

Université d'Orléans

U.F.R. Faculté des Sciences

Licence de Physique

Analyse Numérique pour les Sciences Physiques

Interpolation

Jean-Philippe Grivet, 2003

1 Définition

Dans le chapitre 1, j'ai montré plusieurs méthodes d'approximation d'une fonction. L'interpolation, qui fait l'objet de ce chapitre, poursuit le même but que l'approximation : remplacer une fonction difficile à calculer par une expression plus simple pour, par exemple, pouvoir la calculer numériquement vite et souvent, ou pour évaluer sa dérivée ou son intégrale. Le cadre mathématique est aussi assez semblable. Je m'intéresse à une fonction réelle continue $f(x)$ que je vais encore remplacer par une fonction plus simple $f^*(x)$. La différence réside dans la manière d'imposer la «proximité» de $f(x)$ et de son «substitut» $f^*(x)$. Je vais demander que $f(x) \equiv f^*(x)$ pour un certain nombre de valeurs de la variable indépendante x . On imposait que l'approximant reste dans une bande de largeur 2ϵ entourant la fonction ; on impose maintenant que l'interpolant coïncide avec la fonction pour certaines valeurs de la variable indépendante. On peut ainsi comprendre le terme d'approximation exacte autrefois employé pour désigner l'interpolation.

Dans un passé lointain, alors qu'il n'existait pas d'ordinateur, le calcul d'une fonction compliquée était laborieux. On dressait une table de la fonction pour quelques valeurs de l'argument puis on interpolait entre ces valeurs. De nos jours, cet usage de l'interpolation a beaucoup régressé, mais l'interpolation reste un outil théorique important. D'autre part, on a souvent besoin des valeurs de grandeurs physiques comme la densité, la conductivité électrique ou des chaleurs de réaction, qui, à leur tour, dépendent de la pression, de la température ou des concentrations. Ces grandeurs figurent dans des tables, mais seulement pour quelques valeurs des paramètres. Là encore, je serai amené à interpoler pour déterminer les valeurs qui m'intéressent.

L'échelle internationale de température constitue un autre exemple physique d'interpolation. La température légale est la température thermodynamique, associée à un unique point fixe, le point triple de l'eau à 273,15 K ; elle se mesure à l'aide d'un thermomètre à gaz à volume constant ou d'un thermomètre à rayonnement. Ces deux instruments sont malcommodes et impliquent des manipulations longues et compliquées. Aussi, on a choisi une série de points fixes secondaires (points d'ébullition de l'hydrogène, de l'eau, point de fusion du zinc par exemple) dont les températures ont été mesurées très précisément une fois pour toutes. On a choisi aussi quelques thermomètres «pratiques», comme le thermomètre à résistance de platine. Une fois mesurée la résistance de la sonde pour chaque point fixe, la valeur d'une température intermédiaire est obtenue par interpolation.

J'aborde maintenant le formalisme. Étant donnée une fonction $y = f(x)$ et une fonction $f^*(x; a_0, a_1, \dots, a_n)$ dépendant de x et de $n + 1$ paramètres a_0, a_1, \dots, a_n , le problème d'interpolation consiste à déterminer les a_k de telle manière que les $n + 1$ équations

$$f(x_k) = f^*(x_k; a_0, a_1, \dots, a_n), \quad k = 0, 1, 2, \dots, n \quad (1)$$

soient vérifiées. Dans la suite, j'utiliserai indifféremment les notations $f(x_k) = f_k = y_k$; chaque couple de valeurs x_k, y_k définit un point du plan que l'on appelle un pivot ou un noeud.

Il me faut ensuite décider à quelle classe de fonctions appartient f^* ; le critère le plus important est la façon dont interviennent les paramètres a_k . On distingue les cas où f^* dépend linéairement des a_k de tous les autres ; l'interpolation polynômiale

$$f^* = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

et l'interpolation trigonométrique

$$f^* = a_0 + a_1e^{ix} + a_2e^{2ix} + \dots + a_n e^{inx}$$

relèvent de ce cas. Par contre, l'interpolation rationnelle

$$f^*(x; a_0, a_1, \dots, a_m, b_0, b_1, \dots, b_n) = \frac{a_0 + a_1x + a_2x^2 + \dots + a_mx^m}{b_0 + b_1x + \dots + b_nx^n}$$

(qui par ailleurs fait appel à $m + n + 2$ coefficients) dépend non-linéairement des b_k . Bien que l'interpolation rationnelle soit parfois utilisée, je me limiterai ici à l'interpolation polynomiale qui n'implique que des calculs simples. Les formes trigonométriques seront abordées plus tard.

Vous remarquez qu'à la différence de l'approximation, il ne paraît pas urgent de définir un intervalle dans lequel l'interpolation est valable ; en réalité, cette définition est implicite. Si x_{min} est le plus petit des x_i et x_{max} le plus grand, je parle d'interpolation lorsque $x_{min} \leq x \leq x_{max}$; je montrerai comment estimer l'erreur correspondante. Si, au contraire, je m'intéresse à une valeur de x extérieure à l'intervalle $[x_{min}, x_{max}]$, je pratique une extrapolation et je ne sais pas borner l'erreur dans ce cas ; l'examen de quelques exemples vous montrerait que l'erreur d'extrapolation peut devenir très grande. En pratique donc, l'interpolation ne concerne que les valeurs de x comprises entre la plus petite et la plus grande abscisse des pivots.

2 Méthode des coefficients indéterminés

La méthode la plus directe pour trouver les coefficients inconnus a_k consiste à identifier, sur chaque pivot, fonction et polynôme d'interpolation. Je montre le principe de la méthode sur l'exemple de deux pivots. La fonction f est connue numériquement en deux points, ce qui définit les deux pivots x_0, f_0, x_1, f_1 . La fonction d'interpolation, que je note maintenant $p(x)$, puisqu'il s'agit d'un polynôme, dépend linéairement de deux paramètres inconnus et s'écrit $p = a_0 + a_1x$. Géométriquement, $p(x)$ représente la droite qui passe par les deux pivots. Les conditions d'interpolation sont alors

$$\begin{cases} a_0 + a_1x_0 = f_0 \\ a_0 + a_1x_1 = f_1 \end{cases}$$

Je dois donc, pour déterminer les $\{a_i\}$, résoudre un système de deux équations linéaires à deux inconnues ; la solution est immédiate :

$$a_1 = \frac{f_0 - f_1}{x_0 - x_1} \quad ; \quad a_0 = \frac{x_0f_1 - x_1f_0}{x_0 - x_1}.$$

Ce procédé s'étend sans difficulté au cas d'une fonction du second degré, définie par trois paramètres inconnus ; la courbe représentative de $p(x)$ est une parabole que l'on oblige à passer par trois pivots. En principe, je peux même considérer le cas d'une fonction d'interpolation polynomiale de degré n , dépendant de $n + 1$ constantes inconnues. Celles-ci seront déterminées en imposant que f et f^* coïncident sur $n + 1$ noeuds :

$$\begin{cases} a_0 + a_1x_0 + a_2x_0^2 + \cdots + a_nx_0^n = y_0, \\ a_0 + a_1x_1 + a_2x_1^2 + \cdots + a_nx_1^n = y_1, \\ a_0 + a_1x_2 + a_2x_2^2 + \cdots + a_nx_2^n = y_2, \\ \cdots \\ a_0 + a_1x_n + a_2x_n^2 + \cdots + a_nx_n^n = y_n. \end{cases} \quad (2)$$

