

ÉCOLE NORMALE SUPÉRIEURE
DÉPARTEMENT D'INFORMATIQUE
RAPPORT DE STAGE DE L3

Génération procédurale de paysages à partir de cartes vectorielles

Intégration d'une mini-simulation géologique

Élie MICHEL
29 AOÛT 2014

Stage réalisé du 9 juin au 10 août 2014 dans l'équipe Imagine de l'Inria Grenoble sous la direction de Marie-Paule Cani.

Introduction

Génération procédurale

Alors que les capacités de stockage et d'affichage de contenus ne cessent d'aller croissant, les capacités humaines à créer ce contenu ne peuvent suivre le même rythme. Si dans de nombreux domaines la création de contenu a elle aussi pu être automatisée, le domaine artistique reste par essence très lié à l'humain et dégager des automatismes permettant de l'assister procéduralement, ne serait-ce que partiellement, révèle de nombreux défis.

Pourtant, le besoin de contenus toujours plus détaillés pour les jeux vidéos (simulateurs divers, jeux de rôle, etc) ou pour le monde de l'image (cinéma, publicité, etc) par exemple nécessite de dépasser les limites purement humaines. C'est à ce niveau qu'interviennent les diverses méthodes de génération procédurale de contenu.

Leur but est multiple. Elles doivent être capable de créer du contenu, mais aussi de dialoguer avec les artistes afin que ces derniers puissent expliquer ce qu'ils cherchent. Elles doivent donc être suffisamment générales pour convenir dans de nombreux cas tout en étant capables de générer des détails. Elles doivent être un outil par lequel l'artiste s'exprime et non contre lequel il entre en compétition. De ce fait, la question de l'interactivité et du contrôle que l'on garde sur le résultat d'un algorithme est très souvent questionnée dans la littérature.

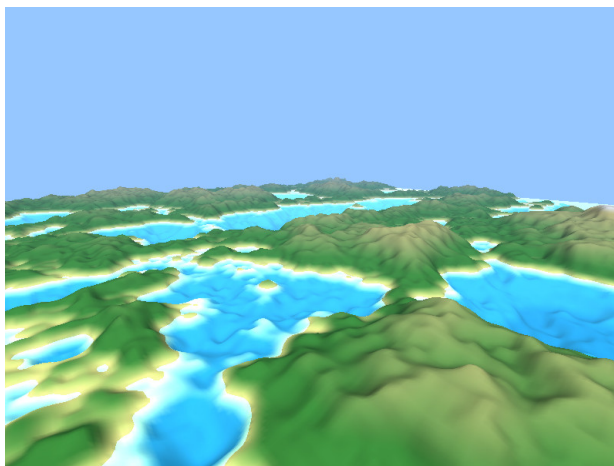
Paysages

La génération de paysages fait partie des premiers éléments abordés par la création procédurale. Dessiner un paysage nécessite en effet un long travail étant donné l'étendue qu'il peut avoir et l'importance de toutes les échelles, de l'horizon aux touffes d'herbe du premier plan.

En plus d'être vastes, les paysages s'organisent autour d'éléments clefs : vallées, sommets, falaises, etc. Le couplage d'une grande complexité et de règles de dessin en apparence simples donne ainsi à l'approche procédurale tout son sens.

Premiers pas

Les premières approches reposent sur la nature fractale des paysages, leur aspect multi-échelle, que Mandelbrot note dès 1968 [Mandelbrot68]. Le *mouvement Brownien fractionnaire* est ainsi utilisé pour générer des cartes de hauteur, basé à partir des années 80 sur le *bruit de Perlin* [Perlin85] ou le *midpoint displacement* [Fournier82].



Terrain généré à partir d'un bruit de Perlin

Ces premières méthodes souffrent de divers problèmes. Le plus flagrant est l'uniformité du terrain généré. Les montagnes sont partout et ont toutes le même profil. Il n'y a aucune organisation globale, même si l'ajout d'octave à grande échelle fait apparaître des zones d'océan et de continent par exemple.

D'autre part, la génération est indépendante de tout contrôle. On ne peut pas imposer à l'algorithme la présence de montagnes à certains endroits ou leur absence ailleurs.

Remarque

Je parle ici essentiellement de la génération du relief, sous forme de carte de hauteur (ou *heightmap*) dans l'immense majorité des cas, et pas de son rendu. La génération de la texture du terrain devrait aller de paire avec celle de la carte de hauteur, surtout si l'on passe par une phase de simulation et que l'on sait donc comment les divers éléments du relief ont été créés (éboulis, falaises, zone humides, etc). La carte de hauteur seule est cependant un bon début puisque c'est elle qui façonne le terrain et permet par la suite d'interagir avec lui (détection de collision, etc).

Filtres

Un des avantages des cartes de hauteurs, c'est la facilité avec laquelle on peut les manipuler. Ce ne sont en effet que des images et donc tous les filtres utilisés en traitement d'image peuvent être directement utilisés. Pour rompre la monotonie des cartes browniennes, on a donc combiné diverses textures de base : cartes de Voronoi pour délimiter des zones, divers bruits, etc. On peut ensuite les combiner par déformation, appliquer des flous, des fusions etc.

Les résultats restent cependant pauvres car ces filtres sont généralement locaux. On ne voit donc à grande échelle aucun effet d'ensemble comme c'est le cas dans la réalité avec les chaînes de montagnes, les grandes vallées, etc.

Approches géologique

De nouveaux filtres ont donc été développés, basés sur une approche géologique du problème. Le principal effet pris en compte est alors l'érosion [Musgrave89], qu'elle soit hydraulique (vallées creusées par l'eau) ou thermique (effondrement des pentes trop abruptes).

Ces méthodes d'érosion sont appliquées à un terrain déjà existant, généré par exemple par une approche fractale [Pytel13] ou bien en diffusant une contrainte donnée par l'utilisateur [Hnaidi10]. Les méthodes vont d'une simulation simplifiée par colonnes d'eau [Beneš02] à des simulations plus précises, mais aussi plus coûteuses [Kristof09].

Le principal inconvénient de ces méthodes est le temps qu'elles prennent. Il faut en effet diffuser de l'information partout dans la carte. On perd alors notamment la possibilité de modéliser le terrain en temps réel : on doit proposer une donnée d'entrée puis attendre un petit moment avant de voir le résultat. Puis réessayer. C'est frustrant pour l'utilisateur et fait perdre du temps.

Les articles les plus récents implémentent donc ces méthodes sur le GPU et retrouvent ainsi cette interactivité [Mei07], [Št'ava08]. Mais d'autres méthodes, moins gourmandes en ressources et plus dans l'esprit de la génération procédurale que de la simulation, ont été recherchées. [Zhou07] recherche les motifs donnés en entrée par l'utilisateur dans une base de données de reliefs existants. [Kelley88] propose une approche qui est encore fractale tout en permettant la création de rivières et vallées.

Cette idée de générer les montagnes à partir d'un réseau de rivières est reprise à plusieurs occasions [Prusinkiewicz93], [Belhadj05], [Génévaux13] et constitue la principale alternative aux méthodes basées sur la simulation.

