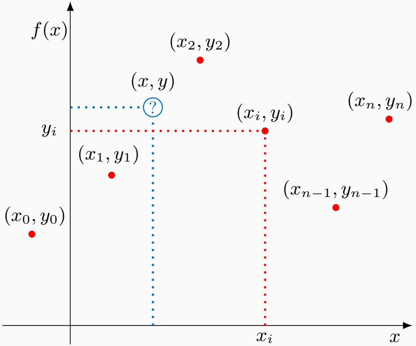
**Principe:** étant donné un ensemble de $n+1$ points $(x_i, y_i) = (x_i, f(x_i))$, $i = 0, \ldots, n$, on souhaite construire une fonction continue $p(x)$ passant par ces $n+1$ points, c-à-d telle que

$$p(x_i) = y_i, \quad i = 0, \ldots, n.$$




Les abscisses $x_i$ sont appelés **noeuds d'interpolation**. Les points $(x_i, y_i)$ sont appelés **points d'interpolation**, ou points de collocation.

Autrement dit, si l'on ne connaît que les points d'interpolation $(x_i, f(x_i))$, est-on capable d'obtenir une approximation de la fonction $f$ pour une valeur de $x$ différentes des $x_i$ ?

**But:** Comprendre le comportement général d'un phénomène à partir de résultats expérimentaux,

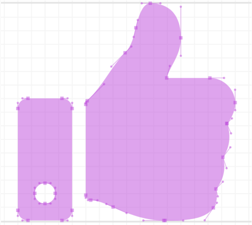
Domaines d'applications très divers: CAO, synthèse d'images, métrologie, géologie, typographie,. . .

Ce que nous allons voir:

- Interpolation de Lagrange,
- Interpolation de Newton,
- Splines cubiques.

Ce que nous n'allons pas voir, mais également utilisé en pratique:

- Courbes de Bézier, B-Splines, NURBS,
- Krigeage,
- Moindres carrés,
- . . .



La solution va être de créer un polynôme de degré suffisamment élevé et passant par les points d'interpolation. On parle de **polynôme d'interpolation**, ou polynôme de collocation.

Un rappel d'algèbre:

**Théorème (fondamental de l'algèbre)**

Soit $p_n$ un polynôme de degré $n$,

$$p_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x_n, \quad a_n \neq 0.$$

$p_n$ possède exactement $n$ racines (réelles ou complexes).

et une conséquence...

**Proposition**

Il existe un **unique** polynôme d'interpolation $p_n$ de degré **au plus $n$** passant par $n+1$ points d'interpolation $(x_i, y_i)_{0 \leq i \leq n}$, où $x_i \neq x_j$ pour $i \neq j$.

**Définition (Interpolant)**

Si l'on suppose que les ordonnées $y_i$ proviennent d'une fonction $f$, c'est-à-dire $y_i = f(x_i)$, alors l'unique polynôme $p_n$ passant par les points $(x_i, f(x_i))$ est appelé **l'interpolant** de $f$.

**Attention !**

L'interpolant de $f$ passant par $n + 1$ points d'interpolation peut être un polynôme de degré inférieur à $n$ (exemple: imaginez les ordonnées $f(x_i)$ alignées sur une droite !).

On a donc unicité du polynôme d'interpolation, mais a-t-on toujours existence de ce polynôme ?

$\rightarrow$ S'obtient par construction.

On a $n+1$ points $(x_i, y_i)$, $0 \le i \le n$, on souhaite trouver $p_n$ tel que

$$\begin{cases} p_n(x_0) = a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_n x_0^n = y_0 \\ p_n(x_1) = a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_n x_1^n = y_1 \\ \qquad \vdots \qquad\qquad\qquad\qquad \vdots \\ p_n(x_n) = a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_n x_n^n = y_n \end{cases}$$

Cela revient à trouver $(a_0, a_1, \ldots, a_n)$ tels que

$$\underbrace{\begin{pmatrix} 1 & x_0 & x_0^2 & x_0^3 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & x_1^3 & \ldots & x_1^n \\ 1 & x_2 & x_2^2 & x_2^3 & \ldots & x_2^n \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 & \ldots & x_n^n \end{pmatrix}}_{V} \underbrace{\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}}_{a} = \underbrace{\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}}_{y}$$

Cela est donc équivalent à résoudre le système $Va = y$. La matrice $V$ est appelée **matrice de Vandermonde**.

Quelques propriétés de la matrice de Vandermonde:

- La matrice de Vandermonde est inversible si et seulement si les abscisses sont distinctes, *i.e* $x_i \neq x_j$ pour $i \neq j$.

  $\rightarrow$ Le polynôme d'interpolation existe seulement si la matrice $V$ est inversible, donc **quand les abscisses $x_i$ sont distinctes**,
- Le conditionnement de la matrice augmente fortement quand $n$ augmente,
- La résolution se comporte mal quand les abscisses sont proches ou petites,
- Si le nombre de points d'interpolation ou leurs positions changent, il faut tout recommencer !

**Conclusion:** Cette approche est utile uniquement du point de vue théorique, car elle donne une condition d'existence du polynôme d'interpolation. On ne l'utilisera pas numériquement.

