Ma221 Lecture 14

Krylov Subspace Methods (KSMs)
for $Ax=b$ and $Ax=dx$

Introduction: Arnoldi and Lanczos

Many KSMs for $Ax=b$

For general $A$: use GMRES
Generalized Minimum Residual
For spd (eg Poisson): use CG
Conjugate Gradients
Later: decision tree to choose based on $A$ (Fig 6.8 text)

Unlike Splitting Methods, KSMs only need a "black-box" to multiply $Ax$
for any $x$. Don't need diagonal of $A$
etc that Jacobi needs

Advantages of only needing "black-box" for $Ax$ (sometimes $A^x$)

1) can write algorithms that are very general, access $A$ only
via subroutine for Ax

2) can solve problems where A not available explicitly, eg arises from doing a complicated simulation, eg from differentiating a program for $y = f(x)$ to get $J_f$, for solving optimization problems (systems exist for automatically differentiating a program for $y = f(x)$)

If we do have an explicit $A(x)$, possible to reorganize KSMs to avoid communication: Dominant cost of KSM often $A\cdot x$. If $A$ large, $k$ iterations of $A\cdot x$ costs $O(k)$ in communication to read $A$ $k$ times. Possible to reorganize KSMs to take $k$ steps while reading $A$ just once. See links on class web page.
How to extract information about A from doing A * x:
Given starting vector y, (say y_1 = b if solving A * x = b), compute y_2 = A * y_1,
... y_n = A * y_{n-1}, y_n = A^{n-1} y_1
K = [y_1, y_2, ..., y_n]
A K = [A y_1, ..., A y_n]
= [y_2, y_3, ..., A^{n-1} y_1]
If K nonsingular, c = - K^{-1} A^{n-1} y_1,
A K = K [c_2, c_3, c_4, ..., c_n - c] = K * C
C = K^{-1} A K = \begin{bmatrix}
0 & 0 & \cdots & c_1 \\
1 & 0 & \cdots & -c_2 \\
0 & 1 & \cdots & -c_3 \\
\vdots & \ddots & \ddots & \ddots \\
0 & \cdots & 0 & -c_n
\end{bmatrix}

upper Hessenberg, companion matrix
i.e. its characteristic polynomial is
p(x) = x^n + \sum_{i=1}^n c_i x^{i-1}
could try to use this decomposition to solve $Ax=b$ or $Ax=Ax$

Disadvantages:

1. $K$ likely to be dense even if $A$ sparse, so solving $Kx=b$ harder than with $A$

2. $K$ likely to be very ill-conditioned, because repeated multiplication by $A \Rightarrow$ power method $\Rightarrow y_i$ converging to $e$ for largest eval

Krylov subspace methods address these disadvantages:

Instead of computing $K$, compute orthogonal $Q$ s.t. leading $k$ columns of $K$ and $Q$ span same subspace:

Def: Krylov subspace

$$\text{span}\{y, Ay, \ldots, A^{k-1}y\}$$

$$= \text{span}\{y, Ay, \ldots, A^my\}$$

$$= \mathcal{K}_k(A, y)$$
Relationship between $K$ and $Q$: $K = QR$

Furthermore, only compute $k < n$ columns of $Q$, compute "best" solution of $Ax = b$ or $Ax = dx$ for $x \in R^k(A, y_i)$. Many definitions of "best" $\Rightarrow$ many algorithms

Ma 221 Lecture 14 Segment 2

$A, y_1, y_{i+1} = Ay_i, K = [y_1, \ldots, y_n]$

$K^T A K = C = \begin{bmatrix} 1 & \vdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} = \text{upper Hessenberg}$

$K = QR \Downarrow C = R^T QAQR$

$\Rightarrow \quad (*) \quad RCR^{-1} = Q^T A Q = H$

$R$ and $R^{-1}$ $\triangledown$, and $C \triangledown \Rightarrow H$ also $\text{upper Hessenberg}$ ($\text{CQ Method in the text}$)

$A$ symmetric $\Rightarrow Q^T AQ = H$ symmetric

$\Rightarrow H$ tri-diagonal
Goal: compute columns of $Q$ and $H$ one at a time, $Q = [q_1, \ldots, q_n]$

$(a) \Rightarrow A Q = Q H$

$j$th column of both sides $\Rightarrow$

\[ A q_j = \sum_{i=1}^{j+1} g_i H_{ij} \]

