Block Krylov Subspace Methods for Solving Large Sparse SVD and Eigenvalue Problems

James Baglama
Department of Mathematics
University of Rhode Island
Kingston, RI 02881
jbaglama@math.uri.edu
http://www.math.uri.edu/∼jbaglama

Lothar Reichel
Department of Mathematical Sciences
Kent State University
Kent, OH 44242
reichel@math.kent.edu
http://www.math.kent.edu/∼reichel

Block Krylov Methods

- A block routine typically requires more computational effort and larger subspaces for acceptable approximations.
- Why block?
  - A block routine can compute multiple or clustered eigenvalues more efficiently than an unblocked routine
  - A block routine can utilize Level 3 BLAS matrix-matrix products
  - Fewer out-of-core memory calls when A cannot be store explicitly

Singular Value Problem

Computing singular triples (σ, u, v) of A.

– Partial singular value decomposition

$$AV = U\Sigma$$

$$U \in \mathbf{R}^{l \times k} \quad U^T U = I \qquad V \in \mathbf{R}^{n \times k} \quad V^T V = I$$

$$\Sigma = \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_k \end{pmatrix} \in \mathbf{R}^{k \times k}$$

– Few of the smallest or largest singular values.

Eigenvalue Problem

Computing eigenpairs (λ, x) of $A \in \mathbf{R}^{n \times n}$.

– Partial eigenvalue decomposition

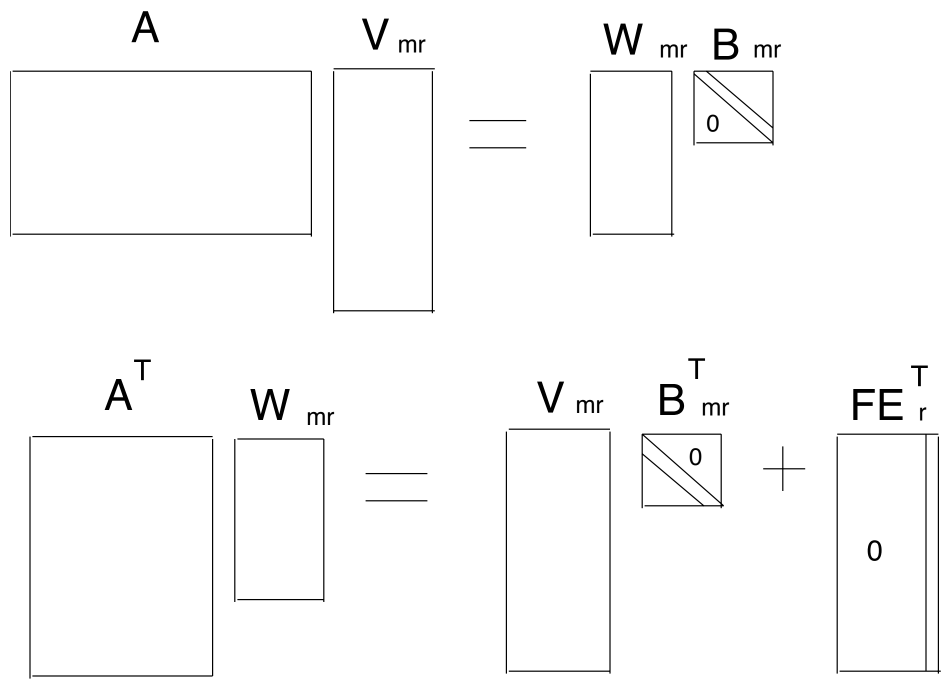
$$AX = XD \qquad X \in \mathbf{C}^{n \times k}$$

$$D = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_k \end{pmatrix} \in \mathbf{C}^{k \times k}$$

– Few of the smallest or largest in magnitude, smallest or largest real part, or smallest or largest imaginary part.

Block Lanczos Bidiagonalization Method

$$
\begin{aligned}
AV_{mr} &= W_{mr}B_{mr} \\
A^TW_{mr} &= V_{mr}B_{mr}^T + FE_r^T
\end{aligned}
\qquad \text{where} \qquad V_{mr}^TV_{mr} = W_{mr}^TW_{mr} = I_{mr},\;\; V_{mr}^TF = 0
$$



Computation of Singular Values

Block Lanczos bidiagonal method:

$$\begin{aligned} AV_{mr} &= W_{mr}B_{mr} \\ A^TW_{mr} &= V_{mr}B_{mr}^T + FE_r^T \end{aligned}$$

Compute the singular values and vectors of B_{mr}:

$\{\sigma_i\}_{i=1}^{mr} \qquad \{v_i\}_{i=1}^{mr} \qquad \{u_i\}_{i=1}^{mr}$ such that $\begin{cases} B_{mr}v_i = \sigma_i u_i \\ B_{mr}^T u_i = \sigma_i v_i \end{cases}$

Let $\tilde{u}_i = W_{mr}u_i \qquad \tilde{v}_i = V_{mr}v_i$

$\|A^T\tilde{u}_i - \sigma_i\tilde{v}_i\| = \|A^TW_{mr}u_i - \sigma_i V_{mr}v_i\| = \|FE_r^Tu_i\|$

When $\|FE_r^Tu_i\| \approx 0$

$\sigma_i \approx$ singular value of A and u_i & $v_i \approx$ singular vectors of A such that
$Av_i = \sigma_i u_i$ and $A^Tu_i = \sigma_i v_i$

Restarted Method for computing singular values

Basic Algorithm

1. Compute: Block Lanczos bidiagonal method for modest value of mr

$$\begin{aligned} AV_{mr} &= W_{mr}B_{mr} \\ A^TW_{mr} &= V_{mr}B_{mr}^T + FE_r^T \end{aligned}$$

2. Compute desired singular triples of B_{mr} and check convergence $\|FE_r^T u_i\| < \text{Tol}$
3. Modify starting vectors V_{1r} and restart

D. Sorensen's (1992) very successful implicitly restarted method (IRL) applied to Lanczos Bidiagonalization method

- (1994) Bjorck et al. Implicitly shifted Bidiagonalization method
- (2004) Kokiopoulou et al. Implicitly Restarted Bidiagonalization method
- (2005) Baglama and Reichel **Augmented Implicitly Restarted Lanczos Bidiagonalization Methods** ⟵ Not IRL!

Augmented Block Lanczos (Ritz) Bidiagonal Method

Block Lanczos bidiagonal method

$$\begin{aligned} AV_{mr} &= W_{mr}B_{mr} \\ A^TW_{mr} &= V_{mr}B_{mr}^T + FE_r^T \end{aligned}$$

Block Lanczos tridiagonal method

$$A^TAV_{mr} = V_{mr}B_{mr}^TB_{mr} + S^{(m)}FE_r^T$$

$$\begin{aligned} A[\tilde{v}_1, \ldots, \tilde{v}_k, \tilde{V}_{\ell r}] &= [\tilde{u}_1, \ldots, \tilde{u}_k, \tilde{W}_{\ell r}]\tilde{B}_{k+\ell r} \\ A^T[\tilde{u}_1, \ldots, \tilde{u}_k, \tilde{W}_{\ell r}] &= [\tilde{v}_1, \ldots, \tilde{v}_k, \tilde{V}_{\ell r}]\tilde{B}_{k+\ell r}^T + \tilde{F}E_r^T \end{aligned}$$

$$\tilde{B}_{k+\ell r} = \begin{pmatrix} \tilde{\sigma}_1 & & & \tilde{r}_1^T & & & 0 \\ & \ddots & & \vdots & & & \\ & & \tilde{\sigma}_k & \tilde{r}_k^T & & & \\ & & & S^{(m+1)} & L^{(m+2)} & & \\ & & & & \ddots & & \\ & & & & & \ddots & L^{(m+\ell)} \\ 0 & & & & & & S^{(m+\ell)} \end{pmatrix}$$

$[\underbrace{\tilde{v}_1, \ldots, \tilde{v}_k}\quad F]$ $\approx$ right singular vectors of A or Ritz of A^TA

$[\underbrace{\tilde{u}_1, \ldots, \tilde{u}_k}\quad AF]$ $\approx$ left singular vectors of A

Augmented Block Lanczos (Harmonic) Bidiagonal Method

$$\begin{aligned} A[\tilde{y}_1, \ldots, \tilde{y}_k, \tilde{V}_{\ell r}] &= [\tilde{u}_1, \ldots, \tilde{u}_k, \tilde{W}_{\ell r}]\tilde{B}_{k+\ell r} \\ A^T[\tilde{u}_1, \ldots, \tilde{u}_k, \tilde{W}_{\ell r}] &= [\tilde{y}_1, \ldots, \tilde{y}_k, \tilde{V}_{\ell r}]\tilde{B}_{k+\ell r}^T + \tilde{F}E_r^T \end{aligned}$$

$$\tilde{B}_{k+\ell r} = \begin{pmatrix} \begin{pmatrix} \sigma'_1 & & & 0 & \\ & \sigma'_2 & & & \hat{C}^{(m+1)} \\ & & \ddots & & \\ & & & \sigma'_k & \\ 0 & & & & \hat{S}^{(m+1)} \end{pmatrix} (R'_{k+r})^{-1} & & & 0 \\ & S^{(m+2)} & L^{(m+3)} & \\ & & \ddots & \\ & & \ddots & L^{(m+\ell)} \\ 0 & & & S^{(m+\ell)} \end{pmatrix}$$

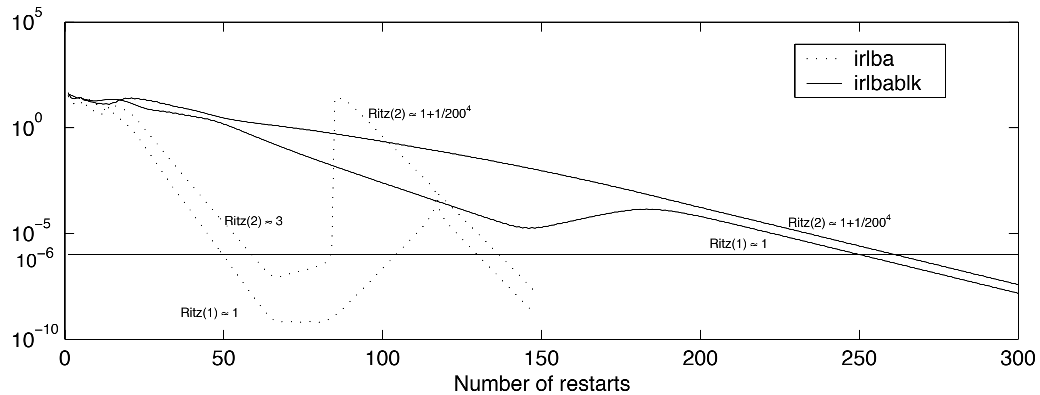
$$\begin{pmatrix} B_{mr}^{-1}U'_k\Sigma'_k & -B_{mr}^{-1}E_rL^{(m+1)} \\ 0 & I_r \end{pmatrix} = Q'_{k+r}R'_{k+r}$$

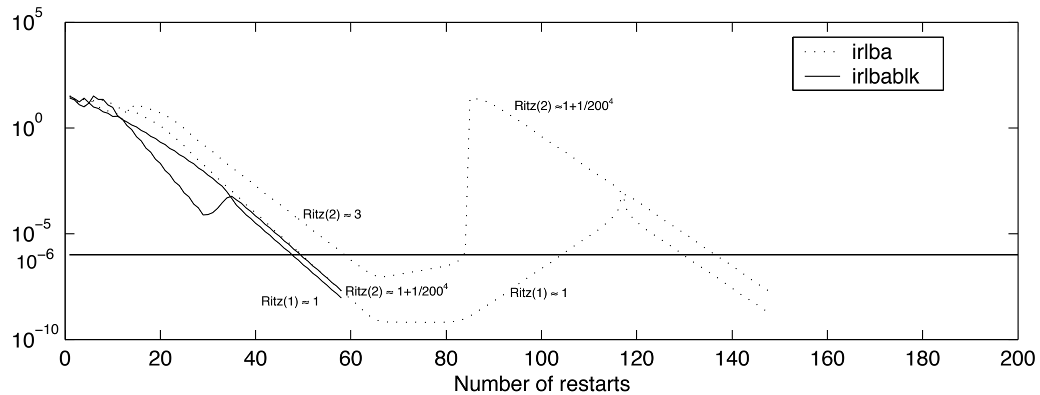
$[\underbrace{\tilde{y}_1, \ldots, \tilde{y}_k}]$ $\approx$ right singular vectors of A

$[\underbrace{\tilde{u}_1, \ldots, \tilde{u}_k}]$ $\approx$ left singular vectors of A

Numerical Example 1

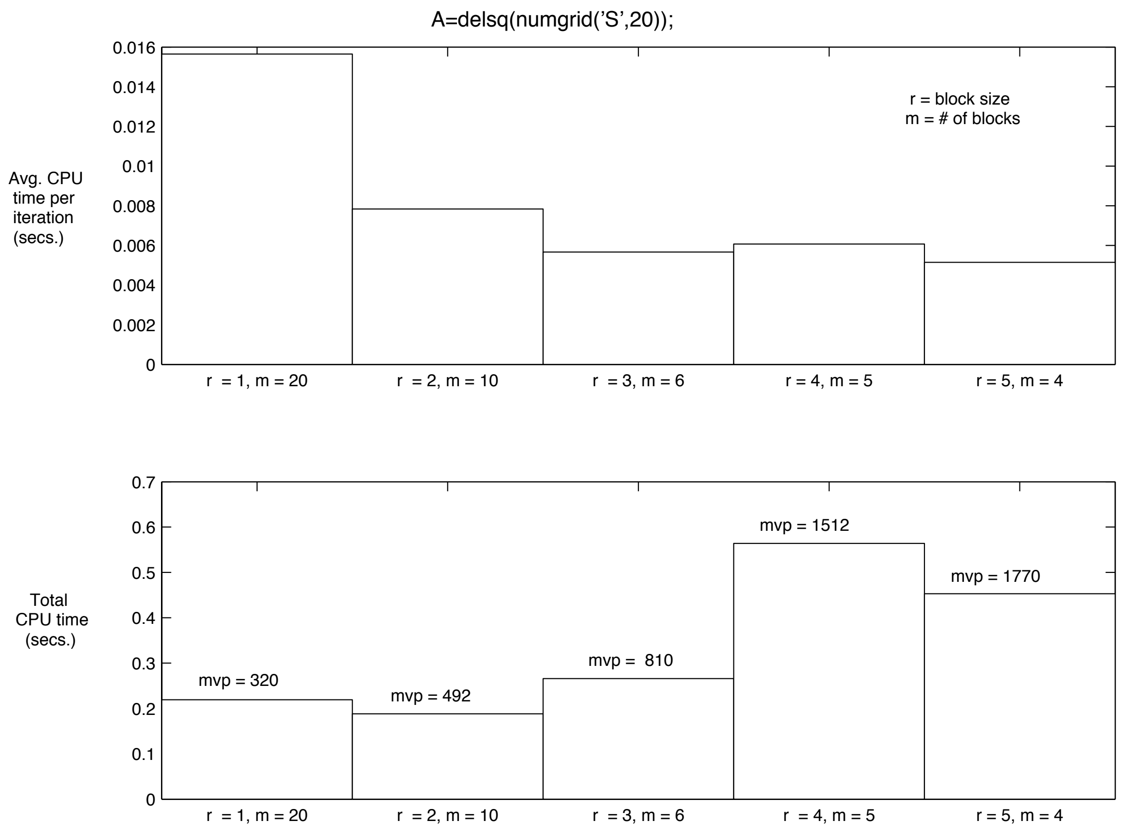
Approximations to singular values of $A := \mathrm{diag}[1, 1+\frac{1}{200^4}, 3, \ldots, 200]$. For the top graph both methods **irlbablk** and **irlba** are using a maximum of 20 Krylov vectors and in the bottom graph the block method **irlbablk** is using a maximum of 40 vectors while the non-block method **irlba** is using a maximum of 20 vectors.





Numerical Example 2

Approximations to 6 largest singular values of $A = delsq(numgrid('S', 20));$. The top graph shows the average CPU time per iteration and bottom graph is the total CPU time and matrix-vector products needed to calculate all 6 singular values within the tolerance of machine epsilon.



Paper and MATLAB Codes for SVD

- J. Baglama and L. Reichel, Restarted Block Lanczos Bidiagonalization Methods, Numerical Algorithms, 43 (2006), pp. 251-272.

- MATLAB codes irlba.m and irlbablk.m
  - My website: http://www.math.uri.edu/∼jbaglama/
  - http://www.netlib.org/numeralgo/na26
    - ∗ MATLAB GUI interface and demo
    - ∗ 7 page tutorial
    - ∗ codes irlba.m and irlbablk.m

Eigenvalue Problem

Computing eigenpairs (λ, x) of $A \in \mathbf{R}^{n\times n}$.

- Partial eigenvalue decomposition

$$AX = XD \qquad X \in \mathbf{C}^{n\times k}$$

$$D = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_k \end{pmatrix} \in \mathbf{C}^{k\times k}$$

Some competitive public domain software for solving large sparse eigenvalue problems

- ARNCHEB , ARPACK , BLZPACK, EA16[1], EB12[1], EB13[1], LASO2, LOPSI, SRRIT, (FORTRAN)
- TRLan (FORTRAN 90)
- `eigifp`, `bleigifp`, `irbleigs`, `jdqr`, `jdzr`, `lobpcg` (MATLAB)

[1] Part of the HSL 2002 collection

Motivation for Augmented Block Householder Arnoldi Method

D. Sorensen's (1992) very successful implicitly restarted method (IRA) applied to block Arnoldi.

(1997) Lehoucq and Maschhoff, bIRAM method
(currently ARPACK does not have a block routine)

- "Exact shifts". There is a one-to-one correspondence between the reduction of the Arnoldi decomposition and the number of shifts applied. In the block form, the ratio becomes one-to-block size causing a significant number of unwanted Ritz values not being applied as shifts. Pick shifts to apply.
  - Augment the Block Krylov subspace with desired vectors

- Orthogonality of Ritz vectors and the case of singular or nearly singular subdiagonal blocks of the block Hesssenberg matrix. Modified Gram-Schmidt (MGS) process with partial or full reorthogonalization. Singular blocks require special care.
  - Block Householder Arnoldi

Remarks on Augmented Block Householder Arnoldi Method

- Single vector case: D. Sorensen's IRA method is mathematically equivalent to augmenting the Krylov subspaces, Morgan (1996).
- Block case: bIRAM method is not mathematically equivalent to augmenting the block Krylov subspaces, Möller (2004).
- Morgan (2005) presents an augmented Arnoldi block routine to solve linear systems of equations
- Möller (2004) presents an augmented Arnoldi block routine for solving nonsymmetric eigenvalue problems.

- Augmented Block Householder Arnoldi Method
  - Uses compact WY representation of the Householder product.
  - Computes a partial Real Schur decomposition, no complex arithmetic, (reordering is required)
  - Updating the new block Arnoldi decomposition cost is the same as bIRAM.

Block Arnoldi Householder Algorithm

1. Set $X := [x_1, \ldots, x_r] \in \mathbf{R}^{n\times r}$;
2. for $j = 0, 1, \ldots, m$
3. Compute the Householder QR-decomposition where $X(jr+1:n, 1:r) = QR$ and $Q = (I + WSW^T)\begin{bmatrix} I \\ 0 \end{bmatrix}$
4. if $j = 0$

   $a.)\ Set \begin{cases} Y & := W \\ T & := S \\ H_{(1,0)} & := R \end{cases}$

   else

   $$b.)\ Set \begin{cases} \begin{bmatrix} H_{(1,j)} \\ \vdots \\ H_{(j,j)} \end{bmatrix} := \begin{bmatrix} X(1:jr, 1:r) \end{bmatrix} \\ \begin{bmatrix} H_{(j+1,j)} \end{bmatrix} \quad \begin{bmatrix} R \end{bmatrix} \end{cases}$$

   $$c.)\ Set \begin{cases} W := \begin{bmatrix} 0 \\ W \end{bmatrix} \\ T := \begin{bmatrix} T & TY^TWS \\ 0 & S \end{bmatrix} \\ Y := \begin{bmatrix} Y & W \end{bmatrix} \end{cases}$$

   end
5. if $j < m$
6. Compute $X := (I + YT^TY^T)A(I + YTY^T)\begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix}$
7. end
8. end

Arnoldi Vectors

$$V_{mr+r} = [V_{(1)}, V_{(2)}, \ldots, V_{(m+1)}] = (I + YTY^T)I_{mr+r}$$

QR decomposition of Krylov matrix

$$(I + YT^TY^T)\ [X, AV_{(1)}, \ldots, AV_{(m)}] = \begin{bmatrix} H_{(1,0)} & \begin{bmatrix} H_{(1,1)} \\ H_{(2,1)} \end{bmatrix} & \cdots & \begin{bmatrix} H_{(1,m)} \\ \vdots \\ H_{(m+1,m)} \end{bmatrix} \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

Block Hessenberg

$$H_{mr+r} = \begin{bmatrix} H_{(1,1)} & & \cdots & & H_{(1,m)} \\ H_{(2,1)} & H_{(2,2)} & & & \\ & & & \ddots & \vdots \\ & & \ddots & \ddots & \\ & & & H_{(m,m-1)} & H_{(m,m)} \\ 0 & & & & H_{(m+1,m)} \end{bmatrix}$$

Arnoldi Decomposition
$AV_{mr} = V_{mr+r}H_{mr+r}$

Augmented Block Arnoldi Householder Algorithm

1. Block Householder Arnoldi decomposition $AV_{mr} = V_{mr+r}H_{mr+r}$
   Note: Vectors V_{mr+r} are stored as $(I + YTY^T)$
2. Compute and sort the Real Schur decomposition $H_{mr}Q_k = Q_kT_k$
3. Check convergence of the k eigenvalues
4. Compute Schur (starting) vectors and residual vectors
   $[X_k \;\; V_{mr:mr+r}] = (I + YTY^T)[Q_k \;\; 0 \;\; I_r]$
5. Place all vectors $([X_k \;\; V_{mr:mr+r}])$ into the compact form $(I + YTY^T)$
6. Set the first k columns and $k + r$ rows of the matrix
   $$H_{k+r} = \begin{bmatrix} T_k \\ H_{(m+1,m)}E_r^T Q_k^{(H_{mr})} \end{bmatrix}$$
7. Goto 1 using $V_{mr:mr+r}$ as the new starting block.

Place all vectors $([X_k\ V_{mr:mr+r}])$ into the compact form $(I + YTY^T)$

1. Set $Y := [X_k\ \ V_{mr:mr+r}]$ and $m := k + r$
2. Set $T := Y_{1:m\times 1:m}$
3. for $i = 1 : m$
   $R_{i,i} =$sign$(Y_{i,i})$
   if $R_{i,i} = 0$ then $R_{i,i} = 1$ end
   $\alpha = 1 + R_{i,i}Y_{i,i}$
   $Y_{i,i} = Y_{i,i} + R_{i,i}$
   $D_{i,i} = 1/Y_{i,i}$
   $Y_{1:m\times i+1:m} = Y_{1:m\times i+1:m} - \frac{R_{i,i}}{\alpha} Y_{1:m\times i} Y_{i\times i+1:m}$
   end
4. $Y_{1:m\times 1:m} = \mathrm{tril}(Y_{1:m\times 1:m}D)$
5. $T = -Y^{-1}_{1:m\times 1:m}(TR + I)Y^{-T}_{1:m\times 1:m}$
6. $Y_{m+1:n\times 1:m} = -Y_{m+1:n\times 1:m}(RY^{-T}_{1:m\times 1:m}T^{-1})$

Notice when n is large the dominating computational expense occurs at step 6, which parallels the computational cost that is encountered when updating the block Arnoldi decomposition in the bIRAM.

Numerical Example 1

Let $A \in \mathbf{R}^{1600\times 1600}$ be obtained by discretizing the 2-dimensional negative Laplace operator on the unit square by the standard 5-point stencil with Dirichlet boundary conditions.

$(k = 3)$

Method	block size	# of blocks	# mvps	CPU time mvps	CPU time Total	$\|AQ_k - Q_kT_k\|_2$
ahbeigs	1	20	472	0.233s	1.53s	$O(10^{-14})$
$tol = 10^{-12}$	2	10	932	0.295s	2.39s	$O(10^{-14})$
ahbeigs	2	10	762	0.141s	1.94s	$O(10^{-12})$
$tol = 10^{-10}$	2	20	516	0.078s	1.95s	$O(10^{-12})$
ahbeigs	2	10	636	0.095s	1.78s	$O(10^{-10})$
$tol = 10^{-8}$	2	20	448	0.096s	1.78s	$O(10^{-10})$
irbleigs	2	10	1040	0.234s	2.03s	$O(10^{-8})$
eigifp	1	20	907	0.41s	1.69s	$O(10^{-13})$
jdqr	1	20	637	0.186s	1.48s	$O(10^{-13})$
eigs	1	20	585	0.31s	0.86s	$O(10^{-14})$

$(k = 100)$

Method	block size	# of blocks	# mvps	CPU time mvps	CPU time Total	$\|AQ_k - Q_kT_k\|_2$
ahbeigs	1	120	1180	0.563s	88.1s	$O(10^{-13})$
$tol = 10^{-12}$	2	60	914	0.21s	42.2s	$O(10^{-13})$
ahbeigs	2	60	860	0.248s	38.7s	$O(10^{-11})$
$tol = 10^{-10}$	2	120	896	0.203s	31.9s	$O(10^{-14})$
ahbeigs	2	60	824	0.219s	36.1s	$O(10^{-9})$
$tol = 10^{-8}$	2	120	772	0.235s	26.8s	$O(10^{-8})$
irbleigs	2	60	44066	12.45s	320.2s	$O(10^{-8})$
eigifp	1	120	22244	10.89s	150.5s	$O(10^{-13})$
jdqr	1	120	9737	3.70s	203.0s	$O(10^{-13})$
eigs	1	120	887	0.231s	10.9s	$O(10^{-13})$

Numerical Example 2

Finding the 22 eigenvalues of largest magnitude for CK656 matrix from the Non-Hermitian Eigenvalue Problem (NEP) Collection with tolerance set at 10^{-12}.

Method	block size	# of blocks	# mvps	CPU time mvps	CPU time Total	$\|AQ_k - Q_kT_k\|_2$
ahbeigs	1	72	435	0.020s	2.64s	$O(10^{-14})$
	2	36	460	0.016s	1.76s	$O(10^{-13})$
	3	24	663	0.047s	2.16s	$O(10^{-13})$
	4	18	876	0.078s	2.31s	$O(10^{-13})$
jdqr	1	72	1440	0.298s	25.3s	$O(10^{-13})$
eigs	1	72	433	0.022s	0.83s	$O(10^{-14})$
Möller(S)	1	72	408			
	2	36	456			
	4	18	840			
Möller(L)	1	72	428			
	2	36	478			
	3	24	669			
	4	18	814			

Numerical Example 3

Finding the 3 eigenvalues of largest imaginary part for the matrix TOLS2000 from the Non-Hermitian Eigenvalue Problem (NEP) Collection with tolerance set at 10^{-9}.

Method	block size	# of blocks	# mvps	CPU time mvps	CPU time Total	$\|AQ_k - Q_kT_k\|_2$
`ahbeigs`	1	30	510	0.031s	2.43s	$O(10^{-7})$
	2	30	1410	0.188s	7.76s	$O(10^{-7})$
	3	30	1854	0.203s	11.59s	$O(10^{-9})$
`eigs`	1	30	766	0.078s	1.52s	$O(10^{-7})$
`Jia`	2	30	1980			
	3	30	1800			

Numerical Example 4

Finding the 4 eigenvalues of largest real part for the generalized eigenvalue problem with matrices BFW782A and BFW782B. The matrix A is nonsymmetric and B is symmetric indefinite. The program **`eigs`** requires B to be positive definite or for the user to input a matrix-vector product routine to compute $B^{-1}Ax$.

Method	block size	# of blocks	Total CPU time	$\|AQ_k - BQ_kT_k\|_2$
`ahbeigs`	1	48	5.77s	$O(10^{-10})$
	2	24	6.97s	$O(10^{-10})$
	3	16	10.78s	$O(10^{-9})$
	4	12	14.00s	$O(10^{-9})$
`jdqz`	1	48	10.56s	$O(10^{-10})$

Paper and MATLAB Code for Eigenvalue Problem

- J. Baglama, Augmented Block Householder Arnoldi Method, Linear Algebra Appl. in press (2008)

- MATLAB code ahbeigs.m
  - My website: http://www.math.uri.edu/∼jbaglama/
  - Mathwork's file exchange http://www.mathworks.com/matlabcentral/fileexchange/

Gene Golub's references for this presentation

- G. H. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, J. SIAM Numer. Anal., Ser. B, 2 (1965), pp. 205–224.
- G. H. Golub, R. Underwood and J. H. Wilkinson, *The Lanczos algorithm for the symmetric* $Ax = \lambda Bx$ *problem*, Report STAN-CS-72-270, Department of Computer Science, Stanford University, Stanford, 1972.
- G. H. Golub, *Some modified matrix eigenvalue problems*, SIAM Review, 15 (1973), pp. 318–334.
- G. H. Golub and R. Underwood, *The block Lanczos method for computing eigenvalues*, Mathematical Software III, J. R. Rice, ed., Academic Press, New York, 1977, pp. 361–377.
- G. H. Golub, F. T. Luk, and M. L. Overton, *A block Lanczos method for computing the singular values and corresponding vectors of a matrix*, ACM Trans. Math. Software, 7 (1981), pp. 149–169.
- G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.
- G. H. Golub and Q. Ye *An Inverse Free Preconditioned Krylov Subspace Method for Symmetric Generalized Eigenvalue Problems*, SIAM J. Sci. Comput., 24 (2002), pp. 312-334.
- **J. Baglama, D. Calvetti, G.H. Golub, and L. Reichel, *Adaptively Preconditioned GMRES Algorithms* ,SIAM J. Sci. Comput., 20, No. 1, (1998), pp. 243-269.**

Thank you!