Exercice2_SDZ

January 26, 2022

2. Résoudre le problème

$$-u'' = x^3, \qquad u(0) = 0, \qquad u(2) = 0$$

en utilisant la méthode de Ritz vue en classe et en utilisant les fonctions de base

$$N_1(x) = x(2-x) \qquad N_2(x) = x^2(2-x), \qquad N_3(x) = x^2(2-x)^2$$

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

0.1 Solution exacte

$$u(x) = -\frac{x^5}{20} + c_1 x + c_2$$

On utilise les conditions aux bords

$$u(0) = c_2 = 0 \longrightarrow c_2 = 0$$

$$u(2) = -\frac{32}{20} + 2c_1 \longrightarrow c_1 = \frac{32}{40} = \frac{4}{5}$$

$$\boxed{u(x) = -\frac{x^5}{20} + \frac{4}{5}x}$$

0.2 Fonctions de base

$$N_1(x) = x(2-x) = -x^2 + 2x$$

$$N_2(x) = -x^3 + 2x^2$$

$$N_3(x) = x^2(4 - 4x + x^2) = x^4 - 4x^3 + 4x^2$$

On a également

$$f(x) = x^3$$

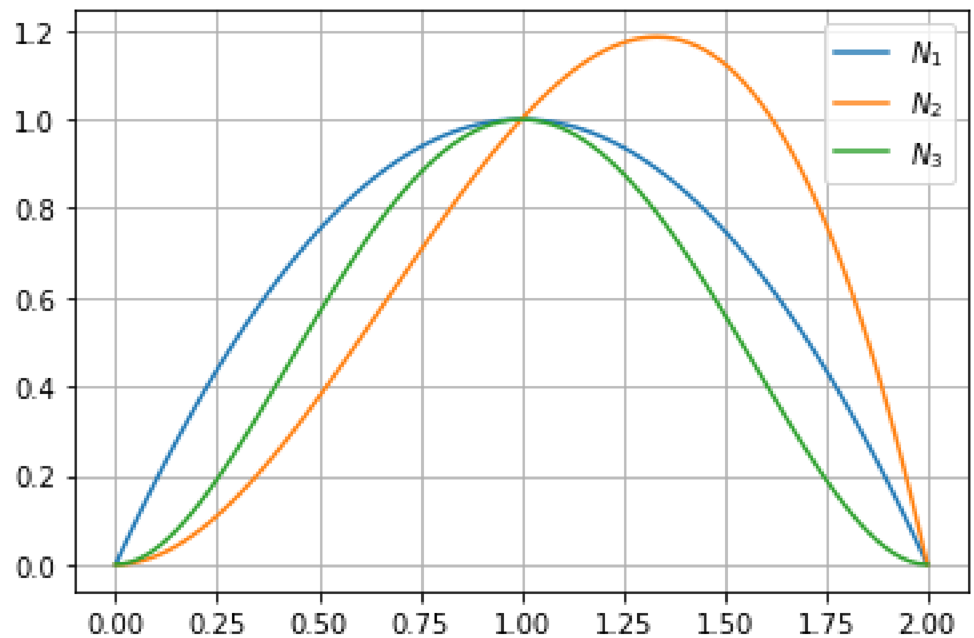
[3]:
```
def u_th(x):
    return -x**5/20 + 4/5*x
```

```
N1 = np.array([0, 0, -1, 2, 0])
N2 = np.array([0, -1, 2, 0, 0])
N3 = np.array([1, -4, 4, 0, 0])
f = np.array([1, 0, 0, 0])
N123 = np.stack([N1, N2, N3])

L = 2

x = np.linspace(0, L, 100)

for i in range(3):
    plt.plot(x, np.polyval(N123[i,:], x), label=f"$N_{i+1}$")
plt.grid()
plt.legend()
plt.show()
```



