Passive Reduced-Order Modelling Using Positive-Real Balancing

Peter Benner

Mathematik in Industrie und Technik
Fakultät für Mathematik

TECHNISCHE UNIVERSITÄT CHEMNITZ

DFG research center Berlin
mathematics for key technologies

benner@mathematik.tu-chemnitz.de

BIRS Workshop on Model Reduction and Matrix Methods
Banff, April 3–8, 2004

Something to keep in mind . . .

1. There is no exact balanced truncation — unless you can do it with symolic computation.

Something to keep in mind . . .

1. There is no exact balanced truncation — unless you can do it with symolic computation.

2. There are different implementations of balanced truncation:

   - different Lyapunov solvers for computing Gramians (Bartels-Stewart, sign function, ADI, . . .);
   - $\mathcal{O}(n^3)$ SVD (using solution itself or Cholesky factor) versus $\mathcal{O}(n)$ SVD (using full-numerical-rank factors).

Something to keep in mind . . .

1. There is no exact balanced truncation — unless you can do it with symolic computation.

2. There are different implementations of balanced truncation:

   - different Lyapunov solvers for computing Gramians (Bartels-Stewart, sign function, ADI, . . .);
   - $\mathcal{O}(n^3)$ SVD (using solution itself or Cholesky factor) versus $\mathcal{O}(n)$ SVD (using full-numerical-rank factors).

3. There is no numerically backwards stable method to solve Lyapunov equations.

   $\Rightarrow$ all implementations are "approximate" BT.

Something to keep in mind . . .

1. There is no exact balanced truncation — unless you can do it with symolic computation.

2. There are different implementations of balanced truncation:

   - different Lyapunov solvers for computing Gramians (Bartels-Stewart, sign function, ADI, . . .);
   - $\mathcal{O}(n^3)$ SVD (using solution itself or Cholesky factor) versus $\mathcal{O}(n)$ SVD (using full-numerical-rank factors).

3. There is no numerically backward stable method to solve Lyapunov equations.

   $\Rightarrow$ all implementations are "approximate" BT.

4. Depending on the implementation, BT can be applied to systems of order (set n=10)

$\mathcal{O}(n^3)$	$\mathcal{O}(n^4)$	$\mathcal{O}(n^5)$	$\mathcal{O}(n^6)$
dense/serial	dense/parallel, sparse	sparse/serial–parallel	sparse/parallel

Something to keep in mind . . .

1. There is no exact balanced truncation — unless you can do it with symolic computation.

2. There are different implementations of balanced truncation:
   - different Lyapunov solvers for computing Gramians (Bartels-Stewart, sign function, ADI, . . .);
   - $\mathcal{O}(n^3)$ SVD (using solution itself or Cholesky factor) versus $\mathcal{O}(n)$ SVD (using full-numerical-rank factors).

3. There is no numerically backward stable method to solve Lyapunov equations.

   $\Rightarrow$ all implementations are "approximate" BT.

4. Depending on the implementation, BT can be applied to systems of order (set n=10)

$\mathcal{O}(n^3)$	$\mathcal{O}(n^4)$	$\mathcal{O}(n^5)$	$\mathcal{O}(n^6)$
dense/serial	dense/parallel, sparse	sparse/serial–parallel	sparse/parallel
SLICOT	PLiCMR, LyaPack	LyaPack, SpaRed	SpaRed

Outline

- Passive systems
- Positive-real balanced truncation
- Implementations
  - Newton's method for algebraic Riccati equations
  - parallel implementation based on matrix sign function
  - sparse implementation based on ADI method
- Numerical examples
- Conclusions and outlook

Linear Systems

Linear time-invariant systems in generalized state-space form:

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \quad t > 0, \qquad x(0) = x_0, \\ y(t) &= Cx(t) + Du(t), \end{aligned}$$

- n generalized states, i.e., $x(t) \in \mathbb{R}^n$ (n is the degree of the system);
- m inputs, i.e., $u(t) \in \mathbb{R}^m$;
- m outputs, i.e., $y(t) \in \mathbb{R}^m$;
- $A - \lambda E$ stable, i.e., $\lambda(A, E) \subset \mathbb{C}^- \cup \{\infty\}$ $\Rightarrow$ system is stable.
- E nonsingular.

Corresponding transfer function:

$$G(s) = C(sE - A)^{-1}B + D$$

Passive Systems

<u>Definition:</u>
A linear system is passive if $\int_{-\infty}^{t} u(\tau)^T y(\tau)\, d\tau \geq 0\ \forall t \in \mathbb{R},\ \forall u \in L_2(\mathbb{R}, \mathbb{R}^m)$.

