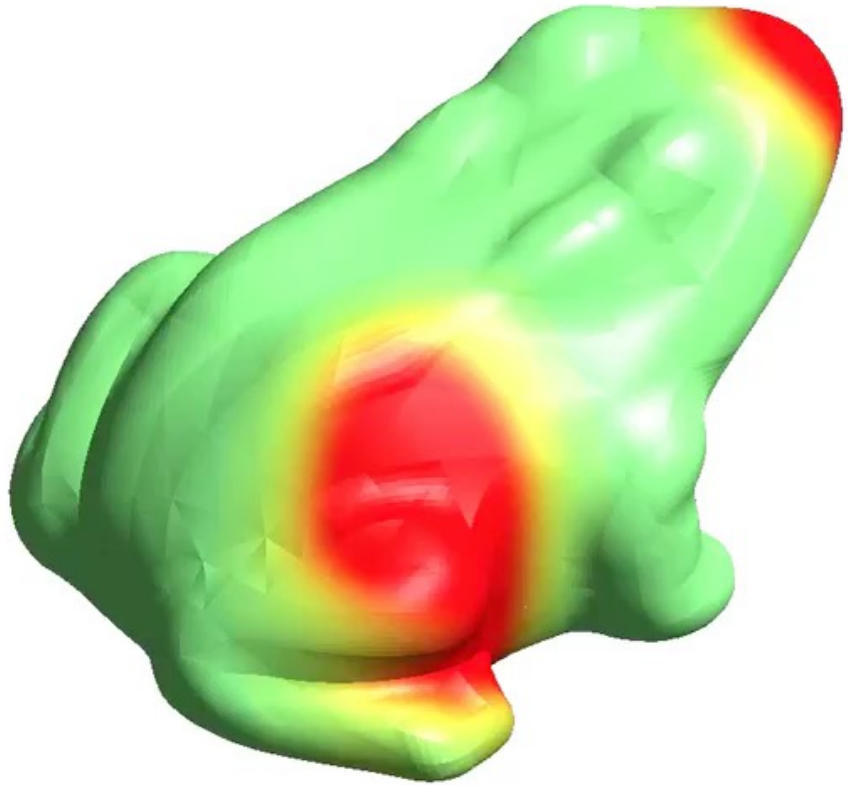
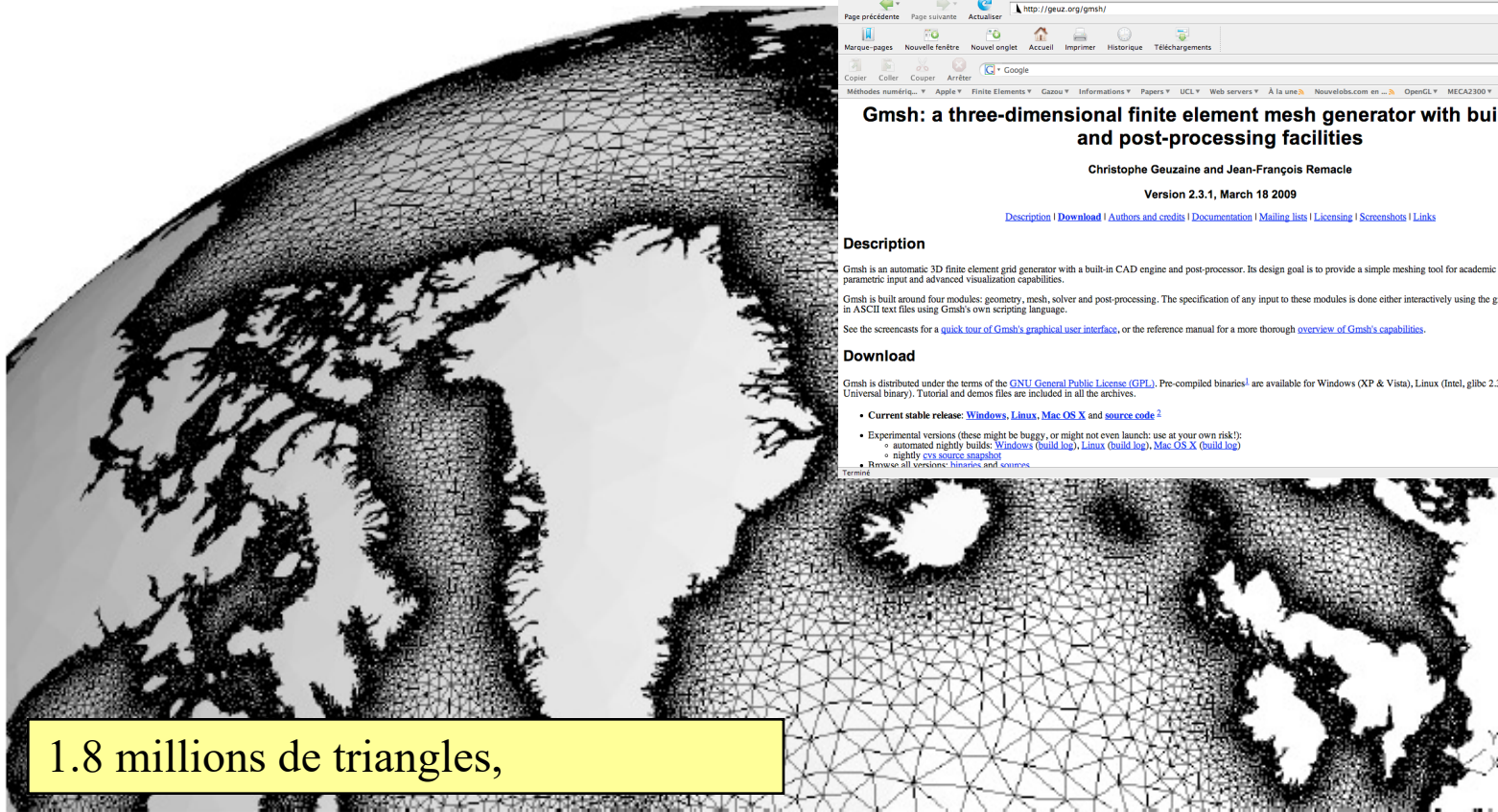
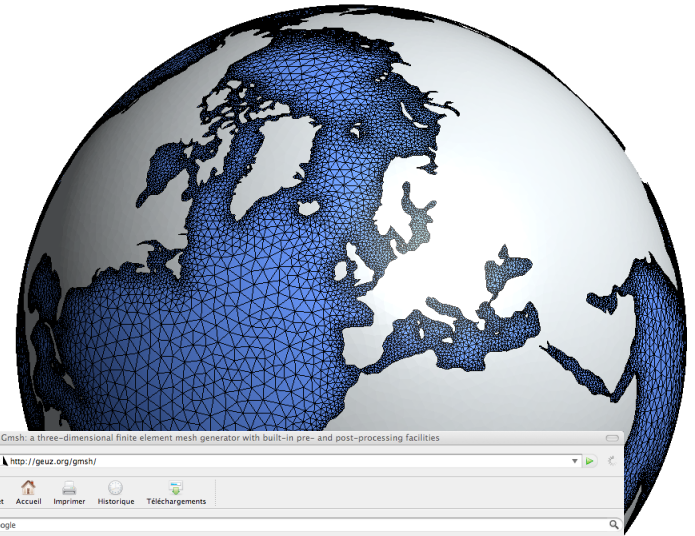


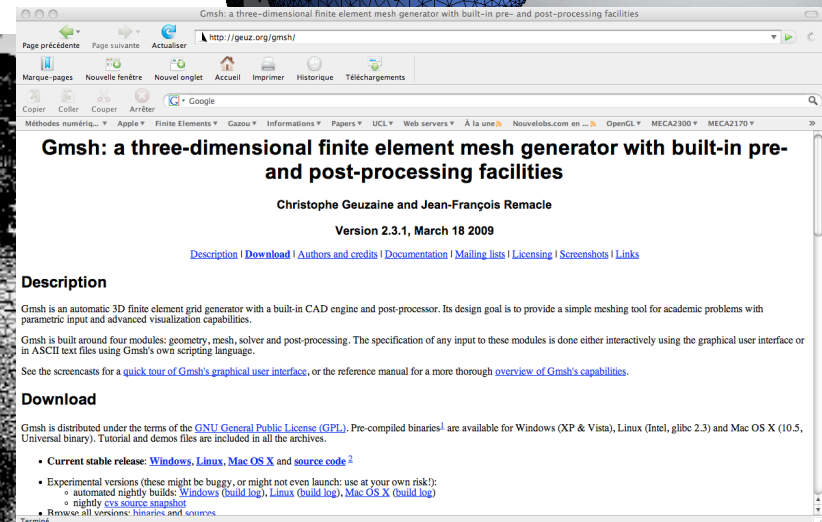
Les méthodes
numériques
A quoi cela sert-il ?



Second-generation Louvain-la-Neuve Ice-ocean Model

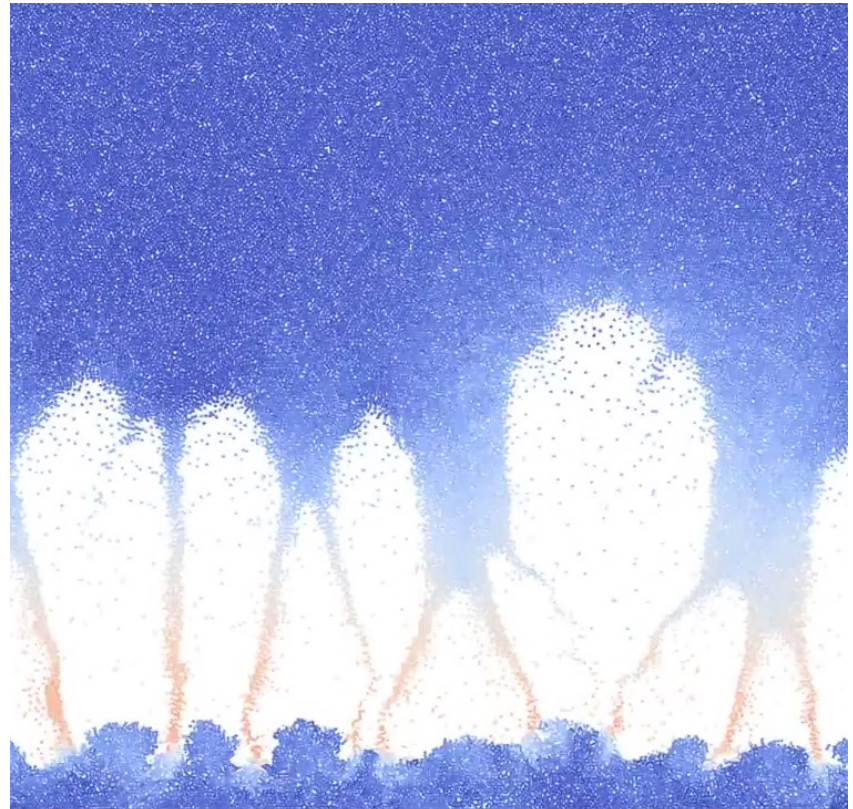


1.8 millions de triangles,



Les méthodes numériques

A quoi cela sert-il ?



Notes de cours

Notes de cours
Enoncés des exercices
Solutions des exercices
Transparents du cours
Ce qui est noté sur le tableau
= la matière de l'examen



Ecole Polytechnique de Louvain



MATHEMATIQUES
ET
S NUMERIQUES
spects facétieux
ur un ordinateur

. Legat

le cour LEPL1104
2018-2019 (version 7.1 14-12-2018)

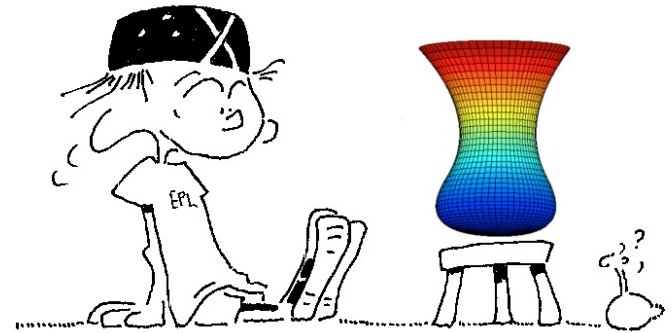


disponibles au SICI et on-line sur le Web

Livres de références...

- CHARLES F. VAN LOAN, *Introduction to Scientific Computing*, Second Edition, Prentice Hall, Upper Saddle River, ISBN 0-13949157-0 (1999).
- JACQUES RAPPAZ, MARCO PICASSO, *Introduction à l'analyse numérique*, Presses polytechniques et universitaires romandes, Lausanne, ISBN 2-88074363-X (2000).
- ANDRÉ FORTIN, *Analyse numérique pour ingénieurs*, Seconde Edition, Presses internationales polytechniques, Montréal, ISBN 2-55300936-4 (2001).
- WILLIAM L. BRIGGS, VAN EMDEN HENSON, STEVE F. MCCORMICK, *A Multigrid Tutorial*, Second Edition, SIAM, Philadelphia, ISBN 0-89871462-1 (2000).
- BRIGITTE LUCQUIN, OLIVIER PIRONNEAU, *Introduction to Scientific Computing*, John Wiley & Sons, New York, ISBN 0-47197266-X (1998).
- ALFIO QUARTERONI, FAUSTO SALERI, *Scientific Computing with MATLAB*, Springer-Verlag, Berlin, ISBN 3-35044363-0 (2003).
- DESMOND J. HIGHAM, NICHOLAS J. HIGHAM *Matlab Guide*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, ISBN 0-89871469-9 (2000).
- MICHAEL T. HEATH *Scientific Computing : an Introduction Survey*, McGraw Hill, New-York, ISBN 0-07-115336-5 (1997).
- K. E. ATKINSON, *An Introduction to Numerical Analysis*, Second Edition, John Wiley & Sons, New York (1989).
- S. D. CONTE, C. DE BOOR, *Elementary Numerical Analysis, An Algorithmic Approach*, Third Edition, McGraw-Hill Book Company, New York (1980).
- B.M. IRONS, N.G. SHRIVE, *Numerical Methods in Engineering and Applied Sciences : numbers are fun*, Second Edition, John Wiley and Sons (1987).
- JOHN H. MATHEWS, *Numerical Methods for Mathematics, Science and Engineering*, Second Edition, Prentice Hall, Englewood Cliffs, ISBN 0-13624990-6 (1992).
- W. H. PRESS, S. A. TEUKOLSKY, W. T. VETTERLING, B. P. FLANNERY *Numerical Recipes in C: The Art of Scientific Computing*, Second Edition, Cambridge University Press, Cambridge (1994).

Comment contacter le titulaire du cours?



- Il n'est pas possible de prendre rendez-vous auprès de moi.
- Il n'est pas possible de me consulter par courriel.
- Si vous me trouvez dans mon bureau et que mon humeur volage est positive, je vous consacrerai un peu de temps (et même parfois beaucoup...)
- Si il est indiqué sur ma porte **ne pas déranger**, mon humeur volage est négative. Il ne faut donc pas me déranger, car votre demande risque de ne pas être traitée avec toute la douceur requise.

Plan des cours de méthodes numériques

**Comment interpoler
une fonction ?**

**Comment approximer
une fonction ?**

**Comment intégrer
numériquement
une fonction ?**

**Comment dériver
numériquement
une fonction ?**

**Comment résoudre
numériquement un
problème aux
valeurs initiales ?**

**Comment résoudre
numériquement un
problème aux
conditions frontières ?**

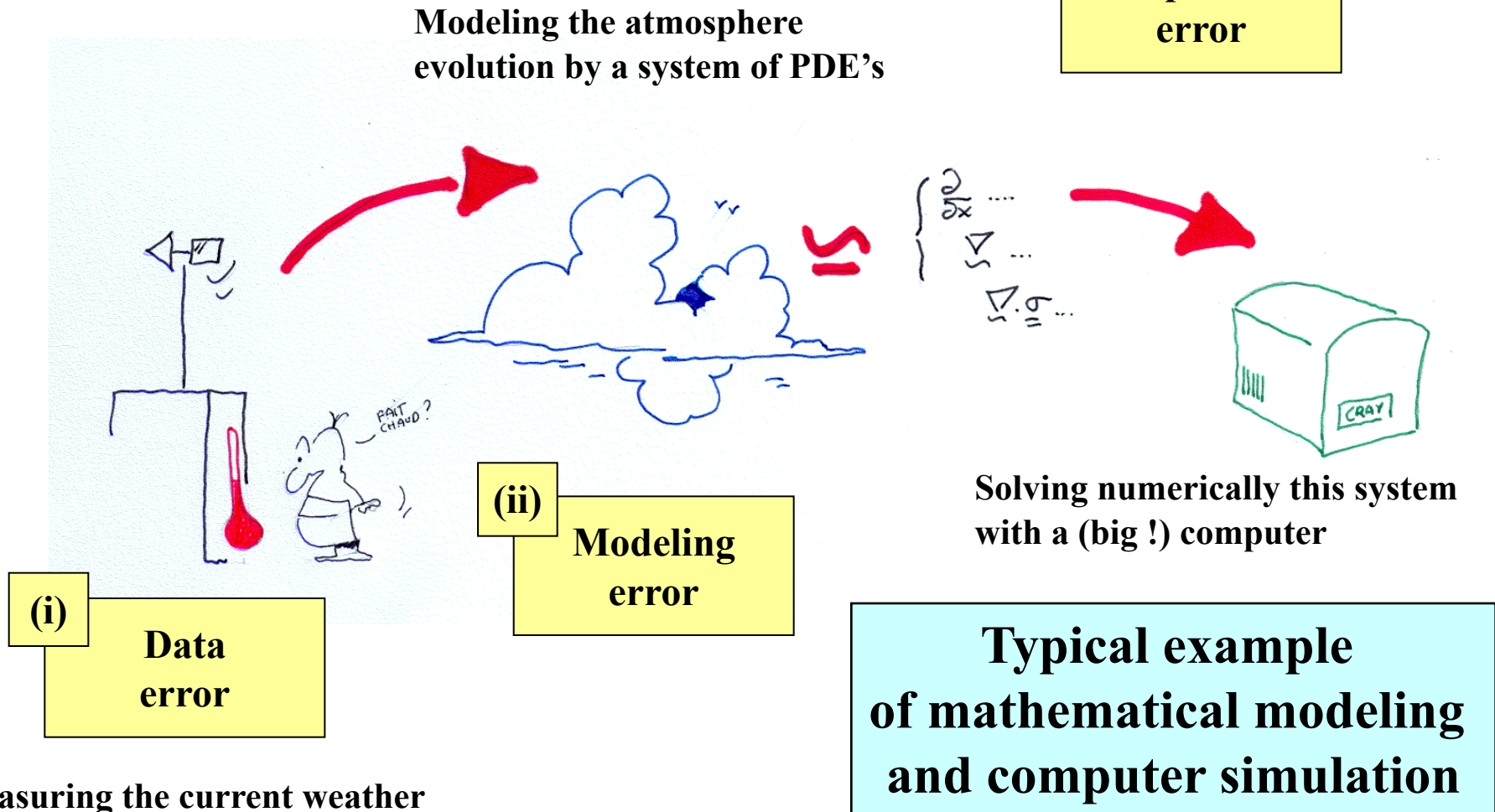
Et les équations non linéaires ?

