TD-cours n°2 Analyse Numérique

EMA

Introduction : Interpolation et approximation

EMA

Interpolation et approximation

Interpolation : on cherche une courbe d'un type répertorié, passant exactement par $n+1$ points.

On dispose d'un support de $n+1$ points $\left(P_k = \left(x_k, y_k\right)\right)_{0 \le k \le n}$

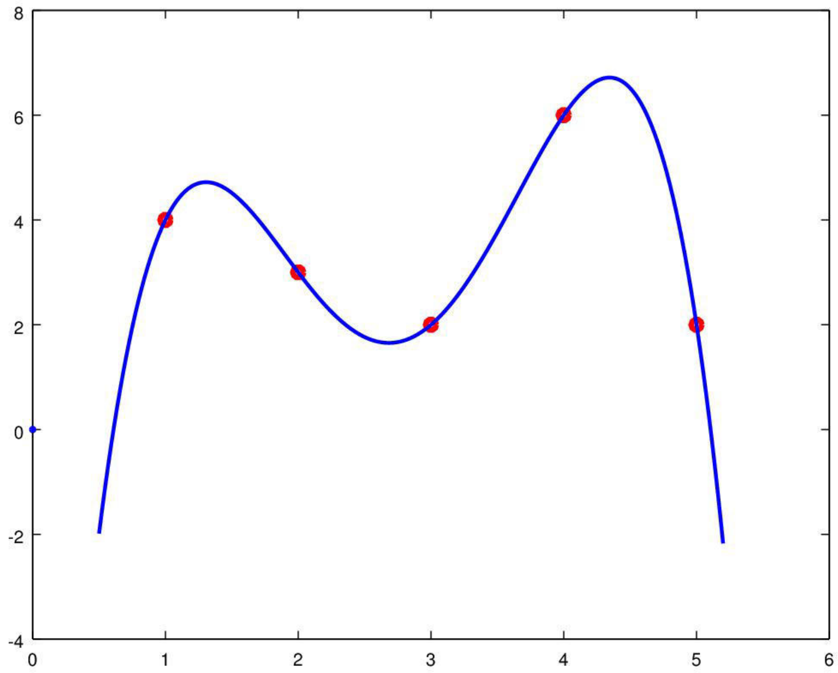
Avec des x_k *2 à 2 distincts :* $x_0 < x_1 < \cdots < x_n$.

Approximation : on cherche une courbe d'un type répertorié, passant « au plus près » des $n+1$ points.

Extrapolation : on propose un point $M = (x, y)$ en calculant y en fonction de x et des $(P_k)_{0 \le k \le n}$.

Interpolation polynômiale

Les courbes que l'on cherche sont des polynômes, passant exactement par les $n + 1$ points.



Un polynôme de degré 4 passant par 5 points.

On pourrait chercher des polynômes trigonométriques, des polynômes exponentiels, etc .

Polynôme d'interpolation

Si on dispose d'un support de $n+1$ points $\left(P_k=\left(x_k, y_k\right)\right)_{0\leq k\leq n}$

il existe un seul polynôme $L(X)=a_0+a_1X+a_2X^2+\cdots+a_nX^n$

tel que $\forall k\in\{0,\ldots,n\}\quad L(x_k)=y_k$.

Les x_k sont ordonnés de façon strictement croissante.

Démonstration non exigible.

- On définit une application clairement linéaire $\Phi:\ \mathbb{R}_n[\mathrm{X}]\to\mathbb{R}^{n+1}$

avec $\Phi(P)=\left(P(x_0),P(x_1),\cdots,P(x_n)\right)$.

Son noyau est réduit à $\left\{0_{\mathbb{R}_n[\mathrm{X}]}\right\}$ seul polynôme à s'annuler $n+1$ fois.

Les espaces ont même dimension, Φ est bijective : existence et unicité.

- Pour chaque $k\in\{0,\ldots,n\}\quad L(x_k)=y_k$

donne une équation : $a_0+a_1x_k+a_2{x_k}^2+\cdots+a_n{x_k}^n=y_k$

On a $n+1$ équations à $n+1$ inconnues.

Fonctions disponibles dans Python

Pour les polynômes,
on peut choisir entre :

- Dans numpy :

poly,   roots

polyval (P, x)

```
import numpy as np
L = [ 1 , 2 , 3]
Pol1 =  np.poly( L)
print('Pol1 = ' , Pol1)
Sol1 = np.roots( Pol1)
print('Sol1 = ' , Sol1)
V1 = np.polyval(Pol1,0)
```

Réponses :

Pol1 = [1 -6 11 -6]

Sol1 = [3. 2. 1.]

Ou :

- Dans numpy.polynomial.polynomial

polyfromroots, polyroots

polyval (x , P) (attention à l'ordre des degrés !)

```
import numpy.polynomial.polynomial as poly
L = [ 1 , 2 , 3]
Pol2= poly.polyfromroots(L)
print('Pol2 = ',Pol2)
Sol2=poly.polyroots(Pol2)
print('Sol2 = ',Sol2)
V2 = poly.polyval(0,Pol2)
```

Réponses :

Pol2 = [-6. 11. -6. 1.]

Sol2 = [1. 2. 3.]

I^{ère} Partie

EMA

Méthode 1 : avec la matrice de Vandermonde MV

$\forall k \in \{0,\ldots,n\}$

$L(x_k) = y_k$

donne le système

$MV \times A = Y$:

qu'on résout en inversant MV.

$$\begin{pmatrix} 1 & x_0 & x_0{}^2 & \cdots & x_0{}^n \\ 1 & x_1 & x_1{}^2 & \cdots & x_1{}^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n-1} & x_{n-1}{}^2 & \cdots & x_{n-1}{}^n \\ 1 & x_n & x_n{}^2 & \cdots & x_n{}^n \end{pmatrix} \times \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix}$$

$$\det(MV) = \prod_{0 \le i < j \le n} (x_j - x_i) \neq 0$$

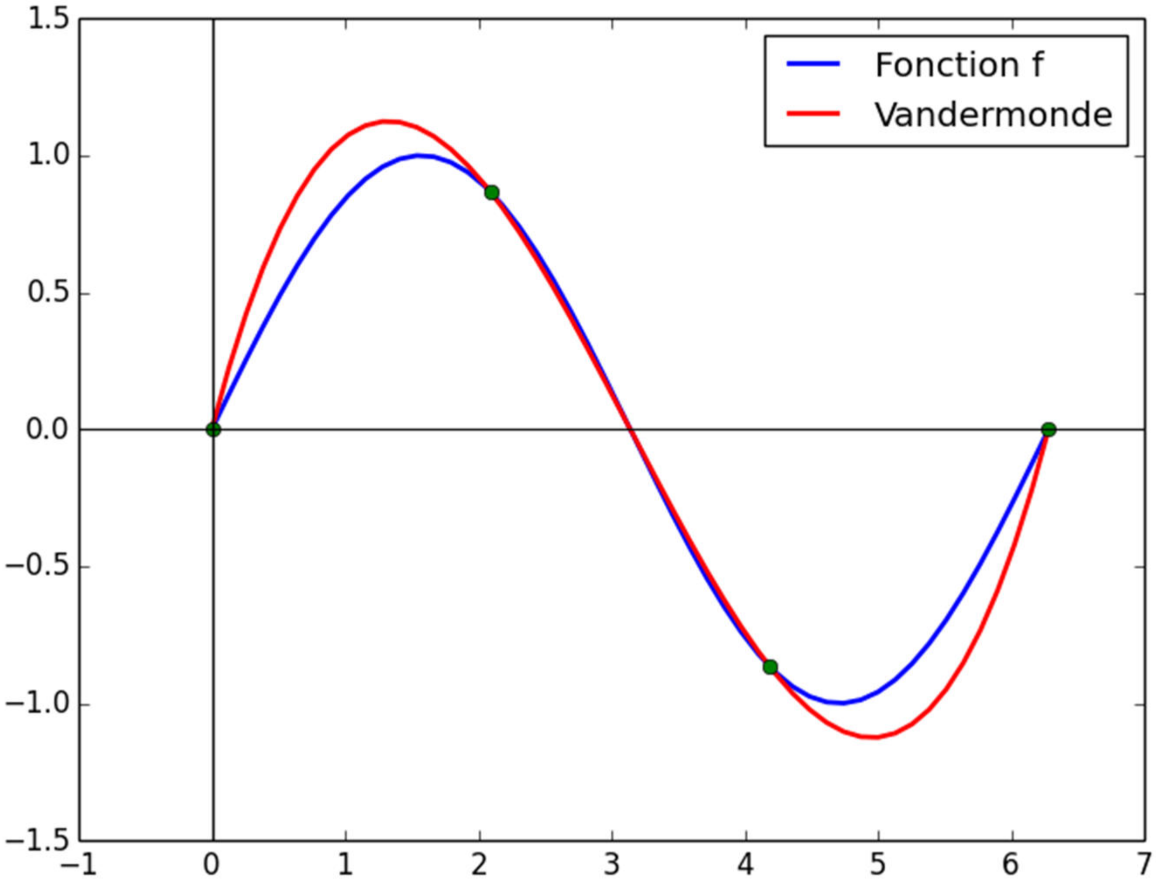
Inconvénient :

- la matrice de Vandermonde est « mal conditionnée », si n est élevé, certains $x_k{}^n$ peuvent être très grands, d'autres très petits (en valeurs absolues). Le calcul de son inverse est alors très sensible aux arrondis donc aux incertitudes de calcul.

