

Comment simuler numériquement la géomorphologie de type alpine

Manceau Thibaut, Gros Alexis, Porterier Tristan

January 25, 2016

Abstract

Avant le TPE nous étions fortement intéressé par la représentation 3D d'un terrain pour des applications telles que les jeux vidéo. Notre intention était donc d'expliquer une grande partie de notre travail déjà fait sur la représentation 3D d'un terrain pré-calculé (environ 6 mois de travail) dans le TPE.

La problématique est 'Comment simuler numériquement la géomorphologie alpine'
'Comment simuler' : Le but du TPE est de reproduire des phénomènes naturels.
'numériquement' : Nous n'avons pas les moyens de faire une maquette, donc cette simulation sera exclusivement réalisée avec de la programmation.
'la géomorphologie' : C'est l'étude de la surface de la terre et de son évolution.
'alpine' : La simulation prendra pour exemple le cas des alpes pour les phénomènes et les types de roches.

Comment on peut le remarquer la problématique ne se prête pas à notre intention initiale ou du moins que pour le résultat final si gain de temps. Partant de cette problématique, nous avons partagé le travail en deux branches principales : l'érosion et la dynamique orogénique (simulation des interactions entre roches).

Dans la partie dédiée à l'érosion on y retrouve des explications sur le phénomène appuyé par des caractéristiques sur les roches et l'application de cet algorithme dans notre simulation.

Quant à elle la branche sur les plaques tectoniques est composée d'une brève description des deux principaux phénomènes : la subduction et l'obduction, ainsi que les différents types de plaque tectonique : continentale et lithosphérique.

Enfin c'est de ces branches se rejoignant dans la programmation du système de simulation.

1 Les Cellules

Premièrement nous avons à définir le plus petit élément de cette simulation, dans ce rôle on y trouvera ce qu'on appellera les cellules. Une cellule est un morceau de terrain sphérique ou cubique d'environ 10 mètres ce qui nous fait un volume de 1000 mètres

cube. Ces cellules ont pour caractéristique immuable leur type de roche et comme caractéristiques mutable la plaque tectonique auquel elles appartiennent, leur position et leur force.

Toutes cellules sont disposées sur une grille pour former un grand bloque indépendant des plaques tectoniques. La disposition des cellules à était sujet à controverses, celles ci doivent former une grille carrée ou un grille du type 'nid d'abeilles' ? Finalement nous avons choisi la forme en 'nid d'abeilles' car elle permet une équidistance entre les cellules.

1.1 Disposition des cellules

Les cellules étant le plus petit élément composant le terrain, leur organisation compte énormément au début de la simulation. Plusieurs dispositions ont été testé. Pour simplifier la programmation les cellules sont disposées que sur un plan pour permettre l'utilisation d'un modèle 2D bien plus simple à écrire et à déboguer, même si dans le code de la simulation nous utilisons des vecteurs en 3D, nous gardons des vélocités strictement en 2D.

Après avoir choisi un modèle planaire, le choix de l'organisation des cellules sur ce plan doit ce faire. La première disposition choisi fut une grille. Lors de la création la simulation doit créer n_x cellules en abscisse et n_y en ordonnée, avec une grille nous faisons une simple itération x à x_n et a l'intérieur une autre itération de y à y_n .

2 Interactions entre cellules

2.1 Recherche des cellules adjacentes

Pour commencer les cellules doivent connaître toutes ses plus proche cellules adjacentes pour pouvoir interagir avec elles. Pour cela nous avons utilisés un arbre KD pour optimiser la recherche des cellules, en effet l'arbre KD permet une recherche des points inclut dans une sphère bien plus rapide qu'une simple recherche linéaire. Au début de chaque nouvelle étape complète de simulation, l'arbre KD est détruit puis remplacé par un nouveau vide, ensuite nous ajoutons toutes les positions des cellules dans l'ordre (en effet le numéro d'ajout et le seul moyen de retrouver la cellules) puis nous exécutons le tri des positions dans l'arbre KD. Après avoir instancié l'arbre chaque cellules peut rechercher ses cellules adjacentes dans un rayon égale à la norme d'un vecteur (1 ; 1) (la distance originelle entre les cellules) puis stocke ces cellules dans une liste utilisé jusqu'à la recréation de l'arbre KD, donc du déplacement des cellules.

