Otto-von-Guericke Universität Magdeburg
Faculty of Mathematics
Summer term 2012

Model Reduction for Dynamical Systems

— Lecture 10 —

Peter Benner

Max Planck Institute for Dynamics of Complex Technical Systems
Computational Methods in Systems and Control Theory
Magdeburg, Germany

benner@mpi-magdeburg.mpg.de

www.mpi-magdeburg.mpg.de/research/groups/csc/lehre/2012_SS_MOR/

Outline

Numerical Methods for Solving Lyapunov Equations
The Sign Function Method

Definition 4.8

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

$$Z = S \begin{bmatrix} J^+ & 0 \\ 0 & J^- \end{bmatrix} S^{-1}$$

the matrix sign function is

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

Numerical Methods for Solving Lyapunov Equations
The Sign Function Method

Definition 4.8

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

$$Z = S \begin{bmatrix} J^+ & 0 \\ 0 & J^- \end{bmatrix} S^{-1}$$

the matrix sign function is

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

Lemma 4.9

Let $T \in \mathbb{R}^{n\times n}$ be nonsingular and Z as in Definition 4.8, then

$$\operatorname{sign}\left(TZT^{-1}\right) = T \operatorname{sign}(Z)\, T^{-1}$$

Numerical Methods for Solving Lyapunov Equations
The Sign Function Method

Computation of $\mathrm{sign}\,(Z)$

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

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

$\Longrightarrow \quad \mathrm{sign}\,(Z) = \lim_{j\to\infty} Z_j.$

$c_j > 0$ is scaling parameter for convergence acceleration and rounding error minimization. In `lyap_sgn_fac`:

$$c_j = \sqrt{\frac{\|Z_j^{-1}\|_F}{\|Z_j\|_F}},$$

based on "equilibrating" the norms of the two summands [HIGHAM '86].

Solving Lyapunov Equations with the Matrix Sign Function Method

Key observation:

If $X \in \mathbb{R}^{n\times n}$ is a solution of $AX + XA^T = W$, then

$$\underbrace{\begin{bmatrix} I_n & -X \\ 0 & I_n \end{bmatrix}}_{=T^{-1}} \underbrace{\begin{bmatrix} A & W \\ 0 & -A^T \end{bmatrix}}_{=:H} \underbrace{\begin{bmatrix} I_n & X \\ 0 & I_n \end{bmatrix}}_{=:T} = \begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix}.$$

Hence, if A is Hurwitz (i.e., asymptotically stable), then

$$\begin{aligned} \operatorname{sign}(H) &= \operatorname{sign}\left(T \begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix} T^{-1}\right) = T \operatorname{sign}\left(\begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix}\right) T^{-1} \\ &= \begin{bmatrix} -I_n & 2X \\ 0 & I_n \end{bmatrix}. \end{aligned}$$

Solving Lyapunov Equations with the Matrix Sign Function Method

Apply sign function iteration $Z \leftarrow \frac{1}{2}(Z + Z^{-1})$ to $H = \begin{bmatrix} A & W \\ 0 & -A^T \end{bmatrix}$:

$$H + H^{-1} = \begin{bmatrix} A & W \\ 0 & -A^T \end{bmatrix} + \begin{bmatrix} A^{-1} & A^{-1}WA^{-T} \\ 0 & -A^{-T} \end{bmatrix}$$

$\Longrightarrow$ Sign function iteration for Lyapunov equation:

$$\begin{aligned} A_0 &\leftarrow A, \quad A_{j+1} \leftarrow \frac{1}{2}\left(A_j + A_j^{-1}\right), \\ W_0 &\leftarrow G, \quad W_{j+1} \leftarrow \frac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right), \end{aligned} \qquad j = 0, 1, 2, \ldots.$$

Define $A_\infty := \lim_{j\to\infty} A_j$, $W_\infty := \lim_{j\to\infty} W_j$.

Theorem 4.10

If A is Hurwitz, then

$$A_\infty = -I_n \qquad \text{and} \qquad X = \frac{1}{2} W_\infty.$$

Solving Lyapunov Equations with the Matrix Sign Function Method
Factored form

Recall sign function iteration for $AX + XA^T + W = 0$:

$$\begin{array}{ll} A_0 \leftarrow A, & A_{j+1} \leftarrow \frac{1}{2}\left(A_j + A_j^{-1}\right), \\ W_0 \leftarrow G, & W_{j+1} \leftarrow \frac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right), \end{array} \qquad j = 0, 1, 2, \ldots.$$

Now consider the second iteration for $W = BB^T$, starting with $W_0 = BB^T =: B_0 + B_0^T$:

$$\begin{aligned} \frac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right) &= \frac{1}{2}\left(B_j B_j^T + A_j^{-1} B_j B_j^T A_j^{-T}\right) \\ &= \frac{1}{2}\left[B_j \quad A_j^{-1} B_j\right]\left[B_j \quad A_j^{-1} B_j\right]^T. \end{aligned}$$

Hence, obtain factored iteration

$$B_{j+1} \leftarrow \frac{1}{\sqrt{2}}\left[B_j \quad A_j^{-1} B_j\right]$$

with $S := \frac{1}{\sqrt{2}} \lim_{j \to \infty} B_j$ and $X = SS^T$.

