

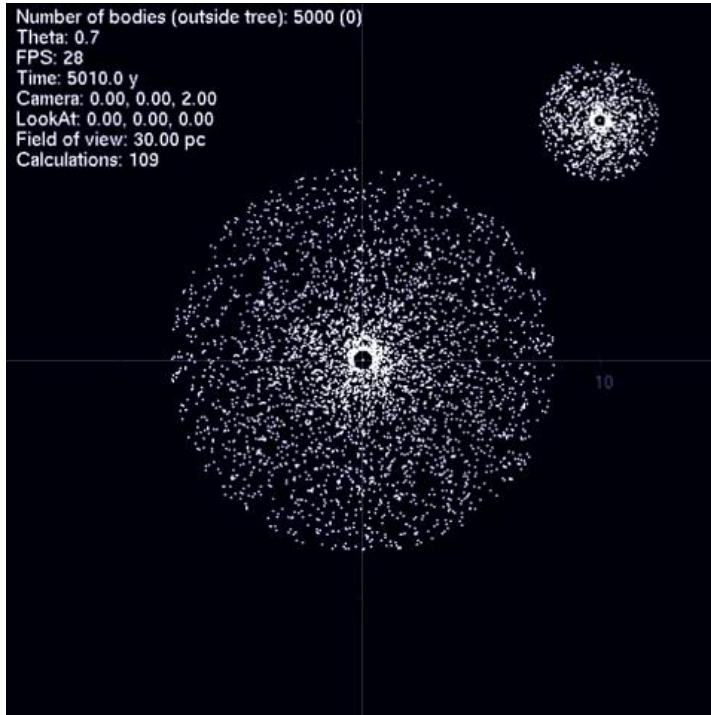
# Modèles de particules et modèles à n-corps

<https://www.coursera.org/learn/modeling-simulation-natural-processes/>

# Aperçu

- Aujourd'hui: systèmes comprenant un grand nombre de particules, corps stellaires, ou autres objets.
- Equations de Newton pour la mécanique classique et leur résolution.
- Différents types de potentiels: corps stellaires, dynamique moléculaire.
- Complexité algorithmique des schémas présentées.  
Astuces et schémas de résolution.

# Exemple 1: problème à n-corps

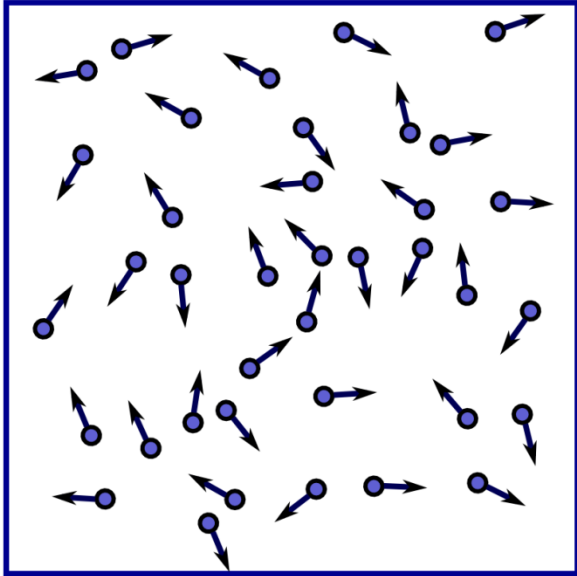


- Dans ce problème, on essaye de prédire le mouvement individuel d'un grand nombre de corps célestes.
- Dans l'exemple à gauche, ces corps sont les étoiles de deux galaxies entrant en collision.
- Il y a un tel nombre d'étoiles qu'on les modélise simplement comme des points (position et vitesse) qui interagissent par les lois de la gravité.

# Modélisation par des points

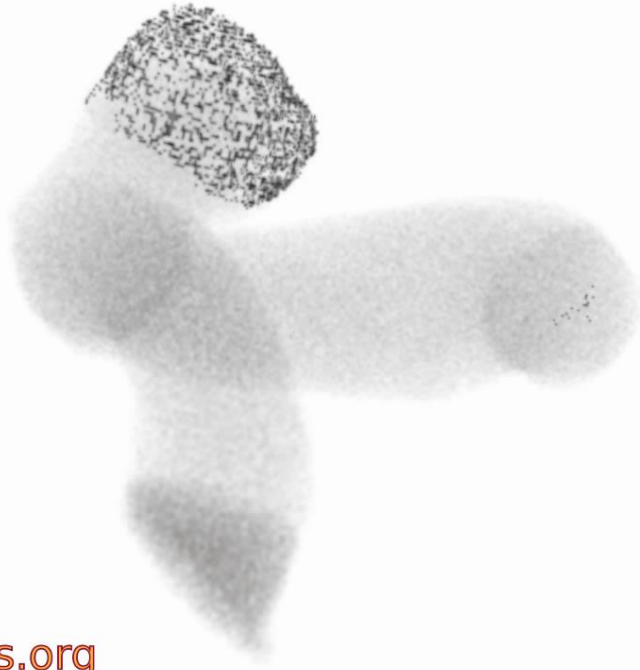
- Dans cette leçon, nous considérons différents modèles consistant d'un grand nombre d'objets avec des propriétés similaires.
- Ces objets interagissent par des forces classiques, Newtoniennes, qui sont connues.
- Afin d'accélérer le calcul, nous représentons ces objets par des simples points, décrits par leur position et vitesse, et potentiellement d'autres propriétés simples.

# Exemple 2: molécules dans un gaz



- Ici, nous modélisons un gaz (ou en tout cas une petite portion) en simulant l'interaction entre des molécules de gaz individuelles.
- Les molécules n'interagissent qu'avec d'autres molécules dans un voisinage immédiat.
- La plupart du temps, elles avancent le long de lignes droites à travers l'espace vide.
- Mais lorsqu'elles s'approchent d'autres molécules, les forces d'interaction prennent de l'importance, et des collisions ont lieu, durant lesquelles les molécules changent leur direction et vitesse.

# Exemple 3: globules rouges



- Dans ce cas, c'est les globules rouges dans le sang humain que l'on modélise par de simple points.
- Les globules rouges sont immergés dans l'écoulement. En plus de l'interaction particule-particule, un couplage fluide-particule est intégré dans le modèle.
- Des globules rouges réels possèdent des propriétés complexes. Ici, celles-ci sont fortement simplifiées afin de pouvoir en calculer un grand nombre (environ un milliard).

# Lois de Newton, forces et potentiels

# Lois de Newton

<b>Première loi:</b>	Dans un repère inertiel, un corps reste au repos ou se propage à une vitesse constante, à moins qu'il subisse l'influence d'une force extérieure.
<b>Deuxième loi:</b>	La somme vectorielle des forces $\mathbf{F}$ agissant sur un corps est égale à la masse $m$ de ce corps multipliée par l'accélération vectorielle du corps: $\mathbf{F} = m\mathbf{a}$ .
<b>Third law:</b>	Lorsqu'un corps exerce une force sur un deuxième corps, le deuxième corps exerce simultanément une force égale en amplitude, mais de direction opposée, sur le premier corps.



# Lois de Newton

<b>Première loi:</b>	Dans un repère inertiel, un corps reste au repos ou se propage à une vitesse constante, à moins qu'il subisse l'influence d'une force extérieure.
<b>Deuxième loi:</b>	La somme vectorielle des forces $\mathbf{F}$ agissant sur un corps est égale à la masse $m$ de ce corps multipliée par l'accélération vectorielle du corps: $\mathbf{F} = m\mathbf{a}$ .
<b>Third law:</b>	Lorsqu'un corps exerce une force sur un deuxième corps, le deuxième corps exerce simultanément une force égale en amplitude, mais de direction opposée, sur le premier corps.

# Lois de Newton

<b>Première loi:</b>	Dans un repère inertiel, un corps reste au repos ou se propage à une vitesse constante, à moins qu'il subisse l'influence d'une force extérieure.
<b>Deuxième loi:</b>	La somme vectorielle des forces $\mathbf{F}$ agissant sur un corps est égale à la masse $m$ de ce corps multipliée par l'accélération vectorielle du corps: $\mathbf{F} = m\mathbf{a}$ .
<b>Third law:</b>	Lorsqu'un corps exerce une force sur un deuxième corps, le deuxième corps exerce simultanément une force égale en amplitude, mais de direction opposée, sur le premier corps.

# Forces

- Dans les modèles présents, nous nous concentrons sur les forces issues de l'interaction réciproque entre objets (alors qu'en général, d'autres types de force peuvent avoir lieu).
- Nous nous restreignons à la **mécanique Newtonienne**. Lorsqu'une interaction est non-classique (exemple: dynamique moléculaire), elle est approximée par une force de type Newtonienne.
- En particulier, l'interaction totale est décomposée en une somme d'interaction entre des **paires d'objets**.

# Exemple: force gravitationnelle

- Nous considérons des interactions entre paires d'objets (ici: corps stellaires).
- Une force gravitationnelle est dirigée vers le corps qui exerce la force, et diminue avec le carré de la distance entre les corps:

$$\mathbf{F}_{ij} = \frac{G m_i m_j (\mathbf{x}_j - \mathbf{x}_i)}{|\mathbf{x}_j - \mathbf{x}_i|^3}$$

- La force totale est la somme des contributions venant de tous les autres corps:

$$\mathbf{F}_i = \sum_{j=1, j \neq i}^N \frac{G m_i m_j (\mathbf{x}_j - \mathbf{x}_i)}{|\mathbf{x}_j - \mathbf{x}_i|^3}$$

# Force vs. potentiel

- Lorsqu'une force est conservatrice, elle peut être dérivée d'un potentiel:

$$F_i = \frac{\partial V}{\partial x_i}$$

- Il est souvent plus intéressant d'étudier le potentiel, plutôt que la force, pour comprendre la nature de l'interaction entre objets. Le potentiel gravitationnel est:

$$V = \sum_{1 \leq i < j \leq N} \frac{G m_i m_j}{|x_j - x_i|}$$

# Deuxième exemple: loi de Coulomb

- La loi de Coulomb en carré-inversé décrit l'interaction électrostatique entre deux particules chargées:

$$\mathbf{F}_{ij} = \frac{q_i q_j}{4\pi\epsilon_0} \frac{(\mathbf{x}_j - \mathbf{x}_i)}{|\mathbf{x}_j - \mathbf{x}_i|^3}$$

- Il s'agit du même type de force que pour la gravité, avec d'autres constantes.
- La force de Coulomb peut être attractive ou répulsive, dépendant du signe des charges.

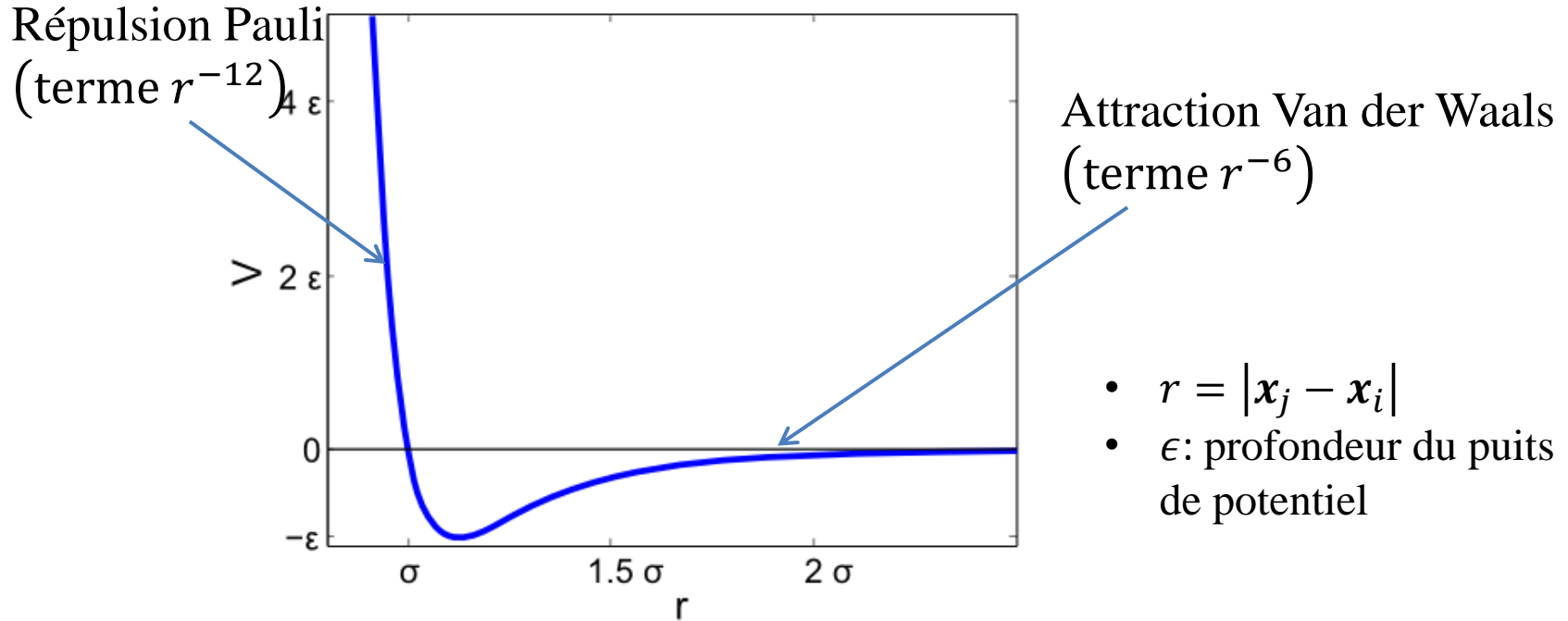
# 3<sup>ème</sup> exple.: potentiel de Lennard-Jones

- Le potentiel de Lennard-Jones est un modèle simple pour l'interaction entre des paires d'atomes ou molécules à charge neutre:

$$V_{ij} = 4\epsilon \left\{ \left( \frac{\sigma}{|\mathbf{x}_i - \mathbf{x}_j|} \right)^{12} - \left( \frac{\sigma}{|\mathbf{x}_i - \mathbf{x}_j|} \right)^6 \right\}$$

- Le terme en puissance de 12 correspond à la répulsion de Pauli.
- Le terme en puissance de 6 est une attraction, agissant sur une distance moyenne (ni courte ni longue): la force de van der Waals.

# Potentiel de Lennard-Jones





# Résumé: nos trois potentiels

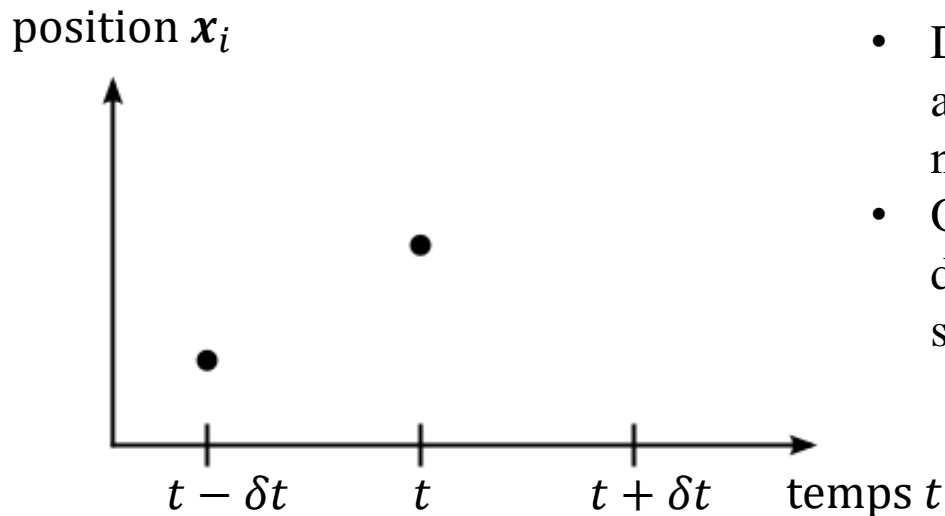
Force	Comportement à grande distance de la force	Commentaire
Gravité	$r^{-2}$	Force de longue portée.
Coulomb	$r^{-2}$	Force de longue portée.
Lennard-Jones	$r^{-7}$	Force de courte portée: tombe rapidement à zéro.

