

Simulation des systèmes gravitationnels et application à la cosmologie

Mattéo CLÉMOT

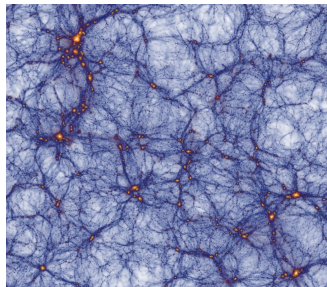
Filière Mathématiques - Physique

Juin 2018

Introduction



M51 vue par Hubble
 $\sim 10^{21} m$



Simulation par l'Institut Max Planck
 $\sim 10^{25} m$

Comment simuler efficacement des systèmes gravitationnels ?

- 1 Mise en place et contraintes
 - Cadre
 - Discrétisation
 - Performance
- 2 Algorithme de Barnes-Hut parallélisé
 - Principe
 - Implémentation parallélisée
 - Procédure de mise à jour
- 3 Application
 - Théorème du viriel
 - Effondrement et énergie

Mise en place et contraintes

Cadre

- Donnée d'un système gravitationnel à simuler : à l'échelle d'une galaxie, d'un amas de galaxie, de l'univers observable...
- Utilisation de corps *virtuels*, de masse identique.
- En termes de nombre caractéristique de corps :

$$1 \ll N_{\text{simulé}} \ll N_{\text{réel}}$$

Cadre

- Donnée d'un système gravitationnel à simuler : à l'échelle d'une galaxie, d'un amas de galaxie, de l'univers observable...
- Utilisation de corps *virtuels*, de masse identique.
- En termes de nombre caractéristique de corps :

$$1 \ll N_{\text{simulé}} \ll N_{\text{réel}}$$

Cadre

- Donnée d'un système gravitationnel à simuler : à l'échelle d'une galaxie, d'un amas de galaxie, de l'univers observable...
- Utilisation de corps *virtuels*, de masse identique.
- En termes de nombre caractéristique de corps :

$$1 \ll N_{\text{simulé}} \ll N_{\text{réel}}$$

Modèle ponctuel

- Potentiel et force d'interaction gravitationnelle dans le modèle ponctuel :

$$\Phi(r) = -\mathcal{G} \frac{m}{r}$$

$$\vec{F}_{i \rightarrow j} = -\mathcal{G} \frac{m_i m_j}{\|\vec{P}_i \vec{P}_j\|^3} \vec{P}_i \vec{P}_j$$

- Inadapté à la simulation de corps *virtuels* : accélérations irréalistes.
 Nécessité d'*adoucir* le potentiel.

Modèle ponctuel

- Potentiel et force d'interaction gravitationnelle dans le modèle ponctuel :

$$\Phi(r) = -\mathcal{G} \frac{m}{r}$$

$$\vec{F}_{i \rightarrow j} = -\mathcal{G} \frac{m_i m_j}{\|\vec{P}_i \vec{P}_j\|^3} \vec{P}_i \vec{P}_j$$

- Inadapté à la simulation de corps *virtuels* : accélérations irréalistes.
 Nécessité d'*adoucir* le potentiel.

Modèle de Plummer

- Potentiel et force dans le modèle de *Plummer* :

$$\Phi_P(r) = -\mathcal{G} \frac{m}{\sqrt{r^2 + \epsilon^2}}$$

$$\vec{F}_{i \rightarrow j} = -\mathcal{G} \frac{m_i m_j}{(\|\vec{P}_i \vec{P}_j\|^2 + \epsilon^2)^{3/2}} \vec{P}_i \vec{P}_j$$

avec ϵ longueur caractéristique du modèle

Discrétisation des équations

- Système de $2N$ équations différentielles vectorielles, soit $2Nd$ équations différentielles scalaires :

$$\left\{ \begin{array}{l} \forall j \in \llbracket 1, N \rrbracket, \frac{d\vec{r}_j}{dt} = \vec{v}_j \\ \forall j \in \llbracket 1, N \rrbracket, m_j \frac{d\vec{v}_j}{dt} = m_j \vec{a}_j = \sum_{\substack{i=1 \\ i \neq j}}^N \vec{F}_{i \rightarrow j} \end{array} \right.$$

- Conservation de l'énergie (intégrateurs *symplectiques*).