Certains ont également cherché à s'affranchir des limitations du modèle de carte de hauteur. [Beneš01] établit un modèle de colonnes permettant de gérer plusieurs strates de roche. [Peytavie09] va plus loin encore en permettant la formation de tunnels et grottes.

Cependant, les cartes de hauteur conservent un avantage de taille : elles sont utilisées depuis longtemps donc des outils existent pour les manipuler (éditeurs d'image) et en faire le rendu (d'autant plus que le rendu représente un travail lourd). J'ai donc préféré en rester aux cartes de hauteur.

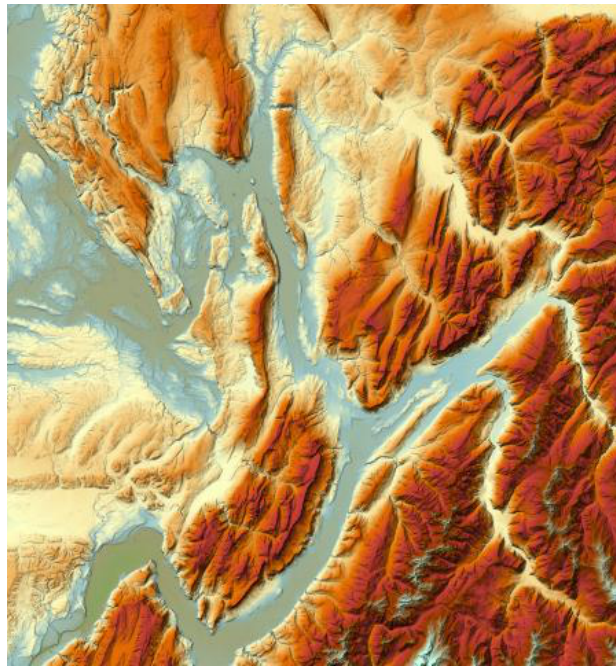
Pour plus d'information sur les différentes méthodes de génération de terrain utilisées, voir [Smelik09] et [Natali13]. Le second inclut des considérations géologiques autres que l'érosion : les événements géologiques (plis, failles, etc). Cependant, il y a encore relativement peu de méthodes, et surtout peu de méthodes simples. En particulier, les représentations employées sont plus complexes que les cartes de hauteur.

Problème initial

Diversité

Bien que prenant en compte l'érosion avec un niveau de détail grandissant, les terrains procéduraux pris dans

leur globalité sont souvent relativement uniformes. On voit apparaître des vallées et les bassins versants sont cohérents, mais les montagnes, bien que localement très détaillées, n'ont pas de forme d'ensemble. Les chaînes montagneuses réelles ont en général un caractère assez marqué. Il est souvent possible de reconnaître très vite dans quel massif une photographie a été prise (pourvu de connaître cet endroit). Et certains éléments, comme les plis très marqués sur la carte de hauteur suivante, donnent une unité à un massif.



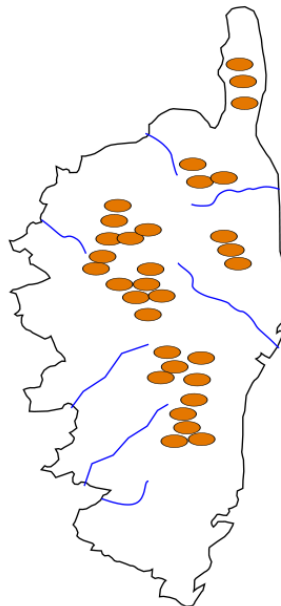
Plis dans les massifs des Bauges et de la Chartreuse (Alpes)

Ces éléments à grande échelle proviennent de phénomènes géologiques non pris en compte par la seule érosion et découlent de la tectonique des plaques. Un des buts qui m'étaient fixés était de générer, en se basant sur ces nouveaux éléments géologiques, un terrain varié.

D'autre part, l'érosion générée par simulation étant basée sur des phénomènes émergents, elle est difficile à prévoir et contrôler.

Contrôle

Comme remarqué en introduction, le contrôle de l'artiste sur le résultat de l'algorithme est un point à ne pas perdre de vue. Afin de pouvoir intégrer mon travail à celui d'A. Émilien, réalisant un éditeur interactif dans le cadre de sa thèse sur la génération procédurale, j'ai utilisé en donnée d'entrée le format qu'il manipule : des cartes vectorielles.



Exemple de carte d'entrée, inspirée de la Corse

La carte est volontairement simple afin de pouvoir être réalisée rapidement. À titre d'exemple, la carte précédente a été dessinée à l'aide d'Inkscape en à peine plus de 5 min. Elle comporte :

1. Des contours d'îles et continents ;
2. Des points de montagnes ;
3. Des rivières principales.

Réutiliser ce format permettait également de pouvoir utiliser les fonctions déjà codées pour l'import de fichiers et éviter de passer du temps sur des détails d'implémentation.

Temps réel

Un dernier point important, là encore pour des raisons d'interactivité, est la rapidité de calcul. L'idéal aurait été de pouvoir générer en temps réel le terrain afin qu'il s'intègre dans l'éditeur d'A. Émilien. Cela m'a donc mené à faire un certain nombre d'approximations tout en essayant de conserver un résultat visuellement acceptable, la finalité étant le justement le rendu et non la simulation.

Reconstitution de plaques

Le premier sous-objectif que je me suis donné a été de déterminer à partir des données d'entrée un ensemble cohérent de plaques tectoniques afin de pouvoir dans un second temps simuler leur effet.

Géologie

J'ai donc tout naturellement étudié comment les géologues ont tracé les contours des plaques terrestres. Malheureusement, ces derniers ne partaient pas du même type d'information que moi. Les données les plus utilisées sont en effet la position des séismes et le type des roches dans les différentes zones, déterminées généralement par forage [Steinberger12]. Je ne pouvais évidemment pas demander ce genre d'information à l'utilisateur et les travaux géologiques m'ont finalement donné assez peu d'information sur les liens **directs** entre la position des régions montagneuses ou des côtes et les limites de plaques et autres éléments géologiques.

Simplification

Il a donc fallu chercher des heuristiques un peu fortes. J'ai alors étudié des cartes géologiques et des reconstitutions des continents sur Terre ou sur d'autres planètes (Europa [Kattenhorn09]) pour essayer d'en retirer des règles simples de découpage.

Je n'ai trouvé aucune méthode visant à déterminer des limites de plaques autrement que par découpage aléatoire. [Viitanen12] donne une bonne base pour comprendre les phénomènes géologiques mis en jeu, bien que les méthodes proposées par la suite manquent d'approfondissement.

Données utilisées

En observant des données réelles, il apparaît que la trajectoire des rivières est peu corrélée avec le découpage des plaques. D'autre part, ce sont — comme on l'imagine aisément — les montagnes qui fournissent le plus d'information. Je me suis donc contenté d'exploiter uniquement les positions des montagnes pour délimiter les plaques.

