

RESEARCH ARTICLE

Open Access



Application of an iterative Golub-Kahan algorithm to structural mechanics problems with multi-point constraints

Carola Kruse^{1*}, Vincent Darrigrand¹, Nicolas Tardieu² , Mario Arioli³ and Ulrich Rude^{1,4}

*Correspondence:
carola.kruse@cerfacs.fr
¹Cerfacs, 29 Avenue Gaspard
Coriolis, 31100 Toulouse, France
Full list of author information is
available at the end of the article

Abstract

Kinematic relationships between degrees of freedom, also named multi-point constraints, are frequently used in structural mechanics. In this paper, the Craig variant of the Golub-Kahan bidiagonalization algorithm is used as an iterative method to solve the arising linear system with a saddle point structure. The condition number of the preconditioned operator is shown to be close to unity and independent of the mesh size. This property is proved theoretically and illustrated on a sequence of test problems of increasing complexity, including concrete structures enforced with pretension cables and the coupled finite element model of a reactor containment building. The Golub-Kahan algorithm converges in only a small number of steps for all considered test problems and discretization sizes. Furthermore, it is robust in practical cases that are otherwise considered to be difficult for iterative solvers.

Keywords: Iterative solvers, Indefinite systems, Saddle point, Golub-Kahan bidiagonalization, Structural mechanics, Multi-point constraints

Introduction

In structural mechanics, it is very common to impose kinematic relationships between degrees of freedom (DOF) in a finite element model (see [1, section 35.2.2] or [2, section 2.6]). Rigid body conditions of a stiff part of a mechanical system or cyclic periodicity conditions on a mesh representing only a section of a periodic structure are typical examples of this approach. Such conditions can also be used to glue non-conforming meshes or meshes containing different types of finite elements. For example, we could link a thin structure modeled by shell finite elements to a massive 3D structure modeled with continuum finite elements. These kinematic relationships are often called multi-point constraints (MPC) in standard finite element software and can be linear or nonlinear [2, section 3.4]. As an industrial example, one can consider the prestressed concrete structure of a reactor containment building that can be modeled as a problem in elasticity for which one-, two- and three-dimensional finite elements (representing metallic cables, the inner shell, and the concrete block) are coupled by MPC.

Before going further, it is important to mention a particular characteristic of several MPC. From a mathematical point of view, it has been found that the best way to introduce the constraints may be on the continuous level, i.e. in the weak form of a partial differential equation, as with the mortar approach [3]. This allows keeping optimal convergence rates when the size of the mesh tends to zero. Unfortunately, in industrial software, the constraints are often imposed on the already discretized equations. As it is usually not possible to make major modifications to existing legacy code, any method of mortar-type becomes unfeasible. In this article, we will focus on the situation where the constraints are introduced on the discrete level. These constraints are purely algebraic and contrary to, e.g., the Stokes equations, where properties of the partial differential equation for the divergence term can be exploited to construct efficient solver-preconditioner combinations [4], the investigation here needs to be done mainly on the matrix level.

MPC can be enforced on the discretized equations by Lagrange multipliers so that the resulting matrix is symmetric indefinite with a two by two block structure, also called a saddle-point structure. The first diagonal block, the (1, 1)-block of size $m \times m$, can be denoted as the “physical block” since it involves, for instance, the displacement DOF. The second diagonal block, the (2, 2)-block of size $n \times n$, is zero. The blocks (1, 2) and (2, 1) are the transposition one of the other and, hence, the matrix is symmetric. It can be denoted as the “Lagrange” block since it describes the interaction between the Lagrange multipliers DOF and the displacement DOF.

The efficient solution of this type of linear system has stimulated intensive research in a wide range of applications fields in the past years; see [5] for a comprehensive review. We will briefly describe the most interesting ones in the context of MPC. One of the commonly used methods is the Schur complement reduction technique [5, section 5], which requires an invertible physical block. It then has the conceptual advantage of manipulating two linear systems of size m and n , rather than one system of size $m + n$. There is, however, the disadvantage that the Schur complement matrix will usually be dense and thus becomes expensive or impossible to compute, except in the case where the number n of constraints is very small.

Krylov subspace methods can be of major interest when solving saddle point systems [6]. In realistic finite element applications, the saddle point matrix can be very poorly conditioned. As it is discussed in [5, section 3.5], when the mesh size parameter h tends to zero, its condition number must be expected to increase. Krylov subspace methods will thus perform poorly with large-sized problems and rely on good, generally problem-specific, preconditioning techniques [5].

Another method to solve the saddle point system is based on an elimination technique [7–9]. This strategy implies major modifications of the matrix of the linear system, whose profile can become much denser. The underlying algorithm is often sequential, where each constraint is treated one after the other. Consequently, this technique can not be used easily in a parallel framework. Finally, the elimination procedure requires to sort the constraints in two sets, namely dependent and independent. This may be simple for some particular constraints but tends to be intractable for fully general constraints.

Still another approach is used in [10]. The authors introduce a projector on the orthogonal complement of the kernel of the constraints matrix \mathbf{A} , the (1, 2) block, and solve the linear system on that subspace with an iterative method. This subspace projection technique is elegant and favorable convergence properties are shown. Unfortunately, the

definition of the projector involves the factorization of the operator $\mathbf{A}^T \mathbf{A}$ which, in many practical cases, can be quite dense, causing the factorization to be expensive in time and space. Furthermore, one forward-backward substitution is needed at each iteration of the iterative method.

Our approach for providing an efficient solver is based on the Golub-Kahan bidiagonalization (GKB) method, which has been widely used in solving least-norm problems and in the computation of the singular value decomposition of rectangular matrices [11, 12]. We revisit a variant of the GKB algorithm for solving indefinite block matrices which was initially proposed by Arioli in [13] and that was found to be robust and accurate. Furthermore, Arioli observed computationally on a series of test problems that the number of iterations increases only weakly with increasing discretization size when some parameters in the GKB algorithm were chosen appropriately. This property comes in handy when solution methods are tested on small problems, but that are targeted for large scale simulations.

In a previous work [14], we have described our implementation of the generalized GKB solver into the parallel linear algebra library PETSc [15–17]. The GKB algorithm is low in memory consumption, as the iterates of the solution are computed on the fly and no basis needs to be stored. This is especially advantageous when it comes to the computation of large scale problems. However, the algorithm needs to solve linear systems of the size of the physical as well as the constraint block in each iteration, which defines the principal bottleneck of the method. In terms of scalability, we have investigated the parallel performance of the solver when applied to two test problems based on the Stokes equations provided in the PETSc examples in [14]. The solver showed high scalability (the best case scenario showed a speedup half as good as the theoretical speedup for a large number of processes), which was favored in this case by the availability of multigrid methods for solving the (1,1)-block corresponding to the discretization of the Laplace equation.

In this paper, we want to show the applicability and performance of the Golub-Kahan solver to another class of problems. We present a systematic convergence study for three sets of linear elastic test problems augmented with MPC and increasing complexity for which commonly used iterative solvers show poor performances. The matrices are generated by the finite element software code_aster¹. Furthermore, from a theoretical point of view, the major contribution of this paper is the proof of Arioli observation that the number of iterations can be made independent of the discretization size when some parameters are chosen appropriately in the GKB method. Additionally, in [13] the constraints were approximations of differential operators and using the inf-sup condition [18] it is possible to choose the parameters in an optimal way. For the considered problems and due to the nature of the discrete MPC, this has never been investigated.

The paper is organized as follows. In the first section, we introduce the problem settings and review briefly the GKB algorithm for saddle point problems. Furthermore, we present our main theorem stating that the number of required iterations is independent of the discretization size, whenever some parameters are well chosen. In the numerical section, we present first the convergence of the GKB solver for the test problem of a cylinder with imposed rigid body conditions, then study a concrete block with pretension cables

¹<http://www.code-aster.org>

and finish with the realistic industrial application of the structural analysis of a reactor containment building. Finally, we will investigate the parallel performance of the GKB algorithm for the example of a prestressed concrete block with an inner sparse direct parallel solver.

Problem settings

In this section, we first introduce the underlying continuous partial differential equations of our test problems. Starting from here, we develop the resulting discrete saddle point system and comment on possible cases of the definiteness of the physical block. Finally, we introduce an augmented Lagrangian approach that transforms the system into one with a positive definite (1,1)-block, as it is needed in the GKB method.

Governing equations

In the following, we focus on the equilibrium of an elastic body under the small displacement hypothesis, for which the problem is to find the displacement field $\mathbf{u} : \bar{\Omega} \rightarrow \mathbb{R}^3$ such that

$$-\text{div}(\sigma(\mathbf{u})) = \mathbf{f}, \quad \text{in } \Omega, \quad (1)$$

$$\sigma(\mathbf{u})\mathbf{n} = \mathbf{h}, \quad \text{on } \Gamma_N, \quad (2)$$

$$\mathbf{u} = \mathbf{u}_D, \quad \text{on } \Gamma_D. \quad (3)$$

Here \mathbf{h} and \mathbf{u}_D are the Neumann and the Dirichlet data and the stress and strain tensors are defined as

$$\sigma(\mathbf{u}) = \mathcal{C}\epsilon(\mathbf{u}), \quad (4)$$

$$\epsilon(\mathbf{u}) = (\nabla\mathbf{u} + \nabla^T\mathbf{u})/2. \quad (5)$$

In the elastic case, \mathcal{C} is the fourth order elastic coefficient (or Hooke's law) tensor satisfying both symmetry and ellipticity conditions. Furthermore, the constitutive law (4) connects linearly σ to the strain tensor field ϵ .

