



MOX-Report No. 46/2026

**A Scalable Deflated Conjugate Gradient Solver for the
Time-Dependent Pseudo-Stress Stokes Problem**

Cancrini, A.; Ciaramella, G.; Antonietti, P.F.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it

<https://mox.polimi.it>

A SCALABLE DEFLATED CONJUGATE GRADIENT SOLVER FOR THE TIME-DEPENDENT PSEUDO-STRESS STOKES PROBLEM

ALESSANDRA CANCRINI*, GABRIELE CIARAMELLA*, AND PAOLA F. ANTONIETTI*

Abstract. We propose a novel iterative solution framework for the unsteady Stokes equations in the pseudo-stress formulation. When solving this class of problems by using implicit time-integration schemes, standard solvers suffer from deteriorating convergence properties for small time steps, independently of the chosen space discretisation method. This is due to the singular modes of the dev-dev operator. For this reason, we introduce a computational framework obtained by combining a deflated Conjugate Gradient method with a W-cycle multigrid scheme that employs a Restricted Additive Schwarz smoother. The key point is to choose the deflation subspace so that the inner system to be solved within a deflated Conjugate Gradient scheme corresponds to a Laplace problem defined on the singular modes of the original dev-dev operator. This results to be independent of the spatial discretisation method and allows one to use efficient multigrid iterative solvers. Numerical experiments show that the proposed strategy significantly accelerates the Conjugate Gradient convergence and provides stable performance with respect to the time step, confirming its robustness for solving linear systems in the pseudo-stress framework.

Key words. unsteady Stokes problem, pseudo-stress formulation, continuous and discontinuous finite element methods, W-cycle multigrid scheme, deflated and flexible Conjugate Gradient methods

MSC codes. 65N30, 65N55, 65F45, 65F50

1. Introduction. The numerical approximation of incompressible viscous flows is a central problem in computational fluid dynamics. In this context, the stress–velocity–pressure formulation of the unsteady Stokes equations has gained considerable interest, particularly due to its relevance for non-Newtonian fluid flow models (see e.g. [10, 33]) and coupled interface problems, where an accurate representation of the stress tensor plays a crucial role [29, 38]. Although the stress can be reconstructed a posteriori in the velocity–pressure formulation by differentiating the velocity field, this approach might compromise the overall accuracy. It is also worth noting that the stress–velocity–pressure formulation introduces an additional difficulty, namely the enforcement of the stress tensor’s symmetry at the discrete level [7, 8]. A possible strategy to circumvent this issue is to introduce the concept of pseudo-stress, which is a nonsymmetric tensor [17, 19]. Motivated by these features, different works have considered the unsteady Stokes problem written in terms of the pseudo-stress variable only, employing DG methods on polytopal grids (see [2, 20] and references therein).

The present work focuses on the development of a novel computational framework for the solution of discrete incompressible viscous flows in pseudo-stress form. The final goal is to develop a scalable solver that is robust with respect to both the time step Δt and the mesh size h . As a first step, in this work we focus on robustness with respect to Δt and investigate how the performance of classical iterative methods is affected by its variation. Numerical experiments show that the Conjugate Gradient (CG) method [31, 40] suffers from a severe deterioration of convergence as Δt decreases. This behavior is due to the conditioning of the system matrix [40, 42], which scales as $1/\Delta t$. Therefore, smaller time steps produce increasingly ill-conditioned linear systems, and thus require a larger number of iterations to reduce the relative residual below a prescribed tolerance [3]. It is well known that, when the condition number becomes large, improving the spectral properties of the system matrix is often essential to achieve scalability. This issue arises independently of the spatial discretisation employed. Indeed, when implicit time integration schemes, such as θ -methods or Backward Differentiation Formulae (BDF), are applied to the pseudo-stress formulation of the unsteady Stokes equations, the resulting algebraic problem is symmetric and

*MOX, Dipartimento di Matematica, Politecnico di Milano, Piazza Leonardo da Vinci 32, Milan, 20133, Italy.
e-mail: alessandra.cancrini@polimi.it, gabriele.ciaramezza@polimi.it, paola.antonietti@polimi.it

positive definite. Nevertheless, already at the semi-discrete level, the spectrum of the operator exhibits a dependence on $1/\Delta t$, leading to increasingly ill-conditioned systems as the time step decreases. As a consequence, the convergence of Krylov subspace methods deteriorates significantly for small values of Δt , regardless of the finite-dimensional approximation adopted in space. Similar behavior has been observed and investigated in our previous work [3].

To address this issue, we propose a novel solution framework based on a deflated Conjugate Gradient method [25, 32, 36] (see also [26, 28, 41]). Deflation techniques are designed to accelerate Krylov subspace iterations by removing from the spectrum those components responsible for slow convergence. A central aspect of the deflated CG method is the choice of the deflation subspace $\mathbf{S} \subset \mathbb{R}^n$, since this choice fully determines the effectiveness of the deflation. For the problem under consideration, the dependence on $1/\Delta t$ can be removed by choosing a subspace \mathbf{S} that contains all singular modes of the deviatoric–deviatoric operator. Once \mathbf{S} is fixed, the resulting inner operator, namely the projection of the original problem to \mathbf{S} , must be efficiently applied to a vector, in order to preserve the overall performance of the iterative scheme. This application requires the solution of a system that arises from the restriction of the original problem to the subspace \mathbf{S} . We will see that, for a suitably chosen \mathbf{S} , which contains singular modes of the deviatoric–deviatoric operator, this inner system coincides with a Laplace equation, opening the possibility of employing fast and robust iterative solvers. An efficient implementation of the deflation strategy also requires the computation of suitable projection operators. These projections involve the solution of the original problem restricted to the deflation subspace \mathbf{S} and therefore represent a crucial component of the overall computational cost. One of the main theoretical contributions of this work is to show that such a restricted operator is equivalent to a Laplace problem both at the continuous and at the discrete level. Furthermore, in the discrete setting, the restricted problem inherits exactly the spatial discretisation adopted for the original pseudo-stress formulation. Therefore, if a continuous Galerkin (CG) or a discontinuous Galerkin (DG) discretisation is employed for the original problem, then the corresponding restricted operator becomes a Laplace problem discretised by the same CG or DG method. This property makes the proposed framework completely independent of the chosen spatial approximation and therefore applicable to a broad class of discretisation techniques.

The identification of the restricted problem with a Laplace operator is particularly advantageous, since efficient multigrid solvers for elliptic problems are well established in the literature [11, 12, 13] (see [4, 5, 6, 30] for multigrid applications related to DG methods). For this reason, the proposed framework combines an outer deflated CG iteration with an inner multigrid solver dedicated to the restricted problem. As smoother for the multigrid scheme, we employ a Restricted Additive Schwarz (RAS) iteration [16, 24, 27], which is well suited to block-structured operators and meshes of general topology. The combination of a W-cycle multigrid strategy with a RAS smoother yields a robust and scalable iterative scheme for the inner system, improving the overall performance of the deflated CG approach. To further reduce the computational cost, the inner multigrid solver is not required to solve the restricted problem to a fixed accuracy. Instead, its stopping criterion is dynamically adjusted according to the residual of the outer CG iteration [32]. This strategy substantially decreases the total computational effort while preserving the convergence properties of the overall algorithm. Since the inner system is solved only approximately, its influence on the outer iteration may vary between steps. To prevent possible degradation of convergence behavior caused by these variations, the deflated CG approach can be enhanced with a flexible variant of the CG method [9, 37, 40], which provides greater stability when the system operator changes during iterations. The numerical experiments presented in this work demonstrate that the proposed methodology significantly improves the convergence behavior of classical Krylov solvers and exhibits robustness with respect to the time step. These results confirm the effectiveness of

the proposed deflation strategy and highlight its potential as a building block towards fully scalable solvers for the pseudo-stress formulation of the unsteady Stokes equations.

The structure of the paper is as follows. Section 2 presents the pseudo-stress weak formulation of the unsteady Stokes problem, the associated time semi-discrete formulation, and the reduced system arising from it. Section 3 is devoted to the fully-discrete setting: we introduce a general spatial discretisation, derive the corresponding discrete reduced system, and present examples of spatial discretisation schemes showing how the abstract theory can be applied in practice. In Section 4 we develop a deflated CG strategy to stabilize the iterative solution of the fully discrete system as $\Delta t \rightarrow 0$, and we propose a multigrid approach with RAS smoothing to efficiently solve the inner system arising from the deflation procedure. We also present a flexible CG variant for solving the system, which improves convergence when the operator changes during iterations. Numerical tests are reported in Section 5, where we assess the performance of the proposed approach by means of experiments in both two- and three-dimensional settings, employing the spatial discretisation based on a discontinuous Galerkin method on polytopal meshes. Finally, in Section 6 we draw some conclusions and an outlook of possible extension of this work.

2. Variational formulation and time discretisation. This section is devoted to the variational setting of the problem. We first introduce the notation and the pseudo-stress weak formulation, then derive a time semi-discrete formulation, and finally present the reduced system together with its Laplacian structure.

2.1. Notation. Let $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, be a bounded open domain with Lipschitz boundary $\partial\Omega$. In what follows, for $X \subseteq \Omega$, the notation $\mathbf{L}^2(X)$ is adopted in place of $[L^2(X)]^d$, and $\mathbb{L}^2(X)$ in place of $[L^2(X)]^{d \times d}$. The scalar product in $L^2(X)$ is denoted by $(\cdot, \cdot)_X$, with the associated norm $\|\cdot\|_X$. Similarly, the Sobolev spaces $\mathbf{H}^\ell(X)$ are defined as $[\mathbf{H}^\ell(X)]^d$, with $\ell \geq 0$, equipped with the norm $\|\cdot\|_{\ell, X}$, assuming conventionally that $\mathbf{H}^0(X) \equiv \mathbf{L}^2(X)$. In addition, we will use $\mathbf{H}(\text{div}, X)$ to denote the space of $\mathbf{L}^2(X)$ vector fields with square-integrable divergence. Then, the notation $\mathbb{H}(\text{div}, X)$ is used for the space of tensor fields with rows belonging to $\mathbf{H}(\text{div}, X)$. Finally, for $\Gamma \subseteq \partial\Omega$ closed subset of $\partial\Omega$, we consider the space $H^{\frac{1}{2}}(\Gamma) = \{v \in L^2(\Gamma) \mid \exists u \in H^1(\Omega) : u|_\Gamma = v\}$ and its dual space $H^{-\frac{1}{2}}(\Gamma)$. The duality product between $v \in H^{\frac{1}{2}}(\Gamma)$ and $w \in H^{-\frac{1}{2}}(\Gamma)$ is denoted by $\langle v, w \rangle_\Gamma$.

2.2. Pseudo-stress weak formulation. We focus on the time-dependent Stokes problem, which models incompressible viscous free flows, and formulate it in terms of a pseudo-stress unknown rather than its classical form. The derivation of the pseudo-stress formulation for the unsteady Stokes problem is presented in [2]. The pseudo-stress variable is defined as $\boldsymbol{\sigma}(\mathbf{u}, p) = \mu \nabla \mathbf{u} - p \mathbb{I}_d$, where \mathbf{u} is the flow velocity, p its pressure, and \mathbb{I}_d the identity matrix in $\mathbb{R}^{d \times d}$. Let $\text{dev}(\boldsymbol{\tau}) = \boldsymbol{\tau} - \frac{1}{d} \text{tr}(\boldsymbol{\tau}) \mathbb{I}_d$ be the deviatoric operator, where $\text{tr}(\cdot)$ is the trace operator. The boundary of Ω is partitioned as $\Gamma_D \cup \Gamma_N = \partial\Omega$, with $\Gamma_D \cap \Gamma_N = \emptyset$. For simplicity, we assume both $|\Gamma_D| > 0$ and $|\Gamma_N| > 0$, with $|\cdot|$ denoting the Hausdorff measure. To strongly enforce the essential traction condition on Γ_N , we define the subspace

$$V = \mathbb{H}_{0, \Gamma_N}(\text{div}, \Omega) = \{\boldsymbol{\eta} \in \mathbb{H}(\text{div}, \Omega) \mid \langle \boldsymbol{\eta} \mathbf{n}, \mathbf{v} \rangle_{\partial\Omega} = 0 \quad \forall \mathbf{v} \in \mathbf{H}_{0, \Gamma_D}^1(\Omega)\},$$

where $\mathbf{H}_{0, \Gamma_D}^1(\Omega) = \{\mathbf{v} \in \mathbf{H}^1(\Omega)^d \mid \mathbf{v} = \mathbf{0} \text{ on } \Gamma_D\}$. Then, the corresponding weak formulation reads as: for any $t \in (0, T]$, find $\boldsymbol{\sigma}(t) \in \mathbb{H}_{0, \Gamma_N}(\text{div}, \Omega)$ such that

$$(2.1) \quad \mathcal{M}(\partial_t \boldsymbol{\sigma}, \boldsymbol{\tau}) + \mathcal{A}(\boldsymbol{\sigma}, \boldsymbol{\tau}) = F(\boldsymbol{\tau}) \quad \forall \boldsymbol{\tau} \in \mathbb{H}_{0, \Gamma_N}(\text{div}, \Omega),$$

where for any $\sigma, \tau \in \mathbb{H}_{0,\Gamma_N}(\operatorname{div}, \Omega)$ we have defined

$$\begin{aligned} \mathcal{M}(\sigma, \tau) &= (\mu^{-1} \mathbf{dev}(\sigma), \mathbf{dev}(\tau))_{\Omega}, & \mathcal{A}(\sigma, \tau) &= (\nabla \cdot \sigma, \nabla \cdot \tau)_{\Omega}, \\ F(\tau) &= (F, \tau)_{\Omega} + \langle g_D, \tau \mathbf{n} \rangle_{\partial\Omega}, \end{aligned}$$

with $\mu > 0$ the fluid viscosity, and $T > 0$ the final observation time. We assume that F is sufficiently regular, $g_D(t) \in \mathbf{H}^{\frac{1}{2}}(\partial\Omega)$, and that the initial condition $\mathbf{dev}(\sigma)(\cdot, t=0) = \sigma_0 \in \mathbb{L}^2(\Omega)$. We point out that, owing to the definition of the stress space $\mathbb{H}_{0,\Gamma_N}(\operatorname{div}, \Omega)$, the duality pairing on the right-hand side of (2.1) is taken over the whole boundary $\partial\Omega$, and not only over Γ_D . To ensure that this term is well defined, following [2], we assume that, for all t , the Dirichlet datum $g_D = g_D(t)$ is the trace of a function in $\mathbf{H}^1(\Omega)$, which, with a slight abuse of notation we still denote by g_D , so that $g_D(t) \in \mathbf{H}^{\frac{1}{2}}(\partial\Omega)$. This allows us to write the duality over $\partial\Omega$ in (2.1). For further details, the reader is referred to [2].

