

III. Interpolation

En mathématiques, l'interpolation polynomiale permet, à partir d'un ensemble de points donné, de construire une fonction polynomiale passant par ces points.

1 Interpolation de Lagrange

On se donne n points $(x_0, y_0), \dots, (x_{n-1}, y_{n-1})$, avec les x_i deux à deux distincts. On se propose de construire un polynôme P de degré minimal passant par les points (x_i, y_i) , c'est à dire, telle que $P(x_i) = y_i$ pour tout $i = 0, \dots, n$.

Exercice 1.1. Soient $(x_0, y_0), \dots, (x_n, y_n)$ des points d'abscisses distinctes.

1. Construire un polynôme L_i tel que

$$L_i(x_j) = \begin{cases} 0 & \text{pour } j \neq i \\ 1 & \text{pour } j = i \end{cases}$$

2. En déduire un polynome P tel que $P(x_i) = y_i$.

Solution :

1. Posons

$$L_i(X) = \prod_{k=0, k \neq i}^{n-1} \frac{X - x_k}{x_i - x_k}$$

On a alors

$$L_i(x_i) = \prod_{k=0, k \neq i}^{n-1} \frac{x_i - x_k}{x_i - x_k} = \prod_{k=0, k \neq i}^{n-1} 1 = 1$$

ainsi que pour $j \neq i$:

$$\begin{aligned} L_i(x_j) &= \prod_{k=0, k \neq i}^{n-1} \frac{x_j - x_k}{x_i - x_k} \\ &= \left(\prod_{k=0, k \neq i, k < j}^{n-1} \frac{x_j - x_k}{x_i - x_k} \right) \times \frac{x_j - x_j}{x_i - x_j} \times \left(\prod_{k=0, k \neq i, k > j}^{n-1} \frac{x_j - x_k}{x_i - x_k} \right) \\ &= \prod_{k=0, k \neq i, k < j}^{n-1} \frac{x_j - x_k}{x_i - x_k} \times 0 \times \prod_{k=0, k \neq i, k > j}^{n-1} \frac{x_j - x_k}{x_i - x_k} \\ &= 0 \end{aligned}$$

Exercice 1.2. Trouver puis tracer le polynome de Lagrange passant par les points

1. $(1, 1), (2, 3), (3, 4)$

2. $(-1, 1), (0, 4), (1, 3), (2, 16)$

2 Splines

Dans bon nombre d'applications, il est impératif d'obtenir des courbes très lisses et régulières passant par un grand nombre de points. C'est le cas en conception assistée par ordinateur (CAO), où l'on cherche à représenter des objets aux formes lisses. Nous avons déjà constaté que l'utilisation de polynômes de degré élevé est délicate et mène parfois à des oscillations de grande amplitude : c'est le phénomène de Runge, illustrée à la figure 3.1. Les polynômes de degré élevé sont alors peu adéquats.

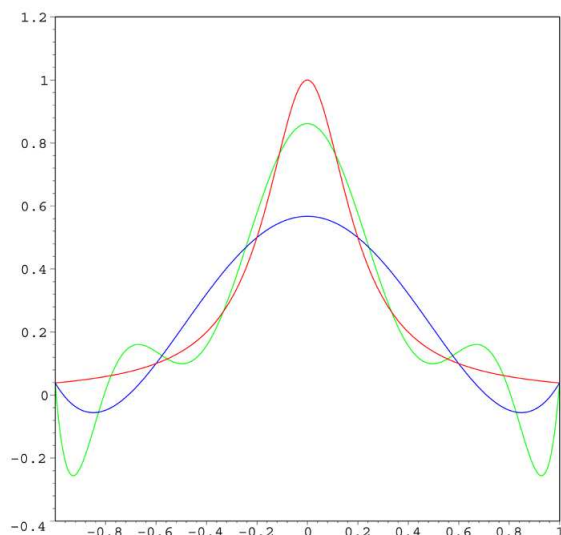


FIGURE 3.1 : En rouge, la courbe à interpoler, en bleu un polynôme interpolateur de degré 5 et en vert un de degré 9. L'approximation est de plus en plus mauvaise.