Il apparaît ici encore un système d'équations linéaires dont les inconnues sont les $n + 1$ nombres a_i et dont le déterminant s'écrit

$$\Delta = \begin{vmatrix} x_0^0 & x_0^1 & x_0^2 & \cdots & x_0^n \\ x_1^0 & x_1^1 & x_1^2 & \cdots & x_1^n \\ x_2^0 & x_2^1 & x_2^2 & \cdots & x_2^n \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_n^0 & x_n^1 & x_n^2 & \cdots & x_n^n \end{vmatrix}. \quad (3)$$

Δ , que l'on appelle un déterminant de van der Monde, s'annule chaque fois que la condition $x_i = x_j$, $i \neq j$ est remplie; le développement de ce déterminant contient donc le facteur $x_i - x_j$. En répétant ce raisonnement pour tous les couples possibles de pivots, je montre facilement que

$$\Delta = \prod_{i < j} (x_i - x_j).$$

Je déduis de ce résultat que Δ est non nul, et donc que le système admet une solution unique, si et seulement si les pivots ont des abscisses strictement distinctes. Je ne fait pas d'autre hypothèse sur les pivots; en particulier, ceux-ci peuvent être rangés dans un ordre arbitraire.

Je dispose maintenant d'une méthode pour construire le polynôme d'interpolation; si j'ai pris soin de choisir des pivots différents, ce polynôme est unique. Dans la pratique, la méthode des coefficients indéterminés est, en fait, rarement utilisée, car on connaît des algorithmes plus stables et/ou plus rapides. Ces algorithmes, inventés par des mathématiciens célèbres des siècles passés, prennent des formes très différentes et produisent des résultats apparemment dissemblables. En réalité, ils aboutissent tous à former l'unique polynôme d'interpolation à partir des pivots disponibles. La première méthode que je présenterai est due à Lagrange.

3 Le polynôme d'interpolation de Lagrange

Dans cette méthode, je fait l'hypothèse (à vérifier) que le polynôme d'interpolation peut s'écrire, pour $n + 1$ pivots $x_i, f(x_i)$

$$p(x) = \sum_{i=0}^n L_i(x) f(x_i). \quad (4)$$

Chaque polynôme élémentaire de Lagrange $L_i(x)$ est de degré n et, par conséquent, p est un polynôme de degré au plus égal à n . Vous voyez que le problème d'interpolation sera résolu si je peux former des L_i répondant aux conditions

$$L_i(x_k) = \delta_{i,k}, \quad i, k = 0, 1, 2, \dots, n. \quad (5)$$

$\delta_{i,k}$ est le symbole de Kronecker, qui vaut 1 lorsque ses deux indices sont égaux et qui vaut 0 dans tous les autres cas. Si, par exemple, $x = x_2$, alors $L_0(x_2) = L_1(x_2) = L_3(x_2) = \dots = L_n(x_2) = 0$ tandis que $L_2(x_2) = 1$. Des $n + 1$ termes de p ne subsiste que f_2 . Ainsi, pour le pivot x_2 , le polynôme d'interpolation coïncide avec la fonction et il en serait de même pour tout autre noeud.

Le polynôme élémentaire L_i s'annule pour tous les pivots sauf x_i ; il est donc proportionnel au polynôme $q(x) = (x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)$. La constante de proportionnalité s'obtient en formant $q(x_i) = (x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)$: L_i est égal au polynôme normalisé $q(x)/q(x_i)$. Le polynôme d'interpolation de Lagrange est maintenant entièrement déterminé; il s'écrit

$$p(x) = \sum_{i=0}^n L_i(x) f_i = \sum_{i=0}^n f_i \prod_{k=0, k \neq i}^n \frac{x - x_k}{x_i - x_k}. \quad (6)$$

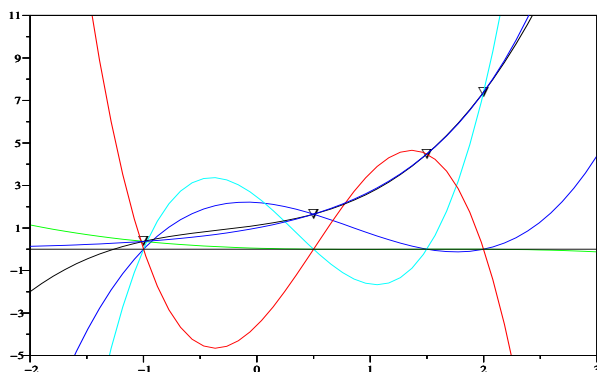
Le calcul numérique de $p(x)$ se programme aisément à l'aide de deux boucles imbriquées; il est bien plus rapide que la résolution du système linéaire du paragraphe précédent. Le fragment de programme ci-dessous réalise ce calcul.

```

x = linspace(xmin, xmax, npts);
pl = zeros(1, npts);
for i = 1:4
    L(i, 1:npts) = ones(1, npts);
    for k = 1:4
        if k <> i
            L(i, :) = L(i, :).*(x-xp(k))/(xp(i)-xp(k));
        end
    end
    L(i, :) = L(i, :)*fn(xp(i));
    pl = pl + L(i, :);
end

```

Les deux premières instructions créent deux vecteurs de zéros, qui vont recevoir respectivement les abscisses et les valeurs successives du polynôme d'interpolation. À partir de la ligne 3, je remplis la ligne i de la matrice L par des 1. Le gros du calcul est fait ligne 7, où je calcule, à l'aide d'une itération, les valeurs du polynôme élémentaire L_i , pour toutes les abscisses simultanément. La figure montre le résultat. J'ai représenté chacune des quantités $L_i(x)f(x_i)$, ainsi que leur somme $p(x)$. Vous pouvez vérifier que les $L_i(x)f(x_i)$ passe par un pivot et un seul, alors que $p(x)$ passe par tous les noeuds.



Cependant, il serait pénible de développer le polynôme de Lagrange et de regrouper les termes pour le mettre sous la forme d'une somme ordonnée selon les puissances de x . Si l'on souhaite vraiment obtenir cette forme, il vaut mieux utiliser un algorithme du à Newton, la méthode des différences divisées.

