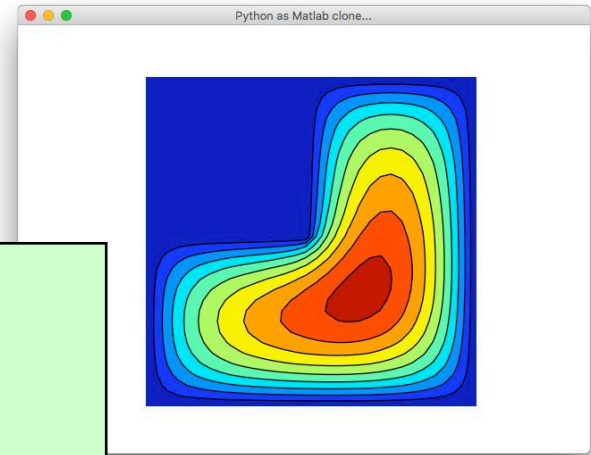


Plan du cours de méthodes numériques

Comment résoudre
numériquement un
problème aux
valeurs initiales ?



Comment interpoler
une fonction ?

Comment dériver
numériquement
une fonction ?

Comment approximer
une fonction ?

Comment résoudre
numériquement un
problème aux
conditions frontières ?

Comment intégrer
numériquement
une fonction ?

Et les équations non-
linéaires ?

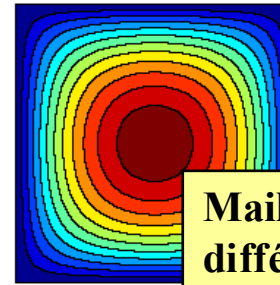
*Comment résoudre numériquement
une équation différentielle ordinaire ?*

Et les méthodes itératives ?

*Comment résoudre numériquement
une équation aux dérivées partielles ?*

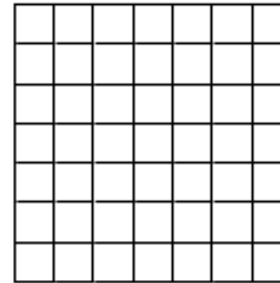
Comment résoudre
numériquement une
équation aux dérivées
partielles ?

A quoi servent les méthodes numériques ?

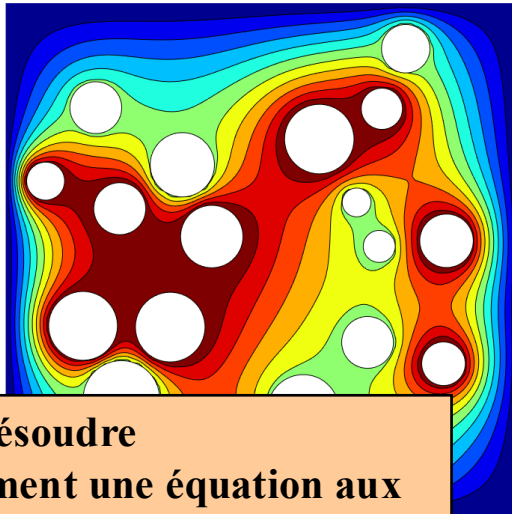
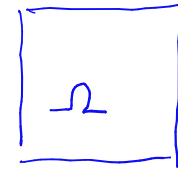


LEPL1104

Maillages structurés:
différences finies



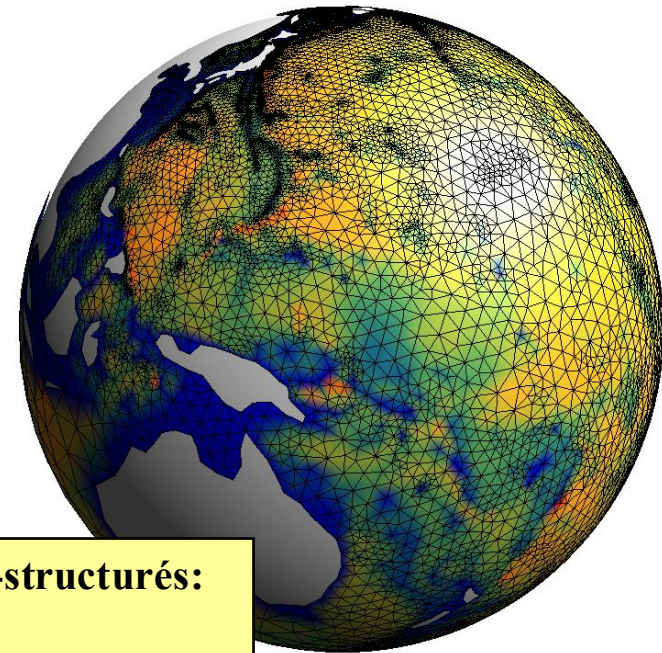
$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f = 0$$



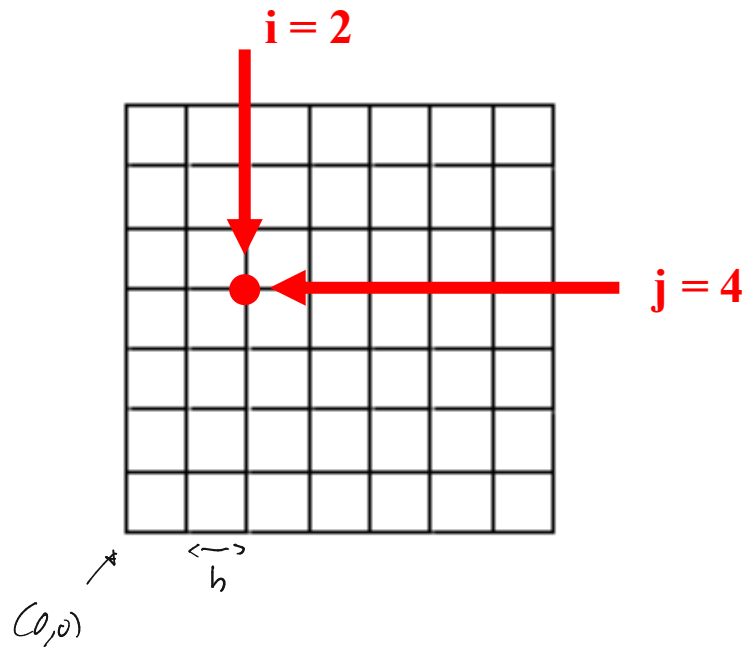
Comment résoudre numériquement une équation aux dérivées partielles avec des conditions aux limites ?

LEPL1110

Maillages non-structurés:
éléments finis

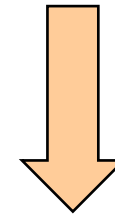


Grille / Maillage

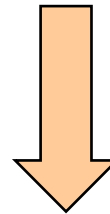


$$X_i = a + ih,$$

$i = 0, \dots, m$



$$\mathbf{X}_{ij} = (X_i, Y_j) = (ih, jh)$$

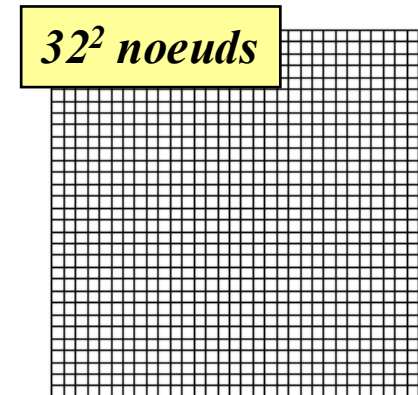
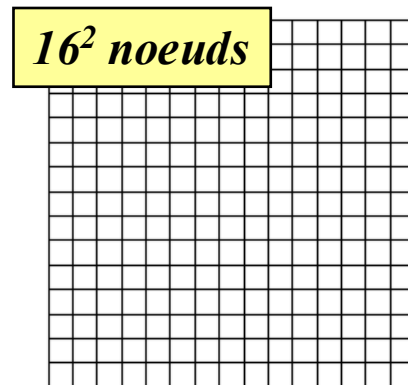
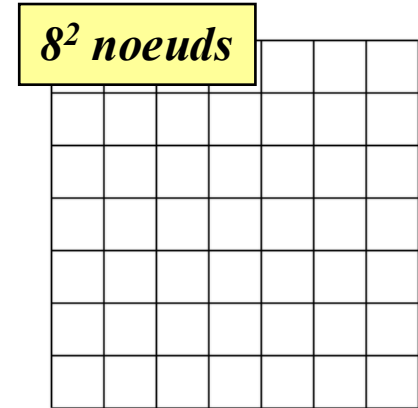
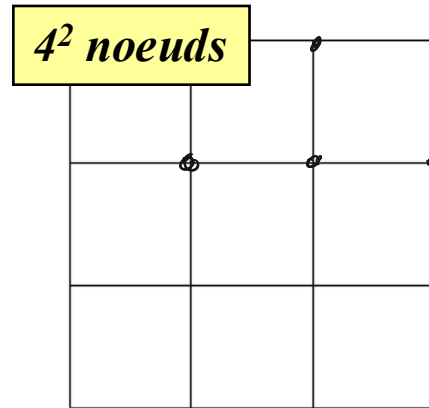
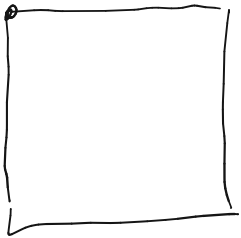


$$U_{ij} = u^h(\mathbf{X}_{ij}) \approx u(\mathbf{X}_{ij})$$

$$h = (b - a)/(m)$$

Méthode des différences finies

$$U_{ij} = u(x_i, y_j)$$



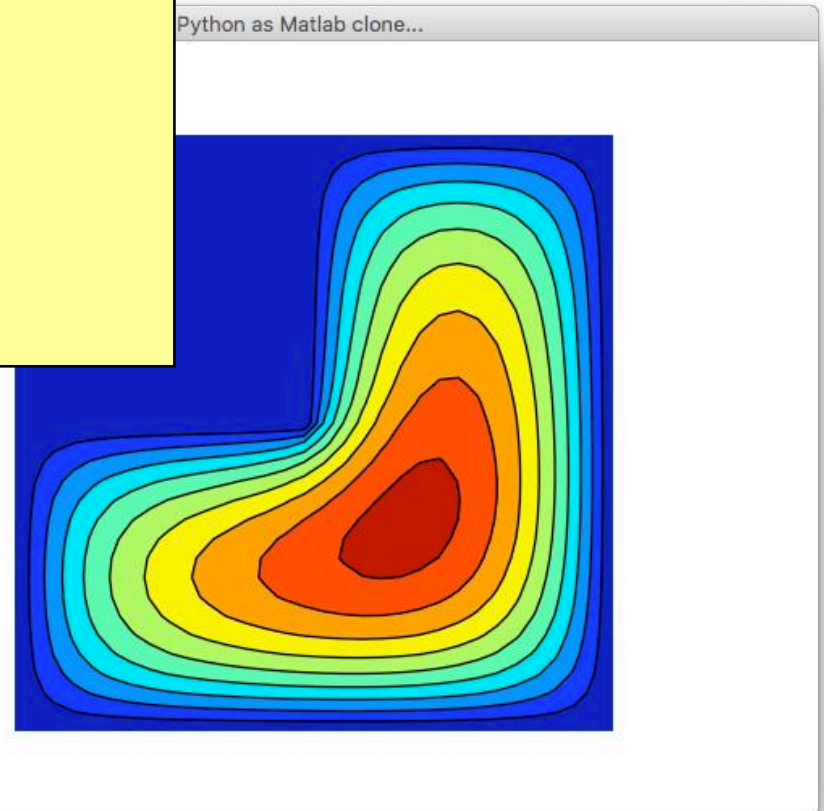
Et finalement...

```
map = numgrid(32)
index = map[map>0]

A = delsq(map)
B = ones(size(index))
U = zeros(shape(map))
U[map>0] = spsolve(A,B)[index-1]

plt.contourf(U,10)
```

*Malheureusement,
les fonctions numgrid et delsq
de MATLAB ne font pas encore
partie de numpy et de scipy !*



$$\rho c \frac{\partial u}{\partial t} = k \frac{\partial^2 u}{\partial x^2}$$

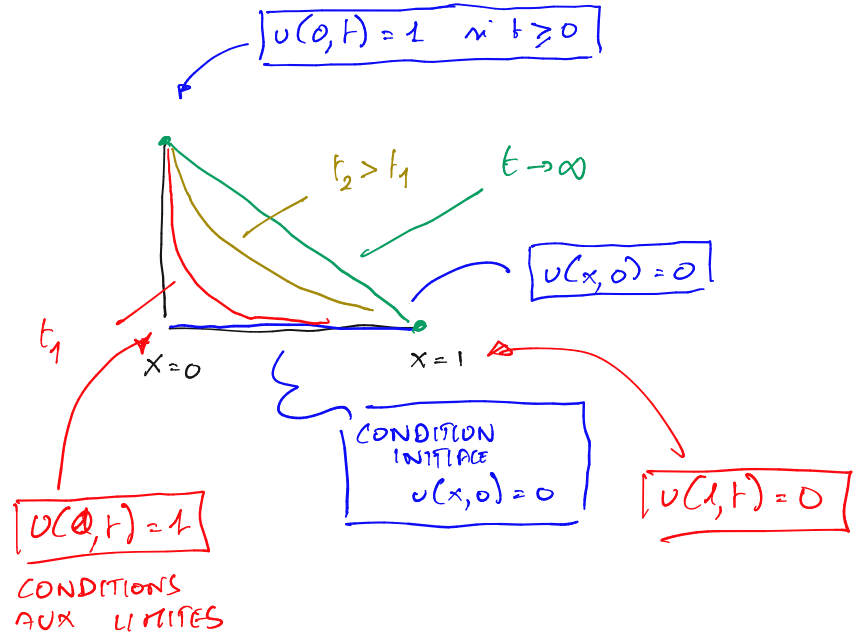
$$\frac{\partial u}{\partial t} = \underbrace{\frac{k}{\rho c}}_{\alpha} \frac{\partial^2 u}{\partial x^2}$$

