Thème: Enjeux sociétaux: Environnement, sécurité, énergie

PROPAGATION ATMOSPHÉRIQUE D'UN POLLUANT

TIPE: 2020/2021

Numéro d'inscription: 49649

Plan du travail

1. Modélisation physique
2. Analyse mathématique du modèle à une dimension (1D)
3. Schémas numériques (Euler explicite)
4. Résultats de simulation (Python)

1. Modélisation physique:

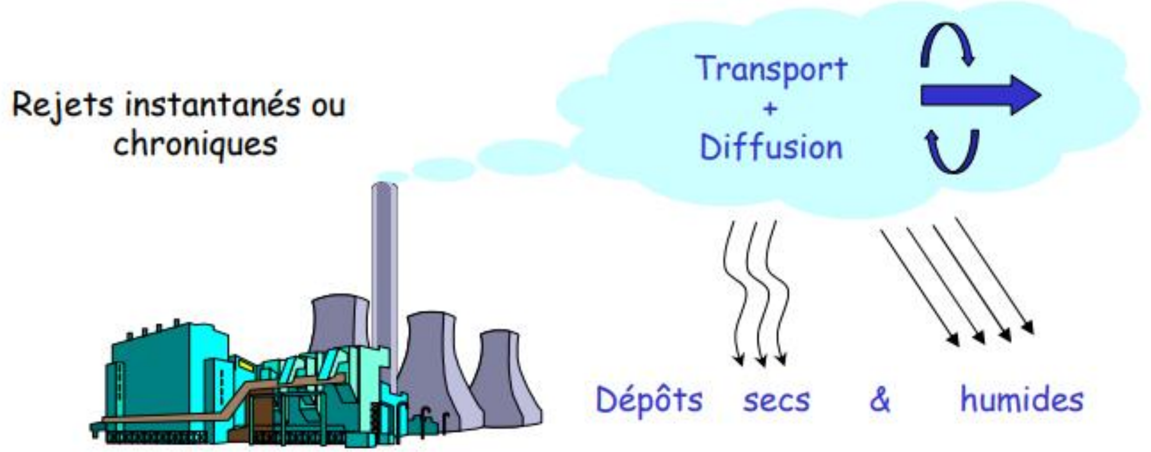
Les polluants primaires, comme les NOx, le SO_2, le CO, les poussières et les Composés Organiques Volatils (COV), sont directement émis dans l'atmosphère. L'existence de ces polluants dans l'atmosphère est rythmée par cinq étapes.

On s'intéresse uniquement aux **mécanismes** réalisant la deuxième étape (qui est la dispersion):

- **La diffusion**
- **Le transport par le vent (la convection)**

Approximations:

- **Le polluant est constitué d'un seul composant chimique . On n'a pas de réaction chimique.**



2. Analyse mathématique du modèle en dimension 1 :

On note Ω un ouvert borné de $I\!R^d, d = 1,2$ ou 3, T un réel, $T > 0$, $\vec{u}(x)$ la vitesse de l'air au point $x, x \in \Omega$, et $c(x,t)$ la concentration au point x et au temps t.

1.1 Modèle de diffusion pure:

✓ **On *considère* le problème de la dispersion d'un polluant dans l'air par diffusion pure donc on néglige la convection (le terme de convection u=0).**

✓ **On suppose que la vitesse est indépendante du temps.**

1.1 Modèle de diffusion pure:

Loi de Fick: $\overrightarrow{j_n} = -D\overrightarrow{\nabla}c$

D désigne le coefficient de diffusion (en m^2s^{-1})

c: est la concentration molaire (en $mol.m^{-3}$)

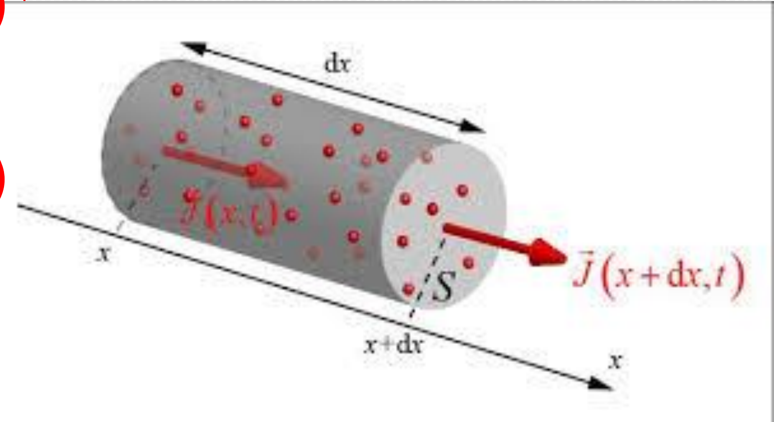
$\overrightarrow{j_n}$: Vecteur densité de courant de particules(en $m^{-2}s^{-1}$)