Solving Lyapunov Equations with the Matrix Sign Function Method
Factored form

Recall sign function iteration for $AX + XA^T + W = 0$:

$$\begin{array}{ll} A_0 \leftarrow A, & A_{j+1} \leftarrow \frac{1}{2}\left(A_j + A_j^{-1}\right), \\ W_0 \leftarrow G, & W_{j+1} \leftarrow \frac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right), \end{array} \qquad j = 0, 1, 2, \ldots.$$

Now consider the second iteration for $W = BB^T$, starting with $W_0 = BB^T =: B_0 + B_0^T$:

$$\begin{aligned} \frac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right) &= \frac{1}{2}\left(B_j B_j^T + A_j^{-1} B_j B_j^T A_j^{-T}\right) \\ &= \frac{1}{2}\left[B_j \quad A_j^{-1} B_j\right]\left[B_j \quad A_j^{-1} B_j\right]^T. \end{aligned}$$

Hence, obtain factored iteration

$$B_{j+1} \leftarrow \frac{1}{\sqrt{2}}\left[B_j \quad A_j^{-1} B_j\right]$$

with $S := \frac{1}{\sqrt{2}} \lim_{j\to\infty} B_j$ and $X = SS^T$.

Solving Lyapunov Equations with the Matrix Sign Function Method
Factored form

Recall sign function iteration for $AX + XA^T + W = 0$:

$$\begin{aligned} A_0 &\leftarrow A, \quad A_{j+1} \leftarrow \tfrac{1}{2}\left(A_j + A_j^{-1}\right), \\ W_0 &\leftarrow G, \quad W_{j+1} \leftarrow \tfrac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right), \end{aligned} \qquad j = 0, 1, 2, \ldots.$$

Now consider the second iteration for $W = BB^T$, starting with $W_0 = BB^T =: B_0 + B_0^T$:

$$\begin{aligned} \frac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right) &= \frac{1}{2}\left(B_j B_j^T + A_j^{-1} B_j B_j^T A_j^{-T}\right) \\ &= \frac{1}{2}\left[B_j \quad A_j^{-1} B_j\right]\left[B_j \quad A_j^{-1} B_j\right]^T. \end{aligned}$$

Hence, obtain factored iteration

$$B_{j+1} \leftarrow \frac{1}{\sqrt{2}}\left[B_j \quad A_j^{-1} B_j\right]$$

with $S := \frac{1}{\sqrt{2}} \lim_{j \to \infty} B_j$ and $X = SS^T$.

Solving Lyapunov Equations with the Matrix Sign Function Method
Factored form

Recall sign function iteration for $AX + XA^T + W = 0$:

$$\begin{array}{ll} A_0 \leftarrow A, \quad A_{j+1} \leftarrow \frac{1}{2}\left(A_j + A_j^{-1}\right), \\ W_0 \leftarrow G, \quad W_{j+1} \leftarrow \frac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right), \end{array} \qquad j = 0, 1, 2, \ldots.$$

Now consider the second iteration for $W = BB^T$, starting with $W_0 = BB^T =: B_0 + B_0^T$:

$$\begin{aligned} \frac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right) &= \frac{1}{2}\left(B_j B_j^T + A_j^{-1} B_j B_j^T A_j^{-T}\right) \\ &= \frac{1}{2}\left[B_j \quad A_j^{-1} B_j\right]\left[B_j \quad A_j^{-1} B_j\right]^T. \end{aligned}$$

Hence, obtain factored iteration

$$B_{j+1} \leftarrow \frac{1}{\sqrt{2}}\left[B_j \quad A_j^{-1} B_j\right]$$

with $S := \frac{1}{\sqrt{2}} \lim_{j\to\infty} B_j$ and $X = SS^T$.

Solving Lyapunov Equations with the Matrix Sign Function Method
Factored form

Factored sign function iteration for $A(SS^T) + (SS^T)A^T + BB^T = 0$

$$\begin{array}{ll} A_0 \leftarrow A, \quad A_{j+1} \leftarrow \frac{1}{2}\left(A_j + A_j^{-1}\right), & \\ B_0 \leftarrow B, \quad B_{j+1} \leftarrow \frac{1}{\sqrt{2}}\left[B_j \quad A_j^{-1}B_j\right], & \end{array} \qquad j = 0, 1, 2, \ldots.$$

Remarks:

- To get both Gramians, run in parallel

$$C_{j+1} \leftarrow \frac{1}{\sqrt{2}} \begin{bmatrix} C_j \\ C_j A_j^{-1} \end{bmatrix}.$$

- To avoid growth in numbers of columns of B_j (or rows of C_j): column compression by RRLQ or truncated SVD.
- Several options to incorporate scaling, e.g., scale "A"-iteration only.
- Simple stopping cirterion: $\|A_j + I_n\|_F \leq tol$.

