Parallel Solution of Large-Scale Algebraic Bernoulli Equations with the matrix Sign Function Method

Enrique S. Quintana-Ortí

Depto. de Ingeniería y Ciencia de Computadores
Universidad Jaume I de Castellón (Spain)
quintana@icc.uji.es

Joint work with:
Sergio Barrachina (UJI), Peter Benner (TU-Chemnitz)

Oslo - June 2005

Algebraic Bernoulli Equation (ABE)

Solve the equation:

$$A^T X + XA - XGX = 0,$$

where

- $A \in \mathbb{R}^{n \times n}$,
- $G \in \mathbb{R}^{n \times n}$ is symmetric,
- $X \in \mathbb{R}^{n \times n}$ is the sought-after solution.

What is the ABE?

A degenerate Algebraic Riccati Equation (ARE):

$$A^T X + XA - XGX + Q = 0,$$

with $Q = 0$, as well as a special case of the more general equation

$$\mathcal{L}(X) + X \left(\prod_{j=1}^{k-1} A_j X \right) = 0,$$

where

- $\mathcal{L}(X)$ is a linear operator,
- $A_j \in \mathbb{R}^{n \times n}$, $j = 1, \ldots, k-1$.

Interest?

Dealing with dynamical linear time-invariant (LTI) systems:

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad t > 0, \qquad x(0) = x^0,$$

- n state-space variables, i.e., n is the order of the system;
- m inputs.

Applications in systems and control theory:

- Stabilization.
- Model reduction.

Interest? (Cont. I)

Stabilization problem: Find $u(t)$ s.t. the solution of

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad t > 0, \qquad x(0) = x^0,$$

asymptotically converges to zero.

The maximal solution X_* of the ABE

$$A^T X + XA - XBB^T X = 0,$$

defines the feedback law

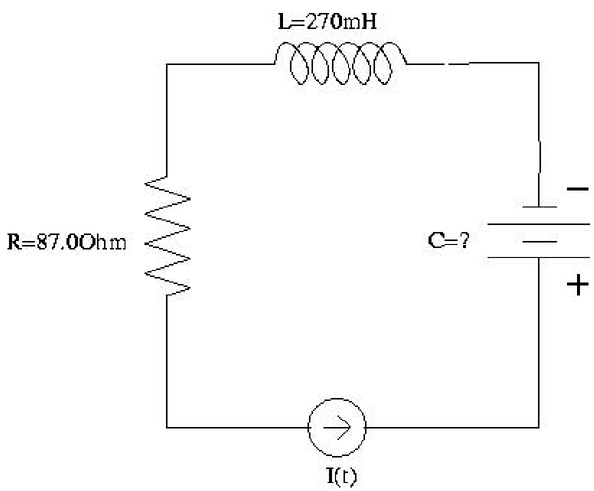
$$u(t) = -B^T X_* x(t), \qquad t \geq 0,$$

which stabilizes the system!

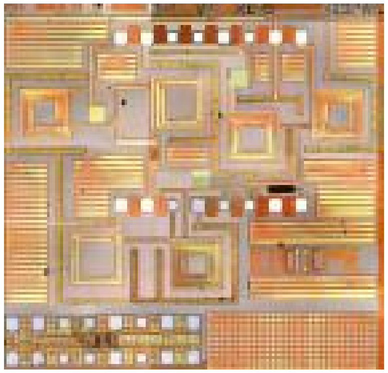
Interest? (Cont. II)

Large-scale unstable dynamical LTI systems [Kamon, Wang, White'98]:

RLC cicuits



VLSI chip design



Large-scale means n as large as $10^3 - 10^4$!

Numerical Solvers

Specialized cases of ARE solvers [Mehrmann'91]:

Compute a basis

$$\left[U^T \; V^T\right]^T, \qquad U,\; V \in \mathbb{R}^{n\times n},$$

for the invariant subspace of the $2n \times 2n$ matrix

$$H \;:=\; \begin{bmatrix} A & G \\ 0 & -A^T \end{bmatrix}$$

associated with the eigenvalues of H in the open left half plane.

Then, the stabilizing solution of the ABE (provided it exists) can be computed by

$$X_* := -VU^{-1}.$$

Numerical Solvers (Cont. I)

Computing the appropriate basis:

- Compute the real Schur form of H (actually, just A) and reorder the eigenvalues.
- Use a spectral projector such as the matrix sign function [Roberts, 80].

Computational and storage costs $\mathcal{O}(n^3)$ and $\mathcal{O}(n^2)$, respectively.

$\Longrightarrow$ Need for parallel computing!

Numerical Solvers (Cont. II)

- Real Schur form of H via the QR algorithm:
  - Implicitly iterative.
  - Composed of fine grain operations.
  - Hard to parallelize, but doable [ScaLAPACK'97].
  - Parallel reordering procedure not available :-(
- Matrix sign function:
  - Explicitly iterative.
  - Claim: Easy to parallelize.

The convergence of the methods depends on the problem:

$\Longrightarrow$ Difficult to infer general results!

Sign Function Method

The classical Newton iteration for the matrix sign function

$$Z_0 \leftarrow Z, \quad Z_{k+1} \leftarrow \frac{1}{2}(Z_k + Z_k^{-1}), \qquad k = 0, 1, 2, \ldots,$$

when applied to $H := \begin{bmatrix} A & G \\ 0 & -A^T \end{bmatrix}$ boils down to

$$\begin{array}{rcl} A_0 & \leftarrow & A, \quad A_{k+1} \leftarrow \frac{1}{2}\left(\frac{1}{c_k}A_k + c_k A_k^{-1}\right), \\ G_0 & \leftarrow & G, \quad G_{k+1} \leftarrow \frac{1}{2}\left(\frac{1}{c_k}G_k + c_k A_k^{-1} G_k A_k^{-T}\right), \\ & & k = 0, 1, 2, \ldots. \end{array}$$

Sign Function Method (Cont. I)

Scaling for acceleration of convergence:

- Determinantal:

$$c_k := |\det(H_k)|^{1/n}.$$

- Optimal norm:

$$c_k := \sqrt{\frac{\|H_k\|_2}{\|H_k^{-1}\|_2}}.$$

- Approximate optimal norm:

$$c_k := \sqrt{\frac{\|H_k\|_F}{\|H_k^{-1}\|_F}} = \sqrt{\frac{\sqrt{2\|A_k\|_F^2 + \|G_k\|_F^2}}{\sqrt{2\|A_k^{-1}\|_F^2 + \|A_k^{-1}G_kA_k^{-T}\|_F^2}}}.$$

Sign Function Method (Cont. II)

Convergence criterion:

- Stop when
$$\|A_{k+1} - A_k\|_F \leq \tau \cdot \|A\|_F,$$
where τ is a tolerance threshold.
- Use $\tau = \sqrt{\varepsilon}$, with ε as the machine precision, and perform 1–3 additional iterations once this criterion is satisfied.

At convergence, after $\bar{k}$ iterations, solve the full-rank linear least-squares problem
$$\begin{bmatrix} G_{\bar{k}} \\ I_n - A_{\bar{k}}^T \end{bmatrix} X = \begin{bmatrix} A_{\bar{k}} + I_n \\ 0_n \end{bmatrix}.$$

Operation Costs

- Matrix factorization and triangular linear systems solves...

$$A_{k+1} \leftarrow \frac{1}{2}\left(A_k + A_k^{-1}\right),$$
$$G_{k+1} \leftarrow \frac{1}{2}\left(G_k + A_k^{-1} G_k A_k^{-T}\right).$$

Cost: $6n^3$ flops per iteration.

- Full-rank least squares problem via QR factorization:

$$\left[\begin{array}{c} G_{\bar{k}} \\ I_n - A_{\bar{k}}^T \end{array}\right] X = \left[\begin{array}{c} A_{\bar{k}} + I_n \\ 0_n \end{array}\right].$$

Cost: $\frac{14}{3}n^3$ flops.

Parallel Implementation

Easy parallelization using, e.g., ScaLAPACK.

Computing the iterates:

$$A_{k+1} \leftarrow \frac{1}{2}\left(A_k + A_k^{-1}\right), \qquad G_{k+1} \leftarrow \frac{1}{2}\left(G_k + A_k^{-1} G_k A_k^{-T}\right).$$

- LU factorization of A, followed by triangular linear system solve, and matrix inversion from LU factors.
- A^{-1} needs to be computed anyway!
  Invert A first and then perform two matrix products.

Parallel Implementation (Cont. I)

Matrix inversion:

- LU factorization, triangular matrix inversion, and triangular linear system solve.
- Direct inversion via Gauss-Jordan elimination [Quintana-Ortí^2, Sun, van de Geijn'01].
  - Well suited for distributed memory machines.
  - Cyclic distribution not necessary for load balance.
  - All computations in rank-k update: High performance.

Numerical Experiments

Unstable systems in the ARE benchmark [Benner, Laub, Mehrmann'95] (AREb) and others.

Stabilization of $(\hat{A}, B) = (A + \delta I_n, B)$.

Test:

- Relative residual

$$\mathcal{R}_1(X_*) := \frac{\|\hat{A}^T X_* + X_* \hat{A} - X_* B B^T X_*\|_1}{\|X_*\|_1}.$$

- Stabilized closed-loop $A - BB^T X_*$?

Numerical Experiments (Cont. I)

Example	n	δ	Iter.	$\mathcal{R}_1(X_*)$	Stab.?	Observ.
AREb 1	2	1.0e-4	4	4.9e-28	Yes	
AREb 2	2	0.0	5	5.8e-15	Yes	
AREb 7	2	0.0	5	0.0e+00	Yes	
AREb 9	2	1.0	4	1.3e-15	Yes	
AREb 10	2	0.0	5	8.5e-09	Yes	
AREb 11	2	0.0	5	1.0e-15	Yes	
AREb 12	3	0.0	7	3.3e-09	Yes	
AREb 13	4	1.0e-8	7	2.9e-10	Yes	
AREb 14	4	0.0	5	5.5e-11	Yes	
AREb 15	39	1.0e-6	6	8.4e-11	Yes	
AREb 16	64	1.0e-4	17	1.2e-11	Yes	
AREb 17	21	1.0	8	5.1e-01	No	All eig. at 0.0
AREb 19	60	1.0e-4	17	1.1e-14	Yes	
RLC	199	1.0e-6	31	1.1e-14	Yes	Gen. system

Parallel Experiments

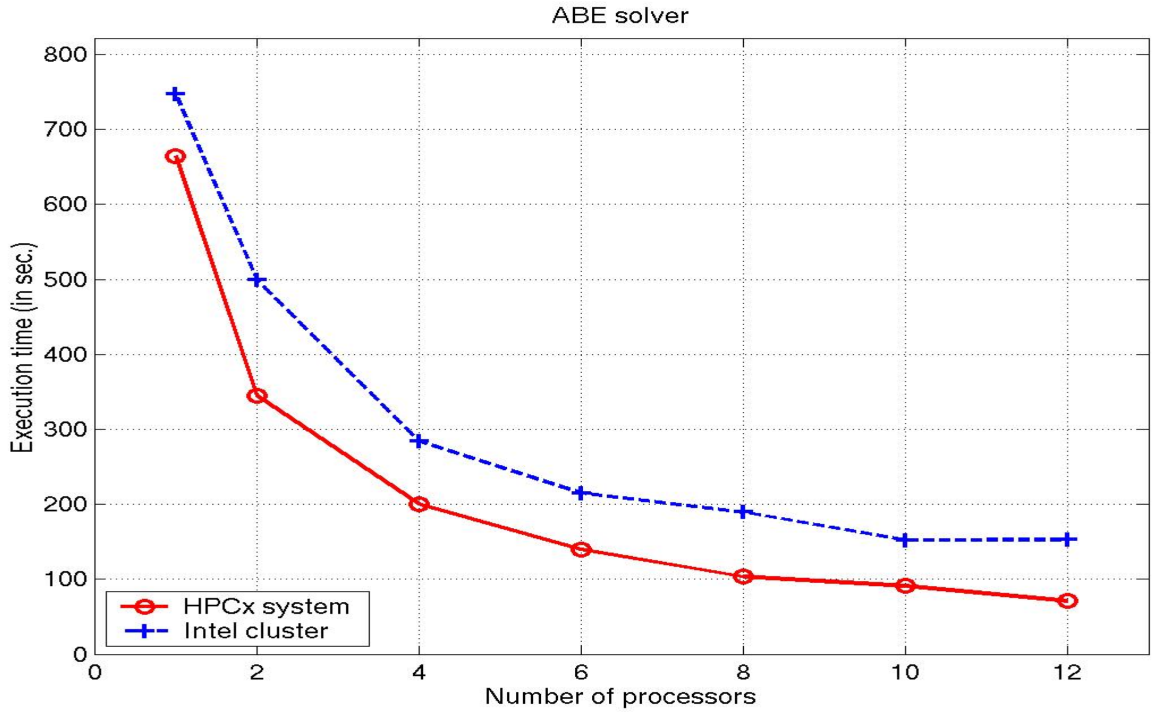
Double precision arithmetic using random unstable systems of order n.

Two parallel platforms:

- HPCx system with POWER4+ Regatta nodes:
  - 32 POWER4+ processor@1.7 GHz, with 1.5 Mbytes of L2 cache, and 32 Gbytes RAM.
- Cluster of Intel processors:
  - 20 Intel Xeon@2.4 GHz, with 1 Gbyte of RAM, connected via Myrinet switches.

Parallel Experiments (Cont. I)

Reduction in execution time for ABE of dimension n=3200.



Parallel Experiments (Cont. II)

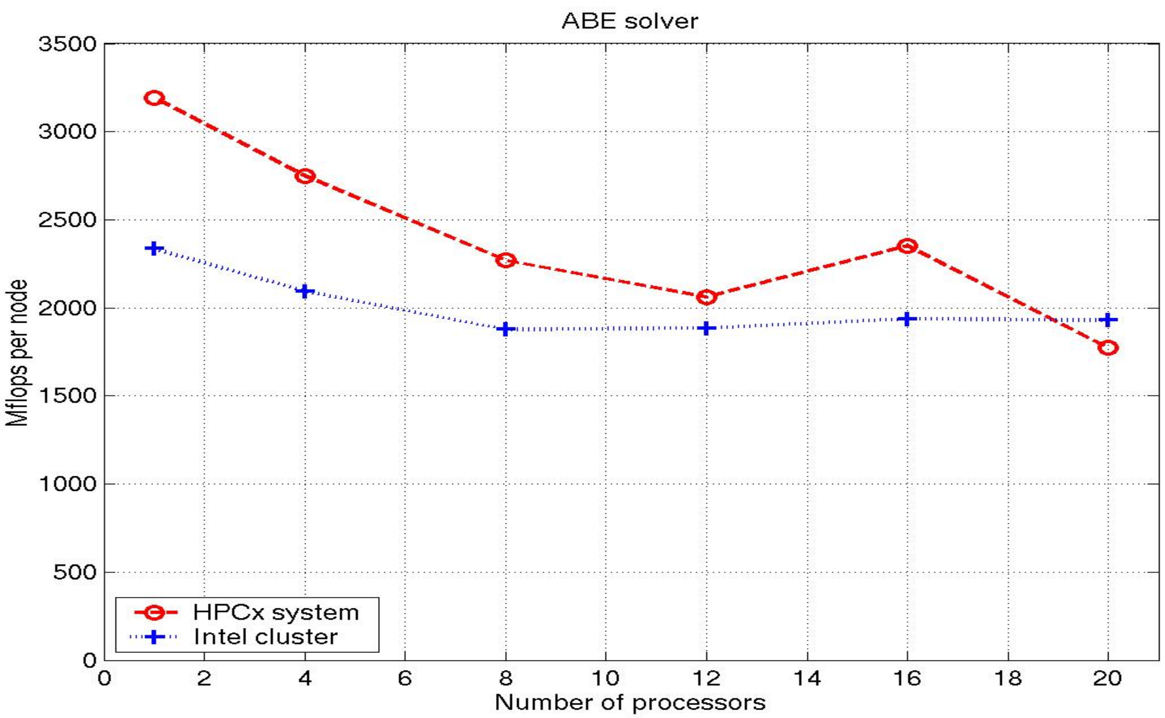
Speed-up for ABE of dimension n=3200.

n_p	2	4	6	8	10	12
HPCx	1.92	3.31	4.75	6.41	7.30	9.38
Intel cluster	1.49	2.62	3.47	3.93	4.90	4.89

Efficiency on Linux cluster quite more reduced!