Discrétisation des équations

- Système de $2N$ équations différentielles vectorielles, soit $2Nd$ équations différentielles scalaires :

$$\left\{ \begin{array}{l} \forall j \in \llbracket 1, N \rrbracket, \frac{d\vec{r}_j}{dt} = \vec{v}_j \\ \forall j \in \llbracket 1, N \rrbracket, m_j \frac{d\vec{v}_j}{dt} = m_j \vec{a}_j = \sum_{\substack{i=1 \\ i \neq j}}^N \vec{F}_{i \rightarrow j} \end{array} \right.$$

- Conservation de l'énergie (intégrateurs *symplectiques*).

Euler ou Verlet ?

Schéma d'Euler

$$\begin{cases} \vec{v}_{n+1} = \vec{v}_n + \tau \vec{a}_n \\ \vec{r}_{n+1} = \vec{r}_n + \tau \vec{v}_n \end{cases}$$

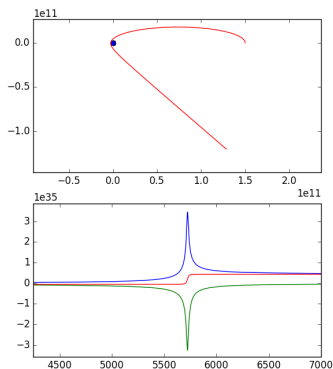
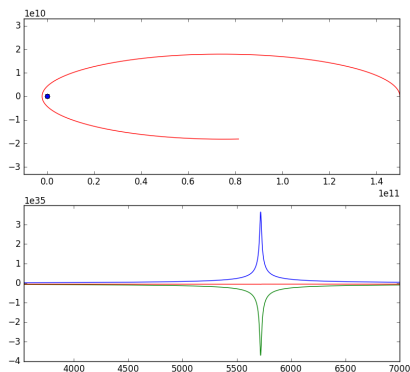


Schéma de Verlet

$$\begin{cases} \vec{r}_{n+1} = \vec{r}_n + \tau \vec{v}_n + \frac{\tau^2}{2} \vec{a}_n \\ \vec{v}_{n+1} = \vec{v}_n + \frac{\tau}{2} (\vec{a}_n + \vec{a}_{n+1}) \end{cases}$$



Algorithme naïf

- Calculer l'interaction entre chaque paire de particule :

$$\binom{N}{2} = \frac{N(N-1)}{2} = \Theta(N^2)$$

En pratique, inutilisable dès 10^4 ou 10^5 corps

- Pistes d'amélioration :
 - Effectuer des approximations pour améliorer asymptotiquement la complexité temporelle
 - Paralléliser les calculs

Algorithme naïf

- Calculer l'interaction entre chaque paire de particule :

$$\binom{N}{2} = \frac{N(N-1)}{2} = \Theta(N^2)$$

En pratique, inutilisable dès 10^4 ou 10^5 corps

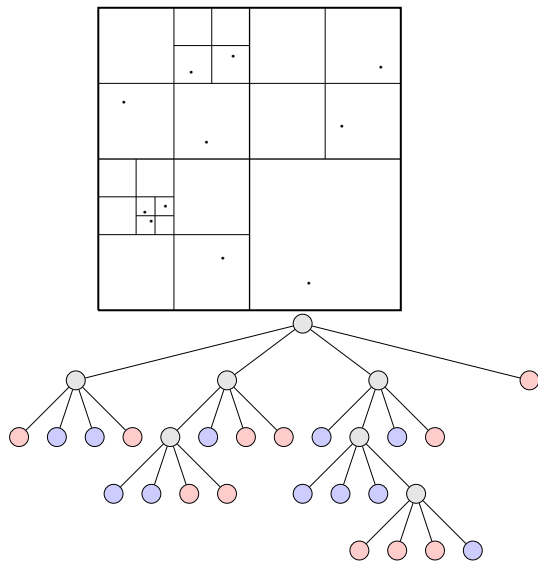
- Pistes d'amélioration :
 - Effectuer des approximations pour améliorer asymptotiquement la complexité temporelle
 - Paralléliser les calculs

Algorithme de Barnes-Hut parallélisé

Josh Barnes et Piet Hut en 1986.

Division récursive du plan :
arbre dont chaque nœud
est d'arité 0 ou 4
(*quadtree*).

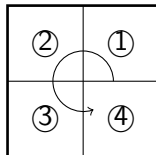
Dans toute feuille, 0 ou 1
corps.



