10. (a) Write a program in MATLAB or Octave which transforms a matrix (given as a parameter) to **Hessenberg form** using Householder transformations.

    (b) 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.

    (c) Retry the power iteration from example 7 with the same starting vector $\begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}^\mathsf{T}$ on the matrix in Hessenberg form. Does the transformation affect convergence? (Try comparing the quality of the found eigenvectors using the formula `(A*u)./u` after 10, 20 and 100 iterations.)

    Hint: Use `format long` if the number of decimals displayed is insufficient. (The opposite is `format short`.)

11. Apply the program from example 10 to the matrix

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

    What happens to the resulting Hessenberg form? Can the algorithm be made more efficient for this case? (Try out the improvements you can think of. Do they actually make your program faster? If not, why not?)

    Hint: Execution times can be measured in MATLAB or Octave using the `tic` and `toc` commands. If the algorithm runs too fast to get meaningful timings, try repeating the computation in a simple `for` loop.

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

    - a matrix (of which we want to find an eigenvector),
    - a starting vector and
    - a constant shift $\mu$,

    and performs the **inverse iteration**.

    (b) Apply the program to the symmetric tridiagonal matrix

$$\begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

    with the starting vector $\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{pmatrix}^\mathsf{T}$ and the shift $\mu = 3.6$.

    (c) Compare the performance (iteration count required to obtain convergence, execution time) with the power iteration (example 7).

13. (a) Write a program in MATLAB or Octave which takes as input parameters:
    - a matrix (of which we want to find an eigenvector) and
    - a starting vector,

    and performs the **Rayleigh quotient iteration**.

    (b) Apply the program to the symmetric tridiagonal matrix

$$\begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

    from example 12 with the same starting vector $\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{pmatrix}^\mathsf{T}$.
    What happens numerically if you do too many iterations? Stop the iteration before the numerical problems show up.

    (c) Compare the performance (iteration count required to obtain convergence, execution time) with the power iteration (example 7) and the inverse iteration with constant shift (example 12).

14. (a) Write a program in MATLAB or Octave which takes as input parameter a matrix (of which we want to find all eigenvalues) and performs the **QR iteration without shifts** (nor deflation steps).

    (b) Apply the program to the symmetric tridiagonal matrix

$$\begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

    from example 12.

    (c) Try running your program on the non-symmetric matrices

$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

    from example 6 and

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

    from example 7. In each case: Does it converge? Why?

    (d) Try running your program on the symmetric tridiagonal matrix

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

    Does it converge? Why?

15. (a) Write a program in MATLAB or Octave which takes as input parameter a symmetric tridiagonal matrix (of which we want to find all eigenvalues) and performs the **QR iteration with Rayleigh quotient shift** and deflation steps. (You can skip the initial tridiagonalization step, because we assume the matrix to already be in tridiagonal form.)

    Hint: For the deflation step, you have to compare the **absolute value** of $A_{j,j+1}$ with $\varepsilon$, because that entry of the matrix can also be negative. (The absolute value signs are missing in the lecture transcript.)

    (b) Apply the program to the symmetric tridiagonal matrix

$$\begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

    from example 12.

    (c) Compare the performance (iteration count required to obtain convergence, execution time) with the QR iteration without shift (example 14).

16. (a) Write a program in MATLAB or Octave which takes as input parameter a symmetric tridiagonal matrix (of which we want to find all eigenvalues) and performs the **QR iteration with Wilkinson shift** and deflation steps. (You can skip the initial tridiagonalization step, because we assume the matrix to already be in tridiagonal form.)

    Hint: The formula for the Wilkinson shift in the lecture transcript has a misprint. The $A_{n-1,n-1}$ in the denominator should be $A_{n,n-1}$, i.e.

    $$\mu = A_{n,n} - \frac{\operatorname{sgn}(\delta) A_{n,n-1}^2}{|\delta| + \sqrt{\delta^2 + A_{n,n-1}^2}}.$$

    Also see the note about the deflation step in example 15 (a).

    (b) Apply the program to the symmetric tridiagonal matrix

    $$\begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

    from example 12.

    (c) Compare the performance (iteration count required to obtain convergence, execution time) with the QR iterations without shift (example 14) and with Rayleigh quotient shift (example 15).

    (d) Try running the algorithm on the symmetric tridiagonal matrix

    $$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

    from example 14 (d). Does it converge?

    (e) Try running the algorithm on the non-symmetric tridiagonal matrix

    $$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

    from example 6. Does it converge? Why?