Prendre en compte deux types de donnée aussi différents que les contours des continents (lignes polygonales) et les montagnes (points) rendrait la détermination d'une heuristique bien plus ardue.

Remarque

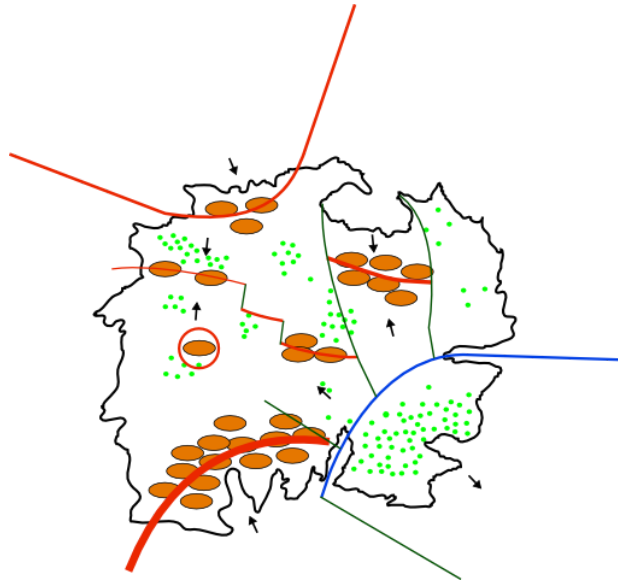
De façon générale, j'ai toujours cherché à commencer par mettre en place la suite des étapes clefs, quitte à simplifier ces étapes, plutôt que de passer trop de temps sur l'une d'elles et donc avoir un résultat déséquilibré, voire pas de résultat du tout si l'une des étapes venait à manquer. Cette méthode peut parfois donner l'impression de ne pas approfondir suffisamment, mais est au final bien plus satisfaisante puisqu'elle permet d'avoir de premiers résultats assez vite et de voir donc le projet évoluer. Cela force également à coder de façon modulaire pour pouvoir remplacer des éléments a posteriori.

Notons également que le découpage des plaques est une étape préliminaire qui ne requiert pas une grande précision puisque c'est plus ses conséquences que le découpage lui-même qui nous intéresse. Il ne doit par conséquent pas nécessiter des calculs trop importants.

Voies explorées

Essai manuel

La première chose à faire était de dessiner à la main ce que je voulais que mon programme obtienne, essayant ainsi de trouver par introspection comment mon instinct décidait d'appliquer telle ou telle découpe.



Découpage à la main. Inclut des frontières convergentes (rouge), divergentes (bleu), et des glissements (vert).

Analogie physique

J'ai ensuite cherché des similitudes avec d'autres problèmes. Étant donné que je cherchais à déterminer un champ de vitesse, j'ai pensé m'inspirer de certains problèmes de la physique des champs. Le champ recherché devait être à circulation partout nulle sauf en les frontières des plaques. Mais je n'ai pas trouvé de problème assez proche sur lequel me baser et les quelques tests faits à partir de formules simples ne permettait pas d'imposer simplement le fait que les contours des plaques (les zones à circulation non nulle) devaient être des courbes fermées.

Mini-plaques

Plutôt que de partir d'une grande plaque et de chercher où la couper, j'ai alors décidé de partir d'un grand nombre de petites cellules et de les fusionner jusqu'à constituer des plaques de taille convenable (où « convenable » est déterminé par un seuil dépendant de l'échelle à laquelle on considère la carte d'entrée).

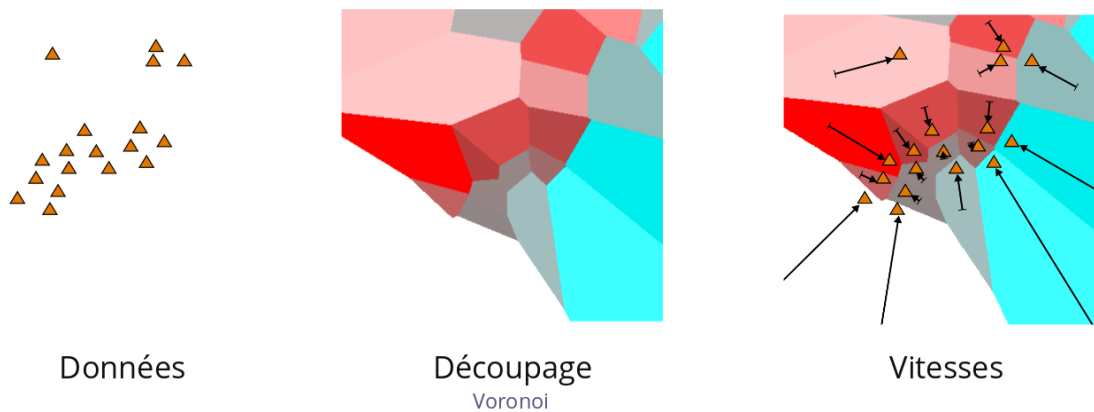
Découpage

Le découpage que j'ai décidé d'utiliser est celui de Voronoi, à partir des montagnes. Il présente l'avantage d'être rapide à calculer sur GPU [Hoff III99] et me permet d'estimer une information importante pour la suite : le mouvement de chaque petite plaque.

Le modèle proposé pour assigner un mouvement aux cellules de Voronoi est de considérer le vecteur allant du centre de gravité de la cellule à sa montagne de référence (celle ayant servi à construire la cellule, c'est-à-dire la plus proche de tout point de celle-ci).

Ainsi, les mouvements des plaques se dirigent naturellement vers les groupes de montagnes. En effet, la cellule d'une montagne se prolonge dans les directions où celle-ci n'a pas de voisine mais est vite arrêtée lorsqu'il y a une autre montagne proche. Le vecteur résultant est donc dirigé, pour chaque plaque, vers les zones montagneuses.

Bien sûr, c'est un mouvement comme un autre, mais il est assez convaincant et se comprend aisément. Et puis une méthode procédurale sans une part d'arbitraire, ça devient de la simulation...



Création des cellules et attribution de leur vitesse.

Fusion

Une fois les cellules déterminées et orientées, elles sont fusionnées de proche en proche lorsque leur mouvement diffère peu (en pratique lorsque le produit scalaire de leur vitesse normalisée est supérieur à un certain seuil).

Afin d'éviter l'apparition d'artefacts (de petites plaques isolées par exemple), les vitesses sont pondérées par l'aire des plaques.

Limitations

Commençons par ce qui n'est pas une limitation : les frontières des plaques ne passent jamais par les montagnes. Par construction. Ce n'est pas une limitation puisque lorsqu'une plaque passe sous une autre, les chaînes de montagnes apparaissent parallèlement à la limite de plaques, en retrait.

D'autre part, les montagnes isolées ne sont sur aucune limite de plaque puisque leur cellule est centrée sur la montagne et se voit donc attribuer une vitesse très faible. Elle est alors vite absorbée par une autre plaque. Là encore, ce n'est pas une limitation puisqu'on peut supposer que les montagnes isolées sont dues à un point chaud, et donc indépendantes des limites de plaques. On peut même choisir de les marquer comme volcaniques.

