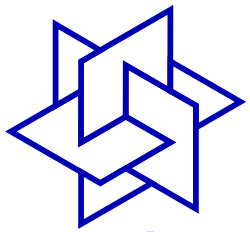
Model Reduction Algorithms Using Spectral Projection Methods

Peter Benner

Fakultät für Mathematik
TU Chemnitz, Germany



DFG research center Berlin
mathematics for key technologies

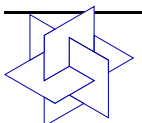
benner@mathematik.tu-chemnitz.de

Dimension Reduction of Large-Scale Systems
Oberwolfach, October 19–24, 2003

Thanks to: Enrique and Gregorio Quintana-Ortí, Rafael Mayo, Ulrike Baur

Outline

- Linear autonomous systems
- Spectral projection methods — the sign function method
- Model reduction
- Balanced truncation
- Stochastic truncation
- Model reduction for unstable systems
- Large-scale problems
  - $\mathcal{H}$-matrix based implementation
  - parallelization
- Numerical examples
- Conclusions



Linear Systems

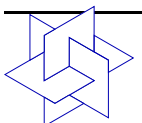
Linear autonomous (time-invariant) systems:

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

- n state-space variables, i.e., $x(t) \in \mathbb{R}^n$ (n is the degree of the system);
- m inputs, i.e., $u(t) \in \mathbb{R}^m$;
- p outputs, i.e., $y(t) \in \mathbb{R}^p$;
- A stable, i.e., $\lambda(A) \subset \mathbb{C}^- \;\Rightarrow$ system is stable.

Corresponding transfer function:

$$G(s) = C(sI_n - A)^{-1}B + D \equiv \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right].$$



Realizations

Laplace transform: $sx(s) - x(0) = Ax(s) + Bu(s), \quad y(s) = Cx(s) + Du(s).$

$$x(0) = 0 \implies y(s) = \left(C(sI_n - A)^{-1}B + D\right) u(s) = G(s)u(s).$$

$(A, B, C, D) \equiv \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ is a realization of the system G.

(Realizations are not unique! m, p are fixed, n is variable!)

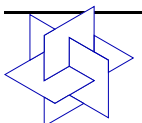
Minimal realization: find minimal degree $\hat{n}$ (= McMillan degree), $\hat{A}, \hat{B}, \hat{C}, \hat{D}$ with

$$G(s) = \hat{C}(sI_{\hat{n}} - \hat{A})^{-1}\hat{B} + \hat{D}.$$

Minimal realization is not unique: for any state-space transformation

$$\mathcal{T} : x \to Tx, \quad (A, B, C, D) \to (TAT^{-1}, TB, CT^{-1}, D)$$

obtain new realization of the system: $D + (CT^{-1})(sI - TAT^{-1})^{-1}(TB) = G(s)$.



Balanced Realization

- Controllability and observability Gramians of G solve Lyapunov equations

$$\boxed{AP + PA^T + BB^T = 0, \qquad A^TQ + QA + C^TC = 0.}$$

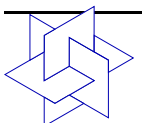
- (A, B, C, D) is a balanced realization of G iff $P = Q = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix}$.

- $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$ are the Hankel singular values (HSV) of the system (invariant under state-space transfromation).

- (A, B, C, D) minimal $\Longrightarrow \exists$ balancing state-space transformation.

  (A, B, C, D) non-minimal $\quad\Longrightarrow\quad P_{1:\hat{n},1:\hat{n}} = Q_{1:\hat{n},1:\hat{n}} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_{\hat{n}} \end{bmatrix}$,

  HSV are $\{\sigma_1, \ldots, \sigma_{\hat{n}}, 0, \ldots, 0\}$.



Spectral Projection Methods

For $Z \in \mathbb{R}^{n\times n}$ with $\lambda(Z) = \Lambda_1 \cup \Lambda_2$, $\Lambda_1 \cap \Lambda_2 = \emptyset$, let $\mathcal{P}$ be a (skew) projector onto the right Z-invariant subspace corresponding to Λ_1.

1. $\operatorname{rank}(\mathcal{P}) = |\Lambda_1| := k, \quad \operatorname{range}(\mathcal{P}) = \operatorname{range}(A\mathcal{P})$.

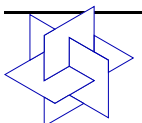
2. Let

$$\mathcal{P} = QRP, \quad R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix} = \left[\begin{array}{cc} \blacktriangleleft & \square \\ & \end{array}\right], \qquad R_{11} \in \mathbb{R}^{k\times k},$$

where P is a permutation matrix. Then obtain block-triangular form

$$\tilde{Z} := Q^T Z Q = \begin{bmatrix} Z_{11} & Z_{12} \\ 0 & Z_{22} \end{bmatrix},$$

where $\lambda(Z_{11}) = \Lambda_1$, $\lambda(Z_{22}) = \Lambda_2$.



The Sign Function Method

[*Roberts '71*]

For $Z \in \mathbb{R}^{n\times n}$ with $\lambda(Z) \cap \imath\mathbb{R} = \emptyset$ and Jordan canonical form

$$Z = S^{-1}\begin{bmatrix} J^+ & 0 \\ 0 & J^- \end{bmatrix} S \qquad \Longrightarrow \qquad \boxed{\operatorname{sign}(Z) := S\begin{bmatrix} I_k & 0 \\ 0 & -I_{n-k} \end{bmatrix} S^{-1}.}$$

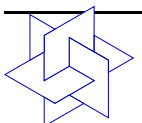
($J^{\pm}$ = Jordan blocks corresponding to $\lambda(Z) \cap \mathbb{C}^{\pm}$)

$\operatorname{sign}(Z)$ is root of $I_n \Longrightarrow$ use Newton's method to compute it:

$$Z_0 \leftarrow Z, \qquad Z_{j+1} \leftarrow \frac{1}{2}\left(c_j Z_j + \frac{1}{c_j} Z_j^{-1}\right), \qquad j = 1, 2, \ldots$$

$\Longrightarrow$ $\boxed{\operatorname{sign}(Z) = \lim_{j\to\infty} Z_j.}$

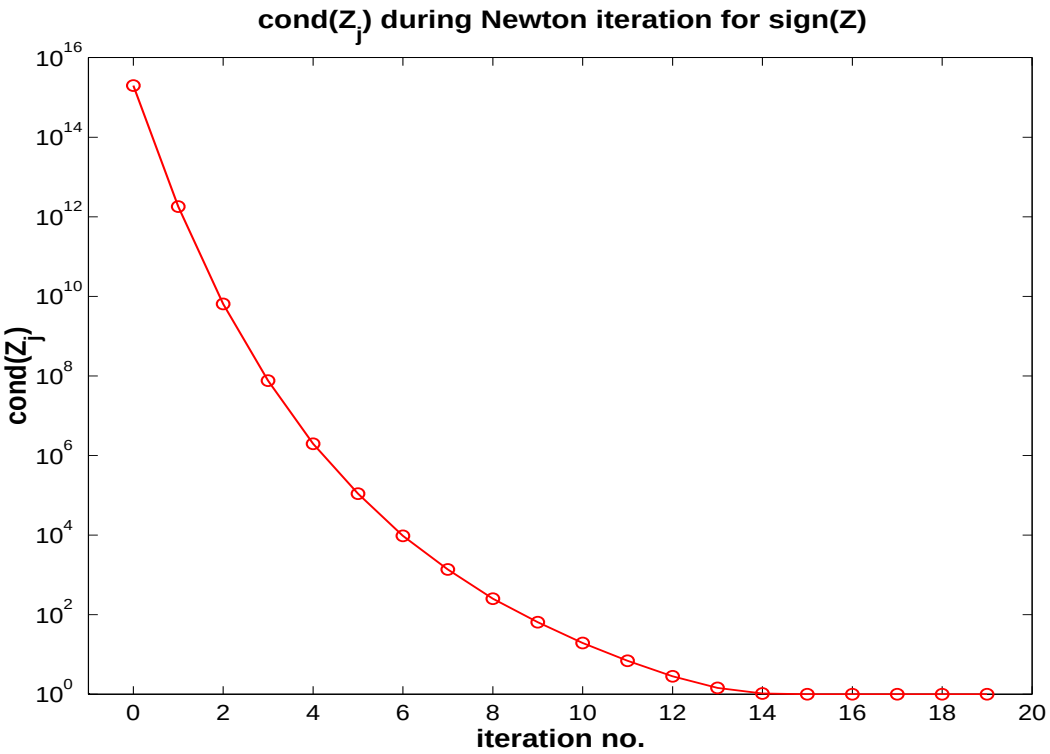
($c_j > 0$ is scaling parameter for convergence acceleration and rounding error minimization.)

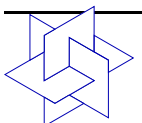


Properties of the Sign Function Method

- $\frac{1}{2}(I_n - \text{sign}(Z))$ is skew projector onto stable Z–invariant subspace.
- Sign function undefined if Z has purely imaginary eigenvalues $\Longrightarrow$ problems for eigenvalues close to imaginary axis.

- Usually, computed invariant subspaces are as accurate as their conditioning admits. [*Byers/He/Mehrmann 1997*]
- Block-triangular form often better conditioned than computation of Schur form. $\Longrightarrow$ Sign function often more accurate than computations based on QR/QZ algorithms.
- Here: $\text{cond}(\text{sign}(Z)) = 1$ as Z stable or anti-stable, hence computation of $\text{sign}(Z)$ itself is well-conditioned problem!





