Received: 1 August 2019 | Revised: 3 April 2020 | Accepted: 29 May 2020

DOI: 10.1002/zamm.201900205

EDITOR'S CHOICE

Low-rank linear fluid-structure interaction discretizations

Roman Weinhandl[1,2] | **Peter Benner**[1,2] | **Thomas Richter**[1]

[1] Institute for Analysis and Numerics, Otto von Guericke University Magdeburg, Universitaetsplatz 2, Magdeburg 39106, Germany

[2] Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstrasse 1, Magdeburg 39106, Germany

Correspondence
Roman Weinhandl, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstrasse 1, 39106 Magdeburg, Germany.
Email: weinhandl@mpi-magdeburg.mpg.de

Funding information
Deutsche Forschungsgemeinschaft, Grant/Award Number: 314838170, GRK 2297 MathCoRe

Fluid-structure interaction models involve parameters that describe the solid and the fluid behavior. In simulations, there often is a need to vary these parameters to examine the behavior of a fluid-structure interaction model for different solids and different fluids. For instance, a shipping company wants to know how the material, a ship's hull is made of, interacts with fluids at different Reynolds and Strouhal numbers before the building process takes place. Also, the behavior of such models for solids with different properties is considered before the prototype phase. A parameter-dependent linear fluid-structure interaction discretization provides approximations for a bundle of different parameters at one step. Such a discretization with respect to different material parameters leads to a big block-diagonal system matrix that is equivalent to a matrix equation as discussed in [1]. The unknown is then a matrix which can be approximated using a low-rank approach that represents the iterate by a tensor. This paper discusses a low-rank GMRES variant and a truncated variant of the Chebyshev iteration. Bounds for the error resulting from the truncation operations are derived. Numerical experiments show that such truncated methods applied to parameter-dependent discretizations provide approximations with relative residual norms smaller than 10^{-8} within a twentieth of the time used by individual standard approaches.

KEYWORDS
ChebyshevT, GMREST, low-rank, parameter-dependent fluid-structure interaction, tensor

1 | INTRODUCTION

A parameter-dependent linear fluid-structure interaction problem as described in Section 2 discretized using bilinear finite elements with a total number of $M \in \mathbb{N}$ degrees of freedom (see Section 3 for details) and $m \in \mathbb{N}$ parameter combinations leads to equations of the form

$$\left(A_0 + \mu_s^i A_1 + \lambda_s^i A_2 + \rho_f^i A_3\right) x_i = b_D \quad \text{for} \quad i \in \{1, \dots, m\}, \tag{1}$$

where the discretization matrices $A_0, A_1, A_2, A_3 \in \mathbb{R}^{M \times M}$ and the right hand side $b_D \in \mathbb{R}^M$ depends on the Dirichlet data and the ith finite element solution $x_i \in \mathbb{R}^M$. The samples of interest are given by the shear moduli $\mu_s^i \in \mathbb{R}$, the first Lamé parameters $\lambda_s^i \in \mathbb{R}$ and the fluid densities ρ_f^i for $i \in \{1, \dots, m\}$.

Equation (1) can directly be written as the linear system

$$\mathcal{A}\begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} b_D \\ \vdots \\ b_D \end{pmatrix}, \tag{2}$$

where $\mathcal{A} \in \mathbb{R}^{Mm \times Mm}$ is a block diagonal matrix. Following [1], Equation (2) can then be translated into the matrix equation

$$A_0 X + A_1 X D_1 + A_2 X D_2 + A_3 X D_3 = B \tag{3}$$

with $B := [\, b_D \big| \cdots \big| b_D \,]$ and diagonal matrices $D_1, D_2, D_3 \in \mathbb{R}^{M \times M}$, where the ith diagonal entry of these diagonal matrices is given by μ_s^i, λ_s^i and ρ_f^i, respectively. In (3), the unknown

$$X = \left[\, x_1 \middle| \cdots \middle| x_m \,\right] \in \mathbb{R}^{M \times m}$$

is a matrix. Now an iterative method to solve linear systems can be modified such that it uses an iterate that is a matrix. It is applied to the big system (2) but computation is kept in the matrix notation (3) by representing the iterate as a matrix instead of a vector. The methods used in this paper fix a rank $R \in \mathbb{N}$, $R \ll M, m$ and represent this iterate as a tensor. The goal is to find a low-rank approximation $\hat{X}$ of rank R

$$\hat{X} = \sum_{j=1}^{R} u_j \otimes v_j^T, u_j \in \mathbb{R}^M \quad \text{and} \quad v_j \in \mathbb{R}^M \ \forall j \in \{1, \dots, R\}$$

that approximates the full matrix X from (3) and therefore provides (parameter-dependent) finite element approximations for all equations in (1).

Fluid-structure interaction problems yield non-symmetric system matrices. Hence, the system matrix $\mathcal{A}$ in (2) is not symmetric. The methods examined in this paper are based on the GMRES method as introduced in a truncated variant in [2] and the Chebyshev method from [3]. These methods will then be compared to a truncated method based on the Bi-CGstab method from [4] similar to Algorithm 3 of [1] and Algorithm 2 of [5].

In Section 2 and Section 3, we derive the matrix equations that appear when dealing with parameter-dependent fluid-structure interaction discretizations. The low-rank framework and related methods are introduced in Section 4 for stationary problems and generalized to non-stationary problems in Section 5. In Section 6, theoretical error bounds for the GMREST and the ChebyshevT method are derived and numerically evaluated in Section 7. The convergences of the truncated approaches presented are compared in numerical experiments in Section 8.

2 | THE STATIONARY LINEAR FLUID-STRUCTURE INTERACTION PROBLEM

Let $d \in \{2, 3\}$, $\Omega \subset \mathbb{R}^d$, $F, S \subset \Omega$ such that $\bar{F} \cup \bar{S} = \bar{\Omega}$ and $F \cap S = \emptyset$, where F represents the fluid and S the solid part. Let $\Gamma_{\text{int}} = \partial F \cap \partial S$ and $\Gamma_f^{\text{out}} \subset \partial F \setminus \partial S$ denote the boundary part where Neumann outflow conditions hold. $\Gamma_f^D = \partial F \setminus (\Gamma_f^{\text{out}} \cup \Gamma_{\text{int}})$ denotes the boundary part where Dirichlet conditions hold. Consider the Stokes fluid equations from Section 2.4.4 of [6] as a model for the fluid part and the Navier–Lamé equations discussed as Problem 2.23 of [6] as a model for the solid part. Both equations are assumed to have a vanishing right hand side. If these two equations are coupled with the kinematic and the dynamic coupling conditions discussed in Section 3.1 of [6], the weak formulation of the stationary coupled linear fluid-structure interaction problem reads

$$\begin{aligned} \langle \nabla \cdot v, \xi \rangle_F &= 0, \\ \mu_s \langle \nabla u + \nabla u^T, \nabla \varphi \rangle_S + \lambda_s \langle \operatorname{div} u, \operatorname{div} \varphi \rangle_S + \nu_f \rho_f \langle \nabla v + \nabla v^T, \nabla \varphi \rangle_F - \langle p, \nabla \cdot \varphi \rangle_F &= 0, \\ \langle \nabla u, \nabla \psi \rangle_F &= 0, \end{aligned} \tag{4}$$

with the trial functions $v \in v_{\text{in}} + H_0^1(\Omega, \Gamma_f^D \cup \Gamma_{\text{int}})^d$ (velocity), where $v_{\text{in}} \in H^1(\Omega)^d$ is an extension of the Dirichlet data on Γ_f^D, $u \in H_0^1(\Omega)^d$ (deformation) and $p \in L^2(F)$ (pressure) and the test functions $\xi \in L^2(F)$ (divergence equation), $\varphi \in H_0^1(\Omega, \partial\Omega \setminus \Gamma_f^{\text{out}})^d$ (momentum equation) and $\psi \in H_0^1(F)^d$ (deformation equation). $\langle \cdot, \cdot \rangle_S$ and $\langle \cdot, \cdot \rangle_F$ denote the $\mathcal{L}^2$ scalar product on S and F, respectively. $\nu_f \in \mathbb{R}$ denotes the kinematic fluid viscosity and $\rho_f \in \mathbb{R}$ the fluid density. The shear modulus $\mu_s \in \mathbb{R}$ and the first Lamé parameter $\lambda_s \in \mathbb{R}$ determine the Poisson ratio of the solid.

Definition 1 (**The Poisson Ratio - Definition 2.18 of [6]**). The Poisson ratio of a solid is given by the number

$$\nu_s^p = \frac{\lambda_s}{2(\lambda_s + \mu_s)}.$$

It describes the compressibility of a solid.

3 | PARAMETER-DEPENDENT DISCRETIZATION

Assume the behavior of a linear fluid-structure interaction model for $m_1 \in \mathbb{N}$ shear moduli, $m_2 \in \mathbb{N}$ first Lamé parameters and $m_3 \in \mathbb{N}$ fluid densities is of interest. The kinematic fluid viscosity $\nu_f \in \mathbb{R}$ is assumed to be fixed. Let the samples of interest be given by the following sets.

$$\{\mu_s^{i_1}\}_{i_1 \in \{1,\dots,m_1\}} \subset \mathbb{R}^+, \text{ a set of shear moduli,}$$

$$\{\lambda_s^{i_2}\}_{i_2 \in \{1,\dots,m_2\}} \subset \mathbb{R}^+, \text{ a set of first Lamé parameters and}$$

$$\{\rho_f^{i_3}\}_{i_3 \in \{1,\dots,m_3\}} \subset \mathbb{R}^+, \text{ a set of fluid densities.}$$

In a bilinear finite element discretization of (4) with a mesh grid size of $N \in \mathbb{N}$, every mesh grid point corresponds to a pressure, a velocity and a deformation variable. In two dimensions, the velocity and deformation are two dimensional vectors, in three dimensions they correspond to a three dimensional vector each. The total number of degrees of freedom is therefore $M = 5N$ in two dimensions and $M = 7N$ in three dimensions.

Let Ω_h be a matching mesh of the domain Ω as defined in Definition 5.9 of [6] with N mesh grid points. $A_0 \in \mathbb{R}^{M \times M}$ is a discrete differential operator restricted to the finite element space with dimension M. It discretizes all operators involved in (4) with a fixed shear modulus $\mu_s \in \mathbb{R}$, a fixed first Lamé parameter $\lambda_s \in \mathbb{R}$ and a fixed fluid density $\rho_f \in \mathbb{R}$. In this paper, Q_1 finite elements as discussed in Section 4.2.1 of [6] are used and we will denote the discrete differential operators by discretization matrices. Moreover, let $A_1, A_2, A_3 \in \mathbb{R}^{M \times M}$ be the discretization matrices of the following operators:

$$\begin{aligned} A_1 &\quad \text{discretizes} \quad \langle \nabla u + \nabla u^T, \nabla \varphi \rangle_S, \\ A_2 &\quad \text{discretizes} \quad \langle \operatorname{tr}(\nabla u) I, \nabla \varphi \rangle_S \quad \text{and} \\ A_3 &\quad \text{discretizes} \quad \langle \nabla v + \nabla v^T, \nabla \varphi \rangle_F. \end{aligned}$$

The parameter-dependent equation

$$\underbrace{\left(A_0 + (\mu_s^{i_1} - \mu_s)A_1 + (\lambda_s^{i_2} - \lambda_s)A_2 + \nu_f(\rho_f^{i_3} - \rho_f)A_3\right)}_{=: A(\mu_s^{i_1}, \lambda_s^{i_2}, \rho_f^{i_3})} x_{i_1 i_2 i_3} = b_D \quad \text{for} \tag{5}$$

$$(i_1, i_2, i_3) \in \{1, \dots, m_1\} \times \{1, \dots, m_2\} \times \{1, \dots, m_3\}$$

is the finite element discretization of (4) related to a shear modulus $\mu_s^{i_1}$, a first Lamé parameter $\lambda_s^{i_2}$ and a fluid density $\rho_f^{i_3}$. The finite element solution is $x_{i_1 i_2 i_3} \in \mathbb{R}^M$ and the right hand side $b_D \in \mathbb{R}^M$ depends on the Dirichlet data.

Remark 1. If the fixed parameters vanish, namely $\mu_s = \lambda_s = \rho_f = 0$, (5) translates to

$$\left(A_0 + \mu_s^{i_1} A_1 + \lambda_s^{i_2} A_2 + \nu_f \rho_f^{i_3} A_3\right) x_{i_1 i_2 i_3} = b_D \quad \text{for} \quad (i_1, i_2, i_3) \in \{1, \dots, m_1\} \times \{1, \dots, m_2\} \times \{1, \dots, m_3\}.$$

At first sight, this presentation seems to be more convenient. But choosing, for instance, the parameters $\mu_s = \mu_s^1$, $\lambda_s = \lambda_s^1$ and $\rho_f = \rho_f^1$ minimizes the number of nonzero entries in the diagonal matrices $D_1, D_2, D_3 \in \mathbb{R}^{m \times m}$ that will be introduced in (7). From a numerical point of view, this is an advantage. Furthermore, vanishing fixed parameters would lead to a singular matrix A_0. This can become a problem if the preconditioner $\mathcal{P}_{A_0}$ from Section 3.2 is used.

Combining all sample combinations in (5) leads to a total of $m = m_1 m_2 m_3$ equations. Written as a linear system, these equations translate to

$$\overbrace{\operatorname*{diag}_{\substack{i_1 \in \{1,\dots,m_1\} \\ [0pt] i_2 \in \{1,\dots,m_2\} \\ i_3 \in \{1,\dots,m_3\}}} \left(A(\mu_s^{i_1}, \lambda_s^{i_2}, \rho_f^{i_3})\right)}^{=: \mathcal{A}} \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1 m_2 m_3} \end{pmatrix} = \begin{pmatrix} b_D \\ \vdots \\ b_D \end{pmatrix}, \tag{6}$$

where diag(·) denotes the operator introduced in Section 1.2.6 of [7] extended to block diagonalization. Even though $\mathcal{A} \in \mathbb{R}^{Mm \times Mm}$ is of block diagonal structure, solving the blocks on the diagonal m times (potentially in parallel) is often not feasible. If 100 samples per parameter are considered, one would have to face $100^3 = 10^6$ blocks already in such a direct approach. This would lead to huge storage requirements for the solution vectors.

