Numerical Analysis - Part II

Anders C. Hansen

Lecture 24

Eigenvalues and eigenvectors

Spectrum preserved under similarity transforms

Theorem 1

Let A and S be $n \times n$ matrices, S being nonsingular. Then $\mathbf{w}$ is an eigenvector of A with eigenvalue λ if and only if $\widehat{\mathbf{w}} = S\mathbf{w}$ is an eigenvector of $\widehat{A} = SAS^{-1}$ with the same eigenvalue.

Proof.

$$A\mathbf{w} = \lambda\mathbf{w} \quad \Leftrightarrow \quad AS^{-1}(S\mathbf{w}) = \lambda\mathbf{w} \quad \Leftrightarrow \quad (SAS^{-1})(S\mathbf{w}) = \lambda(S\mathbf{w}).$$

□

Deflation

Suppose that we have found one solution of the eigenvector equation $A\mathbf{w} = \lambda\mathbf{w}$, where A is again $n \times n$. Then *deflation* is the task of constructing an $(n-1) \times (n-1)$ matrix, B say, whose eigenvalues are the other eigenvalues of A. Specifically, we apply a similarity transformation S to A such that the first column of $\widehat{A} = SAS^{-1}$ is λ times the first coordinate vector $\mathbf{e}_1$, because it follows from the characteristic equation for eigenvalues and from Theorem 1 that we can let B be the bottom right $(n-1) \times (n-1)$ submatrix of $\widehat{A} = SAS^{-1}$. In particular,

$$SAS^{-1} = \widehat{A} = \begin{bmatrix} \lambda & \beta \\ 0 & B \end{bmatrix}.$$

Deflation

We write the condition on S as $(SAS^{-1})\mathbf{e}_1 = \lambda\mathbf{e}_1$. Then the last equation in the proof of Theorem 1 shows that it is sufficient if S has the property $S\mathbf{w} = c\mathbf{e}_1$, where c is any nonzero scalar.

Algorithm for deflation for symmetric A

Suppose that A is symmetric and $\mathbf{w} \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$ are given so that $A\mathbf{w} = \lambda\mathbf{w}$. We seek a nonsingular matrix S such that $S\mathbf{w} = c\mathbf{e}_1$ and such that SAS^{-1} is also symmetric. The last condition holds if S is orthogonal, since then $S^{-1} = S^T$. It is suitable to pick a *Householder reflection*, which means that S has the form

$$H_u = I - 2\mathbf{u}\mathbf{u}^T/\|\mathbf{u}\|^2, \quad \text{where} \quad \mathbf{u} \in \mathbb{R}^n.$$

Algorithm for deflation for symmetric A

Specifically, we recall from the Numerical Analysis IB course that Householder reflections are orthogonal and that, because $H_u\mathbf{u} = -\mathbf{u}$ and $H_u\mathbf{v} = \mathbf{v}$ if $\mathbf{u}^T\mathbf{v} = 0$, they reflect any vector in $\mathbb{R}^n$ with respect to the $(n-1)$-dimensional hyperplane orthogonal to $\mathbf{u}$. So, for any two vectors $\mathbf{x}$ and $\mathbf{y}$ of equal lengths,

$$H_{\mathbf{u}}\mathbf{x} = \mathbf{y}, \quad \text{where} \quad \mathbf{u} = \mathbf{x} - \mathbf{y}.$$

Hence,

$$\left(I - 2\frac{\mathbf{u}\mathbf{u}^T}{\|\mathbf{u}\|^2}\right)\mathbf{w} = \pm\|\mathbf{w}\|\mathbf{e}_1\,, \quad \text{where} \quad \mathbf{u} = \mathbf{w} \mp \|\mathbf{w}\|\mathbf{e}_1\,.$$

Since the bottom $n-1$ components of $\mathbf{u}$ and $\mathbf{w}$ coincide, the calculation of $\mathbf{u}$ requires only $\mathcal{O}(n)$ computer operations. Further, the calculation of SAS^{-1} can be done in only $\mathcal{O}(n^2)$ operations, taking advantage of the form $S = I - 2\mathbf{u}\mathbf{u}^T/\|\mathbf{u}\|^2$, even if all the elements of A are nonzero.

