INSTITUTE of MATHEMATICS

ACADEMY of SCIENCES of the CZECH REPUBLIC

BDDC for mixed-hybrid formulation of flow in porous media with combined mesh dimensions

Jakub Šístek

Jan Březina

Bedřich Sousedík

Preprint No. 20-2015

PRAHA 2015

BDDC for Mixed-Hybrid Formulation of Flow in Porous Media with Combined Mesh Dimensions

Jakub Šístek[1], Jan Březina[2] and Bedřich Sousedík[3]

[1] *Institute of Mathematics, Academy of Sciences of the Czech Republic, Žitná 25, 115 67 Prague 1, Czech Republic*
[2] *Institute for Nanomaterials, Advanced Technology and Innovation, Technical University of Liberec,*
Bendlova 1407/7, 461 17 Liberec 1, Czech Republic
[3] *Department of Mathematics and Statistics, University of Maryland, Baltimore County,*
1000 Hilltop Circle, Baltimore, MD 21250, USA

SUMMARY

We extend the Balancing Domain Decomposition by Constraints (BDDC) method to flows in porous media discretised by mixed-hybrid finite elements with combined mesh dimensions. Such discretisations appear when major geological fractures are modelled by 1D or 2D elements inside three-dimensional domains. In this set-up, the global problem as well as the substructure problems have a symmetric saddle-point structure, containing a 'penalty' block due to the combination of meshes. We show that the problem can be reduced by means of iterative substructuring to an interface problem, which is symmetric and positive definite. The interface problem can thus be solved by conjugate gradients with the BDDC method as a preconditioner. A parallel implementation of this algorithm is incorporated into an existing software package for subsurface flow simulations. We study the performance of the iterative solver on several academic and real-world problems. Numerical experiments illustrate its efficiency and scalability. Preprint version.

Submitted 2013

KEY WORDS: Iterative substructuring, BDDC, saddle-point problems, mixed-hybrid methods, fractured porous media, subsurface flow

1. INTRODUCTION

A detailed description of flow in porous media is essential for building mathematical models with applications in, for example, water management, oil and gas recovery, carbon dioxide (CO_2) sequestration or nuclear waste disposal. In order to set up a reliable numerical model, one needs to have a good knowledge of the problem geometry and input parameters. For example, the flow of water in granite rock, which is a suitable site for nuclear waste disposal, is driven by the complex system of vugs, cavities and fractures with various topology and sizes. These alter the effective permeability, and therefore should be accurately accounted for in the numerical model. There are two main approaches: either the fractures are considered as free-flow regions, or the fractures contain debris and are also modelled as porous media with specific permeabilities. In the first case, a unified approach to modelling free-flow and porous media regions can be provided by the so called

[*] Correspondence to: J. Šístek, Institute of Mathematics, Academy of Sciences of the Czech Republic, Žitná 25, 115 67 Prague 1, Czech Republic. E-mail: sistek@math.cas.cz

Contract/grant sponsor: The research was supported by Czech Science Foundation under project GA ČR 14-02067S, by Academy of Sciences of the Czech Republic through RVO:67985840, and by Ministry of Education, Youth and Sports under project CZ.1.05/2.1.00/01.0005. J. Šístek acknowledges the computing time on *HECToR* provided through the PRACE-2IP project (FP7 RI-283493).

Stokes-Brinkman equation, which reduces to either the Stokes or Darcy model in certain parameter limits, e.g., within the Multiscale Mixed Finite-Element (MsMFE) framework [1]. In this paper, we consider the latter case, and apply Darcy's law to the flow in the reservoir and in the fractures as well; see [2] for a related approach. In either case, the preferential flow in large geological dislocations and their intersections should be considered as two- and one-dimensional flows, respectively. Due to the quite complex structure of the domains, the discretisation is performed using finite element methods (FEM). The resulting meshes are therefore unstructured, and they combine different spatial dimensions (line elements in 1D, triangles in 2D, and tetrahedrons in 3D). The systems of linear equations obtained from the FEM discretisation are often very large, so that using direct methods is prohibitive and iterative solvers are warranted. The systems are typically also bad-conditioned due to the mixing of spatial dimensions, large jumps in permeability coefficients and presence of elements of considerably different sizes, and so they are challenging for iterative solvers as well.

