26. Let H be an $n \times n$ upper Hessenberg matrix and $b = e_1$ be the first n-dimensional unit vector, i.e. $b = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}^\mathsf{T}$.
    (a) What is the structure of the vectors $H^i b$ for $i = 0, \dots, n-1$?
    (b) What are the resulting Krylov spaces $\mathcal{K}_1, \dots, \mathcal{K}_n$ for H and b?
    (c) What vectors q_i ($i = 1, \dots, n$) and what matrices $\hat{H}_i$ will the Arnoldi iteration for H and b produce (up to the sign)?

27. Compute (by hand) the **Ritz numbers**
    - $\theta_1^{(1)}$,
    - $\theta_1^{(2)}$ and $\theta_2^{(2)}$, and
    - $\theta_1^{(3)}$, $\theta_2^{(3)}$ and $\theta_3^{(3)}$

    for the upper Hessenberg matrix

$$A = \begin{pmatrix} 4 & 1 & 0 \\ -2 & 0 & -2 \\ 0 & 1 & 4 \end{pmatrix}$$

    from examples 1 (b) and 3 and the vector $b = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^\mathsf{T}$.
    Hint: Use the results from example 26 (c) and from example 3.

28. (a) Write a program in MATLAB or Octave which takes as input parameters:
    - a square matrix A,
    - a starting vector b (of the same dimension as A) and
    - a number of steps

    and performs the **Arnoldi iteration**.
    (b) Extend your program so that it also produces the **Ritz numbers** (which approximate the eigenvalues of the matrix A from the output of the Arnoldi iteration), at least in the case where all the eigenvalues are real. (You can reuse your algorithms from exercise sheet 2.)
    (c) Verify your result from example 27 using your program.
    (d) Apply the program to the matrix

$$\begin{pmatrix} 13 & 4 & 3 & 9 \\ -1 & -8 & 5 & 0 \\ 2 & 3 & 7 & 1 \\ 6 & -2 & 0 & 4 \end{pmatrix}$$

    from example 7 with the starting vector $\begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}^\mathsf{T}$ (do all 4 iteration steps). Also test it on some larger matrices.

29. Write a program in MATLAB or Octave which takes as input parameters:

    - a square matrix A,
    - a vector b (of the same dimension as A) and
    - a number of steps

    and performs the **GMRES iteration** for $Ax = b$. For testing, compare the output of your algorithm with the builtin `A\b` operator on some arbitrary testcases.
    Hint: There is a misprint in the lecture transcript: In step m, you have to execute step $\boldsymbol{m}$ of the Arnoldi iteration, not step n.

30. (a) Write a program in MATLAB or Octave which takes as input parameters:

    - a symmetric matrix A,
    - a starting vector b (of the same dimension as A) and
    - a number of steps

    and performs the **Lanczos iteration**.

    (b) Extend your program so that it also produces the **Ritz numbers**, which approximate the eigenvalues of the matrix A from the output of the Lanczos iteration. (You can reuse your algorithms from exercise sheet 2.)

    (c) Apply the program to the matrix

    $$\begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 4 & 3 & 2 \\ 2 & 3 & 4 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix}.$$

    from example 11 with the starting vector $\begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}^\mathsf{T}$ (do all 4 iteration steps). Also test it on some larger symmetric matrices.

    (d) Apply the Arnoldi iteration from example 28 to the same matrix with the same starting vector and the same number of iterations and compare the results.

31. (a) Write a program in MATLAB or Octave which takes as input parameters:

    - a symmetric, positive definite matrix A,
    - a vector b (of the same dimension as A) and
    - a number of steps

    and performs the (linear) **conjugate gradient iteration** for $Ax = b$.

    (b) For testing, compare the output of your algorithm with the builtin `A\b` operator on some arbitrary (symmetric positive definite) testcases. (One way to ensure that your matrices will be be symmetric and positive definite is to build them out of the LL^T (Cholesky) form.)

    (c) What can happen if the matrix A is not positive definite? (Consider e.g. $A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix}$ and $b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$.)