

Méthode d'Euler pour les EDP

Ivan Noyer

Lycée Thiers

1 Ordre 1

2 Ordre supérieur

- « Le tout en 1, Informatique », JP Préaux, Ed. Lavoisier,
- Wikipedia, méthode d'Euler

1 Ordre 1

2 Ordre supérieur

Rappel : Méthode d'Euler

- Soit U un ouvert de $\mathbb{R}^2$ et $f : U \rightarrow \mathbb{R}$ une fonction de classe $\mathcal{C}^1$ sur U (i.e. les dérivées partielles existent et sont continues).

Rappel : Méthode d'Euler

- Soit U un ouvert de $\mathbb{R}^2$ et $f : U \rightarrow \mathbb{R}$ une fonction de classe $\mathcal{C}^1$ sur U (i.e. les dérivées partielles existent et sont continues).
- L'équation $y' = f(t, y)$ avec condition de Cauchy $(t_0, y_0) \in U$ et $y(t_0) = y_0$ admet une solution unique y de classe $\mathcal{C}^1$ définie sur un intervalle maximum I que l'on peut approcher ainsi :

$$\begin{aligned} y_0 &= y(t_0) \\ y_{k+1} &= y_k + hf(t_k, y_k) : \text{schéma d'Euler explicite} \\ t_{k+1} &= t_k + h \end{aligned}$$

Le nombre $h > 0$ est appelé *pas*. Cette définition par récurrence de la suite (y_k) est appelée *schéma d'Euler*

Rappel : Méthode d'Euler

- Soit U un ouvert de $\mathbb{R}^2$ et $f : U \rightarrow \mathbb{R}$ une fonction de classe $\mathcal{C}^1$ sur U (i.e. les dérivées partielles existent et sont continues).
- L'équation $y' = f(t, y)$ avec condition de Cauchy $(t_0, y_0) \in U$ et $y(t_0) = y_0$ admet une solution unique y de classe $\mathcal{C}^1$ définie sur un intervalle maximum I que l'on peut approcher ainsi :

$$\begin{aligned} y_0 &= y(t_0) \\ y_{k+1} &= y_k + hf(t_k, y_k) : \text{schéma d'Euler explicite} \\ t_{k+1} &= t_k + h \end{aligned}$$

Le nombre $h > 0$ est appelé *pas*. Cette définition par récurrence de la suite (y_k) est appelée *schéma d'Euler*

- La relation $y_{k+1} = y_k + hf(t_k, y_k)$ est obtenue en faisant un développement de Taylor à l'ordre 1 :

$$y'(t) \simeq \frac{y(t+h) - y(t)}{h} : \text{taux d'accroissement}$$

EDP d'ordre 1

- Le schéma d'Euler explicite s'applique bien aux EDP.
- Soit $F : U \subset \mathbb{R}^n \rightarrow \mathbb{R}^p$ une application de classe $\mathcal{C}^1$ (i.e. les dérivées partielles existent et sont continues) :

$$\frac{\partial F}{\partial x_i}(x_1, \dots, x_{i-1}, \mathbf{x}_i, x_{i+1}, \dots, x_n) \simeq \frac{F(x_1, \dots, x_{i-1}, \mathbf{x}_i + d\mathbf{x}_i, x_{i+1}, \dots, x_n) - F(x_1, \dots, x_{i-1}, \mathbf{x}_i, x_{i+1}, \dots, x_n)}{dx_i}$$

où $x_i + dx_i$ désigne une petite variation autour de x_i .

Exemple

On pose $a < b$ et

$$F : [a, b] \times [0, T] \rightarrow \mathbb{R}$$

tel que $\forall (x, t) \in [a, b] \times [0, T]$ on a $\frac{\partial F(x, t)}{\partial t} = \frac{\partial F(x, t)}{\partial x}$ avec conditions initiales

$$\text{pour tous } x, t : (7) \begin{cases} F(x, 0) = f(x) = x^2 \\ F(0, t) = g(t) = t^2 \end{cases}$$

- On veut déterminer des valeurs approchées de $x \mapsto F(x, t)$ pour $x \in [0, 10]$ et $t \in [0, 15]$ (donc $a = 0, b = 10, T = 15$).

