MMSN
Analyse numérique 2

Chapitre 1 :
Interpolation polynomiale
Cours 1-2-3-4-5

GM 3 Année 2022 - 2023

Contact : A. Tonnoir
antoine.tonnoir@insa-rouen.fr

Un peu de contexte…

Ce cours s'inscrit dans la continuité du cours d'analyse numérique S1 où vous avez vu (notamment) :

- des méthodes de résolution de systèmes linéaires
- des méthodes de résolution d'équations non linéaires

Un peu de contexte…

Ce cours s'inscrit dans la continuité du cours d'analyse numérique S1 où vous avez vu (notamment) :

- des méthodes de résolution de systèmes linéaires
- des méthodes de résolution d'équations non linéaires

On s'intéressera dans ce second semestre aux problèmes suivant :

- approximation polynomiale de fonctions ou de données,
- l'intégration numérique de fonctions,
- la résolution numérique d'EDO.

Un peu de contexte…

Ce cours s'inscrit dans la continuité du cours d'analyse numérique S1 où vous avez vu (notamment) :

- des méthodes de résolution de systèmes linéaires
- des méthodes de résolution d'équations non linéaires

On s'intéressera dans ce second semestre aux problèmes suivant :

- approximation polynomiale de fonctions ou de données, (Chap. 1)
- l'intégration numérique de fonctions, (Chap. 2)
- la résolution numérique d'EDO. (Chap. 3)

Quelques références

Pour ce approfondir ce cours, vous pouvez consulter :

- Calcul scientifique
  A. Quarteroni, F. Saleri, P. Gervasio

- Résolution numérique des EDO
  C. Besse

- Introduction à l'analyse numérique
  E. Hairer, G. Wanner

et bien sur le polycopié (N. Forcadel & A. Tonnoir) associé au cours !

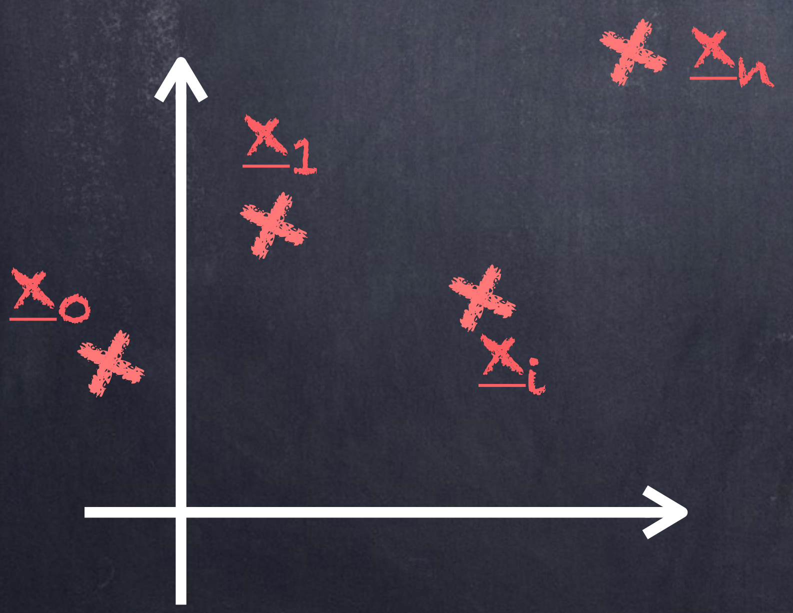
Au programme (Chapitre 1)

Objectif :

Étant donné n+1 points de coordonnées $\underline{x}_i \in \mathbb{R}^2, i \in \{0, \cdots, n\}$ déterminer P un (le ?) polynôme vérifiant

$$P(x_i) = y_i, \ \forall i \in \{0, \cdots, n\} \text{ où } \underline{x}_i = (x_i, y_i)$$

Exemples d'application :



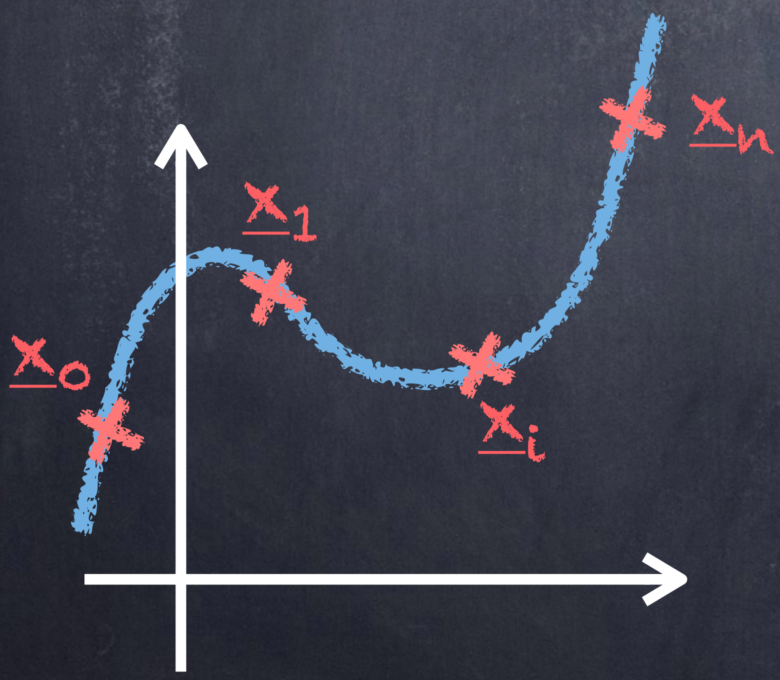
Au programme (Chapitre 1)

Objectif :

Étant donné n+1 points de coordonnées $\underline{x}_i \in \mathbb{R}^2, i \in \{0, \cdots, n\}$ déterminer P un (le ?) polynôme vérifiant

$$P(x_i) = y_i, \ \forall i \in \{0, \cdots, n\} \text{ où } \underline{x}_i = (x_i, y_i)$$

Exemples d'application :



Au programme (Chapitre 1)

Objectif :

Étant donné n+1 points de coordonnées $\underline{x}_i \in \mathbb{R}^2, i \in \{0, \cdots, n\}$ déterminer P un (le ?) polynôme vérifiant

$$P(x_i) = y_i, \ \forall i \in \{0, \cdots, n\} \text{ où } \underline{x}_i = (x_i, y_i)$$

Exemples d'application : Reconstruction de données intermédiaires

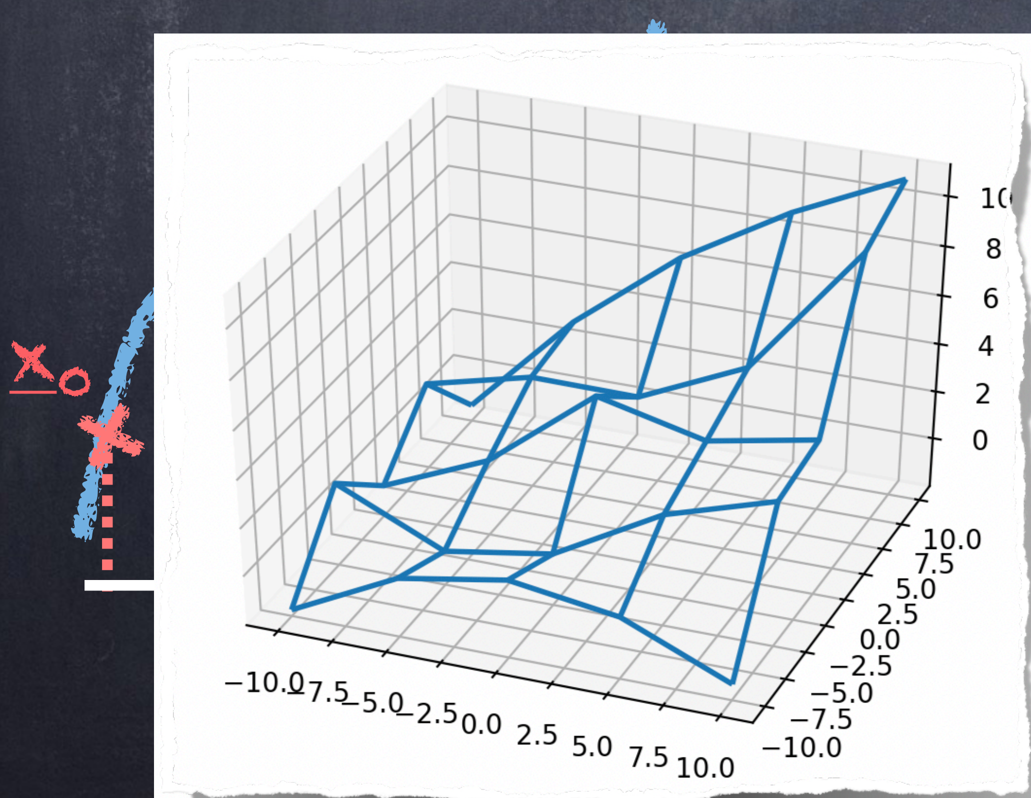
Au programme (Chapitre 1)

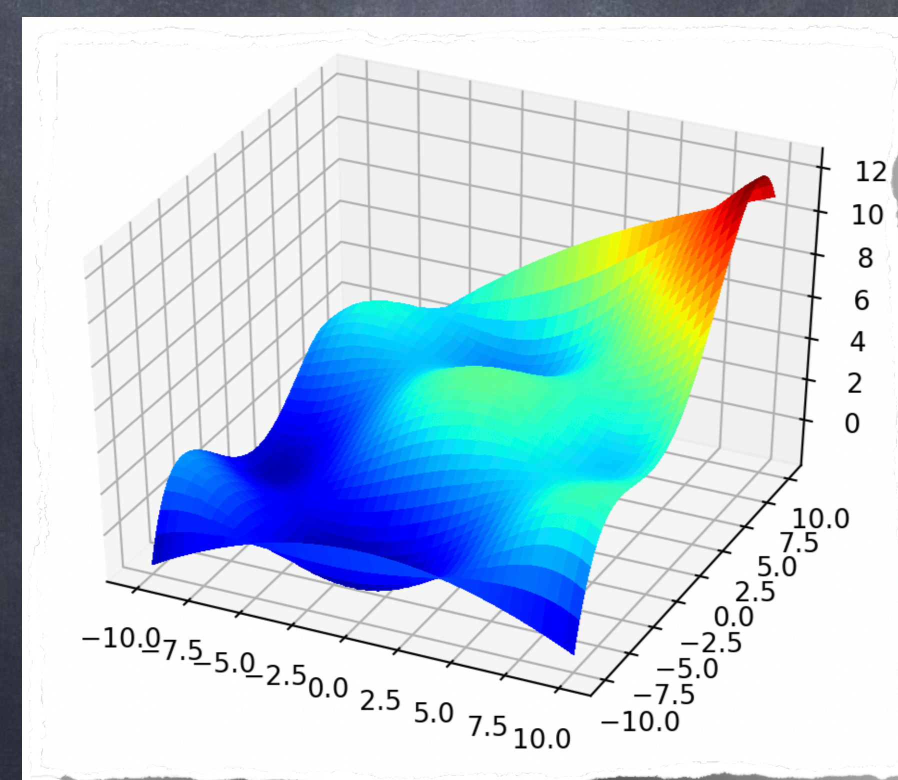
Objectif :

Étant donné n+1 points de coordonnées $\underline{x}_i \in \mathbb{R}^2, i \in \{0, \cdots, n\}$ déterminer P un (le ?) polynôme vérifiant

$$P(x_i) = y_i, \ \forall i \in \{0, \cdots, n\} \text{ où } \underline{x}_i = (x_i, y_i)$$

Exemples d'application : Reconstruction de données intermédiaires





Au programme (Chapitre 1)

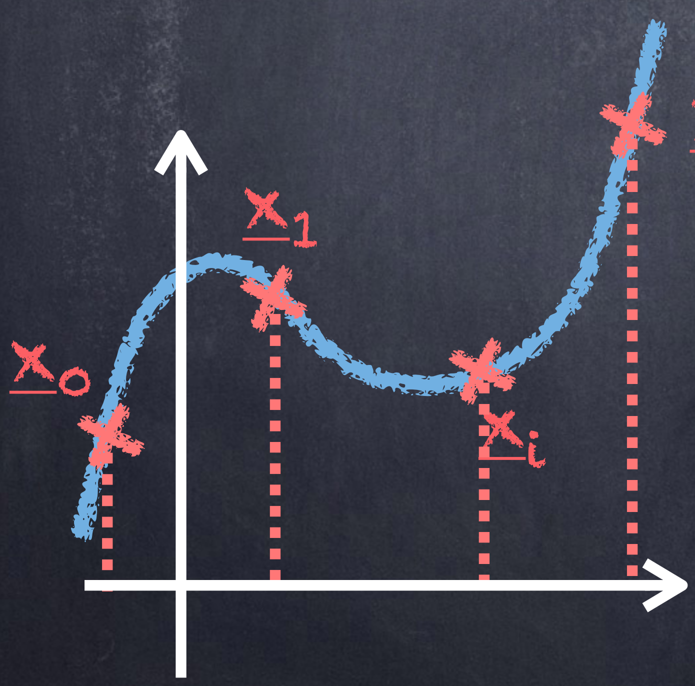
Objectif :

Étant donné n+1 points de coordonnées $\underline{x}_i \in \mathbb{R}^2, i \in \{0, \cdots, n\}$ déterminer P un (le ?) polynôme vérifiant

$$P(x_i) = y_i, \ \forall i \in \{0, \cdots, n\} \text{ où } \underline{x}_i = (x_i, y_i)$$

Exemples d'application :

- Reconstruction de données intermédiaires
- Approximation de fonction f
  $y_i = f(x_i)$



Au programme (Chapitre 1)

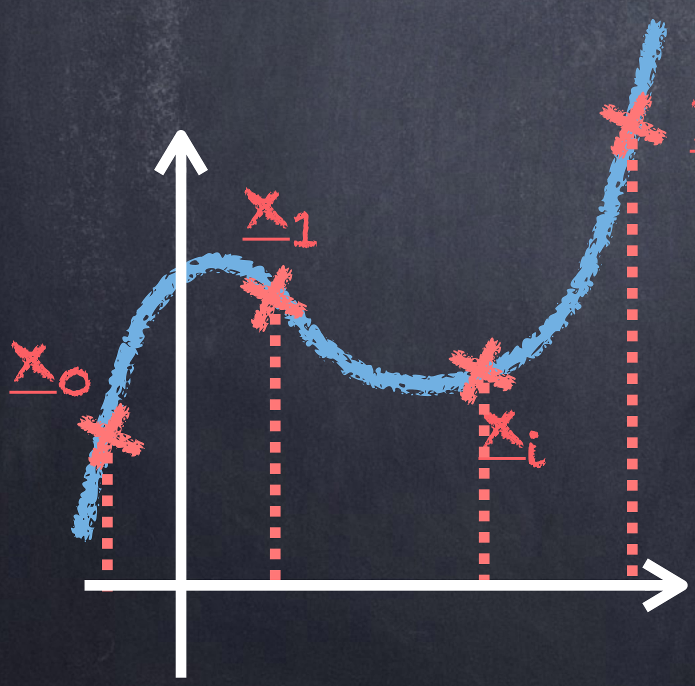
Objectif :

Étant donné n+1 points de coordonnées $\underline{x}_i \in \mathbb{R}^2, i \in \{0, \cdots, n\}$ déterminer P un (le ?) polynôme vérifiant

$$P(x_i) = y_i, \ \forall i \in \{0, \cdots, n\} \text{ où } \underline{x}_i = (x_i, y_i)$$

Exemples d'application :

- Reconstruction de données intermédiaires
- Approximation de fonction f
  $y_i = f(x_i)$
- Logiciel de DAO
  (Dessin Assisté par Ordinateur)



Au programme (Chapitre 1)

Objectif :

Étant donné n+1 points de coordonnées $\underline{x}_i \in \mathbb{R}^2, i \in \{0, \cdots, n\}$ déterminer P un (le ?) polynôme vérifiant

$$P(x_i) = y_i, \ \forall i \in \{0, \cdots, n\} \text{ où } \underline{x}_i = (x_i, y_i)$$

Plan :

I. Interpolation polynomiale et base de Lagrange

II. Évaluation du polynôme d'interpolation

III. Interpolation d'une fonction

IV. Quelques extensions

I. Interpolation polynomiale

1.1 Définition

On appelle problème d'interpolation de Lagrange le problème suivant : trouver P un polynôme vérifiant

$$P(x_i) = y_i, \quad \forall i \in \{0, \cdots, n\}$$

I. Interpolation polynomiale

1.1 Définition

On appelle problème d'interpolation de Lagrange le problème suivant : trouver P un polynôme vérifiant

$$P(x_i) = y_i, \quad \forall i \in \{0, \cdots, n\}$$

Considérons $P \in \mathbb{P}_n$ l'espace des polynômes de degré n et notons par $(\varphi_j)_{j=0,\cdots,n}$ une base. On a alors :

$$P(x_i) = y_i \quad \Leftrightarrow \quad \sum_{j=0}^{n} \alpha_j \varphi_j(x_i) = y_i$$

où les inconnues sont les coefficients α_j.

I. Interpolation polynomiale

1.1 Définition

On appelle problème d'interpolation de Lagrange le problème suivant : trouver P un polynôme vérifiant

$$P(x_i) = y_i, \ \forall i \in \{0, \cdots, n\}$$

Considérons $P \in \mathbb{P}_n$ l'espace des polynômes de degré n et notons par $(\varphi_j)_{j=0,\cdots,n}$ une base. On a alors :

$$P(x_i) = y_i \quad \Leftrightarrow \quad \begin{bmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \vdots & \vdots & & \vdots \\ \varphi_0(x_n) & \varphi_1(x_n) & \cdots & \varphi_n(x_n) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

où les inconnues sont les coefficients α_j.

I. Interpolation polynomiale

1.1 Définition

On appelle problème d'interpolation de Lagrange le problème suivant : trouver P un polynôme vérifiant

$$P(x_i) = y_i, \ \forall i \in \{0, \cdots, n\}$$

Considérons $P \in \mathbb{P}_n$ l'espace des polynômes de degré n et notons par $(\varphi_j)_{j=0,\cdots,n}$ une base. On a alors :

$$P(x_i) = y_i \quad \Leftrightarrow \quad \begin{bmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \vdots & \vdots & & \vdots \\ \varphi_0(x_n) & \varphi_1(x_n) & \cdots & \varphi_n(x_n) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Remarque :

On peut dores et déjà noter que la matrice n'est pas inversible si on a $x_i = x_j$, $j \neq i$

I. Interpolation polynomiale

1.1 Définition

On appelle problème d'interpolation de Lagrange le problème suivant : trouver P un polynôme vérifiant

$$P(x_i) = y_i, \quad \forall i \in \{0, \cdots, n\}$$

1.2 Proposition

Le problème d'interpolation de Lagrange admet une unique solution ssi la matrice de Gram G définie par $G_{ij} = \varphi_j(x_i)$ est inversible.