2.3. Time semi-discrete formulation. We now introduce a generic semi-discrete formulation of (2.1), resulting from a discretisation in time. We restrict our attention to two families of implicit methods: the θ -method and the BDF schemes. Let $\alpha = \alpha(\Delta t)$ denote the coefficient associated with the chosen time discretisation (e.g. $\alpha = \Delta t$ for implicit Euler, $\alpha = \Delta t/2$ for Crank–Nicolson). Then, given the initial datum σ^0 , the semi-discrete problem reads: for any $n = 1, \dots, N_T$ find $\sigma^n \in \mathbb{H}_{0,\Gamma_N}(\operatorname{div}, \Omega)$ such that

$$(2.2) \quad \mathcal{M}(\sigma^n, \tau) + \alpha \mathcal{A}(\sigma^n, \tau) = F_n^*(\tau) \quad \forall \tau \in \mathbb{H}_{0,\Gamma_N}(\operatorname{div}, \Omega),$$

where superscript n means the approximation/evaluation of the given quantity at time $t_n = n\Delta t$, $n = 1, \dots, N_T$, while $F_n^*(\cdot)$ denotes the load functional associated with the chosen implicit time discretisation. In particular, $F_n^*(\cdot)$ depends on previous time steps: only on $n-1$ in a θ -method, and on multiple past steps in BDF schemes.

Let us rewrite the forms $\mathcal{M}(\cdot, \cdot)$, $\mathcal{A}(\cdot, \cdot)$ and $F_n^*(\cdot)$ in a componentwise form:

$$(2.3) \quad \begin{aligned} \mathcal{M}(\sigma, \tau) &= \mu^{-1} \left[\sum_{k=1}^d \left(1 - \frac{1}{d}\right) m(\sigma_{kk}, \tau_{kk}) + \sum_{\substack{k,i=1 \\ k \neq i}}^d \left(-\frac{1}{d} m(\sigma_{kk}, \tau_{ii}) + m(\sigma_{ki}, \tau_{ki}) \right) \right], \\ \mathcal{A}(\sigma, \tau) &= \sum_{k,i,j=1}^d a_{ij}(\sigma_{ki}, \tau_{kj}), & F_n^*(\tau) &= \sum_{i,j=1}^d f_{ij}(\tau_{ij}). \end{aligned}$$

In this way, the formulation is kept at an abstract level, and the associated bilinear and linear forms $m(\cdot, \cdot)$, $a_{ij}(\cdot, \cdot)$ and $f_{ij}(\cdot)$ are defined at the continuous spatial level as follows:

$$m(\sigma, \tau) = \int_{\Omega} \sigma \tau \, dx, \quad a_{ij}(\sigma, \tau) = \int_{\Omega} \partial_i \sigma \partial_j \tau \, dx, \quad f_{ij}(\tau) = \int_{\Omega} F_{ij}^* \tau \, dx.$$

2.4. Reduced system and Laplacian structure. Since the bilinear form $\mathcal{M}(\cdot, \cdot)$ has a nontrivial kernel, there exist nonzero functions in V that are not detected by $\mathcal{M}(\cdot, \cdot)$. To properly treat these singular modes, we introduce a *reduced system* obtained by restricting (2.2) to the kernel of $\mathcal{M}(\cdot, \cdot)$ defined as

$$(2.4) \quad \begin{aligned} \mathbf{S} = \ker(\mathcal{M}) &= \left\{ \sigma \in V \mid \mathcal{M}(\sigma, \tau) = 0 \quad \forall \tau \in V \right\} \\ &= \{ \sigma \in V \mid \sigma_{ii} = \sigma_{jj} \quad \forall i, j = 1, \dots, d, \text{ and } \sigma_{ij} = 0 \text{ for } i \neq j \}. \end{aligned}$$

Both bilinear forms $\mathcal{M}(\cdot, \cdot)$ and $\mathcal{A}(\cdot, \cdot)$ induce seminorms on the underlying space and are therefore only positive semi-definite, i.e., they each admit a nontrivial kernel. As shown

in [2], the kernels of the two forms intersect only trivially, i.e., $\ker(\mathcal{M}) \cap \ker(\mathcal{A}) = \{0\}$, which guarantees that the combined operator is positive definite, and hence induces a norm with respect to which coercivity holds [2]. On the subspace \mathbf{S} , the contribution of $\mathcal{M}(\cdot, \cdot)$ vanishes, while $\mathcal{A}(\cdot, \cdot)$ becomes positive definite. On the complementary subspace, $\mathcal{M}(\cdot, \cdot)$ is positive definite whereas $\mathcal{A}(\cdot, \cdot)$ remains positive semi-definite. The same property persists at the discrete level. This structure also explains the $O(1/\Delta t)$ growth of the condition number of the combined operator: the singular modes of $\mathcal{M}(\cdot, \cdot)$ force the smallest eigenvalue to scale as $O(\Delta t)$, while the largest remains $O(1)$. Hence, the condition number of the corresponding system matrix grows as $O(1/\Delta t)$.

DEFINITION 2.1 (Continuous Reduced System). *The reduced system, obtained by restricting the time semi-discrete problem (2.2) to the subspace \mathbf{S} defined in (2.4), reads as follows: given σ^0 , for any $n = 1, \dots, N_T$ find $\sigma^n \in \mathbf{S}$ such that*

$$\mathcal{M}(\sigma^n, \tau) + \alpha \mathcal{A}(\sigma^n, \tau) = F_n^*(\tau) \quad \forall \tau \in \mathbf{S}.$$

Notice that, for any $\sigma \in \mathbf{S} \subset \mathbb{H}(\operatorname{div}, \Omega)$, the corresponding i th row of σ has only one nonzero component that is σ_{ii} . Thus, recalling that $\mathbb{H}(\operatorname{div}, \Omega)$ is the space of tensor fields with rows belong to $\mathbf{H}(\operatorname{div}, \Omega)$, the diagonal components $\sigma_{ii} = \sigma_{jj} = \sigma$ (for all i and j) belong to $H^1(\Omega)$ and satisfy $\langle \sigma \mathbf{n}, \mathbf{v} \rangle_{\partial\Omega} = 0 \quad \forall \mathbf{v} \in \mathbf{H}_{0,\Gamma_D}^1(\Omega)$. Hence, $\sigma_{ii} = \sigma_{jj} = \sigma \in H_{0,\Gamma_N}^1(\Omega)$.

As a result, the reduced system is a Laplace problem, as stated in the following theorem.

THEOREM 2.2 (Laplacian Structure of the Continuous Reduced System). *Let (2.2) be the semi-discrete formulation of the pseudo-stress Stokes problem in dimension d . Then, the reduced system in Definition 2.1 corresponds to a Laplace problem, i.e., it takes the form: find $\sigma \in \mathbf{S}$ (equivalently $\sigma \in H_{0,\Gamma_N}^1(\Omega)$) such that*

$$\alpha \sum_{i=1}^d a_{ii}(\sigma, \tau) = \sum_{i=1}^d f_{ii}(\tau) \quad \forall \tau \in H_{0,\Gamma_N}^1(\Omega).$$

Proof. Recalling the representations (2.3), the reduced system reads as follows: find $\sigma \in H_{0,\Gamma_N}^1(\Omega)$ such that

$$\mu^{-1} \left[d \left(1 - \frac{1}{d} \right) m(\sigma, \tau) - \frac{d(d-1)}{d} m(\sigma, \tau) \right] + \alpha \sum_{i=1}^d a_{ii}(\sigma, \tau) = \sum_{i=1}^d f_{ii}(\tau) \quad \forall \tau \in H_{0,\Gamma_N}^1(\Omega),$$

so we have $\alpha \sum_{i=1}^d a_{ii}(\sigma, \tau) = \sum_{i=1}^d f_{ii}(\tau)$. Hence, the reduced system corresponds to the standard variational formulation of a Laplace problem. \square

3. Fully-discrete formulation. We can now introduce a fully-discrete formulation by using a generic spatial discretisation: let \mathbf{V}_h be a suitable finite-dimensional space approximating \mathbf{V} . The discrete space \mathbf{V}_h is characterized by a set of local degrees of freedom associated with mesh elements or control volumes, and by a polynomial degree $p_\kappa \geq 1$ (possibly varying from element to element). Then, the fully-discrete approximation to (2.1) reads as follows: for any $n = 1, \dots, N_T$ find $\sigma_h^n \in \mathbf{V}_h$ such that

$$(3.1) \quad \begin{cases} \mathcal{M}^h(\sigma_h^n, \tau_h) + \alpha \mathcal{A}^h(\sigma_h^n, \tau_h) = F_n^{*,h}(\tau_h) & \forall \tau_h \in \mathbf{V}_h, \\ (\sigma_h^0, \tau_h) = (\sigma_{0,h}, \tau_h) & \forall \tau_h \in \mathbf{V}_h, \end{cases}$$

where $\sigma_{0,h}$ is the L^2 -projection of σ_0 onto \mathbf{V}_h . To obtain the corresponding algebraic formulation, we introduce a basis $\{\phi_i, i = 1, \dots, N_h\}$, with $N_h = \dim(\mathbf{V}_h)$, for the space \mathbf{V}_h

and express σ_h as a linear combination of the basis functions ϕ_i . We denote by M (resp. A) the matrix representation of the bilinear form $\mathcal{M}^h(\cdot, \cdot)$ (resp. $\mathcal{A}^h(\cdot, \cdot)$), and by f^* the vector representation of the linear functional $F_n^{*,h}(\cdot)$. Although f^* depends on n , we will focus on the solution of the linear system at a fixed time step and therefore omit this dependence. The algebraic formulation of (3.1) reduces to: given σ_h^0 , for any $n = 1, \dots, N_T$ find σ_h^n such that

$$(3.2) \quad A^* \sigma_h^n = f^*.$$

Here, we have introduced $A^* = M + \alpha A$, which corresponds to $\mathcal{A}^{*,h}(\cdot, \cdot) = \mathcal{M}^h(\cdot, \cdot) + \alpha \mathcal{A}^h(\cdot, \cdot)$, defining a coercive bilinear form.

3.1. Discrete reduced system. After establishing the Laplacian structure of the continuous reduced system in Section 2.4, we now show that an analogous result holds for the fully-discrete formulation. In particular, in the discrete setting, the forms $\mathcal{M}(\cdot, \cdot)$, $\mathcal{A}(\cdot, \cdot)$ and $F_n^*(\cdot)$ are replaced by their discrete counterparts $\mathcal{M}^h(\cdot, \cdot)$, $\mathcal{A}^h(\cdot, \cdot)$ and $F_n^{*,h}(\cdot)$. Accordingly, we define the associated discrete bilinear and linear forms by $m^h(\cdot, \cdot)$, $a_{ij}^h(\cdot, \cdot)$, and $f_{ij}^h(\cdot)$, which depend on the chosen spatial discretisation method.

DEFINITION 3.1 (Discrete Reduced System). *The discrete reduced system, obtained by restricting the fully-discrete problem (3.1) to the subspace $\mathbf{S}^h = \ker(\mathcal{M}^h)$, reads as follows: given σ_h^0 , for any $n = 1, \dots, N_T$ find $\sigma_h^n \in \mathbf{S}^h$ such that*

$$\mathcal{M}^h(\sigma_h^n, \tau_h) + \alpha \mathcal{A}^h(\sigma_h^n, \tau_h) = F_n^{*,h}(\tau_h) \quad \forall \tau_h \in \mathbf{S}^h.$$

The next theorem is the discrete counterpart of Theorem 2.2.

THEOREM 3.2 (Laplacian Structure of the Discrete Reduced System). *Let (3.1) be in dimension d . Then, the discrete reduced system of Definition 3.1 corresponds to a discrete Laplacian problem:*

$$\alpha \sum_{i=1}^d a_{ii}^h(\sigma_h, \tau_h) = \sum_{i=1}^d f_{ii}^h(\tau_h),$$

where $a_{ii}^h(\cdot, \cdot)$ and $f_{ii}^h(\cdot)$ denote the discrete bilinear and linear forms associated with the chosen spatial discretisation method.

Proof. The proof follows directly from that of Theorem 2.2, with adaptations to the discrete setting. \square

We now consider the algebraic formulation corresponding to Theorem 3.2. Let $V \in \mathbb{R}^{N_n \times m}$ be a matrix whose columns form a basis for \mathbf{S}^h . Then, for a generic spatial dimension d , the kernel of the matrix M is the span of the vectors obtained by a vectorization (by row) of the matrices σ_h having an identity block in correspondence with each diagonal component σ_{ii} of the pseudo-stress tensor, and zero blocks elsewhere. Given the subset $\mathcal{I}_d = \{j \in \mathbb{N} \mid j = (i-1)d + i \text{ for } i = 1, \dots, d\}$, the identity matrix $I_d \in \mathbb{R}^{N_n/d^2 \times N_n/d^2}$, and a vector $\mathbf{v} \in \mathbb{R}^{d^2}$, we define

$$(3.3) \quad V = \frac{1}{\sqrt{d}} (\mathbf{v} \otimes I_d), \quad \text{span}(V) = \ker(M), \quad \text{with } v_j = \begin{cases} 1 & \text{if } j \in \mathcal{I}_d, \\ 0 & \text{otherwise,} \end{cases}$$

where $\text{span}(V)$ is understood as the span of the columns of the matrix V . In order to characterise the algebraic structure of the matrix that represents the reduced system, i.e. $V^T A^* V$, we begin by illustrating the structure of the algebraic problem (3.2) in the cases $d = 2$ and $d = 3$.

For $d = 2$, the pseudo-stress tensor $\sigma \in \mathbb{R}^{2 \times 2}$ is represented by its four components, which we collect into the vector of unknowns $\sigma = [\sigma_{11} \ \sigma_{12} \ \sigma_{21} \ \sigma_{22}]^\top$. Then, the matrices M and A become

$$M = \frac{1}{2} \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix} \otimes M_1, \quad A = \begin{bmatrix} C_2 & 0 \\ 0 & C_2 \end{bmatrix} \quad \text{where } C_2 = \begin{bmatrix} B_{11} & B_{12} \\ B_{12}^\top & B_{22} \end{bmatrix},$$

where $M_1, B_{ij} \in \mathbb{R}^{N_h/4 \times N_h/4}$, for $i, j \in \{1, 2\}$, are defined according to the chosen numerical discretisation method. In particular, the blocks B_{ij} are directly derived from the components of the discrete form $a_{ij}^h(\cdot, \cdot)$ for $i, j \in \{1, 2\}$.

We now study the case $d = 3$: the pseudo-stress tensor $\sigma \in \mathbb{R}^{3 \times 3}$, and the vector of unknowns is assembled by collecting all nine components into a single column vector $\sigma = [\sigma_{11} \ \sigma_{12} \ \sigma_{13} \ \sigma_{21} \ \sigma_{22} \ \sigma_{23} \ \sigma_{31} \ \sigma_{32} \ \sigma_{33}]^\top$. For $d = 3$, the matrices M and A become

$$M = \frac{1}{3} \begin{bmatrix} 2 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 2 \end{bmatrix} \otimes M_1, \quad A = \begin{bmatrix} C_3 & 0 & 0 \\ 0 & C_3 & 0 \\ 0 & 0 & C_3 \end{bmatrix} \quad \text{and } C_3 = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12}^\top & B_{22} & B_{23} \\ B_{13}^\top & B_{23}^\top & B_{33} \end{bmatrix},$$

where $M_1, B_{ij} \in \mathbb{R}^{N_h/9 \times N_h/9}$, for $i, j \in \{1, 2, 3\}$ are defined, again, according to the chosen numerical discretisation method. In particular, the blocks B_{ij} are directly derived from the discrete forms $a_{ij}^h(\cdot, \cdot)$ for $i, j \in \{1, 2, 3\}$. We can now characterise the algebraic structure of $V^\top A^* V$, i.e. the matrix that represents the reduced system in dimension d .

COROLLARY 3.3 (Reduced Matrix Structure). *Let (3.2) be the algebraic formulation of the pseudo-stress Stokes problem. Let B_{ii} be the block of the matrix A corresponding to the diagonal component σ_{ii} of the pseudo-stress tensor, and let V be defined as in (3.3). Then, the matrix representing the reduced problem has the expression $V^\top A^* V = \frac{\alpha}{d} \sum_{i=1}^d B_{ii}$, where $\sum_{i=1}^d B_{ii}$ coincides with the stiffness matrix of the discrete Laplacian in dimension d .*

Proof. Since $\text{span}(V) = \ker(M)$, by definition (3.3), we have that $V^\top M V = 0$, and therefore $V^\top A^* V = V^\top (M + \alpha A) V = \alpha V^\top A V$. We now compute $V^\top A V$ explicitly. Recall that V contains an identity block in correspondence with each diagonal component σ_{ii} , $i = 1, \dots, d$, and zero blocks elsewhere, and that A has a block-diagonal structure with d copies of C_d on the block diagonal. Then, the i -th identity block in V^\top extracts the i -th block-row of the i -th copy of C_d . This yields

$$V^\top A = \frac{1}{\sqrt{d}} \begin{bmatrix} \underbrace{B_{11} \ \cdots \ B_{1d}}_{\text{1st block-row of } C_d} & \cdots & \underbrace{B_{d1} \ \cdots \ B_{dd}}_{\text{d-th block-row of } C_d} \end{bmatrix},$$

where $B_{ij} = B_{ji}^\top$ for $i \neq j$. Multiplying on the right by V , all off-diagonal blocks B_{ij} with $i \neq j$ are annihilated. Therefore, only the diagonal blocks B_{ii} survive, and we obtain $V^\top A^* V = \frac{\alpha}{d} \sum_{i=1}^d B_{ii}$. \square

For $d = 2$, we have that $V = \frac{1}{\sqrt{2}} [1 \ 0 \ 0 \ 1]^\top \otimes I_2$, where $I_2 \in \mathbb{R}^{\frac{N_h}{4} \times \frac{N_h}{4}}$ denotes the identity matrix. While the matrix representing the reduced system satisfies $V^\top A^* V = \frac{\alpha}{2} (B_{11} + B_{22})$. Similarly, for $d = 3$ we have that $V = \frac{1}{\sqrt{3}} [1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1]^\top \otimes I_3$, where I_3 now denotes the identity matrix of size $N_h/9$. Moreover, we obtain that the matrix of the reduced system has the following expression: $V^\top A^* V = \frac{\alpha}{3} (B_{11} + B_{22} + B_{33})$.

3.2. Examples of spatial discretisation methods. The fully-discrete formulation (3.1) has been introduced in an abstract form, without specifying the spatial discretisation. This allows the framework to accommodate different numerical methods for the approximation of the spatial operators. In this section, we illustrate two possible spatial discretisations fitting the abstract setting (3.1). In particular, we consider a continuous Galerkin finite element method and a discontinuous Galerkin method on polytopal meshes (PolyDG) (see [22, 23] for further details). In each case, we define the corresponding discrete space V_h and the associated discrete bilinear and linear forms $m^h(\cdot, \cdot)$, $a_{ij}^h(\cdot, \cdot)$ and $f_{ij}^h(\cdot)$. In the following, we use the short-hand notation $(\cdot, \cdot)_{\mathcal{T}_h} = \sum_{\kappa \in \mathcal{T}_h} \int_{\kappa} \cdot$ and $\langle \cdot, \cdot \rangle_{\mathcal{F}_h} = \sum_{F \in \mathcal{F}_h} \int_F \cdot$.

3.2.1. Conforming Galerkin discretisation. In this section, we present the finite element framework used for a conforming Galerkin discretisation of the problem. Let \mathcal{T}_h be a conforming triangulation of the domain Ω , which means that there exists a constant $C > 0$ such that $\sup_{\kappa \in \mathcal{T}_h} h_{\kappa} \leq h$, $\sup_{\kappa \in \mathcal{T}_h} \frac{h_{\kappa}}{\rho_{\kappa}} \leq C$, where h_{κ} and ρ_{κ} denote, respectively, the diameter and the inradius of the simplex κ . For an integer $\ell \geq 0$, we denote by $\mathbb{P}_{\ell}(\kappa)$ the space of real polynomials of degree at most ℓ on κ , and by $\text{RT}_{\ell}(\kappa)$ the Raviart–Thomas space [35, 39] of order ℓ on κ . Analogously, $\text{BDM}_{\ell}(\kappa)$ denotes the Brezzi–Douglas–Marini space [14, 15] of order ℓ on κ . Using these local spaces, we can introduce the conforming approximation $V_h \subset V$ defined as

$$V_h = \{v_h \in V : v_h|_{\kappa} \in \text{RT}_{\ell}(\kappa) \text{ or } \text{BDM}_{\ell+1}(\kappa), \forall \kappa \in \mathcal{T}_h\}.$$

Independently of the specific finite element space adopted, the discrete variational problem retains the same structure as the continuous one. In particular, the corresponding fully-discrete approximation has the form (3.1), where

$$m^h(\sigma, \tau) = (\sigma, \tau)_{\mathcal{T}_h}, \quad a_{ij}^h(\sigma, \tau) = (\partial_i \sigma, \partial_j \tau)_{\mathcal{T}_h}, \quad f_{ij}^h(\tau) = (F_{ij}^*, \tau)_{\mathcal{T}_h}.$$

Remark 3.4. The well-posedness and error analysis of conforming Galerkin discretisations based on RT and BDM finite elements for the pseudo-stress formulation of Stokes-type problems have been addressed in [18, 19].

3.2.2. PolyDG discretisation. We now consider the PolyDG method, a discontinuous Galerkin scheme defined on general polygonal or polyhedral meshes (see [22] for a comprehensive monograph). A detailed stability and convergence analysis of the PolyDG approximation for the pseudo-stress formulation can be found in [2]; see also Remark 3.5 for the assumptions and error estimates relevant to the present work. Let \mathcal{T}_h be a polytopal mesh of the domain Ω , i.e., $\mathcal{T}_h = \bigcup_{\kappa} \kappa$, being κ a general polygon ($d = 2$) or polyhedron ($d = 3$). Given a polytopal element κ , we define by $|\kappa|$ its measure and by h_{κ} its diameter, and set $h = \max_{\kappa \in \mathcal{T}_h} h_{\kappa}$. We let a polynomial degree $p_{\kappa} \geq 1$ be associated with each element $\kappa \in \mathcal{T}_h$ and we denote by $p_h : \mathcal{T}_h \rightarrow \mathbb{N}^* = \{n \in \mathbb{N} : n \geq 1\}$ the piecewise constant function such that $(p_h)|_{\kappa} = p_{\kappa}$. Then, we define the discrete space $V_h = [P_{p_h}(\mathcal{T}_h)]^{d \times d}$, where $P_{p_h}(\mathcal{T}_h) = \prod_{\kappa \in \mathcal{T}_h} \mathbb{P}_{p_{\kappa}}(\kappa)$, and $\mathbb{P}_{\ell}(\kappa)$ is the space of piecewise polynomials in κ of total degree less than or equal to $\ell \geq 1$. We refer the reader to [2, 21, 22] for the main assumptions on \mathcal{T}_h . We define an interface as the intersection of the $(d - 1)$ -dimensional faces of any two neighboring elements of \mathcal{T}_h . In two dimensions ($d = 2$), the interfaces of any element $\kappa \in \mathcal{T}_h$ are line segments, i.e., $(d - 1)$ -dimensional simplexes. In three dimensions ($d = 3$), this will not be the case; we assume that each interface of an element $\kappa \in \mathcal{T}_h$ can be partitioned into a collection of co-planar triangles. Accordingly, we use the term “face” to denote a $(d - 1)$ -dimensional simplex (line segment for $d = 2$ or triangle for $d = 3$) belonging to the boundary of κ . We also decompose the set of faces as $\mathcal{F} = \mathcal{F}_h^I \cup \mathcal{F}_h^D \cup \mathcal{F}_h^N$, where \mathcal{F}_h^I contains the internal faces and \mathcal{F}_h^D and \mathcal{F}_h^N the faces of the Dirichlet and Neumann boundary conditions, respectively. Finally, for sufficiently piecewise

smooth vector- and tensor-valued fields \mathbf{v} and $\boldsymbol{\tau}$, respectively, and for any pair of neighbouring elements κ^+ and κ^- sharing a face $F \in \mathcal{F}_h^I$, we introduce the jump and average operators $[[\mathbf{v}]] = \mathbf{v}^+ \otimes \mathbf{n}^+ + \mathbf{v}^- \otimes \mathbf{n}^-$, $[[\boldsymbol{\tau}]] = \boldsymbol{\tau}^+ \mathbf{n}^+ + \boldsymbol{\tau}^- \mathbf{n}^-$, $\{\{\mathbf{v}\}\} = \frac{(\mathbf{v}^+ + \mathbf{v}^-)}{2}$, $\{\{\boldsymbol{\tau}\}\} = \frac{(\boldsymbol{\tau}^+ + \boldsymbol{\tau}^-)}{2}$, where \otimes is the tensor product in \mathbb{R}^d , \cdot^\pm denotes the trace on F taken within κ^\pm , and \mathbf{n}^\pm is the outer normal vector to $\partial\kappa^\pm$. Accordingly, on boundary faces $F \in \mathcal{F}_h^D \cup \mathcal{F}_h^N$, we set $[[\mathbf{v}]] = \mathbf{v} \otimes \mathbf{n}$, $[[\boldsymbol{\tau}]] = \boldsymbol{\tau} \mathbf{n}$, and $\{\{\mathbf{v}\}\} = \mathbf{v}$, $\{\{\boldsymbol{\tau}\}\} = \boldsymbol{\tau}$. In the following, we use $\nabla_h \cdot$ to denote the element-wise divergence operator. The corresponding fully-discrete PolyDG approximation has the form (3.1), where in this case we define

$$\begin{aligned} m^h(\sigma, \tau) &= (\sigma, \tau)_{\mathcal{T}_h}, & f_{ij}^h(\tau) &= (F_{ij}^*, \tau)_{\mathcal{T}_h}, \\ a_{ij}^h(\sigma, \tau) &= (\partial_i \sigma, \partial_j \tau)_{\mathcal{T}_h} - \langle \{\{\partial_i \sigma\}\}, [[\tau]] \rangle_{\mathcal{F}_h^{I,N}} \\ &\quad - \langle \{\{\partial_j \tau\}\}, [[\sigma]] \rangle_{\mathcal{F}_h^{I,N}} + \langle \gamma_e [[\sigma]], [[\tau]] \rangle_{\mathcal{F}_h^{I,N}}. \end{aligned}$$

Here, $\mathcal{F}_h^{I,N} = \mathcal{F}_h^I \cup \mathcal{F}_h^N$ and the stabilization function $\gamma_e : \mathcal{F}_h^{I,N} \rightarrow \mathbb{R}_+$ is defined as a function of the penalty coefficient $\alpha^* > 0$ as follows:

$$\gamma_e(\mathbf{x}) = \begin{cases} \alpha^* \max_{\kappa \in \{\kappa^+, \kappa^-\}} \frac{p_\kappa^2}{h_\kappa}, & \mathbf{x} \in e, e \in \mathcal{F}_h^I, e \subset \partial\kappa^+ \cap \partial\kappa^-, \\ \alpha^* \frac{p_\kappa^2}{h_\kappa}, & \mathbf{x} \in e, e \in \mathcal{F}_h^N, e \subset \partial\kappa^+ \cap \partial\Gamma_N. \end{cases}$$

Remark 3.5. Let us recall the main assumptions and results from the error analysis of the fully-discrete PolyDG scheme [2]. Let the exact solution $\boldsymbol{\sigma}$ be sufficiently smooth and $\boldsymbol{\sigma}_h^n$ be the solution of (3.2) with $\alpha = \Delta t$ (implicit Euler) and for a sufficiently large penalty coefficient α^* . If (h, p) are quasi uniform, the error in the energy norm satisfies $\|\boldsymbol{\sigma} - \boldsymbol{\sigma}_h\|_E \sim \Delta t + h^p$.

4. Numerical solvers. In this section, we present a numerical framework for the solution of the linear system (3.2). Since A^* is symmetric and positive definite, the CG method is the natural choice. However, as discussed in Section 2.4, the conditioning of A^* deteriorates as $\Delta t \rightarrow 0$, motivating the use of a deflated CG strategy that projects out the directions in $\ker(M)$ and stabilises the iteration count [3]. The structure of the resulting inner system naturally motivates the use of a multigrid solver, where a RAS smoother exploits the block structure of the discretisation. Since the inner system is solved only approximately, the deflation projection applied at each outer iteration is itself inexact, and the corresponding correction varies from one iteration to the next depending on the inner solver tolerance. As a result, if the tolerance of the inner solver is not appropriately related to the outer residual, the classical CG method is no longer guaranteed to be robust. This motivates the use of a flexible conjugate gradient algorithm, specifically designed to accommodate variable system matrices. In summary, our solution strategy combines deflation to achieve robustness with respect to Δt , a multigrid method to efficiently solve the inner system, and a flexible CG scheme to ensure stable convergence of the overall algorithm. In Sections 4.1 - 4.3, we introduce the main building blocks of the proposed solver: the deflated conjugate gradient method, the inner linear system and its multigrid solution strategy, and the flexible CG framework. The corresponding algorithms are presented in Section 4.4.

4.1. The deflated conjugate gradient method. Consider a linear system of equations of the form (3.2), where $A^* \in \mathbb{R}^{N_h \times N_h}$ is symmetric and positive definite and $\boldsymbol{\sigma}_h^n, \mathbf{f}^* \in \mathbb{R}^{N_h}$. To simplify the notation, we denote by \mathbf{x} the solution $\boldsymbol{\sigma}_h^n$ of the linear system. Since the speed of convergence of CG depends on the distribution of the eigenvalues of A^* , the idea of deflation [32, 36] is to “hide” parts of the spectrum of A^* from CG such that the CG iteration operates on an equivalent system with a significantly reduced condition number compared to the original one. The spectral components that are hidden depend on the chosen deflation

subspace $\mathbf{S} \subset \mathbb{R}^{N_h}$, and the resulting improvement in convergence speed is entirely governed by this choice. Given $\mathbf{S}^{\perp A^*} = (A^* \mathbf{S})^\perp$ the A^* -orthogonal complement of \mathbf{S} , it is possible to split the solution \mathbf{x} into a component in \mathbf{S} and a component in $\mathbf{S}^{\perp A^*}$ via the A^* -orthogonal projection $\pi_{A^*}(\mathbf{S}) \in \mathbb{R}^{N_h \times N_h}$ onto \mathbf{S} . Given a matrix $V \in \mathbb{R}^{N_h \times m}$ whose columns form a basis for \mathbf{S} , and $\pi_{A^*}(\mathbf{S}) = V(V^\top A^* V)^{-1} V^\top A^*$, we obtain

$$(4.1) \quad \mathbf{x} = (I - \pi_{A^*}(\mathbf{S})) \hat{\mathbf{x}} + V(V^\top A^* V)^{-1} V^\top \mathbf{f}^*,$$

where $\hat{\mathbf{x}}$ is the solution of the so-called *deflated linear system*, i.e.

$$(4.2) \quad A^*(I - \pi_{A^*}(\mathbf{S})) \hat{\mathbf{x}} = (I - \pi_{A^*}(\mathbf{S}))^\top \mathbf{f}^*.$$

Thus, to obtain \mathbf{x} , one needs to compute the solution $\hat{\mathbf{x}}$ of the deflated system. We note that $A^*(I - \pi_{A^*}(\mathbf{S})) = (I - \pi_{A^*}(\mathbf{S}))^\top A^*(I - \pi_{A^*}(\mathbf{S}))$, as shown in [32]. Then the matrix $A^*(I - \pi_{A^*}(\mathbf{S}))$ is symmetric positive semi-definite and so we can apply the CG method. Matrix singularity poses no obstacle to the standard CG iteration, provided that (4.2) is consistent, i.e., the right hand side $(I - \pi_{A^*}(\mathbf{S}))^\top \mathbf{f}^*$ is in the range of $A^*(I - \pi_{A^*}(\mathbf{S}))$, which has been verified in [32, Lemma 2.1]. To summarise, one interprets deflated CG as the standard CG algorithm applied to the deflated system (4.2). Let $\lambda_1 \geq \dots \geq \lambda_n \geq 0$ denote the eigenvalues of the self-adjoint and positive semi-definite matrix $A^*(I - \pi_{A^*}(\mathbf{S}))$, and $k \in \mathbb{N}$ the largest index such that $\lambda_k \neq 0$. Then the errors of the CG iterates satisfy

$$\|e_i\|_{A^*} \leq 2 \left(\frac{\sqrt{\kappa_{\text{eff}}} - 1}{\sqrt{\kappa_{\text{eff}}} + 1} \right)^i \|e_0\|_{A^*}, \quad i = 0, 1, 2, \dots,$$

where $\kappa_{\text{eff}} = \lambda_1/\lambda_k$ is the effective condition number of the deflated matrix $A^*(I - \pi_{A^*}(\mathbf{S}))$, in contrast to the condition number κ of A^* [32].

Note that applying the deflated CG method requires computing the inverse of $V^\top A^* V$ in (4.1) in the definition of $\pi_{A^*}(\mathbf{S})$. In particular, in the deflated CG method, each iteration requires solving the *inner linear system*

$$(4.3) \quad (V^\top A^* V) \mathbf{z}_{k+1} = V^\top A^* \mathbf{r}_{k+1},$$

where \mathbf{r}_{k+1} is the current residual, and \mathbf{z}_{k+1} is the solution of the inner system. When the dimension of the subspace \mathbf{S} is large, solving the inner problem (4.3) exactly may become too costly. In such cases, it is reasonable to approximate its solution by means of an iterative solver. Previous analyses [32, 34] have shown that deflation methods may react strongly to the level of accuracy achieved in the inner computation. The outer iteration is terminated once

$$\|\mathbf{r}_i\|_2 \leq \tau \|\mathbf{f}^*\|_2 = \varepsilon, \quad 0 < \tau \ll 1,$$

and we seek an analogous stopping rule for the inner iteration, expressed as

$$\|\mathbf{r}_i^c\|_2 \leq \tau^c \|\mathbf{f}^c\|_2.$$

The results reported in [34] indicate that a fixed tolerance of the form

$$(4.4) \quad \tau^c = \varepsilon c_F, \quad 0 < c_F \leq 1,$$

is often sufficient. Furthermore, as pointed out in [32], a high level of accuracy in the inner problem is crucial mainly during the first few outer iterations, whereas it can be gradually relaxed thereafter. This observation motivates the choice of a variable threshold such as

$$(4.5) \quad \tau^c = c_A \frac{\varepsilon}{\|\mathbf{r}_i\|},$$

for some constant $c_A > 0$. We refer to (4.5) as an *adaptive tolerance rule* [32]: the relative accuracy required for the inner system becomes less stringent as the outer residual \mathbf{r}_i decreases, while still ensuring that the outer CG iteration converges to the target precision.