The problem is discretized with standard Lagrange finite elements. In the case of a well-posed mechanical problem, the solution of the linearized problem can be expressed as the following constrained minimization problem

$$\min_{\mathbf{A}^T\mathbf{w}=\mathbf{r}} \frac{1}{2}\mathbf{w}^T\mathbf{W}\mathbf{w} - \mathbf{g}^T\mathbf{w}, \quad (6)$$

$\mathbf{W} \in \mathbb{R}^{m \times m}$ is the tangent stiffness matrix,

$\mathbf{A} \in \mathbb{R}^{m \times n}$ is the linearized full-ranked matrix of the constraints,

$\mathbf{w} \in \mathbb{R}^m$ is the vector of nodal displacement unknowns,

$\mathbf{g} \in \mathbb{R}^m$ is the volume force vector,

$\mathbf{r} \in \mathbb{R}^n$ is the data vector for inhomogeneous constraints,

where m corresponds to the physical DOF, n corresponds to the number of constraints. The constraints describe kinematic relationships between degrees of freedom. They can arise from modeling choices: turn a stiff part of a mechanical system into a perfectly rigid body, apply cyclic periodicity conditions on a mesh representing only a section of a

periodic structure, connect a thin structure modeled by shell finite elements to a massive 3D structure modeled with continuum finite elements in order to correctly transfer forces and moments from the latter to the former. A basic example consists in imposing the equality of two degrees of freedom. In the following, the constraints are coming from the discrete MPC, see [19] for details. With the introduction of Lagrange multipliers \mathbf{p} , the augmented system that gives the optimality conditions for (6) reads

$$\begin{bmatrix} \mathbf{W} & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \mathbf{r} \end{bmatrix} \tag{7}$$

which is the classical shape of a saddle-point linear system. Note that this system is symmetric, but only indefinite as it contains positive and negative eigenvalues. To guarantee that the general system (7) admits a unique solution, we additionally assume that

$$\ker(\mathbf{W}) \cap \ker(\mathbf{A}^T) = \{0\} \text{ and } \ker \mathbf{A} = \{0\}. \tag{8}$$

Case of hanging parts in the model

In many cases, the stiffness matrix \mathbf{W} is symmetric positive definite. This is the case when all rigid body modes are suppressed by adequate boundary conditions and are integrated into the tangent stiffness matrix. Nevertheless, there are also situations where it is not the case. First, one may think of boundary conditions that are not integrated into the tangent stiffness matrix but are included in the constraint matrix \mathbf{A} . Second, as will be studied in Example 2, one may think of a classical model of prestressed concrete. Here a block of concrete is augmented by embedded steel cables. The concrete is modeled with 3D finite elements and the cables with 1D finite elements, whose DOF are coupled to the concrete DOF with the help of MPC. In such a case, the stiffness matrix \mathbf{W} is no longer symmetric positive definite but only symmetric positive semidefinite, caused by the rigid body modes of the steel cables.

In our further analysis, a symmetric positive definite (1,1)-block in (7) is however required. A common method is to apply an *augmented Lagrangian approach* as described in [20]. Let therefore $\mathbf{N} \in \mathbb{R}^{n \times n}$ be a symmetric positive definite matrix and $0 \leq \xi \leq 1$. With the transformation

$$\begin{aligned} \mathbf{M} &= \mathbf{W} + \xi \mathbf{A} \mathbf{N}^{-1} \mathbf{A}^T \\ \mathbf{u} &= \mathbf{w} - \mathbf{M}^{-1}(\mathbf{g} + \xi \mathbf{A} \mathbf{N}^{-1} \mathbf{r}) \\ \mathbf{b} &= \mathbf{r} - \mathbf{A}^T \mathbf{M}^{-1}(\mathbf{g} + \xi \mathbf{A} \mathbf{N}^{-1} \mathbf{r}), \end{aligned} \tag{9}$$

(7) is transformed into the equivalent system

$$\begin{bmatrix} \mathbf{M} & \mathbf{A} \\ \mathbf{A}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{b} \end{bmatrix}. \tag{10}$$

The (1,1)-block is now symmetric positive definite thanks to property (8). This kind of regularization can also be applied when \mathbf{W} is symmetric positive definite, with the goal that for a suitably chosen \mathbf{N} , we may find that (10) becomes easier to solve than the original system. In the following, we will use the notation \mathbf{M} exclusively for a symmetric positive definite matrix and we set $\xi = 1$.

The generalized Golub-Kahan bidiagonalization algorithm

In this section, we first recall Craig’s variant of the generalized Golub-Kahan bidiagonalization (referred to as GKB algorithm in the following) as stated in [13,21]. Then, we prove a convergence property concerning the number of outer iterations of the GKB algorithm. For this, a saddle point system with a symmetric positive definite (1,1)-block \mathbf{W} is required. We apply the augmented Lagrangian approach $\mathbf{M} = \mathbf{W} + \mathbf{A}\mathbf{N}^{-1}\mathbf{A}^T$ and the transformation (9), where the matrix \mathbf{N} is chosen, following the discussion in [20], as

$$\mathbf{N} = \frac{1}{\nu} \mathbf{I}, \quad \nu > 0.$$

The saddle point problem is then of the form described in equation (10).

The algorithm

The generalized GKB used in the following is derived from the application of the standard GKB once to the rectangular matrix $\tilde{\mathbf{A}} = \mathbf{M}^{-\frac{1}{2}}\mathbf{A}\mathbf{N}^{-\frac{1}{2}}$ and once to its transposed $\tilde{\mathbf{A}}^T$ [13,21]. To formulate an iterative algorithm, several methods could, in theory, be used for the bidiagonalization of $\tilde{\mathbf{A}}$ [11,22] and then be generalized to the saddle point system (10). Here, we focus on the aforementioned Craig’s variant algorithm for the solution of saddle point systems, which is presented in Algorithm 1. As stopping criterion, we use a normalized lower bound estimate of the energy norm error $\mathbf{e}_k := \|\mathbf{u} - \mathbf{u}^k\|_{\mathbf{M}}$ described in [13,14]. The algorithm stops once this normalized lower bound undershoots a sufficiently small tolerance τ . Indeed, an upper bound estimate would be a more reliable stopping criterion than the lower bound one. An approach based on the Gauss-Radau quadrature has been presented in [13]. This estimate relies on having an accurate lower bound a , with $0 < a < \sigma_n$, for all singular values σ_n of $\tilde{\mathbf{A}}$, which is in practice very difficult or even impossible to obtain. We will thus use in the following numerical experiments exclusively the lower bound estimate, which is found experimentally to be sharp enough. Note that we deliberately do not use a residual as a stopping criterion, as it is often the case in other iterative algorithms, since Craig’s variant algorithm minimizes the error \mathbf{e}_k in each step. We thus know that the error \mathbf{e}_k reduces monotonically, whereas a residual might lead to fluctuations.

Convergence properties of the GKB algorithm

We next state our main result on the convergence of the GKB method. The standard GKB process applied to $\tilde{\mathbf{A}}$ is equivalent to the generalized GKB applied to the saddle point system (10) [21]. In the following theorem, we show a bound on the condition number of $\tilde{\mathbf{A}}$ which, hence, also gives information about the convergence of the GKB method.

Theorem 1 *Let $\mathbf{M} = \mathbf{W} + \nu\mathbf{A}\mathbf{A}^T$ and \mathbf{W} be symmetric positive definite matrices and $\lambda_1 \leq \dots \leq \lambda_n$ be the eigenvalues of $\mathbf{A}^T\mathbf{W}^{-1}\mathbf{A}$. Let furthermore $\tilde{\mathbf{A}} = \sqrt{\nu}\mathbf{M}^{-\frac{1}{2}}\mathbf{A}$ and $0 < \eta < \frac{\lambda_n - \lambda_1}{\lambda_1}$.*

$$\text{If } \nu \geq \frac{1}{\eta\lambda_1} - \frac{1+\eta}{\eta\lambda_n}, \text{ then } \kappa(\tilde{\mathbf{A}}) \leq \sqrt{1 + \eta}.$$

The proof of the Theorem is given in Appendix . We shall now comment on the meaning of this result.

Algorithm 1 Generalized Golub-Kahan Bidigonalization, Craig’s variant.

Inputs: $\begin{bmatrix} M & A \\ A^T & 0 \end{bmatrix}$, $\begin{bmatrix} 0 \\ b \end{bmatrix}$, maxit, d , τ .
Outputs: $\begin{bmatrix} u \\ p \end{bmatrix}$.

