INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque

|__|__|__|__|__|__|__|__|__|__|

T H E S E

pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité : « Micro et Nano Electronique »

préparée au laboratoire **TIMA** dans le cadre de

l'Ecole Doctorale d'«*Electronique, Electrotechnique, Automatique, Télécommunications, Signal* »

présentée et soutenue publiquement

par

Eleftherios KOLONIS

le 10 Janvier 2006

Titre :

SYSTEMES COMPLEXES, OUTILS DE CAO ET NANOTECHNOLOGIES

Directeur de thèse :
Michael NICOLAIDIS

JURY

M. René David	Président
M. Giovanni De Micheli	Rapporteur
M. Jean Arlat	Rapporteur
M. Michael Nicolaidis	Directeur de thèse
Mme Lorena Anghel	Co-encadrant
M. Aurélien Barrau	Examinateur

Table de matières

Introduction 5

1 Une plateforme de CAO pour les nanotechnologies 7

1.1 Technologies de fabrication nanométriques 9
1.1.1 Solutions alternatives 9
1.1.2 Méthodes de fabrication des nano ordinateurs 12
1.2 Une proposition de CAO pour les nanotechnologies 16
1.2.1 Outil de synthèse 21
1.2.2 Outil de simulation 22
1.3 Cas d'étude pour l'outil de simulation 23
1.4 Solutions architecturales et l'outil d'exploration 24
1.5 Conclusions 31

2 Présentation de l'outil de simulation logicielle 33

2.1 Types d'applications 34
2.1.1 Systèmes cibles 35
2.2 Utilisation de l'outil 38
2.2.1 Description du système cible 39
2.2.2 Initialisation du système 43
2.2.3 Description des visualisations 46
2.2.4 Exécution des simulations 48
2.3 Structure de l'outil 50
2.3.1 Programme central de l'outil 52
2.3.2 Fichier Java de l'application 53
2.3.3 Fichier exécutable de la simulation 57
2.3.4 Bibliothèque d'applications 57
2.4 Conclusions 59

3 Ecosystèmes artificiels 61

3.1 Variables 64
3.2 Fonctions 65
3.3 Possibilités de l'application d'écosystèmes pour créer une variété de systèmes cibles 67
3.3.1 Module d'expression génétique 67
3.3.2 Module de reproduction des organismes et de création des organismes 71
3.3.3 Module de mouvement 72
3.3.4 Module de sélection «naturelle» 73
3.4 Exemples 76
3.4.1 Phylogenèse 76
3.4.2 Création d'organismes artificiels 82
3.5 Conclusions 92

4 Systèmes de particules, lois d'interactions et émergence de l'espace-temps ……95

4.1 Généralités 96
4.1.1 Contraintes des accélérations 97
4.1.2 Description de l'application 99
4.2 Etude de l'Emergence d'un Espace - Temps Relativiste 103
4.2.1 Module de création des objets rigides et des étalons de longueur 110
4.2.2 Module de mesures des longueurs depuis le système de référence S_0 112
4.2.3 Module de mesure depuis le système de référence S des dimensions des objets se trouvant au repos dans S 114
4.2.4 Module de création des horloges 116
4.2.5 Module de mesures des intervalles du temps depuis le système de référence S_0 122
4.2.6 Module de synchronisation des horloges distantes 128
4.2.7 Module de changement de système de référence d'observation 134
4.2.8 Module de réalisation des mesures dans le système de référence S 137
4.2.9 Module de création des objets et processus identiques dans S et S_0 141
4.3 Autres exemples des lois d'interactions 143
4.4 Conclusions 148

5 Conclusions et perspectives 149

Bibliographie 155

Introduction

La poursuite de la miniaturisation dans les technologies microélectroniques, facteur principal de la révolution informatique, aura atteinte ses limites ultimes aux alentours de l'an 2015. Dans un effort mondial de développer des technologies alternatives capables de maintenir la loi de Moore, des nombreuses recherches sont activement conduites sur des nanotechnologies dans de nombreux établissements privés et publiques. Elles devraient révolutionner les systèmes de calcul en intégrant dans un seul composant d'énormes quantités de dispositifs et ceci avec des coûts de fabrications drastiquement plus faibles que les technologies microélectroniques actuelles. Ces tendances auront un impact profond sur les architectures des systèmes de calcul et exigeront un nouveau paradigme de CAO (conception assistée par ordinateur). Dans ce manuscrit, nous proposons une plateforme de CAO, dont l'objectif est de permettre au concepteur de demain d'exploiter de façon efficace l'extraordinaire puissance de calcul prévue d'être apportée par ces technologies. Nous visons en particulier des applications concernant des systèmes complexes composés d'un nombre très élevé des entités simples. Afin de développer cette plateforme de CAO sur une base concrète, nous employons comme cas d'étude de systèmes complexes des écosystèmes artificiels.

Sur cette base, nous décrivons dans un premier temps (chapitre 1) les différents outils nécessaires pour compléter cette plateforme. L'un de ces outils permet à l'utilisateur de décrire facilement, d'expérimenter et de valider par simulation son système cible avant de passer à la phase plus fastidieuse de l'implémentation du système dans un nanoréseau. Le chapitre 2 présente plus en détails cet outil. Les chapitres 3 et 4 montrent la capacité de l'outil à traiter des systèmes complexes de natures variés. En particulier, le chapitre 3 traite le cas des écosystèmes artificiels tandis que le chapitre 4 traite des systèmes composés d'un ensemble des particules dont l'évolution est régis par certaines lois d'interactions. Il illustre la façon dont, sous certaines conditions, les lois d'interactions peuvent donner lieu à l'émergence

d’un espace-temps relativiste. Le chapitre 5 présente des conclusions et des perspectives.

Chapitre 1

1 Une plateforme de CAO pour les nanotechnologies

La miniaturisation de plus en plus poussée des technologies microélectroniques MOS basées sur le silicium, facteur principal de la révolution des technologies d'information, atteindra ses limites ultimes peu après 2015. Les raisons sont aussi bien techniques (courants de fuite, intégrité du signal, dissipation de puissance et réchauffement excessif du circuit associé,...), qu'économiques (coût excessif des chaînes de fabrication) [1]. Transistors à un électron, automates cellulaires quantiques, nano tubes, portes logiques moléculaires, nano cristaux, sont certaines des technologies alternatives candidates pour remplacer les technologies actuelles. Des procédés alternatifs de fabrication de composants nano métriques émergent dans les laboratoires de recherche. Une grande majorité de ces procédés sont des techniques dites « bottom up » qui exploitent les propriétés physiques fondamentales des atomes de s'auto-assembler pour former des molécules et des cristaux.

Un des caractéristiques promotrices des procédés d'auto-assemblage est qu'ils permettent la fabrication de circuits très complexes à un prix réduit et une consommation d'énergie faible, grâce à la sophistication rapide des processus de synthèse chimique. Ils devraient permettre de synthétiser chimiquement les dispositifs électroniques et leurs interconnexions, pour créer des systèmes très complexes et à bas prix. Certaines caractéristiques de tels systèmes sont : une complexité extraordinaire, une régularité élevée, et, hélas, un bas rendement de fabrication. Elles devraient permettre de révolutionner les systèmes de calcul en intégrant des nombres énormes de dispositifs à faible coût. A cause de leurs caractéristiques uniques, ils auront besoin d'un nouveau paradigme de CAO. Ce paradigme devrait considérer les contraintes de ces nouveaux composants et permettre d'employer efficacement leur complexité extraordinaire, afin de créer des systèmes ayant une puissance de calcul sans précédent.

1.1 Technologies de fabrication nanométriques

Au cours des 50 dernières années, la densité des transistors sur les puces des circuits intégrés a augmenté de façon exponentielle, comme prédit par la « loi » de Moore. Grâce aux progrès rapides des technologies des semi-conducteurs, la puissance de calcul des ordinateurs s'est accrue pendant cette période très rapidement. En outre, depuis l'invention du transistor en 1948, il n'y a pas eu de changement fondamental dans les principes de fonctionnement des transistors employés dans nos systèmes électroniques.

Néanmoins, dans les dix années à venir, les technologies actuelles de fabrication des systèmes électroniques atteindront leurs limites ultimes, correspondant à des transistors ayant une longueur de canal de quelques dizaines de nanomètres. Pour ces dimensions, les effets quantiques deviennent dominants et des problèmes techniques (courants de fuites, dissipation de puissance et réchauffement des composants, intégrité du signal, variabilité des paramètres technologiques, ...) interdiront tout miniaturisation subséquente. Cette situation interdira tout progrès supplémentaire sans procéder à un changement technologique profond.

1.1.1 Solutions alternatives

Dans un effort mondial de développer des technologies alternatives capables de maintenir la loi de Moore, de nombreuses recherches sur les nanotechnologies ont été entreprises dans un grand nombre des centres de recherche, privés et publics. Les sections suivantes donnent un aperçu de quelques uns de leurs résultats.

1.1.1.1 SET (Single Electron Transistor)

Comme dans les technologies actuelles, le traitement et le stockage de l'information dans les technologies de transistors à un électron (SET) est basé sur le stockage et le mouvement des électrons. Afin de manipuler les charges électriques électron par électron, ces transistors mettent en jeu le phénomène quantique de l'effet tunnel, et utilisent pour cela des structures appelées jonctions tunnel. Ces structures correspondent à une barrière énergétique séparant deux conducteurs. Les propriétés géométriques et physiques de ces jonctions sont telles qu'il est possible pour une charge de les traverser par effet tunnel. Le perçage de la jonction tunnel d'un nombre restreint d'électrons est contrôlé par le biais d'un champ électrique, lui même contrôlé par un électrode. Les transistors SET ont démontré des propriétés de commutation et d'amplification semblables aux transistors microélectroniques et plusieurs méthodes de fabrication ont été proposées pour ces dispositifs [2], [3], [4].

1.1.1.2 Nanotubes de Carbone

Les nanotubes de carbone sont un sous-produit des recherches sur les fullerènes, qui ont mené à la découverte du C60 "buckyball" par les lauréats du prix Nobel de 1996, Robert F. Curl, Harold W. Kroto, et Richard E. Smalley. Les nanotubes sont constitués des couches de graphite qui s'enveloppent pour créer des formes cylindriques d'une taille de quelques nanomètres de diamètre seulement et, actuellement, jusqu'à un millimètre de longueur. Ainsi, leur rapport de longueur/largeur est extrêmement élevé, fait inédit vu la nature moléculaire des nanotubes.

Les nanotubes de carbone peuvent être conducteurs, semi-conducteurs, ou isolants. Dans l'état semi-conducteur, il a été montré qu'ils disposaient d'une mobilité très élevée, une haute transconductance et de grands parcours-libres moyens [5-10]. Ils sont bien adaptés pour fabriquer des transistors qui fonctionnent à une très haute fréquence. Un nanotube de carbone peut en théorie fonctionner à des fréquences

supérieures au terahertz [11]. Des études et des expérimentations récentes, ont permis un premier progrès vers ce but [12-14]. En conséquence, les nanotubes de carbone sont une alternative prometteuse pour la conception des nano-ordinateurs extrêmement rapides, mais aussi extrêmement intégrés.

1.1.1.3 Les automates cellulaires quantiques (QCA)

Les automates cellulaires quantiques sont également parmi les solutions les plus prometteuses. Ils sont constitués d'une matrice de cellules, composées de quatre points quantiques (quantum dots), positionnés aux coins d'un carré et entourés d'un isolant. Par ailleurs, chaque cellule contient deux électrons en excès qui se repoussent mutuellement et n'ont que deux configurations possibles : les deux coins du carré placés sur l'une de ses diagonales et les deux coins du carré placés sur l'autre diagonale. Nous utilisons l'une de ces configuration pour coder la valeur logique 0 et l'autre pour coder la valeur 1. Comme la cellule est un système quantique, elle peut se retrouver dans une superposition quantique linéaire de ces deux états (0 et 1). Ainsi, une cellule peut coder la plus petite unité d'information quantique, appelée qbit. Comme les cellules sont mises l'une à coté de l'autre, l'état d'une cellule peut affecter l'état des cellules voisines à cause des répulsions entre les électrons qu'elles contiennent. Ainsi, en plaçant des cellules dans diverses configurations nous pouvons créer des portes logiques [15]. Plusieurs implémentations physiques sont proposées pour les qubits [16-24].

Malgré les difficultés de mise au point des automates cellulaires quantiques comportant un nombre suffisamment grand de cellules, le grand intérêt de ces automates réside dans leur potentiel pour effectuer des calculs quantiques, c'est-à-dire de calculs qui opèrent sur des qubits. Comme k qubits codent en même temps 2^k valeurs différentes, un calcul quantique manipulant k qubits permet d'effectuer des calculs opérant en même temps sur 2^k valeurs différentes. Cette propriété fait que le calcul quantique est très prometteur pour la solution de certains problèmes

difficiles comme les problèmes np-complets, pour lesquels le calcul classique reste très inefficace car il nécessite l'énumération de l'ensemble des solutions possibles du problème. Ainsi, les algorithmes quantiques permettant de résoudre divers problèmes d'optimisation, comme ceux découverts par Shor [25] et par Grover [26], suscitent un grand intérêt.

1.1.2 Méthodes de fabrication des nano ordinateurs

Les nouvelles technologies de fabrication sont la clef du progrès dans les nanotechnologies. Dans une certaine mesure, les technologies de fabrication commandent même les idées que les chercheurs se permettent d'imaginer et de proposer. Par conséquent, des ressources et des efforts importants ont été investis ces dernières années pour faire avancer les techniques de fabrication des structures nanométriques.

Grâce à l'expérience accumulée par l'industrie microélectronique mondiale pendant plus d'un demi siècle, et l'énorme infrastructure construite autour des technologies du silicium, la production par cette industrie dans un avenir proche de nano-ordinateurs électroniques, c'est-à-dire utilisant des transistors MOS ayant un canal de quelques dizaines de nanomètres, est quasi-certaine. Par contre, comme nous avons déjà mentionné, la filière MOS atteindra ses limites ultimes aux alentours des années 2012-2015. En outre, un autre aspect doit être considéré. Le coût de fabrication d'un produit est directement lié à son information morphologique. Les matières brutes ont un coût de fabrication minimale qui est proche du coût de l'énergie consommée pour leur fabrication. Les produits fortement sophistiqués tels que les circuits intégrés, correspondent à une information morphologique très élevée. Ainsi, ils exigent des lignes de fabrication fortement sophistiquées et coûteuses, résultant à un coût de fabrication excessif [1].

Grâce aux avancés de notre compréhension des structures fines de la matière (niveaux moléculaires et atomiques), nous pouvons construire des nano structures sans employer le processus de la lithographie. Ces avancés devraient nous

permettre à terme de fabriquer des systèmes de calcul très complexes sans les coûts et/ou les limites de la lithographie ultrafine. Par contre, la méthodologie de conception de ces systèmes ne sera pas obtenue comme une prolongation de la méthodologie de conception des systèmes de calcul basés sur les technologies des transistors MOS. En effet, il est très probable que les processus de fabrication qui seront utilisés pour les nanotechnologies ne vont pas permettre de créer des structures très différenciées pour implémenter une grande diversité de fonctions, comme nous permettent de le faire dans les circuits intégrés les techniques de lithographie. Par exemple, les techniques d'auto-assemblage permettront de créer et d'aligner des dispositifs de taille nanométrique, correspondant à des structures régulières et répétitives, qui ne permettent pas la variabilité des structures requise pour implémenter les fonctions nécessaires pour effectuer des calculs complexes. Par conséquent, ces fonctions seront implémentées après fabrication, en reconfigurant la structure régulière du nano-système, de la manière dont nous configurons actuellement les FPGA. Cette même reconfiguration pourra être utilisée pour masquer les nombreuses parties défaillantes d'un nano-système. Cette situation soulève des nouveaux défis dans la conception des systèmes de calcul du future et peut motiver le développement des architectures de calcul différentes de celles que nous avons trouvé efficaces dans le contexte des technologies conventionnelles du silicium.

Comme, le processus d'auto-assemblage (processus « bottom up ») représente une alternative prédominante [27] pour la fabrication des nano-ordinateurs, nous allons discuter de façon un peu plus détaillée le processus d'auto assemblage, ses caractéristiques et ses contraintes intrinsèques.

1.1.2.1 Auto-assemblage et systèmes de calcul

L'auto-assemblage est une technique de nanofabrication qui est basée sur les propriétés physiques fondamentales des molécules et des cristaux d'auto-assembler des structures avec des dimensions nanométriques à partir des atomes individuels.

Nous pouvons obtenir comme résultat des fils composés de quelques atomes et des éléments actifs qui peuvent être aussi petits qu'une molécule. A titre d'exemple de dispositifs fabriqués par le processus d'auto assemblage nous pouvons citer les nanocables semi-conducteurs [30], les commutateurs moléculaires [28], les SET [29], les points quantiques (quantum dots) [31-33], et les nanotubes de carbone [34-36].

Le processus d'auto-assemblage apparemment préféré par la nature (par exemple formation des cristaux), est un processus « bottom up », considéré comme une alternative prometteuse [27] à l'approche actuellement dominante « top-down » de fabrication de micro et nano systèmes. Par sa nature intrinsèque, ce processus offre un grand nombre d'avantages essentiels. Premièrement, il offre la possibilité, de manipuler les structures au niveau atomique, une des étapes les plus difficiles dans la nanofabrication. En plus, elle peut s'inspirer des exemples biologiques extrêmement riches en procédés, structures et fonctionnalités (l'auto-assemblage est l'une des stratégies les plus importantes utilisées dans la biologie pour le développement des structures complexes et fonctionnelles), voire même d'incorporer directement des structures biologiques comme composants dans les systèmes finaux. Les avantages les plus prometteurs des procédés d'auto-assemblage seraient la possibilité de fabriquer des systèmes très complexes à un prix réduit et une consommation d'énergie faible. Ils devraient permettre de synthétiser chimiquement les dispositifs électroniques et leurs interconnexions, pour créer des systèmes très complexes et à bas pris. Comme cité dans [37], la nouvelle technologie de fabrication des ordinateurs qui sera susceptible de remplacer la technologie actuelle basée sur la lithographie, devrait présenter un grand ensemble d'avantages. Le processus d'auto-assemblage réalise cette condition.

Cependant, tandis que la technologie actuelle de fabrication de systèmes de calcul offre une grande flexibilité en termes de formes géométriques et fonctionnalités associées, la plupart des méthodes récemment développées d'auto-assemblage conduisent à la fabrication des structures fortement homogènes et périodiques [38-40], souvent désignées dans la littérature par le terme de nanocomposant. Comme

nous avons mentionné précédemment, cette limitation nous oblige à utiliser une approche de reconfiguration après fabrication [40-42] afin de différencier les structures homogènes des nanocomposants pour introduire diverses fonctions et/ou masquer leurs défauts [43]. Mais en même temps, cette reconfigurabilité offre une grande flexibilité en permettant d'implémenter pour chaque application l'architecture matérielle la plus adapté.

Un travail pionnier [44], a abordé ces contraintes avec succès. Dans ce projet, les chercheurs des laboratoires de Hewlett Packard ont développé l'ordinateur TERAMAC, un multiprocesseur construit en interconnectant un grand nombre de FPGAs. L'équipe a utilisé intentionnellement des FPGAs défectueux ainsi que des technologies d'interconnections de qualité médiocre produisant une grande quantité d'interconnections défectueuses. Le multiprocesseur était créé avec l'aide d'un outil de synthèse pour FPGA, qui programme le réseau des FPGAs afin d'implémenter des architectures conventionnelles dans les structures régulières d'un tel réseau. Avant cette programmation, des procédures de test et de diagnostic ont été appliquées pour identifier les éléments défectueux du système. Puis, la programmation était faite en sorte que seulement les ressources exemptes des défauts étaient utilisées. Les résultats de ce projet illustrent qu'il est possible d'implémenter des systèmes complexes en programmant un réseau régulier, composé d'un grand nombre de modules simples et identiques et affecté par des grandes densités de défauts. Mais, certains autres aspects devraient être traités afin d'exploiter efficacement la complexité des composants nanotechnologiques pour construire des systèmes d'une capacité de calcul extraordinaire et à faible coût. Ces aspects sont discutés dans la section 1.2.

Grâce à cette flexibilité et à leur potentiel pour une intégration beaucoup plus poussée que les technologies de fabrication actuelles, les nanotechnologies devraient révolutionner les systèmes de calcul en intégrant dans un seul composant un nombre extrêmement grand de dispositifs et ceci avec des coûts de fabrications drastiquement plus faibles que les technologies actuelles. Ces tendances auront un impact profond sur les architectures des systèmes de calcul et exigeront un nouveau

paradigme de CAO (conception assistée par ordinateur). Dans la suite, nous proposons une plateforme de CAO, dont l'objectif est de permettre au concepteur de demain d'exploiter de façon efficace l'extraordinaire puissance de calcul prévue d'être apportée par ces technologies.

1.2 Une proposition de CAO pour les nanotechnologies

Le CAO du futur devra fournir les outils nécessaires pour transformer par la programmation la structure régulière des nanoréseaux à des systèmes de calcul puissants. L'exemple du projet de TERAMAC illustre la puissance de cette approche. L'approche adoptée dans ce projet consiste en la programmation des nanoréseaux pour réaliser des multiprocesseurs ayant une architecture conventionnelle. Les multiprocesseurs seront par la suite programmés par un logiciel conventionnel pour exécuter divers calculs. Le nombre énorme de dispositifs des nanoréseaux devra permettre d'implémenter des nombres sans précédent de processeurs dans des architectures massivement parallèles, résultant à une puissance de calcul sans précédent. Cependant, les ordinateurs parallèles qui sont basés sur des architectures conventionnelles font généralement une très faible utilisation de leur puissance de calcul théorique. En plus, les architectures standard font un gaspillage des ressources de matériel par rapport à une architecture dédié à une l'application cible.

La réflexion adoptée dans cette thèse vise le développement des outils de CAO permettant d'exploiter efficacement les composants nanotechnologiques qui pourraient être construits à l'aide des technologies émergentes. Cette approche a été motivée par les facteurs convergents suivants:

a- La disponibilité des plateformes matériels (composants nanotechnologiques ou nanoréseaux) comportant des quantités énormes des modules identiques, reliés dans un réseau régulier (la promesse des nanotechnologies).

b- La flexibilité de ces plateformes matérielles, qui permettent d'implémenter un système cible en les programmant.

c- La nature de la majorité des systèmes naturels complexes, qui sont composés d'un nombre énorme d'éléments « simples », du même type ou d'un petit nombre de types différents.

d- L'intérêt d'explorer la dynamique de tels systèmes naturels ou artificiels.

Comme exemples de tels systèmes nous pouvons citer des systèmes physiques (par exemple formation de système solaire, interactions galactiques, atmosphère et océans, réactions chimiques, réactions nucléaires, et d'autres systèmes complexes composés de quantités énormes de particules) et systèmes biologiques (par exemple interactions moléculaires dans la cellule, interactions cellulaires dans un organisme, différentiation des cellules dans l'embryon, systèmes neuronaux, interactions entre les organismes dans un écosystème, sociétés de fourmis....). Malgré leur grande diversité, nous pouvons trouver dans les fondements des différents systèmes naturels la caractéristique commune d'être des systèmes réguliers très complexes, composés d'un grand nombre de modules simples et identiques ou appartenant à un ensemble limité de types de modules différents. La compréhension de la dynamique de ce type de systèmes est un défi fondamental de la science moderne. L'expérimentation informatique des systèmes composés d'un grand nombre de modules élémentaires simples, est une manière commode pour étudier l'influence sur la dynamique globale de ce genre de systèmes, des règles qui régissent le comportement des modules élémentaires. Ces expérimentations devront viser aussi bien l'étude des systèmes naturels que l'étude des systèmes artificiels inspirés des systèmes naturels ou inventés par l'homme. L'intérêt de ces derniers est qu'ils permettent de modifier à volonté les règles qui régissent les modules élémentaires, soit pour les simplifier afin de rendre l'étude faisable, soit pour étudier l'influence de diverses modifications sur la dynamique globale du système.

Pour mieux souligner l'importance de telles études, notons qu'étant données les limitations des ressources naturelles de notre planète et la croissance très rapide de notre civilisation technologique, la compréhension de la dynamique des systèmes naturels et la prévision de leur évolution sont essentielles, non seulement pour soutenir le progrès de la civilisation humaine, mais aussi pour son simple maintient. Ainsi, la possibilité de simuler avec précision ces systèmes est une exigence fondamentale pour les systèmes de calcul du futur.

La disponibilité de plateformes matérielles (les nanoréseaux) offrant une puissance de calcul sans précédent (point a), devra permettre une augmentation drastique de la précision de nos simulations des systèmes naturels et artificiels, et une amélioration substantielle de leur compréhension. Cependant, l'utilisation de ces nanoréseaux pour implémenter des architectures parallèles des processeurs classiques, et puis, programmer ces architectures pour simuler les systèmes cible, se traduit par un grand gaspillage des ressources matérielles. Par contre, un gain énorme de puissance de calcul et d'efficacité énergétique sera obtenu si nous programmons les nanoréseaux pour implémenter directement l'application cible. Aujourd'hui, cette approche est le plus souvent économiquement inefficace, en raison du coût élevé de la fabrication des architectures dédiées pour chaque système cible. En outre, l'utilisation des FPGA n'améliore pas de manière significative l'efficacité du calcul par rapport aux ordinateurs standards, à cause du gaspillage des ressources matériels nécessaire pour implémenter des structures reconfigurables comme les FPGA. Cependant, puisque la seule manière d'exploiter la structure régulière des nanoréseaux [40] consiste à les programmer pour implémenter le système de calcul cible (point b), les inconvénients que nous venons de mentionner, concernant l'utilisation des architectures dédiées au lieu d'une architecture de système de calcul standard disparaissent. Ainsi, le gain en efficacité de calcul est obtenu sans aucun coût.

Par conséquent, au lieu des deux étapes utilisées dans le projet TERAMAC :

- **programmer le nanoréseau** pour implémenter une architecture de multiprocesseur standard, et en suite

- **programmer cette architecture** pour simuler le système régulier ciblé,

il semble préférable d'implémenter le système cible dans le nanoréseau dans une seule étape, sans utiliser la deuxième étape de programmation classique. L'implémentation consiste à déterminer l'ensemble des modules du nanoréseau nécessaires pour implémenter le module de base du système cible, et programmer ensuite le nanoréseau pour implémenter autant des modules du système cible que permis par la complexité du nanoréseau. Si nous disposons d'une plateforme dédiée de CAO, l'exécution de ces tâches peut être moins complexe que le développement d'un logiciel dédié pour programmer l'application dans un système de calcul standard. De plus, la nature régulière du nanoréseau et du système cible, permet une exploitation maximale des ressources matérielles, qui seront gaspillées sévèrement si nous adoptons l'approche d'une architecture de système de calcul standard.

Pour rendre gérable l'approche que nous venons de proposer, il est nécessaire de disposer d'une plateforme d'outils de CAO dédiés. Cette plateforme devrait inclure les parties suivantes :

- **un outil «logiciel de simulation»** permettant de décrire sans effort majeur des systèmes complexes composés d'un grand nombre de modules simples et les expérimenter par simulation dans des ordinateurs classiques, voir Figure 1-1.

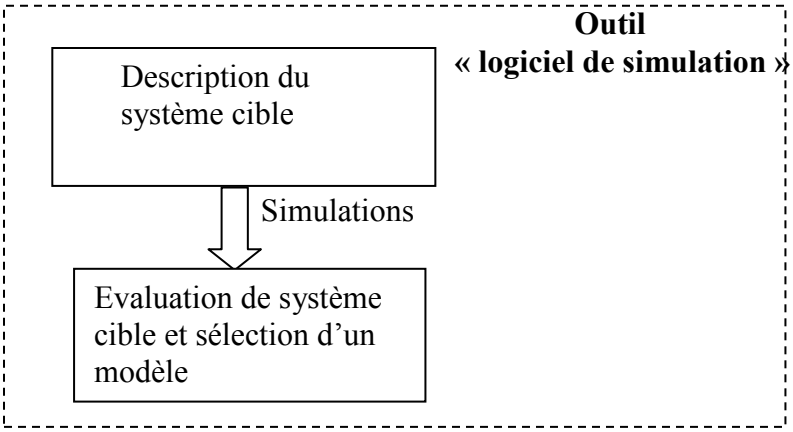


Figure 1-1. Outil de simulation

- **un outil d'exploration architecturale,** permettant l'évaluation de l'efficacité des architectures de nanoréseaux pour implémenter un système complexe cible, voir Figure 1-2.

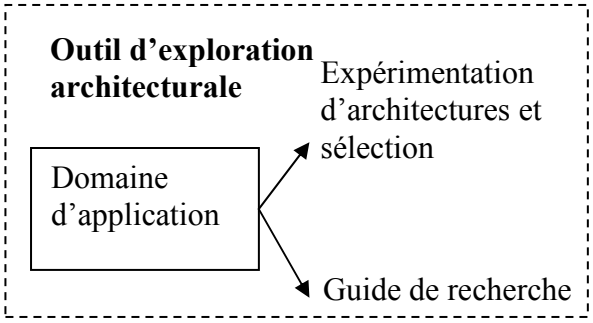


Figure 1-2. Outil d'exploration architecturale

- **un outil de synthèse** permettant d'implémenter le système complexe cible directement sur l'architecture du nanoréseau, voir Figure 1-3.

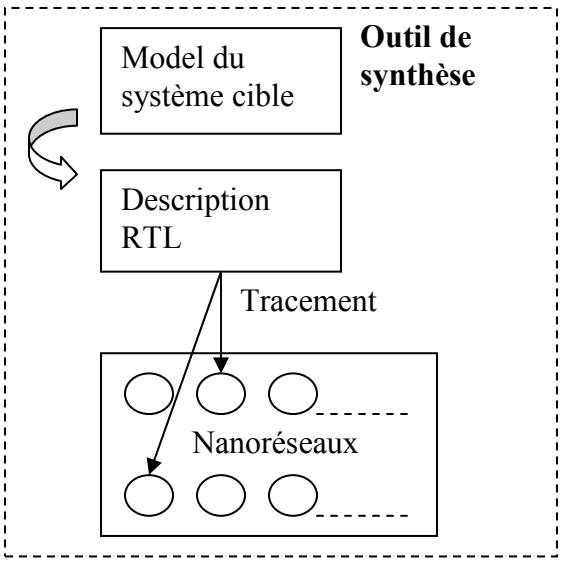


Figure 1-3. Outil de synthèse

Dans la suite nous donnons une description succincte de ces outils.

1.2.1 Outil de synthèse

L'outil de synthèse implémente le système cible choisi sur l'architecture spécifique du nanoréseau ciblé. Cet outil comporte une première partie qui crée une description RTL (niveau de transfert de registres) du module (ou de l'ensemble des modules) de base du système cible. Puisque la complexité des applications considérées émerge plutôt du nombre énorme des modules que de la complexité du ou des modules, dans beaucoup de cas, ces modules seront relativement simples. Dans ce cas, l'outil de synthèse crée une machine à états finis, ayant comme entrées et sorties les signaux de communication entrants et sortants du module et comporte un ensemble de registres d'état qui correspondent aux variables internes du module. L'état suivant de chacun des ces registres est produit par une fonction combinatoire qui code la règle d'évolution de la variable correspondante. Les entrées de cette fonction viennent des registres d'état de la cellule (y compris le registre associé à cette fonction) et des signaux de communication entrants. La mise en œuvre et l'optimisation de cette solution directe est bien entendu triviale, grâce

aux outils de synthèse des machines d'états finis, existantes dans tous les plateformes commerciales de synthèse des circuits intégrés. De plus, le calcul est très rapide puisque chaque état suivant de chaque module est produit dans un seul cycle d'horloge. Mais cette approche est réaliste uniquement si les règles qui régissent l'évolution du système cible sont relativement simples. Dans des applications qui utilisent des règles d'évolution complexes, il sera plus efficace d'implémenter dans chaque module un circuit séquentiel programmable (par exemple un petit processeur classique) pour effectuer ce calcul. Dans ce cas le même circuit pourra être utilisé pour calculer l'un après l'autre l'état de toutes les registres d'état du module. Néanmoins, pour être plus efficace, ce processeur devra être dédié au calcul des règles qui régissent l'état des modules de base du système cible. Dans ce cas un logiciel de synthèse de haut niveau, comme ceux décrits dans [45]-[52], doit être utilisé pour synthétiser ce circuit.

La deuxième partie de l'outil de synthèse est similaire aux outils de synthèse utilisés couramment pour implémenter une description RTL (register transfert level) dans un FPGA [53]-[57]. Une approche similaire peut être utilisée pour développer un outil qui implémente une description RTL sur une architecture de nanoréseau.

1.2.2 Outil de simulation

L'outil de simulation permet la description et l'exploration rapide des systèmes complexes composés d'un grand nombre de modules simples, en effectuant une simulation sur un système de calcul classique. Grâce à cet outil, l'utilisateur n'a pas besoin de développer un logiciel dédié pour chaque système cible. Il doit seulement spécifier, dans un menu, les caractéristiques générales du module : l'ensemble de variables internes du module, des variables de communication entrantes (signaux en provenance des modules voisines) et des variables de communication sortantes (signaux envoyés par le module aux modules voisines), et des règles d'évolution qui déterminent le prochain état de chaque variable en fonction de l'état actuel des

variables internes et des variables de communication entrantes. Cet outil permettra à l'utilisateur d'expérimenter rapidement et à petite échelle (petit nombre de modules) différents modèles d'un système naturel, ou d'expérimenter rapidement et à petite échelle différents modèles de systèmes complexes inventés, en vue d'une étude préliminaire de l'impact des diverses règles d'évolution et des divers choix des variables internes et de communication. Ces expérimentations seront nécessaires pour faire un premier choix du modèle qui sera employé pour les expérimentations à plus grande échelle, avant d'entamer la tâche plus fastidieuse d'implémentation du système cible dans le nanoréseau.

1.2.3 Outil d'exploration architecturale

L'outil d'exploration architecturale permet d'expérimenter diverses architectures de nanoréseau disponibles afin de déterminer les plus adéquates pour implémenter un système complexe cible ou une famille des systèmes complexes. En particulier ces architectures devraient résoudre le problème de communication, qui représente le défi le plus important des systèmes fortement parallèles.

1.3 Cas d'étude pour l'outil de simulation

Afin d'éviter de développer l'outil de simulation sur des considérations abstraites, qui pourraient conduire à un système peu réaliste, nous avons besoin d'une l'application qui servira comme cas d'étude. Cette application devrait inclure toutes les contraintes qui pourraient être rencontrées dans les différents genres de systèmes complexes composés d'un grand nombre de modules simples du même type ou appartenant à un ensemble limité de types de modules. Nous avons constaté que les systèmes inspirés des écosystèmes naturels, composés d'un environnement dans lequel évolue un certain nombre d'organismes, capables de se développer à partir d'une cellule pour donner un organisme adulte (ontogenèse), de se mouvoir,

se nourrir, se reproduire par croissement sexué, s'adapter à l'environnement par le biais de mutations et de sélection naturelle, regroupent tous les caractéristiques nécessaires pour servir de cas d'étude. Car comme nous verrons dans la section suivante, l'évolution spatiale des différents modules (par exemple leur mouvement dans un environnement), peut induire des contraintes très complexes à gérer, particulièrement dans le cas de systèmes qui ne se composent pas par des modules identiques. Des contraintes semblables mais moins rigoureuses sont induites quand la fonction d'un module n'est pas stable à travers le temps. Les systèmes inspirés des écosystèmes naturels mentionnés ci-dessus, ont ces caractéristiques puisque d'un part les organismes se déplacent et d'autre part leurs modules de base (cellules) ne sont pas stables à travers le temps (par exemple différentiation des cellules, mutation de cellules...). Ainsi, ils ont été choisis comme applicatif cas d'étude.

