MOX-Report No. 72/2025

Numerical verification of PolyDG algebraic solvers for the pseudo-stress Stokes problem

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

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

mox-dmat@polimi.it https://mox.polimi.it

Numerical verification of PolyDG algebraic solvers for the pseudo-stress Stokes problem

Paola F. Antonietti, Alessandra Cancrini and Gabriele Ciaramella

Abstract This work focuses on the development of efficient solvers for the pseudo-stress formulation of the unsteady Stokes problem, discretised by means of a discontinuous Galerkin method on polytopal grids (PolyDG). The introduction of the pseudo-stress variable is motivated by the growing interest in non-Newtonian flow models and coupled interface problems, where the stress field plays a fundamental role in the physical description. The space-time discretisation of the problem is obtained by combining the PolyDG approach in space with the implicit Euler method for time integration. The resulting linear system, characterised by a symmetric, positive, definite matrix, exhibits deteriorating convergence with standard solvers as the time step decreases. To address this issue, we investigate two tailored strategies: deflated Conjugate Gradient, which mitigates the effect of the most problematic eigenmodes, and collective Block-Jacobi, which exploits the block structure of the system matrix. Numerical experiments show that both approaches yield iteration counts effectively independent of Δt, ensuring robust performance with respect to the time step. Future work will focus on extending this robustness to the spatial discretisation parameter h by integrating multigrid strategies with the time-robust solvers developed in this study.

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.

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

1 Introduction

The numerical approximation of incompressible viscous flows is a fundamental problem in computational fluid dynamics. In this setting, the pseudo-stress formulation of the unsteady Stokes equations [1] has attracted attention for its relevance to non-Newtonian models and coupled interface problems, where an accurate representation of the stress tensor is essential. To discretise the problem, we employ a discontinuous Galerkin method on polytopal meshes (PolyDG) and the implicit Euler method for time integration. At each time step, this discretisation leads to a linear system whose matrix is symmetric and positive definite. When refining the mesh or varying the time step, it is crucial to rely on linear solvers and preconditioners whose performance remains stable with respect to the discretisation parameters. The main goal of this work is to investigate solver robustness with respect to the time step Δt. As shown by numerical experiments, both the Conjugate Gradient (CG) method [2] and the CG preconditioned with standard Block-Jacobi suffer from a severe deterioration of convergence as Δt decreases. This behaviour is due to the conditioning of the system matrix, which scales as $1/\Delta t$; therefore, smaller time steps produce increasingly ill-conditioned systems and, consequently, a larger number of iterations. The structure of the system matrix motivates the use of two tailored solvers: Deflated CG [3, 4], which accelerates convergence by eliminating the most problematic eigenmodes, and collective Block-Jacobi [5], which treats pseudo-stress components collectively at the element level. Numerical tests show that both approaches yield iteration counts independent of Δt, thereby achieving the desired robustness. This study represents a first step toward a solver that is robust with respect to both Δt and h, with future work incorporating multigrid techniques for polytopal discretisations to further enhance robustness under spatial refinement.

2 Model problem and numerical discretization

2.1 Pseudo-stress weak formulation

We focus on the Stokes problem, which models incompressible viscous free flows, and formulate it in terms of a pseudo-stress unknown rather than its classical expression. The derivation of the pseudo-stress formulation for the unsteady Stokes problem is presented in [1]. The pseudo-stress variable is defined as $\boldsymbol{\sigma}(\boldsymbol{u}, p) = \mu \boldsymbol{\nabla} \boldsymbol{u} - p\mathbb{I}_d$, where $\boldsymbol{u}$ is the flow velocity and p its pressure. Let $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, be an open, convex polygonal/polyhedral domain with Lipschitz boundary $\partial\Omega$, and let $\mathbf{dev}(\boldsymbol{\tau}) = \boldsymbol{\tau} - \frac{1}{d}\mathrm{tr}(\boldsymbol{\tau})\mathbb{I}_d$ be the deviatoric operator, where $\mathrm{tr}(\cdot)$ is the trace operator. Then, the Stokes problem can be rewritten as

