Parallel Order Reduction via Balanced Truncation for Optimal Cooling of Steel Profiles*

José M. Badía[1], Peter Benner[2], Rafael Mayo[1], Enrique S. Quintana-Ortí[1], Gregorio Quintana-Ortí[1], and Jens Saak[2]

[1] Depto. de Ingeniería y Ciencia de Computadores, Universidad Jaume I
12.071–Castellón, Spain
{badia,mayo,quintana,gquintan}@icc.uji.es
[2] Fakultät für Mathematik, Technische Universität Chemnitz
D-09107 Chemnitz, Germany
{benner,jens.saak}@mathematik.tu-chemnitz.de

Abstract. We employ two efficient parallel approaches to reduce a model arising from a semi-discretization of a controlled heat transfer process for optimal cooling of a steel profile. Both algorithms are based on balanced truncation but differ in the numerical method that is used to solve two dual generalized Lyapunov equations, which is the major computational task. Experimental results on a cluster of Intel Xeon processors compare the efficacy of the parallel model reduction algorithms.

Keywords: Linear dynamical systems, 2D heat equation, model reduction, balanced truncation, generalized Lyapunov equations

1 Introduction

We consider the problem of optimal cooling of steel profiles. This problem arises in a rolling mill where different phases in the production process require different temperatures of the raw material. To achieve a high production rate, the temperature has to be reduced rapidly to the level required by the next phase. The cooling process, accelerated by spraying cooling fluids on the surface of the profile, has to be controlled since large gradients in the temperature lead to unwanted deformations, brittleness, loss of rigidity, and other undesirable properties.

The heat distribution in the profile is modeled by an instationary linear heat equation. The standard Galerkin approach for discretizing the heat transfer

* J.M. Badía, R. Mayo, E.S. Quintana-Ortí, and G. Quintana-Ortí were supported by the CICYT project No. TIC2002-004400-C03-01 and FEDER, and project No. P1B-2004-6 of the *Fundación Caixa-Castellón/Bancaixa and UJI*. P. Benner and J. Saak were supported by the DFG Sonderforschungsbereich 393 *Parallele Numerische Simulation für Physik und Kontinuumsmechanik* at TU Chemnitz.

J.C. Cunha and P.D. Medeiros (Eds.): Euro-Par 2005, LNCS 3648, pp. 857–866, 2005.

model in space, as described in Section 1.1, results in a first-order ordinary differential equation of the form:

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), \quad t > 0, \quad x(0) = x^0, \\ y(t) &= Cx(t) + Du(t), \quad t \geq 0. \end{aligned} \tag{1}$$

Here, $x^0 \in \mathbb{R}^n$ contains the initial temperature distribution in the profile, $u(t)$ and $y(t)$ are vectors for the inputs (i.e., temperatures of the cooling fluid) and outputs (i.e., approximate temperature gradients) of the system, respectively, and $E, A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $D \in \mathbb{R}^{p\times m}$. The system in (1) can also be modeled by the transfer function matrix (TFM) $G(s) = C(sE - A)^{-1}B + D$. The number of states, n, is known as the state-space dimension (or the order) of the system.

The goal of model reduction is to find a reduced-order LTI system,

$$\begin{aligned} \hat{E}\dot{\hat{x}}(t) &= \hat{A}\hat{x}(t) + \hat{B}u(t), \quad t > 0, \quad \hat{x}(0) = \hat{x}^0, \\ \hat{y}(t) &= \hat{C}\hat{x}(t) + \hat{D}u(t), \quad t \geq 0, \end{aligned} \tag{2}$$

of order r, with $r \ll n$, and associated TFM $\hat{G}(s) = \hat{C}(sE - \hat{A})^{-1}\hat{B} + \hat{D}$ which approximates $G(s)$. The reduced-order realization can then replace the original high-order system in the design of the optimal controller, thus simplifying by much this phase of the problem.

It is the goal of this paper to demonstrate that model reduction algorithms based on balanced truncation (BT) can effectively be applied to problems of size up to $n = 100,000$ (and even higher) when appropriate numerical algorithms are used.