Première approche plus efficace : **l'interpolation de Lagrange**

Étant donné $(n+1)$ points $(x_i, y_i)$, $0 \leq i \leq n$. Supposons que l'on soit capable de construire $n+1$ polynômes $L_i(x)$ de **degré $\boldsymbol{n}$**, tels que

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

Comme toute combinaison linéaire de polynômes de degré $n$ est un polynôme de degré au plus $n$, le polynôme

$$L(x) = y_0 L_0(x) + y_1 L_1(x) + \cdots + y_n L_n(x) = \sum_{i=0}^{n} y_i L_i(x)$$

est un polynôme de degré au plus $n$. De plus, $\forall\, 0 \leq j \leq n$

$$\begin{aligned} L(x_j) &= y_0 L_0(x_j) + y_1 L_1(x_j) + \cdots + y_j L_j(x_j) + \cdots + y_n L_n(x_j) \\ &= 0 + 0 + \cdots + 0 + y_j + 0 + \cdots + 0 = y_j \end{aligned}$$

Le polynôme $L$ et donc de degré $n$ et passe par tous les points d'interpolation.

$\rightarrow$ $L$ est le polynôme recherché

Comment construire les polynômes $L_i$, $0 \leq i \leq n$ ?

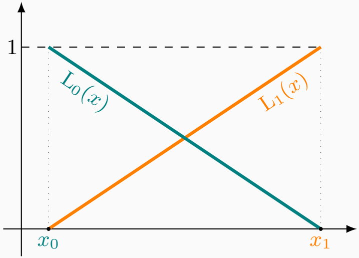
- $n = 1 \rightarrow$ deux points

On cherche deux polynômes de degré 1, $L_0$ et $L_1$ passant par les deux points $(x_0, y_0)$ et $(x_1, y_1)$ et vérifiant

$$\begin{cases} \mathrm{L}_0(x_0) = 1 \\ \mathrm{L}_0(x_1) = 0 \end{cases} \qquad \begin{cases} \mathrm{L}_1(x_0) = 0 \\ \mathrm{L}_1(x_1) = 1 \end{cases}$$

Posons

$$L_0(x) = \frac{(x - x_1)}{(x_0 - x_1)} \text{ et } L_1(x) = \frac{(x - x_0)}{(x_1 - x_0)}.$$




Alors,

$$L_0(x_0) = \frac{(x_0 - x_1)}{(x_0 - x_1)} = 1,$$

$$L_0(x_1) = \frac{(x_1 - x_1)}{(x_0 - x_1)} = 0.$$

De même, $L_1(x_0) = 0$, $L_1(x_1) = 1$.

Ainsi, le polynôme d'interpolation de degré 1 passant par $(x_0, y_0)$ et $(x_1, y_1)$ est

$$\begin{aligned} L(x) &= y_0 L_0(x) + y_1 L_1(x) \\ &= y_0 \frac{(x - x_1)}{(x_0 - x_1)} + y_1 \frac{(x - x_0)}{(x_1 - x_0)} \end{aligned}$$

- $n = 2 \rightarrow$ trois points

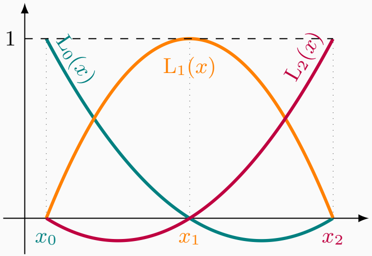
On cherche trois polynômes de degré 2, $L_0$, $L_1$, $L_2$ passant par les points $(x_0, y_0)$, $(x_1, y_1)$, $(x_2, y_2)$ et vérifiant

$$\begin{cases} \mathrm{L}_i(x_i) = 1 & \forall\, i = 1, 2, 3 \\ \mathrm{L}_i(x_j) = 0 & \forall\, i \neq j \end{cases} \tag{1}$$

Posons

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}, \qquad L_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)},$$

$$L_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$




Encore une fois, on a bien

$$L_0(x_0) = 1, L_0(x_1) = 0, L_0(x_2) = 0$$

$$L_1(x_0) = 0, L_1(x_1) = 1, L_1(x_2) = 0$$

$$L_2(x_0) = 0, L_2(x_1) = 0, L_2(x_2) = 1$$

Ainsi,

$$\begin{aligned} L(x) &= y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x) \\ &= y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} \end{aligned}$$

- pour $n+1$ points d'interpolation, on aura

$$L_i(x) = \prod_{\substack{j=0\\ j\neq i}}^{n} \frac{(x-x_j)}{(x_i-x_j)}, \quad i=0,\ldots,n.$$

Ces polynômes vérifient bien

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

et le polynôme d'interpolation sera alors [1]

$$p_n(x) = \sum_{i=0}^{n} y_i L_i(x).$$

Inconvénient : si l'on rajoute un point d'interpolation, il faut recalculer tous les polynômes $L_i$.

[1] voir `script_Interpolation_Lagrange.m` pour exemples numériques.

Interpolation de Newton : se base sur une écriture différente du polynôme recherché.

Lagrange: $p_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$
Newton :

$$\begin{aligned} p_n(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \dots \\ &\qquad + a_{n-1}(x - x_0)(x - x_1)(x - x_2)\dots(x - x_{n-2}) \\ &\qquad + a_n(x - x_0)(x - x_1)(x - x_2)\dots(x - x_{n-1}) \\ &= p_{n-1}(x) + a_n(x - x_0)(x - x_1)(x - x_2)\dots(x - x_{n-1}) \end{aligned}$$

On souhaite toujours que notre polynôme passe par les points d'interpolation $(x_i, y_i)$, $0 \leq i \leq n$, c'est-à-dire $p_n(x_i) = y_i$.

- $i = 0 \rightarrow\ p_n(x_0) = y_0 = a_0 \Rightarrow a_0 = y_0$
- $i = 1 \rightarrow\ p_n(x_1) = y_1 = a_0 + a_1(x_1 - x_0) \Rightarrow a_1 = \frac{y_1 - y_0}{x_1 - x_0}$
- $i = 2 \rightarrow\ p_n(x_2) = y_2 = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1)$

$$\Rightarrow a_2 = \frac{1}{(x_2 - x_0)(x_2 - x_1)}\left((y_2 - a_0) - a_1(x_2 - x_0)\right)$$

- $i = \dots$

Chaque polynôme dépend du polynôme précédent, **les coefficients $a_i$ sont liés entre eux**. On va voir comment les calculer de façon efficace (et automatique).

