Numerical Linear Algebra - Sheet 7

to be handed in until December 6, 2023, 11am.

Problem 1. Let $\mathbf{A}$ be a real symmetric matrix with eigenvalues λ_j, $1 \leq j \leq n$. Show that for any polynomial $p(t) = \sum_{j=0}^{k} a_j t^j$, $a_0, \dots, a_k \in \mathbb{R}$, the matrix

$$p(\mathbf{A}) = \sum_{j=0}^{k} a_j \mathbf{A}^j$$

is also symmetric. Moreover, the eigenvectors of $\mathbf{A}$ and $p(\mathbf{A})$ are identical, while the eigenvalues of $p(\mathbf{A})$ are equal to $p(\lambda_j)$ for $1 \leq j \leq n$.

Problem 2 (Eigenvalues and eigenvectors of a tensor product operator)**.** For vectors $\mathbf{u} \in \mathbb{R}^n$ and $\mathbf{v} \in \mathbb{R}^m$ we define the tensor product as a (block) vector

$$\mathbf{u} \otimes \mathbf{v} = \begin{pmatrix} u_1 v & u_2 v & \cdots & u_n v \end{pmatrix}^T \in \mathbb{R}^{nm}.$$

Let $\mathbf{A}$ and $\mathbf{B}$ be matrices in $\mathbb{R}^{n\times n}$ and $\mathbb{R}^{m\times m}$ respectively. We define their tensor product $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{nm\times nm}$ as

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{n1}\mathbf{B} & \cdots & a_{nn}\mathbf{B} \end{pmatrix}.$$

(a) Show that $(\mathbf{A} \otimes \mathbf{B})(\mathbf{u} \otimes \mathbf{v}) = (\mathbf{A}\mathbf{u}) \otimes (\mathbf{B}\mathbf{v})$.

(b) Prove that the eigenvectors of the tensor product are the pairwise tensor products of the eigenvectos of the individual matrices. What are the corresponding eigenvalues?

Problem 3. Let $\mathbf{L}_d$ be the discretization of the d-dimensional Laplace operator on the unit square by the five-point stencel with a uniform Dirichlet boundary condition to the mesh size $h = \frac{1}{n+1}$. In 1D it is given in terms of the matrix $\mathbf{T}_2$ defined in Problem 4 of Sheet 6 by

$$\mathbf{L}_1 = \frac{1}{h^2}\mathbf{T}_2 \in \mathbb{R}^{n\times n}.$$

In 2D it is given by

$$\mathbf{L}_2 = \frac{1}{h^2}\begin{pmatrix} \mathbf{D} & -\mathbb{I} & & & \\ -\mathbb{I} & \mathbf{D} & -\mathbb{I} & & \\ & \ddots & \ddots & \ddots & \\ & & -\mathbb{I} & \mathbf{D} & -\mathbb{I} \\ & & & -\mathbb{I} & \mathbf{D} \end{pmatrix} \in \mathbb{R}^{n^2\times n^2},$$

where $\mathbb{I} \in \mathbb{R}^{n\times n}$ and $\mathbf{D} \in \mathbb{R}^{n\times n}$ are defined as

$$\mathbb{I} = \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{D} = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix}.$$

(a) Use the results of Problem 2 to show that $\mathbf{L}_2$ could be expressed as

$$\mathbf{L}_2 = (\mathbf{L}_1 \otimes \mathbb{I}) + (\mathbb{I} \otimes \mathbf{L}_1).$$

(b) List the four smallest eigenvalues of $\mathbf{L}_2$. What is their multiplicity?

Problem 4 (Programming)**.** Write a program that solves the 2D Laplace problem

$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

with the Gauss-Seidel iteration, where the system matrix $\mathbf{A} = \mathbf{L}_2 \in \mathbb{R}^{n^2 \times n^2}$ is defined in Problem 3, and the right hand side is given by the constant vector $\mathbf{b} = (1, ..., 1)^T$, by observing the following steps. Try to avoid explicitly forming the system matrix.

(a) Implement a function vmult that performs the matrix-vector product $\mathbf{A} \cdot \mathbf{v}$ of $\mathbf{A}$ with a given vector $\mathbf{v}$, and returns a vector $\mathbf{w}$.

(b) Write a function gauss_seidel_step that implements a single step of the Gauss-Seidel iteration. As in (a) this function should take a vector $\mathbf{v}$ and return the resulting vector $\mathbf{w}$.

(c) Use the null-vector $\mathbf{x}^{(0)} = (0, ..., 0)^T$ as initial guess and test your program with $n = 20$ and $n = 100$. Let $\mathbf{b} = (1, ..., 1)^T$. Observe the evolution of the residual in the 2-norm

$$r^{(k)} = \|\mathbf{A} \cdot \mathbf{x}^{(k)} - \mathbf{b}\|_2$$

for 50 steps of the Gauss-Seidel iteration. Plot the obtained result vector after 1, 5, 15, 30, and 50 iterations.

Note: Be careful not to crash your computer when starting your program with $n = 100$ or more, if you explicitly build the system matrix, because of the possibly high memory usage.