$$\begin{cases} \frac{1}{\mu}\frac{\partial \mathbf{dev}(\boldsymbol{\sigma})}{\partial t} - \boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}) = \boldsymbol{F}, & \text{in } \Omega\times(0,T], \\ \boldsymbol{\nabla}\cdot\boldsymbol{\sigma} = \boldsymbol{g}_D, & \text{on } \Gamma_D\times(0,T], \\ \boldsymbol{\sigma}\,\boldsymbol{n} = \boldsymbol{g}_N, & \text{on } \Gamma_N\times(0,T], \\ \mathbf{dev}(\boldsymbol{\sigma})(\cdot,t=0) = \boldsymbol{\sigma}_0, & \text{in } \Omega, \end{cases}$$

where $\mu > 0$ is the fluid viscosity, and $T > 0$ the final simulation time. 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. We also assume that $\boldsymbol{F} \in \mathbb{H}^1(\Omega)$, $\boldsymbol{g}_N \in \boldsymbol{L}^2(\Gamma_N)$, $\boldsymbol{\sigma}_0 \in \mathbb{L}^2(\Omega)$ and that for any time t the Dirichlet datum $\boldsymbol{g}_D = \boldsymbol{g}_D(t)$ is the trace of a function in $\boldsymbol{H}^1(\Omega)$. To strongly enforce the essential traction condition on Γ_N, we define the subspace

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

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

$$(\mu^{-1}\partial_t \mathbf{dev}(\boldsymbol{\sigma}), \mathbf{dev}(\boldsymbol{\tau}))_\Omega + (\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}, \boldsymbol{\nabla}\cdot\boldsymbol{\tau})_\Omega = (\boldsymbol{F},\boldsymbol{\tau})_\Omega + \langle \boldsymbol{g}_D, \boldsymbol{\tau}\,\boldsymbol{n}\rangle_{\partial\Omega} \qquad (1)$$

for any $\boldsymbol{\tau} \in \mathbb{H}_{0,\Gamma_N}(\mathrm{div},\Omega)$. For further details, the reader is referred to [1].

2.2 PolyDG semi-discrete formulation

We can now introduce the PolyDG semi-discrete formulation of (1). 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 $p_h : \mathcal{T}_h \to \mathbb{N}^* = \{n \in \mathbb{N} : n \geq 1\}$ the piecewise constant function such that $(p_h)|_\kappa = p_\kappa$. Then, we define the discrete space $\boldsymbol{V}_h = [P_{p_h}(\mathcal{T}_h)]^{d\times d}$, where $P_{p_h}(\mathcal{T}_h) = \Pi_{\kappa\in\mathcal{T}_h}\mathbb{P}_{p_\kappa}(\kappa)$, 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 define an interface as the intersection of the $(d-1)$-dimensional faces of any two neighboring elements of $\mathcal{T}_h$. 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, respectively. We refer the reader to [1, 6] for the main assumptions on $\mathcal{T}_h$. Finally, for sufficiently piecewise smooth vector- and tensor-valued fields $\boldsymbol{v}$ 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

$$[[\boldsymbol{v}]] = \boldsymbol{v}^+\otimes\boldsymbol{n}^+ + \boldsymbol{v}^-\otimes\boldsymbol{n}^-, \ [[\boldsymbol{\tau}]] = \boldsymbol{\tau}^+\boldsymbol{n}^+ + \boldsymbol{\tau}^-\mathbf{n}^-, \ \{\{\boldsymbol{v}\}\} = \frac{\boldsymbol{v}^+ + \boldsymbol{v}^-}{2}, \ \{\{\boldsymbol{\tau}\}\} = \frac{\boldsymbol{\tau}^+ + \boldsymbol{\tau}^-}{2},$$