**Définition (Premières différences divisées)**

On appellera première différence divisée, notée $f[x_i, x_{i+1}]$ le rapport

$$f[x_i, x_{i+1}] = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

On souhaite exprimer tous les coefficients $a_i$ en fonction de différences divisées.

Regardons comment s'écrit $p_1(x)$ :
On a $a_1 = \frac{y_1 - y_0}{x_1 - x_0} = f[x_0, x_1]$, ainsi le polynôme $p_1$ s'écrit

$$p_1(x) = a_0 + a_1(x - x_0) = y_0 + f[x_0, x_1](x - x_0)$$

Et on a bien $p_1(x_0) = y_0$ et $p_1(x_1) = y_0 + \frac{y_1 - y_0}{\cancel{x_1 - x_0}}(\cancel{x_1 - x_0}) = y_1$.
$\rightarrow p_1$ est l'unique polynôme d'interpolation de degré 1 passant par les points $(x_0, y_0)$ et $(x_1, y_1)$.

Regardons ce qu'il se passe pour $p_2$

$$\begin{aligned} p_2(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) \\ &= y_0 + f[x_0, x_1](x - x_0) + a_2(x - x_0)(x - x_1) \\ &= p_1(x) + a_2(x - x_0)(x - x_1) \end{aligned}$$

On a vu

$$\begin{aligned} a_2 &= \frac{1}{(x_2 - x_0)(x_2 - x_1)} \left((y_2 - a_0) - a_1(x_2 - x_0)\right) \\ &= \frac{1}{(x_2 - x_0)(x_2 - x_1)} \left((y_2 - y_0) - \frac{y_1 - y_0}{x_1 - x_0}(x_2 - x_0)\right) \end{aligned}$$

que l'on peut également écrire (après quelques calculs)

$$a_2 = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

Cette dernière forme nous amène à définir les énièmes différences divisées.

### Définition (Énièmes différences divisées)

On appelle deuxième différence divisée de la fonction $f$, notée $f[x_i, x_{i+1}, x_{i+2}]$, le rapport

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}$$

Plus généralement, la $p$-ième différence divisée de $f$, notée $f[x_i, x_{i+1}, \ldots, x_{i+p}]$ est définie par

$$f[x_i, x_{i+1}, \ldots, x_{i+p}] = \frac{f[x_{i+1}, x_{i+2}, \ldots, x_{i+p}] - f[x_i, x_{i+1}, \ldots, x_{i+p-1}]}{x_{i+p} - x_i}$$

Avec cette notation, on a $a_2 = f[x_0, x_1, x_2]$ et donc

$$p_2(x) = y_0 + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1)$$

Encore une fois, on a $p_2(x_0) = y_0, p_2(x_1) = y_1, p_2(x_2) = y_2$.

$\rightarrow$ $p_2$ est l'unique polynôme d'interpolation de degré 2 passant par les trois points $(x_0, y_0), (x_1, y_1), (x_2, y_2)$.

**Remarque(s)**

$$\begin{aligned} p_2(x) &= y_0 + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &= p_1(x) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \end{aligned}$$

$p_2$ s'obtient par l'ajout d'un terme de degré 2 à $p_1$. Cela veut dire que si l'on a déjà calculé $p_1$ et que l'on ajoute un point d'interpolation (passant de 2 à 3), on peut facilement obtenir $p_2$.

On aurait envie de dire

$$\begin{aligned} p_3(x) = y_0 &+ f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &+ f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2) \end{aligned}$$

**Et l'on peut !** Ce que venons de voir pour $p_1$ et $p_2$ se généralise aux polynômes de degré $n$.

Le théorème ci-dessous formalise ce que nous venons de voir [2]

**Théorème (Interpolation de Newton)**

L'unique polynôme de degré $n$ passant par $n+1$ points d'interpolation $(x_i, y_i)$, $0 \leq i \leq n$ est donné par

$$\begin{aligned} p_n(x) &= a_0 N_0(x) + a_1 N_1(x) + \cdots + a_n N_n(x) \\ &= p_{n-1}(x) + a_n N_n(x) \end{aligned}$$

avec $N_0(x) = 1$, $N_i(x) = (x - x_0)(x - x_1) \dots (x - x_{i-1})$, $1 \leq i \leq n$ et

$$a_0 = f[x_0] = f(x_0),\ a_i = f[x_0, x_1, x_2, \dots, x_i],\ 1 \leq i \leq n$$

Les polynômes $N_i$ sont les **polynômes de Newton**.

Le calcul des différences divisées $f[x_0, x_1, x_2, \dots, x_i]$ peut être pénible si l'on s'y prend mal. On va voir une façon astucieuse et efficace de les calculer.

[2] voir `script_Interpolation_Newton.m` pour exemples numériques.

On construit une **table des différences divisées.** Calcul colonne par colonne. Exemple avec 4 points d'interpolation:

| Table de différences divisées | | | | |
|---|---|---|---|---|
| $x_i$ | $f(x_i)$ | $f[x_i, x_{i+1}]$ | $f[x_i, x_{i+1}, x_{i+2}]$ | $f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$ |
| $x_0$ | $f(x_0)$ | | | |
| | | $f[x_0, x_1]$ | | |
| $x_1$ | $f(x_1)$ | | $f[x_0, x_1, x_2] = \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}$ | |
| | | $f[x_1, x_2]$ | | $f[x_0, x_1, x_2, x_3]$ |
| $x_2$ | $f(x_2)$ | | $f[x_1, x_2, x_3]$ | |
| | | $f[x_2, x_3]$ | | |
| $x_3$ | $f(x_3)$ | | | |

✓ Il n'est pas nécessaire de placer les points d'interpolation par abscisse croissante.