The matrices have a saddle-point structure

$$\left[\begin{array}{cc} A & \overline{B}^T \\ \overline{B} & -\overline{C} \end{array} \right], \tag{1}$$

where A is symmetric positive definite on the kernel of $\overline{B}$, $\overline{C}$ is symmetric positive semi-definite, and it is positive definite on the kernel of $\overline{B}^T$. The 'penalty' block $\overline{C} \neq 0$ arises from connecting meshes of different spatial dimensions. The iterative solution of systems with this structure is a frequently studied topic; see, for example, [3, 4, 5, 6], the monographs [7, 8] or [9, Chapter 9] and the references therein. However, efficient methodologies for solving saddle-point problems are typically problem dependent.

In this paper, we develop a robust and scalable solver for linear systems with the saddle-point structure as in (1) with the block $\overline{C}$ either zero or nonzero. The solver is tailored to the mixed-hybrid formulation of flow in porous media using the lowest order Raviart-Thomas (RT_0) finite elements with combined mesh dimensions (1D, 2D and 3D). In particular, we adapt the Balancing Domain Decomposition by Constraints (BDDC) method to this type of problems.

The BDDC method is currently one of the most popular methods of iterative substructuring. It has been proposed independently in [10, 11, 12]; see [13, 14] for the proof of equivalence. Even though BDDC has been originally formulated for elliptic problems, it has been successfully extended, for example, beyond elliptic cases [15, 16] and to multiple levels [17, 18]. An optimal set-up has been studied in [19, 20, 21, 22]. A closely related BDDC preconditioner for vector field problems discretised with Raviart-Thomas finite elements has been studied in [23].

We are interested in applications of the BDDC method to saddle-point problems. If $\overline{C} = 0$ in (1), one possible approach is to use an algebraic trick and constrain the iterative solution of the indefinite problem into a *balanced* subspace, which is sometimes also called *benign*, where the operator is positive definite; see [15] for the Stokes problem, and [5, 24, 25] for flow in porous media. However, due to the mixed-hybrid formulation and possible coupling of meshes with different spatial dimensions, $\overline{C} \neq 0$ in general, and we will favour an alternative, *dual* approach here.

Our methodology is as follows. The mixed-hybrid formulation [26, 27] is used in order to modify the saddle-point problem to one which is symmetric and positive definite by means of iterative substructuring. In particular, we introduce a symmetric positive definite Schur complement with respect to interface Lagrange multipliers, corresponding to a part of block $\overline{C}$. The reduced system is solved by the preconditioned conjugate gradient (PCG) method, and the BDDC method is used as a preconditioner. From this perspective, our work can be viewed as a further extension of [6]. Our main effort here is in accommodating the BDDC solver to flows in porous media with combined mesh dimensions. In addition, the presentation of the BDDC algorithm is driven more by an efficient implementation, while it is more oriented towards underlying theory in [6]. We take advantage of the special structure of the blocks in matrix (1) studied in detail in [26, 28, 29]. In particular, the nonzero structure of block $\overline{C}$ resulting from a combination of meshes with different spatial dimensions is considered in [30]. We describe our parallel implementation of the method and study its performance on several benchmark and real-world problems. Another original contribution of this paper is proposing a new scaling operator in the BDDC method suitable for the studied problems. We note that if there is no coupling of meshes with different spatial dimensions present in the discretisation, the block $\overline{C} = 0$ in (1) and our method is almost identical to the one introduced in [6].

The paper is organised as follows. In Section 2, we introduce the model problem. In Section 3, we describe the modelling of fractured porous media and combining meshes of different dimensions. In Section 4, we introduce the substructuring components and derive the interface problem. In Section 5, we formulate the BDDC preconditioner. In addition, the selection of interface weights for BDDC is studied in detail in Section 6. In Section 7, we describe our parallel implementation, and in Section 8 we report the numerical results and parallel performance for benchmark and engineering problems. Finally, Section 9 provides a summary of our work.

