have the box rotating around a fixed molecule. The standard transformation of trajectories has been described in detail in section 4.7. A further problem is that rendering movies with *VMD* and *Tachyon* alone does not allow for motion blur, which makes a movie from an MD simulation appear bizarre at standard frame rates of around 25 frames per second. Motion blur is the effect, that makes fast moving object appears smeared on an image. At the mentioned frame rate the human eye notices that the movement is not smooth in the absence of motion blur. We were able to overcome all the problems mentioned here and will show how we created movies from

GROMACS trajectories in the following.

Everything besides final rendering with *Tachyon*, and the addition of motion blur is implemented by a *Tool Command Language* (TCL) script that is executed within *VMD*. To use this script one prerequisites a trajectory where the protein is centered within the simulation box. How such a trajectory can be obtained is explained in section 4.7. The script begins with the following lines:

```
set tailselect "protein"
set beg_frame "1"
set end_frame "40000"
set stride "2000"
mol load gro initialframe.gro
```

Listing 6.7: Head of the *trajectory_render.tcl* script.

In this case the object to be highlighted in the movie is selected by the *VMD* atom selection *protein*. The first frame of the movie to be rendered from the trajectory is the frame number 1 and the last frame is the 40000th frame. The stride value is used to keep memory usage low. It defines how many frames *VMD* shall load into RAM at once. The file initalframe.gro contains the initial frame of the trajectory. Such a file can be created by the *g_trajconv* utility that is shipped with *GROMACS*. Now we need to define basic display properties, this is done by issuing the following command:

```
display projection Orthographic
axes location Off
display shadows on
display ambientocclusion on
display aoambient 0.80
display aodirect 0.30
```

Listing 6.8: Setting basic display properties in the *trajectory_render.tcl* script.

Here we use the orthographic projection and turn the axes indicator off. Then we want the molecule in this case to be rendered in cartoon representation which is done by the following two lines:

```
mol modselect 0 0 protein
mol modstyle 0 0 NewCartoon 0.3 6. 4.1 0
```

Listing 6.9: Changing the representation to cartoon representation in the *trajectory_render.tcl* script

One can add different representations to these two lines, such as a licorice representation for ligands. Further color changes

can be included. In the following lines we show how one can, for example, change the color of a β-sheet:

```
color Structure Extended_Beta yellow
color change rgb 4 1. 1. 0.4
```

Listing 6.10: Changing the color of β-sheets *trajectory_render.tcl* script.

Here in the first line the color of β-sheets is changed to yellow, and in the second line the red green and blue values of the β-sheet are changed. 4 is *VMD*s index for the yellow color. Other colors are referenced by different indices. In the next step we change the viewpoint in order to look at the protein from the right direction. Further we change the resolution and zoom in on the protein:

```
display resize 600 600
display resetview
rotate x by 20
rotate y by 10
scale by 2.7
```

Listing 6.11: Setting the resolution and the right viewpoint in the *trajectory_render.tcl* script

In the first line, we set the display size to be 600 to 600 pixels, which allows us to render squared images. The resetview command centers the molecule on the screen. The rotate commands turn the molecule around the specified axes by the specified degrees. The scale command zooms in to the protein by the scalefactor specified. Now that we know how we will look at our molecule, we can start the renderloop generating input files for *Tachyon* using:

```
for {set counter $beg_frame} { $counter < $end_frame } \
    { set counter [expr $counter+$stride]} {

    set begwindow $counter
    set endwindow [expr $counter+$stride]
    mol addfile traj.xtc first $begwindow last $endwindow waitfor all

    set number_of_frames [molinfo top get numframes]
    for {set i 1} { $i < $number_of_frames } { incr i } {
        set before [expr $i-1]
        set now [expr $i]
        set one [atomselect top $tailselect frame $before  ]
        set two [atomselect top $tailselect frame $now ]
        set tmatrix [measure fit $two $one]
        set whole [atomselect [$two molid] "all" frame $now ]
        $whole move $tmatrix
    }

    for {set x 1} { $x < [expr $number_of_frames-1] } { incr x } {
```

```
        animate goto $x
        set framename [expr $counter+$x]
        render Tachyon $framename.dat
        /usr/bin/bzip2 $framename.dat
    }

    set number_of_frames_minus_two [expr $number_of_frames-2]

    animate delete beg 0 end $number_of_frames_minus_two skip 0 0
}
```

Listing 6.12: The renderloop in the *trajectory_render.tcl* script

where traj.xtc is the trajectory file obtained from a classical molecular dynamics simulation. Here the outermost loop iterates across the frames in strides as defined in listing 6.7. The first inner loops aligns the molecular structures between the different frames on the trajectory so that the molecule stays in position and does not rotate. The second loop generates the *Tachyon* input files and compresses them in order to save disk space as a lot of frames can consume non-negligible amounts. In the last two lines, one can see that two frames of a stride are always kept to assure proper alignment between the strides. Now that we have obtained numerous *Tachyon* input files, we have to render them. This can been done by looping across the the files and using *Tachyon* to render them all by a command like the one shown in listing 6.6. After all images have been rendered, one has to take care of the motion blur effect. We have created the motion blur effect in oversampling each image. Every final frame of the movie is thus created from multiple rendered images. We found that generating a final frame from 12 to 16 images was sufficient to generate visually appealing movies from trajectories. In our case we did this by adding an alpha channel to each image. An alpha channel is a pixel value that defines the transparency of the pixel. This way we made every pixel of each individual image have an opacity:

$$O = \frac{1}{\text{Number of images to be merged}} \tag{6.7}$$

Then we added the images together forming the final image. If the value of a pixel has not changed over the number of images that have been merged this way the pixel value stays exactly the same. The the value received in the final image is a normed sum of the images that have been used to create the final image, effectively creating the impression of motion blur. We have imple-

mented this in a small script in Objective-C using Apples proprietary *Core Image* framework and hence, the following script only runs on the Mac OS X operating system. The same effect should however be easily implemented in different ways on other platforms. To use our script called *images_blend* one has to compile it in the following way:

```
gcc -O2 image_blend.m -framework AppKit -framework QuartzCore -o image_blend
```

Listing 6.13: Compiling the *images_blend* script

The script then can be run issuing the command:

```
image_blend image1.png image2.png [...] finalimage.png
```

Listing 6.14: Calling the *images_blend* script

where all images before the final image are merged into the finalimage. An image created with this script from the first twelve frames of simulation 31, as indicated in table 4.1 can be found in figure 6.4. Looping across the images created by *Tachyon* one can create such final images for the entire movie. In the final step the resulting blurred images are encoded into a movie file. This can be achieved using a variety of programs that have been written in order to encode videos such as *ffmpeg* or Apple *Quicktime*.

Visualization of Results from Principal Component Analysis

Principal Component Analysis (PCA) has been used to investigate the motions, conformational changes, during our simulations. PCA has been discussed in detail in section 4.7. In PCA one obtains the so called principal components, vectors that give the principal directions in which the motions of the protein happen. Such a principal component holds three values for each object, atom or coarse grained beads, in each simulation. Plotting these vectors onto an initial structure has proofed to be an efficient way to investigate motions in MD simulations. The length of each vector is proportional to the amplitude of motions in the directions that the vectors are pointing to. This way one can either investigate where significant flexibility is to be found on the molecule or protein investigated or where conformational changes might take place. An illustration is shown in figure 6.5. Such a figure can be drawn in *VMD* using the following script:

```
set splitfiledir "/simulation-x/compslit"
```

