Questions to prepare for the written exam, DN2230 (Complete list)

1. Given any matrix $A \in \mathbb{R}^{3\times 3}$, is it possible to find all its eigenvalues exactly (e.g. as $\lambda_1 = \sqrt{3+\sqrt{17}}$)? Given any matrix $B \in \mathbb{R}^{5\times 5}$, is it possible to find all its eigenvalues exactly? Motivate your answers.

2. What is Householder reflections? How could Householder reflections be used to perform a QR-factorization of a given matrix A?

3. What does it mean for two matrices $A, B \in \mathbb{R}^{d\times d}$ to be similar? Show that if A and B are similar, then they have the same eigenvalues.

4. What is a Schur factorization of a matrix $A \in \mathbb{R}^{n\times n}$?

5. Prove that every square matrix A has a Schur factorization.

6. Write down the power method algorithm.

7. Show how the power method can be used to find an eigenvector of a given matrix.

8. Write down the inverse iteration algorithm.

9. What is the relation between the the inverse iteration and power method algorithms?

10. Use Gram-Schmidt orthogonalization to find two vectors, q_1 and q_2, that are orthonormal, and span the same subspace of $\mathbb{R}^4$ as the vectors $a_1 = (1, 2, 3, 4)$ and $a_2 = (0, 2, 0, 3)$. (The actual numbers will differ at the exam.)

11. Write down the QR Algorithm.

12. Show that two consecutive iterates $A^{(k)}$ and $A^{(k+1)}$ of the QR Algorithm are similar. (And hence have the same set of eigenvalues.)

13. Describe in what respect the Simultaneous Iteration and QR Algorithms are equivalent.

14. Write down the QR Algorithm with shifts. (It is not necessary to specify here what shifts are chosen.)

15. Let A be a symmetric matrix. Consider running shifted QR iteration with Rayleigh quotient shifts ($\mu^{(k)} = A^k_{mm}$ in every iteration), yielding a sequence $\mu^{(1)}$, $\mu^{(2)}, \ldots$ of shifts. Also run Rayleigh quotient iteration, starting with $v^{(0)} = [0, \ldots, 0, 1]^T$, yielding a sequence of Rayleigh quotients $\rho^{(1)}$, $\rho^{(2)}, \ldots$ Show that these sequences are identical: $\mu^{(k)} = \rho^{(k)}$, for all k.

16. Define the Krylov space $\mathcal{K}_n(A, b)$ generated by the matrix $A \in \mathbb{R}^{m\times m}$ and the vector $b \in \mathbb{R}^m$.

17. What is the dimension of the Krylov space $\mathcal{K}_3$ generated by the matrix

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix} \text{ and vector } b = \begin{pmatrix} 3 \\ 0 \\ 2 \end{pmatrix}?$$

Why?

18. What does the Arnoldi method compute?

19. Mention one major difference between Lanczos and Arnoldi iteration.

20. Let r_n be the residual at step n of the GMRES iteration. Show that $r_{n+1} \leq r_n$.

21. Show that

$$\|r_n\| = \min_{p_n \in P_n} \|p_n(A)b\|,$$

where r_n is the residual in step n of the GMRES method, and

$$P_n = \{\text{polynomials } p \text{ of degree} \leq n \text{ with } p(0) = 1\}.$$

22. Assume that the matrix A is diagonalizable , satisfying $A = V\Lambda V^{-1}$ for some nonsingular matrix V and diagonal matrix Λ. Show that

$$\frac{\|r_n\|}{\|b\|} \leq \kappa(V) \min_{p_n \in P_n} \|p_n\|_{\Lambda(A)},$$

where $\Lambda(A)$ is the set of eigenvalues of A, $\kappa(V)$ is the condition number of V, and $\|p_n\|_{\Lambda(A)} = \max_{z\in\Lambda(A)} |p(z)|$.

23. Assume that the matrix A is diagonalizable, $A = V\Lambda V^{-1}$, with the condition number of V, $\kappa(V) \leq 10$. Assume also that its spectrum (its eigenvalues) are contained in the disk of radius $1/2$ around $z = 2$ in the complex plane. Show that at step n of the GMRES iteration for solving $Ax = b$, where $\|b\| \leq 10$, the residual satisfies

$$\|r_n\| \leq \frac{100}{4^n}.$$

24. Assume that the matrix A is diagonalizable, $A = V\Lambda V^{-1}$, with the condition number of V, $\kappa(V) \leq 10$. Assume also that its spectrum (its eigenvalues) are contained in the disks of radii $1/2$ around $z = 2$ and $z = -2$ in the complex plane. Show that at every even step $2n$ of the GMRES iteration for solving $Ax = b$, where $\|b\| \leq 10$, the residual satisfies:

$$\|r_{2n}\| \leq 100\left(\frac{9}{16}\right)^n = 100\left(\frac{3}{4}\right)^{2n}.$$

25. What is being minimized in the Conjugate Gradient method?

26. From the definition of the Conjugate Gradient method it can be shown that $x_n \in \mathcal{K}_n$, and that the residuals satisfy $r_n \perp \mathcal{K}_n$. Use this to show that x_n is the unique point in $\mathcal{K}_n$ that minimizes $\|e_n\|_A$. (Recall also that we need A to be symmetric positive definite in order to be able to use the CG method.)

27. Show the following error bound for the Conjugate Gradient method:

$$\frac{\|e_n\|_A}{\|e_0\|_A} \leq \inf_{p \in P_n} \max_{\lambda \in \Lambda(A)} |p(\lambda)|.$$

    Here $e_n = x_* - x_n$ is the error in step n of the CG algorithm,

$$P_n = \{\text{polynomials } p \text{ of degree} \leq n \text{ with } p(0) = 1\},$$

    and $\Lambda(A)$ is the spectrum of A.