Exemple

On pose $a < b$ et

$$F : [a, b] \times [0, T] \rightarrow \mathbb{R}$$

tel que $\forall (x, t) \in [a, b] \times [0, T]$ on a $\frac{\partial F(x, t)}{\partial t} = \frac{\partial F(x, t)}{\partial x}$ avec conditions initiales

$$\text{pour tous } x, t : (7) \begin{cases} F(x, 0) = f(x) = x^2 \\ F(0, t) = g(t) = t^2 \end{cases}$$

- On veut déterminer des valeurs approchées de $x \mapsto F(x, t)$ pour $x \in [0, 10]$ et $t \in [0, 15]$ (donc $a = 0, b = 10, T = 15$).
- On fait un *choix de résolution* : une fois qu'on connaît $x \mapsto F(x, t)$ on veut en déduire $x \mapsto F(x, t + dt)$. On choisit *empiriquement* un schéma à droite en t et à gauche en x .

Exemple

$$\frac{\partial F(x, t)}{\partial t} = \frac{\partial F(x, t)}{\partial x} \text{ avec conditions initiales (7).}$$

- On pose

$$\frac{\partial F}{\partial t}(x, t) = \frac{F(x, t + dt) - F(x, t)}{dt} : \text{schéma d'Euler à droite en } t$$

et

$$\frac{\partial F}{\partial x}(x, t) = \frac{F(x, t) - F(x - dx, t)}{dx} : \text{schéma d'Euler à gauche en } x$$

Exemple

$$\frac{\partial F(x, t)}{\partial t} = \frac{\partial F(x, t)}{\partial x} \text{ avec conditions initiales (7).}$$

- On pose

$$\frac{\partial F}{\partial t}(x, t) = \frac{F(x, t + dt) - F(x, t)}{dt} : \text{schéma d'Euler à droite en } t$$

et

$$\frac{\partial F}{\partial x}(x, t) = \frac{F(x, t) - F(x - dx, t)}{dx} : \text{schéma d'Euler à gauche en } x$$

- On a donc pour tous x, t convenables, $F(x, 0) = x^2$ et

$$\frac{F(x, t + dt) - F(x, t)}{dt} = \frac{F(x, t) - F(x - dx, t)}{dx}$$

Exemple

$$\frac{\partial F(x, t)}{\partial t} = \frac{\partial F(x, t)}{\partial x} \text{ avec conditions initiales (7).}$$

- On pose

$$\frac{\partial F}{\partial t}(x, t) = \frac{F(x, t + dt) - F(x, t)}{dt} : \text{schéma d'Euler à droite en } t$$

et

$$\frac{\partial F}{\partial x}(x, t) = \frac{F(x, t) - F(x - dx, t)}{dx} : \text{schéma d'Euler à gauche en } x$$

- On a donc pour tous x, t convenables, $F(x, 0) = x^2$ et

$$\frac{F(x, t + dt) - F(x, t)}{dt} = \frac{F(x, t) - F(x - dx, t)}{dx}$$

- Donc (en commençant à $x_1 = dx$ puisqu'on connaît $F(0, t)$) :

$$F(x, t + dt) = \frac{dt}{dx}(F(x, t) - F(x - dx, t)) + F(x, t)$$

Autre schéma

On pourrait aussi choisir un schéma à gauche en x et t :



$$\frac{F(x, t) - F(x, t - dt)}{dt} = \frac{F(x, t) - F(x - dx, t)}{dx}$$

Autre schéma

On pourrait aussi choisir un schéma à gauche en x et t :