DIFFUSIVITE THERMIQUE

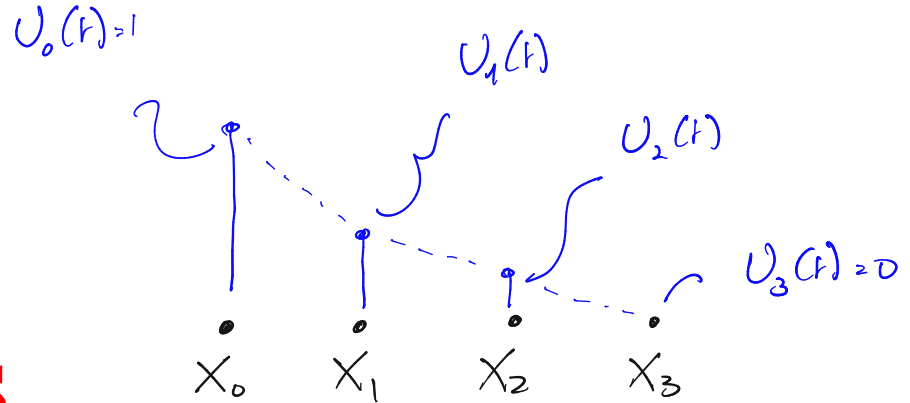
$\left[\frac{S}{s} \right]$

$\left[\frac{m^2}{s} \right]$

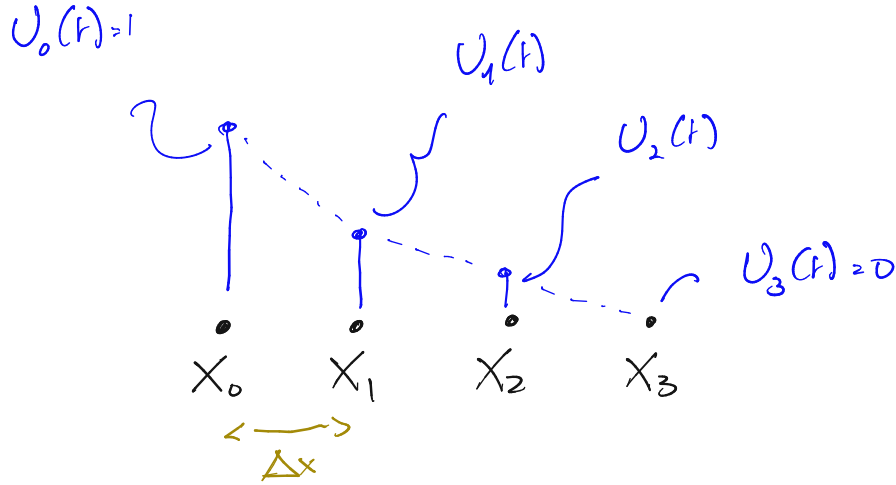
$\left[\frac{S}{m^2} \right]$



Et
 faisons
 varier le temps



Exemple



$$\left. \begin{aligned} \frac{dU_0}{dt} &= 0 \\ \frac{dU_1}{dt} &= \alpha \frac{U_0 + U_2 - 2U_1}{(\Delta x)^2} \end{aligned} \right\}$$

$$\left. \begin{aligned} \frac{dU_2}{dt} &= \alpha \frac{U_1 + U_3 - 2U_2}{(\Delta x)^2} \\ \frac{dU_3}{dt} &= 0 \end{aligned} \right\}$$

$$\underline{u}'(t) = \frac{\alpha}{(\Delta x)^2} \underline{A} \cdot \underline{u}(t)$$

$$\underbrace{\frac{d}{dt} \begin{bmatrix} U_0(t) \\ U_1(t) \\ U_2(t) \\ U_3(t) \end{bmatrix}}_{\underline{u}'(t)} = \frac{\alpha}{(\Delta x)^2} \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\underline{A}} \underbrace{\begin{bmatrix} U_0(t) \\ U_1(t) \\ U_2(t) \\ U_3(t) \end{bmatrix}}_{\underline{u}(t)}$$

EULER EXPLICITE

$\underline{f}(\underline{u}^m)$

$$\underline{u}^{m+1} - \underline{u}^m = \underbrace{\frac{\alpha \Delta t}{(\Delta x)^2}}_{\underline{B}} \underline{A} \cdot \underline{u}^m$$

Différences finies (espace) Euler explicite (temps)

En définissant $\alpha = k/(\rho c)$

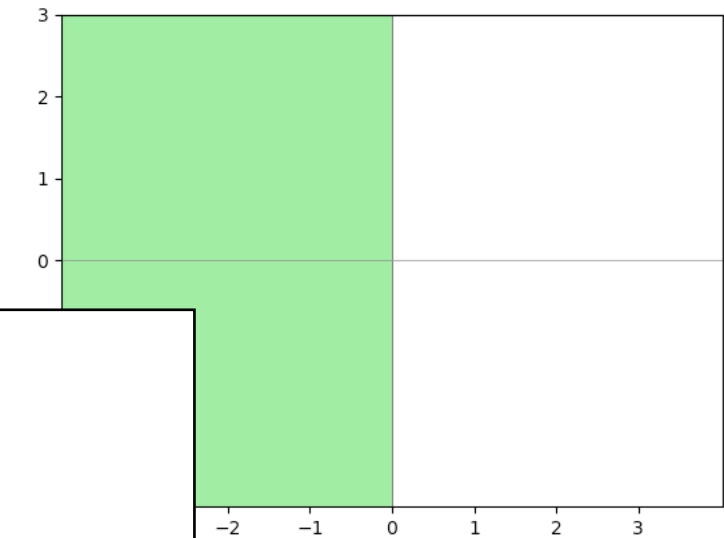
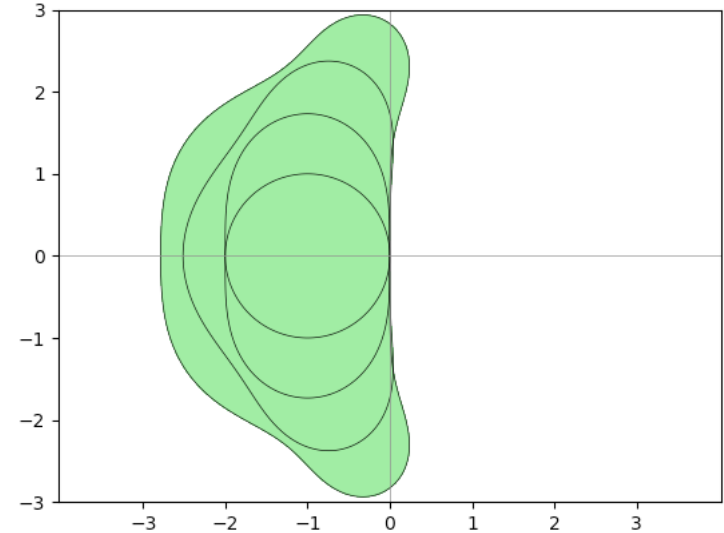
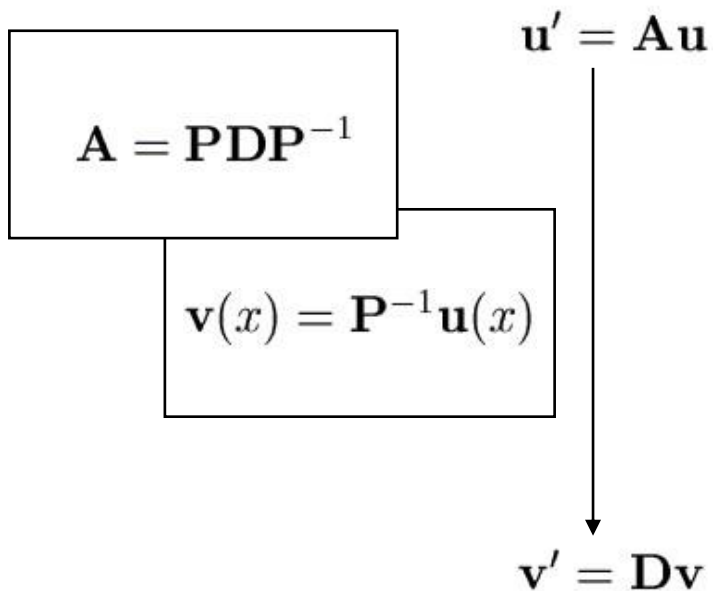
$$\left(\frac{U_i^{n+1} - U_i^n}{\Delta t}\right) = \alpha \left(\frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{(\Delta x)^2}\right)$$

En définissant $\beta = \frac{\alpha \Delta t}{(\Delta x)^2}$,

$$U_i^{n+1} = U_i^n + \beta (U_{i+1}^n + U_{i-1}^n - 2U_i^n)$$

**C'est une itération pour un vecteur qui doit converger vers la solution de régime
C'est quelque chose qu'on a déjà rencontré...**

Stabilité des systèmes

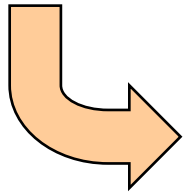


Le problème différentiel initial est équivalent à n équations scalaires

$$v'_i(x) = \lambda_i v_i(x), \quad i = 1, \dots, n$$

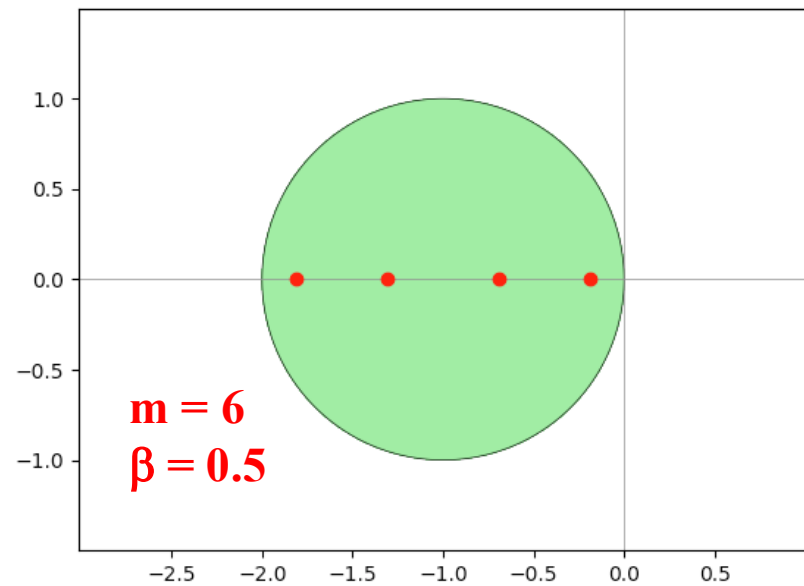
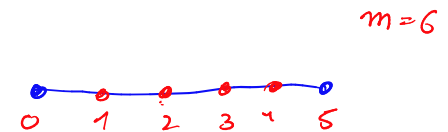
$$\mathbf{u}_{n+1} = \mathbf{u}_n + \beta \mathbf{A} \mathbf{u}_n$$

Euler explicite



$$|1 + \beta \lambda_i| < 1$$

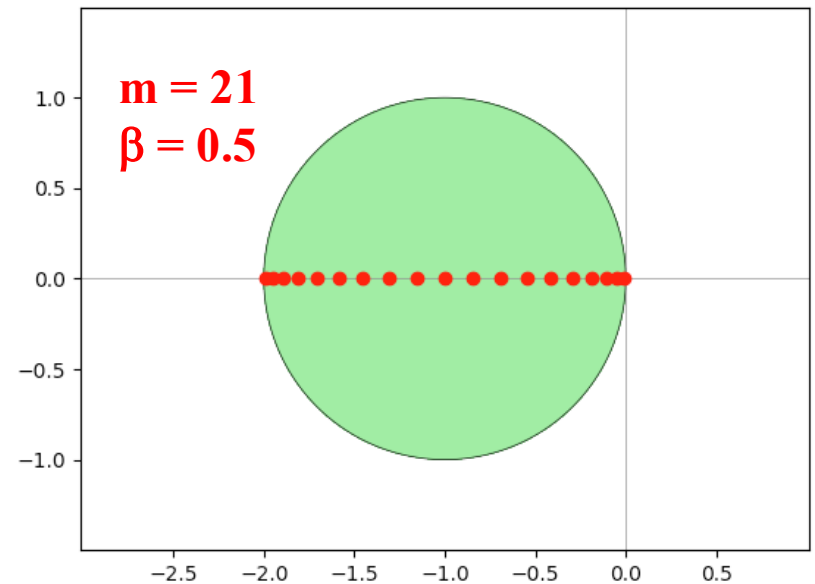
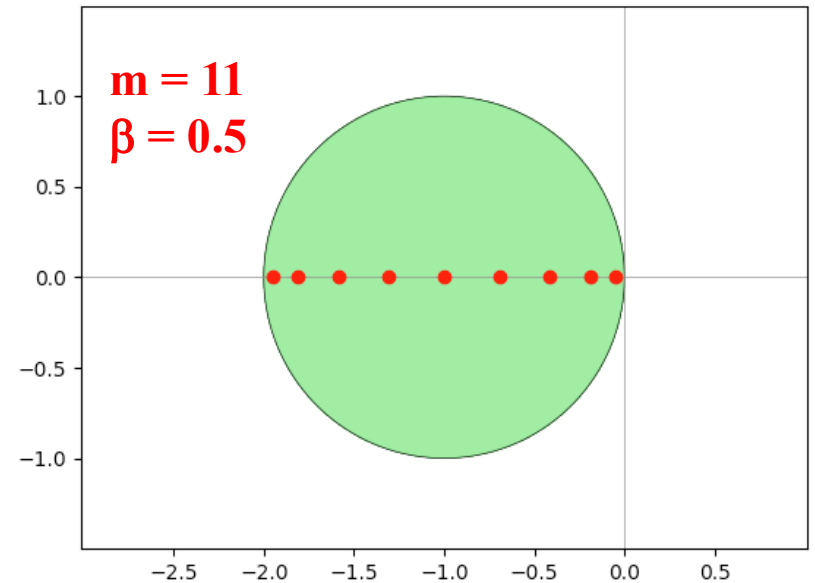
$$\Delta x = 0.2, \Delta t = 0.02$$



$$\Delta x = 0.1, \Delta t = 0.005$$

En raffinant
le maillage...

$$\Delta x = 0.05, \Delta t = 0.00125$$



Adaptons...

$$\mathbf{u}_{n+1} = \underbrace{(\mathbf{I} + \beta \mathbf{A})}_{\mathbf{M}} \mathbf{u}_n$$

Condition pour qu'une méthode itérative converge ?

$$\begin{aligned} \underbrace{(\mathbf{x}_{i+1} - \mathbf{x})}_{\mathbf{e}_{i+1}} &= \mathbf{M}\mathbf{x}_i + \mathbf{c} - \mathbf{M}\mathbf{x} - \mathbf{c} & \mathbf{x}_{i+1} &= \mathbf{M}\mathbf{x}_i + \mathbf{c} \\ &\downarrow \\ &= \mathbf{M}(\mathbf{x}_i - \mathbf{x}) \\ &\downarrow \\ &= \mathbf{M}^{i+1} \underbrace{(\mathbf{x}_0 - \mathbf{x})}_{\mathbf{e}_0} \end{aligned}$$

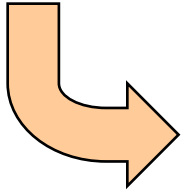
En procédant de la même manière pour chaque étape,

Il faut que...

$$\lim_{i \rightarrow \infty} \mathbf{M}^i \mathbf{e}_0 = 0$$

$$\mathbf{u}_n = \begin{bmatrix} U_1^n \\ U_2^n \\ U_3^n \\ U_4^n \\ U_5^n \\ \\ U_{n_x}^n \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} -2 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & 1 & -2 & 1 & & \\ & & & 1 & -2 & 1 & \\ & & & & & & 1 & -2 \end{bmatrix}$$

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \beta \mathbf{A} \mathbf{u}_n$$



$$|1 + \beta \lambda_i| < 1$$

Et toujours,
les valeurs
propres...