3.1 | The matrix equation

The diagonal entries of D_1, D_2 and $D_3 \in \mathbb{R}^{m \times m}$ are $\mu_s^{i_1} - \mu_s$, $\lambda_s^{i_2} - \lambda_s$ and $\rho_f^{i_3} - \rho_f$, respectively. The order of the diagonal entries has to be chosen such that every parameter combination occurs only once. If $I_{m_1} \in \mathbb{R}^{m_1 \times m_1}$ denotes the $m_1 \times m_1$ identity matrix, a possible sample order would lead to matrices

$$D_1 = I_{m_2 m_3} \otimes \operatorname*{diag}_{\substack{i_1 \in \\ [0pt] \{1,\dots,m_1\}}} (\mu_s^{i_1}), \quad D_2 = I_{m_3} \otimes \operatorname*{diag}_{\substack{i_2 \in \\ [0pt] \{1,\dots,m_2\}}} (\lambda_s^{i_2}) \otimes I_{m_1} \quad \text{and} \quad D_3 = \operatorname*{diag}_{\substack{i_3 \in \\ [0pt] \{1,\dots,m_3\}}} (\rho_f^{i_3}) \otimes I_{m_1 m_2}. \tag{7}$$

As discussed in [1], Equation (2) can then be written as the matrix equation

$$\underbrace{A_0 + A_1 X D_1 + A_2 X D_2 + \nu_f A_3 X D_3}_{=: F(X)} = B := [b_D | \cdots | b_D] \tag{8}$$

where the unknown is the matrix $X = [x_1 | \cdots | x_{m_1 m_2 m_3}] \in \mathbb{R}^{M \times m}$ whose ith column corresponds to the finite element approximation of (4) related to the ith sample combination.

Definition 2 (**Vector and Matrix Notation**). We refer to the representation (6) and (8) as the vector and the matrix notation, respectively. In (6), the unknown is a vector whereas in (8), the unknown is a matrix. Even though both equations express the same, for the theoretical proofs in Section 6, the vector notation is more suitable since considering spaces that are spanned by matrices is rather uncommon. On the other hand, software implementations exploit the low-rank structure of the matrix X in (8). This is why the matrix notation fits better in these cases.

In (5), the right hand side b_D does not depend on any parameter and $A(\mu_s^{i_1}, \lambda_s^{i_2}, \rho_f^{i_3})$ depends linearly on each parameter. Assume that $A(\mu_s^{i_1}, \lambda_s^{i_2}, \rho_f^{i_3})$ is invertible for all parameters. In this case, Theorem 3.6 of [1] is applicable and provides existence of a low-rank approximation of X in (8) with an error bound that implies a stronger error decay than any polynomial in the rank R. However, the constant C in Theorem 3.6 of [1] can become very big but we do not want to go into detail here and refer the interested reader to [1].

3.2 | Preconditioners

The system matrix $\mathcal{A}$ has the structure

$$\mathcal{A} = I_m \otimes A_0 + D_1 \otimes A_1 + D_2 \otimes A_2 + \nu_f D_3 \otimes A_3.$$

Promising choices of preconditioners that were used already in [1] are

$$\mathcal{P}_{A_0} := I_m \otimes A_0$$

or

$$\mathcal{P}_T := I_m \otimes \underbrace{\left(A_0 + \bar{\mu}_s A_1 + \bar{\lambda}_s A_2 + \nu_f \bar{\rho}_f A_3\right)}_{=:P_T}$$

with the means

$$\bar{\mu}_s = \frac{\min\limits_{i_1 \in \{1,\dots,m_1\}} (\mu_s^{i_1} - \mu_s) + \max\limits_{i_1 \in \{1,\dots,m_1\}} (\mu_s^{i_1} - \mu_s)}{2}, \quad \bar{\lambda}_s = \frac{\min\limits_{i_2 \in \{1,\dots,m_2\}} (\lambda_s^{i_2} - \lambda_s) + \max\limits_{i_2 \in \{1,\dots,m_2\}} (\lambda_s^{i_2} - \lambda_s)}{2} \quad \text{and}$$

$$\bar{\rho}_f = \frac{\min\limits_{i_3 \in \{1,\dots,m_3\}} (\rho_f^{i_3} - \rho_f) + \max\limits_{i_3 \in \{1,\dots,m_3\}} (\rho_f^{i_3} - \rho_f)}{2}.$$

The preconditioner $\mathcal{P}_T$ usually provides faster convergence than $\mathcal{P}_{A_0}$, especially if the means $\bar{\mu}_s, \bar{\lambda}_s$ and $\bar{\rho}_f$ are big. Left multiplication of $\mathcal{P}_T^{-1}$ with

$$\mathcal{A}\begin{pmatrix} x_1 \\ \vdots \\ x_{m_1 m_2 m_3} \end{pmatrix}$$

is equivalent to application of P_T^{-1} to $F(X)$ from the left using the matrix notation from (8).

4 | THE LOW-RANK METHODS

Now, we discuss iterative methods that can be applied to solve the big system (6). The iterate is then a vector $x \in \mathbb{R}^{Mm}$. But if the iterate is represented as a matrix instead of a vector, computation can be kept in the matrix notation from (8). For instance, the matrix-vector multiplication in such a global approach corresponds to the evaluation of the function $F(\cdot)$ from (8). The Euclidean norm of the vector

$$\begin{pmatrix} x_1 \\ \vdots \\ x_{m_1 m_2 m_3} \end{pmatrix}$$

from (6) then equals the Frobenius norm of the matrix X in (8), $\|X\|_F$. Low-rank methods that use this approach can be based on many methods such as the Richardson iteration or the conjugate gradient method as discussed in Algorithm 1 and Algorithm 2 of [1]. But since for fluid-structure interaction problems, the matrix $\mathcal{A}$ is not symmetric, the focus in this paper lies on methods that base on the GMRES and the Chebyshev method. As proved in Theorem 35.2 of [8], the GMRES method converges in this case, and so does the Chebyshev method, if all eigenvalues of the system matrix lie in an ellipse that does not touch the imaginary axis as proved in [3]. Also, the Bi-CGSTAB method from [4] is considered for a numerical comparison.

As mentioned, the low-rank methods discussed in this paper use an iterate that is, instead of a matrix, a tensor of order two. The iterate is then given by

$$\hat{X} = \sum_{j=1}^{R} = (u_j \otimes v_j^T) \quad \text{with} \quad u_j \in \mathbb{R}^M, v_j \in \mathbb{R}^m \; \forall j \in \{1, \dots, R\},$$

where the tensor rank $R \in \mathbb{N}$ is kept small such that $R \ll M, m$. The goal of the method is to find a low-rank approximation $\hat{X}$ that approximates the matrix X in (8). The methods GMREST (also mentioned in [2]) and ChebyshevT are such methods and will be explained in the following. They are not just faster than the standard methods applied to m individual equations of the form (5), they also need a smaller amount of storage to store the approximation. If M and m are very big, this plays an important role since the storage amount to store $\hat{X}$ is in $O((M+m)R)$ while the storage amount to store the full matrix X is in $O(Mm)$.

4.1 | Tensor format and truncation

There are several formats available to represent the tensor $\hat{X}$. For $d = 2$, the hierarchical Tucker format (Definition 11.11 of [9]) is equivalent to the Tucker format. It is based on so called minimal subspaces that are explained in Chapter 6 of [9].

Definition 3 (**Tucker Format - Definition 8.1 of [9] for $d = 2$**). Let $V := \mathbb{R}^M \otimes \mathbb{R}^m$, $(r_1, r_2) \in \mathbb{N}^2$. For $d = 2$, the Tucker tensors of Tucker rank (r_1, r_2) are given by the set

$$T_{(r_1,r_2)}(V) := \{v \in V_1 \otimes V_2 \,:\, V_1 \subset \mathbb{R}^M, \; \dim(V_1) = r_1, V_2 \subset \mathbb{R}^m, \; \dim(V_2) = r_2\}.$$

From now on, the set $T_{(R,R)}(V)$ will be denoted by T_R. By a tensor of rank R, a Tucker tensor in $T_{(R,R)}$ is addressed in the following.

As explained in Section 13.1.4 of [9], summation of two arbitrary Tucker tensors of rank R, in general, results in a Tucker tensor of rank $2R$. But to keep a low-rank method fast, the rank of the iterate has to be kept small. This induces the need for a truncation operator.

Definition 4 (**Truncation Operator**). The truncation operator

$$\mathcal{T} \,:\, \mathbb{R}^M \otimes \mathbb{R}^m \to T_R$$

maps a Tucker or a full tensor into the set of Tucker tensors of rank R. The truncation operator is, in the case of tensors of order 2, based on the singular value decomposition and projects its arguments to T_R. For further reading, we refer to Definition 2.5 of [10].

Remark 2. As proved in Section 3.2.3 of [9], it holds

$$\mathbb{R}^M \otimes \mathbb{R}^m \cong \mathbb{R}^{M \times m},$$

where the relation $\cdot \cong \cdot$ denotes spaces that are isomorphic to each other (see Section 3.2.5 of [9]). Since, for our purposes, we consider a matrix that is represented by a tensor, we assume

$$\mathcal{T} \,:\, \mathbb{R}^{M \times m} \to T_R.$$

and if

$$\hat{x} \in T_R,$$

by $\hat{x}$, the full representation of the tensor in $\mathbb{R}^{Mm}$ in vector notation is addressed.

Before we proceed, one more definition is needed.

Definition 5 (**Vectorization restricted to** $\mathbb{R}^{M\times m}$)**.** The vectorization operator

$$\mathrm{vec} : \mathbb{R}^{M\times m} \to \mathbb{R}^{Mm},\ \mathrm{vec}\left(v_1\middle|\cdots\middle|v_m\right) \mapsto \begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix}$$

stacks matrix entries column wise into a vector. Its inverse maps to an $M \times m$ matrix:

$$\mathrm{vec}^{-1} : \mathbb{R}^{Mm} \to \mathbb{R}^{M\times m},\ \mathrm{vec}^{-1}\begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix} = (v_1|\cdots|v_m).$$

Remark 3. The argument of the function $F(\cdot)$ from (8) is tacitly assumed to be a matrix so $F(\hat{x})$ addresses $F(\mathrm{vec}^{-1}(\hat{x}))$ for $\hat{x} \in T_R$.

Since truncation is an operation that is applied after nearly every addition of tensors and multiple times in every iteration, the format that provides the truncation with the least complexity is often the preferred one. According to Algorithm 6 of [11], the htucker toolbox [11] for MATLAB® provides truncation with complexity $(2\max(M,m)R^2 + 2R^4)$ if the input format is in hierarchical Tucker format. The truncation complexity of the TT toolbox [12] for MATLAB that uses the Tensor Train format is in $O(2\max(M,m)R^3)$ as stated in Algorithm 2 of [12].

4.2 | The GMREST and the GMRESTR method

Consider $\mathcal{A}$ from (6), a suitable preconditioner $\mathcal{P} = I_m \otimes P \in \mathbb{R}^{Mm\times Mm}$, a start vector $x_0 \in \mathbb{R}^{Mm}$ and

$$b := \begin{pmatrix} b_D \\ \vdots \\ b_D \end{pmatrix}, \quad r_0 := \mathcal{P}^{-1}(b - \mathcal{A}x_0).$$

l GMRES iterations with the preconditioner $\mathcal{P}$ applied to the system (6) minimize $\|r_0 - \mathcal{P}^{-1}\mathcal{A}z\|_2$ for $z \in \mathbb{R}^{Mm}$ over the Krylov subspace (compare Section 6.2 of [13])

$$\mathcal{K}_l := \mathrm{span}\{r_0, \mathcal{P}^{-1}\mathcal{A}r_0, \ldots, (\mathcal{P}^{-1}\mathcal{A})^{l-1}r_0\}.$$

As mentioned before, from the theoretical point of view, this classical GMRES method is equivalent to the global GMRES method that uses an iterate that is a matrix instead of a vector. But if the iterate is represented by a tensor of a fixed rank R, the truncation operator $\mathcal{T}$ generates an additional error every time it is applied to truncate the iterate or tensors involved back to rank R. With an initial guess $\hat{x}_0 := \mathcal{T}(x_0)$,

$$\hat{b} := \mathcal{T}(b) \quad \text{and} \quad \hat{r}_0 := \mathcal{T}\left(P^{-1}[\hat{b} - F(\hat{x}_0)]\right),$$

l iterations of the truncated GMRES method GMREST that is coded in Algorithm 1 minimize $\|\mathrm{vec}(\mathcal{T}(\hat{r}_0 - P^{-1}F(\hat{z})))\|_2$ for $\hat{z} \in T_R$ over the truncated Krylov subspace

$$\mathcal{K}_l^{\mathcal{T}} := \mathrm{span}\{\mathrm{vec}(\hat{r}_0), \mathrm{vec}\left(\mathcal{T}\left(P^{-1}F(\hat{r}_0)\right)\right), \ldots, \mathrm{vec}\left(\left(\mathcal{T}(P^{-1}F)\right)^{l-1}(\hat{r}_0)\right)\}.$$

Algorithm 1 is a translation of a preconditioned variant of Algorithm 6.9 of [13] to the low-rank framework. The Arnoldi iteration is used to compute an orthogonal basis of $\mathcal{K}_l^{\mathcal{T}}$. The operations involved are translated from the vector notation

Algorithm 1 GMREST(l) (Preconditioned Truncated GMRES Method)

Input: Iteration number l, truncation rank R for $\mathcal{T}$, $F(\cdot)$ from (8), left preconditioner $P \in \mathbb{R}^{M\times M}$, right hand side $\hat{B} \in T_R$ and start matrix $\hat{X} \in T_R$

Output: Approximate solution $\hat{X} \in T_R$

Find $\hat{R} \in T_R$ such that $P\hat{R} = \mathcal{T}(\hat{B} - F(\hat{X}))$.

$z := (\|\hat{R}\|_F \quad 0 \cdots 0 \quad)^T$

$\hat{V}_1 := \frac{\hat{R}}{\|\hat{R}\|_F}$

for $i = 1, \dots, l$ **do**

Find $\hat{W} \in T_R$ such that $P\hat{W} = \mathcal{T}(F(\hat{V}_i))$.

for $k = 1, \dots, i$ **do**

$H_{k,i} := \text{trace}(\hat{V}_k^H \hat{W})$

$\hat{W} := \mathcal{T}(\hat{W} - H_{k,i}\hat{V}_k)$

end for

$H_{i+1,i} := \|\hat{W}\|_F$

$\hat{V}_{i+1} := \hat{W}\frac{1}{H_{i+1,i}}$

end for

Now find a unitary matrix Q such that QH is an upper triangular matrix via Givens rotations. Find y such that $QHy = Qz$.

$\hat{X} = \mathcal{T}(\hat{X} + \sum_{j=1}^{l} y_j \hat{V}_j)$

Algorithm 2 GMRESTR(l, d) (Preconditioned Truncated GMRES Restart Method)

Input: In addition to the inputs of Algorithm 1, a divisor $d \in \mathbb{N}$

Output: Approximate solution $\hat{X} \in T_R$

$d_1 := \text{floor}(\frac{l}{d})$

for $i = 1, \dots, i$ **do**

$\hat{X}$ =GMREST(d) with start matrix $\hat{X}$

end for

to the matrix notation. Therefore, the Euclidean norm of a vector translates to the Frobenius norm of a matrix. The scalar product of two vectors corresponds to the Frobenius scalar product of two matrices,

$$v^H \cdot w = \text{trace}\left(\text{vec}^{-1}(v)^H \,\text{vec}^{-1}(w)\right) \qquad \text{for} \qquad v, w \in \mathbb{R}^{Mm}.$$

But even the standard GMRES method can stagnate due to machine precision. This means that at the lth iteration, the dimension of the numerical approximation of $\mathcal{K}_l$ is smaller than l. As we will see later, the truncation operator brings, in addition to the finite precision error (round-off), a truncation error into play. As a result, the GMREST method can stagnate much earlier than the non truncated full approach. As in the full approach, restarting the method with the actual iterate as initial guess can be a remedy. This restarted variant of the GMREST method, called GMRESTR here, is coded in Algorithm 2.