Traditional algorithms for BT model reduction are available, e.g., in libraries such as SLICOT[1] or the MATLAB control-related toolboxes, and can be employed to reduce models with a few hundreds of state-space variables on current desktop computers. Here we employ two different parallel BT algorithms that allow the reduction of much larger systems, as those arising in the optimal cooling problem. These algorithms are integrated into the parallel libraries for model reduction of large-scale dense and sparse linear systems, PLiCMR [3] and SpaRed [1], respectively.

The paper is structured as follows. We conclude this section by describing with some detail the discretization procedure of the heat transfer model for the steel profile. In Section 2 we review our algorithms for BT model reduction of linear systems. In Section 3 we briefly describe the multilayered architecture of libraries employed by our model reduction algorithms. Finally, the efficacy of the algorithms is reported in Section 4, and some concluding remarks follow in Section 5.

1.1 Discretization of the Optimal Cooling Problem

The optimal cooling problem was discretized using the `ALBERTA`-1.2 fem-toolbox [11]. We applied linear Lagrange elements and used a projection method for the

[1] Available from `http://www.win.tue.nl/niconet/NIC2/slicot.html`.

curved boundaries. An initial mesh (see Fig. 1) was produced using MATLAB `pdetool` function, which implements a Delaunay triangulation algorithm. Finer discretizations were then obtained by global mesh refinement using a bisection refinement model. As a result, we obtained four different mesh resolutions and associated systems of order $n = 1{,}357$, 5,177, 20,209, and 79,841, corresponding to maximum mesh widths (or edge sizes) $h = 5.5280 \times 10^{-2}$, 2.7640×10^{-2}, 1.3820×10^{-2}, and 6.9100×10^{-3}, respectively. The number of nonzero elements in matrices A and E range from about 4% for the smallest model to 0.009% for the largest one. A more detailed description of the model is given in [5].

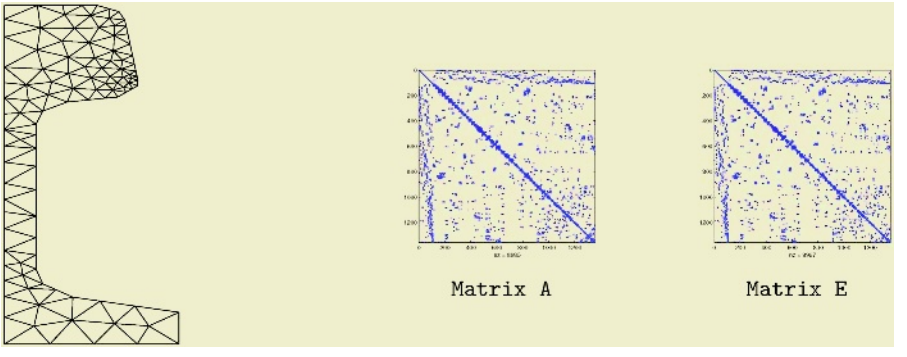

Fig. 1. Finite element discretization of the profile (left) and sparsity pattern (right) for the model of order n=1,357.

In our case, the best approximation error of the finite element discretization that one can expect is of order $\mathcal{O}(h^s)$ for an $s \in [1, 2)$. Here s is very close to 2, for the boundary is almost of class $\mathcal{C}^\infty$, i.e., if the two rightmost edges were smoothly topped off we would have sufficient regularity to obtain $s = 2$ at best. Thus, reducing the model with an error bound smaller than h^2 should not contribute any significant additional error. The reduced-order models presented in Section 4 meet this requirement.

2 Model Reduction of Large Linear Systems

2.1 The Square-Root BT Method

BT model reduction [7, 10, 12, 13] belongs to the family of absolute error methods, which aim at minimizing $\|\Delta_a\|_\infty = \|G - \hat{G}\|_\infty$. Here $\|G\|_\infty$ denotes the $\mathcal{L}_\infty$- or $\mathcal{H}_\infty$-norm of a stable, rational matrix function defined as

$$\|G\|_\infty = \sup_{\omega \in \mathbb{R}} \sigma_{\max}(G(\jmath\omega)), \tag{3}$$

where $\jmath := \sqrt{-1}$ and $\sigma_{\max}(M)$ stands for the largest singular value of M.

BT methods are strongly related to the controllability Gramian W_c and the observability Gramian W_o of the system (1), given by the solutions of the two dual generalized Lyapunov matrix equations

$$AW_cE^T + EW_cA^T + BB^T = 0, \tag{4}$$
$$A^T\tilde{W}_oE + E^T\tilde{W}_oA + C^TC = 0. \tag{5}$$

In the optimal cooling problem, the matrix pair (A, E) is stable (all its generalized eigenvalues are in the open left complex plane), so that W_c and $W_o = E^T\tilde{W}_oE$ are positive semidefinite, and therefore can be factored as $W_c = S^TS$ and $W_o = R^TR$. Here, S and R are usually refered to as the *Cholesky factors* of W_c and W_o.

Consider now the singular value decomposition (SVD) of the product

$$SR^T = U\Sigma V^T = [\,U_1\ U_2\,]\begin{bmatrix}\Sigma_1 & \\ & \Sigma_2\end{bmatrix}[\,V_1\ V_2\,]^T, \tag{6}$$

where U and V are orthogonal matrices, and $\Sigma = \mathrm{diag}\,(\sigma_1, \sigma_2, \ldots, \sigma_n)$ is a diagonal matrix containing the singular values of SR^T; those are also known as the *Hankel singular values* (HSV) of the system.

Given a partitioning of Σ into $\Sigma_1 \in \mathbb{R}^{r\times r}$ and $\Sigma_2 \in \mathbb{R}^{(n-r)\times(n-r)}$, and a conformal partitioning of U and V in (6), the *square-root* (SR) version of BT determines a reduced-order model of order r as

$$\hat{E} = T_lET_r, \quad \hat{A} = T_lAT_r, \quad \hat{B} = T_lB, \quad \hat{C} = CT_r, \quad \hat{D} = D, \tag{7}$$

with the projection matrices T_l and T_r given by

$$T_l = \Sigma_1^{-1/2}V_1^TRE^{-1} \quad \text{and} \quad T_r = S^TU_1\Sigma_1^{-1/2}. \tag{8}$$

The state-space dimension r of the reduced-order model can be chosen adaptively as this method provides a realization $\hat{G}$ satisfying

$$\|\Delta_a\|_\infty = \|G - \hat{G}\|_\infty \leq 2\sum_{j=r+1}^{n}\sigma_j. \tag{9}$$

In the following subsections we revisit two generalized Lyapunov solvers introduced in [2, 6, 8] which provide low-rank approximations to a Cholesky or full-rank factor of the solution matrix. These approximations can reliably substitute S and R in the computation of (6) and (8). For simplicity, we only describe those solvers that obtain approximations of the Cholesky factor of W_c in (4). Analogous iterations provide approximations for the Cholesky factor of W_o in (5).

2.2 Solution of Generalized Lyapunov Equations via the Matrix Sign Function

Since its introduction in [9], the sign function has proved useful in a variety of numerical linear algebra problems. In particular, the following variant of the

Newton iteration for the matrix sign function can be used for the solution of the generalized Lyapunov equation (4):

$$
\begin{aligned}
&A_0 \leftarrow A\\
&\tilde{S}_0 \leftarrow B\\
&k \leftarrow 0\\
&\texttt{repeat}\\
&\quad A_{k+1} \leftarrow \tfrac{1}{\sqrt{2}}\left(A_k + EA_k^{-1}E\right)\\
&\quad \text{Compute the rank-revealing QR (RRQR) decomposition}\\
&\qquad \tfrac{1}{\sqrt{2}}\left[\tilde{S}_k \;\; EA_k^{-1}\tilde{S}_k\right]^T = Q_s\begin{bmatrix} R_s \\ 0 \end{bmatrix}\Pi_s\\
&\quad \tilde{S}_{k+1} \leftarrow R_s\Pi_s\\
&\quad k \leftarrow k+1\\
&\texttt{until } \|A_k - E\|_1 < \tau\|A_k\|_1
\end{aligned}
$$

Here τ is a tolerance threshold for the iteration that is usually set as a function of the problem dimension and the machine precision. Convergence can be improved by using several acceleration techniques. In our case, we employ an approximation of the norm scaling [4]. The RRQR decomposition can be obtained by means of the traditional QR factorization with column pivoting.

Sign function iterations usually present a fast convergence rate, which is ultimately quadratic. However, the inverse of a sparse matrix is in general dense and therefore the previous iterative scheme cannot exploit the sparsity of the matrix A. Note that, on the other hand, we can easily take advantage of the sparsity of E as this matrix is involved in matrix products and it is not modified during the iteration.

On convergence, after j iterations, $\tilde{S} = \frac{1}{\sqrt{2}}\tilde{S}_j E^{-T}$, of dimension $\tilde{l} \times n$, is a full (row-)rank approximation of S^T so that $W_c = S^T S \approx \tilde{S}\tilde{S}^T$.

2.3 Low Rank Solution of Generalized Lyapunov Equations

The cyclic *low-rank alternating direction implicit* (LR-ADI) iteration proposed in [8] can be reformulated for the generalized Lyapunov equation (4) as follows:

$$
\begin{aligned}
&V_0 \leftarrow (A + p_1E)^{-1}B\\
&\hat{S}_0 \leftarrow \sqrt{-2\;\alpha_1}\;V_0\\
&k \leftarrow 0\\
&\texttt{repeat}\\
&\quad V_{k+1} \leftarrow V_k - \delta_k(A + p_{k+1}E)^{-1}EV_k\\
&\quad \hat{S}_{k+1} \leftarrow \left[\hat{S}_k \;,\; \gamma_k V_{k+1}\right]\\
&\quad k \leftarrow k+1\\
&\texttt{until } \|\gamma_k V_k\|_1 < \tau\|\hat{S}_k\|_1
\end{aligned}
$$

In the iteration, $\{p_1, p_2, \ldots\}$, $p_k = \alpha_k + \beta_k\, \jmath$, is a cyclic set of (complex) shift parameters (that is, $p_k = p_{k+t}$ for a given period t), $\gamma_k = \sqrt{\alpha_{k+1}/\alpha_k}$, and $\delta_k = p_{k+1} + \overline{p_k}$, $\overline{p_k}$ being the conjugate of p_k. This iteration may suffer from a slow

convergence rate, which is super-linear at best. Nevertheless, the iteration only requires the solution of linear systems with sparse coefficient matrices and matrix products. The use of sparse direct solvers is recommended here as iterations k and $k+t$ share the same coefficient matrices for the linear systems.

The performance of the LR-ADI iteration strongly depends on the selection of the shift parameters. For further details, see [6, 8, 15].

On convergence, after j iterations, a low-rank matrix $\hat{S} = \hat{S}_j$ of order $n \times \hat{l} = n \times (jm)$, is computed such that $\hat{S}\hat{S}^T$ approximates $W_c = S^T S$.

It should be emphasized that the iterative methods described in the previous two subsections for solving (4)–(5) significantly differ from standard methods used in the MATLAB toolboxes or SLICOT [14]. As the iterative solvers produce low-rank approximations to the full-rank or Cholesky factors, the computation of the SVD in (6) is usually much more efficient: instead of a computational cost of $\mathcal{O}(n^3)$ flops (floating-point arithmetic operations) when using the Cholesky factors, this approach leads to an $\mathcal{O}(m \cdot p \cdot n)$ cost where, in model reduction, often $m, p \ll n$; see [3].

3 Parallel Implementation

The matrix sign function-based iteration basically requires dense linear algebra operations such as matrix products and the solution of linear systems (via matrix factorizations). On the other hand, the LR-ADI iteration is composed of sparse linear algebra operations as matrix-vector products and the solution of sparse linear systems (via direct methods). Once the generalized Lyapunov equations are solved, the final stages of the SR BT method require the computation of an SVD and a few matrix products.

Our approach for dealing with these matrix operations is based on the use of existing parallel linear algebra and communication libraries. (For an extensive list, see `http://www.netlib.org/utk/people/JackDongarra/la-sw.html`.) In Fig. 2 we display the multilayered architecture of libraries employed by our parallel model reduction codes for large-scale dense and sparse systems in PLiCMR and SpaRed, respectively. All model reduction codes employ the parallel dense linear algebra libraries ScaLAPACK and PBLAS. Depending on the structure of the state-space matrix pair (A, E) the kernels in SpaRed also use the banded linear system solvers in ScaLAPACK or the sparse linear system solvers in the packages MUMPS or SuperLU. PARPACK is our key to compute eigenvalue information of large sparse matrix pairs.

4 Experimental Results

All the experiments presented in this section were performed on a cluster of $n_p = 16$ nodes using IEEE double-precision floating-point arithmetic ($\varepsilon \approx 2.2204 \times 10^{-16}$). Each node consists of an Intel Xeon processor@2.4 GHz with 1 GByte of RAM. We employ a BLAS library specially tuned for this processor that

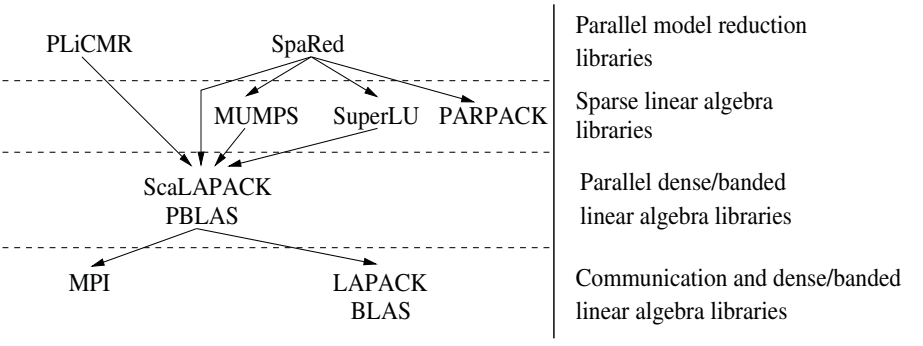

Fig. 2. Multilayered architecture of libraries for model reduction.

achieves around 3800 Mflops (millions of flops per second) for the matrix product (routine `DGEMM` from `http://www.cs.utexas.edu/users/kgoto`).The nodes are connected via a *Myrinet* multistage network and the MPI communication library is specially developed and tuned for this network. The performance of the interconnection network was measured by a simple loop-back message transfer resulting in a latency of 18 μsec. and a bandwidth of 1.4 Gbit/sec.

In this section, we compare the BT parallel routines in libraries PLiCMR and SpaRed (hereafter, PLiCMR-BT and SpaRed-BT). Given the amount of computational resources available and the memory requirements of the sign function-based iteration, we could only apply the PLiCMR-BT routine to the two smaller cases, $n = 1,357$ and $5,177$.

In order to reduce the models, we first compute the HSVs of the system and, from there, we select the order r of the reduced realization. Obviously, a larger order provides a more accurate model, but also increases the cost of those stages involving the reduced system. Table 1 reports the order r of the reduced realization, a bound for the absolute error of the reduced-order realization (computed as in (9)), and the first and $(r+1)$-th HSV of the system. All these data were obtained by applying the SpaRed-BT routine to the system. No difference was found when the PLiCMR-BT routine was applied to compute these parameters for the two smaller problems. Our results hereafter refer to the reduced realizations of order $r = 45$, 45, 70, and 80 for the $n = 1,357$, 5,177, 20,209, and 79,841 cases, respectively.

In order to measure the numerical accuracy of the reduced realizations, in our next experiment we compare the frequency response of the original system, G, with that of the reduced-order realization computed with the PLiCMR-BT and SpaRed-BT routines, $\hat{G}$. Figure 3 reports the absolute error $\|G - \hat{G}\|_\infty$, where the norm is defined as in (3). All four figures show that the absolute error is well below the theoretical bound. For the two smallest problems, there is no notable difference between the PLiCMR-BT and Spared-BT routines.