// Bidiagonalization
 $\beta_1 = \|b\|_{N^{-1}}; \mathbf{q}_1 = N^{-1} \frac{b}{\beta_1};$
 $\mathbf{w} = M^{-1} A \mathbf{q}_1; \alpha_1 = \|\mathbf{w}\|_M; \mathbf{v}_1 = \frac{\mathbf{w}}{\alpha_1};$
// First Approximation
 $\zeta_1 = \frac{\beta_1}{\alpha_1}; \mathbf{d}_1 = \frac{\mathbf{q}_1}{\alpha_1}; \mathbf{p}_1 = -\zeta_1 \mathbf{d}_1; \mathbf{u}_1 = \zeta_1 \mathbf{v}_1;$

$k = 0; \text{convergence} = \text{false};$
while convergence is false and $k < \text{maxit}$:
 $k = k + 1;$
 // Bidiagonalization
 $\mathbf{g} = N^{-1}(A^T \mathbf{v}_k - \alpha_k N \mathbf{q}_k); \beta_{k+1} = \|\mathbf{g}\|_N; \mathbf{q}_{k+1} = \frac{\mathbf{g}}{\beta_{k+1}};$
 $\mathbf{w} = M^{-1}(A \mathbf{q}_{k+1} - \beta_{k+1} M \mathbf{v}_k); \alpha_{k+1} = \|\mathbf{w}\|_M; \mathbf{v}_{k+1} = \frac{\mathbf{w}}{\alpha_{k+1}};$
 // (k + 1)th Approximation
 $\zeta_{k+1} = -\frac{\beta_{k+1}}{\alpha_{k+1}} \zeta_k; \mathbf{d}_{k+1} = \frac{\mathbf{q}_{k+1} - \beta_{k+1} \mathbf{d}_k}{\alpha_{k+1}};$
 $\mathbf{u}_{k+1} = \mathbf{u}_k + \zeta_{k+1} \mathbf{v}_{k+1};$
 $\mathbf{p}_{k+1} = \mathbf{p}_k - \zeta_{k+1} \mathbf{d}_{k+1};$

// Check the convergence with the lower bound
if $k > d$ **then**
 $\xi^2 = \sum_{j=k-d+1}^k \zeta_j^2; \bar{\xi}^2 = \sum_{j=1}^k \zeta_j^2$
 if $\xi / \bar{\xi} \leq \tau$ **then** convergence = true;
end if
end while
 $\mathbf{u} = \mathbf{u}_{k+1}, \mathbf{p} = \mathbf{p}_{k+1},$

Let us recall that if the condition number of a matrix is 1, an iterative algorithm will converge in only one iteration. Consequently, the small condition number $\kappa(\tilde{\mathbf{A}})$ indicates that an iterative solver applied to a linear system defined by $\tilde{\mathbf{A}}$ converges in only a few iterations. We can conclude that also Algorithm 1 should converge with a small number of iterations to a sufficiently accurate solution. Furthermore, the above result states the optimal ν such that $\kappa(\tilde{\mathbf{A}}) \leq \sqrt{1 + \eta}$. This result is however only of theoretical nature, as the eigenvalues λ_i will in practice not be available. In general, if we choose ν “big enough”, the condition number of $\tilde{\mathbf{A}}$ is bounded by $\sqrt{1 + \eta}$. We can thus also expect the number of iterations to be independent of the mesh size for matrices obtained in constrained FEM discretizations, as long as we choose ν appropriately.

We have to keep in mind that linear systems with the matrices \mathbf{M} and \mathbf{N} have to be solved in each iteration in Algorithm 1. While $\mathbf{N}^{-1} = \nu \mathbf{I}$ reduces to a scalar multiplication, the condition number of \mathbf{M} depends on ν and thus on the smallest eigenvalue of $\mathbf{A}^T \mathbf{W}^{-1} \mathbf{A}$. The condition number of the resulting matrix $\mathbf{M} = \mathbf{W} + \nu \mathbf{A} \mathbf{A}^T$ could become very large for big ν . The solution of the linear systems in Algorithm 1 may thus become difficult, and additional numerical errors may be introduced. The possibly high condition number

of \mathbf{M} is especially problematic for large scale problems when an inner direct solver is no longer applicable and an iterative solver is applied. It is thus crucial to find an optimal value of ν to balance the number of inner and outer iterations. The numerical experiments suggest that in practice reasonable values of ν proportional to $\|\mathbf{W}\|_1$ reduce $\kappa(\tilde{\mathbf{A}})$ sensibly, without dramatically increasing the ill-conditioning of \mathbf{M} .

Numerical experiments

We will first present a convergence study and comment on the choice of ν on two test problems. To ensure the reproducibility of the results by the reader, we briefly describe the undertaken manipulations of the matrices received by code_aster (version 14.2). Let $\tilde{\mathbf{W}}$ be the either symmetric positive definite or symmetric positive semi-definite elasticity stiffness matrix obtained by code_aster and \mathbf{A} the constraint matrix. In code_aster, the Lagrange blocks are multiplied by a factor $\gamma := \frac{1}{2}(\min \tilde{\mathbf{W}}_{ii} + \max \tilde{\mathbf{W}}_{ii})$ to equilibrate the scaling of the matrix blocks (see [19]). We obtain the system

$$\begin{bmatrix} \tilde{\mathbf{W}} & \gamma \mathbf{A} \\ \gamma \mathbf{A}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ 0 \end{bmatrix}. \quad (11)$$

In our experiments, we observed that the factor γ and thus the maximum absolute values in the entries of this block matrix were large. Applying the GKB algorithm directly to it led to perturbations in the solution. We thus modified (11) by undoing the scaling, where we call $\mathbf{W} := \frac{1}{\gamma} \tilde{\mathbf{W}}$. To exploit the result of Theorem 1, we furthermore modified the (1, 1)-block to $\mathbf{M} = \mathbf{W} + \nu \mathbf{A} \mathbf{A}^T$ and transform (11) following (9) to obtain a system of type (10).

Influence of ν

In the following numerical examples, we choose the parameter $\nu = \|\mathbf{W}\|_1$ to better represent the energy subject to the MPC constraints, as described in the augmented system. The recommendation of Golub and Greif in [20], who found numerically that $\nu = \frac{\|\mathbf{W}\|}{\|\mathbf{A}\|^2}$ could be a good value, leads to too small an ν for our practical examples and increases the number of iterations noticeably. In the following we show numerically the pertinence of our choice.

Computational details

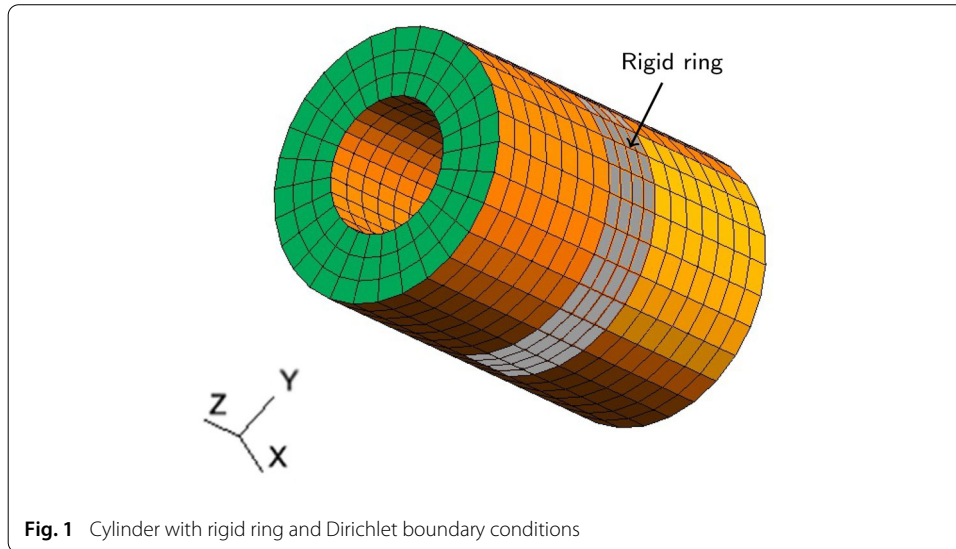
The delay parameter in the stopping criterion is fixed for all the following examples as $d = 5$ as suggested in [13] and the tolerance as $\nu = 10^{-5}$. In all of the following examples, we have used Algorithm 1, that we have implemented amongst fieldsplit preconditioning methods for block matrices in PETSc and that is available in the library since version 3.11. We invoke the algorithm by using the following options:

```
-ksp_type preonly -pc_type fieldsplit -pc_fieldsplit_type gkb
```

The delay parameter, the tolerance and ν are set as follows

```
-pc_fieldsplit_gkb_delay 5
-pc_fieldsplit_gkb_tol 1e-5
-pc_fieldsplit_gkb_nu nu
```

The inversion of \mathbf{M} is done using a direct solver:



```
-fieldsplit_0_ksp_type preonly -fieldsplit_0_pc_type lu
```

The reference solutions are obtained by solving the original augmented system (11) for the corresponding right-hand side received from code_aster, using the LU decomposition from the MULTifrontal Massively Parallel Solvers (MUMPS). In the convergence plot captions, we display the corresponding backward error estimates. However, due to the bad conditioning of the matrix, the backward error estimate can deteriorate. The workaround implemented in code_aster is the double Lagrangian approach described in [23]. The number of Lagrangian DOF doubles and increases the computational cost dramatically. To avoid this approach, as will be detailed in the sequel, we will encapsulate the direct LU decomposition in a Flexible GMRES (FGMRES) solver with a sufficiently small tolerance.

Example 1: Cylinder with rigid inner ring

For our first set of test cases, the domain Ω is chosen as a thick-walled cylinder as illustrated in Fig. 1. The model is a classical linear elasticity system, as described above, with m DOF approximated by a linear finite element method. Dirichlet boundary conditions are imposed on the left end and are shown in green. The others boundaries are with homogeneous Neumann. Furthermore, MPCs are applied to obtain a rigid inner ring, which is illustrated in Fig. 1 by the gray elements. The rest of the boundaries are free. For the derivation of the constraint equations, we refer to [19]. These kinematic relationships ensure that the inner ring resists any kind of outer forces. The stiffness matrix $\tilde{\mathbf{W}}$ is symmetric positive definite and the structure of the saddle point problem is shown in Fig. 2. To illustrate it, we apply a permutation to sort the matrix entries in the (1,1)-block with respect to the size of the diagonal elements of \mathbf{W} , starting from the smallest to the largest. Second, we apply a column permutation to the constraint block \mathbf{A} (and the respective row permutation for \mathbf{A}^T) to obtain the diagonal part in the upper $n \times n$ block.