[16]:
```
A = np.asmatrix(np.zeros([3,3]))
for r in range(3):
    for c in range(3):
        # Intégrale de Nr' x Nc' entre 0 et L
        int = np.polyint(np.polymul(np.polyder(N123[r, :], 1), np.
 ↪polyder(N123[c, :], 1)))
        A[r, c] = np.polyval(int, L) - np.polyval(int, 0)

b = np.asmatrix(np.zeros([3,1]))
```

```
for i in range(3):
    int = np.polyint(np.polymul(f, N123[i,:]))
    b[i,0] = np.polyval(int, L) - np.polyval(int, 0)

print(f"A : \n{A}")

print(f"b : \n{b}")
```

```
A :
[[2.66666667 2.66666667 2.13333333]
 [2.66666667 4.26666667 2.13333333]
 [2.13333333 2.13333333 2.43809524]]
b :
[[2.13333333]
 [3.04761905]
 [1.52380952]]
```

$$Ac = b$$

[12]:

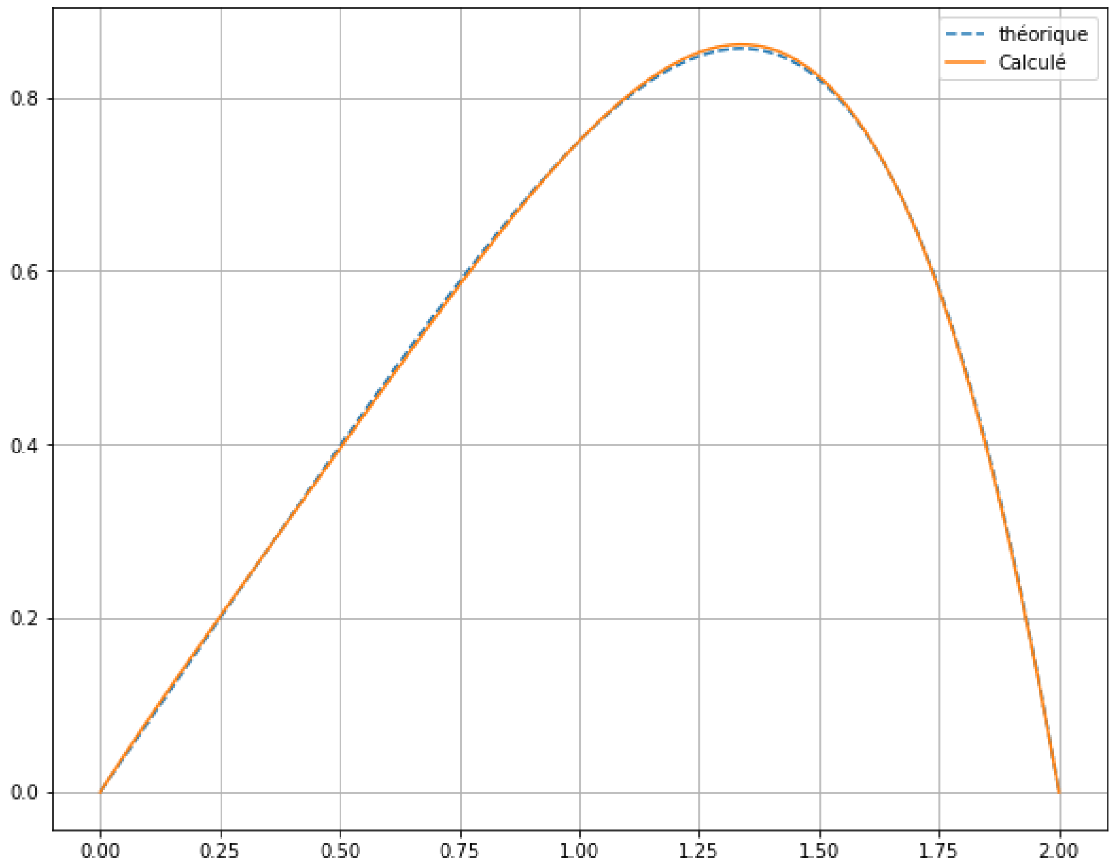
```
c = A.I @ b
print(c)
```

```
[[ 0.42857143]
 [ 0.57142857]
 [-0.25      ]]
```

[13]:

```
def u(x):
    return c[0,0] * np.polyval(N1, x) + c[1,0] * np.polyval(N2, x) + c[2,0] *␣
↪np.polyval(N3, x)

plt.figure(figsize=(10,8))
plt.plot(x, u_th(x), '--', label='théorique')
plt.plot(x, u(x), label='Calculé')
plt.grid()
plt.legend()
plt.show()
```



$$a_{11} = \int_0^L N_1'(x) N_1'(x) dx$$