4.3 | The ChebyshevT method

The Chebyshev method converges for non-symmetric system matrices if, in the complex plane, the eigenvalues can be encircled by an ellipse that does not touch the imaginary axis.

The diagonal blocks of the preconditioned system matrix $\mathcal{P}_T^{-1}\mathcal{A}$ are

$$Bl(i_1, i_2, i_3) := P_T^{-1}\left(A_0 + (\mu_s^{i_1} - \mu_s)A_1 + (\lambda_s^{i_2} - \lambda_s)A_2 + \nu_f(\rho_f^{i_3} - \rho_f)A_3\right) \qquad \text{for} \quad (i_1, i_2, i_3) \in \{1, \dots, m_1\} \times \{1, \dots, m_2\} \times \{1, \dots, m_3\}. \tag{9}$$

Algorithm 3 ChebyshevT(l, d, c) (Preconditioned Truncated Chebyshev Method)

Input: Iteration number l, ellipse by center d and foci $d \pm c$, truncation rank R for $\mathcal{T}$, $F(\cdot)$ from (8), left preconditioner $P \in \mathbb{R}^{M\times M}$, right hand side $\hat{B} \in T_R$ and start matrix $\hat{X} \in T_R$

Output: Approximate solution $\hat{X} \in T_R$

Find $\hat{R}_0$ such that $P\hat{R}_0 = \mathcal{T}(\hat{B} - F(\hat{X}))$

$\hat{\Phi}_0 := \frac{1}{d}\hat{R}_0$

$\hat{X} = \mathcal{T}(\hat{X} + \hat{\Phi}_0)$

$t_0 := 1$

$t_1 := \frac{d}{c}$

for $i = 1, \dots, l$ **do**

$t_{i+1} := 2\frac{d}{c}t_i - t_{i-1}$

$\alpha_i := \frac{2t_i}{ct_{i+1}}$

$\beta_i := \frac{t_{i-1}}{t_{i+1}}$

Find $\hat{R}_i$ such that $P\hat{R}_i = \mathcal{T}(\hat{B} - F(\hat{X}))$.

$\hat{\Phi}_i := \mathcal{T}(\alpha_i\hat{R}_i + \beta_i\hat{\Phi}_{i-1})$

$\hat{X} = \mathcal{T}(\hat{X} + \hat{\Phi}_i)$

end for

Moreover, the parameter-dependent matrices (5) are assumed to be invertible. The eigenvalues of $\mathcal{P}_T^{-1}\mathcal{A}$ denoted by $\Lambda(\mathcal{P}_T^{-1}\mathcal{A})$ therefore coincide with the set

$$\bigcup_{\substack{i_1\in\{1,\dots,m_1\}\\ [20pt] i_2\in\{1,\dots,m_2\}\\ i_3\in\{1,\dots,m_3\}}} \Lambda(Bl(i_1, i_2, i_3)). \tag{10}$$

In numerical tests, it turned out that $R_\Lambda := \{\mathrm{Re}(\alpha) : \alpha \in \Lambda(\mathcal{P}_T^{-1}\mathcal{A})\} \subset (0, \infty)$ for the linear fluid-structure interaction problems considered and the maximum and the minimum of R_Λ do not depend on the number of degrees of freedom, where the operator $\mathrm{Re}(\alpha)$ returns the real part of a complex number $\alpha \in \mathbb{C}$. Unfortunately, so far it is not clear how to derive a useful bound for R_Λ away from 0 and from above. For a discretization, the quantities

$$\Lambda_{\max} := \max\{|\alpha| : \alpha \in \Lambda(\mathcal{P}_T^{-1}\mathcal{A})\} \quad \text{and} \quad \Lambda_{\min} := \min\{|\alpha| : \alpha \in \Lambda(\mathcal{P}_T^{-1}\mathcal{A})\} \tag{11}$$

can therefore be computed using the representation (10) of $\Lambda(\mathcal{P}_T^{-1}\mathcal{A})$ for a small number of degrees of freedom. Since we are using the mean-based preconditioner, the elements in $\Lambda(\mathcal{P}_T^{-1}\mathcal{A})$ lie symmetrically around $x = 1$ in the complex plane (compare (9)). Consider the ellipse with center

$$d := \frac{\Lambda_{\min} + \Lambda_{\max}}{2} \quad \text{and foci} \quad d \pm c \quad \text{for} \quad c := \Lambda_{\max} - d.$$

The imaginary parts of the the elements in $\Lambda(\mathcal{P}_T^{-1}\mathcal{A})$ are so small such that this ellipse encircles all eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$. Moreover, it does not touch the imaginary axis since $\Lambda_{\min} > 0$. The Chebyshev method from [3] can therefore be generalized in the same manner as the GMRES method in Section 4.2 and used to find a low-rank approximation $\hat{X}$ of X in (8). The resulting truncated Chebyshev variant ChebyshevT is coded in (3).

5 | TIME DISCRETIZATION

5.1 | The linear fluid-structure interaction problem

Let $[0, T]$ be a time interval for $T \in \mathbb{R}^+$ and $t \in [0, T]$ be the time variable. The deformation u and the velocity v now depend, in addition, on the time variable t so we write $u(t, x)$ and $v(t, x)$. With the solid density $\rho_s \in \mathbb{R}$, the non stationary

Navier–Lamé equations discussed in Section 2.3.1.2 of [6] fulfill

$$\rho_s \partial_{tt} u - \operatorname{div}(\sigma) = \rho_s \partial_t v - \operatorname{div}(\sigma) = 0, \quad \partial_t u = v.$$

The time term $\rho_f \partial_t v$ coming from the Stokes fluid equations as mentioned in (2.42) of [6] is added to the left side of the momentum equation. The weak formulation of the non-stationary coupled linear fluid-structure interaction problem is given by

$$\begin{aligned} \langle \nabla \cdot v, \xi \rangle_F &= 0, \\ \rho_f \langle \partial_t v, \varphi \rangle_F + \rho_s \langle \partial_t v, \varphi \rangle_S + \mu_s \langle \nabla u + \nabla u^T, \nabla \varphi \rangle_S + \lambda_s \langle \operatorname{tr}(\nabla u) I, \nabla \varphi \rangle_S + \nu_f \rho_f \langle \nabla v + \nabla v^T, \nabla \varphi \rangle_F - \langle p, \nabla \cdot \varphi \rangle_F &= 0, \\ \langle \nabla u, \nabla \psi \rangle_F &= 0 \end{aligned} \tag{12}$$

with regularity conditions $v \in L^2([0,T]; v_{\text{in}} + H_0^1(\Omega, \Gamma_f^D \cup \Gamma_{\text{int}})^d)$, $\partial_t v \in L^2([0,T]; H^{-1}(\Omega)^d)$ for all $(t,x) \in [0,T] \times \Omega$. We use the notation from (4).

5.2 | Time discretization with the θ-scheme

Let $A_t^f, A_t^s \in \mathbb{R}^{M \times M}$ be discretization matrices:

$$A_t^f \quad \text{discretizes} \quad \langle v, \varphi \rangle_F \qquad \text{and} \qquad A_t^s \quad \text{discretizes} \quad \rho_s \langle v, \varphi \rangle_S.$$

Now consider a discretization that splits the time interval $[0,T]$ into $s+1 \in \mathbb{N}$ equidistant time steps. Let the distance between two time steps be Δ_t. The starting time is $t_0 = 0$ and the following times are thus given by $t_i := i\Delta_t$ for $i \in \{1, \dots, s\}$. Let X^i be the approximate solution at time t_i, X^0 is given as the initial value. The given Dirichlet data b_D^i at time t_i for all $i \in \{0, \dots, s\}$ yield the time dependent right hand side

$$B^i := b_D^i \otimes (1, \dots, 1) \quad \text{for} \quad i \in \{0, \dots, s\}.$$

Consider the one-step θ-scheme explained in Section 4.1 of [6]. Using the notation from (8), at time t_i, the following equation is to be solved for X^i:

$$\underbrace{\frac{1}{\Delta_t} A_t^f X^i (\rho_f I_m + D_3) + \frac{1}{\Delta_t} A_t^s X^i + \theta F(X^i)}_{=: F^i(X^i)} = \underbrace{\frac{1}{\Delta_t} A_t^f X^{i-1} (\rho_f I_m + D_3) + \frac{1}{\Delta_t} A_t^s X^{i-1} - (1-\theta) F(X^{i-1}) + \theta B^i + (1-\theta) B^{i-1}}_{=: B^i(X^{i-1})}, \tag{13}$$

where $\theta \in [0,1]$. $F^i(\cdot)$ contains only two sum terms more than $F(\cdot)$ from (8). At time t_i, both Algorithm 1 and Algorithm 3 can be applied to the quasi stationary problem (13) with $F^i(\cdot)$ instead of $F(\cdot)$ and the right hand side $B^i(X^{i-1})$.

5.3 | Preconditioner

At all time steps, the full matrix is given by

$$\mathcal{A}^t := \frac{1}{\Delta_t} (\rho_f I_m + D_3) \otimes A_t^f + \frac{1}{\Delta_t} I_m \otimes A_t^s + \theta \big(I \otimes A_0 + D_1 \otimes A_1 + D_2 \otimes A_2 + \nu_f D_3 \otimes A_3 \big).$$

The mean-based preconditioner, similar to $\mathcal{P}_T$ from Section 3.2, is

$$\mathcal{P}_T^t := I \otimes P_T^t, \quad \text{where} \quad P_T^t := \frac{1}{\Delta_t} (\rho_f + \bar{\rho}_f) A_t^f + \frac{1}{\Delta_t} A_t^s + \theta \big(A_0 + \bar{\mu}_s A_1 + \bar{\lambda}_s A_2 + \nu_f \bar{\rho}_f A_3 \big).$$

Even though the right hand side $B^i(X^{i-1})$ changes with every time step, the system matrix does not.

6 | THEORETICAL ERROR BOUNDS

The convergence proofs of the GMRES method from Theorem 35.2 of [8] and Section 3.4 of [14] base on the fact that the residual of the lth GMRES iterate can be represented as a product of a polynomial in $\mathcal{A}$ and the initial residual since the lth GMRES iterate is a linear combination of the start vector x_0 and the generating elements of $\mathcal{K}_l$. Also, the error bound of the Chebyshev method in [15] relies on the fact that the residual of the lth Chebyshev iterate is such a product. But even if one considers Algorithm 1 and Algorithm 3 in a non preconditioned version, multiplication with the system matrix $\mathcal{A}$ is always disturbed due to the error induced by the truncation operator. The GMREST method minimizes over $\mathcal{K}_l^{\mathcal{T}}$, the truncated Krylov subspace, instead of $\mathcal{K}_l$. In Section 6.2, the basis elements of $\mathcal{K}_l^{\mathcal{T}}$ are represented explicitly taking the truncation accuracy into consideration. Let x_l be the lth GMRES iterate, $\hat{x}_l$ be the lth GMREST iterate. An upper bound of

$$\|x_l - \hat{x}_l\|_2$$

is derived from the accuracy of the basis elements of $\mathcal{K}_l^{\mathcal{T}}$. In relation to Krylov subspace methods, inaccuracies induced by matrix-vector multiplication result in so called inexact Krylov methods and have been discussed in [16]. Iterative processes that involve truncation have been discussed in a general way in [17]. For the lth Chebyshev iterate x_l and the lth ChebyshevT iterate $\hat{x}_l$, the error

$$\|x_l - \hat{x}_l\|_2$$

is bounded in the same way in Section 6.3. These bounds show how the truncation error is propagated iteratively in Algorithm 1 and Algorithm 3 if the machine precision error is neglected.

Remark 4. If $v \in \mathbb{R}^{Mm}$, $\mathcal{T}(v)$ addresses $\mathcal{T}(\text{vec}^{-1}(v))$. Thus for the ease of notation, the truncation operator $\mathcal{T}$ from Definition 4 is regarded as a map

$$\mathcal{T} : \mathcal{R}^{Mm} \to T_R$$

and for $v \in \mathbb{R}^{Mm}$, $\mathcal{T}(v)$ addresses the full representation of the tensor in vector notation, a vector in $\mathbb{R}^{Mm}$.

Definition 6 (**Truncation accuracy**). The truncation operator $\mathcal{T}$ from Definition 4 is said to have accuracy $\epsilon > 0$ if for any $x \in \mathbb{R}^{Mm}$

$$\hat{x} := \mathcal{T}(x) = x + \mathcal{E}_{\hat{x}} \quad \text{with} \quad \mathcal{E}_{\hat{x}} \in \mathbb{R}^{Mm} \quad \text{and} \quad \|\mathcal{E}_{\hat{x}}\|_2 \leq \epsilon$$

holds. $\mathcal{E}_{\hat{x}}$ is the error induced by $\mathcal{T}$ when x is truncated.

6.1 | Matrix-vector product evaluation accuracy

If a tensor is multiplied with a scalar or a matrix, there is no truncation needed since the tensor rank does not grow. But the evaluation of $F(\cdot)$ from (8) involves 4 sum terms. After an evaluation of $F(\cdot)$ with a tensor as argument, the result has to be truncated. To keep complexity low for $\hat{X} \in T_R$, the sum $\mathcal{T}(F(\hat{X}))$, in practice, is truncated consecutively

$$\begin{aligned}
\mathcal{T}\left(F(\hat{X})\right) &\equiv \mathcal{T}\left(\mathcal{T}\left(\mathcal{T}(A_0\hat{X} + A_1\hat{X}D_1) + A_2\hat{X}D_2\right) + \nu_f A_3\hat{X}D_3\right) \\
&= \mathcal{T}\left(\mathcal{T}(A_0\hat{X} + A_1\hat{X}D_1 + A_2\hat{X}D_2 + \mathcal{E}_{\hat{F}_{s_1}}) + \nu_f A_3\hat{X}D_3\right) \\
&= \mathcal{T}\left(A_0\hat{X} + A_1\hat{X}D_1 + A_2\hat{X}D_2 + \nu_f A_3\hat{X}D_3 + \mathcal{E}_{\hat{F}_{s_1}} + \mathcal{E}_{\hat{F}_{s_2}}\right) \\
&= F(\hat{X}) + \mathcal{E}_{\hat{F}_{s_1}} + \mathcal{E}_{\hat{F}_{s_2}} + \mathcal{E}_{\hat{F}_{s_3}}.
\end{aligned}$$

$\mathcal{E}_{\hat{F}_{s_i}}$ denotes the truncation error induced by the truncation of the ith sum term for $i \in \{1,2,3\}$. By Definition 4, $\|\mathcal{E}_{\hat{F}_{s_i}}\|_2 \leq \epsilon$ for all $i \in \{1,2,3\}$. In $\mathcal{T}(F(\cdot))$ are, if the number of summands in $F(\cdot)$ is $K \in \mathbb{N}$, a total of $K-1$ truncations hidden. For a truncation accuracy of $\epsilon > 0$ we have

$$\|\mathcal{T}(F(\hat{X})) - F(\hat{X})\|_2 \leq (K-1)\epsilon.$$

Since K is a small number, usually not bigger than 4, we will neglect this detail and simply assume

$$\|\mathcal{T}(F(\hat{X})) - F(\hat{X})\|_2 \leq \epsilon$$

in the following. To make sure that the stated error bounds are still valid, the truncation accuracy would be asked to, to be exact, less than $\frac{\epsilon}{K-1}$.

6.2 | GMREST error bounds

Let x_l be the lth standard GMRES iterate, $\hat{x}_l$ be the lth GMREST iterate. How big is the difference between the truncated Krylov subspace $\mathcal{K}_l^{\mathcal{T}}$ from Section 4.2 and the Krylov subspace $\mathcal{K}_l$? First we derive explicit representations of the non-normalized basis elements of $\mathcal{K}_l^{\mathcal{T}}$. For the following Lemma, we need the truncation errors $\mathcal{E}_{K^{\mathcal{T}_k}} \in \mathbb{R}^{Mm}$ for $k \in \mathbb{N}$. They are induced by the truncation operator when the kth basis element of the truncated Krylov subspace $\mathcal{K}_l^{\mathcal{T}}$ is computed. For a truncation operator with accuracy $\epsilon > 0$, it holds $\|\mathcal{E}_{\mathcal{K}^{\mathcal{T}_k}}\|_2 \leq \epsilon$ for all $k \in \mathbb{N}$.

Lemma 1 (**Basis Representation of $\mathcal{K}_l^{\mathcal{T}}$**). *Assume* $\dim(\mathcal{K}_l^{\mathcal{T}}) = l$ *and*

$$\hat{r}_0 = \mathcal{T}\left(\mathcal{P}^{-1}(b - \mathcal{A}x_0)\right) = r_0 + \mathcal{E}_{\hat{r}_0}.$$

Let the truncation operator $\mathcal{T}(\cdot)$ have accuracy $\epsilon > 0$. The non-normalized basis elements of $\mathcal{K}_l^{\mathcal{T}}$ are given by

$$\hat{r}_0 \quad \text{and} \quad K^{\mathcal{T}_k} := (\mathcal{P}^{-1}\mathcal{A})^k r_0 + (\mathcal{P}^{-1}\mathcal{A})^k \mathcal{E}_{\hat{r}_0} + \sum_{j=1}^{k} (\mathcal{P}^{-1}\mathcal{A})^{j-1} \mathcal{E}_{K^{\mathcal{T}_{k-j+1}}} \quad \text{for all} \quad k \in \{1, \dots, l-1\}.$$

Proof (*by induction*). For $k = 1$

$$K^{\mathcal{T}_1} = \mathcal{T}\left(\mathcal{P}^{-1}F(\hat{r}_0)\right) = \mathcal{T}(\mathcal{P}^{-1}\mathcal{A}\hat{r}_0) = \mathcal{T}\left(\mathcal{P}^{-1}\mathcal{A}(r_0 + \mathcal{E}_{\hat{r}_0})\right) = \mathcal{P}^{-1}\mathcal{A}r_0 + \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{r}_0} + \mathcal{E}_{K^{\mathcal{T}_1}}$$

and "$k-1 \Rightarrow k$" since

$$\begin{aligned}
K^{\mathcal{T}_k} &= \mathcal{T}(\mathcal{P}^{-1}F)^k(\hat{r}_0) = \mathcal{T}\left(\mathcal{P}^{-1}F(K^{\mathcal{T}_{k-1}})\right) = \mathcal{T}(\mathcal{P}^{-1}\mathcal{A}K^{\mathcal{T}_{k-1}}) \\
&= \mathcal{T}\left(\mathcal{P}^{-1}\mathcal{A}\left((\mathcal{P}^{-1}\mathcal{A})^{k-1}r_0 + (\mathcal{P}^{-1}\mathcal{A})^{k-1}\mathcal{E}_{\hat{r}_0} + \sum_{j=1}^{k-1}(\mathcal{P}^{-1}\mathcal{A})^{j-1}\mathcal{E}_{K^{\mathcal{T}_{k-j}}}\right)\right) \\
&= (\mathcal{P}^{-1}\mathcal{A})^k r_0 + (\mathcal{P}^{-1}\mathcal{A})^k \mathcal{E}_{\hat{r}_0} + \sum_{j=1}^{k-1}(\mathcal{P}^{-1}\mathcal{A})^j \mathcal{E}_{K^{\mathcal{T}_{k-j}}} + \mathcal{E}_{K^{\mathcal{T}_k}} \\
&= (\mathcal{P}^{-1}\mathcal{A})^k r_0 + (\mathcal{P}^{-1}\mathcal{A})^k \mathcal{E}_{\hat{r}_0} + \sum_{j=1}^{k}(\mathcal{P}^{-1}\mathcal{A})^{j-1}\mathcal{E}_{K^{\mathcal{T}_{k-j+1}}}.
\end{aligned}$$

□

Remark 5 (*Truncation Error of $\hat{r}_0$*). Consider the line

$$\textit{Find } \hat{R} \textit{ such that } P\hat{R} = \mathcal{T}\left(\hat{B} - F(\hat{X})\right)$$

of Algorithm 1. In vector notation,

$$\mathcal{P}\hat{r}_0 = \mathcal{T}(b - F(\hat{x}_0)) \tag{14}$$

is solved for $\hat{r}_0$. Usually the initial vector x_0 is chosen such that it can be represented by a tensor of low rank. So we assume $\hat{x}_0 = \mathcal{T}(x_0) = x_0$. If the linear system (14) is solved before truncation we have

$$\|\mathcal{E}_{\hat{r}_0}\|_2 \leq \epsilon.$$

But this is rarely implemented this way. In practice, the right-hand side of (14) is truncated before the linear system is solved for $\hat{r}_0$. In this case,

$$\|\mathcal{E}_{\hat{r}_0}\|_2 \leq \epsilon\|\mathcal{P}^{-1}\|_2$$

holds. The following statements refer to the latter, more practical case. The error bounds that result in the first case can be found in Appendix A.1.

Remark 6 (*Truncation of $\hat{W}$ and Orthogonality*). Consider the line

$$\hat{W} := \mathcal{T}(\hat{W} - H_{k,i}\hat{V}_k)$$

in Algorithm 1. In the lemma above, this truncation is neglected. When the kth basis element $\hat{V}_k$ is set up, there are k extra additions involved due to this line. Let $\mathcal{E}_{\hat{W}}$ be the truncation error that occurs when this line is executed. For the sake of readability, we neglect that they differ from loop iteration to loop iteration. As a consequence, we do not add another index to $\mathcal{E}_{\hat{W}}$. The basis elements are then given by

$$\hat{r}_0 \quad \text{and} \quad K^{\mathcal{T}_k} := (\mathcal{P}^{-1}\mathcal{A})^k r_0 + (\mathcal{P}^{-1}\mathcal{A})^k \mathcal{E}_{\hat{r}_0} + \sum_{j=1}^{k} (\mathcal{P}^{-1}\mathcal{A})^{j-1} \mathcal{E}_{K^{\mathcal{T}_{k-j+1}}} + k\mathcal{E}_{\hat{W}} \quad \text{for} \quad k \in \{1, \dots, l-1\}.$$

Furthermore, we neglect round-off errors incurred from finite precision arithmetic as these are assumed to be much smaller than the truncation errors. The reason why the basis elements of $\mathcal{K}_l$ and $\mathcal{K}_l^{\mathcal{T}}$ obtained from the Arnoldi iteration differ from each other is the truncation error. Even though the elements

$$\{r_0, \mathcal{P}^{-1}\mathcal{A}, \dots, (\mathcal{P}^{-1}\mathcal{A})^{l-1} r_0\} \tag{15}$$

span $\mathcal{K}_l$, they are not orthogonal. But for the sake of notation, we incorporate the error made at the orthogonalization of the basis elements, in the truncated case, into $\mathcal{E}_{\hat{W}}$ and address by (15) the normalized basis elements that result from the Arnoldi iteration. In other words, we tacitly assume that the basis elements (15) of $\mathcal{K}_l$ are orthonormal, write them in the representation (15) and incorporate the error we made at orthogonalization into $\mathcal{E}_{\hat{W}}$. This is just one result of the assumption that we use exact precision.

Lemma 2 (**Error Bound for Truncated Basis Elements**). *Let $\sigma_{\mathcal{P}} := \|\mathcal{P}^{-1}\mathcal{A}\|_2$. Under the assumptions of Lemma 1, for*

$$e_k := \begin{cases} \|\hat{r}_0 - r_0\|_2 & \text{if } k = 0 \\ \|K^{\mathcal{T}_k} - (\mathcal{P}^{-1}A)^k r_0\|_2 & \text{if } k \in \{1, \dots, l-1\} \end{cases},$$

it holds that

$$e_k \leq \epsilon\left(\sum_{j=1}^{k} \sigma_{\mathcal{P}}^{j-1} + \|\mathcal{P}^{-1}\|_2 \sigma_{\mathcal{P}}^k + k\right) \qquad \text{for} \qquad k \in \{0, \dots, l-1\}.$$

Proof. For $k = 0$, we have

$$\begin{aligned}\|\hat{r}_0 - r_0\|_2 &= \|\mathcal{E}_{\hat{r}_0}\|_2 \\ &\leq \epsilon \|\mathcal{P}^{-1}\|_2 \\ &= \epsilon\left(\sum_{j=1}^{k} \sigma_{\mathcal{P}}^{j-1} + \|\mathcal{P}^{-1}\|_2 \sigma_{\mathcal{P}}^{k}\right),\end{aligned}$$

with the convention

$$\sum_{j\in\emptyset} \sigma_{\mathcal{P}}^{j-1} = 0.$$

For $k \geq 1$, we use Lemma 1.

$$\begin{aligned}e_k &= \|(\mathcal{P}^{-1}\mathcal{A})^k \mathcal{E}_{\hat{r}_0} + \sum_{j=1}^{k} (\mathcal{P}^{-1}\mathcal{A})^{j-1} \mathcal{E}_{K^{\mathcal{T}_{k-j+1}}} \|_2 \\ &= \|(\mathcal{P}^{-1}\mathcal{A})^k \mathcal{E}_{\hat{r}_0} + \mathcal{E}_{K^{\mathcal{T}_k}} + \sum_{j=1}^{k-1} (\mathcal{P}^{-1}\mathcal{A})^{j} \mathcal{E}_{K^{\mathcal{T}_{k-j}}} \|_2 \\ &\leq \sigma_{\mathcal{P}}^{k} \epsilon \|\mathcal{P}^{-1}\|_2 + \epsilon + \sum_{j=1}^{k-1} \sigma_{\mathcal{P}}^{j} \epsilon \\ &= \epsilon\left(\sum_{j=1}^{k} \sigma_{\mathcal{P}}^{j-1} + \|\mathcal{P}^{-1}\|_2 \sigma_{\mathcal{P}}^{k}\right).\end{aligned}$$

The truncation error coming from the orthogonalization process mentioned in Remark 6 adds the term ϵk to the error bound. □

The standard GMRES minimizes over the Krylov subspace $\mathcal{K}_l$. In terms of Remark 6, the standard GMRES method finds coefficients $c_i \in \mathbb{R}$ for $i \in \{1, \dots, l\}$ such that

$$x_l = x_0 + c_1 r_0 + c_2 \mathcal{P}^{-1}\mathcal{A} r_0 + \cdots + c_l (\mathcal{P}^{-1}\mathcal{A})^{l-1} r_0.$$

In the same way we can write

$$\hat{x}_l = \hat{x}_0 + d_1 \hat{r}_0 + d_2 K^{\mathcal{T}_1} + \cdots + d_l K^{\mathcal{T}_{l-1}},$$

where the coefficients d_i for $i \in \{1, \dots, l\}$ refer to the coefficients found by the Arnoldi iteration in the GMREST method. This allows to state the following theorem.

Theorem 1 (**Approximation Error of GMREST**). *Let x_l be the lth iterate of the standard GMRES method, $\hat{x}_l$ be the lth iterate of the GMREST method. It holds*

$$\|\hat{x}_l - x_l\|_2 \leq \epsilon \sum_{j=1}^{l} |d_j| \left(\sum_{i=1}^{j-1} \sigma_{\mathcal{P}}^{i-1} + \|\mathcal{P}^{-1}\|_2 \sigma_{\mathcal{P}}^{j-1} + j - 1\right) + \sum_{j=1}^{l} |c_j - d_j| + \epsilon l.$$

Proof.

$$\begin{aligned}\|\hat{x}_l - x_l\|_2 &= \|\hat{x}_0 - x_0 + d_1 \hat{r}_0 - c_1 r_0 + d_2 K^{\mathcal{T}_1} - c_2 \mathcal{P}^{-1}\mathcal{A} r_0 + \cdots + d_l K^{\mathcal{T}_{l-1}} - c_l (\mathcal{P}^{-1}\mathcal{A})^{l-1} r_0 \|_2 \\ &\leq |d_1| e_0 + |d_2| e_1 + \cdots + |d_l| e_{l-1} + |c_1 - d_1| \| \underbrace{r_0}_{[0pt](\star)} \|_2\end{aligned}$$

$$+ |c_2 - d_2| \| \underbrace{\mathcal{P}^{-1}\mathcal{A}r_0}_{(\star)} \|_2 + \cdots + |c_l - d_l| \| \underbrace{(\mathcal{P}^{-1}\mathcal{A})^{l-1} r_0}_{(\star)} \|_2 = (*)$$

We assume that the standard GMRES method does an accurate orthogonalization of the Krylov subspace $\mathcal{K}_l$ (see Remark 6). By the elements $(\star)$ we address the orthonormal basis elements of $\mathcal{K}_l$. They all have an Euclidean norm of 1. Therefore,

$$\begin{aligned}(*) &= \sum_{j=1}^{l} \left(|d_j| e_{j-1} + |c_j - d_j| \right) \\ &\leq \epsilon \sum_{j=1}^{l} |d_j| \left(\sum_{i=1}^{j-1} \sigma_{\mathcal{P}}^{i-1} + \|\mathcal{P}^{-1}\|_2 \sigma_{\mathcal{P}}^{j-1} + j - 1 \right) + \sum_{j=1}^{l} |c_j - d_j|\end{aligned}$$

holds. The additional sum term ϵl comes from the last successive sum in the method where the approximation is built. □

6.3 | ChebyshevT error bounds

Similar to Section 6.2, we derive an error bound for the ChebyshevT method coded in Algorithm 3. Let x_l denote the lth iterate of the standard Chebyshev method and $\hat{x}_l$ denote the lth iterate of the ChebyshevT method.

Remark 7. The ith residual is given by the solution r_i to

$$\mathcal{P} r_i = b - \mathcal{A} x_i.$$

The truncation of r_i yields

$$\hat{r}_i = \mathcal{T}(r_i) = r_i + \mathcal{E}_{\hat{r}_i}, \quad \text{with} \quad \|\mathcal{E}_{\hat{r}_i}\|_2 \leq \epsilon$$

if the truncation operator $\mathcal{T}$ is assumed to have accuracy ϵ. In analogy to Remark 5, the two cases $\|\mathcal{E}_{\hat{r}_i}\|_2 \leq \epsilon$ and $\|\mathcal{E}_{\hat{r}_i}\|_2 \leq \epsilon \|\mathcal{P}^{-1}\|_2$ have to be distinguished. In this section, we consider the latter case. The error bounds of Theorem 2 for the case $\|\mathcal{E}_{\hat{r}_i}\| \leq \epsilon$ can be found in Appendix A.2.

The start vector and the right hand side are assumed to be of low rank, namely

$$\hat{x}_0 = \mathcal{T}(x_0) = x_0 \quad \text{and} \quad \hat{b} = \mathcal{T}(b) = b.$$

In the same way as in Section 6.2, the norm $\|\hat{x}_l - x_l\|_2$ is to be estimated. $\mathcal{E}_{\hat{x}_l^a}$ denotes the total error

$$\mathcal{E}_{\hat{x}_l^a} := \hat{x}_l - x_l,$$

not to be confused with $\mathcal{E}_{\hat{x}_l}$, the truncation error with norm ϵ that occurs when truncating $\hat{x}_l$. The iterative Chebyshev method is a three term recursion. Thus, the Chebyshev iterates itself can be represented by a recursive formula.

Lemma 3 (**Representation of the ChebyshevT Iterates**). *Let the scalars*

$$\alpha_i, \beta_i \in \mathbb{R} \quad \textit{for} \quad i \in \{1, \dots, l\} \quad \textit{and} \quad \hat{\Phi}_i \in T_R \quad \textit{for} \quad i \in \{0, \dots, l\}$$

be given as defined in Algorithm 3. Φ_i denote the non truncated full matrices corresponding to $\hat{\Phi}_i$ if Algorithm 3 is applied and any truncation is neglected. If

$$\hat{r}_0 = r_0 + \mathcal{E}_{\hat{r}_0},$$

it holds that

$$\mathcal{E}_{\hat{x}_0^a} = 0,$$

$$\hat{x}_1 = x_1 + \underbrace{\frac{1}{d}\mathcal{E}_{\hat{r}_0} + \mathcal{E}_{\hat{x}_1}}_{\mathcal{E}_{\hat{x}_1^a}},$$

$$\hat{x}_2 = x_2 + \underbrace{\mathcal{E}_{\hat{\Phi}_1} + \mathcal{E}_{\hat{x}_1^a} + \mathcal{E}_{\hat{x}_2} + \alpha_1(\mathcal{E}_{\hat{r}_1} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_1^a}) + \frac{\beta_1}{d}\mathcal{E}_{\hat{r}_0}}_{\mathcal{E}_{\hat{x}_2^a}} \quad \textit{and}$$

$$\hat{x}_l = x_l + \mathcal{E}_{\hat{\Phi}_{l-1}} + \mathcal{E}_{\hat{x}_{l-1}^a} + \mathcal{E}_{\hat{x}_l} + \sum_{j=1}^{l-1}\alpha_j\left(\prod_{i=1}^{l-j-1}\beta_{i+j}\right)(\mathcal{E}_{\hat{r}_j} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_j^a}) + \left(\prod_{j=1}^{l-1}\beta_j\right)\frac{1}{d}\mathcal{E}_{\hat{r}_0} + \sum_{j=1}^{l-2}\left(\prod_{i=1}^{l-j-1}\beta_{i+j}\right)\mathcal{E}_{\hat{\Phi}_j} \quad \textit{for} \quad l \geq 3,$$

where $\mathcal{E}_{\hat{x}_j^a} := \hat{x}_j - x_j$ *for* $j \in \{0, \dots, l\}$. *We use the convention*

$$\prod_{j\in\emptyset}\beta_j = 1.$$

If a truncation operator of accuracy $\epsilon > 0$ *is used, then certainly* $\|\mathcal{E}_{\hat{x}_i}\|_2 \leq \epsilon$ *but not necessarily* $\|\mathcal{E}_{\hat{x}_i^a}\|_2 \leq \epsilon$ *holds for* $i \in \{0, \dots, l\}$. *The error induced by the truncation operator that truncates* $\hat{\Phi}_i$ *is denoted by* $\mathcal{E}_{\hat{\Phi}_i}$ *for* $i \in \{0, \dots, l\}$.

Proof. $l = 1$:
Provided that $\hat{x}_0 = x_0 = x_0 + \mathcal{E}_{\hat{x}_0^a} \Rightarrow \mathcal{E}_{\hat{x}_0^a} = 0$.

$$\hat{x}_1 = \mathcal{T}(\hat{x}_0 + \frac{1}{d}\hat{r}_0) = \mathcal{T}\left(x_0 + \frac{1}{d}r_0 + \frac{1}{d}\mathcal{E}_{\hat{r}_0}\right) = \underbrace{x_0 + \frac{1}{d}r_0}_{=x_1} + \underbrace{\frac{1}{d}\mathcal{E}_{\hat{r}_0} + \mathcal{E}_{\hat{x}_1}}_{=\mathcal{E}_{\hat{x}_1^a}}$$

$l = 2$:

$$\hat{x}_2 = \mathcal{T}(\hat{x}_1 + \hat{\Phi}_1) = \mathcal{T}\left(\hat{x}_1 + \mathcal{T}(\alpha_1\hat{r}_1 + \beta_1\hat{\Phi}_0)\right) = \mathcal{T}\left(\hat{x}_1 + \mathcal{T}(\alpha_1\hat{r}_1 + \frac{\beta_1}{d}\hat{r}_0)\right)$$

$$= \mathcal{T}\left(x_1 + \mathcal{E}_{\hat{x}_1^a} + \mathcal{T}\left(\alpha_1(r_1 + \mathcal{E}_{\hat{r}_1} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_1^a}) + \frac{\beta_1}{d}(r_0 + \mathcal{E}_{\hat{r}_0})\right)\right) = (\star)$$

since

$$\hat{r}_1 = \mathcal{T}\left(\mathcal{P}^{-1}(b - \mathcal{A}\hat{x}_1)\right) = \mathcal{T}\left(\mathcal{P}^{-1}(b - \mathcal{A}x_1 - \mathcal{A}\mathcal{E}_{\hat{x}_1^a})\right) = r_1 - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_1^a} + \mathcal{E}_{\hat{r}_1}.$$

Thus,

$$(\star) = \mathcal{T}\left(\underbrace{x_1 + \alpha_1 r_1 + \frac{\beta_1}{d}r_0}_{=x_2} + \alpha_1(\mathcal{E}_{\hat{r}_1} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_1^a}) + \frac{\beta_1}{d}\mathcal{E}_{\hat{r}_0} + \mathcal{E}_{\hat{x}_1^a} + \mathcal{E}_{\hat{\Phi}_1}\right)$$

$$= x_2 + \underbrace{\alpha_1(\mathcal{E}_{\hat{r}_1} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_1^a}) + \frac{\beta_1}{d}\mathcal{E}_{\hat{r}_0} + \mathcal{E}_{\hat{x}_1^a} + \mathcal{E}_{\hat{\Phi}_1} + \mathcal{E}_{\hat{x}_2}}_{=\mathcal{E}_{\hat{x}_2^a}}.$$

For the proof for $l \geq 3$, we go by induction. For the initial step $l = 3$, we need

$$\hat{\Phi}_0 = \frac{1}{d}\hat{r}_0 = \frac{1}{d}(r_0 + \mathcal{E}_{\hat{r}_0}) = \Phi_0 + \frac{1}{d}\mathcal{E}_{\hat{r}_0},$$

$$\hat{\Phi}_1 = \mathcal{T}(\alpha_1\hat{r}_1 + \beta_1\hat{\Phi}_0) = \mathcal{T}\left(\alpha_1 r_1 + \beta_1\Phi_0 + \alpha_1(\mathcal{E}_{\hat{r}_1} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_1^a}) + \frac{\beta_1}{d}\mathcal{E}_{\hat{r}_0}\right)$$

$$= \Phi_1 + \alpha_1(\mathcal{E}_{\hat{r}_1} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_1^a}) + \frac{\beta_1}{d}\mathcal{E}_{\hat{r}_0} + \mathcal{E}_{\hat{\Phi}_1}$$

and

$$\hat{\Phi}_2 = \mathcal{T}(\alpha_2\hat{r}_2 + \beta_2\hat{\Phi}_1) = \alpha_2 r_2 + \beta_2\Phi_1 + \alpha_2(\mathcal{E}_{\hat{r}_2} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_2^a}) + \alpha_1\beta_2(\mathcal{E}_{\hat{r}_1} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_1^a}) + \frac{\beta_1\beta_2}{d}\mathcal{E}_{\hat{r}_0} + \beta_2\mathcal{E}_{\hat{\Phi}_1} + \mathcal{E}_{\hat{\Phi}_2}$$

$$= \Phi_2 + \alpha_2(\mathcal{E}_{\hat{r}_2} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_2^a}) + \alpha_1\beta_2(\mathcal{E}_{\hat{r}_1} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_1^a}) + \frac{\beta_1\beta_2}{d}\mathcal{E}_{\hat{r}_0} + \beta_2\mathcal{E}_{\hat{\Phi}_1} + \mathcal{E}_{\hat{\Phi}_2}. \tag{16}$$

Therefore,

$$\hat{x}_3 = \mathcal{T}(\hat{x}_2 + \hat{\Phi}_2)$$

$$= \mathcal{T}\left(x_2 + \Phi_2 + \mathcal{E}_{\hat{x}_2^a} + \alpha_2(\mathcal{E}_{\hat{r}_2} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_2^a}) + \alpha_1\beta_2(\mathcal{E}_{\hat{r}_1} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_1^a}) + \frac{\beta_1\beta_2}{d}\mathcal{E}_{\hat{r}_0} + \beta_2\mathcal{E}_{\hat{\Phi}_1} + \mathcal{E}_{\hat{\Phi}_2}\right)$$

$$= x_3 + \mathcal{E}_{\hat{\Phi}_2} + \mathcal{E}_{\hat{x}_2^a} + \mathcal{E}_{\hat{x}_3} + \alpha_1\beta_2(\mathcal{E}_{\hat{r}_1} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_1^a}) + \alpha_2(\mathcal{E}_{\hat{r}_2} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_2^a}) + \frac{\beta_1\beta_2}{d}\mathcal{E}_{\hat{r}_0} + \beta_2\mathcal{E}_{\hat{\Phi}_1}$$

$$= x_3 + \mathcal{E}_{\hat{\Phi}_2} + \mathcal{E}_{\hat{x}_2^a} + \mathcal{E}_{\hat{x}_3} + \sum_{j=1}^{2}\alpha_j\left(\prod_{i=1}^{3-j-1}\beta_{i+j}\right)(\mathcal{E}_{\hat{r}_j} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_j^a}) + \left(\prod_{j=1}^{2}\beta_j\right)\frac{1}{d}\mathcal{E}_{\hat{r}_0} + \beta_2\mathcal{E}_{\hat{\Phi}_1}.$$

To conclude $l-1 \to l$ we first prove that

$$\hat{\Phi}_{l-1} = \Phi_{l-1} + \mathcal{E}_{\hat{\Phi}_{l-1}} + \sum_{j=1}^{l-1}\alpha_j\left(\prod_{i=1}^{l-j-1}\beta_{i+j}\right)(\mathcal{E}_{\hat{r}_j} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_j^a}) + \left(\prod_{j=1}^{l-1}\beta_j\right)\frac{1}{d}\mathcal{E}_{\hat{r}_0} + \sum_{j=1}^{l-2}\left(\prod_{i=1}^{l-j-1}\beta_{i+j}\right)\mathcal{E}_{\hat{\Phi}_j} \tag{17}$$

under the assumption that this equation holds for $\hat{\Phi}_{l-2}$. For $\hat{\Phi}_2$, this is true since from (16), we have that

$$\hat{\Phi}_2 = \Phi_2 + \mathcal{E}_{\hat{\Phi}_2} + \sum_{j=1}^{2}\alpha_j\left(\prod_{i=1}^{2-j}\beta_{i+j}\right)(\mathcal{E}_{\hat{r}_j} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_j^a}) + \left(\prod_{j=1}^{2}\beta_j\right)\frac{1}{d}\mathcal{E}_{\hat{r}_0} + \beta_2\mathcal{E}_{\hat{\Phi}_1}.$$

The induction step for (17) is as follows.

$$\hat{\Phi}_{l-1} = \mathcal{T}(\alpha_{l-1}\hat{r}_{l-1} + \beta_{l-1}\hat{\Phi}_{l-2})$$

$$= \alpha_{l-1}r_{l-1} + \beta_{l-1}\Phi_{l-2} + \mathcal{E}_{\hat{\Phi}_{l-1}} + \alpha_{l-1}(\mathcal{E}_{\hat{r}_{l-1}} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_{l-1}^a}) + \beta_{l-1}\mathcal{E}_{\hat{\Phi}_{l-2}} + \beta_{l-1}\sum_{j=1}^{l-2}\alpha_j\left(\prod_{i=1}^{l-j-2}\beta_{i+j}\right)(\mathcal{E}_{\hat{r}_j} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_j^a})$$

$$+ \beta_{l-1}\left(\prod_{j=1}^{l-2}\beta_j\right)\frac{1}{d}\mathcal{E}_{\hat{r}_0} + \beta_{l-1}\sum_{j=1}^{l-3}\left(\prod_{i=1}^{l-j-2}\beta_{i+j}\right)\mathcal{E}_{\hat{\Phi}_j}$$

$$= \Phi_{l-1} + \mathcal{E}_{\hat{\Phi}_{l-1}} + \sum_{j=1}^{l-1}\alpha_j\left(\prod_{i=1}^{l-j-1}\beta_{i+j}\right)(\mathcal{E}_{\hat{r}_j} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}_j^a}) + \left(\prod_{j=1}^{l-1}\beta_j\right)\frac{1}{d}\mathcal{E}_{\hat{r}_0} + \sum_{j=1}^{l-2}\left(\prod_{i=1}^{l-j-1}\beta_{i+j}\right)\mathcal{E}_{\hat{\Phi}_j}$$

With this, it follows that

$$\begin{aligned}\hat{x}_l &= \mathcal{T}(\hat{x}_{l-1} + \hat{\Phi}_{l-1}) \\ &= x_l + \mathcal{E}_{\hat{\Phi}_{l-1}} + \mathcal{E}_{\hat{x}^a_{l-1}} + \mathcal{E}_{\hat{x}_l} + \sum_{j=1}^{l-1} \alpha_j \left(\prod_{i=1}^{l-j-1} \beta_{i+j} \right) (\mathcal{E}_{\hat{r}_j} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}^a_j}) + \left(\prod_{j=1}^{l-1} \beta_j \right) \frac{1}{d} \mathcal{E}_{\hat{r}_0} + \sum_{j=1}^{l-2} \left(\prod_{i=1}^{l-j-1} \beta_{i+j} \right) \mathcal{E}_{\hat{\Phi}_j}.\end{aligned}$$

□

Theorem 2 (**Approximation Error of ChebyshevT**). *Let $\sigma_{\mathcal{P}} := \|\mathcal{P}^{-1}\mathcal{A}\|_2$. Under the assumptions of Lemma 3, the following error bounds hold for a truncation operator of accuracy $\epsilon > 0$.*

$$\begin{aligned} e_1 &:= \|\hat{x}_l - x_l\|_2 = \|\mathcal{E}_{\hat{x}^a_1}\|_2 \leq \epsilon \left(1 + \frac{1}{|d|} \|\mathcal{P}^{-1}\|_2 \right), \\ e_2 &:= \|\hat{x}_2 - x_2\|_2 \leq \epsilon \left(3 + |\alpha_1|\sigma_{\mathcal{P}} + \left(|\alpha_1| + \frac{1 + |\beta_1| + |\alpha_1|\sigma_{\mathcal{P}}}{|d|} \right) \|\mathcal{P}^{-1}\|_2 \right) \quad \textit{and} \\ e_l &:= \|\hat{x}_l - x_l\|_2 \leq (1 + |\alpha_{l-1}|\sigma_{\mathcal{P}}) e_{l-1} + \sum_{j=1}^{l-2} |\alpha_j| e_j \sigma_{\mathcal{P}} \prod_{i=1}^{l-j-1} |\beta_{i+j}| \\ &\quad + \epsilon \left(2 + \sum_{j=1}^{l-2} \prod_{i=1}^{l-j-1} |\beta_{i+j}| + \left(\sum_{j=1}^{l-1} |\alpha_j| \prod_{i=1}^{l-j-1} |\beta_{i+j}| + \frac{\prod_{j=1}^{l-1} |\beta_j|}{|d|} \right) \|\mathcal{P}^{-1}\|_2 \right) \quad \textit{for} \quad l \geq 3. \end{aligned}$$

Proof. Let $\epsilon_R > 0$ such that $\|\mathcal{E}_{\hat{r}_i}\|_2 \leq \epsilon_R$ for all $i \in \{1, \dots, l\}$.
$l = 1$:

$$e_1 = \|\mathcal{E}_{\hat{x}^a_1}\|_2 \leq \epsilon + \frac{1}{|d|} \epsilon_R$$

$l = 2$:

$$\begin{aligned} e_2 &= \|\mathcal{E}_{\hat{\Phi}_1} + \frac{1}{d}\mathcal{E}_{\hat{r}_0} + \mathcal{E}_{\hat{x}_1} + \mathcal{E}_{\hat{x}_2} + \alpha_1 \left(\mathcal{E}_{\hat{r}_1} - \mathcal{P}^{-1}\mathcal{A} \left(\frac{1}{d}\mathcal{E}_{\hat{r}_0} + \mathcal{E}_{\hat{x}_1} \right) \right) + \frac{\beta_1}{d}\mathcal{E}_{\hat{r}_0}\|_2 \\ &\leq \|\mathcal{E}_{\hat{\Phi}_1} + \mathcal{E}_{\hat{x}_1} + \mathcal{E}_{\hat{x}_2}\|_2 + |\alpha_1|\sigma_{\mathcal{P}}\|\mathcal{E}_{\hat{x}_1}\|_2 + |\alpha_1|\|\mathcal{E}_{\hat{r}_1}\|_2 + (1 + |\alpha_1|\sigma_{\mathcal{P}} + |\beta_1|)\|\frac{\mathcal{E}_{\hat{r}_0}}{d}\|_2 \\ &\leq (3 + |\alpha_1|\sigma_{\mathcal{P}})\epsilon + \left(|\alpha_1| + \frac{1 + |\beta_1| + |\alpha_1|\sigma_{\mathcal{P}}}{|d|} \right) \epsilon_R \end{aligned}$$

$l \geq 3$:

$$\begin{aligned} e_l &= \left\| \mathcal{E}_{\hat{\Phi}_{l-1}} + \mathcal{E}_{\hat{x}^a_{l-1}} + \mathcal{E}_{\hat{x}_l} + \sum_{j=1}^{l-1} \alpha_j \left(\prod_{i=1}^{l-j-1} \beta_{i+j} \right) (\mathcal{E}_{\hat{r}_j} - \mathcal{P}^{-1}\mathcal{A}\mathcal{E}_{\hat{x}^a_j}) + \left(\prod_{j=1}^{l-1} \beta_j \right) \frac{1}{d}\mathcal{E}_{\hat{r}_0} + \sum_{j=1}^{l-2} \left(\prod_{i=1}^{l-j-1} \beta_{i+j} \right) \mathcal{E}_{\hat{\Phi}_j} \right\|_2 \\ &\leq \underbrace{\|\mathcal{E}_{\hat{x}^a_{l-1}}\|_2}_{=e_{l-1}} + |\alpha_{l-1}|\sigma_{\mathcal{P}}\|\mathcal{E}_{\hat{x}^a_{l-1}}\|_2 + \sum_{j=1}^{l-2} |\alpha_j|\sigma_{\mathcal{P}}\|\mathcal{E}_{\hat{x}^a_j}\|_2 \prod_{i=1}^{l-j-1} |\beta_{i+j}| + \|\mathcal{E}_{\hat{\Phi}_{l-1}} + \mathcal{E}_{\hat{x}_l}\|_2 + \sum_{j=1}^{l-2} \|\mathcal{E}_{\hat{\Phi}_j}\|_2 \prod_{i=1}^{l-j-1} |\beta_{i+j}| \\ &\quad + \sum_{j=1}^{l-1} |\alpha_j| \|\mathcal{E}_{\hat{r}_j}\|_2 \prod_{i=1}^{l-j-1} |\beta_{i+j}| + \frac{\prod_{j=1}^{l-1} |\beta_j|}{|d|} \|\mathcal{E}_{\hat{r}_0}\|_2 \end{aligned}$$

$$\leq (1+|\alpha_{l-1}|\sigma_{\mathcal{P}})e_{l-1} + \sum_{j=1}^{l-2}|\alpha_j|e_j\sigma_{\mathcal{P}}\prod_{i=1}^{l-j-1}|\beta_{i+j}| + \left(2+\sum_{j=1}^{l-2}\prod_{i=1}^{l-j-1}|\beta_{i+j}|\right)\epsilon + \left(\sum_{j=1}^{l-1}|\alpha_j|\prod_{i=1}^{l-j-1}|\beta_{i+j}| + \frac{\prod_{j=1}^{l-1}|\beta_j|}{|d|}\right)\epsilon_R$$

The estimation $\epsilon_R \leq \epsilon\|\mathcal{P}^{-1}\|_2$ leads to the claimed error bounds. □

7 | NUMERICAL EVALUATION OF THE ERROR BOUNDS

In algorithm and software implementations, the accuracy of a truncation operator depends on the truncation rank. If one chooses a rank R, the iterate of the GMREST or the ChebyshevT method is truncated to, the accuracy of the truncation operator is still unknown. Most truncation techniques like the HOSVD for hierarchical Tucker tensors (Section 8.3 and Section 10.1.1 of [9]) or the TT-rounding for TT tensors (Algorithm 1 and 2 of [12]) provide quasi optimality for tensors of order $d > 2$. For tensors of order $d = 2$ they even provide optimality in the sense that the result of the truncation of a matrix to rank R is indeed the best rank R approximation of the matrix. Nonetheless, since the singular value decay of the argument to be truncated is, in general, not known, the truncation operator will be simulated for a numerical evaluation of the error bounds of Theorem 1 and Theorem 2. Using the MATLAB routine `rand()`, a vector

$$\tilde{z} \in \mathbb{R}^{Mm}$$

with entries that are uniformly distributed in the interval (0,1) is constructed first. The argument $x \in \mathbb{R}^{Mm}$ is then truncated using the truncation simulator

$$\mathcal{T}_s(x) := x + \frac{\epsilon}{\|\tilde{z}\|_2}\tilde{z}. \tag{18}$$

Of course, $\tilde{z}$ is computed anew every time $\mathcal{T}_s(\cdot)$ is applied. For this subsection, all the computations are therefore made in the full format and whenever a truncation operator is applied, the truncation simulator $\mathcal{T}_s(\cdot)$ is evaluated. The main advantage of this strategy is that

$$\|\mathcal{T}_s(x) - x\|_2 = \epsilon \; \forall x \in \mathbb{R}^{Mm}.$$

A truncation operator based on the singular value decomposition does not provide such a reliable behavior. Let

$$\{\sigma_i\}_{i\in\{1,\dots,\min\{M,m\}\}},\ \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min\{M,m\}}$$

be the singular values of $\mathrm{vec}^{-1}(x)$ and $\exists k \in \{1,\dots,\min\{M,m\}-1\}$ such that, e.g.,

$$\sigma_k = 10^{-4} \quad \text{and} \quad \sigma_{k+1} = 10^{-10}.$$

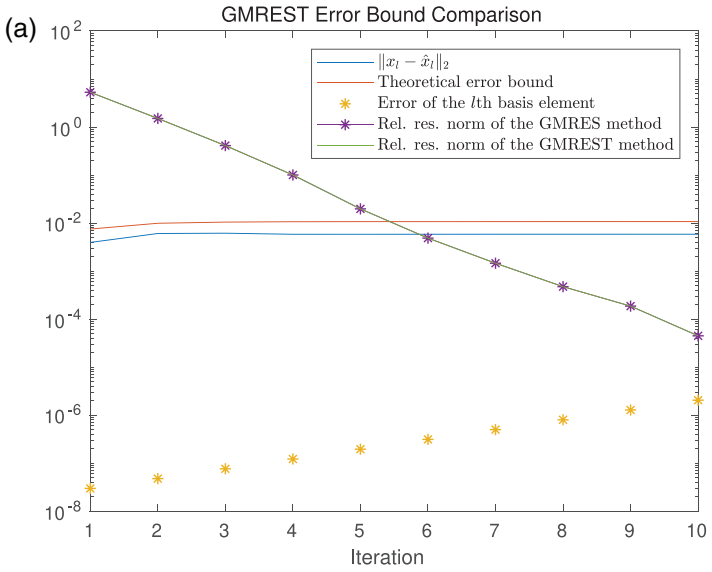
A truncation operator with accuracy $\epsilon = 10^{-5}$ based on the singular value decomposition would provide an approximation of x with accuracy 10^{-10} in this example. So in this sense, the truncation simulator $\mathcal{T}_s$ yields the worst case error every time it is applied.

7.1 | GMREST error Bound

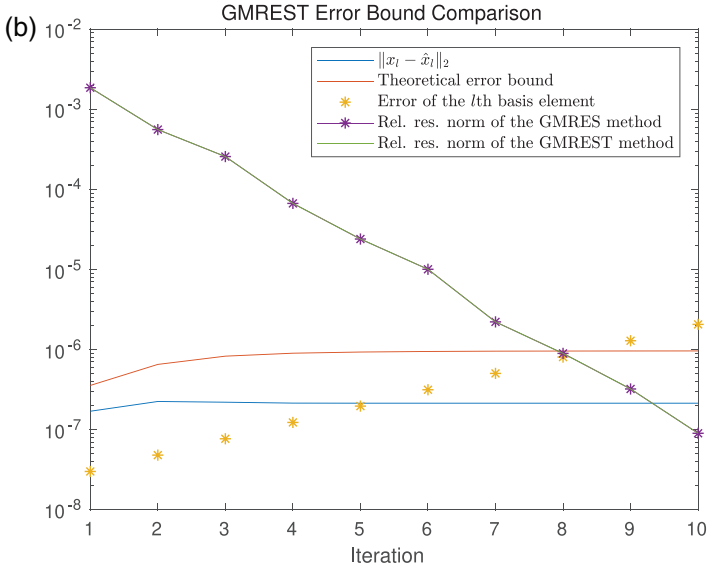
We consider the error bound from Theorem 1 that reads

$$\|\hat{x}_l - x_l\|_2 \leq \epsilon\sum_{j=1}^{l}|d_j|\left(\sum_{i=1}^{j-1}\sigma_{\mathcal{P}}^{i-1} + \|\mathcal{P}^{-1}\|_2\sigma_{\mathcal{P}}^{j-1} + j - 1\right) + \sum_{j=1}^{l}|c_j - d_j| + \epsilon l.$$

FIGURE 1 A numerical evaluation of the theoretical GMREST error bound



$\epsilon = 10^{-12}$. All entries of the start matrix are set to 1.



$\epsilon = 10^{-12}$. $\hat{x}_6$ is used as start matrix.

This theoretical error bound is compared with

$$\|x_l - \hat{x}_l\|_2, \tag{19}$$

where x_l denotes the lth GMRES iterate and $\hat{x}_l$ the lth GMREST iterate. As just explained, everything is computed in the full format and every time a truncation is involved (which affects the GMREST iterate $\hat{x}_l$ only), $\mathcal{T}_s$ from (18) is evaluated. The 3d jetty from Section 8.1 is considered with size $M = 4095$ and a three parameter discretization with a total of $m = 8000$ parameter combinations as used in Section 8.3. We use the estimate $\sigma_{\mathcal{P}} \approx d + c$ with c, d from Section 8.5. In addition, the basis element error bound from Lemma 2 is plotted for a truncation accuracy of $\epsilon = 10^{-12}$. If one starts with a matrix whose entries are all set to 1, the error bound (19) states that $\|x_{10} - \hat{x}_{10}\|_2$ is not bigger than $\approx 10^{-2}$, which can be seen in Figure 1a. The reason for such a tolerant bound is that the first coefficients $d_1, d_2, \ldots$ are very big if the initial guess is bad. But if both methods are restarted with $\hat{x}_6$ as start matrix, these coefficients become smaller as shown in Figure 1b. Also, the relative residual norm of the GMRES iterate, $\frac{\|B - F(x)\|_F}{\|B\|_F}$, and the one of the GMREST iterate, $\frac{\|B - F(\hat{x})\|_F}{\|B\|_F}$, are plotted. So even though $\|x_l - \hat{x}_l\|_2$ stagnates, the residual of the truncated approach still decreases.

The dominating terms are

$$\epsilon \sum_{j=1}^{l} |d_j| \|\mathcal{P}^{-1}\|_2 \sigma_{\mathcal{P}}^{j-1} \quad \text{and} \quad \epsilon \sum_{j=1}^{l} |d_j| \sum_{i=1}^{j-1} \sigma_{\mathcal{P}}^{i-1}.$$

As pointed out above, $d_1, d_2, \ldots$ are big for a bad initial guess. Then, in addition, ϵ can not compensate the (exponential) growth of $\sigma_{\mathcal{P}}^{j-1}$ for $j \in \{1, \ldots, l\}$. Notice that $\sigma_{\mathcal{P}} \approx 1.6$ in this example. Since $\|\mathcal{P}^{-1}\|_2$ is rather big, namely $\approx 3 \cdot 10^4$ in this example, the former of these two terms is bigger for the first 10 iterations. Furthermore, $1.6^{10} \approx 10^2$ and the moduli of the coefficients d_j become smaller the bigger the iteration count is. This is why we do not see an exponential growth of the bound in Figure 1a,b.

