DÉPARTEMENT STPI
3ÈME ANNÉE MIC, 2023-2024.

APPROXIMATION DE FONCTIONS

Partie : CAO

Intervenant : Robin BOUCLIER

Table des matières

Introduction générale

Présentation générale

Classiquement, connaissant la forme analytique d'une fonction, on cherche à en déduire certaines valeurs ponctuelles (valeur en un point précis, racine, minimum...). Dans ce cours, nous nous attachons à résoudre le problème inverse : connaissant certains points (disons certaines valeurs ponctuelles), l'objectif est de construire une forme analytique d'une fonction « naturelle » (c'est-à-dire « raisonnable ») qui passe par ces points (on parlera d'**interpolation**), ou qui passe « au travers » de ces points (on parlera de **lissage**). On peut d'ores et déjà remarquer que la terminologie « naturelle » est utilisée : celle-ci nécessitera d'être précisée rigoureusement.

Afin de répondre à la problématique, il nous faut définir une famille de fonctions qui puisse passer par n'importe quels points, avoir n'importe quelle forme. On pourra alors, dans second temps, utiliser ces fonctions pour représenter (on dira aussi « modéliser ») **n'importe quelle forme**. Nous établirons alors des outils qui sont à la base de ceux utilisés à l'heure actuelle en **Conception Assistée par Ordinateur** (CAO) pour concevoir des formes aussi diverses que des avions, des voitures, des robots, fabriquer des films d'animation...

D'un point de vue résolution, satisfaire n conditions (par exemple du type $f(x_j) = y_j$ pour n points (x_j, y_j) donnés) nécessite de se situer dans un espace vectoriel de dimension n, pour trouver « à tous coups » une solution de préférence unique. Pour cela, il faut encore que cet espace vectoriel « contienne » les fonctions des formes choisies. On se concentrera dans ce cours sur une famille particulière, sans doute la plus utilisée et en fin de compte assez simple : les **splines cubiques**. Chaque fonction est constituée de **morceaux de polynômes de degré 3**, qui se « raccordent correctement », c'est à dire tels que la fonction soit **deux fois continûment dérivable** sur $\mathbb{R}$. Ainsi, pour calculer la valeur d'une telle fonction en un point, il nous suffira de déterminer d'abord de quel polynôme de degré 3 il s'agit, puis de calculer un polynôme de degré 3...

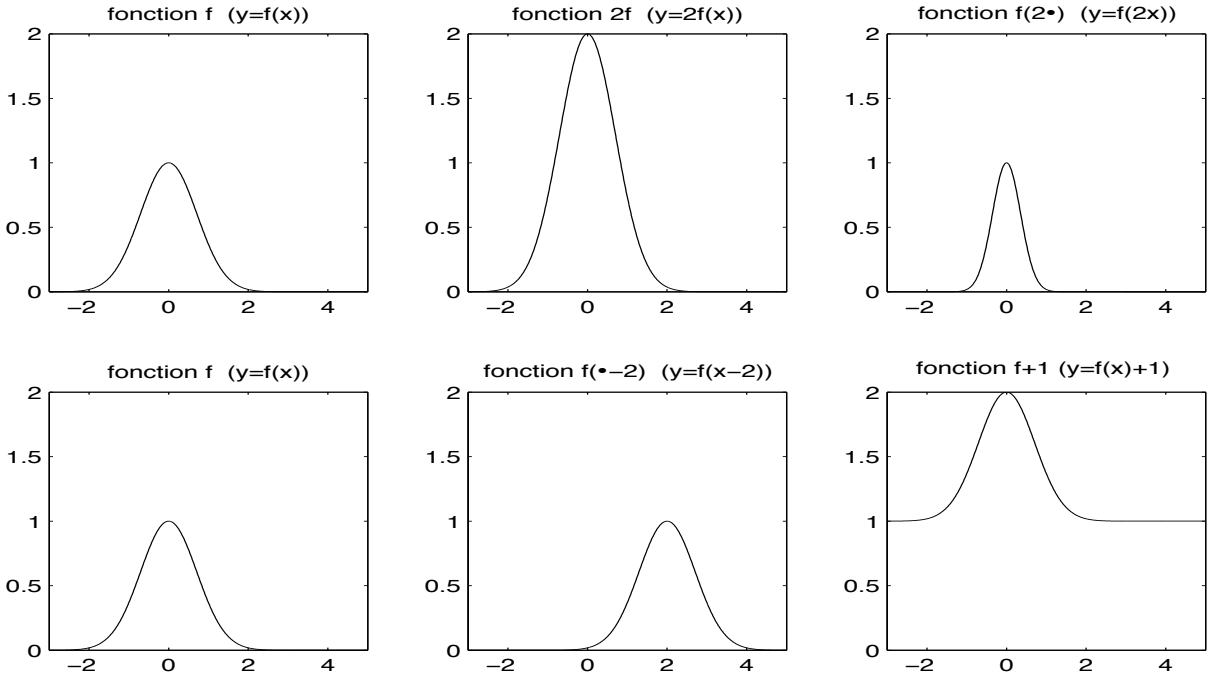
Le cours se décompose en deux parties : dans la première, on s'intéresse plus particulièrement à l'approximation de données tandis qu'on fait le lien avec la représentation d'une forme géométrique dans la seconde partie.

Notations

- $[k : n]$ est l'ensemble des entiers i tels que $k \leq i \leq n$
- $[a \,..\, b]$ est l'ensemble des réels x tels que $a \leq x \leq b$

- De même pour les ensembles de réels $]a\,..\,b[$, $[a\,..\,b[$, $]a\,..\,b]$
- $\mathbb{P}_k$ désigne l'ensemble des polynômes de degré au plus k.
- $\mathrm{L}^2(\mathbb{R})$ désigne l'ensemble des fonctions de $\mathbb{R}$ dans $\mathbb{R}$ qui sont de carré intégrable, c'est à dire l'ensemble des fonctions f de $\mathbb{R}$ dans $\mathbb{R}$ telles que $\int_{\mathbb{R}}(f(x))^2\,\mathrm{d}x$ soit défini. De même, $\mathrm{L}^2([a\,..\,b])$ désigne l'ensemble des fonctions de $[a\,..\,b]$ dans $\mathbb{R}$ qui sont de carré intégrable, c'est à dire l'ensemble des fonctions f de $[a\,..\,b]$ dans $\mathbb{R}$ telles que $\int_{[a\,..\,b]}(f(x))^2\,\mathrm{d}x$ soit défini.
- Le $\bullet$ désigne la variable imposée par le contexte. Ainsi par exemple, si f est une fonction de variable réelle, la fonction $g = f(2\bullet)$ est la fonction définie par $\forall x \in \mathbb{R}\ ,\ g(x) = f(2x)$, et la fonction $h = f(\bullet + a)$ est la fonction définie par $\forall x \in \mathbb{R}\ ,\ h(x) = f(x+a)$.

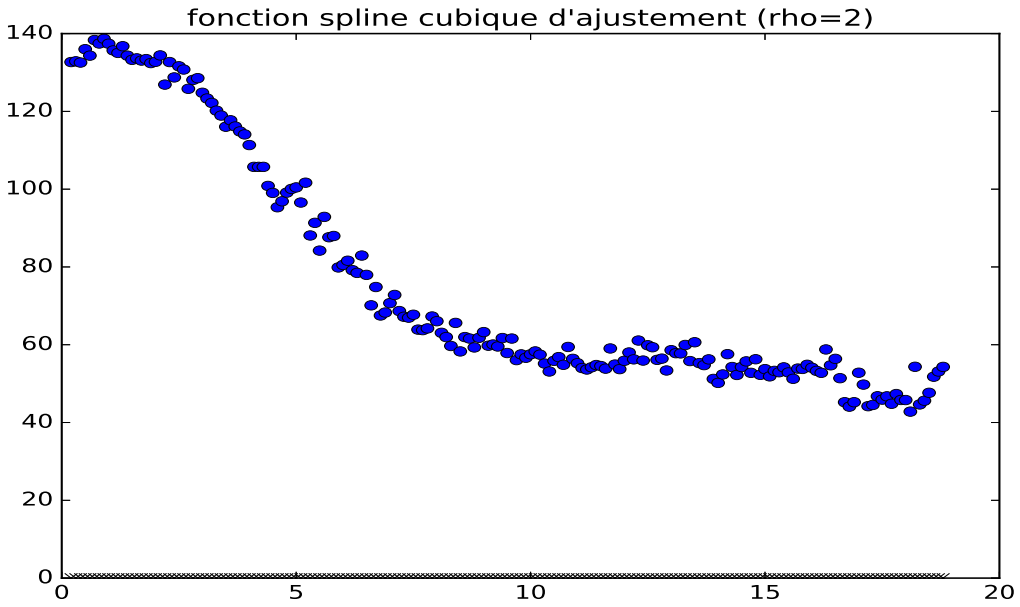
Remarquez que si $g = 2\,f$, la courbe représentative de g est obtenue à partir de celle de f par une dilatation d'axe Oy et de rapport 2 (affinité d'axe Oy et de rapport 2), tandis que si $g = f(2\bullet)$, la courbe représentative de g est obtenue à partir de celle de f par une compression d'axe Ox et de rapport 2 (affinité d'axe Ox et de rapport $\frac{1}{2}$). De la même façon, constatez que si $g = f + a$ la courbe représentative de g est obtenue à partir de celle de f par une translation d'axe Oy de a, tandis que si $g = f(\bullet + a)$, la courbe représentative de g est obtenue à partir de celle de f par une translation d'axe Ox de $-a$. Ceci est illustré par la figure ci-dessous.



Un problème type pour l'approximation de données : PHmétrie

À titre exemple concernant la première partie, nous allons considérer le problème suivant. On dispose de différentes mesures de la neutralisation d'un acide simple par une base simple, et on désire obtenir une bonne évaluation de la position du point d'équilibre, qui est le point d'inflexion d'un modèle sous-jacent au processus. On désire que cette évaluation ne mette en oeuvre aucune

intervention humaine, de façon par exemple à ce qu'elle puisse faire partie d'un ensemble plus complet à résoudre entièrement par ordinateur. On appellera les données $(x_j\,,y_j)_{i=0:n}$, où y_j est le PH mesuré pour un volume d'acide x_j versé. Par exemple, on peut voir un type de répartition de ces données dans la figure ci-dessous.



Les difficultés pour résoudre ce problème sont doubles. D'une part, on ne connait pas du côté de la physique, de modèle mathématique « idéal » pour traduire une telle réaction (c'est-à-dire que la physique ne nous fourni pas de fonction f capable de représenter proprement les données). D'autre part, les données dont on dispose sont « bruitées », c'est à dire ne sont pas exactes (elles possèdent des « erreurs de mesure »). L'objectif de la première partie de ce cours est alors de répondre à cette double problématique. Plus précisément, ce problème de PHmétrie sera résolu en TP.

Première partie

Splines cubiques naturelles, interpolation et lissage de données

Chapitre 1

Introduction aux splines cubiques

1.1 Motivation : avantages et inconvénients des polynômes

Les polynômes ont été les premières fonctions utilisées comme outils pour approcher des données ou d'autres fonctions (par exemple, les développements limités ne sont rien d'autre que l'approximation d'une fonction par un polynôme, celui qui a ses dérivées identiques à celles de la fonction, au point d'approximation). Ils présentent en effet un certain nombre d'avantages mais aussi des inconvénients qui, en fin de compte, limitent fortement leur utilisation à l'heure actuelle pour décrire des données ou des géométries. On cite les principaux avantages et inconvénients des polynômes ci-dessous.

Avantages :

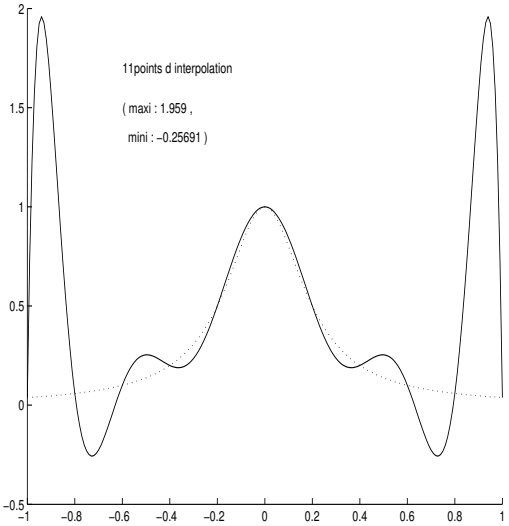
- Simplicité et rapidité de calcul (*cf.* schéma de Horner).
- La dimension de l'espace vectoriel qu'ils génèrent peut être fixée arbitrairement (par le degré des polynômes utilisés).
- Théoriquement, ils peuvent modéliser n'importe quelle fonction puisqu'il y a convergence, en norme de la convergence uniforme, sur un compact et lorsque n tend vers l'infini, du polynôme d'approximation de degré n vers toute fonction continue.

Inconvénients :

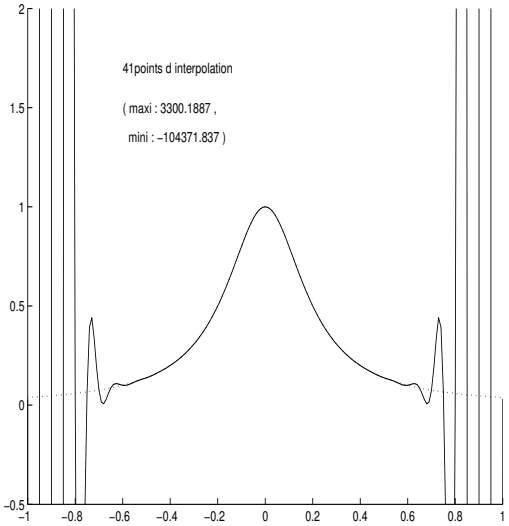
- Le **phénomène de Runge** (oscillations du polynôme d'interpolation) interdit toute utilisation des polynômes pour interpoler.
- La **forme générale** des polynômes est assez particulière et permet mal d'approcher certaines fonctions ou données. Bien sûr la convergence mentionnée ci-dessus garantit que plus on y met le prix (c'est-à-dire plus le degré du polynôme d'approximation est élevé) meilleure sera l'approximation ; mais pour de nombreuses formes (par exemple, extremums très « plats », parties « horizontales » aux extrémités), il faudra un degré important pour que le polynôme ait la forme désirée. À tel point que l'on peut dire que la propriété de convergence est en théorie intéressante, mais qu'elle ne permet pas en pratique de satisfaire les besoins des utilisateurs.
- **Conditionnement** : La détermination du polynôme d'interpolation par méthode directe implique la résolution d'un système linéaire presque toujours très mal conditionnée.

- **Stabilité** : Un petite variation sur les données (valeur de x_j ou y_j) entraine, presque toujours, une variation importante du polynôme résultat. Ce phénomène est lié à la question du conditionnement ci-dessus, mais il est visible dans des circonstances quelquefois différentes.

Ces inconvénients peuvent se résumer en disant que les polynômes sont des fonctions trop « rigides », avec une propension à osciller et des difficultés pour présenter des parties « plates », si bien qu'ils ne conviennent pas pour la plupart des applications pratiques. À titre d'exemple, les deux figures suivantes montrent les oscillations présentées par les polynômes d'interpolation de points régulièrement répartis sur la fonction d'équation $\frac{1}{1+25x^2}$ (illustration du phénomène de Runge).



11 points d'interpolation.



41 points d'interpolation.

Toutes ces raisons font qu'on **utilise que peu les polynômes pour interpoler**, et pratiquement uniquement avec des petits degrés.

L'objectif de ce chapitre est dès lors de construire une autre famille de fonctions qui présente l'essentiel des avantages ci-dessus mentionnés, et qui n'en présente pas les inconvénients. Pour cela, nous allons adopter deux approches distinctes qui conduisent au même résultat : l'approche intuitive « polynômes par morceaux » (voir §1.2) et l'approche plus générale par méthode « variationnelle » (*cf.* §1.5).