Ecrivons l'erreur comme une
combinaison de vecteurs propres...

$$\mathbf{e}_0 = \sum_{j=1}^n \alpha_j \mathbf{v}_j$$



Puisque $\mathbf{M} \mathbf{v}_i = \lambda_i \mathbf{v}_i$,

$$\mathbf{e}_i = \sum_{j=1}^n \alpha_j \lambda_j^i \mathbf{v}_j$$

Pour obtenir...

$$\lim_{i \rightarrow \infty} \mathbf{M}^i \mathbf{e}_0 = \mathbf{0}$$

*... on doit exiger que le rayon spectral
de la matrice \mathbf{M} soit inférieur à l'unité*

$$|\lambda_i| < 1 \quad \forall i$$

Mais, il faut calculer les valeurs propres d'un opérateur laplacien discret quelconque....

$$\beta = \frac{\alpha \Delta t}{(\Delta x)^2} \leq \frac{1}{2}$$

Courant, Friedrichs et Lewy (1928)

Eux, ils ne disposaient pas de Python....

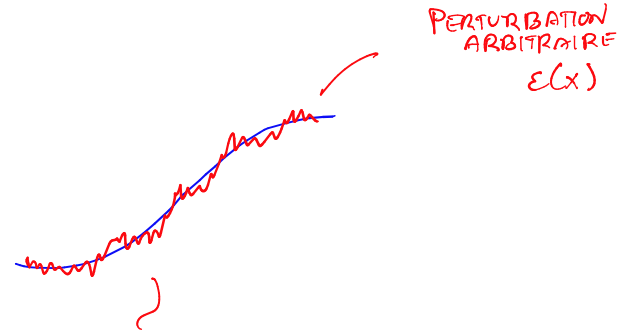
Analyse de stabilité

$$U_i^m \approx \boxed{U^m} \exp(i k X_i)$$

↑
 AMPLITUDE

↑
 NOMBRE D'ONDE QUELCONQUE

↑
 NOMBRE IMAGINAIRE



ON VA LA DECOMPOSER

$$= \text{sin} + \boxed{\text{sin}} + \text{sin} + \dots$$

↑
 ON CHOISIT UNE FREQUENCE ARBITRAIRE
 UN k ARBITRAIRE

Considérons une perturbation quelconque de la forme suivante et analysons son évolution....
 On souhaite que son amplitude diminue.

Analyse de stabilité

Amplitude quelconque de la perturbation

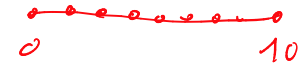
$$U_i^n = U^n e^{ikX_i}$$

The diagram illustrates the components of the perturbation amplitude U_i^n . A blue arrow points from the text 'Amplitude quelconque de la perturbation' to the boxed term U^n in the equation. A green arrow points from the text 'Nombre imaginaire' to the imaginary unit i in the exponent. A blue arrow points from the text 'k quelconque' to the wave number k in the exponent. A red arrow points from the text 'Indice spatial' to the spatial index X_i in the exponent. Another red arrow points from the text 'Indice spatial' to the subscript i of U_i^n .

*Considérons une perturbation quelconque de la forme suivante
et analysons son évolution....*

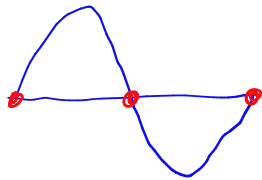
On souhaite que son amplitude diminue.

Quelques k bien choisis...

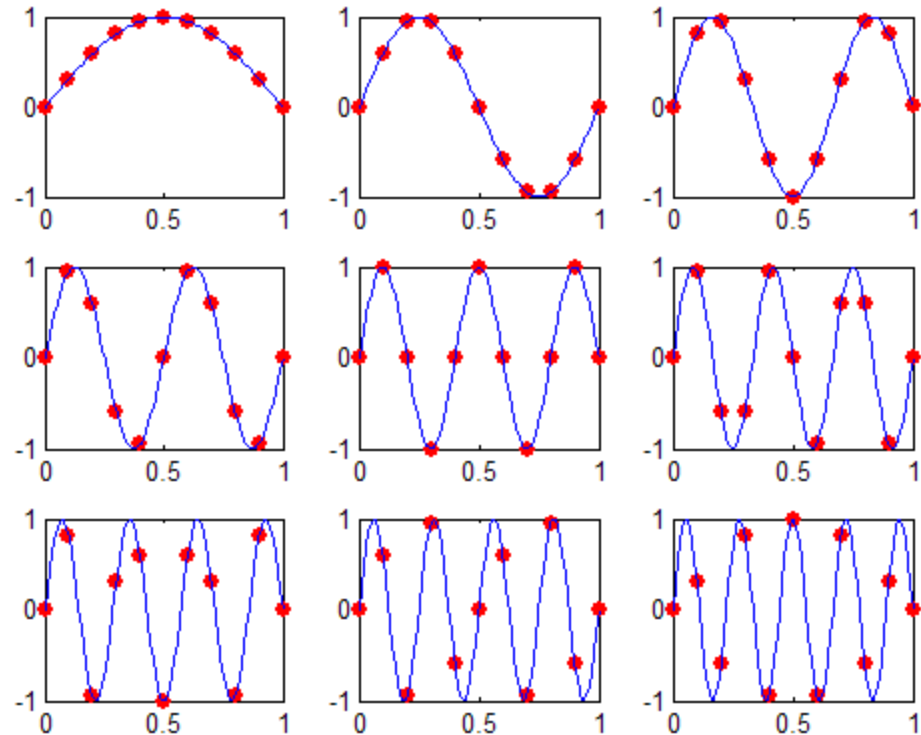


$$U_i^n = U^n \sin\left(\frac{\hat{k}\pi X_i}{L}\right)$$

m = 11
 $\beta = 0.5$



$$\Delta x = 0.1$$



Des perturbations bien particulières et propres...

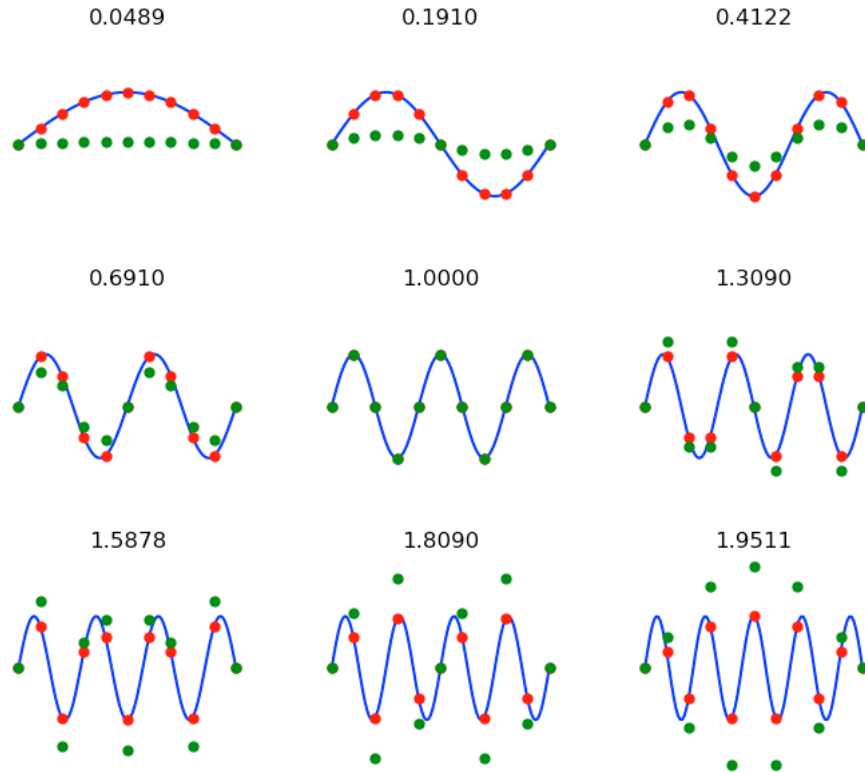
$m = 11$
 $\beta = 0.5$

$$U_i^n = U^n \sin\left(\frac{\hat{k}\pi X_i}{L}\right)$$

$$\beta = \frac{\alpha \Delta t}{(\Delta x)^2}$$

u

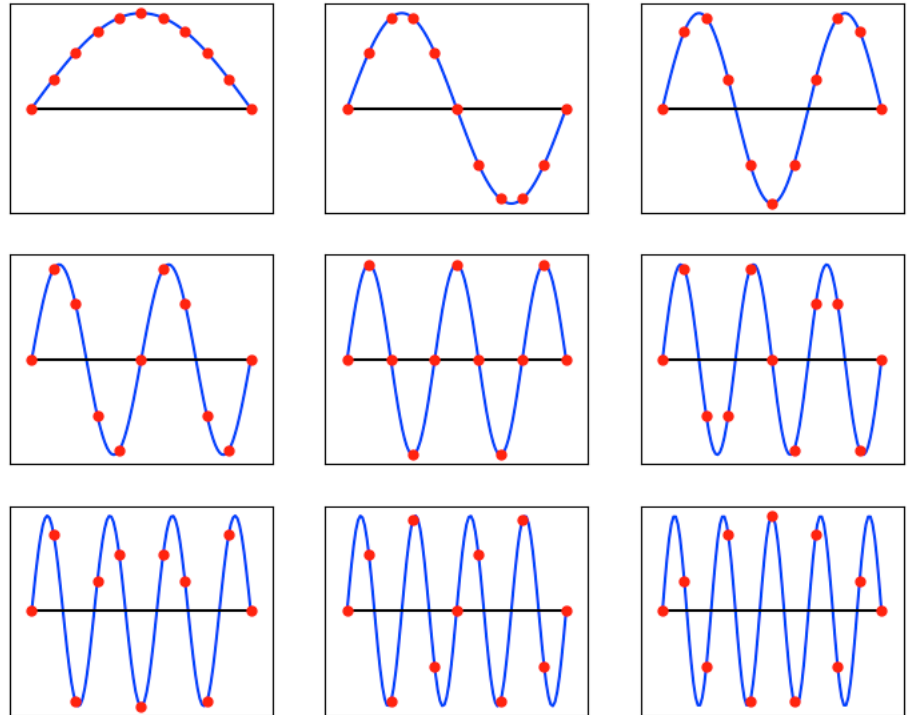
$-\beta \Delta u$



C'est quoi ces perturbations ?

N'importe quelle perturbation peut être écrite comme une combinaison linéaire de ces vecteurs propres de notre opérateur....

$m = 11$
 $\beta = 0.5$



Dimension de l'espace discret = 9
Nombre de vecteurs de base = 9

Calculons
les valeurs
et
les vecteurs
propres...

-0.0489



-0.1910



-0.4122



-0.6910



-1.0000



-1.3090



-1.5878



-1.8090

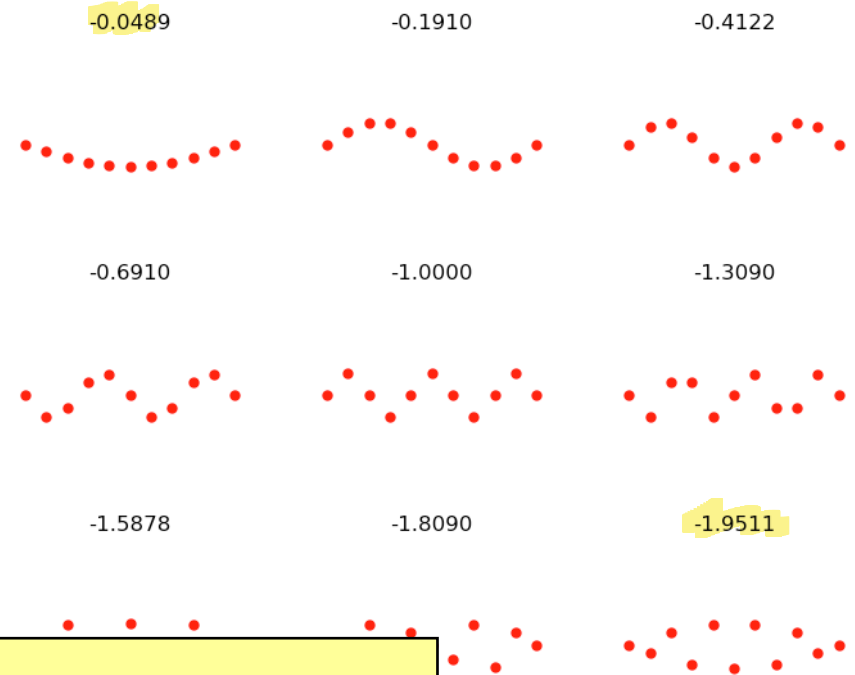


-1.9511

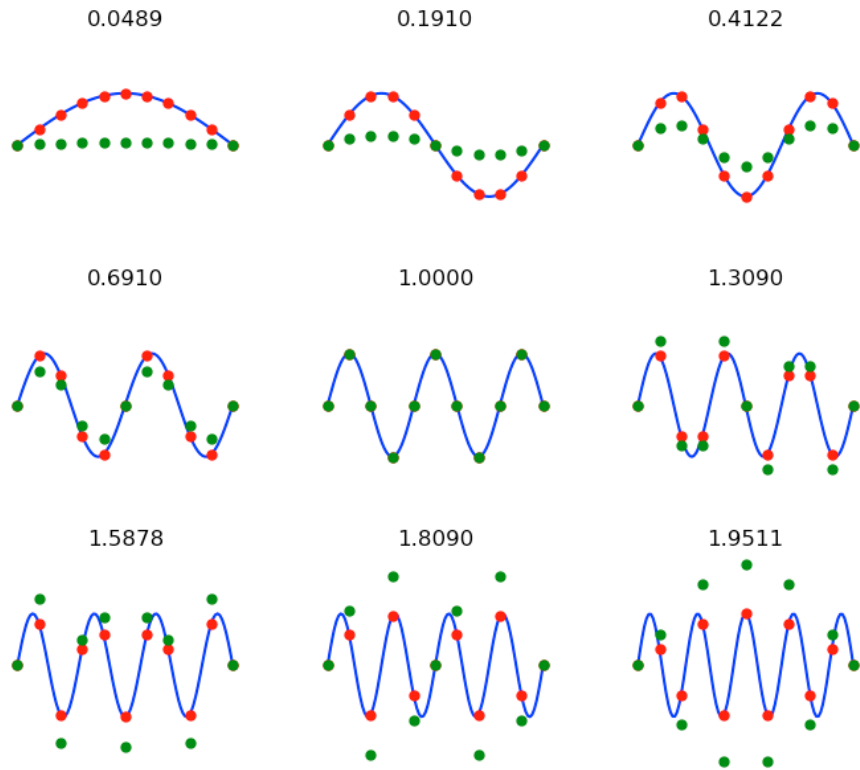


```
m = 11; dx = 1.0/(m-1); beta = (dx*dx)/2
e = array([0,*ones(m-2),0])
D = spdiags([e,-2*e,e],[-1,0,1],m,m)
D = D.T/(dx*dx)
lam,eigen = eig(beta * D.toarray()[1:-1,1:-1])
```