Et les méthodes itératives ?

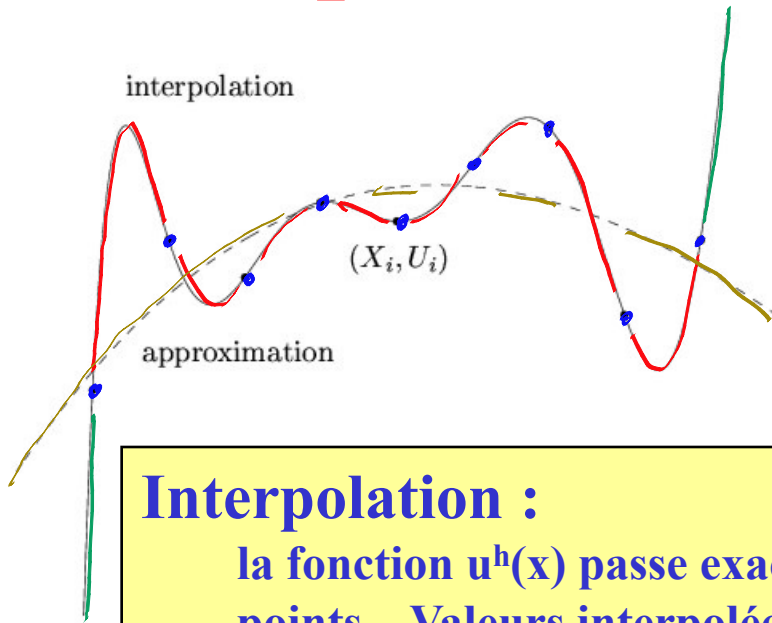
**Comment résoudre
numériquement une
équation aux dérivées
partielles ?**

*Comment résoudre numériquement
une équation différentielle ?*

Weather forecasts are not always reliable...



Interpolation, approximation et extrapolation...



Interpolation :

la fonction $u^h(x)$ passe exactement par les points. Valeurs interpolées entre les points et valeurs **extrapolées** hors de l'intervalle.

Approximation :

la fonction $u^h(x)$ ne passe pas par les points, mais s'en rapproche selon un critère à définir

Fonctions de base
spécifiées a priori

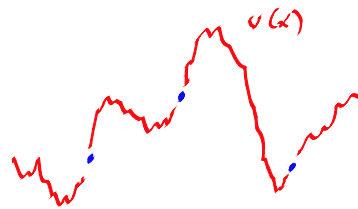
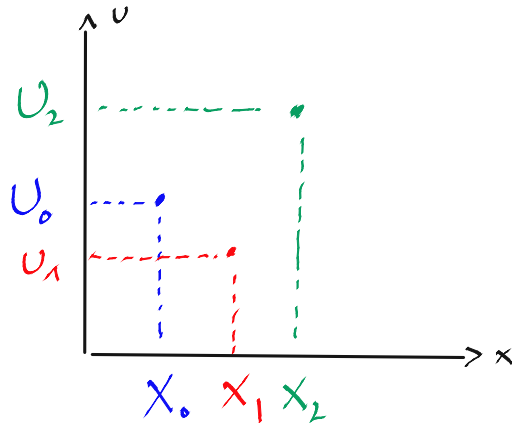


$$u(x) \approx u^h(x) = \sum_{j=0}^n a_j \phi_j(x)$$

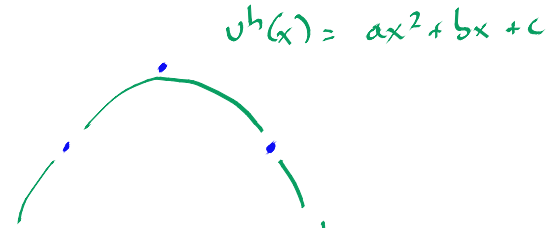


Paramètres inconnus

Matrice de Vandermonde



$v(x) \in \mathcal{U}$
 $\dim(\mathcal{U}) = \infty$



$v^h \in \mathcal{U}^h \subset \mathcal{U}$
 $\dim(\mathcal{U}^h) = 3$

$$v(x) \approx v^h(x) = a_0 \underbrace{\phi_0(x)}_{\substack{\in \mathbb{R} \\ \text{INCONNU}}} + a_1 \underbrace{\phi_1(x)}_{\substack{\in \mathcal{U} \\ \text{FIXES A PRIORI} \\ \text{CONNUES}}} + a_2 \phi_2(x)$$

? $a_0 \ a_1 \ a_2$

tel que

$$\underbrace{v(x_0)}_{v_0} = v^h(x_0) \\ v_1 = v^h(x_1) \\ v_2 = v^h(x_2)$$

EXEMPLE

$$\phi_0(x) = 1$$

$$\phi_1(x) = x$$

$$\phi_2(x) = x^2$$

$$? a_0 a_1 a_2$$

tel que

$$\underbrace{v(X_0)}_{U_0} = v^h(X_0)$$

$$U_1 = v^h(X_1)$$

$$U_2 = v^h(X_2)$$

EXEMPLE

$$\phi_0(x) = 1$$

$$\phi_1(x) = x$$

$$\phi_2(x) = x^2$$

$$\sum_{i=0}^n a_i \phi_i(x_j) = U_j$$

$$j = 0 \dots n$$

$$\begin{cases} a_0 \phi_0(x_0) + a_1 \phi_1(x_0) + a_2 \phi_2(x_0) = U_0 \\ a_0 \phi_0(x_1) + a_1 \phi_1(x_1) + a_2 \phi_2(x_1) = U_1 \\ a_0 \phi_0(x_2) + a_1 \phi_1(x_2) + a_2 \phi_2(x_2) = U_2 \end{cases}$$

$$\underbrace{\begin{bmatrix} \phi_0(x_0) & \phi_1(x_0) & \phi_2(x_0) \\ \phi_0(x_1) & \phi_1(x_1) & \phi_2(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \phi_2(x_2) \end{bmatrix}}_{\underline{\underline{A}}} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}}_{\underline{x}} = \underbrace{\begin{bmatrix} U_0 \\ U_1 \\ U_2 \end{bmatrix}}_{\underline{v}}$$

$$\underline{\underline{A}} \cdot \underline{x} = \underline{v}$$

FORGET ALGEBRA !

$$\underline{x} = \underline{\underline{A}}^{-1} \cdot \underline{v}$$

NO!

Et ce qu'on trouve
dans le syllabus

ON N'INVERSE
PAS UN SYSTEME
ON LE RESOUT !

Problème de l'interpolation

Trouver $(a_0, \dots, a_n) \in \mathbb{R}^{n+1}$ tels que

$$\underbrace{\sum_{j=0}^n a_j \phi_j(X_i)}_{u^h(X_i)} = U_i \quad i = 0, 1, \dots, n.$$


$$\begin{bmatrix} \phi_0(X_0) & \phi_1(X_0) & \dots & \phi_n(X_0) \\ \phi_0(X_1) & \phi_1(X_1) & \dots & \phi_n(X_1) \\ \phi_0(X_2) & \phi_1(X_2) & \dots & \phi_n(X_2) \\ \phi_0(X_3) & \phi_1(X_3) & \dots & \phi_n(X_3) \\ \phi_0(X_4) & \phi_1(X_4) & \dots & \phi_n(X_4) \\ \vdots & \vdots & & \vdots \\ \phi_0(X_n) & \phi_1(X_n) & \dots & \phi_n(X_n) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} U_0 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \\ \vdots \\ U_n \end{bmatrix}$$

Cas particulier : Interpolation polynomiale

$$\phi_j(x) = x^j \quad j = 0, 1, 2, \dots, n.$$



$$u^h(x) = \sum_{j=0}^n a_j x^j$$

Matrice de Vandermonde

$$\begin{bmatrix} 1 & X_0 & \dots & X_0^n \\ 1 & X_1 & \dots & X_1^n \\ \vdots & \vdots & & \vdots \\ 1 & X_n & \dots & X_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} U_0 \\ U_1 \\ \vdots \\ U_n \end{bmatrix}$$

Unicité de l'interpolation polynomiale :

Il existe **un et un seul polynôme d'interpolation** de degré n au plus qui passe par $n + 1$ points d'abscisses distinctes.

Alexandre-Théophile Vandermonde

Né le 28 février 1735 à Paris

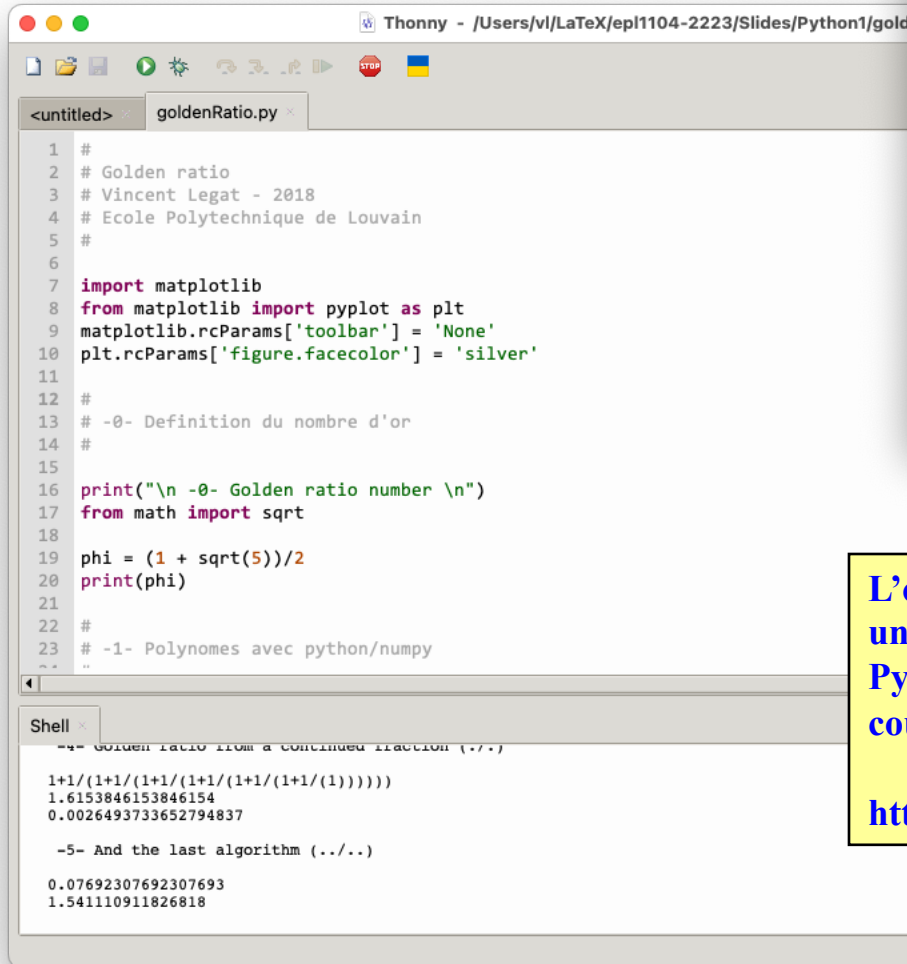
Décédé le 1er janvier 1796 à Paris

Perhaps the name of Vandermonde is best known today for the Vandermonde determinant. While it is certainly true that he made a major contribution to the theory of determinants, yet nowhere in his four mathematical papers does this determinant appear. It is rather strange, therefore, that this determinant should be named after him and several authors have puzzled over the fact for some time.

Lesbesgue's conjecture (first published in 1940) that **it resulted for someone misreading Vandermonde's notation**, and therefore believing that this determinant was in his work, seems the most likely.

<http://www-gap.dcs.st-and.ac.uk/~history/Mathematicians/Vandermonde.html>

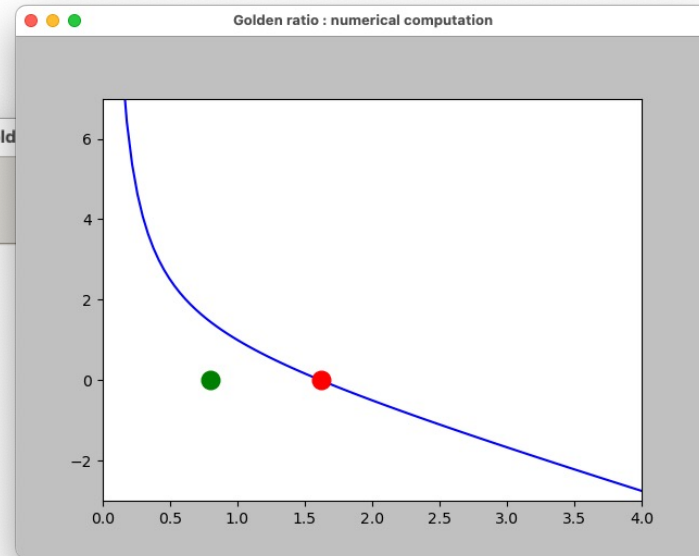
Un éditeur Python pour débutants !



```
<untitled> goldenRatio.py x
1 #
2 # Golden ratio
3 # Vincent Legat - 2018
4 # Ecole Polytechnique de Louvain
5 #
6
7 import matplotlib
8 from matplotlib import pyplot as plt
9 matplotlib.rcParams['toolbar'] = 'None'
10 plt.rcParams['figure.facecolor'] = 'silver'
11
12 #
13 # -0- Definition du nombre d'or
14 #
15
16 print("\n -0- Golden ratio number \n")
17 from math import sqrt
18
19 phi = (1 + sqrt(5))/2
20 print(phi)
21
22 #
23 # -1- Polynomes avec python/numpy
24 "
```

```
Shell x
-4- Golden ratio from a continued fraction (./.)
1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1))))))
1.6153846153846154
0.0026493733652794837

-5- And the last algorithm (./...)
0.07692307692307693
1.541110911826818
```

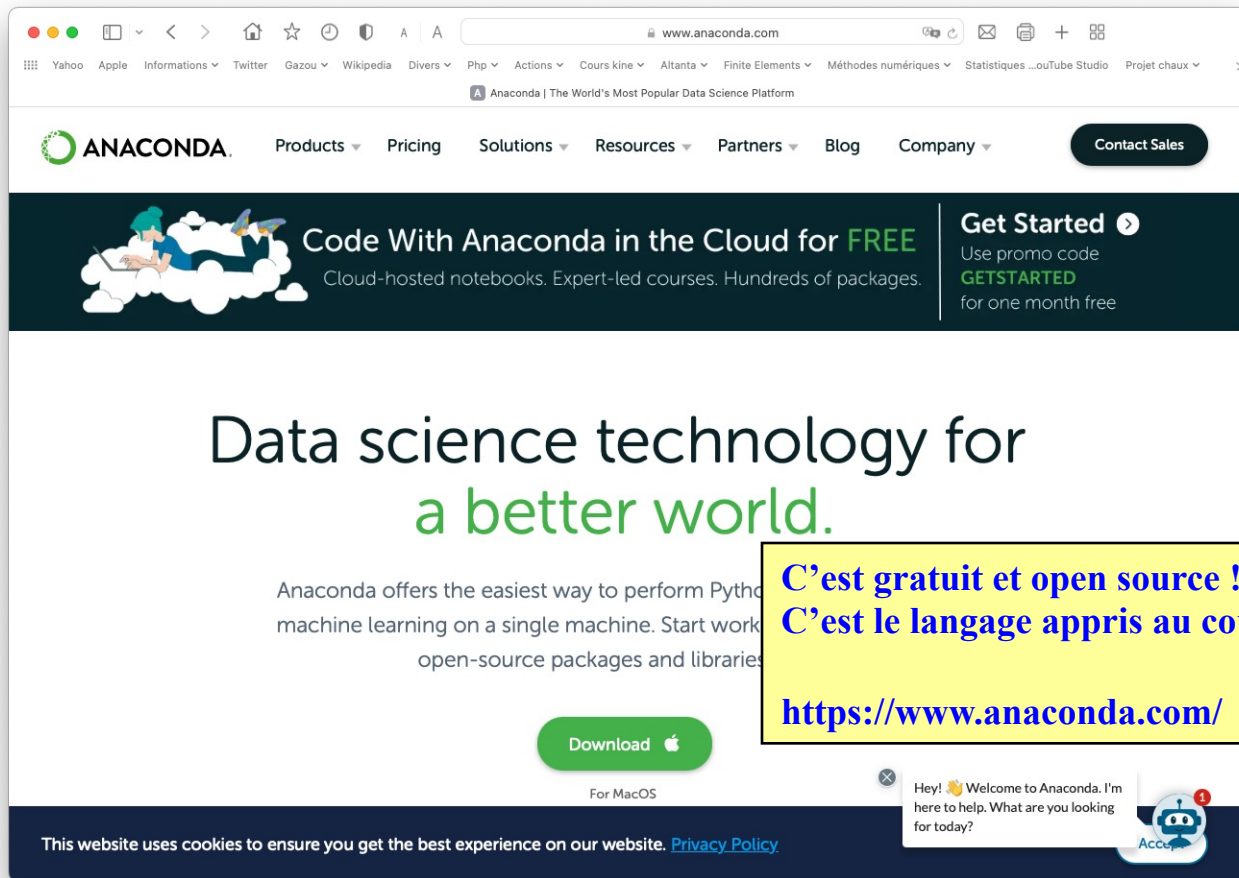


L'option la plus simple et la facile pour exécuter et éditer un programme Python après avoir installé l'interpréteur Python avec Anaconda : c'est aussi l'option choisie dans le cours d'informatique : LEPL1401

<https://thonny.org>

Python, python !