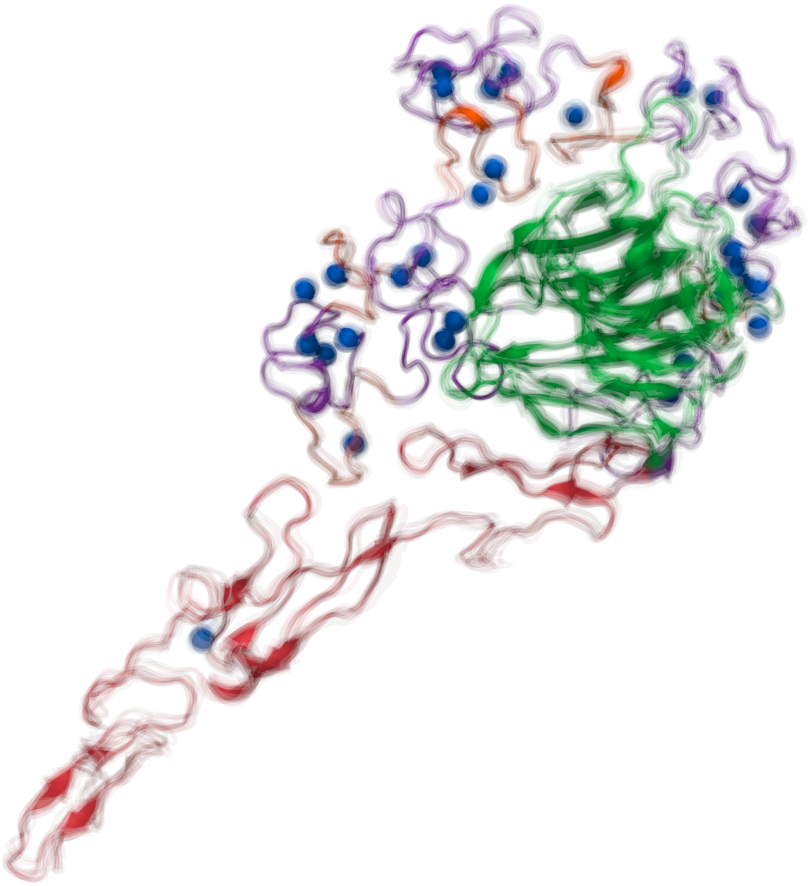

Figure 6.4: Blurred image: This image has been created using the *images_blend* script from the the first twelve frames of simulation 31 as referenced in table 4.1.

```
set vector_file [open $splitfiledir/totalout]

proc vmd_draw_arrow {mol start end} {
    # an arrow is made of a cylinder and a cone 0.5 0.6
    set middle [vecadd $start [vecscale 0.5 [vecinvert [vecsub $end $start]]]]
    set finalend [vecadd $start [vecscale 0.6 [vecinvert [vecsub $end $start]]]]
    graphics $mol cylinder $start $middle radius 0.1
    graphics $mol cone $middle $finalend radius 0.2
}

for {set i 1} { $i < 621 } {incr i } {
    set sel [atomselect top "resid $i and type CA"]
    set basex [expr [$sel get x]]
    set basey [expr [$sel get y]]
    set basez [expr [$sel get z]]
    set base "$basex $basey $basez"
    gets $vector_file line
    scan $line "%s %s %s" vecx vecy vecz
    set vec "$vecx $vecy $vecz"
    set target [vecadd $base $vec]
    vmd_draw_arrow top $base $target
}

close vector_file
```

Listing 6.15: Rendering PCA vectors using VMD *trajectory_render.tcl* script

where the *splitfiledir* is the directory where the scripts *split-evcom* and *gen-v-1-8* have been run. These two scripts generate the files necessary to run the *trajectory_render.tcl* script presented here. For further details the reader is referred to section 4.7. In the script shown here in listing 6.15 the vectors are drawn on the locations of the α-carbons. The number 621 represents the number of residues and thus α-carbons available in this structure. One shall modify the beginning of the for loop and the atom selection according to PCAs made, when using this script.

6.4 Visualization and Molecular Quantum Mechanics

In this section, we show how we visualized the results obtained from molecular quantum mechanics which was treated in detail in chapter 5. One of the principal objectives of such a calculations is to obtain the probability to find an electron in a certain region in space. Visualizing this *electron density* allows one to investigate the deformation of this probability cloud. Hence, one might visually experience physical properties such as polarization. Visualization of the *electron cloud*, the probabil-

Figure 6.5: Visualization of principal components: Here the first eight principal components according to equation (4.130) evaluated from simulations 31, 32, 33 and 34 are shown as gray arrows on the locations of the α-carbons. The EGF-like repeats are highlighted in red, N type repeats from the *Stalk* in orange, C type repeats in violet, the *Globe* in green and the calcium ions in blue. Details of the simulations are highlighted in table 4.1.

ity to find an electron in a certain region in space, has been done using the *volume rendering* technique in *Blender* [39]. In volume rendering a volume containing a voxel density can be visualized. This technique is particularly useful, for instance, in the rendering of clouds, and dates at least back to the early 1980's [144]. In chapter 5 we dealt with particular large molecules (in terms of molecular quantum mechanics). At the time when we first tried to make these kind of visualizations, volume rendering in *VMD* [37] was not able to load a voxel grid of the resolution required to obtain decent results for the large molecules we treated. Volume rendering in *PyMol* [38] was not yet existent. Hence, we turned to *Blender*.

Rendering the Electron Cloud in Blender

In order to render the electron density in *Blender* one has first to transform it into a format that *Blender* understands. *Blender* can read so called *8 bit raw* files. An *8 bit raw* file is nothing else then a file that contains nothing more then 8 bit values. Therefore it has no data specification, file header or other metadata that is stored with it. One can see such a file as an array of values ranging from 0 to 255, which is all that an 8 bit integer can hold. These files can then be used in *Blender*, where one specifies the grid data, hence how many points do exist in x, y, and z direction. These points are mapped to the data stored in the *8 bit raw* file and can influence the volume to be rendered, in many different ways. The datastructure needed to generate files that are readable by *Blender* is expressed in the C language in the following way:

```
for(index_x=0;index_x<sizex;index_x++) {
  for(index_y=0;index_y<sizey;index_y++) {
    for(index_z=0;index_z<sizez;index_z++) {
      index = index_x+index_y*sizex+index_z*sizexBYsizey;
      smalltable[index] = (char)rint(-min+bigtable[index]*rescalefactor);
    }
  }
}
for(i=0;i<num_frames;i++) {
  write(rawfile,smalltable,sizex*sizey*sizez);
}
```

Listing 6.16: Writing an *8 bit raw* file for *Blender*

Where `index_x`, `index_y` and `index_z` are the indices of the points of the grid. In the inner part of the loop a value is rescaled to fit into the 256 values that this datatype offers. In the second loop

in this script, the raw file is written to the disk. In this case, one and the same frame is written multiple times. This is to point out that one could further add to the *8 bit raw* files more and more values which are read by *Blender* in preparing the rendering of several different frames of a movie. This way one has the ability to change the shape of a cloud from frame to frame during a movie. A property that we will use later on. Different *8 bit raw* files can be used to model a voxel set of a volume to be rendered. One *8 bit raw* file can, for instance, be used to generate the density of the voxels, while an other *8 bit raw* file can be used to map different reflexion or transmission colors to different voxels, independent of the density. The reflexion and transmission colors influence the color of light scattered by the volume. The use of multiple *8 bit raw* files to model a single volume is further a feature that we might find useful later on. In order to render a single image of the electron cloud we created a script that generates such *8 bit raw files* from *Gaussian* [124] cube files that contain the electron density. As already pointed out in 5.5 a *Gaussian* cube file can be created using the *cubegen* utility from *Gaussian*. The geometry of such a *Gaussian* cube file can be defined using a small text file. This might be necessary as, other then the name indicates, *cubegen* by default creates *Gaussian* cube files that are cuboides and not cubes in their shape. A file to define the exact structure of *Gaussian* cube file looks like

```
-1,-9.068517,-7.066332,-9.784779
-500,0.03,0.0,0.0
500,0.0,0.03,0.0
500,0.0,0.0,0.03
```

Listing 6.17: Parameters to generate a cube file

The -1 parameter in the first line tells *cubegen* to create formatted files. The next tree parameters are the origin of the *Gaussian* cube file to be generated. The following three lines contain in the first value the number of points to be expressed in each direction followed by the direction vector. The minus in the second line indicates that values are expressed in atomic units. By modifying the length of the direction vector and the number of points to be expressed, we can generate *Gaussian* cube files of arbitrary resolution and ensure that these files contain a cubelike grid. Using this information a *Gaussian* cube file can be created from a *Gaussian* checkpoint file obtained from a quantum-mechanical calculation in the following way:

```
cat geometry | cubegen 0 density=scf checkpoint.fchk cubefile -1
```

Listing 6.18: Generating a cube file

where `geometry` is a textfile containing the geometry of the *Gaussian* cube file as outlined in listing 6.17, `checkpoint.fchk` is a formatted checkpoint file containing the results from a quantum-mechanical calculation and `cubefile` is the *Gaussian* cube file to be written. For the creation of checkpoint files and other information, the reader is referred to the manual shipped with the *Gaussian* software. Once the *Gaussian* cube file has been obtained, it can be converted by our *cube2raw* script into an *8 bit raw* file that is readable by *Blender*. The *cube2raw* script automatically searches for the maximum and minimum density value found in the *Gaussian* cube file and maps values to 8 bit integer values and therefore to values ranging from 0 to 255. This script can be compiled by the following command:

```
gcc -O2 cube2raw.c -o cube2raw
```

Listing 6.19: Compiling the *cube2raw* script

The script can then be run by issuing the command:

```
./cube2raw cubefile rawfile 1
```

Listing 6.20: Compiling the *cube2raw* script

where `cubefile` is a *Gaussian* cube file to be converted, `rawfile` is a *8 bit raw* file to be written. The number 1 indicates that the data should be written only once to the file. Higher numbers generate a larger file writing the same data multiple times to the file, which in general should not be necessary. In *Blender* this file is then used by a cube object for instance. The cube object in *Blender* has to be assigned to be of volume material. Then a texture is added to this material, which is of type *Voxel Data*. In the panel of the texture properties in the voxel data section one can select the file format *8 Bit RAW* and the influences this file has on the volume contained in the cube object. Further selections in *Blender* are self explaining and not further outlined here. An image of the electron density of the 8N *Stalk* repeat made with *Blender* using this method is shown in figure 6.6. We shall note that this molecule contains more then 200 atoms and that this is the first figure of such a large molecule in this representation that we have seen.

