MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

CSC

COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

Towards high performance IRKA on hybrid CPU-GPU systems

Jens Saak

in collaboration with

Georg Pauer
(OVGU/MPI Magdeburg)

Kapil Ahuja, Ruchir Garg
(IIT Indore)

Hartwig Anzt, Jack Dongarra
(ICL Uni Tennessee Knoxville)

Edmond Chow
(Georgia Tech)

17th GAMM Workshop on Applied and Numerical Linear Algebra
Cologne, Germany
September 7-8, 2017

Linear time-invariant (LTI) system:

$$E\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t)$$

$$\begin{aligned} E_r &= W^T E V \\ A_r &= W^T A V \\ B_r &= W^T B \\ C_r &= C V \end{aligned}$$

$x(t) \in \mathbb{R}^n$, $x_r(t) \in \mathbb{R}^r$, $u \in \mathbb{R}^p$, $y, \tilde{y} \in \mathbb{R}^q$

Reduced order model (ROM):

$$E_r \dot{x}_r(t) = A_r x_r(t) + B_r u(t)$$

$$\tilde{y}(t) = C_r x_r(t)$$

MOR of LTI systems often employs the rational Krylov subspace method (RKSM) of some flavor:

- Moment Matching (V, W span rational Krylov subspaces),
- IRKA (iterative rational Krylov algorithm)
- Balanced Truncation (computing the Gramians via ADI or RKSM)

MOR of LTI systems often employs the rational Krylov subspace method (RKSM) of some flavor:

- Moment Matching (V, W span rational Krylov subspaces),
- **IRKA** (iterative rational Krylov algorithm)
- Balanced Truncation (computing the Gramians via ADI or RKSM)

IRKA in a nutshell **e.g.[Gugercin/Antoulas/Beattie '08]**

1. Guess initial ROM or initial shifts,
2. Compute
$$\operatorname{span} V = \operatorname{span}\{(\sigma_1 E - A)^{-1}B, \ldots, (\sigma_r E - A)^{-1}B\},$$
$$\operatorname{span} W = \operatorname{span}\{(\sigma_1 E^T - A^T)^{-1}C^T, \ldots, (\sigma_r E^T - A^T)^{-1}C^T\}.$$
3. Update ROM, set $\sigma = -\Lambda(A_r, E_r)$, repeat until σ has converged.

MOR of LTI systems often employs the rational Krylov subspace method (RKSM) of some flavor:

- Moment Matching (V, W span rational Krylov subspaces),
- IRKA (iterative rational Krylov algorithm)
- Balanced Truncation (computing the Gramians via ADI or RKSM)

Dominant computation in all these methods

Sequences of Shifted Linear Systems (SLS)

$$(A + \sigma_j E)\, X_j = \quad Y_j, \qquad \text{for } j \in \mathcal{I} \subset \mathbb{N},$$

with real data $A \in \mathbb{R}^{n \times n}$, $E \in \mathbb{R}^{n \times n}$, $Y_j \in \mathbb{R}^{n \times m}$, $m \ll n$, but possibly complex shifts $\sigma_j \in \mathbb{C}$.

MOR of LTI systems often employs the rational Krylov subspace method (RKSM) of some flavor:

- Moment Matching (V, W span rational Krylov subspaces),
- IRKA (iterative rational Krylov algorithm)
- Balanced Truncation (computing the Gramians via ADI or RKSM)

Dominant computation in all these methods

Sequences of Shifted Linear Systems (SLS)

$$P_j\,(A+\sigma_j E)\,X_j = P_j Y_j, \qquad \text{for } j \in \mathcal{I} \subset \mathbb{N},$$

with real data $A \in \mathbb{R}^{n\times n}$, $E \in \mathbb{R}^{n\times n}$, $Y_j \in \mathbb{R}^{n\times m}$, $m \ll n$, but possibly complex shifts $\sigma_j \in \mathbb{C}$. Preconditioner $P_j \in \mathbb{R}^{n\times n}$ usually needs update for every j.

Sequences of Shifted Linear Systems (SLS)

$$P_j\,(A + \sigma_j E)\, X_j = P_j Y_j, \qquad \text{for } j \in \mathcal{I} \subset \mathbb{N},$$

We aim at using iterative solvers on hybrid CPU-GPU systems.

