Polynômes

Calcul formel – TP 8

Toutes les questions peuvent être traitées avec l'aide de `Sage`*,*
sauf lorsque l'on vous demande de faire une preuve « à la main » !

1. Racines évidentes d'un polynôme

Soit $P(X) = a_n X^n + a_{n-1} X^{n-1} + \cdots + a_1 X + a_0$ un polynôme à coefficients entiers. Soit $\alpha = \frac{p}{q}$ une fraction irréductible avec p et q entiers relatifs. En utilisant le théorème suivant :

« Si α est racine de P, alors p divise a_0 et q divise a_n. »

on obtient une méthode pour trouver toutes les racines rationnelles d'un polynôme à coefficients entiers (pas trop gros).

1. Écrire une fonction qui prend en argument un polynôme P et renvoie la liste des rationnels $\frac{p}{q}$ susceptibles d'être racine de P.
2. Améliorer votre fonction pour qu'elle renvoie la liste exacte des racines rationnelles de P.
3. Tester vos fonctions avec $P(X) = 45X^5 + 71X^4 - 126X^3 - 138X^2 + 72X - 8$.
4. **Énigme.** Trouver la plus petite valeur $k \in \mathbb{N}$ telle que $336X^5 + 208X^4 + kX^3 - 147X^2 - 19X + 10$ admette au moins 3 racines rationnelles distinctes.

2. Polynômes de Tchebychev

Les polynômes de Tchebychev sont définis par récurrence :

$$T_0(X) = 1, \quad T_1(X) = X \quad \text{et} \quad T_n(X) = 2X\,T_{n-1}(X) - T_{n-2}(X) \quad \text{pour } n \geqslant 2.$$

1. (a) Écrire une fonction qui, pour n fixé, renvoie la liste des polynômes de Tchebychev $(T_0, T_1, \dots, T_n)$.
   (b) (À la main.) Quel est le degré de T_n ? Quel est son coefficient dominant ?
   (c) Tracer quelques graphes de $\frac{1}{2^{n-1}} T_n(X)$ et visualiser à quel point ces graphes approchent la fonction nulle sur l'intervalle $[-1, +1]$.
2. (a) Montrer formellement que pour tout $n \in \mathbb{N}$ et pour tout $\theta \in \mathbb{R}$:

   $$\cos(n\theta) = 2\cos(\theta)\cos((n-1)\theta) - \cos((n-2)\theta).$$

   Indication : Utiliser la méthode `reduce_trig()`.
   (b) En déduire à la main la relation

   $$T_n\big(\cos(\theta)\big) = \cos(n\theta).$$

(c) En déduire à la main que les racines de T_n sont les

$$\cos\left(\frac{(2k-1)\pi}{2n}\right) \qquad k = 1, 2, \dots, n.$$

3. Nous allons exprimer X^n dans la base $(T_0, T_1, \dots, T_n)$. Par exemple : $X^2 = \frac{1}{2}T_2(X) + \frac{1}{2}T_0(X)$.
   Première méthode. Effectuer une succession de divisions euclidiennes par les T_k en commençant par X^n divisé par T_n.
   Seconde méthode. Écrire la matrice $A = (a_{ij})_{0 \leqslant i,j \leqslant n}$ où la j-ème colonne de A est formée des coefficients de T_j dans la base $(1, X, X^2, \dots, X^n)$. Alors la j-ème colonne de A^{-1} est constituée des coefficients de X^j dans la base $(T_0, T_1, T_2, \dots, T_n)$.
   Énigme. Dans l'expression de X^{30} dans cette base, quel est le coefficient devant T_0 ? Il s'exprime sous la forme $\frac{p}{2^{26}}$, donner p.
4. Mettre en place l'interpolation polynomiale de Lagrange de la fonction définie par $f(x) = \frac{1}{1+8x^2}$ sur l'intervalle $[-1, +1]$ pour les $n+1$ racines de $T_{n+1}(X)$. Visualiser que le phénomène de Runge a disparu. (*Indication.* Faire des calculs numériques approchés.)