1.2 Approche « polynômes par morceaux »

Le but à présent est de conserver la forme polynomiale, mais de faire en sorte de ne pas augmenter le degré du polynôme. L'idée pour cela est de fabriquer une fonction qui soit une succession de polynômes de degré peu élevé (degré 3 dans ce cours, degré k dans le cas général) tels que les raccordements entre polynômes soient les plus réguliers possible. Dans la suite, nous considèrerons désormais les x_j rangés dans l'ordre croissant ($x_j < x_{j+1}$).

1.2.1 Interpolation linéaire par morceaux

On illustre d'abord cette idée par un exemple simple appelé « interpolation linéaire par morceaux » : pour interpoler des données $(x_j, y_j)_{j=0:n}$, nous pouvons relier, pour tout $j \in [0 : n-1]$, le point (x_j, y_j) au point (x_{j+1}, y_{j+1}) par un segment de droite. On obtient ainsi une fonction particulièrement simple, définie sur $[x_0 \,..\, x_n]$, qui présente pratiquement tous les avantages mentionnés au §1.1, et pratiquement aucun des inconvénients mentionnés. Remarquons d'ailleurs qu'il est facile de définir une fonction sur $\mathbb{R}$ (le prolongement $\forall x \in]-\infty \,..\, x_0]$, $f(x) = y_0$ et $\forall x \in [x_n \,..\, \infty[$, $f(x) = y_n$ est naturel et simple), et que le calcul de cette fonction est même extrêmement simple puisqu'il n'y a que des polynômes de degré 1 à calculer. À priori, on y gagne donc quasiment à tous les points de vue.

Malheureusement, cette famille de fonctions présente un inconvénient de taille qui la rend inutilisable pour la plupart des applications : la non dérivabilité ponctuelle (en les points d'interpolation), qui rend impossible toute évaluation des dérivées, et qui a une conséquence moins visible à priori, à savoir une faible vitesse de convergence et une distance trop importante entre l'interpolant et la fonction à interpoler.

Exercice 1.2.1 *Pour illustrer le phénomène, tracer l'interpolation linéaire par morceaux de la fonction d'équation* $\frac{1}{1+25x^2}$ *sur* $[-1, 1]$ *avec 11 points d'interpolation. Comparer avec le polynôme d'interpolation correspondant. Tracer enfin la dérivée de la fonction et de l'interpolation linéaire par morceaux.*

1.2.2 Interpolation cubique par morceaux

On cherche par conséquent à améliorer l'idée précédente. Toujours pour avoir des fonctions très simples à calculer, on peut chercher à utiliser des « polynômes par morceaux », c'est-à-dire des fonctions qui soient une succession de polynômes. De plus, comme on l'a vu ci-dessus, on peut exiger que la fonction ainsi obtenue soit « suffisamment dérivable », de façon à pouvoir l'utiliser pour approcher les dérivées du « processus ». Pour cela, il est clair qu'il suffit que les différents polynômes se « raccordent » de façon suffisamment régulière. On se limite dans ce cours au **degré 3** et à la **continuité** $\mathcal{C}^2$. Les fonctions de cette famille seront appelées « splines cubiques », ou bien « fonctions spline cubiques », et seront souvent notées σ. On les définit comme suit :

Définition 1.1 (Approche polynomiale) *On appelle* ***spline cubique de noeuds*** $(x_j)_{j=0:n}$ *toute fonction réelle de variable réelle, qui est deux fois continûment dérivable sur* $\mathbb{R}$ *et dont la restriction à chaque intervalle* $[x_j \,..\, x_{j+1}]$*, ainsi que* $]-\infty \,..\, x_0]$ *et* $[x_n \,..\, +\infty[$ *est un polynôme de degré 3.*

À l'aide de cette famille de fonctions, on peut montrer qu'il est possible d'interpoler n'importe quelle famille de données $(x_j, y_j)_{j=0:n}$, c'est-à-dire en d'autres termes, de trouver une fonction σ de cette famille qui vérifie $\forall j \in [0 : n]$, $\sigma(x_j) = y_j$.

Exercice 1.2.2 *En supposant que l'on connaisse* $\sigma'(x_0) = y'_0$*, et* $\sigma''(x_0) = y''_0$*, montrer qu'il existe une et une seule fonction sur* $[x_0 \,..\, x_n]$ *vérifiant les conditions ci-dessus.*

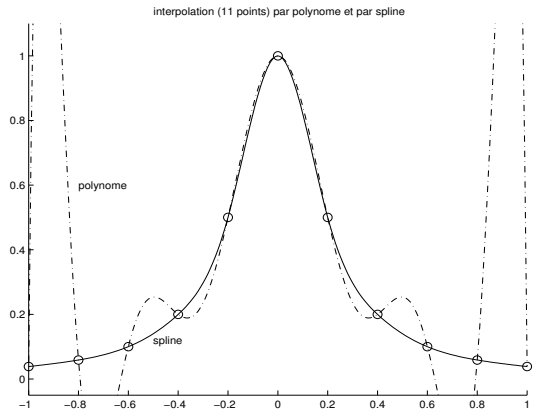
La résolution de l'exercice précédent nous amène à la conclusion suivante : on dispose de « deux degrés de liberté » (puisque nous pouvons fixer arbitrairement $\sigma'(x_0)$ et $\sigma''(x_0)$), c'est-à-dire qu'il nous faut se donner deux conditions supplémentaires pour assurer l'unicité de la fonction σ sur $[x_0 \,..\, x_n]$. Ensuite, il nous faut définir un critère pour prolonger σ sur $]-\infty \,..\, x_0]$ et sur $[x_n \,..\, \infty[$. Il est possible de démontrer (non demandé dans ce cours) que si les conditions sont situées à une seule extrémité de l'intervalle $[x_0 \,..\, x_n]$ (comme préalablement supposé), la détermination de σ est instable. Il faut donc trouver des conditions « aux deux bouts » pour permettre la stabilité de la fonction.

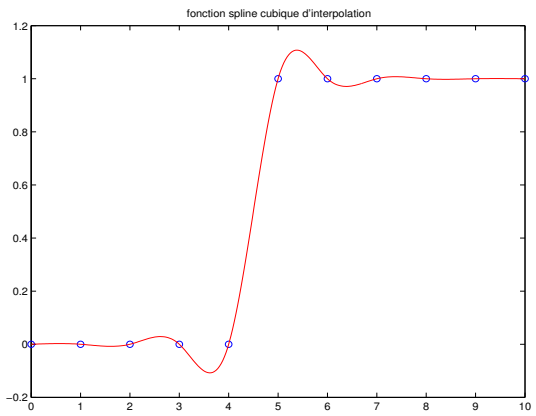
En fonction du choix des conditions de bord sur $[x_0 \,..\, x_n]$ et du critère pour le prolongement, plusieurs sortes de splines stables peuvent être construites. Dans ce cours, on se concentrera sur les **splines cubiques naturelles** qui ont l'intérêt de minimiser les oscillations (*cf.* ci-dessous §1.4). Cette propriété explique l'utilisation intensive de ces fonctions pour l'interpolation et l'approximation de données, ainsi que pour la génération de géométrie en CAO. Pour ces splines, on rajoute les deux conditions suivantes pour la construction sur $[x_0 \,..\, x_n]$: $\sigma''(x_0) = \sigma''(x_n) = 0$. Ensuite, pour le prolongement sur $\mathbb{R}$, on prend tout simplement une évolution affine de σ sur $]-\infty \,..\, x_0]$ et sur $[x_n \,..\, +\infty[$.

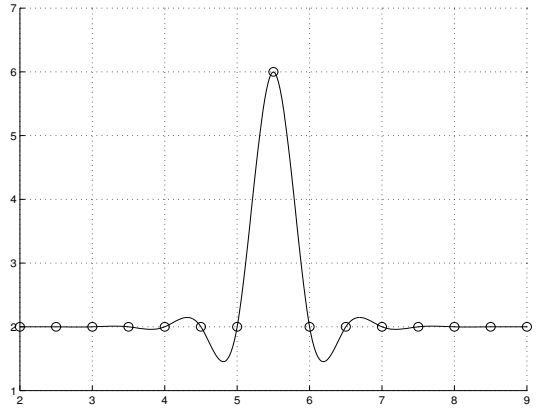
Remarque 1.1 *Une autre famille de spline que l'on rencontrera dans ce cours concerne les* ***splines périodiques*** *qui se construisent aisément dans le cas où $\sigma(x_0) = \sigma(x_n)$. En effet, il suffit alors de prendre les deux conditions $\sigma'(x_0) = \sigma'(x_n)$ et $\sigma''(x_0) = \sigma''(x_n)$ afin d'obtenir une fonction périodique sur $\mathbb{R}$ (de période $x_n - x_0$) et de continuité $\mathcal{C}^2$, ce qui répond complètement au problème posé.*

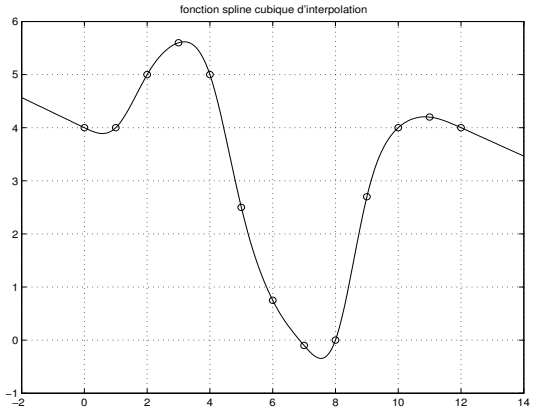
1.3 Exemples

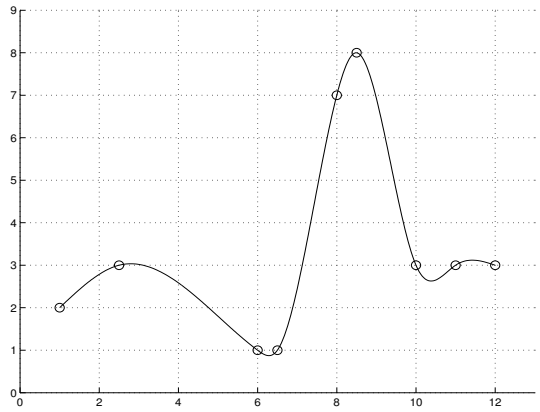
Les figures suivantes présentent les splines cubiques naturelles d'interpolation de certaines données, indiquées par des $\circ$ ou des $\times$. On remarquera que les fonctions obtenues correspondent assez bien au tracé que l'on aurait pu faire « à la main », et on reconnaîtra le phénomène de Runge pour les polynômes (voir, plus précisément, première figure de gauche ci-après).

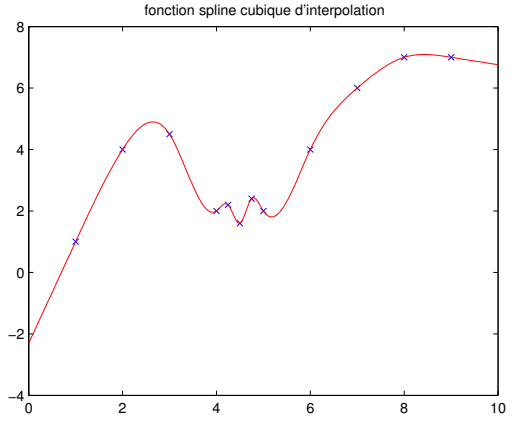


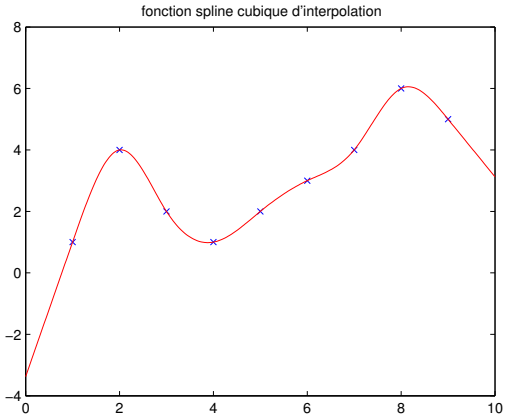


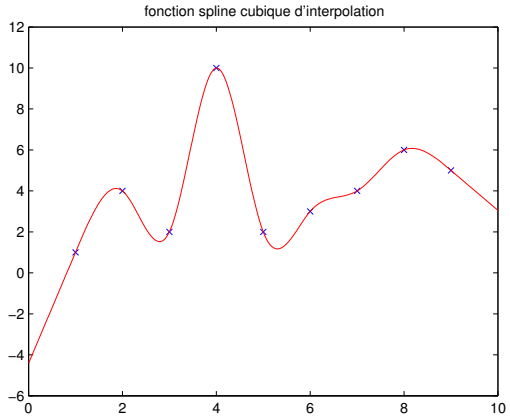












Afin de comprendre (intuitivement) le comportement des splines, la première figure à droite peut être analysée plus attentivement. Les données sont d'abord égales à zéro, puis, « brutalement », égales à 1. On constate, d'une façon qui peut surprendre à priori, que, pour « passer de 0 à 1 », la spline passe par des valeurs négatives, puis dépasse 1. Ce phénomène est caractéristique des splines. On peut dire qu'afin d'être le plus « lisse » possible, et d'éviter des variations trop brutales de pente, la spline « anticipe » le nécessaire virage pour aller de 0 à 1 par un « contre-virage » avant le saut de 0 à 1, puis continue un peu sur sa lancée après avoir atteint la première valeur 1. C'est ainsi pour minimiser la totalité des virages (en fait, comme on le verra au § suivant, la quantité $\int(\sigma''(x))^2dx)$ que la spline présente ces oscillations. Cette propriété est très

générale (et a beaucoup de conséquences pratiques).

1.4 Propriété générale des splines cubiques naturelles

Théorème 1.1 (Carl de Boor, 1963) *Soient $a = x_0$ et $b = x_n$,*

- *Parmi toutes les fonctions f deux fois dérivables, dont la dérivée seconde est dans $\mathrm{L}^2([a\,..\,b])$, et qui interpolent les points $(x_j, y_j)_{j=0:n}$, la spline cubique naturelle est celle qui minimise $\displaystyle\int_a^b (f''(x))^2\,\mathrm{d}x$.*
- *Parmi toutes les fonctions f deux fois dérivables, dont la dérivée seconde est dans $\mathrm{L}^2(\mathbb{R})$, et qui interpolent les points $(x_j, y_j)_{j=0:n}$, la spline cubique naturelle est celle qui minimise $\displaystyle\int_{\mathbb{R}} (f''(x))^2\,\mathrm{d}x$.*

Remarque 1.2 *En notant $\mathrm{D}^{-2}\mathrm{L}^2([a\,..\,b])$ et $\mathrm{D}^{-2}\mathrm{L}^2(\mathbb{R})$ l'ensemble des fonctions dont la dérivée seconde existe et est dans $\mathrm{L}^2([a\,..\,b])$ et $\mathrm{L}^2(\mathbb{R})$, respectivement, le théorème ci-dessus peut aussi s'écrire sous la forme :*

- $\forall f \in \mathrm{D}^{-2}\mathrm{L}^2([a\,..\,b])\,,\ \Big(\forall j \in [0:n]\ ,\ f(x_j) = y_j\Big)$

$$\Longrightarrow \quad \int_a^b (\sigma''(x))^2\,\mathrm{d}x \leq \int_a^b (f''(x))^2\,\mathrm{d}x. \tag{1.1}$$