Efficiency improving strategies

1. recycle / realign preconditioner P_j
2. recycle solvers / subspaces
3. solve shifted linear systems in parallel

Sequences of Shifted Linear Systems (SLS)

$$P_j\,(A+\sigma_j E)\,X_j = P_j Y_j, \qquad \text{for } j \in \mathcal{I} \subset \mathbb{N},$$

We aim at using iterative solvers on hybrid CPU-GPU systems.

Efficiency improving strategies

1. recycle / realign preconditioner P_j

   extending [Anzt/Chow/S./Dongarra '16]

2. recycle solvers / subspaces

3. solve shifted linear systems in parallel

Sequences of Shifted Linear Systems (SLS)

$$P_j\,(A+\sigma_j E)\,X_j = P_j Y_j, \qquad \text{for } j \in \mathcal{I} \subset \mathbb{N},$$

We aim at using iterative solvers on hybrid CPU-GPU systems.

Efficiency improving strategies

1. recycle / realign preconditioner P_j

   extending [Anzt/Chow/S./Dongarra '16]

2. recycle solvers / subspaces

   following [Ahuja/de Sturler/Gugercin/Chang '12]

3. solve shifted linear systems in parallel

1. reuse preconditioner for small matrix changes, i.e. similar shifts
2. update incomplete factorization preconditioner
   [Bellavia/De Simone/di Serafino/Morini '11, Benzi/Bertaccini '03, Bertaccini '04, Calgaro/Chehab/Saad '10, Duintjer Tebbens/Tüma '07]

Main Issues

- reuses reduce efficiency of the preconditioner
- updates have limited potential for parallel execution

Goal

highly parallel preconditioner update and application

Preconditioned iterative solution of

Sequences of Shifted Linear Systems (SLS)

$$P_j\left(A+\sigma_j E\right)X_j = P_j Y_j, \qquad \text{for } j \in \mathcal{I} \subset \mathbb{N},$$

for A, E symmetric and σ_j real.

Idea: Minimize $\left\|A - U^T U\right\|$ via fixed point approach [Chow/et al. '15]

Preconditioned iterative solution of

Sequences of Shifted Linear Systems (SLS)

$$P_j\,(A + \sigma_j E)\,X_j = P_j Y_j, \qquad \text{for } j \in \mathcal{I} \subset \mathbb{N},$$

for A, E symmetric and σ_j real.

Idea: Minimize $\left\|A - U^T U\right\|$ via fixed point approach [Chow/et al. '15]

Initialization

If $|\sigma_j - \sigma_{j-1}|$ is small enough, then $\|P_j - P_{j-1}\|$ is small
$\Rightarrow P_{j-1}$ good initial guess for P_j.

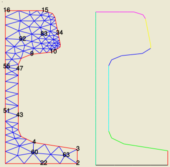
Use [Chow/Patel '15, Chow/Anzt/Dongarra '15] for (asynchronous) iterative computation of incomplete factorization preconditioners.

Reusing and Realigning Preconditioners

The Real aka. Symmetric Case

	Time [ms]				Speedup		
		Number of sweeps			Number of sweeps		
	IC	1	3	5	1	3	5
371	4.88	0.03	0.06	0.08	167.70	87.14	57.48
1 357	9.01	0.03	0.06	0.08	309.62	158.07	107.39
5 177	15.60	0.04	0.08	0.12	421.62	200.00	134.48
20 209	32.00	0.08	0.19	0.30	405.58	167.54	105.26
79 841	66.40	0.23	0.62	1.01	286.21	106.58	65.74
317 377	142.00	0.78	2.25	3.73	182.75	63.11	38.07
1 265 537	323.00	2.94	8.77	14.50	109.86	36.83	22.28

Test example:
Oberwolfach Collection Steel Cooling Benchmark[Benner/S. '05]
here using the FEniCS reimplementation by Maximilian Behr



Consider

$$\mathcal{A}x = b$$

for $\mathcal{A} \in \mathbb{C}^{n\times n}$, x, $b \in \mathbb{C}^n$.
With $\mathcal{A}_\mathfrak{r}$, $\mathcal{A}_\mathfrak{i} \in \mathbb{R}^{n\times n}$, $x_\mathfrak{r}$, $x_\mathfrak{i}$, $b_\mathfrak{r}$, $b_\mathfrak{i} \in \mathbb{R}^n$ write

$$\mathcal{A} = \mathcal{A}_\mathfrak{r} + \imath\mathcal{A}_\mathfrak{i}, \quad x = x_\mathfrak{r} + \imath x_\mathfrak{i}, \quad b = b_\mathfrak{r} + \imath b_\mathfrak{i}.$$

Idea **e.g.[**Day/Heroux '01, Benzi/Bertaccini '08**]**

Rewrite to real linear system

$$\begin{bmatrix} \mathcal{A}_\mathfrak{r} & -\mathcal{A}_\mathfrak{i} \\ \mathcal{A}_\mathfrak{i} & \mathcal{A}_\mathfrak{r} \end{bmatrix} \begin{bmatrix} x_\mathfrak{r} \\ x_\mathfrak{i} \end{bmatrix} = \begin{bmatrix} b_\mathfrak{r} \\ b_\mathfrak{i} \end{bmatrix}$$

and employ block preconditioner.

In our special case

$$\mathcal{A}_{j,r} = A + \mathrm{Re}(\sigma_j)\, E, \qquad \mathcal{A}_{j,i} = \mathrm{Im}(\sigma_j)\, E$$

Let

$$J_j = \begin{bmatrix} 0 & -\mathrm{Im}(\sigma_j) \\ \mathrm{Im}(\sigma_j) & 0 \end{bmatrix}.$$

Then

$$\mathcal{A}_j \tilde{V}_j = \tilde{W}_j \quad \Leftrightarrow \quad \begin{bmatrix} \mathcal{A}_{j,\mathfrak{r}} & -\mathcal{A}_{j,\mathfrak{i}} \\ \mathcal{A}_{j,\mathfrak{i}} & \mathcal{A}_{j,\mathfrak{r}} \end{bmatrix} \begin{bmatrix} V_{j,\mathfrak{r}} \\ V_{j,\mathfrak{i}} \end{bmatrix} = \begin{bmatrix} W_{j,\mathfrak{r}} \\ W_{j,\mathfrak{i}} \end{bmatrix}$$

$$\Leftrightarrow \quad (I_2 \otimes \mathcal{A}_{j,\mathfrak{r}} + J_j \otimes E)\tilde{V}_j = \tilde{W}_j$$

In our special case

$$\mathcal{A}_{j,r} = A + \mathrm{Re}(\sigma_j)\, E, \qquad \mathcal{A}_{j,i} = \mathrm{Im}(\sigma_j)\, E$$

Let

$$J_j = \begin{bmatrix} 0 & -\mathrm{Im}(\sigma_j) \\ \mathrm{Im}(\sigma_j) & 0 \end{bmatrix}.$$

Then

$$\mathcal{A}_j \tilde{V}_j = \tilde{W}_j \quad \Leftrightarrow \quad \begin{bmatrix} \mathcal{A}_{j,\mathfrak{r}} & -\mathcal{A}_{j,\mathfrak{i}} \\ \mathcal{A}_{j,\mathfrak{i}} & \mathcal{A}_{j,\mathfrak{r}} \end{bmatrix} \begin{bmatrix} V_{j,\mathfrak{r}} \\ V_{j,\mathfrak{i}} \end{bmatrix} = \begin{bmatrix} W_{j,\mathfrak{r}} \\ W_{j,\mathfrak{i}} \end{bmatrix}$$
$$\Leftrightarrow \quad (I_2 \otimes \mathcal{A}_{j,\mathfrak{r}} + J_j \otimes E)\tilde{V}_j = \tilde{W}_j$$

Idea: $P := (I_2 \otimes \mathcal{A}_{j,\mathfrak{r}})^{-1} = \left(I_2 \otimes \mathcal{A}_{j,\mathfrak{r}}^{-1}\right)$