4 le polynôme de Newton

4.1 Interpolation linéaire

Il est commode de revenir à l'interpolation linéaire. Lorsque je dis que la fonction $f(x)$ est bien représentée par une interpolation linéaire dans un certain intervalle, cela signifie que le rapport

$$\frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

(la pente moyenne de la courbe représentative) est à peu près indépendant de x_0 et x_1 dans cet intervalle. Ce rapport s'appelle la différence divisée du premier ordre, relative à x_0 et x_1 , et se note généralement

$$f[x_0, x_1] \equiv \frac{f(x_1) - f(x_0)}{x_1 - x_0}. \quad (7)$$

Remarquez que $f[x_0, x_1] = f[x_1, x_0]$. Je peux aussi écrire la relation approchée

$$f[x_0, x] \cong f[x_0, x_1] \quad (8)$$

ou encore

$$f(x) \cong f(x_0) + (x - x_0)f[x_0, x_1]. \quad (9)$$

Vous reconnaissez, au second membre, le polynôme du premier degré qui interpole f sur les pivots (x_0, x_1) , une expression que je noterai $p_{0,1}$ dans la suite. Il est commode d'introduire les notations

$$p_0(x) \equiv f[x_0] \equiv f(x_0), \quad (10)$$

si bien que $f[x_0]$ est la différence divisée d'ordre zéro et $p_0(x)$ est le polynôme d'interpolation de degré zéro qui coïncide avec f en x_0 . La formule (9) prend alors une forme qu'il sera facile de généraliser plus tard :

$$p_{0,1} = f[x_0] + (x - x_0)f[x_0, x_1]. \quad (11)$$

À moins que la fonction f ne soit réellement linéaire en x , la pente de la sécante $f[x_0, x_1]$ va dépendre des abscisses x_0 et x_1 . Si f est du second degré en x , la pente de la sécante joignant les points d'abscisses x_0 et x est elle-même une fonction linéaire de x , à x_1 constant. Je m'attends donc à ce que la quantité

$$\frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

soit indépendante des trois arguments; dans le cas général, on appelle ce rapport la différence divisée d'ordre 2 relative aux abscisses x_0, x_1, x_2 et on le note $f[x_0, x_1, x_2]$. Cette expression est encore symétrique par rapport à tous ses arguments. Je peux alors écrire la relation (8)

$$f[x_0, x] - f[x_0, x_1] = f[x_0, x] - f[x_1, x_0] = (x - x_1)f[x_0, x_1, x]$$

d'où je déduis la relation exacte qui remplace (11)

$$f(x) = f(x_0) + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x].$$

Telle quelle, cette formule est peu utile, car connaître $f[x_0, x_1, x]$ revient à connaître $f(x)$, ce que je cherche justement à éviter. Mais l'erreur $E(x)$ commise en remplaçant $f(x)$ par $p_{0,1}$ est donnée par

$$f - p_{0,1} = (x - x_0)(x - x_1)f[x_0, x_1, x],$$

un résultat que je généraliserais plus tard.

4.2 Les différences divisées

Les différences divisées d'ordres $0, 1, 2, \dots, k$ sont définies par les relations de récurrence

$$f[x_0] = f(x_0) \quad f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} \quad f[x_0, \dots, x_k] = \frac{f[x_1, \dots, x_k] - f[x_0, \dots, x_{k-1}]}{x_k - x_0} \quad (12)$$

Dans chaque numérateur, les $k - 1$ premiers arguments du premier terme sont identiques aux $k - 1$ derniers arguments du deuxième terme et le dénominateur est la différence entre les arguments qui diffèrent d'un terme à l'autre. Il est possible de démontrer par récurrence que $f[x_0, \dots, x_k]$ est une fonction totalement symétrique de ses $k + 1$ arguments ; l'ordre de ceux-ci est donc indifférent. J'en déduis que $f[x_0, x_1, \dots, x_k]$ peut s'exprimer comme le quotient de la différence de deux différences divisées d'ordre $k - 1$ (ayant $k - 1$ arguments en commun) par la différence entre arguments différents :

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_0, x_1, x_2] - f[x_1, x_2, x_3]}{x_0 - x_3} = \frac{f[x_0, x_2, x_3] - f[x_1, x_2, x_3]}{x_0 - x_1}.$$

Jusqu'à présent, j'ai fait semblant d'ignorer la possibilité que deux abscisses puissent être égales ; si cela se produit, on peut donner un sens à la différence divisée par passage à la limite :

$$\lim_{x \rightarrow 0} f[x + h, x] = \lim \frac{f(x + h) - f(x)}{h} = f'(x).$$

4.3 La formule de Newton

D'après la définition (12), je peux écrire, en remplaçant à chaque fois un des x_i par x

$$\begin{aligned} f(x) &= f[x_0] + (x - x_0)f[x_0, x], \\ f[x_0, x] &= f[x_0, x_1] + (x - x_1)f[x_0, x_1, x], \\ &\dots = \dots \\ f[x_0, \dots, x_{n-1}, x] &= f[x_0, \dots, x_n] + (x - x_n)f[x_0, \dots, x_n, x]. \end{aligned}$$

En substituant la première équation dans la seconde, j'obtiens l'expression déjà vue

$$f(x) = f[x_0] + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x]$$

Par des substitutions successives, j'arrive finalement à

$$\begin{aligned} f(x) = f[x_0] &+ (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] \\ &+ (x - x_0)(x - x_1) \cdots (x - x_{n-1})f[x_0, x_1, \dots, x_n] \\ &+ (x - x_0)(x - x_1) \cdots (x - x_n)f[x_0, x_1, \dots, x]. \end{aligned} \quad (13)$$

Je passe de cette expression «formelle» (que je ne peux pas utiliser puisque j'ignore le dernier terme) à la formule d'interpolation de Newton en négligeant le terme inconnu :

$$\begin{aligned} p_{0,1,2,\dots,n} = f[x_0] &+ (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] \\ &+ (x - x_0)(x - x_1) \cdots (x - x_{n-1})f[x_0, x_1, \dots, x_n] \end{aligned} \quad (14)$$

L'erreur commise en remplaçant f par p vaut

$$E(x) = (x - x_0)(x - x_1) \cdots (x - x_n)f[x_0, x_1, \dots, x].$$

Je donnerai plus tard une forme plus pratique de cette erreur.

Voici un exemple de calcul de différences divisées. J'utilise les mêmes données qu'au paragraphe précédent ; bien que cela ne soit pas nécessaire, j'ai rangé les abscisses par ordre croissant, ce qui rend la suite de opérations plus facile à suivre. Les pivots sont numérotés implicitement de 0 à 3.

ordre :	0	1	2	3
-1	0,367879			
		1,280842		
		1,5	0,853895	
0,5	1,648721		1,979073	
		2,83297	2,5	0,791629
		1	2,83297	1,196215
1,5	4,481689		2,981766	3
		2,907367	1,5	1,987844
		0,5	5,814734	0,398738
2	7,389056			

La première colonne contient les valeurs de x , la deuxième celles de f , qui sont identiques aux différences divisées d'ordre zéro. Je donne dans la colonne suivante les valeurs des quantités $x_i - x_j$ et $f_i - f_j$, dont le quotient fournit la différence divisée d'ordre 1, reportée dans la colonne numérotée 1. Le calcul se poursuit sans difficulté jusqu'à la différence divisée d'ordre 3, la seule que je puisse calculer avec les données dont je dispose.