"The system cannot generate energy."

system is passive $\Longleftrightarrow$ its transfer function is positive real

<u>Definition:</u>
A *real*, rational matrix-valued function $G : \mathbb{C} \to \bar{\mathbb{C}}^{m \times m}$ is positive real if

1. G is analytic in $\mathbb{C}^+ := \{s \in \mathbb{C} \mid \mathrm{Re}(s) > 0\}$,
2. $G(s) + G^T(\bar{s}) \geq 0$ for all $s \in \mathbb{C}^+$.

Motivation

For passive linear system, compute passive reduced-order system with computable global error bound.

- Task often encountered in circuit simulation, VLSI chip design.
- Padé-type methods in general do not preserve passivity, post-processing necessary [Bai/(Feldmann)/Freund '98,'01].
  - PRIMA [Odabasioglu et al.'96,'97] preserevs passivity for interconnect models, basically Arnoldi process.
  - SyPVL preserves passivity for RLC circuits [Feldmann/Freund '96,'97].
  - LR-ADI/dominant subspace approximation can preserve passivity [Li/White '01].
- No computable error bounds available for Krylov-type methods.

Here: alternative approach for general passiv systems based on positive-real balancing.

Positive-Real Balancing

Set

$$R := D + D^T \qquad \text{(positive real } \Rightarrow R \geq 0, \quad \text{here assume } R > 0\text{)},$$
$$\hat{A} := A - BR^{-1}C,$$

and consider the two dual positive-real algebraic Riccati equations (PAREs)

$$\begin{aligned} 0 &= \hat{A}PE^T + EP\hat{A}^T + EPC^TR^{-1}CPE^T + BR^{-1}B^T, \\ 0 &= \hat{A}^TQE + E^TQ\hat{A} + EQBR^{-1}B^TQE + C^TR^{-1}C. \end{aligned}$$

Let $P_{\min}, Q_{\min} > 0$ be the minimal solutions, then the system is positive real balanced iff

$$P_{\min} = E^T Q_{\min} E = \operatorname{diag}(\sigma_1, \ldots, \sigma_n).$$

$P_{\min}, E^T Q_{\min} E$ are called positive-real Gramians.

Note: $P_{\min}, Q_{\min} > 0$ are the stabilizing solutions of the PAREs.

Positive-Real Balanced Truncation I

1. Find positive-real balancing equivalence transformation

$$(E,A,B,C,D) \quad \rightarrow \quad (TES^{-1}, TAS^{-1}, TB, CS^{-1}, D) =: (\tilde{E},\tilde{A},\tilde{B},\tilde{C},\tilde{D}),$$

$$P_{\min} = E^T Q_{\min} E \quad = \quad \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix}, \qquad \begin{array}{rcl} \Sigma_1 & = & \mathrm{diag}(\sigma_1,\ldots,\sigma_r), \\ \Sigma_2 & = & \mathrm{diag}(\sigma_{r+1},\ldots,\sigma_n), \end{array}$$

$$\sigma_1 \geq \sigma_2 \geq \quad \ldots \quad \geq \sigma_r > \sigma_{r+1} \geq \sigma_{r+2} \geq \ldots \geq \sigma_n > 0.$$

2. Truncate the states $\tilde{x}_{r+1},\ldots,\tilde{x}_n$ of the balanced system

$$(\tilde{E},\tilde{A},\tilde{B},\tilde{C},\tilde{D}) = \left(\begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix}, \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} C_1 & C_2 \end{bmatrix}, D \right),$$

i.e., the reduced-order model and transfer function are

$$\begin{array}{rcl} (E_r, A_r, B_r, C_r, D_r) & := & (E_{11}, A_{11}, B_{11}, C_{11}, D) \\ G_r(s) & = & C_r(sE_r - A_r)^{-1}B_r + D_r \end{array}$$

Positive-Real Balanced Truncation II

Properties:

- Reduced-order model is passive ($\Rightarrow$ stable),
- relative error "bound" if $\|G\|_{H_\infty} \gg \|D\|_2$:

$$\frac{\|G - G_r\|_{H_\infty}}{\|G\|_{H_\infty}} \approx \frac{\|G - G_r\|_{H_\infty}}{\|G + D^T\|_{H_\infty}} \leq 2\|R\|_2^2 \|G_r + D^T\|_{H_\infty} \sum_{k=r+1}^{n} \sigma_k$$

Computation:

- Analogous to balanced truncation: let $P_{\min} = S^T S$, $E^T Q_{\min} E = R^T R$, transformation matrices and reduced-order model are computed from SVD of SR^T.
- Often, $P_{\min}, Q_{\min}$ have low numerical rank
  $\Longrightarrow$ use (numerical) full-rank factors rather than Cholesky factors
  $\Longrightarrow$ cheap SVD, cost $\sim \mathcal{O}(n)$ instead of $\mathcal{O}(n^3)$
  $\Longrightarrow$ need method to compute factored solutions of PAREs.

Newton's Method for AREs

Consider algebraic Riccati equation (ARE)

$$0 = \mathcal{R}(Q) = C^T C + A^T Q + QA - QBB^T Q.$$

Frechét derivative of $\mathcal{R}$ at Q:

$$\mathcal{R}_Q^{'} : Z \to (A - BB^T Q)^T Z + Z(A - BB^T Q)$$

Newton-Kantorovich method:

$$Q_{j+1} = Q_j - \left(\mathcal{R}_{Q_j}^{'}\right)^{-1} \mathcal{R}(Q_j), \quad j = 0, 1, 2, \ldots$$

$\Longrightarrow$ Newton's method for AREs

[Kleinman '68, Mehrmann '91, Lancaster/Rodman '95]

```
FOR j = 0, 1, 2, ...
    A_j ← A − B B^T Q_j =: A − B K_j.
    Solve Lyapunov equation    A_j^T N_j + N_j A_j = −R(Q_j).
    Q_{j+1} ← Q_j + N_j.
END FOR j
```

$\texttt{FOR}\ j = 0,\, 1,\, 2,\, \ldots$

$\quad A_j \leftarrow A - BB^T Q_j =: A - BK_j.$

$\quad$ Solve Lyapunov equation $\quad A_j^T N_j + N_j A_j = -\mathcal{R}(Q_j).$

$\quad Q_{j+1} \leftarrow Q_j + N_j.$

$\texttt{END FOR}\ j$

- Convergence:
  - A_j is stable $\forall\ j \geq 0$.
  - $0 \leq Q_\infty \leq \ldots \leq Q_{j+1} \leq Q_j \leq \ldots \leq Q_1$.
  - $\lim_{j\to\infty} \|\mathcal{R}(Q_j)\|_F = 0$,
  - $\lim_{j\to\infty} Q_j = Q_\infty \geq 0$ (quadratically),
  - acceleration of (initially slow) convergence possible using line searches.

Need efficient Lyapunov solver, depending on data structures and computing full-numerical-rank factors.

Factored Newton Iteration

Rewrite Newton's method for AREs [*Kleinman '68*]

$$A_j^T N_j + N_j A_j = -\mathcal{R}(Q_j)$$

$$\Updownarrow$$

$$A_j^T \underbrace{(Q_j + N_j)}_{=Q_{j+1}} + \underbrace{(Q_j + N_j)}_{=Q_{j+1}} A_j = \underbrace{-C^T C - Q_j B B^T Q_j}_{=:-W_j W_j^T}$$

Let $Q_j = Y_j Y_j^T$ for $\mathrm{rank}\,(Y_j) \ll n$:

$$A_j^T \left(Y_{j+1} Y_{j+1}^T\right) + \left(Y_{j+1} Y_{j+1}^T\right) A_j = -W_j W_j^T$$

Parallelization based on Matrix Sign Function

Want method for solving Lyapunov equations which computes full-rank factor Y_{j+1} directly (without ever forming Q_{j+1}).

Consider

$$F^T X + XF + E = 0.$$

Newton's method for the matrix sign function yields [Roberts '71]:

$$\begin{array}{l} F_0 \leftarrow F, \quad E_0 \leftarrow E, \\ \texttt{for } j = 0, 1, 2, \ldots \\ \qquad F_{k+1} \leftarrow \frac{1}{2c_k}\left(F_k + c_k^2 F_k^{-1}\right), \\ \qquad E_{k+1} \leftarrow \frac{1}{2c_k}\left(E_k + c_k^2 F_k^{-T} E_k F_k^{-1}\right). \end{array} \quad \Longrightarrow \quad X_* = \tfrac{1}{2}\lim_{j\to\infty} E_k$$

Here: $E = B^T B$ or $C^T C$, $F = A^T$ or A, want factor R of solution.

Solving Lyapunov Equations for Full-Rank Factor

Consider now

$$A^T X + XA + C^T C = 0.$$

For $E_0 = C_0^T C_0 := C^T C$, $C \in \mathbb{R}^{p\times n}$ obtain

$$E_{k+1} = \frac{1}{2c_k}\left(E_k + c_k^2 A_k^{-T} E_k A_k^{-1}\right) = \frac{1}{2c_k}\begin{bmatrix} C_k \\ c_k C_k A_k^{-1} \end{bmatrix}^T \begin{bmatrix} C_k \\ c_k C_k A_k^{-1} \end{bmatrix}.$$

$\Longrightarrow$ Re-write E_k–iteration:

$$C_0 := C, \quad C_{k+1} := \frac{1}{\sqrt{2c_k}}\begin{bmatrix} C_k \\ c_k C_k A_k^{-1} \end{bmatrix}, \quad \Longrightarrow \quad \frac{1}{\sqrt{2}} \lim_{k\to\infty} C_k = R_*$$

Problem: $C_k \in \mathbb{R}^{p_k \times n} \quad \Longrightarrow \quad C_{k+1} \in \mathbb{R}^{2p_k\times n}$

Cure: Limit work space by computing rank-revealing QR factorization in each step.