Le découpage atteint ses limites lorsqu'il est confronté à une ligne de montagnes simple. On peut penser que par une telle ligne l'utilisateur suggère un massif allongé qui correspond instinctivement à une limite de plaque. Pourtant, si cette ligne n'est pas un amas de points, le découpage de Voronoi ne fait apparaître aucune frontière alignée avec ces montagnes. Une solution un peu pédestre est alors de démultiplier les montagnes mais cette solution élimine les deux comportements précédemment décrits.

Ce problème aurait pu être évité en utilisant une triangulation de Delaunay, dual du diagramme de Voronoi. Mais on perd alors la méthode d'attribution de leur vitesse aux plaques.

D'autre part, on traite les vitesses comme *absolues* alors que la présence de montagne indique une convergence, c'est à dire un mouvement *relatif*. Cela fait parfois apparaître des limites divergentes superflues.

Il n'est pas non plus possible de considérer la rotation des plaques. Mais bon, c'est un effet beaucoup moins important que la translation à court terme. Enfin court terme pour les mouvements de plaque, donc bien assez long pour notre problème.

On peut également noter que la position des bords de l'image joue un rôle dans l'attribution des vitesses des plaques. Cependant, l'absence d'information sur les zones en dehors de l'image permet de s'autoriser à supposer ce que l'on veut à leur propos. Ce sera un problème seulement le jour où on cherchera à calculer ces limites par morceaux (pour traiter de très grandes cartes par exemple).

La dernière, mais pas des moindres, des limitations est l'absence de prise en compte des types de plaques (océaniques et continentales). C'est un point que j'aurais aimé traiter mais qui nécessitait d'ajouter l'exploitation des contours des continents, ce qui n'est toujours pas fait à ce jour.

Construction des plis

Une fois les plaques constituées et orientées, il m'a fallu trouver comment répercuter cette information sur le relief, en créant des plis, des failles, etc.

Une fois encore, on trouve peu d'antécédents dans le domaine de la génération procédurale, et celles que l'on trouve sont complexes, à la limite de la simulation [Natali13], pour des résultats parfois peu différents des autres méthodes [Alex06].

D'autre part, toutes ces méthodes cherchent à produire au plus un élément géologique et non un ensemble. Elles ne travaillent donc pas à l'échelle que je recherche.

À nouveau, je suis allé voir des articles plus orientés vers la géologie. Un certain nombre sont des simulations numériques dont le but est la précision, et non la vraisemblance, et prennent alors en compte un grand nombre de phénomènes physiques, la courbure de la Terre, etc. Des méthodes bien trop lourdes pour mon objectif qui, rappelons le, est entre autres la génération en temps réel.

Certains articles m'ont permis de trouver des détails sur l'aspect des plis et failles et autres recouvrements de strates à différentes échelles [Malaveille10], [Calcagno08]. J'ai également essayé de visualiser des données réelles à l'aide du logiciel GPlate, utilisé en géodynamique [Boyden11].

Voies explorées

Simulation

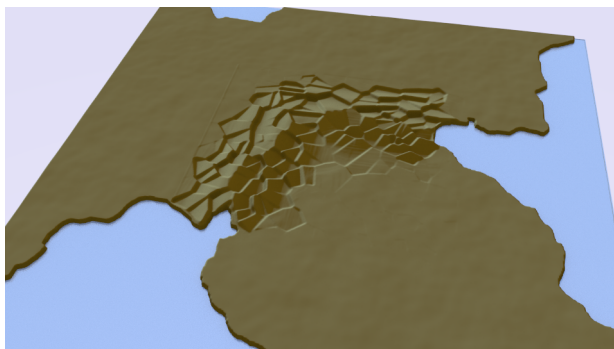
J'ai commencé par chercher à m'inspirer des méthodes de simulation utilisées en géologie en les simplifiant drastiquement à l'aide de règles tirées des observations dans les articles précédents.

Seulement, les simulations physiques sont longues à calculer. Et longues à coder lorsqu'on ne l'a jamais fait. J'ai passé un certain temps à étudier les méthodes classiques de résolution d'équations différentielles (éléments finis, Runge-Kutta, etc.) puis ai fini par abandonner cette idée, considérant que j'avais passé déjà bien assez de temps sur le sujet.

D'autant plus que pour obtenir un résultat complet par simulation, il est nécessaire de prendre en compte un grand nombre d'étapes qui se succèdent, à différentes périodes. Les simuler toutes prendrait un temps bien trop long, alors qu'elles sont vite estompées et que seuls certains éléments en restent.

Craquèlement

Cherchant une approche totalement différente, l'observation des grandes plaques inclinées de la Chartreuse m'a menée à découper en petites plaques le terrain (par un Voronoi aléatoire par exemple) pour le craqueler en inclinant ces plaques au voisinage des frontières convergentes et les abaissant aux frontières divergentes pour reproduire l'effondrement.



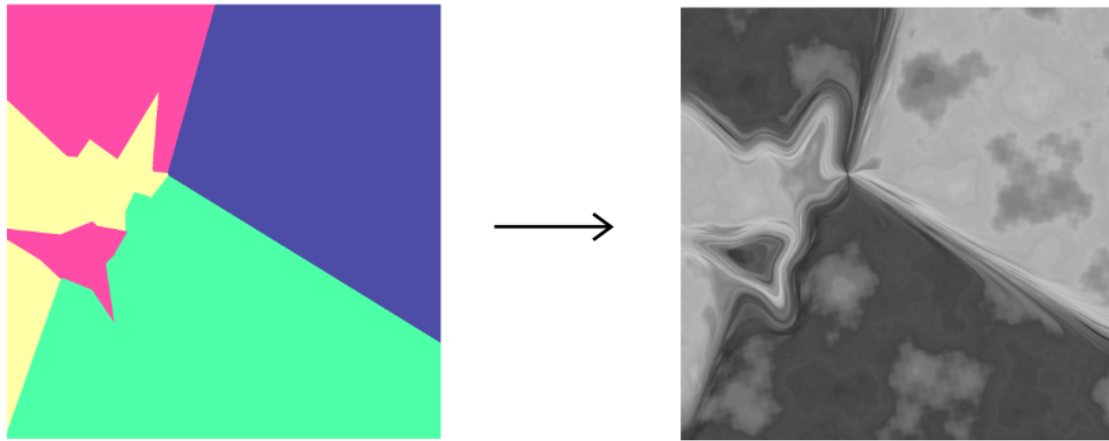
Craquèlement du terrain sur une limite de plaque dessinée à la main.

Le principal problème est que cette méthode fait apparaître un terrain totalement plat en dehors des limites des plaques. Or, il s'agissait de réduire la monotonie des terrains générés donc là, c'est raté.

De plus, le résultat est très dépendant de la taille des craquelures et celle-ci est totalement arbitraire. La méthode ne donne donc qu'un seul niveau de détails, ce qui est peu satisfaisant.

