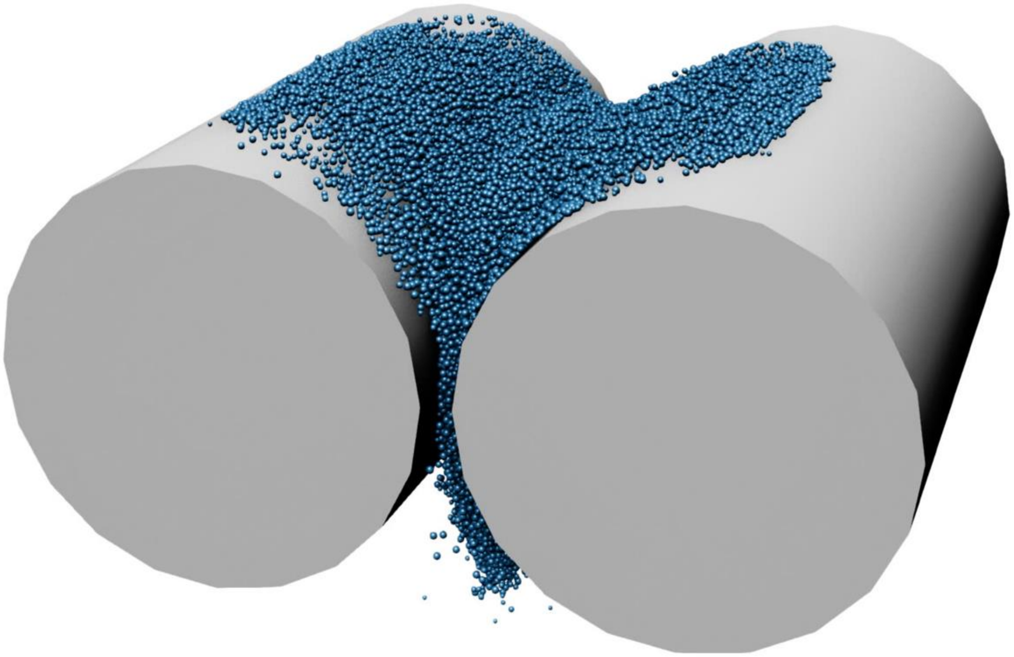
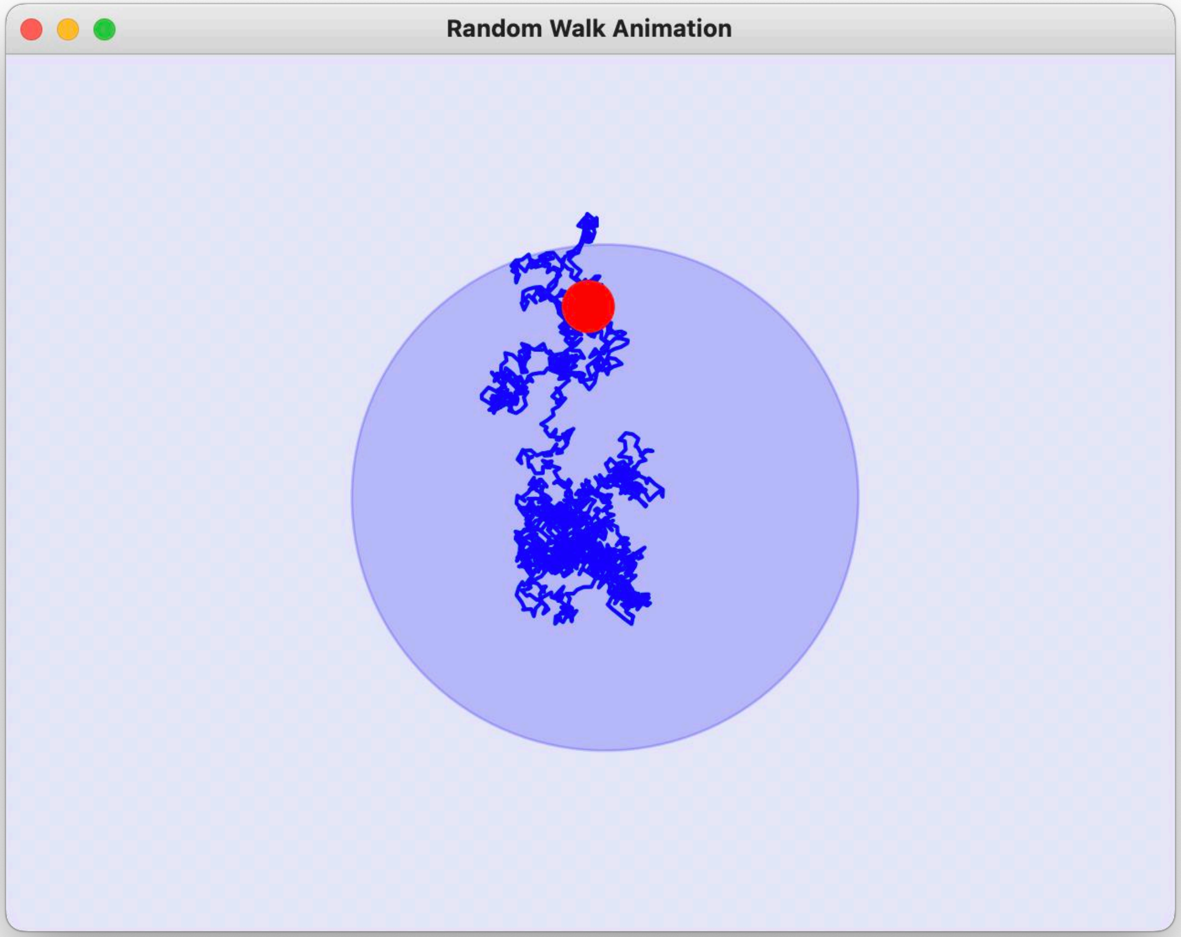
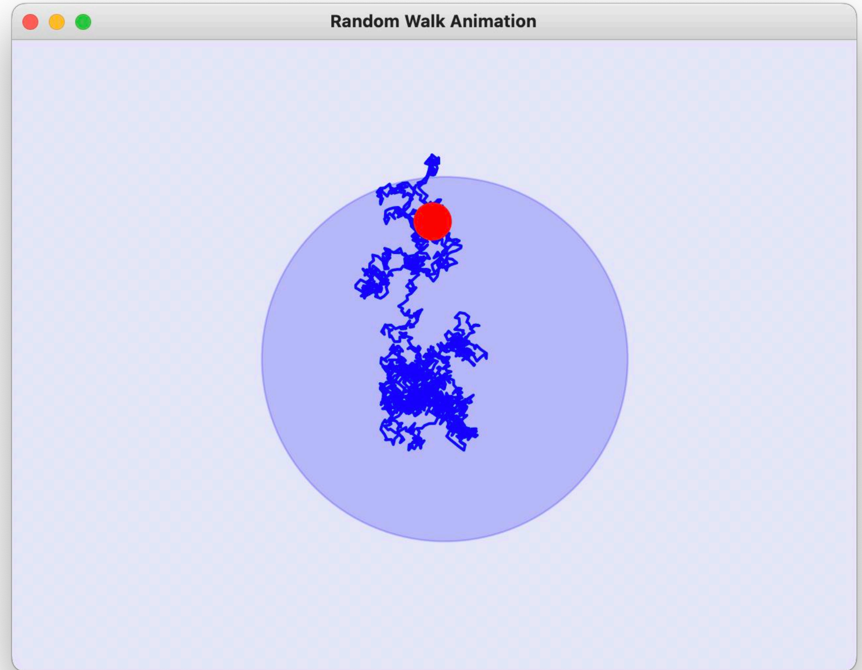
A quoi servent les méthodes numériques ?

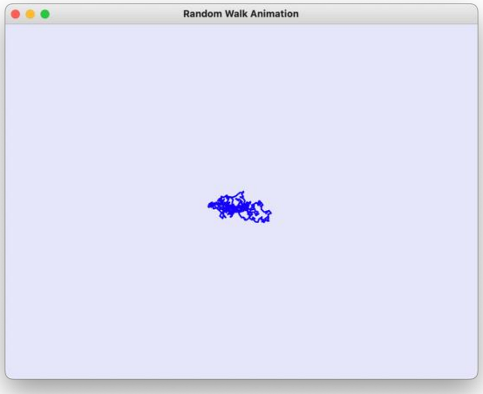


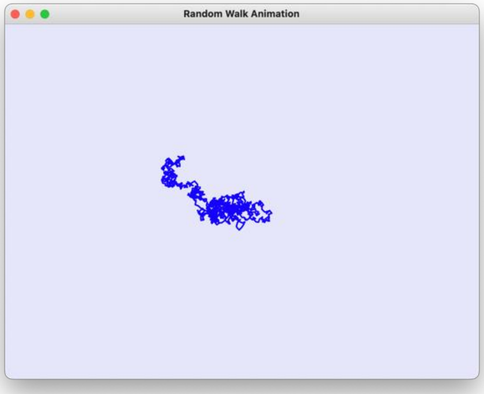


Faire une animation !



Comme Walt Disney !

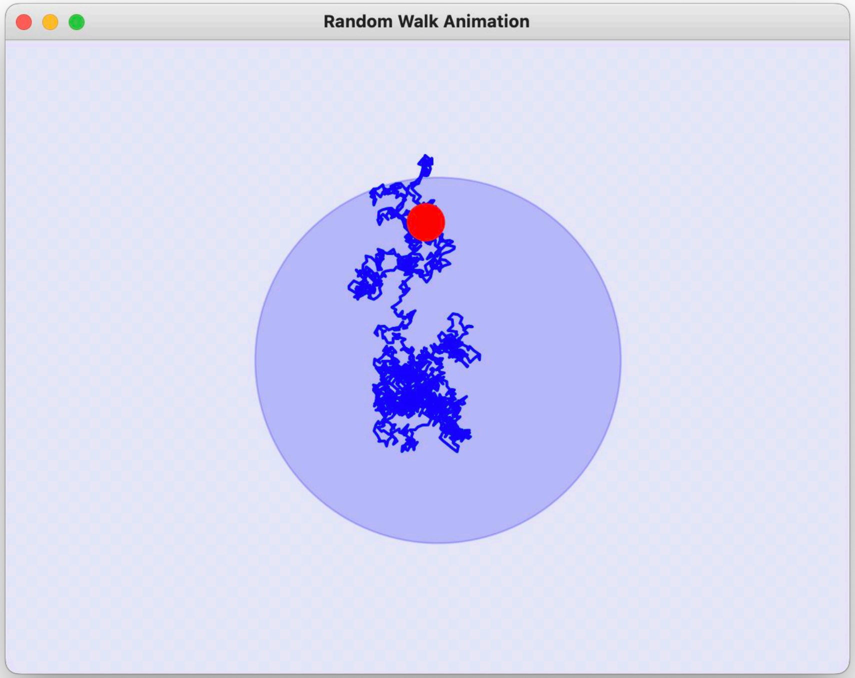




Définir le dessin de chaque frame !

