Exercise sheet 5

Numerical Analysis 2022

1 PS.5 (Banach's Fixed Point Theorem)

Given $\Phi : [0,1] \to \mathbb{R}$ with $\Phi(x) = \frac{1}{3}x^2 + \frac{1}{2}$, consider the iterative scheme

$$x_{k+1} := \Phi(x_k).$$

Prove that the sequence $(x_k)_{k\in\mathbb{N}}$ linearly converges for all $x_0 \in [0,1]$.

2 PS.5 (Banach's Fixed Point Theorem)

Consider the informal expression

$$\sqrt{2+\sqrt{2+\sqrt{2+\sqrt{\dots}}}}$$

a) Rewrite the above expression as an iterative scheme $x_{k+1} = ...$

b) What is the value of the above expression and what are the convergence properties of the iterative scheme?

c) Illustrate the convergence order numerically in julia by plotting $|x_k - x|$ in a logarithmic scale for $k = 1, \dots, 20$.

3 PS.5 (diagonally dominant)

Consider the matrices

$$A_1 = \begin{pmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 1 & 2 \end{pmatrix}, \qquad A_2 = \begin{pmatrix} 2 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \qquad A_3 = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}.$$

a) Which of these 3 matrices is strictly diagonally dominant?

b) Investigate the convergence of Jacobi's method with respect to A_1, A_2, and A_3 and the right-hand-side $b = (0,0,0)^\top$ and the initial vector $x^{(0)} = (1,1,1)^\top$ by explicitly computing and simplifying the recursion.

4 PS.5 (iterative schemes)

Jacobi:

$$x_i^{(k+1)} = \frac{1}{a_{i,i}} \left(b_i - \sum_{j \neq i} a_{i,j} x_j^{(k)} \right), \quad i = 1, \ldots, n, \tag{1}$$

$$= \frac{1}{a_{i,i}} \left(b_i - \sum_{j=1}^{n} a_{i,j} x_j^{(k)} \right) + x_i^{(k)} \tag{2}$$

Gauss-Seidel:

$$x_i^{(k+1)} = \frac{1}{a_{i,i}} \left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{i,j} x_j^{(k)} \right), \quad i = 1, \ldots, n, \tag{3}$$

a) Implement both schemes in julia as myJacobi2 and myGaussSeidel2 in-situ, i.e., without keeping all the iterates.

b) Observe the rule of thumb: Gauss-Seidel requires fewer iterations than Jacobi's method.

c) Observe that the simple parallel version of Jacobi's method is indeed faster. (Before starting julia (or jupyter) type in the terminal:

"export JULIA_NUM_THREADS=8"

(or whatever number of cores your processor provides, 2 or 4,...))