C++ et programmation orientée objet

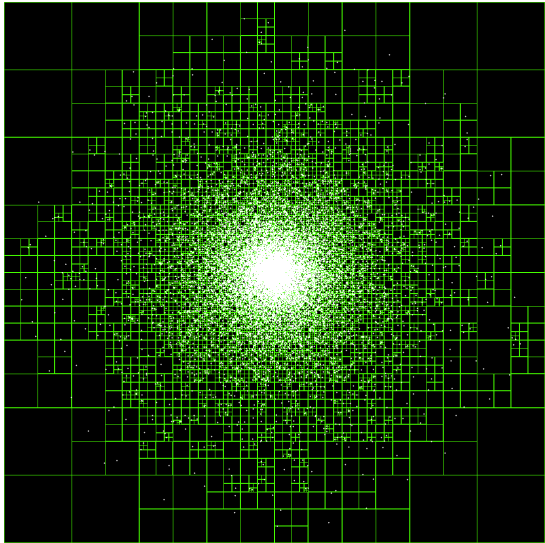
```
class Quadtree
{
public:
    //constructeurs, destructeur
private:
    Quadtree *_f1;
    Quadtree *_f2;
    Quadtree *_f3;
    Quadtree *_f4;

    bool _leaf;
    uint _n;
    uint _nodes;
    float _x1;
    float _x2;
    float _y1;
    float _y2;
    float _xb;
    float _yb;
};
```



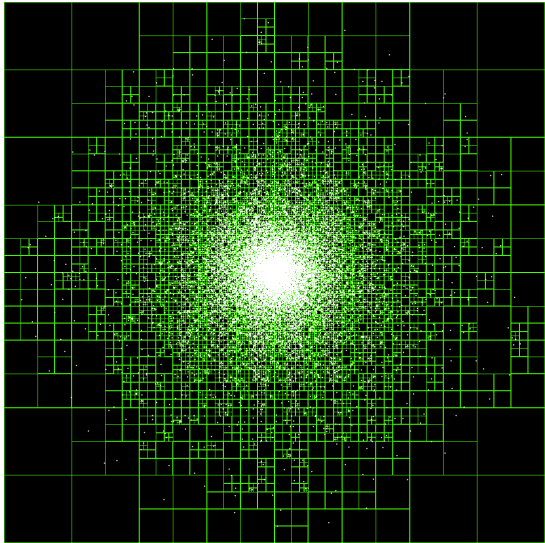
Construction de l'arbre

- Insertion des corps un par un : descente récursive dans l'arbre jusqu'à ce qu'il soit seul dans son nœud.
- Si hauteur en $\mathcal{O}(\ln N)$, insertion en $\mathcal{O}(\ln N)$. Coût total de la construction en $\mathcal{O}(N \ln N)$.



Construction de l'arbre

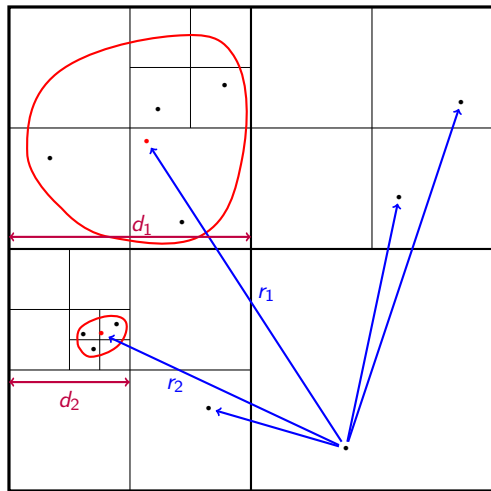
- Insertion des corps un par un : descente récursive dans l'arbre jusqu'à ce qu'il soit seul dans son nœud.
- Si hauteur en $\mathcal{O}(\ln N)$, insertion en $\mathcal{O}(\ln N)$. Coût total de la construction en $\mathcal{O}(N \ln N)$.



