LSMR: An iterative algorithm for sparse least-squares problems

David Fong   Michael Saunders

Institute for Computational and Mathematical Engineering (iCME)
Stanford University

2nd IMA Conference on Numerical Linear Algebra and Optimisation
University of Birmingham, UK   September 13–15, 2010

LSMR in one slide

solve $Ax = b$
min $\|Ax - b\|_2$

LSMR in one slide

solve $Ax = b$
min $\|Ax - b\|_2$

min $\left\| \begin{pmatrix} A \\ \lambda I \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2$

LSMR in one slide

solve $Ax = b$
min $\|Ax - b\|_2$

$$\min \left\| \begin{pmatrix} A \\ \lambda I \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2$$

LSQR $\equiv$ CG on the normal equation
LSMR $\equiv$ MINRES on the normal equation

LSMR in one slide

solve $Ax = b$
min $\|Ax - b\|_2$

$$\min \left\| \begin{pmatrix} A \\ \lambda I \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2$$

LSQR $\equiv$ CG on the normal equation
LSMR $\equiv$ MINRES on the normal equation

- Almost same complexity as LSQR
- Better convergence properties for inexact solves

LSQR

Iterative algorithm for

$$\min \left\| \begin{pmatrix} A \\ \lambda I \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2$$

LSQR

Iterative algorithm for

$$\min \left\| \begin{pmatrix} A \\ \lambda I \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2$$

Properties

- A is rectangular ($m \times n$) and often sparse
- A can be an operator
- CG on the normal equation $\quad (A^T A + \lambda^2 I)x = A^T b$
- Av, $A^T u$ plus $O(m + n)$ operations per iteration

Monotone convergence of residual

Measure of Convergence

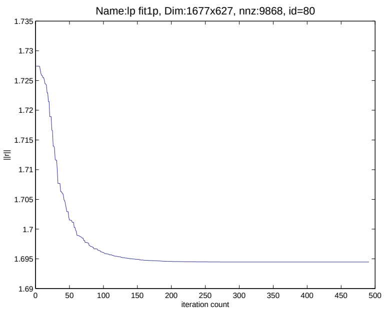
- $r_k = b - Ax_k$
- $\|r_k\| \to \|\hat{r}\|$, $\|A^T r_k\| \to 0$

Monotone convergence of residual

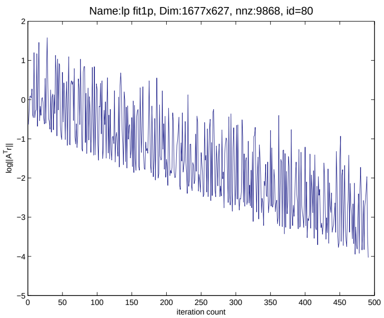
Measure of Convergence

- $r_k = b - Ax_k$
- $\|r_k\| \to \|\hat{r}\|$, $\|A^T r_k\| \to 0$

LSQR $\|r_k\|$



LSQR $\log \|A^T r_k\|$



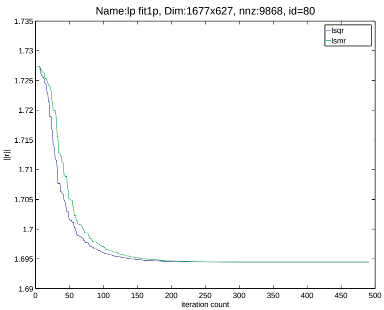
Monotone convergence of residual

Measure of Convergence

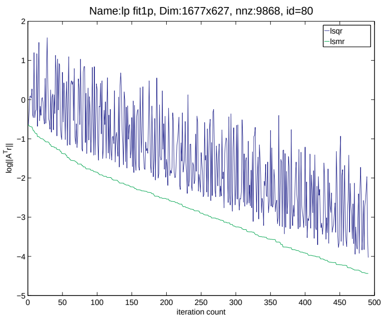
- $r_k = b - Ax_k$
- $\|r_k\| \rightarrow \|\hat{r}\|$, $\|A^T r_k\| \rightarrow 0$

— LSQR
— LSMR

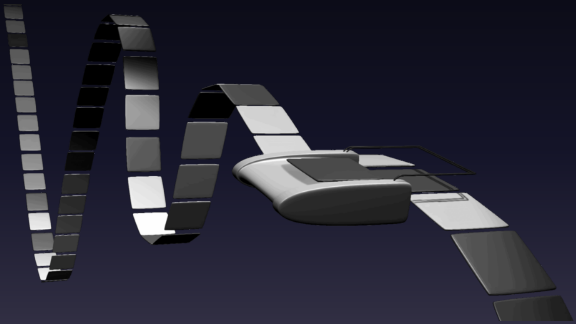
$\|r_k\|$



$\log \|A^T r_k\|$



LSMR Algorithm



Golub-Kahan bidiagonalization

Given A $(m \times n)$ and b $(m \times 1)$

Direct bidiagonalization

$$U^T \begin{pmatrix} b & A \end{pmatrix} V = B$$

Golub-Kahan bidiagonalization

Given A $(m \times n)$ and b $(m \times 1)$

Direct bidiagonalization

$$U^T \begin{pmatrix} b & A \end{pmatrix} V = B$$

Iterative bidiagonalization

1. $\beta_1 u_1 = b, \;\; \alpha_1 v_1 = A^T u_1$
2. for $k = 1, 2, \ldots,$ set

$$\begin{aligned} \beta_{k+1} u_{k+1} &= A v_k - \alpha_k u_k \\ \alpha_{k+1} v_{k+1} &= A^T u_{k+1} - \beta_{k+1} v_k \end{aligned}$$

Golub-Kahan bidiagonalization (2)