where $\otimes$ is the tensor product in $\mathbb{R}^d$, $\cdot^\pm$ denotes the trace on F taken within $\kappa^\pm$, and $\boldsymbol{n}^\pm$ 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 $[[\boldsymbol{v}]] = \boldsymbol{v} \otimes \boldsymbol{n}$, $[[\boldsymbol{\tau}]] = \boldsymbol{\tau}\boldsymbol{n}$, and $\{\{\boldsymbol{v}\}\} = \boldsymbol{v}$, $\{\{\boldsymbol{\tau}\}\} = \boldsymbol{\tau}$. In the following, we use $\nabla_h \cdot$ to denote the element-wise divergence operator, and we use the short-hand notation $(\cdot,\cdot)_{\mathcal{T}_h} = \sum_{\kappa\in\mathcal{T}_h}\int_\kappa \cdot$ and $\langle\cdot,\cdot\rangle_{\mathcal{F}_h} = \sum_{F\in\mathcal{F}_h}\int_F \cdot$. We consider the following semi-discrete PolyDG approximation to (1): for any $t \in (0,T]$, find $\boldsymbol{\sigma}_h(t) \in \boldsymbol{V}_h$ s.t.

$$\begin{cases} \mathcal{M}(\partial_t\boldsymbol{\sigma}_h,\boldsymbol{\tau}_h) + \mathcal{A}(\boldsymbol{\sigma}_h,\boldsymbol{\tau}_h) = F(\boldsymbol{\tau}_h) & \forall\,\boldsymbol{\tau}_h \in \boldsymbol{V}_h, \\ (\boldsymbol{\sigma}_h(0),\boldsymbol{\tau}_h) = (\boldsymbol{\sigma}_0,\boldsymbol{\tau}_h) & \forall\,\boldsymbol{\tau}_h \in \boldsymbol{V}_h, \end{cases} \tag{2}$$

where for any $\boldsymbol{\sigma},\boldsymbol{\tau} \in \boldsymbol{V}_h$ we have defined

$$\begin{aligned} \mathcal{M}(\boldsymbol{\sigma},\boldsymbol{\tau}) &= (\mu^{-1}\mathbf{dev}(\boldsymbol{\sigma}),\mathbf{dev}(\boldsymbol{\tau}))_{\mathcal{T}_h}, \\ \mathcal{A}(\boldsymbol{\sigma},\boldsymbol{\tau}) &= (\boldsymbol{\nabla}_h\cdot\boldsymbol{\sigma},\boldsymbol{\nabla}_h\cdot\boldsymbol{\tau})_{\mathcal{T}_h} - \langle\{\{\boldsymbol{\nabla}_h\cdot\boldsymbol{\sigma}\}\},[[\boldsymbol{\tau}]]\rangle_{\mathcal{F}_h^{I,N}} \\ &\quad -\langle\{\{\boldsymbol{\nabla}_h\cdot\boldsymbol{\tau}\}\},[[\boldsymbol{\sigma}]]\rangle_{\mathcal{F}_h^{I,N}} + \langle\gamma_e[[\boldsymbol{\sigma}]],[[\boldsymbol{\tau}]]\rangle_{\mathcal{F}_h^{I,N}}, \\ F(\boldsymbol{\tau}) &= (\boldsymbol{F},\boldsymbol{\tau})_{\mathcal{T}_h} + \langle\boldsymbol{g}_D,\boldsymbol{\tau}\boldsymbol{n}\rangle_{\mathcal{F}_h^D} + \langle\boldsymbol{g}_N,\gamma_e\boldsymbol{\tau}\boldsymbol{n} + (\boldsymbol{\nabla}_h\cdot\boldsymbol{\tau})\rangle_{\mathcal{F}_h^N}. \end{aligned}$$

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

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

2.3 PolyDG fully-discrete formulation

We introduce a basis $\{\phi_i, i = 1,\dots,N_h\}$ for the space $\boldsymbol{V}_h$ and express $\boldsymbol{\sigma}_h$ as a linear combination of these basis functions, where the unknown coefficients are functions of time. We collect the latter in the vector $\boldsymbol{\sigma}_h$, denote by M (resp. A) the matrix representation of the bilinear form $\mathcal{M}(\cdot,\cdot)$ (resp. $\mathcal{A}(\cdot,\cdot)$), and by $\boldsymbol{f}$ the vector representation of the linear functional $F(\cdot)$. The algebraic formulation of (2) reduces to: for any time $t \in (0,T]$, find $\boldsymbol{\sigma}_h(t) \in \boldsymbol{V}_h$ s.t.