# Intégration temporelle des équations du mouvement

# Principes de l'intégration temporelle

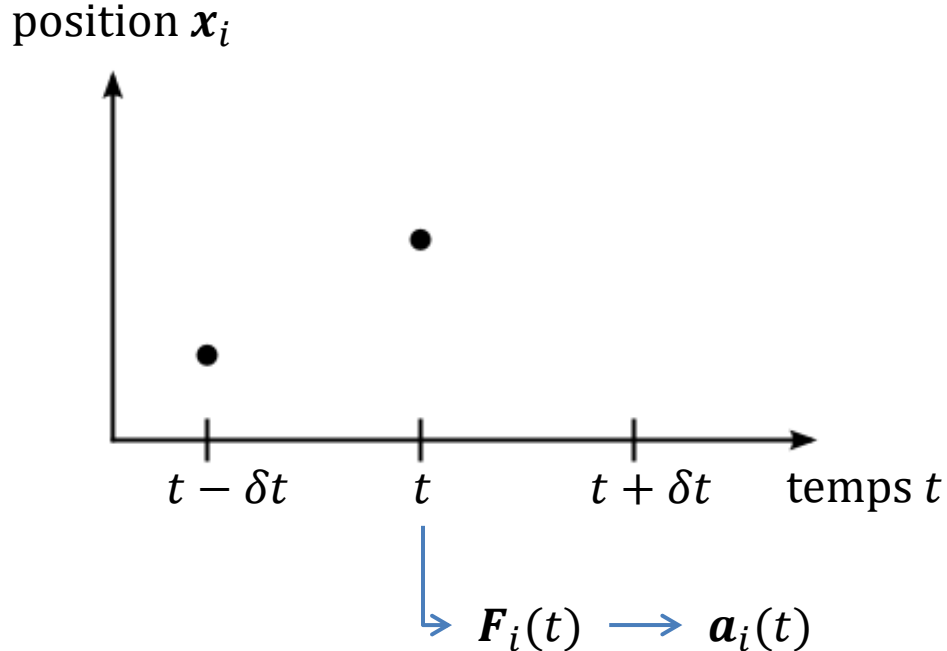
- Nous discrétisons l'axe temporel en intervalles  $\delta t$ . Une itération du schéma numérique transporte le système du temps  $t$  au temps  $t + \delta t$ .
- A chaque itération, l'état du système est entièrement décrit par
  - la position  $\mathbf{x}_i$  de chacun des objets.
  - la vitesse  $\mathbf{v}_i$  de chacun des objets.
- Nous allons maintenant introduire l'algorithme de Verlet qui est précis et efficace.

# Version originale de l'algo de Verlet



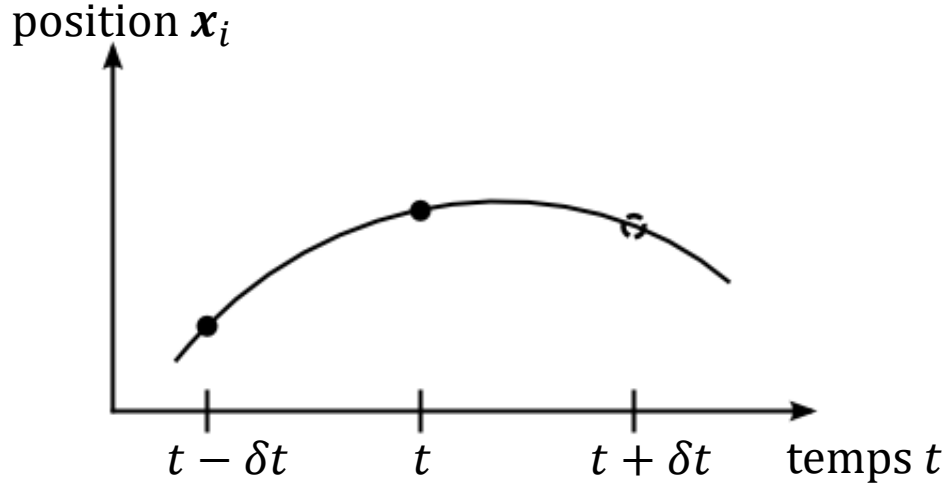
- Dans la version originale de cet algorithme, la vitesse  $v_i$  des objets n'est pas gardée en mémoire.
- On ne répertorie que les positions  $x_i$  des objets à deux pas de temps successifs,  $t - \delta t$  et  $t$ .

# Version originale de l'algo de Verlet



- Dans la version originale de cet algorithme, la vitesse  $\mathbf{v}_i$  des objets n'est pas gardée en mémoire.
- On ne répertorie que les positions  $\mathbf{x}_i$  des objets à deux pas de temps successifs,  $t - \delta t$  et  $t$ .
- A partir de la position de tous les objets au temps  $t$ , nous calculons la force  $\mathbf{F}_i(t)$ , et à partir de la loi de Newton, l'accélération  $\mathbf{a}_i(t)$  sur le  $i$ -ème objet.

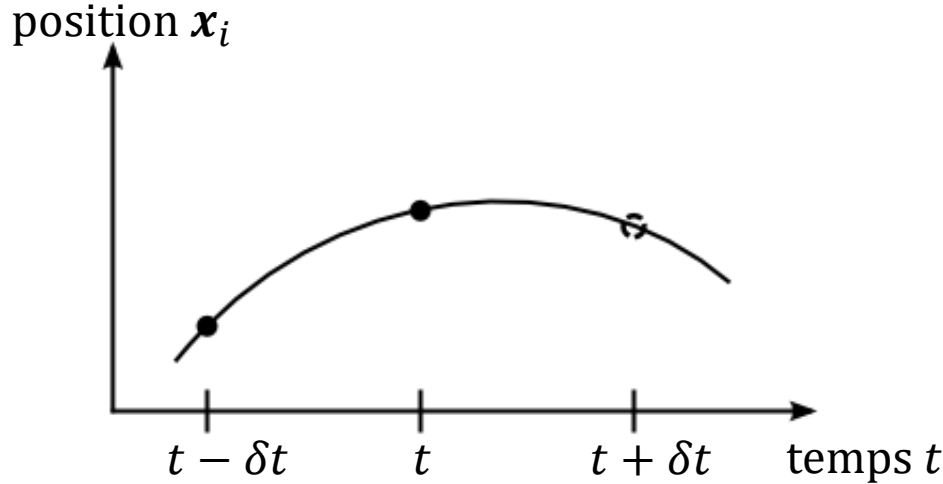
# Version originale de l'algo de Verlet



Par des différences finies, l'accélération et la position sont mis en relation avec une précision du deuxième ordre:

$$a_i(t) \approx \frac{1}{\delta t^2} (x_i(t + \delta t) - 2x_i(t) + x_i(t - \delta t))$$

# Version originale de l'algo de Verlet



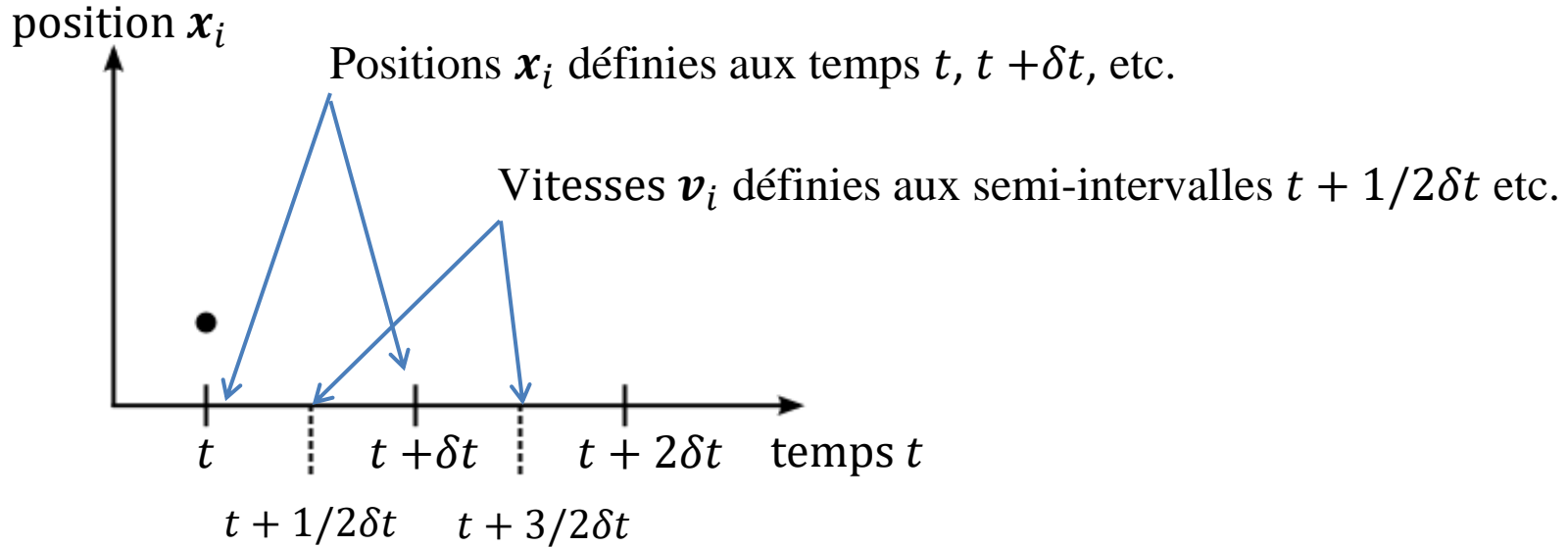
Cette équation est résolue pour obtenir la nouvelle position  $x_i(t + \delta t)$ :

$$a_i(t) \approx \frac{1}{\delta t^2} (x_i(t + \delta t) - x_i(t) + x_i(t - \delta t))$$

$$x_i(t + \delta t) \approx \delta t^2 a_i(t) + 2x_i(t) + x_i(t - \delta t)$$

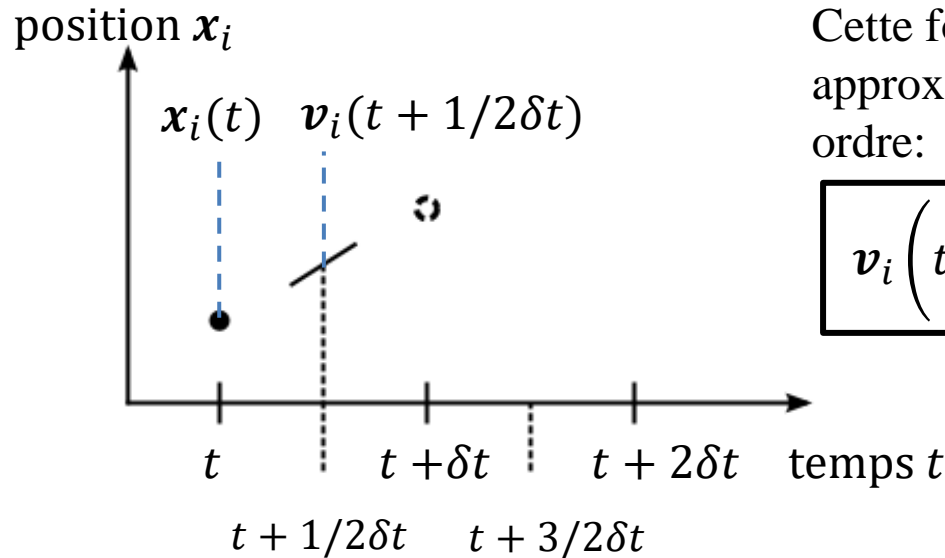
# Algorithme de Verlet («leap-frog»)

Idée:





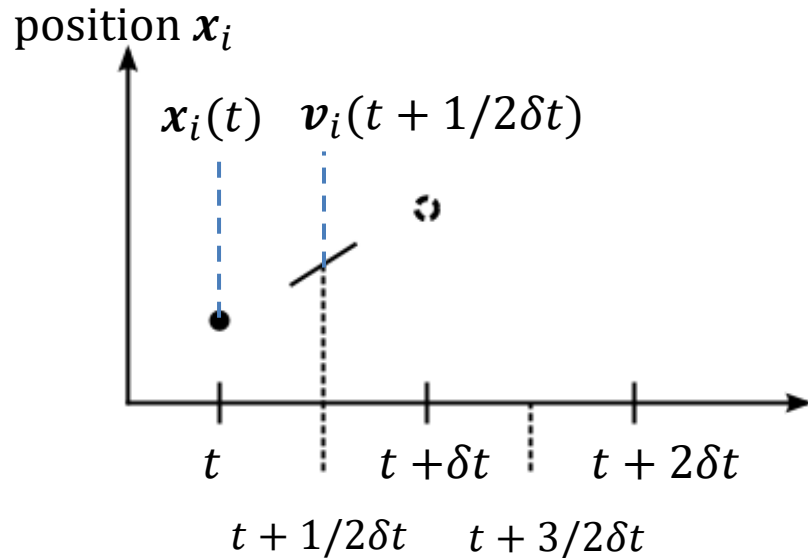
# Algorithme de Verlet («leap-frog»)



Cette fois-ci, c'est la vitesse qui est approximée par un schéma du deuxième ordre:

$$v_i\left(t + \frac{1}{2\delta t}\right) \approx \frac{1}{\delta t} (x_i(t + \delta t) - x_i(t))$$

# Algorithme de Verlet («leap-frog»)



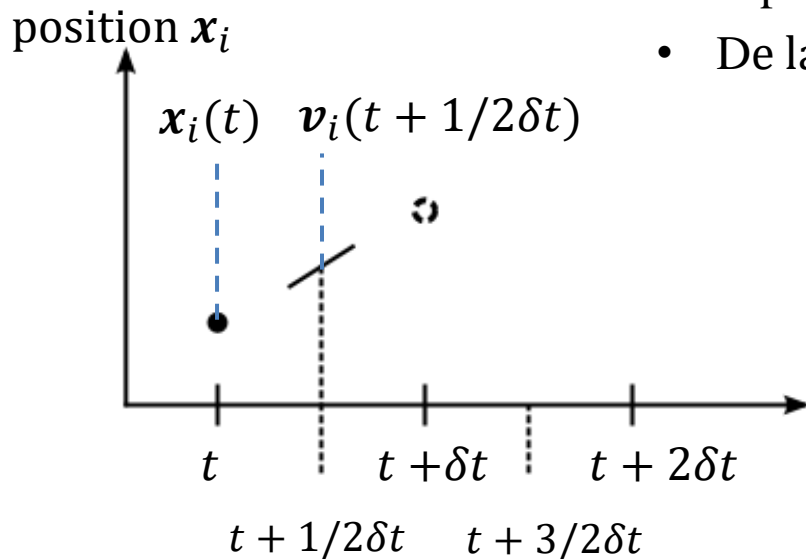
Et on obtient  $x_i(t + \delta t)$ :

$$v_i\left(t + \frac{1}{2}\delta t\right) \approx \frac{1}{\delta t}(x_i(t + \delta t) - x_i(t))$$

$$x_i(t + \delta t) \approx x_i(t) + \delta t v_i\left(t + \frac{1}{2}\delta t\right)$$

# Algorithme de Verlet («leap-frog»)

- A partir des  $\mathbf{x}_i(t + \delta t)$ , on trouve  $\mathbf{a}_i(t + \delta t)$ .
- De la même manière on calcule  $\mathbf{v}_i\left(t + \frac{3}{2}\delta t\right)$ :



$$\mathbf{x}_i(t + \delta t) \approx \mathbf{x}_i(t) + \delta t \mathbf{v}_i\left(t + \frac{1}{2}\delta t\right)$$

$$\mathbf{v}_i\left(t + \frac{3}{2}\delta t\right) \approx \mathbf{v}_i\left(t + \frac{1}{2}\delta t\right) + \delta t \mathbf{a}_i(t + \delta t)$$

# Mais comment calculer les forces?

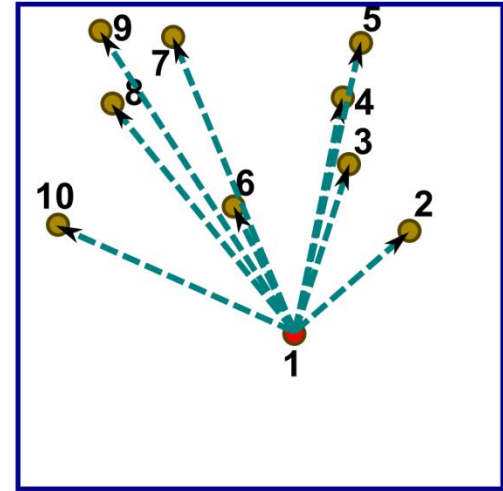
# Complexité d'un calcul de forces

Pour simuler un système à  $N$  objets, il faut:

- Calculer l'évolution temporelle de manière itérative.
- A chaque itération, calculer la force agissant sur chaque objet, et donc, calculer  $N$  forces.
- Pour chacune des  $N$  forces, calculer la contribution des  $N-1$  autres objets.

Nombre total d'opérations:  $N(N - 1)$

Sans simplification, le calcul de la force pour une itération possède une complexité d'ordre  $N^2$ .



