



Grenoble INP – ENSIMAG
École Nationale Supérieure d'Informatique et de Mathématiques Appliquées

Rapport de projet de fin d'études

Effectué dans l'équipe Artis - LJK/INRIA

Simuler un ciel nuageux évolutif

Eric Heitz
3^e année – Option Image
3 mai 2010 – 30 septembre 2010

Equipe Artis
INRIA Grenoble
655 avenue de l'Europe
Montbonnot
38 334 Saint Ismier Cedex France

Responsable de stage
Fabrice Neyret

Tuteur de l'école
Franck Hétroy

Remerciements

Je tiens à remercier Fabrice Neyret pour son encadrement engagé et sympathique, pour sa disponibilité et pour avoir été capable de me supporter pendant 5 mois. Espérons que la suite, en thèse, se passera aussi bien !

Merci aussi à Franck Hétroy, qui m'a mis en contact avec Fabrice, et à Marianne Genton pour la lutte administrative qui a permis la réalisation de ce stage.

Un grand merci à toute l'équipe Artis qui m'a réservé un accueil chaleureux. La qualité de l'équipe, aussi bien sur le plan humain que sur le plan scientifique, m'a convaincu d'y rester en thèse à l'issue de ce stage.

Et pour finir, un clin d'oeil à Laurent Belcour, qui fait d'excellents gâteaux, et jettera sans doute un oeil à ce rapport rien que pour m'embêter :-)

Table des matières

1	Introduction	3
I	Background et état de l'art	5
2	Physique des nuages convectifs	6
2.1	Le cycle de vie du cumulus	6
2.2	La convection	7
2.2.1	La température potentielle	7
2.2.2	Cellules de Bénard	8
2.3	La condensation	8
2.3.1	Le point de rosée	9
2.3.2	La chaleur latente	9
2.4	La turbulence	11
2.4.1	L'instabilité de Kelvin-Helmholtz	11
2.4.2	La cascade de Kolmogorov	12
3	La simulation de fluide en graphisme	13
3.1	Les équations de Navier-Stokes	13
3.1.1	Conservation de la quantité de mouvement	14
3.1.2	Conservation de la masse	14
3.2	Simulation eulérienne	14
3.2.1	Description d'une simulation eulérienne	15
3.2.2	Les méthodes eulériennes multi-résolution	15
3.3	Simulation lagrangienne	17
3.3.1	Description d'une simulation lagrangienne	17
3.3.2	Les méthodes lagrangiennes adaptatives	18
3.4	Simulation en espace de vortacité	19
4	La simulation de nuages en graphisme	21
4.1	Les nuages eulérien	21
4.2	Automates cellulaires	21
4.3	Méthodes par similarité	22

4.4	Méthodes procédurales	22
4.5	Simulation qualitative	22
II	Contributions	23
5	Analyse du problème	24
5.1	Cahier des charges	24
5.1.1	Performances	24
5.1.2	Une approche physique intuitive	25
5.1.3	Une physique scalable	26
5.2	Discussion préalable	26
5.2.1	Utilisation d'une grille	27
5.2.2	Utilisation de particules	27
5.2.3	La physique	28
5.2.4	Scalabilité	29
6	Notre modèle	30
6.1	Présentation générale	30
6.2	Particules	32
6.2.1	L'eau et ses changements de phase	32
6.2.2	La température potentielle	32
6.2.3	Le mélange avec l'atmosphère	32
6.3	Octree	33
6.3.1	Construction de l'octree	33
6.3.2	Accès aux valeurs	33
6.3.3	Création du champ de vitesse	33
6.4	Adaptativité	34
6.4.1	Le changement de résolution	35
6.4.2	Initialisation d'une zone	35
7	Résultats	37
7.1	Test de la physique	37
7.2	Performances	37
7.3	Paramétrage	39
7.4	Scalabilité	39
8	Conclusion	42

Chapitre 1

Introduction

Le domaine des jeux vidéos ou des effets spéciaux est de plus en plus exigeant en réalisme et en complexité. A l'heure où les applications les plus récentes proposent l'exploration d'univers toujours plus grands et détaillés, la représentation fidèle et convaincante des scènes naturelles reste un défi difficile.

Le ciel et ses nuages sont des éléments essentiels à ces scènes extérieures. Réussir à représenter ces objets pour une application graphique se décompose généralement en deux problèmes distincts. Le rendu a pour but de produire à l'écran une image réaliste de ces entités en fonction de leur représentation en mémoire alors que la simulation permet de générer et de modifier ces données. Si les dernières méthodes de rendu produisent des images à l'aspect très impressionnant, on cherche aussi à observer des formes et des mouvements plausibles. Il faut donc savoir générer et simuler l'évolution de ces entités correctement. C'est la question qui est au coeur de ce stage.

Des difficultés clés rendent ce problème non trivial : comment simuler un ciel nuageux à grande échelle sans faire exploser la mémoire et en restant à temps interactif ? Comment rendre la simulation intuitive et contrôlable ?

Depuis quelques années, le domaine du graphisme importe les modèles de simulation numérique de l'ingénierie mathématique pour accroître la fidélité à la physique. Ces méthodes ont pour but de résoudre des équations avec la plus grande exactitude possible. Pourtant, pour des applications ludiques, on se contenterait aussi bien d'un résultat plausible. De plus, cette exactitude se paie d'un temps de calcul et de consommation mémoire très élevés qui rendent leur utilisation difficilement interactive. Enfin, l'esprit de ces méthodes est de résoudre une équation à l'échelle la plus fine possible en espérant faire émerger des phénomènes de niveau supérieur. Ce phénomène d'émergence est très dépendant de certains paramètres parfois inconnus et quasi-impossible à anticiper. Il est donc peut commode de contrôler ces simulations pour créer des effets particuliers.

Les phénomènes qui sont la source de la mécanique du nuage convectif peuvent se décomposer sur plusieurs échelles, du détail le plus fin d'un nuage à l'ensemble d'une zone convective. Plutôt que de tout simuler en même temps et à un niveau très fin comme en calcul scientifique, notre idée est de découpler les échelles et d'utiliser des descripteurs pertinents pour chacune d'entre elles. Le découplage des échelles permettra aussi de choisir

dynamiquement le niveau de détail nécessaire à la simulation en fonction d'un critère de visibilité comme la distance ou la présence dans le champ de la caméra. Inversement, on cherchera une méthode pour reconstruire efficacement les plus petites échelles en cas de besoin.

En découplant les phénomènes et en choisissant la bonne physique pour chacun d'entre eux, on espère obtenir un modèle multi-échelle, scalable, robuste et facilement paramétrable qui fonctionnera en temps réel et produira des résultats plausibles et convaincants.

Première partie
Background et état de l'art

Chapitre 2

Physique des nuages convectifs

Les cumulus sont de la famille des nuages d'origine convective. Par les journées chaudes, le soleil réchauffe l'air près du sol et provoque l'ascension de bulles - aussi appelées des thermiques - par poussée d'Archimède. En fonction de la nature de leur environnement au sol, elles peuvent être chargées d'humidité. En montant, la pression atmosphérique diminue et la bulle atteint le point de rosée à partir duquel l'eau gazeuse au sein de la bulle finit par se condenser. C'est cette eau liquide qui rend les nuages visibles.

Les phénomènes physiques importants qui caractérisent les cumulus sont la convection thermique qui fait monter les thermiques, la condensation qui produit de l'eau liquide et la turbulence qui est un phénomène typique des fluides.

Ce chapitre a pour but de présenter le cycle de vie des nuages convectifs et les différentes lois physiques qui interviennent. Plus de détails sur la physique peuvent être trouvés dans [WH77].

2.1 Le cycle de vie du cumulus

Les premières bulles arrivées au point de rosée ne font pas très long feu. A ce moment-là, l'environnement ambiant est encore sec et le mélange entre la bulle et l'atmosphère a vite fait de la faire disparaître après une brève apparition. Les premiers nuages se présentent souvent sous la forme de petits éléments déchiquetés. C'est le stade du *cumulus fractus* (figure 2.1). Même si ces nuages ont une durée de vie très limitée, ils contribuent à enrichir l'environnement en vapeur d'eau et à favoriser l'évolution de leurs successeurs.

Après la mort des fractus, l'atmosphère devient plus clémente aux bulles et laisse place au *cumulus humilis* (figure 2.2). C'est le premier stade du cumulus avec son aspect typique surfacique bourgeonnant. On peut considérer qu'un humilis est la résultante d'une seule bulle et a une dynamique assez limitée.

Lorsque d'autres bulles montent et viennent fusionner pour former un amas plus important, on parle de *cumulus mediocris* (figure 2.3). Le mediocris a un développement principalement vertical. A ce moment-là, si l'air est suffisamment humide et instable à cause de la chaleur libérée par la condensation, le nuage peut devenir lui-même sa propre source



FIGURE 2.1 – Cumulus fractus.



FIGURE 2.2 – Cumulus humilis.



FIGURE 2.3 – Cumulus mediocris.



FIGURE 2.4 – Cumulus congestus.

de convection et atteindre le stade de *cumulus congestus* (figure 2.4), lesquels peuvent dégénérer en *cumulonimbus*.

2.2 La convection

Le comportement convectif du cumulus est caractérisé par une colonne d'air chaud montant au coeur du nuage et d'air froid descendant à l'extérieur. La poussée d'Archimède est la force à l'origine de ces mouvements et peut s'exprimer par différence de *température potentielle*.

2.2.1 La température potentielle

La température potentielle d'un fluide est celle qu'il aurait si on le ramenait au niveau de la mer de façon adiabatique, c'est-à-dire en changeant uniquement sa pression et pas ses autres variables d'état. La température potentielle est très pratique pour calculer la poussée d'Archimède entre deux fluides. En effet, si on comparait les températures absolues, il faudrait aussi tenir compte des variations de pression dans le fluide. La température potentielle permet de décrire les deux effets avec une seule variable. La force de flottabilité qu'exerce l'atmosphère sur une bulle plus chaude est orientée vers le haut et est proportionnelle à la différence de température potentielle. On a

$$F = g \frac{\theta_b - \theta_a}{\theta_a}$$

avec θ_a est la température potentielle de l'atmosphère à cette altitude et θ_b celle de la bulle. La constante g est celle de l'accélération gravitationnelle. Une bulle poursuit son ascension

jusqu'à atteindre un point d'équilibre auquel sa température potentielle égale à celle de l'environnement et donc la force nulle. Si les frottements sont peu importants, la bulle peut monter plus haut grâce à son inertie, voire osciller autour du point d'équilibre.

Lorsqu'une bulle monte adiabatiquement, sa température potentielle est constante et sa température réelle perd $10^\circ\text{C}/\text{km}$ lorsqu'elle est sèche. Si la bulle se condense en montant, la condensation relâche de la chaleur latente qui vient augmenter la température potentielle. Dans ce cas, la valeur réelle ne diminue que de $6^\circ\text{C}/\text{km}$. Le profil standard de l'atmosphère n'a pas une température potentielle constante en fonction de l'altitude. En moyenne, on peut considérer que près du sol, sa température réelle diminue de $6.4^\circ\text{C}/\text{km}$. Cette courbe change bien sûr en fonction du lieu et du jour. De plus, elle peut contenir des inversions thermiques faisant plafond.

2.2.2 Cellules de Bénard

On peut observer les cellules de Bénard lorsqu'on chauffe une casserole d'eau. Le fluide au fond devient plus chaud, moins dense et monte par poussée d'Archimède. Inversement, le liquide qui atteint la surface se refroidit, se densifie et replonge vers le bas. Ces mouvements verticaux qui résultent de l'instabilité de Taylor-Rayleigh s'organisent en cellules convectives (figure 2.5). On les reconnaît facilement à leur forme hexagonale.

