Generalized MINRES and LSQR

Orthogonal tridiagonalization of general matrices

Michael Saunders

Systems Optimization Laboratory

Dept of Management Science and Engineering
Stanford University

CME 510 Linear Algebra and Optimization Seminar
Stanford University, October 3, 2007

Outline

1. History

Outline

1. History
2. Tridiagonalization of symmetric A

Outline

1. History
2. Tridiagonalization of symmetric A
3. Bidiagonalization of rectangular A

Outline

1. History
2. Tridiagonalization of symmetric A
3. Bidiagonalization of rectangular A
4. Tridiagonalization of unsymmetric A

Outline

1. History
2. Tridiagonalization of symmetric A
3. Bidiagonalization of rectangular A
4. Tridiagonalization of unsymmetric A
5. Original motivation

Outline

1. History
2. Tridiagonalization of symmetric A
3. Bidiagonalization of rectangular A
4. Tridiagonalization of unsymmetric A
5. Original motivation
6. Symmetric $Ax = b$

Outline

1. History
2. Tridiagonalization of symmetric A
3. Bidiagonalization of rectangular A
4. Tridiagonalization of unsymmetric A
5. Original motivation
6. Symmetric $Ax = b$
7. Unsymmetric $Ax = b$

Outline

1. History
2. Tridiagonalization of symmetric A
3. Bidiagonalization of rectangular A
4. Tridiagonalization of unsymmetric A
5. Original motivation
6. Symmetric $Ax = b$
7. Unsymmetric $Ax = b$
8. Elizabeth Yip's aim

Outline

Outline

History of iterative solvers

History

- 1950 Lanczos tridiagonalization of symmetric A

  for eigenvalues. Products Av plus a few vectors

History

- 1950 Lanczos tridiagonalization of symmetric A

  for eigenvalues. Products Av plus a few vectors

- 1952 Lanczos method of "minimized iterations"

  for posdef $Ax = b$

History

- 1950 Lanczos tridiagonalization of symmetric A

  for eigenvalues. Products Av plus a few vectors

- 1952 Lanczos method of "minimized iterations"

  for posdef $Ax = b$

- 1952 Hestenes and Stiefel CG method

  for posdef $Ax = b$

History

- 1950 Lanczos tridiagonalization of symmetric A

  for eigenvalues. Products Av plus a few vectors

- 1952 Lanczos method of "minimized iterations"

  for posdef $Ax = b$

- 1952 Hestenes and Stiefel CG method

  for posdef $Ax = b$

- 1965 Golub-Kahan bidiagonalization of general A

  for SVD

History

- 1950 Lanczos tridiagonalization of symmetric A
  for eigenvalues. Products Av plus a few vectors
- 1952 Lanczos method of "minimized iterations"
  for posdef $Ax = b$
- 1952 Hestenes and Stiefel CG method
  for posdef $Ax = b$
- 1965 Golub-Kahan bidiagonalization of general A
  for SVD
- 1971 Paige thesis on Lanczos tridiagonalization
  for eigenvalues

History

- 1950 Lanczos tridiagonalization of symmetric A
  for eigenvalues. Products Av plus a few vectors
- 1952 Lanczos method of "minimized iterations"
  for posdef $Ax = b$
- 1952 Hestenes and Stiefel CG method
  for posdef $Ax = b$
- 1965 Golub-Kahan bidiagonalization of general A
  for SVD
- 1971 Paige thesis on Lanczos tridiagonalization
  for eigenvalues
- 1975 Paige-Saunders SYMMLQ and MINRES
  Lanczos tridiagonalization for indefinite $Ax = b$

History

- 1950 Lanczos tridiagonalization of symmetric A
  for eigenvalues. Products Av plus a few vectors
- 1952 Lanczos method of "minimized iterations"
  for posdef $Ax = b$
- 1952 Hestenes and Stiefel CG method
  for posdef $Ax = b$
- 1965 Golub-Kahan bidiagonalization of general A
  for SVD
- 1971 Paige thesis on Lanczos tridiagonalization
  for eigenvalues
- 1975 Paige-Saunders SYMMLQ and MINRES
  Lanczos tridiagonalization for indefinite $Ax = b$
- 1982 Paige-Saunders LSQR
  Golub-Kahan bidiagonalization for general $Ax = b$, $\min \|Ax - b\|$