Le problème, lorsque l'on utilise des polynômes de faible degré, provient du fait qu'il faut en utiliser plusieurs pour relier tous les points.

Définition 2.1. En mathématique, une spline est une fonction définie par morceaux par des polynômes.

On peut mesurer la régularité d'une fonction par le biais de ses dérivées. En effet, plus une fonction est différentiable, plus la courbe qui lui est associée est lisse et plus la fonction est régulière.

Définition 2.2. Soit I un intervalle de \mathbb{R} . Une courbe $\mathcal{C} : I \rightarrow \mathbb{R}$ est dite C^n sur I pour $n \in \mathbb{N}$ si elle est n fois dérivable et si sa dérivée n ème est continue. On note $C^n(I)$ l'ensemble des courbes $I \rightarrow \mathbb{R}$ qui sont C^n sur I .

Exercice 2.3.

- Quels sont les courbes C^0 ?
- Montre que la fonction $x \mapsto |x|$ est C^0 mais pas C^1 sur $[-1, 1]$.
- Pour quel valeur de n , un polynôme de degré k est il C^n sur \mathbb{R} .

Définition 2.4. Soient $I = [a, b]$ un intervalle de \mathbb{R} et $a = x_0 < x_1 < \dots < x_{n-1} = b$ des éléments de I . La courbe $S : I \rightarrow \mathbb{R}$ est appelée spline de degré k si $S \in C^{k-1}(I)$ et si la

restriction de S à chaque intervalle $[x_i, x_{i+1}]$ est un polynôme de degré k pour $i = 0, \dots, n - 2$. Les points $(x_i, S(x_i))$ sont appelés points de contrôle.

Pour définir une spline S de degré k il nous faut n points et $n - 1$ polynômes q_i de degré k . La spline est alors définie de la manière suivante :

$$S(x) = q_i(x) = \sum_{j=0}^k a_{i,j} (x - x_i)^j \quad \text{pour } x \in [x_i, x_{i+1}] \text{ avec } i \in \{0, \dots, n - 2\}$$

La translation par $-x_i$ dans l'expression de q_i permet de simplifier les calculs lors de la construction de la spline et ne compléxifie pas vraiment sa définition.

Exercice 2.5. Créer une structure `Spline` permettant de représenter une spline en C++.

Solution. Il faut stocker le nombre de points, les abscisses des points et les polynômes. Le polynôme

$$q_i(x) = \sum_{j=0}^k a_{i,j} (x - x_i)^j$$

sera représenté par le polynôme

$$p_i(x) = \sum_{j=0}^k a_{i,j} x^j.$$

On vérifie immédiatement la relation $q_i(x) = p_i(x - x_i)$. Ce qui donne la structure suivante :

```
struct Spline{
    usint nb_points;
    Rationnel* x;
    Polynome* polynomes;
};
```

Exercice 2.6. Ecrire une fonction

```
void draw_spline(Spline S, Fenetre F, Color c)
```

permettant de tracer une spline S de couleur c vue à travers la fenêtre S .

3 Spline linéaire

La fonction spline la plus simple est de degré 1 : la spline linéaire.

Exercice 3.1. Soient $I = [a, b]$ un intervalle et $(x_0, y_0), \dots, (x_n, y_n)$ des points de \mathbb{R}^2 avec $a = x_0 < x_1 < \dots < x_n = b$. Montrer que la courbe obtenu en reliant successivement les points (x_i, y_i) est une spline linéaire.

Exercice 3.2. Calculer la spline linéaire d'interpolation des points $P_0 = (0, 1), P_1 = (1, 3), P_2 = (2, 2), P_3 = (3, 3)$.

