Experimental Evaluation of the Parallel Model Reduction Routines in PSLICOT[1]

Peter Benner[2], Enrique S. Quintana-Ortí[3], and Gregorio Quintana-Ortí[4]

August 2002

[1]This document presents research results of the European Community BRITE-EURAM III Thematic Networks Programme NICONET (contract number BRRT-CT97-5040) and is distributed by the Working Group on Software WGS. *WGS secretariat*: Mrs. Ida Tassens, ESAT - Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, 3001-Leuven-Heverlee, BELGIUM. This report is available by anonymous ftp from `wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/SLWN2002-7.ps.Z`
[2]Institut für Mathematik, Technische Universität Berlin, D-10623 Berlin, Germany.
[3]Depto. de Ingeniería y Ciencia de Computadores, Universidad Jaume I, 12080–Castellón, Spain.
[4]Same address as second author.

Abstract

We report an experimental evaluation, including numerical aspects and parallel performance, of the parallel routines for absolute error model reduction in PSLICOT based on iterative solution of the underlying matrix (Lyapunov) equations. The frequency response and the performance of the parallel routines are compared with those of the analogous codes in SLICOT.

Contents

1 Introduction

Consider the transfer function matrix (TFM) $G(s) = C(sI - A)^{-1}B + D$, and the associated stable, but not necessarily minimal, realization of a linear time-invariant (LTI) system,

Continuous LTI system:

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

Discrete LTI system:

$$\begin{array}{rcll} x_{k+1} & = & Ax_k + Bu_k, & x_0 = x^0, \\ y_k & = & Cx_k + Du_k, & k = 0, 1, 2, \ldots, \end{array} \tag{2}$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, and $D \in \mathbb{R}^{p \times m}$; the number of states, n, is also said to be the order of the system. In model reduction, the goal is to find a reduced-order LTI system

Continuous LTI system:

$$\begin{array}{rcll} \dot{\hat{x}}(t) & = & \hat{A}\hat{x}(t) + \hat{B}\hat{u}(t), & t > 0 \quad \hat{x}(0) = \hat{x}^0, \\ \hat{y}(t) & = & \hat{C}\hat{x}(t) + \hat{D}\hat{u}(t), & t \geq 0, \end{array} \tag{3}$$

Discrete LTI system:

$$\begin{array}{rcll} \hat{x}_{k+1} & = & \hat{A}\hat{x}_k + \hat{B}\hat{u}_k, & \hat{x}_0 = \hat{x}^0, \\ \hat{y}_k & = & \hat{C}\hat{x}_k + \hat{D}\hat{u}_k, & k = 0, 1, 2, \ldots, \end{array} \tag{4}$$

of order r, $r \ll n$, and associated TFM $\hat{G}(s) = \hat{C}(sI - \hat{A})^{-1}\hat{B} + \hat{D}$ which approximates $G(s)$.

Model reduction of large-scale systems arises, among others, in control of large flexible structures or large power systems, manipulation of fluid flow, circuit simulation, VLSI design, etc. [9, 10, 11, 13, 20]. Reliable methods for model reduction usually require $\mathcal{O}(n^3)$ flops and storage for $\mathcal{O}(n^2)$ numbers. While systems of order n in the hundreds may be treated on current desktop computers (using the SLICOT[1] model reduction software, for instance), large-scale applications clearly require the use of parallel computing techniques to obtain the reduced-order system. Model reduction was therefore identified in the NICONET[2] project as one of the tasks were parallel computing could provide the necessary computational power and, as a response, a parallel library for model reduction (PLiCMR[3]) was developed which was used as a basis for PSLICOT [7].

In this paper, we report experimental results using the parallel kernels in PLiCMR to obtain reduced-order models for a number of LTI systems. Both the numerical aspects of the reduced-order models and the parallel performance of the algorithms are evaluated. In particular, the accuracy of the computed reduced-order model is analyzed by comparing the frequency response of the original system with those of the reduced-order models computed using the parallel routines and the equivalent serial routines from SLICOT. A benchmark with several examples recently collected in [9] is employed for this purpose. The efficiency of the parallel algorithms is evaluated via two large-scale problems.

The paper is structured as follows. In Section 2 we briefly revisit the methods offered by the parallel library. In Section 3 we report numerical and parallel experiments using the kernels in PLiCMR. Finally, some concluding remarks follow in Section 4.

[1] Available from `http://www.win.tue.nl/niconet/NIC2/slicot.html`.
[2] Visit `http://www.win.tue.nl/niconet`.
[3] See `http://spine.act.uji.es/p̃licmr`.

2 PLiCMR: A Parallel Library for Model Reduction

Our parallel model reduction library provides absolute error methods, based on truncated state-space transformations, which try to minimize $\|\Delta_a\|_\infty = \|G - \hat{G}\|_\infty$. Here, $\|G\|_\infty$ denotes the $\mathcal{L}_\infty$- or $\mathcal{H}_\infty$-norm of a stable, rational matrix function defined as

$$\|G\|_\infty = \operatorname*{ess\,sup}_{\omega \in \mathbb{R}} \sigma_{\max}(G(\jmath\omega)), \tag{5}$$

where $\jmath := \sqrt{-1}$ and $\sigma_{\max}(M)$ is the largest singular value of the matrix M. Specifically, three different methods are offered as part of the library: Balanced truncation (BT) methods [19, 22, 23, 25], singular perturbation approximation (SPA) methods [17], and Hankel-norm approximation (HNA) methods [15]. All three methods are strongly related to the controllability Gramian W_c and the observability Gramian W_o of the systems (1) and (2), given by the solutions of two "coupled" linear matrix equations:

Continuous LTI system: Lyapunov equations

$$\begin{aligned} AW_c + W_cA^T + BB^T &= 0, & (6)\\ A^TW_o + W_oA + C^TC &= 0, & (7) \end{aligned}$$

Discrete LTI system: Stein equations

$$\begin{aligned} AW_cA^T - W_c + BB^T &= 0, & (8)\\ A^TW_oA - W_o + C^TC &= 0. & (9) \end{aligned}$$

As A is assumed to be stable, W_c and W_o are positive semidefinite and therefore can be factored as $W_c = S^TS$ and $W_o = R^TR$. The factors S and R are called the *Cholesky factors* of the Gramians. The absolute error model reduction methods employed here compute a singular value decomposition (SVD) of the product SR^T for determining the reduced-order model.

Consider now the SVD

$$SR^T = U\Sigma V^T,$$

where $\Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ is a diagonal matrix containing the singular values of SR^T, and U, V are orthogonal matrices. Here, σ_1 is also known as the Hankel norm of the system. The difference (absolute error) between between the TFM of the original system and that of a reduced-order system of order r (computed with an absolute error method) is theoretically bounded by

$$\|G(s) - G_r(s)\|_\infty \le 2 \sum_{i=r+1}^{n} \sigma_i\,. \tag{10}$$

Our methods basically differ from those in SLICOT in two aspects:

- The parallel routines in PLiCMR exploit the fact that the Cholesky factors of the Gramians are often of low numerical rank. Thus, instead of computing the Cholesky factors by Hammarling's method [16, 24] (as done in the serial routines in SLICOT), our parallel routines compute full column rank factors of the Gramians using the sign function method. In [4, 5] this technique is reported to achieve considerable savings in computational and storage costs, both in theory and in practice.
- A procedure with enhanced accuracy, introduced in [12], is then employed to compute the SVD of the product of the factors of the Gramians.

The parallel routines in PLiCMR are named using the parallel standard in SLICOT [1]. In particular, three routines are provided in PLiCMR:

– `pab09ax`: BT method (analogous to SLICOT routine `ab09ad`).

– `pab09bx`: SPA method (analogous to SLICOT routine `ab09bd`).

– `pab09cx`: HNA method (analogous to SLICOT routine `ab09cd`).

Square-root (SR) and balancing-free square-root (BFSR) implementations are provided for the BT and SPA methods. Moreover, the routines can be applied to both continuous and discrete-time LTI systems.

The parallelization heavily relies on the use of the parallel infrastructure in ScaLAPACK [8] (including PBLAS and BLACS), and the serial computational libraries LAPACK and BLAS (see Figure 1). Therefore, the library can be migrated to any parallel architecture where these computational libraries and a communication library like MPI or PVM are available. Details of the parallelization are given in [4, 5, 6, 7].

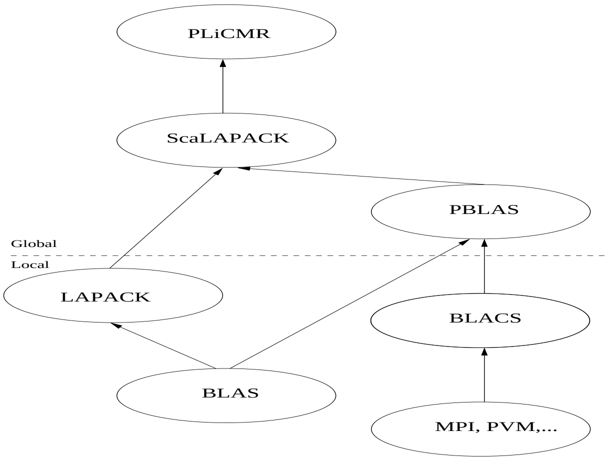

Figure 1: PLiCMR and the underlying libraries.

PLiCMR is available at `ftp://ftp.esat.kuleuven.ac.be/pub/WGS/SLICOT/plicmr.tar.gz`. The version maintained at `http://spine.act.uji.es/~plicmr` might, at some stages, contain later updates than the version integrated into PSLICOT. A Web service provided at this address allows to experiment with the parallel routines on a cluster composed of 32 nodes (owned by the parallel and scientific computing group of the Universidad Jaume I). More details on this Web service and a similar Mail service are given in [2, 3]; see also the URI given above.

3 Experimental Results

All the experiments presented in this section were performed on Intel Pentium-II processors using IEEE double-precision floating-point arithmetic ($\varepsilon \approx 2.2204 \times 10^{-16}$). The parallel algorithms were evaluated on a parallel distributed cluster of 32 nodes. Each node consists of an Intel Pentium-II processor at 300 MHz, and 128 MBytes of RAM. We employ a BLAS library, specially tuned for the Pentium-II processor as part of the ATLAS and ITXGEMM projects [26, 14], that achieves around 180 Mflops (millions of flops per second) for the matrix product (routine `DGEMM`). The nodes are connected via a *Myrinet* crossbar network; the communication library BLACS is based on an implementation of MPI 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 33 μsec and a bandwidth of 200 Mbit/sec. We made use of the LAPACK, PBLAS, and ScaLAPACK libraries wherever possible.

We first describe the numerical performance using a collection of benchmark examples for model reduction of linear time invariant dynamical systems [9]. The parallel efficiency is then evaluated by means of two large-scale problems.

3.1 Numerical performance

This subsection compares the frequency responses $|G_{jk}(\jmath\omega)|, 1 \leq j \leq p, 1 \leq k \leq m$, at frequencies ω, of the transfer function from input k to output j with those of the reduced-order systems computed with the SLICOT and PLiCMR routines.

Ten different examples are used for this evaluation, coming from very different applications. For a detailed description of the models see [9] and the references therein. We only employ a subset of the examples from [9] as neither PSLICOT nor SLICOT routines can treat the generalized state-space models given in the benchmark collection.

Table 1 shows the parameters of the systems used here. In order to fix the order of the reduced-order system automatically, the SLICOT and PLiCMR routines select r so that

$$\sigma_r > \max(\tau_1, n \cdot \varepsilon \cdot \sigma_1) > \sigma_{r+1},$$

where ε is the machine precision and τ_1 is a user-specified tolerance threshold. In our case, we set $\tau_1 = \eta \cdot \sigma_1$, where the value η is adjusted for each particular case as shown in the table. The SPA and HNA methods also employ a second tolerance threshold equal to $\max(\tau_2, n \cdot \varepsilon \cdot \sigma_1)$ in order to determine a minimal realization of the system. In our experiments we set $\tau_2 = 0$.

As we did not find any difference between the SR and BFSR approaches, in the experiments we only report the results of the latter.

Example	n	m	p	σ_1	η	r
Eady	598	1	1	$9.93 \times 10^{+2}$	1.0×10^{-3}	9
CDplayer	120	2	2	$1.17 \times 10^{+6}$	1.0×10^{-8}	42
FOM	1006	1	1	$5.00 \times 10^{+1}$	1.0×10^{-3}	10
Random	200	1	1	$8.19 \times 10^{+6}$	1.0×10^{-6}	7
PDE	84	1	1	$5.34 \times 10^{+0}$	1.0×10^{-3}	2
Heat (cont.)	200	1	1	3.25×10^{-2}	1.0×10^{-3}	4
Heat (disc.)	200	1	1	3.33×10^{-2}	1.0×10^{-7}	11
ISS	270	3	3	5.79×10^{-2}	1.0×10^{-3}	36
Build	48	1	1	2.50×10^{-3}	1.0×10^{-3}	30
Beam	348	1	1	$2.38 \times 10^{+3}$	1.0×10^{-3}	12

Table 1: Parameters of the examples employed in the numerical evaluation of the parallel model reduction routines.

3.1.1 Eady example

This is an example modeling the atmospheric storm track. The atmosphere in the region (e.g., in the midlatitude of the Pacific) is discretized using an horizontal division of 1000×1000 km^2 and a vertical division 10 km long. Time is discretized in periods of 9 hours. Figure 2 reports the frequency response of this example.

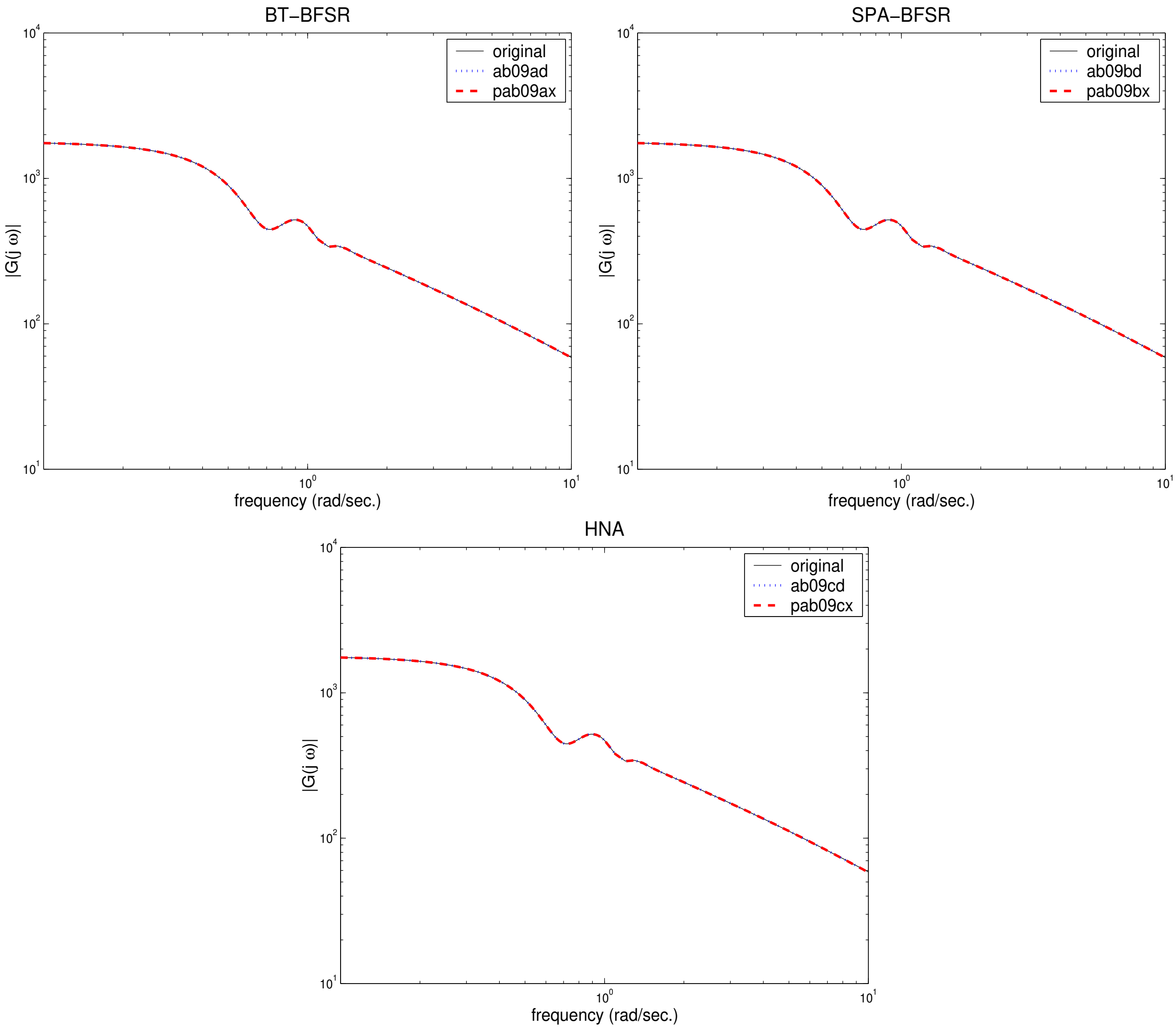

Figure 2: Frequency response of the eady example.

As to be expected from the smooth frequency response of the system, an almost perfect match of the system response is obtained by all reduced-order models computed by the SLICOT and PSLICOT routines.

3.1.2 CD player example

This example is a frequently used benchmark example for model reduction. The model describes the dynamic behaviour of a portable Compact Disc player. In particular, the system models the mechanism of the swing arm and the lens of the CD. The goal here is to obtain a low-cost controller which is both fast and robust to external shocks. The two main difficulties here are that the Hankel singular values of the system decay without significant gap and that the possible reduction is limited by the amount of states needed in the reduced-order model necessary for catching all the significant peeks in the frequency response, in particular for G_{12} and G_{21}.

Figures 3 and 4 report the dynamics of the rotating arm. The different figures show the frequency response of each input-output pair of this system.

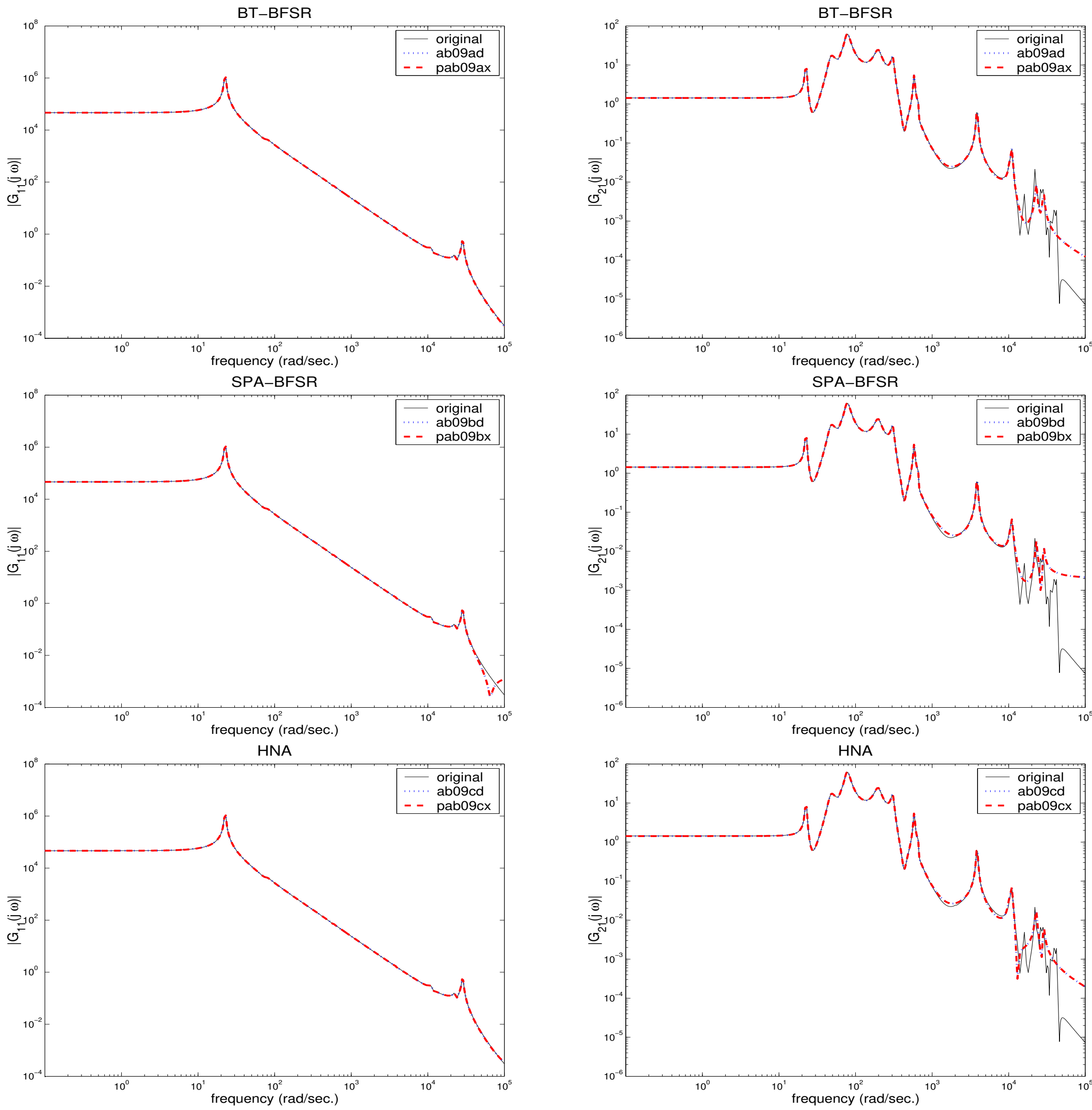

Figure 3: Frequency response of the CD player example: 1st input vs. 1st output (left) and 1st input vs. 2nd output (right).

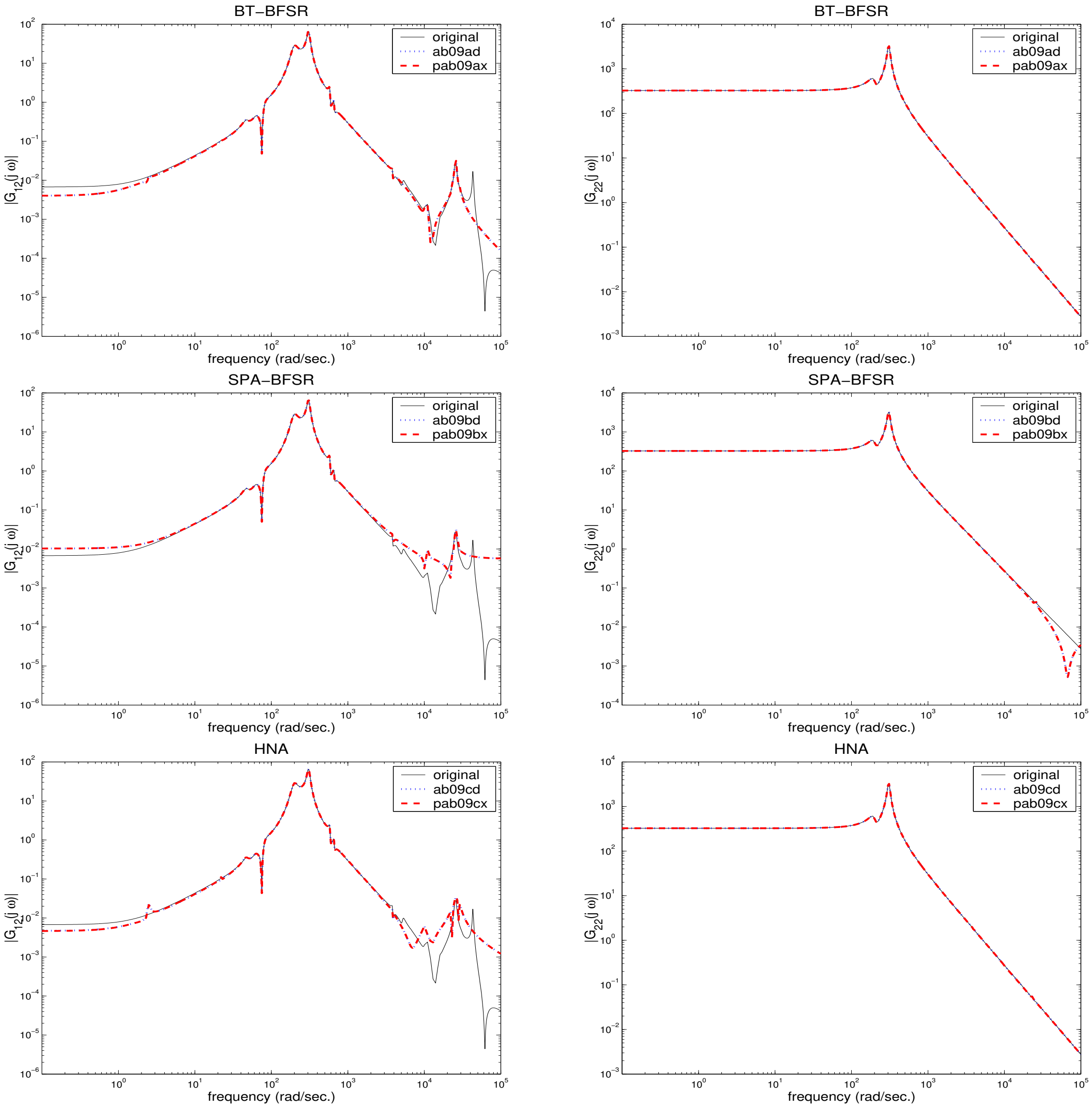

Figure 4: Frequency response of the CD player example: 2nd input vs. 1st output (left) and 2nd input vs. 2nd output (right).

There is no visible difference in the reduced-order models computed by SLICOT or PSLICOT, but the three different model reduction technologies (BT, SPA, HNA) show some (dis-)advantages when compared to each other. Here, the BT method appears to be slightly favorable when having a closer look at the high frequency range of G_{12} and G_{21}. But note that the differences mostly appear in a frequency range with small gain.

3.1.3 FOM example

This full-order model (FOM) of a large dynamical system has six badly damped poles, i.e., three pairs of complex conjugate poles with large imaginary part which are responsible for the three significant peeks in the frequency response. Except for these peeks, the dynamics of this system are smooth as all the remaining poles are real and well separated from the imaginary axis. There is no difficulty to be expected in reducing the order of this system significantly.

Figure 5 reports the frequency response of this example.

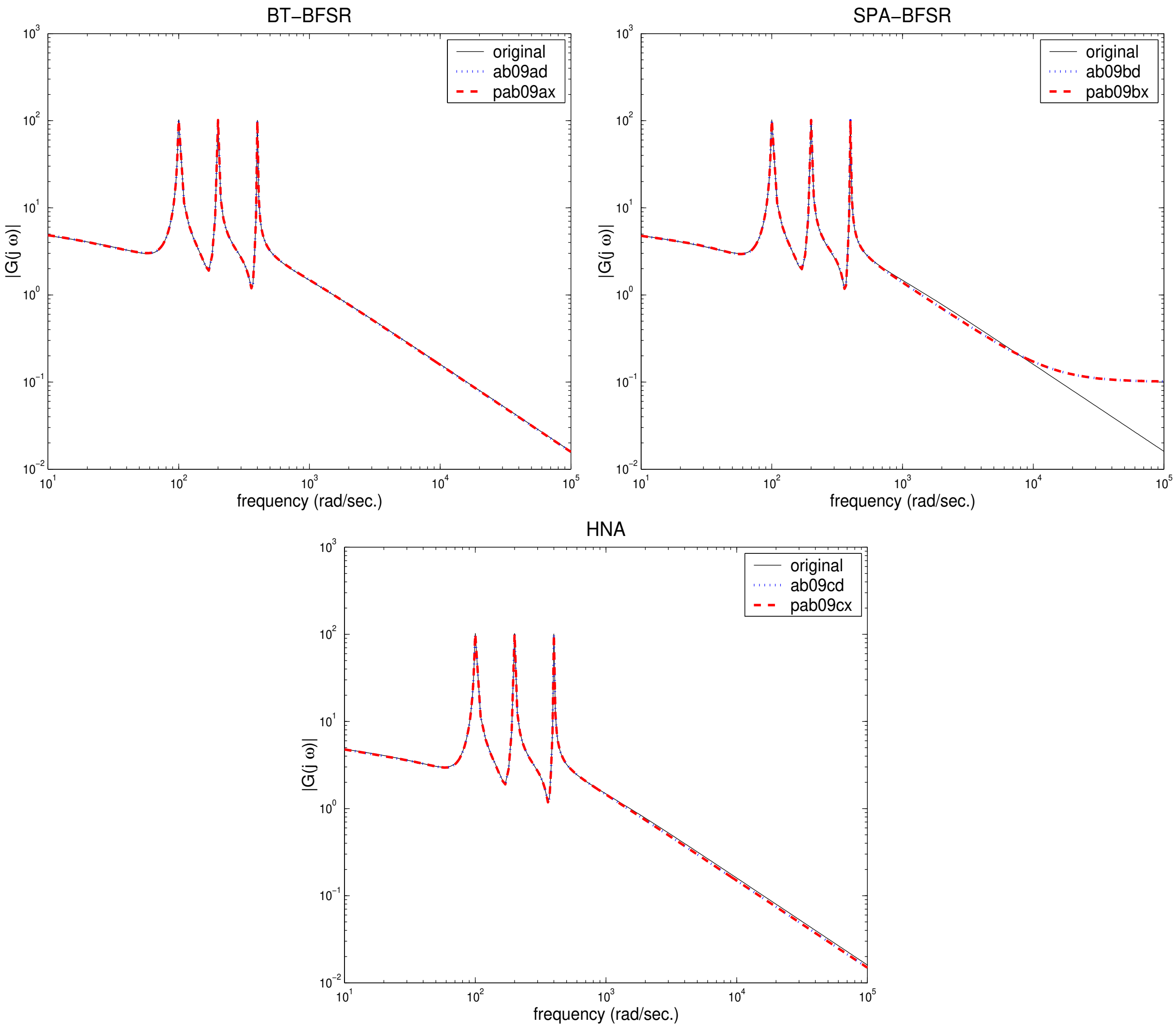

Figure 5: Frequency response of the FOM example.

According to the theoretical properties of singular perturbation approximation, the SPA codes produce a deviation of the reduced-order models at high frequencies for both SLICOT and PSLICOT codes.

3.1.4 Random example

This is a randomly generated single-input, single-output model. Figure 6 reports the frequency response in this case.

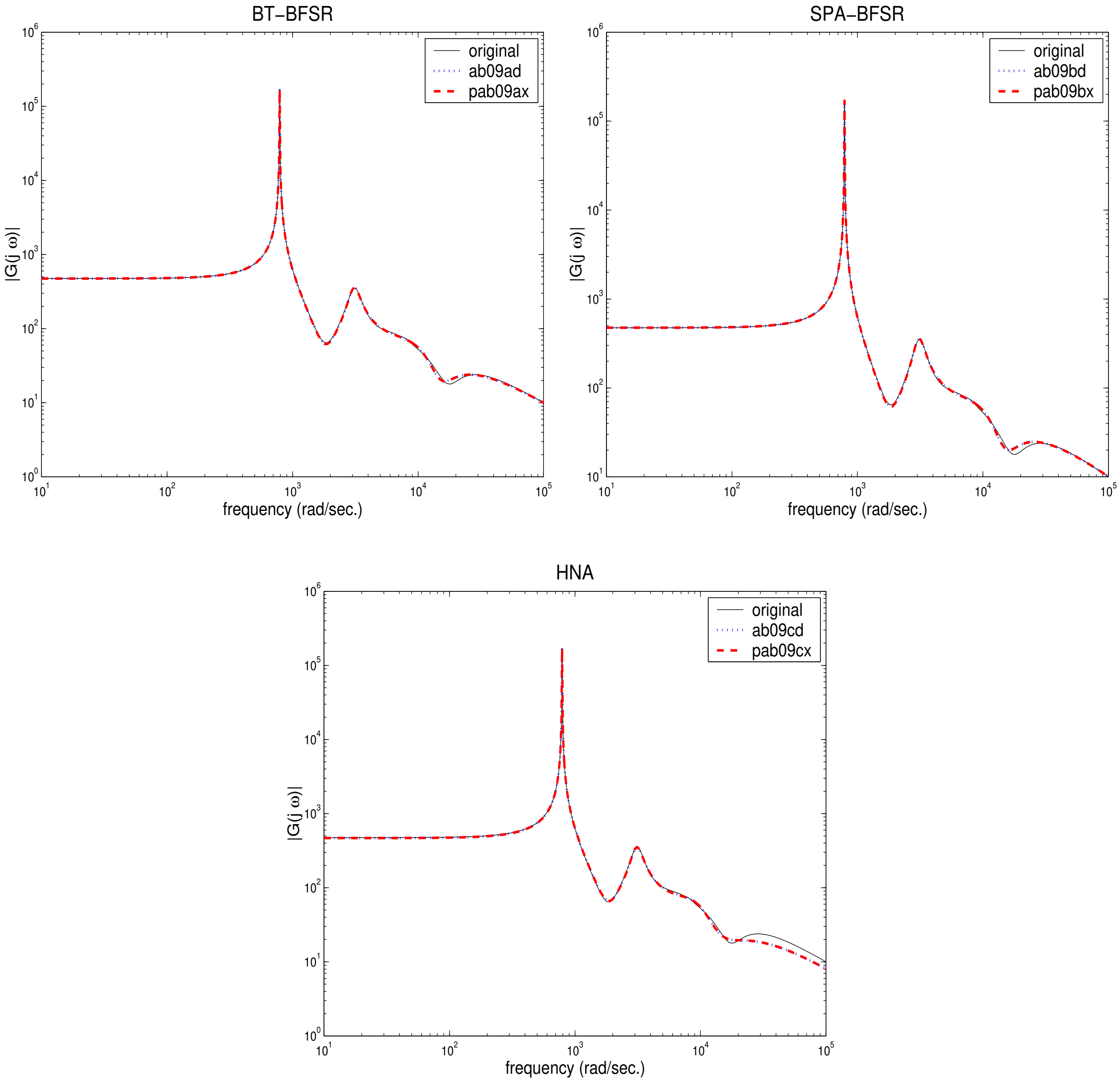

Figure 6: Frequency response of the random example.

Here, the HNA implementations show an unexpected loss of accuracy for high frequencies. Increasing the dimension of the reduced-order model somewhat gives a more accurate model that again shows no significant difference when compared to the original system.

3.1.5 PDE example

The space discretization of control problems for linear parabolic differential equations (PDEs) leads to continuous LTI systems as in (1). This example corresponds to a 2-dimensional convection-diffusion equation with point control semi-discretized by centered differences. Here, the diffusive part of the PDE is dominant, resulting in fast decay of the Hankel singular values and a very smooth frequency response of the corresponding dynamical system.

Figure 7 reports the frequency response of an example arising from a PDE.

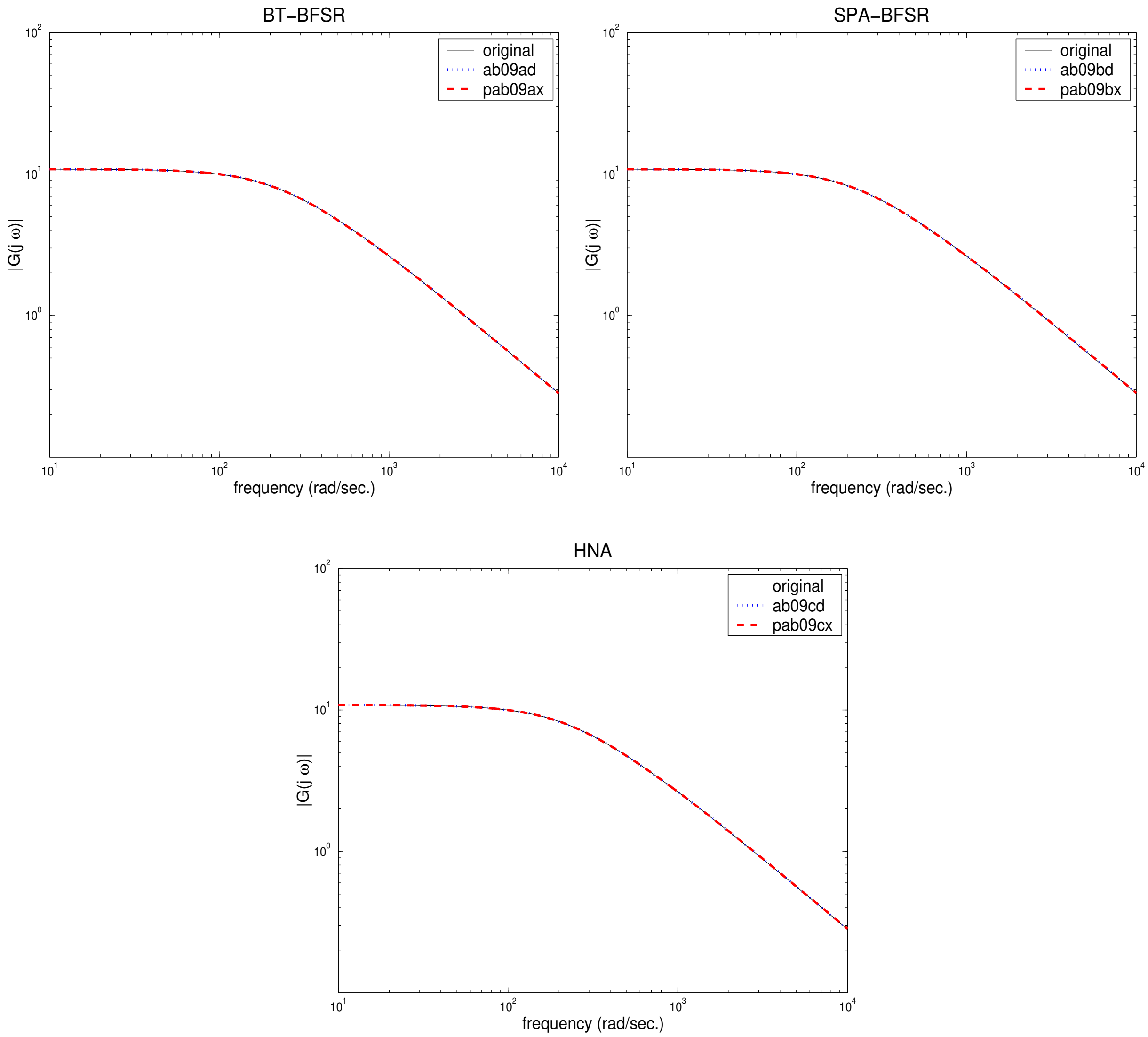

Figure 7: Frequency response of the PDE example.

None of the methods or codes has difficulties to obtain a perfect match with a reduced-order model of very small size (here, $r = 2$!).

3.1.6 Heat equation example: continuous case

The heat equation example is another example of a semidiscretized point control problem for a parabolic PDE. The given equation models the heat diffusion in a (1-dimensional) thin rod with a single heat source. The spatial domain is discretized in this case in segments of length $h = \frac{1}{N+1}$ and again centered differences are used to approximate the diffusion operator. Figure 8 reports the frequency response for this model.

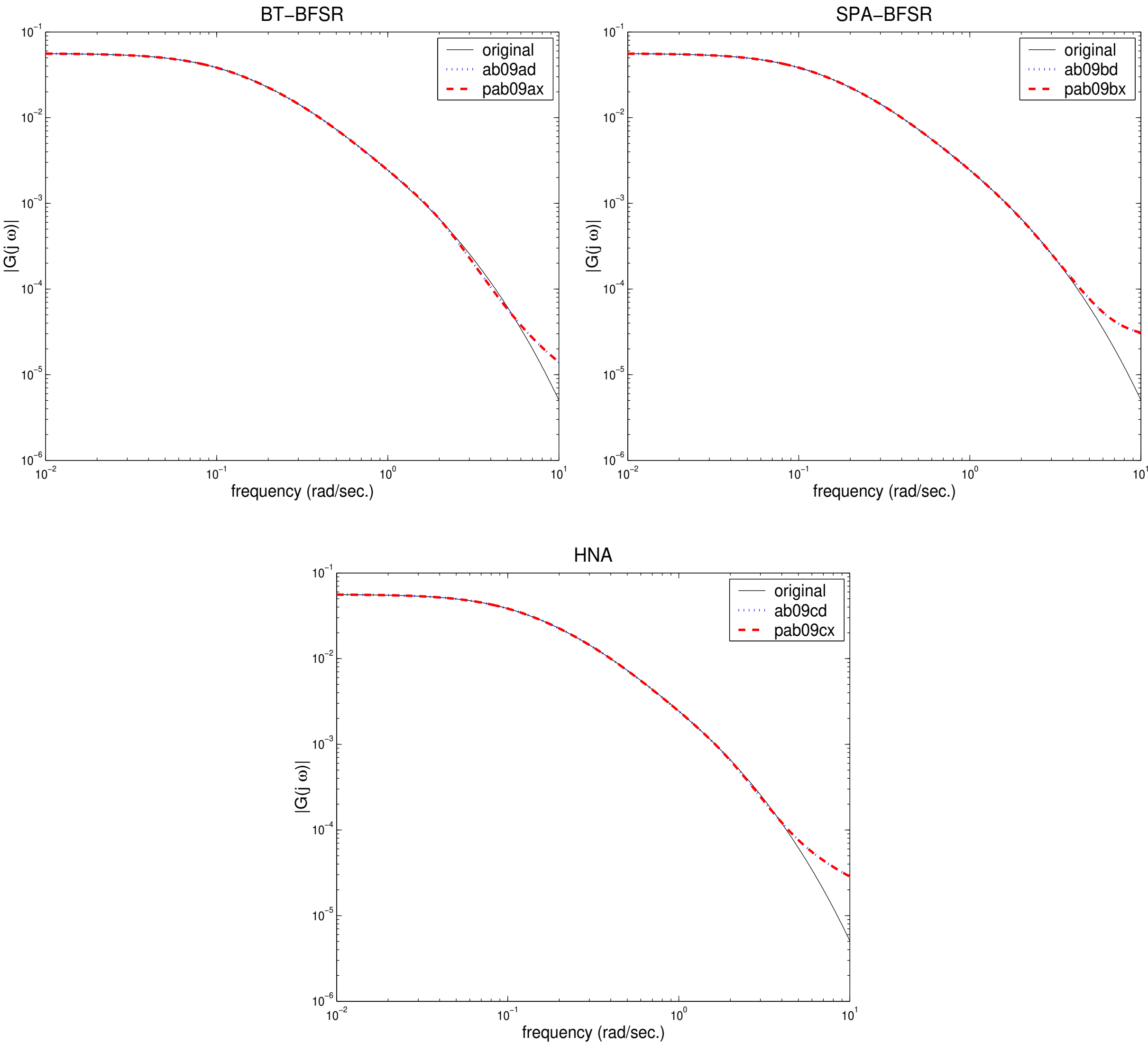

Figure 8: Frequency response of the heat equation (continuous case) example.

For all three methods and all codes a slight deviation in the frequency response of the reduced-order model for increasing frequencies is visible. Here, the BT approach is slightly worse than the SPA and HNA methods as a visible error is encountered at lower frequencies.

3.1.7 Heat equation example: discrete case

Here, the same equation as in the last section is used. This time, a full discretization of the control problem for the heat equation is obtained using using the Crank–Nicholson scheme. This results in a discrete LTI system as in (2). In this case, the Hankel singular value decay slightly slower than in the continuous case, leading to a higher dimension of the reduced-order system if the same approximation error is to be achieved. Figure 9 reports the frequency response for the discrete heat equation.

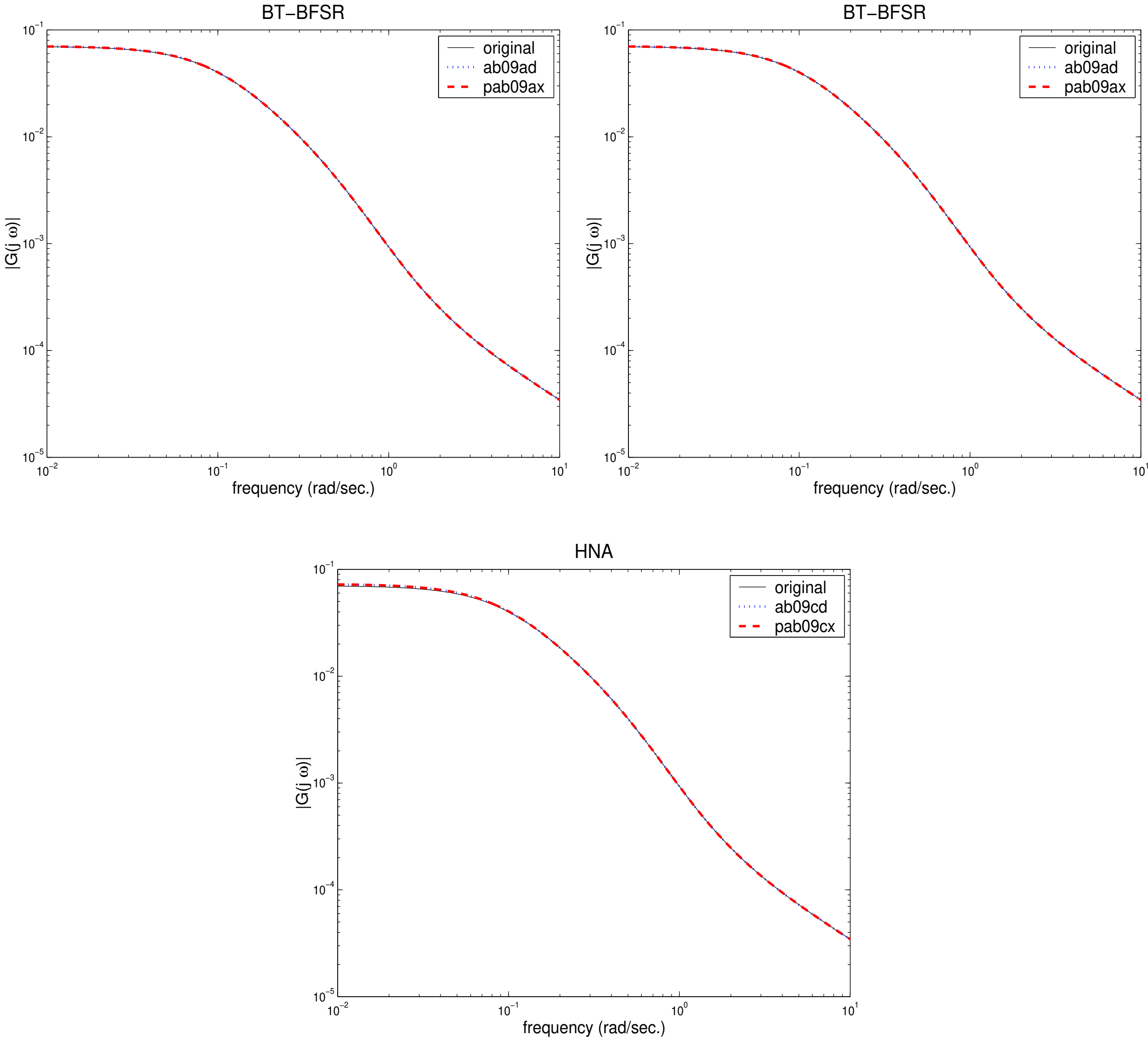

Figure 9: Frequency response of the heat equation (discrete case) example.

Again, none of the methods or codes has difficulties to compute a reduced-order model that accurately approximates the original model.

A comparison of the execution times of the SLICOT/PSLICOT codes for this example is given in Section 3.2, showing a significant gain in efficiency of the computational approach used in the parallel codes.

3.1.8 ISS example

This is a (linearized second-order) system modeling component 1R (the Russian service model) of the International Space Station (ISS). The challenge here is to control a flexible structure in space in real-time so that it becomes necessary to reduce the flex models in order to complete the analysis in a timely manner. Figures 10–14 report the frequency responses associated with each input and output pair for this component.

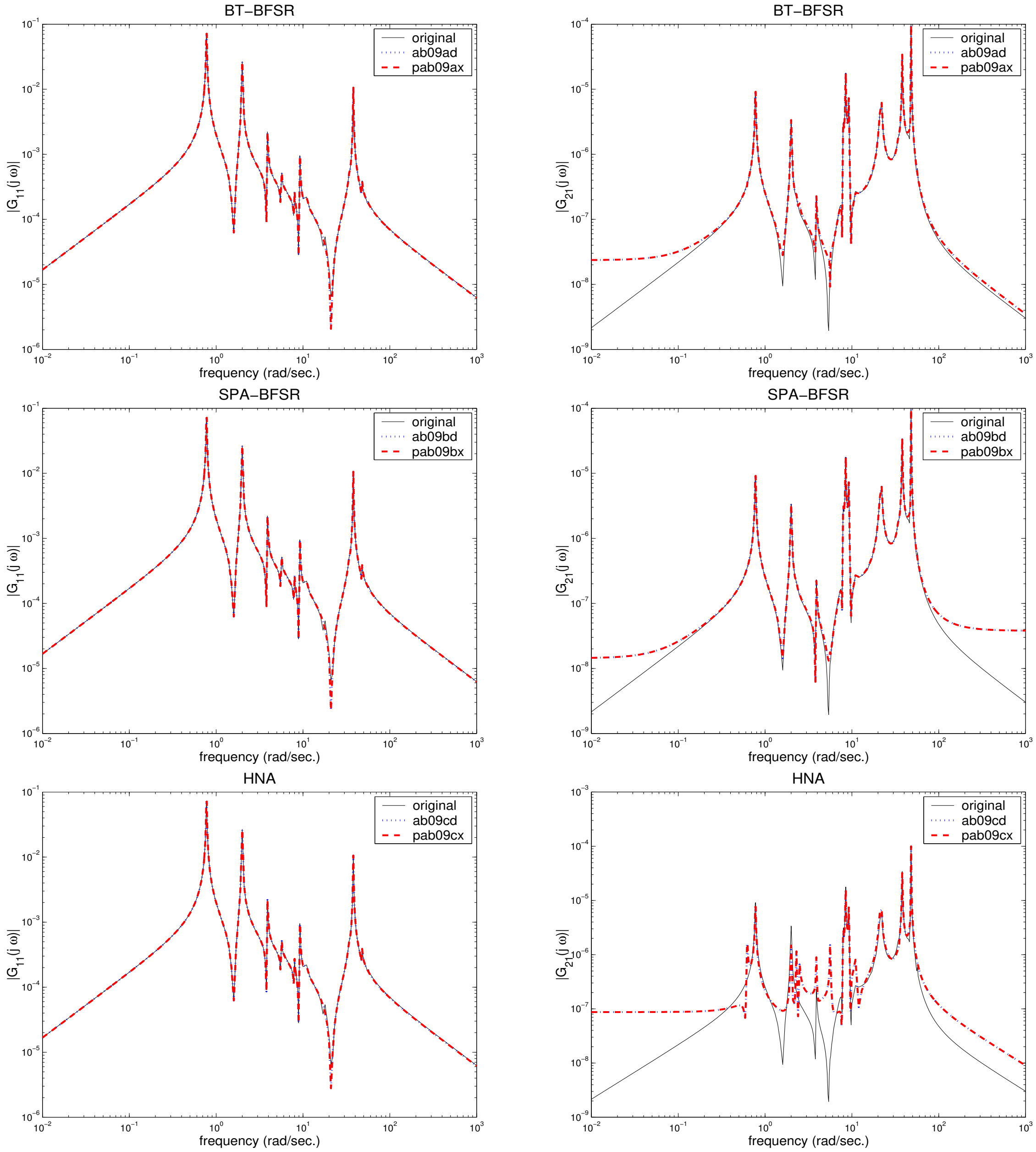

Figure 10: Frequency response of the ISS example: 1st input vs. 1st output (left) and 1st input vs. 2nd output (right).

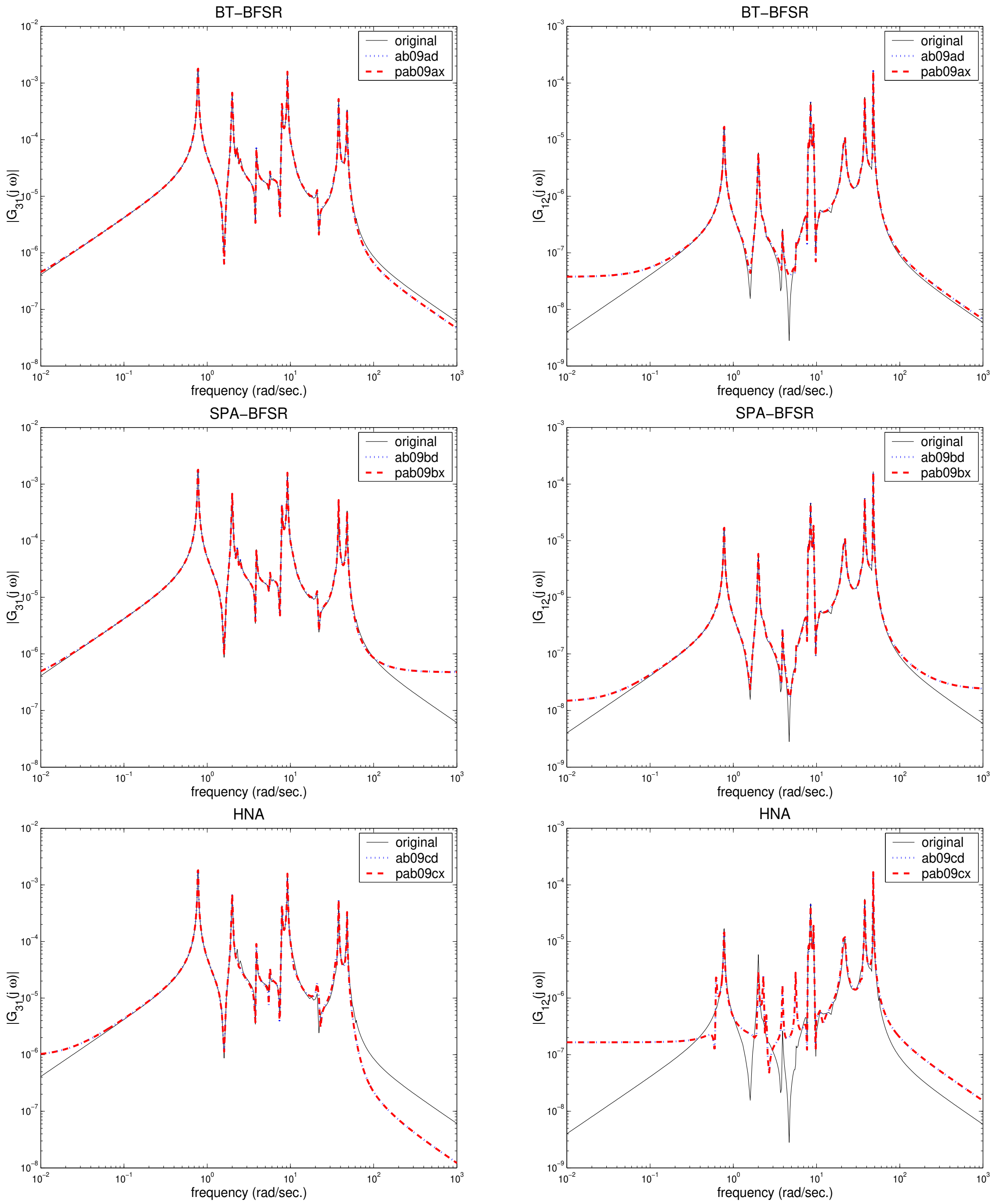

Figure 11: Frequency response of the ISS example: 1st input vs. 3rd output (left) and 2nd input vs. 1st output (right).

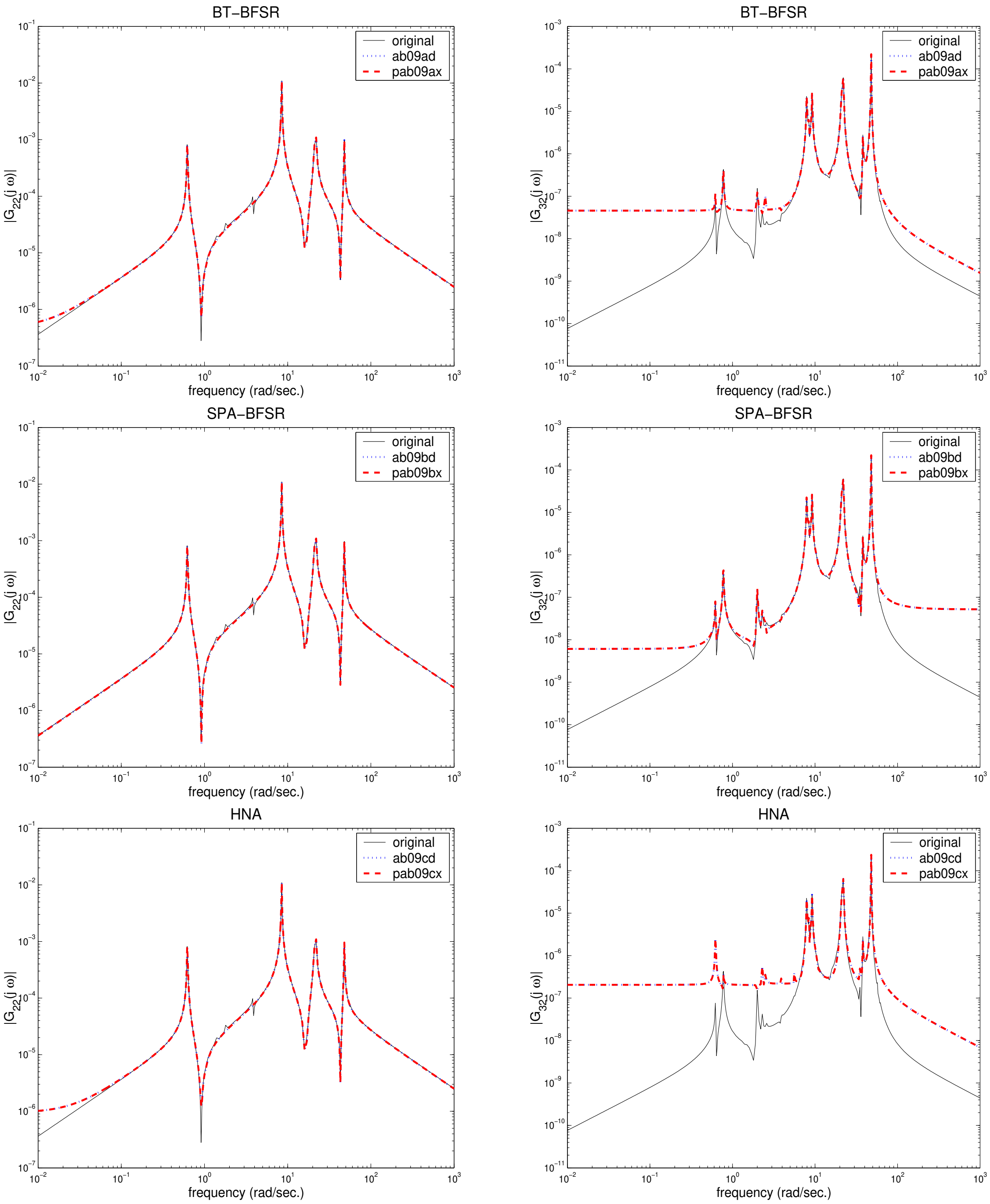

Figure 12: Frequency response of the ISS example: 2nd input vs. 2nd output (left) and 2nd input vs. 3rd output (right).