On observe la même chose dans la formation d'un ciel de cumulus. Le cœur du nuage correspond à la montée des bulles chaudes, tandis que l'extérieur est formé d'air froid descendant. Ce mouvement inverse explique pourquoi les cumulus sont souvent régulièrement espacés les uns des autres, en particulier au stade du *mediocris*. C'est la mise en place de ces cellules qui permet d'alimenter les mêmes nuages et à les faire évoluer jusqu'au stade du *congestus* plutôt que de répartir la matière uniformément dans le ciel comme pour les *stratus*.

2.3 La condensation

Les bulles ascendantes se refroidissent par dépressurisation en montant. Elles atteignent le point de rosée lorsque la température est assez basse pour qu'elles soient saturées en vapeur dont l'excès se condense en gouttelettes. Ce changement de phase libère de la chaleur latente qui augmente la température potentielle de la bulle et lui permet de monter encore plus haut. Ce phénomène provoque une réaction en chaîne assimilable à une explosion : plus la bulle monte, plus elle se condense, plus sa température augmente, plus elle monte, etc. En général, la bulle se mélange avec l'environnement pendant l'ascension et voit sa proportion d'eau baisser, ce qui met un terme à sa montée explosive. Cependant, il arrive que certaines bulles atteignent la stratosphère (environ 15km à notre latitude). Cette couche a une température potentielle très importante et bloque la montée des bulles. Elles peuvent aussi atteindre auparavant le point de congélation, engendrant un comportement analogue au point de rosée.

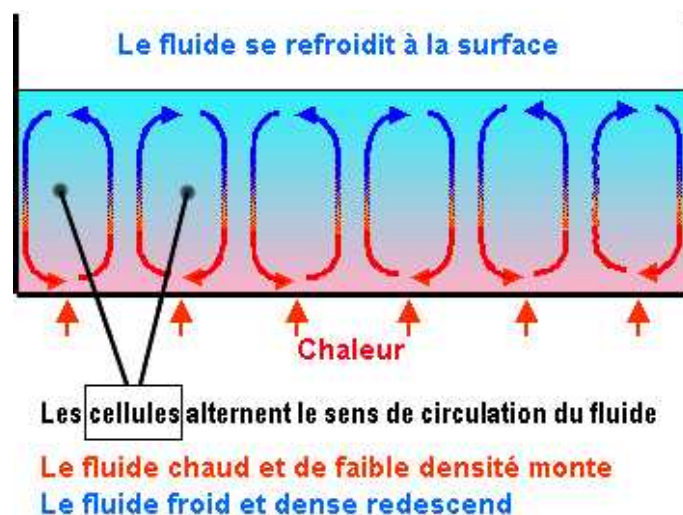


FIGURE 2.5 – Le phénomène de convection s’organise en cellules de Bénard (Wikipédia).

2.3.1 Le point de rosée

Le point de rosée d’un gaz correspond à la température à laquelle la pression partielle de vapeur d’eau atteint la pression de vapeur saturante. Le diagramme de la figure 2.6 montre que plus la température est basse, plus la vapeur d’eau a tendance à saturer vite et à passer en phase liquide. En connaissant la quantité d’eau de la bulle et sa température potentielle, on peut estimer facilement l’altitude à laquelle la condensation commence.

2.3.2 La chaleur latente

La chaleur latente est la quantité de chaleur qui est relâchée au moment de la transition de phase de l’état gazeux vers l’état liquide. C’est une transition exothermique. En effet, l’eau à l’état liquide possède une énergie moins importante qu’à l’état gazeux. Cette énergie E est dégagée sous forme de chaleur et peut-être calculée

$$E = Lm \quad (2.1)$$

où L est le coefficient de chaleur latente de l’eau (en kJ/kg) et m la quantité d’eau passée en phase liquide (en kg).

Inversement, lorsque le nuage se mélange à son environnement, son eau est dissipée sur de plus grands volumes. Ceci a pour effet de faire baisser la pression partielle d’eau, voire de la faire repasser en-dessous de la pression partielle saturante. Dans ce cas, un changement de phase inverse se produit et l’eau liquide s’évapore. Le processus d’évaporation est endothermique. Il nécessite la même quantité d’énergie qui a été dégagée par la condensation. Une partie de la température du nuage est absorbée par l’évaporation et s’en suit une chute de température. L’interaction avec l’air ambiant se produisant à la frontière du

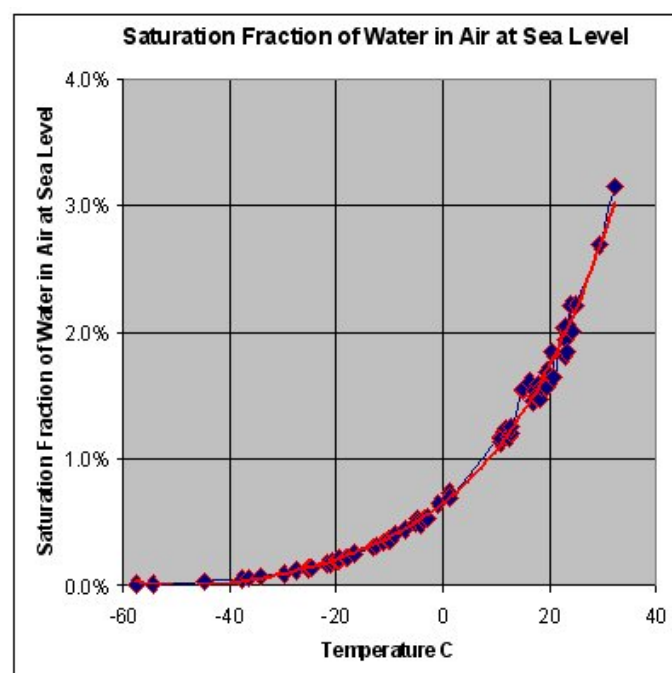


FIGURE 2.6 – Le pourcentage d'eau pour atteindre la pression partielle de vapeur saturante en fonction de la température (Wikipédia).



FIGURE 2.7 – L’instabilité de Kelvin-Helmholtz produit des vagues à la surface du nuage.

nuage, c’est à cet endroit qu’à lieu l’évaporation et le refroidissement qui provoque une poussée d’Archimède inverse. C’est pour cette raison que les bords des nuages semblent couler le long de sa surface vers le bas ou que les tourelles semblent s’effondrer après leur formation.

2.4 La turbulence

La turbulence caractérise un état tourbillonnaire dans le champ de vitesse d’un fluide. Dans le cas des nuages, elle peut-être observée à plusieurs échelles et provient de différentes sources. A grande échelle, les cellules convectives créent un mouvement turbulent de type anneau de vorticit  dans la structure globale du nuage. A l’interface entre l’int rieur et l’ext rieur, les frottements avec le vent sont   l’origine de l’instabilit  de *Kelvin-Helmholtz*. Aux plus petites  chelles, le m lange avec l’air ambiant et les jeux de changement de phase provoquent une turbulence complexe. Elle peut  tre caract ris e statistiquement par la cascade de Kolmogorov.

2.4.1 L’instabilit  de Kelvin-Helmholtz

L’interface de deux fluides se d plaçant   des vitesses diff rentes subit un effet de torsion qui engendre un des vortex orthogonaux au d placement. Sous certaines conditions, on observe des vagues   l’interface du nuage (figure 2.7). Cette instabilit  favorise le m lange avec l’air ambiant et ajoute de la complexit  dans la forme et la dynamique du nuage. A grande  chelle, ce type de m canisme produit des vagues en bande perpendiculaires au vent.

2.4.2 La cascade de Kolmogorov

Le *nombre de Reynold* d'un fluide avec les caractéristiques typiques du nuage est généralement assez élevé. Ceci signifie qu'à grande échelle c'est l'inertie qui domine le comportement du fluide, tandis que les forces visqueuses ne sont effectives qu'aux échelles très petites. Les forces appliquées sur le fluide (par exemple les dégagements thermiques) lui apportent de l'énergie. Cette dernière est peu dissipée par viscosité à grande échelle. Par contre, la turbulence est progressivement transformée vers des échelles inférieures plus sujettes à la viscosité qui favorise la dissipation. C'est le principe de la cascade de Kolmogorov : la turbulence passe des grandes échelles aux petites. Ce phénomène rejoint le principe d'entropie en thermodynamique : le désordre d'un système augmente et est irréversible. En effet, la transmission de la turbulence se fait sans mémoire. Les petites échelles ne conserve ni la géométrie ni l'anisotropie des grandes. On peut alors caractériser le spectre de turbulence d'un fluide de façon statistique en ne tenant compte que de sa quantité globale d'énergie. Cette propriété est intéressante pour modéliser les caractéristiques à des échelles plus petites que la discrétisation numérique.

Chapitre 3

La simulation de fluide en graphisme

Les créateurs d'animations sont toujours à la recherche d'un résultat visuellement de plus en plus proche de la réalité. L'utilisation de modèles physiques ajoute du réalisme au graphisme et permet à l'animateur de créer des effets complexes sans efforts particuliers. Dans le cas des fluides, les dernières méthodes utilisent les équations de Navier-Stokes qui décrivent le comportement d'une particule de gaz ou de liquide. Les modèles de calcul utilisés pour résoudre ces équations se divisent en deux grandes familles. Les méthodes eulériennes cherchent à satisfaire le modèle mathématique sur des points fixes de l'espace, généralement construit sur une grille de calcul. Les méthodes dites lagrangiennes utilisent des particules qui suivent le mouvement du fluide. Dans le premier cas, c'est l'espace qui est discrétisé, dans le second, c'est le fluide lui-même.

Ces techniques de résolution appartiennent au domaine de la *CFD* (Computational Fluid Dynamics) et sont très classiques en ingénierie. Nous adoptons ici le point de vue du graphisme qui se réapproprie les méthodes les plus simples. Nous verrons comment elles fonctionnent avec leurs inconvénients et avantages respectifs. On s'intéressera aussi à leurs variantes multi-résolution.

3.1 Les équations de Navier-Stokes

Les équations de Navier-Stokes exprimées en eulérien décrivent le comportement des fluides en faisant l'approximation que celui-ci est une matière continue. Comme la plupart des équations de la physique, elles traduisent la conservation de certaines quantités au fil du temps telles que la quantité de mouvement, la masse ou l'énergie ainsi que leurs équivalences, comme la seconde loi de Newton. Ce sont des contraintes qui doivent être satisfaites par la vitesse \mathbf{u} du fluide en tout point. Le cas de l'énergie est assez compliqué et passe généralement à la trappe pour les applications graphiques. Il reste donc deux équations représentant respectivement la loi de Newton et la conservation de la masse pour le cas des fluides incompressibles :

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{u} \cdot \nabla \mathbf{u} - \frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \quad (3.1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (3.2)$$

où t représente le temps, p la pression et \mathbf{f} les forces appliquées au fluide. Les paramètres scalaires ρ et ν sont respectivement la densité et la viscosité du fluide.