Évaluation de force

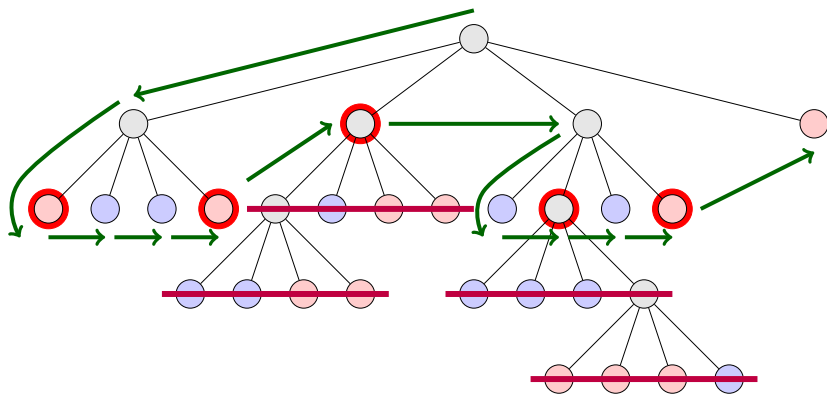
On approxime la force exercée sur un corps par le barycentre d'un nœud dès qu'on a l'inégalité

$$\frac{d}{r} \leq \theta$$



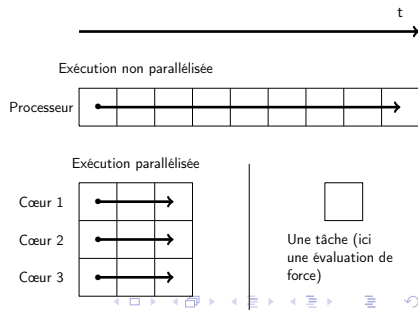
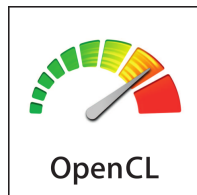
Évaluation de force

Parcours en profondeur *avec sauts*.



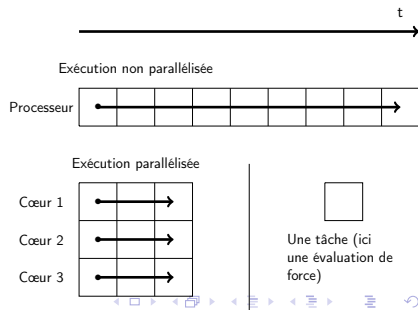
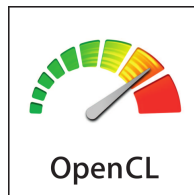
Calcul parallélisé

- Architecture parallélisée : microprocesseurs multicœurs, processeurs graphiques.
- Problème : classes récursives non adaptée à *OpenCL*. Adaptation des algorithmes requise.

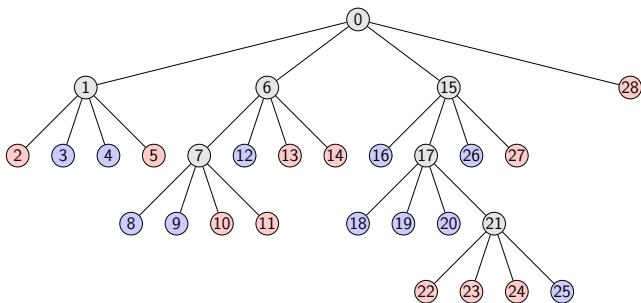


Calcul parallélisé

- Architecture parallélisée : microprocesseurs multicœurs, processeurs graphiques.
- Problème : classes récursives non adaptée à *OpenCL*. Adaptation des algorithmes requise.

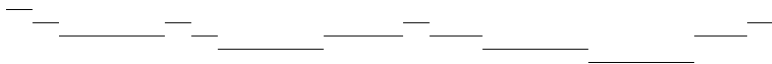
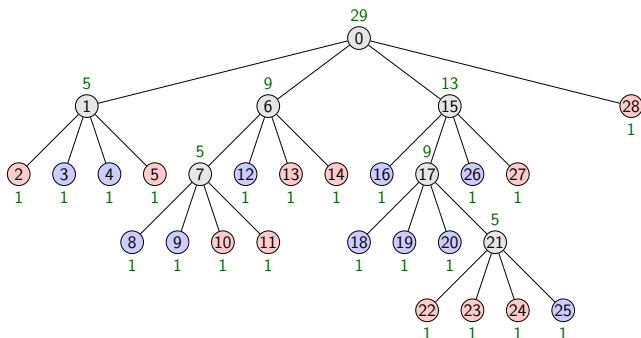