Plein de choses super fun !



The screenshot shows the Anaconda website homepage. At the top, there's a navigation bar with the Anaconda logo and links for Products, Pricing, Solutions, Resources, Partners, Blog, and Company. A 'Contact Sales' button is also present. Below the navigation bar, a dark blue banner features an illustration of a person coding on a laptop, with the text 'Code With Anaconda in the Cloud for FREE' and 'Cloud-hosted notebooks. Expert-led courses. Hundreds of packages.' To the right of this banner, a 'Get Started' button is visible, along with a promo code 'GETSTARTED' for one month free. The main heading reads 'Data science technology for a better world.' Below this, a paragraph states 'Anaconda offers the easiest way to perform Python machine learning on a single machine. Start working with open-source packages and libraries.' A green 'Download' button with an Apple logo is shown, with 'For MacOS' written below it. At the bottom, a cookie notice states 'This website uses cookies to ensure you get the best experience on our website. Privacy Policy.' A small chatbot window is open in the bottom right corner, saying 'Hey! Welcome to Anaconda. I'm here to help. What are you looking for today?'.

www.anaconda.com

Yahoo Apple Informations Twitter Gazou Wikipedia Divers Php Actions Cours kine Altanta Finite Elements Méthodes numériques Statistiques YouTube Studio Projet chaud >>

ANAconda | The World's Most Popular Data Science Platform

ANAconda Products Pricing Solutions Resources Partners Blog Company Contact Sales

Code With Anaconda in the Cloud for **FREE**
Cloud-hosted notebooks. Expert-led courses. Hundreds of packages.

Get Started
Use promo code
GETSTARTED
for one month free

Data science technology for a better world.

Anaconda offers the easiest way to perform Python machine learning on a single machine. Start working with open-source packages and libraries.

Download 

For MacOS

This website uses cookies to ensure you get the best experience on our website. [Privacy Policy](#)

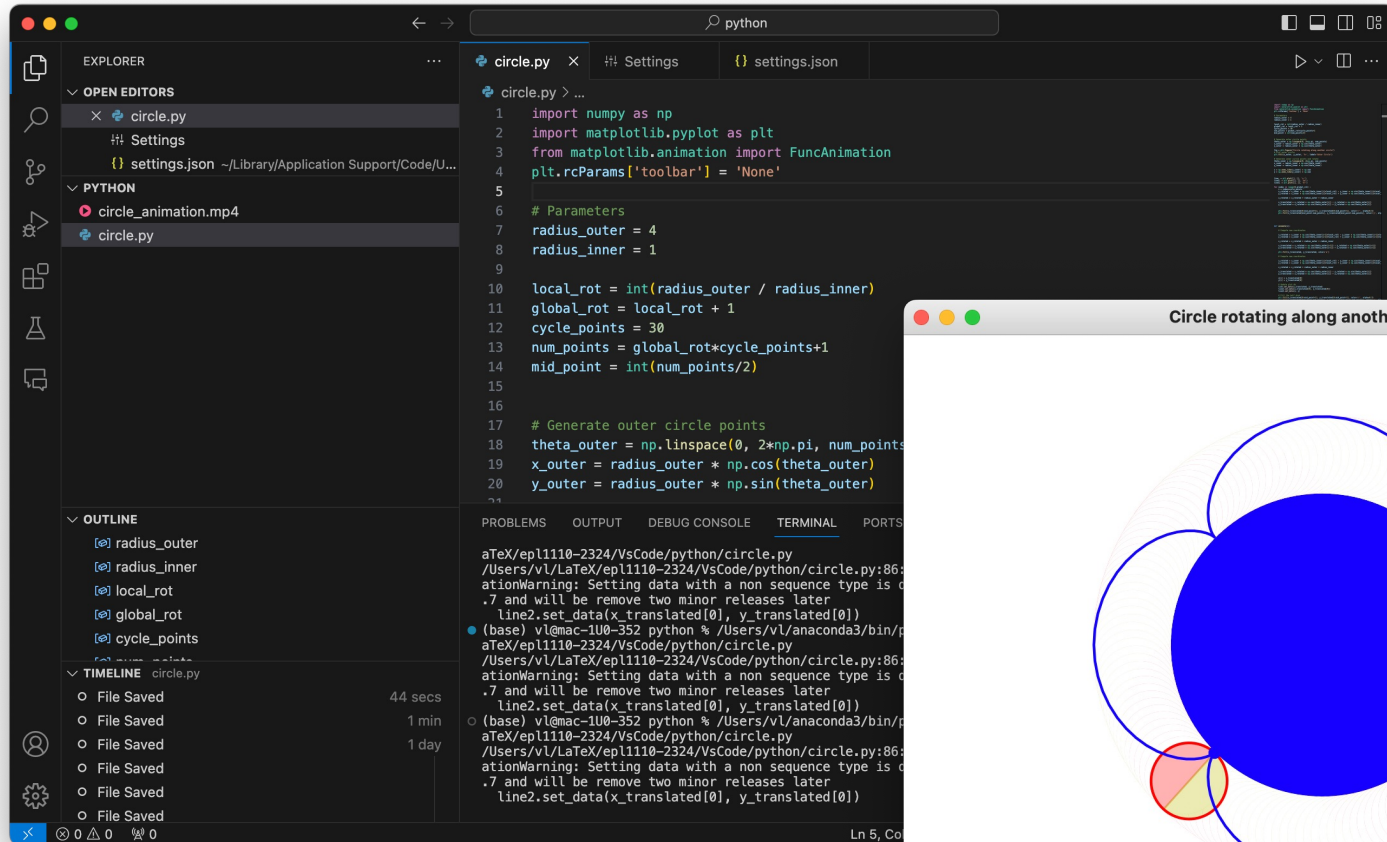
Hey! 🐍 Welcome to Anaconda. I'm here to help. What are you looking for today?

C'est gratuit et open source !

C'est le langage appris au cours d'informatique !

<https://www.anaconda.com/>

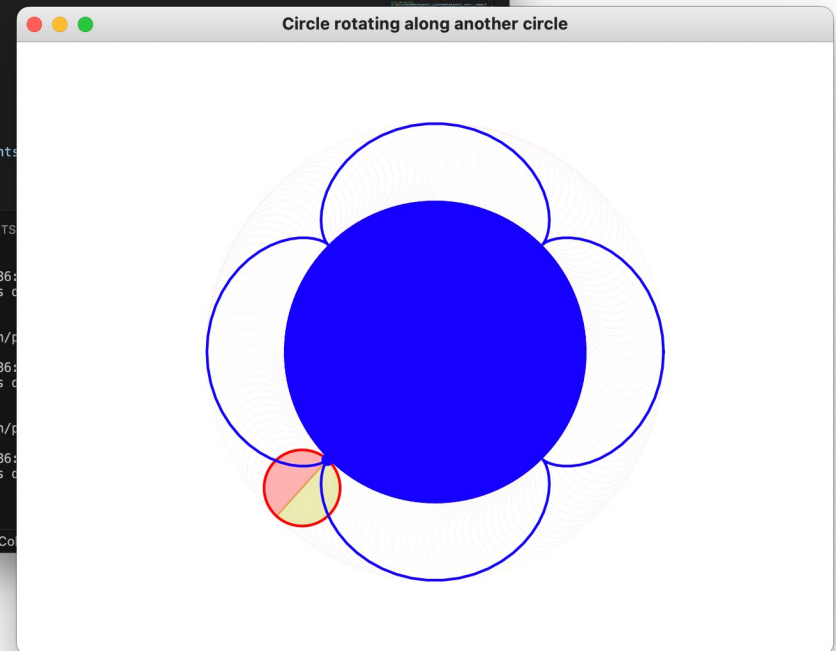
Comment avoir l'intelligence artificielle sur votre ordinateur ?



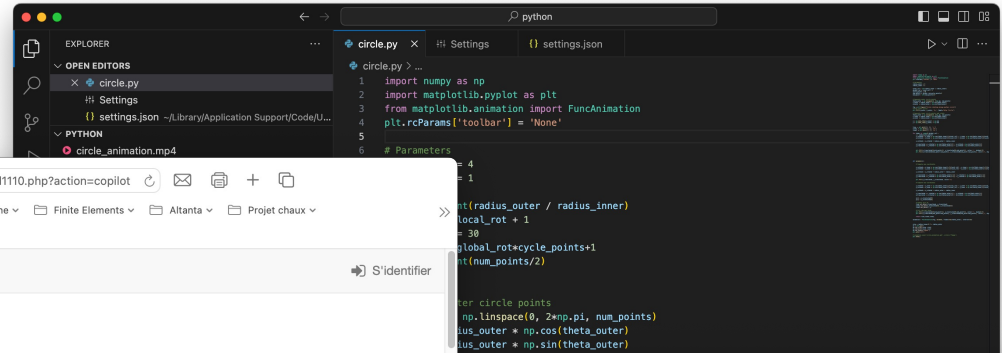
The screenshot shows a VS Code editor with a Python script named `circle.py`. The script uses `numpy` and `matplotlib.pyplot` to generate a series of overlapping circles. The parameters are defined as follows:

- `radius_outer = 4`
- `radius_inner = 1`
- `local_rot = int(radius_outer / radius_inner)`
- `global_rot = local_rot + 1`
- `cycle_points = 30`
- `num_points = global_rot * cycle_points + 1`
- `mid_point = int(num_points / 2)`

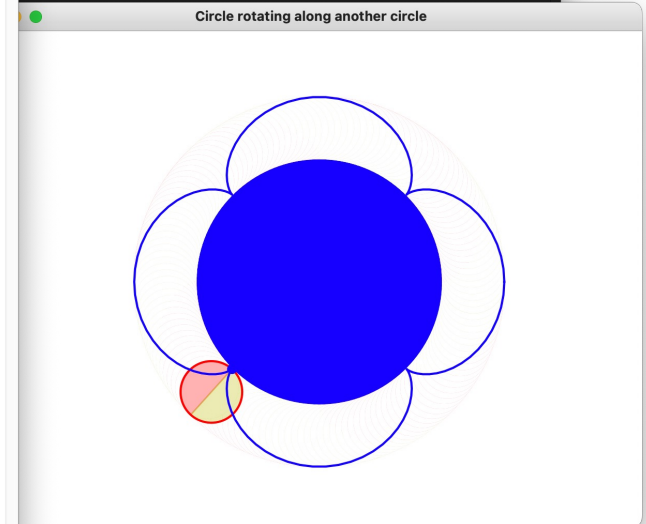
The script generates the outer circle points using `np.linspace(0, 2*np.pi, num_points)` and `np.cos`/`np.sin` for the x and y coordinates. The terminal output shows the execution of the script, with a warning about setting data with a non-sequence type.



Ce n'est pas bien compliqué ! Il suffit d'être astucieux !



```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from matplotlib.animation import FuncAnimation
4 plt.rcParams['toolbar'] = 'None'
5
6 # Parameters
7 radius_outer = 4
8 radius_inner = 1
9 local_rot = 1
10 num_points = 30
11 global_rot = cycle_points + 1
12 int(num_points/2)
13
14 # Circle points
15 np.linspace(0, 2*np.pi, num_points)
16 radius_outer * np.cos(theta_outer)
17 radius_outer * np.sin(theta_outer)
```



perso.uclouvain.be/vincent.legat/zouLab/epl1110.php?action=copilot

zouLab 1.0.0 : epl1110

EPL1110 News Horaire Documents S'identifier



Introduction aux éléments finis (LEPL1110)

Vincent Legat
Jean-François Remacle
Louvain School of Engineering
Université catholique de Louvain

Il faut d'abord t'identifier :-)

News Documents Videos & podcasts !

Installer vscode et copilot sur votre ordinateur ! Programmes Python Devoirs en C

Compilation et exécution d'un programme

Pour construire votre projet d'éléments finis, nous allons utiliser un certain nombre d'outils. Cette année, nous ferons appel à l'intelligence artificielle pour vous aider à écrire votre code. Pour cela, nous allons utiliser un éditeur de code, Vscode, et un plugin, Copilot, qui est une intelligence artificielle développée par OpenAI. Nous allons également utiliser un gestionnaire de version, Git, et un dépôt de code, GitHub. Enfin, nous allons utiliser un compilateur, GCC, pour compiler notre code. Nous allons également utiliser un outil de gestion de projet, CMake, pour générer les fichiers de configuration de notre projet. Evidemment, tout cela peut paraître compliqué, mais nous allons vous guider pas à pas pour vous aider à maîtriser ces outils. Et même pour écrire cette page web, l'intelligence artificielle de Copilot m'a aidé à écrire ce texte. C'est assez magique en fait !

Installation de VsCode

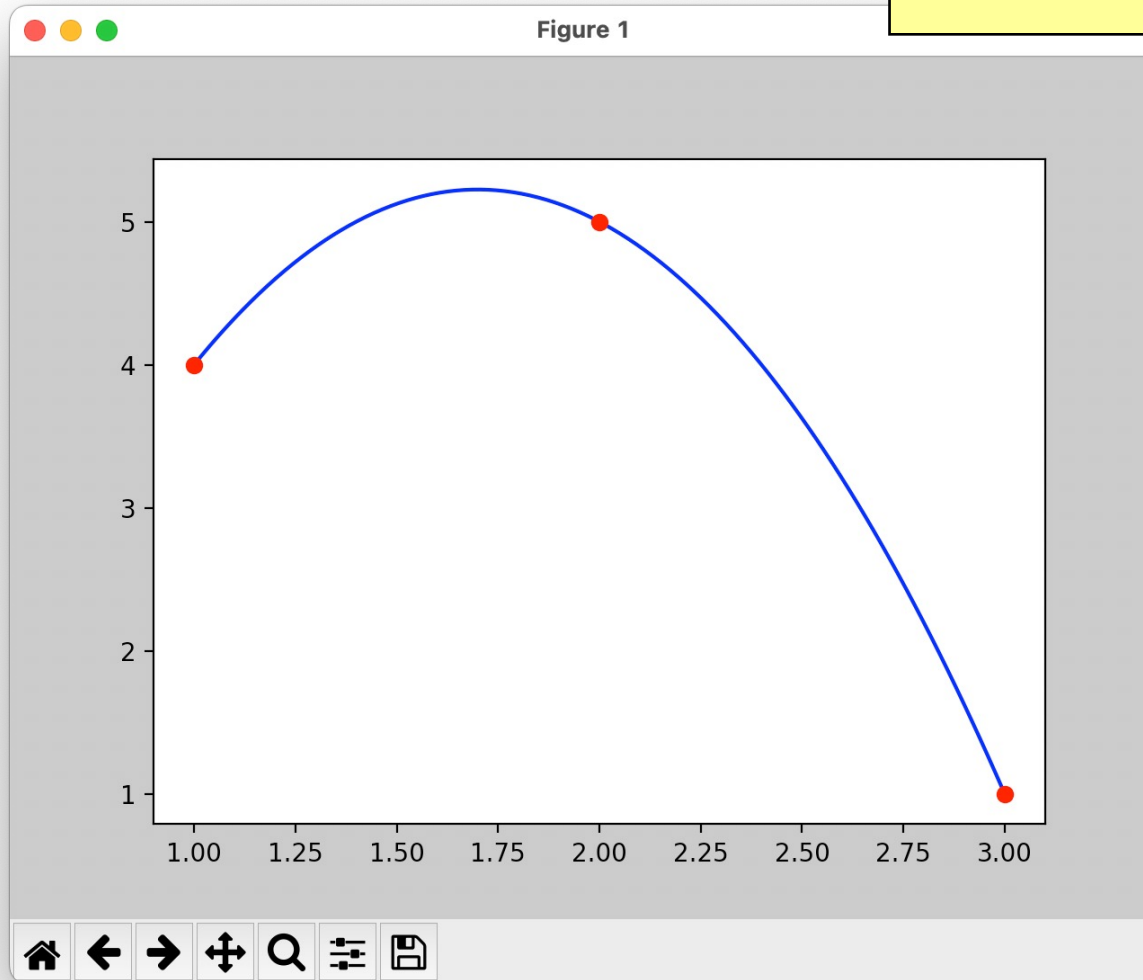
Attention : **nous utiliserons VsCode (Visual Studio Code) qui est un logiciel distinct de Visual Studio**. Visual Studio est un environnement de développement intégré (IDE Integrated Development Environment), tandis que VsCode est un éditeur de code léger et polyvalent.

