Géodésiques et recherche du plus court chemin

1 Préambule

La recherche d'un chemin de plus petite longueur sur une surface, établie comme contrainte, représente un problème d'optimisation classique. L'objectif de l'étude présentée dans ce rapport est la mise en place d'un algorithme permettant d'obtenir un chemin de plus petite longueur entre deux points donnés d'une surface définie implicitement. Un algorithme d'affichage du chemin sur la surface par triangulation a été réalisé et sera utilisé mais ne sera pas présenté dans ce rapport.

2 Introduction

J'ai décidé de rechercher les courbes géodésiques reliant deux points donnés sur une surface définie implicitement. En effet, il est possible d'établir une équation différentielle vérifiée par ces courbes. L'équation différentielle permet de trouver, grâce à la méthode d'Euler, une courbe géodésique partant du point de départ sur la surface pour un vecteur vitesse initiale fixé. J'ai alors choisi de définir la fonction qui à une vitesse initiale associe l'écart entre le point d'arrivée désiré et le point d'arrivée obtenu pour cette vitesse initiale. On s'est ainsi ramené à un problème de minimisation.

3 Modalités de l'action

3.1 Contexte

Soit $F : \mathbb{R}^3 \to \mathbb{R}$ de classe C^3.
On définit S surface de $\mathbb{R}^3$ par $S = \{(x, y, z) : F(x, y, z) = 0\}$.
On suppose que $\forall x \in S,\ \vec{\nabla} F(x) \neq \vec{0}$.
On choisit $(a, b) \in S^2$ où a est le point de départ et b le point d'arrivée désiré.

3.2 Les géodésiques et leur équation différentielle

Définition. Soit $m : [0, 1] \to \mathbb{R}^3$ de classe C^2.
On dit que m est une géodésique de S si elle vérifie les propriétés suivantes :
- $\forall t \in [0, 1], m(t) \in S$
- $\forall t \in [0, 1], m''(t)$ colinéaire à $\vec{\nabla} F(m(t))$

Proposition. Si $m : [0, 1] \to S$ est une géodésique alors

$$\forall t \in [0, 1],\ m''(t) = -\frac{\langle m'(t) | H_F(m(t)) \cdot m'(t) \rangle}{\|\vec{\nabla} F(m(t))\|^2} \vec{\nabla} F(m(t))$$

où H_F est la hessienne de F.

Démonstration.

$$\forall t \in [0,1],\ F(m(t)) = 0$$

$$\forall t \in [0,1],\ \langle m'(t) | \vec{\nabla} F(m(t)) \rangle = 0$$

$$\forall t \in [0,1],\ \langle m''(t) | \vec{\nabla} F(m(t)) \rangle + \langle H_F(m(t)) \cdot m'(t) | m'(t) \rangle = 0$$

Or, comme m est une géodésique, $\forall t \in [0,1],\ \exists\, \alpha(t) \in \mathbb{R}$ tel que $m''(t) = \alpha(t)\vec{\nabla} F(m(t))$.

$$\text{Il vient, } \forall t \in [0,1],\ \alpha(t) = -\frac{\langle m'(t) | H_F(m(t)) \cdot m'(t) \rangle}{\|\vec{\nabla} F(m(t))\|^2}$$

$$\text{Enfin, } \forall t \in [0,1],\ m''(t) = -\frac{\langle m'(t) | H_F(m(t)) \cdot m'(t) \rangle}{\|\vec{\nabla} F(m(t))\|^2} \vec{\nabla} F(m(t)) \quad \textbf{(1)}$$

3.3 Utilisation de l'équation différentielle

On pose $m : \begin{cases} [0,1] \times T_{a,\vec{v}} \to S \\ (t, \vec{v}) \mapsto m(t, \vec{v}) \end{cases}$ où $T_{a,\vec{v}}$ est l'ensemble des vecteurs tangents à S en a.

Ici, $m(t, \vec{v})$ représente la position à l'instant t de la courbe vérifiant l'équation **(1)** avec pour conditions initiales : $\begin{cases} m(0) = a \\ m'(0) = \vec{v} \end{cases}$.

On cherche à minimiser la fonction $G : \begin{cases} T_{a,\vec{v}} \to \mathbb{R} \\ \vec{v} \mapsto \|m(1, \vec{v}) - b\| \end{cases}$

Soit $(\vec{v}_1, \vec{v}_2)$ une base orthonormée de $T_{a,\vec{v}}$ et $\vec{v} \in T_{a,\vec{v}}$ quelconque.
La méthode du gradient numérique requiert l'utilisation de $\vec{\nabla} G(\vec{v}) = \partial_{\vec{v}_1} G(\vec{v})\, .\vec{v}_1 + \partial_{\vec{v}_2} G(\vec{v})\, .\vec{v}_2$.
On définit :

$$\forall t \in [0,1],\ \phi_{\vec{v}_1}(t, \vec{v}) = \frac{d}{dh}(m(t, \vec{v} + h\vec{v}_1))_{h=0} = D_{\vec{v}_1}(m(t, \vec{v}))$$

Ainsi :

$$\phi_{\vec{v}_1}(0, \vec{v}) = \vec{0} \text{ car } \forall h, m(0, \vec{v} + h\vec{v}_1) = a.$$

et

$$\partial_{\vec{v}_1} G(\vec{v}) = \frac{\langle \phi_{\vec{v}_1}(1, \vec{v}) | m(1, \vec{v}) - b \rangle}{\|m(1, \vec{v}) - b\|} \quad \textbf{(2)}$$

On peut définir m sur $\mathbb{R} \times T_{a,\vec{v}}$ produit d'ouverts et on suppose m de classe C^2.
Par le théorème de Schwarz,

$$\forall t \in [0,1],\ \phi''_{\vec{v}_1}(t) = \frac{\partial}{\partial h}\left(\frac{\partial^2 m(t, \vec{v} + h\vec{v}_1)}{\partial t^2}\right)_{h=0}$$

On utilise **(1)** :

$$\phi''_{\vec{v}_1} = (\langle H_F(m)\cdot m'|m'\rangle)\left(2\frac{\langle H_F(m)\cdot\phi_{\vec{v}_1}|\vec{\nabla}F(m)\rangle}{\|\vec{\nabla}F(m)\|^4}\,\vec{\nabla}F(m) - \frac{H_F(m)\cdot\phi_{\vec{v}_1}}{\|\vec{\nabla}F(m)\|^2}\right) - \left(\langle H_F(m)\cdot\phi'_{\vec{v}_1}|m'\rangle + \langle H_F(m)\cdot m'|\phi'_{\vec{v}_1}\rangle + \langle I_F(m)\phi_{\vec{v}_1}m'|m'\rangle\right)\frac{\vec{\nabla}F(m)}{\|\vec{\nabla}F(m)\|^2}$$

(3)

où m, $\phi_{\vec{v}_1}$ et leurs dérivées sont évaluées en $(t,\vec{v})$.

Par ailleurs, $I_F(m)$ peut être identifiée à une matrice $3\times3\times3$ correspondant aux dérivées partielles d'ordre 3 de F.

3.4 Réalisation de l'algorithme

L'algorithme prend en argument N le nombre de points permettant la discrétisation du segment $[0,1]$, le point de départ a, le point d'arrivée b, la fonction F définissant la surface S et ϵ représentant l'écart désiré entre le point d'arrivée obtenu et b.

3.4.1 Initialisation

— Définition du gradient $\vec{\nabla}F$, de la hessienne H_F et de I_F pour la surface S
— Définition de la fonction gradient et évaluation du gradient au point de départ $\vec{\nabla}F(a)$
— Choix d'une base quelconque $(\vec{v}_x,\vec{v}_y)$ de $T_{a,\vec{v}}$ grâce aux coordonnées de $\vec{\nabla}F(a)$

```
if gradient[0] > 0.0001 :
    VX = [float(gradient[2])/100, 0, - float(gradient[0])/100]
    VY = [float(gradient[1])/100, -float(gradient[0])/100, 0]
elif gradient[1] > 0.0001 :
    VX = [0, float(gradient[2])/100, - float(gradient[1])/100]
    VY = [float(gradient[1])/100, -float(gradient[0])/100, 0]
else :
    VX = [0, float(gradient[2])/100, - float(gradient[1])/100]
    VY = [float(gradient[2])/100, 0, - float(gradient[0])/100]
```

FIGURE 3.1 – Extrait de l'algorithme

— Orthonormalisation de Schmidt pour obtenir une base orthonormée $(\vec{v}_1,\vec{v}_2)$ de $T_{a,\vec{v}}$
— Discrétisation de $[0,1]$ en N points et introduction d'un compteur de calculs de géodésiques
— Introduction des équations différentielles **(1)** et **(3)** utilisées dans la boucle
— Calcul d'un premier écart entre b et le point d'arrivée obtenu par résolution de **(1)** grâce à la fonction *scipy.odeint* pour $\vec{v}=\vec{v}_1+\vec{v}_2\in T_{a,\vec{v}}$

3.4.2 Méthode de minimisation

— Utilisation d'une boucle Tant que avec pour test : $ecart > \epsilon$

Le corps de la boucle est composé de plusieurs étapes :
— Calcul de $\phi_{\vec{v}_1}$ et de $\phi_{\vec{v}_2}$ par résolution de l'équation **(3)** grâce à la méthode d'Euler
— Calcul de $\partial_{\vec{v}_1}G$ et de $\partial_{\vec{v}_2}G$ avec **(2)**
— Calcul de la nouvelle vitesse $\vec{v}_{nouvelle}$ par la méthode du gradient : $\vec{v}_{nouvelle}=\vec{v}_{precedente}-\beta.\vec{\nabla}G(\vec{v}_{precedente})$
— Calcul de l'écart obtenu avec la nouvelle vitesse initiale $\vec{v}$ par résolution de l'équation **(1)** grâce à la fonction *scipy.odeint* et incrémentation du compteur

3.4.3 Introduction du hasard

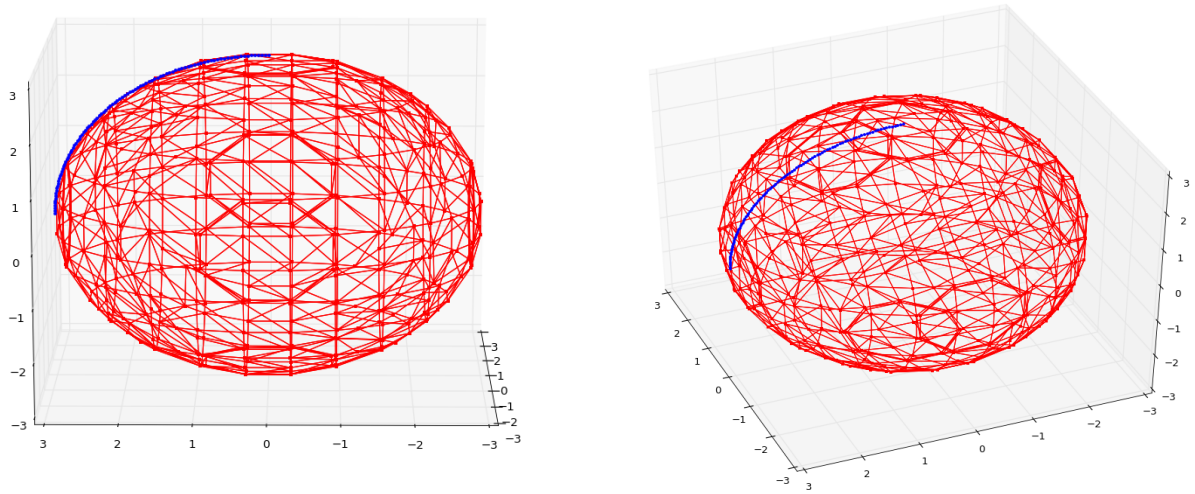
Si le compteur est un multiple de 100 et l'écart toujours trop élevé, on relance le procédé avec une vitesse obtenue par combinaison linéaire aléatoire de $\vec{v}_1$ et de $\vec{v}_2$.

4 Résultats et exploitation

4.1 Recherche sur la sphère

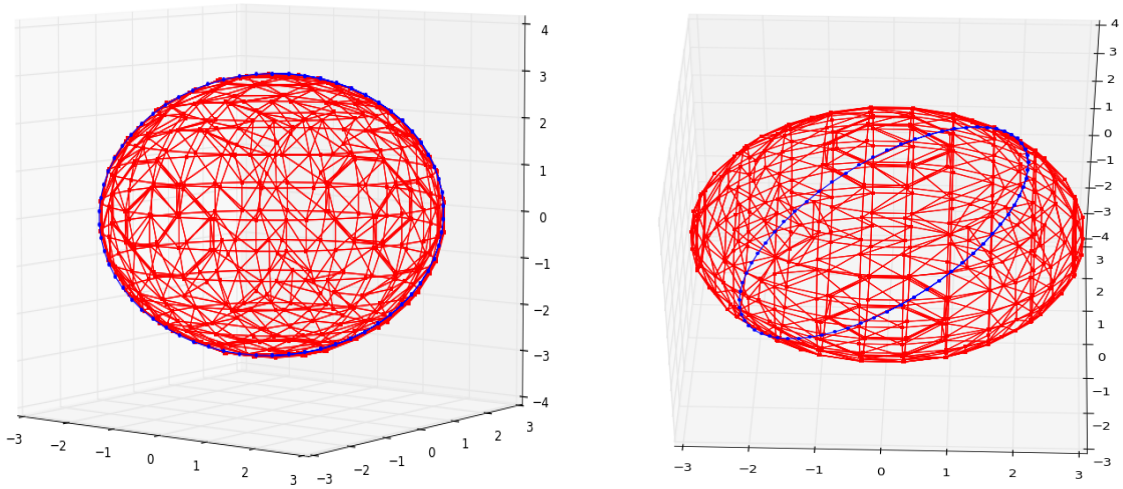
Exemple 1 :
Arguments :
$F : (x, y, z) \mapsto (x^2 + y^2 + z^2) - 9 \quad a = (0, 3, 0) \quad b = (0, 0, -3) \quad N = 100 \quad \epsilon = 10^{-4}$