Méthode avec matrice de Vandermonde

Réponse :
[0.   1.86073502
 -0.88843553   0.09426594]

Polynôme de degré 3, interpolant la fonction sinus en 4 points équidistants.



Instabilité par rapport aux données

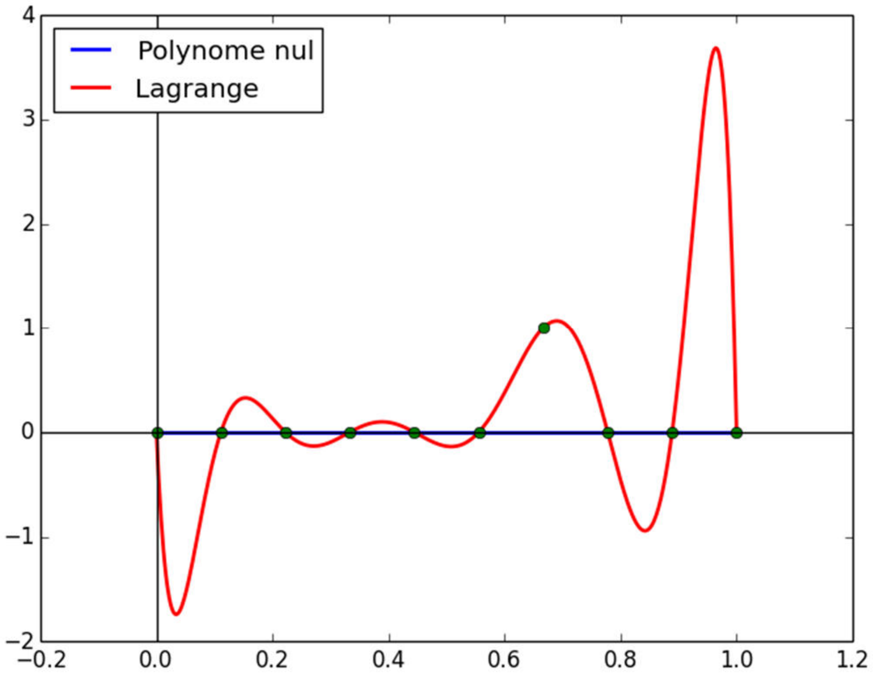
Dès qu'on perturbe un point, le polynôme est profondément modifié, même en dehors de la proximité de la modification.

Exemple : le polynôme qui passe par les 10 points équidistants sur [0,1] d'ordonnées $y_i = 0$ est le polynôme constant nul.

On modifie le 7ème point sur 10, en posant :

$y_7 = 1.$

On obtient un polynôme gravement perturbé, même aux points d'abscisses éloignées de $x_7 = 0{,}7$

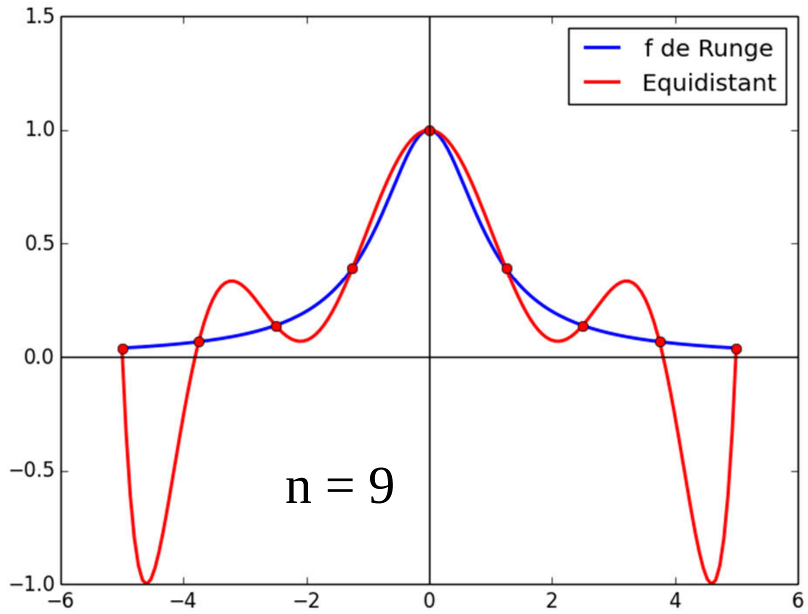


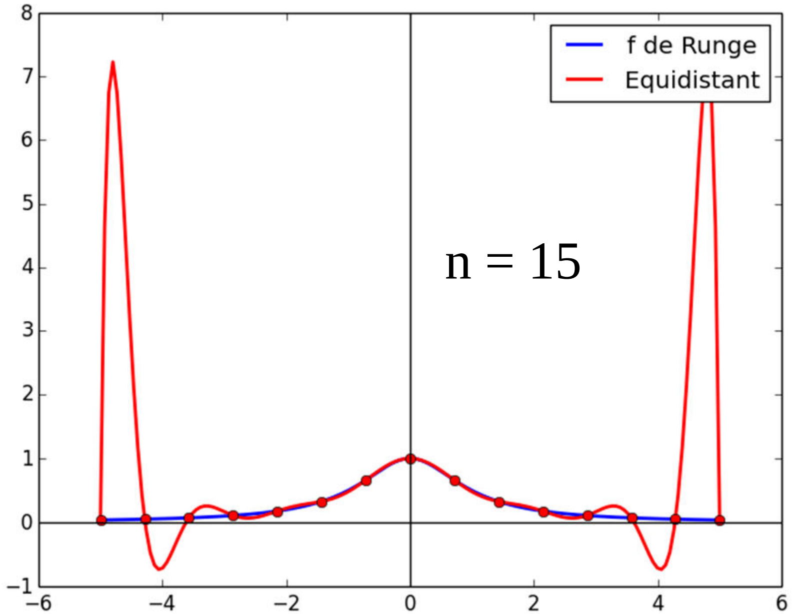
Exemple de Runge

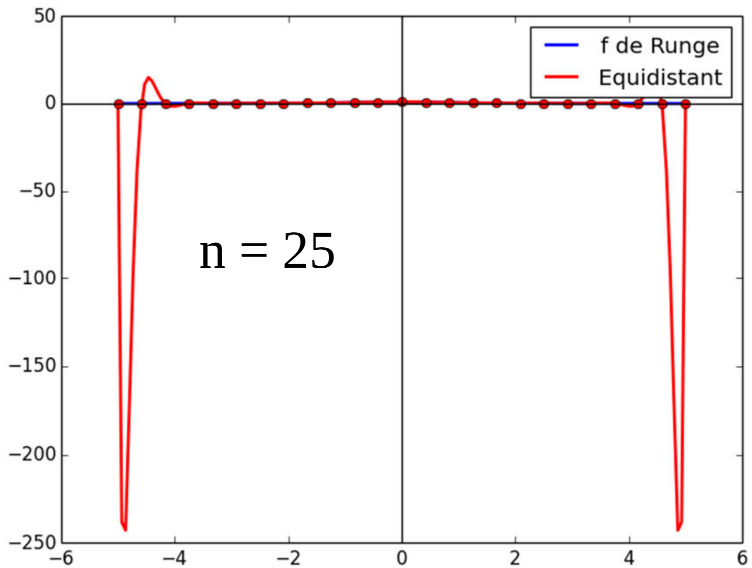
Polynôme de degré $n-1$, interpolant la fonction f en n points équidistants :

$$f(x) = \frac{1}{1+x^2} \quad \text{sur } [-5\ ,\ 5\ ]$$

En bleu la courbe de f
En rouge, Lagrange « équidistant ».







Incertitude de méthode dans la méthode de Lagrange

Démonstration non exigible.

Considérons f de classe C^{n+1} et un x (fixé). Soit $\varepsilon(x) = f(x) - L(x)$, et $\Pi(x) = \prod_{k=0}^{k=n}(x - x_k)$

Si $x \in \{x_0, \cdots, x_n\}$, alors $\varepsilon(x) = 0$. Si $x \neq x_k$ pour tout $k \in \{0, \cdots, n\}$, alors :

il existe une constante K telle que $\varepsilon(x) = \dfrac{K}{(n+1)!}\Pi(x)$, avec $K = \dfrac{(n+1)!\varepsilon(x)}{\Pi(x)}$.