$$M\dot{\boldsymbol{\sigma}}_h(t) + A\boldsymbol{\sigma}_h(t) = \boldsymbol{f}(t) \quad \forall t \in (0,T], \tag{3}$$

with initial condition $\boldsymbol{\sigma}_h(0) = \boldsymbol{\sigma}_{0,h}$, being the latter the vector representation of the $\boldsymbol{L}^2$-projection of $\boldsymbol{\sigma}_0$ onto $\boldsymbol{V}_h$. To integrate in time (3) we use the θ-method. Note that, in this framework, an explicit time-stepping scheme cannot be employed since it would lead to an algebraic system with a singular matrix. So, we discretize our problem by applying the implicit Euler method (i.e. for $\theta = 1$), and we obtain: for any $n = 1,\dots,N_T$ find $\boldsymbol{\sigma}_h^{n+1}$ such that

$$A^*\,\boldsymbol{\sigma}_h^{n+1} = \boldsymbol{f}^* \tag{4}$$

with $N_T = \Delta t/T$, and where the superscript n means the approximation/evaluation of the given quantity at time $t_n = n\Delta t$, $n = 0, \dots, N_T$. Here, we have introduced the operator $A^* := M + \Delta t A$, together with the corresponding bilinear form $\mathcal{A}^*(\cdot,\cdot) := \mathcal{M}(\cdot,\cdot) + \Delta t \mathcal{A}(\cdot,\cdot)$, which defines a symmetric and positive definite operator. Furthermore, we have defined the right-hand side vector $\boldsymbol{f}^* := M\boldsymbol{\sigma}_h^n + \Delta t \boldsymbol{f}^{n+1}$, with the associated operator $\mathcal{F}^*(\cdot) := \mathcal{M}(\boldsymbol{\sigma}^n, \cdot) + \Delta t \mathcal{F}(\cdot)$. We also introduce the vector of unknowns

$$\boldsymbol{\sigma} = \left[\boldsymbol{\sigma}_{11}^\top \ \boldsymbol{\sigma}_{12}^\top \ \boldsymbol{\sigma}_{21}^\top \ \boldsymbol{\sigma}_{22}^\top\right]^\top. \tag{5}$$

The matrix A^* in (4) is defined in terms of M and A, where M is positive semi-definite and A is positive definite. Their structures are given by:

$$M = \begin{bmatrix} \frac{1}{2} & 0 & 0 & -\frac{1}{2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{1}{2} & 0 & 0 & \frac{1}{2} \end{bmatrix} \otimes M_1, \qquad A = \begin{bmatrix} B_1 & B_2 & 0 & 0 \\ B_2 & B_3 & 0 & 0 \\ 0 & 0 & B_1 & B_2 \\ 0 & 0 & B_2 & B_3 \end{bmatrix},$$

where $M_1, B_1, B_2, B_3 \in \mathbb{R}^{\frac{n}{4}\times\frac{n}{4}}$. In particular, $M_1(i,j) = \int_\Omega \phi_j \phi_i$, while $B_1(i,j) \approx \int_\Omega \partial_x\phi_j \, \partial_x\phi_i$, $B_2(i,j) \approx \int_\Omega \partial_x\phi_j \, \partial_y\phi_i$, and $B_3(i,j) \approx \int_\Omega \partial_y\phi_j \, \partial_y\phi_i$.

3 Numerical solvers

In this section, we study numerical solvers for the linear system (4) arising from the proposed discretisation. Since the system matrix A^* is symmetric and positive definite, the CG method is the natural solver. However, A^* depends on the singular matrix M; therefore, as the time step $\Delta t \to 0$, the conditioning of A^* deteriorates and the number of CG iterations increases. To obtain robustness with respect to Δt, we consider two strategies: Deflated CG, which removes the influence of the kernel of M, and CG preconditioned with collective Block-Jacobi, which exploits the block structure of the matrix to improve convergence.