Test cases and convergence results

We define four test problems, Prob. 1 to Prob. 4, with increasing resolution in Table 1, where *nnz* stands for the non-zero entries of the respective sparse matrices. The trans-

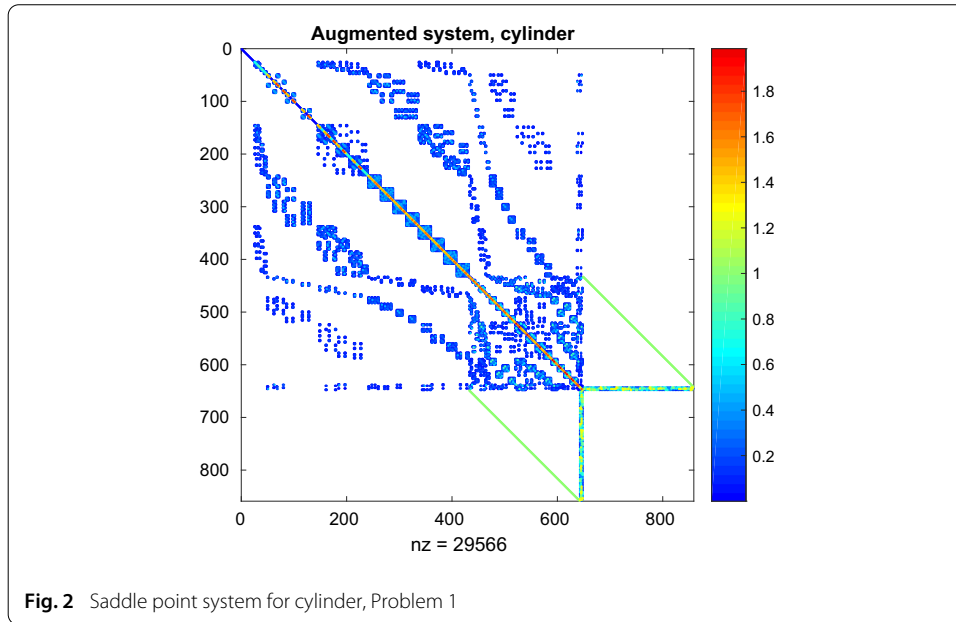


Table 1 Definition of the cylinder test problems Prob. 1 to Prob. 4

	m	n	$nnz (Mm)$	$nnz (A)$	$nnz (W)$
Prob. 1	648	210	30080	1259	28,296
Prob. 2	2520	714	1,47,800	4985	139,636
Prob. 3	6384	1674	4,09,246	10,045	3,92,816
Prob. 4	46,620	8814	33,67,462	26,436	32,62,086

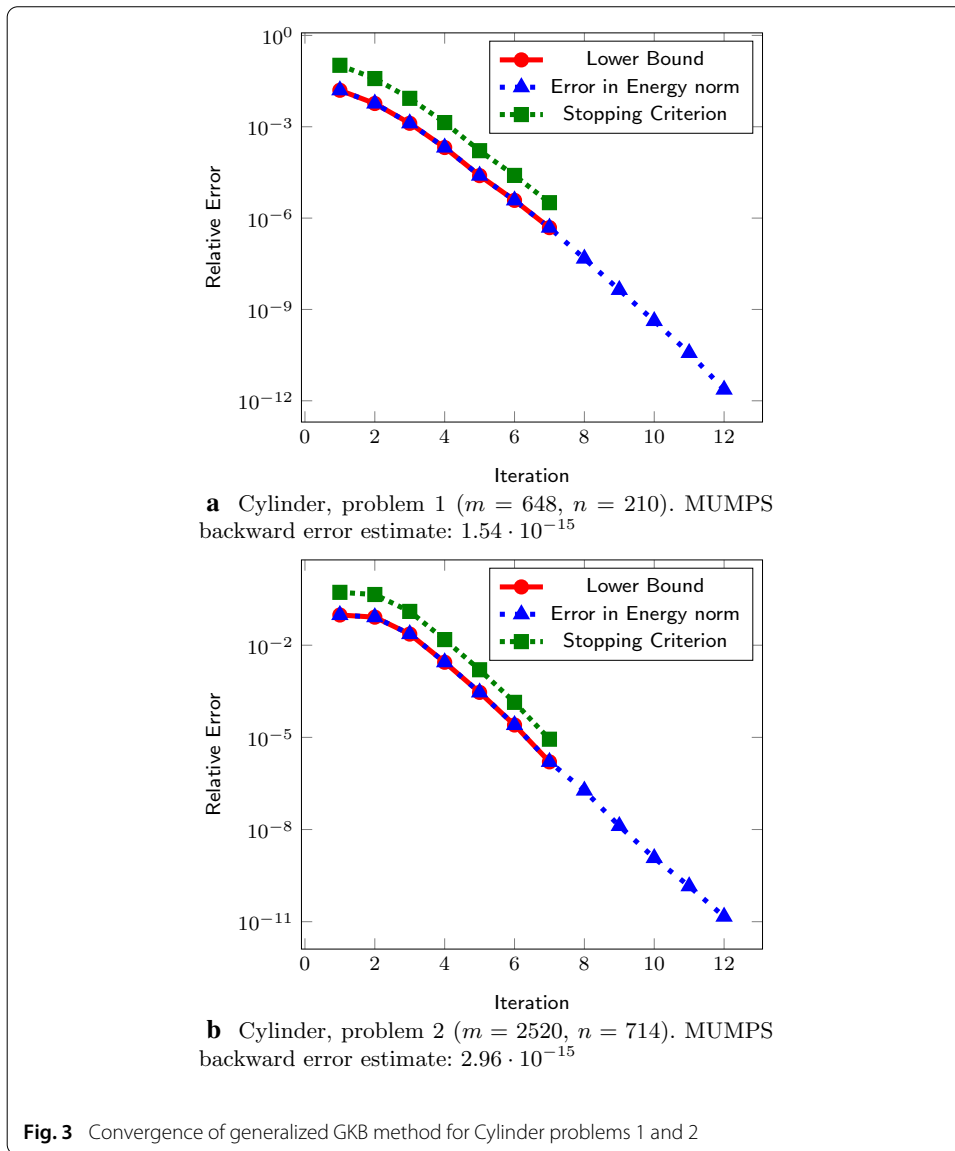
Table 2 Norms and 2-norm condition numbers of matrices for Problems 1–3, and 1-norm condition number of Problem 4 of Cylinder examples

	$\nu = \ W\ _1$	$\kappa (M)$	$\kappa (W)$	$\ A\ _1$
Prob. 1	9.13	$2.2 \cdot 10^5$	$1.9 \cdot 10^3$	6.27
Prob. 2	8.95	$1.9 \cdot 10^6$	$5.6 \cdot 10^3$	5.79
Prob. 3	8.85	$7.9 \cdot 10^6$	$1.1 \cdot 10^4$	5.74
Prob. 4	8.96	$5.0 \cdot 10^8$	$1.2 \cdot 10^5$	5.34

formation of M according to (9) increases the number of nonzero entries, but the ratios still stay reasonably small. In Table 2, the condition numbers and norms of the matrices corresponding to Prob. 1 to Prob. 4 are presented. As predicted, the condition number of M does increase in ν .

The convergence plots with lower bound estimates of the GKB method are presented in Figs. 3 and 4. We normalize the stopping criterion, the error and lower bound estimate differently, which explains the gap observed in the plots. While we use the energy norm of the approximated solution of (10), i.e. $\xi = \|u^k\|_M$, for the stopping criterion in Algorithm 1, we divide the error and the lower bound estimate for presentation in the Figures by the energy norm of the reference solution w of (7) which is in general smaller.