Ce résultat découle directement de la formulation matricielle du problème.

I. Interpolation polynomiale

1.1 Définition

On appelle problème d'interpolation de Lagrange le problème suivant : trouver P un polynôme vérifiant

$$P(x_i) = y_i, \quad \forall i \in \{0, \cdots, n\}$$

1.3 Théorème

Si les $(\varphi_j)_{j=0,\cdots,n}$ forment une base de $\mathbb{P}_n$, et $x_i \neq x_j$, $\forall j \neq i$ alors la matrice de Gram G est inversible, et donc le problème d'interpolation admet une unique solution.

Preuve : au (vrai) tableau ! ■

I. Interpolation polynomiale

1.1 Définition

On appelle problème d'interpolation de Lagrange le problème suivant : trouver P un polynôme vérifiant

$$P(x_i) = y_i, \ \forall i \in \{0, \cdots, n\}$$

1.3 Théorème

Si les $(\varphi_j)_{j=0,\cdots,n}$ forment une base de $\mathbb{P}_n$, et $x_i \neq x_j, \ \forall j \neq i$ alors la matrice de Gram G est inversible, et donc le problème d'interpolation admet une unique solution.

Remarque :

Attention, le résultat n'est pas vrai si les $(\varphi_j)_{j=0,\cdots,n}$ forment une famille libre de $\mathbb{P}_k$, $k > n$.

I. Interpolation polynomiale

Comment obtenir l'expression de P ?

- **Base canonique** : $\mathbb{P}_n = \text{vect}\{1, x, x^2, \cdots, x^n\}$

Dans ce cas, on doit résoudre le système linéaire

$$\begin{bmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

pour obtenir la décomposition $P(x) = \alpha_0 + \alpha_1 x + \cdots + \alpha_n x^n$

I. Interpolation polynomiale

Comment obtenir l'expression de P ?

- **Base canonique** : $\mathbb{P}_n = \text{vect}\{1, x, x^2, \cdots, x^n\}$

Dans ce cas, on doit résoudre le système linéaire

$$\begin{bmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

pour obtenir la décomposition $P(x) = \alpha_0 + \alpha_1 x + \cdots + \alpha_n x^n$

Difficulté :

Résoudre un système linéaire mal conditionné !

I. Interpolation polynomiale

Comment obtenir l'expression de P ?

- Base canonique : $\mathbb{P}_n = \text{vect}\{1, x, x^2, \cdots, x^n\}$

- Base de Lagrange :

Idée : Choisir les $(\varphi_j)_{j=0,\cdots,n}$ t.q. la matrice de Gram

$$G = \begin{bmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \vdots & \vdots & & \vdots \\ \varphi_0(x_n) & \varphi_1(x_n) & \cdots & \varphi_n(x_n) \end{bmatrix}$$

soit la matrice identité.

I. Interpolation polynomiale

Comment obtenir l'expression de P ?

- Base canonique : $\mathbb{P}_n = \mathrm{vect}\{1, x, x^2, \cdots, x^n\}$

- Base de Lagrange :

Idée : Choisir les $(\varphi_j)_{j=0,\cdots,n}$ t.q. la matrice de Gram

$$G = \begin{bmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \vdots & \vdots & & \vdots \\ \varphi_0(x_n) & \varphi_1(x_n) & \cdots & \varphi_n(x_n) \end{bmatrix}$$

soit la matrice identité.

Autrement dit, on cherche φ_j t.q. $\varphi_j(x_i) = \delta_{ij}$.

I. Interpolation polynomiale

Comment obtenir l'expression de P ?

- Base canonique : $\mathbb{P}_n = \text{vect}\{1, x, x^2, \cdots, x^n\}$

- Base de Lagrange :

1.4 Proposition

La famille de fonction $(L_i^n)_{i=0,\cdots,n}$ définie par

$$L_i^n(x) := \frac{\prod_{j\neq i}(x - x_j)}{\prod_{j\neq i}(x_i - x_j)}$$

forme une base de $\mathbb{P}_n$ et vérifie $L_j^n(x_i) = \delta_{ij}$.

Preuve : au (vrai) tableau !

I. Interpolation polynomiale

Comment obtenir l'expression de P ?

- Base canonique : $\mathbb{P}_n = \text{vect}\{1, x, x^2, \cdots, x^n\}$

- Base de Lagrange : $\mathbb{P}_n = \text{vect}\{L_0^n, L_1^n, \cdots, L_n^n\}$

Dans ce cas, l'expression du polynôme d'interpolation dans la base de Lagrange est directe :

$$P(x_i) = y_i \Leftrightarrow \alpha_i = y_i \quad \Rightarrow \quad P(x) = \sum_{j=0}^{n} y_j L_j^n(x)$$

I. Interpolation polynomiale

Comment obtenir l'expression de P ?

- Base canonique : $\mathbb{P}_n = \text{vect}\{1, x, x^2, \cdots, x^n\}$

- Base de Lagrange : $\mathbb{P}_n = \text{vect}\{L_0^n, L_1^n, \cdots, L_n^n\}$

Dans ce cas, l'expression du polynôme d'interpolation dans la base de Lagrange est directe :

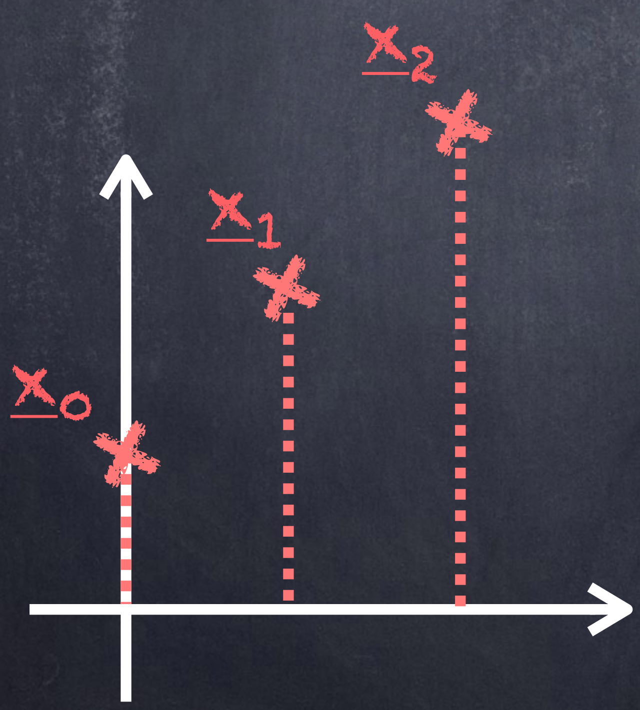
$$P(x_i) = y_i \Leftrightarrow \alpha_i = y_i \quad \Rightarrow \boxed{P(x) = \sum_{j=0}^{n} y_j L_j^n(x)}$$

I. Interpolation polynomiale

Comment obtenir l'expression de P ?

- Base canonique : $\mathbb{P}_n = \text{vect}\{1, x, x^2, \cdots, x^n\}$
- Base de Lagrange : $\mathbb{P}_n = \text{vect}\{L_0^n, L_1^n, \cdots, L_n^n\}$

Exemple :



Considérons les points

$\underline{x}_0 = (0,1)$, $\underline{x}_1 = (1,2)$, $\underline{x}_2 = (2,3)$

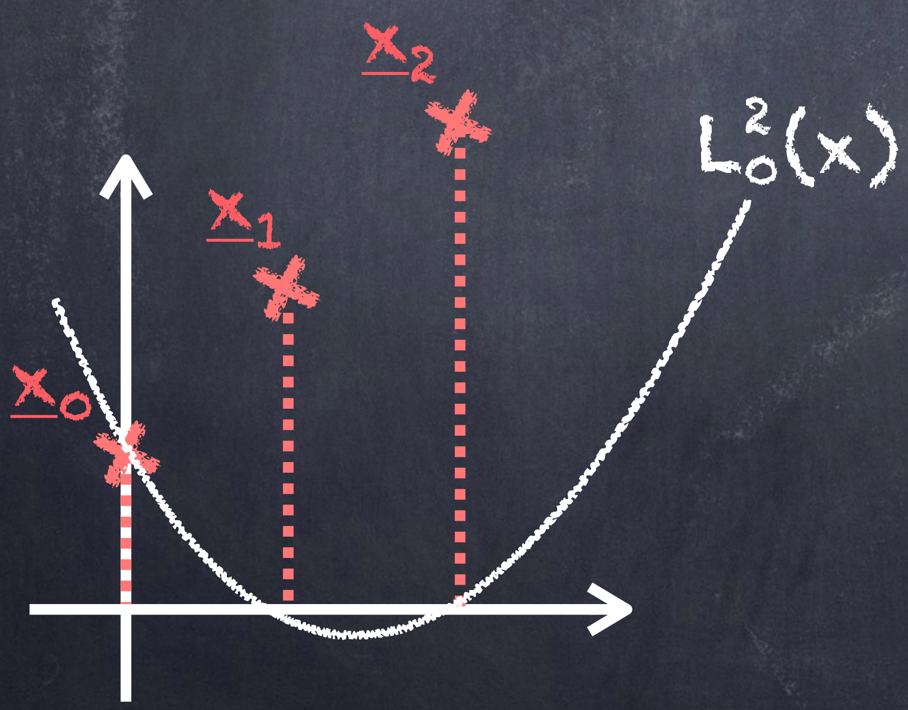
Alors :

I. Interpolation polynomiale

Comment obtenir l'expression de P ?

- <u>Base canonique</u> : $\mathbb{P}_n = \text{vect}\{1, x, x^2, \cdots, x^n\}$
- <u>Base de Lagrange</u> : $\mathbb{P}_n = \text{vect}\{L_0^n, L_1^n, \cdots, L_n^n\}$

Exemple :



Considérons les points

$\underline{x}_0 = (0, 1), \;\; \underline{x}_1 = (1, 2), \;\; \underline{x}_2 = (2, 3)$

Alors :

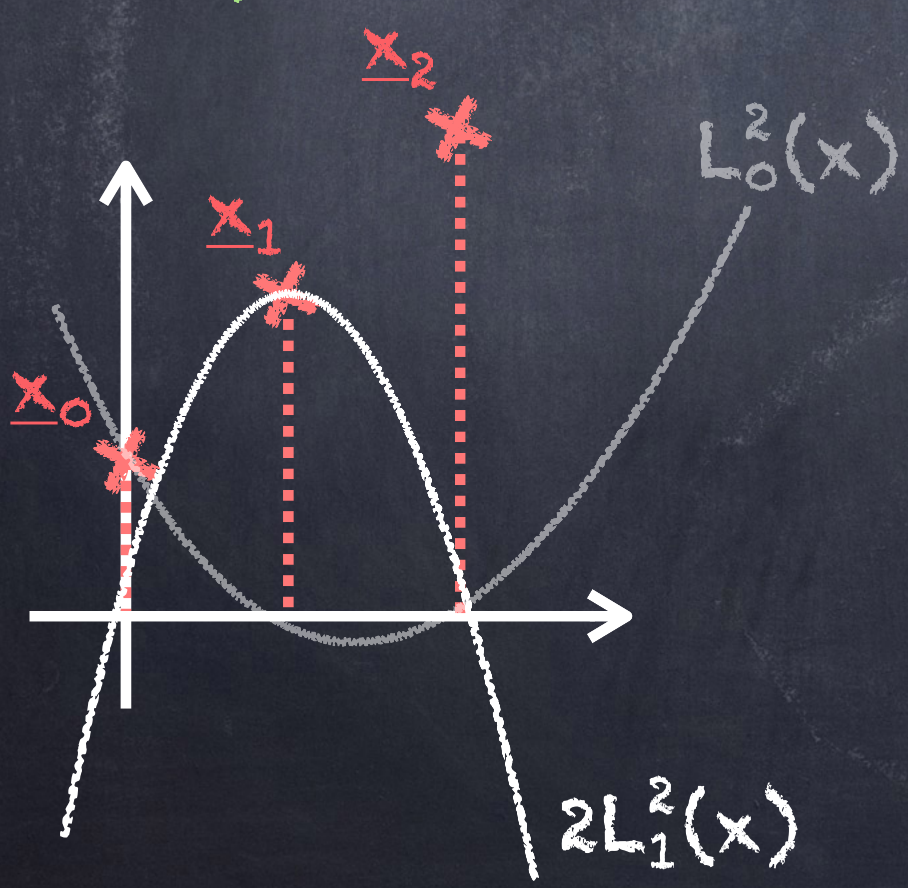
$P(x) = 1 \times L_0^2(x)$

I. Interpolation polynomiale

Comment obtenir l'expression de P ?

- <u>Base canonique</u> : $\mathbb{P}_n = \text{vect}\{1, x, x^2, \cdots, x^n\}$
- <u>Base de Lagrange</u> : $\mathbb{P}_n = \text{vect}\{L_0^n, L_1^n, \cdots, L_n^n\}$

Exemple :



Considérons les points

$\underline{x}_0 = (0,1), \ \underline{x}_1 = (1,2), \ \underline{x}_2 = (2,3)$

Alors :

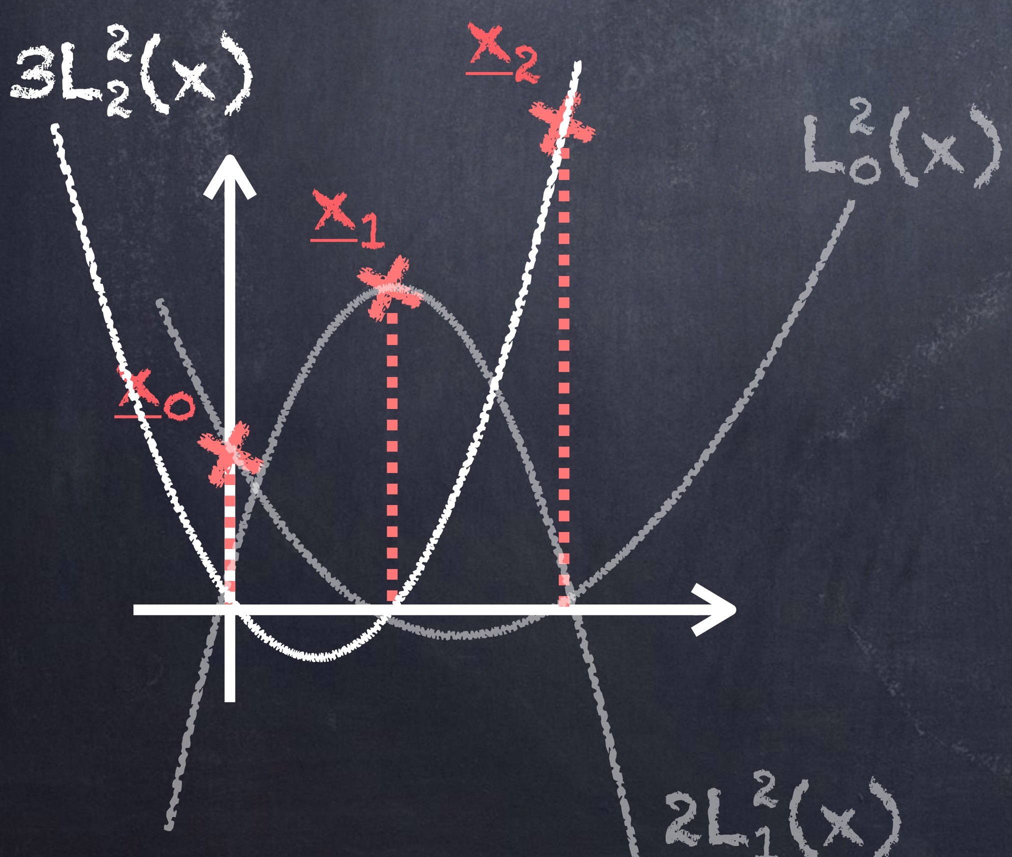
$$P(x) = 1 \times L_0^2(x) + 2 \times L_1^2(x)$$

I. Interpolation polynomiale

Comment obtenir l'expression de P ?

- Base canonique : $\mathbb{P}_n = \text{vect}\{1, x, x^2, \cdots, x^n\}$
- Base de Lagrange : $\mathbb{P}_n = \text{vect}\{L_0^n, L_1^n, \cdots, L_n^n\}$

Exemple :



Considérons les points

$\underline{x}_0 = (0,1)$, $\underline{x}_1 = (1,2)$, $\underline{x}_2 = (2,3)$

Alors :

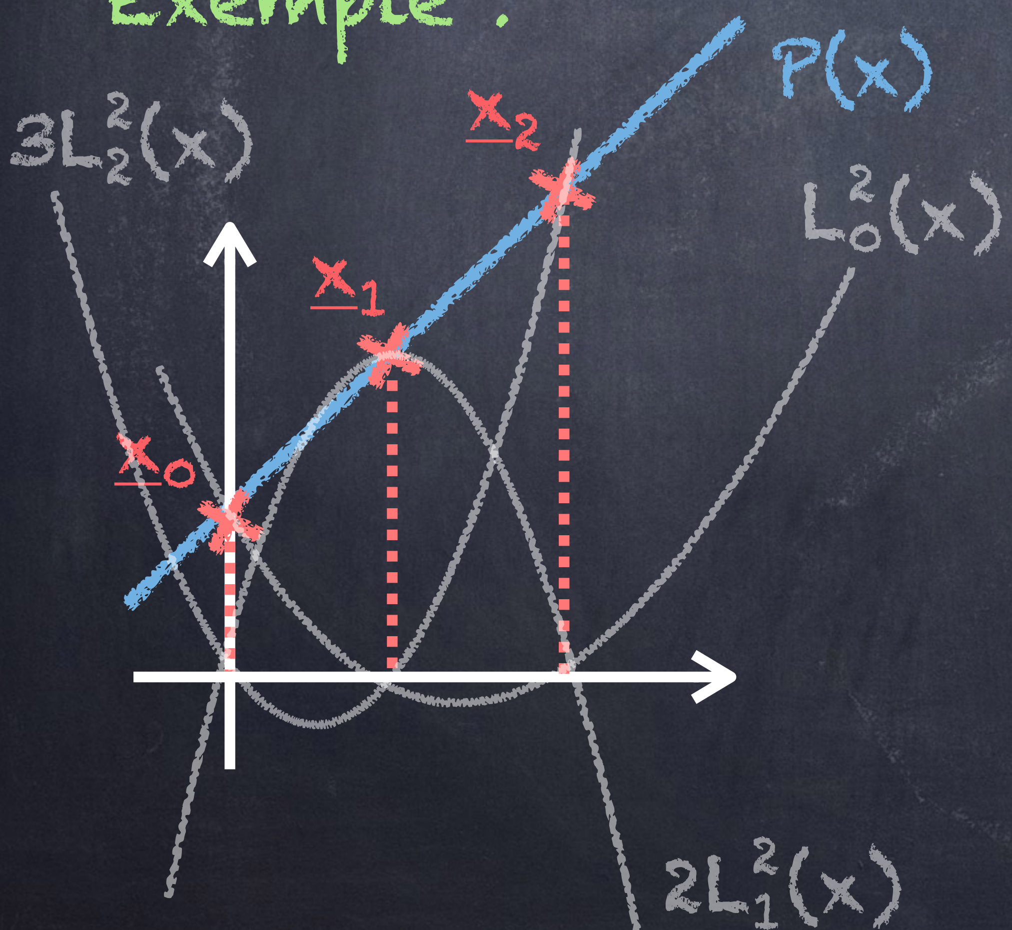
$$P(x) = 1 \times L_0^2(x) + 2 \times L_1^2(x) + 3 \times L_2^2(x)$$

I. Interpolation polynomiale

Comment obtenir l'expression de P ?