© 2020 Vincent Legat

Contact - Support

Faisons le calcul

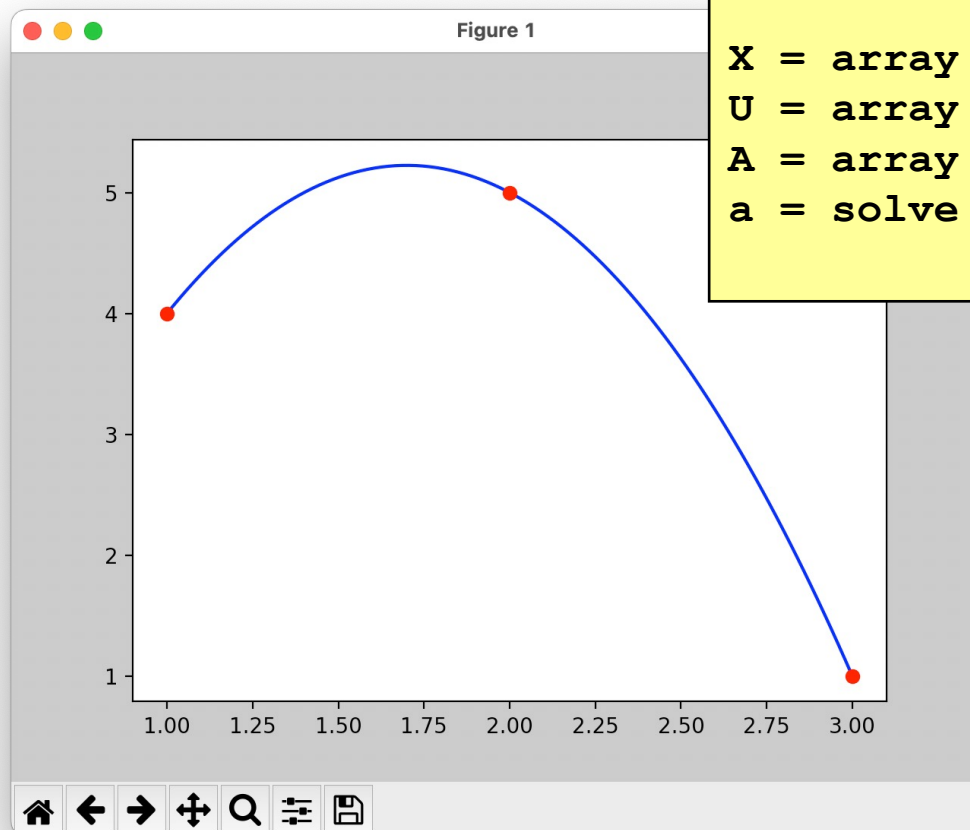
```
X = array([1,2,3])  
U = array([4,5,1])  
A = array([X**0,X**1,X**2]).T  
a = solve(A,U)
```



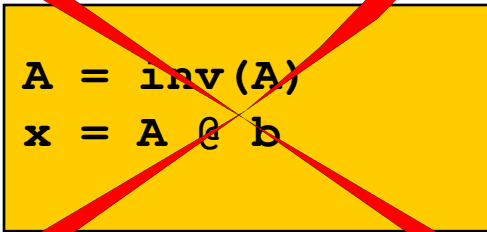
Faisons le calcul avec numpy

```
from numpy import *  
from numpy.linalg import solve
```

```
X = array([1,2,3])  
U = array([4,5,1])  
A = array([X**0,X**1,X**2]).T  
a = solve(A,U)
```



A propos de la résolution d'un système linéaire...



```
A = inv(A)
x = A \ b
```

```
x = solve(A,b)
```

*On résout un système linéaire,
on ne l'inverse jamais....
(J. Meinguet)*

A frequent misuse of **inv** arises when solving the system of linear equations. One way to solve this is with

```
x = inv(A)*b
```

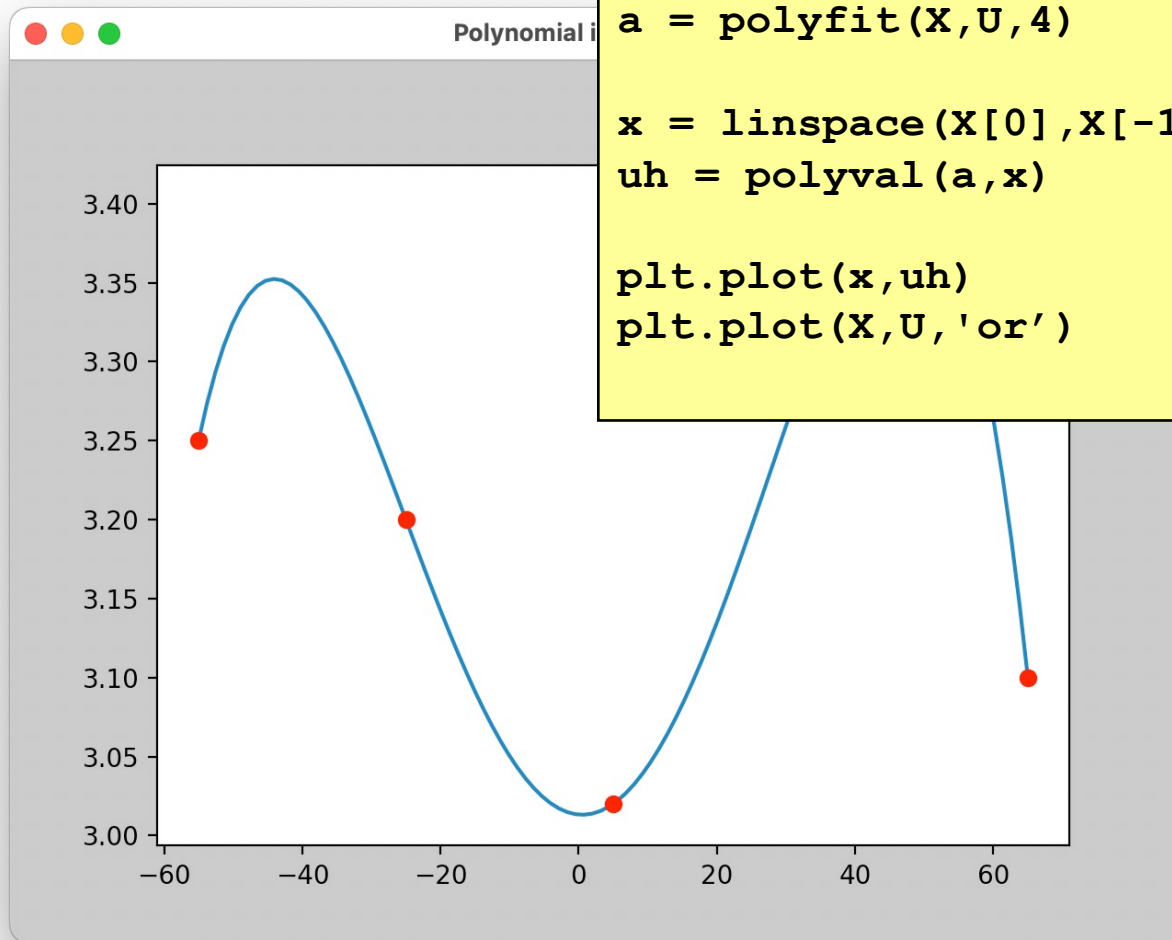
A better way, from both an execution time and numerical accuracy standpoint, is to use the matrix division operator

```
x = A\b
```

This produces the solution using Gaussian elimination, without forming the inverse.

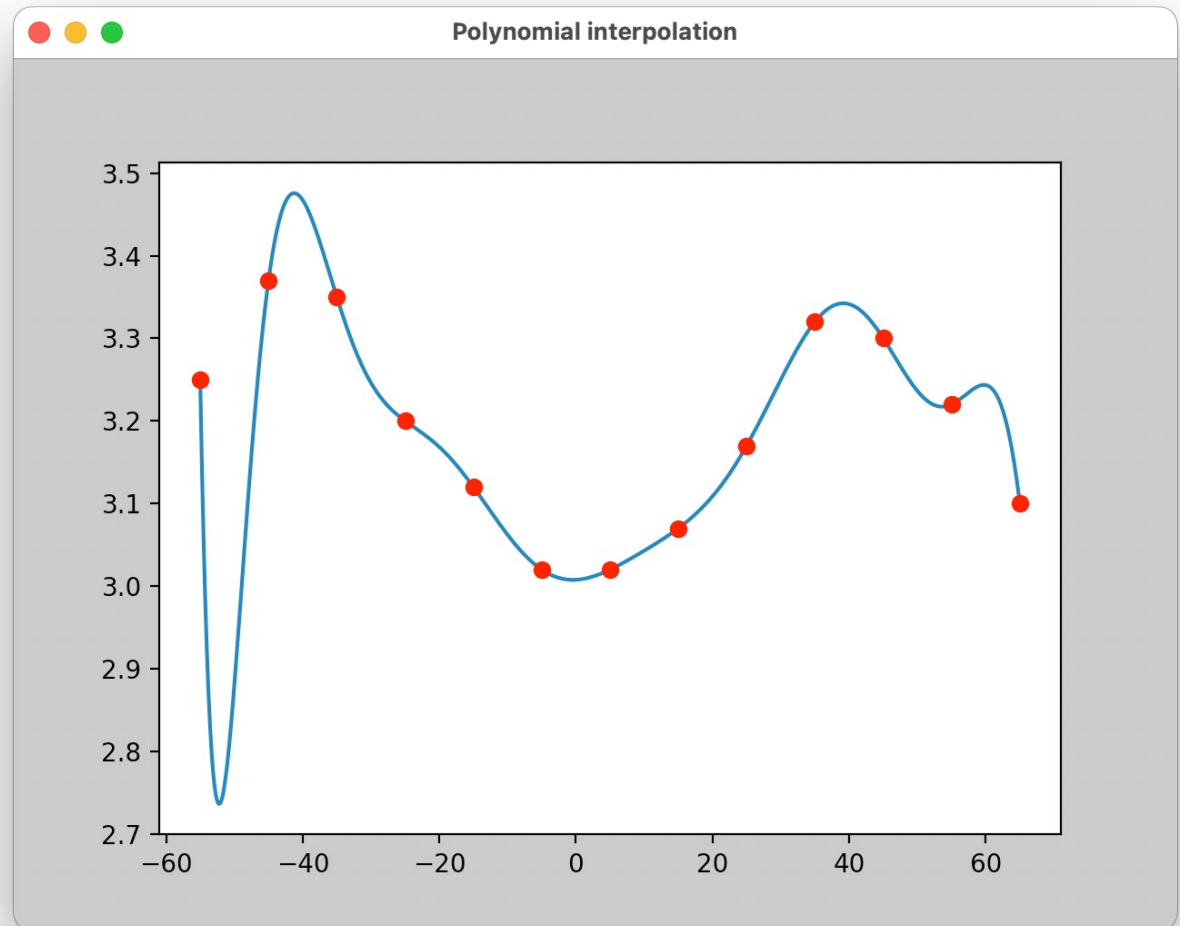
Exemple

```
from numpy import *  
from matplotlib import pyplot as plt  
  
X = [ -55, -25,  5,  35,  65]  
U = [3.25, 3.20, 3.02, 3.32, 3.10]  
a = polyfit(X,U,4)  
  
x = linspace(X[0],X[-1],100)  
uh = polyval(a,x)  
  
plt.plot(x,uh)  
plt.plot(X,U, 'or')
```

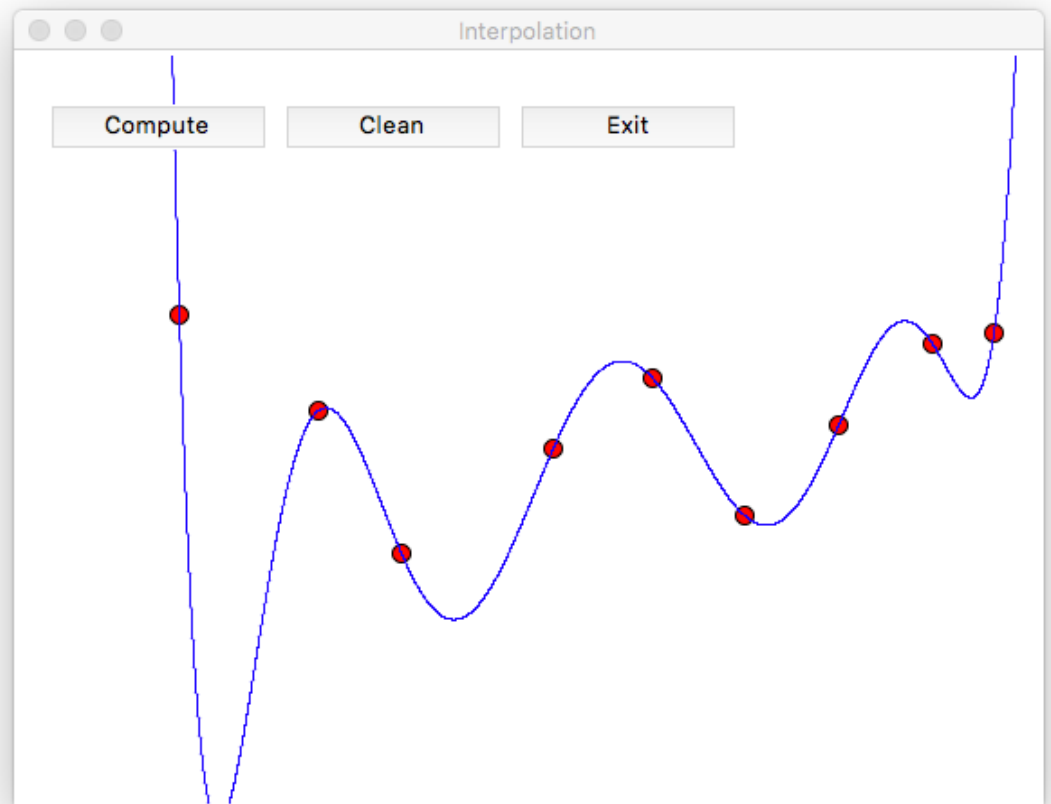


Davantage de données...

Latitude	
65	3.10
55	3.22
45	3.30
35	3.32
25	3.17
15	3.07
5	3.02
-5	3.02
-15	3.12
-25	3.20
-35	3.35
-45	3.37
-55	3.25



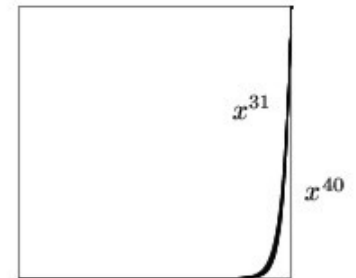
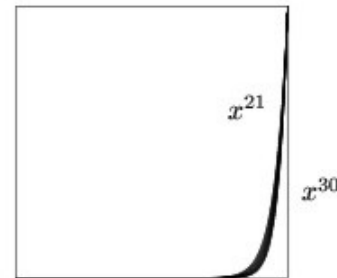
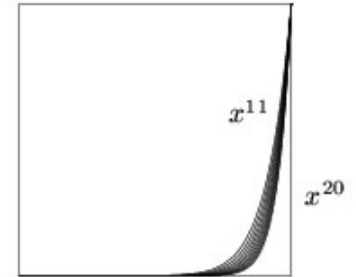
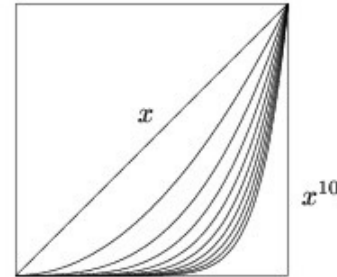
Le polynôme d'interpolation est
un être bien turbulent...



Les monômes



$$u^h(x) = \sum_{j=0}^n a_j x^j$$



$$\begin{bmatrix} 1 & X_0 & \dots & X_0^n \\ 1 & X_1 & \dots & X_1^n \\ \vdots & \vdots & & \vdots \\ 1 & X_n & \dots & X_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} U_0 \\ U_1 \\ \vdots \\ U_n \end{bmatrix}$$

Matrice de Vandermonde...

Le système linéaire devient
de plus en plus **mal conditionné**,
lorsque n augmente !

C'est quoi un
système mal
conditionné ?

Données



$$\begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 11 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$



Solution exacte :
a=2.0 et b=0.0

Système linéaire mal conditionné

Un système linéaire est mal conditionné si une petite variation des données entraîne une très grande variation des résultats.

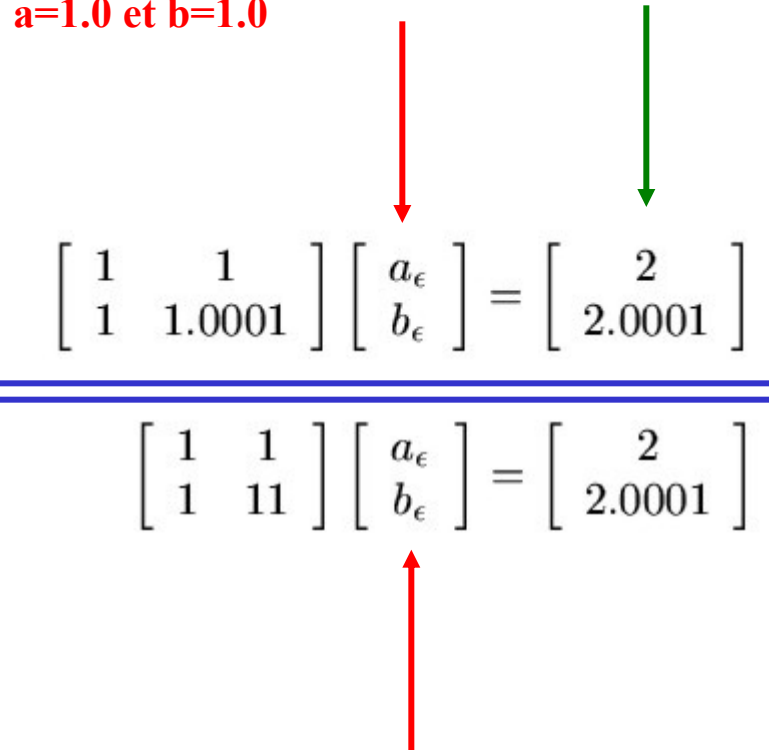
Il s'agit d'une propriété qui est directement liée au système linéaire et qui est donc totalement indépendante de la méthode numérique pour résoudre ce système.

Perturbons
un peu les
deux
systèmes

Système mal conditionné

Solution perturbée :
 $a=1.0$ et $b=1.0$

Données légèrement
perturbées


$$\begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \end{bmatrix} \begin{bmatrix} a_{\epsilon} \\ b_{\epsilon} \end{bmatrix} = \begin{bmatrix} 2 \\ 2.0001 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 11 \end{bmatrix} \begin{bmatrix} a_{\epsilon} \\ b_{\epsilon} \end{bmatrix} = \begin{bmatrix} 2 \\ 2.0001 \end{bmatrix}$$

Solution perturbée :
 $a=1.99999$ et $b=0.00001$

Système bien conditionné

Comment savoir si un système est bien conditionné ?