We next report in Tables 2 and 3 the specific results for each one of the model reduction methods. In particular, we report the number of iterations required

Table 1. Order and absolute error of the reduced realizations.

n	r	Δ_a	σ_1	σ_{r+1}
1,357	45	6.5e-7	3.5e-4	5.2e-7
5,177	45	1.3e-6	3.5e-4	9.0e-8
20,209	45	1.4e-4	2.5e-2	7.0e-6
20,209	60	2.7e-5	2.5e-2	1.6e-6
20,209	70	8.6e-6	2.5e-2	4.7e-7
79,841	45	2.2e-4	2.6e-2	1.1e-5
79,841	60	4.6e-5	2.6e-2	2.4e-6
79,841	80	5.4e-6	2.6e-2	3.1e-7

Table 2. Results for the PLiCMR-BT routine.

n	#iter.	l	k	$\mathcal{R}_{W_c}(S)$	$\mathcal{R}_{W_o}(R)$	n_p	Time (sec.)
1,357	8	310	181	1.04e-22	7.20e-14	4	21.1
5,177	8	351	209	1.48e-21	9.95e-14	16	142.6

for convergence (#iter.), the dimensions of the low-rank approximations of S and R (labeled as l and k, respectively), and the absolute residuals of these approximations, computed as

$$\mathcal{R}_{W_c}(S) := \|A(S^TS)E^T + E(S^TS)A^T + BB^T\|_F, \quad \text{and}$$
$$\mathcal{R}_{W_o}(R) := \|A^T(E^{-T}R^TRE^{-1})E + E^T(E^{-T}R^TRE^{-1})A + C^TC\|_F.$$

Finally, we also provide the number of nodes involved in the reduction (n_p), and the execution time required by the corresponding algorithm. A comparison of the results in both tables show a much slower convergence rate for the SpaRed-BT algorithm which then provides approximations to the Cholesky factors of much larger order than the PLiCMR-BT algorithm. On the other hand, the use of an approximation of larger order also explains, in part, the slightly better absolute residuals for the SpaRed-BT algorithm. However, the most important difference between the two methods lies in the execution times. For the two smallest problems, using a smaller number of computational resources (processors), the SpaRed-BT algorithm provides the reduced realization faster than the PLiCMR-BT algorithm. Furthermore, only the kernel from the SpaRed library can reduce the largest two cases, while doing so with the PLiCMR kernel would require a number of resources much larger than available in our cluster.

5 Concluding Remarks

We have compared two parallel BT model reduction algorithms using a problem arising from a semi-discretization of a controlled heat transfer process for optimal cooling of a steel profile. The algorithms are included in the parallel libraries for model reduction PLiCMR and SpaRed and basically differ in the generalized Lyapunov equation solver that is employed in each case: the PLiCMR-BT

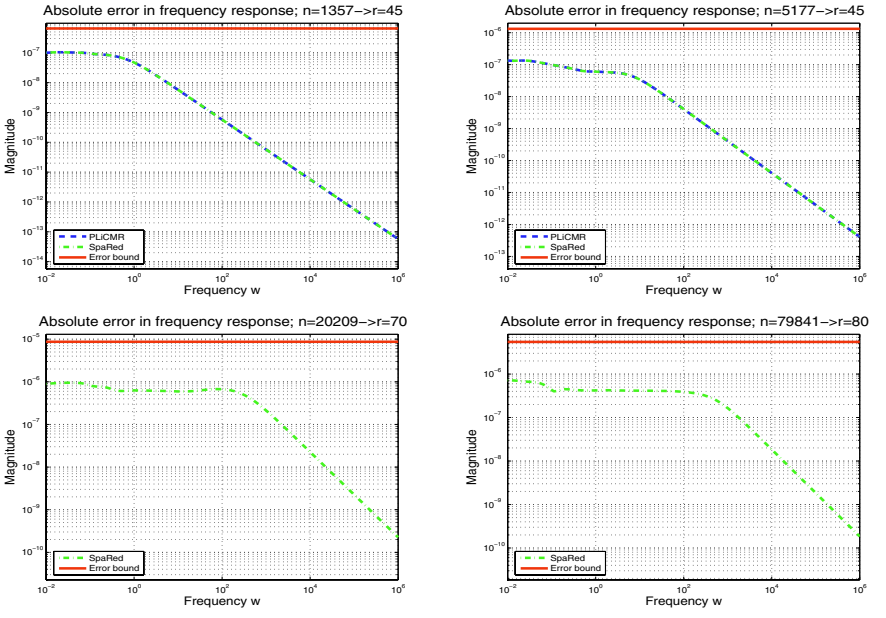

Fig. 3. Absolute error in the frequency response for the reduced-order realizations.

