Summary of the course 2014h ed.

Anton Evgrafov

November 19, 2014

Linear algebra background:

- Gauss elimination, LU/Cholesky/LDL-factorization;
- Rank/null space/range/adjoint;
- Eigenvalues, -vectors, -spaces; algebraic and geometric multiplicities;
- Induced/Frobenius matrix norm;
- Symmetric/Hermitian/normal/unitary matrices;
- Similarity (unitary or not), diagonalization, Jordan normal form, Schur factorization;
- Solution perturbation $\sim \kappa(A)$

QR-factorization

$$A = QR, \qquad Q^*Q = I, \qquad R_{ij} = 0, i \geq j$$

Used for:

- Solving least-squares problems (e.g. in GMRES/DQGMRES/MINRES/SYMMLQ)
- QR-algorithm for eigenvalues

Computational algorithms:

- (modified) Gramm-Schmidt orthogonalization
- Hausholder reflections
- Givens rotations

Diagonalization methods

Rare instances, known:

1. Eigenvalues/vectors $A = Q\Lambda Q^T$
2. Fast algorithm (e.g. FFT) $v \leftrightarrow Qv$

$$Ax = b \implies x = Q[\Lambda^{-1}(Q^T b)]$$

Matrix-splitting methods

$$
\begin{aligned}
A &= M - N \\
x_{k+1} &= M^{-1}(Nx_k + b) \\
e_{k+1} &= M^{-1}Ne_k
\end{aligned}
$$

Theorem

$\|e_k\| \to 0 \ \forall x_0 \iff \rho(M^{-1}N) < 1.$

Asymptotically $\|e_{k+1}\|/\|e_k\| \sim \rho(M^{-1}N)$.

Slow!

Matrix-splitting methods

Standard examples:

- Jacobi: $M = \mathrm{diag} A$
- Gauss-Seidel: $M = \mathrm{triu} A$
- Block variations

Use:

- Smoothers in multigrid
- Preconditioners: additive Schwarz $\approx$ block-Jacobi; multiplicative Schwarz $\approx$ block-GS

Projection operators

$$P^2 = P$$

- $I - P$ also projector
- Null space/range: $\mathrm{ker} P \oplus \mathrm{range} P = \mathbb{C}^n$
- $y = Px \iff$
  $y \in \mathrm{range} P \;\&\; x - y = (I - P)x \in \mathrm{ker} P \iff$
  $y \in \mathrm{range} P \;\&\; x - y = (I - P)x \perp [\mathrm{ker} P]^\perp$
- Orthogonal projectors: $P^T = P$

Used in this course: analysis of projection methods

Projection/Petrov-Galerkin framework

$$\dim \mathcal{K} = \dim \mathcal{L} << \dim \mathbb{C}^n$$

Find: $x \in x_0 + \mathcal{K}$ such that $r = b - Ax \perp \mathcal{L}$

Projection/Petrov-Galerkin framework

$$\dim \mathcal{K} = \dim \mathcal{L} << \dim \mathbb{C}^n$$

Find: $x \in x_0 + \mathcal{K}$ such that $r = b - Ax \perp \mathcal{L}$

- Error projection: A SPD, $\mathcal{L} = \mathcal{K}$

Theorem

$$x \in \arg\min_{y \in \mathcal{K}} \|y - A^{-1}b\|_A$$

Projection/Petrov-Galerkin framework

$$\dim \mathcal{K} = \dim \mathcal{L} << \dim \mathbb{C}^n$$

Find: $x \in x_0 + \mathcal{K}$ such that $r = b - Ax \perp \mathcal{L}$

- Error projection: A SPD, $\mathcal{L} = \mathcal{K}$

Theorem

$$x \in \arg\min_{y \in \mathcal{K}} \|y - A^{-1}b\|_A$$

- Residual projection: $\mathcal{L} = A\mathcal{K}$

Theorem

$$x \in \arg\min_{y \in \mathcal{K}} \|b - Ay\|_2$$

1D variations:

- $\mathcal{L} = \mathcal{K} = e_k \iff$ Gauss-Seidel
- $\mathcal{L} = \mathcal{K} = r = b - Ax \iff$ steepest descent
- $\ldots$

Krylov subspace methods:

$$\mathcal{K} = \mathcal{K}_m(A, r_0) = \mathrm{span}\langle r_0, Ar_o, \dots, A^{m-1} r_0 \rangle$$

- $\mathcal{L} = \mathcal{K}$: FOM, CG, SYMMLQ
- $\mathcal{L} = A\mathcal{K}$: GMRES, MINRES

Arnoldi:

Generating ON basis for $\mathcal{K}_m(A, r_0)$!
$V_m^T V_m = I$, $\mathrm{range} V_m = \mathcal{K}_m(A, r_0)$
$AV_m = V_{m+1}\bar{H}_m$, $[\bar{H}_m]_{ij} = 0$ if $i > j + 1$.
Algorithms: (modified) Gramm-Schmidt, Hausholder
Arnoldi used in: FOM, GMRES

Cannot store Arnoldi vectors:

- Restarts: FOM(k), GMRES(k)
- Incomplete (partial) orthogonalization: IOM/DIOM, QGMRES/DQGMRES

Lanczos:

$$A = A^T \implies H_m = H_m^T \implies \text{tri-diagonal!}$$

In Gramm-Schmidt, only need to orthogonolaze to two previous Arnoldi vectors (3-term recursion) $\implies$ computational savings $O(m^2 n) \to O(mn)$.