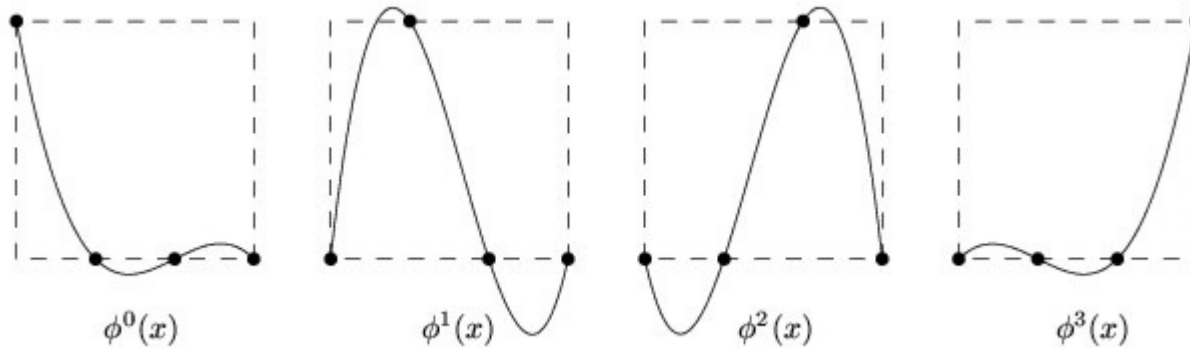
```
A = [[1,1],[1,1.0001]]  
b = [2,2]  
bpert = [2,2.0001]
```

```
x = solve(A,b)  
xpert = solve(A,bpert)
```

```
for i in range(len(x)):  
    print("    x[%d] = %14.7e and xpert[%d] = %14.7e" % (i,x[i],i,xpert[i]))
```

```
lambdas,vect = eig(A)  
print("    condition number      = %12.3e" % cond(A))  
print("    determinant            = %12.3e" % det(A))  
print("    lambda[0]                = %12.3e" % lambdas[0])  
print("    lambda[1]                = %12.3e" % lambdas[1])  
print("    lambda[0] * lambda[1]    = %12.3e" % (lambdas[0]*lambdas[1]))  
print("    lambda[1] / lambda[0]    = %12.3e" % (lambdas[1]/lambdas[0]))
```

```
vi — bash — 59x19  
===== Good condition  
x[0] = 2.0000000e+00 and xpert[0] = 1.9999900e+00  
x[1] = 0.0000000e+00 and xpert[1] = 1.0000000e-05  
condition number      = 1.232e+01  
determinant           = 1.000e+01  
lambda[0]             = 9.010e-01  
lambda[1]             = 1.110e+01  
lambda[0] * lambda[1] = 1.000e+01  
lambda[1] / lambda[0] = 1.232e+01  
===== Bad condition  
x[0] = 2.0000000e+00 and xpert[0] = 1.0000000e+00  
x[1] = 0.0000000e+00 and xpert[1] = 1.0000000e+00  
condition number      = 4.000e+04  
determinant           = 1.000e-04  
lambda[0]             = 5.000e-05  
lambda[1]             = 2.000e+00  
lambda[0] * lambda[1] = 1.000e-04  
lambda[1] / lambda[0] = 4.000e+04  
bash-3.2$
```



$$\phi_i(x) = \frac{(x - X_0)(x - X_1) \dots (x - X_{i-1})(x - X_{i+1}) \dots (x - X_n)}{(X_i - X_0)(X_i - X_1) \dots (X_i - X_{i-1})(X_i - X_{i+1}) \dots (X_i - X_n)}$$

Idée...

$$u^h(x) = \sum_{i=0}^n U_i \phi_i(x)$$

**Les fonctions de
base de Lagrange**

**Choisir les fonctions de base afin que la
matrice du système soit la matrice unité !**

Même solution par unicité de l'interpolation !

Pas de problème de conditionnement !

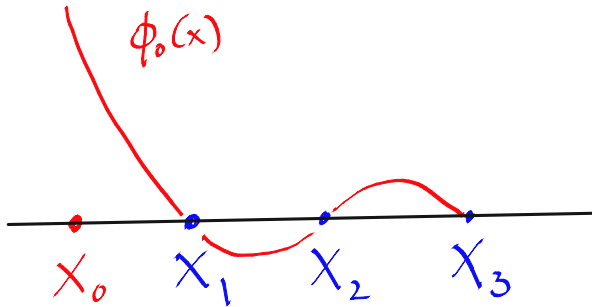
Polynomes de Lagrange

$$\varphi(X_0) = 1$$

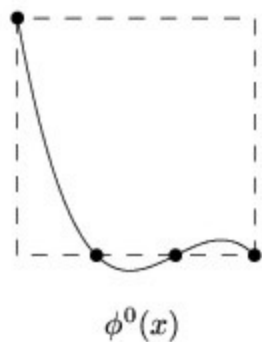
$$\varphi(X_1) = 0$$

$$\varphi(X_2) = 0$$

$$\varphi(X_3) = 0$$



$$\frac{(x - X_1)(x - X_2)(x - X_3)}{(X_0 - X_1)(X_0 - X_2)(X_0 - X_3)} = \phi_0(x)$$



Les fonctions de base de Lagrange

$$\phi_i(x) = \frac{(x - X_0)(x - X_1) \dots (x - X_{i-1})(x - X_{i+1}) \dots (x - X_n)}{(X_i - X_0)(X_i - X_1) \dots (X_i - X_{i-1})(X_i - X_{i+1}) \dots (X_i - X_n)}$$

Erreur d'interpolation

Fonction
inconnue



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



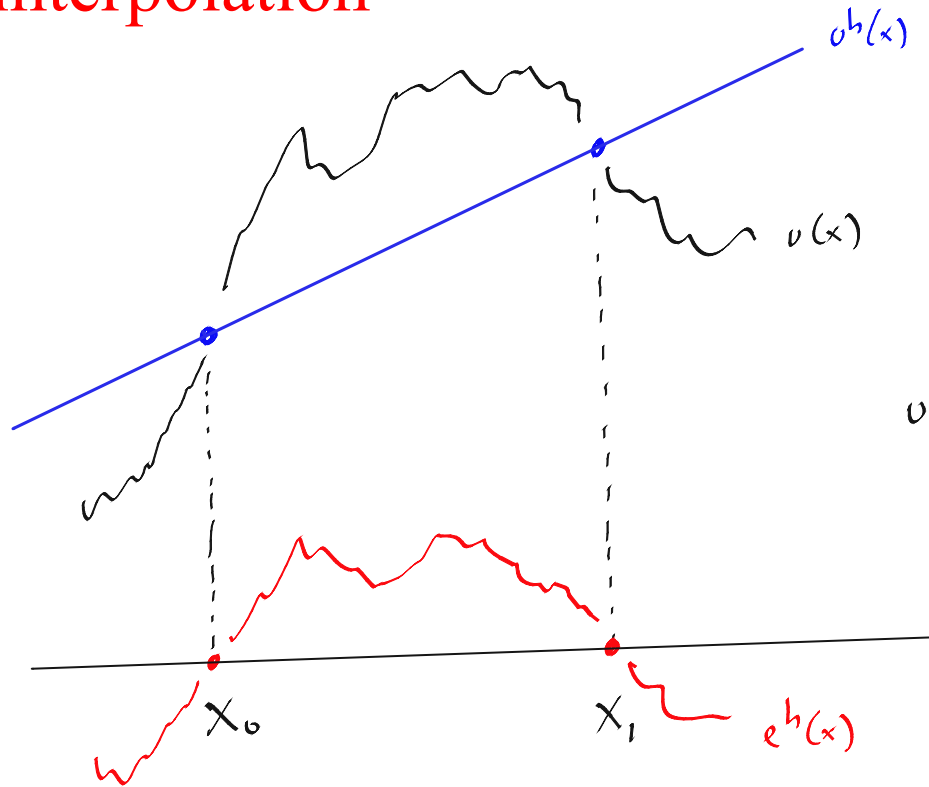
Interpolation
polynomiale

Soit $X_0 < X_1 < \dots < X_n$ les abscisses des points d'interpolation. Si la fonction $u(x)$ est définie sur l'intervalle $[X_0, X_n]$ et qu'elle est $(n+1)$ fois dérivable sur $]X_0, X_n[$, alors pour tout $x \in]X_0, X_n[$, il existe $\xi(x) \in]X_0, X_n[$ tel que :

Théorème 1.1.

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

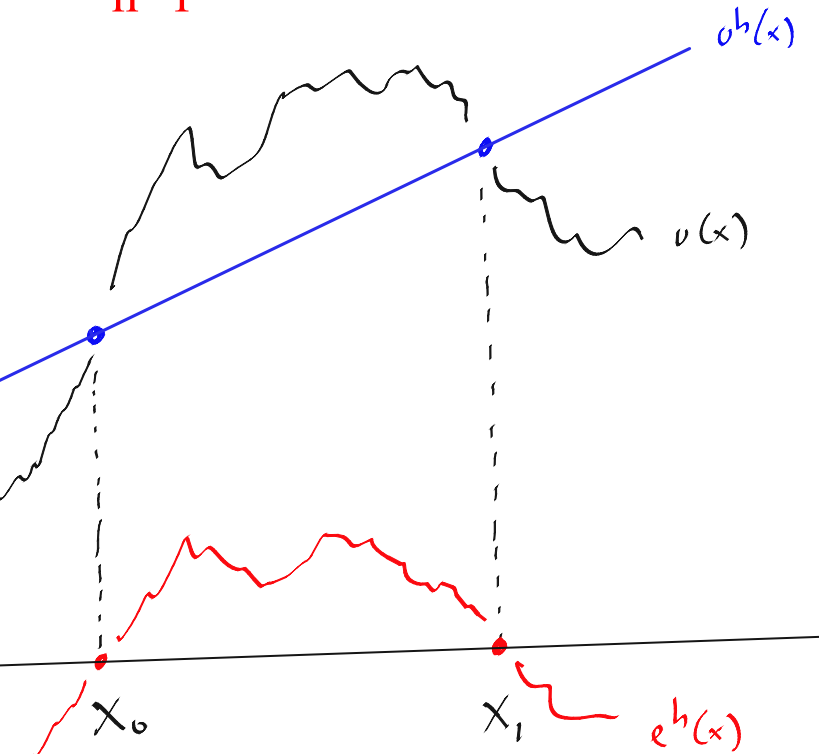
Erreur d'interpolation



$$v(x) - v^h(x) = \frac{v''(\xi(x))}{2!} (x - x_0)(x - x_1)$$

Démonstration

$n=1$

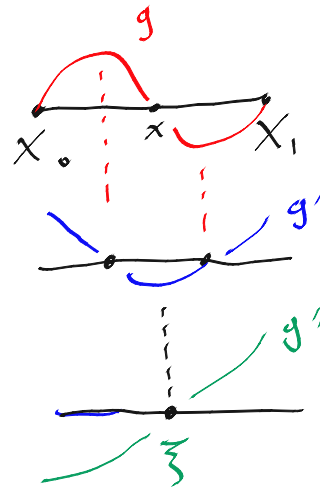


$$g(t) = \underbrace{v(t) - v^h(t)}_{e^h(t)} - e^h(x) \underbrace{\frac{(t - X_0)(t - X_1)}{(x - X_0)(x - X_1)}}_{=1}$$

$$\begin{array}{lcl} t = X_0 & = & 0 \\ t = X_1 & = & 0 \\ t = x & = & e^h(x) \end{array}$$

$$\begin{array}{lcl} & = & 0 \\ & = & 0 \\ & = & 1 \end{array}$$

$$\begin{array}{lcl} & = & 0 \\ & = & 0 \\ & = & e^h(x) \end{array}$$



$\exists \xi(x)$ tel que

$$g''(\xi(x)) = 0$$

$$v''(\xi(x)) - \frac{e^h(x) \cdot 2}{(x - X_0)(x - X_1)} = 0$$

□

$$e^h(x) = f(h)$$

$$e^h(x) = \frac{v''(\xi(x))}{2!} (x - x_0)(x - x_1)$$

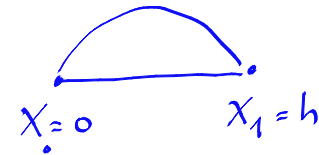
$$|e^h(x)| \leq \underbrace{\left| \frac{v''(\xi(x))}{2!} \right|}_{< \frac{C_2}{2}} \underbrace{|(x - x_0)(x - x_1)|}_{< \frac{h^2}{4}}$$

$$|e^h(x)| \leq \underbrace{\frac{C_2}{2}}_{\mathcal{O}(h^2)} \underbrace{\frac{h^2}{4}}_{\mathcal{O}(h^2)}$$

VUE COMME UNE FONCTION DE h
POUR UN x FIXE

$$\exists C \text{ tel que } \left| \frac{f(h)}{h^2} \right| \leq C$$

AU VOISINAGE DE ZER



$$f(x) = x(x-h)$$

$$f'(x) = 2x - h$$

$$f'(x_m) = 0 \rightarrow x_m = h/2$$

$$f(x_m) = \frac{h^2}{4}$$

And it is done !

The Big O notation :-)

ORDRE = 1	2
2	4
3	8
4	16
5	32

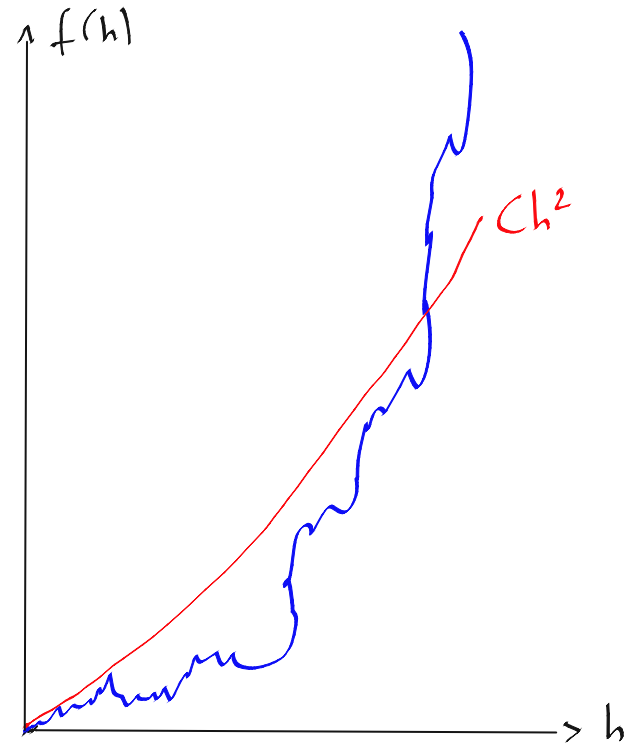
"f(h)"

$$\underbrace{|e^h(x)|}_{\text{}} \leq \underbrace{\frac{C_2}{2}}_{\text{}} \underbrace{\frac{h^2}{4}}_{\mathcal{O}(h^2)}$$

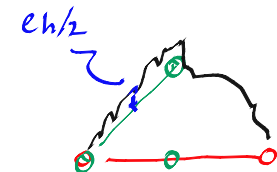
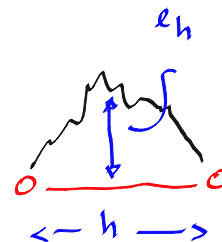
VUE COMME UNE FONCTION DE h
POUR UN x FIXE

$$\exists C \text{ tel que } \left| \frac{f(h)}{h^2} \right| \leq C$$

AU VOISINAGE DE ZER



$$\boxed{C \left[\frac{h}{2} \right]^m}_{\text{ERREUR } h/2} = \frac{1}{2^m} \boxed{C [h]^m}_{\text{ERREUR } h}$$

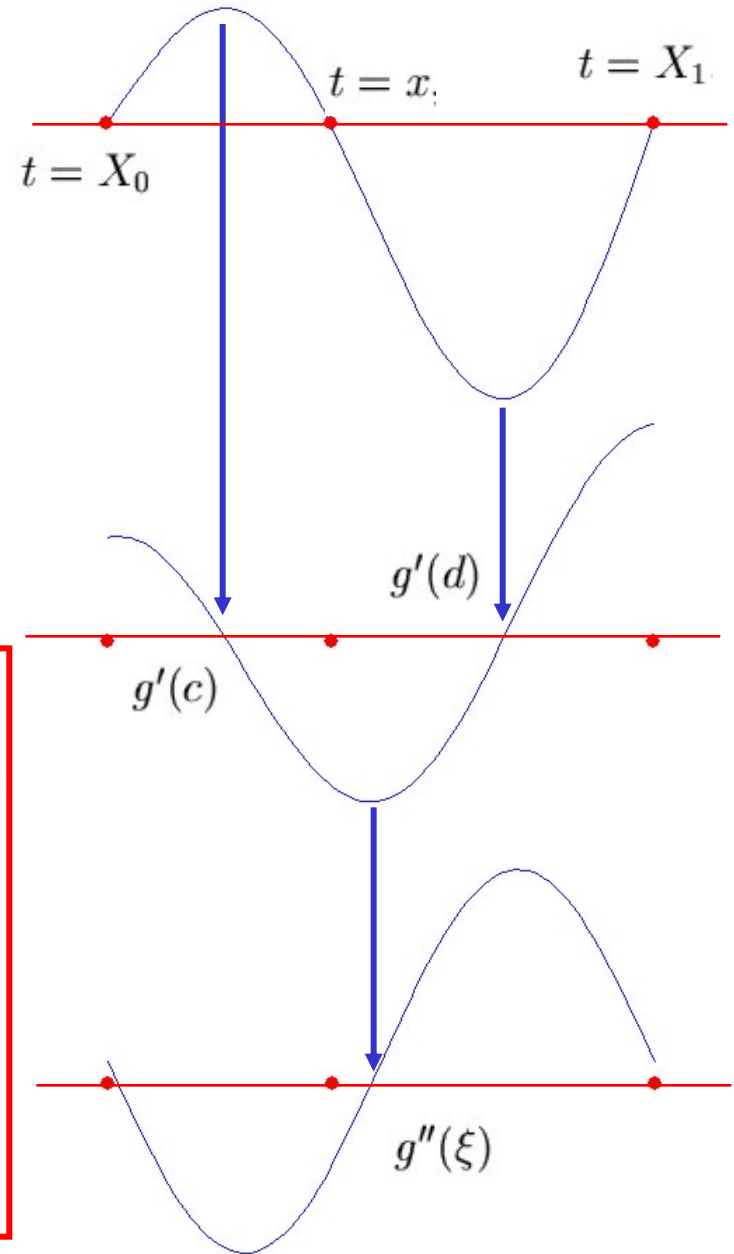


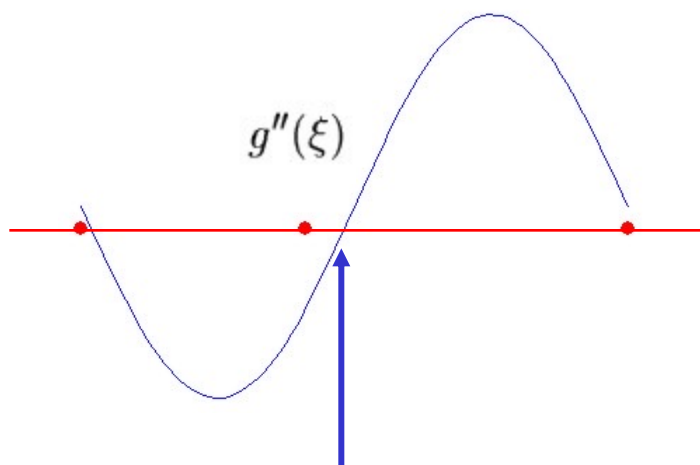
Nous allons démontrer uniquement le cas $n=1$
Ceci est le début de la démonstration :-)

Utilisons le théorème de Rolle...

$$g(t) = u(t) - u^h(t) - e^h(x) \frac{(t - X_0)(t - X_1)}{(x - X_0)(x - X_1)}$$

**On a choisi cette fonction
car elle s'annule en $t = x$, $t = X_0$ et $t = X_1$**





$$\begin{aligned}
 g''(\xi) &= 0 \\
 \downarrow \\
 u''(\xi) - \underbrace{(u^h)''(\xi)}_{=0} - e^h(x) \frac{2}{(x - X_0)(x - X_1)} &= 0 \\
 \downarrow \\
 e^h(x) &= \frac{u''(\xi)}{2} (x - X_0)(x - X_1)
 \end{aligned}$$



Ceci est la fin de la démonstration :-)

$$|e^h(x)| \leq \frac{C_{n+1}}{(n+1)!} \underbrace{|(x-X_0)(x-X_1)\cdots(x-X_n)|}_{\leq n! h^{n+1}/4}$$

En définissant une nouvelle constante $C = \frac{n! C_{n+1}}{4(n+1)!}$

$$|e^h(x)| \leq Ch^{n+1}$$

$$e^h(x) = \mathcal{O}(h^{n+1})$$

Borne d'erreur

On dit qu'une fonction $f(h)$ est un grand ordre de h^m au voisinage de 0, s'il existe une constante C telle que :

$$\left| \frac{f(h)}{h^m} \right| \leq C$$

au voisinage de 0. On écrit alors $f(h) = \mathcal{O}(h^m)$.

