



HAL
open science

Modélisation gros grains et simulation multi-agents - Application à la membrane interne mitochondriale

Charles Lales

► **To cite this version:**

Charles Lales. Modélisation gros grains et simulation multi-agents - Application à la membrane interne mitochondriale. Modélisation et simulation. Université Sciences et Technologies - Bordeaux I, 2007. Français. NNT: . tel-00452853

HAL Id: tel-00452853

<https://theses.hal.science/tel-00452853>

Submitted on 3 Feb 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

PRÉSENTÉE À

L'UNIVERSITÉ BORDEAUX I

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET
D'INFORMATIQUE

Par **Charles Lales**

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

**Modélisation gros grains et simulation multi-agents
Application à la membrane interne mitochondriale**

Soutenue le : 13 décembre 2007

Après avis des rapporteurs :

Daniel BORGIS Professeur
Frédéric GUINAND ... Professeur

Devant la commission d'examen composée de :

D. BORGIS	Professeur	Rapporteur
F. GUINAND	Professeur	Rapporteur
P. BALLET	Maître de Conférences	Examineur
M. BEURTON-AIMAR	Maître de Conférences	Co-Directeur
O. COULAUD	Directeur de Recherche	Examineur
J.-P. MAZAT	Professeur	Co-Directeur
R. STRANDH	Professeur	Directeur

Remerciements

Mes remerciements s'adressent en premier lieu à mon directeur de thèse Robert Strandh, ainsi qu'à mes encadrants, Marie Beurton-Aimar et Jean-Pierre Mazat. Ils ont su me faire largement bénéficier de leur expertise dans leur domaine respectif. Je les remercie pour l'environnement de travail riche et dynamique qu'ils m'ont proposé de rejoindre il y a trois ans. Je leur suis également grandement reconnaissant pour leurs qualités humaines et la patience dont ils ont fait preuve à mon égard.

Je tiens à remercier chaleureusement Daniel Borgis et Frédéric Guinand pour m'avoir fait l'honneur de lire attentivement mon mémoire. Je tiens à exprimer ma gratitude pour l'intérêt qu'ils ont porté à l'égard de mon travail.

Je remercie Olivier Coulaud d'avoir accepté de présider mon jury, ainsi que Robert Strandh et Pascal Ballet pour avoir accepté de prendre part à ce jury.

Je tiens également à remercier :

- tous ceux du LaBRI et du laboratoire de physiopathologie mitochondriale pour la remarquable ambiance de travail qu'ils ont tous contribué à créer et la motivation à toujours pousser plus loin la critique du travail effectué pour n'avoir de cesse de vouloir faire mieux.*
- toutes les membres du GT SMABio pour les échanges toujours agréables et enrichissants, plus particulièrement Pascal Ballet pour ces conseils extérieurs et avisés ainsi que Guillaume Beston qui me fit découvrir les SMA il y a déjà quelques années et grâce à qui j'ai pu poursuivre dans la recherche.*
- Berend Smit et Maddalena Venturoli pour leur gentillesse et les discussions enrichissantes que j'ai pu avoir avec eux lors des workshops du CECAM ainsi que Mounir Tarek et Juan Elezgaray pour leurs conseils avisés me permettant de mieux appréhender la biophysique des modèles gros grains.*
- Claire, Christelle, Marion, Marc, Cathel, Aurélie, Manu, Maissa, les Rémi's et tous ceux qui, dans le cadre d'activités associatives, que ce soit l'AFoDIB, 2D2B, AquiDoc ou le Réseau BIOTechno, m'ont fait découvrir d'autres domaines en communiquant avec passion sur leur sujet d'étude et avec qui j'ai vécu des expériences enrichissantes.*
- tous ceux qui ont contribué à faire de Bordeaux une ville d'accueil fort agréable.*

Je tiens à remercier ma famille, mes parents et mes grand-parents, plus particulièrement papi-net et mam'web pour m'avoir soutenu et épaulé durant toutes ces longues années d'étude, surtout dans les moments difficiles.

Table des matières

Introduction	7
1 Le contexte biologique	11
1.1 La mitochondrie	11
1.1.1 Origine : l'hypothèse de l'endosymbiose	12
1.1.2 Structure et dynamique des mitochondries	13
1.2 Le métabolisme mitochondrial	14
1.3 Les lipides	15
1.3.1 Les acides gras	15
1.3.2 Les phospholipides	15
1.4 Les membranes biologiques	18
1.5 La membrane interne mitochondriale	20
1.6 En résumé	20
2 Les paradigmes pour la modélisation cellulaire	23
2.1 Échelles et paradigmes	23
2.1.1 La modélisation multi-échelles et multi-paradigmes	24
2.1.2 Le couplage de modèles	26
2.2 Différents paradigmes disponibles : du global au local	26
2.2.1 Modélisation différentielle	27
2.2.2 Modélisation logique des réseaux génétiques	29
2.2.3 Les réseaux de Petri	29
2.2.4 Les automates cellulaires	30
2.3 Les Systèmes Multi-Agents	33
2.3.1 L'émergence	33
2.3.2 Les modèles d'agents	34
2.3.3 La gestion de l'environnement : l'espace	35
2.3.4 La gestion de l'environnement : le temps	38
2.3.5 Intérêt de la modélisation SMA pour les processus biologiques	39
2.3.6 Les plateformes de SMA	39
2.4 En résumé	45
3 Une membrane à l'échelle mésoscopique	47
3.1 La modélisation par modèles continus élastiques	47

3.1.1	Les modèles de type Ginzburg-Landau	48
3.1.2	Les modèles basés sur l'hamiltonien de Canham-Helfrich	48
3.2	Modèles continus vs modèles à particules	50
3.3	La modélisation par mécanique moléculaire	51
3.3.1	Modélisation par champs de force	52
3.3.2	De l'échelle microscopique à l'échelle mésoscopique	54
3.4	Les modèles à grains d'atomes : l'échelle mésoscopique	54
3.4.1	Les grains : les différents types, le nombre par lipide	55
3.4.2	La modélisation flexible des lipides	61
3.4.3	La modélisation rigide des lipides	65
3.4.4	La modélisation du solvant : explicite / implicite	65
3.4.5	Les potentiels intermoléculaires type Lennard-Jones	69
3.5	Les méthodes de calibrage	72
3.5.1	Validation par données macroscopiques	72
3.5.2	Validation par données microscopiques	73
3.5.3	Validation par données croisées	73
3.6	Vers un premier modèle à grains	74
4	MitoMAS	75
4.1	Définition d'un lipide modèle	75
4.1.1	Intégration d'une molécule de DMPC	75
4.1.2	Choix de la granularité	76
4.1.3	Extension du modèle à 3 grains	77
4.2	Les interactions entre grains d'atomes	77
4.2.1	Les interactions intramoléculaires	77
4.2.2	Les interactions intermoléculaires	79
4.3	Un solvant implicite grâce à un potentiel par paire	81
4.4	Modélisation de l'environnement	82
4.4.1	L'espace	82
4.4.2	Le temps	83
4.5	La dynamique des molécules	83
4.5.1	L'équation du mouvement	83
4.5.2	Modèle de déplacement des molécules	84
4.5.3	Gestion du pas de temps	86
4.6	En résumé : le modèle bio-physique	86
4.7	Le modèle agent de MitoMAS	87
4.7.1	La modélisation orientée objet	87
4.7.2	Les agents biologiques.	88
4.7.3	Différents modèles d'interactions	90
4.7.4	Les agents environnement	92
4.7.5	Des comportements spécifiques	93
4.7.6	Les interactions intermoléculaires	95
4.7.7	L'architecture du moteur de simulation	95

4.7.8	Le cycle de vie des agents : l'algorithme de la simulation	96
4.8	Conception du simulateur	97
4.8.1	La visualisation	98
4.8.2	Paramétrage d'une simulation	98
4.8.3	Le module de communication	99
4.9	En résumé : le modèle informatique	99
5	Les expérimentations <i>in-silico</i>	101
5.1	Plan d'expérience <i>in-silico</i>	101
5.2	Une membrane générique	102
5.2.1	Implémentation d'un phospholipide	102
5.2.2	Différentes configurations initiales du système	102
5.2.3	Initialisation aléatoire des lipides	103
5.2.4	Initialisation des lipides en monocouche	105
5.2.5	Initialisation des lipides en bicouche	108
5.2.6	Analyse des résultats	112
5.3	Vers une membrane interne mitochondriale	113
5.4	Extension : les systèmes mixtes phospholipides/protéines	115
5.4.1	Des protéines intramembranaires génériques	116
5.4.2	La chaîne respiratoire	118
5.4.3	Les résultats de simulation	119
5.5	En résumé	120
	Conclusion	121
	Bibliographie	129
	Annexes	139
A	Quelques éléments de mathématiques	139
A.1	Quaternions et rotations 3D	139
A.1.1	L'ensemble \mathbb{C} des nombres complexes	139
A.1.2	L'ensemble \mathbb{H} des quaternions	140
A.1.3	Pour aller plus loin...	144
A.2	Les modélisations différentielles	146
A.3	Les méthodes d'intégration	147
A.3.1	Difficultés	147
A.3.2	Deux familles de méthodes d'intégration	147
A.3.3	Intégration explicite	148
A.3.4	Intégration implicite	149
A.3.5	Analyses :	150
B	Quelques éléments de modélisation par SMA	153
B.1	Les plateformes génériques	153

B.2	Les plateformes à programmation simplifiée	153
B.3	Les plateformes à programmation avancée	154
B.4	Les méthodes de discrétisation de l'espace	154
Index		159

Introduction

La modélisation des objets biologiques et de leurs comportements au niveau moléculaire permet de mieux comprendre différents phénomènes observables au niveau macroscopique. Les techniques de la mécanique moléculaire représentent les objets biologiques au niveau atomique. Ces techniques constituent des outils performants pour étudier les structures et les dynamiques biologiques pour des temps de l'ordre de la nanoseconde et des tailles de l'ordre du nanomètre. A cause des temps de calcul, les simulations sont néanmoins restreintes à quelques centaines de molécules amphiphiles ce qui, pour une bicouche lipidique, correspond à une surface de l'ordre de la dizaine de nanomètres carrés.

Depuis quelques années, une technique alternative de résolution plus grossière, les modèles à grains d'atomes ou modèles "*gros grains*", offre la possibilité d'étudier les phénomènes biologiques à des échelles de temps de l'ordre du dixième de microseconde et d'espace de l'ordre du dixième de micromètre. Ces valeurs sont beaucoup plus proches de celles nécessaires pour l'étude de phénomènes comme la formation des membranes. Déjà éprouvés depuis de nombreuses années dans le domaine de la physique des polymères, ces modèles *gros grains* ont été employés plus récemment pour étudier les membranes lipidiques.

Dans le cadre de la modélisation d'une membrane lipidique, différents paradigmes de modélisation peuvent être envisagés. En adoptant un point de vue le plus global possible, c'est-à-dire en manipulant des variables moyennées sur l'ensemble du système moléculaire, il est possible de décrire une dynamique de ce système en utilisant un paradigme à base d'équations différentielles (EDO, EDP). Néanmoins, la prise en compte d'une part de la spatialisation des molécules et d'autre part, de la singularité de leurs conformations, souligne les limites de ce type de paradigme. En choisissant une approche de plus en plus locale des systèmes moléculaires, on peut faire appel à d'autres paradigmes tels que la modélisation logique, les réseaux de Petri, les automates cellulaires ou encore les Systèmes Multi-Agents (SMA).

Les SMA proposent de représenter chaque objet biologique par un agent capable d'exprimer toutes les spécificités de la molécule, en terme d'orientation, de conformation voire de dynamique interne. Par nature les SMA sont des systèmes dynamiques et distribués d'entités en interaction avec leur voisinage local, exactement à l'instar des lipides constituant les membranes, en précisant toutefois que dans ce cas, on parle généralement d'agents réactifs alors que les SMA plus traditionnels utilisés par exemple pour simuler le comportement d'insectes sociaux sont plutôt des agents cognitifs. Pour les simulations moléculaires, les agents sont situés dans un espace 3D et se déplacent dans cet environnement grâce à une équation du mouvement qui obéit aux lois de la physique classique.

Ce travail présente un modèle SMA pour la modélisation de la membrane interne mitochondriale. Les mitochondries sont des organites intracellulaires ayant la particularité de posséder deux membranes. De nombreux complexes enzymatiques participant au métabolisme de la mitochondrie sont situés dans cette membrane. C'est le cas par exemple de la chaîne respiratoire, un ensemble de complexes enzymatiques imbriqués dans la membrane interne mitochondriale, qui assurent la phosphorylation de l'adénosine diphosphate (ADP) en adénosine triphosphate (ATP), molécule énergétique utilisable par le reste de la cellule. Pour améliorer la compréhension du fonctionnement de cette chaîne respiratoire, nous proposons une modélisation de la membrane interne mitochondriale

seule qui servira de base à l'inclusion des complexes enzymatiques membranaires. Pour ce modèle nous nous attacherons à modéliser les forces qui s'exercent entre les lipides et les mouvements de ces molécules, qui produisent les ondulations et les invaginations caractéristiques de cette membrane interne.

Parmi les objectifs de cette thèse, nous souhaitons définir un modèle *gros grains* des lipides pour ensuite proposer une conception agent qui prenne en compte les lipides et leurs interactions. Enfin, nous souhaitons implémenter ce SMA sous la forme d'une plateforme capable de simuler des bicouches lipidiques suffisamment grandes sur des temps suffisamment longs pour en observer la dynamique. Nous souhaitons réaliser ce processus de modélisation en conservant la possibilité d'étendre le modèle pour intégrer des protéines telles que les complexes enzymatiques. Cette plateforme devra nous permettre de simuler des systèmes constitués d'un ou plusieurs types de lipides. L'un des phénomènes que nous souhaitons obtenir est l'apparition de structures telles que les micelles ou les replis des bicouches comme résultat de la dynamique des interactions entre lipides. Parmi les nombreux paramètres du modèle, nous souhaitons pouvoir en identifier certains susceptibles d'augmenter l'efficacité des simulations sans pour autant en altérer les résultats qualitatifs.

Ce mémoire se décompose en cinq chapitres. Le premier chapitre exposera le contexte biologique, à savoir la membrane interne mitochondriale et les phospholipides. Le second chapitre sera consacré à la présentation de plusieurs paradigmes capables de modéliser des systèmes biologiques. Nous finirons ce chapitre par une présentation du paradigme le plus adapté à une modélisation locale des objets biologiques et de leurs interactions : les SMA. Le troisième chapitre présentera le cas spécifique de la modélisation d'une membrane biologique. Il y sera fait notamment une revue d'un certain nombre de modèles *gros grains* disponibles dans la littérature. L'ensemble des propriétés de ces modèles nous permettra de spécifier un premier modèle à grains pour lipide dans le quatrième chapitre. Les différents choix de modélisation concernant ce modèle à grains donneront lieu à une conception agent sous la forme d'un SMA capable de simuler une membrane, voire d'inclure des peptides membranaires. Le cinquième et dernier chapitre rassemble les résultats de simulation des différents systèmes modélisés.

Résumé Cette thèse porte sur la modélisation de la membrane interne mitochondriale et des complexes enzymatiques de la chaîne respiratoire imbriqués dans cette bicouche phospholipidique. Une alternative aux techniques de la mécanique moléculaire qui représentent les objets biologiques au niveau atomique sont les modèles à grains d’atomes, ou modèles “*gros grains*”, qui offrent la possibilité d’étudier les phénomènes biologiques à des échelles de temps de l’ordre du dixième de microseconde et d’espace de l’ordre du dixième de micromètre, ce qui correspond à des valeurs beaucoup plus proches de celles nécessaires pour l’étude de phénomènes comme la formation de replis de la membrane. Le modèle proposé représente les phospholipides, constituants de la membrane, sous la forme de trimères rigides rectilignes avec un solvant implicitement modélisé. Ce modèle *gros grains* donne lieu à une conception orientée agent qui est implémentée sous la forme d’un Système Multi-Agents (SMA) appelé MitoMAS.

La plateforme MitoMAS nous a permis de simuler des systèmes avec différentes distributions initiales des phospholipides. Les systèmes initialisés de manière aléatoire montrent une agrégation des lipides pour minimiser l’exposition des queues au solvant. Ce phénomène naturel tend à valider la modélisation implicite du solvant. Les simulations réalisées avec des monocouches lipidiques ont permis de souligner l’influence de la distance de *cutoff* concernant les potentiels intermoléculaires. À partir de ces résultats, nous proposons de réduire cette distance de *cutoff* à 1nm ce qui augmente sensiblement l’efficacité des simulations. Avec une initialisation des lipides en bicouche, nous pouvons observer, sous certaines conditions de pression latérale, l’apparition de replis similaires à ceux de la membrane interne. Un ensemble de simulations avec deux types de phospholipides différenciés soit par la taille des queues, soit par le nombre de ces queues, produit des membranes qualitativement différentes mais néanmoins présentant des distances de plus proche voisin relativement identiques. MitoMAS nous a aussi permis de simuler des systèmes mixtes phospholipides/protéines membranaires. En présence de petites protéines génériques imbriquées, les replis de la membrane ne sont plus les mêmes et on observe dans certains cas des confinements de protéines sans que celles-ci soient proches les unes des autres dans la membrane. Les perspectives en terme d’expérimentations *in-silico* sont nombreuses. On pourra par exemple tester d’autres interactions intermoléculaires (formes et intensités différentes pour les potentiels). Les premiers résultats de modélisation d’un des complexes enzymatiques de la chaîne respiratoire montrent qu’il est possible de représenter une protéine uniquement par le maillage de sa surface et ce, en réutilisant les types que nous avons défini pour la membrane.

Mots clefs modélisation, simulation, membrane, mitochondrie, modèle gros grains, systèmes multi-agents (SMA)

Chapitre 1

Le contexte biologique

Ce chapitre expose le contexte biologique de cette thèse, en l’occurrence la mitochondrie et plus spécifiquement la chaîne respiratoire constituée d’un ensemble de complexes imbriqués dans la membrane interne mitochondriale.

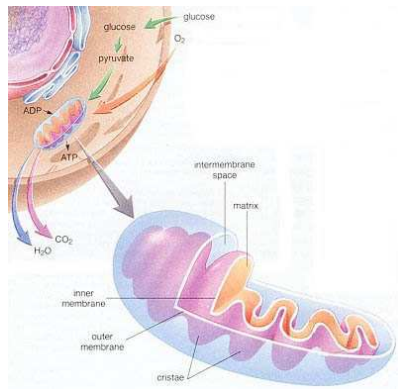
Après une présentation de la mitochondrie, l’attention est portée plus précisément sur les objets biologiques à modéliser, c’est-à-dire les complexes enzymatiques et les phospholipides, constituants de la membrane interne mitochondriale. Pour choisir à la fois l’approche de modélisation et les “modalités” de cette modélisation, il est important de bien spécifier les propriétés et les variabilités entre les différents objets biologiques à prendre en compte.

1.1 La mitochondrie

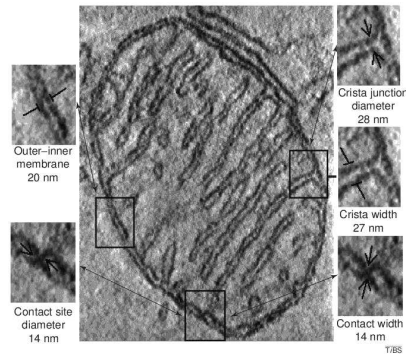
La mitochondrie est l’objet d’études scientifiques depuis près de 150 ans puisque déjà en 1857, Kölliker décrivait les aspects de la mitochondrie dans le muscle. En 1890, Altmaan a décrit une technique de coloration des mitochondries et formule l’hypothèse de leur autonomie métabolique et génétique. En 1937, Hans Adolf Krebs a construit un modèle qu’il appela “*citric acid cycle*” [Kre40]. Ce cycle a lieu dans la mitochondrie chez les eucaryotes. En 1940-43, Albert Claude a isolé les mitochondries dans des cellules du foie [Cla44]. En 1948-50, Kennedy et Lehninger ont montré que des voies métaboliques telles que le cycle de Krebs, la β -oxydation et la phosphorylation oxydative ont lieu tous dans la mitochondrie. En 1978, Peter Mitchell a obtenu le Prix Nobel pour sa théorie chimiosmotique détaillant un mécanisme original pour le fonctionnement des oxydations phosphorylantes [MIT79]. En 1981, Sanger et son équipe ont publié la séquence de l’ADN mitochondrial humain [ABB⁺81]. Finalement, Boyer et Walker, eux aussi, ont obtenu le Prix Nobel en 1997 pour leurs études sur la structure et le fonctionnement de l’ATP synthétase [Boy97].

La mitochondrie est un **organite intracellulaire** (en vert sur la Fig.1.1.3(a)), lieu de la phosphorylation oxydative qui régénère l’ATP¹, la molécule énergétique utilisable par la cellule. D’une dimension de 1 à 2 μm de long et de 0,5 à 1 μm de large, la mitochondrie possède **deux bicouches phospholipidiques**, une externe et une interne, qui délimitent trois espaces : l’espace extra-mitochondrial (cytoplasme de la cellule), l’espace intermembranaire et la matrice mitochondriale (Fig.1.1.1(a)).

¹L’adénosine triphosphate (ATP) est une molécule utilisée chez tous les organismes vivants pour fournir de l’énergie aux réactions chimiques.



(a) La mitochondrie est un organe intra-cellulaire.

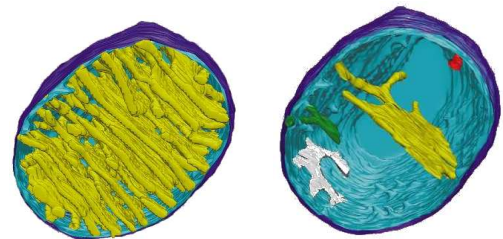


(b) Dimensions de certaines particularités des mitochondries.

FIG. 1.1.1 – La mitochondrie.

La tomographie de coupes épaisses de mitochondries permet de créer une vue 3D de cet organe (Fig.1.1.2). Le modèle entier (Fig.1.1.2(a)) montre toutes les crêtes en jaune, la membrane interne en bleu clair et la membrane externe en bleu foncé. Une sélection de quatre crêtes représentatives marquées de différentes couleurs (Fig.1.1.2(b)) permet de mieux apprécier **leur complexité structurelle** et leurs points d'attache sur la membrane interne. Ces crêtes peuvent être perçues comme des replis de la membrane interne augmentant sa surface en occupant le volume de la matrice mitochondriale.

La figure 1.1.1(b) montre certaines de ces spécificités structurelles, identifiées et agrandies de part et d'autres de l'image, avec les dimensions moyennées sur différents tomogrammes de mitochondries reconstruites *in situ* dans des tissus neuronaux, des tissus adipeux bruns et de *Neurospora crassa*.



(a) Modèle entier.

(b) Sélection de quatre crêtes.

FIG. 1.1.2 – Images de synthèse générées à partir de segments 3D de tomogrammes de mitochondries du cervelet de poulet (d'après Frey, T. G. et al. [FM00]).

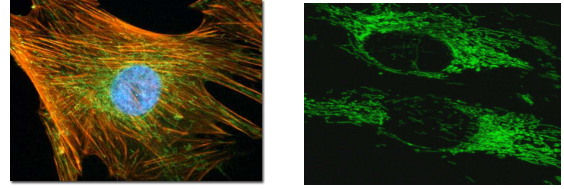
1.1.1 Origine : l'hypothèse de l'endosymbiose

Le fait que la mitochondrie possède son ADN propre, comme les chloroplastes, fait pencher pour une origine exogène de la mitochondrie qui serait la lointaine descendante d'une bactérie "*adoptée*" par les premiers eucaryotes. Cette hypothèse, **l'endosymbiose**, a été développée et argumentée par Lynn Margulis dès 1966, puis a été appuyée par le séquençage de l'ADN spécifique des mitochondries en 1980. Il semble qu'au cours de l'évolution l'ADN de la "*bactérie mitochondrie*" ait subi diverses évolutions, perdu certains gènes, souvent au profit de l'ADN de la cellule hôte. Parallèlement à ce report de la synthèse de certaines protéines vers l'hôte, ce dernier a développé un arsenal de translocases, enzymes permettant le transfert des protéines depuis le cytosol vers la matrice mitochondriale. La plupart des complexes enzymatiques (ex : l'ATP-synthase) sont formés par la juxtaposition de polypeptides synthétisés à partir de l'ADN mitochondrial et de l'ADN du noyau.

1.1.2 Structure et dynamique des mitochondries

Les mitochondries sont capables de fusionner entre elles pour former un ensemble de tubules interconnectés appelé réseau mitochondrial (Fig.1.1.3). Des phénomènes de fission et de fusion (Fig.1.1.4) sont à l'origine de la reconfiguration permanente de ce réseau mitochondrial. La taille du réseau mitochondrial par cellule est régulée par l'activité cellulaire. La topologie de ce réseau mitochondrial varie d'un tissu à un autre au sein d'un même organisme mais aussi d'une espèce à une autre (Fig.1.1.5).

Cette complexité structurelle et dynamique rend particulièrement délicate la modélisation de la membrane interne dans laquelle sont imbriqués les complexes enzymatiques.



(a) En bleu, le noyau, en orange le cytosquelette et en vert les mitochondries (source : Rossignol et al., 2004). www.microscopyu.com).

FIG. 1.1.3 – Le réseau mitochondrial.

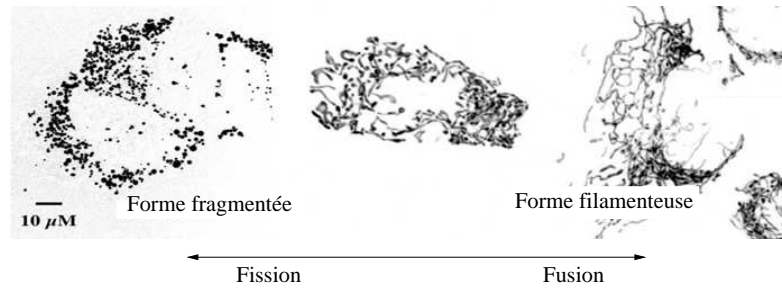
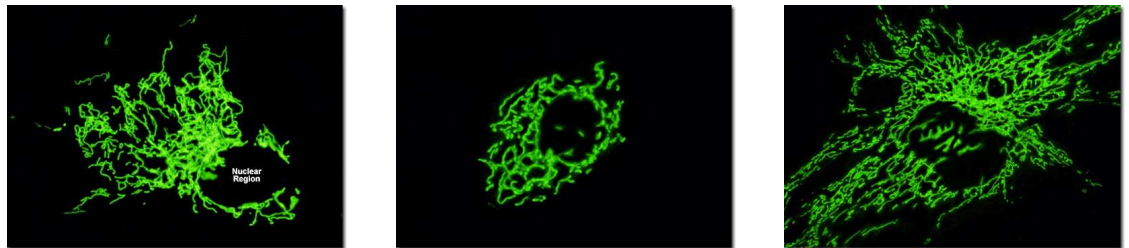


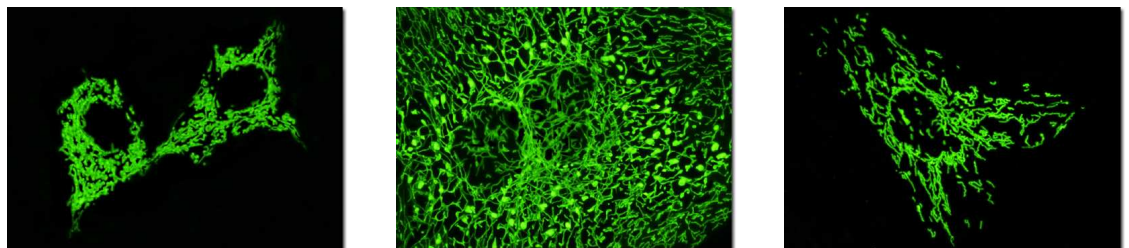
FIG. 1.1.4 – Fusion et fission du réseau mitochondrial (images de Werner Koopman, 2005).



(a) Fibroblastes de poumon humain (lignée MRC-5).

(b) Carcinomes épithéliaux pulmonaires humains (lignée A-549).

(c) Fibroblastes rénaux de singe vert (lignée CV-1).



(d) Cellules épithéliales rénales de porc (lignées LLC - PK1).

(e) Cellules épithéliales rénales de chien (lignée MDCK).

(f) Fibroblastes de peau de cerf *Muntiacus*.

FIG. 1.1.5 – Diversité topologique du réseau mitochondrial.

1.2 Le métabolisme mitochondrial

Dans la cellule, les réactions de l'anabolisme (synthèses) nécessitent de l'énergie. Au contraire, les réactions du catabolisme (dégradation) libèrent (globalement) de l'énergie. Cette énergie doit être stockée pour être échangée entre les différentes voies métaboliques. Elle est stockée dans des composés riches en énergie (qui libèrent au moins 21 kJ/mol d'énergie lors de leur hydrolyse) en particulier l'ATP.

Lors des réactions du catabolisme que sont la glycolyse (dans le cytosol) et le cycle de Krebs (dans la mitochondrie) il y a production de coenzymes réduits (NADH et FADH₂). Ces coenzymes vont être ré-oxydés par la chaîne respiratoire (en aérobiose) au niveau des crêtes de la membrane interne mitochondriale.

La chaîne respiratoire

Il existe 5 complexes protéiques et coenzymes impliqués dans les oxydations phosphorylantes de la chaîne respiratoire (Fig.1.2.1). Les 4 premiers complexes (I, II, III et IV) interviennent dans le transport des électrons et le cinquième complexe (V) intervient dans la synthèse d'ATP.

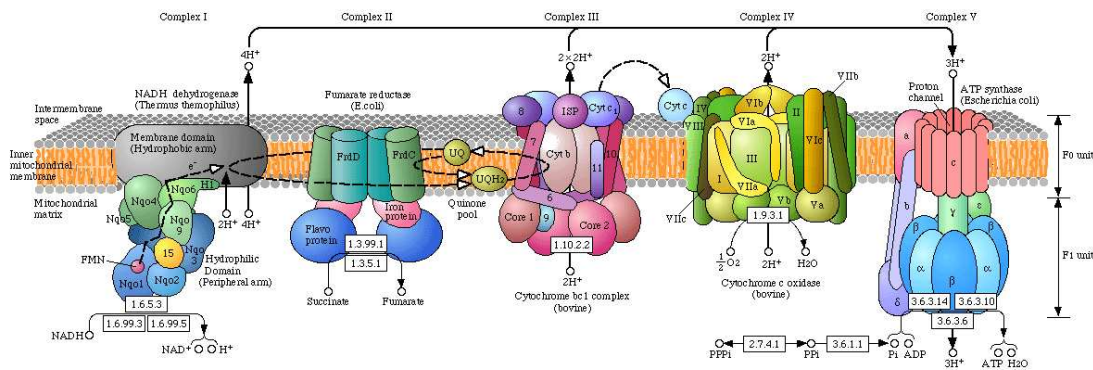
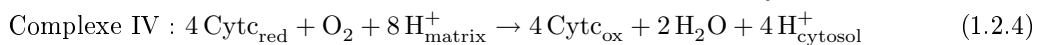
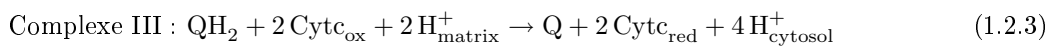
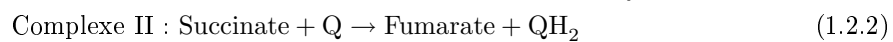


FIG. 1.2.1 – "Oxidative-phosphorylation reference pathway" de KEGG

Sans entrer dans le détail du fonctionnement de la chaîne respiratoire, nous pouvons résumer l'activité des 5 complexes et leur "coopération" au moyen des 5 équations de réactions ci-dessous (Eq.(1.2.1) à Eq.(1.2.5)).



On notera au vue du schéma 1.2.1 que chaque complexe est composé de plusieurs dizaines de sous-unités et qu'ils sont de taille relativement importante surtout si on les compare avec les constituants de la membrane, les phospholipides, dans lesquels ils sont enchâssés.

La section suivante a pour objectif de présenter les composants des membranes biologiques que sont les phospholipides. Nous commencerons par l'étude des lipides car les phospholipides sont à la base des lipides.

1.3 Les lipides

Les lipides jouent un rôle important dans la cellule, notamment comme **composants majeurs des membranes**. Ils constituent également une importante quantité d'énergie stockée et ils sont directement impliqués dans la signalisation cellulaire, à la fois comme hormones stéroïdes (oestrogènes et testostérone) et comme messagers potentiels qui transportent les signaux des récepteurs membranaires aux cibles situées à l'intérieur de la cellule.

1.3.1 Les acides gras

Les acides gras sont des acides carboxyliques à chaîne aliphatique hydrophobe saturée ou insaturée. Appartenant à la catégorie des lipides, ils consistent en une longue chaîne hydrocarbonée, contenant le plus souvent de 16 à 18 atomes de carbone, avec un groupe carboxyle (COO-) à une extrémité.

Les acides gras insaturés contiennent une ou plusieurs double liaisons entre des atomes de carbone. Ces doubles liaisons induisent un coude dans la structure des chaînes hydrocarbonées et sont à l'origine du rayon de Stokes assimilable à la largeur du cylindre ou du cône contenant le lipide. Les positions de ces doubles liaisons suggèrent la forme géométrique (cylindre ou cône) la plus appropriée pour modéliser le lipide.

Dans les acides gras saturés, tous les atomes de carbone sont liés au plus grand nombre possible d'atomes d'hydrogène. Les longues chaînes hydrocarbonées des acides gras ne contiennent que des liaisons non polaires C-H, qui sont incapables d'interagir avec l'eau. La **nature hydrophobe** de ces chaînes d'acides gras est responsable de la plupart des comportements des lipides complexes, particulièrement pour ce qui est de la formation des membranes biologiques.

Nom commun	Formule chimique	Désignation courte	Masse moléculaire (Kda)
Acides gras saturés			
laurique	$CH_3(CH_2)_{10}COOH$	12 :0	200.3
myristique	$CH_3(CH_2)_{12}COOH$	14 :0	228.4
palmitique	$CH_3(CH_2)_{14}COOH$	16 :0	256.4
stéarique	$CH_3(CH_2)_{16}COOH$	18 :0	284.4
arachidique	$CH_3(CH_2)_{18}COOH$	20 :0	312.5
lignocérique	$CH_3(CH_2)_{22}COOH$	24 :0	368.6
Acides gras insaturés			
palmitoléique	$CH_3(CH_2)_5CH = CH(CH_2)_7COOH$	16 :1(n-7)	254.4
oléique	$CH_3(CH_2)_7CH = CH(CH_2)_7COOH$	18 :1(n-9)	282.4
linoléique	$CH_3(CH_2)_4CH = CHCH_2CH = CH(CH_2)_7COOH$	18 :2(n-6)	280.4
(α)-linoléique	$CH_3CH_2CH = CHCH_2CH = CHCH_2CH = CH(CH_2)_7COOH$	18 :3(n-3)	278.4
arachidonique	$CH_3(CH_2)_4(CH = CHCH_2)_3CH = CH(CH_2)_3COOH$	20 :4(n-6)	304.5

TAB. 1.3.1 – Les principaux acides gras cellulaires (d'après [For79]).

Le tableau 1.3.1 regroupe les différents acides gras cellulaires et donne notamment le nombre d'atomes de carbone, avec, pour les acides gras insaturés, la position de la double liaison. En d'autres termes, ces données permettent de déduire la **longueur des queues** des phospholipides et l'**encombrement stérique** des phospholipides pourvus d'acide gras insaturés. Ces informations nous seront nécessaires par la suite car dans le cadre d'une modélisation suffisamment locale pour prendre en compte les molécules, nous aurons besoin de spécifier la taille et le volume de chaque type de lipides.

1.3.2 Les phospholipides

Les phospholipides sont les principaux constituants des membranes cellulaires. Ces lipides possédant un groupe phosphate sont des molécules amphiphiles. Ils ont une affinité pour les composés

polaires grâce à leurs têtes hydrophiles (groupe phosphate et groupe X), et pour les composés apolaires grâce à leurs queues hydrophobes (acides gras notés R1 et R2). Ainsi, dans l'eau, les phospholipides peuvent s'organiser en micelles², en membranes ou en d'autres structures de manière à minimiser les contacts entre l'eau et leurs queues hydrophobes. La figure 1.3.1 présente quatre des principaux phospholipides que l'on retrouvera par la suite. On peut remarquer que ces types de phospholipides se différencient entre eux par leur tête hydrophile (groupe X en violet).

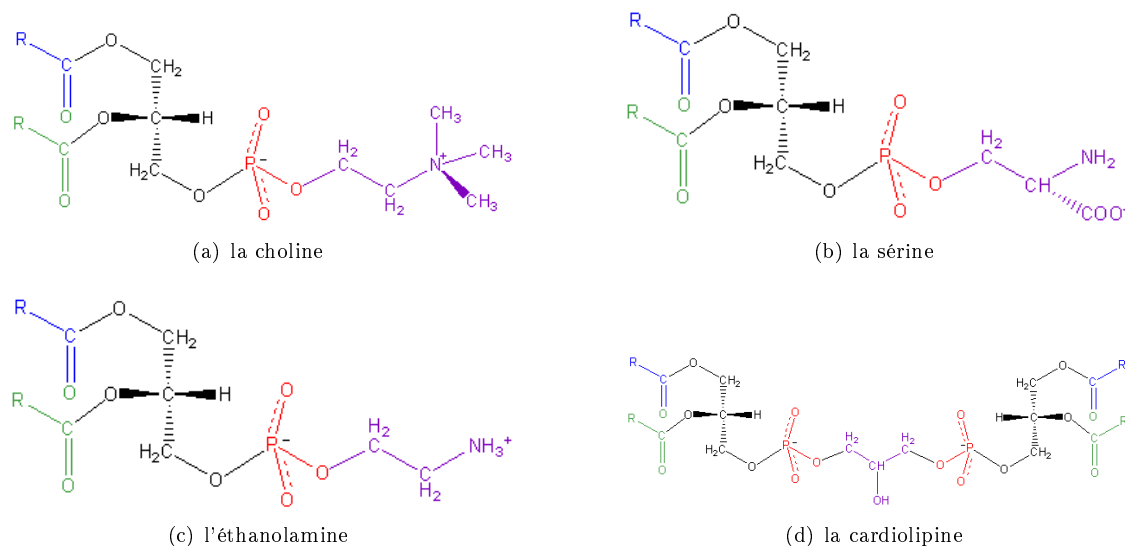


FIG. 1.3.1 – Phospholipides génériques. Bleu/vert : les acides gras, noir : le glycérol, rouge : le phosphate et violet : le composé hydroxylé estérifiant le phosphate (images : wikipedia).

► Propriétés des phospholipides

Le tableau 1.3.2 donne une répartition de différents acides gras pour certains phospholipides issus des mitochondries de foie de rat [WH69]. Ces données montrent que si la tête hydrophile détermine effectivement le type du phospholipide, plusieurs variétés peuvent coexister et se distinguer uniquement par la longueur de leur queue, c'est-à-dire les acides gras qui les constituent. Par exemple, la phosphatidyl-choline, phospholipide caractérisé par la choline (Fig.1.3.1(a)), est préférentiellement associée à l'acide gras stéarique (18 :0) en R1 et à l'acide arachidonique (20 :4) en R2.

Le tableau 1.3.3 synthétise les données précédentes (Tab.1.3.1 et Tab.1.3.2) en terme d'acide gras associés majoritairement à chaque type de phospholipides, de longueur des queues, d'encombrement stérique et de charge des têtes. Les variations d'un type à l'autre ne nous paraissent pas considérables, excepté pour la cardiolipine (diphosphatidyl-glycérol) qui est constituée de deux PG, soit de quatre queues apolaires.

Parmi les propriétés des phospholipides, les plus importantes sont la longueur des queues et le nombre de queues par phospholipide car elles sont directement liées à la conformation de la molécule. Dans ce qui suit, nous présentons en détail chaque type de phospholipides en faisant un distinguo entre les molécules à deux queues et celles à quatre queues.

²Une micelle est un agrégat sphéroïdal de molécules possédant une tête polaire hydrophile dirigée vers le solvant et une chaîne hydrophobe dirigée vers l'intérieur.

Position	Acides gras						
	14 :0	16 :0	18 :0	18 :1	18 :2	20 :4	22 :6
la phosphatidyl-choline (PC)							
R1		23	65	7	1	trace	
R2	1	6	4	13	23	39	7
la phosphatidyl-éthanolamine (PE)							
R1		25	65	8			
R2	2	11	8	8	10	46	13
la phosphatidyl-sérine(PS)							
R1		5	93	1			
R2		6	29	8	4	32	19
le phosphatidyl-glycérol (PG)							
R1		7	3	3	81		
R2		3	1	34	50	2	1

TAB. 1.3.2 – Distribution des acides gras de certains phospholipides de foie de rat (d'après Wood et Harlow [WH69]).

		PC 18 :0 20 :4	PE 18 :0 20 :4	PS 18 :0 20 :4	PG 18 :2 18 :2	CL (18 :2 18 :2)*2
tête	taille (A^3)	33	25	40	80	26
	masse ($g.mol^{-1}$)	87	33	88	75	58
	charge nette (à pH7)	0	0	-1	-1	-2
queue	taille (A)(nb double carbone * longueur double carbone)	27 30	27 30	27 30	27 27	27 27
	degrés insaturation (nb double liaison)	1	1	1	2	4
	masse ($g.mol^{-1}$)(cf nb carbone)	252 280	252 280	252 280	252 252	(252 252) x2
	hydrophobie : position et degrés de l'angle (rayon de Stokes)	4 30° (12)	4 30° (12)	4 30° (12)	2 30° (12)	2 30° (12)

TAB. 1.3.3 – PC (phosphatidyl-choline), PE (phosphatidyl-éthanolamine), PS (phosphatidyl-sérine), PG (phosphatidyl-glycérol), CL (cardiolipine)

► Phospholipides à deux queues

► Au sens le plus strict, la lécithine désigne uniquement la **phosphatidyl-choline** (PC), c'est-à-dire un lipide formé à partir d'une choline, d'un phosphate, d'un glycérol et de deux acides gras (Fig.1.3.1(a)). Dans ce contexte, il serait plus juste de parler des lécithines car il ne s'agit pas d'une seule molécule mais d'un groupe dont la composition en acide gras varie d'une molécule à l'autre (Tab.1.3.2). En général, une lécithine comporte un acide gras saturé et un acide gras insaturé. La phosphatidylcholine est un lipide abondant dans les extractions cellulaires et constitue près de la moitié des phospholipides des membranes animales. Les composés poly-insaturés sont regroupés en position R2 et les acides gras saturés les plus abondants en position R1 (Tab.1.3.2).

► Le seul composé de **phosphatidyl-sérine** (PS) est la forme diacyl : 1,2-diacyl-sn-glycero-3-phospho-L-serine ou phosphatidylserine, que l'on trouve uniquement dans les cellules animales. La phosphatidylsérine présente en général des chaînes très insaturées. La formule (Fig.1.3.1(b)) donne clairement trois groupes ionisables : un acide diester phosphorique, un groupe amino et une fonction carboxyle.

► Dans les tissus de mammifères et de plantes, la **phosphatidyl-éthanolamine** (PE) (Fig.1.3.1(c)) apparaît généralement en moindre proportion que la phosphatidylcholine. Dans les bactéries, c'est néanmoins le principal phospholipide présent. La distribution des acides gras (Tab.1.3.2) montre un haut degré d'insaturation comme avec la phosphatidylcholine.

► Phospholipides à quatre queues

► Le **phosphatidyl-glycérol** (PG), le plus simple des polyol-phospholipides, a été isolé et analysé en 1958 [Ben58]. Il peut être défini comme 1,2-diacyl-sn-glycero-3-phospho-1'-sn-glycérol.

Le phosphatidyl-glycérol est formé de deux acides gras, liés par des liaisons ester à un premier glycérol dont la troisième fonction est estérifiée par un acide phosphorique comme dans les autres phospholipides. L'autre fonction acide estérifie une deuxième molécule de glycérol. Ce lipide est commun mais de très petites quantités sont relevées dans les tissus animaux, essentiellement dans les mitochondries. Dans les plantes, ils constituent de 20 à 30% des phospholipides, essentiellement dans les chloroplastes. Dans les bactéries les proportions peuvent s'élever à plus de 70% des lipides.

➤ Également référencé historiquement comme la cardiolipine (CL), le **diphosphatidyl-glycérol** est présent presque exclusivement dans les mitochondries et les bactéries. Il peut représenter près de 20% des lipides mitochondriaux. La cardiolipine a été découverte dans un tissu de coeur de boeuf en 1942 mais a été par la suite identifiée comme non spécifique de ce type de tissus. La cardiolipine (Fig.1.3.1(d)) est un diphosphatidyl-glycérol, dans lequel deux acides phosphatidiques estérifient symétriquement les deux fonctions alcool primaires de la même molécule de glycérol. Curieusement, la cardiolipine de mammifères contient à plus de 90% un seul acide gras, l'acide linoléique. Par ailleurs, il a été montré que l'espèce de cardiolipine la plus abondante entre divers organismes et tissus (coeur humain, lymphoblastes humain, foie de rat, drosophiles, levures, cellules végétales) contient seulement un ou deux types d'acides gras, ce qui génère un haut degré d'uniformité structurelle [SRX⁺05]. Ce phospholipide a beaucoup été étudié depuis que son implication dans le fonctionnement la chaîne respiratoire mitochondriale a été montré dans les membranes thylakoïdes des chloroplastes et des bactéries [Rob93]. Les phosphatidyl-glycérol et les cardiolipines sont présents dans la membrane interne des mitochondries où ils sont liés spécifiquement aux enzymes intrinsèques, comme la cytochrome oxydase de la chaîne respiratoire mitochondriale.

En se basant sur ces observations et sur les données du tableau 1.3.3, nous avons choisi de porter notre attention sur deux types de phospholipides : les cardiolipines (CL) et les phosphatidylcholines (plus précisément les dimyristoyl-phosphatidylcholines ou DMPC) pour composer les membranes mixtes que nous modéliserons par la suite.

Nous venons de faire une présentation des composants majeurs des membranes biologiques. Les phospholipides se distinguent les uns des autres par la nature de leur tête, c'est-à-dire l'alcool qui estérifie une seconde fois l'acide phosphorique mais se différencie également par leurs acides gras qui déterminent la longueur et l'encombrement stérique des molécules. Par ailleurs d'autres molécules, comme le cholestérol, sont susceptibles de s'ajouter aux phospholipides pour modifier sensiblement certaines propriétés des membranes comme par exemple la fluidité, ou la manière dont les phospholipides de même type s'associent préférentiellement pour former des microdomaines (ou *rafts*).

Les parties suivantes porteront sur les membranes biologiques en général, leur diversité de forme et de composition, leurs propriétés communes et celles qui les différencient pour ensuite introduire la membrane interne mitochondriale, support de la chaîne respiratoire.

1.4 Les membranes biologiques

La vie dans toute sa diversité n'est devenue possible qu'après l'apparition de membranes **séparant et protégeant** les systèmes vivants primitifs de leur environnement tout en autorisant des échanges de matière régulés avec lui (Fig.1.4.1).

Dans les cellules évoluées, des membranes assurent aussi la **compartmentation** des différentes activités d'une cellule et le transport de molécules choisies entre les différents compartiments.

La figure 1.4.2(a) donne le schéma d'une cellule avec notamment ses membranes : 2. membrane nucléaire; 4. vésicule; 5. réticulum endoplasmique rugueux; 6. appareil de Golgi; 8. réticulum endoplasmique lisse; 9. mitochondrie; 10. vacuole; 11. cytoplasme; 12. lysosome.

En biologie cellulaire, la membrane (Fig.1.4.2(b)) désigne un **assemblage de molécules en un double feuillet** séparant la cellule de son environnement et délimitant les organites à l'intérieur de celle-ci [LS95].

Les lipides (phospho- et glyco-lipides) assurent la structure bidimensionnelle de base, la fluidité, la souplesse et l'étanchéité nécessaires aux fonctions des membranes et, servent de solvant pour les protéines membranaires.

Les protéines jouent des rôles très variés : pompes ou canaux pour le transport sélectif de molécules ou d'ions, transducteurs pour les signaux chimiques (électriques ou hormonaux), récepteurs et points d'attache divers ; elles participent aussi directement à la résistance mécanique des membranes et à la régulation de leurs formes.

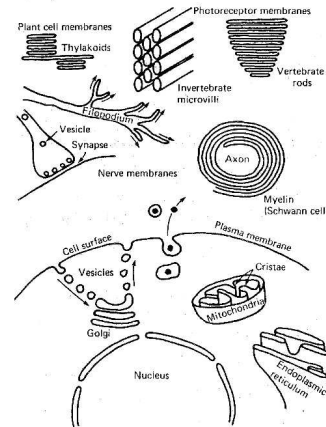
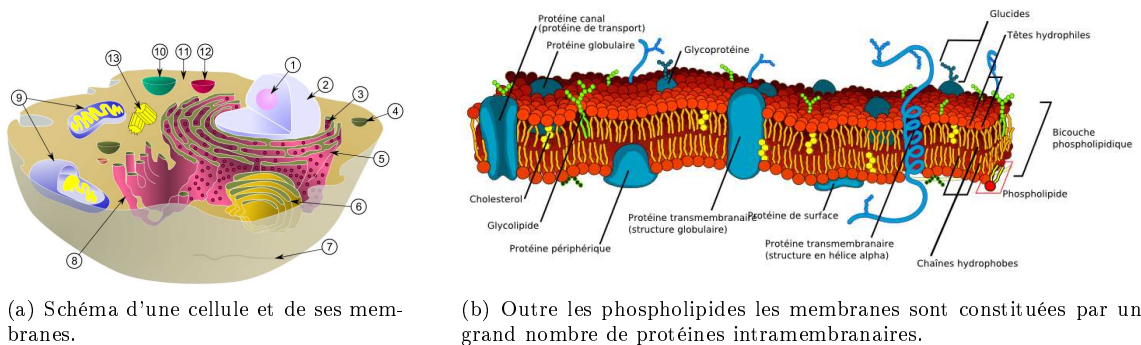


FIG. 1.4.1 – Diversité des membranes biologiques [IT73].



(a) Schéma d'une cellule et de ses membranes.

(b) Outre les phospholipides les membranes sont constituées par un grand nombre de protéines intramembranaires.

FIG. 1.4.2 – Membranes cellulaires (source : Wikipedia).

Les compositions en lipides sont **différentes d'une membrane à une autre** (Fig.1.4.3). Les constituants lipidiques majeurs des cellules animales sont deux phospholipides, phosphatidylcholine (PC) et phosphatidyl-éthanolamine, ceux des cellules végétales sont des glycolipides, digalactosyldiglycérine (DGDG) et monogalactosyldiglycérine (MGDG). Dans les cellules animales, les membranes plasmiques contiennent davantage de cholestérol (30% en poids) que les membranes des mitochondries ou du réticulum endoplasmique (3 à 6%). Les membranes myéliniques des cellules nerveuses sont plus riches en glycolipides que les autres, elles contiennent également moins de protéines, 25% en poids au lieu de 50% dans une membrane plasmique [ABL86].

Les constituants ne sont généralement pas distribués uniformément dans une membrane. L'intérieur et l'extérieur d'une membrane biologique sont dissymétriques. Cette **dissymétrie de composition entre les deux feuillets** est associée à une dissymétrie de fluidité et d'une dissymétrie dans le sens d'orientation des protéines.

La **distribution des constituants n'est pas homogène** non plus le long des membranes. D'un côté, des interactions préférentielles lipide-protéine changent localement la composition en lipides. De l'autre, les protéines forment aussi des amas immobiles et localisés, soit par auto-association, soit par ancrage chimique à des molécules extérieures à la membrane : anticorps, polysaccharides, filaments du cytosquelette, réseaux polymériques divers. La composition locale des membranes peut aussi s'adapter à leur courbure.

Du point de vue des lipides, les effets principaux sont relativement simples. Il est établi que le mélange de lipides plus ou moins saturés, plus ou moins rigides (comme le cholestérol), avec une tête polaire plus ou moins large comme les cholines, les lysolécithines (lécithines à une queue) et les

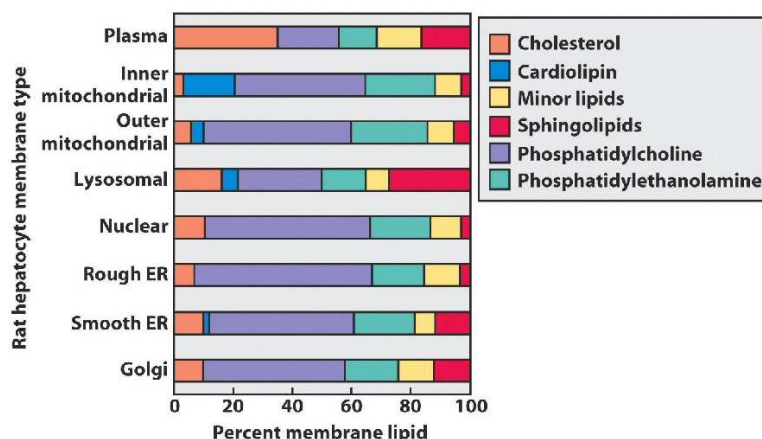


FIG. 1.4.3 – Composition de différentes membranes biologiques (sources : Département de Chimie et Biochimie de l’Université du Texas).

éthanolamines, permet de réguler la fluidité des membranes, de favoriser l’insertion des protéines et de contrôler les déformations de courbure. La plupart des effets s’interprètent directement en termes de contraintes locales, d’empilement des molécules, de courbure spontanée et de rigidité. D’autres sont plus subtils car ils dépendent du couplage entre différentes variables. On peut montrer que la rigidité d’une membrane est diminuée lorsque celle-ci a la possibilité d’ajuster sa composition en fonction des courbures qui lui sont imposées [Sac90]. Les effets liés aux protéines sont encore plus complexes.

1.5 La membrane interne mitochondriale

La mitochondrie est un organe intracellulaire qui présente la spécificité d’être constituée de deux membranes : une membrane externe et une membrane interne. Cette structure atypique peut être expliquée par l’origine exogène de l’organe (voir l’hypothèse de l’endosymbiose, section 1.1.1).

Ces deux membranes se révèlent très différentes.

La membrane externe est formée de 50% de protéines et de 50% de lipides polaires. Elle contient de nombreuses protéines appelées porines qui forment des canaux aqueux au travers de la membrane et laissent passer toutes les molécules hydrophiles d’une masse moléculaire inférieure à 10 000 daltons.

La membrane interne est beaucoup moins perméable que la membrane externe. Elle est composée de 80% de protéines et de 20% de lipides (proportion en masse). Elle contient, en proportion élevée relativement aux autres membranes, des phospholipides doubles, les **cardiolipines**, renfermant 4 acides gras. Les phospholipides majoritairement représentés restent la **phosphatidyl-choline** (PC) et la **phosphatidyl-éthanolamine** (PE).

1.6 En résumé

Les objectifs premiers de cette thèse sont donc :

- la modélisation des **phospholipides** et leur organisation naturelle susceptible de produire des pliements de cette membrane interne;
- la modélisation de la **membrane interne mitochondriale** avec ses propriétés macroscopiques telles que l’épaisseur, la fluidité et une certaine rigidité;

A terme, il sera éventuellement intéressant de modéliser des **protéines membranaires** dans la membrane mitochondriale interne.

En effet, nous avons vu en début de chapitre que la membrane interne contient les complexes enzymatiques de la chaîne respiratoire. Parmi les rôles attribués à ces protéines intramembranaires, nous pouvons citer : le confinement d'intermédiaires de réaction [SP00], la stabilisation de complexes protéiques [APBBFS⁺04], l'augmentation de la capacité d'insertion de protéine pour la membrane interne [APN⁺98] et la génération de replis, appelés également crêtes ou cristae, modifiant la morphologie de cette membrane [PVC⁺02].

De tous ces phénomènes résultant des interactions phospholipides/protéines, nous souhaiterions être en mesure d'évaluer, d'une part, l'impact de la présence de ces protéines membranaires au niveau des replis, et d'autre part, si effectivement les replis sont susceptibles de produire des situations de confinement des protéines membranaires.

Chapitre 2

Les paradigmes pour la modélisation cellulaire

Ce chapitre présente certaines approches employées dans le cadre de la modélisation des processus biologiques. Dans un premier temps, nous préciserons quelques définitions d'échelle, de modélisation multi-échelles, de paradigme ou encore de modélisation multi-paradigmes. Ensuite nous expliciterons plus particulièrement certaines approches de modélisation depuis les modélisations les plus globales jusqu'aux modélisations les plus locales, cette distinction portant sur le niveau d'abstraction des éléments modélisés. Nous terminerons par une revue des modèles et des plateformes multi-agents utilisés dans le contexte de la biologie.

2.1 Échelles et paradigmes

Selon la discipline, voire le contexte historique, la définition de modèle et de modélisation peut varier notablement. Sans entrer dans une discussion propre à l'épistémologie, nous retiendrons une définition contemporaine donnée par J. Ferber [Fer95] : *“Un modèle, en science, est une image stylisée et abstraite d'une portion de réalité”*.

La simulation de processus biologiques, et plus précisément la simulation cellulaire, devient un enjeu important de nos jours. En effet, le domaine de la recherche en biologie a dû acquérir, classer et stocker la masse des données issues entre autre du séquençage, avec une prise en compte des problèmes qui se posent pour aller du gène à la formation de la protéine, à son contrôle et à sa signification physiologique et physiopathologique. Il s'agit maintenant d'analyser ces données hétérogènes, d'identifier les événements significatifs et de les interpréter, en utilisant les techniques de modélisation et de simulation.

Après l'étude du génome (ADN), du transcriptome (ARN) et du protéome (protéines) vient l'étude de l'interactome qui inclut l'étude du métabolisme. Or l'étude du métabolisme cellulaire, et donc celle du métabolisme mitochondrial, impose des contraintes délicates à prendre en compte avec les paradigmes standards manipulant des variables moyennées comme les systèmes d'équations différentielles (souvent utilisés en modélisation biologique).

Au titre des difficultés spécifiques à cette étude de l'interactome, nous pouvons citer :

- le **partitionnement de l'espace** par les membranes biologiques et, plus particulièrement pour notre étude, par la membrane interne mitochondriale et ses nombreux replis ;
- les **interactions** entre les objets biologiques de différente nature comme par exemple l'interaction entre les membranes et les protéines ;
- la **complexité des objets** biologiques que sont les complexes enzymatiques en terme de structure, de dynamique interne et la nécessité de prendre en compte l'ordonnement des réactions enzymatiques.

On considère en général que le nombre important de molécules impliquées dans les processus

biologiques peut justifier une utilisation satisfaisante de modèles dit “*continus*” manipulant des variables moyennées sur l’ensemble de la population moléculaire (loi des grands nombres). Toutefois ces modèles ne permettent que difficilement de prendre en compte la localisation des molécules dans des structures pouvant discrétiser l’espace de manière complexe, comme c’est le cas avec la membrane interne mitochondriale.

Nous allons maintenant préciser les notions de multi-échelles et de multi-paradigmes.

2.1.1 La modélisation multi-échelles et multi-paradigmes

► La modélisation multi-échelles

Par définition¹, une échelle est une proportion de taille entre la représentation d’une chose et la chose représentée. Une modélisation multi-échelles intègre donc simultanément différentes proportions de taille d’un même système. Par exemple, une modélisation multi-échelles d’une problématique biologique intègre plusieurs échelles ou finesses de représentation des objets biologiques. Dans ce cas, on pourra s’intéresser aux interactions entre des bancs de poissons en se plaçant soit au niveau de la population de poissons dans son ensemble, soit au niveau des groupes ou encore au niveau d’un poisson. De la même manière, la modélisation des organes d’un être vivant peut se faire au niveau des tissus cellulaires, des cellules ou au niveau moléculaire.

La figure 2.1.1 montre les échelles spatio-temporelles caractéristiques des systèmes vivants. Certains ensembles tels que les molécules, les cellules, les organismes, les populations et les écosystèmes y sont représentés suivant leurs dimensions logarithmiques [Pav94]. On peut noter que l’augmentation des dimensions selon une échelle va de paire avec l’augmentation des dimensions sur l’autre. Ainsi l’étude de la réplication d’une protéine peut être réalisée à l’échelle de la journée, alors qu’en général, si on s’intéresse au devenir d’un organe, il sera nécessaire d’augmenter l’échelle de temps de l’étude.

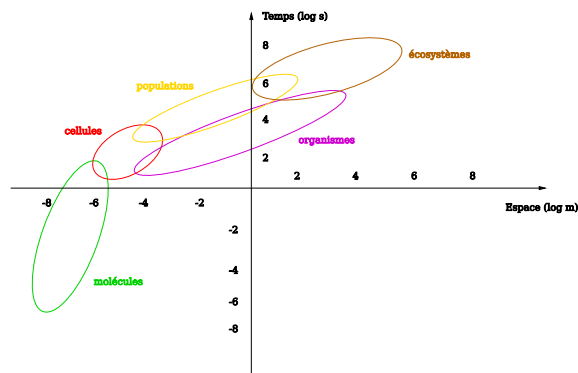


FIG. 2.1.1 – Échelles spatio-temporelles caractéristiques des systèmes vivants [Pav94].

► La modélisation multi-paradigmes

Un paradigme est une représentation du monde, une manière de voir les choses, un modèle cohérent de vision du monde qui repose sur une base définie : matrice disciplinaire, modèle théorique ou courant de pensée.

Cette définition, qui renvoie à la notion de modèle, a évolué dans le contexte de l’activité scientifique pour devenir “*le référentiel d’un système de pensée*” : c’est-à-dire une certaine conception du monde, des enjeux et des méthodes d’une discipline scientifique considérés comme valables par ceux qui en sont les praticiens [Kuh72]. Nous pouvons alors parler d’un paradigme de modélisation comme étant l’ensemble des définitions et formalismes, des méthodes, des outils et des techniques

¹dictionnaire de Wikipédia, 2007

qui caractérisent cette activité de modélisation.

La figure 2.1.2 montre, en les positionnant relativement aux échelles de temps et d'espace pris en compte, différents paradigmes capables de modéliser des systèmes moléculaires. La chimie quantique décrit le comportement électronique des atomes et des molécules par le biais d'équations mathématiques qui modélisent le "cortège électronique". La dynamique moléculaire et les modèles à grains d'atomes calculent l'évolution d'un système de particules en interaction. Enfin il est possible d'opter pour une approche continue avec une équation ou un système d'équations capable de décrire le système moléculaire dans son ensemble.

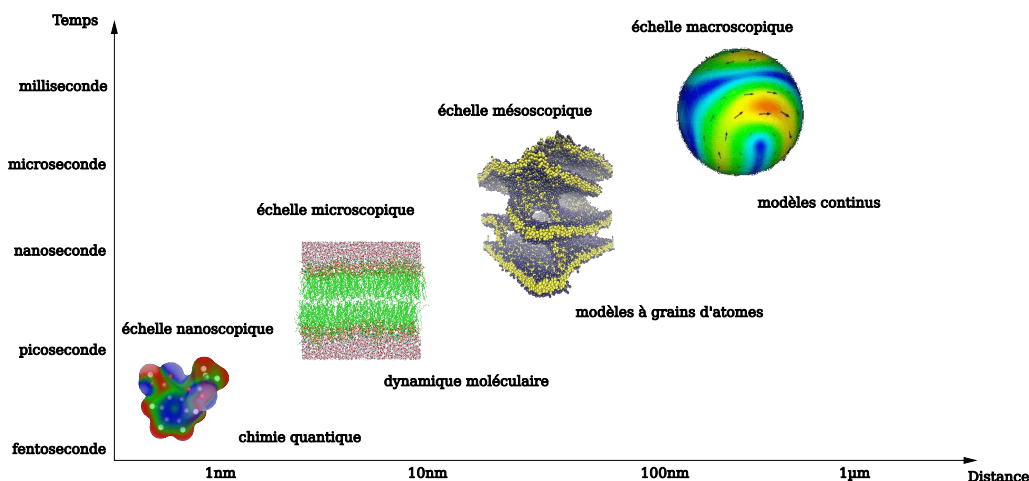


FIG. 2.1.2 – Différents paradigmes pour les systèmes moléculaires.