Model Reduction

Given

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

find reduced model

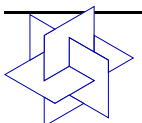
$$\begin{aligned} \dot{\tilde{x}}(t) &= \tilde{A}\tilde{x}(t) + \tilde{B}u(t), \\ \tilde{y}(t) &= \tilde{C}\tilde{x}(t) + \tilde{D}u(t), \end{aligned}$$

of degree $\ell \ll n$ with $\tilde{y}(t) \in \mathbb{R}^p$ and output error

$$y - \tilde{y} = Gu - \tilde{G}u = (G - \tilde{G})u$$

such that

$\|y - \tilde{y}\|$ "small" or, respectively, $\|G - \tilde{G}\|$ "small".



Truncation Methods

- For arbitrary state-space transformation $T \in \mathbb{R}^{n\times n}$:

$$TAT^{-1} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad TB = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad CT^{-1} = \begin{bmatrix} C_1 & C_2 \end{bmatrix}.$$

  with $A_{11} \in \mathbb{R}^{\ell\times\ell}, \ldots$.

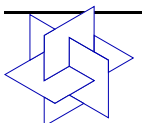
- Partition $T = \begin{bmatrix} T_l \\ W_l \end{bmatrix}$, $T_l \in \mathbb{R}^{\ell\times n}$, $T^{-1} = \begin{bmatrix} T_r & W_r \end{bmatrix}$, $T_r \in \mathbb{R}^{n\times\ell}$.

- Reduced-order model: $\left[\begin{array}{c|c} \tilde{A} & \tilde{B} \\ \hline \tilde{C} & \tilde{D} \end{array}\right] = \left[\begin{array}{c|c} A_{11} & B_1 \\ \hline C_1 & D \end{array}\right] = \left[\begin{array}{c|c} T_lAT_r & T_lB \\ \hline CT_r & D \end{array}\right]$

  with projected dynamics: $\tilde{x} = T_rT_lx$.

- Choice of T, ℓ such that $\|y - \tilde{y}\|$ is "small"!

  Note: $\lim_{\omega\to\infty}(G(\imath\omega) - \tilde{G}(\imath\omega)) = D - D = 0$.



Absolute Error Methods

Recall: want reduced-order model

$$\begin{aligned} \dot{\tilde{x}}(t) &= \tilde{A}\tilde{x}(t) + \tilde{B}u(t), \\ \tilde{y}(t) &= \tilde{C}\tilde{x}(t) + \tilde{D}u(t), \end{aligned}$$

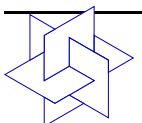
of degree $\ell \ll n$ with small absolute error.

Note: for $\|G\|_\infty := \operatorname{ess\,sup}_{\omega\in\mathbb{R}} \sigma_{\max}(G(\imath\omega))$, we have

$$\|G(s) - \tilde{G}(s)\|_\infty = \sup_{u\in\mathcal{H}_2} \frac{\|(G-\tilde{G})u\|_2}{\|u\|_2} = \sup_{u\in\mathcal{H}_2} \frac{\|y-\tilde{y}\|_2}{\|u\|_2}$$

Hence, $\|y - \tilde{y}\|_2 \leq \|G - \tilde{G}\|_\infty \|u\|_2$.

Consequence of Paley-Wiener Theorem: $\|y - \tilde{y}\|_2 \equiv \|y - \tilde{y}\|_{\mathcal{H}_2} \equiv \|y - \tilde{y}\|_{\mathcal{L}_2[0,\infty)}$



Balanced Truncation

For balanced realization $G(s) \equiv \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ with $P = Q =: \begin{bmatrix} \tilde{\Sigma} & \\ & \Sigma_2 \end{bmatrix}$ the reduced-order model

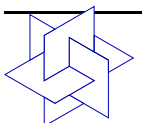
$$\tilde{G}(s) \equiv \left[\begin{array}{c|c} \tilde{A} & \tilde{B} \\ \hline \tilde{C} & \tilde{D} \end{array}\right] \equiv \left[\begin{array}{c|c} A_{11} & B_1 \\ \hline C_1 & D \end{array}\right]$$

is balanced, minimal, stable. The Gramians are $\tilde{P} = \tilde{Q} = \tilde{\Sigma} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_\ell \end{bmatrix}$.

$\Longrightarrow$ Computable global error bound $\boxed{\|G - \tilde{G}\|_\infty \leq 2 \sum_{k=\ell+1}^{n} \sigma_k.}$

$\Longrightarrow$ adaptive choice of ℓ.

Balancing transformation often ill-conditioned. Remedy: compute T_l, T_r , such that $\tilde{G}$ is not balanced, but error bound holds! [*Safonov/Chiang 1989, Varga 1991*]



Balanced Truncation: SR method

[Heath/Laub/Paige/Ward '87, Tombs/Postlethwaite '87]

Gramians are spd $\Longrightarrow P = S^T S, \quad Q = R^T R$. For better numerical robustness, use S, R instead of P, Q:

$$\sigma\left(SR^T\right)^2 = \lambda(PQ), \quad \operatorname{cond}\left(SR^T\right) = \sqrt{\operatorname{cond}\left(PQ\right)}.$$

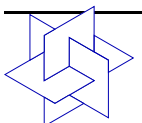
Note: $S^{-T}(PQ)S^T = (SR^T)(SR^T)^T = (U\Sigma V^T)(V\Sigma U^T) = U\Sigma^2 U^T$.

Compute balancing transformation using SVD:

$$SR^T = [U_1\, U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}, \qquad \Sigma_1 = \operatorname{diag}(\sigma_1, \ldots, \sigma_\ell)$$

$$\Downarrow$$

$$T_l = \Sigma_1^{-1/2} V_1^T R, \qquad T_r = S^T U_1 \Sigma_1^{-1/2}.$$



Computing the Factors S, R

Standard approach:

$S, R = \left[\begin{smallmatrix} \ddots & \\ & \ddots \end{smallmatrix}\right] \in \mathbb{R}^{n \times n}$ are Cholesky factors of P, Q.

Use Hammarling's method: solve Lyapunov equations

$$A(SS^T) + (SS^T)A^T + BB^T = 0, \qquad A^T(R^TR) + (R^TR)A + C^TC = 0$$

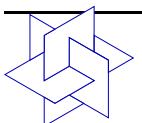
by reducing A to Schur form (QR algorithm) and solve resulting linear system for S, R by backsubstitution.

Approach here:

$S \in \mathbb{R}^{\operatorname{rank}(P) \times n}$, $R \in \mathbb{R}^{\operatorname{rank}(Q) \times n}$ are full rank factors of P, Q.

Advantages:

- more reliable if Cholesky factors are numerically singular;
- more efficient if $\operatorname{rank}(P), \operatorname{rank}(Q) \ll n$;
- SVD is cheaper, e.g., semi-discretized point control of 1D heat equation with $n = 1000$: $\operatorname{rank}(P) \approx \operatorname{rank}(Q) \approx 20 \Rightarrow \mathcal{O}(10^5)$ flops instead of $\mathcal{O}(10^{10})$, i.e., factor $100,000$.



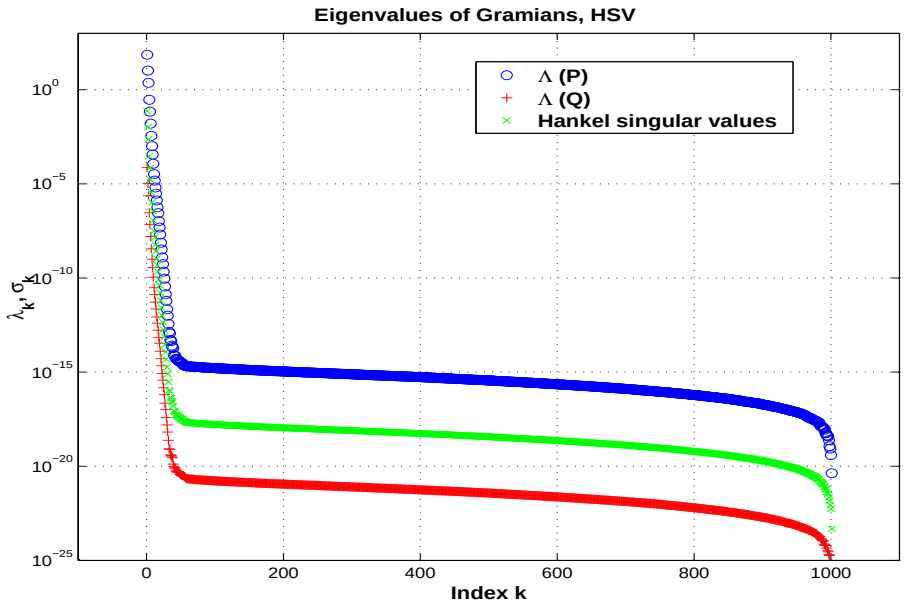
Example for Cheaper SVD

Complexity of SVD in standard approach: $\mathcal{O}(n^3)$.

Often, for large-scale systems Gramians have low numerical rank.