Algorithm for deflation for symmetric A

After deflation, we may find an eigenvector, $\widehat{\mathbf{w}}$ say, of SAS^{-1}. Then the new eigenvector of A, according to Theorem 1, is $S^{-1}\widehat{\mathbf{w}} = S\widehat{\mathbf{w}}$, because Householder matrices, like all symmetric orthogonal matrices, are *involutions*: $S^2 = I$.

Givens rotations

The notation $\Omega^{[i,j]}$ denotes the following $n \times n$ matrix

$$\Omega^{[i,j]} = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & c & & s & & \\ & & & \ddots & & & \\ & & -s & & c & & \\ & & & & & \ddots & \\ & & \uparrow & & \uparrow & & 1 \\ & & i & & j & & \end{bmatrix}, \quad c^2 + s^2 = 1.$$

Generally, for any vector $\mathbf{a}_k \in \mathbb{R}^n$, we can find a matrix $\Omega^{[i,j]}$ such that

$$\Omega^{[i,j]}\mathbf{a} = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & c & & s & & \\ & & & \ddots & & & \\ & & -s & & c & & \\ & & & & & \ddots & \\ & & \uparrow & & \uparrow & & 1 \\ & & i & & j & & \end{bmatrix} \begin{bmatrix} a_{1k} \\ \vdots \\ a_{ik} \\ \vdots \\ a_{jk} \\ \vdots \\ a_{nk} \end{bmatrix} = \begin{bmatrix} a_{1k} \\ \vdots \\ r \\ \vdots \\ 0 \\ \vdots \\ a_{nk} \end{bmatrix} \begin{matrix} \\ \\ \leftarrow i \\ \\ \leftarrow j \\ \\ \\ \end{matrix}$$

$$c = \frac{a_{ik}}{\sqrt{a_{ik}^2 + a_{jk}^2}},$$

$$s = \frac{a_{jk}}{\sqrt{a_{ik}^2 + a_{jk}^2}},$$

$$r = \sqrt{a_{ik}^2 + a_{jk}^2}.$$

Givens rotations

1) We can choose $\Omega^{[i,j]}$ so that any prescribed element $\widetilde{a}_{jk}$ in the j-th row of $\widetilde{A} = \Omega^{[i,j]} A$ is zero.
2) The rows of $\widetilde{A} = \Omega^{[i,j]} A$ are the same as the rows of A, except that the i-th and j-th rows of the product are linear combinations of the i-th and j-th rows of A.
3) The columns of $\widehat{A} = \widetilde{A} \Omega^{[i,j]\,T}$ are the same as the columns of $\widetilde{A}$, except that the i-th and j-th columns of $\widehat{A}$ are linear combinations of the i-th and j-th columns of $\widetilde{A}$.
4) $\Omega^{[i,j]}$ is an orthogonal matrix, thus $\widehat{A} = \Omega^{[i,j]} A \Omega^{[i,j]\,T}$ inherits the eigenvalues of A.
5) If A is symmetric, then so is $\widehat{A}$.

Transformation to upper Hessenberg – Givens

Transformation to an upper Hessenberg form: We replace A by $\widehat{A} = SAS^{-1}$, where S is a product of Givens rotations $\Omega^{[i,j]}$ chosen to annihilate subsubdiagonal elements $a_{j,i-1}$ in the $(i-1)$-st column:

$$\begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \overset{\Omega^{[2,3]}\times}{\longrightarrow} \begin{bmatrix} * & * & * & * \\ \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet \\ * & * & * & * \end{bmatrix} \overset{\times\Omega^{[2,3]T}}{\longrightarrow} \begin{bmatrix} * & \bullet & \bullet & * \\ * & \bullet & \bullet & * \\ 0 & \bullet & \bullet & * \\ * & \bullet & \bullet & * \end{bmatrix} \overset{\Omega^{[2,4]}\times}{\longrightarrow} \begin{bmatrix} * & * & * & * \\ \bullet & \bullet & \bullet & \bullet \\ 0 & * & * & * \\ 0 & \bullet & \bullet & \bullet \end{bmatrix} \overset{\times\Omega^{[2,4]T}}{\longrightarrow} \begin{bmatrix} * & \bullet & * & \bullet \\ * & \bullet & * & \bullet \\ 0 & \bullet & * & \bullet \\ 0 & \bullet & * & \bullet \end{bmatrix} \overset{\Omega^{[3,4]}\times}{\longrightarrow} \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ 0 & \bullet & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet \end{bmatrix} \overset{\times\Omega^{[3,4]T}}{\longrightarrow} \begin{bmatrix} * & * & \bullet & \bullet \\ * & * & \bullet & \bullet \\ 0 & * & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet \end{bmatrix}$$