$\Rightarrow$ D'après la loi de Fick, l'équation de diffusion de ce modèle dans le cas unidimensionnel (1d) est alors:

$$\frac{\partial c}{\partial t}(x,t) - \frac{\partial^2 c}{\partial x^2}(x,t) = 0 \quad x \in]0,1[, t \in]0,T[\quad \textbf{(1)}$$

$$c(0,t) = c(1,t) = 0 \qquad t \in [0,T[\quad \textbf{(2)}$$

$$c(x,0) = c_0(x) \quad x \in [0,1]; c_0(x) \text{ donné} \quad \textbf{(3)}$$



En supposant que D= 1

Quelques valeurs de coefficient de diffusion:

Molécule	Nombre de masse A	$D\ [\mathrm{m^2 s^{-1}}]$
Diffusion dans l'air à 0°C		
di-hygrogène H_2	2	$6,3.10^{-5}$
di-oxygène O_2	32	$1,8.10^{-5}$
Diffusion dans l'eau à 15°C		
eau H_20	18	$2,0.10^{-9}$
di-oxygène O_2	32	$1,0.10^{-9}$
glucose $C_6H_{12}O_6$	180	$6,7.10^{-10}$
hémoglobine	68000	$6,9.10^{-11}$

a.1) Remarque sur l'équation de diffusion d'un polluant:

Unicité de la solution Soient c_1 et c_2 deux solutions, on pose: $c = c_1 - c_2$.

$c(x,0) = 0$ pour tout $x \in [0,1]$.

En multipliant **(1)** par c et en intégrant sur [0,1]

$$\int_0^1 \frac{\partial c}{\partial t}(x,t)c(x,t)dx - \int_0^1 \frac{\partial^2 c}{\partial x^2}(x,t)c(x,t)dx = 0 \text{ pour tout } t \in]0,T[$$

À l'aide d'une intégration par parties et des conditions aux bords, on obtient:

$$\int_0^1 \frac{\partial c}{\partial t}(x,t)c(x,t)dx + \int_0^1 (\frac{\partial c}{\partial x}(x,t))^2 dx = 0 \text{ pour tout } t \in]0,T[$$

a.2) Remarque sur l'équation de diffusion d'un polluant:

Soit:

$$\frac{1}{2}\frac{d}{dt}\int_0^1 c^2(x,t)dx \leq 0 \text{ pour tout } t \in]0,T[$$

Donc $E(t) = \int_0^1 c^2(x,t)dx$ est une fonction décroissante et positive. De plus, $E(0) = 0;$

➔ **On finit par conclure que:**

$$c(x,t) = 0 \quad \text{pour tout } x \in [0,1]$$

Alors $c_1(x,t)=c_2(x,t)$
Ce qui traduit que la solution de l'équation de diffusion est <u>unique.</u>

▪ Expression de la solution à l'aide de la méthode de séparation de variables:

• **Les deux cas <u>λ > 0</u> et <u>λ = 0</u> sont impossibles d'après les conditions aux bords (2)**

• **Cas où <u>λ < 0</u> :**

On pose: $\lambda = -a^2$ et $\psi''(x) + a^2\psi(x) = 0.$

$\psi(x) = Acos(ax) + Bsin(ax)$. D'après les conditions aux bords (5): $A = 0$ et $Bsin(a) = 0$, d'où $a = k\pi, k \in Z$. (Le cas $B = 0$ est exclu pour les mêmes raisons que précédemment)

➔ **La solution de l'équation est:**

$$\boxed{c(x,t) = A_k \exp(-k^2\pi^2 t) sin(k\pi x), k \in Z, A_k \in I\!R}$$

1.2 Modèle de convection pure:

On considère à présent le cas de la convection pure unidimensionnel (1d).