La diagonale de la table correspond aux coefficients $a_i$ du polynôme d'interpolation de Newton.

$$p_3(x) = y_0 + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2)$$

L'ordre des points n'a pas d'importance, par unicité du polynôme on doit obtenir le même résultat **à la fin.**

| Table de différences divisées | | | | |
|---|---|---|---|---|
| $x_i$ | $f(x_i)$ | $f[x_i, x_{i+1}]$ | $f[x_i, x_{i+1}, x_{i+2}]$ | $f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$ |
| $x_0$ | $f(x_0)$ | | | |
| | | $f[x_0, x_1]$ | | |
| $x_1$ | $f(x_1)$ | | $f[x_0, x_1, x_2]$ | |
| | | $f[x_1, x_2]$ | | $f[x_0, x_1, x_2, x_3]$ |
| $x_2$ | $f(x_2)$ | | $f[x_1, x_2, x_3]$ | |
| | | $f[x_2, x_3]$ | | |
| $x_3$ | $f(x_3)$ | | | |

$$
\begin{aligned}
p_3(x) &= y_3 + f[x_2, x_3](x - x_3) + f[x_1, x_2, x_3](x - x_3)(x - x_2) + f[x_0, x_1, x_2, x_3](x - x_3)(x - x_2)(x - x_1) \\
&= y_0 + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2)
\end{aligned}
$$

Quelques observations/conclusions:

- Si l'on utilise tous les points d'interpolation, on a unicité du polynôme (mais les polynômes intermédiaires ne seront pas forcément les mêmes).
- Si l'on ajoute un point d'interpolation (augmentant de 1 le degré du polynôme d'interpolation), **on ne recommence pas tout**. Il suffit de rajouter une ligne dans notre table des différences divisées, et de rajouter un terme de degré $n+1$ à notre polynôme de degré $n$ déjà calculé.
- Par unicité du polynôme passant par les $n+1$ points d'interpolation, que vous utilisiez les polynômes de Lagrange ou les polynômes de Newton, vous devez obtenir le même résultat final !

Comment évaluer l'erreur commise en approximant une fonction $f$ par son interpolant $p_n$ ?

$\rightarrow$ Étude de l'erreur d'interpolation

L'erreur d'interpolation $E_n$ est définie par

$$E_n(x) = f(x) - p_n(x)$$

Puisque $p_n(x_i) = y_i = f(x_i)$, il est clair que l'erreur d'interpolation est nulle aux points d'interpolation ($E_n(x_i) = 0$) !

On cherche à évaluer l'erreur pour $x$ «quelconque» dans l'intervalle $[x_0, x_n]$.

Dans la suite, on suppose $x_0 < x_1 < \cdots < x_n$.

Soit $x^*$ tel que $x^* \neq x_i$, $i = 0, \ldots, n$ et $x^* \in [x_0, x_n]$.
Considérons l'interpolation de Newton et ajoutons le point $(x^*, f(x^*))$ à notre table de différences divisées. Par définition de l'interpolation on a

$$f(x^*) = p_{n+1}(x^*) = p_n(x^*) + f[x_0, x_1, \ldots, x_n, x^*](x - x_0) \ldots (x - x_n)$$

Cela nous donne

$$E_n(x^*) = f(x^*) - p_n(x^*) = f[x_0, x_1, \ldots, x_n, x^*](x - x_0) \ldots (x - x_n)$$

Si je sais évaluer $f[x_0, x_1, \ldots, x_n, x^*]$ alors je connais l'erreur d'interpolation.

**Problème:** on ne connaît pas $f(x^*)$ ! on ne peut donc pas calculer la différence divisée.

**Solution:** il est possible de montrer que pour $x^* \in [x_0, x_n]$ (et en supposant $f$ suffisamment régulière)

$$f[x_0, x_1, \ldots, x_n, x^*] = \frac{f^{(n+1)}(\xi_{x^*})}{(n+1)!}, \text{ pour } \xi_{x^*} \in [x_0, x_n]$$

On caractérise donc l'erreur en fonction de la dérivée de la fonction $f$.

**Théorème (Erreur d'interpolation)**

Soit $x_0 < x_1 < \cdots < x_n$ les abscisses des points d'interpolation. Si $f$ est $n+1$ fois dérivable sur $[x_0, x_n]$, alors $\forall\, x \in [x_0, x_n], \exists\, \xi_x \in [x_0, x_n]$ tel que

$$E_n(x) = f(x) - p_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}(x-x_0)(x-x_1)\dots(x-x_n)$$

- Par unicité du polynôme, l'erreur d'interpolation est la même que vous utilisiez les polynômes de Newton ou Lagrange (ou même Vandermonde).
- En pratique, $\xi_x$ n'est pas connu (dans la vraie vie, pas plus que $f$ !)
- Comme pour les développements de Taylor, dans le cas ou l'on connait la fonction $f$, on a recourt à une majoration de la dérivée,

$$|E_n(x)| \leq \frac{1}{(n+1)!} \max_{x_0 \leq \xi_x \leq x_n} |f^{(n+1)}(\xi_x)||(x-x_0)\dots(x-x_n)|$$

- Si on ne connaît pas la fonction $f$, la majoration est impossible. Dans ce cas, si on connait un point d'interpolation $(x_{n+1}, f(x_{n+1}))$ supplémentaire, on fait l'approximation suivante

$$f[x_0, x_1, \ldots, x_n, x_{n+1}] \approx \frac{f^{(n+1)}(\xi_x)}{(n+1)!}$$

et alors

$$E_n(x) \approx f[x_0, \ldots, x_n, x_{n+1}](x - x_0)\ldots(x - x_n)$$

$$\Rightarrow \boldsymbol{E_n(x) \approx p_{n+1}(x) - p_n(x)}$$

+ : facile à calculer,

- : pas toujours d'une grande précision...