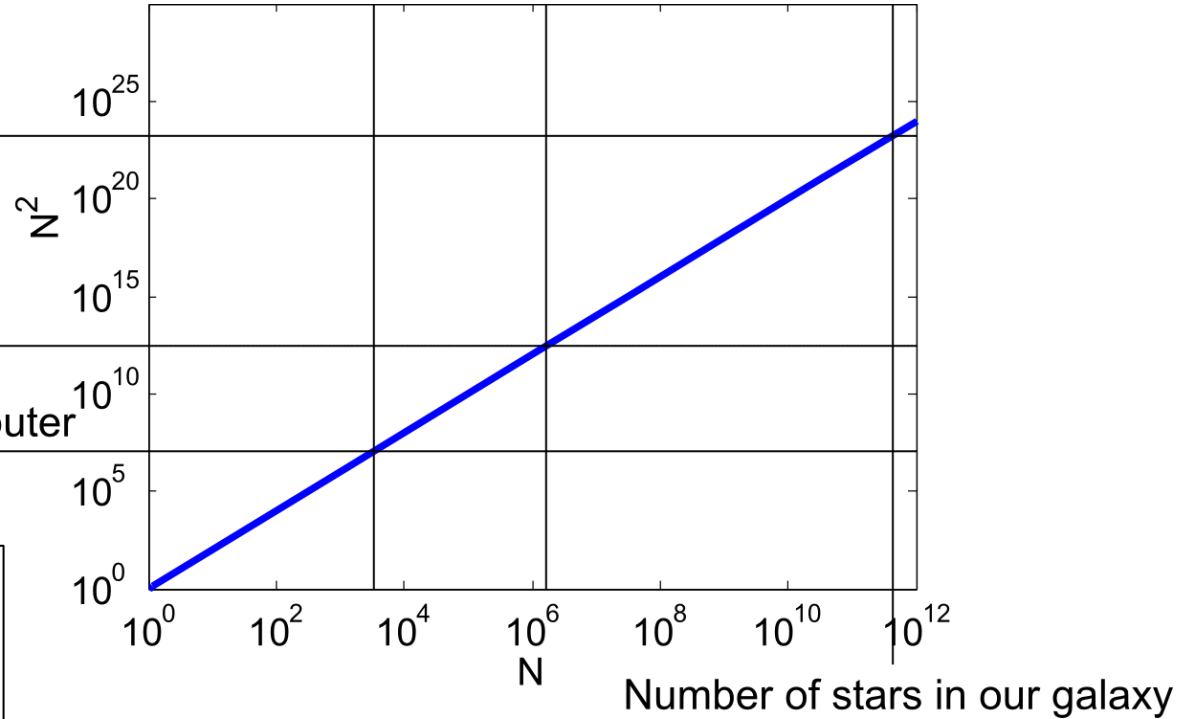
( $N-1$ ) forces agissant sur un objet

# Peut-on calculer $N^2$ interactions en une seconde?

Requires 100 billion  
supercomputers

1 second on world's fastest  
supercomputer

1 second on a desktop computer



*Il s'agit d'estimations. La valeur effective dépend du type de potentiel, ainsi que de la structure de données utilisées dans l'algorithme.*

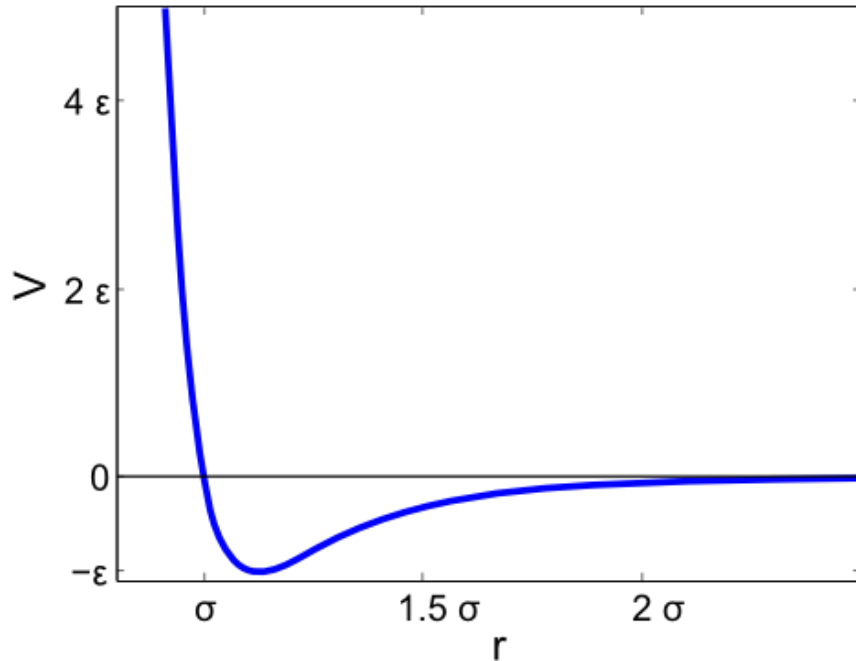
# Complexité algorithmique

- La complexité algorithmique liée à la résolution d'un problème à  $N$  particules/corps par force brute est de  $N^2$ .
- Il est alors impossible de simuler un système contenant plus d'un million d'objets, même sur un superordinateur.
- Il nous faut des stratégies pour réduire le nombre d'opérations effectuées.

# Le potentiel Lennard-Jones: distance de cutoff

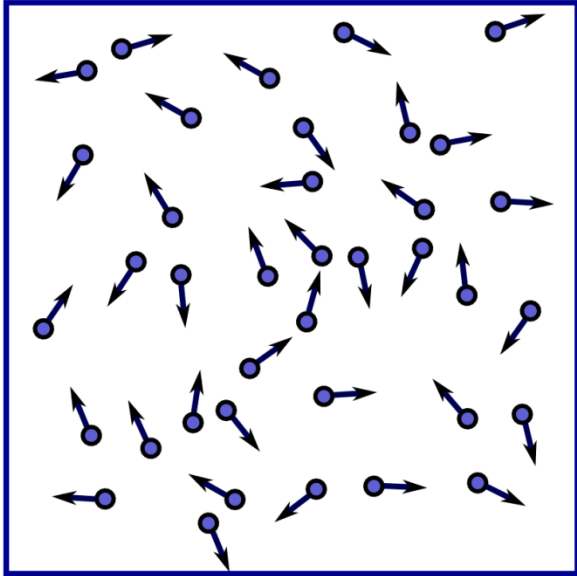


# Retour au potentiel de Lennard-Jones



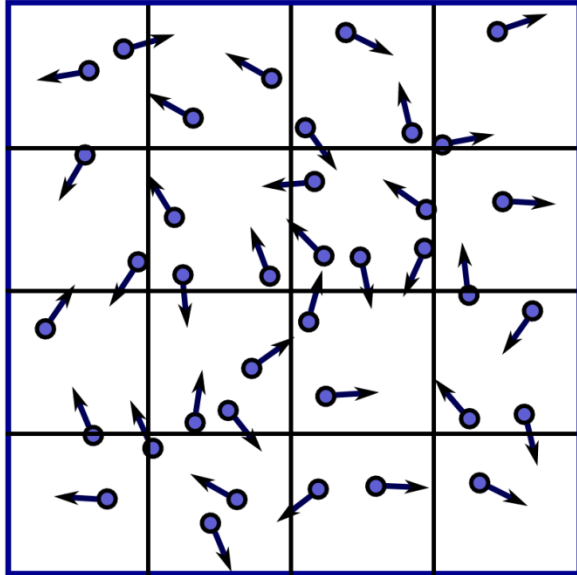
- Le potentiel tombe rapidement à zéro, puisqu'à longue distance il évolue en  $r^{-6}$ .
- A une distance de  $d_c = 2.5\sigma$ , il ne vaut plus qu'à-peu-près 1% de  $\epsilon$ .
- Idée: on utilise  $d_c$ , ou une autre distance, comme **distance de cut-off**, et on néglige toutes les particules se trouvant au-delà de cette distance, pour accélérer le calcul.

# Comment identifier les particules proches?



- S'il fallait tester toutes les particules pour vérifier si elles se trouvent en-dessous de la distance de cut-off, il faudrait effectuer  $N$  tests par particules: en terme de complexité algorithmique, on n'y gagnerait rien.

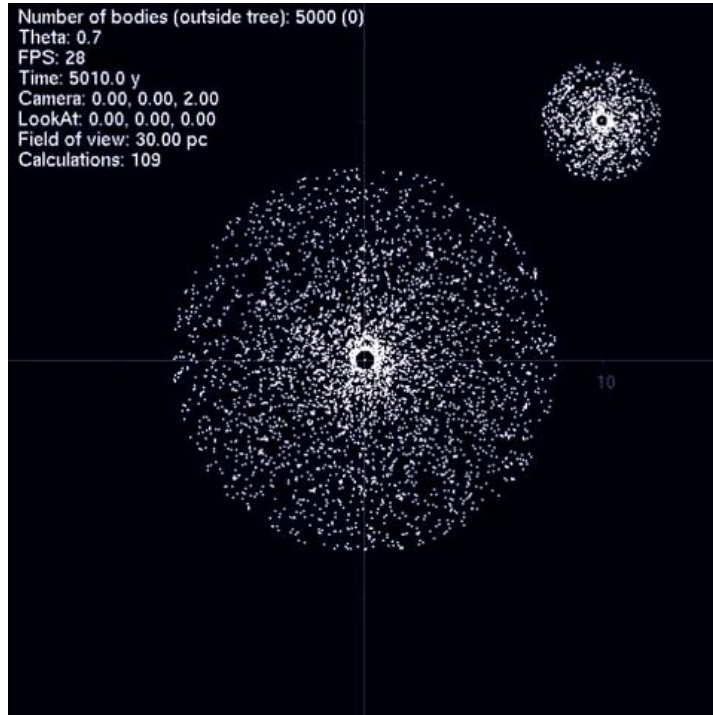
# Comment identifier les particules proches?



- S'il fallait tester toutes les particules pour vérifier si elles se trouvent en-dessous de la distance de cut-off, il faudrait effectuer  $N$  tests par particules: en terme de complexité algorithmique, on n'y gagnerait rien.
- Solution: méthode de grille. On superpose une grille à maillage égal à l'espace des particules.
- Les particules «s'attachent» à la grille, ce qui permet de les retrouver facilement.

# Problème à N-corps: Evaluation des forces de gravité

# Problème à N-corps

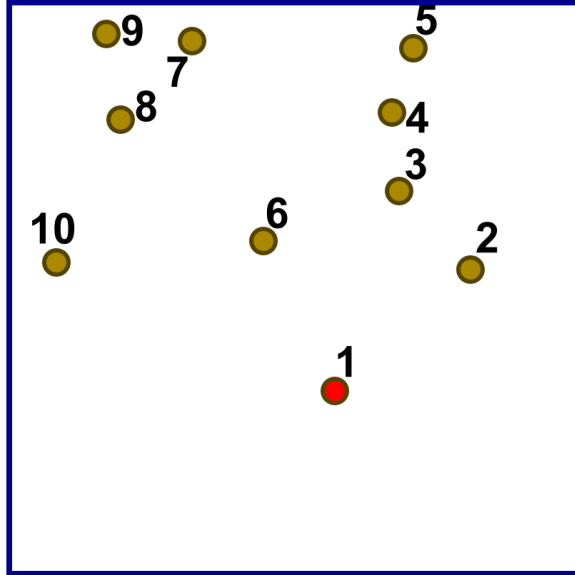


- Dans le problème à N-corps, on considère les forces gravitationnelles entre tous les objets (ici: des étoiles).
- Les forces agissent à longue distance: Les deux galaxies effectuent une rotation autour de leur centre de masse commun.
- Une distance de cut-off ne serait pas utile dans ce cas-là: il faut tenir compte de toutes les interactions.

# Simplification: méthodes d'arbre

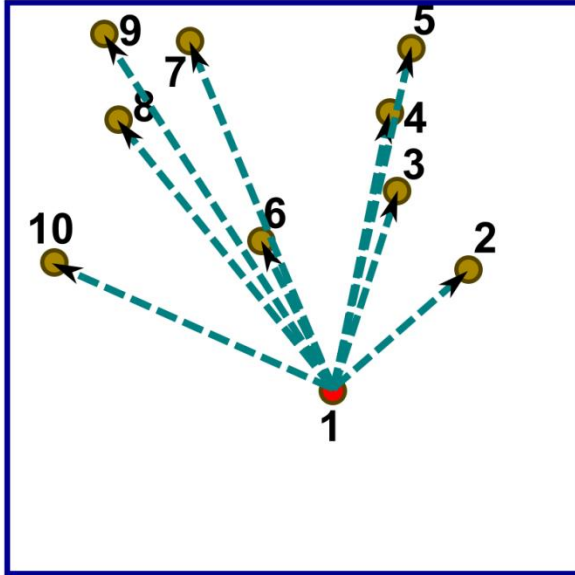
- Idée: si un groupe de corps est suffisamment lointain, on le modélise comme un corps unique, situé en son centre de masse.
- Exemple: afin de simuler l'interaction entre un grand nombre de galaxies, on pourrait considérer chaque galaxie comme un corps unique, au lieu de simuler toutes ses étoiles.
- Dans les méthodes d'arbre, cette idée est généralisée, et les corps sont regroupés de manière dynamique afin d'accélérer le calcul.

# Exemple: interaction entre 10 corps



- On souhaite calculer la force totale agissant sur le corps 1.

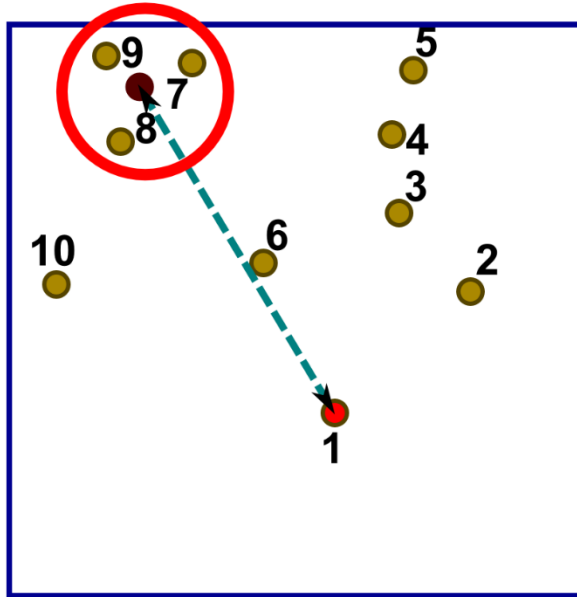
# Exemple: interaction entre 10 corps



- On souhaite calculer la force totale agissant sur le corps 1.
- Résultat exact: calcul et somme des 9 forces venant des 9 autres corps.

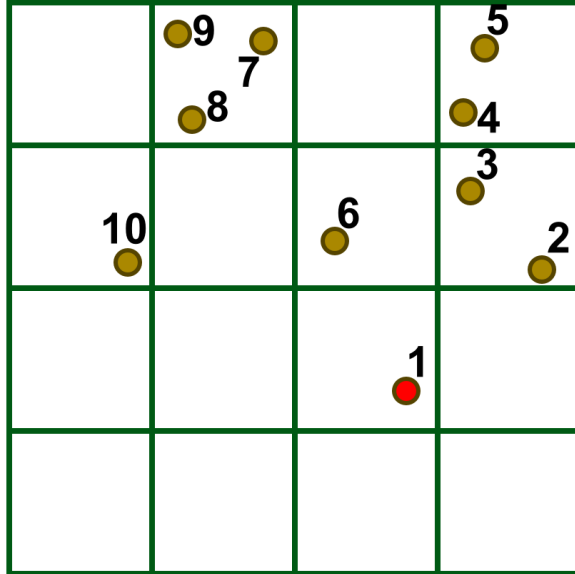


# Réduction d'un groupe à un seul corps



- Idée: on accélère le calcul en traitant un groupe comme un seul corps, situé au centre de masse du groupe.
- Cette approximation est raisonnable si le groupe est suffisamment lointain.

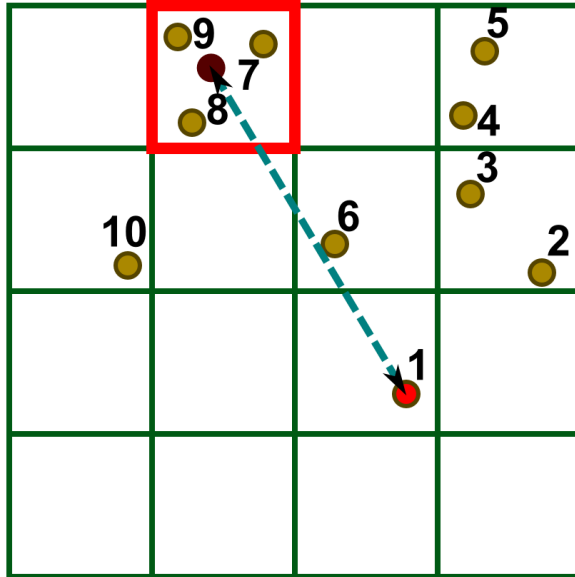
# Comment définir ces groupes?



## Premier essai: grille régulière

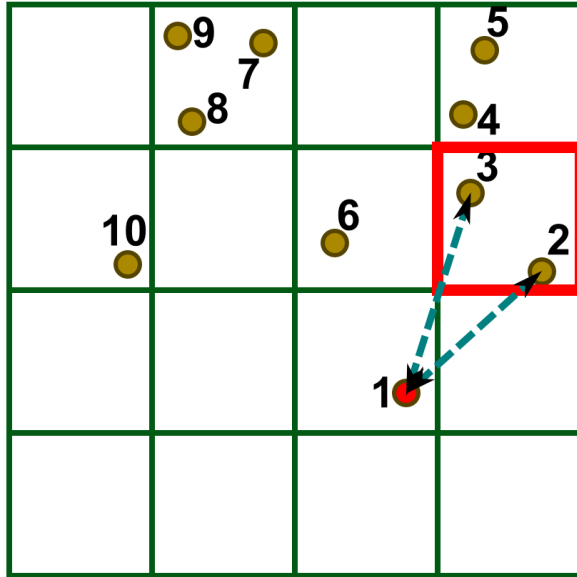
- Subdivision de l'espace en cellules de taille égale.
- Une cellule = un groupe.