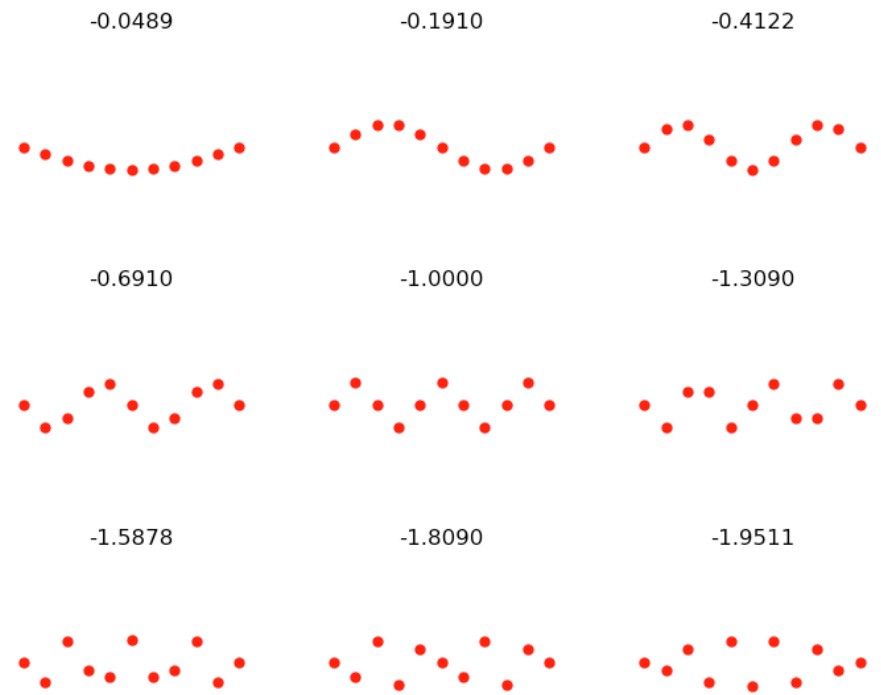
Trier et dessiner les vecteurs propres



```
order = argsort(-lam.real)
X = linspace(0,1,m); V = zeros(m)
for i in range(m-2):
    plt.subplot(msqrt,msqrt,i+1)
    V[1:-1] = eigen[:,order[i]]
    plt.plot(X,V,'.r',markersize=10)
    plt.title("%6.4f" % lam[order[i]].real)
    plt.xlim((-0.1,1.1))
    plt.ylim((-2.1,2.1))
    plt.axis('off')
```



$$U_i^n = U^n \sin\left(\frac{\hat{k}\pi X_i}{L}\right)$$



Et c'est
vraiment cela ?

Propagation des erreurs

$$U_i^n = U^n e^{ikX_i}$$

$$U_i^{n+1} = U_i^n + \beta (U_{i-1}^n + U_{i+1}^n - 2U_i^n)$$

$$X_{i+1} = X_i + \Delta x$$

$$U^{n+1} e^{ikX_i} = U^n e^{ikX_i} + \beta U^n (e^{ikX_i} e^{-ik\Delta x} + e^{ikX_i} e^{ik\Delta x} - 2e^{ikX_i})$$

$$U^{n+1} = U^n \left(1 + \beta (e^{-ik\Delta x} + e^{ik\Delta x} - 2) \right)$$

$$2 \cos(k\Delta x)$$

EN PRENANT
LE CAS LE PLUS
DEFAVORABLE !

$$\cos = -1$$

$$|U^{n+1}| = |U^n| |1 - 4\beta|$$

$$|1 - 4\beta| < 1$$

$$-1 < 1 - 4\beta < 1$$

$$\rightarrow 4\beta < 2 \rightarrow \boxed{\beta = 1/2}$$

Un peu d'algèbre

$$\boxed{\beta < \frac{1}{2}}$$



$$|1 + \beta(2 \cos(k\Delta x) - 2)| \leq 1$$

Un peu
d'algèbre ...

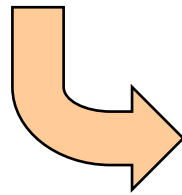


En prenant le cas le plus défavorable
où $k\Delta x = \pi$,



$$-1 \leq 1 - 4\beta$$

$$2\beta \leq 1$$



Condition de stabilité
très pénalisante sur le pas de temps....

$$\beta = \frac{\alpha\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$$

Courant, Friedrichs et Lewy (1928)

Une autre application



Equation hyperbolique

Equation d'onde

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

|
[m²/s²]



$$u(x, t) = \underbrace{f(x+ct)} + \underbrace{g(x-ct)}$$

SOLUTION
D'ALEMBERT

$$\underbrace{c^2 f''} = \underbrace{f''}$$

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2}$$

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f = 0$$

Equation elliptique
Equation de Poisson

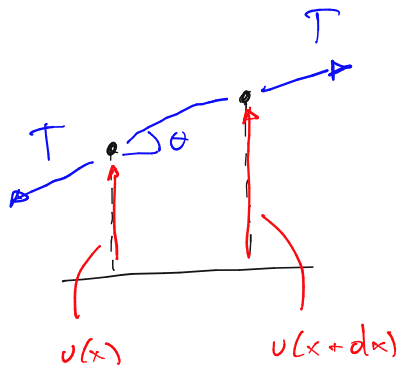
Equation hyperbolique
Equation d'onde

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

Equation parabolique
Equation de la chaleur

Un peu de physique



SI PETITS
DEPLACEMENTS

$$\sin(\theta) \approx \frac{\sin(\theta)}{\cos(\theta)} \approx \frac{\partial u}{\partial x}$$

$$m \vec{a} = \sum \vec{F}$$

$$dx \rho \frac{\partial^2 u}{\partial t^2} = T \left(\underbrace{\sin(\theta(x+dx))}_{\frac{\partial u}{\partial x}(x+dx)} - \underbrace{\sin(\theta(x))}_{\frac{\partial u}{\partial x}(x)} \right)$$

$$\rho \frac{\partial^2 u}{\partial t^2} = \frac{1}{dx} \left[\underbrace{\frac{\partial u}{\partial x}(x+dx)}_{\frac{\partial^2 u}{\partial x^2}} - \frac{\partial u}{\partial x}(x) \right] T$$

$$\frac{\partial^2 u}{\partial t^2} = \underbrace{\frac{T}{\rho}}_{c^2} \frac{\partial^2 u}{\partial x^2}$$

Exemple : corde vibrante

Conditions initiales

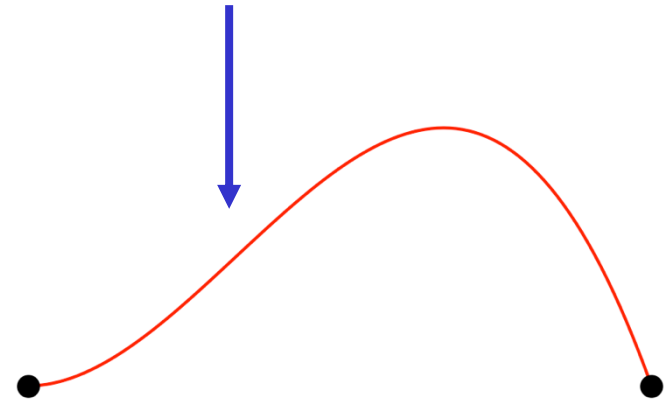
$$u(x, 0) = u_0 \left(1 - \frac{x}{L}\right) \left(\frac{x}{L}\right)^2$$
$$\frac{\partial u}{\partial t}(x, 0) = 0$$

Tension présente dans la corde [N]

$$\rho \frac{\partial^2 u}{\partial t^2} = T_0 \frac{\partial^2 u}{\partial x^2},$$



Masse par unité de longueur [kg/m]



$$u(0, t) = 0 \quad u(L, t) = 0$$

Conditions aux limites

```
from matplotlib import animation
```

```
def frame(i):
```

```
    L = 1; u0 = 1; c = 1
```

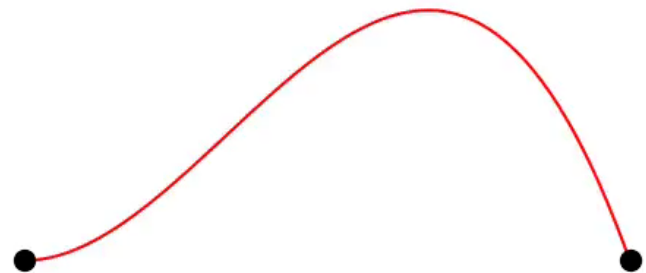
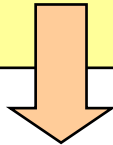
```
    t = i*L/(c*10); x = linspace(0,L,100)
```

```
    plt.clf(); plt.axis("off")
```

```
    plt.plot(x,waveAnalytic(x,t,L,c,u0),"-r")
```

```
movie = animation.FuncAnimation(plt.figure(), frame, 200)
```

```
movie.save('wave.mp4', fps=30)
```



Faire un film
avec matplotlib

Solution analytique

$$u(x, t) = \sum_{n=1}^{\infty} C_n \sin\left(\frac{n\pi x}{L}\right) \cos\left(\frac{n\pi ct}{L}\right)$$

Conditions initiales

$$u(x, 0) = u_0 \left(1 - \frac{x}{L}\right) \left(\frac{x}{L}\right)^2$$

$$\frac{\partial u}{\partial t}(x, 0) = 0$$

$$\rho \frac{\partial^2 u}{\partial t^2} = T_0 \frac{\partial^2 u}{\partial x^2}$$

$$u(0, t) = 0 \quad u(L, t) = 0$$

Conditions aux limites

Comment satisfaire la condition initiale ?

$$u(x, 0) = u_0 \left(\frac{x^2}{L^2} - \frac{x^3}{L^3} \right),$$

$$\sum_{n=1}^{\infty} C_n \sin \left(\frac{n\pi x}{L} \right) = u_0 \left(\frac{x^2}{L^2} - \frac{x^3}{L^3} \right),$$

En vertu de l'orthogonalité des sinus,

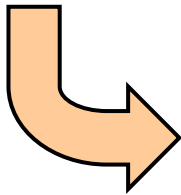
$$\frac{L}{2} C_n = u_0 \int_0^L \left(\frac{x^2}{L^2} - \frac{x^3}{L^3} \right) \sin \left(\frac{n\pi x}{L} \right) dx,$$

$$C_n = 2u_0 \int_0^1 (t^2 - t^3) \sin(n\pi t) dt$$

$$= 2u_0 \left[\left(\frac{2 - (n\pi)^2 t^2}{(n\pi)^3} - \frac{6t - (n\pi)^2 t^3}{(n\pi)^3} \right) \cos(n\pi t) \right]_0^1$$

$$= 2u_0 \left(\frac{2 - (n\pi)^2 - 6 + (n\pi)^2}{(n\pi)^3} (-1)^n - \frac{2}{(n\pi)^3} \right) = \frac{4u_0}{n^3 \pi^3} (2(-1)^{n+1} - 1)$$

Un peu
d'algèbre



```
def waveAnalyticBasic(x,t,L,c,u0):
    u = zeros(size(x))
    for n in range(1,201):
        u = u + ( sin(n*pi * x/L)
                 * cos(n*pi * c*t/L)
                 * (2*(-1)**(n+1) - 1)/(n**3) )
    u = u * 4.0 * u0/(pi**3)
    return u
```

10, 50, 200 ou 1000 termes...
Comment choisir ?

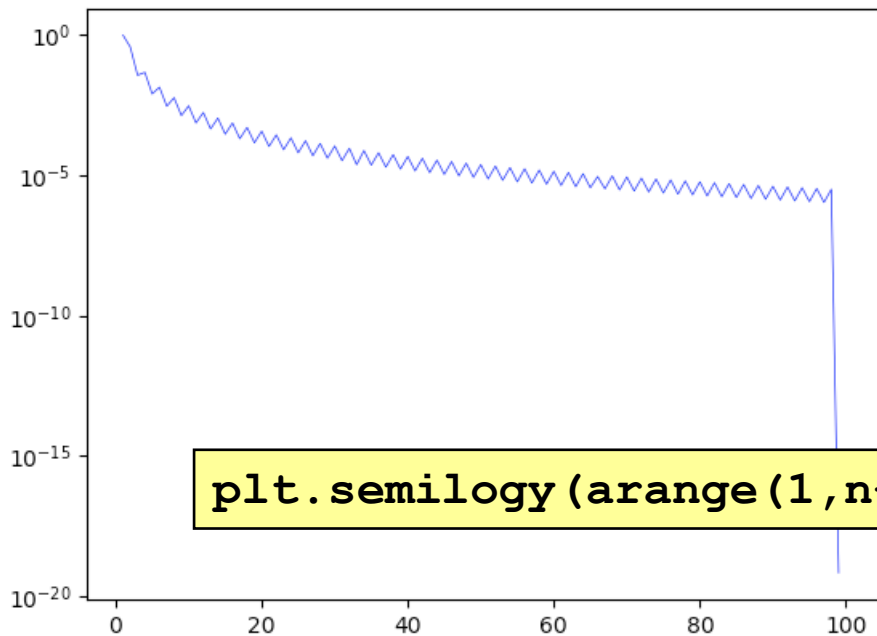
10, 20, 50 ou 100 termes...

Comment choisir ?