Exercice 3.3. Ecrire en pseudo langage, puis en C++ une fonction

```
void spline_lineaire(Point* P, usint n)
```

qui trace la spline linéaire passant par n points $P[0], \dots, P[n - 1]$ de la fenêtre SDL.

La courbe obtenue par spline linéaire n'est pas vraiment régulière, il faut donc augmenter le degré du polynômes.

4 Splines quadratiques

L'interpolation quadratique est l'une des méthodes les plus simples. Le but est de relier les points par des courbes de degrés deux. Mais si nous nous contentons de faire passer une parabole par les trois premiers points puis une autre par les trois suivants, la courbe comportera des cassures. En effet, à la jonction de deux paraboles, la pente de l'une n'est pas la même que celle de l'autre. Ce procédé ne présente donc aucun avantage sur l'interpolation linéaire.

Pour supprimer ces cassures et obtenir une courbe plus régulière, nous modifions les conditions :

- la courbe doit passer par deux points imposés
- en chaque point intérieur, la pente de la parabole à droite doit être égale à la pente de la parabole à gauche.

Un problème demeure : il n'y a pas de condition sur la pente de la première parabole. Nous devons donc choisir la pente initiale arbitrairement.

Pour $i = 0, \dots, n - 1$, on note P_i les points de \mathbb{R}^2 de coordonnées (x_i, y_i) tels que $a = x_0 < x_1 < \dots < x_{n-1} = b$. La parabole définie sur l'intervalle $[x_i, x_{i+1}]$ est $q_i(x) = a_i(x - x_i)^2 + b_i(x - x_i) + c_i$.

Exercice 4.1. Montrer que si on a

- $q_i(x_i) = y_i$ et $q_i(x_{i+1}) = y_{i+1}$ pour $i = 0, \dots, n - 2$,
- $q'_{i-1}(x_i) = q'_i(x_i)$ pour $i = 1, \dots, n - 2$.

alors la courbe $S : [a, b] \rightarrow \mathbb{R}$ définie par

$$S(x) = q_i(x) \quad \text{pour } x \in [x_i, x_{i+1}]$$

est une spline de degré 2.

Exercice 4.2.

- Quels conditions doivent vérifier a_i, b_i et c_i pour que q_i passent par P_i et P_{i+1} ?
- Quel est la valeur de la pente de q_i en x_i ? et celle de q_{i-1} en x_i ?
- En déduire, en fonction de $x_i, x_{i+1}, y_i, y_{i+1}, a_{i-1}, b_{i-1}$ les valeurs de a_i, b_i et c_i .

Quelles sont q_0, q_1, q_2 et q_3 pour $P_0 = (0, 1), P_1 = (1, 3), P_2 = (2, 2), P_3 = (3, 3)$ et une pente initiale de 1. Et pour une pente initiale de -1 ?

Exercice 4.3. Ecrire en pseudo langage, puis en C++ une fonction

```
void spline_quadratique(Point* P, usint n, Rationnel pente)
```

qui trace la spline quadratique de pente initiale `pente` passant par n points $P[0], \dots, P[n-1]$ de la fenêtre SDL.

Exercice 4.4. Tracer la spline quadratique passant par les points $P_0 = (0, 1), P_1 = (1, 3), P_2 = (2, 2), P_3 = (3, 3)$ pour les pentes initiales : 1, 0 et -1 . Que remarquez-vous ?

Comme vous avez pu le constater, le choix de la pente initiale est très important. La courbe peut avoir une allure complètement différente selon ce choix. De plus, la pente initiale influence toute la courbe. Les courbes sont donc imprévisibles. Une parabole ne possédant pas de point d'inflexion, ceux-ci doivent absolument se trouver sur l'un des points de contrôles. Cela est aussi un problème en vue d'une utilisation pratique.

Définition 4.5. Soit $C : I \rightarrow \mathbb{R}$ une courbe de classe \mathcal{C}^2 . On dit que $x \in C$ est un point d'inflexion de C si la dérivée seconde de C s'annule en change de signe en x .