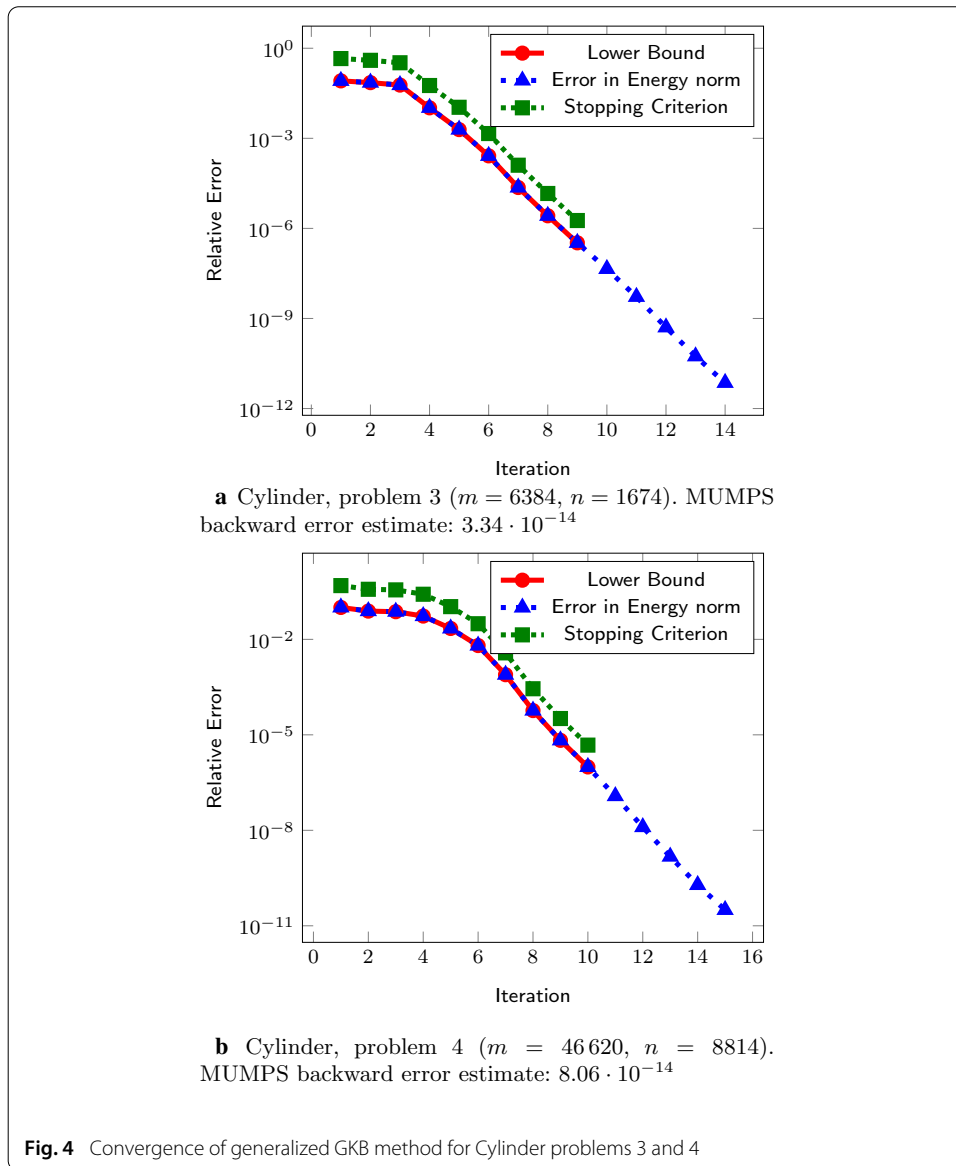
The stopping criterion of the GKB algorithm reaches the required tolerance of 10^{-5} already after 7 iterations for the smallest problem and after 7, 9 and 10 for Problems 2–4 (see Figs. 3 and 4), respectively. The lower bound for the error at iteration k is however computed only when iteration $k + d$ has been reached. Consequently, the GKB stops only



after 12 to 15 iterations. This, combined with the normalization above, explains why the final errors are remarkably smaller than the sought precision. We observe that although the number of DOF increases from Prob. 1 to 4, the number of iterations increases only marginally for each finer discretization and the algorithm stops after 15 iterations at the most. To obtain complete independence of the mesh size as it is shown in Theorem 1, ν would need to be chosen bigger, as it will be discussed next.

Discussion on ν

Theorem 1 provides the optimal choice of ν to make the number of iterations in Algorithm 1 independent of the mesh size of the finite element discretization. In general, we are not able to compute λ_1 and λ_n in Theorem 1 which would allow us to obtain a more precise estimate of ν for a given η . For the smallest three test problems above, we are however able to do so using MATLAB. Hence, we can compare our recommendation to



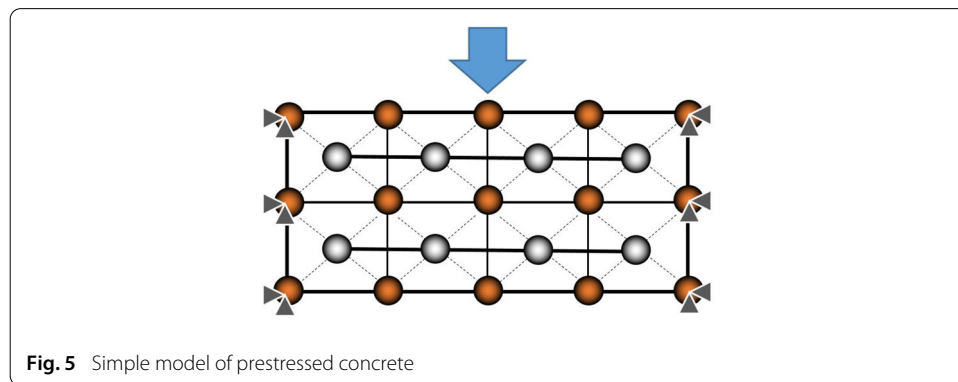
the optimal value ν_η , say, for some η . Let us use $\eta = 1$, i.e. $\kappa(\tilde{\mathbf{A}}) \leq \sqrt{2}$. Table 3 shows that the choice of $\nu = \|\mathbf{W}\|_1$ leads to smaller values than would be necessary for Theorem 1 and $\eta = 1$. Indeed, with the optimal $\nu_{\eta=1}$ a constant iteration number of 8 is attained for Prob. 2 and 3 instead of 12 and 14 respectively for the chosen ν . In Table 4, we present a short study of possible choices for ν and Prob. 4. It shows that the number of iterations decreases with increasing ν and it confirms that the modification of the (1,1)-block plays a major role for the speed of convergence of the GKB method. Note that this behavior agrees with Theorem 1. As a conclusion, $\nu = \|\mathbf{W}\|_1$ is not the choice leading to the lowest number of iterations. It can, however, be easily and cheaply computed and we see numerically, that it is an acceptable compromise for our following practical cases.

Table 3 Optimal parameter ν_η of Theorem 1, with $\eta = 1$ and condition numbers of matrices for the cylinder problems

	λ_1	λ_n	ν_η	$\kappa(\tilde{A})^2$	iter	$\kappa(M)$
Prob. 1	0.06	114.4	16.7	2.00	11	$3.9 \cdot 10^5$
Prob. 2	$7.6 \cdot 10^{-3}$	450.2	132.4	1.99	9	$2.8 \cdot 10^7$
Prob. 3	$2.8 \cdot 10^{-3}$	$1.0 \cdot 10^3$	356.2	1.99	8	$3.1 \cdot 10^8$

Table 4 Different choices for ν for cylinder problem 4, $\epsilon_{GKB}=1e-5$ and $d = 5$

ν	#iter (=: k)	$\frac{\ u-u_k\ _M}{\ u\ _M}$	$\frac{\ p-p_k\ _2}{\ p\ _2}$
$\eta = 0, \nu = 1 (M=W)$	336	$7.88 \cdot 10^{-7}$	$6.37 \cdot 10^{-6}$
1	30	$4.31 \cdot 10^{-8}$	$3.75 \cdot 10^{-8}$
17	13	$4.60 \cdot 10^{-11}$	$5.02 \cdot 10^{-11}$
133	10	$3.08 \cdot 10^{-10}$	$4.44 \cdot 10^{-10}$
357	9	$8.87 \cdot 10^{-10}$	$6.53 \cdot 10^{-10}$



Example 2: Prestressed concrete

As our second set of examples, we consider a simple model of a concrete block with embedded pretension cables. The block is clamped on its lateral faces and submitted to a constant pressure on its top face. All materials are elastic. Figure 5 presents a projected view to the 2D surface: the orange points are the concrete nodes and the gray points are the cable nodes. The cable nodes are only constrained by linear relationships with the concrete nodes, so that the displacement of the cables included in a given concrete element is a linear combination of the displacement of the concrete nodes, $\mathbf{u}_{cables} = \sum_{i=0}^4 a_i \mathbf{u}_{concrete}^x + b_i \mathbf{u}_{concrete}^y$. The vectors a and b are the barycentric coordinates of the cable node with respect to the concrete element [19]. The stiffness matrix for this model problem is only symmetric positive semi-definite caused by the rigid body motions of the steel cables. We illustrate the particular structure of the saddle point problem in Fig. 6. The (1,1)-block contains rows and columns with only zero entries. However, the non-singularity of the full system (7) is ensured, since (8) is satisfied.

Numerical experiments

We consider three test problems, called Prob. 1 to 3, with increasing discretization size (see Table 5 for their definition). Owing to the singular (1,1)-block, we rely on the augmented Lagrangian approach (9–10) and choose $\nu = \|\mathbf{W}\|_1$. The right-hand sides are provided by code_aster. The algorithm converges in only 9 iterations for each of the three test

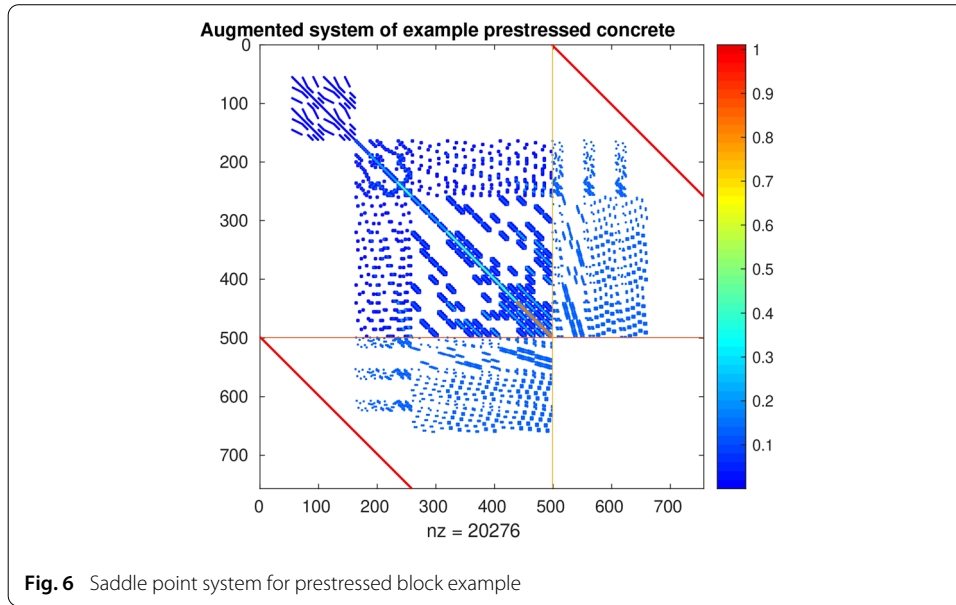


Fig. 6 Saddle point system for prestressed block example

Table 5 Example prestressed concrete: GKB convergence for $\epsilon_{GKB} = 10^{-5}$ and $d = 5$

	m	n	#iter (=k)	$\frac{\ u-u_k\ _M}{\ u\ _M}$	$\frac{\ u-u_k\ _2}{\ u\ _2}$	$\frac{\ p-p_k\ _2}{\ p\ _2}$
Prob. 1	498	258	9	$1.12 \cdot 10^{-13}$	$8.13 \cdot 10^{-14}$	$1.12 \cdot 10^{-13}$
Prob. 2	3207	1590	9	$3.49 \cdot 10^{-12}$	$3.47 \cdot 10^{-12}$	$4.78 \cdot 10^{-12}$
Prob. 3	23043	11382	9	$2.30 \cdot 10^{-11}$	$2.30 \cdot 10^{-11}$	$2.83 \cdot 10^{-11}$