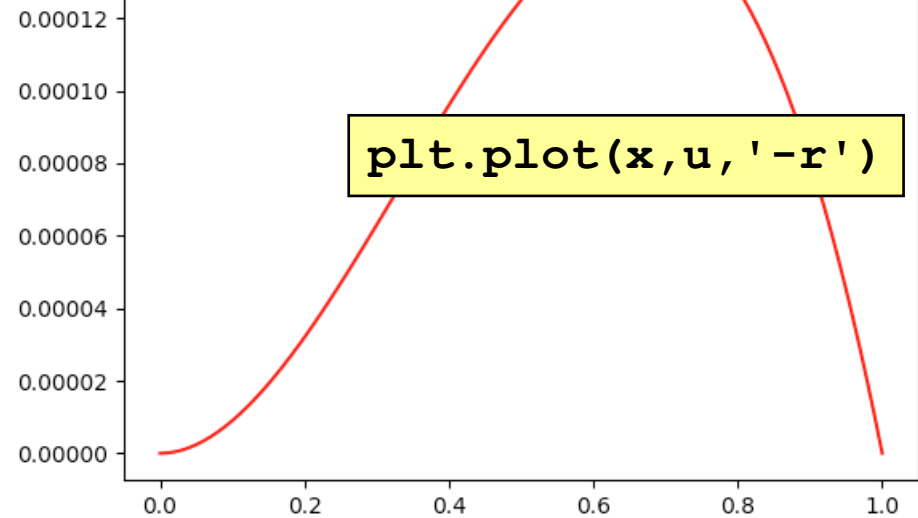
```
def waveAnalyticStupid(x,t,L,c,u0):
    tol = 1e-8; nmax = 2000; n = 1; errorloc = 2*tol
    u = zeros(size(x)); delta = zeros(size(x))
    error = zeros(nmax+1)
    while errorloc > tol and n < nmax:
        delta = ( sin(n*pi * x/L)
                  * cos(n*pi * c*t/L)
                  * (2*(-1)**(n+1) - 1)/(n**3) )
        u = u + delta
        errorloc = max(abs(delta))
        error[n] = errorloc
        n = n+1
    u = u * 4.0 * u0/(pi**3)
    if (n == nmax) :
        print("Error, not converged yek, yek :-(")
    return u,error,n
```

Et boum !

```
>>> x = linspace(0,1,100)
>>> u,error,n = waveAnalytic(x,0,L,c,u0)
>>> print(n)
100
```



```
plt.semilogy(arange(1,n+1),error[1:n+1],'-b')
```

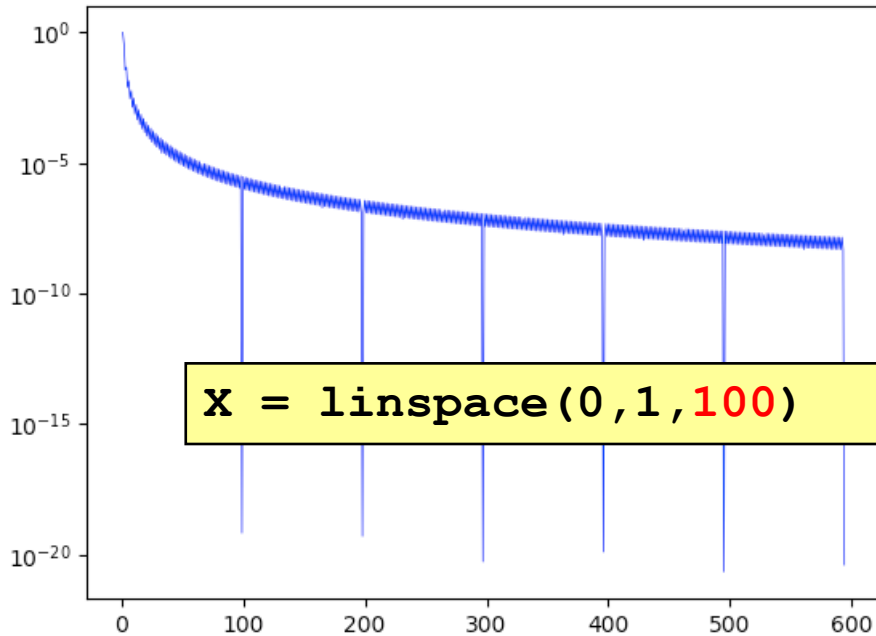


```
plt.plot(x,u,'-r')
```

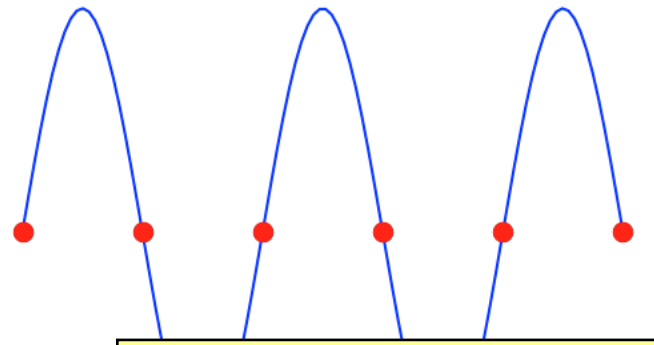
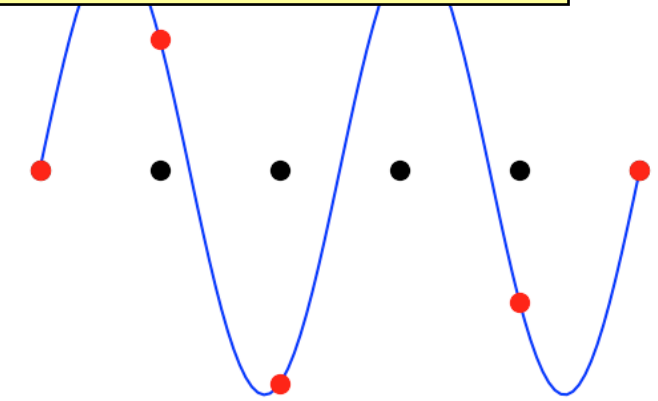
Une meilleure implémentation !

```
def waveAnalytic(x,t,L,c,u0):
    tol = 1e-8; nmax = 2000; n = 1; errorloc = 2*tol
    u = zeros(size(x)); delta = zeros(size(x))
    error = zeros(nmax+1)
    error[0] = errorloc
    while errorloc > tol and n < nmax:
        delta = ( sin(n*pi * x/L)
                  * cos(n*pi * c*t/L)
                  * (2*(-1)**(n+1) - 1)/(n**3) )
        u = u + delta
        error[n] = max(abs(delta))
        errorloc = max(error[n],error[n-1]) ← !!!
        n = n+1
    u = u * 4.0 * u0/(pi**3)
    if (n == nmax) :
        print("Error, not converged yek, yek :-(")
    return u,error,n
```

Termes de la série



```
X = linspace(0, 1, 6)
U = sin(4*pi * X)
```



```
X = linspace(0, 1, 6)
U = sin(5*pi * X)
```

$$u(x, t) = \frac{4u_0}{\pi^3} \sum_{n=1}^{\infty} \frac{1}{n^3} \left(2(-1)^{n+1} - 1 \right) \sin\left(\frac{n\pi x}{L}\right) \cos\left(\frac{n\pi ct}{L}\right)$$

Différences finies spatiales

Equation
aux dérivées partielles
du second ordre

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$



$$\frac{d^2 U_i}{dt^2} = c^2 \frac{U_{i+1} - 2U_i + U_{i-1}}{(\Delta x)^2}$$

$$U_i(t) = u^h(X_i, t) \approx u(X_i, t)$$
$$i = 1, \dots, m$$

Système de m équations
différentielles ordinaires
du second ordre

Deux options possibles pour la discrétisation temporelle

$$\frac{d^2U_i}{dt^2} = c^2 \frac{U_{i+1} - 2U_i + U_{i-1}}{(\Delta x)^2}$$

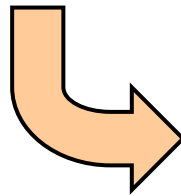
**Système de m équations
différentielles ordinaires
du second ordre**

$$\begin{cases} \frac{dU_i}{dt} = V_i \\ \frac{dV_i}{dt} = c^2 \frac{U_{i+1} - 2U_i + U_{i-1}}{(\Delta x)^2} \end{cases}$$

**Système de 2m équations
différentielles ordinaires
du premier ordre**

$$\begin{cases} \frac{dU_i}{dt} = V_i \\ \frac{dV_i}{dt} = c^2 \frac{U_{i+1} - 2U_i + U_{i-1}}{(\Delta x)^2} \end{cases}$$

2m équadiffs !



$$\frac{dY_i}{dt} = \sum_{j=1}^{2m} A_{ij} Y_j$$

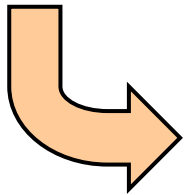
$$Y_i = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix},$$

$$A_{ij} = \left[\begin{array}{cccc|cccc} & & & & 1 & & & & & \\ & & & & & 1 & & & & & \\ & & & & & & 1 & & & & \\ & & & & & & & 1 & & & \\ & & & & & & & & 1 & & \\ & & & & & & & & & 1 & \\ 0 & & & & & & & & & & \\ a & -2a & a & & & & & & & & \\ & a & -2a & a & & & & & & & \\ & & a & -2a & a & & & & & & \\ & & & a & -2a & a & & & & & \\ & & & & & & a & & & & \\ & & & & & & & 0 & & & \end{array} \right]$$

$$a = \frac{c^2}{(\Delta x)^2}$$

Equations différentielles ordinaires avec RK45 :-)

$$\frac{dY_i}{dt} = \sum_{j=1}^{2m} A_{ij} Y_j$$

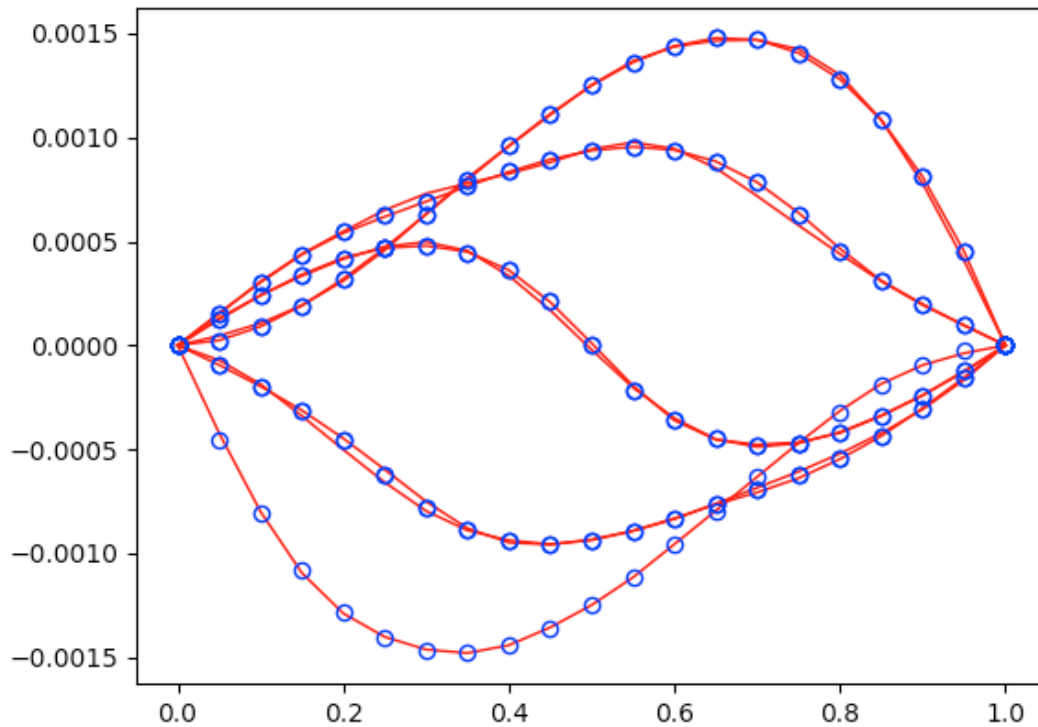


```
x = linspace(0,L,m)
y0 = zeros(2*m);
y0[0:m] = u0 * ((x/L)**2)*(1 - x/L)
sol = solve_ivp(dfdt,[0,Lt],y0,method="RK45",
               rtol=1e-6,atol=1e-9)
```

```
def dfdt(t,y):
    u = y[:m]; v = y[m:]
    dudt = v; dvdt = Dt * u
    return [*dudt,*dvdt]
```

Résultats par RK45 :-)

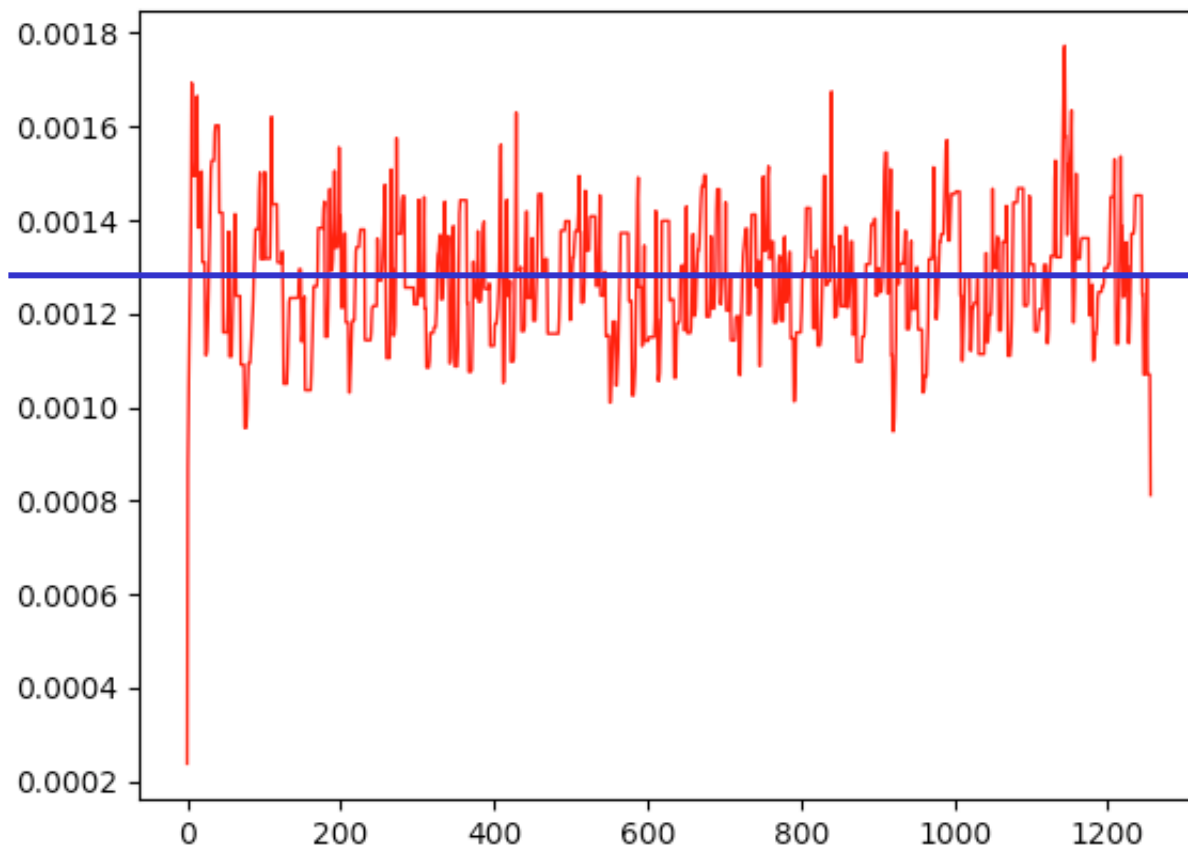
RK45 : 21 valeurs nodales (m=20)



Solution analytique

Résultat sur une période

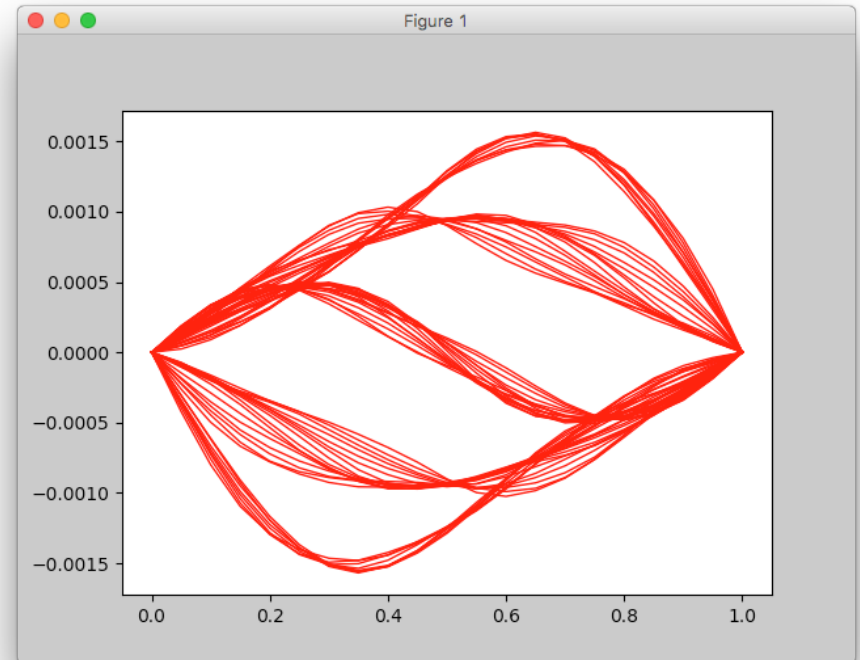
Evolution du pas de temps



$\Delta t \approx 10^{-3}$

Comment obtenir une courbe à un instant précis ?