In our special case

$$\mathcal{A}_{j,r} = A + \mathrm{Re}(\sigma_j)\,E, \qquad \mathcal{A}_{j,i} = \mathrm{Im}(\sigma_j)\,E$$

Let

$$J_j = \begin{bmatrix} 0 & -\mathrm{Im}(\sigma_j) \\ \mathrm{Im}(\sigma_j) & 0 \end{bmatrix}.$$

counter diagonal blocks are:

$$\pm\mathrm{Im}(\sigma_j)\,(A + \mathrm{Re}(\sigma_j)\,E)^{-1}E = \frac{\pm\mathrm{Im}(\sigma_j)}{\mathrm{Re}(\sigma_j)}(A + \mathrm{Re}(\sigma_j)\,E)^{-1}\,(A + \mathrm{Re}(\sigma_j)\,E - A)$$
$$= \frac{\pm\mathrm{Im}(\sigma_j)}{\mathrm{Re}(\sigma_j)}\,\left(I_n - (A + \mathrm{Re}(\sigma_j)\,E)^{-1}A\right).$$

Then

$$\mathcal{A}_j\tilde{V}_j = \tilde{W}_j \quad \Leftrightarrow \quad \begin{bmatrix} \mathcal{A}_{j,r} & -\mathcal{A}_{j,i} \\ \mathcal{A}_{j,i} & \mathcal{A}_{j,r} \end{bmatrix}\begin{bmatrix} V_{j,r} \\ V_{j,i} \end{bmatrix} = \begin{bmatrix} W_{j,r} \\ W_{j,i} \end{bmatrix}$$
$$\Leftrightarrow \quad (I_2 \otimes \mathcal{A}_{j,\mathfrak{r}} + J_j \otimes E)\tilde{V}_j = \tilde{W}_j$$

Idea: $P := (I_2 \otimes \mathcal{A}_{j,\mathfrak{r}})^{-1} = \left(I_2 \otimes \mathcal{A}_{j,\mathfrak{r}}^{-1}\right)$

P3-GMRES iterations for SLS (tol=10^{-8}, maxiter=250)

	$-3.71\,10^2 + \imath\,2.39\,10^2/10^{-2}/10^{-6}$			$-3.71\,10^{-6}/10^4 + 2.39\,10^2\imath$	
no prec.	174	136	136	250	11
block ILU(0)	54	41	41	130	5
block ILU(1)	35	26	26	85	4
block ILU(2)	26	19	19	65	4
block ILU(3)	21	14	14	54	4
block \	15	2	1	38	4
full ILU(0)	52	41	41	125	5

[1] model data by Maximilian Behr

Recycling Solvers and Subspaces

Recall IRKA needs:

$$\operatorname{span} V = \operatorname{span}\{(\sigma_1 E - A)^{-1}B, \dots, (\sigma_k E - A)^{-1}B\},$$
$$\operatorname{span} W = \operatorname{span}\{(\sigma_1 E^T - A^T)^{-1}C^T, \dots, (\sigma_k E^T - A^T)^{-1}C^T\}.$$

Recall IRKA needs:

$$\begin{aligned}\operatorname{span} V &= \operatorname{span}\{(\sigma_1 E - A)^{-1}B, \ldots, (\sigma_k E - A)^{-1}B\},\\ \operatorname{span} W &= \operatorname{span}\{(\sigma_1 E^T - A^T)^{-1}C^T, \ldots, (\sigma_k E^T - A^T)^{-1}C^T\}.\end{aligned}$$

Idea [Ahuja/de Sturler/Gugercin/Chang '12]

- Need to solve with dual operators $\rightsquigarrow$ Use BiCG

Recycling Solvers and Subspaces

Recall IRKA needs:

$$\operatorname{span} V = \operatorname{span}\{(\sigma_1 E - A)^{-1}B, \dots, (\sigma_k E - A)^{-1}B\},$$
$$\operatorname{span} W = \operatorname{span}\{(\sigma_1 E^T - A^T)^{-1}C^T, \dots, (\sigma_k E^T - A^T)^{-1}C^T\}.$$

Idea [Ahuja/de Sturler/Gugercin/Chang '12]

