9 Overview of Eigenvalue Algorithms

As already mentioned earlier, any eigenvalue algorithm needs to be an *iterative* one. Some not so good approaches are:

1. Compute the roots of the characteristic polynomial. This is usually a very ill-conditioned problem, and should be used only for small pencil and paper calculations.

2. The power iteration listed in most textbooks yields only the largest eigenvalue, and usually converges very slowly. We will, however, consider this method as the basis for more efficient methods later.

More promising approaches are those based on the linear algebra facts reviewed earlier such as diagonalization or triangularization. In particular, the Schur factorization reveals *all* eigenvalues for *any* square matrix. Another advantage of this sort of approach is that there will be some similarities to the matrix factorization algorithms studied earlier.

While it is impractical to compute the *exact* Schur factorization, we can use the following iterative approach to get close:

$$Q_j^* \dots Q_2^* Q_1^* A Q_1 Q_2 \dots Q_j \to T \qquad \text{for } j \to \infty.$$

Here the Q_j should ideally be "simple" unitary matrices, and T upper triangular.

As mentioned earlier, the eigenvalues of real matrices can be complex, so the algorithms we devise will have to be able to handle the complex case, also.

9.1 A General Two-Phase Procedure

In order to produce a sequence converging to an upper triangular T one usually attempts a two-phase procedure:

1. Convert A to *upper Hessenberg* form.

2. Produce a sequence of upper Hessenberg matrices that converges to an upper triangular T.

Here a matrix is *upper Hessenberg* if $H_{ij} = 0$ for $i > j + 1$, i.e., it is an upper triangular matrix with additional nonzero entries on the sub-diagonal:

$$H = \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \end{bmatrix}.$$

We will go into more details later.

9.2 Why Use a Two-Phase Approach?

The main reason is that it turns out to be more efficient. The operations count for phase 1 can be shown to be $\mathcal{O}(m^3)$. Usually one performs $\mathcal{O}(m)$ iterations of phase 2 at a cost of $\mathcal{O}(m^2)$ each. This leads to a total cost of $\mathcal{O}(m^3)$.

A more simplistic approach would require $\mathcal{O}(m)$ (or more) iterations at a cost of $\mathcal{O}(m^3)$ each for a total of $\mathcal{O}(m^4)$.

Remark If A is Hermitian, then phase 1 will produce a tridiagonal matrix (which is both upper and lower Hessenberg) so that phase 2 can be completed at a total of $\mathcal{O}(m^2)$ operations.

9.3 Reduction to Hessenberg or Tridiagonal Form

Our goal is to produce a Schur factorization

$$A = QTQ^*$$

with unitary Q and upper triangular T.

In our first attempt we use Householder reflections to attempt to immediately produce the upper triangular matrix T, i.e.,

$$A = \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \quad \longrightarrow \quad Q_1^* A = \begin{bmatrix} x & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \\ 0 & x & x & x \end{bmatrix}.$$

This looks promising. However, we need to perform *similarity transforms* (in order to preserve the eigenvalues of A), so we also need to right-multiply by Q_1. In general, that results in

$$Q_1^* A Q_1 = \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix},$$

and we are back to where we started from. This is illustrated in the Maple worksheet `473_Hessenberg.mws`.

Remark We will completely forget this approach. It will come in useful later as one possibility for phase 2.

In our second attempt we again use Householder reflectors. But this time we concentrate only on phase 1, i.e., produce an upper Hessenberg matrix only. Schematically we have

$$A = \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \quad \longrightarrow \quad Q_1^* A = \begin{bmatrix} x & x & x & x \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{bmatrix}.$$

The notation $\mathbf{x}$ vs. x indicates that the value in this position is still considered generically nonzero, but it may have changed. Note that we do not touch row 1 at all. Instead we operate only on rows 2 and below. Now right-multiplication by Q_1 yields

$$Q_1^* A Q_1 = \begin{bmatrix} x & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ x & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{bmatrix},$$

since we do not touch column 1.

In the next step we do not touch both rows 1 and 2, i.e.,

$$Q_2^* Q_1^* A Q_1 = \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{x} & \mathbf{x} \end{bmatrix},$$

and the completed similarity transform yields

$$Q_2^* Q_1^* A Q_1 Q_2 = \begin{bmatrix} x & x & \mathbf{x} & \mathbf{x} \\ x & x & \mathbf{x} & \mathbf{x} \\ 0 & x & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{x} & \mathbf{x} \end{bmatrix} = H,$$

since now columns 1 and 2 remain unchanged. This approach is also illustrated in the Maple worksheet `473_Hessenberg.mws`.

The following algorithm is very similar to the one proposed earlier for the QR factorization.

Algorithm (Householder Reduction)

for $k = 1 : (m-2)$

$\boldsymbol{x} = A(k+1:m, k)$ (instead of $A(k:m,m)$ for QR)

$\boldsymbol{v}_k = \text{sign}(\boldsymbol{x}(1))\|\boldsymbol{x}\|_2 \boldsymbol{e}_1 + \boldsymbol{x}$

$\boldsymbol{v}_k = \boldsymbol{v}_k / \|\boldsymbol{v}_k\|_2$

$A(k+1:m, k:m) = A(k+1:m, k:m) - 2\boldsymbol{v}_k \left(\boldsymbol{v}_k^* A(k+1:m, k:m)\right)$

$A(1:m, k+1:m) = A(1:m, k+1:m) - 2\left(A(1:m, k+1:m)\boldsymbol{v}_k\right)\boldsymbol{v}_k^*$

end

The last line of the algorithm is new and reflects the right-multiplication needed for the similarity transformation.

Remark As for the QR factorization algorithm the matrices Q are not explicitly formed, only the reflection vectors $\boldsymbol{v}_k$ are stored. One can use the algorithm `formQ` from the computer assignment to compute the matrix Q if needed.

The operations count is $\mathcal{O}(\frac{10}{3}m^3)$ which is more than twice the $\mathcal{O}(\frac{4}{3}m^3)$ required for the QR factorization since there are no known zeros whose computation can be avoided in the right-multiplication.

If A is Hermitian, then one can take advantage of symmetry and get a tridiagonal matrix in $\mathcal{O}(\frac{4}{3}m^3)$ operations.

Finally, it can be shown that Householder reduction to Hessenberg form is *backward stable*.