Lanczos used in: D-LANCZOS ($\approx$ CG), MINRES ($\approx$ DQGMRES on symmetric A), SYMMLQ, Lanczos method for eigenvalues (+Rayleigh-Ritz)

"Typical" convergence theorem:

Convergence, GMRES, diagonalizable case: $A = X \Lambda X^{-1}$

Theorem

$$\|r_k\|_2 \leq \kappa(X) \min_p \max_i |p(\lambda_i)|$$

p-polynomial degree k, $p(0) = 1$.

Preconditioning:

- Left: $M^{-1}Ax = M^{-1}b$
- Right: $AM^{-1}u = b$, $x = M^{-1}u$.

Goal: $M^{-1}A$ is easier for Krylov methods than A (e.g. has few clustered eigenvalues)

$M^{-1} \approx A^{-1}$ but "simple" to apply (every Krylov iteration)

ILU, SAI, multigrid, domain decomposition (DD)...

2-grid algorithm:

1. (pre-)smooth error/residual: underrelaxed Jacobi, GS, ...
2. restrict residual to coarse grid: $r^H = I_h^H r^h$
3. appx. solve $A^H e^H = -r^H$
4. interpolate onto fine grid: $e^h = I_H^h e^H$
5. apply correction: $x^h = x^h - e^h$
6. (post-)smooth error/residual after interpolations
7. repeat

Multigrid: recursion at step 3.

Domain decomposition:

- PDE-level: overlapping (Schwarz) vs non-overlapping (Schur). Then discretize and solve! Use special block-structure of the discretized system when solving.
- Discrete level: additive/multiplicative Schwarz

SVD decomposition and applications

$$A = U\Sigma V^*$$

- U, V - unitary
- Σ - diagonal, positive

$$\sigma = [\lambda(AA^*)]^{1/2} = [\lambda(A^*A)]^{1/2}$$

Uses:

- Least-squares
- Low rank approximation of A
- Range space
- Matrix norms

Eigenvalue algorithms for small problems:

Idea: eigenvalue revealing similarity transformation (Schur factorization): $A = QRQ^T$.

Phase I: Hessenberg (non-symmetric)/tridiagonalization (symmetric)

$$A = QHQ^T, \qquad Q^TQ = I, H_{ij} = 0, i > j + 1$$

Phase I algorithms: Hausholder transformations

Eigenvalue algorithms for small problems:

Phase II algorithms:

- Rayleigh quotient: $\lambda \approx r(v) = [v^T A v]/[v^T v]$
- Power iteration: $v_k = A^k v_0/\|\cdot\|$; eigenvector corr. largest in magnitude eigenvalue; slow convergence
- Inverse iteration: $v_k = A^{-k} v_0/\|\cdot\|$; smallest in magn eigenvalue
- Shifted inverse iteration: $v_k = (A - \mu I)^{-k} v_0/\|\cdot\|$ eigenvalue closest to μ
- Rayleigh iteration: $v_k = (A - \lambda_k I)^{-k} v_0/\|\cdot\|$, $\lambda_k = r(v_k)$. *Cubically convergent* for symmetric matrices!

Eigenvalue algorithms for small problems:

Phase II algorithms:

- Simultaneous versions of power iterations; and their relation to
- QR-algorithm with shifts: $Q_k R_k = A_k - \mu_k I$;
  $A_{k+1} = R_k Q_k + \mu_k I$.
- Power/inverse power iteration for first/last column of Q_k
- Wilkinson shift

Perturbation analysis for symmetric eigenvalue problems

- Eigenvalues (symm matrices):

$$|\lambda(A) - \lambda(A+E)| \leq \|E\|_2$$

$$|\alpha - \lambda(A)| \leq \|Av - \alpha v\|_2, \qquad \forall \|v\|_2 = 1$$

- Eigenvectors (symm matrices):

$$\frac{1}{2}\sin(2\,\text{angle}) \sim \frac{\|\text{perturbation}\|_2}{\text{eigenvalue gap}}$$

Eigenvalue algorithms for large problems:

- Lanczos for tridiagonalization: $T_k = Q_k^T A Q_k$
- Rayleigh-Ritz idea: $\lambda(A) \approx \lambda(T_k)$; error estimates available
- Use Phase-II algorithms on T_k.

Most important omissions:

- Bi-orthogonalization Krylov methods ($\approx$ non-symmetric Lanczos): bi-CG, bi-CG-stab, QMR, TFQMR (transpose-free QMR). Most need $v \mapsto A^T v$ in addition to $v \mapsto Av$
- Multigrid cycles: V, W, F, full-multigrid
- Algebraic multigrid
- Other eigenvalue algorithms for “small” problems: Jacobi (most accurate); divide & conquer (fastest for symmetric matrices);
- Other eigenvalue algorithms for “large” non-symmetric problems: $\approx$ Rayleigh-Ritz idea + Arnoldi or biorthogonalization
- Algorithms for SVD