Parallel Solution of Large-Scale and Sparse Generalized algebraic Riccati Equations

J.M. Badía, R. Mayo & E.S. Quintana

Univ. Jaime I, Castellón

P. Benner

Technische Universität Chemnitz

Contents

Algebraic Riccati Equation (ARE)

$$0 = A^T XE + E^T XA - E^T XBR^{-1}B^T XE + C^T QC =: \mathcal{R}(X)$$

$$A,\ E \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m},\ C \in \mathbb{R}^{p\times n},$$

$R \in \mathbb{R}^{m\times m}$ symmetric positive definite, and

$Q \in \mathbb{R}^{p\times p}$ symmetric positive semidefinite.

- **Applications**
  - Model reduction,
  - Filtering,
  - Design of dynamical linear systems, ...

1 Introduction (II)

- Many methods for solving small dense AREs.
  - Computational cost: $O(n^3)$.
  - Spatial cost: $O(n^2)$.
- **Problem:** What about large equations?
  - $n = 100,000$
  - Computations cost: $O(10^{15}) flops \approx 250\ hours\ at\ 1\ GFLOPs$!!!
  - Spatial cost: $O(10^{10})\ bytes \approx 80 GBytes$!!!

The problem of the scale

- $O(10^2)$:
  - MATLAB toolboxes.
  - sequential libraries, i.e. `SLICOT`.
- $O(10^3)$:
  - parallel libraries for dense AREs, i.e. `PLiCOC`.
- $O(10^4) - O(10^5)$: We need new methods that,
  - exploit the sparsity (or band) of the matrices,
  - use parallel architectures.

2 Newton's method

➤ **Newton's method for AREs**, [Kleinman'68, Penzl'00]

1) Compute the Cholesky factorization $Q = \bar{Q}\bar{Q}^T$

2) Compute the Cholesky factorization $R = \bar{R}\bar{R}^T$

3) $\bar{C} := \bar{Q}^T C$

4) $\bar{B} := E^{-1}BR^{-1} = ((E^{-1}B)\bar{R}^{-T})\bar{R}^{-1}$

repeat with $j := 0, 1, 2, \ldots$

5) $K_j := E^T X_j B R^{-1} = E^T X_j E \bar{B}$

6) $\hat{C}_j := \begin{bmatrix} \bar{C} \\ \bar{R}^T K_j^T \end{bmatrix}$

7) Solve **for** X_{j+1}:

$$0 = (A - BK_j^T)^T X_{j+1} E + E^T X_{j+1}(A - BK_j^T) + \hat{C}_j^T \hat{C}_j$$

until $\|X_j - X_{j-1}\| < \tau \|X_j\|$

2 Newton's method (II)

- If $(A - BR^{-1}B^T X_0,\ E)$ is a stable matrix pair $\Rightarrow$ The method converges quadratically to the solution X.
- Usually A, E are sparse, and $m, p \ll n$.
- **Problem:** X is dense.
- **Solution:** Exploit that X is often of low-numerical rank.
  - Approximate $X \approx \hat{R}\hat{R}^T$, $\hat{R} \in \mathbb{R}^{n \times r}$, with $r \ll n$.
- Modify Newton's method to iterate on $\hat{R}_j \hat{R}_j^T$ instead of X_j.

2 Newton's method (III)

Basic operations and Computational cost

1) Cholesky factorization	$p^3/3$
2) Cholesky factorization	$m^3/3$
3) Dense matrix product	$n^2 p$
4) Dense/Sparse systems (with B/E)	$2m^2n$ + Sparse solve (E)
repeat with $j := 0, 1, 2, \ldots$	
5) 2 matrix products	$4r_j mn$, $r_j \ll n$
6) matrix product	$2m^2n$
7) Lyapunov equation	see later
until $\|X_j - X_{j-1}\| < \tau \|X_j\|$	

3 Low Rank Solution of Lyapunov Equations

- **Lyapunov equation to solve on each iteration of Newton's method**

$$0 = (A - BK^T)^T YE + E^T Y(A - BK^T) + \hat{C}^T \hat{C}$$

$A,\ E \in \mathbb{R}^{n \times n}$, $B, K \in \mathbb{R}^{n \times m}$, and $\hat{C} \in \mathbb{R}^{(p+m) \times n}$

- We are interested on computing a full-rank factor $S \in \mathbb{R}^{n \times s}$, with $s \ll n$, such that $SS^T \approx Y$.