History (contd)

- 1981 Saunders, 2 months in Sweden
  Tridiagonalization for unsymmetric $Ax = b$
  Coded and tested USYMLQ

History (contd)

- 1981 Saunders, 2 months in Sweden

  Tridiagonalization for unsymmetric $Ax = b$

  Coded and tested USYMLQ

- 1982 (July) Yip, SIAM meeting at Stanford

  "CG method for unsymmetric matrices applied to PDE problems"

History (contd)

- 1981 Saunders, 2 months in Sweden

  Tridiagonalization for unsymmetric $Ax = b$

  Coded and tested USYMLQ

- 1982 (July) Yip, SIAM meeting at Stanford

  "CG method for unsymmetric matrices applied to PDE problems"

- 1982 (Oct) Simon, Sparse Matrix Symposium

  "The Lanczos algorithm for . . . nonsymmetric linear systems"

  (?? Seems to be LSQR with partial reorthogonalization)

History (contd)

- 1981 Saunders, 2 months in Sweden
  Tridiagonalization for unsymmetric $Ax = b$
  Coded and tested USYMLQ
- 1982 (July) Yip, SIAM meeting at Stanford
  "CG method for unsymmetric matrices applied to PDE problems"
- 1982 (Oct) Simon, Sparse Matrix Symposium
  "The Lanczos algorithm for ... nonsymmetric linear systems"
  (?? Seems to be LSQR with partial reorthogonalization)
- 1988 Saunders, Simon, and Yip, SINUM 25
  "Two CG-type methods for unsymmetric linear equations"
  (USYMLQ and USYMQR $\equiv$ GMINRES)

History (contd)

- 2006 Reichel and Ye

  "A generalized LSQR algorithm" (GLSQR)

  Unsymmetric tridiagonalization, focused on rectangular A

History (contd)

- 2006 Reichel and Ye
  "A generalized LSQR algorithm" (GLSQR)
  Unsymmetric tridiagonalization, focused on rectangular A
- 2007 Golub, Stoll, and Wathen (draft)
  "Approximation of outputs"
  Unsymmetric tridiagonalization, focused on $Ax = b$, $A^T y = c$
  and estimation of $c^T x$ and $b^T y$

Tridiagonalization of symmetric A using orthogonal matrices

Symmetric A

- Tridiagonalization for dense EVD (eigenvalues)

$$V_1^T A = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ & * & * & * \\ & * & * & * \end{pmatrix}, \quad V_1^T A V_1 = \begin{pmatrix} * & * & & \\ * & * & * & * \\ & * & * & * \\ & * & * & * \end{pmatrix} \quad \cdots \rightarrow \begin{pmatrix} * & * & & \\ * & * & * & \\ & * & * & * \\ & & * & * \end{pmatrix}$$

$$V^T A V = T \quad \Rightarrow \quad AV = VT$$

Symmetric A

- Tridiagonalization for dense EVD (eigenvalues)

$$V_1^T A = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ & * & * & * \\ & * & * & * \end{pmatrix}, \quad V_1^T A V_1 = \begin{pmatrix} * & * & & \\ * & * & * & * \\ & * & * & * \\ & * & * & * \end{pmatrix} \quad \cdots \rightarrow \begin{pmatrix} * & * & & \\ * & * & * & \\ & * & * & * \\ & & * & * \end{pmatrix}$$

$$V^T A V = T \quad \Rightarrow \quad AV = VT$$

- Symmetric Lanczos process on A, b

$$\beta_1 v_1 = b$$

$$p_1 = Av_1 \qquad \alpha_1 = v_1^T p_1$$
$$\beta_2 v_2 = p_1 - \alpha_1 v_1$$

$$p_2 = Av_2 \qquad \alpha_2 = v_2^T p_2$$
$$\beta_3 v_3 = p_2 - \alpha_2 v_2 - \beta_1 v_1$$