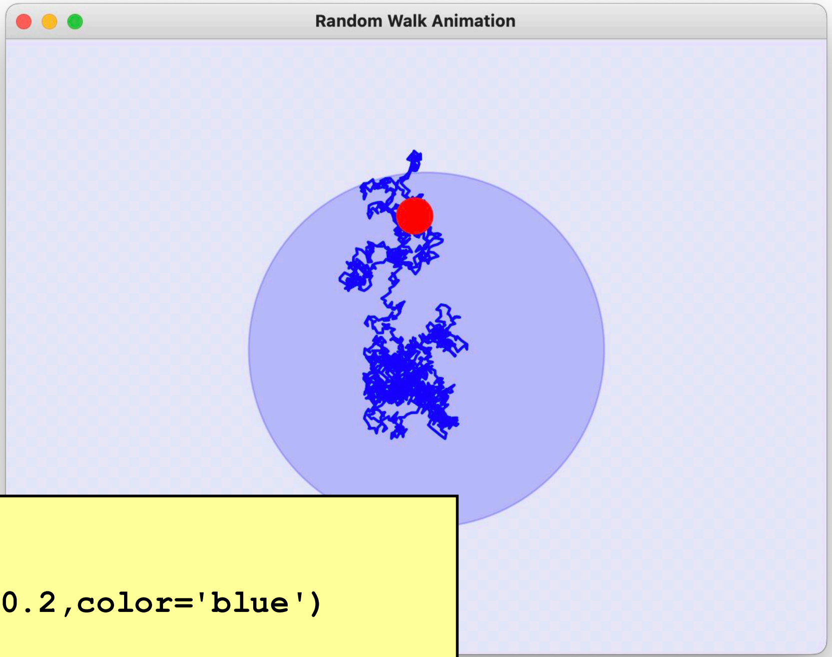
```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as ani

x = np.zeros(2001)*np.nan; x[0] = 0
y = np.zeros(2001)*np.nan; y[0] = 0

def animate(frame):
  x[frame+1] = x[frame] + 0.1*(np.random.rand()-0.5);
  y[frame+1] = y[frame] + 0.1*(np.random.rand()-0.5);
  plt.cla()
  plt.plot(x,y,'-',color='blue')

fig = plt.figure("Random Walk Animation")
ani.FuncAnimation(fig,animate,frames=2000,interval=50,repeat=False)
plt.show()
```

Et le dessin complet…



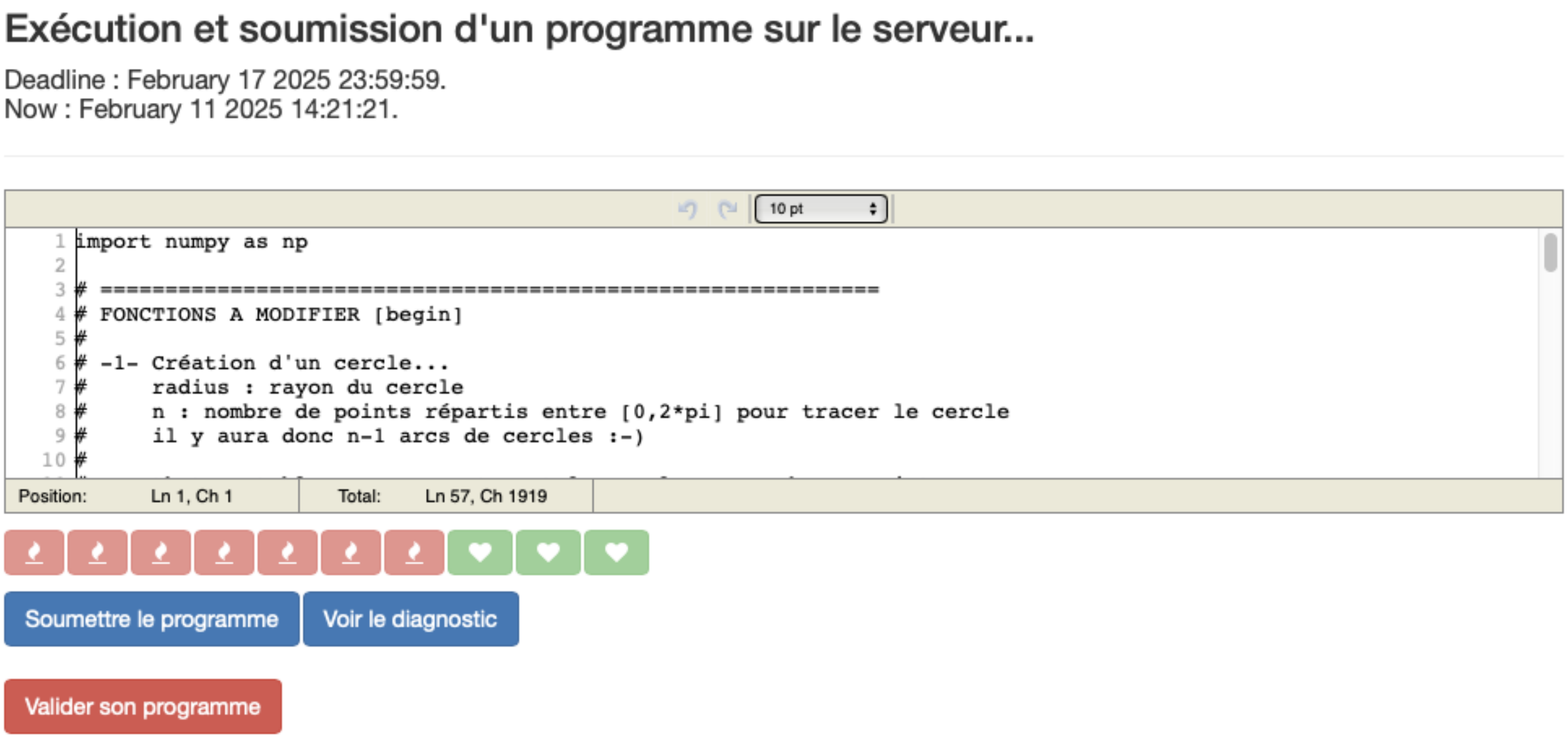

```
def animate(frame):
  xnew = x[frame] + 0.1*(np.random.rand()-0.5); x[frame+1] = xnew
  ynew = y[frame] + 0.1*(np.random.rand()-0.5); y[frame+1] = ynew

  plt.cla()
  ax = plt.gca()
  ax.set_xlim(-2,2); ax.set_ylim(-2,2)
  ax.set_aspect('equal'); plt.axis('off')

  circle = plt.Circle((0,0),1.5,fill=True,alpha=0.2,color='blue')
  plt.plot(x, y,'-',color='blue')
  plt.plot(x[frame+1],y[frame+1],'o',markersize=20, color='red')
  ax.add_artist(circle)
```

Et un truc encore mieux !




```
fig = plt.figure("Random Walk Animation")
ax = plt.gca()
circle = plt.Circle((0,0),1.5,fill=True,alpha=0.2,color='blue')
line1, = plt.plot([],[],'-',color='blue')
line2, = plt.plot([],[],'o',markersize=20,color='red')
ax.add_artist(circle)
x = np.zeros(2001) * np.nan; x[0] = 0
y = np.zeros(2001) * np.nan; y[0] = 0

def animate(frame):
   x[frame+1] = x[frame] + 0.1*(np.random.rand()-0.5)
   y[frame+1] = y[frame] + 0.1*(np.random.rand()-0.5)
   line1.set_data(x,y)
   line2.set_data([x[frame+1]],[y[frame+1]])
   return line1,line2,

ani.FuncAnimation(fig,animate,frames=2000,interval=50,repeat=False)
plt.show()
```

Python
for dummies !

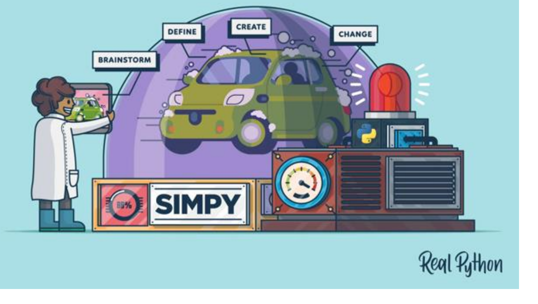


Date limite pour le problème : 17/Fev/2025 23:59:59
et nous sommes aujourd'hui : 12/Fev/2025 10:30:00...

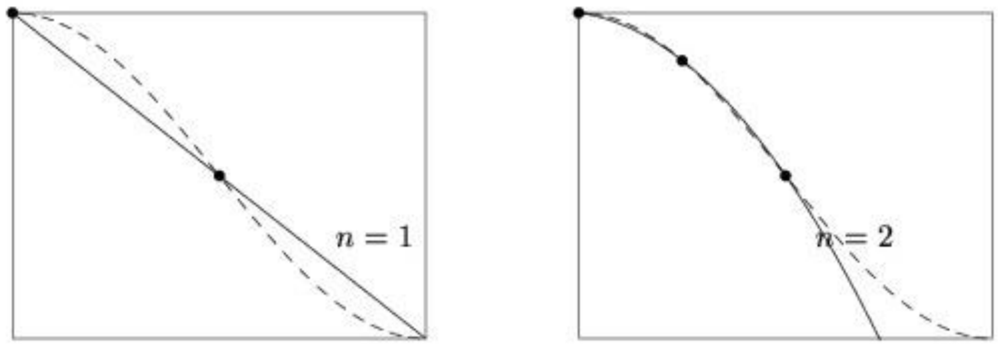
Comment obtenir un développement de Taylor ?

```
def macLaurinCompute(u,x,n,X) :
  ut = 0
  Ut = 0
  dU = np.zeros(n+1)
  for i in range(0,n+1):
    dudx = diff(u,x,i)
    dUdx = dudx.subs(x,0)
    dU[i] = dUdx
    if dUdx != 0:
       term = dUdx*(x**i)/factorial(i)
       Ut += term.subs(x,X)
       ut += term
  return [ut,Ut,dU]
```

```
import numpy as np
from sympy import *

def main() :
  x = symbols('x’); u = cos(x)
  n = 8
  X  = 1.5; U = u.subs(x,X)
  [ut,Ut,dU]  = macLaurinCompute(u,x,n,X)
```

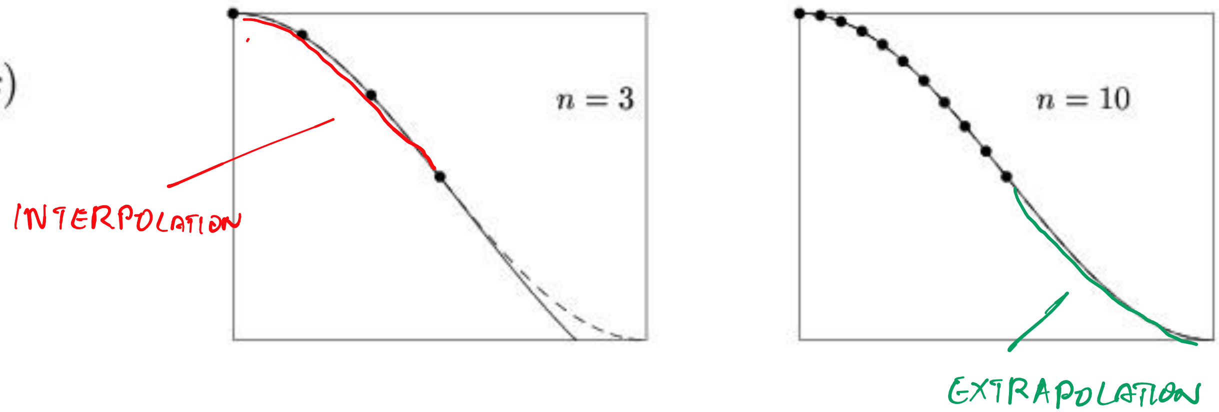


Convergence



Convergence de l'interpolation polynomiale de cos(x)

