Modélisation/Calculs scientifiques - GME5

TD 1

Algorithmique illustrée

G. Vinsard

Gerard.Vinsard@univ-lorraine.fr

27 février 2014

Objectifs

- Apprendre à utiliser Maxima

- Apprendre à écrire l'algorithme d'un problème non-trivial soit le calcul de points équidistants sur une courbe 2D paramétrée a priori quelconque.

Prise en main de (wx)Maxima

- Charger le fichier « instructions_de_base.wxm » sous wxmaxima ;

- Exécuter une par une les instructions et vérifier que les résultats d'exécution correspondent bien aux explications ;

- Optionnellement, si ce travail est fini avant que le reste du groupe l'ait accompli, faire les exercices de la fin du fichier.

Le problème et le début de son analyse

► Les coordonnées de la courbe paramétrée sont $(x(t), y(t))$ pour $t \in [0, T]$. Les points équidistants le long de la courbe seront, pour $N+1$ points

$$\left\{ \begin{array}{ll} (x(t_n^N), y(t_n^N)) \text{ où} & t_0^N = 0, t_N^N = T \\ & \forall\ n = 0, \ldots N-2\ :\ d(t_n^N, t_{n+1}^N) = d(t_{n+1}^N, t_{n+2}^N) \end{array} \right\}$$

si

$$d(t_n, t_p) = \sqrt{(x(t_n) - x(t_p))^2 + (y(t_n) - y(t_p))^2}$$

► La question est de trouver les t_n^N pour $n = 1, \ldots, N-1$. Une façon de faire est de procéder de façon itérative, c'est à dire que si les t_n^N sont connus pour un N donné, les t_n^{2N} seront tels que

$$t_{2p}^{2N} = t_p^N\ ;\ t_{2p+1}^{2N} \text{ déterminé par } d(t_p^N, t_{2p+1}^{2N}) = d(t_{2p+1}^{2N}, t_{p+1}^N)$$

Cette façon de faire présente un petit défaut (non rédhibitoire) qui apparaîtra par la suite

Tâches premières

- Écrire l'algorithme permettant de trouver $t_{n+\frac{1}{2}}$ tel que

$$d(t_n, t_{n+\frac{1}{2}}) = d(t_{n+\frac{1}{2}}, t_{n+1})$$

lorsque t_n et t_{n+1} sont connus et dans l'hypothèse que la solution est unique.

- Le tester manuellement sur les cas des

Demi cercle : $\begin{cases} x(t) = \cos t \\ y(t) = \sin t \end{cases} \quad 0 < t < \pi$; parabole : $\begin{cases} x(t) = t \\ y(t) = t^2/2 \end{cases} \quad 0 < t < \pi$

- Le coder et vérifier qu'il donne les résultats attendus sur les exemples du demi cercle et de la parabole.

Tâches finales

▶ Écrire l'algorithme général permettant de faire le passage des t_n^N aux t_n^{2N} s'appuyant sur l'algorithme précédent.

▶ Le coder et vérifier qu'il donne les résultats attendus sur des exemples du demi cercle et de la parabole.

▶ Le tester avec

$$\begin{cases} x(t) = \sin t \\ \\ y(t) = t^2/2 \end{cases} \qquad 0 < t < \pi$$

On voit que les points ne sont pas équidistants. Pourquoi ?

▶ Il est possible d'améliorer l'algorithme par exemple en remplaçant les t_{2p}^{2N} par les valeurs t_{2p}^{2N*} telles que

$$d(t_{2p-1}^{2N*}, t_{2p}^{2N}) = d(t_{2p}^{2N*}, t_{2p+1}^{2N})$$

juste après avoir calculé les t_{2p+1}^{2N} et éventuellement en réitérant le processus. Mais ce n'est pas demandé.