3.1 The Deflated CG method

Consider a linear system of equations of the form (4), where $A^* \in \mathbb{R}^{n\times n}$ is symmetric and positive definite and $\boldsymbol{\sigma}_h^{n+1}$, $\boldsymbol{f}^* \in \mathbb{R}^n$. To simplify the notation, we denote by $\boldsymbol{x}$ the solution $\boldsymbol{\sigma}_h^{n+1}$ of the linear system. It is well known that the speed of convergence of CG depends on the condition number of the matrix A^* [2, 5], and to improve the rate of convergence, it can become mandatory to use a preconditioner. A possible way to precondition is via deflation [3, 4]. Since the speed of convergence of CG depends on the distribution of the eigenvalues of A^*, the idea of deflation 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 $\mathbf{S} \subset \mathbb{R}^n$, 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 $\boldsymbol{x}$ into a component in $\mathbf{S}$ and a component in $\mathbf{S}^{\perp_{A^*}}$ via the A^*-orthogonal projection $\pi_{A^*}(\mathbf{S}) \in \mathbb{R}^{n\times n}$ onto $\mathbf{S}$. Given a matrix $V \in \mathbb{R}^{n\times n}$ whose columns form a basis for $\mathbf{S}$, and $\pi_{A^*}(\mathbf{S}) = V(V^\top A^* V)^{-1} V^\top A^*$, we obtain

$$\boldsymbol{x} = (I - \pi_{A^*}(\mathbf{S}))\,\hat{\boldsymbol{x}} + V(V^\top A^* V)^{-1} V^\top \boldsymbol{f}^*, \tag{6}$$

where $\hat{\boldsymbol{x}}$ is the solution of the so-called *deflated linear system*

$$A^*(I - \pi_{A^*}(\mathbf{S}))\,\hat{\boldsymbol{x}} = (I - \pi_{A^*}(\mathbf{S}))^\top \boldsymbol{f}^*. \tag{7}$$

Thus,to obtain $\boldsymbol{x}$, one needs to compute the solution $\hat{\boldsymbol{x}}$ of the deflated system. The matrix $A^*(I - \pi_{A^*}(\mathbf{S}))$ is symmetric positive semi-definite [4] and so we can apply the CG method. Matrix singularity poses no obstacle to the standard CG iteration, provided that (7) is consistent, i.e., the right hand side $(I-\pi_{A^*}(\mathbf{S}))^\top \boldsymbol{f}^*$ is in the range of $A^*(I - \pi_{A^*}(\mathbf{S}))$. To summarise, one interprets deflated CG as the standard CG algorithm applied to the deflated system (7). In order to define the A^*-orthogonal projection $\pi_{A^*}(\mathbf{S})$, we introduce V as the kernel of the matrix M:

$$V = \ker(M) = \operatorname{span}\left\{ \frac{1}{\sqrt{2}} \begin{bmatrix} I \\ 0 \\ 0 \\ I \end{bmatrix} \otimes M_1 \right\}.$$

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. Note that applying the deflated CG method requires computing the inverse of $V^\top A^* V$ in (6). It can be shown that this term is approximately $\frac{\Delta t}{2}(B_1 + B_3)$, which corresponds to a Laplacian-type operator. Inverting this operator must therefore be handled appropriately, but this aspect lies outside the scope of the present work.

3.2 Collective Block-Jacobi as preconditioner for CG

To solve the linear system (4), the standard Block-Jacobi method is commonly used as a preconditioner for the CG method. In this approach, the diagonal blocks are defined by separating each component of the solution vector as in (5), so that the components are treated independently across the elements. Accordingly, for each component $\boldsymbol{\sigma}_{ij}$, all elements appear consecutively in the global ordering. In contrast, the collective Block-Jacobi approach groups the four components associated with a

given element into a single block, so that the unknowns belonging to element κ are kept adjacent. It can be shown that the resulting permuted matrix is block diagonally dominant, in agreement with the definition in [7, 8], and this also explains the improved robustness with respect to Δt observed in the numerical tests.

