TDsm (1) — Courbes de Bézier - Design

1. login sur **ensibm**
2. récupérer le fichier *BEZIER.tgz* qui contient un programme Matlab
   - en le copiant avec *cp /home/perms/hahmann/BEZIER.tgz*
   - ou en le téléchargeant à l'URL[1]
   - ou en allant sur le Kiosk ENSIMAG, étagère du cours Modélisation Géométrique.
3. *gunzip BEZIER.tgz*
3. *tar -xvf BEZIER.tar* ainsi un dossier BEZIER est créé.
5. *cd BEZIER*
6. démarrer *matlab*
7. Exécuter la commande *Courbe_de_Bezier*. C'est un programme graphique qui, à partir d'un ensemble de $n+1$ points de contrôle $\boldsymbol{b}_i \in \mathbb{R}^2$, $i = 0, \ldots, n$ calcule et dessine une courbe de Bézier de degré n

$$x(t) = \sum_{i=0}^{n} \boldsymbol{b}_i B_i^n(t) \ , \quad t \in [0,1].$$

8. *Choix des points de contrôle*
   L'utilisateur entre les points avec la souris (bouton gauche) dans une fenêtre graphique du programme (le bouton ENTER signale la fin de la saisi).

Votre travail:
Dans la version actuelle du programme l'évaluation d'un point sur la courbe se fait en multipliant les points de contrôle $\boldsymbol{b}_i$ avec les polynômes de Bernstein $B_i^n(t)$.
- Remplacer l'évaluation par l'algorithme de De Casteljau.

en option:
- Interpolation: sélectionner $n+1$ points avec la souris sur l'écran d'ordinateur, puis interpoler par une courbe de Bézier de degré n, voir aussi TDsm (2).
- Interaction: Modification de la courbe en déplacant un point de contrôle. Faire du programme un outil interactif pour la modélisation de courbes. Utiliser Drag & Drop.

Moyens informatiques aux choix:
- Le même programme en Scilab (*TP_BEZIER_Scilab.tgz)* est disponible.
- MATLAB (petite doc est disponible[1]) ou sur le site de Mathworks[2]
- programme en C/C++ et fenêtre X11 (un programme d'exemple)[3]
- programme Matlab illustrant la fonction Drag & Drop[1]

[1] `http://ljk.imag.fr/membres/Stefanie.Hahmann/ENSIMAG/COURS/mod_geom.html`
[2] `http://www.mathworks.fr/help/techdoc/`
[3] `http://ljk.imag.fr/membres/Stefanie.Hahmann/ENSIMAG/students.html`

ENSIMAG 2ème année **Stefanie HAHMANN**
Modélisation Géométrique **2011/12**

TDsm (2) — Courbes de Bézier - Interpolation

Interpolation par courbe de Bézier

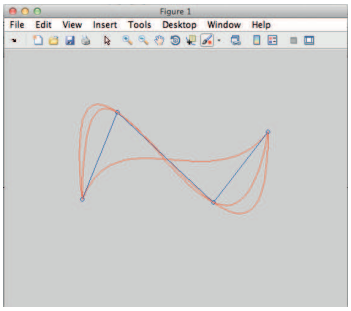
Soient $n+1$ points $\boldsymbol{P}_0, \ldots, \boldsymbol{P}_n \in \mathbb{R}^2$ et $n+1$ paramètres correspondants $t_0, \ldots, t_n$. On cherche la courbe de Bézier

$$x(t) = \sum_{i=0}^{n} \boldsymbol{b}_i B_i^n(t) \ , \quad t \in [a, b]$$

de degré n qui interpole ces points.

Votre travail

1. Formulez mathematiquement la solution du problème.
2. Utilisez le programme MATLAB $Courbe\_de\_Bezier$ du TDsm(1) pour programmer la solution. Ajoutez deux items au menu principal qui correspondent à deux solutions différentes, i.e. à deux choix de parametrisation différents. On appelera les deux nouveaux boutons dans le menu "Interpol. equidistante"et "Interpol. chordale".
3. On utilisera le bouton "NEW" dans le menu pour entrer à la souris les points $\boldsymbol{P}_i$. Les deux nouveaux boutons servent ensuite à exécuter l'interpolation equidistante et chordale sur le $même$ ensemble de points $\boldsymbol{P}_i$. On doit pouvoir afficher les courbes interpolantes en même temps, si possible avec des couleurs différentes.
4. Que remarquez vous en comparant les deux solutions ?



Choix de la paramétrisation

Une paramétrisation est associée aux points d'entrée. Ce sont les valeurs scalaires t_i associées aux points $\boldsymbol{P}_i$. Il y a deux facons courantes de les choisir.

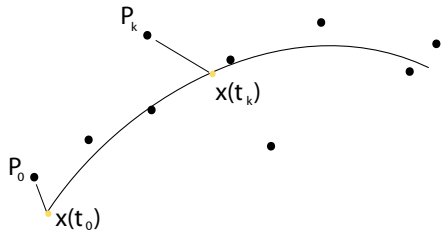
- **Paramétrisation équidistante**
  Les t_i sont réparties de façon equidistante dans l'intervalle des paramètres $[a, b]$ de la courbe $\boldsymbol{x}(t)$.
  $$t_i = a + \frac{i}{n+1}(b-a), \qquad i = 0, \ldots, n+1$$
- **Paramétrisation chordale**
  Les t_i sont réparties proportionels à la distance entre les points $\boldsymbol{P}_i$.