► Corrélation entre la modélisation multi-échelles et multi-paradigmes

La modélisation multi-échelles et la modélisation multi-paradigmes sont généralement étroitement liées. En terme d'association, plusieurs cas de figure sont possibles :

- Dans une **modélisation multi-échelles**, il est possible de ne faire appel qu'à **un seul paradigme**. Par exemple, dans le cadre des systèmes multi-agents, l'étude de l'émergence a parfois suscité le développement de modèles intégrant les différentes échelles comme autant de paramètres définissant les groupes et les rôles attribués aux mêmes agents (modèle agent, groupe, rôle ou AGR, voir Ferber et al. [FG]).
- A une **même échelle du système**, il est possible de représenter certaines variables selon un **premier paradigme** et d'autres suivant un **second paradigme** différent. Par exemple, les individus d'une population peuvent être modélisés par un Système Multi-Agents (SMA) dans leur déplacement et par des EDO pour ce qui est de leur métabolisme.
- Une **modélisation multi-échelles** d'une problématique biologique peut intégrer des modèles basés sur des **paradigmes différents**. En effet, il est souvent plus simple d'associer le passage d'une échelle à une autre à un changement d'approche. Par exemple, dans un modèle proies/prédateurs, l'interaction locale entre les proies et les prédateurs peut être gérée par le biais d'un SMA tandis qu'une épidémie affectant la population des proies peut-être modélisée par des équations différentielles.

On notera que dans tous les cas de modélisation multi-paradigmes, qu'elle intègre ou non une modélisation multi-échelles, il est nécessaire de prévoir le couplage entre les différents paradigmes, autrement dit la discussion entre les différents types de modèles qui peuvent partager des variables tout en les représentant de manières différentes.

2.1.2 Le couplage de modèles

Parmi les modélisations utilisant le couplage entre différents modèles, nous pouvons donner les exemples suivants qui sont des couplages multi-paradigmes :

➤ couplage entre la mécanique quantique et la dynamique moléculaire pour la modélisation de la propagation de fissures dans les matériaux. L'association de ces deux paradigmes a comme avantage d'augmenter sensiblement les temps simulés.

➤ couplage entre un SMA et une équation différentielle. L'exemple est un couplage de paradigmes pour l'étude du copépode, petit organisme marin appartenant à la famille des zoo-planctons. Pour modéliser l'interaction entre une population de cellules de phytoplancton et une population de copépodes, un premier modèle (SMA) se focalise principalement sur le déplacement du copépode et la prédation tandis qu'un second modèle (EDO) s'occupe uniquement de la gestion de l'énergie dans le métabolisme du copépode. Cette application a été réalisée grâce au formalisme DEVS (*Discret Event system Specification*) qui permet de définir une **interface fonctionnelle** entre différents paradigmes [ZSKP95]. A partir de la définition d'un ensemble de variables de type entrée-sortie pour chaque paradigme, DEVS permet d'établir une communication entre ces paradigmes.

Dans la section suivante, nous examinerons plusieurs paradigmes couramment utilisés en modélisation des systèmes biologiques.

2.2 Différents paradigmes disponibles : du global au local

Il existe de nombreuses théories et approches mathématiques pour modéliser les phénomènes biologiques depuis les modèles à base d'équations différentielles, stochastiques jusqu'aux automates cellulaires.

paradigme	temps	espace	niveau d'abstraction
équation différentielle ordinaire	continu	non pris en compte	global : variables moyennées sur l'ensemble des éléments du système
équation différentielle ordinaire par compartiment	continu	pris en compte (compartimentation de l'espace)	intermédiaire : variables moyennées restreintes aux populations des compartiments
équation différentielle partielle	continu	pris en compte (expressions analytiques simples)	global : variables moyennées sur l'ensemble des éléments du système
modélisation stochastique	généralement continu	généralement non pris en compte	intermédiaire : variables aléatoires correspondantes aux densités de probabilité des éléments du système
modélisation logique	généralement discrétisé (transition d'un état à un autre)	a priori non pris en compte (mais des sous-classes de molécules peuvent être associées à des régions de l'espace)	intermédiaire : regroupement des éléments de même type
réseau de Petri	continu ou discrétisé (transitions)	pris en compte (au travers des places)	intermédiaire : un jeton par élément mais temps (transitions) et espace (places) sont discrétisés
automate cellulaire	généralement discrétisé (mais peut être simulé continu)	nécessairement discrétisé (quadrillage fixe)	intermédiaire : autant d'attributs par cellule (tant que ceux-ci ne dépendent pas de l'espace)
système multi-agents	généralement discrétisé (mais peut être simulé continu)	généralement continu	fin : toutes les singularités des éléments prises en compte

TAB. 2.2.1 – Différents paradigmes et leur relation au temps, à l'espace et aux éléments modélisés.

D'une certaine manière, il est possible de positionner les différents paradigmes les uns par rapport aux autres en se basant sur le niveau d'abstraction des éléments modélisés, depuis le plus global jusqu'au plus local. En ce qui concerne le niveau global lorsque le nombre d'éléments est

suffisant, il est possible de modéliser le système dans son ensemble en manipulant des variables moyennées réelles (et non entières), ce qui parfois amène à parler d'une moitié de cellule ou d'un quart de mouton. A l'inverse, du point de vue le plus local possible, chaque élément du système peut être modélisé dans toute sa singularité. Entre ces deux positions antinomiques, d'autres paradigmes attachent plus ou moins de détail au temps, à l'espace ou aux éléments. Le tableau 2.2.1 résume pour chacun des paradigmes leur niveau de prise en compte du temps et de l'espace ainsi que leur niveau d'abstraction des objets biologiques modélisés.

Nous allons nous consacrer maintenant à la présentation de ces formalismes en les restituant dans le contexte de la biologie.

2.2.1 Modélisation différentielle

En mathématique, une équation différentielle est une **relation entre une ou plusieurs fonctions inconnues et leurs dérivées** (voir l'annexe A.2 pour une définition plus complète). La modélisation de systèmes par équations différentielles est actuellement largement répandue en biologie. Elle comporte certains avantages essentiels :

- l'**approche est formalisée**. En effet, une équation mathématique est universellement compréhensible, des **solutions analytiques** peuvent être trouvées et si ce n'est pas le cas, des **simulations numériques** peuvent être effectuées.
- un système d'équations différentielles permet de **décrire l'évolution d'un système** composé d'un grand nombre d'éléments ou de nombreux types d'interactions entre plusieurs ensembles d'éléments.

Les équations différentielles sont utilisées pour construire des modèles mathématiques de phénomènes physiques et biologiques, par exemple pour l'étude de la radioactivité ou la mécanique céleste. De même, les réseaux de régulation biologique sont classiquement modélisés à l'aide de systèmes d'équations différentielles ordinaires, partielles ou encore stochastiques (voir la revue de H. de Jong [dJ02]).

Dans ce qui suit nous donnons un ensemble de définitions et d'exemples, tirés notamment de l'ouvrage d'A.Pavé [Pav94], permettant de mieux comprendre les systèmes différentiels.

► Les équations différentielles ordinaires

Une équation différentielle ordinaire (EDO) est une équation contenant une variable dite indépendante t , un nombre fini de fonctions de t , appelées variables dépendantes, $y_k(t)$ ou y_k et leurs dérivées par rapport à t , notées $\dot{y}_k(t)$ ou $y'_k(t)$. En biologie, la variable indépendante t représente le plus souvent le temps.

Pour illustrer le champs d'application des EDO, voici certains cas d'utilisation de ces systèmes différentiels :

► La dynamique des populations

On considère une population de taille y et un taux de reproduction $r(t, y)$. Son évolution temporelle est décrite par l'équation différentielle (2.2.1) pour une condition initiale $y(0) = y_0$ donnée.

$$\dot{y} = r(t, y)y \quad (2.2.1)$$

Suivant que l'on désire modéliser une croissance exponentielle ou non, on pourra choisir respectivement pour $r(t, y)$ un taux constant $r(t, y) = r$ ou un modèle logistique de la forme $r(t, y) = (1 - y)$.

► Le modèle proie-prédateur

C'est un modèle d'interaction entre plusieurs espèces. Un modèle simple qui décrit la relation prédateur- proie est le modèle de Lotka (1925) et Volterra (1926) Eq.(2.2.2). Supposons que l'espèce 1 est la proie et l'espèce 2 est le prédateur, on pourra définir l'évolution des populations des deux

espèces à partir des deux équations suivantes :

$$\begin{cases} \dot{y}_1 &= ry_1 - ay_1y_2 \\ \dot{y}_2 &= -dy_2 + by_1y_2 \end{cases} \quad (2.2.2)$$

La première équation décrit l'évolution de la population y_1 en fonction de sa croissance exponentielle ry_1 à laquelle on soustrait le prélèvement des prédateurs $-ay_1y_2$. La seconde équation décrit l'évolution de la population du prédateur y_2 en fonction de sa mortalité $-dy_2$ et de sa capacité à se reproduire en utilisant la population des proies by_1y_2 . Ce modèle peut engendrer des solutions oscillantes entretenues, solutions qu'il est néanmoins difficile d'observer expérimentalement. En effet, une croissance exponentielle suppose un milieu non limitant en substrat et même pour des expériences en laboratoire impliquant deux populations de bactéries où cette condition est remplie, il n'est pas évident d'obtenir un taux de mortalité du prédateur suffisant par rapport au temps de génération de la proie. Toutefois, ce modèle est très souvent utilisé en écologie ou biologie des populations et est considéré comme un bon modèle pour simuler par exemple une épidémie dans une population.

► La cinétique chimique

Considérons une réaction qui permet la production d'une espèce A_2 à partir d'une espèce A_1 ou inversement. Cette réaction peut être décrite par le schéma (2.2.3) où n est un entier positif appelé le coefficient stoechiométrique de A_1 et où k_{21} , k_{12} représentent respectivement les vitesses de réaction dans un sens ou l'autre.



On dénote y_1 , y_2 les concentrations de A_1 et A_2 . Cette réaction se traduit par le système d'EDO (2.2.4).

$$\begin{cases} \dot{y}_1 &= n \cdot k_{12}y_2 - n \cdot k_{21}y_1^n \\ \dot{y}_2 &= -k_{12}y_2 + k_{21}y_1^n \end{cases} \quad (2.2.4)$$

Si on rajoute une réaction consommant une fraction de A_2 , nous avons pour $n = 1$ un processus de la forme (2.2.5). Ce modèle simple peut être utilisé pour représenter la dynamique d'un médicament ou d'une drogue dans le corps humain. Dans ce cas A_1 et A_2 correspondent à la concentration du médicament d'une part dans le tissu et d'autre part dans le sang. Le taux k_0 représente la vitesse à laquelle A_2 est éliminé par les reins. Le système correspondant est donné par (2.2.6).



$$\begin{cases} \dot{y}_1 &= k_{12}y_2 - k_{21}y_1 \\ \dot{y}_2 &= -k_{12}y_2 + k_{21}y_1 - k_0y_2 \end{cases} \quad (2.2.6)$$

Les EDO manipulent des variables moyennées sur l'ensemble du système. Or dans le cas qui fait l'objet de cette étude, la membrane interne mitochondriale, il nous apparaît essentiel d'intégrer la compartimentalisation de l'espace induite par les nombreux replis.

Il existe deux approches pour prendre en compte l'espace dans un système différentiel : soit une compartimentation de l'espace permet à des systèmes d'EDO de se "côtoyer" (EDO par compartiment), soit les variables manipulées sont dépendantes à la fois du temps et de l'espace (EDP).

► Les modèles par compartiment

L'idée de compartimenter un volume découle souvent d'un besoin de prendre en compte l'espace et de différencier certaines régions. Chaque compartiment contiendra alors ses propres ensembles d'éléments dont l'évolution est modélisée par un système d'équations différentielles (généralement EDO). En associant à chaque particularité locale de l'espace un compartiment, il devient aussi

possible de prendre en compte une certaine régionalisation. On gardera en mémoire que dans un modèle par compartiment il est nécessaire de définir les équations décrivant les flux d'éléments d'un compartiment à un autre. L'utilisation d'un modèle compartimenté permet d'intégrer la définition d'un ensemble de paramètres comme par exemple la géométrie des compartiments, les connexions entre ces compartiments, les valeurs initiales des variables dans tous les compartiments et les conditions aux limites des variables.

Par exemple, dans le modèle de Lotka-Volterra que nous venons de présenter, il est possible d'ajouter une segmentation de l'environnement dans lequel évoluent les proies et les prédateurs, cette segmentation étant associée à des discontinuités locales de l'environnement.

De tels modèles conduisent très souvent à l'explosion du nombre de paramètres à prendre en compte dans le système d'équations et rendent d'autant plus difficile la recherche d'une solution analytique ou même la mise en oeuvre de simulations.

Le paragraphe suivant présente une autre solution à la prise en compte de caractéristiques liées à une spatialisation des phénomènes étudiés.

► Les équations différentielles partielles

Les équations aux dérivées partielles (EDP) sont utilisées lorsque l'on désire intégrer au modèle plus d'une variable indépendante. Ceci permet donc de modéliser à la fois l'espace et le temps comme des variables indépendantes. Lorsqu'une solution analytique existe, il deviendra possible de définir un gradient de concentration des différents éléments du système. Alors que les ensembles de solutions d'une équation différentielle ordinaire sont paramétrés par les conditions initiales, dans le cas des EDP les conditions aux limites se présentent plutôt sous la forme de fonctions. Malheureusement cette technique conduit également à l'accroissement du nombre de solutions possibles.

Dans le contexte de la modélisation des systèmes vivants, il est difficile de connaître avec précision les valeurs des paramètres nécessaires ainsi que la forme des fonctions qui caractérisent ce système. Par ailleurs, la spatialisation proposée par les EDP est limitée par l'utilisation de variables moyennées. Aussi, il peut être intéressant de se tourner vers des modèles qualitatifs pour explorer d'autres possibilités de modélisation.

2.2.2 Modélisation logique des réseaux génétiques

Les équations logiques utilisent des variables booléennes ou multi-valuées [GK73] [TTK95]. Le développement de modèles logiques pour différents réseaux biologiques a déjà abouti à d'intéressantes avancées en terme d'analyse de structure de réseaux biologiques (comme les circuit "*feedback*" ou de rétroaction) et des propriétés dynamiques correspondantes [TTK95]. En utilisant l'approche logique généralisée de R.Thomas, D. Thieffry et al. ont développé un logiciel, GIN-sim, qui permet à un biologiste de spécifier un modèle de régulation et de tester l'évolution du système étant donné des états initiaux [CRMT03].

Cependant, comme le nombre d'états (logiques ou discrets) possibles du système augmente exponentiellement avec le nombre d'éléments impliqués dans le réseau biologique, il est souvent nécessaire d'opter pour d'autres approches capables de prendre en compte des réseaux biologiques sans avoir recours à une discrétisation du système et au dénombrement de ses états possibles.

2.2.3 Les réseaux de Petri

Les réseaux de Petri, proposés pour la première fois par Carl Adam Petri [Pet66], sont une des nombreuses représentations mathématiques d'un système discret distribué.

Un réseau de Petri décrit graphiquement la structure d'un système distribué comme un **graphe direct biparti avec annotations** (Fig.2.2.1(a)). Il possède des noeuds **places**, des noeuds

transitions et des arcs orientés connectant les places aux transitions.

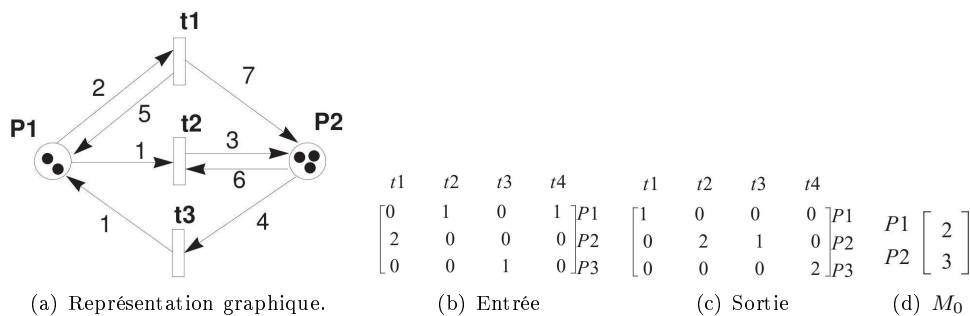


FIG. 2.2.1 – Une réseau de Petri.

Formellement, un réseau de Petri peut être représenté sous la forme d'un quadruplé (P, T, E, S) où

- P est un ensemble fini de **places** ;
- T un ensemble fini de **transitions** ;
- E est une application $E : P \times T \rightarrow \mathbb{N}$ appelée **application d'incidence avant** (Fig.2.2.1(b)) ;
- S est une autre application $S : P \times T \rightarrow \mathbb{N}$ appelée **application d'incidence arrière** (Fig.2.2.1(c)).

Le marquage M d'un réseau correspond à son état. Formellement, un marquage est une application $M : P \rightarrow \mathbb{N}$ donnant pour chaque place le nombre de jetons qu'elle contient. Le marquage initial est généralement noté M_0 (Fig.2.2.1(d)).

Un réseau de Petri évolue lorsqu'on exécute une transition : des jetons sont pris dans les places en entrée de cette transition et envoyés dans les places en sortie de cette transition. L'exécution d'une transition (on parle alors de son franchissement) est une opération indivisible qui est conditionnée par la présence de jeton sur la place d'entrée. L'exécution d'un réseau de Petri est **non déterministe**, car il peut y avoir à un instant donné autant de possibilités d'évolution que de transitions franchissables à cet instant.

Les recherches sur les réseaux de Petri ont une longue histoire de presque 40 ans dans la plupart des domaines liés à la simulation. Les premiers cas d'utilisation de réseaux de Petri pour modéliser les **voies métaboliques** ont été réalisés par Reddy et al. [RML93] et Hofstadt [Hof94]. Genrich et al. [GKV01] ont modélisé des voies métaboliques en utilisant des réseaux de Petri spécifiques, appelés réseaux de Petri colorés, en assignant les vitesses de réactions enzymatiques aux transitions. Voss et al. [VHK03] ont utilisé différemment ces mêmes réseaux de Petri colorés pour réaliser une analyse qualitative des états d'équilibre des voies métaboliques. Les réseaux de Petri stochastiques ont été utilisés pour modéliser une grande variété de voies métaboliques : la réplication des plasmides ColE1 [GP98], la réponse du facteur de transcription $\sigma32$ à un choc thermique [SPB01] et les interactions cinétiques d'une invasion virale [SYSY02]. Miyano et al. ont modélisé le réseau de régulation génétique du phage λ avec un formalisme hybride de réseaux de Petri mêlant les dynamiques discrètes et continues [MDNM00].

Les réseaux de Petri ont montré leur efficacité en terme de simulation discrète (ou non) des mécanismes réactionnels en biologie. Toutefois les informations sur la conformation des molécules ou leurs interactions tridimensionnelles restent extrêmement complexes à spécifier dans un réseau de Petri.

Nous allons maintenant présenter les systèmes multi-agents et leurs ancêtres les automates cellulaires qui intègrent plus aisément ces données.

2.2.4 Les automates cellulaires

Dans les années 40, John von Neumann, travaillant sur les systèmes auto-réplicatifs (inspiré par les travaux de Stanislaw Ulam sur la croissance des cristaux) produisit le premier automate

cellulaire (AC), le **copieur et constructeur universel**, basé sur une grille à deux dimensions où chaque cellule pouvait prendre 29 états. Von Neumann y construisit un motif particulier qui pouvait reproduire sans fin des copies de lui même [NB666]. Dans les années 1970, un automate cellulaire à deux dimensions et deux états nommé **le jeu de la vie**, conçu par John Conway, connut un grand succès.

Un automate cellulaire consiste en une grille de **cellules** pouvant chacune prendre à un instant donné un **état** parmi un ensemble fini d'états. Le temps est discrétisé et l'état d'une cellule A au temps $t+1$ est fonction de l'état au temps t d'un nombre fini de cellules définissant le **voisinage** de A. À chaque nouvelle unité de temps, les mêmes règles sont appliquées à toutes les cellules de la grille, produisant une nouvelle **génération** de cellules dépendant entièrement de la génération précédente.

Formellement, un AC peut être défini comme un **quadruplé** $(d, \mathbb{Q}, \mathbb{V}, \delta)$ où :

- d est la dimension de l'automate, et \mathbb{Z}^d l'espace discret de dim. d ;
- \mathbb{Q} est son alphabet ;
- $\mathbb{V} \subseteq \mathbb{Z}^d$ est son voisinage (un sous-ensemble fini de \mathbb{Z}^d) ;
- $\delta : \mathbb{V}^a \rightarrow \mathbb{V}$ est la **règle locale de transition** (ou changement d'état des cellules) avec $a = |\mathbb{V}|$ est l'arité de l'automate.

L'automate cellulaire le plus simple que l'on puisse concevoir consiste en une grille unidimensionnelle ($d = 1$) de cellules ne pouvant prendre que deux états : 0 ou 1, $\mathbb{Q} = \{0, 1\}$, avec un voisinage constitué pour chaque cellule d'elle-même et des deux cellules qui lui sont adjacentes. Pour la cellule i , le voisinage \mathbb{V} est égal à $\{i-1, i, i+1\}$ et l'arité de l'automate a est égal à 3. Chacune des cellules pouvant prendre deux états, il existe $2^3 = 8$ configurations (ou motifs) possibles d'un tel voisinage. Pour que l'automate cellulaire fonctionne, il faut définir la règle locale de transition δ , c'est-à-dire l'état, à la génération suivante, d'une cellule pour chacun de ces motifs. Il y a $2^8 = 256$ façons différentes de s'y prendre, soit 256 automates cellulaires différents de ce type. Les automates de cette famille sont dit **élémentaires**. On les désigne souvent par un entier entre 0 et 255 dont la représentation binaire est la suite des états pris par l'automate sur les motifs successifs 111, 110, 101, etc.

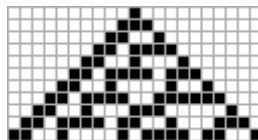
À titre d'exemple, considérons l'automate cellulaire défini par le tableau de la figure 2.2.2(a). Dans ce tableau, la première ligne contient la description des 8 voisinages possibles d'une cellule i sous la forme d'un motif $i-1, i, i+1$ dans l'espace à 1 dimension. La seconde ligne donne la valeur de la cellule i au pas de temps suivant. La règle d'évolution donnée dans le tableau signifie que si par exemple, à un temps t donné, une cellule i est à l'état 1, sa voisine de gauche à l'état 1 et sa voisine de droite à l'état 0, au temps $t+1$ cette cellule i sera à l'état 0.

Par convention cette règle de transition locale est nommée la **règle 30**, car 30 s'écrit 00011110 en binaire, nombre binaire inscrit dans la seconde ligne du tableau. Si l'on part d'une grille initiale où toutes les cellules sont à l'état 0 sauf une, on aboutit à la figure 2.2.2(b) où chaque ligne est le résultat de la ligne précédente.

Les **motifs de certains coquillages**, comme les cônes et les cymbiolae, sont générés par des mécanismes s'apparentant au modèle des automates cellulaires. Les cellules responsables de la pigmentation sont situées sur une bande étroite le long de la bouche du coquillage. Chaque cellule

motif initial	111	110	101	100	011	010	001	000
valeur suivante de la cellule centrale	0	0	0	1	1	1	1	0

(a) Règle 30.



(b) Résultat de la règle 30.

(c) *Conus textile* présentant sur sa coquille un motif similaire.

FIG. 2.2.2 – Automate cellulaire et phénomène biologique.

sécrète des pigments selon la sécrétion (ou l'absence de sécrétion) de ses voisines et l'ensemble des cellules produit le motif de la coquille au fur et à mesure de sa croissance. Par exemple, l'espèce *Conus textile* (Fig.2.2.2(c)) présente un motif ressemblant à la règle 30 précédemment décrite.

Stephen Wolfram [Wol84] a étudié les 256 règles possibles et a proposé une première **classification** basée non pas sur les règles de l'automate cellulaire mais sur le devenir général de configurations initialement aléatoires :

- Classe I : presque toute configuration initiale conduit à un état homogène.
- Classe II : des structures stables ou périodiques émergent.
- Classe III : des comportements chaotiques apparaissent avec des motifs aperiodiques.
- Classe IV : émergence de structures complexes capables de se propager. Le jeu de la Vie en est un parfait exemple.

Une alternative à cette classification est la classification d'Eppstein qui se focalise sur l'évolution des systèmes pour des configurations en fonction de leur état initial.

► Le jeu de la vie

Le jeu de la vie, automate cellulaire imaginé par John Horton Conway en 1970, est probablement, à l'heure actuelle, le plus connu de tous les automates cellulaires.

Cet automate cellulaire est une grille bidimensionnelle où chaque cellule peut prendre deux valeurs (0 ou 1, mais on parle plutôt de **vivante** ou **morte**) et où son état futur est déterminé par son état actuel et par le nombre de cellules qui ont comme valeur **vivante** parmi les huit qui l'entourent. La règle d'évolution précise que si une cellule vivante est entourée par deux ou trois cellules vivantes, elle reste en vie à la génération suivante, sinon elle meurt. Par ailleurs si une cellule morte est entourée par exactement trois cellules vivantes, elle naît à la génération suivante. En apparence simples, ces règles font **émerger une forte complexité** : structures stables, périodiques, planeurs (ou *gliders*), etc.

La figure 2.2.3 montre, à partir d'une initialisation aléatoire des états des cellules, comment le système évolue pour aboutir à une densité moindre de cellules vivantes tout en exhibant des sous-systèmes locaux stables, oscillants ou se déplaçant. La figure 2.2.3(c) donne les transitions d'états d'un planeur et précise que le déplacement n'est qu'apparent puisque les cellules ne se déplacent pas réellement.

On peut sur le même principe imaginer un grand nombre de variantes du jeu de la vie en modifiant les seuils de naissance ou de survie, ou en ajoutant des états supplémentaires, etc. On peut aussi ne prendre en compte que les voisins immédiats de la case considérée pour déterminer les changements d'états d'une cellule, ou encore compter les états des cellules voisines dans un rayon de deux cases, voire plus. On peut également attribuer un poids plus important à certaines cases du voisinage qu'à d'autres.

En terme d'application à la biologie, les automates cellulaires ont été utilisés pour l'étude

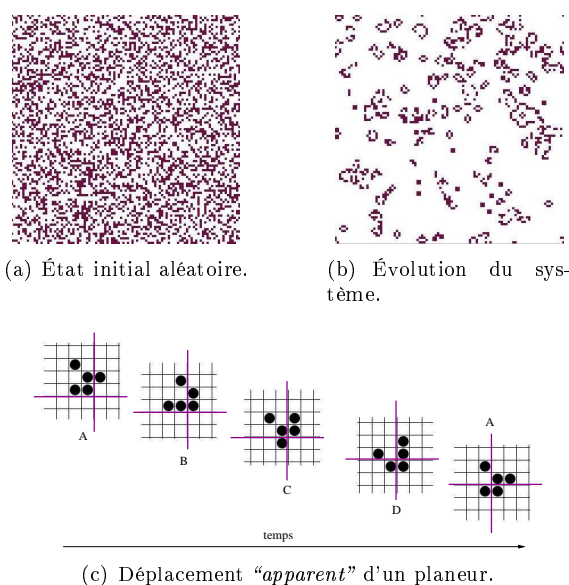


FIG. 2.2.3 – “Life” de John Conway.

du **réseau idiotypique**² [dBvdLH93] des **réponses immunitaires** [CS96] et de l'**évolution du système immunologique** [Bez01]. Fleischer et al. dans [FB93] proposent un modèle d'**organismes multi-cellulaires** où un automate cellulaire, couplé à un SMA, gère la diffusion moléculaire. Mobbey et al. dans [MCS⁺03] ont étendu un modèle basé sur un automate cellulaire stochastique afin de réaliser une étude 2D de l'**agrégation de prions sur les surfaces neuronales**. Ce modèle leur permet de simuler à la fois les feuilletts β et les liaisons faibles entre protéines.

Comme nous venons de le voir, les AC permettent de représenter aisément des entités (les cellules) positionnées dans un espace (2D). Toutefois le fait que ces cellules restent fixes dans la grille limite les capacités à réellement exprimer un déplacement d'objet dans cet espace. Nous allons maintenant nous intéresser aux Systèmes Multi-Agents (SMA) qui fournissent une réponse plus adéquate à la modélisation de mouvements.

2.3 Les Systèmes Multi-Agents

Les Systèmes Multi-Agents (SMA) sont des **modèles orientés individus** (*individual-based model*). Dans ces modèles, les simulations sont basées sur les conséquences globales d'interactions locales entre les membres d'une population. Ces individus peuvent représenter des plantes ou des animaux dans un écosystème, des véhicules dans la circulation, des personnes dans une foule, ou des acteurs dans une animation ou un jeu. Les modèles orientés individus sont aussi appelés "*orientés entités*" ou "*orientés agents*".

Un SMA est un modèle informatique composé d'agents, c'est-à-dire d'entités informatiques autonomes capables d'agir sur elles-mêmes et sur leur environnement. Dans un univers multi-agents, l'agent peut communiquer avec d'autres agents. Son comportement est alors la conséquence de ses observations, de ses compétences et des interactions avec les autres agents [Fer95]. Le temps est directement intégré sous la forme d'une horloge qui rythme les **cycles de vie** des agents, c'est-à-dire des séquences d'actions qui leur sont propres. Les SMA sont donc des **modèles essentiellement d'interactions**, la dynamique du phénomène étudié étant le produit de ces interactions.

Certains de ces modèles sont spatialement explicites, ce qui signifie que chaque individu est associé à un emplacement dans un espace géométrique. Dans ce contexte, les agents seront autorisés à se déplacer (si cela est pertinent). Dans une simulation écologique, on pourra avoir par exemple un animal mobile avec des plantes immobiles. Les modèles spatialement explicites peuvent utiliser des espaces continus (coordonnées exprimées en valeurs réelles) ou discrets (grilles et positions exprimées en valeurs entières). D'autres modèles sont spatialement implicites : par exemple, la simulation d'un réseau d'ordinateurs peut utiliser un modèle individus centrés où chaque ordinateur connecté est un agent sans que sa localisation soit pertinente.

2.3.1 L'émergence

La notion d'émergence est très étroitement liée aux SMA en tant que systèmes complexes d'entités distribuées en interaction locale. Bien que l'objet de nombreuses controverses, on peut retenir comme définition de l'émergence qu'elle désigne l'**apparition de nouvelles caractéristiques ou structures** à un certain degré de complexité, ce qui peut être défini comme suit :

- le comportement global du système ne peut être caractérisé à partir de l'analyse de ses parties. Ce qui est généralement résumé par l'expression "*le tout est plus que la somme de ses parties*";
- la seule analyse des parties du système ne permet pas de prédire le comportement global.

On a tendance à distinguer deux types d'émergence, l'émergence faible et l'émergence forte [Lr005]. On parle d'**émergence "faible"** lorsque des structures ou propriétés émergentes sont identifiées comme telles par un observateur extérieur. Dans le cas des molécules d'eau, leur cristallisation

²Le réseau idiotypique regroupe l'ensemble des mécanismes de régulation de la réponse anticorps grâce à la production d'anticorps anti-idiotypes dirigés contre les anticorps produits en réponse contre un antigène, puis contre ces anticorps anti-idiotypes.

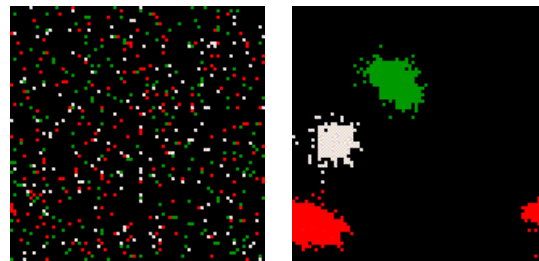
en glace est un phénomène qui n'appartient ni à l'hydrogène, ni à l'oxygène, ses constituants. Il s'agit toutefois d'un phénomène qui peut s'expliquer à partir des propriétés de l'hydrogène et de l'oxygène. Il n'y a donc pas de propriété réellement nouvelle, si ce n'est celle qu'un humain peut observer.

L'**émergence "forte"** concerne des propriétés intrinsèques au système et indépendantes de toute observation. Ces nouvelles caractéristiques vont interagir avec les autres constituants du système de manière originale. Il n'existe alors plus de lien causal entre les constituants de la structure émergente et ses propres propriétés. On peut citer comme exemple d'émergence forte la fonction assurée par un organe alors qu'elle résulte d'une interaction complexe entre des molécules. C'est l'existence de telles propriétés émergentes fortes qui est discutée car on pourrait considérer que l'impossibilité de voir le lien entre le niveau micro et le niveau macro résulte des limites de l'être humain.

► L'exemple des termites

La figure 2.3.1 montre un exemple bien connu de tri de grains (de différentes couleurs) par un ensemble de termites : à partir d'un état initial aléatoire 2.3.1(a), on constate la formation de tas de couleur uniforme après un certain nombre de pas de temps. Il s'agit ici d'une émergence dite faible dont le résultat est obtenu à partir de deux règles simples :

- Si un agent ne porte rien et se trouve à proximité d'un grain, alors il prend un grain quelle que soit sa couleur.
- Si l'agent porte déjà un grain et passe à proximité d'un grain de même couleur, alors il le pose à côté.



(a) État initial aléatoire. (b) Évolution du système.

FIG. 2.3.1 – Exemple d'émergence.

Les tas obtenus sont dus à un effet cumulatif des comportements individuels au même endroit de l'environnement, donc à des interactions indirectes. La notion "*complexe*" de tas, quant à elle, n'est pas présente au niveau de l'individu et pourtant elle apparaît à l'échelle de la population (ie de l'observation), et ce, sans aucune intervention d'un agent superviseur qui disposerait d'une perception (ie d'une représentation) globale de l'environnement.

Nous allons maintenant détailler les différents éléments constitutifs d'un système multi-agents.

2.3.2 Les modèles d'agents

Un agent est défini par un ensemble d'attributs, c'est-à-dire une collection de variables située dans le temps et dans l'espace et par des comportements, c'est-à-dire des mécanismes de modification de ces variables. Le cycle de vie d'un agent est constitué de trois étapes successives : perception de son environnement, décision, action. Ce cycle de vie est réitéré récursivement à chaque pas de la simulation. A ce niveau il est d'usage de distinguer deux catégories d'agents : les **agents réactifs** et les **agents cognitifs**. Cette classification est basée sur le degré de cognition ou la complexité des processus qui aboutissent à une action.

► Les agents réactifs

Les agents réactifs peuvent être vus comme des entités autonomes qui perçoivent leur environnement et agissent sur lui ou sur d'autres agents de façon "*réflexe*". De cette définition relativement réductrice, nous pouvons tirer deux caractéristiques essentielles :

- l'agent a une individualité, un comportement propre,

- il est capable d’interagir.

La **perception** est le processus par lequel l’agent acquiert des informations sur le monde qui l’entoure. La perception peut être vue comme la réception passive par l’agent d’un évènement provenant de l’extérieur de l’agent. Dans certains cas, cette réception est liée à une action antérieure de l’agent qui a “*interrogé*” son environnement, par exemple pour connaître son voisinage.

Dans un SMA réactif, l’action d’un agent se résume par l’*influence* sur son environnement ou sur les autres agents. Elle peut être de différentes natures : déplacement, destruction ou modification d’un objet de l’environnement, dépôt d’informations etc. En général, on distingue les notions d’action ou de réaction qui peuvent être considérées comme une interface fonctionnelle de l’agent vis-à-vis de l’extérieur (l’environnement et les autres agents) et les fonctions internes à l’agent. Les concepts d’action et réaction sont à mettre en relation avec celui d’**autonomie**, également caractérisée par toutes les transitions internes de l’agent.

► Les agents cognitifs

Les agents cognitifs peuvent être perçus comme des agents réactifs augmentés de **mémoire** et d’**états mentaux** avec des fonctions de transitions comme la **cognition**, c’est-à-dire un système de règles agissant sur cette mémoire et ces états mentaux. C’est dans ce contexte que les notions de décision et de planification d’actions par des agents prennent tout leur sens.

B. Schattenberg et A.M Uhrmacher [SU01] ont utilisé cette approche pour la spécification d’agents planificateurs en robotique. Ce domaine fournit le plus grand nombre d’exemples d’applications de SMA cognitifs pour résoudre des problèmes de collaboration entre robots (parfois spécialisés pour certaines tâches), de gestion de processus globaux voire d’interaction homme/robot.

Dans le contexte de la modélisation d’objets biologiques, on se limite en général à l’utilisation d’agents réactifs.

2.3.3 La gestion de l’environnement : l’espace

L’environnement est une composante essentielle d’un SMA. Il possède au moins les caractéristiques suivantes :

- une métrique utilisée comme référentiel spatial pour l’ensemble des agents ;
- un espace possédant généralement des limites ou des frontières ;
- des entités passives (sans comportement) ou des paramètres le caractérisant ;

L’environnement est le siège des interactions et des communications indirectes. Par exemple il pourra contenir les traces volatiles déposées par des agents au cours de leurs déplacements dans l’environnement. Ces traces peuvent alors être perçues par d’autres agents qui modifieront leur comportement en conséquence. Cette représentation indirecte des interactions transitant par l’environnement peut mener à des comportements émergents de l’ensemble des agents [D93]. Enfin, dans la mesure où les agents ont une perception limitée de l’environnement [Fer95], celui-ci conditionne de fait les interactions entre les agents situés, comme par exemple dans le cadre de la détection du voisinage. Cette représentation de l’espace est une des avancées majeures apportées par les SMA. En effet, elle rend possible la modélisation d’interactions discrètes, locales et ponctuelles.

► Le nombre de dimensions de l’espace

Une espace en 2 dimensions (2D) simplifie généralement les calculs en terme par exemple de superposition de structures ou de détection de collisions. Cependant un espace en 3 dimensions (3D) est souvent préférable et ce pour plusieurs raisons.

D’un point de vue théorique, certains phénomènes n’ont pas les mêmes propriétés en 2D et en 3D. Prenons l’exemple de la marche aléatoire et de la propriété de récurrence. Considérons une marche aléatoire isotrope sur le réseau (ou maille) \mathbb{Z}^d à d dimensions spatiales. On peut toujours

choisir de prendre le point de départ de cette marche comme origine O du système de coordonnées cartésiennes. La question de la récurrence consiste alors à se demander si on peut trouver au moins un instant t positif fini pour lequel la particule repasse par l'origine O [Sdw05]. La marche aléatoire sera dite récurrente si et seulement si la probabilité que la particule repasse à l'origine O pour un certain instant t ultérieur fini vaut 1. Or cette propriété de récurrence dépend fortement de la dimension de l'espace (théorème de Pólya [Pól21]) : pour $d = 1$ et $d = 2$, la marche aléatoire isotrope est récurrente tandis que pour $d = 3$ et au-delà, la marche aléatoire isotrope n'est pas récurrente (on dit alors qu'elle est transitoire).

D'un point de vue pratique, il est parfois essentiel de pouvoir simuler des structures tridimensionnelles composées d'objets volumineux engendrant un encombrement donné de l'espace. Par exemple, la modélisation de la rencontre d'un complexe enzymatique et d'un métabolite peut être fortement dépendante de la conformation tridimensionnelle de ce complexe. De même, l'émergence d'une membrane biologique à partir de phospholipides peut dépendre fortement du nombre de dimensions modélisées. Par ailleurs, la visualisation en 3D peut se révéler plus informative lorsqu'on modélise des systèmes biologiques.

► La gestion des frontières

➤ Un espace borné

Lorsqu'on utilise un espace borné, c'est-à-dire fini et limité, il est nécessaire de prendre en compte les frontières de cet espace. Par exemple, les agents rencontrant une frontière lors de leur déplacement peuvent rebondir suivant un angle égal à l'angle d'incidence ou tout simplement s'arrêter au contact de la frontière. En général, il sera nécessaire de définir un comportement particulier des agents pour la modélisation de ces frontières.

L'utilisation d'un espace borné se justifie principalement en cas de présence de structures délimitant le système modélisé (comme par exemple une membrane), les frontières pouvant alors représenter des faces infranchissables du volume simulé.

➤ Un espace clos

Dans les modélisations cellulaires on modélise très souvent l'espace comme un monde clos. Pour obtenir cette propriété de l'espace, la solution la plus communément utilisée est de modéliser un espace torique par le biais d'un parallélépipède rectangle (ou pavé droit de l'espace) à faces jointives [G9797]. Dans un espace torique on peut calculer une aire ou un volume (comme pour un espace clos) mais aucune frontière n'est susceptible de contrarier le déplacement d'un agent. Les conditions aux frontières de l'espace caractérisent la nature torique du volume : lorsqu'un agent sort par une face, il est réintroduit dans le volume par la face opposée. D'un point de vue algorithmique, l'implémentation de ce comportement ne pose aucune difficulté.

► Les méthodes de discrétisation de l'espace

Comme nous l'avons dit, à chaque pas de temps, un agent, dans un SMA doit pouvoir connaître son voisinage. L'optimisation de cette opération conditionne grandement les performances des simulations. Une prise en compte de l'ensemble des agents pour la recherche d'un voisinage correspond à une complexité quadratique $O(n^2)$ où n est le nombre d'agents. Or il est possible d'améliorer la complexité de cette recherche des voisins, puisque par définition, les interactions entre les agents d'un SMA sont locales.

Le partitionnement de l'espace doit permettre une présélection pertinente des agents avant de calculer réellement les distances. Cette présélection permettra de distinguer les agents qui sont trop éloignés les uns des autres pour interagir, et ce, sans faire de calcul de distance.

D'une manière générale, le partitionnement de l'espace cherche à :

- diviser l'espace en au moins deux partitions disjointes ;
- attribuer à chaque agent une partition ;
- éventuellement établir un système hiérarchique modélisant d'une certaine manière la proximité relative des partitions entre-elles.

La problématique du partitionnement se rencontre dans de nombreux domaines³. Le choix d'une méthode de partitionnement devra prendre en compte les propriétés du système simulé, comme la répartition des agents dans le volume ainsi que l'ampleur de leurs déplacements. Par exemple, une distribution homogène d'agents sur tout l'espace avec des changements réguliers de leurs positions diffère beaucoup d'un regroupement d'agents localisés en certains points de l'espace avec des mises à jours ponctuelles et individuelles.

L'optimisation reste un compromis entre :

- le coût supplémentaire de l'attribution des partitions à chaque agent à chaque nouvelle itération de la simulation,
- le gain de temps réalisé pour une requête multiplié par le nombre de requêtes effectuées pendant une itération.

La figure 2.3.2 montre comment les différentes méthodes de partitionnement peuvent être classées en fonction de leurs caractéristiques.

Ces méthodes de partitionnement peuvent ou non tenir compte de la **répartition des éléments** dans l'espace. Les méthodes dites "fixes" utilisent des positions prédéterminées pour les hyperplans séparateurs là où les méthodes dites "adaptables" calculent les positions des hyperplans en fonction de celles des éléments dans l'espace, la construction étant dynamique et non prédictible.

Un autre caractéristique importante est la portée ou l'**étendue de coupe** des hyperplans. Les méthodes *grille* auront tendance à découper tout l'espace et la représentation des données se fera sous forme de tableaux à plusieurs dimensions. Les méthodes dites *brickwall* limiteront l'extension des hyperplans dans la partition à diviser. On préférera alors une représentation hiérarchique des données sous forme d'arbre.

Enfin, le **nombre d'hyperplans** et leurs dimensions permettent de distinguer les méthodes de partitionnement. Certaines méthodes n'utiliseront qu'un hyperplan à la fois, généralement de dimension $d - 1$ pour diviser un espace de dimensions d . D'autres en revanche feront appel à k hyperplans, généralement $k = d$, toujours de dimension $d - 1$.

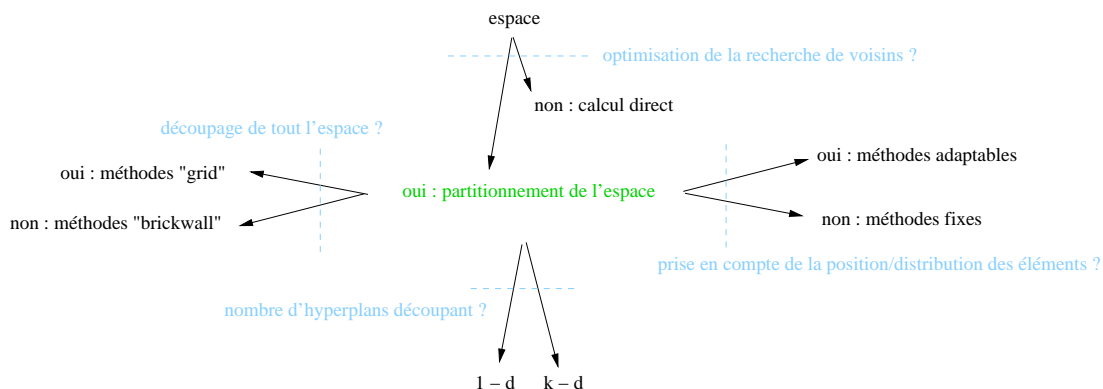


FIG. 2.3.2 – Différentes classifications des méthodes de partitionnement de l'espace.

> **La grille** : le partitionnement de l'espace dans une grille est régulier, a priori fixe, non hiérarchique et utilise simultanément $d * n$ hyperplans pour diviser un espace à d dimensions en n^d partitions, carrés ou voxels.

> **Les arbres** : les techniques dites hiérarchiques souvent représentées par des arbres sont capables de tirer partie avantageusement de la répartition spatiale des éléments. Néanmoins, dans le cas d'un système dynamique et homogène, les opérations d'insertion ou de mise à jour de l'arbre deviennent trop coûteuses en temps par rapport au gain réalisé pour la recherche des voisins.

³L'annexe B.4 expose différentes méthodes à partir d'arbres ou de grilles couramment utilisées pour le partitionnement de l'espace.

La section suivante s'intéresse à la prise en compte du temps dans un SMA.

2.3.4 La gestion de l'environnement : le temps

Pour la simulation de phénomènes physiques ou biologiques, la question de la prise en compte du temps est une question importante. En effet, d'une part on souhaite le plus souvent réaliser des simulations qui se déroulent linéairement en fonction du temps et d'autre part les SMA induisent une modélisation discrète du temps (contrairement à la modélisation continue des équations différentielles). Dans ce contexte, un des éléments de cette discrétisation est le réglage du pas de temps qui pourra être suivant les situations un pas de temps fixe ou variable. Il faudra aussi se poser la question de l'adéquation entre la taille de ce pas de temps et le temps simulé souhaité. Cette question étant très corrélée à l'application, nous discuterons de ces choix lors du chapitre 4.

Par principe, un SMA suppose une cohérence temporelle des interactions entre les agents. C'est à dire qu'il est nécessaire que les agents impliqués dans ces interactions se trouvent dans le même état d'avancement de la simulation. Dans le cas contraire, il faudrait soit revenir dans le passé, soit attendre que les agents "*en retard*" rattrapent les autres. Les phénomènes émergents s'expriment souvent en terme d'organisation spatiale. Or pour maintenir une cohérence des positions des différents agents, il faut assurer la cohérence temporelle de l'ensemble du système sur toute la durée de la simulation.

Dans la section suivante nous présentons deux méthodes de gestion du temps : le synchronisme et l'asynchronisme. La différence entre ces deux approches est simple et concerne la réciprocity des actions.

► Le synchronisme

Dans un SMA géré de manière synchrone, tous les agents doivent être dans des situations similaires, c'est-à-dire que les interactions doivent être traitées simultanément car symétriques. Dans le cas d'interactions entre molécules par exemple, rien ne permettrait de justifier physiquement une dissymétrie. D'un point de vue informatique, sachant qu'un ordinateur travaille le plus souvent en séquence et non en parallèle, il n'est pas possible de traiter simultanément plusieurs agents. Mais une solution simple est communément utilisée : faire évoluer en parallèle deux états du système, un état présent et un état futur. Les interactions tiendront compte des éléments du système à un temps t pour produire des effets qui resteront mémorisés jusqu'à ce que tous les agents aient été traités et que tout le système bascule au temps $t + \delta t$. Ce mécanisme est similaire au "*double buffering*". Bien sûr la mise en place de ce *double buffering* est coûteux en espace mémoire et peut pénaliser significativement les capacités de simulation du système.

► L'asynchronisme

Dans un SMA géré de manière asynchrone, on autorise des dissimilarités dans les interactions. Autrement dit, la mise à jour d'un agent induit immédiatement un changement de la configuration de l'environnement des autres agents. Si dans un premier temps cette approche semble peu réaliste, puisqu'elle suggère qu'il n'y ait plus d'équivalence entre action et réaction, le processus moyenné sur un grand nombre d'éléments et avec des pas de temps réduits peut n'interférer que légèrement avec les phénomènes simulés. Dès lors, il s'avère moins coûteux en terme d'espace mémoire de réaliser des simulations sans contrainte de *double buffering*, c'est-à-dire en ne stockant qu'un seul état du système, l'état au temps t .

Pour ce qui est de l'asynchronisme, on peut distinguer deux approches :

- soit tous les agents sont mis à jour exactement une seule fois à chaque pas de la simulation. L'asynchronisme se traduit ici par un réordonnancement systématique des "appels" de mise à jour des agents par tirage aléatoire sur l'ensemble des agents à mettre à jour.
- soit un (seul) agent est mis à jour à chaque pas de temps de la simulation avec une sélection aléatoire de l'agent à mettre à jour (sélection a priori uniformément aléatoire sur l'ensemble

des agents). Aussi, même si aucun contrôle n'est porté sur le nombre de mises à jour effectuées par agent, ce processus garantit un nombre moyen identique de mise à jour pour tous les agents sur un temps suffisamment long.

2.3.5 Intérêt de la modélisation SMA pour les processus biologiques

Les SMA correspondent à un paradigme de modélisation "*orienté individu*" ou encore "*basé sur les particules*". Pour prendre en compte toutes les singularités des objets biologiques évoqués en chapitre 1, il nous semble approprié de concevoir un SMA constitué d'agents réactifs capables d'interagir à la manière des phospholipides et des protéines intramembranaires, puisque nous pourrions conserver des informations :

- de conformation tridimensionnel et donc caractériser les sites actifs ;
- d'orientation dans l'espace pour modéliser la disponibilité des sites actifs ;
- et de dynamique interne induisant des modifications de la conformation.

Pour ce qui est de la gestion de l'environnement par les SMA, nous pouvons relever un certain nombre de points importants :

- un espace à 2 dimensions peut sembler plus simple à manipuler, néanmoins il est préférable de modéliser un espace en 3 dimensions pour éviter des biais dans les phénomènes émergents ou tout simplement faciliter la visualisation du système ;
- la gestion des frontières peut également induire des biais de simulation or avec un volume torique on a l'avantage de ne pas présenter de discontinuité de l'espace ;
- les interactions entre les agents situés d'un SMA dépendent de la distance entre ces agents. Plusieurs techniques existent pour partitionner l'espace et limiter le nombre de distances à réellement calculer. Le choix entre ces différentes discrétisations doit tenir compte de la répartition des agents dans l'espace et de leurs déplacements.

En ce qui concerne la gestion du temps, les SMA requièrent la discrétisation du temps sous la forme d'un ensemble de pas de temps de durée constante ou variable. Dans tous les cas, la synchronisation des cycles de vie des agents est une question que doit se poser le modélisateur car c'est un choix susceptible de biaiser les simulations.

La section suivante présente plusieurs plateformes SMA, certaines très génériques et d'autres plus orientées vers les applications en biologie.

2.3.6 Les plateformes de SMA

Le développement d'un SMA est relativement lourd d'autant qu'il s'accompagne le plus souvent d'une interface graphique. Il existe donc un intérêt indéniable à utiliser un outil de développement dédié à la création d'un SMA.

On peut trouver des plateformes pouvant modéliser par exemple :

- les agents mobiles (Voyager, Odissey, Aglet, etc.),
- les agents cognitifs (AgentBuilder, etc.) avec la norme FIPA⁴ (Jade, FIPA-OS, etc.), ou la norme KQML (JAT, JAT-Lite, etc.),
- les agents collaboratifs (Zeus, JAFMAS, KAoS, JAFIMA, etc.),
- les simulations de SMA (NetLogo, Cormas, Swarm, etc.).

On notera que bien que chacune de ces plateformes propose un large éventail de modèles agents, aucune ne peut être considérée comme la plateforme générique "universelle" et les modélisateurs doivent faire un choix pertinent.

On peut distinguer les plateformes suivant le niveau de programmation requis pour créer une simulation.

⁴Foundation for Intelligent Physical Agents - www.fipa.org

Les **plateformes à programmation simplifiée** (*computer-aided software engineering* ou CASE) désignent les environnements de développement graphiques qui facilitent la création rapide de logiciels. Les applications générées proposent en général un compromis entre ergonomie, fonctionnalités et vitesse d'exécution. Elles ont le plus souvent une interface ergonomique permettant une construction simplifiée (voire graphique) des agents avec une génération de code et une exécution transparentes. Proposant plus ou moins de choix concernant le modèle des agents (cycle de vie, interactions, etc.) et les paramètres des simulations, elles fournissent généralement différents moyens de visualisation : un rendu direct de la simulation, une production de vidéo ou alors l'utilisation d'observables pour générer des statistiques globales ou locales.

Ces environnements intégrés de modélisation sont généralement plus qu'un simple langage de programmation. A priori l'utilisateur n'a nul besoin d'être expert dans le langage de programmation sous-jacent. Cependant, il se peut parfois que le modélisateur ait à appréhender une interface complexe qui peut se révéler tout aussi compliquée qu'un langage de programmation à part entière, et ce, même si certaines plateformes proposent des langages de programmation "visuels" (StarLogo TNG). L'annexe B.2.1 fournit un tableau récapitulatif des plateformes à programmation simplifiée.

Les **plateformes à programmation avancée** proposent des bibliothèques ou paquets capables de gérer la physique d'un monde virtuel ou la communication entre les agents. Ces bibliothèques ont l'avantage de fournir des fonctions génériques sans avoir à tout coder soi-même, mais en contrepartie il est nécessaire d'avoir les connaissances suffisantes pour au moins les exploiter correctement voire les augmenter si besoin est et si la licence le permet (ce qui est généralement le cas). L'annexe B.3.1 fournit un tableau récapitulatif des différentes plateformes ou bibliothèques nécessitant une programmation avancée.

Nous allons présenter cinq plateformes SMA plus ou moins spécifiques des applications en biologie.

► Les plateformes génériques de simulation

Les plateformes génériques de simulation sont capables de concevoir une grande diversité de SMA et de les utiliser pour simuler des phénomènes très différents. D'une manière générale, ces plateformes peuvent produire des SMA modélisant des phénomènes très différents, par exemple physiques, biologiques, démographiques, sociologiques, etc. Cette capacité à prendre en compte des systèmes dynamiques dissemblables suppose une certaine généralité dans les primitives comportementales fournies par la plateforme.

➤ StarLogo

StarLogo [SKR01] est un environnement de programmation et de modélisation pour étudier le fonctionnement des systèmes complexes décentralisés. Il est possible de modéliser des phénomènes émergents naturels comme le vol groupé des oiseaux, les colonies de fourmis, etc.

Mitchel Resnick, du Massachusetts Institute of Technology (MIT), créa cet outil à destination des enfants. En effet, StarLogo est une variante du langage de programmation Logo créé pour initier les plus jeunes à l'informatique. Le Logo est un langage de programmation réflexif et fonctionnel. Il a été adapté du Lisp dans les années 60 au MIT par Wally Feurzeig et Seymour Papert, tout en étant plus lisible, ce qui lui a valu le surnom de "*Lisp sans parenthèses*". En proposant une telle plateforme, Resnick a aussi inventé la notion d'outil philosophique. Jusqu'ici, la pensée s'exprimait essentiellement par l'écriture ou par la parole. Elle va maintenant pouvoir s'incarner dans un programme car un code StarLogo n'a pas pour but de produire une "*application*" qui doit être utilisée, mais d'aider à penser, à comprendre, voire à convaincre⁵.

Avec la version initiale de Logo, il est possible de générer des dessins et des animations en attribuant des comportements minimaux à des agents génériques (ou "*tortues*"). StarLogo étend ce principe en agentifiant l'environnement qui est constitué de parcelles 2D (ou

⁵Resnick a également inventé une nouvelle sorte d'écriture : "*l'essai actif*". Il s'agit d'un mélange de texte écrit et de programmes informatiques (par exemple une page web dans laquelle sont inscrites des applets Java) permettant au lecteur de tester et comprendre les idées exprimées en jouant de manière interactive avec les programmes.

pavés ou encore “*patches*”) pouvant interagir deux à deux ou avec les agents mobiles (“*turtles*”). Par exemple, ces tortues peuvent percevoir leur environnement local en se référant aux informations données par les *patches* environnants.

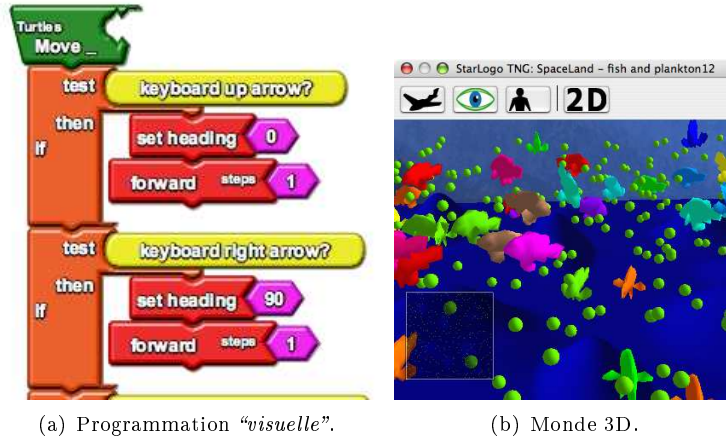


FIG. 2.3.3 – StarLogo TNG.

StarLogo TNG est la prochaine génération de StarLogo développé par le MIT Teacher Education Program. Il apporte plusieurs avancées pour rendre encore plus accessible au grand public la conception d’un SMA avec notamment un langage de programmation visuel inédit (Fig.2.3.3(a)), le code devenant alors un empilement de blocs colorés, et un rendu 3D du monde simulé (Fig.2.3.3(b)).

Si effectivement cette plateforme confère un environnement idéal pour une introduction pragmatique (voire ludique) à la conception et la simulation d’un SMA, il y a néanmoins certaines limitations comme par exemple l’absence de mécanisme simple pour gérer précisément le temps.

➤ NetLogo

NetLogo [GT005] est un langage de programmation de multi-agents et un environnement de modélisation. Créé vers la fin des années 90 par Uri Wilensky, NetLogo a été conçu dans l’esprit du langage Logo pour être accessible aux novices tout en ne limitant pas les programmeurs expérimentés. NetLogo, tout comme StarLogo, possède lui aussi son ensemble spécifique d’instructions augmentant les primitives Logo d’origine.

NetLogo distingue trois type d’agents *turtles*, *patches*, *observer*. Les agents *turtles* se déplacent librement dans un espace 3D illimité (tore) et divisé en une grille de *patches*. Chaque *patch* est un carré sur lequel un agent peut se déplacer suivant une certaine direction. Les variables peuvent être globales, locales à une procédure, propres à un agent. Les comportements sont définis par des procédures ou des fonctions constituées d’une suite de commandes qui sont soit des mots du lexique, soit des appels d’autres procédures (ou fonctions).

La commande `ask` est probablement la commande la plus importante, elle permet de demander à un ensemble d’agents d’exécuter, dans un ordre quelconque une séquence d’instructions. Il est possible de discriminer dans les deux ensembles *turtles* et *patches* des sous-ensembles particuliers, par exemple à l’aide des commandes `turtles-here` ou `turtles-at` (des

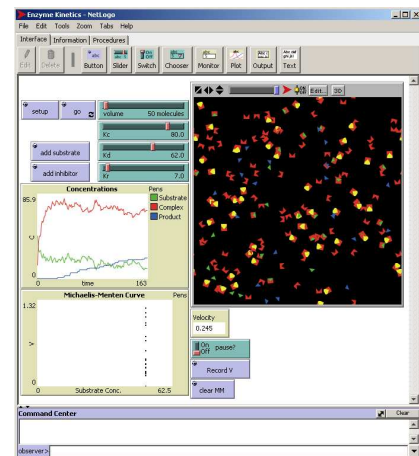


FIG. 2.3.4 – La plateforme NetLogo utilisée pour modéliser une réaction enzymatique.

agents sur une certaine localité) ou de l'opérateur `with` (des agents qui ont une certaine propriété). On peut spécialiser les tortues en espèces (ce qui revient à créer des groupes de tortues) permettant ainsi d'attribuer des comportements différents. Comme il ne s'agit que d'une discrimination syntaxique (le système implémente en fait les tortues avec une variable `breed`) on peut à tout moment faire passer une tortue d'une espèce à une autre.

Cette plateforme est accompagnée d'une bibliothèque conséquente de modèles notamment pour la biologie. Parmi les problématiques abordées, nous pouvons citer la cinétique enzymatique avec un seul substrat (Fig.2.3.4), la réponse immunitaire, la synthèse de protéines, la croissance et la résistance de tumeurs au traitement chimique, la régulation génétique (operon lactose, promoteurs logiques,...) et bien d'autres encore consultables (et exécutables sous forme d'applet Java pour la plupart⁶).

NetLogo bénéficie du support d'une large communauté produisant de nombreux modèles dans des domaines très différents. Sans avoir la programmation visuelle de StarLogo, NetLogo propose de nombreuses primitives pour concevoir un SMA. Néanmoins, tout comme StarLogo, cette plateforme est limitée par le langage Logo et l'impossibilité de préciser des structures complexes avec ce langage.

➤ Breve

Breve⁷ est un environnement 3D de simulation conçu pour modéliser les systèmes décentralisés et la vie artificielle. L'objectif de ce système est de permettre d'implémenter rapidement des simulations décentralisées tout en fournissant une plateforme efficace pour la conception élaborée de simulation de vie artificielle ou de SMA. Pour ce faire, Breve prend en charge certains services comme la communication entre les agents, la représentation dans un espace 3D, le rendu graphique, etc.

Avec Breve (Fig.2.3.5), il est possible d'utiliser les langages python [R0606] ou `steve`⁸ pour écrire des scripts définissant le comportement des agents. Par ailleurs, la plateforme fournit un moteur de rendu OpenGL, un moteur physique avec détection des collisions. Breve peut être utilisé pour une grande variété de domaines d'application comme par exemple simuler des créatures virtuelles, des écosystèmes artificiels, des systèmes de biologie moléculaire, etc. Parmi les exemples d'applications disponibles pour la biologie, nous pouvons citer la modélisation du switch du phage λ ⁹.

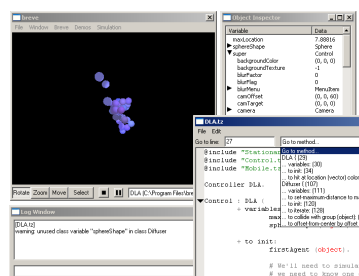


FIG. 2.3.5 – L'environnement Breve.

Cette plateforme offre la possibilité de concevoir un SMA en bénéficiant des avantages de la programmation orientée objet. Le modélisateur accélère la conception puisqu'il bénéficie de nombreux services orientés SMA et le niveau d'interaction avec les objets visualisés permet une lecture aisée de la scène. Néanmoins, il reste certaines limitations comme par exemple le moteur physique qui ne permet pas de définir une gestion spécifique du temps.

➤ Les plateformes orientées biologie

Les simulateurs spécifiques permettent de simuler des SMA modélisant un phénomène biologique précis.

➤ BioDyn

⁶<http://ccl.northwestern.edu/netlogo/models/community/>

⁷<http://www.spiderland.org/breve.pdf>

⁸<http://www.spiderland.org/node/2632>

⁹<http://www.spiderland.org/breve/users.php>

BioDyn¹⁰ est une plateforme dédiée à la conception et à la simulation d'agents réactifs situés notamment pour la biologie. Son objectif est de fournir un moyen rapide de tester des modèles d'organisation des systèmes vivants. Pour ce faire, BioDyn propose des fonctionnalités intéressantes : des formes géométriques libres (avec détection des collisions), plusieurs type d'interactions simultanément disponibles, attribuables et paramétrables, des comportements minimaux des agents aisément programmables et une gestion physiquement réaliste du temps et de l'espace. Cette plateforme se décline en plusieurs versions parmi lesquelles une version 2D, 3D et une version simplifiée au niveau de l'interface dédiée à l'enseignement.

Parmi les exemples d'applications biologiques, nous pouvons citer les réactions enzymatiques (Fig.2.3.6(a)), l'immunologie, la glycolyse, les déformations membranaires (Fig.2.3.6(b)), la propagation de métastases (Fig.2.3.6(c)), etc.