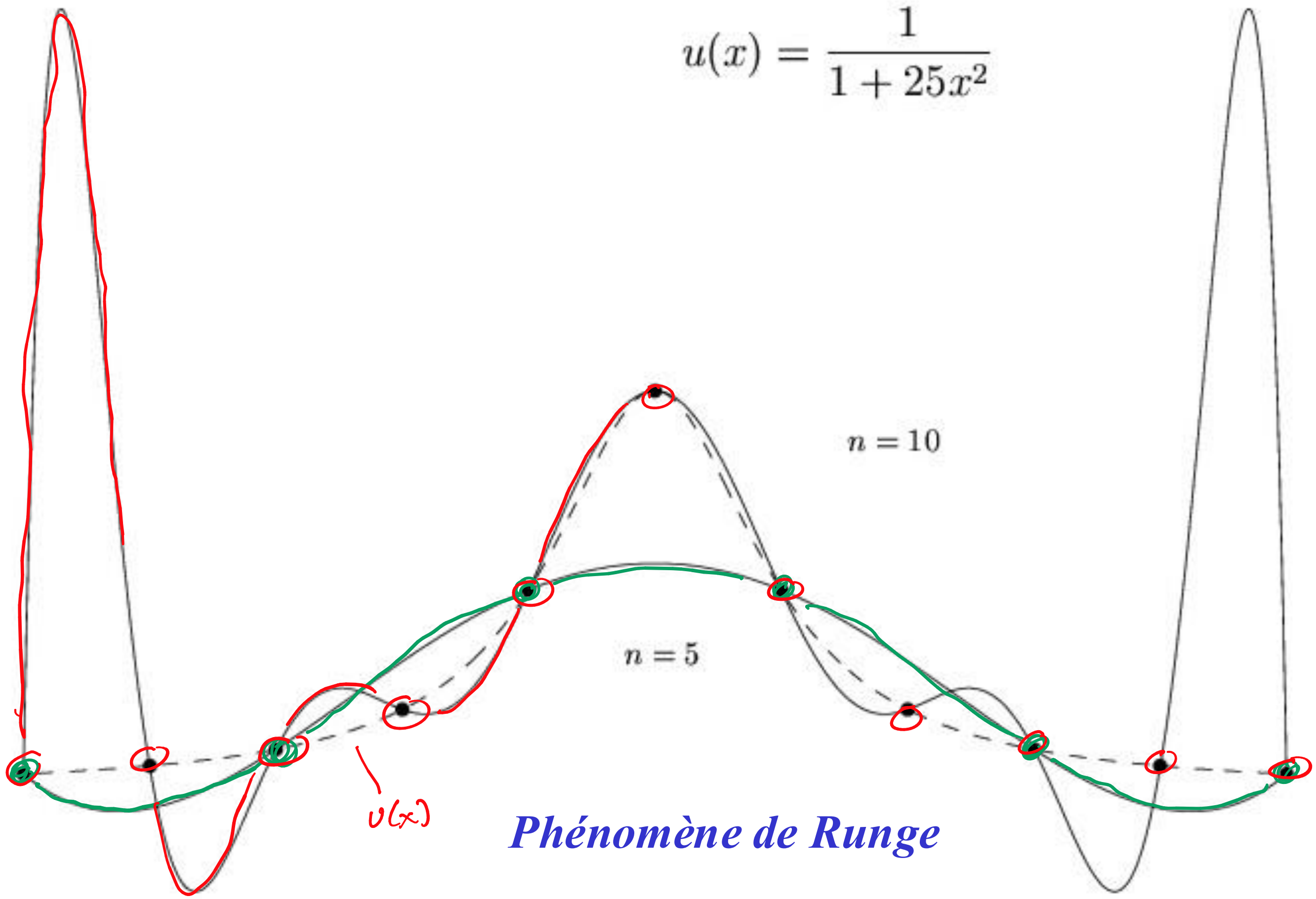
$$e^h(x) = u(x) - u^h(x)$$



Définition 1.3. *Une interpolation est dite* convergente *si l'erreur d'interpolation tend vers zéro lorsque le nombre de degrés de liberté, c'est-à-dire n tend vers l'infini :*

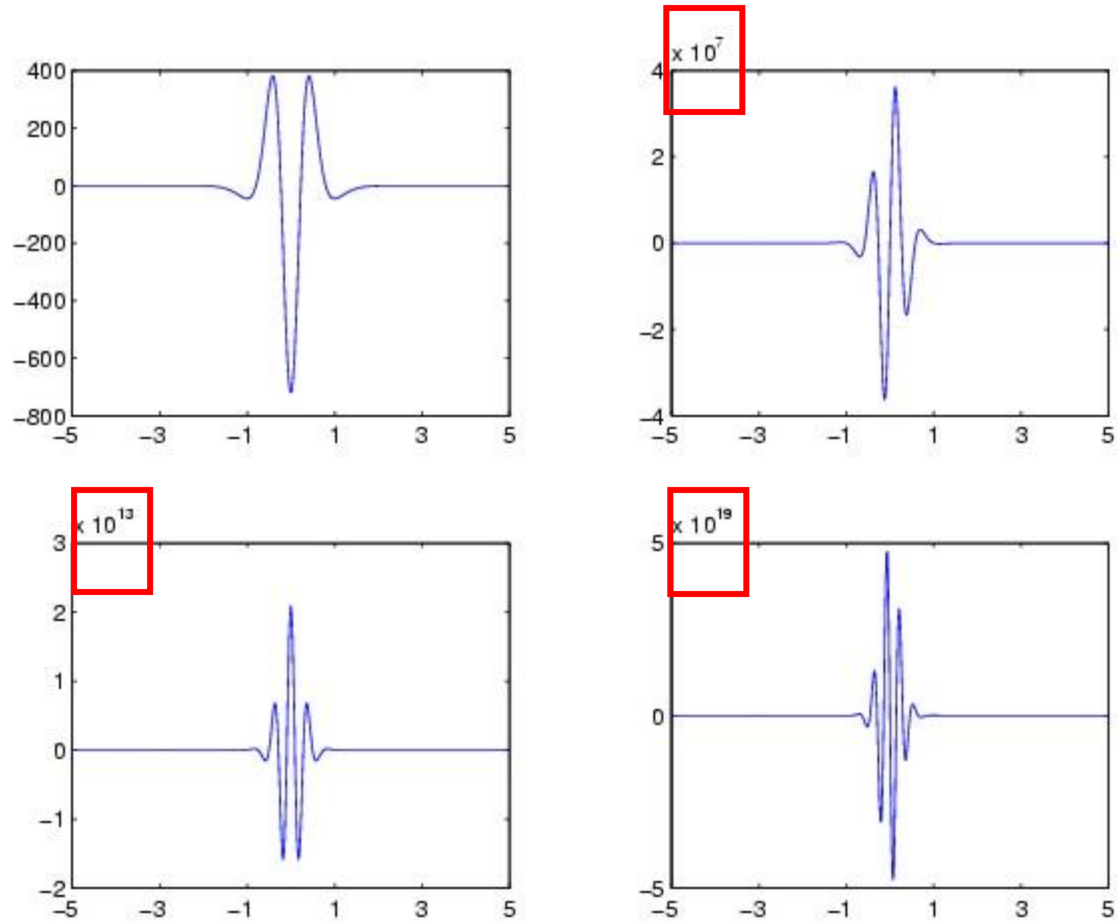
$$\lim_{n\to\infty} e^h(x) = 0 \quad pour\ x \in [X_0, X_n].$$

L'interpolation polynomiale, parfois cela ne converge pas…



Phénomène de Runge

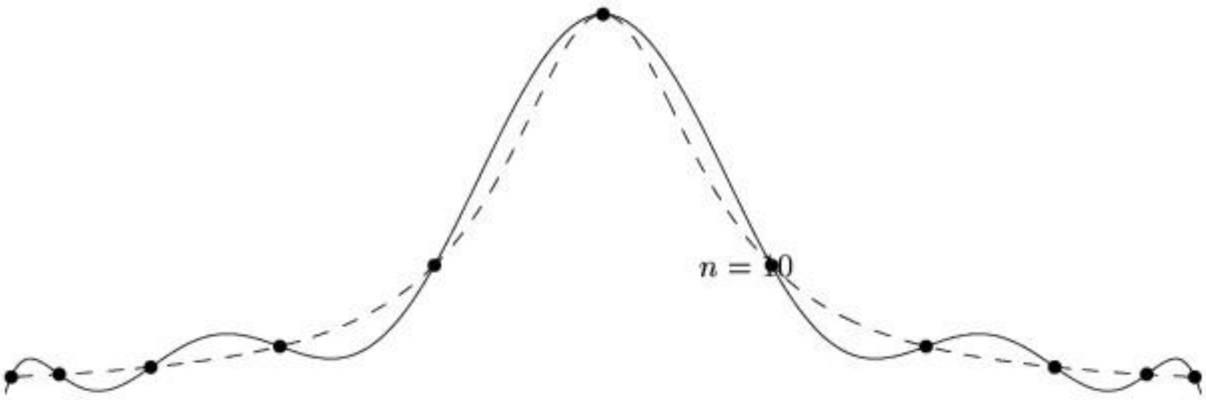
Why does it not work ?



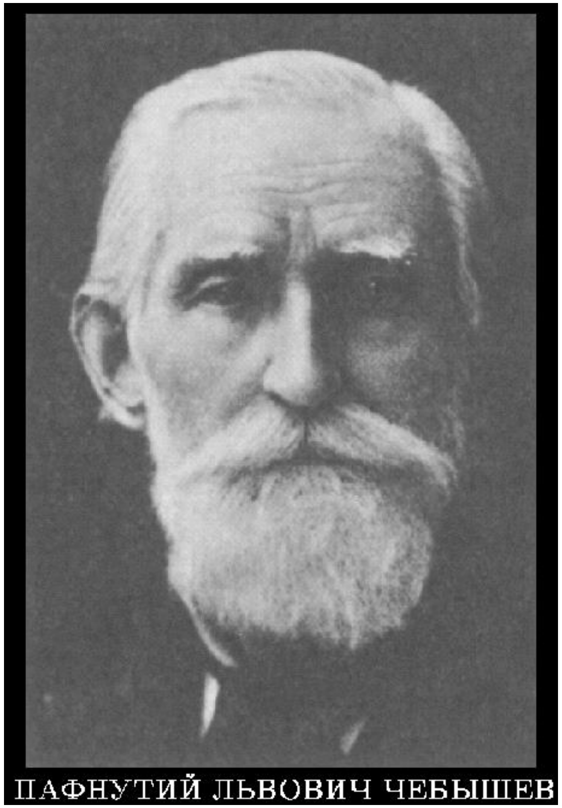
Dérivées d'ordre 6, 11, 16 et 21 de la fonction de Runge

$$e^h(x) = \boxed{\frac{u^{(n+1)}(\xi(x))}{(n+1)!}} (x - X_0)(x - X_1)(x - X_2) \cdots (x - X_n).$$

Parfois, on peut sauver la mise…



Abscisses de Chebyshev

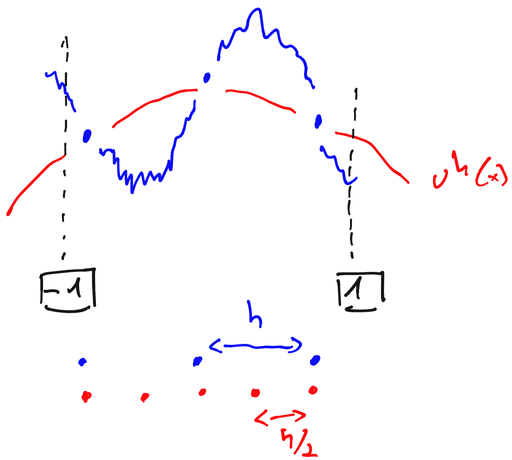


Pafnuty Lvovitch Chebyshev (1821-1894)