Our notation does not, for simplicity, distinguish between finite element functions and corresponding algebraic vectors of degrees of freedom, and between linear operators and matrices within a specific basis—the meaning should be clear from the context. The transpose of a matrix is denoted by superscript ^{T} and the energy norm of a vector x is denoted by $\|x\|_M = \sqrt{x^T M x}$, where M is a symmetric positive definite matrix.

2. MODEL PROBLEM

Let Ω be an open bounded polyhedral domain in $\mathbb{R}^3$. We are interested in the solution of the following problem, combining Darcy's law and the equation of continuity written as

$$\mathbb{k}^{-1}\mathbf{u} + \nabla p = -\nabla z \quad \text{in } \Omega, \tag{2}$$

$$\nabla \cdot \mathbf{u} = f \quad \text{in } \Omega, \tag{3}$$

$$p = p_N \quad \text{on } \partial\Omega_N, \tag{4}$$

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega_E, \tag{5}$$

subject to boundary conditions on $\partial\Omega = \overline{\partial\Omega}_N \cup \overline{\partial\Omega}_E$, where $\partial\Omega_N$ stands for the part of the boundary with *natural* (Dirichlet) boundary condition, and $\partial\Omega_E$ for the part with *essential* (Neumann) boundary condition. In applications, the variable $\mathbf{u}$ describes the velocity of the fluid and p the pressure (head) in an aquifer Ω, $\mathbb{k}$ is a symmetric positive definite tensor of the hydraulic conductivity, $-\nabla z = (0, 0, -1)^T$ is the gravity term, and $\mathbf{n}$ is the outer unit normal vector of $\partial\Omega$. The term ∇z is present due to the fact, that $\mathbf{u}$ satisfies $\mathbf{u} = -\mathbb{k}\nabla p_h$, where $p_h = p + z$ is the piezometric head. For a thorough discussion of application background we refer, e.g., to monographs [31, 32].

Let $\mathcal{T}$ be the triangulation of domain Ω consisting of N_E simplicial elements with characteristic size h. We introduce a space

$$\mathbf{H}(\Omega; \mathrm{div}) = \left\{ \mathbf{v} : \mathbf{v} \in L^2(\Omega);\ \nabla \cdot \mathbf{v} \in L^2(\Omega) \text{ and } \mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega_E \right\}, \tag{6}$$

equipped with the standard norm. Let $\mathbf{V} \subset \mathbf{H}(\Omega, \mathrm{div})$ be the space consisting of the lowest order Raviart-Thomas (RT_0) functions and let $Q \subset L^2(\Omega)$ be the space consisting of piecewise constant functions on the elements of the triangulation $\mathcal{T}$. We refer, e.g., to monograph [33] for a detailed description of the mixed finite elements and the corresponding spaces.

In the *mixed finite element approximation* of problem (2)–(5) we look for a pair $\{\mathbf{u}, p\} \in \mathbf{V} \times Q$ that satisfies

$$\int_\Omega \mathbb{k}^{-1}\mathbf{u} \cdot \mathbf{v}\, dx - \int_\Omega p \nabla \cdot \mathbf{v}\, dx = -\int_{\partial\Omega_N} p_N \mathbf{v} \cdot \mathbf{n}\, ds - \int_\Omega v_z\, dx, \quad \forall \mathbf{v} \in \mathbf{V}, \tag{7}$$

$$-\int_\Omega q \nabla \cdot \mathbf{u}\, dx = -\int_\Omega f q\, dx, \quad \forall q \in Q. \tag{8}$$

In the discrete formulation, we need p_N and f only sufficiently regular so that the integrals in the weak formulation (7)–(8) make sense, namely $p_N \in L^2(\partial\Omega_N)$, $f \in L^2(\Omega)$.

Next, we recall the mixed-hybrid formulation. It was originally motivated by an effort to modify the saddle-point problem (7)–(8) to one which leads to symmetric positive definite matrices.

Nevertheless, this formulation is also suitable for a combination of meshes with different spatial dimensions, which will be described in detail in the next section.