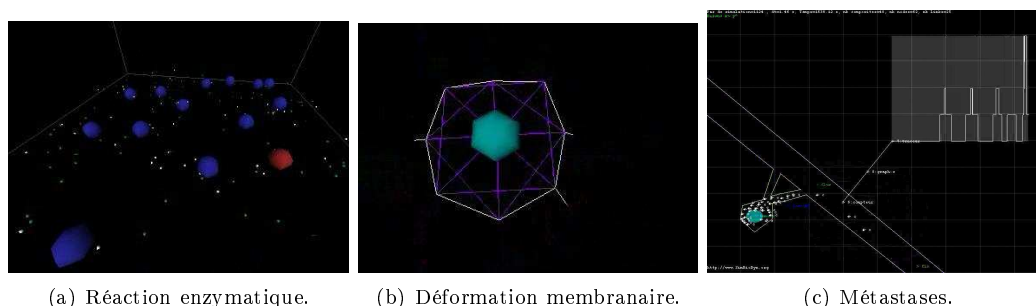


FIG. 2.3.6 – La plateforme BioDyn et certaines de ses applications.

Cette plateforme de simulation, restreinte aux agents situés réactifs, est très bien adaptée pour simuler des systèmes biologiques. Néanmoins, il n'est pas évident de spécifier une gestion particulière du temps pour le moteur physique ni d'avoir la main sur la discrétisation de l'espace.

> HSIM

HSIM¹¹ est un simulateur qui permet la modélisation et la simulation de la formation, du mouvement et de la dissociation d'agglomérats de molécules. Les agents sont ici définis comme étant des molécules évoluant librement dans un espace 3D limité par une membrane cellulaire (monde borné). Ce simulateur n'est pas restreint à un modèle particulier d'interactions entre macromolécules mais permet de modéliser toute sorte d'interactions à l'aide d'un langage de modélisation qui lui est propre. Le simulateur est configuré à l'aide d'un fichier décrivant les divers types de molécules, leur représentation, leurs interactions et l'état initial du système.

Dans ce qui suit, nous présentons un exemple de définition des agents et des règles régissant leurs interactions. Le mot clef `molécule` permet de définir le nom, la couleur voire la durée de vie de chaque type de molécule. Le mot clef `membrane` définit une molécule contrainte à rester sur la surface du volume de simulation.

```

molécule e2 (255, 255, 0), // enzyme 2 en jaune
        s1 (0, 250, 0), // un substrat en vert
        s2 (200, 200, 200), // celui de E2 en gris
        s3 (0, 250, 250) [1000]; // un autre substrat en cyan et un temps de vie de 1000 pas de simulation
membrane e1 (255, 0, 0); //protéine membranaire (diffusion en 2D) en rouge

```

Les règles d'interaction entre molécules sont de 4 types : réaction, association, dissociation et catalyse. Les règles suivantes modélisent la transformation du substrat S1 en S2 catalysé par l'enzyme E1.

```

s1 + e1      -> s1(1) * e1(1) [0.4]; // E1 reconnaît et fixe son substrat S1
{s1}e1 + e2  -> e1 (1) * e2 (1) [0.8]; // en présence de S1, E1 prend de l'affinité pour E2
{~s1}e1 * e2 -> e1 + e2 [0.0003]; // sans son substrat E1 perd son affinité pour E2
{e1}s1 + {e1}e2 -> s2(1) * e2(1) [1.0]; // E1 transforme S1 en S2 et le fixe a E2
{e2}e1 * {e2}s2 -> e1 + s2 [1.0]; // S2 se libère de E1 (mais reste fixe par E2)

```

¹⁰<http://pagesperso.univ-brest.fr/~ballet/>

¹¹<http://www.lri.fr/~pa/BioInformatics/>

Parmi les exemples disponibles, nous pouvons citer des réactions enzymatiques, les assemblages de protéines, etc.

La figure 2.3.7 montre une simulation de HSIM pour modéliser la polymérisation de filaments d'actine. Les règles d'interaction ne concernent que la formation des dimères, la formation des polymères par une des extrémités d'un filament et la rephosphorylation, autrement dit la dépolymérisation des filaments. À aucun moment il n'est question de privilégier une direction dans le volume de simulation. Or on remarque un phénomène émergent intéressant : les différents cycles de polymérisation et de dépolymérisation des filaments dans un volume oblong privilégient visiblement les directions dans le sens de la longueur de la cellule.

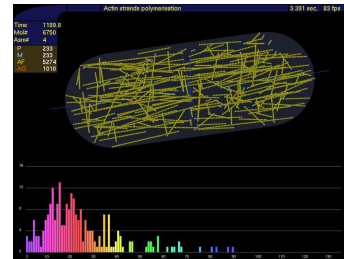


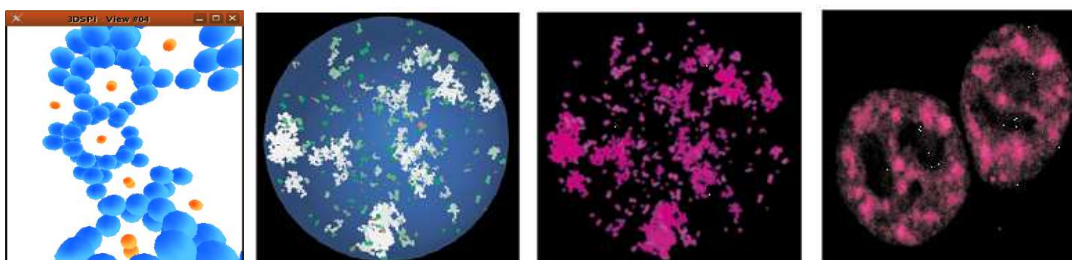
FIG. 2.3.7 – HSIM et son application pour la polymérisation de filaments d'actine.

Ce simulateur, grâce à son environnement de simulation (visualisation, détection des associations des molécules) et à son langage de script, est adapté pour la modélisation de réactions enzymatiques. Néanmoins, le volume de simulation est un monde borné, ce qui est une contrainte forte de modélisation. Par ailleurs, il n'est pas évident de paramétrer finement la description des agents, la gestion du temps et le moteur physique du système.

> 3DSpi

3DSpi¹² est un simulateur multi-agents dédié à l'étude de la dynamique de structures multi-protéiques. Sur la base d'un nombre restreint de paramètres pour modéliser les protéines et leurs interactions, ce simulateur a pour vocation d'aider à comprendre quelles sont les conditions minimales aboutissant à un comportement particulier des structures multi-protéiques. 3DSpi peut ainsi servir à tester l'hypothèse comme quoi de telles structures peuvent émerger et rester stables grâce uniquement aux propriétés des protéines et à leurs interactions.

Les auteurs observent la formation de structures multi-protéiques très dynamiques (Fig.2.3.8(a)) avec une transition de phase en ce qui concerne la forme de ces structures en fonction de l'affinité entre les protéines [SRP⁺05]. De plus, dans des conditions particulières de calibrage, les structures générées (Fig.2.3.8(b) et Fig.2.3.8(c)) exhibent une organisation spatiale assimilable aux observations (Fig.2.3.8(d)) [CSM⁺07]. D'autres simulations, telles la formation ou la dissociation de corps nucléaires, sont en cours d'étude avec ce modèle.



(a) Détail des structures protéiques émergentes. (b) Vue d'ensemble du volume de simulation. (c) La même image après traitement des couleurs. (d) Observations expérimentales.

FIG. 2.3.8 – Le simulateur 3DSpi et son application pour l'étude des corps nucléaires.

Ce simulateur se décline en plusieurs versions, une des plus récentes proposant des agents qui interagissent uniquement avec des potentiels par paire type Lennard-Jones (ces potentiels issus de la dynamique moléculaire seront expliqués dans le chapitre suivant). Par ailleurs, le temps est

¹²<http://bsmc.insa-lyon.fr/moceme/>

discrétisé avec des pas de temps variables : un processus évalue, en fonction de l'erreur d'intégration autorisée, le laps de temps approprié pour faire évoluer le système.

Ce simulateur offre un paramétrage du moteur physique très proche de ce qui nous paraît nécessaire pour notre application. Toutefois, il n'est pas possible de définir avec 3DSpi une dynamique des corps rigides pour un ensemble d'agents composant la même structure multi-protéiques. Enfin, ce simulateur n'offre pas de manière simple de spécifier les types d'agents et leurs relations de composition.

► Limitations des plateformes multi-agents

De l'étude de ces différentes plateformes, nous pouvons retenir un certain nombre de points. Certaines plateformes génériques fournissent des avantages non négligeables pour aider le biologiste ou le modélisateur à concevoir un SMA et à évaluer le potentiel de certaines hypothèses concernant des phénomènes biologiques. Dans tous les cas, nous avons vu qu'il y a certaines limitations inhérentes soit au langage de programmation sous-jacent (par exemple Logo pour StarLogo et NetLogo), soit aux spécificités de la plateforme, qui, en prenant en charge un aspect de la modélisation par agent (communication, moteur physique, rendu, etc.) ne permet pas d'adapter le service offert.

Des simulateurs spécifiques à certaines applications en biologie réduisent le nombre de paramètres à définir par l'utilisateur ce qui aide à mieux comprendre des phénomènes émergents complexes tels que l'orientation préférentielle des filaments d'actine dans une cellule oblong ou l'agrégation de corps nucléaires pour former des structures multi-protéiques singulières. Ces simulateurs se montrent très performants dans leur domaine d'application, mais il est difficile d'augmenter leur modèle agent et les choix de modélisation tels que la dynamique du système sont très spécifiques des applications pour lesquelles ils ont été développés. Ils ne pourront pas être considérés comme des outils génériques de modélisation pour la biologie.

2.4 En résumé

Dans ce chapitre, nous avons étudié différentes approches de modélisation, avec comme fil conducteur un point de vue du système qui va du global au local. Pour notre étude sur la membrane interne mitochondriale, un des points majeurs que nous désirons étudier est le repliement de cette membrane, et ceci en liaison avec les interactions phospholipidiques. Dans ce contexte, il nous apparaît que des modèles basés sur des équations différentielles ne nous permettront pas de caractériser précisément le comportement des phospholipides alors que les systèmes multi-agents semblent offrir un cadre adéquat à nos simulations, et ce pour les raisons suivantes :

- prise en compte de la singularité des différentes molécules : volume, conformation, composition ;
- gestion de l'espace : influence de l'environnement local ;
- gestion du temps : possibilité de spécifier la discrétisation.

Nous avons choisi de modéliser un système qui sera "mono-échelle" car toutes nos interactions se produisent au niveau moléculaire et nous espérons pouvoir observer l'apparition des replis de la membrane en tant que phénomène émergent du SMA.

Différentes plateformes plus ou moins génériques sont disponibles et simplifient la tâche du modélisateur. Néanmoins, les services rendus sont généralement au détriment de l'expressivité du modèle agent ou des spécificités de la dynamique du système. En d'autres termes, le modèle agent que nous avons conçu et qui sera développé au chapitre 4 nécessitera l'implémentation de notre propre simulateur.

Avant d'aborder notre modèle multi-agents dans son ensemble, nous allons nous intéresser aux différents paramètres et comportements caractérisant les phospholipides et les membranes biologiques. Ceci fait l'objet du prochain chapitre.

Chapitre 3

Une membrane à l'échelle mésoscopique

Ce chapitre présente différentes approches de modélisation de la membrane : les modèles continus à l'échelle macroscopique de la bicouche, la mécanique moléculaire à l'échelle microscopique des atomes et les modèles à grains d'atomes ou *gros grains* à l'échelle mésoscopique des lipides. Bien que nous n'ayons pas utilisé de modèle continu, nous avons choisi des les présenter afin que le lecteur puisse comparer cette approche couramment rencontrée avec celle que nous avons choisie. A contrario, la description des méthodes liées à la mécanique moléculaire ou aux modèles *gros grains* sera directement utile à la compréhension de notre modèle. Nous ferons donc une revue détaillée des différents modèles *gros grains* afin d'apprécier leurs aptitudes à modéliser les membranes biologiques.

3.1 La modélisation par modèles continus élastiques

Les modèles continus sont très souvent déployés pour l'étude de phénomènes à l'échelle macroscopique temporelle et spatiale, comme par exemple la déformation d'une vésicule ou les modes de fluctuation thermique des bicouches lipidiques [BLB06, NPW04].

Les distances modélisées étant grandes devant la taille des molécules dans un modèle continu, la membrane est représentée comme un film fin et élastique décrit par l'équation de son énergie libre de déformation ou énergie élastique H . La figure 3.1.1 donne une vue schématique de membrane (a) et des modes de déformation qui participent au calcul de l'énergie élastique : un terme lié à la courbure (b), un terme d'extension-compression de la membrane (c) et, pour les membranes polymérisées ou en phase gel uniquement, un terme de cisaillement (d).

Il existe plusieurs types de modèles continus parmi lesquels on peut distinguer :

- les modèles de type Ginzburg-Landau qui décrivent le système par les densités en éléments hydrophobes et hydrophiles ;
- les modèles basés sur l'hamiltonien de Canham-Helfrich qui conçoit le système comme un assemblage d'interfaces infiniment fines caractérisées par leurs constantes d'élasticité.

Dans les sections suivantes, nous évoquerons les modèles de type Ginzburg-Landau avant de détailler le calcul de l'énergie libre dans le cas d'un modèle basé sur l'hamiltonien de Canham-Helfrich.

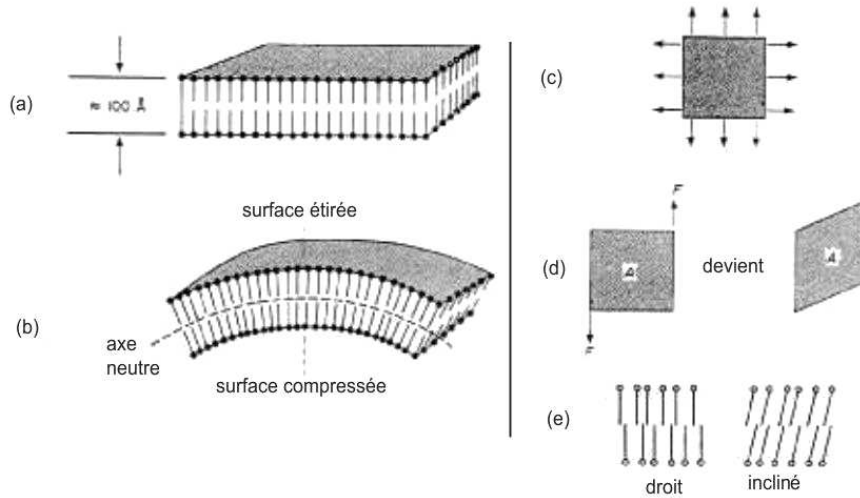


FIG. 3.1.1 – Différents modes de déformation d'une membrane (d'après [Can70] et [Hel73]).

3.1.1 Les modèles de type Ginzburg-Landau

Dans les modèles de type Ginzburg-Landau [GS], l'énergie libre H est décrite par un paramètre scalaire (ou un vecteur) localisé spatialement. La minimisation globale de cette énergie libre est traduite sous la forme d'une équation différentielle partielle, qui décrit le système à l'équilibre.

Gompper et Klein [GK92] ont défini un modèle de type Ginzburg-Landau pour modéliser un mélange d'eau et de molécules amphiphiles. La concentration de ces molécules amphiphiles est décrite par un scalaire tandis que leur orientation est donnée par un vecteur. Les diagrammes de phase sont calculés en minimisant la fonction de l'énergie libre. Il existe différentes phases ordonnées correspondantes à des structures lamellaires, hexagonales ou cubiques. Les diagrammes de phase obtenus montrent que le changement entre ces différentes phases ordonnées va de paire avec une augmentation de la concentration des molécules amphiphiles. A partir de ces résultats il est possible d'identifier des paramètres qui indiquent que le système a tendance à s'organiser sous forme de bicouche ou de micelle.

Cette méthode est donc capable de reproduire des diagrammes de phase qualitativement corrects pour des systèmes aussi étendus qu'une membrane. Néanmoins, la paramétrisation de la microdynamique sous-jacente rend difficile la compréhension des résultats quantitatifs et le lien entre le modèle et les systèmes biologiques reste souvent qualitatif.

3.1.2 Les modèles basés sur l'hamiltonien de Canham-Helfrich

Historiquement les premiers modèles continus ont été proposés indépendamment par Canham [Can70] et Helfrich [Hel73].

La première déformation est l'**extension-compression**, dont l'énergie H_{ext} peut être mise sous la forme Eq.(3.1.1) et dépend quadratiquement de l'augmentation ou de la diminution relative de surface.

$$H_{ext} = \frac{1}{2} \chi_{ce} \left(\frac{\Delta A}{A} \right)^2 \quad (3.1.1)$$

Dans le cas où la membrane n'est pas en phase fluide mais **en phase gel ou polymérisée**, alors un terme de cisaillement doit être pris en compte. Si μ_{cis} est le module de cisaillement, λ le taux d'extension latérale ($\lambda = (L_0 + \Delta L)/L_0$) alors l'**énergie due au cisaillement** H_{cis} prend la forme Eq.(3.1.2).

$$H_{cis} = \frac{1}{2} \mu_{cis} (\lambda^2 + \lambda^{-2} - 2)^2 \quad (3.1.2)$$

Un terme d'inclinaison des molécules est aussi rajouté par Helfrich (mais se révèle négligeable par la suite). Si \vec{n} désigne le vecteur unitaire normal à la membrane, et \vec{d} le vecteur unitaire correspondant à la direction moyenne des molécules, si on note k_t le module de rigidité élastique d'inclinaison, alors on peut écrire le terme d'**énergie d'inclinaison** H_{incl} sous la forme Eq.(3.1.3).

$$H_{incl} = \frac{1}{2}k_t(\vec{n} \wedge \vec{d})^2 \quad (3.1.3)$$

Enfin, il faut encore compléter cette énergie par le terme dû à la courbure de la membrane. On montre que la manière la plus générale d'écrire l'**élasticité de courbure** H_{courbe} , à partir d'une forme quadratique, est équivalente à la forme Eq.(3.1.4) où k est le module de rigidité de courbure, k_G est le module de rigidité de courbure gaussienne, et c_1 et c_2 les deux courbures principales de la membrane, c_0 la courbure spontanée.

$$H_{courbe} = \frac{1}{2}k(c_1 + c_2 - c_0)^2 + \frac{1}{2}k_G c_1 c_2 \quad (3.1.4)$$

Partant du principe que l'on peut négliger le terme de courbure gaussienne par invariance topologique et qu'avec la définition classique de la tension de surface, $\sigma = \partial F / \partial A$, on obtient alors un terme d'**énergie de tension** H_{tens} sous la forme Eq.(3.1.5).

$$H_{tens} = \sigma \frac{\Delta A}{A} \quad (3.1.5)$$

Finalement, dans le cas où la membrane n'est pas étirée et donc où l'énergie d'étirement ne jouera pas de rôle, on peut écrire l'**énergie de cette membrane** en négligeant les termes d'inclinaison et de cisaillement sous la forme Eq.(3.1.6) connu sous le nom d'**hamiltonien de Canham-Helfrich**.

$$H = \int \int_A \left[\frac{1}{2}k(c_1 + c_2 - c_0)^2 + \sigma \frac{\Delta A}{A} \right] dS \quad (3.1.6)$$

Pour modéliser les **fluctuations thermiques d'une membrane**, nous pouvons réécrire l'hamiltonien de Canham-Helfrich sous l'autre forme Eq.(3.1.7), en notant $u(\vec{r}_\perp)$ l'écart au point r_\perp au plan moyen (Fig.3.1.2) et avec σ la tension et k le module de rigidité de courbure.

$$H = \int \int_A \left[\frac{1}{2}k(\nabla^2 u)^2 + \frac{1}{2}\sigma(\nabla u)^2 \right] dS \quad (3.1.7)$$

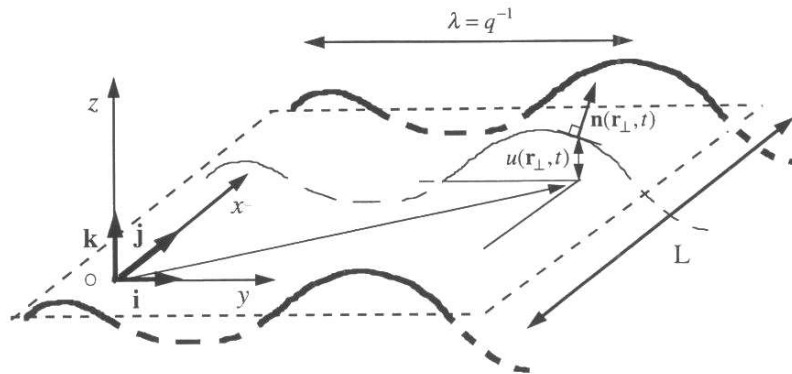


FIG. 3.1.2 – Représentation schématique d'une membrane initialement plane, sans courbure spontanée, de longueur L , d'aire projetée $S_0 = L^2$ dans le plan (x, y) et soumise à des fluctuations. Les déplacements transversaux de la membrane sont des fonctions $z = u(x, y)$ (d'après [Man99]).

Néanmoins, **en présence de protéines**, l'hamiltonien de Canham-Helfrich Eq.(3.1.7) se complexifie sensiblement en intégrant deux autres termes :

- le premier terme, H_{cd} , indique le **couplage entre la courbure de la membrane et la diffusion de la protéine** ; en effet il est logique de penser que suivant sa forme une protéine diffusera pour se localiser dans les zones où la courbure est la plus adaptée à sa forme Eq.(3.1.8) avec Λ le coefficient de couplage et ψ la concentration *algébrique*¹ de protéine ;

$$H_{cd} = \int \int_A 2\Lambda\psi\nabla_{\perp}^2 u dS \quad (3.1.8)$$

- le second terme, H_d , aura tendance à contrebalancer le premier, ce qui est dû tout simplement à la **diffusion des protéines des zones les plus concentrées vers les zones les moins concentrées**. Ce terme est exprimé par Eq.(3.1.9) avec χ le module de compressibilité pour la diffusion des protéines dans la membrane et ψ_0 la concentration "*algébrique*" de protéine optimale.

$$H_d = \int \int_A \frac{\chi}{2}(\psi - \psi_0)^2 dS \quad (3.1.9)$$

On peut alors finalement écrire l'hamiltonien complet du système membrane + protéines [MBRP01] sous la forme Eq.(3.1.10).

$$H = \int \int_A \left[\frac{1}{2}k(\nabla^2 u)^2 + \frac{1}{2}\sigma(\nabla u)^2 + 2\Lambda\psi\nabla_{\perp}^2 u + \frac{\chi}{2}(\psi - \psi_0)^2 \right] dS \quad (3.1.10)$$

En utilisant ce modèle, F.Brown [Bro03] a étudié la régulation de la mobilité des protéines par les fluctuations thermiques d'une membrane de globule rouge. Il suggère sur la base d'arguments théoriques et de simulations, que les ondulations thermiques d'une bicouche lipidique peuvent soutenir un mécanisme qui permet aux protéines de passer d'une région de la surface membranaire à une autre, les deux régions adjacentes étant séparées par une structure interne, par exemple le cytosquelette.

3.2 Modèles continus vs modèles à particules

Les modèles continus sont capables de modéliser des bicouches lipidiques avec des protéines intramembranaires Eq.(3.1.10). Les propriétés telles que la rigidité de liaison, la courbure spontanée et la surface de tension sont utilisées pour décrire l'état physique du système, et **les paramètres liés à ces propriétés** constituent des "*input*" ou **paramètres d'initialisation** pour le modèle. Néanmoins, ces paramètres sont des **variables moyennées** sur des ensembles de molécules et cette approche **ne permet pas de manipuler des éléments au niveau microscopique**. Or nous souhaitons une modélisation de la membrane où les tensions de surface responsables des courbures locales et des replis de la bicouche résultent des interactions entre les constituants de la membrane. C'est pourquoi nous nous sommes tournés vers les paradigmes "*centrés sur les particules*".

La figure Fig.3.2.1 présentent trois approches de ce type qui se différencient essentiellement par la granularité de représentation du système réel :

- en bas à gauche, une simulation atomistique d'une bicouche constituée de phospholipides appelés 1-stearoyl-2-docosahexaenoyl-sn-glycero-3-phosphocholine (SDPC, 18 :0/22 :6 PC) ;
- au milieu, un modèle à *gros grains* d'une bicouche lipidique sous aucune tension ;
- en haut à droite, une surface de membrane maillée par triangularisation aléatoire.

¹une concentration algébrique est une concentration où on compte positivement les protéines dans un sens et négativement celles dans l'autre ; si ρ^+ est la concentration de protéines dans un sens et ρ^- la concentration dans l'autre, alors $\psi = \rho^+ - \rho^-$

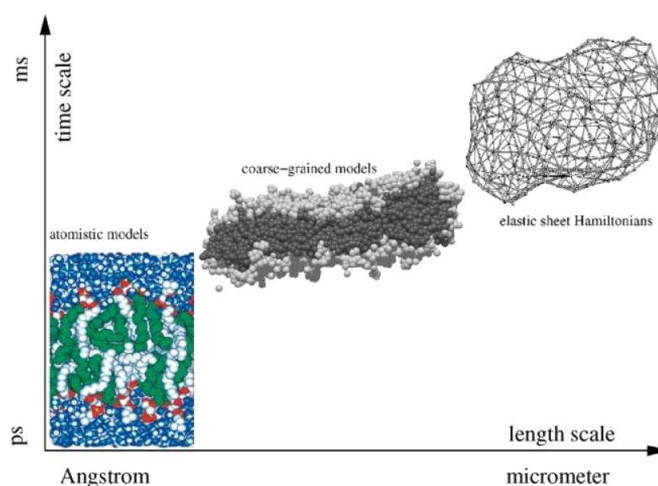


FIG. 3.2.1 – Différentes approches “basées sur les particules” pour étudier des membranes (d’après Müller et al. [MKS06]).

L’utilisation de ces modèles conduit à traduire le système biologique étudié *in vivo* en un système *in silico* de particules en interaction. Les molécules peuvent avoir leur propre conformation et les interactions intermoléculaires peuvent être prises en compte très précisément. L’objectif est d’étudier comment les protéines transmembranaires perturbent localement l’environnement lipidique et réciproquement, comment les replis de la bicouche peuvent induire des confinements et influencer les dynamiques d’association des complexes.

La section suivante présente l’approche par mécanique moléculaire qui manipule des représentations atomiques des molécules.

3.3 La modélisation par mécanique moléculaire

Pour simuler des systèmes moléculaires, la modélisation moléculaire utilise des méthodes de calculs telles que la Mécanique Quantique (QM) (ab initio ou semi-empirique), la Dynamique Moléculaire (MD), la Mécanique Moléculaire (MM) ou, depuis quelques années, de l’association QM/MM qui paraît très porteuse (Voir [Lea96], [LB00], [BA82]).

L’expression *mécanique moléculaire* désigne actuellement une méthode de calcul qui permet, a priori, d’obtenir des résultats de conformation et d’énergie moléculaire en se basant sur la mécanique classique. Apparue en 1930 [And30], ce n’est qu’à partir des années soixantes qu’elle s’est réellement développée, quand les ordinateurs furent plus accessibles et plus performants. Les premières simulations ont été faites par Rahman [Rah64], à partir de ses travaux sur la simulation de l’argon liquide avec un temps de simulation de 10^{-11} s puis avec la simulation de l’eau liquide [RS71].

La MM est basée sur l’approximation de Born-Oppenheimer [BO27] selon laquelle les électrons sont beaucoup plus rapides que les noyaux, ceci permettant de traiter les électrons implicitement. C’est une méthode empirique où les atomes (les noyaux) sont représentés par des masses ou des sphères, et les liaisons par des ressorts de différentes forces.

Les valeurs de ces forces proviennent de données expérimentales de spectroscopie vibrationnelle², de données de diffraction des rayons X (ou cristallographie³) ou de valeurs obtenues par les méthodes *ab-initio*. Le calcul de l’ensemble de ces forces, selon les règles établies pour des oscillateurs harmoniques, permet d’obtenir l’énergie moléculaire et de déterminer la structure tridimensionnelle

²La spectroscopie vibrationnelle ou spectroscopie d’absorption IR ou encore spectroscopie de diffusion Raman, permet la détermination des fréquences de vibration d’une molécule qui dépendent de la masse des atomes impliqués dans le mode normal de vibration ainsi que de la force des liaisons interatomiques. Ceci fournit des informations essentielles sur la structure d’une molécule.

³La cristallographie est l’étude des substances cristallines à l’échelle atomique. L’arrangement spatial des atomes dans la matière est étroitement lié à ses propriétés vis à vis de la lumière.

d'une molécule.

La section suivante présente les champs de force utilisés dans les calculs de dynamique moléculaire. Les définitions sont tirées de [KM02] et [Fel00].

3.3.1 Modélisation par champs de force

Par définition, un champ de force en dynamique moléculaire désigne l'ensemble des équations et des paramètres décrivant l'énergie potentielle d'un système de particules, typiquement, un ensemble fini de molécules. Les différents potentiels expriment l'idée de pénalités énergétiques associées à la déviation par rapport à des valeurs de *référence* ou d'*équilibre*. L'énergie d'une molécule dépend des positions relatives de ses atomes. Elle correspond à la différence d'énergie entre la molécule réelle et une molécule hypothétique dans laquelle toutes les valeurs de la structure comme la longueur des liaisons et les angles de liaisons sont exactement à leurs valeurs idéales (valeurs d'équilibre).

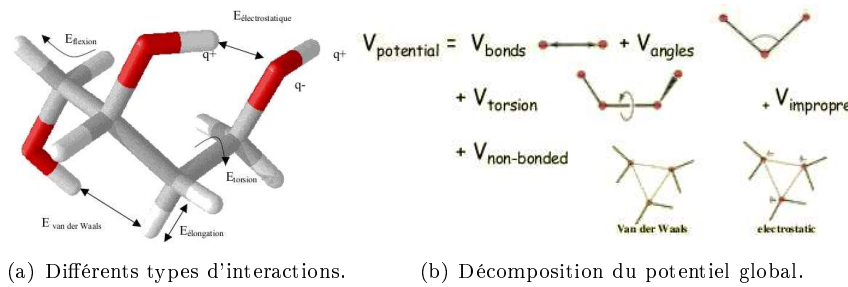


FIG. 3.3.1 – L'énergie d'une molécule dépend des positions relatives de ses atomes.

Cette énergie potentielle ou énergie stérique peut être approximée comme étant la somme de plusieurs contributions (Fig.3.3.1 et Eq.(3.3.1)) que l'on distingue généralement en deux groupes :

- les interactions entre atomes liés par deux ou trois liaisons chimiques : énergie de liaison E_l , de flexion E_θ , de torsion E_τ, \dots
- les interactions entre atomes non liés ou séparés par plus de trois liaisons chimiques : énergie électrostatique E_{el} , de van der Waals E_{vdw}, \dots

$$E_{MM} = E_{\text{atomes liés}} + E_{\text{atomes non liés}} = E_l + E_\theta + E_\tau + E_{vdw} + E_{el} + \dots \quad (3.3.1)$$

Interactions entre atomes liés

➤ **L'énergie de liaison** ou d'élongation (*stretching*) mesure la quantité d'énergie mise en jeu par les variations de longueur des liaisons après extension ou compression à partir de leurs valeurs d'équilibre. Il existe plusieurs potentiels pour modéliser cette énergie, notamment le potentiel de Morse et le potentiel harmonique basé sur la loi de Hooke⁴.

➤ Le **potentiel de Morse** Le potentiel de Morse est considéré comme le plus réaliste, surtout pour ce qui est du phénomène de dissociation, c'est-à-dire de la séparation de deux particules. Cette propriété devient même essentielle si l'on souhaite étudier des ruptures de matériau. La forme analytique est donnée par Eq.(3.3.2) avec n le nombre de liaisons, D_c et r_i respectivement la constante de force et la longueur de la liaison i .

$$E_{lMorse} = \sum_1^n D_c (1 - e^{-a(r_i - r_i^0)})^2 \quad (3.3.2)$$

⁴La loi de Hooke est une loi de comportement des solides soumis à une déformation élastique de faible amplitude.

Toutefois l'utilisation du potentiel de Morse est relativement coûteux en temps de calcul. ➤ Le **potentiel harmonique** est donnée par Eq.(3.3.3) avec $k_{l,i}$ et r_i respectivement la constante de force et la longueur de la liaison i , n le nombre de liaisons.

$$E_l = 1/2 \sum_1^n k_{l,i} (r_i - r_i^0)^2 \quad (3.3.3)$$

Ce potentiel est le plus utilisé car bien que plus simple que le potentiel de Morse, il est considéré comme suffisant et se révèle plus commode à calculer.

➤ **L'énergie de flexion** ou de pliage (*bending*) détermine la quantité d'énergie impliquée par les variations d'angle entre deux liaisons contiguës à partir d'un angle de liaison à l'équilibre. Dans le cas de l'**approximation harmonique**, cette expression, également basée sur la loi de Hooke, donne la forme Eq.(3.3.4) avec $k_{\theta,ij}$ la constante de force de flexion, θ_{ij} et θ_{ij}^0 respectivement l'angle de liaison et l'angle de liaison à l'équilibre et n le nombre de liaisons dans la molécule.

$$E_\theta = 1/2 \sum_{ij} k_{\theta,ij} (\theta_{ij} - \theta_{ij}^0)^2 \quad (3.3.4)$$

➤ **L'énergie de torsion** traduit la modification de l'énergie lors de la rotation d'une molécule autour d'une liaison. L'expression mathématique la plus commune permettant de décrire comment l'énergie d'une molécule évolue en fonction de l'angle de torsion est la série de Fourier Eq.(3.3.5) où τ_i décrit l'angle de torsion, $A_{i,n}$ la constante de force qui contrôle l'amplitude de la courbure, n la périodicité de $A_{i,n}$ et ϕ la phase. L'énergie de torsion est modélisée par une simple fonction périodique présentant des minima multiples (rotamères).

$$E_\tau = 1/2 \sum_i A_{i,n} [1 + \cos(n\tau_i - \phi)] \quad (3.3.5)$$

Ces énergies sont toujours utilisées dans des modèles de dynamique moléculaire. Les interactions entre deux atomes non liés sont responsables de l'encombrement stérique qui joue un rôle important dans la géométrie d'une molécule.

Interactions entre atomes non liés

➤ **L'énergie d'interaction de Van der Waals** entre 2 atomes est la somme des énergies des forces attractives et répulsives. L'expression de l'énergie d'interactions de Van der Waals est exprimée en général sous la forme d'un **potentiel de Lennard-Jones** Eq.(3.3.6) avec r_{ij} la distance séparant deux atomes non liés i et j , r_{ij}^0 la somme des rayons de Van der Waals des 2 atomes i et j , A_{ij} et B_{ij} des constantes de Van der Waals. Il existe une écriture plus connue du potentiel de Lennard-Jones en Eq.(3.3.7) où r est la distance séparant deux atomes non liés, σ le diamètre de collision et ϵ la profondeur du puits.

$$E_{vdwLJ} = \sum_{ij} \left[A_{ij} \left(\frac{r_{ij}^0}{r_{ij}} \right)^{12} - B_{ij} \left(\frac{r_{ij}^0}{r_{ij}} \right)^6 \right] \quad (3.3.6)$$

$$V_{LJ}(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right] \quad (3.3.7)$$

On peut aussi écrire l'énergie d'interactions sous la forme d'un **potentiel de Buckingham** Eq.(3.3.8) avec r_{ij} la distance séparant deux atomes non liés i et j , A_{ij} , B_{ij} et C_{ij} des constantes universelles et ϵ_{ij} , r_i^0 et C_j^0 des paramètres dépendant du type d'atomes considérés.

$$E_{vdwBuck} = \sum_{ij} \epsilon_{ij} \left[A_{ij} \exp \left(-B_{ij} \left(1 - \frac{r_{ij}}{r_i^0 + r_j^0} \right) \right) - C_{ij} \left(\frac{r_i^0 + r_j^0}{r_{ij}} \right)^6 \right] \quad (3.3.8)$$

➤ **L'énergie d'interaction électrostatique** augmente avec la polarité des liaisons chimiques et peut être particulièrement importante, spécialement dans des molécules contenant des

hétéroatomes⁵. C'est une contribution longue portée entre paires d'atomes chargés. Elle est représentée en utilisant un potentiel coulombien Eq.(3.3.9) avec r_{ij} la distance séparant deux atomes non liés i et j , q la charge partielle d'un atome et ϵ_{ij} la constante diélectrique. Les charges partielles sont déduites empiriquement ou proviennent de calculs de mécanique quantique et de potentiels électrostatiques moléculaires.

La forme analytique est parfois modifiée pour tenir compte du solvant de manière implicite, en utilisant une constante diélectrique effective ϵ_{eff} variant en fonction de la distance. La seule justification de cette approximation : les distances courtes sont plus probablement intra-moléculaires (faible ϵ) qu'au travers du solvant (fort ϵ).

$$E_{el} = \sum_{ij} \frac{q_i q_j}{\epsilon_{ij} r_{ij}} \quad (3.3.9)$$

► La troncature des potentiels

Les interactions non liées (électrostatiques et de van der Waals) doivent être calculées pour toutes les paires de particules (atomes ou grains) du système. La complexité est N^2 pour un système à N particules. Toutefois on notera que ces potentiels diminuent rapidement avec la distance r : LJ $\sim 1/r^6$ et Coulomb $\sim 1/r$. Aussi, afin de réduire le temps de calcul, il est d'usage de ne considérer que les particules présentes dans un voisinage paramétré par une **distance de coupure** ou rayon de coupure r_c . Une troncature brutale peut poser des problèmes de discontinuité du potentiel et induire des incohérences, c'est pourquoi certaines équations d'atténuation sont intégrées à la forme initiale.

3.3.2 De l'échelle microscopique à l'échelle mésoscopique

Nous venons de voir quels sont les modèles des principales interactions prises en compte dans le cadre de la dynamique moléculaire. Les particules considérées dans les équations sont ici des atomes. Ce qui pose évidemment le problème du nombre de molécules qu'il est possible de simuler dans un temps raisonnable et avec un ordinateur de configuration standard. C'est pourquoi il existe des modèles à granularité plus haute : les modèles à grains d'atomes. Dans ces modèles, certaines interactions intramoléculaires sont ignorées comme par exemple l'énergie de torsion ou l'énergie des angles dièdres impropres (ou mouvements de courbure hors du plan). Il en va de même pour l'énergie électrostatique qui peut parfois ne pas être prise en compte. Ces choix de modélisation, qui peuvent sembler être éloignés de la réalité physique des systèmes, se révèlent non pénalisant dès lors qu'on étudie des phénomènes globaux, éventuellement à des échelles plus grandes que l'échelle microscopique, où la prise en compte du nombre de molécules est plus important que l'extrême précision de la représentation d'une seule molécule.

Après avoir étudié la modélisation macroscopique par modèles continus élastiques puis la modélisation moléculaire à l'échelle microscopique, nous allons présenter les modèles à grains d'atomes qui se situent à l'échelle intermédiaire mésoscopique et montrer pourquoi cette approche nous paraît en adéquation avec la problématique biologique.

3.4 Les modèles à grains d'atomes : l'échelle mésoscopique

Les techniques de modélisation à l'échelle détaillée de l'atome, par exemple la dynamique moléculaire (*molecular dynamics* ou MD [KM02] [Fel00]), constituent des outils performants pour étudier les structures et les dynamiques biologiques pour des temps de l'ordre de la nanoseconde ($10^{-9}s$) et des tailles de l'ordre du nanomètre ($10^{-9}m$) avec des résolutions de l'ordre respectivement de la femtoséconde ($10^{-15}s$) et de l'ångström ($10^{-10}m$, noté Å). Or à cause des temps de

⁵Un hétéroatome est un atome d'une molécule organique différent du carbone ou de l'hydrogène et non métallique. Les plus fréquents sont l'oxygène, l'azote, le soufre, etc.

calcul, les simulations sont restreintes à quelques centaines de molécules amphiphiles ce qui pour une bicouche lipidique correspond à une aire de l'ordre de la dizaine de nanomètres carrés ($10^{-8}m^2$) alors que la surface d'une mitochondrie est de l'ordre de la dizaine de micromètres carrés ($10^{-5}m^2$).

Les modèles à grains d'atomes (ou *coarse graining*) de résolution plus grossière offrent la possibilité d'explorer des dynamiques cruciales pour de nombreux phénomènes biologiques à des échelles de temps et d'espace plus grandes. Par exemple la figure 3.4.1 précise que ces modèles sont capables de modéliser des structures de la taille des radeaux lipidiques (ou *raft*), c'est-à-dire de l'ordre du dixième de micromètre ($10^{-7}m$) et des dynamiques telles que la diffusion des lipides ou le repliement d'une bicouche de l'ordre du dixième de micro-seconde ($10^{-7}s$). Déjà éprouvés depuis de nombreuses années dans le domaine de la physique des polymères [G7979], des modèles à grains ont été développés plus récemment pour étudier les membranes lipidiques ([BLB06], [Lyu05], [IV05], [AV07], [SIV06]), les protéines ([Toz05], [GT06], [KJD06], [NAC+05], [CV06]), les peptides ([LYZ06], [LZ06], [LV07]), l'ADN ([VBS05], [SZS05]) et même les ribosomes ([TVFB03],[WRBJ04]).

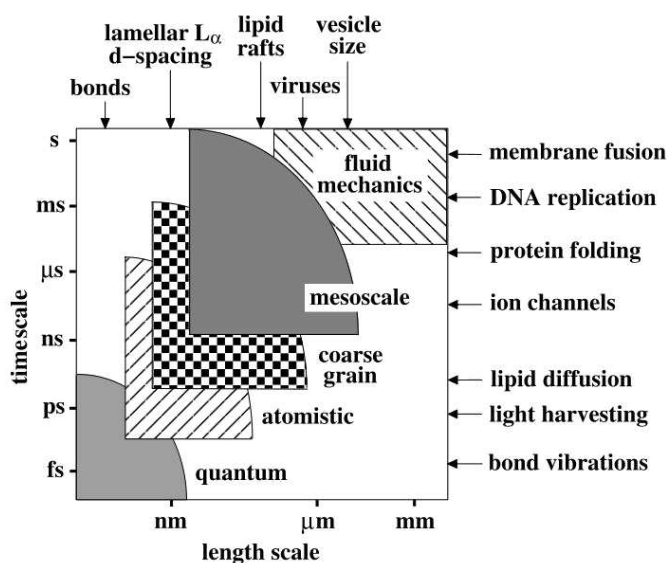


FIG. 3.4.1 – Diagramme schématisant des échelles temporelles et spatiales accessibles par différentes techniques, notamment le coarse-graining. Certaines structures et événements membranaires caractéristiques sont indiqués (d'après [NSK04]).

L'objet de notre étude étant la membrane interne mitochondriale, nous nous attacherons dans un premier temps à présenter une revue des modèles mésoscopiques des lipides, molécules amphiphiles constituant les membranes biologiques.

La nature amphiphile des molécules et leur géométrie sont des caractéristiques importantes des systèmes auto-assemblés tels que les membranes. Ces deux propriétés essentielles doivent a priori être conservées par le modèle à grains. Le choix de la granularité du modèle, c'est-à-dire sa finesse de représentation, passe par l'identification des types et du nombre de grains nécessaires pour modéliser ces propriétés.

3.4.1 Les grains : les différents types, le nombre par lipide

La relation entre le nombre de types et le nombre de grains pour la modélisation d'un lipide n'est pas évidente. De fait, un nombre moindre de types de grain n'est pas nécessairement synonyme d'un moindre nombre de grains par lipide. Par exemple, Groot et Warren [GW97] définissent seulement 2 types mais attribuent près de 13 grains pour un lipide là où Lyubartsev [Lyu05] utilise 4 types de grain pour un lipide constitué seulement de 10 grains.

Pour un même nombre de types, on peut faire varier le nombre de grains par lipide pour par exemple tester différentes **mixtures de tensioactifs**⁶ (*surfactants*) (Goetz et Lipowsky [GL98]). On peut également chercher à modéliser des **mixtures de deux types de phospholipides** différenciés par leurs acides gras (par exemple DLPC :DSPC), autrement dit par la longueur de leurs queues (Stevens [SHW03, Ste05]). Enfin, des lipides différenciés uniquement par le nombre de grains permet d'évaluer l'**impact de la granularité**⁷ sur certaines observables macroscopiques, comme l'épaisseur de la bicouche, et microscopiques, comme l'aire par lipide (Kranenburg et al. [KNS04]).

Le tableau 3.4.1 contient une synthèse d'un certain nombre de modèles consultables dans la littérature (en prenant soin de se restreindre à la modélisation des solutés). Pour les lipides, la modélisation porte essentiellement sur des molécules de DMPC.

	nb types	nb grains (par défaut DMPC)
Morikawa et Saito [MS94]	1	0.5
Ayton et Voth [AV02], Drouffe et al. [DML91]	1	1
Cooke et al. [CD05] Noguchi et al. [NT01]	2	3 (1+2)
Sintes et Baumgärthner [SB97, SB98b]	2	5 (1+4)
Goetz et Lipowsky [GL98]	2	5 (1+4) à 11 (3+2*4)
Lenz et Schmid [LS05]	2	6 (1+5)
Stevens [SHW03, Ste05]	2	11 (3+2*4) à 19 (3+2*8)
Groot et Warren [GW97], Venturoli et al. [VS99], Yamamoto et Hyodo [YMH02]	2	13 (3+2*5)
Kranenburg et al. [KNS04]	2	13 (3+5*2) et 38 (12+2*13)
Farago [Far03]	2bis ^a	3 (1+1+1)
Brannigan et al. [BPB05]	3	5 (1+1+3)
Groot et Rabone [GR01]	3	13 (2+1+2*5)
Marrink et al. [MM04]	3 à 9 ^b	DLPC : 10 (2+2+2*3) et DMPC, DPPC : 12 (2+2+2*4) et DSPC : 14 (2+2+2*5)
Lyubartsev [Lyu05]	4	10 (1+1+2*(1+3))
Klein et al. [NSK04], Izvekov et al. [IV05]	5	13(1+1+1+2*(1+4))
Venturoli et al. [VSS05]	3 ^c	13(3+2*5) avec {4*(3+10+3) ou 7*(3+10+3) ou 43*(3+10+3)}

TAB. 3.4.1 – Différents modèles à grains consultables dans la littérature en fonction du nombre de types de grain et du nombre de grains par lipide.

^aFarago propose 2 types de grain, hydrophile et hydrophobe, mais marque néanmoins une différence entre les deux queues hydrophobes.

^bMarrink et al. [MM04] proposent 3 types dont 2 possèdent 4 sous-types optionnels ce qui donne bien au maximum 9 types distincts (sans compter le type pour le solvant).

^cVenturoli étend son précédent modèle à 2 types [VS99] pour prendre en compte 3 tailles de protéine intramembranaire.

Nous allons maintenant expliquer les grandes lignes des modèles à grains en fonction du nombre de types définis.

► 1 type de grain

Dans leur modèle, Morikawa et Saito [MS94] propose de prendre en compte **simultanément 2 lipides opposés** dans la bicouche. Le volume correspondant peut être soit cylindrique (Fig.3.4.2(a)) soit conique (Fig.3.4.2(b)). Mais la géométrie de ce dimère artificiel est ici un paramètre du modèle, et non une conséquence résultante des interactions entre grains comme dans les modèles décrits dans les paragraphes suivants. Les lipides s'organisent sous formes de micelles et de vésicules qui varient en fonction de la forme choisie pour le volume des lipides.

⁶Un tensioactif ou agent de surface est une molécule amphiphile, c'est-à-dire qu'elles présentent deux parties de polarité différente, l'une lipophile (miscible dans l'huile) et apolaire, l'autre hydrophile (miscible dans l'eau) et polaire.

⁷On associe la granularité au niveau de finesse de la représentation d'une molécule.

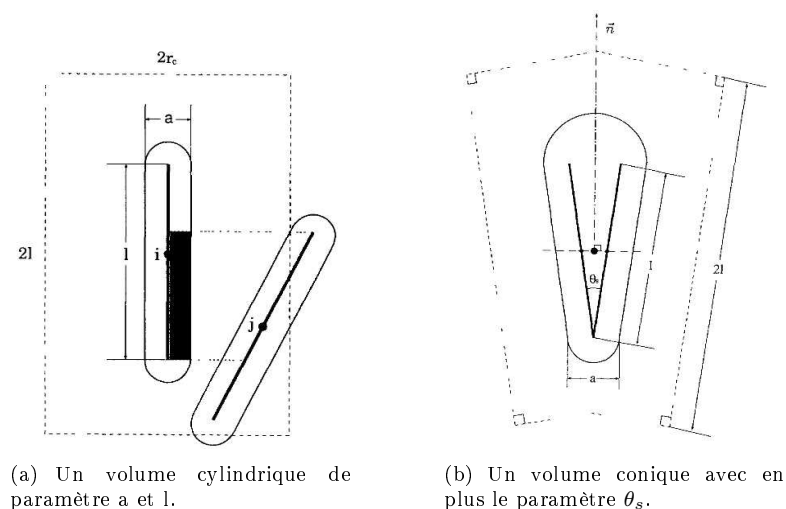


FIG. 3.4.2 – Un modèle par barre dure (ou *hard-rod*) de deux lipides l'un en face de l'autre (d'après Morikawa et Saito [MS94]).

Les seuls autres modèles à **1 seul grain par lipide** sont ceux de Ayton et Voth [AV02] et de Drouffe et al. [DML91]. A notre connaissance, il n'existe pas de modèles ayant un seul type de grain et modélisant un lipide avec plusieurs grains de cet unique type.

Ces modèles à un seul type autorisent la simulation de systèmes comptant plusieurs milliers de molécules mais perdent certaines propriétés intéressantes de la membrane. Hormis Morikawa et Saito qui associent à deux lipides opposés un volume cylindrique paramétré (Fig.3.4.2), le choix de prendre un grain par lipide, c'est-à-dire de réduire une molécule relativement longiligne ($\sim 30\text{\AA}$) à une sphère centrée sur son centre de gravité, ne permet pas de rendre suffisamment compte de l'encombrement stérique et la conformation de cette molécule. Le maillage de sphères résultant ne peut pas modéliser la rigidité ou les modes d'ondulation de la membrane.

► 2 types de grain

De façon assez naturelle, on trouve de nombreux modèles ayant **2 types** de grains pour représenter les molécules amphiphiles comme les phospholipides : un type **tête** (polaire, donc hydrophile) et un type **queue** (apolaire et par conséquent hydrophobe). Le nombre de grains par type varie d'un modèle à un autre. Il est possible d'attribuer plus ou moins de détail aux queues ou à la tête en augmentant le nombre de grains qui y sont dédiés.

A partir de ces 2 types, on trouve des modèles utilisant **3 grains** par lipide soit 1 grain pour la tête et 2 grains pour la queue. Cooke et al.[CD05] ont ainsi modélisé des lipides flexibles et Noguchi et Takasu[NT01] des trimères rectilignes rigides.

Sintes et Baumgärthner [SB97, SB98b] ont quant à eux modélisé des lipides par des chaînes flexibles ayant chacune **5 grains**, où chaque grain représente 3 ou 4 groupes chimiques CH_2 . La tête des lipides est représentée pour un seul grain, les autres grains modélisant la queue. Avec un tel modèle ils seront les premiers à proposer une modélisation implicite du solvant (voir section 3.4.4).

Lenz et Schmid [LS05] ont développé un modèle où chaque lipide est constitué de **7 grains**. La figure 3.4.3 montre 6 grains de même taille par queue (gris clair) et 1 grain plus gros pour la tête (gris foncé).

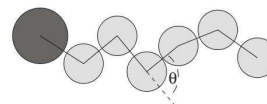


FIG. 3.4.3 – Modèle à 7 grains (d'après Lenz et Schmid [LS05]).

Le nombre de grains consacrés aux têtes et aux queues peut varier en fonction de la structure

modélisée.

Pour Stevens [SHW03], les lipides constitués de **11 à 19 grains** sont construits en liant de 4 à 8 grains hydrophobes pour chacune des deux queues aliphatiques à 3 grains hydrophiles pour la tête. Dans [Ste04] et [Ste05], Stevens étudie la formation de domaine dans une bicouche constituée de deux sortes de lipides différenciés par le nombre N_T de grains pour chaque queue (les têtes restent à 3 grains). Un premier système contient 12096 lipides avec $N_T = 4$ et $N_T = 8$ par queue dans une proportion 2 :1. Un second système contient des lipides avec $N_T = 4$ et $N_T = 6$ par queue dans une proportion 1 :1 et correspond à un système de référence (DLPC :DSPC). L'idée étant qu'en faisant varier le nombre N_T de grains de type queue, il modélise différents types de lipides.

Venturoli et al. [VS99] ont étendu le modèle de Groot et Warren [GW97] en rajoutant un type dédié au solvant. Les types sont notés (w) pour le solvant (*water*), (h) pour la tête hydrophile et (t) pour la queue hydrophobe. La figure 3.4.4 montre deux types de structure avec des grains tête (gris foncé) et des grains queues (blancs). Un tensioactif est une chaîne linéaire de grains h et de grains t. Par exemple une chaîne linéaire avec 1 grain tête et 5 grains queue est noté ht_5 . L'effet de la longueur de la queue sur la bicouche lipidique peut être ainsi étudié avec des lipides de tailles différentes. Par exemple un phospholipide tel la dimyristoylphosphatidylcholine (DMPC) peut être représenté par 3 grains tête rattachés à 5 grains pour chaque queue, ce qui porte le nombre total à **13 grains** et sera noté $h_3(t_5)_2$. Venturoli et al. [VSS05] ont étendu leur modèle à grains pour intégrer des peptides.

Kranenburg et al. [KNS04] s'inspirent des paramètres utilisés par Groot et Rabone [GR01] pour évaluer deux modèles de lipides. La figure 3.4.5 montre ces deux modèles qui se différencient par le nombre de grains attribués pour chaque portion du lipide. Les grains têtes sont gris et les grains queues sont blancs. Les volumes par grain sont de 30Å^3 (**38 grains** Fig.3.4.5(a)) et de 90Å^3 (**13 grains** Fig.3.4.5(b)).

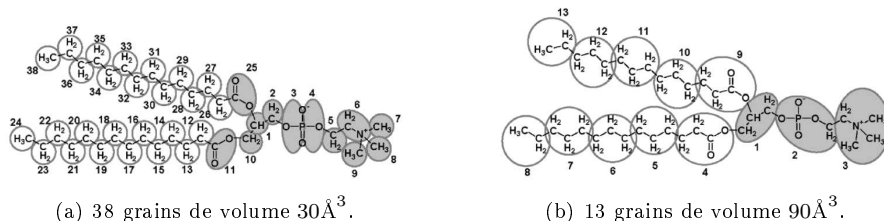


FIG. 3.4.5 – Différentes représentations de la même molécule de DMPC (d'après Kranenburg et al. [KNS04]).

Rien qu'avec 2 types de grain, certains auteurs proposent des modèles proches d'une représentation *atome-unifié*, c'est-à-dire pourvus d'un grand nombre de grains. Plus le nombre de grains est important, plus on modélise l'encombrement stérique, c'est-à-dire le volume de la molécule, ce qui permet entre autre de mieux gérer les problèmes de collision. Par ailleurs, l'ajout de liaisons et d'angles sujets à des potentiels de flexion peuvent rendre compte précisément de la flexibilité du lipide.

Définir deux types, un type tête et un type queue, permet de modéliser la nature amphiphile d'un lipide. Il semble néanmoins que pour une bonne prise en compte de la structure membranaire, une gestion plus fine du lipide soit nécessaire. Dans le paragraphe suivant, nous présentons les modèles à trois types de grain.

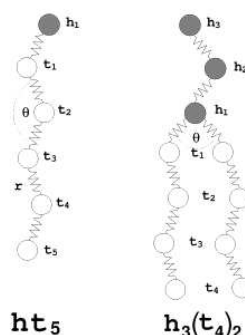


FIG. 3.4.4 – Tête et queues peuvent être représenté par un nombre variable de grains (d'après Venturoli et al. [VS99]).

► 3 types de grain

Farago [Far03] propose un modèle qui affiche seulement 2 types mais utilisant 3 grains pour modéliser un phospholipide rigide : 2 grains hydrophobes et 1 grain hydrophile (Fig.3.4.6(a)). Toutefois, les potentiels intermoléculaires qui sont appliqués aux deux grains hydrophobes sont différents non seulement par les valeurs des paramètres (Tab.3.4.2) mais aussi par les formes analytiques des potentiels type Lennard-Jones utilisés (Fig.3.4.12). Ce modèle peut être considéré comme un modèle à 3 types de grain.

Brannigan et al. [BPB05] propose d'utiliser **5 grains** pour un lipide (Fig.3.4.6(b)). Le grain numéroté 1 est identifié comme le groupe tête hydrophile. Le grain 2 est associé à l'interface entre les parties hydrophiles et hydrophobes. Les grains 3 à 5 correspondent à la région queue hydrophobe de la molécule amphiphile.

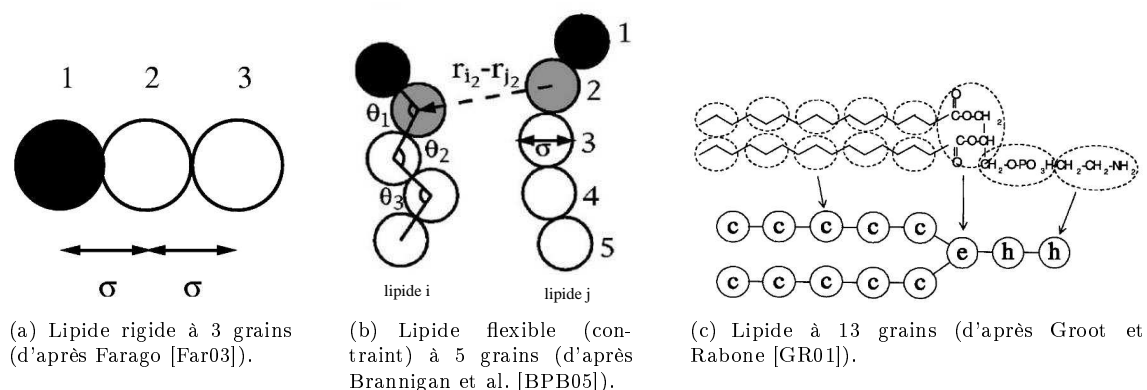


FIG. 3.4.6 – Différents modèles utilisant 3 types de grain pour représenter un lipide.

Groot et Rabone [GR01] étendent le modèle initialement proposés par Groot et Warren [GW97] en rajoutant un type de grain supplémentaire pour les lipides (Fig.3.4.6(c)). Le phospholipide modélisé est une phosphatidyl-éthanolamine (PE) avec deux queues en C_{15} (acide palmitique) ce qui correspond à une molécule de 1,2-dipalmitoyl-phosphatidyl-éthanolamine (PPPE). Cette PPPE est représentée par 5 grains de type (c) pour les queues hydrophobes, 1 grain de type (e) pour le lien ester (c'est-à-dire l'interface entre les parties hydrophiles et hydrophobes) et 2 grains de type (h) pour la tête hydrophile, soit un total de **13 grains**.

► Plus de 3 types de grain

Nous passons en revue dans cette section des modèles à plus de 3 types de grain.

La figure 3.4.7(a) présente le modèle à **4 types** de grain de Lyubartsev [Lyu05] : la choline (noté N en bleu), le phosphate (noté P en jaune), le glycerol (noté CO en rouge) et les groupes carboné des queues (notés CH en vert). Les lipides sont constitués d'un groupe tête de 2 grains liés par l'intermédiaire de deux grains glycérol à deux queues ayant 4 grains chacune, soit un total de **10 grains**.

Klein et al. [NSK04] ont développé un modèle mésoscopique qui est explicitement paramétré pour le DMPC et propose plus de types de grain, **5 types** exactement, que le modèle précédent de Marrink. La figure 3.4.7(b) montre une molécule de DMPC composée de **13 grains** pour représenter les 118 atomes : 8 grains hydrophobes pour les deux queues et 5 hydrophiles pour la tête : 1 grain positivement chargé pour la choline, 1 grain négativement chargé pour le phosphate, 1 grain glycerol et 2 grains ester rattachés aux deux queues.

Izvekov and Voth [IV05] utilisent le même modèle avec seulement une dénomination différente : (CH) pour le groupe choline, (PH) pour le groupe phosphate, (GL : $CH_2-CH-CH_2$) pour le groupe glycerol, (E1 et E2) pour les deux groupes ester, (SM : $(CH_2)_3$) pour les triplés de carbones de la queues et (ST : $(CH_2)_2-CH_3$) pour les deux extrémités des chaînes hydrophobes.

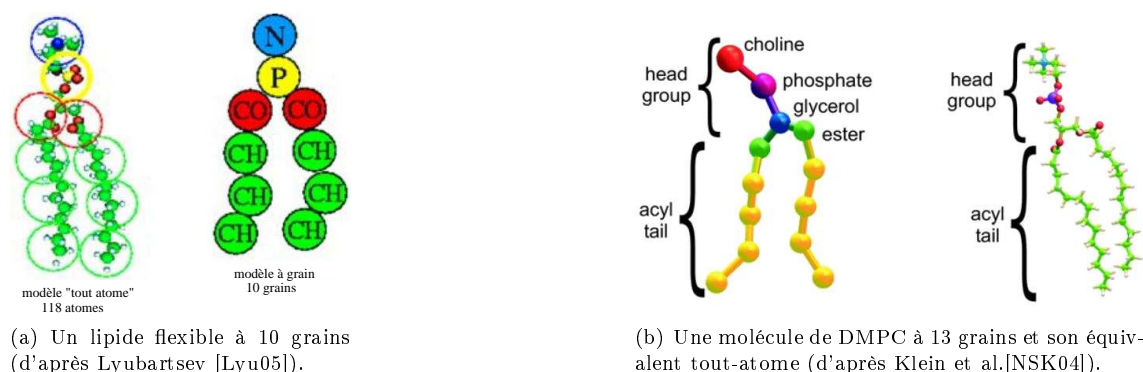


FIG. 3.4.7 – Différents modèles utilisant plus de 3 types de grain pour représenter un lipide.

► Des types et des sous-types

Marrink et al. [MM04] ont proposé un modèle capable d'inclure la nature chimique des espèces moléculaires tout en définissant aussi peu de types différents de grain que possible (Fig.3.4.8). Ainsi ils disposent de **4 types** de grain pour construire facilement différentes architectures de molécules : polaire (P), nonpolaire (N), apolaire (C), et chargé (Q). Les sites polaires (P) représentent les groupes neutres d'atomes qui peuvent se dissoudre facilement dans l'eau (e.g., éthylène glycol), les sites apolaires (C) représentent les parties hydrophobes (e.g., butane), et les groupes nonpolaires (N) sont utilisés pour les groupes mixés partiellement polaires, partiellement apolaires (e.g., propanol). Les grains chargés (Q) sont réservés pour les groupes ionisés (e.g., ammonium).

Pour les types (N) et (Q), **4 sous-types** sont distingués : (0, d, a, et da). Ces sous-types permettent des réglages fins des interactions sur la base de la nature chimique des atomes représentés par les groupes CG. Le sous type (0) est appliqué aux groupes d'atomes incapables d'induire des liaisons hydrogènes, (d) et (a) aux groupes qui peuvent intervenir dans une liaison hydrogène respectivement en temps que donneur ou accepteur et (da) aux groupes capables d'être donneurs et accepteurs. Suivant ce modèle, chaque grain doit avoir une équivalence au solvant de 4 molécule d'eau.

Un des avantages de ce modèle est que différents phospholipides peuvent être construits avec les mêmes "briques" structurales. Par exemple, pour calculer les aires par lipide de plusieurs phospholipides, la taille des queues peut être modifiée en enlevant ou rajoutant des grains apolaires. Ainsi une queue constituée d'un acide laurique est modélisée avec 3 grains (C), l'acide myristique et palmitique avec 4 grains (C), et l'acide stéarique avec 5 grains (C) (cf Tab.1.3.1). Un groupe de tête pour la phosphatidyl-éthanolamine (PE) peut être modéliser par deux grains (Q) tout comme la PC, mais dans ce cas, ils seront du sous-type plus polaire (da) pour imiter les capacités accrues de la PE à être donneur de pont hydrogène.

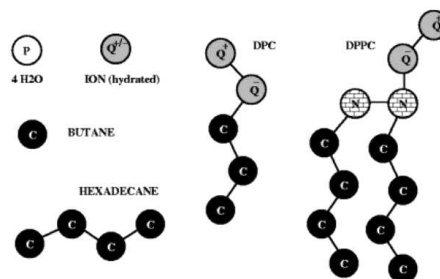


FIG. 3.4.8 – Modèle à grains générique : eau, ions, butane, hexadecane, DPC et DPPC (d'après Marrink et al. [MM04]).