The $\bullet$-elements have changed through a single transformation while the $*$-elements remained the same.

It is seen that every element that we have set to zero remains zero, and the final outcome is indeed an upper Hessenberg matrix. If A is symmetric then so will be the outcome of the calculation, hence it will be tridiagonal. In general, the cost of this procedure is $\mathcal{O}(n^3)$.

Transformation to upper Hessenberg – Householder

Alternatively, we can transform A to upper Hessenberg using *Householder reflections*, rather than Givens rotations. In that case we deal with a column at a time, taking $\mathbf{u}$ such that, with $H_u = I - 2\mathbf{u}\mathbf{u}^T/\|\mathbf{u}\|^2$, the i-th column of $\widetilde{B} = H_u B$ is consistent with the upper Hessenberg form. Such a $\mathbf{u}$ has its first i coordinates vanishing, therefore $\widehat{B} = \widetilde{B} H_u^T$ has the first i columns unchanged, and all new and old zeros (which are in the first i columns) stay untouched.

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \xrightarrow{H_1\times} \begin{bmatrix} * & * & * & * & * \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet & \bullet \end{bmatrix} \xrightarrow{\times H_1^T} \begin{bmatrix} * & \bullet & \bullet & \bullet & \bullet \\ * & \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet & \bullet \end{bmatrix} \xrightarrow{H_2\times} \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & \bullet & \bullet & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet & \bullet \end{bmatrix} \xrightarrow{\times H_2^T} \begin{bmatrix} * & * & \bullet & \bullet & \bullet \\ * & * & \bullet & \bullet & \bullet \\ 0 & * & \bullet & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet & \bullet \end{bmatrix} \xrightarrow{H_3\times} \begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & \bullet & \bullet & \bullet \\ 0 & 0 & 0 & \bullet & \bullet \end{bmatrix} \xrightarrow{\times H_3^T} \begin{bmatrix} * & * & * & \bullet & \bullet \\ * & * & * & \bullet & \bullet \\ 0 & * & * & \bullet & \bullet \\ 0 & 0 & * & \bullet & \bullet \\ 0 & 0 & 0 & \bullet & \bullet \end{bmatrix}$$

The QR algorithm

The "plain vanilla" version of the QR algorithm is as follows. Set $A_0 = A$. For $k = 0, 1, \ldots$ calculate the QR factorization $A_k = Q_k R_k$ (here Q_k is $n \times n$ orthogonal and R_k is $n \times n$ upper triangular) and set $A_{k+1} = R_k Q_k$. The eigenvalues of A_{k+1} are the same as the eigenvalues of A_k, since we have

$$A_{k+1} = R_k Q_k = Q_k^{-1}(Q_k R_k)Q_k = Q_k^{-1} A_k Q_k, \tag{1}$$

a similarity transformation. Moreover, $Q_k^{-1} = Q_k^T$, therefore if A_k is symmetric, then so is A_{k+1}.
If for some $k \geq 0$ the matrix A_{k+1} can be regarded as "deflated", i.e. it has the block form

$$A_{k+1} = \begin{bmatrix} B & C \\ D & E \end{bmatrix},$$

where B, E are square and $D \approx 0$, then we calculate the eigenvalues of B and E separately (again, with QR, except that there is nothing to calculate for 1×1 and 2×2 blocks). As it turns out, such a "deflation" occurs surprisingly often.