L'interpolation quadratique n'est pas très "naturelle", le tracé ne correspond pas à ce que nous pourrions attendre par exemple dans un logiciel de dessin.

Pour toutes ces raisons, l'interpolation quadratique n'est pratiquement pas utilisée. Plusieurs méthodes existent pour supprimer ces désavantages, l'une d'elle est d'augmenter encore le degré des polynômes et de demander un recollement des dérivées secondes.

5 Splines cubiques

Pour $i = 0, \dots, n$, on note toujours P_i les points de \mathbb{R}^2 de coordonnées (x_i, y_i) tels que $a = x_0 < x_1 < \dots < x_{n-1} = b$. On cherche à définir une fonction $S : [a, b] \rightarrow \mathbb{R}$ définie par morceaux à l'aide de polynômes q_i de degrés trois sur les intervalles $[x_i, x_{i+1}]$. On obtient alors une spline cubique si et seulement si

$$q_i(x_i) = y_i \quad \text{pour } i = 0, \dots, n-2 \quad (3.1)$$

$$q_i(x_{i+1}) = y_{i+1} \quad \text{pour } i = 0, \dots, n-2 \quad (3.2)$$

$$q'_i(x_i) = q'_{i-1}(x_i) \quad \text{pour } i = 1, \dots, n-2 \quad (3.3)$$

$$q''_i(x_i) = q''_{i-1}(x_i) \quad \text{pour } i = 1, \dots, n-2 \quad (3.4)$$

On pose $q_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$. On a exactement $4 \times n - 1 = 4n - 4$ inconnue et $(n - 1) + (n - 1) + (n - 2) + (n - 2) = 4n - 6$ équations. Il nous manque donc 2 équations. Ces équations pourront faire intervenir $S'(x_0), S''(x_0), S'(x_{n-1})$ et $S''(x_{n-1})$.

Remplaçons les équations précédentes en les q_i par des équations en a_i, b_i, c_i et d_i . La première donne $d_i = y_i$. On doit calculer les deux premières dérivées de q_i :

$$q'_i(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i \quad \text{et} \quad q''_i(x) = 6a_i(x - x_i) + 2b_i$$

Posons $h_i = x_{i+1} - x_i$. L'équation (3.4) donne

$$q''_i(x_i) = 2b_i = q''_{i-1}(x_i) = 6a_{i-1}(x_i - x_{i-1}) + 2b_{i-1}$$

et donc

$$b_i = 3a_{i-1}h_{i-1} + b_{i-1}. \quad (3.5)$$

L'équation (3.3) donne

$$q'_i(x_i) = c_i = q'_{i-1}(x_i) = 3a_{i-1}(x_i - x_{i-1})^2 + 2b_{i-1}(x_i - x_{i-1}) + c_{i-1}$$

et donc

$$c_i = 3a_{i-1}h_{i-1}^2 + 2b_{i-1}h_{i-1} + c_{i-1} \quad (3.6)$$

L'équation (3.2) donne

$$q_i(x_{i+1}) = a_i(x_{i+1} - x_i)^3 + b_i(x_{i+1} - x_i)^2 + c_i(x_{i+1} - x_i) + d_i = a_i h_i^3 + b_i h_i^2 + c_i h_i + y_i = y_{i+1}$$

et donc

$$a_i = \frac{y_{i+1} - y_i}{h_i^3} - \frac{c_i}{h_i^2} - \frac{b_i}{h_i}. \quad (3.7)$$

5.1 Splines simples

Si on fixe les valeurs de b_0 et c_0 , ce qui revient à fixer $S'(x_0)$ et $S''(x_0)$ alors la spline se calcule facilement.

Exercice 5.1. Ecrire une fonction

```
Spline get_spline_cubique(Point* P, usint n, float p1, float p2)
```

qui retourne la spline cubique S passant par les points $P[0], \dots, P[n-1]$ et tel que $S'(x_0) = p_1$ et $S''(x_0) = p_2$.