Abscisses de Chebychev

$$\underbrace{|u(x)-u^h(x)|}_{e^h(x)} \leq \left|\frac{u^{(n+1)}(\xi(x))}{(n+1)!}\right| \left|(x-X_0)(x-X_1)\dots(x-X_n)\right|$$

$$\left|(x-X_0)(x-X_1)\dots(x-X_n)\right| \leq \frac{n!\,h^{n+1}}{4}$$



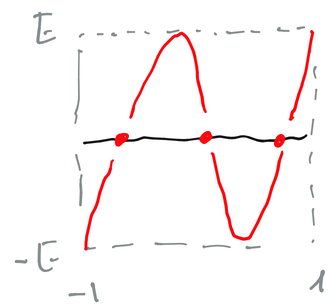
$u^{(n+1)}(\xi(x))$: $n \nearrow$ ↗

$\dfrac{h^{n+1}}{4(n+1)}$: $n \nearrow$, $h \searrow$ ↘

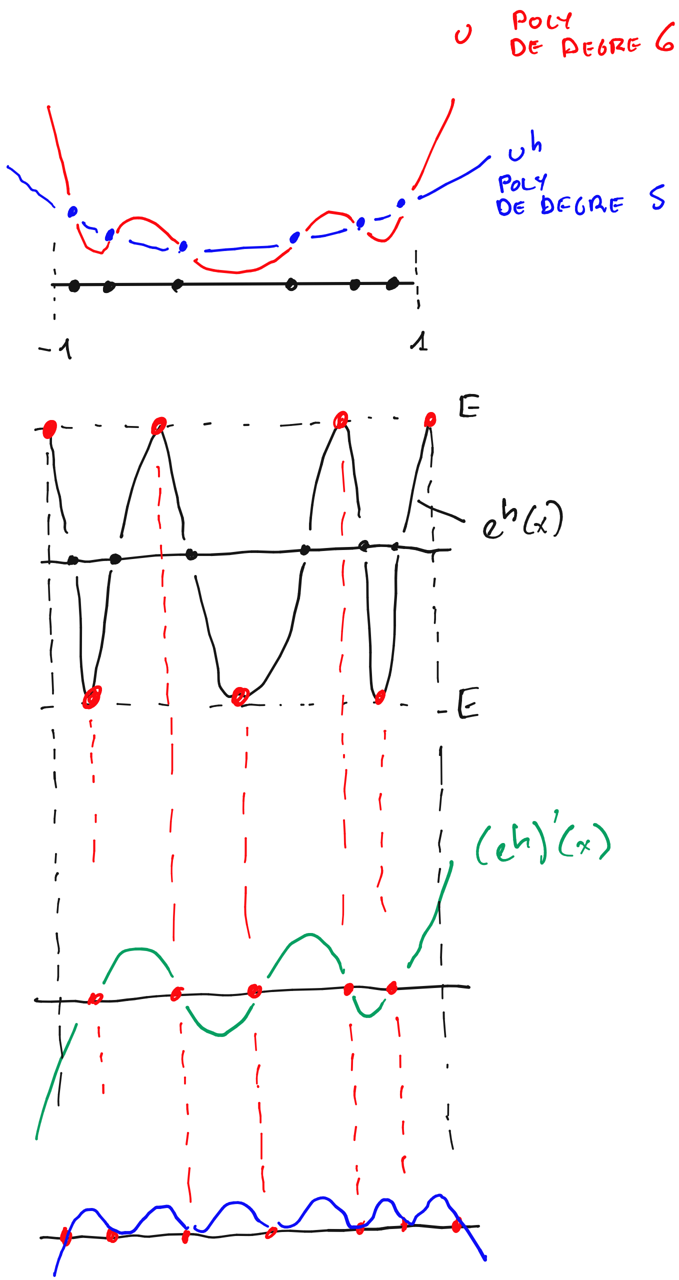
ESTIMATION ASYMPTOTIQUE DE L'ERREUR

$h \to 0$

$$X_i = \cos\left[\frac{(2i+1)}{2(n+1)}\pi\right]$$



DIMINUER CECI LE PLUS EFFICACEMENT POSSIBLE



$$a_6 x^6 + a_5 x^5 + a_4 x^4 \;\cdots\; a_0$$

$$b_5 x^5 \;\cdots\; b_0$$

pu = [a_6 a_5 .... a_0]

puh = [b_5 .... b_0]

peh = pu − [0 * puh]

[a_6 .... a_0] [0 b_5 ... b_0]

pdeh = peh * [6 5 4 3 2 1 0]

pdeh = pdeh[0:6]

$a_6^2 x^{12} \ldots$

$$\left((e^h)'(x)\right)^2 (1-x^2) = (n+1)^2 \left(E^2 - (e^h)^2(x)\right)$$

POLYNOME DE DEGRE 12

POLYNOME DE DEGRE 12

$(6 a_6 x^5 \;\ldots\;)^2 x^2$

$6^2 a_6^2 x^{12}$

2 racines simples

5 racines doubles

$$\left((e^h)'(x)\right)^2(1-x^2) = (n+1)^2\left(E^2-(e^h)^2(x)\right)$$

$$e'\sqrt{1-x^2} = \pm(n+1)\sqrt{E^2-e^2}$$

$$\frac{e'}{E}\frac{1}{\sqrt{1-\left(\frac{e}{E}\right)^2}} = \pm(n+1)\frac{1}{\sqrt{1-x^2}}$$

$$\left(\arccos\left(\frac{e}{E}\right)\right)' = \pm(n+1)\left(\arccos(x)\right)'$$

$$\arccos\left(\frac{e}{E}\right) = \pm(n+1)\arccos(x) + C$$

CAR $e(\pm 1) = E$

$$e^h(x) = E\cos\left[(n+1)\underbrace{\arccos(x)}_{\Theta}\right]$$

$T_{n+1}(x)$

X_i ZEROS DE $T_{n+1}(x)$

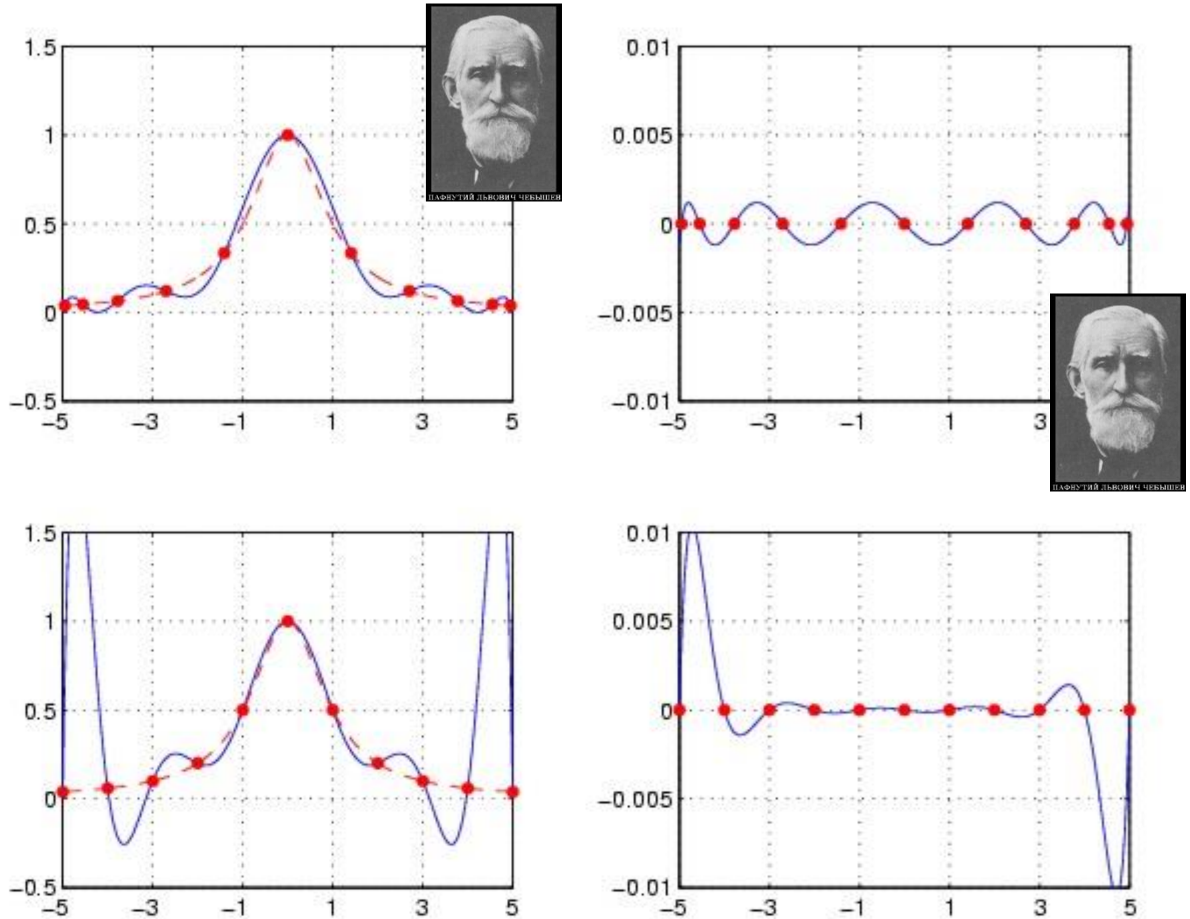
$$0 = \cos\left[(n+1)\arccos(X_i)\right]$$

$$\pi/2 + i\pi = (n+1)\arccos(X_i)$$

$$X_i = \cos\left[\frac{(2i+1)\pi}{2(n+1)}\right]$$

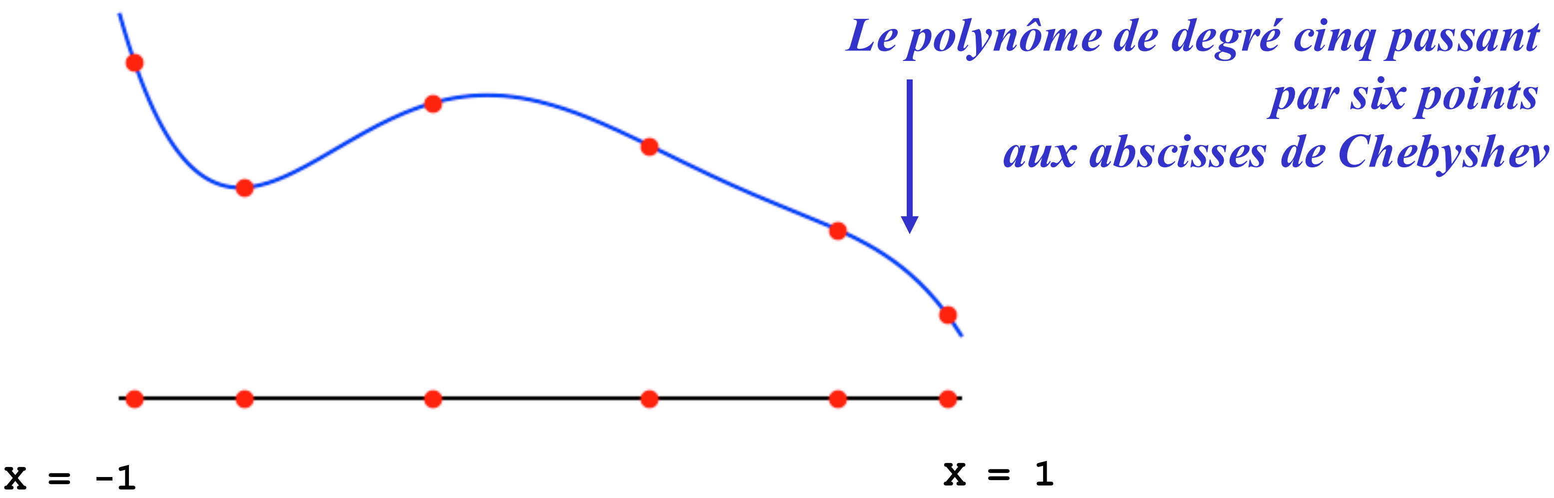
Polynômes de Chebychev

Why does it work ?



