$$q_2 = v_2 - \frac{(q_1^T v_2)}{\|q_1\|^2} \cdot q_1$$

$$q_1^T q_2 = q_1^T v_2 - q_1^T \frac{q_1^T v_2}{\cancel{\|q_1\|}^2} \cancel{q_1}$$

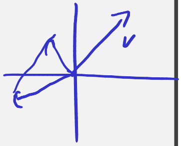
Schur form

Show: Every matrix is orthonormally similar to an upper triangular matrix, i.e. $A = QUQ^T$. This is called the Schur form or Schur factorization.

$Av = \lambda v$

$V = \operatorname{span}(v)$

$Q_1 = \left(v \;\middle|\; \text{Basis of } V^\perp \right)$



$V^\perp = \{ w : v^T w = 0 \}$

$$A = Q_1 \begin{pmatrix} \lambda_1 & ? \\ 0 & \\ \vdots & \\ 0 & \end{pmatrix} Q_1^T$$

Schur form $\longrightarrow$ $= Q_n \begin{pmatrix} & \end{pmatrix} Q^T$ ← unique?

maybe not

Schur Form: Comments, Eigenvalues, Eigenvectors

$A = QUQ^T$. For complex λ:

- Either complex matrices, or
- 2×2 blocks on diag.

If we had a Schur form of A, how can we find the eigenvalues?

on the diagonal

And the eigenvectors?

(orig $U - \lambda I$)

$$U - \lambda I = \begin{pmatrix} U_{11} & \vec{u} & U_{13} \\ & 0 & \vec{v}^T \\ & & U_{33} \end{pmatrix}$$

$$x = \begin{bmatrix} U_{11}^{-1}\vec{u} \\ -1 \\ 0 \end{bmatrix}$$

$(U - \lambda I)x = 0$

$Ux = \lambda x$

$y = Qx \Rightarrow Ay = \lambda y$

Computing Multiple Eigenvalues

All Power Iteration Methods compute one eigenvalue at a time.
What if I want *all* eigenvalues?

1. "Deflation": Similarity transform to

$$\begin{pmatrix} \lambda_1 & * \\ & B \end{pmatrix},$$

Find eigvec of B, keep going

2. Simultaneous Iteration

Simultaneous Iteration

What happens if we carry out power iteration on multiple vectors simultaneously?

$X_0 =$ some random matrix

$X_{k+1} = A X_k$

Problems:

- need to normalize
- X_i gets ill-conditioned, all columns converge to dominant eigenvector

Orthogonal Iteration

$X_0 \in \mathbb{R}^{n \times p}$ $(p \leq n)$

for $k = 0, 1, \ldots$

- $Q_k R_k = X_k$
- $X_{k+1} = A Q_k$

$\oplus$ converges.

$\ominus$ linear conv.

expensive

Toward the QR Algorithm

$$Q_0 R_0 = X_0$$

$$X_1 = A Q_0$$

$$Q_1 R_1 = X_1 = A Q_0 \implies Q_1 R_1 Q_0^T = A$$

Once $Q_{n+1} \approx Q_n$ (Q^*), $Q^* R^* Q^{*T} = A$ Schur fac !

lower of $Q_n^T A Q_n$ is desired criterion

Demo: Orthogonal Iteration [cleared]

QR Iteration/QR Algorithm

Ortho. iter.

$X_0 = A$

$Q_k R_k = X_k$

$X_{k+1} = A Q_k.$

- $\hat{X}_k = Q_k^T A Q_k = \bar{X}_{k+1}$
- $Q_0 = \bar{Q}_0$

$Q_k = \bar{Q}_0 \bar{Q}_1 \cdots \bar{Q}_k$

QR iter.

$\bar{X}_0 = A$

$\bar{Q}_k \bar{R}_k = \bar{X}_k$

$\bar{X}_{k+1} = \bar{R}_k \bar{Q}_k$

$\downarrow$

Schur form

QR Iteration: Incorporating a Shift

How can we accelerate convergence of QR iteration using shifts?

$$\bar X_0 = A$$

$$\bar Q_k \bar R_k = \bar X_k - \sigma_k I$$

$$\bar X_{k+1} = \bar R_k \bar Q_k + \sigma_k I$$

$$\bar X_{k+1} = \bar R_k \bar Q_k + \sigma_k I = [\bar Q_k^T \bar X_k - \cancel{\bar Q_k^T \sigma_k}] \bar Q_k \cancel{+ \sigma_k I}$$

$$\bar X_{k+1} = \bar Q_k \bar X_k^T \bar Q_k$$

Shift:

- Bottom right entry of $\bar X_k$
- Bottom right 2×2, find eigenvalues analytically, pick one

QR Iteration: Computational Expense

A full QR factorization at each iteration costs $O(n^3)$–can we make that cheaper?

Demo: Householder Similarity Transforms [cleared]