1.4 Solutions architecturales et l'outil d'exploration

Comme nous l'avons déjà mentionné, une fois que les fonctions des modules (entrées, sorties, variables internes et règles d'évolution) ont été déterminées, le système peut être programmé dans une plateforme matérielle de type FPGA. Cependant, l'exploitation efficace des ressources matérielles dépend de l'architecture du nanoréseau utilisé et de la manière dont le système cible est implémenté sur cette architecture. Pour des applications dans lesquelles les modules ne sont pas mobiles et leur fonction reste fixe, nous pouvons implémenter facilement la fonction des modules sur une plateforme reprogrammable de type FPGA, et ceci de façon similaire à l'implémentation des fonctions par un circuit intégré quelconque. Cependant, pour des applications qui utilisent des modules mobiles, et dont les fonctions peuvent varier dans le temps, la gestion des interactions entre les modules peut devenir très compliquée. Cette situation se rencontre dans les systèmes inspirés des écosystèmes naturels. Ainsi leur choix comme cas d'étude nous permettra de prendre en compte les contraintes trouvés

propres à d'autres systèmes cibles, y compris les plus exigeants. Pour supporter le mouvement des modules nous avons trois options :

a- La première option consiste à coder la position de chaque module dans l'environnement en employant une variable de position. Dans ce cas, le module se déplace dans l'environnement en changeant la valeur de sa variable de position, mais sa position dans le nanoréseau reste stable. Ainsi, les modules situés dans des endroits physiquement éloignés dans le nanoréseau peuvent être voisins dans l'environnement simulé et, par ce fait, doivent interagir. Pour assurer l'interaction entre tels modules, soit nous devons disposer d'un nanoréseau qui comporte des interconnections de complexité exponentielle afin de pouvoir relier à tout moment des modules se trouvant dans n'importe quelles parties du nanoréseau, soit nous devons dépenser un nombre de cycles proportionnels à la taille du réseau, afin de transférer l'information d'interaction entre deux modules très distantes dans le nanoréseau, en utilisant seulement des interconnections locales. Ces solutions affecteront sévèrement soit la quantité des ressources matérielles nécessaires pour implémenter une application donnée, soit la vitesse de l'exécution. Pour éviter ces problèmes, nous pouvons implémenter l'application en sorte que seulement des interactions locales sont employées. Deux approches de ce type sont décrites par la suite.

b- La deuxième option n'emploie pas une variable de position pour chaque module. Au lieu de cela, la position de chaque module dans l'environnement est identique à sa position physique dans le nanoréseau. Dans ce cas, quand un module se déplace dans l l'environnement il doit également se déplacer dans le nanoréseau. Pour des applications où tous les modules sont identiques, tous les parties du nanoréseau implémentent la même fonction (c'est-à-dire la fonction du module élémentaire). Ainsi, le mouvement du module exige de transférer seulement les valeurs des variables d'un module à une partie voisine inoccupé du nanoréseau, ou échanger les valeurs des variables de deux modules occupant des parties voisines dans le nanoréseau. Ceci peut être facilement effectué dans des architectures standards de type FPGA (des architectures de FPGA

reprogrammables ou non reprogrammables). Cependant, si l'application utilise des modules de plusieurs types, le déplacement d'un module exige le transfert à une partie voisine, non seulement des valeurs de ses variables mais également de sa fonction. Dans des architectures non reprogrammables, le passage de la fonction d'un module dans une partie voisin du réseau est impossible. Par contre, il est possible dans des architectures FPGA reprogrammables conventionnelles, mais leur re-programmation est une tâche difficile puisqu'elle ne peut pas être faite localement. En effet, l'information de la re-programmation doit être transférée à un système externe chargé de modifier la mémoire de configuration afin de modifier la fonction des parties concernées de FPGA. Effectuer cette tâche pour un grand nombre de modules mobiles affectera de façon drastique les performances du système. Ainsi, les architectures conventionnelles de type FPGA ne sont pas appropriées dans ce contexte. Pour pouvoir implémenter cette approche de façon plus efficace, l'architecture du nanoréseau devra fournir des connections pour charger d'une façon directe les valeurs contenues dans les cellules de mémoire de configuration d'un partie du réseau, aux cellules de mémoire de configuration d'une partie voisine. Cette solution architecturale permet également l'utilisation efficace du matériel dans les applications où la fonction d'un module peut changer à travers le temps. Par exemple, dans le cas de l'ontogenèse, le phénomène de différentiation peut modifier la fonction des cellules. Une autre situation où la fonction de cellules change à travers le temps correspond aux mutations des cellules. Ainsi, la programmation locale peut également être employée pour traiter ce cas aussi.

c- Dans les applications où les différents modules occupent une petite partie de l'espace dans lequel ils évoluent, l'utilisation de la position physique d'un module dans le nanoréseau comme sa position dans l'espace simulé (solution suggéré dans le point b), gaspillera une grande quantité de ressources matérielles. Ceci se produira, lorsque toutes les positions inoccupées de l'espace simulé correspondront à des parties obligatoirement inutilisées du nanoréseau. Pour éviter cette situation, nous proposons une solution qui fusionne les solutions des points a et b. Avec cette solution, la position de chaque module est déterminée par une variable interne. Afin

de se déplacer, le module augmente ou diminue le composant x et/ou y de cette variable. Cependant, afin de résoudre le problème grave de communication entre les modules (discuté dans le point a), nous imposons quelques contraintes qui maintiennent une corrélation spatiale entre les positions des modules dans l'espace simulé et leurs positions dans le nanoréseau. Considérons un module qui occupe la position (a, b) dans le nanoréseau. Par la suite, ce module sera mentionné comme module (a, b). Ses modules voisins les plus proches dans le nanoréseau occupent les positions (a-1, b-1), (a-1, b), (a-1, b+1), (a, b-1), (a, b+1), (a+1, b-1), (a+1, b), (a+1, b+1). Nous imposons les contraintes spatiales suivantes : Le composant x de la variable de position du module (a, b) ne prend jamais une valeur inférieure ou l'égale au composant x des variables de position des modules (a-1, b-1), (a-1, b), (a-1, b+1). Le composant x de la variable de position du module (a, b) ne prend jamais une valeur supérieure ou l'égale au composant x des variables de position des modules (a+1, b-1), (a+1, b), (a+1, b+1). Le composant y de la variable de position du module (a, b) ne prend jamais une valeur inférieure ou l'égale au composant y des variables de position des modules (a-1, b-1), (a, b-1), (a+1, b-1). Le composant y de la variable de position du module (a, b) ne prend jamais une valeur supérieure ou l'égale au composant y des variables de position des modules (a-1, b+1), (a, b+1), (a+1, b+1). Avec ces contraintes, les modules entourant le module (a, b) dans l'espace simulé sont les même que dans le nanoréseau (c'est-à-dire (a-1, b-1), (a-1, b), (a-1, b+1), (a, b-1), (a, b+1), (a+1, b-1), (a+1, b), (a+1, b+1)). Par conséquent, ce sont les seuls modules avec lesquels (a, b) peut interagir. En effet, tout module (a, b) pourra interagir seulement avec ses voisins immédiats dans le nanoréseau, et pour cette raison, n'utilisera que des communications locales. Ainsi, tout en maintenant des communications locales comme dans le point b, cette solution n'exige pas l'utilisation d'un nombre de modules égaux au nombre des positions de l'espace simulé. Elle est donc appropriée pour les applications dans lesquelles les modules occupent une petite partie de l'espace simulé. Par exemple, dans un écosystème artificiel, les organismes vivants peuvent laisser inoccupées de grandes parties de l'espace. Avec la solution proposée ici, un organisme vivant peut s'éloigner d'un autre organisme juste en en modifiant les variables de position des

cellules qui le composent, c'est-à-dire, sans avoir besoin d'occuper une partie différente du nanoréseau pour chaque nouvelle position, comme c'était le cas dans l'approche décrite en point a. Ainsi, nous ne sommes pas obligés d'associer une partie du nanoréseau à chaque position inoccupée de l'espace simulé, évitant le gaspillage des ressources matérielles.

Néanmoins, pour tirer profit de cet avantage, nous devons maintenir les contraintes spatiales qui ont été décrites ci-dessus. Afin de maintenir ces contraintes, chaque module reçoit de ses huit modules voisins dans le nanoréseau la valeur de leur variable de position. Quand les nouvelles valeurs des variables de position de deux modules sont en contradiction avec les contraintes spatiales, l'emplacement des modules dans le nanoréseau est modifié pour satisfaire ces contraintes. Par exemple, ceci se produira, si deux organismes vivants changent leur position relative et quelques cellules d'un organisme ayant une valeur précédemment supérieure en y ou de x de quelques cellules de l'autre organisme, aura une valeur inférieure après ce changement. Quand cette situation se produit, elle est détectée facilement par les modules (cellules), et les modules se repositionnent dans des nouvelles parties du nanoréseau afin de satisfaire les contraintes spatiales. Ce changement de positions sera effectué localement, en employant une architecture de nanoréseaux permettant la re-programmation locale des fonctions des modules du nanoréseau comme décrit dans le point b. Cette architecture doit permettre de transférer la fonction d'un module à un module voisin inutilisé, ou d'échanger les fonctions de deux modules voisins. Dans certaines situations, ce processus peut impliquer que tous les modules d'un certain secteur du nanoréseau (par exemple toutes les cellules d'un organisme), et peut prendre de nombreuses cycles pour être effectué, puisque le repositionnement de quelques cellules peut impliquer celui de nouvelles cellules, et ainsi de suite. Mais à l'exception des situations très spéciales, le repositionnement n'impliquera pas de grands secteurs du nanoréseau, puisqu'il n'est pas propagé dans l'espace vide entourant un organisme ou un groupe d'organismes regroupés dans une partie de l'espace. Ceci étant dit, quand une partie du nanoréseau est reprogrammée, il ne peut pas effectuer des calcules et

envoyer des signaux de communication valides vers ses modules voisins. Pour traiter ce problème nous pouvons utiliser l'une des solutions suivantes :

i- Le module concerné active un signal pour informer ses voisins de son indisponibilité. En réponse, les modules voisins n'exécutent pas de nouveaux calculs, jusqu'à ce que le signal soit désactivé. De plus, au prochain cycle ils activent un signal semblable pour informer leurs voisins, qui interrompent le calcul à leur tour, et ainsi de suite. Une vague des interruptions est ainsi propagée dans le nanoréseau, mais elle est arrêtée rapidement, quand l'espace libre entourant les organismes est atteinte. Ainsi, ces interruptions demeurent locales et n'affectent pas l'opération du reste du nanoréseau. Cette solution est efficace mais peut ne pas paraître très rigoureuse, car, en principe, l'ensemble du système doit évoluer en même temps. Ainsi, il n'est pas rigoureux de faire évoluer les modules du système pendant que l'évolution de certains d'entre eux est interrompue. Mais, nous noterons que cette interruption n'aura pas des conséquences graves sur un organisme, par exemple un herbivore, qui reste figé pendant que ses prédateurs évoluent, tant que ces prédateurs se trouvent à une distance telle qu'il n'y a pas d'interactions possibles. Ainsi, cette solution n'aura pas des conséquences importantes sur le comportement global du système étudié, et nous pouvons accepter cet manque de rigueur vue l'amélioration très importante des performances du système.

ii- Comme dans le cas précédent, la vague d'interruptions s'arrête quand elle atteint l'espace libre. Mais dans le cas présent nous prenons soin de façon à ce que les modules soient toujours dans le même cycle d'opération, chaque fois qu'elles se trouvent en situation d'interagir. Pour ce faire, chaque module comporte une variable dite de cycle, qui indique le nombre des cycles actifs du module. A chaque cycle d'activité du module, cette variable est incrémentée. Quand l'évolution d'un module est interrompue pour effectuer un repositionnement dans le nanoréseau, cette variable reste inchangée. Quand des modules voisins dans le nanoréseau se trouvent voisins dans l'espace simulé et doivent interagir, ils comparent leurs variables de cycle respectives. Si dans un instant la variable de cycle d'un module est inférieure à la variable de cycle d'un autre module voisin dans le nanoréseau, et

si la distance qui sépare ces modules dans l'espace simulé est égale à de la distance maximale qu'ils peuvent parcourir dans un nombre de cycles égal à la différence de leurs variables de cycle, l'activité du module ayant la plus grande variable de cycle est interrompue jusqu'à ce que la condition décrite ci-dessus n'est plus vraie. Le test de cette condition est effectué facilement car il nécessite des échanges d'informations (variables de cycle et de position) uniquement entre modules voisins dans le nanoréseau.

Suite à la discussion précédente, nous pouvons déterminer les architectures les mieux adaptées aux différentes types de systèmes cibles:

1. Pour des applications comportant des modules privés de mouvement, des architectures conventionnelles de type FPGA (reprogrammables ou non) seront utilisées.

2. Pour des applications utilisant un seul type de modules et munis mouvement, des architectures reprogrammables conventionnelles de type de FPGA seront employées. La position de chaque module sera déterminée par une variable de position.

3. Pour des applications qui utilisent des plusieurs types de modules munis de mouvement, nous utiliserons des architectures permettant le transfert local de la fonction d'une partie du nanoréseau à une partie voisine. Deux approches différentes seront employées pour déterminer la position de chaque module dans l'espace simulé :

3a. Si le nombre des positions de l'espace simulé est similaire au nombre des modules du système cible qui peuvent être programmés dans le nanoréseau l'espace du nanoréseau, la position d'un module sera identique à sa position physique dans le nanoréseau.

3b. Si l'espace simulé est beaucoup plus grand que l'espace du nanoréseau, la position d'un module sera déterminée par une variable de position. En plus, des

contraintes spatiales seront imposées aux variables de position des modules, comme décrit dans le point c.

Ces architectures traitent le problème de communication considéré comme étant le plus grand défi dans les architectures parallèles. L'adoption d'une solution efficace pour ce problème est essentielle, puisqu'il est aggravé par le parallélisme total de l'approche considérée ici. Cependant, les critères pour déterminer la meilleure architecture pour une application donnée peuvent ne pas être clairs (considérez par exemple les critères pour choisir l'approche 3a ou 3b). Un outil dédié à l'exploration architecturale devient alors essentiel afin de permettre l'évaluation de ce type de choix. Cet outil aura quelques similarités avec l'outil de simulation (décrit dans la section 1.3), mais il doit être capable d'incorporer à chaque module simulé les contraintes de l'architecture matérielle choisie. Par exemple, au cas 3b, il devra introduire les contraintes spatiales qui sont imposées à la variable de position de chaque module. Cet outil sera utilisé pour déterminer la performance de chaque solution (coût de matériel/vitesse) et pour déterminer l'influence de chaque architecture sur le comportement du système simulé, car, comme nous avons vu précédemment (point i), certains choix peuvent avoir un important impact sur le comportement du système.

1.5 Conclusions

Dans ce chapitre nous avons décrit une plateforme de CAO dont l'objectif est de permettre une utilisation efficace de l'énorme puissance de calcul promise par les nanotechnologies. Les systèmes de calcul fabriqués dans ces technologies auront une capacité de calcul qui devra permettre de simuler avec une grande précision l'évolution des systèmes complexes composés d'un nombre énorme d'éléments simples. Mais un nouveau paradigme de CAO sera nécessaire pour exploiter efficacement ces ressources. Nous avons décrit les différentes parties d'une telle plateforme, qui devrait inclure :

- **un outil de simulation** permettant de décrire sans effort majeur des systèmes complexes composés par grand nombre de modules simples, et étudier leur comportement par dans des systèmes de calcul traditionnels.

- **un outil de synthèse** permettant d'implémenter le système complexe cible sur l'architecture d'un nanoréseau.

- **un outil d'exploration architecturale** permettant d'évaluer l'efficacité d'un ensemble d'architectures pour implémenter un système complexe cible ou une famille de tels systèmes.

Nous avons aussi discuté différents choix d'architectures pour les nanoréseaux, en relation avec les contraintes imposés par la simulation des systèmes complexes composés par un grand nombre d'éléments simples. Les contraintes les plus sévères sont celles concernant la communication entre les modules élémentaires composant ces systèmes. Nous avons proposé certaines solutions qui devront apporter une réponse satisfaisante à ce problème en particulier pour certaines des applications visée dans ce travail. Néanmoins, la réalisation de l'ensemble des outils décrits dans ce chapitre nécessite une quantité de travail qui dépasse le cadre de cette thèse. Ainsi, dans les chapitres suivants, nous concentrons nos efforts sur la réalisation d'un de ces outils, à savoir, l'outil de simulation, ainsi qu'à son utilisation pour étudier certains de nos systèmes cibles.

Chapitre 2

2 Présentation de l'outil de simulation logicielle

Dans ce chapitre nous présentons l'outil de simulation logicielle, qui permet de simuler dans des ordinateurs standard, des systèmes complexes composés d'un grand nombre de modules identiques. Nous présentons, d'abord, les types d'applications cible de cet outil, ensuite nous décrivons son utilisation et finalement nous présentons sa structure.

2.1 Types d'applications

L'outil permet d'implémenter des systèmes complexes composés d'un grand nombre d'entités simples, identiques ou appartenant à un groupe limité d'entités simples, qui interagissent afin de produire le comportement complexe du système global. Un grand nombre des systèmes naturels font partie de cette catégorie, puisque les systèmes naturels sont souvent composés d'un nombre énorme d'entités simples. Parmi eux, nous pouvons citer des systèmes naturels (formation des systèmes solaires, atmosphère, océans, réactions chimiques, interactions des particules élémentaires…) et des systèmes biologiques (interactions moléculaires à l'intérieur des cellules, interactions des cellules dans un organisme, systèmes neuronaux, sociétés des fourmis …).

L'émergence du comportement riche et complexe de ces systèmes est une conséquence du grand nombre d'entités et de la grande complexité de leurs interactions plutôt que de la complexité des entités élémentaires qui les façonnent. Tous ces systèmes ont en commun le fait qu'ils sont des systèmes régulières et complexes, composés d'un grand nombre de modules simples. Ainsi, malgré leur grande diversité, cette ressemblance nous permet de créer une plateforme logiciel unique permettant à l'utilisateur de spécifier aisément toutes sortes de systèmes de ce genre et de les simuler pour étudier leur comportement.

2.1.1 Systèmes cibles

Les systèmes que nous ciblons sont des systèmes composés d'un grand nombre d'entités élémentaires. L'évolution globale du système est produite par des pas élémentaires pendant lesquels toute entité élémentaire calcule son état suivant selon certaines fonctions (ou lois) à partir de son état propre et de l'état des entités avec lesquelles elle interagit (Figure 2-1). Ainsi, l'analyse de ces systèmes équivaut à l'analyse de deux aspects : le type d'entités qui composent le système cible (leur variables ainsi que leur fonctions d'évolution), et le type d'interactions entre entités. Dans la suite, nous allons présenter ces deux aspects de manière analytique.

2.1.1.1 Entités élémentaires

D'une façon générale, une entité élémentaire peut être décrite par son comportement, c'est-à-dire de la façon dont elle évolue dans un environnement donné. D'une façon plus concrète, une entité élémentaire peut être décrite par un ensemble des variables d'état et un ensemble de fonctions. A chaque variable d'état est associée une ou plusieurs fonctions (ou lois) qui gouvernent son évolution en fonction de l'état de l'entité (la valeur des variables d'état) et de l'état des entités avec lesquelles elle interagit (son environnement), voir Figure 2-1.

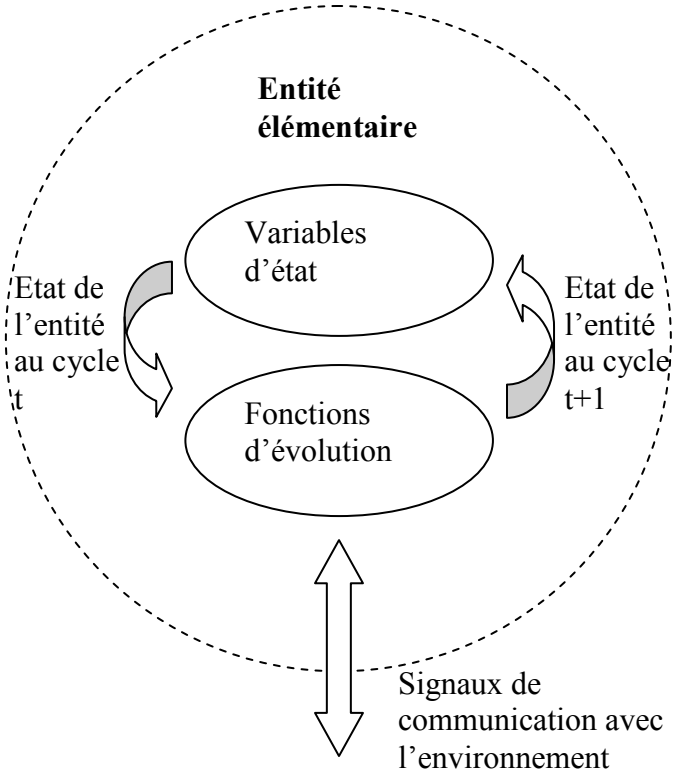


Figure 2-1 : Description d'une entité élémentaire

A titre d'exemple, dans une application étudiant des organismes artificiels constitués d'un ensemble de cellules, nous décrirons une cellule par

- un ensemble des variables d'état (par exemple : des variables qui codent l'information génétique de l'organisme, des variables qui guident la différentiation des cellules pendant l'ontogenèse et la création d'organes et des variables qui déterminent l'état physiologique des cellules - leur énergie, leur température…)
- un ensemble des fonctions associées qui agissent sur ces variables (par exemple : fonctions de différentiation de la cellule, fonctions de métabolisme, de diffusion de la température…).

Dans des applications du domaine de la physique, les variables d'état des particules élémentaires seront les variables de position, de vitesse, … alors que les fonctions associées seront celles correspondant aux lois des interactions entre les particules.

2.1.1.2 Conditions d'interactions et position des entités élémentaires

Les fonctions d'interaction qui déterminent le changement de l'état d'une entité élémentaire lorsqu'elle interagit avec d'autres entités ne suffisent pas pour déterminer l'évolution de cet état. Il faut aussi déterminer quelles sont les entités avec lesquelles elle va interagir dans une situation donnée. Il est possible d'introduire dans les fonctions d'interaction les conditions qui déterminent ces entités. Néanmoins, pour plus de flexibilité, nous pouvons introduire ces conditions de façon indépendante. Dans certaines applications, ces conditions seront fonction de la distance entre entités. Dans ce cas, il est possible de considérer plusieurs cas possibles :

- **Des interactions locales** qui apparaissent entre les entités élémentaires voisines. Le voisinage est défini par la distance entre les entités. L'espace peut avoir une, deux ou plusieurs dimensions. Nous pouvons considérer un espace de type matriciel, où les entités élémentaires occupent des positions discrètes, ou un espace continu.

Dans un espace matricielle à k dimensions, le pas de la matrice sera égal à une unité de longueur et la variable de position de chaque entité élémentaire sera un k-tuple constitué de k nombres entiers. Dans ce cas, le voisinage d'une entité élémentaire comportera 3^k-1 positions voisines.

Dans le cas d'un espace matriciel à 2 dimensions, la variable de position d'une entité élémentaire est une paire (x, y) constituée de deux nombres entiers x et y, et son voisinage comportera 8 positions : (x-1,y), (x+1,y), (x,y-1), (x,y+1), (x-1,y-1), (x+1,y-1), (x-1,y+1), (x+1,y+1).

Dans un espace continu à k dimensions, la variable de position d'une entité élémentaire sera un k-tuple dont les éléments sont des nombres réels. Dans ce cas, une entité élémentaire sera dans le voisinage d'interaction d'une autre entité élémentaire si leur distance est inférieure d'une certaine valeur. Par exemple, si nous considérons une distance Euclidienne, le voisinage d'interaction d'une particule élémentaire de position (x, y, z) dans un espace continu à trois dimensions sera la sphère définie par la relation $\sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2} \leq l$, où l est une constante définie par l'utilisateur). Toute particule dont la variable de position (x_2, y_2, z_2) vérifie cette relation va interagir avec la première particule.

- **Des interactions globales** dans lesquelles la distance n'est pas prise en compte dans la condition d'interaction. Cependant, la distance pourra intervenir dans les fonctions qui déterminent la façon dont changent les variables d'état des entités en interaction. Ainsi, dans des applications venant de la physique, deux particules pourront interagir à n'importe quelle distance, mais l'intensité de leur interaction dépendra de la distance qui les sépare.

Notons que dans des applications venant de la physique, nous pouvons considérer que les interactions sont locales ou globales, selon le modèle considéré. Ainsi, classiquement, deux particules chargées électriquement interagiront à n'importe quelle distance. Mais dans un modèle selon lequel cette interaction se manifeste par échange de photons, nous traiterons l'interaction selon le modèle locale.

2.2 Utilisation de l'outil

L'outil permet à l'utilisateur de créer des nouvelles applications avec un effort minimal, en décrivant le nombre et le type d'entités élémentaires qui composent le système cible, ainsi que les lois d'évolution de leur état et le type interactions avec les autres cellules. Une fois l'application décrite, l'utilisateur peut procéder à son

initialisation, à la description des visualisations et à l'exécution des simulations de son évolution.

2.2.1 Description du système cible

Comme nous l'avons cité ci-dessus, une entité élémentaire comporte des variables d'état et des lois qui calculent la valeur suivante de ces variables, en fonction de l'état présent de l'entité en question et de l'état d'autres entités élémentaires du système.

La description du système cible consiste en la description des entités qui le façonnent. Selon l'application, les entités élémentaires du système cible peuvent être identiques ou elles peuvent appartenir à un petit groupe d'entités de types différents. Par exemple dans une société de fourmis toutes les entités du système cibles appartiennent au même type (type « fourmis »), tandis que dans un autre exemple des réactions chimiques il y a plusieurs types d'entités élémentaires (correspondants aux différents atomes et molécules qui font partie de la réaction). Pour décrire son système cible, l'utilisateur doit décrire les différentiels types d'entités élémentaires (variables d'état et fonctions associées), déterminer le nombre total d'entités, et attribuer à chacune des entités un de ces types. Il doit aussi leurs attribuer un état initial.

Pour décrire les types d'entités qui composent le système (leurs variables et les fonctions qui gouvernent leur évolution), l'utilisateur doit exécuter trois tâches différentes situées dans le sous-menu « define elements » de l'outil :

- **Créer une nouvelle entité.** Pour chaque type d'entité élémentaire du système cible nous pouvons définir son nom et lui attribuer des caractéristiques, par exemple la couleur qui lui sera associé. Dans la Figure 2-2 nous présentons une copie d'écran de la fenêtre de création d'une nouvelle entité (cellule de type isolation thermique ayant une couleur gris).

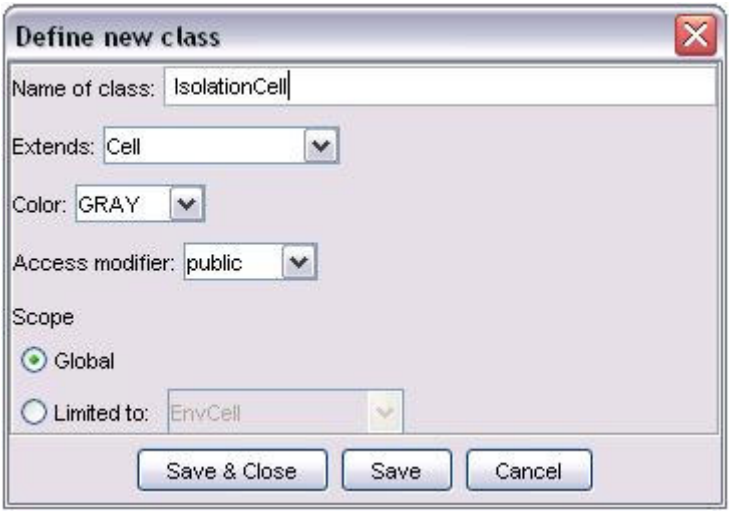


Figure 2-2 : Définition d'une nouvelle entité.

Créer une nouvelle variable. Pour chaque variable différente d'une entité donnée nous pouvons définir son nom, son type et le type d'entités élémentaires auxquelles elle appartienne. Dans la Figure 2-3 nous présentons une copie d'écran de la fenêtre de création d'une nouvelle variable (variable de coefficient d'isolation thermique d'une cellule de type isolation thermique. Sa valeur initiale par défaut est 0.7).

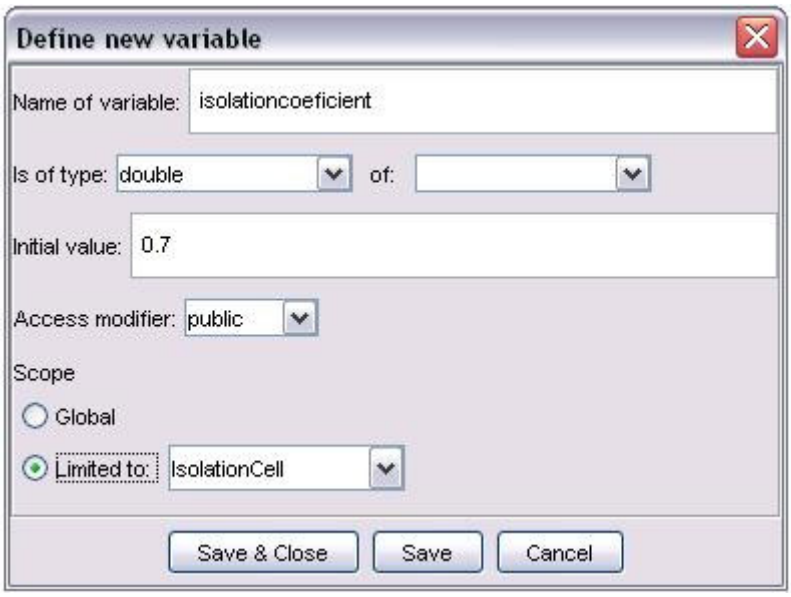


Figure 2-3 : Définition d'une nouvelle variable.

- **Créer une nouvelle fonction.** Pour chaque fonction différente nous définissons son nom, la variable sur laquelle elle agit et nous décrivons la fonction qu'elle réalise. Dans la Figure 2-4 nous présentons une copie d'écran de la fenêtre de création d'une nouvelle fonction. La fonction qui est décrite dans cet exemple calcule la diffusion de la température dans un onanisme. La partie droite de la fenêtre correspond au code de la fonction. Ainsi, afin de simplifier la tâche de l'utilisateur l'outil offre plusieurs possibilités :

- d'utiliser des fonctions qui sont déjà programmées dans l'application courante. L'utilisateur peut naviguer dans l'application afin de trouver des variables et/ou des fonctions qui appartiennent à des entités de l'application déjà spécifiées et les utiliser pour spécifier la fonction courante (partie gauche de la fenêtre).

- d'utiliser des fonctions qui se trouvent dans la bibliothèque d'applications de l'outil.

La description d'une fonction se fait par l'utilisateur en langue Java.

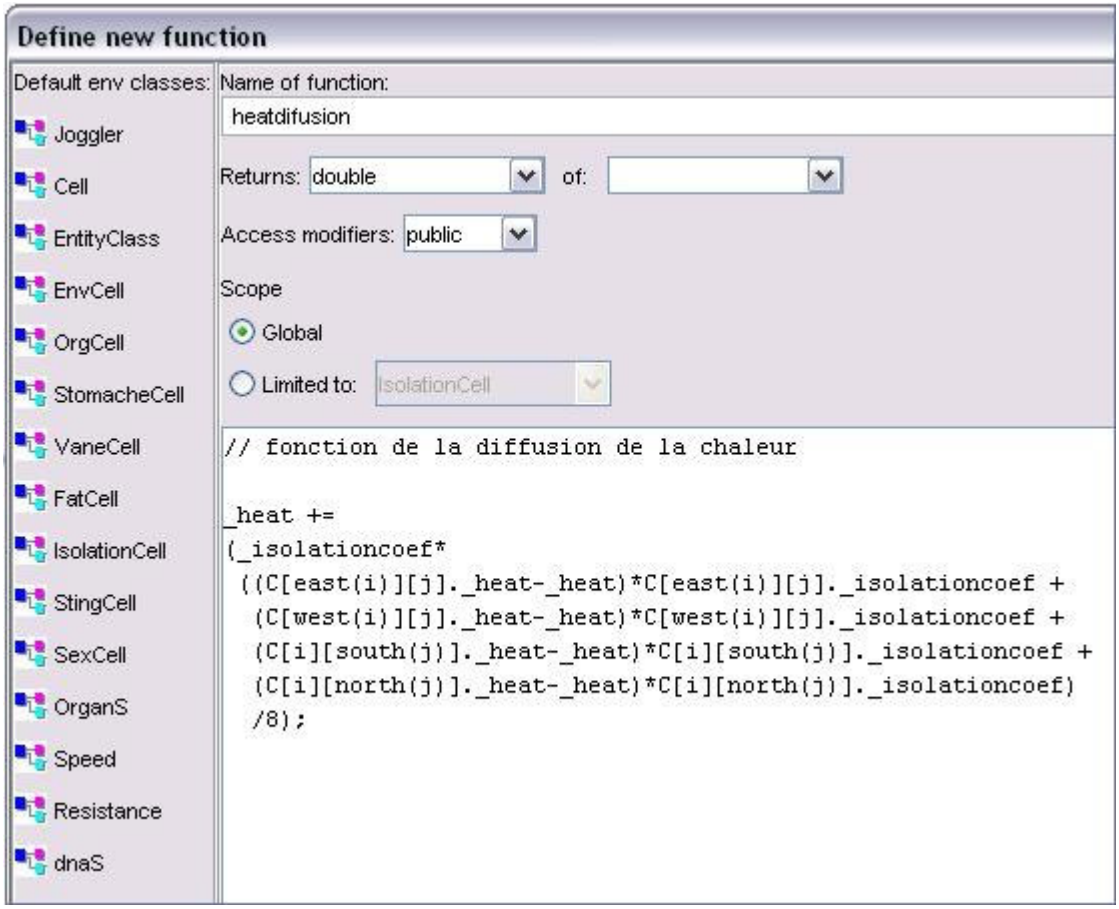


Figure 2-4 : Définition d'une nouvelle fonction.

2.2.2 Initialisation du système

Après avoir décrit le système cible, l'utilisateur peut procéder à son initialisation. L'initialisation du système consiste à l'initialisation des variables d'état des entités élémentaires, y compris l'initialisation de la variable de position (positionnement dans l'espace de l'application). En sélectionnant le bouton « positioning » dans le sous-menu « initialisation » une fenêtre appropriée sera ouverte selon le type d'application – positionnement discret, continu ou sans positionnement.

2.2.2.1 Positionnement discret des entités élémentaires

Afin d'initialiser une application avec positionnement discret des entités élémentaires (espace matricielle) l'utilisateur doit :

- choisir les dimensions de l'espace de l'application (par exemple 200x200 pour l'application montrée dans la Figure 2-5).
- sélectionner un type d'entité parmi les types d'entités qui se trouvent dans la partie droite et en haut de la fenêtre d'initialisation dans la Figure 2-5. Dans la partie droite et en bas de la fenêtre apparaissent les variables de ce type d'entités et leurs valeurs initiales.
- initialiser les variables de l'entité sélectionnée.
- positionner à l'aide de la souris, l'entité dans l'espace de l'application. Ce positionnement visuel est plus facile que l'initialisation des variables de position, qui reste toutefois possible si l'utilisateur le désire, ou dans le cas d'un positionnement en utilisant une approche globale, comme par exemple la génération aléatoire des variables de position, ou suivant une fonction de distribution analytique. Quand l'utilisateur emploie la souris pour positionner une entité, l'outil initialise la variable de position de cette entité à la valeur correspondante.
- répéter les trois étapes ci-dessus afin d'initialiser toutes les entités de l'application.

L'utilisateur peut aussi déplacer une entité dans l'espace et/ou déplacer un groupe d'entités en les sélectionnant à l'aide de la souri. L'utilisateur peut s'il le désire initialiser tout le fond de l'application (toute la matrice) avec une seule entité.

La Figure 2-5 représente une copie d'écran qui démontre l'initialisation d'une application « biologique » (organismes artificiels). Dans la partie droite en haute de la fenêtre se trouvent les différents types d'entités élémentaires (par exemple entité d'environnement, d'estomac…), en milieu de la fenêtre quelques attributs généraux du système et en bas de la fenêtre apparaissent les variables de l'entité sélectionnée. Dans cette figure, les variables qui apparaissent correspondent à l'entité d'environnement. La valeur par défaut de ces variables peut être modifiée par l'utilisateur avant le positionnement de l'entité dans l'espace de l'application. Dans la partie gauche de la fenêtre se trouve la matrice à deux dimensions qui représente l'espace de l'application.

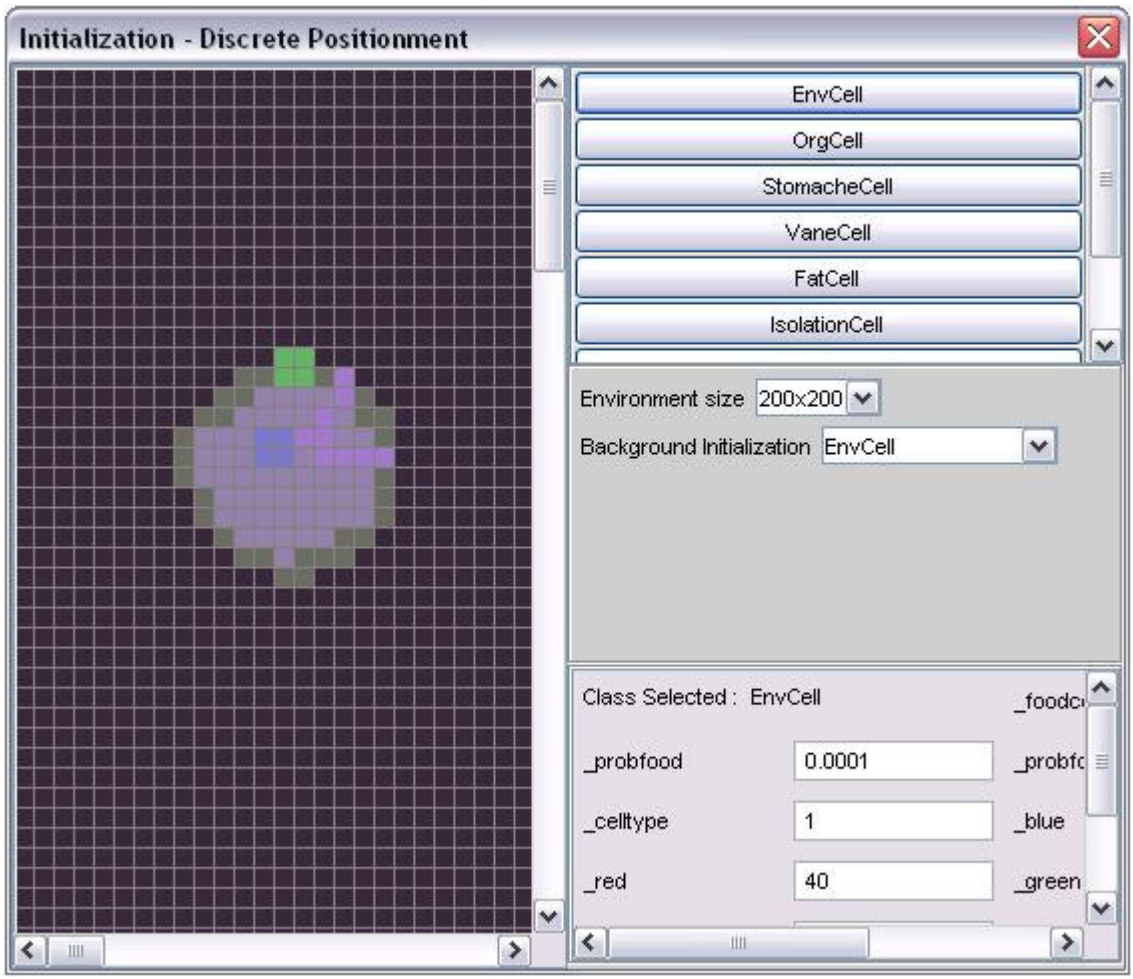


Figure 2-5 : initialisation d'une application avec positionnement discret des entités élémentaires.