Solution.

```
Spline get_spline_cubique(Point* P, usint n, float p1, float p2):
    Spline S
    S.nb_points=n
    S.x=new float[n]
    S.polynomes=new Polynome[n-1]
    S.x[0]=P[0].x
    Polynome q(3)
    q[0]=P[0].y//d_0
    q[1]=p1//c_0
    q[2]=p2/2//b_0
    h=P[1].x-P[0].x//h_0
    q[3]=((P[1].y-P[0].y)-q[2]*h^2-q[1]*h)/(h^3)//a_0
    S.polynomes[0]=q
    Polynome qp=q
    S.x[1]=P[1].x
    Pour i de 1 jusque n-2 faire:
        S.x[i+1]=P[i+1].x
        q[0]=P[i].y//d_i
        hp=h//h_i-1
        h=P[i+1].x-P[i].x//h_i
        q[1]=3*qp[3]*hp*hp+2*qp[2]*hp+qp[1]//c_i
        q[2]=3*qp[3]*hp+qp[2]//b_i
        q[3]=((P[i+1].y-P[i].y)-q[2]*h*h-q[1]*h)/(h*h*h)//a_i
        S.polynomes[i]=q
        qp=q
    Retourner S
```

Une autre spline simple est celle où l'on fixe $S'(x_{n-1})$ et $S''(x_{n-2})$.

Malheureusement, ce ne sont pas ces splines qui sont utilisées.

5.2 Splines moins simple

Pour pouvoir résoudre c'est spolie il faut modifier le système d'équation. Tout d'abord on supprime un maximum d'inconnue. On pose $m_i = S''(x_i)$ pour $i = 0, \dots, n-2$.

Exercice 5.2. Calculer b_i en fonction des m_i .

Solution. On a $S''(x_i) = q_i''(x_i)$ et donc $m_i = q_i''(x_i) = 2b_i$. On a donc

$$b_i = \frac{m_i}{2}. \quad (3.8)$$

Exercice 5.3. A partir de (3.4) calculer a_i en fonction des m_i .

Solution. On doit avoir $q''_{i-1}(x_i) = m_i$. De

$$m_i = q''_{i-1}(x_i) = 6a_{i-1} + 2b_{i-1} \quad \text{et} \quad b_{i-1} = m_{i-1}$$

on obtient $a_{i-1} = \frac{m_i - m_{i-1}}{6h_{i-1}}$ et donc

$$a_i = \frac{m_{i+1} - m_i}{6h_i} \quad (3.9)$$

Exercice 5.4. A partir de (3.1) et (3.2) calculer c_i en fonction des m_i .

Solution. De $q_i(x_i) = y_i$ on obtient $d_i = y_i$. De $q_i(x_{i+1}) = y_{i+1}$, on obtient

$$y_{i+1} = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i$$

soit encore

$$c_i = \frac{1}{h_i}(y_{i+1} - y_i - a_i h_i^3 - b_i h_i^2) = \frac{y_{i+1} - y_i}{h_i} - a_i h_i^2 - b_i h_i$$

En remplaçant a_i, b_i par leurs valeurs, on obtient

$$c_i = \frac{y_{i+1} - y_i}{h_i} - h_i \frac{m_{i+1} - m_i}{6} - h_i \frac{m_i}{2} \quad (3.10)$$

A ce stade on sait calculer les coefficients a_i, b_i, c_i et d_i à partir des h_i, m_i et y_i . Les h_i et y_i étant connues il nous reste à trouver les m_i .

Exercice 5.5. A partir de (3.3) trouver une relation liant m_{i-1}, m_i et m_{i+1} .