ENSIMAG 2ème année **Stefanie HAHMANN**
Modélisation Géométrique **2011/12**

TDsm (3) — Courbes de Bézier - Approximation

Projet noté: Approximation d'un nuage de points par une courbe de Bézier
Soient $N+1$ points $\boldsymbol{P}_0, \ldots, \boldsymbol{P}_N \in \mathbb{R}^2$ et $N+1$ paramètres correspondants $t_0 \leq t_1 \leq \cdots \leq t_N$, $t_i \in [a, b]$ (à vous de faire ce choix de la paramétrisation).
On cherche la courbe de Bézier $x(t) = \sum_{i=0}^{n} \boldsymbol{b}_i B_i^n(t),\ t \in [a, b]$ de degré n qui approxime au sens des moindres carrés cet ensemble de $N+1$ points.

$$\sum_{k=0}^{N} \|\boldsymbol{P}_k - \boldsymbol{x}(t_k)\|^2 \rightarrow \min .$$



Votre travail
Le travail attendu n'est pas une simple implémentation d'une méthode. Il est attendu que vous vous posez des questions par rapport aux données d'entrée et à la qualité des résultats obtenus.

1. Formulez mathematiquement la solution du problème.
2. Utilisez p.ex. le programme MATLAB $Courbe\_de\_Bezier$ du TDsm(1) pour programmer la solution. On entre à la souris les points $\boldsymbol{P}_i$, on choisit la apramétrisation et le degré n de la courbe $\boldsymbol{x}$. Deux nouveaux boutons servent ensuite à exécuter l'approximation equidistante et chordale sur le $même$ ensemble de points $\boldsymbol{P}_i$ donné initialement.
3. En plus: Faites de votre programme un outil qui facilite $l'interaction$ et $l'analyse$ des résultats (inventez des outils p.ex.). A vous d'imaginer des fonctionalités supplémentaires de votre programme. P.ex. on doit pouvoir afficher deux courbes simultanément si elles approximent le même ensemble de points, mais possédant une paramétrisation différente. Choix des couleurs....
4. Permettez à l'utilisateur de varier la valeur de n, le degré de la courbe.

Votre rapport
Rediger un rapport contenant

- la formulation mathématique du problème et la solution,
- les résultats obtenus,
- un grand nombre d'illustrations et exemples !!
- une analyse des deux algorithmes,
- et vos observartions.

Vous pouvez vous poser un certain nombre de questions et tenter d'y réponde, comme p.ex.

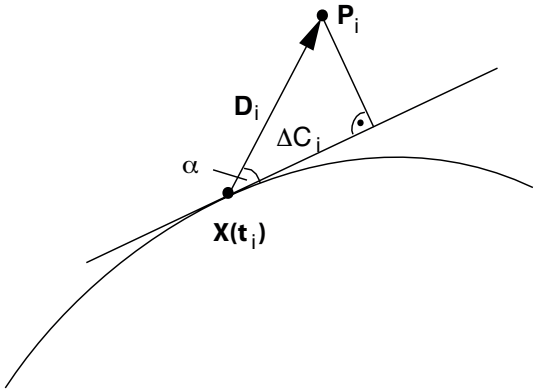
- Comment la valeur de n influence t'elle le résultat? Peut-on fixer une valeur "optimale" par defaut? si oui, laquelle?
- Peut-on comparer les solutions? Avec quels critères?
- Il y a t'il des ensembles de données pour lesquelle une ou l'autre méthode marche bien ou mieux?

Choix de la paramétrisation
voir TDsm (2).

En option
Il existe un algorithme qui permet d'améliorer la qualité de l'approximation.

Algorithme de correction des paramètres:

Soit $\boldsymbol{D}_k := \boldsymbol{P}_k - \boldsymbol{x}(t_k)$.



Input: $\boldsymbol{P}_k$, $k = 0, \ldots, N$.
Output: $\boldsymbol{x}(t)$, $t \in [a, b[$.
Procédure:

0. Choix des valeurs de paramètres t_k,
   valeur limite L (du nombre d'itérations),
   erreur ε,
   j=0.
1. Calculer μ (longueur du polygone $\{\boldsymbol{P}_k\}$)
2. Déterminer la courbe de Bézier $\boldsymbol{x}(t)$ approximant par la méthode des moindres carrés.
3. `FOR` i=0 `TO` N `DO`
4. Calculer $\Delta C_i = \frac{<\boldsymbol{D}_i, \boldsymbol{x}'(t_i)>}{\|\boldsymbol{x}'(t_i)\|}$ /* approximation du changement de paramètre nécessaire */
5. Calculer $t_i^* = t_i + \Delta C_i \frac{b-a}{\mu}$ /* nouveau paramètre*/
   Calculer $\|\boldsymbol{D}_i\|$ et $\|\boldsymbol{D}_i^*\|$
   `IF` $\|\boldsymbol{D}_i\| > \|\boldsymbol{D}_i^*\|$ ($t_i = t_i^*$ AND GOTO 6)
   `ELSE` $\Delta C_i = \Delta C_i/2$, et `goto` 5.
6. $\alpha_i = arccos(\frac{<\boldsymbol{D}_i^*, \boldsymbol{x}'(t_i^*)>}{\|\boldsymbol{D}_i^*\|\|\boldsymbol{x}'(t_i^*)\|})$
   `END FOR`
7. Calculer $\alpha = \max(\alpha_i)$
   j = j+1
   `IF` $|\pi/2 - \alpha| < \varepsilon$
   `THEN` calculer la courbe de Bézier approximant pour les derniers paramètres t_i et stop
   `ELSE IF` $j < L$ `GOTO` 2.

Attention ! vérfiez toujours $t_i^* \in [a, b]$ et $t_i \leq t_{i+1}$.