Numerical Methods for Solving Lyapunov Equations
The ADI Method

Recall Peaceman Rachford ADI:
Consider $Au = s$ where $A \in \mathbb{R}^{n \times n}$ spd, $s \in \mathbb{R}^n$. ADI Iteration Idea:
Decompose $A = H + V$ with $H, V \in \mathbb{R}^{n \times n}$ such that

$$(H + pI)v = r$$
$$(V + pI)w = t$$

can be solved easily/efficiently.

ADI Iteration

If H, V spd $\Rightarrow \exists p_k,\ k = 1, 2, \ldots$ such that

$$\begin{aligned} u_0 &= 0 \\ (H + p_k I)u_{k-\frac{1}{2}} &= (p_k I - V)u_{k-1} + s \\ (V + p_k I)u_k &= (p_k I - H)u_{k-\frac{1}{2}} + s \end{aligned}$$

converges to $u \in \mathbb{R}^n$ solving $Au = s$.

Numerical Methods for Solving Lyapunov Equations
The ADI Method

Recall Peaceman Rachford ADI:
Consider $Au = s$ where $A \in \mathbb{R}^{n \times n}$ spd, $s \in \mathbb{R}^n$. ADI Iteration Idea:
Decompose $A = H + V$ with $H, V \in \mathbb{R}^{n \times n}$ such that

$$\begin{array}{c}(H + pI)v = r \\ (V + pI)w = t\end{array}$$

can be solved easily/efficiently.

ADI Iteration

If H, V spd $\Rightarrow \exists p_k,\ k = 1, 2, \ldots$ such that

$$\begin{array}{rcl} u_0 &=& 0 \\ (H + p_k I)u_{k-\frac{1}{2}} &=& (p_k I - V)u_{k-1} + s \\ (V + p_k I)u_k &=& (p_k I - H)u_{k-\frac{1}{2}} + s \end{array}$$

converges to $u \in \mathbb{R}^n$ solving $Au = s$.

Numerical Methods for Solving Lyapunov Equations

The Lyapunov operator

$$\mathcal{L}: \quad P \quad \mapsto \quad AX + XA^T$$

can be decomposed into the linear operators

$$\mathcal{L}_H : X \mapsto AX, \qquad \mathcal{L}_V : X \mapsto XA^T.$$

In analogy to the standard ADI method we find the

ADI iteration for the Lyapunov equation [Wachspress '88]

$$\begin{aligned} X_0 &= 0 \\ (A + p_k I)X_{k-\frac{1}{2}} &= -W - X_{k-1}(A^T - p_k I) \\ (A + p_k I)X_k^T &= -W - X_{k-\frac{1}{2}}^T(A^T - p_k I). \end{aligned}$$

Numerical Methods for Solving Lyapunov Equations
Low-Rank ADI

Consider $AX + XA^T = -BB^T$ for stable A, $B \in \mathbb{R}^{n\times m}$ with $m \ll n$.

ADI iteration for the Lyapunov equation [Wachspress '95]

For $k = 1, \ldots, k_{\max}$

$$\begin{aligned} X_0 &= 0 \\ (A + p_k I)X_{k-\frac{1}{2}} &= -BB^T - X_{k-1}(A^T - p_k I) \\ (A + p_k I)X_k^T &= -BB^T - X_{k-\frac{1}{2}}^T(A^T - p_k I) \end{aligned}$$

Rewrite as one step iteration and factorize $X_k = Z_k Z_k^T$, $k = 0, \ldots, k_{\max}$

$$\begin{aligned} Z_0 Z_0^T &= 0 \\ Z_k Z_k^T &= -2p_k(A + p_k I)^{-1}BB^T(A + p_k I)^{-T} \\ &\quad +(A + p_k I)^{-1}(A - p_k I)Z_{k-1}Z_{k-1}^T(A - p_k I)^T(A + p_k I)^{-T} \end{aligned}$$

$\ldots \rightsquigarrow$ low-rank Cholesky factor ADI

[Penzl '97/'00, Li/White '99/'02, B./Li/Penzl '99/'08, Gugercin/Sorensen/Antoulas '03]

Numerical Methods for Solving Lyapunov Equations
Low-Rank ADI

Consider $AX + XA^T = -BB^T$ for stable A, $B \in \mathbb{R}^{n\times m}$ with $m \ll n$.