Example:

- Linear 1D heat equation with point control on $[\,0,\,1\,]$,
- FE-discretization with linear B-splines,
- $h = 1/1000$ ($\Longrightarrow n = 1001$).



$$\begin{array}{rcll} P & \approx & S^{(s)}(S^{(s)})^T, & S^{(s)} \in \mathbb{R}^{s\times n} \\ Q & \approx & R^{(s)}(R^{(s)})^T, & R^{(r)} \in \mathbb{R}^{r\times n} \end{array} \quad \Rightarrow \quad \text{SVD of } S^{(s)}(R^{(s)})^T \text{ has complexity } \mathcal{O}(r^2(s+r)).$$

Here: $s \approx r \approx 20 \implies \mathcal{O}(10^5)$ flops instead of $\mathcal{O}(10^{10})$.

Solving Lyapunov Equations with the Sign Function Method

Consider Lyapunov equation $F^T X + XF + E = 0$, F stable.

$$\Longrightarrow \begin{bmatrix} I_n \\ -X_* \end{bmatrix} \text{ is stable invariant subspace of } Z := \begin{bmatrix} F & 0 \\ E & -F^T \end{bmatrix}.$$

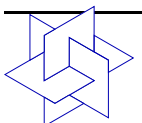
Apply sign function Newton iteration $Z_{j+1} \leftarrow (Z_j + Z_j^{-1})/2$ to Z.

$$\Longrightarrow \boxed{\operatorname{sign}(Z) = \lim_{j\to\infty} Z_j = \begin{bmatrix} -I_n & 0 \\ 2X_* & I_n \end{bmatrix}.}$$

Newton iteration (with scaling) is equivalent to

$$\begin{aligned} &F_0 \leftarrow F, \quad E_0 \leftarrow E, \\ &\texttt{for } j = 0, 1, 2, \ldots \\ &\qquad F_{j+1} \leftarrow \frac{1}{2c_j}\left(F_j + c_j^2 F_j^{-1}\right), \\ &\qquad E_{j+1} \leftarrow \frac{1}{2c_j}\left(E_j + c_j^2 F_j^{-T} E_j F_j^{-1}\right). \end{aligned}$$

$$\Longrightarrow \boxed{X_* = \tfrac{1}{2}\lim_{j\to\infty} E_j}$$



Semidefinite Lyapunov Equations

Here: $E = B^T B$ or $C^T C$, $F = A^T$ or A.

Want factor R of solution of $A^T Q + Q A + C^T C = 0$.

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

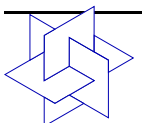
$$E_{j+1} = \frac{1}{2c_j}\left(E_j + c_j^2 A_j^{-T} E_j A_j^{-1}\right) = \frac{1}{2c_j}\begin{bmatrix} C_j \\ c_j C_j A_j^{-1} \end{bmatrix}^T \begin{bmatrix} C_j \\ c_j C_j A_j^{-1} \end{bmatrix}.$$

$\Longrightarrow$ re-write E_j–iteration:

$$\boxed{C_0 := C, \qquad C_{j+1} := \frac{1}{\sqrt{2c_j}}\begin{bmatrix} C_j \\ c_j C_j A_j^{-1} \end{bmatrix}.}$$

Problem: $C_j \in \mathbb{R}^{p_j \times n} \quad \Longrightarrow \quad C_{j+1} \in \mathbb{R}^{2p_j \times n}$,
i.e., the necessary workspace doubles in each iteration.

Two approaches in order to limit work space...



Compute Cholesky factor R_c of Q

Require $p_j \leq n$: for $j > \log_2(n/p)$ compute QR factorization

$$\frac{1}{\sqrt{2c_j}} \begin{bmatrix} C_j \\ c_j C_j A_j^{-1} \end{bmatrix} = U_j \begin{bmatrix} \hat{C}_j \\ 0 \end{bmatrix}, \qquad \hat{C}_j = \begin{bmatrix} \diagdown \end{bmatrix} \in \mathbb{R}^{n \times n}.$$

$$\Longrightarrow \quad E_j = \hat{C}_j^T \hat{C}_j, \qquad \boxed{R_c = \tfrac{1}{\sqrt{2}} \lim_{j \to \infty} \hat{C}_j}$$

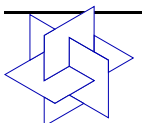
Compute full-rank factor R_f of Q

In every step compute rank-revealing QR factorization:

$$\frac{1}{\sqrt{2c_j}} \begin{bmatrix} C_j \\ c_j C_j A_j^{-1} \end{bmatrix} = U_{j+1} \begin{bmatrix} R_{j+1} & T_{j+1} \\ 0 & S_{j+1} \end{bmatrix} \Pi_{j+1},$$

where $R_{j+1} \in \mathbb{R}^{p_{j+1} \times p_{j+1}}$, $p_{j+1} = \operatorname{rank}\left(\begin{bmatrix} C_j \\ c_j C_j A_j^{-1} \end{bmatrix} \right)$. Then

$$C_{j+1} := [\, R_{j+1} \; T_{j+1} \,] \Pi_{j+1}, \quad E_{j+1} = C_{j+1}^T C_{j+1}, \qquad \boxed{R_f = \tfrac{1}{\sqrt{2}} \lim_{j \to \infty} C_j}$$



Model Reduction Based on Relative Errors

Compute reduced-order system such that relative error $\|\Delta_{\rm rel}\|_\infty$ becomes "small",

$$\tilde{G}(s) = G(s)(I + \Delta_{\rm rel}).$$

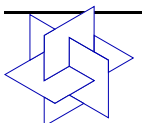
For $p = m$, D full-rank: find $\displaystyle\operatorname*{argmin}_{\text{degree}(\tilde{G}) \le \ell} \|G^{-1}(G - \tilde{G})\|_\infty$.

Balanced stochastic truncation (BST): [*Desai/Pal '84, Green '88*]

- Compute balancing transformation for controllability Gramian of $G(s)$ and observability Gramian W of right spectral factor $\hat{C}(sI - \hat{A})^{-1}\hat{B} + \hat{D}$ of power spectrum $\Phi(s) := G(s)G^T(-s)$.
- W is stabilizing solution of algebraic Riccati equation (ARE): ($E := DD^T$)

$$0 \quad = \quad C^T E^{-1} C + (A - \hat{B}E^{-1}C)^T W + W(A - \hat{B}E^{-1}C) + W\hat{B}E^{-1}\hat{B}^T W.$$

  Numerical solution via Newton's method with line search [*B. '97, B./Byers '98*]
  Newton iteration: solve Lyapunov equation using factored sign function iteration $\rightsquigarrow$ factored solution of ARE. [*B./Byers/Quintana-Ortí×2 '00*]



Advantages of Stochastic Truncation

- Global relative error bound $\boxed{\|\Delta_{\mathrm{rel}}\|_\infty \leq \prod_{j=\ell+1}^{n} \frac{1+\mu_j}{1-\mu_j} - 1, \ \lambda(PW) = \{\mu_j^2\} \subset [0,1]}$

  $\Rightarrow$ uniform approximation of transfer function over whole frequency domain.
- Additional system properties are preserved compared to balanced truncation:
  - right-half plane zeroes $\rightsquigarrow$ reduced-order model of minimum-phase system is minimum phase;
  - robust stability (controller for reduced model stabilizes full-order plant) [*Safonov/Chiang '88*].
- $\exists$ error bounds for phase (for BT, only error bounds for magnitude).
- Straightforward error bound for inverse system:

$$\|G^{-1} - \tilde{G}^{-1}\|_\infty \leq \|\Delta_{\mathrm{rel}}\|_\infty \|\tilde{G}^{-1}\|_\infty$$

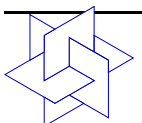
  $\rightsquigarrow$ can solve inverse problems

$$\boxed{y = Gu, \quad y \text{ known from measurements}, \quad \text{compute } u = G^{-1}y}$$

  for a given accuracy threshold with reduced-order model.

Analogous computational techniques for

- LQG balancing (reduce plant and controller at the same time),
- positive real balancing (preserve passivity).



Model Reduction of Unstable Systems

Use additive decomposition of transfer function,

$$G(s) \;=\; G_-(s) + G_+(s), \qquad G_-(s) \text{ stable}, \quad G_+(s) \text{ anti-stable},$$

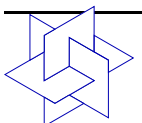
and reduce $G_-(s)$ using BT etc., keep $G_+(s)$ (dominates the dynamics).

Need block-diagonalization of A:

$$\hat{A} := U^{-1}AU = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} \Rightarrow \hat{B} := U^{-1}B =: \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \; \hat{C} := CU =: [\, C_1 \; C_2 \,],$$

Then

$$\begin{aligned} G(s) &= \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} (sI_k - A_{11})^{-1} & \\ & (sI_{n-k} - A_{22})^{-1} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} + D \\ &= \left\{C_1(sI_k - A_{11})^{-1}B_1\right\} + \left\{C_2(sI_{n-k} - A_{22})^{-1}B_2 + D\right\} =: G_-(s) + G_+(s), \end{aligned}$$



Block-Diagonalization via the Sign Function Method

1. Compute $\operatorname{sign}(A)$ and obtain spectral projector $\mathcal{P}_- := \frac{1}{2}(I_n - \operatorname{sign}(Z)) \Rightarrow$ obtain block-triangular form from QR decomposition of $\mathcal{P}_-$

$$\mathcal{P}_- = QRP, \qquad \tilde{A} := Q^T A Q = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}.$$

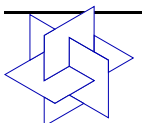
2. Solve Sylvester equation $A_{11}Y - YA_{22} + A_{12} = 0$. Then

$$\hat{A} := V^{-1}\tilde{A}V = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}, \qquad V := \begin{bmatrix} I_k & Y \\ 0 & I_{n-k} \end{bmatrix}.$$

$A_{11}, -A_{22}$ stable $\Rightarrow$ solve Sylvester equation via sign function method:

$$\begin{aligned} E_0 &:= A_{11}, & E_{j+1} &:= \tfrac{1}{2}\left(E_j + E_j^{-1}\right), \\ F_0 &:= A_{22}, & F_{j+1} &:= \tfrac{1}{2}\left(F_j + F_j^{-1}\right), & j = 0, 1, 2, \ldots. \\ W_0 &:= A_{12}, & W_{j+1} &:= \tfrac{1}{2}\left(W_j + E_j^{-1} W_j F_j^{-1}\right), \end{aligned}$$

$\Rightarrow \lim_{j\to\infty} E_j = -I_k$, $\lim_{j\to\infty} F_j = I_{n-k}$, and $Y = \frac{1}{2}\lim_{j\to\infty} W_j$.



Model Reduction for Large-Scale Systems

Large-scale dense problems of size $n = \mathcal{O}(10^k)$, $m, p \ll n$ arise, e.g., from discretization of integral equations via BEM or wavelet techniques.

$n = \mathcal{O}(10^3)$ $\rightsquigarrow$ sign function based methods applicable on current workstations.

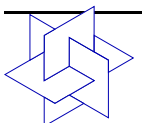
$n = \mathcal{O}(10^4)$ $\rightsquigarrow$ parallelization on PC or workstation cluster using off-the-shelf computer technology (standard chips, Fast Ethernet,...) and software (MPI, PBLAS, BLACS, ScaLAPACK).

Large-scale sparse systems, e.g., from 3D FEM models, large-scale circuits, etc. $\rightsquigarrow$ use the same ideas (truncation methods, factored Gramians), but need sparse Lyapunov/Riccati solvers.

- Balanced truncation: [*Penzl '98, Li/White '99–'02, Antoulas/Sorensen/Gugercin/Zhou '00–'03*]
- Stochastic truncation: [*B. '0x*]

Alternative:

sparse representation (approximation) of A using hierarchical matrices
[*Hackbusch/Khoromskij/Grasedyck '03, Ph.D. thesis Baur in progress*]



$\mathcal{H}$-Matrix Implementation

Recall: solution of the Lyapunov equation

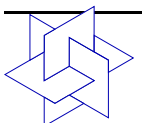
$$FP + PF^T + BB^T = 0$$

with the sign function method:

$$\begin{aligned} F_0 &= A, \quad E_0 = BB^T \\ F_{j+1} &= \frac{1}{2}(F_j + F_j^{-1}) \\ E_{j+1} &= \frac{1}{2}(E_j + A_j^{-T} E_j A_j^{-1}) \end{aligned}$$

involves the inversion, addition and multiplication of $n \times n$ matrices
$\hookrightarrow$ complexity: $\mathcal{O}(n^3)$

Approximation of A in $\mathcal{H}$-matrix format, use of the formatted $\mathcal{H}$-matrix arithmetic
$\hookrightarrow$ complexity: $\mathcal{O}(n \log^q n)$.



Hierarchical Matrices: Short Introduction

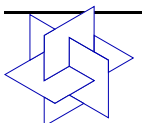
Hierarchical ($\mathcal{H}$-)matrices are a data-sparse approximation of large, dense matrices arising

- from the discretisation of non-local integral operators occuring in BEM,
- as inverses of FEM discretized elliptic differential operators,

but can also be used to represent FEM matrices directly.

Properties of $\mathcal{H}$-matrices:

- only few data are needed for the representation of the matrix,
- matrix-vector multiplication can be performed in almost linear complexity ($\mathcal{O}(n \log^q n)$),
- building sums, products, inverses is of "almost" linear complexity.



Hierarchical Matrices: Construction

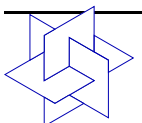
Consider matrices over a product index set $I \times I$.

Partition $I \times I$ by the $\mathcal{H}$-tree $T_{I\times I}$, where a problem dependend admissibility condition is used to decide whether a block $t \times s \subset I \times I$ allows for a low rank approximation.

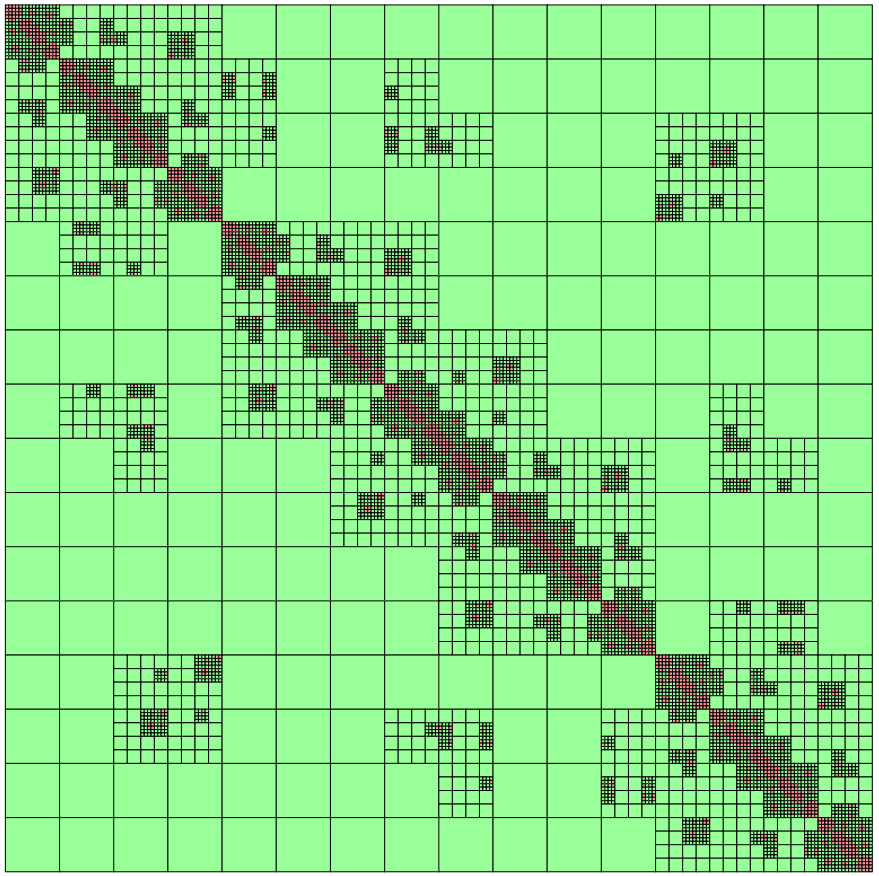
Definition: Hierarchical matrices ($\mathcal{H}$-matrices)
The set of the hierarchical matrices is defined by

$$\mathcal{H}(T_{I\times I}, k) := \{M \in \mathbb{R}^{I\times I} |\ \mathrm{rank}(M|_{t\times s}) \leq k \,\forall \text{ admissible leaves } t\times s \text{ of } T_{I\times I}\}$$

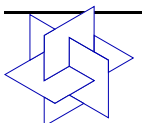
Submatrices of $M \in \mathcal{H}(T_{I\times I}, k)$ corresponding to inadmissible leaves are stored as dense blocks whereas those corresponding to admissible leaves are stored in factorized form as rank-k matrices, called R_k-format.



Example



Stiffness matrix of 2D heat equation with distributed control and isolation BC
$n = 1024$ and $k = 4$



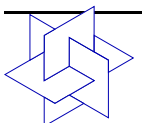
Hierarchical Matrices: Formatted Arithmetic

$\mathcal{H}(T_{I\times I}, k)$ is not a linear subspace of $\mathbb{R}^{I\times I}$ $\rightsquigarrow$ formatted arithmetics
$\rightsquigarrow$ projection of the sum, product and inverse into $\mathcal{H}(T_{I\times I}, k)$

1. Formatted Addition ($\oplus$)
   with complexity $\mathcal{N}_{\mathcal{H}\oplus\mathcal{H}} = \mathcal{O}(nk^2 \log n))$ (for sparse $T_{I\times I}$)
   Corresponds to best approximation (in the Frobenius-norm).

2. Formatted Multiplication ($\odot$)
   $\mathcal{N}_{\mathcal{H}\odot\mathcal{H}} = \mathcal{O}(nk^2 \log^2 n)$ (under some technical assumptions on $T_{I\times I}$)

3. Formatted inversion ($\widetilde{Inv}$)
   $\mathcal{N}_{\mathcal{H},\widetilde{Inv}} = \mathcal{O}(nk^2 \log^2 n)$ (under some technical assumptions on $T_{I\times I}$)



Error propagation

Sign function iteration with formatted $\mathcal{H}$-matrix arithmetic:

$$\tilde{A}_0 = (A)_{\mathcal{H}}, \qquad \tilde{A}_{j+1} = \frac{1}{2}(\tilde{A}_j \oplus \widetilde{Inv}(A_j))$$

Accuracy control for iterates $\rightsquigarrow k = \mathcal{O}(\log \frac{1}{\delta} + \log \frac{1}{\rho}))$, where

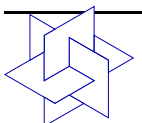
$$\begin{aligned} \|(\tilde{A}_j^{-1} - \widetilde{\mathrm{Inv}}(\tilde{A}_j))\|_2 &\leq \delta \\ \|(\tilde{A}_j^{-1} + \widetilde{\mathrm{Inv}}(\tilde{A}_j)) - (\tilde{A}_j^{-1} \oplus \widetilde{\mathrm{Inv}}(\tilde{A}_j))\|_2 &\leq \rho \end{aligned}$$

$\Longrightarrow$ forward error bound (assuming $c_j(\delta+\rho)\|A_j^{-1}\|_2 < 1 \ \forall j$):

$$\|\tilde{A}_j - A_j\|_2 \leq c_j(\delta + \rho),$$

where

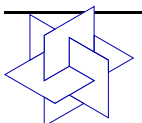
$$c_0 = \|\tilde{A}_0 - A\|_2(\delta+\rho)^{-1}, \quad c_{j+1} = \frac{1}{2}\left(1 + c_j + c_j \frac{\|A_j^{-1}\|_2^2}{1 - c_j(\delta+\rho)\|A_j^{-1}\|_2}\right).$$



Parallelization

- Newton iteration for sign function easy to parallelize – need basic linear algebra (systems of linear equations, matrix inverse, matrix addition, matrix product).
- Use MPI, BLACS for communication, PBLAS and ScaLAPACK for numerical linear algebra $\longrightarrow$ portable code.
- Development of software library PLiCMR.
- Testing on PC Cluster (Linux) with 32 Intel Pentium II-300MHz processors.
  - workspace/processor: 128 MBytes.
  - Myrinet Switch, bandwidth $\approx$ 100 Mbit/sec.
- Results on 1 processor: SLICOT codes, based on computation of Cholesky factors via Hammarling's method.

SLICOT = Subroutine Library in Control Theory, available from http://www.win.tue.nl/niconet



Numerical Examples

Benchmark tests:

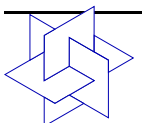
1. 1D heat equation with point control

Control of temperature distribution in thin rod with heat source in the middle $\Longrightarrow$ 1D heat equation with homogeneous Dirichlet boundary. Discretization with FEM, linear elements.

$n =$ dimension of the FE ansatz space.
$m = 1$: spatially constant heat source.
$p = 1$: temperature is measured in one interval.

2. Simulation of catalytic reactor (taken from ABB *gPROMS* tutorial)

- FE discretization of boundary control problem for coupled PDE system (conservation laws, reaction-diffusion equations, Robin and Neumann boundary conditions), linearization around working point.
- Dynamics: oxidation (o-Xylene to phthalic anhydrite).
- Control: external cooling of the reactor.
- $n = 1171$, $m = 6$, $p = 4$.



3. "Random" systems with given McMillan degree and given rank of Gramians.

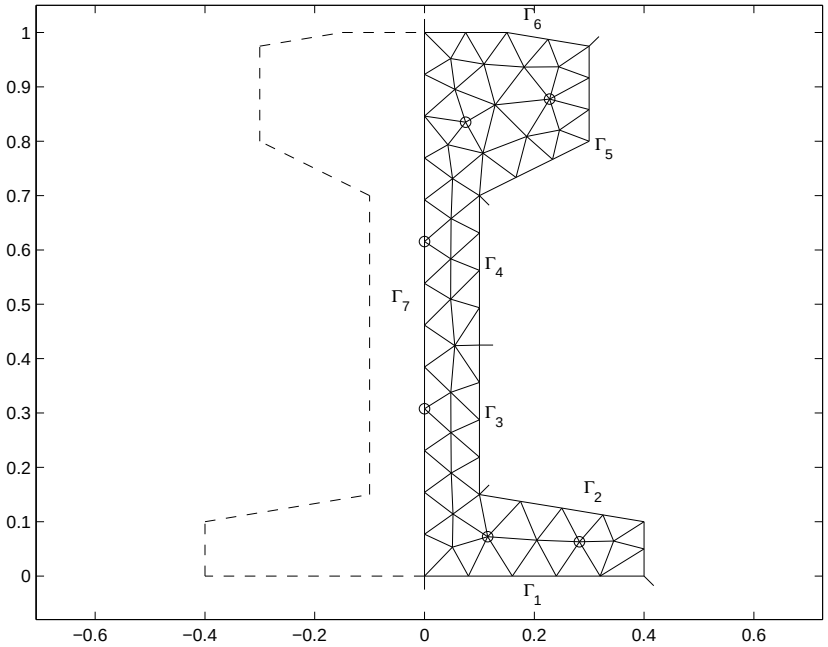
4. Optimal cooling of steel profiles (model by Mannesmann/Demag, [*Tröltzsch/Unger, Penzl 1999*])

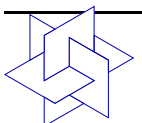
   - Mathematical model: boundary control for linearized 2D heat equation.

$$\begin{aligned} \frac{\partial}{\partial t}x &= \frac{\lambda}{c\cdot\rho}\Delta x, && \xi\in\Omega \\ \frac{\partial}{\partial n}x &= \frac{1}{\lambda}(u_k - x), && \xi\in\Gamma_k,\ k=1,\ldots,6, \\ \frac{\partial}{\partial n}x &= 0, && \xi\in\Gamma_7. \end{aligned}$$

   $\Longrightarrow m = p = 6$

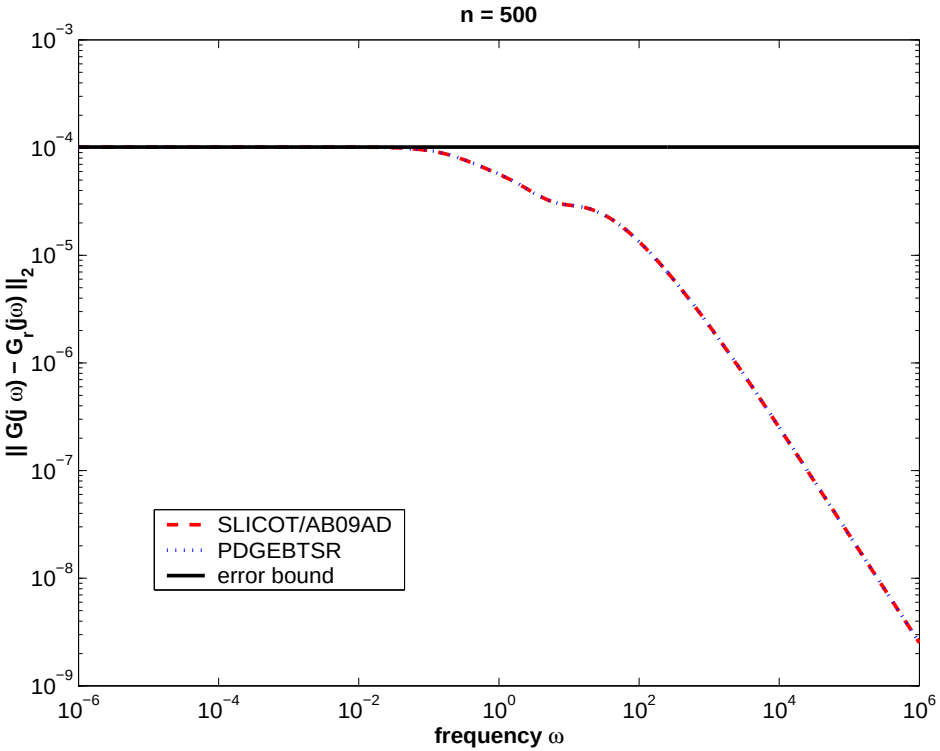
   - FEM Discretization, initial mesh ($n = 821$).
     2 steps of mesh refinement $\Longrightarrow n = 3113$.

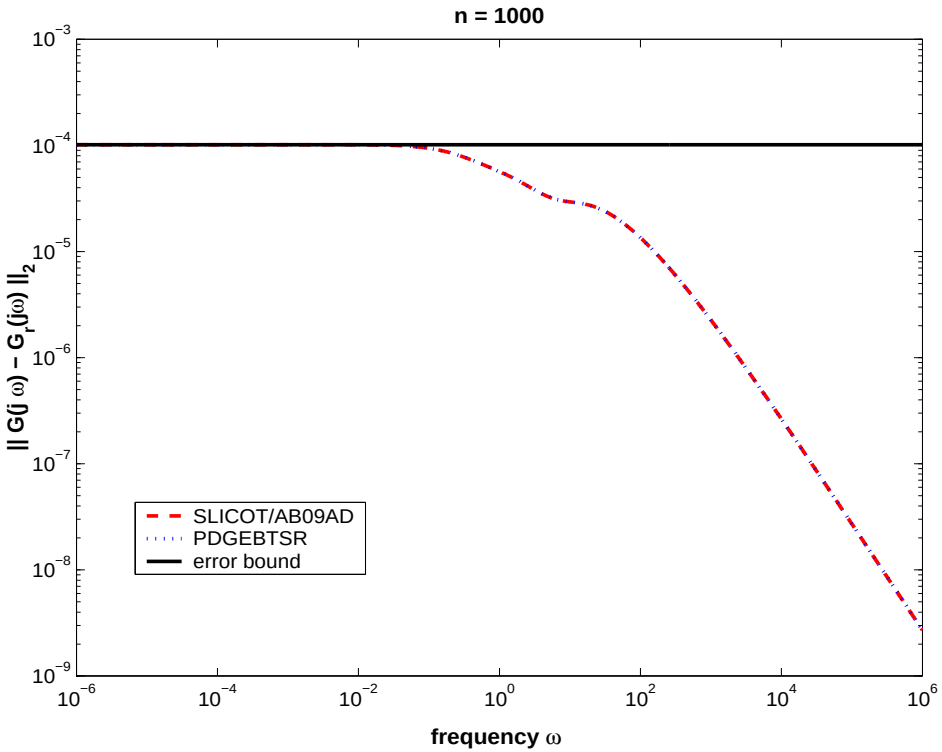


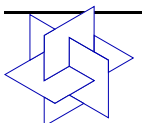


Absolute Error for Balanced Truncation

Example 1: $\mathrm{rank}\,(P) = 32$, $\mathrm{rank}\,(Q) = 38(37)$, $\ell = 6$.



