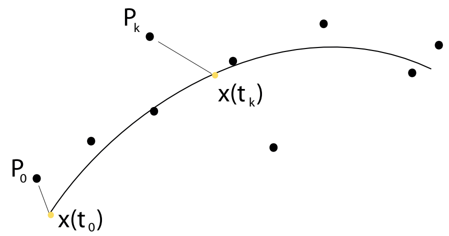
ENSIMAG 2ème année — **Stefanie HAHMANN**
Modélisation Géométrique — **2011/12**

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.

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



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 aprامétrisation 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.

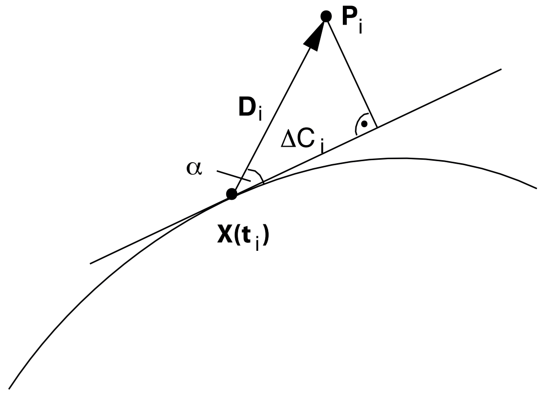
- Comment la valeur de n influence t'elle le résultat? Peut-on fixer une valeur "optimale" par defaut? si oui, laquelle?
- Peut-on comparer les solutions? Avec quels critères?
- Il y a t'il des ensembles de données pour lesquelle une ou l'autre méthode marche bien ou mieux?

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 $t_i \leq t_{i+1}$.