- **Method:**

  Low-**R**ank **A**lternating **D**irection **I**mplicit iteration (LR-ADI). [Penzl'00, LiW'02]

3 Low Rank Solution of Lyapunov Equations (II)

- **LR-ADI iteration**

  1) $V_0 := ((A - BK^T)^T + \sigma_1 E^T)^{-1} \hat{C}^T$

  2) $S_0 := \sqrt{-2\,\alpha_1}\, V_0$

  repeat with $l := 0, 1, 2, \ldots$

  3) $V_{l+1} := V_l - \delta_l ((A - BK^T)^T + \sigma_{l+1} E^T)^{-1} V_l$

  4) $S_{l+1} := [S_l \ , \ \gamma_l V_{l+1}]$

  until $\|\gamma_l V_l\|_1 < \tau \|S_l\|_1$

- $\{\sigma_1, \sigma_2, \ldots\}$, is a cyclic set of (possibly complex) shift parameters (that is, $\sigma_l = \sigma_{l+t}$ for a given period t).
- The convergence rate depends on the shift parameters. Super-linear at best.
- At each iteration the dimension of S increases by $(p+m)$ columns $\Rightarrow S_{\bar{l}} \in \mathbb{R}^{n \times \bar{l}(p+m)}$.

3 Low Rank Solution of Lyapunov Equations (III

- Main step of the LR-ADI iteration:

  Solution of the linear system $((A - BK^T)^T + \sigma E^T)V = W$

- **Problem:** even if A and E are sparse, the coefficient matrix is not necessarily sparse.

- **Solution:** Apply the *Sherman-Morrison-Woodbury (SMW) formula:*

$$(\bar{A} - BK^T)^{-1} = \bar{A}^{-1} + \bar{A}^{-1}B(I_m - K^T\bar{A}^{-1}B)^{-1}K^T\bar{A}^{-1}.$$

3 Low Rank Solution of Lyapunov Equations (IV

- Application of the SMW formula.

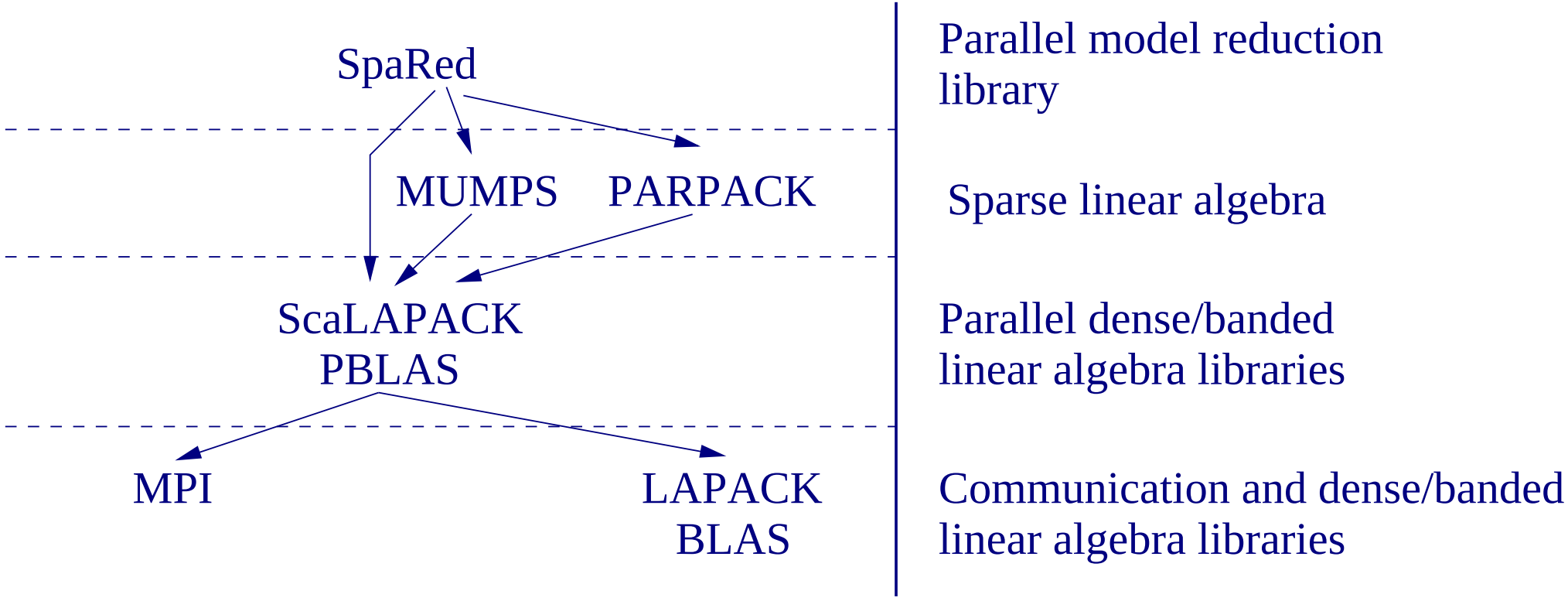
  1) $V := \bar{A}^{-T} W$
  2) $T := \bar{A}^{-T} K$
  3) $F := I_m - B^T T$
  4) $T := T F^{-1}$
  5) $V := V + T(B^T V)$

- Steps 1 and 2 require the solution of two sparse linear systems.
- Steps 3, 4 and 5 operate with small dense matrices, $F \in \mathbb{R}^{m \times m}$, $T \in \mathbb{R}^{n \times m}$.
  - Step 3, $2m^2 n$ flops.
  - Step 4, $2m^3/3 + m^2 n$ flops.
  - Step 5, $4mn(m+p)$ flops.