The QR iteration for upper Hessenberg matrices

If A_k is upper Hessenberg, then its QR factorization by means of the Givens rotations produces the matrix

$$R_k = Q_k^T A_k = \Omega^{[n-1,n]} \dots \Omega^{[2,3]} \Omega^{[1,2]} A_k \,,$$

which is upper triangular. The QR iteration sets $A_{k+1} = R_k Q_k = R_k \Omega^{[1,2]T} \Omega^{[2,3]T} \dots \Omega^{[n-1,n]T}$, and it follows that A_{k+1} is also upper Hessenberg, because

$$\begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \end{bmatrix} \overset{\times \Omega^{[1,2]T}}{\longrightarrow} \begin{bmatrix} \bullet & \bullet & * & * \\ \bullet & \bullet & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \end{bmatrix} \overset{\times \Omega^{[2,3]T}}{\longrightarrow} \begin{bmatrix} * & \bullet & \bullet & * \\ * & \bullet & \bullet & * \\ 0 & \bullet & \bullet & * \\ 0 & 0 & 0 & * \end{bmatrix} \overset{\times \Omega^{[3,4]T}}{\longrightarrow} \begin{bmatrix} * & * & \bullet & \bullet \\ * & * & \bullet & \bullet \\ 0 & * & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet \end{bmatrix}$$

Thus a strong advantage of bringing A to the upper Hessenberg form initially is that then, in every iteration in QR algorithm, Q_k is a product of just $n-1$ Givens rotations. Hence each iteration of the QR algorithm requires just $\mathcal{O}(n^2)$ operations.

The QR iteration for symmetric matrices

We bring A to the upper Hessenberg form, so that the QR algorithm commences from a symmetric tridiagonal matrix A_0, and then the technique on the previous slide is applied for every k as before. Since both the upper Hessenberg structure and symmetry is retained, each A_{k+1} is also *symmetric tridiagonal* too.

It follows that, whenever a Givens rotation $\Omega^{[i,j]}$ combines either two adjacent rows or two adjacent columns of a matrix, the total number of nonzero elements in the new combination of rows or columns is at most five. Thus there is a bound on the work of each rotation that is independent of n. Hence each QR iteration requires just $\mathcal{O}(n)$ operations.

Notation

To analyse the matrices A_k that occur in the QR algorithm 5.13, we introduce

$$\bar{Q}_k = Q_0 Q_1 \cdots Q_k, \qquad \bar{R}_k = R_k R_{k-1} \cdots R_0, \qquad k = 0, 1, \ldots. \quad (2)$$

Note that $\bar{Q}_k$ is orthogonal and $\bar{R}_k$ upper triangular.

Lemma 2 (Fundamental properties of $\bar{Q}_k$ and $\bar{R}_k$)

A_{k+1} is related to the original matrix A by the similarity transformation $A_{k+1} = \bar{Q}_k^T A \bar{Q}_k$. Further, $\bar{Q}_k \bar{R}_k$ is the QR factorization of A^{k+1}.

Proof. We prove the first assertion by induction. By (1), we have $A_1 = Q_0^T A_0 Q_0 = \bar{Q}_0^T A \bar{Q}_0$. Assuming $A_k = \bar{Q}_{k-1}^T A \bar{Q}_{k-1}$, equations (1)-(2) provide the first indentity

$$A_{k+1} = Q_k^T A_k Q_k = Q_k^T (\bar{Q}_{k-1}^T A \bar{Q}_{k-1}) Q_k = \bar{Q}_k^T A \bar{Q}_k \, .$$

The second assertion is true for $k = 0$, since $\bar{Q}_0 \bar{R}_0 = Q_0 R_0 = A_0 = A$. Again, we use induction, assuming $\bar{Q}_{k-1} \bar{R}_{k-1} = A^k$. Thus, using the definition (2) and the first statement of the lemma, we deduce that

$$\begin{aligned} \bar{Q}_k \bar{R}_k &= (\bar{Q}_{k-1} Q_k)(R_k \bar{R}_{k-1}) = \bar{Q}_{k-1} A_k \bar{R}_{k-1} = \bar{Q}_{k-1} (\bar{Q}_{k-1}^T A \bar{Q}_{k-1}) \bar{R}_{k-1} \\ &= A \bar{Q}_{k-1} \bar{R}_{k-1} = A \cdot A^k = A^{k+1} \end{aligned}$$