# Comment définir ces groupes?



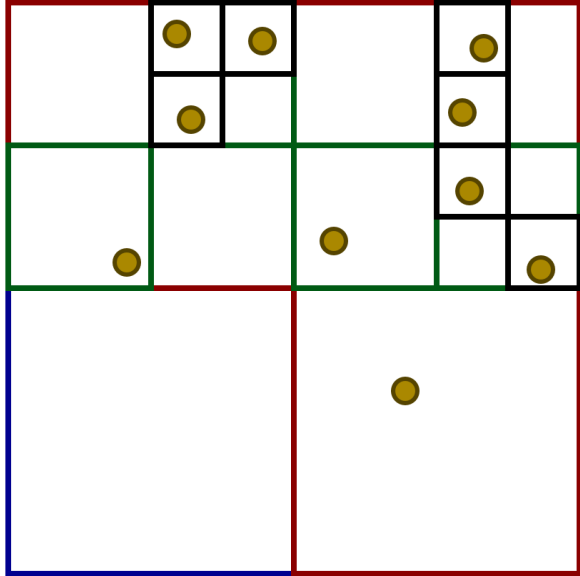
- Pour certains corps, cette stratégie est parfaite.

# Comment définir ces groupes?



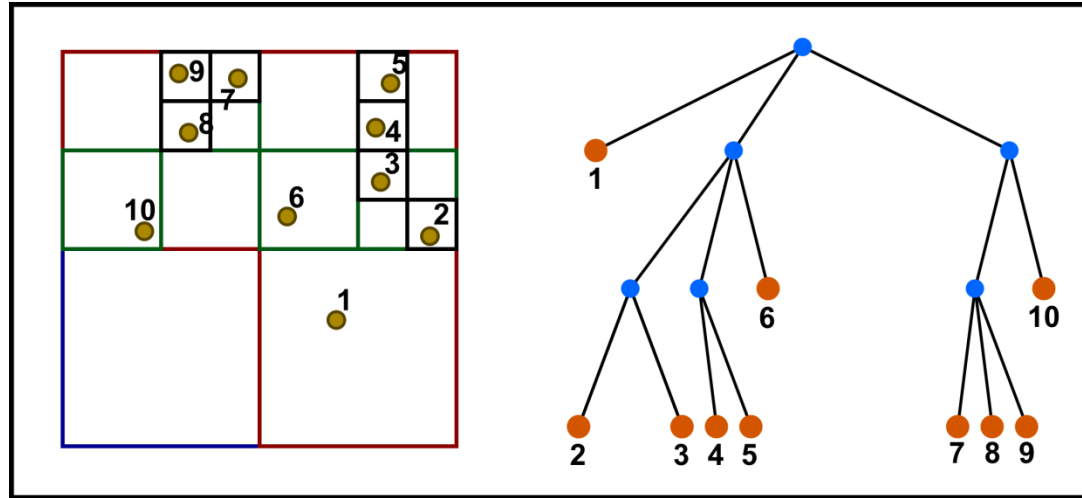
- Pour certains corps, cette stratégie est parfaite.
- Mais pour d'autres, la taille de la cellule est inadéquate.
- Les corps proches nécessitent de petits groupes.
- Les corps lointains nécessitent de grands groupes.

# Solution: algorithmes d'arbre



- L'espace est sous-divisé récursivement en des cellules de plus en plus petites.
- Pour les corps proches, on utilisera les petites cellules, et autrement les grandes cellules.
- L'algorithme de **Barnes-Hut**, que nous présentons ici, est un grand classique.

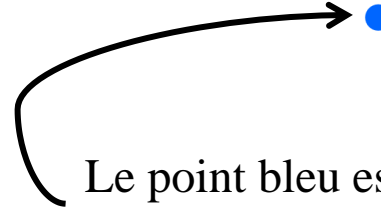
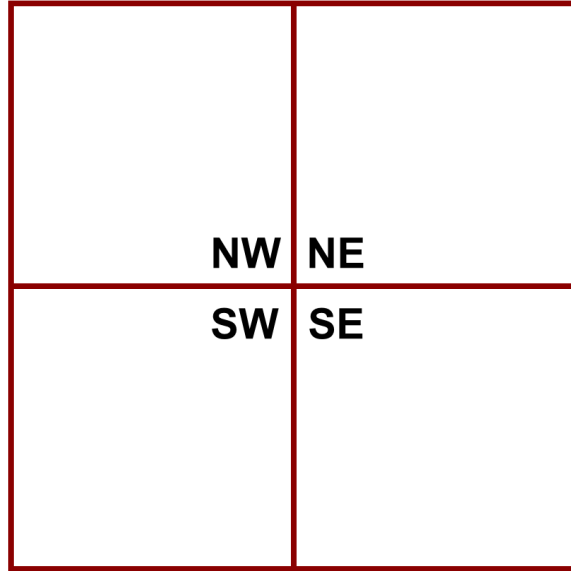
# Algorithme de Barnes-Hut



L'espace est sous-divisé de manière réursive en **quadrants** (en 3D: **octants**).

La relation entre les quadrants est représentée par une structure en arbre, le **quad-tree** (en 3D: **octree**).

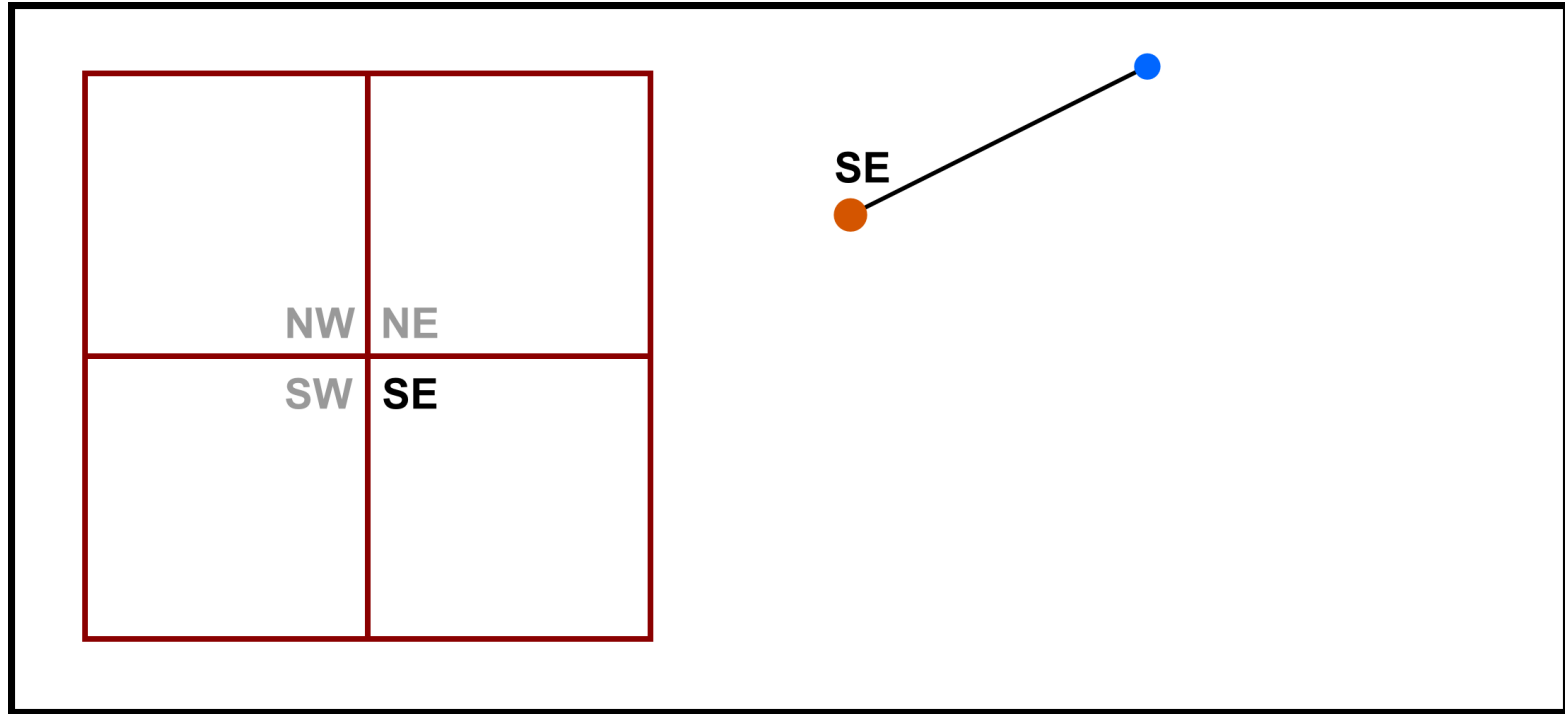
# Nommage des quadrants



Le point bleu est la racine de l'arbre. Il représente tout l'espace.

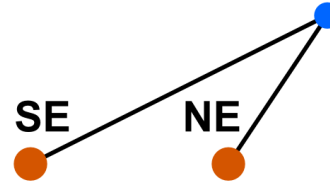
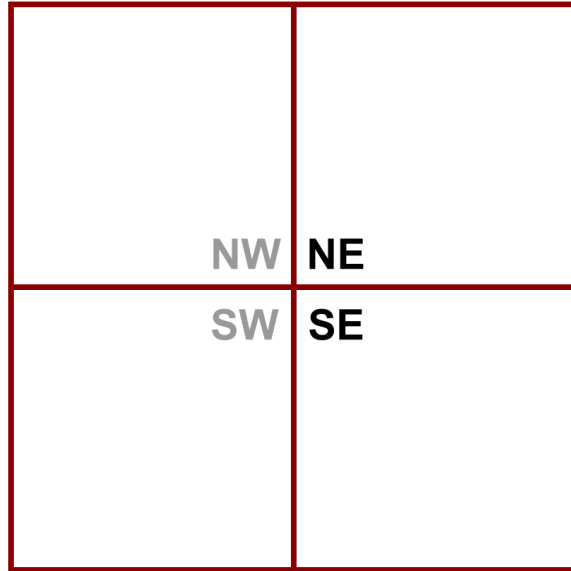
- SE = South-East
- NE = North-East
- SW = South-West
- NW = North-West

# Quadrant Sud-Est

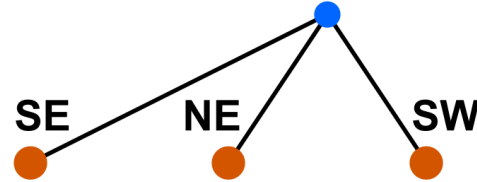
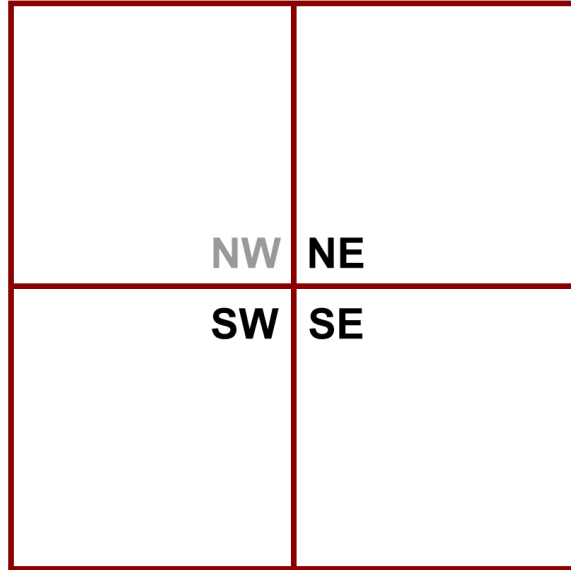




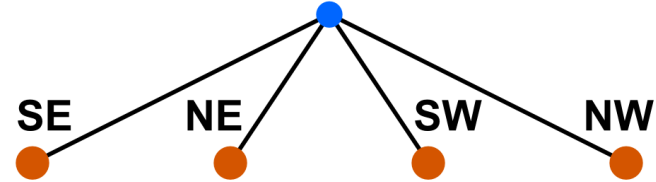
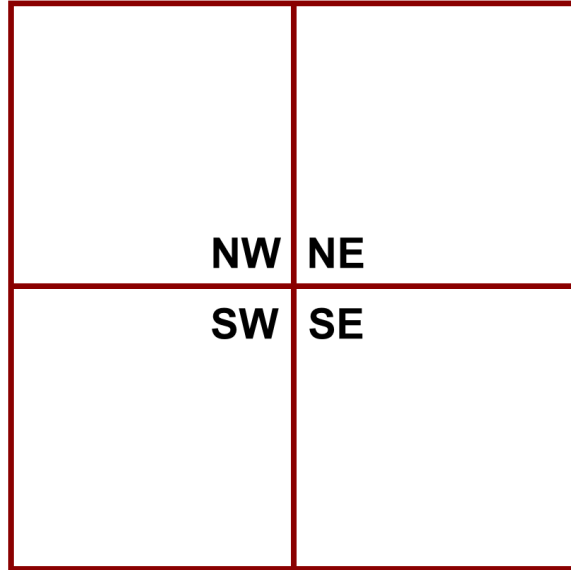
# Quadrant Nord-Est



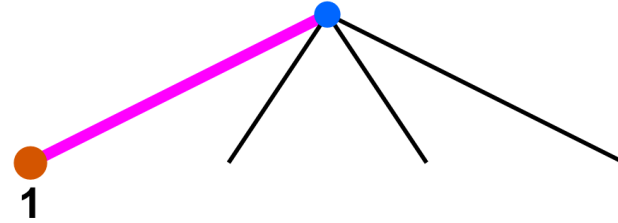
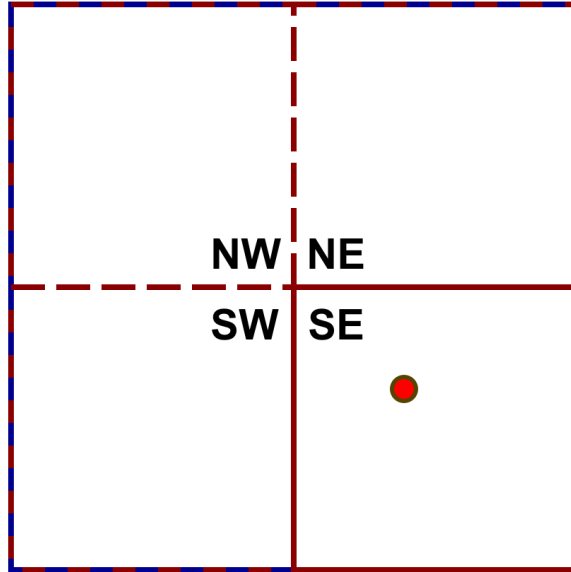
# Quadrant Sud-Ouest



# Quadrant Nord-Ouest

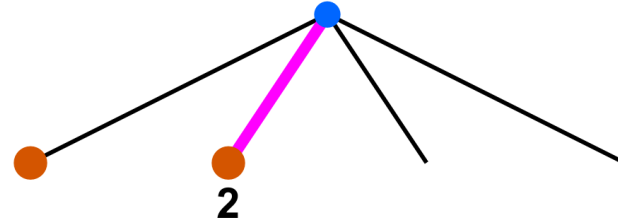
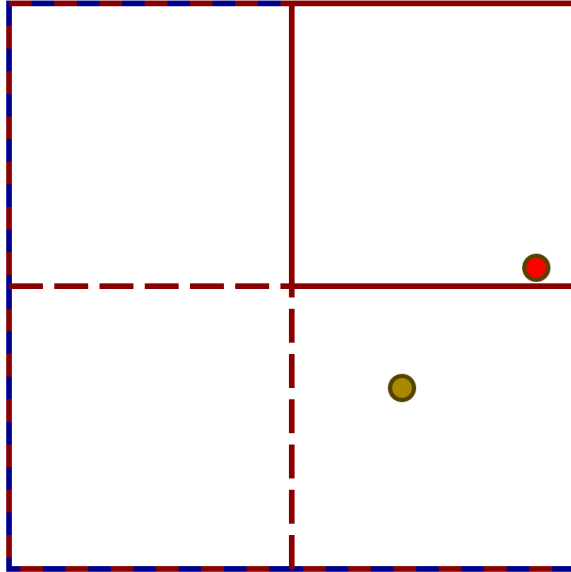


# Insertion du premier corps



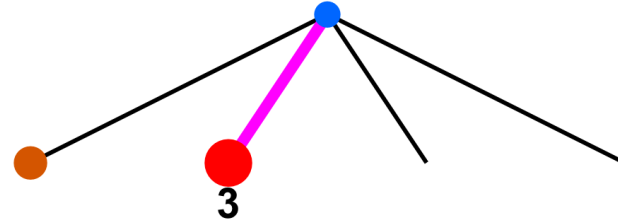
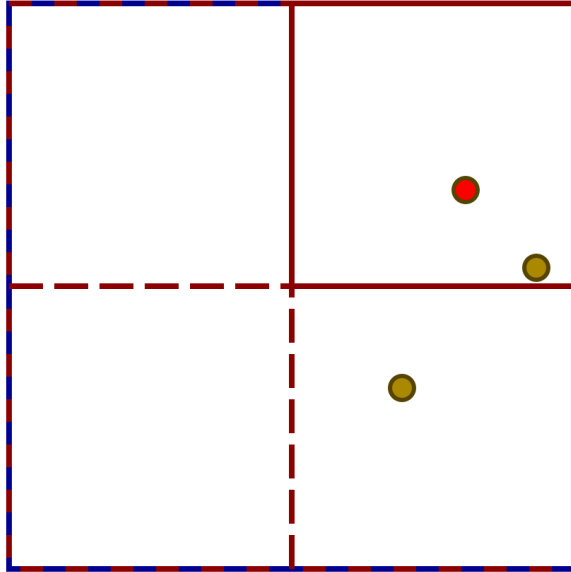
- Il s'agit du deuxième nœud de l'arbre.
- Il est placé dans la branche Sud-Est de la racine.
- Il s'agit d'un **nœud extérieur**: il ne possède pas de sous-branches.