4.2. Multigrid solver for the inner system. We then choose V as the kernel of the matrix M : this choice is motivated by the fact that the kernel of M captures the problematic spectral components of the system, and deflating this subspace effectively removes the directions responsible for slow convergence of CG as $\Delta t \rightarrow 0$. This corresponds precisely to defining the matrix V as in (3.3) and the deflation subspace $\mathbf{S}^h = \ker(\mathcal{M}^h)$. Then, the inner system corresponds to the discrete reduced system defined in Definition 3.1. From the analysis carried out in Section 3.1, it follows that the inner system corresponds to a Laplacian problem. This feature immediately suggests the use of multigrid methods as efficient solvers for the associated inner problem. In our numerical tests, the inner system is handled by a multigrid iterative scheme employing a W-cycle strategy, and as a smoother we use a RAS iteration. We now introduce an iterative multigrid strategy to efficiently solve the inner problem (4.3). Let us rewrite the inner system (4.3) as

$$(4.6) \quad A_J^c \mathbf{z}_J = \mathbf{f}_J^c,$$

and consider a sequence of finite-dimensional inner product spaces $\mathbf{W}_1, \mathbf{W}_2, \dots, \mathbf{W}_J$ (possibly non-nested). The inner product on \mathbf{W}_k is denoted by $(\cdot, \cdot)_k$. Then, we define the symmetric positive definite bilinear form $\mathcal{A}_k^c(\cdot, \cdot)$ on $\mathbf{W}_k \times \mathbf{W}_k$ for $k = 1, \dots, J$. Our goal is to efficiently solve (4.6) at the finest level (i.e. for $k = J$); indeed, the multigrid algorithm that will be presented generates iterative procedures for solving the problem on \mathbf{W}_J . Let $A_k^c : \mathbf{W}_k \rightarrow \mathbf{W}_k$ be the (symmetric positive definite) operator such that

$$(A_k^c \mathbf{u}, \mathbf{v}) = \mathcal{A}_k^c(\mathbf{u}, \mathbf{v}) \quad \text{for all } \mathbf{u}, \mathbf{v} \in \mathbf{W}_k.$$

Then the corresponding algebraic formulation at the finest level reads as $A_J^c \mathbf{z}_J = \mathbf{f}_J^c$, i.e. (4.6). We now recall the main ingredients of the multigrid algorithm [12, 13]: a *smoother* operator, and *prolongation* and *restriction* operators. Denoting by S_k the *smoother*, the j -th smoothing iteration reads as

$$\mathbf{z}_k^{(j+1)} = \mathbf{z}_k^{(j)} + S_k(\mathbf{f}_k^c - A_k^c \mathbf{z}_k^{(j)}), \quad j = 1, 2, \dots$$

Now, since the spaces are (possibly) non-nested, i.e., $\mathbf{W}_{k-1} \not\subset \mathbf{W}_k$, the most natural way to define the prolongation operator is through the L^2 -projection; see, e.g. [5]. Thus, the operator $I_{k-1}^k : \mathbf{W}_{k-1} \rightarrow \mathbf{W}_k$ is then defined by

$$(I_{k-1}^k \mathbf{u}_H, \mathbf{v}_h)_{L^2(\Omega)} = (\mathbf{u}_H, \mathbf{v}_h)_{L^2(\Omega)} \quad \forall \mathbf{v}_h \in \mathbf{W}_k.$$

The restriction operator $I_k^{k-1} : \mathbf{W}_k \rightarrow \mathbf{W}_{k-1}$ is then defined as the adjoint of I_{k-1}^k with respect to the $L^2(\Omega)$ -inner product, that is,

$$(I_k^{k-1} \mathbf{v}_h, \mathbf{u}_H)_{L^2(\Omega)} = (\mathbf{v}_h, I_{k-1}^k \mathbf{u}_H)_{L^2(\Omega)} \quad \forall \mathbf{u}_H \in \mathbf{W}_{k-1}.$$

To enhance the performance of the W-cycle, we employ a RAS-type smoothing method [24]. Consider the linear system (4.6), and let the set of unknowns be partitioned into blocks \mathcal{B}_i , with $i = 1, \dots, n_b$, so that every component z_j of the global vector \mathbf{z} belongs to at least one block. For each block \mathcal{B}_i , we denote by $R_i : \mathbb{R}^n \rightarrow \mathbb{R}^{n_i}$ the operator that extracts from the global vector the subvector associated with \mathcal{B}_i , where n_i is the size of the block. With a slight abuse of notation, we use the same symbol for both the operator and its matrix representation. Its transpose, R_i^T , acts as a zero-extension prolongation operator from \mathbb{R}^{n_i} to \mathbb{R}^n . The corresponding local matrix is defined as $(A_k^c)_i = R_i A_k^c R_i^T$, which represents the restriction of A_k^c to the block \mathcal{B}_i . Thus, each block requires solving the small local problem

$(A_k^c)_i z_i^{(j)} = R_i (f_k^c - A_k^c z_k^{(j)})$. Once the local problems have been solved, the RAS iteration updates the global solution as follows:

$$z_k^{(j+1)} = z_k^{(j)} + \sum_{i=1}^{n_b} R_i^\top D_i z_i^{(j)},$$

where the matrix D_i denotes a diagonal (partition of unity) weighting operator. In general, its diagonal entries d_{jj} , $j = 1, \dots, n$, are chosen as the reciprocal of the number of blocks that include the unknown z_j . Then, the RAS smoother operator is defined as $S_k = \sum_{i=1}^{n_b} R_i^\top D_i (A_k^c)_i^{-1} R_i$. We define the blocks \mathcal{B}_i as the collection of degrees of freedom associated with the i -th mesh element and all degrees of freedom belonging to its immediate neighboring elements, resulting in a one-layer overlap. This choice balances computational efficiency with the smoothing effectiveness of the RAS iteration within the W-cycle.

4.3. Flexible conjugate gradient method. In the proposed approach, the system operator is not applied exactly at each iteration. Indeed, due to the deflation strategy, each outer iteration requires the solution of an inner linear system, which is handled approximately by a multigrid method. As a consequence, the effective operator seen by the outer iteration varies from step to step. The classical convergence theory of the CG method, however, relies on the assumption that the underlying operator is fixed, symmetric positive definite, and applied exactly at every iteration. When this assumption is violated, for instance due to inexact inner solves or variable tolerances, standard CG may lose robustness and exhibit degraded convergence. To address this issue, flexible Krylov methods are required. In particular, we employ the Flexible Conjugate Gradient (FCG) method [37], which is specifically designed to accommodate variable operators. The main distinction with respect to classical CG lies in the explicit A -orthogonalization of each new search direction against the previous ones. At iteration i , the new direction is orthogonalized with respect to a selected number of earlier directions, determined by a sequence of truncation parameters $\{m_i\}$. In the absence of truncation (FCG(∞)), orthogonality is enforced with all previously generated directions, i.e., $m_i = i$ for all i . In Algorithm 4.1 we report the untruncated FCG scheme (FCG(∞)) for solving the system under consideration, i.e. $A^* \mathbf{x} = \mathbf{f}^*$, where $A^* \in \mathbb{R}^{N_h \times N_h}$ is symmetric positive definite, $\mathbf{f}^* \in \mathbb{R}^{N_h}$, and $\mathbf{x}_0 \in \mathbb{R}^{N_h}$ is an initial guess (see also Algorithm 2.1 in [37]).

Algorithm 4.1 FCG (untruncated Flexible Conjugate Gradient method)

- 1: Input: rhs \mathbf{f}^* , initial guess \mathbf{x}_0 , tolerance tol , map $(\mathbf{d}, \text{tol}) \rightarrow A^*(\mathbf{d}, \text{tol})$
 - 2: Output: solution \mathbf{x}
 - 3: $\mathbf{r}_0 = \mathbf{f}^* - A^*(\mathbf{x}_0, \text{tol})$;
 - 4: **for** $i = 0, 1, \dots$ **do**
 - 5: $\mathbf{d}_i = \mathbf{r}_i - \sum_{k=0}^{i-1} \frac{(\mathbf{r}_i, A^*(\mathbf{d}_k, \text{tol}))}{(\mathbf{d}_k, A^*(\mathbf{d}_k, \text{tol}))} \mathbf{d}_k$;
 - 6: $\alpha_i = \frac{(\mathbf{d}_i, \mathbf{r}_i)}{(\mathbf{d}_i, A^*(\mathbf{d}_i, \text{tol}))}$;
 - 7: $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{d}_i$;
 - 8: $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i A^*(\mathbf{d}_i, \text{tol})$;
 - 9: **end for**
-

Compared to the CG approach, the untruncated FCG method incurs higher computational and memory costs, as it stores all previous search directions and performs full orthogonalization at each iteration. For a moderate number of iterations, however, this overhead remains acceptable and it is compensated by the improved robustness of the method. Moreover, the added advantage of full flexibility may justify this extra cost. In addition, by using a fully flexible, unrestarted scheme, we avoid delicate parameter tuning and obtain reliable convergence even when the operator is applied inexactly (see also Remark 5.1).

Remark 4.1. An alternative approach to handle the loss of robustness of standard CG is the use of the Generalized Minimum Residual (GMRES) method [40] as outer solver, which builds an orthonormal basis of the Krylov subspace via the Arnoldi process and minimises the residual over the entire subspace at each iteration. As discussed in Remark 5.2 below, GMRES eliminates the convergence issues observed with standard CG under inexact inner solves, confirming its robustness in this setting. The choice of FCG over GMRES in our framework is motivated by the fact that FCG retains the structure of a conjugate gradient method and exploits the symmetry of A^* .

4.4. Numerical algorithms. In this section, we detail the numerical algorithms that define the flexible deflated CG method and the multigrid W-cycle scheme adopted in our numerical simulations. We start by recalling the interpretation introduced in Section 4.1: the (flexible) deflated CG method can be viewed as the application of the (flexible) CG algorithm to the deflated linear system (4.2). In Algorithm 4.2 we report the flexible deflated CG scheme for solving (3.2), i.e., the problem $A^*x = f^*$.

Algorithm 4.2 FDCG (Flexible Deflated Conjugate Gradient method)

- 1: Input: matrix A^* , rhs f^* , initial guess x_0 , smoothing steps m_1 and m_2 , tolerance tol , maximum number of iterations n_{\max} , matrix V , prolongation I_{k-1}^k and restriction I_k^{k-1}
 - 2: Output: solution x
 - 3: $Z_J = V^T A^* V$;
 - 4: **for** $k = J, \dots, 2$ **do**
 - 5: $Z_{k-1} = I_k^{k-1} Z_k I_{k-1}^k$;
 - 6: **end for**
 - 7: $(z, \text{tol}) \rightarrow \hat{z}(z, \text{tol}) := \text{Multigrid}(\{Z_k\}_{k=1}^J, V^T z, 0, m_1, m_2, \text{tol}, n_{\max})$;
 - 8: $(z, \text{tol}) \rightarrow \pi(z, \text{tol}) := V \hat{z}(z, \text{tol})$;
 - 9: Operator M_D and rhs f_D of the deflated system:
 - 10: $(z, \text{tol}) \rightarrow M_D(z, \text{tol}) := A^*(z - \pi(A^*z, \text{tol}))$;
 - 11: $f_D = f^* - \pi^T(f^*, \text{tol})$;
 - 12: $\hat{x} = \text{FCG}(f_D, x_0, \text{tol}, M_D)$;
 - 13: $x = (\hat{x} - \pi(A^*\hat{x}, \text{tol})) + \pi(f^*, \text{tol})$;
-

Then, we turn our attention to the multigrid component of the proposed solver, which is employed to solve the inner system (4.6). Given an initial guess $z_0 \in W_J$ and choosing the parameters $m_1, m_2 \in \mathbb{N}$ (pre-smoothing and post-smoothing steps), the multigrid W-cycle procedure used to approximate z_J is summarized in Algorithm 4.3. We denote by $z_{n+1} = \text{Wcycle}(\{A_k^c\}_{k=1}^J, f_J^c, z_n, J, m_1, m_2)$ the approximation obtained after one iteration of the non-nested W-cycle scheme. The recursive procedure is detailed in Algorithm 4.4, where we also note that on the coarsest level $k = 1$ the problem is solved exactly. Here, we consider the general problem: find $z \in W_k$ such that $A_k^c z = f^c$.

Algorithm 4.3 Multigrid (Multigrid W-cycle iteration for the solution of (4.6))

- 1: Input: coarse-level matrices $\{A_k^c\}_{k=1}^J$, rhs f_J^c , initial guess z_0 , smoothing steps m_1 and m_2 , tolerance tol , maximum number of iterations n_{\max}
 - 2: Output: solution z , number of iterations n
 - 3: **while** $n < n_{\max}$ **and** $\|f_J^c - A_J^c z_n\| / \|f_J^c\| > \text{tol}$ **do**
 - 4: $z_{n+1} = \text{Wcycle}(\{A_k^c\}_{k=1}^J, f_J^c, z_n, J, m_1, m_2)$;
 - 5: $n = n + 1$;
 - 6: **end while**
-

Algorithm 4.4 `Wcycle` (one W-cycle iteration for $k \geq 2$ levels)

```

1: Input: coarse-level matrices  $\{A_k^c\}_{k=1}^J$ , rhs  $f^c$ , initial guess  $z_0$ , level  $k$ , smoothing steps  $m_1$  and  $m_2$ 
2: Output: solution  $z$ 
3: if  $k = 1$  then
4:   Wcycle( $\{A_k^c\}_{k=1}^J, f^c, z_0, 1, m_1, m_2$ ) =  $(A_1^c)^{-1} f^c$ ;
5: else
6:   Pre-smoothing:
7:   for  $i = 1, \dots, m_1$  do
8:      $z^{(i)} = z^{(i-1)} + S_k(f^c - A_k^c z^{(i-1)})$ ;
9:   end for
10:  Coarse grid correction:
11:   $r_{k-1} = I_k^{k-1}(f^c - A_k^c z^{(m_1)})$ ;
12:   $\bar{e}_{k-1} = \text{Wcycle}(\{A_k^c\}_{k=1}^J, r_{k-1}, 0, k-1, m_1, m_2)$ ;
13:   $e_{k-1} = \text{Wcycle}(\{A_k^c\}_{k=1}^J, r_{k-1}, \bar{e}_{k-1}, k-1, m_1, m_2)$ ;
14:   $z^{(m_1+1)} = z^{(m_1)} + I_{k-1}^k e_{k-1}$ ;
15:  Post-smoothing:
16:  for  $i = m_1 + 2, \dots, m_1 + m_2 + 1$  do
17:     $z^{(i)} = z^{(i-1)} + S_k(f^c - A_k^c z^{(i-1)})$ ;
18:  end for
19:  Wcycle( $\{A_k^c\}_{k=1}^J, f^c, z_0, k, m_1, m_2$ ) =  $z^{(m_1+m_2+1)}$ ;
20: end if

```

5. Numerical results. In this section, we present several numerical experiments aimed at assessing the performance of the proposed numerical framework for the solution to (3.2) at each time step. Numerical results are presented in both two- and three-dimensional settings, employing the spatial discretisation based on the PolyDG method introduced in Section 3.2.2, in order to thoroughly assess the performance of the proposed approach.

5.1. PolyDG Finite Element discretisation - 2D case. In this section, all computations have been implemented in the open-source MATLAB library `lymph` [1]. We consider a two-dimensional problem defined on the unit square $\Omega = (0, 1)^2$, and the following exact solution $\sigma(x, t) = \sin(2t) [\sin(\pi x) \sin(\pi y), 0, 0, -\sin(\pi x) \sin(\pi y)]^\top$; then $F(x, t)$ and the boundary data are computed accordingly, in particular Dirichlet boundary conditions are imposed on the top and right boundaries, while Neumann conditions are applied on the remaining part of the boundary. For the numerical simulations we set $\mu = 1$, the penalty coefficient $\alpha^* = 10$, the polynomial degree $p = 3$, and $\alpha = \Delta t$ (implicit Euler scheme). We employ the sets of polygonal meshes illustrated in Figure 1. In particular, the finest grids (Level 4) shown in Figure 1 contain 512 (Set 1), 1024 (Set 2), 2048 (Set 3), and 4096 (Set 4) polygonal elements. Starting from each of these initial meshes, we construct a sequence of non-nested partitions. Each coarse mesh is generated independently of the finer one, with the only requirement that the number of elements is approximately one quarter of that of the corresponding finer mesh, as commonly done in the literature (see, e.g., [4, 5]). As a preliminary step, we intend to frame the problem by reporting the condition number of the matrices involved, thereby providing an initial quantitative reference. In particular, in Table 1, we report the condition number $\kappa(A^*)$ and the effective condition number $\kappa_{\text{eff}}(A^*(I - \pi_{A^*}(\mathbf{S})))$, for the four finest grids ($h_1 \approx 0.082$, $h_2 \approx 0.059$, $h_3 \approx 0.043$, $h_4 \approx 0.030$), and for different time steps Δt . We observe that, for small Δt , κ is proportional to $1/\Delta t$ as expected, while κ_{eff} remains uniformly bounded with respect to Δt . The grey-shaded entries in the tables indicate comparable time steps and mesh sizes, and thus, according to Remark 3.5, the regime where time and space discretisation errors are balanced.

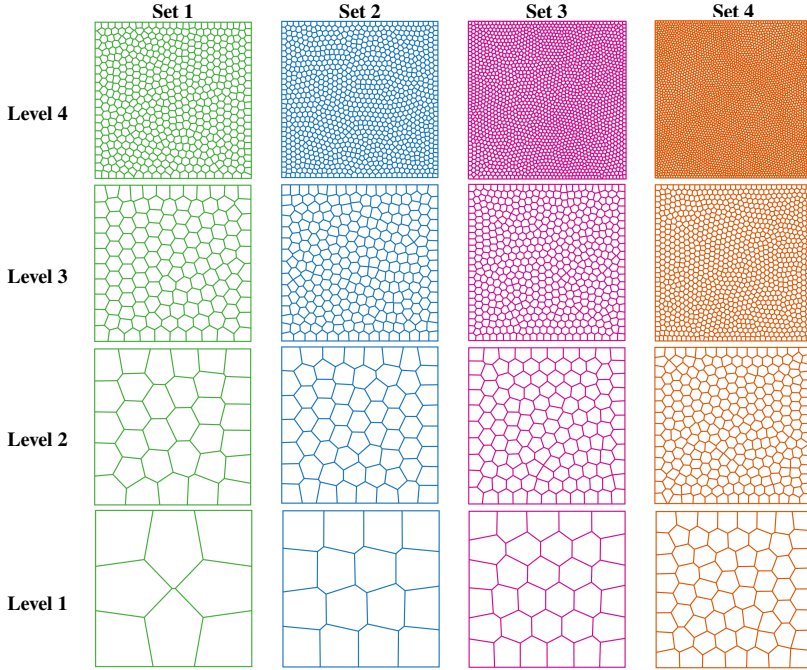


FIG. 1. Hierarchies of non-nested polygonal grids used for numerical simulations of Section 5.1. Finest grid of Set 1 with 512 elements; finest grid of Set 2 with 1024 elements; finest grid of Set 3 with 2048 elements; finest grid of Set 4 with 4096 elements.

TABLE 1
 $\kappa(A^*)$ (left) and $\kappa_{\text{eff}}(A^*(I - \pi_{A^*}(S)))$ (right) as functions of h and Δt .

| Δt | h_1 | h_2 | h_3 | h_4 | h_1 | h_2 | h_3 | h_4 |
|------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| 10^{-2} | $1.7 \cdot 10^6$ | $3.6 \cdot 10^6$ | $7.4 \cdot 10^6$ | $1.5 \cdot 10^7$ | $4.4 \cdot 10^4$ | $9.1 \cdot 10^4$ | $1.9 \cdot 10^5$ | $4.0 \cdot 10^5$ |
| 10^{-3} | $1.4 \cdot 10^6$ | $3.0 \cdot 10^6$ | $6.2 \cdot 10^6$ | $1.3 \cdot 10^7$ | $1.2 \cdot 10^4$ | $1.5 \cdot 10^4$ | $3.3 \cdot 10^4$ | $5.4 \cdot 10^4$ |
| 10^{-4} | $1.4 \cdot 10^6$ | $2.8 \cdot 10^6$ | $5.9 \cdot 10^6$ | $1.2 \cdot 10^7$ | $3.5 \cdot 10^3$ | $4.8 \cdot 10^3$ | $9.2 \cdot 10^3$ | $1.7 \cdot 10^4$ |
| 10^{-5} | $1.4 \cdot 10^6$ | $2.8 \cdot 10^6$ | $5.8 \cdot 10^6$ | $1.2 \cdot 10^7$ | $8.3 \cdot 10^2$ | $1.3 \cdot 10^3$ | $2.4 \cdot 10^3$ | $4.2 \cdot 10^3$ |
| 10^{-6} | $1.7 \cdot 10^6$ | $3.2 \cdot 10^6$ | $6.1 \cdot 10^6$ | $1.2 \cdot 10^7$ | $1.9 \cdot 10^2$ | $3.1 \cdot 10^2$ | $6.4 \cdot 10^2$ | $1.2 \cdot 10^3$ |
| 10^{-7} | $6.5 \cdot 10^6$ | $7.4 \cdot 10^6$ | $1.0 \cdot 10^7$ | $1.6 \cdot 10^7$ | $8.0 \cdot 10^1$ | $9.3 \cdot 10^1$ | $1.4 \cdot 10^2$ | $2.5 \cdot 10^2$ |
| 10^{-8} | $6.4 \cdot 10^7$ | $6.9 \cdot 10^7$ | $7.7 \cdot 10^7$ | $7.4 \cdot 10^7$ | $8.1 \cdot 10^1$ | $9.0 \cdot 10^1$ | $1.2 \cdot 10^2$ | $1.3 \cdot 10^2$ |
| 10^{-9} | $6.4 \cdot 10^8$ | $6.9 \cdot 10^8$ | $7.7 \cdot 10^8$ | $7.4 \cdot 10^8$ | $8.1 \cdot 10^1$ | $9.0 \cdot 10^1$ | $1.2 \cdot 10^2$ | $1.3 \cdot 10^2$ |
| 10^{-10} | $6.4 \cdot 10^9$ | $6.9 \cdot 10^9$ | $7.7 \cdot 10^9$ | $7.4 \cdot 10^9$ | $8.1 \cdot 10^1$ | $9.0 \cdot 10^1$ | $1.2 \cdot 10^2$ | $1.3 \cdot 10^2$ |

Table 2 reports the iteration counts needed by the CG and the deflated CG algorithms to reduce the relative residual below a tolerance $\text{tol} = 10^{-8}$, for different time steps Δt and mesh sizes h . In the deflated CG strategy, the inner system is solved exactly by a direct solver. For each method, the table reports the number of iterations required to reduce the relative residual below 10^{-8} , considering only the solution of the linear system (3.2) arising at a single time step. The iteration counts are obtained as the average over 10 independent tests, in which the initial condition is perturbed using randomly generated data. In particular, Table 2 shows that the CG iteration counts increase with mesh refinement ($h \rightarrow 0$) and exhibit a growth proportional to $\sqrt{\Delta t}$ as $\Delta t \rightarrow 0$. In contrast, the deflated CG method yields dramatically fewer iterations as $\Delta t \rightarrow 0$, confirming the effectiveness of the deflation strategy.

Table 3 shows the results obtained with the deflated CG method, where the inner system is now solved by a multigrid W-cycle approach with RAS as smoother. The multigrid hierarchy

TABLE 2
Iteration counts for CG (left) and deflated CG (right) as functions of h and Δt , with $tol = 10^{-8}$.

| Δt | h_1 | h_2 | h_3 | h_4 | h_1 | h_2 | h_3 | h_4 |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|
| 10^{-2} | 3766 | 5255 | 7538 | 10631 | 2528 | 3463 | 5003 | 6846 |
| 10^{-3} | 3110 | 4417 | 6231 | 8080 | 1057 | 1430 | 1989 | 2709 |
| 10^{-4} | 2936 | 4110 | 5807 | 8037 | 465 | 609 | 836 | 1091 |
| 10^{-5} | 2940 | 4090 | 5741 | 7941 | 208 | 277 | 374 | 489 |
| 10^{-6} | 3185 | 4247 | 5845 | 8021 | 102 | 133 | 187 | 243 |
| 10^{-7} | 6012 | 5845 | 6745 | 8161 | 69 | 75 | 89 | 113 |
| 10^{-8} | 14910 | 15581 | 16618 | 16315 | 70 | 75 | 83 | 84 |

consists of four levels (as shown in Figure 1), and we apply $m = 5$ pre- and post-smoothing steps. In particular, we report the number of outer iterations of the deflated CG method (Out), the total number of multigrid iterations required to solve the inner problem (Tot), and the average number of multigrid inner iterations performed per outer iteration (Inn). In the first block row of Table 3, we present the results obtained by setting the tolerance of the inner system equal to 0.01 times the outer tolerance, i.e., as defined in (4.4) with $c_F = 0.01$. Then, we replace the fixed tolerance used for the inner multigrid solver with an adaptive tolerance. More precisely, at each outer iteration the tolerance for the inner system is set as a function of the outer residual, following the criterion (4.5) with a user-defined parameter $c_A > 0$. In our tests, we explore different values of c_A in order to assess their impact on the behavior of the method. Smaller values of c_A enforce a tighter accuracy for the inner solver, resulting in more robust convergence of the overall method, although at the cost of an increased number of inner iterations. On the other hand, larger values of c_A lead to a faster inner solver and fewer iterations, but also to a higher incidence of convergence failures in the outer deflated CG method (indicated by “-” in Table 3). For comparison, the second block row of Table 3 presents the behavior for $c_A = 0.02$, while in the third block row we report the results for $c_A = 0.01$, and in the fourth block row those corresponding to $c_A = 0.005$. When convergence is achieved, the adaptive tolerance strategy systematically outperforms the fixed tolerance approach in terms of computational cost, as clearly shown by the total number of multigrid iterations (Tot) reported in the corresponding columns.

When using an adaptive tolerance for the inner multigrid solver, certain choices of the user-defined parameter c_A may lead to situations where the deflated CG method fails to converge. To address this issue, we employ the untruncated version of flexible CG, i.e. FCG(∞), for the outer solver. To verify the effectiveness of the FCG approach, we replicate the tests reported in Table 3, this time employing the FCG(∞) variant for the outer solver. The numerical results presented in Table 4 show that, with FCG(∞), all previously critical cases now converge successfully. Moreover, convergence is preserved in all other non-critical cases, indicating that the use of FCG enhances robustness without compromising the performance of the method in well-behaved scenarios.

Remark 5.1. In this work we consider the variant FCG(∞), where no truncation of the search directions is applied. If a mixed truncation–restart strategy [37] is employed, with $m_0 = 0$ and $m_i = \max\{1, \text{mod}(i, m_{\max} + 1)\}$, the performance of the method depends on the user-defined parameters (c_A, m_{\max}). In particular, m_{\max} must be tuned with respect to c_A , as different combinations may lead to either robust or deteriorated convergence. For this reason, we propose the untruncated FCG(∞) approach, which avoids the need to tune the parameters.