Let $\mathcal{F}$ denote the set of inter-element *faces* of the triangulation $\mathcal{T}$. We now introduce several additional spaces. First, let us define the space $\mathbf{V}^{-1}$ by relaxing the condition of continuity of the normal components in the space $\mathbf{V}$ on inter-element boundaries $\mathcal{F}$. More precisely, we define local spaces $\mathbf{V}^i$ for each element $T^i \in \mathcal{T}$, $i = 1, \dots, N_E$, by

$$\mathbf{V}^i = \left\{ \mathbf{v} \in \mathbf{H}(T^i; \mathrm{div}) : \mathbf{v} \in RT_0(T^i) \right\}, \tag{9}$$

and put $\mathbf{V}^{-1} = \mathbf{V}^1 \times \cdots \times \mathbf{V}^{N_E}$. Next, we define the space of Lagrange multipliers Λ consisting of functions that take constant values on individual inter-element faces in $\mathcal{F}$,

$$\Lambda = \left\{ \lambda \in L^2(\mathcal{F}) : \lambda = \mathbf{v} \cdot \mathbf{n}|_{\mathcal{F}},\ \mathbf{v} \in \mathbf{V} \right\}. \tag{10}$$

In particular, $\lambda = 0$ on $\partial\Omega$ for any $\lambda \in \Lambda$.

In the *mixed-hybrid finite element approximation* of problem (2)–(5), we look for a triple $\{\mathbf{u}, p, \lambda\} \in \mathbf{V}^{-1} \times Q \times \Lambda$ that satisfies

$$\sum_{i=1}^{N_E} \left[\int_{T^i} \mathbb{k}_i^{-1} \mathbf{u} \cdot \mathbf{v}\, dx - \int_{T^i} p \nabla \cdot \mathbf{v}\, dx + \int_{\partial T^i \setminus \partial\Omega} \lambda (\mathbf{v} \cdot \mathbf{n})|_{\partial T_i}\, ds \right] \tag{11}$$

$$= - \int_{\partial\Omega_N} p_N \mathbf{v} \cdot \mathbf{n}\, ds - \sum_{i=1}^{N_E} \int_{T^i} v_z\, dx, \quad \forall \mathbf{v} \in \mathbf{V}^{-1},$$

$$- \sum_{i=1}^{N_E} \left[\int_{T^i} q \nabla \cdot \mathbf{u}\, dx \right] = - \int_{\Omega} f q\, dx, \quad \forall q \in Q, \tag{12}$$

$$\sum_{i=1}^{N_E} \left[\int_{\partial T^i \setminus \partial\Omega} \mu (\mathbf{u} \cdot \mathbf{n})|_{\partial T_i}\, ds \right] = 0, \quad \forall \mu \in \Lambda. \tag{13}$$

Equation (13) imposes a continuity condition on the normal component of the velocity (also called *normal flux*) $\mathbf{u} \cdot \mathbf{n}$ across $\mathcal{F}$ which guarantees that $\mathbf{u} \in \mathbf{V}$. This condition also implies the equivalence of the two formulations (7)–(8) and (11)–(13). We note that the Lagrange multipliers λ can be interpreted as the approximation of the trace of p on $\mathcal{F}$, see [34] for details.

Let us now write the matrix formulation corresponding to (11)–(13) as

$$\begin{bmatrix} A & B^T & B_{\mathcal{F}}^T \\ B & 0 & 0 \\ B_{\mathcal{F}} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \\ \lambda \end{bmatrix} = \begin{bmatrix} g \\ f \\ 0 \end{bmatrix}. \tag{14}$$

It is important to note that A is block diagonal with N_E blocks, corresponding to elements T^i, $i = 1, \dots, N_E$, and each of the blocks is symmetric positive definite, cf. the first term in (11). It was shown in [28] that the system of equations (14) can be reduced (twice) to the Schur complement corresponding to the Lagrange multipliers λ and solved efficiently by a direct or iterative solver. Here, we will look for an efficient solution of a slightly modified, and in general also block dense, system which is introduced in the next section.

