Section 6.3.4: Shifted QR Iteration

Jim Lambers

March 29, 2021

- The efficiency of the QR Iteration for computing the eigenvalues of an $n \times n$ matrix A is significantly improved by first reducing A to a Hessenberg matrix H, so that only $O(n^2)$ operations per iteration are required, instead of $O(n^3)$.
- However, the iteration can still converges very slowly, so additional modifications are needed to make the QR Iteration a practical algorithm for computing the eigenvalues of a general matrix.

Single Shift Strategy

- In general, the pth subdiagonal entry of H converges to zero at the rate

$$\left|\frac{\lambda_{p+1}}{\lambda_p}\right|,$$

  where λ_p is the pth largest eigenvalue of A in magnitude.
- It follows that convergence can be particularly slow if eigenvalues are very close to one another in magnitude.
- Suppose that we *shift* H by a scalar μ, meaning that we compute the QR Factorization of $H - \mu I$ instead of H.
- Then, we update H to obtain a new Hessenberg $\tilde{H}$ by multiplying the QR factors in reverse order as before, but then adding μI.
- The algorithm is as follows.

Algorithm. (Shifted QR Iteration) Given a $A \in \mathbb{R}^{n\times n}$, the following algorithm computes the Real Schur Decomposition $A = QTQ^T$ of A.

Use Hessenberg Reduction to compute Q_0 so that $H_1 = Q_0^T A Q_0$ is upper Hessenberg
for $k = 1, 2, \ldots$ **do** until convergent
  Choose a shift μ_k
  $H_k - \mu_k I = Q_k R_k$ (QR Factorization)
  $H_{k+1} = R_k Q_k + \mu_k I$
end for

- During each iteration, we have

$$\begin{aligned} H_{k+1} &= R_k Q_k + \mu_k I \\ &= Q_k^T (H_k - \mu_k I) Q_k + \mu_k I \\ &= Q_k^T H_k Q_k - \mu_k Q_k^T Q_k + \mu_k I \\ &= Q_k^T H_k Q_k - \mu_k I + \mu_k I \\ &= Q_k^T H_k Q_k. \end{aligned} \tag{1}$$

- So, we are still performing an orthogonal similarity transformation of H_k, but with a different Q_k.
- It follows that the convergence rate becomes $|\lambda_{p+1} - \mu_k|/|\lambda_p - \mu_k|$.
- Therefore, if μ_k is close to an eigenvalue, convergence of a particular subdiagonal entry will be much more rapid.
- In fact, it can be shown that if H_k is unreduced, and μ_k is an exact eigenvalue of H_k, then $[H_{k+1}]_{n,n-1} = 0$; that is, decoupling occurs in just one iteration.
- If μ_k is not an eigenvalue of H_k, but is still close to an eigenvalue, then $H_k - \mu_k I$ is nearly singular, which means that its columns are nearly linearly dependent.
- It follows that $[R_k]_{nn}$ is small, and it can be shown that $[H_{k+1}]_{n,n-1}$ is also small, and $[H_{k+1}]_{nn} \approx \mu_k$.
- Therefore, the problem is nearly decoupled, and μ_k is revealed by the structure of H_{k+1} as an approximate eigenvalue of H_k, and therefore of A as well.
- This suggests the **single shift strategy**: using $[H_k]_{nn}$ as the shift μ_k during each iteration, because if $[H_k]_{n,n-1}$ is small compared to $[H_k]_{nn}$, then this choice of shift will drive $[H_k]_{n,n-1}$ toward zero.
- In fact, it can be shown that this strategy generally causes $[H_k]_{n,n-1}$ to converge to zero *quadratically*, meaning that only a few similarity transformations are needed to achieve decoupling.
- This improvement over the linear convergence rate reported earlier is due to the changing of the shift during each step.