The process can be summarized by

$$\begin{aligned} b &= V_k(\beta_1 e_1) \\ AV_k &= U_{k+1} B_k \\ A^T U_k &= V_k B_k^T \begin{pmatrix} I_k \\ 0 \end{pmatrix} \end{aligned}$$

where

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

Golub-Kahan bidiagonalization (3)

V_k spans the Krylov subspace:

$$\mathrm{span}\{v_1, \ldots, v_k\} = \mathrm{span}\{A^Tb,\ (A^TA)A^Tb,\ \ldots,\ (A^TA)^{k-1}A^Tb\}$$

Golub-Kahan bidiagonalization (3)

Define $x_k = V_k y_k$

Subproblem to solve

$$\min_{y_k} \|r_k\| = \min_{y_k} \|\beta_1 e_1 - B_k y_k\| \qquad \text{(LSQR)}$$

$$\min_{y_k} \|A^T r_k\| = \min_{y_k} \left\| \bar{\beta}_1 e_1 - \begin{pmatrix} B_k^T B_k \\ \bar{\beta}_{k+1} e_k^T \end{pmatrix} y_k \right\| \qquad \text{(LSMR)}$$

where $r_k = b - A x_k, \ \bar{\beta}_k = \alpha_k \beta_k$

Least squares subproblem

$$\min_{y_k} \left\| \bar{\beta}_1 e_1 - \begin{pmatrix} B_k^T B_k \\ \bar{\beta}_{k+1} e_k^T \end{pmatrix} y_k \right\|$$

Least squares subproblem

$$\min_{y_k} \left\| \bar{\beta}_1 e_1 - \begin{pmatrix} B_k^T B_k \\ \bar{\beta}_{k+1} e_k^T \end{pmatrix} y_k \right\|$$
$$= \min_{y_k} \left\| \bar{\beta}_1 e_1 - \begin{pmatrix} R_k^T R_k \\ q_k^T R_k \end{pmatrix} y_k \right\|$$

$$Q_{k+1} B_k = \begin{pmatrix} R_k \\ 0 \end{pmatrix}, \quad R_k^T q_k = \bar{\beta}_{k+1} e_k$$

Least squares subproblem

$$\begin{aligned}
&\min_{y_k} \left\| \bar\beta_1 e_1 - \begin{pmatrix} B_k^T B_k \\ \bar\beta_{k+1} e_k^T \end{pmatrix} y_k \right\| \\
&= \min_{y_k} \left\| \bar\beta_1 e_1 - \begin{pmatrix} R_k^T R_k \\ q_k^T R_k \end{pmatrix} y_k \right\| && Q_{k+1} B_k = \begin{pmatrix} R_k \\ 0 \end{pmatrix}, \quad R_k^T q_k = \bar\beta_{k+1} e_k \\
&= \min_{t_k} \left\| \bar\beta_1 e_1 - \begin{pmatrix} R_k^T \\ \varphi_k e_k^T \end{pmatrix} t_k \right\| && t_k = R_k y_k, \quad q_k = (\bar\beta_{k+1}/(R_k)_{k,k}) e_k = \varphi_k e_k
\end{aligned}$$

Least squares subproblem

$$\begin{aligned}
&\min_{y_k} \left\| \bar\beta_1 e_1 - \begin{pmatrix} B_k^T B_k \\ \bar\beta_{k+1} e_k^T \end{pmatrix} y_k \right\| \\
&= \min_{y_k} \left\| \bar\beta_1 e_1 - \begin{pmatrix} R_k^T R_k \\ q_k^T R_k \end{pmatrix} y_k \right\| && Q_{k+1} B_k = \begin{pmatrix} R_k \\ 0 \end{pmatrix}, \quad R_k^T q_k = \bar\beta_{k+1} e_k \\
&= \min_{t_k} \left\| \bar\beta_1 e_1 - \begin{pmatrix} R_k^T \\ \varphi_k e_k^T \end{pmatrix} t_k \right\| && t_k = R_k y_k, \quad q_k = (\bar\beta_{k+1}/(R_k)_{k,k}) e_k = \varphi_k e_k \\
&= \min_{t_k} \left\| \begin{pmatrix} z_k \\ \bar\zeta_{k+1} \end{pmatrix} - \begin{pmatrix} \bar R_k \\ 0 \end{pmatrix} t_k \right\| && \bar Q_{k+1} \begin{pmatrix} R_k^T & \bar\beta_1 e_1 \\ \varphi_k e_k^T & 0 \end{pmatrix} = \begin{pmatrix} \bar R_k & z_k \\ 0 & \tilde\zeta_{k+1} \end{pmatrix}
\end{aligned}$$

Least squares subproblem

$$\min_{y_k} \left\| \bar\beta_1 e_1 - \begin{pmatrix} B_k^T B_k \\ \bar\beta_{k+1} e_k^T \end{pmatrix} y_k \right\|$$

$$= \min_{y_k} \left\| \bar\beta_1 e_1 - \begin{pmatrix} R_k^T R_k \\ q_k^T R_k \end{pmatrix} y_k \right\| \qquad Q_{k+1} B_k = \begin{pmatrix} R_k \\ 0 \end{pmatrix}, \quad R_k^T q_k = \bar\beta_{k+1} e_k$$

$$= \min_{t_k} \left\| \bar\beta_1 e_1 - \begin{pmatrix} R_k^T \\ \varphi_k e_k^T \end{pmatrix} t_k \right\| \qquad t_k = R_k y_k, \quad q_k = (\bar\beta_{k+1}/(R_k)_{k,k}) e_k = \varphi_k e_k$$

$$= \min_{t_k} \left\| \begin{pmatrix} z_k \\ \bar\zeta_{k+1} \end{pmatrix} - \begin{pmatrix} \bar R_k \\ 0 \end{pmatrix} t_k \right\| \qquad \bar Q_{k+1} \begin{pmatrix} R_k^T & \bar\beta_1 e_1 \\ \varphi_k e_k^T & 0 \end{pmatrix} = \begin{pmatrix} \bar R_k & z_k \\ 0 & \tilde\zeta_{k+1} \end{pmatrix}$$

Things to note

$x_k = V_k y_k, \quad t_k = R_k y_k, \quad z_k = \bar R_k t_k,$ two cheap QRs

Least squares subproblem (2)

Remember $x_k = V_k y_k, \quad t_k = R_k y_k, \quad z_k = \bar{R}_k t_k$
R_k and $\bar{R}_k$ both upper-bidiagonal

Least squares subproblem (2)

Remember $x_k = V_k y_k, \quad t_k = R_k y_k, \quad z_k = \bar{R}_k t_k$
R_k and $\bar{R}_k$ both upper-bidiagonal

Key steps to compute x_k

$$x_k = V_k y_k$$

Least squares subproblem (2)

Remember $x_k = V_k y_k, \quad t_k = R_k y_k, \quad z_k = \bar{R}_k t_k$
R_k and $\bar{R}_k$ both upper-bidiagonal

Key steps to compute x_k

$$\begin{aligned} x_k &= V_k y_k \\ &= W_k t_k \end{aligned} \qquad R_k^T W_k^T = V_k^T$$

Least squares subproblem (2)

Remember $x_k = V_k y_k, \quad t_k = R_k y_k, \quad z_k = \bar{R}_k t_k$
R_k and $\bar{R}_k$ both upper-bidiagonal

Key steps to compute x_k

$$\begin{aligned} x_k &= V_k y_k \\ &= W_k t_k & R_k^T W_k^T &= V_k^T \\ &= \bar{W}_k z_k & \bar{R}_k^T \bar{W}_k^T &= W_k^T \end{aligned}$$

Least squares subproblem (2)

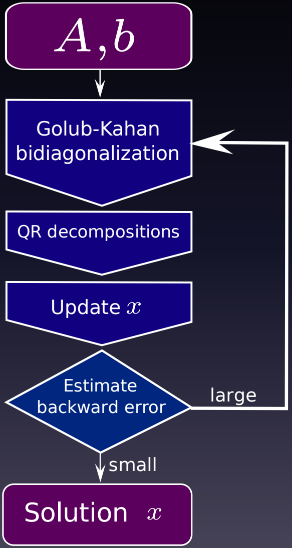
Remember $x_k = V_k y_k, \quad t_k = R_k y_k, \quad z_k = \bar{R}_k t_k$
R_k and $\bar{R}_k$ both upper-bidiagonal

Key steps to compute x_k

$$
\begin{aligned}
x_k &= V_k y_k \\
&= W_k t_k & R_k^T W_k^T &= V_k^T \\
&= \bar{W}_k z_k & \bar{R}_k^T \bar{W}_k^T &= W_k^T \\
&= x_{k-1} + \zeta_k \bar{w}_k
\end{aligned}
$$

where $z_k = \begin{pmatrix} \zeta_1 & \zeta_2 & \cdots & \zeta_k \end{pmatrix}^T$

Flow chart of LSMR



Flow chart of LSMR

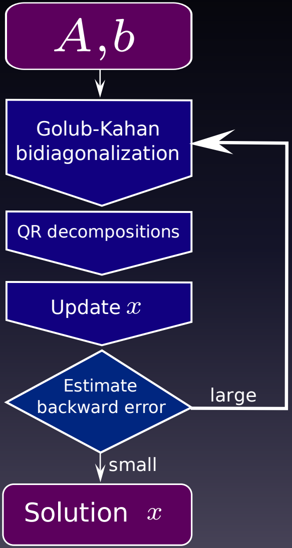
Computational cost

$Av, A^Tv, 3m + 3n$

$O(1)$

$3n$

$O(1)$



Computational and storage requirement

	Storage		Work	
	m	n	m	n
MINRES on $A^TAx = A^Tb$	Av_1	x, v_1, v_2, w_1, w_2		8
LSQR	Av, u	x, v, w	3	5
LSMR	Av, u	$x, v, h, \bar{h}$	3	6

where h_k, $\bar{h}_k$ are scalar multiples of w_k, $\bar{w}_k$

Numerical experiments

Test Data

- University of Florida Sparse Matrix Collection
- `LPnetlib`: Linear Programming Problems
- `A = (Problem.A)'`   `b = Problem.c`   (127 problems)

Numerical experiments

Test Data

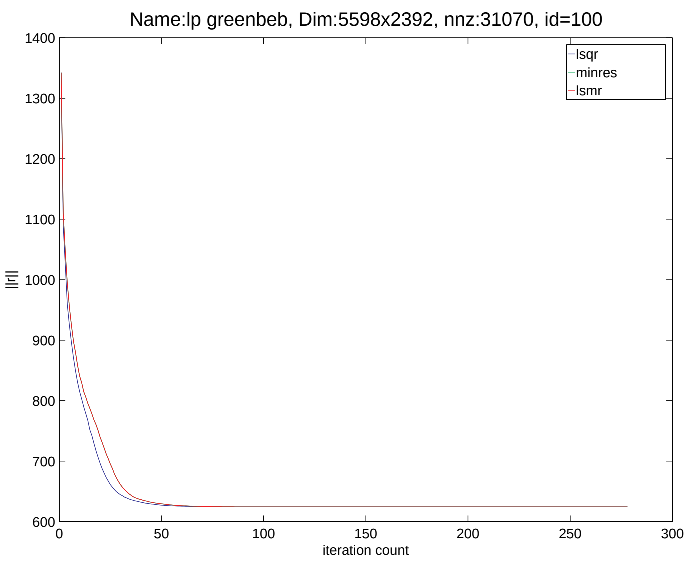
- University of Florida Sparse Matrix Collection
- `LPnetlib`: Linear Programming Problems
- `A = (Problem.A)'`   `b = Problem.c`   (127 problems)

Solve $\min \|Ax - b\|_2$

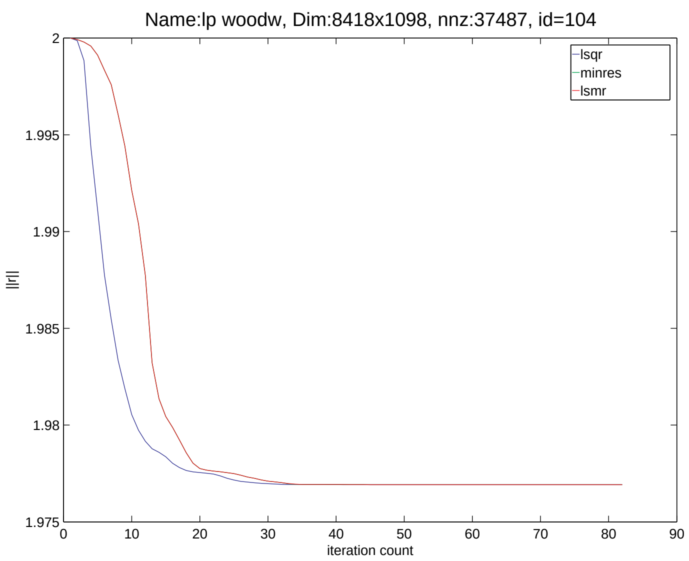
with LSQR and LSMR

- Examples of $\|r_k\|$
- Backward error tests: $\mathrm{nnz}(A) \leq 63220$
- Reorthogonalization: $\mathrm{nnz}(A) \leq 15977$

$\|r_k\|$ for LSQR and LSMR – typical



$\|r_k\|$ for LSQR and LSMR – rare



Backward error – estimates

$$(A + E_i)^T (A + E_i) x = (A + E_i)^T b$$

Backward error – estimates

$$(A + E_i)^T (A + E_i)x = (A + E_i)^T b$$

Two estimates given by Stewart (1975 and 1977)

$$E_1 = \frac{ex^T}{\|x\|^2} \qquad \|E_1\| = \frac{\|e\|}{\|x\|} \qquad e = \hat{r} - r$$

$$E_2 = -\frac{rr^T A}{\|r\|^2} \qquad \|E_2\| = \frac{\|A^T r\|}{\|r\|}$$

where $\hat{r}$ is the residual for the exact solution

Backward error – estimates

$$(A + E_i)^T (A + E_i) x = (A + E_i)^T b$$

Two estimates given by Stewart (1975 and 1977)

$$E_1 = \frac{ex^T}{\|x\|^2} \qquad \|E_1\| = \frac{\|e\|}{\|x\|} \qquad e = \hat{r} - r$$

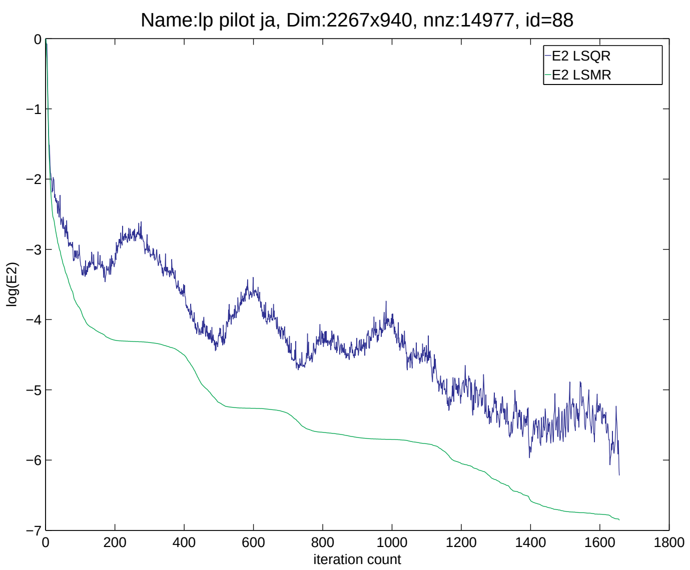
$$E_2 = -\frac{rr^T A}{\|r\|^2} \qquad \|E_2\| = \frac{\|A^T r\|}{\|r\|}$$

where $\hat{r}$ is the residual for the exact solution

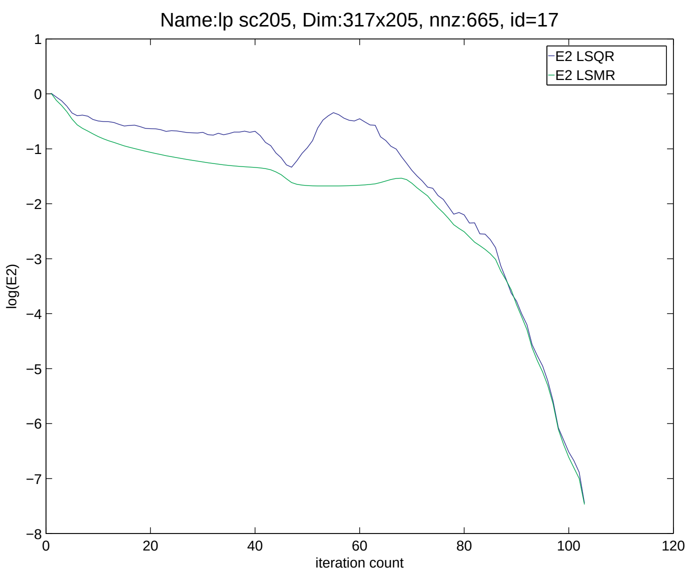
Note

$\|E_2\|$ is computable

$\log_{10}\|E_2\|$ for LSQR and LSMR – typical



$\log_{10} \|E_2\|$ for LSQR and LSMR – rare



Backward error - optimal

$$\mu(x) \equiv \min_E \|E\| \quad \text{st} \quad (A+E)^T(A+E)x = (A+E)^T b$$

Exact $\mu(x)$ (Waldén, Karlson, & Sun 1995, Higham 2002)

$$C \equiv \begin{bmatrix} A & \frac{\|r\|}{\|x\|}\left(I - \frac{rr^T}{\|r\|^2}\right) \end{bmatrix} \qquad \mu(x) = \sigma_{\min}(C)$$

Backward error - optimal

$$\mu(x) \equiv \min_E \|E\| \quad \text{st} \quad (A+E)^T(A+E)x = (A+E)^T b$$

Cheaper estimate $\tilde{\mu}(x)$ (Grcar, Saunders, & Su 2007)

$$K = \begin{pmatrix} A \\ \frac{\|r\|}{\|x\|} I \end{pmatrix} \qquad v = \begin{pmatrix} r \\ 0 \end{pmatrix}$$

$$\min_y \|Ky - v\| \qquad \tilde{\mu}(x) = \frac{\|Ky\|}{\|x\|}$$

Backward error - optimal

$$\mu(x) \equiv \min_E \|E\| \quad \text{st} \quad (A+E)^T(A+E)x = (A+E)^T b$$

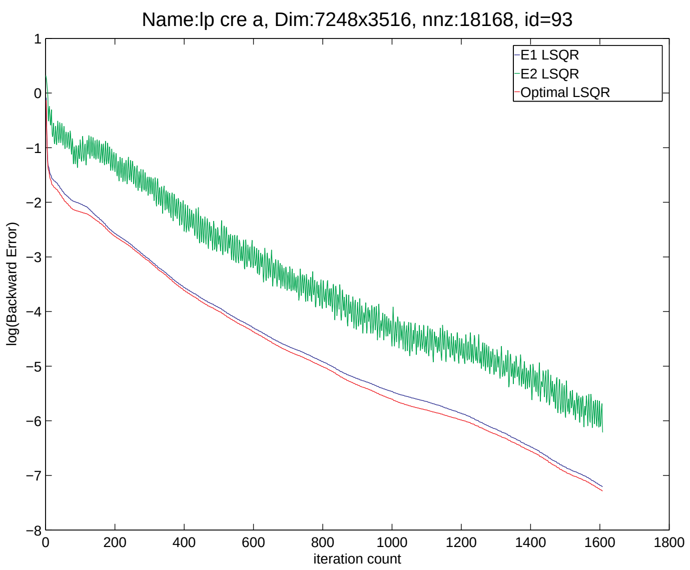
Cheaper estimate $\tilde{\mu}(x)$ (Grcar, Saunders, & Su 2007)

$$K = \begin{pmatrix} A \\ \frac{\|r\|}{\|x\|} I \end{pmatrix} \qquad v = \begin{pmatrix} r \\ 0 \end{pmatrix}$$

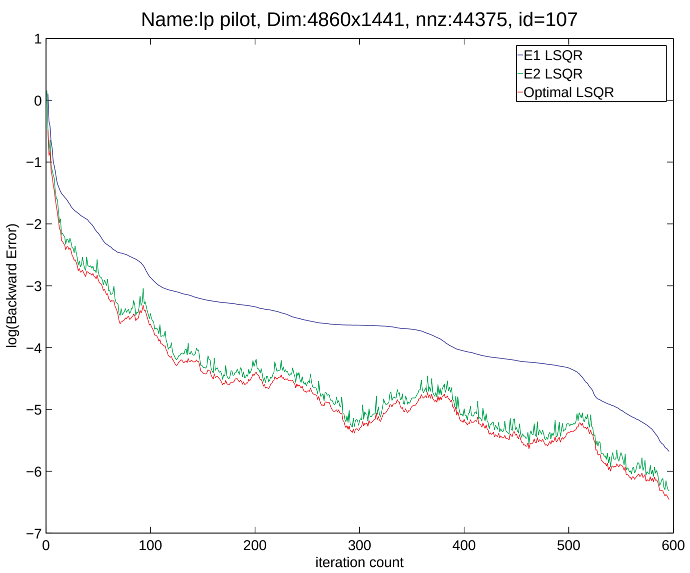
$$\min_y \|Ky - v\| \qquad \tilde{\mu}(x) = \frac{\|Ky\|}{\|x\|}$$

```
r   = b - A*x;
p   = colamd(A);
eta = norm(r)/norm(x);
K   = [A(:,p); eta*speye(n)];
v   = [r; zeros(n,1)];
[c,R]   = qr(K,v,0);
mutilde = norm(c)/norm(x);
```

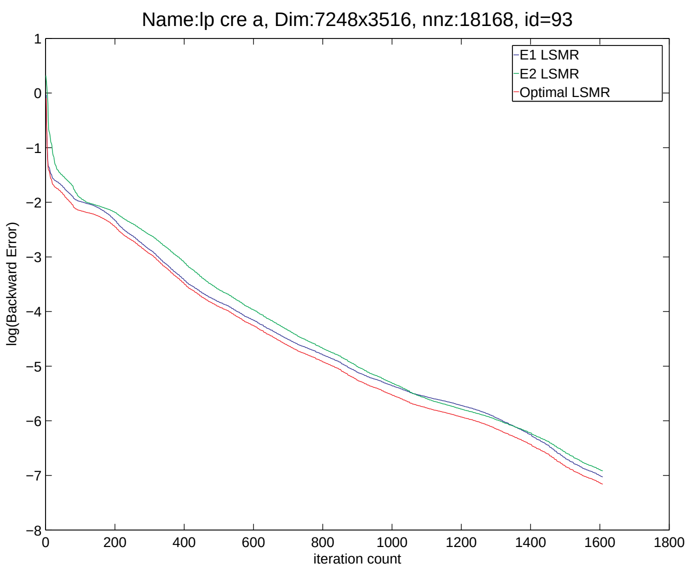
Backward errors for LSQR – typical



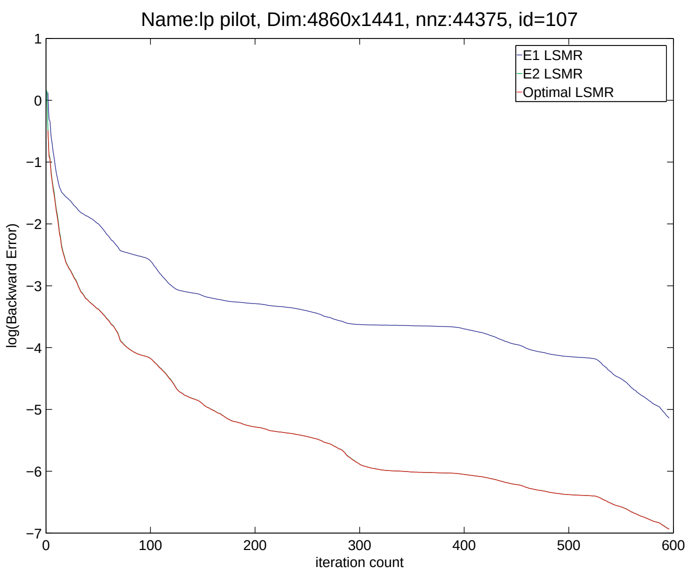
Backward errors for LSQR – rare



Backward errors for LSMR – typical



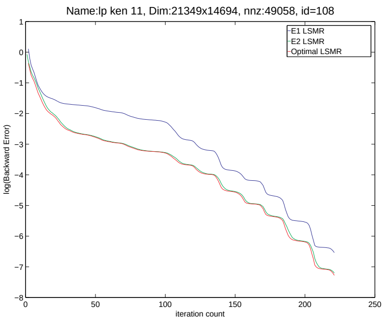
Backward errors for LSMR – rare



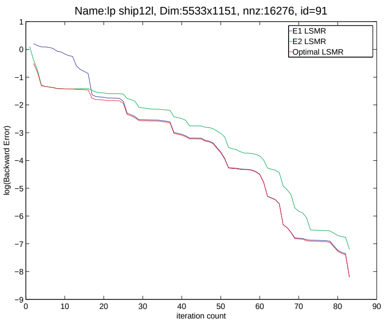
For LSMR

$\|E_2\| \approx$ optimal BE almost always

Typical: $\|E_2\| \approx \tilde{\mu}(x)$



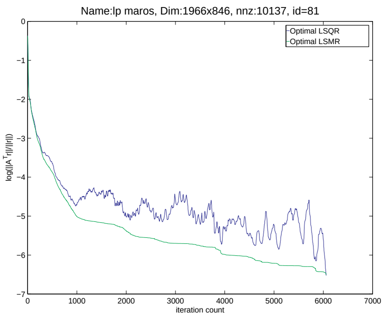
Rare: $\|E_1\| \approx \tilde{\mu}(x)$



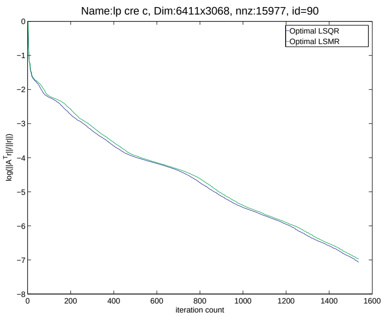
For LSMR, optimal BE $\tilde{\mu}(x)$ seems to be monotonic

For LSQR, usually not

Typical for LSQR and LSMR



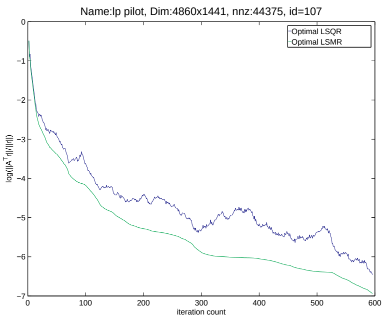
Rare LSQR, typical LSMR



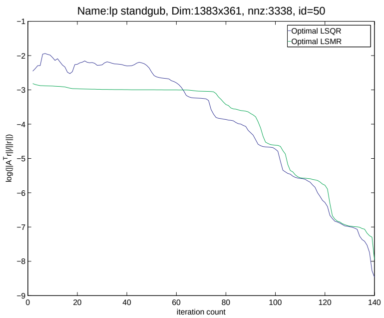
Optimal backward errors

$\tilde{\mu}(x^{\mathrm{LSMR}}) \leq \tilde{\mu}(x^{\mathrm{LSQR}})$ almost always

Typical



Rare



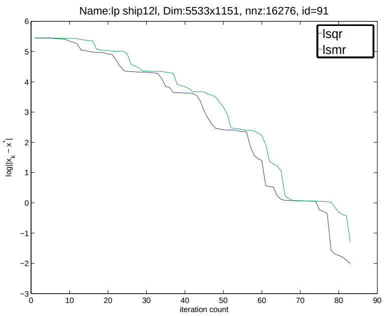
Errors

- $\|x^{\mathrm{LSQR}} - x^*\|$ is monotonic
- $\|x^{\mathrm{LSMR}} - x^*\|$ seems to be monotonic
- $\|x^{\mathrm{LSQR}} - x^*\| \leq \|x^{\mathrm{LSMR}} - x^*\|$

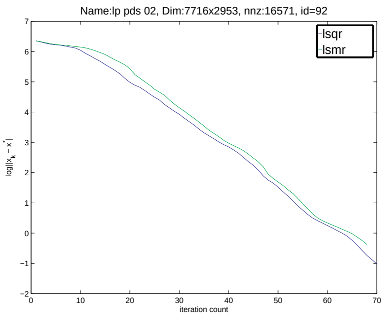
Errors

- $\|x^{\text{LSQR}} - x^*\|$ is monotonic
- $\|x^{\text{LSMR}} - x^*\|$ seems to be monotonic
- $\|x^{\text{LSQR}} - x^*\| \le \|x^{\text{LSMR}} - x^*\|$

LSQR error $\le$ LSMR error



Both give min-length x



Space-time trade-offs

LSMR is well-suited for limited memory computations.

What if we have

- more memory
- Av expensive

Can we speed things up?

Space-time trade-offs

LSMR is well-suited for limited memory computations.

What if we have

- more memory
- Av expensive

Can we speed things up?

Some ideas:

- Reorthogonalization
- Restarting
- Local reorthogonalization

Reorthogonalization

Golub-Kahan process

Infinite precision	Finite precision
U_k, V_k orthonormal	Lose orthogonality
At most $\min(m, n)$ iterations	Could take $10n$ or more

Reorthogonalization

Golub-Kahan process

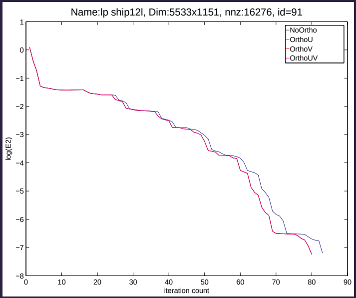
Infinite precision	Finite precision
U_k, V_k orthonormal	Lose orthogonality
At most $\min(m, n)$ iterations	Could take $10n$ or more

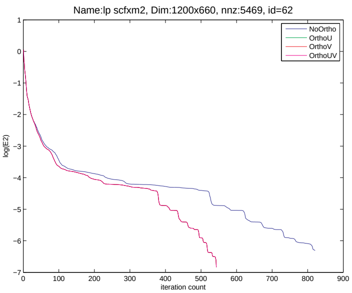
Apply modified Gram-Schmidt to u_{k+1} and/or v_{k+1}:

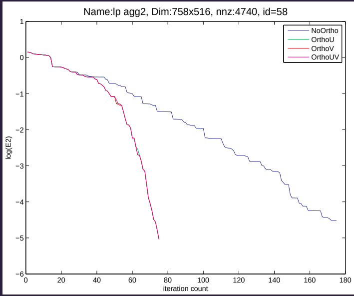
$$u \leftarrow u - (u_j^T u)u_j \qquad j = k,\ k-1,\ k-2,\ \ldots$$

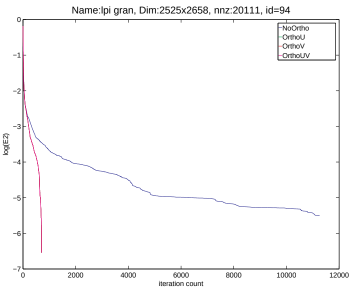
(similarly for v)

Effects of reorthogonalization on various problems

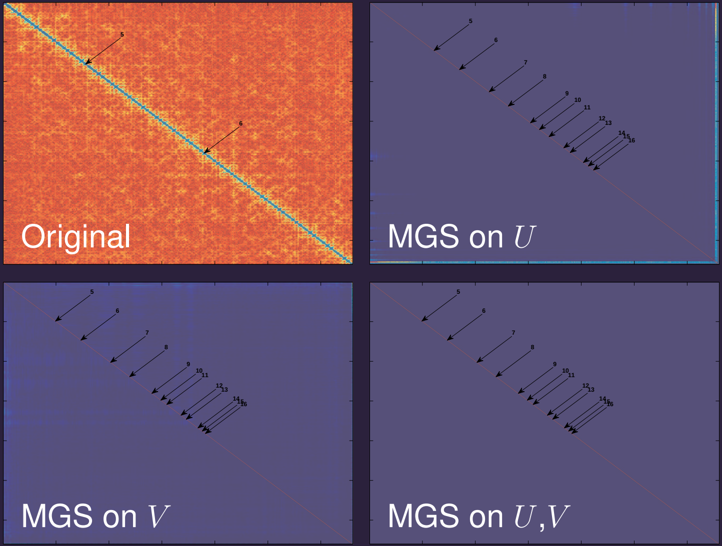




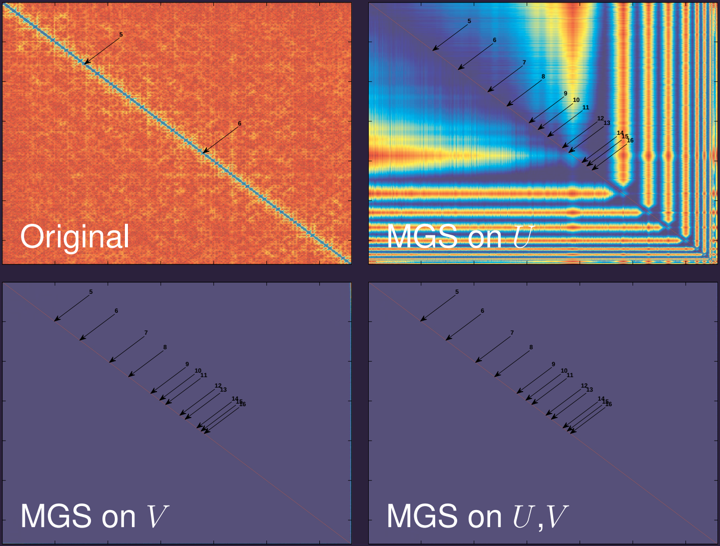




Orthogonality of U_k



Orthogonality of V_k



What we learnt so far

- Reorthogonalizing V_k (only) is sufficient
- Reorthogonalizing U_k (only) is nearly as good
- x_k converges the same for all options