- **Base canonique** : $\mathbb{P}_n = \text{vect}\{1, x, x^2, \cdots, x^n\}$
- **Base de Lagrange** : $\mathbb{P}_n = \text{vect}\{L_0^n, L_1^n, \cdots, L_n^n\}$

Exemple :



Considérons les points

$\underline{x}_0 = (0,1)$, $\underline{x}_1 = (1,2)$, $\underline{x}_2 = (2,3)$

Alors :

$$P(x) = 1 \times L_0^2(x) + 2 \times L_1^2(x) + 3 \times L_2^2(x)$$

$$= \frac{(x-1)(x-2)}{2} - 2x(x-2) + 3\frac{x(x-1)}{2}$$

$$= x + 1$$

Au programme (Chapitre 1)

Objectif :

Étant donné n+1 points de coordonnées $\underline{x}_i \in \mathbb{R}^2, i \in \{0, \cdots, n\}$ déterminer P le polynôme vérifiant

$$P(x_i) = y_i, \ \forall i \in \{0, \cdots, n\} \text{ où } \underline{x}_i = (x_i, y_i)$$

Plan :

I. Interpolation polynomiale et base de Lagrange

II. Évaluation du polynôme d'interpolation

a) Le schéma de Neville-Aitken

b) Les différences divisées et l'algorithme d'Horner

III. Interpolation d'une fonction

IV. Quelques extensions

II. a) Schéma de Neville-Aitken

La base de Lagrange nous permet d'obtenir l'expression du polynôme d'interpolation facilement :

$$P(x) = \sum_{j=0}^{n} y_j L_j^n(x) \quad \text{où} \quad L_i^n(x) := \frac{\prod_{j\neq i}(x - x_j)}{\prod_{j\neq i}(x_i - x_j)}$$

Cependant, cette expression n'est pas pratique pour évaluer P en un point $x \neq x_i$ lorsque n devient grand.

II. a) Schéma de Neville-Aitken

La base de Lagrange nous permet d'obtenir l'expression du polynôme d'interpolation facilement :

$$P(x) = \sum_{j=0}^{n} y_j L_j^n(x) \quad \text{où} \quad L_i^n(x) := \frac{\prod_{j\neq i}(x - x_j)}{\prod_{j\neq i}(x_i - x_j)}$$

Cependant, cette expression n'est pas pratique pour évaluer P en un point $x \neq x_i$ lorsque n devient grand.

Notons par $T_k^i \in \mathbb{P}_k$ le polynôme d'interpolation vérifiant :

$$T_k^i(x_j) = y_j, \quad \forall j \in \{i, \cdots, i+k\}$$

Remarque : On aura en particulier $T_0^i(x) = y_i$

II. a) Schéma de Neville-Aitken

Notons par $T_k^i \in \mathbb{P}_k$ le polynôme d'interpolation vérifiant :

$$T_k^i(x_j) = y_j, \quad \forall j \in \{i, \cdots, i+k\}$$

2.1 Proposition

Les $T_k^i \in \mathbb{P}_k$ vérifient la relation suivante :

$$T_{k+1}^i(x) = \frac{T_k^i(x)(x_{i+k+1} - x) - T_k^{i+1}(x)(x_i - x)}{(x_{i+k+1} - x_i)}$$

Preuve : au (vrai) tableau !

∎

II. a) Schéma de Neville-Aitken

Notons par $T_k^i \in \mathbb{P}_k$ le polynôme d'interpolation vérifiant :

$$T_k^i(x_j) = y_j, \quad \forall j \in \{i, \cdots, i+k\}$$

2.1 Proposition

Les $T_k^i \in \mathbb{P}_k$ vérifient la relation suivante :

$$T_{k+1}^i(x) = \frac{T_k^i(x)(x_{i+k+1} - x) - T_k^{i+1}(x)(x_i - x)}{(x_{i+k+1} - x_i)}$$

Remarques :

La relation ci-dessus nous indique que pour évaluer T_{k+1}^i en x, on a besoin uniquement de $T_k^i(x)$ et $T_k^{i+1}(x)$.

II. a) Schéma de Neville-Aitken

Notons par $T_k^i \in \mathbb{P}_k$ le polynôme d'interpolation vérifiant :

$$T_k^i(x_j) = y_j, \quad \forall j \in \{i, \cdots, i+k\}$$

2.1 Proposition

Les $T_k^i \in \mathbb{P}_k$ vérifient la relation suivante :

$$T_{k+1}^i(x) = \frac{T_k^i(x)(x_{i+k+1} - x) - T_k^{i+1}(x)(x_i - x)}{(x_{i+k+1} - x_i)}$$

Remarques :

La relation ci-dessus nous indique que pour évaluer T_{k+1}^i en x, on a besoin uniquement de $T_k^i(x)$ et $T_k^{i+1}(x)$.

De plus, on notera que $T_n^0(x) = P(x)$.

II. a) Schéma de Neville-Aitken

Algorithme d'évaluation de P en x=a (à la main) :

$$
\begin{array}{c|l}
x_0 & y_0 = T_0^0(a) \\
x_1 & y_1 = T_0^1(a) \\
\vdots & \quad \vdots \\
x_i & y_i = T_0^i(a) \\
\vdots & \quad \vdots \\
x_n & y_n = T_0^n(a)
\end{array}
$$

II. a) Schéma de Neville-Aitken

Algorithme d'évaluation de P en x=a (à la main) :

$$\begin{array}{c|cc}
x_0 & y_0 = T_0^0(a) & \longrightarrow T_1^0(a) \\
x_1 & y_1 = T_0^1(a) & \nearrow \\
\vdots & \vdots & \\
x_i & y_i = T_0^i(a) & \\
\vdots & \vdots & \\
x_n & y_n = T_0^n(a) &
\end{array}$$

II. a) Schéma de Neville-Aitken

Algorithme d'évaluation de P en x=a (à la main) :

$$
\begin{tikzcd}[row sep=small]
x_0 & y_0 = T_0^0(a) \arrow[r] & T_1^0(a) \\
x_1 & y_1 = T_0^1(a) \arrow[ur] \arrow[r] & T_1^1(a) \\
\vdots & \vdots & \vdots \\
x_i & y_i = T_0^i(a) \arrow[ur] \arrow[r] & T_1^i(a) \\
\vdots & \vdots & \vdots \\
 & & T_1^{n-1}(a) \\
x_n & y_n = T_0^n(a) \arrow[ur] &
\end{tikzcd}
$$

II. a) Schéma de Neville-Aitken

Algorithme d'évaluation de P en x=a (à la main) :

$$
\begin{tikzcd}[row sep=small]
x_0 & y_0 = T_0^0(a) \arrow[r] & T_1^0(a) & \cdots & T_{n-1}^0(a) \arrow[r] & \boxed{T_n^0(a)} \\
x_1 & y_1 = T_0^1(a) \arrow[ur] \arrow[r] & T_1^1(a) & \cdots & T_{n-1}^1(a) \arrow[ur] & \\
\vdots & \vdots & \vdots & \cdots & \cdots & \\
x_i & y_i = T_0^i(a) \arrow[ur] \arrow[r] & T_1^i(a) & & & \\
\vdots & \vdots & \vdots & \cdots & & \\
 & & T_1^{n-1}(a) & & & \\
x_n & y_n = T_0^n(a) \arrow[ur] & & & &
\end{tikzcd}
$$

II. a) Schéma de Neville-Aitken

Algorithme d'évaluation de P en x=a (à la main) :

$$
\begin{tikzcd}
x_0 & y_0 = T_0^0(a) \arrow[r] & T_1^0(a) & \cdots & T_{n-1}^0(a) \arrow[r] & \boxed{T_n^0(a)} \\
x_1 & y_1 = T_0^1(a) \arrow[r] \arrow[ur] & T_1^1(a) & \cdots & T_{n-1}^1(a) \arrow[ur] & \\
\vdots & \vdots & \vdots & \cdots & \cdots & \\
x_i & y_i = T_0^i(a) \arrow[r] \arrow[ur] & T_1^i(a) & & & \\
\vdots & \vdots & \vdots & \cdots & & \\
 & & T_1^{n-1}(a) & & & \\
x_n & y_n = T_0^n(a) \arrow[ur] & & & &
\end{tikzcd}
$$

Remarque :

L'ordre des points x_i ne compte pas.

II. a) Schéma de Neville-Aitken

Algorithme d'évaluation de P en x=a (à la main) :

$$\begin{tikzcd}[column sep=small]
x_0 & y_0 = T_0^0(a) \arrow[r] & T_1^0(a) & \cdots & T_{n-1}^0(a) \arrow[r] & \boxed{T_n^0(a)} \\
x_1 & y_1 = T_0^1(a) \arrow[ur] \arrow[r] & T_1^1(a) & \cdots & T_{n-1}^1(a) \arrow[ur] & \\
\vdots & \vdots & \vdots & \cdots & \ddots & \\
x_i & y_i = T_0^i(a) \arrow[u] \arrow[r] & T_1^i(a) & & & \\
 & & \vdots & \ddots & & \\
\vdots & \vdots & T_1^{n-1}(a) & & & \\
x_n & y_n = T_0^n(a) \arrow[u] & & & &
\end{tikzcd}$$

Remarque :

L'ordre des points x_i ne compte pas. En revanche, changer le point d'évaluation nécessite de tout recalculer !

II. a) Schéma de Neville-Aitken

Algorithme d'évaluation de P en x=a :

Pour k = 0 à n-1

  Pour i = 0 à n-k-1

    $T_{k+1}[i] \leftarrow \dfrac{T_k[i](x[i+k+1]-a) - T_k[i+1](x[i]-a)}{(x[i+k+1] - x[i])}$

  $T_k \leftarrow T_{k+1}$

Complexité : $O(7n^2/2)$

À chaque itération i, on effectue 4 différences, 2 produits et une division, soit au total 7 opérations. Au total, on a donc

$$\sum_{k=0}^{n-1}\sum_{i=0}^{n-k} 7 = 7\sum_{k=0}^{n-1}(n-k) = 7\frac{n(n+1)}{2} \text{ opérations}$$

II. a) Schéma de Neville-Aitken

Algorithme d'évaluation de P en x=a :

```
Pour k = 0 à n-1
  Pour i = 0 à n-k-1
```

$$T_{k+1}[i] \leftarrow \frac{T_k[i](x[i+k+1]-a) - T_k[i+1](x[i]-a)}{(x[i+k+1] - x[i])}$$

$$T_k \leftarrow T_{k+1}$$

Complexité : ~~O(7n²/2)~~ O(5n²/2)

À chaque itération i, on effectue 4 différences, 2 produits et une division, soit au total 7 opérations. Au total, on a donc

$$\sum_{k=0}^{n-1}\sum_{i=0}^{n-k} 7 = 7\sum_{k=0}^{n-1}(n-k) = 7\frac{n(n+1)}{2} \text{ opérations}$$

II. b) Différences divisées et Horner

Revenons un peu en arrière, et cherchons une base t.q. :

- la décomposition dans cette base est facile à obtenir,
- l'évaluation du polynôme avec cette base soit facile.

Base de Newton :

Idée : Choisir les $(\varphi_j)_{j=0,\cdots,n}$ t.q. la matrice de Gram

$$G = \begin{bmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \vdots & \vdots & & \vdots \\ \varphi_0(x_n) & \varphi_1(x_n) & \cdots & \varphi_n(x_n) \end{bmatrix}$$

triangulaire (inférieur), i.e. $\varphi_j(x_i) = 0 \quad \forall j > i$

II. b) Différences divisées et Horner

Base de Newton :

2.2 Proposition

La famille de fonction $(N_i)_{i=\{0,\cdots,n\}}$ définie par

$$N_0(x) = 1 \text{ et } N_i(x) = \prod_{j=0}^{i-1}(x - x_j)$$

forme une base de $\mathbb{P}_n$ et vérifie $N_j(x_i) = 0 \ \forall j > i$.

Preuve : au (vrai) tableau ! ∎

II. b) Différences divisées et Horner

Base de Newton : $\mathbb{P}_n = \text{vect}\{N_0, N_1, \cdots, N_n\}$

La décomposition du polynôme dans cette base pourrait être obtenu par résolution du sytème linéaire ($O(8n^2)$).

$$\begin{bmatrix} N_0(x_0) & 0 & \cdots & 0 \\ N_0(x_1) & N_1(x_1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ N_0(x_n) & N_1(x_n) & \cdots & N_n(x_n) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

II. b) Différences divisées et Horner

Base de Newton : $\mathbb{P}_n = \text{vect}\{N_0, N_1, \cdots, N_n\}$

La décomposition du polynôme dans cette base pourrait être obtenu par résolution du sytème linéaire ($O(8n^2)$).

Mais on peut faire mieux ! Posons :

$$f[x_i, \cdots, x_{i+j+1}] = \frac{f[x_{i+1}, \cdots, x_{i+j+1}] - f[x_i, \cdots, x_{i+j}]}{x_{i+j+1} - x_i}, \; f[x_i] = f(x_i)$$

2.2 Proposition

Le polynôme d'interpolation est donné par

$$P(x) = f[x_0]N_0(x) + f[x_1, x_1]N_1(x) + \cdots + f[x_0, x_1, \cdots, x_n]N_n(x)$$

Preuve : au (vrai) tableau ! $\blacksquare$

II. b) Différences divisées et Horner

Algorithme des différences divisées :

$$
\begin{array}{c|l}
x_0 & y_0 = f[x_0] \\
x_1 & y_1 = f[x_1] \\
\vdots & \quad\vdots \\
x_i & y_i = f[x_i] \\
\vdots & \quad\vdots \\
x_n & y_n = f[x_n]
\end{array}
$$

II. b) Différences divisées et Horner

Algorithme des différences divisées :

$$\begin{array}{c|ccc}
x_0 & y_0 = f[x_0] & \longrightarrow & f[x_0, x_1] \\
x_1 & y_1 = f[x_1] & \nearrow & \\
\vdots & \vdots & & \\
x_i & y_i = f[x_i] & & \\
\vdots & \vdots & & \\
x_n & y_n = f[x_n] & &
\end{array}$$

II. b) Différences divisées et Horner

Algorithme des différences divisées :

$$
\begin{array}{c|lcl}
x_0 & y_0 = f[x_0] & \longrightarrow & f[x_0, x_1] \\
x_1 & y_1 = f[x_1] & \nearrow \longrightarrow & f[x_1, x_2] \\
\vdots & \quad \vdots & & \quad \vdots \\
x_i & y_i = f[x_i] & \nearrow \longrightarrow & f[x_i, x_{i+1}] \\
\vdots & \quad \vdots & & \quad \vdots \\
 & & & f[x_{n-1}, x_n] \\
x_n & y_n = f[x_n] & \nearrow &
\end{array}
$$

II. b) Différences divisées et Horner

Algorithme des différences divisées :

$$
\begin{array}{c|llllll}
x_0 & y_0 = \boxed{f[x_0] & \longrightarrow & f[x_0, x_1] & \cdots & f[x_0, \cdots, x_{n-1}] \longrightarrow & *} \\
x_1 & y_1 = f[x_1] & \nearrow \longrightarrow & f[x_1, x_2] & \cdots & f[x_1, \cdots, x_n] & \nearrow \\
\vdots & \quad \vdots & & \vdots & \cdots & \ddots & \uparrow \\
x_i & y_i = f[x_i] & \nearrow \longrightarrow & f[x_i, x_{i+1}] & & & \\
 & & & \vdots & \ddots & & f[x_0, \cdots, x_n] \\
\vdots & \quad \vdots & & & & & \\
 & & \nearrow & f[x_{n-1}, x_n] & & & \\
x_n & y_n = f[x_n] & & & & &
\end{array}
$$

II. b) Différences divisées et Horner

Algorithme des différences divisées :

$$
\begin{tikzcd}
x_0 & y_0 = f[x_0] \arrow[r] & f[x_0, x_1] & \cdots & f[x_0, \cdots, x_{n-1}] \arrow[r] & * \\
x_1 & y_1 = f[x_1] \arrow[r] \arrow[ur] & f[x_1, x_2] & \cdots & f[x_1, \cdots, x_n] \arrow[ur] & \\
\vdots & \vdots & \vdots & \cdots & \cdots & \\
x_i & y_i = f[x_i] \arrow[r] \arrow[ur] & f[x_i, x_{i+1}] & & & \\
\vdots & \vdots & \vdots & \cdots & & f[x_0, \cdots, x_n] \arrow[uuuu] \\
 & & f[x_{n-1}, x_n] & & & \\
x_n & y_n = f[x_n] \arrow[ur] & & & &
\end{tikzcd}
$$

Remarque :

L'ordre des points x_i ne compte pas.

II. b) Différences divisées et Horner

Algorithme des différences divisées :

```
Pour k = 0 à n-1
    Pour i = 0 à n-k-1
        F_{k+1}[i] ← (F_k[i+1] − F_k[i]) / (x[i+k+1] - x[i])
    a_{k+1} ← F_{k+1}[0]
    F_k ← F_{k+1}
```

Complexité : $O(3n^2/2)$

À chaque itération i, on effectue 2 différences et une division, soit au total 3 opérations. Au total, on a donc

$$\sum_{k=0}^{n-1}\sum_{i=0}^{n-k} 3 = 3\sum_{k=0}^{n-1}(n-k) = 3\frac{n(n+1)}{2} \text{ opérations}$$

II. b) Différences divisées et Horner

Voyons maintenant comment évaluer notre polynôme

$$P(x) = f[x_0]N_0(x) + f[x_0, x_1]N_1(x) + \cdots + f[x_0, x_1, \cdots, x_n]N_n(x)$$

exprimé dans la base de Newton.

