

Parallel Numerics

Exercise 9: Gradient methods, Preconditioning, and Eigenvalues

1) Gradient Methods

Consider the linear system $Ax = b$ where

$$A = \begin{pmatrix} 11 & -9 \\ -9 & 11 \end{pmatrix}, b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \text{and} \quad x = \begin{pmatrix} \frac{11}{40} \\ \frac{9}{40} \end{pmatrix}. \quad (1)$$

- i) Is the matrix A symmetric positive definite (SPD)?
- ii) Apply the first two iterations of the gradient method (steepest descent). Use the initial vector $x^{(0)} = (0, 0)^T$.
- iii) Show that the residuals $r^{(k)}$ and $r^{(k-2)}$, $k \geq 2$, are parallel in \mathbb{R}^2 for the gradient method.
- iv) Solve (1) with the CG method for the initial solution $x^{(0)} = (0, 0)^T$. Compare your results to part ii). (see algorithm snippet below for the CG algorithm).
- v) Consider the Conjugate Gradient method that computes the solution x iteratively as a series $\{x^{(k)}\}$:

$$\begin{aligned} p^{(0)} &= r^{(0)} = b - Ax^{(0)} \\ \alpha^{(k)} &= -\frac{\langle r^{(k)}, r^{(k)} \rangle}{\langle p^{(k)}, Ap^{(k)} \rangle} \\ x^{(k+1)} &= x^{(k)} - \alpha^{(k)} p^{(k)} \\ r^{(k+1)} &= r^{(k)} + \alpha^{(k)} Ap^{(k)} \\ \mathbf{if} \quad \|r^{(k+1)}\|_2^2 &\leq \epsilon \quad \mathbf{then break} \\ \beta^{(k)} &= \frac{\langle r^{(k+1)}, r^{(k+1)} \rangle}{\langle r^{(k)}, r^{(k)} \rangle} \\ p^{(k+1)} &= r^{(k+1)} + \beta^{(k)} p^{(k)} \end{aligned}$$

Implement this algorithm first for a sequential environment and afterwards for a parallel environment. To keep the parallelisation as simple as possible, use functions to compute the inner products, the matrix-vector multiplications and the addition of a vector, multiplied with a scalar, to another vector. Parallelise this algorithm by methods previously acquired in this course.

2) (Block) Sparse Approximate Inverses

- i) What is preconditioning and why is it necessary/useful?
- ii) What are the main advantages of the SPAI (Sparse Approximate Inverse) algorithm?
- iii) How is it possible to parallelize the computation of a static SPAI algorithm performing no pattern updates? What about the load balancing? (Give only thoughts, no pseudocode)
- iv) Derive a block version of the static SPAI algorithm without pattern updates.
- v) Derive a block version of SPAI using pattern updates.

3) Sparse matrices and graphs

Let us consider the following cut-out of the munich subway network:



- i) Interpret the network as a graph G and give the corresponding adjacency matrix A_G as well as the incidence matrix I_G .
- ii) Give the transitive closure of G .
- iii) We want to quantify the importance of the different subway stations. Therefore, we proceed similarly to google when computing "pageranks": A station is considered "important" if other "important" stations are connected to it. The importance r_i of a station p_i distributes simultaneously to all outgoing connections and can be computed via

$$r_i = \sum_{\substack{j \\ p_j \rightarrow p_i}} \frac{r_j}{d_j}, \quad (2)$$

where d_j is the outgoing degree of a node p_j .

- a) Formulate (2) in matrix vector notation.
- b) Which property has the occurring matrix?
- c) Compute the "importance" of the various subway stations. Which problem occurs during the computation and how can you solve it?