Considérons alors $\varphi(y) = f(y) - L(y) - \dfrac{K}{(n+1)!}\Pi(y)$, fonction C^{n+1} de y.

φ s'annule en $n+2$ points distincts : les $(x_k)_{0 \leq k \leq n}$ et x. Avec le thm de Rolle φ' s'annule en $n+1$ points distincts, par itération φ'' s'annule en n points distincts, etc ... et $\varphi^{(n+1)}$ s'annule en un $c \in [a,b]$ contenant les $(x_k)_{0 \leq k \leq n}$ et x.

Mais $\varphi^{(n+1)}(y) = f^{(n+1)}(y) - K$, avec $L^{(n+1)}(y) = 0$ et $\Pi^{(n+1)}(y) = (n+1)!$ donc $K = f^{(n+1)}(c)$.

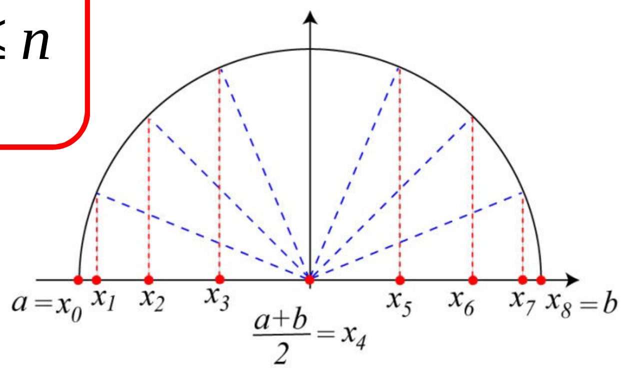
Si f de classe C^{n+1} : $|\varepsilon(x)| = |f(x) - L(x)| = \dfrac{|f^{(n+1)}(c)|}{(n+1)!} \prod_{k=0}^{k=n} |x - x_k|$

où $c \in [a,b]$ segment qui contient tous les x_k et x.

Incertitude de méthode avec le support de Tchebychev

On minimise l'incertitude de méthode en choisissant le support des

$$\boxed{x_k = \frac{1}{2}\left[(b+a)-(b-a)\cos\left(\frac{k\pi}{n}\right)\right] \quad 0 \le k \le n}$$



et alors $|\Pi(x)| \le \dfrac{(b-a)^{n+1}}{2^{n+1}}$. (admis)

$$\boxed{|\varepsilon(x)| = |f(x) - T(x)| \le \frac{\left|f^{(n+1)}(c)\right|}{(n+1)!}\,\frac{(b-a)^{n+1}}{2^{n+1}}}$$

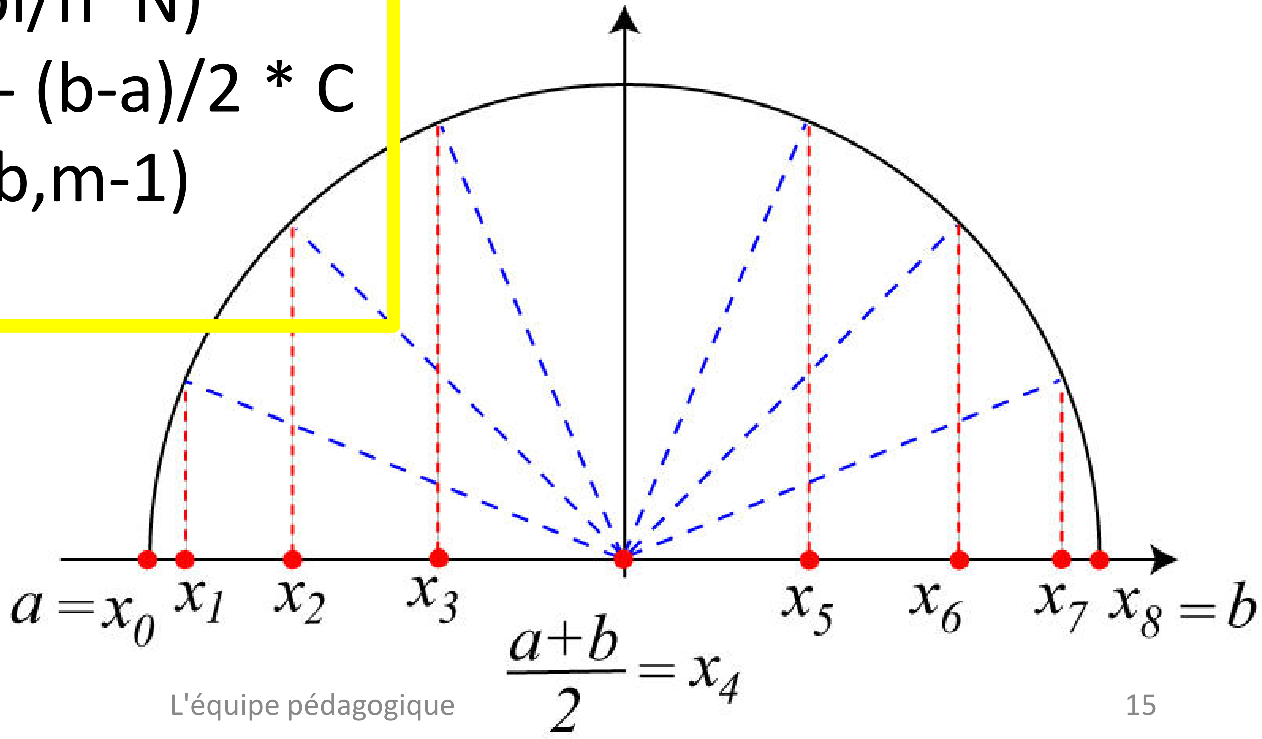
Support de Tchebychev

On obtient ainsi un procédé plus stable, en imposant un support d'interpolation particulier dans $[a, b]$:

```
def Tchebychev(a,b,n):
    N = np.arange(n+1)
    C = np.cos(np.pi/n*N)
    return (a+b)/2 - (b-a)/2 * C
xe = Tchebychev(a,b,m-1)
ye = f(xe)
```