3.1.1 Conservation de la quantité de mouvement

L'équation 3.1 décrit l'évolution temporelle du vecteur vitesse \mathbf{u} en un point fixe de l'espace. Il est influencé par les caractéristiques de \mathbf{u} dans son voisinage ainsi que par les forces qui s'exercent sur le fluide. On distingue différents termes :

- $\mathbf{u} \cdot \nabla \mathbf{u}$ est l'advection du fluide par lui-même. Ceci signifie que la vitesse en un point évolue en fonction de la vitesse des particules de fluides arrivant vers ce point.
- $\frac{1}{\rho} \nabla p$ est le gradient de pression. Comme la pression a tendance à s'uniformiser dans un fluide, le fluide est poussée vers les zones basse pression et expulsée de celles où elle est trop importante.
- \mathbf{f} sont les forces externes qui s'appliquent sur le fluide. Elles peuvent venir de différentes sources : gravité, interaction avec des solides..

3.1.2 Conservation de la masse

Dans notre cas, un point de l'espace ne peut ni accumuler ni produire du fluide, il n'y a ni sources ni puits, ni de variations de densité. On dit que le fluide est incompressible ou a une divergence nulle. C'est imposé par l'équation 3.2. Cette condition est assez difficile à satisfaire numériquement et nécessite la résolution d'un gros système d'équations linéaires. Par commodité, certains solvers remplacent cette équation en ajoutant un terme de pression proportionnelle à la divergence. Cette astuce rend le calcul plus aisé mais peut produire des artefacts de simulation comme des oscillations ou des ondes de pression artificielles.

3.2 Simulation eulérienne

Un solveur eulérien résoud l'équation sur un ensemble de points fixes de l'espace. La discrétisation se fait à deux niveaux. D'une part, l'espace dans lequel évolue le fluide est discrétisé pour être représenté par un nombre fini de point. Ceci se fait généralement en utilisant une grille. D'autre part, les équations elles-mêmes sont discrétisées en utilisant une méthodes comme les *différences finies*, les *éléments finis* ou encore les *volumes finis*. Il faut ensuite résoudre l'équation en utilisant un schéma de calcul. On parle de calcul explicite lorsqu'on applique une formule de type $\mathbf{u}_{t+\Delta t} = \mathbf{u}_t + \Delta t \mathbf{f}(\mathbf{u}_t)$. Le calcul de $\mathbf{u}_{t+\Delta t}$ ne dépend que de \mathbf{u}_t et peut être fait directement. Cette méthode facile peut s'avérer très instable

car rien n'impose la cohérence entre des points adjacents. Le calcul implicite consiste à introduire des contraintes sur le résultat en écrivant $\mathbf{u}_{t+\Delta t} = \mathbf{u}_t + \Delta t \mathbf{f}(\mathbf{u}_{t+\Delta t})$. Ceci conduit à un système matriciel à inverser. Le calcul est plus laborieux mais le résultat y gagne en stabilité car la cohérence entre points voisins est imposée.

La simulation eulérienne, sortie tout droit de l'ingénierie mathématique, a été importée dans le domaine du graphisme par Forster et Metaxas [FM96]. Cette approche souffrait de gros problèmes de stabilité dus à l'utilisation d'un schéma explicite de calcul. Stam propose une approche semi-lagrangienne pour rendre la simulation plus stable [Sta99, Sta03]. L'idée est de remplacer le calcul du terme d'advection en traçant le chemin inverse de particules virtuelles pour déduire la vitesse du fluide arrivant sur un point de l'espace. Cette approche garantit une grande stabilité de calcul. Le même chercheur décrit aussi une résolution de Navier-Stokes dans un espace de fréquence via la FFT [Sta01]. Si cette approche est plus chère en complexité, elle a le mérite de rendre triviaux la plupart des calculs passés en fréquentiel et permet d'écrire des solveurs très facilement. Néanmoins, une restriction importante, due à la nature périodique des fonctions fréquentielles, est que l'espace de la simulation est cyclique. Un autre inconvénient des simulations eulériennes est la dissipation numérique. Les valeurs d'une maille ont tendance à être étalées sur plusieurs mailles après quelques pas de calcul. Ceci peut détruire certaines caractéristiques importantes du champ de vitesse, en particulier sa vorticité. Une astuce pour gérer ce problème, le vorticity confinement [FSJ01], consiste à réinjecter de la vorticité au fluide via des forces externes artificielles après chaque pas de calcul.

3.2.1 Description d'une simulation eulérienne

L'algorithme décrit ici utilise les techniques les simples. Il décrit la résolution sur une grille en utilisant la méthode des différences finies avec un schéma explicite. Le calcul des valeurs de la grille se décompose en 4 étapes. Pour chaque pas de calcul d'une valeur de temps Δt on calcule :

1. L'ajout des forces $\mathbf{u}_{t+\Delta t} = \mathbf{u}_t + \Delta t \mathbf{f}_t$
2. L'advection peut se résoudre en eulérien pur $\mathbf{u}_{t+\Delta t} = \mathbf{u}_t - \mathbf{u}t \cdot \nabla \mathbf{u}_t$ ou en semi-lagrangien (figure 3.1).
3. La viscosité se calcule facilement en opérant une diffusion des valeurs du fluide proportionnelle à ν .
4. Les trois premières étapes résolvent l'équation de conservation de la quantité de mouvement. Il faut encore satisfaire la conservation de la masse en assurant une divergence nulle de \mathbf{u} . Celle-ci se fait en résolvant un système d'équations linéaires qui imposent cette contrainte. C'est une étape très coûteuse.

3.2.2 Les méthodes eulériennes multi-résolution

En multi-résolution, on distingue l'approche multi-échelle de l'approche adaptative. La première joue sur le phénomène à plusieurs échelles différentes en même temps, tandis que

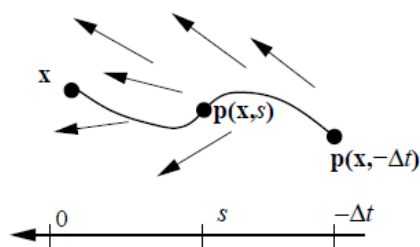


FIGURE 3.1 – L'approche semi-lagrangienne consiste à tracer le chemin inverse d'une particule de fluide pour déduire sa vitesse [Sta99].

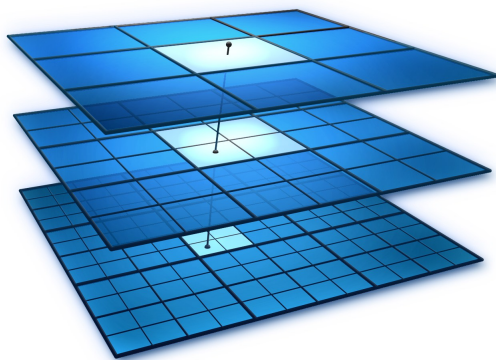


FIGURE 3.2 – Une méthode multi-grille évalue le résultat à une résolution grossière et l'utilise comme première approximation sur une grille plus fine (Image : www.robodesign.ro).

la seconde ne joue que sur une seule échelle qui est variable en temps et en espace.

Les méthodes multi-grille sont souvent utilisées en CFD. Plusieurs des étapes vues précédemment nécessitent la répétition de certains calculs pour faire converger chaque étape. En particulier, les inversions de systèmes matriciels ne peuvent se faire de façon directe (algorithmes de pivots) en raison de la très grande taille des matrices à inverser. On utilise des méthodes numériques d'inversion itérative. Un calcul est répété jusqu'à ce qu'un critère de convergence soit satisfait. Celui-ci est atteint d'autant plus vite que la solution de départ est proche de la solution visée. Pour accélérer la convergence, on cherche à trouver de "bonnes" solutions de départ. Une façon de les obtenir est de commencer par trouver des résultats à faible résolution sur des grilles grossières. Résoudre sur une grille simple permet d'obtenir rapidement une première solution qui sera utilisée sur la grille plus raffinée et qui convergera d'autant plus vite. Ce genre de méthode a fait son apparition depuis peu en graphisme [LZF10].

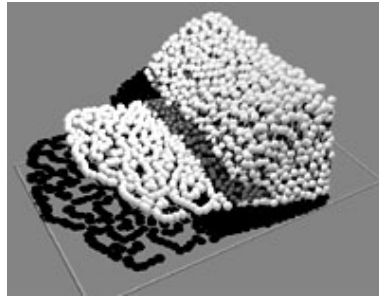


FIGURE 3.3 – Le fluide est représenté comme une somme de particules. (Wikipedia)

Les méthodes adaptatives ne jouent pas sur plusieurs échelles successivement mais, au contraire, adaptent la résolution à certains endroits en fonction de critères spécifiques. Par exemple, Losasso fait une simulation sur un octree [LGF04] dont le niveau de raffinement dépend du niveau de détail visuel.

3.3 Simulation lagrangienne

La méthode particulière SPH (Smoothed Particle Hydrodynamics) a été utilisée pour la première fois dans le domaine du calcul astrophysique. L'idée est de représenter le fluide comme une somme de particules (figure 3.3) transportant certaines quantités physiques comme leur vitesse, leur densité, leur température, etc. La méthode a été popularisée en graphisme par Desbrun [DG96] pour les substances déformables. L'utilisation spécifique aux fluides est présentée dans [MCG03]. Simultanément est sortie une méthode avec un schéma implicite [PTB⁺03] pour repositionner les particules de façon optimale après chaque pas de calcul.

Les particules ont pour avantage majeur par rapport au calcul sur grille de trivialiser le terme d'advection et la conservation de la masse. Les particules transportant avec elles leurs variables d'états, l'advection est réalisée automatiquement et sans instabilités. De plus, comme le nombre de particules est constant, la conservation de la masse est assurée. Cette division particulière est aussi pratique pour faire interagir des fluides avec différentes propriétés physiques [MSKG05]. En revanche, la reconstruction d'une valeur physique en une position arbitraire de l'espace implique de trouver les particules dans le voisinage de ce point. Pour ne pas avoir un algorithme d'une complexité en $O(n^2)$ il faut construire et maintenir une structure de données spatiale.

3.3.1 Description d'une simulation lagrangienne

Le noyau Les attributs physique de chaque particule sont lissés par une fonction de poids W , aussi appelée fonction noyau. Ces noyaux doivent avoir certaines propriétés pour être à la fois physiquement cohérents et utilisables en pratique :

1. Ils sont normalisés, c'est à dire que leur intégrale vaut 1.
2. Ils sont symétriques puisque la plupart des effets en mécanique des fluides sont isotropes.
3. Ils ont un support fini afin que seule les particules au voisinage d'un point aient une influence sur lui. Cette distance de lissage définit la notion de voisinage.
4. Ils sont facilement différentiables, pour pouvoir calculer des gradients de quantité physique.

L'utilisation d'un noyau plutôt que d'une différence finie ressemble à la discrétisation des éléments finis en eulérien. Le calcul des dérivées peut se faire de façon analytique et pré-calculée en transférant la différentiation sur le noyau plutôt que sur la quantité elle-même. Une quantité physique A , son gradient et son laplacien peuvent ainsi être reconstruits à n'importe quel point \mathbf{p} de l'espace en additionnant les contributions de toutes les particules avoisinantes pondérées par leur noyau

$$A(\mathbf{p}) = \sum_i m_i \frac{A_i}{\rho_i} W(\mathbf{p}_i - \mathbf{p}) \quad (3.3)$$

$$\nabla A(\mathbf{p}) = \sum_i m_i \frac{A_i}{\rho_i} \nabla W(\mathbf{p}_i - \mathbf{p}) \quad (3.4)$$

$$\nabla^2 A(\mathbf{p}) = \sum_i m_i \frac{A_i}{\rho_i} \nabla^2 W(\mathbf{p}_i - \mathbf{p}) \quad (3.5)$$

où A_i , m_i , ρ_i et \mathbf{p}_i sont respectivement la quantité physique, la masse, la densité et la position associées à la particule i .