Plis par distorsion de bruit

Un moyen d'obtenir un résultat multi-échelle est d'utiliser des éléments fractales. J'ai donc généré des plis à partir d'un *mouvement Brownien fractionnaire* en le distordant selon la carte de vitesse des plaques.



Distorsion à partir de la carte de vitesse

Le résultat ainsi obtenu est extrêmement rapide à calculer (temps constant pour chaque pixel), multi-échelle, facilement animable (en jouant sur l'amplitude de la distorsion) et simple à mettre en place.

Limitations

Cette méthode fait apparaître, comme le montre l'image précédente, des éléments de relief forts en dehors des frontières de plaques. Cette carte est donc multipliée par un coefficient décroissant en la distance à une frontière de plaque.

Une limitation plus fâcheuse est l'absence de distinction entre frontières convergentes et divergentes. Les plis apparaissent partout. Là encore, la multiplication par une carte est une solution.

Les plis peuvent paraître un peu trop réguliers, mais on peut jouer sur ce point en ajoutant des octaves au bruit. Ce n'est cependant pas nécessaire car à plus petite échelle, c'est l'érosion implémentée par la suite qui prend le dessus pour dessiner les détails.

Un dernier point : l'animation obtenue est assez peu réaliste. Certains niveaux de détail trop fins font un effet de vague parcourant le terrain par exemple. Mais l'animation n'est pas la priorité.

Construction du terrain final

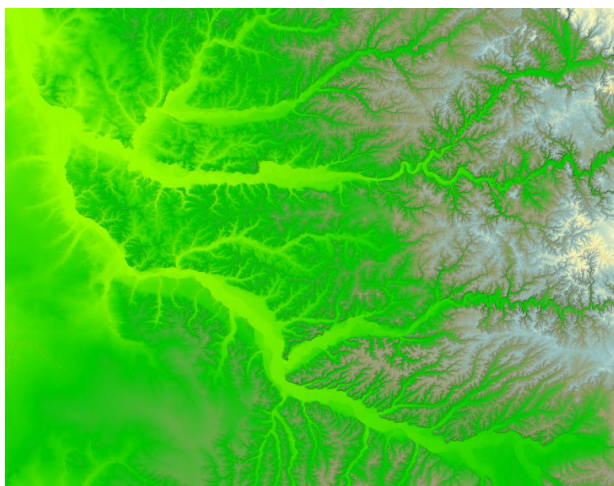
À partir de cette carte de pli, essentiellement deux possibilités s'offraient à moi :

- L'utiliser pour influencer des méthodes procédurales du type de **[Génevaux13]**.
- Lui appliquer divers algorithmes d'érosion.

Bien que l'algorithme de **[Génevaux13]** prenne en entrée une carte permettant de spécifier des zones montagneuses, il ne permet pas à cette carte d'avoir un grand niveau de détail. En effet, celle-ci permet de spécifier où placer des montagnes, mais n'impose rien sur la forme à leur donner. Les détails de la carte de plis étant de l'ordre de la taille des montagnes, ils seraient perdus.

Érosion

Même si prendre en compte les phénomènes géologiques comme les plis est bien pour les détails à grande échelle, ça ne donne rien à petite échelle. Les plis doivent servir de base à l'érosion, qui est le phénomène principal dans le façonnement des paysages.



Forte influence de l'érosion en Aquitaine

L'érosion me permet en plus :

- D'estomper les plis là où l'utilisateur ne veut pas de plis. On respecte donc mieux la contrainte.
- De prendre en compte les rivières (*c.f.* partie suivante).

L'algorithme d'érosion doit donc être influençable par des paramètres extérieurs, et non complètement autonome. J'ai commencé par utiliser l'algorithme de [Pytel13], mais sa violation de certains principes physiques comme la conservation de la matière le rend assez complexe à modifier. Il est donc difficile de prévoir intuitivement les conséquences de certaines modifications.

Lorsque l'algorithme respecte des contraintes physiques, on sait quelles contraintes sont impératives pour la stabilité du système et quelles contraintes peuvent être contournées. Sinon, étant donné que l'on joue sur des phénomènes émergents, on ne peut rien prévoir.

J'ai finalement utilisé la méthode présentée par [Št'ava08] comme base. À l'instar de beaucoup de méthodes, elle utilise une simulation d'eau par colonnes pour le transport de sédiments. Elle inclut ensuite l'effondrement au-delà d'un angle limite, sur lequel il est possible de jouer pour pouvoir éroder plus les zones sur les frontières de plaques où l'utilisateur n'avait pas spécifié la présence de montagnes et de varier les paysages.

Un autre avantage de l'algorithme de [Št'ava08] est qu'il est décrit de sorte à être codé sur GPU, ce qui permet d'accélérer grandement les choses. Cependant, l'objectif du rendu en temps réel est de plus en plus difficile à remplir avec l'ajout de l'érosion.

Un des travaux les plus chronophages du traitement de l'érosion, outre l'implémentation de diverses méthodes, fut de déterminer les paramètres les plus adaptés. Ils ne varient pas tous de la même façon au passage à l'échelle et sont fortement interdépendants.

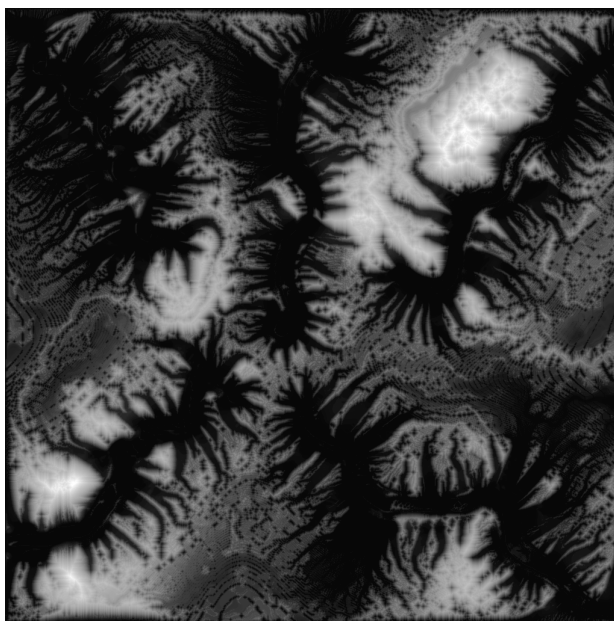
D'autre part, j'avais assez peu l'habitude de coder sur GPU et à gérer son grand silence face aux erreurs comme les divisions par zéro.

Influence des rivières

La carte donnée en entrée prenait en compte les montagnes, par l'intermédiaire des plis, ainsi que les contours des continents, ajoutés aux plis, mais pas les rivières dessinées par l'utilisateur. C'est pour les prendre en compte que l'algorithme d'érosion devait permettre une certaine souplesse.

Plusieurs méthodes ont été envisagées afin d'imposer ces rivières :

- Attirer l'eau en ajoutant une force allant vers les rivières. Le problème est que l'eau s'accumule dans les rivières, et avec elle tous les sédiments du terrain (à long terme). J'ai essayé d'y remédier en supprimant les sédiments dans les rivières, contre la règle de conservation de la matière. Le système reste stable à condition d'imposer une altitude minimum et les rivières ressemblent alors plus à des fjords. Dans certains cas, ce peut être le but recherché, mais cette idée manque de souplesse et ne permet en aucun cas d'imposer de petites rivières.