3. Algorithme d'Estrin

Il s'agit d'un travail au cours duquel il faut mettre en œuvre un algorithme récursif. Si vous n'avez jamais étudié d'algorithmes récursifs, il vaut mieux en découvrir quelques uns. Par exemple, lire la partie 5 de « Algorithmes et mathématiques ».

L'algorithme d'Estrin est une alternative à l'algorithme de Horner pour évaluer un polynôme P en une valeur α. Soit $P(X) = a_nX^n + a_{n-1}X^{n-1} + \cdots + a_kX^k + \cdots + a_1X + a_0 \in K[X]$. Fixons $\alpha \in K$.
L'idée est de séparer le polynôme en deux parties :

$$P(\alpha) = \big(\underbrace{a_n\alpha^{n-m} + a_{n-1}\alpha^{n-m-1} + \cdots + a_m}_{P_h(\alpha)}\big) \cdot \alpha^m + \big(\underbrace{a_{m-1}\alpha^{m-1} + a_{m-2}\alpha^{m-2} + \cdots + a_0}_{P_b(\alpha)}\big)$$

puis on recommence, en séparant en deux la partie haute $P_h(\alpha)$, puis en deux la partie basse $P_b(\alpha)$. On continue ainsi jusqu'à obtenir des polynômes de degré 1 que l'on évalue.
L'entier m choisi s'obtient par $m = 2^p$ où $p = \lceil \log_2(n) \rceil - 1$ avec $\lceil x \rceil$ la partie entière supérieure de x (qui s'obtient par `ceil(x)`).

1. (a) Écrire une fonction qui, pour α et m fixés, renvoie la liste préalable des α^k où les k sont des puissances de 2 : $[\alpha^2, \alpha^4, \alpha^8, \dots, \alpha^m]$.
   (b) Combien de multiplications faut-il pour calculer tous ces éléments ? (Indication : c'est beaucoup moins que ce que vous pensez !).
2. Implémenter l'algorithme récursif :

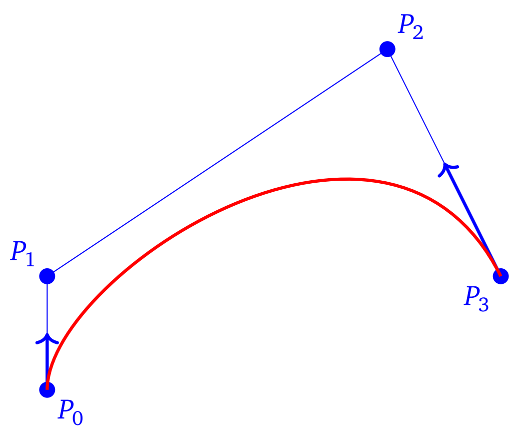
```
Entrées : P, alpha
Sortie : P(alpha)

fonction eval_estrin(P,alpha)
    n ← degre(P)
    si n ⩽ 1
        renvoyer a_0 + a_1 * alpha
    sinon
        calculer m
        extraire Pbas et Phaut de P
        renvoyer eval_estrin(Phaut,alpha) * alpha^m + eval_estrin(Pbas,alpha)
```

3. **Énigme.** Compter le nombre total de multiplications nécessaires pour calculer $P_N(\alpha)$, avec $N = 50$, lorsque
$$P_N(X) = \sum_{k=0}^{N}(k+1)X^{3k} \qquad \text{et} \qquad \alpha = 2.$$
Indications :
   - Il s'agit de compter le nombre de multiplications à faire dans les questions 1 et 2. Donc le nombre de fois où l'on effectue l'opération `*`.
   - Ne comptez pas les multiplications `0 * alpha` ou `0 * alpha^m` (où l'un des termes est nul).
   - Exemple. Pour $N = 10$, il faudrait un total 25 multiplications (les 4 multiplications préalables de la question 1 et 21 multiplications pour la question 2).
4. Comparer avec le nombre de multiplications nécessaires par l'algorithme d'Horner.