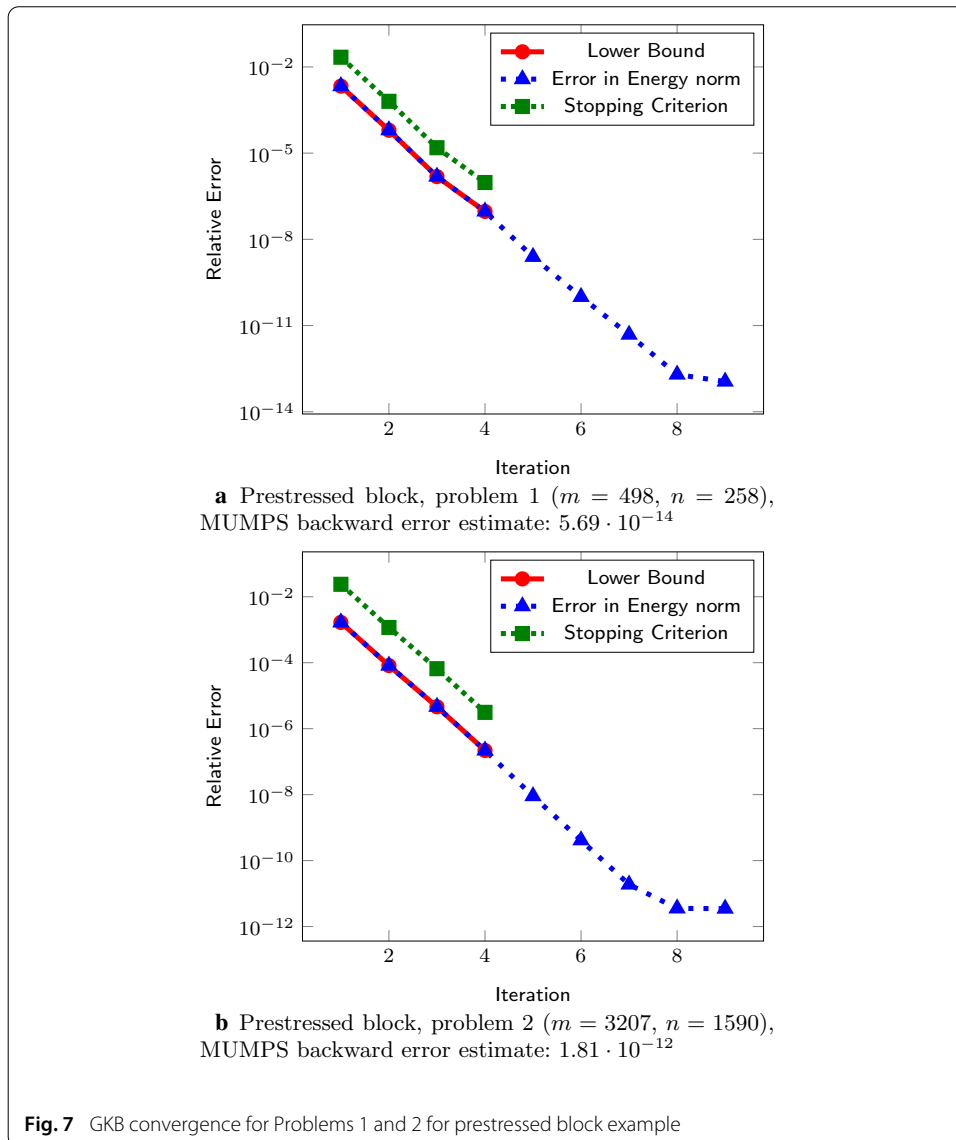
problems (see Table 5). Although the result of Theorem 1 does not apply to this case, the number of iterations stays constant and is bounded with increasing problem size. Furthermore, the energy error is already smaller than the tolerance after only 4 iterations, but we recall that the lower bound estimate for the iterate u_4 is only computed at iteration $4 + d$. The convergence of the energy error and the lower bound estimate are presented in Figs. 7 and 8. We observe that due to the degradation of the quality of the direct resolution, the convergence curves cannot reach machine precision when increasing the size of the problems.

Application to industrial example and parallel implementation

As a final example, we study a critical industrial application: the structural analysis of the reactor containment building of a nuclear power plant. This structure is designed to resist external aggression as well as internal accidents and we are concerned in the present study in the behavior of the building submitted to high pressure applied on the internal face of the building. To enhance the concrete’s strength, plenty of reinforcement cables are embedded within and a metallic shell is attached to its inner face. The model thus requires the coupling of three-dimensional elements (the concrete), two-dimensional elements (the inner shell), and one-dimensional elements (the metallic prestressing cables), Fig. 9.

Numerical experiments

The matrix and right-hand side are generated by code_aster. The discretization is illustrated in Fig. 9 and the blocks are of size $m = 283797$ and $n = 158928$. The number of



constraints is thus more than 50% of the number of physical DOF. We apply the permutations as explained above in Example 1 and obtain the matrix presented in Fig. 10. The stiffness matrix is symmetric positive semi-definite and contains rows and columns with only zero entries in the (1,1)-block. The nonsingularity of (7) is ensured by the property (8).

As in the previous examples, we scale the saddle point system (11) with the factor $\gamma = \frac{1}{2}(\min \tilde{W}_{ii} + \max \tilde{W}_{ii})$ and apply the augmented Lagrangian approach with $\nu = \|\mathbf{W}\|_1$. We apply the algorithm to the unpermuted system as it is obtained from code_aster. Since the resolution of (11) by MUMPS returns a quite large backward estimate (10^{-5}), the reference solution lacks precision. To overcome this issue, we encapsulate the MUMPS direct solver in a FGMRES solver sets with a relative tolerance of 1^{-15} .

Also for this realistic industrial application, the algorithm stops after 9 ($4 + d$) iterations and the relative errors of the final iterates \mathbf{u}_9 and \mathbf{p}_9 in the energy and 2-norm are summarized in Table 6. The lower bound estimates are presented in Fig. 11.