Résultats :
Nombre de chemins calculés : *compteur* $= 9$
Durée : 1.87 secondes



Exemple 2 :
Arguments :
$F : (x, y, z) \mapsto (x^2 + y^2 + z^2) - 9 \quad a = (0, 0, 3) \quad b = (0, 0, -3) \quad N = 100 \quad \epsilon = 10^{-4}$

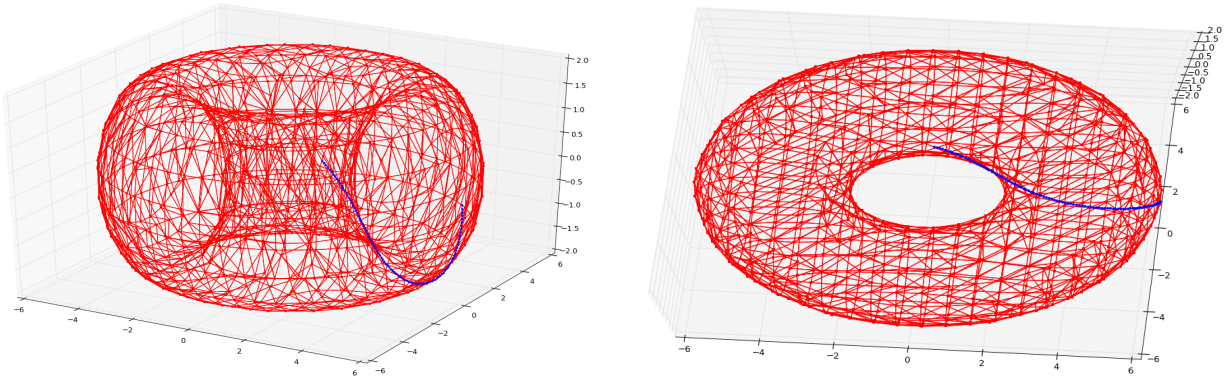
Résultats :
Nombre de chemins calculés : *compteur* $= 38$
Durée : 2.29 secondes



4.2 Recherche sur le tore

Exemple 3 :
Arguments :
$F : (x, y, z) \mapsto (x^2 + y^2 + z^2 + 12)^2 - 64 \times (x^2 + y^2)$ $\quad a = (6, 0, 0)$ $\quad b = (0, 2, 0)$ $\quad N = 100$ $\quad \epsilon = 10^{-3}$

Résultats :
Nombre de chemins calculés : $compteur = 69$
Durée : 9.1 secondes



4.3 Analyse des résultats

L'**exemple 1** est satisfaisant puisque le chemin obtenu correspond au quart d'arc réalisant la plus petite longueur sur la sphère. Même si la minimalité est difficilement démontrable sur le tore, le chemin de l'**exemple 3** est également de petite longueur.

Cependant, dans l'**exemple 2**, le chemin obtenu n'est évidemment pas de longueur minimale puisqu'il réalise plusieurs tours de la sphère au lieu du demi-arc reliant les pôles.

4.4 Temps de réponse et complexité

Rapidité : Pour $N = 100$, le temps de résolution et d'affichage ne dépasse pas 30 secondes.

Complexité : En prenant la multiplication pour opération élémentaire, la complexité de l'algorithme est en $O(compteur \times N)$.

4.5 Limites

— Pour obtenir une base du plan tangent (*Figure 3.1*), $\vec{\nabla} F$ ne doit pas s'annuler en a ni prendre de trop petites valeurs.

— Par ailleurs, le chemin obtenu semble dépendre de la première vitesse choisie à l'initialisation donnant parfois naissance à des chemins trop longs.

4.6 Possibilité d'amélioration

Afin d'utiliser la dépendance en la première vitesse, il serait intéressant de lancer plusieurs fois l'algorithme pour des premières vitesses choisies aléatoirement puis de conserver le plus court chemin obtenu.

5 Conclusion

Cet algorithme présente une approche analytique de la problématique posée. En effet, l'équation différentielle vérifiée par les géodésiques m'a permis de mettre en place un algorithme de minimisation principalement basé sur la résolution d'équations différentielles. Après amélioration, il pourrait permettre de trouver un chemin de longueur minimale entre deux points d'un large éventail de surfaces. Néanmoins, une méthode discrète ou l'établissement d'une carte de distance pourraient permettre de contrôler la qualité du chemin obtenu.

966 mots

6 Bibliographie additionnelle

[1] *https ://fr.wikipedia.org/wiki/Algorithme_ du_ gradient.* Dernière modification en mars 2017. Consulté en mars 2017.