Department of Applied Mathematics
Faculty of Electrical Engineering, Mathematics and Computer Sciences
Dr. K. Smetana, Dr. M. Walter

June 14, 2019

Exam for the Course

Scientific Computing (code: 191551200)

2018/2019

The total number of points is 54 points. In order to get the grade 5.5 you will need at most 27 points. No notes or calculators are allowed. **Explain each step in your solution.** This exam has 9 tasks.

Task 1 (LU decomposition) (4 + 2 points)

Consider the linear system of equations $Ax = b$ with

$$A = \begin{pmatrix} 1 & 0 & 2 \\ 4 & 4 & 5 \\ 3 & 4 & 2 \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}. \tag{1}$$

a) Compute the LU decomposition of A defined in (1) without pivoting, where L should be a unit lower triangular matrix.

b) Use the just determined LU decomposition of the matrix A defined in (1) to solve the linear system of equations $Ax = b$.

Task 2 (SVD) (8 points)

Let $A = U\Sigma V^T$ be the SVD of $A \in \mathbb{R}^{m\times n}$, $m \geq n$, with orthogonal matrices $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$, and $r \leq n$ be the number of positive singular values of A. Let furthermore null(A) be the null space (or kernel) of A.

a) Prove: $r = \text{rank}(A)$

b) Prove: $\text{range}(A) = \text{span}\{u_1, \ldots, u_r\}$, where $u_1, \ldots, u_r$ are the first r columns of U.

c) Prove: $\text{null}(A) = \text{span}\{v_{r+1}, \ldots, v_n\}$, where $v_{r+1}, \ldots, v_n$ are the $r+1$-th – n-th columns of V.

d) Consider the circle $S = \{x \in \mathbb{R}^2 \mid \|x\| = 1\}$ and the set $\{Ax, x \in S\}$. Name the vectors in the right part of the picture handed out and give a short explanation.

Task 3 (Comparison of iterative solvers for linear systems of equations) (10 points)

We are given the task to solve (approximately) a linear system of equations $Ax = b$, where the matrix $A \in \mathbb{R}^{n\times n}$ is large, sparse, symmetric, and positive definite. Which algorithms that we discussed during the course can be recommended to compute such an approximation and why? Comment also briefly why other methods we discussed are not suited for the current task. Compare the (two) algorithms that are suited for the task in terms of ...

- their key ideas (precise formulas are not needed; state also to which class of iterative solvers the respective method belongs to)
- their convergence behavior/rate
- their computational costs.

Based on your comparison decide which method you would recommend to use.

Task 4 (Existence of weak solutions) (2 points)
Explain why it is important to first analyze the existence and uniqueness of the weak solution before dealing with its approximation.

Task 5 (Stability of Finite Element approximation) (7 points)
Let $I = (a,b) \subset \mathbb{R}$, $f \in L^2(I)$ and define the vector space $X := \{v \in H^1(I) : v(a) = 0\}$ equipped with the norm $\|v\|_X := \left(\int_I (v'(x))^2\,dx\right)^{1/2}$. Moreover, let $y \in X$ be the weak solution of the problem, satisfying

$$\int_I y'(x)v'(x)\,dx + \int_I y(x)v(x)\,dx = \int_I f(x)v(x)\,dx \quad \forall v \in X. \tag{2}$$

Additionally, let $I_h := \{x_0, \ldots, x_N\} \subset I$ be a mesh with $x_0 = a,\ x_N = b,\ x_{j+1} = x_j + h_j,\ h_j > 0$ and define $I_j := (x_{j-1}, x_j),\ j = 1, \ldots, N$ and $h := \max\limits_{j=0,\ldots,N-1} h_j$. Finally, let $X_h \subset X$ be the linear Finite Element space

$$X_h := \{\phi_h \in C^0([a,b]) \mid \phi_h(a) = 0, \phi_h|_{I_j} \in \mathbb{P}_1\ \forall j = 1, \ldots, N\}$$

and $y_h \in X_h$ the Finite Element approximation of the weak solution $y \in X$ of (2). Show that the algorithm "Compute the FE approximation $y_h \in X_h$ for given data $f \in L^2(I)$" is stable, i.e. show that small errors in the data in the L^2-norm lead to small errors in FE approximation in the X-norm. Assume that linear systems of equations are solved exactly.

Task 6 (Power method) (7 points)
Consider an Hermitian matrix $A \in \mathbb{C}^{n \times n}$ whose eigenvalues satisfy

$$|\lambda_1| = |\lambda_2| > |\lambda_3| \geq |\lambda_4| \geq \ldots \geq |\lambda_n| \quad \text{and} \quad \lambda_1 > 0 \text{ and } \lambda_2 = \lambda_1 \text{ or } \lambda_2 = -\lambda_1.$$

Can the power method still be used to approximate λ_1? (Argumentation is sufficient, no proof necessary). If yes, what can be said about the eigenvector the method is converging to? Hint: Rewrite the term $A^k z^{(0)}$, where $z^{(0)}$ denotes the starting vector of the power method exploiting that A is Hermitian.