$q_j$ orthonormal $\Rightarrow$ multiply both sides by $g_m$

\[ g_m^T A q_j = \sum_{i=1}^{j+1} g_m^T g_i H_{ij} = H_{mj} \quad 1 \leq m \leq j \]

\[ H_{j+1, j} g_{j+1} = A q_j - \sum_{i=1}^{j} g_i H_{ij} \]

Arnoldi algorithm for (partial) reduction to Hessenberg form:

\[ q_1 = y_1 / \| y_1 \|_2 \]

for $j = 1$ to $k$

\[ z = A q_j \]

for $i = 1$ to $j$ \quad run Modified Gram-Schmidt (MGS) on $z$

\[ H_{ij} = g_i^T z \]

\[ z = z - H_{ij} g_i \]

end for

\[ H_{j+1, j} = \| z \|_2 \]

\[ q_{j+1} = z / H_{j+1, j} \]
end for

Vectors $q_j$ called Arnoldi vectors
cost = $k$ multiplications by $A$
$+ O(k^2 n)$ flops for A GS

If we stop after $k$ steps, what have we learned about $A$?

\[ Q = [Q_k, Q_u] \]

\[ Q_k = [q_1, \ldots, q_k] \text{ known} \]

\[ Q_u \text{ known} \]

\[ H = Q^T AQ = [Q_k, Q_u]^T A [Q_k, Q_u] \]

\[ = [Q_k^T A Q_k, Q_k^T A Q_u, Q_u^T A Q_k, Q_u^T A Q_u] = [H_k, H_{ku}; H_{uk}, H_u] \]

$H$ upper Hessenberg; so are $H_k$ and $H_u$

\[ H_{ku} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \Rightarrow H_k, H_{ku} \text{ known} \]

$H_{uk}, H_u$ unknown

If $A = A^T$: $H = T = \begin{bmatrix} \alpha_1 & \beta_1 & \beta_2 & \beta_3 & \cdots & \alpha_n \\ \beta_1 & \beta_2 & \beta_3 & \cdots & \alpha_n \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}$
Equate column $j$ of both sides of $AQ = QT$

$$A q_j = B_{j-1} q_{j-1} + \alpha_j q_j + \beta_j q_{j+1}$$

Multiply both sides by $q_j^T$, getting

$$q_j^T A q_j = \alpha_j$$

Lanczos Algorithm for (partial) reduction to tridiagonal form of $A = A^T$

$$q_1 = \frac{q_1}{\| q_1 \|_2}, B_0 = 0, q_0 = 0$$

for $j = 1$ to $k$

$$z = A \cdot q_j$$

$$\alpha_j = \frac{z^T z}{\| z \|_2}$$

$$z = z - \alpha_j q_j - \beta_{j-1} q_{j-1}$$

$$\beta_j = \| z \|_2$$

$$q_{j+1} = z / \beta_j$$

end for

Notation

$\text{span} (q_1, \ldots, q_k)$ is a Krylov subspace

$\mathcal{K}_k (A, y_j) = \text{span} (g_1, \ldots, g_k)$

or $\mathcal{K}_k$ for short

$H_k$ or $(T_k$ for Lanczos)$)$ projection of $A$ onto $\mathcal{K}_k$
Our goal is to solve $Ax=b$ or $Ax=\lambda x$ by finding "best" solution in $K_k$.

Consider $Ax=\lambda x$: use evals of $H_k$ (or $T_k$) as approximate evals of $A$:

To estimate error:

$H_k y = \lambda y$

$H \begin{bmatrix} y \\ 0 \end{bmatrix} = \begin{bmatrix} H_k & H_x \end{bmatrix} \begin{bmatrix} y \\ 0 \end{bmatrix} = \begin{bmatrix} H_k y \\ H_x y \end{bmatrix} = \begin{bmatrix} \lambda y \\ H_x y \end{bmatrix}$

$A Q = Q H$

$A (Q \begin{bmatrix} y \\ 0 \end{bmatrix}) = \lambda (Q \begin{bmatrix} y \\ 0 \end{bmatrix}) + q_{k+1} H_{k+1} y_k$

approx evect of $A$

So if $H_{k+1} y_k$ small $\Rightarrow$ evect/eval pair $(y', d) = (Q \begin{bmatrix} y \\ 0 \end{bmatrix}, d)$ has a small residual

$\| A y' - dy' \|_2 = | H_{k+1} y_k |$

$A = A^T \Rightarrow$ we know from Chap 5, Thm 5.5, $| H_{k+1} y_k |$ bounds
error in $\lambda$

Numerical experiments: (Fig 7.2 in text) plot evals of $T_k$ vs $k$ to see how they converge