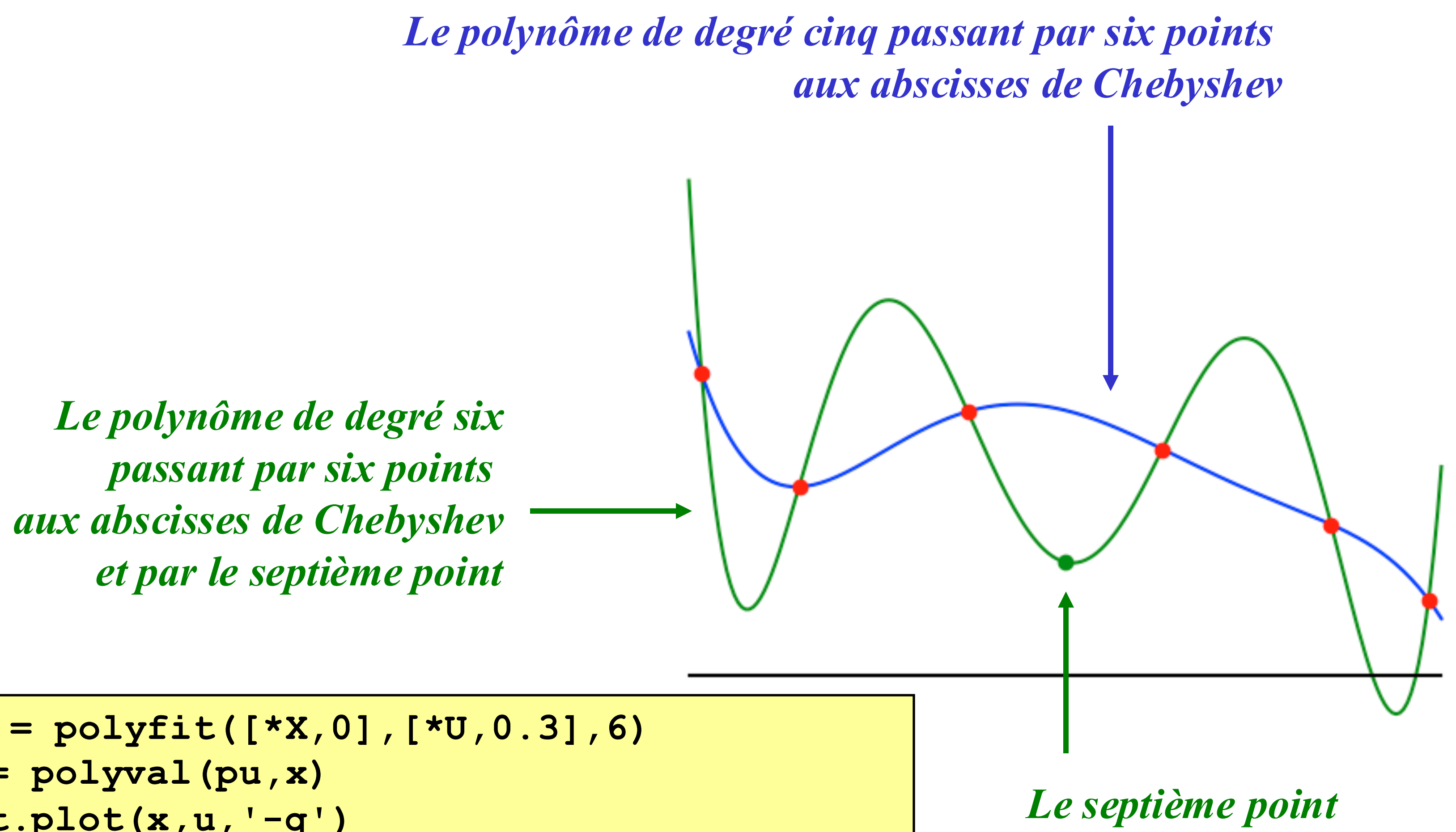
$$e^h(x) = \frac{u^{(n+1)}(\xi(x))}{(n+1)!}(x - X_0)(x - X_1)(x - X_2)\cdots(x - X_n).$$

Six points aux abscisses de Chebyshev….



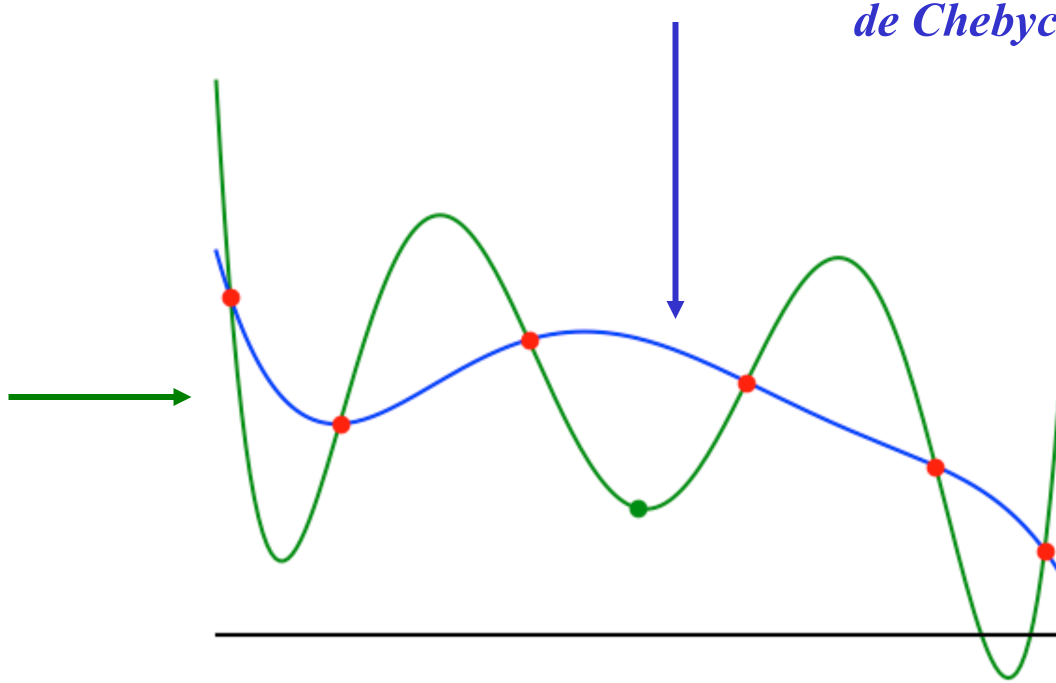

```
n = 5
X = cos(pi * (2*arange(0,n+1) + 1)/((n+1) * 2))
U = [0.2,0.4,0.6,0.7,0.5,0.8]

puh = polyfit(X,U,5)
x = linspace(-1,1,200)
uh = polyval(puh,x)
plt.plot(x,uh,'-b')
plt.plot(X,U,'or')
```

Ajoutons un septième point !

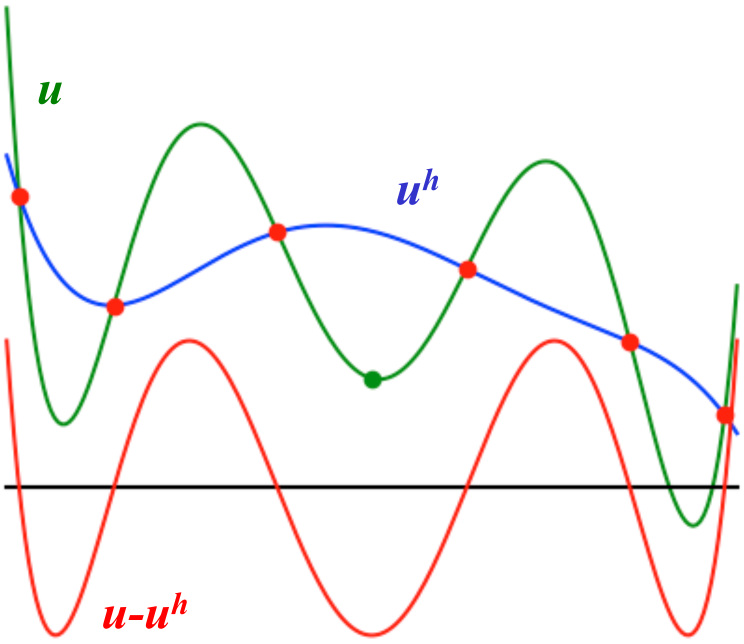



```
pu = polyfit([*X,0],[*U,0.3],6)
u = polyval(pu,x)
plt.plot(x,u,'-g')
plt.plot([0],[0.3],'og')
```

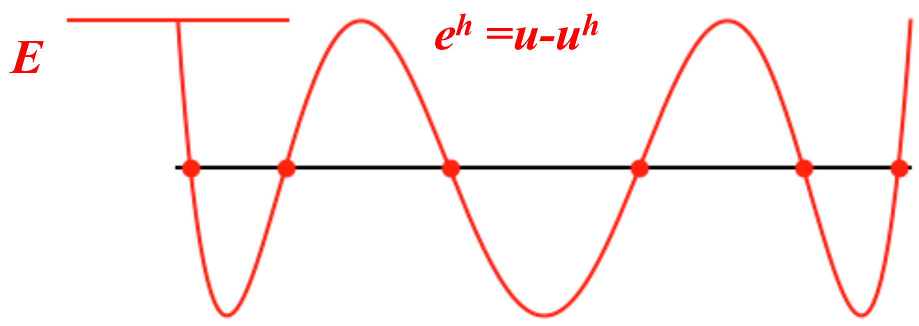
Interpolons un polynôme par un polynôme…



…c'est bête, je sais :-)



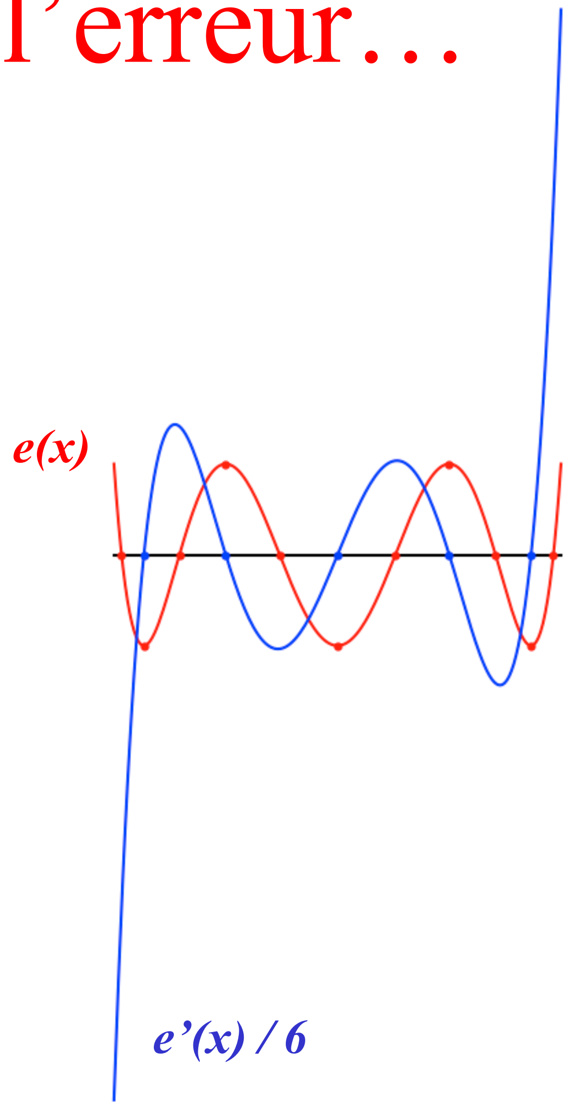

```
error = u - uh
E = error[0]
plt.plot(x,error,'-r');
```