$$T_k = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \beta_3 & \\ & * & * & * \\ & & \beta_k & \alpha_k \end{pmatrix}$$

$$V_k = \begin{pmatrix} v_1 & v_2 & \ldots & v_k \end{pmatrix}$$

$$AV_k = V_k T_k + \beta_{k+1} v_{k+1} e_k^T$$

Bidiagonalization of rectangular A

Rectangular A

- Bidiagonalization for dense SVD (Golub and Kahan 1965)

$$U_1^T A = \begin{pmatrix} * & * & * & * \\ & * & * & * \\ & * & * & * \\ & * & * & * \\ & * & * & * \end{pmatrix}, \quad U_1^T A V_1 = \begin{pmatrix} * & * & & \\ & * & * & * \\ & * & * & * \\ & * & * & * \\ & * & * & * \end{pmatrix} \quad \cdots \longrightarrow \begin{pmatrix} * & * & & \\ & * & * & \\ & & * & * \\ & & & * \\ & & & \end{pmatrix}$$

$$U^T A V = B \quad \Rightarrow \quad AV = UB, \quad A^T U = V B^T$$

Rectangular A

- Bidiagonalization for dense SVD (Golub and Kahan 1965)

$$U_1^T A = \begin{pmatrix} * & * & * & * \\ & * & * & * \\ & * & * & * \\ & * & * & * \\ & * & * & * \end{pmatrix}, \quad U_1^T A V_1 = \begin{pmatrix} * & * & & \\ & * & * & * \\ & * & * & * \\ & * & * & * \\ & * & * & * \end{pmatrix} \quad \cdots \rightarrow \begin{pmatrix} * & * & & \\ & * & * & \\ & & * & * \\ & & & * \\ & & & \end{pmatrix}$$

$$U^T AV = B \quad \Rightarrow \quad AV = UB, \quad A^T U = VB^T$$

- Golub-Kahan process on A, b

$$\beta_1 u_1 = b, \quad \alpha_1 v_1 = A^T u_1$$

$$\beta_2 u_2 = Av_1 - \alpha_1 v_1$$
$$\alpha_2 v_2 = A^T u_2 - \beta_2 v_1$$

$$B_k = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & * & * & \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}$$

$$U_k = \begin{pmatrix} u_1 & u_2 & \dots & u_k \end{pmatrix}$$
$$V_k = \begin{pmatrix} v_1 & v_2 & \dots & v_k \end{pmatrix}$$

$$AV_k = U_{k+1} B_k, \quad A^T U_k = V_k L_k^T$$

Upper or lower bidiagonal?

- Dense A

$$AV = UB \quad = \quad U \begin{pmatrix} * & * & & \\ & * & * & \\ & & * & * \\ & & & * \\ & & & \end{pmatrix}$$

Upper or lower bidiagonal?

- Dense A

$$AV = UB = U \begin{pmatrix} * & * & & \\ & * & * & \\ & & * & * \\ & & & * \\ & & & \end{pmatrix}$$

- Sparse A with $b = \beta_1 u_1$

$$AV_k = U_{k+1} B_k \quad \Rightarrow \quad \begin{pmatrix} b & AV_k \end{pmatrix} = U_{k+1} \begin{pmatrix} \beta_1 e_1 & B_k \end{pmatrix}$$

$$\Rightarrow \quad \begin{pmatrix} b & A \end{pmatrix} \begin{pmatrix} 1 & \\ & V_k \end{pmatrix} = U_{k+1} \begin{pmatrix} * & * & & \\ & * & * & \\ & & * & * \\ & & & * \end{pmatrix}$$

Tridiagonalization of unsymmetric or rectangular A (the "new method")