Intérêt d'un modèle à 3 types de grain et à 3 grains par lipide

De l'étude des différents modèles à grains, nous pouvons retenir plusieurs points :

- les modèles à 1 type de grain associent un grain par lipide, ce qui produit des structures qui ne dotent pas à la membrane modélisée des propriétés importantes telles que la rigidité ou les modes d'ondulation ;

- les modèles à 2 types permettent de prendre en compte la nature amphiphile des lipides avec une tête hydrophile et une queue hydrophobe mais les modèles à 3 types produisent des membranes avec plus de cohésion, et ce même si la bicouche se trouve en phase fluide, c'est-à-dire avec des lipides désordonnés ;
- les modèles avec plus de 3 types de grain sont capables de représenter plus finement les propriétés des différentes portions d'un lipide mais ils requièrent un calibrage plus complexe.

En ce qui concerne le nombre de grains par lipide, plus il y a de grains pour un lipide, plus son encombrement stérique et sa conformation peuvent être représentés correctement. Néanmoins, l'augmentation du nombre de points d'interactions induit des simulations plus coûteuses en terme de temps de calcul.

Pour notre étude, nous avons choisi de prendre comme référence le modèle d'O.Farago [Far03] qui propose de modéliser les lipides avec 3 grains, chaque grain étant d'un type différent. Comme nous le montrerons au chapitre 4, ce modèle sera étendu pour mieux couvrir notre champ d'application.

Nous allons nous intéresser maintenant aux interactions intramoléculaires des modèles à grains, c'est-à-dire la représentation flexible ou rigide des lipides. En effet, à partir de 2 grains ou 3 grains par lipide pour définir 1 liaison ou 1 angle entre 2 liaisons, il faut se poser la question de la prise en compte de la flexibilité du lipide, et ce quelque soit le nombre de types de grain. En ce qui concerne cette dynamique interne, deux approches diamétralement opposées sont possibles (Fig.4.2.1) :

- un **modèle flexible** permettant aux grains de modifier leurs positions relatives ;
- un **modèle rigide** imposant des positions relatives constantes.

La section suivante contient une présentation de ces deux approches.

3.4.2 La modélisation flexible des lipides

De l'étude des différents modèles à grains, il semble que l'utilisation d'un grand nombre de grains par lipide aille de paire avec l'utilisation d'un modèle flexible mais qu'il n'y ait pas de corrélation évidente entre le nombre de types de grain et le degré de flexibilité du lipide.

La section 3.3.1 a présenté les champs de force de dynamique moléculaire avec les différents types d'interactions entre les atomes d'une même molécule : un potentiel de liaison entre deux atomes, de flexion entre deux liaisons, de torsion autour d'une liaison et d'autres termes de contribution négligeable. Dans le cas des modèles à grains, il s'avère que seuls les potentiels de liaison et de flexion sont utilisés. Par exemple, la figure 3.4.9 montre une molécule générique modélisée par 5 grains (A) et la modélisation flexible qui en est faite (B) grâce à des potentiels de liaison et de flexion. La section suivante fournit différentes expressions pour ces deux types de potentiel.

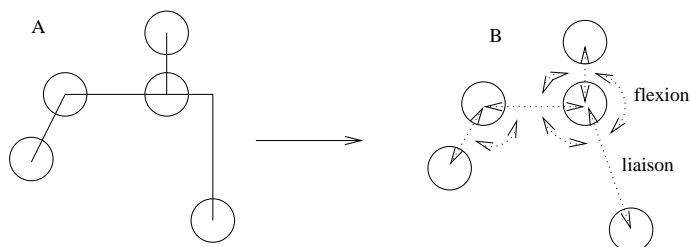


FIG. 3.4.9 – Modèle flexible d'une molécule générique par des potentiels de liaison et de flexion.

► Les potentiels de liaison

Les potentiels de liaison expriment une énergie en fonction de la distance entre deux grains, les écarts par rapport à la distance à l'équilibre se traduisant par des forces qui s'opposent à l'écartement des grains. Nous allons présenter maintenant différentes expressions mathématiques

de ces forces qui ont été utilisées en général pour modéliser des membranes à base de phospholipides de type DMPC.

➤ Le potentiel harmonique

L'expression type du potentiel harmonique est donnée en Eq.(3.4.1). Dans cette équation, r est la distance entre deux grains, K (ou k_{bond}) la constante de raideur et r_0 (ou σ) la distance à l'équilibre.

$$E = K(r - r_0)^2 \quad (3.4.1)$$

Venturoli et al. [VS99] utilisent les ressorts harmoniques initialement décrits par Groot et Warren dans [GW97] Eq.(3.4.2) avec $k_r = 2$. Toutefois la première forme Eq.(3.4.2) ne retient pas les grains en dessous du rayon de coupure ou *cutoff* r_c ce qui laisserait les chaînes s'entrecroiser sans qu'il n'y ait d'interaction. Pour résoudre ce **problème de superposition des structures**, M.Venturoli[Ven04] propose d'utiliser une seconde expression de la force de liaison sous la forme d'un **ressort de Hookean** Eq.(3.4.3).

$$U_{spring}(r_{i,i+1}) = \frac{k_r}{2} r_{i,i+1}^2 \Rightarrow \vec{F}_{spring}(r_{i,i+1}) = K_r \vec{r}_{i,i+1} \quad (3.4.2)$$

$$\vec{F}_{spring}(r_{i,i+1}) = K_r (r_{i,i+1} - r_{eq}) \hat{r}_{i,i+1} \quad (3.4.3)$$

Pour Marrink et al. [MM04], les interactions entre grains d'un même lipide sont décrites par un potentiel harmonique faible $V_{bond}(r)$ avec une distance à l'équilibre $R_{bond} = \sigma = 0.47nm$ identique pour tous les types. Dans ce modèle, la distance entre deux grains liés est en moyenne plus petite que celle entre deux grains non liés pour lesquels la distance à l'équilibre est $2^{1/6}\sigma$. La constante de force $K_{bond} = 1250kJ.mol^{-1}nm^{-2}$ autorise d'importants écarts par rapport à la distance d'équilibre ($\sim 15\%$). Dans ce modèle, les interactions de Lennard-Jones sont exclues entre les grains liés.

$$V_{bond}(r) = 1/2K_{bond}(r - R_{bond})^2 \quad (3.4.4)$$

➤ Le potentiel harmonique dérivé

Sintes et Baumgärthner [SB97, SB98b] modélisent des lipides flexibles grâce à un potentiel type harmonique Eq.(3.4.5). Les paramètres sont : $\epsilon_b = 250kcal/mol$, $b_0 = 3.9\text{\AA}$ et la distance de coupure pour les liaisons harmoniques $|b - b_0| > 3\text{\AA}$. Ce potentiel concernant les grains d'une même molécule est à additionner avec ceux modélisant les interactions intermoléculaires et l'hydrophobie (cf Eq.(3.4.19)).

$$U_{bond} = \sum_{bonds} \epsilon_b (b/b_0 - 1)^2 \quad (3.4.5)$$

➤ Le potentiel FENE

Le potentiel FENE (*finite extensible nonlinear elastic*) est introduit par Kremer et Grest [KG90]. L'expression type de ce potentiel est donnée en Eq.(3.4.6) avec r la distance entre deux grains, K une constante énergétique et r_0 la distance à l'équilibre.

$$E = -\frac{1}{2}K r_0^2 \ln \left[1 - \left(\frac{r}{r_0} \right)^2 \right] \quad (3.4.6)$$

Dans le modèle de Stevens [SHW03], le potentiel de liaison U_{bond} Eq.(3.4.8) est la somme d'un potentiel attractif FENE avec la partie purement répulsive du potentiel de Lennard-Jones U_{LJ} (cf Eq.(3.4.29) de la section 3.4.5) avec une distance de coupure courte $r_c = 2^{1/6}\sigma$ (σ étant l'unité référence de longueur). La juxtaposition de ces deux potentiels permet d'utiliser le même pas de

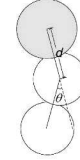
temps d'intégration pour les interactions liantes et non liantes.

$$V_{bond}(r) = -\frac{1}{2}k_{bond}r_{\infty}^2 \log [1 - (r/r_{\infty})^2] \quad (3.4.7)$$

$$U_{bond} = -\frac{1}{2}k_0R_0^2 \ln(1 - r^2/R_0^2) + U_{LJ}(r) \quad (3.4.8)$$

$$V_s(d) = \begin{cases} -\frac{1}{2}k_s d_s^2 \ln \left\{ 1 - [(d - d_0)/d_s]^2 \right\} & \text{pour } |d - d_0| < d_s \\ \infty & \text{pour } |d - d_0| > d_s \end{cases} \quad (3.4.9)$$

Wang et Frenkel [WF05a] utilisent ce type de potentiel pour leur lipide flexible à 3 grains (Fig.3.4.10), sous l'écriture Eq.(3.4.9) où d est la distance entre les deux grains, k_s la constante du ressort, d_0 la distance au repos et d_s la distance de coupure.



Dans ce modèle, le potentiel FENE devient un simple potentiel harmonique pour $d \approx d_0$ et a une distance de coupure logarithmique à $d = d_0 \pm d_s$.

FIG. 3.4.10 – Potentiel FENE pour un trimère flexible (d'après Wang et Frenkel [WF05a]).

► Le potentiel FENE étendu

Lenz et Schmid [LS05] utilisent une forme étendue Eq.(3.4.10) du potentiel FENE permettant de modéliser des variabilités dans les tailles des grains par le facteur Δ (positif ou négatif) ajouté à r .

$$E = -\frac{1}{2}KR_0^2 \ln \left[1 - \left(\frac{r - \Delta}{R_0} \right)^2 \right] \quad (3.4.10)$$

Après avoir étudié différentes manières de maintenir une distance entre deux grains, il reste à traiter la question de la rigidité de la structure.

Certains modèles n'appliquent pas de contrainte de rigidité, comme par exemple Lyubartsev [Lyu05] dans sa modélisation d'un lipide par 10 grains pris parmi 4 types (Fig.3.4.7(a)). Les structures sont alors libres "d'onduler" ce qui semble se révéler assez problématique dans le cas des membranes fluides.

La section suivante présente différentes expressions pour le potentiel de flexion.

► Les potentiels de flexion

Les potentiels de flexion expriment une énergie en fonction de l'angle entre deux liaisons consécutives.

► Le potentiel harmonique

L'expression du potentiel harmonique est donnée en Eq.(3.4.11) avec K une constante énergétique (éventuellement appelée constante de raideur) et θ_0 l'angle à l'équilibre.

$$E = K(\theta - \theta_0)^2 \quad (3.4.11)$$

Cooke et al. [CD05] utilisent ce type de potentiel sous la forme réécrite Eq.(3.4.12) avec $\frac{1}{2}k_{bend}\sigma^2\vartheta^2$ pour l'angle $\pi - \vartheta$ entre les trois grains.

$$V_{bend}(\vartheta) = \frac{1}{2}k_{bend}(\vartheta - 4\sigma)^2 \quad (3.4.12)$$

Venturoli et al. [VS99] utilisent ce type le potentiel harmonique sous la forme Eq.(3.4.13) pour pouvoir tester l'impact des chaînes insaturées sur la rigidité de la membrane. Les paramètres sont : θ l'angle entre deux liaisons consécutives, la rigidité du potentiel $k_{\theta} = 3$ et l'angle

à l'équilibre $\theta_0 = \pi$. Si aucun angle au repos n'est précisé, alors le lipide est complètement flexible. Une résistance à la flexion peut être modélisée par un angle d'équilibre $\theta_0 = 180^\circ$. Pour modéliser une chaîne carbonée cis-insaturée due à une double liaison, un angle à l'équilibre $\theta_0 = 135^\circ$ peut être utilisé dès lors qu'une cis-insaturation induit un coude dans la chaîne lipidique.

$$U_{bend}(r_{i-1,i}, r_{i,i+1}) = \frac{k_\theta}{2}(\theta - \theta_0)^2 = \frac{k_\theta}{2} \left(\frac{\vec{r}_{i-1,i} \cdot \vec{r}_{i,i+1}}{|\vec{r}_{i-1,i}| |\vec{r}_{i,i+1}|} - \cos\theta_0 \right)^2 \quad (3.4.13)$$

Dans le modèle de Stevens [SHW03], la flexibilité du lipide s'écrit sous la forme de Eq.(3.4.14) avec θ l'angle formé par deux liaisons successives, $k_\theta = 2\epsilon/\text{rad}^2$ et $\theta_0 = 180^\circ$. La **géométrie de la tête est ainsi maintenue** et la géométrie d'une des queues permet de différencier les lipides saturés des lipides insaturés.

$$U_{bend} = k_\theta(\theta - \theta_0)^2 \quad (3.4.14)$$

➤ Le potentiel cosinus

L'expression du potentiel de type cosinus est donnée en Eq.(3.4.15) avec θ l'angle entre deux liaisons consécutives et K une constante énergétique. De nombreux modèles (Brannigan et al. [BPB05], Wang et Frenkel [WF05a], Lenz et Schmid [LS05] et Goetz et Lipowsky [GL98]) ont utilisé ce potentiel dans des versions modifiées pour des modèles de 3 à 5 grains.

$$E = K(1 + \cos\theta) \quad (3.4.15)$$

➤ Le potentiel cosinus carré

L'expression type de ce potentiel est donnée en Eq.(3.4.16) avec θ l'angle entre deux liaisons, K une constante énergétique et θ_0 l'angle à l'équilibre.

$$E = K(\cos\theta - \cos\theta_0)^2 \quad (3.4.16)$$

Marrink et al. [MM04] utilisent ce type de potentiel qu'ils disent "*faible*" sous l'écriture légèrement différente Eq.(3.4.17). A la base, l'angle à l'équilibre $\theta_0 = 180^\circ$, avec une constante de force $K_{angle} = 25\text{kJ}\cdot\text{mol}^{-1}\cdot\text{rad}^{-2}$ qui autorise des déviations de 30° .

$$V_{angle}(\theta) = 1/2K_{angle}(\cos(\theta) - \cos(\theta_0))^2 \quad (3.4.17)$$

En plus de pouvoir construire différents phospholipides avec les mêmes "*briques*" structurales, un autre des avantages de ce modèle est que le paramétrage des potentiels d'interaction n'est pas lié à un type de lipide. Par exemple, pour calculer les aires par lipide de plusieurs phospholipides, les coude introduits par les doubles liaisons cis des acides gras insaturés peuvent être modélisés par un angle à l'équilibre $\theta_0 = 120^\circ$ et en augmentant la rigidité de la chaîne légèrement $K_{angle} = 35\text{kJ}\cdot\text{mol}^{-1}\cdot\text{rad}^{-2}$.

Sintes et Baumgärthner [SB97, SB98b] contraignent également la rigidité de leur lipide à 5 grains avec ce type de potentiel. Les paramètres sont $\theta_0 = 180^\circ$ et $\epsilon_0 = 15\text{kcal/mol}$ pour Eq.(3.4.18).

$$U_{angles} = \sum_{angles} \epsilon_0(\cos\theta + 1)^2 \quad (3.4.18)$$

➤ Intérêts et limitations d'une modélisation flexible des lipides

L'utilisation de potentiels pour modéliser les liaisons entre les grains et les angles entre ces liaisons permet de contraindre la géométrie des lipides. La flexibilité induite permet d'obtenir dans certains cas des résultats plus cohérents avec les données issues d'expérimentations ([NTN] et [LS95]). Malheureusement les différents potentiels intramoléculaires ainsi modélisés conduisent à l'application de forces intramoléculaires bien plus élevées que celles induites par les potentiels

intermoléculaires. Or pendant une simulation, le pas de temps utilisé est lié à l'intensité des forces. Une trop grande disparité entre l'intensité des forces intra- et inter-moléculaires conduit à consacrer l'essentiel des calculs réalisés lors d'un pas de temps au calcul des forces intramoléculaires. Comme nous allons le voir dans la section suivante, il est possible de retirer cette charge calculatoire en optant pour un modèle rigide.

3.4.3 La modélisation rigide des lipides

Une autre solution pour gérer la modélisation des lipides est d'utiliser un modèle rigide.

En général, la cinétique des modèles rigides fait appel à une **dynamique des corps rigides** [DML91] [NT01] [Far03]. Dans le contexte des modèles à grains, les interactions entre les molécules opèrent sur les grains. Les forces et les couples induits sont cumulés au niveau du centre de gravité du lipide et génèrent les déplacements linéaires et les rotations de la structure.

Wanga et Frenkel font remarquer dans [WF05a, WF05b] qu'avec des modèles rigides capables de reproduire des phénomènes d'auto-assemblage et de transition de phase cohérents avec les données observées, il n'est toutefois pas évident d'étudier des mécanismes qui sont très sensibles à la flexibilité des queues des lipides comme la formation de défauts dans les membranes et l'apparition de pores.

Néanmoins cette approche par corps rigide simplifie la gestion des conformations complexes des molécules et permet, nous le verrons par la suite, de lier le pas de temps d'intégration uniquement à l'intensité des potentiels intermoléculaires. En d'autres termes, l'utilisation d'un modèle rigide évite de consacrer du temps à calculer les petites oscillations des grains autour des valeurs d'équilibre. C'est pour ces raisons que, bien que le modèle que nous développerons au chapitre 4 soit capable d'intégrer des modèles flexibles ou rigides, nous avons choisi dans un premier temps d'implémenter le modèle rigide.

Après avoir étudié les différentes possibilités de prendre en compte les interactions intramoléculaires, la section suivante s'intéresse à la modélisation du solvant.

3.4.4 La modélisation du solvant : explicite / implicite

On considère en général en modélisation de membrane biologique que ces membranes sont en environnement aqueux. Il est donc indispensable de fournir une représentation de cet environnement. Dans les modèles mésoscopiques de systèmes eau-lipides, deux approches sont possible pour prendre en compte le solvant :

- une représentation explicite avec l'utilisation de grains regroupant un certain nombre de molécules d'eau ;
- une représentation implicite par l'utilisation de forces effectives qui agissent sur les lipides pour former des agrégats et pour garantir leur stabilité dans le solvant.

Dans ce qui suit, nous allons étudier dans un premier temps les approches explicites de modélisation du solvant puis nous évaluerons des techniques pour abstraire les grains d'eau en se focalisant sur le rendu de l'hydrophobie des lipides à l'échelle mésoscopique.

► Solvant explicite

Dans le cas d'un solvant explicite, les molécules d'eau sont généralement regroupées par 3 ou 4 pour former un grain, et ce, pour des raisons d'équivalence en masse avec les grains dédiés aux solutés. L'intérêt d'un solvant explicite réside dans la modélisation fine de la pression que les grains d'eau exercent sur chaque feuillet d'une membrane, cette pression garantissant en grande partie la cohésion de la bicouche.

Il existe deux moyens de spécifier ces nouveaux grains :

- soit ce sont des grains de **même type que les grains hydrophobes** de type tête (Stevens [SHW03]).

- soit ce sont des grains d'un **type supplémentaire** : Goetz et Lipowsky [GL98], Venturoli et al. [VS99] définissent le type (w) représentant 3 molécules d'eau pour un volume évalué à $9nm^3$ par grain, Marrink et al. [MM04] définissent le type (P) pour des grains polaires représentant 4 molécules d'eau et enfin Klein et al. [NSK04] utilisent un calibrage des interactions eau-eau pour obtenir la température de fusion (énergie) et la densité de l'eau à température ambiante (volume).

La définition d'un nouveau type revient, par voie de conséquence, à rajouter des paramètres au modèle initial, ce qui risque de compliquer la phase de calibrage. Par ailleurs, le solvant est un bon candidat pour un traitement à moindre niveau de détail car il joue souvent un rôle de spectateur. Il compte néanmoins pour une partie non négligeable du système et se révèle être très coûteux en calcul si on choisit par exemple de traiter en détail les interactions électrostatiques longues distances.

➤ Solvant implicite

L'idée d'une modélisation implicite du solvant repose sur le fait que les interactions hydrophobes ne sont pas explicitement prises en compte. Les temps de calculs peuvent être ainsi significativement réduits, dès lors que les mouvements des particules de solvant, qui représentent une fraction substantielle du système, ne sont plus explicitement pris en compte. De fait, une des difficultés d'une représentation "*water-free*" est de restituer le comportement hydrophobe des molécules amphiphiles à l'origine de la formation d'agrégats, les empêchant ainsi de se dissoudre dans le solvant.

Dans un modèle à solvant implicite, l'obtention d'une bicouche stable nécessite l'intervention d'une **force externe**. Il existe plusieurs méthodes pour définir cette force, soit en contraignant les têtes des lipides à se coller à une surface, soit à se lier entre elles. On peut également utiliser des potentiels effectifs d'attraction entre les lipides pour maintenir les structures macroscopiques. Nous allons maintenant donner les détails de ces solutions.

➤ Solvant implicite avec contrainte de l'environnement

Sintes et Baumgärthner [SB97, SB98b] ont été parmi les premiers à développer un modèle à grains à solvant implicite pour les bicouches lipidiques. Afin de maintenir la cohésion de la bicouche, les têtes des lipides sont **attachées à des films continus** qui laissent les molécules diffuser librement à proximité de la surface de ces films. L'expression du potentiel global est donnée par Eq.(3.4.19) ce qui correspond respectivement aux liaisons entre deux grains (U_{bond} est donné par Eq.(3.4.5)), aux flexions entre deux liaisons consécutives (U_{angles} est donné par Eq.(3.4.18)) et aux interactions stériques ($U_S(r_{ij})$) et hydrophobes ($U_H(z)$).

$$U = U_{bond} + U_{angles} + U_S(r_{ij}) + U_H(z) \quad (3.4.19)$$

$U_S(r_{ij})$ est un potentiel à coeur dur utilisé entre les molécules mais aussi entre les grains séparés par plus d'une liaison. $U_H(z)$ pour représenter les interactions fortes des têtes polaires avec la surface de l'eau. Ce phénomène est modélisé par le potentiel suivant :

$$U_H(z) = \begin{cases} 0 & \text{pour } 40 - \sigma_L < z < \sigma_L \\ \infty & \text{pour } \sigma_L < z < 40 - \sigma_L \end{cases} \quad (3.4.20)$$

Ainsi les lipides sont contraints de diffuser à proximité des deux surfaces ou interfaces planes, l'épaisseur de la membrane étant de 40\AA .

Lenz et Schmid [LS05] ont développé un modèle à solvant implicite où la stabilité induite par le solvant est obtenue en considérant deux types d'approche. La première approche, inspirée du modèle de Sintes et Baumgärtner [SB97, SB98b], consiste à forcer les lipides à former une bicouche en les **confinant entre deux plans parallèles** : le plan (x,y) et celui défini par $z = z_{upper} > 0$. Les grains de type queue sont confinés par le potentiel $V_{ST}(z)$ Eq.(3.4.21) tandis que les grains de

type tête le sont par le potentiel $V_{SH}(z)$ Eq.(3.4.22).

$$V_{ST}(z) = \begin{cases} V_{FENE}(z) & \text{pour } z < 0 \\ V_{FENE}(z - z_{upper}) & \text{pour } z_{upper} < z \\ 0 & \text{sinon} \end{cases} \quad (3.4.21)$$

$$V_{SH}(z) = \begin{cases} V_{FENE}(z) & \text{pour } 0 < z < \frac{1}{2}z_{upper} \\ V_{FENE}(z - z_{upper}) & \text{pour } \frac{1}{2}z_{upper} < z < z_{upper} \\ 0 & \text{sinon} \end{cases} \quad (3.4.22)$$

Un avantage de cette approche illustrée par la figure 3.4.11(a) est le moindre coût en temps de calcul. Néanmoins les plans qui confinent les têtes limitent les déformations et les ondulations possibles de la bicouche.

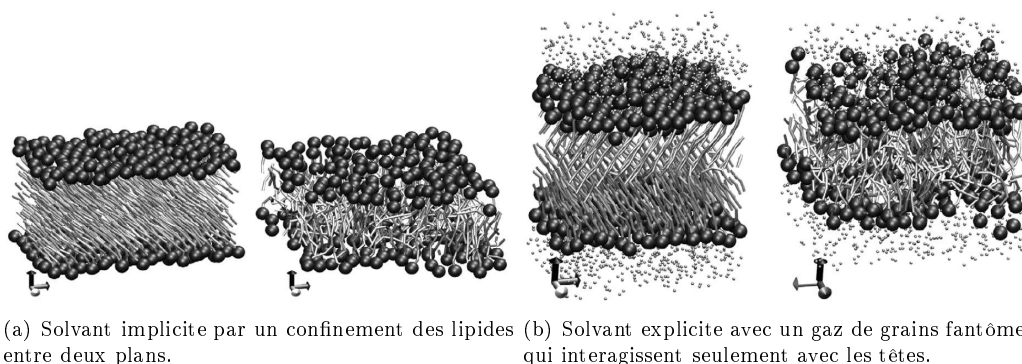


FIG. 3.4.11 – Deux approches pour prendre en compte le solvant (d’après Lenz et Schmid [LS05]). Dans les deux cas, l’image de gauche correspond à une phase gel ($p = 1.0$ et $T = 0.9$) et celle de droite, à une phase liquide ($p = 1.0$ et $T = 1.0$) avec p et T respectivement la pression et la température.

Cette limitation est résolue par la seconde approche qui propose de stabiliser la bicouche grâce à un **gaz environnant de grains fantômes** de solvant. Ces grains n’interagissent pas entre eux mais uniquement avec les têtes des lipides (Fig.3.4.11(b)). Les deux modèles ont été validés en reproduisant différentes manipulations qui montrent que par exemple, le changement de la valeur d’un paramètre comme la température, modifier la membrane en la faisant passer d’une phase gel à une phase fluide.

Une autre approche de modélisation par solvant implicite consiste à **inclure les interactions hydrophobes dans les interactions effectives** entre lipides résultant d’un potentiel par paire. Les paragraphes suivants expliquent plusieurs formules de potentiels incluant cette approche. Toutefois, on notera dès à présent que ces types d’interactions effectives peuvent aboutir à la formation de membranes qui ne sont pas fluides (quelque soit la température considérée) ou qui se désagrègent simplement à partir d’une certaine température (Drouffe et al. [DML91]).

> Solvant implicite par potentiel multi-corps

Les potentiels multi-corps (*multibody potential*) peuvent être utilisés pour représenter implicitement le solvant. Dans ce cas, l’interaction entre les molécules d’eau et les molécules amphiphiles est reproduite par un potentiel qui dépend de la densité locale des grains hydrophobes et qui exprime l’augmentation de l’énergie libre due aux contacts des molécules d’eau avec les acides gras.

Noguchi et Takasu [NT01] utilisent ces potentiels multi-corps entre les grains hydrophobes pour maintenir la cohésion du système. Les lipides interagissent par un potentiel répulsif à coeur mou

($U_{rep}(r)$) additionné à un potentiel attractif "hydrophobe" ($U_{hp}(\rho)$) :

$$U = \sum_{i \neq i'} U_{rep}(|\vec{r}_{i,j} - \vec{r}_{i',j'}|) + \sum_{j=2,3} U_{hp}(\rho_{i,j}) \quad (3.4.23)$$

$$\text{avec } U_{rep}(r) = \epsilon e^{-20(r/\sigma-1)} \quad (3.4.24)$$

$$\text{et } U_{hp}(\rho)/\epsilon = \begin{cases} -0.5\rho & \text{pour } \rho < \rho^* - 1 \\ 0.25(\rho - \rho^*)^2 - c & \text{pour } \rho^* - 1 \leq \rho < \rho^* \\ -c & \text{pour } \rho^* \leq \rho \end{cases} \quad (3.4.25)$$

Les grains se repoussent les uns des autres par le potentiel à coeur mou U_{rep} Eq.(3.4.24) avec la distance de coupure $r_c = 1.3\sigma$. Le potentiel mutli-corps $U_{hp}(\rho)$ "hydrophobe" Eq.(3.4.25) est fonction de la **densité locale** $\rho_{i,j}$ Eq.(3.4.26) des grains hydrophobes i et j où la fonction $h(r)$ Eq.(3.4.27) est amortie (*shifted*) entre 1.6σ et 2.2σ .

$$\rho_{i,j} = \sum_{i \neq i', j'=2,3} h(|\vec{r}_{i,j} - \vec{r}_{i',j'}|) \quad (3.4.26)$$

$$h(r) = \begin{cases} 1 & \text{pour } r < 1.6\sigma \\ \frac{1}{\exp[20(r/\sigma-1.9)]+1} & \text{pour } 1.6\sigma \leq r < 2.2\sigma \\ 0 & \text{pour } 2.2\sigma \leq r \end{cases} \quad (3.4.27)$$

A faible densité ($\rho < \rho^* - 1$), $U_{hp}(\rho)$ est linéaire en densité et se comporte comme un potentiel par paire entre les grains hydrophobes voisins de différentes molécules. A densité élevée, l'interaction attractive atteint une limite indépendante de la densité locale, ce qui évite aux lipides de s'agglutiner en formant des structures extrêmement denses.

Wang et Frenkel [WF05a] ont étendu le modèle de Noguchi et Takasu aux molécules flexibles (Fig.3.4.10) et utilisent un potentiel multi-corps similaire pour imiter l'hydrophobie en intégrant la densité locale $\tilde{\rho}$ des grains hydrophobes :

$$U_{hp}(\tilde{\rho}) = \begin{cases} \epsilon[k_{\tilde{\rho}}(\tilde{\rho} - 0.75) + U_0] & \text{pour } \tilde{\rho} < 0.75 \\ \epsilon[(1/\tilde{\rho})^{12} - 2(1/\tilde{\rho})^6 + 1] & \text{pour } 0.75 \leq \tilde{\rho} < 1 \\ 0 & \text{pour } 1 \leq \tilde{\rho} \end{cases} \quad (3.4.28)$$

➤ Solvant implicite par potentiel par paire

Toujours dans la perspective de diminuer la complexité des simulations, notamment en réduisant la charge calculatoire dédiée au phénomène d'hydrophobie, on peut se passer des potentiels multi-corps qui, à chaque pas de temps et pour chaque grain, requièrent l'évaluation de la densité locale des grains hydrophobes. Par exemple il est possible de tirer partie des interactions effectives par paire entre les molécules pour représenter directement l'hydrophobie.

Brannigan et al. [BPB05] montrent qu'un potentiel attractif par paire suffisamment "couvrant" produit une bicouche fluide stable sans avoir à utiliser un solvant explicite. Il utilise un potentiel attractif qui est **appliqué uniquement entre des grains spéciaux dit d'interface** qui relie les têtes hydrophiles aux queues hydrophobes (en gris dans la figure 3.4.6(b)). Ces grains spéciaux sont chargés d'imiter les interactions hydrophobes et de contrôler la tension interfaciale en l'absence de solvant.

Enfin Farago [Far03] a réussi à modéliser implicitement le solvant grâce à des interactions par paire (cf Fig.3.4.12) tout en modélisant une lipide rigide (Fig.3.4.6(a)). Nous détaillerons ce modèle dans la section 3.4.5 car il a servi de base à la conception de notre modèle de membrane mitochondriale.

➤ Intérêts de la modélisation implicite du solvant

Dans cette section, nous avons vu qu'il était possible de représenter explicitement ou implicitement le solvant.

La première option conserve des interactions par paire entre le solvant et les lipides mais se révèle trop coûteuse en calculs. La seconde option est nettement plus performante mais réclame une attention toute particulière pour ce qui est de la prise en compte de l'hydrophobie des lipides et de la préservation de la cohésion de la bicouche quelque soit la température du système.

3.4.5 Les potentiels intermoléculaires type Lennard-Jones

Cette section présente les différentes expressions mathématiques possibles pour les potentiels intermoléculaires. Hormis Marrink et al. [MM04] qui modélisent les **interactions électrostatiques** par un potentiel de Coulomb écranté⁸ pour les groupes chargés (Q), les modèles n'intègrent en général que des potentiels type Lennard-Jones, c'est-à-dire des potentiels œuvrant sur des distances plus courtes que le potentiel de Coulomb.

► Potentiel indépendant du type de grain

Dans le modèle de Stevens [SHW03], les particules interagissent par un potentiel standard de Lennard-Jones 6-12 Eq.(3.4.29) avec le même minimum d'énergie et la même distance de coeur dur pour toutes les paires. La distinction des phases est obtenue en choisissant la distance de coupure des interactions **polaires/apolaires** au minimum d'énergie ne conservant ainsi que la partie répulsive. Autrement dit, toutes les interactions **queue/queue**, **tête/tête**, **tête/solvant** et **solvant/solvant** bénéficient de la même distance de coupure $r_c = 2.5\sigma$. Seules les interactions **tête/queue** et **queue/solvant** sont purement répulsives avec une distance de coupure plus courte $r_c = 2^{1/6}\sigma$.

$$U_{LJ}(r) = 4\epsilon [(\sigma/r)^{12} - (\sigma/r)^6] \quad (3.4.29)$$

► Potentiel dépendant du type de grain

Wang et Frenkel [WF05a] avec leur modèle flexible de lipide à 3 grains (types h et t) utilisent des potentiels de Lennard-Jones tronqués

$$U_{tt}(r) = \epsilon [(\frac{\sigma_{tt}}{r})^{12} - 2(\frac{\sigma_{tt}}{r})^6 + U_{c0}] \quad (3.4.30)$$

$$U_{hh}(r) = \epsilon [(\frac{\sigma_{hh}}{r})^{12} - 2(\frac{\sigma_{hh}}{r})^6 + U_{c1}] \quad (3.4.31)$$

$$U_{th}(r) = \epsilon [(\frac{\sigma_{th}}{r})^{12} - 2(\frac{\sigma_{th}}{r})^6 + U_{c1}] \quad (3.4.32)$$

Les distances de coupure pour les potentiels $U_{tt}(r)$, $U_{hh}(r)$ et $U_{th}(r)$ sont respectivement $2\sigma_{tt}$, σ_{hh} et σ_{th} ce qui restreint les interactions attractives aux grains des queues. U_{c0} et U_{c1} sont les quantités qui remplacent les potentiels réduits au delà des distances de coupure.

► Potentiel adouci

Pour Lenz et Schmid [LS05], les grains des lipides interagissent via le potentiel de Lennard-Jones adouci Eq.(3.4.33). Dans le cas d'une interaction entre grains de type queue, le potentiel a ainsi une contribution attractive avec $r_{cutoff} = 2\sigma$. Pour les interactions entre les grains de type tête ou entre les grains de type tête et de type queue, on aura un potentiel à coeur mou, c'est-à-dire juste la contribution répulsive du potentiel avec un $r_{cutoff} = \sigma$.

$$V_{LJ}^{shifted}(r) = \begin{cases} V_{LJ}(r) - V_{LJ}(r_{cutoff}) & \text{pour } r < r_{cutoff} \\ 0 & \text{pour } r \geq r_{cutoff} \end{cases} \quad (3.4.33)$$

$$\text{avec } V_{LJ}(r) = \epsilon \left(\left(\frac{\sigma}{r} \right)^{12} - 2 \left(\frac{\sigma}{r} \right)^6 \right)$$

⁸L'écrantage du champ électrique consiste en l'atténuation du champ électrique en raison de la présence de porteurs de charge électrique mobiles dans un solvant.

► Potentiel tronqué

Le potentiel tronqué de Weeks-Chandler-Andersen (potentiel WCA) est un potentiel d'interaction de courte portée et purement répulsive. Avec des lipides flexibles à trois grains, Cooke et al. [CD05] utilisent ce potentiel pour modéliser les interactions d'exclusion de volume et la répulsion de courte portée. Le potentiel WCA (Eq.(3.4.34)) est construit par troncature d'un potentiel de Lennard-Jones à la distance *cutoff* de $\sqrt[6]{\sigma}$, correspondant au minimum du puits du potentiel et par un amortissement appliqué à tout le potentiel d'une quantité équivalente à la profondeur du puits ϵ .

$$U_{WCA}(r) = \begin{cases} 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right] + \epsilon & \text{pour } r \leq \sqrt[6]{\sigma} \\ 0 & \text{pour } r > \sqrt[6]{\sigma} \end{cases} \quad (3.4.34)$$

► Potentiel étendu et indépendant du type de grain

Avec des lipides flexibles à 5 grains parmi 3 types (**tête**, **interface** et **queue** de la figure 3.4.6(b)), Brannigan et al. [BPB05] modélisent les interactions intermoléculaires par des combinaisons de potentiels par paire pris parmi trois potentiels différents. L'interaction répulsive $U_{rep}(r)$ (Eq.(3.4.35)) agit entre toutes les paires de grains et l'attraction/dispersion $U_{att}(r)$ (Eq.(3.4.36)) agit entre toutes les paires **queue/interface** et **queue/queue**. L'attraction douce (*soft*) $U_{int}(r)$ (Eq.(3.4.37)) intervient entre toutes les paires **interface/interface**. Les potentiels sont tronqués aux distances 2σ , 2σ et 3σ pour respectivement les interactions $U_{rep}(r)$, $U_{att}(r)$ et $U_{int}(r)$. Avec les unités de référence $\epsilon = 2.75kJ/mol$ et $\sigma = 0.75nm$, les paramètres utilisés sont : $k_bT = 0.9\epsilon$, $c_{rep} = 0.4\epsilon$, $c_{att} = 1\epsilon$, $c_{int} = 3\epsilon$. c_{bend} varie de 5ϵ à 10ϵ : pour $c_{bend} < 5\epsilon$, il y a apparition de pores, et pour $c_{bend} > 15\epsilon$, la bicouche est structurellement trop ordonnée.

$$U_{rep}(r) = c_{rep} \left(\frac{\sigma}{r}\right)^{12} \quad (3.4.35)$$

$$U_{att}(r) = -c_{att} \left(\frac{\sigma}{r}\right)^6 \quad (3.4.36)$$

$$U_{int}(r) = -c_{int} \left(\frac{\sigma}{r}\right)^2 \quad (3.4.37)$$

Pour Goetz et Lipowsky[GL98], les interactions répulsives entre les grains hydrophiles et hydrophobes sont modélisées par le potentiel répulsif de coeur mou (*soft-core potential*) U_{SC} (Eq.(3.4.38)) tandis que les interactions attractives sont modélisées par un potentiel de Lennard-Jones U_{LJ} (Eq.(3.4.39)). Pour minimiser le nombre de paramètres, toutes les fonctions de potentiel d'énergie ont le même rayon de coupure $r_c = 2.5\sigma$. On a $\sigma_{SC} = 1.05\sigma$ ce qui permet d'obtenir une répulsion coeur-dur du potentiel à coeur mou de même intensité que la répulsion du potentiel de Lennard-Jones.

$$U_{SC}(r) = 4\epsilon \left(\frac{\sigma_{SC}}{r}\right)^9 \quad (3.4.38)$$

$$U_{LJ}(r) = 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r}\right)^{12} - \left(\frac{\sigma_{ij}}{r}\right)^6 \right] \quad (3.4.39)$$

► Potentiel étendu et dépendant du type de grain

Avec des lipides rigides à trois grains, Farago [Far03] propose de prendre en compte les interactions hydrophobes grâce à un ensemble de potentiels uniquement par paire (Fig.3.4.12).

Les grains sont numérotés 1, 2, 3 respectivement pour le grain de type **tête**, le grain de type **queue** intermédiaire et le grain de type **queue** terminale (Fig.3.4.6(a)).

Ainsi les potentiels répulsifs U_{12} Eq.(3.4.40) et U_{13} Eq.(3.4.41) représentent les interactions entre un grain hydrophile de type **tête** et un grain hydrophobe de type **queue**. Les potentiels attractifs de type Lennard-Jones U_{ii} Eq.(3.4.42) décrivent les interactions entre deux grains de même type. Enfin le potentiel attractif U_{23} Eq.(3.4.43) décrit l'interaction entre les grains 2 et 3, tous deux hydrophobes. Tous ces potentiels ont la même distance de coupure $r_c = 2.5\sigma$.

$$U_{12}^{LJ}(r) = 4\epsilon_{12}\left(\frac{\sigma_{12}}{r}\right)^{12} \quad (3.4.40)$$

$$U_{13}^{LJ}(r) = 4\epsilon_{13}\left(\frac{\sigma_{13}}{r}\right)^{18} \quad (3.4.41)$$

$$U_{ii}^{LJ}(r) = 4\epsilon_{ii}\left[\left(\frac{\sigma_{ii}}{r}\right)^{12} - \left(\frac{\sigma_{ii}}{r}\right)^6\right] \quad (3.4.42)$$

$$U_{23}^{LJ}(r) = 4\epsilon_{23}\left[\left(\frac{\sigma_{23}}{r}\right)^2 - \left(\frac{\sigma_{23}}{r}\right)\right] \quad (3.4.43)$$

Le tableau 3.4.2 donne les valeurs des différents paramètres de ces potentiels avec l'unité référence de longueur $\sigma = 1nm$ et l'unité d'énergie kT .

σ_{ij}/σ	tête (1)	queue (2)	queue (3)	ϵ_{ij}/kT	(1)	(2)	(3)
(1)	1.1	1.15	1.4		0.1875	1.1375	200
(2)		1.05	0.525			1.75	375
(3)			1				1.875

TAB. 3.4.2 – Jeu de paramètres pour les potentiels utilisés par Farago [Far03].

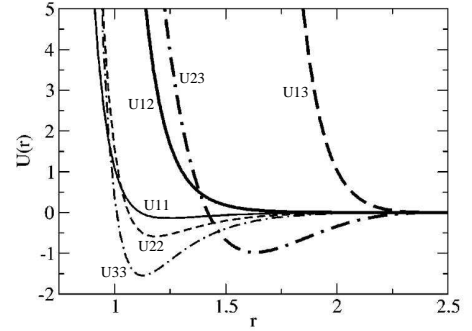


FIG. 3.4.12 – Farago utilise simultanément différents potentiels.

► Potentiel discrétisé

Pour Marrink et al. [MM04], les interactions non liantes entre les sites sont décrites par un potentiel de Lennard-Jones 6-12 discrétisé en 5 niveaux pour simplifier le modèle et dont l'intensité déterminée par le paramètre ϵ_{ij} (cf Eq.(3.4.39)) prend sa valeur parmi celles présentées en Tab.3.4.3 : interaction attractive (I, $\epsilon_{ij} = 5kJ/mol$), semi-attractive (II, $\epsilon_{ij} = 4.2kJ/mol$), intermédiaire (III, $\epsilon_{ij} = 3.4kJ/mol$), semi-répulsive (IV, $\epsilon_{ij} = 2.6kJ/mol$), et répulsive (V, $\epsilon_{ij} = 1.8kJ/mol$). Le niveau I modélise les interactions fortes polaires du solvant, le niveau III modélise les interactions non polaires des chaînes aliphatiques (les acides gras) et le niveau V modélise la répulsion hydrophobe entre les phases polaires et non polaires. Pour toutes ces types d'interactions, la même longueur effective est prise $\sigma_{ij} = 0.47nm$. Par ailleurs le cutoff $r_c = 1.2nm$ correspond approximativement à 2.5σ . Les paramètres sont calibrés en utilisant des systèmes alcane-eau pour reproduire les densités expérimentales, les solubilités mutuelles et les taux de diffusion relative.

type	sous type	P		N			C		Q		
		O	d	a	da	O	d	a	da		
P		I	IV	III	III	II	V	I	I	I	I
N	O		III	III	III	III	III	III	III	III	III
	d			II	II	II	IV	III	III	II	II
	a				II	II	IV	III	II	III	II
	da					I	V	III	II	II	I
C							III	V	V	V	V
Q	O							III	III	III	II
	d								III	II	I
	a									III	I
	da										I

TAB. 3.4.3 – Une discrétisation de l'intensité d'un potentiel intermoléculaire type Lennard-Jones (d'après Marrink et al. [MM04]).

► Potentiels à coeur mou

Les modèles suivants ont la particularité d'utiliser la la dynamique dissipative des particules (DPD, voir section 4.5.1) pour décrire l'évolution du système ce qui leur permet notamment d'utiliser des **potentiels extrêmement adoucis** (*ultra soft-core potential*) par rapport à ce qu'il a été présenté jusqu'à présent.

Pour décrire les interactions intermoléculaires, Venturoli et al. [VS99] utilisent une force dite "conservative"⁹ (Eq.(3.4.44)) dérivant d'un potentiel à coeur mou initialement introduit par Groot et Warren [GW97] :

$$\overrightarrow{F}_{ij}^C(r_{ij}) = \begin{cases} a_{ij}(1 - r_{ij}/r_c)r_{ij} & \text{pour } r_{ij} < r_c \\ 0 & \text{pour } r_{ij} \geq r_c \end{cases} \quad (3.4.44)$$

avec $\overrightarrow{r}_{ij} = \overrightarrow{r}_j - \overrightarrow{r}_i$ soit $r_{ij} = |\overrightarrow{r}_{ij}|$ la distance entre deux grains i et j et $\hat{r}_{ij} = \overrightarrow{r}_{ij}/r_{ij}$ le vecteur unitaire.

L'étude des différents modèles disponibles dans la littérature (notamment ceux dont nous avons déjà parlé dans les sections précédentes : Shillcock et al.[SL02], Yamamoto et al.[YMH02] ou encore Laradji et al.[LSK05]) montrent une grande variété de valeurs pour les paramètres utilisés pour caractériser les potentiels à coeur mou. Tous ces modèles permettent de réaliser des simulations cohérentes avec les données expérimentales. Etant donné que notre modèle de référence est celui d'O.Farago, nous ne détaillons pas ces différents modèles et laissons le lecteur se rapporter aux articles déjà cités.

3.5 Les méthodes de calibrage

Différentes méthodes complémentaires les unes des autres sont envisageables pour calibrer un modèle. Dans notre contexte, l'exercice consiste essentiellement à définir des valeurs pour les paramètres d'interactions intramoléculaires (dans le cas de molécules flexibles) et d'interactions intermoléculaires.

La validation des résultats des simulations passe par la confrontation avec les données expérimentales mais peut également être réalisée en comparant ces résultats avec les données générées par les simulations de dynamique moléculaire. Pour réaliser le paramétrage des simulations, il est possible d'utiliser des compositions homogènes ou non de molécules. Par exemple Marrink et al. [MM04] utilisent des mélanges alcanes/eau pour retrouver des propriétés cohérentes de densité, de compressibilité et de diffusion, puis intègrent la présence d'ions sodium et chlorure pour régler les interactions entre ions pour enfin passer à des systèmes de phospholipides, en l'occurrence des molécules de DPPC.

3.5.1 Validation par données macroscopiques

Noguchi et Takasu [NT01] ont validé leur modèle de lipide rigide à 3 grains en vérifiant qu'en fonction de la température considérée les molécules s'assemblent en micelles ou en vésicules. En diminuant la température, les structures formées varient d'une myriade de petites micelles à de plus grosses micelles, puis à des micelles reliées entre elles pour finalement aboutir à des vésicules fluides.

Sintes and Baumgärtner [SB97, SB98b] ont validé leur modèle à solvant implicite pour les bicouches lipidiques pures en reproduisant les valeurs d'aire moyenne par lipide (l'utilisation de ce paramètre sera détaillée dans le chapitre 5) et d'inclinaison moyenne (paramètre d'ordre) obtenues expérimentalement. Les mêmes auteurs ont également étendu ce modèle pour étudier l'attraction

⁹par rapport aux deux autres forces incluses dans la DPD : $\overrightarrow{F}_{ij}^R$ la force aléatoire du mouvement brownien et $\overrightarrow{F}_{ij}^D$ la force dissipatrice liée à la friction du grain dans le solvant. cf section 4.5.1 pour une description plus complète de la DPD.

propagée des lipides entre des protéines transmembranaires de forme cylindrique [SB98b] et des protéines de forme conique tronquée [SB98a].

Farago [Far03] a validé son modèle de lipide rigide trimérique à solvant implicite en explorant par des simulations de Monte Carlo le diagramme de phase des bicouches lipidiques en fonction de l'aire projetée par lipide. En augmentant l'aire projetée il a été observé que la membrane modèle passe par une transition de phase gel/fluide. Dans la phase gel, les lipides sont immobiles et s'organisent sous forme d'hexagone. Dans la phase liquide, les lipides sont libres de diffuser dans le plan de la membrane. Par ailleurs, la propriété d'élasticité du système a été caractérisée en calculant la tension de surface et le mode de courbure à partir du spectre des ondulations thermiques de la phase fluide. La valeur des modes de courbure trouvée en utilisant le modèle s'avère être du même ordre que les valeurs expérimentales pour les bicouches pures lipidiques [LS95].

3.5.2 Validation par données microscopiques

Flekkoy et Coveney [FC99] ont proposé le premier travail pour lier la dynamique dissipative des particules (DPD, voir section 4.5.1) à la dynamique microscopique sous-jacente en introduisant une procédure pour **dériver une DPD** pour un modèle à grains à partir des simulations atomistiques. Par exemple, la méthode de paramétrage de Groot et Rabone [GR01] requiert uniquement les compressibilités et solubilités correctes des différents composés entre eux.

Izvekov and Voth [IV05] ont utilisé une **approche dite "multi-échelles"** (*multiscale coarse-graining* ou MS-CG) pour **dériver systématiquement les paramètres** des potentiels mésoscopiques à partir des interactions atomistiques. Pour ce faire, une méthode de mise en correspondance des forces (*force-match method* ou *FM method*) permet de déterminer les forces effectives par paire à partir des trajectoire et des forces des dynamiques atomistiques. On détermine les **potentiels de force moyenne** (*Potential of Mean Force* ou *PMFs*) entre les grains. Cette approche est une extension de la méthode originelle des moindres carrés suggérée par Ercolessi et Adams [EA94]. Le champ de force est réglé en utilisant une **interpolation par spline**¹⁰ des forces calculées sur les atomes regroupés par grain. Dans ce cas, les fonctions de distribution radiale et les profils de densité dans une bicouche de DMPC ne servent plus à calibrer le modèle à grains, mais permettent une évaluation de celui-ci.

3.5.3 Validation par données croisées

Il est tout à fait possible d'ajouter aux données macroscopiques expérimentales des données issues de simulations atomistiques, on parlera alors de validation par données croisées.

Par exemple Klein et al. [NSK04] combinent des données expérimentales (densité, tension de surface) et les simulations *tout-atome* (RDFs) d'une bicouche équilibrée de DMPC dans la phase L_α pour construire leur modèle à grains (Fig 3.4.7(b)). Cette procédure est généralement coûteuse en temps de calcul mais il existe différentes stratégies d'optimisation. La méthode la plus simple mais également la plus lente est **la procédure d'optimisation simplexe automatisée** (ou *automated simplex optimization procedure*) introduite par Müller-Plathe et al. [FSBMP99]. Une limitation est que cette procédure automatique se déroule à "*l'aveuglette*" et peut donner lieu à des résultats physiquement irréalistes dus notamment aux effets de compensation (effet de corrélation entre les paramètres). Le paquet APPSPACK du Sandia National Laboratory¹¹ propose une stratégie d'optimisation sophistiquée. C'est une méthode de recherche de structure asynchrone, tolérante aux fautes et parallèle qui a été spécifiquement conçue pour des problèmes nécessitant des évaluations longues des fonctions (typiquement des simulations prenant plusieurs heures).

Il est également possible d'utiliser le **potentiel de force moyenne** obtenu par l'inversion de Boltzmann des RDFs cibles, et d'itérer suivant (3.5.1).

$$V_{n+1}(r) = V_n(r) + kT \ln \frac{RDF_n(r)}{RDF_{target}(r)} \quad (3.5.1)$$

¹⁰Dans le domaine de l'analyse numérique, une spline est une fonction définie par morceaux par des polynômes.

¹¹<http://software.sandia.gov/appspack/>

On peut enfin utiliser une **méthode d'optimisation basée sur le gradient** proposée par Lyubartsev et Laaksonen [Lyu95] qui peut être utilisée pour calculer la matrice complète des premières dérivées partielles et relier alors les paramètres d'entrée aux variations des observables à partir d'une seule simulation. Lyubartsev [Lyu05] a appliqué cette méthode pour construire un modèle à solvant implicite pour les lipides et les bicouches lipidiques. Des auto-assemblages spontanés de ces lipides en bicouches et vésicules sont obtenus avec des profils de densité en accord avec ceux calculés à partir de simulations atomistiques. Ce résultat est intéressant puisque les potentiels d'interactions sont issus d'un système de lipides dilués dans de l'eau, et non d'une bicouche pleinement formée.

3.6 Vers un premier modèle à grains

En résumé, de toutes les informations que nous venons de donner sur les modèles à grains, nous pouvons retenir qu'en nous plaçant à l'échelle de l'atome, nous aurions à tenir compte de toutes les molécules, y compris les nombreuses molécules d'eau constituant le solvant.

Un modèle à grains d'atomes permet de moyenner une portion de molécule en représentant cette portion par un ensemble d'atomes, le grain, et en ne retenant que son centre de gravité pour conditionner le comportement du grain. Ce compromis entre perte de degrés de liberté dans la représentation des atomes et gain en efficacité permet de simuler des temps et des volumes (ou des aires) en accord avec ceux requis pour l'observation d'une membrane, ceci en conservant un minimum d'informations structurelles des molécules.

Toutefois, la revue des différents modèles à grains pour les lipides en général, et plus particulièrement les membranes, attire l'attention sur un ensemble de points qu'il est important de prendre en compte :

- comment modéliser un lipide : quels types ? combien de grains par lipide ? quel modèle pour la flexibilité moléculaire ?
- comment prendre en compte le solvant ? explicitement ou implicitement ?
- comment faire interagir les molécules ? quels potentiels utiliser ?
- comment calibrer le modèle, quelles sont les sources de données disponibles ?

Dans le cadre de cette étude, nous avons décidé de concevoir un modèle inspiré du modèle de Farago. Ce modèle appelé MitoMAS pour *Mitochondria Multi-Agents System* fait l'objet du chapitre qui suit.

Chapitre 4

MitoMAS

Ce chapitre a pour objectif de présenter nos choix de modélisation pour MitoMAS, le modèle à grains conçu pour simuler la membrane interne mitochondriale. Les sections suivantes détaillent la structure de notre lipide *modèle* et les forces qui lui sont appliquées. Nous verrons ensuite qu'il est possible d'étendre ce modèle à d'autres objets biologiques comme les protéines intramembranaires.

4.1 Définition d'un lipide modèle

Avant de donner les détails de notre modèle, nous allons décrire la structure lipidique que nous avons choisi de représenter pour modéliser la membrane interne mitochondriale.

4.1.1 Intégration d'une molécule de DMPC

Une molécule de dimyristoyl-phosphatidylcholine (DMPC) se trouve non seulement être très souvent modélisée dans la littérature mais aussi le lipide le plus représenté dans la membrane interne mitochondriale. C'est cette structure qui nous a servi d'exemple pour développer notre modèle *gros grains* de lipide. Nous aborderons plus loin dans ce mémoire les possibilités d'extension aux autres types de lipides ou molécules.

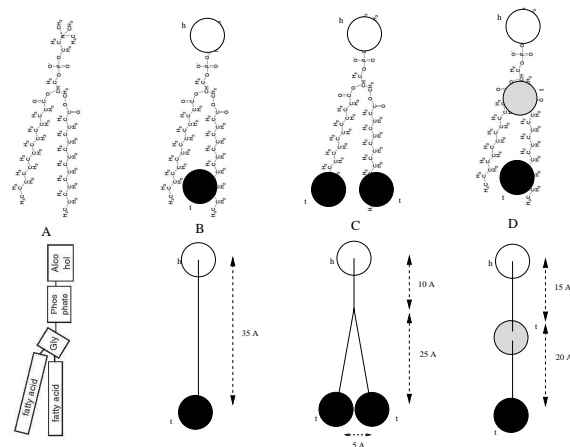


FIG. 4.1.1 – Différents modèle à grains pour la dimyristoyl-phosphatidylcholine DMPC. A : représentation schématique, B : modèle 2 types de grain (1 tête et 1 queue), C : modèle 2 types de grain (1 tête et 2 queues), D : modèle 3 types de grain (tête, queue et interface).

La figure Fig.4.1.1 montre la structure atomique d'une molécule de DMPC et sa représentation schématique annotée (partie A), et un ensemble de solutions possibles pour représenter cette molécule à partir de deux grains (partie B) ou trois grains ou types de grain (parties C et D). Le schéma C montre qu'on peut choisir d'ajouter une seconde queue en extrémité de molécule, représentant ainsi le fait qu'un lipide est constitué de deux acides gras. Cette distinction permet notamment, si l'on opte pour un modèle flexible de molécule, d'associer une mobilité plus grande pour le grain de l'acide gras insaturé (présence d'une double liaison, c'est-à-dire d'un coude dans la chaîne) ce qui est cohérent avec les données biologiques. Néanmoins le vide laissé entre la tête et la ou les queues (parties B et C) risquent d'aboutir à des problèmes de superposition des volumes peu réaliste. En choisissant d'aligner les 3 grains (partie D), on obtient une structure trimérique capable de mieux remplir le volume moléculaire. Ce modèle correspond au modèle d'O.Farago, c'est également celui que nous avons retenu.

4.1.2 Choix de la granularité

Dans le chapitre 3 (section 3.4.1) nous avons exposé différents modèles à grains utilisés pour représenter les lipides. Rappelons quelques éléments déterminants quant au choix du type et du nombre de grains.

En ce qui concerne le nombre de types de grain :

- **2 types** de grain correspondent au strict minimum pour prendre en compte des molécules amphiphiles, en distinguant un type **tête** hydrophile et un type **queue** hydrophobe ;
- **3 types** de grain permettent de définir des sites privilégiés d'interaction des molécules, essentiellement au centre de la structure, afin d'augmenter la cohésion de la membrane même en phase fluide caractérisée par l'agitation des lipides ;
- **plus de 3 types** pour un seul et même lipide permettent de différencier plus précisément les portions d'un lipide. Toutefois cette solution a comme inconvénient majeur que le calibrage du modèle devient très complexe.

Nous retenons donc qu'un ensemble de 2 types suffit à prendre en compte l'hydrophobie des lipides. Pour des raisons de cohésion de la membrane, nous avons choisi de conserver la distinction supplémentaire proposée par Farago entre le type **queue** à l'extrémité du lipide et le type **queue** en position intermédiaire. Bien qu'en terme d'hydrophobie, ces deux sous-types se comportent de manière similaire puisqu'ils définissent des types queues hydrophobes, il sera plus commode de parler de 3 types de grain (**tête**, **interface** et **queue**) pour un lipide pour mieux différencier les comportements liés à leur position. Dans MitoMAS on aura donc un modèle de lipide à partir de 3 types de grain.

A partir de ces 3 types, il est assez naturel de considérer qu'au moins un grain par type est nécessaire pour modéliser notre phospholipide.

Dans la figure 4.1.2 les grains sont numérotés 1, 2, 3 respectivement pour le grain bleu de type **tête**, le grain violet de type **interface** et le grain rouge de type **queue**. Rappelons qu'en utilisant 3 grains par lipide, nous obtenons une modélisation correcte de son encombrement stérique et aussi une certaine flexibilité à la molécule puisque ces 3 grains définissent un angle. L'orientation de la molécule est elle aussi correctement restituée ce qui confère à la membrane des propriétés importantes telles que la rigidité ou un rayon de courbure. La figure 4.1.2 montre une molécule de dimyristoyl-phosphatidylcholine (DMPC) modélisée de cette manière par un trimère.

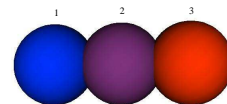


FIG. 4.1.2 – Un phospholipide à 3 grains pris parmi 3 types pour former un trimère.

4.1.3 Extension du modèle à 3 grains

A partir de notre modèle à 3 types de grain, nous pouvons modifier la finesse de représentation d'un lipide. D'une manière générale, le modèle d'une molécule doit rester suffisamment ouvert pour contenir un nombre n de grains. Ainsi, dans l'optique de tester des modèles différents comme ceux de Brannigan (Fig.4.1.3(a)) ou encore de Groot et Rabone (Fig.4.1.3(b)), nous avons laissé la possibilité au modélisateur de modifier le nombre de grains utilisés par type. Par exemple quelques essais réalisés avec des lipides à 2 queues ont montré des modifications sensibles des repliements de la membrane. Nous avons également prévu que notre modèle puisse être étendu à la simulation de la présence de complexes enzymatiques dans la membrane. Dans ce cas, il serait possible de représenter un complexe par une enveloppe de grains respectant les portions hydrophobes et hydrophiles des protéines membranaires. Cela conduit à un accroissement du nombre de grains par type utilisés pour décrire une molécule comme le montre la figure 4.1.3(c).

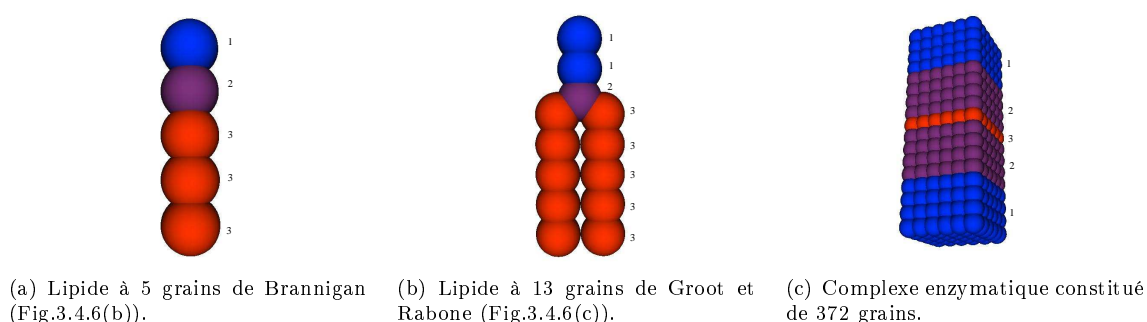


FIG. 4.1.3 – Différentes molécules assemblées avec 3 types de grain.

4.2 Les interactions entre grains d'atomes

Pour la modélisation des interactions entre les grains d'atomes, nous avons le choix entre différentes solutions : modèle des sphères dures, potentiels multi-corps, potentiels par paire. Comme nous l'avons évoqué au chapitre 3, les deux premières solutions ne sont pas optimales d'un point de vue calculatoire.

En effet, dans le premier cas, comme l'interpénétration des volumes de deux sphères dures est interdite, cela implique de détecter les conflits et de proposer des solutions en terme de nouvelles positions. C'est pourquoi on préférera, comme le propose les deux autres méthodes, traiter des positions ponctuelles dans l'espace, plutôt que des sphères (volumes) en prenant soin d'anticiper le cas, peu probable, de deux positions de l'espace exactement superposées.

L'utilisation des potentiels multi-corps nécessite le calcul pour chaque grain de son voisinage sous la forme d'une densité locale des autres grains. Dans le cas de la membrane qui est un environnement extrêmement dense, ce calcul est trop coûteux.

Nous nous sommes donc placés dans un cadre d'**interactions deux à deux** (*pairwise potential*) entre les centres de gravités des grains situés spatialement.

Nous allons maintenant décrire les forces qui conditionnent les interactions intra- et inter-moléculaires dans ce modèle.

4.2.1 Les interactions intramoléculaires

Pour ce qui est de la dynamique interne des molécules, nous avons vu en section 3.4.2 que les modèles à grains disponibles dans la littérature utilisent des **potentiels de liaison** entre deux grains ou des **potentiels de flexion** entre deux liaisons pour modéliser une certaine flexibilité des queues ou encore des structures spécifiques pour les têtes de certains phospholipides. La figure 4.2.1

montre différentes possibilités procurant plus ou moins de flexibilité à une structure générale de molécule représentée par 5 grains (A) :

- une représentation **flexible** (B), la plus répandue, avec des **potentiels de liaison** et des **potentiels de flexion**, ce qui permet de paramétrer la dimension et la flexibilité de chaque section de la structure. A noter que le voisinage intramoléculaire de chaque grain reste le même au cours de la simulation.
- une représentation **flexible** (C) assurée uniquement par des **potentiels de liaison** avec éventuellement une distance limite au delà de laquelle les grains trop éloignés n’interagissent plus (D). Cette solution permet par exemple de maintenir une conformation donnée pour un site actif constitué de grains éloignés dans la séquence de la protéine, et ce, sans qu’il n’y ait besoin de contraindre toute la structure intermédiaire. Dans ce cas, le voisinage intramoléculaire de chaque grain peut évoluer au cours de la simulation, en fonction de la conformation de la molécule.
- une représentation **rigide** (E) qui impose des positions relatives fixes aux grains. Il n’est pas question ici de voisinages intramoléculaires puisqu’il n’y aura aucun potentiel entre les grains de la même molécule.

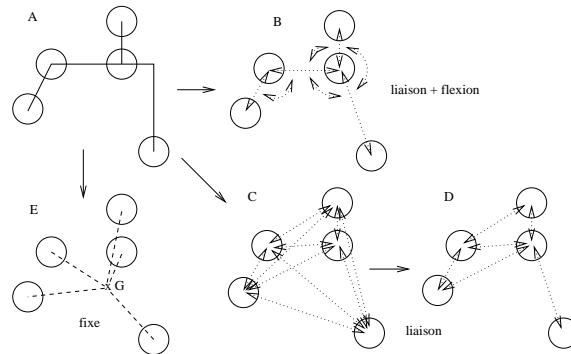


FIG. 4.2.1 – Plusieurs modèles, plus ou moins flexibles, d’une même molécule.