2.2.2.2 Positionnement continu des entités élémentaires

L'initialisation d'une application à espace continu est effectuée de manière semblable avec l'initialisation d'une application à espace discret à la différence près que la partie gauche de la fenêtre d'initialisation représente un espace continu.

2.2.2.3 Initialisation des applications sans positionnement des entités élémentaires

Afin d'initialiser une application sans positionnement des entités élémentaires l'utilisateur doit :

- définir le nombre d'entités de l'application
- sélectionner un type d'entité
- initialiser les variables du type d'entité sélectionnée
- répéter les deux étapes ci-dessus afin d'initialiser toutes les entités de l'application

2.2.3 Description des visualisations

Après avoir initialisé le système cible, l'utilisateur peut définir la façon dont le système sera visualisé pendant les simulations. En lui donnant une visualisation appropriée, le système devient intelligible. En observant cette visualisation pendant les simulations, l'utilisateur pourra extraire des informations pertinentes et tirera des conclusions. Par défaut, pendant la création d'une entité élémentaire, l'outil lui associe trois variables de visualisation qui correspondent aux couleurs rouge, bleue et jaune. La combinaison de ces trois variables produit la couleur de l'entité (Figure 2-6).

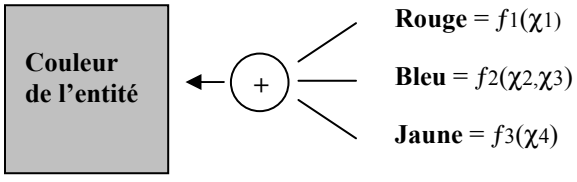


Figure 2-6 : Description de la visualisation d'une entité

Ces variables comme toutes les variables des entités du système, sont initialisées et des fonctions d'évolution leur sont associées. Ainsi, ces trois variables représentent l'état (ou une partie de l'état) de l'entité.

Par exemple, dans une application d'écosystème artificiel, l'observation de l'évolution de la couleur des cellules nous permet de suivre les différentes étapes de la différentiation des cellules pendant l'ontogenèse et savoir à la fin à quel organe appartient chaque cellule. Pour mieux suivre l'évolution des cellules nous avons utilisé les trois couleurs pour représenter différentes propriétés. Ainsi, la variation de la température d'une cellule est représentée par une variation du constituant rouge de la couleur par une fonction : $f1(t)$. Les autres constituants sont liés aux autres variables physiologiques de la cellule. L'utilisateur peut exécuter plusieurs simulations de la même application avec différentes visualisations. Ainsi, l'utilisateur peut observer l'évolution du système cible de plusieurs point de vues (c'est-à-dire en visualisant divers variables physiologiques) et extraire toutes informations possibles.

L'utilisateur peut aussi visualiser l'évolution de certaines variables avec l'aide des représentations graphiques. Pour ce faire, il doit définir les variables qu'il veut observer et la fréquence de ses observations. La figure 2-7 démontre des représentations graphiques qui font preuve de l'adaptation génétique d'une espèce d'organismes (phylogenèse) aux changements environnementaux. Dans cette application, une population d'organismes artificiels qui se déplacent dans un environnement, se nourrissent et se multiplient, sont soumis au processus de la sélection naturelle. Par la suite, des changements environnementaux sont produits à la simulation (partie basse de la figure). L'espèce adapte par sélection naturelle le gène qui définie la façon dont les organismes se déplacent (partie haute de la figure) en fonction de la difficulté au déplacement dans le milieu et la distribution de la nourriture. La population moyenne est aussi affectée par ces changements (milieu de la figure).

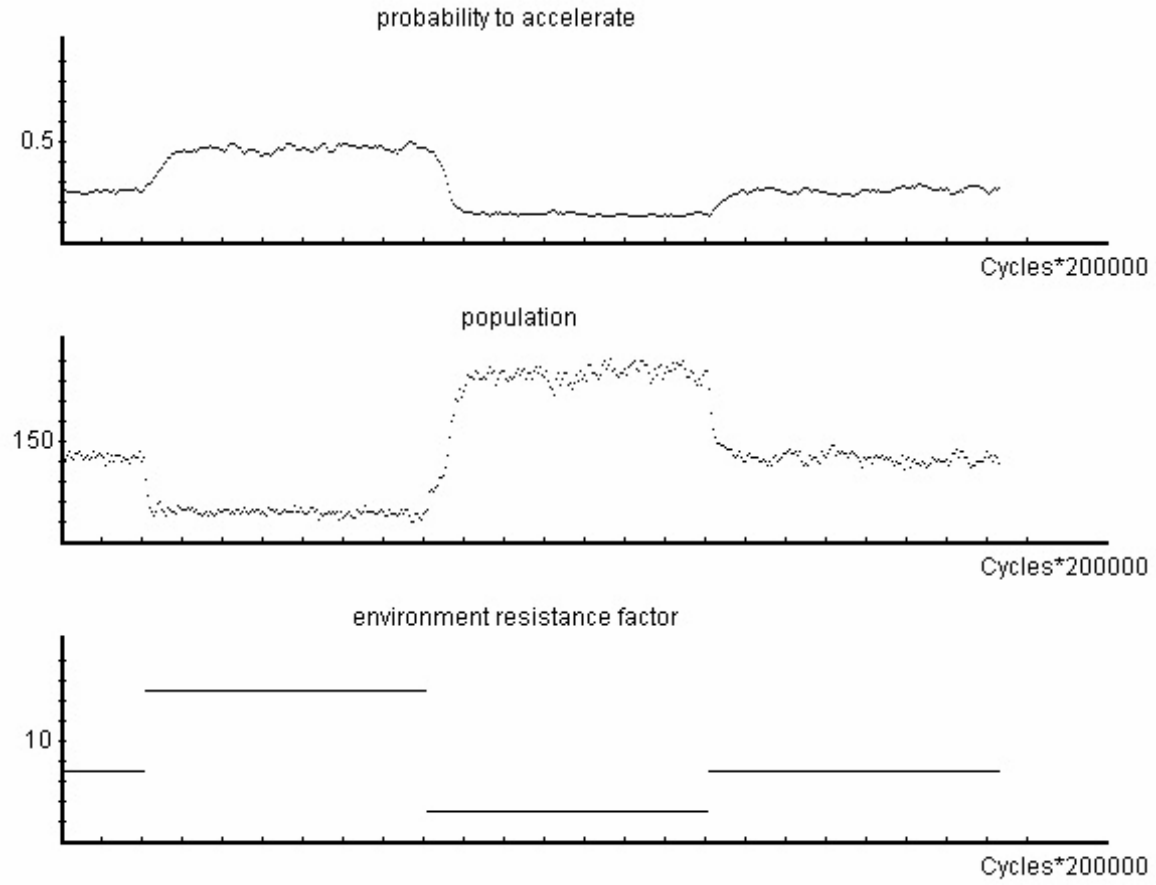


Figure 2-7 : représentation graphique.

2.2.4 Exécution des simulations

Après la description du système cible, son initialisation et la description des visualisations, l'utilisateur peut procéder à l'exécution des simulations. Afin d'exécuter des simulations l'utilisateur :

- génère le fichier Java de l'application.
- exécute la simulation
- gère l'exécution.

Dans la Figure 2-8, nous présentons une copie d'écran de la fenêtre d'exécution des simulations. L'utilisateur peur démarrer et arrêter l'exécution, modifier la vitesse d'exécution, focaliser dans certains endroits de l'espace de l'application ou visualiser l'évolution des représentations graphiques. Il peut aussi modifier certains paramètres du système cible, sans devoir initialiser chaque fois l'application. Ainsi, l'utilisateur peut rapidement effectuer plusieurs simulations et étudier l'influence de ces paramètres sur le comportement du système.

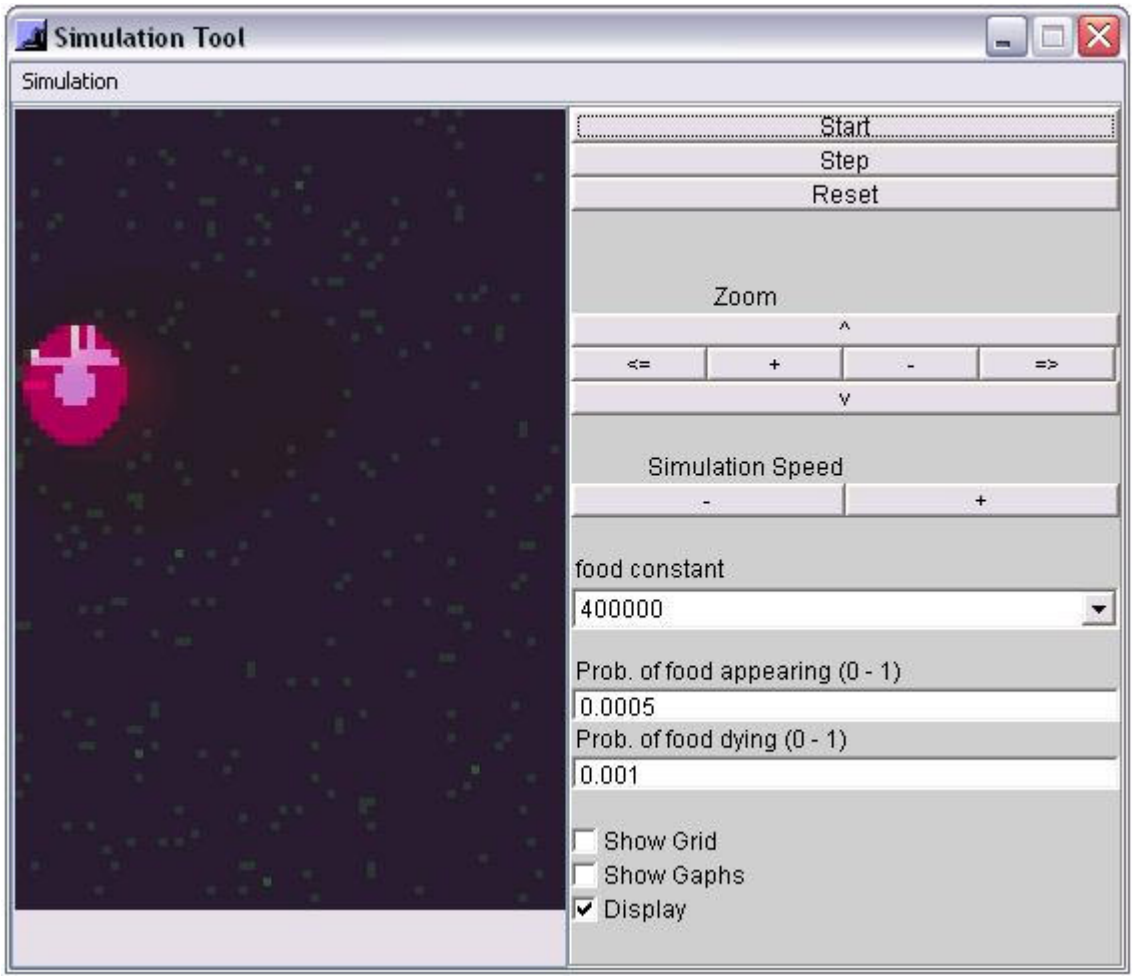


Figure 2-8 : exécution des simulations

2.3 Structure de l'outil

L'outil est structuré de la façon suivante (Figure 2-9) :

1 programme central de l'outil - fenêtres d'interface graphique

2 bibliothèque d'applications

3 fichier exécutable

4 fichiers Java d'applications

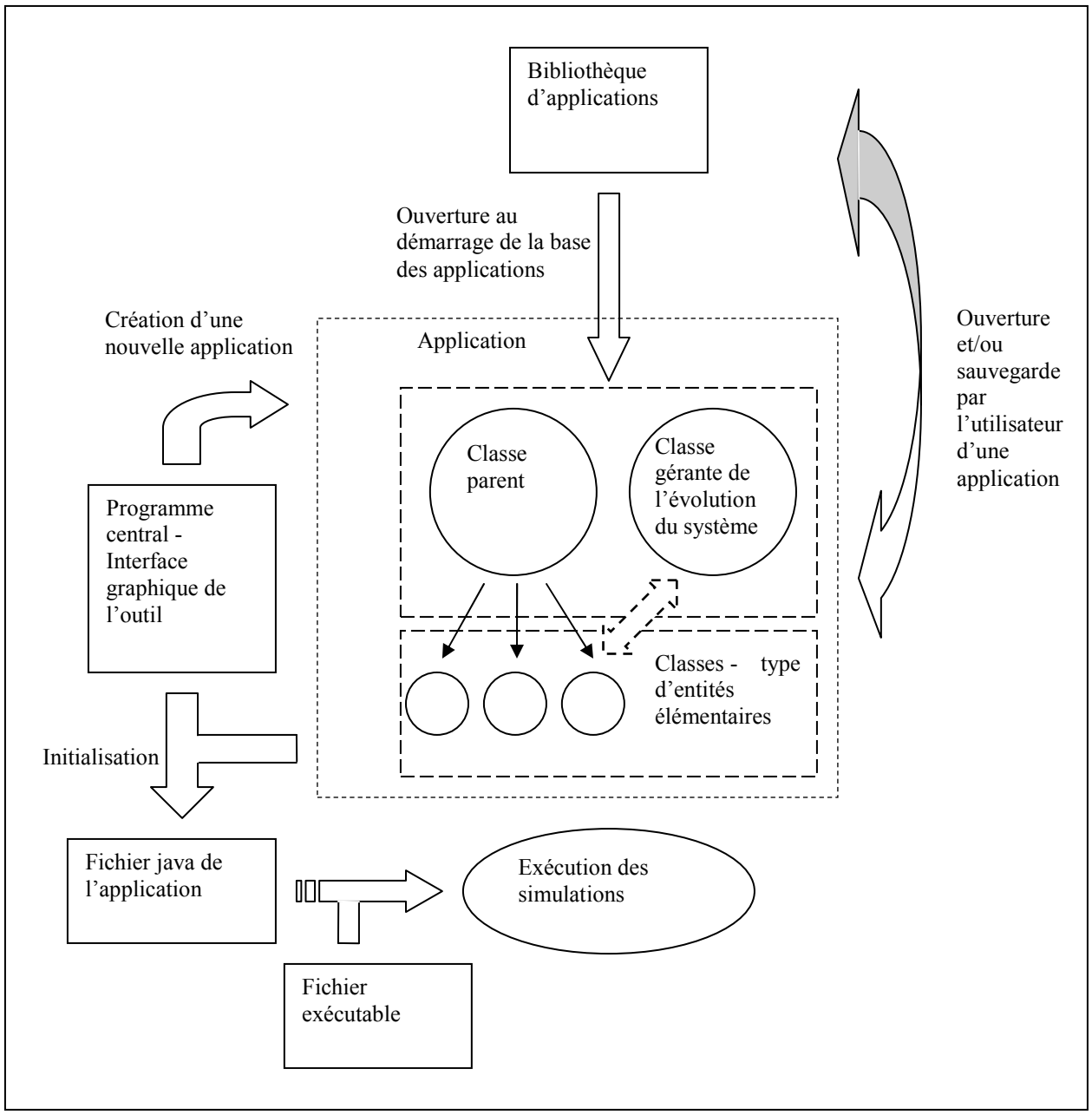


Figure 2-9 : Structure de l'outil

2.3.1 Programme central de l'outil

Le programme central de l'outil est un fichier qui est exécuté à chaque démarrage de l'outil. Ce programme :

- **crée l'interface graphique pendant le démarrage de l'outil** (Figure 2-10)**.** Cette interface contient le menu central de l'outil à partir duquel l'utilisateur peut déclencher des actions telles que : ouvrir et sauvegarder des applications, décrire des nouvelles applications, exécuter des simulations…

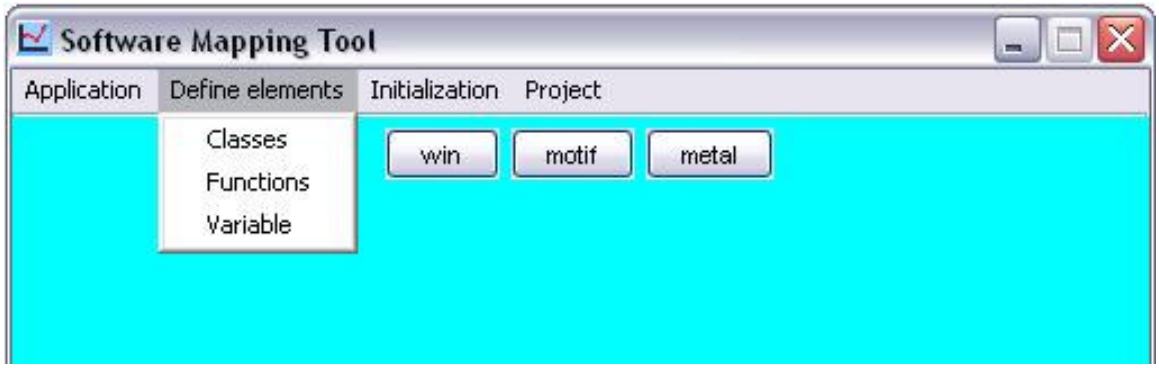


Figure 2-10 : L'interface graphique

- **ouvre un fichier pendant le démarrage de l'outil et à partir de la bibliothèque.** Ce fichier crée la structure de base des fichiers Java des applications. Cette structure de base (sa fonction sera présentée au chapitre suivant) en combinaison avec la description du système cible est utilisée pour la création du fichier Java qui correspond à l'application.
- **gère les actions déclenchées par l'utilisation des menus de l'interface graphique, c'est à dire :**

- génère les fenêtres appropriées (fenêtre de création des nouveaux types d'entités, des nouvelles fonctions et variables, d'initialisation…)

- ouvre une application à partir de la bibliothèque des applications

- sauvegarde les applications créées par l'utilisateur dans la bibliothèque d'applications

- génère le fichier Java qui correspond à l'application

- compile ce fichier avant l'exécution des simulations en faisant appel à un compilateur Java

- exécute le fichier exécutable qui génère la fenêtre appropriée afin d'effectuer des simulations

2.3.2 Fichier Java de l'application

Java est un langage oriente objet. Il utilise des classes qui sont considerées comme des abstractions d'une entité, en définissant ses caracteristiques génériques. Une classe contient des attributs (éléments de donnés) et des méthodes (fonctions ou procedures qui donnent accés ou modifient les atributs de la classe). Les objets sont des instances des classes.

Les classes que nous utilisons dans le fichier Java d'application appartiennent à trois catégories : classe gérante, classe parent et classe de type d'entité élémentaire (figure 2-11). Nous allons présenter ces classes dans les paragraphes suivantes.

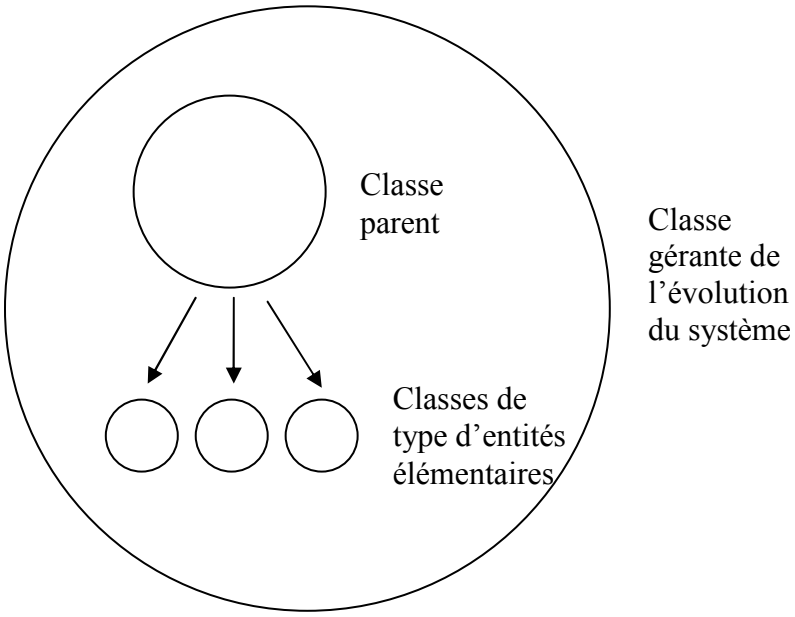


Figure 2-11 : Les différentes classes du fichier d'une application

- **La classe gérante**. L'ensemble des états des entités élémentaires du système compose l'état du système. A chaque cycle l'état du système est stocké dans deux variables d'état; l'une stockant son état présent et l'autre son état passé. La fonction de la classe gérante consiste à gérer l'évolution du système cible étape par étape en produisant les nouveaux cycles (générations) du système et ses visualisations. En particulier, dans cette classe sont sauvegardées des informations générales de l'application (le nombre d'entités, leur type, et les informations qui définissent le positionnement et l'initialisation de ces entités). La classe gérante, utilise ces informations et la description des entités élémentaires (qui se trouve dans les classes de type d'entité élémentaire) pour déterminer l'évolution du système. Plus précisément, à chaque nouveau cycle du système, la classe gérante utilise l'état présent du cycle précédent qui se trouve dans l'une des variables d'état, ensuite applique les fonctions d'évolution des entités élémentaires et à la fin elle produit l'état présent du nouveau cycle. Cet état est stocké dans la deuxième variable d'état, tandis que l'état stocké dans la première variable d'état devient l'état passé. Ainsi, en répétant ces étapes et basculant d'une variable d'état à l'autre, la

classe gérante engendre l'évolution du système. Dans la figure 2-12, nous présentons la fonctionnalité de la classe gérante. Dans cette figure nous observons les deux variables d'état du système; l'une stockant son état présent et l'autre son état passé. A chaque cycle l'état présent bascule d'une variable d'état à l'autre.

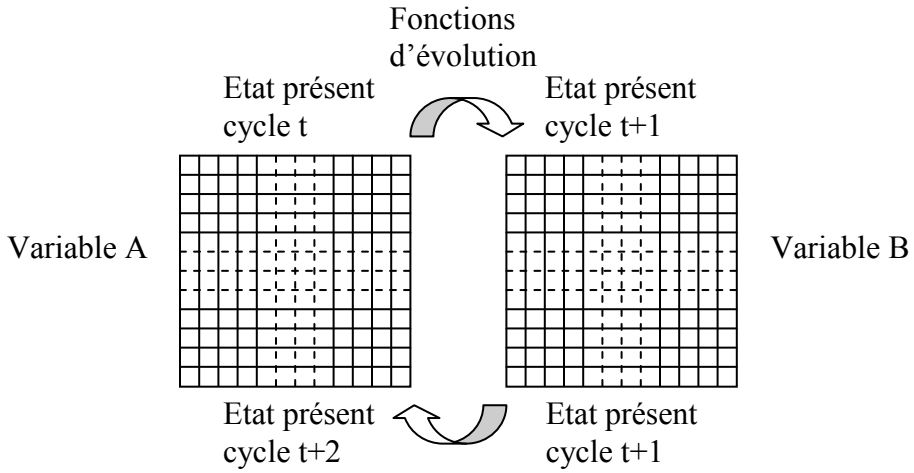


Figure 2-12 : Fonctionnalité de la classe gérante

- **La classe parente** contient des variables et des fonctions qui sont communes à toutes les types d'entités du système cible (par exemple les variables de la couleur rouge, bleu et jaune des entités élémentaires qui sont utilisé pour produire leurs visualisation pendant les simulations) et sa fonction consiste à hériter ces attributs à toutes les classes – type d'entités élémentaires.
- **classes de type d'entités élémentaires** qui correspondent aux différents types d'entités élémentaires et contiennent les variables de ces types d'entités et leurs fonctions associées.

Le fichier Java d'une application est un fichier qui contient la description de l'application. Il sera utilisé pour l'exécution des simulations. En termes plus techniques, c'est un fichier de type « .Java » contenant toutes les classes de

l'application. Ce fichier sera compilé avant l'exécution pour produire les fichiers de type « .classe » correspondant à chaque classe des fichiers et qui seront utilisés pendant l'exécution des simulations.

Le ficher Java d'une application est créé en sélectionnant le bouton « generate application » situé dans le sous-menu « project » du menu central de l'outil (Figure 2-13).



Figure 2-13 : création du fichier Java d'une application

Le programme central de l'outil utilise un fichier de donnés du système qui est sauvegardé à la bibliothèque d'applications. Ce fichier contient la description générale de la classe gérante et de la classe parent ainsi que des informations générales sur l'application afin de créer la classe gérante et la classe parent. Ces informations (nombre et type d'entités élémentaires, leurs positions, leurs état initial,…) sont données par l'utilisateur pendant l'initialisation du système cible. D'autre part, le programme central utilise la description des types d'entités élémentaires pour produire les classes – type d'entités élémentaires du même fichier.

2.3.3 Fichier exécutable de la simulation

Ce fichier est exécuté par le programme central de l'outil afin de réaliser une simulation. La fonction de ce programme consiste à :

- créer la fenêtre de la simulation qui offre à l'utilisateur la possibilité de gérer l'exécution de la simulation, comme nous avons vu dans la Figure 2-8.
- créer une instance de la classe gérante de l'application.
- produire des générations de l'application et ses visualisations avec l'aide de cette classe.

2.3.4 Bibliothèque d'applications

Quand l'utilisateur crée une application qu'il considère intéressante, il peut la sauvegarder dans la bibliothèque des applications, afin de pouvoir la réutiliser à l'avenir.

Le type de l'espace d'une application (continu ou discret) est figé une fois pour tout. A part cette restriction, une application est décrite dans la bibliothèque des applications de façon flexible afin de pouvoir la réutiliser et l'adapter. Ainsi, les types des entités élémentaires d'une application sont paramétrables. Par exemple, le nombre des variables d'état d'un type d'entités peut être modifié, les fonctions associées à ces variables peuvent être modifiés et leurs constantes sont paramétrables. L'utilisateur peut sélectionner un sous-ensemble des types d'entités proposés par l'application. Il peut définir le nombre de chaque type d'entités, les initialiser et déterminer les paramètres de visualisation. Néanmoins, l'utilisateur peut, s'il le souhaite, sauvegarder dans un fichier la valeur de certains de ces paramètres.

En relation avec une application, l'utilisateur peut ajouter une variable à la classe parent qui sera ainsi héritée par tous les types d'entités élémentaires, au lieu de la rajouter à chaque type. Il peut aussi associer à une variable de la classe parent une

fonction qui deviendra ainsi commune à tous les types d'entités. Par exemple, dans une application traitant un écosystème artificiel, l'utilisateur peut rajouter à la classe parent une variable qui représente la température des cellules et lui associer une fonction qui représente la diffusion de la température.

Chaque application est sauvegardée dans de un fichier de type «.DATA» (fichiers de donnés) dont la structure est donnée ci-dessous.

Type de l'espace

1^{ier} type d'entité
Nom du type d'entité
Nombre des variables hérités de la classe parent
1ère variable *: nom : type : valeur défaut : accès*
2ème variable *: nom : type : valeur défaut : accès*
…
Nombre des variables propres à ce type d'entité
1ère variable *: nom : type : valeur défaut : accès*
2ème variable *: nom : type : valeur défaut : accès*
…
Nombre des fonctions
1ère fonction *= nom (paramètres de la fonction) : type de retour : accès*
2ème fonction *= nom (paramètres de la fonction) : type de retour : accès*
…

2ème type d'entité
…

Modifications de la classe parent
Nombre des variables rajoutées
1ère variable rajoutée*: nom : type : valeur défaut : accès*
2ème variable rajoutée*: nom : type : valeur défaut : accès*
…

Nombre des fonctions rajoutées

1ère fonction rajoutée *= nom (paramètres de la fonction) : type de retour : accès*

2ème fonction rajoutée *= nom (paramètres de la fonction) : type de retour : accès*

…

2.4 Conclusions

Dans ce chapitre nous avons présenté l'outil de simulation logicielle, qui permet de simuler dans des ordinateurs standard, des systèmes complexes composés d'un grand nombre de modules identiques. Nous avons présenté, les types d'applications cible de cet outil, son utilisation et sa structure. Dans la suite, nous allons présenter deux exemples d'applications qui ont été implémentés par cet outil. Le premier exemple consiste des systèmes biologiquement inspirés composés d'un environnement dans lequel évolue un certain nombre d'organismes artificiels, qui ont les possibilités de mouvement, de réplication, d'évolution d'une cellule à un organisme (ontogenèse), de mutation, et d'adaptation à l'environnement. Cette application a servi comme cas d'étude, afin d'éviter de développer l'outil de simulation logicielle sur des considérations abstraites. Le second exemple d'application concerne le domaine de la physique fondamentale. Cette application est très différente des écosystèmes et illustre ainsi l'universalité de notre outil, qui permet à l'utilisateur de réaliser avec un moindre effort et d'explorer par simulation divers types de systèmes complexes constitués d'un ensemble d'entités simples.

Chapitre 3

3 Ecosystèmes artificiels

Dans ce chapitre nous présentons le premier exemple d'application qui concerne le domaine de la biologie. Cet exemple a servi comme cas d'étude pour le développement de l'outil. La structure riche et complexe des écosystèmes naturels est une source d'inspiration pour la création de systèmes artificiels. Dans un premier temps, nous décrivons le système cible. Ensuite, nous présentons quelques expérimentations.

Dans le domaine de la vie artificielle, des systèmes complexes composés d'un grand nombre d'entités simples, qui interagissent afin de produire le comportement complexe du système global, ont souvent été l'objet d'études. En particulier, en expérimentant avec les lois qui gouvernent l'évolution des entités élémentaires, les chercheurs ont pu reproduire des comportements complexes qui sont similaires aux comportements des êtres vivants. Souvent ces systèmes sont des systèmes simplistes. Un exemple classique est le jeu de la vie de Conway. Il s'agit d'un automate cellulaire, dans lequel émergent diverses formes qui peuvent se déplacer, s'autocopier, ou avoir d'autres comportements intéressants. Dans cet automate cellulaire, chaque cellule n'a que deux états : d'être vivante ou morte. Elle change son état en fonction de son état propre et du nombre des cellules vivantes de son voisinage. La simplicité impressionnante des lois qui gouvernent cet automate, ainsi que la complexité des structures émergeantes, démontre l'intérêt de cette approche.

Parmi les expérimentations les plus récentes, nous pouvons citer les automates cellulaires modélisant des processus chimiques caractéristiques des êtres vivants (auto catalyse, auto assemblage, organisation spontanée des structures complexes et dynamiques [58], [59]). Dans d'autres expérimentations, des automates cellulaires sont implémentés pour lesquelles des vers (« worms ») autocopiants augmentent leur complexité par le processus de l'évolution génétique [60]. En outre, des réseaux parallèles des FPGAs reconfigurables ont démontré la possibilité d'auto organisation des blocs de logique booléenne et l'émergence des structures autocopiantes [61].

Notre application est capable de produire des systèmes plus sophistiqués et éprouve une grande flexibilité pour spécifier et simuler aisément diverses variantes

d'écosystèmes artificiels. Par exemple, à l'aide de notre application, nous pouvons étudier le processus de l'ontogenèse, implémenter des organismes capables de se déplacer, se nourrir, se reproduire par croisement sexué, ou se diriger par fonction « olfactive » vers les sources de nourriture, et étudier leur évolution par sélection « naturelle » vers des formes adaptés aux conditions environnementales (phylogenèse).

En utilisant certains des principes fondamentaux des écosystèmes naturels et en les simplifiant, nous aboutissons à des systèmes artificiels, certes incomparablement plus simples que leurs contreparties naturelles, mais qui exhibent certains des caractéristiques et propriétés de ces contreparties. Grâce à leur structure riche, ces systèmes artificiels sont particulièrement souhaitables pour guider le développement d'un outil qui permettra à son utilisateur de décrire facilement et ensuite expérimenter par simulation un vaste ensemble de systèmes artificiels composés par un grand nombre d'éléments simples.

Ainsi, notre cas d'étude concerne des systèmes simplifiés, inspirés des écosystèmes naturels. Ces systèmes seront constitués d'un espace ou environnement virtuel, dans lequel évoluent des organismes multicellulaires artificiels ayant des propriétés inspirées du monde réel, telles que la réplication et la différentiation des cellules qui aboutissent à la création d'un organisme adulte à partir d'une cellule unique (ontogenèse), ou bien le mouvement, la capture de la nourriture, la reproduction sexuelle, ou l'adaptation à l'environnement grâce au processus de la mutation génétique et le « crossing-over », processus qui peut aboutir à la création de nouvelles espèces par phylogenèse. Puisque les organismes naturels sont composés des entités élémentaires (les cellules) qui interagissent afin de produire le comportement complexe des organes et de l'organisme entier, ces systèmes font très naturellement partie des applications visées par notre outil.

3.1 Variables

De la même façon que dans le cas des organismes naturels, les cellules d'un organisme virtuel partagent toutes la même information génétique (l'ADN) et sont soumises aux mêmes fonctions élémentaires décrites par la biologie moléculaire. Mais malgré cette similarité, comme dans les organismes vivants, la fonction des cellules diffère selon leur différentiation et leur état physiologique. En utilisant un langage plus technique la cellule artificielle est définie par ses variables d'état et par des fonctions qui déterminent l'évolution de ces variables.

Dans le cas des cellules artificielles, nous allons utiliser les variables d'état suivantes :

a) Les variables de l'ADN. La valeur de ces variables est fixe pour toutes les cellules d'un organisme pendant toute la durée de sa vie et peut varier au moment de la création d'une cellule embryonnaire pendant la reproduction :

- grâce au processus de « crossing-over » afin de permettre la variabilité des organismes d'une espèce et

- grâce à la mutation génétique afin de permettre l'évolution des espèces.

De la même manière que dans le cas de l'ADN des organismes naturels, l'information génétique est groupée dans des compartiments d'information connus sous le nom de gènes. L'information qui est codée dans les gènes détermine le type et la forme des différents organes de l'organisme ainsi que leurs fonctions.

b) Les variables de différentiation. Ces variables changent leur état pendant le processus de l'ontogenèse, mais leur valeur reste fixe pendant la durée de la vie adulte d'un organisme. Elles guident la différentiation d'une cellule et définissent sa spécialisation. Les différents types d'organes peuvent assurer diverses fonctions telles que le mouvement, la capture de la nourriture, la digestion, la reproduction …

c) Les variables d'état physiologique d'une cellule (variables physiologiques)**.** Ces variables changent leur état pendant toute la vie d'un organisme et indiquent l'état physiologique de la cellule, par exemple sa température, ses protéines, son énergie, …

3.2 Fonctions

De façon similaire à toutes les applications adressées par notre outil, à chaque entité élémentaire est associée une fonction qui gouverne son évolution. Ces fonctions calculent l'état suivant des entités du système selon leur état propre et l'état des entités avec lesquelles cette entité interagit. Dans les exemples que nous avons traités avec l'outil, les fonctions associées aux variables des cellules de l'organisme sont les suivantes :

a) Fonctions associées aux variables de l'ADN de l'organisme. Ces fonctions agissent pendant la reproduction. Elles utilisent des générateurs pseudo aléatoires qui déterminent quels gènes de chaque organisme parent seront transmis à l'organisme enfant. Ce phénomène est connu sous le nom « crossing-over ». D'autres générateurs pseudo aléatoires déterminent avec une probabilité choisie par l'utilisateur, l'occurrence d'une mutation génétique dans l'ADN de l'organisme enfant, le site ou le gène affecté par la mutation et la nouvelle valeur de ce site.

b) Fonctions associées aux variables de différentiation. Ces fonctions agissent pendant l'ontogenèse en utilisant l'information génétique et l'état physiologique des cellules (variables d'ADN et variables physiologiques). Les variables d'ADN contiennent la description de la manière selon laquelle évoluent les variables de différentiation, en fonction de l'état des variables physiologiques de la cellule (protéines, température…). Cette évolution engendre la différentiation des cellules. Ainsi, quand les variables physiologiques de la cellule réalisent une condition codée dans les variables d'ADN, les variables de différentiation changent leurs valeurs en enchaînant ainsi la différentiation de la cellule. Cette condition peut

être vue comme un simple seuil de la valeur d'une variable physiologique ou une condition plus complexe impliquant plusieurs variables physiologiques et activant des modifications sur plusieurs variables de différentiation.

c) Fonctions associées aux variables physiologiques. Ces fonctions agissent pendant toute la vie de l'organisme. Parmi les exemples des fonctions liées aux variables physiologiques nous pouvons citer les fonctions de diffusion de la chaleur, les fonctions de diffusion des protéines, les fonctions du métabolisme,…

Les organismes artificiels se déplacent dans un environnement virtuel afin de se nourrir et de se reproduire. L'environnement est aussi constitué des entités élémentaires que nous appellerons des entités élémentaires d'environnement. Une entité élémentaire d'environnement reste relativement simple par rapport aux cellules des organismes et comportent peu des variables (par exemple, variable de température, variable de nourriture, variable de résistance au mouvement si nous considérons un milieu dans lequel cette résistance varie...). La variable qui représente la température évolue suivant une fonction de diffusion. D'autre part, dans les exemples que nous avons expérimentés, la variable qui indique la présence de nourriture dans chaque entité d'environnement évolue de manière pseudo aléatoire avec une probabilité pondérée par la quantité de nourriture présente dans l'entité d'environnement et dans ces voisins.