# Insertion du deuxième corps



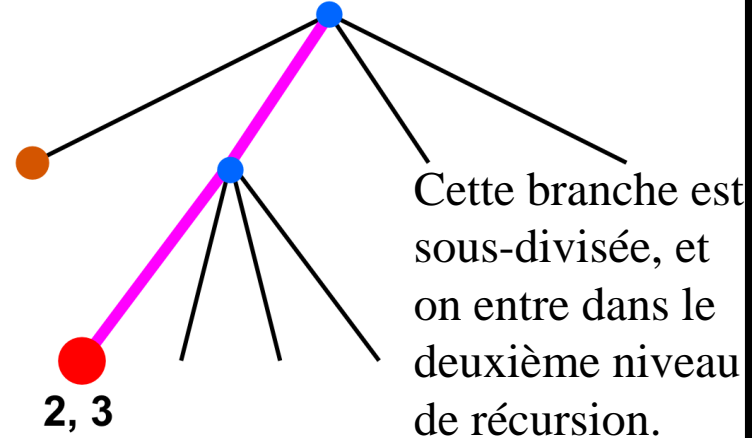
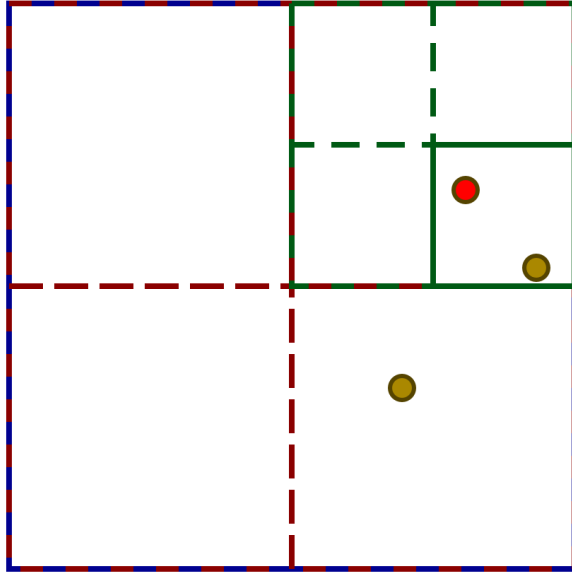
- Le deuxième corps se situe dans le quadrant Nord-Est.
- On le place donc dans la branche Nord-Est de la racine.

# Insertion du troisième corps



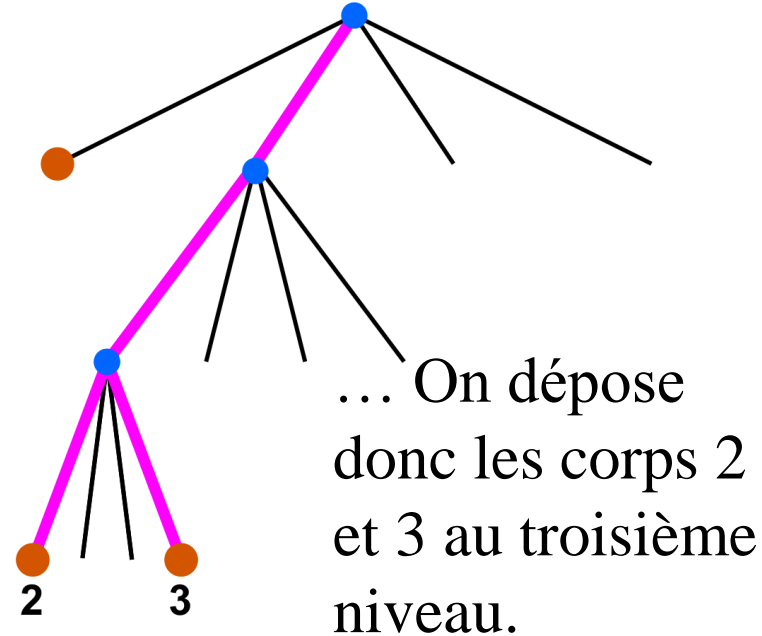
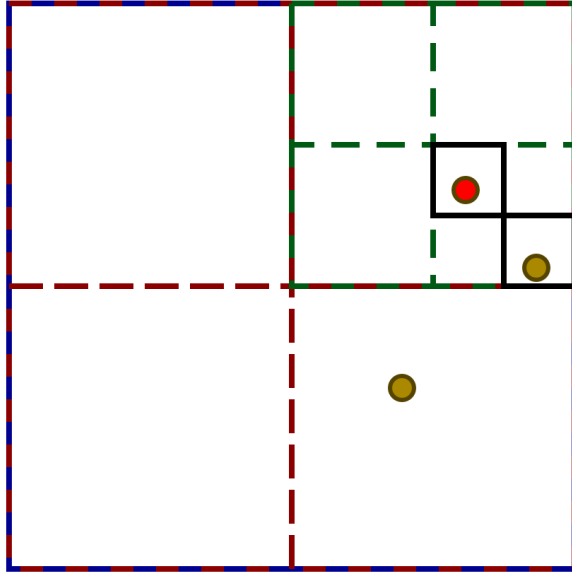
- Le troisième corps se trouve aussi dans le quadrant Nord-Est.
- Mais le nœud correspondant est déjà occupé par le corps 2.
- Les nœuds extérieurs ne peuvent avoir qu'un seul corps: il y a un conflit.

# Insertion du troisième corps



Mais au deuxième niveau, les corps 2 et 3 se trouvent à nouveau dans le même quadrant: le conflit n'est pas résolu.

# Insertion du troisième corps





# Propriétés d'un noeud: masse et centre-de-masse

- Les nœuds internes créés lors de l'insertion du troisième corps représentent des cellules avec, jusqu'à présent, deux corps (cellule rouge au premier niveau, cellule verte au deuxième niveau de récursion).
- Ces deux cellules possèdent le même centre de masse, qu'on va mémoriser au niveau du nœud pour usage futur:

$$\begin{aligned} m &= m_1 + m_2 \\ \mathbf{x} &= \frac{(\mathbf{x}_1 m_1 + \mathbf{x}_2 m_2)}{m} \end{aligned}$$

# Insertion d'un corps: algorithme

Insertion d'un corps  $b$  dans un nœud  $n$  par récursion, en commençant par la racine.

1.	Si le noeud $n$ ne contient pas de corps, on y dépose le nouveau corps $b$ .
2.	Si le noeud $n$ est un noeud interne (donc, possède des branches), on met à jour la masse et le centre-de-masse de $n$ . Le corps $b$ est inséré récursivement dans le bon quadrant.
3.	Si le noeud $n$ est externe, alors le nouveau corps $b$ est en conflit avec un autre corps qui s'y trouve déjà. On sous-divise alors la région en créant 4 branches. On met à jour le centre-de-masse et la masse totale de $n$ . Ensuite, on insère les deux corps dans le ou les quadrants appropriés, puis procède par récursion jusqu'à ce qu'il n'y ait plus de conflit.

# Insertion d'un corps: algorithme

Insertion d'un corps  $b$  dans un nœud  $n$  par récursion, en commençant par la racine.

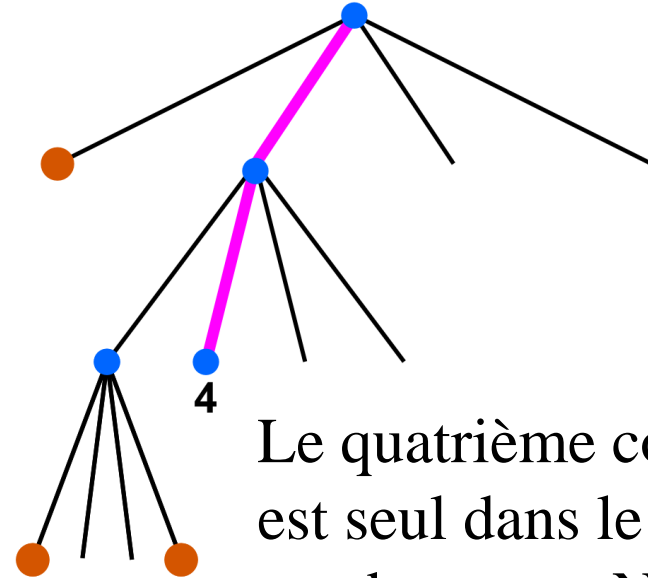
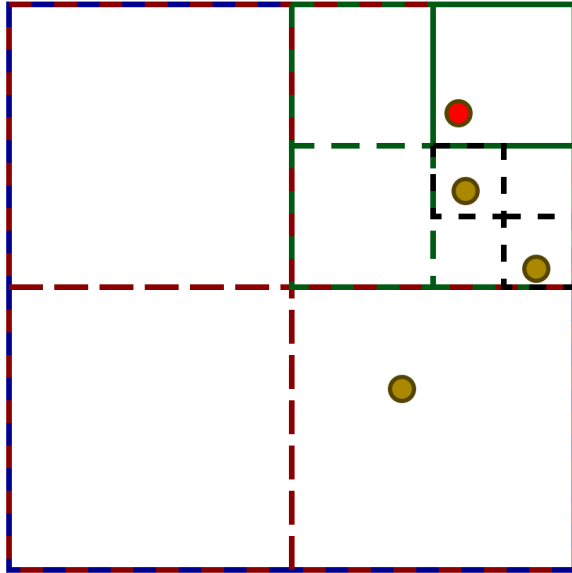
1.	Si le noeud $n$ ne contient pas de corps, on y dépose le nouveau corps $b$ .
2.	Si le noeud $n$ est un noeud interne (donc, possède des branches), on met à jour la masse et le centre-de-masse de $n$ . Le corps $b$ est inséré récursivement dans le bon quadrant.
3.	Si le noeud $n$ est externe, alors le nouveau corps $b$ est en conflit avec un autre corps qui s'y trouve déjà. On sous-divise alors la région en créant 4 branches. On met à jour le centre-de-masse et la masse totale de $n$ . Ensuite, on insère les deux corps dans le ou les quadrants appropriés, puis procède par récursion jusqu'à ce qu'il n'y ait plus de conflit.

# Insertion d'un corps: algorithme

Insertion d'un corps  $b$  dans un nœud  $n$  par récursion, en commençant par la racine.

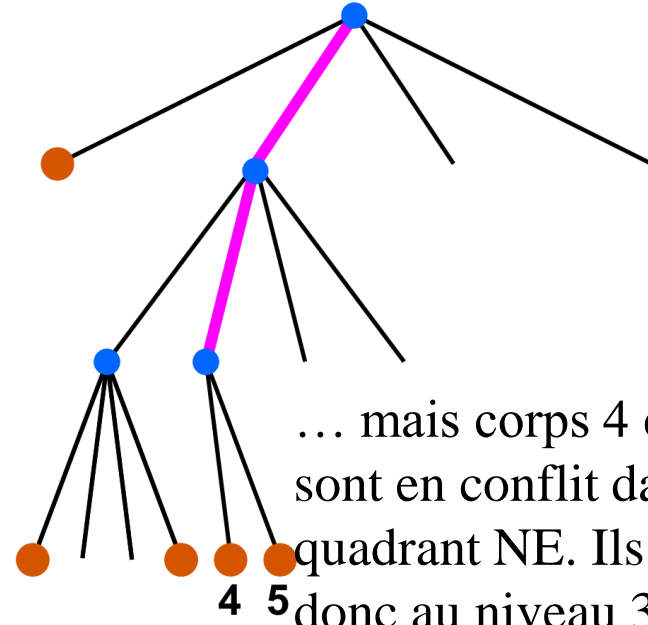
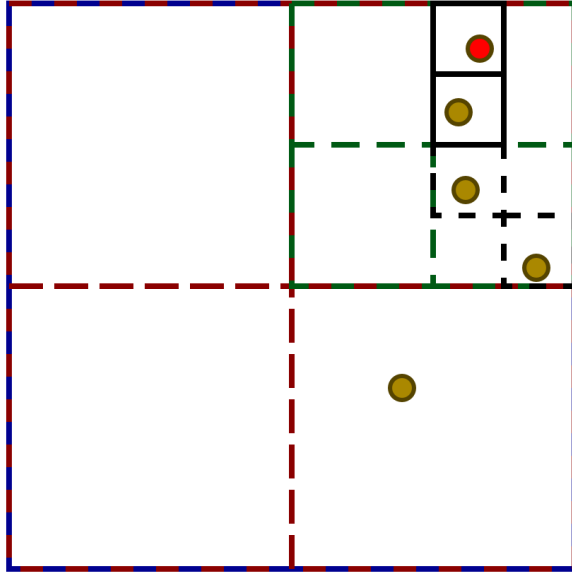
1.	Si le noeud $n$ ne contient pas de corps, on y dépose le nouveau corps $b$ .
2.	Si le noeud $n$ est un noeud interne (donc, possède des branches), on met à jour la masse et le centre-de-masse de $n$ . Le corps $b$ est inséré récursivement dans le bon quadrant.
3.	Si le noeud $n$ est externe, alors le nouveau corps $b$ est en conflit avec un autre corps qui s'y trouve déjà. On sous-divise alors la région en créant 4 branches. On met à jour le centre-de-masse et la masse totale de $n$ . Ensuite, on insère les deux corps dans le ou les quadrants appropriés, puis procède par récursion jusqu'à ce qu'il n'y ait plus de conflit.

# Insertion du quatrième corps



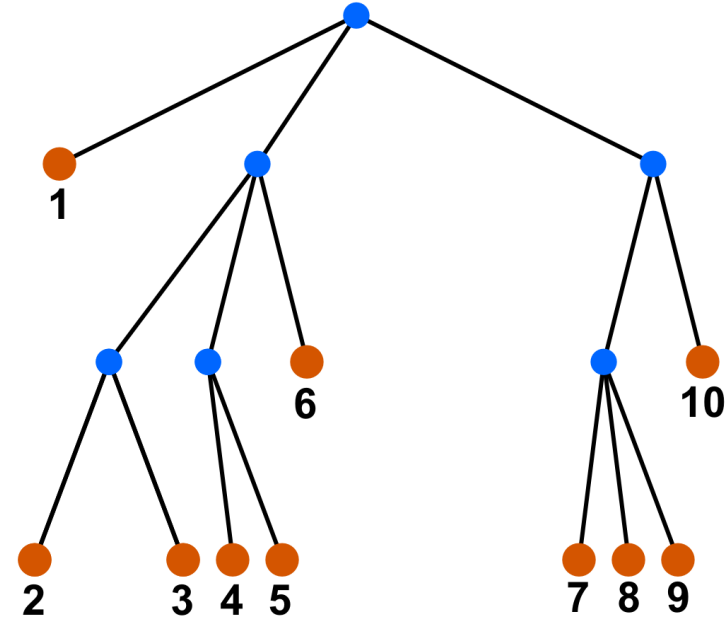
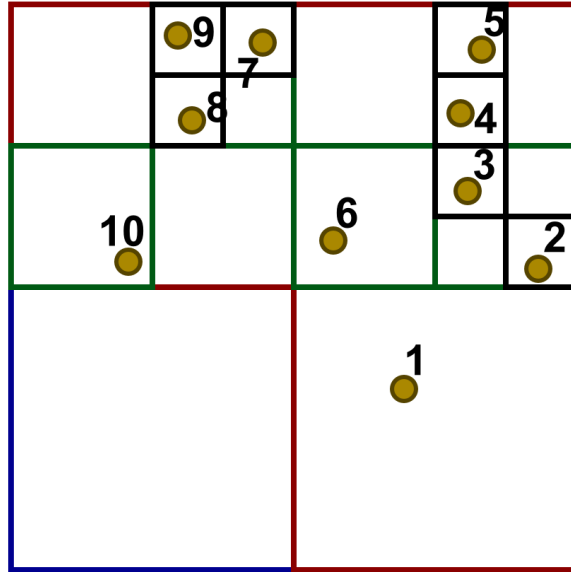
Le quatrième corps  
est seul dans le  
quadrant vert NE.

# Insertion du cinquième corps



... mais corps 4 et 5  
sont en conflit dans le  
quadrant NE. Ils vont  
donc au niveau 3.