$$x_k = \frac{a+b}{2} + \frac{a-b}{2}\cos\left(\frac{\pi k}{n}\right)$$

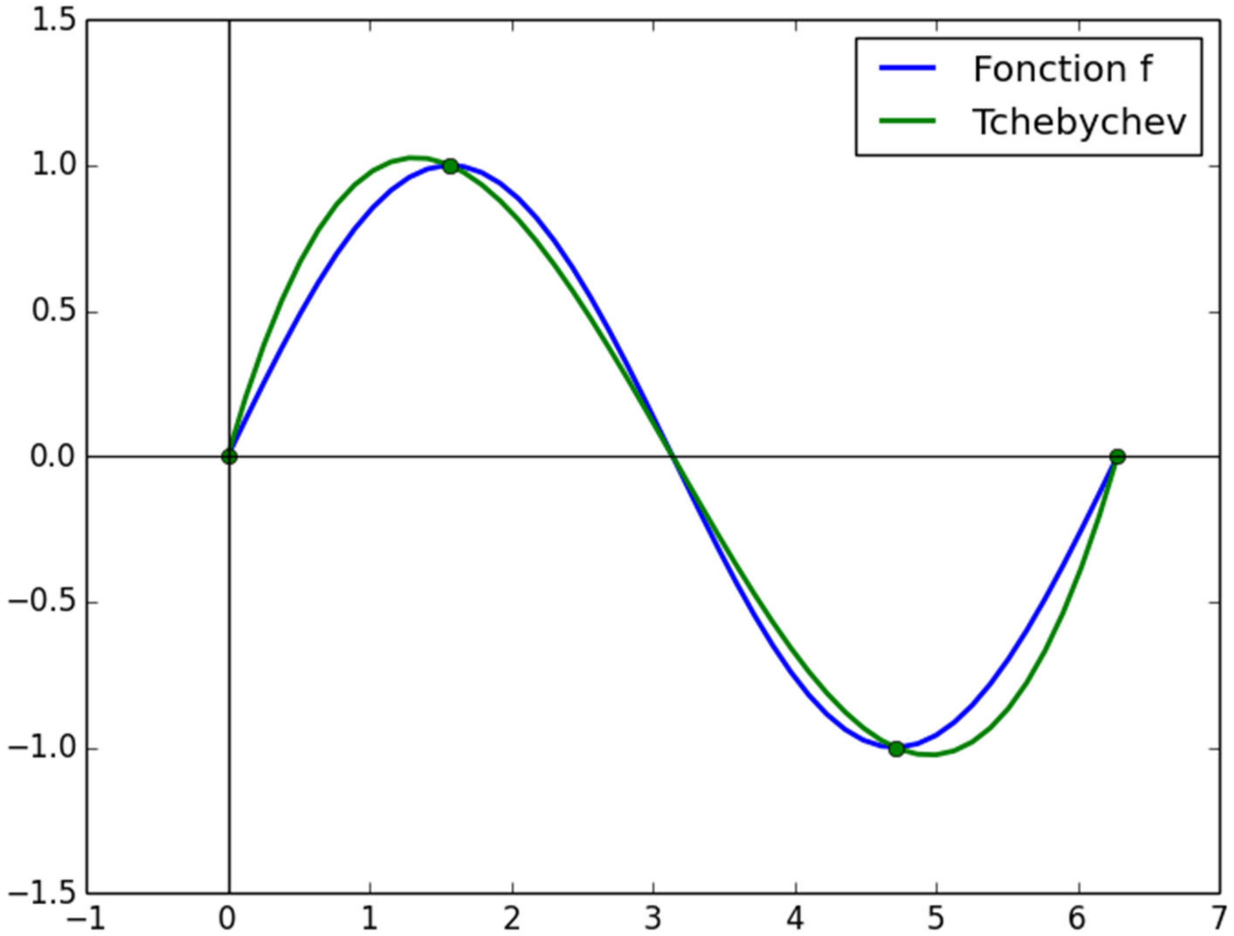
où $k \in \{0, \dots, n\}$



Méthode avec support de Tchebychev

Réponse :
[0.   1.69765273
-0.81056947 0.08600409]

Polynôme de degré 3, interpolant la fonction sinus en 4 points du support de Tchebychev.

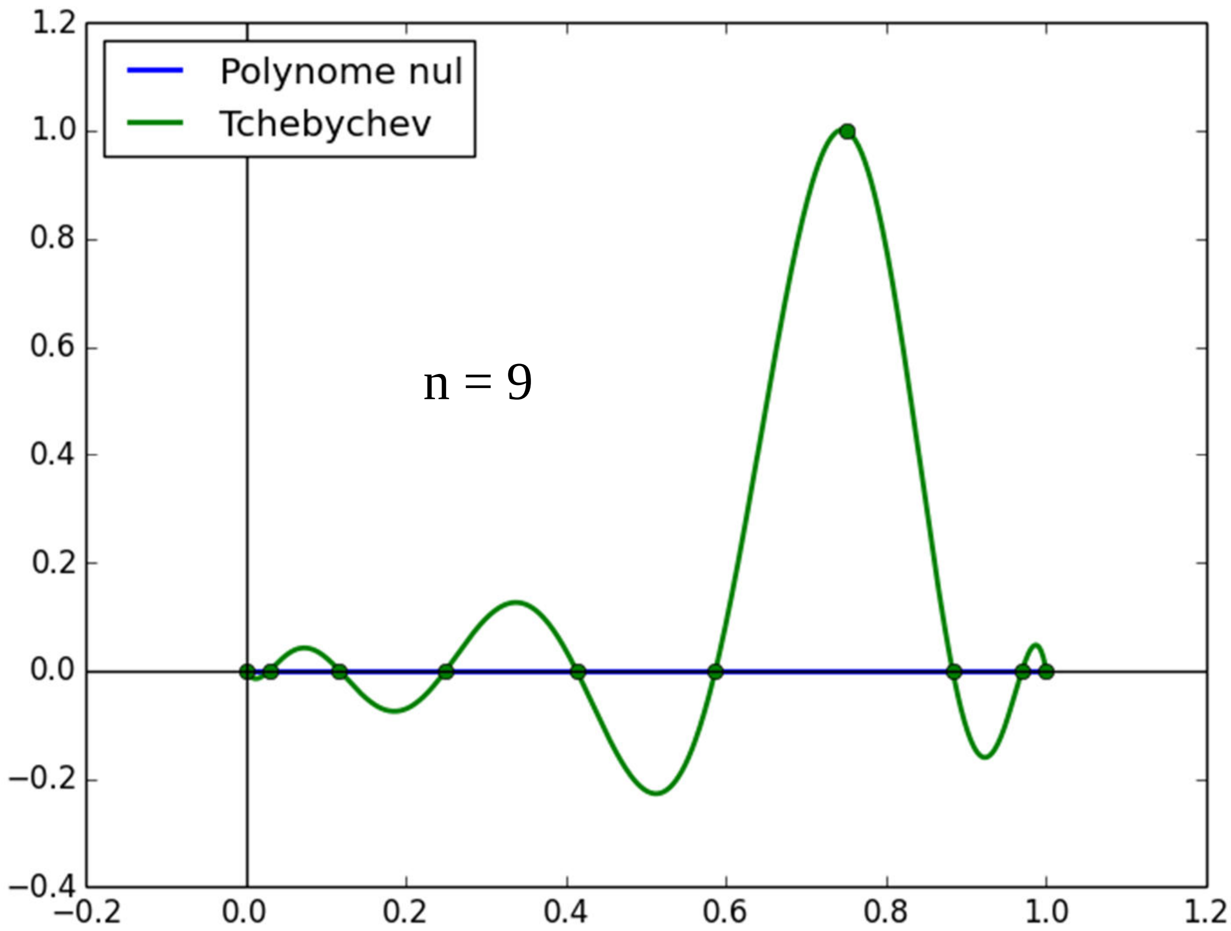


Support de Tchebychev

Meilleure stabilité par rapport aux données.

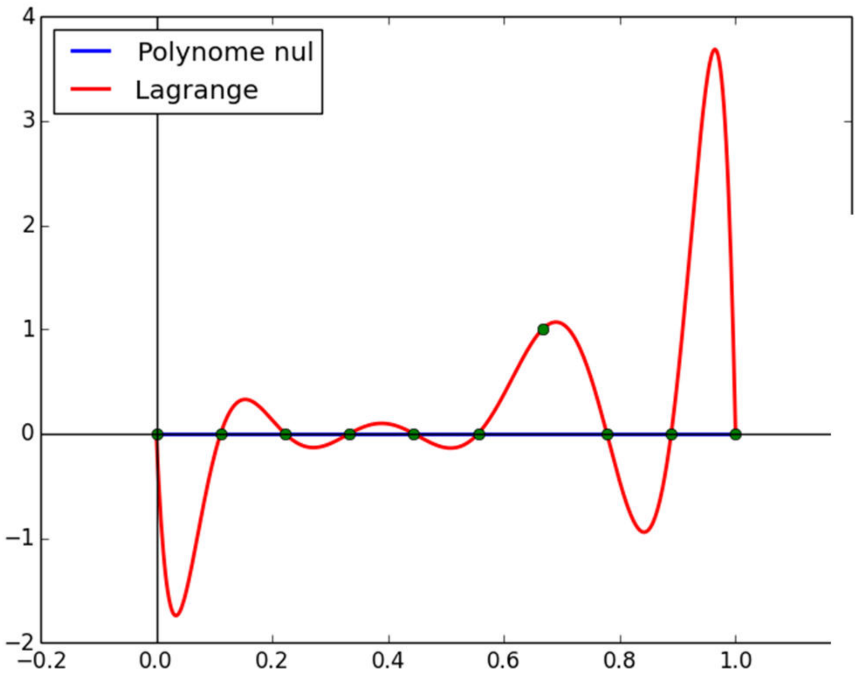
Si on perturbe un point, le polynôme n'est vraiment modifié, qu'à proximité de la modification.

On a modifié le 7ème point sur 10, en posant : $y_7 = 1$.

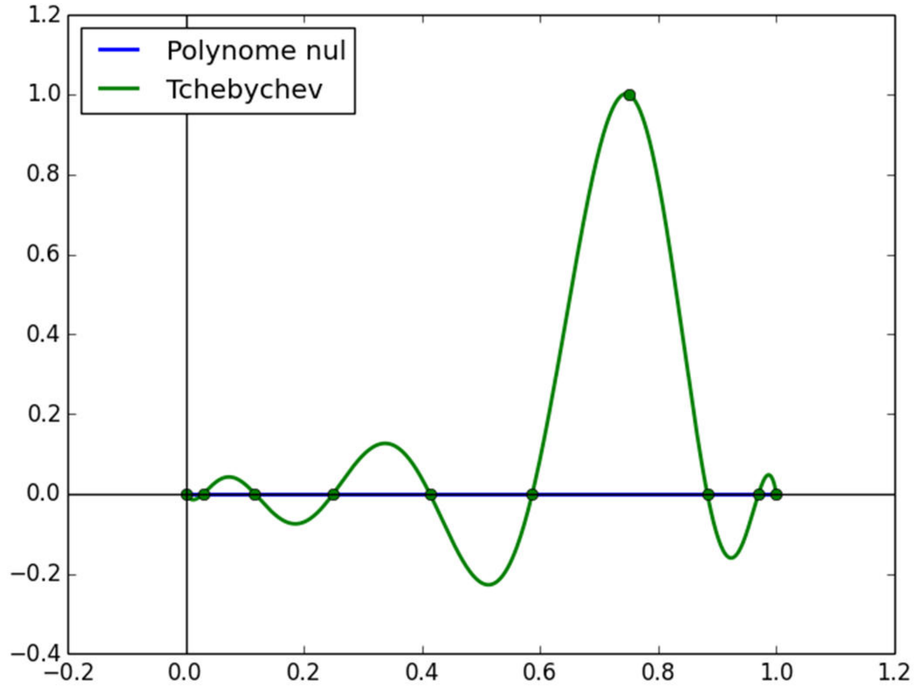


Comparaison selon le support

Effet d'une perturbation d'une donnée :



En rouge, polynôme avec support équidistant.