and the lemma is true. □

Relation between QR and the power method

Assume that the eigenvalues of A have different magnitudes,

$$|\lambda_1| < |\lambda_2| < \cdots < |\lambda_n|, \quad \text{and let} \quad \mathbf{e}_1 = \sum_{i=1}^{n} c_i \mathbf{w}_i = \sum_{i=1}^{m} c_i \mathbf{w}_i \tag{3}$$

be the expansion of the first coordinate vector in terms of the normalized eigenvectors of A, where m is the greatest integer such that $c_m \neq 0$.

Relation between QR and the power method

Consider the first columns of both sides of the matrix equation

$$A^{k+1} = \bar{Q}_k \bar{R}_k.$$

By the power method arguments, the vector $A^{k+1}\mathbf{e}_1$ is a multiple of $\sum_{i=1}^{m} c_i(\lambda_i/\lambda_m)^{k+1}\mathbf{w}_i$, so the first column of A^{k+1} tends to be a multiple of $\mathbf{w}_m$ for $k \gg 1$. On the other hand, if $\mathbf{q}_k$ is the first column of $\bar{Q}_k$, then, since $\bar{R}_k$ is upper triangular, the first column of $\bar{Q}_k\bar{R}_k$ is a multiple of $\mathbf{q}_k$.
Therefore $\mathbf{q}_k$ tends to be a multiple of $\mathbf{w}_m$. Further, because both $\mathbf{q}_k$ and $\mathbf{w}_m$ have unit length, we deduce that $\mathbf{q}_k = \pm\mathbf{w}_m + \mathbf{h}_k$, where $\mathbf{h}_k$ tends to zero as $k \to \infty$. Therefore,

$$A\mathbf{q}_k = \lambda_m\mathbf{q}_k + o(\mathbf{1}), \quad k \to \infty. \tag{4}$$

Theorem 3 (The first column of A_k)

Let conditions (3) be satisfied. Then, as $k \to \infty$, the first column of A_k tends to $\lambda_m \mathbf{e}_1$, making A_k suitable for deflation.

Proof. By Lemma 2, the first column of A_{k+1} is $\bar{Q}_k^T A \bar{Q}_k \mathbf{e}_1$, and, using (4), we deduce that

$$A_{k+1}\mathbf{e}_1 = \bar{Q}_k^T A \bar{Q}_k \mathbf{e}_1 = \bar{Q}_k^T A \mathbf{q}_k \overset{(4))}{=} \bar{Q}_k^T [\lambda_m \mathbf{q}_k + o(\mathbf{1})] \overset{(*)}{=} \lambda_m \mathbf{e}_1 + o(\mathbf{1}),$$

where in $(*)$ we used that $\bar{Q}_k^T \mathbf{q}_k = \mathbf{e}_1$ by orthogonality of $\bar{Q}$, and that $\|\bar{Q}_k \mathbf{x}\|_2 = \|\mathbf{x}\|_2$ because an orthogonal mapping is an isometry. □

Relation between QR and inverse iteration

In practice, the statement of Theorem 3 is hardly ever important, because usually, as $k \to \infty$, the off-diagonal elements in the bottom row of A_{k+1} tend to zero *much faster* than the off-diagonal elements in the first column. The reason is that, besides the connection with the power method, the QR algorithm also enjoys a close relation with *inverse iteration*.

Let again

$$|\lambda_1| < |\lambda_2| < \cdots < |\lambda_n|, \quad \text{and let} \quad \mathbf{e}_n^T = \sum_{i=1}^{n} c_i \mathbf{v}_i^T = \sum_{i=s}^{n} c_i \mathbf{v}_i^T \tag{5}$$

be the expansion of the last coordinate row vector $\mathbf{e}_n^T$ in the basis of normalized *left eigenvectors* of A, i.e. $\mathbf{v}_i^T A = \lambda_i \mathbf{v}_i^T$, where s is the least integer such that $c_s \neq 0$.

