Evaluating Parallel Algorithms for Solving Sylvester-Type Matrix Equations

Direct Transformation-Based versus Iterative Matrix-Sign-Function-Based Methods

Robert Granat and Bo Kågström,
Umeå University and HPC2N, Sweden

Outline

- Sylvester-Type Matrix Equations
- Direct Transformation-Based Methods
- Matrix-Sign-Function-Based Methods
- Matrix Equations as Linear Systems
- Condition Estimation
- ScaLAPACK Environment
- Parallel Implementations
- Experimental Evalutation
- Summary
- Future/Ongoing Work
- References

Sylvester-Type Matrix Equations

- Sylvester-type matrix equations arise in many applications in science and engineering: block diagonalization of matrices in Schur form, condition estimation of eigenvalue problems, control theory etc.
- The continuous-time Sylvester equation (SYCT):

$$AX - XB = C, \quad A \in R^{M \times M}, B \in R^{N \times N}, C \in R^{M \times N}$$

- Solution is unique iff $\lambda(A) \cap \lambda(B) = \varnothing$
- If we can solve SYCT, we can solve many other similar equations
- We consider and compare two different methods

Direct Transformation-Based Methods (1)

- To solve SYCT apply Bartels-Stewart's method:
  - Transform *A* and *B* to real Schur form:
    $$T_A = Q^T A Q, \quad T_B = P^T B P$$
  - Update the matrix *C* with respect to the transformations
    $$\tilde{C} = Q^T C P$$
  - Solve the reduced triangular system
    $$T_A \tilde{X} - \tilde{X} T_B = \tilde{C}$$
  - Transform the solution back to the original coordinate system
    $$X = Q \tilde{X} P^T$$
- No extra conditions is imposed on *A* or *B* by the method

Direct Transformation-Based Methods (2)

- Triangular problem is solved by blocking

$$AX - XB = C \Leftrightarrow A_{ii}X_{ij} - X_{ij}B_{ij} = C_{ij} - (\sum_{k=i+1}^{D_a} A_{ik}X_{kj} - \sum_{k=1}^{j-1} X_{ik}B_{kj})$$

- In parallel, we traverse the matrix *C*/*X* along its block diagonals and solve SYCT subsystems and do GEMM-updates w r t the subsolutions on the nodes

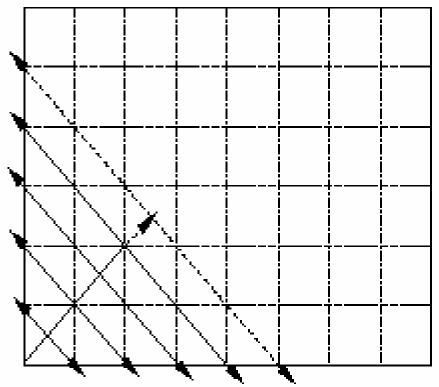


Fig. 3. Traversing the matrix C/X when solving $AX - XB = C$.

Matrix-Sign-Function-Based Methods (1)

- Let $Z \in R^{p \times p}, \quad \lambda(Z) \subset R$ have the Jordan decomposition

$$Z = S \begin{bmatrix} J^- & 0 \\ 0 & J^+ \end{bmatrix} S^{-1} \qquad J^- \in C^{k \times k}, J^+ \in C^{(p-k)\times(p-k)}$$

where J^- and J^+ contain the Jordan blocks with eigenvalues in the open left and right half planes, respectively.

- The *matrix sign function* is defined as

$$sign(Z) = S \begin{bmatrix} -I_k & 0 \\ 0 & I_{p-k} \end{bmatrix} S^{-1}$$

- The matrix sign function can be computed via Newton iteration for the equation $Z^2 = I$:

$$\begin{cases} Z_0 = Z \\ Z_{k+1} = (Z_k + Z_k^{-1})/2 \end{cases}$$

Matrix-Sign-Function-Based Methods (2)

- [J.D. Roberts, 11]: $sign(Z) = \lim_{k\to\infty} Z_k$, and
$$sign\left(\begin{bmatrix} A & -C \\ 0 & B \end{bmatrix}\right) + I_{m+n} = 2\begin{bmatrix} 0 & X \\ 0 & I \end{bmatrix}$$
- Sign-function method can be applied to SYCT iff *A* and *–B* are c-stable: $\mathrm{Re}(\lambda_i) < 0, \forall i$
- Parallel implementation by Benner and Quitana-Orti (in PSLICOT)

Matrix Equations as Linear Systems

- All linear matrix equations can be represented as a linear system of equations:

$$op(A)X - Xop(B) = C \quad \Longleftrightarrow \quad Z_{SYCT}x = y$$

$$Z_{SYCT} = I_N \otimes op(A) - op(B)^T \otimes I_M$$

$$x = vec(X), \;\; y = vec(C)$$

In blocked algorithms, $Zx = y$ representation is used in kernels for solving small-sized matrix equations.

Condition Estimation

- An important quantity in the perturbation theory for Sylvester-type equations is the *separation between two matrices*:

$$sep(A,B) = \inf_{\|X\|_F=1} \|op(A)X - Xop(B)\|_F = \sigma_{\min}(Z_{SYCT}) = \|Z_{SYCT}^{-1}\|_2^{-1}$$

- To compute *sep(A,B)* exactly costs $O(M^3N^3)$ flops (only of interest in theory). We want to compute a reliable but low cost estimate (serially as well as in parallel).
- We apply a general method (Hager'84, Higham'88, Kågström-Poromaa'92) for estimating $\|A^{-1}\|_1$ which only uses matrix vectors products:

$$A^{-1}x \qquad A^{-T}x$$

- ⇒ we can estimate *sep(A,B)* for SYCT by solving the equation itself to an $O(M^2N + MN^2)$ cost.
- Notice that when *sep(A, B)* is tiny, the SYCT equation is close to singular, i.e., ill-conditioned (compare with the scalar case $x = c / (a - b)$).

ScaLAPACK Environment

- HPC library for dense linear algebra on distributed memory machines
- Buildt on LAPACK, BLAS, PBLAS, BLACS
- Fortan 77 SPMD object-oriented programming style
- 2D processor grid
- All matrices are blockpartitioned by rows and columns and distributed using 2D block-cyclic mapping

Parallell Implementations (1)

- ScaLAPACK-style implementation of Bartels-Stewart's method (Granat-Kågström-Poromaa):
  - Reduction to triangular form
    - Hessenberg reduction – PDGEHRD
    - QR-algorithm – PDLAHQR
  - Transforming rhs and solution to tri. problem – PDGEMM
  - Solving the triangular problem
    - Kernel SYCT-solver – DTRSYL
    - GEMM-updates – DGEMM
  - Resulting routine PGESYCTD

Parallell Implementations (2)

- ScaLAPACK-style implementation of the Matrix-sign-function-based method (Benner and Quintana-Orti):
  - LU decomposition – PDGETRF
  - Solving linear systems of equations – PDGETRS
  - Inversion based on LU decomposition – PDGETRI
  - Solving triangular systems with multiple rhs – PDTRSM
  - Pivoting of a distributed matrix - PDLAPIV
  - Resulting routine from PSLICOT – psb04md

Experimental Evaluation (1)

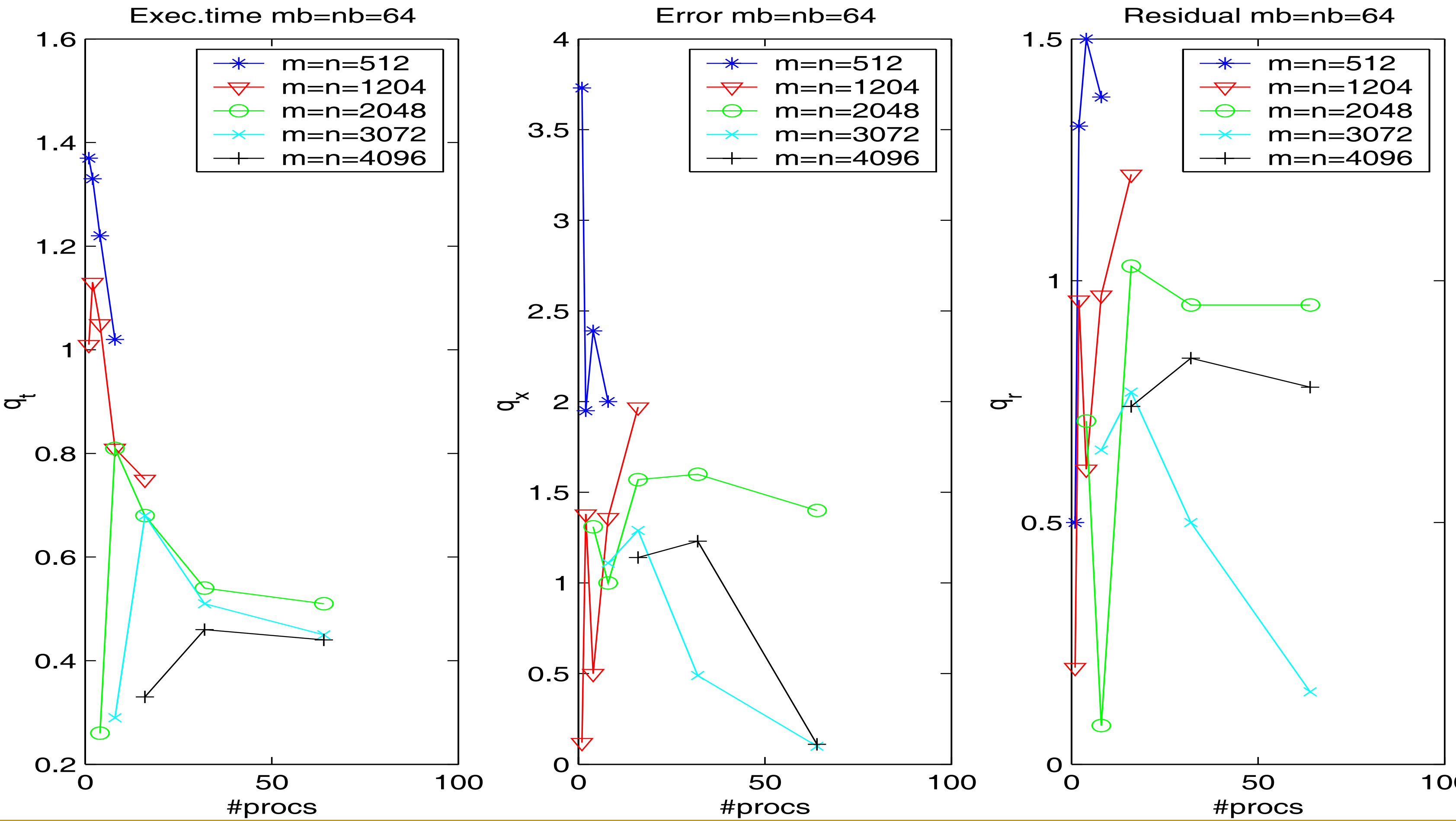
- Two target parallel computers:
  - IBM SP system
    - 64 thin 120MHz nodes with 128MB RAM
    - 150 Mbyte/sec peak network bandwidth
    - Well-balanced: t_flop/t_comm = 0.11
  - Linux Super Cluster
    - 120 dual 1.667MHz nodes with 1GB RAM
    - 667 Mbyte/sec peak network bandwidth
    - Less well-balanced: t_flop/t_comm = 0.025

Experimental Evaluation (2)

- Test problem matrices: $A = Q(\alpha D_A + \beta M_A)Q^T$
- We present three performance ratios q_T, q_X, q_R
  - Measured parallel execution time
  - Accuracy:
    - Frobenius norm of absolute error: $\|X - \tilde{X}\|_F$
    - Frobenius norm of absolute residual: $\|A\tilde{X} - \tilde{X}B - C\|$
- If any ratio > 1, psb04md shows better results, otherwise PGESYCTD performs equal or better
- Condition estimation by computing a lower bound of $sep^{-1}(A, B)$

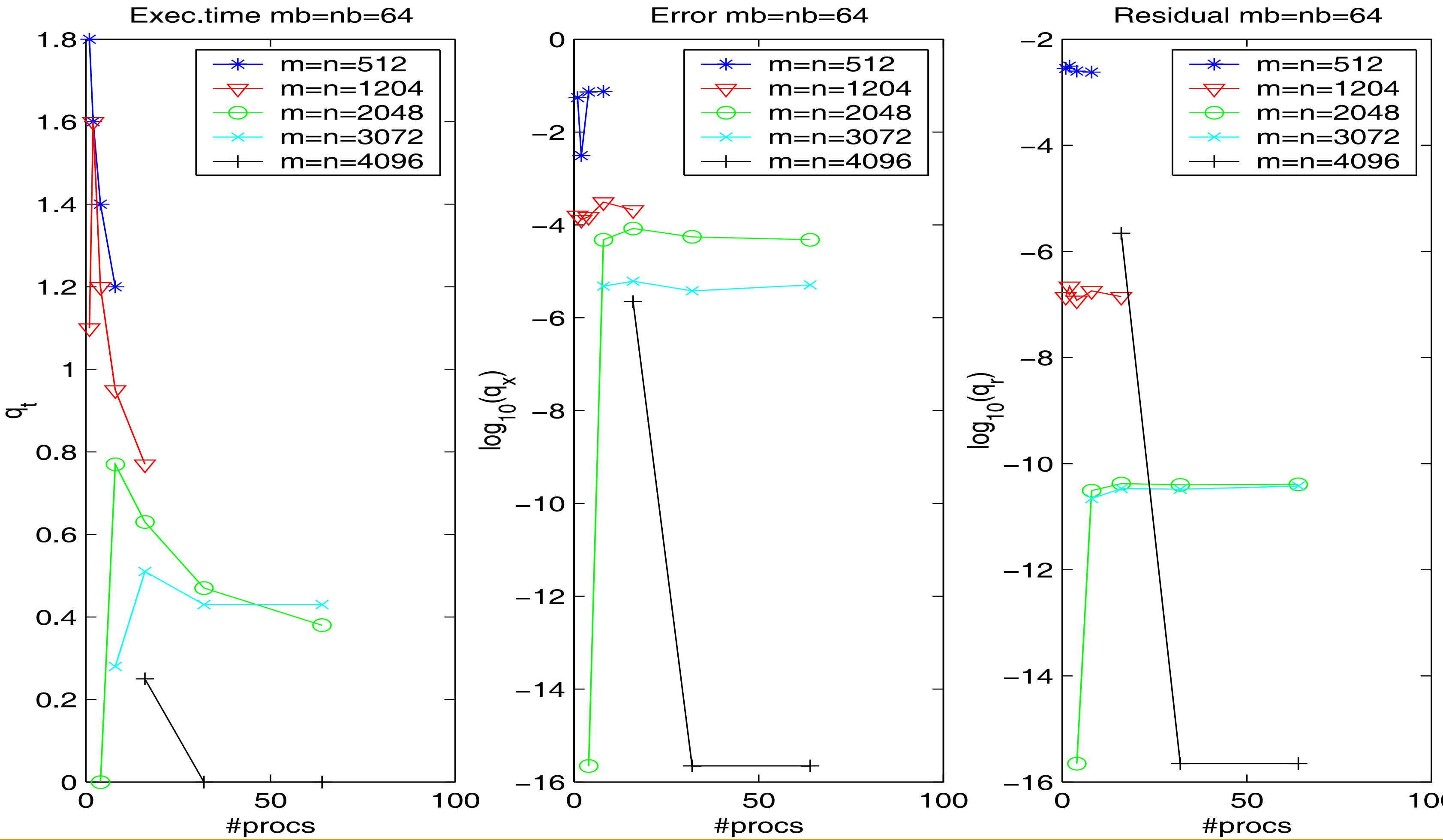
Experimental Evaluation (3)

Wellconditioned problems on IBM SP $sep_{est}^{-1}(A,B) \approx 10^{-3}$



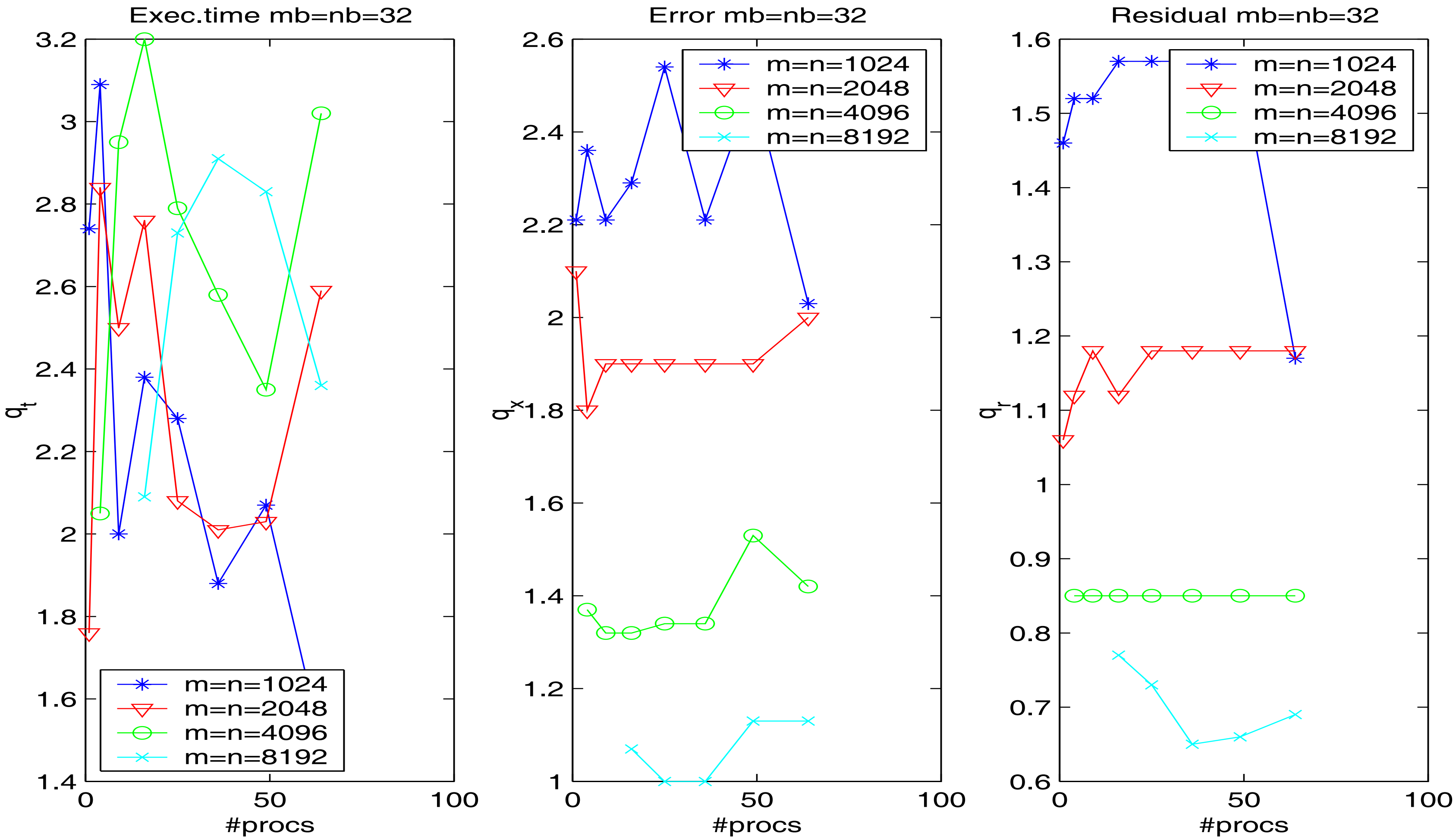
Experimental Evaluation (4)

Illconditioned problems on IBM SP $10^{-1} \le sep_{est}{}^{-1}(A,B) \le 10^{5}$



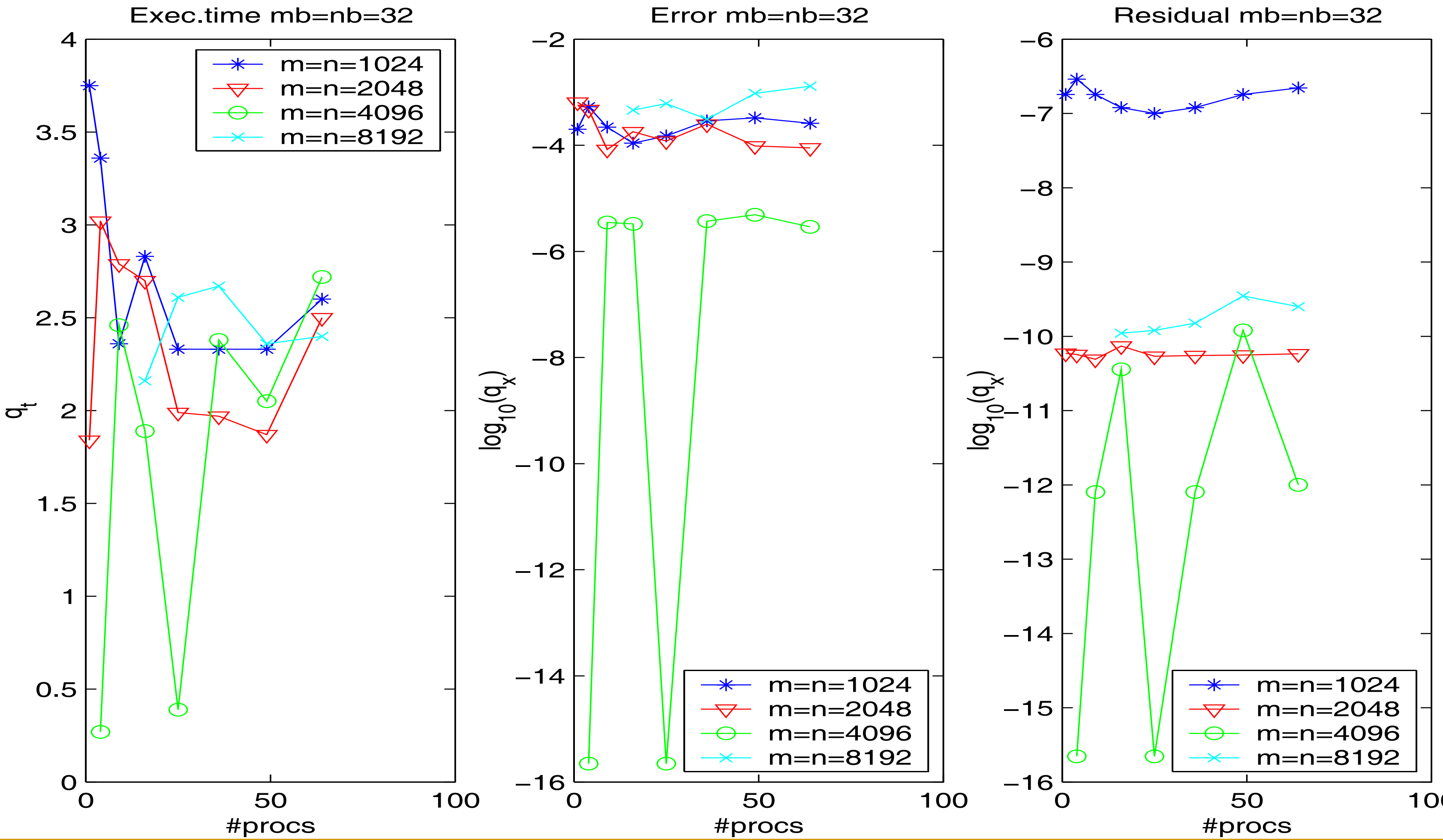
Experimental Evaluation (5)

Wellconditioned problems on Linux Super Cluster $sep_{est}^{-1}(A,B) \approx 10^{-3}$



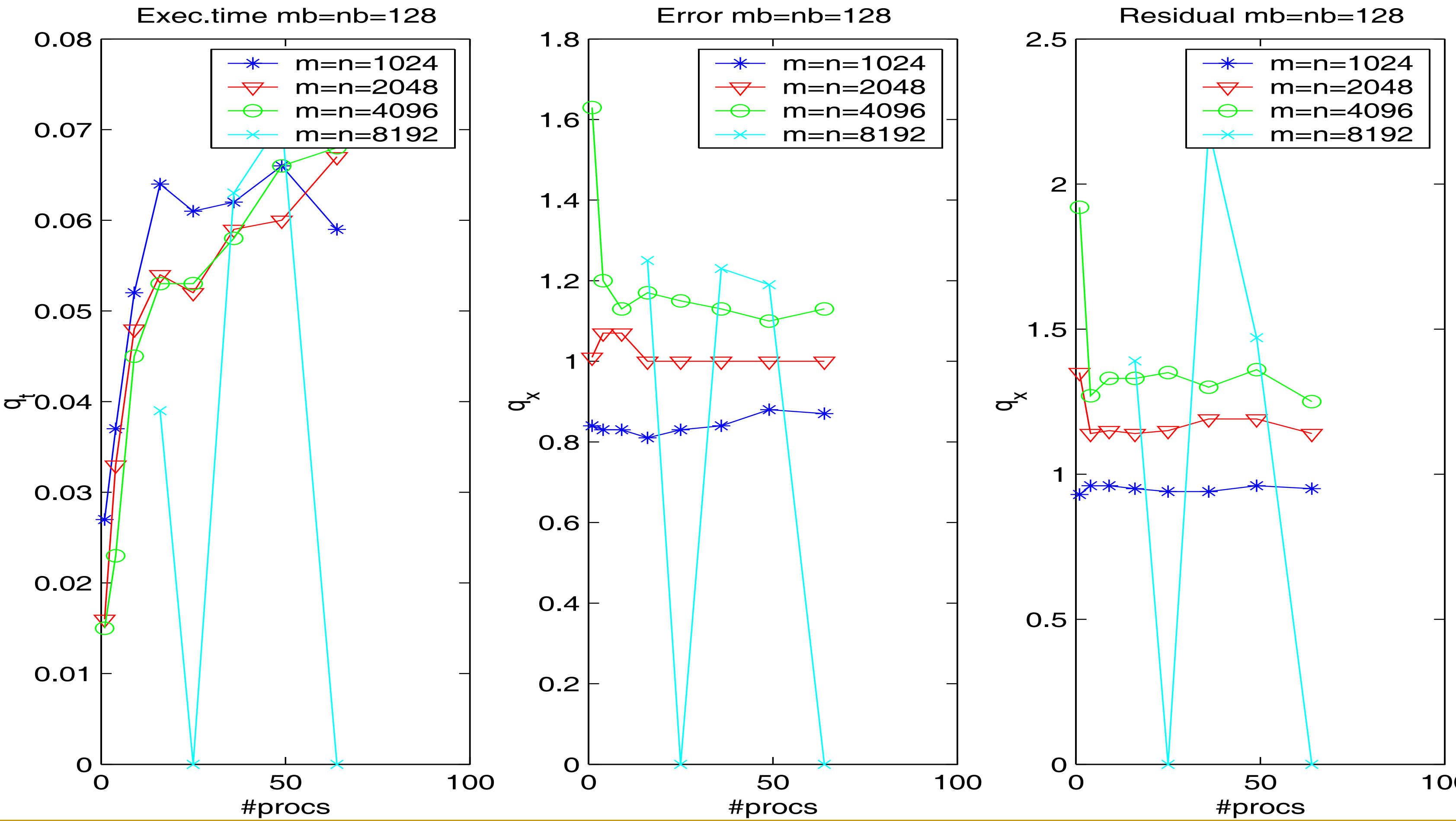
Experimental Evaluation (6)

Illconditioned problems on Linux Super Cluster $10^0 \leq sep_{est}{}^{-1}(A,B) \leq 10^6$



Experimental Evaluation (7)

Wellconditioned ***triangular*** problems on Linux Super Cluster $sep_{est}^{-1}(A,B) \approx 10^{-3}$