La simulation fonctionne ensuite en appliquant, par exemple, un schéma eulérien à chacune des particules pour les termes de pression, de viscosité et de forces extérieures. La pression en un point \mathbf{p} est calculée comme étant proportionnelle à la différence entre la densité $\rho(\mathbf{p})$ en ce point avec la densité moyenne du milieu ρ_0

$$p(\mathbf{p}) = k(\rho(\mathbf{p}) - \rho_0) \quad (3.6)$$

3.3.2 Les méthodes lagrangiennes adaptatives

Les simulations particulières ne semblent pas proposer d'approches multi-échelle. En revanche, il existe des méthodes adaptatives. L'approche standard consiste à fusionner ou éclater les particules sous certains critères [DC99]. Cette approche a été réutilisée pour d'autres cas particuliers [APKG07, EDG08, LZH02, HHK08]. La gestion des particules devient difficile lorsque des particules de tailles différentes interagissent. Le problème peut-être résolu partiellement par l'utilisation de particules gigognes virtuelles à l'interaction des différents niveaux [RK06].

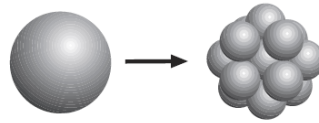


FIGURE 3.4 – La discrétisation du fluide est raffinée selon un certain critère. [DC99]

3.4 Simulation en espace de vorticité

Les méthodes présentées ci-dessus ont pour point commun de toujours travailler dans l'espace de vitesse. C'est-à-dire que c'est le champ de vitesse qui est travaillé directement. Il existe un espace alternatif : l'espace de vorticité. Le théorème de Helmholtz-Hodge nous dit qu'un champ de vitesse de divergence nulle est entièrement déterminé par sa vorticité. Dans ce cas, les deux espaces sont dits "duaux", la connaissance de l'un détermine complètement l'autre. Se placer dans l'espace de vorticité pour faire les calculs présente deux avantages majeurs. D'une part, le champ de vitesse correspondant est toujours de divergence nulle, donc il n'est pas nécessaire de faire du rééchantillonnage ou d'introduire des termes de pression pour compenser la divergence. La loi de Newton garantissant la conservation de la quantité de mouvement est aussi facilement représentable en vorticité. Comme pour la vitesse, on peut appliquer un schéma eulérien ou lagrangien, direct ou indirect, et résoudre la simulation dans cet espace. D'autre part, le champ de vorticité a la propriété d'être représentable de façon très compacte. Il prend des formes typiques de filaments ou d'anneaux (figure 3.5). Certaines techniques du graphisme tirent partie de ces avantages pour réaliser des simulations de fluide à la fois réalistes et légères [AN05, ANSN06].

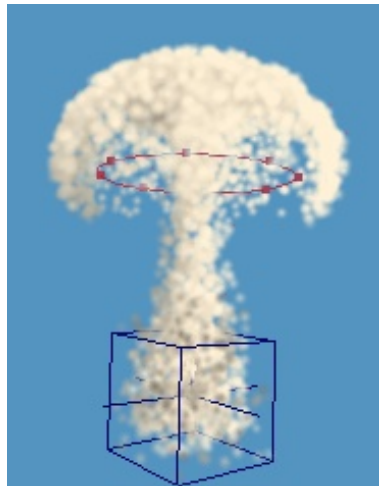


FIGURE 3.5 – Des particules advectées par le champ de vitesse d'un anneau de vorticit . [AN05]

Chapitre 4

La simulation de nuages en graphisme

Si de nombreux modèles de rendu ont été proposés pour le cas des nuages, la question de l'animation est traitée de façon relativement discrète. Avant de rentrer dans le vif du sujet, nous allons passer en revue et discuter les méthodes existantes.

4.1 Les nuages eulérien

Comme la plupart des fluides, les nuages ont déjà fait l'objet de simulation sur grille. La première version est parue dans un article de Kajiya [KVH84] sur le ray tracing et ne servait qu'à produire des jeux de test pour l'algorithme de rendu. Dobashi a repris l'idée plus récemment [MYND01, DKNY08]. Le principe est de faire une simulation de fluide relativement classique en tenant compte des effets thermodynamiques et de changement de phase. A l'échelle de la planète, on peut aussi faire une simulation [DYN06].

Les inconvénients majeurs de ce genre de simulation sont le temps de calcul (loin d'être temps réel), la difficulté à contrôler la simulation et le côté scalabilité difficilement obtainable.

4.2 Automates cellulaires

Dobashi propose introduit aussi les automates cellulaires comme alternatives à la CFD [DNO98, DKY⁺00]. Les automates cellulaires sont aussi exécutés sur une grille de calcul et offrent une alternative à Navier-Stokes en représentant le phénomène de façon simplifiée par des états discrets très simples et par un ensemble de règles booléennes. L'avantage est la simplicité et le déterminisme du calcul : pas d'instabilités ni des problème de convergence. La nature booléenne du calcul permet également un stockage et une implémentation très efficaces. L'inconvénient principal réside dans la simplicité du modèle : la physique est "trop" simple et ne permet pas de reproduire des mouvements réalistes. En outre, les problèmes inhérents au calcul sur grille sont toujours présents (contrôle, scalabilité).

4.3 Méthodes par similarité

Les méthodes par similarité utilisent les dénominateurs communs des nuages pour factoriser le calcul [GD07, WPK08]. Des caractéristiques telles que la forme ou la vitesse peuvent être factorisées pour offrir des simulations plus légères. Des physiciens ont cherché à reproduire la forme moutonneuse typique des cumulus en simulant des particules montantes avec un champ de vitesse prédéfini [WZ10]. Ces méthodes ont pour gros défaut de produire des nuages trop similaires et redondants.

4.4 Méthodes procédurales

Il existe des méthodes procédurales qui produisent des effets de fluide. Par exemple, un champ de vitesse de divergence nulle peut être créé [PT05, BHN07, Dew05] pour reproduire des impressions de mouvement de fluide. Le flow noise [PN01] peut aussi servir à animer une texture qui produit un effet de nuage. Ces méthodes procédurales ont pour gros avantages de ne coûter que l'évaluation des parties visibles. Il n'y a pas de simulation à proprement parler. Le problème est qu'il est impossible d'obtenir des effets dans lesquels interviennent des forces particulières comme la thermodynamique du nuage. En revanche, le côté procédural peut-être utile pour enrichir une animation comme c'est parfois le cas pour les simulation sur grille. Les dernières techniques eulériennes utilisent des méthodes d'amplification [SRF05, KTJG08, YKH⁺09]. Le comportement global du fluide est évalué sur une grille à résolution grossière. Des détails plus fin sont ensuite ajoutés de façon procédurale sans simulation. Ce genre de technique permet d'économiser du temps du calcul tout en produisant des résultats haute résolution.

4.5 Simulation qualitative

Neyret propose un modèle de simulation qualitative dans lequel des heuristiques différentes sont utilisées pour modéliser la physique entrant en jeu dans la formation des nuages convectifs [Ney97]. Il modélise des bulles ascendantes qui s'agglomèrent en cumulus. Plusieurs échelles sont traitées. L'entité nuage est considérée comme une somme de bulles organisées, la bulle interagit avec les autres bulles et l'atmosphère et se décompose en une hiérarchie de sous-bulles turbulentes dans l'esprit de la cascade de Kolmogorov. Cette approche est la première dans une logique multi-échelle/multi-phénomène en simulation de nuage. Cependant, elle est plus présentée dans l'esprit d'un recueil d'idées et d'une première ébauche que dans celle d'un modèle abouti et constituera un point de départ intéressant pour ce stage.

Deuxième partie
Contributions

Chapitre 5

Analyse du problème

Les méthodes existantes pour la simulation de nuage ont pour principaux défaut d'être soit inutilisables dans le cadre d'une application temps réel et inadaptées au cas de l'exploration de gros paysages (méthodes eulériennes, automates cellulaires), soit redondantes et insuffisamment riches en détails pour être convaincantes (méthodes par similarité ou procédurales). Ce stage vise l'élaboration d'une technique réaliste et adaptée au cas du temps réel dans un contexte de forte scalabilité. L'application Google Earth est un bon exemple de ce que nous visons. Le but est de donner l'illusion au spectateur d'assister à l'évolution nuageuse détaillée d'un paysage explorable et potentiellement infini. Les idées fortes de cette ambition sont la scalabilité du calcul et la souplesse du modèle utilisé. Le temps de calcul devra être utilisé intelligemment et être consacré principalement aux parties visibles de la simulation. Les parties cachées devront être traitées plus simplement et de façon plus légère. On souhaite aussi être capable de reconstruire dynamiquement une situation nuageuse sur une portion d'espace absente de la mémoire mais susceptible d'être découverte ou d'être approchée. Après avoir fixé un cahier des charges précis, on discutera des choix possibles et de leur combinatoire.

5.1 Cahier des charges

Nous cherchons à mettre au point un modèle à performances temps réel sur un PC standard, donc économe en calcul et en mémoire, facilement paramétrable et capable de reproduire les différents modes du cycle de vie du cumulus (voir chapitre 2). La scalabilité et la reconstruction dynamique sont les astuces à développer pour rendre cela possible.

5.1.1 Performances

Mettre le label "temps réel" sur un modèle de calcul signifie que sa complexité est suffisamment faible pour être exécuté de façon interactive. Par complexité, on entend généralement le temps de calcul pour une résolution suffisante. On cherche aussi à ce que le coût mémoire soit raisonnable. Il ne sera, par exemple, pas envisageable de stocker l'état

complet à haute résolution de l'atmosphère au-dessus d'un paysage de centaines voire de milliers de mètres carrés en même temps. Bien sûr, il est difficile de définir objectivement ce que signifient "beaucoup" ou "peu" en terme de temps de calcul pour juger la qualité d'un modèle. Il y a plusieurs raisons à cette difficulté :

- Le matériel. la puissance des architectures informatiques se multiplie chaque année. De nouvelles possibilités de calcul, en particulier sur GPU, ouvrent de nouveaux horizons. Des techniques considérées hier comme inabordables pour le temps réel deviennent aujourd'hui accessibles par simple augmentation de la puissance de calcul matérielle.
- L'implémentation logicielle. Pour un même modèle de calcul, certains choix d'implémentation peuvent produire des effets surprenants sur les performances. En effet, revisiter des stratégies d'allocation mémoire ou de localité temporelle et spatiale dans les structures de données peut faire gagner un facteur 10 sur la vitesse d'exécution.
- La complexité théorique. Une approche théorique est en pratique largement insuffisante pour pouvoir comparer deux algorithmes. Un algorithme peut s'avérer plus efficace qu'un autre de complexité inférieure si les facteurs par lesquels on les multiplie sont très différents. Dans la pratique, les différents algorithmes utilisés on généralement la complexité théorique optimale. On gagne surtout en jouant sur les facteurs multiplicatifs grâce à de bonnes heuristiques. Le gain s'observe alors de façon empirique et l'évaluer théoriquement est quasiment impossible.

Dans ces conditions, il est difficile de définir le "temps réel". Cependant, l'ordre de grandeur du temps de calcul est suffisant, sachant qu'on se compare à des simulations de CFD pouvant demander des heures de calcul. Le but de cet stage n'étant pas non plus de fournir l'implémentation optimale mais plutôt un modèle, on considérera que l'objectif est atteint s'il tourne interactivement ($>1\text{fps}$) sur une architecture CPU "classique" en 2010 (2GHz) avec une consommation mémoire bornée et raisonnable (quelques dizaines de Mo?). Dans ces conditions, la porte sera ouverte à une implémentation efficace voire un portage GPU qui devraient multiplier les performances, sans compter les optimisations conceptuelles possibles comme l'économie des zones à simuler, le calcul des parties cachées, etc.