- $$\frac{F(x, t) - F(x, t - dt)}{dt} = \frac{F(x, t) - F(x - dx, t)}{dx}$$

- Donc

$$F(x, t) = \frac{1}{dx - dt}(dx F(x, t - dt) - dt F(x - dx, t))$$

1 Ordre 1

2 Ordre supérieur

Approximation de la dérivée seconde Soit y de classe $\mathcal{C}^2$, on a

- $y(x + h) = y(x) + hy'(x) + \frac{h^2}{2}y''(x) + o(h^2)$

Approximation de la dérivée seconde Soit y de classe $\mathcal{C}^2$, on a

- $y(x + h) = y(x) + hy'(x) + \frac{h^2}{2}y''(x) + o(h^2)$
- $y(x - h) = y(x) - hy'(x) + \frac{h^2}{2}y''(x) + o(h^2)$

Approximation de la dérivée seconde Soit y de classe $\mathcal{C}^2$, on a

- $y(x + h) = y(x) + hy'(x) + \frac{h^2}{2}y''(x) + o(h^2)$
- $y(x - h) = y(x) - hy'(x) + \frac{h^2}{2}y''(x) + o(h^2)$
- Donc $y(x + h) + y(x - h) = 2y(x) + h^2y''(x) + o(h^2)$

Approximation de la dérivée seconde Soit y de classe $\mathcal{C}^2$, on a

- $y(x + h) = y(x) + hy'(x) + \frac{h^2}{2}y''(x) + o(h^2)$
- $y(x - h) = y(x) - hy'(x) + \frac{h^2}{2}y''(x) + o(h^2)$
- Donc $y(x + h) + y(x - h) = 2y(x) + h^2y''(x) + o(h^2)$
- Alors

$$y''(x) = \frac{y(x + h) + y(x - h) - 2y(x)}{h^2} + \underbrace{o(1)}_{\text{marge d'erreur}}$$

Approximation de la dérivée seconde Soit y de classe $\mathcal{C}^2$, on a

- $y(x + h) = y(x) + hy'(x) + \frac{h^2}{2}y''(x) + o(h^2)$
- $y(x - h) = y(x) - hy'(x) + \frac{h^2}{2}y''(x) + o(h^2)$
- Donc $y(x + h) + y(x - h) = 2y(x) + h^2y''(x) + o(h^2)$
- Alors

$$y''(x) = \frac{y(x + h) + y(x - h) - 2y(x)}{h^2} + \underbrace{o(1)}_{\text{marge d'erreur}}$$

- En fait, l'erreur est plus petite que $o(1)$ lorsqu'on travaille avec une fonction de classe $\mathcal{C}^3$

Approximation de la dérivée seconde Soit y de classe $\mathcal{C}^3$, on a

- $$y(x+h) = y(x) + hy'(x) + \frac{h^2}{2}y''(x) + \frac{h^3}{6}y^{(3)}(x) + o(h^3)$$

Approximation de la dérivée seconde Soit y de classe $\mathcal{C}^3$, on a

- $y(x + h) = y(x) + hy'(x) + \frac{h^2}{2}y''(x) + \frac{h^3}{6}y^{(3)}(x) + o(h^3)$
- $y(x - h) = y(x) - hy'(x) + \frac{h^2}{2}y''(x) - \frac{h^3}{6}y^{(3)}(x) + o(h^3)$

Approximation de la dérivée seconde Soit y de classe $\mathcal{C}^3$, on a

- $y(x + h) = y(x) + hy'(x) + \frac{h^2}{2}y''(x) + \frac{h^3}{6}y^{(3)}(x) + o(h^3)$
- $y(x - h) = y(x) - hy'(x) + \frac{h^2}{2}y''(x) - \frac{h^3}{6}y^{(3)}(x) + o(h^3)$
- Donc $y(x + h) + y(x - h) = 2y(x) + h^2y''(x) + o(h^3)$

Approximation de la dérivée seconde Soit y de classe $\mathcal{C}^3$, on a

- $y(x + h) = y(x) + hy'(x) + \frac{h^2}{2}y''(x) + \frac{h^3}{6}y^{(3)}(x) + o(h^3)$
- $y(x - h) = y(x) - hy'(x) + \frac{h^2}{2}y''(x) - \frac{h^3}{6}y^{(3)}(x) + o(h^3)$
- Donc $y(x + h) + y(x - h) = 2y(x) + h^2y''(x) + o(h^3)$
- Alors

$$y''(x) = \frac{y(x + h) + y(x - h) - 2y(x)}{h^2} + o(h)$$

EDP

- On a

$$\frac{\partial F}{\partial x_i}(\dots, x_i, \dots) \simeq \frac{F(\dots, x_i + h, \dots) - F(\dots, x_i, \dots)}{h}$$