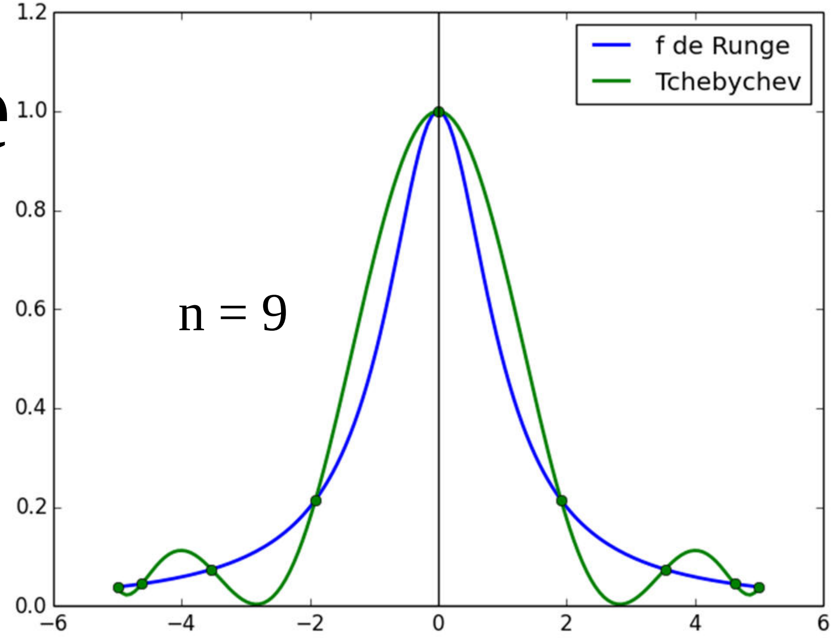


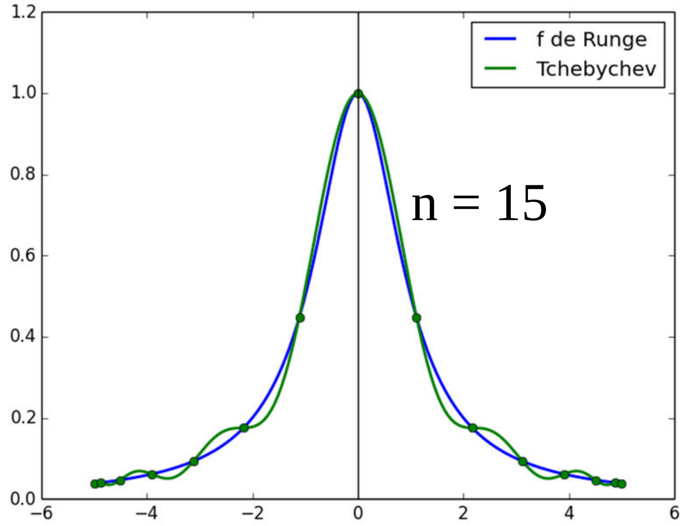
En vert, avec support de Tchebychev.

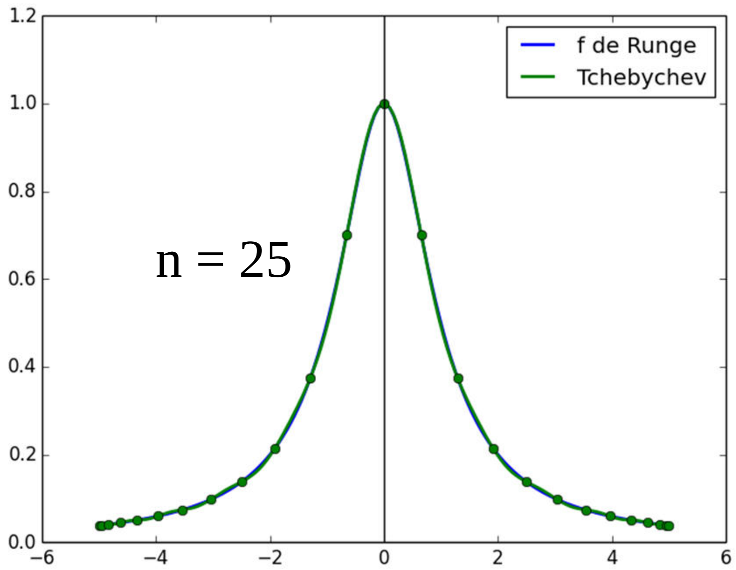
Exemple de Runge

$f(x) = \frac{1}{1+x^2}$ sur $[-5\ ,\ 5\ ]$

En bleu la courbe de f
En vert, support de Tchebychev.