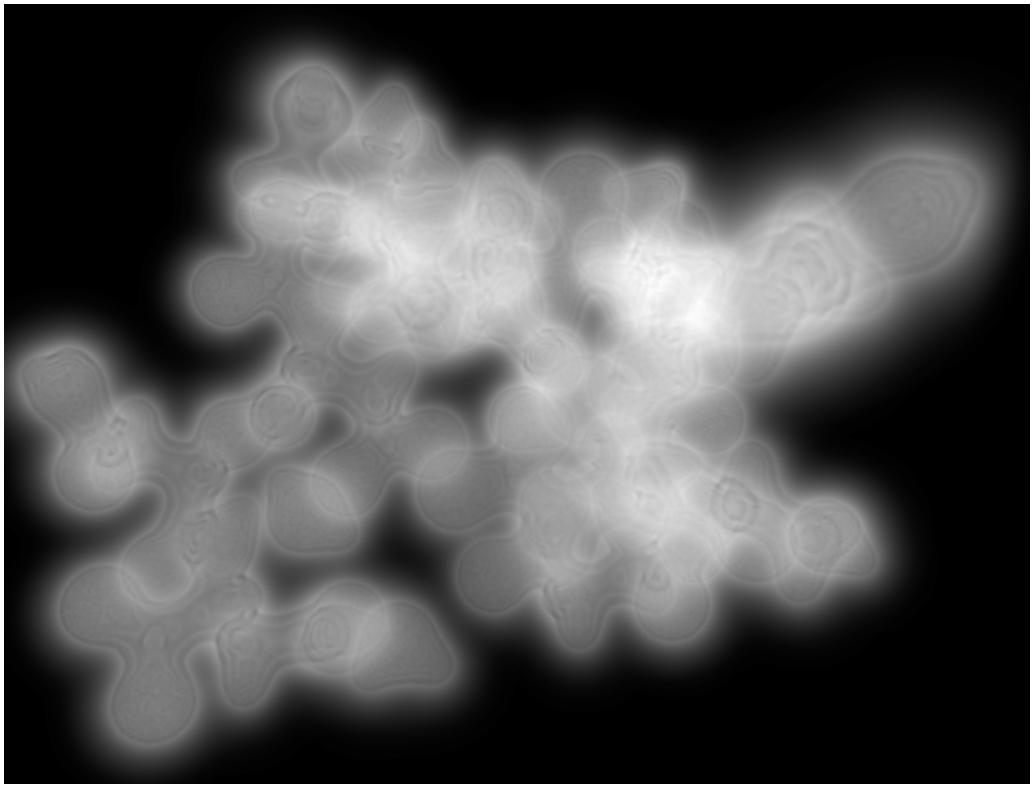

Figure 6.6: The electron cloud of the 8N repeat from the *Stalk* of the TSP SD rendered using the volume rendering techniques in *Blender* [39].

Making a Movie from Results Obtained from Molecular Quantum Mechanics

Further intrigued, we wanted to make a movie using our new tool. This should allow us to investigate visually what happens on a quantum molecularlevel if a calcium ion encounters an aspartic amino acid, a process that is crucial to understand the dynamics of the *Stalk* repeats from the TSP SD. In order to do so we calculated the wave function of an aspartic acid together with a calcium atom at different distances. These calculations where made using BHandHLYP/6-31G** theory. They are however inherently wrong as no water molecules around the calcium ions do exist and the structures were not minimized in order to make this movie. The theory and how such quantum calculations are performed is outlined in chapter 5. Anyway, we were successful in rendering our calcium-aspartic acid interaction in vacuum, and believe that renderings like this can provide insights into quantum molecular processes. Indeed one can see the electron density deform as the conformation of the calculated system changes. We will describe here the steps in order to make such a movie:

In the first place one has to generate a bunch of *Gaussian* input files for all the calculations, where the geometry of a part of the system stays the same, while an other part is moving. In our case the aspartic amino acid always stays at rest, while the calcium ion is moving from frame to frame (from *Gaussian* input file to *Gaussian* input file) to a different position. One shall be referred to the reference manual that comes with *Gaussian* how to generate such input files. All the input files run a quantum mechanical calculation in order to obtain the electron density around each molecule, and result in a separate checkpoint file from which a *Gaussian* cube file is generated. The big problem in here is that every *Gaussian* cube file by default results in a different geometry. The molecules in these files are stored at a different location in a different rotation. To overcome this issue we generated a *cubealign* script that allows us to align the systems and to generate *Gaussian* cube files of the same geometry. In order to do so we calculated the tensor of inertia, centered the non moving part, in our case the aspartic acid, and calculated a basis transformation so that the main inertial axes align with the coordinate axes of the cube. These basis then can be

used as initial vectors in the generation of the *Gaussian* cube file as outlined in listing 6.17 and hence all cubes will contain the same grid points around the non moving part, the aspartic amino acid in our case. Our *cubealign* script can be compiled on a Unix system in the following way:

```
gcc -O2 cubealign.c -lblas -llapack -L/path -o cubealign
```

Listing 6.21: Compiling the *cubealign* script on a Unix system

Here `-lblas` and `-llapack` calls the *lapack* library necessary in order to calculate the eigenvectors of the inertial tensor, hence the inertial axes. `-L/path` indicates the path where these libraries can be found. On a Mac OS X this script can be compiled using the *vecLib* framework that contains optimized *blas* and *lapack* functions.

```
gcc -O2 cubealign.c -framework vecLib -o cubealign
```

Listing 6.22: Compiling the *cubealign* script on Mac OS X

The script is then run in issuing the following command in order to generate text snippets like the ones shown in listing 6.17. These small text files serve as input files in order to create the *Gaussian* cubes in the correct basis:

```
./cubealign input.cube outformat exceptions
```

Listing 6.23: Running the *cubealign* script

Here `input.cube` is the original *Gaussian* cube file that has not undergone a coordinate transformation, `outformat` is the name of the text snippet that contains the input parameters as highlighted in listing 6.17 in order to generate the *Gaussian* cube file in the aligned coordinates. `exceptions` is a file containing the moving parts of the molecule, which in this case is the calcium ion. The format of this file is rather simple. Each line in this file stands for an atom to be excluded in the inertial axes calculation. Such a line then contains a number, which is the number of the atom to be excluded. The number of the atom is defined by the occurrence in the *Gaussian* file. Beware that the first line containing an atom in the cube file is here indexed with number 0. Using the new coordinate files created with this script for every frame for which a previously *Gaussian* cube file has been generated, a new *Gaussian* cube file in the aligned coordinates is now built. Afterwards one has to create an *8 bit raw* file that contains all the information of all the *Gaussian* cube files that have

been obtained for each frame in the movie. In order to do this we created a new script called *multicube2raw*. This was necessary as one has to find the maximum and the minimum value of a grid point over all the *Gaussian* Cube files in order to correctly convert them into 8 bit integer formats. As shown before the *8 bit raw* file can hold the voxel data for more then one frame of a movie, which was further used in this script. The *multicube2raw* script is compiled by the command:

```
gcc -O2 multicube2raw.c -o multicube2raw
```

Listing 6.24: Compiling the *multicube2raw* script

The script then can be called by the command:

```
./multicube2raw cubefile1 cubefile2 [...] cubefileN outfile.raw
```

Listing 6.25: Running the *multicube2raw* script

where `cubefile1` ...`cubefileN` are the *Gaussian* cube files that contain for instance the electron density for each frame of our movie. The parameter `outfile.raw` is the *8 bit raw* file that contains the voxel information for all of the frames of the new movie to be rendered.