7.2 | ChebyshevT error bound

In this subsection, the approximation error from Theorem 2 is numerically examined. Let x_l be the lth Chebyshev iterate and $\hat{x}_l$ be the lth ChebyshevT iterate. For the ChebyshevT method, similar to the GMRES comparison, $\mathcal{T}_s$ is used as truncation operator.

Remark 8. If preconditioned with $\mathcal{P}_T$, in theory,

$$\epsilon_R \leq \epsilon \|\mathcal{P}_T^{-1}\|_2$$

holds as mentioned in Remark 7. But, due to the bad condition of the preconditioner and machine precision, in practice, ϵ_R is often bigger. The line

$$\textit{Find } \hat{R}_i \textit{ such that } P\hat{R}_i = \mathcal{T}\left(\hat{B} - F(\hat{X})\right).$$

is executed in every iteration of Algorithm 3 which sometimes leads to real errors that are higher than the error bound from Theorem 2. In contrast to this, the line

$$\textit{Find } \hat{W} \in T_R \textit{ such that } P\hat{W} = \mathcal{T}\left(F(\hat{V}_i)\right).$$

in Algorithm 1 is less vulnerable. To circumvent this problem, for the error bound of Theorem 2, the error ϵ_R is computed explicitly. This value instead of $\epsilon\|\mathcal{P}_T^{-1}\|_2$ is then used to compute the theoretical error bounds that are compared with the real errors.

We use the same configuration as used in Section 7.1. Even though the theoretical error bound literally explodes, for $\epsilon = 10^{-12}$, the truncated method converges roughly as good as the non truncated method until iteration 10 for the 3d jetty model from Section 8.1 as shown in Figure 2a. But the convergence of the ChebyshevT method deteriorates remarkably after 4 iterations for $\epsilon = 10^{-6}$ if compared to the full approach (see Figure 2b).

The two terms in the error bound that are not multiplied with ϵ are

$$(1 + |\alpha_{l-1}|\sigma_{\mathcal{P}})e_{l-1} \quad \text{and} \tag{20}$$

$$\sum_{j=1}^{l-2} |\alpha_j| e_j \sigma_{\mathcal{P}} \overbrace{\prod_{i=1}^{l-j-1} |\beta_{i+j}|}^{(\star)}. \tag{21}$$

The coefficients β_i in the Chebyshev method have norms that are smaller than 1. This becomes clear if one considers the recursive computation formula for the Chebyshev polynomials (see (2.4) in [3]) evaluated at $\frac{d}{c}$ with $|c| < |d|$. The coefficients β_i are then given as a fraction where the numerator has a norm that is smaller than the denominator. The product $(\star)$ becomes smaller the higher the iteration number is and therefore, the term (21) becomes negligibly small, at

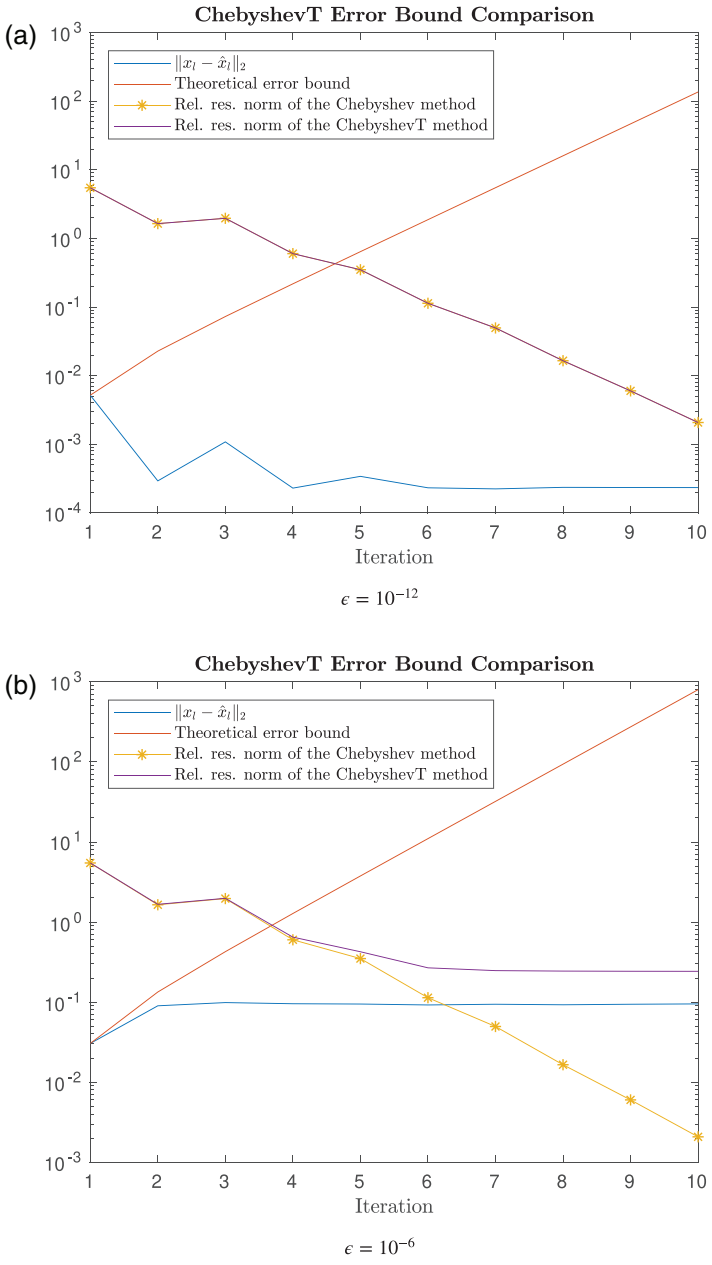


FIGURE 2 A numerical evaluation of the error bound from Theorem 2. All entries of the start matrix are set to 1

least if it is compared with the term Equation (20). For our configuration, $c = 0.6$. Hence, the coefficients α_i are bigger than 1 on the other hand (see (2.24) in [3]). For $\sigma_{\mathcal{P}} \approx c + d = 1.6$

$$1 + |\alpha_{l-1}|\sigma_{\mathcal{P}} \geq 2.6.$$

This explains why the first term in Theorem 2, the term Equation 20, dominates the error bound and makes it explode. Thus, when using ChebyshevT, one has to be very careful about the choice of the truncation tolerance.

8 | NUMERICAL EXAMPLES

8.1 | A three dimensional jetty in a channel

The geometric configuration of a 3d jetty in a channel is given by

$$\Omega := (0,8) \times (0,8) \times (0,4), S := (3,4) \times (0,8) \times (0,2) \text{ and } F := \Omega \setminus \bar{S}.$$

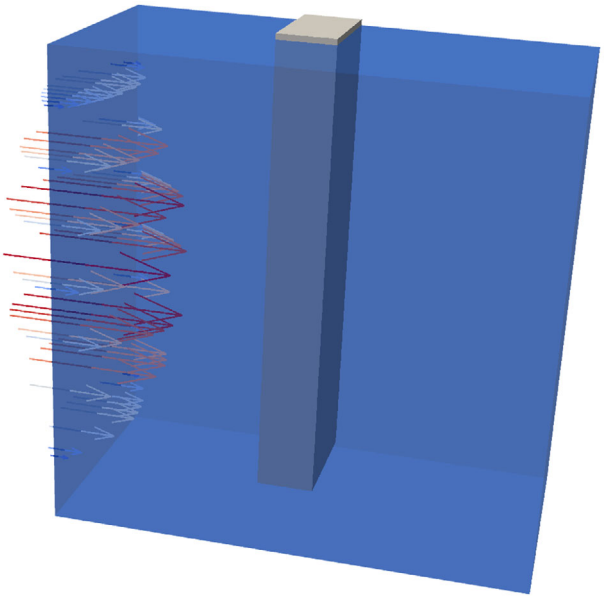


FIGURE 3 The initial configuration of the jetty where the Dirichlet data simulate an inflow from the left

With the velocity

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \in \mathbb{R}^3, \text{ the deformation } \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \in \mathbb{R}^3 \text{ and coordinates } (x, y, z) \in \bar{\Omega},$$

the Dirichlet inflow on the left boundary is given by

$$v = \begin{pmatrix} \frac{1}{2} y(8-y)z(4-z) \\ 0 \\ 0 \end{pmatrix} \quad \text{if } x = 0.$$

The geometric configuration is illustrated in Figure 3. On the right, for $x = 8$, do nothing boundary outflow conditions as discussed in Section 2.4.2 of [6] hold. The surface is at $y = 8$. There, v_2 and u_2 vanish. Everywhere else on $\partial(\Omega)$, the velocity and the deformation fulfill zero Dirichlet boundary conditions.

8.2 | Stabilization of equal-order finite elements

Because we use Q_1 finite elements for the pressure, velocity and the deformation, we have to stabilize the Stokes fluid equations on the fluid part. For this, stabilized Stokes elements as explained in Lemma 4.47 of [6] are used.

8.3 | Three parameter discretization

Problem (4) is discretized with respect to

20 shear moduli $\mu_s^{i_1} \in [30000, 50000]$,

20 first Lamé parameters $\lambda_s^{i_2} \in [100000, 200000]$ and

20 fluid densities $\rho_f^{i_3} \in [50, 200]$.

The kinematic fluid viscosity is fixed to $\nu_f = 0.01$. The shear modulus and first Lamé parameter ranges cover solids with Poisson ratios between $\frac{1}{3}$ (e.g., concrete) and 0.43478 (e.g., clay). The total number of equations is $m = 20^3 = 8000$ and the number of degrees of freedom is $M = 192423$.

In the following computations, MATLAB 2017b on a CentOS 7.6.1810 64bit with 2 AMD EPYC 7501 and 512GB of RAM is used. The htucker MATLAB toolbox [11] is used to realize the Tucker format T_R. The preconditioners are decomposed into a permuted LU decomposition using the MATLAB builtin command `lu()`. All methods start with a start matrix whose entries are all set to 1.

8.4 | GMREST

A standard GMRES approach is compared with the GMRESTR method from Algorithm 2.

By "standard GMRES approach", the standard GMRES method applied to $m = 8000$ different equations of the form (5) is meant. It is once restarted after 8 iterations so it uses a total of 16 iterations per equation. For all 8000 separate standard GMRES methods, 5 preconditioners given by

$$A_0 + (D_1)_{i,i}A_1 + (D_2)_{i,i}A_2 + (D_3)_{i,i}A_3 \text{ for } i \in \{800, 2400, 4000, 5600, 7200\} \tag{22}$$

are set up where the diagonal matrices $\{D_j\}_{j\in\{1,2,3\}}$ are the ones from (7).

The GMRESTR method uses 6 iterations per restart and is restarted 3 times. The mean based preconditioner $\mathcal{P}_T$ is used. The times to compute the preconditioners (one in the case of GMRESTR, 5 in the case of standard GMRES) can be found in the column Precon. in Table 1. The method itself took the time that is listed in the column Comp. and the column Total is then the sum of these times. Both methods result in 8000 approximations. Each of these approximations (x axis) then provides a certain accuracy (y axis) that is plotted in Figure 4. The standard GMRES method applied to 8000 equations in this way provides accuracies that are plotted in red within about 90 hours and 32 minutes. The approximations the

TABLE 1 GMRESTR compared with standard GMRES

		Computation times (in minutes)		
Method	**Approx. storage**	**Precon.**	**Comp.**	**Total**
GMRESTR	$O[(M+m+R)R]$			
($R = 200$)	$\approx$ 306.12MB	1.24	179.88	**181.12**
Standard GMRES	$O(Mm)$			
(8000 times)	$\approx$ 11.47GB	6.63	5426.23	**5432.86**

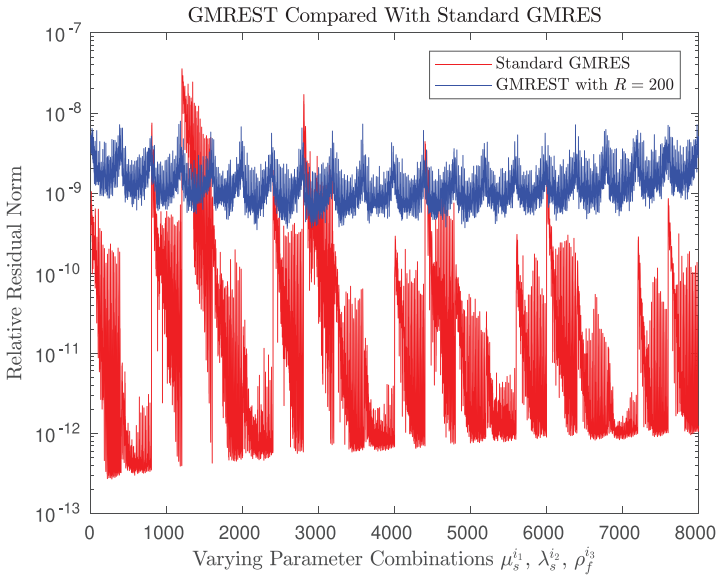


FIGURE 4 The standard GMRES method applied to 8000 separate equations (relative residual norms in red) is compared with the GMREST method (relative residual norms in blue). The relative residual norms are $\frac{\|b_D - A(\mu_s^{i_1}, \lambda_s^{i_2}, \rho_f^{i_3})x_{i_1,i_2,i_3}\|_2}{\|b_D\|_2}$, where x_{i_1,i_2,i_3} is the approximation related to the parameters $\mu_s^{i_1}$, $\lambda_s^{i_2}$ and $\rho_f^{i_3}$

GMRESTR method provides have accuracies that are plotted in blue. The GMRESTR method took only about 181 minutes to compute these approximations as one can see in Table 1. Also, the storage that is needed to store the approximation varies significantly. The rank 200 approximation, in the Tucker format, requires only about 306MB whereas the full matrix requires almost 12GB.

8.5 | ChebyshevT

Before the Chebyshev method can be applied, the extreme eigenvalues of $\mathcal{P}_T^{-1}\mathcal{A}$ have to be estimated as explained in Section 4.3. An estimation of $\Lambda_{\max}$ and $\Lambda_{\min}$ from (11) involves the estimation of extreme eigenvalues for m different matrices if the representation (10) is considered. But we restrict to an estimation of

$$\bar{\Lambda}_{\max} = \max_{\substack{i_1\in\{1,m_1\}\\ i_2\in\{1,m_2\}\\ i_3\in\{1,m_3\}}} \Lambda(Bl(i_1,i_2,i_3)) \approx \Lambda_{\max} \quad \text{and} \quad \bar{\Lambda}_{\min} = \min_{\substack{i_1\in\{1,m_1\}\\ i_2\in\{1,m_2\}\\ i_3\in\{1,m_3\}}} \Lambda(Bl(i_1,i_2,i_3)) \approx \Lambda_{\min}.$$

With the mean based preconditioner $\mathcal{P}_T$, this leads to $d = 1$ and $c = 0.6$ in this configuration. The time needed to compute $\bar{\Lambda}_{\max}$ and $\bar{\Lambda}_{\min}$ on a coarse grid with $M = 735$ degrees of freedom is listed in the column "Est." in Table 2.