EDP

- On a

$$\frac{\partial F}{\partial x_i}(\dots, x_i, \dots) \simeq \frac{F(\dots, x_i + h, \dots) - F(\dots, x_i, \dots)}{h}$$

- Et en se fiant à ce qui a été établi à l'ordre 2 :

$$\frac{\partial^2 F}{\partial x_i^2}(\dots, x_i, \dots) \simeq \frac{F(\dots, x_i + h, \dots) + F(\dots, x_i - h, \dots) - 2F(\dots, x_i, \dots)}{h^2}$$

Diffusion de la chaleur le long d'une barre métallique

Une barre métallique de longueur ℓ a un coefficient de diffusion K . Une extrémité est reliée à une source de chaleur de température T_1 , l'autre à une source de température T_2 telle que $T_1 < T_2$. $T(x, t)$ est la température au point d'abscisse x au temps t .

$$\frac{\partial T}{\partial t}(x, t) = K \frac{\partial^2 T}{\partial x^2}(x, t)$$

- Schéma à droite en t

$$\frac{\partial T}{\partial t}(x, t) \simeq \frac{T(x, t + dt) - T(x, t)}{dt}$$

Diffusion de la chaleur le long d'une barre métallique

$$\frac{\partial T}{\partial t}(x, t) = K \frac{\partial^2 T}{\partial x^2}(x, t)$$

- Schéma à droite en t

$$\frac{\partial T}{\partial t}(x, t) \simeq \frac{T(x, t + dt) - T(x, t)}{dt}$$



$$\frac{\partial^2 T}{\partial x^2}(x, t) \simeq \frac{T(x + dx, t) + T(x - dx, t) - 2T(x, t)}{dx^2}$$

Diffusion de la chaleur le long d'une barre métallique

$$\frac{\partial T}{\partial t}(x, t) = K \frac{\partial^2 T}{\partial x^2}(x, t)$$

- Schéma à droite en t

$$\frac{\partial T}{\partial t}(x, t) \simeq \frac{T(x, t + dt) - T(x, t)}{dt}$$



$$\frac{\partial^2 T}{\partial x^2}(x, t) \simeq \frac{T(x + dx, t) + T(x - dx, t) - 2T(x, t)}{dx^2}$$

- On choisit (par exemple) de calculer $T(x, t + dt)$ en fonction des températures aux temps précédents

$$T(x, t + dt) \simeq K \frac{dt}{dx^2} (T(x + dx, t) + T(x - dx, t) - 2T(x, t)) + T(x, t)$$

Dans le code, les températures au temps t et abscisse quelconque doivent être connues.

Tracé 2D

- Conditions initiales : il faut fixer une valeur pour K , ℓ , T_1 et T_2 . On doit connaître la température au temps t_0 (par exemple une constante ou une fonction ne dépendant que de x). On souhaite tracer l'évolution de la température le long de la barre sur la période $[t_0, t_{\max}]$.

Tracé 2D

- Conditions initiales : il faut fixer une valeur pour K , ℓ , T_1 et T_2 . On doit connaître la température au temps t_0 (par exemple une constante ou une fonction ne dépendant que de x). On souhaite tracer l'évolution de la température le long de la barre sur la période $[t_0, t_{\max}]$.
- Il faut fixer un pas de temps dt (exprimé par exemple en secondes). A un instant t fixé, les températures aux différentes abscisses sont stockées dans un tableau de N_x abscisses correspondant à un pas d'espace dx . Ce tableau est mis à jour toutes les dt secondes.