5.1.2 Une approche physique intuitive

En plus d'être rapide, léger et réaliste, notre modèle se voudra aussi facile à utiliser et à incorporer à un moteur graphique existant. Un défaut majeur des méthodes sur grille est la quasi-impossibilité de configurer simplement une simulation pour obtenir un effet précis. Le problème vient du nombre trop important des paramètres à régler. Trouver les conditions initiales, les paramètres physique et la bonne résolution, sans oublier leur combinatoire, est souvent un casse-tête. Afin d'éviter ce genre de soucis, notre modèle devra satisfaire les critères suivants :

- Faible dimensionalité des paramètres. Moins il y a de paramètres à configurer, plus

il est facile de trouver la configuration qui permet de créer la situation désirée. Cette condition ne doit pas pour autant réduire la richesse de la physique à reproduire !

- Paramètres intuitifs. Pour un graphiste, il est plus simple de régler un paramètre "peu nuageux - très nuageux" qu'une variable thermodynamique. En plus d'être peu nombreux, nos paramètres devront de préférence être intuitifs et agir directement sur quelque chose d'observable dans la simulation.
- Orthogonalité des paramètres. La configuration est plus simple lorsque les effets d'une variable sont indépendant de ceux des autres. On cherchera à proposer des paramètres avec une faible corrélation.

5.1.3 Une physique scalable

La scalabilité est la possibilité du modèle à passer à l'échelle, c'est-à-dire à rester robuste même en gérant une masse de données très importante. Notre but est de produire une simulation dont le degré de précision dépend de la position de l'observateur. Les parties proches de la caméra seront simulées avec le degré le plus fin de détail. Les portions de paysage non visibles ou plus distantes utiliserons une physique moins fine. Cette stratégie permet de répartir au mieux le peu de temps de calcul dont on dispose entre deux affichages. Pour mettre cette stratégie au point, il faut définir :

- Un critère qui détermine la précision requise pour chaque zone à simuler. Celui-ci dépendra principalement de la proximité de la zone et de sa visibilité.
- La physique à simuler. Le modèle visé est multi-échelle. La physique est simulée des échelles les plus grandes vers les plus petites, les grandes influençant les petites uniquement. Ceci permet de fixer une certaine profondeur de simulation en fonction du critère. Il faudra être capable de passer de façon fluide d'une échelle à une autre quitte à perdre en précision ou, au contraire, à reconstruire artificiellement un niveau de détail non disponible.
- L'instantiation de nouvelles zones. Passer d'un niveau de détail à un niveau plus fin nécessite la reconstruction des détails perdus. De la même façon, on veut aussi pouvoir recréer facilement une situation complète en partant de rien quand l'utilisateur change de point de vue. Il faut donc être capable de créer des conditions initiales réalistes sans avoir à faire une simulation depuis le début.

5.2 Discussion préalable

De nombreux choix sont possibles pour chacune des composantes du modèle. Ceux-ci sont principalement : la représentation mémoire des données, le modèle physique, le schéma de calcul, et la gestion multi-échelle de chacun de ces aspects.

5.2.1 Utilisation d'une grille

Le calcul sur grille régulière pour résoudre un Navier-Stokes est complètement prohibée par notre cahier des charges, autant dans l'aspect "temps réel" que dans la recherche de quelque chose de simple et d'intuitif. Il pourrait par contre, via d'autres méthodes, servir à produire des effets "grande-échelle" à basse résolution comme le vent.

La représentation mémoire peut aussi être réalisé sur une grille. L'intérêt est en particulier de pouvoir accéder directement au contenu d'une zone et de son voisinage sans avoir à reconstruire sa valeur physique comme avec des particules. De plus, la topologie de la grille donne directement accès aux zones voisines. Par contre, utiliser une grille régulière impose de choisir une résolution. Une résolution trop importante surcharge la mémoire et les calculs tandis qu'une résolution trop faible produit un phénomène de lissage artificiel. Il devient problématique s'il a une influence notable sur le résultat de la simulation. Ces inconvénients peuvent être surmontés en utilisant une grille adaptative, comme un octree. L'octree présente l'avantage de pouvoir adapter sa résolution aux zones importantes et de restreindre le phénomène de lissage.

5.2.2 Utilisation de particules

Si les particules simplifient le calcul et suppriment des problèmes comme celui de la conservation de la masse ou de la dissipation et son idéales pour économiser le stockage des zones vides de nuage, elles en posent d'autres, en particulier en terme de reconstruction de valeur. Ce problème est omniprésent : dans la reconstruction des valeurs physiques, dans l'interaction entre les bulles, dans les stratégies multi-résolution..

L'interaction entre les bulles peut-être calculée bulle à bulle. Dans ce cas, il est indispensable d'utiliser une structure de données pour trouver les bulles du voisinage de façon efficace. Un autre moyen peut être d'utiliser un "proxy" pour gérer cette interaction. L'idée est de mettre à jour une structure de données temporaire en fonction de la disposition des particules. Cette structure de données, plus simple topologiquement que les particules, permet ensuite de mettre ces dernières à jour. Dans ce cas, on perd sans doute en précision mais on gagne en efficacité.

L'interaction avec l'atmosphère est aussi une question difficile. Il n'est pas pensable d'introduire des particules d'atmosphère qui interagiraient avec les particules de nuage. Comment savoir si une particule interagit avec l'atmosphère (à la surface du nuage) ou bien si elle est à l'intérieur du nuage ?

Une hiérarchie de particules pourrait être utilisée pour faire du multi-échelle lagrangien. Une particule mère contenant des particules filles, celles-ci étant elles-mêmes les mères

d'autres particules. Cette approche permettrait de découpler les échelles comme on le souhaite. Plusieurs questions se posent :

- Quel genre d'informations apporte une particule fille à sa particule mère ? Contient-elle des informations relatives (de type ondelette) ou bien absolues ? Comment gérer la hiérarchie dans le premier cas ? Comment gérer la physique dans le second cas ?
- Comment gérer une fille qui s'éloigne trop de sa mère ? La rattacher à une autre mère ? Comment trouver celle-ci ? Comment faire si la nouvelle mère n'existe pas ? La gestion de la hiérarchie particulière pose des problèmes de topologie bien plus compliqués que dans le cas d'une grille statique.
- La dispersion des particules filles engendre-t-elle une déformation de la particule mère ? Comment représenter cette déformation (vectoriel, discret) ? Comment traiter une particule trop déformée (par exemple trop étirée) ?

Ce sujet semble prometteur mais, faute de temps, nous n'aurons pas l'occasion de l'explorer dans le cadre de cette étude.

La physique de la particule peut-être entièrement gérée par le champ de valeurs reconstruit, comme pour le calcul de la vitesse de la particule. Mais on peut aussi considérer qu'une particule correspond à une thermique émise du sol et la laisser gérer une partie de sa physique, en particulier la condensation, de façon interne.

5.2.3 La physique

Nous avons vu que les équations de Navier-Stokes sont des équations de conservation, de la quantité de mouvement ou de la masse.

Le mouvement Le mouvement peut-être recréé selon plusieurs paradigmes.

- La dynamique calcule les variations de vitesse en fonction des forces exercées. Dans ce cas, l'inertie joue un rôle fondamental. Sans l'action d'une force, chaque point conserve sa vitesse.
- Le quasi-statique, contrairement à la dynamique, utilise des vitesses presque nulles à chaque instant. L'application de forces entraîne un déplacement ponctuel sans introduire une quantité de mouvement à conserver. On considère que le système n'a pas d'inertie.
- La cinématique consiste à décrire directement la vitesse, sans considérer de forces venant la modifier. Celle-ci peut, par exemple, être décrite par une fonction du temps.

Le critère de divergence nulle peut être satisfait de différentes manières. Cette contrainte est indispensable pour éviter une accumulation de matière en un point de l'espace. Il y a

trois possibilités :

- L'imposer sur grille ou avec des particulaires nécessite un rééchantillonnage par résolution d'un système linéaire compliqué. C'est une technique robuste mais coûteuse.
- On peut aussi introduire une pseudo pression proportionnelle à la divergence. C'est une façon simple et peu coûteuse de le faire, mais au risque d'introduire des oscillations.
- Une autre possibilité est de créer un champ de vitesse avec des outils procéduraux qui satisfont automatiquement ce critère.

5.2.4 Scalabilité

Les approches sur grille ou particules offrent chacune leurs options de représentations scalables. Des questions génériques restent néanmoins ouvertes :

- Comment reconstruire les détails d'une l'échelle plus fine ? Il n'est pas envisageable de les resimuler depuis le début. On peut en revanche les resimuler sur une courte période de temps ou bien les placer et les initialiser en utilisant une méthode statistique.
- Comment reconstruire un pan complet de paysage ? C'est le même problème mais sans informations initiales.

La reconstruction de détails plus fins d'une zone est nécessaire lorsque l'observateur s'en approche. L'initialisation de la nouvelle échelle peut-être faite en plaçant les détails en utilisant une heuristique statistique. Dans ce cas, le risque est que la disposition peut introduire des instabilités comme une superposition créant une pression artificielle. Une simulation de quelques pas de temps peuvent aider à recréer un état stable.

L'initialisation d'une portion de paysage est le même problème mais en partant de rien ou d'une résolution très asique. Recréer des détails est équivalent à introduire une fréquence supérieure en conservant une certaine quantité. Le problème de l'initialisation est différent dans la mesure où la quantité à conserver est inconnue. Il faut être capable de la recréer en mélangeant efficacement simulation et descripteurs heuristiques. Une fois la première échelle de détails recrée, on se ramène au problème de reconstruction des détails plus fins.

Chapitre 6

Notre modèle

Nous avons discuté les alternatives pour chacun des aspects de la simulation dans le chapitre précédent. Ce chapitre-ci est consacré à la présentation de notre modèle et à la justification des choix qu'il implique. On commencera par le présenter de façon globale, avant de visiter plus en détails les différentes briques qui le constituent.

6.1 Présentation générale

Nous avons vu que les différentes méthodes de calcul présentent chacune des avantages et des inconvénients différents et souvent complémentaires. En mariant ces méthodes de façon intelligente, on peut tirer le meilleur de chacune d'entre elles. Notre modèle utilise les ingrédients suivants :

Des particules pour représenter les proportions d'eau dans les bulles en phase liquide et gazeuse et gérer la partie thermodynamique de la physique. Ceci permet de ne stocker que ce qui est utile, d'éviter le problème de diffusion lié au stockage sur grille et trivialisé l'advection des quantités qui sont simplement portées par les particules. On peut ainsi gérer finement le stockage de matière et ses transformations thermodynamiques. Dans notre cas, une particule s'apparente aux thermiques humides émises par le sol et qui s'agglomèrent en nuages.

Un octree qui servira de "proxy" pour la reconstruction du champ spatial d'attributs physiques et le calcul des déplacements des particules. A chaque pas de calcul, les particules sont splattées dans l'octree avec leurs attributs physiques. L'octree s'occupe de reconstituer une représentation multi-échelle des valeurs physiques et notre algorithme les utilise pour calculer les instabilités thermiques en chaque point et à chaque échelle. Ainsi, plutôt que de calculer la flottabilité absolue, on évaluera ce terme relativement à chaque échelle et à l'environnement correspondant. L'intensité totale en un lieu est la somme des termes partiels à toutes les échelles, à la manière des ondelettes. Ceci permet de calculer un champ de vitesse multi-échelle qui représente efficacement les phénomènes de convection.