In our case we did the whole procedure twice once generating an *8 bit raw* file containing the electron density, and a second *8 bit raw* file containing the electrostatic potential. The electrostatic *8 bit raw* file has then been mapped to a color gradient in *Blender* which in turn was mapped to the reflected color of a voxel in the volume to be rendered. The electron density was intuitively mapped to the volumes density. Hence, we obtained the electron cloud colored with the current electrostatic potential values. A still frame of the movie generated using this procedure can be found in figure 6.7.

At the end of this section we would further like to point out that QM simulation packages such as *CPMD* [145], without going into any details of such simulations, do support the possibility to output *Gaussian* cube files. Methods like those shown here should thus be applicable to such simulations, and we are eager to see the first movie made with *Blender* from such a type of simulation.

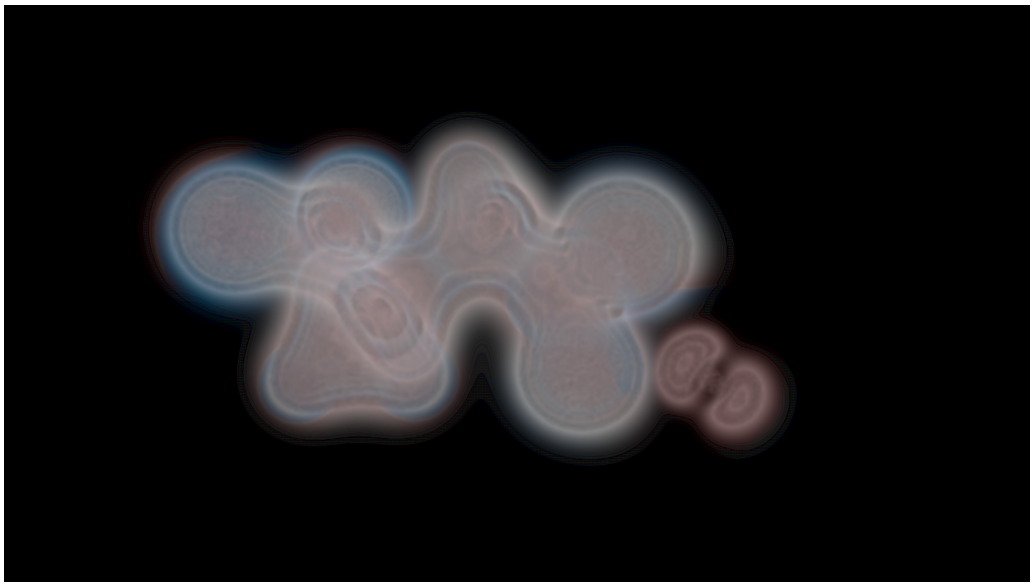

Figure 6.7: The electron cloud of a calcium- aspartic amino acid complex: Areas colored in blue indicate a negative electrostatic potential, while areas with a positive potential are indicated in red.

7 | Conclusion

During this work we have touched several different fields of science and were even capable to bring science and art a bit closer. As shown in the chapter 3, we started this work by looking at the biology of the *Thrombospondin* (TSP) molecule and more specific its *Signature Domain* (SD). We continued in performing *Molecular Dynamics* (MD) simulations as shown in chapter 4. Arriving at the limits of what is feasible in performing MD simulations due to the specifics of the TSP SD and the large number of calcium- protein interactions, we turned to molecular *Quantum Mechanics* (QM). Using molecular QM as outlined in chapter 5 we investigated the electron density of a certain region in the TSP SD and obtained preliminary results that need future work in order to be correctly analyzed and interpreted. We further developed a new algorithm to derive partial charges using new programming techniques, that include a massive parallel programming approach using OpenCL. This allowed us to run our new partial charges algorithm on different kinds of Hardware such as *Graphics Processing Units* (GPU)s. Besides all of this we developed methods in order to better visualize and hence to better understand the processes that we have investigated. Visualization as outlined in chapter 6 also plays an artistic role and is of aesthetic value. That our works have been selected not only for scientific but also artistic, or mixed artistic scientific conferences shows how far this work reaches, and how many different fields one has to treat in order to get an overview of a miniature object, a domain of a protein inside our own body. Many different things about the methods used, about the results obtained have been discussed and concluded in the different chapters.

The question one might pose now is where does this work continue and how does the future in modeling TSP and its SD look like. We would like to make some final statements here before

ending this work with some highlights of what we have created and observed:

One should know that almost all the structures of the different domains of the TSPs are known, which was in detail pointed out in chapter 3. Even tough several structures and a mechanical theoretical model are probably hard to obtain, as several modules feature significant flexibility. We think that an other way should be not like us going down from the classical MD description on an atomic level to a sub- atomic level, but to do the inverse, in going from an atomic level, to a mechanical level, where proteins are described by building blocks of continuum mechanics and to investigate the protein using such models. Even though this has not been treated in here, we believe that this work would be interesting, and might hint several results. Something completely different is the ambiguity of *Cluster of Differentiation* (CD-47) binding. We have outlined in chapter 3 that peptides from the TSP SD have been indicated to be CD-47 binding, however these binding sites seem, as outlined in chapter 4, to be buried. As this interaction is important for several biological functions we think further work should be done to enlighten these processes. Many further different things could be envisioned, but we shall end here pointing out once more the key features of this work.

The highlights of this work are probably the identification of *exchangeable* ions on the *Stalk* of the TSP SD, and the first insights into the dynamics of this large domain. This has been in detail described and discussed in chapter 4. Further we think that our new partial charges algorithm is one of the key products that have been created as a part of this work. This new approach is departing in several ways from the existing ones. And the reader shall be referred to chapter 5 for further details. However besides these we hope that this work is identified to be a contribution to many different more topics, and are interested to see what others might do with the work that has been started in many different ways in here, and where those works might lead us.

A | Annex

A.1 Presentations

Parts of this work were presented at the following congresses:

1. Société Français de Biophysique - SFB 2010: in Colle sur Loup, France with the poster: *Molecular Dynamics of Thrombospondin*
2. Groupe de Graphisme et Modélisation Moléculaire - GGMM 2011: in La Rochelle, France with the poster: *Molecular Dynamics - Thrombospondin Signature Domain*
3. Intelligent Systems for Molecular Biology - ISMB 2011: in Vienna, Austria with the poster: *Insights to the Thrombospondin Signature Domain*
4. 3DSig 2011 : in Vienna, Austria with the poster: *Dynamics of the Thrombospondine Signature Domain*
5. European Biophysics Congress - EBSA 2011: in Budapest, Hungary with the Poster: *Thrombospondin's C-Terminus*
6. Blender Conference 2011: in Amsterdam, Netherlands with the talk: *Blender in Quantum/Bio- Chemistry*
7. Rencontre des Chimistes Théoriciens Francophones - RCTF 2012: in Marseille, France with the talk: *Multicube - De la fonction d'onde aux charges partielles* and the poster: *Multicube: De la fonction d'onde aux charges partielles*
8. 3DSig 2012: in Long Beach, California USA with the poster: *The Depletion of Calcium Ions Explored by MD Simulations: A case study on Thrombospondin*

A.2 Acknowledgments

- The region of Champagne Ardenne is thanked for funding this project.
- The theses directors Manuel Dauchez and Laurent Martiny as well as the people working in the *Laboratoire de Signalisation et Récepteurs Matriciels* (SiRMa) are greatfully thanked for guiding the student of this theses and for having helped wherever possible.
- Catherine Etchebest and her team *Dynamique des Structures et Interactions des Macromolécules Biologiques* (DSIMB) are acknowledged for the great cooperation that was achieved during this project.
- Eric Henon is greatfully thanked for his scientific aid in the field of quantum chemistry.
- We thank Helmut Satzger and Andrea Weikert for their great free *Blender* courses at the *Leibniz Rechenzentrum* LRZ in Munich that we have attended in 2011.
- We are thankful for the courses of the *Chimistes Théoriciens du Grand-Est* that we have attended in 2010.
- Michael Nigels is thanked for the organization of the *European Molecular Biology Organization* (EMBO) courses at the Pasteur Institute in Paris 2010 which we attended.
- We thank *Transtec France*, the *Centre Informatique National de l'Enseignement Supérieur* (CINES), the *Centre de Calcul de Champagne Ardenne ROMEO* and the *Platforme modelisation moléculaire multiéchelle* (PM3) which have gratefully dedicated computational time on their machines to our projects.
- Finally the author thanks his family, friends and all those who provided discussions, social and every other kind of imaginable support that have helped to realize this thesis.