Le polynôme de Newton d'ordre 3 s'écrit alors

$$p_{0,1,2,3} = 0,367879 + (x-0,5)0,853895 + (x-0,5)(x-1,5)0,791629 + (x-0,5)(x-1,5)(x-2)0,398738.$$

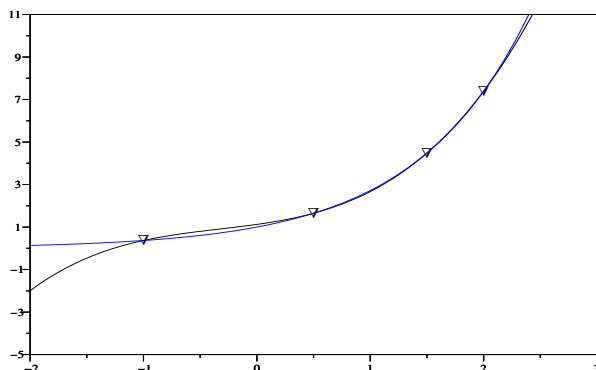
```

// differences divisees de Newton      1
// creation d'une serie de valeurs     2
x = [0.5, -1.0, 2.0, 1.5];           3
y = exp(x);                           4
neff = input("ordre de l interpolation (<= 3: "); 5
// calcul des differences              6
for i = 1:neff+1                       7
    t(i) = y(i);                        8
    for j = i-1:-1:1                    9
        t(j) = (t(j+1) - t(j))/(x(i) - x(j)); 10
    end                                  11
    a(i) = t(1);                        12
end                                     13
a                                       14
z = input("valeur de l argument: ");  15
// calcul du polynome                 16
pol = a(neff+1);                       17
z = linspace(-2,3);                    18
for i = neff:-1:1                      19
    pol = pol.*(z-x(i))+a(i);          20
end                                     21

```


<pre>rect = [-3,4,0,10]; plot2d(z', [pol', exp(z)'])</pre>	22 23
--	----------

Sa forme, très semblable à celle de Horner, rend le calcul facile pour toute valeur de x . Le listing ci-dessus montre un programme Scilab qui accomplit la même tâche. La figure illustre le résultat ; comme le polynôme d'interpolation est unique, le résultat final est indiscernable de celui du paragraphe précédent.



5 Erreur d'interpolation

Je cherche à majorer l'erreur commise en remplaçant la fonction $f(x)$ par le polynôme de degré n $p(x)$, lequel satisfait aux $n + 1$ conditions d'interpolation

$$p(x_i) = f(x_i) \equiv f_i, \quad i = 0, 1, 2, \dots, n.$$

Je définis tout d'abord le polynôme de degré $n + 1$

$$\pi(x) \equiv (x - x_0)(x - x_1) \cdots (x - x_n) = \prod_0^n (x - x_i)$$

puis la fonction auxiliaire

$$F(x) \equiv f(x) - p(x) - K\pi(x),$$

où K est une constante. Il est possible de choisir la constante K de telle manière que $F(X) = 0$, X étant un point quelconque de l'axe réel. J'appelle I le plus petit intervalle contenant X et tous les x_i . Alors la fonction F admet $n + 2$ zéros sur I . En effet, $f - p = 0$ lorsque $x = x_i$ et, simultanément, $\pi(x_i) = 0$. De plus, F s'annule en X .

Le théorème de Rolle appliqué à la fonction F me permet d'affirmer que la fonction $F'(x)$ présente au moins $n + 1$ zéros sur ce même intervalle ; en appliquant ce même théorème aux dérivées successives de F , je démontre que F'' admet au moins n zéros dans I , que $F^{(3)}$ en admet au moins $n - 1$ et finalement que $F^{(n+1)}$ s'annule au moins une fois dans I , à condition que cette dérivée existe ; j'appelle ξ ce point. La dérivée d'ordre $n + 1$ de F se calcule aisément ; $p^{(n+1)} = 0$

puisque p est de degré n . Le terme de degré le plus élevé de π est x^{n+1} , si bien que la dérivée cherchée est $\pi^{(n+1)} = (n+1)!$. Je déduis que

$$F^{(n+1)}(\xi) = f^{(n+1)}(\xi) - K(n+1)! = 0$$

ou encore que

$$K = \frac{f^{(n+1)}(\xi)}{(n+1)!}$$

d'où je tire que

$$f(X) - p(X) = K\pi(X) = \frac{\pi(X)}{(n+1)!} f^{(n+1)}(\xi).$$

Je peut donc énoncer le théorème suivant.

Si une fonction f possède une dérivée d'ordre $n+1$, alors, pour toute valeur X de l'argument, il existe un nombre ξ appartenant au plus petit intervalle qui contient X et tous les pivots x_i et satisfaisant à

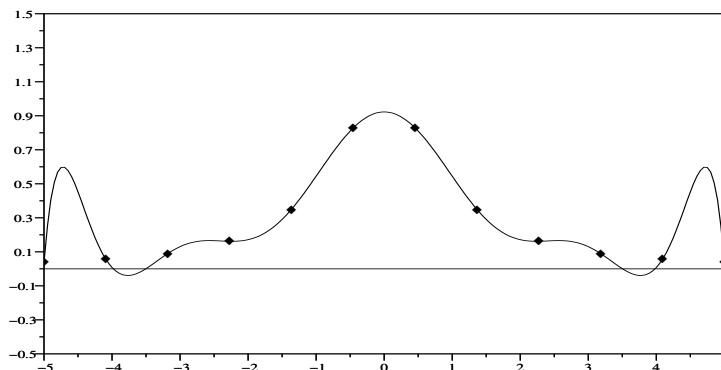
$$f(X) - p_{0,1,2,\dots,n}(X) = \frac{\pi(X)f^{(n+1)}(\xi)}{(n+1)!}. \quad (15)$$

En pratique, comme je ne connaît pas le nombre ξ et que je cherche un résultat valable pour tout l'intervalle I , j'utiliserai un majorant de la valeur absolue du second membre :

$$|f - p| \leq \frac{1}{(n+1)!} \sup_{x \in I} |\pi f^{(n+1)}|.$$

Cette relation est souvent assez pessimiste. D'autre part, son utilité pratique est assez faible parce que si f est compliquée (et c'est pour cela que je souhaite la remplacer par un polynôme), sa dérivée d'ordre $n+1$ a toutes les chances d'être impossible à majorer facilement.

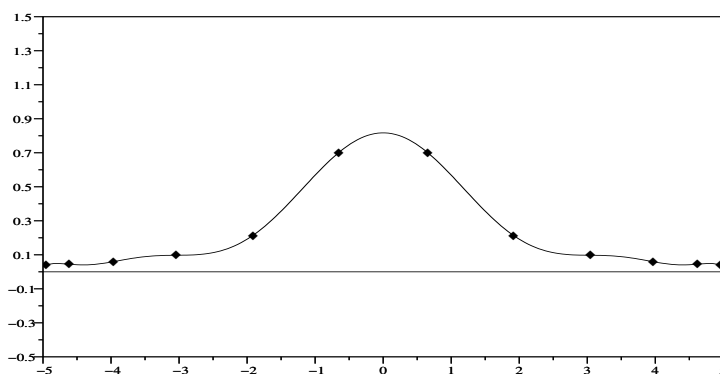
On pourrait penser que l'erreur diminue lorsque le nombre de pivots augmente, ou plutôt, lorsque ceux-ci deviennent de plus en plus serrés. Il n'en est rien ; le théorème de Faber établit au contraire l'existence de fonctions pour lesquelles l'erreur augmente avec le nombre de noeuds. La figure ci-dessous vous présente un exemple de cette propriété paradoxale, connue sous le nom de phénomène de Runge. J'ai interpolé la fonction $1/(1+x^2)$ en utilisant des pivots régulièrement espacés ; vous voyez que l'erreur d'interpolation devient très grande aux bords de l'intervalle.