Carte de hauteur générée en attirant l'eau dans les rivières.

- Rendre les précipitations plus importantes au niveau des rivières. Cette méthode ne permet pas de faire traverser aux rivières les arrêtes importantes car l'eau les quitte très rapidement. Elle les ravine sans en modifier la crête.
- Prendre en compte les rivières dès la création de la carte de relief initiale. Le principal problème étant de la coupler avec la carte de plis de façon cohérente. C'est la méthode que j'ai finalement adoptée.

La carte initiale doit laisser à l'eau la possibilité de s'écouler afin d'éviter la formation d'océans non voulus. Elle est construite par diffusion de hauteur depuis les océans en suivant le cours des rivières. Les phénomènes de diffusion sont souvent utilisés lors de la programmation GPU car ils sont fortement parallèle et nécessite un nombre important d'itérations, deux points pour lesquels les cartes graphiques ont été conçues.

La carte ainsi obtenue est alors totalement indépendante des plis. Il faut donc fusionner les deux cartes tout en prenant soin de conserver la possibilité d'écoulement (au moins à une certaine échelle car au-delà l'érosion peut corriger le problème) et l'aspect graphique des plis. Ce mélange se fait alors en fonction de la distance à la rivière la plus proche : à proximité des rivières et océans, la carte de rivière est prise en compte fortement. Puis plus on s'éloigne, plus la carte de plis joue un rôle important. Il y a bien sûr un certain nombre de paramètres à ajuster lors de cette étape afin de prendre en compte plutôt l'une ou l'autre des cartes.

Il en résulte que les plis sont essentiellement visibles dans les zones montagneuses et que dans les plaines le terrain est principalement façonné par l'érosion, ce qui est en accord avec l'observation de terrains réels.

Limitations

Bien que les fleuves principaux tels que dessinés par l'utilisateur apparaissent effectivement, il ne sont pas les seuls de leur envergure. C'est un avantage lorsque l'utilisateur spécifie pas ou peu de rivières, mais ce peut également être un problème dans d'autres cas.

Lorsque les rivières sont trop longues, la diffusion, qui les fait monter en pente constante depuis les océans, les mène à être plus haute que les régions voisines. Il faudrait normaliser la pente en prenant en compte la longueur des rivières, mais une simple diffusion ne permet pas un tel calcul puisqu'elle n'agit qu'à une échelle locale.

Couplage mouvement/érosion

La méthode de création des plis permettant l'animation, j'ai essayé de coupler ce mouvement avec l'érosion. Le résultat est cependant peu convaincant. Les mouvements interviennent en réalité à une échelle de temps si différente qu'il n'est pas réaliste de simuler les deux en même temps. Et, si le mouvement est fortement ralenti, la carte de hauteur initiale est effacée avant que le terrain final soit constitué.

Rendu

Les premiers calculs que j'ai effectués se faisaient en boîte noire : je lançais le calcul, attendais parfois plusieurs

dizaines de secondes qu'il se fasse, puis observait le résultat final.

C'est la façon la plus simple de faire les choses, mais ça rend le débogage et les tests plus longs. Le résultat final n'est généralement pas la seule information intéressante, surtout dans le cadre de phénomènes émergents où il est important de voir « naître » les anomalies. D'autre part, il n'est pas simple d'estimer théoriquement au bout de combien d'itérations arrêter la simulation.

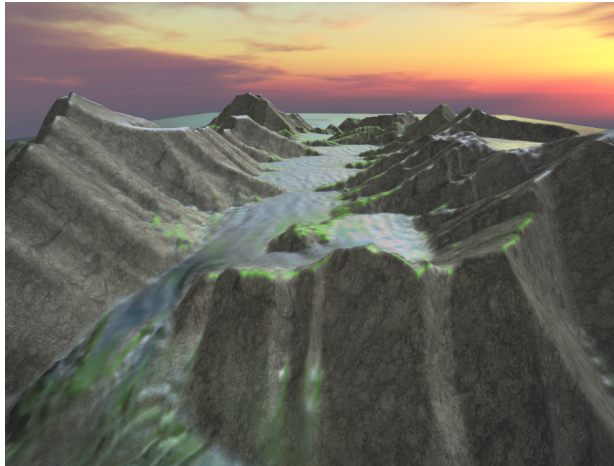
On peut alors essayer d'arrêter le calcul plus tôt afin de déterminer, éventuellement par dichotomie, à quel moment un problème survient, mais c'est fastidieux.

Quant à enregistrer régulièrement l'état de la simulation, cela rend le calcul encore plus long.

A. Émilien m'a donc aidé à mettre en place un rendu 3D en temps réel de la scène en cours de calcul. Non seulement cela a permis de voir le développement de l'érosion, mais également d'interagir directement avec elle en ajoutant par exemple de l'eau en cliquant sur une région.

Le code pour GPU étant compilé à la volée, il est également possible de modifier directement les règles d'érosion *via* un *shader* et d'en voir le résultat en temps réel.

Résultats



Montagnes pendant la simulation d'érosion.

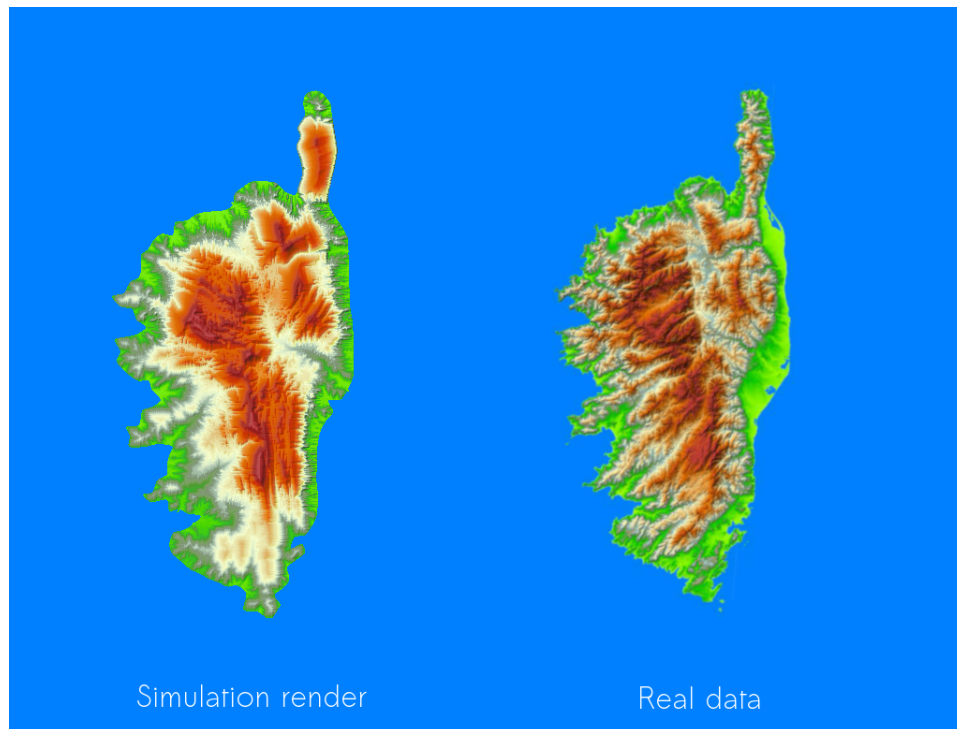
Corse

Voici les résultats donnés avec pour entrée la carte inspirée de la Corse donnée en exemple au début de ce rapport :



Rendu 3D pendant la simulation d'érosion.

