Department for Mathematics and Computer Science
Freie Universität Berlin
Prof. Dr. V. John, john@wias-berlin.de
André-Alexander Zepernick, a.zepernick@fu-berlin.de

Berlin, 13.01.2025

Numerical Mathematics II

Exercise Problems 11 (Final Exercise Sheet)

Attention: The approach for getting a solution has to be clearly presented. All statements have to be proved, auxiliary calculations have to be written down. Statements given in the lectures can be used without proof.

1. *BDF2 with variable step size.* Derive a method by using the following approach. Consider three subsequent nodes x_{k-1}, x_k, x_{k+1} with the mesh widths $h_{k+1} = x_{k+1} - x_k$ and $h_k = x_k - x_{k-1}$. Denote the numerical approximations of the solutions by y_{k-1}, y_k, and y_{k+1}. Now, take the interpolation polynomial $p(x)$ through these points and require that $p'(x_{k+1}) = f(x_{k+1}, y_{k+1})$. This gives a method that is called BDF2. Give this method by expressing as much terms as possible with $\sigma_{k+1} = h_{k+1}/h_k$ and order the terms with respect to y_{k-1}, y_k, and y_{k+1}. **3 points**

2. *Properties of M-matrices.* Prove the following statements.

   (a) Let $A \in \mathbb{R}^{n\times n}$ be an M-matrix. Each matrix that is obtained by multiplying arbitrary rows or columns of A with positive numbers is an M-matrix. **3 points**

   (b) The sum of two M-matrices is not necessarily an M-matrix.
   Hint: Find a counter example with matrices from $\mathbb{R}^{2\times 2}$, keeping in mind that M-matrices are in some sense diagonally dominant matrices. **3 points**

3. *Preconditioned conjugate gradient (PCG) method.* Continue Problem 3 from Exercise sheet 09. Consider meshes with

$$h \in \{1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1024\}.$$

   Implement the preconditioned conjugate gradient method (Algorithm 8.8) for solving these equations. Use as preconditioner

   (a) $M = I$, i.e., no preconditioning, same as Problem 3 from Exercise sheet 09,

   (b) $M = \text{diag}(A)$, Jacobi preconditioner, see Remark 8.2,

   (c) $M = SSOR(A)$, see Remark 8.2,

   (d) $M = L^T L$, where L is the incomplete Cholesky factorization of A with zero fill-in. The incomplete Cholesky decomposition is a variant of ILU for symmetric matrices for which $L = U^T$. If you use MATLAB, then use the command `ichol`, otherwise try to figure out whether a routine for the incomplete Cholesky factorization is available for the programming language that you use.

Give the number of iterations for solving the system. What can be observed? Find an explanation for the behavior of the Jacobi preconditioner.
Hint. Since only the number of iterations is of interest, the solution of the linear systems with the preconditioner can be implemented with the backslash command. Also the SSOR preconditioner can be implemented in the form given in Remark 8.2. **5 points**

The exercise problems should be solved in groups of four students. The solutions have to be submitted until **Monday, Jan. 20, 2025, 10:00 a.m.** via the whiteboard.