Rectangular A

- Tridiagonalization for dense EVD (eigenvalues)

$$U_1^T A = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ & * & * & * \\ & * & * & * \end{pmatrix}, \quad U_1^T A V_1 = \begin{pmatrix} * & * & & \\ * & * & * & * \\ & * & * & * \\ & * & * & * \end{pmatrix} \quad \cdots \rightarrow \begin{pmatrix} * & * & & \\ * & * & * & \\ & * & * & * \\ & & * & * \end{pmatrix}$$

$$U^T A V = T \quad \Rightarrow \quad AV = UT, \quad A^T U = V T^T$$

Rectangular A

- Tridiagonalization for dense EVD (eigenvalues)

$$U_1^T A = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ & * & * & * \\ & * & * & * \end{pmatrix}, \quad U_1^T A V_1 = \begin{pmatrix} * & * & & \\ * & * & * & * \\ & * & * & * \\ & * & * & * \end{pmatrix} \quad \cdots \rightarrow \begin{pmatrix} * & * & & \\ * & * & * & \\ & * & * & * \\ & & * & * \end{pmatrix}$$

$$U^T A V = T \quad \Rightarrow \quad AV = UT, \quad A^T U = VT^T$$

- Bi-tridiagonalization process on A, b, c

$$\beta_1 u_1 = b \qquad \gamma_1 v_1 = c$$

$$p_1 = A v_1 \qquad \alpha_1 = u_1^T p_1$$

$$\beta_2 u_2 = p_1 - \alpha_1 u_1 - \gamma_1 u_0$$

$$q_1 = A^T u_2$$

$$\gamma_2 v_2 = q_1 - \alpha_1 v_1 - \beta_1 v_0$$

$$T_k = \begin{pmatrix} \alpha_1 & \gamma_2 & & \\ \beta_2 & \alpha_2 & \gamma_3 & \\ & * & * & * \\ & & \beta_k & \alpha_k \end{pmatrix}$$

$$U_k = \begin{pmatrix} u_1 & u_2 & \ldots & u_k \end{pmatrix}$$

$$V_k = \begin{pmatrix} v_1 & v_2 & \ldots & v_k \end{pmatrix}$$

$$\begin{aligned} AV_k &= U_k T_k + \beta_{k+1} u_{k+1} e_k^T \\ A^T U_k &= V_k T_k^T + \gamma_{k+1} v_{k+1} e_k^T \end{aligned}$$

Original motivation (1981)

Original motivation (1981)

- CG, SYMMLQ, MINRES work well for symmetric $Ax = b$

Original motivation (1981)

- CG, SYMMLQ, MINRES work well for symmetric $Ax = b$
- Bi-tridiagonalization of unsymmetric A is no more than twice the work and storage per iteration

Original motivation (1981)

- CG, SYMMLQ, MINRES work well for symmetric $Ax = b$
- Bi-tridiagonalization of unsymmetric A is no more than twice the work and storage per iteration
- If A is symmetric, we get Lanczos

Original motivation (1981)

- CG, SYMMLQ, MINRES work well for symmetric $Ax = b$
- Bi-tridiagonalization of unsymmetric A is no more than twice the work and storage per iteration
- If A is symmetric, we get Lanczos
- If A is nearly symmetric, total itns should be not much more

Solving symmetric $Ax = b$ via Lanczos

Symmetric $Ax = b$

Lanzcos process:

$$AV_k = V_{k+1}H_k, \qquad H_k = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \beta_3 & \\ & * & * & * \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}$$

Symmetric $Ax = b$

Lanzcos process:

$$AV_k = V_{k+1} H_k, \qquad H_k = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \beta_3 & \\ & * & * & * \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}$$

Suppose $x_k = V_k w_k$ for some w_k

Symmetric $Ax = b$

Lanzcos process:

$$AV_k = V_{k+1} H_k, \qquad H_k = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \beta_3 & \\ & * & * & * \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}$$

Suppose $x_k = V_k w_k$ for some w_k

- $r_k = b - Ax_k$

Symmetric $Ax = b$

Lanzcos process:

$$AV_k = V_{k+1}H_k, \qquad H_k = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \beta_3 & \\ & * & * & * \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}$$

Suppose $x_k = V_k w_k$ for some w_k

- $r_k = b - Ax_k$
- $r_k = V_{k+1}(\beta_1 e_1 - H_k w_k)$

Symmetric $Ax = b$

Lanzcos process:

$$AV_k = V_{k+1} H_k, \qquad H_k = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \beta_3 & \\ & * & * & * \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}$$