Je peux aussi réaliser une interpolation extrêmement précise en choisissant «bien» les pivots. Les pivots définis par l'instruction Scilab

```
for k = 1:npiv, xp(k) = 0.5*(xmax - xmin)*cos( (2*k-1)*%pi/(2*npiv) ); end;
```

sont particulièrement efficaces, comme le montre la figure ci-dessous.



L'explication de ce bon comportement est traitée en exercice. Elle tient à ce que ces abscisses sont proportionnelles aux zéros du polynôme de Tschebychef d'ordre `npiv`.

6 Interpolation entre pivots équidistants

Supposons que je dispose d'une table des valeurs d'une fonction $f(x)$ calculées pour des valeurs entières de x et que j'ai besoin de connaître $f(3,4)$. Je peux utiliser les connaissances acquises en lisant les paragraphes précédents et interpoler entre les valeurs $f(3)$ et $f(4)$ par exemple. Les algorithmes de Lagrange ou de Newton s'appliquent sans difficulté, mais ici les données présentent une caractéristique intéressante : les pivots sont régulièrement espacés. Une telle situation se rencontre fréquemment lorsque l'on calcule à la main et de nombreuses méthodes ont été développées pour tirer parti au mieux de ce cas particulier. D'autre part, le même formalisme est utilisé pour résoudre numériquement les équations différentielles et les équations aux dérivées partielles. Ceci justifie que je consacre quelques pages à l'interpolation entre pivots équidistants.

6.1 Les différences finies

Étant donné une fonction $f(x)$ et une constante positive h , je défini la différence latérale ascendante

$$\Delta_h f(x) \equiv f(x+h) - f(x).$$

En général, la constante h est définie par le contexte, ce qui me dispense de la faire figurer en indice. Comme je ne travaillerai, dans la suite, qu'avec des abscisses équidistantes $x_i = x_0 + ih$, $i = 0, 1, 2, 3, \dots$, j'utiliserai la notation plus concise

$$\Delta f(x_i) \equiv \Delta f_i = f(x_{i+1}) - f(x_i) \equiv f_{i+1} - f_i. \quad (16)$$

h s'appelle le pas ou l'intervalle tabulaire et $f(x_{i+1}) = f(x_i+h)$. Je défini maintenant par récurrence les différences latérales d'ordre supérieur

$$\Delta^{p+1}f \equiv \Delta^p f(x+h) - \Delta^p f(x), \quad p \geq 0. \quad (17)$$

Par convention, $\Delta^0 f(x) \equiv f(x)$. Ainsi, $\Delta^2 f = f(x+2h) - 2f(x+h) + f(x)$. Le calcul pratique des différences finies se fait commodément à l'aide d'un tableau (semblable à celui utilisé pour former les différences divisées, mais sans divisions).

x_i	f_i	Δf_i	$\Delta^2 f_i$	$\Delta^3 f_i$
x_0	f_0			
		Δf_0		
x_1	f_1		$\Delta^2 f_0$	
		Δf_1		$\Delta^3 f_0$
x_2	f_2		$\Delta^2 f_1$	
		Δf_2		$\Delta^3 f_1$
x_3	f_3		$\Delta^2 f_2$	
		Δf_3		$\Delta^3 f_2$
x_4	f_4		$\Delta^2 f_3$	
		Δf_4		
x_5	f_5	\vdots	\vdots	

6.2 La formule d'interpolation de Newton

Il existe une relation simple entre le polynôme de Newton (formé à l'aide de différences divisées) et les différences latérales : pour tout entier k positif,

$$f[x_0, x_1, x_2, \dots, x_k] = \frac{1}{k!h^k} \Delta^k f_0. \quad (18)$$

Cette formule se démontre par récurrence. Elle est manifestement vraie pour $k = 0$. J'ai, pour $k = 1$,

$$f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0} = \frac{1}{h} \Delta f_0,$$

ce qui est conforme au résultat proposé. Je suppose maintenant que la formule est vraie pour tout ordre $k \leq r$. Alors, pour $k = r + 1$, la propriété (12) montre que

$$f[x_0, x_1, \dots, x_{k+1}] = \frac{f[x_1, x_1, \dots, x_{r+1}] - f[x_0, \dots, x_r]}{x_{r+1} - x_0}.$$

En remplaçant les différences divisées d'ordre r du second membre par leurs expressions, je trouve

$$\frac{1}{(r+1)h} \left[\frac{1}{r!h^r} \Delta^r f_1 - \frac{1}{r!h^r} \Delta^r f_0 \right] = \frac{1}{(r+1)!h^{r+1}} \Delta^{r+1} f_0,$$

ce qui établit le résultat pour $k = r + 1$. Il reste à utiliser (18) pour transformer le polynôme d'interpolation de Newton. Je défini une variable réduite

$$\mu = \frac{x - x_0}{h}.$$

Je trouve ensuite que $x - x_j = h(\mu - j)$ et que

$$(x - x_0)(x - x_1) \cdots (x - x_k) = \mu(\mu - 1) \cdots (\mu - k)h^{k+1}.$$

Insérant ces résultats dans (14), j'obtiens

$$p_{0,1,\dots,n}(x) = f_0 + \mu h \frac{\Delta f_0}{h} + \mu(\mu - 1)h^2 \frac{\Delta^2 f_0}{2!h^2} + \cdots + \mu(\mu - 1) \cdots (\mu - n + 1) \frac{\Delta^n f_0}{n!h^n}.$$

Je peux donner à cette formule une allure plus compacte en introduisant des analogues fractionnaires des coefficients du binôme

$$C_\mu^k \equiv \frac{\mu(\mu - 1) \cdots (\mu - k + 1)}{k!}. \quad (19)$$

Avec cette définition, le polynôme d'interpolation s'écrit

$$p_{0,1,\dots,n}(x) = \sum_{j=0}^n C_\mu^j \Delta^j f_0. \quad (20)$$

Cette formule générale donne, pour $n = 1$

$$p_{0,1}(x) = f_0 + \mu \Delta f_0$$

et, pour $n = 2$,

$$p_{0,1,2} = f_0 + \mu \Delta f_0 + \frac{1}{2} \mu(\mu - 1) \Delta^2 f_0.$$

Voici un exemple complet d'interpolation par les différences finies et la formule de Newton. J'ai préparé un tableau de f et de ses différences latérales.

i	x_i	f_i	Δf	$\Delta^2 f$	$\Delta^3 f$	$\Delta^4 f$	$\Delta^5 f$
0	1	1					
			0,61051				
1	1,1	1,61051		0,26730			
			0,87781		0,0795		
2	1,2	2,48832		0,3468		0,0144	
			1,22461		0,0939		0,0012
3	1,3	3,71293		0,4407		0,0156	
			1,66531		0,1095		
4	1,4	5,37824		0,5502			
			2,21551				
5	1,5	7,59375					

Pour cette fonction assez régulière, les différences diminuent lorsque l'ordre croît, mais le nombre de chiffres significatifs diminue aussi.