2.3 Proposition (Algorithme de Horner)

En posant $q_{i-1} = f[x_0, \cdots, x_i] + (a - x_i)q_i \ \forall i \in \{0, n\}$ et $q_n = 0$,

on a $P(a) = q_{-1}$.

<u>Preuve</u> : au (vrai) tableau ! ■

II. b) Différences divisées et Horner

Algorithme de Horner :

```
Pour i = 0 à n
    q_{i-1} ← F[i] + (a-x_i) q_i
```

Complexité : O(3n)

II. b) Différences divisées et Horner

Algorithme de Horner :

```
Pour i = 0 à n
| q_{i-1} ← F[i] + (a-x_i)q_i
```

Complexité : $O(3n)$

Avec cette approche, pour évaluer P en x=a, il faut donc :

- calculer la décomposition dans la base de Newton (Différences divisées) $O(3n^2/2)$
- appliquer l'algorithme de Horner $O(3n)$

$\Big\}\ O(3n^2/2)$

II. b) Différences divisées et Horner

Algorithme de Horner :

```
Pour i = 0 à n
    q_{i-1} ← F[i] + (a-x_i) q_i
```

$$\text{Pour } i = 0 \text{ à } n: \quad q_{i-1} \leftarrow F[i] + (a - x_i)\,q_i$$

Complexité : O(3n)

Avec cette approche, pour évaluer P en x=a, il faut donc :

$$\left.\begin{array}{l}\text{calculer la décomposition dans la base de Newton (Différences divisées) } O(3n^2/2) \\ \text{appliquer l'algorithme de Horner } O(3n)\end{array}\right\} O(3n^2/2)$$

Remarque :

Changer le point d'évaluation nécessite uniquement d'appliquer de nouveau l'algorithme d'Horner !

II. b) Différences divisées et Horner

Au final :

Neville Aitken	DD + Horner

II. b) Différences divisées et Horner

Au final :

Neville Aitken	DD + Horner
- Complexité : $O(5n^2/2)$	+ Complexité : $O(3n^2/2)$

II. b) Différences divisées et Horner

Au final :

Neville Aitken	DD + Horner
- Complexité : $O(5n^2/2)$	+ Complexité : $O(3n^2/2)$
- Evaluation en un nouveau point : $O(5n^2/2)$	+ Evaluation en un nouveau point : $O(3n)$

II. b) Différences divisées et Horner

Au final :

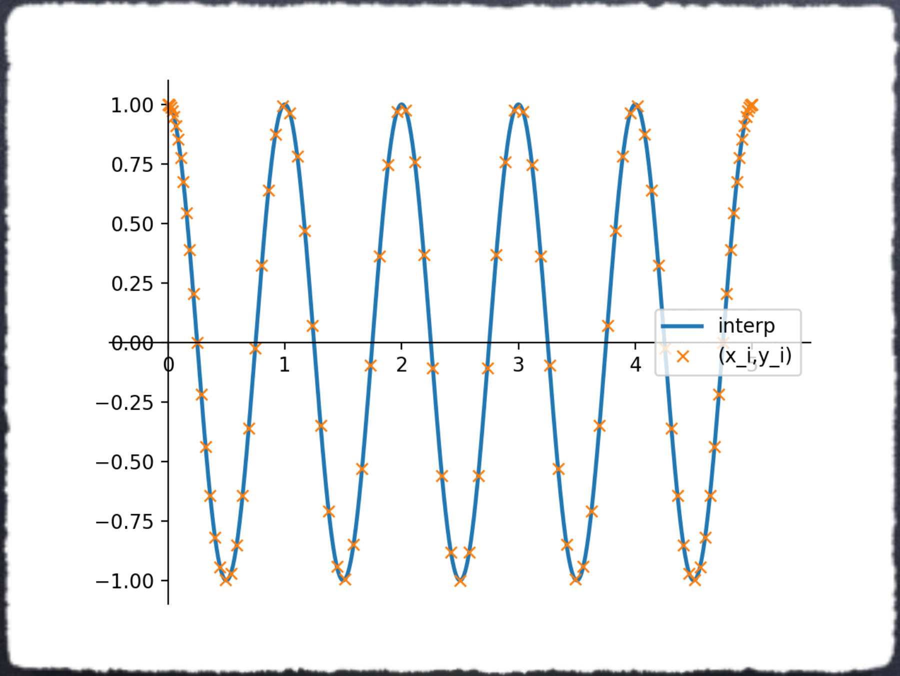
Neville Aitken

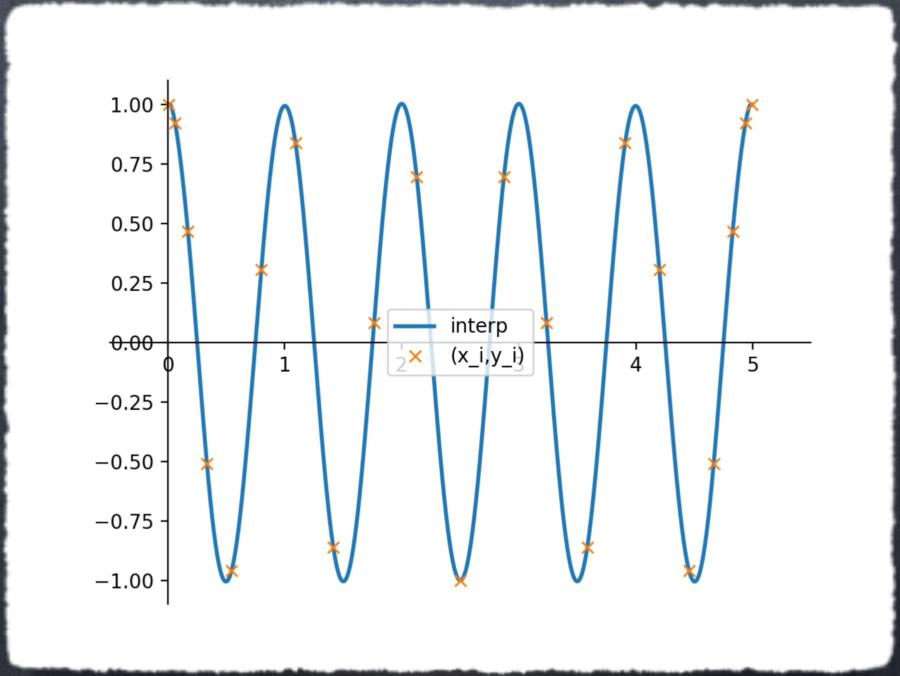
- Complexité : $O(5n^2/2)$
- Evaluation en un nouveau point : $O(5n^2/2)$

DD + Horner

+ Complexité : $O(3n^2/2)$
+ Evaluation en un nouveau point : $O(3n)$

MAIS...





Au programme (Chapitre 1)

Objectif :

Étant donné n+1 points de coordonnées $\underline{x}_i \in \mathbb{R}^2, i \in \{0, \cdots, n\}$ déterminer P le polynôme vérifiant

$$P(x_i) = y_i, \ \forall i \in \{0, \cdots, n\} \text{ où } \underline{x}_i = (x_i, y_i)$$

Plan :

I. Interpolation polynomiale et base de Lagrange

II. Évaluation du polynôme d'interpolation

III. Interpolation d'une fonction

   a) Erreur d'interpolation

   b) Meilleurs points d'interpolation

IV. Quelques extensions

III. a) Erreur d'interpolation

Considérons une fonction f: $\mathbb{R} \to \mathbb{R}$ continue.

3.1 Théorème de Weierstrass (admis)

Pour toute fonction f continue, il existe une suite de polynôme $P_n \in \mathbb{P}_n$ t.q.

$$\lim_{n \to +\infty} \|f - P_n\|_\infty = 0$$

III. a) Erreur d'interpolation

Considérons une fonction $f: \mathbb{R} \to \mathbb{R}$ continue.

3.1 Théorème de Weierstrass (admis)

Pour toute fonction f continue, il existe une suite de polynôme $P_n \in \mathbb{P}_n$ t.q.

$$\lim_{n \to +\infty} \|f - P_n\|_\infty = 0$$

Ce résultat nous indique que toute fonction continue peut être approchée par un polynôme, donc…

Idée : Approcher f par le polynôme d'interpolation P_n vérifiant : $P_n(x_i) = y_i$, où $y_i = f(x_i) \quad \forall i \in \{0, \cdots, n\}$

III. a) Erreur d'interpolation

3.2 Définition

On appelle polynôme d'interpolation de Lagrange associé à f le polynôme $P_n f \in \mathbb{P}_n$ vérifiant

$$P_n f(x_i) = f(x_i), \quad \forall i \in \{0, \cdots, n\}$$

III. a) Erreur d'interpolation

3.2 Définition

On appelle polynôme d'interpolation de Lagrange associé à f le polynôme $P_n f \in \mathbb{P}_n$ vérifiant

$$P_n f(x_i) = f(x_i), \ \forall i \in \{0, \cdots, n\}$$

3.3 Théorème

Si f est de classe C^{n+1} sur $[a,b]$, avec $a \leq x_0 < \cdots < x_n \leq b$ alors $\forall x \in [a,b]$, il existe $\xi \in [\min(x_0, x), \max(x_n, x)]$ t.q.

$$f(x) - P_n f(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} v_n(x) \text{ où } v_n(x) = \prod_{i=0}^{n} (x - x_i)$$

<u>Preuve :</u> au (vrai) tableau ! ∎

III. a) Erreur d'interpolation

3.3 Théorème

Si f est de classe C^{n+1} sur $[a,b]$, avec $a \leq x_0 < \cdots < x_n \leq b$

alors $\forall x \in [a,b]$, il existe $\xi \in [\min(x_0,x), \max(x_n,x)]$ t.q.

$$f(x) - P_n f(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} v_n(x) \text{ où } v_n(x) = \prod_{i=0}^{n} (x - x_i)$$

3.4 Corollaire

On en déduit : $\|f - P_n f\|_\infty \leq \dfrac{\|f^{(n+1)}\|_\infty}{(n+1)!} \|v_n\|_\infty$

Preuve : au (vrai) tableau ! ∎

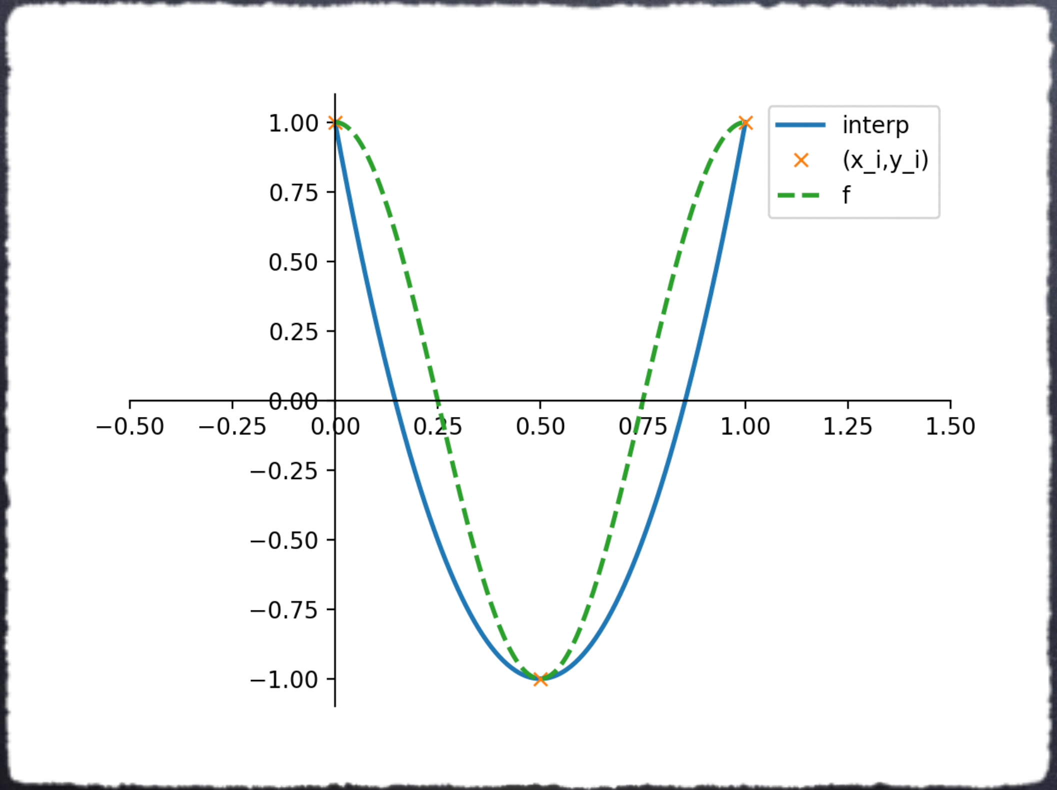
III. a) Erreur d'interpolation

Peut-on déduire la convergence ?

$$\|f - P_n f\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \|v_n\|_\infty \xrightarrow[n \to +\infty]{} 0 \ ?$$

Exemple :

Prenons par exemple les points équirépartis $x_i = i\dfrac{b-a}{n} + a$ et la fonction $f(x) = \cos(2\pi x)$ sur $[0,1]$:



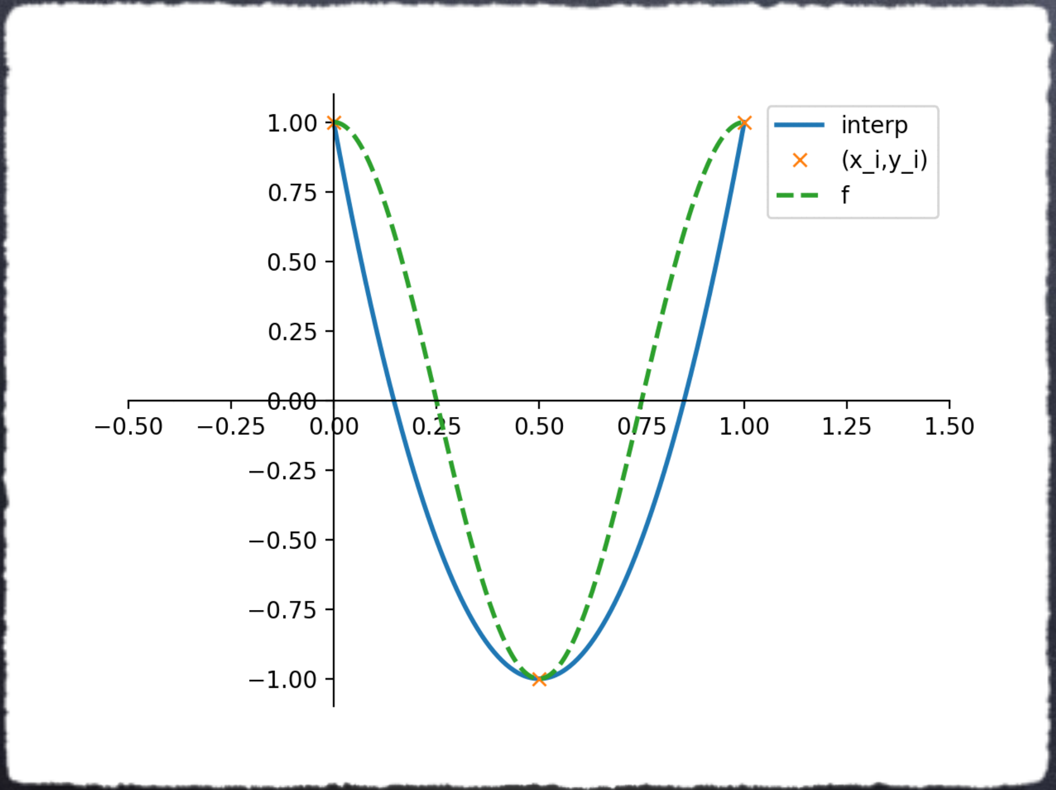
III. a) Erreur d'interpolation

Peut-on déduire la convergence ?

$$\|f - P_n f\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}\|v_n\|_\infty \xrightarrow[n \to +\infty]{} 0 \ ?$$

Exemple :

Prenons par exemple les points équirépartis $x_i = i\dfrac{b-a}{n} + a$ et la fonction $f(x) = \cos(2\pi x)$ sur $[0,1]$:



Sur cet exemple, on observe bien la convergence, ce qui est logique car :

$$\frac{(2\pi)^{n+1}}{(n+1)!} \xrightarrow[n\to+\infty]{} 0$$

(détails au (vrai) tableau)

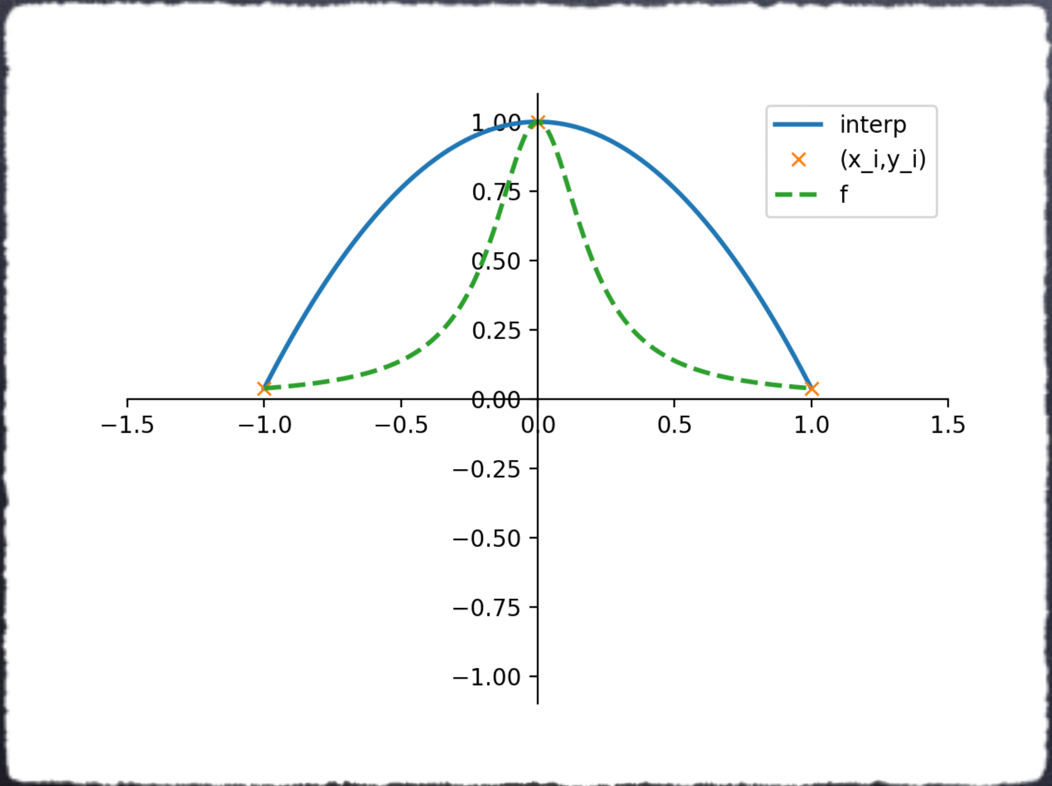
III. a) Erreur d'interpolation

Peut-on déduire la convergence ?