Tracé 2D

- Conditions initiales : il faut fixer une valeur pour K , ℓ , T_1 et T_2 . On doit connaître la température au temps t_0 (par exemple une constante ou une fonction ne dépendant que de x). On souhaite tracer l'évolution de la température le long de la barre sur la période $[t_0, t_{\max}]$.
- Il faut fixer un pas de temps dt (exprimé par exemple en secondes). A un instant t fixé, les températures aux différentes abscisses sont stockées dans un tableau de N_x abscisses correspondant à un pas d'espace dx . Ce tableau est mis à jour toutes les dt secondes.
- Toutes les Δ secondes (où Δ est un multiple de dt), on décide de tracer la courbe des températures en fonction de l'abscisse. On ne peut pas tracer cette courbe à tous les instants $t_0 + kdt$ car le graphique serait surchargé et illisible.

Tracé 2D

Application numérique

- Conditions initiales :

$K = 10^{-5} \text{m}^2 \text{s}^{-1}$, $\ell = 1 \text{m}$, $T_1 = 20^\circ \text{C}$, $T_2 = 100^\circ \text{C}$. Initialement la température est de T_1 tout au long de la barre mais T_2 sur une extrémité. On souhaite tracer l'évolution de la température le long de la barre durant 5h (soit 18000 secondes).

Tracé 2D

Application numérique

- Conditions initiales :

$K = 10^{-5} \text{m}^2 \text{s}^{-1}$, $\ell = 1 \text{m}$, $T_1 = 20^\circ \text{C}$, $T_2 = 100^\circ \text{C}$. Initialement la température est de T_1 tout au long de la barre mais T_2 sur une extrémité. On souhaite tracer l'évolution de la température le long de la barre durant 5h (soit 18000 secondes).

- Le temps est exprimé en secondes, le pas temporel est noté $dt = 1$. A un instant t , les températures aux différentes abscisses sont stockées dans un tableau de pas dx (exprimé en mètres),

Tracé 2D

Application numérique

- Conditions initiales :
 $K = 10^{-5} \text{m}^2 \text{s}^{-1}$, $\ell = 1 \text{m}$, $T_1 = 20^\circ \text{C}$, $T_2 = 100^\circ \text{C}$. Initialement la température est de T_1 tout au long de la barre mais T_2 sur une extrémité. On souhaite tracer l'évolution de la température le long de la barre durant 5h (soit 18000 secondes).
- Le temps est exprimé en secondes, le pas temporel est noté $dt = 1$. A un instant t , les températures aux différentes abscisses sont stockées dans un tableau de pas dx (exprimé en mètres),
- On met à jour les températures entre $t_{\min} = 0$ et $t_{\max} = 5\text{h}$. Toutes les 2000 secondes, on trace les températures en fonction des abscisses à l'instant t .

Tracé 2D

Application numérique

- Conditions initiales :
 $K = 10^{-5} \text{m}^2 \text{s}^{-1}$, $\ell = 1 \text{m}$, $T_1 = 20^\circ \text{C}$, $T_2 = 100^\circ \text{C}$. Initialement la température est de T_1 tout au long de la barre mais T_2 sur une extrémité. On souhaite tracer l'évolution de la température le long de la barre durant 5h (soit 18000 secondes).
- Le temps est exprimé en secondes, le pas temporel est noté $dt = 1$. A un instant t , les températures aux différentes abscisses sont stockées dans un tableau de pas dx (exprimé en mètres),
- On met à jour les températures entre $t_{\min} = 0$ et $t_{\max} = 5\text{h}$. Toutes les 2000 secondes, on trace les températures en fonction des abscisses à l'instant t .
- On choisit $N_x = 100$ abscisses entre 0 et 1 m. Le pas d'espace est donc $dx = \frac{1}{N_x - 1}$. Le long de la barre, la température est 20°C sauf à l'abscisse $x = 1$ où elle vaut 100°C .

Tracé 2D

Code : Initialisation

```
1 import numpy as np
2 K=1e-5 # coef diffusion
3 T1 , T2 = 20, 100 # températures aux extrémités
4 tmax = 5*3600 # 5heures
5 Nx = 100 #100 abscisses
6 x=np.linspace(0,1,Nx) # la subdivision régulière des abs
7 dx = 1/(Nx-1) # poteaux et intervalles...
8 T = T1*np.ones(Nx) # tableau des températures initiales
9 T[Nx-1] = T2 # source de chaleur
10 dt = 1 # pas de temps, 1s
```

- L'élément **T[k]** désigne la température au temps courant et à l'abscisse $0 + k \cdot dx$ pour $k \in \llbracket 0, N_x - 1 \rrbracket$.