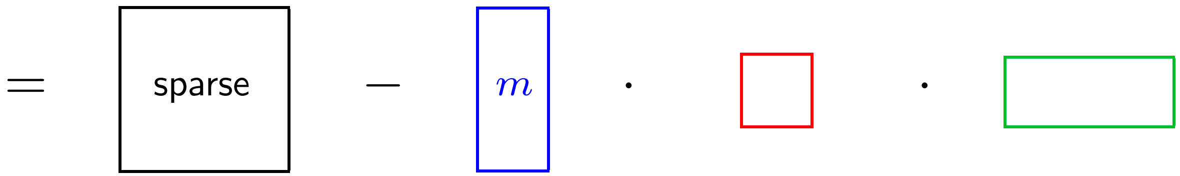
Sparse Implementation

Need method for solving Lyapunov equations

$$A_j^T \left(Y_{j+1}Y_{j+1}^T\right) + \left(Y_{j+1}Y_{j+1}^T\right) A_j = -W_j W_j^T, \quad \text{where} \quad W_j = [C^T, Y_j(Y_j^T B)],$$

- which computes Y_{j+1} directly (without ever forming X_{j+1}), and
- uses the structure of A_j,

$$A_j = A - BK_j \quad = \quad A \quad - \quad B \quad \cdot \quad (B^T Y_j) \quad \cdot \quad Y_j^T,$$



Note: as $m \ll n$, we can efficiently apply Sherman-Morrison-Woodbury formula

$$(A - BK_j)^{-1} \quad = \quad (I_n + A^{-1}B(\underbrace{I_m - K_j A^{-1}B}_{m\times m})^{-1}K_j)A^{-1}$$

$$= \quad (I_n + \hat{B}(I_m - K_j\hat{B})^{-1}K_j)A^{-1}$$

ADI Method for Lyapunov Equations

- For $A \in \mathbb{R}^{n\times n}$ stable, $W \in \mathbb{R}^{n\times w}$ $(w \ll n)$, consider Lyapunov equation

$$A^T X + XA = -WW^T.$$

- ADI Iteration: [*Wachspress '88*]

$$\begin{aligned}(A^T + p_k I)X_{(j-1)/2} &= -WW^T - X_{k-1}(A - p_k I)\\ (A^T + \overline{p_k} I){X_k}^T &= -WW^T - X_{(j-1)/2}(A - \overline{p_k} I)\end{aligned}$$

with parameters $p_k \in \mathbb{C}^-$ and $p_{k+1} = \overline{p_k}$ if $p_k \notin \mathbb{R}$.

- For $X_0 = 0$ and proper choice of p_k: $\lim_{k\to\infty} X_k = X$ superlinear.
- Re-formulation using $X_k = Z_k Z_k^T$ yields iteration for Z_k...

Factored ADI Iteration

[Penzl '97, Li/Wang/White '99, B./Li/Penzl]

Set $X_k = Z_k Z_k^T$, some algebraic manipulations $\Longrightarrow$

$$
\begin{array}{l}
V_1 \quad \leftarrow \quad \sqrt{-2\mathrm{Re}\,(p_1)}(A^T + p_1 I)^{-1} W, \qquad Z_1 \;\leftarrow\; V_1 \\
\texttt{FOR } j = 2, 3, \ldots \\
\qquad V_k \leftarrow \sqrt{\frac{\mathrm{Re}\,(p_k)}{\mathrm{Re}\,(p_{k-1})}} \left(I - (p_k + \overline{p_{k-1}})(A^T + p_k I)^{-1}\right) V_{k-1}, \qquad Z_k \leftarrow \left[\; Z_{k-1} \quad V_k \;\right]
\end{array}
$$

$$\Downarrow$$

$$Z_{k_{\max}} = \left[\; V_1 \quad \ldots \quad V_{k_{\max}} \;\right]$$

where

$$V_k \; = \boxed{} \in \mathbb{C}^{n \times w}$$

and

$$Z_{k_{\max}} Z_{k_{\max}}^T \approx X$$

Note: Implementation in real arithmetic possible by combining two steps.

Newton-ADI for ARE

[*B./Li/Penzl*]

Solve Lyapunov equation

$$(A - BK_j)^T Y_{j+1} Y_{j+1}^T + Y_{j+1} Y_{j+1}^T (A - BK_j) = -W_j W_j^T$$

with factored ADI iteration.

$$\Downarrow$$

Obtain low-rank approximations $Z_0, Z_1, \ldots, Z_{k_{\max}}$ to Lyapunov solution.

$$\Downarrow$$

Newton's method with factored iterates $Q_{j+1} = Y_{j+1} Y_{j+1}^T = Z_{k_{\max}} Z_{k_{\max}}^T$.

$$\Downarrow$$

Factored solution of ARE: $Q \approx Y_{j_{\max}} Y_{j_{\max}}^T$.

Application to Positive-Real Balancing I

Factored Newton-ADI not directly applicable to PAREs!

Need modification: right-hand side of Lyapunov equation $\neq -W_j W_j^T$.

$$\mathrm{RHS} = -C^T R^{-1} C + Q_j B R^{-1} B^T Q_j =: -\tilde{C}\tilde{C}^T + \tilde{B}_j \tilde{B}_j^T$$

with $R > 0$.

Lyapunov equation is non-singular linear system of equations $\Longrightarrow$ write

$$A_j^T Q_{j+1} + Q_{j+1} A_j = -\tilde{C}^T \tilde{C} + \tilde{B}_j^T \tilde{B}_j$$

as

$$A_j^T (Q_{j+1} - Q_{j+1}) + (Q_{j+1} - Q_{j+1}) A_j = -W_j W_j^T - (-W_j W_j^T).$$

$\Longrightarrow$ Solve two Lyapunov equations per step with equal Lyapunov operator.

Application to Positive-Real Balancing II

To get factored Newton iterates need factor of

$$Q := Q_{j_{\max}} - Q_{j_{\max}} = Z_{j_{\max}} Z_{j_{\max}}^T - Z_{j_{\max}} Z_{j_{\max}}^T \geq 0.$$

Solution: similar to stochastic balanced truncation [*Varga/Fasol '93, Varga '00*]

Get full-rank factor from stable, nonnegative Lyapunov equation

$$A^T(Z^T Z) + (Z^T Z)A + C^T C = 0$$

where

$$C = R^{-\frac{1}{2}}C - R^{-\frac{1}{2}}B\,[Z_{j_{\max}}, Z_{j_{\max}}] \begin{bmatrix} Z_{j_{\max}} \\ -Z_{j_{\max}} \end{bmatrix}.$$

Complexity

$$\text{Cost} \sim \{ (\#\text{ Newton iterations for } P) + (\#\text{ Newton iterations for } Q) \} \cdot \text{mean}(\#\text{ ADI iterations}) \cdot (\text{cost for solving linear systems} + (\#\text{ of ADI parameters}) \cdot (\text{factorization of shifted systems}))$$

$$\sim \mathcal{O}(nnz(A))$$

$\Longrightarrow$ scales with cost for solving $Ax = b$.

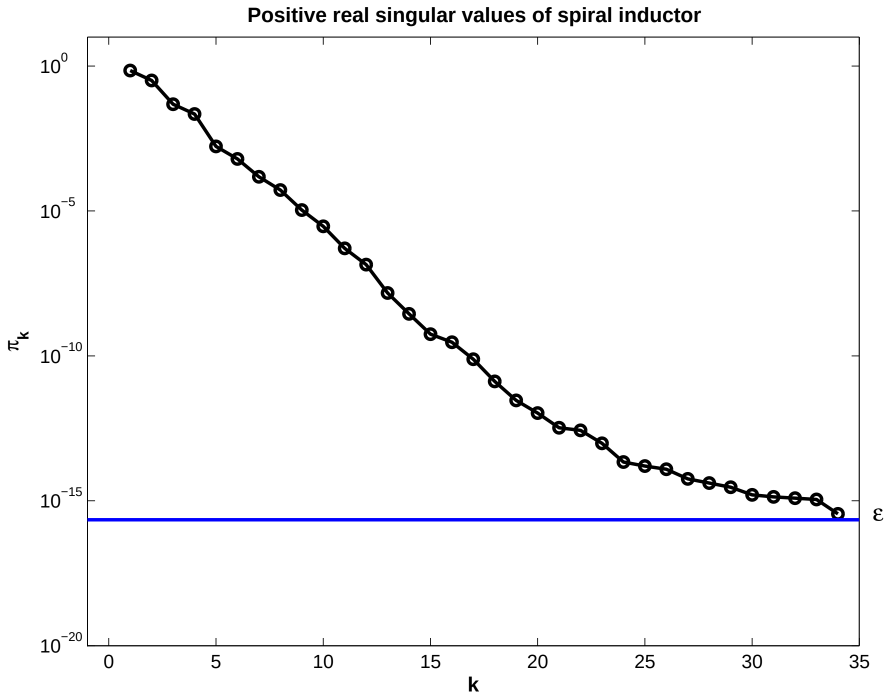
Numerical Example I

Model for inductance extraction of on-chip planar square spiral inductor suspended over a copper plane [Li/White '02].

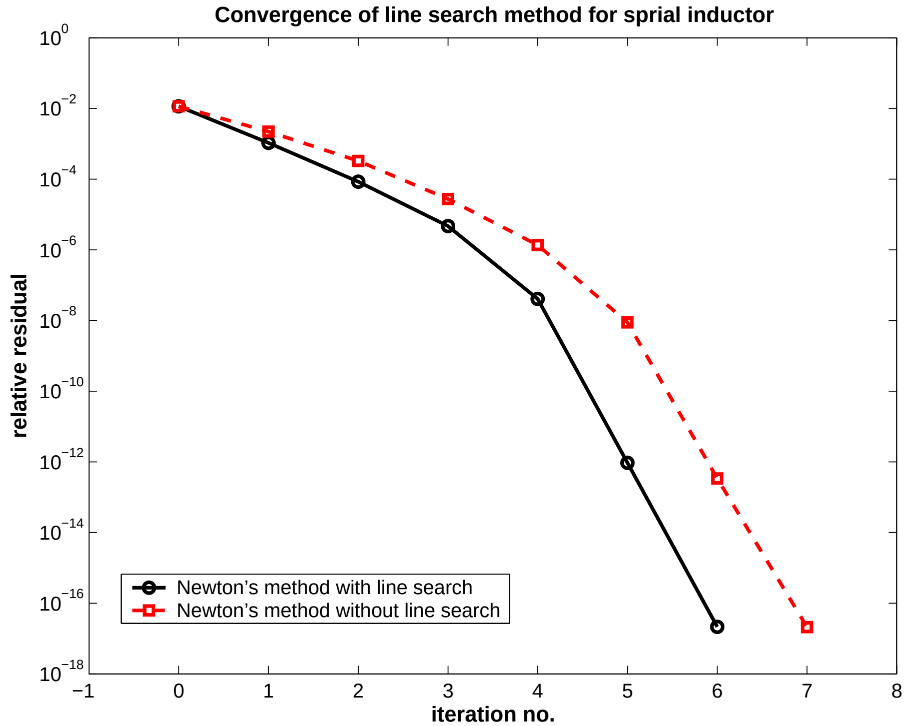
- System properties:
  - $A = A^T$, $C = B^T$
    $\Longrightarrow$ symmetric system;
  - $n = 500$, $m = p = 1$;
  - $D = 0$
    $\Rightarrow$ regularization using $D = 10^{-2}$.

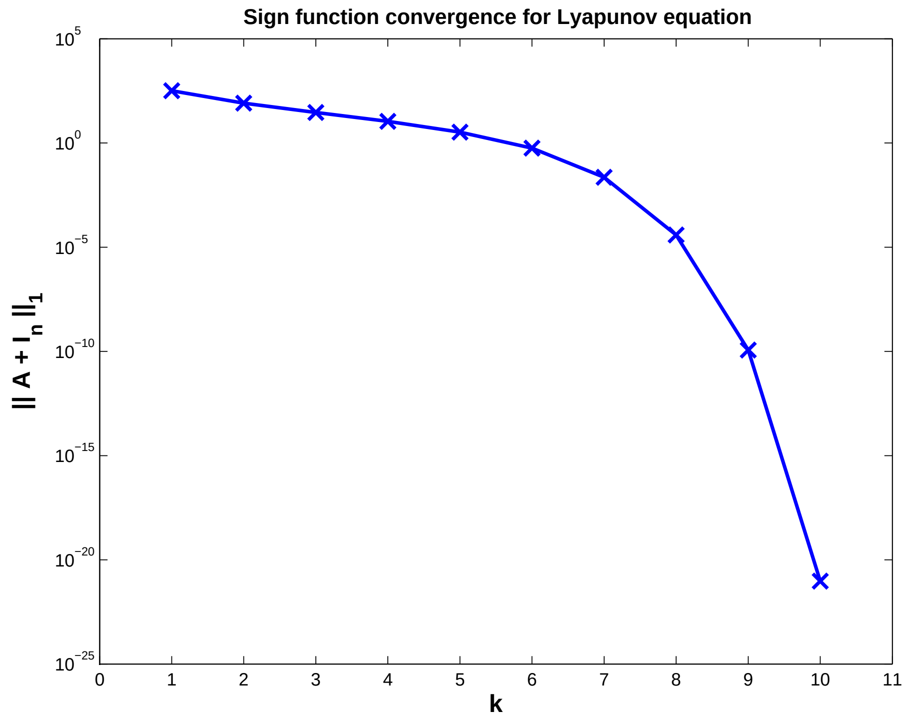
- Numerical rank of Gramians is 34.