Suppose $x_k = V_k w_k$ for some w_k

- $r_k = b - Ax_k$
- $r_k = V_{k+1}(\beta_1 e_1 - H_k w_k)$
- $\|r_k\|$ will be small if $H_k w_k \approx \beta_1 e_1$

Symmetric $Ax = b$

Lanzcos process:

$$AV_k = V_{k+1} H_k, \qquad H_k = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \beta_3 & \\ & * & * & * \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}$$

Suppose $x_k = V_k w_k$ for some w_k

- $r_k = b - Ax_k$
- $r_k = V_{k+1}(\beta_1 e_1 - H_k w_k)$
- $\|r_k\|$ will be small if $H_k w_k \approx \beta_1 e_1$

Three subproblems make $H_k w_k \approx \beta_1 e_1 \quad \Rightarrow \quad$ CG, SYMMLQ, MINRES

Symmetric $Ax = b$

Lanzcos process:

$$AV_k = V_{k+1} H_k, \qquad H_k = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \beta_3 & \\ & * & * & * \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}$$

Suppose $x_k = V_k w_k$ for some w_k

- $r_k = b - Ax_k$
- $r_k = V_{k+1}(\beta_1 e_1 - H_k w_k)$
- $\|r_k\|$ will be small if $H_k w_k \approx \beta_1 e_1$

Three subproblems make $H_k w_k \approx \beta_1 e_1 \quad \Rightarrow \quad$ CG, SYMMLQ, MINRES
(e.g. $T_k w_k = \beta_1 e_1$ for CG)

Symmetric → Unsymmetric

Lanczos on $\begin{pmatrix} I & A \\ A^T & \end{pmatrix} \begin{pmatrix} r \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$ (general A)

leads to Golub-Kahan and LSQR

Symmetric → Unsymmetric

Lanczos on $\begin{pmatrix} I & A \\ A^T & \end{pmatrix} \begin{pmatrix} r \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$ (general A)

leads to Golub-Kahan and LSQR

Lanczos on $\begin{pmatrix} & A \\ A^T & \end{pmatrix} \begin{pmatrix} y \\ x \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}$ (square A)

is not equivalent to bi-tridiagonalization (but seems worth trying!)

Symmetric → Unsymmetric

Lanczos on $\begin{pmatrix} I & A \\ A^T & \end{pmatrix} \begin{pmatrix} r \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$ (general A)

leads to Golub-Kahan and LSQR

Lanczos on $\begin{pmatrix} & A \\ A^T & \end{pmatrix} \begin{pmatrix} y \\ x \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}$ (square A)

is not equivalent to bi-tridiagonalization (but seems worth trying!)

Lanczos on $\begin{pmatrix} I & A \\ A^T & \end{pmatrix} \begin{pmatrix} r \\ x \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}$ (general A)

is not equivalent either (Who would like to try?)

Solving unsymmetric $Ax = b$
via bi-tridiagonalization

Unsymmetric $Ax = b$

Bi-tridiag process:

$$\begin{aligned} AV_k &= U_k T_k + \beta_{k+1} u_{k+1} e_k^T &\equiv& \quad U_{k+1} H_k^{\beta} \\ A^T U_k &= V_k T_k^T + \gamma_{k+1} v_{k+1} e_k^T &\equiv& \quad V_{k+1} H_k^{\gamma} \end{aligned}$$

Unsymmetric $Ax = b$

Bi-tridiag process:

$$
\begin{aligned}
AV_k &= U_k T_k + \beta_{k+1} u_{k+1} e_k^T &\equiv& \quad U_{k+1} H_k^\beta \\
A^T U_k &= V_k T_k^T + \gamma_{k+1} v_{k+1} e_k^T &\equiv& \quad V_{k+1} H_k^\gamma
\end{aligned}
$$

Suppose $x_k = V_k w_k$ for some w_k

Unsymmetric $Ax = b$

Bi-tridiag process:

$$\begin{aligned} AV_k &= U_k T_k + \beta_{k+1} u_{k+1} e_k^T &\equiv& \quad U_{k+1} H_k^{\beta} \\ A^T U_k &= V_k T_k^T + \gamma_{k+1} v_{k+1} e_k^T &\equiv& \quad V_{k+1} H_k^{\gamma} \end{aligned}$$