ADI iteration for the Lyapunov equation [Wachspress '95]

For $k = 1, \ldots, k_{\max}$

$$\begin{aligned} X_0 &= 0 \\ (A + p_k I)X_{k-\frac{1}{2}} &= -BB^T - X_{k-1}(A^T - p_k I) \\ (A + p_k I)X_k^T &= -BB^T - X_{k-\frac{1}{2}}^T(A^T - p_k I) \end{aligned}$$

Rewrite as one step iteration and factorize $X_k = Z_k Z_k^T$, $k = 0, \ldots, k_{\max}$

$$\begin{aligned} Z_0 Z_0^T &= 0 \\ Z_k Z_k^T &= -2p_k (A + p_k I)^{-1} BB^T (A + p_k I)^{-T} \\ &\quad + (A + p_k I)^{-1}(A - p_k I) Z_{k-1} Z_{k-1}^T (A - p_k I)^T (A + p_k I)^{-T} \end{aligned}$$

... $\rightsquigarrow$ low-rank Cholesky factor ADI

[Penzl '97/'00, Li/White '99/'02, B./Li/Penzl '99/'08, Gugercin/Sorensen/Antoulas '03]

Numerical Methods for Solving Lyapunov Equations
Low-Rank ADI

Consider $AX + XA^T = -BB^T$ for stable A, $B \in \mathbb{R}^{n \times m}$ with $m \ll n$.

ADI iteration for the Lyapunov equation [WACHSPRESS '95]

For $k = 1, \ldots, k_{\max}$

$$\begin{aligned} X_0 &= 0 \\ (A + p_k I)X_{k-\frac{1}{2}} &= -BB^T - X_{k-1}(A^T - p_k I) \\ (A + p_k I)X_k^T &= -BB^T - X_{k-\frac{1}{2}}^T(A^T - p_k I) \end{aligned}$$

Rewrite as one step iteration and factorize $X_k = Z_k Z_k^T$, $k = 0, \ldots, k_{\max}$

$$\begin{aligned} Z_0 Z_0^T &= 0 \\ Z_k Z_k^T &= -2p_k(A + p_k I)^{-1}BB^T(A + p_k I)^{-T} \\ &\quad +(A + p_k I)^{-1}(A - p_k I)Z_{k-1}Z_{k-1}^T(A - p_k I)^T(A + p_k I)^{-T} \end{aligned}$$

. . . $\rightsquigarrow$ low-rank Cholesky factor ADI

[PENZL '97/'00, LI/WHITE '99/'02, B./LI/PENZL '99/'08, GUGERCIN/SORENSEN/ANTOULAS '03]

Balanced Truncation
Numerical Methods for Solving Lyapunov Equations

$$Z_k = [\sqrt{-2p_k}(A + p_k I)^{-1}B,\ (A + p_k I)^{-1}(A - p_k I)Z_{k-1}]$$

[Penzl '00]

Observing that $(A - p_i I)$, $(A + p_k I)^{-1}$ commute, we rewrite $Z_{k_{\max}}$ as

$$Z_{k_{\max}} = [z_{k_{\max}},\ P_{k_{\max}-1}z_{k_{\max}},\ P_{k_{\max}-2}(P_{k_{\max}-1}z_{k_{\max}}),\ \ldots,\ P_1(P_2 \cdots P_{k_{\max}-1}z_{k_{\max}})],$$

[Li/White '02]

where

$$z_{k_{\max}} = \sqrt{-2p_{k_{\max}}}(A + p_{k_{\max}} I)^{-1}B$$

and

$$P_i := \frac{\sqrt{-2p_i}}{\sqrt{-2p_{i+1}}}\left[I - (p_i + p_{i+1})(A + p_i I)^{-1}\right].$$

Balanced Truncation
Numerical Methods for Solving Lyapunov Equations

$$Z_k = [\sqrt{-2p_k}(A + p_k I)^{-1}B,\ (A + p_k I)^{-1}(A - p_k I)Z_{k-1}]$$

[Penzl '00]

Observing that $(A - p_i I)$, $(A + p_k I)^{-1}$ commute, we rewrite $Z_{k_{\max}}$ as

$$Z_{k_{\max}} = [z_{k_{\max}},\ P_{k_{\max}-1} z_{k_{\max}},\ P_{k_{\max}-2}(P_{k_{\max}-1} z_{k_{\max}}),\ \ldots,\ P_1(P_2 \cdots P_{k_{\max}-1} z_{k_{\max}})],$$

[Li/White '02]

where

$$z_{k_{\max}} = \sqrt{-2p_{k_{\max}}}(A + p_{k_{\max}} I)^{-1}B$$

and

$$P_i := \frac{\sqrt{-2p_i}}{\sqrt{-2p_{i+1}}}\left[I - (p_i + p_{i+1})(A + p_i I)^{-1}\right].$$

Numerical Methods for Solving Lyapunov Equations

Lyapunov equation $0 = AX + XA^T + BB^T$.