- Si l'on peut choisir les points d'interpolation $x_i$, on a tout intérêt à les prendre **proche** de l'abscisse $x$ où l'on souhaite interpoler notre fonction.

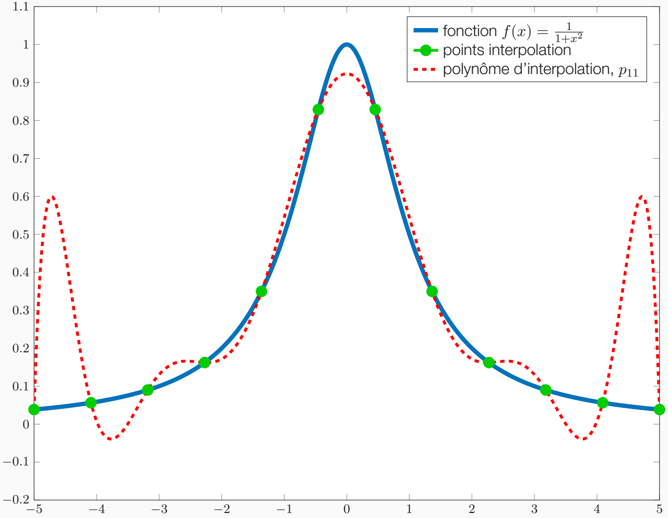
De manière général, le terme d'erreur $E_n$ est un polynôme de degré $n+1$, **pouvant fortement osciller**.

$\rightarrow$ Phénomène de Runge : pour certaines fonctions, l'erreur augmente quand le nombre de points d'interpolation augmente.

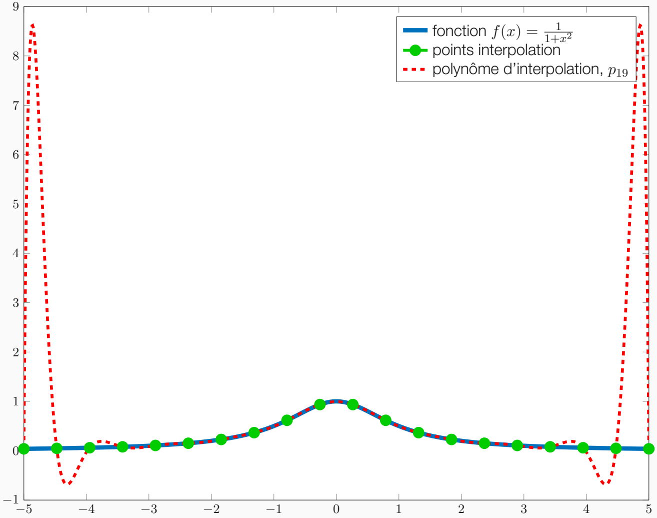
Pour illustrer le phénomène, considérons

$$f(x) = \frac{1}{1+x^2}$$

et regardons le comportement de son polynôme d'interpolation $p_n$, pour différentes valeurs de $n$.




Phénomène de Runge, $n = 11$ (12 points d'interpolation).




Phénomène de Runge, $n = 19$ (20 points d'interpolation).

$\rightarrow$ **L'erreur augmente** avec le nombre de points !

Conclusion: L'interpolation de Lagrange/Newton n'est pas adaptée à toutes les fonctions $f$.

$\rightarrow$ Il est parfois peu prudent d'utiliser un grand nombre de point si l'on ne connaît pas le comportement de la fonction.

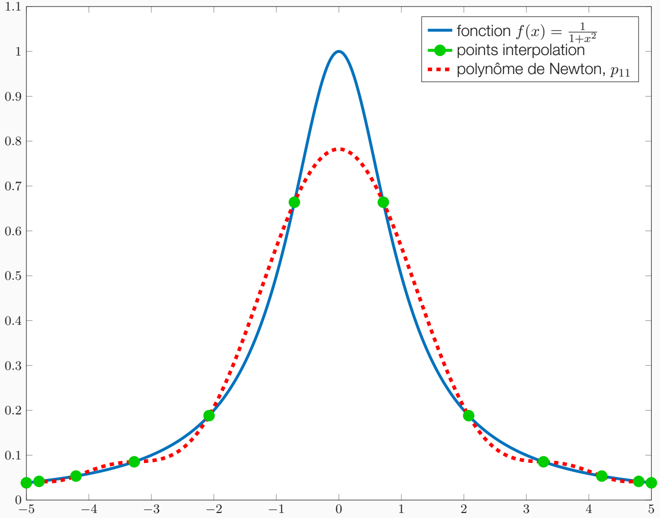
**Solution partielle:** mieux positionner les abscisses des points d'interpolation, de sorte à minimiser l'expression

$$|(x - x_0)(x - x_1)\dots(x - x_n)|$$

apparaissant dans le terme d'erreur $E_n$. Comment ?
$\rightarrow$ Abscisses de Tchebychev

$$x_i = \frac{a+b}{2} - \frac{b-a}{2}\cos\left(\frac{i\pi}{n}\right),\ i = 0, \dots, n.$$