Suppose $x_k = V_k w_k$ for some w_k
Three subproblems make $H_k^{\beta} w_k \approx \beta_1 e_1 \quad \Rightarrow \quad$ UCG, USYMLQ, USYMQR

Unsymmetric $Ax = b$

Bi-tridiag process:

$$\begin{aligned} AV_k &= U_k T_k + \beta_{k+1} u_{k+1} e_k^T &\equiv& \; U_{k+1} H_k^{\beta} \\ A^T U_k &= V_k T_k^T + \gamma_{k+1} v_{k+1} e_k^T &\equiv& \; V_{k+1} H_k^{\gamma} \end{aligned}$$

Suppose $x_k = V_k w_k$ for some w_k
Three subproblems make $H_k^{\beta} w_k \approx \beta_1 e_1 \quad \Rightarrow \quad$ UCG, USYMLQ, USYMQR

Similarly, let $y_k = U_k \bar{w}_k$ to solve $A^T y = c$
Three subproblems make $H_k^{\gamma} y_k \approx \gamma_1 e_1$

Unsymmetric $Ax = b$

Bi-tridiag process:

$$
\begin{aligned}
AV_k &= U_k T_k + \beta_{k+1} u_{k+1} e_k^T &\equiv& \quad U_{k+1} H_k^{\beta} \\
A^T U_k &= V_k T_k^T + \gamma_{k+1} v_{k+1} e_k^T &\equiv& \quad V_{k+1} H_k^{\gamma}
\end{aligned}
$$

Suppose $x_k = V_k w_k$ for some w_k
Three subproblems make $H_k^{\beta} w_k \approx \beta_1 e_1 \quad \Rightarrow \quad$ UCG, USYMLQ, USYMQR

Similarly, let $y_k = U_k \bar{w}_k$ to solve $A^T y = c$
Three subproblems make $H_k^{\gamma} y_k \approx \gamma_1 e_1$

Not much extra effort to get both x_k and y_k

Elizabeth Yip's motivation (1982)

(Boeing Computer Services Co.)

Elizabeth's SIAM conference abstract (1982)

CG method for unsymmetric matrices applied to PDE problems

We present a CG-type method to solve $Ax = b$, where A is an arbitrary nonsingular unsymmetric matrix. The algorithm is equivalent to an orthogonal tridiagonalization of A.

Elizabeth's SIAM conference abstract (1982)

CG method for unsymmetric matrices applied to PDE problems

We present a CG-type method to solve $Ax = b$, where A is an arbitrary nonsingular unsymmetric matrix. The algorithm is equivalent to an orthogonal tridiagonalization of A.

Each iteration takes more work than the orthogonal bidiagonalization proposed by Golub-Kahan, Paige-Saunders for sparse least squares problems (LSQR).

Elizabeth's SIAM conference abstract (1982)

CG method for unsymmetric matrices applied to PDE problems

We present a CG-type method to solve $Ax = b$, where A is an arbitrary nonsingular unsymmetric matrix. The algorithm is equivalent to an orthogonal tridiagonalization of A.

Each iteration takes more work than the orthogonal bidiagonalization proposed by Golub-Kahan, Paige-Saunders for sparse least squares problems (LSQR).

However, ... the condition number for our tridiagonalization is the square root of that for the bidiagonalization.

Elizabeth's SIAM conference abstract (1982)

CG method for unsymmetric matrices applied to PDE problems

We present a CG-type method to solve $Ax = b$, where A is an arbitrary nonsingular unsymmetric matrix. The algorithm is equivalent to an orthogonal tridiagonalization of A.

Each iteration takes more work than the orthogonal bidiagonalization proposed by Golub-Kahan, Paige-Saunders for sparse least squares problems (LSQR).

However, . . . the condition number for our tridiagonalization is the square root of that for the bidiagonalization.

We apply a preconditioned version (Fast Poisson) to the difference equation of unsteady transonic flow with small disturbances. (Compared with ORTHOMIN(5))

Numerical results with unsymmetric tridiagonalization

Numerical results (SSY 1988)

$$A = \begin{pmatrix} B & -I & & & \\ -I & B & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & B & -I \\ & & & -I & B \end{pmatrix} \qquad B = \text{tridiag}\begin{pmatrix} -1-\delta & 4 & -1+\delta \end{pmatrix}$$

400×400 $\qquad$ 20×20

Numerical results (SSY 1988)

$$A = \begin{pmatrix} B & -I & & & \\ -I & B & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & B & -I \\ & & & -I & B \end{pmatrix} \qquad B = \text{tridiag}\begin{pmatrix} -1-\delta & 4 & -1+\delta \end{pmatrix}$$

400×400 $\qquad$ 20×20

Megaflops to reach $\|r\| \le 10^{-6}\|b\|$:

δ	0.0	0.01	0.1	1.0	10.0	100.0
ORTHOMIN(5)	0.31	0.57	0.75	0.83	2.55	2.11
LSQR	0.28	1.38	1.48	0.80	0.57	0.27
USYMQR	0.30	1.88	1.98	1.41	0.99	0.64

Numerical results (SSY 1988)

$$A = \begin{pmatrix} B & -I & & & \\ -I & B & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & B & -I \\ & & & -I & B \end{pmatrix}$$

400×400

$$B = \text{tridiag}\begin{pmatrix} -1-\delta & 4 & -1+\delta \end{pmatrix}$$

20×20

Megaflops to reach $\|r\| \le 10^{-6}\|b\|$:

δ	0.0	0.01	0.1	1.0	10.0	100.0
ORTHOMIN(5)	0.31	0.57	0.75	0.83	2.55	2.11
LSQR	0.28	1.38	1.48	0.80	0.57	0.27
USYMQR	0.30	1.88	1.98	1.41	0.99	0.64

Bottom line:
ORTHOMIN sometimes good, can fail. LSQR always better than USYMQR

Numerical results (Reichel and Ye 2006)

- Focused on rectangular A and least-squares
  (Forgot about SSY88 and USYMQR — hence GLSQR)

Numerical results (Reichel and Ye 2006)

- Focused on rectangular A and least-squares
  (Forgot about SSY88 and USYMQR — hence GLSQR)
- Three numerical examples (all square!)

Numerical results (Reichel and Ye 2006)

- Focused on rectangular A and least-squares
(Forgot about SSY88 and USYMQR — hence GLSQR)
- Three numerical examples (all square!)
- Remember $x_1 \propto c$

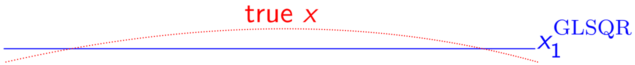
Numerical results (Reichel and Ye 2006)

- Focused on rectangular A and least-squares
  (Forgot about SSY88 and USYMQR — hence GLSQR)
- Three numerical examples (all square!)
- Remember $x_1 \propto c$
- Focused on choice of c
  stopping early
  looking at $x_k = \begin{pmatrix} x_{k1} & x_{k2} & \dots & x_{kn} \end{pmatrix}$

Numerical results (Reichel and Ye 2006)

- Focused on rectangular A and least-squares
  (Forgot about SSY88 and USYMQR — hence GLSQR)
- Three numerical examples (all square!)
- Remember $x_1 \propto c$
- Focused on choice of c
  stopping early
  looking at $x_k = \begin{pmatrix} x_{k1} & x_{k2} & \dots & x_{kn} \end{pmatrix}$

Example 1: We know $x \approx$ constant. Choose $c = \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix}^T$



Example 2 (Star cluster)

- 256×256 pixels ($n = 65536$), 470 stars

Example 2 (Star cluster)