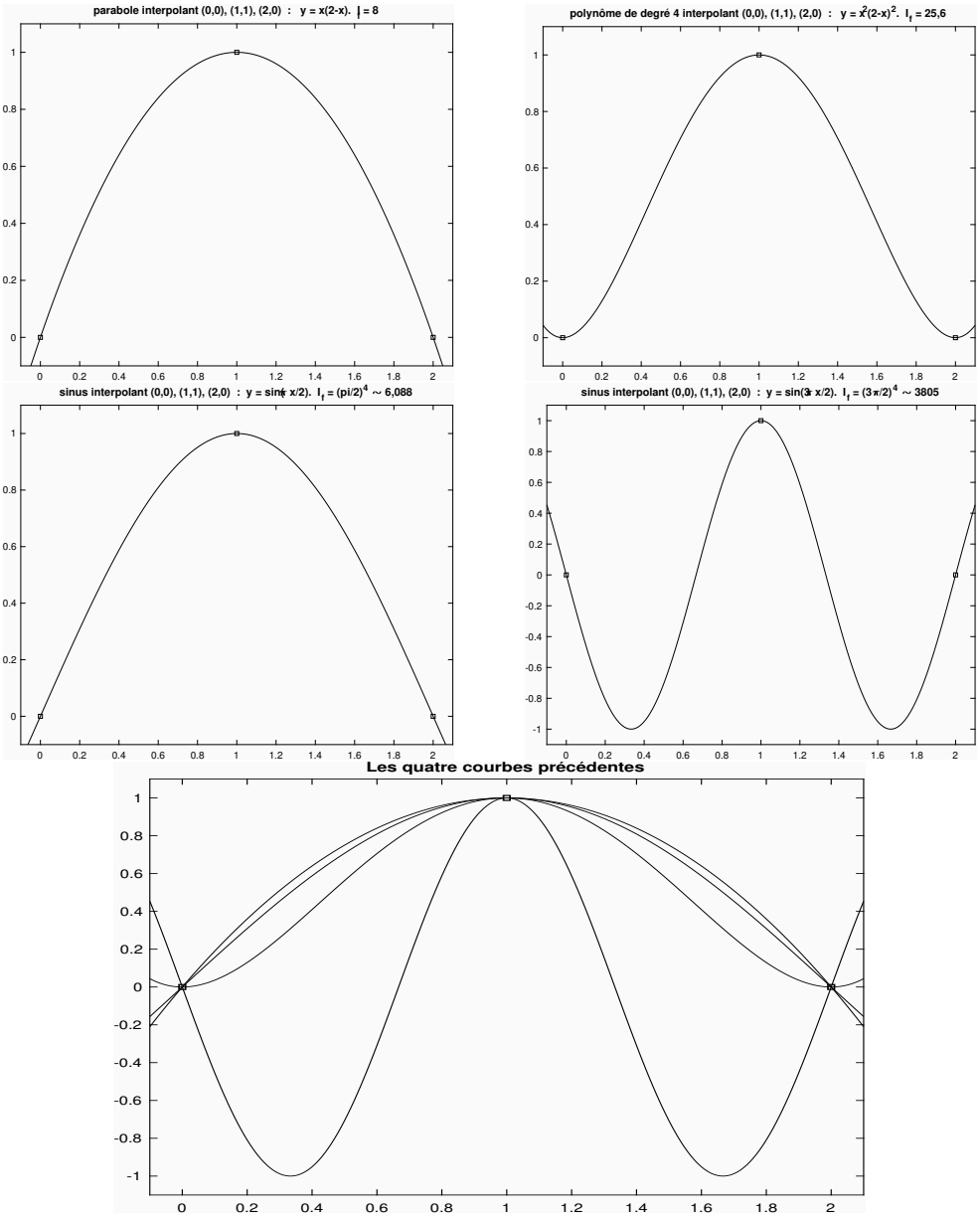
- $\forall f \in \mathrm{D}^{-2}\mathrm{L}^2(\mathbb{R})\,,\ \Big(\forall j \in [0:n]\ ,\ f(x_j) = y_j\Big)$

$$\Longrightarrow \quad \int_{\mathbb{R}} (\sigma''(x))^2\,\mathrm{d}x \leq \int_{\mathbb{R}} (f''(x))^2\,\mathrm{d}x. \tag{1.2}$$

Démonstration. La démonstration de ce théorème s'appuie essentiellement sur l'intégration par parties et les conditions de bord des splines cubiques naturelles. Celle-ci sera faite en cours. □

Ce théorème est important car $\int_a^b (f''(x))^2\,\mathrm{d}x$ (resp. $\int_{\mathbb{R}} (f''(x))^2\,\mathrm{d}x$) est significatif de l'oscillation de la fonction f entre a et b (resp. sur $\mathbb{R}$). Afin de s'en rendre compte, on peut prendre un cas particulier : on interpole les trois points $(0\,,0)$, $(1\,,1)$ et $(2\,,0)$ avec différentes fonctions (qui oscillent plus ou moins), et on évalue la quantité $I_f = \int_a^b (f''(x))^2\,\mathrm{d}x$. Pus précisément, on prend les exemples suivants :

- f est la parabole passant par les trois points. Alors $f(x) = x(2-x)$, et $I_f = 8.$ (voir figure en haut à gauche ci-dessous)
- f est le polynôme de degré 4 passant par les trois points, et dont la tangente aux premier et dernier point est horizontale. Alors $f(x) = x^2(2-x)^2$, et $I_f = 25{,}6.$ (voir figure en haut à droite ci-dessous)
- $f(x)$ est une sinusoïde, de période $2/(2p{+}1)$, pour $p{\in}\mathbb{N}$. Alors $f(x) = (-1)^p \sin(\frac{2p+1}{2}\pi x)$, et $I_f = (\frac{2p+1}{2}\pi)^4$, soit, si $p{=}0$, $I_f \simeq 6{,}088$ (voir figure au milieu à gauche ci-dessous) et si $p{=}1$, $I_f \simeq 493$ (voir figure au milieu à droite ci-dessous).



On peut observer que différentes fonctions f ont une valeur très différente de I_f, et que ceci semble bien être relié à l'oscillation de la fonction f (voir figure en bas ci-dessus qui compare les différentes fonctions). Ceci est très général : avec les mains, plus une fonction oscille, plus sa dérivée première a besoin de variations importantes (pour que la fonction puisse « monter », « descendre », etc), donc plus la dérivée seconde doit être importante.

Le théorème exprime donc le fait que la spline cubique naturelle est **la fonction qui oscille le moins possible** (sur $[a\,..\,b]$ comme sur $\mathbb{R}$), parmi toutes les fonctions qui interpolent les données $(x_j, y_j)_{j=0:n}$.

Exercice 1.4.1 *Retrouver les différentes fonctions d'interpolation des trois points* $(0\,,0)$, $(1\,,1)$ *et* $(2\,,0)$ *ci-dessus puis, calculer* I_f *pour chacune d'elle.*

1.5 Approche variationnelle

Il s'agit maintenant de prendre le problème dans l'autre sens. Dans l'approche « polynômes par morceaux » (§1.2), on construit la spline comme une fonction polynomiale par morceaux pour interpoler des données et on remarque que, moyennant certaines conditions de bord, on aboutit à la fonction qui oscille le moins. À présent, le but est de rechercher la fonction d'interpolation qui oscille le moins afin de retrouver la spline cubique naturelle telle que définie au §1.2.

Pus précisément, d'un point de vue mathématique, soit la semi-norme $|f|$ défini par $|f|^2 = \int_{\mathbb{R}}(f''(x))^2\,dx$. Il s'agit alors de rechercher la fonction qui minimise $|f|$ sur l'ensemble des fonctions f pour lesquelles cette quantité a un sens (c'est-à-dire sur $\mathrm{D}^{-2}\mathrm{L}^2(\mathbb{R})$) et qui interpolent les données $(x_j, y_j)_{j=0:n}$. La détermination d'une telle fonction nécessite un travail théorique assez important qui sort du cadre de ce cours. Évidemment, on aboutit à la spline cubique naturelle d'interpolation. Il en découle la définition suivante :

Définition 1.2 (Approche variationnelle) *On appelle **spline cubique naturelle d'interpolation des points** $(x_j, y_j)_{j=0:n}$ l'unique fonction σ de $\mathrm{D}^{-2}\mathrm{L}^2(\mathbb{R})$ telle que :*

- $\forall j \in [0:n]\ ,\ \ \sigma(x_j) = y_j$.
- $\forall f \in \mathrm{D}^{-2}\mathrm{L}^2(\mathbb{R})\ ,\ \ \Big(\forall j \in [0:n]\ ,\ f(x_j) = y_j\Big) \Rightarrow \int_{\mathbb{R}}(\sigma''(x))^2\,\mathrm{d}x\ \leq\ \int_{\mathbb{R}}(f''(x))^2\,\mathrm{d}x$.

Remarque 1.3 *Dans la définition ci-dessus, on peut remplacer $\mathrm{D}^{-2}\mathrm{L}^2(\mathbb{R})$ par $\mathrm{D}^{-2}\mathrm{L}^2([a\,..\,b])$ et $\int_{\mathbb{R}}$ par $\int_a^b$.*

Proposition 1.1 *La spline cubique naturelle d'interpolation des points $(x_j, y_j)_{j=0:n}$ vérifie sur $[x_0\,..\,x_n]$:*

- *Sur chaque $[x_j\,..\,x_{j+1}]$, σ est un polynôme de degré 3.*
- *σ, σ', et σ'' sont définies et continues sur tout le domaine d'étude.*
- *$\sigma''(x_0) = \sigma''(x_n) = 0$.*

Et, il faut ajouter dans le cas d'une étude sur $\mathbb{R}$:

- *Sur $]-\infty\,..\,x_0]$ et sur $[x_n\,..\,+\infty[$, σ est un polynôme de degré 1.*

***Réciproquement** :*
Si une fonction σ satisfait les conditions ci-dessus et $\forall j \in [0:n]\ ,\ \sigma(x_j) = y_j$, alors σ est la spline cubique naturelle d'interpolation des points $(x_j, y_j)_{j=0:n}$.
On dit : « σ est un polynôme de degré 3 par morceaux », dont la dérivée seconde est continue, et nulle en x_0 et x_n.

Remarque 1.4 (Analogie mécanique) *On considère souvent que $\int_a^b (f''(x))^2\,dx$ est une (grossière) approximation de $\int_a^b \frac{(f''(x))^2}{1+(f'(x))^2}\,dx$ qui est l'intégrale du carré de la courbure de la fonction f. En mécanique, il est bien connu que cette quantité est proportionnelle à l'« énergie de déformation » d'un câble infiniment mince qui aurait la forme de la fonction f. Ainsi, la spline cubique naturelle dessine la forme d'équilibre que prendrait un câble infiniment mince qui serait astreint à passer par les points $(x_j, y_j)_{j=0:n}$. C'est de là que vient le nom « spline » (mot anglais désignant les « lattes » qu'utilisent les dessinateurs pour tracer des courbes passant par des points).*

1.6 Intérêt des fonctions $\mathcal{C}^2$

En complément, on termine ce chapitre en présentant deux situations concrètes et deux situations plus scientifiques qui illustrent l'importance d'avoir des fonctions suffisamment dérivables.

Situation A :

On suppose tout d'abord que l'on se situe sur une route, et que le tracé de la route présente une discontinuité de la dérivée seconde. Pour simplifier, on peut considérer le cas d'une route droite suivie d'un arc de cercle. Le conducteur doit alors tourner brutalement son volant pour passer, sans transition, de la droite au cercle, ce qui bien sûr est impossible de faire exactement, et entraînerait nécessairement un écart par rapport à l'axe de la voie empruntée. Ce point est particulièrement important en ce qui concerne les bretelles de sortie d'autoroute, pour lesquelles la diminution progressive du rayon de courbure du tracé de la route est indispensable.

Situation B :

Dans la même idée, on suppose ensuite que l'on se situe dans un train, et que là aussi, la voie ferrée est constituée d'un segment de droite suivi d'un arc de cercle. Il s'agit maintenant du confort des passagers : au changement brutal de courbure, se produit un changement brutal de force centrifuge, et les passagers se trouvent tout d'un coup « repoussés » à l'extérieur du virage. Par contre, si le changement de courbure est progressif, les passagers peuvent réagir progressivement à la force centrifuge qui augmente en se penchant de façon à se situer selon la verticale apparente.

Situation C :

Concernant une situation plus scientifique, on peut reprendre le problème de l'introduction, à savoir le cas d'un chimiste qui désire déterminer le point d'équilibre d'une solution acide-base. Celui-ci étant le point d'inflexion, son existence nécessite que la fonction qui interpole les données soit au moins deux fois dérivable.

Situation D :

Enfin, on peut terminer par examiner la situation de quelqu'un qui doit approcher la solution d'une équation différentielle d'ordre k. Il pourra le faire à l'aide d'une fonction, au moins k fois dérivable, qui satisfait « au mieux » l'équation différentielle.

Chapitre 2

Expression des splines cubiques

Il existe plusieurs façons d'écrire les splines cubiques. On peut distinguer deux types d'écriture : les écritures locales et les écritures globales. La version locale (c'est-à-dire que la fonction est définie de façon différente sur chaque intervalle $[x_j \,..\, x_{j+1}[$) est la plus simple et la plus rapide du point de vue des calculs, c'est pourquoi, c'est celle-ci qui est majoritairement utilisée. Les écritures globales (c'est-à-dire que la fonction est définie à l'aide d'une unique formule sur $\mathbb{R}$) impliquent quant à elles un cout de calcul plus élevé et conduisent, en général, à des systèmes linéaires mal conditionnés lorsque l'on cherche à représenter des données ou une géométrie avec la spline. Par conséquent, nous nous focalisons dans ce chapitre sur l'écriture local de la spline uniquement.

2.1 Expression analytique

Dans les écritures locales, on cherche à définir la spline σ comme un polynôme de degré 3 sur chaque intervalle $[x_j \,..\, x_{j+1}[$. Pour l'évaluation en un x donné, il faudra donc d'abord déterminer dans quel intervalle se trouve x puis user de la formule analytique relative à cet intervalle. La définition du polynôme de degré 3 correspondant à la restriction de la spline σ à chaque intervalle $[x_j \,..\, x_{j+1}[$ se fait naturellement à l'aide de son développement limité en x_j. $\forall x \in [x_j \,..\, x_{j+1}[$, on peut écrire :

$$\sigma(x) = \sigma(x_j) + (x - x_j)\,\sigma'(x_j) + \frac{(x - x_j)^2}{2}\,\sigma''(x_j) + \frac{(x - x_j)^3}{6}\,\sigma'''(x_j+). \qquad (2.1)$$

Exercice 2.1.1 *Montrer que tout polynôme de degrés 3 dans $[x_j \,..\, x_{j+1}[$ peut toujours s'écrire, de façon exacte, sous la forme ci-dessus.*

Remarque 2.1 *Il est à noter que, du fait des propriétés de la spline, les quantités $\sigma(x_j)$, $\sigma'(x_j)$, $\sigma''(x_j)$ sont parfaitement définies. Par contre, $\sigma'''(x_j)$ n'est en général pas défini. En effet, σ''' est une fonction constante par morceaux car σ est un polynôme de degré 3 par morceaux et x_j est précisément une jonction entre deux polynômes de degré 3. Par contre, $\sigma'''(x_j+) = \lim_{x \to x_j\,,\, x > x_j} \sigma'''(x)$ est parfaitement défini, et est égal à la valeur constante de $\sigma'''(x)$ dans l'intervalle $]x_j \,..\, x_{j+1}[$.*

On note, dans la suite, pour tout $j \in [0:n]$,

$$\sigma_j = \sigma(x_j),\ \sigma'_j = \sigma'(x_j),\ \sigma''_j = \sigma''(x_j),\ \text{et}\ \sigma'''_j = \sigma'''(x_j+) = \lim_{x\to x_j\,,\,x>x_j} \sigma'''(x), \tag{2.2}$$

de sorte que l'écriture de σ sur $\mathbb{R}$ peut se résumer, de façon analytique, comme suit :

Définition 2.1 (Expression locale de σ)

$$\begin{aligned}
&\forall j \in [0:n-1]\ ,\ \ \forall x \in [x_j\,..\,x_{j+1}]\ ,\\
&\qquad \sigma(x) \ = \sigma(x_j) + (x-x_j)\sigma'(x_j) + \frac{(x-x_j)^2}{2}\,\sigma''(x_j) + \frac{(x-x_j)^3}{6}\,\sigma'''(x_j+)\\
&\qquad\qquad = \sigma_j + (x-x_j)\sigma'_j + \frac{(x-x_j)^2}{2}\,\sigma''_j + \frac{(x-x_j)^3}{6}\,\sigma'''_j.\\
&\forall x \in]-\infty\,..\,x_0]\ ,\\
&\qquad \sigma(x) \ = \sigma_0 + (x-x_0)\sigma'_0.\\
&\forall x \in [x_n\,..\,+\infty[\ ,\\
&\qquad \sigma(x) \ = \sigma_n + (x-x_n)\sigma'_n.
\end{aligned} \tag{2.3}$$

Remarque 2.2 *Il est à noter que le calcul des dérivées de la spline est direct avec l'expression* (2.3)*. On trouve sur* $[x_j\,..\,x_{j+1}]$*, bien évidemment,* $\sigma'(x) = \sigma'_j + (x-x_j)\sigma''_j + \frac{(x-x_j)^2}{2}\sigma'''_j$ *et* $\sigma''(x) = \sigma''_j + (x-x_j)\sigma'''_j$.