Le modèle de MitoMAS permet d’implémenter toutes ces situations à partir d’un modèle générique d’interactions intra- et inter-moléculaires. Nous donnerons tous les détails des mécanismes qui permettent ceci dans la section 4.7.3 lorsque nous expliquerons les différents modèles de MitoMAS.

La prise en compte des différents types de dynamique interne nécessite qu’à partir du modèle générique chaque modèle d’interactions intramoléculaires définisse les comportements à spécialiser en fonction de l’approche retenue. Par exemple, un modèle de type (B) donnera la possibilité au modélisateur de préciser les expressions mathématiques pour les potentiels de liaison et de flexion (cf section 3.4.2) œuvrant entre un grain et son voisinage intramoléculaire. Il lui faudra également préciser les valeurs des paramètres utilisés pour ces potentiels.

Dans un premier temps, nous avons privilégié le **modèle rigide**, notamment pour simplifier le calibrage et augmenter l’efficacité des calculs. En effet, lorsqu’on utilise un modèle flexible, les constantes de raideur des potentiels intramoléculaires induisent des petits déplacements des grains trop nombreux. Ces “*vibrations*” autour des positions d’équilibre réduisent sensiblement les performances de la simulation. Dans le modèle rigide, la rigidité du lipide se traduit par des **positions relatives fixes des grains** tout au long de la simulation et le moteur physique implémente une **dynamique des corps rigides**. Les forces qui opèrent sur les grains sont cumulées au niveau du centre de gravité G de la molécule pour générer le déplacement linéaire de l’ensemble de la structure, et par conséquent des grains. Par ailleurs, ces mêmes forces appliquées à des distances non nulles de G induisent des couples à l’origine des mouvements de rotation¹.

¹le moteur physique implémenté pour l’occasion utilise une extension non commutative des nombres complexes pour gérer les rotations et compositions de rotation dans l’espace 3D : les quaternions. Un quaternion H s’écrira $H = a + bi + cj + dk$ avec $a, b, c, d \in \mathbb{R}$ et $i, j, k \in \mathbb{C}$. Voir en annexes pour plus de détails.

Dans tous les cas, l'ensemble des interactions sont issues de potentiels par paire ne concernant que les grains. En d'autres termes, il n'y a pas d'interaction directe entre les molécules. Cette caractéristique est importante puisqu'elle permet à plusieurs modèles d'interactions intramoléculaires de se "côtoyer" dans la même simulation.

Dans le futur, nous espérons pouvoir utiliser cette capacité de notre modèle à faire se côtoyer les types d'interactions pour la réalisation de simulations hybrides incorporant des lipides flexibles et rigides. Ceci devrait nous permettre de mieux étudier l'influence de cette caractéristique notamment sur les replis de la membrane.

Partant sur un modèle de lipide par un trimère rigide, il reste à définir les interactions entre les molécules.

4.2.2 Les interactions intermoléculaires

La section 3.4.5 sur les différents potentiels intermoléculaires utilisés dans les modèles à grains pour les membranes biologiques nous a permis de mettre en lumière un certain nombre de points importants concernant les propriétés des potentiels à adopter.

► La forme des potentiels

A priori au moins deux comportements doivent être conservés :

- une **répulsion** pour rendre compte l'encombrement stérique voire éventuellement la conformation d'un complexe enzymatique ;
- une **attraction** pour assurer un minimum de cohésion aux structures émergentes.

Typiquement, ces deux comportements se retrouvent dans le potentiel de Lennard-Jones employé en dynamique moléculaire.

Avant de développer les potentiels type Lennard-Jones que nous avons utilisés, notons les observations suivantes :

- Les intensités des forces sont directement à mettre en relation avec les déplacements des grains. Autrement dit, plus les intensités sont fortes, plus les déplacements sont grands mais aussi plus les erreurs d'intégration sont grandes. Une diminution du pas de temps d'intégration résout éventuellement ce problème, mais en contre partie réduit le temps simulé final. Une autre solution, moins évidente, consiste simplement à **amoindrir l'intensité des forces** mais on s'écarte alors des champs de forces usuels de la dynamique moléculaire.
- Les interactions entre particules sont cumulatives, autrement dit et cela semble bien naturel, une particule subira l'influence simultanément de plusieurs autres particules si celles-ci se trouvent être suffisamment proches. La notion de proximité est associée à la **distance de coupure**, c'est-à-dire le rayon de la sphère d'interaction centrée sur la particule. En réduisant cette sphère d'interaction, on abaisse l'intensité de la force moyenne appliquée à un grain ce qui permet d'augmenter le temps d'intégration.

Comme nous allons le voir, notre solution a consisté essentiellement à sélectionner des distances de coupure aussi faibles que possible sans compromettre la cohésion de la membrane.

► Plusieurs potentiels type Lennard-Jones

En partant de la forme standard d'un potentiel $V(r)$ de Lennard-Jones 12-6 déjà introduite en Eq.(3.3.7), nous avons implémenté un ensemble de potentiels correspondant aux différents cas d'interactions.

Les potentiels répulsifs U_{12} Eq.(4.2.1) et U_{13} Eq.(4.2.2) représentent respectivement les interactions **tête/interface** (T/I) et **tête/queue** (T/Q). Les potentiels attractifs/répulsifs de type Lennard-Jones U_{ii} Eq.(4.2.3) décrivent les interactions entre deux grains de même type (T/T, I/I et Q/Q). Enfin le potentiel attractif/répulsif (à coeur mou) U_{23} Eq.(4.2.4) décrit l'interaction

interface/queue (I/Q). Pour rappel, on choisit l'unité référence de longueur $\sigma = 1nm$ soit le diamètre d'un grain.

$$\text{T/I : } U_{12}^{LJ}(r) = 4\epsilon_{12}\left(\frac{\sigma_{12}}{r}\right)^{12} \quad (4.2.1)$$

$$\text{T/Q : } U_{13}^{LJ}(r) = 4\epsilon_{13}\left(\frac{\sigma_{13}}{r}\right)^{18} \quad (4.2.2)$$

$$\text{T/T, I/I et Q/Q : } U_{ii}^{LJ}(r) = 4\epsilon_{ii}\left[\left(\frac{\sigma_{ii}}{r}\right)^{12} - \left(\frac{\sigma_{ii}}{r}\right)^6\right] \quad (4.2.3)$$

$$\text{I/Q : } U_{23}^{LJ}(r) = 4\epsilon_{23}\left[\left(\frac{\sigma_{23}}{r}\right)^2 - \left(\frac{\sigma_{23}}{r}\right)\right] \quad (4.2.4)$$

Le tableau 4.2.1(a) donne la distance à l'équilibre σ_{ij} entre un grain de type i et un grain de type j . Cette distance est rapportée à l'unité référence de longueur $\sigma = 10\text{\AA}$. Le tableau 4.2.1(b) donne la constante d'énergie ϵ_{ij} du potentiel appliqué entre un grain de type i et un grain de type j . Cette intensité est rapportée à l'unité énergétique kT où k est la constante de Boltzmann et T la température du système.

(a) Les distances à l'équilibre.				(b) L'intensité des potentiels.			
σ_{ij}/σ	tête (1)	interface (2)	queue (3)	ϵ_{ij}/kT	tête (1)	interface (2)	queue (3)
tête (1)	1.1	1.15	1.4	tête (1)	0.1875	1.1375	200
interface (2)		1.05	0.525	interface (2)		1.75	375
queue (3)			1	queue (3)			1.875

TAB. 4.2.1 – Jeu de paramètres pour les potentiels intermoléculaires.

La figure 4.2.2(b) montre les différentes courbes obtenues lors du calcul des différents potentiels avec ce jeu de paramètres. Tous ces potentiels ont la même distance de coupure $r_c = 2.5\sigma$. Dans le chapitre 5 nous montrerons un ensemble de résultats autour du choix de ce paramètre de coupure. Les courbes comparées montrent que la distance de répulsion la plus grande est entre les têtes et les queues (typiquement U_{13}) et que les puits de potentiel sont d'autant plus prononcés que les interactions concernent les types hydrophobes (par exemple U_{33} et U_{23}).

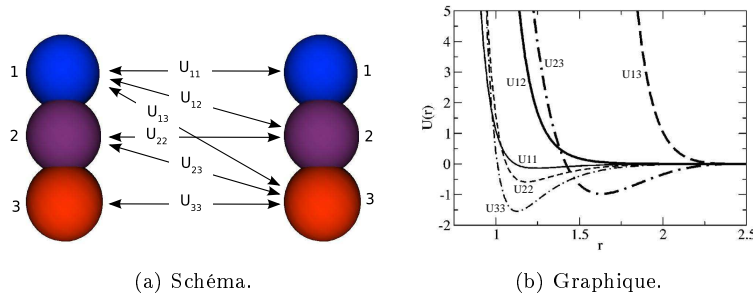


FIG. 4.2.2 – Différents potentiels type Lennard-Jones plus ou moins adoucis.

Remarque : nous choisissons de ne pas prendre en compte les interactions électrostatiques car les phospholipides composant la membrane interne mitochondriale ne sont pas chargés.

► Choix des paramètres de répulsion/attraction

Parmi les paramètres ayant un impact sur les membranes, nous pouvons jouer sur les paramètres internes des lipides, tels que la longueur ou le nombre de grain, mais également les paramètres externes notamment ceux qui sont liés aux potentiels intermoléculaires. Concernant ces potentiels de type Lennard-Jones et œuvrant entre les molécules, nous pouvons sélectionner les paramètres à explorer par ordre décroissant de leur influence sur l'évolution de la dynamique du système :

- σ_c , la distance de coupure, qui est à mettre en relation avec la "pression" exercée sur un lipide mais aussi la vitesse d'exécution de la simulation ;

- ϵ_{ij} et σ_{ij} , les deux paramètres du potentiel de LJ entre les types de grain i et j , qui sont respectivement à mettre en relation avec l'amplitude de la force résultante et avec la distance à l'équilibre;
- les exposants du potentiel de LJ, dont le choix est à mettre en relation avec la sensibilité à la distance du potentiel entre deux grains.

Pour illustrer ce dernier point, la figure 4.2.3 montre l'impact des deux exposants utilisés dans un potentiel type LJ U_{ii} , en traçant, avec les paramètres du tableau 4.2.1, les courbes de la force conservative Eq.(4.2.5) pour différentes valeurs du paramètre a reliant les exposants entre eux.

$$V(r, a) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{2a} - \left(\frac{\sigma}{r} \right)^a \right] \Rightarrow |\vec{F}^{C_{inter}}(r, a)| = 4\epsilon \left(2a \frac{\sigma^{2a}}{r^{2a+1}} - a \frac{\sigma^a}{r^{a+1}} \right) \quad (4.2.5)$$

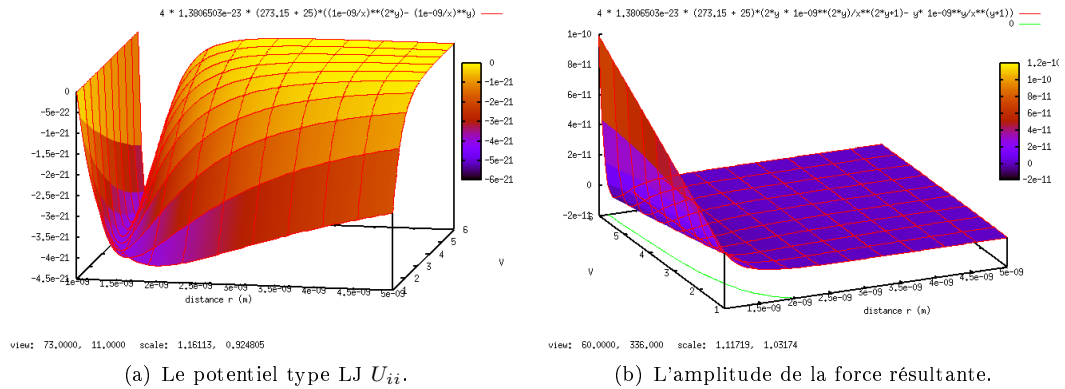


FIG. 4.2.3 – Évolution de l'interaction entre deux grains de même type en fonction de la distance r et des exposants du potentiel.

Le graphique 4.2.3(b) explicite le comportement de deux grains en interaction. A partir du moment où ces deux grains entre en interactions, ils s'attirent (zone violette, l'intensité négative de la force est synonyme d'attraction). Lorsque la distance qui les sépare devient trop courte, les forces s'inversent et les deux grains se repoussent (zone rouge, l'intensité positive de la force est ici synonyme de répulsion). La courbe verte de la figure 4.2.3(b) correspond aux points de changement de signe de la force. Autrement dit, cette courbe verte donne, en fonction des exposants, l'évolution du point d'équilibre en terme de distance. On peut voir que plus l'exposant est grand, plus la distance d'équilibre est petite et plus le "pic" positif est prononcé. D'après ce graphique, on observe qu'une distance de coupure de $\sigma = 2.5nm$ se révèle suffisamment grande pour contenir toutes les variations du potentiel et il s'avère possible de prendre des valeurs plus réduites $\sigma \sim 1nm$.

4.3 Un solvant implicite grâce à un potentiel par paire

Nous avons choisi de modéliser implicitement le solvant, c'est-à-dire qu'il n'y a **pas de grain d'eau** regroupant des molécules H_2O . Ce choix est le résultat de plusieurs observations développées plus longuement dans la section 3.4.4 :

- une représentation explicite est trop coûteuse en temps de calcul vu la proportion de solvant dans le volume de simulation ;
- une représentation implicite par contraintes externes, comme par exemple des plans de confinement, ne permet pas à la membrane de former des replis (ce qui est directement un des objectifs de notre simulation) ;
- une représentation implicite par potentiels multi-corps pénalise les performances de la simulation puisqu'il faut définir les densités locales pour chacun des grains.

L'option choisie par O.Farago[Far03] semble correspondre à une modélisation optimale du solvant : les interactions hydrophobes sont implicitement contenues dans des potentiels par paire.

4.4 Modélisation de l’environnement

4.4.1 L’espace

Toutes les molécules sont **libres** de se déplacer dans un **volume 3D** continu torique.

► Un tore

Pour modéliser notre membrane, nous avons besoin de simuler un monde clos, ceci afin d’éviter la modélisation de frontières. Comme nous l’avons vu dans la section consacrée à la gestion de l’environnement pour les SMA, le tore est le plus souvent utilisé pour modéliser un espace clos dans lequel évoluent les agents d’un SMA. Nous avons également retenu cette modélisation pour notre membrane. Ceci nous permettra d’éviter de gérer les biais dus à la présence de frontières irréalistes aux bornes de notre volume.

► Une grille

Les potentiels intermoléculaires permettent l’utilisation d’une distance de coupure (noté r_c) au delà de laquelle les contributions des forces sont considérées comme négligeables. La valeur de cette borne supérieure des distances d’interaction sera corrélée au choix de la taille des cellules dans le partitionnement de l’espace.

Comme évoqué en section 2.3.3, le choix le plus réaliste dans notre contexte est un partitionnement de l’espace sous forme de **grille**, autrement dit un découpage régulier, fixe et non hiérarchique du volume. En effet, dans le cas des membranes biologiques, le système modélisé correspond à une répartition très régulière, homogène et structurée des molécules dans l’espace. De plus, tous les éléments du système évoluent simultanément et il faut donc disposer d’une structuration de l’espace efficace en terme de mise à jour du contenu des cellules.

Avant de calculer les distances entre les grains, ce pré-filtre exclut du voisinage d’un grain ceux qui s’en trouvent trop éloignés, ce qui permet de **réduire sensiblement le nombre de calculs** effectués pour les distances. Les dimensions des voxels sont les plus petites possibles tout en restant plus grandes que la distance de coupure.

Nous verrons par la suite que ces subdivisions de l’espace permettent notamment de **simuler efficacement la nature torique** du volume de simulation. Cette représentation du tore par une grille nous permet également de fournir une solution performante au problème du calcul des distances entre deux points dans le tore. En effet, un espace torique modifie le calcul de cette distance dès lors que le plus court chemin peut ne pas être la simple ligne droite directe d’un point de l’espace à un autre, mais peut nécessiter de traverser au moins une des frontières. La figure 4.4.1 illustre cette situation :

- dans la partie (a), on peut voir le calcul simple de la distance entre deux points A et B dans un plan borné.
- dans la partie (b), A et B sont deux points dans un espace torique en 2D. On peut voir qu’il existe 9 chemins à partir de A pour atteindre B.
- la partie (c) montre que le plus court chemin entre A et B dans cette espace est le chemin vert et non le rouge comme on aurait pu le supposer à première vue. En effet, ce chemin traverse une frontière et ne correspond pas au chemin “direct”.

A noter que le nombre de calculs augmente encore plus avec un espace 3D puisque dans ce cas, il faut considérer 27 chemins possibles, et ce pour le calcul d’une seule distance. On comprend aisément la motivation à trouver une méthode efficace résumant en une seule opération le calcul d’une distance.

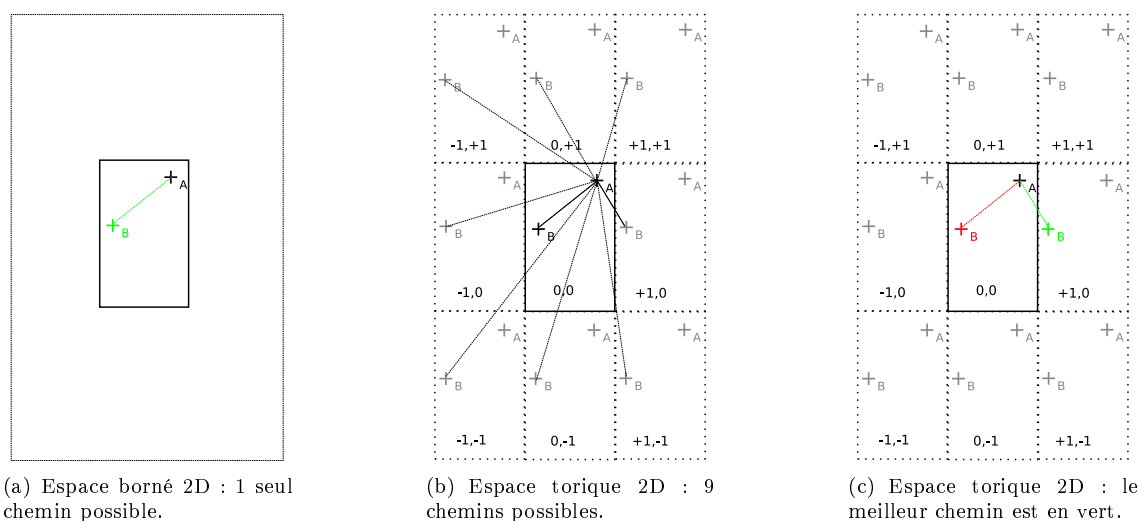


FIG. 4.4.1 – Illustration de la difficulté que pose la nature torique d'un espace pour le calcul de la distance entre deux points A et B.

4.4.2 Le temps

Suite à ce que nous avons vu en section 2.3.4, nous pouvons préciser que les simulations d'une membrane biologique se situent dans le domaine des **simulations temporalisées** de molécules qui entrent **ponctuellement en interactions non prédictibles**. Une boucle générale de simulation assurera donc un **synchronisation globale** par une technique assimilable à du *double buffering* (voir section 4.7.8).

Par la suite ce choix s'avèrera d'autant plus approprié qu'il permettra d'intégrer une **technique à pas de temps variable** (ou adaptatif) sans pénaliser les simulations. Un pas de temps variable induit quelques difficultés théoriques résolues en section 4.5.3, notamment par rapport à la modélisation physiquement réaliste du déplacement brownien des molécules.

4.5 La dynamique des molécules

4.5.1 L'équation du mouvement

L'équation du mouvement est l'expression mathématique qui répond à la question suivante : *“comment établir la corrélation entre les forces appliquées aux molécules et les déplacements de ces mêmes molécules ?”*.

En effet, les interactions entre les molécules se traduisent par des forces, qui dérivent en l'occurrence de potentiels de type Lennard-Jones (voir section 4.2.2). Dans la perspective d'étudier les dynamiques d'évolution des systèmes simulés, il est nécessaire de définir une équation du mouvement.

À ce niveau, nous allons présenter la *Dissipative particle dynamics* (DPD), une technique utilisée pour modéliser la dynamique des grains dans un solvant.

► La DPD

La DPD est une **technique stochastique** de simulation², qui a été initialement proposée par Hoogerbrugge et Koelman en 1992 ([HK92] puis [KH93]). En combinant différents aspects de la

²Par stochastique on entend qu'on a uniquement introduit un terme de stochasticité dans l'équation du mouvement.

dynamique moléculaire et des gaz sur réseau³ (*Lattice Gas Automata* ou LGA), cette méthode explore des échelles hydrodynamiques de temps et d'espace bien plus grande que celles accessibles par la dynamique moléculaire tout en évitant les biais induits par le réseau de la seconde méthode. Hoogerbrugge and Koelman ont montré par simulations et dérivations théoriques que l'algorithme de DPD peut sous certaines conditions obéir aux équations de Navier-Stokes⁴, c'est-à-dire se révéler correcte d'un point de vue thermodynamique. Ce premier modèle a été substantiellement reformulé et légèrement modifié par Espanol et Warren [EW95] en 1995 pour garantir un état d'équilibre thermique et s'assurer qu'une distribution propre de Boltzmann était bien générée. Groot et Warren [GW97] ont également contribué à préciser la nature du bruit blanc⁵ et la friction ainsi qu'à tester différents algorithmes améliorant les performances des simulations.

Dans la DPD, l'ensemble des forces agissantes sur un grain i est exprimé par la sommation sur tous les grains j du voisinage du grain i de trois forces de type additives par paire (Eq.(4.5.1)).

$$\vec{f}_i = \sum_{j \neq i} (\vec{F}_{ij}^C + \vec{F}_{ij}^D + \vec{F}_{ij}^R) \quad (4.5.1)$$

$$\vec{F}_{ij}^C = \begin{cases} a_{ij}(1 - r_{ij}/R_c)\hat{r}_{ij} & \text{pour } r_{ij} < R_c \\ 0 & \text{pour } R_c \leq r_{ij} \end{cases} \quad (4.5.2)$$

$$\vec{F}_{ij}^D = -\eta\omega^D r_{ij}(\hat{r}_{ij} \cdot \vec{v}_{ij})\hat{r}_{ij} \quad (4.5.3)$$

$$\vec{F}_{ij}^R = \sigma\omega^R r_{ij}\zeta_{ij}\hat{r}_{ij} \quad (4.5.4)$$

$$\text{avec } \omega^R(r) = \begin{cases} (1 - r/R_c) & \text{pour } r < R_c \\ 0 & \text{pour } R_c \leq r \end{cases} \quad (4.5.5)$$

Le premier terme \vec{F}_{ij}^C fait référence aux **forces conservatives**, c'est-à-dire aux forces œuvrant entre deux grains. Pour ce qui est des forces intermoléculaires, la plupart des études de DPD utilisent des potentiels à *coeur mou* type Eq.(4.5.2) où les paramètres $a_{ij} > 0$ représentent l'intensité maximale de la répulsion, $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$ le vecteur séparant les grains i et j (d'ou une distance de séparation $r_{ij} = |\vec{r}_{ij}|$) et R_c la distance de coupure du potentiel.

Les **forces dissipatives** \vec{F}_{ij}^D (Eq.(4.5.3)) et les **forces aléatoires** \vec{F}_{ij}^R (Eq.(4.5.4)) expriment respectivement la friction des grains dans le solvant et le mouvement brownien de ces mêmes grains dû aux chocs avec les molécules du solvant. Dans ces expressions, $\vec{v}_{ij} = \vec{v}_i - \vec{v}_j$ est la vitesse relative du grain i par rapport au grain j , η est le coefficient de friction, σ est l'amplitude du bruit et ζ_{ij} est une variable aléatoire suivant la loi normale centrée réduite $\mathcal{N}(0, 1)$.

4.5.2 Modèle de déplacement des molécules

A la suite de l'étude de ces modèles de forces proposées par la DPD, nous avons choisi de conserver nous aussi la distinction entre ces trois façons d'influencer le mouvement des molécules. Dans notre modèle nous aurons une **force conservative** \vec{F}^C et une **force dissipative** \vec{F}^D relativement similaires à celles de la DPD ainsi que le calcul d'un **mouvement brownien** pour les molécules ajoutant un terme aléatoire aux calculs des déplacements de la molécule (voir section 4.5.2).

Dans MitoMAS, la représentation du solvant est implicite, il est donc nécessaire d'intégrer au modèle l'influence de ce solvant sur le comportement des molécules. C'est pourquoi, parmi les potentiels intermoléculaires qui participent au calcul de la **force conservative**, l'un de ces potentiels (cf Eq.(4.2.2)) représente la répulsion entre les têtes et les queues simulant ainsi l'affinité des têtes pour le solvant et l'hydrophobie des queues.

³On désigne par LGA un type d'automate cellulaire destiné à simuler le comportement d'un fluide, ie déterminer des solutions numériques aux équations de Navier-Stokes. Un gaz sur réseau modélise l'espace sous la forme d'un réseau à deux ou trois dimensions. Le fluide est modélisé par des particules de masse identique qui peuvent se déplacer entre les noeuds du réseau.

⁴En mécanique des fluides, les équations de Navier-Stokes sont des équations aux dérivées partielles non-linéaires qui décrivent le mouvement des fluides dans l'approximation des milieux continus.

⁵Un bruit blanc est une réalisation d'un processus aléatoire dans lequel la densité spectrale de puissance est la même pour toutes les fréquences.

Il est également nécessaire d'ajouter la représentation du mouvement des molécules induit par la rencontre des grains et des molécules d'eau. C'est cette interaction qui est codée par le mouvement brownien (ce mouvement brownien est non contraint car il ne privilégie aucune direction).

► L'équation du mouvement

Nous allons maintenant donner les détails de l'équation du mouvement du modèle MitoMAS.

Comme il est proposé dans l'ouvrage de H.C.Berg[B9393] et expliqué ci-dessous, nous avons simplifié les lois usuelles de la mécanique classique.

Considérons :

- \vec{X} le vecteur position d'un grain dans l'espace;
- $\dot{\vec{X}} = \frac{d\vec{X}}{dt}$ son vecteur vitesse;
- $\ddot{\vec{X}} = \frac{d^2\vec{X}}{dt^2}$ son vecteur accélération.

Nous venons de décomposer la force globale agissant sur un grain comme la somme de trois forces (Eq.(4.5.1)). Dans notre modèle, nous écrivons :

- la force conservative \vec{F}^C est la somme des forces issues des potentiels intermoléculaires;
- la force dissipative \vec{F}^D s'écrit sous la forme Eq.(4.5.6) avec un coefficient de friction $\lambda = 6\pi r\eta$ où η est la viscosité du solvant et r le rayon du grain;

$$\vec{F}^D = -\lambda\dot{\vec{X}} \text{ (loi de Stokes)} \quad (4.5.6)$$

avec $\lambda = 6\pi r\eta$

- la force aléatoire \vec{F}^R qui est à l'origine du mouvement brownien.

A partir de la seconde loi de Newton (Eq.(4.5.7)), qui établit l'accélération d'un corps comme la résultante des forces qui s'y appliquent, nous pouvons donc réécrire notre équation :

$$\sum_i \vec{f}_i = m\ddot{\vec{X}} \quad (4.5.7)$$

$$-\lambda\dot{\vec{X}} + \vec{F}^C + \vec{F}^R = m\ddot{\vec{X}} \quad (4.5.8)$$

Dans un solvant, le coefficient de friction du milieu est tel que **la force de friction prédomine** l'équation Eq.(4.5.8). Avec r le rayon d'un grain, on a un coefficient de friction $\lambda = 6\pi r\eta = \lambda_c r$ qui évolue en r et $m = \rho 4/3\pi r^3$ (ρ , la masse volumique du grain) qui évolue en r^3 , on obtient l'équation Eq.(4.5.9). Pour r très petit, on peut conserver cette forme Eq.(4.5.10).

$$\vec{F}^C + \vec{F}^R = m\rho r^3\ddot{\vec{X}} + \lambda_c r\dot{\vec{X}} \quad (4.5.9)$$

$$\text{d'où } \vec{F}^C + \vec{F}^R = \lambda_c r\dot{\vec{X}} + o(r) \quad (4.5.10)$$

Ainsi, à l'échelle mésoscopique, les forces appliquées à une molécules peuvent être directement liées à sa vitesse $\dot{\vec{X}}$, et non plus à son accélération. **L'inertie est dominée par l'amortissement.** En réalité, nous ne souhaitons pas appréhender l'agitation thermique des molécules sous la forme de la force \vec{F}^R mais **directement sous la forme d'un déplacement aléatoire** conforme aux équations régissant le mouvement brownien.

► Le mouvement brownien

L'équation de la diffusion, phénomène induit par le mouvement brownien des molécules, est décrite par la densité de probabilité volumique $P_D^v(r, \tau)$ qu'une particule ait parcouru une distance r point à point, durant un temps τ , sous la forme de l'équation Eq.(4.5.11) avec d la dimension de

l'espace et D le coefficient de diffusion. Ce coefficient de diffusion fait intervenir la température T (agitation thermique), la constante des gaz parfait R , le nombre d'Avogadro N_{av} et le coefficient de friction $\lambda = 6\pi\eta r$.

$$P_D^v(r, \tau) = \left(\frac{1}{4\pi D\tau}\right)^{\frac{d}{2}} e^{-\frac{r^2}{4D\tau}} \quad (4.5.11)$$

avec $D = \frac{RT}{N_{av}\lambda} = \frac{RT}{N_{av}6\pi\eta r}$

Plutôt que de chercher directement à générer un vecteur dans l'espace 3D puis de lui attribuer une norme, il est plus simple de manipuler la densité de probabilité volumique en choisissant $d = 1$ comme indiqué en Eq.(4.5.12) pour obtenir chacune des trois coordonnées du vecteur de "déplacement brownien" ν en trois tirages indépendants. La densité de probabilité d'une loi normale (ou gaussienne) $\mathcal{N}(\mu, \sigma^2)$ (où $\sigma > 0$) est donnée par l'équation Eq.(4.5.13). On passera de l'Eq.(4.5.12) à l'Eq.(4.5.13) en prenant $\sigma = \sqrt{2D\tau}$ et $\mu = 0$.

$$P_D^v(r, \tau) = \left(\frac{1}{4\pi D\tau}\right)^{\frac{1}{2}} e^{-\frac{r^2}{4D\tau}} \quad (4.5.12)$$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (4.5.13)$$

En rajoutant le "déplacement brownien" à l'équation Eq.(4.5.10), on aboutit à l'équation du mouvement Eq.(4.5.14) de notre modèle où U , V et W sont trois variables aléatoires indépendantes qui suivent la loi normale centrée réduite $\sim \mathcal{N}(0, 1)$.

$$\vec{X}_{t+\tau} - \vec{X}_t = \frac{\sum_i f_i}{\lambda} \tau + \sqrt{2D\tau} \begin{pmatrix} U \\ V \\ W \end{pmatrix} \quad (4.5.14)$$

Ce qui donne en remplaçant le coefficient de friction λ et celui de diffusion D par leurs expressions respectives :

$$\vec{X}_{t+\tau} - \vec{X}_t = \frac{\sum_i f_i}{6\pi\eta r} \tau + \sqrt{\frac{RT}{N_{av}3\pi\eta r}} \tau \begin{pmatrix} U \\ V \\ W \end{pmatrix} \quad (4.5.15)$$

4.5.3 Gestion du pas de temps

Étant donné une équation du mouvement il existe différents algorithmes pour calculer les positions des éléments. A partir de l'étude de ces méthodes (données en annexe A.3), nous avons choisi de simplement intégrer l'équation du mouvement (4.5.15) sur le temps variable τ . Cette approximation, motivée essentiellement pour des raisons de performance, est d'autant plus justifiée que nous avons lié les forces non plus à l'accélération mais directement à la vitesse. Dans la perspective de minimiser les erreurs d'intégration, nous proposons de borner le déplacement linéaire des grains à 1 Å qui se révèle être l'ordre de grandeur des liaisons entre les atomes. Nous nous étendrons plus longuement sur ces différentes étapes dans la section 4.7.8.

4.6 En résumé : le modèle bio-physique

Nous avons présenté dans cette section les principales propriétés du modèle de MitoMAS.

Nous partons sur la base de 3 types de grains, ce qui permet dans un premier temps de modéliser les lipides sous la forme de trimères linéaires pour notamment rendre compte correctement de l'encombrement stérique des molécules. Avec ces mêmes types, il est possible de représenter différents modèles de lipides, en attribuant par exemple plus de grains par type, et même d'étendre le modèle à la représentation de protéines telles que les complexes membranaires. Toutes ces

molécules peuvent être rigides ou flexibles. Dans ce dernier cas, la dynamique interne de la molécule implique des potentiels de liaison ou de flexion.

En ce qui concerne les interactions intermoléculaires, nous proposons d'intégrer différentes expressions de potentiel attractif/répulsif, dont certaines sont des variantes éloignées des potentiels de Lennard-Jones. Parmi ces potentiels, la répulsion stricte entre les têtes et les queues modélise l'hydrophobie des lipides, ce qui nous permet d'avoir un modèle à solvant implicite. Le volume de simulation est un pavé droit de l'espace à faces jointives qui simule efficacement un espace torique grâce à sa discrétisation par une grille.

En décomposant la force globale appliquée à un grain en trois termes et en utilisant une simplification de la mécanique newtonnienne, nous obtenons une équation du mouvement qui, d'une part, lie les forces appliquées aux grains à leurs vitesses et d'autre part, inclut un mouvement brownien directement sous la forme d'un déplacement. Enfin, pour limiter les erreurs d'intégration de cette équation du mouvement, nous optons pour un mécanisme de pas de temps variable qui borne le déplacement linéaire des grains.

Après avoir spécifié toutes les propriétés physiques du modèle, nous présentons maintenant la conception agent qui lui est associée.

4.7 Le modèle agent de MitoMAS

4.7.1 La modélisation orientée objet

La plateforme agent de MitoMAS a été réalisée suivant le paradigme objet (*Object-Oriented Programming*)[AC96] qui consiste à définir et à assembler des briques logicielles appelées objets ; un objet représente un concept, une idée ou toute entité du monde physique, comme une molécule ou un atome.

Dans un modèle orienté objet, les entités (ou instances) sont définies à partir de classes et de relations entre ces classes. Une classe peut être vue comme un type de donnée. Elle est caractérisée par des attributs et des comportements. On parle de relation d'héritage lorsqu'une classe est spécialisée en sous-classes et de relation de composition ou d'agrégation entre une classe et les classes qui caractérisent ses attributs. L'intérêt de ces modèles est de favoriser la conception de modules génériques faciles à étendre pour des besoins spécifiques. Dans le contexte de la modélisation d'un modèle à grains, nous avons choisi ce paradigme afin de décrire un cadre général pour l'implémentation d'un phospholipide sous forme de grains, cadre qui sera spécialisé suivant que l'on désirera implémenter des modèles rigides, flexibles, avec plus ou moins de grains.

Les sections suivantes présentent l'architecture du coeur de MitoMAS, c'est-à-dire le moteur de simulation. Sans entrer dans une dissection du code, ce qui n'aurait que peu d'intérêt, nous allons présenter les classes essentielles du modèle.

► Une conception orientée agent

En terme de modélisation agent, la première classe du modèle aussi appelée super-classe, est la classe **Agent** qui regroupe les propriétés et les comportements communs à ses sous-classes **BioAgent** et **GridAgent**. Dans la figure 4.7.1, on trouve le diagramme de classes qui montre la relation d'héritage entre ces trois classes. On notera que la classe **BioAgent** est elle-même spécialisée en deux sous-classes : **Phospholipid** et **EnzymaticComplex**. Les comportements génériques tels que la méthode `life_cycle` sont définis à partir de la classe

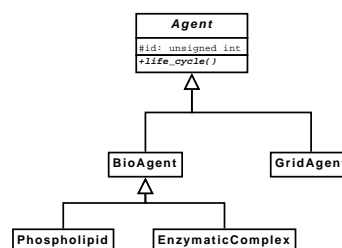


FIG. 4.7.1 – La super classe **Agent** et ses dérivées.

Agent car elle caractérise tout comportement d'agent. Chaque sous-classe aura ensuite la responsabilité de redéfinir les actions de ses instances.

4.7.2 Les agents biologiques.

► La classe `BioAgent`

Comme son nom l'indique, la classe `BioAgent` (Fig.4.7.2) prend en charge la représentation des objets biologiques.

> Les attributs

Un `BioAgent` est défini principalement par un identifiant `id` (hérité de la classe `Agent`) et par un attribut `collectionOfAtomSets` qui référence la collection des grains qui le composent, chaque grain étant une instance de la classe `AtomSet` (cf paragraphe suivant). Un `BioAgent` possède une position dans l'espace 3D (les attributs `absoluteSpatialPosition3D` et `nextAbsoluteSpatialPosition3D`). Les quatre attributs suivants sont utilisés pour capturer le résultat du calcul des forces (l'attribut `conservativeForce`) et des mouvements (l'attribut `randomMove`) envoyés au `BioAgent` par le modèle d'interactions (cf paragraphe suivant). Nous avons ensuite un ensemble d'attributs dédiés à la gestion des mouvements ou de la position spatiale du `BioAgent`. Par exemple on peut relever l'attribut `spatialOrientation` qui est de type `MMQuaternion`. La classe `MMQuaternion` (nom présentée dans le diagramme de classes) est une redéfinition pour MitoMAS d'un type qui permet la gestion de 4 réels définissant une rotation dans l'espace et des opérations telles que la composition de rotations successives (cf annexe A.1 pour une explication complète). Dans le cas où on utilise une équation du mouvement qui ne prend pas en compte la masse des objets, il est possible de compenser en donnant une valeur d'inertie (l'attribut `inertia`⁶). Pour faciliter la gestion des déplacements du `BioAgent`, chaque `BioAgent` connaît l'élément de la grille auquel il appartient grâce à l'attribut `gridAgentOfSpatialPosition`. On considère que le `GridAgent` du `BioAgent` est celui qui contient le centre de gravité de la molécule.

> Les comportements

Parmi les comportements de cette classe, on retrouve la méthode `life_cycle` héritée de la super classe `Agent`. C'est à ce niveau que sont précisés les détails des actions du `BioAgent` pour un pas de temps (la méthode `life_cycle` de la classe `Agent` est une simple méthode abstraite, c'est-à-dire sans code associé). Les deux comportements `determine_next_state` et `compute_next_state` permettent de gérer l'état présent et futur du `BioAgent` avant que `swap_states` ne les intervertisse. Ce mécanisme de *double buffering* permet notamment de synchroniser les `BioAgents` entre eux. C'est aussi au niveau de ces deux comportements qu'est réalisée la gestion du pas de temps (variable ou non) comme nous le verrons par la suite. Deux autres méthodes ont trait à l'initialisation du `BioAgent` : `initStructure` et `initPositions`. Le comportement `initStructure` crée les grains en les liant les uns aux autres suivant la structure moléculaire (définition du voisinage intramoléculaire de chaque grain, ce voisinage intramoléculaire étant présenté par la suite) tandis que `initPositions` positionne dans l'espace le `BioAgent` et ses grains pour construire son volume.

► Les classes `Phospholipid` et `EnzymaticComplex`

Grâce au mécanisme d'héritage, les deux sous-classes `Phospholipid` et `EnzymaticComplex` bénéficient des comportements de la super-classe `BioAgent`. Ainsi ces classes contiennent toutes deux des collections de grains. Ces deux classes dérivées rajoutent certaines instructions au comportement `initPositions` de base fourni par la classe `BioAgent`. En effet, la manière de disposer les grains diffère entre un phospholipide et une protéine (par exemple un complexe enzymatique) car dans ce dernier cas, les dimensions du volume sont mises à contribution pour positionner les grains.

⁶Avec une équation du mouvement déduite par exemple de la seconde loi de Newton, il est possible de coder l'inertie à partir de la masse des grains qui composent le `BioAgent`.

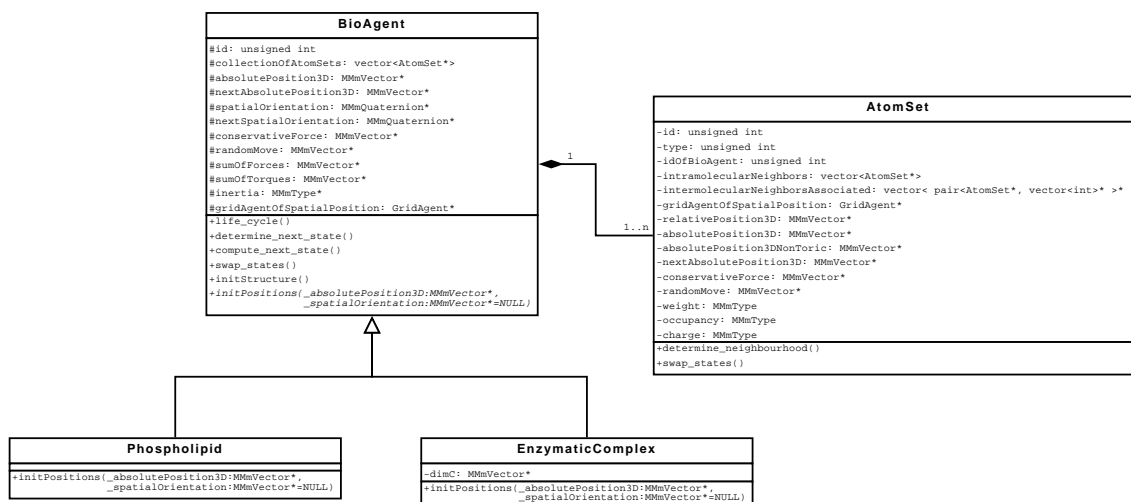


FIG. 4.7.2 – Diagramme de classes de BioAgent et de ses dérivées.

➤ La classe AtomSet

La classe `AtomSet`, correspondant à un grain d'atomes, est l'entité de plus fine granularité de notre modèle.

➤ Les attributs

En ce qui concerne les grains d'atomes, bien que nous ayons défini 3 types de grain dans notre modèle *théorique* (section 4.1.2), au niveau de l'implémentation nous constatons que les seules différences qui existent entre les types de grain se situent au niveau du choix des méthodes de calcul des forces à appliquer entre chaque paire de grains. Nous avons donc choisi de faire une seule classe de grain et de stocker le type réel du grain dans un attribut de la classe `AtomSet`. Lorsque deux grains interagissent, le bon calcul est alors choisi en fonction des types respectifs de chacun. Nous allons maintenant décrire la classe `AtomSet`.

L'identifiant et le type d'un grain sont respectivement l'attribut `id` et `type`. Pendant le calcul des interactions entre deux grains, il est nécessaire de savoir si ces deux grains appartiennent au même `BioAgent`. Un accès rapide à cette information est fourni en stockant l'identifiant du `BioAgent` à l'intérieur de l'`AtomSet` (attribut `idOfBioAgent`). Les interactions intramoléculaires nécessitent l'identification des grains voisins directs : l'attribut `intramolecularNeighbors`. Ce voisinage définit un sous-ensemble des grains constituant le `BioAgent`. Les conditions de la définition de ce sous-ensemble peuvent être reliées soit au modèle de lipide (flexible ou rigide), soit au fait que la molécule contienne un très grand nombre de grains et que tous ne soient pas en interaction directe. La présence de cet attribut fournit à la fois une manière simple et souple de coder cette information. L'attribut `gridAgentOfSpatialPosition` correspond au `GridAgent` qui contient l'`AtomSet`. On notera qu'un `AtomSet` peut appartenir à un `GridAgent` différent de celui qui contient le `BioAgent` de l'`AtomSet`. Enfin un ensemble de vecteurs et de données concernant la position et la dynamique d'un grain. Il faut toutefois préciser le rôle de l'attribut `absolutePosition3DNonToric` : ce vecteur correspond à la position non modifiée par le volume torique. Cette donnée intervient dans le calcul des couples dans le cas d'un modèle rigide de molécule.

➤ Les comportements

En dehors du comportement `swap_states` qui fait tout simplement échos à `swap_states` de la classe `BioAgent`, l'autre comportement de la classe `AtomSet` est `determine_neighbourhood` qui se charge de constituer le voisinage de l'`AtomSet` en mettant à contribution son `gridAgentOfSpatialPosition`. Ce comportement est appliqué au niveau de l'`AtomSet` et non au niveau du `BioAgent` afin de tenir compte du fait qu'un `BioAgent` peut se répartir entre plusieurs `GridAgents`, c'est-à-dire que ses `AtomSets` peuvent être associés à plus d'un `GridAgent`. On risquerait alors de fausser l'évaluation du voisinage d'un `AtomSet` si on se référait seulement au `GridAgent` associé à la position du `BioAgent` (en l'occurrence, de son centre de gravité), ce qui

induirait un biais dans le calcul des interactions.

4.7.3 Différents modèles d'interactions

Nous avons vu dans le chapitre précédent qu'une molécule peut être modélisée comme étant rigide ou flexible. Dans le premier cas, les grains conservent des positions relatives fixes tout au long de la simulation. Dans le second cas, des potentiels de liaison ou de flexion permettent aux grains d'avoir plus ou moins de liberté dans leurs positions tout en étant contraints de respecter des distances et des angles. En modélisation orienté objet, il existe un mécanisme simple pour employer plusieurs modèles (d'interactions dans notre cas) dans une même simulation. On appelle ce mécanisme le polymorphisme. Cela consiste à donner le même nom de fonction ou de méthode à différentes portions de code. Suivant que l'on attache l'un ou l'autre de ces codes à une classe, elle adoptera le comportement correspondant de façon transparente pour les autres objets de la simulation. La figure 4.7.3 montre l'ensemble des classes utiles pour l'implémentation des modèles d'interactions intra- et inter-moléculaires.

► La classe `ModelOfInteractions`

La classe `ModelOfInteractions` est utilisée pour accéder aux différents modèles d'interactions inter- ou intra-moléculaires proposés pour la simulation.

Dans l'état actuel des simulations, le modélisateur détermine à l'initialisation du système si les interactions seront flexibles ou rigides. Ce choix est répercuté au niveau des `BioAgents` qui à chaque pas de temps doivent connaître le modèle d'interaction à appliquer entre leurs `AtomSets` et les voisins de ces `AtomSets` (cf la définition du voisinage d'un `AtomSet`). Cette procédure s'applique dans le cas où la composition en `BioAgent` est homogène dans toute la simulation. Dans le cas où l'on désirerait mélanger plusieurs types de `BioAgents` (`Phospholipid` et `ComplexEnzymatic`) dans un même volume de simulation, différents modèles d'interactions flexibles ou rigides pourront être définis et l'attribution dynamique d'un modèle spécifique pourra alors être réalisée en fonction de la composition du `GridAgent` qui contient le `BioAgent` dont on souhaite faire évoluer l'état. Par exemple un phospholipide à proximité d'un complexe enzymatique pourrait adopter un modèle d'interactions flexible à un pas de temps, puis au pas de temps suivant retrouver un modèle d'interaction rigide s'il venait à se déplacer dans un autre `GridAgent` ne contenant que des phospholipides.

> Les attributs

Les attributs `maximalDisplacement` et `minimalTimeStep` correspondent respectivement au maximum autorisé pour le déplacement (linéaire) d'un grain dans l'optique de limiter les erreurs d'intégration et au pas de temps (variable) courant de la simulation. L'attribut `dimOfWorld` permet de connaître les dimensions du volume de simulation.

Concernant les potentiels intermoléculaires, l'attribut `interPotentialFunctionsTab` fournit pour toute combinaison d'interactions entre une paire d'`AtomSets` la fonction qui permet de calculer cette interaction. L'attribut `paramInter` permet de connaître pour chaque paire d'`AtomSets` en interaction les valeurs de paramètres à utiliser dans le calcul du potentiel. Il s'agit des exposants et des valeurs de ϵ et σ pour les équations de Eq.(4.2.1) à Eq.(4.2.4). Les attributs `distFunctionsTab` et `distValuesTab` sont optionnels (choix à la compilation) et correspondent à l'utilisation de deux matrices carrées de taille n (n étant le nombre d'`AtomSets` instanciés) pour limiter les calculs des distances tout en prenant en compte le partitionnement de l'espace. Notons que sans partitionnement de l'espace, la matrice carrée des distances peut être réduite à une matrice triangulaire, limitant ainsi le nombre de calculs de distance. Les attributs `intensityMaxOfSumOfForces` et `intensityMaxOfSumOfTorques` servent à calculer le pas de temps à appliquer pour le pas de simulation en cours (voire section 4.2.1). Les autres attributs sont impliqués dans le calcul des futures positions des `BioAgents` et de leurs `AtomSets`. Pour la génération de nombres aléatoires (attributs `rng_x`, `rng_y` et `rng_z`), nous avons utilisé la bibliothèque `boost`⁷ qui fournit notamment un ensemble de générateurs de nombres aléatoires. Pour nos simulations nous avons actuellement sélectionné

⁷<http://www.boost.org/>

le générateur “*lagged_fibonacci607*” qui garantit de grandes séquences de nombres aléatoires ayant de bonnes propriétés d’indépendance.

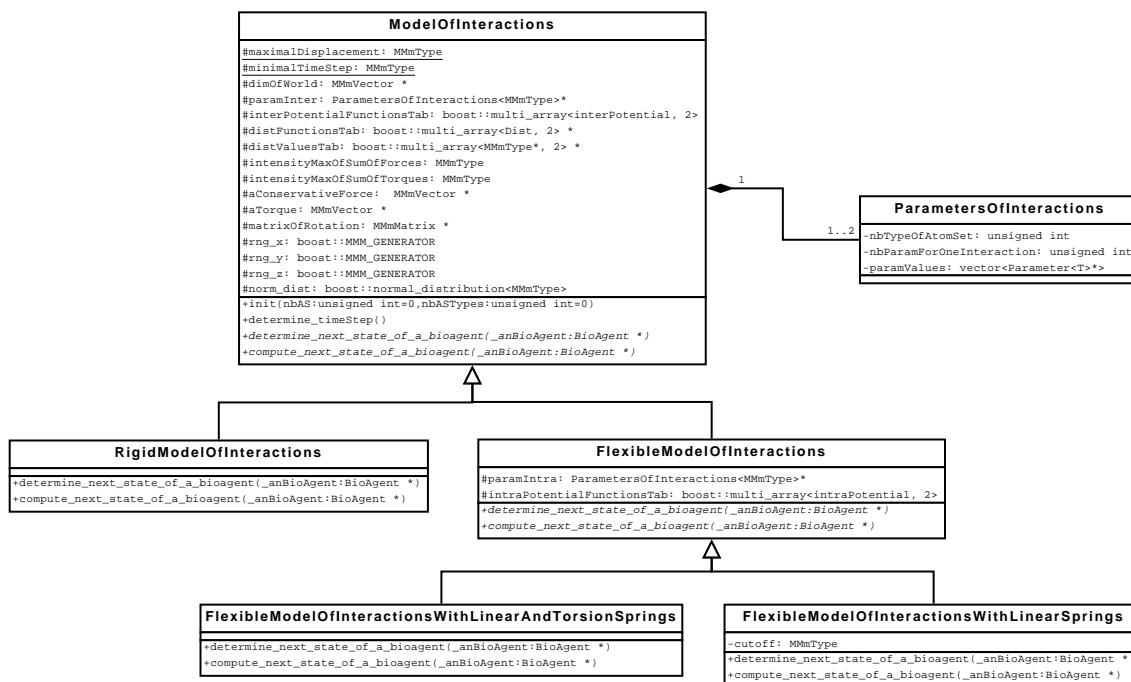


FIG. 4.7.3 – Diagramme de classes des spécialisation de ModelOfInteractions.

> Les comportements

La méthode `init` permet d’initialiser la simulation (matrices optionnelles `distFunctionsTab` et `distValuesTab` et les matrices des interactions) à partir d’un nombre d’`AtomSets` et de leur type. En dehors du comportement `determine_timeStep` qui sert à calculer le pas de temps à employer pour le pas de simulation en cours, on peut remarquer les comportements `determine_next_state_of_a_bioagent` et `compute_next_state_of_a_bioagent` qui sont des méthodes abstraites qui obligent les sous-classes de `ModelOfInteractions` à définir ces deux étapes du cycle de vie d’un `BioAgent`.

> Les classes dérivées

La sous-classe `RigidModelOfInteractions` fournit l’implémentation du calcul des équations des potentiels intermoléculaires et l’équation du mouvement Eq.(4.5.15). La sous-classe `FlexibleModelOfInteractions` donne également l’implémentation de ces équations pour le modèle flexible mais aussi le calcul des forces intramoléculaires entre les `AtomSets` d’un même `BioAgent`. Pour ce faire, cette sous-classe possède deux attributs supplémentaires, `intraPotentialFunctionsTab` et `paramIntra`, qui s’occupent respectivement de fournir les potentiels intramoléculaires et leurs paramètres à appliquer entre les grains du même `BioAgent` et leurs voisins intramoléculaires. Les forces issues de ces potentiels intramoléculaires s’ajoutent à celles déjà issues des interactions intermoléculaires. Par la suite, le pas de temps à utiliser pour calculer les déplacements des grains est défini en fonction de l’intensité de ces forces sommées. On comprend donc pourquoi des modèles flexibles, générant des forces intramoléculaires à forte intensité, sont susceptibles de ralentir la simulation dans son ensemble.

Les deux sous-classes de `FlexibleModelOfInteractions` :

- `FlexibleModelOfInteractionsWithLinearAndTorsionSprings` et
- `FlexibleModelOfInteractionsWithLinearSprings`

sont deux propositions d’implémentation de deux modèles flexibles différents. Dans le premier cas, les potentiels de flexion et de liaison concernent uniquement les grains voisins directs (exemple B dans la figure 4.2.1) tandis que dans le second cas, les potentiels de liaison sont calculés entre un grain et les autres grains du même `BioAgent` considérés comme voisins, l’attribut *distance de*

coupure délimitant le voisinage (exemples C et D de la même figure). Ceci complique sensiblement les calculs puisqu'en plus de l'identification du voisinage intermoléculaire d'un grain, il faut lui définir son voisinage intramoléculaire à chaque pas de simulation.

4.7.4 Les agents environnement

► La classe Grid

La classe `Grid` sert à discrétiser le volume de simulation. Ces grilles sont des partitions régulières et non hiérarchiques. Dans l'état actuelle des simulations, l'espace n'est discrétisé que par une seule grille. Toutefois il est souvent utile de pouvoir mettre en place différentes finesses de grilles pour optimiser certaines tâches et pratiquer une certaine forme de multi-échelles. Par exemple, une première grille peut s'occuper de gérer les interactions entre lipides avec une distance de coupure courte, une deuxième grille peut rajouter des interactions plus longues distances avec une distance de coupure plus grande et enfin, une troisième grille peut gérer la diffusion de substances chimiques ou simuler un gradient de température modifiant localement l'agitation des molécules.

La figure 4.7.4 donne le diagramme de classes synthétique de `Grid`.

> Les attributs

Les attributs `collectionOfGridAgents` contient l'ensemble des éléments de type `GridAgent` qui discrétisent la grille.

> Les comportements

Parmi les comportements, `init` partitionne la grille initiale et instancie les `GridAgents`. Ce partitionnement est opéré pour que les dimensions d'un `GridAgent` soient les plus petites possibles tout en restant supérieures à la distance de coupure. On maximise ainsi l'efficacité du partitionnement de l'espace puisque les voisinages sont restreints tout en respectant la distance de coupure des potentiels intermoléculaires. Une fois les `GridAgents` créés, le comportement `fill_neighborhood` attribue à chaque `GridAgent` son voisinage direct. Pour un `GridAgent`, ce voisinage direct correspond aux 26 `GridAgents` contigus dans l'espace 3D, en tenant compte bien sûr de la nature torique de cet espace. Le comportement `go_one_step` induit simplement le comportement `life_cycle` de chaque `GridAgent`. La méthode `give_to_something_its_gridAgent` permet de renvoyer à un `BioAgent` ou un `AtomSet` le `GridAgent` qui le contient. La méthode `give_to_gridAgents_their_modelOfInteractions` permet d'ajouter à chaque `GridAgent` le modèle d'interactions qu'il devra attribuer aux `BioAgents` qu'il contient.

► La classe GridAgent

La classe `GridAgent` (Fig.4.7.4) prend en charge l'environnement et intègre la discrétisation de l'espace.

> Les attributs

Tout comme un `BioAgent`, un `GridAgent` est défini par un identifiant `id` (hérité de la classe `Agent`) Parmi les attributs de `GridAgent`, la dimension `dim` est la même pour toute les instances de cette classe. L'attribut `collectionOfNeighborsAssociated` permet à un `GridAgent` d'accéder rapidement à ses voisins. L'attribut `collectionOfAtomSets` est l'ensemble des `AtomSets` contenus par le `GridAgent` à un instant `t` de la simulation. L'attribut `countOfBioAgentTypesHosted` est utilisé pour dénombrer les types de `BioAgent` contenus également à un instant `t`.

> Les comportements

En tant que classe dérivée d'`Agent`, la classe `GridAgent` définit le comportement `life_cycle`. Cette méthode doit, d'une part, choisir l'interaction intramoléculaire à appliquer en fonction de la composition du `GridAgent` (grâce à l'attribut `countOfBioAgentTypesHosted`) pour ce pas de simulation mais aussi exécuter la méthode `give_to_an_atomSet_its_neighbors` qui se charge d'attribuer un voisinage aux `AtomSets` qu'il contient. Comme expliqué précédemment dans la section consacrée à la classe `AtomSet`, l'attribution d'un voisinage est un comportement manipulant un `AtomSet` et non un `BioAgent` car un `BioAgent` est composé d'`AtomSets` qui peuvent être contenus dans différents `GridAgents`.

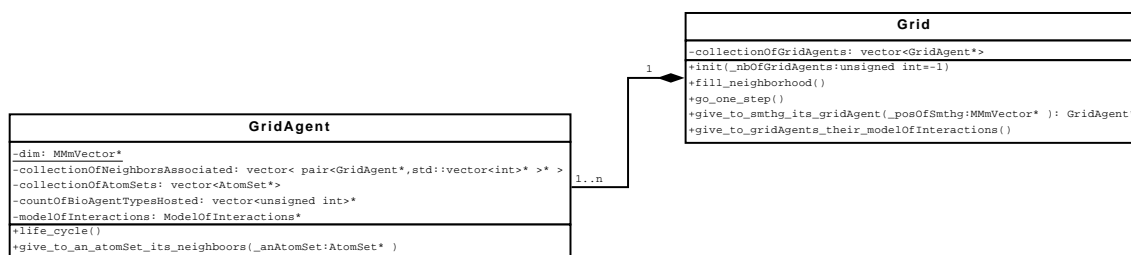


FIG. 4.7.4 – Diagramme de classes de GridAgent.

Les parties suivantes présentent dans le détail certains comportements impliquant les GridAgents, les BioAgents et leurs AtomSets.

4.7.5 Des comportements spécifiques

► La définition d'un voisinage intermoléculaire pour un AtomSet

Les interactions n'étant effectives que sur les AtomSets (ou grains), la définition d'un voisinage intermoléculaire consiste à fournir à chaque AtomSet contenu par le GridAgent l'ensemble des AtomSets qui constituent son voisinage intermoléculaire, c'est-à-dire les AtomSets à distance suffisamment restreinte pour entrer en interaction intermoléculaire et qui n'appartiennent pas au même BioAgent.

A chaque pas de la simulation, la première étape consiste donc à associer les AtomSets à leur GridAgent suivant l'algorithme 4.1 constant en temps.

Algorithme 4.1 Attribution d'un GridAgent à un AtomSet.

ENTRÉES: un AtomSet AS pourvu de coordonnées absolues $\{x, y, z\}$,
la Grid G dédiée à la gestion de l'espace, matrice de $l * m * n$ GridAgents, tous de dimension $\{a, b, c\}$
SORTIES: le GridAgent GA contenant l'AtomSet AS ou NULL si invalide

```

i ← [AS.x/a]
j ← [AS.y/b]
k ← [AS.z/c]
si  $0 \leq i < l$  et  $0 \leq j < m$  et  $0 \leq k < n$  alors
    GA ← G[i][j][k]
    Retourner GA
sinon
    Retourner NULL //exception à gérer au niveau du code appelant
fnsi
  
```

A noter qu'une alternative possible est la mise à jour individuelle des associations AtomSet/GridAgent. Par exemple, un grain se déplaçant et détectant le franchissement d'une frontière peut déclencher lui-même son transfert d'un GridAgent à un autre. Cette solution n'a pas été retenue, car la simulation se déroulant de manière synchrone, tous les agents se déplacent à chaque pas de temps, on peut donc supposer qu'un grand nombre d'AtomSets vont changer de GridAgent. Vu l'efficacité de l'algorithme 4.1, nous optons pour ce processus de ré-initialisation systématique des associations.

La seconde étape consiste à fournir à un grain la collection de grains avec lesquels il va interagir. L'AtomSet AS connaît son GridAgent hôte GA. Il lui suffit de lui demander son voisinage (Alg. 4.2). Comme le voisinage est utilisé pour le calcul des interactions intermoléculaires, il est nécessaire de s'assurer que le voisinage de AS n'inclut pas d'AtomSets appartenant au même BioAgent que lui ($\text{AtomSet}_i.\text{idBioAgent}() \neq \text{AS}.\text{idBioAgent}()$).

Algorithme 4.2 Attribution d'un voisinage à un AtomSet.

ENTRÉES: un AtomSet AS
SORTIES: le voisinage (variable) $\mathcal{V}(AS)$ de AS
 GA \leftarrow le GridAgent contenant AS ($AS \subset GA$)
 $\mathcal{V}_{GA} \leftarrow$ le voisinage (constant) de GA (26-connexité)
pour tout GridAgent $_i \in \{\mathcal{V}_{GA}, GA\}$ **faire**
 pour tout AtomSet $_i \subset$ GridAgent $_i$ **faire**
 si AtomSet $_i$.idBioAgent() \neq AS.idBioAgent() **alors**
 $\mathcal{V}(AS)$.ajouter(AtomSet $_i$)
 fin si
 fin pour
fin pour
 Retourner $\mathcal{V}(AS)$

Nous venons d'étudier le mécanisme d'attribution d'un voisinage à un AtomSet, ce mécanisme impliquant le GridAgent qui contient cet AtomSet. Par faciliter le calcul des distances dans l'espace torique, nous avons augmenté l'information caractérisant le voisinage.

► Prise en compte de l'espace torique

Un des problèmes majeurs dans la simulation d'un espace torique est que concrètement il est représenté par un volume tel qu'un parallélépipède dont il faut *réassembler* les frontières artificiellement. Comme nous l'avons vu dans la section 4.4.1, le calcul des distances doit être modifié pour tenir compte du fait qu'un GridAgent qui se trouve à une extrémité de la grille est en fait très proche du GridAgent qui se trouve à l'autre extrémité. Pour construire ce *recalage* des GridAgents frontières, nous avons défini un code qui caractérise la position relative des voisins d'un GridAgent. Cette position relative est un tripler d'entiers pris parmi $\{1,0,-1\}$.

Par exemple la figure 4.7.5 montre l'ensemble des positions relatives des voisins contigus du GridAgent 5 colorié en gris foncé. Parmi les voisins tous coloriés en gris clair de ce GridAgent, les GridAgents 1,2,6,9 et 10 sont associés à des triplets $\{0,0,0\}$ tandis que les GridAgents 4, 8 et 12 sont associés à des triplets $\{0,-1,0\}$. Le repère de la figure permet de rattacher chaque entier de ces triplets à la dimension qui leur est due.

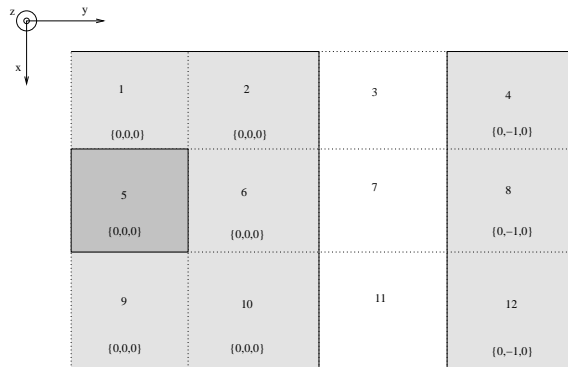


FIG. 4.7.5 – Voisinage élaboré d'un GridAgent.