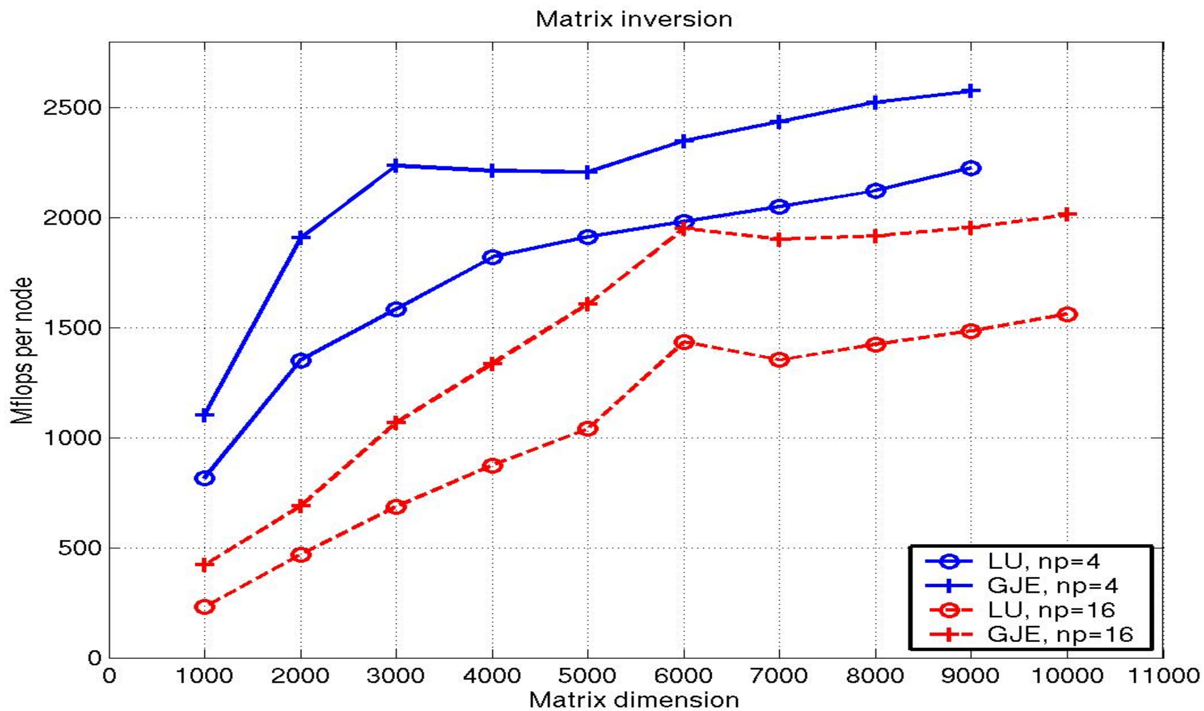
Parallel Experiments (Cont. III)

Scalability for ABEs of dimension $n/\sqrt(n_p)$=3200.



Parallel Experiments (Cont. IV)

Impact of matrix inversion procedure (not included in ABE solvers):



Concluding Remarks

- Matrix sign function is a efficient tool to solve large-scale ABE.
- As with most iterative algorithms, performance depends on the problem.
- Convergence criterion works fine in most situations.
- Determinantal scaling "seems" to work better than approx. optimal norm scaling.
- Easy and efficient (Intel cluster?) parallelization.

Some other conclusions/future work:

- Direct extension to $A^T XE + E^T XA - E^T XGXE = 0$.
- Possible specialization to compute a full-rank factor of X.
- Easy to exploit symmetry of A.

References

1. P. Benner, A. Laub, V. Mehrmann. A collection of benchmark examples for the numerical solution of AREs I. Technical Report SPC95_22, Fakultät für Mathematik, TU Chemnitz-Zwickau, Chemnitz (Germany), 1995.
2. S. Blackford et al. *ScaLAPACK Users' Guide*. SIAM, Philadelphia, PA, 1997.
3. M. Kamon, F. Wang, and J. White. Recent improvements for fast inductance extraction and simulation packaging. In *Proc. of the IEEE 7th Meeting on Electrical Performance of Electronic Packaging*, 281–284, 1998.
4. V. Mehrmann. *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution*. Lecture Notes in Control & Information Sciences 163, 1991.
5. E.S. Quintana-Ortí, G. Quintana-Ortí, X. Sun, and R. van de Geijn. A note on parallel matrix inversion. *SIAM J. Sci. Comput.*, 22:1762–1771, 2001.
6. J. Roberts. Linear model reduction and solution of the algebraic Riccati equation by use of the sign function. *Internat. J. Control*, 32:677–687, 1980.