Linéarisation de l'arbre



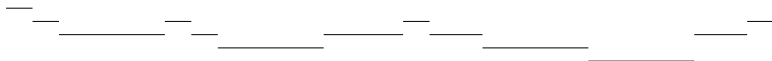
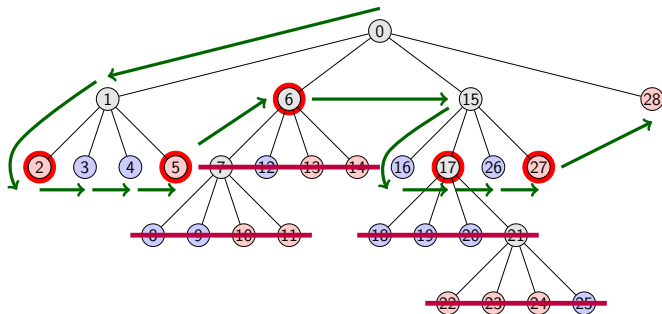
Noeud	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Sauts																													

Linéarisation de l'arbre



Noeud	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	
Sauts	29	5	1	1	1	1	9	5	1	1	1	1	1	1	1	13	1	9	1	1	1	1	5	1	1	1	1	1	1	1

Linéarisation de l'arbre



Nœud	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	
Sauts	29	5	1	1	1	1	9	5	1	1	1	1	1	1	1	13	1	9	1	1	1	1	5	1	1	1	1	1	1	1



- Calcul de l'arbre difficilement parallélisable.
- Exploiter le faible déplacement par rapport au système :
procédure de mise à jour de l'arbre évitant son calcul total.
- En pratique, entre 5 et 10 fois plus rapide.

- Calcul de l'arbre difficilement parallélisable.
- Exploiter le faible déplacement par rapport au système :
procédure de mise à jour de l'arbre évitant son calcul total.
- En pratique, entre 5 et 10 fois plus rapide.

- Calcul de l'arbre difficilement parallélisable.
- Exploiter le faible déplacement par rapport au système :
procédure de mise à jour de l'arbre évitant son calcul total.
- En pratique, entre 5 et 10 fois plus rapide.

Application

Théorème du viriel en mécanique

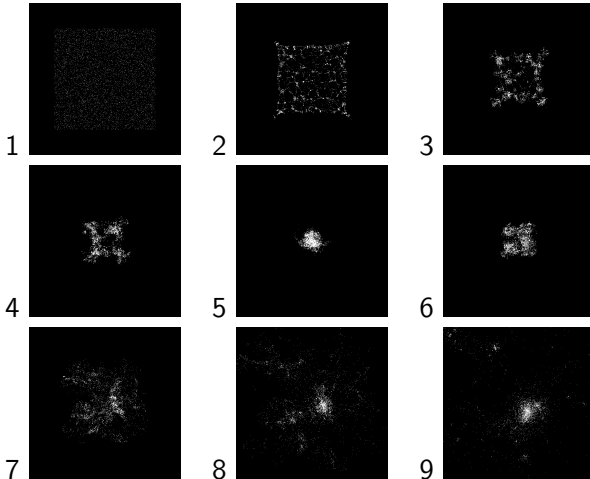
Si I est le moment d'inertie du système, alors

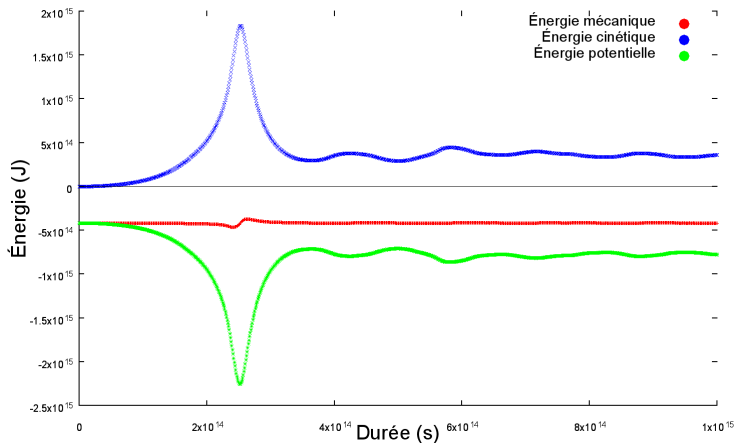
$$2K + U = \frac{d^2 I}{dt^2}$$

En particulier, pour un système à l'équilibre dynamique, on a la relation du viriel

$$2K + U = 0$$

Simulation à partir d'une distribution initiale statique et uniforme sur un carré :





À la dernière étape $-\frac{U}{K} \simeq 2.09$

Conclusion