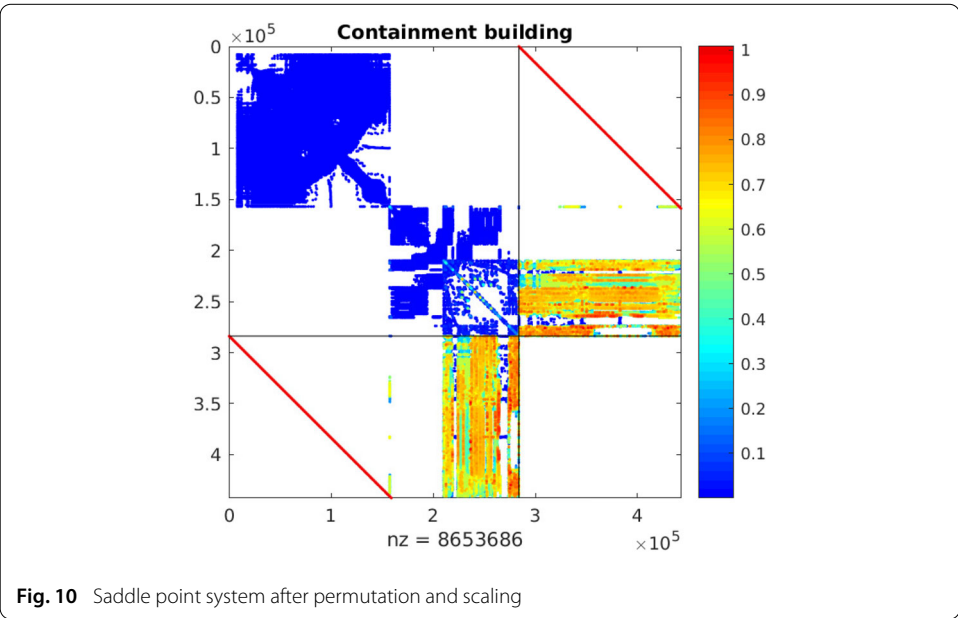
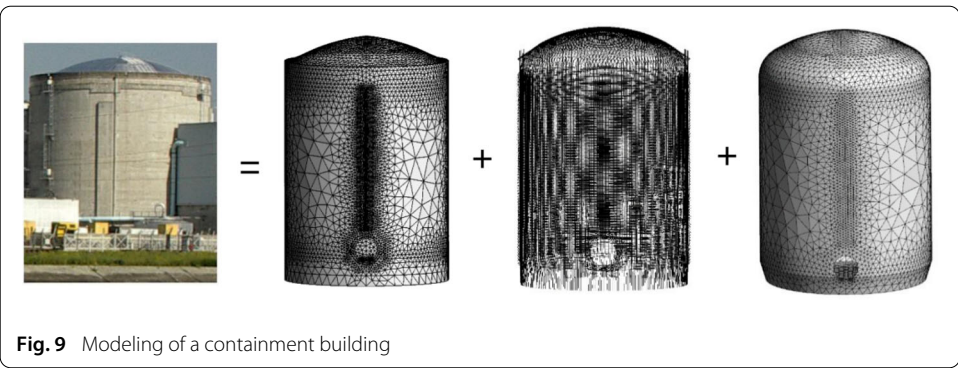
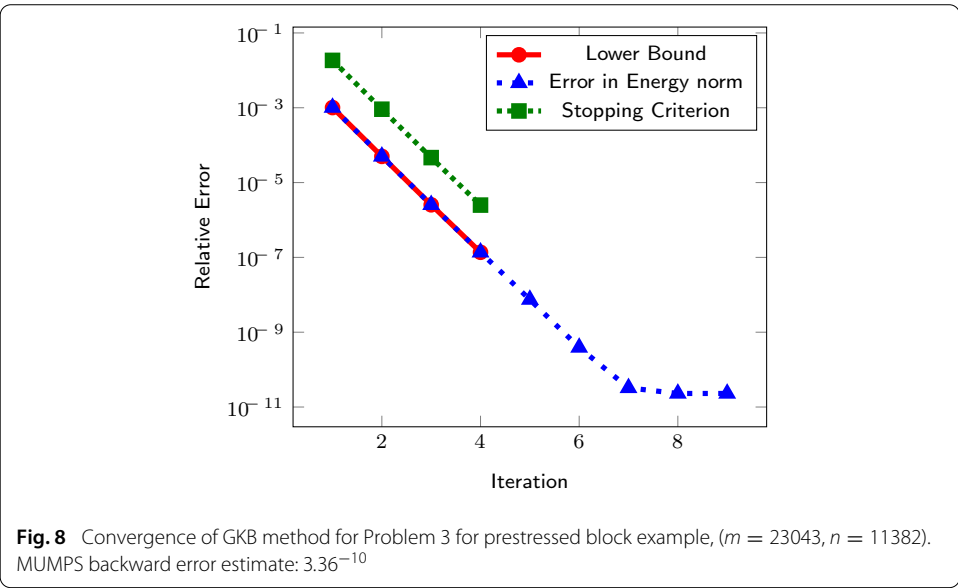
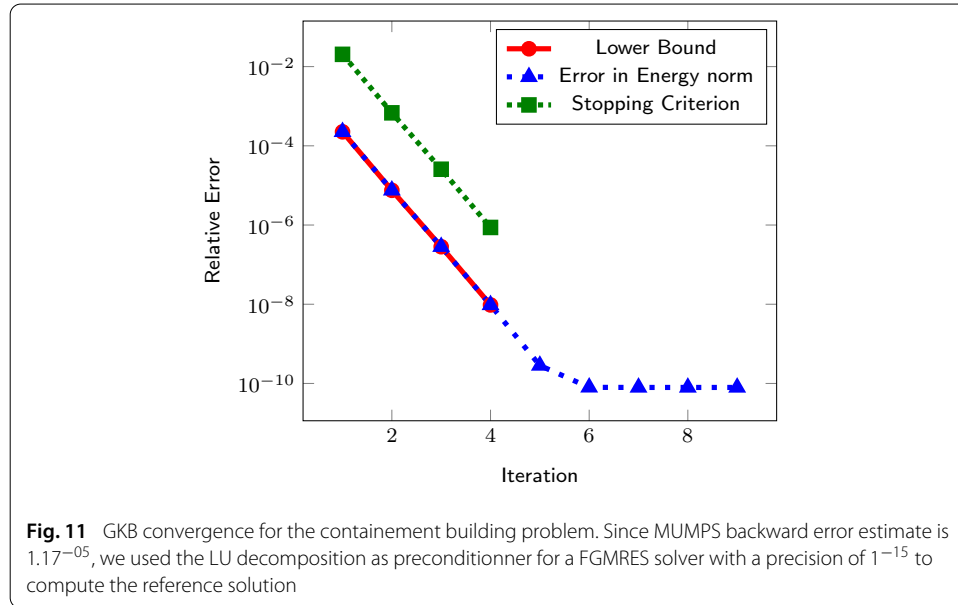


Table 6 GKB convergence for the containment building example

m	n	#iter (=: k)	$\frac{\ u-u_k\ _M}{\ u\ _M}$	$\frac{\ u-u_k\ _2}{\ u\ _2}$	$\frac{\ p-p_k\ _2}{\ p\ _2}$
283797	158928	9	$7.93 \cdot 10^{-11}$	$1.27 \cdot 10^{-10}$	$8.83 \cdot 10^{-11}$



We notice again that the final errors are remarkably small and that they are by several orders of magnitude better than the required stopping tolerance.

Strong scaling

In this section we provide an example of practical use of the GKB algorithm. We consider a prestressed concrete block as in Example 2 of sizes $m = 1538103$ and $n = 750000$ for a system with total size of 2288103 DOF.

The experiments were run on a Lenovo cluster made of bi-socket Intel Skylake (Xeon Gold 6140) computational nodes with 18 cores per socket at 2.3 GHz et with 96 GO of memory per socket. PETSc is compiled with the INTEL compiler (version 18.0.1.163) and the solver is run on a maximum of 10 cores per socket and up to 10 nodes.

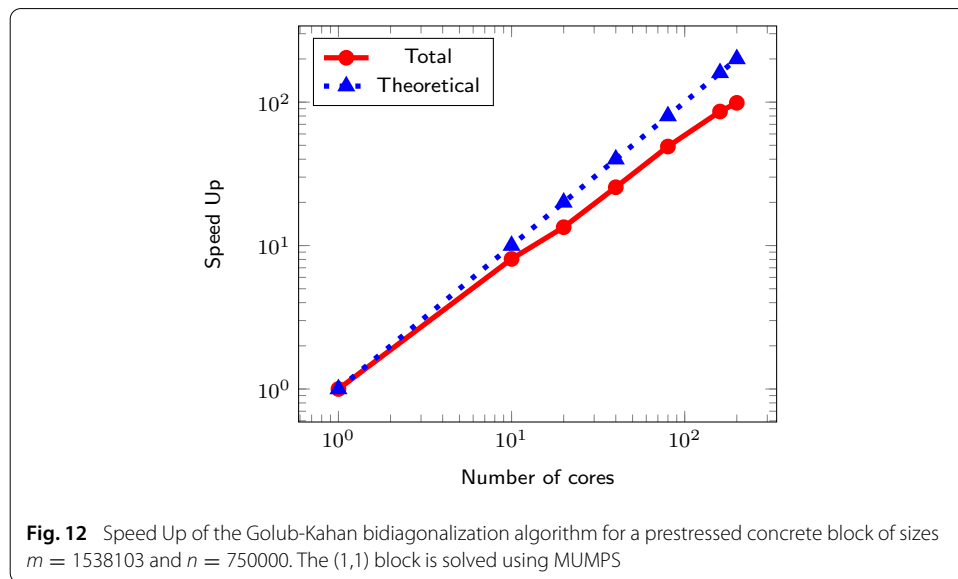
We show, respectively, the runtime and the speed-up of the solver in Table 7 and Fig. 12. Moreover, the LU decomposition of the augmented (1,1)-block and its resolution takes consistently 99% of the computational efforts. Thus the scalability of the GKB algorithm relies entirely on the scalability of the resolution of the (1,1) block. We observe that the speed-up reaches half of the ideal speed-up. Such a result is consistent to what we observed in [14].

Conclusions

In this work, we presented an algorithm based on the Golub-Kahan bidiagonalization (GKB) method and applied it to problems in structural mechanics. These problems exhibit the difficulty that multi-point constraints (MPC) are imposed on the discretized finite element formulation. We showed that the GKB algorithm converges in only a few iterations for each of the two classes of problems. In particular, we confirmed our main result of The-

Table 7 Summary of the run times of the strong scalability of the GKB algorithm for a prestressed concrete block of sizes $m = 1538103$ and $n = 750000$

#Cores	Total solver (s)	GKB time (s)
1	16,713.2	11.8
10	2071.8	8.58
20	1244.73	5.73
40	655.38	2.7
80	340.81	1.63
160	194.26	1.2
200	169.05	1.13



orem 1: The number of GKB iterations is independent of the discretization size for a given problem, whenever we choose the stabilization parameter μ appropriately. This has also been true for the example of a block of prestressed concrete, although the leading block is singular and does not satisfy the requirements of Theorem 1. The errors obtained for the final iterates are remarkably small and since the lower bound of the error at iteration k can only be computed at $k + d$, they undershoot the required tolerance by several orders of magnitude. Summarizing, the proposed algorithm presents an alternative to the more commonly used standard iterative solvers and, in particular, the ones provided currently in code_aster.