Algorithm [Penzl '97/'00, Li/White '99/'02, B. 04, B./Li/Penzl '99/'08]

$$V_1 \leftarrow \sqrt{-2\,\mathrm{Re}\,p_1}(A+p_1 I)^{-1}B, \qquad Z_1 \leftarrow V_1$$

FOR $k = 2, 3, \ldots$

$$V_k \leftarrow \sqrt{\frac{\mathrm{Re}\,p_k}{\mathrm{Re}\,p_{k-1}}}\left(V_{k-1} - (p_k + \overline{p_{k-1}})(A + p_k I)^{-1}V_{k-1}\right)$$

$$Z_k \leftarrow \begin{bmatrix} Z_{k-1} & V_k \end{bmatrix}$$

$$Z_k \leftarrow \mathrm{rrlq}(Z_k, \tau) \qquad \text{column compression}$$

At convergence, $Z_{k_{\max}} Z_{k_{\max}}^T \approx X$, where (without column compression)

$$Z_{k_{\max}} = \begin{bmatrix} V_1 & \ldots & V_{k_{\max}} \end{bmatrix}, \quad V_k = \begin{bmatrix} \, \end{bmatrix} \in \mathbb{C}^{n\times m}.$$

Note: Implementation in real arithmetic possible by combining two steps [B./Li/Penzl '99/'08] or using new idea employing the relation of 2 consecutive complex factors [B./Kürschner/Saak '11].

Numerical Methods for Solving Lyapunov Equations

Lyapunov equation $0 = AX + XA^T + BB^T$.

Algorithm [Penzl '97/'00, Li/White '99/'02, B. 04, B./Li/Penzl '99/'08]

$$V_1 \leftarrow \sqrt{-2\,\mathrm{Re}\, p_1}(A + p_1 I)^{-1} B, \qquad Z_1 \leftarrow V_1$$

FOR $k = 2, 3, \ldots$

$$V_k \leftarrow \sqrt{\frac{\mathrm{Re}\, p_k}{\mathrm{Re}\, p_{k-1}}} \left(V_{k-1} - (p_k + \overline{p_{k-1}})(A + p_k I)^{-1} V_{k-1} \right)$$

$$Z_k \leftarrow \left[\begin{array}{cc} Z_{k-1} & V_k \end{array} \right]$$

$$Z_k \leftarrow \mathrm{rrlq}(Z_k, \tau) \qquad \text{column compression}$$

At convergence, $Z_{k_{\max}} Z_{k_{\max}}^T \approx X$, where (without column compression)

$$Z_{k_{\max}} = \left[\begin{array}{ccc} V_1 & \ldots & V_{k_{\max}} \end{array} \right], \quad V_k = \Big[\,\Big] \in \mathbb{C}^{n \times m}.$$

Note: Implementation in real arithmetic possible by combining two steps [B./Li/Penzl '99/'08] or using new idea employing the relation of 2 consecutive complex factors [B./Kürschner/Saak '11].

Numerical Results for ADI

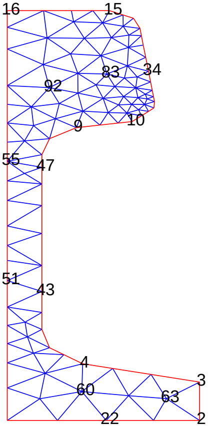
Optimal Cooling of Steel Profiles

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

$$
\begin{aligned}
c \cdot \rho \frac{\partial}{\partial t} x &= \lambda \Delta x, & \xi \in \Omega \\
\lambda \frac{\partial}{\partial n} x &= \kappa (u_k - x), & \xi \in \Gamma_k,\ 1 \leq k \leq 7, \\
\frac{\partial}{\partial n} x &= 0, & \xi \in \Gamma_7.
\end{aligned}
$$

$\Longrightarrow m = 7, p = 6.$

- FEM Discretization, different models for initial mesh ($n = 371$), 1, 2, 3, 4 steps of mesh refinement $\Rightarrow$ $n = 1357, 5177, 20209, 79841.$



Source: Physical model: courtesy of Mannesmann/Demag.

Math. model: Tröltzsch/Unger 1999/2001, Penzl 1999, Saak 2003.

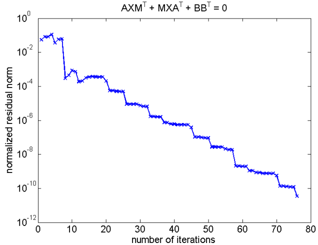
Numerical Results for ADI

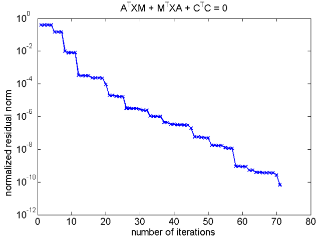
Optimal Cooling of Steel Profiles

- Solve dual Lyapunov equations needed for balanced truncation, i.e.,

$$APM^T + MPA^T + BB^T = 0, \quad A^TQM + M^TQA + C^TC = 0,$$

  for 79, 841.
- 25 shifts chosen by Penzl heuristic from 50/25 Ritz values of A of largest/smallest magnitude, no column compression performed.
- New version in M.E.S.S. requires no factorization of mass matrix!
- Computations done on Core2Duo at 2.8GHz with 3GB RAM and 32Bit-MATLAB.