L'objectif des chapitres suivants de cette première partie du cours va donc être de déterminer les 4-uplets $\left(\sigma_j, \sigma'_j, \sigma''_j, \sigma'''_j\right), \forall j \in [0:n]$ tels que la spline exprimée Eq. (2.3) représente, en un certain sens, des données.

2.2 Algorithmes de calcul

2.2.1 Calcul en un point isolé

Considérons que nous possédions les quantités $\left(\sigma_j, \sigma'_j, \sigma''_j, \sigma'''_j\right)_{j=0:n}$, on s'intéresse à présent au développement de méthodes numériques capable d'évaluer la valeur $\sigma(x)$ de σ en un point x fixé en utilisant la formule (2.3). Comme indiqué ci-dessus, il faut d'abord déterminer j tel que $x \in [x_j\,..\,x_{j+1}[$, puis $\sigma(x)$ compte tenu que $x \in [x_j\,..\,x_{j+1}[$. Ceci peut se faire par l'algorithme suivant (il calcule et utilise $j=-1$ si $x < x_0$ et $j=n$ si $x > x_n$) :

```
% Algorithme CalculSigma(x)
% Calcule σ(x) connaissant (x_j, σ_j, σ'_j, σ''_j, σ'''_j)_{j=0:n} (et bien sûr x)
    j ← −1  ;  A ← σ_0  ;  B ← σ'_0  ;  C ← 0  ;  D ← 0
    Tant que j < n  et  x ≥ x_{j+1}
        ‖  j ← j + 1
```

$$\begin{array}{l} \| \quad A \leftarrow \sigma_j \ ; \ B \leftarrow \sigma'_j \ ; \ C \leftarrow \sigma''_j/2 \ ; \ D \leftarrow \sigma'''_j/6 \\ h_x = x - x_{\max(j,0)} \\ \sigma(x) = A + h_x \Big(B + h_x \big(C + h_x \mathrm{D} \big) \Big) \end{array}$$

Remarque 2.3 *On peut remarquer que l'on a utilisé la formulation de Horner (dernière ligne) pour limiter le nombre d'opérations afin d'évaluer le polynôme sur* $[x_j \,..\, x_{j+1}[$.

2.2.2 Calcul pour une famille de points

Pour plusieurs valeurs de x successives, rangées dans l'ordre croissant, il est inutile de faire repartir la recherche de j à 0. On peut se contenter de partir de la valeur j correspondant à la valeur de x précédente.

Exercice 2.2.1 *En vous inspirant des développements §2.2.1, construire un algorithme efficace qui calcule* $\sigma(x)$ *pour tous les* x *allant de* x_{min} *à* x_{max} *par pas de* δ_x *(*x_{min} *comme* x_{max} *peut être inférieur à* x_0*, compris entre* x_0 *et* x_n*, ou supérieur à* x_n*). Cet exercice sera réalisé en TP afin de programmer les splines cubiques.*

Chapitre 3

Spline cubique naturelle d'interpolation

Étant donnés des points $(x_j, y_j)_{j=0:n}$, il s'agit dans ce chapitre de déterminer la spline cubique naturelle d'interpolation de ces points, c'est à dire telle que $\forall j \in [0:n]\ ,\ \sigma(x_j) = y_j$. On note dans ce chapitre $h_j = x_{j+1} - x_j$, $\forall j \in [0:n-1]$. On utilise l'écriture locale du chapitre précédent pour exprimer la spline. Ainsi, on recherche les 4-uplets $\left(\sigma_j, \sigma'_j, \sigma''_j, \sigma'''_j\right), \forall j \in [0:n]$ permettant d'exprimer la spline cubique naturelle d'interpolation avec la formule (2.3).

3.1 Propriétés des splines cubiques naturelles

Théorème 3.1 (Lien entre les 4-uplets) *Soit σ une spline cubique naturelle de nœuds $(x_j)_{j=0:n}$. On utilise la notation (2.2). Les vecteurs $(\sigma_j)_{j=0:n}$, $(\sigma'_j)_{j=0:n}$, $(\sigma''_j)_{j=0:n}$, $(\sigma'''_j)_{j=0:n}$ sont reliés par les équation suivantes :*

$$\forall j \in [0:n-1]\ , \qquad \sigma'''_j = \frac{(\sigma''_{j+1} - \sigma''_j)}{h_j} \tag{3.1}$$

$$\forall j \in [0:n-1]\ , \qquad \sigma'_j = \frac{\sigma_{j+1} - \sigma_j}{h_j} - \frac{h_j}{6}(\sigma''_{j+1} + 2\sigma''_j) \tag{3.2}$$

$$\begin{cases} \sigma''_0 = \sigma''_n = 0 \\ \forall j \in [1:n-1]\ , \quad \frac{h_{j-1}}{6}\,\sigma''_{j-1} + \frac{h_{j-1}+h_j}{3}\,\sigma''_j + \frac{h_j}{6}\,\sigma''_{j+1} = \frac{\sigma_{j+1}-\sigma_j}{h_j} - \frac{\sigma_j - \sigma_{j-1}}{h_{j-1}} \end{cases} \tag{3.3}$$

$$\sigma'''_n = 0 \quad \textit{et} \quad \sigma'_n = \sigma'_{n-1} + h_{n-1}\sigma''_{n-1} + \frac{h_{n-1}^2}{2}\,\sigma'''_{n-1} \tag{3.4}$$

Démonstration. La démonstration de ce théorème repose principalement sur l'écriture de la continuité de σ, σ' et σ'' en x_j. Celle-ci sera faite en cours. □

3.2 Détermination de la spline d'interpolation

Avec le théorème 3.1 en main, il devient à présent facile d'établir une méthode de détermination de la spline cubique naturelle d'interpolation. Il suffit de bien organiser les calculs. La méthode est donnée par le corollaire du théorème 3.1 ci-dessous :

Corollaire 3.1 *La spline cubique naturelle d'interpolation des points $(x_j, y_j)_{j=0:n}$ peut être déterminée par la méthode suivante :*

1. ***Détermination des** σ_j **:** le vecteur $(\sigma_j)_{j=0:n}$ est déterminé par la relation $\forall j \in [0:n]\ ,\ \sigma_j = y_j$.*
2. ***Détermination des** σ''_j **:** le vecteur $(\sigma''_j)_{j=0:n}$ est ensuite déterminé en résolvant le système linéaire donné par* (3.3) *ci-dessus.*
3. ***Détermination des** σ'''_j **:** le vecteur $(\sigma'''_j)_{j=0:n}$ est alors déterminé par la relation* (3.1) *ci-dessus (et on sait que $\sigma'''_n = 0$).*
4. ***Détermination des** σ'_j **:** le vecteur $(\sigma'_j)_{j=0:n-1}$ est finalement déterminé par la relation* (3.2) *et σ'_n s'obtient en utilisant la relation $\sigma'_{j+1} = \sigma'_j + h_j\sigma''_j + \frac{h_j^2}{2}\,\sigma'''_j$ pour $j = n-1$.*
5. *Pour x fixé, $\sigma(x)$ est alors calculé par la relation* (2.3) *(via l'algorithme §2.2.1).*

Démonstration. La démonstration est triviale à partir du théorème 3.1. □

Remarque 3.1 *Le système linéaire* (3.3) *s'écrit, sous forme matricielle, comme suit :*

$$\begin{pmatrix} \frac{h_0+h_1}{3} & \frac{h_1}{6} & 0 & & & \\ \frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & & 0 & \\ 0 & \frac{h_2}{6} & \frac{h_2+h_3}{3} & \frac{h_3}{6} & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & \frac{h_{n-3}}{6} & \frac{h_{n-3}+h_{n-2}}{3} & \frac{h_{n-2}}{6} \\ & & & & \frac{h_{n-2}}{6} & \frac{h_{n-2}+h_{n-1}}{3} \end{pmatrix} \begin{pmatrix} \sigma''_1 \\ \sigma''_2 \\ . \\ . \\ . \\ \sigma''_{n-1} \end{pmatrix} = \begin{pmatrix} \frac{\sigma_2-\sigma_1}{h_1} - \frac{\sigma_1-\sigma_0}{h_0} \\ \frac{\sigma_3-\sigma_2}{h_2} - \frac{\sigma_2-\sigma_1}{h_1} \\ . \\ . \\ . \\ \frac{\sigma_n-\sigma_{n-1}}{h_{n-1}} - \frac{\sigma_{n-1}-\sigma_{n-2}}{h_{n-2}} \end{pmatrix} \tag{3.5}$$

Remarque 3.2 *Le système linéaire* (3.3) *est un système tridiagonal, stable, et dont la résolution (voir algorithme présenté ci-dessous au §3.5) est rapide ($\mathcal{O}(n)$ opérations).*

3.3 Cas particulier où les noeuds sont équidistants

Lorsque les données sont équidistantes ($h_j = h$), le système (3.3) s'écrit plus simplement sous la forme :

$$\begin{cases} \sigma''_0 \ = \ \sigma''_n \ = \ 0 \\ \forall j \in [1:n-1]\,,\ \ \sigma''_{j-1} \ + \ 4\,\sigma''_j \ + \ \sigma''_{j+1} = \frac{6}{h^2}\left(\sigma_{j+1} \ - \ 2\,\sigma_j \ + \ \sigma_{j-1}\right) \end{cases} ; \tag{3.6}$$

ou encore, sous forme matricielle :

$$\begin{pmatrix} 4 & 1 & 0 & & & & \\ 1 & 4 & 1 & & & 0 & \\ 0 & 1 & 4 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & 0 & & & 1 & 4 & 1 \\ & & & & & 1 & 4 \end{pmatrix} \begin{pmatrix} \sigma''_1 \\ \sigma''_2 \\ . \\ . \\ . \\ \sigma''_{n-1} \end{pmatrix} = \frac{6}{h^2} \begin{pmatrix} \sigma_2 - 2\sigma_1 + \sigma_0 \\ \sigma_3 - 2\sigma_2 + \sigma_1 \\ . \\ . \\ . \\ \sigma_n - 2\sigma_{n-1} + \sigma_{n-2} \end{pmatrix} . \quad (3.7)$$

On constate que le premier membre de chacune de ces équations est 6 fois la moyenne pondérée des dérivées secondes de σ aux noeuds, tandis que le second membre est 6 fois la différence divisée seconde de σ aux noeuds, c'est à dire 6 fois l'approximation discrète de cette dérivée seconde.

Remarque 3.3 *`Python` évalue le conditionnement de la matrice pour $n = 10$, 100, et 1000 à respectivement 2.8443, 2.9981, et 3.000. Ce système linéaire est donc extrêmement stable.*

3.4 Variante

On peut aussi déterminer un système linéaire en σ'.

Exercice 3.4.1 *Dans le cas de noeuds équidistants, déterminer ce système linéaire en vous servant des formules de continuité du théorème du* (3.1)*. Comparer le alors au système linéaire ci-dessus et voir en quoi le second membre est ou non une approximation du vecteur σ'.*

3.5 Algorithme de détermination

Un système tridiagonal est particulièrement rapide à résoudre par la méthode de Gauss, de sorte que la détermination d'une spline d'interpolation est très aisée. On présente dans la suite un algorithme qui réalise cette détermination. La matrice est notée A et le second membre B.

```
% Algorithme DéterminationSplineInterpolation
% Détermine les σ_j, σ'_j, σ''_j, et σ'''_j
%   de la spline cubique d'interpolation des points (x_j , y_j)_{j=0:n}
```

1. `% Initialisation du système linéaire`

   $A \leftarrow 0$

   `Pour` $j = 0 : n-1$, $h_j \leftarrow x_{j+1} - x_j$

   `Pour` $j = 1 : n-2$, $A_{j,j+1} \leftarrow A_{j+1,j} \leftarrow h_j \,/\, 6$

   `Pour` $j = 1 : n-1$, $A_{j,j} \leftarrow (h_{j-1} + h_j) \,/\, 3$

   `Pour` $j = 1 : n-1$, $B_j \leftarrow (\sigma_{j+1} - \sigma_j)/h_j - (\sigma_j - \sigma_{j-1})/h_{j-1}$

2. % Résolution du système linéaire
   % (triangulation de la matrice A, puis résolution du système triangulaire)
   Pour j de 1 à $n-2$
   $\quad A_{j+1,j+1} \leftarrow A_{j+1,j+1} - A_{j+1,j}\, A_{j,j+1} \Big/ A_{j,j}$
   $\quad B_{j+1} \leftarrow B_{j+1} - A_{j+1,j}\, B_j \Big/ A_{j,j}$

   $\sigma''_{n-1} \leftarrow B_{n-1}/A_{n-1,n-1}$
   Pour j de $n-2$ à 1 pas de -1
   $\quad \sigma''_j \leftarrow \Big(B_j - A_{j,j+1}\, \sigma''_{j+1}\Big) \Big/ A_{j,j}$
   $\sigma''_0 \leftarrow \sigma''_n \leftarrow 0$

3. % Détermination des σ'_j, et des σ'''_j
   Pour j de 0 à $n-1$
   $\quad \sigma'_j \leftarrow (\sigma_{j+1} - \sigma_j)/(x_{j+1} - x_j) - (x_{j+1} - x_j)\,(\sigma''_{j+1} + 2\sigma''_j)/6$
   $\quad \sigma'''_j \leftarrow (\sigma''_{j+1} - \sigma''_j)/(x_{j+1} - x_j)$
   $\sigma'''_n \leftarrow 0$
   $\sigma'_n \leftarrow \sigma'_{n-1} + (x_n - x_{n-1})\,\sigma''_{n-1} + (x_n - x_{n-1})^2\,\sigma'''_{n-1}/2$

Remarque 3.4 *Pour garantir la performance d'un point de vue informatique, on s'attachera, au cours de la programmation, à utiliser un format* sparse *pour déclarer la matrice A. En effet, seuls les termes pour 3 diagonales doivent être stockées (on sait* a priori *que les autres termes sont nuls). C'est avec ce type de stockage que fonctionne n'importe quel code de calcul scientifique (industriel ou académique) lorsque celui-ci est voué à tourner sur un grand nombre de degrés de liberté.*

Exercice 3.5.1 *Donner le nombre exact d'opérations associées à la résolution du système linéaire permettant d'obtenir les σ''_j.*

Chapitre 4

Splines cubiques naturelles d'ajustement

4.1 Principe

4.1.1 Motivation

Suivant les applications, il peut parfois être plus judicieux de « lisser » les données plutôt que de les interpoler. Par lisser les données, on veut dire, de façon grossière, « passer au travers » des points. Pour commencer, on expose ci-dessous deux situations où lisser les données parait nécessaire.

Situation A :

On reprend tout d'abord le problème de PH-métrie exposé dans l'introduction. Pour celui-ci, on s'attend à rencontrer des « erreurs de mesure » non négligeables, si bien qu'il apparait indispensable de trouver une fonction qui ne passe pas par tous les points, mais au contraire qui lisse les données afin d'atténuer « au mieux » ces erreurs. Plus précisément, l'objectif est de trouver le « bon » point d'inflexion du processus sous-jacent (c'est-à-dire celui qui possède un sens physique). Si nous déterminons une courbe qui passe par tous les points, nous sommes sûrs d'avoir plusieurs points d'inflexion. Il nous faut donc trouver une courbe qui suit l'allure générale, sans pour autant passer exactement par les points.