$$\|f - P_n f\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}\|v_n\|_\infty \xrightarrow[n\to+\infty]{} 0 \ ?$$

Exemple 2 (Runge) :

Prenons par exemple les points équirépartis $x_i = i\dfrac{b-a}{n} + a$ et la fonction $f(x) = \dfrac{1}{1+25x^2}$ sur $[-1,1]$:



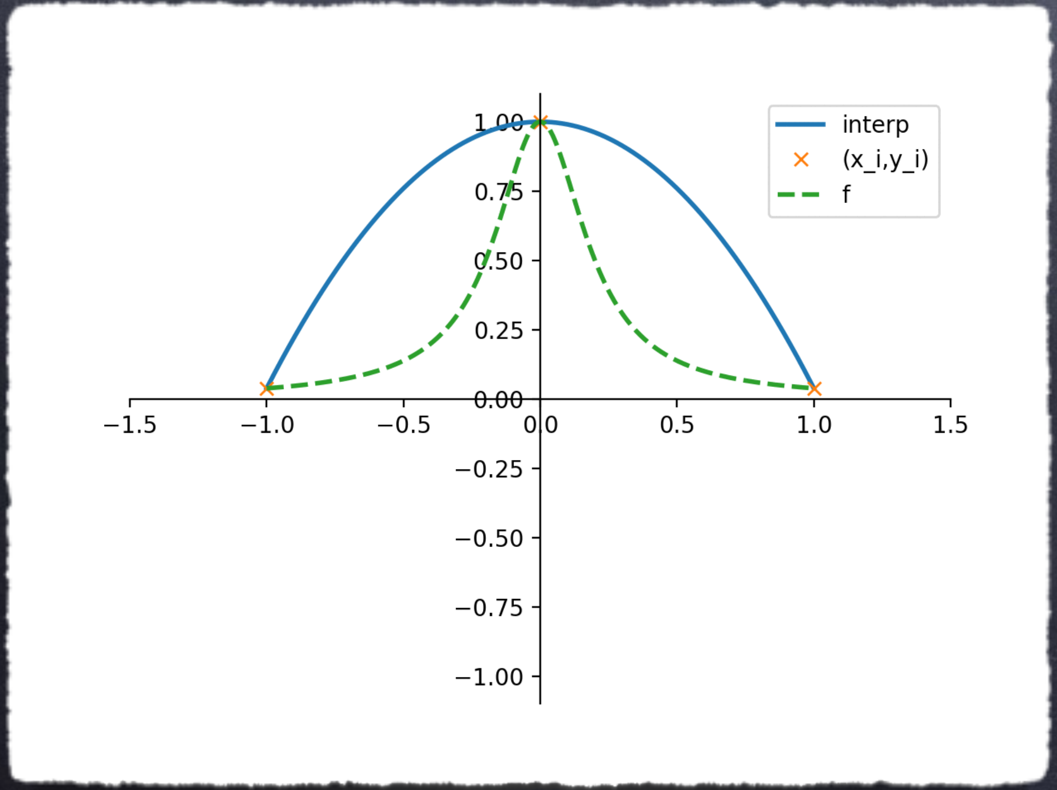
III. a) Erreur d'interpolation

Peut-on déduire la convergence ? NON !

$$\|f - P_n f\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \|v_n\|_\infty \xrightarrow[n \to +\infty]{} 0 \ ?$$

Exemple 2 (Runge) :

Prenons par exemple les points équirépartis $x_i = i\dfrac{b-a}{n} + a$ et la fonction $f(x) = \dfrac{1}{1+25x^2}$ sur $[-1,1]$:



On observe ici le phénomène de Runge !

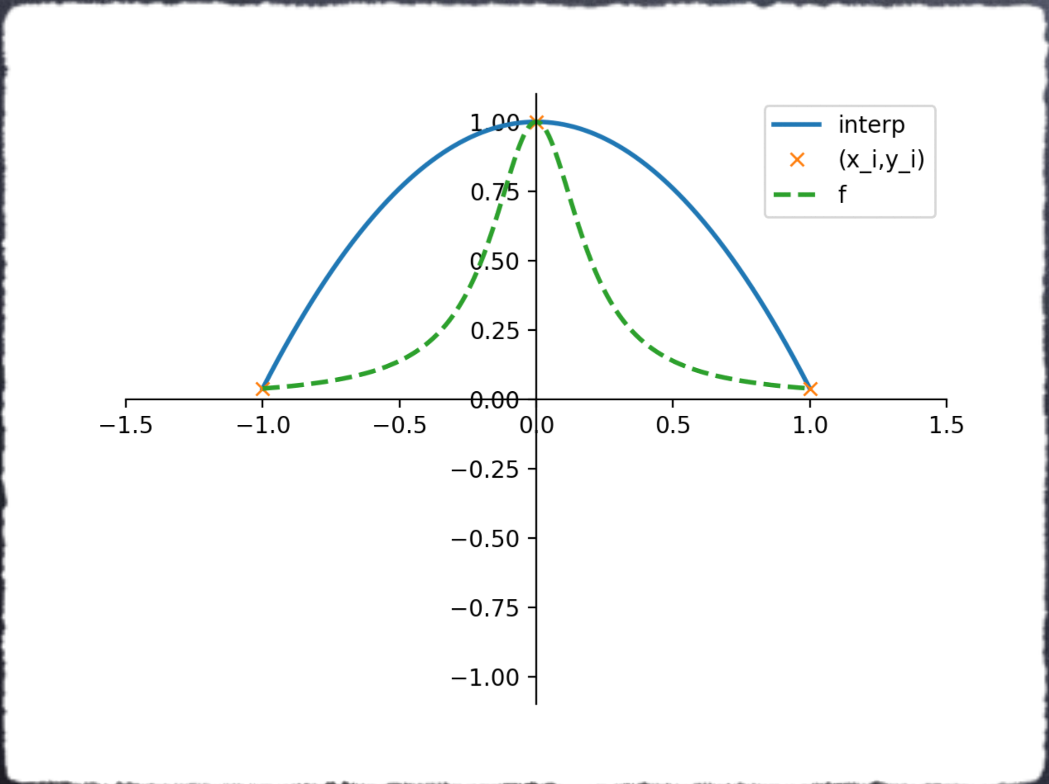
III. a) Erreur d'interpolation

Peut-on déduire la convergence ? NON !

$$\|f - P_n f\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}\|v_n\|_\infty \xrightarrow[n \to +\infty]{} 0 \ ?$$

Exemple 2 (Runge) :

Prenons par exemple les points équirépartis $x_i = i\frac{b-a}{n} + a$ et la fonction $f(x) = \frac{1}{1+25x^2}$ sur $[-1,1]$:



On observe ici le phénomène de Runge !

Peut-on mieux choisir les points d'interpolation ?

III. b) Meilleurs points d'interpolation

Revenons à notre majoration :

$$\|f - P_n f\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \|v_n\|_\infty$$

où on rappelle que $v_n(x) = \prod_{i=0}^{n} (x - x_i)$

III. b) Meilleurs points d'interpolation

Revenons à notre majoration :

$$\|f - P_n f\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \boxed{\|v_n\|_\infty}$$

où on rappelle que $v_n(x) = \prod_{i=0}^{n} (x - x_i)$

Problème :

Déterminer les coordonnées x_i minimisant $\|v_n\|_\infty$.

III. b) Meilleurs points d'interpolation

Revenons à notre majoration :

$$\|f - P_n f\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \|v_n\|_\infty$$

où on rappelle que $v_n(x) = \prod_{i=0}^{n} (x - x_i)$

Problème :

Déterminer les coordonnées x_i minimisant $\|v_n\|_\infty$.

Pour simplifier l'étude, on se ramènera au cas où

$$a = -1 \leq x_0 < \cdots < x_n \leq 1 = b$$

ce qui est toujours possible en utilisant le changement de variable:

$$x = (\hat{x}\frac{b-a}{2}) + \frac{b+a}{2}, \text{ où } \hat{x} \in [-1, 1]$$

3.5 Définition

On appelle polynôme de Chebychev de première espèce et de degré k le polynôme définie par

$$T_k(x) = \cos(k \arccos(x)) \ \forall x \in [-1, 1]$$

3.6 Théorème

Les polynômes de Chebychev vérifient la relation de de récurrence suivante :

$$T_{k+1}(x) = 2x T_k(x) - T_{k-1}(x), \quad \forall k \geq 1$$

Preuve : au (vrai) tableau ! ■

III. b) Meilleurs points d'interpolation

3.5 Définition

On appelle polynôme de Chebychev de première espèce et de degré k le polynôme définie par

$$T_k(x) = \cos(k \arccos(x)) \ \forall x \in [-1, 1]$$

3.6 Théorème

Les polynômes de Chebychev vérifient la relation de de récurrence suivante :

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x), \quad \forall k \geq 1$$

Remarque :

Ce théorème nous « rassure » sur le fait que le polynôme $T_k \in \mathbb{P}_k$.

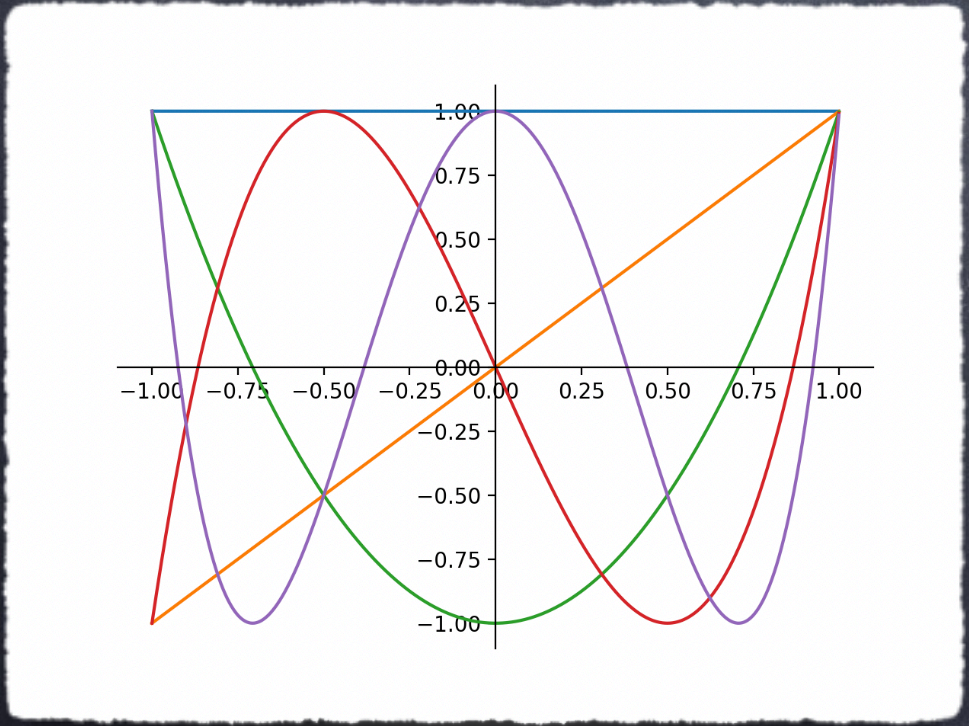
III. b) Meilleurs points d'interpolation

3.5 Définition

On appelle polynôme de Chebychev de première espèce et de degré k le polynôme définie par

$$T_k(x) = \cos(k \arccos(x)) \;\forall x \in [-1, 1]$$

Illustration :



T_0, T_1, T_2, T_3 et T_4

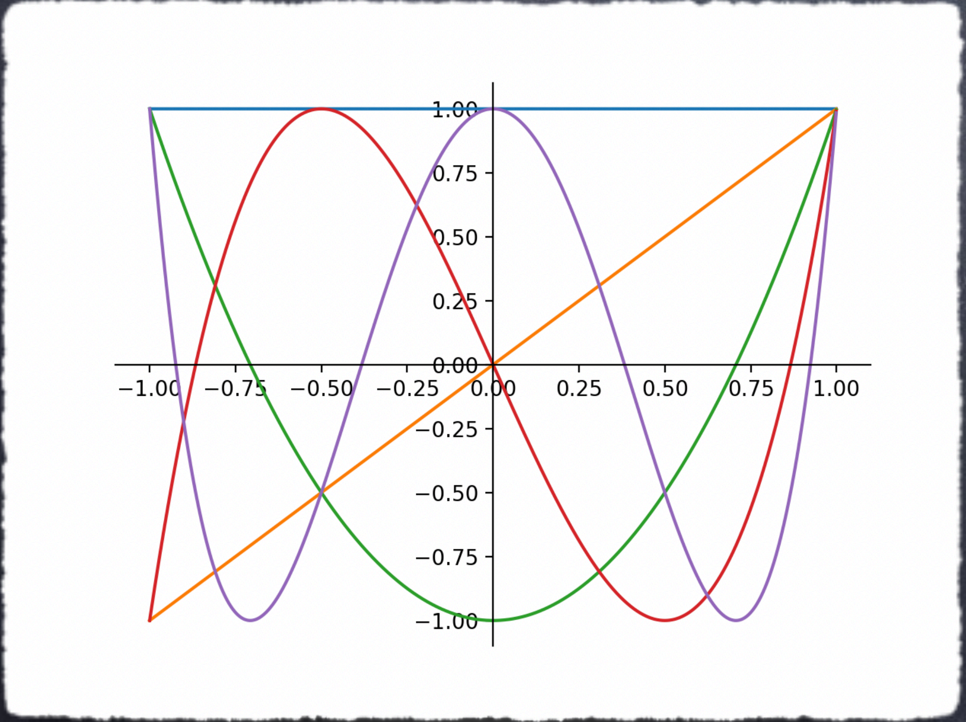
III. b) Meilleurs points d'interpolation

3.5 Définition

On appelle polynôme de Chebychev de première espèce et de degré k le polynôme définie par

$$T_k(x) = \cos(k \arccos(x)) \ \forall x \in [-1,1]$$

Illustration :



T_0, T_1, T_2, T_3 et T_4

Remarque :
À l'aide de la définition ci-dessus, on remarque facilement que $T_k(x) \in [-1,1], \quad \forall x \in [-1,1]$

III. b) Meilleurs points d'interpolation

Problème :

Déterminer les x_i minimisant $\|v_n\|_\infty$ où $v_n(x) = \prod_{i=0}^{n}(x - x_i)$

Posons Γ_n l'ensemble des polynômes de degré n :

- unitaire, c'est à dire t.q. $P(x) = x^n + \cdots$
- et possédant n racines réelles distinctes.

Notre problème peut alors se reformuler ainsi :

Trouver $P \in \Gamma_{n+1}$ minimisant $\|P\|_\infty$

III. b) Meilleurs points d'interpolation

Problème :

Déterminer les x_i minimisant $\|v_n\|_\infty$ où $v_n(x) = \prod_{i=0}^{n}(x - x_i)$

Posons Γ_n l'ensemble des polynômes de degré n :

- unitaire, c'est à dire t.q. $P(x) = x^n + \cdots$
- et possédant n racines réelles distinctes.

Notre problème peut alors se reformuler ainsi :

Trouver $P \in \Gamma_{n+1}$ minimisant $\|P\|_\infty$

Remarque :

Le polynôme $\dfrac{T_{k+1}}{2^k} \in \Gamma_{k+1}$ (détails au (vrai) tableau)

III. b) Meilleurs points d'interpolation

3.7 Théorème

Une solution du problème

$$\text{Trouver } P \in \Gamma_{n+1} \text{ minimisant } \|P\|_\infty$$

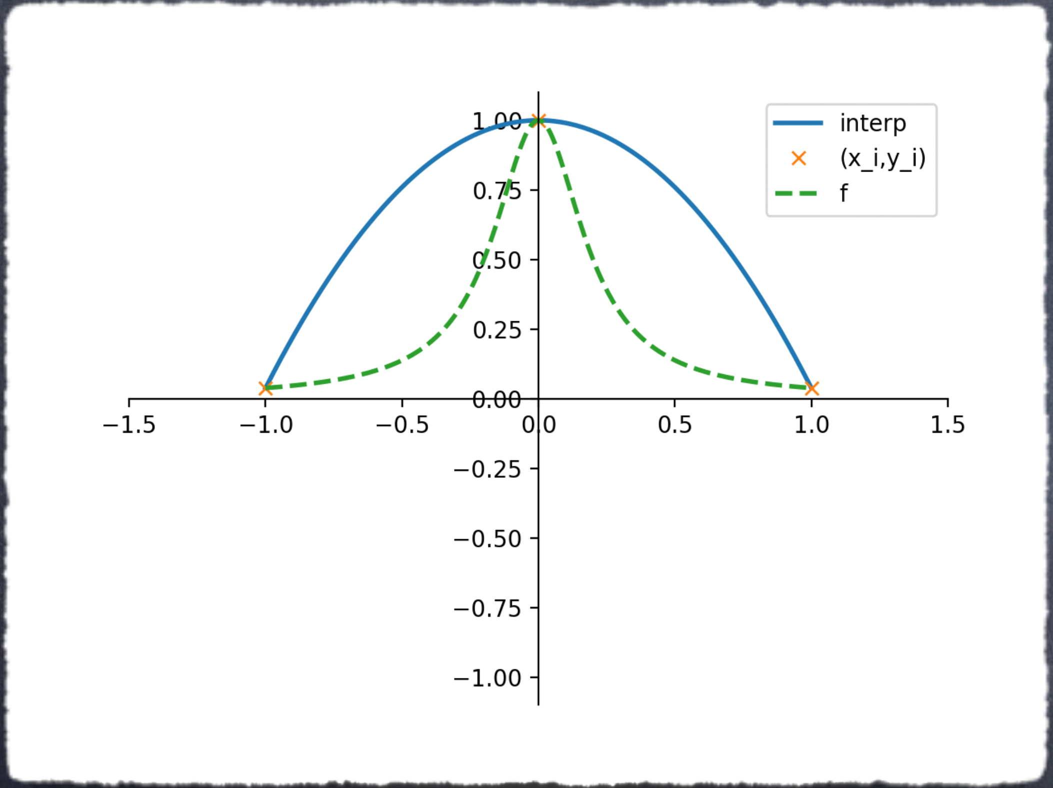
est donnée par $\dfrac{T_{n+1}}{2^n}$.