CPU times: 626 / 356 sec.

Numerical Results for ADI

Scaling / Mesh Independence **Computations by Martin Köhler '10**

- $A \in \mathbb{R}^{n \times n} \equiv$ FDM matrix for 2D heat equation on $[0,1]^2$ (LYAPACK benchmark `demo_l1`, $m = 1$).
- 16 shifts chosen by Penzl heuristic from 50/25 Ritz values of A of largest/smallest magnitude.
- Computations using 2 dual core Intel Xeon 5160 with 16 GB RAM.

Numerical Results for ADI

Scaling / Mesh Independence **Computations by Martin Köhler '10**

- $A \in \mathbb{R}^{n \times n} \equiv$ FDM matrix for 2D heat equation on $[0,1]^2$ (LYAPACK benchmark `demo_l1`, $m = 1$).
- 16 shifts chosen by Penzl heuristic from 50/25 Ritz values of A of largest/smallest magnitude.
- Computations using 2 dual core Intel Xeon 5160 with 16 GB RAM.

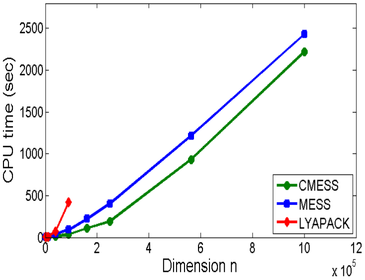
CPU Times

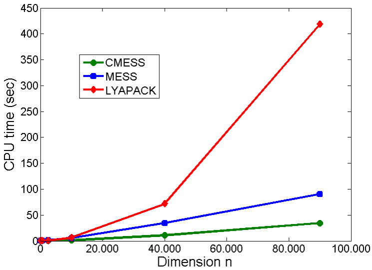
n	M.E.S.S. (C)	LyaPack	M.E.S.S. (MATLAB)
100	0.023	0.124	0.158
625	0.042	0.104	0.227
2,500	0.159	0.702	0.989
10,000	0.965	6.22	5.644
40,000	11.09	71.48	34.55
90,000	34.67	418.5	90.49
160,000	109.3	out of memory	219.9
250,000	193.7	out of memory	403.8
562,500	930.1	out of memory	1216.7
1,000,000	2220.0	out of memory	2428.6

Numerical Results for ADI

Scaling / Mesh Independence **Computations by Martin Köhler '10**

- $A \in \mathbb{R}^{n \times n} \equiv$ FDM matrix for 2D heat equation on $[0,1]^2$ (LYAPACK benchmark `demo_l1`, $m = 1$).
- 16 shifts chosen by Penzl heuristic from 50/25 Ritz values of A of largest/smallest magnitude.
- Computations using 2 dual core Intel Xeon 5160 with 16 GB RAM.





Note: for $n = 1,000,000$, first sparse LU needs $\sim 1,100$ sec., using UMFPACK this reduces to 30 sec.

Factored Galerkin-ADI Iteration

Lyapunov equation $0 = AX + XA^T + BB^T$

Projection-based methods for Lyapunov equations with $A + A^T < 0$:

1. Compute orthonormal basis $\mathrm{range}\,(Z)$, $Z \in \mathbb{R}^{n \times r}$, for subspace $\mathcal{Z} \subset \mathbb{R}^n$, $\dim \mathcal{Z} = r$.
2. Set $\hat{A} := Z^T A Z$, $\hat{B} := Z^T B$.
3. Solve small-size Lyapunov equation $\hat{A}\hat{X} + \hat{X}\hat{A}^T + \hat{B}\hat{B}^T = 0$.
4. Use $X \approx Z\hat{X}Z^T$.

Examples:

- Krylov subspace methods, i.e., for $m = 1$:

$$\mathcal{Z} = \mathcal{K}(A, B, r) = \mathrm{span}\{B, AB, A^2B, \ldots, A^{r-1}B\}$$

[Saad '90, Jaimoukha/Kasenally '94, Jbilou '02–'08].