In the same manner as in the preceding subsection, for comparison, a standard Cheyshev approach is applied to 8000 equations of the form (5). The resulting accuracies are visualized in Figure 5. The standard Chebyshev method uses 20 iterations at each equation and, in total, the same 5 preconditioners (22) as the standard GMRES uses. The ChebyshevT method iterates, in total, 24 times and uses $\mathcal{P}_T$, the mean based preconditioner. The ChebyshevT method is restarted 3 times with 6 iterations per restart. Compared to this, 24 iterations without restart took about the same time but provide approximation accuracies that are slightly worse.

TABLE 2 ChebyshevT compared with standard Chebyshev

		Computation times (in minutes)			
Method	**Approx. storage**	**Est.**	**Precon.**	**Comp.**	**Total**
ChebyshevT	$O[(M+m+R)R]$				
($R = 200$)	$\approx$ 306.12MB	0.013	1.24	177.99	**179.243**
Standard Chebyshev	$O(Mm)$				
(8000 times)	$\approx$ 11.47GB	0.013	6.63	5490.85	**5497.493**

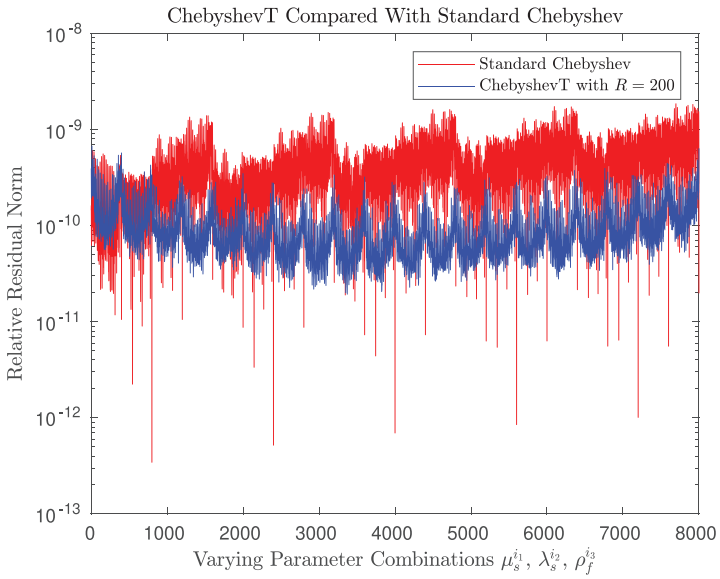


FIGURE 5 The standard Chebyshev method applied to 8000 separate equations (relative residual norms in red) is compared with the ChebyshevT method (relative residual norms in blue)

8.6 | Comparison with the Bi-CGstab method

Another method that also works for non-symmetric matrices is the Bi-CGstab method [4]. It was not considered in the first place because it can break down under some circumstances as explained in Section 2.3.8 of [18]. The preconditioned truncated variant similar to Algorithm 3 of [1] but strictly based on [4] is compared with the GMRESTR and the ChebyshevT method in Figure 6. The truncated Bi-CGstab method is applied with 6 iterations per restart. If once restarted, in total, the method iterates 12 times. The resulting approximation accuracy is indeed better than the one obtained when iterating 12 times directly without any restart.

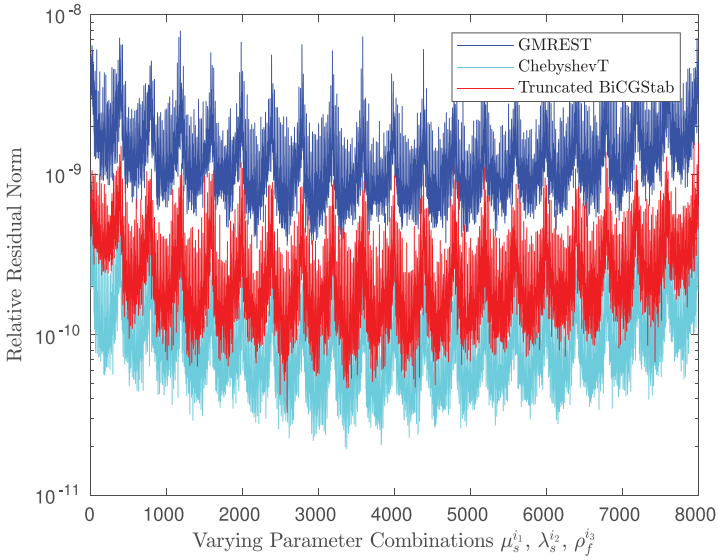


FIGURE 6 The approximation accuracies for the GMREST (blue), the ChebyshevT (cyan) and the truncated Bi-CGstab (red)

To avoid early stagnation, the residual at step i is computed directly

$$\hat{R}_i = \mathcal{T}\left(\hat{B} - F(\hat{X})\right).$$

The main reason why the Bi-CGstab takes more time (compare Table 3), is the truncation. If M and m are big, truncation after every addition is indispensable. In this way, in every for-loop in the preconditioned Bi-CGstab Algorithm[4], a total of 12 truncations occur (3 times, $F(\cdot)$ is evaluated). In a for-loop of Algorithm 3, we have 6 truncations only. In the outer for-loop of Algorithm 1, $3 + i$ truncations occur. It turned out that, in the implementation of the preconditioned Bi-CGStab method of [4], the truncation of the sum

$$s = r_{i-1} - \alpha v_i$$

can be left out. Leaving out further truncations did not lead to a better performance of the truncated Bi-CGstab method.

TABLE 3 Computation time comparison of the truncated approaches

	Computation times (in minutes)			
Method (R=200)	**Est.**	**Precon.**	**Comp.**	**Total**
ChebyshevT	0.013	1.24	177.99	**179.24**
GMREST	-	1.24	179.88	**181.12**
Truncated Bi-CGstab	-	1.24	302.94	**304.18**

9 | CONCLUSIONS

The truncated methods discussed in this paper provide approximations with relative residual norms smaller than 10^{-8} within less than a twentieth of the time needed by the correspondent standard approaches that solve the m equations individually. This raises the question how these methods perform when applied to nonlinear problems.

Since the truncation error affects, in addition to the machine precision error, the accuracy of the Arnoldi orthogonalization, the GMREST method should preferably be applied in a restarted version. Mostly, the ChebyshevT method is a bit faster and a bit more accurate than the GMREST method. But the main disadvantage of the ChebyshevT method is that the ellipse that contains the eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ described by the foci $d \pm c$ has to be approximated newly every time the parameter configuration changes. In this matter, the GMREST method can be seen as a method that is a bit more flexible if compared to the ChebyshevT method.

Also, the ChebyshevT and the truncated BiCGstab methods can and preferably should be applied in a restarted manner. If not restarted, the methods stagnate after a few iterations already. The reason is a numerical issue initiated by the bad condition of the mean-based preconditioner.

There is still further investigation needed regarding the error bounds. If the GMREST method is applied, the coefficients c_j are not known. The ChebyshevT bound is rather pessimistic and merely of theoretical nature. The method seems to converge too fast such that the truncation error does not really play a role in the cases examined.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge funding received by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 314838170, GRK 2297 MathCoRe.

ORCID

Roman Weinhandl https://orcid.org/0000-0002-9728-8682
Peter Benner https://orcid.org/0000-0003-3362-4103
Thomas Richter https://orcid.org/0000-0003-0206-3606

REFERENCES

[1] Kressner, D., Tobler, C.: Low-rank tensor Krylov subspace methods for parametrized linear systems. SIAM J. Matrix Anal. Appl. 32(4), 1288–1316 (2011)
[2] Ballani, J., Grasedyck, L.: A projection method to solve linear systems in tensor format. Numer. Linear Algebra Appl. 20(1), 27–43 (2013)
[3] Manteuffel, T.A.: The Tchebychev iteration for nonsymmetric linear systems. Numer. Math. 28(3), 307–327 (1977)
[4] van der Vorst, H.A.: Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Statist. Comput. 13(2), 631–644 (1992)
[5] Benner, P., Breiten, T.: Low rank methods for a class of generalized Lyapunov equations and related issues. Numer. Math. 124(3), 441–470 (2013)
[6] Richter, T.: Fluid-Structure Interactions. Lecture Notes in Computational Science and Engineering, vol. 118. Springer, Cham (2017)
[7] Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th ed. *Johns Hopkins Studies in the Mathematical Sciences*. Johns Hopkins University Press, Baltimore, MD (2013)
[8] Trefethen, L.N., Bau III, D.: Numerical Linear Algebra. Society for Industrial and Applied Mathematics, Philadelphia, PA (1997)
[9] Hackbusch, W.: Tensor Spaces and Numerical Tensor Calculus. Springer Series in Computational Mathematics, vol. 42. Springer, Heidelberg (2012)
[10] Grasedyck, L.: Hierarchical singular value decomposition of tensors. SIAM J. Matrix Anal. Appl. 31(4), 2029–2054 (2009/10)
[11] Kressner, D., Tobler, C.: Algorithm 941: htucker—a MATLAB toolbox for tensors in hierarchical Tucker format. ACM Trans. Math. Software 40(3), 22 (2014)
[12] Oseledets, I.V.: Tensor-train decomposition. SIAM J. Sci. Comput. 33(5), 2295–2317 (2011)
[13] Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd ed. Society for Industrial and Applied Mathematics, Philadelphia, PA (2003)
[14] Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Statist. Comput. 7(3), 856–869 (1986)
[15] Calvetti, D., Golub, G.H., Reichel, L.: An adaptive Chebyshev iterative method for nonsymmetric linear systems based on modified moments. Numer. Math. 67(1), 21–40 (1994)
[16] Simoncini, V., Szyld, D.B.: Theory of inexact Krylov subspace methods and applications to scientific computing. SIAM J. Sci. Comput. 25(2), 454–477 (2003)
[17] Hackbusch, W., Khoromskij, B.N., Tyrtyshnikov, E.E.: Approximate iterations for structured matrices. Numer. Math. 109(3), 365–383 (2008)
[18] Barrett, R., Berry, M., Chan, T.F., et al.: Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. Society for Industrial and Applied Mathematics, Philadelphia, PA (1994)

How to cite this article: Weinhandl R, Benner P, Richter T. Low-rank linear fluid-structure interaction discretizations. *Z Angew Math Mech*. 2020;100:e201900205. https://doi.org/10.1002/zamm.201900205

APPENDIX A: ERROR BOUNDS

A.1 | GMREST error bounds

Lemma 4 (**Error Bound for Truncated Basis Elements**). *Under the assumptions of Lemma 2, with* $\|\mathcal{E}_{\hat{r}_0}\|_2 \leq \epsilon$, *it holds*

$$e_k \leq \epsilon \sum_{j=1}^{k+1} \sigma_{\mathcal{P}}^{j-1} + \epsilon k \qquad \text{for} \qquad k \in \{0, \dots, l-1\}.$$

Proof. The case $k = 0$ is clear. For $k \geq 1$, we have

$$e_k = \|(\mathcal{P}^{-1}\mathcal{A})^k \mathcal{E}_{\hat{r}_0} + \mathcal{E}_{K^{\mathcal{T}_k}} + \sum_{j=1}^{k-1} (\mathcal{P}^{-1}\mathcal{A})^j \mathcal{E}_{K^{\mathcal{T}_{k-j}}} \|_2 \leq \sigma_{\mathcal{P}}^k \epsilon + \epsilon + \sum_{j=1}^{k-1} \sigma_{\mathcal{P}}^j \epsilon = \epsilon \sum_{j=1}^{k+1} \sigma_{\mathcal{P}}^{j-1}.$$

The term ϵk is to be added to the error bound as explained in Remark 6. □

Theorem 3 (**Approximation Error of GMREST**). *Under the assumptions of Theorem 1 and* $\|\mathcal{E}_{\hat{r}_0}\|_2 \leq \epsilon$, *it holds*

$$\|\hat{x}_l - x_l\|_2 \leq \epsilon \sum_{j=1}^{l} |d_j| \left(\sum_{i=1}^{j} \sigma_{\mathcal{P}}^{i-1} + j - 1 \right) + \sum_{j=1}^{l} |c_j - d_j| + \epsilon l.$$

Proof. We use the proof of Theorem 1 and the error bound of Lemma 4 to estimate

$$\|\hat{x}_l - x_l\|_2 \leq \sum_{j=1}^{l} \left(|d_j| e_{j-1} + |c_j - d_j| \right) \leq \epsilon \sum_{j=1}^{l} |d_j| \left(\sum_{i=1}^{j} \sigma_{\mathcal{P}}^{i-1} + j - 1 \right) + \sum_{j=1}^{l} |c_j - d_j|.$$

Truncation in the last line of Algorithm 1 leads to the additional sum term ϵl. □

A.2 | ChebyshevT error bounds

Theorem 4 (**Approximation Error of ChebyshevT**). *Under the assumptions of Theorem 2 and* $\|\mathcal{E}_{\hat{r}_i}\|_2 \leq \epsilon$ *for all* $i \in \{1, \dots, l\}$, *the error bounds of the ChebyshevT method translate to*

$$e_1 := \|\hat{x}_l - x_l\|_2 = \|\mathcal{E}_{\hat{x}_1^a}\|_2 \leq \epsilon \left(1 + \frac{1}{|d|} \right),$$

$$e_2 := \|\hat{x}_2 - x_2\|_2 \leq \epsilon \left(3 + |\alpha_1|\sigma_{\mathcal{P}} + |\alpha_1| + \frac{1 + |\beta_1| + |\alpha_1|\sigma_{\mathcal{P}}}{|d|} \right) \quad \text{and}$$

$$e_l := \|\hat{x}_l - x_l\|_2 \leq (1 + |\alpha_{l-1}|\sigma_{\mathcal{P}}) e_{l-1} + \sum_{j=1}^{l-2} |\alpha_j| e_j \sigma_{\mathcal{P}} \prod_{i=1}^{l-j-1} |\beta_{i+j}|$$

$$+ \epsilon \left(2 + \sum_{j=1}^{l-2} \prod_{i=1}^{l-j-1} |\beta_{i+j}| + \sum_{j=1}^{l-1} |\alpha_j| \prod_{i=1}^{l-j-1} |\beta_{i+j}| + \frac{\prod_{j=1}^{l-1} |\beta_j|}{|d|} \right) \quad \text{for} \quad l \geq 3.$$

Proof. For all cases, we can use the proof of Theorem 2 and estimate $\epsilon_R \leq \epsilon$. □