B | Bibliography

[1] Baenziger NL, Brodie GN, Majerus PW. A Thrombin-Sensitive Protein of Human Platelet Membranes. Proceedings of the National Academy of Sciences. 1971;68(1):240--243.

[2] Adolph KW. A thrombospondin homologue in Drosphilia melanogaster cDNA and protein structure. Gene. 2001;269:177--184.

[3] Gutierrez LS. The Role of Thrombospondin 1 on Intestial Inflammation and Carcinogenesis. Biomarkers Insights. 2008;3:171--178.

[4] Lopez-Dee Z, Pidcock K, Gutierrez LS. Thrombospondin-1: Multiple Paths to Inflammation. Mediators of Inflammation. 2011;2011(296069).

[5] Ichii T, Koyama H, Tanaka S, Shioi A, Okuno Y, Otani S, et al. Thrombospondine-1 Mediates Smooth Muscle Cell Proliferation Induced by Interaction With Human Platelets. Arterisclerosis, Thrombosis and Vascular Biology. 2002;22:1286--1292.

[6] Lopez N, Gregg D, Vasudevan S, Hassanain H, Goldschmidt-Clermont P, Kovacic H. Thrombospondin 2 Regulates Cell Proliferation Induced by Rac1 Redox-Dependent Signaling. Molecular and Cellular Biology. 2003;23(15):5401--5408.

[7] Bornstein P. Thrombospondins as matricellular modulators of cell function. Journal of Clinical Investigation. 2001;107(8):929--934.

[8] Bornstein P, Agah A, Kyriakides TR. The role of thrompospondines 1 and 2 in the regulation of cell-matrix interactions, collagen fibril formation and the response to

injoury. International Journal of Biochemistry AND Cell Biology. 2004;36:1115--1125.

[9] Iruela-Arispe ML, Luque A, Lee N. Thrombospondin modules and angiogenesis. International Journal of Biochemistry and Cell Biology. 2004;36:1070--1078.

[10] Lawler PR, Lawler J. Molecular Basis for the Regulation of Angiogenesis by Thrombospondin-1 and -2. Cold Spring Harbor Perspectives in Medicine. 2012;2(5):a006627.

[11] Yang Q, Tian Y, Liu S, Zeine R, Chlenski A, Salwen HR, et al. Thrombospondin-1 Peptide ABT-510 Combined with Valproic Acid Is an Effective Antiangiogenesis Strategy in Neuroblastoma. Cancer Research. 2007;67:1716--1724.

[12] Rath GM, Schneider C, Dedieu S, Rothhut B, Soula-Rothhut M, Ghoneim C, et al. The C-terminal CD47/IAP-binding domain of thrombospondin-1 prevents camptothecin- and doxorubicin- induced apoptosis in human thyroid carcinoma cells. Biochemica et Biophysica Acta. 2006;1763:1125--1134.

[13] Li F, Cao Y, Townsend CM, Ko TC. TGF-beta Signaling in Colon Cancer Cells. World Journal of Surgery. 2005;29(3):306--311.

[14] Guo N, Krutzsch HC, Inman JK, Roberts D. Thrombospondin 1 and Type I Repeat Peptides of Thrombospondin 1 Specifically Induce Apoptosis of Endothelial Cells. Cancer Research. 1997;57:1735--1742.

[15] Briggs MD, Chapman KL. Pseudoachondroplasia and Multiple Epiphyseal Dysplasia: Mutation Review, Molecular Interactions and Genotype to Phenotype Correlations. Human Mutation. 2002;19:465--478.

[16] Iruela-Arispe ML, Liska DJ, Sage EH, Bornstein P. Differential Expression of Thrombospondine 1, 2 and 3 During Murine Development. Developmental Dynamics. 1993;197:40--56.

[17] Christopherson KS, Ullian EM, Stokes CCA, Mullowney CE, Hell JW, Agah A, et al. Thrombospondins Are Astrocyte-Secreted Proteins that Promote CNS Synaptogenesis. Cell. 2005;120:421--433.

[18] Li Y, Tong X, Rumala C, Clemons K, Wang S. Thrombospondin 1 Deficiency Reduces Obesity-Associated Inflammation and Improves Insulin Sensitivity in Diet Induced Obese Mouse Model. PLoS ONE. 2011;6(10):e22656.

[19] Ebbinghaus S, Hussain M, Tannir N, Gordon M, Desai AA, Knight RA, et al. Phase 2 Study of ABT-510 in Patients with Previously Untreated Advanced Renal Cell Carcinoma. Clinical Cancer Research. 2007;13:6689--6695.

[20] Nabors LB, Fiveash JB, Markert JM, Kekan MS, Gillespie GY, Huang Z, et al. A Phase 1 Trial of ABT-510 Concurrent With Standard Chemoradiation for Patients with Newly Diagnosed Glioblastoma. Archives of Neurology. 2010;67(3):313--319.

[21] Rusk A, McKeegan E, Haviv F, Majest S, Henkin J, Khanna C. Preclinical Evaluations of Antiangiogenic Thrombospondin-1 Peptide Mimetics, ABT-526 and ABT-510, in Companion Dogs with Naturally Occurring Cancers. Clinical Cancer Research. 2006;12:7444--7455.

[22] Rusk A, Cozzi E, Stebbins M, Vail D, Graham J, Valli V, et al. Cooperative Activity of Cytotoxic Chemotherapy with Antiangiogenic Thrombospondin-1 Peptides, ABT-526 in Pet Dogs with Relapsed Lymphoma. Clinical Cancer Research. 2006;12:7456--7464.

[23] Garside SA, Henkin J, Morris KD, Norvell SM, Thomas FH, Fraser HM. A Thrombospondin-Mimetic Peptide, ABT-898, Suppreses Angiogenesis and Promotes Follicular Atresia in Pre- and Early-Antral Follices in Vivo. Endocrinology. 2010;151(12):5905--5915.

[24] Kosfeld MD, Frazier WA. Identification of a New Cell Adhesion Motif in Two Homologous Peptides from the COOH-terminal Cell Binding Domain of Human Thrombospondin. The Journal of Biological Chemistry. 1993;268(12):8808--8814.

[25] Lawler J, Wienstein R, Hynes RO. Cell Attachment to Thrombospondin: The Role of ARG-GLY-ASP, Calcium, and Integrin Receptors. The Journal of Cell Biology. 1988;107(6):2351--2361.

[26] Lymn JS, Patel MK, Clunn GF, Sarafina SJ, Gallagher KL, Huges AD. Thrombospondin-1 differentially induces chemotaxis and DNA synthesis of human venous smooth muscle cells at the receptor-binding level. Journal of Cell Science. 2002;115:4353--4360.

[27] Isenberg JS, Martin-Manso G, Maxhimer JB, Roberts DD. Regulation of nitric oxide signaling by thrombospondin-1: Implications for anti-angiogenic therapies. Nature Reviews Cancer. 2009;9(3):182--194.

[28] Adler BJ, Wainwright TE. Studies in Molecular Dynamics. I. General Method. The Journal of Chemical Physics. 1959;31:459--461.

[29] Shaw DE, Deneroff MM, Dror RO, Kuskin JS, Larson RH, Salmon JK, et al. Anton, a Special-Purpose Machine for Molecular Dynamics Simulation. Communications of the ACM. 2008;51(7):91--97.

[30] Hess B, Kutzner C, van der Spoel D, Lindahl E. Gromacs 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation. Journal of Chemical Theory and Computation. 2008;4(3):435--447.

[31] Philipps JC, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, et al. Scalable Molecular Dynamics with NAMD. Journal of Computational Chemistry. 2005;26:1781--1802.

[32] Annis DS, Gunderson KA, Mosher DF. Immunochemical Analysis of the Structrue of the Signature Domains of Thrombospondin-1 and Thrombospondin-2 in Low Calcium Concentrations. Journal of Biological Chemistry. 2007;282(37):27067--27075.

[33] Singh UC, Kollman PA. An approach to computing electrostatic charges for molecules. Journal of Computational Chemistry. 1984;5(2):129--145.

