Technische Universität München
Lehrstuhl für Informatik V
Univ.-Prof. Dr. M. Bader
Carsten Uphoff

SS 2017
Sheet 1
02.05.2017

Scientific Computing II

Iterative Solvers

Exercise 1: Repetition "Finite Differences"

Consider the one-dimensional Poisson equation with homogeneous Dirichlet conditions

$$\begin{aligned} -\frac{d^2u}{dx^2} &= f(x), \quad x \in (0,1), \\ u(0) = u(1) &= 0. \end{aligned} \tag{1}$$

(a) Discretise the Poisson equation by finite differences using an equidistant mesh size $h = 1/N$ and $N+1$ grid points.

(b) Write the finite difference approximation from (a) in matrix-vector form $Au = b$. Therefore, define the entries of the matrix $A \in \mathbb{R}^{N+1\times N+1}$.

(c) Write the finite difference approximation as $Au = b$, where $A \in \mathbb{R}^{N-1\times N-1}$ and $b \in \mathbb{R}^{N-1}$, by substituting the values for $u(0)$ and $u(1)$.

Row-Wise Derivation of Smoothers

Besides the matrix-based derivation (see lecture slides), most smoother methods can also be easily derived row-wise. Each row of the linear system reads:

$$\sum_j A_{ij}u_j = b_i, \;\; i = 1, ..., N \tag{2}$$

The i-row can be separated into $A_{ii}u_i + \sum_{i\neq j} A_{ij}u_j = b_i$. Rearranging yields

$$u_i = \frac{1}{A_{ii}}\left(b_i - \sum_{j\neq i} A_{ij}u_j\right). \tag{3}$$

- *Jacobi method*
  For the right hand side of Eq. (3) we use the iterative solution at iteration step (n) and obtain the new iterative solution $(n+1)$:

$$u_i^{(n+1)} = \frac{1}{A_{ii}}\left(b_i - \sum_{j\neq i} A_{ij}u_j^{(n)}\right).$$

- *Weighted Jacobi method*
  We introduce a weighting factor ω and split the left hand side $u_i = \frac{1}{\omega} u_i + \left(1 - \frac{1}{\omega}\right) u_i$. We can now evaluate parts of u_i at (n) or $(n+1)$. We obtain:

$$\begin{aligned} \tfrac{1}{\omega} u_i^{(n+1)} + \left(1 - \tfrac{1}{\omega}\right) u_i^{(n)} &= \frac{1}{A_{ii}} \left(b_i - \sum_{j \neq i} A_{ij} u_j^{(n)} \right) \\ \Leftrightarrow \qquad u_i^{(n+1)} &= \frac{\omega}{A_{ii}} \left(b_i - \sum_{j \neq i} A_{ij} u_j^{(n)} \right) + (1 - \omega)\, u_i^{(n)}. \end{aligned}$$

  The right hand side of the last equation corresponds to a weighted average of the last solution $u_i^{(n)}$ and the solution predicted by the (non-weighted) Jacobi method.

- *Gauss-Seidel method*
  We solve the right hand side of Eq. (3) with both new and old values $u_j^{(n)}$, $u_j^{(n+1)}$. By this method, we can only use one array to store the solution u since we can immediately write the entries at $(n+1)$ into the original positions of the solutions $u_j^{(n)}$. The method reads:

$$u_i^{(n+1)} = \frac{1}{A_{ii}} \left(b_i - \sum_{j<i} A_{ij} u_j^{(n+1)} - \sum_{j>i} A_{ij} u_j^{(n)} \right).$$

- *Successive-Over-Relaxation method*
  Similar to weighted Jacobi, we split the left hand side $u_i = \frac{1}{\omega} u_i + \left(1 - \frac{1}{\omega}\right) u_i$ and evaluate parts of u_i at (n) or $(n+1)$, but use both old and new values on the right hand side.

$$\begin{aligned} \tfrac{1}{\omega} u_i^{(n+1)} + \left(1 - \tfrac{1}{\omega}\right) u_i^{(n)} &= \frac{1}{A_{ii}} \left(b_i - \sum_{j<i} A_{ij} u_j^{(n+1)} - \sum_{j>i} A_{ij} u_j^{(n)} \right) \\ \Leftrightarrow \qquad u_i^{(n+1)} &= \frac{\omega}{A_{ii}} \left(b_i - \sum_{j<i} A_{ij} u_j^{(n+1)} - \sum_{j>i} A_{ij} u_j^{(n)} \right) + (1 - \omega)\, u_i^{(n)}. \end{aligned}$$

  So the right hand side is an average of the last solution $u_i^{(n)}$ and the solution predicted by the Gauß-Seidel method.

Exercise 2: Eigenvalues and eigenvectors

Show that the discretised sine, i.e. $u_i = sin(k\pi ih)$, is an eigenvector with eigenvalue $\lambda = (4/h^2)\sin^2(k\pi h/2)$ of the finite difference matrix A in Exercise 1(c). You may use the following trigonometric identities:

$$\sin(a+b) + \sin(a-b) = 2\sin(a)\cos(b) \tag{4}$$

$$\cos(2x) = 1 - 2\sin^2(x) \tag{5}$$

Exercise 3: Fourier Analysis for Jacobi Methods

In this exercise we are interested in the smoothing properties of the weighted Jacobi method applied on the Problem of Exercise 1 with zero right-hand side ($f(x) = 0$).

(a) Formulate the weighted Jacobi method and write the iteration scheme in the form

$$u_i^{(n+1)} = \sum_j M_{ij} u_i^{(n)}.$$

(b) Determine the eigenvalues and eigenvectors of M.

Hints: From the lecture you know that $M := I - \omega \operatorname{diag}(A)^{-1} A$ for weighted Jacobi. Combine this fact with the results from Exercise 2 in order to compute the eigendecomposition easily.

(c) How can the eigendecomposition be used to calculate the reduction of error in each smoothing iteration?

(d) For a multigrid algorithm we are interested in removing the "high frequencies", say $N/2 \leq k \leq N$. Show that $\omega = 2/3$ is the best choice in the sense that it solves

$$\omega = \min_{\omega'} \max_{N/2 \leq k \leq N} |\lambda_k(\omega')|,$$

where λ_k is the k-th eigenvalue. What is the reasoning of this criterion?