REVIEW OF METHODS FOR COMPUTING EVALS & EVECS

Suppose $A \in \mathbb{R}^{m \times m}$ $\quad A = A^t$

Simple power iteration:

b = random vector
$w_0 = b / \|b\|$
FOR $k = 1, 2, 3, \ldots$
$\quad$ ~~$w_k = A w_k$~~
$\quad b = A w_{k-1}$
$\quad w_k = \frac{1}{\|b\|} b$
END

Then $w_k = \dfrac{A^k b}{\|A^k b\|}$ and $w_k \to$ top evec v_1 at speed $O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$

The Rayleigh quotient $w_k^* A w_k \to \lambda_1$ at speed $O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right)$

Algorithm 27.3. Rayleigh Quotient Iteration

$v^{(0)}$ = some vector with $\|v^{(0)}\| = 1$
$\lambda^{(0)} = (v^{(0)})^T A v^{(0)}$ = corresponding Rayleigh quotient
for $k = 1, 2, \ldots$
$\quad$ Solve $(A - \lambda^{(k-1)} I) w = v^{(k-1)}$ for w $\qquad$ apply $(A - \lambda^{(k-1)} I)^{-1}$
$\quad v^{(k)} = w / \|w\|$ $\qquad$ normalize
$\quad \lambda^{(k)} = (v^{(k)})^T A v^{(k)}$ $\qquad$ Rayleigh quotient

Convergence is <u>extremely</u> fast
Tridiagonalize A first to accelerate iteration
$\Longrightarrow O(m)$ cost per step

BLOCK POWER ITERATION

Suppose: $\begin{cases} A \in \mathbb{R}^{m\times m} & A = A^t \\ |\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \dots \end{cases}$

B_0 = random matrix of size $m \times b$.
FOR $k = 1, 2, 3, \dots$
  $B_k = A B_{k-1}$
END
$[Q, \sim] = qr(B_k)$

Then $col(Q)$ approximates $span\{v_j\}_{j=1}^{b}$

& Evals of $Q^* A Q$ approximate $\lambda_1, \lambda_2, \dots, \lambda_b$.

Numerically unstable!

Stabilized version:

B_0 = random matrix of ~~size~~ size $m \times b$
$[Q_0, \sim] = qr(B_0, 0)$
FOR $k = 1, 2, 3, \dots$
  $B = A Q_{k-1}$
  $[Q_k, \sim] = qr(B, 0)$
END.

If exact arithmetic is used, then

~~the span (Q_k) = sp~~ $col(Q_k) = col(Q) = col(A^k B_0)$

ALGORITHMS FOR FINDING ALL EVALS

Suppose $A \in \mathbb{C}^{m \times m}$ and we seek all evals/evecs of A.

General template:

Step 1 Find Hessenberg form $A = QHQ^*$ with Q unitary & $H = \begin{bmatrix} h_{11} & h_{12} & h_{13} & - \\ h_{21} & h_{22} & h_{23} & - \\ 0 & h_{32} & h_{33} & - \\ 0 & 0 & h_{43} & \\ | & \backslash & , & \end{bmatrix}$

Can be done using Householder reflectors, deterministic, very stable, $O(m^3)$ cost.

Step 2 Use iterative method to drive H to upper triangular form, so that $H = \tilde{Q} T \tilde{Q}^*$ is the Schur factorization.

The result is the Schur fact. $A = (Q\tilde{Q})\, T\, (Q\tilde{Q})^*$

Special case: If $A = A^*$, then H is tridiagonal, and T is diagonal.
Fewer flops, faster convergence.

ITERATIVE METHODS FOR DIAGONALIZING A MATRIX

Suppose $A \in \mathbb{R}^{m \times m}$ & $A = A^*$ for simplicity.

Basic QR iteration:

$A_0 = A$
FOR $k = 1, 2, 3, \ldots$
  $[Q_k, R_k] = qr(A_{k-1})$
  $A_k = R_k Q_k$
END

Observe: $A_1 = R_1 Q_1 \underset{\uparrow}{=} Q_1^* A_0 Q_1$ so A_1 & A_0 are similar.

$A_0 = Q_1 R_1 \Rightarrow R_1 = Q_1^* A_0$

Convergence can be greatly accelerated using "shifts"

Algorithm 28.2. "Practical" QR Algorithm

$(Q^{(0)})^T A^{(0)} Q^{(0)} = A$ — $A^{(0)}$ is a tridiagonalization of A

for $k = 1, 2, \ldots$

  Pick a shift $\mu^{(k)}$ — e.g., choose $\mu^{(k)} = A^{(k-1)}_{mm}$

  $Q^{(k)} R^{(k)} = A^{(k-1)} - \mu^{(k)} I$ — QR factorization of $A^{(k-1)} - \mu^{(k)} I$

  $A^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} I$ — Recombine factors in reverse order

  If any off-diagonal element $A^{(k)}_{j,j+1}$ is sufficiently close to zero,

    set $A_{j,j+1} = A_{j+1,j} = 0$ to obtain

    $\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} = A^{(k)}$

    and now apply the QR algorithm to A_1 and A_2.

Combination of block power iteration and shifted inverse iteration.