Définition 1.1.

Ordre d'une méthode numérique

Définition 1.2.

Une approximation numérique dont le terme d'erreur est $\mathcal{O}(h^m)$ est dite d'ordre m .

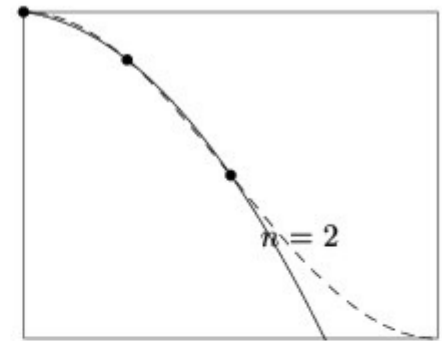
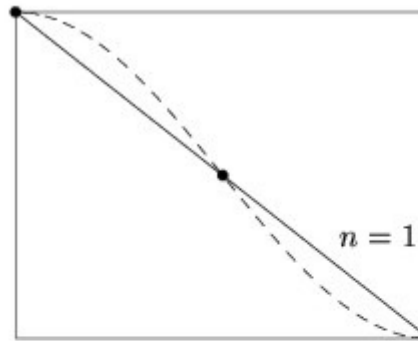
Polynôme de Taylor de degré n est **souvent** d'ordre $n+1$

Interpolation polynomiale par $n+1$ points équidistants est **souvent** d'ordre $n+1$

*Signification intuitive de
l'ordre d'une méthode
numérique*

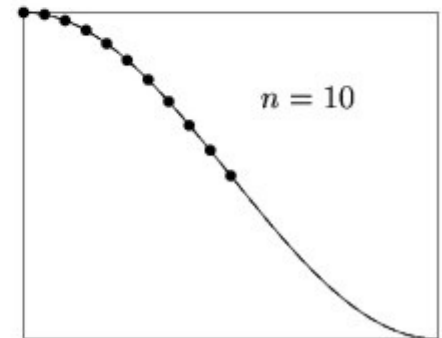
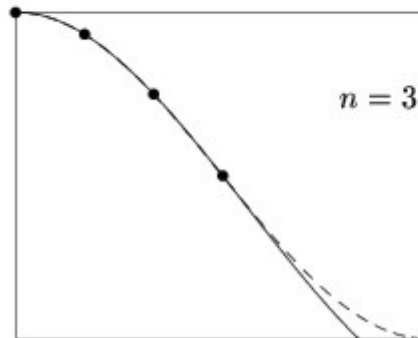
$$C \left(\frac{h}{2} \right)^m = \frac{1}{2^m} C h^m$$

Convergence



Convergence de l'interpolation polynomiale de $\cos(x)$

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

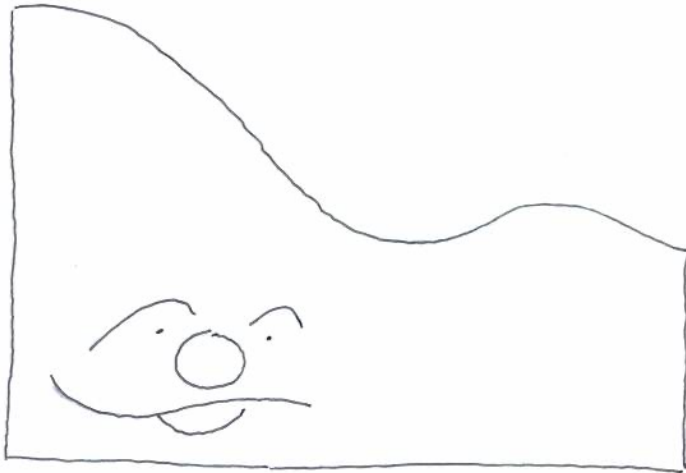


Définition 1.3.

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 :

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

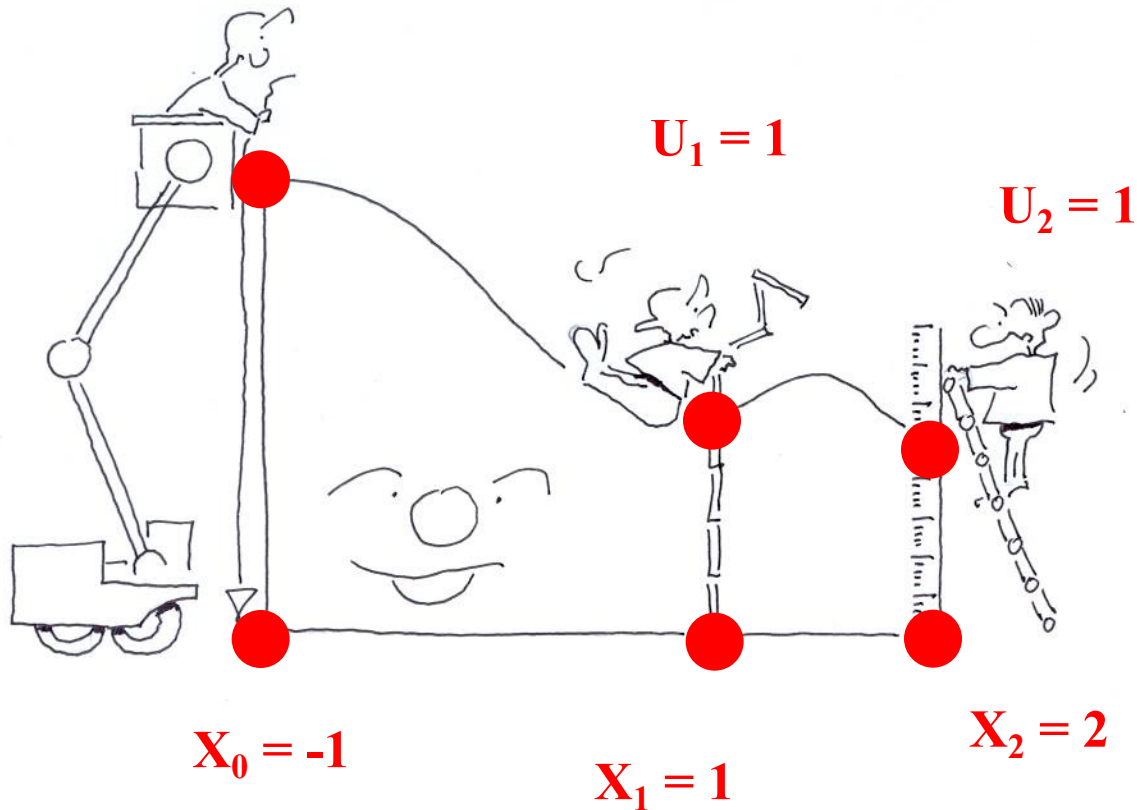
Une courbe....



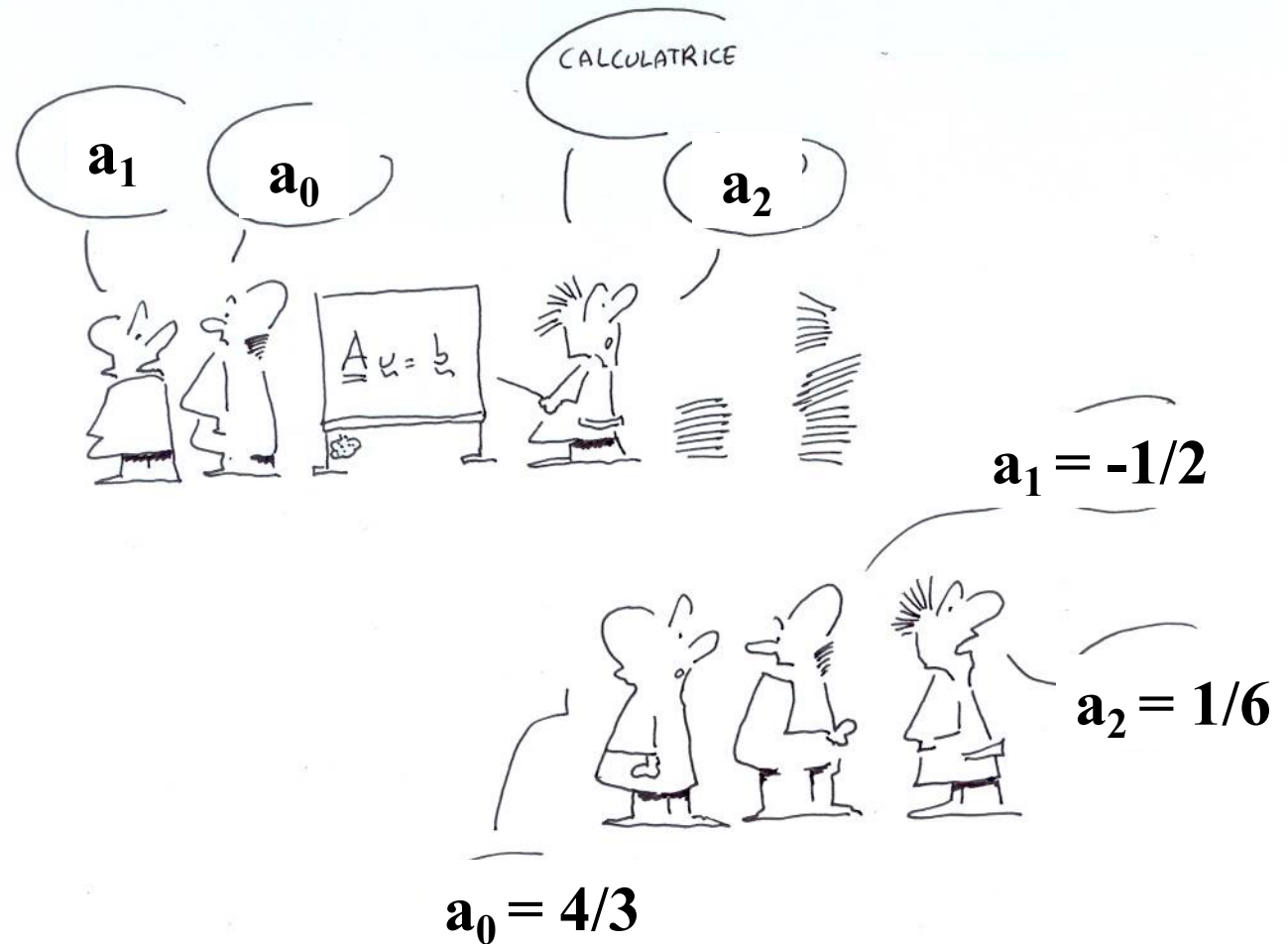
Comment la représenter sur un ordinateur ?

Prise de mesures

$$U_0 = 2$$



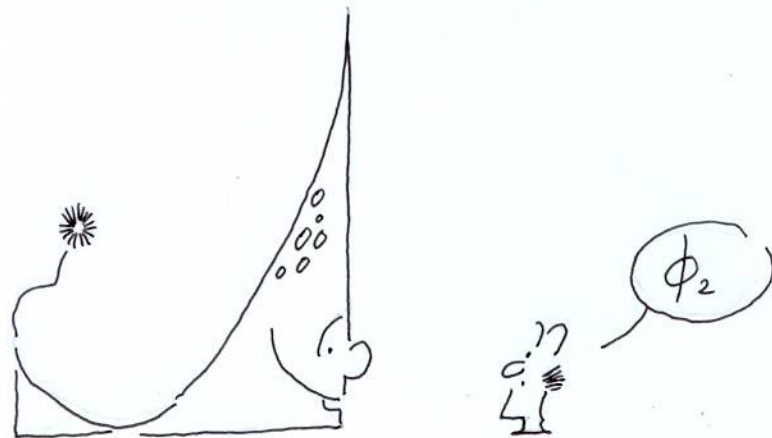
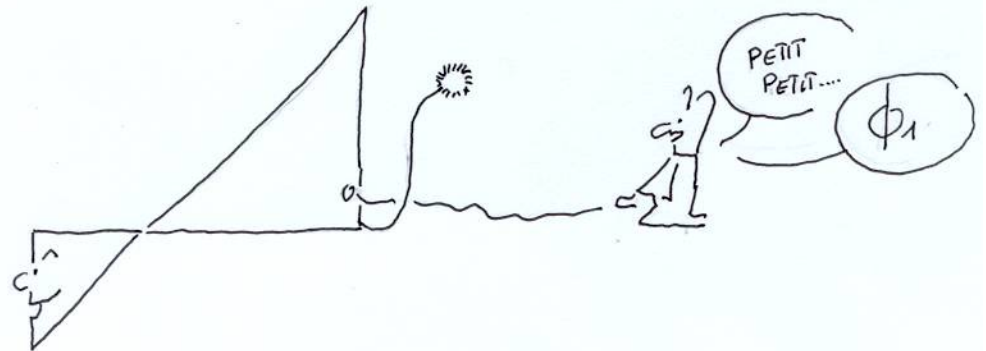
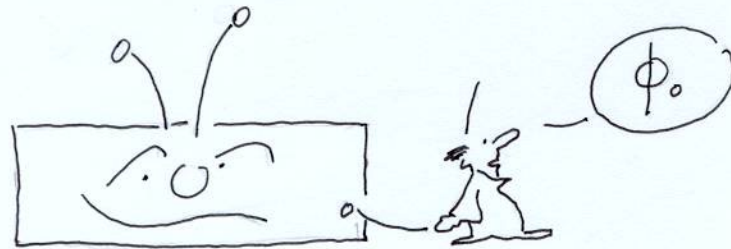
Utilisation des ressources informatiques facultaires



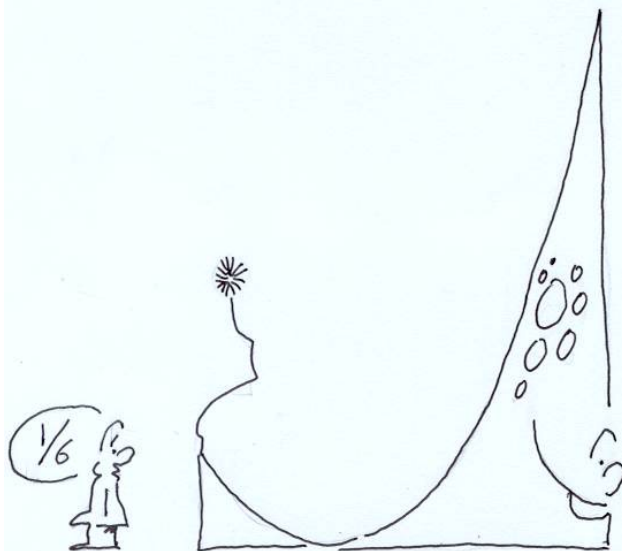
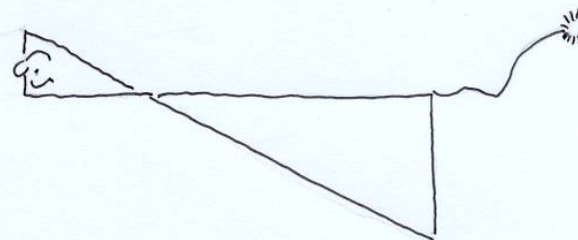
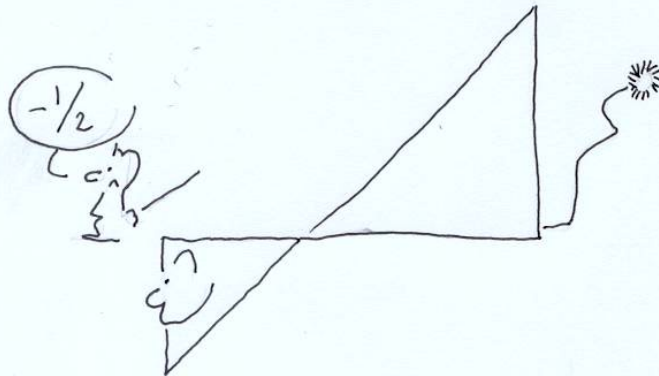
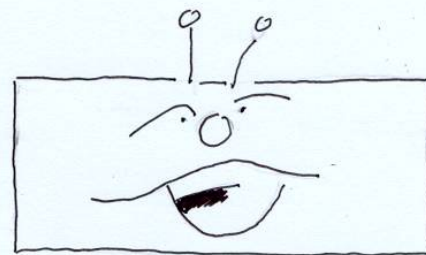
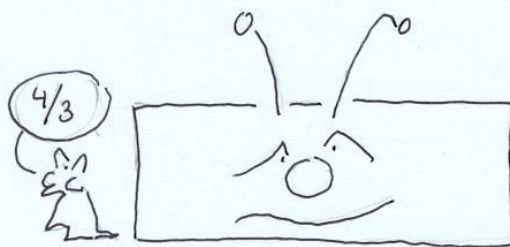
Allons
chercher
quelques
monstres
apprivoisés...