Par la suite, l'attribution du voisinage à un AtomSet tient compte des positions relatives des GridAgents qui contiennent les AtomSets voisins. En d'autres termes, une paire voisinage P **{un-AS-voisin, une-position-relative}** associe un AtomSet voisin à une référence sur la position relative du GridAgent dont il provient.

Enfin, l'algorithme 4.3 montre comment, grâce à ce voisinage élaboré, il est possible de tirer partie de l'information supplémentaire contenue par la variable **une-position-relative** pour calculer directement la distance associée au chemin le plus court entre deux grains.

Algorithme 4.3 Calcul de la distance entre un grains et un de ses voisins.

ENTRÉES: un AtomSet AS,
 une paire voisinage P **{un-AS-voisin, une-position-relative}**
 le volume V avec ses dimensions $\{V[i], i \in \{0..d\}\}$
SORTIES: la distance du chemin le plus court entre AS et un-AS-voisin
 un-AS-voisin \leftarrow P.AS-voisin
 une-position-relative \leftarrow P.position-relative
 positionTmp.x \leftarrow un-AS-voisin.x + position-relative[0] * V[0]
 positionTmp.y \leftarrow un-AS-voisin.y + position-relative[1] * V[1]
 positionTmp.z \leftarrow un-AS-voisin.z + position-relative[2] * V[2]
 dist \leftarrow distance-euclidienne (AS.position, positionTmp)
 Retourner dist

Nous venons de voir comment il est possible de tirer partie du partitionnement de l'espace pour modéliser efficacement la nature torique de l'espace. Nous allons maintenant nous intéresser à une autre tâche à la charge d'un `GridAgent` : l'attribution à un `BioAgent` qu'il contient du modèle d'interactions intramoléculaires à mettre en oeuvre.

4.7.6 Les interactions intermoléculaires

Pour rappel, la section 4.2.2 donne les différents potentiels et leurs paramètres à appliquer entre les trois types de grain définis par notre modèle.

L'algorithme 4.4 explique comment sont attribués à l'exécution les différents potentiels type LJ ainsi que leurs paramètres. Ce mécanisme implique les deux structures `interPotentialFunctionsTab` et `paramInter` qui sont les matrices contenant respectivement les potentiels type LJ et leurs paramètres à mettre en oeuvre entre une paire de grains donnée. Le codage du type d'un `AtomSet` sous la forme d'un entier (par exemple : 0 pour le type `tête`, 1 pour le type `interface` et 2 pour le type `queue`) permet un accès direct aux différentes matrices. La fonction `dist` calcule les distances suivant l'algorithme 4.3 donné précédemment. La valeur de retour de cet algorithme, le vecteur \vec{F} , sera utilisé par l'`AtomSet` AS pour mettre à jour ses coordonnées.

Algorithme 4.4 Utilisation des différents potentiels type LJ avec les paramètres associés.

```

ENTRÉES: un AtomSet AS,
  {AtomSets} l'ensemble des voisins de AS,
  interPotentialFunctionsTab la matrice  $n \times n$  des potentiels type LJ pour n types de grain,
  paramInter la matrice  $n \times n \times k$  des k paramètres associés à chaque potentiel
SORTIES:  $\vec{F}$  la force conservative exercée sur AS par son voisinage
pour tout  $AS_i \in \{\text{AtomSets}\}$  faire
  i  $\leftarrow$  AS.type
  j  $\leftarrow$   $AS_i$ .type
   $\epsilon \leftarrow$  paramInter[i][j][0]
   $\sigma \leftarrow$  paramInter[i][j][1]
  exposant1  $\leftarrow$  paramInter[i][j][2]
  exposant2  $\leftarrow$  paramInter[i][j][3]
  d  $\leftarrow$  dist(AS,  $AS_i$ )
  si  $d \leq R_c$  alors
    I  $\leftarrow$  interPotentialFunctionsTab[i][j](d,  $\epsilon$ ,  $\sigma$ , exposant1, exposant2)
  sinon
    I  $\leftarrow$  0
  fin si
   $\vec{F}_i \leftarrow (AS_i.\text{position} - AS.\text{position}) \times I/d$ 
   $\vec{F}.\text{ajouter}(\vec{F}_i)$ 
fin pour
Retourner  $\vec{F}$ 

```

Pour compléter la présentation de notre conception orientée agent, la partie suivante donne les spécificités du moteur de simulation MitoMAS et de la boucle constituant un pas de temps, c'est-à-dire une étape du cycle de vie des agents.

4.7.7 L'architecture du moteur de simulation

La figure 4.7.6 présente l'architecture générale du moteur de simulation ou *noyau*, c'est-à-dire les classes uniquement dédiées à la description des entités de la simulation (pour les classes déjà présentées, seul le nom apparaît). L'ensemble de la plateforme MitoMAS contient également un module de visualisation 3D et un module de communication pour mettre en place des simulations en mode client/serveur. Ces modules seront présentés par la suite.

► La classe World

La classe `World` définit le volume de simulation.

> Les attributs

Les différentes instances de `Grid` qui discrétisent l'espace sont référencées par l'attribut `collectionOfGrids` tandis que les différents `BioAgents` le sont par l'attribut

collectionOfBioAgents. Cette classe référence également l'ensemble des instances de ModelOfInteractions qui sont fournies avec la plateforme. Enfin deux attributs indiquent l'état d'avancement d'une simulation : numStep et simulatedTime qui correspondent respectivement au nombre de pas de la simulation et au temps total simulé⁸.

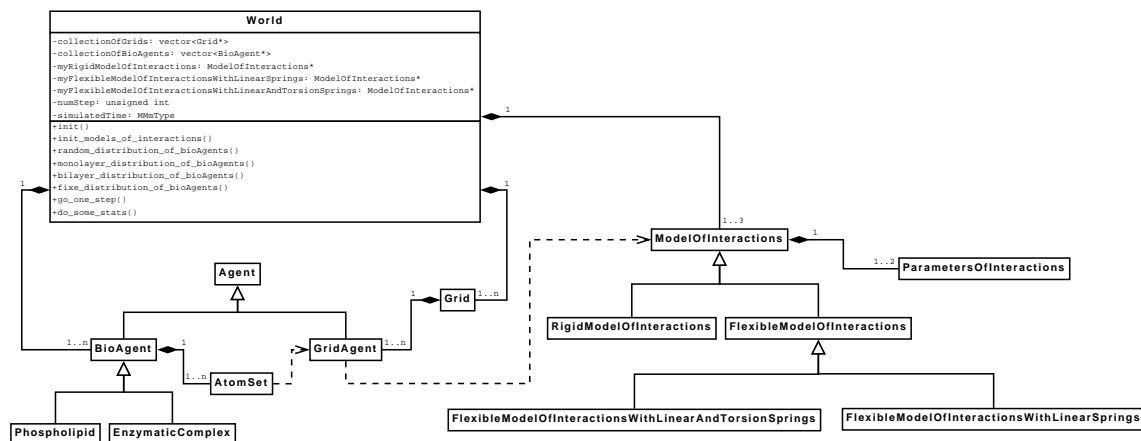


FIG. 4.7.6 – Diagramme de classes du noyau.

➤ Les comportements

Parmi les comportements de World, `init` crée les différentes instances de Grid, de BioAgent et de ModelOfInteractions. Il est possible d'initialiser une simulation de différentes manières :

- `random_distribution_of_bioAgents` positionne et oriente aléatoirement les BioAgents ;
- `monolayer_distribution_of_bioAgents` et `bilayer_distribution_of_bioAgents` positionnent les BioAgents respectivement en monocouche et en bicouche ;
- `fixe_distribution_of_bioAgents` permet un positionnement “manuel” de certains BioAgents, en vu par exemple de produire une sauvegarde qui servira de fichier d'initialisation spécifique.

Il est également possible de restaurer une précédente simulation à partir d'un fichier de sauvegarde qui dans ce cas initialisera directement la simulation. Ensuite le comportement `init_models_of_interactions` initialise les différents modèles d'interactions notamment en fonction du nombre de grains et des types de grain. Le comportement `go_one_step` correspond à une boucle de simulation et induit notamment les méthodes `life_cycle` des différents agents. Enfin, `do_some_stats` est le comportement qui permet de générer des données quantitatives sur le système simulé. Dans le chapitre 5, nous détaillerons pour les différents résultats l'ensemble des conditions d'initialisation qui ont été choisies.

Nous allons maintenant préciser l'algorithme de la simulation, autrement dit la succession des comportements déclenchés à chaque pas de simulation.

4.7.8 Le cycle de vie des agents : l'algorithme de la simulation

Après la phase d'initialisation, une simulation consiste à effectuer des cycles de comportements ponctués par les pas de temps. Cette suite d'instructions, donnée par l'algorithme 4.5, constitue la boucle principale du noyau de simulation.

Comme nous l'avons dit, dans un premier temps, tous les GridAgents de chaque instance de Grid exécutent leur comportement `life_cycle`, avec notamment la détermination du bon modèle d'interactions intramoléculaires à attribuer aux BioAgents. Les GridAgents sont réciproquement associés aux BioAgents et aux AtomSets qu'ils contiennent.

Puis les BioAgents s'activent et exécutent leur comportement `life_cycle`. Une **première passe**

⁸Étant donné que le pas de temps est variable, il n'est pas possible de déduire le temps simulé du nombre de pas de la simulation

sur l'ensemble des **BioAgents** consiste à prévoir le prochain état de chaque **BioAgent**, autrement dit à calculer les forces mises en jeux. Après avoir défini le pas de temps τ correct à utiliser lors des calculs d'intégration, la **seconde passe** sur l'ensemble des **BioAgents** génère le prochain état de chaque **BioAgent** en utilisant notamment τ pour le calcul du prochain état avant de déclencher l'échange (*swap*) de l'état actuel par le prochain état qui vient d'être calculé.

Il ne reste plus qu'à incrémenter le temps simulé et le compteur de pas de simulation. Ces deux variables qui peuvent éventuellement être utilisées pour paramétrer la durée de la simulation.

Algorithme 4.5 Boucle principale de la simulation.

ENTRÉES: {Grids} l'ensemble des grilles partitionnant le volume simulé,
 {BioAgents} l'ensemble de BioAgents
SORTIES: rien

```

boucler
  pour tout  $G_i \in \{\text{Grids}\}$  faire
    pour tout  $GA_i \subset G_i$  faire
       $GA_i.\text{go-one-step}()$ 
      attribution à  $GA_i$  de son modèle d'interactions intramoléculaires
      association  $GA/BA$  et  $GA/AS$ 
    fin pour
  fin pour
  intensitee-maximale-des-forces  $\leftarrow 0$ 
  pour tout  $BA_i \in \{\text{BioAgents}\}$  faire
    un-GA  $\leftarrow BA_i.GA_{host}$ 
    un-modele  $\leftarrow$  un-GA.modele-intramoleculaire
    un-modele.prevoir-prochain-etat( $BA_i$ )a
  fin pour
  calcul du prochain pas de temps variable  $\tau$  à partir
  pour tout  $BA_i \in \{\text{BioAgents}\}$  faire
    un-GA  $\leftarrow BA_i.GA_{host}$ 
    un-modele  $\leftarrow$  un-GA.modele-intramoleculaire
    un-modele.calculer-prochain-etat( $BA_i, \tau$ )
    un-GA.swap-etats()
  fin pour
   $\tau_{total}.\text{ajouter}(\tau)$ b
  nb-boucles.ajouter(1)c
fin boucle
  Retourner rien
  
```

^adéclenche notamment l'algorithme 4.4 et modifie intensitee-maximale-des-forces

^bce décompte du temps simulé permet d'arrêter au bout d'un temps donné

^cce décompte des boucles effectuées permet d'arrêter au bout d'un nombre de boucles donné

Après avoir présenté la conception orienté agent du noyau, notamment son architecture et sa boucle principale de simulation, la partie suivante s'attache à décrire les fonctionnalités des modules qui accompagnent le noyau.

4.8 Conception du simulateur

Nous venons de présenter le noyau que nous avons développé spécifiquement pour MitoMAS. Ceci ne signifie pas que nous n'avons pas réutilisé des bibliothèques dédiées à des tâches spécifiques. Nous avons déjà parlé de la bibliothèque boost qui nous a fourni différentes fonctionnalités autour des nombres aléatoires. Nous avons également utilisé dans cette bibliothèque la fonction de sérialisation des données. Cela nous a permis de mettre en place facilement et efficacement⁹ des mécanismes de sauvegarde des simulations sous la forme txt ou xml.

Nous allons maintenant présenter rapidement comment nous avons implémenté le module de visualisation 3D à partir de 2 bibliothèques : VTK¹⁰ (*Visualization ToolKit*) et GLFW¹¹ (*GL FrameWork*).

⁹Par exemple, la sérialisation de boost propose un système de version pour prendre en compte l'évolution des modèles de nos agents.

¹⁰<http://www.kitware.com/products/vtktextbook.html>

¹¹<http://glfw.sourceforge.net/GLFWUsersGuide26.pdf>

4.8.1 La visualisation

La visualisation est un module¹² (Fig.4.8.1(a)) à part entière tout à fait détachable (à la compilation) du noyau, ce qui permet par exemple de lancer le simulateur en mode serveur, c'est-à-dire sans aucun rendu 3D ou en mode graphique. Historiquement la plateforme MitoMAS a été développée avec GLFW qui, bien qu'offrant moins de fonctionnalités que VTK, reste facile à inclure dans une archive. Dans un deuxième temps, nous avons enrichi la plateforme d'une autre bibliothèque de rendu : la bibliothèque VTK. VTK propose plus de fonctionnalités en termes d'interactivité avec la visualisation, comme par exemple la possibilité de changer la résolution du rendu, ce qui peut s'avérer très utile lorsqu'il y a un grand nombre d'objets à visualiser. La bibliothèque VTK est plus délicate à intégrer dans une distribution, c'est la raison pour laquelle nous avons laissé la possibilité de choisir l'une ou l'autre bibliothèque à la compilation de la plateforme MitoMAS.

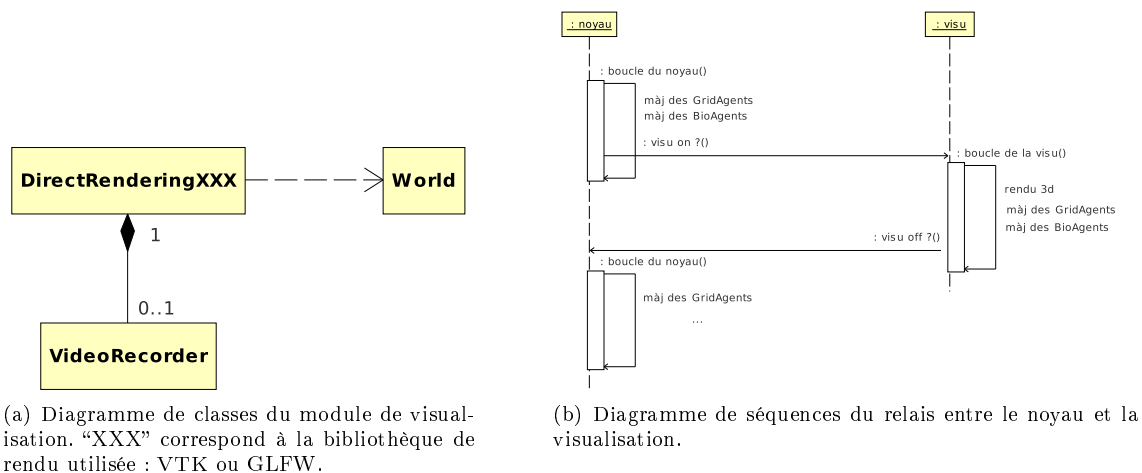


FIG. 4.8.1 – Le module de visualisation.

Comme le montre la figure 4.8.1(b), si le mode graphique est activé, alors les pas de simulation sont déclenchés par la boucle de rendu de cette visualisation, ce qui évitera des problèmes d'accès concurrents. A noter : il ne suffit pas de compiler avec les options `render RENDERING="XXX"` pour bénéficier d'une fenêtre de rendu, il faut encore préciser en lançant le programme, que l'on souhaite avoir une vue sur le système simulé (simplement `$/MMm -visu`).

4.8.2 Paramétrage d'une simulation

Par défaut, une compilation sans option produit un programme qui se lance en mode console, les sorties étant, par voie de fait, des sorties console. Dans ce mode, les simulations pourront être paramétrées par un ensemble de valeurs passées en ligne de commande au moment de l'exécution. Comme nous l'avons dit, il est également possible de recharger la sauvegarde d'une précédente simulation.

Si on compile en mode graphique `$make render RENDERING="VTK"` ou `$make render RENDERING="GLFW"`, il sera également possible de produire une vidéo du rendu de la simulation par l'option `-video` à l'exécution. L'animation est obtenue par ajout successif d'images capturées par le moteur de rendu, le processus étant conditionné par le temps simulé et non le nombre de pas de simulation effectués. En d'autres termes, une nouvelle image n'est pas systématiquement créée à chaque pas de simulation, mais tous les τ_{video} . Par exemple, une visualisation directe de l'autoassemblage de lipides en micelles permet de se rendre compte des accélérations que peut prendre la simulation si les interactions entre grains sont faibles. Pendant ces dynamiques rapides, il faut tout simplement enregistrer plus d'image pour obtenir une temporalisation correcte de la dynamique observée. Le chapitre 5 nous donnera l'occasion de donner plus de détail sur les différentes simulations observées.

¹²Le répertoire des sources : `mmm_gui`.

4.8.3 Le module de communication

Les simulations multi-agents sont de fait des simulations coûteuses en temps de calcul. Elles paraissent donc être des candidats naturels pour des exécutions distribuées. Un cas d'utilisation d'une telle distribution est le découplage des calculs effectués par le noyau et de la visualisation du résultat de ces calculs, par exemple avec un rendu 3D de la scène. Dans cet objectif, nous avons ajouté un module de communication à base de sockets¹³ qui permet ce découplage. Cette exécution distribuée n'est pas réservée à la visualisation est peut être aussi utilisée pour réaliser des statistiques ponctuelles au cours des simulations. D'une façon générale, nous avons implémenté un mécanisme de sauvegarde de l'état courant du système qui peut être utilisé par différents modules indépendants du calcul des simulations. La mise en place de sockets conduit à la production de deux exécutables, l'un en mode serveur, l'autre en mode client. En d'autres termes, le même code source, compilé coté serveur en mode `noyau`, produit un simulateur et, compilé coté client en mode `render`, génère un exécutable augmenté d'un modeleur 3D pour visualiser le système simulé.

4.9 En résumé : le modèle informatique

Nous venons de présenter un exemple d'architecture multi-agents spécifiquement conçue pour la modélisation moléculaire et intégrant les moteurs physiques nécessaires aux calculs des interactions entre des molécules de type phospholipide. Nous avons vu que ce modèle pouvait être étendu à la prise en compte soit d'autres types de phospholipide, par exemple en modifiant le nombre de grains par molécule, soit de complexes enzymatiques. Les modèles d'interactions peuvent aussi être étendus en ajoutant d'autres sous-classes que celles qui sont proposées actuellement par la plateforme MitoMAS.

Nous allons maintenant présenter l'ensemble des résultats que nous avons pu obtenir avec cette plateforme pour la simulation de la membrane interne mitochondriale.

¹³des objets appelés socket qui représentent des interfaces de programmation pour la communication entre les processus.

Chapitre 5

Les expérimentations *in-silico*

Ce chapitre est consacré à l'exposé des simulations que nous avons réalisées et à l'analyse des résultats.

5.1 Plan d'expérience *in-silico*

En ce qui concerne la modélisation de la membrane interne mitochondriale, il nous a semblé raisonnable de débiter les expérimentations *in-silico* par des systèmes simples, puis de les complexifier pour se rapprocher des membranes biologiques. Suivant cette complexification graduelle des systèmes simulés, voici notre **plan d'expérience *in-silico*** :

- simuler des systèmes composés d'un seul type de lipides, en l'occurrence des molécules de dimyristoyl-phosphatidylcholine (DMPC), dans différentes conditions initiales : initialisation aléatoire, en monocouche et en bicouche ;
- simuler des mixtures composées d'au moins deux types de lipides différenciés soit par la longueur des queues, soit par le nombre de queues par lipide, pour à terme constituer une membrane mitochondriale selon les proportions issues des données expérimentales ;
- simuler des systèmes lipides/peptides membranaires, pour à terme, intégrer les complexes de la chaîne respiratoire en commençant par le dimère du complexe IV.

Outre une **évaluation qualitative** passant par une appréciation visuelle, des **données quantitatives** sont produites pour évaluer les simulations par rapport aux données issues de dynamique moléculaire ou d'expérimentations.

Dans ce but, un certain nombre de statistiques (ou observables¹) seront présentées et illustrées de quelques simulations tout au long des sections suivantes. Les simulations permettent d'étudier des phénomènes sur des temps d'observation de l'ordre du centième de microseconde (10^{-8} s), ce qui veut dire que certaines observables sont en mesure d'apprécier une **dynamique du système**, et non plus uniquement la distribution statique de ses éléments. Dans certains cas, l'évolution des systèmes tels que les bicouches nécessitera même de définir une nouvelle observable capable de suivre les changements importants des structures lipidiques.

¹Une observable est une opération de mesure, c'est-à-dire l'acquisition de la valeur ou d'un intervalle de valeurs d'un paramètre physique, ou plus généralement d'une information sur un système physique.

5.2 Une membrane générique

5.2.1 Implémentation d'un phospholipide

Comme nous l'avons présenté en section 4.1.2, notre premier modèle de lipide **trimérique rectiligne rigide** est composé de **3 grains**.

A partir des données que nous avons analysées pour le DMPC² (cf tableau 1.3.3 du chapitre 1) nous avons défini une taille de 30\AA pour le lipide dont chaque grain a un rayon $\sim 5\text{\AA}$.

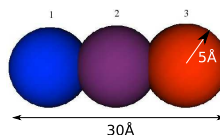


FIG. 5.2.1 – Une molécule de DMPC modélisée un trimère linéaire rigide.

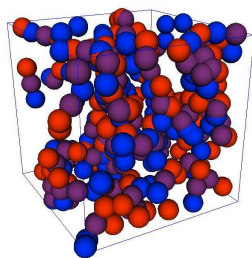
5.2.2 Différentes configurations initiales du système

Dans tout processus de simulation, il est toujours intéressant de tester l'influence de la configuration du système. En biologie humide, on connaît différentes structures possibles pour un ensemble de phospholipides : micelles, monocouche, bicouche... Nous avons voulu savoir dans quelles conditions nous pouvions obtenir la simulation de ces structures.

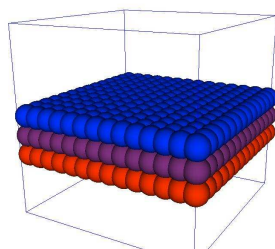
Tous les exemples qui suivent présentent le même volume de simulation de $100 \times 100 \times 100 \text{\AA}^3$.

► Initialisation aléatoire

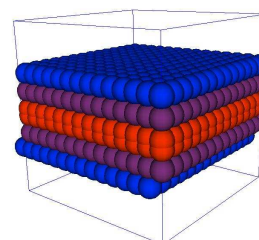
La figure 5.2.2(a) montre une **distribution initiale aléatoire** de 100 phospholipides où les positions et les orientations sont toutes deux tirées aléatoirement. La densité des molécules est dans ce cas ~ 0.1 lipide par nm^3 . L'absence d'organisation apparente nous laisse penser que les paramètres utilisés n'induisent pas de biais évidents.



(a) 100 lipides - distribution aléatoire.



(b) 200 lipides en monocouche.



(c) 400 lipides en bicouche.

FIG. 5.2.2 – Différentes initialisations du volume de simulation.

► Distribution régulière en monocouche et en bicouche

La figure 5.2.2(b) montre une initialisation de la simulation avec une population de 200 phospholipides régulièrement disposés en **monocouche**. Cette situation pourrait correspondre à la disposition d'une couche de phospholipides placée à l'interface de deux milieux. A noter que ces 200 phospholipides répartis sur une coupe de $100 \times 100 \text{\AA}^2$ du volume de simulation ont par conséquent une aire par lipide de 50\AA^2 , ce qui est cohérent avec les données expérimentales. Cette aire par lipide sera expliquée par la suite mais cette observable est à mettre en relation avec la notion de densité, c'est-à-dire en moyenne l'aire réservée à un seul lipide. Enfin l'image 5.2.2(c) est

²1,2-dimyristoylphosphatidylcholine (C14 :0)

une population de 400 phospholipides répartis régulièrement sous la forme d'une **bicouche**³. Les phospholipides bénéficient également d'une aire par lipide de 50\AA^2 , ce système étant simplement le double du précédent.

Notons que les distributions sont extrêmement régulières ce qui ne correspond pas vraiment à des situations réalistes, que ce soit *in-vitro* avec des systèmes artificiels ou *in-vivo* avec les membranes biologiques.

Pour remédier au problème de distribution trop régulière, une solution consiste à lancer une simulation uniquement avec un mouvement brownien restreint au plan (XY) parallèle à la monocouche ou à la bicouche puis à faire une sauvegarde du système au bout d'un temps simulé suffisamment long ou un nombre suffisant de pas de simulation⁴. Après 1000 pas de temps, les temps simulés sont de l'ordre de la dizaine de nanosecondes, et les configurations sont bien plus satisfaisantes. Les figures 5.2.3(a) et 5.2.3(b) correspondent respectivement à une configuration monocouche⁵ et bicouche⁶, avec dans les deux cas une aire par lipide qui, en moyenne, reste égale à 50\AA^2 . Lors de ce brassage uniquement suivant le plan (XY), il n'y a pas de passage de lipides d'une monocouche à une autre (*flip-flop*) et chaque feuillet conserve donc le même nombre de lipides.

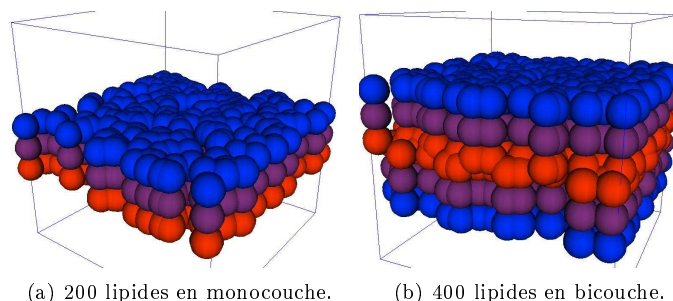


FIG. 5.2.3 – Distributions mélangées.

Tout en ne considérant des populations de phospholipides constituées que d'un seul type de phospholipides, en l'occurrence des molécules de DMPC, ces trois configurations initiales ouvrent les champs à de nombreuses expérimentations *in-silico*.

Les propriétés structurales dépendent de plusieurs paramètres comme par exemple la morphologie des lipides, la composition de la bicouche, *ie* si la bicouche contient un ou plusieurs types de lipides, ou d'autres molécules comme du cholestérol ou des protéines, ou de l'état thermodynamique du système comme la température. Certaines observables peuvent être mesurées expérimentalement sur des modèles simplifiés et calculées lors des simulations. Ces quantités peuvent alors servir de valeur référence pour calibrer un modèle et tester son efficacité ou peuvent être un résultat à part entière et donner de nouveaux éléments de compréhension du comportement des lipides.

5.2.3 Initialisation aléatoire des lipides

Le processus d'initialisation instancie les positions et les orientations des lipides de manière aléatoire (voir Fig.5.2.4(b) et Fig.5.2.5(b)). Les lipides sont donc répartis uniformément dans tout le volume, pour ces simulations nous analyserons l'effet des variations de la densité des molécules dans le volume.

³commande : `$/MMm -visu -nba 400 -distrib b`

⁴la visualisation n'est pas utile dans ce cas : `$/MMm -nba 200 -distrib m -ns 1000 -save`

⁵commande : `$/MMm -visu -i output/MMm_save_DATE_HEURE.txt`

⁶même commande

► Les résultats de simulation

La figure 5.2.4 montre l'évolution d'un système initialisé avec 1000 lipides dans un volume de $2880nm^3$, ce qui correspond à une concentration de 0.34 lipide par nm^3 . Ces valeurs ont été choisies arbitrairement puisque ce type de situation ne se rencontre pas dans la nature. Bien évidemment, cela peut par contre correspondre à une expérience effectuée dans un tube à essai dans lequel on aurait préalablement mélangé les lipides dans un solvant.

Pour la simulation nous avons utilisé une distance de coupure (*cutoff*) est de 15\AA . Après un temps simulé de $8.62636ns$, les lipides se sont apparemment organisés pour former des micelles aplaties recourbées sur elles-mêmes (Fig.5.2.4(b)) minimisant ainsi l'exposition des queues au solvant. Comme nous utilisons le modèle défini au chapitre précédent, le solvant est implicitement modélisé aux travers des potentiels par paire entre grains. Ces potentiels poussent les lipides à adopter cette configuration minimisant l'énergie potentielle globale. Ce comportement correspond au comportement généralement pour des lipides dans un solvant. A priori, les potentiels utilisés semblent donc fournir une modélisation tout à fait acceptable du solvant.

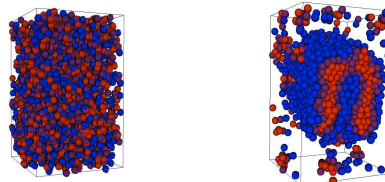
La figure 5.2.5 présente l'évolution d'un système de 500 lipides aléatoirement distribués dans un espace de $200x200x200\text{\AA}^3$ soit $8000nm^3$, ce qui correspond à une concentration de 0.0625 lipide par nm^3 , soit une densité cinq fois inférieure à la concentration précédente.

Malgré cette faible densité, on observe encore la formation d'agrégats sphéroïdaux minimisant l'exposition des queues au solvant. La dynamique du système montre que ces agrégats, que l'on peut assimiler à des micelles, se déplacent tout en conservant leur cohésion, et qu'elles sont capables de s'agréger les unes aux autres pour former des structures plus importantes. Au bout d'un certain temps simulé, on obtient des structures d'assez grande taille qui s'apparentent plus à des portions de monocouche qu'à des micelles sphériques.

Ces simulations montrent que des systèmes de lipides aléatoirement répartis dans l'espace sont capables de produire des structures limitant l'exposition des queues des lipides au solvant, même si celui-ci est implicite. En comparant les figures 5.2.4(c) et 5.2.5(c), on note qu'en changeant la concentration initiale en lipide, on obtient des structures relativement compactes dans un cas et moins connectées dans l'autre.

En terme de variation de paramètres, l'utilisation d'une distance de coupure plus grands ne fait qu'accélérer légèrement les dynamiques (la vitesse d'apparition des structures), mais ne modifie pas les résultats qualitatifs en terme d'organisation des lipides. Par contre, en terme de performance, la distance de coupure influe beaucoup. Par exemple, avec une distance de coupure de 15\AA , le second volume de simulation ($200x200x200\text{\AA}^3$) contenant 500 lipides est partitionné en 2197 *GridAgents*. En moyenne 30 000 distances sont calculées peu après l'initialisation pour atteindre près de 100 000 distances dans la configuration montrée en figure 5.2.5(c). Avec une distance de coupure de 25\AA , le volume est partitionné en 512 *GridAgents* et le nombre moyen de distances calculées augmente

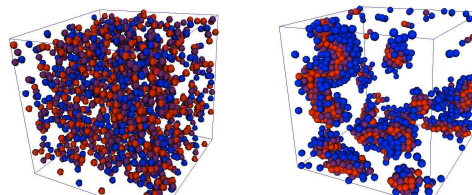
volume (\AA^3)	nb BA	cutoff (\AA)
120x120x200	1000	15



(b) Initialisation aléatoire. (c) Après $8.62636e-09s$ simulée.

FIG. 5.2.4 – Initialisation aléatoire des lipides.

volume (\AA^3)	nb BA	cutoff (\AA)
200x200x200	500	15



(b) Initialisation aléatoire. (c) Après $8.08364e-09s$ simulée.

FIG. 5.2.5 – Initialisation aléatoire des lipides.

à 120 000 après l'initialisation pour atteindre 236 000 à l'état stable. Il faut noter, qu'en plus de l'augmentation du nombre des distances à calculer par pas de simulation, les temps d'intégration sont sensiblement diminués. En d'autres termes, l'efficacité des simulations est clairement conditionnée par ce paramètre. On peut donc conclure que pour ce type de simulation à initialisation aléatoire qu'il est tout à fait possible d'utiliser pour la distance de coupure les valeurs minimales de l'intervalle que nous avons décrit à la section 4.2.2 sans pour autant altérer les résultats qualitatifs des simulations.

5.2.4 Initialisation des lipides en monocouche

Il est admis qu'en général les lipides s'organisent en monocouches ou en bicouches, il n'est donc pas réellement indispensable de partir d'une initialisation aléatoire pour étudier la dynamique de ces assemblages de lipides. Toutefois, certaines précautions doivent être prises sur le choix de l'état initial. En effet, commencer à calculer des interactions entre des lipides répartis régulièrement sur une monocouche ou une bicouche est assez peu réaliste et ne correspond pas aux assemblages lipidiques (naturels ou de synthèse). Nous avons donc choisi de procéder en deux étapes. Premièrement, les lipides sont disposés en couche régulière (cf Fig.5.2.2(b)) puis une simulation métant en oeuvre uniquement un mouvement brownien restreint au plan (XY) parallèle à la monocouche permet de générer une configuration plus réaliste (cf Fig.5.2.3(a)). Après un certain nombre de pas de temps (en général quelques centaines), nous réalisons une sauvegarde du système. C'est cette configuration qui sera alors utilisée comme état initial de la simulation réaliste, c'est-à-dire la simulation appliquant l'intégralité de notre modèle aux mouvements des molécules.

► Les observables

Plusieurs statistiques permettent d'apprécier l'évolution d'un tel système plan.

► La diffusion

Même si la structure globale semble fixe, les lipides diffusent librement dans le feuillet auquel ils appartiennent. On peut distinguer les mouvements des molécules par diffusion latérale, par diffusion rotationnelle et par passage d'un feuillet à l'autre dans le cas des bicouches. Ces phénomènes de diffusion sont observés expérimentalement mais également lors de simulations de dynamique moléculaire.

De tous ces types de diffusion, nous n'allons nous intéresser qu'à la **diffusion latérale** des lipides. Cette dispersion D des lipides mesurée au temps τ sera évaluée par l'expression Eq.(5.2.1) où N_{grains} est le nombre total des grains concernés, $\vec{r}_{i_{init}}$ et \vec{r}_i sont respectivement la position initiale du grain i et celle au temps τ .

$$D(\tau) = \frac{\sum_i \langle \vec{r}_i - \vec{r}_{i_{init}} \rangle^2}{N_{grains} \tau} \quad (5.2.1)$$

► Le paramètre d'ordre

Le paramètre d'ordre du lipide est un moyen de **quantifier l'organisation des chaînes lipidiques**. On peut expérimentalement mesurer cette observable par la "*deuterium substitution NMR spectroscopy*". Il permet notamment d'apprécier l'état d'un feuillet : phase gel ou phase fluide. Issue de la dynamique moléculaire, cette observable est utilisable pour les modèles à grains. L'expression (5.2.2) ne concerne qu'un seul lipide. L'angle ϕ est l'angle entre le vecteur formé par deux grains et la normale du plan de la bicouche avec \vec{n} le vecteur unité normal et $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$ le vecteur entre les grains.

$$S = \frac{1}{2} \langle 3 \cos^2 \phi - 1 \rangle \quad (5.2.2)$$

avec $\cos \phi = \frac{\vec{r}_{ij} \cdot \vec{n}}{r_{ij}}$

Le paramètre d'ordre correspond à la moyenne de la forme (5.2.2) sur l'ensemble des lipides du système. Il a une valeur égale à 1 si \vec{r}_{ij} est en moyenne parallèle à la normale de la bicouche, 0 si l'orientation est aléatoire et -0.5 si ce vecteur est en moyenne parallèle au plan de la bicouche. Aussi cette observable, sur l'ensemble de la molécule ou sur des régions de celle-ci, peut être mise en corrélation avec la température pour, par exemple, caractériser une transition de phase entre une phase gel, où les chaînes lipidiques sont ordonnées, et une phase liquide où les queues lipidiques deviennent désordonnées (Sintes et al. [SB97, SB98b]).

➤ L'aire par lipide

L'aire par lipide projetée sur le plan de la bicouche, A_p , peut être expérimentalement estimé par rayon-X ou par dispersion de neutron (neutron scattering experiments) [NZTN⁺96], ou, indirectement, à partir du profil des paramètres d'ordre des lipides [Nag93].

Pour les monocouches simulées, l'aire par lipide est souvent calculée en divisant l'aire totale de la couche par le nombre de lipides dans la monocouche. Il est néanmoins important de noter que si la monocouche exhibe de grandes ondulations, la valeur ainsi calculée de l'aire par lipide est sous-estimée [ISL05].

Plusieurs études relatives à cette observable suggèrent différentes valeurs caractéristiques de la cohésion de la couche lipidique (de l'ordre de 50\AA^2). O.Farago [Far03] étudie la diffusion latérale des lipides en fonction de leur densité et obtient une valeur $A_p \sim (28.125)^2\text{\AA}^2$ pour la transition de phase gel/fluide. D'autres auteurs ont retenu des valeurs légèrement différentes : 57 à 68\AA^2 [BPB05], 59.6\AA^2 pour le DMPC (la dimyristoyl-phosphatidylcholine) à 72.5\AA^2 pour le DOPC (la dioleoyl-phosphatidylcholine) [NTN] ou encore de 46\AA^2 pour DPPC à 72\AA^2 pour le DOPC [MM04].

L'aire par lipide apparaît comme une statistique communément employée pour apprécier les résultats d'une simulation, de part sa sensibilité à certains paramètres tels que la rigidité ou la longueur des lipides. Néanmoins, comme nous allons le voir par la suite, les monocouches (et les bicouches) ne restent pas des systèmes plans. C'est pourquoi nous proposons d'associer à l'aire par lipide la moyenne des distances au plus proche voisin, et ce entre chaque type de grains. Ainsi, nous serons en mesure d'apprécier une densité de surface notamment des grains de type *tête* pour les états de la membrane obtenus en fin de simulation.

➤ Les résultats de simulation

Une première série de simulations permet de tester l'implémentation de l'observable **diffusion latérale**.

Nous avons voulu tester si en partant d'une monocouche de 100 lipides rassemblés au centre du volume de simulation, la seule application du mouvement brownien nous permet d'observer une diffusion correcte des lipides dans le plan. Les trois schémas de la figure 5.2.6 montrent depuis l'étape d'initialisation l'évolution du système au cours du temps.

A noter qu'en utilisant l'expression (5.2.1) pour calculer cette diffusion latérale, il faut se référer à la densité de probabilité radiale $P_D^r(r, \tau)$ qui s'exprime à partir de la densité de probabilité volumique $P_D^v(r, \tau)$ suivant Eq.(5.2.3)

$$P_D^r(r, \tau) = 4\pi r^2 P_D^v(r, \tau) \quad (5.2.3)$$

Les diagrammes associés aux simulations nous montre l'évolution des valeurs de diffusion latérale pour l'ensemble de la population des

volume (\AA^3)	nb BA	cutoff (\AA)
54x54x200	100	-

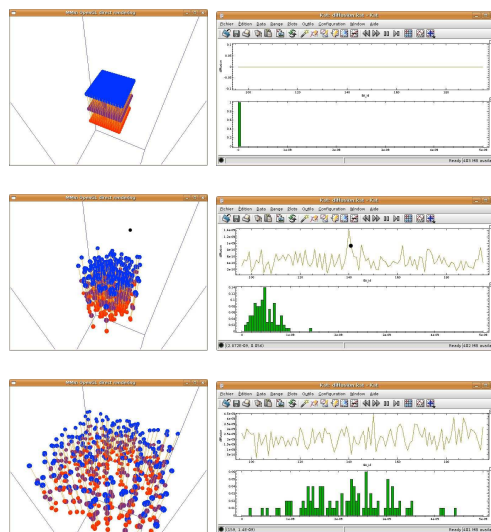


FIG. 5.2.6 – Diffusion latérale des lipides.

lipides, ce qui est le reflet de l'évolution du système.

Une deuxième série de simulations permet de tester l'implémentation de l'observable **paramètre d'ordre**. Le système de simulation est identique au précédent en terme de densité de lipides et de volume, mais la dynamique utilisée tient compte des interactions entre les lipides.

On observe un repli progressif de la monocouche (fig.5.2.7) jusqu'à former une micelle aplatie. Les histogrammes de droite centrés sur zéro indiquent la proportion des lipides adoptant un angle donné. Plus il y a de lipides qui s'écartent de la normale au plan (XY) de la monocouche, plus l'histogramme s'aplatit. A la vue de ces histogrammes, on peut constater que le paramètre d'ordre rend bien compte de l'organisation des lipides les uns par rapport aux autres.

Une troisième série de simulations permet de tester l'implémentation de l'observable **aire par lipide** suivant le calcul proposé précédemment, basé sur la distance au plus proche voisin (distance ppv). Pour proposer une équivalence entre les deux statistiques, des simulations uniquement avec mouvement brownien suivant le plan (XY) ont été faites avec des aires par lipides de 30\AA^2 à 100\AA^2 par pas de 10\AA^2 . Nous considérerons qu'une distribution aléatoire des lipides produit une distance ppv minimale. D'autre part, la racine carrée de l'aire par lipide donne directement une distance théorique maximale du plus proche voisin. Les valeurs produites par les simulations suivantes seront donc bornées au minimum par la valeur issue d'un positionnement aléatoire et au maximum par la valeur théorique (cf courbes de la figure 5.2.8). A la lecture des points positionnés sur le graphique, nous pouvons quantifier l'augmentation de la surface d'un feuillet (monocouche ou bicouche) avec replis.

> Simulations libres

La figure 5.2.9 montre l'évolution d'un système initialisé avec 500 lipides dans un volume de $120 \times 120 \times 200\text{\AA}^3$, ce qui correspond à une aire par lipide $\sim 28.8\text{\AA}^2$. La distance de coupure est de 15\AA . Après un temps simulé de $62.7867ns$, la monocouche s'est repliée sur elle pour former une sorte de micelle aplatie (Fig.5.2.9(c)), minimisant toujours l'exposition des queues au solvant.

volume (\AA^3)	nb BA	cutoff (\AA)
54x54x200	100	15

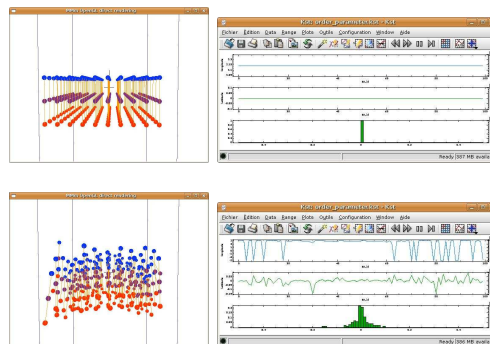


FIG. 5.2.7 – Paramètre d'ordre des lipides.

volume (\AA^3)	nb BA	cutoff (\AA)
AxAx200	1000	-

aire par lipide (\AA^2)	30	40	50	60	70	80	90	100
distance ppv théorique (\AA)	5.4	6.3	7.1	7.7	8.4	8.9	9.4	10
distance ppv simulée (\AA)	3.0	3.4	3.8	4.3	4.6	4.8	5.2	5.5

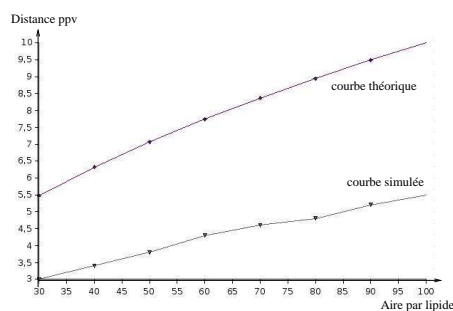


FIG. 5.2.8 – Aire par lipide et distances au plus proche voisin.

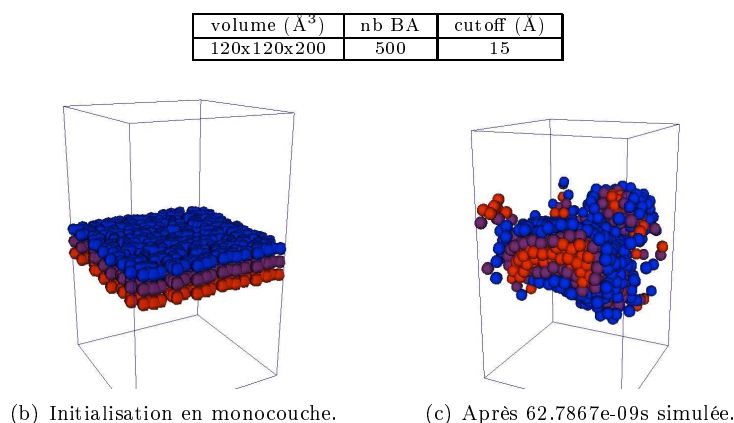


FIG. 5.2.9 – Une simulation avec initialisation en monocouche des lipides.

Le tableau 5.2.1 contient les résultats d'une série de simulations dans des conditions similaires (monocouche de 500 lipides avec une aire par lipide de 28.8\AA^2). Seule la distance de coupure varie et il est donc possible d'évaluer l'impact de ce paramètre sur l'organisation des lipides. Systématiquement, les systèmes convergent vers une micelle aplatie. Les chiffres ont été rapportés au même temps simulé de $5e^{-10}s$ pour simplifier les comparaisons. En supplément des données contenues dans ce tableau, des vidéos générées automatiquement permettent d'évaluer la dynamique du système.

distance cutoff	temps simulé	durée de la simulation	nbDist	nbStep	temps moyen simulé par step
25A	5e-10s	68h	651522/651522	50814	0.99e-14s
20A	5e-10s	33h20	431857/431857	44942	1.11e-14s
15A	5e-10s	7h50	153942/153942	29701	1.68e-14s
10A	5e-10s	3h50	67516/67516	33018	1.51e-14s
5A	5e-10s	0h28	2781/2781	8032	6.23e-14s

TAB. 5.2.1 – Effet de la distance de coupure (*cutoff*) sur des simulations de monocouche.

On observe qu'avec une distance de coupure trop restreinte, la cohésion de la monocouche n'est pas conservée. Par ailleurs, plus cette distance est grande, plus la pression exercée sur les lipides induit une dynamique rapide de repliement de la monocouche. Les temps de calcul, par rapport aux résultats qualitatifs suggèrent qu'il n'est pas nécessaire d'utiliser des distances de coupure trop grandes pour obtenir une structure convenable. Ce n'est en rien une validation de l'état final (micelle aplatie) mais seulement une observation concernant un paramètre qui influe beaucoup sur le temps des simulations.

5.2.5 Initialisation des lipides en bicouche

Comme précédemment énoncé, il est possible d'initialiser des simulations avec des systèmes de lipides préalablement disposés sous forme de bicouches. Les mêmes précautions concernant la disposition initiale des lipides sont assurées par une procédure en deux temps. A partir d'une initialisation en bicouche régulière (cf Fig.5.2.3(b)), un certain nombre de pas de simulation impliquant uniquement le mouvement brownien selon le plan (XY) permet d'obtenir une configuration plus acceptable (cf Fig.5.2.3(b)). Les sauvegardes de ces configurations nous serviront de points de départ pour des simulations plus réalistes.

► Les observables

➤ Le profil de densité

Tout comme le profil de distribution radiale, le profil de densité perpendiculaire au plan de la

bicouche renseigne sur la distribution relative des grains. La référence n'est plus un point, mais le plan (XY) séparant les deux feuillet.

Le calcul est simple, puisqu'il s'agit de projeter les positions des grains sur l'axe z. On obtient les profils des densités de probabilité pour chaque type de grains en fonction de la distance r au plan de la bicouche.

➤ L'épaisseur de la bicouche

L'épaisseur de la bicouche peut être expérimentalement estimée par des mesures de diffraction rayon X, plus exactement à partir de la distance entre les pics de diffraction des têtes des lipides dans les deux feuillet opposés de la bicouche.

Dans les simulations, cette observable peut être calculée comme la distance entre les deux pics observés dans le profil de densité perpendiculaire au plan de la bicouche.

➤ Les résultats de simulation

Une première série de simulations permet de tester l'implémentation de l'observable **profil de densité perpendiculaire au plan**. Dans la figure 5.2.10, les histogrammes à droite représentent les profils de densité perpendiculaires au plan mesurés sur les simulations de gauche. Ces histogrammes montrent la distribution des grains de part et d'autre du plan (XY) de la bicouche. L'état initial très régulier de la bicouche produit un histogramme qui a le mérite d'être didactique : tous les grains étant alignés sur les mêmes plans, il n'y a que 6 classes non nulles sur tout l'histogramme. Les couleurs des classes sont directement liées aux types de lipides représentés. Au cours de la simulation, les grains quittent les plans initiaux et on note une dispersion des types dans les classes adjacentes Remarque : cette observable n'est valide que pour un système plan (comme une bicouche) puisqu'on dénombre des projections sur l'axe z.

volume (\AA^3)	nb BA	cutoff (\AA)
54x54x200	2x200	15

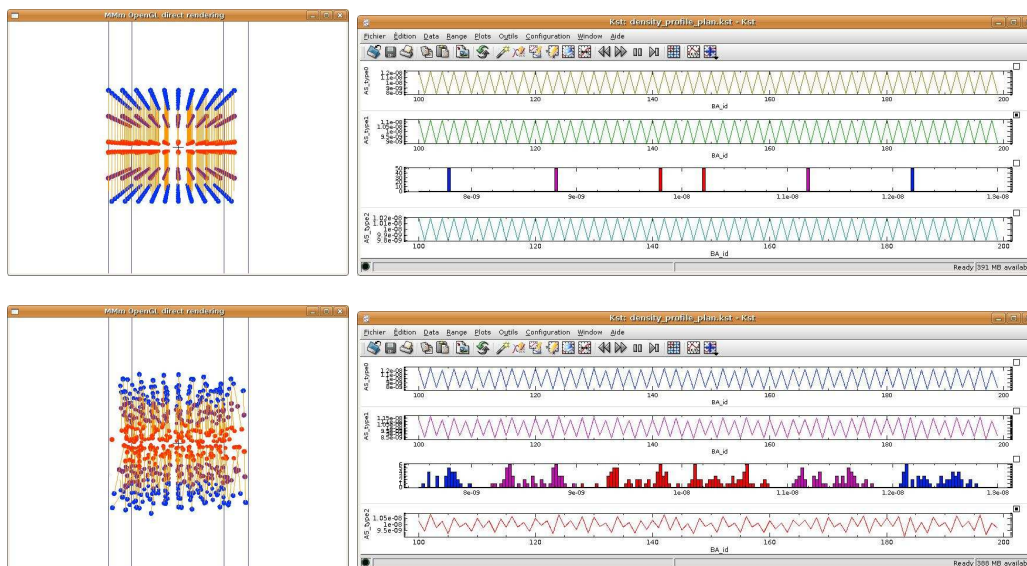


FIG. 5.2.10 – Profil de densité.

L'épaisseur de la bicouche peut, dans le cas des systèmes plans, être directement extraite du profil de densité. En l'occurrence, les bicouches sont initialisées avec un espace de 4\AA entre chaque feuillet, plus précisément entre chacun des deux plans contenant les grains de type **queue** de chaque feuillet. Il s'avère que dans les premières nanosecondes de simulation, les feuillet semblent maintenir cet écart. Il n'y a pas d'inter-pénétration des lipides puisqu'il est toujours possible de distinguer deux cloches de distribution de part et d'autre du plan ($z=0$), mais les grains de type **queue** se

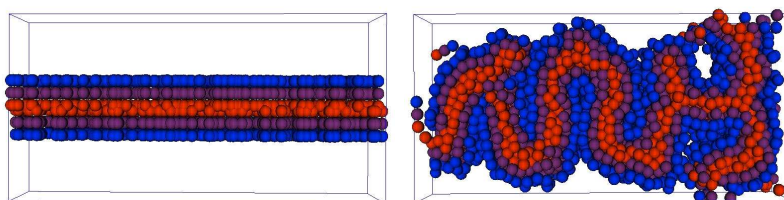
trouvent être presque accolés les uns aux autres entre les deux feuillets.

➤ Simulations libres

Cette partie regroupe les résultats de différentes simulations de bicouche. En jouant sur certains paramètres comme la distance de coupure ou en changeant les conditions initiales de densité de lipides par exemple, on peut déduire un certain nombre de points intéressants. Ces simulations sont pour la plupart réalisées dans un volume de $50 \times 300 \times 150 \text{Å}^3$. Ce volume relativement mince facilite l'observation des ondulations de la bicouche. Chacune de ces simulations nécessite plusieurs jours de calcul sur une machine équipée d'un pentium IV 3Ghz 1MoRAM.

La figure 5.2.11 montre l'évolution d'un système de 1040 lipides répartis sur les deux feuillets. Les interactions intermoléculaires sont tronquées à une distance de coupure de 10Å . A l'initialisation, l'aire par lipide est $\sim 28,8 \text{Å}^2$, ce qui correspond à une forte densité de molécules. La bicouche conserve sa cohérence tout en évoluant vers un état moins contraint arborant de nombreux replis pour augmenter sa surface (Fig.5.2.11(c)).

volume (Å^3)	nb BA	cutoff (Å)
$50 \times 300 \times 150$	2×520	10



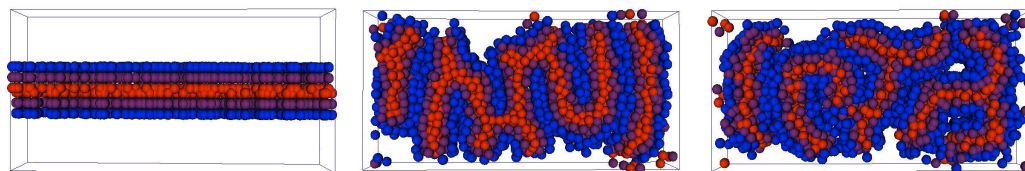
(b) Initialisation ($A_p = 28,8 \text{Å}^2$).

(c) Après 1.173×10^{-9} s simulée.

FIG. 5.2.11 – Bicouche homogène : simulation 1.

La figure 5.2.12 montre l'évolution d'un système dans des conditions identiques à celui présenté en Fig.5.2.11. On illustre juste ici la reproductibilité des simulations en terme de structures qualitatives résultantes, même si le mouvement brownien induit inévitablement des différences dans le détail de ces structures (qui ne sont pas superposables).

volume (Å^3)	nb BA	cutoff (Å)
$50 \times 300 \times 150$	2×520	10



(b) Initialisations ($A_p = 28,8 \text{Å}^2$).

(c) Simulation 2 après 2.8×10^{-9} s simulée.

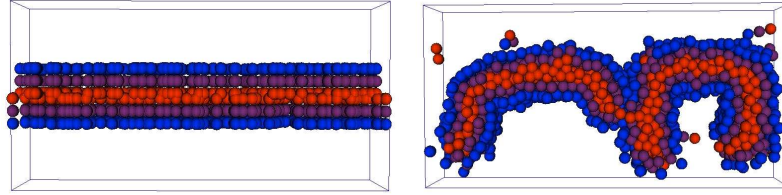
(d) Simulation 3 après 3.78×10^{-9} s simulée.

FIG. 5.2.12 – Bicouche homogène : 2 autres exemples de simulation.

La figure 5.2.13 montre l'évolution d'un système qui ne diffère du précédent que par la densité initiale de la bicouche. En effet, les 600 lipides répartis sur deux feuillets dans un volume de $50 \times 300 \times 150 \text{Å}^3$ ont une aire par lipide 50Å^2 . Toujours contrainte, la bicouche évolue pour adopter

de nouveau des replis augmentant sa surface. Néanmoins, le nombre et la taille de ces replis sont moins importants. Ces résultats semblent assez cohérents puisque plus les lipides sont initialement comprimés les uns contre les autres (ce qui se traduit par une aire par lipide moindre), plus la bicouche cherchera à se déployer pour augmenter sa surface.

volume (\AA^3)	nb BA	cutoff (\AA)
50x300x150	2x300	10

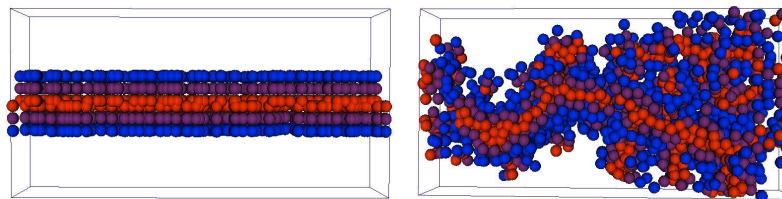
(b) Initialisation ($A_p = 50\text{\AA}^2$).

(c) Après 2.943e-9s simulée.

FIG. 5.2.13 – Bicouche homogène : influence de l'aire par lipide.

La figure 5.2.14 montre l'évolution d'un système identique au précédent (Fig.5.2.13), la seule différence réside dans la distance de coupure, ici fixée à 25\AA . La bicouche se courbe comme précédemment mais ne conserve pas sa cohésion. Dans un premier temps, ce phénomène nous a induit en erreur. Nous pensons que plus la distance de coupure est grande, plus la pression sur les lipides est grande, ce qui a pour effet de rompre la cohésion des structures. En optant pour une visualisation de meilleure qualité (VTK), nous avons pu observer l'origine de la dislocation de la membrane. L'erreur provenait d'un biais de modélisation de l'espace torique : le volume de simulation a une largeur de 50\AA alors que $r_c = 25\text{\AA}$. En d'autres termes, des phénomènes de prises en compte multiples d'un même voisin déséquilibraient les pressions exercées sur les lipides. Avec un volume corrigé de $75\text{x}300\text{x}150\text{\AA}^3$ et initialisé de la même manière, la membrane conserve sa cohésion.

volume (\AA^3)	nb BA	cutoff (\AA)
50x300x150	2x300	25

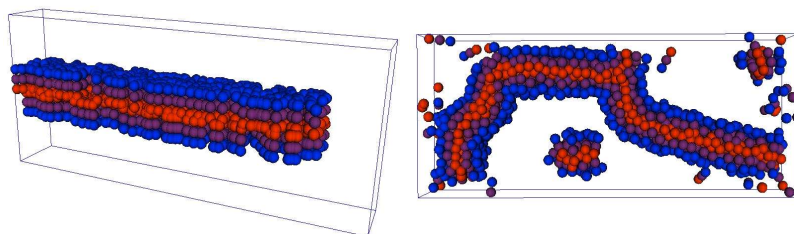
(b) Initialisation ($A_p = 50\text{\AA}^2$).

(c) Après 0.375e-9s simulée.

FIG. 5.2.14 – Bicouche homogène : distance de coupure de 25\AA .

La figure 5.2.15 présente toujours l'évolution d'un système de 600 lipides en bicouche (aire par lipide $\sim 28.8\text{\AA}^2$) mais dans un volume différent. Les lipides sont initialisés sur une surface restreinte de $50\text{x}300\text{\AA}^2$ mais sont libres par la suite de s'étendre sur une surface de $50\text{x}350\text{\AA}^2$. Dans les deux cas, la hauteur est fixée à 150\AA . La bicouche conserve sa cohésion et forme des ondulations qui correspondent à une extension "verticale" suivant l'axe z, mais cette bicouche s'étire également "horizontalement" suivant l'axe y et tire partie de l'espace vide.

volume (\AA^3)	nb BA	cutoff (\AA)
$50 \times 300 \times 150 \rightarrow 50 \times 350 \times 150$	2x300	10

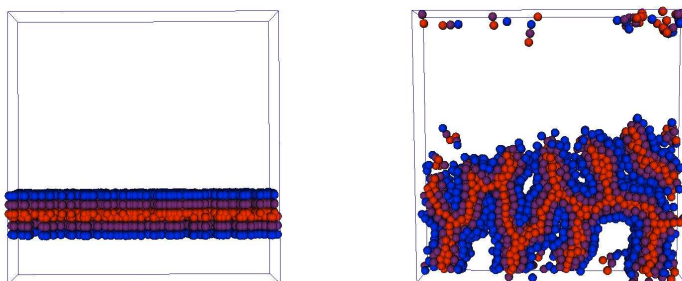
(b) Initialisation ($A_p = 50 \text{\AA}^2$).

(c) Après 39.278e-9s simulée.

FIG. 5.2.15 – Bicouche homogène : extension du volume suivant l'axe y.

La figure 5.2.16 montre un système constitué de 1040 lipides en bicouche, soit une aire par lipide de 28.8\AA^2 . L'extension du volume de simulation passe dans ce cas par l'augmentation de sa hauteur (suivant l'axe z). Sans que la structure finale soit superposable avec Fig.5.2.11, de nouveau la membrane développe des replis pour augmenter sa surface jusqu'à atteindre un état d'équilibre (Fig.5.2.16(c)) qui n'évolue plus (modulo les petites variations dues au mouvement brownien). La bicouche initialement comprimée se replie donc jusqu'à atteindre un état d'équilibre induit intrinsèquement par sa tension/pression superficielle. On constate que bien que la membrane ait la place nécessaire pour s'étaler dans le volume (suivant l'axe z), elle n'en fait rien.

volume (\AA^3)	nb BA	cutoff (\AA)
$50 \times 300 \times 150 \rightarrow 50 \times 300 \times 300$	2x520	10

(b) Initialisation ($A_p = 28.8 \text{\AA}^2$).

(c) Après 2.1e-9s.

FIG. 5.2.16 – Bicouche homogène : extension du volume suivant l'axe z.

5.2.6 Analyse des résultats

De l'ensemble de ces simulations de systèmes composés d'un seul type de lipides disposés initialement en bicouches, nous pouvons retenir plusieurs points :

- les populations de lipides conservent leur cohésion ; les bicouches peuvent modifier leur structure mais restent des bicouches ;
- l'agitation thermique exprimée par le mouvement brownien induit des perturbations locales ce qui a pour effet de produire des structures stables non superposables mais de même nature ;
- l'aire par lipide exprimant la densité des lipides à l'état initial (bicouche plane) est à mettre en corrélation avec l'évolution de la bicouche. En effet, plus les lipides sont comprimés (par

exemple avec une aire par lipide de 28.8\AA^2), plus la bicouche va former des replis pour augmenter sa surface dans un volume fixe.

- la bicouche est capable de s'étendre suivant le plan (XY), ce qui correspond en quelque sorte à une translation d'une partie de ses lipides. Tout comme les ondulations, cette extension vise à atteindre un état d'équilibre où les lipides bénéficient d'une plus grande aire par lipide.
- sans contrainte de hauteur, la bicouche produit quand même des replis d'une *longueur d'onde* $\sim 50\text{\AA}$ sans pour autant s'étendre dans tout le volume de simulation. Même si la nature torique du volume de simulation implique une pression latérale sur la membrane à l'origine de ces ondulations, les propriétés de ces replis sont à rattacher aux lipides eux-mêmes et à leurs interactions.

5.3 Vers une membrane interne mitochondriale

Les membranes biologiques, comme la membrane interne mitochondriale, sont composées de plusieurs types de lipides. Les lipides d'un même type peuvent adopter dans certaines conditions des comportements privilégiant leur regroupement pour former des microdomaines (*rafts*, Fig.5.3.1(a)).

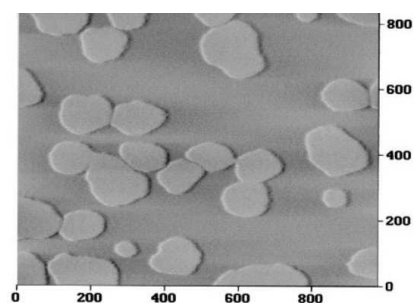
Stevens a exploré la formation de ces microdomaines dans les bicouches faites d'un mélange de lipides longs et courts [SHW03, Ste05]. Il a montré que des domaines en phase gel se forment à une température intermédiaire par rapport aux températures de fusion des lipides long et des lipides courts. Dans ces micro-domaines, différentes organisations des lipides peuvent être observées : la figure 5.3.1(b)(A) montre des lipides longs d'un côté opposés à des lipides courts tandis que la figure 5.3.1(b)(B) montre un regroupement des lipides de même type. L'arrangement A des lipides, en accord avec les données expérimentales [ZJTR04], minimise l'exposition des longues queues à l'environnement aqueux.

Par ailleurs, le tableau 1.4.3 donne notamment la composition de la membrane interne mitochondriale. Parmi les composants majoritaires, la phosphatidyl-choline (DMPC) et de la phosphatidyl-éthanolamine (POPE) sont des phospholipides très semblables. La cardiolipine qui est en troisième position, présente une structure différente avec 4 queues (voir section 1.3.2).

