Simultaneous Inverse Iteration $\iff$ QR Algorithm

- Last lecture we showed that “pure” QR $\iff$ simultaneous iteration applied to $I$, and the first column evolves as in power iteration.
- But it is also equivalent to simultaneous inverse iteration applied to a “flipped” $I$, and the last column evolves as in inverse iteration.
- To see this, recall that $A^k = Q(k) R(k)$ with
  $$Q(k) = \prod_{j=1}^{k} Q(j) = \begin{bmatrix} q_1^{(k)} & q_2^{(k)} & \cdots & q_m^{(k)} \end{bmatrix}$$
- Invert and use that $A^{-1}$ is symmetric:
  $$A^{-k} = (R(k)^{-1} Q(k))^T = Q(k) (R(k))^{-T}$$

The Shifted QR Algorithm

- Since the QR algorithm behaves like inverse iteration, introduce shifts $\mu^{(k)}$ to accelerate the convergence:
  $$A^{(k-1)} - \mu^{(k)} I = Q(k) R(k)$$
  $$A^{(k)} = R(k) Q(k) + \mu^{(k)} I$$
- We then get (same as before):
  $$A^{(k)} = (Q(k)^T)^{-1} A^{(k-1)} Q(k) = (Q(k)^T)^{-1} A Q(k)$$
  and (different from before):
  $$(A - \mu^{(k)} I)(A - \mu^{(k-1)} I) \cdots (A - \mu^{(1)} I) = Q(k) R(k)$$
- Shifted simultaneous iteration – last column of $Q(k)$ converges quickly.

Choosing $\mu^{(k)}$: The Rayleigh Quotient Shift

- Natural choice of $\mu^{(k)}$: Rayleigh quotient for last column of $Q(k)$
  $$\mu^{(k)} = \frac{(q_m^{(k)})^T A q_m^{(k)}}{(q_m^{(k)})^T q_m^{(k)}} = (q_m^{(k)})^T A q_m^{(k)}$$
- Rayleigh quotient iteration, last column $q_m^{(k)}$ converges cubically.
- Convenient fact: This Rayleigh quotient appears as $m, m$ entry of $A^{(k)}$ since $A^{(k)} = (Q(k)^T)^{-1} A Q(k)$.
- The Rayleigh quotient shift corresponds to setting $\mu^{(k)} = A_{mm}^{(k)}$.

Choosing $\mu^{(k)}$: The Wilkinson Shift

- The QR algorithm with Rayleigh quotient shift might fail, e.g. with two symmetric eigenvalues.
- Break symmetry by the Wilkinson shift
  $$\mu = a_m - \text{sign} (\delta) b_m^2 / \left( \left| \delta \right| + \sqrt{\delta^2 + b_m^2} \right)$$
  where $\delta = (a_{m-1} - a_m) / 2$ and $B = \begin{bmatrix} a_{m-1} & b_{m-1} \\ b_{m-1} & a_m \end{bmatrix}$ is the lower-right submatrix of $A^{(k)}$.
- Always convergence with this shift, in worst case quadratically.
**A Practical Shifted QR Algorithm**

<table>
<thead>
<tr>
<th>Algorithm: “Practical” QR Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>((Q^{(0)})^T A^{(0)} Q^{(0)} = A)</td>
</tr>
<tr>
<td>for (k = 1, 2, \ldots)</td>
</tr>
<tr>
<td>Pick a shift (\mu^{(k)})</td>
</tr>
<tr>
<td>e.g., choose (\mu^{(k)} = A_{\text{trim}}^{(k-1)})</td>
</tr>
<tr>
<td>(Q^{(k)} R^{(k)} = A^{(k-1)} - \mu^{(k)} I)</td>
</tr>
<tr>
<td>QR factorization of (A^{(k-1)} - \mu^{(k)} I)</td>
</tr>
<tr>
<td>(A^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} I)</td>
</tr>
<tr>
<td>Recombine factors in reverse order</td>
</tr>
<tr>
<td>If any off-diagonal element (A_{j,j+1}^{(k)}) is sufficiently close to zero,</td>
</tr>
<tr>
<td>set (A_{j,j+1}^{(k)} = A_{j+1,j} = 0) to obtain</td>
</tr>
</tbody>
</table>
| \[
| \begin{bmatrix}
| A_1 & 0 \\
| 0 & A_2 \\
| \end{bmatrix} = A^{(k)}
| \]
| and now apply the QR algorithm to \(A_1\) and \(A_2\) |

**Stability and Accuracy**

- The QR algorithm is backward stable:
  \[
  \tilde{Q} \tilde{\Lambda} \tilde{Q}^T = A + \delta A, \quad \frac{\|\delta A\|}{\|A\|} = O(\epsilon_{\text{machine}})
  \]
  where \(\tilde{\Lambda}\) is the computed \(\Lambda\) and \(\tilde{Q}\) is an exactly orthogonal matrix.
- The combination with Hessenberg reduction is also backward stable.
- Can be shown (for normal matrices) that \(|\tilde{\lambda}_j - \lambda_j| \leq \|\delta A\|_2\), which gives
  \[
  \frac{|\tilde{\lambda}_j - \lambda_j|}{\|A\|} = O(\epsilon_{\text{machine}})
  \]
  where \(\tilde{\lambda}_j\) are the computed eigenvalues.