Ludovic D’ANjou-MAdore

Simon Lepage

Jérôme Pagé

Jonathan Simard

**Projet d'intégration en Sciences informatiques et mathématiques**

201-201-RE, gr.00001

Curved\_Fractals

**Document de Conception**

Travail présenté à

M. Walid Boulabiar

Département d'Informatique

Cégep Limoilou

Le 09/02/2018

Table des matières

[Introduction 1](#_Toc505332206)

[Titre de section 1](#_Toc505332207)

[Titre de sous-section 1](#_Toc505332208)

[Titre de sous-sous-section 1](#_Toc505332209)

[Conclusion 1](#_Toc505332210)

[Annexe 2](#_Toc505332211)

[Médiagraphie 3](#_Toc505332212)

# Description du projet

**Objectifs**

Un programme de visualisation de fractales dans un espace non-euclidien (donc doté de courbure) permettrait d’introduire une compréhension de concepts mathématiques abstraits et complexes (Courbure d’une variété riemannienne et l’influence d’une métrique sur celle-ci) en permettant à l’utilisateur de visualiser en temps réel la courbure qui découle d’une métrique spécifiée. Une compréhension des fractales est acquise en même temps.

**Description détaillée**

***Concepts présents dans l’application***

L’application met en application premièrement les concepts mathématiques suivants :

* Les Variétés Riemanniennes dans leurs constructions.
  + La spécification d’une métrique comme généralisation du produit scalaire
    - au point p
  + L’expression des Symboles de Christoffel en fonction du tenseur métrique
    - [https://fr.wikipedia.org/wiki/Symboles\_de\_Christoffel#Expression\_%C3%A0\_partir\_du\_tenseur\_m%C3%A9trique](https://fr.wikipedia.org/wiki/Symboles_de_Christoffel#Expression_à_partir_du_tenseur_métrique)
  + L’influence des symboles de Christoffel sur la dérivée covariante
    - https://fr.wikipedia.org/wiki/D%C3%A9riv%C3%A9e\_covariante#Champ\_de\_vecteurs\_2
  + L’influence de la dérivée covariante sur le tenseur de courbure
    - [https://fr.wikipedia.org/wiki/Courbure#D%C3%A9finition\_du\_tenseur\_de\_courbure](https://fr.wikipedia.org/wiki/Courbure#Définition_du_tenseur_de_courbure)
  + Calcul d’une immersion de l’espace qui contient les fractales dans l’espace écran
    - <https://en.wikipedia.org/wiki/Riemannian_manifold#The_pullback_metric>

Les fractales seront transformés point par point en fonction de la métrique pour obtenir l'effet de courbure.

* + La transformation ce fait soit par un opérateur de forme qui passe par la courbure sectionnelle (TODO)
  + ou par l'endomorphisme de courbure construit à partir du tenseur de courbure.
  + Interpolation linéaire pour passer d’une description discrète à une description linéaire du champ tensoriel métrique :
    - pour 0<= i <=1

On remarque bien, à la complexité des équations en jeux, de l'utilisation de concepts complexes comme la dérivation covariante jusqu'au nombre effarent de terme, une fois la notation d'Einstein enlevée, que la transmission ce celles-ci au GPU sera un problème technique en soi et que sans cette transmission, la seule alternative est de les calculer sur le CPU, perdant beaucoup des avantages d'une carte graphique...

Ensuite, les concepts informatiques utilisés dans l'application sont principalement ceux reliés à l'imagerie par ordinateur. En effet, pour bénéficier du plein potentiel des cartes graphiques modernes, la majorité des calculs concernant le calcul de la fractale et la transformation de celle-ci en fonction de la courbure seront exécutés sur le GPU, par l'entremise des Shaders, des petits programmes exécutés pour chaque sommet d'un modèle et pour chaque pixels de l’écran, parallélisés, qui permettent d'effectuer les transformations (ici appliquer la courbure sur la fractale) et effets (ici l'affichage de la fractale en soi) voulus de manière extrêmement efficace. De plus, envoyer une partie substantielle des calculs sur la carte graphique a des effets bénéfiques d'un point de vue technique par exemple, en permettant d'alléger grandement la structure du modèle MVC.

**Concepteurs, rôles et justifications**

TODO

**Type D'application**

Windows

**Technologies impliquées**

Eclipse, GPU avec support pour openGL 3.2 et plus

Librairies : Jmonkey (openGL), JavaFX

**Références et documentation**

http://graphics.cs.ucdavis.edu/~joy/NSF-IIS-0916289/Papers/Obermaier2012b.pdf

# User Stories

|  |  |
| --- | --- |
| Acteur ou rôle : | Utilisateur |
| Scénario ou story : | En tant qu’utilisateur, je veux pouvoir spécifier les composantes du champ tensoriel métrique et voir les effets de la courbure de celui-ci sur une fractale |
| Détail ou description : | 1. Recevoir les chaines de caractères de l’utilisateur représentant les composantes du champ tensoriel métrique. 2. Valider la structure et la syntaxe des composantes et les interpréter pour construire le champ tensoriel métrique en Java. 3. Construire une texture 3D (2D Array Textures ou un tableau 2x2 de matrices 2x2 sur OGL 4.3 et +) en évaluant le tenseur à chaque pixel (les composantes x et y de la texture représentent la position de chaque pixel et le niveau (pour le 2DTextures Array) représente la nième composante du tenseur a un point (pour le tableau 2x2, en fixant la position, on obtient une matrice, la représentation locale du tenseur métrique). 4. Envoyer la structure de données choisie sur le shader et calculer la déformation pour chaque pixel. |
| Tests d’acceptation : | Afficher l’ensemble de Mandelbrot et confirmation visuelle de la déformation. |
| Complexité : | 7 |
| Effort : | 3j/personne **ou** 4 |
| Commentaires : | -Lors du changement de tenseur spécifié par l’utilisateur, on recalcule une texture et on met à jour la référence dans le shader  -lors d’un zoom, on doit « zoomer » aussi les coordonnés d’accès à la représentation en mémoire du tenseur. Donc, si on utilise une texture 3d, l’interpolation ce fait automatique pour chaque niveau. Si on utilise un tableau de matrice, dans chaque direction, on doit interpoler entre les composantes. (par exemple : voir l’interpolation linéaire dans la description des concepts) |

# Conclusion

Inscrire votre texte ici. Inscrire votre texte ici. Inscrire votre texte ici. Inscrire votre texte ici.

# Annexe

# Médiagraphie