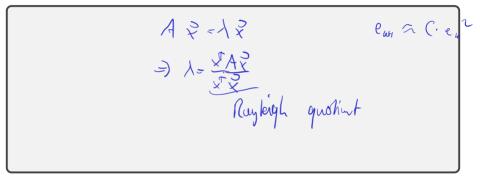


What about Eigenvalues?

Power Iteration generates eigenvectors. What if we would like to know eigenvalues?



Demo: Power Iteration and its Variants [cleared]

Schur form: Motivation

 $\times A \times D$

For finding multiple eigenvalues, want factorization that allows access to all eigenvalues and eigenvectors. Suggestions? $Q \land Q^r = M \qquad \text{diagond}$

Schur form: Motivation

For finding multiple eigenvalues, want factorization that allows access to all eigenvalues and eigenvectors.

Suggestions?

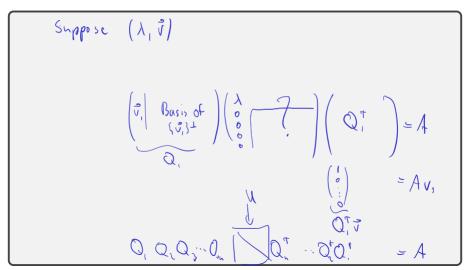
Diagonalization A = XDX⁻¹ cannot provide what we need: it does not always exist.

Even if it did exist, computing/applying X⁻¹ would be subject to rounding concerns.

Idea: use a similarity transform with orthogonal matrices.

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.



Schur Form: Comments, Eigenvalues, Eigenvectors

- $A = QUQ^T$. For complex λ :
 - Either complex matrices, or
 - \blacktriangleright 2 × 2 blocks on diag.

If we had a Schur form of A (no 2×2 blocks), can we find the eigenvalues?

 $QAQ^{\dagger} = U$

And the eigenvectors?

Find elynes of U, use similarly.
$$\tilde{X} = (U_{ii}^{T} \tilde{u}_{ij}^{T} - I_{ij}^{T} U_{ij}^{T})^{T}$$

 $W - \lambda T = \begin{pmatrix} U_{ii} \tilde{u} & U_{ij} \\ 0 & \tilde{v}^{T} \\ 0 & \tilde{v}^{T} \end{pmatrix} \stackrel{\mathfrak{I}}{\Rightarrow} (U - \lambda T) \stackrel{\mathfrak{I}}{\approx} = 0$
 $(U_{\gamma_{ij}}) \stackrel{\mathfrak{I}}{\Rightarrow} (U_{\chi} = \lambda T)$

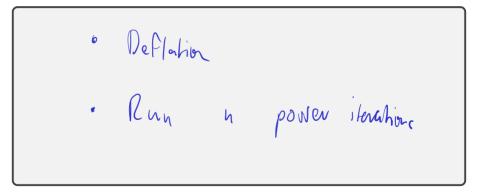
$$\vec{y} = Q^T \vec{x}$$
 $Q AQT = U$

 $A \vec{y} = Q^T U Q Q^T \vec{x} = Q^T U Q$

 $A \vec{y} = Q^T U Q Q^T \vec{x} = Q^T U \vec{x} = Q^T U$

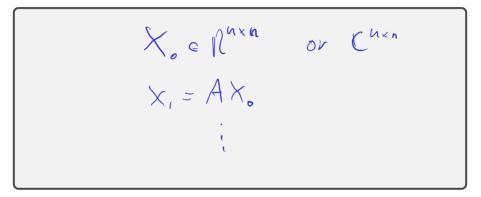
Computing Multiple Eigenvalues

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

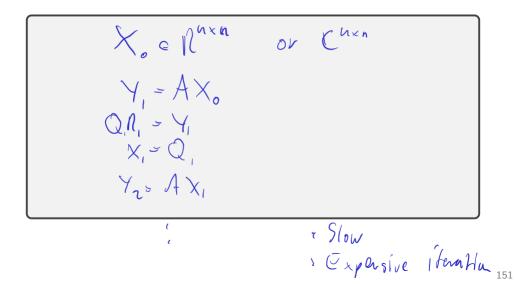


Simultaneous Iteration

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



Orthogonal Iteration



Toward the QR Algorithm

$$Q_{0} R_{6} = X_{0}$$

$$X_{1} = A Q_{0}$$

$$Q_{1} R_{1} = X_{1} = A Q_{0} \implies A = Q_{1} R_{1} Q_{0}^{T}$$

$$X_{1} = A Q_{1}$$

$$Q_{1} R_{1} = X_{1}$$

$$Q_{1} R_{1} = Q_{1}^{T} A Q_{1} \implies A$$

$$Q_{1} R_{1} = X_{1}$$

$$Q_{1} R_{1} = Q_{1}^{T} A Q_{1} \implies A$$

Demo: Orthogonal Iteration [cleared]

QR Iteration/QR Algorithm

QR alg.

$$\begin{split} \vec{X}_{o} &= A \\ \vec{Q}_{k} \vec{R}_{k} = \vec{X}_{n} \\ \vec{X}_{k+1} &= \vec{R}_{k} \vec{Q}_{k} \\ \vec{X}_{k+1} &= \vec{Q}_{k} \vec{R}_{k} \\ \vec{A}^{2} &= \vec{Q}_{k} \vec{R}_{k} \\ \vec{Q}_{k+1} \\ \vec{Q}_$$

Proof sketch: Equivalence of QR iteration/Orth. iteration

Orthogonal Iteration (no bars)

 $\blacktriangleright X_0 := A$ \triangleright $Q_0 R_0 := X_0$. where we may choose $Q_0 = \overline{Q}_0$ $\hat{X}_0 = Q_0^H A Q_0 =$ $Q_0^H Q_0 R_0 Q_0 = R_0 Q_0$ $\blacktriangleright X_1 := AQ_0$ $\triangleright Q_1 R_1 := X_1$ and because of $X_1 = Q_0 Q_0^H A Q_0 = Q_0 \bar{X}_1 =$ $Q_0 \bar{Q}_1 \bar{R}_1$ we may choose $Q_1 = Q_0 \bar{Q}_1 = \bar{Q}_0 \bar{Q}_1$

QR Iteration (with bars)

$$\bar{X}_0 := A$$
$$\bar{Q}_0 \bar{R}_0 := A$$

$$\bar{X}_2 := \bar{R}_1 \bar{Q}_1 \bar{X}_2 = Q_1^H A Q_1 = \hat{X}_1$$

Demo: QR Iteration [cleared]

QR Iteration: Forward and Inverse

QR iteration may be viewed as performing inverse iteration. How?