  Example. Consider the 2×2 matrix

$$H_k = \begin{bmatrix} a & b \\ \epsilon & 0 \end{bmatrix}, \quad \epsilon > 0,$$

  that arises naturally when using $[H_k]_{nn}$ as a shift. To compute the QR Factorization of H_k, we perform a single Givens rotation to obtain $H_k = G_k R_k$, where

$$G_k = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}, \quad c = \frac{a}{\sqrt{a^2 + \epsilon^2}}, \quad s = \frac{\epsilon}{\sqrt{a^2 + \epsilon^2}}.$$

Performing the similarity transformation $H_{k+1} = G_k^T H_k G_k$ yields

$$\begin{aligned} H_{k+1} &= \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a & b \\ \epsilon & 0 \end{bmatrix} \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \\ &= \begin{bmatrix} ac^2 + bcs + \epsilon cs & bc^2 - acs - \epsilon s^2 \\ -acs - bs^2 + \epsilon c^2 & -bcs + as^2 - \epsilon cs \end{bmatrix} \\ &= \begin{bmatrix} a + bcs & bc^2 - \epsilon \\ -bs^2 & -bcs \end{bmatrix}. \end{aligned}$$

We see that the one subdiagonal entry is

$$-bs^2 = -b\frac{\epsilon^2}{\epsilon^2 + a^2},$$

compared to the original entry ϵ. It follows that if ϵ is small compared to a and b, then subsequent QR steps will cause the subdiagonal entry to converge to zero quadratically. For example, if

$$H = \begin{bmatrix} 0.6324 & 0.2785 \\ 0.0975 & 0.5469 \end{bmatrix},$$

then the value of h_{21} after each of the first four QR steps is 0.1574, -0.0038, 2.1072×10^{-5}, and -6.931×10^{-10}. □

- The following alternative perspective is helpful for understanding why the single-shift strategy yields rapid convergence. We define

$$\hat{Q}_k = Q_1 Q_2 \cdots Q_k, \quad \hat{R}_k = R_k R_{k-1} \cdots R_1.$$

- Then it can be shown that

$$\hat{Q}_k \hat{R}_k = (H_1 - \mu_k I)(H_1 - \mu_{k-1} I) \cdots (H_1 - \mu_1 I). \tag{2}$$

- If QR Iteration converges, the columns of $\hat{Q}_k$ converge to the Schur vectors of H_1. Furthermore, from (2) we obtain

$$\prod_{i=1}^{k} (H_1 - \mu_i I)^T \hat{Q}_k = \hat{R}_k^T.$$

- That is, the columns of $\hat{Q}_k$ are the result of k iterations of **Inverse Iteration** applied to H_1^T, with shifts $\mu_1, \mu_2, \ldots, \mu_k$.

- For a general matrix A, Inverse Iteration is the Power Method applied to $(A - \mu I)^{-1}$, which converges to the eigenvector corresponding to the eigenvalue closest to μ.

- Therefore, the kth QR iteration is one iteration of the Power Method applied to $(H_1 - \mu_k I)^{-T}$.

- Inverse Iteration is very rapidly convergent when used with a shift that is an approximate eigenvalue.

- It is also worth noting that

$$\mathbf{e}_n^T \hat{Q}_k^T H_1 \hat{Q}_k \mathbf{e}_n = [H_{k+1}]_{nn}.$$

That is, the Rayleigh Quotient of H_1 and the last column of $\hat{Q}_k$ is the shift for the next iteration, which indicates that this last column is converging to the *left* eigenvector corresponding to the eigenvalue to which the shifts are converging.

Double Shift Strategy

- Unfortunately, the single shift strategy is not very effective if H has complex eigenvalues.
- An alternative is the **double shift strategy**, which is used if the two eigenvalues, μ_1 and μ_2, of the lower-right 2×2 block of H are complex.
- Then, these two eigenvalues are used as shifts in consecutive iterations to achieve quadratic convergence in the complex case as well. That is, we compute

$$\begin{aligned} H - \mu_1 I &= Q_1 R_1 \\ H_1 &= R_1 Q_1 + \mu_1 I \\ H_1 - \mu_2 I &= Q_2 R_2 \\ H_2 &= R_2 Q_2 + \mu_2 I. \end{aligned} \tag{3}$$