- Reduced-order selection:
  $r = \max\{j \,|\, \sigma_j \geq 10^{-6}\}$
  $\Longrightarrow r = 11.$



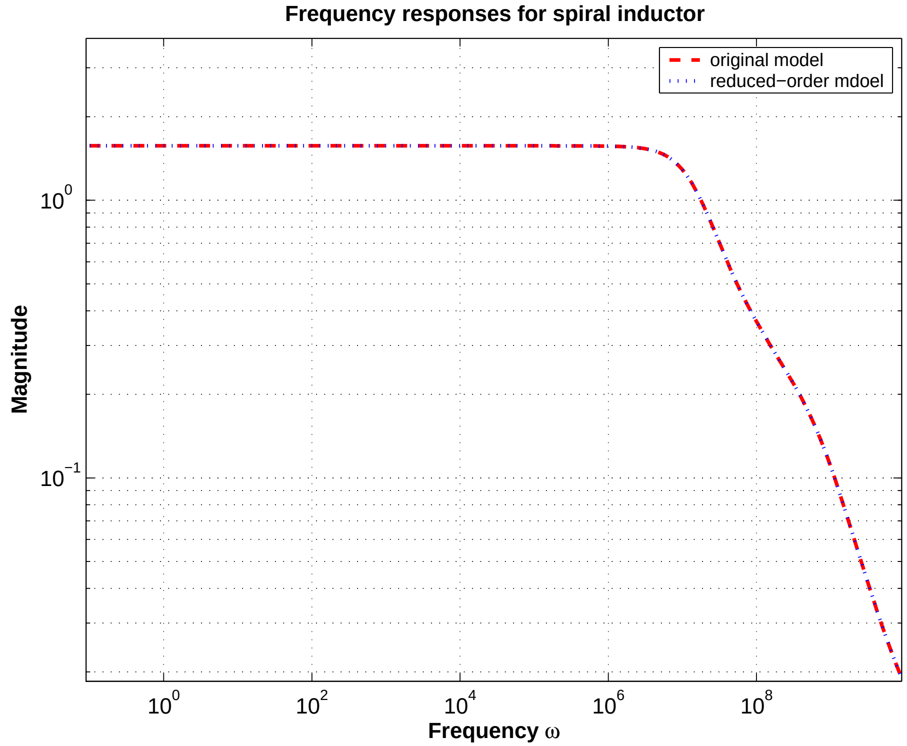
Example I: Convergence History

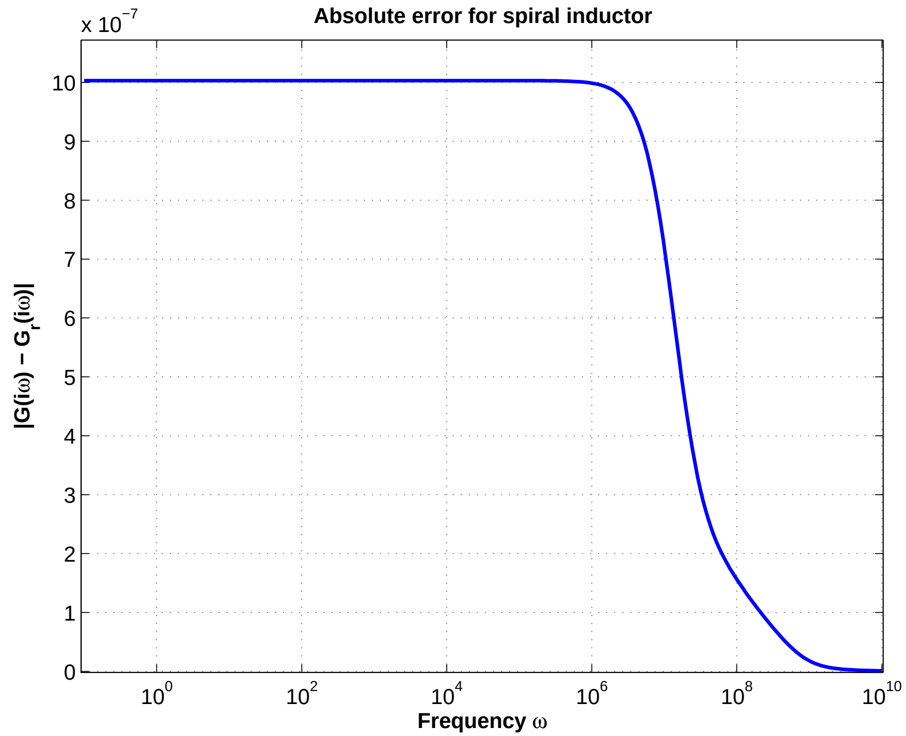




Note: Lyapunov residual $\approx 2 \cdot 10^{-5}$, for Bartels-Stewart $\approx 8 \cdot 10^{-5}$.