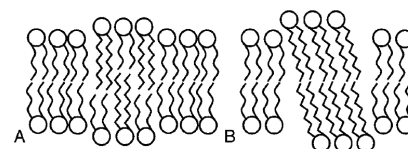
Pour tester l'impact de cette composition hétérogène de la membrane, nous avons choisi de réaliser des simulations présentant des mélanges de plusieurs types de phospholipides. Dans la première série de simulations, les deux populations de phospholipides se distinguent par la longueur des queues (DMPC/POPE). Dans les simulations suivantes, c'est le nombre de queues qui différencie les deux types de phospholipides (DMPC/cardiolipine). Nous allons maintenant discuter des résultats obtenus pour ces simulations.

Les résultats de simulation

Les simulations suivantes montrent l'impact d'une composition mixte de 2 types de phospholipides sur l'évolution d'une bicouche. Les potentiels sont exactement les mêmes, tout comme la distance de coupure réduite à 10\AA , seule la nature des lipides diffère.



(a) Mixture DLPC/DSPC refroidi de 70°C à 25°C aboutissant à la formation de domaines en forme de disque de DSPC entourés par une bicouche fluide de DLPC



(b) Différentes organisations des domaines d'une mixture de deux types de lipides (d'après Stevens [SHW03, Ste04, Ste05]).

FIG. 5.3.1 – Les microdomaines lipidiques.

► Deux types dissemblables par la longueur des queues

La figure 5.3.2 montre l'évolution d'un système mixte de 2x300 lipides. Les deux types de lipides sont équitablement répartis entre les deux feuillet. En d'autres termes, chaque feuillet est composé pour moitié du premier type et pour moitié du second type. Le premier type de lipides, identifié par un grain bleu (type **tête**), a une taille $\sim 30\text{\AA}$. Le second type, identifié par un grain vert (type **tête**), est sensiblement plus grand, avec une taille $\sim 35\text{\AA}$. Cette différence entre les tailles a été obtenue en éloignant le grain de type **queue** des deux autres grains constituant le lipide.

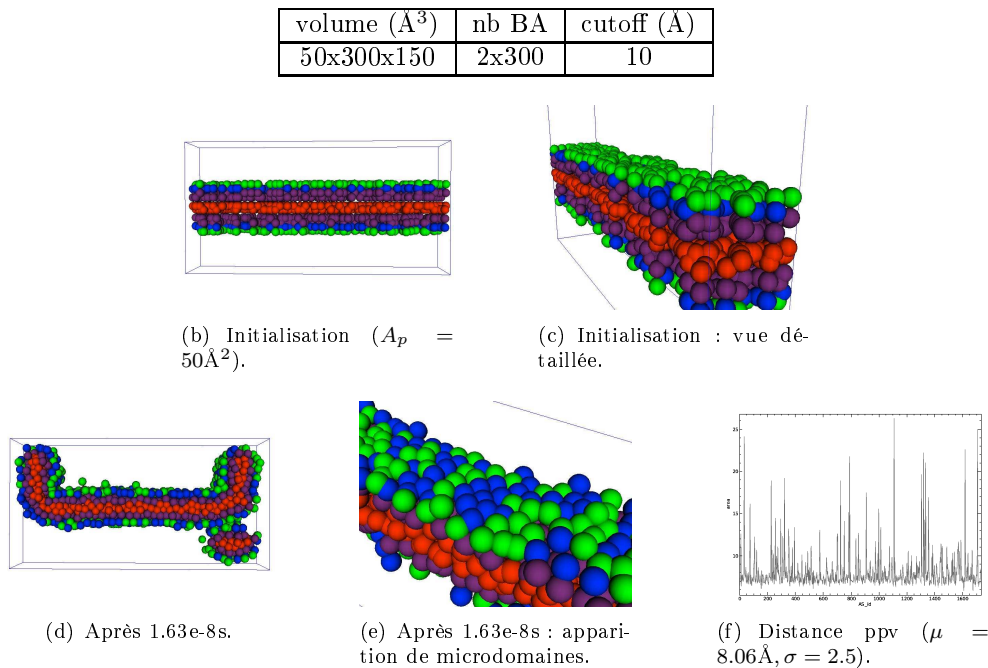


FIG. 5.3.2 – Bicouche mixte avec deux types de lipides dissemblables par la longueur des queues.

Comme le montrent les deux figures 5.3.2(d) et 5.3.2(e), après env. 10^6 pas de temps, soit $1.63\text{e-}8\text{s}$ de temps simulé, la bicouche observée offre certaines particularités. Cette bicouche de deux types de lipides dissemblables par la longueur de la queue cherche à s'étendre dans le volume de simulation mais adopte des structures étonnamment rectilignes. La membrane conserve sa cohésion et on observe clairement la formation de microdomaines de lipides ayant tous le même type (cf les zones de couleur uniforme dans la figure 5.3.2(e)). Malgré l'apparition de ces phénomènes, on peut constater que le calcul des distances ppv nous fournit une moyenne de ces distances $\mu = 8.06\text{\AA}$ assez proche des valeurs observées pour les systèmes homogènes précédents ($\mu \sim 7.5\text{\AA}$). Toutefois, l'observation de ce graphique montre bien qu'il existe ponctuellement des situations différentes pour lesquels la distance est nettement plus grande que la moyenne.

► Deux types dissemblables par le nombre de queues

La figure 5.3.3 montre l'évolution d'un système de 2 populations de 300 lipides ayant une ou deux queues. Comme précédemment, les deux types de lipides sont équitablement répartis entre les deux feuillet. La taille est la même $\sim 30\text{\AA}$ pour tous les lipides.

La figure 5.3.3(b) montre à gauche un premier type de lipides, identifié par un grain bleu (type **tête**), avec une seule queue (qui modélise 2 acides gras). À droite, le second type de lipides, identifié par un grain vert (type **tête**), a deux queues (qui modélisent 4 acides gras), c'est-à-dire que ce lipide est composé de 5 grains pris parmi les trois types du modèle. Attention : les grains bleus et verts sont tous des grains de type **tête**.

Dans cette simulation, nous pouvons immédiatement constater d'après les figures 5.3.3(e)(f) et

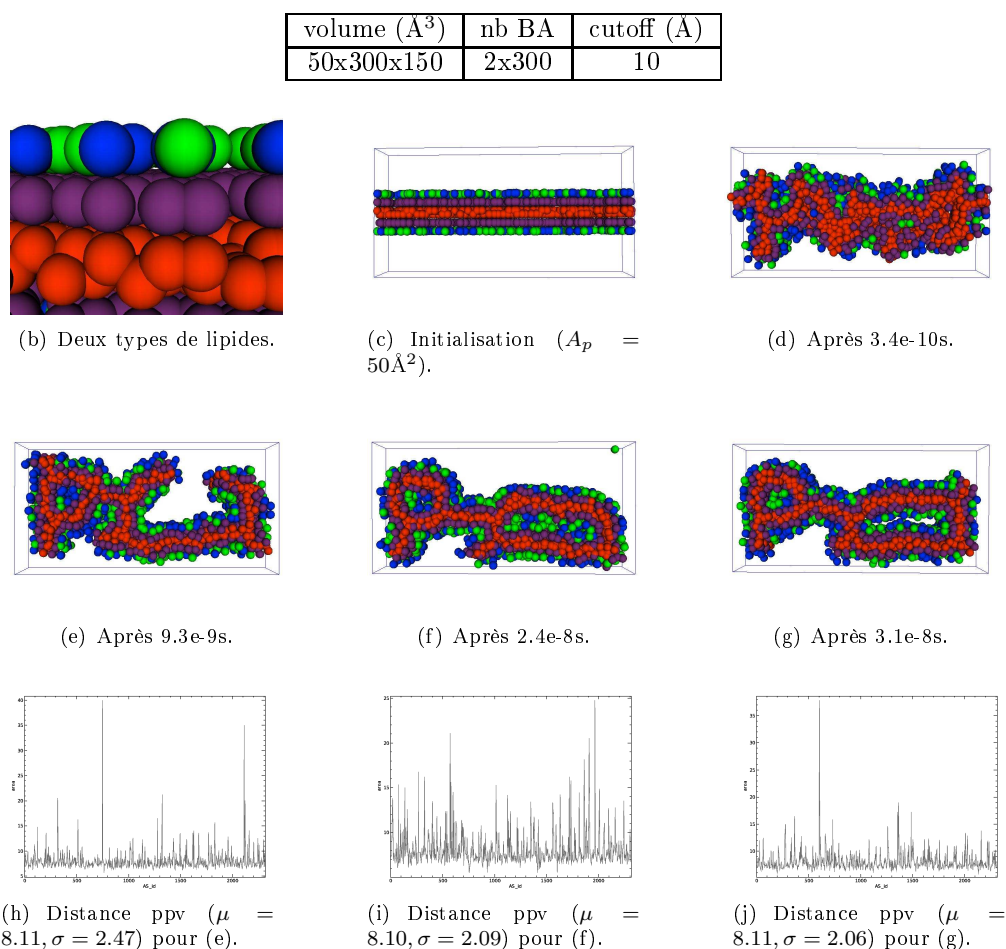


FIG. 5.3.3 – Bicouche mixte avec deux types de lipides dissemblables par le nombre de queues.

(g) que cette bicouche de deux types de lipides dissemblables par le nombre de queues (autrement dit par l'encombrement stérique) évolue d'une toute autre manière que l'autre bicouche hétérogène. Nous avons constaté qu'après un certain laps de temps (Fig.5.3.3(g)), cette bicouche se stabilise avec une structure bien plus complexe que précédemment. Différents cloisonnements apparaissent progressivement sans qu'il y ait d'interactions longues distances entre les lipides des extrémités des branches qui se rejoignent. Les histogrammes associés aux différents temps simulés montrent de plus que la distance de ppv tend à s'uniformiser (diminution de l'écart type).

5.4 Extension : les systèmes mixtes phospholipides/protéines

Avant d'étudier des systèmes mixtes tels que les complexes enzymatiques imbriqués dans la membrane interne mitochondriale, il est nécessaire de définir la représentation d'une protéine. Le fait que ces structures soient intra-membranaires et bien plus volumineuses que les phospholipides entraîne deux conséquences : il faut modéliser la surface d'interaction complexes/phospholipides et il n'est pas nécessaire de "remplir" le volume des complexes.

Nous commencerons l'étude de ces systèmes mixtes par l'utilisation de protéines membranaires génériques correspondant à de petits peptides de synthèse.

5.4.1 Des protéines intramembranaires génériques

L'**insertion** de protéines intramembranaires est une problématique de plus en plus étudiée de nos jours. Par exemple, la revue des différents modèles à grains pour membrane en section 3.4.1 cite Venturoli et al. [VSS05] pour leurs travaux sur l'insertion de protéines cylindriques de 3 diamètres différents, exprimés ici en nombre de grains : $4*(3+10+3)$, $7*(3+10+3)$ ou $43*(3+10+3)$.

Outre les mécanismes d'insertion, l'**impact** des peptides intramembranaires sur les structures de la bicouche ou encore les **migrations** de ces éléments nageant dans les phospholipides peuvent être l'objet de simulations intéressantes.

La description des interactions entre protéines membranaires est complexe et nous devons distinguer deux types d'interactions : les interactions directes, spécifiques ou non spécifiques qui existeraient même en l'absence de membrane et les interactions indirectes liées à la présence de la membrane. Les **interactions directes** sont très nombreuses, attractives ou répulsives selon les cas : interactions de volume exclu, interactions électrostatiques, forces de van der Waals, forces de solvation ou interactions spécifiques entre sites particuliers des protéines. Les **interactions indirectes**, elles, sont toujours attractives entre protéines identiques et ont un même mécanisme en commun : celui par lequel une contrainte quelconque créée par une protéine est relaxée si une seconde protéine vient se mettre au voisinage de la première. Ces contraintes peuvent être associées à des compressions ou des dilatations locales, à des variations de composition en lipides ou à des déformations de courbure de la membrane.

Avant d'analyser les résultats obtenus pour les différentes simulations, nous allons présenter en détail les différentes simulations réalisées.

► Les résultats de simulation

La figure 5.4.1 montre l'évolution d'un système mixte lipide/protéine. La bicouche est composée de 2×300 lipides ce qui correspond à des conditions initiales identiques à la simulation présentée en figure 5.2.13 (notamment une aire par lipide 50Å^2).

volume (Å^3)	nb BA	cutoff (Å)
$50 \times 300 \times 150$	2×300	10

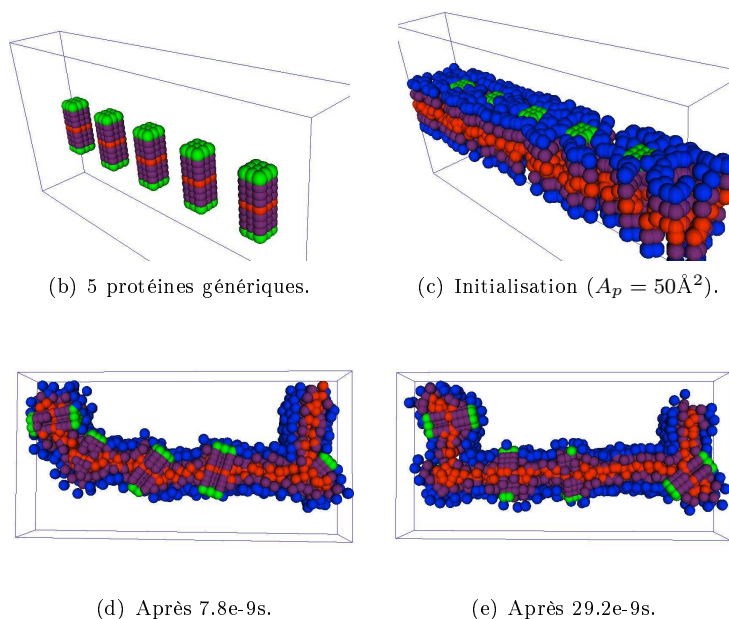


FIG. 5.4.1 – Bicouche mixte lipides/protéines avec 5 protéines génériques imbriquées.

La figure 5.4.1(c) montre 5 protéines génériques imbriquées dans la membrane, chacune d'un volume de $9 \times 9 \times 40 \text{ \AA}^3$. Ces petites protéines (ou peptides) sont constitués de 74 grains d'atomes pris parmi les trois types de grains préalablement définis pour les lipides. L'attribution des types est fonction de la portion représentée de la molécule : la "ceinture" hydrophobe est constituée de grains rouge de type **queue**, cette portion centrale est entourée de deux portions constituées de grains violets de type **interface**. Enfin, les surfaces de la molécule en contact avec le solvant sont constituées de grains verts de type **tête**. Les grains verts hydrophobes facilitent la localisation des protéines dans la membrane.

La figure 5.4.2 montre l'évolution d'un système mixte lipide/protéine identique au précédent à l'exception d'une bicouche plus densément peuplée de lipides : 2×520 soit une aire par lipide $\sim 28.8 \text{ \AA}^2$. Les 5 peptides sont toujours introduits de manière régulière dans la membrane.

volume (\AA^3)	nb BA	cutoff (\AA)
50x300x150	2x520	10

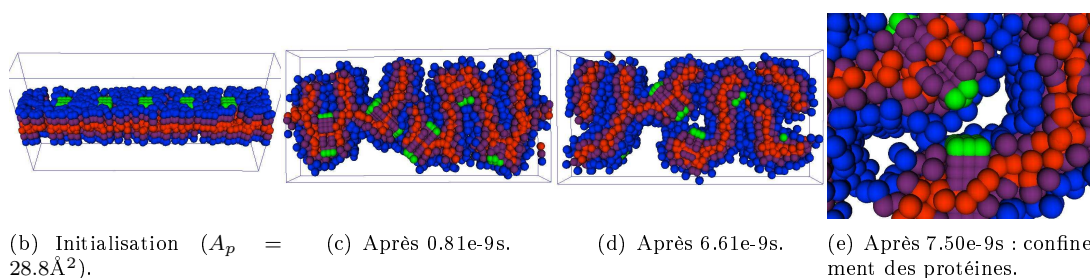


FIG. 5.4.2 – Bicouche mixte lipides/protéines avec 5 protéines génériques imbriquées.

La figure 5.4.3 montre l'évolution d'un système mixte lipide/protéine avec une bicouche composée de 2×800 lipides dans laquelle sont inclus 25 peptides génériques. Les dimensions du volume de simulation sont $200 \times 200 \times 150 \text{ \AA}^3$ ce qui donne une aire par lipide à l'initialisation de 50 \AA^2 . Cette bicouche présente la même densité en lipide que le premier système simulés en figure 5.2.13. Les peptides membranaires sont toujours constitués de 74 grains d'atomes également pris parmi les trois types de grains définis pour les lipides. Les grains verts permettent de suivre les déplacements de ces peptides, chacun d'un volume de $9 \times 9 \times 40 \text{ \AA}^3$.

volume (\AA^3)	nb BA	cutoff (\AA)
200x200x150	2x800	10

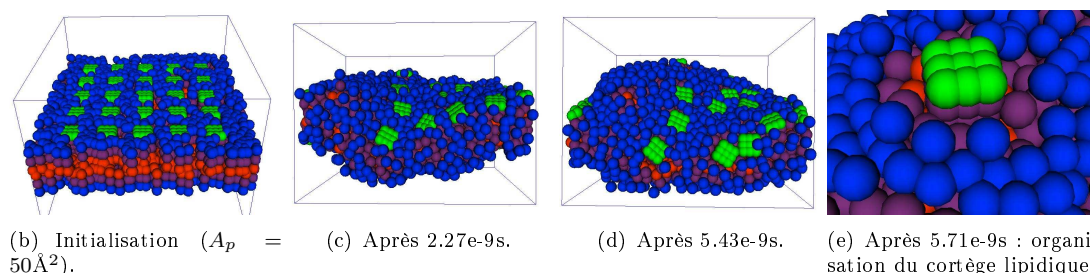


FIG. 5.4.3 – Bicouche mixte lipides/protéines avec 25 protéines génériques imbriquées.

► Analyse des résultats

De l'ensemble de ces simulations de systèmes mixtes lipides/protéines nous pouvons retenir les points suivants :

- la présence de peptides membranaires n'altère pas la cohésion de la bicouche ;

- les peptides diffusent dans la membrane jusqu'à atteindre une position stable;
- dans certains cas, les replis de la bicouche confinent les peptides à proximité les uns des autres, sans que ces derniers ne soient sur la même portion de membrane (Fig.5.4.2(e));
- la présence de peptides membranaires semble inciter les lipides en contact à adopter des orientations de biais pour encercler la protéine (Fig.5.4.3(e)).

5.4.2 La chaîne respiratoire

Pour rappel de la section 1.2, il existe 5 ensembles de protéines et de coenzymes impliqués dans les oxydations phosphorylantes de la chaîne respiratoire. Ces complexes diffusent librement au sein de la membrane interne et les électrons passent de l'un à l'autre par l'intermédiaire d'un transporteur liposoluble mobile le coenzyme Q (CoQ) et le cytochrome C fixé à la membrane.

Dans ce qui suit, nous allons présenter le potentiel électrostatique assimilable à la surface d'interaction entre des molécules et étudier un moyen de modéliser cette surface. Nous nous cantonnerons à présenter le complexe IV qui a fait l'objet des premiers essais de modélisation.

► Modélisation de protéines

Le potentiel électrostatique est en corrélation avec le moment dipolaire, l'**électronégativité** et les charges partielles. C'est une aide visuelle permettant d'apprécier la polarité relative d'une molécule. Le potentiel électrostatique moléculaire est l'énergie potentielle d'un proton localisé à une position particulière à proximité de la molécule. Un potentiel électrostatique négatif correspond à une attraction du proton par les densités d'électron dans la molécule (teintes rouges). Un potentiel électrostatique positif correspond à une répulsion du proton par les noyaux atomiques dans les régions de faible densité d'électrons (teintes bleues). L'**isosurface de densité d'électrons** est une surface sur laquelle la densité d'électrons a une valeur déterminée. Le potentiel électrostatique à différents points de cette isosurface est indiqué par une teinte du rouge au bleu. Plus il y a de différences rouge/bleu, plus la molécule est polaire. Plus il y a de teintes claires (blanc), plus la molécule est apolaire. Il est possible d'associer la surface d'interaction entre deux molécules à une isosurface de densité d'électrons caractérisant le potentiel électrostatique.

Étant donné que les interactions entre grains sont des potentiels par paire (de points situés dans l'espace), la prise en compte directe d'une surface risque de pénaliser sensiblement les performances des simulations. Nous souhaitons donc modéliser cette surface grâce à la discrétisation d'une isosurface donnée sous la forme d'un maillage de point permettant de définir la position des grains. Néanmoins, jusqu'à présent, nous n'avons pas trouvé de moyen efficace d'automatiser la discrétisation d'une iso-surface. Dans un premier temps, une approximation même grossière des volumes et des conformations des complexes devrait quand même générer des résultats intéressants.

► L'implémentation du complexe IV

Le complexe IV, la cytochrome c oxydase (EC-1.9.3.1, Fig.5.4.4), dispose de 19 structures référencées dans la PDB⁷. La structure idPDB 1V54 fournie par [TSK⁺] montre que ce complexe est composé de deux monomères symétriques. Le volume peut être assimilé à un pavé haut de 100Å, long de 150Å et large de 70Å.

⁷La banque de données sur les protéines du Research Collaboratory for Structural Bioinformatics, plus communément appelée Protein Data Bank ou PDB est une collection mondiale de données sur la structure tridimensionnelle (ou structure 3D) de macromolécules biologiques (source Wikipedia).

La figure 5.4.4(a) est une vue de côté du complexe IV. On observe une ceinture blanche entourant le complexe et correspondant à la portion apolaire imbriquée dans la membrane, c'est-à-dire la surface de contact avec les queues des lipides. La vue de dessus, figure 5.4.4(b), permet de distinguer les deux monomères fixé l'un à l'autre pour former le complexe IV. Une vue détaillée de chacun de ces deux monomères est donnée par la figure 5.4.4(c).

En observant attentivement les surfaces de contact entre ces deux monomères sous la forme d'isosurface de potentiel électrostatique, on note une “*correspondance électrostatique*” entre des zones que l'on pourrait considérer comme points d'ancrage des deux monomères.

Chacune de ces structures peut être assimilée à un pavé haut de 100Å, long de 75Å et large de 70Å.

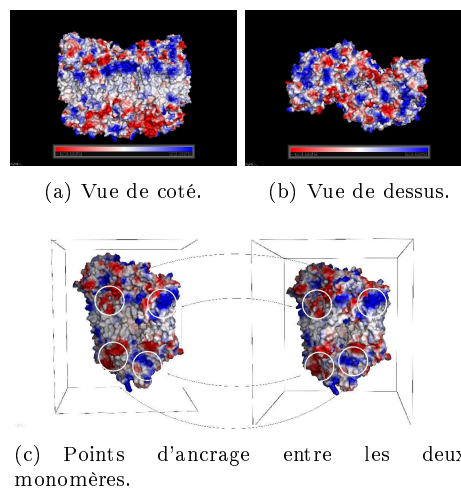


FIG. 5.4.4 – Complexe IV - Surface du potentiel électrostatique.

5.4.3 Les résultats de simulation

Un complexe IV, sous sa forme dimérique, est modélisé par une instance d'`EnzymaticComplex` (sous-classe de `BioAgent`). L'objet est haut de 100Å, long de 150Å et large de 70Å. La construction paramétrée de cette structure respecte un écartement de 10Å entre les grains et requiert quelques 652 grains. Seule la surface est constituée de grains puisqu'a priori les grains internes n'ont pas à interagir avec l'extérieur. La structure est rigide ce qui permet l'emploi d'une dynamique des corps rigides identiques à celle des phospholipides (voir section 4.2.1) et l'application de l'équation du mouvement Eq.(4.5.15). Les grains sont pris parmi les 3 types de grains définis pour les phospholipides, à savoir un type hydrophile (bleu), un type hydrophobe intermédiaire (violet) et un type hydrophobe fort (rouge). Cette réutilisation des types existants évite d'avoir des paramètres supplémentaires à calibrer.

Dans un premier temps, on s'attache à modéliser ces deux monomères pour tenter d'observer la formation d'un dimère. Les deux monomères sont positionnés à proximité l'un de l'autre dans un volume de simulation porté à $200 \times 200 \times 200 \text{Å}^3$. Les potentiels sont les mêmes entre les différents types de points que ceux utilisés pour les lipides. Les points d'ancrage mis en évidence sur la figure 5.4.4(c) sont ici modélisés par des grains de type `queue` sujets à un potentiel dont le puits énergétique est plus prononcé que les potentiels effectifs entre les autres types (cf Fig.4.2.2(b) du chapitre 4).

Les différentes simulations réalisées dans un premier temps sans bicouche donnent deux types de convergences : soit les monomères atteignent des positions suffisamment proches pour entrer en interaction et former un dimère qui alors ne se séparera plus par la suite, soit les monomères diffusent suivant un mouvement brownien dans tout le volume de simulation, et mettront un temps nettement plus long à former un dimère. En présence d'une bicouche, la probabilité que les deux monomères entrent en interaction est sensiblement augmentée.

En conclusion, chacun des monomères du complexe IV nécessite un ensemble de 652 grains

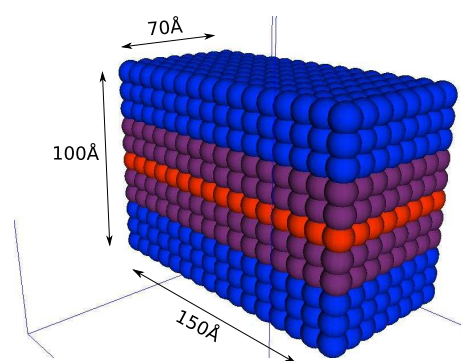
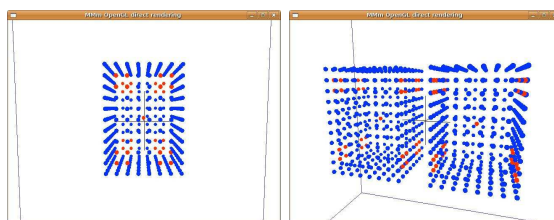


FIG. 5.4.5 – Complexe IV - modélisation par grain et dimensions.

volume (\AA^3)	nb BA	cut off (\AA)
200x200x200	2x800 + 2	10



(b) Un monomère.

(c) Deux monomères.

FIG. 5.4.6 – Complexe IV - modélisation de deux monomères.

pour modéliser la surface d'interaction avec les lipides. Sans rajouter de nouveau type aux 3 types de grains définis pour les lipides, il est possible de modéliser les points d'ancrage de ces monomères. Les toutes premières simulations de systèmes mixtes phospholipides/monomères du complexe IV montrent que sans la présence d'une bicouche pour contraindre les monomères à rester sur un espace 2D, ces derniers sont susceptibles de diffuser longtemps avant de se retrouver suffisamment à proximité pour pouvoir former le dimère complet du complexe IV.

5.5 En résumé

Ce chapitre a présenté l'ensemble des simulations réalisées grâce à la plateforme MitoMAS.

Les premiers systèmes simulés correspondent à des ensembles de lipides initialisés aléatoirement dans le volume de simulation. D'une manière générale, les lipides cherchent à s'agglutiner pour former des agrégats et minimisent ainsi l'exposition des queues au solvant. Ce résultat cohérent avec les données expérimentales montre qu'il est possible d'utiliser une représentation implicite du solvant. Par ailleurs, les structures de ces agrégats dépendent de la densité initiale en lipides.

Par la suite, les simulations de lipides en monocouche ont permis de mettre en avant l'influence de la distance de coupure (*cutoff*). En terme d'efficacité des simulations, plus la distance de coupure est grande, plus il y a de distances calculées et plus les pas de temps d'intégration sont courts pour compenser des intensités élevées des forces. En terme de résultats, les systèmes évoluent vers des configurations qualitativement semblables mais les dynamiques se révèlent d'autant plus rapides que la distance de coupure est grande. A l'issue de ces simulations, nous proposons d'utiliser une distance de coupure réduite à 10\AA .

Les simulations de bicouches produisent des replis dans certaines conditions de pression latérale, cette pression provenant d'une densité importante des lipides à l'initialisation. Nous avons montré que le nombre et les dimensions de ces replis est fonction justement de cette densité initiale qui se traduit par une aire par lipide réduite. Nous proposons une nouvelle observable basée sur la distance du plus proche voisin permettant de suivre la répartition des lipides les uns par rapport aux autres tout au long de l'évolution de la structure de la membrane. Nous avons vu que des propriétés intrinsèques aux lipides, comme par exemple la longueur de leurs queues ou le nombre de ces queues, modifient qualitativement les replis de la membrane sans pour autant faire varier sensiblement la disposition des lipides les uns par rapport aux autres. Par ailleurs, nous avons pu observer la formation de microdomaines constitués d'un seul type de lipides.

Enfin, les simulations de systèmes mixtes lipides/protéines ont montré que la structure des replis est sensible à la présence de protéines membranaires et que ces replis sont capables de produire des confinements localisés de protéines sans pour autant que celles-ci soient proches les unes des autres sur la membrane. Les premières simulations des deux monomères du complexes IV de la chaîne respiratoire ont montré qu'il était possible de représenter une protéine uniquement en modélisant sa surface avec des grains et ce, en réutilisant les types que nous avons défini pour la membrane.

Conclusion

Dans ces travaux de thèse, nous avons étudié différents paradigmes de modélisation des systèmes biologiques en mettant en avant leurs capacités ou leurs difficultés à prendre en compte les particularités des objets biologiques et de l'environnement dans lequel ils évoluent. Les modèles basés sur les équations différentielles permettent une analyse au niveau global de la dynamique du système par l'utilisation de variables moyennées mais en contre partie, la spacialisation des molécules et leur conformation sont des informations délicates à prendre en compte avec ce paradigme. Avec une approche de plus en plus locale, nous avons abordé les réseaux de Petri, les automates cellulaires pour enfin étudier les Systèmes Multi-Agents (SMA) capables de modéliser les systèmes moléculaires dans toutes leurs spécificités.

L'approche la plus fine utilisée en biologie se situe au niveau atomique. En effet, les modèles basés sur les particules, que sont les techniques de la mécanique moléculaire, représentent les objets biologiques au niveau atomique ce qui donne des outils performants pour étudier les structures et les dynamiques biologiques pour des temps de l'ordre de la nanoseconde et des tailles de l'ordre du nanomètre. Néanmoins, les simulations d'un système de particules en interaction sont rapidement limitées par la puissance de calcul des ordinateurs. Depuis quelques années, les modèles à grains d'atomes ou modèles gros grains, se révèlent être un bon compromis entre la finesse de représentation des molécules et l'efficacité des simulations puisqu'en l'occurrence, ils offrent la possibilité d'étudier les phénomènes biologiques à des échelles de temps de l'ordre du dixième de microseconde et d'espace de l'ordre du dixième de micromètre, ce qui permet notamment l'étude de phénomènes comme la formation des replis des membranes.

Dans le cadre de la modélisation de la membrane interne mitochondriale, siège des réactions enzymatiques de la chaîne respiratoire, ces travaux de thèse proposent un modèle gros grains qui représente les phospholipides sous la forme de trimères rectilignes rigides de grains d'atomes. Les interactions intermoléculaires sont exprimées au travers d'un ensemble de potentiels de type Lennard-Jones. Ces potentiels modélisent implicitement le solvant et prennent en compte l'hydrophobie des queues des lipides. L'équation du mouvement que nous avons définie contient une composante brownienne pour rendre compte de l'agitation thermique des molécules.

Ce modèle gros grains théorique a donné lieu à une conception agent où les objets biologiques sont pris en compte par des agents réactifs situés spatialement et en interaction locale avec leur voisinage. Cette conception a fait l'objet d'une implémentation sous la forme de la plateforme MitoMAS permettant de réaliser des simulations de systèmes de lipides voire des systèmes mixtes lipides/protéines. Les possibilités de visualisation et la production de statistiques permettent une évaluation qualitative des dynamiques et une caractérisation quantitative des systèmes simulés.

Les résultats de ces travaux en terme de modélisation montrent la capacité de cette association modèle *gros grains* et SMA à intégrer des informations de conformation, d'orientation et de spacialisation sur un nombre suffisant d'objets biologiques pour simuler des phénomènes biologiques à l'échelle mésoscopique tels que ceux impliqués dans la chaîne respiratoire.

En terme de simulations, la plateforme MitoMAS nous a permis d'étudier différents systèmes de molécules. Les systèmes initialisés de manière aléatoire montrent une agrégation des lipides pour minimiser l'exposition des queues au solvant. Ce phénomène naturel tend à valider la modélisation implicite du solvant. Nous avons montré que la densité initiale des lipides détermine les structures

des agrégats observés. Les simulations réalisées avec des monocouches lipidiques ont permis de souligner l'influence de la distance de *cutoff* pour le calcul des potentiels intermoléculaires. Avec une initialisation des lipides en bicouche, nous avons pu observer sous certaines conditions de pression latérale l'apparition de replis similaires à ceux de la membrane interne. Ces replis permettent d'augmenter la surface de la membrane jusqu'à atteindre une densité en lipide de cette surface correspondant à un minimum d'énergie. A ce niveau, nous proposons une observable basée sur la distance du plus proche voisin permettant d'apprécier l'évolution de la disposition des lipides les uns par rapport aux autres, et ce indépendamment de la forme de la bicouche. Un ensemble de simulations avec deux types de phospholipides différenciés soit par la taille des queues, soit par le nombre de ces queues, produit des membranes qualitativement différentes mais présentant des distances de plus proche voisin relativement identiques. MitoMAS nous a aussi permis de simuler des systèmes mixtes phospholipides/protéines membranaires. En présence de petites protéines génériques imbriquées, les replis de la membrane ne sont plus les mêmes et on observe dans certains cas des confinements de protéines sans que celles-ci soient proches les unes des autres dans la membrane.

Les perspectives de ces recherches sont très diverses, notamment en termes d'expérimentations *in-silico*. Il est dorénavant possible d'évaluer l'impact de certains paramètres comme la flexibilité des molécules, la présence de solvant, la distance de *cutoff* ou encore de rechercher des potentiels plus *doux* qui permettent d'atteindre des temps simulés plus grands dans des temps de calcul raisonnables. Les premiers résultats de modélisation d'un des complexes enzymatiques de la chaîne respiratoire montrent qu'il est possible de représenter une protéine uniquement par le maillage de sa surface et ce, sans avoir à définir de nouveaux types de grains. Notre objectif est donc d'utiliser ces résultats pour intégrer d'autres complexes dans la membrane et ainsi d'évaluer des hypothèses spécifiques quant au fonctionnement coopératif des complexes enzymatiques.

Bibliographie

- [ABB⁺81] S. Anderson, A. T. Bankier, B. G. Barrell, M. H. L. de Bruijn, A. R. Coulson, J. Drouin, I. C. Eperon, D. P. Nierlich, B. A. Roe, F. Sanger, P. H. Schreier, A. J. H. Smith, R. Staden, and I. G. Young. Sequence and organization of the human mitochondrial genome. *nat*, 290 :457–465, 1981.
- [ABL86] *Biologie moléculaire de la cellule*. Flammarion Médecine Sciences, 1986.
- [AC96] Martín Abadi and Luca Cardelli. *A Theory of Objects*. Springer, 1996.
- [And30] D. H. Andrews. The relation between the raman spectra and the structure of organic molecules. *Phys. Rev.*, 36(3) :544–554, Aug 1930.
- [APBBFS⁺04] R. Acin-Pérez, M. Bayona-Bafaluy, P. Fernandez-Silva, R. Moreno-Loshuertos, A. Pérez-Martos, C. Bruno, C. Moraes, and J. Enriquez. Respiratory complex iii is required to maintain complex i in mammalian mitochondria. *Molecular Cell*, 2004.
- [APN⁺98] I. Arnold, K. Pfeiffer, W. Neupert, R. A. Stuart, and H. Schagger. Yeast mitochondrial flf0-atp synthase exists as a dimer : identification of three dimer-specific subunits. *EMBO J.*, 1998.
- [AV02] G. Ayton and G.A. Voth. Bridging microscopic and mesoscopic simulations of lipid bilayers. *Biophys. J.*, 83(6) :3357–3370, 2002.
- [AV07] G.S. Ayton and G.A. Voth. Multiscale simulation of transmembrane proteins. *J Struct Biol*, 157 :570–578, 2007.
- [B9393] *Random Walks in Biology*. Princeton University Press, 1993.
- [BA82] U. Burkert and N. L. Allinger. *Molecular Mechanics*. Washington, DC, 1982.
- [Ben58] B. Benson, A.A. and Maruo. Plant phospholipids. i. identification of the phosphatidyl glycerols. In *Biochim Biophys Acta*, volume 27, pages 189–195, 1958.
- [Bez01] M. Bezzi. Modeling evolution and immune system by cellular automata. *Rivista del Nuovo Cimento*, 24(1), 2001.
- [BLB06] G. Brannigan, L.C.L. Lin, and F.L.H. Brown. Implicit solvent simulation models for biomembranes. *Eur. Biophys. J. Biophys. Lett.*, 35(2) :104–124, 2006.
- [BO27] M. Born and R. Oppenheimer. Zur quantentheorie der molekeln. *Annalen der Physik*, 1927.
- [Boy97] Paul D. Boyer. The atp synthase a splendid molecular machine. *Annual Review of Biochemistry*, 66(1) :717–749, 1997.
- [BPB05] G. Brannigan, P.F. Phillips, and F.L.H. Brown. Flexible lipid bilayers in implicit solvent. *Phys. Rev. E*, 72, 2005.
- [Bro03] F. L. H. Brown. Regulation of protein mobility via thermal membrane undulations. *Biophys. J.*, 2003.
- [Can70] P. B. Canham. The minimum energy of bending as a possible explanation of the biconcave shape of human red blood cell. *J. Theor. Biol.*, 1970.
- [CD05] I.R. Cooke and M. Deserno. Solvent-free model for self-assembling fluid bilayer membranes : Stabilization of the fluid phase based on broad attractive tail potentials. *J. Chem. Phys.*, 123(22), 2005.

- [Cla44] Albert Claude. THE CONSTITUTION OF MITOCHONDRIA AND MICROSOMES, AND THE DISTRIBUTION OF NUCLEIC ACID IN THE CYTOPLASM OF A LEUKEMIC CELL. *J. Exp. Med.*, 80(1) :19–29, 1944.
- [CRMT03] C. Chaouiya, E. Remy, B. Mossé, and D. Thieffry. Qualitative analysis of regulatory graphs : a computational tool based on a discrete formal framework. In A. Benvenuti, L. De Santis and L. Farina, editors, *PFIRST Multidisciplinary International Symposium on Positive Systems : Theory and Applications*, volume 294 of *Lecture Notes in Control and Information Science*, pages 119–126. Springer-Verlag, 2003.
- [CS96] F. Celada and P. E. Seiden. Affinity maturation and hypermutation in a simulation of the humoral immune response. *European J. Immunology*, 26 :1350–1358, 1996.
- [CSM⁺07] A. Coulon, H. Soula, O. Mazet, O. Gandrillon, and G. Beslon. Modélisation cellulaire pour l'émergence de structures multiprotéiques auto-organisées. *Technique et Science Informatiques (TSI)*, 26 :125–150, 2007.
- [CV06] J.W. Chu and G.A. Voth. Coarse-grained modeling of the actin filament derived from atomistic-scale simulations. *Biophys J*, 90 :1572–1582, 2006.
- [D93] PhD thesis.
- [dBvdLH93] Rob J. de Boer, Jan D. van der Laan, and Pauline Hogeweg. *Randomness and Pattern Scale in the Immune Network : A Cellular Automata Approach*. Thinking About Biology. Addison-Wesley, 1993.
- [dJ02] H. de Jong. Modeling and simulation of genetic regulatory systems : A literature review. *J. Comp. Biol.*, 9(1) :69–105, 2002.
- [DML91] J.-M. Drouffe, A. C. Maggs, and S. Leibler. Computer Simulations of Self-Assembled Membranes. *Science*, 254 :1353–1356, November 1991.
- [EA94] F. Ercolessi and J. B. Adams. Interatomic potentials from first-principles calculations : the force-matching method. *Europhys. Lett.*, 1994.
- [EW95] P. Espanol and P. Warren. Statistical-mechanics of dissipative particle dynamics, 1995.
- [Far03] O. Farago. "water-free" computer model for fluid bilayer membranes. *J. Chem. Phys.*, 119(1) :596–605, 2003.
- [FB93] Kurt Fleischer and Alan H. Barr. A simulation testbed for the study of multicellular development : The multiple mechanisms of morphogenesis. pages 389–416, 1993.
- [FC99] Eirik G. Flekkoy and Peter V. Coveney. From molecular dynamics to dissipative particle dynamics. *Phys. Rev. Lett.*, 83(9) :1775–1778, Aug 1999.
- [Fel00] S.E. Feller. Molecular dynamics simulations of lipid bilayers. *Curr Opin Colloid Interface Sci*, 5 :217–223, 2000.
- [Fer95] J. Ferber. *Les Systèmes Multi-Agents*. Inter-Editions, 1995.
- [FG] J. Ferber and O. Gutknecht. A meta-model for the analysis and design of organizations in multi-agent systems. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS98)*.
- [FM00] T. G. Frey and C. A. Mannella. The internal structure of mitochondria. *Trends Biochem. Sc.*, 2000.
- [For79] M.W. Formo. *Physical properties of fats and fatty acids*. Bailey's industrial oil and fat products, volume 1. Fourth edition. Wiley-Interscience : New-York, 1979.
- [FSBMP99] R. Faller, H. Schmitz, O. Biermann, and F. Müller-Plathe. Automatic parameterization of force fields for liquids by simplex optimization. *J. Comput. Chem.*, 1999.
- [G7979] *Scaling Concepts in Polymer Physics*. Cornell University Press, 1979.
- [G9797] *Modern Differential Geometry of Curves and Surfaces with Mathematica*. CRC Press, 1997.

- [GG98] Volker Gaede and Oliver Günther. Multidimensional access methods. *ACM Comput. Surv.*, 30(2) :170–231, 1998.
- [GK73] L. Glass and S.A. Kauffman. The logical analysis of continuous, nonlinear biochemical control networks. *J. Theor. Biol.*, 39 :103–129, 1973.
- [GK92] G. Gompper and Stefan Klein. Ginzburg-landau theory of aqueous surfactant solutions. *J. Phys. II*, 1992.
- [GKV01] H. Genrich, R Kuffner, and K. Voss. Executable petri net models for the analysis of metabolic pathways. *International Journal on Software Tools for Technology Transfer*, 3(4) :394–404, 2001.
- [GL98] R. Goetz and R. Lipowsky. Computer simulations of bilayer membranes : Self-assembly and interfacial tension. *J. Chem. Phys.*, 108(17) :7397–7409, 1998.
- [GP98] P.J.E. Goss and J. Peccoud. Quantitative modeling of stochastic systems in molecular biology by using stochastic petri nets. In *Proc. Natl. Acad. Sci. USA 95*, page 6750, 1998.
- [GR01] R.D. Groot and K.L. Rabone. Mesoscopic simulation of cell membrane damage, morphology change and rupture by nonionic surfactants. *Biophys. J.*, 81 :725–736, 2001.
- [GS] G. Gompper and M. Schick. Self-assembling amphiphilic systems. In *Phase Transitions and Critical Phenomena*.
- [GT005] *Simulation for the Social Scientist*. Open University Press European, 2005.
- [GT06] H. Gohlke and M.F. Thorpe. A natural coarse graining for simulating large biomolecular motion. *Biophys J*, 91 :2115–2120, 2006.
- [Gut84] Antonin Guttman. R-trees : a dynamic index structure for spatial searching. In *SIGMOD '84 : Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, pages 47–57, New York, NY, USA, 1984. ACM Press.
- [GW97] Robert D. Groot and Patrick B. Warren. Dissipative particle dynamics : Bridging the gap between atomistic and mesoscopic simulation. *The Journal of Chemical Physics*, 107(11) :4423–4435, 1997.
- [Hel73] W. Helfrich. Elastic properties of lipid bilayers : theory and possible experiments. *Z. Naturforsch. C*, 1973.
- [HK92] P. J. Hoogerbrugge and J. M. V. A. Koelman. Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics. *Europhysics Letters*, 19 :155–+, June 1992.
- [Hof94] R. Hofestadt. A petri net application to model metabolic processes. *SAMS*, 16 :113–122, 1994.
- [ISL05] A. Imparato, J.C. Shillecock, and R. Lipowsky. Shape fluctuations and elastic properties of two-component bilayer membranes. *Europhys. Lett.*, 69(4) :650–656, 2005.
- [IT73] J. Israelachvili and D. Tabor. Molecular films-shear properties. *Nature Physical Science*, 241 :148–+, February 1973.
- [IV05] S Izvekov and G.A. Voth. A multiscale coarse-graining method for biomolecular systems. *J. Phys. Chem. B*, 109 :2469–2473, 2005.
- [KG90] Kurt Kremer and Gary S. Grest. Dynamics of entangled linear polymer melts : A molecular-dynamics simulation. *The Journal of Chemical Physics*, 1990.
- [KH93] J. M. V. A. Koelman and P. J. Hoogerbrugge. Dynamic simulations of hard-sphere suspensions under steady shear. *Europhysics Letters*, 21 :363–+, January 1993.
- [KJD06] O. Kurkcuoglu, R.L. Jernigan, and P. Doruker. Loop motions of triosephosphate isomerase observed with elastic networks. *Biochemistry*, 45 :1173–1182, 2006.
- [KM02] M. Karplus and J.A. McCammon. Molecular dynamics simulations of biomolecules. *Nat Struct Biol*, 9(9) :646–52, 2002.
- [KNS04] M. Kranenburg, J.-P. Nicolas, and B. Smit. Comparison of mesoscopic phospholipid-water models. *Phys. Chem. Chem. Phys.*, 6(16) :4142–4151, 2004.

- [Kre40] H A Krebs. The citric acid cycle and the szent-györgyi cycle in pigeon breast muscle. *Biochem. J.*, 34(5) :775–0, 1940.
- [Kuh72] T. Kuhn. *La Structure des révolutions scientifiques*. nouvelle trad. par Laure Meyer, 1983. Flammarion, 1972.
- [LB00] K. B. Lipkowitz and D. B. Boyd. *Reviews in Computational Chemistry*. 1990-2000.
- [Lea96] A. R. Leach. *Molecular Modelling. Principles and Applications*. Addison Wesley Longman, Essex, England, 1996.
- [Lr005] *A Different Universe : Reinventing Physics from the Bottom Down*. Basic Books, 2005.
- [LS95] R. Lipowsky and E. Sackmann. *Structure and Dynamics of Membranes*. Handbook of Biological Physics, 1. Elsevier, Amsterdam, 1995.
- [LS05] O. Lenz and F. Schmid. A simple computer model for liquid lipid bilayers. *J. Mol. Liq.*, 117(1-3) :147–152, 2005.
- [LSK05] M. Laradji and P.B. Sunil-Kumar. Domain growth, budding, and fission in phase-separation self-assembled fluid bilayers. *J. Chem. Phys.*, 123(22), 2005.
- [LV07] P. Liu and G.A. Voth. Smart resolution replica exchange : an efficient algorithm for exploring complex energy landscapes. *J Chem Phys*, 126 :45106, 2007.
- [Lyu95] A. Lyubartsev, A.P. and Laaksonen. Calculation of effective interaction potentials from radial distribution functions : A reverse monte carlo approach. *Phys. Rev. E*, 52 :3730–3737, 1995.
- [Lyu05] A.P. Lyubartsev. Multiscale modeling of lipids and lipid bilayers. *Eur. Biophys. J. Biophys. Lett.*, 35 :53–61, 2005.
- [LYZ06] E. Lyman, F.M. Ytreberg, and D.M. Zuckerman. Resolution exchange simulation. *Phys Rev Lett*, 96 :28105, 2006.
- [LZ06] E. Lyman and D.M. Zuckerman. Resolution exchange simulation with incremental coarsening. *J Chem Theory Comput*, 2 :656–666, 2006.
- [Man99] J.-B. Manneville. *Fluctuations de membranes actives*. PhD thesis, University Paris VII, 1999.
- [MBRP01] J.-B. Manneville, P. Bassereau, S. Ramaswamy, and J. Prost. Active membrane fluctuations studied by micropipet aspiration. *Phys. Rev. E*, 2001.
- [McN01] J. McNames. A fast nearest-neighbor algorithm based on a principal axis search tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9) :964–976, 2001.
- [MCS+03] David L. Mobley, Daniel L. Cox, Rajiv R. P. Singh, Rahul V. Kulkarni, and Alexander Slepoy. Simulations of oligomeric intermediates in prion diseases. *Biophysical Journal*, 85 :2213–2223, 2003.
- [MDNM00] H. Matsuno, A. Doi, M. Nagasaki, and S. Miyano. Hybrid petri net representation of gene regulatory network. In *Pacific Symposium on Biocomputing 2000*, page 338, 2000.
- [MIT79] Peter MITCHELL. Compartmentation and Communication in Living Systems. Ligand Conduction : a General Catalytic Principle in Chemical, Osmotic and Chemiosmotic Reaction Systems. *European Journal of Biochemistry*, 95(1) :1–20, 1979.
- [MKS06] M. Müller, K. Katsov, and M. Schick. Biological and synthetic membranes : What can be learned from a coarse-grained description? *Physics Reports*, 434 :113–176, 2006.
- [MM04] A.H. de Marrink, S.J. and Vries and A.E. Mark. Coarse grained model for semi-quantitative lipid simulations. *J. Chem. Phys.B*, 108(2) :750–760, 2004.
- [MNPT06] Yannis Manolopoulos, Alexandros Nanopoulos, Apostolos N. Papadopoulos, and Yannis Theodoridis. *R-Trees : Theory and Applications*. Advanced Information and Knowledge Processing. Springer, 2006.

- [MS94] R. Morikawa and Y. Saito. Hard rod and frustum model of two-dimensional vesicles. *Journal de Physique II*, 4 :145–161, 1994.
- [NAC⁺05] M. Neri, C. Anselmi, M. Cascella, A. Maritan, and P. Carloni. Coarse-grained model of proteins incorporating atomistic detail of the active site. *Phys Rev Lett*, 95 :218102, 2005.
- [Nag93] F. Nagle. Area/lipid of bilayers from nmr. *Biophys. J.*, 64 :1476–1481, 1993.
- [NB666] *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966.
- [NPW04] D.R. Nelson, T. Piran, and S. Weinberg. *Statistical Mechanics of Membranes and Surfaces*. second ed. World Scientific, Singapore, 2004.
- [NSK04] C.F. Nielsen, S.O. and Lopez, G. Srinivas, and M.L. Klein. Coarse grain models and the computer simulation of soft materials. *J. Phys. : Condens. Matter*, 16(15) :R481–R512, 2004.
- [NT01] H. Noguchi and M. Takasu. Self-assembly of amphiphiles into vesicles : A brownian dynamics simulation. *Phys. Rev. E*, 6404(4), 2001.
- [NTN] J.F. Nagle and S. Tristram-Nagle. Structure of lipid bilayers. In *Biochim. Biophys. Acta*.
- [NZTN⁺96] F. Nagle, R. Zhang, S. Tristram-Nagle, W. Sun, H.I. Petrache, and R.M. Suter. X-ray structure determination of fully hydrated 1 alpha phase dipalmitoylphosphatidylcholine bilayers. *Biophys. J.*, 70 :1419–1431, 1996.
- [Pav94] A. Pavé. *Modélisation en Biologie et en Ecologie*. Aléas, 1994.
- [Pet66] C.A. Petri. Communication with automata. In *Tech. Rep. RADC-TR-65-377*, volume 1, pages 609–627. Colloq. Math. Soc. János Bolyai, Vol. 10. New York : Griffiss Air Force Base, 1966.
- [PVC⁺02] P. Paumard, J. Vaillier, B. Coulary, J. Schaeffer, V. Soubannier, D. M. Mueller, D. Brethes, J. P. di Rago, and J. Velours. The atp synthase is involved in generating mitochondrial cristae morphology. *The EMBO Journal*, 2002.
- [Pól21] Georg Pólya. Über eine aufgabe der wahrscheinlichkeitsrechnung betreffend die irrfahrt im strafennetz. *Mathematische Annalen*, 1921.
- [R0606] *The Python Language Reference Manual (version 2.5)*. Network Theory Ltd, 2006.
- [Rah64] A. Rahman. *Phys. Rev.*, 136 :405, 1964.
- [RML93] V.N. Reddy, M.L. Mavrouniotis, and M.N. Liebman. Petri net representations in metabolic pathways. In *Proc. First ISMB*, pages 328–336, 1993.
- [Rob93] N.C. Robinson. Functional binding of cardiolipin to cytochrome c oxidase. *J Bioenerg Biomembr.*, 25(2) :153–63, 1993.
- [RS71] A. Rahman and F. H. Stillinger. *J. Chem. Phys.*, 55 :3336, 1971.
- [Sac90] E. Sackmann. Molecular and global structure and dynamics of membranes and lipid bilayers. *Can. J. Phys.*, 1990.
- [SB97] T. Sintes and A. Baumgärtner. Protein attraction in membranes induced by lipid fluctuations. *Biophys. J.*, 73(5) :2251–2259, 1997.
- [SB98a] T. Sintes and A. Baumgärtner. Interaction of wedge-shaped proteins in flat bilayer membranes. *J. Phys. Chem., B* 102(36) :7050–7057, 1998.
- [SB98b] T. Sintes and A. Baumgärtner. Membrane-mediated protein attraction. a monte-carlo study. *Physica A*, 249 :571–575, 1998.
- [Sdw05] *An Introduction to Markov Processes*. Springer-Verlag, 2005.
- [SHW03] Mark J. Stevens, Jan H. Hoh, and Thomas B. Woolf. Insights into the molecular mechanism of membrane fusion from simulation : Evidence for the association of splayed tails. *Phys. Rev. Lett.*, 91(18) :188102, Oct 2003.
- [SIV06] Q. Shi, S. Izvekov, and G.A. Voth. Mixed atomistic and coarse-grained molecular dynamics : simulation of a membrane bound ion channels. *J Phys Chem B*, 110 :15045–15048, 2006.

- [SKR01] *Adventures in Modeling : Exploring Complex Dynamic Systems with StarLogo*. Teachers College Press, 2001.
- [SL02] J.C. Shillcock and R. Lipowsky. Equilibrium structure and lateral stress distribution of amphiphilic bilayers from dissipative particle dynamics. *J. Phys. Chem.*, 2002.
- [SP00] H. Schagger and K. Pfeiffer. Supercomplexes in the respiratory chains of yeast and mammalian mitochondria. *EMBO J.*, 2000.
- [SPB01] R. Srivastava, M.S. Peterson, and W.E. Bentley. Stochastic kinetic analysis of the escherichia coli stress circuit using 32-targeted antisense. *Biotechnol. Bioeng.*, 75(1) :120–129, 2001.
- [SRP+05] H. Soula, C. Robardet, F. Perrin, S. Gripon, G. Beslon, and O. Gandrillon. Modeling the emergence of multi-protein dynamic structures by principles of self-organization through the use of 3dsp, a multi-agent-based software. *BMC Bioinformatics*, 228(6), 2005.
- [SRX+05] M. Schlame, M. Ren, Y. Xu, M.L. Greenberg, and I. Haller. Molecular symmetry in mitochondrial cardiolipins. *Chem. Phys. Lipids*, 138 :38–49, 2005.
- [Ste04] M. J. Stevens. Coarse-grained simulations of lipid bilayers. *Journal of Chemical Physics*, 121 :11942–11948, December 2004.
- [Ste05] M.J. Stevens. Complementary matching in domain formation within lipid bilayers. *J. Am. Chem. Soc.*, 127 :15330–15331, 2005.
- [SU01] B. Schattner and A.M. Uhrmacher. Agents in modeling and simulation : exploiting the metaphor. In *Proceedings of the IEEE*, volume 89, pages 158–173, 2001.
- [SYSY02] R. Srivastava, L. You, J. Summers, and J. Yin. Stochastic versus deterministic modeling of intracellular viral kinetics. *J. Theor. Biol.*, 218 :309, 2002.
- [SZS05] J. Sun, Q. Zhang, and T. Schlick. Electrostatic mechanism of nucleosomal array folding revealed by computer simulation. *Proc Natl Acad Sci USA*, 102 :8180–8185, 2005.
- [Toz05] V. Tozzini. Coarse-grained models for proteins. *Curr Opin Struct Biol*, 15 :144–150, 2005.
- [TSK+] T. Tsukihara, K. Shimokata, Y. Katayama, H. Shimada, K. Muramoto, H. Aoyama, M. Mochizuki, K. Shinzawa-Itoh, E. Yamashita, M. Yao, Y. Ishimura, and S. Yoshikawa. The low-spin heme of cytochrome c oxidase as the driving element of the proton-pumping process. In *Proc.Natl.Acad.Sci.Usa*.
- [TTK95] R. Thomas, D. Thieffry, and M. Kauffman. *Dynamical behaviour of biological regulatory networks-I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state*. Bull. Math. Biol., 57. Elsevier, Amsterdam, 1995.
- [TVFB03] F. Tama, M. Valle, J. Frank, and C.L. Brooks. Dynamic reorganization of the functionally active ribosome explored by normal mode analysis and cryo-electron microscopy. *Proc Natl Acad Sci USA*, 100 :9319–9323, 2003.
- [VBS05] E. Villa, A. Balaeff, and K. Schulten. Structural dynamics of the lac repressor dna complex revealed by a multiscale simulation. *Proc Natl Acad Sci USA*, 102 :6783–6788, 2005.
- [Ven04] M. Venturoli. *Mesoscopic models of lipid bilayers and bilayers with embedded proteins : a Dissipative Particle Dynamics study*. PhD thesis, University of Amsterdam, 2004.
- [VHK03] K. Voss, M. Heiner, and I. Koch. Steady state analysis of metabolic pathways using petri nets. *InSilico Biology*, 3(3) :367, 2003.
- [VS99] M. Venturoli and B. Smit. Simulating the self-assembly of model membranes. *PhysChemComm*, 2(10) :7397, 1999.
- [VSS05] M. Venturoli, B. Smit, and M.M. Sperotto. Simulation studies of protein-induced bilayer deformations, and lipid-induced protein tilting, on a mesoscopic model for lipid bilayers with embedded proteins. *Biophys. J.*, 88 :1778–1798, 2005.

- [WF05a] Z.J. Wang and D. Frenkel. Modeling flexible amphiphilic bilayers : A solvent-free off-lattice monte carlo study. *J. Chem. Phys.*, 122(23), 2005.
- [WF05b] Z.J. Wang and D. Frenkel. Pore nucleation in mechanically stretched bilayer membranes. *J. Chem. Phys.*, 123(15), 2005.
- [WH69] R. Wood and R.D. Harlow. Structural studies of neutral glycerides and phosphoglycerides of rat liver. *Arch. Biochem. Biophys.*, 131 :495–501, 1969.
- [Wol84] Stephen Wolfram. Universality and complexity in cellular automata. *Physica D*, 1984.
- [WRBJ04] Y. Wang, A.J. Rader, I. Bahar, and R.L. Jernigan. Global ribosome motions revealed with elastic network model. *J Struct Biol*, 147 :302–314, 2004.
- [YMH02] S. Yamamoto, Y. Maruyama, and S. Hyodo. Dissipative particle dynamics study of spontaneous vesicle formation of amphiphilic molecules. *J. Chem. Phys.*, 116 :5842–5849, 2002.
- [ZJTR04] J. Zhang, B. Jing, N. Tokutake, and S.L. Regen. Transbilayer complementarity of phospholipids. a look beyond the fluid mosaic model. *J. Am. Chem. Soc.*, 126 :10856–10857, 2004.
- [ZSKP95] Bernard P. Zeigler, Hae Sang Song, Tag Gon Kim, and Herbert Praehofer. Devs framework for modelling, simulation, analysis, and design of hybrid systems. pages 529–551, 1995.

