Homework — Numerical Linear Algebra

Dimitri Tabatadze · 13:59, December 8, 2023

1.

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \ P = I$$

The richardson method will go like this:

$$\begin{aligned} I\frac{x^{(k+1)} - x^{(k)}}{\tau} + Ax^{(k)} &= b \\ x^{(k+1)} - x^{(k)} &= \tau I^{-1} b - \tau I^{-1} A x^{(k)} \\ x^{(k+1)} &= (I - \tau I^{-1} A) x^{(k)} + \tau b \\ &\Downarrow \\ \tau_{\text{opt}} &= \frac{2}{\lambda_{\max}(I^{-1}A) + \lambda_{\min}(I^{-1}A)} \\ &= \frac{2}{\lambda_{\max}(A) + \lambda_{\min}(A)} \\ &= \frac{2}{1+3} \\ &= \frac{1}{2} \end{aligned}$$

2. I used the code from the task 3 to calculate the solution:

$$x \approx \begin{pmatrix} 0.995 \\ 0.957 \\ 0.791 \end{pmatrix}$$

3. I think that this code is very much self-descriptive.

```python
import numpy as np


def sor_step(a, x, b, omega):
    for i in range(len(x)):
        x[i] = omega / a[i][i] * (b[i] - sum(a[i][j] * x[j] for j in range(len(x)) if j != i)) \
            + (1 - omega) * x[i]

    return x


n = int(input("n             \t> "))
a = [[float(input(f"a[{i}][{j}] \t> ")) for j in range(n)] for i in range(n)]
b = [float(input(f"b[{i}]       \t> ")) for i in range(n)]
x = [float(input(f"x0[{i}]      \t> ")) for i in range(n)]
omega = float(input("omega         \t> "))
tol = float(input("TOL           \t> "))
max_iters = float(input("max_iters \t> "))

iters = 0

while iters < max_iters:
```

```
    px = [x_i for x_i in x]
    x = sor_step(a, x, b, omega)
    e = max([abs(x[i] - px[i]) for i in range(len(x))]) / max(map(abs,px))
    if e < tol:
        print(x)
        break
    iters += 1
else:
    print("iteration limit exceeded")
```

4. I picked $x^{(0)}$ to be a vector with all ones. Using the code I wrote for the task 3, this is the solution I get:

$$x \approx \begin{pmatrix} 1.53872\dots \\ 0.73141\dots \\ 0.10797\dots \\ 0.17328\dots \\ 0.04055\dots \\ 0.08524\dots \\ 0.16644\dots \\ 0.12197\dots \\ 0.10125\dots \\ 0.09045\dots \\ 0.07202\dots \\ 0.07026\dots \\ 0.06875\dots \\ 0.06324\dots \\ 0.05971\dots \\ 0.05570\dots \\ 0.05187\dots \\ 0.04924\dots \\ 0.04677\dots \\ 0.04448\dots \\ 0.04246\dots \\ 0.04053\dots \\ 0.03876\dots \\ 0.03717\dots \\ 0.03570\dots \\ 0.03434\dots \\ 0.03309\dots \\ 0.03191\dots \\ 0.03082\dots \\ 0.02980\dots \\ 0.02885\dots \\ 0.02795\dots \\ 0.02711\dots \\ 0.02632\dots \\ 0.02557\dots \\ 0.02486\dots \\ 0.02419\dots \\ 0.02356\dots \\ 0.02296\dots \\ 0.02239\dots \\ 0.02184\dots \\ 0.02133\dots \\ 0.02083\dots \\ 0.02036\dots \\ 0.01991\dots \\ 0.01948\dots \\ 0.01906\dots \\ 0.01867\dots \\ 0.01829\dots \\ 0.01792\dots \\ 0.01757\dots \\ 0.01723\dots \\ 0.01691\dots \\ 0.01660\dots \\ 0.01630\dots \\ 0.01601\dots \\ 0.01573\dots \\ 0.01546\dots \\ 0.01519\dots \\ 0.01494\dots \\ 0.01470\dots \\ 0.01446\dots \\ 0.01423\dots \\ 0.01401\dots \\ 0.01380\dots \\ 0.01359\dots \\ 0.01338\dots \\ 0.01318\dots \\ 0.01297\dots \\ 0.01278\dots \\ 0.01270\dots \\ 0.01252\dots \\ 0.01237\dots \\ 0.01220\dots \\ 0.01129\dots \\ 0.01114\dots \\ 0.01217\dots \\ 0.01201\dots \\ 0.01542\dots \\ 0.01523\dots \end{pmatrix}$$