2.2 Propagation par front de cellules

Pour propager la vélocité plusieurs méthodes ont été envisagé, la première entant le plus simple ce composé simplement d'un itération linéaire pour toutes les cellules existantes, ainsi chacune d'elle devra propager sa propre vélocité à ses cellules adjacentes. Mais cette méthode est totalement erronée car si nous faisons l'itération de toutes les autres cellules avant la cellule en collision, celle ci n'aura d'effet que sur ses seuls cellules adjacentes et non sur toute la plaque. Pour remédier à ce problème nous avons du aborder une approche par fronts partant de la cellules en collision et s'étendant sur

toute la plaque formant ainsi une onde. Deux méthodes s'offraient à nous, la première est de trouver toutes les cellules dans un intervalle de rayons par rapport à la cellule en collision, la deuxième quant à elle est de créer un front d'origine avec seulement la cellules en collision et que chaque front créer le suivant en ajoutant les cellules n'ayant jamais participé au front. Chacune de ces méthodes ont leurs avantages et inconvénients. La première ne fonctionne que avec une plaque rectangulaire, en effet si la plaque est coude les cellules de l'autre côté du coude seront déjà avant celles dans le coude si la cellule en collision se trouve sur une des extrémités, ce problème n'est plus avec la deuxième méthode mais celle ci a pour désavantage de ne pas former un front circulaire mais basé sur la disposition des cellules, donnant ainsi un front hexagonal pour une disposition en nid-d'abeille. Pensant le pour et le contre et en tenant compte du temps de calculs (notre implémentation d'arbre KD ne permettait pas une recherche par intervalle de rayons ce qui nous oblige à faire deux recherches par rayon et de supprimer le doublons) nous avons choisi la méthode récursive. Pour ce faire les cellules possèdent maintenant deux nouvelles propriétés mutable : une pour savoir si la cellule a déjà fait partie du front (cela signifie aussi que la cellule a déjà propagée sa vitesse) et l'autre pour savoir si la cellule fait actuellement partie du front, ces deux variables sont remis à faux au début de chaque étape complète de simulation. Pour chaque cellules en collision nous créons un nouveau front ne contenant que la cellules en collision, toutes les cellules contenues dans le front sont considérées comme déjà calculées et ne vont plus faire partie du front, donc elles voient leur première propriété mis à vrai et leur deuxième à faux. Ensuite chaque cellules dans le front itère sur toutes ses cellules adjacentes et les ajoute si leurs deux propriétés sont fausses (elle n'ont pas et ne font pas partie du front), chaque cellules remplissant ces conditions et ajouté dans le front et voit sa deuxième propriété mis à vrai car elle fait désormais partie du front. Ainsi nous créons nouveau front grâce à l'ancien sans cellules en doublons. Dès que le front ne contient plus aucune cellules cela signifie que nous avons fini avec la cellule en collision actuelle et pouvons passer à la prochaine cellule en collision ou recommencer un étape complète de simulation.

2.3 Calques de vitesse

Pour éviter des interférences entre les différentes propagations de vitesse des cellules en collision, nous utilisons des calques uniques contenant une vitesse par cellules, chacun de ces calques sont liés à une cellule en collision. Au début de l'étape complète de simulation chaque cellule réserve dans ses variables une liste de vecteur pour chaque cellules en collision puis pour chaque propagation de vitesse des cellules en collision, les cellules affectées ne travaillent que sur le vecteur correspondant au numéro de la cellule en collision. Puis à la fin de l'étape complète de simulation tous les vecteurs sont ajoutés entre eux pour donner le déplacement à appliquer à la cellule.

2.4 Interaction entre cellules dans un modèle sans friction, compression et réaction

Comme nous l'avons expliqué chaque cellule dans le front doit appliquer une vitesse aux cellules qui feront partie du front une itération après. Mais quelle doit être la direction de cette vitesse et sa norme ? Prenant le cas de deux cellules C_1 et C_2 . C_1 fait partie du front et C_2 une de ses adjacentes non calculées, C_1 reçoit une vitesse \vec{w} , C_2 devra recevoir une vitesse \vec{v} et \vec{w} représente le vecteur de C_1 vers C_2 . L'estimation de la direction de la vitesse dans un modèle sans friction est des plus simple, elle équivaut

au vecteur allant de la cellules C_1 vers C_2 mais l'estimation de la norme de ce vecteur est quant à elle bien plus compliquée. La fonction calculant cette norme doit apporter les résultats suivants : lorsque le vecteur \vec{w} est colinéaire à \vec{u} (les deux cellules sont alignées sur la vitesse) la cellule C_2 doit recevoir toute la vitesse $\vec{v} = \vec{v}$ et, au contraire, si \vec{w} et \vec{u} forment un angle $\frac{\pi}{2}$ alors la cellule C_2 ne doit recevoir aucune vitesse, $\vec{v} = \vec{0}$, car elle est disposée latéralement à la cellule C_1 et n'étant pas dans un modèle avec friction elle ne bouge pas. La loi répondant à ces deux cas est le "Centre instantané de rotation d'un solide". Cette loi permet de mettre en relation la vitesse à deux points de contacts sur deux plans d'un solide grâce à une rotation. Prenons l'exemple d'une échelle (le modèle est sans friction et sans gravité) posée avec un angle faible α sur un mur perpendiculaire au sol, elle forme deux points de contact P_1 en haut de l'échelle contre le mur et P_2 en bas de l'échelle contre le sol. Si nous appuyons sur le haut de l'échelle dans la direction du mur celle-ci va avoir tendance à glisser en direction du sol. En instantané nous pouvons faire l'approximation de ce mouvement par une rotation dont le centre serait l'intersection des deux droites perpendiculaires au sol et au mur passant respectivement par les points P_1 et P_2 . Nous nommerons \vec{u} la vitesse de l'échelle à P_1 , \vec{v} la vitesse à P_2 lors du glissement, C le centre de rotation, d la distance entre C et P_1 , f la distance entre C et P_2 et enfin α l'angle de rotation. Cette rotation nous permet de lier la norme de la vitesse appliquée sur le haut de l'échelle \vec{u} et celle du glissement \vec{v} grâce à α , d et f .

$$\|\vec{u}\| = d \times \alpha$$

$$\|\vec{v}\| = f \times \alpha$$

Dans le cas de nos deux cellules C_1 et C_2 en compression, nous ne connaissons pas α ni le centre C donc d et f de même. Nous nommerons β l'angle obtenu entre \vec{u} et \vec{w} représentant l'angle entre la vitesse et le vecteur de direction entre les deux cellules. Cette angle nous permet de calculer les deux distances d et f :

$$d = \frac{\|\vec{w}\|}{\sin \beta}$$

$$f = \frac{\|\vec{w}\|}{\tan \beta}$$

Puis α est déduit de la norme de \vec{u} et d :

$$\alpha = \frac{\|\vec{u}\|}{d}$$

La formule complète pour calculer la norme de \vec{v} se présente :

$$\|\vec{v}\| = \frac{\|\vec{w}\| \times \|\vec{u}\|}{\tan \beta \times d}$$

L'implémentation de cette loi doit traiter quelques cas particuliers pour éviter toutes divisions par zéro, si $\cos \beta = 0$ alors $\vec{v} = \vec{0}$ et si $\sin \beta = 0$ alors $\vec{v} = \vec{u}$.

Cette formule convient parfaitement pour une interaction avec qu'une seule cellule, mais dans la plupart des cas il y en a plus, en effet au début de la simulation une cellule du front peut avoir au maximum quatre cellules adjacentes non calculées. Nous ne pouvons pas utiliser la même formule car la somme de la norme des vitesses appliquées aux cellules pourrait être supérieure à $\|\vec{u}\|$, cela signifierait une création de vitesse. Pour palier à ce problème toutes les vitesses sont divisées par le nombre de cellules adjacentes non calculées.

$$\sum_{i=1}^n \|\vec{v}_{max}\| = 1$$

$$\vec{v}_{max} = \frac{\vec{v}}{n}$$

2.5 Compression et traction entre cellules

Maintenant que nous avons la formule pour la répartition de la vitesse entre les cellules nous devons gérer les cas de compressions et traction. Les cellules adjacentes doivent toujours viser une distance parfaite équivalent à la norme d'un vecteur $(1, 1)$.

2.6 Friction entre cellules

3 Limites matérielles