On suppose que la vitesse u est constante et que c(x,t) ne diffuse pas et est uniquement transportée par le fluide.

➔ L'équation de ce modèle est :

(4)
$$\boxed{\frac{\partial c}{\partial t}(x,t) + u\frac{\partial c}{\partial x}(x,t) = 0,\ x \in]0,1[, t \in]0,T[}$$

: C'est l'équation de transport

Avec la condition initiale:

(5)
$$\boxed{c(x,0) = c_0(x) x \in [0,1]; c_0(x) \text{ donné}}$$

1.2 Modèle de convection pure:

La solution de (4)-(5) est:

$$c(x,t) = c_0(x - ut), x \in]0,1[, t \in [0,T[$$

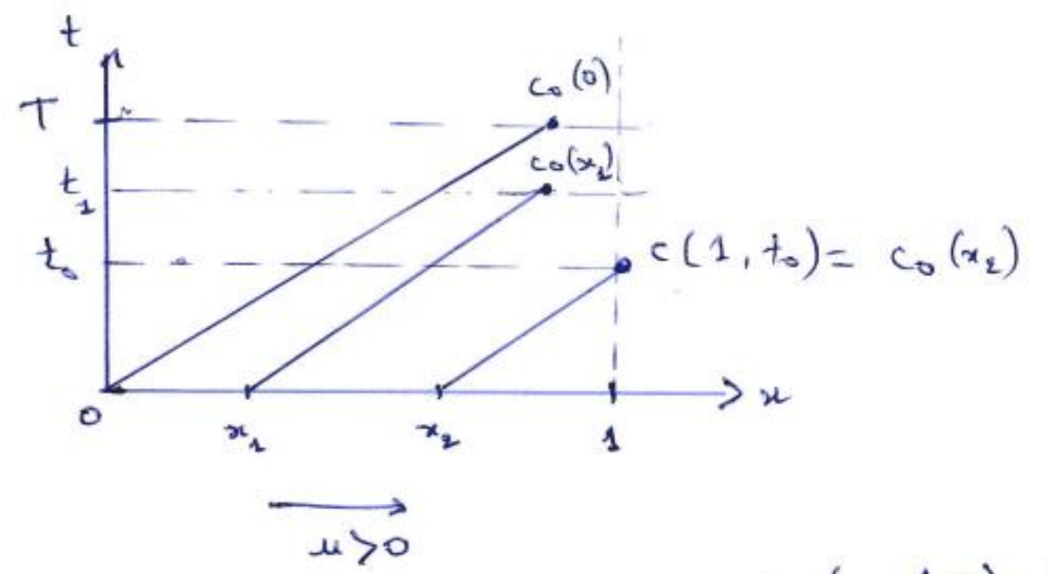
➔ **Nous remarquons alors que la concentration est effectivement transportée selon le champ de vitesse u (voir la figure de la solution)**

Traçage de la solution de l'équation de transport (1D):

Puisque le champ de vitesse u est constant, les trajectoires dans l'espace (x,t) sont des droites de pente u^{-1}:

$x - ut = \text{constante}$

On a tracé ses trajectoires (pour le cas où $u > 0$):



Solution de l'équation de transport (cas 1D) ($u > 0$).

N.B: **La condition au bord qui est associée à l'équation de transport dépend du signe de la vitesse u :**

$$\begin{cases} c(0,t) \text{ donné} & si \quad u = \text{constante} > 0 \\ c(1,t) \text{ donné} & si \quad u = \text{constante} < 0 \end{cases}$$

Comparaison entre le cas de diffusion pure et le cas de convection pure:

L a concentration c(x,t) s'amortit au cours du temps dans le cas d'une diffusion pure alors que dans le cas d'une convection pure c(x,t) ne l'est pas.

1.3 Modèle de convection-diffusion:

On considère maintenant le modèle de convection-diffusion unidimensionnel (1d) avec la vitesse u constante.

➔ **Les équations de ce modèle sont :**

$$\frac{\partial c}{\partial t}(x,t) - \sigma \frac{\partial^2 c}{\partial x^2}(x,t) + u \frac{\partial c}{\partial x}(x,t) = 0,\ x \in]0,1[, t \in]0,T[$$

avec la condition initiale:

$$c(x,0) = c_0(x) x \in [0,1]; c_0(x) \text{ donné}$$

et une condition sur tout le bord:

$$c(0,t) = c(1,t) = 0,\ t \in [0,T[$$

3. Schémas numériques:

Méthodes des différences finies: (Schéma d'Euler explicite)

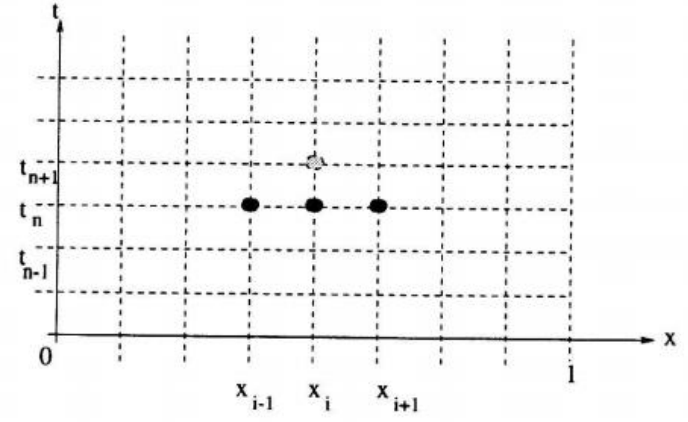
On note c(x,t) la solution exacte du modèle.

On discrétise la géométrie spatiale et le temps ainsi on pose:

$$x_i = i\Delta x, \Delta x = \frac{1}{N} \text{ et } t_n = n\Delta t, \Delta t = \frac{T}{M}.$$

soit $h = \Delta x$ le pas d'espace et $k = \Delta t$ le pas de temps. Le point (x_i, t_n), $0 \leq i \leq N, 0 \leq n \leq M$ est un point du « maillage »,

c_i^n *l'approximation de* $c(x_i, t_n)$.



Cas du modèle de diffusion pure unidimensionnel:

✓ Pour le terme en temps, le développement de Taylor suivant:

$$c(x_i,t_{n+1}) = c(x_i,t_n) + k\frac{\partial c}{\partial t}(x_i,t_n) + \frac{k^2}{2}\frac{\partial^2 c}{\partial t^2}(x_i,t_n) + \ldots$$

Et donc:

$$\frac{\partial c}{\partial t}(x_i,t_n) = \frac{c(x_i,t_{n+1}) - c(x_i,t_n)}{k} + O(k).$$

De même, on montre que:

$$\frac{\partial^2 c}{\partial x^2}(x_i,t_n) = \frac{c(x_{i+1},t_n) - 2c(x_i,t_n) + c(x_{i-1},t_n)}{h^2} + O(h^2)$$

En injectant ces deux expressions dans l'équation de diffusion on obtient :

$$\frac{c(x_i,t_{n+1}) - c(x_i,t_n)}{k} - \frac{c(x_{i+1},t_n) - 2c(x_i,t_n) + c(x_{i-1},t_n)}{h^2} = O(k) + O(h^2)$$

pour $1 \leq i \leq N-1, 1 \leq n \leq M-1$.

Cas du modèle de diffusion pure unidimensionnel:

On en déduit alors le schéma numérique:

$$\begin{cases} \dfrac{c_i^{n+1} - c_i^n}{k} - \dfrac{c_{i+1}^n - 2c_i^n + c_{i-1}^n}{h^2} = 0 \\ \text{pour } 1 \le i \le N-1,\ 1 \le n \le M-1 \\ c_0^n = c_N^n = 0,\ 0 \le n \le M \quad \text{: Conditions aux limites} \\ c_i^0 = c_0(x_i) 0 \le i \le N \quad \text{: Condition initiale} \end{cases}$$

$$\Rightarrow \quad c_i^{n+1} = c_i^n + \frac{k}{h^2}(c_{i+1}^n - 2c_i^n + c_{i-1}^n) 1 \le i \le N-1, n = 1,2,3,\ldots$$

Cas du modèle de convection pure unidimensionnel:

On pose la vitesse u constante et strictement positive.
Pour approcher le terme en espace, le développement de Taylor donne:

$$c(x_{i+1},t_n) = c(x_i,t_n) + h\frac{\partial c}{\partial x}(x_i,t_n) + \frac{h^2}{2}\frac{\partial^2 c}{\partial x^2}(x_i,t_n) + \ldots$$

ce qui donne:

$$\frac{\partial c}{\partial x}(x_i,t_n) = \frac{c(x_{i+1},t_n) - c(x_i,t_n)}{h} + O(h)$$

Notons que nous avons également:

$$\frac{\partial c}{\partial x}(x_i,t_n) = \frac{c(x_i,t_n) - c(x_{i-1},t_n)}{h} + O(h)$$

Nous obtenons dans ce cas, la formule aux différences finies décentrée suivante:

$$\boxed{\frac{c(x_i,t_n) - c(x_{i-1},t_n)}{h} \approx \frac{\partial c}{\partial x}(x_i,t_n)}$$

➔ Avec un tel choix de décentrage, nous obtenons le schéma numérique suivant:

$$\begin{cases} \dfrac{c_i^{n+1} - c_i^n}{k} + u\dfrac{c_i^n - c_{i-1}^n}{h} = 0 \\ \text{pour } 1 \le i \le N-1, n=1,2,3,\ldots \\ c_0^n = 0, n \ge 0 \quad \text{: Conditions aux limites} \\ c_i^0 = c_0(x_i), 0 \le i \le N \quad \text{: Condition initiale} \end{cases}$$

N.B: Par ailleurs, ce schéma présente l'inconvénient (mineur) suivant: il introduit de la diffusion artificielle. En effet, nous avons:

$$u\frac{c_i^n - c_{i-1}^n}{h} = u\left(\frac{c_{i+1}^n - c_{i-1}^n}{2h} - \frac{h}{2}\frac{c_{i+1}^n - 2c_i^n + c_{i-1}^n}{h^2}\right)$$

Si on injecte le membre à droite de cette équation dans le schéma numérique, ce dernier devient équivalent au schéma centré pour l'équation de convection-diffusion avec comme coefficient de diffusivité : $\boxed{\frac{h}{2}u}$

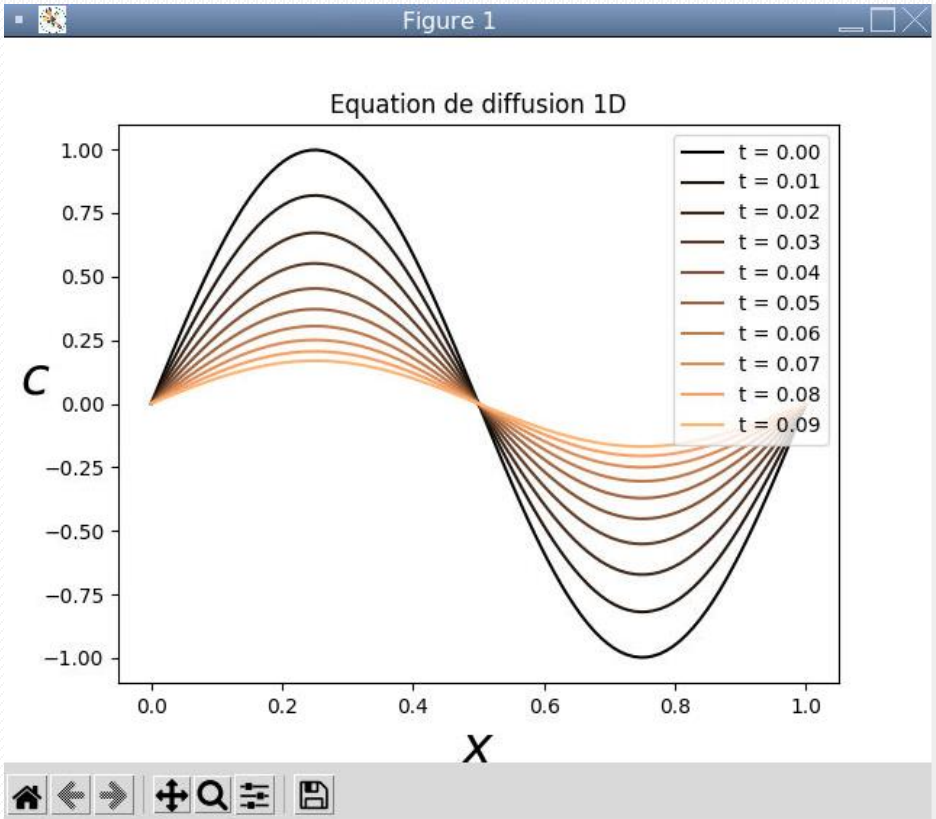
Extension au cas de la convection diffusion 2D:

Soit:

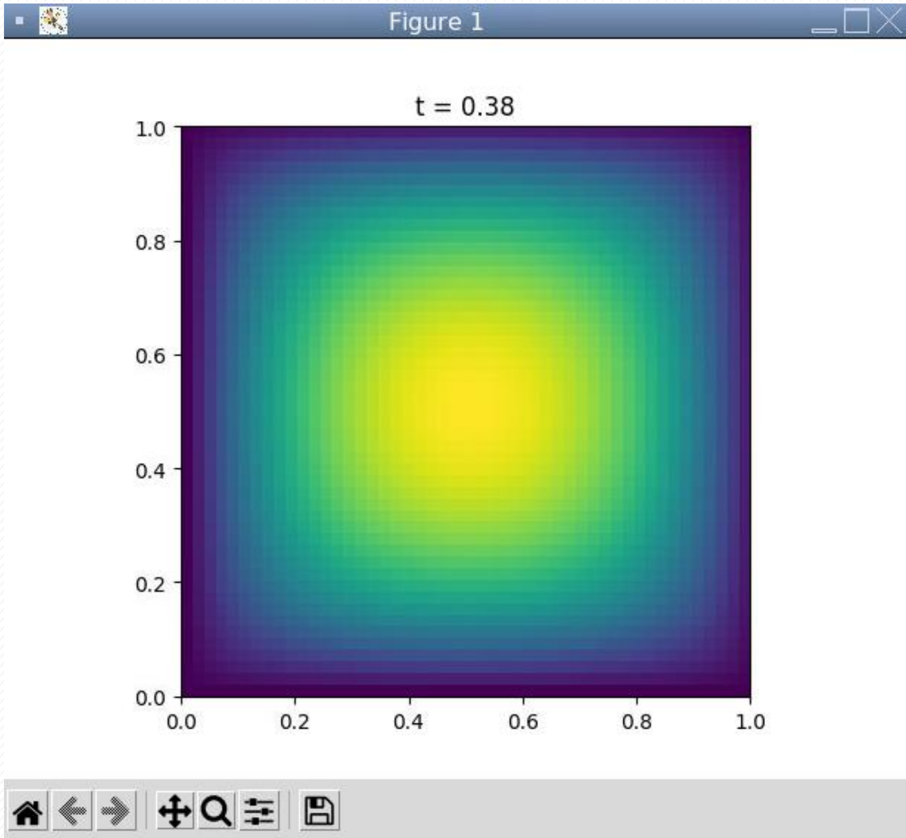
$d = 2, \Omega =]0,1[\times]0,1[$; $\vec{u} = (u_1, u_2)^T$, u_1 constante positive et $u_2(x,y) \leq 0, \forall(x,y) \in \Omega$. Alors, nous obtenons le schéma numérique suivant:

$$
\left\{
\begin{array}{l}
\dfrac{c_{i,j}^{n+1} - c_{i,j}^{n}}{\Delta t} - \sigma\left[\dfrac{c_{i+1,j}^{n} - 2c_{i,j}^{n} + c_{i-1,j}^{n}}{\Delta x^2} + \dfrac{c_{i,j+1}^{n} - 2c_{i,j}^{n} + c_{i,j-1}^{n}}{\Delta y^2}\right] \\
+u_1 \dfrac{c_{i,j}^{n} - c_{i-1,j}^{n}}{\Delta x} + u_2(x_i, y_j)\dfrac{c_{i,j+1}^{n} - c_{i,j}^{n}}{\Delta y} = 0 \\
\text{pour } i=1,\ldots,N-1, j=1,\ldots,L-1, n=0,1,2,\ldots \\
c_{0,j}^{n} = c_{N,j}^{n} = c_{i,0}^{n} = c_{i,L}^{n} = 0, \; i=0,..,N, j=0\ldots L, n=0,1,2,\ldots \\
c_{i,j}^{0} = c_0(x_i, y_j) \; i=1,..,N-1, j=1\ldots L-1
\end{array}
\right.
$$

