Exercice6_SDZ

November 26, 2021

[2]:
```
import numpy as np
import matplotlib.pyplot as plt
```

Problème 6. Discretisez l'équation

$$-\mu u''(x) + u'(x) = 1$$

avec les conditions au bord Dirichlet homogènes $u(0) = u(1) = 0$ et le coefficient de diffusion $\mu = 0.04$ Utilisez les différences finies (stencil à 3 points), avec la discrétisation décentrée pour le terme $u'(x)$ et avec la taille de maille constante $h_a = 0.25$.

Solution: la solution analytique est $u(x) = x - \frac{1-e^{\frac{x}{\mu}}}{1-e^{\frac{1}{\mu}}}$ *et la solution par les différences finies est* $\vec{u} = (0.2477, 0.4813, 0.6124)^T$.

On commence par poser les équations

$$-\mu u''(x) + u'(x) = 1 \qquad u(0) = u(1) = 0$$

On discrétise avec des différences centrées à 3 points pour u'' et avec des différences rétrogrades d'ordre $n = 1$ pour u'

$$u''(x) \approx \frac{u(x+h) - 2u(x) + u(x+h)}{h^2} \qquad u'(x) = \frac{u(x) - u(x-h)}{h}$$

Si on mets tout ensemble on a

$$\frac{1}{h^2}\left(-\mu u(x-h) + 2\mu u(x) - \mu u(x+h) + hu(x) - hu(x-h)\right) = 1$$

$$\frac{1}{h^2}\left((-\mu - h)u(x-h) + (2\mu + h)u(x) - \mu u(x+h)\right)$$

On peut l'écrire sous forme de matrice

$$\frac{1}{h^2}\begin{pmatrix} 2\mu + h & -\mu & 0 \\ -\mu - h & 2\mu + h & -\mu \\ 0 & -\mu - h & 2\mu + h \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} 1 + \frac{\mu + h}{h^2}u(0) \\ 1 \\ 1 + \frac{\mu}{h^2}u(1) \end{pmatrix}$$

Comme $u(0) = u(1) = 0$

$$\frac{1}{h^2}\begin{pmatrix} 2\mu + h & -\mu & 0 \\ -\mu - h & 2\mu + h & -\mu \\ 0 & -\mu - h & 2\mu + h \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

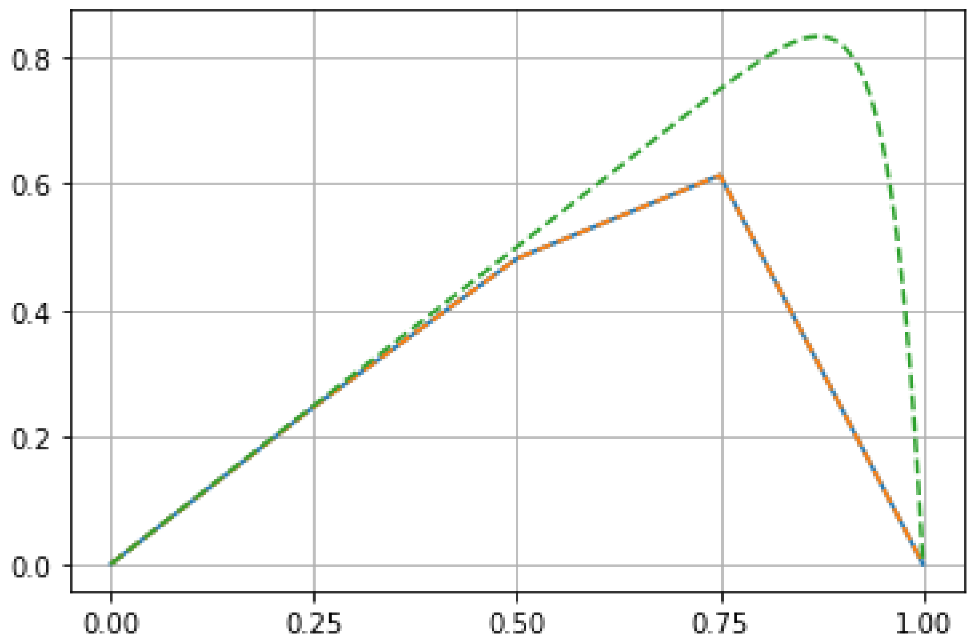
Aucune idée comment le résoudre à la main… c'est bien compliqué… Mais sur Python ça marche très bien

[8]:

```
h = 0.25
mu = 0.04
ustart = 0
uend = 0

uth = np.array([ustart, 0.2477, 0.4813, 0.6124, uend])
xth = np.linspace(0, 1, 5)
def analytic(x):
    return x - (1-np.exp(x/mu))/(1-np.exp(1/mu))

N = int(1/h)+1
x = np.linspace(0, 1, N)
xos = np.linspace(0, 1, 1000)

A = np.zeros([N-2,N-2])
A += np.diagflat(np.ones(N-2)*(2*mu+h))
A += np.diagflat(np.ones(N-3)*(-mu), 1)
A += np.diagflat(np.ones(N-3)*(-mu-h), -1)
A = 1/h**2 * np.matrix(A)


F = np.ones(N-2)
F[0] += (mu+h)/h**2*ustart
F[-1] += mu/h**2*uend

u = np.zeros(N)
u[1:-1] = A.I @ F


plt.xticks(x)
plt.grid()
plt.plot(x, u)
plt.plot(xth, uth, '--')
plt.plot(xos, analytic(xos), '--')
```

[8]:

```
[<matplotlib.lines.Line2D at 0x1ec62c97e20>]
```


0.8
0.6
0.4
0.2
0.0
0.00
0.25
0.50
0.75
1.00