Solution. A partir de $q'_i(x_i) = q'_{i-1}(x_i)$, on obtient

$$c_i = 3a_{i-1}h_{i-1}^2 + 2b_{i-1}h_{i-1} + c_{i-1}$$

On remplace $a_{i-1}, b_{i-1}, c_{i-1}$ et c_i par leur expression en fonction des m_i pour obtenir

$$\frac{y_{i+1} - y_i}{h_i} - h_i \frac{m_{i+1} - m_i}{6} - h_i \frac{m_i}{2} = h_{i-1} \frac{m_i - m_{i-1}}{2} + m_{i-1} h_{i-1} + \frac{y_i - y_{i-1}}{h_{i-1}} - h_{i-1} \frac{m_i - m_{i-1}}{6} - h_{i-1} \frac{m_{i-1}}{2}$$

ce qui en multipliant par 6 et isolant les termes avec du m_i du même côté donne

$$6 \left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}} \right) = h_i(m_{i+1} - m_i) + 3h_i m_i + 3h_{i-1}(m_i - m_{i-1}) + 6m_{i-1}h_{i-1} \\ - h_{i-1}(m_i - m_{i-1}) - 3h_{i-1}m_{i-1}$$

et donc

$$6 \left(\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}} \right) = h_{i-1}m_{i-1} + 2(h_{i-1} + h_i)m_i + h_i m_{i+1} \quad (3.11)$$

Pour trouver m_i avec $i = 0, \dots, n-1$ on résout un système linéaire.

Exercice 5.6. Poser ce système.

Solution. On résout

$$\underbrace{\begin{bmatrix} h_0 & 2(h_0 + h_1) & h_1 & & & \\ & h_1 & 2(h_1 + h_2) & h_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & h_{n-3} & 2(h_{n-3} + h_{n-2}) & h_{n-2} \end{bmatrix}}_A \times \underbrace{\begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{n-1} \end{bmatrix}}_M = 6 \underbrace{\begin{bmatrix} \frac{y_2 - y_1}{h_1} - \frac{y_1 - y_0}{h_0} \\ \frac{y_3 - y_2}{h_2} - \frac{y_2 - y_1}{h_1} \\ \vdots \\ \frac{y_n - y_{n-1}}{h_{n-1}} - \frac{y_{n-1} - y_{n-2}}{h_{n-2}} \end{bmatrix}}_B$$

La matrice à $n - 2$ lignes et n colonnes. Il faut donc rajouter deux lignes.

Avant de voir comment calculer les m_i , regardons comment, une fois les m_i connus calculer la spline.

Exercice 5.7. Ecrire une fonction

```
Spline get_spline_cubique(Point* P, float* M, usint n)
```

qui retourne la spline cubique d'interpolation de $P[0], \dots, P[n-1]$ à partir des coefficients $M[0], \dots, M[n-1]$.

5.3 Splines naturelles

Une spline cubique S définie sur $[a, b]$ est dite *naturelle* si $S''(a) = S''(b) = 0$. Cette condition impose $m_0 = m_{n-1} = 0$. Le système devient donc

$$\underbrace{\begin{bmatrix} 1 & & & & & \\ h_0 & 2(h_0 + h_1) & h_1 & & & \\ & h_1 & 2(h_1 + h_2) & h_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & h_{n-3} & 2(h_{n-3} + h_{n-2}) & h_{n-2} \\ & & & & & 1 \end{bmatrix}}_{A_N} \times \underbrace{\begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{n-1} \end{bmatrix}}_M = 6 \underbrace{\begin{bmatrix} 0 \\ \frac{y_2 - y_1}{h_1} - \frac{y_1 - y_0}{h_0} \\ \frac{y_3 - y_2}{h_2} - \frac{y_2 - y_1}{h_1} \\ \vdots \\ \frac{y_n - y_{n-1}}{h_{n-1}} - \frac{y_{n-1} - y_{n-2}}{h_{n-2}} \\ 0 \end{bmatrix}}_{B_N}$$

La matrice A n'est pas triangulaire supérieur mais tridiagonale, on ne peut donc pas résoudre immédiatement le système.