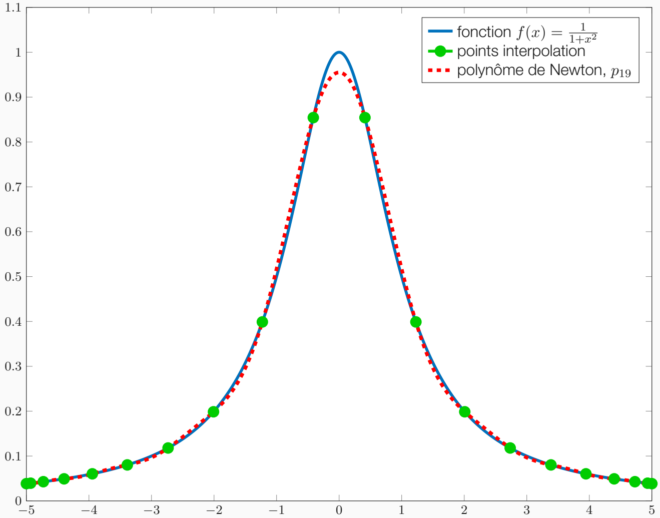
L'intervalle $[a, b]$ étant l'intervalle sur lequel on cherche à interpoler la fonction.




Phénomène de Runge, $n = 11$ (12 points d'interp.), abscisses de Tchebychev.

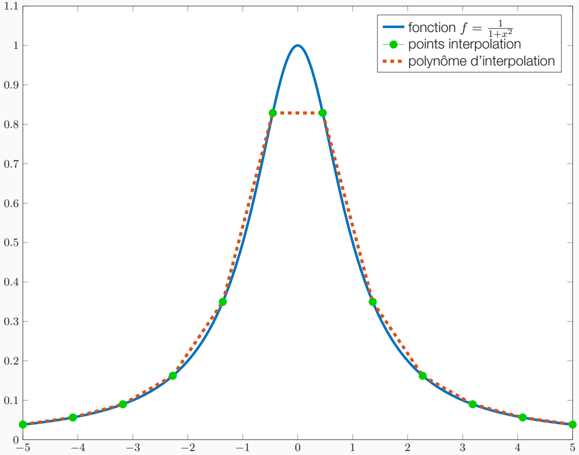
$\rightarrow$ Toujours quelques oscillations présentent...

...que l'on parvient a atténuer en augmentant le nombre de points d'interpolation.




Phénomène de Runge, $n = 19$ (20 points d'interp.), abscisses de Tchebychev.

**Autre solution:** interpolation linéaire par morceaux [3] - on construit un polynôme d'interpolation de degré 1 sur chaque sous-intervalle $[x_i, x_{i+1}]$,




Interpolation linéaire par morceaux, 8 points d'interpolation

[3] voir `script_InterpolationLinParMorceaux.m` pour exemples numériques.

Avantage de l'interpolation linéaire par morceaux:

- Évite l'utilisation de polynômes de degrés élevés (et donc le phénomène de Runge, même avec des abscisses équi-distantes). Au lieu d'avoir une erreur d'interpolation portant sur $n+1$ points, on a une erreur d'interpolation sur chaque sous-intervalle $[x_i, x_{i+1}]$.

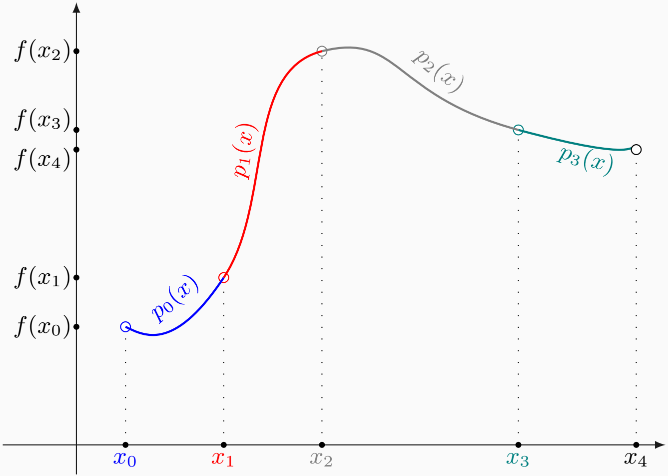
Inconvénient majeur:

- Le polynôme d'interpolation est **continu mais n'est pas dérivable aux abscisses d'interpolation $\boldsymbol{x_i}$**.

On va garder l'idée d'interpolation par morceaux, mais en utilisant sur chaque sous-intervalle un polynôme d'un degré nous garantissant un polynôme d'interpolation suffisamment «lisse».

Polynôme de degré 2 $\rightarrow$ on peut imposer la continuité des dérivées premières aux $x_i$.
Polynôme de degré 3 $\rightarrow$ on peut imposer la continuité des dérivées premières et secondes aux $x_i$.

C'est le principe de base des **splines cubiques**: utiliser des polynômes de degré 3 sur chaque sous-intervalle, et les «recoller» adéquatement.




Recollement de 4 polynômes de degré 3

Soit $(x_i, y_i = f(x_i))$, $0 \leq i \leq n$, les $n+1$ points d'interpolation.