Preuve : au (vrai) tableau ! ■

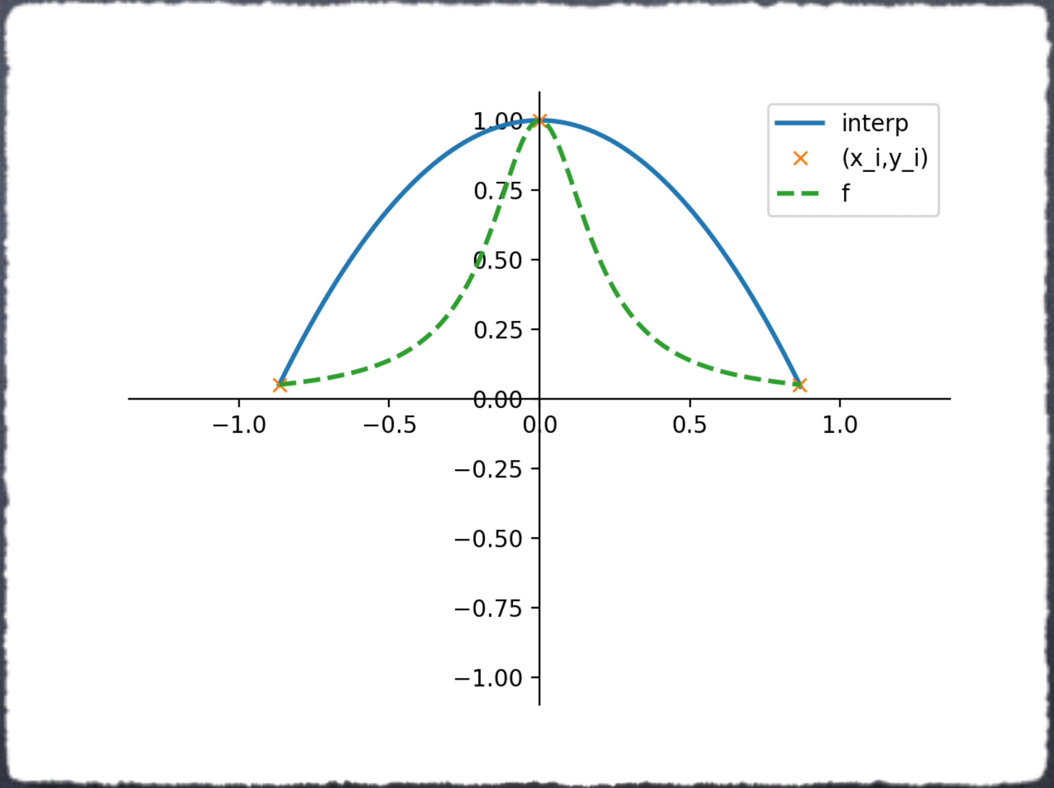
III. b) Meilleurs points d'interpolation

Exemple (Runge) :

Reprenons la fonction $f(x) = \dfrac{1}{1+25x^2}$ sur $[-1,1]$



Points équirépartis

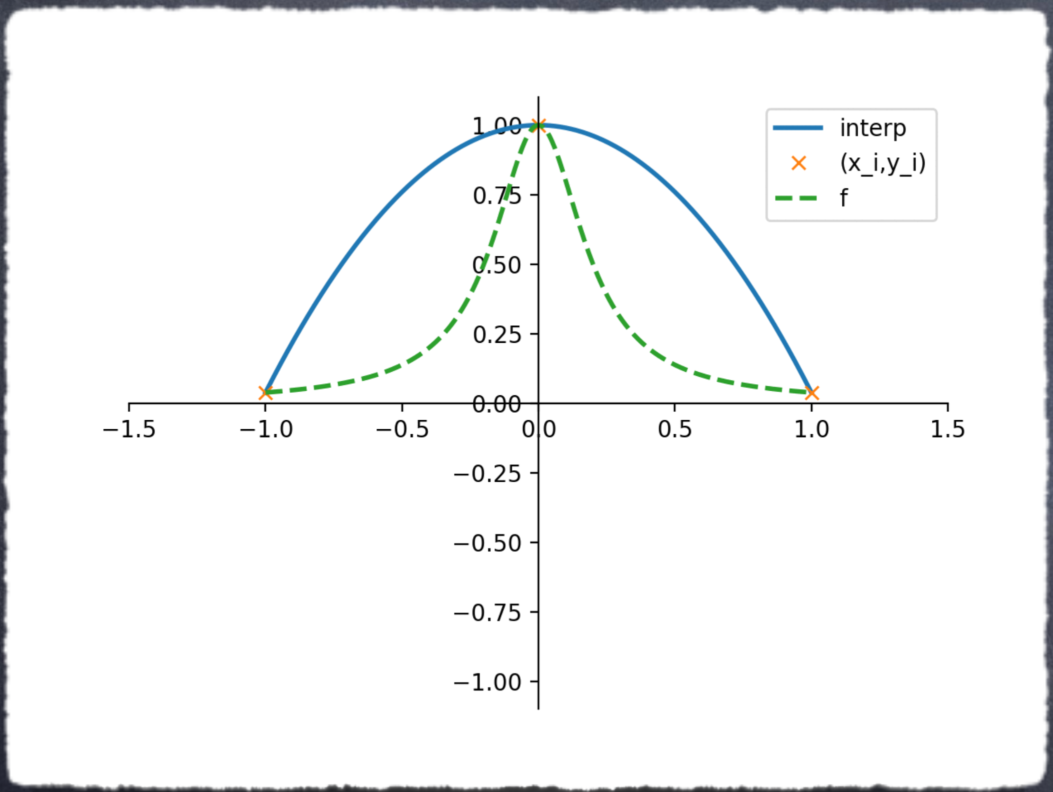


Points Chebychev

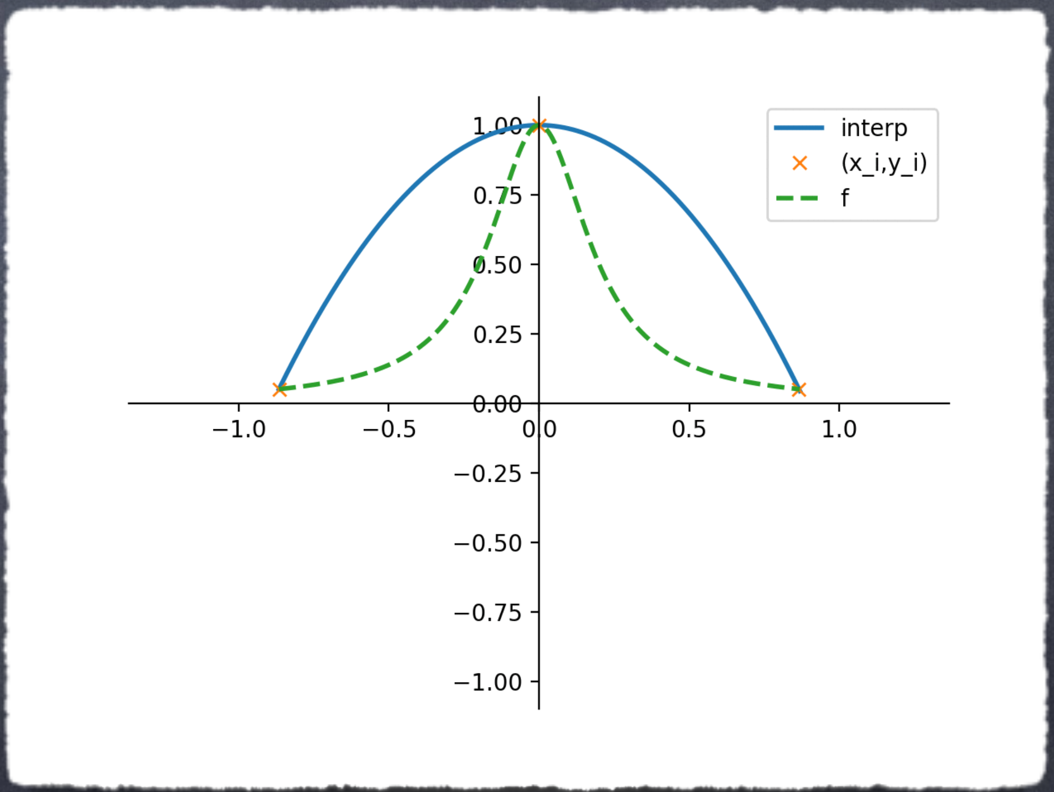
III. b) Meilleurs points d'interpolation

Exemple (Runge) :

Reprenons la fonction $f(x) = \dfrac{1}{1+25x^2}$ sur $[-1,1]$



Points équirépartis

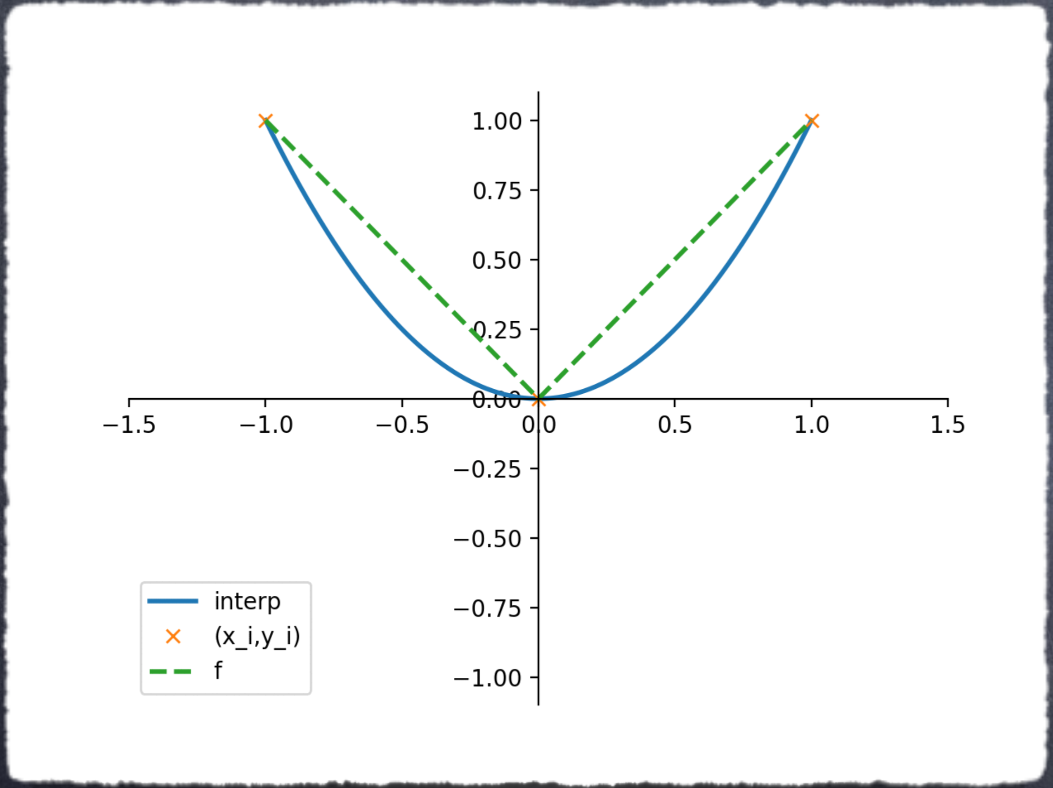


Points Chebychev

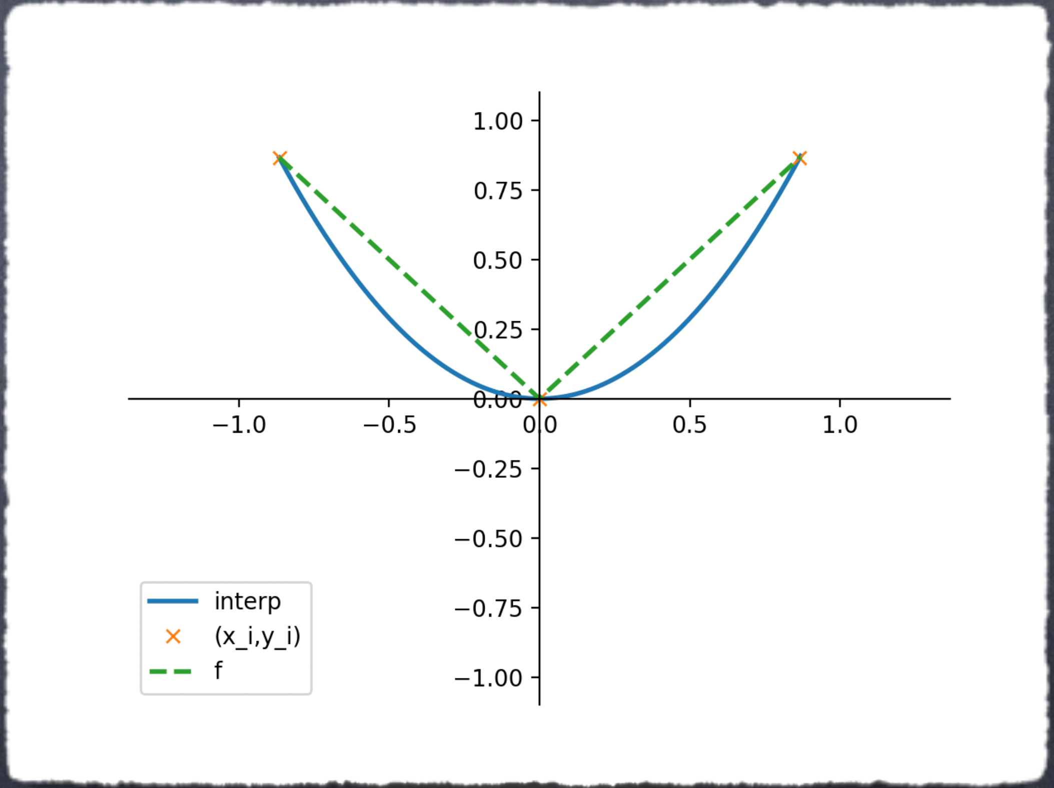
III. b) Meilleurs points d'interpolation

Exemple (non régulier) :

Considérons la fonction $f(x) = |x|$ sur $[-1,1]$



Points équirépartis

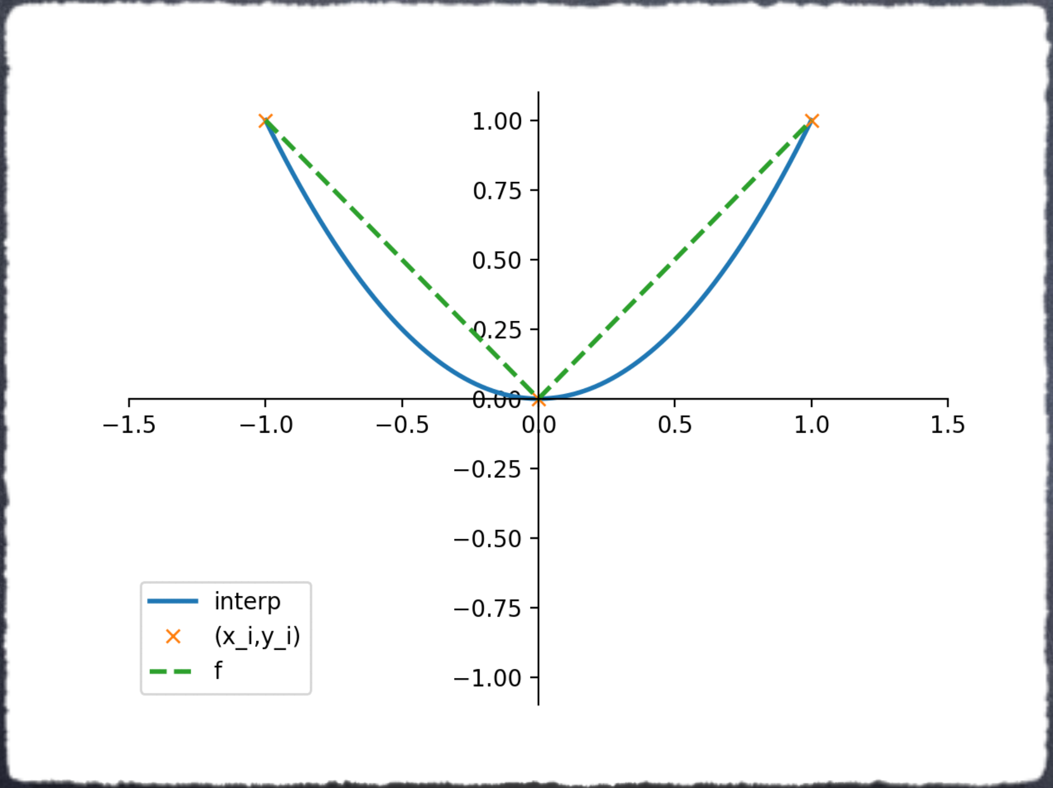


Points Chebychev

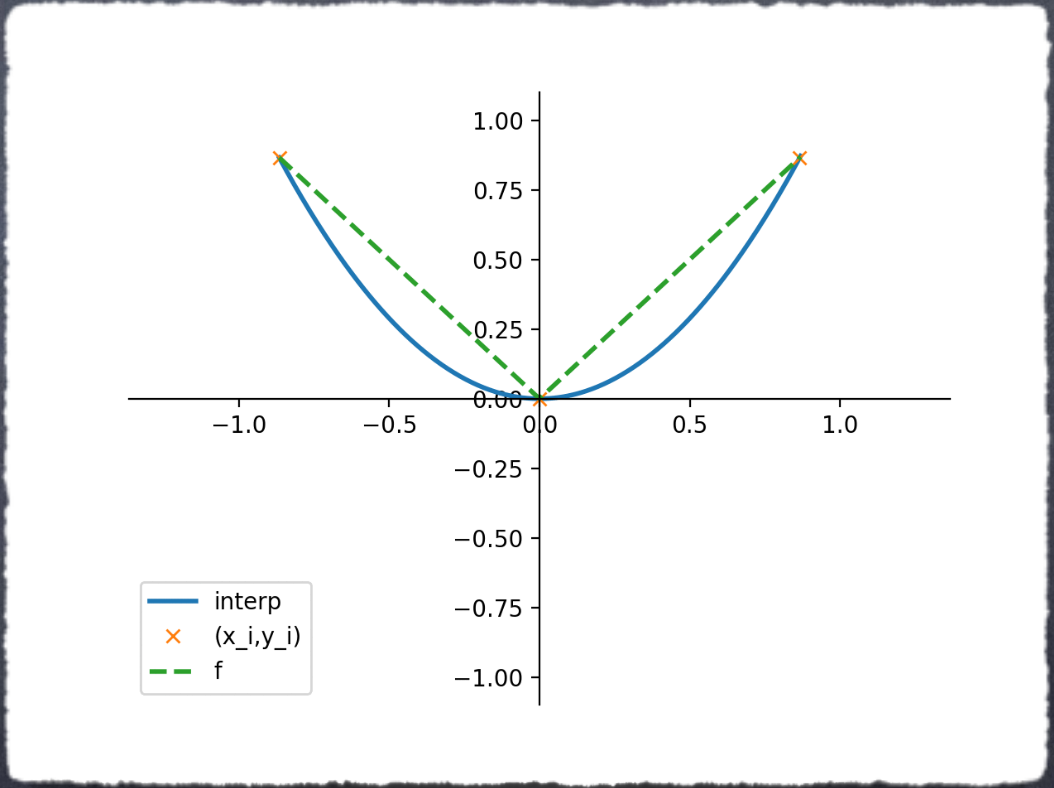
III. b) Meilleurs points d'interpolation

Exemple (non régulier) :

Considérons la fonction $f(x) = |x|$ sur $[-1,1]$



Points équirépartis



Points Chebychev

III. b) Meilleurs points d'interpolation

Les points de Chebychev résolvent-ils tous nos soucis ?

$$\|f - P_n f\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \boxed{\|v_n\|_\infty}$$

En choisissant $v_n = \frac{T_{n+1}}{2^n}$, on a $\|v_n\|_\infty = \frac{1}{2^n}$.