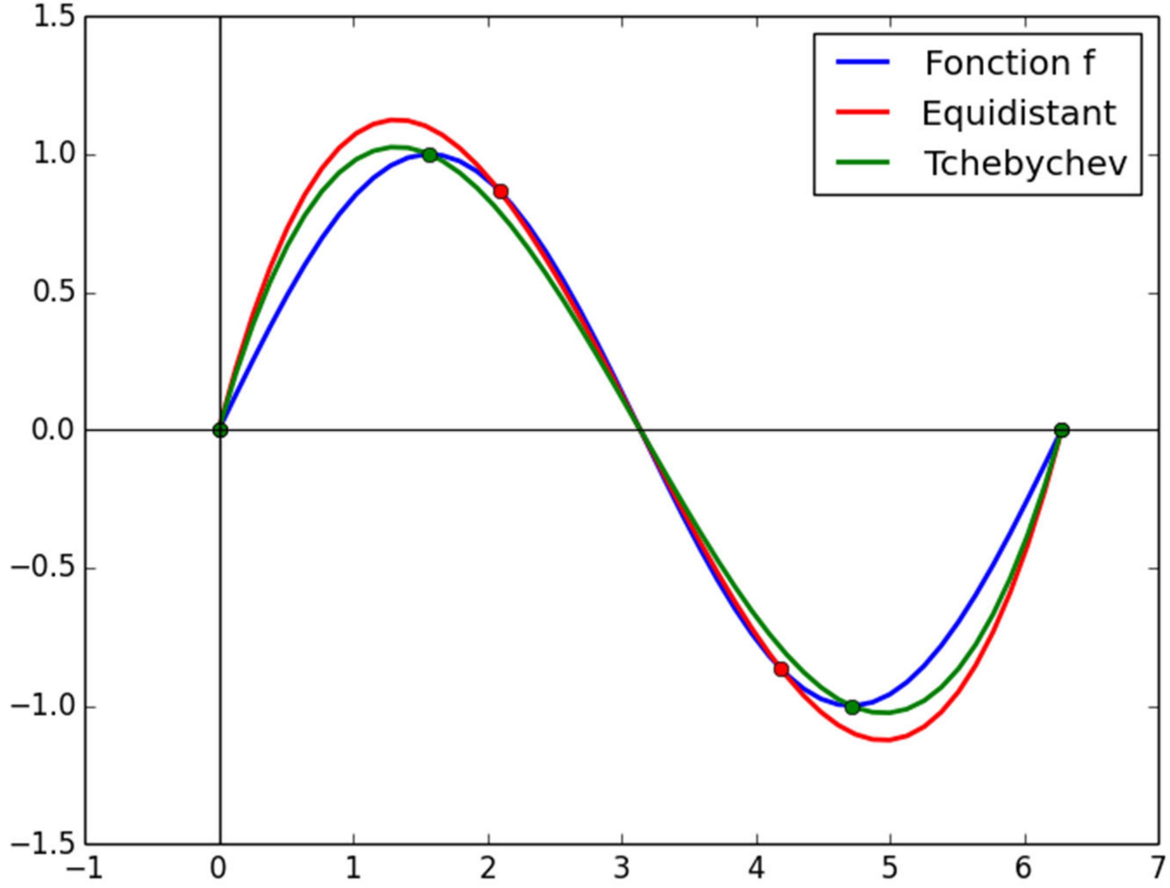


Comparaisons selon le support

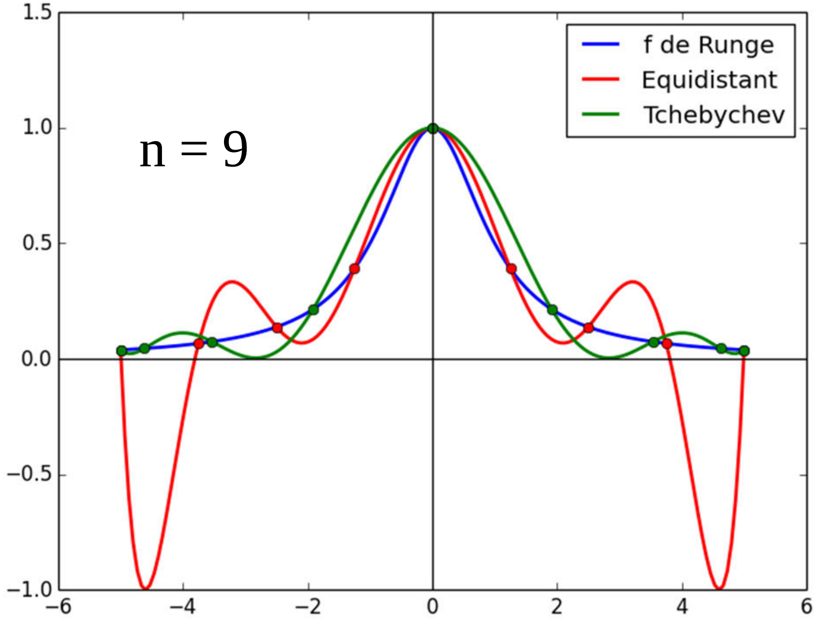
Comparaison :

En rouge, polynôme avec support équidistant.

En vert, avec support de Tchebychev.



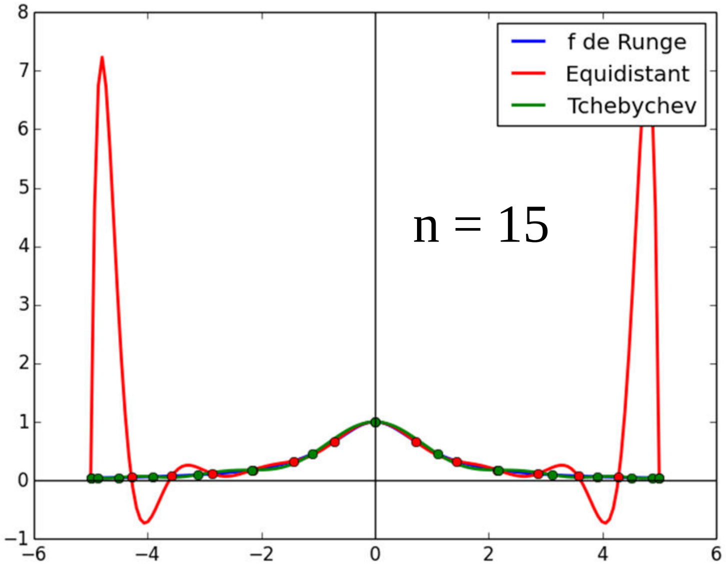
Comparaisons

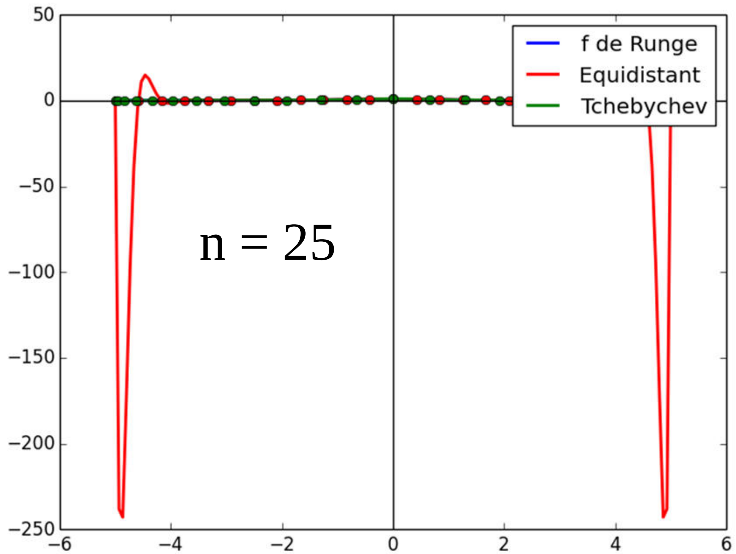


Exemple de Runge

$f(x) = \frac{1}{1+x^2}$ sur $[-5\ ,\ 5\ ]$

En bleu la courbe de f
En rouge, avec support équidistant,
En vert, support de Tchebychev.





II^{ème} Partie

EMA

Méthode 2 : avec les polynômes de Lagrange

Pour k de 0 à n, on définit les polynômes de Lagrange du support

$$\boxed{L_k(x) = \prod_{j=0,\, j\neq k}^{j=n} \frac{x - x_j}{x_k - x_j} \quad \text{avec} \quad L_k(x_j) = 0 \ \text{ si } \ j \neq k \ \text{ et } L_k(x_k) = 1.}$$

On peut remplir une matrice, avec en colonnes C_{k+1} les coefficients du polynome $L_k(x) = \sum_{i=0}^{n} L_{k,i} \ x^i$:

$$ML = \begin{pmatrix} L_{0,0} & L_{1,0} & \cdots & L_{n,0} \\ L_{0,1} & L_{1,1} & \cdots & L_{n,1} \\ \vdots & \vdots & & \vdots \\ L_{0,n-1} & L_{1,n-1} & \cdots & L_{n,n-1} \\ L_{0,n} & L_{1,n} & \cdots & L_{n,n} \end{pmatrix}$$

Le polynôme L est unique, $\boxed{\text{et } L(x) = \sum_{k=0}^{n} y_k L_k(x) \ \text{ convient,}}$

car pour tout j : $L(x_j) = \sum_{k=0}^{n} y_k L_k(x_j) = y_j$ avec $L_k(x_j) = \delta_j^k$.

La colonne A des coefficients de $L(x) = \sum_{i=0}^{n} L_i \ x^i$ est alors $A = ML \times Y$.

Lien entre les méthodes :

$$ML = \begin{pmatrix} L_{0,0} & L_{1,0} & \cdots & L_{n,0} \\ L_{0,1} & L_{1,1} & \cdots & L_{n,1} \\ \vdots & \vdots & & \vdots \\ L_{0,n-1} & L_{1,n-1} & \cdots & L_{n,n-1} \\ L_{0,n} & L_{1,n} & \cdots & L_{n,n} \end{pmatrix} \text{ et } MV = \begin{pmatrix} 1 & x_0 & x_0{}^2 & \cdots & x_0{}^n \\ 1 & x_1 & x_1{}^2 & \cdots & x_1{}^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n-1} & x_{n-1}{}^2 & \cdots & x_{n-1}{}^n \\ 1 & x_n & x_n{}^2 & \cdots & x_n{}^n \end{pmatrix}$$

sont inverses l'une de l'autre. Calculer ML évite d'inverser MV.

Démonstration non exigible.

Pour $Y = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$, $ML \times Y$ est la colonne des coefficients du polynôme L, tel que $L(x_j) = 1$ pour tout $j \in \{0, \cdots, n\}$, c'est le polynome constant $L(x) = 1$.