**Idée générale des splines cubiques**:
Réaliser une interpolation cubique sur les $n$ sous-intervalles $[x_i, x_{i+1}]$ en imposant **la continuité des dérivées premières et secondes.** Soit

$$S(x) = \begin{cases} p_0(x) & x \in [x_0, x_1] \\ p_1(x) & x \in [x_1, x_2] \\ \vdots & \vdots \\ p_{n-1}(x) & x \in [x_{n-1}, x_n] \end{cases}$$

la spline recherchée. On cherche à définir sur chaque sous intervalle un polynôme de degré 3 sous la forme

$$p_i(x) = f_i + f_i'(x - x_i) + \frac{f_i''}{2}(x - x_i)^2 + \frac{f_i'''}{6}(x - x_i)^3, \quad 0 \leq i \leq n-1$$

où $f_i$, $f_i'$, $f_i''$ et $f_i'''$ sont des constantes (et **ne sont pas les dérivées de $f$ !**)

✓ L'écriture des $p_i$ sous la forme d'un développement de Taylor permet d'interpréter plus facilement les coefficients $f_i$, $f_i'$, $f_i''$ et $f_i'''$. On a en effet

$$f_i = p_i(x_i), \quad f_i' = p_i'(x_i), \quad f_i'' = p_i''(x_i), \quad f_i''' = p_i'''(x_i)$$

**On veut** :

- $S(x_i) = y_i$, pour $0 \leq i \leq n \rightarrow \; n+1$ conditions,
- $S(x)$ continue aux $n-1$ nœuds internes, c'est-à-dire $p_i(x_{i+1}) = p_{i+1}(x_{i+1})$, $1 \leq i \leq n-1$ $\rightarrow \; n-1$ conditions,
- $S'(x)$ et $S''(x)$ continues aux $n-1$ nœuds internes, c'est-à-dire

$$p_i'(x_{i+1}) = p_{i+1}'(x_{i+1}),\; 1 \leq i \leq n-1 \rightarrow \; n-1 \text{ conditions}$$
$$p_i''(x_{i+1}) = p_{i+1}''(x_{i+1}),\; 1 \leq i \leq n-1 \rightarrow \; n-1 \text{ conditions}$$

**Bilan** : On a $4n-2$ équations pour $4n$ inconnues ($n$ polynômes, chaque polynôme a $4$ inconnues) $\rightarrow$ Système sous-déterminé.

Il va exister plusieurs splines cubiques pour les mêmes points d'interpolation.

**Objectif:** On va exprimer toutes les inconnues du problème en fonction des coefficients $f_i''$.

$\rightarrow$ On va ramener la construction de la spline à la résolution d'un système matriciel $n+1 \times n+1$, où les inconnues seront les $f_i'', i = 0, \cdots, n$.

**Remarque(s)**

On introduit une variable supplémentaire : $\boldsymbol{f_n''}$, la dérivée seconde de $p_{n-1}$ au noeud $x_n$ (en pratique, cette valeur sera imposée).

$$f_n'' = p_{n-1}''(x_n)$$

Commençons par exprimer les conditions en fonctions des $f_i$, $f_i'$, $f_i''$ et $f_i'''$, et voyons ensuite comment tout exprimer en fonction de $f_i''$

**Expressions des conditions** en fonction des $f_i$, $f_i'$, $f_i''$ et $f_i'''$ : On note $h_i = x_{i+1} - x_i$, pour $0 \leq i \leq n-1$.

- À la première extrémité $x_i$ de chaque intervalle $[x_i, x_{i+1}]$, le polynôme $p_i(x)$ passe au point $(x_i, f(x_i))$. On a
$$p_i(x_i) = f_i = f(x_i), \text{ pour } i = 0, 1, \cdots, n-1. \tag{2}$$
- L'équation de continuité en $x_{i+1}$ nous donne que $p_i(x_{i+1}) = p_{i+1}(x_{i+1}) = f(x_{i+1})$. On en déduit une condition sur $f_i'$
$$f_i' = f[x_i, x_{i+1}] - \frac{h_i f_i''}{3} - \frac{h_i f_{i+1}''}{6}, \text{ pour } i = 0, 1, \cdots, n-1 \tag{3}$$
- La continuité de la dérivée première aux $(n-1)$ nœuds intérieurs ($p_i'(x_{i+1}) = p_{i+1}'(x_{i+1})$) nous donne
$$f_{i+1}' = f_i' + f_i'' h_i + \frac{f_i'''}{2} h_i^2, \text{ pour } i = 0, 1, \cdots, n-2. \tag{4}$$
- La continuité de la dérivée seconde aux $(n-1)$ nœuds intérieurs ($p_i''(x_{i+1}) = p_{i+1}''(x_{i+1})$) nous donne
$$f_i''' = \frac{f_{i+1}'' - f_i''}{h_i} \text{ pour } i = 0, 1, \cdots, n-2. \tag{5}$$

En exprimant la condition (4) à l'aide des conditions (3) et (5), nous obtenons un système en $f_i''$, soit pour $1 \leq i \leq n-1$:

$$\frac{h_{i-1}}{h_{i-1}+h_i} f_{i-1}'' + 2f_i'' + \frac{h_i}{h_{i-1}+h_i} f_{i+1}'' = 6f\left[x_{i-1}, x_i, x_{i+1}\right] \tag{6}$$

$$\begin{pmatrix} ? & ? & ? & \dots & ? & ? \\ \frac{h_0}{h_1+h_0} & 2 & \frac{h_1}{h_1+h_0} & 0 & \dots & 0 \\ 0 & \frac{h_1}{h_1+h_2} & 2 & \frac{h_2}{h_1+h_2} & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \frac{h_{n-2}}{h_{n-2}+h_{n-1}} & 2 & \frac{h_{n-1}}{h_{n-2}+h_{n-1}} \\ ? & ? & ? & \dots & ? & ? \end{pmatrix} \begin{pmatrix} f_0'' \\ f_1'' \\ \vdots \\ \vdots \\ f_{n-1}'' \\ f_n'' \end{pmatrix} = \begin{pmatrix} ? \\ 6f[x_0, x_1, x_2] \\ \vdots \\ \vdots \\ 6f[x_{n-2}, x_{n-1}, x_n] \\ ? \end{pmatrix}$$

C'est un système **tridiagonal** dont **la première ligne et la dernière ligne ne sont pas définies**. On a besoin de deux conditions supplémentaires pour remplir ces lignes, faisant intervenir $f_0''$ et $f_n''$.

