

Exercice4_SDZ

November 26, 2021

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

0.1 Exercice 4

Problème 4. Discretisez l'équation

$$-u''(x) = 4(u(x) - x) \quad \text{sur } (0, 1)$$

avec des conditions au bord

$$u(0) = 0 \quad \text{et} \quad u(1) = 2.$$

Utilisez les différences finies (stencil à 3 points), avec la taille de maille constante

a) $h_a = 1/3$, et

b) $h_b = 1/4$, et

Solution: la solution analytique est $u(x) = \frac{e^2}{e^4-1} (e^{2x} - e^{-2x}) + x$ et la solution approchée par des différences finies est:

- $\vec{u}_a = (0.5343, 1.1579)^T \approx (u(h_a), u(2h_a))^T$,
- $\vec{u}_b = (0.3951, 0.8265, 1.3395)^T \approx (u(h_b), u(2h_b), u(3h_b))^T$.

On commence par poser les équations

$$-u''(x) = 4(u(x) - x) \quad u(0) = 0 \quad u(1) = 2$$

$$u''(x) = -4u(x) + 4x$$

On discrétise $u''(x)$

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

On a donc

$$\frac{u(x-h) - 2u(x) + u(x+h)}{h^2} = -4u(x) + 4x$$

Qu'on peut transformer en

$$\frac{u(x-h) - 2u(x) + u(x+h)}{h^2} + 4u(x) = \underbrace{4x}_{f(x)}$$

On peut ensuite combiner le $4u(x)$ dans l'expression pour $u''(x)$

$$\frac{u(x-h) - 2u(x) + 4h^2u(x) + u(x+h)}{h^2} = \underbrace{4x}_{f(x)}$$

Il n'y a plus qu'à construire la matrice A (exemple avec $h = \frac{1}{4}$)

$$A = \begin{pmatrix} -2 + 4h^2 & 1 & 0 \\ 1 & -2 + 4h^2 & 1 \\ 0 & 1 & -2 + 4h^2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} + \frac{1}{h^2} \begin{pmatrix} -0 \\ 0 \\ -2 \end{pmatrix}$$

On résout ensuite, soit à la main, soit en utilisant l'inverse de A

$$\vec{u} = A^{-1}\vec{F}$$

```
[90]: uend = 2
      ustart = 0

      def f(x):
          return 4*x

      def analytic(x):
          #return np.exp(2)/(np.exp(4)-1)*(np.exp(2*x)-np.exp(-2*x))+x # Solution
          ↪fausse du prof
          return x + mpmath.csc(2)*np.sin(2*x) # Solution de WolframAlpha

      def th(x):
          match len(x)-2:
              case 2:
                  return np.array([ustart, 0.5343, 1.1579, uend])
              case 3:
                  return np.array([ustart, 0.3951, 0.8265, 1.3395, uend])
          return np.zeros_like(x)

      plt.figure(figsize=(12,4))
      H = [1/3, 1/4]
      for i, h in enumerate(H):
          N = int(np.round(1/h -1))
          A = np.zeros([N,N])
          A += np.diagflat(np.ones(N)*(-2+4*h**2))
          A += np.diagflat(np.ones(N-1), 1)
```

```

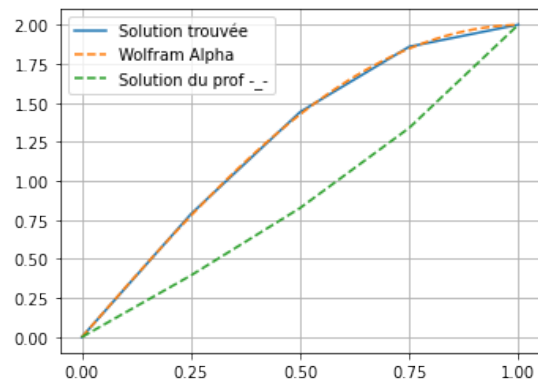
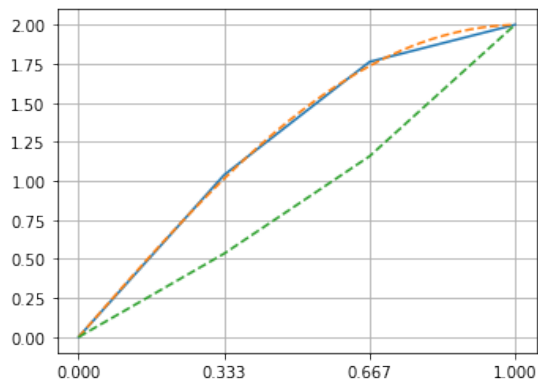
A += np.diagflat(np.ones(N-1), -1)
A = np.matrix(A * 1/h**2)

x = np.linspace(0, 1, N+2)
xos = np.linspace(0, 1, 1000)

u = np.zeros(N+2)
u[-1] = 2
F = f(x[1:-1])
F[0] += -ustart/h**2
F[-1] += -uend/h**2
u[1:-1] = A.I @ F

plt.subplot(101 + i + len(H)*10)
plt.xticks(x)
plt.plot(x, u)
plt.plot(xos, analytic(xos), '--')
plt.plot(x, th(x), '--')
plt.grid()
plt.legend(["Solution trouvée", "Wolfram Alpha", "Solution du prof -_-"])
plt.show()
print("La donnée est fausse, il a utilisé un +u'")

```



La donnée est fausse, il a utilisé un +u''

[]: