GMINRES or GLSQR?

Michael Saunders
Systems Optimization Laboratory, Stanford University

Symposium on Gene Golub's Legacy:
Matrix Computations – Foundation and Future
Stanford University, March 1, 2008

Outline

1. Iterative matrix reductions

Outline

1. Iterative matrix reductions
2. CG-type iterative solvers

Outline

1. Iterative matrix reductions
2. CG-type iterative solvers
3. Bi-tridiagonalization of general A

Outline

1. Iterative matrix reductions
2. CG-type iterative solvers
3. Bi-tridiagonalization of general A
4. Numerical results

Outline

Abstract

Given a general matrix A and starting vectors (b, c), we can construct orthonormal matrices U_k and V_k that reduce A to tridiagonal form: $AV_k \approx U_k T_k$ and $A^T U_k \approx V_k T_k^T$. Saunders, Simon, and Yip (1988) proposed methods for solving square systems $Ax = b$ and $A^T y = c$ simultaneously. The solver USYMQR becomes equivalent to MINRES in the symmetric case with $b = c$.

The method was rediscovered by Reichel and Ye (2006) with emphasis on rectangular systems. For implementation reasons it was regarded as a generalization of LSQR (although it does not reduce to LSQR in any special case). The method has now been applied to two square systems by Golub, Stoll, and Wathen (2007) with focus on estimating $c^T x$ and $y^T b$.

Iterative matrix reductions

• Lanczos	1950	Symmetric
• Arnoldi	1951	Square
• Golub-Kahan	1965	Rectangular
• "New" tridiagonalization	1981–1988, 2006	Rectangular

We call them *processes*

Lanczos-type processes

- Need starting vector $v_1 = b$ (Assume $\|b\| = 1$)

Lanczos-type processes

- Need starting vector $v_1 = b$ (Assume $\|b\| = 1$)
- Use matrix-vector products Av_k and perhaps $A^T u_k$

Lanczos-type processes

- Need starting vector $v_1 = b$ (Assume $\|b\| = 1$)
- Use matrix-vector products Av_k and perhaps $A^T u_k$
- Generate $V_k = \begin{pmatrix} v_1 & v_2 & \dots & v_k \end{pmatrix}$ and H_k or B_k

Lanczos-type processes

- Need starting vector $v_1 = b$ (Assume $\|b\| = 1$)
- Use matrix-vector products Av_k and perhaps $A^T u_k$
- Generate $V_k = \begin{pmatrix} v_1 & v_2 & \dots & v_k \end{pmatrix}$ and H_k or B_k

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

Lanczos-type processes

- Need starting vector $v_1 = b$ (Assume $\|b\| = 1$)
- Use matrix-vector products Av_k and perhaps $A^T u_k$
- Generate $V_k = \begin{pmatrix} v_1 & v_2 & \dots & v_k \end{pmatrix}$ and H_k or B_k

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

$V_k^T V_k = I$ in theory, but at least $\|v_k\| = 1$

CG-type iterative solvers

for $Ax = b$

• MINRES	Lanczos	$AV_k = V_{k+1}H_k$
• GMRES	Arnoldi	$AV_k = V_{k+1}H_k$
• LSQR	Golub-Kahan bidiagonalization	$AV_k = U_{k+1}B_k$ $A^TU_{k+1} = V_kB_k^T + \cdots$
• GMINRES? GLSQR?	S-Simon-Yip bi-tridiagonalization	$AV_k = U_{k+1}H_k$ $A^TU_k = V_{k+1}\bar{H}_k$

CG-type iterative solvers

for $Ax = b$

• MINRES	Lanczos	$AV_k = V_{k+1} H_k$
• GMRES	Arnoldi	$AV_k = V_{k+1} H_k$
• LSQR	Golub-Kahan bidiagonalization	$AV_k = U_{k+1} B_k$ $A^T U_{k+1} = V_k B_k^T + \cdots$
• GMINRES? GLSQR?	S-Simon-Yip bi-tridiagonalization	$AV_k = U_{k+1} H_k$ $A^T U_k = V_{k+1} \bar{H}_k$

They all solve $\min \|X w_k - e_1\|$ and define $x_k = V_k w_k$

$(X = H_k$ or $B_k)$

MINRES

$$AV_k = V_{k+1}H_k, \qquad v_1 = b, \qquad H_k = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \beta_3 & \\ & * & * & * \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}$$

Let $x_k = V_k w_k$, where we solve $\min \| H_k w_k - e_1 \|$

MINRES

$$AV_k = V_{k+1}H_k, \qquad v_1 = b, \qquad H_k = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \beta_3 & \\ & * & * & * \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}$$

Let $x_k = V_k w_k$, where we solve $\min \|H_k w_k - e_1\|$

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

MINRES

$$AV_k = V_{k+1}H_k, \qquad v_1 = b, \qquad H_k = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \beta_3 & \\ & * & * & * \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}$$

Let $x_k = V_k w_k$, where we solve $\min \|H_k w_k - e_1\|$

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

MINRES

$$AV_k = V_{k+1}H_k, \qquad v_1 = b, \qquad H_k = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \beta_3 & \\ & * & * & * \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}$$

Let $x_k = V_k w_k$, where we solve $\min \| H_k w_k - e_1 \|$

- $Ax_k - b = V_{k+1}(H_k w_k - e_1)$
- Will be small if $H_k w_k \approx e_1$
- 3 subprobs make $H_k w_k \approx e_1 \quad \rightarrow$ CG, SYMMLQ, MINRES

MINRES

$$AV_k = V_{k+1}H_k, \qquad v_1 = b, \qquad H_k = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \beta_3 & \\ & * & * & * \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix}$$

Let $x_k = V_k w_k$, where we solve $\min \|H_k w_k - e_1\|$

- $Ax_k - b = V_{k+1}(H_k w_k - e_1)$
- Will be small if $H_k w_k \approx e_1$
- 3 subprobs make $H_k w_k \approx e_1$ $\longrightarrow$ CG, SYMMLQ, MINRES
- $v_1 = b \Rightarrow$ no need to assume V_k has orthogonal columns

Symmetric $\rightarrow$ Unsymmetric $Ax \approx b$

Lanczos on $\begin{pmatrix} I & A \\ A^T & -\delta^2 I \end{pmatrix} \begin{pmatrix} r \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$ (general A)
with CG-type subproblem gives 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!)

Tridiagonalization of general A using orthogonal matrices

Bi-tridiagonalization

- 1988 Saunders, Simon, and Yip, SINUM 25

  "Two CG-type methods for unsymmetric linear equations"

  Focus on square A

  USYMLQ and USYMQR (GSYMMLQ and GMINRES)

Bi-tridiagonalization

- 1988 Saunders, Simon, and Yip, SINUM 25

  "Two CG-type methods for unsymmetric linear equations"

  Focus on square A

  USYMLQ and USYMQR (GSYMMLQ and GMINRES)

- 2006 Reichel and Ye

  "A generalized LSQR algorithm"

  Focus on rectangular A

  GLSQR

Bi-tridiagonalization

- 1988 Saunders, Simon, and Yip, SINUM 25

  "Two CG-type methods for unsymmetric linear equations"

  Focus on square A

  USYMLQ and USYMQR (GSYMMLQ and GMINRES)

- 2006 Reichel and Ye

  "A generalized LSQR algorithm"

  Focus on rectangular A

  GLSQR

- 2007 Golub, Stoll, and Wathen

  "Approximation of the scattering amplitude"

  Focus on $Ax = b$, $A^T y = c$

  and estimation of $c^T x$, $b^T y$ (without x, y)

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 and MINRES etc

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 and MINRES etc
- If A is nearly symmetric, total itns should be not much more

Elizabeth Yip'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 Yip'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 Yip'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))

Unsymmetric $Ax = b$, $A^T y = c$

Bi-tridiag process starting with (b, c)

$$\begin{aligned} AV_k &= U_{k+1} H_k \\ A^T U_k &= V_{k+1} \bar{H}_k \end{aligned}$$

Unsymmetric $Ax = b$, $A^T y = c$

Bi-tridiag process starting with (b, c)

$$\begin{aligned} AV_k &= U_{k+1} H_k \\ A^T U_k &= V_{k+1} \bar{H}_k \end{aligned}$$

Suppose $x_k = V_k w_k$ for some w_k

Unsymmetric $Ax = b$, $A^T y = c$

Bi-tridiag process starting with (b, c)

$$\begin{aligned} AV_k &= U_{k+1} H_k \\ A^T U_k &= V_{k+1} \bar{H}_k \end{aligned}$$

Suppose $x_k = V_k w_k$ for some w_k

Three subproblems make $H_k w_k \approx e_1 \quad \Rightarrow \quad$ UCG, USYMLQ, USYMQR

Unsymmetric $Ax = b$, $A^T y = c$

Bi-tridiag process starting with (b, c)

$$\begin{aligned} AV_k &= U_{k+1} H_k \\ A^T U_k &= V_{k+1} \bar{H}_k \end{aligned}$$

Suppose $x_k = V_k w_k$ for some w_k

Three subproblems make $H_k w_k \approx e_1 \quad \Rightarrow \quad$ UCG, USYMLQ, USYMQR

Similarly, let $y_k = U_k \bar{w}_k$ to solve $A^T y = c$

Three subproblems make $\bar{H}_k \bar{w}_k \approx e_1$

Unsymmetric $Ax = b$, $A^T y = c$

Bi-tridiag process starting with (b, c)

$$\begin{aligned} AV_k &= U_{k+1} H_k \\ A^T U_k &= V_{k+1} \bar{H}_k \end{aligned}$$

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

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

Not much extra effort to get both x_k and y_k

Numerical results with unsymmetric tridiagonalization

Numerical results (SSY 1988)

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

400×400

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

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)

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

400×400 $\qquad$ 20×20

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

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 $x_1 \propto c$

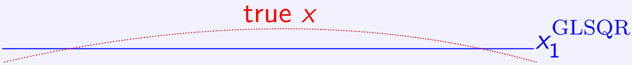
Numerical results (Reichel and Ye 2006)

- 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)

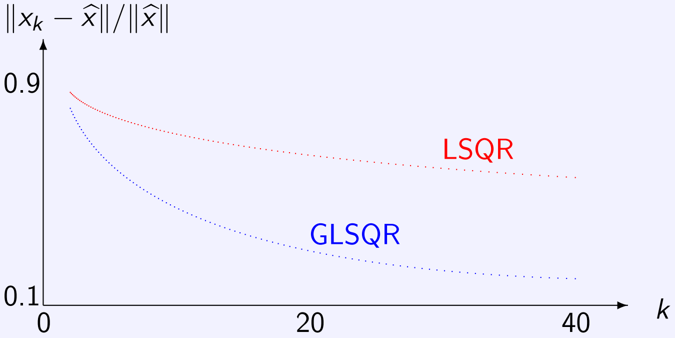
- 256×256 pixels ($n = 65536$), 470 stars
- Square $Ax \approx b$, choose $c = b$

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:

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

Subspaces

- Unsymmetric Lanczos generates two Krylov subspaces:

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

- Bi-tridiagonalization generates

$$
\begin{aligned}
U_{2k} &\in \operatorname{span}\{b \;\; AA^T b \;\; \dots \;\; (AA^T)^{k-1} b \;\; Ac \;\; (AA^T)Ac \;\; \dots\} \\
V_{2k} &\in \operatorname{span}\{c \;\; A^T Ac \;\; \dots \;\; (A^T A)^{k-1} c \;\; A^T b \;\; (A^T A)A^T b \;\; \dots\}
\end{aligned}
$$

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 (2007) 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 (2007) use bi-triagonalization with GLSQR to do likewise
  - Matrices, moments, and quadrature

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 (2007) use bi-triagonalization with GLSQR to do likewise
  - Matrices, moments, and quadrature
  - Golub, Minerbo, and Saylor
    Nine ways to compute the scattering cross-section
    (1): Estimating $c^T x$ iteratively

Block Lanczos

The Bi-tridiagonalization process is equivalent to

- Block Lanczos on $A^T A$ with starting block $\begin{pmatrix} c & A^T b \end{pmatrix}$
  Parlett 1987

Block Lanczos

The Bi-tridiagonalization process is equivalent to

- Block Lanczos on A^TA with starting block $\begin{pmatrix} c & A^Tb \end{pmatrix}$
  Parlett 1987
- Block Lanczos on $\begin{pmatrix} & A \\ A^T & \end{pmatrix}$ with starting block $\begin{pmatrix} & b \\ c & \end{pmatrix}$
  Golub, Stoll, and Wathen 2007

Block Lanczos

The Bi-tridiagonalization process is equivalent to

- Block Lanczos on $A^T A$ with starting block $\begin{pmatrix} c & A^T b \end{pmatrix}$
  Parlett 1987
- Block Lanczos on $\begin{pmatrix} & A \\ A^T & \end{pmatrix}$ with starting block $\begin{pmatrix} & b \\ c & \end{pmatrix}$
  Golub, Stoll, and Wathen 2007

There are two ways of spreading light.
To be the candle
or the mirror that reflects it.

– Edith Wharton

Block Lanczos

The Bi-tridiagonalization process is equivalent to

- Block Lanczos on A^TA with starting block $\begin{pmatrix} c & A^Tb \end{pmatrix}$
  Parlett 1987
- Block Lanczos on $\begin{pmatrix} & A \\ A^T & \end{pmatrix}$ with starting block $\begin{pmatrix} & b \\ c & \end{pmatrix}$
  Golub, Stoll, and Wathen 2007

There are two ways of spreading light.
To be the candle
or the mirror that reflects it.

– Edith Wharton

Block Lanczos

The Bi-tridiagonalization process is equivalent to

- Block Lanczos on A^TA with starting block $\begin{pmatrix} c & A^Tb \end{pmatrix}$
  Parlett 1987
- Block Lanczos on $\begin{pmatrix} & A \\ A^T & \end{pmatrix}$ with starting block $\begin{pmatrix} & b \\ c & \end{pmatrix}$
  Golub, Stoll, and Wathen 2007

There are two ways of spreading light.
To be the candle
or the mirror that reflects it.

– Edith Wharton

Messages from Marcus Grote (Switzerland) and Yong Sun (China)