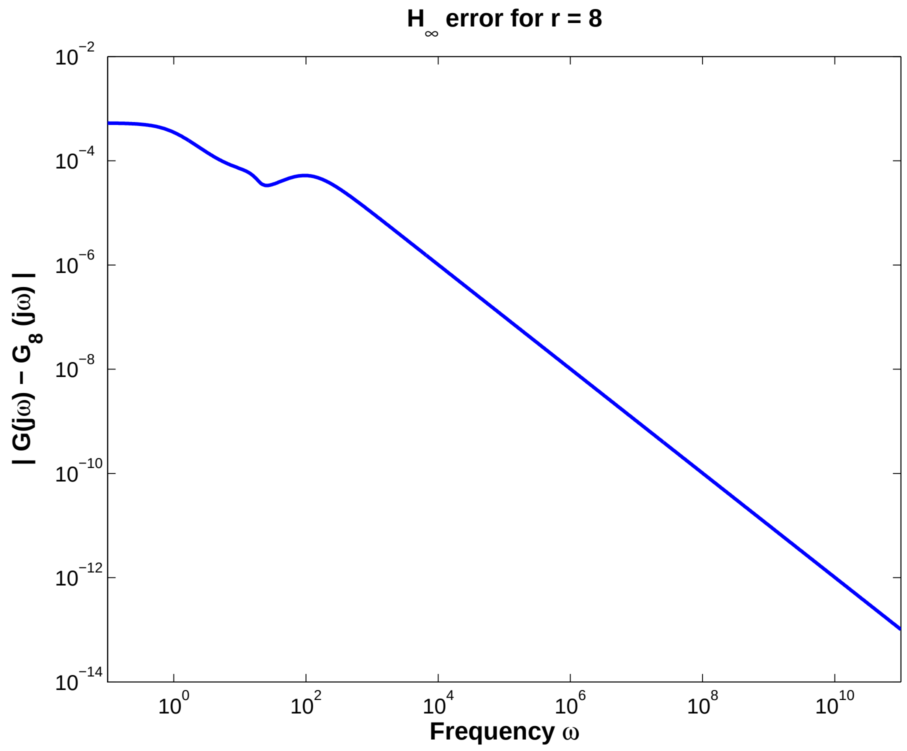
Example I: Error





Example II: RLC Network

- RLC circuit [Antoulas/Gugercin '03].
- $n = 1000$, $m = p = 1$.
- Numerical ranks of Gramians are 90, 75.
- Reduced-order selection:
  $r = \max\{j \,|\, \sigma_j/\sigma_1 \geq 10^{-4}\}$
  $\Longrightarrow r = 8$.
- 6 Newton iterations, 9 sign iterations each.
- total time in Matlab: $\approx$ 15 min.



Conclusions

- Pros
  - Guaranteed passive reduced-order models.
  - Reduced-order models more accurate than models computed via moment matching/PVL for same order (expected...).
  - Global error bound at the horizon (exists for slightly modified system, see [*Antoulas/Gugercin '03*]).
  - Applicable to fairly large models, even in MATLAB.
- Cons
  - More expensive than moment matching/PVL regarding computation time and memory requirements.
  - Descriptor case not yet treatable.
- Parallel implementation based on sign function for software library PLiCMR available.
- Implementation for sparse systems based on ADI in progress.

Open Problems

- Computable global error bound based on positive-real singular values.
- Positive-real balancing for descriptor systems.
- How far can we go with dense parallel implementations?

  So far, $n = 10,000$ on a small cluster (25 nodes).
- Efficient re-use of parameters in ADI iteration.

More on Model Reduction

The Oberwolfach Model Reduction Benchmark Collection:

- Initiative to have real-word test examples for model reduction, started at Oberwolfach meeting November 2003.
- Linear–nonlinear, first–second order, combinations thereof.
- Submission procedure with peer review.
- Requires short description and data in matrix-market format.
- Maintained and administrated at IMTEK (Institute for Microsystems Technology), simulation group (Prof. Jan Korvink), University of Freiburg, Germany.

  `http://www.imtek.uni-freiburg.de/simulation/benchmark/`

Upcoming 200?:

- Special issue Order Reduction of Large-Scale Systems of LINEAR ALGEBRA AND ITS APPLICATIONS.
- Proceedings of Oberwolfach Model Reduction Workshop, LNCSE, Springer.