La courbe à un instant fixé est
obtenue par une interpolation
linéaire entre les courbes des deux
pas de temps les plus proches...



```
nP = 8
t = sol.t; tplot = linspace(0,Lt,8*nP+1)
y = sol.y; yplot = interp1d(t,y)(tplot)
plt.plot(x,yplot[:m,:], '-r',linewidth=1)
```

Les 65 courbes sont dessinées en
une seule instruction...

Energie cinétique

$$E_c(t) = \frac{\rho}{2} \int_0^L \left(\frac{\partial u}{\partial t}(x, t) \right)^2 dx$$

$$E_p(t) = \frac{T_0}{2} \int_0^L \left(\frac{\partial u}{\partial x}(x, t) \right)^2 dx$$

Energie potentielle

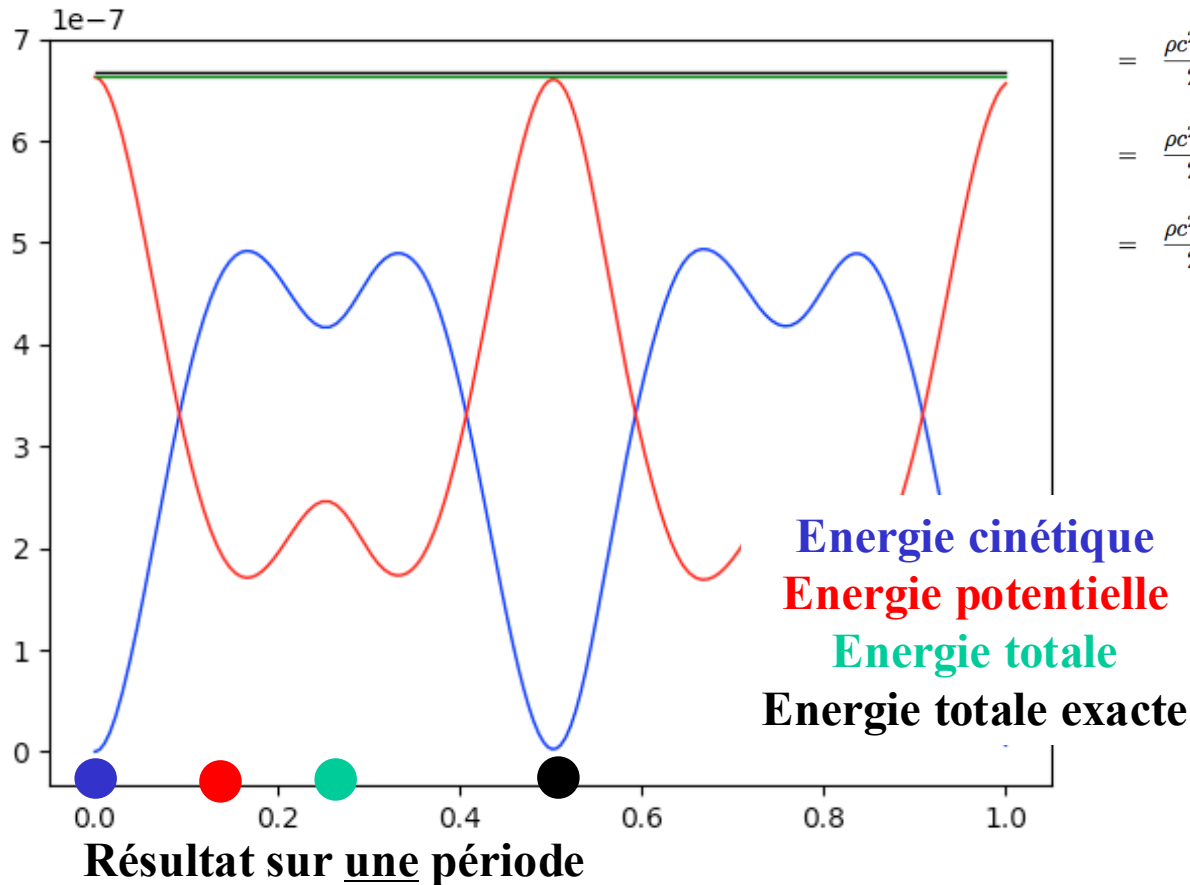
Un peu de
physique



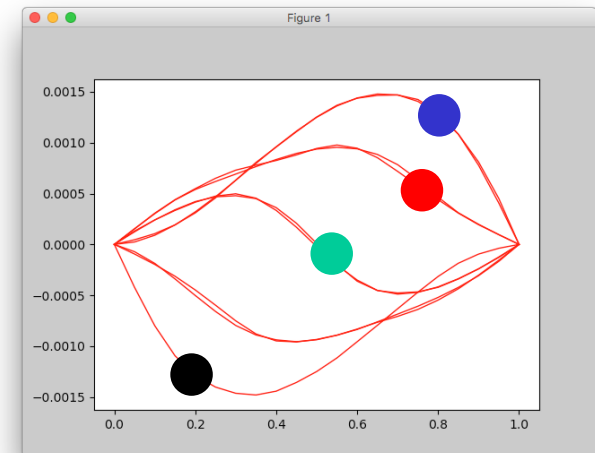
L'énergie totale est conservée...

$$\begin{aligned} \overbrace{E_c(t) + E_p(t)}^E &= \frac{\rho}{2} \int_0^L \left(\frac{\partial u}{\partial t} \right)^2 dx + \frac{\rho c^2}{2} \int_0^L \left(\frac{\partial u}{\partial x} \right)^2 dx \\ &\downarrow \\ \frac{dE}{dt} &= \rho \int_0^L \left(\frac{\partial^2 u}{\partial t^2} \frac{\partial u}{\partial t} + c^2 \frac{\partial u}{\partial x} \frac{\partial^2 u}{\partial x \partial t} \right) dx \\ &= \rho \int_0^L \left(c^2 \frac{\partial^2 u}{\partial x^2} \frac{\partial u}{\partial t} + c^2 \frac{\partial u}{\partial x} \frac{\partial^2 u}{\partial x \partial t} \right) dx \\ &= \rho c^2 \int_0^L \frac{\partial}{\partial x} \left(\frac{\partial u}{\partial x} \frac{\partial u}{\partial t} \right) dx = \rho c^2 \underbrace{\left[\frac{\partial u}{\partial x} \frac{\partial u}{\partial t} \right]_0^L}_{=0} \end{aligned}$$

Numériquement...



$$\begin{aligned}
 E &= \frac{\rho c^2}{2} \int_0^L \left(\frac{\partial u}{\partial x}(x, 0) \right)^2 dx \\
 &= \frac{\rho c^2 u_0^2}{2} \int_0^L \left(\frac{2x}{L^2} - \frac{3x^2}{L^3} \right)^2 dx \\
 &= \frac{\rho c^2 u_0^2}{2} \int_0^L \left(\frac{4x^2}{L^4} - \frac{12x^3}{L^5} + \frac{9x^4}{L^6} \right) dx \\
 &= \frac{\rho c^2 u_0^2}{2} \left[\frac{4x^3}{3L^4} - \frac{12x^4}{4L^5} + \frac{9x^5}{5L^6} \right]_0^L \\
 &= \frac{\rho c^2 u_0^2}{2} \left(\frac{4}{3L} - \frac{3}{L} + \frac{9}{5L} \right) = \frac{\rho c^2 u_0^2}{2L} \left(\frac{20 - 45 + 27}{15} \right) = \frac{\rho c^2 u_0^2}{15L}
 \end{aligned}$$



Deux options possibles pour la discrétisation temporelle

$$\frac{d^2 U_i}{dt^2} = c^2 \frac{U_{i+1} - 2U_i + U_{i-1}}{(\Delta x)^2}$$

**Système de m équations
différentielles ordinaires
du second ordre**

$$\begin{cases} \frac{dU_i}{dt} = V_i \\ \frac{dV_i}{dt} = c^2 \frac{U_{i+1} - 2U_i + U_{i-1}}{(\Delta x)^2} \end{cases}$$

**Système de 2m équations
différentielles ordinaires
du premier ordre**

Différences finies centrées...

Système de m équations différentielles ordinaires du second ordre

$$\frac{d^2 U_i}{dt^2} = c^2 \frac{U_{i+1} - 2U_i + U_{i-1}}{(\Delta x)^2}$$



$$\frac{U_i^{n+1} - 2U_i^n + U_i^{n-1}}{(\Delta t)^2} = c^2 \frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{(\Delta x)^2}$$

Système de m équations aux récurrences à deux termes

$$\frac{U_i^{n+1} - 2U_i^n + U_i^{n-1}}{(\Delta t)^2} = c^2 \frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{(\Delta x)^2}$$

$$U_i^{n+1} = 2U_i^n + \beta^2 (U_{i+1}^n - 2U_i^n + U_{i-1}^n) - U_i^{n-1}$$

$$\beta = \frac{c\Delta t}{\Delta x}$$

Implémentation

```

X = linspace(0,L,nx+1)
Uo = (X/L)**2 * (1-X/L)
Uoo = (2*Uo + (beta**2)*D @ Uo) / 2
for t in range(nt):
    U = 2*Uo + (beta**2)*D @ Uo - Uoo
    Uoo = Uo
    Uo = U

```

Conditions initiales

$$u(x, 0) = u_0 \left(1 - \frac{x}{L}\right) \left(\frac{x}{L}\right)^2$$

$$\frac{\partial u}{\partial t}(x, 0) = 0$$

Analyse de stabilité

Amplitude quelconque de la perturbation

$$U_i^n = U^n e^{ikX_i}$$

Diagram illustrating the components of the perturbation equation $U_i^n = U^n e^{ikX_i}$:

- A blue arrow points from the text "Amplitude quelconque de la perturbation" to the boxed term U^n .
- A green arrow points from the text "Nombre imaginaire" to the imaginary unit i in the exponent.
- A blue arrow points from the text "k quelconque" to the wave number k in the exponent.
- A red arrow points from the text "Indice spatial" to the subscript i on the left-hand side U_i^n .

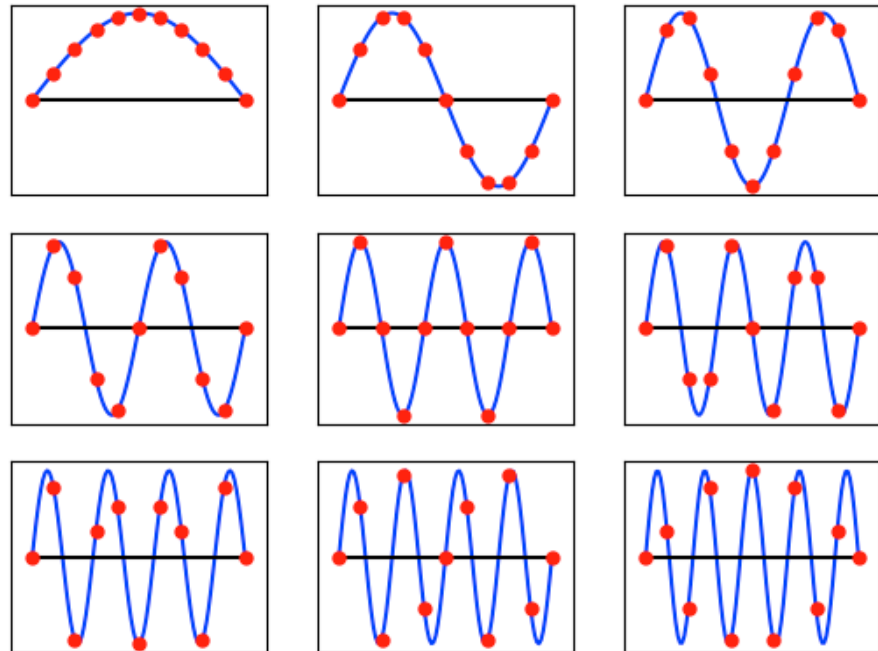
*Considérons une perturbation quelconque de la forme suivante
et analysons son évolution....*

On souhaite que son amplitude diminue.

Quelques k bien choisis...