[34] Mulliken RS. Electronic Population Analysis on LCAO-MO Molecular Wave Functions. Journal of Chemical Physics. 1955;23:1833--1840.

[35] Kendrew JC, Bodo G, Dintzis HM, Parrish RG, Wyckoff H, Phillips DC. A Three-Dimensional Model of the Myoglobin Molecule Obtained by X-Ray Analysis. Nature. 1958;181:662--666.

[36] Levinthal C. Molecular Model-building by Computer. Scientific American. 1966;214(6):42--52.

[37] Humphrey W, Dalke A, Schulten K. Visual Molecular Dynamics. Journal of Molecular Graphics. 1996;14:33--38.

[38] Schrödinger, LLC. PyMOL; 2010. The PyMOL Molecular Graphics System.

[39] The Blender Foudation, Amsterdam et al. Blender; 2011. Blender 3D version 2.49 and 2.53.

[40] Sid B, Sartelet H, Bellon G, Btaouri HE, Rath G, Delorme N, et al. Thrombospondin 1: a multifunctional protein implicated in teh regulation of tumor growth. Critical Reviews in Oncology Hematology. 2004;49:245--258.

[41] Adams JC, Monk R, Taylor AL, Ozbek S, Fascetti N, Baumgartner S, et al. Characterisation of Drosophilia Thrombospondin Defines an Early Origin of Pentameric Thrombospondins. Journal of Molecular Biology. 2003;328:479--494.

[42] Sweetwyne MT, Pallero MA, Lu A, Graham LVD, Murphy-Ullrich JE. The Calreticulin-Binding Sequence of Thrombospondin 1 Regulates Collagen Expression and Organization During Tissue Remodeling. The American Journal of Pathology. 2010;177(4):1710--1724.

[43] Calzada MJ, Sipes JM, Krutzsch HC, Yurchenco PD, Annis DS, Mosher DF, et al. Recognition of the N-terminal Modules of Thrombospondin-1 and Thrombospondin-2 by α6β1 Integrin. The Journal of Biological Chemistry. 2003;278(42):40679--40687.

[44] Calzada MJ, Zhou L, Sipes JM, Zhang J, Krutzsch HC, Iruela-Astripe ML, et al. α4β1 Integrin Mediates Selective Entothelial Cell Responses to Thrombospondin 1 and 2 in Vitro and Modulates Angiogenesis in Vivo. Circulation Research. 2004;94:462--470.

[45] Krutzsch HC, Choe BJ, Sipes JM, Guo N, Roberts DD. Identification of an α3β1 Integrin Recognition Sequence in Thrombospondin-1. The Journal of Biological Chemistry. 1999;274(34):24080--24086.

[46] Li Z, Calzada MJ, Sipes JM, Cashel JA, Krutzsch HC, Annis DS, et al. Interactions of thrombospondins with α4β1

integrin and CD47 differentially modulate T cell behavoir. The Journal of Cell Biology. 2002;157(3):509--519.

[47] Voland C, Serre CM, Delmas P, Clézardin P. Platelet-Osteosarcoma Cell Interaction is Mediated Through a Specific Fibrinogen-Binding Sequence Located Within the N-Terminal Domain of Thrombospondin 1. Journal of Bone and Mineral Research. 2000;15(2):361--368.

[48] Tan K, Duquette M, Liu J, Zhang R, Joachimiak A, Wang J, et al. The Structures of the Thrombospondin-1 N-terminal Domain and Its Complex with a Synthetic Pentameric Heparin. Structure. 2006;14:33--42.

[49] Tan K, Duquette M, Liu J, Shanmugasundaram K, Joachimiak A, Gallagher JT, et al. Heparin-induced cis- and trans-Dimerization Modes of Thrombospondin-1 N-termina Domain. The Journal of Biological Chemistry. 2008;283(7):3932--3941.

[50] Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The Protein Data Bank. Nucleic Acids Research. 2000;28(1):235--242.

[51] Larsen CB, Lawler J, Mosher DF. Structures of thrombospondins. Cell Mol Life Sci. 2008;65:672--686.

[52] Malashkevich VN, Kammerer RA, Efimov VP, Engel J. The Crystal Structure of a Five-Stranded Coiled Coil in COMP: A Protorype Ion Channel? Science Magazine. 1996;274(5288):761--765.

[53] Guo Y, Bozic D, Malashkevich VN, Kammerer RA, Schulthess T, Engel J. All-trans retinol, vitamin D and other hydrophobic compounds bind in the axial pore of the five-stranded coiled-coil domain of cartilage oligomeric matrix protein. The EMBO Journal. 1998;17(18):5265--5272.

[54] Özbek S, Engel J, Stetefeld J. Storage function of cartilage oligomeric matrix protein: the crystal structure of the coiled-coil domain complex with vitamin D3. The EMBO Journal. 2002;21(22):5960--5968.

[55] Misenheimer TM, Huwiler KG, Annis DS, Mosher DF. Physical Characterization of the Procollagen Module of Human Thrombospondin 1 Expressed in Insect Cells. The Journal of Biological Chemistry. 2000;275(52):40938--40945.

[56] Lopez-Dee ZP, Chittur SV, Patel B, Stanton R, Wakeley M, Lippert B, et al. Thrombospondin-1 Type 1 Repeats in a Model of Inflammatory Bowel Disease: Transcript Profile and Therapeutic Effects. PLoS One. 2012;7(4):e34590.

[57] Tan K, Duquette M, Liu J, Dong Y, Zhang R, Joachimiak A, et al. Crystal structure of the TSP-1 type 1 repeats a novel layered fold and its biological implication. The Journal of Cell Biology. 2002;159(2):373--382.

[58] Bein K, Simons M. Thrombospondin Type 1 Repeats Interact with Matrix Metalloproteinase 2. The Journal of Biological Chemistry. 2000;275(41):32167--32173.

[59] Calzada MJ, Annis DS, Zeng B, Marcinkiewicz C, Banas B, Lawler J, et al. Indentification of Novel β1 Integrin Binding Sites in the Type 1 and Type 2 Repeats of Thrombospondin 1. The Journal of Biological Chemistry. 2004;279(40):41734--41743.

[60] Zhang X, Kazerounian S, Duquette M, Perruzzi C, Nagy JA, Dvorak HF, et al. Thrombospondin-1 modulates vascular endothelial growth factor activity at the receptor level. The FASEB Journal. 2009;23:3368--3376.

[61] Asch AS, Silbiger S, Heimer E, Nachman RL. Thrombospondin sequence motif (CSVTCG) is responsible for CD-36 binding. Biochemical and Biophysical Research Communications. 1992;182(3):1208--1217.

[62] Miao W, Seng WL, Duquette M, Lawler P, Laus C, Lawler J. Thrombospondin-1 Type 1 Repeat Recombinant Proteins Inhibit Tumor Growth through Transforming Growth Factor-β -dependent and -independent Mechanisms. Cancer Research. 2001;61:7830--7839.

[63] de Peredo AG, Klein D, Macek B, Hess D, Peter-Katalinic J, Hofsteenge J. C-Mannosylation and O-Fucosylation of Thrombospondin Type 1 Repeats. Molecular and Cellular Proteomics. 2002;1:11--18.

[64] Misenheimer TM, Hannah BA, Annis DS, Fischer DF. Interactions among the Three Structural Motifs of the C-Terminal Region of Human Thrombospondin-2. Biochemistry. 2003;42:5125--5132.

[65] Lawler J, Chao FC, Cohen CM. Evidence for Calcium-sensitive Structure in Platelet Thrombospondin. The Journal of Biological Chemistry. 1982;257(20):12257--12265.

[66] Lawler J, Derick LH, Connolly JE, Chen J, Chao FC. The Structure of Human Platelet Thrombospondin. The Journal of Biological Chemistry. 1985;260(6):3762--3772.

[67] Edgar RC. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. BMC Bioinformatics. 2004;5(113).

[68] Kvansakul M, Adams JC, Hohenester E. Structure of a thrombospondin C-terminal fragment reveals a novel calcium core in type 3 repeats. EMBO Journal. 2004;23:1223--1233.

[69] Carlson CB, Bernstein DA, Annis DS, Misenheimer TM, Hannah BA, Mosher DF, et al. Structure of the calcium-rich signature domain of humain thrombospondin-2. Nature Structural and Molecular Biology. 2005;12(10):910--914.