Algorithm 1 Modèle : mise à jour pour un pas de simulation.

```

1: {Splating des particules dans l'octree.}
2: for chaque particule  $p$  de la simulation do
3:   Déterminer le niveau  $n_p$  de précision requis en  $p$ .
4:   Splatter  $p$  dans l'octree au niveau  $n_p$ .
5: end for
6: {Mise à jour de l'octree.}
7: Mettre à jour toutes les échelles de l'octree jusqu'à la racine.
8: Mettre à jour les paramètres pour créer un champ de dans l'octree à chaque échelle.
9: {Advection des particules.}
10: for chaque particule  $p$  de la simulation do
11:   Accumuler les vitesses en  $p$  entre l'échelle 0 et  $n_p$ .
12:   Advecter la particule en utilisant la vitesse accumulée.
13: end for
14: {Mise à jour de la thermodynamique des particules.}
15: for chaque particule  $p$  de la simulation do
16:   Appliquer le mélange entre la particule et son environnement (évalué à l'échelle supérieure dans l'octree).
17:   Mettre à jour les proportions d'eau liquide et gazeuse de la particule.
18: end for

```

Les particules sont advectées avec ce champ de vitesse. L'octree sert aussi de proxy pour faire interagir chaque bulle avec son environnement. De plus, la topologie de l'octree est utilisée pour déterminer les parties internes et externes du nuage. Cette connaissance est importante pour le critère d'adaptativité.

Du procédural pour la génération des champs de vitesse dans l'octree. Si l'octree peut déterminer où ont lieu les phénomènes convectifs, encore faut-il être capable de créer un champ de vitesse correspondant. De plus, ce champ de vitesse doit être de divergence nulle. Pour respecter cette condition sans avoir à faire de calculs lourds, nous utilisons la primitive "anneau de vorticité" (figure 3.5) qui génère un champ de vitesse d'allure convective et de divergence nulle et qui peut être évaluée analytiquement [AN05]. Cette primitive est paramétrée (position, rayon, force) à partir de la force de flottabilité évaluée dans l'octree. Nous ne le faisons pas dans notre implémentation, mais on pourrait aussi facilement ajouter une composante de bruit de divergence nulle pour reproduire la turbulence avec la loi de Kolmogorov [BPC99].

En résumé, nous associons un stockage et une gestion compacte de la matière, une évaluation multi-échelle de la convection, et un calcul procédural paramétré avec précision pour reproduire la physique attendue. L'algorithme 1 décrit le comportement du modèle.

6.2 Particules

Les particules utilisées dans le modèle sont les thermiques émises par le sol. L'eau qu'elles transportent subit des changements de phase dégageant ou absorbant de la chaleur. Ces bulles sont modélisées par leurs quantités d'eau en phase gazeuse et liquide, par leur températures potentielles et par leurs volumes.

6.2.1 L'eau et ses changements de phase

Chaque bulle possède une certaine quantité d'eau sous forme gazeuse et sous forme liquide. L'équilibre de cette répartition peut-être estimé en calculant la pression partielle de vapeur saturante. Lorsque la vapeur d'eau présente dans la bulle excède la quantité saturante, le surplus passe en phase liquide. Inversement, si la saturation n'est pas atteinte, l'eau liquide repasse à l'état gazeux. Notre modèle part du principe que ces changements de phase ne sont pas complètement instantanés et sont sujets à une certaine inertie. On adopte ainsi l'équation de passage de l'état gazeux à l'état liquide

$$\begin{aligned}\Delta w_l &= \Delta t \nu_{\text{phase}} (w_g - w_s) \\ \Delta w_g &= -\Delta t \nu_{\text{phase}} (w_g - w_s)\end{aligned}\tag{6.1}$$

où w_l est la quantité d'eau liquide, w_g la quantité d'eau gazeuse et w_s la quantité saturante. Le coefficient ν_{phase} est la vitesse de changement de phase de l'eau. Celle-ci dépend de nombreux paramètres et est difficile à évaluer. Cependant, sa valeur est importante car elle a une influence sur la stabilité des nuages. Nous la définirons de manière arbitraire comme dans le modèle de Dobashi [MYND01, DKNY08].

6.2.2 La température potentielle

La température potentielle est celle que la bulle aurait si elle était ramenée instantanément aux conditions de pression du sol. Elle dépend donc très fortement des conditions initiales au sol. Cependant, cette température peut-être modifiée par le dégagement ou l'absorption de chaleur latente des changements de phase. La quantité de chaleur ΔT libérée est exprimée par

$$\Delta T = -\Delta t L \Delta w_l\tag{6.2}$$

avec L le coefficient de chaleur latente de l'eau (cf équation 2.1).

6.2.3 Le mélange avec l'atmosphère

Le mélange entre la bulle et son environnement est un mécanisme important de la physique des nuages. Il a une influence sur la proportion d'eau ainsi que sur la température potentielle de la bulle. L'apport à la bulle de matière venant de l'atmosphère, plus sèche et froide, fait baisser sa proportion d'eau et sa pression saturante. Il en résulte une évaporation

de l'eau liquide. Ce changement de phase produit un refroidissement qui peut stopper son ascension, voire la faire redescendre. On fait l'évaluation des attributs moyens du voisinage de la bulle via l'octree. Les quantités de matière échangées entre la bulle et son voisinage sont proportionnelles à sa surface S ainsi qu'à un coefficient de mélange $\nu_{\text{mélange}}$. La quantité d'eau totale w et la température potentielle de la bulle varient de la façon suivante

$$\begin{aligned}\Delta w &= \Delta t S \nu_{\text{mélange}} (w_0 - w) \\ \Delta T &= \Delta t S \nu_{\text{mélange}} (T_0 - T)\end{aligned}\tag{6.3}$$

où w_0 et T_0 sont respectivement la quantité d'eau et la température potentielle de l'environnement de la bulle. Le coefficient $\nu_{\text{mélange}}$ est inconnu et à paramétrer arbitrairement. Un modèle plus sophistiqué pourrait aussi prendre en compte la vitesse ou la turbulence de la bulle.

6.3 Octree

L'octree est reconstruit à chaque pas en fonction de la répartition des particules dans l'espace. Il permet de paramétrer les primitives qui créent un champ de vitesse et de calculer les attributs physiques au voisinage des bulles.

6.3.1 Construction de l'octree

L'octree est créé autour des particules à la résolution demandée par le critère d'adaptativité. Les valeurs des attributs physiques des bulles y sont alors accumulées aux positions correspondantes. L'octree sauvegarde les quantités d'eau et de température potentielle. De plus, on calcule un critère de présence particulaire pour chacune de ses cellules. Il correspond au volume de particules présentes divisé par le volume de la cellule.

La construction commence par le niveau le plus fin de l'octree en passant en revue toutes les particules. Il faut ensuite mettre les noeuds supérieurs à jour en remontant dans la hiérarchie.

6.3.2 Accès aux valeurs

La reconstruction des attributs physiques au sein de l'octree se fait par interpolation linéaire entre les valeurs stockées venant des bulles et le profil standard atmosphérique au prorata du coefficient de présence particulaire.

6.3.3 Création du champ de vitesse

Notre modèle fait l'approximation quasi-statique des nuages. Nous faisons l'hypothèse que les bulles n'ont pas d'inertie. Nous calculeront donc directement des vitesses, plutôt

que des forces et des accélérations. Ce choix est motivé par le fait que le calcul quasi-statique assure une plus grande stabilité. De plus, si cette hypothèse est fautive, elle reste raisonnable dans la mesure où les nuages subissent des frottements suffisamment importants pour que l'inertie ne joue pas un rôle majeur dans leur mécanique.

Le champ de vitesse doit être fidèle au phénomène convectif ayant lieu lorsqu'il y a des différences de température potentielle. Pour chaque cellule, à toutes les échelles, on calcule la différence de température potentielle avec les cellules adjacentes et on la stocke comme force de flottabilité relative.

Une cellule de l'octree génère un anneau de vorticit  (cf [AN05] pour la param trisation) concentrique   la cellule et de m me taille. Ceci se justifie par le fait que la torsion   cette  chelle a lieu   l'interface entre la cellule et ses cellules voisines. La force de l'anneau est proportionnelle   la pouss e d'Archim de calcul e pr c demment. Il est orient  vers le haut si elle est positive et vers le bas dans le cas contraire.

La vitesse s'appliquant sur une bulle est calcul e en fonction des anneaux de vorticit  dans son voisinage   toutes les  chelles. L' chelle la plus fine   utiliser est fix e en fonction du crit re d'adaptivit . Cette  chelle est fix e   deux niveaux. Il y a, d'une part, la profondeur maximale de l'octree, d'autre part la taille des bulles. Nous d cisons de lier ces deux  chelles en construisant l'octree de mani re   ce que son niveau le plus fin soit du m me ordre de grandeur que la taille de la bulle la plus grande de son voisinage. La taille des bulles d pend du crit re d'adaptivit  (visibilit , distance, etc.).

L' quation 6.4 d crit le calcul de la vitesse en un point p . Pour chaque  chelle i , les cellules p_i au voisinage de p ajoutent une contribution v_i . Celle-ci est la vitesse g n r e par un anneau de vorticit  centr  en p_i , de rayon correspondant   l' chelle i de l'octree, de force $\text{flott}(p_i)$ et appliqu  au point p . Le terme $\text{flott}(p_i)$ est la valeur relative de flottabilit  calcul e au point p_i   l' chelle i .

$$v(p) = \sum_{\text{echelle } i} \sum_{p_i \in \text{voisinage de } p} v_i(p_i, \text{flott}(p_i), p) \quad (6.4)$$

6.4 Adaptivit 

Notre mod le a la capacit  d'adapter sa complexit    un crit re de visibilit . La taille des bulles et le niveau maximal de l'octree   utiliser   un point de l'espace d pendent de celui-ci. Seules les zones visibles et proches n cessitent une haute r solution. En connaissant la position et l'orientation de la cam ra, on peut savoir si une zone est visible et   quelle distance elle se trouve. Le crit re de visibilit  est une fonction qui combine la distance et l'orientation d'une zone. La r solution demand e est plus faible lorsque la zone est loin. De fa on analogue, on  conomise aussi ce qui n'est pas dans le champ de la cam ra en baissant la r solution. Enfin, les zones trop  loign es ne sont pas simul es et seront reconstruites en cas de besoin. Les probl mes   r soudre sont le passage d'une r solution   une autre pour une zone donn e, la gestion de la transition au moment de changement de r solution et l'initialisation d'une zone invisible mais susceptible d' tre d couverte.

6.4.1 Le changement de résolution

Nous avons expliqué que la profondeur maximale de l'octree s'adapte à la taille des bulles. Contrôler la résolution de la simulation revient donc à contrôler :

- La résolution des particules. On détermine ainsi le nombre de particules et la profondeur de construction l'octree.
- Le profondeur de calcul dans l'octree. On peut choisir de ne pas ajouter les contributions des échelles les plus fines au calcul de la vitesse. Comme la vitesse est évaluée d'échelle en échelle façon "ondelette", supprimer les échelles les plus fines permet d'économiser du temps de calcul tout en gardant une cohérence dans le mouvement.
- La résolution des bulles au moment du rendu voire de leur amplification procédurale.