Remark 5.2. For completeness, we also tested a GMRES variant with adaptive inner tolerance, following the same strategy as the FCG(∞) counterpart. In all tested configurations, GMRES converged robustly, without exhibiting the convergence issues observed with standard

TABLE 3
Iteration counts for deflated CG for different values of c_F in (4.4) and c_A in (4.5).

| c_F or c_A | Δt | h_1 | | | h_2 | | | h_3 | | | h_4 | | |
|----------------|------------|-------|------|-----|-------|------|-----|-------|-------|-----|-------|-------|-----|
| | | Out | Tot | Inn | Out | Tot | Inn | Out | Tot | Inn | Out | Tot | Inn |
| $c_F = 0.01$ | 10^{-4} | 467 | 6779 | 15 | 623 | 9426 | 15 | 700 | 11100 | 16 | 1102 | 20764 | 19 |
| | 10^{-5} | 212 | 3066 | 14 | 278 | 4232 | 15 | 376 | 5790 | 15 | 489 | 9115 | 19 |
| | 10^{-6} | 102 | 1512 | 15 | 135 | 2039 | 15 | 190 | 2899 | 15 | 238 | 4435 | 19 |
| | 10^{-7} | 69 | 1006 | 15 | 75 | 1143 | 15 | 89 | 1363 | 15 | 114 | 2017 | 18 |
| | 10^{-8} | 70 | 1023 | 15 | 74 | 1118 | 15 | 83 | 1282 | 15 | 83 | 1524 | 18 |
| $c_A = 0.02$ | 10^{-4} | 479 | 2174 | 5 | 638 | 3132 | 5 | 860 | 4196 | 5 | 1127 | 6136 | 5 |
| | 10^{-5} | – | – | – | – | – | – | – | – | – | 503 | 2074 | 4 |
| | 10^{-6} | – | – | – | – | – | – | – | – | – | – | – | – |
| | 10^{-7} | 70 | 203 | 3 | 103 | 243 | 2 | – | – | – | – | – | – |
| | 10^{-8} | 70 | 213 | 3 | 75 | 212 | 3 | 83 | 216 | 3 | 117 | 253 | 2 |
| $c_A = 0.01$ | 10^{-4} | 476 | 2411 | 5 | 634 | 3446 | 5 | 854 | 4635 | 5 | 1118 | 6819 | 6 |
| | 10^{-5} | 217 | 888 | 4 | 319 | 1264 | 4 | 384 | 1616 | 4 | 501 | 2315 | 5 |
| | 10^{-6} | 136 | 406 | 3 | 245 | 575 | 2 | – | – | – | 283 | 900 | 3 |
| | 10^{-7} | 69 | 227 | 3 | 103 | 266 | 3 | 162 | 336 | 2 | 174 | 397 | 2 |
| | 10^{-8} | 70 | 237 | 3 | 74 | 232 | 3 | 83 | 243 | 3 | 85 | 244 | 3 |
| $c_A = 0.005$ | 10^{-4} | 476 | 2635 | 6 | 629 | 3742 | 6 | 854 | 5162 | 6 | 1114 | 7512 | 7 |
| | 10^{-5} | 215 | 971 | 5 | 303 | 1360 | 4 | 382 | 1784 | 5 | 499 | 2555 | 5 |
| | 10^{-6} | 134 | 438 | 3 | 171 | 548 | 3 | 296 | 791 | 3 | 276 | 988 | 4 |
| | 10^{-7} | 69 | 249 | 4 | 78 | 264 | 3 | 90 | 296 | 3 | 148 | 405 | 3 |
| | 10^{-8} | 70 | 261 | 4 | 74 | 263 | 4 | 83 | 268 | 3 | 84 | 269 | 3 |

TABLE 4
Iteration counts for FCG(∞) for different values of c_F in (4.4) and c_A in (4.5).

| c_F or c_A | Δt | h_1 | | | h_2 | | | h_3 | | | h_4 | | |
|----------------|------------|-------|------|-----|-------|------|-----|-------|-------|-----|-------|-------|-----|
| | | Out | Tot | Inn | Out | Tot | Inn | Out | Tot | Inn | Out | Tot | Inn |
| $c_F = 0.01$ | 10^{-4} | 467 | 6779 | 15 | 623 | 9426 | 15 | 700 | 11100 | 16 | 1102 | 20764 | 19 |
| | 10^{-5} | 212 | 3066 | 14 | 278 | 4232 | 15 | 376 | 5790 | 15 | 489 | 9115 | 19 |
| | 10^{-6} | 102 | 1512 | 15 | 135 | 2039 | 15 | 190 | 2899 | 15 | 238 | 4435 | 19 |
| | 10^{-7} | 69 | 1006 | 15 | 75 | 1143 | 15 | 89 | 1363 | 15 | 114 | 2017 | 18 |
| | 10^{-8} | 70 | 1023 | 15 | 74 | 1118 | 15 | 83 | 1282 | 15 | 83 | 1524 | 18 |
| $c_A = 0.02$ | 10^{-4} | 425 | 2016 | 5 | 564 | 2880 | 5 | 764 | 3846 | 5 | 1003 | 5629 | 6 |
| | 10^{-5} | 249 | 825 | 3 | 355 | 1141 | 3 | 500 | 1511 | 3 | 467 | 1980 | 4 |
| | 10^{-6} | 132 | 344 | 3 | 314 | 596 | 2 | 390 | 744 | 2 | 428 | 930 | 2 |
| | 10^{-7} | 70 | 203 | 3 | 102 | 242 | 2 | 281 | 427 | 2 | 276 | 459 | 2 |
| | 10^{-8} | 70 | 213 | 3 | 75 | 212 | 3 | 83 | 216 | 3 | 116 | 252 | 2 |
| $c_A = 0.01$ | 10^{-4} | 423 | 2238 | 5 | 561 | 3159 | 6 | 761 | 4270 | 6 | 996 | 6247 | 6 |
| | 10^{-5} | 206 | 854 | 4 | 294 | 1207 | 4 | 357 | 1543 | 4 | 463 | 2205 | 5 |
| | 10^{-6} | 132 | 400 | 3 | 233 | 560 | 2 | 404 | 823 | 2 | 268 | 875 | 3 |
| | 10^{-7} | 69 | 227 | 3 | 102 | 265 | 3 | 159 | 333 | 2 | 170 | 393 | 2 |
| | 10^{-8} | 70 | 237 | 3 | 74 | 232 | 3 | 83 | 243 | 3 | 84 | 243 | 3 |
| $c_A = 0.005$ | 10^{-4} | 423 | 2443 | 6 | 561 | 3440 | 6 | 759 | 4734 | 6 | 994 | 6881 | 7 |
| | 10^{-5} | 203 | 941 | 5 | 280 | 1308 | 5 | 355 | 1707 | 5 | 462 | 2429 | 5 |
| | 10^{-6} | 130 | 435 | 3 | 164 | 538 | 3 | 280 | 769 | 3 | 261 | 964 | 4 |
| | 10^{-7} | 69 | 249 | 4 | 77 | 263 | 3 | 89 | 295 | 3 | 144 | 401 | 3 |
| | 10^{-8} | 70 | 261 | 4 | 74 | 263 | 4 | 83 | 268 | 3 | 84 | 269 | 3 |

CG under inexact inner solves. GMRES therefore represents a viable alternative to FCG(∞).

5.2. DG Finite Element discretisation - 3D case. In this section, all results are obtained with the FEniCS software. We consider $\Omega = (0, 1)^3$ and we set $\mathbf{F} = \mathbf{0}$ and $\boldsymbol{\sigma}_0 = \mathbf{0}$ as initial condition. We impose $\nabla \cdot \boldsymbol{\sigma} = (\sin(\pi y)\sin(\pi z), 0, 0)^\top$ on $\Gamma_{\text{in}} = \{0\} \times (0, 1) \times (0, 1)$, $\boldsymbol{\sigma} \mathbf{n} = \mathbf{0}$ on $\Gamma_{\text{out}} = \{1\} \times (0, 1) \times (0, 1)$, and $\nabla \cdot \boldsymbol{\sigma} = \mathbf{0}$ in the remaining part of the boundary. For

TABLE 5

Left table: Iteration counts for CG (left) and deflated CG (right) as functions of h and Δt , with $\text{tol} = 10^{-8}$.
 Right table: Iteration counts for deflated CG for mesh 3, for different values of c_F in (4.4) and c_A in (4.5).

| Δt | h_1 | h_2 | h_3 | h_1 | h_2 | h_3 |
|------------|-------|-------|-------|-------|-------|-------|
| 1 | 897 | 1537 | 2947 | 271 | 445 | 823 |
| 0.5 | 767 | 1311 | 2555 | 242 | 394 | 733 |
| 0.1 | 477 | 845 | 1646 | 153 | 273 | 483 |
| 10^{-2} | 252 | 405 | 769 | 58 | 112 | 219 |
| 10^{-3} | 258 | 328 | 523 | 24 | 41 | 75 |
| 10^{-4} | 329 | 417 | 578 | 11 | 17 | 29 |
| 10^{-5} | 362 | 528 | 855 | 7 | 9 | 13 |

| Δt | $c_F = 0.01$ | | | $c_A = 0.02$ | | | $c_A = 0.005$ | | |
|------------|--------------|-------|-----|--------------|-------|-----|---------------|-------|-----|
| | Out | Tot | Inn | Out | Tot | Inn | Out | Tot | Inn |
| 0.5 | 972 | 31625 | 33 | 1081 | 12687 | 12 | 1017 | 14494 | 14 |
| 0.1 | 522 | 16656 | 32 | 603 | 7684 | 13 | 559 | 8741 | 16 |
| 10^{-2} | 219 | 6595 | 30 | 242 | 2905 | 12 | 220 | 3344 | 15 |
| 10^{-3} | 75 | 2148 | 29 | 75 | 928 | 12 | 75 | 1081 | 14 |
| 10^{-4} | 29 | 824 | 28 | 29 | 342 | 12 | 29 | 407 | 14 |

the numerical simulations, we set $\mu = 0.5$, the penalty coefficient $\alpha^* = 40$, we fix $p = 1$ and $\alpha = \Delta t$ (implicit Euler scheme in time). We consider three nested meshes: mesh 1 consisting of 48 tetrahedra ($h_1 \approx 0.8660$), mesh 2 made by 384 tetrahedra ($h_2 \approx 0.4330$), and mesh 3 with 3072 tetrahedra ($h_3 \approx 0.2165$). These meshes are first employed to compare the performance of CG and deflated CG methods, and are then used to construct the multigrid hierarchy, repeating the tests carried out in Section 5.1 for the 2D case. In particular, in Table 5 (left) we report the iteration counts needed by the CG and the deflated CG algorithms to reduce the relative residual below 10^{-8} , for different time steps Δt and mesh sizes h . In this case, in the deflated CG strategy, the inner system is solved exactly by a direct solver. The 3D results exhibit the same trend observed in 2D: CG deteriorates with mesh refinement and as $\Delta t \rightarrow 0$, while the deflated CG remains uniformly efficient and robust with respect to Δt .

We now repeat the same set of tests for the deflated CG method, where the inner system is now solved by a multigrid W-cycle approach with RAS as smoother, in the 3D setting. In this case, the finest grid used is the one consisting of 3072 tetrahedral elements (mesh 3). The multigrid hierarchy is therefore built on the three nested levels (mesh 1 \subset mesh 2 \subset mesh 3), and we employ $m = 10$ pre- and post-smoothing steps. In Table 5 (right) we test one value of the fixed parameter c_F and two values of the adaptive parameter c_A to assess their effect on the efficiency and robustness of the method. The observed results are consistent with the 2D case, confirming the same qualitative behavior.

In Figure 2, we report the pressure and velocity field corresponding to the test case introduced above. The pressure is reconstructed from the numerical pseudo-stress solution through the relation $p = -\frac{1}{d} \text{tr}(\sigma)$, while the velocity is recovered by using $\mathbf{u}(t) = \mathbf{u}_0 + \int_0^t (\nabla \cdot \sigma(s)) ds$. The results shown in the figure are obtained on the finest mesh of the hierarchy (i.e., mesh 3 consisting of 3072 tetrahedral elements) with a time step $\Delta t = 0.5$. To reconstruct the velocity, a single implicit Euler step is performed. The simulation is carried out using the deflated CG method, with the inner linear systems solved by a three-level nested multigrid algorithm and an adaptive inner tolerance (with $c_A = 0.02$). The plots confirm that the proposed approach is able to accurately reproduce the expected physics of the system.

6. Conclusions. In this work, we have proposed a numerical solver for the linear system arising from conforming and discontinuous finite element discretisations of the pseudo-stress unsteady Stokes problem. The main difficulty is related to the deterioration of the conditioning as the time step decreases. To address this issue, we introduce a deflation-based strategy that removes the contribution of the kernel components, thereby ensuring robustness with respect to the time step. The resulting inner system is solved through an efficient multigrid solver. Since the inner system is solved only approximately, we employ a flexible Conjugate Gradient method to guarantee stable convergence of the overall algorithm. As a result, the proposed approach combines deflation, multigrid acceleration, and flexible Krylov techniques into a scalable solver. The robustness of the method is confirmed by numerical results, which show

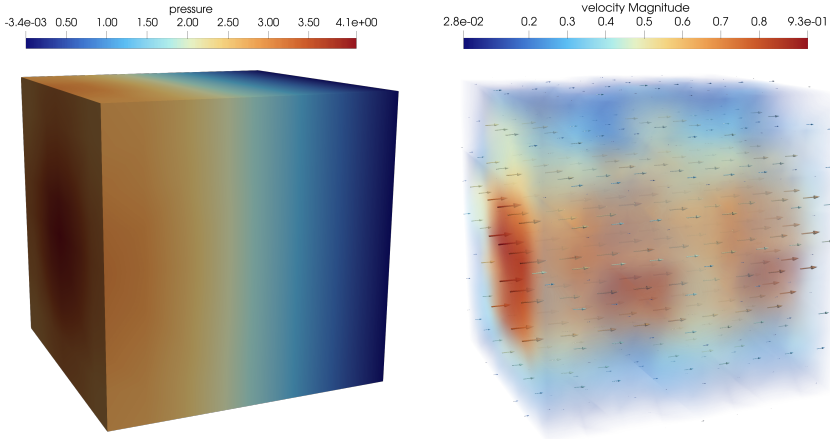


FIG. 2. Test case of Section 5.2. Left: computed pressure field p_h by means of the relation $p_h = -\frac{1}{3}\text{tr}(\sigma_h)$. Right: computed velocity field \mathbf{u}_h by means of the relation $\mathbf{u}_h = \mathbf{u}_{0,h} + \int_0^T \nabla \cdot \sigma_h(s) ds$, with final time $T = 0.5$.

stable convergence with respect to the time step. Possible further developments may include the investigation of multigrid methods as preconditioners for the outer Conjugate Gradient solver, in order to achieve robustness also with respect to the mesh size. Another interesting direction is the extension of the proposed solver to other time discretisation schemes, including Runge–Kutta methods, for the pseudo-stress Stokes problem.

Acknowledgments. This work is funded by the European Union (ERC SyG, NEMESIS, project number 101115663). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. All authors are members of the Indam GNCS group.

REFERENCES

- [1] P. F. ANTONIETTI, S. BONETTI, M. BOTTI, M. CORTI, I. FUMAGALLI, AND I. MAZZIERI, *lymph: discontinuous polytopal methods for Multi-PHysics differential problems*, ACM Trans. Math. Softw., 51 (2025), pp. 1–22.
- [2] P. F. ANTONIETTI, M. BOTTI, A. CANCRINI, AND I. MAZZIERI, *A polytopal discontinuous Galerkin method for the pseudo-stress formulation of the unsteady Stokes problem*, Comput. Methods Appl. Mech. Eng., 447 (2025), p. 118404.
- [3] P. F. ANTONIETTI, A. CANCRINI, AND G. CIARAMELLA, *Numerical verification of PolyDG algebraic solvers for the pseudo-stress Stokes problem*, arXiv preprint, (2025). arXiv:2512.02537. Accepted in DD29 Proceedings, Springer.
- [4] P. F. ANTONIETTI, P. HOUSTON, M. SARTI, AND M. VERANI, *Multigrid algorithms for hp-version interior penalty discontinuous Galerkin methods on polygonal and polyhedral meshes*, Numer. Math., 54 (2014), pp. 2009–2041.
- [5] P. F. ANTONIETTI AND G. PENNESI, *V-cycle multigrid algorithms for discontinuous Galerkin methods on non-nested polytopic meshes*, Journal of Sci. Comp., 78 (2017), pp. 625 – 652.
- [6] P. F. ANTONIETTI, M. SARTI, M. VERANI, AND L. T. ZIKATANOV, *A uniform additive Schwarz preconditioner for high-order discontinuous Galerkin approximations of elliptic problems*, J. of Sci. Comp., 70 (2017), pp. 608–630.
- [7] D. N. ARNOLD, J. J. DOUGLAS, AND C. P. GUPTA, *A family of higher order mixed finite element methods for plane elasticity*, Numer. Math., 45 (1984), pp. 1–22.
- [8] D. N. ARNOLD AND R. WINTER, *Mixed finite elements for elasticity*, Numer. Math., 92 (2002), pp. 401–419.
- [9] O. AXELSSON, *Iterative Solution Methods*, Cambridge University Press, 1996.
- [10] F. P. T. BAAIJENS, *Mixed finite element methods for viscoelastic flow analysis: A review*, J. Non-Newtonian Fluid Mech., 79 (1998), pp. 361–385.

- [11] J. H. BRAMBLE, R. E. EWING, J. E. PASCIAK, AND J. SHEN, *The analysis of multigrid algorithms for cell centered finite difference methods*, *Advances in Comp. Math.*, 5 (1996), pp. 15–29.
- [12] J. H. BRAMBLE AND J. E. PASCIAK, *The analysis of smoothers for multigrid algorithms*, *Math. of Comp.*, 58 (1992), pp. 467–488.
- [13] J. H. BRAMBLE, J. E. PASCIAK, AND J. XU, *The analysis of multigrid algorithms with nonnested spaces or noninherited quadratic forms*, *Math. of Comp.*, 56 (1991), pp. 1–34.
- [14] F. BREZZI, J. DOUGLAS, R. DURAN, AND M. FORTIN, *Mixed finite elements for second order elliptic problems in three variables*, *Numer. Math.*, 51 (1987), pp. 237–250.
- [15] F. BREZZI, J. DOUGLAS, AND L. MARINI, *Two families of mixed finite elements for second order elliptic problems*, *Numer. Math.*, 47 (1985), pp. 217–235.
- [16] X.-C. CAI AND M. V. SARKIS, *A restricted additive Schwarz preconditioner for general sparse linear systems*, *SIAM J. Sci. Comput.*, 21 (1999), pp. 792–797.
- [17] Z. CAI, B. LEE, AND P. WANG, *Least-squares methods for incompressible Newtonian fluid flow: Linear stationary problems*, *SIAM J. Numer. Anal.*, 42 (2004), pp. 843–859.
- [18] Z. CAI, C. TONG, P. VASSILEVSKI, AND C. WANG, *Mixed finite element methods for incompressible flow: Stationary Stokes equations*, *Numer. Meth. Partial Diff. Equ.*, 26 (2009), pp. 957 – 978.
- [19] Z. CAI AND Y. WANG, *A multigrid method for the pseudostress formulation of Stokes problems*, *SIAM J. on Sci. Comp.*, 29 (2007), pp. 2078–2095.
- [20] A. CANCRINI, *A polytopal discontinuous Galerkin method for the Stokes-Brinkman problem with application to the modeling of the glymphatic system*, master’s thesis, Politecnico di Milano - Dipartimento di Matematica, Milan (Italy), a.y. 2022/2023.
- [21] A. CANGIANI, Z. DONG, AND E. H. GEORGIOULIS, *hp-version discontinuous Galerkin methods on essentially arbitrarily-shaped elements*, *Math. of Comp.*, 91 (2022), pp. 1–35.
- [22] A. CANGIANI, Z. DONG, E. H. GEORGIOULIS, AND P. HOUSTON, *hp-version discontinuous Galerkin methods on polygonal and polyhedral meshes*, *SpringerBriefs Math.*, Springer, 2017.
- [23] A. CANGIANI, E. H. GEORGIOULIS, AND P. HOUSTON, *hp-version discontinuous Galerkin methods on polygonal and polyhedral meshes*, *Math. Models and Methods in Appl. Sciences*, 24 (2014), pp. 2009–2041.
- [24] G. CIARAMELLA AND M. J. GANDER, *Iterative methods and preconditioners for systems of linear equations*, *Fundam. Algorithms*, SIAM, 2022.
- [25] Z. DOSTÁL, *Conjugate gradient method with preconditioning by projector*, *Int. J. of Comp. Math.*, 23 (1988), pp. 315–323.
- [26] J. FRANK AND C. VUIK, *On the construction of deflation-based preconditioners*, *SIAM J. Sci. Comput.*, 23 (2001), pp. 442–462.
- [27] M. J. GANDER, *Schwarz methods over the course of time*, *Electronic Transactions on Numer. Anal.*, 31 (2008), pp. 228–255.
- [28] A. GAUL, M. GUTKNECHT, J. LIESEN, AND R. NABBEN, *A framework for deflated and augmented Krylov subspace methods*, *SIAM J. Matrix Anal. Appl.*, 34 (2013), pp. 495–518.
- [29] M. I. GERRITSMAN AND T. N. PHILLIPS, *Compatible spectral approximations for the velocity-pressure-stress formulation of the Stokes problem*, *SIAM J. Sci. Comput.*, 20 (1999), pp. 1530–1550.
- [30] J. GOPALAKRISHNAN AND G. KANSCHAT, *A multilevel discontinuous Galerkin method*, *Numer. Math.*, 95 (2003), pp. 527–550.
- [31] M. R. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, *J. Res. Natl. Bur. Stand.*, 49 (1952), pp. 409–435.
- [32] K. KAHL AND H. RITTICH, *The deflated conjugate gradient method: Convergence, perturbation and accuracy*, *Linear Algebra Appl.*, 515 (2017), pp. 111–129.
- [33] R. KEUNINGS, *Advances in the computer modeling of the flow of polymeric liquids*, *Comp. Fluid Dyn. J.*, 9 (2001), pp. 449–458.
- [34] R. NABBEN AND C. VUIK, *A comparison of deflation and the balancing preconditioner*, *SIAM J. on Sci. Comput.*, 27 (2006), pp. 1742–1759.
- [35] J. C. NÉDÉLEC, *Mixed finite elements in R^3* , *Numer. Math.*, 35 (1980), pp. 315–341.
- [36] R. A. NICOLAIDES, *Deflation of conjugate gradients with applications to boundary value problems*, *SIAM J. Numer. Anal.*, 24 (1987), pp. 355–365.
- [37] Y. NOTAY, *Flexible conjugate gradients*, *SIAM J. on Sci. Comp.*, 22 (2000), pp. 1444–1460.
- [38] W. QIU AND L. ZHAO, *$H(\text{div})$ -conforming HDG methods for the stress-velocity formulation of the Stokes equations and the Navier–Stokes equations*, *Numer. Math.*, 156 (2024), pp. 1639–1678.
- [39] P. RAVIART AND J. THOMAS, *A mixed finite element method for second order elliptic problems*, *Math. Aspects of Finite Element Methods*, (1977), pp. 292–315.
- [40] Y. SAAD, *Iterative methods for sparse linear systems*, 2nd ed., SIAM, 2003.
- [41] Y. SAAD, M. YEUNG, J. ERHEL, AND F. GUYOMARCH, *A deflated version of the conjugate gradient algorithm*, *SIAM J. Sci. Comput.*, 21 (2000), pp. 1909–1926.
- [42] A. VAN DER SLUIS AND H. VAN DER VORST, *The rate of convergence of conjugate gradients*, *Numer. Math.*, 48 (1986), pp. 543–560.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

- 44/2026** Bonetti, S; Botti, M.; Antonietti, P.F.
Splitting strategies for the fully-coupled nonlinear thermo-hydro-mechanical problem
- 45/2026** Antonietti, P.F.; Botti, M.; Parolini, N.; Pederzoli, V.; Verani, M.
Polytopal Discontinuous Galerkin Discretizations of Coupled Non-Newtonian Stokes-Darcy Systems
- 43/2026** Micheletti, S.
A validated MATLAB framework for sparse vectorized finite element assembly
- 42/2026** Fumagalli, I.; Campioni, M.; Sirtori, A.; Pagani, S.; Levi, R.; Politi, L. S.; Capo, G.; Antonietti, P. F.
Patient-specific computational mechanics of functional lumbar spine units
- 41/2026** Sosta, L.; Ciancarelli, C; Marini, L.; Pagani, S.; Regazzoni, F.; Parolini, N.
Physics-constrained identification of graph-based thermal networks for spacecraft digital twins
- 40/2026** Marchesin, L.; Menafoglio, A.; Secchi, P.
A Convolution Process for Sea Surface Temperature Hot-Spot Identification in the Mediterranean Sea
- 39/2026** Patanè, G.; Menafoglio, A.; Krauth, A.; Fechner, P.; Dede', L.; Colosimo, B.M.; Nicolussi, F.
K-Models: a Flexible and Interpretable Method for Ordinal Clustering with Application to Antigen-Antibody Interaction Profiles
- 38/2026** Clemente, A.; Arnone, E.; Mateu, J.; Sangalli, L.M.
Nonparametric estimators over metric graphs
- 37/2026** Centofanti, E.; Ziarelli, G.; Scacchi, S.; Pavarino, L.F.
A Neural Latent Dynamics Approach for Solving Inverse Problems in Cardiac Electrophysiology
- 36/2026** Botti, M.; Mascotto, L.; Mosconi, M.
A nonconforming method for a generalized Darcy-Forchheimer model