Situation B :

Le deuxième exemple concerne la modélisation du cours de la bourse. Les courbes correspondantes sont maintenant assez bien connues du public, et spontanément, on ne s'intéresse pas, en général, aux petites oscillations, mais plutôt à l'évolution générale : croissance, décroissance, accélération,... En fait, quand on regarde une telle courbe (dans laquelle il n'y a aucune « erreur de mesure », puisque chaque point est connu exactement), on imagine une autre courbe, qui, elle, passe au travers des données : on fait donc un « lissage » des données.

Le **lissage** de données parait nécessaire lorsque l'on cherche à extraire d'une multitude données un **comportement général** : on préfère alors passer « au mieux » au travers des points $(x_j, y_j)_{j=0:n}$.

4.1.2 Méthode

Afin de réaliser le lissage de données, il nous faut considérer deux préoccupations :

- Suivre l'allure des données (c'est-à dire passer « près » des données).
- Obtenir une courbe qui oscille le moins possible.

Pour utiliser conjointement les deux critères, on déroule une méthode classique d'optimisation : on exprime chacune des deux préoccupations par une quantité positive à minimiser et on effectue la somme pondérée de ces deux quantités. La pondération, au travers d'un coefficient, permettra de donner plus ou moins d'importance à l'un ou l'autre des deux critères suivant l'application.

Critère : « Passer près des points ».

Le premier critère s'exprime au sens des moindres carrés : $E_{MC}(f) = \sum_{j=0:n}(f(x_j) - y_j)^2$. De façon plus générale, si l'on veut que la fonction passe plus près de certains points, et peut-être moins près de certains autres, nous pouvons utiliser le critère suivant :

$$E_{MC}(f) = \sum_{j=0:n} \rho_j \, (f(x_j) - y_j)^2, \tag{4.1}$$

qui fait intervenir les réels $(\rho_j)_{j=0:n}$ strictement positifs, ceux-ci représentant donc l'importance relative de chacun des points.

Critère : « Osciller le moins possible ».

Pour le deuxième critère, nous le définissons de façon analogue à celui utilisé pour définir les splines cubiques naturelles (approche variationnelle §1.5). On veut que la dérivée seconde soit, globalement, faible. On prend donc :

$$E_{flexion}(f) \;=\; \int_{\mathbb{R}} (f''(x))^2 \, \mathrm{d}x. \tag{4.2}$$

On se situe alors dans l'espace $\mathrm{D}^{-2}\mathrm{L}^2(\mathbb{R})$ pour que la quantité (4.2) ait un sens.

Compromis entre les deux critères.

Le compromis entre les deux critères est réalisé par la mise en place d'un nouveau critère E_ρ, impliquant ρ (réel strictement positif) :

Définition 4.1 (Expression de E_ρ)

$$E_\rho(f) = E_{flexion}(f) + \rho E_{MC}(f) = \int_{\mathbb{R}} (f''(x))^2 \, \mathrm{d}x + \rho \sum_{j=0:n} \rho_j (f(x_j) - y_j)^2. \tag{4.3}$$

On examine à présent l'influence du paramètre ρ à choisir par le modélisateur en fonction de la situation. Celui-ci va permettre de doser le lissage. En effet, supposons dans un premier temps que ρ soit grand : pour minimiser E_ρ, il sera nécessaire d'avoir $E_{MC}(f) = \sum_{j=0:n} \rho_j (f(x_j) - y_j)^2$

faible, quitte à « relâcher » un peu la partie $E_{flexion}(f) = \int_{\mathbb{R}} f''(x))^2 \, \mathrm{d}x$. On obtiendra alors une courbe qui passe très « près » des points, quitte à osciller davantage. Dans le cas limite $\rho \to \infty$, on va par conséquent tendre vers la spline cubique naturelle d'interpolation. Lorsqu'au contraire ρ est faible : il faudra que $E_{flexion}(f) = \int_{\mathbb{R}} f''(x))^2 \, \mathrm{d}x$ soit faible pour minimiser E_ρ, et donc que la fonction soit peu oscillante, quitte à ce que la fonction obtenue passe relativement « loin » des points. Dans le cas limite $\rho \to 0$, on va tendre vers une droite (et, plus précisément, vers la droite des moindres carrés si $\rho_j = 1, \forall j \in [0 : n]$).

Remarque 4.1 *Au bilan, les ρ_j représentent l'importance relative du point numéro j (par rapport aux autres points), alors que le paramètre ρ représente la force plus ou moins grande du lissage global. Dans la pratique, seuls les produits $\rho\,\rho_j$ comptent, et c'est pourquoi la littérature utilise souvent les seuls paramètres ρ_j. Néanmoins, il nous semble plus clair de conserver les deux paramètres dans ce cours.*

4.2 Définition et détermination de la spline d'ajustement

4.2.1 Définition de la spline d'ajustement

Nous pouvons maintenant aborder la question de l'existence, de l'unicité et de la détermination d'une fonction minimisant E_ρ. Comme pour l'interpolation, les données sont les points $(x_j, y_j)_{j=0:n}$. Aussi, nous référons à la notation (2.2) que nous allons utiliser dans la suite.

Proposition 4.1 *$\forall f \in \mathrm{D}^{-2}\mathrm{L}^2(\mathbb{R})$, soit $E_\rho(f)$ définie par* (4.3) *:*

- *Pour toute famille $(x_j, y_j, \rho\rho_j)_{j=0:n}$, il existe une et une seule fonction $\sigma_\rho \in \mathrm{D}^{-2}\mathrm{L}^2(\mathbb{R})$ minimisant E_ρ.*
- *La fonction ainsi définie est une spline cubique naturelle de noeuds $(x_j)_{j=0:n}$.*
- *Cette fonction σ_ρ vérifie de plus (en posant $\sigma'''_{\rho\,-1} = 0$) :*

$$\forall j \in [0 : n] \,, \qquad \sigma'''_{\rho\,j} - \sigma'''_{\rho\,j-1} = \rho\rho_j(y_j - \sigma_{\rho_j}). \tag{4.4}$$

Démonstration. La démonstration des points 1 et 3 concerne votre cours d'optimisation et sort donc du cadre de ce module se focalisant sur le splines. La démonstration du point 2 est quant à elle importante : elle sera faite en cours. Pour cela, on va montrer que σ_ρ qui minimise E_ρ est aussi la fonction qui minimise $E_{flexion}(g)$ sur l'ensemble des fonctions $g \in \mathrm{D}^{-2}\mathrm{L}^2(\mathbb{R})$ passant par les points (x_j, σ_{ρ_j}). □

Remarque 4.2 *L'item 2 a toute son importance d'un point de vue théorique. Il signifie qu'on ne crée pas de nouvelles fonctions pour le lissage de données.*

Remarque 4.3 *La relation* (4.4) *est d'intérêt d'un point de vue calcul : c'est elle qui fait le lien entre des valeurs de la spline d'ajustement et les données. Elle exprime le fait que la distance entre $\sigma_\rho(x_j)$ et la donnée y_j est proportionnel au saut de la dérivée troisième en x_j, le coefficient de proportionnalité étant $\rho\rho_j$.*

Définition 4.2 (Formulation variationnelle) *On appelle spline cubique naturelle d'ajustement (ou de lissage) des points $(x_j, y_j)_{j=0:n}$, affectés des poids $(\rho\rho_j)_{j=0:n}$, l'unique fonction qui réalise le minimum de $E_\rho(f) = \int_{\mathbb{R}}(f''(x))^2 \, \mathrm{d}x + \rho \sum_{j=0:n} \rho_j \, (f(x_j) - y_j)^2$ sur l'ensemble des fonctions de $\mathrm{D}^{-2}\mathrm{L}^2(\mathbb{R})$.*

En d'autres termes, la spline cubique naturelle d'ajustement des points $(x_j, y_j)_{j=0:n}$, affectés des poids $(\rho\rho_j)_{j=0:n}$ est l'unique fonction de $\mathrm{D}^{-2}\mathrm{L}^2(\mathbb{R})$ telle que :

$$\forall f \in \mathrm{D}^{-2}\mathrm{L}^2(\mathbb{R}) \, , \qquad E_\rho(\sigma_\rho) \leq E_\rho(f).$$

Remarque 4.4 (Conséquence) *La spline cubique naturelle d'ajustement des points $(x_j, y_j)_{j=0:n}$, affectés des poids $(\rho_j)_{j=0:n}$, est aussi la spline cubique naturelle d'interpolation des points $(x_j, \sigma_{\rho_j})_{j=0:n}$. Remarquons cependant qu'à ce stade on ne connaît pas encore les $(\sigma_{\rho_j})_{j=0:n}$. Le paragraphe suivant (§4.2.2) nous permettra de les déterminer, et donc de déterminer la spline cubique naturelle d'ajustement des points $(x_j, y_j)_{j=0:n}$, affectés des poids $(\rho_j)_{j=0:n}$.*

Remarque 4.5 (Analogie mécanique) *On a vu (*cf. *remarque 1.4) que la spline cubique naturelle d'interpolation des points $(x_j, y_j)_{j=0:n}$ pouvait être considérée comme un modèle mathématique d'un câble auquel on imposerait de passer par les points $(x_j, y_j)_{j=0:n}$. Avec un raisonnement similaire, on peut interpréter la spline d'ajustement comme le modèle mathématique d'un câble de rigidité $\frac{1}{\rho}$ qui serait relié aux points de données $(x_j, y_j)_{j=0:n}$ par des mini-ressorts de rigidité ρ_j. Cette analogie permet de ressentir, mécaniquement, que la spline d'ajustement va « couper les sommets » des données, et donc lisser celles-ci.*

4.2.2 Détermination de la spline d'ajustement

L'objectif à présent est de déterminer la spline σ_ρ d'ajustement. Pour cela, comme mentionné dans la remarque 4.4 ci-dessus, on va chercher la spline d'interpolation des points $(x_j, \sigma_{\rho_j})_{j=0:n}$. Il nous faut donc trouver les $(\sigma_{\rho_j})_{j=0:n}$ puis, on pourra appliquer la démarche du chapitre précédent pour calculer la spline d'interpolation correspondante (voir corollaire 3.1). Dans la suite, on oublie l'indice ρ pour simplifier les notations. La spline cubique naturelle d'ajustement que l'on recherche est noté σ et $\left(\sigma_j, \sigma'_j, \sigma''_j, \sigma'''_j\right)_{j=0:n}$ désignent les 4-uplets associés.

Cas des noeuds équidistant et $(\rho_j)_{j=0:n}$ constant.

Dans ce §, on prendra $\forall j \in [0:n] \, , \;\; \rho_j = 1$, et on notera $\sigma''_{-1} = \sigma''_{n+1} = 0$. On rappelle que σ étant une spline cubique naturelle, on a aussi $\sigma''_0 = \sigma''_n = 0$. On commence donc par rechercher les $(\sigma_j)_{j=0:n}$. En usant de (4.4) puis de (3.1), on peut écrire :

$$\sigma_j = y_j - \frac{\sigma'''_j - \sigma'''_{j-1}}{\rho} \qquad \Longrightarrow \qquad \sigma_j = y_j - \frac{\sigma''_{j+1} - 2\sigma''_j + \sigma''_{j-1}}{\rho h}. \tag{4.5}$$

Ensuite, en utilisant cette relation (4.5) pour éliminer les quantités inconnues σ_j dans chacune des équations (3.6), c'est-à-dire dans :

$$\sigma''_{j+1} + 4\sigma''_j + \sigma''_{j-1} = 6 \, \frac{\sigma_{j+1} - 2\sigma_j + \sigma_{j-1}}{h^2}, \tag{4.6}$$

on trouve, en posant $\mu = 6/\rho h^3$:

Proposition 4.2

$$\forall j \in [1 : n-1],$$
$$\mu\sigma''_{j-2} + (1-4\mu)\sigma''_{j-1} + (4+6\mu)\sigma''_j + (1-4\mu)\sigma''_{j+1} + \mu\sigma''_{j+2} = 6\tfrac{y_{j-1}-2y_j+y_{j+1}}{h^2}. \tag{4.7}$$

Le vecteur $\sigma'' = (\sigma''_j)_{j=1:n-1}$ est donc déterminé par la résolution d'un système linéaire pentadiagonal (donc en $\mathcal{O}(n)$ opérations). Le vecteur $\sigma''' = (\sigma'''_j)_{j=0:n}$ est alors déterminé à l'aide des équations (3.1). Ensuite, on calcule le vecteur $\sigma = (\sigma_j)_{j=0:n}$ avec (4.4), et enfin, le vecteur $\sigma' = (\sigma'_j)_{j=0:n}$ s'obtient à partir de (3.2) et (3.4). En conséquence, la spline σ est entièrement déterminée puisque les vecteurs σ, σ', σ'' σ''' sont connus (il suffit d'appliquer la relation (2.3) pour calculer $\sigma(x)$ en un point x quelconque).

Remarque 4.6 *Le système linéaire* (4.7) *s'écrit sous la forme :*

$$\begin{pmatrix} 4+6\mu & 1-4\mu & \mu & & & \\ 1-4\mu & 4+6\mu & 1-4\mu & \mu & & 0 \\ \mu & 1-4\mu & 4+6\mu & 1-4\mu & \mu & \\ & & \ddots & \ddots & \ddots & \\ & 0 & \mu & 1-4\mu & 4+6\mu & 1-4\mu \\ & & & \mu & 1-4\mu & 4+6\mu \end{pmatrix} \begin{pmatrix} \sigma''_1 \\ \sigma''_2 \\ . \\ . \\ . \\ \sigma''_{n-1} \end{pmatrix} = \frac{6}{h^2} \begin{pmatrix} y_2 - 2y_1 + y_0 \\ y_3 - 2y_2 + y_1 \\ . \\ . \\ . \\ y_n - 2y_{n-1} + y_{n-2} \end{pmatrix} \tag{4.8}$$

Cas où les noeuds sont non équidistants et/ou les ρ_i quelconques.

On obtient de la même manière :

Proposition 4.3

$$\forall j \in [1 : n-1]\,, \qquad \alpha_j\,\sigma''_{j-2} + \beta_j\,\sigma''_{j-1} + \gamma_j\,\sigma''_j + \delta_j\,\sigma''_{j+1} + \varepsilon_j\,\sigma''_{j+2} = \zeta_j, \tag{4.9}$$

avec :

$$\begin{aligned}
\zeta_j &= 6\left(\frac{y_{j+1}-y_j}{h_j} - \frac{y_j - y_{j-1}}{h_{j-1}}\right) ; \\
\alpha_j &= \frac{6}{\rho\rho_{j-1}h_{j-2}h_{j-1}} \qquad ; \qquad \varepsilon_j = \frac{6}{\rho\rho_{j+1}h_jh_{j+1}} ; \\
\beta_j &= h_{j-1} - 6\,\frac{h_{j-1}+h_j}{\rho\rho_j h_{j-1}^2 h_j} - 6\,\frac{h_{j-2}+h_{j-1}}{\rho\rho_{j-1}h_{j-2}h_{j-1}^2} \;\; ; \;\; \delta_j = h_j - 6\,\frac{h_j+h_{j+1}}{\rho\rho_{j+1}h_j^2h_{j+1}} - 6\,\frac{h_{j-1}+h_j}{\rho\rho_j h_{j-1}h_j^2} ; \\
\gamma_j &= 2(h_{j-1}+h_j) + \frac{6}{\rho\rho_{j+1}h_j^2} + \frac{6}{\rho\rho_{j-1}h_{j-1}^2} + 6\,\frac{(h_{j-1}+h_j)^2}{\rho\rho_j h_{j-1}^2 h_j^2}.
\end{aligned}$$

4.2.3 Algorithme de détermination

L'algorithme de détermination de la spline d'ajustement σ_ρ est donc simple et s'inspire directement de celui de la détermination de la spline d'interpolation (système pentadiagonal à résoudre au lieu d'un système tridiagonal).