# Et ainsi de suite ...



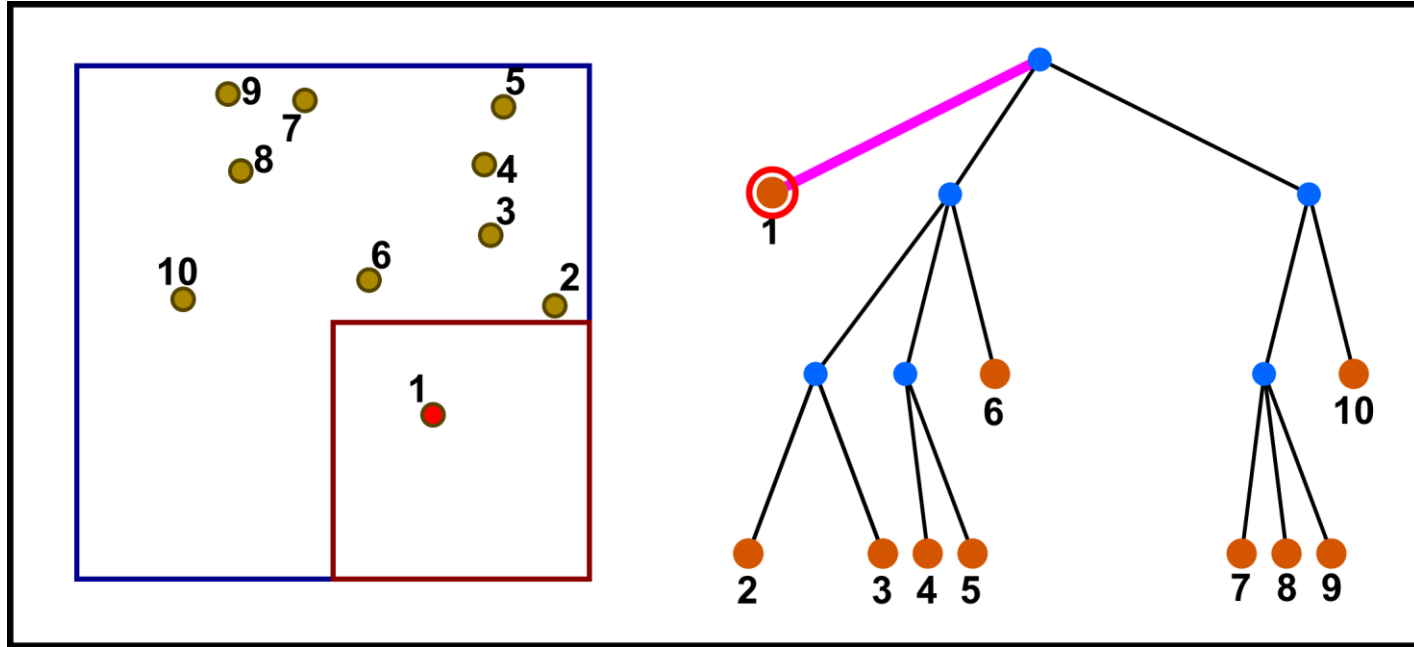
# Algorithme Barnes-Hut: utilisation du quadtree



# Calcul de la force sur corps 1

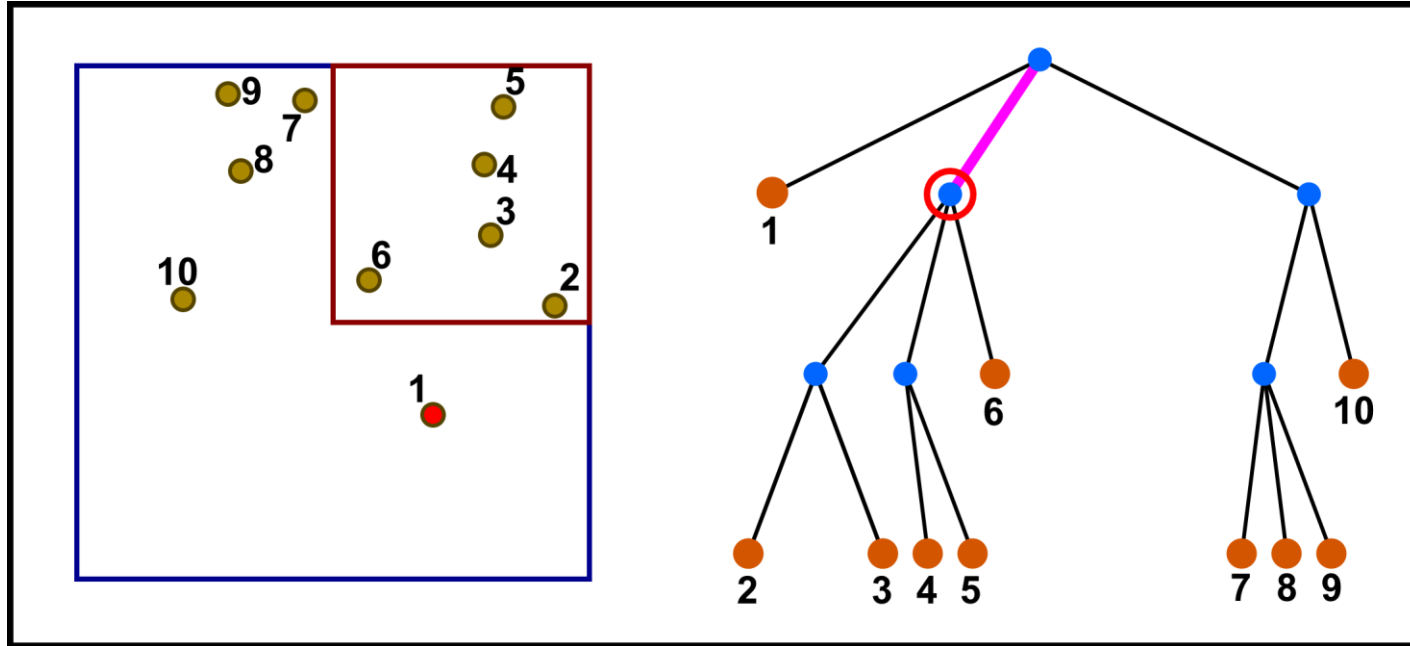
- Nous utilisons maintenant le quadtree pour calculer la force de gravité agissant sur le corps 1.
- Pour ça, nous traversons l'arbre de manière récursive. Au premier niveau, nous considérons les premiers quadrants rouges. S'ils sont suffisamment éloignés du corps 1, nous pouvons utiliser leur masse et centre de masse.
- Autrement, nous pénétrons récursivement dans ces domaines afin de considérer des quadrants plus petits.

# Parcours de l'arbre: quadrant SE



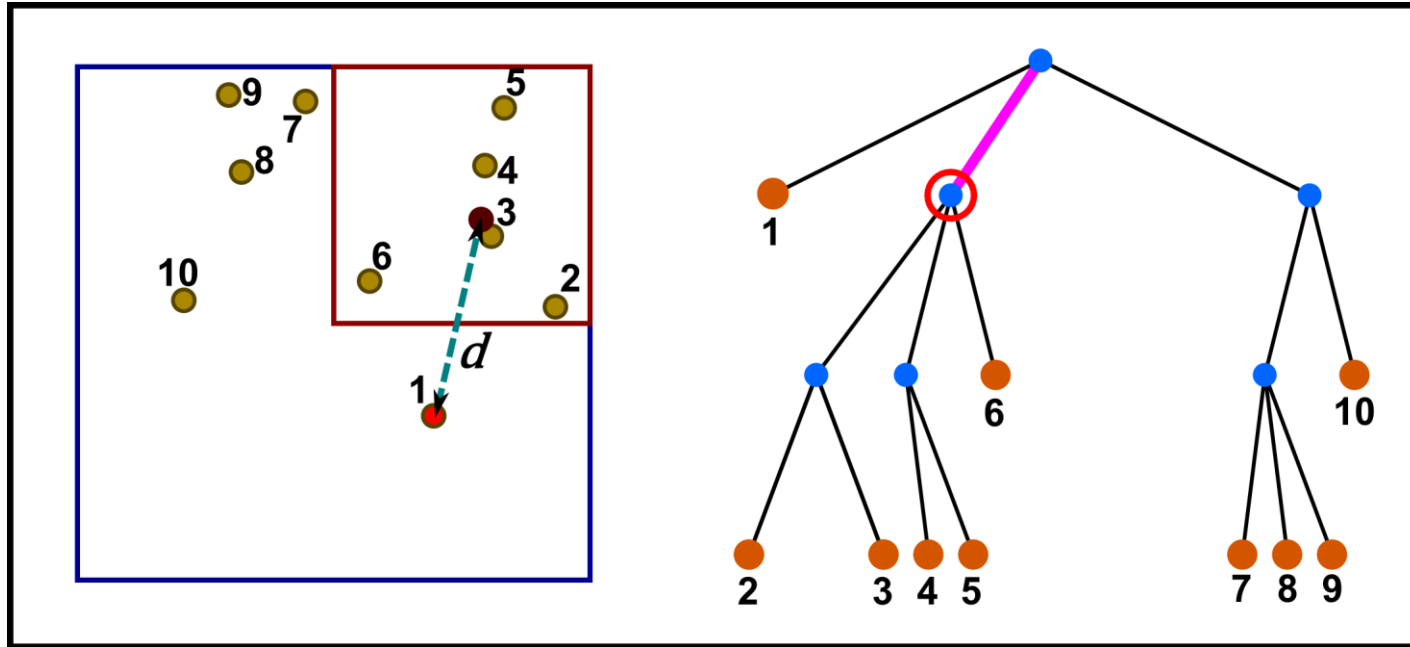
Le quadrant SE, au premier niveau, est peuplé par le corps 1 lui-même: rien à faire.

# Parcours de l'arbre: quadrant NE



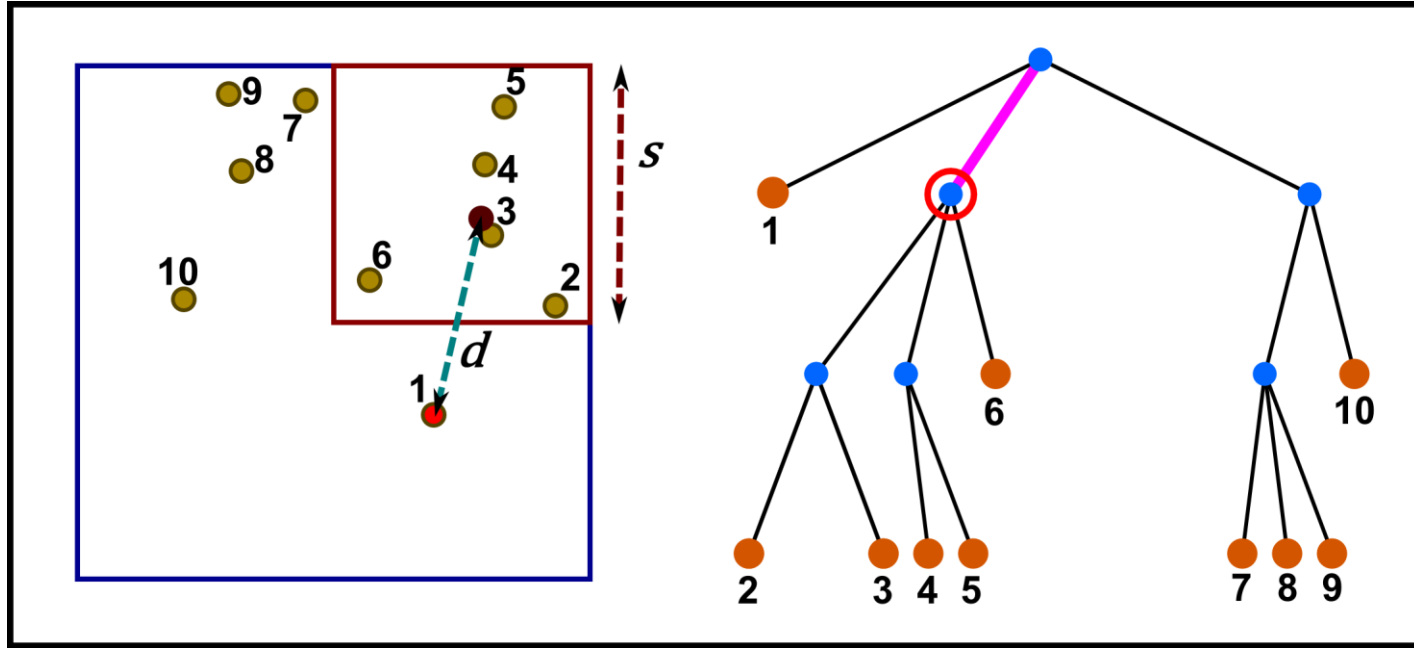
Il y a 5 corps dans le quadrant NE.

# Centre de masse du quadrant NE



Le centre de masse de ce quadrant se situe à une distance  $d$  du corps 1.

# Ce domaine est-il suffisamment éloigné?



Comme critère, nous comparons la distance  $d$  avec la taille  $s$  du quadrant.

# Critère theta

- Pour décider si un domaine est suffisamment éloigné, nous comparons sa taille  $s$  avec la distance  $d$  de son centre de masse.
- Nous décidons qu'un domaine est suffisamment éloigné si

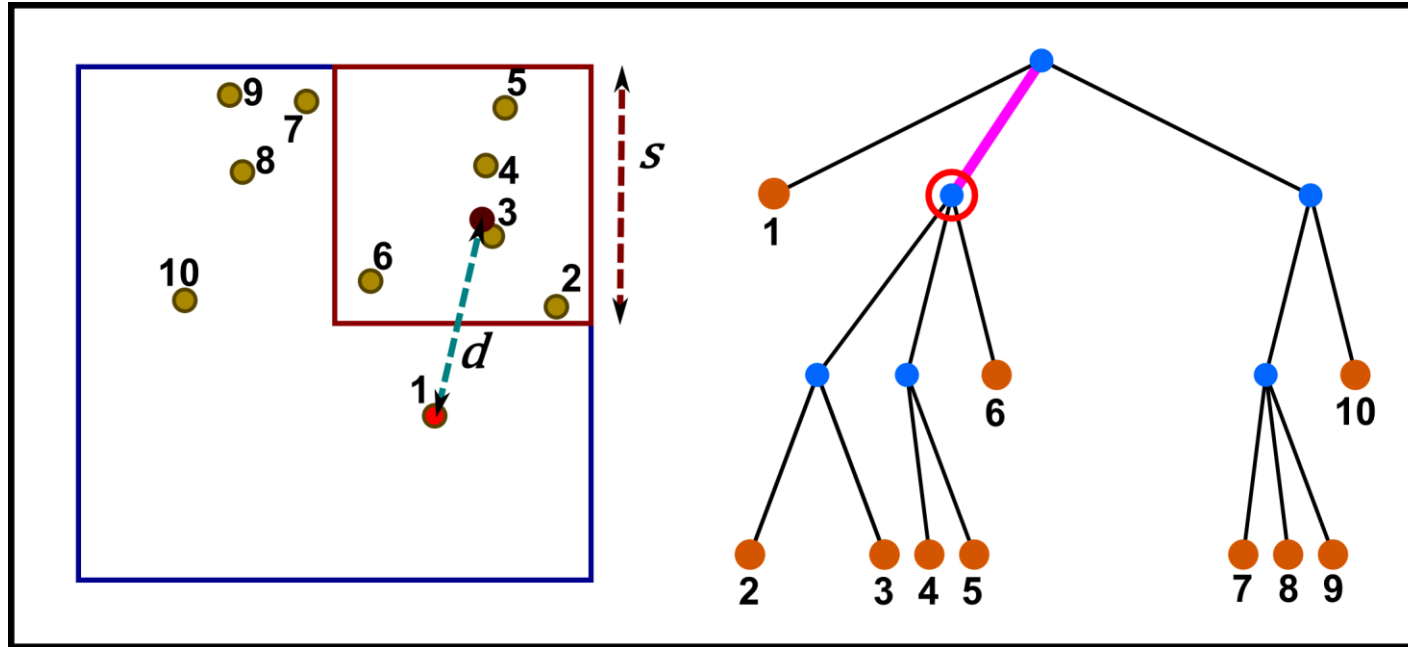
$$\frac{s}{d} < \theta$$

Où  $\theta$  est une valeur de seuil donnée. Dans cet exemple, nous travaillerons avec  $\theta = 0.5$ , une valeur commune pour l'algorithme de Barnes-Hut.