Table des figures

1.1.1 La mitochondrie.	12
1.1.2 Images de synthèse générées à partir de segments 3D de tomogrammes de mitochondries du cervelet de poulet (d'après Frey, T. G. et al. [FM00]).	12
1.1.3 Le réseau mitochondrial.	13
1.1.4 Fusion et fission du réseau mitochondrial (images de Werner Koopman, 2005).	13
1.1.5 Diversité topologique du réseau mitochondrial.	13
1.2.1 "Oxidative-phosphorylation reference pathway" de KEGG	14
1.3.1 Phospholipides génériques. Bleu/vert : les acides gras, noir : le glycérol, rouge : le phosphate et violet : le composé hydroxylé estérifiant le phosphate (images : wikipedia).	16
1.4.1 Diversité des membranes biologiques [IT73].	19
1.4.2 Membranes cellulaires (source : Wikipedia).	19
1.4.3 Composition de différentes membranes biologiques (sources : Département de Chimie et Biochimie de l'Université du Texas).	20
2.1.1 Échelles spatio-temporelles caractéristiques des systèmes vivants [Pav94].	24
2.1.2 Différents paradigmes pour les systèmes moléculaires.	25
2.2.1 Une réseau de Petri.	30
2.2.2 Automate cellulaire et phénomène biologique.	31
2.2.3 "Life" de John Conway.	32
2.3.1 Exemple d'émergence.	34
2.3.2 Différentes classifications des méthodes de partitionnement de l'espace.	37
2.3.3 StarLogo TNG.	41
2.3.4 La plateforme NetLogo utilisée pour modéliser une réaction enzymatique.	41
2.3.5 L'environnement Breve.	42
2.3.6 La plateforme BioDyn et certaines de ses applications.	43
2.3.7 HSIM et son application pour la polymérisation de filaments d'actine.	44
2.3.8 Le simulateur 3DSpi et son application pour l'étude des corps nucléaires.	44
3.1.1 Différents modes de déformation d'une membrane (d'après [Can70] et [Hel73]).	48
3.1.2 Représentation schématique d'une membrane initialement plane, sans courbure spontanée, de longueur L , d'aire projetée $S_0 = L^2$ dans le plan (x, y) et soumise à des fluctuations. Les déplacements transversaux de la membrane sont des fonctions $z = u(x, y)$ (d'après [Man99]).	49

3.2.1	Différentes approches “basées sur les particules” pour étudier des membranes (d’après Müller et al. [MKS06]).	51
3.3.1	L’énergie d’une molécule dépend des positions relatives de ses atomes.	52
3.4.1	Diagramme schématique des échelles temporelles et spatiales accessibles par différentes techniques, notamment le coarse-graining. Certaines structures et événements membranaires caractéristiques sont indiqués (d’après [NSK04]).	55
3.4.2	Un modèle par barre dure (ou <i>hard-rod</i>) de deux lipides l’un en face de l’autre (d’après Morikawa et Saito [MS94]).	57
3.4.3	Modèle à 7 grains (d’après Lenz et Schmid [LS05]).	57
3.4.4	Tête et queues peuvent être représenté par un nombre variable de grains (d’après Venturoli et al. [VS99]).	58
3.4.5	Différentes représentations de la même molécule de DMPC (d’après Kranenburg et al. [KNS04]).	58
3.4.6	Différents modèles utilisant 3 types de grain pour représenter un lipide.	59
3.4.7	Différents modèles utilisant plus de 3 types de grain pour représenter un lipide.	60
3.4.8	Modèle à grains générique : eau, ions, butane, hexadecane, DPC et DPPC (d’après Marrink et al. [MM04]).	60
3.4.9	Modèle flexible d’une molécule générique par des potentiels de liaison et de flexion.	61
3.4.10	Potentiel FENE pour un trimère flexible (d’après Wang et Frenkel [WF05a]).	63
3.4.11	Deux approches pour prendre en compte le solvant (d’après Lenz et Schmid [LS05]). Dans les deux cas, l’image de gauche correspond à une phase gel ($p = 1.0$ et $T = 0.9$) et celle de droite, à une phase liquide ($p = 1.0$ et $T = 1.0$) avec p et T respectivement la pression et la température.	67
3.4.12	Farago utilise simultanément différents potentiels.	71
4.1.1	Différents modèle à grains pour la dimyristoyl-phosphatidylcholine DMPC. A : représentation schématique, B : modèle 2 types de grain (1 tête et 1 queue), C : modèle 2 types de grain (1 tête et 2 queues), D : modèle 3 types de grain (tête, queue et interface).	75
4.1.2	Un phospholipide à 3 grains pris parmi 3 types pour former un trimère.	76
4.1.3	Différentes molécules assemblées avec 3 types de grain.	77
4.2.1	Plusieurs modèles, plus ou moins flexibles, d’une même molécule.	78
4.2.2	Différents potentiels type Lennard-Jones plus ou moins adoucis.	80
4.2.3	Évolution de l’interaction entre deux grains de même type en fonction de la distance r et des exposants du potentiel.	81
4.4.1	Illustration de la difficulté que pose la nature torique d’un espace pour le calcul de la distance entre deux points A et B.	83
4.7.1	La super classe <code>Agent</code> et ses dérivées.	87
4.7.2	Diagramme de classes de <code>BioAgent</code> et de ses dérivées.	89
4.7.3	Diagramme de classes des spécialisation de <code>ModelOfInteractions</code>	91
4.7.4	Diagramme de classes de <code>GridAgent</code>	93
4.7.5	Voisinage élaboré d’un <code>GridAgent</code>	94
4.7.6	Diagramme de classes du noyau.	96
4.8.1	Le module de visualisation.	98
5.2.1	Une molécule de DMPC modélisée un trimère linéaire rigide.	102
5.2.2	Différentes initialisations du volume de simulation.	102

5.2.3	Distributions mélangées.	103
5.2.4	Initialisation aléatoire des lipides.	104
5.2.5	Initialisation aléatoire des lipides.	104
5.2.6	Diffusion latérale des lipides.	106
5.2.7	Paramètre d'ordre des lipides.	107
5.2.8	Aire par lipide et distances au plus proche voisin.	107
5.2.9	Une simulation avec initialisation en monocouche des lipides.	108
5.2.10	Profil de densité.	109
5.2.11	Bicouche homogène : simulation 1.	110
5.2.12	Bicouche homogène : 2 autres exemples de simulation.	110
5.2.13	Bicouche homogène : influence de l'aire par lipide.	111
5.2.14	Bicouche homogène : distance de coupure de 25Å.	111
5.2.15	Bicouche homogène : extension du volume suivant l'axe y.	112
5.2.16	Bicouche homogène : extension du volume suivant l'axe z.	112
5.3.1	Les microdomaines lipidiques.	113
5.3.2	Bicouche mixte avec deux types de lipides dissemblables par la longueur des queues.	114
5.3.3	Bicouche mixte avec deux types de lipides dissemblables par le nombre de queues.	115
5.4.1	Bicouche mixte lipides/protéines avec 5 protéines génériques imbriquées.	116
5.4.2	Bicouche mixte lipides/protéines avec 5 protéines génériques imbriquées.	117
5.4.3	Bicouche mixte lipides/protéines avec 25 protéines génériques imbriquées.	117
5.4.4	Complexe IV - Surface du potentiel électrostatique.	119
5.4.5	Complexe IV - modélisation par grain et dimensions.	119
5.4.6	Complexe IV - modélisation de deux monomères.	120
A.1.1	plan complexe	139
A.1.2	multiplication de deux complexes	140
A.1.3	L'ensemble de Julia quaternionique pour $q(n + 1) = q(n)^2 + c$, avec $c = (-0.747, -0.256, -0.3, 0.0)$	145
A.1.4	Ensembles de Mandelbrot classique (2D) et quaternionique (4D)	146
B.4.1	Partitionnement par grille.	155
B.4.2	Partitionnement par arbre BSP (Binary Space Partitioning).	156
B.4.3	Partitionnement par kd-tree.	156
B.4.4	Partitionnement par kdb-tree (pour k-dimensional b-arbre).	156
B.4.5	Partitionnement par quadtree.	157
B.4.6	Quadtree.	157
B.4.7	Partitionnement par octree.	158
B.4.8	Partitionnement par R-arbre ou R-tree.	158
B.4.9	Ce PAT (Principal Axis Tree) ou arbre en composantes principales a une profondeur de 2 et divise une partition mère en 7 partitions filles. L'arbre ainsi construit est statistiquement le mieux équilibré, surtout pour des distributions non-homogènes et cependant périodiques comme ici.	158

Liste des tableaux

1.3.1 Les principaux acides gras cellulaires (d'après [For79]).	15
1.3.2 Distribution des acides gras de certains phospholipides de foie de rat (d'après Wood et Harlow [WH69]).	17
1.3.3 PC (phosphatidyl-choline), PE (phosphatidyl-éthanolamine), PS (phosphatidyl-sérine), PG (phosphatidyl-glycérol), CL (cardiolipine)	17
2.2.1 Différents paradigmes et leur relation au temps, à l'espace et aux éléments modélisés.	26
3.4.1 Différents modèles à grains consultables dans la littérature en fonction du nombre de types de grain et du nombre de grains par lipide.	56
3.4.2 Jeu de paramètres pour les potentiels utilisés par Farago [Far03].	71
3.4.3 Une discrétisation de l'intensité d'un potentiel intermoléculaire type Lennard-Jones (d'après Marrink et al. [MM04]).	71
4.2.1 Jeu de paramètres pour les potentiels intermoléculaires.	80
5.2.1 Effet de la distance de coupure (<i>cutoff</i>) sur des simulations de monocouche.	108
B.2.1 Différentes plates-formes ou environnement intégré de modélisation offrant un accès simplifié à l'expérimentation in-silico.	154
B.3.1 Différentes plates-formes à programmation avancée.	155

Liste des algorithmes

4.1	Attribution d'un <code>GridAgent</code> à un <code>AtomSet</code>	93
4.2	Attribution d'un voisinage à un <code>AtomSet</code>	94
4.3	Calcul de la distance entre un grains et un de ses voisins.	94
4.4	Utilisation des différents potentiels type LJ avec les paramètres associés.	95
4.5	Boucle principale de la simulation.	97

Chapitre A

Quelques éléments de mathématiques

Dans cette partie des annexes, nous allons présenter les quaternions qui sont des objets mathématiques directement liés au moteur physique de MitoMAS. Ensuite nous donnerons un cadre plus précis aux systèmes d'équations différentielles abordés dans le chapitre 2. Puis nous traiterons des méthodes d'intégration que nous avons étudiées avant d'opter pour une intégration simple grâce à l'équation du mouvement de MitoMAS qui relie les forces appliquées aux molécules directement à leurs vitesses.

A.1 Quaternions et rotations 3D

Les sections suivantes présentent les quaternions et leur utilisation pour gérer efficacement les rotations dans l'espace 3d. Nous précisons ici notamment les opérations quaternions/matrices de rotation qui sont implémentées dans notre moteur physique de dynamique des corps rigides. Pour rappel, dans le cas de molécules rigides, les forces appliquées sur les grains à distance non nulle du centre de masse de la molécule induisent des couples à l'origine des rotations de cette molécule. En l'occurrence, la composition des rotations successives à chaque pas de temps de la simulation se traduit sous la forme d'une multiplication de quaternions.

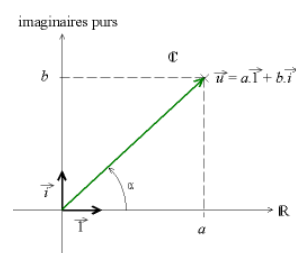
A.1.1 L'ensemble \mathbb{C} des nombres complexes

► Notion intuitive

En mathématiques, les nombres complexes sont une extension naturelle des nombres réels. Ils sont apparus en 1545 comme intermédiaires de calcul pour résoudre des équations du troisième degré ($a_3x^3 + a_2x^2 + a_1x + a_0 = 0$) dont on connaissait des solutions mais pour lesquelles l'application des formules de Cardan (Tartaglia-Cardan pour être précis) faisait appel à des racines dont les carrés seraient négatifs.

Les nombres complexes, comme tout concept mathématique, constituent à la fois une théorie et un outil potentiel. Pour les physiciens, par exemple, les nombres complexes constituent surtout un moyen très commode de simplifier les notations : on manipule deux valeurs distinctes avec un seul nom, une rotation s'exprime par une simple multiplication, etc.

Géométriquement, tout nombre complexe peut être représenté comme un point dans un plan appelé le plan complexe. On est donc dans un plan géométrique muni d'un repère, l'axe horizontal est muni du vecteur $\vec{1}$, l'axe vertical est muni du vecteur \vec{i} . L'ensemble des nombres réels peut être représenté par une droite du plan complexe.



► Forme algébrique

Quand les nombres complexes sont écrits sous la forme $z = a + ib$, on parle de forme algébrique. Les nombres a et b sont des réels, alors que le symbole i est tel que $i^2 = -1$. Le conjugué de $z = a + ib$ est $\bar{z} = a - ib$.

► Formes trigonométrique et exponentielle

Pour un nombre complexe non nul z de partie réelle a et de partie imaginaire b , il existe un réel strictement positif r et des réels θ tels que $a = r \cos \theta$ et $b = r \sin \theta$. Les réels θ diffèrent les uns des autres d'un multiple entier de 2π et sont appelés argument de z . On a $z = a + ib = r(\cos \theta + i \sin \theta) = re^{i\theta}$. La notation $z = r(\cos \theta + i \sin \theta)$ est appelée *forme trigonométrique* du nombre z , et la notation $z = re^{i\theta}$ est appelée *forme exponentielle* du nombre z . Dans ces deux formes, r est le module ou valeur absolue de z , θ est l'argument de z (modulo 2π). On peut alors noter le conjugué du nombre z sous ces deux formes : $\bar{z} = re^{-i\theta}$ (forme exponentielle) et $\bar{z} = r(\cos \theta - i \sin \theta)$. Cependant, cette dernière notation n'est pas une notation sous forme trigonométrique : la forme trigonométrique du conjugué est $\bar{z} = r(\cos(-\theta) + i \sin(-\theta))$.

► Règles de calcul

- $(a + ib) + (a' + ib') = (a + a') + i(b + b')$
- $(a + ib)(a' + ib') = (aa' - bb') + i(ab' + a'b)$
- si a ou b est non-nul, on voit que $\frac{a}{a^2+b^2} - i\frac{b}{a^2+b^2}$ est l'inverse de $a + ib$ (autrement noté $\frac{1}{r}e^{-i\theta}$).

En fait, on voit que si l'on prend un vecteur quelconque du plan, si α est l'angle qu'il fait avec l'axe des réels, alors la multiplication imaginaire d'un autre vecteur par revient à faire

- une rotation d'angle α
- une dilatation de $\|\vec{u}\|$

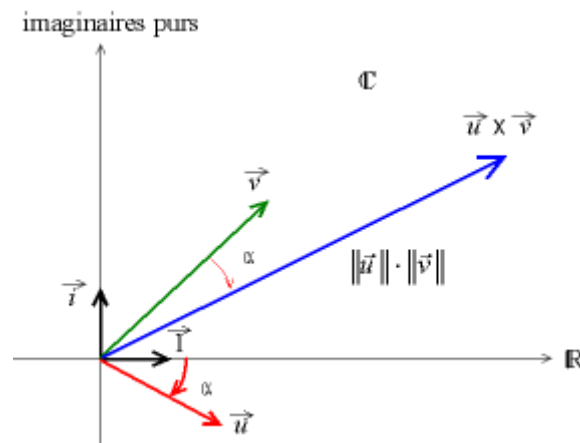


FIG. A.1.2 – multiplication de deux complexes

A.1.2 L'ensemble \mathbb{H} des quaternions

Les quaternions, notés \mathbb{H} , sont un type de nombres hypercomplexes, constituant une extension des nombres complexes, extension similaire à celle qui avait conduit des nombres réels aux nombres complexes.

On utilisera par la suite les notations suivantes : $q = a + ib + jc + kd$ ou encore $q = q_w + iq_x + jq_y + kq_z$.

► **Découverte**

Les quaternions furent inventés par l'irlandais William Rowan Hamilton en 1843. Hamilton recherchait des manières d'étendre les nombres complexes (qui peuvent être assimilés à des points d'un plan) à des dimensions plus élevées de l'espace. Il ne réussit pas à le faire pour la dimension trois mais la dimension quatre produisit les quaternions.

Selon ses dires, il marchait un jour dehors le long du canal royal, avec son épouse quand la solution sous forme de relations $i^2 = j^2 = k^2 = ijk = -1$ lui apparut soudainement à l'esprit. Il grava alors promptement ces relations avec un couteau dans une pierre du pont de Brougham (maintenant appelé Broom Bridge) à Dublin.

Cette découverte entraîna l'abandon de l'utilisation exclusive des lois commutatives, une avancée radicale pour l'époque. Les vecteurs et les matrices faisaient encore partie du futur, mais Hamilton venait en quelque sorte d'introduire le produit vectoriel et le produit scalaire des vecteurs.

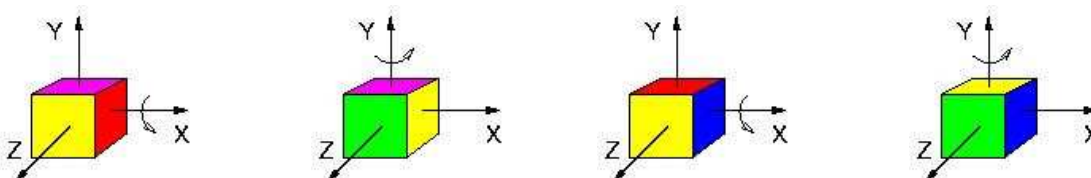
Hamilton décrivit un quaternion comme quadruplet de nombres réels, le premier élément étant un « scalaire », et les trois éléments restants formant un « vecteur », ou « imaginaire pur ».

► **Propriétés mathématiques**

L'algèbre des quaternions est :

- associatif : $(q_1 \cdot q_2) \cdot q_3 = q_1 \cdot (q_2 \cdot q_3)$
- distributif : $(q_1 \cdot q_2) \cdot q_3 = q_1 \cdot q_2 \cdot q_3$
- non commutative : $q_1 \cdot q_2 \neq q_2 \cdot q_1$
- partiellement anticommutative : $1 \cdot i = i \cdot 1 = i$ mais $i \cdot j = k$ et $j \cdot i = -k$

Cette non commutativité est d'ailleurs tout à fait compatible avec une interprétation géométrique des quaternions, par exemple les rotations vectorielles du plans sont commutatives mais celle de l'espace ne le sont pas.



(a) Une rotation autour de l'axe X suivie d'une rotation autour de l'axe Y.

(b) Une rotation autour de l'axe Y suivie d'une rotation autour de l'axe X.

► **Arithmétique**

► **Addition** Associative et commutative : $(a + ib + jc + kd) + (e + if + jg + kh) = (a + e) + i(b + f) + j(c + g) + k(d + h)$

► **Multiplication** Associative mais non commutative en général, distributive par rapport à l'addition et satisfaisant aux règles de calcul suivantes, fournies par la table :

	1	i	j	k
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

Le produit de 2 quaternions $q_1 = a + ib + jc + kd$ et $q_2 = a' + ib' + jc' + kd'$ est défini comme suit :

$$q_1 \cdot q_2 = aa' - bb' - cc' - dd' + i(ab' + ba' - cd' + dc') + j(ac' + ca' - db' + bd') + k(ad' + da' - bc' + cb')$$

➤ **Conjugaison** On définit le conjugué d'un quaternion $q = a + ib + jc + kd$ par : $\bar{q} = a - ib - jc - kd$. Le produit d'un quaternion par son conjugué donne : $q \cdot \bar{q} = (a + ib + jc + kd) \cdot (a - ib - jc - kd) = a^2 + b^2 + c^2 + d^2$

➤ **Normalisation** Norme $(q) = \|q\| = \sqrt{q \cdot \bar{q}} = \sqrt{a^2 + b^2 + c^2 + d^2}$

➤ **Inverse** L'inverse d'un quaternion est : $q^{-1} = \frac{\bar{q}}{q \cdot \bar{q}} = \frac{\bar{q}}{\|q\|^2} = \frac{\bar{q}}{a^2 + b^2 + c^2 + d^2}$

➤ L'ensemble \mathbb{S} des quaternions de norme 1

Ce sont ces quaternions qui nous intéressent plus particulièrement pour représenter les rotations en tant qu'isométries.

On appelle \mathbb{S} le sous groupe de \mathbb{H} des quaternions de norme 1 :

$$q \in \mathbb{S} \Leftrightarrow \|q\| = \sqrt{a^2 + b^2 + c^2 + d^2} = 1$$

En termes rigoureux, il y a homéomorphie de groupe entre \mathbb{S} et le groupe des rotations de \mathbb{R}^3 . Autant dire à toute rotation géométrique on peut associer un quaternion q et donc l'application linéaire R_q qui la représente.

➤ Applications

Aujourd'hui, les quaternions trouvent leur place en infographie, en théorie de la commande, dans le traitement du signal, dans la commande de mouvement et la mécanique orbitale, principalement pour représenter les rotations et les orientations en dimension trois. Par exemple, il est fréquent que les systèmes de commande de déplacement d'un vaisseau spatial soient régis en termes de quaternions. La raison est qu'effectuer beaucoup d'opérations sur les quaternions est numériquement plus précis que d'effectuer beaucoup d'opérations sur les matrices.

➤ **Méthodes pour les rotations** Numériquement, une rotation dans l'espace peut être caractérisée par :

- 3 scalaires, des angles par exemple de Cardan ou d'Euler, avec des singularités de définitions dans certaines configurations
- la représentation géométrique de la rotation par un axe unitaire et un angle, soit 4 scalaires reliés par une relation (axe unitaire)
- la matrice de rotation P ou de passage avec 9 scalaires reliés par 6 relations (${}^t P = P$ et les 3 vecteurs colonnes unitaires)
- un quaternion (4 scalaires vérifiant une relation)

Dans tous les cas, l'objet mathématique créé a trois degrés de liberté, mais ils sont plus ou moins dilués dans tous les nombres qui composent l'objet. Le choix d'un formalisme ou d'un autre se fait en regardant le nombre de composants, la simplicité des opérations (composition de rotations, application d'une rotation à un vecteur, extraction de l'axe et de l'angle de rotation), l'existence de singularités.

	nb paramètres	combine les rotations par multiplication ?	supporte les transformations affines ?
angles d'Euler	3	non	non
{axis, angle}	4	non	non
quaternions	4	oui	non
matrices	16 (9 si rotation)	oui	oui

➤ **Rotation d'un point dans l'espace** On peut déterminer les coordonnées d'un point après rotation sans utiliser de matrices.

Soient :

- la position originale du point $p_1 = 0 + ix_1 + jy_1 + kz_1$
- la position résultante du point $p_2 = 0 + ix_2 + jy_2 + kz_2$
- le quaternion associé à la rotation $q = q_w + iq_x + jq_y + kq_z$
- le conjugué de q : $\bar{q} = q_w - iq_x - jq_y - kq_z$

On a :

$$p_2 = q.p_1.\bar{q}$$

D'où :

$$p_2 \begin{cases} x_2 = x_1.(q_w^2 + q_x^2 - q_y^2 - q_z^2) + y_1.(2.q_x.q_y - 2.q_w.q_z) + z_1.(2.q_x.q_z + 2.q_w.q_y) \\ y_2 = x_1.(2.q_w.q_z + 2.q_x.q_y) + y_1.(q_w^2 - q_x^2 + q_y^2 - q_z^2) + z_1.(-2.q_w.q_x + 2.q_y.q_z) \\ z_2 = x_1.(-2.q_w.q_y + 2.q_x.q_z) + y_1.(2.q_w.q_x + 2.q_y.q_z) + z_1.(q_w^2 - q_x^2 - q_y^2 + q_z^2) \end{cases}$$

Cet algorithme est utilisé tel quel dans le programme. Les molécules sont constituées de points d'interaction et d'un centre de gravité. A chaque pas de temps, la position 3D des points d'interaction est obtenu par rotation (quaternion) au niveau de l'origine du repère absolu puis translation pour retrouver le repère local associé au centre de gravité de la molécule.

➤ **Composition des rotations-quaternions** Prenons l'exemple d'une application à la mécanique, avec un repérage par les angles d'Euler (précession ψ , nutation θ , rotation propre φ), ce qui donne lieu à 3 rotations successives dans l'ordre indiqué d'abord ψ puis θ puis φ , autour d'axes différents, rotations dont nous notons $q_1(\psi)$, $q_2(\theta)$, $q_3(\varphi)$ les quaternions associés et $M_1(\psi)$, $M_2(\theta)$, $M_3(\varphi)$ les matrices de passage associées.

On obtient :

- la rotation totale R résultant de la composition des 3 rotations successives :

$$R = R(\psi) \circ R(\theta) \circ R(\varphi)$$

- la matrice de la rotation globale résultant des 3 rotations dans l'ordre ψ puis θ puis φ , soit la matrice de passage :

$$M = M_1(\psi).M_2(\theta).M_3(\varphi)$$

- le quaternion Q associé, soit le produit (on remarque l'ordre inverse du produit des matrices de passage) :

$$q = q_3(\varphi).q_2(\theta).q_1(\psi)$$

En effet Si V est un vecteur initial, V_1 son image par la rotation ψ , V_2 l'image de V_1 par celle en θ et V_3 l'image de V_2 par celle en φ , alors :

$$\left. \begin{array}{l} V_3 = q_3(\varphi).V_2.\bar{q}_3(\varphi) \\ V_2 = q_2(\theta).V_1.\bar{q}_2(\theta) \\ V_1 = q_1(\psi).V.\bar{q}_1(\psi) \end{array} \right\} \Rightarrow V_3 = q_3(\varphi).q_2(\theta).q_1(\psi).V.\bar{q}_1(\psi).\bar{q}_2(\theta).\bar{q}_3(\varphi)$$

➤ **Conversion quaternion → matrice** Avec un quaternion $q = q_w + iq_x + jq_y + kq_z$ la matrice équivalente pour représenter la même rotation est :

$1 - 2.(q_y^2 + q_z^2)$	$2.(q_x.q_y - q_z.q_w)$	$2.(q_x.q_z + q_y.q_w)$
$2.(q_x.q_y + q_z.q_w)$	$1 - 2.(q_x^2 + q_z^2)$	$2.(q_y.q_z - q_x.q_w)$
$2.(q_x.q_z - q_y.q_w)$	$2.(q_y.q_z + q_x.q_w)$	$1 - 2.(q_x^2 + q_y^2)$

Cela sous-entend un quaternion est normalisé ($q_w^2 + q_x^2 + q_y^2 + q_z^2 = 1$). Si tel n'est pas le cas, il doit l'être avant de faire la conversion.

m_{00}	m_{01}	m_{02}
m_{10}	m_{11}	m_{12}
m_{20}	m_{21}	m_{22}

➤ **Conversion matrice → quaternion** À partir de la matrice de rotation suivante :

Si la trace = $m_{00} + m_{11} + m_{22} + 1 > 0$ alors :

$$q \begin{cases} q_w = \sqrt{1 + m_{00} + m_{11} + m_{22}}/2 \\ q_x = (m_{21} - m_{12})/(4q_w) \\ q_y = (m_{02} - m_{20})/(4q_w) \\ q_z = (m_{10} - m_{01})/(4q_w) \end{cases}$$

Si la trace de la matrice est $<$ ou $=$ à 0, il faut identifier l'élément le plus grand dans la diagonale principale de la matrice et les calculs se font en fonction de la position de cette valeur.

➤ **Conversion quaternion → {axis, angle}** À partir du quaternion $q = q_w + q_x i + q_y j + q_z k$, si on représente une rotation par un vecteur unitaire (x, y, z) et un angle de révolution *angle*, on obtient :

$$\{axis, angle\} \begin{cases} x = q_x / \sqrt{1 - q_w^2} \\ y = q_y / \sqrt{1 - q_w^2} \\ z = q_z / \sqrt{1 - q_w^2} \\ angle = 2 \cdot \arccos(q_w) \\ angle = 2 \cdot \arcsin(\sqrt{q_x^2 + q_y^2 + q_z^2}) \end{cases}$$

➤ **Conversion {axis, angle} → quaternion** La conversion réciproque :

$$q \begin{cases} q_w = \cos(angle/2) \\ q_x = a_x \cdot \sin(angle/2) \\ q_y = a_y \cdot \sin(angle/2) \\ q_z = a_z \cdot \sin(angle/2) \end{cases}$$

avec :

$$\text{un axe normalisé : } a_x^2 + a_y^2 + a_z^2 = 1$$

$$\text{un quaternion tout autant normalisé : } \cos^2(angle/2) + a_x^2 \cdot \sin^2(angle/2) + a_y^2 \cdot \sin^2(angle/2) + a_z^2 \cdot \sin^2(angle/2) = 1$$

A.1.3 Pour aller plus loin...

➤ **Au-delà des quaternions, les nombres hypercomplexes**

À la fin de l'année 1843 John Graves et Arthur Cayley découvrent indépendamment une algèbre de dimension huit : les octonions. Celle-ci n'est plus associative. On peut ainsi créer une infinité d'algèbres du même type en appliquant la construction de Cayley-Dickson à l'algèbre de rang inférieur.

Quelques propriétés intéressantes sont à noter :

- à chaque rang les dimensions des nombres sont doublées
- à chaque rang une propriété est perdue

n	2^n	nom	limite	définition
0	1	réels		
1	2	complexes	perte de la comparaison	
2	4	quaternions	perte de la commutativité	$\forall (x, y) \in \mathbb{R}^2, x \cdot y = y \cdot x$
3	8	octonions	perte de l'associativité	$\forall (x, y, z) \in \mathbb{R}^3, (x \cdot y) \cdot z = x \cdot (y \cdot z)$
4	16	sédénions	perte de l'alternativité*	$\forall (x, y) \in \mathbb{R}^2, x \cdot (x \cdot y) = (x \cdot x) \cdot y$ et $(x \cdot y) \cdot y = x \cdot (y \cdot y)$

* l'alternativité est une propriété plus faible que l'associativité (les quaternions en sont ici clairement l'exemple).

Après les octonions, les algèbres contiennent des diviseurs de zéro ($x.y = 0$ n'implique plus $x = 0$ ou $y = 0$), ce qui implique que leurs multiplications ne conservent plus les normes.

► Les fractales

Considérons le plus simple des polynômes complexes étudiés : $P(z) = z^2 + c$ qui conduit à la suite :

$$z(n+1) = z(n)^2 + c$$

On peut remplacer les nombres complexes par des quaternions dans la formule de l'ensemble de Mandelbrot ou des ensembles de Julia. La représentation graphique de ces ensembles paraît à première vue impossible mais on peut considérer ces objets comme des objets à trois dimensions qui évoluent dans le temps (la quatrième dimension). Donc si l'on donne à une des parties imaginaires une valeur constante, tout se passe comme si l'on regardait un objet à trois dimensions à un moment précis de son évolution. En faisant des constructions graphiques pour des valeurs successives de la variable « temps » il est même possible de faire une animations où l'on voit l'ensemble apparaître progressivement, grandir, changer de forme, puis diminuer jusqu'à disparaître.

► **Les ensembles de Julia quaternioniques** Soient c une constante complexe arbitraire et z une variable également complexe. On calcule la valeur du polynôme pour une valeur de départ de z puis on donne à z la valeur ainsi trouvée et on recommence le calcul avec cette nouvelle valeur. Le résultat est à nouveau injecté dans z et on recommence, théoriquement un nombre infini de fois. Pour certaines valeurs de départ de z le résultat se maintient au fil de itérations successives dans un intervalle bien limité. Au contraire pour d'autres valeurs la fonction diverge et le point représentant le résultat s'échappe vers l'infini. En fait on démontre que le calcul divergera si au cours des itérations successives le module de z dépasse la valeur 2, ce qui permet d'arrêter le calcul dès que cette valeur est atteinte.



FIG. A.1.3 – L'ensemble de Julia quaternionique pour $q(n+1) = q(n)^2 + c$, avec $c = (-0.747, -0.256, -0.3, 0.0)$

Les points correspondants forment un ensemble bien défini dans le plan complexe, qu'on a pris l'habitude d'appeler ensemble de Julia. Notons qu'il y a une infinité d'ensembles de Julia puisqu'on peut donner n'importe quelle valeur à la constante c .

► **L'ensemble de Mandelbrot quaternionique** Supposons maintenant que dans le polynôme précédent c ne soit plus une constante, mais une variable et représentons le résultat dans le plan complexe de c (et non dans le plan complexe de z comme dans le cas des ensembles de Julia). Pour chaque valeur de c , c'est-à-dire pour chaque pixel de l'écran, itérons le polynôme en partant de la valeur $z=0$ et cherchons l'ensemble des points pour lesquels le polynôme ne diverge pas. Nous obtenons un nouvel ensemble qui a été découvert et étudié par Mandelbrot vers 1980.



FIG. A.1.4 – Ensembles de Mandelbrot classique (2D) et quaternionique (4D)

A.2 Les modélisations différentielles

En mathématique, une équation différentielle est une **relation entre une ou plusieurs fonctions inconnues et leurs dérivées**. Soit E un espace vectoriel de dimension finie. Par définition, une équation différentielle est une équation de la forme suivante $F(x, y, y', \dots, y^{(n)}) = 0$ où F est une fonction continue sur un ouvert U de $\mathbb{R} \times E^n$, appelé domaine. L'ordre de cette équation différentielle est l'ordre n de la plus haute dérivée y apparaissant. Soient y une fonction de x définie d'un intervalle I dans E et $y', y'', \dots, y^{(n)}$, les dérivées successives de la fonction y . Cette fonction y est dite solution si elle est de classe \mathcal{C}^n et si $\forall x \in I, F(x, y(x), y'(x), \dots, y^{(n)}(x)) = 0$. Résoudre une équation différentielle revient à trouver les fonctions solutions y .

Par exemple, l'équation différentielle $y'' + y = 0$ a une solution générale de la forme : $y(x) = A.\cos x + B.\sin x$, où A, B sont des constantes (qu'on peut déterminer si on ajoute des conditions initiales).

Les EDO

Une équation différentielle ordinaire (EDO) est une équation contenant une variable dite indépendante t (ou souvent note aussi par x), un nombre fini de fonctions de t , appelé variables dépendantes, $y_k(t)$ ou y_k et leurs dérivées par rapport à la variable indépendante t , notée par $\dot{y}_k(t)$ (notation habituelle si t signifie le temps) ou $y'_k(t)$.

$$y' + 2ty = 0 \quad (\text{A.2.1})$$

$$F(t, y, y') = 0 \quad (\text{A.2.2})$$

$$F(t, y, y', \dots, y^{(n-1)}, y^{(n)}) = 0 \quad (\text{A.2.3})$$

$$y^{(n)} = f(t, y, y', \dots, y^{(n-1)}) \quad (\text{A.2.4})$$

Considérons l'équation différentielle donnée par (A.2.1) où t est la variable indépendante, et y la variable dépendante. On vérifie aisément que la fonction $y = \exp^{-t^2}$ satisfait cette équation. Il en est de même avec la famille de fonctions $y = Ce^{-t^2}$ pour tout constante C réelle. L'équation (A.2.1) est une équation différentielle du premier ordre. La forme générale d'une équation différentielle du premier ordre est (A.2.2). La fonction $y = y(t)$ s'appelle solution de (A.2.2) sur un intervalle J , si $y'(t)$ existe pour tout $t \in J$ et $F(t, y(t), y'(t)) = 0$ pour $t \in J$. Si une solution de (A.2.2) est constante pour tout t , i.e. $y(t) = \bar{y}$ et $F(t, \bar{y}, 0) = 0$, on dit que \bar{y} est un point d'équilibre ou point stationnaire de (A.2.2). La forme générale d'une équation différentielle d'ordre n est (A.2.3). Une solution de (A.2.3) sur un intervalle J est une fonction n fois dérivable qui satisfait (A.2.3) pour $t \in J$. On appelle (A.2.3) une équation différentielle explicite, si elle s'écrit comme (A.2.4). Autrement on l'appelle une équation différentielle implicite. L'équation (A.2.4) s'appelle une équation autonome si f ne dépend pas de t .

Un système d'équations différentielles ordinaires contient plusieurs variables dépendants. Par exemple, considérons (A.2.5) qui est un système d'équations du premier ordre de deux équations

différentielles. Une solution de (A.2.5) est donnée par $y = (y_1, y_2) = (\sin(t^2), \cos(t^2))$.

$$\begin{cases} y_1' - 2ty_2 = 0 \\ y_2' + 2ty_1 = 0 \end{cases} \quad (\text{A.2.5})$$

Souvent nous cherchons des solutions qui satisfont certaines conditions supplémentaires. Par exemple, un problème typique lié à l'équation (A.2.4) est de trouver une solution $y(t)$ telle que $(y(t_0), y'(t_0), \dots, y^{(n-1)}(t_0)) = (y_0, y_0', \dots, y_0^{(n-1)})$ pour certains valeurs de t_0 et $(y_0, y_0', \dots, y_0^{(n-1)})$. C'est le problème du Cauchy pour l'équation (A.2.4) relatif aux conditions initiales $(t_0, y_0, y_0', \dots, y_0^{(n-1)})$.

Les EDP

Dans le cas de plus qu'une variable indépendante on étudie en général une équation aux dérivées partielles (EDP). Par exemple, considérons l'EDP donnée par (A.2.6) où y est une fonction de deux variables indépendantes t et x . On vérifie aisément que $y(t, x) = e^{-4tx}$ est une solution de (A.2.6) ou plus généralement $y(t, x) = e^{-4tx}u(x-t)$ pour toute fonction dérivable u . Alors que les ensembles de solutions d'une équation différentielle ordinaire sont paramétrés par un ou plusieurs paramètres correspondant aux conditions supplémentaires, dans le cas des EDP les conditions aux limites se présentent plutôt sous la forme de fonction; intuitivement cela signifie que l'ensemble des solutions est beaucoup plus grand, ce qui est vrai dans la quasi-totalité des problèmes.

$$y' + 2ty = 0 \quad (\text{A.2.6})$$

Souvent des équations différentielles ordinaires apparaissent si on cherche des solutions particulières des EDP. Par exemple, si on cherche solutions de la forme $y(t, x) = z(x+t)$ de (A.2.6) on retrouve l'équation différentielle (A.2.1). De plus on peut interpréter des EDP comme des EDO avec un nombre infini de variables dépendantes.

A.3 Les méthodes d'intégration

A.3.1 Difficultés

On peut employer différentes méthodes d'intégration, toutes demandent d'adapter le pas de temps d'intégration à la valeur des accélérations. Si le pas de temps est trop petit, on perdra du temps de calcul alors que s'il est trop grand le système divergera. La difficulté à intégrer dépend de l'amplitude de l'accélération et de sa direction. Un schéma de Newton-Cotes intégrera ainsi parfaitement une accélération constante (la gravité par exemple), quelle que soit son intensité. Que la direction de celle-ci change un peu au cours du temps et il faudra réduire de façon drastique le pas de temps.

A.3.2 Deux familles de méthodes d'intégration

Il existe deux familles de méthodes pour résoudre ces équations : explicites et implicites. Les méthodes de résolution explicites ont des propriétés de stabilité très contraintes par la condition de Courant Frederick Levy qui impose de petits pas de progression temporelle. À l'opposé, les méthodes implicites disposent d'une stabilité inconditionnelle (théorique) qui permet l'utilisation de grands pas de progression temporelle.

Les méthodes explicites sont les plus simples pour les résoudre mais comportent de mauvaises propriétés de stabilité : la convergence n'est assurée que si le pas d'intégration temporelle est petit, ce qui implique un nombre important d'itérations. Les schémas explicites sont donc en pratique peu utilisés. La formulation implicite est beaucoup plus efficace car il n'y a pas de restrictions sur le pas. Cette formulation, bien que nécessitant la résolution d'un système linéaire, reste cependant plus rapide.

Les deux familles de méthodes d'intégration numériques, i.e. explicite et implicite peuvent être divisées en deux catégories : les méthodes à pas unique et à pas multiples. Les méthodes à pas multiples ont recours à au minimum deux valeurs précédemment calculées pour obtenir la valeur suivante. Elles ne sont pas par conséquent adaptées à l'animation car les anciennes valeurs sont inconsistantes dans le cadre d'interactions utilisateurs ou bien de collisions.

A.3.3 Intégration explicite

Les méthodes d'intégration explicites sont les plus simples et s'appuient directement sur le développement de Taylor donnant la nouvelle position p' d'un point p comme égale à $p' = p + vdt + adt^2 + o(dt^2)$ pour un pas de temps dt donné.

En plus de la méthode d'Euler, voici certaines des autres méthodes implémentées dans les codes de dynamique moléculaire :

- algorithme de Verlet
- algorithme de vitesse Verlet
- algorithme de Beeman
- algorithme Leap-Frog

La plus simple d'entre elles est la **méthode d'Euler**, du nom de son inventeur Leonhard Euler, qui n'utilise que des développements limités d'ordre 1 pour écrire :

$$v_{t+dt} = v_t + a_t dt$$

$$p_{t+dt} = p_t + v_t dt$$

Simpliste, elle offre néanmoins parfois de bons résultats. L'erreur commise est donnée par la différence entre cette intégration et le développement de Taylor de p_{t+dt} :

$$\frac{1}{2} p_t'' dt^2 + O(dt^3).$$

Pour de petits dt , l'erreur dominante par pas est proportionnelle à dt^2 . Pour résoudre le problème sur un laps de temps donné t , le nombre de pas doit être proportionnel à $1/dt$ et ainsi l'erreur totale à l'issue de ce temps t sera proportionnelle à dt (l'erreur par pas * le nombre de pas). Pour cette raison, la méthode d'Euler est dite de premier ordre. Ce qui rend cette méthode moins efficace (pour de petit dt) que d'autres techniques d'ordre plus grand comme les méthodes de Runge-Kutta ou les méthodes linéaires multipas.

En faisant dépendre la nouvelle position de la nouvelle vitesse et non plus de celle du pas précédent, on obtient le **schéma d'Euler modifié**. En pratique beaucoup plus stable que le précédent. Il a été démontré comme étant d'ordre 4 par Provost dans sa thèse [Pro97] et donne souvent des résultats supérieurs à ceux de Runge- Kutta 2.

$$v_{t+dt} = v_t + a_t dt$$

$$p_{t+dt} = p_t + v_{t+dt} dt$$

On pourra parfois lui préférer la **méthode de Newton-Cotes** :

$$v_{t+dt} = v_t + a_t dt$$

$$p_{t+dt} = p_t + v_t dt + adt^2$$

qui présente l'intéressant avantage d'intégrer correctement les accélérations constantes (la gravité par exemple).

Une autre méthode souvent implémenté dans les codes de dynamique moléculaire est l'**algorithme de Verlet**, du nom du physicien français Loup Verlet qui le proposa en 1967. Cette

technique offre une meilleure stabilité que les méthodes simples d'Euler en calculant la position au prochain pas à partir des positions du pas courant et passé, sans utiliser la vitesse :

$$p_{t+dt} = p_t + (p_t - p_{t-dt}) + adt^2 = 2p_t - p_{t-dt} + adt^2$$

La vitesse à chaque pas n'est donc calculé qu'au pas suivant :

$$v_t = \frac{p_{t+dt} - p_{t-dt}}{2dt}$$

Un algorithme dérivé de celui-ci est l'**algorithme de vitesse Verlet** qui propose une approche similaire mais inclut la vitesse :

$$v_{t+dt} = v_t + \frac{a_t + a_{t+dt}}{2} dt$$

$$p_{t+dt} = p_t + v_t dt + \frac{1}{2} a_t dt^2$$

Il existe des méthodes d'ordres plus élevés, offrant plus de précision, mais nécessitant en contrepartie le calcul de la force pour des positions intermédiaires entre t et $t + dt$. L'ordre de ces méthodes dépend du nombre de calculs de forces supplémentaires à effectuer. Les plus utilisées sont celles de Runge-Kutta, d'ordre 2 et 4 (voir [PTVF92] pour plus de détails). Plus chères, elles sont également mal appropriées aux simulations dynamiques faisant apparaître des collisions et où apparaissent des discontinuités dans le mouvement.

A.3.4 Intégration implicite

L'intégration implicite fut remise au goût du jour par Baraff et Witkin dans [BW98]. Elle donne d'excellents résultats pour des forces élevées que l'intégration explicite ne sait gérer, ou au prix d'un pas de temps réductible.

L'idée consiste à ne plus trouver la nouvelle position en fonction des forces connues à un instant, mais à chercher la position où les forces à $t + dt$ auraient conduit, si on les avait intégrées en partant de la position courante. Plutôt qu'un bond en aveugle dans le futur, on cherche un état où les forces sont compatibles avec la position. Mathématiquement, Euler deviendrait :

$$v_{t+dt} = v_t + a_{t+dt} dt$$

$$p_{t+dt} = p_t + v_{t+dt} dt$$

Cela demande donc de connaître a_{t+dt} pour trouver p_{t+dt} , ce qui paraît impossible puisque la force dépend elle-même de la position.

On écrit alors un développement limité à l'ordre 1 de l'expression de la force en fonction de la position

$$f' = f + \frac{\partial f}{\partial p} dp$$

qui va permettre de connaître approximativement la force d'une position proche. Puisque la force dépend de la position d'un point et de celle de ses voisins, l'expression $\frac{\partial f}{\partial p}$ va être représentée par une matrice qu'il va falloir inverser.

Le coût de l'inversion de la matrice à chaque pas de temps est largement compensé par le gain en stabilité de la méthode et par le pas de temps qui peut alors être employé, en comparaison avec le pas de temps infime nécessaire avec de fortes amplitudes de forces dans une intégration explicite.

Desbrun a repris cette méthode en l'accélérant jusqu'à l'obtention de temps réel en approximant la matrice jacobienne $\frac{\partial f}{\partial p}$ comme constante et donc pré-inversible [DSB99]. Il montre que cela revient notamment à ajouter une viscosité artificielle.

Miller a adopté la méthode intitulée forward Euler pour animer des vers et des serpents [Mil88]. Les schémas Runge-Kutta, d'ordre 2 et plus, ont recours à des valeurs intermédiaires, engendrant

une précision et une stabilité meilleure. L'intégration de Verlet, aussi connue sous d'autres appellations, est probablement la méthode la plus employée en moteurs physiques (voir [BFA02, KANB03]). Tout en conservant la simplicité de la méthode de forward Euler, l'intégration de Verlet est plus stable.

Des travaux portent sur la fusion des deux grandes familles IMPlicite/EXplicite, i.e. IMEX, pour profiter des avantages de chacune.

A.3.5 Analyses :

il ne suffit pas de proposer un système pour intégrer les équations différentielles, encore faut-il pouvoir évaluer la qualité des méthodes pour des comparaisons par exemple. Les trois notions abordées dans cette analyse :

- la convergence : est-ce que la méthode approxime la solution,
- la précision : avec quelle efficacité la méthode approxime la solution,
- la stabilité : comment sont gérés les erreurs.

convergence : une méthode numérique est dite convergente si la solution numérique approche la solution exacte avec un pas h tendant vers 0. Plus précisément, pour tout $t^* > 0$

$$\lim_{h \rightarrow 0^+} \max_{n=0,1,\dots,[t^*/h]} \|y_{n,h} - y(t_n)\| = 0$$

Toutes les méthodes citées précédemment convergent. En fait, la convergence est une condition sine-qua-non pour une méthode de résolution. Aussi la stabilité et la précision sont les deux critères principaux pour adopter une méthode d'intégration ; le choix dépend du problème à traiter.

consistance et ordre Supposons la méthode numérique :

$$y_{n+k} = \Psi(t_{n+k}; y_n, y_{n+1}, \dots, y_{n+k-1}; h)$$

L'erreur locale de cette méthode est l'erreur commise par un pas, c'est à dire la différence entre le résultat de la méthode (en supposant qu'il n'y ait pas d'erreur aux pas précédents) et la solution exacte :

$$\delta_{n+k}^h = \Psi(t_{n+k}; y(t_n), y(t_{n+1}), \dots, y(t_{n+k-1}); h) - y(t_{n+k})$$

La méthode est dite consistante si :

$$\lim_{h \rightarrow 0} \frac{\delta_{n+k}^h}{h} = 0$$

La méthode est d'ordre p si :

$$\delta_{n+k}^h = O(h^{p+1}) \quad \text{quand } h \rightarrow 0$$

Une méthode est consistante si elle a un ordre plus élevé que 0 (par exemple la méthode d'Euler a un ordre 1). La consistance est une condition nécessaire mais pas suffisante pour assurer la convergence.

Un concept associé est l'erreur globale, l'erreur accumulée à chaque pas pour atteindre le temps t . D'une manière explicite, l'erreur globale au temps t est $y_N - y(t)$ où $N = (t - t_0)/h$. L'erreur globale d'une méthode à un pas d'ordre p est $O(h^p)$, en particulier si la méthode est convergente. Ce résultat n'est pas forcément valable pour les méthodes multipas.

stabilité : dans certains cas, l'utilisation de méthode standard comme la méthode d'Euler, les méthodes explicites de Runge-Kutta ou les méthodes multipas (e.g., les méthodes d'Adams-Bashforth) provoquent des instabilités dans les solutions, alors que d'autres méthodes peuvent

produire des solutions stables. Le comportement délicat des solutions, ne provenant pas nécessairement d'une équation complexe, est décrit comme une rigidité de l'équation souvent causée par différentes échelles de temps dans le processus décrit. Ces problèmes "rigides" se rencontrent souvent dans les cinétiques chimiques, la théorie du contrôle, la biologie, les mécaniques du solide ou encore les prédictions météorologiques.

Chapitre B

Quelques éléments de modélisation par SMA

Dans cette partie des annexes, nous complétons certains sujets abordés dans le chapitre 2 lors de la présentation des SMA. En l'occurrence, nous regroupons différentes plateformes SMA susceptibles d'être utilisées dans le cadre de la modélisation de processus biologiques selon le niveau de programmation requis. Ensuite, nous présentons différentes techniques de partitionnement de l'espace exploitables pour un SMA. Ces différentes techniques tiennent compte de la répartition des agents et se révèlent plus ou moins adaptées en fonction de la dynamique du système modélisé.

B.1 Les plateformes génériques

D'une manière générale il y a plusieurs plates-formes de SMA qui peuvent être classées comme suivant :

- les agents mobiles (Voyager, Odissey, Aglet, etc.),
- les agents cognitifs (AgentBuilder, etc.) avec la norme FIPA¹ (Jade, FIPA-OS, etc.), ou la norme KQML (JAT, JAT-Lite, etc.),
- les agents collaboratifs (Zeus, JAFMAS, KAoS, JAFIMA, etc.),
- les simulations de SMA (NetLogo, Cormas, Swarm, etc.).

Toutes ces plates-formes sont destinées à un type d'agent ou à un type d'application (simulation, mobilité). Il n'y a pas encore de plateforme générique "universelle" et les modélisateurs/concepteurs doivent faire un choix pertinent.

B.2 Les plateformes à programmation simplifiée

Les plateformes à programmation simplifiée (de l'anglais *computer-aided software engineering* ou CASE) désignent les environnements de développement graphiques qui facilitent la création rapide de logiciels. Les applications générées proposent en général un compromis entre ergonomie, fonctionnalités et vitesse d'exécution. Elles ont le plus souvent une interface ergonomique permettant une construction simplifiée (voire graphique) des agents avec une génération de code et une exécution transparentes. Proposant plus ou moins de choix concernant le modèle des agents (cycle de vie, interactions, etc..) et les paramètres des simulations, elles fournissent généralement différents moyens de visualisation : un rendu direct de la simulation, une génération de vidéo ou alors l'utilisation d'observables pour générer des statistiques globales ou locales.

¹Foundation for Intelligent Physical Agents - www.fipa.org

Ces environnements intégrés de modélisation sont généralement plus qu'un simple langage de programmation. A priori l'utilisateur n'a nul besoin d'être expert dans le langage de programmation sous-jacent. Cependant, il se peut parfois que le modélisateur ait à appréhender une interface complexe qui peut se révéler tout aussi compliquée qu'un langage de programmation à part entière, et ce, même si certaines plate-forme propose des langages de programmation "visuels" (StarLogo TNG).

Le tableau (Tab.B.2.1) présentes différentes plates-formes de modélisation.

	langage	OS	licence	commentaires	dernière version	màj
CORMAS ^a	Cincom VisualWorks Smalltalk	multiplateforme	charte	développé par le CIRAD - dédié à la gestion de ressources naturelles	? / ? / 2004	
HSIM ^b	C++ (+langage de script)	Linux	gratuit mais pas encore opensource	utilise un ensemble de règles pour modéliser les interactions entre les différentes protéines	??	
Mobidyc ^c	Cincom VisualWorks Smalltalk	multiplateforme	GPL ^d	développé par l'INRA Avignon - champs d'application : écologie, biologie et environnement	04/042006 V2.22	
NetLogo (Net-Logo3D) ^e	Java/Logo	multiplateforme	gratuit pour des buts éducatifs et de recherche	large communauté, de nombreux tutoriaux et modèles disponibles dans une grande gamme de domaine d'application	16/02/2007 v3.1.4	
SeSAM ^f	Java	multiplateforme	LGPL ^g	NetLogo'like	24/06/2005 v1.9.2	
BioDyn ^h	C++/python	Windows	gratuit mais pas encore opensource	plusieurs versions dont 2D, 3D et éducative (interface simplifiée)	2007 v3	
StarLogo (StarLogo TNG for 3D) ⁱ	Java/Logo	multiplateforme	gratuit pour des buts éducatifs mais pas encore opensource	extension du langage de programmation Logo créé par le MIT pour des buts éducatifs - propose un langage de programmation visuel innovant dans la version 3D StarLogo TNG - dispose d'un livre dédié <i>Adventures in Modeling</i>	29/07/2006 v3	

TAB. B.2.1 – Différentes plates-formes ou environnement intégré de modélisation offrant un accès simplifié à l'expérimentation in-silico.

^aCORMAS, *Common-pool Resources and Multi-Agent Systems*, <http://cormas.cirad.fr/indexeng.htm>

^bHSIM, <http://www.lri.fr/~pa/BioInformatics/>

^cMobidyc, *MOdélisation Basée sur les Individus pour la DYNAMique des Communautés*, http://www.avignon.inra.fr/internet/unites/biometrie/mobidyc_projet/

^dGPL : the GNU General Public License, <http://www.gnu.org/copyleft/gpl.html>

^eNetLogo, Wilensky, U. (1999), <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL

^fSeSAM, *Shell for Simulated Agent Systems*, <http://www.simsesam.de/>

^gLGPL : la GNU Lesser General Public License, le successeur de la GPL

^hBioDyn, <http://pagesperso.univ-brest.fr/~ballet/>

ⁱStarLogo, <http://education.mit.edu/starlogo/>

B.3 Les plateformes à programmation avancée

Les plateformes à programmation avancée proposent des bibliothèques ou paquets capable de gérer la physique d'un monde virtuel ou la communication entre les agents. Ces bibliothèques ont l'avantage de fournir des fonctions génériques sans avoir à tout coder soi-même, mais en contre partie il est nécessaire d'avoir les connaissances suffisantes pour au moins les exploiter correctement voire les augmenter si besoin est et si la licence le permet (ce qui est généralement le cas).

Le tableau (Tab.B.3.1) indique différentes plateformes ou bibliothèques en précisant certaines des fonctionnalités assurées.

B.4 Les méthodes de discrétisation de l'espace

Partant d'une revue sur les différentes méthodes de partitionnement [GG98], nous présentons dans ce qui suit certaines des plus communes.

	langage	OS	licence	commentaires	dernière version	maj
BREVE ^a	steve ^b	multiplateforme	GPL	inclut un moteur physique avec détection des collisions et un moteur de rendu OpenGL	22/01/2007 v2.5.1	
DIETAgents ^c	Java	multiplateforme	GPL	les agents sont autonomes tout en étant légers, ce qui permet de simuler plus de 100 000 agents dans une seule machine virtuelle (JVM)	14/03/2005 v0.97	
Madkit ^d + OdeJava ^e	Java/C	multiplateforme	GPL / LGPL / BSD ^f	implémente un modèle AGR (Agent/Group/Rôle) ^g - exploite le moteur physique ODE ^h	22/11/2005 v4.1.2 30/10/2004 v0.8	
Swarm ⁱ	Java, Objective-C	multiplateforme	GPL	à l'origine développé par le Santa Fe Institute - comprend un ensemble de bibliothèques	28/10/2005 v2.2	

TAB. B.3.1 – Différentes plates-formes à programmation avancée.

^aBREVE, <http://www.spiderland.org/>

^bsteve est un langage orienté objet, http://www.spiderland.org/breve/breve_docs/docs/steve.html

^cDIETAgents, <http://diet-agents.sourceforge.net/>

^dMadkit, <http://www.madkit.org/>

^eOdeJava, *Open Dynamics Engine binding for Java*, <http://odejava.org/>

^fBSD licence est un acronyme pour la "Berkeley Software Distribution license agreement" et est une des licences les plus utilisées pour les programmes gratuits (sous ensemble des programmes opensources).

^gAGR (Agent/Group/Rôle) est un modèle organisationnel : les agents sont situés dans un groupe et jouent un rôle.

^hOpen Dynamics Engine (ODE) : une bibliothèque logicielle open source se plaçant dans la catégorie des moteurs physiques et servant à simuler l'interaction physique de corps rigides. <http://www.ode.org/>

ⁱSwarm, <http://www.swarm.org/>

la grille : le partitionnement de l'espace est ici régulier, à priori fixe, non hiérarchique et utilise simultanément $d*n$ hyperplans pour diviser un espace à d dimensions en n^d carrés, voxels ou encore partitions (Fig.B.4.1). Avant le réel calcul des distances, un pré-filtre parmi les éléments permet de réduire le nombre de calculs des distances de manière sensible.

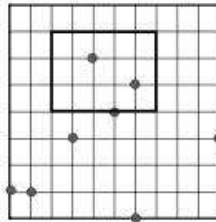


FIG. B.4.1 – Partitionnement par grille.

les arbres de partitionnement binaire de l'espace : (en anglais : BSP trees pour Binary Space Partitioning trees) à chaque itération l'espace à d dimensions est scindé en deux par un hyperplan de dimension $d - 1$ se bornant à ne découper qu'une seule partition et non tout l'espace (Fig.B.4.2). La position des hyperplans dépend de la position des éléments dans l'espace et généralement, un paramètre seuil sur le nombre maximal d'élément par feuille contrôle l'arrêt du partitionnement.

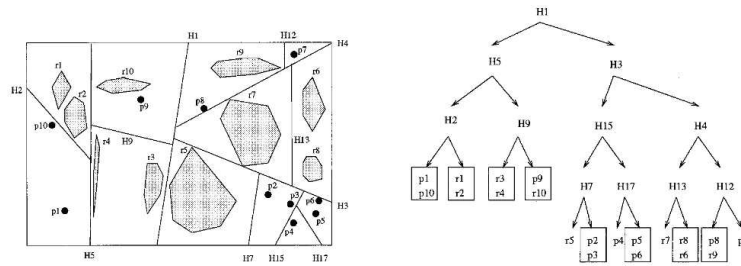


FIG. B.4.2 – Partitionnement par arbre BSP (Binary Space Partitioning).

les arbres kd ou kd-arbres : (en anglais kd-trees pour k-dimensional tree) est un cas spécial des arbres BSP vus précédemment : les hyperplans "locaux" sont iso-orienté, c'est à dire qu'ils sont colinéaires à un sous-ensemble de la base de l'espace (Fig.B.4.3). La lettre k se réfère à la dimension de l'espace et un 3d-tree est tout simplement un octree.

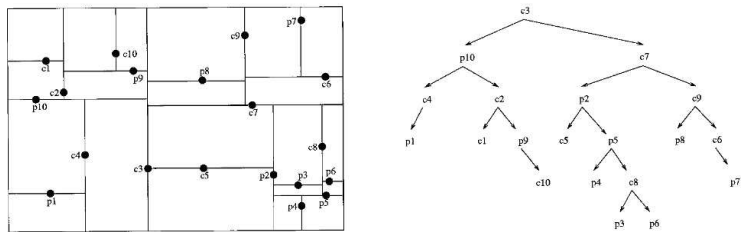


FIG. B.4.3 – Partitionnement par kd-tree.

A noter que certaines méthodes cherchent à équilibrer de tels arbres pour diminuer encore le temps moyen de recherche, notamment en mixant les kd-trees avec des b-trees qui sont par construction des arbres équilibrés : les structures de données sont alors des kdb-trees (Fig.B.4.4). Attention, les b-arbres ne sont pas nécessairement des arbres binaires.

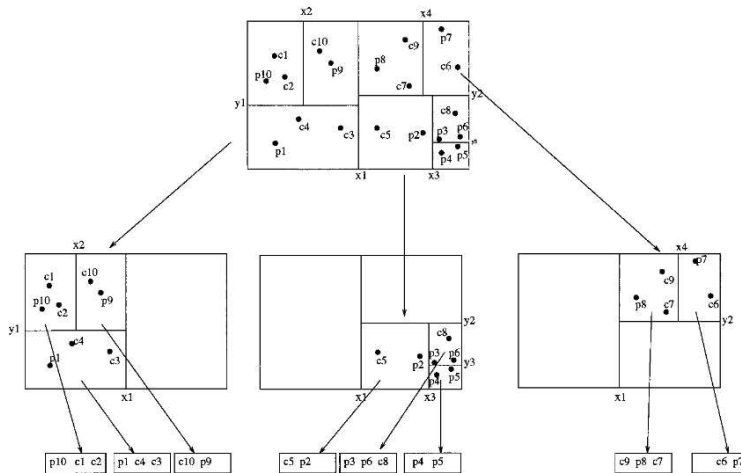


FIG. B.4.4 – Partitionnement par kdb-tree (pour k-dimensionnal b-arbre).

les quadrees : le partitionnement divise une partition mère en quatre partitions filles en utilisant des hyperplans réguliers "locaux" de dimension 1 (Fig.B.4.5). Le processus récursif ne s'arrête que lorsque chaque feuille ou quadrant n'a au plus qu'un seul élément associé.

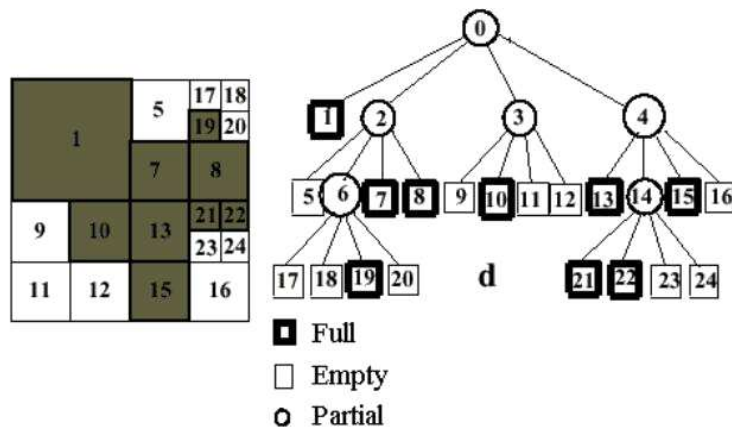


FIG. B.4.5 – Partitionnement par quadtree.

Les hyperplans peuvent systématiquement passer par des éléments de l'espace (Fig.B.4.6(a)) ou alors ne jamais en contenir (Fig.B.4.6(b)).

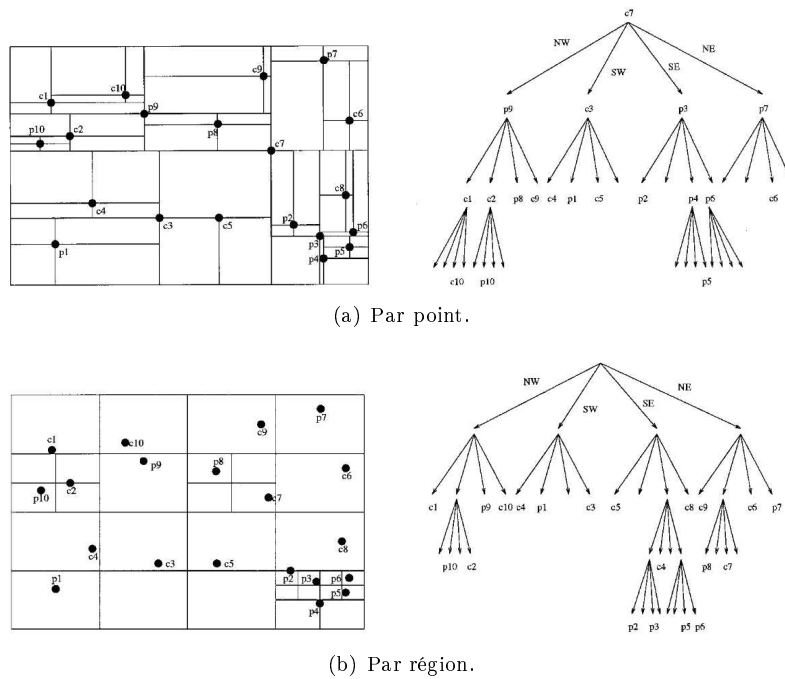


FIG. B.4.6 – Quadtree.

les octrees : sont à la 3d ce que les quadtrees sont à la 2d (Fig.B.4.7). Ici une itération du partitionnement se fait à l'aide de 3 hyperplan de dimension 2.

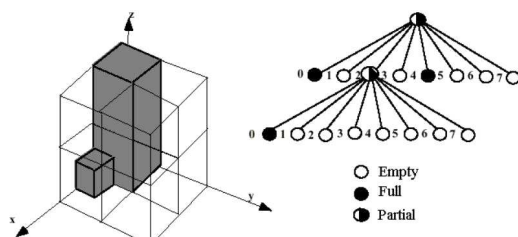


FIG. B.4.7 – Partitionnement par octree.

les R-arbres ou R-trees : le partitionnement génère une structure hiérarchique avec des partitions qui peuvent être recouvrantes entre-elles ([Gut84] et plus récemment [MNPT06]). Deux bornes encadrent le nombre de partitions filles autorisées par partition mère. L'arbre est parfaitement équilibré ; toutes les feuilles se trouvent être à la même profondeur (Fig.B.4.8). Cette structure désavantage complètement le temps de construction ou de mise à jour au profit d'une efficacité accrue pour la recherche. Dès lors que certaines partitions se recouvrent, le calcul d'attribution des agents à leur partition devient plus coûteux puisqu'on ne peut plus se permettre de ne travailler que sur les coordonnées des éléments et qu'il faut passer en revue toutes les partitions.

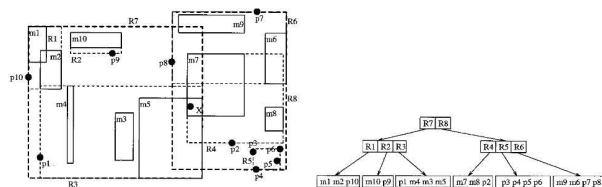


FIG. B.4.8 – Partitionnement par R-arbre ou R-tree.

le PAT ou Principal Axis Tree : est une technique de partitionnement de l'espace proposée par McNames[McN01] et basée sur l'analyse en composantes principales (ou ACP) qui, pour un nuage d'éléments donnés, est capable de proposer les directions qui exprimeront le mieux la dispersion des éléments. En d'autre terme, c'est ici statistiquement le meilleur partitionnement de l'espace (Fig.B.4.9). Néanmoins, le coût de construction de la structure et de la répartition des éléments dans les partitions est trop important par rapport au gain inhérent à l'équilibre des éléments dans les feuilles de cet arbre.

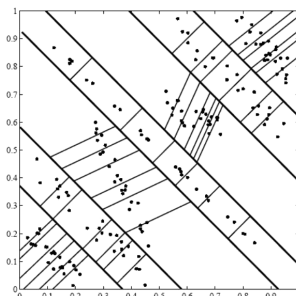


FIG. B.4.9 – Ce PAT (Principal Axis Tree) ou arbre en composantes principales a une profondeur de 2 et divise une partition mère en 7 partitions filles. L'arbre ainsi construit est statistiquement le mieux équilibré, surtout pour des distributions non-homogènes et cependant périodiques comme ici.

Index

- échelle de modélisation, 24
- 3DSpi, 47
- automates cellulaires, 32
- BioDyn, 46
- Breve, 45
- coarse-graining, 59
- Dissipative Particle Dynamics (DPD), 93
- dynamique moléculaire, 55
- dynamique moléculaire - bending ou de flexion, 57
- dynamique moléculaire - champs de force, 56
- dynamique moléculaire - interaction électrostatique, 58
- dynamique moléculaire - interaction de van der Waals, 58
- dynamique moléculaire - stretching ou d'élongation, 57
- dynamique moléculaire - torsion, 57
- EDO, 28
- EDP, 30
- HSIM, 46
- mécanique moléculaire, 55
- MMm - architecture, 109
- modélisation à grains d'atomes, 59
- modélisation couplage de modèles, 26
- modélisation différentielle - EDO, 28
- modélisation différentielle - EDP, 30
- modélisation individu-centrée, 34
- modélisation logique, 30
- modélisation multi-échelles, 24
- modélisation multi-paradigmes, 24
- modélisation par automates cellulaires, 32
- modélisation par réseaux de Petri, 30
- mouvement brownien, 95
- NetLogo, 44
- paradigme de modélisation, 24
- réseaux de Petri, 30
- SMA, 34
- SMA - équation du mouvement, 92
- SMA - agent cognitif, 37
- SMA - agent réactif, 36
- SMA - gestion de l'espace, 37, 91, 173
- SMA - gestion du temps, 40, 92, 96
- SMA - les plateformes, 42, 43, 46, 171, 172
- SMA - modèles à grains d'atomes, 83
- StarLogo, 43