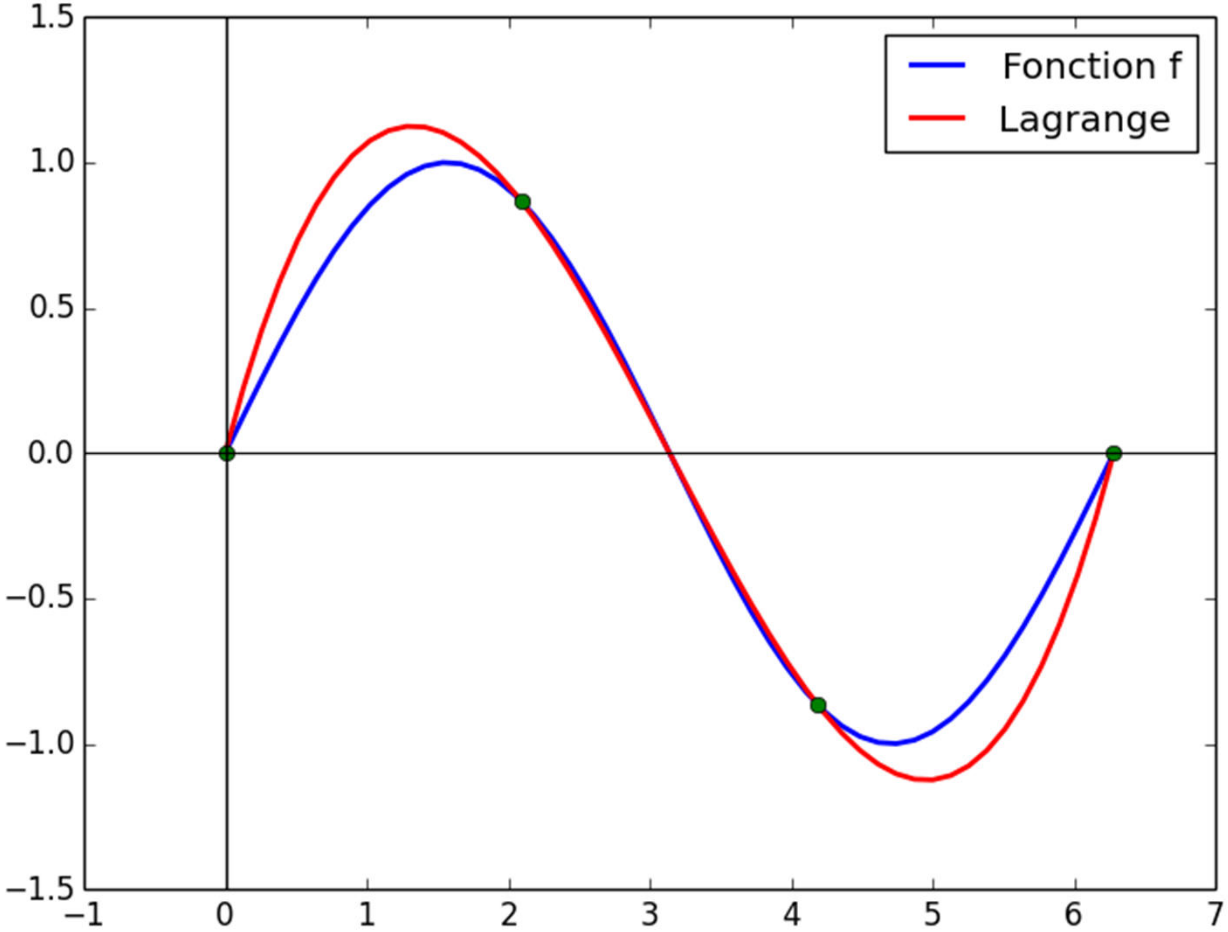
Pour $k \in \{1, \cdots, n\}$ et $Y = \begin{pmatrix} x_0{}^k \\ \vdots \\ x_n{}^k \end{pmatrix}$, $ML \times Y$ est la colonne des coefficients du polynôme L, tel que $L(x_j) = x_j{}^k$ pour tout $j \in \{0, \cdots, n\}$, qui est $L(x) = x^k$.

donc $ML \times MV = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$

Méthode avec polynômes de Lagrange

Réponse :
[0.   1.86073502
 -0.88843553   0.09426594]

Polynôme de degré 3, interpolant la fonction sinus en 4 points équidistants. C’est le même qu’avec la matrice de Vandermonde.



Méthode 2 : avec les polynômes de Lagrange (suite)

Avantages :

- Mieux conditionnée : on n'a pas à inverser de matrice.

La matrice ML est l'inverse de la matrice MV.

- Les polynômes L_k ne dépendent que du support des $(x_j)_{0 \leq j \leq n}$

On peut les calculer une fois pour toute,
avantage pour les expérimentateurs.

Inconvénients :

- Les polynômes obtenus sont instables par rapport aux données.

Dès qu'on perturbe un point, le polynôme est profondément modifié,
même en dehors de la proximité de la modification.

- Si on ajoute un point, on doit recommencer tous les calculs.

D'autres méthodes, comme les différences divisées,
permettent de pallier cette difficulté.

III^{ème} Partie

EMA

Interpolation : cas paramétré

L'idée, assez astucieuse, consiste, à parcourir un système de points (x_k, y_k) avec $1 \leq k \leq n$, dans un certain ordre, mais qui ne soit pas nécessairement à abscisses (x_k) croissantes, et donc qui peut se refermer.

On introduit un paramètre non géométrique $t \in [0,1]$, et une subdivision régulière où $h = \frac{1}{n}$ donc $t_i = \frac{i}{n}$ $0 \leq i \leq n$.

On calcule alors les polynômes :

P_X pour le support (t_i, x_i) avec $0 \leq i \leq n$

P_Y pour le support (t_i, y_i) avec $0 \leq i \leq n$

Et on trace l'arc paramétré $(P_X(t), P_Y(t))$ pour $t \in [0,1]$.

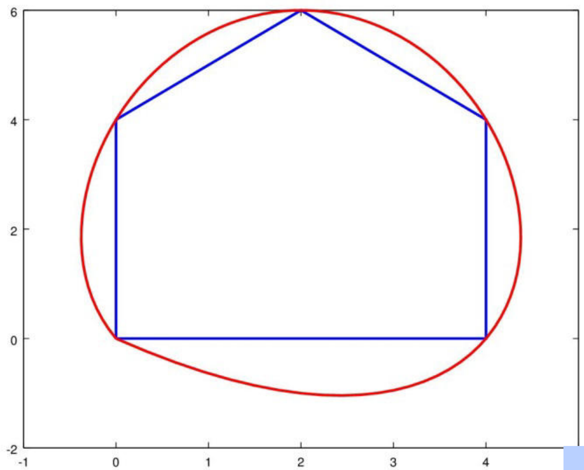
Exemples d’applications :

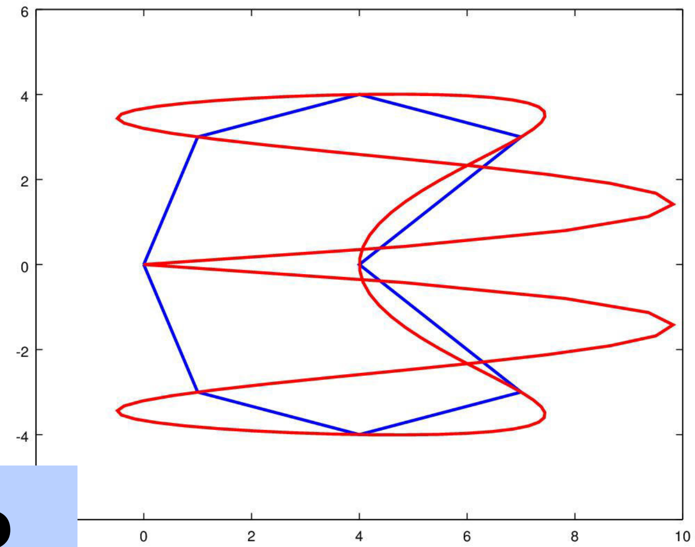
```
# Une maison
XE = np.array([0,4,4,2,0,0])
YE = np.array([0,0,4,6,4,0])
```

```
# Le Pacman
XE = np.array([0,1,4,7,4,7,4,1,0])
YE = np.array([0,-3,-4,-3,0,3,4,3,0])
```

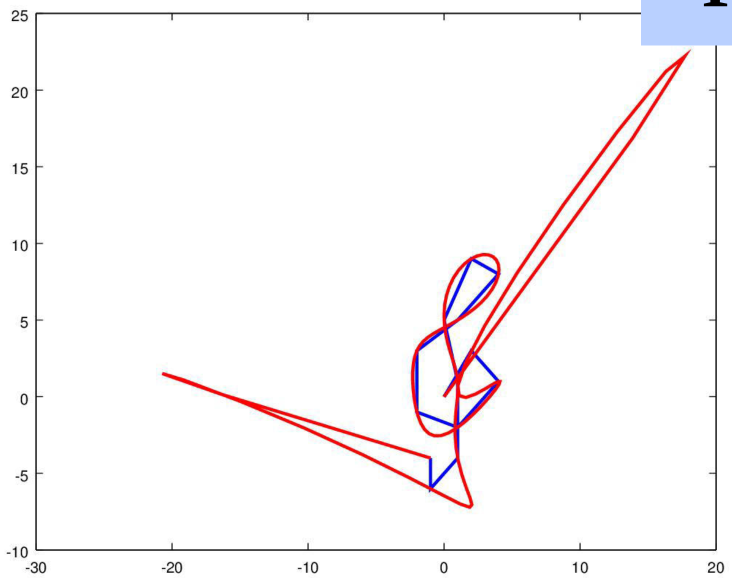
```
# La clef de sol
XE = np.array([0,2,4,1,-2,-2,1,4,2,0,1,1,-1,-1])
YE = np.array([0,3,1,-2,-1,3,5,8,9,5,1,-4,-6,-4])
```