La question est alors :
Existe-t-il des fonctions dont la norme infinie des dérivées croit plus vite que $(n+1)!2^n$?

et la réponse est… oui ! Donc les points de Chebychev ne fonctionnent pas pour toute fonction.

III. b) Meilleurs points d'interpolation

Les points de Chebychev résolvent-ils tous nos soucis ?

$$\|f - P_n f\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}\|v_n\|_\infty$$

3.8 Théorème (admis)

Pour chaque séquence de points $x_i^{(n)}$, il existe au moins une fonction f continue pour laquelle $\lim\limits_{n\to+\infty}\|f - P_n f\|_\infty > 0$

3.9 Théorème (admis)

Pour chaque fonction f continue, il existe une séquence de points $x_i^{(n)}$ t.q. $\lim\limits_{n\to+\infty}\|f - P_n f\|_\infty = 0$

Au programme (Chapitre 1)

Objectif :

Étant donné n+1 points de coordonnées $\underline{x}_i \in \mathbb{R}^2, i \in \{0, \cdots, n\}$ déterminer P le polynôme vérifiant

$$P(x_i) = y_i, \ \forall i \in \{0, \cdots, n\} \text{ où } \underline{x}_i = (x_i, y_i)$$

Plan :

I. Interpolation polynomiale et base de Lagrange

II. Évaluation du polynôme d'interpolation

III. Interpolation d'une fonction

IV. Quelques extensions

a) Interpolation d'Hermite

b) Interpolation multivariée

IV. a) Interpolation d'Hermite

4.1 Définition

Soit $f \in C^M$, $M \in \mathbb{N}^*$. Le problème d'interpolation d'Hermite consiste à déterminer $P_n f \in \mathbb{P}_n$ t.q.

$$P^{(j)}(x_i) = f^{(j)}(x_i), \ \forall i \in \{0, \cdots, n\}, \forall j \in \{0, \cdots, M\}$$

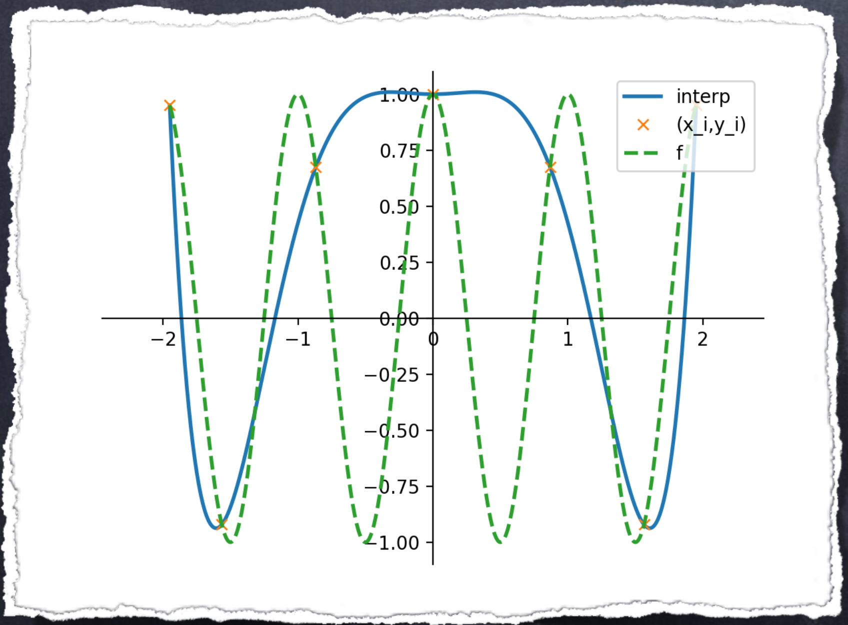
IV. a) Interpolation d'Hermite

4.1 Définition

Soit $f \in C^M$, $M \in \mathbb{N}^*$. Le problème d'interpolation d'Hermite consiste à déterminer $P_n f \in \mathbb{P}_n$ t.q.

$$P^{(j)}(x_i) = f^{(j)}(x_i), \quad \forall i \in \{0, \cdots, n\}, \forall j \in \{0, \cdots, M\}$$

Illustration :



$M = 0$
(Lagrange)

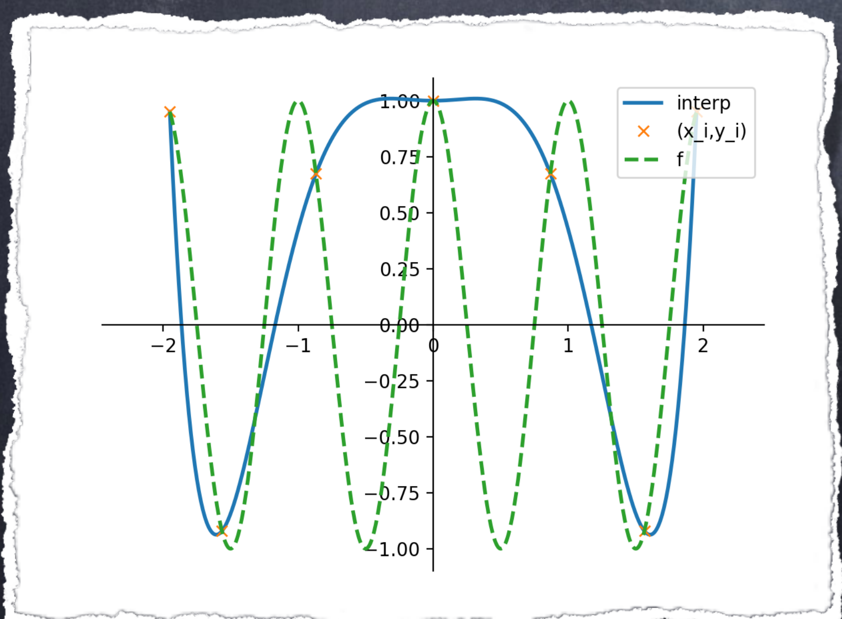
IV. a) Interpolation d'Hermite

4.1 Définition

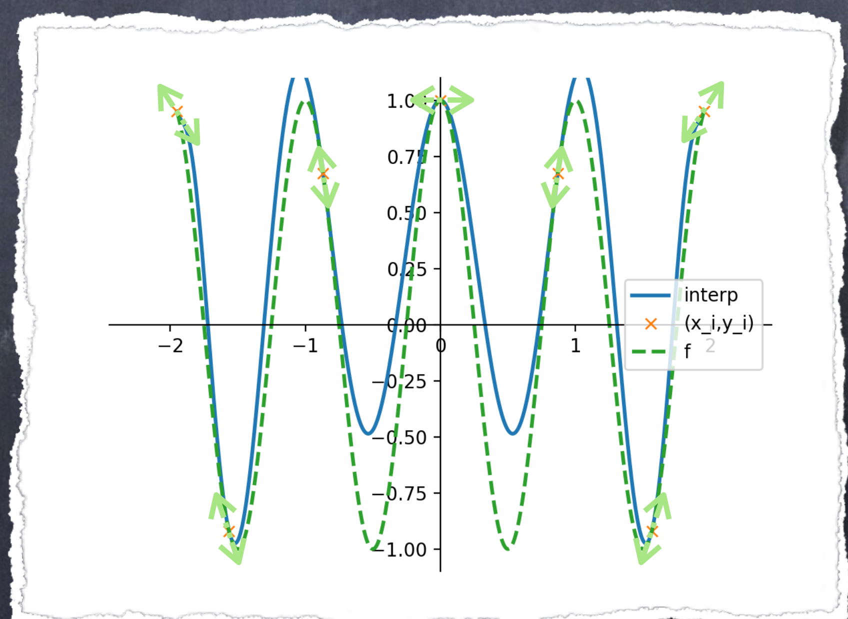
Soit $f \in C^M$, $M \in \mathbb{N}^*$. Le problème d'interpolation d'Hermite consiste à déterminer $P_n f \in \mathbb{P}_n$ t.q.

$$P^{(j)}(x_i) = f^{(j)}(x_i), \quad \forall i \in \{0, \cdots, n\}, \forall j \in \{0, \cdots, M\}$$

Illustration :



$M = 0$
(Lagrange)



$M = 1$

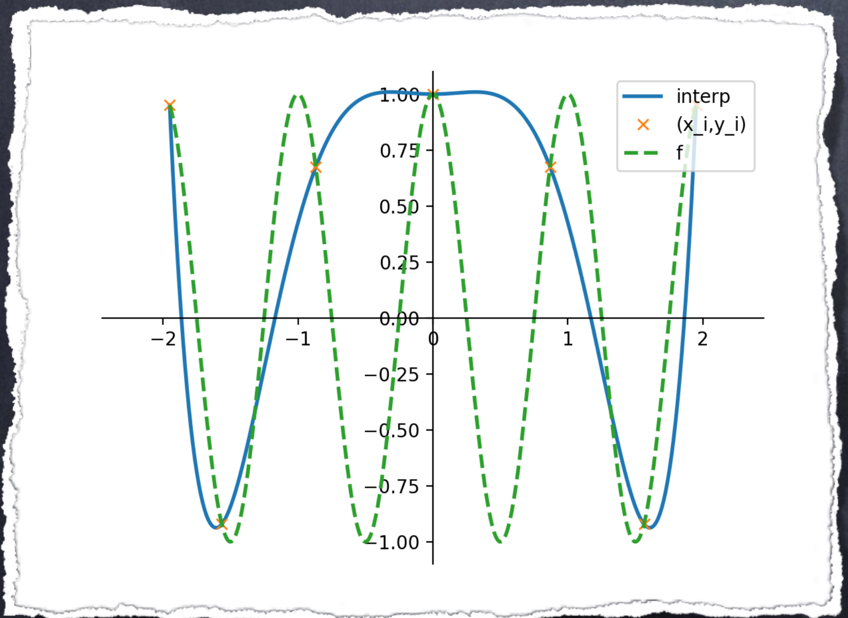
IV. a) Interpolation d'Hermite

4.1 Définition

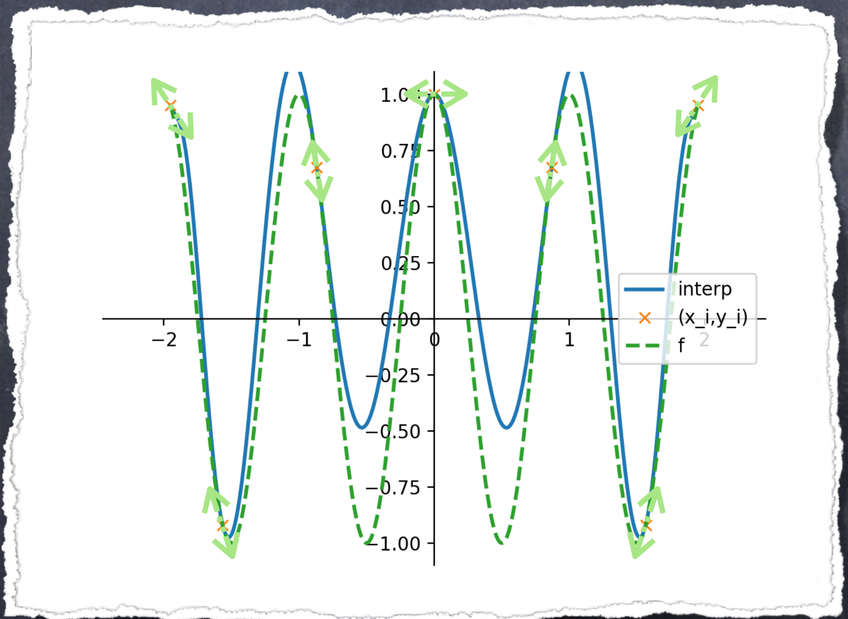
Soit $f \in C^M$, $M \in \mathbb{N}^*$. Le problème d'**interpolation d'Hermite** consiste à déterminer $P_n f \in \mathbb{P}_n$ t.q.

$$P^{(j)}(x_i) = f^{(j)}(x_i), \quad \forall i \in \{0, \cdots, n\}, \forall j \in \{0, \cdots, M\}$$

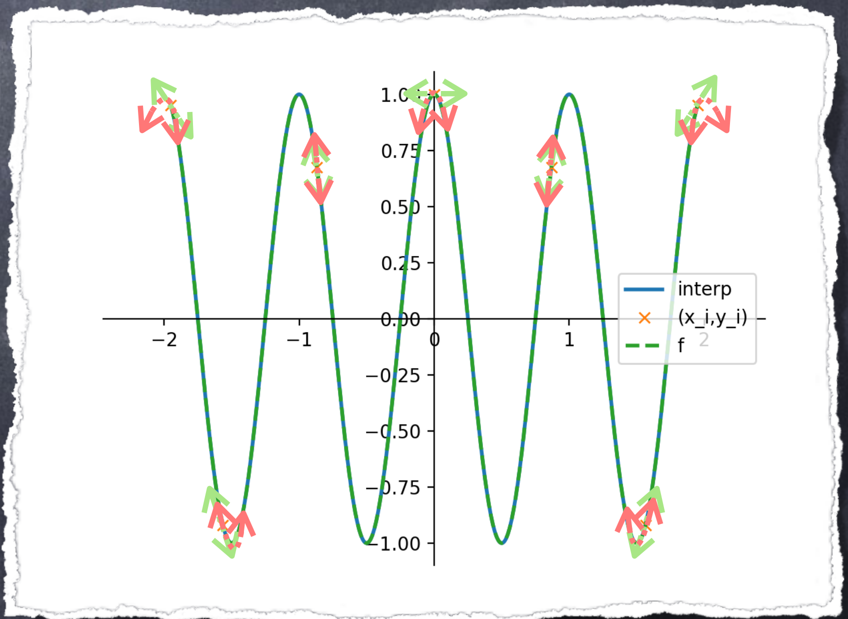
Illustration :



M = 0
(Lagrange)



M = 1



M = 2

IV. a) Interpolation d'Hermite

4.1 Définition

Soit $f \in C^M$, $M \in \mathbb{N}^*$. Le problème d'interpolation d'Hermite consiste à déterminer $P_n f \in \mathbb{P}_n$ t.q.

$$P^{(j)}(x_i) = f^{(j)}(x_i), \quad \forall i \in \{0, \cdots, n\}, \forall j \in \{0, \cdots, M\}$$

4.2 Théorème

Le problème d'interpolation d'Hermite admet une unique solution $P_n f \in \mathbb{P}_N$ où $N = (M+1) \times (n+1) - 1$

Preuve : au (vrai) tableau ! $\blacksquare$

IV. a) Interpolation d'Hermite

4.1 Définition

Soit $f \in C^M$, $M \in \mathbb{N}^*$. Le problème d'interpolation d'Hermite consiste à déterminer $P_n f \in \mathbb{P}_n$ t.q.

$$P^{(j)}(x_i) = f^{(j)}(x_i), \quad \forall i \in \{0, \cdots, n\}, \forall j \in \{0, \cdots, M\}$$

4.2 Théorème

Le problème d'interpolation d'Hermite admet une unique solution $P_n f \in \mathbb{P}_N$ où $N = (M+1) \times (n+1) - 1$

4.3 Théorème (admis)

$$\forall x \in [a,b], \quad \exists \xi \in [a,b] \text{ t.q. } f(x) - P_n f(x) = \frac{f^{(N+1)}(\xi)}{(N+1)!} \prod_{i=0}^{n} (x - x_i)^M$$

IV. a) Interpolation d'Hermite

Évaluation du polynôme d'interpolation d'Hermite ?

Autrement dit, comment généraliser les algorithmes de Neville-Aitken, Différences Divisées et Horner.

IV. a) Interpolation d'Hermite

Évaluation du polynôme d'interpolation d'Hermite ?

Autrement dit, comment généraliser les algorithmes de Neville-Aitken, Différences Divisées et Horner.

Idée : « Dédoubler » les points d'interpolation !

Pour $M = 1$ par exemple, on pose :

$$y_0 = y_1 = x_0 < y_2 = y_3 = x_1 < \cdots < y_{2n} = y_{2n+1} = x_n$$

IV. a) Interpolation d'Hermite

Évaluation du polynôme d'interpolation d'Hermite ?

Autrement dit, comment généraliser les algorithmes de Neville-Aitken, Différences Divisées et Horner.

Idée : « Dédoubler » les points d'interpolation !