What can be improved

- May still use too much memory
- Need more flexibility for space-time trade-off

Reorthogonalization with Restarting

Restarting LSMR

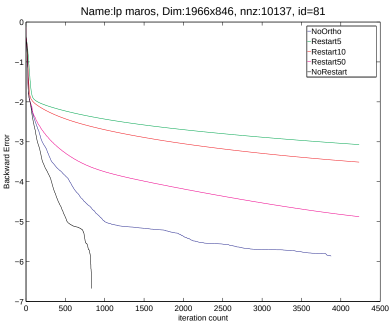
$$r_k = b - Ax_k \qquad \min \|A\Delta x - r_k\|$$

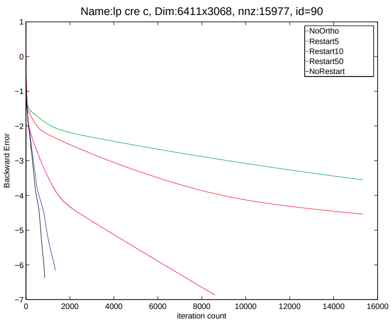
Reorthogonalization with Restarting

Restarting LSMR

$$r_k = b - Ax_k \qquad \min \|A\Delta x - r_k\|$$

Restarting leads to stagnation





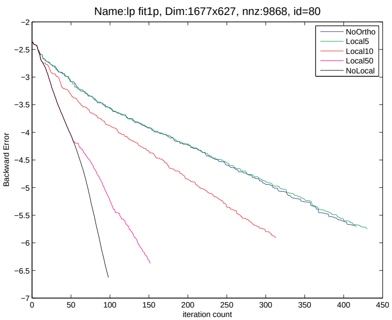
Local reorthogonalization

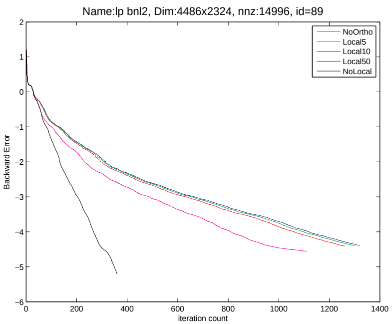
- Reorthogonalize wrto only the last l vectors
- Partial speed-up
- Less memory
- Depends on efficiency of Av and $A^T u$

Local reorthogonalization

- Reorthogonalize wrto only the last l vectors
- Partial speed-up
- Less memory
- Depends on efficiency of Av and $A^T u$

Speed up with local reorthogonalization





Conclusions

LSMR has the good properties of LSQR and more

- $\|r_k\|$ seems monotonic (nearly as small as for LSQR)
- $\|x_k - x^*\|$ also
- $\|A^T r_k\|$ is monotonic

Conclusions

LSMR has the good properties of LSQR and more

- $\|r_k\|$ seems monotonic (nearly as small as for LSQR)
- $\|x_k - x^*\|$ also
- $\|A^T r_k\|$ is monotonic

Stewart backward errors $\quad \|E_1\| = \frac{\|\hat{r} - r_k\|}{\|x_k\|} \quad \|E_2\| = \frac{\|A^T r_k\|}{\|r_k\|}$

- $\|E_1\|$ monotonic if $\|r_k\|$ monotonic (theorem)

Conclusions

LSMR has the good properties of LSQR and more

- $\|r_k\|$ seems monotonic (nearly as small as for LSQR)
- $\|x_k - x^*\|$ also
- $\|A^T r_k\|$ is monotonic

Stewart backward errors $\|E_1\| = \frac{\|\hat{r} - r_k\|}{\|x_k\|}$ $\|E_2\| = \frac{\|A^T r_k\|}{\|r_k\|}$

- $\|E_1\|$ monotonic if $\|r_k\|$ monotonic (theorem)
- $\|E_2\|$ usually monotonic (1 exception in 127 cases)

Conclusions

LSMR has the good properties of LSQR and more

- $\|r_k\|$ seems monotonic (nearly as small as for LSQR)
- $\|x_k - x^*\|$ also
- $\|A^T r_k\|$ is monotonic

Stewart backward errors $\|E_1\| = \frac{\|\hat{r} - r_k\|}{\|x_k\|}$ $\|E_2\| = \frac{\|A^T r_k\|}{\|r_k\|}$

- $\|E_1\|$ monotonic if $\|r_k\|$ monotonic (theorem)
- $\|E_2\|$ usually monotonic (1 exception in 127 cases)
- $\|E_2\| \approx$ optimal backward error (seems monotonic)

Conclusions

LSMR has the good properties of LSQR and more

- $\|r_k\|$ seems monotonic (nearly as small as for LSQR)
- $\|x_k - x^*\|$ also
- $\|A^T r_k\|$ is monotonic

Stewart backward errors $\quad \|E_1\| = \frac{\|\hat{r} - r_k\|}{\|x_k\|} \quad \|E_2\| = \frac{\|A^T r_k\|}{\|r_k\|}$

- $\|E_1\|$ monotonic if $\|r_k\|$ monotonic (theorem)
- $\|E_2\|$ usually monotonic (1 exception in 127 cases)
- $\|E_2\| \approx$ optimal backward error (seems monotonic)
- $\Rightarrow$ reliable rule for stopping early

Acknowledgements

- David Fong
- Chris Paige
- Stanford Graduate Fellowship
- ONR and AHPCRC

Acknowledgements

- David Fong
- Chris Paige
- Stanford Graduate Fellowship
- ONR and AHPCRC

- First presented at 2010 Copper Mountain Conference
- Jon Claerbout (Stanford Geophysics)
  Nov 2009: Computational success $\equiv 1 - \|A^T r_k\| / \|A^T b\|$

Paper and Implementations

http://www.stanford.edu/group/SOL/software.html
Report SOL 2010-2 submitted to SISC
Matlab and F90 code