Communication-Optimal Eigenvalue/SVD Algorithms

Grey Ballard

ParLab Winter 2010 Retreat

Summary

New set of communication-optimal algorithms for symmetric eigenproblem, SVD, nonsymmetric eigenproblem and generalized eigenproblem

- Optimal for sequential and parallel machines
  - minimizes both words and messages moved
- Asymptotically less communication than standard algorithms
  - constant factor more arithmetic
- Use divide-and-conquer approach with randomization
- Use matrix multiplication and QR decomposition as subroutines

Note: a completely different approach, using Successive Band Reduction, minimizes the number of words moved on a sequential machine in solving the symmetric eigenproblem or computing the SVD without increasing the arithmetic (very much)

Motivation

Sequential algorithms

Communication lower bounds:

$$\#\text{ words} = \Omega\left(\frac{n^3}{\sqrt{M}}\right) \qquad \#\text{ messages} = \Omega\left(\frac{n^3}{M^{3/2}}\right)$$

Factors exceeding lower bounds:

	New Algorithms		$LAPACK$	
	# words	# messages	# words	# messages
Symm Eig/SVD Nonsymm Eig	$O(1)$	$O(1)$	$O\left(\sqrt{M}\right)$	$O\left(\sqrt{M}\right)$

Parallel algorithms

Communication lower bounds:

$$\#\text{ words} = \Omega\left(\frac{n^2}{\sqrt{P}}\right) \qquad \#\text{ messages} = \Omega\left(\sqrt{P}\right)$$

Factors exceeding lower bounds:

	New Algorithms		$ScaLAPACK$	
Algorithm	# words	# messages	# words	# messages
Symm Eig/SVD	$O(\log P)$	$O(\log^2 P)$	$O(\log P)$	$O\left(\frac{n}{\sqrt{P}}\right)$
Nonsymm Eig			$O\left(\sqrt{P}\log P\right)$	$O(n \log P)$

Divide and Conquer Algorithm

Goal of divide and conquer step is to divide the spectrum along a curve in the complex plane and orthogonally transform the matrix to block triangular in order to compute the eigenvalues of the diagonal blocks as subproblems

Algorithm 1: *Splitting the spectrum of a matrix A along unit circle*

Implicit Repeated Squaring of A^{-1}
Compute invariant subspace of $\left(I + (A^{-1})^{2^k}\right)^{-1}$
Apply orthogonal transformation to A so that $Q^T A Q = \begin{bmatrix} A_{11} & A_{21} \\ \varepsilon & A_{22} \end{bmatrix}$

▷ Similar algorithm works for generalized eigenproblem

Möbius transformations

- In order to split the spectrum along any line or circle, we can use transformations of the form $f(z) = (\alpha z + \beta)^{-1}(\gamma z + \delta)$, for complex constants $\alpha, \beta, \gamma, \delta$
- Repeatedly square (implicitly) $(\alpha A + \beta I)^{-1}(\gamma A + \delta I)$ at first step of Algorithm 1
- Choose constants differently at each step of divide and conquer

Implicit Repeated Squaring (IRS)

- No inverses computed
- Algorithm 2 yields for each j,

$$C_j^{-1} D_j = \left(C_{j-1}^{-1} D_{j-1}\right)^2 = \left(C^{-1} D\right)^{2^j}$$

- For divide and conquer step, set

$$C = \alpha A + \beta I, \quad D = \gamma A + \delta I$$

from Möbius transformation

Algorithm 2: *IRS of $C^{-1}D$*

Input: $C_0 = C, \quad D_0 = D$
repeat
$\quad \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \cdot \begin{bmatrix} R_j \\ 0 \end{bmatrix} = \text{qr}\left(\begin{bmatrix} D_j \\ -C_j \end{bmatrix}\right)$
$\quad C_{j+1} = Q_{12}{}^* \cdot C_j$
$\quad D_{j+1} = Q_{22}{}^* \cdot D_j$
until R_j converges
Output: $C_j, \ D_j$

Generalized Randomized Rank-Revealing Decomposition (GRURV)

Algorithm 3: *GRURV of $C^{-1}D$*

Input: $C, \ D$
generate random matrix B
$V \cdot R = \text{qr}(B)$
$Q \cdot R_2 = \text{qr}(D \cdot V^*)$
$R_1 \cdot U = \text{rq}(Q^* \cdot C)$
Output: $U, (R_1, R_2, V)$

- No inverses computed
- Algorithm 3 yields

$$C^{-1} D = U^* (R_1{}^{-1} R_2) V$$

- Rank-revealing with high probability assuming $C^{-1}D$ has a gap in its singular values (from IRS)

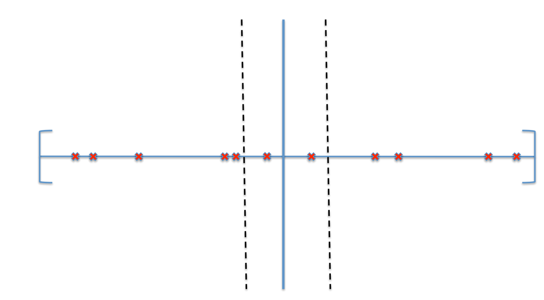
Randomized Bisection

One divide and conquer step can make progress either by splitting the spectrum or by reducing the search space. Use randomized bisection to choose a splitting line:

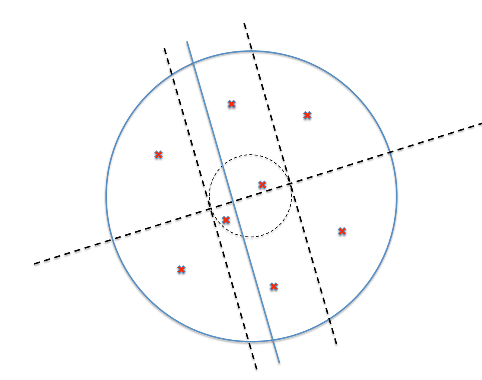
- bisection limits the number of steps necessary
- randomization ensures progress with high probability

Symmetric/SVD case

1. Find bounding interval
2. Set range around midpoint
3. Pick random split within range



Nonsymmetric case



1. Find bounding circle
2. Set range around midpoint
3. Pick random slope
4. Pick random perpendicular within range

Numerical Experiments

Convergence criteria

Algorithm 4: *R convergence*

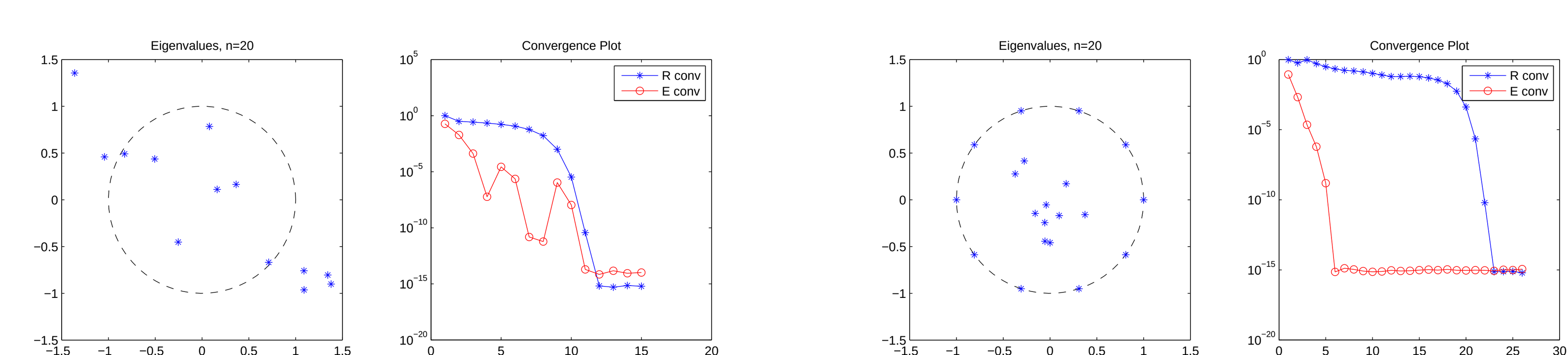
$A_0 = A, \ B_0 = I$
repeat
$\quad \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \cdot \begin{bmatrix} R_j \\ 0 \end{bmatrix} = \text{qr}\left(\begin{bmatrix} B_j \\ -A_j \end{bmatrix}\right)$
$\quad A_{j+1} = Q_{12}{}^* \cdot A_j$
$\quad B_{j+1} = Q_{22}{}^* \cdot B_j$
until $\frac{\|R_j - R_{j-1}\|}{\|R_{j-1}\|}$ is small
$U = \text{GRURV}(A_j + B_j, A_j)$
$A_{\text{new}} = U \cdot A \cdot U^* = \begin{bmatrix} A_{11} & A_{12} \\ E_{21} & A_{22} \end{bmatrix}$

Algorithm 5: *E convergence*

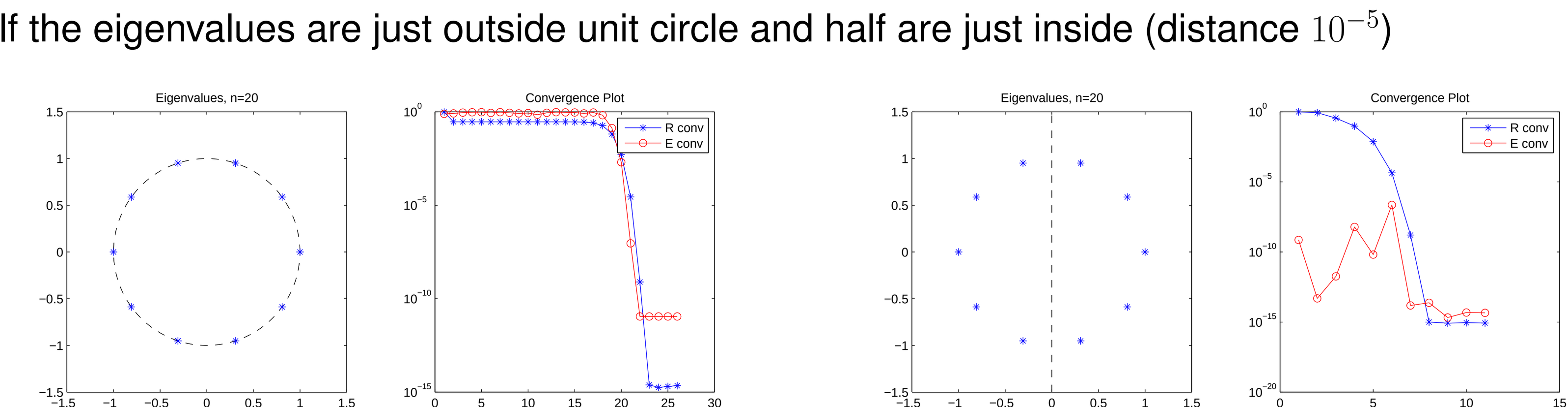
$A_0 = A, \ B_0 = I$
repeat
$\quad \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \cdot \begin{bmatrix} R_j \\ 0 \end{bmatrix} = \text{qr}\left(\begin{bmatrix} B_j \\ -A_j \end{bmatrix}\right)$
$\quad A_{j+1} = Q_{12}{}^* \cdot A_j$
$\quad B_{j+1} = Q_{22}{}^* \cdot B_j$
$\quad U = \text{GRURV}(A_j + B_j, A_j)$
$\quad A_{\text{new}} = U \cdot A \cdot U^* = \begin{bmatrix} A_{11} & A_{12} \\ E_{21} & A_{22} \end{bmatrix}$
until $\frac{\|E_{21}\|}{\|A\|}$ is small

Convergence plots

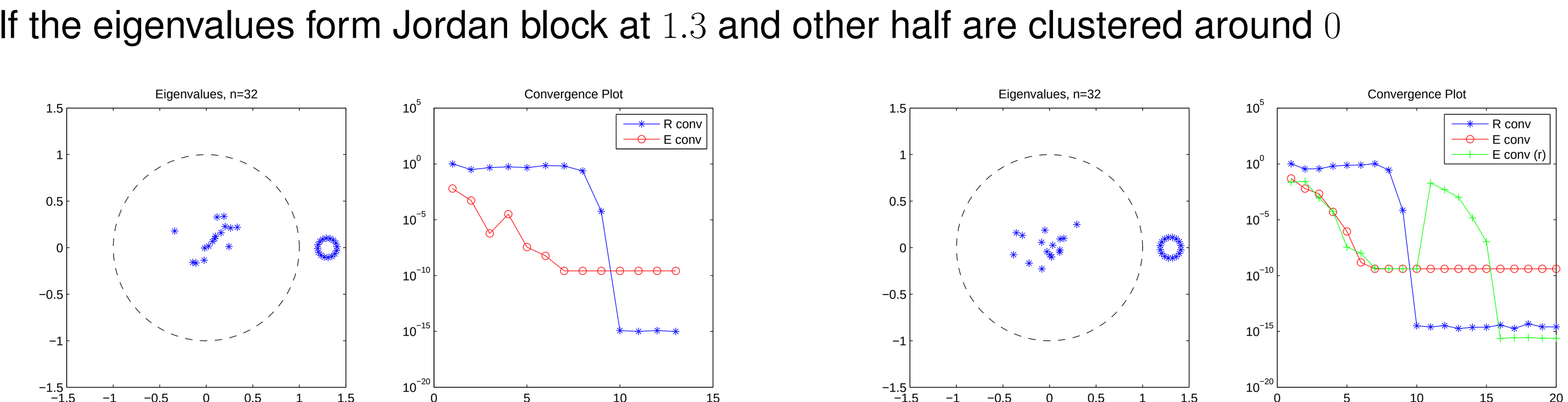
Random eigenvalues (left), half just outside unit circle (distance 10^{-5}) and half around 0 (right)



Half the eigenvalues are just outside unit circle and half are just inside (distance 10^{-5})



Half the eigenvalues form Jordan block at 1.3 and other half are clustered around 0



▷ Green curve shows convergence restarting after 10 iterations (setting $A_j = A_{\text{new}}$, $B_j = I$)