- K-PIK [Simoncini '07],

$$\mathcal{Z} = \mathcal{K}(A, B, r) \cup \mathcal{K}(A^{-1}, B, r).$$

Factored Galerkin-ADI Iteration

Lyapunov equation $0 = AX + XA^T + BB^T$

Projection-based methods for Lyapunov equations with $A + A^T < 0$:

1. Compute orthonormal basis $\mathrm{range}(Z)$, $Z \in \mathbb{R}^{n \times r}$, for subspace $\mathcal{Z} \subset \mathbb{R}^n$, $\dim \mathcal{Z} = r$.
2. Set $\hat{A} := Z^T AZ$, $\hat{B} := Z^T B$.
3. Solve small-size Lyapunov equation $\hat{A}\hat{X} + \hat{X}\hat{A}^T + \hat{B}\hat{B}^T = 0$.
4. Use $X \approx Z\hat{X}Z^T$.

Examples:

- Krylov subspace methods, i.e., for $m = 1$:

$$\mathcal{Z} = \mathcal{K}(A, B, r) = \mathrm{span}\{B, AB, A^2B, \ldots, A^{r-1}B\}$$

  [Saad '90, Jaimoukha/Kasenally '94, Jbilou '02–'08].

- K-PIK [Simoncini '07],

$$\mathcal{Z} = \mathcal{K}(A, B, r) \cup \mathcal{K}(A^{-1}, B, r).$$

Factored Galerkin-ADI Iteration

Lyapunov equation $0 = AX + XA^T + BB^T$

Projection-based methods for Lyapunov equations with $A + A^T < 0$:

1. Compute orthonormal basis $\mathrm{range}(Z)$, $Z \in \mathbb{R}^{n \times r}$, for subspace $\mathcal{Z} \subset \mathbb{R}^n$, $\dim \mathcal{Z} = r$.
2. Set $\hat{A} := Z^T A Z$, $\hat{B} := Z^T B$.
3. Solve small-size Lyapunov equation $\hat{A}\hat{X} + \hat{X}\hat{A}^T + \hat{B}\hat{B}^T = 0$.
4. Use $X \approx Z\hat{X}Z^T$.

Examples:

- ADI subspace [B./R.-C. Li/Truhar '08]:

$$\mathcal{Z} = \mathrm{colspan}\left[\ V_1, \quad \ldots, \quad V_r\ \right].$$

Note:

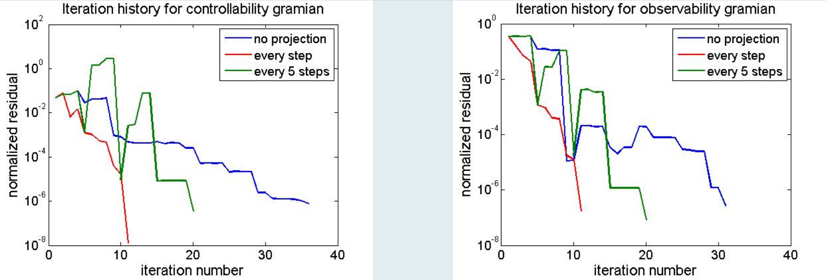
1. ADI subspace is rational Krylov subspace [J.-R. Li/White '02].
2. Similar approach: ADI-preconditioned global Arnoldi method [Jbilou '08].

Numerical Methods for Solving Lyapunov Equations
Numerical examples for Galerkin-ADI

FEM semi-discretized control problem for parabolic PDE:

- optimal cooling of rail profiles,
- $n = 20,209,\ m = 7,\ p = 6.$

Good ADI shifts



CPU times: 80s (projection every 5th ADI step) vs. 94s (no projection).

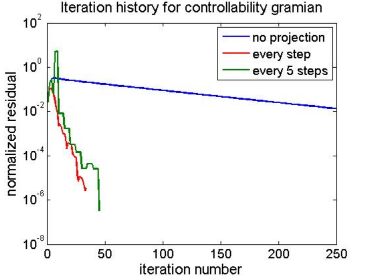
Computations by Jens Saak '10.

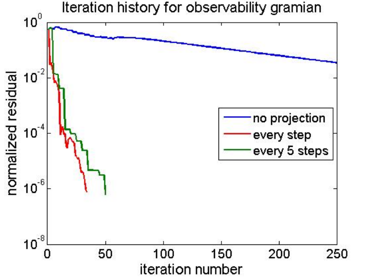
Numerical Methods for Solving Lyapunov Equations
Numerical examples for Galerkin-ADI

FEM semi-discretized control problem for parabolic PDE:

- optimal cooling of rail profiles,
- $n = 20,209,\ m = 7,\ p = 6.$

Bad ADI shifts





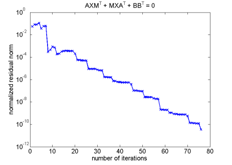
CPU times: 368s (projection every 5th ADI step) vs. 1207s (no projection).

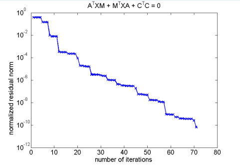
Computations by Jens Saak '10.

Numerical Methods for Solving Lyapunov Equations

Numerical examples for Galerkin-ADI: optimal cooling of rail profiles, $n = 79,841$.