Je cherche la valeur de $f(1,25)$. Je décide d'utiliser toutes les valeurs disponibles, à commencer par f_0 . J'ai donc $\mu = (1,25 - 1)/0,1 = 2,5$. Les termes successifs du polynôme d'interpolation s'écrivent

f_0	1
$(2, 5)\Delta^1 f_0$	1, 526275
$\frac{(2, 5)(1, 5)}{2}\Delta^2 f_0$	0, 501187
$\frac{(2, 5)(1, 5)(0, 5)}{6}\Delta^3 f_0$	0, 024844
$\frac{(2, 5)(1, 5)(0, 5)(-0, 5)}{24}\Delta^4 f_0$	-0, 000563
$\frac{(2, 5)(1, 5)(0, 5)(-0, 5)(-1, 5)}{120}\Delta^5 f_0$	0, 000014
$p(1, 25)$	3, 051758

Il est facile, avec cette méthode, de suivre les « progrès » de l'interpolation : je m'arrête dès que je rencontre une contribution assez petite.

Il existe bien d'autres formules d'interpolation, associées aux noms de Gauss, Bessel et autres ; certaines seront présentées en exercices.

7 Interpolation de Hermite

Dans le but de construire une fonction d'interpolation f^* encore plus précise, je peux imposer à celle-ci des conditions supplémentaires. Hermite a proposé que les dérivées de f^* coïncident avec les dérivées correspondantes de f pour certains pivots. Je vais suivre cette idée, en me limitant à l'interpolation polynomiale et en ne considérant que la dérivée première. Pour tous les pivots, j'impose donc que

$$p(x_i) \equiv f_i \quad ; \quad p'(x_i) \equiv f'(x_i) = f'_i, \quad i = 0, 1, 2, \dots, n. \quad (21)$$

En d'autres termes, $p(x)$ interpole f et $p'(x)$ interpole f' , sur les mêmes pivots. Je suppose que le polynôme d'interpolation de Hermite, ainsi défini, peut s'écrire sous une forme analogue à celle de Lagrange

$$p(x) = \sum_{i=0}^n H_i(x) f_i + \sum_{i=0}^n \bar{H}_i(x) f'_i, \quad (22)$$

où les H_i, \bar{H}_i sont des polynômes réels distincts. Quel en est le degré ? Le polynôme p doit satisfaire aux $2n + 2$ conditions (21) : il comporte donc $2n + 2$ coefficients ajustables et est de degré $2n + 1$. Les polynômes élémentaires H et \bar{H} ont donc un degré au plus égal à $2n + 1$.

Les polynômes H_i sont tels que p interpole f ; ils obéissent donc à des conditions identiques à celles que j'ai écrites pour les polynômes élémentaires de Lagrange. Il paraît de même raisonnable que les polynômes dérivés \bar{H}'_i soient choisis de façon à ce que p' interpole f' , d'où une nouvelle série de conditions. Mais cela ne suffit pas : il ne faut pas que les H'_i viennent perturber l'interpolation de f' , ni que les \bar{H}_i détruisent l'interpolation de f : ces polynômes doivent être nuls sur les pivots.

J'obtiens ainsi quatre séries de conditions

$$\begin{aligned}
 H_i(x_k) &= \delta_{i,k} & i, k = 0, 1, 2, \dots, n, & \quad (a) \\
 H_i'(x_k) &= 0 & i, k = 0, 1, 2, \dots, n, & \quad (b) \\
 \bar{H}_i(x_k) &= 0 & i, k = 0, 1, 2, \dots, n, & \quad (c) \\
 \bar{H}_i'(x_k) &= \delta_{i,k} & i, k = 0, 1, 2, \dots, n. & \quad (d)
 \end{aligned} \tag{23}$$

Chaque polynôme élémentaire obéit à $2n + 2$ conditions et peut donc être de degré $2n + 1$ au plus.

Je commence par construire le polynôme \bar{H}_i . Il s'annule ainsi que sa dérivée sur tous les pivots $k \neq i$; il admet donc des racines doubles en ces points et aussi une racine simple en x_i . Il s'exprime facilement à l'aide du polynôme élémentaire de Lagrange $L_i(x)$ construit sur les mêmes pivots et du facteur $x - x_i$:

$$\bar{H}_i(x) = C[L_i(x)]^2(x - x_i).$$

La dérivée $\bar{H}_i'(x_i)$ vaut 1 (condition 23(d)), ce qui impose $C \equiv 1$.

Le polynôme $H_i(x)$ admet tous les pivots sauf x_i comme racines doubles. Il peut donc s'écrire

$$H_i(x) = [L_i(x)]^2 t_i(x),$$

en désignant par t_i un polynôme du premier degré tel que H_i respecte les conditions (23a,b) en x_i :

$$H_i(x_i) = [L_i(x_i)]^2 t_i(x_i) = t_i(x_i) = 1 \quad ; \quad H_i'(x_i) = 2L_i(x_i)L_i'(x_i) + L_i^2(x_i)t_i'(x_i) = 0.$$

Un calcul sans difficulté montre que

$$t_i(x) = 1 - 2(x - x_i)L_i'(x_i).$$

L'erreur d'interpolation s'obtient par un raisonnement calqué sur celui de §3; elle s'écrit

$$f(X) - p(X) = \frac{[\pi(X)]^2}{(2n+2)!} f^{(2n+2)}(\xi), \tag{24}$$

l'abscisse ξ appartenant au plus petit intervalle qui contient X et tous les pivots x_i .

8 Interpolation inverse

Lorsque je dispose d'une table de valeurs d'une fonction f , c'est à dire d'un ensemble de couples x_i, f_i , et que je cherche la valeur de l'argument x correspondant à une valeur de f qui ne figure pas dans la table, je dois effectuer une interpolation inverse. Il s'agit en fait d'interpoler la fonction inverse f^{-1} à partir de pivots qui ne sont généralement pas équidistants.

La recherche d'une racine de l'équation $f(x) = 0$ peut relever de ce formalisme. Si je connais les valeurs de f pour des abscisses encadrant la racine, j'obtiens une bonne approximation de celle-ci par interpolation.

Exemple. Je cherche la solution de $\cos x = x$, sachant qu'elle est voisine de 0,7 radian. Je forme la table suivante

x	$\cos x - x$
0,5	0,377583
0,6	0,225336
0,7	0,064842
0,8	-0,103293
0,9	-0,278390

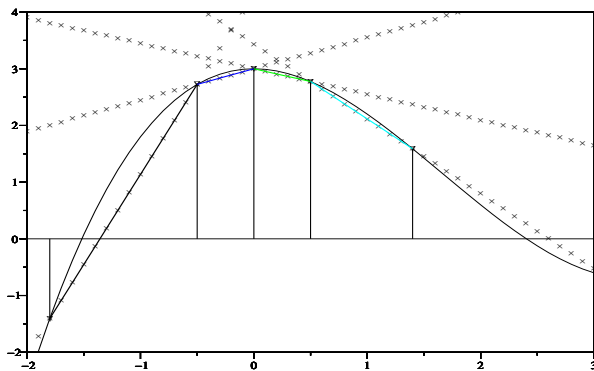
La racine est telle que $0,7 > f^{-1}(0) > 0,8$; j'en trouve une approximation par interpolation linéaire selon Lagrange :

$$x^* = 0,7 \frac{0 - (-0,103293)}{0,064842 - (-0,103293)} + 0,8 \frac{0 - 0,064842}{-0,103293 - 0,064842}$$

soit $x^* = 0,738565$ et $f(x^*) = 0,00087$.