[70] Carlson CB, Lui Y, Keck JL, Mosher DF. Influences of the N700S Thrombospondin-1 Polymorphism of Protein Structure and Stability. Journal of Biological Chemistry. 2008;283(29):20069--20076.

[71] Tan K, Duquette M, Joachimiak A, Lawler J. The crystal structure of the signature domain of cartilage oligomeric matrix protein: implications for collagen, glycosaminoglycan and integrin binding. FASEB Journal. 2009;23:2490--2501.

[72] Mann HH, Özbek S, Engel J, Paulsson M, Wagener R. Interactions between the Cartilage Oligomeric Matrix Protein and Matrilins. The Journal of Biological Chemistry. 2004;279(24):25294--25298.

[73] Rosenberg K, Olsson H, Mörgelin M, Heinegård D. Cartilage Oligomeric Matrix Protein Shows High Affinity Zinc-dependent Interaction with Triple Helical Collagen. The Journal of Biological Chemistry. 1998;273(32):20397--20403.

[74] Holden P, Meadows RS, Chapman KL, Grant ME, Kadler KE, Briggs MD. Cartilage Oligomeric Matrix Protein Interacts with Type IX Collagen, and Disruptions to These

Interactions Identify a Pathogenetic Mechanism in a Bone Dysplasia Family. The Journal of Biological Chemistry. 2001;276(8):6046--6055.

[75] Agarwal P, Zwolanek D, Keene DR, Schulz J, Blumbach K, Heinegård D, et al. Collagen XII and XIV, New Partners of Cartilage Oligomeric Matrix Protein in the Skin Extracellular Matrix Suprastructure. The Journal of Biological Chemistry. 2012;287(27):22549--22559.

[76] Topol EJ, McCarthy J, Gabriel S, Moliterno DJ, Rogers WJ, Newby LK, et al. Single Nucleotide Polymorphisms in Multiple Novel Thrombospondin Genes May Be Associated With Familial Premature Myocardial Infarction. Circulation. 2001;104:2641--2644.

[77] Isenberg JS, Maxhimer JB, Hyodo F, Pendrak ML, Ridnour LA, DeGraff WG, et al. Thrombospondin-1 and CD47 Limit Cell and Tissue Survival of Radiation Injury. The American Journal of Pathology. 2008;173(4):1100--1112.

[78] Isenberg JS, Annis DS, Pendrak ML, Ptaszynska M, Frazier WA, Mosher DF, et al. Differential Interactions of Thrombospondin-1,-2, and -4 with CD47 and Effects on cGMP Signaling in Ischemic Injury Responses. The Journal of Biological Chemistry. 2009;284(2):1116--1125.

[79] Jorgensen WL, Tirado-Rives J. The OPLS Potential Functions for Proteins. Energy Minimizations for Crystals of Cyclic Peptides and Crambin. Journal of the American Chemical Society. 1988;110(6):1657--1666.

[80] Jorgensen WL, Maxwell DS, Tirado-Rives J. Development and Testing of the OPLS All Atom Force Field on Conformational Energetics and Properties of Organic Liquids. Journal of the American Chemical Society. 1996;118:11225--11236.

[81] Oostenbrink C, Villa A, Mark AE, van Gusteren WF. A Biomolecular Force Field Based on Free Enthalphy of Hydration and Solvation: The GROMOS Force Field Parameter Sets 53A5 and 53A6. Journal of Computational Chemistry. 2004;25:1656--1676.

[82] Project E, Nachliel E, Gutman M. Parametrization of Ca+2-Protein Interactions for Molecular Dynam-

ics Simulations. Journal of Computational Chemistry. 2007;29:1163--1169.

[83] Goldstein H. Classical Mechanics. 2nd ed. Addison Wesley; 1980.

[84] Meyberg K, Vachenauer P. Höhere Mathematik 2: Differentialgleichungen, Funktionentheorie, Fourier-Analysis, Variationsrechnung. 4th ed. Springer Verlag Berlin Heidelberg; 2003.

[85] Jänich K. Mathematik 2: Geschrieben für Physiker. Springer Verlag Berlin Heidelberg; 2002.

[86] Ryckaert J, Ciccotti G, Berendsen HJC. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkans. Journal of Computational Physics. 1977;23(3):327--341.

[87] Hess B, Bekker H, Berendsen HJC, Fraaije JGEM. LINCS: A Linear Constraint Solver for Molecular Simulations. Journal of Computational Chemistry. 1997;18(12):1463--1472.

[88] Jackson JD. Classical Electrodynamcs. 3rd ed. Wiley; 1998.

[89] Greiner W. Klassische Elektrodynamik. sixth ed. Wissenschaftlicher Verlag Harri Deutsch, Frankfurt am Main; 2002.

[90] Darden T, York D, Pedersen L. Particle mesh Ewald: An N.log(N) method for Ewald sums in large systems. Journal of Chemical Physics. 1993;98(12):10089--10092.

[91] Tironi IG, Sperb R, Smith PE, van Gunsteren WF. A generalized reaction field for molecular dynamics simulations. Journal of Chemical Physics. 1994;102(13):5451--5459.

[92] Ewald PP. Die Berechnungen optischer und elektrostatischer Gitterpotentiale. Annalen der Physik. 1921;369(3):253--287.

[93] Nolting W. Grundkurs Theoretische Physik 6: Statistische Physik. 5th ed. Springer Verlag Berlin Heidelberg; 2005.

[94] Landau LD, Lifschitz EM. Lehrbuch der theoretischen Physik V: Statistische Physik Teil 1. eighth ed. Akademie-Verlag Berlin; 1987.

[95] Reichl LE. A Modern Course in Statistical Physics. 2nd ed. Wiley; 1998.

[96] Berendsen HJC, Postma JPM, van Gunsteren WF, DiNola A, Haak JR. Molecular dynamics with coupling to an external bath. Journal of Chemical Physics. 1984;81(8):3684--3690.

[97] Bussi G, Donaldio D, Parinello M. Canonical Sampling through velocity rescaling. Journal of Chemical Physics. 2007;126(014101).

[98] Parrinello M, Rahman A. Polymorphic transitions in single crystals: A new molecular dynamics mathod. Journal of Applied Physics. 1981;52(12):7182--7190.

[99] Berendsen HJC, Postma JPM, van Gusteren WF, Hermans J. Interaction Models for Water in Relation to Protein Hydration. In: Pullman B, editor. Intramolecular Forces. D. Reidel Publishing; 1981. p. 331--342.

[100] Jorgensen WL, Chandrasekhar J, Madura JD, Impey RW, Klein ML. Comparison of simple potential functions for simulating liquid water. Journal of Chemical Physics. 1983;79:926--935.

[101] Klinke R, Silbernagel S, editors. Lehrbuch der Physiologie. 4th ed. Georg Thieme Verlag Stuttgart; 2003.

[102] Byrd RH, Lu P, Nocedal J, Zhu C. A Limited Memory Algorithm for Bound Constrained Optimization. SIAM Journal on Scientific Computing. 1995;16(5):1190--1208.

[103] Wolfram Research Inc . Mathematica; 2008. Mathematica: Version 7.0, Wolfram Research, Inc.

[104] Kabsch W, Sander C. Dictonary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers. 1983;22:2577--2637.

[105] Izenman AJ. Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning. Springer; 2008.

[106] Lightstone FC, Schwegler E, Allesch M, Gygi F, Galli G. A First-Principles Molecular Dynamics Study of Calcium in Water. ChemPhysChem. 2005;6:1745--1749.

[107] Ikeda T, Boero M, Terakura K. Hydration properties of magnesium and calcium ions from constrained first prin-

ciples molecular dynamics. Journal of Chemical Physics. 2007;127(074503).

[108] Williams T, Kelley C, Lang R, Kotz D, Campbell J, Elber G, et al. Gnuplot; 2011. The gnuplot program version 4.4.

[109] Planck M. Zur Theorie des Gesetzes der Energieverteilung im Normalspectrum. Verhandlungen der Deutschen Physikalischen Gesellschaft. 1900;2(17):237--245.

[110] Demtröder W. Experimentalphysik 3 Atome, Molekühle und Festkörper. 2nd ed. Springer Verlag Berlin Heidelberg; 2000.

[111] Schrödinger E. Quantisierung als Eigenwertproblem. Annalen der Physik. 1926;384:361--376.

[112] Messiah A. Quantum Mechanics Two Volumes Bound as One. Dover Publications; 1999.

[113] Atkins P, Friedman R. Molecular Quantum Mechanics. 5th ed. Oxford University Press; 2010.