4 Numerical results

In this section, we present numerical experiments aimed at assessing the performance of the proposed strategies for solving the algebraic system (4) at each time step, implemented in the open source MATLAB library `lymph` [9]. For all numerical tests, the parameters are fixed as follows: the polynomial degree is set to $p = 3$, the domain considered is $\Omega = (0,1)^2$ and the polytopal meshes consist of 50 elements ($h_0 \approx 0.2462$), 100 elements ($h_1 \approx 0.1759$), 200 elements ($h_2 \approx 0.1260$), 400 elements ($h_3 \approx 0.0909$), and 800 elements ($h_4 \approx 0.0637$). The grey-shaded entries in the tables indicate the regime where the time step and mesh size are balanced according to $\|\boldsymbol{\sigma} - \boldsymbol{\sigma}_h\|_E \sim \Delta t + h^p$ [1], corresponding to the most stable and efficient solver performance. First, we compute the condition number of A^* for these five values of h and different time steps Δt (left part of Table 1), observing that for small Δt the condition number is proportional to $1/\Delta t$. The right part of Table 1 reports the condition number of the matrix preconditioned with the collective Block-Jacobi method, highlighting the improvement in conditioning due to the preconditioner. Additional numerical tests (not shown here) confirm that the condition number of the standard Block–Jacobi preconditioned matrix exhibits the same dependence on Δt as that of A^*.

Δt	h_0	h_1	h_2	h_3	h_4	h_0	h_1	h_2	h_3	h_4
10^{-2}	$8.5{\cdot}10^5$	$1.6{\cdot}10^6$	$3.0{\cdot}10^6$	$6.5{\cdot}10^6$	$1.3{\cdot}10^7$	$7.2{\cdot}10^5$	$1.2{\cdot}10^6$	$2.7{\cdot}10^6$	$8.0{\cdot}10^6$	$1.3{\cdot}10^7$
10^{-3}	$5.6{\cdot}10^5$	$1.0{\cdot}10^6$	$1.9{\cdot}10^6$	$3.9{\cdot}10^6$	$8.0{\cdot}10^6$	$4.3{\cdot}10^5$	$7.4{\cdot}10^5$	$1.6{\cdot}10^6$	$4.5{\cdot}10^6$	$7.6{\cdot}10^6$
10^{-4}	$5.2{\cdot}10^5$	$9.4{\cdot}10^5$	$1.8{\cdot}10^6$	$3.5{\cdot}10^6$	$7.3{\cdot}10^6$	$3.6{\cdot}10^5$	$6.5{\cdot}10^5$	$1.4{\cdot}10^6$	$4.1{\cdot}10^6$	$6.9{\cdot}10^6$
10^{-5}	$5.7{\cdot}10^5$	$9.8{\cdot}10^5$	$1.8{\cdot}10^6$	$3.5{\cdot}10^6$	$7.2{\cdot}10^6$	$2.8{\cdot}10^5$	$5.2{\cdot}10^5$	$1.1{\cdot}10^6$	$3.4{\cdot}10^6$	$6.3{\cdot}10^6$
10^{-6}	$1.7{\cdot}10^6$	$1.9{\cdot}10^6$	$2.5{\cdot}10^6$	$4.1{\cdot}10^6$	$7.8{\cdot}10^6$	$2.7{\cdot}10^5$	$4.8{\cdot}10^5$	$8.7{\cdot}10^5$	$2.5{\cdot}10^6$	$4.7{\cdot}10^6$
10^{-7}	$1.6{\cdot}10^7$	$1.6{\cdot}10^7$	$1.5{\cdot}10^7$	$1.5{\cdot}10^7$	$1.7{\cdot}10^7$	$2.7{\cdot}10^5$	$4.8{\cdot}10^5$	$8.5{\cdot}10^5$	$2.3{\cdot}10^6$	$4.1{\cdot}10^6$
10^{-8}	$1.6{\cdot}10^8$	$1.6{\cdot}10^8$	$1.5{\cdot}10^8$	$1.4{\cdot}10^8$	$1.5{\cdot}10^8$	$2.7{\cdot}10^5$	$4.7{\cdot}10^5$	$8.5{\cdot}10^5$	$2.2{\cdot}10^6$	$4.1{\cdot}10^6$
10^{-9}	$1.6{\cdot}10^9$	$1.6{\cdot}10^9$	$1.5{\cdot}10^9$	$1.4{\cdot}10^9$	$1.4{\cdot}10^9$	$2.7{\cdot}10^5$	$4.7{\cdot}10^5$	$8.5{\cdot}10^5$	$2.2{\cdot}10^6$	$4.0{\cdot}10^6$
10^{-10}	$1.6{\cdot}10^{10}$	$1.6{\cdot}10^{10}$	$1.5{\cdot}10^{10}$	$1.4{\cdot}10^{10}$	$1.4{\cdot}10^{10}$	$2.7{\cdot}10^5$	$4.7{\cdot}10^5$	$8.5{\cdot}10^5$	$2.2{\cdot}10^6$	$4.0{\cdot}10^6$