Task 7 (PageRank) (4 points)
Reminder definition of PageRank algorithm: Denote with $w_1, \ldots, w_n$ all webpages in the considered network. Collecting the PageRanks of those webpages in a vector $\mathbf{R}$ we may formulate the following system of equations for their PageRank:

$$\mathbf{R} = d(P + A)\mathbf{R} + o\frac{1-d}{n}, \qquad \text{where } o = (1, \ldots, 1)^T, \tag{3}$$

$$P_{ij} = \begin{cases} 1/(C(w_j)) & \text{if } w_j \text{ points to } w_i, \\ 0 & \text{else} \end{cases} \qquad \text{and} \quad A_{:,j} = \begin{cases} 1/n & \text{if the } j-\text{th column of P is 0}, \\ 0 & \text{else} \end{cases}. \tag{4}$$

Here, $A_{:,j}$ denotes the j-th column of A, $C(w_j)$ denotes the number of links going out of page w_j and $0 < d < 1$ denotes some constant damping factor. If we scale $\mathbf{R}$ such that $o^T\mathbf{R} = 1$, (3) is equivalent to the problem: $\mathbf{R} = (d(P + A) + oo^T\frac{1-d}{n})\mathbf{R}$. The Simplified PageRank algorithm just considers the following system of equations:

$$\mathbf{R} = dP\mathbf{R}. \tag{5}$$

Actual task: Explain which problems occur with the Simplified PageRank algorithm if the network contains dangling nodes. Moreover, elaborate on how these problems are solved with the PageRank algorithm (3). Illustrate your discussion with a small graph and the corresponding P matrix.

Algorithm 0.1: A prototype algorithm for element-wise sampling

Input : $m \times n$ matrix A; integer $s > 0$ denoting the number of elements to be sampled; probability distribution p_{ij} $i = 1, \dots, m, j = 1, \dots, n$ with $\sum_{i,j=1} p_{ij} = 1$.

Output: matrix $\widetilde{A} \in \mathbb{R}^{m \times n}$

$\widetilde{A} \leftarrow$ all-zero sparse $m \times n$ matrix

for $t = 1, \dots, s$ **do**

  Set $\widetilde{A}_{i(t),j(t)} :=?$

Task 8 (Element-wise sampling) (2 + 1 + 1 points)

We want to use Algorithm 0.1 to construct a random sketch of a matrix $A \in \mathbb{R}^{m \times n}$ via randomized sampling.

a) How do we have to choose $\widetilde{A}_{i(t),j(t)}$ such that element-wise the expectation of $\widetilde{A}$ is equal to the original matrix A? Proof is required.

b) Explain why performing uniform sampling, i.e. setting $p_{ij} = 1/(mn)$ for all i, j is in general not a good idea.

c) Explain why element-wise sampling does in general require a large number of samples s to obtain satisfactory results.

Task 9 (Strength of two models for the Steiner tree problem) (6 points)

Reminder: Definition and notation for Steiner trees and Steiner arborescences:

Let $G = (V, E)$ be an undirected graph, let $c \in \mathbb{R}^E_+$ be a nonnegative objective vector and let $T \subseteq V$ be a set of terminal nodes. A Steiner tree is a subset $F \subseteq E$ of edges such that F connects all terminals, i.e., for each pair $s, t \in T$ of terminals there is an s-t-path that only uses edges of F. The Steiner tree problem is to find a Steiner tree with mininum costs $c(F) := \sum_{e \in F} c_e$. Note that a Steiner tree may in fact contain cycles, but due to nonnegativity of c there always exists an optimal solution that is a tree.

Similar to the TSP we consider the bidirected version of G, denoted by $D = (V, A)$, i.e.,

$$A := \{(u, v), (v, u) : \{u, v\} \in E\}$$

contains the two arcs corresponding to each edge of G. Additionally, we fix a node $r \in T$. A Steiner arborescence is a subset $B \subseteq A$ of arcs such that for every other terminal $k \in T \setminus \{r\}$ there is a k-r-path only using arcs of B.

Remark: One can think of Steiner arborescences as Steiner trees that are "directed towards r".

Actual task:

Show that the two following integer programming models for the Steiner tree problem are equivalent with respect to the x-variables.

The first model has only edge variables $x \in \{0, 1\}^E$ with $x_e = 1 \iff e \in F$. It reads

$$\begin{aligned} \min \quad & c^\intercal x \\ \text{s.t.} \quad & x(\delta(S)) \geq 1 && \forall S \subseteq V : S \cap T \neq \emptyset,\ r \notin S && (6a) \\ & x_e \in \{0, 1\} && \forall e \in E. && (6b) \end{aligned}$$

The second model uses, in addition to edge variables, flow variables $f^{(k)} \in \mathbb{R}^A$ for all $k \in T \setminus \{r\}$ ensuring existence of a k-r-flow of value 1. It reads

$$\begin{aligned}
\min\ & c^\intercal x \\
\text{s.t.}\quad & 0 \leq f^{(k)}_{(u,v)} \leq x_{\{u,v\}} && \forall \{u,v\} \in E,\ \forall k \in T \setminus \{r\} && \text{(7a)} \\
& f^{(k)}(\delta^{\text{out}}(v)) - f^{(k)}(\delta^{\text{in}}(v)) = \begin{cases} +1 & \text{if } v = k \\ -1 & \text{if } v = r \\ 0 & \text{otherwise} \end{cases} && \forall v \in V\ \forall k \in T \setminus \{r\} && \text{(7b)} \\
& x_e \in \{0,1\} && \forall e \in E && \text{(7c)}
\end{aligned}$$

overview points

1	2	3	4	5	6	7	8	9
6	8	10	2	7	7	4	4	6

Table 1: Total: 54 points