Les 3 fonctions de base d'espace
discret de dimension 3.

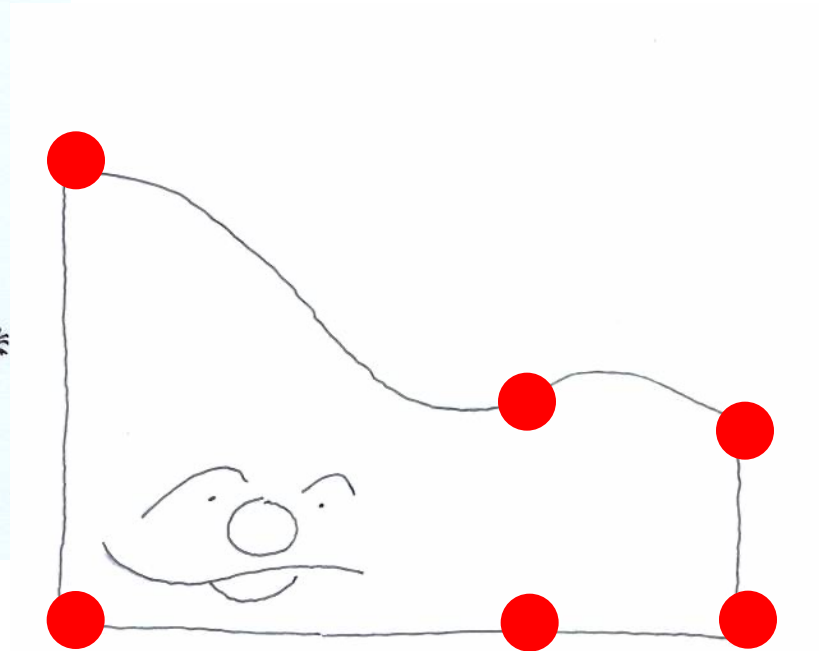
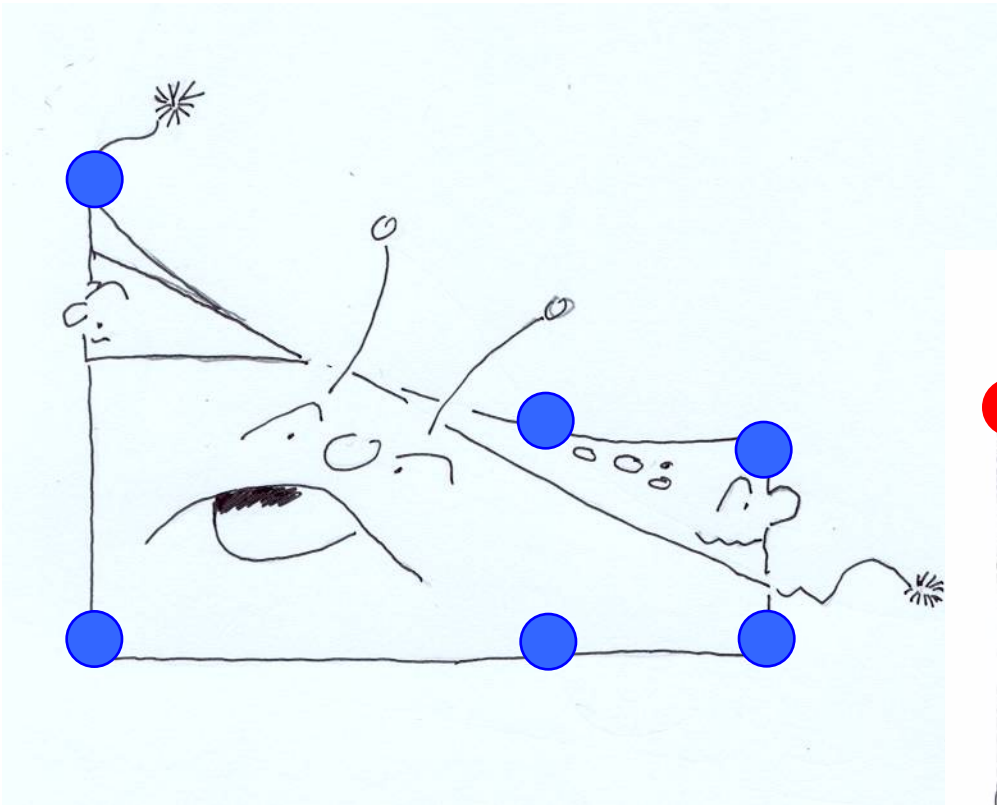
Base d'un espace vectoriel



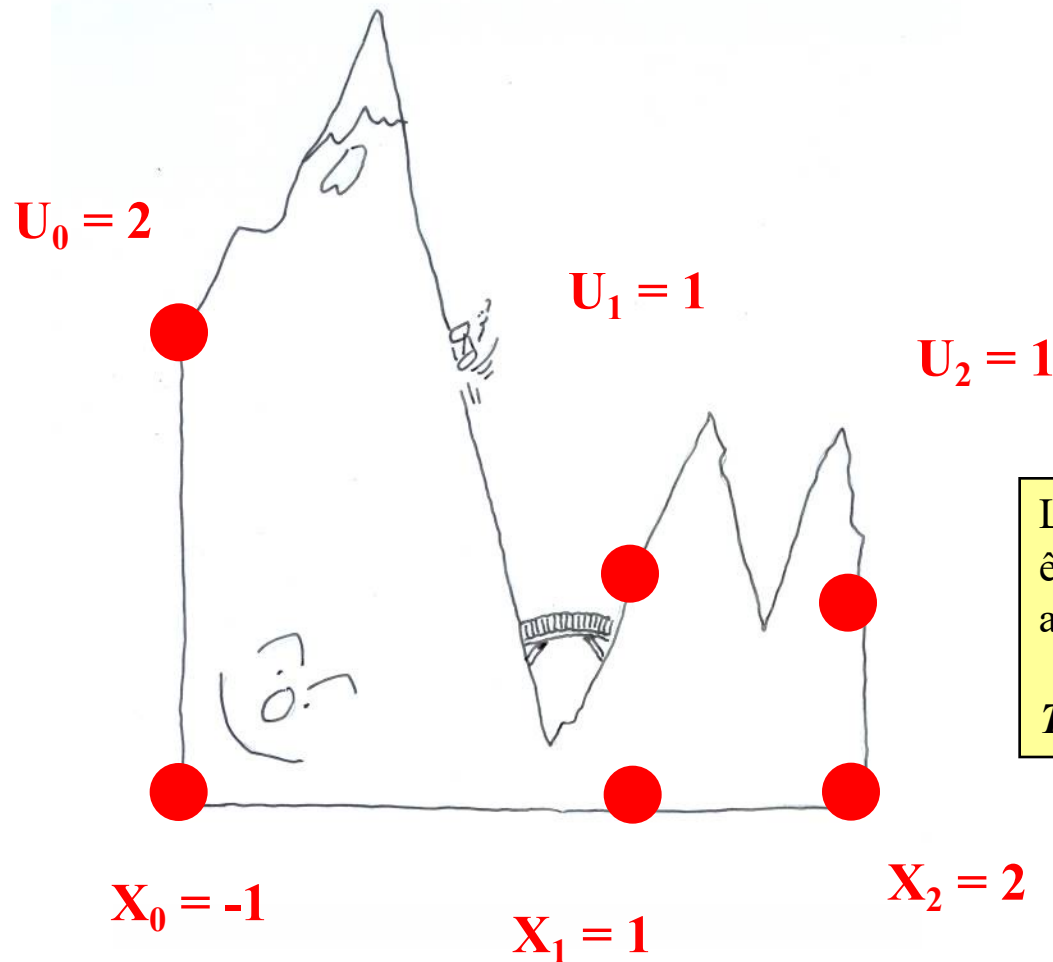
Etes-vous
un bon
dresseur ?



Et la phase critique...



Et si j'avais l'esprit montagnard...



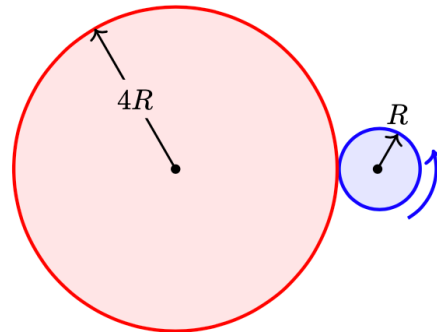
La prise de mesures ne peut pas être réalisée de manière arbitraire...

Théorie de l'échantillonnage

Une question surprenante ?

Q10

Un cylindre de rayon R roule sans glissement sur un cylindre de rayon $4R$.
Le grand cylindre ne bouge pas !
Après avoir parcouru un tour complet du grand cylindre,
le petit cylindre aura effectué n rotations sur lui-même.

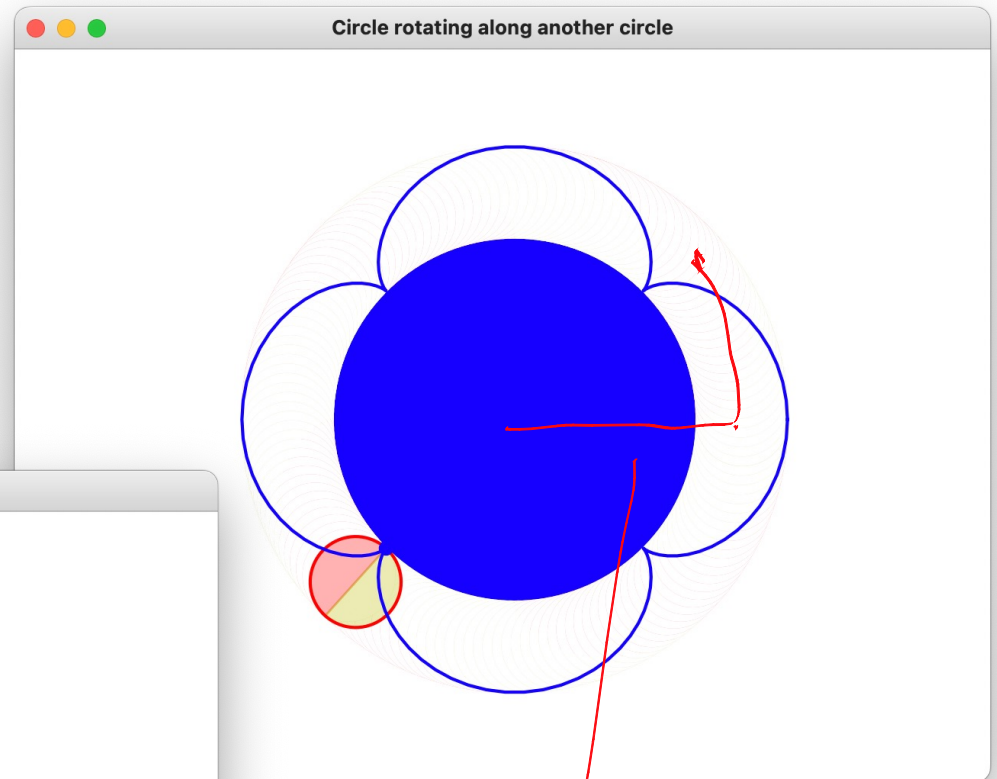
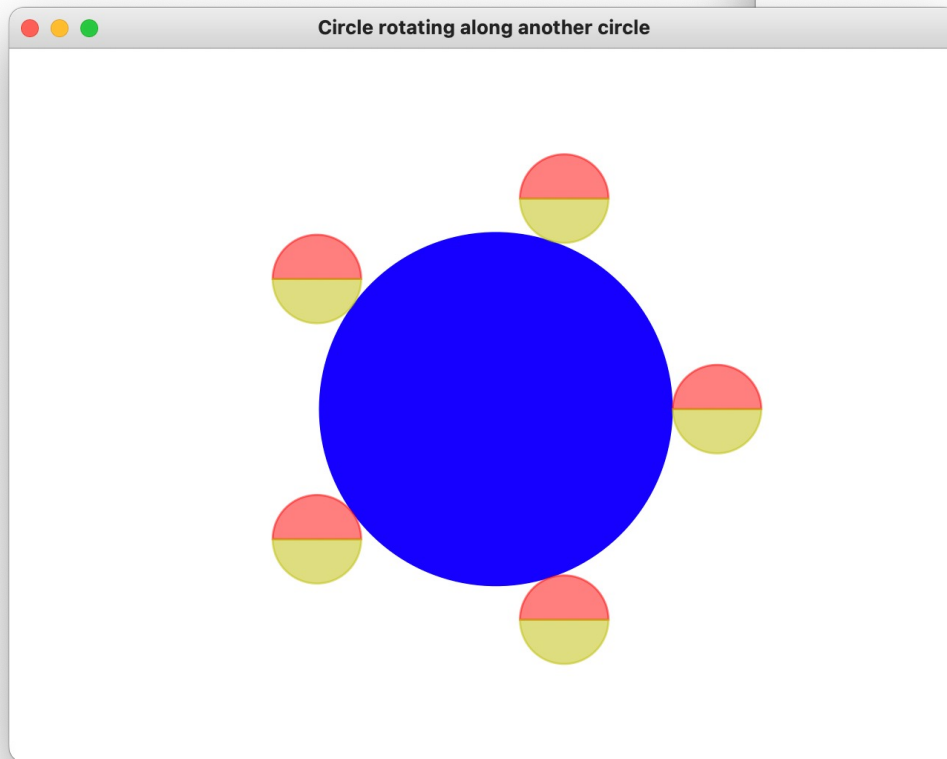


Quel est ce nombre n de rotations ?

- A** $n = 3$
- B** $n = 4$
- C** $n = 5$
- D** $n = 6$
- E** $n = 7$

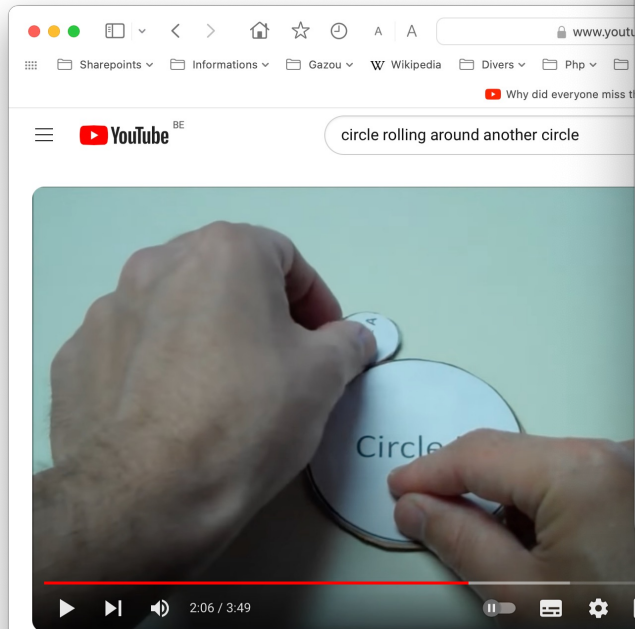
- | | |
|----------|-------------------------------------|
| A | ☐ |
| B | ☐ |
| C | ☒ |
| D | ☐ |
| E | ☐ |

Le premier devoir !



SR :-)

Le premier devoir !



Why did everyone miss this SAT Math question?



MindYourDecisions ✓
3,04 M d'abonnés

S'abonner

44 k



Partager



3 M de vues · il y a 8 ans Interview Riddles And Puzzles

Problem: Circle A has $\frac{1}{3}$ the radius of circle B, and circle A rolls one trip around circle B. How many times will circle A revolve in total?