Les simulations lagrangiennes adaptatives doivent être capables de réaliser deux opérations : la fusion et l'éclatement des particules. Si la seconde est facile à réaliser, la première est plutôt difficile car elle doit tenir compte de la géométrie des particules à fusionner. Nous tirons parti d'avoir à notre disposition un octree dans lequel sont déjà splattés les attributs physiques des bulles. Plutôt que d'appliquer des opérateurs fusion/éclatement sur les bulles existantes, on rééchantillonne directement les valeurs stockées dans l'octree à la nouvelle résolution. De nouvelles bulles sont créées et les anciennes sont détruites.

Un rééchantillonnage brutal pose néanmoins des problèmes de discontinuité dans la simulation. Le positionnement des nouvelles particules peut créer des situations instables. De plus, la transition doit être visuellement masquée pour ne pas choquer le spectateur. Nous choisissons de superposer les deux résolutions pendant un laps de temps suffisant pour assurer une transition fluide. Après la création des nouvelles particules, les anciennes ne sont pas directement supprimées. Les bulles vont être blendées avec un coefficient de contribution $\alpha = 0$ affecté aux nouvelles bulles. Les anciennes se voient affectées du coefficient $1 - \alpha$. Ce coefficient intervient dans le splatting des attributs physiques dans l'octree ainsi que dans l'algorithme de rendu. La valeur α augmente progressivement jusqu'à atteindre 1. Lorsque c'est le cas, les anciennes bulles ne contribuent plus ni à la simulation ni au rendu et peuvent être détruites.

6.4.2 Initialisation d'une zone

Les zones au-delà d'une certaine distance ne sont pas simulées. Lorsqu'on s'en approche, il faut savoir reconstituer une situation initiale. Certaines heuristiques permettent de positionner des bulles grossières de manière plausible :

- Le profil atmosphérique et l'heure de la journée donnent des renseignements précieux sur le stade dans lequel se trouve nos cumulus. En déterminant le stade, on peut savoir quelle nombre de bulles distribuer dans l'atmosphère.
- Le point de rosée et le plafond des nuages peuvent aussi être calculés. Ceci nous indique dans quel intervalle répartir les bulles.

- Les cellules de Bénard peuvent être approximées en tenant compte des propriétés du sol. Comme notre modèle ne gère pas physiquement le sol, on peut les placer aléatoirement.
- La resimulation rapide d'une situation à très basse résolution, depuis le début ou depuis une situation estimée.

Chapitre 7

Résultats

Ce chapitre présente les premiers résultats de l'implémentation de notre modèle, à 1 mois de la fin du stage. Nous avons développé une maquette 3D non-scalable et une maquette 2D avec une partie des mécanismes de rééchantillonnage.

7.1 Test de la physique

Notre implémentation 3D intègre la partie physique (convection, changements de phase, etc.) du modèle. En outre, nous avons implémenté une visualisation 3D simplifiée, à base de point-splatting, pour pouvoir mieux juger l'allure des résultats. C'est une bonne base pour vérifier s'il reproduit bien les formes de nuage attendues. Le tableau 7.1 montre l'évolution d'un nuage créé avec une seule source de thermiques. On reconnaît bien les différentes phases du cycle de vie du cumulus : le *fractus* avec son aspect déchiqueté, puis l'*humilis* après une condensation plus forte et le *mediocris* avec un premier développement vertical avant de devenir un *congestus*. La simulation reproduit de nombreuses caractéristiques : ascension au centre du nuage, mélange avec l'atmosphère, évaporation et descente à l'extérieur.

7.2 Performances

La simulation présentée dans le tableau 7.1 fonctionne avec un maximum de 300 particules et tourne à 30 images par seconde (sans compter l'algorithme de rendu). Notre implémentation reste interactive (5fps) sur une simulation avec 3000 particules. L'utilisation de l'algorithme de rendu divise le nombre d'images par seconde par un facteur 2. Il existe des méthodes dédiées plus efficaces et constituent un sujet d'étude en soi.

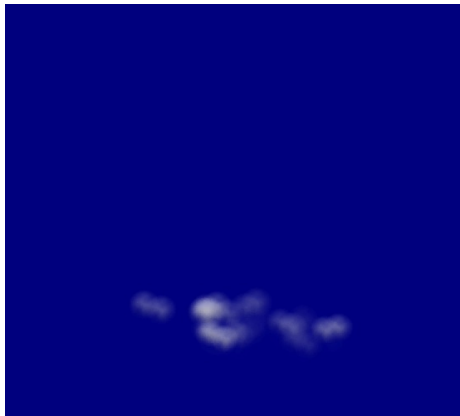



Fractus	Humilis
 A photograph of a Fractus cloud, characterized by a dark, broken, and irregular appearance with some white patches.	 A photograph of a Humilis cloud, appearing as a low, white, and somewhat flat layer of clouds.
Mediocris	Congestus
 A photograph of a Mediocris cloud, showing a more developed, white, and somewhat puffy appearance.	 A photograph of a Congestus cloud, appearing as a tall, white, and highly textured, puffy cloud.

TABLE 7.1 – Le modèle reproduit les différents cycle de la vie du cumulus.

7.3 Paramétrage

La simulation est encore assez sensible à certains paramètres physiques :

- La vitesse ν_{phase} de changement de phase de l'eau. Si elle est trop grande, la chaleur latente dégagée par condensation produit très vite des tourelles instables et on obtient un nuage chaotique et sans structure. Au contraire, une valeur trop faible empêche le nuage d'évoluer vers les stades mediocris et congestus.
- Le coefficient ν_{melange} qui permet de calculer à quelle vitesse une bulle se mélange à son environnement. Il a une influence directe sur la durée de vie du nuage. S'il est trop important, les nuages meurent prématurément. Une valeur trop petite ne permet pas de reproduire le phénomène de stoppe l'ascension du nuage. On obtient alors des nuages uniquement constitués de tourelles et sans base.
- Le profil atmosphérique. Il a une influence directe sur la forme des nuages. Par exemple, une inversion thermique peut entraver l'ascension des bulles et créer une forme d'enclume au sommet du nuage.
- La création des bulles. La fréquence et la répartition des bulles jouent un rôle important dans l'évolution du nuage. Le modèle semble suffisamment stable pour reproduire l'effet "cellules de Bénard" avec une répartition initiale homogène (figure 7.2), ce qui nous dégage de la contrainte d'avoir à piloter la position des nuages, notre simulation le faisant spontanément. La fréquence de création des bulles reste à définir.

Une difficulté supplémentaire est la forte corrélation de ces paramètres. Il serait intéressant d'arriver à les fusionner, voire, à les décrire qualitativement pour les rendre plus intuitifs. L'idéal serait de proposer une interface utilisateur avec une poignée de nombres ou de courbes et un retour visuel rapide.

7.4 Scalabilité

Notre stratégie de scalabilité ne peut fonctionner que si l'algorithme produit des résultats cohérents à des résolutions différentes, ce qui attendu si notre modélisation physique est correcte. Le tableau 7.3 présente les résultats d'une simulation 2D à des échantillonnages respectifs de 250, 1000 et 4000 particules pour un même volume. Précisons bien qu'ici, aucun algorithme de rendu n'a été utilisé pour le cas 2D. Les bulles existantes sont simplement affichées. Ceci explique pourquoi les résultats ressemblent peu visuellement à du nuage. Ce qui nous intéresse dans ces images est la cohérence de l'évolution produite à chacune des résolutions. Les images de la première ligne représentent deux nuages en cours de formation. Si l'ensemble semble connexe, la base est en fait abaissée au centre et correspond à la frontière entre deux cellules de Bénard. Les secondes images montrent une situation dans laquelle des tourelles commencent à se former. En particulier, sur le nuage de gauche, on voit nettement une tourelle pousser vers le côté droit. On peut dire que les caractéristiques importantes de cette simulation sont présentes à chaque résolution.

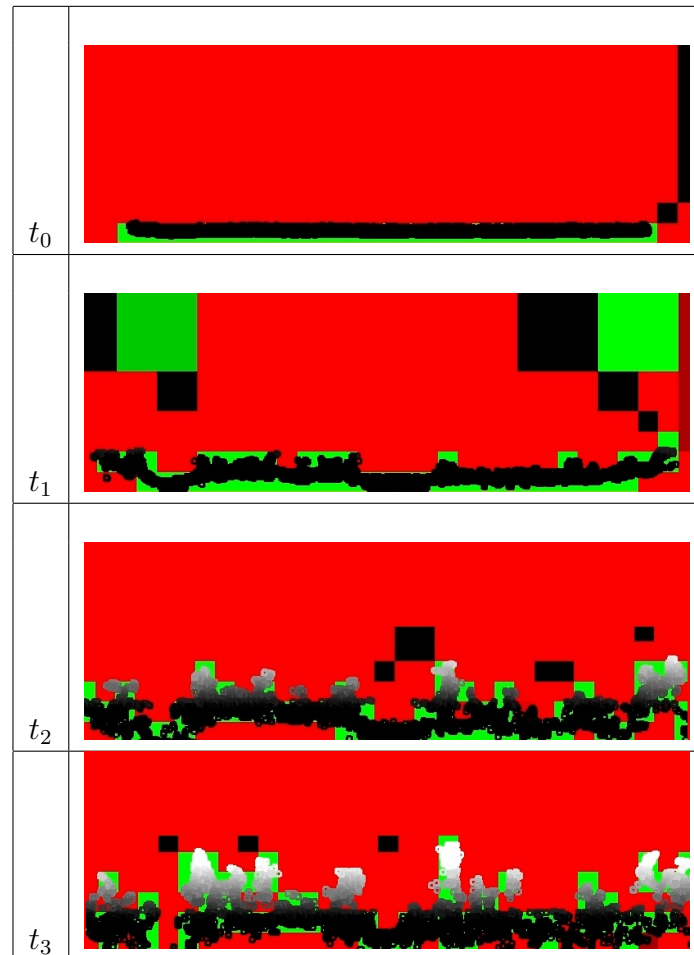


TABLE 7.2 – La simulation fait émerger des cellules de Bénard à partir d’une répartition initiale homogène. Les bulles sont représentées en niveau de gris (blanche si présence d’eau liquide). Les cellules de l’octree sont vertes si la flottabilité relative est positive (courant ascendant), rouge dans le cas contraire (courant descendant). On remarque que la physique multi-échelle produit un effet de Bénard fractal.