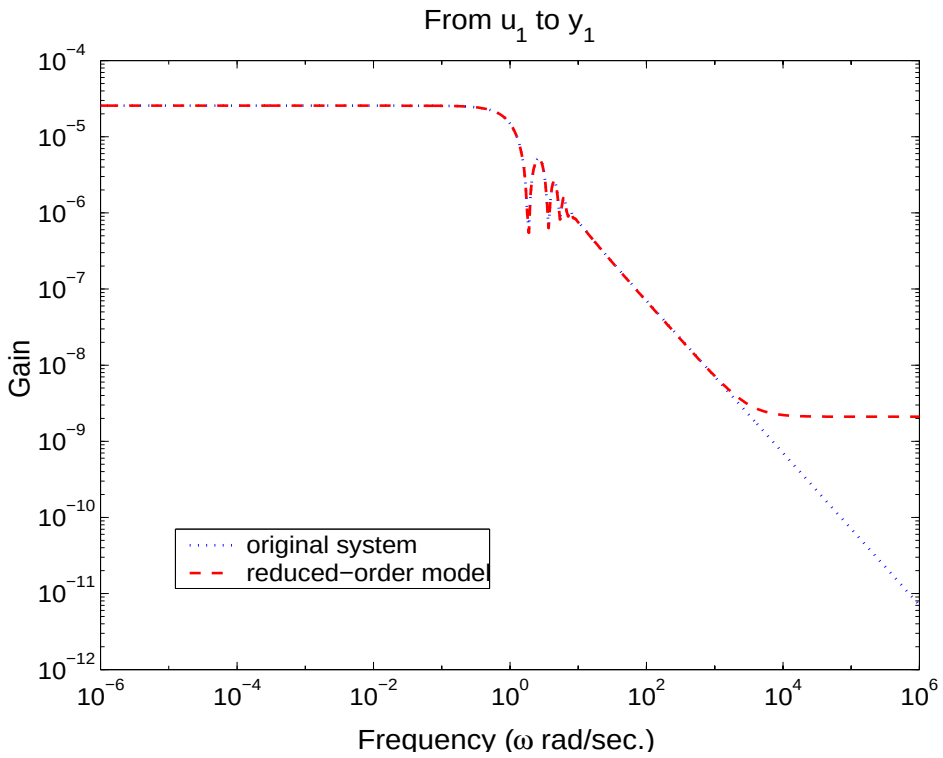


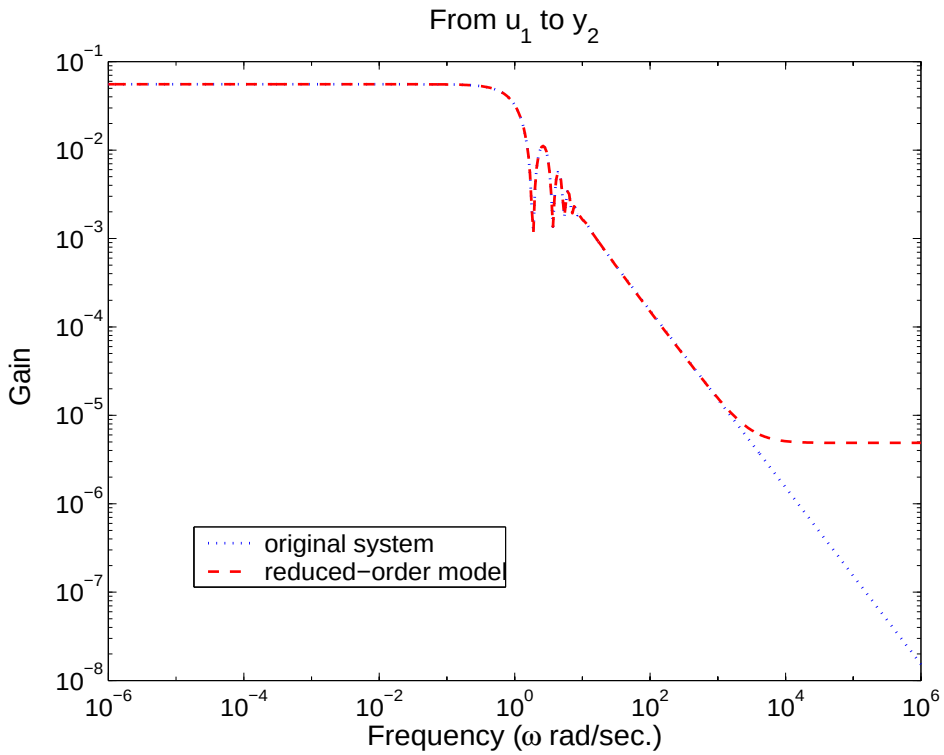


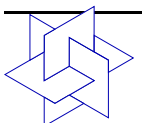
Absolute Error for Singular Perturbation Approximation

SPA: minimal realization via BT, compute $\tilde{G}$ such that $G(0) = \tilde{G}(0)$ and $\frac{\|G-\tilde{G}\|_\infty}{2} \leq \sum_{k=\ell+1}^{n} \sigma_k$.

Example 2: $\operatorname{rank}(P) = 124$, $\operatorname{rank}(Q) = 93$, $\ell = 40$

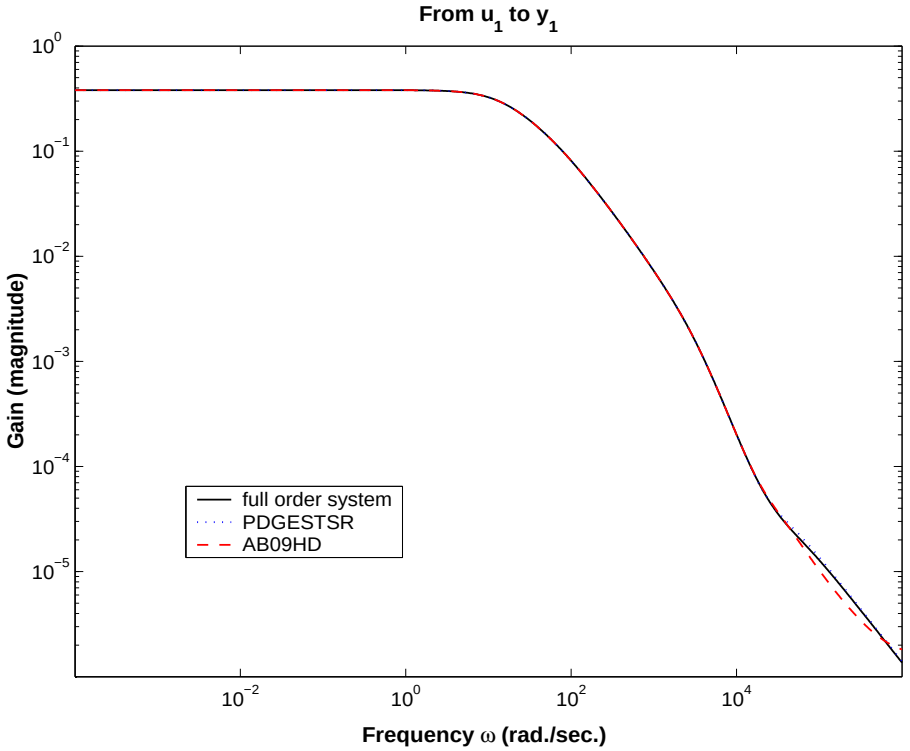


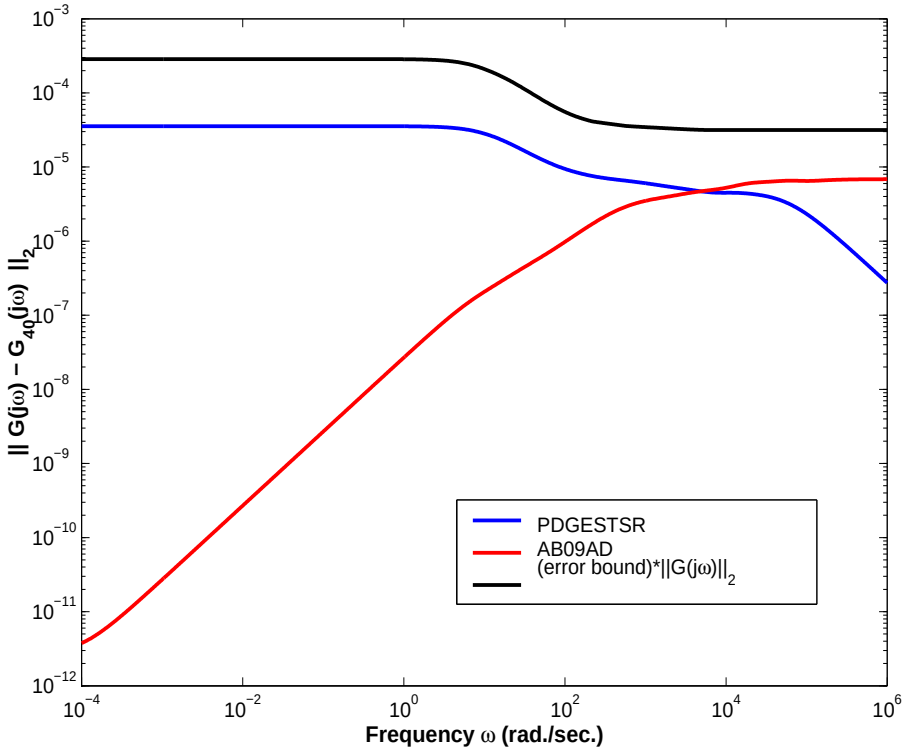




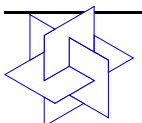
Uniform Approximation of BST/Example 4

- $n = 821$: $\mathrm{rank}\,(P) = 165$, $\mathrm{rank}\,(Q) = 210$, $\ell = 40 \;\Rightarrow\; \|G - \tilde{G}\|_\infty \leq 3.2 \cdot 10^{-4} \|G\|_\infty$



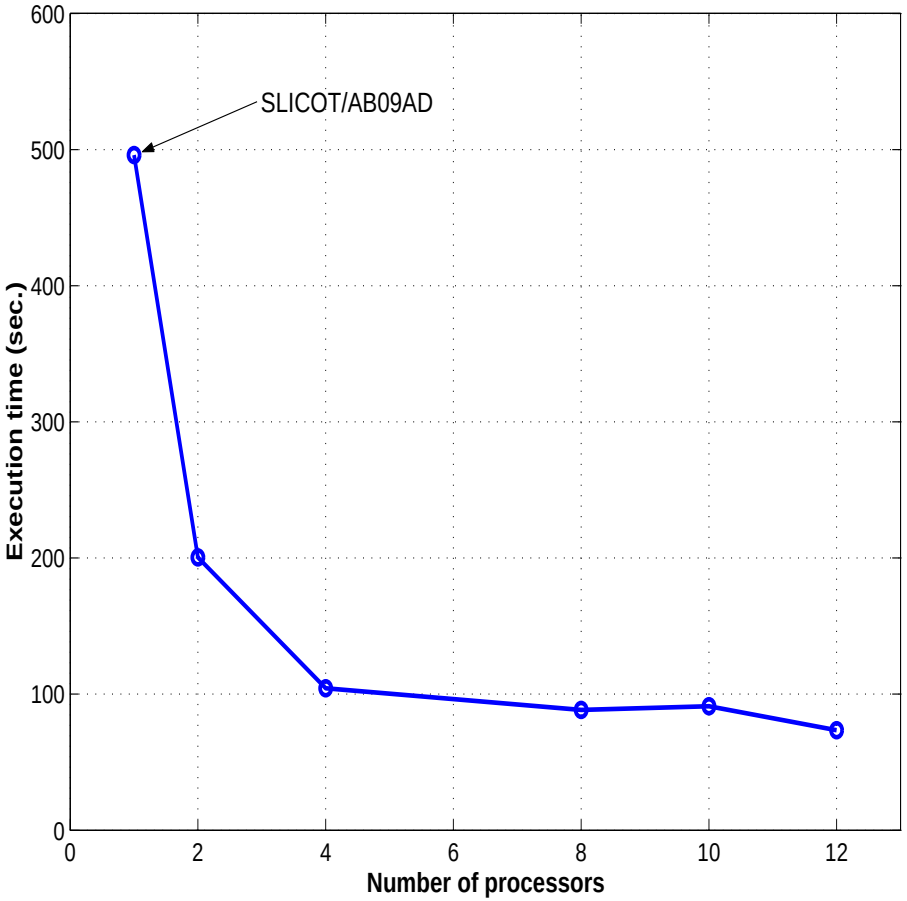


- $n = 3113$: $\mathrm{rank}\,(P) = 179$, $\mathrm{rank}\,(Q) = 204$,
  - (numerical) stochastic McMillan degree: $\mu_{\hat{n}} \leq n \cdot \mu_1 \cdot \mathbf{u} \;\Rightarrow\; \hat{n} = 135$.
  - $\ell = 40 \;\Rightarrow\; \|G - \tilde{G}\|_\infty \leq 2.8 \cdot 10^{-4} \|G\|_\infty$
  - SVD: $<$ 1sec. on 2 processors using full-rank factors, $\approx$ 25 minutes using Cholesky factors.

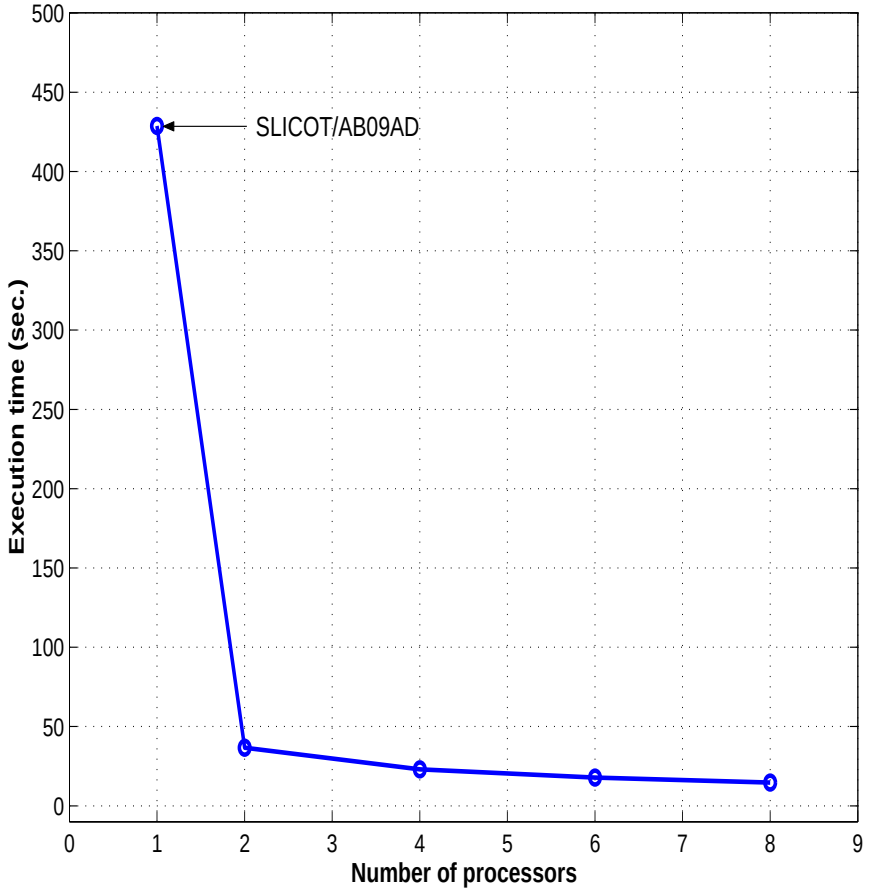


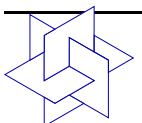
Speed-Up/Efficiency of Parallel Algorithms