Comme nous avons précisé ci-dessus, le choix des écosystèmes artificiels comme cas d'étude pour le développement de notre outil est motivé par la richesse de ce paradigme, et le fait qu'il regroupe des cas suffisamment variés et suffisamment complexes pour garantir qu'un outil développé pour pouvoir supporter ces systèmes pourra aussi supporter une vaste variété de systèmes réguliers constitués d'un grand nombre d'entités simples. En particulier, les entités de base (cellules des organismes artificiels) ont un comportement riche produit par une grande variété de variables et de fonctions. Leur état évolue en interagissant avec différents types d'entités telles les cellules environnantes et les cellules des organes des organismes. De plus, elles se déplacent dans le milieu et peuvent se transformer par

le processus de la différentiation et se multiplier ou se transformer elles mêmes en des entités différents (« crossing-over », mutations). La capacité de l'outil à traiter d'autres types de systèmes réguliers a été par la suite validée par une deuxième application venant de la physique, un domaine assez différent du domaine biologique.

Néanmoins, dans le contexte des écosystèmes artificiels nous avons choisi de considérer un espace discret de type matrice à deux dimensions, dans lequel évoluent les entités élémentaires du système, notamment les cellules des organismes et les entités d'environnement, en occupant des positions spécifiques dans la matrice déterminées par deux variables discrètes (x, y). Etant conscients que ce choix constituera une limitation si elle s'imposait comme choix unique, nous avons pris soin de donner à l'outil la capacité de traiter aussi le cas des espaces continus et des dimensions supérieures à deux.

3.3 Possibilités de l'application d'écosystèmes pour créer une variété de systèmes cibles

L'application d'écosystèmes est décrite de manière générique et donne à l'utilisateur la possibilité, de créer aisément des écosystèmes artificiels allants de la création spontanée par le milieu des organismes artificiels à l'ontogenèse, la sélection naturelle et la phylogenèse. Dans la suite, nous présentons les différents modules qui fournissent cet aspect générique à l'application.

3.3.1 Module d'expression génétique

L'utilisateur peut créer des organismes virtuels en positionnant les cellules une par une à leur place exacte dans l'organisme, pendant la phase de l'initialisation. Pour ce faire, il suffit de :

- choisir le type d'organe parmi les types d'entités élémentaires qui apparaissent dans la fenêtre d'initialisation (voir Figure 3-1)

- initialiser ses variables qui représentent les paramètres de leur fonctionnement et leur état physiologique

- la positionner à l'aide de la souris dans la matrice qui représente l'espace de l'application et

- répéter cette tache pour toutes les cellules jusqu'à la création de l'organisme.

Ainsi, il peut étudier l'équilibre des fonctions physiologiques et métaboliques de l'organisme. Dans la Figure 3-1 nous avons créé un organisme qui est composé d'un estomac (cellules bleus au centre d'organisme), un organe de capture de nourriture (cellules mauves), un organe de reproduction (cellules jaunes), un organe d'attaque (cellules vertes) et des cellules de la peau ou d'isolation thermique (cellules qui entourent l'organisme)

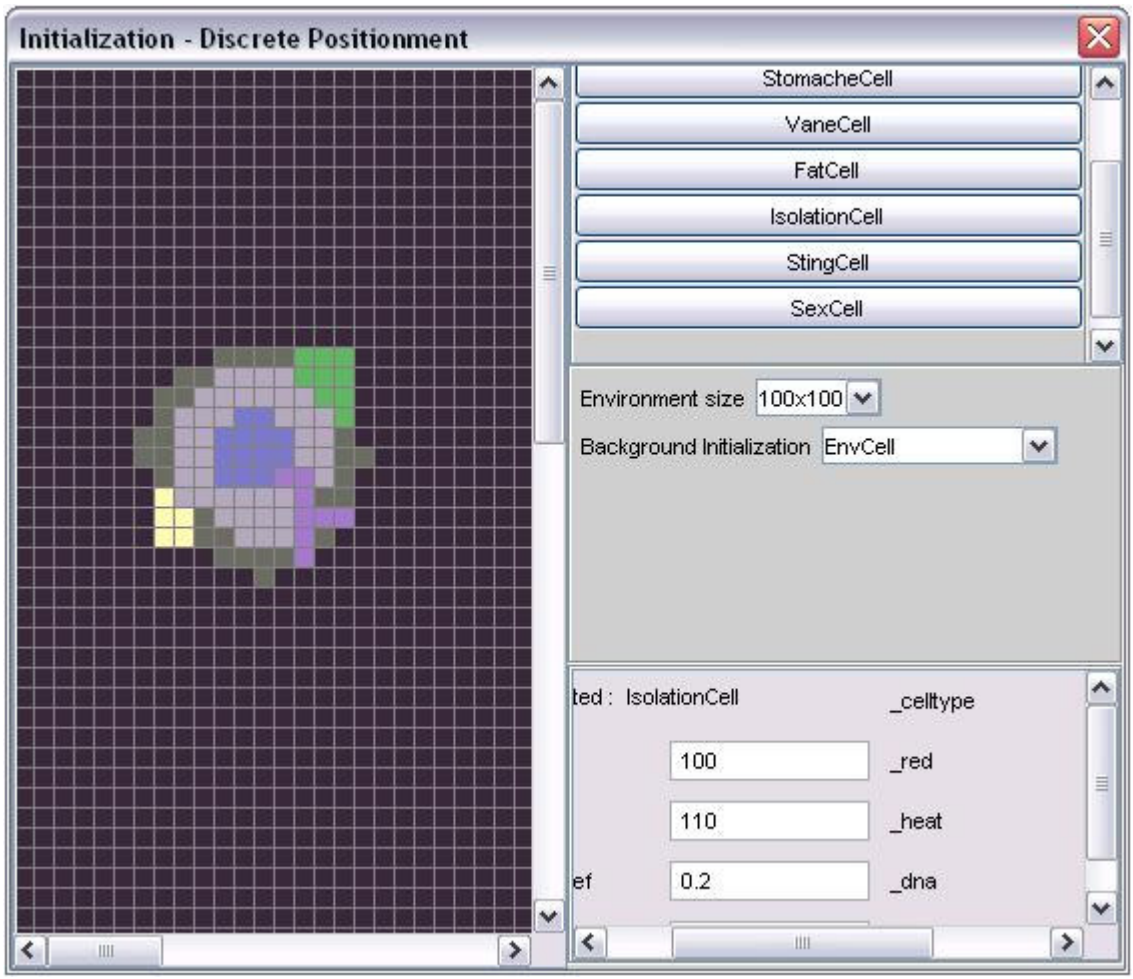


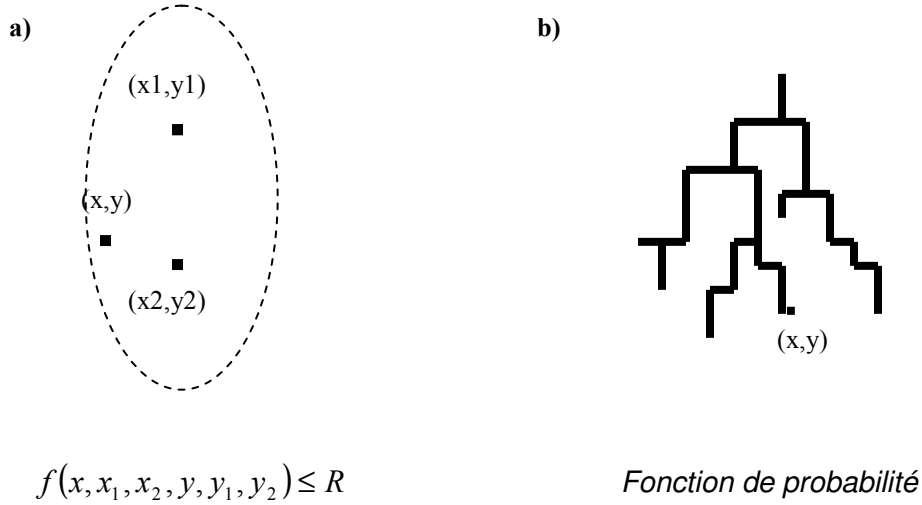
Figure 3-1 : Initialisation d'une application cellule par cellule

D'autre part, l'utilisateur peut coder l'information génétique d'un organisme à l'aide du module d'expression génétique et laisser l'outil gérer le processus de réplication et de différentiation des cellules qui résulte à la création d'un organisme adulte à partir d'une cellule souche. Ainsi, l'utilisateur choisit la forme de chaque organe de l'organisme de l'ensemble de formes de base génériques déjà programmés, règle les paramètres de la forme et ensuite définie la position de l'organe dans l'organisme.

L'outil gère l'expression génétique de la façon suivante : chaque cellule de l'organisme porte l'ADN qui contient des informations codant différents aspects de l'organisme adulte, notamment l'information sur les différents organes, leurs fonctions, leur forme et leur position. Aussi, comme décrit précédemment, l'ADN code des conditions dans lesquelles, une cellule va se différencier pour donner naissance à une cellule d'un organe de l'organisme, ou va se répliquer en cellule identique, si elle dispose d'une quantité d'énergie suffisante pour se répliquer.

Les fonctions qui sont utilisées pour produire la forme des organes peuvent être déterministes ou pseudo aléatoires.

- **Les fonctions déterministes** définissent la forme des organes en déterminant si une cellule appartient à un organe donné selon sa position. La Figure 3-2a démontre que la cellule (x,y) appartient à l'organe en question, si elle réalise la condition : $f(x, x_1, x_2, y, y_1, y_2) \leq R$. Les positions (x1,y1) et (x2,y2) sont les centres de la forme.
- **Les fonctions pseudo aléatoires** créent la forme de l'organe concerné localement et progressivement, suivant une direction qui s'éloigne de l'origine de la forme (comme la création de racines ou de branches d'arbres), avec des paramètres qui correspondent aux probabilités de changement de direction et de création des jonctions. Par exemple, avec ce type de fonctions sont créés les artères des organismes. La Figure 3-2b présente un tel exemple.



$f(x, x_1, x_2, y, y_1, y_2) \leq R$ *Fonction de probabilité*

Figure 3-2 : Fonctions de création de formes

3.3.2 Module de reproduction des organismes et de création des organismes

Ce module a deux fonctions. D'une part, il est utilisé pendant la reproduction des organismes afin de produire l'information génétique de l'organisme enfant. Il produit le nouveau ADN basée sur l'information génétique des organismes parents par le processus de « crossing-over » et le processus des mutations génétiques. Le « crossing-over » consiste à combiner de façon aléatoire les gènes de l'ADN des organismes parents afin de produire l'ADN de l'organisme enfant. Dans ce processus peuvent intervenir des mutations génétiques consistant à altérer avec une probabilité donnée un (ou plusieurs) gène de l'organisme enfant en lui attribuant une nouvelle valeur aléatoire.

D'autre part, ce module peut être utilisé afin de produire une chaîne d'ADN de façon aléatoire, dans un processus de création spontanée des organismes artificiels.

3.3.3 Module de mouvement

L'approche utilisée pour décrire et simuler des systèmes complexes consiste à les décrire au niveau de leur structure la plus fine (les entités élémentaires qui le façonnent et leurs interactions), puis en simulant l'évolution de ces entités, faire émerger le comportement du système global.

Afin de décrire par cette approche le mouvement d'un organisme dans un environnement qui lui impose une résistance au mouvement, l'outil va procéder de la façon suivante.

Il va calculer la vitesse de chaque entité élémentaire en fonction de sa vitesse actuelle et des interactions (forces) avec les entités voisines. Ensuite il va déterminer la position de chaque entité élémentaire en fonction de sa position actuelle et de sa vitesse. Ainsi, une cellule qui se trouve à la surface extérieure de l'organisme à la direction de son mouvement interagit avec les entités d'environnement (force de frottement) ce qui résulte au changement de sa vitesse, et par ce l'intensité des ses interaction avec ses cellules voisines dans l'organisme (propagation de la force de frottement). Les accélérations des organismes et les forces entre organismes seront décrites par des processus similaires.

Le module de mouvement offre à l'utilisateur un niveau d'abstraction alternatif pour la description des mouvements des organismes, qui est plus facile à décrire et plus rapide à simuler. Dans ce niveau d'abstraction, un organisme est traité dans son ensemble comme une entité avec une vitesse, une position, une forme et une masse. Pendant l'évolution de l'organisme, le module du mouvement calcule la masse de l'organisme en additionnant les masses de toutes les cellules de l'organisme. De plus, le module calcule la masse musculaire de l'organisme et la résistance qui s'oppose à son mouvement en fonction de sa forme dans toutes les directions possibles de mouvement. Quand l'organisme accélère, le module du mouvement calcule son accélération en fonction de la masse musculaire des organes moteurs (force d'accélération) et de sa masse totale. Le module calcule

aussi l'accélération du à la résistance au mouvement, en fonction de sa vitesse, du facteur de la résistance de l'environnement et du coefficient de résistance relatif à la forme que présente l'organisme dans la direction du mouvement. Finalement, le module calcule les nouvelles vitesses des organismes après un contact ou une collision.

3.3.4 Module de sélection «naturelle»

La sélection naturelle est un processus idéal pour étudier l'adaptation génétique d'une espèce à son environnement. Par contre, c'est un processus assez long. Le module de sélection «naturelle» reproduit un processus inspiré de la sélection naturelle, mais engendre une évolution et une adaptation beaucoup plus rapides. L'utilisateur peut employer ce module pour étudier la sélection naturelle mettant en jeu un ou plusieurs gènes d'un organisme. Pour ce faire, il décrit l'organisme et son environnement et spécifie les gènes en question. Il définit la population initiale de l'organisme en question et éventuellement des populations d'autres organismes. Ensuite, la simulation de cet écosystème est lancée. Pendant cette simulation, les gènes sélectionnés par l'utilisateur subissent des mutations avec une probabilité définie par l'utilisateur. Au bout d'un grand nombre d'itérations l'utilisateur vérifie si les gènes de la population parviennent à se stabiliser, et si c'est le cas, il considère qu'ils se sont adaptés à l'écosystème donné. Il peut alors vérifier les propriétés que ces gènes confèrent à l'organisme et faire le lien avec les contraintes que l'environnement impose à l'organisme. Ce processus peut être plus ou moins long. L'utilisateur peut agir sur la vitesse de convergence en choisissant les probabilités de mutation des différents gènes. Afin d'agir de façon plus spécifique sur le processus de l'évolution, le module offre à l'utilisateur la possibilité de guider de façon plus contrôlé ce processus.

Par exemple, si un seul gène est soumis au processus de l'évolution génétique, l'utilisateur pourra forcer le module de sélection naturelle à procéder de la façon suivante :

- placer d'abord dans l'environnement donné un organisme correspondant à la description de l'utilisateur et deux autres ayant des mutations dans le gène choisi, et les laisser évoluer.

- une fois qu'un seul organisme reste vivant, le module relance cette étape avec trois nouveaux organismes. Un des trois est identique à l'organisme qui a survécu (qui a des bonnes chances d'être mieux adapté que les deux autres) et les autres deux sont des mutations de cet organisme.

Afin d'assurer la convergence du gène affecté, le module produit des mutants qui ont ce gène légèrement altéré. Ainsi, dans l'un des mutants la valeur de la variable du gène en question peut être légèrement diminuée et dans l'autre légèrement augmentée. Plus généralement, si plusieurs gènes sont soumis au processus de l'évolution, l'utilisateur peut forcer le module à créer à chaque étape un nombre d'organismes mutants, suffisant pour couvrir toutes les combinaisons de mutation possibles. Comme précédemment, ces étapes sont répétées jusqu'à la convergence des gènes (leur adaptation à l'environnement).

Dans la Figure 3-3 nous présentons la fonctionnalité du module de sélection naturelle. Nous observons dans cette figure qu'à chaque fois trois organismes sont placés dans l'environnement artificiel. Un parmi eux est l'organisme survivant de la dernière étape du module et les autres deux sont des mutations génétiques de cet organisme.

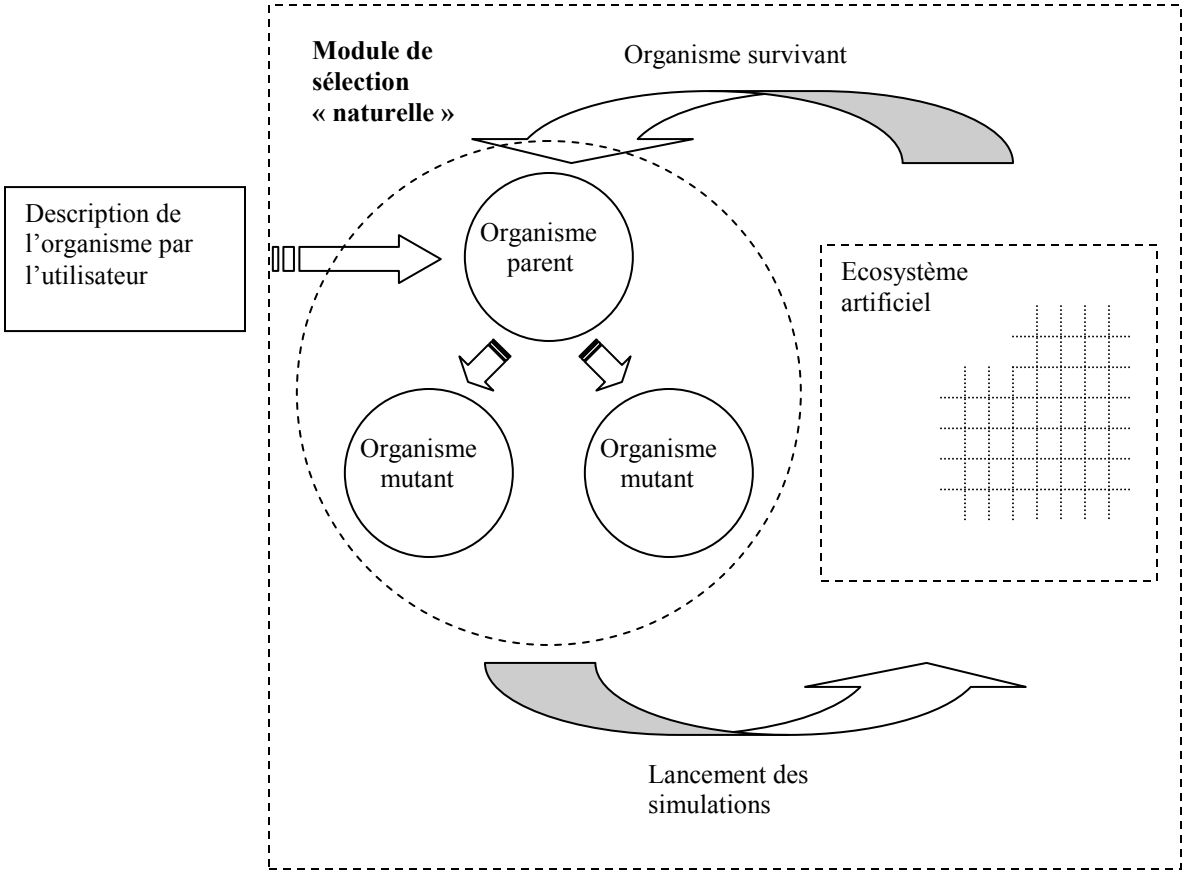


Figure 3-3 : Le Module de sélection « naturelle »

Grâce à ces modules, l'utilisateur peut créer et expérimenter une grande variété d'écosystèmes artificiels. A titre d'illustration, nous présentons dans la suite de ce chapitre deux exemples d'expérimentation.

3.4 Exemples

3.4.1 Phylogenèse

Dans cette section nous présentons des expérimentations qui traitent le processus de la phylogenèse et l'adaptation d'une espèce aux changements environnementaux. Les organismes virtuels de ces expérimentations naissent, évoluent et se reproduisent dans un environnement artificiel. Cet environnement impose certaines contraintes sur l'évolution et la survie des organismes. En particulier, la nourriture, produite de manière aléatoire dans l'espace et le temps, est limitée et les organismes doivent se déplacer pour la trouver, la capturer et la transformer en énergie nécessaire pour leurs fonctions physiologiques et leur survie. De plus, l'environnement peut avoir une température plus froide par rapport à la température des organismes, les obligeant de dépenser de l'énergie pour maintenir leur température. L'environnement impose également une résistance à leur mouvement.

Dans un environnement avec une cadence de production de nourriture, une température et un facteur de résistance au mouvement donné, la sélection naturelle favorisera la survie et la multiplication des organismes qui, suivant leur information génétique, utilisent leur énergie de façon optimale, par rapport à ces conditions, pour se déplacer, se nourrir, se développer et se reproduire. Par conséquent, une espèce d'organismes virtuels qui évolue dans un tel environnement, adaptera par le processus de la phylogenèse, les gènes qui affectent la façon dont les organismes utilisent leur énergie pour accomplir ces taches. Dans l'expérimentation suivante, nous analysons l'adaptation des gènes concernés d'une espèce par le processus de la phylogenèse selon les changements environnementaux.

3.4.1.1 Présentation et analyse des résultats expérimentaux

Dans l'analyse qui suit, nous focalisons notre attention sur l'évolution du gène qui détermine la quantité d'énergie que l'organisme dépense pour se mouvoir, ce qui détermine de façon indirecte la quantité de déplacements que l'organisme effectue. Nous allons en particulier étudier comment les organismes adaptent la distribution de la consommation d'énergie entre diverses tâches afin de s'adapter aux contraintes de l'environnement.

Afin de survivre et se reproduire, les organismes capturent la nourriture se trouvant dans l'environnement et la transforment en énergie, indispensable pour les fonctions métaboliques de base, pour leur développement, pour le maintien de leur température, pour se déplacer et pour se reproduire. L'utilisation de leur énergie d'une manière optimale est cruciale pour leur survie. Dans un premier temps, nous décrivons les paramètres qui déterminent la quantité de l'énergie qu'un organisme consomme pour chacune de ces tâches. Puis, nous exécuterons quelques simulations et nous examinerons la façon dont le processus de la sélection naturelle affecte ces paramètres en fonction des paramètres de l'environnement.

Consommation d'énergie pour la croissance : Un organisme pendant sa croissance a besoin d'énergie afin de produire de nouvelles cellules par le processus de mitose. La quantité d'énergie qui est nécessaire pour la croissance d'un organisme est proportionnelle au nombre de cellules créées par la division cellulaire (réplication des cellules). L'utilisateur peut définir une quantité fixe d'énergie par cellule avant la simulation.

Consommation d'énergie pour le métabolisme : Chaque cellule d'un organisme consomme une quantité d'énergie par cycle, afin d'exécuter ses fonctions métaboliques de base nécessaires pour l'entretien de la vie. Cette quantité aussi est déterminée par l'utilisateur. Cette quantité peut varier selon le type de la cellule et la fonction qu'elle réalise.

Consommation d'énergie pour la reproduction : Pendant la reproduction, les organismes consomment une certaine quantité d'énergie, dont une partie est transmise à l'organisme enfant, afin de lui fournir les ressources nécessaires pour sa survie initiale.

Consommation d'énergie pour le mouvement : Dans le modèle considéré ici, les lois du mouvement s'inspirent de la physique, mais la conservation de l'énergie n'est pas établie au niveau des lois physiques, étant donné que nos simulations sont effectuées au niveau des fonctions comportementales.

Pour assurer la fonction du mouvement, les organismes consomment de l'énergie pour accélérer et pour maintenir une certaine vitesse. L'énergie nécessaire pour accélérer d'une vitesse V1 à une vitesse V2 est :

$$Ek = 1/2 \times m \times (V_2^{\,2} - V_1^2) \qquad \text{(eq1)}$$

D'autre part, l'énergie consommée par un organisme pour maintenir sa vitesse est :

$$Er = Fs \times Fr \times V^2 \qquad \text{(eq2)}$$

où Fs est un facteur qui dépend de la forme qui présente l'organisme dans la direction de son mouvement et Fr est un facteur représentant la résistance de l'environnement.

Consommation d'énergie pour le maintien de la température : L'énergie consommée pour cette tache et par unité de temps est donnée par l'équation 3:

$$Eh = \int^{surface} Fh \cdot \Delta\theta / nc \cdot ds \qquad \text{(eq3)}$$

où Fh est un facteur qui représente le taux de transfert d'énergie thermique (chaleur) à travers les cellules d'isolation de l'organisme (sa peau), nc l'épaisseur de la peau à un point de la surface, $\Delta\theta$ la différence de la température entre l'organisme et son environnement et ds le différentiel de surface.

Dans la Figure 3-4 nous démontrons l'adaptation génétique de l'espèce aux changements environnementaux. La partie basse de la figure montre les changements environnementaux qui sont produits à la simulation. La partie haute de la figure démontre l'adaptation du gène de l'espèce qui définit la façon dont les organismes se déplacent en fonction de la difficulté au déplacement dans le milieu. La partie au milieu de la figure reflète les changements de la population moyenne.

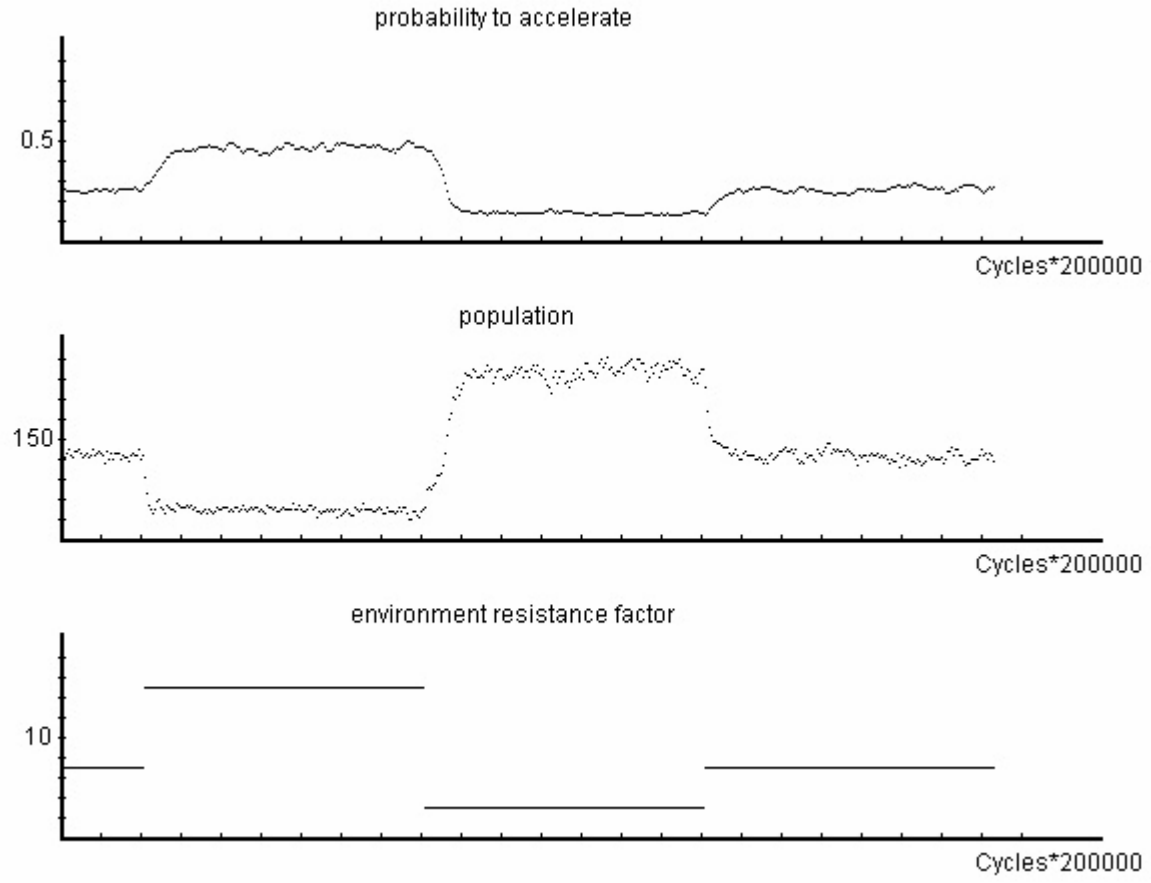


Figure 3-4 : représentation graphique

La population moyenne augmente dans un environnement imposant une petite résistance au mouvement et diminue dans un environnement imposant une grande résistance au mouvement. Ce comportement est prédictible puisque dans le premier cas, les organismes dépensent moins d'énergie pour se mouvoir et trouver de la nourriture et peuvent survivre plus longtemps. D'autre part, par les processus du « crossing-over » et des mutations génétiques, l'espèce adapte le gène qui détermine la quantité d'énergie dépensée pour le mouvement. Ainsi, dans un environnement imposant une grande résistance au mouvement, les organismes adaptent ce gène de façon à ce qu'ils consomment plus d'énergie pour leur mouvement. Cette adaptation est nécessaire car si les organismes ne dépensaient pas suffisamment d'énergie pour se mouvoir avec une certaine cadence, ils ne pourront même pas capturer une quantité de nourriture suffisante pour couvrir les besoins énergétiques de leurs fonctions vitales (par exemple, la maintenance de leur niveau de température). Les résultats des simulations sont par conséquent en accord avec les prédictions théoriques, et le système montre des capacités évolutives similaires aux systèmes naturels.

Les résultats de la simulation démontrent l'adaptation de l'espèce par évolution aux changements environnementaux.

Fonction « olfactive » : Dans une autre expérimentation nous avons considéré des organismes artificiels avec des organes olfactifs. Comme dans l'expérimentation décrite ci-dessus, les organismes virtuels de cette expérimentation naissent, évoluent et se reproduisent dans un environnement artificiel. Par contre, dans cet environnement la nourriture se produise avec une petite cadence. Ainsi, il est crucial pour les organismes d'utiliser les informations recueillies par leur système « olfactif » afin de se déplacer d'une façon optimale pour la trouver, la capturer et la transformer en énergie nécessaire pour leurs fonctions physiologiques et leur survie.

Les organismes de cette application ont des capacités à se diriger par fonction « olfactive » vers les sources de nourriture. Ce comportement est produit par des mécanismes simples, tels que des organes olfactifs placés sur différentes parties de

l'organisme et des fonctions simples comparant les réponses de ces organes aux molécules odoriférantes diffusées dans l'environnement. Les organismes virtuels de cette expérimentation se reproduisent et par le processus de la phylogenèse changent les gènes qui codent les paramètres des fonctions – réponses aux signaux reçus par les organes olfactifs qui déterminent la direction du mouvement des organismes.

Au début des simulations, les organismes se déplaçaient de façon aléatoire. Avec le processus de la sélection naturelle, les organismes ont acquis la capacité d'utiliser leurs organes olfactifs et se déplacer vers la nourriture, afin de la capturer. Dans la Figure 3-5, nous observons des organismes qui se déplacent vers la nourriture. La couleur verte indique l'odeur de la nourriture. Chaque organisme a quatre cellules olfactives dans quatre directions opposées. Les cellules rouges qui se trouvent en bas, à la partie gauche et droite des organismes correspondent aux cellules olfactives qui sont stimulées par la présence de l'odorat (voir Figure 3-5).

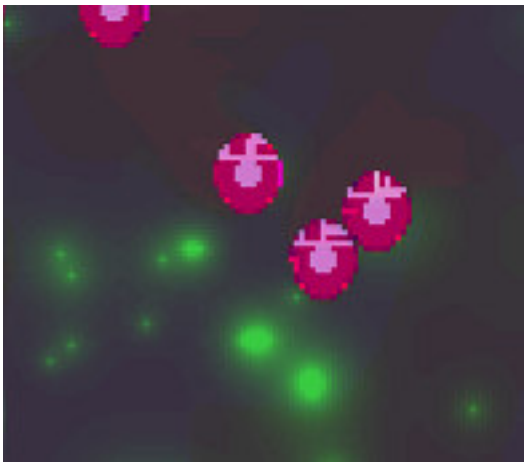


Figure 3-5 : fonction « olfactive »

3.4.2 Création d'organismes artificiels

L'élucidation des processus qui on conduit à l'apparition de la vie sur terre fait parti des plus grands défis scientifiques de nos jours et de nombreuses études lui ont étaient consacrées. Dans certains essais visant à reproduire les conditions chimiques et environnementales qui existaient sur terre avant l'apparition de la vie, des éléments inorganiques simples, tels que l'eau, le dioxyde de carbone, le méthane et l'ammoniac ont étés soumis à l'action du rayonnement infrarouge et des éclairs [62]. Quelques semaines plus tard, les expérimentateurs ont observé l'apparition des acides aminés, les briques élémentaires des protéines. D'autres expérimentations, mimant les conditions terrestres pré biotiques, ont réussi à créer des molécules organiques complexes qui constituent les briques élémentaires de l'ADN.

Le but de telles expérimentations est de montrer que des processus naturels peuvent mener à la création des briques élémentaires de la vie en grandes quantités. Ensuite, au gré des rencontres et combinaisons fortuites de ces briques en grandes quantités et pendant des très longues périodes de temps, l'objectif est d'aboutir aux toutes premières entités rudimentaires ayant des caractéristiques d'organismes vivants. De telles caractéristiques sont par exemple la faculté de récupérer des ressources dans l'environnement pour pouvoir maintenir leur structure malgré les agressions environnementales qui pourraient la détruire, l'empêcher de s'agrandir, ou de se répliquer. Ainsi, une explication plausible de l'apparition de la vie sur terre suggère que des processus impliquant des gazs simples présents dans l'atmosphère terrestre, des explosions volcaniques, de la lumière solaire et des orages ont créé, il y a trois quatre billions d'années, une « soupe primaire » d'éléments organiques dans les océans. La concentration de la matière organique augmentait dans des lacs desséchés ou sur les côtes, et en présence de l'énergie (par exemple du rayonnement solaire) des molécules plus grandes se formaient. En absence des prédateurs (des organismes vivants) ces molécules circulaient librement dans la « soupe primaire » qui devenait de plus en plus concentrée.

Ensuite, comme présenté dans [62], peut-être après des centaines de millions d'années, d'une manière aléatoire a été créée une molécule avec la capacité de se répliquer, (c'est-à-dire produire des copies d'elle-même), en utilisant les briques élémentaires (grandes molécules) se trouvant en grandes quantités dans la « soupe primaire ». Le processus de l'autoréplication pourra s'expliquer en imaginant cette molécule comme une matrice composé des différentes briques élémentaires avec chaque bloc de base ayant la potentialité de s'agglutiner à un autre bloc de base, résultant à la création d'une chaîne des briques élémentaires identique à la chaîne initiale, ou représentant son négatif qui produira par un processus similaire la molécule initiale. Ce processus ressemble à la création des cristaux, mais si les deux chaînes se séparent, nous obtenons deux nouvelles molécules ayant la capacité de l'autoréplication.

Néanmoins, dans les conditions hasardeuses de la nature, le processus de la reproduction d'une copie n'est pas parfait. Ainsi, des fausses copies naissaient de temps en temps, dont certaines pourraient être plus performantes que l'original. De cette façon, la « soupe primaire » devenait plus riche, avec plusieurs descendants de cette molécule, des populations dont la taille dépendait de la stabilité de chaque molécule, de leurs capacités d'exploiter les ressources environnementales pour se défendre ou s'agrandir et leur aptitude à proliférer par autoréplication. Ainsi, diverses propriétés favoriseraient la population de certaines molécules comme par exemple la vitesse de reproduction qui favoriserait les molécules les plus fertiles, ou la précision des copies qui favoriserait les molécules produisant les copies les plus fidèles.

Un autre élément important dans ce processus d'évolution est la compétition [62]. Dans la « soupe primaire » les quantités des grandes molécules organiques, qui servaient comme des briques élémentaires pour les molécules autocopiantes, étaient limitées et par conséquence, précieuses. Pour les obtenir, les différents types de molécules se « battaient » pour devenir plus stables ou rendre les autres moins stables. C'était une bataille de survie qui se faisait bien entendu inconsciemment, par la seule action de la sélection naturelle, c'est-à-dire que les

molécules autocopiantes les plus stables et/ou ayant la capacité de rendre les autres instables survivaient plus que les autres et leur population augmentait plus rapidement et s'imposait. Les améliorations des molécules autocopiantes qui dérivaient par ce processus de la sélection naturelle étaient cumulatives et au fil des générations les molécules autocopiantes devenaient de plus en plus complexes et ingénieuses. Ainsi, quelques espèces de molécules autocopiantes « inventaient » des méthodes pour casser les liaisons chimiques d'autres molécules afin d'utiliser leurs briques élémentaires pour leur propre reproduction. D'autres espèces se protégeaient en formant autour d'elles un contour de protéine, aboutissant aux premières cellules que nous pouvons considérer comme vivantes. La compétition devenait de plus en plus forte et l'évolution était question de temps.