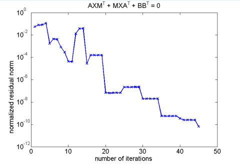
MESS w/o Galerkin projection and column compression

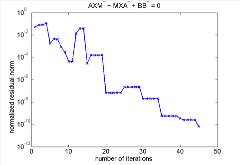




Rank of solution factors: 532 / 426

MESS with Galerkin projection and column compression



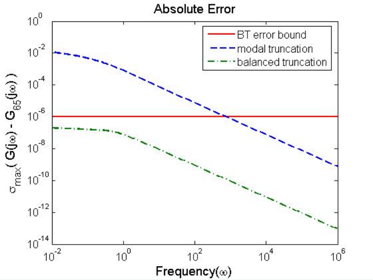


Rank of solution factors: 269 / 205

Balanced Truncation

Numerical example for BT: Optimal Cooling of Steel Profiles

$n = 1357$, Absolute Error

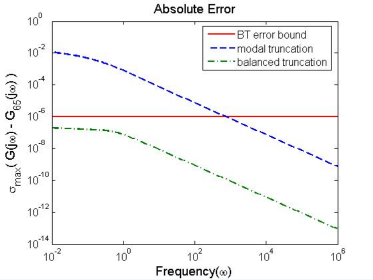


- BT model computed with sign function method,
- MT w/o static condensation, same order as BT model.

Balanced Truncation

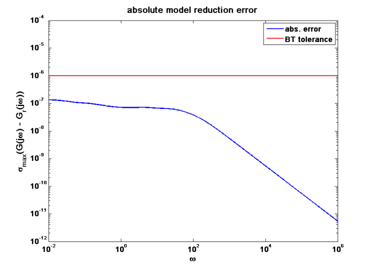
Numerical example for BT: Optimal Cooling of Steel Profiles

$n = 1357$, Absolute Error



- BT model computed with sign function method,
- MT w/o static condensation, same order as BT model.

$n = 79841$, Absolute Error



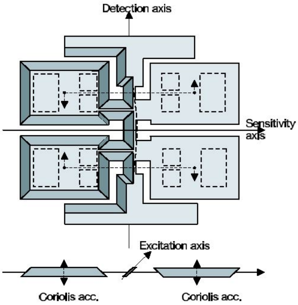
- BT model computed using M.E.S.S. in MATLAB,
- dualcore, computation time: <10 min.

Balanced Truncation

Numerical example for BT: Microgyroscope (Butterfly Gyro)

- By applying AC voltage to electrodes, wings are forced to vibrate in anti-phase in wafer plane.
- Coriolis forces induce motion of wings out of wafer plane yielding sensor data.

- Vibrating micro-mechanical gyroscope for inertial navigation.
- Rotational position sensor.



Source: The Oberwolfach Benchmark Collection http://www.imtek.de/simulation/benchmark

Courtesy of D. Billger (Imego Institute, Göteborg), Saab Bofors Dynamics AB.

Balanced Truncation

Numerical example for BT: Microgyroscope (Butterfly Gyro)

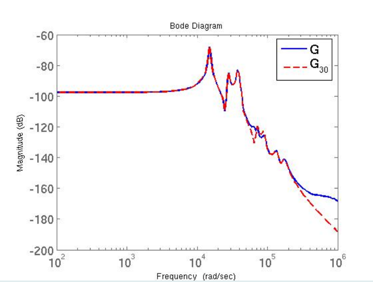
- FEM discretization of structure dynamical model using quadratic tetrahedral elements (ANSYS-SOLID187)
  $\rightsquigarrow$ $n = 34,722$, $m = 1$, $p = 12$.
- Reduced model computed using SpaRed, $r = 30$.

Balanced Truncation

Numerical example for BT: Microgyroscope (Butterfly Gyro)

- FEM discretization of structure dynamical model using quadratic tetrahedral elements (ANSYS-SOLID187)
  $\rightsquigarrow$ $n = 34,722$, $m = 1$, $p = 12$.
- Reduced model computed using SpaRed, $r = 30$.

Frequency Repsonse Analysis

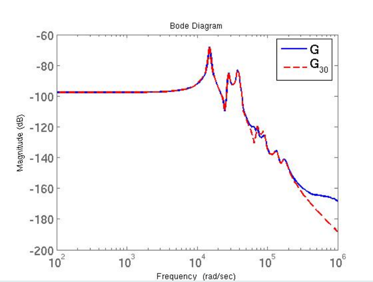


Balanced Truncation

Numerical example for BT: Microgyroscope (Butterfly Gyro)

- FEM discretization of structure dynamical model using quadratic tetrahedral elements (ANSYS-SOLID187)
  $\rightsquigarrow$ $n = 34,722$, $m = 1$, $p = 12$.
- Reduced model computed using SpaRed, $r = 30$.

Frequency Repsonse Analysis



Hankel Singular Values