Example 3/balanced truncation:
$n = 1000, m = p = 100, \hat{n} = \ell = 50$



Example 3/BT, discrete-time:
$n = 1000, m = p = 100, \hat{n} = \ell = 50$

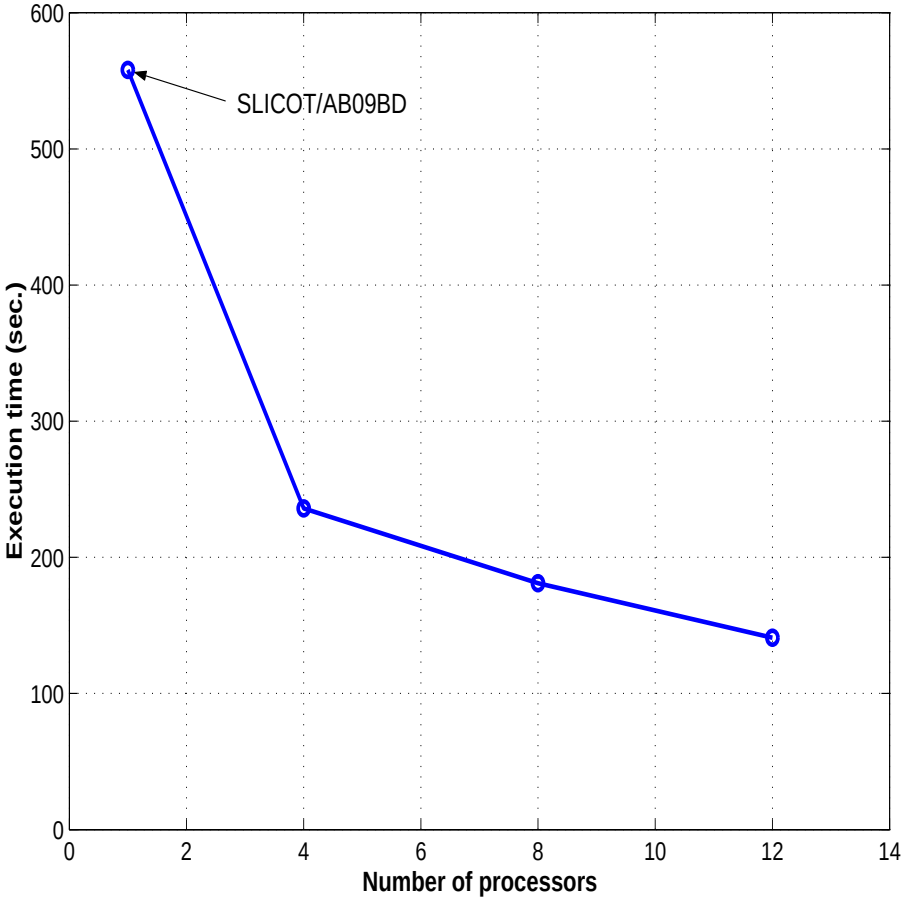




Speed-Up/Efficiency of Parallel Algorithms

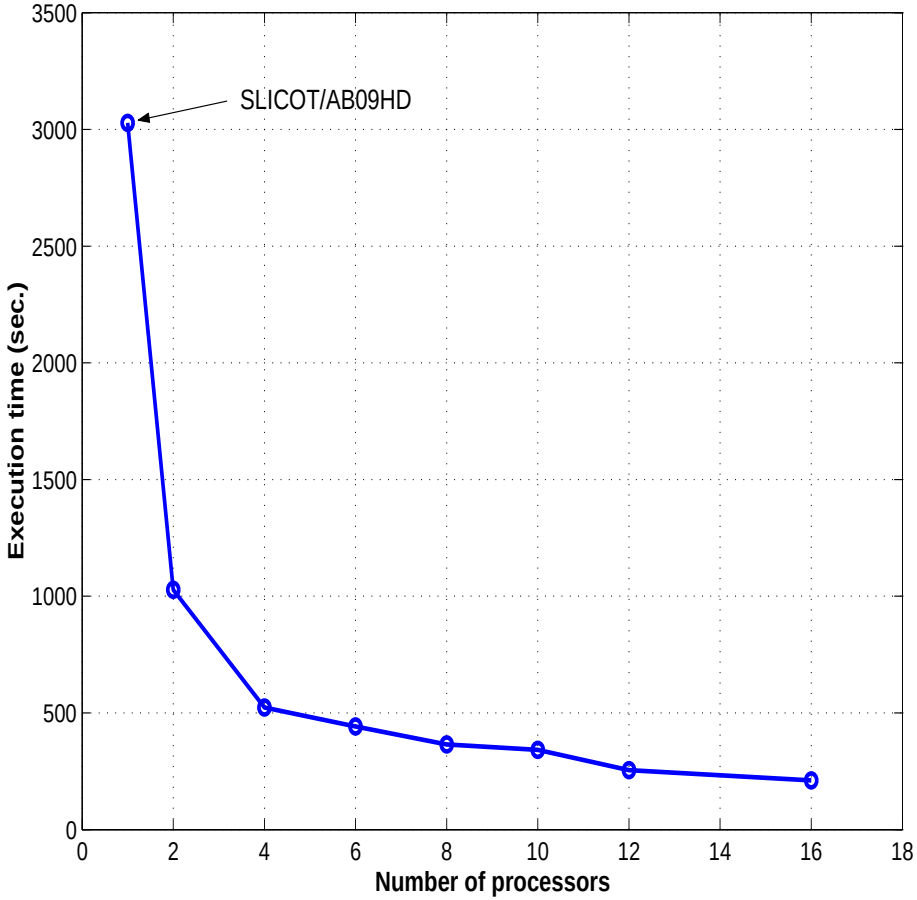
Example 2/SPA:

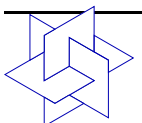
$n = 1171, m = 6, p = 4, \ell = 40$



Example 3/BST:

$n = 1000, m = 10, p = 10, \ell = 40$

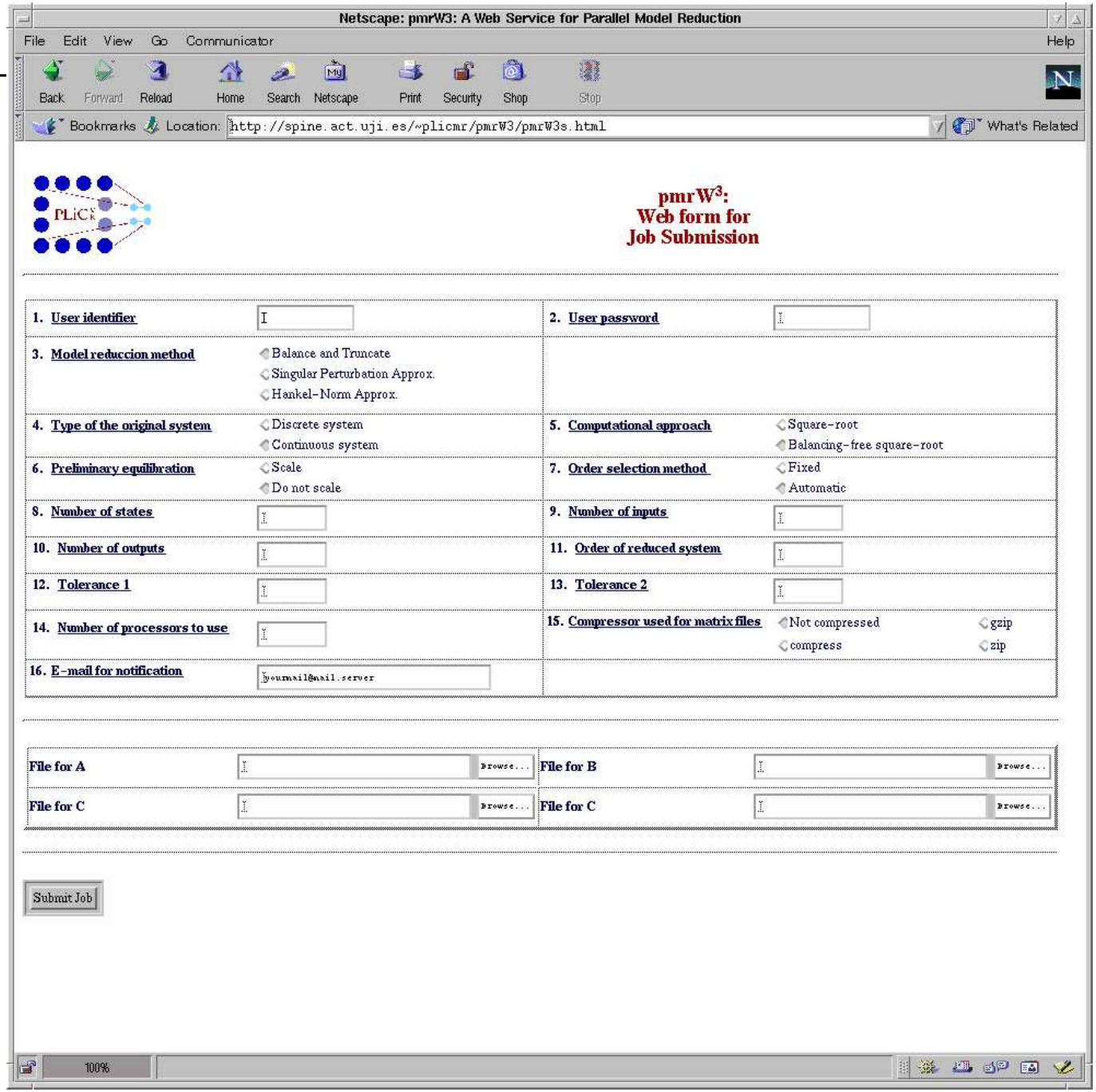




Remote Computing: PLiCMR Web Interface

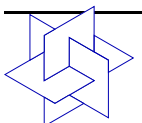
- Upload data on cluster.
- Select MR method.
- Submit job.
- User receives reduced-order model
  - A, B, C, D
  - HSV
  - infos

  via e-mail.



Conclusions

- Not shown, but "available" spectral projection-based methods:
  - BT et al for discrete-time systems: compute factors of Gramians using factored Smith-Iteration,
  - optimal Hankel norm approximation,
  - modal truncation,
  - "generic" frequency-weighted BT,
  - BT for descriptor systems using spectral projection-based approach equivalent to [*Stykel '01–'03*],
  - truncation for SISO and symmetric systems using the cross-Gramian $\rightsquigarrow$ factored solution of Sylvester equations, need product QR for non-square factors!
- Parallel implementations are collected in software library PLiCMR and are integrated into parallel version of SLICOT, accessible via remote computing.
- Implementations of methods for sparse systems based on ADI in progress.



Open Problems

- Error bounds for interconnecting compact models using frequency-weighted BT-based models?
- In circuit simulation, need passive reduced systems for DAE systems $\rightsquigarrow$ positive real balancing applicable?
- Exploiting PDE structures $\rightsquigarrow$ $\mathcal{H}$-matrix based BT, combination with POD for nonlinear systems?
- Sign function implementation using formatted arithmetic based on wavelet compression techniques $\rightsquigarrow$ promising results by [*Schneider/Harbrecht '03*].
- How far can we go with dense parallel implementations?

  With 32-node (2 P4 with 1 GB each) cluster can apply (dense) BT for $n = 75,000$ in single precision.