- Need to solve with dual operators $\rightsquigarrow$ Use BiCG
- MIMO systems (i.e., $p, q > 1$) repeatedly solve with different right hand sides $\rightsquigarrow$ update subspace, i.e. recycle

Recycling Solvers and Subspaces

Recall IRKA needs:

$$\operatorname{span} V = \operatorname{span}\{(\sigma_1 E - A)^{-1}B, \ldots, (\sigma_k E - A)^{-1}B\},$$
$$\operatorname{span} W = \operatorname{span}\{(\sigma_1 E^T - A^T)^{-1}C^T, \ldots, (\sigma_k E^T - A^T)^{-1}C^T\}.$$

Idea [Ahuja/de Sturler/Gugercin/Chang '12]

- Need to solve with dual operators $\rightsquigarrow$ Use BiCG
- MIMO systems (i.e., $p, q > 1$) repeatedly solve with different right hand sides $\rightsquigarrow$ update subspace, i.e. recycle
- upon convergence: $\sigma_j^{(i)} \approx \sigma_j^{(i+1)}$ $\rightsquigarrow$ employ deflation, i.e., recycle

Recall IRKA needs:

$$\operatorname{span} V = \operatorname{span}\{(\sigma_1 E - A)^{-1}B, \dots, (\sigma_k E - A)^{-1}B\},$$
$$\operatorname{span} W = \operatorname{span}\{(\sigma_1 E^T - A^T)^{-1}C^T, \dots, (\sigma_k E^T - A^T)^{-1}C^T\}.$$

Idea [Ahuja/de Sturler/Gugercin/Chang '12]

- Need to solve with dual operators $\rightsquigarrow$ Use BiCG
- MIMO systems (i.e., $p, q > 1$) repeatedly solve with different right hand sides $\rightsquigarrow$ update subspace, i.e. recycle
- upon convergence: $\sigma_j^{(i)} \approx \sigma_j^{(i+1)}$ $\rightsquigarrow$ employ deflation, i.e., recycle

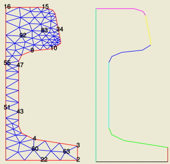
RBiCG for IRKA

Current focus with Kapil Ahuja, Ruchir Garg and Georg Pauer: CUDA accelerated version

MATLAB® results [Ahuja/de Sturler/Gugercin/Chang '12]

n	σ	r	drop tol	IRKA steps	iteration count BiCG	iteration count RBiCG	iteration count Ratio	time(s) BiCG	time(s) RBiCG	time(s) Ratio
20209	40	20	0.01	31	3032	1434	2.11	73.82	54.28	1.36
79841	50	20	0.005	44	6324	2547	2.48	742.83	505.09	1.47

Test example:
Oberwolfach Collection Steel Cooling Benchmark[Benner/S. '05]
avilable from:
`https://portal.uni-freiburg.de/imteksimulation/downloads/benchmark/Steel%20Profiles%20%2838881%29`



Next: Parallel Solution of the Shifted Systems

Observation:

Iterative solver needs sparse matrix vector product $(A + \sigma_j E)x_j = y_j$.

$\rightsquigarrow$ memory bandwidth bounded.

Idea:

Define: $X = [x_1, \ldots, x_r]$, $Y = [y_1, \ldots, y_r]$ and $\Sigma = \operatorname{diag}(\sigma)$

Then

$$Y = AX + EX\Sigma,$$

and

- sparse matrix vector product $\rightsquigarrow$ sparse-dense matrix matrix product
- $2r$ sparse matrix sweeps become 2 sparse matrix sweeps

Requires short recurrence method (e.g. BiCG) to limit memory usage

Conclusions & Outlook

Conclusions

- analysis of the problem structure gives rise to 3 promising implementation candidates
- Preconditioner realignment and Subspace recycling can even be combined

Outlook

- proper HPC implementation pending
- comparison of the strategies $\rightsquigarrow$ Master Thesis Georg Pauer

Conclusions & Outlook

Conclusions

- analysis of the problem structure gives rise to 3 promising implementation candidates
- Preconditioner realignment and Subspace recycling can even be combined

Outlook

- proper HPC implementation pending
- comparison of the strategies $\rightsquigarrow$ Master Thesis Georg Pauer

Thank you for your attention!