Table 1: Condition number of the matrix A^* (left) and condition number of the matrix preconditioned with collective Block-Jacobi (right), as functions of the mesh size h and the time step Δt.

Table 2 and Table 3 compare the convergence behavior of the standard CG, Deflated CG, and CG preconditioned with standard and collective Block-Jacobi for different time steps Δt and mesh sizes h. For each method, the tables report the

number of iterations required to reduce the (preconditioned) relative residual below 10^{-8}, considering only the solution of the linear system (4) 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 standard CG iteration counts grow rapidly for small Δt and refined meshes, while the Deflated CG method yields dramatically fewer iterations, confirming the effectiveness of the deflation strategy in improving convergence.

Δt	h_1	h_2	h_3	h_4	h_1	h_2	h_3	h_4
10^{-2}	1702	2442	3439	4727	1170	1627	2256	3188
10^{-3}	1348	1949	2542	3559	492	660	912	1286
10^{-4}	1291	1816	2329	3264	221	285	390	525
10^{-5}	1344	1848	2339	3236	106	133	181	220
10^{-6}	2011	2322	2640	3421	62	65	85	100
10^{-7}	5020	5228	5367	5356	61	56	62	57
10^{-8}	9020	10993	13665	14888	60	56	62	58

Table 2: Iteration counts for Conjugate Gradient (left) and Deflated CG (right) as functions of h and Δt, with tol = 10^{-8}.

In Table 3 we observe that CG preconditioned with standard Block-Jacobi considerably reduces iteration counts compared to the unpreconditioned CG, but the number of iterations still depends on both Δt and h. In contrast, CG preconditioned with collective Block-Jacobi exhibits nearly uniform iteration counts, independent of Δt.

Δt	h_1	h_2	h_3	h_4	h_1	h_2	h_3	h_4
10^{-2}	848	1152	1653	2207	630	871	1242	1680
10^{-3}	665	917	1272	1624	498	673	959	1205
10^{-4}	637	860	1081	1494	465	635	836	1108
10^{-5}	784	987	1149	1521	448	615	768	1075
10^{-6}	1443	1605	1799	2058	441	613	762	1056
10^{-7}	1960	2670	3398	4071	443	617	772	1067
10^{-8}	1745	2466	3524	4835	444	616	782	1074

Table 3: Iteration counts for CG preconditioned with standard Block-Jacobi (left) and CG preconditioned with collective Block-Jacobi (right) as functions of h and Δt, with tol = 10^{-8}.

Both Deflated CG and CG preconditioned with collective Block-Jacobi are robust with respect to Δt, although Deflated CG requires inverting a Laplacian-type operator during setup. To achieve robustness with respect to the mesh size h as well, the natural next step is to combine and adapt these strategies within a multigrid framework.