Erreur d'interpolation

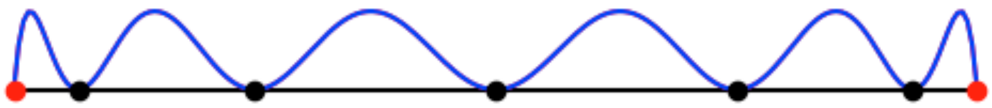
Dérivons et divisons l'erreur…

```
peh = pu - [0,*puh]
eh = polyval(peh, x)
plt.plot(x,eh,'-r')

pdeh = peh * [6,5,4,3,2,1,0]
pdeh = pdeh[0:-1]
deh = polyval(pdeh,x)/6
plt.plot(x,deh,'-b')
```



Et encore quelques petites manipulations…

$$\left(e'(x)\right)^2 \left(1 - x^2\right) \quad = \quad (n+1)^2 \left(E^2 - e^2(x)\right)$$



```
Xd = roots(pdeh)
Ed = polyval(peh,Xd)
E = Ed[0]

plt.plot(x,E**2-eh**2,'-r')
plt.plot(x,(1-x**2)*(deh**2),'-b')
plt.plot(Xd,zeros(size(Xd)),'ok')
plt.plot([-1,1],[0,0],'or')
```

Une solution analytique d'une équation différentielle ?

$$(e'(x))^2 (1 - x^2) = (n+1)^2 (E^2 - e^2(x))$$

$$\downarrow$$

$$\frac{\frac{e'(x)}{E}}{\sqrt{1 - \left(\frac{e}{E}\right)^2}} = \pm(n+1)\frac{1}{\sqrt{1-x^2}}$$

```
>>> from sympy import *
>>> x = symbols('x')
>>> f = 1/sqrt(1-x**2)
>>> integrate(f)
asin(x)
```

Et si on dérive la primitive…

```
>>> from sympy import *
>>> x = symbols('x')
>>> f = 1/sqrt(1-x**2)
>>> integrate(f)
asin(x)
>>> diff(asin(x))
1/sqrt(-x**2 + 1)
>>> diff(acos(x))
-1/sqrt(-x**2 + 1)
```

Une petite solution analytique comme le faisaient les anciens…

$$\left(e'(x)\right)^2 \left(1-x^2\right) = (n+1)^2 \left(E^2 - e^2(x)\right)$$

$$\downarrow$$

$$\frac{\frac{e'(x)}{E}}{\sqrt{1-\left(\frac{e}{E}\right)^2}} = \pm(n+1)\frac{1}{\sqrt{1-x^2}}$$

$$\downarrow$$

$$\arccos\left(\frac{e(x)}{E}\right) = \pm\left((n+1)\arccos(x) + C\right)$$

$$\downarrow$$

En vertu de la parité du cosinus !

$$e(x) = E\cos\left((n+1)\arccos(x) + C\right)$$

$$\downarrow$$

En imposant que e(1) = E

$$e(x) = E\underbrace{\cos\left((n+1)\arccos(x)\right)}_{T_{n+1}(x)}$$

Polynôme de Chebyshev de degré n+1

Drôle d'expression pour un polynôme, non ?

Théorème 1.2. *Les polynômes de Chebyshev* $T_{n+1}(x) = \cos((n+1)\arccos(x))$ *définis sur l'intervalle* $[-1,1]$ *satisfont la relation de récurrence*

$$T_{i+1}(x) = 2x\, T_i(x) - T_{i-1}(x), \quad i = 1, 2, 3, \ldots,$$

avec $T_0(x) = 1$ *et* $T_1(x) = x$.

Calcul des polynômes de Chebyshev : formule de récurrence

Démonstration : Définissons $\theta = \arccos(x)$ et écrivons :

$$\begin{aligned} T_{i+1}(x) &= \cos((i+1)\theta) \\ &= \cos(\theta)\cos(i\theta) - \sin(\theta)\sin(i\theta) \end{aligned}$$

$$\begin{aligned} T_{i-1}(x) &= \cos((i-1)\theta) \\ &= \cos(\theta)\cos(i\theta) + \sin(\theta)\sin(i\theta) \end{aligned}$$

$$\begin{aligned} T_{i+1}(x) + T_{i-1}(x) &= 2\cos(\theta)\cos(i\theta) \\ &= 2xT_i(x) \end{aligned}$$

□

Abscisses de Chebyshev

$$0 = \overbrace{\cos\left((n+1)\arccos(X_i)\right)}^{T_{n+1}(X_i)} \qquad i = 0, \ldots, n$$

$$\downarrow$$

$$\frac{\pi/2 + i\pi}{(n+1)} = \arccos(X_i) \qquad i = 0, \ldots, n$$

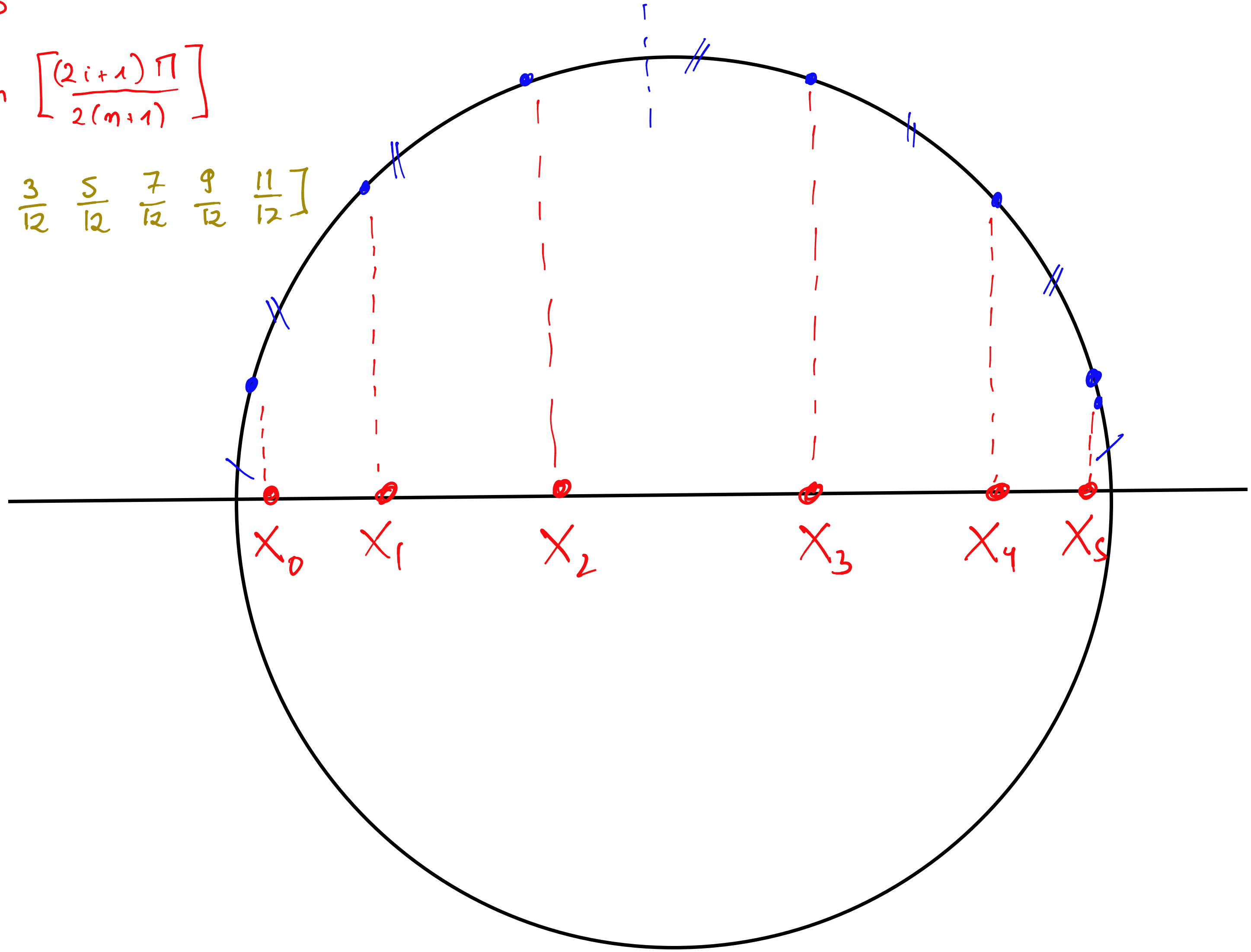
$$\downarrow$$

$$\cos\left(\frac{(2i+1)\pi}{2(n+1)}\right) = X_i \qquad i = 0, \ldots, n$$

$n = 5$

$$X_i = \cos\left[\frac{(2i+1)\pi}{2(n+1)}\right]$$

$$\cos\left[\pi\left(\frac{1}{12}\ \frac{3}{12}\ \frac{5}{12}\ \frac{7}{12}\ \frac{9}{12}\ \frac{11}{12}\right)\right]$$



Abscisses
de Chebychev

Interpolation polynomiale : bilan

- **Pour une fonction u(x) très régulière : fonction cosinus**

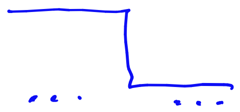
  Convergence de l'interpolation polynomiale

- **Pour une fonction u(x) suffisamment régulière : fonction de Runge**

  Divergence pour des abscisses équidistantes
  Convergence pour les abscisses de Chebyshev

- **Pour une fonction u(x) peu régulière : fonction échelon**

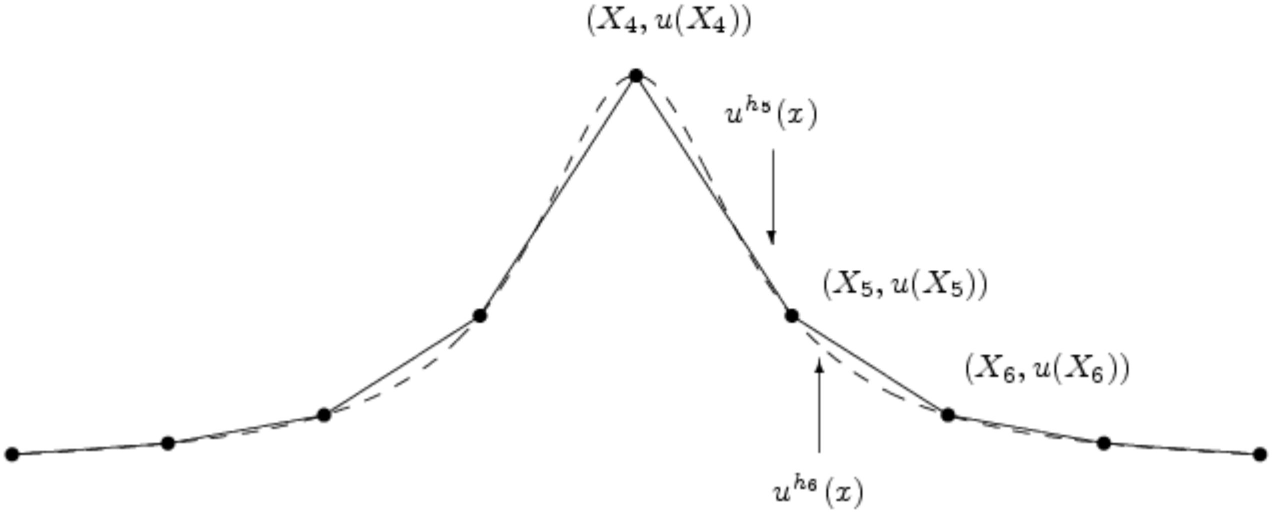
  Divergence !
  Eviter l'interpolation polynomiale de degré élevé



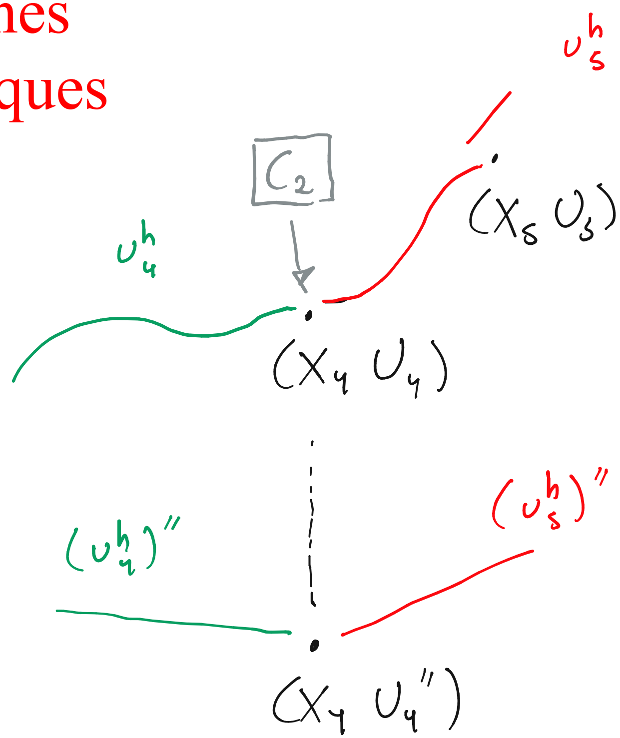
Idée :

Utiliser une interpolation par morceaux composée des polynômes de degré bas !