Pour $M = 1$ par exemple, on pose :

$$y_0 = y_1 = x_0 < y_2 = y_3 = x_1 < \cdots < y_{2n} = y_{2n+1} = x_n$$

Notons par $T_k^i \in \mathbb{P}_k$ le polynôme d'interpolation vérifiant

$(T_k^i)^{(j_l)}(y_l) = f^{(j_l)}(y_l)\ \forall l \in \{i, \cdots, i+k\}, \forall j_l \in \{0, M_l = (l-i) \bmod 2\}$

où $T_0^i(x) = f(y_i)$

IV. a) Interpolation d'Hermite

Évaluation du polynôme d'interpolation d'Hermite ?

Idée : Posons $y_0 = y_1 = x_0 < y_2 = y_3 = x_1 < \cdots < y_{2n} = y_{2n+1} = x_n$
(M=1)

4.4 Théorème

On a la relation suivante entre les T_k^i

$$T_{k+1}^i(x) = \begin{cases} \dfrac{(y_{i+k+1} - x)T_k^i(x) - (y_i - x)T_k^{i+1}(x)}{(y_{i+k+1} - y_i)} & \text{si} \quad y_{i+k+1} \neq y_i \\ T_k^i(x) + f'(y_i)(x - y_i) & \text{sinon} \end{cases}$$

Preuve : au (vrai) tableau ! ■

IV. a) Interpolation d'Hermite

Évaluation du polynôme d'interpolation d'Hermite ?

Idée : Posons $y_0 = y_1 = x_0 < y_2 = y_3 = x_1 < \cdots < y_{2n} = y_{2n+1} = x_n$
(M=1)

4.4 Théorème

On a la relation suivante entre les T_k^i

$$T_{k+1}^i(x) = \left| \begin{array}{ll} \dfrac{(y_{i+k+1} - x)T_k^i(x) - (y_i - x)T_k^{i+1}(x)}{(y_{i+k+1} - y_i)} & \text{si} \quad y_{i+k+1} \neq y_i \\ T_k^i(x) + f'(y_i)(x - y_i) & \text{sinon} \end{array} \right.$$

Remarque :

On peut généraliser le résultat pour M > 1.

IV. a) Interpolation d'Hermite

Algorithme de Neville-Aitken (Hermite) :

$$\begin{array}{c|l}
y_0 & f(y_0) = T_0^0(a) \\
y_1 & f(y_1) = T_0^1(a) \\
y_2 & f(y_2) = T_0^0(a) \\
\vdots & \qquad \vdots \\
y_{2n+1} & f(y_{2n+1}) = T_0^{2n+1}(a)
\end{array}$$

IV. a) Interpolation d'Hermite

Algorithme de Neville-Aitken (Hermite) :

$$
\begin{array}{c|l}
y_0 & f(y_0) = T_0^0(a) \\
y_1 & f(y_1) = T_0^1(a) \\
y_2 & f(y_2) = T_0^0(a) \\
\vdots & \quad\vdots \\
y_{2n+1} & f(y_{2n+1}) = T_0^{2n+1}(a)
\end{array}
$$

$$\boxed{T_1^0(a) = f(y_0) + f'(y_0)(a - y_0)}$$

IV. a) Interpolation d'Hermite

Algorithme de Neville-Aitken (Hermite) :

$$
\begin{array}{c|l}
y_0 & f(y_0) = T_0^0(a) \longrightarrow T_1^0(a) = f(y_0) + f'(y_0)(a - y_0) \\
y_1 & f(y_1) = T_0^1(a) \longrightarrow T_1^1(a) = \dfrac{(y_2 - a)f(y_1) - (y_1 - a)f(y_2)}{(y_2 - y_1)} \\
y_2 & f(y_2) = T_0^0(a) \\
\vdots & \qquad \vdots \\
y_{2n+1} & f(y_{2n+1}) = T_0^{2n+1}(a)
\end{array}
$$

IV. a) Interpolation d'Hermite

Algorithme de Neville-Aitken (Hermite) :

$$
\begin{array}{l|lll}
y_0 & f(y_0) = T_0^0(a) & T_1^0(a) = f(y_0) + f'(y_0)(a - y_0) & \cdots \\
y_1 & f(y_1) = T_0^1(a) & T_1^1(a) = \dfrac{(y_2 - a)f(y_1) - (y_1 - a)f(y_2)}{(y_2 - y_1)} & \cdots \\
y_2 & f(y_2) = T_0^2(a) & & \\
\vdots & \vdots & \ddots & \\
y_{2n+1} & f(y_{2n+1}) = T_0^{2n+1}(a) & &
\end{array}
$$

IV. a) Interpolation d'Hermite

Évaluation du polynôme d'interpolation d'Hermite ?

Idée : Posons $y_0 = y_1 = x_0 < y_2 = y_3 = x_1 < \cdots < y_{2n} = y_{2n+1} = x_n$

4.4 Théorème

On a la relation suivante entre les T_k^i

$$T_{k+1}^i(x) = \begin{cases} \dfrac{(y_{i+k+1} - x)T_k^i(x) - (y_i - x)T_k^{i+1}(x)}{(y_{i+k+1} - y_i)} & \text{si} \quad y_{i+k+1} \neq y_i \\ T_k^i(x) + f'(y_i)(x - y_i) & \text{sinon} \end{cases}$$

Remarque :

À l'aide de cette relation, on peut également généraliser l'algorithme des différences divisées.

IV. a) Interpolation multivariée

Étant donné $(n+1)$ points $(x_i, y_i, z_i)_{i=0,\cdots,n} \in \mathbb{R}^3$, déterminer P un polynôme de deux variables vérifiant :

$$P(x_i, y_i) = z_i \quad \forall i \in \{0, \cdots, n\}$$

IV. a) Interpolation multivariée

Étant donné (n+1) points $(x_i, y_i, z_i)_{i=0,\cdots,n} \in \mathbb{R}^3$, déterminer P un polynôme de deux variables vérifiant :

$$P(x_i, y_i) = z_i \quad \forall i \in \{0, \cdots, n\}$$

De manière général, ce problème n'est pas évident et il faut en particulier définir la notion de polynôme de degré n en 2D...

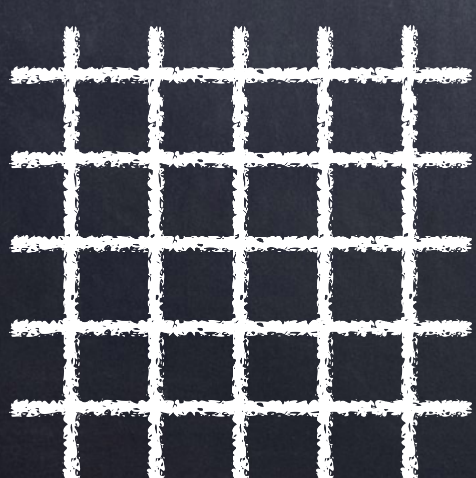
IV. a) Interpolation multivariée

Étant donné $(n+1)$ points $(x_i, y_i, z_i)_{i=0,\cdots,n} \in \mathbb{R}^3$, déterminer P un polynôme de deux variables vérifiant :

$$P(x_i, y_i) = z_i \quad \forall i \in \{0, \cdots, n\}$$

De manière général, ce problème n'est pas évident et il faut en particulier définir la notion de polynôme de degré n en 2D…

Plus simplement, considérons le cas de données sur une grille cartésienne :



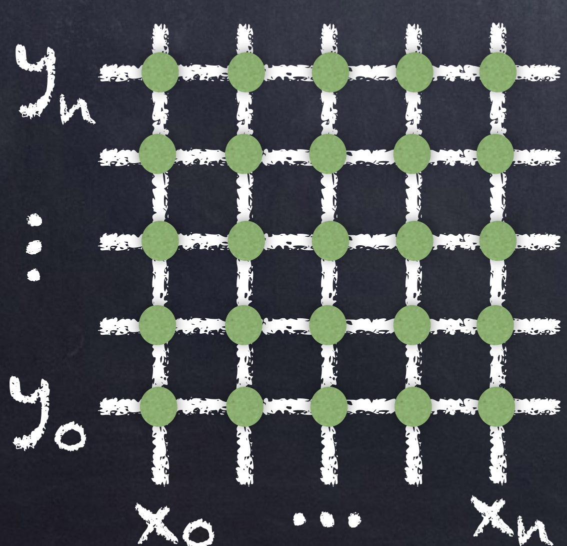
IV. a) Interpolation multivariée

Étant donné (n+1) points $(x_i, y_i, z_i)_{i=0,\cdots,n} \in \mathbb{R}^3$, déterminer P un polynôme de deux variables vérifiant :

$$P(x_i, y_i) = z_i \quad \forall i \in \{0, \cdots, n\}$$

De manière général, ce problème n'est pas évident et il faut en particulier définir la notion de polynôme de degré n en 2D...

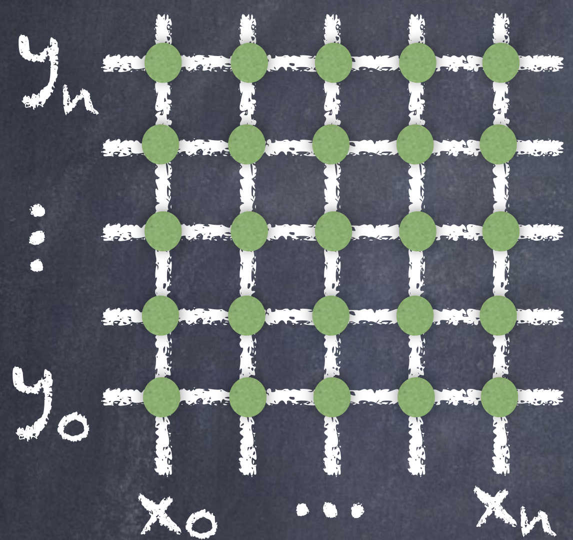
Plus simplement, considérons le cas de données sur une grille cartésienne :



$$\boxed{P(x_k, y_l) = z_{k,l} \quad \forall (k,l) \in \{0, \cdots, n\} \times \{0, \cdots, n\}}$$

IV. a) Interpolation multivariée

Étant donné $(n+1)^2$ points, déterminer P un polynôme de deux variables vérifiant :



$$P(x_k, y_l) = z_{k,l} \quad \forall (k,l) \in \{0, \cdots, n\} \times \{0, \cdots, n\}$$

4.5 Définition

On appelle $\mathbb{Q}_n$ l'espace de vectoriel définie par $\mathbb{Q}_n = \text{vect}\{x^i y^j, \quad \forall (i,j) \in \{0, \cdots, n\}^2\}$

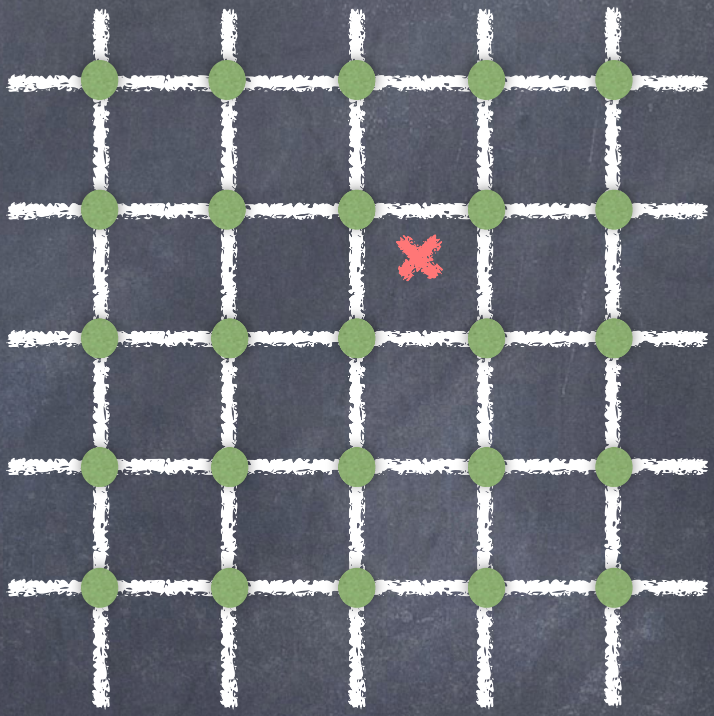
4.6 Théorème (admis)

Le problème d'interpolation admet une unique solution dans $\mathbb{Q}_n$.

IV. a) Interpolation multivariée

Évaluation du polynôme d'interpolation ?

On souhaite évaluer P en un point (x^*, y^*).

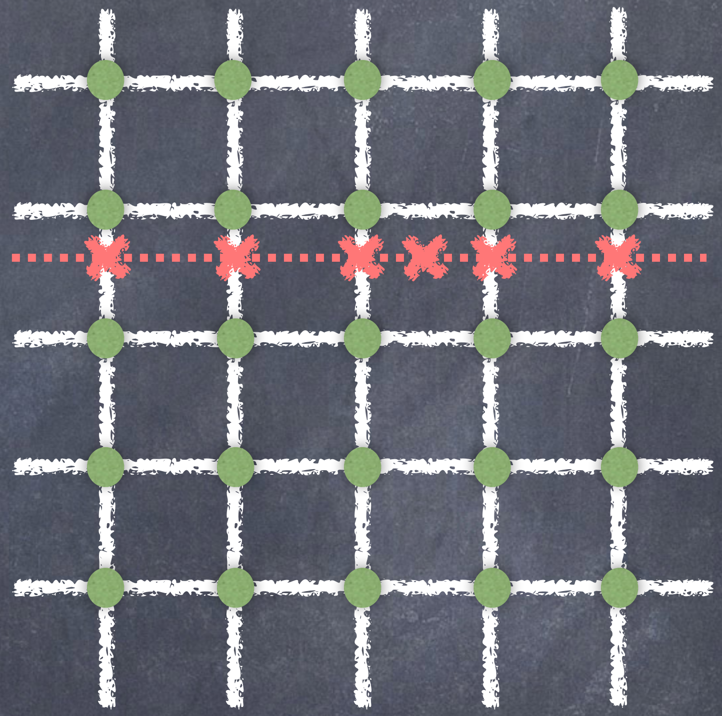


IV. a) Interpolation multivariée

Évaluation du polynôme d'interpolation ?

On souhaite évaluer P en un point (x^*, y^*).

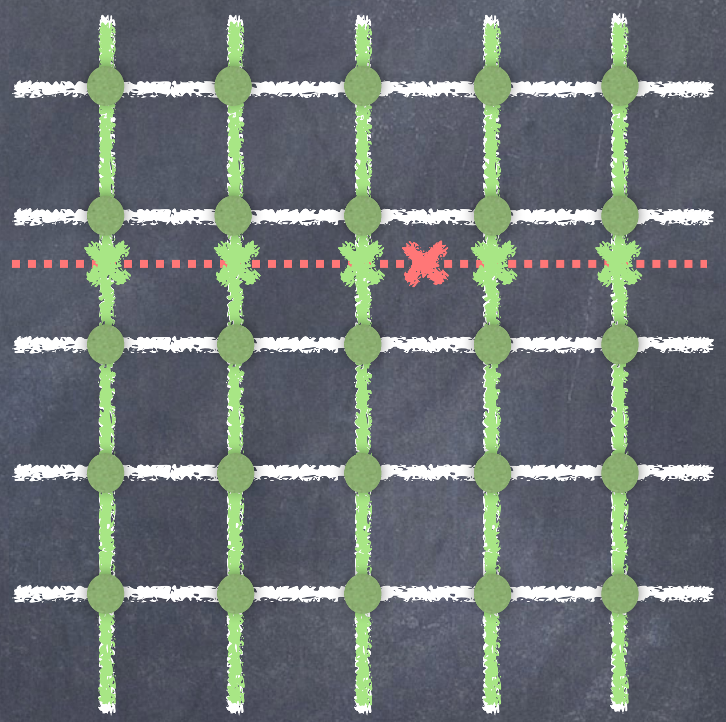
Pour ce faire, on commence par évaluer $P(x_i, y^*)$



IV. a) Interpolation multivariée

Évaluation du polynôme d'interpolation ?

On souhaite évaluer P en un point (x^*, y^*).



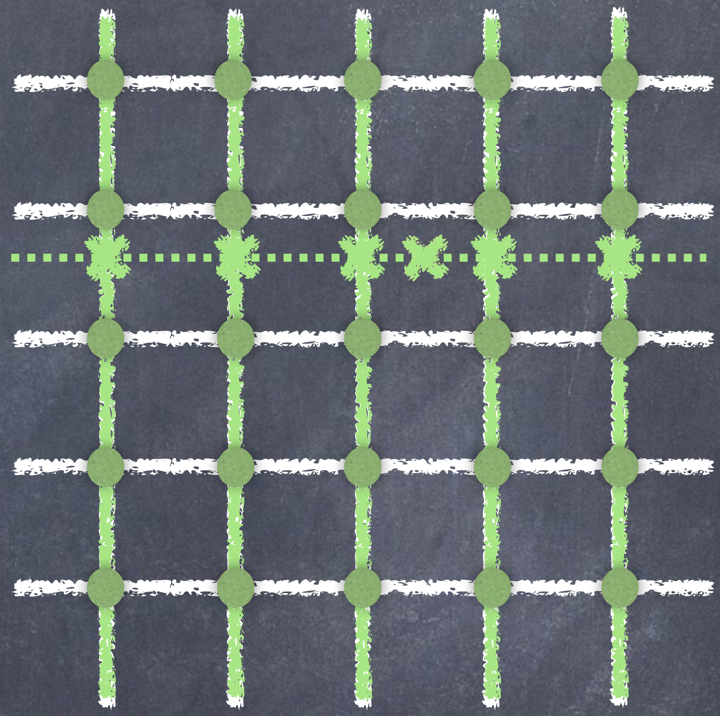
Pour ce faire, on commence par évaluer $P(x_i, y^*)$

(en remarquant que $P(x_i, y) \in \mathbb{P}_n$)

IV. a) Interpolation multivariée

Évaluation du polynôme d'interpolation ?

On souhaite évaluer P en un point (x^*, y^*).



Pour ce faire, on commence par évaluer $P(x_i, y^*)$

(en remarquant que $P(x_i, y) \in \mathbb{P}_n$)

Enfin, connaissant $P(x_i, y^*)$ et en utilisant le fait que $P(x, y^*) \in \mathbb{P}_n$ on peut déduire $P(x^*, y^*)$

Plan détaillé du chapitre 1

Plan :

I. Interpolation polynomiale et base de Lagrange

II. Évaluation du polynôme d'interpolation

a) Le schéma de Neville-Aitken

b) Les différences divisées et l'algorithme d'Horner

III. Interpolation d'une fonction

a) Erreur d'interpolation

b) Meilleurs points d'interpolation

IV. Quelques extensions

a) Interpolation d'Hermite

b) Interpolation multivariée