References

1. P. F. Antonietti, M. Botti, A. Cancrini, I. Mazzieri, *A polytopal discontinuous Galerkin method for the pseudo-stress formulation of the unsteady Stokes problem*, Computer Methods in Applied Mechanics and Engineering, 447 (2025).
2. M. R. Hestenes, E. Stiefel, *Methods of conjugate gradients for solving linear systems*, Journal of research of the National Bureau of Standards, 49 (1952).
3. R. A. Nicolaides, *Deflation of conjugate gradients with applications to boundary value problems*, SIAM Journal on Numerical Analysis, 1987.
4. K. Kahl, H. Rittich, *The deflated conjugate gradient method: Convergence, perturbation and accuracy*, Linear Algebra and its Applications, 515 (2017).
5. G. Ciaramella, M. J. Gander, *Iterative methods and preconditioners for systems of linear equations*, SIAM, Fundamentals of Algorithms, 2022.
6. A. Cangiani, Z. Dong, E. H. Georgoulis, P. Houston, *hp-version discontinuous Galerkin methods on polytopic meshes*, SpringerBriefs in Mathematics, Springer Int. Publishing, 2017.
7. D. G. Feingold, R. S. Varga, *Block diagonally dominant matrices and generalizations of the Gerschgorin circle theorem*, Pacific Journal of Mathematics, 12 (1962).
8. H. S. Price, *A practical application of block diagonally dominant matrices*, Mathematics of Computation, 19 (1965).
9. P. F. Antonietti, S. Bonetti, M. Botti, M. Corti, I. Fumagalli, I. Mazzieri, *lymph: discontinuous poLYtopal methods for Multi-PHysics differential problems*, arXiv:2401.13376, 2024.

MOX Technical Reports, last issues

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

71/2025 Paolo, F.; Tonini, A.; Valenti, G.; Hoxha, S.; Bassareo, P.P.; Dede', L.; Quarteroni, A.; Dimopoulos, K.
Analytical interpretation of hemodynamic data in patients with intracardiac shunts: role of mathematical modeling

70/2025 Greco, M.; Milan, G.; Ieva, F.; Secchi, P.
The Social Growth Index: Measuring Socioeconomic Resilience at the Municipal Level in Italy

Greco, M.; Milan, G.; Ieva, F.; Secchi, P.
The Social Growth Index: Measuring Socioeconomic Resilience at the Municipal Level in Italy

69/2025 Marino, F.; Guagliardi, O.; Di Stazio, F.; Mazza, E.; Paganoni, A.M.; Tanelli, M.
Measuring Academic Stress and Well-Being in Higher Education: A Psychometric Study

68/2025 Leimer Saglio, C. B.; Corti, M.; Pagani, S.; Antonietti, P. F.
A novel mathematical and computational framework of amyloid-beta triggered seizure dynamics in Alzheimer's disease

Leimer Saglio, C. B.; Corti, M.; Pagani, S.; Antonietti, P. F.
A novel mathematical and computational framework of amyloid-beta triggered seizure dynamics in Alzheimer's disease

67/2025 Antonietti, P.F.; Corti, M.; Gómez S.; Perugia, I.
Structure-preserving local discontinuous Galerkin discretization of conformational conversion systems

66/2025 Speroni, G.; Mondini, N.; Ferro, N.; Perotto, S.
A topology optimization framework for scaffold design in soilless cultivation

65/2025 Pottier, A.; Gelardi, F.; Larcher, A.; Capitanio, U.; Rainone, P.; Moresco, R.M.; Tenace, N.; Colecchia, M.; Grassi, S.; Ponzoni, M.; Chiti, A.; Cavinato, L.
MOSAIK: A computational framework for theranostic digital twin in renal cell carcinoma

64/2025 Celora, S.; Tonini, A.; Regazzoni, F.; Dede', L. Parati, G.; Quarteroni, A.
Cardiocirculatory Computational Models for the Study of Hypertension