4 Parallel Solution

- Basic linear algebra operations:
  - Cholesky factorization.
  - Matrix product.
  - Linear system solution (dense, sparse and band).
- **Main idea:** Combine and exploit sequential and parallel libraries and tools.

Multilayered architecture of the ARE solver



4 Parallel Solution (III)

Newton's method				
Step	Operation	Matrices	Parallel lib.	Routine
1	Factorize Q	All dense	ScaLAPACK	`p_potrf`
2	Factorize R	All dense	ScaLAPACK	`p_potrf`
3	Compute $\bar{Q}^T C$	All dense	PBLAS	`p_gemm`
4.1	Solve $E^{-1}B$	Sparse E/	MUMPS or	`_mumps_c` or
		dense (BR^{-1})	ScaLAPACK	`p_gbsv`
4.2	Solve $((E^{-1}B)\bar{R}^{-T})\bar{R}^{-1}$	All dense	PBLAS	`p_trsm`$\times 2$
5	Compute $\hat{R}_j(\hat{R}_j^T \bar{B})$	All dense	PBLAS	`p_gemm`$\times 2$
6	Compute $\bar{R}^T K_j^T$	All dense	PBLAS	`p_gemm`
7	Solve Lyapunov eq.	Sparse E, A/	SpaRed	LR-ADI iter.
		dense B, K_j		

4 Parallel Solution (IV)

LR-ADI iteration				
Step	Operation	Matrices	Parallel lib.	Routine
1	Compute V_0	Sparse E, A/ dense $B, K, \hat{C}$	SpaRed	SMW formula
3	Compute V_{l+1}	Sparse E, A/ dense B, K, V_l	SpaRed	SMW formula

4 Parallel Solution (V)

SMW formula				
Step	Operation	Matrices	Parallel lib.	Routine
1	Solve $\bar{A}^{-T}W$	Sparse $\bar{A}$/	MUMPS or	`_mumps_c`
		dense W	ScaLAPACK	`p_gbsv`
2	Solve $\bar{A}^{-T}K$	Sparse $\bar{A}$/	MUMPS or	`_mumps_c`
		dense K	ScaLAPACK	`p_gbsv`
3	Compute $I_m - B^T T$	All dense	PBLAS	`p_gemm`
4	Solve TF^{-1}	All dense	ScaLAPACK	`p_gesv`
5	Compute $V + T(B^T V)$	All dense	PBLAS	`p_gemm`$\times 2$

Experimental environment

- *ra:* cluster with 34 nodes
- Node:
  - Intel Xeon Processor@2.4 GHz
  - 1GByte RAM, 512 KByte L2 cache
- Myrinet network (2Gbps)
- Software:
  - Intel Compilers v.9
  - MPI v.1.2.5. dev: `ch_gm`

Matrices examples

- **Example 1:**
  - Origin: Finite difference discretization.
  - Equation: 2-D heat equation.
  - Size: variable (i.e. $n = 160,000$).
  - SISO system, $m = p = 1$.
- **Example 2:**
  - Application: Manufacturing steel profiles with moderate temperature gradients.
  - Equation: 2-D heat equation.
  - Size: $n = 79,841$.
  - $m = 7, p = 6$.

5 Experimental Results (III)

Execution times of the ARE solver

- 16 processors (4×4 ScaLAPACK grid, block size nb : 32).

	n	#iter. Newton	#shifts LR-ADI	Avg. #iter. LR-ADI	r	Ex. time
Example 1	160,000	11	10	80	160	1h 25m 5s
Example 2	79,841	5	20	100	1300	1h 4m 30s

- The main factor is the memory space. Virtual memory needed with less processors.
- The parallel performance strongly depends on
  - the sparse system solver,
  - the sparsity pattern of the matrix pair (A, E).

6 Conclusions

- Large sparse AREs $O(10^5)$ can be solved,
  - Exploiting the sparsity of the matrices,
  - Using parallel architectures to increase,
    - the computational power,
    - the memory size.
- A new parallel solver has been implemented: `SpaRed` . Main features:
  - Newton's iteration exploiting the low rank of the solution.
  - LR-ADI iteration to solve a Lyapunov equation on each Newton's iteration.
  - SMV formula to exploit the sparsity of the matrices on the LR-ADI iteration.
  - Combination and explotation of different parallel libraries: PBLAS, ScaLAPACK, MUMPS, ...