- Dans ce cas, le critère revient à:

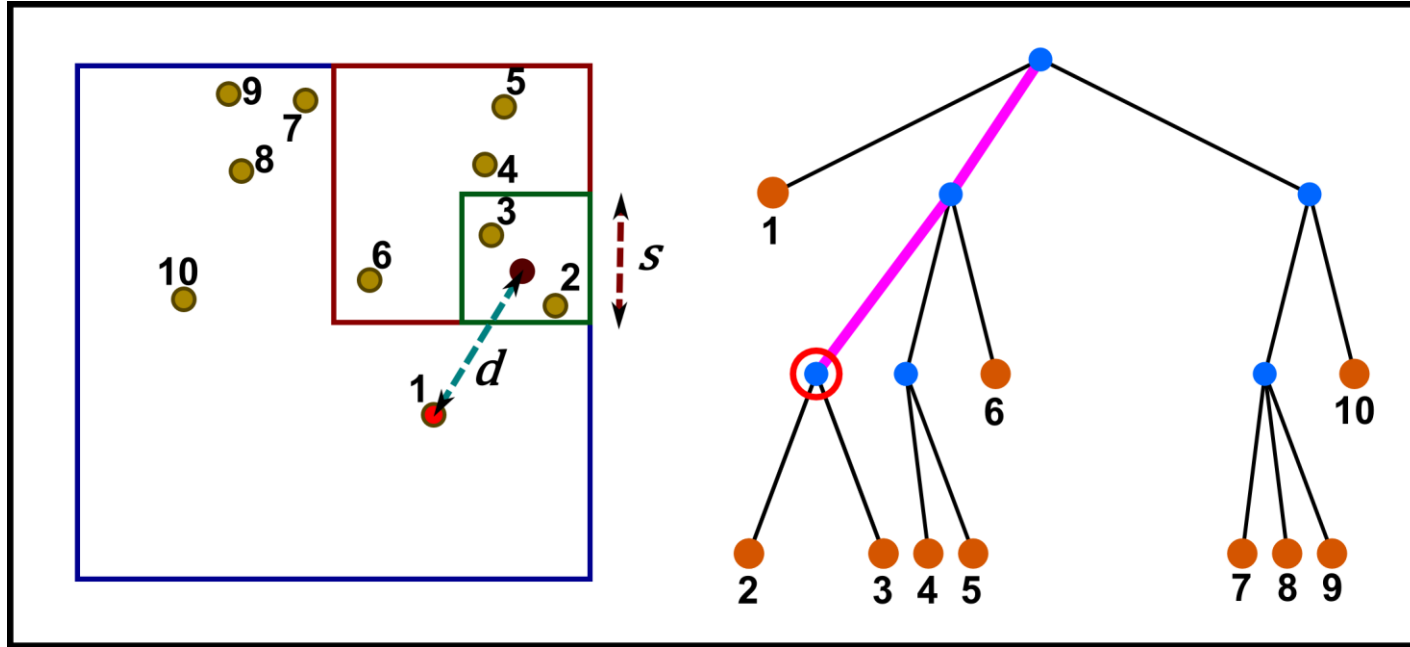
$$d > 2s$$

# Application du critère theta



Ici,  $d < 2s$ . Domaine rejeté!

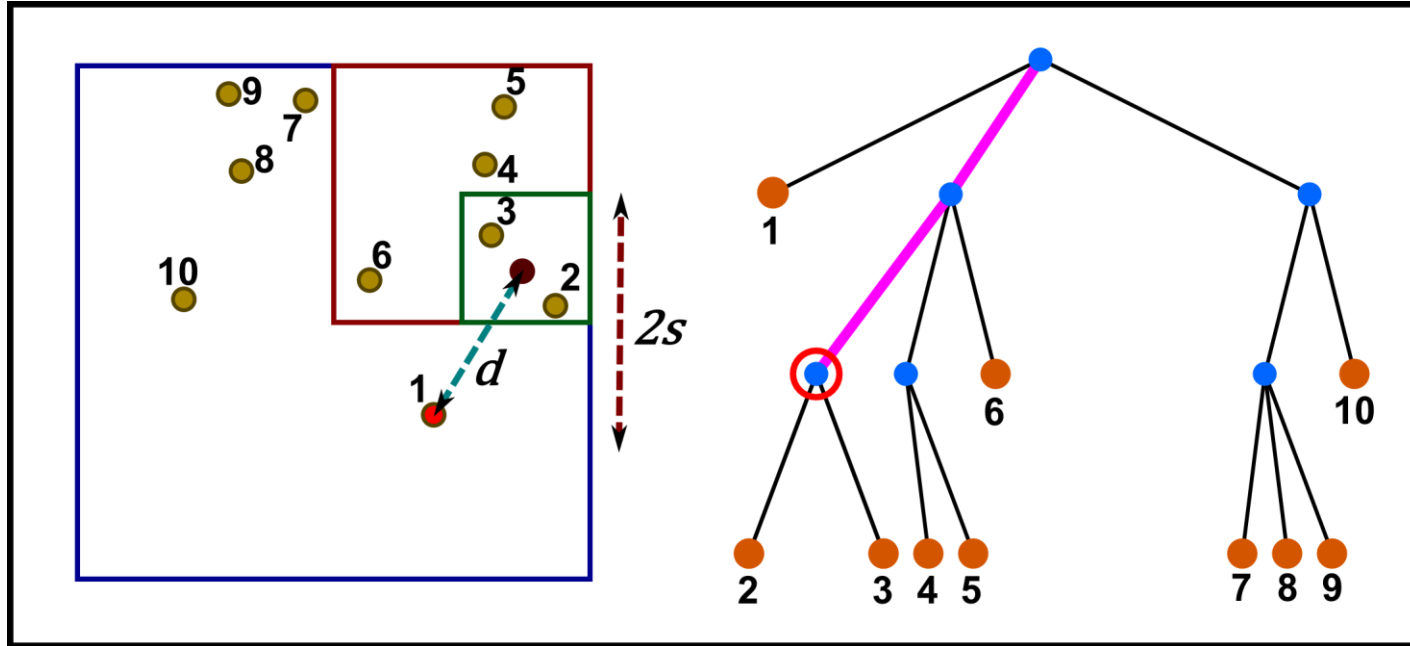
# Test du domaine SE au 2<sup>ème</sup> niveau



Cette fois-ci le domaine est plus petit...

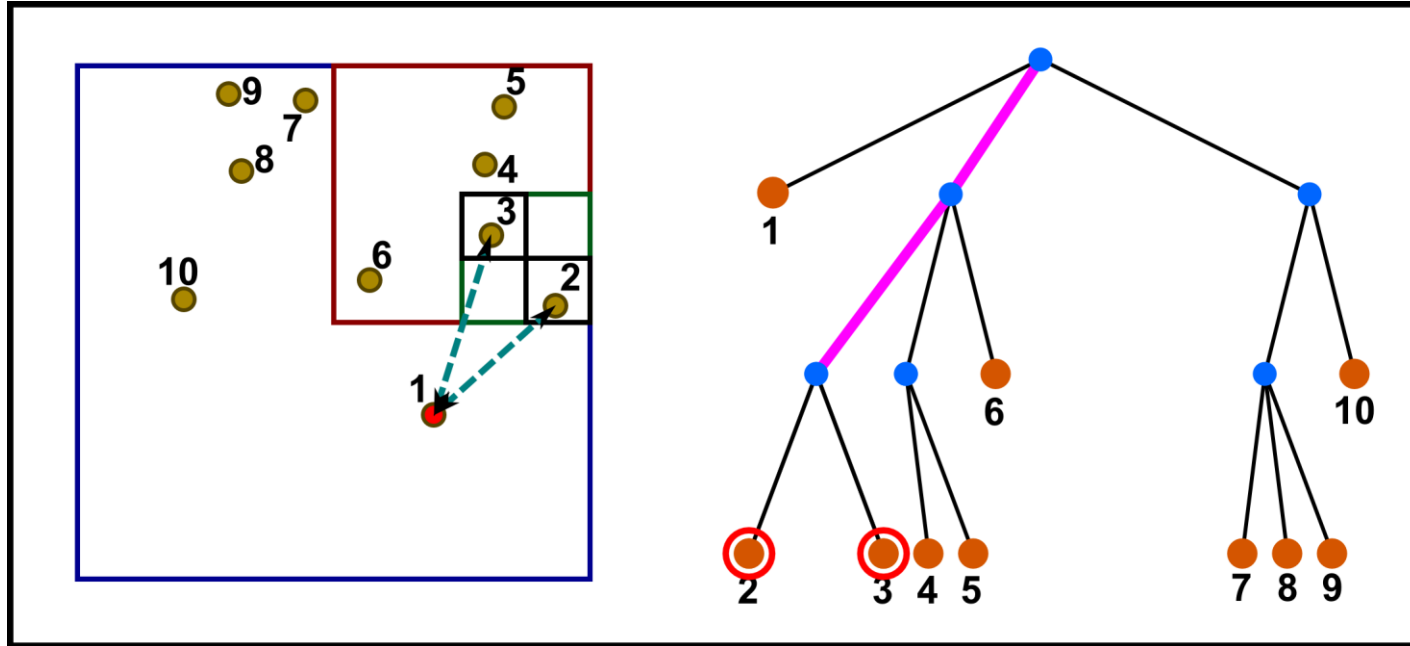


# Test du domaine SE au 2<sup>ème</sup> niveau



... mais néanmoins,  $d < 2s$ . Domaine rejeté encore!

# Il faut descendre au 3<sup>ème</sup> niveau



Ces noeuds-ci sont extérieurs, et représentent des corps seuls. Acceptés par défaut.

# Force agissant sur un corps: algorithme

Pour calculer la force sur  $b$ , on applique un procédé récursif, commençant au nœud de la racine.

1.	Si le noeud actuel est un noeud extérieur (et ne correspond pas au corps $b$ ), on calcule la force que ce noeud exerce sur $b$ , et on rajoute cette valeur à la force totale agissant sur le corps $b$ .
2.	Autrement, on calcule le rapport $s/d$ . Si $s/d < \theta$ , on traite ce noeud interne comme un seul objet, et on calcule la force qu'il exerce sur $b$ , pour rajouter cette valeur à la force totale de $b$ .
3.	Autrement, on procède récursivement avec toutes les branches du noeud actuel.

# Force agissant sur un corps: algorithme

Pour calculer la force sur  $b$ , on applique un procédé récursif, commençant au nœud de la racine.

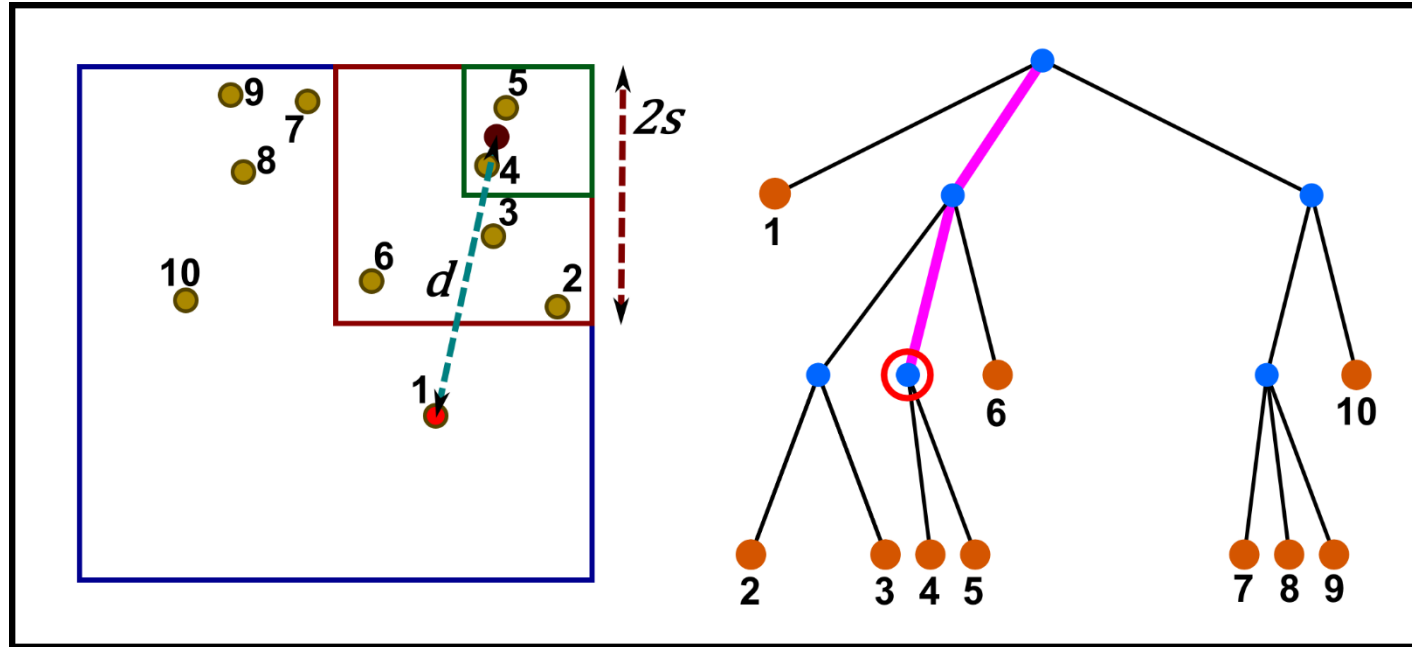
1.	Si le noeud actuel est un noeud extérieur (et ne correspond pas au corps $b$ ), on calcule la force que ce noeud exerce sur $b$ , et on rajoute cette valeur à la force totale agissant sur le corps $b$ .
2.	Autrement, on calcule le rapport $s/d$ . Si $s/d < \theta$ , on traite ce noeud interne comme un seul objet, et on calcule la force qu'il exerce sur $b$ , pour rajouter cette valeur à la force totale de $b$ .
3.	Autrement, on procède récursivement avec toutes les branches du noeud actuel.

# Force agissant sur un corps: algorithme

Pour calculer la force sur  $b$ , on applique un procédé récursif, commençant au nœud de la racine.

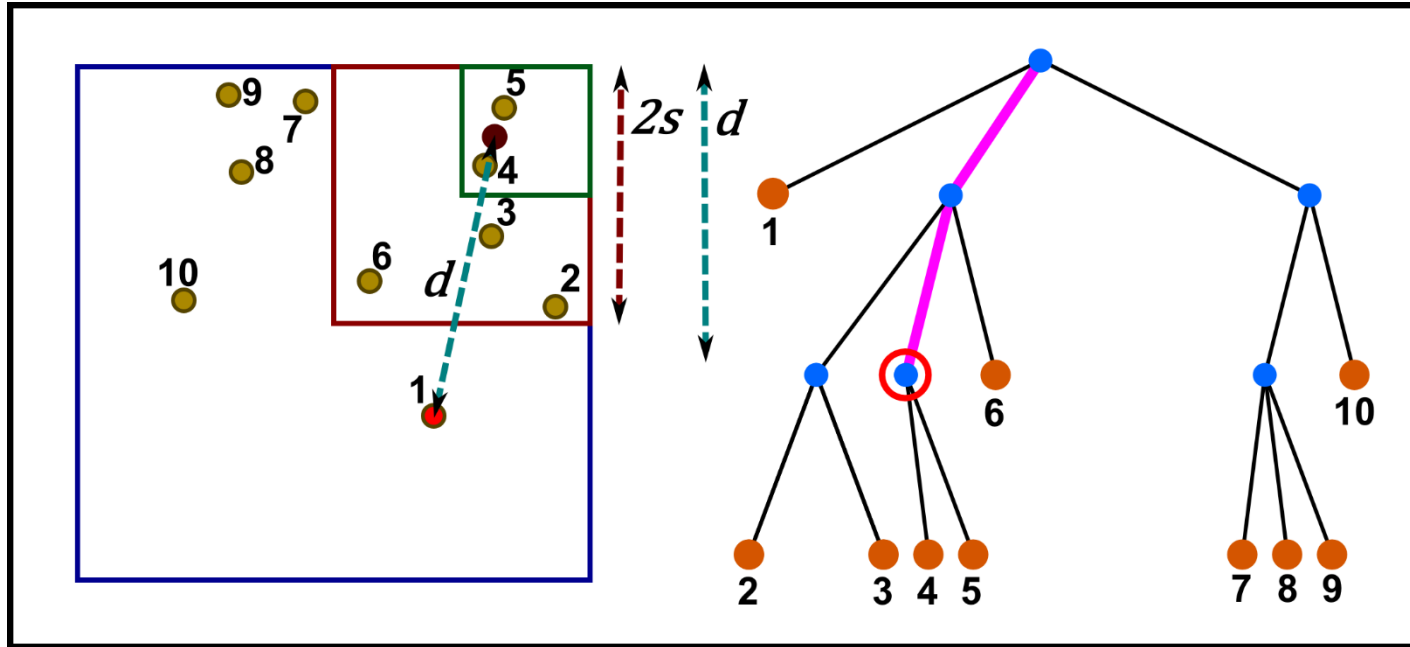
1.	Si le noeud actuel est un noeud extérieur (et ne correspond pas au corps $b$ ), on calcule la force que ce noeud exerce sur $b$ , et on rajoute cette valeur à la force totale agissant sur le corps $b$ .
2.	Autrement, on calcule le rapport $s/d$ . Si $s/d < \theta$ , on traite ce noeud interne comme un seul objet, et on calcule la force qu'il exerce sur $b$ , pour rajouter cette valeur à la force totale de $b$ .
3.	Autrement, on procède récursivement avec toutes les branches du noeud actuel.

# On poursuit avec le quadr. NE, niveau 2



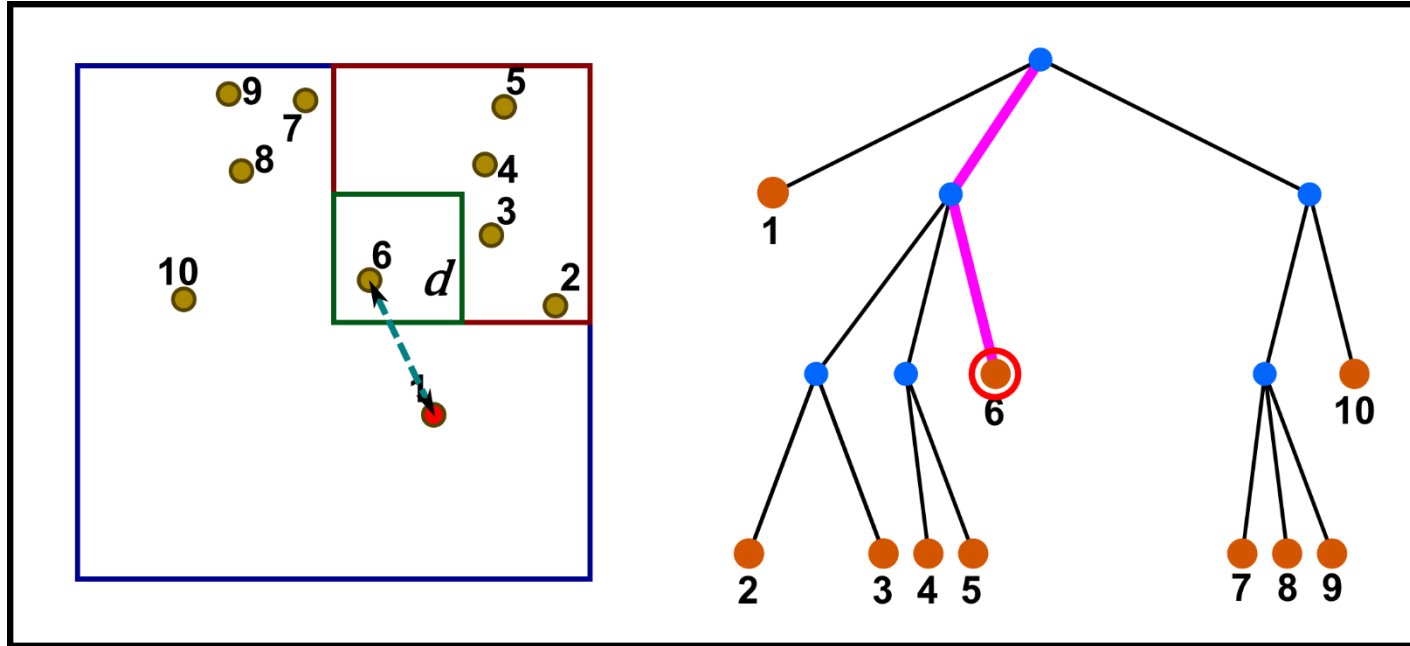
Il y a deux corps dans le quadrant NE vert. Pouvons-nous les remplacer par leur centre de masse?

# On poursuit avec le quadr. NE, niveau 2



Oui, on peut!  $d > 2s$ .

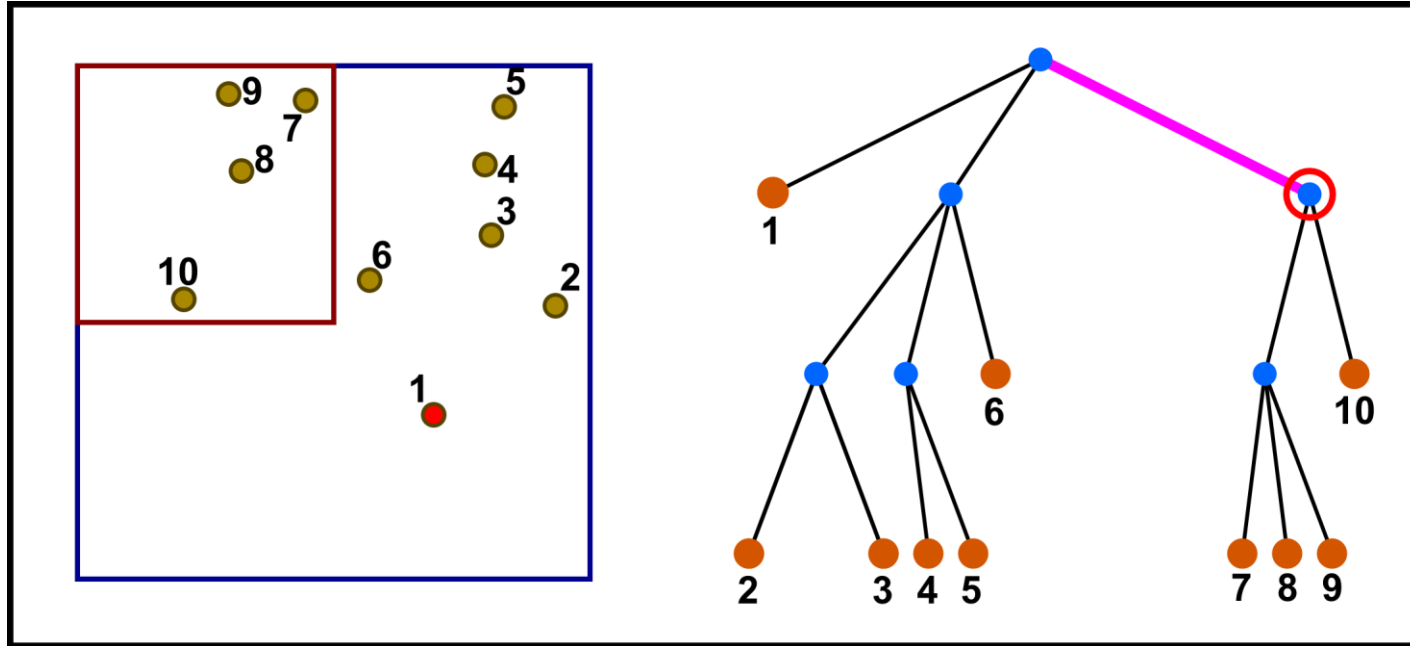
# Et le quadrant SE, niveau 2



Il s'agit d'un nœud extérieur, qu'on accepte par défaut.

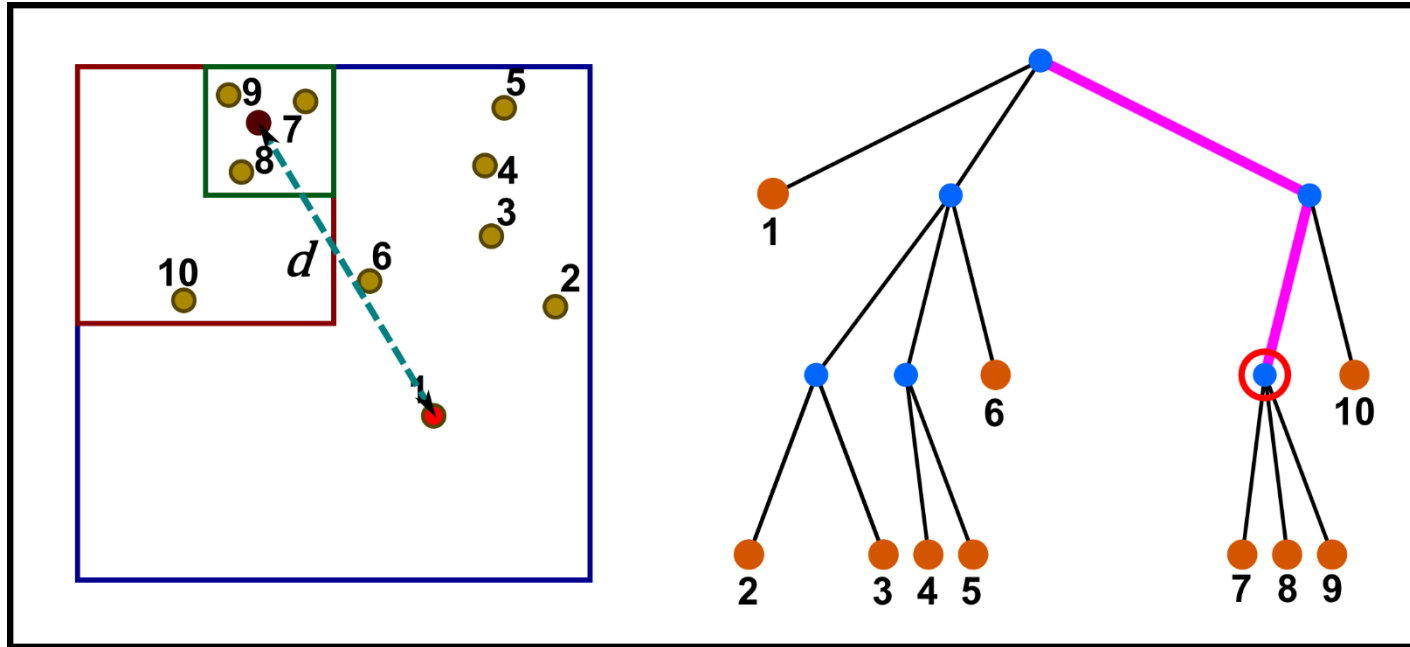


# Retour au niveau 1: quadrant NW



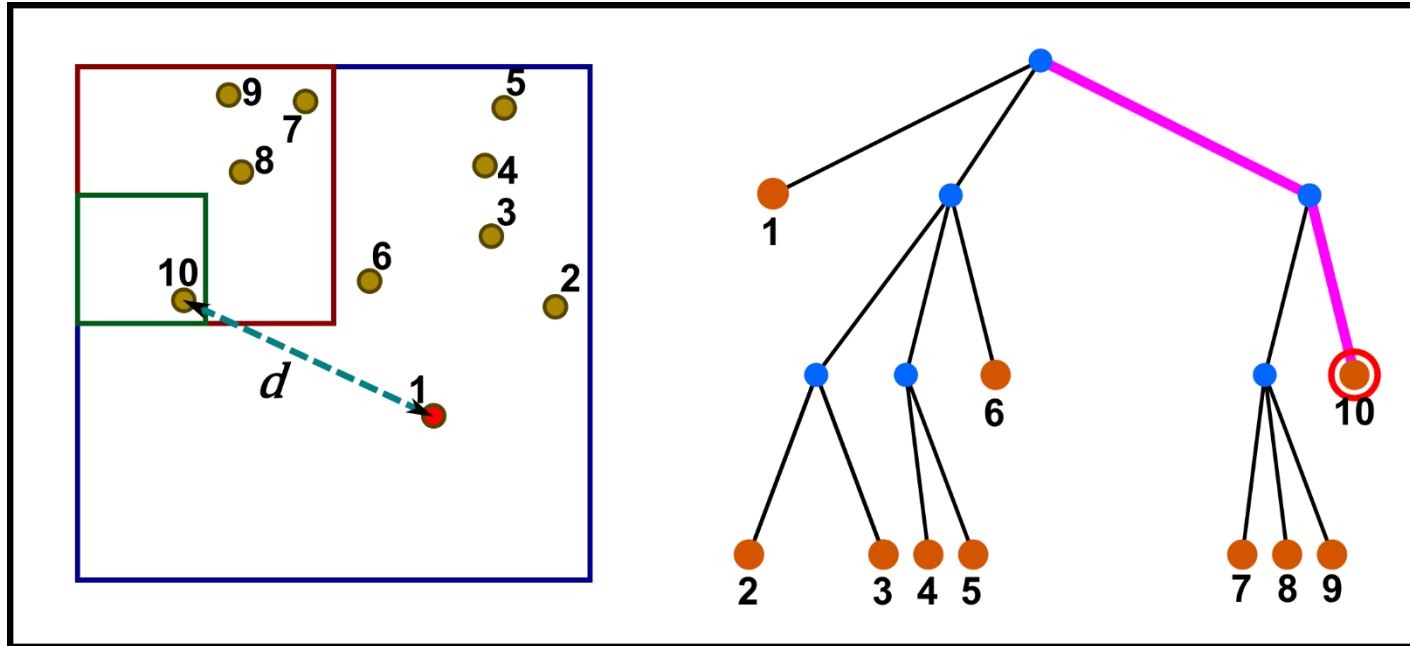
Ce quadrant ne vérifie clairement pas le critère theta...

# Niveau 3, quadrant NE



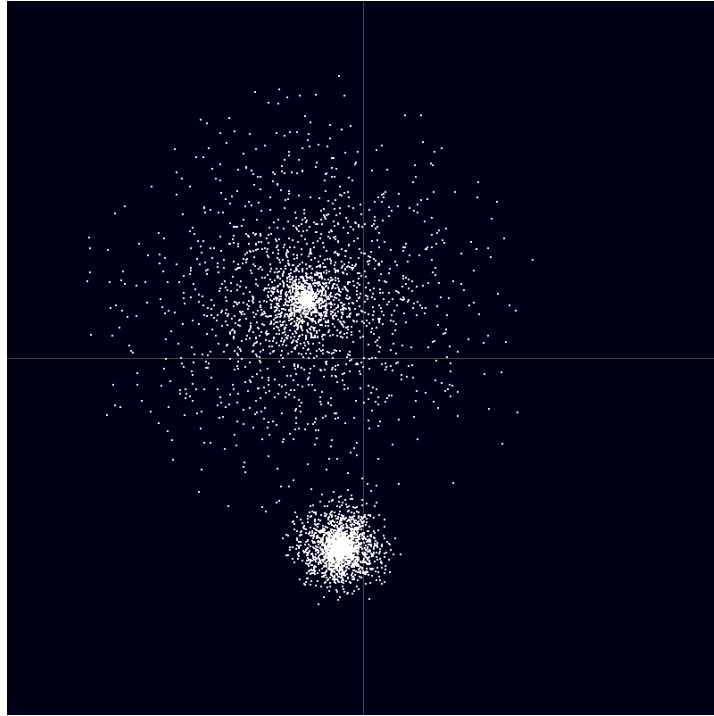
...mais son sous-quadrant NE fais l'affaire (croyez-moi...).

# Dernier quadrant ...



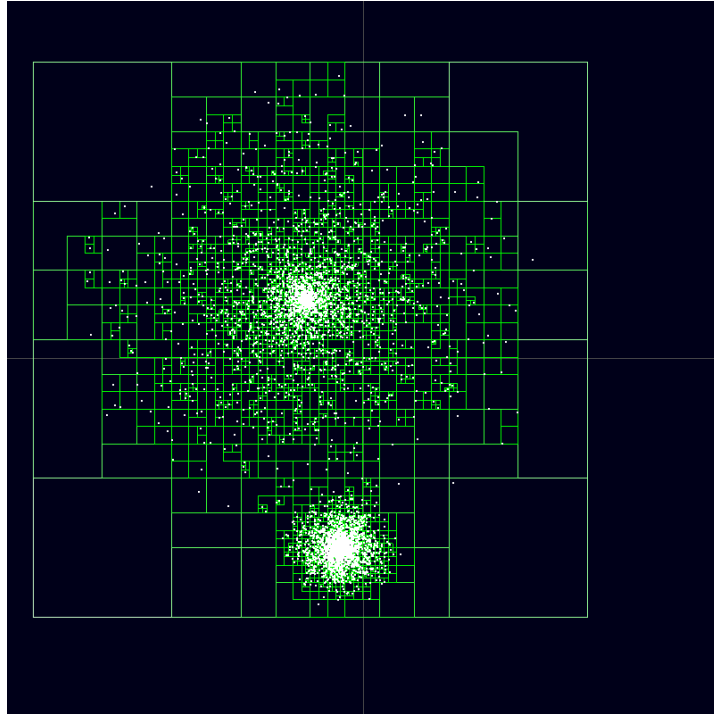
Et nous terminons l'algorithme par un nœud extérieur, qui est accepté par défaut.

# Avec 5000 corps: même procédé



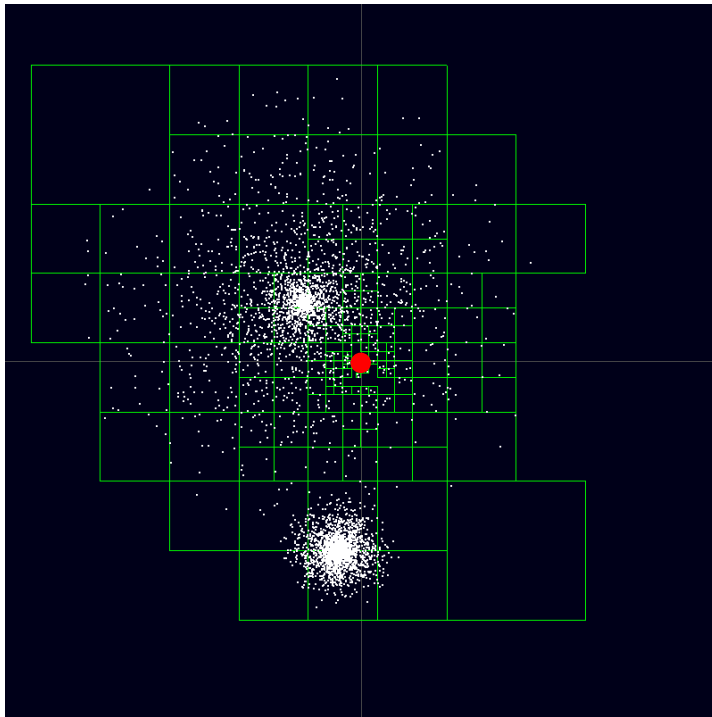
- Avec 10 corps, l'algorithme de Barnes-Hut ne nous fait pas vraiment gagner du temps.
- Mais avec davantage de corps, comme ici avec un nombre de 5000, l'algorithme accélère le calcul de manière substantielle.

# Avec 5000 corps: même procédé



- Structure du quadtree avec 5000 corps.

# Avec 5000 corps: même procédé



- Cellules choisies selon le critère theta pour calculer la force agissant sur le point rouge.

Attribution:

Toutes les images et animations du problème à 5000-corps proviennent de l'éditeur Wikipedia Eclipse.sx et sont distribuées sous un license [CC BY-SA 3.0](https://creativecommons.org/licenses/by-sa/3.0/) .