9 Interpolation par intervalle

Cherchant à améliorer la qualité de l'interpolation d'une fonction donnée, je suis tenté d'augmenter le nombre n de pivots (ou de noeuds). Comme vous le savez, cette solution est souvent vouée à l'échec, parce que le terme d'erreur contient deux facteurs, le polynôme $\pi(x)$ et la dérivée d'ordre n (ou $2n$) de la fonction qui sont vraisemblablement rapidement croissant avec n .



Je peux cependant augmenter à volonté le nombre de pivots, *sans* faire croître l'ordre d'interpolation. La figure montre le principe de ce tout petit miracle. Pour interpoler la fonction représentée par la courbe en noir, j'ai défini 5 pivots (d'abscisses -1,8, -0,5, 0, 0,5, 1,4) et, ce faisant, 4 intervalles. Je pratique une interpolation du premier degré dans chaque intervalle; la fonction linéaire correspondante est définie pour tout x : elle est représentée par une ligne pointillée. Cependant, je n'utilise que la restriction de cette fonction à l'intervalle considéré, c'est-à-dire que j'assimile l'arc de courbe par le segment représenté en trait épais.

La fonction d'interpolation f^* est maintenant plus compliquée : elle admet une définition différente dans chaque intervalle. De plus, sa dérivée n'est plus continue. Ce type d'interpolation peut être rendu aussi précis que l'on veut en multipliant le nombre d'intervalles. On peut aussi le perfectionner en choisissant une méthode plus précise dans chaque intervalle, interpolation parabolique ou interpolation de Hermite.

Cependant, dans certaines applications, la disparition de la continuité de la dérivée pose problème : imagine-t-on un bureau d'étude qui calcule une coque de bateau formée d'une série de facettes raccordées par des angles vifs ? Il existe un formalisme qui combine l'avantage d'une interpolation «par morceaux» et la continuité des dérivées de la fonction d'interpolation : c'est la méthode des splines, que j'explique dans le paragraphe suivant.

10 Interpolation «spline»

Le mot «spline» désigne en anglais une règle souple et élastique (une baguette de bois ou une latte) dont les dessinateurs se servaient pour tracer des courbes lisses (ou régulières) passant par des points imposés. Le calcul du profil d'un pont ou d'une aile d'avion fournit les coordonnées d'un ensemble de points ; il appartient ensuite au dessinateur de produire une courbe esthétique ou aérodynamique mais passant par les points calculés. La forme de la règle souple était ajustée en y suspendant des poids ou en y attachant des ressorts en des points bien choisis. L'interpolation spline est la traduction algébrique de ce savoir-faire.

Je vais détailler un type particulier de fonctions spline, les splines cubiques, qui sont fréquemment utilisées et sont représentatives de cette catégorie de fonctions. Je me propose d'interpoler la fonction $f(x)$ sur l'intervalle $I = [a, b]$. Pour cela, je choisis une partition de I

$$a \equiv x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n \equiv b. \quad (25)$$

Ces $n+1$ abscisses définissent n sous-intervalles $[x_{i-1}, x_i]$. La fonction spline $s(x)$ répond alors aux contraintes suivantes

- Dans chaque intervalle $[x_{i-1}, x_i]$, la fonction d'interpolation est un polynôme de degré trois.
- La fonction s , sa dérivée première s' et sa dérivée seconde s'' sont continues dans I .
- $s(x)$ interpole la fonction f sur $[a, b]$, $s(x_i) = f_i, i = 0, 1, \dots, n$.

Pour que le problème de la détermination de la fonction s soit soluble, il faut au moins que le nombre d'équations soit égal au nombre d'inconnues. Cela est-il le cas ? Je peux poser

$$s(x) = a_i + b_i x + c_i x^2 + d_i x^3 \quad x_{i-1} \leq x \leq x_i \quad i = 1, 2, \dots, n. \quad (26)$$

Les inconnues sont les $4n$ coefficients a_i, b_i, c_i, d_i . Les conditions d'interpolation fournissent $n+1$ contraintes. Les conditions de continuité s'appliquent à chaque frontière entre sous-intervalles ; il y a $n-1$ points communs à deux intervalles et 3 séries de conditions, ce qui engendre $3n-3$ contraintes. Je dispose donc en tout de $4n-2$ relations pour déterminer $4n$ inconnues. Ce léger déficit n'est pas gênant et j'introduirai bientôt deux conditions supplémentaires qui me permettront de résoudre entièrement le problème.

Plutôt que de faire intervenir directement les coefficients du polynôme s , il est commode d'utiliser, pour la mise en équations, les quantités $m_i \equiv s''(x_i)$, $i = 0, 1, 2, \dots, n$. Comme s doit être du troisième degré sur $[x_i, x_{i+1}]$, sa dérivée seconde est une fonction linéaire de x que j'écris sous une forme qui minimise les calculs à venir et assure la continuité de s'' :

$$\begin{aligned} s''(x) &\equiv \frac{1}{h_i} [(x_{i+1} - x)m_i + (x - x_i)m_{i+1}], \\ h_i &\equiv x_{i+1} - x_i, \\ i &= 0, 1, 2, \dots, n-1. \end{aligned} \quad (27)$$

J'intègre maintenant deux fois par rapport à x , ce qui fait apparaître deux constantes arbitraires C_i et D_i

$$s(x) = \frac{(x_{i+1} - x)^3 m_i + (x - x_i)^3 m_{i+1}}{6h_i} + C_i(x_{i+1} - x) + D_i(x - x_i).$$

J'impose maintenant les conditions d'interpolation pour obtenir les expressions suivantes des constantes d'intégration

$$C_i = \frac{f_i}{h_i} - \frac{h_i m_i}{6} \quad D_i = \frac{f_{i+1}}{h_i} - \frac{h_i m_{i+1}}{6}.$$

Le polynôme s prend alors une forme assez symétrique

$$\begin{aligned} s(x) &= \frac{(x_{i+1} - x)^3 m_i + (x - x_i)^3 m_{i+1}}{6h_i} + \frac{(x_{i+1} - x)f_i + (x - x_i)f_{i+1}}{h_i} - \frac{h_i}{6} [(x_{i+1} - x)m_i + (x - x_i)m_{i+1}], \\ x_i &\leq x \leq x_{i+1}, \\ i &= 0, 1, 2, \dots, n-1. \end{aligned} \quad (28)$$

Vous pouvez vérifier que s est bien continue sur $[a, b]$. Il me reste à prendre en compte la continuité de s' ; pour cela, j'exprime cette dérivée dans deux intervalles successifs. Dans $[x_i, x_{i+1}]$

$$s' = \frac{-(x_{i+1} - x)^2 m_i + (x - x_i)^2 m_{i+1}}{2h_i} + \frac{f_{i+1} - f_i}{h_i} - \frac{(m_{i+1} - m_i)h_i}{6}$$

alors que dans l'intervalle précédent $[x_{i-1}, x_i]$:

$$s' = \frac{-(x_i - x)^2 m_{i-1} + (x - x_{i-1})^2 m_i}{2h_{i-1}} + \frac{f_i - f_{i-1}}{h_{i-1}} - \frac{(m_i - m_{i-1})h_{i-1}}{6}.$$

Ces deux expressions doivent vérifier

$$\lim_{x \rightarrow x_i^+} = \lim_{x \rightarrow x_i^-} \quad i = 1, 2, \dots, n-1.$$

Après un peu d'algèbre, je trouve l'équation assez simple suivante

$$\frac{h_{i-1}}{6} m_{i-1} + \frac{h_{i-1} + h_i}{3} m_i + \frac{h_{i+1}}{6} m_{i+1} = \frac{y_{i+1} - f_i}{h_i} - \frac{f_i - f_{i-1}}{h_{i-1}}, \quad i = 1, 2, \dots, n-1. \quad (29)$$

Je viens d'obtenir $n-1$ équations linéaires pour les $n+1$ inconnues m_i . De plus, ces équations ont une structure simple : chacune ne fait intervenir que trois inconnues.