J'ai appliqué à la carte de hauteur résultante un dégradé et un ombrage inspirés de ceux de GéoPortail afin de comparer avec des données réelles.

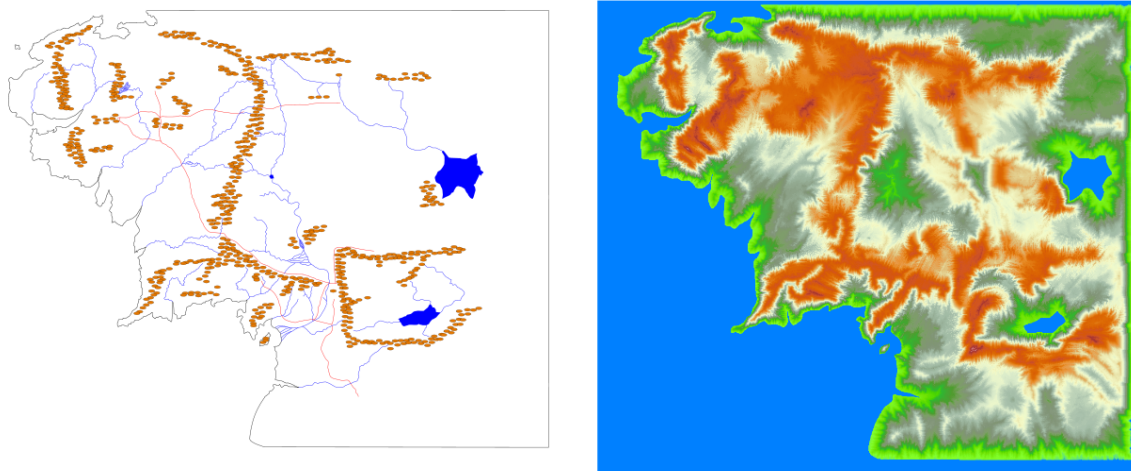


Comparaison entre la carte obtenue à partir du rapide schéma vectoriel et des données réelles.

Le relief n'est évidemment pas le même, mais on voit sur la carte générée un effet d'ensemble : des plis alignés verticalement apparaissent. Ils suivent la chaîne grossièrement tracée par les montagnes de la carte vectorielle.

Carte de plus grande ampleur : Terres du Milieu

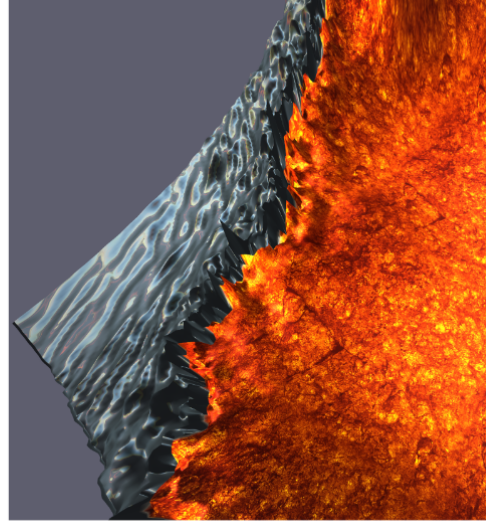
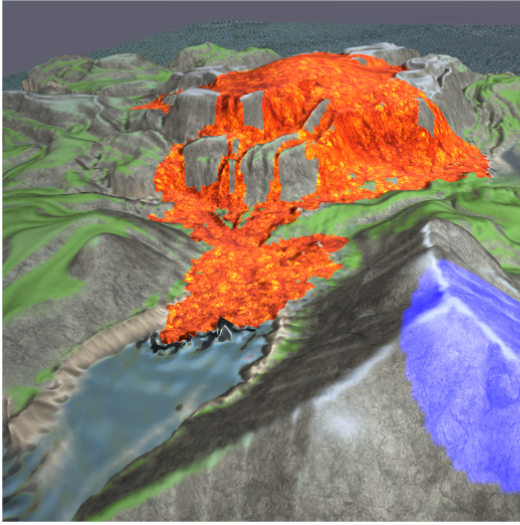
J'ai ensuite testé l'algorithme sur une carte plus complexe, celle des Terres du Milieu (monde imaginé par J.R.R. Tolkien, théâtre d'une grande partie de ses œuvres). Il n'existe pas de carte réelle avec laquelle comparer, mais le but est justement de pouvoir faire le rendu de terrains imaginaires de ce type.



Génération d'une carte de hauteur inspirée des Terres du Milieu

Volcans

Les volcans ne sont pas gérés du tout par la simulation, bien que j'ai essayé de faire parcourir une sorte de point chaud sur le carte, mais j'ai ajouté la possibilité de mettre de la lave sur le terrain. Ça m'a servi en particulier à pouvoir ajouter de la matière dans la vue interactive pour pouvoir faire des tests.



Volcan

Améliorations possibles

Outre les limitations des différentes méthodes exposées ci-avant, de nombreux points peuvent être explorés en partant de l'état actuel de mon travail.

- Un des inconvénients révélés par l'exemple de la Terre du Milieu est l'impossibilité de distinguer des petites et des grandes montagnes. Il serait intéressant d'ajouter un paramètre d'amplitude aux données d'entrées.
- J'ai suggéré que les montagnes isolées pouvaient être considérées comme des volcans, mais l'algorithme n'en fait rien. Il pourrait ajouter des montagnes à l'aide de la lave en ces points.
- Prendre en compte les contours des continents pour le calcul des plaques me semble un point important, mais je n'ai pas encore trop d'idée sur comment m'y prendre.
- Les rifts océaniques sont assez différents des frontières continentales. Il est possible de les dessiner assez facilement en considérant le squelette morphologique des océans puis en le mettant « en escalier ». Le phénomène d'escalier n'est pas encore expliqué par les géologues mais assez flagrant.
- L'algorithme de [Št'ava08] est prévu pour prendre en compte plusieurs strates de différents types de roches. Je n'exploite pas totalement cette possibilité car je n'ai dans mon implémentation que deux couches : le lit et les sédiments. Ajouter des couche initialement déformées par les plis peut donner des résultats intéressants, ne serait-ce que sur le rendu final des textures.
- La distorsion est, je pense, une bonne base. Il est facile à comprendre et donc à manipuler, rapide et donne des plis intéressants. Mais il subsiste un grand nombre de défauts potentiellement simples à éliminer : zones de divergences, découpage net de plaques dans les animations, etc.