The final example of the reactor containment building is a realistic application showing the applicability of the method. However, the dimensions of the matrices are still relatively small. We built a larger problem to conduct a speedup study on the GKB algorithm. It results that the quality of the inner sub-solver drive the quality of the global speedup. For other applications, the number of degrees of freedom (DOF) might be in the order of tens of millions, and the inner direct solver Multifrontal Massively Parallel Solvers (MUMPS) will no longer perform satisfactorily. It is thus indispensable to solve the inner linear system defined by \mathbf{M} with an iterative scheme, which results in an inner-outer iterative method. The study of such algorithms will be the subject of future work. Additionally, with the GKB algorithm used as presented here and when the condition number worsens,

we observe a plateau in the convergence. Usually, this is not an issue since the precision of the solver is much smaller than the discretization error. However, to overcome this issue, the practitioner might encapsulate the inner LU with an iterative method as well as the GKB solver.

Acknowledgements

None.

Authors' contributions

The methodology was initially suggested by MA. The implementation, numerical studies, and the writing of the draft of the manuscript have been mainly carried out by CK and VD. NT provided mechanical knowledge and examples. UR supervised the study. All authors participated in the writing and approved the final manuscript.

Funding

CK and NT were partly funded by the PIA Pamsim grant no. P113165-2621644/DOS0022362/DOS0022360 of Bpifrance.

Availability of data and material

The experiments can be reproduced by using code_aster.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Cerfacs, 29 Avenue Gaspard Coriolis, 31100 Toulouse, France, ²IMSIA, UMR 9219 EDF-CNRS-CEA-ENSTA, Université Paris Saclay, 828 Boulevard des Maréchaux, 91762 Palaiseau, France, ³Libera Università Mediterranea, Casamassima, Bari, Italy, ⁴Friedrich-Alexander-Universität Erlangen-Nürnberg, Cauerstr. 11, 91058 Erlangen, Germany.

Appendix

Proof of Theorem 1

Proof Let $\sigma_1 \leq \dots \leq \sigma_n$ be the elliptic singular values defined by

$$\begin{cases} \mathbf{A}\mathbf{p}_i &= \sigma_i \mathbf{M}\mathbf{x}_i, \mathbf{x}_i^T \mathbf{M}\mathbf{x}_i = \delta_{ij} \\ \mathbf{A}^T \mathbf{x}_i &= \sigma_i \frac{1}{v} \mathbf{p}_i, \frac{1}{v} \mathbf{p}_i^T \mathbf{p}_j = \delta_{ij} \\ \sigma_i &= \mathbf{x}_i^T \mathbf{A}\mathbf{p}_i. \end{cases}$$

It follows

$$v\mathbf{A}^T \mathbf{M}^{-1} \mathbf{A}\mathbf{p}_i = \sigma_i^2 \mathbf{p}_i.$$

Thus $\mu_i = \sigma_i^2$ are the eigenvalues of

$$v\mathbf{A}^T (\mathbf{W} + v\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A}.$$

With the Sherman-Morrison formula, we obtain

$$v\mathbf{A}^T (\mathbf{W} + v\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A} = v\mathbf{A}^T \mathbf{W}^{-1} \mathbf{A} (\mathbf{I} + v\mathbf{A}^T \mathbf{W}^{-1} \mathbf{A})^{-1}$$

Let $\lambda_1 \leq \dots \leq \lambda_n$ be the eigenvalues of $\mathbf{A}^T \mathbf{W}^{-1} \mathbf{A}$. Then

$$\mu_i = \frac{v\lambda_i}{1 + v\lambda_i} \quad \forall i.$$

We obtain for the condition number of $\tilde{\mathbf{A}} = \sqrt{v}\mathbf{M}^{-\frac{1}{2}} \mathbf{A} = v(\mathbf{W} + v\mathbf{A}\mathbf{A}^T)^{-\frac{1}{2}} \mathbf{A}$

$$\kappa^2(\tilde{\mathbf{A}}) = \frac{\mu_{max}}{\mu_{min}} = \frac{v\lambda_n}{1 + v\lambda_n} \frac{1 + v\lambda_1}{v\lambda_1}$$

It follows that if $\nu \geq \frac{1}{\eta\lambda_1} - \frac{1+\eta}{\eta\lambda_n}$, then $\kappa(\tilde{\mathbf{A}}) \leq \sqrt{1 + \eta}$. □

Received: 30 April 2020 Accepted: 21 October 2020

Published online: 17 November 2020

References

1. Smith M. ABAQUS/Standard User's Manual, Version 6.14. 2017
2. Dhondt G. The Finite Element Method for Three-Dimensional Thermomechanical Applications. 2004. <https://doi.org/10.1002/0470021217>.
3. Bernardi C, Maday Y, Patera AT. A new nonconforming approach to domain decomposition: the mortar element method. *Nonlinear Partial Eq Appl*. 1994;25:13–51.
4. Elman HC, Silvester DJ, Wathen AJ. Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics. Oxford: Numerical mathematics and scientific computation. Oxford University Press; 2014. <https://doi.org/10.1093/acprof:oso/9780199678792.00110.1093/acprof:oso/9780199678792.001.0001>.
5. Benzi M, Golub GH, Liesen J. Numerical solution of saddle point problems. *Acta Numer*. 2005;14:1–137. <https://doi.org/10.1017/S0962492904000212>.
6. Saad Y. Iterative Methods for Sparse Linear Systems, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia 2003. <https://doi.org/10.1137/1.9780898718003>
7. Abel JF, Shephard MS. An algorithm for multipoint constraints in finite element analysis. *Int J Numer Methods Eng*. 1979;14(3):464–7. <https://doi.org/10.1002/nme.1620140312>.
8. Jendele L, Cervenka J. On the solution of multi-point constraints—application to FE analysis of reinforced concrete structures. *Comput Struct*. 2009;87(15):970–80. <https://doi.org/10.1016/j.compstruc.2008.04.018> Computational Structures Technology.
9. Chow E, Manteuffel TA, Tong C, Wallin BK. Algebraic elimination of slide surface constraints in implicit structural analysis. *Int J Numer Methods Eng*. 2003;57(8):1129–44. <https://doi.org/10.1002/nme.720>.
10. Saint-Georges P, Notay Y, Warzée G. Efficient iterative solution of constrained finite element analyses. *Comput Methods Appl Mech Eng*. 1998;160(1):101–14. [https://doi.org/10.1016/S0045-7825\(97\)00286-7](https://doi.org/10.1016/S0045-7825(97)00286-7).
11. Golub G, Kahan W. Calculating the singular values and pseudo-inverse of a matrix. *J Soc Indu Appl Math Series B*. 1965;2(2):205–24. <https://doi.org/10.1137/0702016>.
12. Golub GH, van Loan CF. Matrix Computations, 4th edn. JHU Press, Baltimore (2013). <http://www.cs.cornell.edu/cv/GVL4/golubandvanloan.htm>
13. Arioli M. Generalized Golub-Kahan bidiagonalization and stopping criteria. *SIAM J Matrix Anal Appl*. 2013;34(2):571–92. <https://doi.org/10.1137/120866543>.
14. Kruse C, Sosonkina M, Arioli M, Tardieu N, Rude U. Parallel performance of an iterative solver based on the Golub-Kahan bidiagonalization. In: Wyrzykowski R, Deelman E, Dongarra J, Karczewski K, editors. Parallel processing and applied mathematics, lecture notes in computer science, vol. 12043. Cham: Springer; 2020. p. 104–16. https://doi.org/10.1007/978-3-030-43229-4_10.
15. Balay S, Abhyankar S, Adams MF, Brown J, Brune P, Buschelman K, Dalcin L, Dener A, Eijkhout V, Gropp WD, Karpeyev D, Kaushik D, Knepley MG, May DA, McInnes LC, Mills RT, Munson T, Rupp K, Sanan P, Smith BF, Zampini S, Zhang H, Zhang H. PETSc users manual. Technical Report ANL-95/11 - Revision 3.11, Argonne National Laboratory. 2019 <http://www.mcs.anl.gov/petsc>
16. Balay S, Abhyankar S, Adams MF, Brown J, Brune P, Buschelman K, Dalcin L, Dener A, Eijkhout V, Gropp WD, Karpeyev D, Kaushik D, Knepley MG, May DA, McInnes LC, Mills RT, Munson T, Rupp K, Sanan P, Smith BF, Zampini S, Zhang H, Zhang H. PETSc Web page. <http://www.mcs.anl.gov/petsc> 2019
17. Balay S, Gropp WD, McInnes LC, Smith BF. Efficient management of parallelism in object oriented numerical software libraries. In: Arge E, Bruaset AM, Langtangen HP, editors. Modern software tools in scientific computing. Basel: Birkhäuser Press; 1997. p. 163–202. https://doi.org/10.1007/978-1-4612-1986-6_8.
18. Boffi D, Brezzi F, Fortin M. Mixed Finite Element Methods and Applications. Berlin: Springer; 2013. <https://doi.org/10.1007/978-3-642-36519-5>.
19. Pellet J. Conditions de liaison de corps solide. *Tech Rep*. 2011a;R3(03):02.
20. Golub GH, Greif C. On solving block-structured indefinite linear systems. *SIAM J Sci Comput*. 2003;24(6):2076–92. <https://doi.org/10.1137/S1064827500375096>.
21. Orban D, Arioli M. Iterative Solution of Symmetric Quasi-Definite Linear Systems. *SIAM Spotlights*. Society for Industrial and Applied Mathematics, Philadelphia (2017). <https://doi.org/10.1137/1.9781611974737>
22. Paige CC, Saunders MA. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans Math Softw*. 1982;8(1):43–71. <https://doi.org/10.1145/355984.355989>.
23. Pellet J. Dualisation des conditions aux limites. *Tech Rep*. 2011b;R3(03):01.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.