Exercice_demo_SDZ

January 26, 2022

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

[54]:
```
def u0(x):
    return -1/12*x**4 + 1/6*x + 1
a = 0
b = 1
Ni = [np.array([1]), np.array([1, 0]), np.array([1, 0, 0]), np.array([1, 0, 0,␣
↪0]), np.array([1, 0, 0, 0, 0])]
```

[56]:
```
i = np.arange(4) + 1
N = 4
R = np.repeat(i, len(i), axis=0).reshape(-1,len(i))
C = np.repeat(i, len(i), axis=0).reshape(-1,len(i), order='F')

A = R * C / (R + C - 1)

v1 = 1
f = 1/(3+i)  - 1/6 * v1 # intégrale de x^2 * xi dx entre 0 et L

ui = np.concatenate([np.array([1]), np.linalg.inv(A) @ f])


def u(x):
    Nis = np.array([np.polyval(N, x) * k for N,k in zip(Ni, ui)])
    out = np.sum(Nis, axis=0)
    return out

plt.figure(figsize=(10,10))

for n in [10, 20]:
    x = np.linspace(a, b, n)
    plt.plot(x, u(x), 'x')

xup = np.linspace(a, b, 100)

plt.plot(xup, u0(xup), "--")
```

```
plt.plot(0, 1, "r*")
neumann_length = 0.05
plt.plot([xup[-1]-neumann_length,xup[-1]+neumann_length], [u0(xup[-1]) + 1/
 ↪6*neumann_length, u0(xup[-1]) - 1/6 * neumann_length], "r-")
plt.grid()
plt.show()
```