$m = 11$

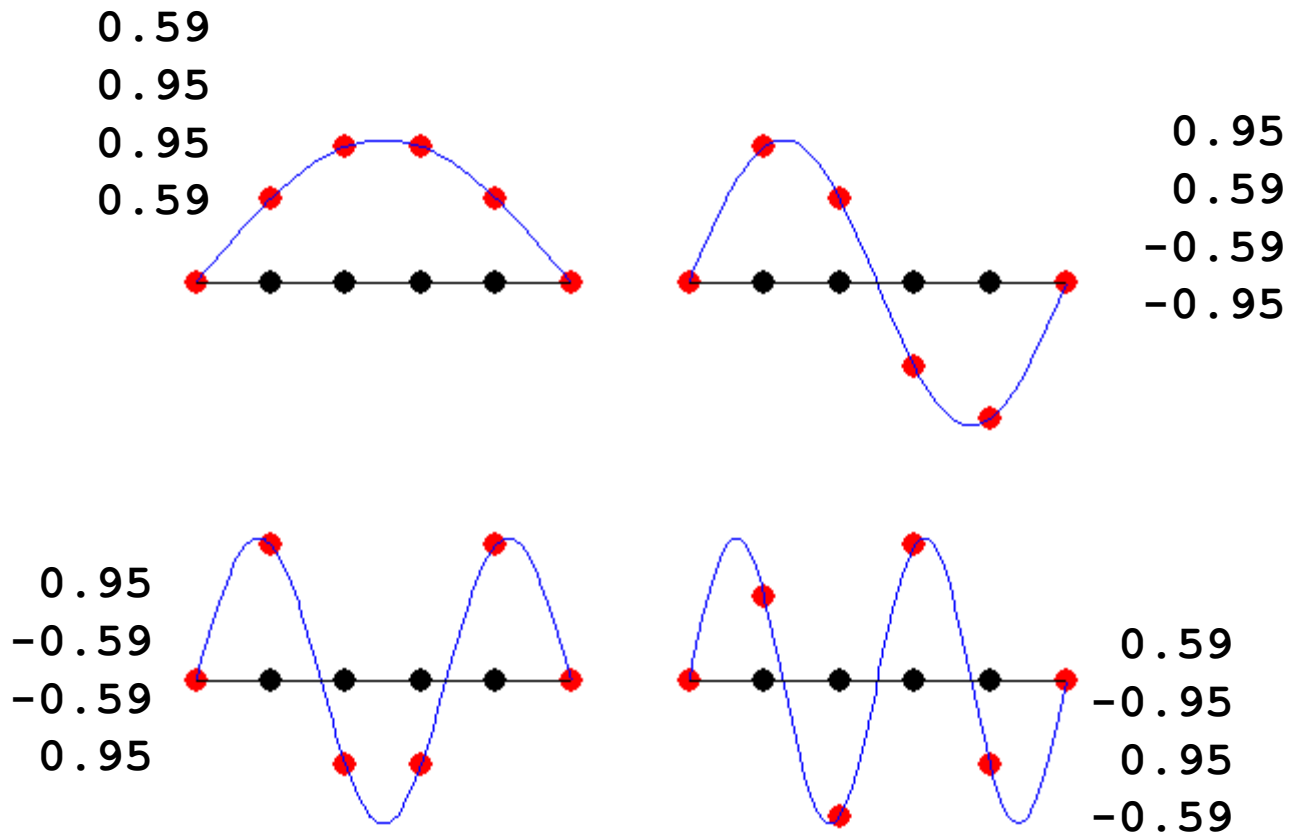
$$U_i^n = U^n \sin\left(\frac{\hat{k}\pi X_i}{L}\right)$$



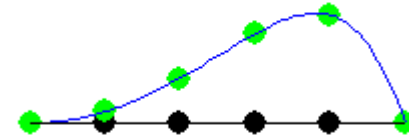
$$\Delta x = 0.1, \Delta t = 0.005$$

Un espace discret de dimension 4

```
X = linspace(0,1,6)
for n=1:4
    U = sin(n * pi * X);
    U(2:5)
end
```



Soit un vecteur tout à fait
quelconque de valeurs
nodales ...



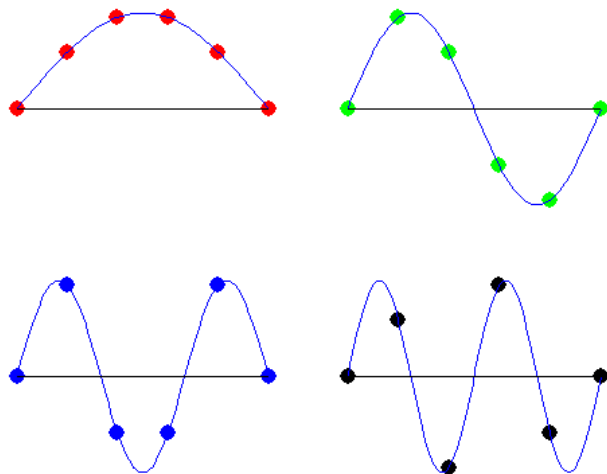
```
x = linspace(0,1,100); u = 2*x.^2.*(1-x.^4);  
X = linspace(0,1,6);    U = 2*X.^2.*(1-X.^4);  
  
plot(X,zeros(size(X)),'.k',...  
      X,U,'.g',...  
      x,u,'-b',...  
      [0 1],[0 0],'-k','Markersize',30);
```

N'importe
 quel vecteur est
 une combili de
 $\sin(n\pi)$

```
A = sin(X(2:5)' * [1:4] * pi)
U = 2 * X.^2 .* (1 - X.^4);
alpha = A \ U(2:5)'
```

A' =

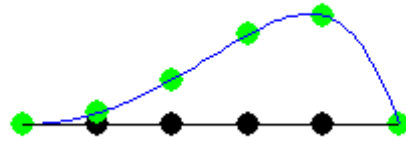
0.5878	0.9511	0.9511	0.5878
0.9511	0.5878	-0.5878	-0.9511
0.9511	-0.5878	-0.5878	0.9511
0.5878	-0.9511	0.9511	-0.5878



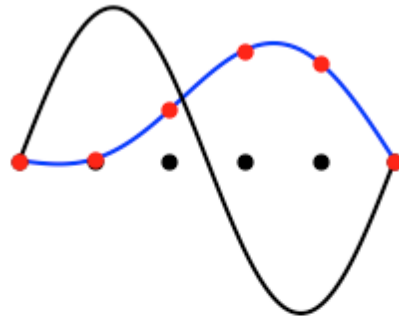
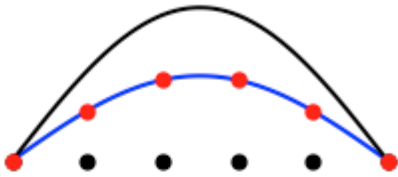
alpha =

0.5535
-0.3311
0.0972
-0.0391

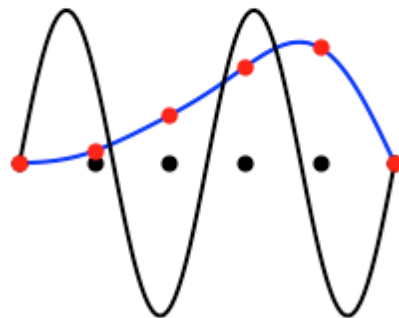
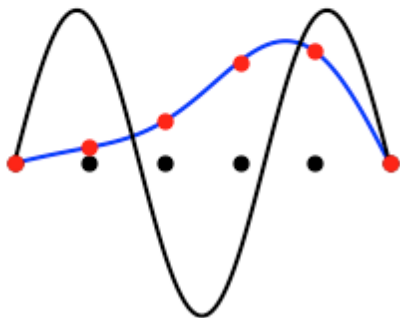
Et je peux
trouver...



```
uh = zeros(100,1)';  
for i=1:4  
    phi = sin(i*pi*x);  
    uh = uh + alpha(i)*phi;  
    plot(x,uh,'-b',x,phi,'-k');  
end
```



alpha =
0.5535
-0.3311
0.0972
-0.0391



... cette
combinaison
de $\sin(n\pi)$

Propagation des erreurs

$$\begin{aligned}U_i^{n+1} &= 2U_i^n + \beta^2 \left(U_{i+1}^n - 2U_i^n + U_{i-1}^n \right) - U_i^{n-1} \\&= U_i^n \left(2 + \beta^2 \left(e^{ik\Delta x} - 2 + e^{-ik\Delta x} \right) \right) - U_i^{n-1} \\&= U_i^n \left(2 + \beta^2 \left(2 \cos(k\Delta x) - 2 \right) \right) - U_i^{n-1} \\&= U_i^n \left(2 - 4\beta^2 \sin^2 \left(\frac{k\Delta x}{2} \right) \right) - U_i^{n-1}\end{aligned}$$

$$U_i^n = U^n e^{ikX_i}$$

Commentaire :

L'analyse de stabilité est ce qu'il y a de plus joli dans le cours !

Méthodes à pas liés :
l'analyse de stabilité nécessite la
résolution d'un polynôme...

$$U_i^{n+1} = U_i^n \left(2 - 4\beta^2 \sin^2 \left(\frac{k\Delta x}{2} \right) \right) - U_i^{n-1}$$

$$U^2 \cancel{U_i^{n-1}} = U \cancel{U_i^{n-1}} \left(2 - 4\beta^2 \sin^2 \left(\frac{k\Delta x}{2} \right) \right) - \cancel{U_i^{n-1}}$$



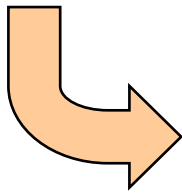
$$U^2 = 2U \underbrace{\left(1 - 2\beta^2 \sin^2 \left(\frac{k\Delta x}{2} \right) \right)}_a - 1$$

Un peu
d'algèbre ...

$$U^2 - 2aU + 1 = 0$$

$$U = a \pm \sqrt{a^2 - 1}$$

- $a^2 = 1$: une racine réelle unique et unitaire
- $a^2 < 1$: deux racines complexes conjuguées de module unitaire
- $a^2 > 1$: deux racines réelles distinctes (une supérieure à un, une inférieure à un)



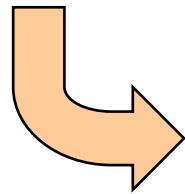
Condition de
stabilité

$$a^2 \leq 1$$

$$\left(1 - 2\beta^2 \sin^2\left(\frac{k\Delta x}{2}\right)\right)^2 \leq 1$$

$$1 - 4\beta^2 \sin^2\left(\frac{k\Delta x}{2}\right) + 4\beta^4 \sin^4\left(\frac{k\Delta x}{2}\right) \leq 1 \quad \dots \text{ et un peu}$$

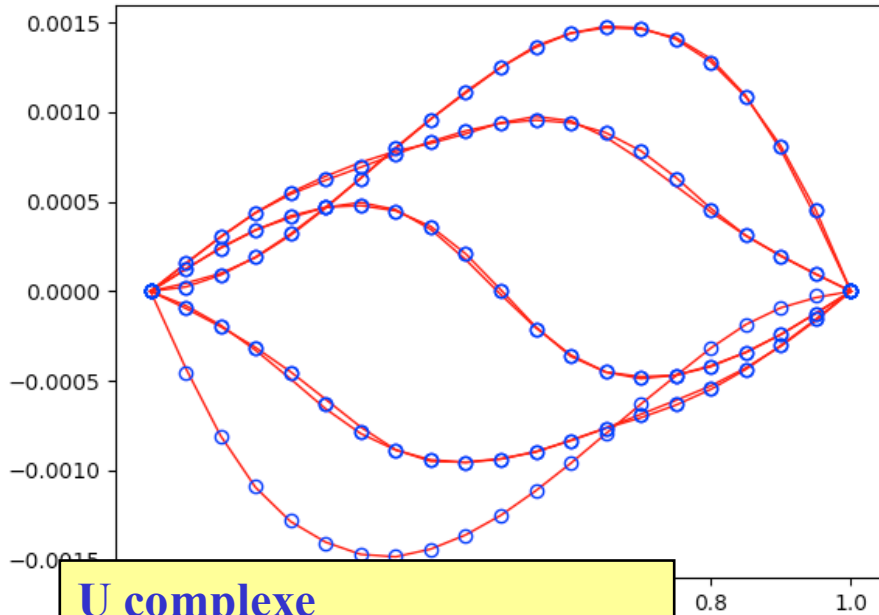
$$-1 + \beta^2 \sin^2\left(\frac{k\Delta x}{2}\right) \leq 0 \quad \text{de calcul}$$



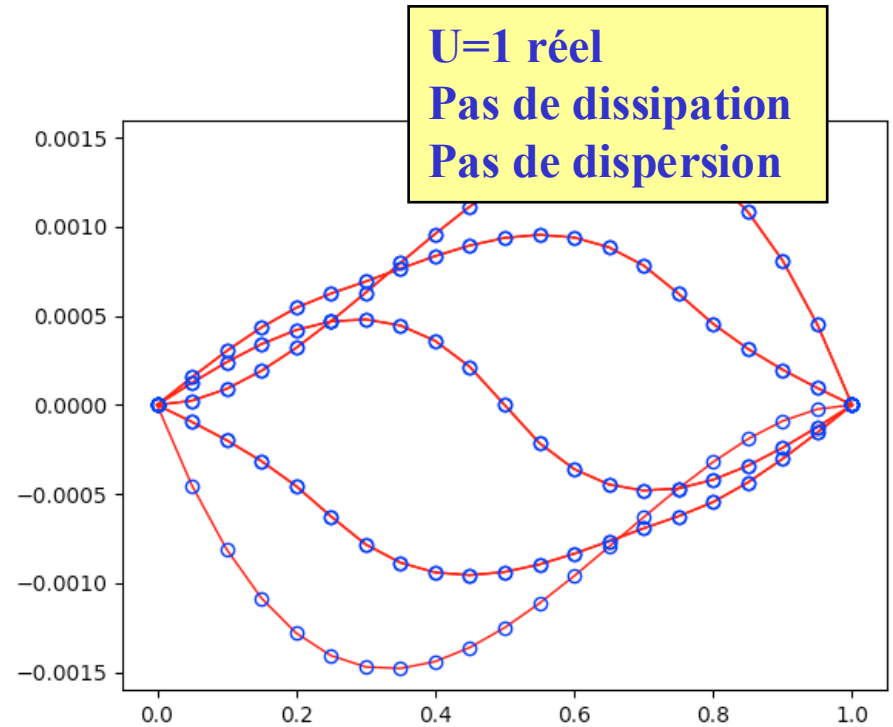
$$\beta = \frac{c\Delta t}{\Delta x} \leq 1$$

**Condition de stabilité
pour une équation d'onde....**

$$\beta = 0.5$$



U complexe
Pas de dissipation
Mais présence de dispersion

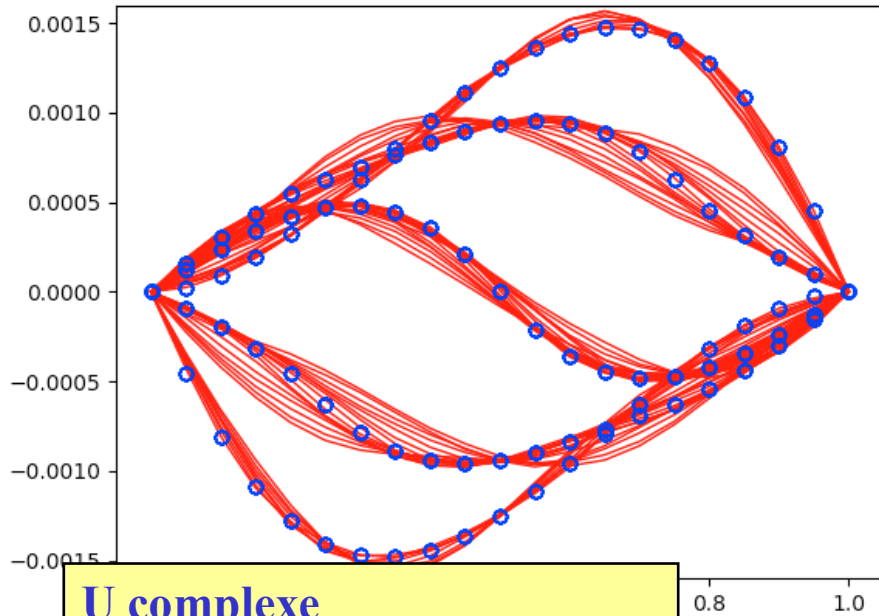


U=1 réel
Pas de dissipation
Pas de dispersion

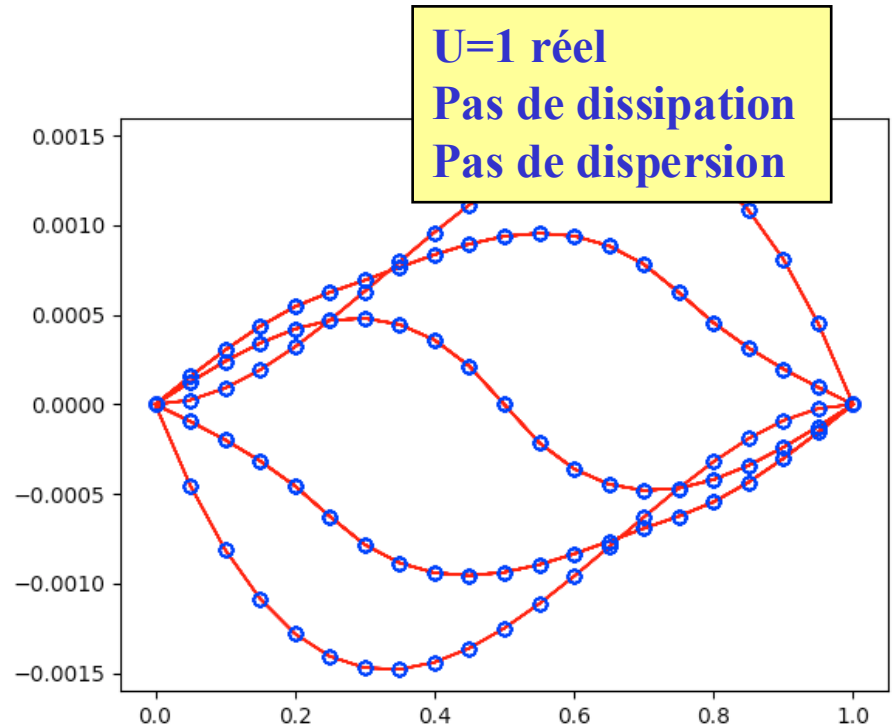
$$\beta = 1.0$$

Résultat sur une période

$$\beta = 0.5$$



U complexe
Pas de dissipation
Mais présence de dispersion

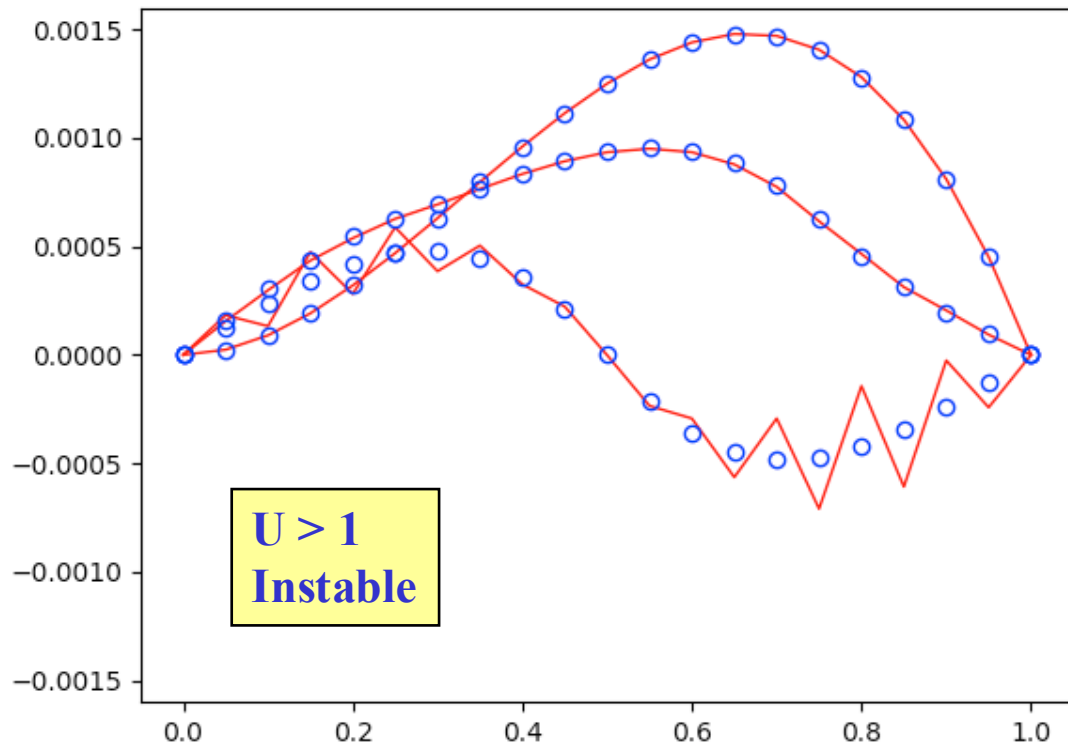


U=1 réel
Pas de dissipation
Pas de dispersion

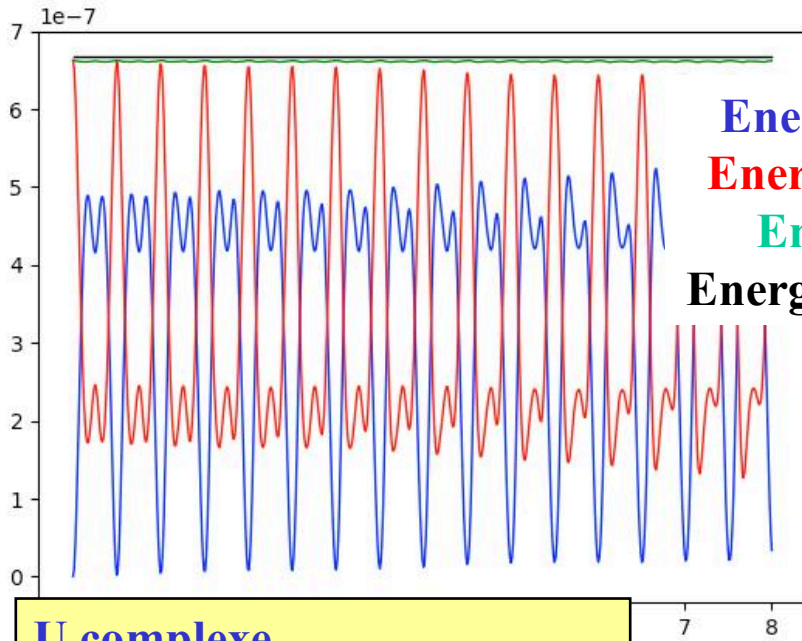
$$\beta = 1.0$$

Résultat sur huit périodes

$$\beta = 1.1$$

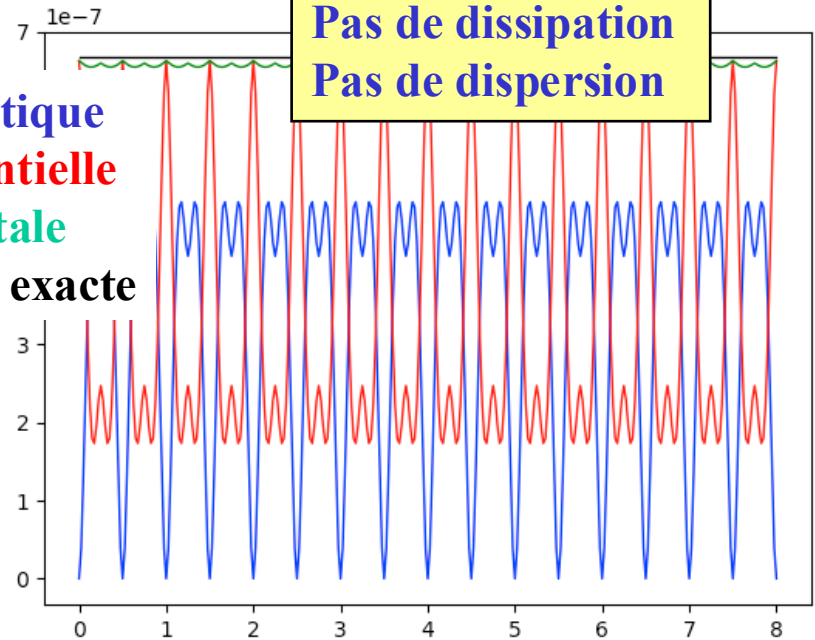


$$\beta = 0.5$$



U complexe
Pas de dissipation
Mais présence de dispersion

Energie cinétique
Energie potentielle
Energie totale
Energie totale exacte

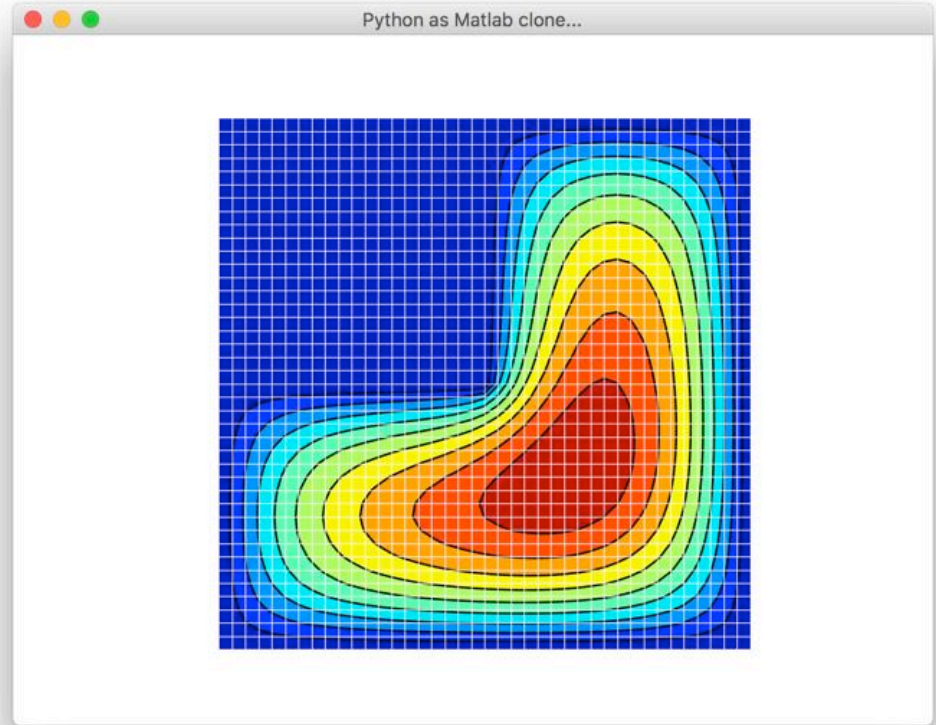


U=1 réel
Pas de dissipation
Pas de dispersion

$$\beta = 1.0$$

Résultat sur huit périodes

Les méthodes
numériques
en 25-26 !



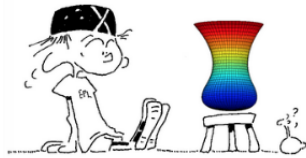
C'est presque fini....

C'est presque fini !

Remise du dernier problème : le lundi 18 mai à 23h59 !

Il est toutefois possible de le faire ce soir !

Et l'examen, Monsieur :-)



Méthodes numériques (LFSAB1104-LICAR1104-LEPL1104)

Vincent Legat
Louvain School of Engineering
Université catholique de Louvain

News	Documents	Python	Examen du 7 janvier 2019	
Liste des étudiants	Equipe didactique	Groupes et locaux	Locaux et horaires des exercices	Evaluation continue
Soumettre le devoir 0 <i>solveRatio</i>	Soumettre le devoir 10 <i>Diffusion</i>			
Evaluation 1 <i>Interpolation</i>	Evaluation 2 <i>Hermite</i>	Evaluation 3 <i>Spline</i>	Evaluation 4 <i>PolyPeriodicFit</i>	Evaluation 5 <i>Trapezoids</i>
Evaluation 6 <i>Romberg</i>	Evaluation 7 <i>DoublePrime</i>	Evaluation 8 <i>Lorenz</i>	Evaluation 9 <i>Ladders</i>	

Le numéro magique de l'examen sera différent de celui de l'interrogation.

Quelques rappels utiles pour l'examen du 7 janvier 2019

A connaître pour le jour de l'examen :

Votre année d'étude : FSA 1 BA

Votre noma : 44841700

Votre auditoire : **Unspecified**

Votre numéro magique pour le classement de la question ouverte : **400**

Pour remettre (et éventuellement reprendre) une feuille blanche pour l'examen FSAB1104, il n'est pas nécessaire de se déplacer le jour de l'examen...

- C'est très simple. Il suffit de cliquer ci-dessous :

 Remettre une feuille blanche

- Les étudiants qui se présentent dans les salles d'examen **devront obligatoirement rester deux heures dans la salle d'examen** même si ils souhaitent remettre une feuille blanche. Vous êtes donc vraiment encouragés à utiliser le formulaire électronique pour remettre virtuellement votre feuille blanche.
- Il ne sera pas possible de remettre une feuille blanche en vous présentant dans la salle d'examen : la procédure informatique de soumission virtuelle est obligatoire.

 Etudiants dans l'auditoire A10

 Etudiants dans l'auditoire A01

Et l'examen, Monsieur :-)

Question python

Un programme de 10 lignes à écrire

Les programmes des transparents sont supposés compris

Les 11 problèmes python sont supposés compris

Question « application de l'acquis »

Exercice simple fortement inspiré des exercices simples des notes

Question « compréhension »

Exercice plus subtil dû à la créativité de l'enseignant

Pas de calculatrice,

Formulaire manuscrit recto-verso,

Les travaux python interviendront pour 10% de la note de l'examen.

The so-called Stupid Single Student Behaviour Law