Plusieurs façons de faire. On peut prendre :

- $f_0'' = a$ et $f_n'' = b \to$ On impose la valeur des dérivées secondes,
- $f_0'' = 0$ et $f_n'' = 0 \to$ Spline «naturelle»,
- $p_0'(x_0) = a$ et $p_{n-1}'(x_n) = b \to$ dérivée première imposée, cela donne en terme de $f_i''$

$$\begin{cases} 2f_0'' + f_1'' = \dfrac{6}{h_0}(f[x_0, x_1] - a) \\ f_{n-1}'' + 2f_n'' = \dfrac{6}{h_{n-1}}(b - f[x_{n-1}, x_n]) \end{cases}$$

- $p_0'''(x_1) = p_1'''(x_1)$ et $p_{n-2}'''(x_{n-1}) = p_{n-1}'''(x_{n-1}) \to$ Spline «not a knot »: on impose la continuité des dérivées troisièmes, cela donne en terme $f_i''$

$$\begin{cases} h_1 f_0'' - (h_0 + h_1) f_1'' + h_0 f_2'' = 0 \\ h_{n-1} f_{n-2}'' - (h_{n-2} + h_{n-1}) f_{n-1}'' + h_{n-2} f_n'' = 0 \end{cases}$$

- **D'autres choix sont possibles !**

Dans le cas où $h_i = h \ \forall \ i$, (points équidistants) le système (6) est simplifié

$$\frac{1}{2} f''_{i-1} + 2 f''_i + \frac{1}{2} f''_{i+1} = 6 f \left[ x_{i-1}, x_i, x_{i+1} \right]$$

Une fois les $f''_i$ déterminés, on exprime $f'_i$ et $f''_i$ en fonction des $f''_i$, cela nous donne

$$\begin{aligned} f'_i &= f[x_i, x_{i+1}] - h_i \frac{f''_i}{3} - h_i \frac{f''_{i+1}}{6} \\ f'''_i &= \frac{f''_{i+1} - f''_i}{h_i} \end{aligned}$$

Pour la spline not a knot, les conditions imposées reviennent à retirer virtuellement les points $x_1$ et $x_{n-1}$. En effet, en $x_1$, les polynômes $p_0$ et $p_1$ coïncident ainsi que leurs trois dérivées premières. Comme $p_0$ et $p_1$ sont de degrés 3, ils s'agit en fait d'un seul et même polynôme sur les deux premiers intervalles. Idem en $x_{n-1}$.

**Méthode pour calculer une spline cubique [4] :**

- Calcul des $h_i = x_{i+1} - x_i$, $0 \leq i \leq n-1$.
- Calcul des deuxièmes différences divisées $f[x_{i-1}, x_i, x_{i+1}]$, $1 \leq i \leq n-1$.
- Résolution du système tridiagonal (6) avec la première et la dernière ligne de la matrice et du second membre adaptées aux deux conditions supplémentaires choisies. On obtient $f_i''$ pour $0 \leq i \leq n$.
- On calcule les autres coefficients :

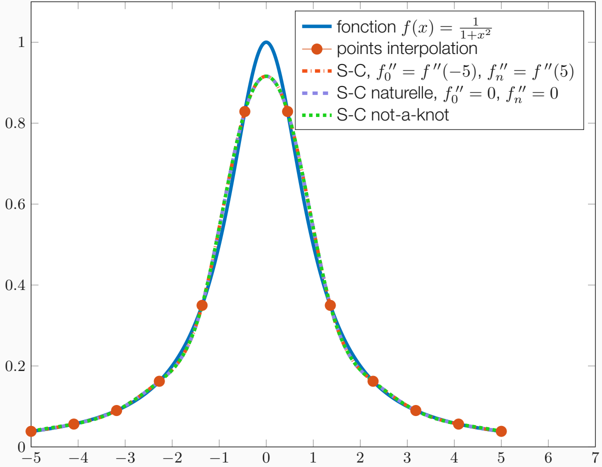
$$\begin{aligned} f_i &= f(x_i) \\ f_i' &= f\left[x_i, x_{i+1}\right] - \frac{h_i f_i''}{3} - \frac{h_i f_{i+1}''}{6} \\ f_i''' &= \frac{f_{i+1}'' - f_i''}{h_i} \end{aligned}$$

- On construit les polynômes composant la spline, pour $i = 0, \cdots, n-1$ :

$$p_i(x) = f_i + f_i'(x - x_i) + \frac{f_i''}{2}(x - x_i)^2 + \frac{f_i'''}{6}(x - x_i)^3$$

[4] voir `script_Spline_cubique.m` pour exemples numériques. Pour exécuter le script, la fonction `resolution_fi_pp.m` est nécessaire.

Retour sur la fonction $f(x) = \frac{1}{1+x^2}$




Splines cubiques, 12 points d'interpolation

$\rightarrow$ Les différents types de splines permettent d'éviter le phénomène de Runge.

**Je dois être capable de :**

- Construire un polynôme d'interpolation par les méthodes de Vandermonde, de Lagrange et de Newton,
- Connaître le terme d'erreur, comprendre son caractère oscillatoire,
- Savoir comment choisir des points pour construire une interpolation,
- Comprendre le concept de spline, connaître les particularités et savoir vérifier toutes ses propriétés (distinguer une spline d'une interpolation cubique par morceaux),
- Construire une spline cubique naturelle.