Nous observons que ces processus mettent en jeu des très nombreuses étapes impliquant des combinaisons de la matière complexes, donc très peu probables de se produire au hasard et qui doivent de surcroît s'enchaîner les unes après les autres pour donner naissance même à des organisme monocellulaires les plus simples qui puissent exister. Vu la probabilité extrêmement faible qu'a chacune de ces étapes pour se produire, l'existence des ingrédients impliqués en très grandes quantités est nécessaire. Ceci devient possible uniquement pour des molécules et leurs combinaisons qui, une fois apparues dans une de ces étapes, ont la capacité de proliférer en très grande quantité pour créer ainsi les conditions de réalisation de l'étape suivante. Ainsi, la capacité de prolifération devient une des propriétés sine qua non pour les molécules et leurs combinaisons qui mènent à l'apparition de la vie. Il n'est donc pas un hasard que cette propriété soit partagée par tous les organismes vivants.

Dans la suite nous présentons des expérimentations qui s'inspirent de ces processus, mais bien entendu, il s'agit d'une implémentation très modeste. Avant de parler de cette expérimentation, résumons certaines des conditions nécessaires pour l'apparition de la vie :

- L'existence, dans les lois qui régissent la matière, d'un potentiel de génération des molécules organiques (briques de la vie).
- L'existence, dans les lois qui régissent la matière, d'un potentiel de combinaison des briques de la vie en molécules plus complexes ayant la propriété de se combiner (au fil des rencontres fortuites), avec les briques de la vie pour produire leur propre copie (molécules autocopiantes)
- L'existence, dans une ou plusieurs parties de l'univers, des conditions de création en grande quantité des ces briques ainsi que des conditions permettant la stabilité des molécules autocopiantes

Une autre condition concerne le temps nécessaire pour que les différentes étapes menant à la vie se réalisent. Ainsi, étant données les quantités et les densités des briques de la vie presentes dans une région de l'univers (par exemple la terre), et les valeurs des probabilités données de formation des molécules autocopiantes pour ces densités, un temps suffisamment long doit s'écouler pour que les rencontres fortuites des briques de la vie donnent naissance à un nombre relativement élevé de molécules autocopiantes, pour que par la suite ces molécules se reproduisent en grandes quantité, divergent, et que les copies divergées se reproduisent de nouveaux en grandes quantités. Pour commencer une nouvelle étape, laquelle, après le passage d'un temps suffisamment long aboutira par combinaison des molécules autocopiantes à des structures autocopiantes plus complexes et ainsi de suite.

Certaines conditions supplémentaires sont nécessaires pour qu'il y a une évolution vers des organismes de plus en plus complexes, telles que :

- L'existence, dans les lois qui régissent la matière, d'un potentiel de combinaison des briques de la vie en molécules ayant diverses propriétés et « exploitables » par les molécules autocopiantes , à commencer par des propriétés les plus simples, telles que la production des molécules complexes de protéines, l'acquisition par leur biais de la résistance aux agressions environnementaux (voir même aux agressions des autres molécules autocopiantes), la capacité de casser

d'autres molécules autocopiantes pour exploiter leur ressources, jusqu'aux propriétés les plus avancées telles que les fonctions de motricité, du sens, ou de l'intelligence.

Nous observons, que mises à part les conditions concernant le potentiel de la matière de se combiner pour donner naissance aux briques de la vie, aux molécules autocopiantes et aux molécules ayant des fonctions permettant une adaptation de plus en plus performante, nous avons aussi des conditions concernant les quantités de matière (très grandes) et les durées du temps (très longues) nécessaires pour que des molécules de très grande complexité, ayant donc une probabilité d'apparition très faible, soient créées et prolifèrent en grande quantité afin de créer à chaque fois les conditions de réalisation de l'étape suivante du processus.

Ne disposant ni la puissance de calcul nécessaire pour simuler des quantités de matière énormes, ni des temps de calcul extrêmement longs, nécessaires pour reproduire un processus semblable dans nos mondes artificiels, nous n'allons expérimenter des processus de création de vie artificielle que de façon caricaturale. Pour ce faire, nous créerons des applications que nous allons doter du potentiel de l'apparition d'ADN artificiel (variables codantes et fonctions associées) et de la combinaison des diverses types de cet ADN pour former des cellules. Ce potentiel consiste à la génération aléatoire de ces variables et fonctions avec une probabilité déterminée par l'utilisateur de l'outil. Vu le temps de calcul relativement faible que nous disposons, nous avons d'intérêt à donner à ces probabilités des valeurs relativement élevées. Nous allons ensuite laisser fonctionner la simulation pour voir les ADN et les cellules se former de façon aléatoire et évoluer dans l'écosystème pour produire des organismes artificiels de plus en plus performants par la sélection « naturelle ». Néanmoins, rien n'empêche, si nous disposons des capacités de calcul plus importantes et des temps d'expérimentations très longs, d'utiliser notre logiciel pour raffiner ce processus en étapes de plus en plus élémentaires afin de se rapprocher de plus en plus aux étapes qui pourraient être impliquées dans le processus naturels menant à l'apparition de la vie. Un outil comme celui que nous

avons développé facilitera grandement cette tâche en permettant la mise en place de telles expérimentations sans avoir à développer à chaque fois un logiciel dédié.

Pour réaliser cette « caricature » de création de vie artificielle, nous avons créé une nature artificielle consistant en un espace dans lequel existent des sources de nourriture. Ensuite nous avons muni cette nature avec le potentiel de générer des entités (cellules) contenant des gènes (variables et fonctions associées). Pour effectuer la génération des cellules dans une partie de l'espace, nous avons choisi d'utiliser une probabilité proportionnelle à l'énergie accumulée (par exemple température) et aux quantités de nourriture présentes dans cette partie de l'espace. La mise en jeu de la nourriture dans la détermination de la valeur de cette probabilité signifie ici que les briques de la vie sont à la fois les éléments constituants et les éléments nutritifs des premières cellules. L'information codée par les gènes correspond aux différents paramètres qui déterminent différentes fonctions, telles que le nombre et le type des différents organes de l'organisme, leur place, les paramètres qui déterminent leur forme, ainsi que les paramètres qui déterminent les fonctions des cellules de ces organes. Ces informations forment l'ensemble de l'ADN d'une cellule et leurs valeurs sont générées de façon aléatoire. Une fois que nous avons préparé notre application, nous lançons des simulations pour réaliser des expérimentations de création de vie artificielle.

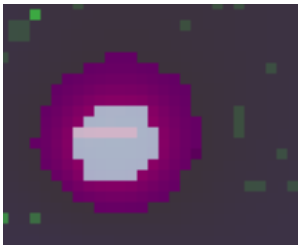
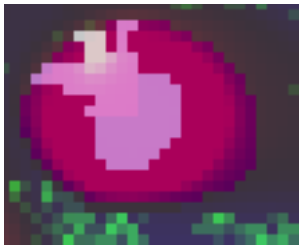
3.4.2.1 Simulations

Après avoir doté l'application avec le potentiel de générer des cellules contenant de l'ADN codant diverses fonctions (isolation thermique, organes de capture et d'absorption de la nourriture, organes moteurs, organes de reproduction) et ayant diverses probabilités d'apparition, nous avons exécuté des simulations. Dans ces simulations, nous avons observé une évolution vers des formes de plus en plus performantes. Nous présentons dans la suite quelques observations.

Lors de ces expérimentations nous observons la formation d'une grande variété d'organismes. Dans la grande majorité des cas, les organismes qui apparaissent manquent de l'ingénierie et meurent rapidement. Par exemple dans la Figure 3-6a nous observons un organisme qui n'a pas un organe de capture de nourriture correctement développé et aussi, ne peut pas se nourrir. Cet organisme ne peut pas survivre. Au gré des créations fortuites, des organismes plus ingénieux naissent qui ont une durée de vie moyenne plus élevée. Par exemple des organismes qui forment une couche extérieure des cellules d'isolation thermique (correspondant à l'organe de la peau). Ces organismes n'ont pas acquis la capacité de se nourrir, mais ont une vie moyenne plus longue parce qu'ils ont réduit les déperditions d'énergie (chaleur) vers l'environnement. Par la suite nous observons l'apparition des organismes qui ont un organe de capture de la nourriture et un organe de transformation de la nourriture en énergie. Ces organismes ne se déplacent pas mais utilisent les petites quantités de nourriture qui se trouvent autour d'eux pour prolonger leur survie. Ainsi, après quelque temps, des formes plus stables et performantes commencent à peupler l'environnement.

Dans les Figures 3-6b et 3-6c, nous observons des organismes qui ont la possibilité de se déplacer, de capturer de la nourriture et de la digérer. Dans la Figure 3-6d et 3-6e, nous observons des organismes carnivores, qui se nourrissent en absorbant les cellules des autres organismes. Finalement, dans la figure 3-6f nous observons un organisme qui a la possibilité de se multiplier.

a b c


d e f



Figure 3-6 : Création des organismes

Dans les figure 3-7a et 3-7b nous observons un organisme carnivore en train de se nourrir en absorbant les cellules d'un autre organisme. Les cellules vertes correspondent aux cellules de l'organe d'attaque de l'organisme

a b

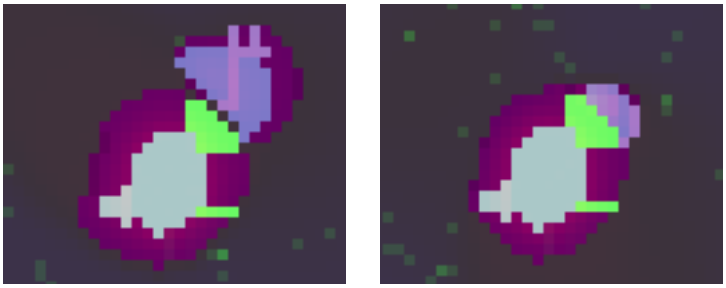
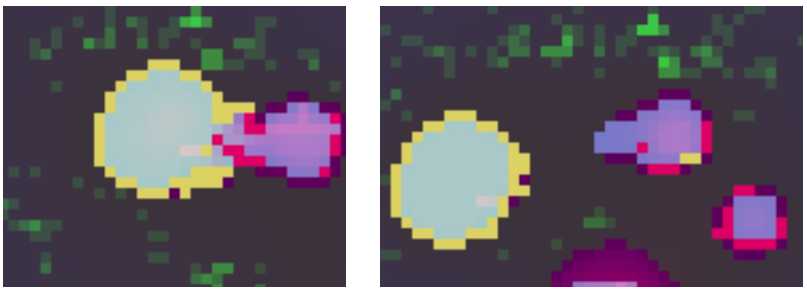


Figure 3-7 : attaque d'un organisme carnivore

Dans les figures 3-8a, 3-8b et 3-8c nous observons l'accouplement de deux organismes donnant naissance à deux nouveaux organismes.

a b



c

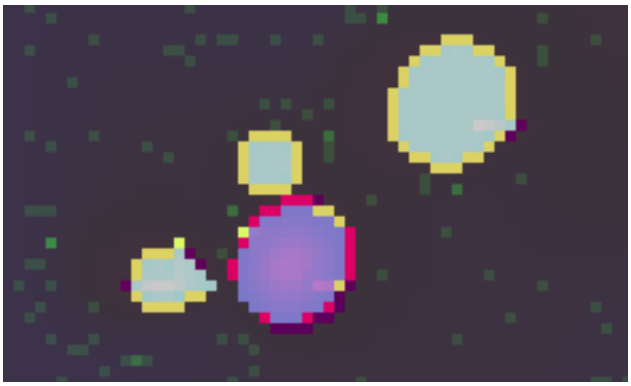


Figure 3-8 : reproduction

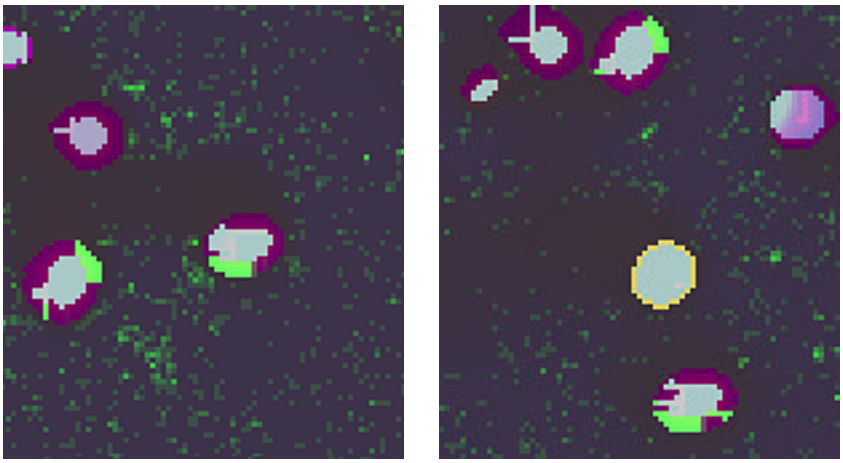
Tous ces organismes naissent de manière aléatoire. Par contre, après un grand nombre de cycles de simulation, nous observons dans l'écosystème artificiel un nombre d'organismes qui ont survécu plus longtemps grâce à ces aptitudes.

Cette complexification des organismes et augmentation des performances se produit par un processus aléatoire soutenu par le fait que, comme dans la nature, à chaque période de ce processus se sont les organismes de la population existante les plus performants, pour l'environnement donné, qui vont survivre plus longtemps et vont proliférer plus rapidement et vont ainsi s'imposer. Ainsi, dans la figure 3-9a nous observons l'apparition de quelques organismes carnivores parmi la population de l'écosystème. Ces organismes sont plus performants. Dans la figure 3-9b les premiers organismes ayants la possibilité de se répliquer apparaissent.

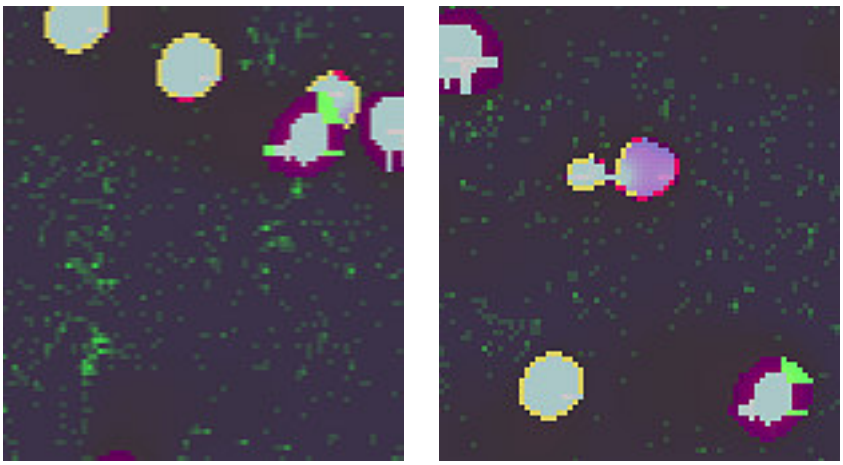
La figure 3-9c montre un organisme carnivore en train de se nourrir en absorbant les cellules d'un organisme tandis que la figure 3-9d montre l'acte de la reproduction de deux organismes. Ces deux espèces sont les plus performantes et, comme nous observons à la figure 3-9e, après un grand nombre de cycles de simulation, elles sont les seules à survivre.

Les résultats de ces simulations démontrent un phénomène de la nature qui est présenté dans [63]. Ce phénomène est la formation des groupes symbiotiques de deux espèces différentes qui peut permettre l'évolution des organismes d'une espèce, qui autrement ne pourraient pas survivre. Dans notre exemple, les organismes carnivores se nourrissent des organismes de l'autre espèce et, ainsi, acquiescèrent de l'énergie et prolongent leur survie. Nous pouvons constater que, comme c'est souvent le cas pour les groupes symbiotiques [63], une espèce est indépendante et l'autre dépendante.

a b



c d



e

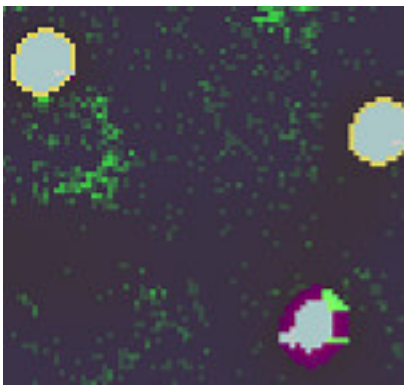


Figure 3-9 : évolution de l'écosystème artificiel

3.5 Conclusions

Dans ce chapitre nous avons présenté la première application qui a été implémentée par l'outil de simulation logicielle. Cette application concerne des systèmes biologiquement inspirés composés d'un environnement dans lequel évolue un certain nombre d'organismes artificiels, qui ont les possibilités du mouvement, de la réplication, de l'évolution d'une cellule à un organisme (ontogenèse), de la mutation, et de l'adaptation à l'environnement. Cette application a servi comme cas

d'étude, afin d'éviter de développer l'outil de simulation logicielle sur des considérations abstraites.

Nous avons effectué plusieurs expérimentations afin de démontrer la richesse des systèmes biologiquement inspirés ainsi que la capacité de cette application de traiter divers processus. En particulier, nous avons expérimenté sur le processus de la phylogenèse et l'adaptation d'une espèce aux changements environnementaux et sur des processus de création des organismes artificiels, inspirés des processus qui on conduit à l'apparition de la vie sur terre.

Dans la suite nous allons présenter le deuxième exemple d'application implémentée par l'outil de simulation logicielle qui concerne le domaine de la physique fondamentale. Cette application est très différente des écosystèmes et illustre ainsi l'universalité de notre outil, qui permet à l'utilisateur de réaliser avec un moindre effort et d'explorer par simulation divers types de systèmes complexes constitués d'un ensemble d'entités simples.

Chapitre 4

4 Systèmes de particules, lois d'interactions et émergence de l'espace-temps

4.1 Généralités

Notre second exemple d'application concerne le domaine de la physique fondamentale. Cette application est très différente des écosystèmes et illustre ainsi l'universalité de notre outil, qui permet à l'utilisateur de décrire avec un moindre effort divers types de systèmes constitués d'un ensemble d'entités simples et de les explorer par simulation. En effet, comme les systèmes physiques sont composés par des entités élémentaires, les particules, ces systèmes font bien partie des applications adressées par notre outil. Ainsi, afin de décrire et simuler un tel système il suffira de décrire à l'aide du menu de notre outil les types des particules et les fonctions selon lesquelles évolue leur état, puis donner le nombre de chaque type des particules et leur état initial. Notre outil nous permet en effet de décrire et simuler aisément ce type des systèmes complexes. Mais dans ce chapitre nous voulons aller plus loin et étudier l'émergence dans ce type des systèmes certaines propriétés globales telles que la structure globale de l'espace-temps. Pour étudier cette structure il ne suffira pas d'avoir un outil permettant de décrire et simuler le système complexe, mais aussi de disposer des modules permettant d'analyser cette structure. Ceci est l'un des aspects importants adressés dans ce chapitre.

Dans notre univers, certaines propriétés globales semblent inabordables à une approche qui décrit un système uniquement par le biais de l'évolution d'un ensemble d'entités élémentaires. C'est en effet le cas de la structure de l'espace-temps décrite par la théorie de la relativité. En conséquence, l'étude de la structure de l'espace-temps semble inabordable à notre outil de simulation. Malgré cette inadéquation apparente, c'est justement ces aspects que nous allons étudier dans la suite du manuscrit. Nous allons adopter pour cela une approche qui considère que les entités primaires de l'univers sont les particules, tandis que l'espace et le temps sont le sous-produit de l'évolution de l'état des particules [64]. Dans cette approche, la position de chaque particule fait partie des variables d'état de la particule, et l'espace émerge comme une conséquence de l'évolution des états des particules : à tout moment l'espace est déterminé par l'ensemble des valeurs des variables des

positions des particules qui composent l'univers. Selon la même vision, il y a émergence du temps si et seulement si les lois de l'évolution des particules sont invariables dans l'«espace» et dans le «temps».

4.1.1 Contraintes des accélérations

On se pose alors la question : comment l'évolution de l'état des particules peut engendrer la structure globale de l'espace-temps, comme celle décrite par les transformations de Galilée ou les transformations de Lorentz.

Selon la démonstration donnée dans [64], cette structure est déterminée par la forme particulière des lois des interactions des particules. Nous résumons dans la suite certains résultats obtenus dans [64].

- Si l'intensité des interactions entre particules ne dépend pas de la vitesse des particules, la structure de l'espace-temps est galiléenne. Ainsi, cette condition définie la famille galiléenne des lois d'interaction : toute lois d'interaction qui la vérifie engendre un espace-temps galiléen.
- Si l'intensité des interactions dépend de la vitesse des particules en vérifiant une certaine condition appelée dans [64] « Contrainte Relativiste des Accélérations », la structure de l'espace-temps vérifie les transformations de Lorentz. Ainsi, cette condition définie la famille des lois d'interaction de Lorentz: toute lois d'interaction qui la vérifie engendre un espace-temps conforme aux transformations de Lorentz. Ainsi, des lois d'interactions différentes de celles en vigueur dans notre univers engendreront aussi un espace-temps conforme aux transformations de Lorentz, si elles vérifient cette condition.
- Si l'intensité des interactions dépend de la vitesse des particules mais vérifie des conditions différentes de la « Contrainte Relativiste des Accélérations », la structure de l'espace-temps sera « relativiste » (les intervalles du temps et les longueurs changent si nous changeons de système de référence inertiel), et elle vérifiera des transformations différentes de celles de Lorentz. Néanmoins, si nous

imposons des conditions sur la structure de l'espace-temps et sur l'expression des lois de la physique dans différents systèmes de référence inertiels, nous pouvons trouver que les seuls cas possibles sont ceux exprimés par les transformations de Galilée ou de Lorentz. Ainsi, si nous exigeons que les lois restent invariables dans tout système de référence inertiel et les transformations des coordonnées spatiales et temporelles qui opèrent lors du passage d'un système de référence à un autre sont symétriques de celles qui opèrent lors du passage inverse, nous tombons obligatoirement sur les transformations de Lorentz ou de Galilée. Par contre, si nous exigeons uniquement l'invariabilité des lois dans tout système de référence inertiel, d'autres transformations pourraient exister [64]. De telles transformations existeront s'ils existent trois fonctions de v ; $n_t(v)$, $n_x(v)$ et s(v) qui vérifient les conditions suivantes :

- $\begin{cases} n_x(v) = n_x(-v) \\ n_x(0) = 1 \\ n_x(v) \geq 0 \end{cases}$

$\begin{cases} n_t(v) = n_t(-v) \\ n_t(0) = 1 \\ n_t(v) \geq 0 \end{cases}$ (C.1)

$\begin{cases} s(v) = -s(-v) \\ s(0) = 0 \\ s(v)v \geq 0 \end{cases}$

- L'équation $vn_t(v) - un_t(v) - d = 0$ donne pour v une solution réelle unique pour chaque valeur de d, telle que $d \geq 0$ et pour chaque valeur de vitesse de u permise dans l'univers en question.
- Il existe une solution s(v) indépendante de u de l'équation suivante

$$n_t\left(\frac{n_x(v)u}{n_t(v)+s(v)u}+v\right)\left(\frac{n_x(v)u}{n_t(v)+s(v)u}\right)=n_t\left(\frac{-n_x(v)u}{n_t(v)-s(v)u}+v\right)\left(\frac{n_x(v)u}{n_t(v)-s(v)u}\right) \qquad \text{(eq1)}$$

La recherche de telles fonctions $n_t(v)$, $n_x(v)$ et $s(v)$ qui sont différentes de :

$$\begin{cases} n_x(v) = n_t(v) = 1 \\ s(v) = 0 \end{cases}$$ (cas Galiléen) et

$$\begin{cases} n_t(v) = \gamma_v \\ n_x(v) = 1/\gamma_v \\ s(v) = \gamma_v \, v/c^2 \end{cases}$$, avec $\gamma_v = \frac{1}{\sqrt{1 - v^2/c^2}}$ (cas des transformations de Lorentz), reste un problème ouverte.

Si nous disposons d'un ensemble de fonctions $n_x(v)$, $n_t(v)$ et $s(v)$, nous pouvons déduire de la façon décrite dans [64] une condition similaire à la « Contrainte Relativiste des Accélérations ». Cette condition définie à son tour une famille des lois d'interactions qui donnent toutes lieu aux mêmes transformations des coordonnées spatiales et temporelles. D'autres part, si les lois d'interaction d'un « univers » constitué d'un ensemble de particules sont connues, nous pouvons déduire les fonctions $n_x(v)$, $n_t(v)$ et $s(v)$, de la façon décrite dans [64].

4.1.2 Description de l'application

Dans la suite, nous allons décrire comment notre outil permet de créer des expérimentations concernant les mesures des longueurs et du temps dans des univers régis par diverses lois d'interactions des particules, afin de déterminer la structure correspondante de l'espace-temps (transformations lors du passage d'un système de référence inertiel à un autre). Etant donné qu'aucun espace-temps n'existe à priori dans ces univers, mais tout est engendré par un ensemble d'entités (particules) et leurs lois d'interactions, nous avons à faire à des systèmes que notre outil est déjà programmé à traiter. Ainsi, nous pouvons expérimenter de tels univers sans grand effort. Il suffit de décrire le type et le nombre des particules qui constituent l'univers cible et leurs lois d'interactions et lancer ensuite la simulation. Néanmoins, cet outil ne nous permettra pas de déterminer les transformations qui s'opèrent lors des changements des systèmes de référence inertiels. Pour cela,

nous devons pouvoir faire des mesures des distances et du temps dans chaque système de référence inertiel. Ainsi, nous devons disposer des unités de longueur et du temps dans chaque système de référence inertiel. Nous devons aussi disposer des moyens de synchronisation des horloges distantes dans chaque système de référence inertiel. Ceci est par exemple nécessaire afin de mesurer la longueur d'un objet en mouvement par rapport à un système de référence inertiel, car, les coordonnées des extrémités de l'objet doivent être déterminées au même instant dans ce système de référence.

La synchronisation de deux horloges distantes dans un système de référence inertiel se fait en lançant au même instant à partir du milieu de la distance qui les sépare, deux objets qui se déplacent vers les deux horloges avec des vitesses égales mais des signes opposés. Pour déterminer que les vitesses de ces objets sont égales mais des signes opposés, nous devons pouvoir déterminer les instants de leur passage par des positions différentes. Mais pour cela nous avons besoin des horloges placées sur ces positions et synchronisées. Nous arrivons dans un problème de type cercle vicieux. Dans la relativité ce problème est réglé car nous connaissons à priori (par mesures expérimentales) que la vitesse de la lumière est constante dans tout système de référence inertiel et dans toute direction. Nous effectuons alors la synchronisation en utilisant des photons comme objets de synchronisation. Mais si nous ne connaissons pas à priori des objets ayant une vitesse constante, ou si une telle vitesse n'existe pas dans l'univers cible, nous devons trouver un principe de synchronisation différent et universel, c'est-à-dire valable dans tout type d'univers. Nous allons utiliser un tel principe, décrit dans [64], qui sera présenté à la section 4.2.6. Il règle le problème du cercle vicieux que nous avons mentionné, en utilisant des horloges en mouvement comme objets de synchronisation.

En conclusions, notre outil permet de décrire aisément les univers en question et les simuler, mais nous avons besoin certains nouveaux modules logiciels :

- Un module de création des objets rigides et des unités de mesure de longueur (étalons de longueur).
- Un module de mesure des longueurs depuis le système de référence $S_{0.}$
- Un module de mesure depuis le système de référence S des dimensions des objets au repos dans S.
- Un module de création des unités de mesure du temps (horloges).
- Un module de mesures des intervalles du temps depuis le système de référence $S_{0.}$
- Un module de synchronisation des horloges distantes.
- Un module de changement de système de référence d'observation.
- Un module de réalisation des mesures dans le système de référence S.
- Un module de création des objets et processus identiques dans S et S_0

Ainsi, à l'aide de ces modules, l'utilisateur peut créer dans chaque système de référence inertiel des étalons de longueur, des horloges et synchroniser des horloges distantes. Ensuite, il peut créer dans un système de référence inertiel S des objets et processus identiques à des projets et processus se trouvant dans le système de référence $S_{0,}$ effectuer des mesures dans chaque système de référence et comparer les résultats des mesures effectuées. Dans la Figure 4-1, nous présentons un diagramme bloc de l'outil et de ces modules qui démontre la façon dont les différents modules sont employés, afin de créer des processus dans chaque système de référence inertiel et d'effectuer des mesures.

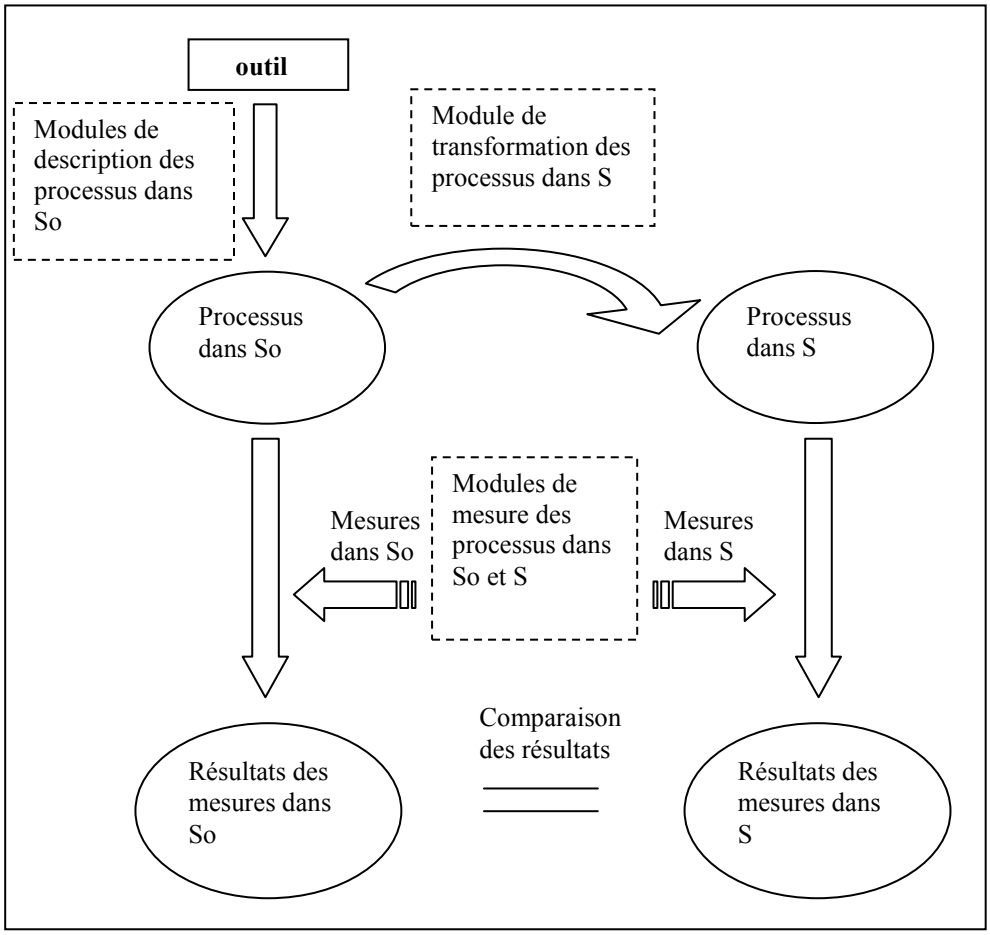


Figure 4-1 : diagramme bloc de l'outil et de ces modules

Dans la suite du manuscrit, nous allons expérimenter des univers régis par des lois d'interaction vérifiant la « Contrainte Relativiste des Accélérations », afin de vérifier s'ils engendrent un espace-temps conforme aux transformations de Lorentz. Afin de rendre la discussion moins abstraite, nous allons décrire les nouveaux modules en utilisant ces expérimentations comme exemples d'illustration.

4.2 Etude de l'Emergence d'un Espace - Temps Relativiste

La physique moderne a recours à deux théories pour décrire notre univers. La relativité qui décrit la structure de l'espace-temps, et la mécanique quantique qui décrit le comportement des particules élémentaires. Selon la relativité restreinte, l'espace-temps de notre univers a une structure décrite par les transformations de Lorentz. La conséquence de cette structure est que la longueur d'un objet en mouvement se contracte par rapport à un observateur immobile et la cadence d'un processus en mouvement se dilate par rapport à l'observateur immobile.

Selon la vision proposée dans [64], la contraction des longueurs et la dilatation du temps est la conséquence de la forme particulière des lois d'interaction des particules. Si l'intensité des interactions entre particules change quand les particules sont en mouvement, les distances auxquelles les particules s'équilibrent formant un objet changent quand l'objet est en mouvement. Cette situation a pour effet la modification des dimensions de l'objet.

De la même façon la cadence de l'évolution d'un ensemble de particules change quand cet ensemble est en mouvement, modifiant ainsi les cadences de l'évolution des processus.

Les transformations de Lorentz sont alors la conséquence de ces modifications ainsi que des résultats de mesures des dimensions des objets, des distances entre objets et des durées des processus qui en résultent. En particulier, il est montré dans [64] que ces mesures sont conformes aux transformations de Lorentz si et seulement si les accélérations induites aux particules par leurs interactions vérifient une condition particulière appelé « Contrainte Relativiste des Accélérations ». Nous pouvons décrire cette condition de la façon suivante :

Considérons un vecteur de vitesse $\bar{v}$ et 3 axes cartésiens X, Y, Z, tels que X est parallèle à $\bar{v}$. Nous appelons S_0 le système de référence inertiel attaché aux axes X,

Y et Z. Considérons une interaction entre deux particules 1 et 2, dont les variables de position et de vitesse sont $\vec{P_1}, \vec{P_2}$ et $\overline{u_1}, \overline{u_2}$. Soit $\overline{u_1} = (u_{1x}, u_{1y}, u_{1z})$ et $\overline{u_2} = (u_{2x}, u_{2y}, u_{2z})$, l'expression de $\overline{u_1}$ et $\overline{u_2}$ selon les axes X, Y, Z, et $\vec{r}$ = (r_x, r_y, r_z) l'expression de la distance entre $\vec{P_1}$ et $\vec{P_2}$ selon les mêmes axes.

Soit $\overline{a_1}$ et $\overline{a_2}$ les accélérations induites aux particules 1 et 2 par cette interaction. Considérons la même interaction entre deux particules 1' et 2', identiques aux particules 1 et 2, mais dans le cas présent leur distance est égale à (r_x/γ_v, r_y, r_z), où

$$\gamma_v = \frac{1}{\sqrt{1 - v^2/c^2}} \qquad \text{(eq2)}$$

et leurs vitesses respectives sont égales à :

$$\overline{u_1'} = \left(\frac{u_{1x} + v}{1 + v \cdot u_{1x}/c^2}, \frac{u_{1y}/\gamma_v}{1 + v \cdot u_{1x}/c^2}, \frac{u_{1z}/\gamma_v}{1 + v \cdot u_{1x}/c^2} \right) \qquad \text{(eq3)}$$

$$\text{et } \overline{u_2'} = \left(\frac{u_{2x} + v}{1 + v \cdot u_{2x}/c^2}, \frac{u_{2y}/\gamma_v}{1 + v \cdot u_{2x}/c^2}, \frac{u_{2z}/\gamma_v}{1 + v \cdot u_{2x}/c^2} \right) \qquad \text{(eq4)}$$

où c est une constante.

Alors, les composantse, selon les axes X, Y et Z, de l'accélération induite sur particule 1' seront cette fois :

$$a_{1x}' = \frac{a_{1x}}{\gamma_v^3 (1 + v \cdot u_{1x}/c^2)^3} \qquad \text{(eq5)}$$

$$a_{1y}' = \frac{a_{1y}}{\gamma_v^2 (1 + v \cdot u_{1x}/c^2)^2} - \frac{(v \cdot u_{1y}/c^2) a_{1x}}{\gamma_v^2 (1 + v \cdot u_{1x}/c^2)^3}, \qquad \text{(eq6)}$$

$$a_{1z}' = \frac{a_{1z}}{\gamma_v^2(1+v\cdot u_{1x}/c^2)^2} - \frac{(v\cdot u_{1z}/c^2)a_{1x}}{\gamma_v^2(1+v\cdot u_{1x}/c^2)^3} \qquad \text{(eq7)}$$

L'expression de l'accélération induite sur la particule 2' est similaire.

Cette condition se généralise en considérant l'interaction de la particule 1 avec plusieurs particules 2, 3, , k de la façon suivante :

Soit $\vec{r_i}$ = (r_{ix}, r_{iy}, r_{iz}), i ∈ {2, 3, …, k} la distance entre les particules 1 et i, $\vec{u_i}$ = (u_{ix}, u_{iy}, u_{iz}), i ∈ {1, 2, …, k} la vitesse de la particule i, et $\vec{a_1}$ =(a_{ix}, a_{iy}, a_{iz}) l'accélération totale induite sur la particule 1 par ses interactions avec les particules 2, 3, .., k.