Conclusion

J'ai choisi ce stage principalement par goût pour la génération procédurale, et en particulier celle de montagnes. Le domaine de l'informatique graphique est un des premiers domaines de l'informatique auquel j'ai été confronté et qui me tient à cœur. Et je ne me suis pas trompé : les travaux de recherche que j'ai mené correspondent à l'idée que je me faisais de la génération procédurale de paysages.

D'autre part, j'ai pu prendre pleinement conscience du monde de la recherche. J'y suis déjà confronté à l'École, mais y travailler et discuter quotidiennement avec ses acteurs (thésards, post-docs, chercheurs, etc) m'a fait découvrir pas mal de choses.

Je pense aussi avoir progressé techniquement. J'ai été amené à manipuler un code écrit par d'autres, j'ai pu perfectionner ma maîtrise du C++, du GLSL et de la logique un peu différente de la programmation sur GPU, etc. C'était également un point auquel je tenais.

Et je dois reconnaître que ce stage restera aussi un bon souvenir ! Je tiens d'ailleurs à remercier une fois encore tout l'équipe de là-bas pour l'accueil chaleureux et la bonne ambiance.

Ressources

- [Mandelbrot68] Mandelbrot, B. B., & Van Ness, J. W. (1968). Fractional Brownian motions, fractional noises and applications. *SIAM review*, 10(4), 422-437.
- [Fournier82] Fournier, A., Fussell, D., & Carpenter, L. (1982). Computer rendering of stochastic models. *Communications of the ACM*, 25(6), 371-384.
- [Perlin85] Perlin, K. (1985). An Image Synthesizer. In *SIGGRAPH '85: Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques* (Vol. 19, pp. 287-296). ACM.
- [Kelley88] Kelley, A. D., Malin, M. C., & Nielson, G. M. (1988). *Terrain simulation using a model of stream erosion* (Vol. 22, No. 4, pp. 263-268). ACM.
- [Musgrave89] Musgrave, F. K., Kolb, C. E., & Mace, R. S. (1989, July). The synthesis and rendering of eroded fractal terrains. In *ACM SIGGRAPH Computer Graphics* (Vol. 23, No. 3, pp. 41-50). ACM.
- [Prusinkiewicz93] Prusinkiewicz, P., & Hammel, M. (1993, May). A fractal model of mountains and rivers. In *Graphics Interface* (Vol. 93, pp. 174-180). CANADIAN INFORMATION PROCESSING SOCIETY.
- [Hoff III99] Hoff III, K. E., Keyser, J., Lin, M., Manocha, D., & Culver, T. (1999, July). Fast computation of generalized Voronoi diagrams using graphics hardware. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (pp. 277-286). ACM Press/Addison-Wesley Publishing Co..
- [Beneš01] Beneš, B., & Forsbach, R. (2001). Layered data representation for visual simulation of terrain erosion. In *Computer Graphics, Spring Conference on*, 2001. (pp. 80-86). IEEE.
- [Beneš02] Beneš, B., & Forsbach, R. (2002). Visual simulation of hydraulic erosion.
- [Belhadj05] Belhadj, F., & Audibert, P. (2005, November). Modeling landscapes with ridges and rivers: bottom up approach. In *Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia* (pp. 447-450). ACM.
- [Alex06] Alex, J. E. (2006). Creating landscapes with simulated colliding plates.
- [Mei07] Mei, X., Decaudin, P., & Hu, B. G. (2007, October). Fast hydraulic erosion simulation and visualization on GPU. In *Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on* (pp. 47-56). IEEE.
- [Zhou07] Zhou, H., Sun, J., Turk, G., & Rehg, J. M. (2007). Terrain synthesis from digital elevation models. *Visualization and Computer Graphics, IEEE Transactions on*, 13(4), 834-848.
- [Calcagno08] Calcagno, P., Chilès, J. P., Courrioux, G., & Guillen, A. (2008). Geological modelling from field data and geological knowledge: Part I. Modelling method coupling 3D potential-field interpolation and geological rules. *Physics of the Earth and Planetary Interiors*, 171(1), 147-157.
- [Št'ava08] Št'ava, O., Beneš, B., Brisbin, M., & Křivánek, J. (2008, July). Interactive terrain modeling using hydraulic erosion. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (pp. 201-210). Eurographics Association.
- [Kattenhorn09] Kattenhorn, S. A., & Hurford, T. (2009). Tectonics of Europa. *Europa*, 199-236.
- [Kristof09] Krištof, P., Beneš, B., Křivánek, J., & Št'ava, O. (2009, April). Hydraulic erosion using smoothed particle hydrodynamics. In *Computer Graphics Forum* (Vol. 28, No. 2, pp. 219-228). Blackwell Publishing Ltd.
- [Smelik09] Smelik, R. M., De Kraker, K. J., Tutenel, T., Bidarra, R., & Groenewegen, S. A. (2009, June). A survey of procedural methods for terrain modelling. In *Proceedings of the CASA Workshop on 3D Advanced Media In Gaming And Simulation (3AMIGAS)* (pp. 25-34).
- [Hnaidi10] Hnaidi, H., Guérin, E., Akkouche, S., Peytavie, A., & Galin, E. (2010, September). Feature based terrain generation using diffusion equation. In *Computer Graphics Forum* (Vol. 29, No. 7, pp. 2179-2186). Blackwell Publishing Ltd.
- [Malaveille10] Malaveille, J. (2010). Impact of erosion, sedimentation, and structural heritage on the structure and kinematics of orogenic wedges: Analog models and case studies. *GSA Today*, 20(1), 4-10.

- [Boyden11]** Boyden, J. A., Müller, R. D., Gurnis, M., Torsvik, T. H., Clark, J. A., Turner, M., ... & Cannon, J. S. (2011). Next-generation plate-tectonic reconstructions using GPlates. *Geoinformatics: cyberinfrastructure for the solid earth sciences*, 95-114.
- [Steinberger12]** Steinberger, B., Torsvik, T. H., & Becker, T. W. (2012). Subduction to the lower mantle—a comparison between geodynamic and tomographic models. *Solid Earth*, 3(2), 415-432.
- [Viitanen12]** Viitanen, L. (2012). Physically Based Terrain Generation: Procedural Heightmap Generation Using Plate Tectonics.
- [Génevaux13]** Génevaux, J. D., Galin, E., Guérin, E., Peytavie, A., & Beneš, B. (2013). Terrain generation using procedural models based on hydrology. *ACM Transactions on Graphics (TOG)*, 32(4), 143.
- [Natali13]** Natali, M., Lidal, E. M., Viola, I., & Patel, D. (2013). Modeling terrains and subsurface geology. *Proceedings of EuroGraphics 2013 State of the Art Reports (STARs)*, 155-173.
- [Pytel13]** Pytel, A., & Mann, S. (2013). Self-organized approach to modeling hydraulic erosion features. *Computers & Graphics*, 37(4), 280-292.