Introduction to eigenvalue problems.

Note Title

- Recall some facts -

(1) $X \in \mathbb{C}^{m\times m}$ be nonsingular, then $A \mapsto X^{-1}AX$ is called a "similarity transformation".

- Two matrices A & B are said to be similar if there exists an invertible matrix X such that $B = X^{-1}AX$.
- Similarity is an equivalence relation on the set of $m\times m$ matrices.
- If A & B are similar matrices, they represent the same linear transformation albeit w.r.t. different bases. So A and B will have the same characteristic polynomial, eigenvalues & algebraic, geometric multiplicities, since these entities are independent of choice of basis.

  However, they may not have the same e-vectors, since e-vector expression is basis dependent.

(2) alg. multi. $\geq$ geom. multi. in general.

- If λ is an e-value for which alg. multi. > geom. multi. then λ is said to be a "defective e-value."
- A matrix that has one or more defective e-values is called a "defective matrix".
- A is non-defective $\Longleftrightarrow$ A has an e-value decomposition $X\Lambda X^{-1}$.

  ($X\Lambda X^{-1}$ is called an "e-value revealing factorization").

(3) Some other eigenvalue-revealing factorizations:

(i) Unitary diagonalization - If the e-vectors of A form an orthogonal basis of the underlying vector space, then the unitary matrix Q, whose columns are these e-vectors of A, diagonalizes A i.e.

$$A = Q \Lambda Q^*.$$

Then A is said to be unitarily diagonalizable.

Spectral theorem: A matrix is unitarily diagonalizable $\Leftrightarrow$ it is normal.

(ii) Schur factorization - $A = Q T Q^*$, where Q is unitary & T is upper Δ^r.

Note that e-values of A appear on the diagonal of T since A & T are similar.

Theorem: Every square matrix $A \in \mathbb{C}^{m \times m}$ has a Schur factorization.
(Schur canonical form).

(Note that this is the theorem - "every square matrix over $\mathbb{C}$ is triangulable" from Lecture 1.)

Theorem (Real Schur canonical form) If $A \in \mathbb{R}^{m \times m}$,

$\exists$ a real orthogonal matrix V such that $A = V T V^T$, where T is quasi-upper Δ^r. This means T is block upper Δ^r with 1×1 & 2×2 blocks on the diagonal. E-values of T (and hence of A) are e-values of its diagonal blocks.

$$\begin{bmatrix} [\cdot\cdot] & & \\ & [\cdot] & * \\ 0 & & [\;] \end{bmatrix}$$

(Refer to Theorem 4.3, page 147, of Demmel's book for proof & details)

Q: Why can we not have direct methods for e-value computations (i.e. why do we have only iterative methods?)

The following problems are equivalent -

(i) finding e-values of a $m \times m$ matrix.

(ii) finding roots of a (monic) polynomial of degree m.

$$[A]_{m\times m} \longmapsto \text{roots of char poly.}$$

$$\text{Companion matrix of } p(x) = \begin{bmatrix} 0 & & & & -a_0 \\ 1 & \ddots & & & -a_1 \\ & \ddots & \ddots & & \vdots \\ & & \ddots & 0 & \vdots \\ & & & 1 & -a_{m-1} \end{bmatrix} \longleftarrow p(x) = x^m + a_{m-1}x^{m-1} + \dots + a_1 x + a_0$$

Abel's impossibility theorem: It is known, from a deep result in mathematics - known as Abel's theorem - that there does not exist a formula for finding roots of an arbitrary polynomial (degree ≥ 5), given its coefficients.

This means there is no uniform quadratic-discriminant-like formula to find roots of a polynomial of degree ≥ 5.

Hence there cannot be an algorithm that produces the exact roots of an arbitrary polynomial (and hence the e-values of an arbitrary square matrix) in a fixed number of steps.

Thus, any e-value finding algorithm must be iterative.

Note: The algorithms for e-value problems are split into 2 groups - direct and iterative.

Direct methods also use iteration - they are termed 'direct' only because they are considered more reliable and almost always converge.

Iterative methods are usually applied to sparse matrices or matrices for which matrix-vector multiplication is the only convenient operation to perform. The convergence properties of these methods depend strongly on the matrix entries.

We will study the following algorithms -

(I) Direct methods

- General use algorithms (in particular, for non-symmetric matrices)
  - Power iteration
  - Inverse iteration
  - Orthogonal iteration
  - QR iteration
- Two phase method & reduction to Hessenberg form.
- Algorithms for symmetric matrices -
  - Rayleigh quotient iteration
  - Jacobi's method.
  - Bisection method
  - Divide-and-conquer.

(II) Iterative methods -

- Arnoldi
- GMRES
- Lanczos.

(Topics left:
1. Computing the SVD (Golub-Kahan bidiagonalization)
2. Perturbation theory for e-value problem.
3. Basic ideas in Krylov subspace methods
4. Preconditioning.
)