Alors, si pour k particules 1', 2', , k' identiques aux premières, la distance entre la particule 1' et la particule i', i' ∈ {2', 3', …, k'} est $\overline{r_i'}$ = (r_{ix}/γ_v, r_{iy}, r_{iz}) et la vitesse de la particule i', i' ∈ {1', 2', …, k'} est donnée par l'équation :

$$\overline{u_i'} = \left(\frac{u_{ix}+v}{1+v\cdot u_{ix}/c^2}, \frac{u_{iy}/\gamma_v}{1+v\cdot u_{ix}/c^2}, \frac{u_{iz}/\gamma_v}{1+v\cdot u_{ix}/c^2} \right) \qquad \text{(eq8)}$$

les composants de l'accélération totale induite sur la particule 1' par ses interactions avec les particules 2', 3', , k' seront :

$$a_{1x}' = \frac{a_{1x}}{\gamma_v^3(1+v\cdot u_{1x}/c^2)^3}, \qquad \text{(eq9)}$$

$$a_{1y}' = \frac{a_{1y}}{\gamma_v^2(1+v\cdot u_{1x}/c^2)^2} - \frac{(v\cdot u_{1y}/c^2)a_{1x}}{\gamma_v^2(1+v\cdot u_{1x}/c^2)^3}, \qquad \text{(eq10)}$$

$$a_{1z}' = \frac{a_{1z}}{\gamma_v^2 (1 + v \cdot u_{1x}/c^2)^2} - \frac{(v \cdot u_{1z}/c^2) a_{1x}}{\gamma_v^2 (1 + v \cdot u_{1x}/c^2)^3} . \qquad \text{(eq11)}$$

Cette condition définit une famille de lois d'interaction, mais pour effectuer des expérimentations nous avons besoin d'une expression exacte des lois. Comme exemple de lois nous allons considérer le cas où l'accélération induite à une particule 1 par une particule 2 est proportionnelle à r^{-k}, où r est la distance entre les deux particules. Pour k = 2 l'accélération est proportionnelle à r^{-2}, comme dans le cas des lois des interactions familières entre deux charges électriques (interaction coulombienne) ou entre deux corps massifs (gravitation). Dans nos expérimentations, nous allons considérer le cas où l'accélération dépend des positions présentes des particules (pas d'effet de retard). La valeur de l'accélération dépendra aussi des vitesses des particules. En considérant que ces lois doivent être conformes à la « Contrainte Relativiste des Accélérations » nous obtenons l'expression suivante [64] :

$$a_u = \frac{1}{\gamma_u^3 m_2} \cdot \frac{\gamma_v \cdot g_1 \cdot g_2}{\left(\gamma_v^2 a^2 + b^2\right)^{(k+1)/2}} \cdot \frac{a \cdot u_a + b \cdot u_b}{u} \qquad \text{(C.2)}$$

$$a_{\perp u}^2 = \left(\frac{1}{\gamma_u m_2} \cdot \frac{\gamma_v g_1 g_2}{(\gamma_v a^2 + b^2)^{(k+1)/2}} \right)^2 \cdot \left[\begin{array}{l} \dfrac{\left(b \cdot u_a - a \cdot u_b - b \cdot v \cdot u^2/c^2\right)^2}{u^2} + \dfrac{u_c^2}{u^2}(a^2 + b^2) + \\ \dfrac{b^2 \cdot u^2}{c^4}(u_a^2 + u_b^2) + \dfrac{2b \cdot v}{c^2}\left(a \cdot u_b - b \cdot u_a\right) \end{array} \right] \qquad \text{(C.3)}$$

où $\gamma_v = \frac{1}{\sqrt{1 - v^2/c^2}}$, $\gamma_u = \frac{1}{\sqrt{1 - u^2/c^2}}$, g1 et g2 sont des constantes liées au type des particules et au type des interactions, m_2 est la masse au repos de la particule 2, $\bar{v} = (v_x, v_y, v_z)$ et $\bar{u} = (u_x, u_y, u_z)$ les vitesses des particules 1 et 2. u_a représente la projection de $\bar{u}$ sur l'axe A qui est parallèle à $\bar{v}$, u_b est la projection de

$\bar{u}$ sur l'axe B qui est perpendiculaire à $\bar{v}$ et se trouve sur le plan défini par $\bar{v}$ et $\bar{r}$, et u_c est la projection de $\bar{u}$ sur l'axe C, perpendiculaire aux axes A et B. D'autre part, a et b et c sont les projections de la distance $\bar{r}$ des deux particules sur les axes A, B et C. Finalement, a_u et $a_{\perp u}$ sont les projections de l'accélération de la particule 2 sur un axe parallèle à $\bar{u}$ et sur un axe qui est perpendiculaire à $\bar{u}$ et qui se trouve dans le plan de $\bar{r}$ et $\bar{u}$.

Dans le cas où $\bar{v}, \bar{u}$ et $\bar{r}$ sont sur le même plan, l'expression pour $a_{\perp u}$ devient plus simple :

$$a_{\perp u} = \frac{1}{\gamma_u \cdot m_2} \frac{\gamma_v \cdot g_1 \cdot g_2}{(\gamma_v{}^2 \mathrm{a}^2 + \mathrm{b}^2)^{(k+1)/2}} \cdot \frac{\mathrm{b} \cdot u_a - \mathrm{a} \cdot u_b - \mathrm{b} \cdot v \cdot u^2 / c^2}{u} \qquad \text{(C.3b)}$$

Notons aussi que dans le cas où $\bar{v} = 0$, nous pouvons utiliser comme axes A et B tous axes perpendiculaires entre eux, se trouvant sur le plan défini par $\bar{u}$ et $\bar{r}$. D'autre part, dans le cas où $\bar{u} = 0$, nous trouvons l'expression simplifiée $\bar{a} = \frac{1}{m_2} \frac{\gamma_v g_1 g_2}{(\gamma_v{}^2 a^2 + b^2 + c^2)^{(k+1)/2}} \bar{r}$, où a, b, c sont les coordonnées de $\bar{r}$ sur trois axes perpendiculaires A, B et C du système de référence inertiel S_0 (le système de référence attaché aux axes cartésiens X, Y, Z), dont A est l'axe parallèle à $\bar{v}$ et B et C sont des axes perpendiculaires à A et entre eux, que nous pouvons choisir arbitrairement.

Nous rappelons que dans les simulations, tous les calculs des positions, vitesses et accélérations des particules se font par rapport aux trois axes X, Y et Z fixes d'un système de référence inertiel S_0. Mais, si nous exprimons les différents variables par rapport à ces axes nous obtenons des expressions très complexes qui rendent les lois illisibles. Pour cette raison, les différentes variables sont exprimées ci-dessus par rapport à des axes variables du S_0 (les axes A, B, C ainsi que les axes

parallèles et perpendiculaires à la vitesse $\bar{u}$) dans lesquelles les expressions sont plus simples. Néanmoins, nous allons implémenter dans l'outil les transformations de changement des coordonnées cartésiennes classiques afin de pouvoir exprimer toutes les variables par rapport aux axes X, Y et Z. Ces expressions sont ensuite utilisées par l'outil pendant la phase de simulation. Plus précisément, nous remplaçons u_a, u_b, u_c, a, b et c par des expressions utilisant les projections de $\bar{r}$ et $\bar{u}$ sur X, Y et Z. Nous obtenons ces expressions en utilisant les transformations habituelles permettant de passer de trois axes A, B et C à trois axes X, Y, Z. Ces transformations sont possibles, car les vecteurs $\bar{r}$ = (x_2-x_1, y_2-y_1, z_2-z_1), $\bar{v} = (v_x, v_y, v_z)$, et $\bar{u} = (u_x, u_y, u_z)$ que nous avons utilisés pour définir les axes A, B, et C sont exprimés par rapport aux axes X, Y et Z.

Ensuite, nous utilisons les transformations qui permettent d'obtenir les coordonnées de l'accélération $\bar{a}$ par rapport aux axes X, Y et Z, à partir des coordonnées a_u et $a_{\perp u}$ de cette accélération, qui sont exprimées par rapport aux axes u et $\perp$ u. Cette transformation est aussi possible, car nous connaissons les coordonnées par rapport à X, Y et Z des vecteurs $\bar{r}$ et $\bar{u}$ qui définissent l'orientation des axes u et $\perp$ u.

Le cas du photon : Nous pouvons aussi considérer un type particulier de particules, les photons. Ces particules ont les particularités suivantes :

- la norme de la vitesse du photon (dans le système de référence inertiel S_0) est égale à c (c étant la constante qui apparaît dans la « Contrainte Relativiste des Accélérations »).
- le rapport $g_2/(\gamma_u m_2)$ dans les relations C.2 et C.3 est égal à 1

Ces particularités impliquent que les interactions ne modifient pas la norme de la vitesse du photon. En effet, si la vitesse du photon est à un moment donné égale à

c, dans l'expression C.2 nous aurons u = c, ce qui donne $1/\gamma_u = 0$. Dans C.2 en posant $g_2/(\gamma_u m_2) = 1$ et $1/\gamma_u = 0$, nous trouvons $a_u=0$. Ainsi, il ne reste que le terme $a_{\perp u}$ de l'accélération (expression C.3), qui est perpendiculaire à la vitesse du photon. Dans ce cas, l'interaction va modifier la direction de la vitesse du photon, mais pas sa norme, qui restera ainsi dans le système de référence S_0 toujours égale à c. La constance de la vitesse du photon que nous venons de constater n'est valable que dans le système de référence S_0. Dans les autres systèmes de référence inertiels, la valeur de sa vitesse dépendra alors des étalons et procédures des mesures. Il dépendra donc des lois d'interaction des autres particules qui déterminent les rapports des valeurs des étalons de mesures utilisés dans chaque système de référence inertiel par rapport aux valeurs des étalons utilisés dans le système de référence S_0. Mais nous verrons que pour des lois vérifiant la « Contrainte Relativiste des Accélérations », la vitesse des photons sera la même dans tous les systèmes de référence inertiels.

Notons que l'égalité $g_2/(\gamma_u m_2) = 1$ correspondra à une interaction du type « gravitationnelle » entre le photon et une particule quelconque, dans laquelle $g_2/(\gamma_u m_2) = 1$ serait le rapport (masse gravitationnelle du photon)/(masse inertielle du photon).

Ces lois déterminent comment évoluent les valeurs des variables d'état (la position et la vitesse) des entités élémentaires du système qui sont dans le cas présent les particules. Il suffit alors de les introduire dans l'outil ainsi que le nombre des particules, leur type (valeurs de k, de m, de g1 et de g2) et leur état initial (positions et vitesses). L'outil va ensuite gérer la simulation de ce système. Ainsi nous n'aurons pas besoin de développer de logiciel spécifique pour cette simulation. Néanmoins, nous avons besoin de développer des modules permettant à l'utilisateur de réaliser les étalons de longueur et du temps (horloges) dans les différents systèmes de référence inertiels, synchroniser les horloges distantes dans les différents systèmes de référence, effectuer des mesures et visualiser le système selon le point de vue des observateurs se trouvant dans des systèmes de référence

inertiels quelconques, sachant que l'état du système obtenue par les calculs correspond au point de vue de l'observateur se trouvant dans le système de référence S_0.

Notons finalement que nos expérimentations se feront pour le cas bidimensionnel. La première raison de ce choix est la possibilité d'observer sur écran le système bidimensionnel. D'autre part, les transformations de Lorentz étant identiques sur les deux axes perpendiculaires à la vitesse relative de deux systèmes de référence inertiels, tous leurs effets sont reproductibles en deux dimensions.

4.2.1 Module de création des objets rigides et des étalons de longueur

À l'aide de ce module, l'utilisateur peut former des objets rigides au repos dans le système de référence S_0, constitués d'un ensemble de particules en équilibre. Nous pouvons sélectionner des particules qui obéissent à plusieurs interactions. Par exemple une interaction induisant une accélération égale à $\frac{p_1 p_2}{mr^t}$ et une autre induisant une accélération égale à $\frac{q_1 q_2}{mr^s}$, quand les deux particules en interaction sont au repos. Bien entendu, quand les particules sont en mouvement, les accélérations induites par ces interactions sont données par les expressions C.2, C.3 en remplaçant k par t et g1, g2 par p1, p2 pour la première interaction et k par s et g1, g2 par q1, q2 pour la deuxième interaction. Pour nos expérimentations nous avons choisit t=2 et s=3. Les particules composant un objet rigide seront équilibrées dans des positions dans les quelles la somme vectorielle de ces accélérations devient nulle.

Si nous avons un ensemble de particules subissant des interactions suivant les équations C.2 et C.3, et dont les positions et/ou vitesses initiales sont arbitraires, ils peuvent osciller autour de leur position d'équilibre sans se stabiliser afin de former des objets à dimensions stables. Pour faciliter la formation des tels objets, le module

de création des objets rigides et des étalons de longueur peut, à la demande de l'utilisateur, introduire une accélération du type « frottement », qui a une direction opposée par rapport à la vitesse instantanée de la particule et s'annule quand la vitesse de la particule devient elle même nulle. Nous pouvons par exemple utiliser une accélération proportionnelle à la valeur de la vitesse ou de son carré. Puisque cette accélération devient nulle quand la vitesse de la particule devient nulle, cette accélération, introduite de façon « artificielle » par le module, n'influence pas les positions d'équilibre des particules. Ainsi, les dimensions des objets rigides obtenus de cette façon sont déterminées uniquement par les accélérations induites sur les particules par leurs interactions.

Il est aussi utile de créer des objets rigides dans un système de référence inertiel S ayant une vitesse $\bar{v}$ quelconque par rapport à S_0, pour comparer leurs dimensions. Pour cela, le module utilise une accélération « artificielle » appelée « frottement dans le système de référence S » dont la valeur est proportionnelle à la somme vectorielle $\bar{u}$ - $\bar{v}$ ou au carré de la norme de cette somme, selon le choix de l'utilisateur. Cette accélération devient nulle quand $\bar{u} = \bar{v}$. Dans ce cas, les particules formant l'objet rigide s'équilibrent dans des positions dans lesquelles la somme vectorielle des accélérations induites par leurs interactions deviennent nulles et dans cette équilibre la vitesse de chaque particule est égale à $\bar{v}$. Ainsi, l'objet rigide obtenu de cette façon se trouve au repos dans S et ses dimensions sont déterminées uniquement par les accélérations induites sur les particules par leurs interactions.

En employant ce module l'utilisateur peut aussi créer le même objet rigide au repos dans le système de référence S_0 et dans un système de référence inertiel S ayant une vitesse $\bar{v}$ quelconque par rapport à S_0. Ces objets sont utiles pour déterminer l'impact de changement de système de référence inertiel sur les dimensions des objets.

L'utilisateur peut choisir un objet rigide constitué d'un ensemble de particules en équilibre comme étalon de longueur. Il peut utiliser le module pour réaliser cet étalon dans le système de référence S_0 et dans tout système de référence inertiel S ayant une vitesse $\bar{v}$ quelconque par rapport à S_0. Par simplicité, dans nos expérimentations nous avons choisi comme étalon de longueur un objet très simple, composé de deux particules en équilibre sous l'effet de leurs interactions.

4.2.2 Module de mesures des longueurs depuis le système de référence S_0

Afin de faciliter nos expériences de mesures, nous avons développé un module capable à la demande de l'utilisateur, de mesurer dans le système de référence S_0 les longueurs des objets se trouvant au repos dans S_0 et des objets se trouvant au repos dans des systèmes de référence inertiels S ayant une vitesse $\bar{v}$ quelconque par rapport à S_0. Ce module compare les dimensions de l'objet mesuré dans les directions de deux axes perpendiculaires X et Y, avec l'étalon de longueur du système de référence S_0. Quand l'objet mesuré se trouve au repos dans S_0, les axes X et Y peuvent être choisis de façon arbitraire. Quand l'objet à mesurer se trouve au repos dans un système de référence S qui se déplace avec une vitesse $\bar{v}$ par rapport à S_0, l'axe X est parallèle à $\bar{v}$ et l'axe Y est perpendiculaire à $\bar{v}$.

L'utilisateur utilise le module décrit dans la section 4.2.1 pour créer l'étalon de longueur du système de référence S_0. Il utilise aussi le même module pour créer des objets rigides dans S_0 et dans n'importe quel système de référence inertiel S. Il fait ensuite une requête au module de mesure des longueurs dans S_0, qui lui retourne les dimensions de objets mesurées dans S_0.

La Figure 4-2 présente une copie d'écran de différents objets identiques, les uns au repos dans S_0 et les autres dans un système de référence inertiel S. Ces objets sont, l'étalon de longueur (composée par deux particules) placé une fois selon l'axe parallèle à $\bar{v}$ (axe X) et une fois selon l'axe perpendiculaire à $\bar{v}$ (axe Y), un objet

triangulaire (composé par trois particules) et un objet de forme quelconque composé par cinq particules.

a)

b)

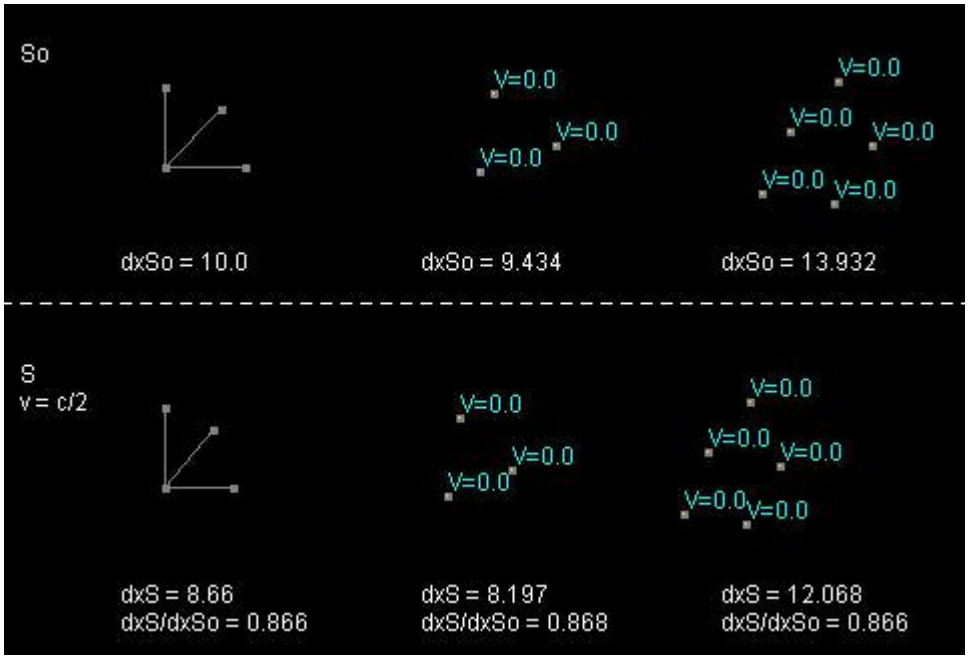


Figure 4-2 : les mêmes objets rigides subissent une contraction de longueur de 0.866 sur l'axe X dans un système de référence inertiel S se déplaçant avec une vitesse égale à la moitié de la constante c (vitesse maximale ou vitesse de la lumière) sur l'axe X. Ils vérifient la relation de contraction des longueurs $l' = l//\gamma_v$

Les mesures selon l'axe Y donnent toujours les mêmes résultats pour les objets dans les deux systèmes de référence inertiels. D'autre part, les mesures selon l'axe X donnent pour objets dans les deux systèmes de référence inertiels des résultats qui vérifient toujours la relation de contraction des longueurs $l' = l/\gamma_v$. Cette relation est toujours vérifiée dans des simulations faites pour tout sorte d'objets et pour toute valeur de la vitesse v, confirmant le fait que les lois d'interactions utilisées produisent des objets qui vérifient pour l'observateur du système de référence S_0 la relation de la contraction des longueurs de la relativité restreinte.

4.2.3 Module de mesure depuis le système de référence S des dimensions des objets se trouvant au repos dans S

Nous avons développé un module capable à la demande de l'utilisateur de mesurer dans des systèmes de référence inertiels S ayant une vitesse $\bar{v}$ quelconque par rapport à S_0, les longueurs des objets se trouvant au repos dans S. Ce module compare avec l'étalon de longueur du système de référence S les dimensions de l'objet sous mesure dans la direction de l'axe X parallèle à $\bar{v}$ et de l'axe Y perpendiculaire à $\bar{v}$.

L'utilisateur utilise le module décrit dans la section 4.2.1 pour créer l'étalon de longueur du système de référence S. Il utilise aussi le même module pour créer des objets rigides dans le système de référence inertiel S. Il fait ensuite une requête au module de mesure des longueurs dans S, qui lui retourne les dimensions des objets au repos dans S et mesurées dans S.

La figure 4-3 présente une copie d'écran montrant différents objets au repos dans un système de référence inertiel S. Ces objets sont, l'étalon de longueur de S placé une foi selon l'axe parallèle à $\bar{v}$ (axe X) et une fois selon l'axe perpendiculaire à $\bar{v}$ (axe Y), un objet triangulaire (composé par trois particules), et un objet de forme quelconque composé par une pluralité de particules. Ces objets sont identiques aux objets des Figures 4-2a et 4-2b.

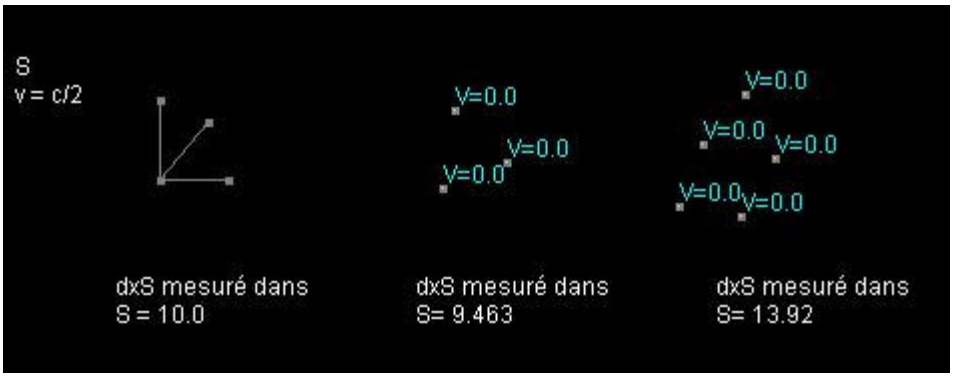


Figure 4-3 : différents objets au repos dans un système de référence inertiel S et mesurés par l'étalon de longueur de S

Toutes les mesures (selon l'axe X et l'axe Y) donnent les mêmes résultats que les mesures des mêmes objets se trouvant au repos dans S_0 et mesurés dans S_0 (figure 4-2a). L'explication de ce résultat est simple. Dans la direction de l'axe Y, les dimensions de l'étalon de longueur et des objets au repos dans S sont identiques aux longueurs de l'étalon de longueur et des objets au repos dans S_0. Dans la direction de l'axe X, la dimension de l'étalon de longueur est réduite d'un facteur égal à $1/\gamma_v$ par rapport à la dimension de l'étalon de longueur au repos dans S_0. Les longueurs au repos dans S sont aussi réduites par le même facteur. Ainsi, les comparaisons des longueurs des objets au repos dans S avec l'étalon de longueur au repos dans S donnent dans l'axe X aussi les mêmes résultats avec les comparaisons effectuées dans le système de référence S_0.

Dans la section précédente nous avons trouvé que la longueur dans la direction de l'axe X, d'un objet au repos dans S et mesurée dans le système de référence S_0 est réduite d'un facteur $1/\gamma_v$ par rapport à la longueur du même objet se trouvant au repos dans S_0 et mesuré dans S_0. Par conséquent, la longueur dans la direction de l'axe X, d'un objet au repos dans S et mesurée dans le système de référence S_0 est réduite d'un facteur $1/\gamma_v$ par rapport à la longueur du même objet toujours au repos

dans S mais mesuré cette fois dans S. Cette relation exprime la contraction des longueurs constatée dans la relativité restreinte.

4.2.4 Module de création des horloges

L'utilisateur, peut utiliser une configuration de particules qui donnent lieu à un processus périodique comme horloge. Pour nos expérimentations nous avons choisi deux telles conformations.

4.2.4.1 Première horloge

Dans tout système de référence inertiel, la première conformation consiste en un photon effectuant des aller-retour dans un tube se trouvant au repos dans ce système de référence, et dont la longueur mesurée dans le même système de référence est égale à l. Le tube peut être orienté selon n'importe quelle direction.

Pour créer l'horloge dans un système inertiel S ayant une vitesse $\bar{v}$, par rapport à S_0, nous avons besoin de connaître les composantes de la longueur du tube dans S_0 ainsi que la direction de la vitesse du photon dans S_0. Puisque la longueur du tube au repos dans le système de référence S, mesurée dans ce même système de référence, est égale à l, alors, selon les résultats des mesures obtenus dans les sections précédentes, la longueur du même tube mesurée par l'observateur de S_0, sera égale à l quand son orientation est perpendiculaire à $\bar{v}$,et égale à l/γ_v quand son orientation est parallèle à $\bar{v}$. Pour des orientations intermédiaires, les composants de la longueur du tube mesurées par l'observateur de S_0 auront la forme $l_x = \frac{\sqrt{l^2 - a^2}}{\gamma_v}$, $l_y = a$. L'orientation du tube dans S est donnée par le rapport a/l. Si nous avons défini cette orientation nous connaissons la valeur de a, et ainsi nous pouvons calculer les valeurs des l_x et l_y.

La vitesse du photon vue par le système de référence S_0, quand le tube est parallèle à $\bar{v}$ sera aussi parallèle à $\bar{v}$. Quand le tube est perpendiculaire à $\bar{v}$ cette vitesse aura une composante égale à $\bar{v}$, et en une composante perpendiculaire à $\bar{v}$ et de valeur $\sqrt{c^2 - v^2}$ (la norme de la vitesse du photon dans S_0 reste toujours c).

En utilisant ces relations, nous avons implémenté un module, qui, à la demande de l'utilisateur, génère des horloges de ce type dans tout système de référence inertiel choisi par l'utilisateur et ayant une orientation (a/l) définie par l'utilisateur.

Nous employons ce module pour créer dans un système de référence inertiel S plusieurs horloges dont le tube a des orientations diverses, et dont une extrémité se trouve dans la même position. Dans la Figure 4-4a nous montrons une copie d'écran de la simulation de ces horloges à l'instant du départ des photons de cette extrémité. La Figure 4-4b montre une copie d'écran correspondant à l'arrivée du photon de la première horloge dans la deuxième extrémité. La deuxième horloge est en décalage par rapport à cette horloge. La Figure 4-4c montre une copie d'écran correspondant au retour des photons dans la première extrémité.



Figure 4-4a : Les horloges à l'instant du départ des photons



Figure 4-4b : Les horloges à l'instant de l'arrivée du photon de la première horloge à l'autre extrémité



Figure 4-4c : Les horloges à l'instant de l'arrivée des photons à la première extrémité

Nous observons que, l'instant du retour est identique pour toutes les horloges. Ainsi, si nous utilisons la durée d'un aller-retour comme étalon du temps, cet étalon sera identique pour toutes les orientations du tube. Dans la Figure 4-5a, nous montrons aussi l'étalon de longueur composé par deux particules, comme défini précédemment. Cet étalon est placé au repos dans S et suivant les mêmes orientations que les tubes des horloges. Nous observons que la longueur l du tube est toujours 5 fois la longueur de l'étalon de longueur, et ceci pour toutes les orientations. L'observateur se trouvant dans S constatera alors que la vitesse des photons est la même dans toutes les directions, puisque les photons parcourent la même distance en même temps. La Figure 4-5b montre l'horloge et l'étalon placés au repos dans S_0 et dans diverses orientations. Nous observons dans ce cas aussi que la longueur du tube vaut toujours 5 fois la longueur de l'étalon. Ainsi, si un

observateur se place dans S_0 et ensuite se place dans S, il constatera que les horloges utilisées dans les deux systèmes de référence sont identiques.

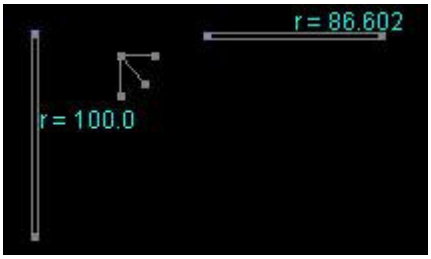


Figure 4-5a : l'horloge et l'étalon placés au repos dans S

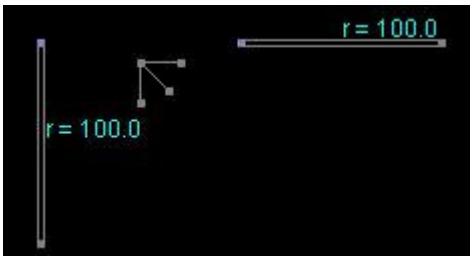


Figure 4-5b : l'horloge et l'étalon placés au repos dans S_0

4.2.4.2 Deuxième horloge

La deuxième conformation de particules utilisée comme horloge, consiste en deux particules interagissant suivant une loi qui induit au repos une accélération égale à $\frac{g_1 g_2}{mr^2}$. Bien entendu, quand les particules sont en mouvement, l'accélération est donnée par les expressions C.2 et C.3, dans lesquelles nous remplaçons k par 2. La première particule est placée au repos dans le système de référence inertiel dans lequel nous construisons l'horloge. La deuxième particule, vue par l'observateur de ce système de référence, exécute un mouvement circulaire dans une distance r_0 autour de la première.

Pour construire l'horloge dans S_0, nous plaçons la première particule au repos dans S_0. La deuxième particule est placée à la verticale de la première, à une distance r_0, et sa vitesse initiale est horizontale et égale à :

$$w = \sqrt{\sqrt{\frac{q_1^4 q_2^4}{4m_2^4 \cdot r_0^4 \cdot c^4} + \frac{q_1^2 q_2^2}{m_2^2 \cdot r_0^2}} - \frac{q_1^2 q_2^2}{2m_2^2 \cdot r_0^2 \cdot c^2}} \qquad \text{(eq.12)}$$

Nous associons à la première particule une masse très élevée par rapport à l'intensité de l'interaction des deux particules, de façon à ce que cette particule reste au repos dans S_0 malgré l'effet de cette interaction. La simulation de cette configuration montre que la deuxième particule effectue un mouvement circulaire autour de la première, dans une distance égale à r_0, ceci est illustré par la Figure 4-6a, dans laquelle, nous avons placé l'étalon de longueur r_0 dans diverse direction et ayant toujours sa première extrémité au centre de l'horloge (position de la première particule. La ligne en pointillé montre que le parcours de la deuxième particule passe toujours par la deuxième extrémité de l'étalon).

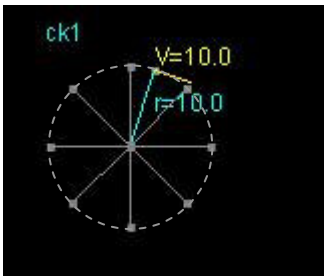


Figure 4-6a : l'horloge et l'étalon placés au repos dans S_0

Pour construire l'horloge dans un système de référence inertiel S se déplaçant par rapport à S_0 avec une vitesse $\bar{v}$ (considérée ici avec une direction horizontale), nous donnons à la première particule une vitesse initial égale à $\bar{v}$, et nous lui donnons comme précédemment une masse élevée pour que sa vitesse reste inchangée. Ainsi la particule reste au repos dans S. Nous plaçons la deuxième

particule à la verticale de la première et à une distance égale à r_0 (vue du S_0 mais aussi du S puisque r_0 est perpendiculaire à $\bar{v}$), et nous lui donnons une vitesse initiale horizontale et égale à $\frac{w+v}{1+w\cdot v/c^2}$. La simulation de cette configuration montre que, vue par l'observateur de S, la deuxième particule effectue un mouvement circulaire autour de la première, dans une distance égale à r_0, ceci est illustré dans la Figure 4-6b, dans laquelle, nous montrons la forme de la trajectoire effectuée par la particule 2 autour de la particule 1.

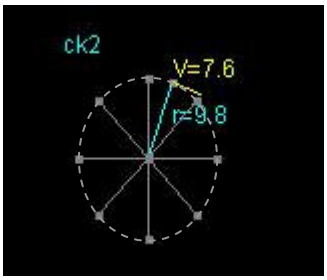


Figure 4-6b : l'horloge et l'étalon placés au repos dans S

Cette figure ne représente pas l'évolution de la position de la particule 2 vue de S_0 mais l'évolution de sa distance de la particule 1 vue de S_0. Nous avons aussi placé l'étalon de longueur r_0 du système de référence S, dans diverses directions, et ayant toujours sa première extrémité au centre de l'horloge (position de la première particule. La ligne en pointillé montre que le parcours de la deuxième particule passe toujours par la deuxième extrémité de l'étalon). Ainsi, pour l'observateur de S, la trajectoire de la deuxième particule passe toujours à une distance r_0 de la première. Un observateur qui se mettra au repos dans S_0 et ensuite dans S, conclura que les horloges de ces systèmes de référence sont identiques.

Notons que dans la suite, à chaque fois que nous montrons la trajectoire d'une particule vue de S_0 autour d'une autre se trouvant au repos dans S, cette trajectoire

ne représentera pas l'évolution de la position de la particule, mais l'évolution de la distance entre les deux particules.

Nous avons utilisé ces résultats pour créer un module qui génère ce type d'horloge dans n'importe quel système de référence inertiel, à la demande de l'utilisateur. L'état initial des particules a été obtenu analytiquement. Par contre, la preuve que dans un système de référence inertiel S la deuxième particule produira un parcours circulaire, sachant que sont accélération est donnée par les expressions C.2 et C.3, est très difficile à obtenir analytiquement. L'approche de simulation est dans ce cas très utile.

L'obtention des états initiaux des particules de façon analytique était possible car nous avons prédit que les expressions C.2 et C.3 produiront un espace-temps conforme aux transformations de Lorentz. Dans le cas où nous avons des lois pour lesquelles nous ne connaissons pas la structure de l'espace temps qu'elles engendrent, l'approche analytique n'est pas applicable. Afin, de contourner ce problème, nous avons aussi donné une option au module qui consiste à déterminer la valeur de la vitesse initiale de la deuxième particule 2 (considérant toujours que sa position est à la verticale de la première particule et leur distance égale à r_0). Pour déterminer la valeur de la vitesse initiale de la deuxième particule, le module lui associe une première valeur et selon la forme de la trajectoire il augmente ou réduit la valeur de la vitesse. En répétant cette étape plusieurs fois la trajectoire converge rapidement à la trajectoire circulaire.

4.2.5 Module de mesures des intervalles du temps depuis le système de référence S_0

Afin de faciliter nos expériences de mesure du temps, nous avons développé un module capable à la demande de l'utilisateur de mesurer dans le système de référence S_0 les durées des processus qui ont lieu dans S_0 ainsi que des processus qui ont lieu dans des systèmes de référence inertiels S ayant une vitesse $\bar{v}$

quelconque par rapport à S_0. Ce module compare la durée d'un processus mesuré avec l'étalon du temps du système de référence S_0. Dans ces mesures, nous utilisons comme horloge le système composé de deux particules dont la deuxième effectue des trajectoires circulaires autour de la première, comme discuté dans la section précédente. L'étalon du temps correspondra à la durée d'une révolution complète de la deuxième particule. Nous avons utilisé le module décrit dans la section précédente pour créer une telle horloge au repos dans S_0.

Nous avons développé un module qui mesure la durée d'un processus quelconque en la comparant avec l'étalon du temps dans S_0. Pour ce faire, ce module compte le nombre des révolutions effectuées par la particule 2 de l'horloge de S_0 pendant la durée du processus en question.

Nous avons créé plusieurs tubes de diverses longueurs et directions, dans lesquels un photon effectue des allers-retours. Pour ce faire, nous avons modifié le module de la section précédente (voir paragraphe 4.2.4.1), de façon à générer des tubes de longueur quelconque. Pour chaque tube créé dans S_0 nous avons créé un tube identique dans S. Nous avons utilisé le module de mesure des durées des processus pour déterminer la durée d'un aller-retour dans chaque tube. La Figure 4-7a est une copie d'écran montrant les tubes créés dans S_0 et le temps d'un aller-retour de leur photon en nombre des révolutions de la particule 2 de l'horloge de S_0. La Figure 4-7b est une copie d'écran montrant les tubes correspondants créées dans S et le temps d'un aller-retour de leur photon en nombre des révolutions de la particule 2 de l'horloge de S_0. Nous trouvons que dans tous les cas, la durée t_0 d'un aller-retour de photon dans un tube au repos dans S_0 et la durée t d'un aller-retour de photon dans le tube au repos dans S sont liées par la relation $t = \gamma_v t_0$. Cette relation correspond à la dilatation du temps de la relativité restreinte.