...afficher plus

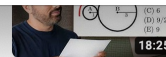
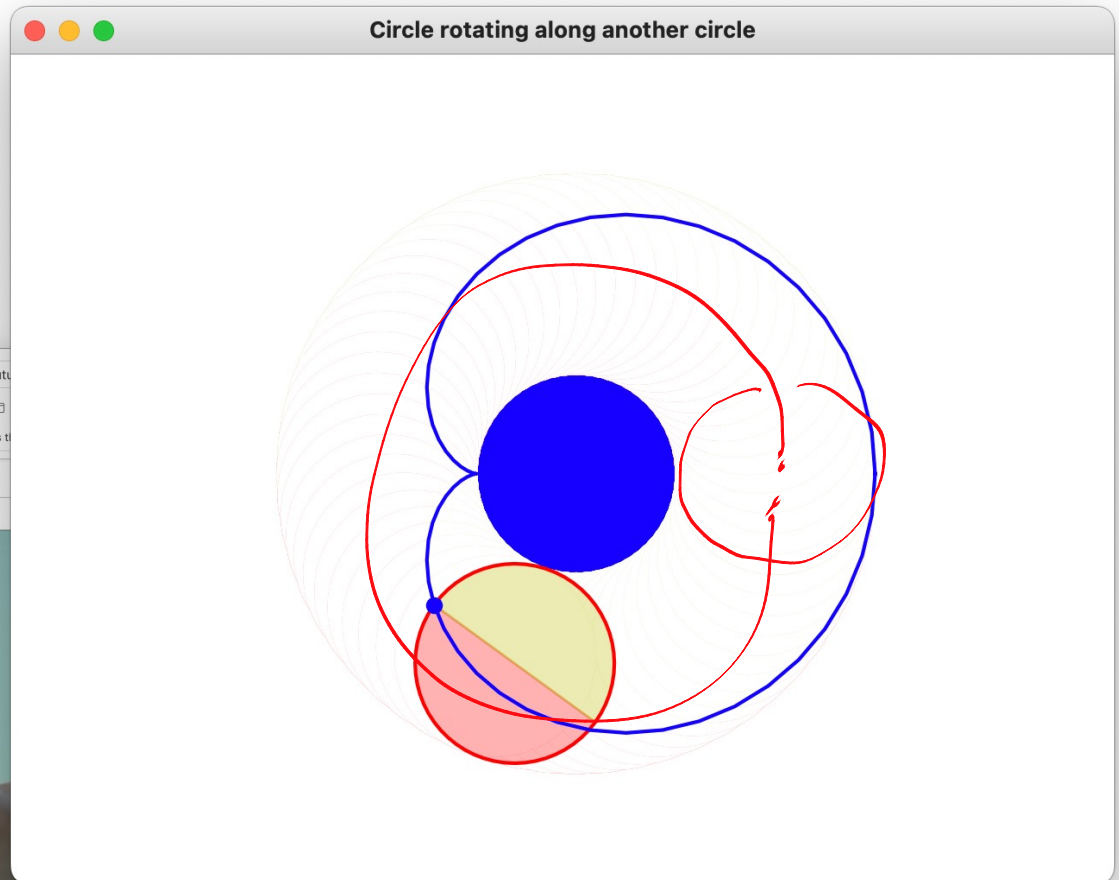
Effectuer des achats dans le magasin MindYourDecisions



Mind Your Decisions Appa...
25,15 €



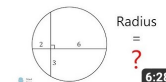
Mind Your Decisions Appa...
27,94 €



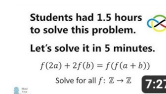
Veritasium ✓
8,4 M de vues · il y a 2 mois



The coin rotation paradox
mathAHA
27 k vues · il y a 7 ans



How To Solve For The Radius. Challenging 1970s Math...
MindYourDecisions ✓
2,4 M de vues · il y a 4 ans



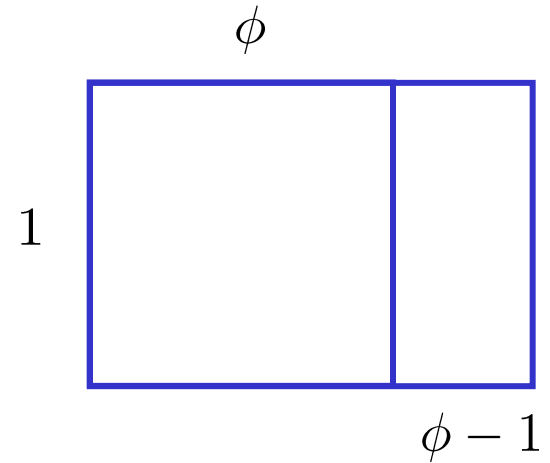
Solving An Insanely Hard Problem For High School...
MindYourDecisions ✓
3,3 M de vues · il y a 4 ans



What Positive Number Doubles When The Last Digit Moves To...

The Golden Ratio

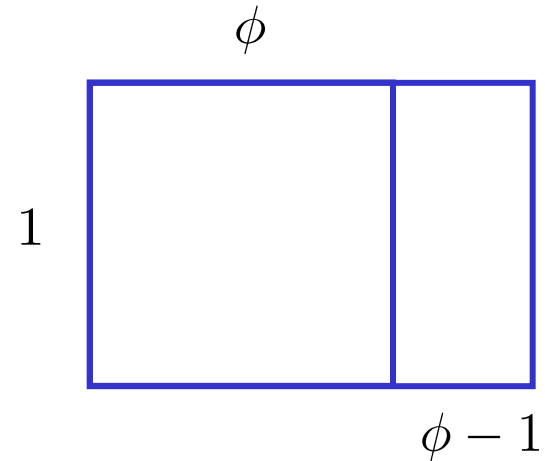
```
from math import sqrt  
  
phi = (1.0 + sqrt(5.0)) / 2.0  
print(phi)
```



The Golden Ratio

```
from math import sqrt

phi = (1.0 + sqrt(5.0)) / 2.0
print(phi)
```



$$\frac{1}{\phi} = \frac{\phi - 1}{1}$$

↓

$$\phi^2 - \phi - 1 = 0$$

```
bash-3.2$ python
Python 3.6.3 |Anaconda custom (64-bit)| (default, Oct 6 2017, 12:04:38)
[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from math import sqrt
>>> phi = (1.0 + sqrt(5.0)) / 2.0
>>> print(phi)
1.618033988749895
>>> 
```

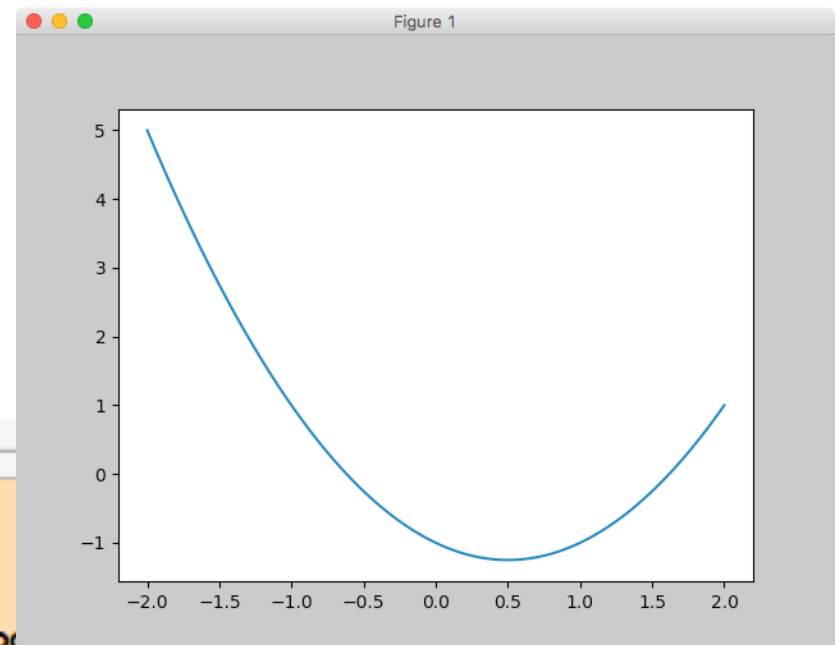
Polynômes avec Python

$$\phi^2 - \phi - 1 = 0$$

$$\phi = \frac{1 \pm \sqrt{5}}{2}$$

```
p = [1,-1,-1]
r = roots(p)
print(r)
```

```
x = linspace(-2,2,100)
plt.plot(x,polyval(p,x))
plt.show()
```



```
bash-3.2$ python goldenratio.py
-0- Golden ratio number
1.618033988749895
-1- Golden ratio number as roots of p
[ 1.61803399 -0.61803399]
```

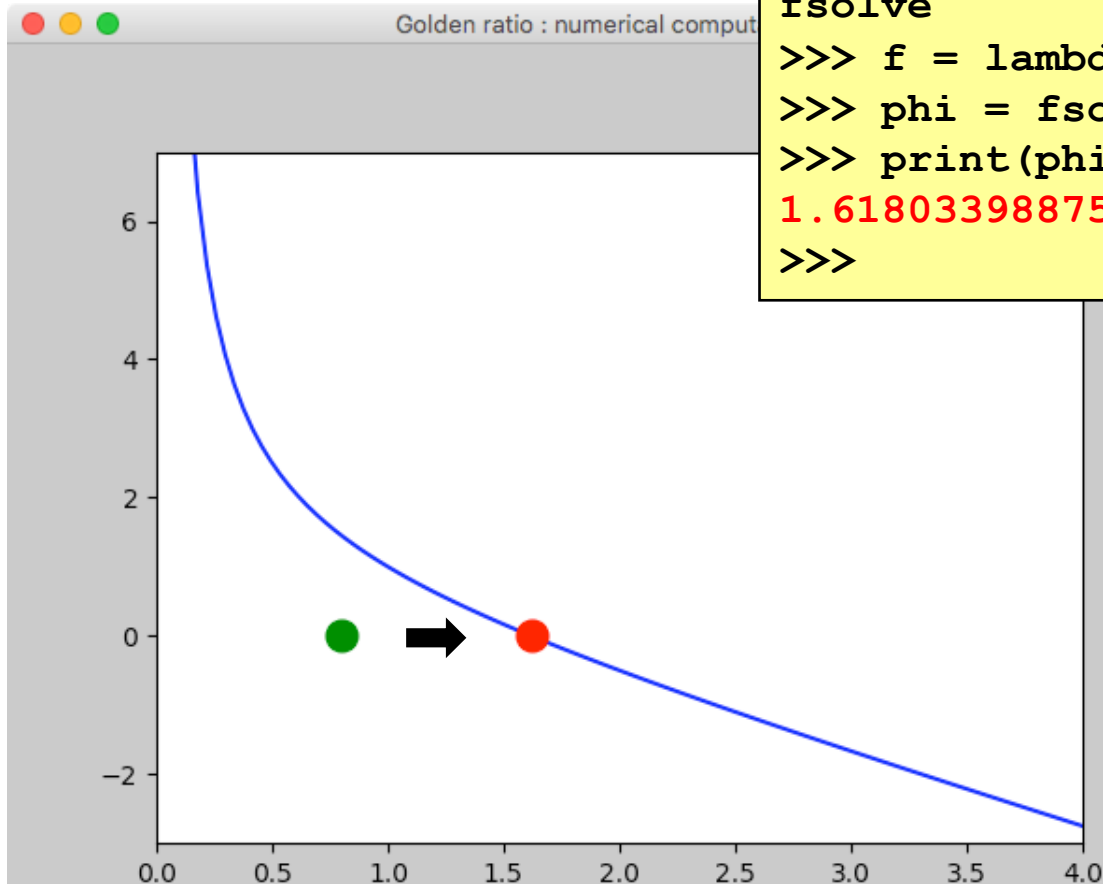
Calcul symbolique : sympy

$$\phi^2 - \phi - 1 = 0$$

$$\phi = \frac{1 \pm \sqrt{5}}{2}$$

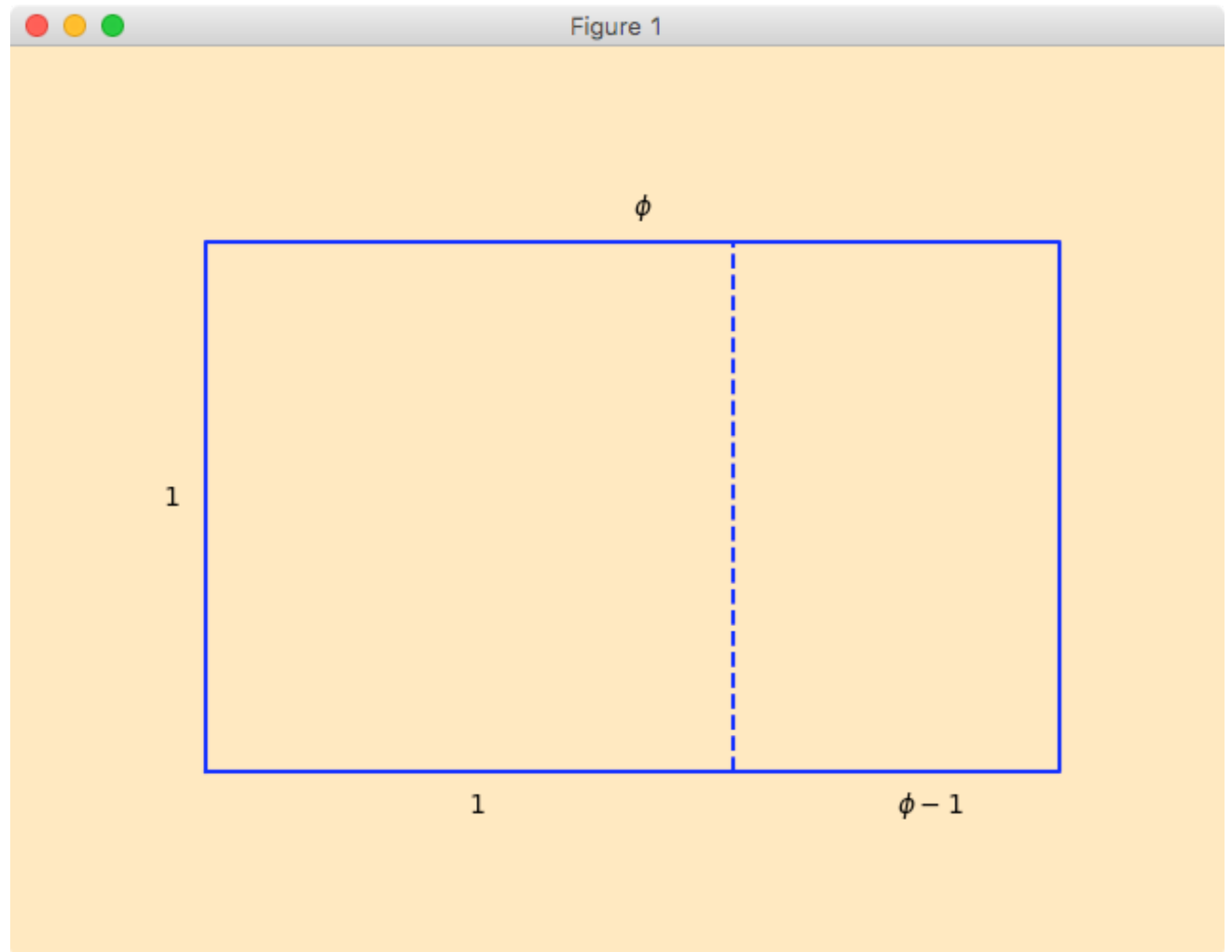
```
>>> from sympy import symbols,solve,evalf
>>> x = symbols('x')
>>> r = solve(1/x-x+1,x)
>>> print(r)
[1/2 + sqrt(5)/2, -sqrt(5)/2 + 1/2]
>>> phi = r[0]
>>> print(phi)
1/2 + sqrt(5)/2
>>> print(phi.evalf(50))
1.6180339887498948482045868343656381177203091798058
>>> print(phi.evalf())
1.61803398874989
```

Résolution numérique : scipy

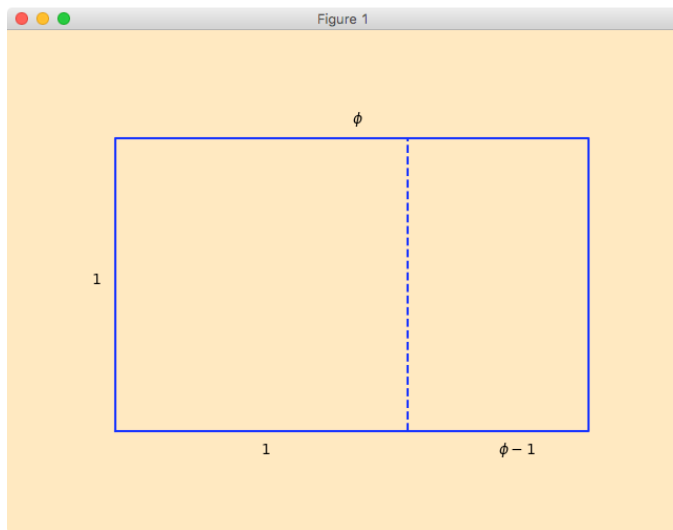


```
>>> from scipy.optimize import  
fsolve  
>>> f = lambda x : 1/x - (x-1)  
>>> phi = fsolve(f,0.8)  
>>> print(phi[0])  
1.61803398875  
>>>
```

Faire des jolis dessins...



py-files



```
def goldRect():
```

```
    """ plots the golden rectangle """
```

```
    phi = (1 + sqrt(5.0)) / 2
```

```
    x = [0,phi,phi,0,0]
```

```
    y = [0,0,1,1,0]
```

```
    u = [1,1]
```

```
    v = [0,1]
```

LaTeX commands

```
    plt.axis('equal')
```

```
    plt.axis('off')
```

```
    plt.text(phi/2,1.1,' $\phi$ ')
```

```
    plt.text((1+phi)/2,-0.1,' $\phi - 1$ ')
```

```
    plt.text(-0.1, 0.5,'1')
```

```
    plt.text( 0.5,-0.1,'1')
```

```
    plt.plot(x,y,'-b')
```

```
    plt.plot(u,v,'--b')
```

```
    plt.show()
```

```
>>> from goldenRatioRectangle import goldRect
```

```
>>> goldRect()
```

```
>>> help(goldRect)
```

Fractions continues...

$$\begin{array}{c} \frac{1}{\phi} = \frac{\phi - 1}{1} \\ \downarrow \\ 1 + \frac{1}{\phi} = \phi \\ \downarrow \\ 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{\dots}}} = \phi \end{array}$$

```
>>> from math import sqrt
>>> n = 6
>>> p = '1'
>>> for k in range(n):
...     p = '1+1/(' + p + ')'
...
>>> print(p)
1+1/(1+1/(1+1/(1+1/(1+1/(1+1/(1))))))
>>> phi = eval(p)
>>> print(phi)
1.6153846153846154
>>> err = (1+sqrt(5))/2 - phi
>>> print(err)
0.0026493733652794837
```

La variable **p** est une chaîne de caractères générée en commençant par un seul '1' et en encadrant à plusieurs reprises cette chaîne avec '1+1/(' à l'avant et ')' à l'arrière. Quelle que soit la longueur de cette chaîne, il s'agit d'une expression Python valide.

Une autre implémentation...

$$\begin{array}{ccc} \frac{1}{\phi} & = & \frac{\phi - 1}{1} \\ \downarrow & & \\ 1 + \frac{1}{\phi} & = & \phi \\ \downarrow & & \\ \frac{\phi_n + 1}{\phi_n} & = & \phi_{n+1} \end{array}$$

```
>>> from math import sqrt
>>> n = 6
>>> p = 1
>>> q = 1
>>> for k in range(n):
...     s = p
...     p = p + q
...     q = s
...
>>> phi = eval('%d/%d' % (p,q))
>>> print(phi)
1.6153846153846154
>>> err = (1 + sqrt(5))/2 - phi
>>> print(err)
0.0026493733652794837
```