Summary

Routine	Generality	Reliability	Speed	Accuracy
PGESYCTD	*A* and *B* must have no eigenvalues in common	Always delivers a result	▪ Up to four times faster on the most balanced parallel platform ▪ Always much faster for triangular problems (even on less balanced platform)	▪ Always much better for illconditioned problems ▪ Tends to deliver the smallest residual norm
psb04md	▪ *A* and *B* must have no eigenvalues in common ▪ *A* and -*B* must be c-stable	Did not always converge for illconditioned problems	Always faster for general problems on the less balanced platform when converging	Slightly better absolute error norm for wellconditioned problems

Ongoing/Future Work

- New implementations for all transpose and sign-variants of the non-generalized standard matrix equations
- Software package *SCASY* will also contain generalized solvers and parallel condition estimators
- Ongoing investigation of hybrid algorithms with fast kernels from HPC library *RECSY*

$op(A)X \pm Xop(B) = C$	SYCT	√
$op(A)X + Xop(A^T) = C$	LYCT	√
$op(A)Xop(B) \pm X = C$	SYDT	√
$op(A)Xop(A^T) - X = C$	LYDT	√
$\begin{cases} op(A)X \pm Yop(B) = C, \\ op(D)X \pm Yop(E) = F \end{cases}$	GCSY	
$op(A)Xop(B) \pm op(D)Xop(E) = C$	GSYL	
$op(A)Xop(A^T) - op(E)Xop(E^T) = C$	GLYCT	
$op(A)X(E^T) + op(E)Xop(A^T) = C$	GLYDT	

References

[1] R.H. Bartels and G.W. Stewart. Algorithm 432: Solution of the Equation AX+XB = C, *Comm. ACM*, 15(9):820-826, 1972.

[2] P. Benner, E.S. Quintana-Orti. Solving Stable Generalized Lyapunov Equations with the matrix sign functions, *Numerical Algorithms*, 20(1), pp. 75-100, 1999.

[3] P. Benner, E.S. Quintana-Orti, G. Ouintana-Orti. Numerical Solution of Discrete Schur Stable Linear Matrix Equations on Multicomputers, *Parallel Alg. Appl*., Vol. 17, No 1, pp. 127-146, 2002.

[4] R. Granat, B. Kågström, P. Poromaa. Parallel ScaLAPACK-style Algorithms for Solving Continuous-Time Sylvester Equations. In H. Kosch et al. (eds), Euro-Par 2003 Parallel Processing. *Lecture Notes in Computer Science*, Vol. 2790, pp. 800-809, 2003.

[5] W.W. Hager, Condition Estimates, *SIAM J. Sci. Statist. Comput.* 5, pp. 311-316,1984.

[6] N. J. Higham, FORTAN codes for Estimating the One-Norm of a Real of Complex Matrix, *ACM Trans. Of Math. Software*, Vol. 14, No. 4, pp. 381-396, December 1988

[7] I. Jonsson and B. Kågström, Recursive Blocked Algorithms for Solving Triangular Matrix Equations - Part I: One-Sided and Coupled Sylvester-Type Equations, *ACM Trans. Math. Software*, Vol. 28, No. 4, pp 393-415, 2002.

[8] I. Jonsson and B. Kågström, Recursive Blocked Algorithms for Solving Triangular Matrix Equations - Part II:Two-Sided and Generalized Sylvester and Lyapunov Equations, *ACM Trans. Math. Software*, Vol. 28, No. 4, pp 416-435, 2002.

[9] B. Kågström and P. Poromaa, Distributed and shared memory block algorithms for the triangular Sylvester Equation with Sep^{-1} estimators, *SIAM J. Matrix Anal. Appl.,* 13 (1992), pp. 99-101.

[10] P. Poromaa. Parallel Algorithms for Triangular Sylvester Equations: Design, Scheduling and Scalability Issues. In Kågström et al. (eds), *Applied Parallel Computing. Large Scale Scientific and Industrial Problems, Lecture Notes in Computer Science*, Vol. 1541, pp. 438-446, Springer-Verlag, 1998.

[11] J.D. Roberts. Linear model reduction and solution of the algebraic Ricatti Equation by use of the sign function, *Intern. J. Control*, 32:667-687, 1980.