Relation between QR and inverse iteration

Assuming that A is nonsingular, we can write the equation $A^{k+1} = \bar{Q}_k \bar{R}_k$ in the form $A^{-(k+1)} = \bar{R}_k^{-1} \bar{Q}_k^T$. Consider the bottom rows of both sides of this equation: $\mathbf{e}_n^T A^{-(k+1)} = (\mathbf{e}_n^T \bar{R}_k^{-1}) \bar{Q}_k^T$.
By the inverse iteration arguments, the vector $\mathbf{e}_n^T A^{-(k+1)}$ is a multiple of $\sum_{i=s}^{n} c_i (\lambda_s/\lambda_i)^{k+1} \mathbf{v}_i^T$, so the bottom row of $A^{-(k+1)}$ tends to be multiple of $\mathbf{v}_s^T$. On the other hand, let $\mathbf{p}_k^T$ be the bottom row of $\bar{Q}_k^T$. Since $\bar{R}_k$ is upper triangular, its inverse $\bar{R}_k^{-1}$ is upper triangular too, hence the bottom row of $\bar{R}_k^{-1} \bar{Q}_k^T$, is a multiple of $\mathbf{p}_k^T$.
Therefore, $\mathbf{p}_k^T$ tends to a multiple of $\mathbf{v}_s^T$, and, because of their unit lengths, we have $\mathbf{p}_k^T = \pm \mathbf{v}_s^T + \mathbf{h}_k^T$, where $\mathbf{h}_k \to 0$, i.e.,

$$\mathbf{p}_k^T A = \lambda_s \mathbf{p}_k^T + o(\mathbf{1}), \quad k \to \infty. \tag{6}$$

The bottom row of A_k

Theorem 4 (The bottom row of A_k)

Let conditions (5) be satisfied. Then, as $k \to \infty$, the bottom row of A_k tends to $\lambda_s \mathbf{e}_n^T$, making A_k suitable for deflation.

Proof. By Lemma 2, the bottom row of A_{k+1} is $\mathbf{e}_n^T \bar{Q}_k^T A \bar{Q}_k$, and similarly to the previous proof we obtain

$$\mathbf{e}_n^T A_{k+1} = \mathbf{e}_n^T \bar{Q}_k^T A \bar{Q}_k = \mathbf{p}_k^T A \bar{Q}_k \overset{(6)}{=} [\lambda_s \mathbf{p}_k^T + o(\mathbf{1})] \bar{Q}_k = \lambda_s \mathbf{e}_n^T + o(\mathbf{1}). \tag{7}$$

the last equality by orthogonality of $\bar{Q}_k$. □

Single shifts

As we saw in previous lectures, there is a huge difference between power iteration and inverse iteration: the latter can be accelerated arbitrarily through the use of shifts. The better we can estimate $s_k \approx \lambda_s$, the more we can accomplish by a step of inverse iteration with the shifted matrix $A_k - s_k I$. Theorem 4 shows that the bottom right element $(A_k)_{nn}$ becomes a good estimate of λ_s. So, in the *single shift technique*, the matrix A_k is replaced by $A_k - s_k I$, where $s_k = (A_k)_{nn}$, before the QR factorization:

$$\begin{aligned} A_k - s_k I &= Q_k R_k, \\ A_{k+1} &= R_k Q_k + s_k I. \end{aligned}$$

Single shifts

A good approximation $s_k = (A_k)_{nn}$ to the eigenvalue λ_s generates even better approximation of $s_{k+1} = (A_{k+1})_{nn}$ to λ_s, and convergence is accelerating at a higher and higher rate (it will be the so-called cubic convergence $|\lambda_s - s_{k+1}| \leq \gamma\,|\lambda_s - s_k|^3$). Note that, similarly to the original QR iteration, we have

$$A_{k+1} = Q_k^T(Q_k R_k + s_k I)Q_k = Q_k^T A_k Q_k\,,$$

hence $A_{k+1} = \bar{Q}_k^T A \bar{Q}_k$, but note also that $\bar{Q}_k \bar{R}_k \neq A^{k+1}$, but we have instead

$$\bar{Q}_k \bar{R}_k = \prod_{m=0}^{k}(A - s_m I)$$