1. *Les notations α_j, β_j, γ_j, δ_j, ε_j et ζ_j sont des abrégés pour les quantités définies au § précédent ; le cas ρ constant et noeuds équidistants est indiqué entre parenthèses.*

$A \leftarrow 0$
Pour j de 3 à $n-1$, $A(j,j-2) \leftarrow \alpha_j$ (μ)
Pour j de 2 à $n-1$, $A(j,j-1) \leftarrow \beta_j$ $(1-4\mu)$
Pour j de 1 à $n-1$, $A(j,j) \leftarrow \gamma_j$ $(4+6\mu)$
Pour j de 1 à $n-2$, $A(j,j+1) \leftarrow \delta_j$ $(1-4\mu)$
Pour j de 1 à $n-3$, $A(j,j+2) \leftarrow \varepsilon_j$ (μ)
Pour j de 1 à $n-1$, $B(j) \leftarrow \zeta_j$ $(\,6\,(y_{i+1}-2y_i+y_{i-1})\,/\,h^2\,)$

2. *Certains peuvent préférer écrire explicitement les boucles* `Pour` *contenant* $\min(i{+}2, n{-}1)$ *(quatre instructions ... mais attention alors à gérer proprement le cas $i = n-2$)*

```
Pour i de 1 à n−2
    ‖  Pour j de i+1 à min(i+2, n−1)
    ‖      ‖  Pour k de i+1 à min(i+2, n−1)
    ‖      ‖      ‖  A(j,k) ← A(j,k) − A(j,i) A(i,k)/A(i,i)
    ‖      ‖  B(j) ← B(j) − A(j,i) B(i)/A(i,i)
σ''_{n−1} ← B(n−1)/A(n−1,n−1)
σ''_{n−2} ← ( B(n−2) − A(n−2,n−1) σ''_{n−1} ) / A(n−2,n−2)
Pour i de n−3 à 1 par pas de −1
    ‖  σ''_i ← ( B(i) − A(i,i+2) σ''_{i+2} − A(i,i+1) σ''_{i+1} ) / A(i,i)
σ''_0 ← 0   ;   σ''_n ← 0
```

3. $\sigma_0 \leftarrow y_0 - \sigma''_1/(h_0\rho\rho_0)$ $\quad (y_0 - \sigma''_1\,\mu\,\frac{h^2}{6})$
$\sigma_n \leftarrow y_n - \sigma''_{n-1}/(h_{n-1}\rho\rho_n)$ $\quad (y_n - \sigma''_{n-1}\,\mu\,\frac{h^2}{6})$

```
Pour i de 1 à n−1
    ‖  σ_i ← y_i − (σ''_{i+1} − σ''_i)/(h_i ρ ρ_i) + (σ''_i − σ''_{i−1})/(h_{i−1} ρ ρ_i)
    ‖                  (y_i − (σ''_{i+1} − 2σ''_i + σ''_{i−1}) μ h²/6 )
```

Le reste est identique au cas de l'interpolation. Remarquons que dans le cas où ρ n'est pas constant et/ou les noeuds ne sont pas équidistants, seules changent les formules dans 1 et 3. Comme pour l'interpolation, sur le plan informatique, il serait maladroit de stocker toute la matrice A, de dimension $(n{-}1, n{-}1)$; il vaut mieux utiliser un format $sparse$ et stocker uniquement les 5 diagonales où les termes sont non nuls.

Deuxième partie

B-Splines, courbes B-Splines, outils pour la CAO

Chapitre 5

B-splines à nœuds équidistants

On s'intéresse à présent à l'utilisation des notions abordées dans la première partie du cours afin de **représenter** (on dit aussi « modéliser ») n'importe quelle **forme géométrique**. Pour cela, on commence par introduire, dans ce chapitre, les fonctions **B-splines cubiques** qui forment une très bonne base de l'espace des splines. Celles-ci sont à la base des outils utilisés à l'heure actuelle en **Conception Assistée par Ordinateur** (CAO). On apercevra alors le potentiel de cette famille de fonctions au travers de la génération de **courbes B-splines**. Pour simplifier, on va se placer ici dans le cadre de nœuds équidistants, c'est-à-dire tels que $h = h_j = x_{j+1} - x_j$, $\forall j \in [0 : n-1]$ (ou encore $x_j = x_0 + jh$, $\forall j \in [0 : n]$).

5.1 B-splines cubiques

5.1.1 Construction

Les splines cubiques naturelles de nœuds fixés $(x_j)_{j=0:n}$ formant un espace vectoriel, l'objectif dans cette section est de construire une base « pratique » de cet espace. Afin de simplifier les calculs, il faut que les éléments de cette base aient un support aussi petit que possible. On cherche ainsi une spline cubique naturelle à support borné, c'est-à-dire nulle en dehors d'un intervalle borné. Pour cela, on peut remarquer qu'il suffit de prendre (en plus d'avoir σ'' nulle aux extrémités), σ et σ' nulles aux extrémités. Il reste alors à ajouter $\sigma(x_j) = \alpha$ (avec $\alpha \in \mathbb{R}^*$) en un certain $j \in [1 : n-1]$ de sorte d'avoir σ non identiquement nulle. Nous nous retrouvons donc avec au minimum 5 conditions d'« interpolation » : nous avons donc besoin de 5 nœuds pour définir l'élément de base. On prend alors, tout simplement, les 5 nœuds -2, -1, 0, 1 et 2.

Proposition 5.1 *Si σ est une spline cubique naturelle de nœuds -2, -1, 0, 1, 2 et telle que $\sigma(-2) = \sigma'(-2) = \sigma'(2) = \sigma(2) = 0$ et $\sigma(0) = \alpha$, alors σ est la spline cubique naturelle d'interpolation des points $(-2; 0)$, $(-1; \frac{\alpha}{4})$, $(0; \alpha)$, $(1; \frac{\alpha}{4})$, $(2; 0)$.*

La fonction de base appelée **B-spline** est alors tout simplement cette fonction avec $\alpha = 2/3$.

Démonstration. La démonstration des différents points de ce paragraphe est aisée et est ainsi laissé au lecteur. On peut, par exemple, procéder avec les exercices ci-dessous. Des éléments de

corrections seront donnés en cours. □

Exercice 5.1.1 *Montrer que l'ensemble des splines cubiques naturelles de nœuds $(x_j)_{j=0:n}$ constitue bien un espace vectoriel.*

Exercice 5.1.2 *Soit σ la spline cubique naturelle de nœuds $(x_j)_{j=0:n}$ et telle que $\sigma(x_0) = \sigma'(x_0) = \sigma'(x_n) = \sigma(x_n) = 0$, montrer que σ est à support borné. Montrer alors que si $\sigma(x_j) \neq 0$ en un certain $j \in [1 : n-1]$, elle n'est pas identiquement nulle.*

Exercice 5.1.3 *Montrer enfin la proposition 5.1 et en déduire que la B-spline est la seule spline cubique naturelle de support $[-2\,..\,2]$ de nœuds -2, -1, 0, 1, 2 et telle que $\sigma(-2) + \sigma(-1) + \sigma(0) + \sigma(1) + \sigma(2) = 1$.*

5.1.2 Définition et expression

Définition 5.1 *On appelle **B-spline cubique** (on note B) la spline cubique naturelle d'interpolation des points $(-2;0), (-1;\frac{1}{6}), (0;\frac{2}{3}), (1;\frac{1}{6}), (2;0)$.*

La B-spline cubique naturelle s'exprime donc comme suit (écriture locale) :

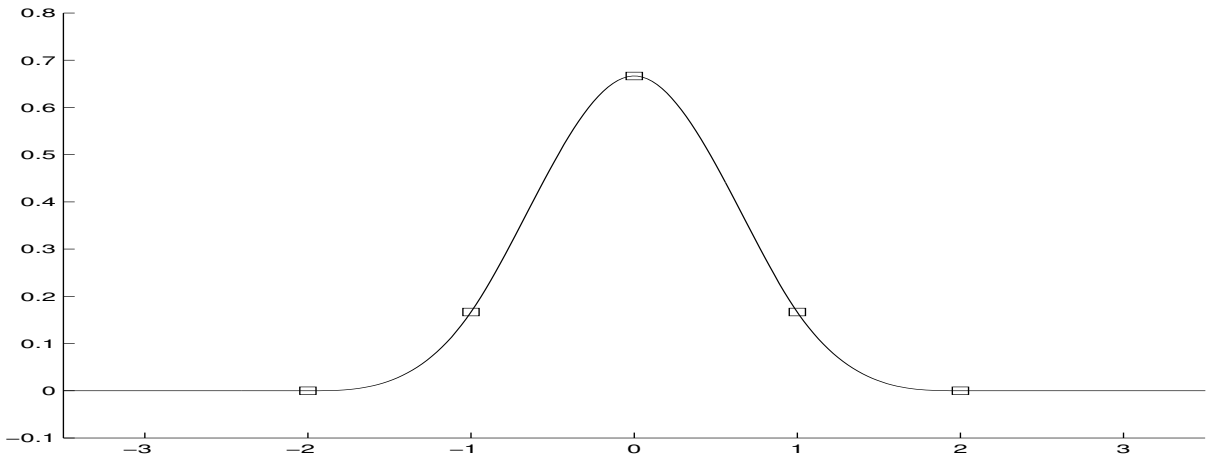
$$\begin{array}{lll}
\forall x \in [-2\,..\,-1]\,, & B(x) = (x+2)^3/6 & = (x^3+6x^2+12x+8)/6 \\
\forall x \in [-1\,..\,0]\,, & B(x) = (x+2)^3/6 - 4(x+1)^3/6 & = (-3x^3-6x^2+4)/6 \\
\forall x \in [0\,..\,1]\,, & B(x) = (x+2)^3/6 - 4(x+1)^3/6 + x^3 & = (3x^3-6x^2+4)/6 \\
\forall x \in [1\,..\,2]\,, & B(x) = (x+2)^3/6 - 4(x+1)^3/6 + x^3 - 4(x-1)^3/6 & = (2-x)^3/6
\end{array} \tag{5.1}$$

d'où les valeurs ponctuelles suivantes :

x	-2	-1	0	1	2
$B(x)$	0	$1/6$	$2/3$	$1/6$	0
$B'(x)$	0	$1/2$	0	$-1/2$	0
$B''(x)$	0	1	-2	1	0
$B'''(x+)$	1	-3	3	-1	0

(5.2)

et son graphe :



Proposition 5.2 *On peut noter les quelques propriétés suivantes pour la B-spline cubique :*

- ***Support borné et positivité :***

$$\forall x \in]-\infty\,..\,2] \cup [2\,..\,+\infty[,\ B(x) = 0 \quad \text{et} \quad \forall x \in]-2\,..\,2[, B(x) > 0.$$

- ***Partition de l'unité aux nœuds :***

$$\sum_{i\in\mathbb{Z}} B(i) = \sum_{i\in[-1:1]} B(i) = 1.$$

- ***Partition de l'unité sur $\mathbb{R}$:***

$$\forall x\in\mathbb{R}\,, \quad \sum_{i\in\mathbb{Z}} B(x-i) = \sum_{i\in[\lfloor x\rfloor-1:\lfloor x\rfloor+2]} B(x-i) = 1.$$

- ***Interpolation sur $\mathbb{Z}$:***
  B est aussi la spline d'interpolation des points $\cdots 0, 0, 0, \frac{1}{6}, \frac{2}{3}, \frac{1}{6}, 0, 0, 0, \cdots$.
- ***Intégrale et énergie de flexion :***

$$\int_{\mathbb{R}} B(x)\mathrm{d}x = \int_{-2}^{2} B(x)\mathrm{d}x = 1 \quad \text{et} \quad \int_{\mathbb{R}} (B''(x))^2\mathrm{d}x = \int_{-2}^{2} (B''(x))^2\mathrm{d}x = \frac{8}{3}.$$

Démonstration. La démonstration de ces propriétés est aisée à partir de la formule analytique de la B-spline cubique Eq. (5.1). □

Remarque 5.1 *On comprend à présent mieux l'intérêt d'avoir pris $\alpha = 2/3$ dans la formule générale de la proposition 5.1 : c'est pour garantir la partition de l'unité. On verra dans la suite que cette propriété est importante en pratique.*

5.2 Approximations et courbes B-splines

5.2.1 Préliminaire : approximation affine par morceaux

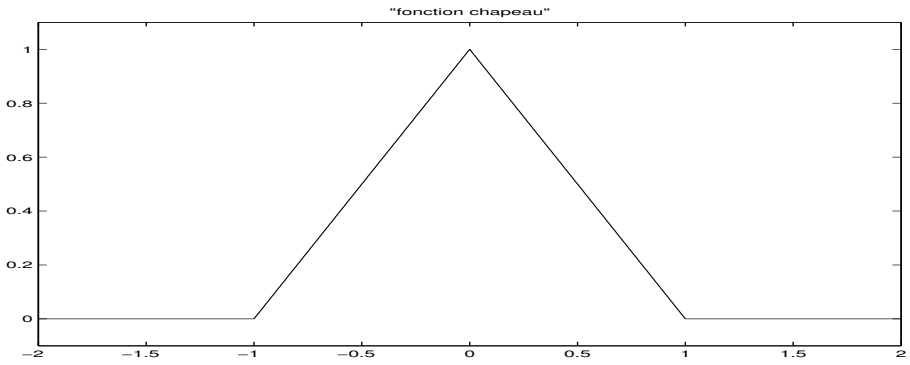
Avant de se focaliser sur les approximations et courbes que l'on peut générer à partir des B-splines cubiques introduites dans la section précédente, on commence, pour plus de clarté, à fixer les idées sur le cas plus simple de fonctions affines par morceaux et de régularité $\mathcal{C}^0$ aux nœuds. On peut alors parler de « spline linéaire » de nœuds $(x_j)_{j=0:n}$ et de « B-spline linéaire » concernant les éléments de base.

En reprenant la même démarche que dans la section précédente pour construire une base « pratique » de l'espace des splines linéaires, on aboutit à la fonction classique « chapeau » pour la B-spline linéaire, c'est-à-dire au polynôme caractéristique de Lagrange. On notera donc cette fonction de base L. Celle-ci est bien évidemment affine par morceaux et $\mathcal{C}^0$. Elle est de plus

nulle en dehors de l'intervalle $]-1\,..\,1[$, et telle que $L(0)=1$ (afin de respecter les propriétés de partition de l'unité). Elle s'exprime tout simplement comme suit :

$$\begin{array}{ll} \forall x \in [-1\,..\,0]\,, & L(x) = 1+x \\ \forall x \in [0\,..\,1]\,, & L(x) = 1-x \end{array}$$

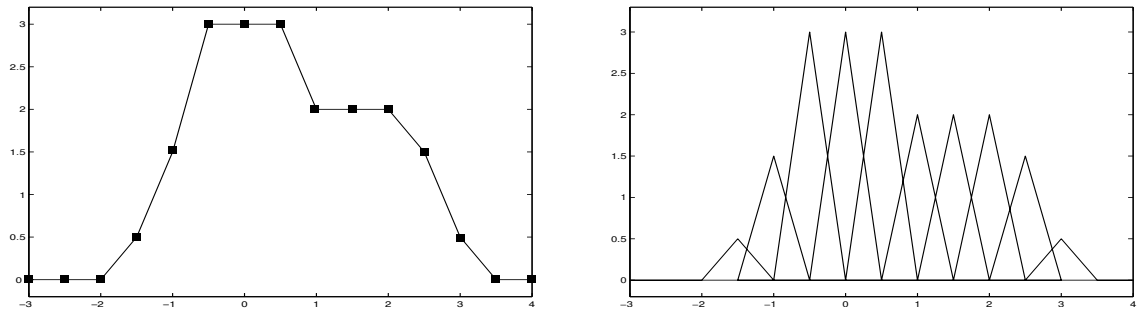
et son graphe est le suivant :



Pour des données $(x_0 + i\,h\,, y_i)_{i=0:n}$, on peut alors remarquer que l'approximation affine par morceaux notée τ dans la suite peut se définir comme suit :

$$\forall x \in [x_0\,..\,x_n]\,, \qquad \tau(x) = \sum_{i=0:n} y_i\,L_i(x) \quad \text{avec} \quad L_i(x) = L((x-x_i)/h). \tag{5.3}$$

L_i est la fonction chapeau centrée sur x_i et de support $[x_{i-1}\,..\,x_{i+1}]$. Par exemple, on obtient le graphe de gauche ci-dessous avec x=[-3 :.5 :4] et y= [0 0 0 .5 1.5 3 3 3 2 2 2 1.5 .5 0 0] . Le graphe de droite présente quant à lui les différents termes $y_i\,L_i(x)$ générant, par somme, la figure de gauche.