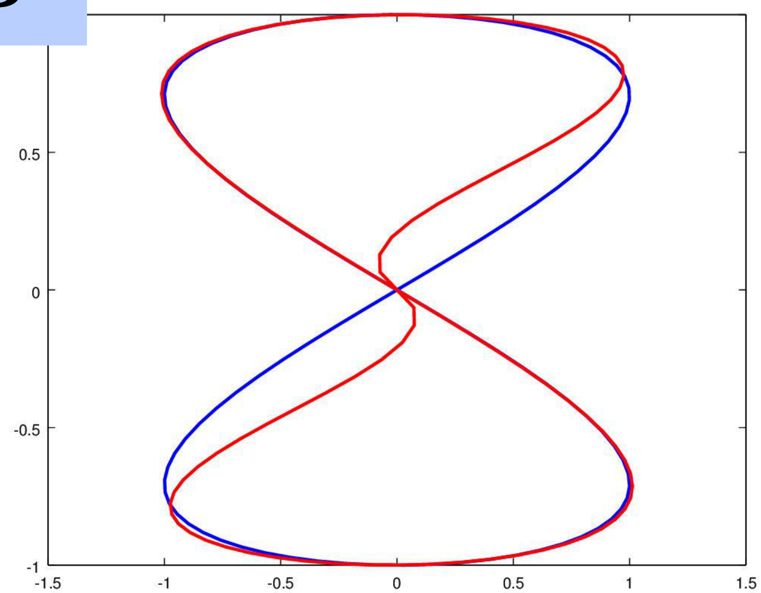
```
# Un huit
TE = np.linspace(-np.pi,np.pi,11)
XE = np.sin(2*TE)
YE = np.sin(TE)
```

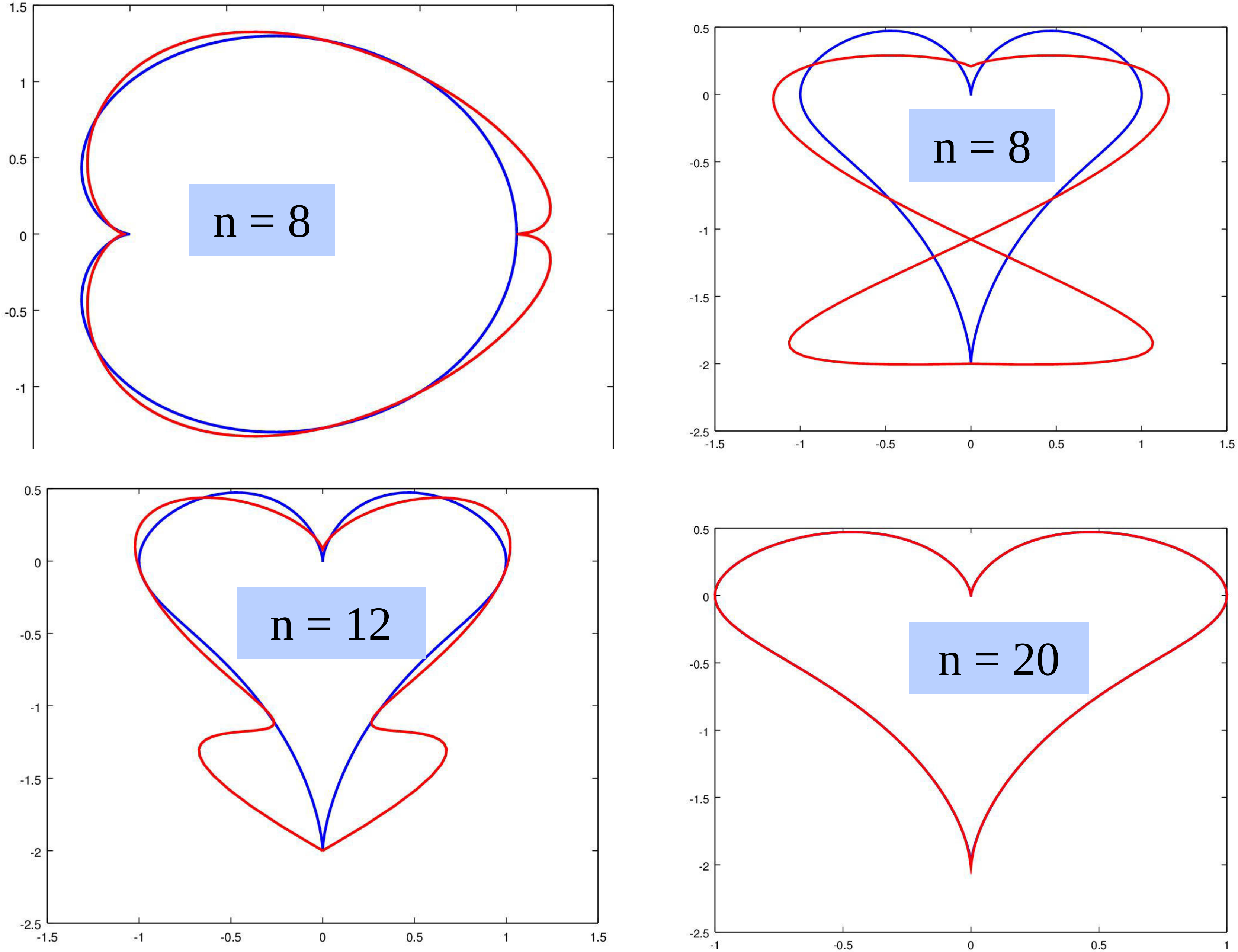

6
4
2
0
-2
-1
0
1
2
3
4


6
4
2
0
-2
-4
0
2
4
6
8
10

n = 8


25
20
15
10
5
0
-5
-10
-30
-20
-10
0
10
20

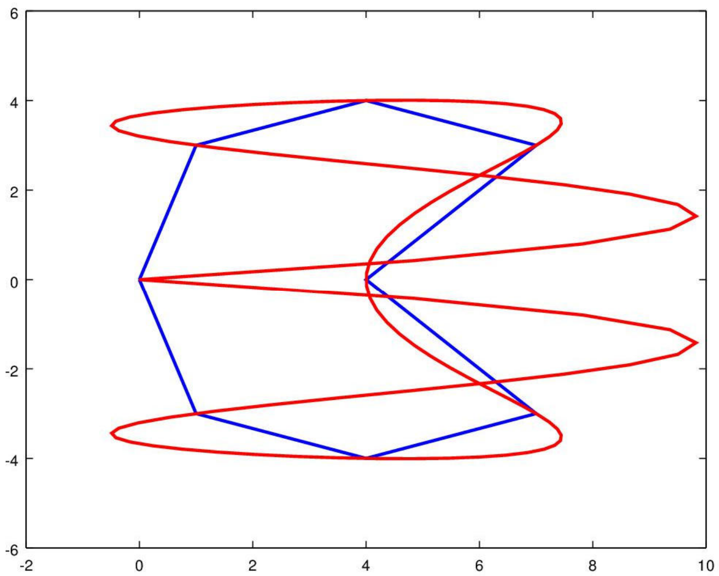

0.5
0
-0.5
-1
-1.5
-1
-0.5
0
0.5
1
1.5

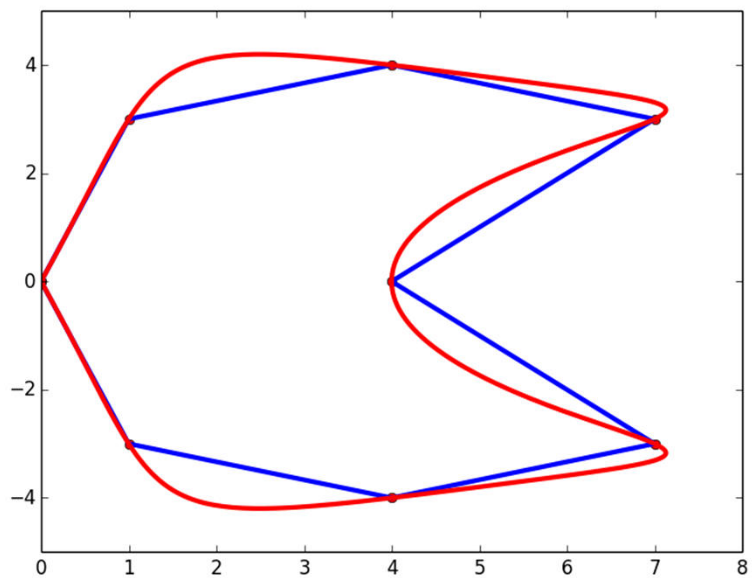

n = 8
n = 8
n = 12
n = 20

Interpolation : cas paramétré

On peut aussi utiliser une subdivision de Tchebychev, au lieu de la subdivision régulière pour le paramétrage, ce qui va améliorer le rendu final.

Avec une subdivision régulière et avec une subdivision de Tchebychev.





Interpolation : cas paramétré

On peut aussi utiliser une subdivision de Tchebychev, au lieu de la subdivision régulière pour le paramétrage, ce qui va améliorer le rendu final.

Avec une subdivision régulière   et   avec une subdivision de Tchebychev.