Table 3. Results for the SpaRed-BT routine.

n	#iter.	l	k	$\mathcal{R}_{W_c}(\hat{S})$	$\mathcal{R}_{W_o}(\hat{R})$	n_p	Time (sec.)
1,357	82	574	492	4.9e-24	1.0e-14	1	14.2
5,177	98	686	588	8.5e-23	1.5e-14	1	35.4
20,209	76	532	456	1.5e-14	2.0e-13	4	151.4
79,841	78	546	468	3.9e-13	9.6e-14	16	484.7

method is based on the sign function iteration while the SpaRed BT method employs a generalization of the LR-ADI iteration.

The experimental results show that, for the optimal cooling of steel profiles problem, the SpaRed-BT algorithm is clearly the best option. By exploiting the sparsity of the problem this algorithm requires less computational resources, provides the answer faster, and allows the reduction of problems which could not be dealt with using the PLiCMR-BT algorithm. On the other hand, for the small-dimension problems, the models computed by the PLiCMR-BT algorithm are presumably more accurate and can serve as a reference for those computed with SpaRed.

References

1. R.M. Badía, P. Benner, R. Mayo, and E.S. Quintana-Ortí. Parallel algorithms for balanced truncation model reduction of sparse systems. In *Proc. of PARA'04 – Workshop on State-of-the-Art in Scientific Computing*, Lecture Notes in Computer Science. Springer-Verlag, Berlin, Heidelberg, New York, to appear.

2. P. Benner, E. S. Quintana-Ortí, and G. Quintana-Ortí. Parallel model reduction of large-scale descriptor linear systems via balanced truncation. In *Proceedings of the 6th International Meeting on High Performance Computing for Computational Science. VECPAR'04*, number 3402 in Lecture Notes in Computer Science, pages 340–353. Springer-Verlag, Berlin, Heidelberg, New York, 2005.
3. P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. State-space truncation methods for parallel model reduction of large-scale systems. *Parallel Comput.*, 29:1701–1722, 2003.
4. P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. Solving linear matrix equations via rational iterative schemes. *Journal of Scientific Computing*, to appear.
5. P. Benner and J. Saak. A semi-discretized heat transfer model for optimal cooling of steel profiles. In P. Benner, V. Mehrmann, and D. Sorensen, editors, *Dimension Reduction of Large-Scale Systems*, volume 45 of Lecture Notes in Computational Science and Engineering, pages 353–356. Springer-Verlag, Berlin/Heidelberg, Germany, 2005.
6. J.-R. Li and J. White. Low rank solution of Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 24(1):260–280, 2002.
7. B.C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. Automat. Control*, AC-26:17–32, 1981.
8. T. Penzl. A cyclic low rank Smith method for large sparse Lyapunov equations. *SIAM J. Sci. Comput.*, 21(4):1401–1418, 2000.
9. J.D. Roberts. Linear model reduction and solution of the algebraic Riccati equation by use of the sign function. *Internat. J. Control*, 32:677–687, 1980.
10. M.G. Safonov and R.Y. Chiang. A Schur method for balanced-truncation model reduction. *IEEE Trans. Automat. Control*, AC–34:729–733, 1989.
11. A. Schmidt and K. Siebert. *Design of Adaptive Finite Element Software; The Finite Element Toolbox ALBERTA*, volume 42 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin/Heidelberg, 2005.
12. M.S. Tombs and I. Postlethwaite. Truncated balanced realization of a stable non-minimal state-space system. *Internat. J. Control*, 46(4):1319–1330, 1987.
13. A. Varga. Efficient minimal realization procedure based on balancing. In *Prepr. of the IMACS Symp. on Modelling and Control of Technological Systems*, volume 2, pages 42–47, 1991.
14. A. Varga. Model reduction software in the SLICOT library. In B.N. Datta, editor, *Applied and Computational Control, Signals, and Circuits*, volume 629 of *The Kluwer International Series in Engineering and Computer Science*, pages 239–282. Kluwer Academic Publishers, Boston, MA, 2001.
15. E.L. Wachspress. The ADI model problem, 1995. Available from the author.