Interpolation linéaire par morceaux



Splines cubiques



BE CAREFUL

$$U_4'' \neq u''(X_4)$$

$$u_5^h(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0$$

n+1 POINTS
n FONCTIONS
4n COEFFICIENTS

$$u_4^h(X_4) = U_4 = u_5^h(X_4)$$

$$(u_4^h)'(X_4) = (u_5^h)'(X_4) \neq (u)'(X_4)$$

$$(u_4^h)''(X_4) = (u_5^h)''(X_4) \neq (u)''(X_4)$$

$$2(n-1) + 2 + 2(n-1)$$

IL Y A
4n - 2 CONDITIONS

IL MANQUE 2 CONDITIONS

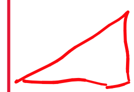
$$(u^h)''(X_0) = 0$$

$$(u^h)''(X_n) = 0$$

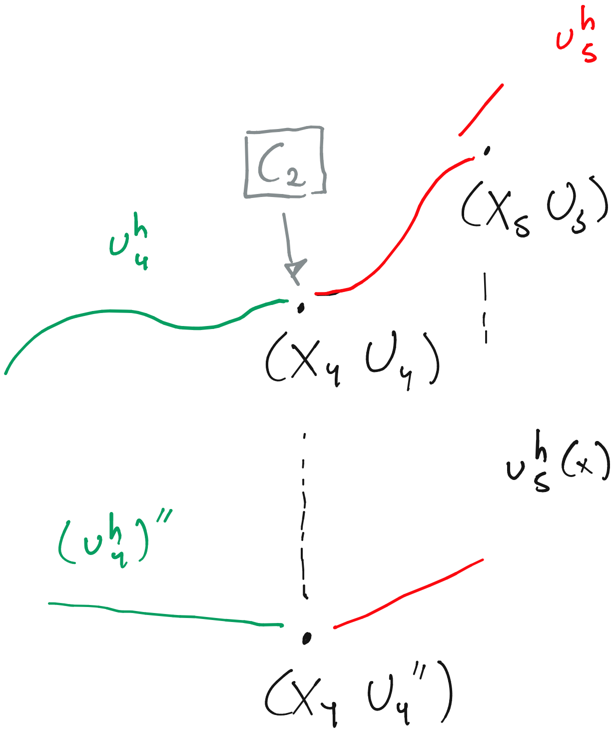
Etape 1
Dérivée seconde ok

$$(u_5^h)''(x) = U_4'' \frac{(x - X_5)}{(X_4 - X_5)} + U_5'' \frac{(x - X_4)}{(X_5 - X_4)}$$

$\frac{X_5 - x}{h}$



$\frac{x - X_4}{h}$



$$u_5^h(x) = U_4'' \frac{(X_5 - x)^3}{6h} + U_5'' \frac{(x - X_4)^3}{6h} + A\frac{(X_5 - x)}{h} + B\frac{(x - X_4)}{h}$$

$Cx + D$

$$U_4 = U_4'' \frac{h^3}{6h} + A$$

$$A = U_4 - U_4'' \frac{h^2}{6}$$

$$B = U_5 - U_5'' \frac{h^2}{6}$$

Etape 2
Interpolation ok

Etape 3
Dérivée première ok

$$u_5^h(x) = U_4'' \frac{(X_5 - x)^3}{6h} + U_5'' \frac{(x - X_4)^3}{6h} + A\frac{(X_5 - x)}{h} + B\frac{(x - X_4)}{h}$$

$$(u_4^h)'(X_4) = (u_5^h)'(X_4)$$

$$= U_4'' \frac{-3h^2}{6h} - \frac{A}{h} + \frac{B}{h}$$

$$\left[\frac{(X_5 - x)^3}{6h}\right]'_{X_4} = -3\frac{(X_5 - x)^2}{6h}\Big|_{X_4}$$

$$\frac{hU_4''}{2} + \left[\frac{U_4 - U_3}{h}\right] - \left[\frac{U_4'' - U_3''}{h}\right]\frac{h^2}{6} = -\frac{hU_4''}{2} + \left[\frac{U_5 - U_4}{h}\right] - \left[\frac{U_5'' - U_4''}{h}\right]\frac{h^2}{6}$$

$$\left[\frac{U_3 - 2U_4 + U_5}{h}\right] = \frac{h}{6}\left[U_3'' + 4U_4'' + U_5''\right]$$

$$\boxed{A = U_4 - U_4''\frac{h^2}{6}}$$

$$\boxed{B = U_5 - U_5''\frac{h^2}{6}}$$

Et le système
à résoudre est finalement…

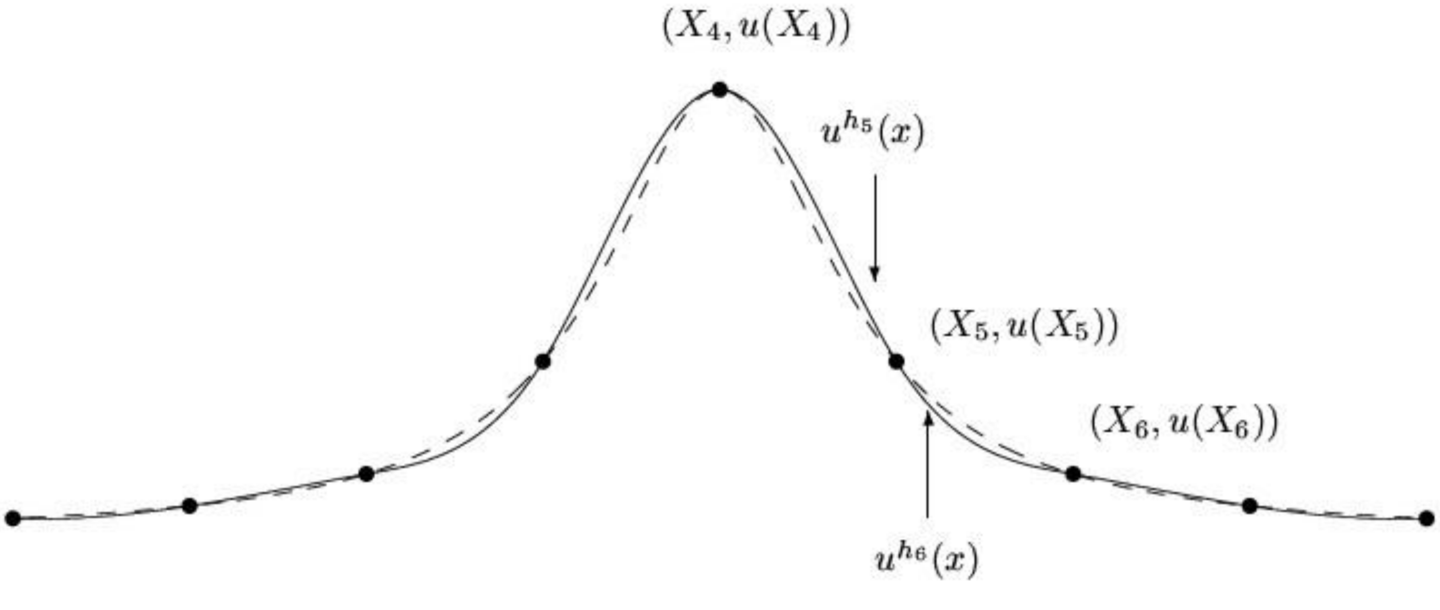
$$\left[\frac{U_3 - 2U_4 + U_5}{h}\right] = \frac{h}{6}\left[U_3'' + 4U_4'' + U_5''\right]$$

$$\frac{h^2}{6}\begin{bmatrix} 1 & & & & & \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & & \ddots & & \\ & & & 1 & 4 & 1 \\ & & & & & 1 \end{bmatrix}\begin{bmatrix} U_0'' \\ \vdots \\ U_4'' \\ \vdots \\ U_n'' \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ U_3 - 2U_4 + U_5 \\ \vdots \\ 0 \end{bmatrix}$$

Interpolation par splines cubiques

$$u^{h_i}(x) = a_i + b_i x + c_i x^2 + d_i x^3 \qquad i = 1, 2, \ldots, n$$

4n coefficients inconnus



Comment trouver les coefficients ?

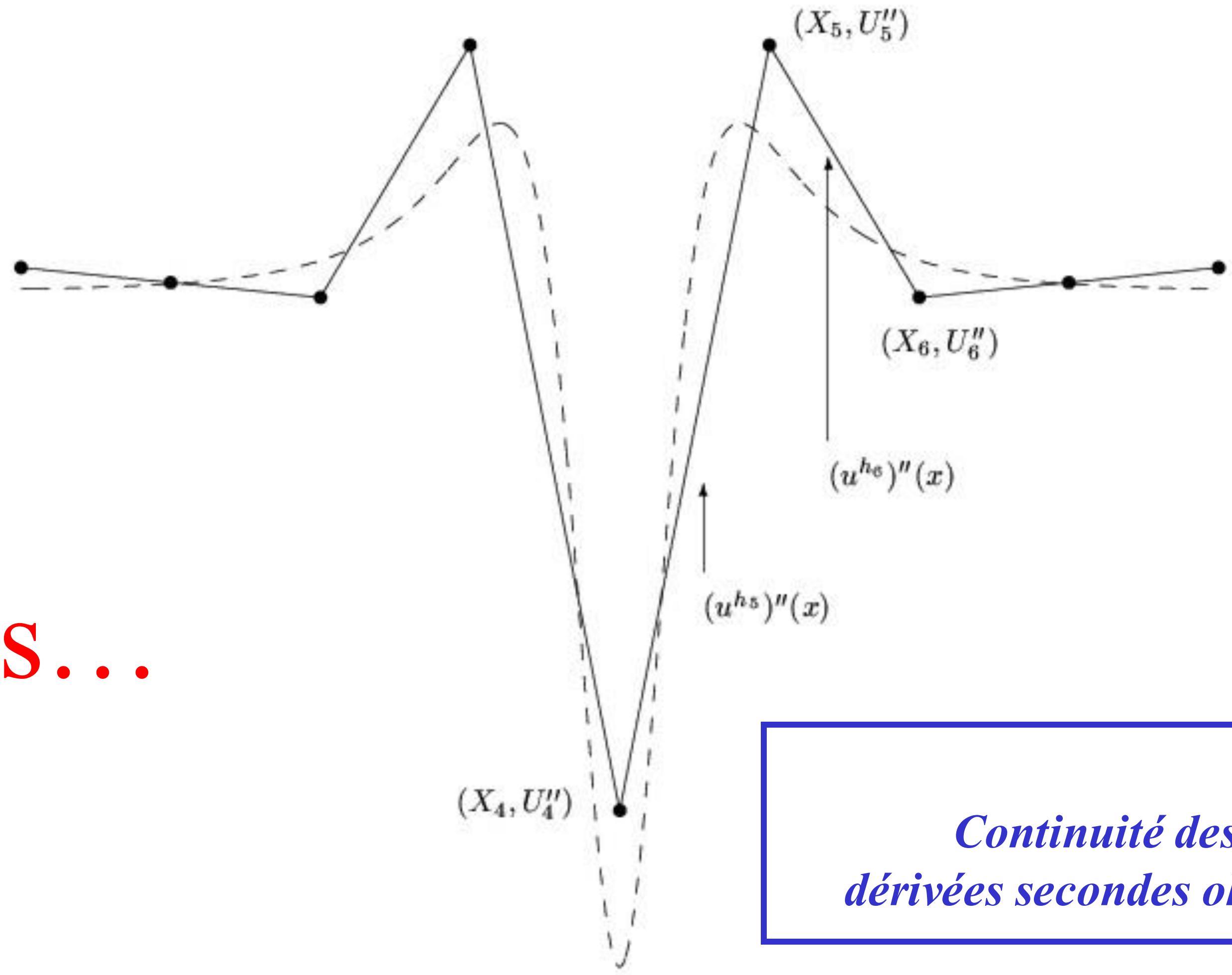
$$\begin{aligned} u^{h_1}(X_0) &= U_0 \\ u^{h_n}(X_n) &= U_n \end{aligned}$$