[114] Koch W, Holthausen M. A Chemist's Guide to Density Functional Theory. 2nd ed. Wiley; 2001.

[115] Roothaan CCJ. New Developments in Molecular Orbital Theory. Reviews of Modern Physics. 1951;23:69--89.

[116] Hall GG. The Molecular Orbital Theory of Chemical Valency. VIII. A method of Calculating Ioniazation Potentials. Proceedings of the Royal Society A. 1951;205(1081):541--552.

[117] Hohenberg P, Kohn W. Inhomogeneous Electron Gas. Physical Review. 1964;136(3B):B864--B871.

[118] Kohn W, Sham LJ. Self-Consistent Equations Including Exchange Correlation Effects. Physical Review. 1965;140(4A):A1133--A1138.

[119] Becke AD. Density-functional thermochemistry. III The role of exact Exchange. Journal of Chemical Physics. 1993;98:5648--5652.

[120] Lee C, Yang W, Parr RG. Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density. Physical Review B. 1988;37:785--789.

[121] S H Vosko LW, Nusair M. Accurate spin dependent electron liquid correlation energies for spin density calculation: a critical analyses. Candadian Journal of Physics. 1980;58(8):1200--1211.

[122] Stephens PJ, Devlin FJ, Chabalowski CF, Frisch MJ. Ab Inito Calculation of Vibrational Absorption and Circular Diochroism Spectra Using Density Functional Force Fields. The Journal of Physical Chemistry. 1994;98(45):11632--11627.

[123] Becke AD. A new mixing of Hartree- Fock and local density-functional theories. Journal of Chemical Physics. 1993;98:1372--1377.

[124] Frisch MJ, Trucks GW, Schlegel HB, Scuseria GE, Robb MA, Cheeseman JR, et al.. Gaussian 09 Revision A.1;. Gaussian Inc. Wallingford CT 2009.

[125] YU W, Liang L, Lin Z, Ling S, Haranczyk M, Gutowski M. Comparison o Some Representative Density Functional Theory and Wave Function Theory Methods for Studies of Amino Acids. Journal of Computational Chemistry. 2008;30(4):589--600.

[126] Cieplak P, Cornell WD, Bayly C, Kollman PA. Application of the Multimolecule and Multiconformational RESP Methodology to Biopolymers: Charge Derivation of DNA, RNA and Proteins. Journal of Computational Chemistry. 1995;16(11):1357--1377.

[127] Reynolds CA, Essex JW, Richards WG. Atomic Charges for Variable Molecular Conformations. Journal of the American Chemical Society. 1991;114:9075--9079.

[128] Comell WD, Cieplak P, Bayly CI, Kollman PA. Application of RESP Charges To Calculate Conformational Energies, Hydrogen Bond Energies, and Free Energies of Solvation. Journal of the American Chemical Society. 1993;115:9620--9631.

[129] Spackman MA. Potential Derived Charges Using a Geodesic Point Selection Scheme. Journal of Computational Chemistry. 1996;17(1):1--18.

[130] Breneman CM, Wiberg KB. Determining Atom-Centered Monopoles from Molecular Electrostatic Potentials. The

Need for High Sampling Density in Formamide Conformational Analysis. Journal of Computational Chemistry. 1990;11(3):361--373.

[131] Chipot C, Angyan J, Millot C. Statistical analyses of distributed multipoles derived from molecular electrostatic potentials. Molecular Physics. 1998;96(6):881--895.

[132] Thompson JD, Xidos JD, Sonbuchner TM, Cramer CJ, Truhlar DG. More Reliable Partial Atomic Charges When Using Diffuse Basis Sets. PhysChemComm. 2002;5:117--134.

[133] of the IEEE Computer Society MSC. IEEE Standard for Floating-Point Arithmetic. IEEE Standard 754-2008. 2008;p. 1--58.

[134] Bondi A. van der Waals Volumes and Radii. The Journal of Physical Chemistry. 1964;68(3):441--451.

[135] Mantina M, Chamberlin AC, Valero R, Cramer CJ, Truhlar DG. Consisten van der Waals Radii for the Whole Main Group. The Journal of Physical Chemistry A. 2009;p. 5806--5812.

[136] Schaftenaar G, Noordik JH. Molden: a pre- and post-processing program for molecular and electronic structures. Journal of Computer-Aided Molecular Design. 2000;14:123--134.

[137] Wang B, Truhlar DG. Including Charge Penetration Effects in Molecular Modelling. Journal of Chemical Theroy and Computation. 2010;6:3330--3342.

[138] Wang B, Truhlar DG. Partial Atomic Charges and Screened Charge Models for the Electrostatic Potential. Journal of Chemical Theory and Computation. 2012;8:1989--1998.

[139] NIST Computational Chemistry Comparison and Benchmark Database; 2011. NIST Standard Reference Database Number 101.

[140] Kajiya JT. The Rendering Equation. Siggraph. 1986;20(4):143--150.

[141] Stone JE. Master Theses: An Efficient Library for Parallel Raytracing and Animation. University of Missouri-Rolla; 1998.

[142] Johnson GT, Autin L, Goodsell DS, Sanner MF, Olson AJ. ePMV Embeds Molecular Modeling into Professional Animation Software Environements. Structure. 2011;19:293--303.

[143] Zoppe M, Porozov Y, Andrei R, Cianchetta S, Zini MF, Loni T, et al. Using Blender for molecular animation and scientific representation. In: Blender Conference; 2008. .

[144] Kajiya JT. Ray Tracing Volume Densities. Computer Graphics. 1984;18(3):165--174.

[145] IBM Corp, MPI für Festkörperforschung Stuttgart. CPMD; 2008. Copyright MPI für Festkörperforschung Stuttgart 1997-2001,Copyright IBM Corp 1990-2008.

Molecular Modelization of Matrikines and Matrix Proteins: Theoretical Approaches

Abstract:

This document deals with the various aspects of molecular modeling of a protein in the extracellular matrix. In this work the thrombospondin signature domain is studied in detail, using various methods from different fields of science. Due to the complexity of the protein, a classical molecular dynamics and a quantum molecular approach is chosen to gain insights into the thrombospondin signature domain. During this process a new algorithm that allows to derive partial charges from *ab-initio* calculations has been developed. In order to run this algorithm in an efficient way new programming techniques, such as massive multiparallel programming on graphics processors has been used. Further new visualization methods have been either tried or developed effectively generating a new kind of art from the scientific results that have been obtained in this project. Besides these developments several interesting insights into the dynamics of thrombospondins have been revealed. We were able to identify the exchangeable ions within the thrombospondin signature domain, and were further capable to verify and extend existing models for a low calcium signature domain structure. Binding sites important for cancer drug design have also been evaluated using the molecular dynamics simulations.

Keywords: Thrombospondin, Molecular Modeling, Molecular Dynamics, Molecular Quantum Mechanics, Rendering, Visualization

Ce document traite des aspects différents de la modélisation moléculaire d'une protéine de la matrice extracellulaire. Le domaine de signature des thrombospondines est étudié en détail dans ce document en faisant appel aux champs variés de la science. La complexité de ce domaine nécessite l'utilisation des approches de la mécanique quantique moléculaire en plus de la dynamique moléculaire classique. En reliant les deux approches un nouveau algorithme a été développé afin d'attribuer des charges partielles à partir des résultats obtenus par des méthodes *ab-initio.* Pour implémenter cet algorithme d'une façon efficace, les nouvelles méthodes de la programmation massivement parallèle ont été utilisées qui nous permettent entre autres de faire tourner cet algorithme sur les processeurs graphiques. Nous avons également utilisé et développé des nouvelles méthodes de la visualisation moléculaire rapprochant l'art à la science. En plus, de ces développements nous avons obtenus des résultats intéressants sur la dynamique du domaine de signature. Nous avons pu identifier les ions échangeables de ce domaine. Nous avons vérifié et élargie les modèles existants de la thrombospondine dans des concentrations faibles de calcium. En utilisant la dynamique moléculaire nous avons également évalué des sites importants de fixations pour le développement de nouveaux médicaments contre le cancer.

Mots-cléfs: Thrombospondine, Modélisation Moléculaire, Dynamique Moléculaire, Mécanique Quantique Moléculaire, Rendu, Visualisation

Laboratoire SiRMA, CNRS FRE 3481 MEDyC Université de Reims Champagne-Ardennes, Pôle Sciences, Moulin de la Housse - BP1039, 51687 Reims Cedex 2