3. MODELLING OF FRACTURES

In this section, we recall the main ideas of the discrete model of the flow in fractured porous media that is based on connection of meshes of different dimensions as described in [30]. Let us denote the full domain by $\Omega_3 = \Omega$. Next, consider lower-dimensional domains $\Omega_{d-1} \subset \Omega_d$, $d = 2, 3$, such that Ω_2 consists of polygons and Ω_1 consists of line segments. We will also assume that $\partial\Omega_1 \subset \partial\Omega_2 \subset \partial\Omega_3$. The first condition requires that a domain of a lower dimension cannot poke out of the domain of higher dimension, while the second condition prevents domains of lower dimension from having boundaries in the interior of domains of higher dimension. We impose these conditions to avoid technical difficulties in the analysis. However, numerical evidence suggests that these conditions are not necessary, and in fact, they are not satisfied for the real-world problems presented in Section 8.2.

For every dimension $d = 1, 2, 3$, we introduce a triangulation $\mathcal{T}_d$ of the domain Ω_d that consists of finite elements T_d^i, $i = 1, \ldots, N_E^d$ and satisfies the compatibility conditions

$$T_{d-1}^i \subset \mathcal{F}_d, \quad \text{where } \mathcal{F}_d = \bigcup_k \partial T_d^k \setminus \partial\Omega_d, \tag{15}$$

$$T_{d-1}^i \cap \partial T_d^j \text{ is either } T_{d-1}^i \text{ or } \emptyset, \tag{16}$$

for every $i \in \{1, \ldots, N_E^{d-1}\}$, $j \in \{1, \ldots, N_E^d\}$, and $d = 2, 3$. This means that elements of a lower dimension match faces of elements of the higher dimension.

We consider equations (3)–(5) on the domains Ω_d, $d = 1, 2, 3$, completed by a slight modification of the Darcy's law (2):

$$\mathbb{k}_d^{-1} \frac{\mathbf{u}_d}{\delta_d} + \nabla p_d = -\nabla z, \tag{17}$$

where $\mathbf{u}_d$ stands for the velocity integrated over the cross-section for $d = 1, 2$, i.e. the units of $\mathbf{u}_3$, $\mathbf{u}_2$, and $\mathbf{u}_1$ are ms^{-1}, $\mathrm{m}^2\mathrm{s}^{-1}$, and $\mathrm{m}^3\mathrm{s}^{-1}$, respectively. In addition, $\delta_3 = 1$, δ_2 is the thickness of a fracture, and δ_1 is the cross-section of a 1D preferential channel. The effective fluid source f_2 on Ω_2 is given as

$$f_2 = \delta_2 \tilde{f}_2 + \mathbf{u}_3^+ \cdot \mathbf{n}^+ + \mathbf{u}_3^- \cdot \mathbf{n}^-, \tag{18}$$

where $\tilde{f}_2$ is the density of external fluid sources, and the normal fluxes from the two faces of the 3D continuum surrounding the fracture are given through the Robin (also called Newton) boundary conditions

$$\mathbf{u}_3^+ \cdot \mathbf{n}^+ = \sigma_3^+ (p_3^+ - p_2), \tag{19}$$

$$\mathbf{u}_3^- \cdot \mathbf{n}^- = \sigma_3^- (p_3^- - p_2). \tag{20}$$

In the last formula, $\sigma_3^{+/-} > 0$ are the transition coefficients (cf. [2] for possible choices) and p_3^+, p_3^- are the traces of pressure p_3 on the two sides of the fracture. The effective fluid source f_1 on Ω_1 is similar,

$$f_1 = \delta_1 \tilde{f}_1 + \sum_k \mathbf{u}_2^k \cdot \mathbf{n}^k, \tag{21}$$

where $\tilde{f}_1$ is the density of external fluid sources. In the 3D ambient space, the 1D channel can be connected to k faces of 2D fractures. Thus

$$\mathbf{u}_2^k \cdot \mathbf{n}^k = \sigma_2^k (p_2^k - p_1) \tag{22}$$

is the normal flux from the connected fracture k, $\sigma_2^k > 0$ is the transition coefficient, and p_2^k is the trace of pressure p_2 on the face of fracture k.

In the following, we describe the discrete mixed-hybrid formulation of the problem. The formulation and discussion of the continuous problem can be found in [30]. Let us consider spaces

$$\mathbf{V}^{-1} = \mathbf{V}_1^{-1} \times \mathbf{V}_2^{-1} \times \mathbf{V}_3^{-1},\ \mathbf{V}_d^{-1} = \prod_{i=1}^{N_E^d} \mathbf{V}^i(T_d^i), \quad Q = Q_1 \times Q_2 \times Q_3,\ Q_d = L^2(\Omega_d). \tag{23}$$

For the definition of the space Λ, we cannot follow (10) directly, since e.g. on Ω_2, we need to distinguish values of λ_3 on two sides of a fracture. Thus, we introduce a separate value for every non-boundary side of every element:

$$\Lambda(T_d^i) = \left\{ \lambda \in L^2(\partial T_d^i \setminus \partial\Omega_d) : \lambda = \mathbf{v} \cdot \mathbf{n}|_{\partial T_d^i}, \mathbf{v} \in \mathbf{V}_d \right\}, \tag{24}$$

where $\mathbf{V}_d$ is defined in the same way as the space $\mathbf{V}$ but on the domain Ω_d. Further, we identify values on faces/points that are not aligned to the fractures/channels:

$$\Lambda_d = \Big\{ \lambda \in \prod_{i=1}^{N_E^d} \Lambda(T_d^i);\ \lambda|_{\partial T_d^i} = \lambda|_{\partial T_d^j} \quad \text{on face } F = \partial T_d^i \cap \partial T_d^j \quad \text{if } F \cap \Omega_{d-1} = \emptyset \Big\}. \tag{25}$$

Finally, we redefine $\Lambda = \Lambda_1 \times \Lambda_2 \times \Lambda_3$. In the *mixed-hybrid finite element approximation* of the flow in fractured porous media we seek a triple $\{\mathbf{u}, p, \lambda\} \in \mathbf{V}^{-1} \times Q \times \Lambda$ that satisfies

$$a(\mathbf{u}, \mathbf{v}) + b(p, \mathbf{v}) + b_{\mathcal{F}}(\lambda, \mathbf{v}) = \langle g, \mathbf{v} \rangle, \qquad \forall \mathbf{v} \in \mathbf{V}^{-1}, \tag{26}$$

$$b(\mathbf{u}, q) - c(p, q) - c_{\mathcal{F}}(q, \lambda) = \langle f, q \rangle, \qquad \forall q \in Q, \tag{27}$$

$$b_{\mathcal{F}}(\mathbf{u}, \mu) - c_{\mathcal{F}}(p, \mu) - \widetilde{c}(\lambda, \mu) = 0, \qquad \forall \mu \in \Lambda, \tag{28}$$

with

$$a(\mathbf{u}, \mathbf{v}) = \sum_{d=1}^{3} \sum_{i=1}^{N_E^d} \left[\int_{T_d^i} \frac{1}{\delta_d} \mathbb{k}_d^{-1} \mathbf{u}_d \cdot \mathbf{v}_d \, dx \right], \tag{29}$$

$$b(\mathbf{u}, q) = -\sum_{d=1}^{3} \sum_{i=1}^{N_E^d} \left[\int_{T_d^i} q_d \, (\nabla \cdot \mathbf{u}_d) \, dx \right], \tag{30}$$

$$b_{\mathcal{F}}(\mathbf{u}, \lambda) = \sum_{d=1}^{3} \sum_{i=1}^{N_E^d} \left[\int_{\partial T_d^i \setminus \partial \Omega_d} \lambda|_{\partial T_d^i} \, (\mathbf{u}_d \cdot \mathbf{n}) \, ds \right], \tag{31}$$

$$c(p, q) = \sum_{d=2}^{3} \sum_{i=1}^{N_E^d} \left[\int_{\partial T_d^i \cap \Omega_{d-1}} \sigma_d \, p_{d-1} \, q_{d-1} \, ds \right], \tag{32}$$

$$c_{\mathcal{F}}(p, \mu) = -\sum_{d=2}^{3} \sum_{i=1}^{N_E^d} \left[\int_{\partial T_d^i \cap \Omega_{d-1}} \sigma_d \, p_{d-1} \, \mu_d \, ds \right], \tag{33}$$

$$\widetilde{c}(\lambda, \mu) = \sum_{d=2}^{3} \sum_{i=1}^{N_E^d} \left[\int_{\partial T_d^i \cap \Omega_{d-1}} \sigma_d \, \lambda_d \, \mu_d \, ds \right], \tag{34}$$

$$\langle g, \mathbf{v} \rangle = -\sum_{d=1}^{3} \sum_{i=1}^{N_E^d} \int_{\partial T_d^i \cap \partial \Omega_N} p_N \, (\mathbf{v} \cdot \mathbf{n}) \, ds - \sum_{d=1}^{3} \sum_{i=1}^{N_E^d} \int_{T_d^i} v_z \, dx, \tag{35}$$

$$\langle f, q \rangle = -\sum_{d=1}^{3} \int_{\Omega} \delta_d \, \tilde{f}_d \, q_d \, dx. \tag{36}$$