4. Résultats de simulation: (python)



Simulation de l' équation de diffusion pure 1D (code 1)



Simulation de l'équation de convection-diffusion 2D (code 2)

ANNEXES:

Code 1

tipe.py - C:\Users\DELL\Desktop\tipe.py (3.7.4)

File Edit Format Run Options Window Help

```python
import numpy as np
import matplotlib.pyplot as plt
# PHYSICAL PARAMETERS
K = 0.5      #Diffusion coefficient
L = 1.0      #Domain size
Time = 0.1   #Integration time
# NUMERICAL PARAMETERS
NX = 100     #Number of grid points
NT = 1000    #Number of time steps
dx = L/(NX-1)   #Grid step (space)
dt = Time/NT    #Grid step (time)
### MAIN PROGRAM ###
# Initialisation
x = np.linspace(0.0,1.0,NX)
c = np.sin(2*np.pi*x)
RHS = np.zeros((NX))
plt.figure()
# Main loop
for n in range(0,NT):
   for j in range (1, NX-1):
      RHS[j] = dt*K*(c[j-1]-2*c[j]+c[j+1])/(dx**2)
   for j in range (1, NX-1):
      c[j] += RHS[j]
#Plot every 100 time steps
   if (n%100 == 0):
      plotlabel = "t = %1.2f" %(n * dt)
      plt.plot(x,c, label=plotlabel,color = plt.get_cmap('copper')(float(n)/NT))
plt.xlabel(u'$x$', fontsize=26)
plt.ylabel(u'$c$', fontsize=26, rotation=0)
plt.title(u'Equation de diffusion 1D')
plt.legend()
plt.show()
```

Code 2

diffusion.py - C:\Users\DELL\Desktop\diffusion.py (3.7.4)

File Edit Format Run Options Window Help

```python
import numpy as np
import matplotlib.pyplot as plt
if 'qt' in plt.get_backend().lower():
    try:
        from PyQt4 import QtGui
    except ImportError:
        from PySide import QtGui
# PHYSICAL PARAMETERS
sigma = 0.5    #Diffusion coefficient
Lx = 1.0   #Domain size x
Ly = 1.0   #Domain size y
Time = 0.4 #Integration time
# NUMERICAL PARAMETERS
NT = 2000      #Number of time steps
NX = 50        #Number of grid points in x
NY = 50        #Number of grid points in y
dt = Time/NT   #Grid step (time)
dx = Lx/(NX-1) #Grid step in x (space)
dy = Ly/(NY-1) #Grid step in y (space)
xx = np.linspace(0,Lx,NX)
yy = np.linspace(0,Ly,NY)
plt.ion()
plt.figure()
### MAIN PROGRAM ###
c = np.zeros((NX,NY))
RHS = np.zeros((NX,NY))
# Main loop
for n in range(0,NT):
   RHS[1:-1,1:-1] = dt*( (c[:-2,1:-1]-2*c[1:-1,1:-1]+c[2:,1:-1])/(dx**2)  \
                       + (c[1:-1,:-2]-2*c[1:-1,1:-1]+c[1:-1,2:])/(dy**2) )
   c[1:-1,1:-1] += RHS[1:-1,1:-1]
#Plot every 100 time steps
   if (n%100 == 0):
      plotlabel = "t = %1.2f" %(n * dt)
```

Suite code 2 :

```
### MAIN PROGRAM ###
c = np.zeros((NX,NY))
RHS = np.zeros((NX,NY))
# Main loop
for n in range(0,NT):
   RHS[1:-1,1:-1] = dt*( (c[:-2,1:-1]-2*c[1:-1,1:-1]+c[2:,1:-1])/(dx**2)  \
                      + (c[1:-1,:-2]-2*c[1:-1,1:-1]+c[1:-1,2:])/(dy**2) )
   c[1:-1,1:-1] += RHS[1:-1,1:-1]
#Plot every 100 time steps
   if (n%100 == 0):
      plotlabel = "t = %1.2f" %(n * dt)
      plt.pcolormesh(xx,yy,c, shading='flat')
      plt.title(plotlabel)
      plt.axis('image')
      plt.draw()
      if 'qt' in plt.get_backend().lower():
          QtGui.qApp.processEvents()
plt.show()
```