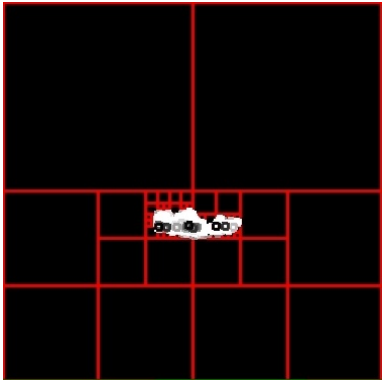
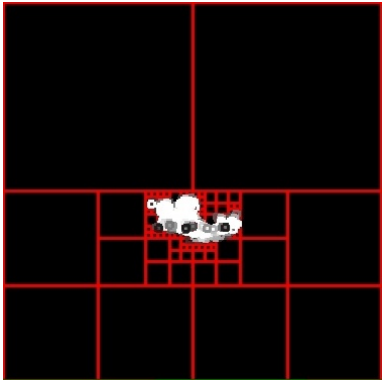
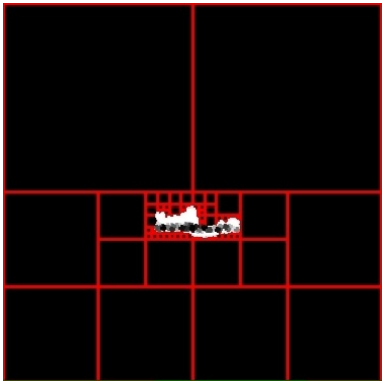
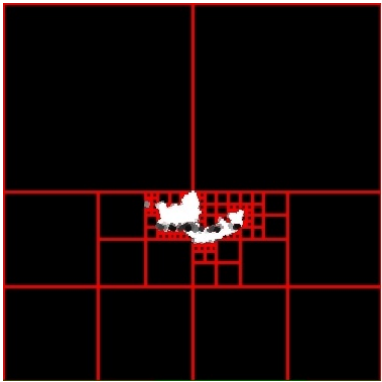
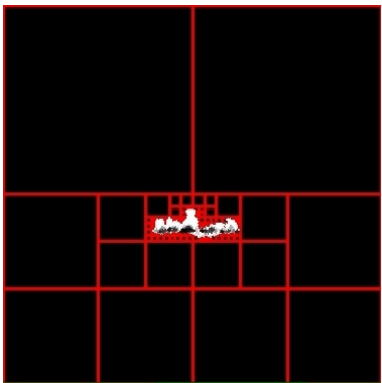
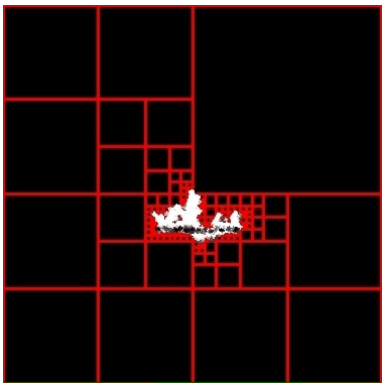
	t_0	t_1
250 particules		
1000 particules		
4000 particules		

TABLE 7.3 – La simulation produit les mêmes caractéristiques à différentes résolutions.

Chapitre 8

Conclusion

Les résultats préliminaires montrent que notre modèle reproduit bien la physique des nuages convectifs en 3D. C'est la première approche temps-réel qui reproduit l'évolution complète des cumulus, depuis leur naissance jusqu'à leur apogée en congestus. Le modèle est à la fois simple et robuste.

Les briques "rééchantillonnage" et "simulation multi-résolution" ont été testées en 2D et sont prometteuses. Le passage d'une résolution à une autre préserve les caractéristiques importantes de la forme observable (base, tourelles, etc.). L'utilisation du blending pour faciliter la transition entre les différentes résolutions assurera une cohérence encore plus importante. La prochaine étape consistera à intégrer ces briques dans l'implémentation 3D et à les relier au critère de visibilité. Nous espérons réaliser ces étapes d'ici la fin du stage.

L'algorithme de reconstruction d'un pan de paysage non disponible en mémoire est aussi à implémenter et à tester.

Un travail supplémentaire sur le paramétrage est aussi à envisager pour pouvoir présenter un modèle plus facile à utiliser. Néanmoins, si les paramètres actuels sont encore un peu nombreux, leur manipulation reste abordable et les changer produit des effets visuels directs et facilement anticipables, ce qui est loin d'être le cas pour les simulations de type CFD.

Bibliographie

- [AN05] Alexis Angelidis and Fabrice Neyret. Simulation of smoke based on vortex filament primitives. In *SCA '05 : Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 87–96, New York, NY, USA, 2005. ACM.
- [ANSN06] Alexis Angelidis, Fabrice Neyret, Karan Singh, and Derek Nowrouzezahrai. A controllable, fast and stable basis for vortex based smoke simulation. In *SCA '06 : Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 25–32, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.
- [APKG07] Bart Adams, Mark Pauly, Richard Keiser, and Leonidas J. Guibas. Adaptively sampled particle fluids. In *SIGGRAPH '07 : ACM SIGGRAPH 2007 papers*, page 48, New York, NY, USA, 2007. ACM.
- [BHN07] Robert Bridson, Jim Houriham, and Marcus Nordenstam. Curl-noise for procedural fluid flow. *ACM Trans. Graph.*, 26(3) :46, 2007.
- [BPC99] Daniel Barrero, Mathias Paulin, and René Caubet. A Physics Based Multi-Resolution Model for the Simulation of Turbulent Gases and Combustion. In N. Magnenat-Thalmann and D. Thalmann, editors, *Eurographics Workshop on Computer Animation and Simulation, Milano, Italy, 07/09/99-09/09/99*, pages 177–188, Wien-NewYork, septembre 1999. Springer.
- [DC99] Mathieu Desbrun and Marie-Paule Cani. Space-time adaptive simulation of highly deformable substances. Rapport de recherche 3829, INRIA, December 1999.
- [Dew05] Ivan Dewolf. Abstract divergence-free noise. Technical report, 2005.
- [DG96] Mathieu Desbrun and Marie-Paule Gascuel. Smoothed particles : a new paradigm for animating highly deformable bodies. In *Proceedings of the Eurographics workshop on Computer animation and simulation '96*, pages 61–76, New York, NY, USA, 1996. Springer-Verlag New York, Inc.
- [DKNY08] Yoshinori Dobashi, Katsutoshi Kusumoto, Tomoyuki Nishita, and Tsuyoshi Yamamoto. Feedback control of cumuliform cloud formation based on computational fluid dynamics. *ACM Trans. Graph.*, 27(3) :1–8, 2008.
- [DKY⁺00] Yoshinori Dobashi, Kazufumi Kaneda, Hideo Yamashita, Tsuyoshi Okita, and Tomoyuki Nishita. A simple, efficient method for realistic animation of clouds.

- In *SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 19–28, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [DNO98] Yoshinori Dobashi, Tomoyuki Nishita, and Tsuyoshi Okita. Animation of clouds using cellular automaton. In *Proc. of Computer Graphics and Imaging'98*, pages 251–256, 1998.
- [DYN06] Yoshinori Dobashi, Tsuyoshi Yamamoto, and Tomoyuki Nishita. A controllable method for animation of earth-scale clouds. *CASA 2006*, pages 43–52, 2006.
- [EDG08] Benoît Crespin Emmanuelle Darles and Djamchid Ghazanfarpour. Une approche multirésolution lagrangienne pour la simulation de vagues déferlantes. In *Revue Electronique Francophone d'Informatique Graphique*, 2008.
- [FM96] Nick Foster and Dimitri Metaxas. Realistic animation of liquids. *Graph. Models Image Process.*, 58(5) :471–483, 1996.
- [FSJ01] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. Visual simulation of smoke. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 15–22, New York, NY, USA, 2001. ACM.
- [GD07] Jakub Grudziński and Adrian Debowski. Clouds and atmospheric phenomena simulation in real-time 3d graphics. In *MIRAGE'07 : Proceedings of the 3rd international conference on Computer vision/computer graphics collaboration techniques*, pages 117–127, Berlin, Heidelberg, 2007. Springer-Verlag.
- [HHK08] Woosuck Hong, Donald H. House, and John Keyser. Adaptive particles for incompressible fluid simulation. *Vis. Comput.*, 24(7) :535–543, 2008.
- [KTJG08] Theodore Kim, Nils Thürey, Doug James, and Markus Gross. Wavelet turbulence for fluid simulation. *ACM Trans. Graph.*, 27(3) :1–6, 2008.
- [KVH84] James T. Kajiya and Brian P Von Herzen. Ray tracing volume densities. *SIGGRAPH Comput. Graph.*, 18(3) :165–174, 1984.
- [LGF04] Frank Losasso, Frédéric Gibou, and Ron Fedkiw. Simulating water and smoke with an octree data structure. In *SIGGRAPH '04 : ACM SIGGRAPH 2004 Papers*, pages 457–462, New York, NY, USA, 2004. ACM.
- [LZF10] Michael Lentine, Wen Zheng, and Ronald Fedkiw. A novel algorithm for incompressible flow using only a coarse grid projection. *ACM Trans. Graph.*, 29(4) :1–9, 2010.
- [LZH02] S. F. Li D. Qian L. Zhang, W. K. Liu and S. Hao. Survey of multi-scale meshfree particle methods. In *Meshfree Methods for Partial Differential Equations*, pages 441–458, 2002.
- [MCG03] Matthias Müller, David Charypar, and Markus Gross. Particle-based fluid simulation for interactive applications. In *SCA '03 : Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages

- 154–159, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [MSKG05] Matthias Müller, Barbara Solenthaler, Richard Keiser, and Markus Gross. Particle-based fluid-fluid interaction. In *SCA '05 : Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 237–244, New York, NY, USA, 2005. ACM.
- [MYND01] Ryo Miyazaki, Satoru Yoshida, Tomoyuki Nishita, and Yoshinori Dobashi. A method for modeling clouds based on atmospheric fluid dynamics. In *PG '01 : Proceedings of the 9th Pacific Conference on Computer Graphics and Applications*, page 363, Washington, DC, USA, 2001. IEEE Computer Society.
- [Ney97] Fabrice Neyret. Qualitative simulation of cloud formation and evolution. In D. Thalmann and M. Van de Panne, editors, *8th Eurographics Workshop on Computer Animation and Simulation (EGCAS'97)*, pages 113–124, New York City, NY, September 1997. Eurographics, Springer Wein.
- [PN01] Ken Perlin and Fabrice Neyret. Flow noise. In *28th International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2001, Technical Sketches and Applications, August, 2001*, pages 187–187, Los Angeles, CA, Etats-Unis, 2001.
- [PT05] Mayur K. Patel and Noah Taylor. Simple divergence-free fields for artistic simulation. *J. Graphics Tools*, 10(4) :49–60, 2005.
- [PTB⁺03] Simon Premoze, Tolga Tasdizen, James Bigler, Aaron Lefohn, and Ross T. Whitaker. Particle-based simulation of fluids. In *Comp. Graph. Forum (Eurographics Proc.)*, volume 22, pages 401–410, 2003.
- [RK06] Philip Dutre Leonidas J. Guibas Mark Pauly Richard Keiser, Bart Adams. Multiresolution particle-based fluids. Technical report, ETH Zurich, 2006.
- [SRF05] Andrew Selle, Nick Rasmussen, and Ronald Fedkiw. A vortex particle method for smoke, water and explosions. *ACM Trans. Graph.*, 24(3) :910–914, 2005.
- [Sta99] Jos Stam. Stable fluids. In *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 121–128, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [Sta01] Jos Stam. A simple fluid solver based on the FFT. *J. Graph. Tools*, 6(2) :43–52, 2001.
- [Sta03] Jos Stam. Real-time fluid dynamics for games. In *Proceedings of the Game Developer Conference*, 2003.
- [WH77] John M. Wallace and Peter V. Hobbs. *Atmospheric Science : An Introductory Survey*. Academic Press, 1977.
- [WPK08] Bei Wang, Jingliang Peng, and C. C. Kuo. Cumulus cloud synthesis with similarity solution and particle/voxel modeling. In *ISVC '08 : Proceedings of the 4th International Symposium on Advances in Visual Computing*, pages 65–74, Berlin, Heidelberg, 2008. Springer-Verlag.

- [WZ10] Yong Wang and Giovanni Zocchi. Shape of fair weather clouds. *Phys Rev Lett*, 104(11) :118502, 2010.
- [YKH⁺09] Jong-Chul Yoon, Hyeong Ryeol Kam, Jeong-Mo Hong, Shin-Jin Kang, and Chang-Hun Kim. Procedural synthesis using vortex particle method for fluid simulation. *Comput. Graph. Forum*, 28(7) :1853–1859, 2009.