Tracé 2D

Code : Initialisation

```

1 import numpy as np
2 K=1e-5 # coef diffusion
3 T1 , T2 = 20, 100 # températures aux extrémités
4 tmax = 5*3600 # 5heures
5 Nx = 100 #100 abscisses
6 x=np.linspace(0,1,Nx) # la subdivision régulière des abs
7 dx = 1/(Nx-1) # poteaux et intervalles...
8 T = T1*np.ones(Nx) # tableau des températures initiales
9 T[Nx-1] = T2 # source de chaleur
10 dt = 1 # pas de temps, 1s

```

- L'élément **$T[k]$** désigne la température au temps courant et à l'abscisse $0 + k \cdot dx$ pour $k \in \llbracket 0, N_x - 1 \rrbracket$.
- Au départ **$T[k]$** désigne $T(0 + k \cdot dx, 0)$.

Tracé 2D

Code : Initialisation

```

1 import numpy as np
2 K=1e-5 # coef diffusion
3 T1 , T2 = 20, 100 # températures aux extrémités
4 tmax = 5*3600 # 5heures
5 Nx = 100 #100 abscisses
6 x=np.linspace(0,1,Nx) # la subdivision régulière des abs
7 dx = 1/(Nx-1) # poteaux et intervalles...
8 T = T1*np.ones(Nx) # tableau des températures initiales
9 T[Nx-1] = T2 # source de chaleur
10 dt = 1 # pas de temps, 1s

```

- L'élément **T[k]** désigne la température au temps courant et à l'abscisse $0 + k \cdot dx$ pour $k \in \llbracket 0, N_x - 1 \rrbracket$.
- Au départ **T[k]** désigne $T(0 + k \cdot dx, 0)$.
- Le tableau des abscisses **x** ne sert qu'à l'affichage (paramètre de la fonction **plot**) pas aux calculs.

Construction des courbes

Rappel :

$$T(x, t + dt) \simeq K \frac{dt}{dx^2} (T(x + dx, t) + T(x - dx, t) - 2T(x, t)) + T(x, t) (*)$$

Calcul (sans mémorisation) du tableau des températures pour chaque instant t en fonction des abscisses. Un tracé toutes les 2000 secondes.

```

1 import matplotlib.pyplot as plt
2 c = K*dt/dx**2 # la constante de l'équation (*)
3 def les_courbes():
4     for i in range(tmax):#les temps
5         t=np.zeros(Nx) # tableau températures à t_i + dt
6         for k in range(1,Nx-1):#les abscisses
7             t[k]=T[k]+c*(T[k+1]+T[k-1]-2*T[k])
8         t[0],t[Nx-1]=T1,T2 # sources chaleur aux extrémités
9         T=t # mise à jour de T
10        if i%2000 == 0: # un tracé toutes les 2000 secondes
11            plt.plot(x,T,label="t={}".format(i))

```

Code

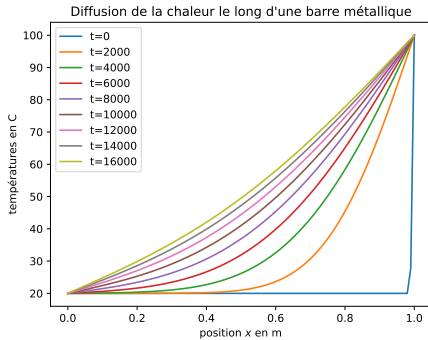
Décoration du graphique

On écrit des indications sur les axes (**xlabel** et **ylabel**), on fixe un titre (**title**) et on affiche les légendes (**label**) avant d'afficher le graphique (**show**) :

```
1 #initialisation, puis
2 les_courbes()
3
4 plt.xlabel('position $x$ en m') # légende sur axe des x
5 plt.ylabel('températures en C') # légende axe des y
6 plt.title('Diffusion de chaleur sur barre métallique')
7 plt.legend() # affichage des légendes
8 plt.show()
```

Graphique

Figure – Graphique obtenu



Surface et **meshgrid**

- On se place dans un repère orthonormé direct à 3 axes désignant abcsisse, temps et températures.