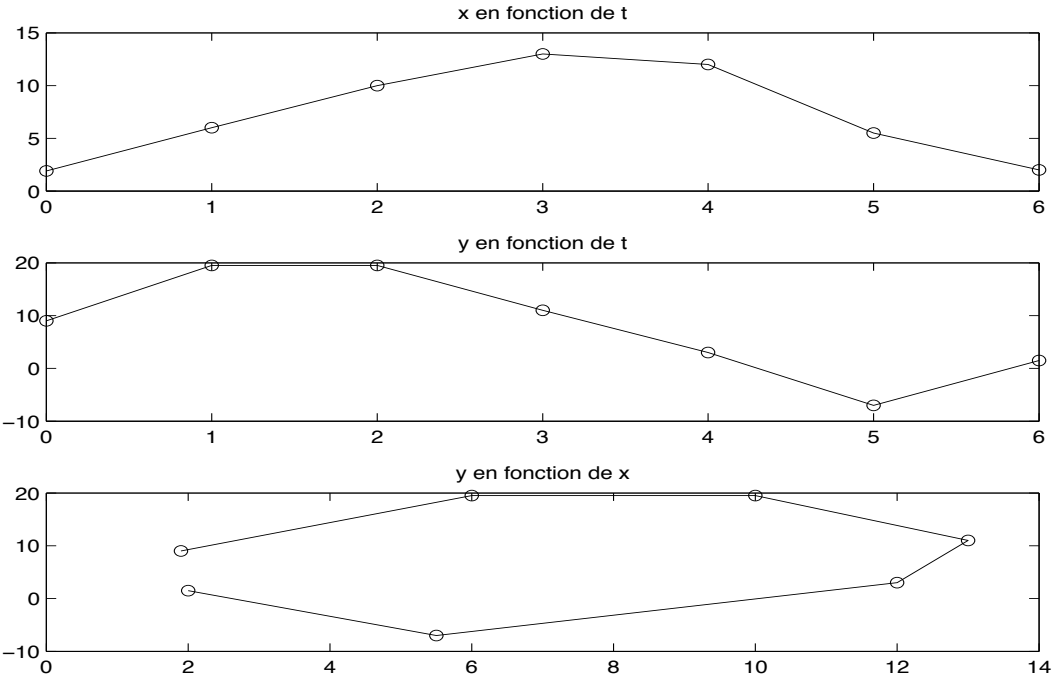
Exercice 5.2.1 *À titre d'exercice, montrer que la forme* (5.3) *permet bien de générer l'approximation affine des données* $(x_0 + i\,h\,, y_i)_{i=0:n}$.

Remarque 5.2 *Ce concept est à la base de la méthode d'approximation d'EDP par éléments finis que vous verrez l'an prochain. L'idée sera alors de construire un espace d'approximation de la solution de l'EDP avec de telles fonctions chapeaux. On pourra encore aller plus loin en utilisant des B-splines de plus hauts degrés (telles que cubiques) pour résoudre une EDP : on parlera alors d'analyse isogéométrique.*

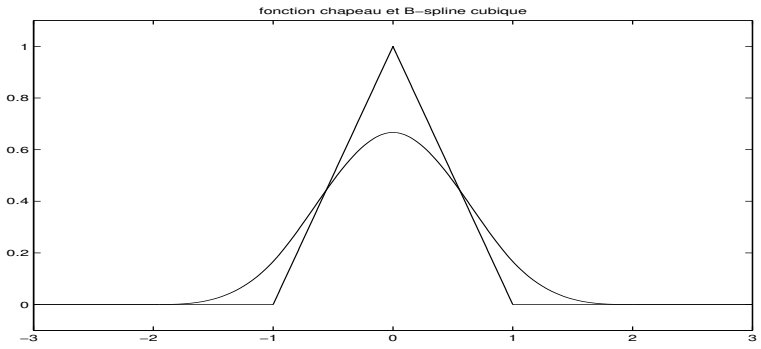
De la même façon, on peut aussi facilement générer une courbe affine par morceaux passant par les $n+1$ points $(X_i, Y_i)_{i=0:n}$ de $\mathbb{R}^2$. Pour cela, il nous faut travailler en paramétrique. On prend un paramètre $t \in [0\,..\,n]$ et on définit la fonction paramétrique f telle que :

$$\forall t \in [0\,..\,n]\,, \qquad f(t) \;=\; \begin{cases} x(t) = \displaystyle\sum_{i=0:n} X_i\, L(t-i) \\ y(t) = \displaystyle\sum_{i=0:n} Y_i\, L(t-i) \end{cases} .$$

Il est clair que f interpole les points $(X_i, Y_i)_{i=0:n}$ (car $f(i) = (X_i, Y_i)$). De plus, f est affine sur chaque composante pour $t \in [i\,..\,i+1]$. Par conséquent, la courbe représentative de f est bien le polygone reliant les points $(X_i, Y_i)_{i=0:n}$, c'est-à-dire la courbe affine par morceaux passant par $(X_i, Y_i)_{i=0:n}$. On montre à titre d'exemple les graphes obtenus pour X=[1.9, 6, 10, 13, 12, 5.5, 2] et Y=[9, 19.5, 19.5, 11, 3, -7, 1.5] ci-dessous :



La démarche pour générer les approximations et courbes B-splines cubiques devient dès lors intuitive : on va faire de même que pour le cas affine par morceaux, c'est-à-dire que l'on va effectuer une combinaison linéaire des données et des B-splines cubiques. La figure ci-dessous représente sur un même graphe la fonction chapeau et la B-spline cubique. On s'aperçoit que la B-spline cubique est une sorte de fonction chapeau « arrondie » (ou « régularisée » puisque la fonction chapeau est $\mathcal{C}^0$ alors que la B-spline est $\mathcal{C}^2$). On va donc retrouver, en utilisant les B-splines cubiques, une courbe « régulière » (« lisse ») qui passe près des points $(X_i, Y_i)_{i=0:n}$.



5.2.2 Définitions

De même que pour la fonction L_i, pour un pas h donné, on notera dans la suite B_i la B-spline de pas h centrée en x_i, c'est à dire la fonction définie par $\forall x \in \mathbb{R}\ ,\ \ B_i(x) = B((x - x_i)/h)$.

Définition 5.2 (Approximation B-spline) *Soient les données $(x_i = x_0 + i\,h\,, y_i)_{i=0:n}$. On appelle **approximation B-spline de** $(x_i, y_i)_{i=0:n}$ la spline cubique naturelle σ définie comme suit :*

- *Avec les points supplémentaires (« points fantômes ») :*

$$y_{-1} = 2\,y_0 - y_1 \quad ; \quad y_{n+1} = 2\,y_n - y_{n-1} \quad ; \quad y_{n+2} = 2\,y_{n+1} - y_n,$$

- *σ est telle que :*

$$\left\{ \begin{array}{lcl} \forall x \in [x_0\,..\,x_n]\ ,\ \ \sigma(x) & = & \displaystyle\sum_{i \in [-1:n+1]} y_i B_i(x) \\ \forall x \in]-\infty\,..\,x_0]\ ,\ \ \sigma(x) & = & y_0 + (x - x_0)\,\dfrac{y_1 - y_0}{x_1 - x_0} \\ \forall x \in [x_n\,..\,+\infty[\ ,\ \ \sigma(x) & = & y_n + (x - x_n)\,\dfrac{y_n - y_{n-1}}{x_n - x_{n-1}} \end{array} \right. . \qquad (5.4)$$

Remarque 5.3 *L'introduction de points fantômes est nécessaire ici du fait que les $(B_i)_{i=0:n}$ ne forment pas une base de l'espace vectoriel des splines cubiques naturelles de nœuds $(x_i)_{i=0:n}$. En effet, la spline $\sum_{i \in [0:n]} y_i B_i(x)$ possède des nœuds en x_{-1} et x_{n+1}. Une façon de résoudre le problème serait de prendre des B-splines à nœuds non-équidistants et d'utiliser des points quadruples aux bords. Ici, on contourne le problème en définissant des points supplémentaires.*

Remarque 5.4 *On peut remarquer qu'avec la définition du point supplémentaire (x_{-1}, y_{-1}) (respectivement (x_{n+1}, y_{n+1})) l'approximation B-spline passe par le point (x_0, y_0) (resp. (x_n, y_n)) et elle est tangente en ce point au segment reliant (x_0, y_0) et (x_1, y_1) (resp. (x_{n-1}, y_{n-1}) et (x_n, y_n)).*

Définition 5.3 (Courbe B-spline) *Soient $n+1$ points $(P_i)_{i=0:n} = (X_i, Y_i)_{i=0:n}$ de $\mathbb{R}^2$. On appelle* ***courbe B-spline des points*** *$(P_i)_{i=0:n}$ la courbe paramétrique C définie par :*

$$\forall t \in [0\,..\,n]\,, \qquad C(t) = \begin{cases} \sigma_x(t) &= \displaystyle\sum_{i\in[-1:n+1]} X_i\, B(t-i) \\ \sigma_y(t) &= \displaystyle\sum_{i\in[-1:n+1]} Y_i\, B(t-i) \end{cases} .$$

Le polygone reliant les points $(P_i)_{i=0:n}$ est appelé polygone de contrôle de la courbe B-spline.

Remarque 5.5 *Il est à noter que σ_x et σ_y sont tout simplement les approximations B-splines des données $(i, X_i)_{i=0:n}$ et $(i, Y_i)_{i=0:n}$.*

5.2.3 Valeurs aux nœuds

À partir de la forme analytique (5.4) et des valeurs données dans le tableau (5.2), il est aisé de déterminer directement les 4-uplets $(\sigma_j, \sigma'_j, \sigma''_j, \sigma'''_j)_{j=0:n}$ associés à l'approximation B-spline.

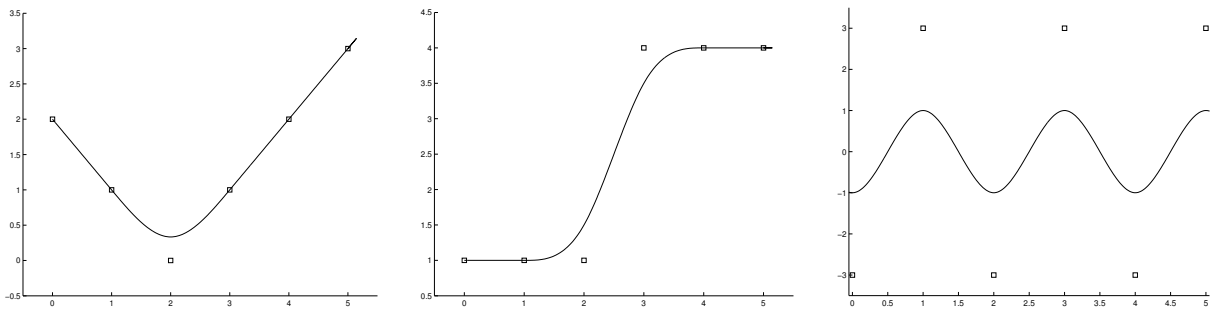
Proposition 5.3 *L'approximation B-spline admet les valeurs suivantes :*

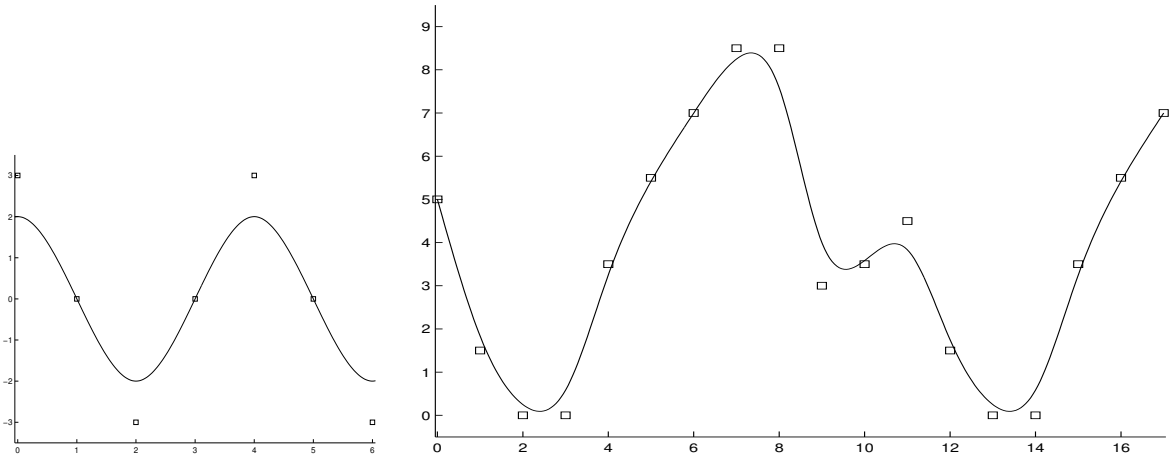
$$\forall j \in [0:n], \quad \begin{cases} \sigma_j &= (y_{j+1} + 4y_j + y_{j-1})/6 \\ \sigma'_j &= (y_{j+1} - y_{j-1})/2h \\ \sigma''_j &= (y_{j+1} - 2y_j + y_{j-1})/h^2 \\ \sigma'''_j &= (y_{j+2} - 3y_{j+1} + 3y_j - y_{j-1})/h^3 \end{cases} . \tag{5.5}$$

Démonstration. La démonstration s'appuie sur le tableau des valeurs de B aux nœuds (*cf.* Eq. (5.2)) et sur la définition 5.2. □

5.2.4 Quelques approximations B-splines

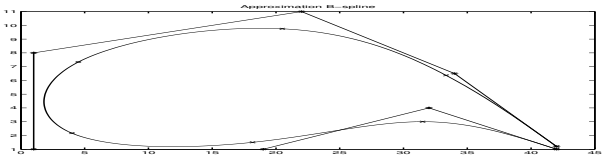
Les $(x_j\,, y_j)$ sont les petits carrés. On remarque que les courbes ne passent pas, en général par les données, excepté lorsque plusieurs données sont alignées.



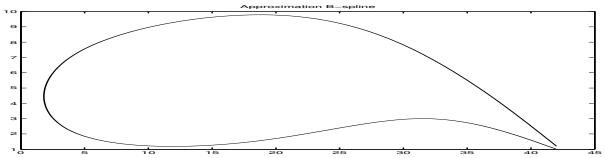


5.2.5 Quelques courbes B-splines

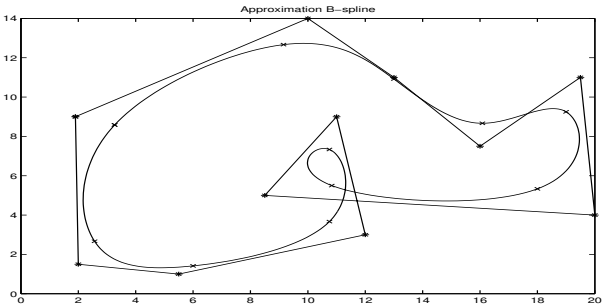
Les courbes ci-après seront réalisées en TP.



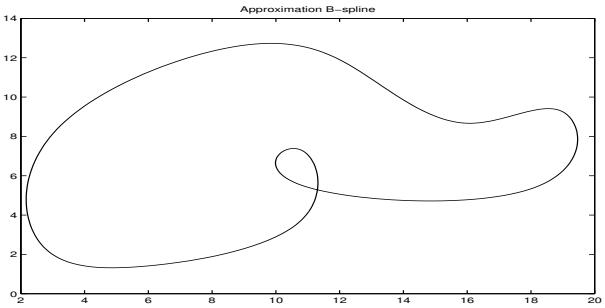
(avec le polygone de contrôle)



(sans le polygone de contrôle)



(avec le polygone de contrôle)



(sans le polygone de contrôle)

5.2.6 Quelques propriétés

Proposition 5.4 *On peut noter tout d'abord les quelques propriétés suivantes pour l'approximation B-spline σ des données $(x_j, y_j)_{j=0:n}$:*

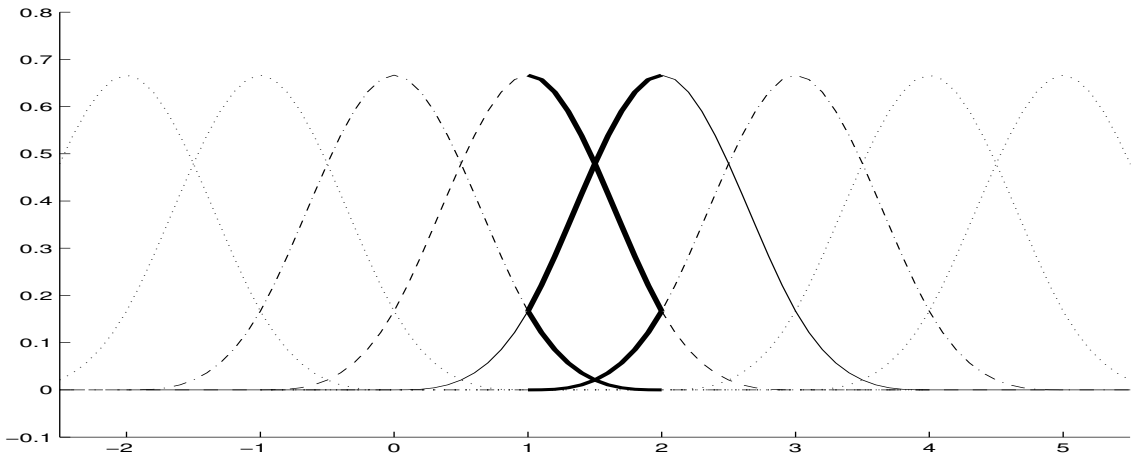
- *σ est une **spline cubique naturelle** de nœuds $(x_j)_{j=0:n}$.*
- *$\mathbb{P}_1$ **reproduction** :*

$$\forall p \in \mathbb{P}_1 \quad , \Big(\forall j \in [0:n] \, , \qquad y_j = p(x_j) \Big) \; \Rightarrow \sigma = p.$$

(Rappel : $\mathbb{P}_k$ désigne l'ensemble des polynômes de degré au plus k.)

- ***Approximation locale** : Comme illustré par la figure suivante, seules 4 B-splines sont non nulles dans l'intervalle $[x_j \,..\, x_{j+1}]$. Ce sont les B-splines centrées en x_{j-1}, x_j, x_{j+1}, x_{j+2}.*

Démonstration. Seul le deuxième point n'est pas trivial. Il est pourtant important puisqu'il signifie qu'on peut représenter correctement des formes géométriques rectilignes. La démonstration de celui-ci est basée sur l'utilisation de la partition de l'unité. Elle sera faite en cours. □

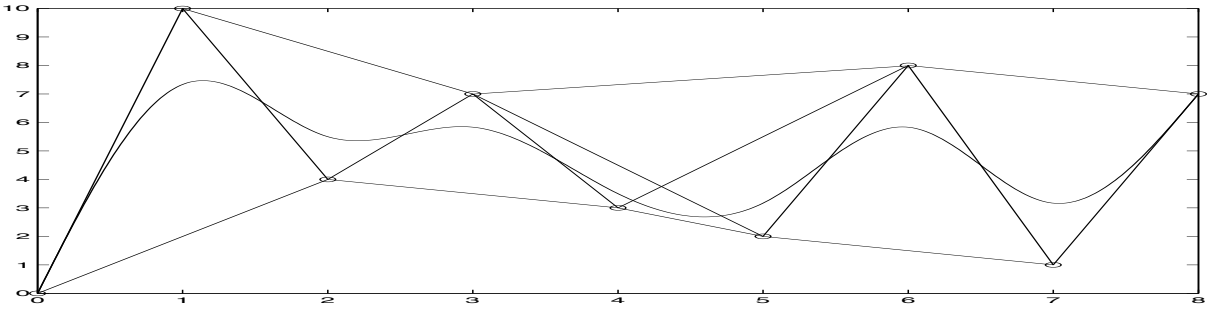


En conséquence du dernier point, dans l'intervalle $[x_j \,..\, x_{j+1}]$, σ s'écrit encore :

$$\sigma(x) = \sum_{i=j-1:j+2} y_i \, B_i(x),$$

de sorte que $\forall x \in [x_j \,..\, x_{j+1}], \sigma(x)$ ne dépend que de $y_{j-1}, y_j, y_{j+1}, y_{j+2}$, et peut-être considérée comme une moyenne pondérée de $y_{j-1}, y_j, y_{j+1}, y_{j+2}$ (puisque les B_i respectent la partition de l'unité). De plus, comme les coefficients $B_i(x)$ pour $i \in [j-1 : j+2]$ de cette moyenne sont positifs, $\sigma(x)$ appartient à l'enveloppe convexe [1] des points $(x_i, y_i)_{i=j-1:j+2}$. Une illustration est donnée dans la figure suivante. σ est à l'intérieur d'une « enveloppe convexe glissante » : σ « suit l'allure » des données.

1. On appelle enveloppe convexe d'une famille finie de points $(P_j)_{j=1:k}$ l'ensemble des points situés à l'intérieur du plus petit polygone convexe reliant les points $(P_j)_{j=1:k}$.



Proposition 5.5 *Les deux propriétés suivantes doivent donc être ajoutées :*

- $\forall x \in [x_j \,..\, x_{j+1}]$,
  $\sigma(x)$ *appartient à* ***l'enveloppe convexe*** *des points* $(x_i, y_i)_{i=j-1:j+2}$.
- ***Reproduction*** $\mathbb{P}_1$ ***local*** *: si les points* $(x_i, y_i)_{i=j-1:j+2}$ *sont alignés, alors sur* $[x_j \,..\, x_{j+1}]$ σ *est le segment de droite reliant* (x_j, y_j) *à* (x_{j+1}, y_{j+1}).

5.2.7 Algorithme de calcul

Pour être optimal d'un point de vue efficacité, on se base sur les formules explicites (5.5) pour déterminer, numériquement, une approximation B-spline. Il ne reste alors plus qu'à évaluer la spline en certains points à l'aide la routine d'évaluation classique vue dans la première partie. Au bilan, on peut procéder comme suit :

$y_{-1} \leftarrow 2\,y_0 - y_1\,;\, y_{n+1} \leftarrow 2\,y_n - y_{n-1}\,;\, y_{n+2} \leftarrow 2\,y_{n+1} - y_n$

```
Pour j de 0 à n
|| x_j ← x_0 + jh  ;  σ_j ← (y_{j+1} + 4y_j + y_{j-1})/6
|| σ'_j ← (y_{j+1} − y_{j-1})/2h
|| σ''_j ← (y_{j+1} − 2y_j + y_{j-1})/h²
|| σ'''_j ← (y_{j+2} − 3y_{j+1} + 3y_j − y_{j-1})/h³
```

$A \leftarrow 0, B \leftarrow 0, C \leftarrow \sigma'_0, D \leftarrow \sigma_0, j \leftarrow -1$

```
Pour x de a à b par pas de Δx:
|| Tant que x ≥ x_{j+1} et j ≤ n
||     || j ← j + 1
||     || A ← σ'''_j/6  ;  B ← σ''_j/2
||     || C ← σ'_j  ;  D ← σ_j
|| xx ← x − x_{max(j,0)}
|| S ← D + xx (C + xx (B + xx A))
|| la valeur de σ(x) est dans S
```

Remarque 5.6 *Il est important de noter qu'aucune inversion de système linéaire n'est nécessaire pour la détermination de l'approximation B-spline, ce qui rend la procédure extrêmement efficace.*

Chapitre 6

Spline des moindres carrés

En terme d'utilisation, les B-splines cubiques introduites dans le chapitre précédent offrent aussi la possibilité de filtrer des données bruitées. On peut en effet utiliser cette base pour calculer efficacement une **spline des moindres carrés**.

6.1 Principe : fonction de moindres carrés dans un espace vectoriel

Le problème que l'on souhaite résoudre dans ce chapitre est très général si bien qu'il peut être rencontré dans de nombreux contextes autres que celui des fonctions splines. Il se traduit comme suit. Étant données $N+1$ mesures $(x_i, y_i)_{i=0:N}$ (les données), on désire faire passer une fonction f (c'est-à-dire un certain modèle mathématique), non pas par les données mais près de celles-ci. On désire de plus que la fonction appartienne à un espace vectoriel de dimension $n+1$ (avec $n \leq N$), dont on connait une base (que nous noterons ici $(C_j)_{j=0:n}$). On désire enfin que la fonction représente, au mieux, les mesures.

Pour ce faire, le principe est d'avoir recours à une formulation vue en optimisation du type **problème de moindres carrés**. On définit une mesure « simple » de la distance entre les données et la fonction, et on cherche la fonction de la forme $f = \sum_{j=0}^{n} a_j\, C_j$ qui minimise cette distance. D'un point de vue pratique, le caractère « simple » pour la mesure est introduit ici pour caractériser une distance qui se dérive facilement de façon à pouvoir en chercher le minimum aisément. On prend alors habituellement la norme L_2.

On aboutit ainsi au problème des moindres carrés suivants : on cherche $a^* \in \mathbb{R}^{n+1}$ (avec $a^* = (a_j^*)_{j=0:n}$) tel que :

$$a^* = \underset{a\in\mathbb{R}^n}{\arg\min}\, E(a) \quad \text{avec} \quad E(a) \;=\; \sum_{i=0}^{N} \left(\Big(\sum_{j=0}^{n} a_j C_j(x_i) \Big) - y_i \right)^2 . \tag{6.1}$$

Exercice 6.1.1 *Montrer que le problème* (6.1) *est un problème de moindres carrés linéaires, c'est-à-dire que la fonctionnelle peut s'écrire* $E(a) = \|F(a)\|_2^2$ *avec* $F(a) = Ua - Y$ *où* U *(matrice* $(N+1)\times(n+1)$*) et* $Y \in \mathbb{R}^{N+1}$ *sont à spécifier.*

Exercice 6.1.2 *À l'aide de son cours d'optimisation, montrer qu'il y a bien existence et unicité de la solution au problème* (6.1) *et que celle-ci s'obtient en résolvant le système linéaire suivant :*

$$\forall k \in [0:n]\ , \qquad \frac{\partial E(a)}{\partial a_k} = 0. \tag{6.2}$$

6.2 Application aux splines

6.2.1 Position du problème

Nous allons à présent appliquer la démarche de la section précédente à la situation où l'espace vectoriel est l'espace des splines cubiques naturelles. En accord avec la définition 5.2, il nous faut tout d'abord choisir les nœuds $(X_j)_{j=0:n}$ de l'approximation B-spline σ puis, minimiser la quantité :

$$E \ = \ \sum_{i=0}^{N} \rho_i (\sigma(x_i) - y_i)^2.$$

Les réels $(\rho_i)_{i=0:N}$ strictement positifs sont introduits pour pondérer l'importance relative de chacun des points.

Pour réaliser la minimisation, il faut prendre quelques précautions pour gérer proprement les bords. En effet, l'approximation B-spline de la définition 1.2 repose, au niveau des bords, sur la définition de points fantômes. Ces derniers étant reliés aux points (X_0, a_0), (X_1, a_1), (X_{n-1}, a_{n-1}) et (X_n, a_n), on a en vérité affaire à une optimisation des valeurs $(a_i)_{i=-1:n+1}$ sous contraintes. On propose dans ce cours de contourner la difficulté en utilisant une autre expression de l'approximation B-spline.

Remarque 6.1 *La position des nœuds de σ est importante. Effectivement, comme un polynôme de degré trois possède au maximum un minimum local et un maximum local, une spline cubique ne peut avoir plus d'un maximum et d'un minimum entre deux nœuds. Cela limite donc de façon claire les oscillations dans chaque intervalle. Il en résulte que plus X_j et X_{j+1} sont éloignés plus la spline des moindres carrés aura un effet de filtrage sur l'intervalle $[X_j..X_{j+1}]$. On va donc pouvoir se servir de cette propriété afin de filtrer du bruit de mesure sur les données x_i.*

6.2.2 Autre expression de l'approximation B-spline

Plutôt que de rajouter des points fantômes, on peut utiliser des B-splines particulières aux bords pour exprimer l'approximation B-spline.

Proposition 6.1 *L'approximation B-spline des points $(x_j, a_j)_{j=0:n}$ telle qu'introduite définition 1.2 peut encore s'écrire :*

$$\forall x \in [x_0 \,..\, x_n] \,, \qquad \sigma(x) = \sum_{j=0:n} a_j \, C_j(x),$$

où les fonctions $(C_j)_{j=0:n}$ sont définies à partir des B-splines cubiques comme suit :

- $C_0 = 2B_{-1} + B_0 \quad ; \quad C_1 = B_1 - B_{-1}$,
- $\forall j \in [2 : n-2] \,, \;\; C_j = B_j$,
- $C_{n-1} = B_{n-1} - B_{n+1} \quad ; \quad C_n = 2B_{n+1} + B_n$.

Démonstration. La démonstration est triviale étant donné la définition des points fantômes dans le chapitre 1. □

6.2.3 Définition et détermination

On détermine maintenant la spline de nœuds fixés $(X_j)_{j\in[0:n]}$ (équi-répartis), qui soit « le plus proche possible » (au sens de la norme L_2) des données $(x_i, y_i)_{i\in[0:N]}$.

Proposition 6.2 *Soient $N+1$ données $(x_i, y_i)_{i\in[0:N]}$, et $n+1$ abscisses équidistantes $(X_j)_{j\in[0:n]}$,*

- *Il existe un et un seul vecteur $a = (a_j)_{j\in[0:n]}$ telle que l'approximation B-spline σ des points $(X_j, a_j)_{j\in[0:n]}$ minimise la quantité :*

$$E = \sum_{i\in[0:N]} \rho_i \, (\sigma(x_i) - y_i)^2.$$

- *Le vecteur a est solution du système linéaire :*

$$\forall k \in [0 : n] \,, \qquad \sum_{j\in[0:n]} a_j \left(\sum_{i\in[0:N]} \rho_i \, C_j(x_i) \, C_k(x_i) \right) = \sum_{i\in[0:N]} \rho_i \, C_k(x_i) \, y_i,$$

c'est-à-dire :

$$\forall k \in [0 : n] \,, \qquad \sum_{j\in[0:n]} A_{kj} \, a_j = S_k,$$

avec :

$$\forall k, j \in [0 : n] \,, \qquad A_{k\,j} = \sum_{i\in[0:N]} \rho_i \, C_k(x_i) \, C_j(x_i),$$

$$\forall k \in [0 : n] \,, \qquad S_k = \sum_{i\in[0:N]} \rho_i \, C_k(x_i) \, y_i.$$

Démonstration. En appliquant le travail réalisé dans la section 2.1, la démonstration de ces deux propriétés devient triviale. □

Remarque 6.2 *La matrice A est une matrice 7-diagonales et peut encore s'écrire $A = U^T U$, où U est la matrice $(N+1) \times (n+1)$ donnée par :*

$$\forall i \in [0:N] \,, \qquad \forall j \in [0:n] \,, \qquad U_{i\,j} = \sqrt{\rho_i}\, C_j(x_i).$$

Le système linéaire s'écrit alors $U^T U\, a = U^T Y$ avec $Y_i = \sqrt{\rho_i}\, y_i$.