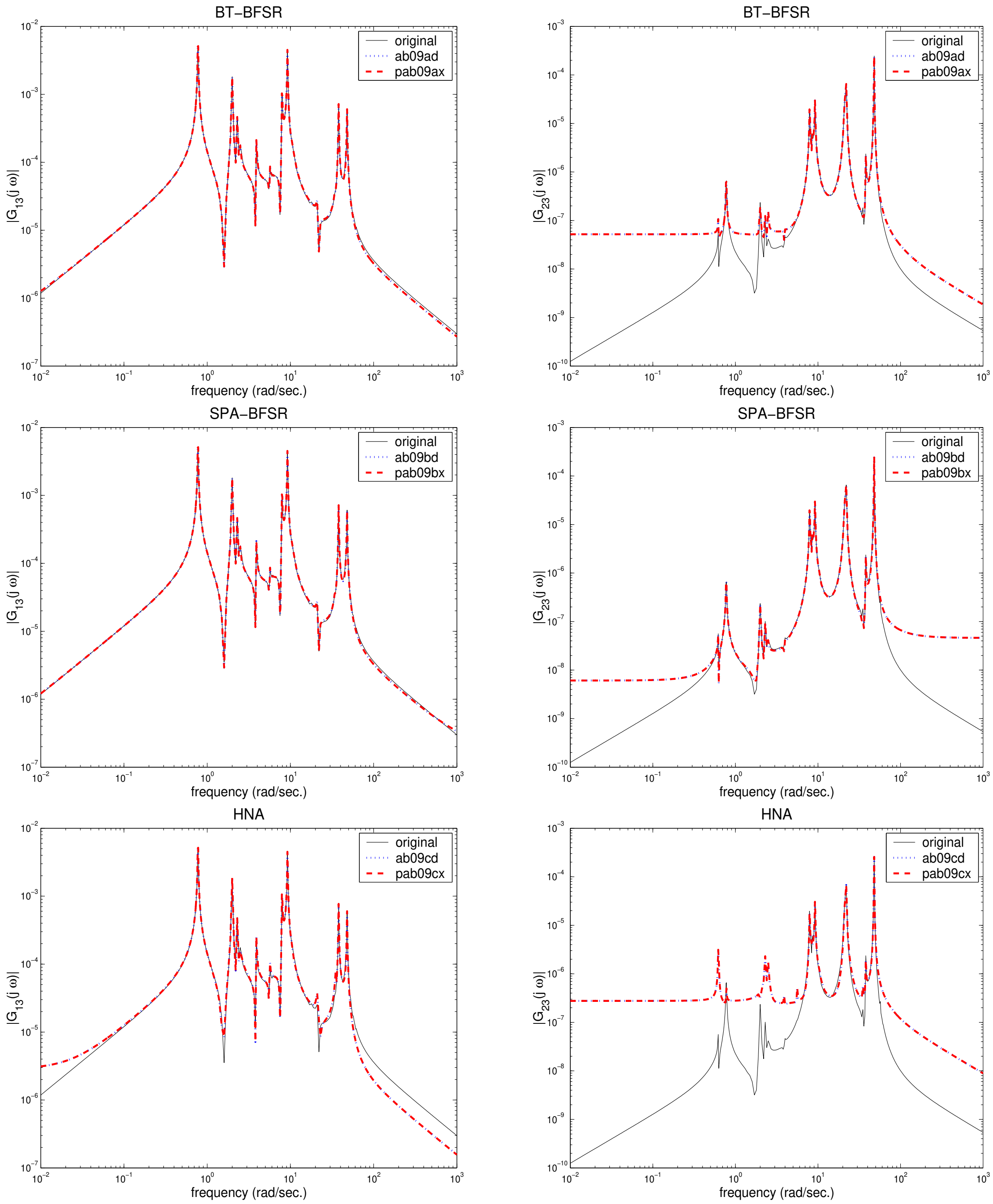

Figure 13: Frequency response of the ISS example: 3rd input vs. 1st output (left) and 3rd input vs. 2nd output (right).

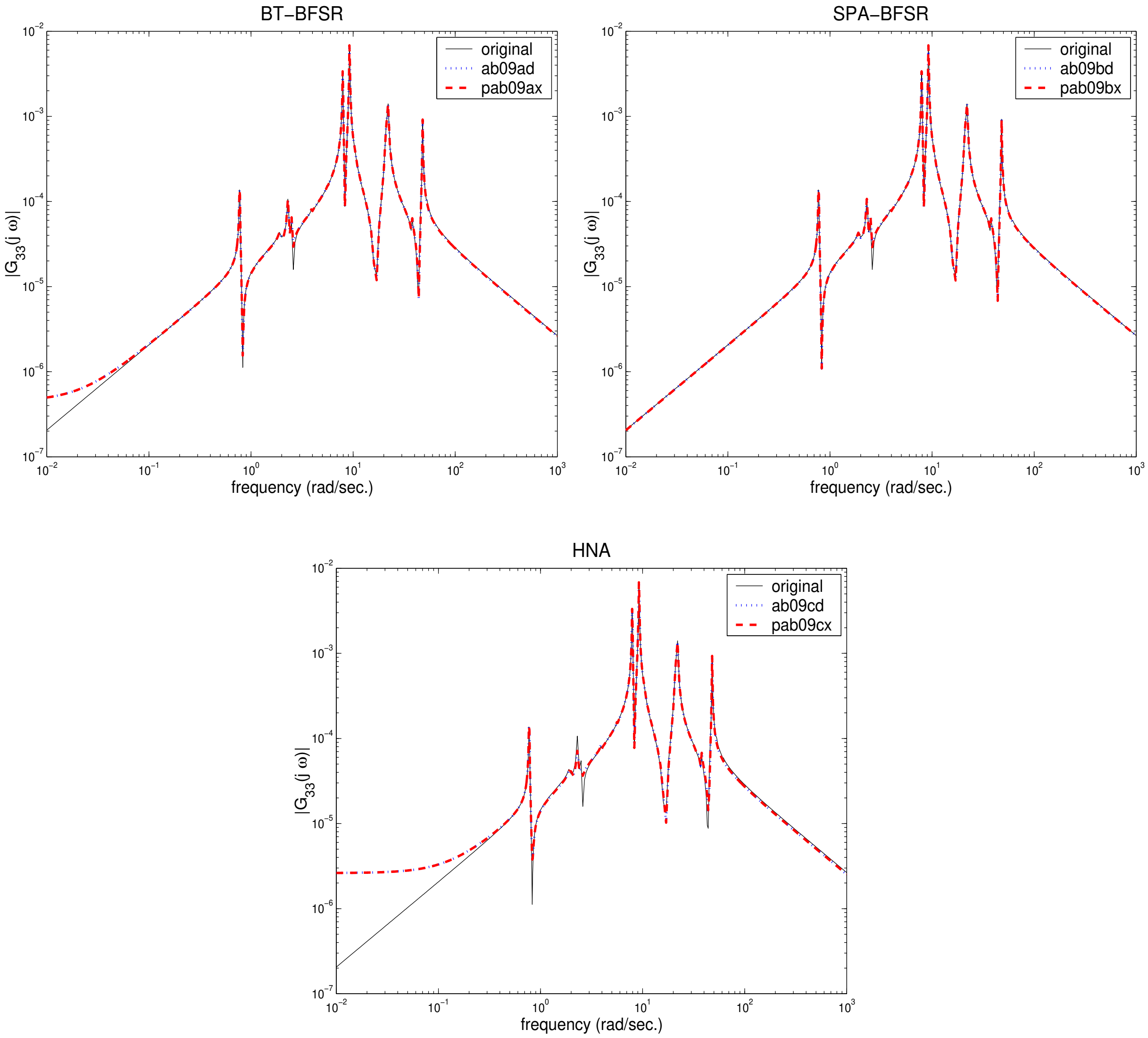

Figure 14: Frequency response of the ISS example: 3rd input vs. 3rd output.

The large visible deviation in some of the plots, in particular for the HNA reduced-order models and in two cases also for the BT models, is due to the nature of absolute error methods: as the gain in the corresponding input-output channels is much smaller than, e.g., in the $(1,1)$-component and hence much smaller than the H_∞-norm of the system, a better approximation cannot be expected as the absolute error is small enough, even though the plots give a different impression. Here, a relative error analysis would uncover the bad approximation of the HNA method (or the BT method in the $(3,2)$ component) in these cases.

3.1.9 Building example

This example comes from modeling vibrations of the *Los Angeles University Hospital*. The building was discretized with 8 floors each having 3 degrees of freedom which correspond to the displacements in the x and y directions, and the rotation. A second-order differential equation is thus obtained which is then transformed into an standard continuous-time LTI system. Figure 15 reports the frequency response of the model.

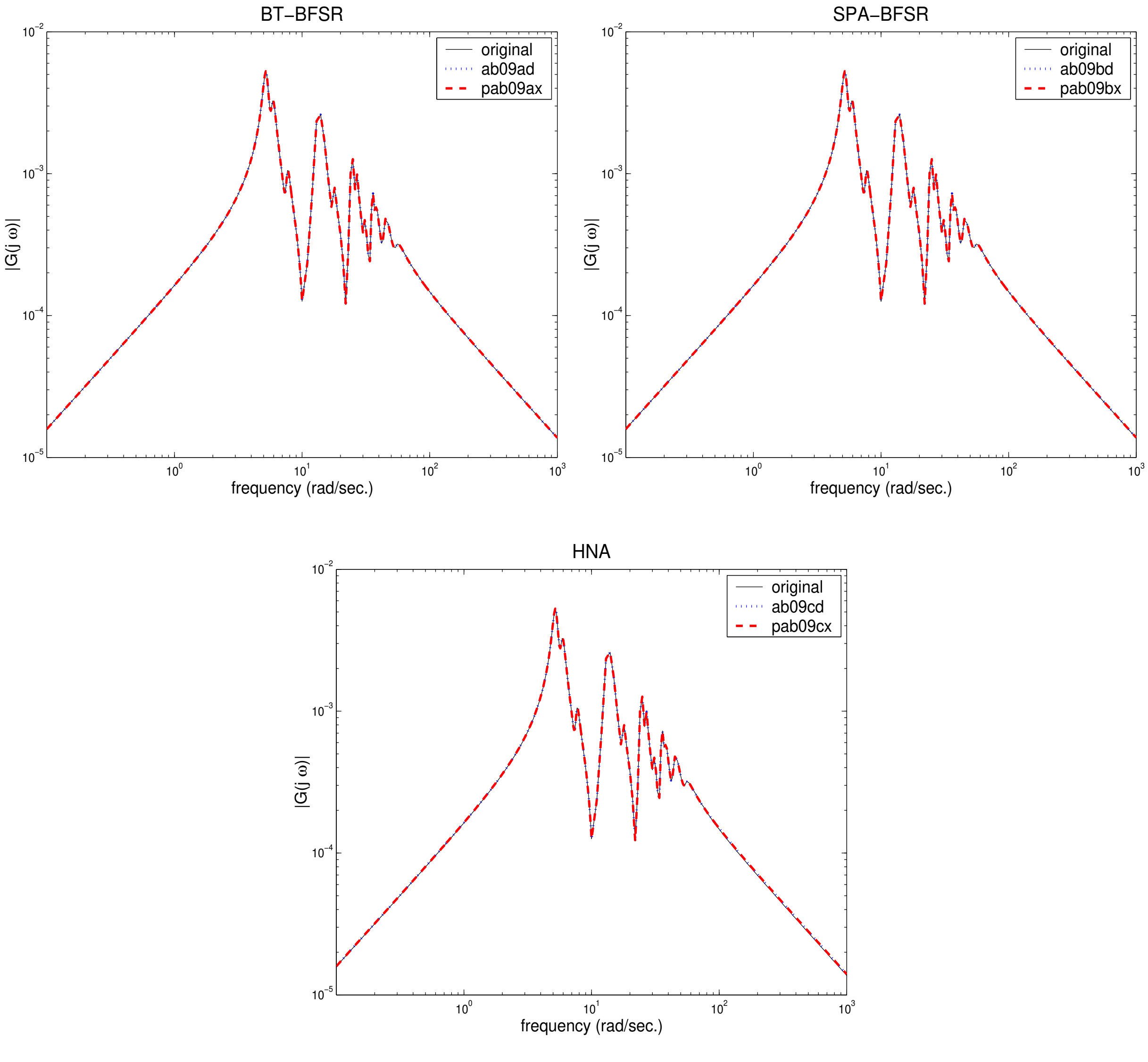

Figure 15: Frequency response of the building example.

A very good approximation is obtained by all the computed reduced-order models, but the achieved reduction is not quite satisfactory (from 48 to 30 states). The problem here is that all poles have imaginary parts, resulting in a difficult-to-approximate frequency response of the original system (just counting peeks shows that the reduced-order model needs at least 20 states to match all of them).

3.1.10 Beam example

This system results from the discretization of a hyperbolic PDE modeling a clamped beam with proportional damping, resulting in a second-order system. Here the input contains the force applied to the structure while the output represents the displacement produced by this input. Figure 16 reports the frequency response of this model.

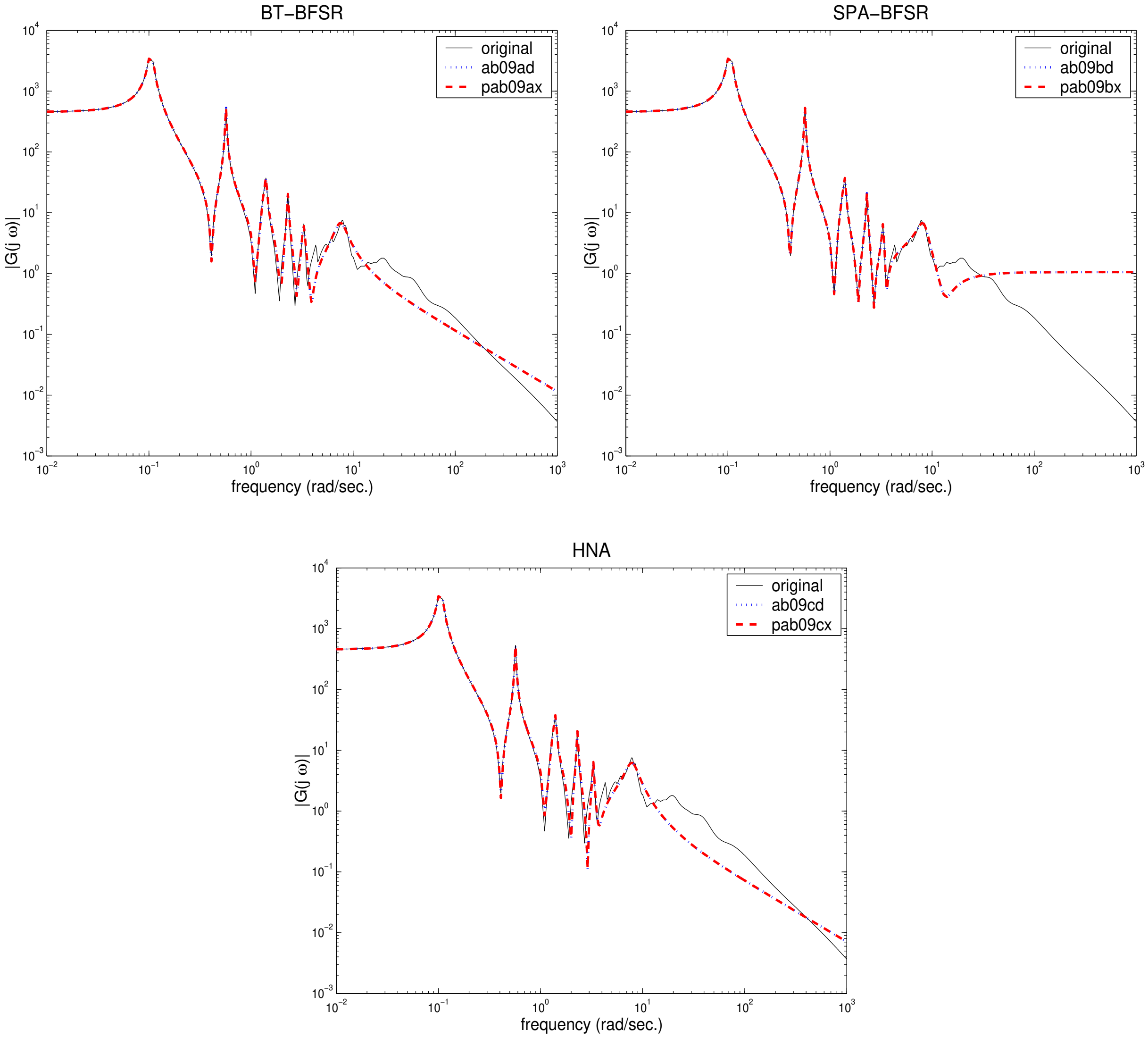

Figure 16: Frequency response of the beam example.

Though most of the poles of the system are slightly undamped, very good approximation at the low frequencies (which are relevant to vibration analysis) is achieved with a very low order model by all three methods. Again, no difference in the frequency responses of the models computed by the SLICOT and PSLICOT routines is visible.

3.1.11 Approximation Errors

Table 2 shows the absolute error, $\|\Delta_a\|_\infty = \|G - \hat{G}\|_\infty$, of the reduced-order systems computed with the serial and the parallel routines and the theoretical bound (10) for the different examples. We used the MATLAB function `normhinf` taken from the Robust Control Toolbox [18] in order to compute the H_∞ norms of the errors, except for the CD player example where we had to use a grid search as `normhinf` failed to return correct results in this example. The error bound was obtained from the Hankel singular values provided in the data files corresponding to [9].

Example	bound (10)	`ab09ad`	`pab09ax`	`ab09bd`	`pab09bx`	`ab09cx`	`pab09cx`
Eady	$1.1 \times 10^{+0}$	4.6×10^{-1}	4.6×10^{-1}	4.1×10^{-1}	4.1×10^{-1}	4.9×10^{-1}	4.9×10^{-1}
CDplayer	2.4×10^{-1}	2.0×10^{-2}	2.0×10^{-2}	2.2×10^{-2}	2.2×10^{-2}	3.8×10^{-2}	5.9×10^{-2}
FOM	1.0×10^{-1}	1.0×10^{-1}	1.0×10^{-1}	1.0×10^{-1}	1.0×10^{-1}	7.1×10^{-2}	7.1×10^{-2}
random	$1.8 \times 10^{+1}$	$3.0 \times 10^{+1}$	$3.1 \times 10^{+1}$	$3.1 \times 10^{+1}$	$2.7 \times 10^{+1}$	$2.5 \times 10^{+1}$	$4.1 \times 10^{+1}$
PDE	1.0×10^{-2}	4.4×10^{-3}	4.4×10^{-3}	7.4×10^{-3}	6.8×10^{-3}	7.4×10^{-3}	7.4×10^{-3}
Heat (cont.)	3.4×10^{-5}	2.6×10^{-5}	2.6×10^{-5}	2.8×10^{-5}	2.8×10^{-5}	2.9×10^{-5}	2.9×10^{-5}
Heat (disc.)	2.7×10^{-9}	2.1×10^{-8}	2.1×10^{-8}	2.1×10^{-8}	2.2×10^{-8}	1.6×10^{-7}	1.5×10^{-7}
ISS	1.8×10^{-3}	1.1×10^{-4}	1.1×10^{-4}	1.1×10^{-4}	1.1×10^{-4}	1.1×10^{-4}	1.1×10^{-4}
Build	2.7×10^{-5}	4.9×10^{-6}	4.9×10^{-6}	4.8×10^{-6}	4.8×10^{-6}	4.6×10^{-6}	4.6×10^{-6}
Beam	$1.2 \times 10^{+1}$	$2.4 \times 10^{+0}$	$2.4 \times 10^{+0}$	$1.7 \times 10^{+0}$	$1.7 \times 10^{+0}$	$2.0 \times 10^{+0}$	$2.0 \times 10^{+0}$

Table 2: Absolute errors of the reduced-order models computed for the examples employed in the numerical evaluation of the parallel model reduction routines.

In a few cases, the theoretical error bound (10) is not satisfied by the computed reduced-order models. This might be due to rounding errors in computing the models and/or the H_∞-norms. A detailed investigation of this discrepancy between theory and practice is necessary.

3.2 Parallel efficiency

In this subsection, we report performance with respect to execution times and parallel efficiency using the BT-BFSR, SPA-BFSR, and HNA methods to reduce some large-scale systems. Similar results were obtained when the SR approach was employed in all examples.

3.2.1 Large-scale random example

We analyze the performance of our parallel model reduction algorithms using a random continuous LTI system constructed as follows. First, we generate a random positive semidefinite diagonal Gramian $W_c = \mathrm{diag}\,(\Sigma_{q_1}, \Sigma_{q_2}, 0_{q_3}, 0_{q_4})$, where $\Sigma_{q_1} \in \mathbb{R}^{q_1 \times q_1}$ contains the desired Hankel singular values for the system and $\Sigma_{q_2} \in \mathbb{R}^{q_2 \times q_2}$. Then, we construct a random positive semidefinite diagonal Gramian $W_o = \mathrm{diag}\,(\Sigma_{q_1}, 0_{q_2}, \Sigma_{q_3}, 0_{q_4})$, with $\Sigma_{q_3} \in \mathbb{R}^{q_3 \times q_3}$. Next, we set A to a random stable diagonal matrix and compute $F = -(AW_c + W_cA^T)$ and $G = -(A^TW_o + W_oA)$. Thus,

$$
\begin{aligned}
F &= \mathrm{diag}\,(f_1, f_2, \ldots, f_{q_1+q_2}, 0_{q_3+q_4})\,, \\
G &= \mathrm{diag}\,(g_1, g_2, \ldots, g_{q_1}, 0, \ldots, 0, g_{q_1+q_2+1}, \ldots, g_{q_1+q_2+q_3}, 0_{q_4})\,.
\end{aligned}
$$

A matrix $B \in \mathbb{R}^{n \times (q_1+q_2)}$ such that $F = BB^T$ is then obtained as

$$
B = \mathrm{diag}\left(\sqrt{f_1}, \sqrt{f_2}, \ldots, \sqrt{f_{q_1+q_2}}\right).
$$

The procedure for obtaining C is analogous. The LTI system is finally transformed into $A := U^T AU$, $B := U^T B$, and $C := CU$ using a random orthogonal transformation $U \in \mathbb{R}^{n\times n}$. The system thus defined has a minimal realization of order $r = q_1$. The Cholesky factors satisfy rank $(S) = q_1 + q_2$ and rank $(R) = q_1 + q_3$.

We first evaluate the execution time of the parallel model reduction algorithms. In the example, we set $n = 1000$, $m = p = 100$, and $q_1 = q_2 = q_3 = 50$. As there is no noticeable gap in the Hankel singular value distribution of the system, we obtain a reduced-order model of order $r = 40$.

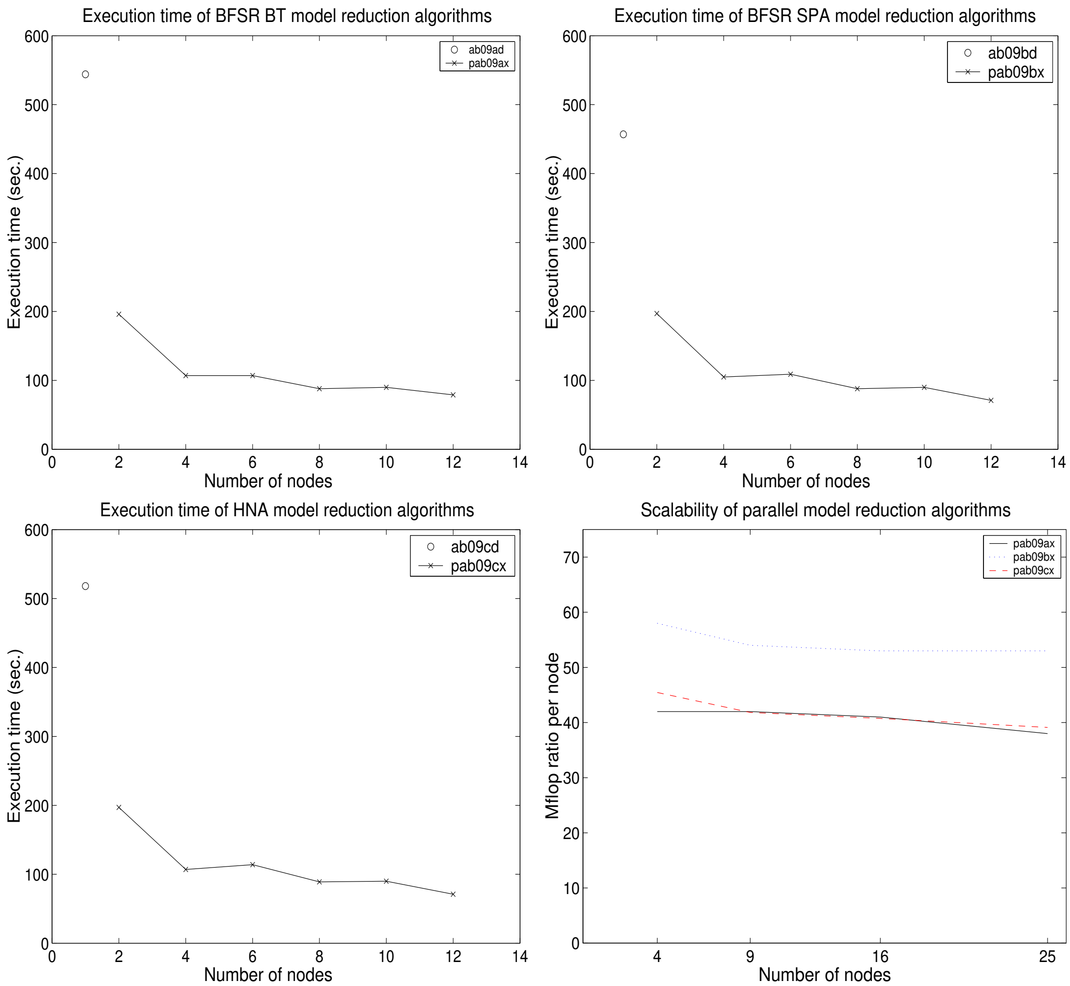

Figure 17: Performance of the parallel model reduction algorithms.