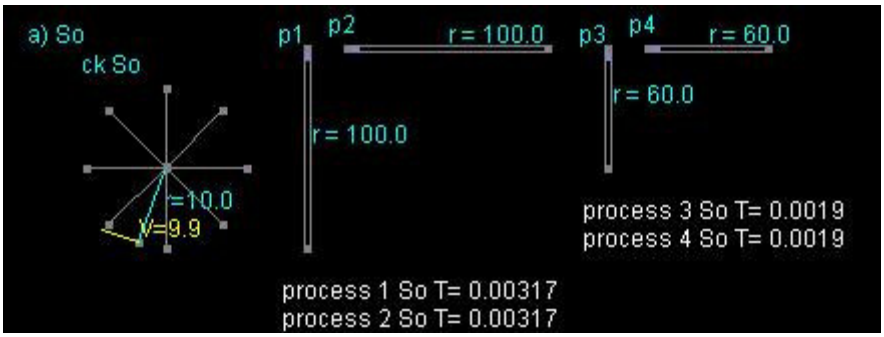


Figure 4-7a : Mesure depuis S_0 du temps des allers-retours des photon dans le référentiel inertiel S_0 pour divers tubes

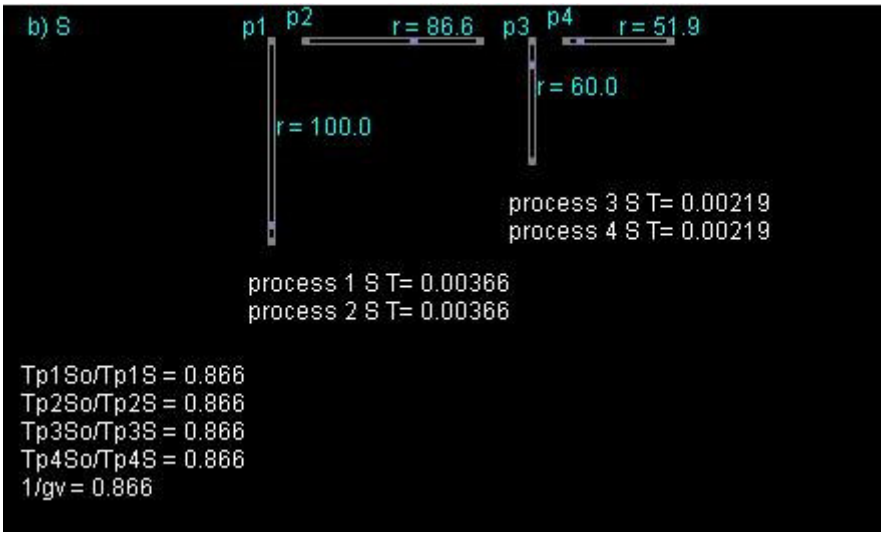


Figure 4-7b : Mesure depuis S_0 du temps des allers-retours des photon dans le référentiel inertiel S pour diverses tubes

Nous avons aussi créé plusieurs systèmes composés d'une paire de particules dont l'un effectue un parcours circulaire autour de l'autre à diverses distances. Pour ce faire, nous avons modifié le module de la section précédente, de façon à rendre paramétrable le rayon de la trajectoire de la deuxième particule. De cette façon nous pouvons choisir la valeur de ce rayon à volonté. Pour chaque système créé dans S_0 nous avons créé un système identique dans S. Nous avons utilisé le module de mesure de durées des processus pour déterminer la durée d'une révolution

complète de la particule 2 de chaque système. La Figure 4-8a est une copie d'écran montrant les systèmes créés dans S_0, ainsi que le temps d'une révolution complète de leur particule 2, en nombre des révolutions de la particule 2 de l'horloge de S_0. La Figure 4-8b est une copie d'écran montrant les systèmes correspondants créées dans S et le temps de la révolution complète de leur particule 2 en nombre des révolutions de la particule 2 de l'horloge de S_0. De nouveau nous trouvons que dans tous les cas, la durée t_0 d'une révolution complète de la particule 2 d'un système au repos dans S_0 et la durée t d'une révolution complète de la particule 2 d'un système au repos dans S, sont liées par la relation de la dilatation du temps de la relativité restreinte ($t = \gamma_v t_0$).

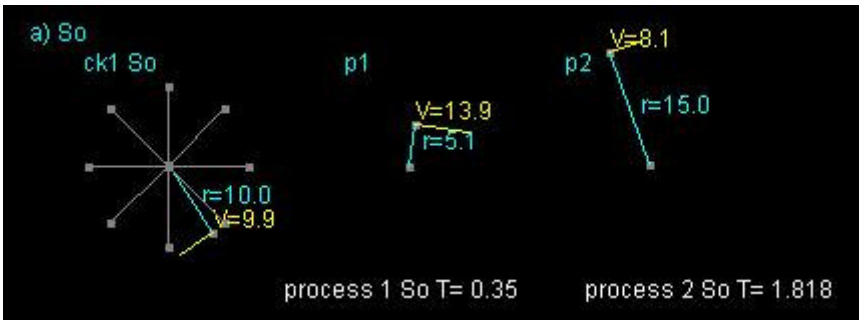


Figure 4-8a : Mesure depuis S_0 du temps du parcours circulaire des particules autour d'autres dans diverses distances dans le référentiel inertiel S_0.

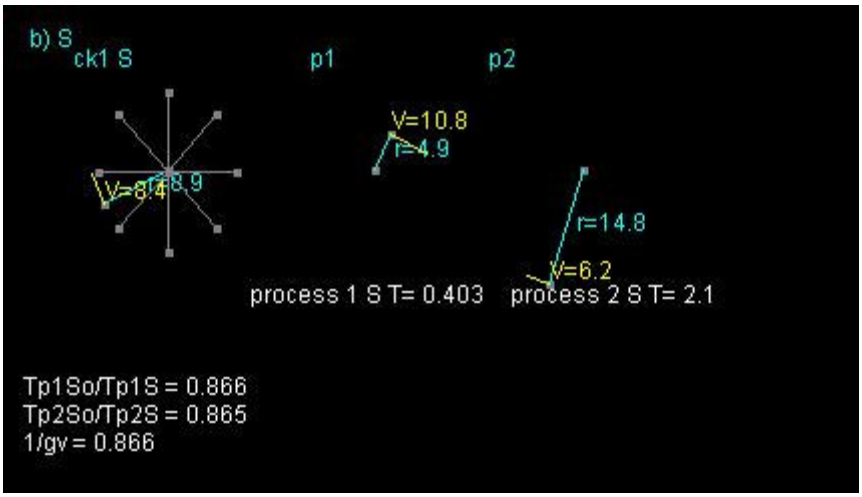


Figure 4-8b : Mesure depuis S_0 du temps du parcours circulaire des particules autour d'autres dans diverses distances dans le référentiel inertiel S

La Figure 4-9a montre d'autres systèmes de particules. Ces systèmes sont composés d'une particule (1) au repos dans S_0 et d'une deuxième particule (2) évoluant autour de la première. La position initiale de la particule 2 est à la verticale de la particule 1. La direction de sa vitesse initiale est horizontale. La valeur de sa vitesse initiale et sa distance initiale de la particule 1 sont choisies de façon arbitraire. Ainsi nous n'imposons pas des trajectoires circulaires et par conséquent des trajectoires elliptiques apparaissent. La Figure 4-9b montre des systèmes de particules identiques à ceux de la Figure 4-9a, mais cette fois il se trouve dans un système de référence inertiel S ayant une vitesse $\bar{v}$ par rapport à S_0. Nous avons développé un module capable de créer ces systèmes. Pour ce faire, ce module place dans la Figure 4-9a plusieurs points sur la trajectoire de la particule 2. Il détermine les coordonnées x et y de ces points en considérant la particule 1 comme origine des coordonnées. Le module utilise la transformation (x, y) → (x/γ_v, y). Les nouvelles coordonnées donnent la distance dans laquelle doit évoluer la particule 2 autour de la particule 1, quand le système des particules se trouve dans le système de référence S. Ensuite, le module crée une paire de particules telle que :

- la particule 1 se déplace avec une vitesse $\bar{v}$ par rapport à S_0,
- la position initiale de la particule 2 est à la verticale de la particule 1
- la distance initiale des particules 1 et 2 est la même que la distance initiale de ces particules dans la Figure 4-9a.
- la direction de la vitesse initiale de la particule 2 est horizontale
- une valeur de vitesse initiale est donnée à la particule 2 et sa trajectoire est analysée pour vérifier si sa distance de la particule 1 est conforme aux distances déterminées précédemment par la transformation (x, y) → (x/γ_v, y).
- selon le résultat de cette analyse, la valeur de la vitesse de la particule 2 est augmentée ou diminuée successivement, jusqu'au moment où sa trajectoire passe par les positions déterminées par la transformation (x, y) → (x/γ_v, y).

Les mesures des durées t_0 d'une révolution complète des particules 2 dans les systèmes de la Figure 4-9a et les mesures des durées t d'une révolution complète des particules 2 dans les systèmes de la figures 4-9b donnent toujours la relation de la dilatation du temps de la relativité restreinte ($t_0 = \gamma_v t$).

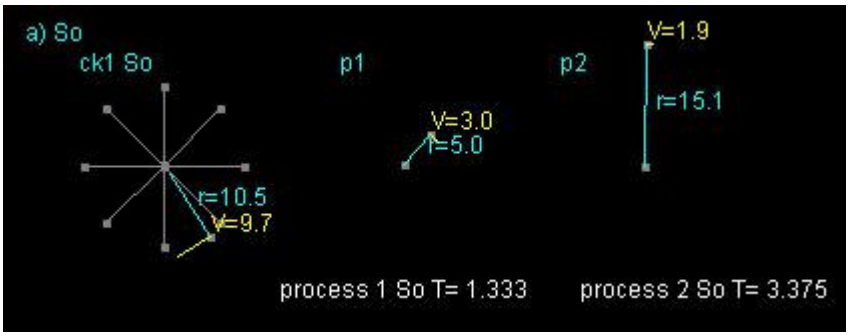


Figure 4-9a : Mesure depuis S_0 du temps du parcours elliptique des particules autour d'autres dans diverses distances dans le référentiel inertiel So

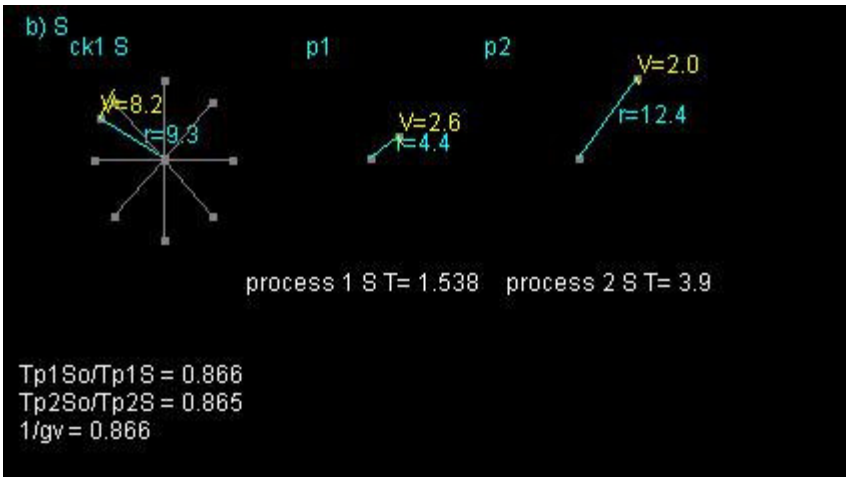


Figure 4-9b : Mesure depuis S_0 du temps du parcours elliptique des particules autour d'autres dans diverses distances dans le référentiel inertiel S

4.2.6 Module de synchronisation des horloges distantes

Pour effectuer divers types de mesures dans un système de référence, nous avons souvent besoin de synchroniser des horloges distantes dans ce système de référence. Ainsi, pour mesurer dans un système de référence inertiel S la longueur d'un objet en mouvement par rapport au système de référence, nous devons déterminer la distance à laquelle se trouvent les extrémités de cet objet à un instant donné par rapport au système de référence S. Par exemple, nous pouvons disposer d'une règle et déterminer la position de ses extrémités par rapport aux graduations de la règle, au même instant du système de référence S. Par conséquent nous devons être capables de synchroniser deux horloges du système de référence S se trouvant sur ces positions. De la même façon, si nous voulons mesurer la vitesse d'un objet par rapport au système de référence S, nous devons placer des horloges synchronisées sur diverses positions de la trajectoire de l'objet et observer leurs indications aux moments où l'objet en mouvement les croise. Nous avons à nouveau besoin de synchroniser des horloges distantes dans le système de référence S.

Dans le système de référence S_0, nous avons la synchronisation des horloges distantes intrinsèque, car, la simulation détermine à chaque pas de calcul un nouveau état pour tous les particules de l'univers et les lois utilisées pour ce calcule sont indépendantes de la valeur de la variable de position des particules. Ainsi, les horloges de S_0 placées dans des positions distantes (valeurs différentes des variables de position des particules qui les composent) vont évoluer de la même façon. Par conséquent, si elles sont initialisées identiquement au début de la simulation (ou à n'importe quel pas de calcul), elles resteront synchronisées à chaque pas du calcul.

Pour synchroniser deux horloges distantes dans un autre système de référence inertiel S, nous devons lancer vers chacune de ces horloges au même instant et depuis le milieu de la distance qui les sépare deux objets qui se déplacent avec la même vitesse par rapport au système de référence S. Puis, au moment où chaque objet croise l'horloge correspondant, positionner cette horloge dans son origine (temps 0). Néanmoins, pour vérifier si les deux objets de synchronisation se déplacent par rapport au système de référence S avec la même vitesse (mais de directions opposées), nous devons préalablement placer des horloges synchronisées tout au long de leur trajectoire. Mais nous ne disposons pas de telles horloges préalablement à cette expérience. Si nous connaissons des objets qui ont une vitesse constante dans toutes les directions d'un système de référence inertiel, nous pouvons les utiliser comme objets de synchronisation. C'est le principe utilisé dans la relativité, où les objets de synchronisation sont des photons. Mais cette approche n'est pas générale et ne s'applique pas dans le cas où nous n'avons pas connaissance de tels objets, et bien sûr dans des univers dans lesquels tels objets n'existent pas. Or, nous voulons implémenter un module de synchronisation générique utilisable aussi dans de tels univers. Pour ce faire, nous avons adopté un principe utilisant comme objets de synchronisation des horloges en mouvement [64].

Soit A et B les positions où se trouvent les deux horloges que nous cherchons à synchroniser. Selon ce principe, la distance qui sépare les positions A et B est partagée par un certain nombre des points dans des intervalles identiques, en

utilisant l'étalon de longueur du système de référence S. Depuis le milieu M de cette distance sont lancées vers les positions A et B deux horloges identiques. Nous observons les indications de ces horloges à chaque fois qu'elles croisent un des ces points. Si leurs indications sont identiques, ces horloges peuvent être utilisées comme objets de synchronisation.

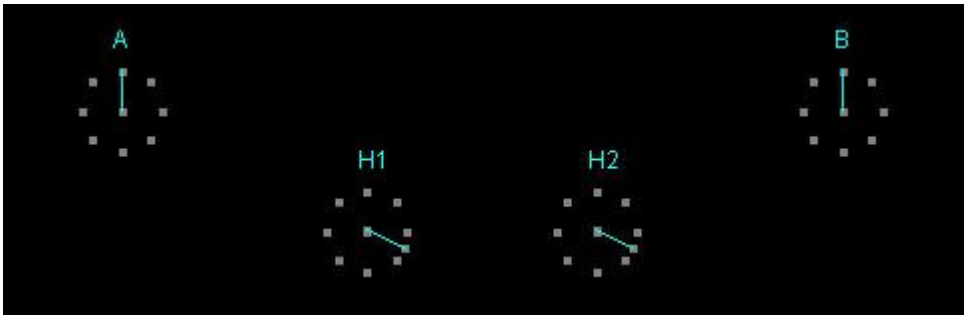
Notre module de synchronisation créer deux horloges H1 et H2 identiques, composées de deux particules dont l'une effectue un mouvement circulaire autour de l'autre. Le module lance au même instant et depuis la position M l'horloge H1 vers la position A et l'horloge H2 vers la position B. Au moment du lancement ces horloges sont placées à leur origine (la particule deux à la verticale de la particule 1). Ensuite, le temps indiqué par chaque horloge est déterminé en comptant le nombre des révolutions de la particule 2 autour de la particule 1. Au début, les vitesses de ces horloges dans le système de référence S_0 sont choisies respectivement égales a u + v et u – v, où u est une valeur quelconque. Les indications des ces horloges sont observées à chaque fois qu'elles croisent les points A et B. Si l'indication de l'horloge H2 est en retard par rapport à l'indication de l'horloge H1, la vitesse de H2 est augmentée, si l'horloge H2 est en avance, alors cette vitesse est diminuée. L'expérience est répétée avec cette nouvelle vitesse de H2. Au bout d'un certain nombre des répétitions les indications des horloges H1 et H2 deviennent égales. Alors, au moment où les horloges H1 et H2 croisent les positions A et B, deux horloges placées sur ces positions sont initialisées à 0. Nous obtenons de cette façon la synchronisation de ces horloges.

Nous avons utilisé ce module pour effectuer la synchronisation des horloges distantes dans divers systèmes de référence inertiels.

Pour le système de référence S_0, nous avons placé à des positions distantes A et B deux horloges placées à leur origine au début de la simulation. Nous avons utilisé des horloges H1 et H2 ayant par rapport à S_0 des vitesses égales à u et à –u. La Figure 4-10 est une copie d'écran qui montre le résultat de cette expérience. Nous observons que les horloges H1 et H2 croisent les positions A et B (ce qui

correspond au même temps de simulation puisque les deux horloges ont la même vitesse et traversent la même distance) avec les mêmes indications. Ce résultat valide ce qui a était dit plutôt à propos de la synchronisation intrinsèque des horloges distantes du système de référence S_0. Par conséquent, dans le système de référence S_0, nous pouvons mesurer le temps soit en comptant le nombre des révolutions des horloges immobiles dans S0, qui sont mises au même pas de calcul de la simulation, soit en comptant le nombre des pas de calcul de la simulation.

a



b



c

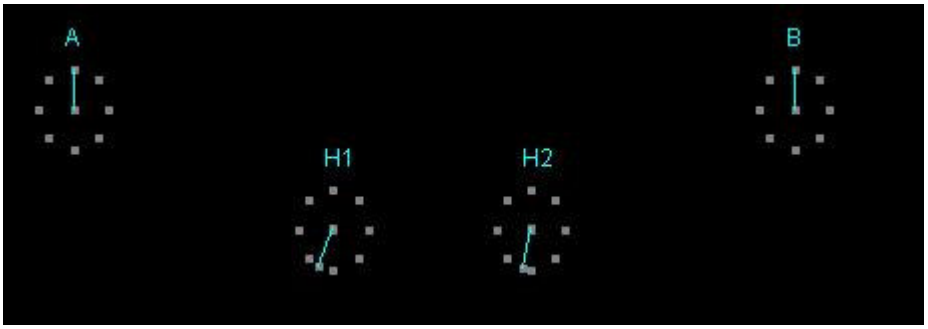


Figure 4-10 : synchronisation des horloges dans S_0

Nous avons aussi effectué l'expérience pour divers systèmes de référence inertiels S en utilisant plusieurs valeurs pour leur vitesse relative v par rapport au système de référence S_0. La Figure 4-11 montre le résultat d'une de ces expériences. La Figure 4-11a est une copie d'écran montrant les deux horloges se déplaçant vers les points A et B. La Figure 4-11b est une copie d'écran montrant l'instant t_1 de l'arrivée de l'horloge H1 sur la position A. Comme cette visualisation est un résultat de simulation, elle montre cet instant tel qu'il est observé depuis le système de référence S_0. Nous observons que à cet instant l'horloge H2 n'est pas encore arrivée sur la position B. La Figure 4-11c montre l'instant t_2 de l'arrivée de l'horloge H2 sur la position B. Par conséquent, l'horloge Ha du système de référence S placé sur la position A est mis à son origine à l'instant t_1 du système de référence S_0, tandis que l'horloge Hb du système de référence S placé sur la position B est mis à son origine à l'instant t_2 du système de référence S_0. Ainsi, les horloges Ha et Hb vues par le système de référence S_0 sont désynchronisées (voir Figure 4-11d). En répétant l'expérience pour plusieurs systèmes de référence inertiels S, nous trouvons que cette désynchronisation vérifie toujours la relation $t_1 - t_2 = \frac{\gamma_v v}{c^2} l$, où v est la vitesse du système de référence inertiel S par rapport au système de référence S_0 et l est la longueur de l'intervalle AB mesurée dans le système de

référence S. Nous retrouvons la relation de désynchronisation en vigueur dans la relativité restreinte.

a



b



c



d



Figure 4-11 : synchronisation des horloges dans S

4.2.7 Module de changement de système de référence d'observation

Les copies d'écran montrées dans les figures précédentes représentent des images de différents objets observées depuis le système de référence inertiel S_0. Pour visualiser les images de ces objets observées depuis un système de référence inertiel S, nous devons :

i. Prendre en compte le déplacement S par rapport à S_0. Ainsi l'origine $x_0' = 0$ de S se trouve au temps t de S_0 à la position $x_0 = vt$ de S0.
ii. Prendre en compte le fait que l'étalon de longueur r_s de S est différent de l'étalon de longueur r_0 de S_0, et multiplier par r_0/r_s la distance, selon l'axe X, entre deux positions x1' et x2' de S mesurée dans S_0, afin de déterminer la valeur de la même distance mesurée dans S.

En appliquant les transformations décrites aux points i et ii dans le cas d'une particule qui se trouve au temps t de S_0 sur la position x de S_0, nous trouvons que la position correspondante dans S est $x' = r_0/r_s(x - vt)$. La valeur de r_0/r_s est déterminée par les mesures effectuées dans la section 4.2.2. Pour le cas des lois d'interactions vérifiant la Contrainte Relativiste des Accélérations, le rapport r_0/r_s a été trouvé égal à γ_v, ce qui donne $x' = \gamma_v(x - vt)$. En plus des conditions i et ii, il faut :

iii. Prendre en compte le fait que l'étalon de temps t_s de S est différent de l'étalon de temps t_0 de S_0, et multiplier par t_0/t_s le temps écoulé dans l'origine de l'axe X de S, mesurée dans S_0, afin de déterminer la durée du même temps mesurée par un horloge de S placé sur l'origine de l'axe X de S.

iv. Prendre en compte la désynchronisation ds mesurée dans S_0 de deux horloges synchronisées dans S et se trouvant dans une distance x' mesurée dans S. ds est une fonction de la distance x'.

En appliquant les transformations décrites dans iii et iv dans le cas d'une particule qui se trouve au temps t de S_0 sur la position x de S_0, nous trouvons que le temps correspondant dans S est $t' = (t-d_s)t_0/t_s$.
La valeur de t_0/t_s est déterminée par les mesures effectuées dans la section 2.5. Pour le cas des lois d'interactions vérifiant la Contrainte Relativiste des Accélérations, le rapport t_0/t_s a été trouvé égal à $1/\gamma_v$. La valeur de d_s est déterminée par les mesures effectuées dans la section 2.6. Pour le cas des lois d'interactions vérifiant la Contrainte Relativiste des Accélérations, d_s a été trouvé égal à $x'v\gamma_v/c^2$. Ainsi nous aurons $t' = (t- x'v\gamma_v/c^2)/\gamma_v$, et puisque $x' = \gamma_v(x - vt)$, nous aurons $t' = [t- \gamma_v(x - vt)v\gamma_v/c^2]/\gamma_v$, ce qui donne $t' = \gamma_v[t- xv/c^2]$.

Suivant la discussion précédente, une particule ayant les coordonnées (x, y, t) dans S_0 .aura dans S les coordonnées $x' = r_0/r_s(x - vt)$, $y' = y$, $t' = (t-d_s)t_0/t_s$, où les valeurs des rapports r_0/r_s et t_0/t_s sont déterminées une fois pour toutes par les mesures décrites dans les sections 4.2.2 et 4.2.5, tandis que la valeur de d_s, dépendante de x', est déterminée pour chaque x et t en utilisant la valeur $x' = r_0/r_s(x - vt)$ ainsi que le module de synchronisation des horloges décrit dans la section 4.2.6, pour trouver la valeur de d_s correspondante à cette valeur de x'.

Le module de changement de système de référence, effectue ces transformations pour trouver les coordonnées d'un ensemble de particules dans S, à partir de leurs coordonnées dans S_0, déterminées par la simulation du cet ensemble de particules.

La recomposition des images d'un ensemble de particules effectuée selon ces transformations afin de passer du système de référence S_0 au système de référence S, et réalisée au fur et à mesure de la simulation. A chaque pas de la simulation, les coordonnées des particules sont transformées comme décrit précédemment. Après le pas i de la simulation (instant ti dans S_0), l'image des particules dans S à l'instant ti' = $\gamma_v$ti est visualisée. Mais au début de la simulation, nous ne visualisons pas d'images dans S, car elles seront incomplètes. Par exemple, au pas 0 de simulation, temps ti = 0, nous aurons pour le temps ti' de S uniquement l'image des particules se trouvant à l'origine de l'axe X de S. Ceci est une implication de la relation t' = (t-d_s)t_0/t_s [t' = γ_v(t- xv/c^2) pour les interactions vérifiant la Contrainte Relativiste des Accélérations], laquelle, pour t = 0 et pour des particules se trouvant dans des positions x positives, produit des images correspondant à des valeurs négatives de t'. Pour résoudre ce problème, nous stockons dans une mémoire temporaire toutes les images transformées des particules correspondantes à des valeurs t'$\geq$0. Dès que nous avons dans la mémoire l'image complète du système des particules pour t'=0, nous la visualisons et nous l'effaçons de la mémoire. Ensuite une fois nous avons compléter une image pour t' = dv', nous visualisons cette image et nous l'effaçons de la mémoire temporaire, et ainsi de suite pour t'= 2dv', t'=3dv' etc., où dv' correspond au pas temporel choisi pour la visualisation des images du système de référence S. Notons que pour une position x dans S_0 et pour le pas temporel dt de la simulation, l'expression t' = γ_v(t- xv/c^2) donne un pas temporel pour l'image de x dans S égal à dt' = γ_v(t+dt - xv/c^2) - γ_v(t - xv/c^2) = $\gamma_v$dt. Ainsi, le pas temporelle dans S pour tout paire d'images successives d'une position x de S_0 est constant et égal à dt' = $\gamma_v$dt.

Nous faisons attention de produire une image de chaque particule pour chaque pas de visualisation. Pour ce faire, après avoir stocké la position x' dans S d'une particule pour le $k^{ème}$ pas de visualisation, (temps de visualisation kdv') nous stockons dans une mémoire tampon la position x' de cette particule pour le dernier pas de simulation. Lorsque nous arrivons au premier cycle m de simulation tel que (m-1)$\gamma_v$dt $\leq$ (k+1)dv' $\leq$ m$\gamma_v$dt, nous choisissons le cycle de simulation m ou m-1 qui

donne la valeur temporelle ((m-1)$\gamma_v$dt ou m$\gamma_v$dt)) la plus proche de (k+1)dv'. Nous stockons comme image de la particule au temps (k+1)dv' sa position x' correspondant à ce cycle de simulation. Pour obtenir des images d'une bonne précision nous choisissons une valeur dv' beaucoup plus grande que $\gamma_v$dt. D'autre part, nous choisissons les mêmes pas de visualisation dv et dv' pour la visualisations des images des particules vues des S_0 et de S pour éviter que la visualisation déforme les relations entre ces deux points de vue.

Notons que pour illustrer cette approche de façon concrète nous avons utilisé la relation t' = γ_v(t- xv/c^2), mais l'approche est aussi applicable dans le cas de la relation générique t' = (t-d_s)t_0/t_s. Notons aussi que la façon dont nous effectuons les différentes mesures dans les sections 4.2.2 à 4.2.7, c'est-à-dire sans faire des hypothèses restrictives sur les lois des interactions, fait que les mesures et modules présentés dans ces sections sont génériques, c'est-à-dire elles sont valides pour toutes sortes de lois d'interactions, et non pas uniquement pour des lois d'interactions vérifiant la Contrainte Relativiste des Accélérations interactions (qui donnent lieu à l'espace-temps particulier décrit par les transformations de Lorentz).

4.2.8 Module de réalisation des mesures dans le système de référence S

Les résultats obtenus dans les sections précédentes, montrent que, dans le cas d'interactions vérifiant la Contrainte Relativiste de Accélérations, les mesures des dimensions des objets qui se trouvent au repos dans un système de référence inertiel S effectuées dans le système de référence S_0 donnent, dans la direction de la vitesse relative de S par rapport à S_0, des résultats conformes à la contraction des longueurs de la relativité restreinte (x_0 = x/γ_v). D'autre part, les mesures des durées des processus ayant lieu à la même position du système de référence inertiel S, effectuées dans le système de référence S_0, donnent des résultats conformes à la dilatation du temps de la relativité restreinte (t_0 = γ_vt). Néanmoins, selon la relativité restreinte, la contraction des longueurs doit être aussi observée quand nous

mesurons à partir de S un objet au repos dans S_0, et la dilatation du temps doit être aussi observée quand nous mesurons à partir de S la durée d'un processus ayant lieu dans la même position de S_0.

Intuitivement, nous devrions observer le contraire, car l'étalon de longueur de S subit la même contraction que les autres objets au repos dans S. Ainsi, quand cet étalon est comparé avec un objet au repos dans S_0 pour déterminer sa longueur, il devrait donner une augmentation de la longueur de cet objet et non pas une contraction.

Ce « paradoxe" est levé en considérant la désynchronisation dans S_0 des horloges synchrones dans S. En effet, la mesure dans S de la longueur d'un objet qui se trouve au repos dans S_0, met en jeux la valeur dans S des coordonnées des extrémités de l'objet dans le même instant du système de référence S. Ceci correspond aux valeurs dans S des coordonnées des extrémités de l'objet dans deux instants différents de S_0 (désynchronisation dans S_0 des événements distants et synchrones dans S). Le mouvement de S par rapport à S_0 fait que les coordonnées en question considérées dans ces instants différents se retrouvent rapprochées et conduisent à la contraction de la longueur des objets au repos dans S_0 et mesurés dans S, et ceci malgré la contraction de l'étalon de longueur du système de référence S. Un effet analogue produit par cette désynchronisation affecte les mesures du temps et soulève un « paradoxe » similaire concernant la dilatation de la durée des processus ayant lieu dans la même position dans S_0 et mesurés dans S. Ainsi, nous nous attendons à retrouver la contraction des longueurs et la dilatation du temps des mesures depuis le système de référence S des dimensions des objets au repos dans S_0 et des durées des processus ayant lieu dans la même position de S_0.

Pour mesurer depuis le système de référence S les dimensions des objets au repos dans S_0 et les durées des processus ayant lieu dans la même position de S_0, nous devons créer les objets et processus que nous voulons mesurer, ensuite procéder

au changement de système de référence pour passer dans le système de référence S, et enfin effectuer ces mesures.

Afin de créer des objets au repos dans S_0 et des processus ayant lieu dans la même position de S_0 nous employons les modules développés dans les sections précédentes.

Afin de procéder au changement de système de référence nous employons le module de la section précédente (voir paragraphe 4.2.7). Après le changement de système de référence, nous pouvons mesurer depuis S les longueurs des objets au repos dans S_0 et les durées des processus ayant lieu sur la même position dans S_0, de la même façon que nous mesurons depuis S_0 les longueurs des objets au repos dans S et les durées des processus ayant lieu sur la même position dans S. Ainsi, nous pouvons effectuer les mesures depuis S en employant les modules développés précédemment pour effectuer des mesures depuis S_0.

La Figure 4-12a est une copie d'écran montrant la vision depuis S_0 de l'étalon des longueurs de S_0 et de S ainsi que de la trajectoire de la particule 2 autour de la particule 1 de l'horloge au repos dans S_0 et de l'horloge au repos dans S. Comme précédemment, pour l'horloge au repos dans S, la trajectoire de la particule 2 ne représente pas l'évolution de la position de cette particule vue de S_0 mais l'évolution de sa distance par rapport à la particule 1, vue de S_0. Dans la même figure, nous trouvons deux paires de particules dont l'une effectue, autour de l'autre, une trajectoire circulaire. Nous avons deux telles paires identiques, l'une évoluant dans S_0 et l'autre dans S.

La Figure 4-12b présente une copie d'écran montrant les mêmes objets mais selon le point de vue du système de référence S, c'est-à-dire, après avoir effectué le changement de système de référence. Nous observons une situation symétrique à celle de la Figure 4-12a, c'est-à-dire que les longueurs dans l'axe x des objets et de l'étalon de longueur au repos dans S_0 sont réduits par rapport aux mêmes objets et étalon au repos dans S. Les résultats des mesures des longueurs depuis le système

de référence S montrent que cette réduction est conforme à la contraction des longueurs décrite dans la relativité restreinte.

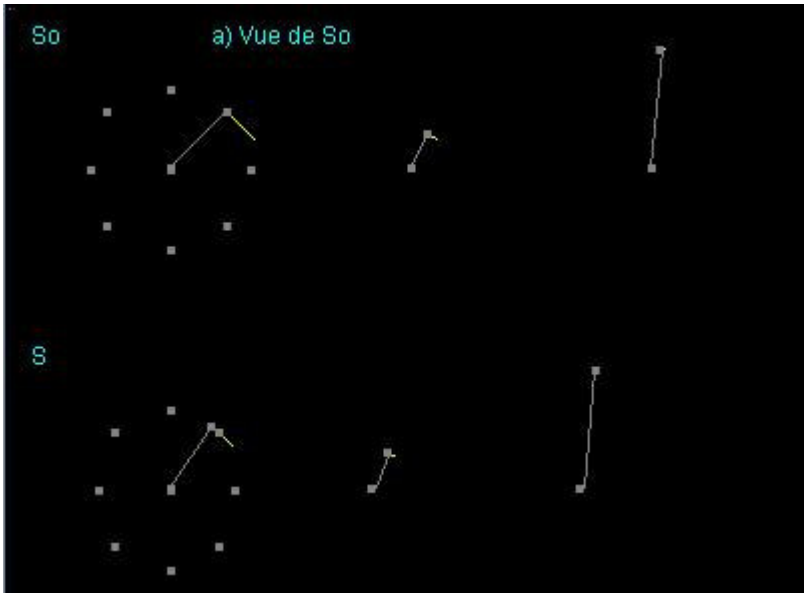


Figure 4-12a : vision depuis S_0 de l'horloge et des particules effectuant des trajets circulaires, au repos dans S_0 et au repos dans S

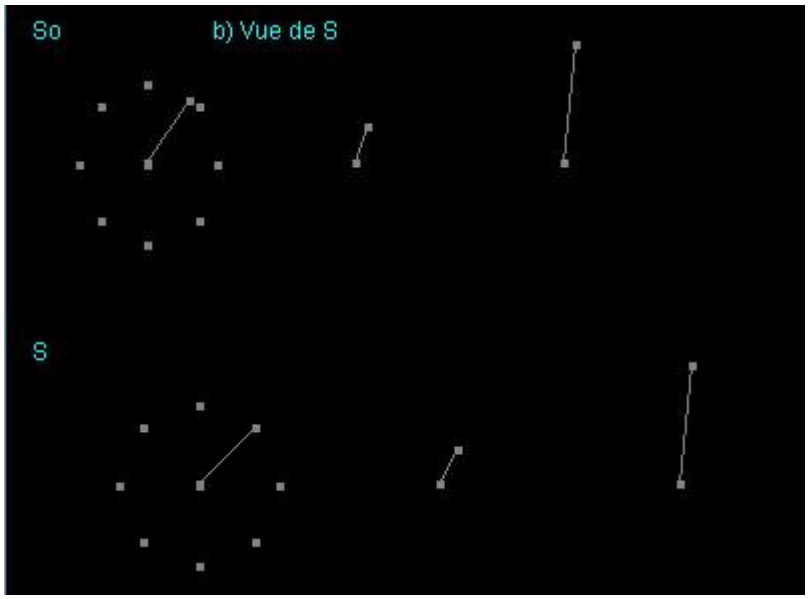


Figure 4-12b : vision depuis S de l'horloge et des particules effectuant des trajets circulaires, au repos dans S_0 et au repos dans S

Nous observons aussi un changement symétrique dans l'évolution des processus. Ainsi, dans la Figure 4-12b, la trajectoire de la particule 2 de l'horloge au repos dans S est devenue maintenant circulaire et identique à la trajectoire dans la Figure 4-12a de la trajectoire de la particule 2 de l'horloge au repos dans S_0. Cette modification symétrique est aussi vraie pour la forme de la trajectoire de la particule 2 des autres paires des particules. D'autre part, la mesure des durées des processus montre que la durée de la révolution de la particule 2 de l'horloge de S_0 est augmentée par rapport à la durée de la révolution de la particule 2 de l'horloge de S. Nous observons aussi une augmentation similaire pour la durée de l'évolution de la particule 2 des autres paires des particules. Cette augmentation est conforme à la dilatation du temps décrite dans la relativité restreinte.