Surface et **meshgrid**

- On se place dans un repère orthonormé direct à 3 axes désignant abscisse, temps et températures.
- On veut tracer l'ensemble des points $(x, t, T(x, t))$, les triplets abscisse, temps, température. Il s'agit d'une surface.

Surface et **meshgrid**

- On se place dans un repère orthonormé direct à 3 axes désignant abscisse, temps et températures.
- On veut tracer l'ensemble des points $(x, t, T(x, t))$, les triplets abscisse, temps, température. Il s'agit d'une surface.
- On constitue d'abord un *maillage*, c'est à dire un ensemble de points $(x, t, 0)$ du plan horizontal $T = 0$.

Surface et **meshgrid**

- On se place dans un repère orthonormé direct à 3 axes désignant abscisse, temps et températures.
- On veut tracer l'ensemble des points $(x, t, T(x, t))$, les triplets abscisse, temps, température. Il s'agit d'une surface.
- On constitue d'abord un *maillage*, c'est à dire un ensemble de points $(x, t, 0)$ du plan horizontal $T = 0$.
- A la verticale de chacun de ces points $(x, t, 0)$, on place le point de coordonnée $(x, t, T(x, t))$.

Comprendre **meshgrid**

- Pour construire le maillage, c'est la fonction **meshgrid** qui est conseillée.

Comprendre **meshgrid**

- Pour construire le maillage, c'est la fonction **meshgrid** qui est conseillée.

```
1 a=np.linspace(0,3,num=4)#0,1,2,3
2 b=np.linspace(5,7,num=3)#5,6,7
3 A,B=np.meshgrid(a,b)
4 print(A) # a est dupliqué sur |b| lignes
5 print(B) # b est dupliqué sur |a| colonnes
```

Comprendre **meshgrid**

- Pour construire le maillage, c'est la fonction **meshgrid** qui est conseillée.

```
1 a=np.linspace(0,3,num=4)#0,1,2,3
2 b=np.linspace(5,7,num=3)#5,6,7
3 A,B=np.meshgrid(a,b)
4 print(A) # a est dupliqué sur |b| lignes
5 print(B) # b est dupliqué sur |a| colonnes
```

- On obtient

```
[[0.  1.  2.  3.]
 [0.  1.  2.  3.]
 [0.  1.  2.  3.]]
[[5.  5.  5.  5.]
 [6.  6.  6.  6.]
 [7.  7.  7.  7.]]
```

Comprendre **meshgrid**

```
[[0.  1.  2.  3.]  
 [0.  1.  2.  3.]  
 [0.  1.  2.  3.]]  
[[5.  5.  5.  5.]  
 [6.  6.  6.  6.]  
 [7.  7.  7.  7.]]
```

Comprendre **meshgrid**



```
[[0.  1.  2.  3.]  
 [0.  1.  2.  3.]  
 [0.  1.  2.  3.]]  
[[5.  5.  5.  5.]  
 [6.  6.  6.  6.]  
 [7.  7.  7.  7.]]
```

- Pour construire le maillage, on prend un coefficient $a_{i,j}$ de la première matrice et le coefficient correspondant $b_{i,j}$. On en déduit le point de coordonnées $(a_{i,j}, b_{i,j}, 0)$.

Comprendre **meshgrid**



```
[[0.  1.  2.  3.]  
 [0.  1.  2.  3.]  
 [0.  1.  2.  3.]]  
[[5.  5.  5.  5.]  
 [6.  6.  6.  6.]  
 [7.  7.  7.  7.]]
```

- Pour construire le maillage, on prend un coefficient $a_{i,j}$ de la première matrice et le coefficient correspondant $b_{i,j}$. On en déduit le point de coordonnées $(a_{i,j}, b_{i,j}, 0)$.
- Pour construire la surface dans l'espace, la fonction **plot_surface(a,b,X)** prend en paramètre une troisième matrice X de mêmes dimensions que a, b et place sur le graphique le point de coordonnées $(a_{i,j}, b_{i,j}, X_{i,j})$.