4. Courbes de Bézier

1. **Polynômes de Bernstein.**
Définir une fonction `bernstein` de paramètres n et k entiers avec $0 \leqslant k \leqslant n$ qui renvoie le polynôme de Bernstein
$$B_k^n(X) = \binom{n}{k}X^k(1-X)^{n-k}.$$
Rappel : $\binom{n}{k}$ est le coefficient du binôme de Newton.
Facultatif et à la main :
   (a) Montrer que pour $0 < x < 1$ et $0 \leqslant k \leqslant n$, on a l'inégalité $B_k^n(x) > 0$.
   (b) En écrivant $1 = X + (1-X)$ vérifier que : $\sum_{k=0}^{n} B_k^n(X) = 1$.
   (c) Montrer la formule de récurrence : $B_k^n = XB_{k-1}^{n-1} + (1-X)B_k^{n-1}$, $(0 < k < n)$.
2. **Courbes de Bézier.**
Étant donnée une liste de ***points de contrôle*** $P_0, P_1, \dots, P_n$ du plan, tracer la ***courbe de Bézier*** associée à cette liste qui est la courbe paramétrée donnée par :
$$\sum_{k=0}^{n} B_k^n(t)P_k \qquad t \in [0,1].$$
On additionne les points comme des vecteurs : $\lambda P + \mu Q$ est le point $\begin{pmatrix}\lambda x_P + \mu x_Q \\ \lambda y_P + \mu y_Q\end{pmatrix}$.
Le cas de quatre points de contrôle est particulièrement intéressant : la courbe passe par les points extrémités P_0 et P_3 (mais pas par les autres). La tangente en P_0 est dirigée par $\overrightarrow{P_0P_1}$ et celle en P_3 par $\overrightarrow{P_3P_2}$.



3. **Énigme.**

   Quel nombre à trois chiffres se cache derrière ces trois listes de points de contrôle :

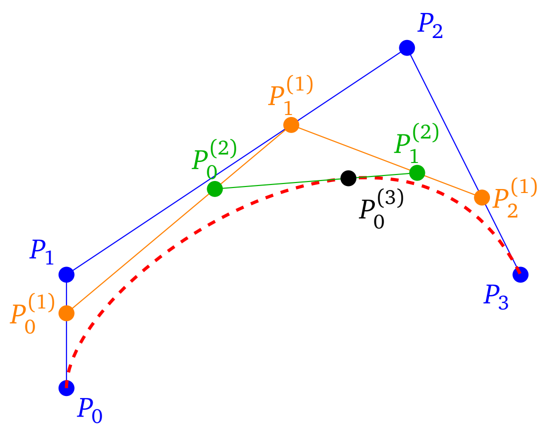
   - $[(-1,1),(-3,1),(-2,3),(-1,3),(0,0),(-2,0)]$
   - $[(1,0),(0,0),(0,3),(2,3),(2,0),(1,0)]$
   - $[(2,0),(4,0),(4,1),(1,1),(1,1),(4,1),(4,2),(2,2)]$

4. **Algorithme de Casteljau.**

   Il existe une autre méthode pour tracer des points de la courbe de Bézier.

   Soient $P_0, P_1, \ldots, P_n$ les $n+1$ points de contrôle. Soit $t \in [0,1]$. On construit d'abord n points : $P_0^{(1)}, P_1^{(1)}, \ldots, P_{n-1}^{(1)}$ où $P_k^{(1)} = (1-t)P_k + tP_{k+1}$ est un point du segment $[P_k, P_{k+1}]$. Puis on recommence. À partir de $n-i+1$ points, $P_0^{(i)}, P_1^{(i)}, \ldots, P_{n-i}^{(i)}$, on construit $n-i$ points, $P_0^{(i+1)}, P_1^{(i+1)}, \ldots, P_{n-i-1}^{(i+1)}$. À la fin, il ne reste plus qu'un point $P_0^{(n)}$ qui est un point de la courbe de Bézier.

   Voici un exemple avec 4 points de contrôle et avec $t = \frac{2}{3}$.



   Mettre en œuvre cet algorithme pour tracer la courbe et le tester avec la liste de points de contrôle : $(0,0),(0,1),(3,3),(4,1)$.