- To avoid complex arithmetic when using complex shifts, the **double implicit shift strategy** is used.
- We first note that because $\mu_1 = a + bi$ and $\mu_2 = a - bi$ are a complex-conjugate pair, it follows that $\mu_1 + \mu_2 = ab$ and $\mu_1 \mu_2 = a^2 + b^2$ are real. Therefore, the matrix

$$M = (H - \mu_2 I)(H - \mu_1 I) = H^2 - (\mu_1 + \mu_2)H + \mu_1 \mu_2 I$$

  is real.
- From (2), we then have

$$M = (Q_1 Q_2)(R_2 R_1).$$

  That is, $(Q_1 Q_2)(R_2 R_1)$ represents the QR Factorization of a real matrix, even though the individual matrices involved are complex.
- Furthermore,

$$\begin{aligned} H_2 &= R_2 Q_2 + \mu_2 I \\ &= Q_2^T Q_2 R_2 Q_2 + \mu_2 Q_2^T Q_2 \\ &= Q_2^T (Q_2 R_2 + \mu_2 I) Q_2 \\ &= Q_2^T H_1 Q_2 \\ &= Q_2^T (R_1 Q_1 + \mu_1 I) Q_2 \\ &= Q_2^T (Q_1^T Q_1 R_1 Q_1 + \mu_1 Q_1^T Q_1) Q_2 \\ &= Q_2^T Q_1^T (Q_1 R_1 + \mu_1 I) Q_1 Q_2 \\ &= Q_2^T Q_1^T H Q_1 Q_2. \end{aligned}$$

- That is, $Q_1 Q_2$ is the orthogonal matrix that implements the similarity transformation of H to obtain H_2.
- Therefore, we could use exclusively real arithmetic by forming M, computing its QR Factorization to obtain $M = ZR$, and then computing $H_2 = Z^T H Z$, since $Z = Q_1 Q_2$, in view of the uniqueness of the QR Factorization.
- However, M is computed by squaring H, which requires $O(n^3)$ operations. Thus the efficiency gained by first reducing A to Hessenberg form has been lost.

- We can work around this difficulty using the Implicit Q Theorem.
- Instead of forming M in its entirety, we only form its first column, which, being a second-degree polynomial of a Hessenberg matrix, has only three nonzero entries.
- We compute a Householder reflection P_0 that makes $M\mathbf{e}_1$ a multiple of $\mathbf{e}_1$.
- Then, we compute $P_0^T H P_0$, which is no longer Hessenberg, because P_0 operates on the first three rows and columns of H.
- Finally, we apply a series of Householder reflections $P_1, P_2, \ldots, P_{n-2}$ that restore Hessenberg form.
- Because the reflections $P_1, P_2, \ldots, P_{n-2}$ do not affect the first row (when applied on the left) or column (when applied on the right), it follows that if we define $\tilde{Z} = P_0 P_1 P_2 \cdots P_{n-2}$, then Z and $\tilde{Z}$ have the same first column.
- Since both matrices implement similarity transformations that preserve the Hessenberg form of H, it follows from the Implicit Q Theorem that Z and $\tilde{Z}$ are essentially equal, and that they essentially produce the same updated matrix H_2.
- This variation of a Hessenberg QR step is called a **Francis QR Step**.
- A Francis QR step requires about $10n^2$ operations, with an additional $10n^2$ operations if orthogonal transformations are being accumulated to obtain the entire Real Schur Decomposition.
- Generally, the entire QR Algorithm, including the initial reduction to Hessenberg form, requires about $10n^3$ operations, with an additional $15n^3$ operations to compute the orthogonal matrix Q such that $A = QTQ^T$ is the Real Schur Decomposition of A.
- After each QR iteration, it is necessary to check whether a subdiagonal entry of H is sufficiently small so that it can be "declared" to be zero, and therefore the eigenvalue problem can be decoupled.
- This allows the iteration to continue with smaller matrices, which substantially reduces computational expense because each iteration requires $O(p^2)$ floating-point operations when applied to a $p \times p$ matrix.