$$\begin{aligned} u^{h_i}(X_i) &= U_i & i = 1, \ldots, n-1 \\ u^{h_{i+1}}(X_i) &= U_i & i = 1, \ldots, n-1 \end{aligned}$$

$$\begin{aligned} (u^{h_i})'(X_i) &= (u^{h_{i+1}})'(X_i) & i = 1, \ldots, n-1 \\ (u^{h_i})''(X_i) &= (u^{h_{i+1}})''(X_i) & i = 1, \ldots, n-1 \end{aligned}$$

4n-2 conditions

$$(u^{h_i})'' = U''_{i-1}\frac{(x - X_i)}{(X_{i-1} - X_i)} + U''_i\frac{(x - X_{i-1})}{(X_i - X_{i-1})}$$



Ecrivons…

Continuité des dérivées secondes ok

Intégrons…

$$(u^{h_i})'' = U''_{i-1}\frac{(X_i - x)}{h_i} + U''_i\frac{(x - X_{i-1})}{h_i}$$

En intégrant deux fois,

$$(u^{h_i}) = U''_{i-1}\frac{(X_i - x)^3}{6h_i} + U''_i\frac{(x - X_{i-1})^3}{6h_i} + A_i\frac{(X_i - x)}{h_i} + B_i\frac{(x - X_{i-1})}{h_i}$$

$$U_{i-1} = U''_{i-1}\frac{h_i^3}{6h_i} + A_i\frac{h_i}{h_i} \quad \text{et} \quad U_i = U''_i\frac{h_i^3}{6h_i} + B_i\frac{h_i}{h_i}$$

$$A_i = U_{i-1} - \frac{U''_{i-1}\,h_i^2}{6} \qquad B_i = U_i - \frac{U''_i\,h_i^2}{6}$$

Continuité de la fonction ok

Calculons…

Continuité des dérivées premières ok

$$(u^{h_i})'(X_i) = (u^{h_{i+1}})'(X_i)$$

$$\frac{U_i''h_i}{2} + \frac{(U_i - U_{i-1})}{h_i} - \frac{(U_i'' - U_{i-1}'')h_i}{6} = -\frac{U_i''h_{i+1}}{2} + \frac{(U_{i+1} - U_i)}{h_{i+1}} - \frac{(U_{i+1}'' - U_i'')h_{i+1}}{6}$$

$$\frac{(2U_i'' + U_{i-1}'')h_i}{6} + \frac{(U_i - U_{i-1})}{h_i} = \frac{(U_{i+1} - U_i)}{h_{i+1}} - \frac{(U_{i+1}'' + 2U_i'')h_{i+1}}{6}$$

$$\frac{h_i}{6}\,U_{i-1}'' + \frac{2(h_i + h_{i+1})}{6}\,U_i'' + \frac{h_{i+1}}{6}\,U_{i+1}'' = \frac{(U_{i+1} - U_i)}{h_{i+1}} - \frac{(U_i - U_{i-1})}{h_i}$$

$$i = 1, \ldots, n-1$$

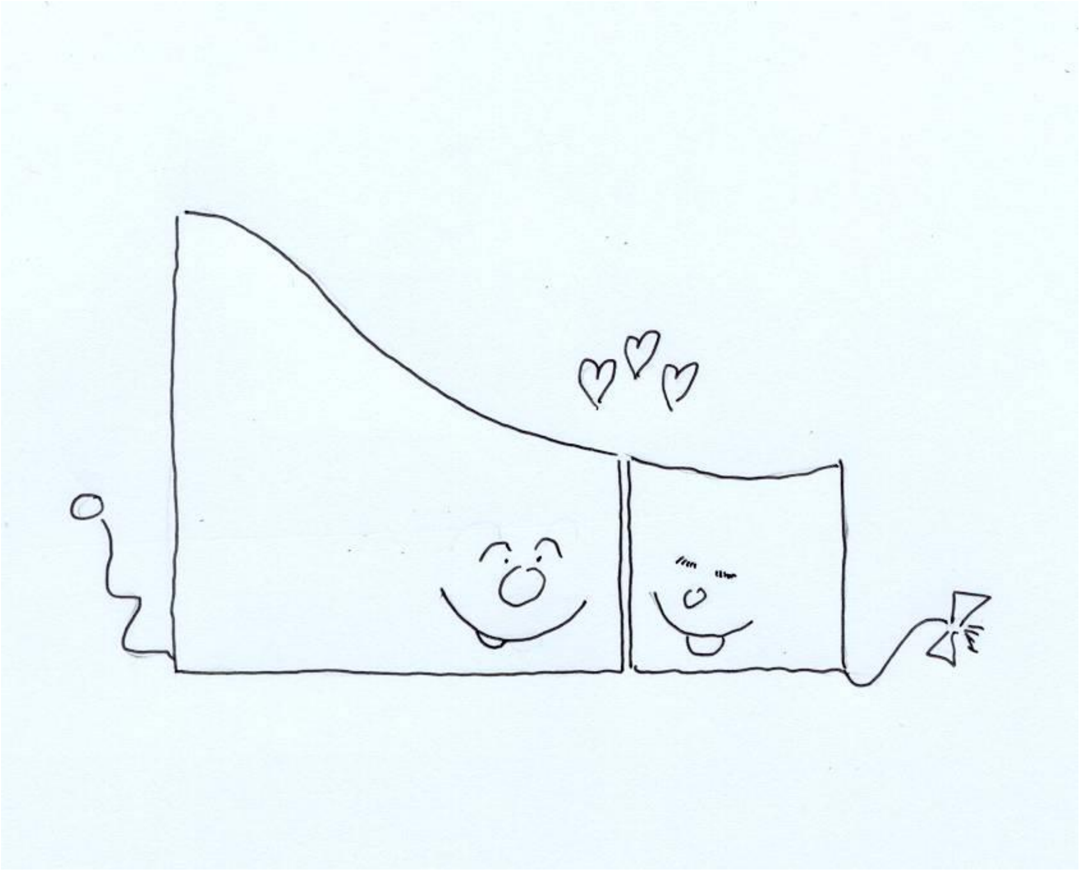
Abscisses équidistantes

2 conditions supplémentaires
Courbe spline naturelle

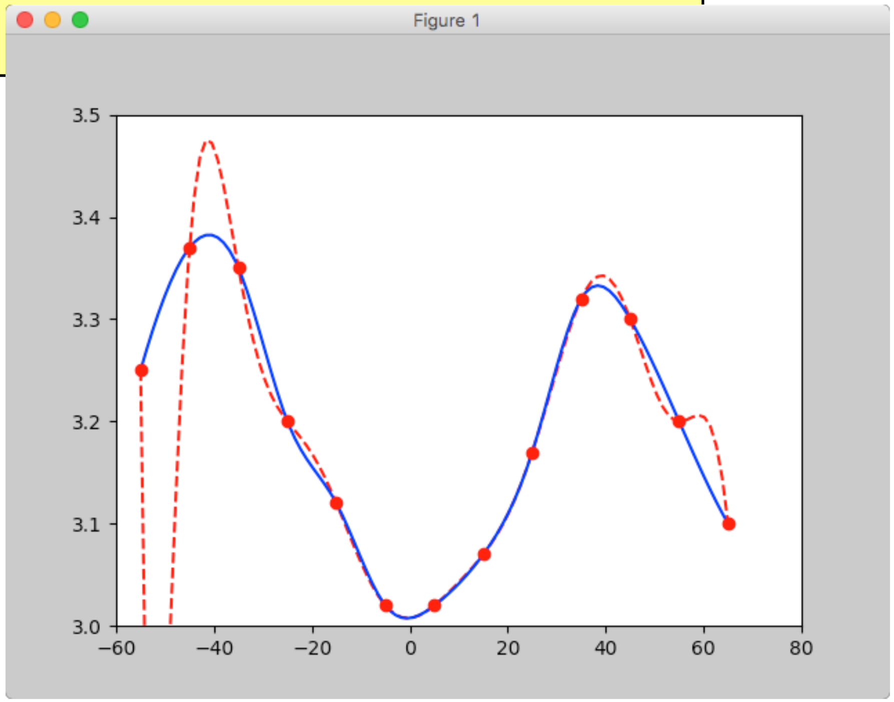
$$U_0'' = 0 \quad \text{et} \quad U_n'' = 0$$

$$\frac{h^2}{6}\begin{bmatrix} 1 & 0 & & & & & & & \\ 1 & 4 & 1 & & & & & & \\ & 1 & 4 & 1 & & & & & \\ & & 1 & 4 & 1 & & & & \\ & & & 1 & 4 & 1 & & & \\ & & & & \cdot & \cdot & \cdot & & \\ & & & & & 1 & 4 & 1 & \\ & & & & & & 1 & 4 & 1 \\ & & & & & & & 0 & 1 \end{bmatrix}\begin{bmatrix} U_0'' \\ U_1'' \\ U_2'' \\ U_3'' \\ U_4'' \\ \vdots \\ U_{n-2}'' \\ U_{n-1}'' \\ U_n'' \end{bmatrix} = \begin{bmatrix} 0 \\ U_0 - 2U_1 + U_2 \\ U_1 - 2U_2 + U_3 \\ U_2 - 2U_3 + U_4 \\ U_3 - 2U_4 + U_5 \\ \vdots \\ \vdots \\ U_{n-2} - 2U_{n-1} + U_n \\ 0 \end{bmatrix}$$

Ou de manière plus poétique…



```
from numpy import *
from scipy.interpolate import CubicSpline as spline
from matplotlib import pyplot as plt

X = arange(-55,70,10)
U = [3.25, 3.37, 3.35, 3.20, 3.12, 3.02, 3.02,
           3.07, 3.17, 3.32, 3.30, 3.20, 3.10]
x = linspace(X[0],X[-1],100)
uhLag = polyval(polyfit(X,U,len(X)-1),x)
uhSpl = spline(X,U)

plt.plot(x,uhLag,'--r',x,uhSpl(x),'-b')
plt.plot(X,U,'or')
```



Exemple