Je dois maintenant imaginer deux équations supplémentaires. Il est d'usage d'imposer une condition raisonnable à chaque extrémité (a, b) de l'intervalle. Je peux, par exemple, supposer connues les pentes du polynôme d'interpolation en ces points :

$$s'(x_0) = f'_0 \quad s'(x_n) = f'_n, \quad (30)$$

où f'_0, f'_n ont des constantes. En utilisant les expressions de s' vues précédemment, pour $i = 0$ et pour $i = n - 1$, j'obtiens les deux équations manquantes

$$\frac{h_0}{3}m_0 + \frac{h_0}{6}m_1 = \frac{f_1 - f_0}{h_0} - f'_0,$$

$$\frac{h_{n-1}}{6}m_{n-1} + \frac{h_{n-1}}{3}m_n = f'_n - \frac{f_n - f_{n-1}}{h_{n-1}}.$$

L'ensemble des équations définissant les m_i peut être mis sous forme matricielle

$$AM = B$$

en définissant

$$B^T \equiv \left[\frac{f_1 - f_0}{h_0} - f'_0, \frac{f_2 - f_1}{h_1} - \frac{f_1 - f_0}{h_0}, \dots, \frac{f_n - f_{n-1}}{h_{n-1}} - \frac{f_{n-1} - f_{n-2}}{h_{n-2}}, f'_n - \frac{f_n - f_{n-1}}{h_{n-1}} \right],$$

$$M^T \equiv [m_0, m_1, \dots, m_n],$$

$$A = \begin{bmatrix} \frac{h_0}{3} & \frac{h_0}{6} & 0 & & \dots & 0 \\ \frac{h_0}{6} & \frac{h_0+h_1}{3} & \frac{h_1}{6} & & & \\ 0 & \frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & & \\ 0 & & & \ddots & & \\ \vdots & & & & \frac{h_{n-2}}{6} & \frac{h_{n-2}+h_{n-1}}{3} & \frac{h_{n-1}}{6} \\ 0 & & \dots & & \frac{h_{n-1}}{6} & \frac{h_{n-1}}{3} \end{bmatrix}$$

Ces équations définissent ce que l'on appelle parfois l'interpolant spline «complet». Je suis parvenu à concilier nombre de pivots arbitraire et continuité des deux premières dérivées, mais il y a un prix à payer : je dois maintenant résoudre un problème global (représenté par un système d'équations) et toute modification locale d'une donnée (par exemple de f_2) se répercutera sur toutes les valeurs de la fonction spline. Je n'aborderai pas ici l'étude de l'erreur d'interpolation, qui nécessite de trop longs développements et que l'on trouvera dans les ouvrages spécialisés.

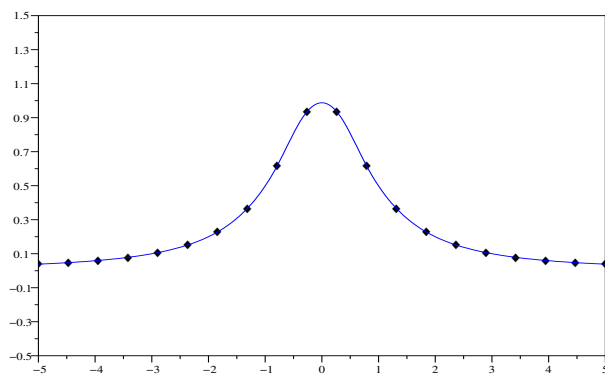
Le formalisme assez touffu qui précède est entièrement masqué à l'intérieur des deux fonctions de Scilab `splin` et `interp`. Je remarque tout d'abord que Scilab utilise, pour déterminer complètement les m_i , deux conditions qui ne font intervenir aucune donnée supplémentaire. Dans ce logiciel, on impose en effet la continuité de la dérivée troisième de s pour le deuxième et l'avant-dernier pivot. Cela revient à dire que s est un interpolant spline pour les noeuds $x_0, x_2, x_3, \dots, x_{n-2}, x_n$, mais interpole cependant sur tous les pivots $x_0, x_1, x_2, \dots, x_{n-1}, x_n$.

Je reprend l'exemple du §3, l'interpolation de la fonction $1/(1+x^2)$. Voici le programme Scilab, suivi du graphe correspondant.

```

//interpolation spline
//p
deff("y = fn(x)", "y = (1.0)./(1+x.*x)");
npts = 200;
npiv = input("nombre de pivots: ");
xmin = -5; xmax = 5;
x = linspace(xmin, xmax, npts);
xp = linspace(xmin, xmax, npiv);
xset("window", 0), xbase(0), xset("mark size", 3)
plot2d(xp, fn(xp), -4, rect = [xmin, -0.5, xmax, 1.5])
spl1 = splin(xp, fn(xp));
spl2 = interp(x, xp, fn(xp), spl1);
plot2d(x, spl2, 2, "000")

```



Le phénomène de Runge a disparu et l'interpolation est parfaitement régulière.

11 Interpolation à deux ou plusieurs dimensions

Les algorithmes exposés ci-dessus se généralisent de façon plus ou moins laborieuse à deux ou plusieurs dimensions. La méthode des éléments finis, utilisée pour résoudre les équations aux dérivées partielles, fait un usage intensif de l'interpolation à plusieurs variables; vous trouverez les meilleures introductions à l'interpolation à plusieurs dimensions dans les livres traitant des éléments finis. Le graphisme sur ordinateur (CAO, DAO) est un autre «consommateur» important d'algorithmes d'interpolation.

Je mentionne pour terminer une application classique de l'interpolation à deux variables : le tracé de courbes de niveau. Soit $f(x, y)$ une fonction définie dans une région du plan. Je peux représenter cette fonction comme une surface $z = f(x, y)$ (en fait comme une projection sur le papier ou l'écran de cette surface). Je peux aussi dessiner la courbe du plan xOy définie par l'équation $f(x, y) = C$. Les bulletins météo comportent souvent des cartes avec des isobares, lieux des points où la pression atmosphérique prend une valeur donnée. Pour dessiner automatiquement une courbe

de niveau, je dois d'abord construire un tableau de valeurs de f correspondant à tous les couples possibles d'abscisses $[x_i]$ et d'ordonnées $[y_j]$, donc en tous points d'un quadrillage couvrant la région qui m'intéresse. Je repère ensuite, sur les droites horizontales $x = x_i$ et sur les verticales $y = y_j$ les points pour lesquels $f = C$. Enfin, je relie ces points entre eux par une interpolation bidimensionnelle.