Figure 17 reports the execution times of the serial routines for model reduction in SLICOT and the corresponding parallel algorithms as the number of nodes, n_p, is increased. The results show a considerable acceleration achieved by the parallel algorithm (with even super speed-ups). This is partially due to the efficiency of the Lyapunov solvers used in our algorithms which compute factors of the Gramians in compact (full-rank) form instead of square matrices, thus requiring less computations. Comparison of the results on 2 and 4 nodes roughly shows the efficiency of the parallel algorithm. The execution time is reduced by a factor of almost 2 (the number of resources, that is nodes, is doubled). Using a larger number of nodes does not achieve a significant reduction of the execution time due to the small ratio

$n/\sqrt{n_p}$.

We next evaluate the scalability of our parallel algorithms. As the memory of the system does not allow to test the serial algorithms on larger problems, in the experiment we fix the problem size per node using $n/\sqrt{n_p} = 800$, $m/\sqrt{n_p} = 400$, $p/\sqrt{n_p} = 400$, and $q/\sqrt{n_p} = 200$, with $q_1 = q_2 = q_3 = q$. In the bottom right plot in Figure 17 we report the Mflop ratio per node for the parallel model reduction algorithms. The figure shows a high scalability of all three algorithms, as there is only a minor decrease in the Mflop ratio per node as the number of nodes is increased up to 25 (a problem of size $n = 4000$). The scalability confirms that a larger problem can be solved by increasing proportionally the number of nodes employed.

3.2.2 Large-scale rail example

A second large-scale continuous LTI model was employed in the performance evaluation. This system comes from a finite element discretization of a steel cooling process described by a boundary control problem for a linearized 2-dimensional heat equation; see [21] and references therein. The system has 6 inputs and outputs, and $n = 821$ or 3113 states, depending on the meshsize. As there is no significant gap in the Hankel singular values of the system, in this experiment we computed a reduced-order system of fixed order $r = 40$.

The system with $n = 821$ was reduced using the SLICOT routines in about 4 minutes. The parallel routines computed the reduced-order system in half of this time using 4 processors. The system with $n = 3113$ could not be solved using the serial routines as the system matrices were too large to be stored in a single node. Our parallel routines provided the solution in around 15 minutes (using 16 processors). This shows a second relevant advantage of parallel computing: larger problems can be solved due to the larger storage capacity.

3.2.3 Heat equation example: discrete case

We use again the discrete LTI system resulting from a full discretization of a control problem for the 1-dimensional heat equation from subsection 3.1.7. In all tests, a reduced-order model of size $r = 9$ is computed. In the comparisons we omit the HNA routines from SLICOT/PSLICOT as they only use a bilinear transformation and employ the corresponding algorithms for the continuous-time case.

Figure 18 shows the execution times for the PSLICOT routines (results on 1 node) and the PSLICOT routines (results on 2, 4, 6, and 8 nodes).

A significant speed-up is obtained by the parallel routines. As there is a superlinear speed-up from 1 to 2 nodes this shows again that the different computational approach used in the parallel routines (i.e., using full-rank factors of the Gramians instead of Cholesky factors) often is a lot more efficient than that used in SLICOT.

Figure 19 shows the scalability of the parallel codes analogous to subsection 3.2.1. The figure shows a high scalability of the codes with Mflop rates that are even a little better than in the continuous-time test reported in subsection 3.2.1.

4 Concluding Remarks

We have presented an experimental evaluation of the numerical behaviour and the performance of the parallel algorithms for model reduction in PLiCMR.

No difference were encountered between the use of the SR and the BFSR approaches in any these examples. Also, closely similar frequency responses were obtained for the TFM of the reduced-order

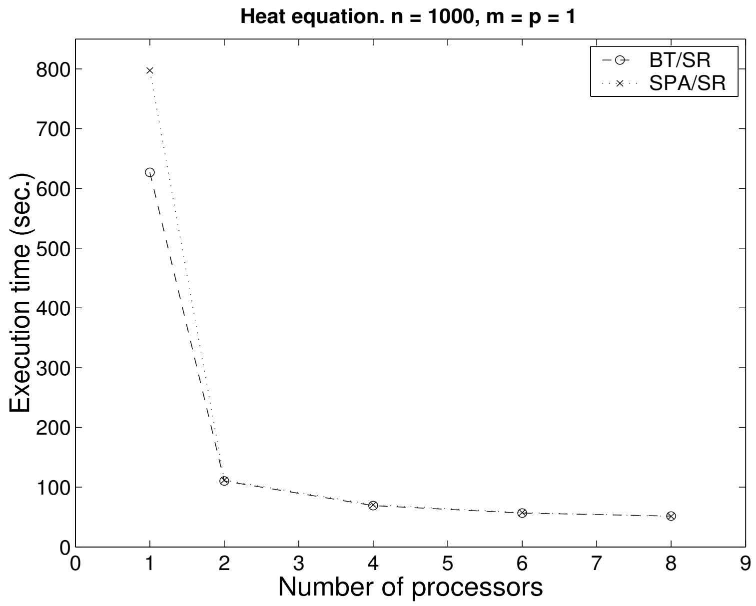

Figure 18: Execution times for BT and SPA approaches.

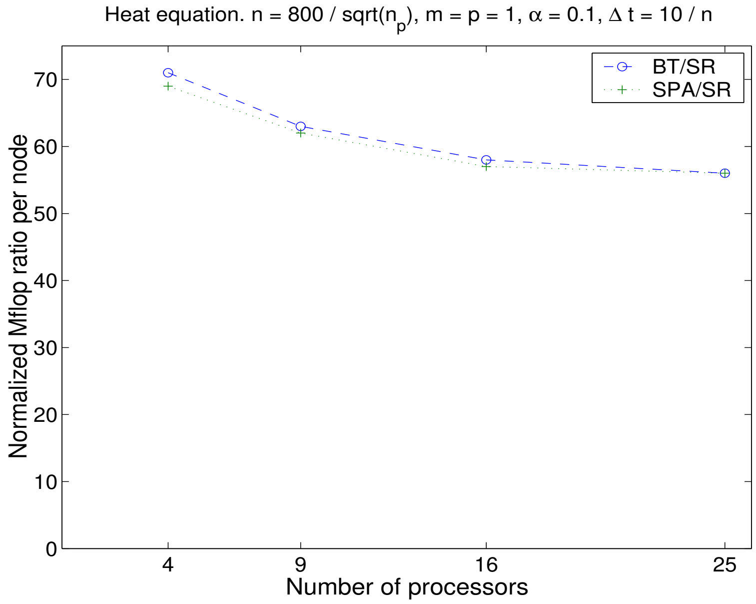

Figure 19: Scalability of the parallel codes for the discrete heat equation.

systems computed by the serial routines from SLICOT and the parallel routines in PLiCMR.

References

[1] I. Blanquer, D. Guerrero, V. Hernández, E.S. Quintana-Ortí, and P. Ruíz. Parallel-SLICOT Implementation and Documentation Standards. SLICOT Working Note 1998–1, September 1998. Available from `http://www.win.tue.nl/niconet/NIC2/reports.html`.

[2] P. Benner, R. Mayo, E.S. Quintana-Ortí, and G. Quintana-Ortí. Enhanced services for remote model reduction of large-scale dense linear systems. SLICOT Working Note 2002–1, January 2002. Available from `http://www.win.tue.nl/niconet/NIC2/reports.html`.

[3] P. Benner, R. Mayo, E.S. Quintana-Ortí, and G. Quintana-Ortí. Enhanced services for remote model reduction of large-scale dense linear systems. In *Proc. of the PARA'02 Conference. Lecture Notes in Computer Science 2367*, J. Fagerholm, J. Haataja, J. Jarvinen, M. Lyly, P. Raback, and V. Savolainen, Eds., pages 329–338, 2002.

[4] P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. Balanced truncation model reduction of large-scale dense systems on parallel computers. Mathematical and Computer Modeling of Dynamical Systems, Vol. 6(4), pp. 383–405, 2000.

[5] P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. Efficient numerical model reduction methods for discrete-time systems. Proceedings 3rd IMACS Symposium on Mathematical Modeling, volume 1, pages 277–280, 2000.

[6] P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. Singular perturbation approximation of large, dense linear systems. In *Proc. 2000 IEEE Intl. Symp. CACSD, Anchorage, Alaska, USA, September 25–27, 2000*, pages 255–260. Omnipress, Madison, WI, 2000.

[7] P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. PSLICOT routines for model reduction of stable large-scale systems. Proceedings Third NICONET Workshop on Numerical Software in Control Engineering, pages 39–44, 2001.

[8] L.S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R.C. Whaley. *ScaLAPACK Users' Guide*. SIAM, Philadelphia, PA, 1997.

[9] Y. Chahlaoui and P. Van Dooren. A collection of benchmark examples for model reduction of linear time invariant dynamical systems. SLICOT Working Note 2002–2, February 2002. Available from `http://www.win.tue.nl/niconet/NIC2/reports.html`.

[10] J. Cheng, G. Ianculescu, C.S. Kenney, A.J. Laub, and P. M. Papadopoulos. Control-structure interaction for space station solar dynamic power module. *IEEE Control Systems*, pages 4–13, 1992.

[11] P. Y. Chu, B. Wie, B. Gretz, and C. Plescia. Approach to large space structure control system design using traditional tools. *AIAA J. Guidance, Control, and Dynamics*, 13:874–880, 1990.

[12] Z. Drmač. Accurate Computation of the Product-Induced Singular Value Decomposition with Applications. *SIAM J. Numer. Anal.*, 35(5):1969–1994, 1998.

[13] L. Fortuna, G. Nummari, and A. Gallo. *Model Order Reduction Techniques with Applications in Electrical Engineering*. Springer-Verlag, 1992.

[14] J. Gunnels, G. Henry, R. van de Geijn. A family of high-performance matrix multiplication algorithms. In *Computational Science - ICCS 2001, Part I, Lecture Notes in Computer Science 2073*, V. Alexander, J. Dongarra, B. Julianno, R. Renner, and C. Kenneth Tan, Eds. 51–60.

[15] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞ norms. *Internat. J. Control*, 39:1115–1193, 1984.

[16] S.J. Hammarling. Numerical solution of the stable, non-negative definite Lyapunov equation. *IMA J. Numer. Anal.*, 2:303–323, 1982.

[17] Y. Liu and B.D.O. Anderson. Controller reduction via stable factorization and balancing. *Internat. J. Control*, 44:507–531, 1986.

[18] The MathWorks, Inc., Cochituate Place, 24 Prime Park Way, Natick, Mass, 01760. *The MATLAB Robust Control Toolbox, Version 2.0.7*, 2000.

[19] B. C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. Automat. Control*, AC-26:17–32, 1981.

[20] C.R. Paul. *Analysis of Multiconductor Transmission Lines.* Wiley–Interscience, Singapur, 1994.

[21] T. Penzl. Algorithms for model reduction of large dynamical systems. Technical Report SFB393/99-40, Sonderforschungsbereich 393 *Numerische Simulation auf massiv parallelen Rechnern*, TU Chemnitz, Germany, 1999.

[22] M.G. Safonov and R.Y. Chiang. A Schur method for balanced-truncation model reduction. *IEEE Trans. Automat. Control*, AC–34:729–733, 1989.

[23] 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.

[24] A. Varga. A note on Hammarling's algorithm for the discrete Lyapunov equation. *Sys. Control Lett.*, 15(3):273–275, 1990.

[25] A. Varga. Efficient minimal realization procedure based on balancing. In *Prepr. of the IMACS Symp. on Modeling and Control of Technological Systems*, volume 2, pages 42–47, 1991.

[26] R.C. Whaley and J. Dongarra. Automatically tuned linear algebra software. In *Proc. of SC98*, 1998. (Electronic publication).