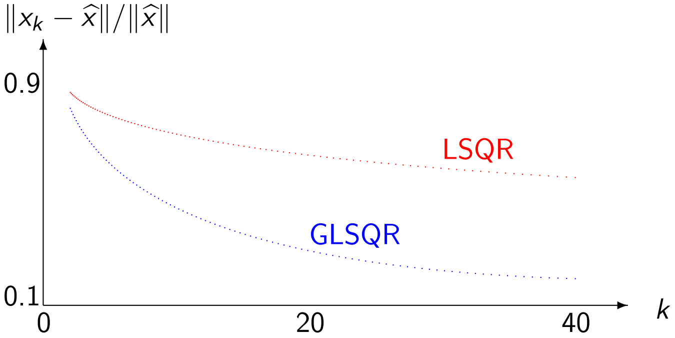
- 256×256 pixels ($n = 65536$), 470 stars
- Square $Ax \approx b$, choose $c = b$

Example 2 (Star cluster)

- 256×256 pixels ($n = 65536$), 470 stars
- Square $Ax \approx b$, choose $c = b$
- Compare error in x_k^{LSQR} and x_k^{GLSQR} for 40 iterations

Example 2 (Star cluster)

- 256×256 pixels ($n = 65536$), 470 stars
- Square $Ax \approx b$, choose $c = b$
- Compare error in x_k^{LSQR} and x_k^{GLSQR} for 40 iterations



Conclusions

Subspaces

- Unsymmetric Lanczos generates two Krylov subspaces:

$$
\begin{aligned}
U_k &\in \operatorname{span}\{b \quad Ab \quad A^2 b \quad \ldots \quad A^{k-1} b\} \\
V_k &\in \operatorname{span}\{c \quad A^T c \quad (A^T)^2 c \quad \ldots \quad (A^T)^{k-1} c\}
\end{aligned}
$$

Subspaces

- Unsymmetric Lanczos generates two Krylov subspaces:

$$\begin{array}{lll} U_k & \in & \text{span}\{b \;\; Ab \;\; A^2 b \;\; \ldots \;\; A^{k-1} b\} \\ V_k & \in & \text{span}\{c \;\; A^T c \;\; (A^T)^2 c \;\; \ldots \;\; (A^T)^{k-1} c\} \end{array}$$

- Bi-tridiagonalization generates

$$\begin{array}{lll} U_{2k} & \in & \text{span}\{b \;\; AA^T b \;\; \ldots \;\; (AA^T)^{k-1} b \;\; Ac \;\; (AA^T)Ac \;\; \ldots\} \\ V_{2k} & \in & \text{span}\{c \;\; A^T Ac \;\; \ldots \;\; (A^T A)^{k-1} c \;\; A^T b \;\; (A^T A) A^T b \;\; \ldots\} \end{array}$$

Functionals $c^T x$, $b^T y$

- Lu and Darmofal (SISC 2003) use unsymmetric Lanczos with QMR to solve $Ax = b$ and $A^T y = c$ *simultaneously* and to estimate $c^T x$ and $b^T y$ at a *superconvergent rate*:

$$|c^T x_k - c^T x| \approx |b^T y_k - b^T y| \approx \frac{\|b - Ax_k\| \|c - A^T y_k\|}{\sigma_{\min}(A)}$$

Functionals $c^T x$, $b^T y$

- Lu and Darmofal (SISC 2003) use unsymmetric Lanczos with QMR to solve $Ax = b$ and $A^T y = c$ *simultaneously* and to estimate $c^T x$ and $b^T y$ at a *superconvergent rate*:

$$|c^T x_k - c^T x| \approx |b^T y_k - b^T y| \approx \frac{\|b - Ax_k\| \|c - A^T y_k\|}{\sigma_{\min}(A)}$$

- Golub, Stoll and Wathen (draft 2007) plan to use bi-triagonalization with GLSQR to do likewise

Functionals $c^T x$, $b^T y$

- Lu and Darmofal (SISC 2003) use unsymmetric Lanczos with QMR to solve $Ax = b$ and $A^T y = c$ *simultaneously* and to estimate $c^T x$ and $b^T y$ at a *superconvergent rate*:

$$|c^T x_k - c^T x| \approx |b^T y_k - b^T y| \approx \frac{\|b - Ax_k\| \|c - A^T y_k\|}{\sigma_{\min}(A)}$$

- Golub, Stoll and Wathen (draft 2007) plan to use bi-triagonalization with GLSQR to do likewise

Thanks for your patience!!