Exercice 5.8. Ecrire une fonction

```
Spline get_spline_cubique_naturelle(Point* P, usint n)
```

retournant la spline cubique naturelle d'interpolation de $P[0], P[1], \dots, P[n-1]$

Exercice 5.9. Comparer de manière qualitative cette spline par rapport à la spline cubique avec les conditions seulement en x_0 et la spline quadratique.

6 Spline cubique paramétrée

Définition 6.1. Soient t_0, \dots, t_{n-1} et x_i, y_i des réels pour $i = 0, \dots, n - 1$. La spline cubique paramétrée d'interpolation des (x_i, y_i) est la fonction

$$\begin{aligned} [t_0, t_{n-1}] &\rightarrow \mathbb{R}^2 \\ t &\mapsto (S_x(t), S_y(t)) \end{aligned}$$

où S_x est une spline cubique d'interpolation des points (t_i, x_i) et S_y est une spline cubique d'interpolation des points (t_i, y_i) .

On pose $h_i = t_{i+1} - t_i$. Afin de simplifier les calculs de la spline, il peut être judicieux de considérer h_i constant et même $h_i = 1$. Dans ce cas, on a $t_i = i$.

Exercice 6.2. Réécrire et simplifier le système d'équations donnant une spline cubique passant par (t_i, x_i) en supposant $t_0 = 0$ et $h_i = 1$.

Solution. On a

$$\underbrace{\begin{bmatrix} 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \end{bmatrix}}_{A^P} \times \underbrace{\begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{n-1} \end{bmatrix}}_M = 6 \underbrace{\begin{bmatrix} x_2 - x_0 \\ x_3 - x_1 \\ \vdots \\ x_n - x_{n-2} \end{bmatrix}}_{B^P}$$

Définition 6.3. Une spline cubique paramétrée $t \mapsto (S_x(t), S_y(t))$ est dite naturelle si $S_x(t)$ et $S_y(t)$ sont des splines naturelles.

Exercice 6.4. Ecrire le système d'équations donnant une spline cubique naturelle passant par (t_i, x_i) en supposant $t_0 = 0$ et $h_i = 1$.

Solution. On a

$$\underbrace{\begin{bmatrix} 1 & & & & & \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \\ & & & & & 1 \end{bmatrix}}_{A_N^P} \times \underbrace{\begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{n-1} \end{bmatrix}}_M = 6 \underbrace{\begin{bmatrix} 0 \\ x_2 - x_0 \\ x_3 - x_1 \\ \vdots \\ x_n - x_{n-2} \\ 0 \end{bmatrix}}_{B_N^P}$$

Exercice 6.5. Modifier le système précédent pour obtenir un système où la matrice principale est triangulaire supérieur.

6.1 Splines cubique paramètres fermée

Définition 6.6. Soient $x_0 < x_1 < \dots < x_{n-1}$ des réels et y_0, \dots, y_{n-1} des réels aussi. Une spline cubique S d'interpolation est dite *périodique* si on a $S'(x_0) = S'(x_{n-1})$ et $S''(x_0) = S''(x_{n-1})$.

Pour représenter une courbes fermée paramètres de \mathbb{R}^2 , on va évidemment utiliser des splines périodiques.

Exercice 6.7. Ecrire le système d'équations donnant une spline cubique périodique passant par (t_i, x_i) en supposant $t_0 = 0$ et $h_i = 1$. La condition $S''(x_0) = S''(x_n)$ se traduit immédiatement en $m_0 = m_{n-1}$. La condition $S'(x_0) = S'(x_{n-1})$ est plus difficile car elle ne fait pas intervenir directement les dérivées secondes de S . On a

$$S'(x_0) = c_0 = x_1 - x_0 - \frac{m_1 - m_0}{6} - \frac{m_0}{2} = x_1 - x_0 - \frac{m_1}{6} + 2\frac{m_0}{3}$$

On a $q'_i(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i$ et $q''_i(x) = 6a_i(x - x_i) + 2b_i$