Tracé 3D

- On importe le constructeur d'axes 3D. Il prend en paramètre une figure vide et retourne un objet modélisant les axes en 3 dimensions.

Tracé 3D

- On importe le constructeur d'axes 3D. Il prend en paramètre une figure vide et retourne un objet modélisant les axes en 3 dimensions.
- En supposant déclarées 3 matrices X, Y, Z de même taille, on construit la surface des points $(X_{i,j}, Y_{i,j}, Z_{i,j})$ par :

```
1 from mpl_toolkits.mplot3d import Axes3D
2 ax=Axes3D(plt.figure())
3 ax.plot_surface(X,Y,Z)
4 plt.show()
```

Retour à l'équation de la chaleur

- Contrairement au tracé 2D, on mémorise les tableaux des températures à t fixé mais x variable. On obtient un tableau 2D dont les lignes donnent les températures à t fixé mais x variable et les colonnes les températures à t variable mais x fixé.

Retour à l'équation de la chaleur

- Contrairement au tracé 2D, on mémorise les tableaux des températures à t fixé mais x variable. On obtient un tableau 2D dont les lignes donnent les températures à t fixé mais x variable et les colonnes les températures à t variable mais x fixé.
- On construit le tableau des temps. Entre 0 et $t_{\max}$: il y a $t_{\max} + 1$ temps.

Retour à l'équation de la chaleur

- Contrairement au tracé 2D, on mémorise les tableaux des températures à t fixé mais x variable. On obtient un tableau 2D dont les lignes donnent les températures à t fixé mais x variable et les colonnes les températures à t variable mais x fixé.
- On construit le tableau des temps. Entre 0 et $t_{\max}$: il y a $t_{\max} + 1$ temps.

```

1 K=1e-5#coef diffusion
2 T1,T2,tmax=20, 100,5*3600
3 Nx=100 # 100 abscisses
4 dx, dt , c=1/(Nx-1), 1, K*dt/dx**2
5 Nt = tmax+1 # nombre de temps
6 Z=np.empty((Nx,Nt)) # le tableau à 2 dim. des F(x,t)
7 Z[:,0]=T1*np.ones(Nx) # 1ere col, temp. à t=0
8 Z[Nx-1,0]=T2 # source de chaleur
9 lest = np.linspace(0,tmax,Nt) # tableau des temps
10 lesx = np.linspace(0,1,Nx) # tableau des abscisses

```

Retour à l'équation de la chaleur

- On remplit le tableau Z des températures

```
1 for t in range(Nt-1):  
2     Z[0,t+1],Z[Nx-1,t+1]=T1,T2  
3     for x in range(1,Nx-1):  
4         Z[x,t+1]=c*(Z[x+1,t]+Z[x-1,t]-2*Z[x,t])+Z[x,t]
```

Retour à l'équation de la chaleur

- On remplit le tableau Z des températures

```
1 for t in range(Nt-1):
2     Z[0,t+1],Z[Nx-1,t+1]=T1,T2
3     for x in range(1,Nx-1):
4         Z[x,t+1]=c*(Z[x+1,t]+Z[x-1,t]-2*Z[x,t])+Z[x,t]
```

- On trace la figure en ne gardant qu'un temps sur 60 (donc une colonne sur 60 pour Z)

```
1 from mpl_toolkits.mplot3d import Axes3D
2 Z=Z[:,::60] #on garde une colonne sur 60
3 lest=lest[:,::60] # on garde un temps sur 60
4 # 2 matrices à |lesx| lignes et |lest| colonnes
5 a,b=np.meshgrid(lest,lesx)
6 ax = Axes3D(plt.figure())
7 ax.plot_surface(a,b,Z)
8 plt.show()
```

Graphique

Figure – Graphique obtenu