4.2.9 Module de création des objets et processus identiques dans S et S_0

Les résultats obtenus dans les sections précédentes, montrent que les simulations des objets et processus engendrés par des particules dont les lois de l'évolution sont décrits par les règles C.2 et C.3 et les mesures des dimensions des objets et des durées des processus depuis le système de référence S_0 et des systèmes de référence inertiels quelconques, donnent des résultats conformes aux transformations de Lorentz (relations de contraction des longueurs et de dilatation du temps ainsi que les relations de désynchronisation dans un système de référence inertiel des horloges synchronisées dans un autre système de référence inertiel). Nous pouvons alors raisonnablement supposer que ces mesures donneront les mêmes résultats pour tout objet et processus et considérer que ces lois engendrent toujours des objets et processus conformes à ces transformations. Nous pouvons ainsi considérer que les transformations de Lorentz sont en vigueur. Dans ce cas, quand nous avons dans S_0 un objet ou un processus composé d'un ensemble de particules, nous pouvons facilement construire l'objet ou processus identique se

trouvant dans un système de référence inertiel quelconque. Pour ce faire, nous considérons dans S_0 la conformation initiale des particules, c'est-à-dire, les distances initiales entre les particules ainsi que les vitesses initiales de ces particules et nous les transformons en utilisant les relations de la contraction des longueurs et de la transformation des vitesses imposées par les transformations de Lorentz. Ainsi, nous transformons les distances en utilisant la relation (x, y) → (x/γ_v, y) et les vitesses en utilisant la relation : $(u_x, u_y) \rightarrow \left(\frac{u_{1x} + v}{1 + v \cdot u_{1x}/c^2}, \frac{u_{1y}/\gamma_v}{1 + v \cdot u_{1x}/c^2} \right)$.

Nous obtenons ainsi la conformation initiale des particules composant l'objet ou le processus mais se trouvant cette fois dans S. Cette façon de créer des objets et processus identiques dans S et S0 est plus simple que la technique utilisée précédemment qui essayait de proche en proche de trouver la vitesse initiale d'une particule qui produisait une trajectoire liée avec la trajectoire de la particule d'origine par la relation (x, y) → (x/γ_v, y). De plus, cette approche est facilement applicable dans des processus engendrés par un nombre important de particules, ce qui est très difficile avec la première approche. Bien entendu, quand nous étudions un nouveau cas des lois d'interaction de particules, la dernière approche n'est pas applicable depuis le début des simulations et mesures, c'est-à-dire avant d'avoir déterminé les relations qui lient les mesures des dimensions des objets et les durées des processus effectuées dans divers systèmes de référence inertiels.

Nous avons développé un module qui utilise ces transformations afin de créer dans un système de référence inertiel S des objets et processus identiques à des objets et processus se trouvant dans le système de référence S_0. Nous pouvons alors créer dans S_0 divers objets et processus, employer ce module pour créer dans n'importe quel système de référence inertiel des objets et processus identiques, utiliser les modules décrits précédemment pour effectuer des mesures dans S_0 et dans S, et vérifier si les résultats vérifient les relations de la relativité restreinte.

La figure 4-13 montre des particules dans S_0 ainsi que des particules identiques au premières mais se trouvant dans S. Ces dernières sont créées par le module que

nous venons de décrire. Les résultats des mesures obtenus sont conformes à la relation de contraction des longueurs de la relativité restreinte.

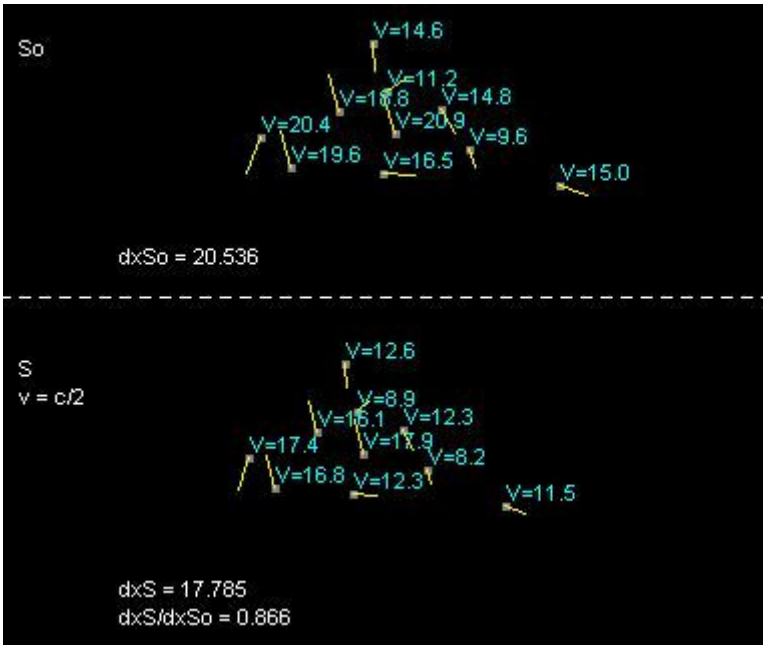


Figure 4-13 : Création des particules identiques dans le système de référence S

4.3 Autres exemples des lois d'interactions

Les modules décrits précédemment peuvent rendre très aisées les expérimentations utilisant diverses lois d'interaction et permettent de déterminer la structure de l'espace temps que ces lois engendrent. Dans la suite nous expérimentons des lois plus « bizarres » par rapport aux lois expérimentées précédemment, mais qui vérifient toujours la « Contrainte Relativiste d'Accélérations ».

Comme un premier exemple nous considérons des lois telles que la force induite par une particule 1 au repos sur une particule 2 au repos ou en mouvement est égale à $\frac{g_1 \cdot g_2}{\left(d_1 \mathrm{r}^{k1} + d_2 \mathrm{r}^{k2}\right)}$, où g1 et g2 sont des constantes associées respectivement à la

particule 1 et 2 et d_1 et d_2 sont des constantes indépendantes des particules en interaction.

En considérant que ces lois doivent être conformes à la « Contrainte Relativiste des Accélérations » nous obtenons les expressions suivantes pour l'accélération induite sur la particule 2, dans le cas général où la vitesse de la particule 1 est $\bar{v}$ et celle de la particule 2 est $\bar{u}$:

$$a_u = \frac{1}{\gamma_u{}^3 m_2} \cdot \frac{\gamma_v \cdot g_1 \cdot g_2}{d_1\left(\gamma_v{}^2 \mathrm{a}^2 + \mathrm{b}^2\right)^{(k1+1)/2} + d_2\left(\gamma_v{}^2 \mathrm{a}^2 + \mathrm{b}^2\right)^{(k2+1)/2}} \cdot \frac{\mathrm{a} \cdot u_a + \mathrm{b} \cdot u_b}{u} \qquad \text{(C.4)}$$

$$a_{\perp u}^2 = \left(\frac{1}{\gamma_u m_2} \cdot \frac{\gamma_{\mathrm{v}} \cdot g_1 \cdot g_2}{d_1\left(\gamma_v{}^2 \mathrm{a}^2 + \mathrm{b}^2\right)^{(k1+1)/2} + d_2\left(\gamma_v{}^2 \mathrm{a}^2 + \mathrm{b}^2\right)^{(k2+1)/2}} \right)^2$$

$$\left[\begin{array}{l} \dfrac{\left(b \cdot u_a - \mathrm{a} \cdot u_b - \mathrm{b} \cdot v \cdot u^2 / c^2\right)^2}{u^2} + \dfrac{u_c{}^2}{u^2}\left(\mathrm{a}^2 + \mathrm{b}^2\right) + \\ \dfrac{b^2 \cdot u^2}{c^4}\left(u_a{}^2 + u_b{}^2\right) + \dfrac{2\mathrm{b} \cdot v}{c^2}\left(\mathrm{a} \cdot u_b - \mathrm{b} \cdot u_a\right) \end{array} \right] \qquad \text{(C.5)}$$

Si les vecteurs des vitesses $\bar{v}$ et $\bar{u}$ des particules 1 et 2 et le vecteur $\bar{r}$ connectant leurs positions sont sur le même plan, la dernière expression se simplifie en résultant :

$$a_{\perp u} = \frac{1}{\gamma_u m_2} \cdot \frac{\gamma_{\mathrm{v}} \cdot g_1 \cdot g_2}{d_1\left(\gamma_v{}^2 \mathrm{a}^2 + \mathrm{b}^2\right)^{(k1+1)/2} + d_2\left(\gamma_v{}^2 \mathrm{a}^2 + \mathrm{b}^2\right)^{(k2+1)/2}}$$

$$\left[\frac{b \cdot u_a - \mathrm{a} \cdot u_b - \mathrm{b} \cdot v \cdot u^2 / c^2}{u} \right] \qquad \text{(C.5b)}$$

Selon un autre exemple de lois d'interactions, la force induite par une particule 1 au repos ou en mouvement sur une particule 2 est égale à $\frac{g_1 \cdot g_2}{\mathrm{d}^r}$, où g1 et g2 sont des

constantes associées respectivement aux particules 1 et 2 en interaction, et d est une constante indépendante des particules en interaction.

En considérant que ces lois doivent être conformes à la « Contrainte Relativiste des Accélérations » nous obtenons les expressions suivantes pour l'accélération induite sur la particule 2, dans le cas général où la vitesse de la particule 1 est $\bar{v}$ et celle de la particule 2 est $\bar{u}$:

:

$$a_u = \frac{1}{\gamma_u{}^3 m_2} \cdot \frac{\gamma_v \cdot g_1 \cdot g_2}{d^{\left(\gamma_v{}^2 \mathrm{a}^2 + \mathrm{b}^2\right)^{1/2}} \left(\gamma_v{}^2 \mathrm{a}^2 + \mathrm{b}^2\right)^{1/2}} \cdot \frac{\mathrm{a} \cdot u_a + \mathrm{b} \cdot u_b}{u} \qquad \text{(C.6)}$$

$$a_{\perp u}^2 = \left(\frac{1}{\gamma_u m_2} \cdot \frac{\gamma_v \cdot g_1 \cdot g_2}{d^{\left(\gamma_v{}^2 \mathrm{a}^2 + \mathrm{b}^2\right)^{1/2}} \left(\gamma_v{}^2 \mathrm{a}^2 + \mathrm{b}^2\right)^{1/2}} \right)^2 \left[\begin{array}{l} \dfrac{\left(b \cdot u_a - \mathrm{a} \cdot u_b - \mathrm{b} \cdot v \cdot u^2 / c^2\right)^2}{u^2} + \dfrac{u_c{}^2}{u^2} (\mathrm{a}^2 + \mathrm{b}^2) + \\ \dfrac{b^2 \cdot u^2}{c^4} (u_a{}^2 + u_b{}^2) + \dfrac{2\mathrm{b} \cdot v}{c^2} (\mathrm{a} \cdot u_b - \mathrm{b} \cdot u_a) \end{array} \right] \qquad \text{(C.7)}$$

Si les vecteurs des vitesses $\bar{v}$ et $\bar{u}$ des particules 1 et 2 et le vecteur $\bar{r}$ connectant leurs positions sont sur le même plan, la dernière expression se simplifie en

$$a_{\perp u} = \frac{1}{\gamma_u m_2} \cdot \frac{\gamma_v \cdot g_1 \cdot g_2}{d^{\left(\gamma_v{}^2 \mathrm{a}^2 + \mathrm{b}^2\right)^{1/2}} \left(\gamma_v{}^2 \mathrm{a}^2 + \mathrm{b}^2\right)^{1/2}} \frac{b \cdot u_a - \mathrm{a} \cdot u_b - \mathrm{b} \cdot v \cdot u^2 / c^2}{u} \qquad \text{(C.7b)}$$

L'expérimentation de ces exemples de lois à l'aide des modules développés précédemment est très facile.

La figure 4-14 montre une particule qui effectue une trajectoire circulaire autour d'une autre ainsi que l'étalon de longueur dans diverses orientations, créer dans S_0 en utilisant des particules et des lois d'interactions vérifiant les conditions (C.6 et C.7). Les mêmes objets sont créés dans S.

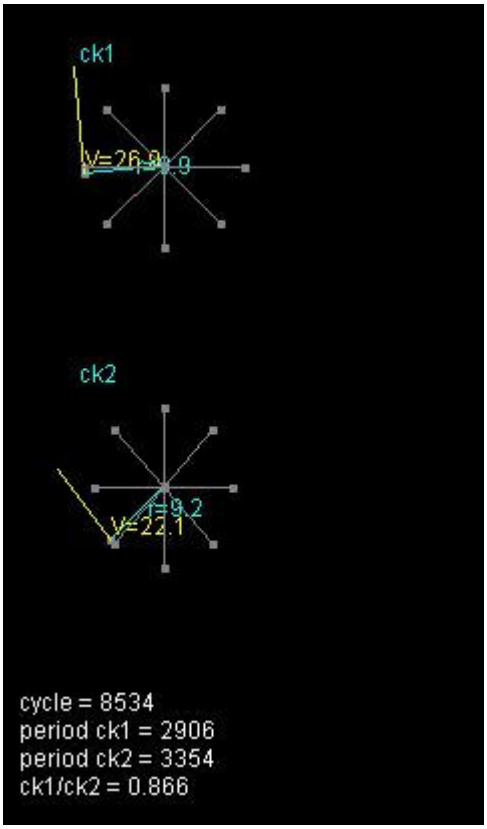


Figure 4-14 : Trajectoire circulaire d'une particule dans les systèmes inertiels de référence S et S_0 en utilisant les lois d'interaction qui sont décrits par les conditions C.6 et C.7.

La figure 4-15 montre une deuxième expérimentation. Nous avons utilisé les mêmes particules et lois que dans le cas de la Figure 4-14. Par contre, dans cette expérimentation, la valeur de la vitesse initiale de la particule qui effectue un parcours circulaire dans l'expérimentation précédente est choisie de façon arbitraire. Ainsi nous n'imposons pas une trajectoire circulaire. La trajectoire qui se produit n'est pas une trajectoire elliptique comme c'était le cas dans section 4.2.5 (voir figure 4-9). Nous observons que dans ce cas une trajectoire « elliptique » tournante se produit.

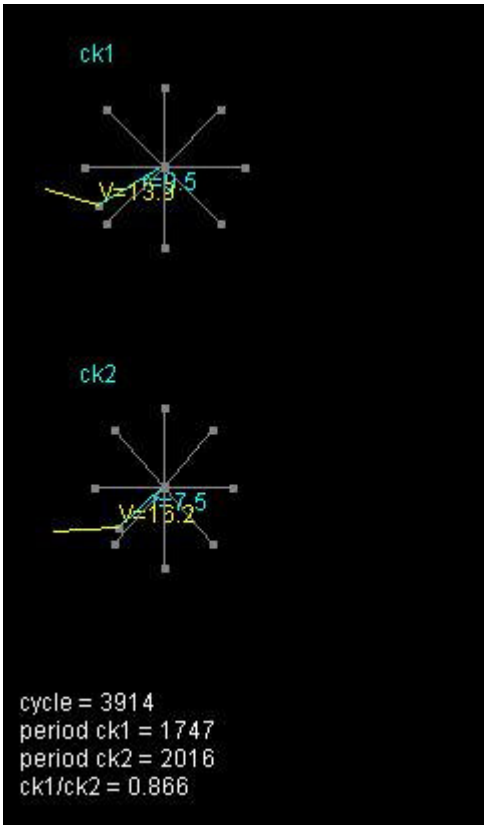


Figure 4-14 : Trajectoire « elliptique » tournante d'une particule dans les systèmes inertiels de référence S et S_0 en utilisant les lois d'interaction qui sont décrits par les conditions C.6 et C.7.

Les résultats de ces expérimentations illustrent que les lois que nous venons de décrire, donnent à nouveau lieu à un espace-temps qui vérifie la relativité restreint et valident les résultats obtenus dans [64], selon lesquels, toutes lois d'interactions qui vérifient la « Contrainte Relativiste des Accélérations » engendrent un espace-temps conforme à la relativité » restreinte.

4.4 Conclusions

Dans ce chapitre nous avons présenté la deuxième application qui a été implémentée par l'outil de simulation logicielle. Cette application concerne des petits « univers » constitués d'un ensemble des particules dont l'évolution est régie par certaines lois d'interactions. Le but de ces expérimentations consistait à étudier la façon dont sous certaines conditions, les lois d'interactions peuvent donner lieu à l'émergence d'un espace-temps relativiste. Pour effectuer ces expérimentations, nous avons enrichi l'outil de simulation d'un ensemble de modules permettant de créer des étalons de longueur et des horloges dans divers systèmes de référence inertiels, de synchroniser des horloges distantes dans un système de référence inertiel, d'effectuer des mesures des distances et du temps dans divers systèmes de référence inertiels, et de procéder au changement de système de référence inertiel, en visualisant sur l'écran l'image du système vue par l'observateur du système de référence choisi. Ces expériences ont effectivement confirmé que les lois d'interactions qui vérifient une certaine condition (appelée Contrainte Relativiste des Accélérations) engendrent un espace-temps conforme aux transformations de Lorentz.

Chapitre 5

5 Conclusions et perspectives

Le point de départ des travaux présentés dans ce manuscrit est la convergence de plusieurs facteurs : l'avènement annoncé des nanotechnologies qui devraient permettre l'intégration dans un seul composant (le nanocomposant) d'un nombre sans précédent de dispositifs ; la structure régulière et configurable de ces composants qui permettront d'implémenter par programmation l'architecture de calcul de notre choix ; l'intérêt d'étudier des systèmes complexes, naturels ou artificiels, composés par un nombre très élevé d'éléments simples. En constatant cette convergence, nous avons proposé dans le chapitre 1 le développement d'une plateforme de CAO, permettant d'exploiter de façon efficace l'extraordinaire puissance de calcul qui pourrait être apportée par ces technologies, afin de simuler des systèmes complexes. Cette plateforme devrait permettre à l'utilisateur de configurer la structure régulière des nanocomposants pour implémenter directement dans cette structure les éléments composant le système cible et l'expérimenter par simulation. Dans le même chapitre, nous avons proposé des solutions permettant de résoudre, pour certaines catégories de systèmes complexes, le problème de communications entre les éléments composant le système, ce qui devrait apporter une accélération drastique des simulations.

L'effort requis pour développer la plateforme CAO proposée dépasse largement le cadre de cette thèse. Ainsi, nous avons concentré nos efforts sur le développement d'un des outil composant la plateforme. Il s'agit d'un outil permettant à l'utilisateur de décrire, d'expérimenter et de valider facilement son système cible par simulation sur un ordinateur standard, avant de passer à la phase bien plus fastidieuse de l'implémentation dans un nanocomposant. Après avoir développé cet outil, nous l'avons utilisé pour expérimenter divers systèmes complexes.

La première famille de ces systèmes choisis se compose des écosystèmes artificiels. Leur expérimentation a montré une grande flexibilité de l'outil permettant de spécifier et simuler avec un faible effort diverses variantes d'écosystèmes artificiels. Il a aussi montré la capacité des règles que nous avons utilisées dans nos écosystèmes artificiels pour faire émerger des « organismes » capables de se développer, à partir d'une cellule « embryonnaire », en organismes adultes

possédant divers organes et fonctions (ontogenèse), d'acquérir par évolution divers fonctions comme se déplacer, se nourrir, se reproduire par croisement sexué, ou se diriger par fonction « olfactive » vers les sources de nourriture, tout en évoluant par sélection « naturelle » vers des formes adaptés aux conditions environnementales (phylogenèse). Ces comportements ont étés observés au bout d'un temps de simulation sur un PC relativement court, allant de quelques heures pour les plus simples à moins de 24 heures pour les plus complexes. Le comportement le plus évolué, la capacité de se diriger vers les sources de nourriture en utilisant une fonction d'olfaction, peut s'apparenter à un comportement intelligent qui résulte d'un réflexe inné acquis lors du processus d'évolution. Ce comportement est produit par des mécanismes simples, tels que des organes olfactifs placés sur différentes parties de l'organisme et des fonctions simples comparant les réponses de ces organes aux molécules odoriférants diffusés dans l'environnement. Ces réflexes peuvent être apparentés aux réflexes qui guident le comportement des insectes vivant en société. Ainsi, une perspective pour des travaux à venir consiste à enrichir l'écosystème en lui donnant le potentiel d'engendrer des organismes possédant divers fonctions réflexes, et exécuter de longues simulations afin de voir s'il donnera lieu à des organismes spécialisés pour effectuer des tâches complémentaires et organisés en société.

Une seconde famille de systèmes que nous avons expérimentée concerne des petits « univers » constitués d'un ensemble des particules dont l'évolution est régie par certaines lois d'interactions. Le but de ces expérimentations consistait à étudier la façon dont sous certaines conditions, les lois d'interactions peuvent donner lieu à l'émergence d'un espace-temps relativiste. Pour effectuer ces expérimentations, nous avons enrichi l'outil de simulation d'un ensemble de modules permettant de créer des étalons de longueur et des horloges dans divers systèmes de référence inertiels, de synchroniser des horloges distantes dans un système de référence inertiel, d'effectuer des mesures des distances et du temps dans divers systèmes de référence inertiels, et de procéder au changement de système de référence inertiel, en visualisant sur l'écran l'image du système vue par l'observateur du système de

référence choisi. Ces expériences ont effectivement confirmé que les lois d'interactions qui vérifient une certaine condition (appelée Contrainte Relativiste des Accélérations) engendrent un espace-temps conforme aux transformations de Lorentz. Une perspective pour des travaux à venir consiste à étudier le cas de systèmes à particules régis par des lois d'interactions vérifiant des conditions différentes de la Contrainte Relativiste des Accélérations, pour étudier la structure de l'espace temps qu'elles engendreront. Une autre perspective consiste à étudier des ensembles de particules régis par des lois d'interactions stochastiques et en particulier des interactions quantiques, comme par exemple des interactions utilisant des particules médiatrices.

Les expérimentations effectuées sur des systèmes aussi différents que les écosystèmes et les systèmes à particules, montrent que notre outil de simulation est capable de traiter une grande diversité de systèmes complexes composés par des éléments simples. D'autre part la façon dont nous avons décrit divers écosystèmes artificiels et systèmes de particules, grâce à notre outil de simulation, nous permet de varier à volonté nos expérimentations et d'explorer des règles et des lois conduisant à des structures de plus en plus riches. C'est ce que nous comptons de faire dans l'avenir immédiat. Ces expérimentations, très modestes, vu la puissance de calcul et les temps de simulation mis en jeu, mais aussi très instructives, soulignent, si besoin était, l'intérêt de posséder un support matériel d'une puissance de calcul plusieurs ordres de grandeur supérieur à nos PC actuels, ainsi que des moyens (plateforme de CAO) permettant avec un faible effort d'exploiter efficacement ce support afin de simuler des systèmes naturelles et artificiels d'une complexité autrement plus grande par rapport aux systèmes que nous avons considérés ici.

Ainsi, une dernière perspective pour les travaux à venir, mais pas la moindre, concerne le développement des autres outils de cette plateforme et en particulier de l'outil d'exploration architecturale. C'est l'outil qui demandera le plus grand effort, étant donné que l'outil de synthèse s'apparente aux outils de synthèse de haut niveau et aux outils d'implémentation des descriptions RTL sur FPGA. Bien entendu,

ces outils ne verrons leur plein intérêt qu'après l'arrivé en maturité des processus nanométriques de fabrication et l'arrivé sur le marché des nanocomposants. Néanmoins, l'outil d'exploration architecturale pourra dès son développement être utilisé pour évaluer l'efficacité de l'implémentation d'un système complexe sur une architecture donnée de nanocomposant, ou pour évaluer diverses architectures des nanocomposants afin de déterminer les plus adéquates pour implémenter un système complexe de cible ou une famille des systèmes complexes. Ces types d'expérimentations pourront entre autres, être utiles pour guider la recherche des architectures pertinentes de nanocomposants en déterminant celles qui seront les plus efficaces dans le contexte de simulation des systèmes complexes.

Bibliographie

[1] International Technology Roadmap for Semiconductors. <http://public.itrs.net >, 2004.

[2] S. Altmeyer, A. Hamidi, B. Spangenberg, and H. Kurz, “77 K single electron transistors fabricated with 0.1 mm technology ”, J. Appl. Phys., Vol. 81, pp. 8118-8120, June 1997.

[3] D. Klein, R. Roth, A.K.L. Lim, A.P. Alivisatos, and P. McEuen, “A single electron transistor made from a candmium selenide nanocrystal”, Nature, Vol. 389, pp.699-701, October 1997.

[4] J. Shirakashi, K. Matsumoto, N. Miura, and M. Konagai, “Single-electron charging effects in Nb/Nb oxide-based single-electron transistors at room temperature”, Appl. Phys. Lett., Vol. 72, pp. 1893-1995, April 1998.

[5] S. Rosenblatt, Y. Yaish, J. Park, J. Gore, V. Sazonova, and P. L. McEuen, “High performance electrolyte-gated carbon nanotube transistors”, Nano Letters 2, 869 (2002).

[6] S. J. Wind, J. Appenzeller, R. Martel, V. Derycke, and P. Avouris, “Vertical scaling of carbon nanotube field-effect transistors using top gate electrodes”, Appl. Phys. Lett. 80, 3817 (2002).

[7] A. Javey, H. Kim, M. Brink, Q. Wang, A. Ural, J. Guo, P. McIntyre, P. McEuen, M. Lundstrom, and H. J. Dai, “High Dielectrics For Advanced Carbon Nanotube Transistors and Logic”, Nature Materials 1, 241 (2002).

[8] A. Javey, J. Guo, Q. Wang, M. Lundstrom, and H. J. Dai, “Ballistic Carbon Nanotube Field-Effect Transistors”, Nature 424, 654 (2003).

[9] Y. Yaish, J. Y. Park, S. Rosenblatt, V. Sazonova, M. Brink, and P. L. McEuen, "Electrical nanoprobing of semiconducting carbon nanotubes using an atomic force microscope", Phys. Rev. Lett. 92 (2004).

[10] P. L. McEuen, M. S. Fuhrer, and H. K. Park, "Single-walled carbon nanotube electronics," Ieee T rans. Nanotechnol, vol. 1, pp. 78-85, 2002.

[11] B. M. Kim, T. Brintlinger, E. Cobas, M. S. Fuhrer, H. M. Zheng, Z. Yu, R. Droopad, J. Ramdani, and K. Eisenbeiser, "High-performance carbon nanotube transistors on SrTiO3/Si substrates", Appl. Phys. Lett. 84, 1946 (2004).

[12] P. J. Burke, "AC Performance of Nanoelectronics: Towards a Ballistic THz Nanotube Transistor", Solid-State Electronics 48, 1981 (2004).

[13] D. V. Singh, K. A. Jenkins, J. Appenzeller, D. Neumayer, A. Grill, and H. S. P. Wong, “Frequency Response of Top-Gated Carbon Nanotube Field-Effect Transistors,” IEEE Trans. Nanotechnol. 3, 383

(2004).

[14] X. Huo, M. Zhang, P. C. H. Chan, Q. Liang, and Z. K. Tang, "High-frequency S. parameters characterization of back-gate carbon nantoube field-effect transistors," in IEEE IEDM Tech. Digest, 2004, p. 691.

[15] A.O. Orlov, R. Kummamuru, J. Timler, C.S. Lent, G.L. Snider, G.H. Bernstein, “Experimental studies of quantum-dot cellular automata devises”, Mesoscopic Tunnelling Devises, 2004: ISBN: 81-271-0007-2

[16] D. Loss and D.P. DiVincenzo, Phys. Rev. A 57, 120 (1998); cond-mat/9701055.

[17] J.I. Cirac and P. Zoller, “Quantum Computations with Cold Trapped Ions,” Phys. Rev. Lett. 74, 4091 (1995); C. Monroe et al., ibid. 75, 4714 (1995).

[18] Q.A. Turchette et al., “Measurement of Conditional Phase Shifts for Quantum Logic,” Phys. Rev. Lett. 75, 4710 (1995).

[19] D. Cory, A. Fahmy, and T. Havel, “Experimental Implementation of Fast Quantum Searching,” Proc. Nat. Acad. Sci. U.S.A. 94, 1634 (1997)

[20] B. Kane, “A Silicon-based Nuclear Spin Quantum Computer”, Nature 393, 133 (1998).

[21] A. Shnirman, G. Schön, and Z. Hermon, “Quantum Manipulations of Small Josephson Junctions,” Phys. Rev. Lett. 79, 2371 (1997).

[22] D.V. Averin, “Adiabatic quantum computation with Cooper pairs,” Solid State Commun. 105, 659 (1998).

[23] L.B. Ioffe et al., Nature 398, 679 (1999).

[24] T.P. Orlando et al., “Superconducting persistent-current qubit,“ Phys. Rev. B 60, 15398 (1999).

[25] P. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," Siam Journal of Computing, 26, October, 1997, pp. 1484-1509 (special issue on quantum computation). Revision of Proc. 35th Symposium on Foundations of Computer Science (FOCS), 1994.

[26] L.K. Grover, “Quantum mechanics helps in searching for a needle in a. haystack,” Phys. Rev. Lett. 79, 325 (1997).

[27] “Nanotechnology and the Department of Defence”, Institute for Defence Analyses, IDA Paper P-3531

[28] C. Collier, G. Mattersteig, E. Wong, Y. Luo, K. Beverly, J. Sampaio, F. Raymo, J. Stoddart, and J. Heath. A [2]Catenane-Based Solid State Reconfigurable Switch. Science, 289:1172–1175, 2000.

[29] Persson et al, “A Self-Assembled Single-Electron Tunneling Devise”, Ann NY Acad Sci., 1998, 852 : 188-196

[30] Yi Cui, Lincoln J. Lauhon, Mark S. Gudiksen, Jianfang Wang, and Charles M. Lieber, “Diameter-Controlled Synthesis of Single Crystal Silicon Nanowires,” Applied Physics Letters, 78(15):2214–2216, 2001.

[31] Motlan, T.L. Tansley and E.M. Goldys, “The effects of growth temperature on the structure of GaSb/GaAs quantum dots by MOCVD”, 11-th International Semiconducting and Insulating Materials Conference Canberra Australia 3-7 July 2000.

[32] Motlan, E.M. Goldys, T.L. Tansley, “MOCVD GaSb/GaAs Quantum Dots”, Fall Meeting, Materials Research Society, Boston, MA, USA, 27 November - 1 December 2000.

[33] Motlan, K. S. A. Butcher, E. M. Goldys, and T. L. Tansley, “Optical spectroscopy of GaSb/GaAs self-assembled quantum dots grown by MOCVD”, 2000 Conference on Optoelectronic and Microelectronic Materials and Devices, Melbourne, Australia, 6-8 December 2000

[34] Ting Guo, Pavel Nikolaev, Andrew G. Rinzler, David Tománek, Daniel T. Colbert, and Richard E. Smalley, “Self-Assembly of Tubular Fullerenes”, J. Phys. Chem. 99, 10694 (1995).

[35] Young Hee Lee, Seong Gon Kim, and David Tománek, “Catalytic growth of single-wall nanotubes: An ab initio study”, Phys. Rev. Lett. 78, 2393 (1997).

[36] David Tománek, “Self-assembly of carbon nanotubes”, Mol. Mat. 10, 9-16 (1998).

[37] M. Meyyappan, “Nanotechnology: Opportunities and Challenges”, NASA Ames Research Center, http://www.ipt.arc.nasa.gov

[38] V Ng et al, “Nanostructure array fabrication with temperature-controlled self-assembly techniques” 2002 Nanotechnology 13 554-558

[39] N. Bowden, A. Terfort, J. Carbeck, and G.M. Whitesides, "Self-assembly of mesoscale objects into ordered two-dimensional arrays," Science, v. 276, April 1997, pp. 233-235.

[40] A. DeHon et al, “Sub-lithographic Semiconductor Computing Systems”, Appearing in HotChips 15, Stanford University, August 17–19, 2003

[41] Andr´e DeHon, “Array-Based Architecture for FET-based, Nanoscale Electronics”, IEEE Transactions on Nanotechnology, 2(1):23–32, March 2003.

[42] S. Goldstein et al, “Reconfigurable computing and electronic nanotechnology”, ASAP ’03, pp. 132-142, June 2003

[43] J. Han, P. Jonen, “A system architecture solution for unreliable nanoelectronic devices”, Nanotechnology, Vol. 14, No. 2, pp. 224-230, 2003.

[44] J.R. Heath, P.J. Kuekes, G.S. Snider, R.S. Williams, “A Defect-Tolerant Architecture: Opportunities for Nanotechnologies”, Science, June 98, Vol.280, p.p.1716-1721.

[45] Jerraya. A.A, et al, “Behavioral Synthesis and Component Reuse with VHDL”, Kluwer Academic Publishers, 1996.

[46] Hurk, J. Dilling, E., “System Level Design, a VHDL Based Approach”, Proceedings of Euro-DAC 1995.

[47] Elliott, J.P., “Understanding Behavioral Synthesis: A Prac-tical Guide to High-Level Design”, Kluwer Academic Pub-lishers, 1999.

[48] Gajski, D., Ramacahndran, L., “Introduction to high level synthesis”, IEEE Design and Test Computer, October 1994.

[49] Genoe, M., et al., “On the use of VHDL-based behavioral synthesis for telecom ASIC design”, In Proc. Intl. Symposium on System Synthesis ISSS'95, Feb. 1995.

[50] Ku D. and De Micheli, G., “High-level Synthesis of ASICs under Timing and Synchronization Constraints”, Kluwer Academic Publishers, 1992.

[51] Lee M.T., et al., “Domain-specific high-level modeling and synthesis for ATM switch design using VHDL”, DAC'96.

[52] Ku, D.C., De Micheli, D., “HardwareC – A Language for Hardware Design”, Stanford Univ, Tech. Rep. CSL-TR-90-419, 1988.

[53] Xue-Jie Zhang, Kam-wing Ng, Gilbert H. Young, “High-level synthesis starting from RTL and using genetic algorithms for dynamically reconfigurable FPGAs”, Proc. of the 1998 ACM fifth international symposium on Field-programmable.

[54] J. Cong and Y. Ding, “Combinational Logic Synthesis for LUT Based Field Programmable Gate Arrays”, ACM Trans. on Design Automation of Electronic Systems, vol. 1, no. 2, pp. 145-204 April 1996.

[55].J. Cong and C. Wu, “Optimal FPGA Mapping and Retiming with Efficient Initial State Computation”, IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems, vol. 18, no. 11, pp 1595 - 1607, November 1999.

[56] H. Krupnova, C. Rabedaoro, G. Saucier, “Synthesis and floorplanning for large hierarchical FPGAs”, Proc. of the 1997 ACM fifth international symposium on Field-programmable.

[57] Timothy J. Callahan, Philip Chong, André DeHon, John Wawrzynek, “Fast module mapping and placement for datapaths in FPGAs”, Pages: 123 - 132. of the 1998 ACM fifth international symposium on Field-programmable.

[58] A. Dorin, “Creating a Physicaly-Based, Virtual Metabolism With Solid Cellular Automata”, Artificial Life VII, pp. 13-20

[59] P. S. di Fenirio, “A Less Abstract Artificial Life Chemestry”, Artificial Life VII, pp. 49-53

[60] H. Sayama, “Self Replicating Worms That Increase Structural Complexity Throughout Gene Transmition”, Artificial Life VII, pp.20-30

[61] G. Tangen,, “Self-Organization in Micro-Configurable Hardware”, Artificial Life VII, pp. 31-38

[62] R. Dawkins, “The Selfish gene”, Oxford, 1989

[63] R. A. Watson, T. Reil, J. B. Pollack, “Mutualism, Parasitism, and Evolutionary Adaption”, Artificial Life VII, pp. 170-178

[64] M. Nicolaidis, “Une Philosophie Numérique des Univers”, TIMA EDITIONS, 2005 ISBN: 2-916187-00-6

Résumé

L'intitulé de notre thèse est Systèmes Complexes, outils de CAO et Nanotechnologies. Il concerne l'étude des techniques permettant de configurer un support contenant un nombre très élevé des éléments simples et interconnectés dans un réseau très complexe, de façon à émuler des systèmes complexes naturels ou artificiels. Dans ce contexte nous proposons une plateforme d'outils CAO qui aide à l'implémentation et à la simulation de ces types de systèmes. Dans un premier temps, nous élaborons un premier outil de cette plateforme permettant d'expérimenter par simulation classique les lois qui gouvernent l'évolution du système cible. Cette simulation permet de valider ou de modifier ces lois avant implémentation dans le nanoréseau. Ensuite, nous utilisons cet outil pour expérimenter divers systèmes complexes tels que des écosystèmes artificiels et des systèmes des particules dans lesquels une géométrie de l'espace-temps relativiste émerge comme une conséquence du type des lois d'interaction des particules.

Mots-clés

nano, nanotechnologie, CAO, systèmes complexes, biologie, vie artificielle, physique, relativité restreinte

Title

Complex Systems, CAD Tools and Nanotechnologies

Abstract

The title of our thesis is Complex Systems, CAD tools and Nanotechnologies. It concerns the study of the techniques allowing to configure a circuit containing a very high number of very simple elements and inter-connected in a very complex network, in order to emulate natural or artificial complex systems. In this context we propose a platform of CAD tools which helps the implementation and the simulation of this type of systems. Initially, we implemented a first tool of this platform allowing to experiment by simulation the laws which govern the evolution of the target system. This simulation allows to validate or alter these laws before the implementation in the nanonetwork. Then, we use this tool to experiment on various complex systems, such as artificial ecosystems and systems of particles in which a relativistic space time geometry emerges, as a consequence of the type of laws of interactions of the particles.

Keywords

nano, nanotechnology, CAD, EDA, complex systems, biology, artificial life, physics, special theory of relativity

Intitulé et adresse du Laboratoire

Laboratoire TIMA, 46 avenue Félix Viallet, 38031 Grenoble Cedex, France

ISBN : 2-84813-083-0
ISBNE : 2-84813-083-0