    When showing this result you may use the fact that the CG method minimizes the norm $\|e_n\|_A$ over $\mathcal{K}_n$ in each step.

28. Prove the following error bound for the Conjugate Gradient method:

$$\frac{\|e_n\|_A}{\|e_0\|_A} \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^n.$$

    Here $e_n = x_* - x_n$ is the error in step n of the CG algorithm and κ is the 2-norm condition number of the matrix A in the system $Ax = b$ being solved.

    In your proof you may use that there exists a degree n polynomial T_n (the Chebyshev polynomial) which satisfies

    - $|T_n(x)| \leq 1$, for $-1 \leq x \leq 1$,
    - $T_n(x) = \frac{1}{2}[(x + \sqrt{x^2-1})^n + (x + \sqrt{x^2-1})^{-n}]$, for $|x| > 1$.

29. Name a significant difference between the GMRES and QMR methods. Name a significant difference between the QMR and the CGS methods. Given a linear system $Ax = b$, where A is nonsingular, name a method which is certain to converge, at least in exact arithmetic. Under what conditions can we apply the Conjugate Gradient method to solve the linear system $Ax = b$?

30. What is the Jacobi method for solving $Ax = b$? What is the damped Jacobi?

31. Consider the system $Ax = b$, where

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 & 2 & -1 \\ 0 & 0 & \cdots & 0 & -1 & 2 \end{pmatrix},$$

i.e. a finite difference discretization of the Poisson equation

$$\begin{aligned} -u''(x) &= f(x), \quad 0 < x < 1, \\ u(0) = u(1) &= 0. \end{aligned}$$

Show that the damping coefficient in the damped Jacobi iteration can be chosen such that the high frequency modes of the error $e^n = x^n - x^*$ are decreased substantially each step, while the low frequency modes are decreased just a little, for every choice of the damping $0 \leq \omega \leq 1$. (Hint: you may use that A has the eigenvectors $\{w^k\}_{k=1}^{m-1}$, where $w_j^k = \sin \frac{jk\pi}{m}$, with corresponding eigenvalues $\lambda_k = 4 \sin^2 \frac{k\pi}{2m}$.)

32. Give an example of a prolongation operator.

33. Give an example of a restriction operator.

34. Write down the two-grid method.

35. Write down the Multi-grid V-cycle.

36. Write down the Multi-grid W-cycle.

37. Derive an estimate of the computational complexity of one iteration step of the multi-grid method. I.e. if seen as a function, what is the computational complexity of a call to MG on the finest level L? The work of the call to MG on the level L also contains the work of all the recursive calls to MG on the coarser levels. Explain your assumptions. Under what conditions does it hold that this work can be bounded by a constant times the size of the system on the finest scale, n_L?

38. Denote the error after m two-grid iterations e^m. It then holds that $e^{m+1} = M_h^{2h} e^m$, for some matrix M_h^{2h}. Prove this. Also state how M_h^{2h} can look, in terms of smoothing operators, prolongation etc. Prove your claim.

39. Denote the error after m γ-Multi-grid iterations e^m. It then holds that $e^{m+1} = M_h e^m$, for some matrix M_h. Prove this. Also state and prove a recursion relation showing how M_l can be computed using M_{l-1} (here $M_l \equiv M_{h_l}$). The recursion relation may include smoothing operators, prolongation etc.

40. Describe in what sense it could be said that FMG is more computationally efficient than MGI.

41. State and prove a theorem saying that a Multi-grid iteration reduces the error by a factor that is independent of the step size.

42. Explain why, in order to show error reduction of multi-grid iterations, one needs to show error reduction of the two-grid method.

43. Nested iteration does what?

44. Write down the full multi-grid algorithm.

45. State and prove a theorem on the computational complexity of the full multi-grid method.

46. State and prove a theorem about the full multi-grid method achieving accuracy comparable to the discretization error.

47. Describe how the multi-grid method could be of use when solving the heat equation numerically.

48. Describe two different ways, both using multi-grid, to solve the equation

$$\begin{aligned} -u'' + e^u &= f, \quad \text{in } (0,1), \\ u(0) = u(1) &= 0. \end{aligned}$$

49. Let A be a full $n \times n$-matrix with $\operatorname{rank}(A) = k \ll n$. Propose and describe an efficient algorithm for computing the matrix-vector product Ax and estimate its computational complexity.

50. Let A be the $N \times N$ matrix with entries $a_{ij} = 1/|i-j|$ when $i \neq j$, and $a_{ij} = 0$ for $i = j$. Present an algorithm for efficient computation of the matrix-vector product Ax (with some given vector x).

    You may use that if a submatrix A_{lmk} of A is separated from the diagonal, then there exists a $k \times k$-matrix B_{lmk} with rank $J+1$ such that

    $$|(A_{lmk})_{ij} - (B_{lmk})_{ij}| \leq 4^{-J}.$$

    Here A_{lmk} denotes a submatrix of A containing all elements a_{ij} of A such that $l \leq i < l+k$ and $m \leq j < m+k$. Moreover, B_{lmk} can be computed explicitly as

    $$B_{lmk} = \bar{B}^T \bar{A} \bar{\Gamma},$$

    where $\bar{B}$ and $\bar{\Gamma}$ are $(J+1) \times k$-matrices, and $\bar{A}$ is an $(J+1) \times (J+1)$-matrix.

    What will be the computational complexity of your proposed algorithm? How is this complexity shown?