The system (26)–(28) now leads to the matrix form

$$\begin{bmatrix} A & B^T & B_{\mathcal{F}}^T \\ B & -C & -C_{\mathcal{F}}^T \\ B_{\mathcal{F}} & -C_{\mathcal{F}} & -\widetilde{C} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \\ \lambda \end{bmatrix} = \begin{bmatrix} g \\ f \\ 0 \end{bmatrix}. \tag{37}$$

We note that (37) is related to (26)–(28) in the same way as (14) is related to (11)–(13). The main difference in the structure of the matrices between (37) and (14) is the additional block $\overline{C} = \begin{bmatrix} C & C_{\mathcal{F}}^T \\ C_{\mathcal{F}} & \widetilde{C} \end{bmatrix}$ related to the normal fluxes between dimensions and arising from (19)–(20) and (22) via (32)–(34). In particular, the modified right-hand side of the continuity equation for two-dimensional and one-dimensional elements, f_2 and f_1, incorporates pressure unknowns on 2D and 1D elements, and traces of pressure on 3D and 2D elements at the fracture, which are nothing but the Lagrange multipliers on 3D and 2D elements in the mixed-hybrid method. Consequently, $p_3^{+/-} = \lambda_3^{+/-}$ in (19)–(20), and $p_2^k = \lambda_2^k$ in (22).

Assuming δ_d bounded and greater than zero, and using the fact that $\mathbb{k}_d$ corresponds to a symmetric positive definite matrix, we see from (29) that block A in (37) is symmetric positive definite. Block $\overline{C}$ is symmetric positive semi-definite since

$$c(p,p) + 2c_{\mathcal{F}}(p,\lambda) + \widetilde{c}(\lambda,\lambda) = \sum_{d=2}^{3} \sum_{i=1}^{N_E^d} \left[\int_{\partial T_d^i \cap \Omega_{d-1}} \sigma_d (p_{d-1} - \lambda_d)^2 \, ds \right]. \tag{38}$$

The following theorem is a standard result, e.g. [33, Theorem 1.2]. Here, we rewrite it in a form suitable for our setting and we verify the assumptions for the specific blocks of the matrix in (37). We will further denote $\overline{Q} = Q \times \Lambda$, $\overline{p} = (p, \lambda) \in \overline{Q}$, $\overline{q} = (q, \mu) \in \overline{Q}$, and $\overline{b}(\mathbf{u}, \overline{q}) = b(\mathbf{u}, q) + b_{\mathcal{F}}(\mathbf{u}, \mu)$.

Theorem 1
Let natural boundary conditions (4) be prescribed at a certain part of the boundary, i.e. $\partial\Omega_{N,d} \neq \emptyset$ for at least one $d \in \{1,2,3\}$. Then the discrete mixed-hybrid problem (37) has a unique solution.

Proof
Let us first investigate the structure of the matrix in (37) more closely. Let us number the unknowns within each block of (37) with respect to spatial dimension $d \in \{1,2,3\}$. The matrix then takes the form of 9×9 blocks,

$$\begin{bmatrix}
A_{11} & & & B_{11}^T & & & B_{\mathcal{F},11}^T & & \\
& A_{22} & & & B_{22}^T & & & B_{\mathcal{F},22}^T & \\
& & A_{33} & & & B_{33}^T & & & B_{\mathcal{F},33}^T \\
B_{11} & & & -C_{11} & & & & -C_{\mathcal{F},12}^T & \\
& B_{22} & & & -C_{22} & & & & -C_{\mathcal{F},23}^T \\
& & B_{33} & & & & & & \\
B_{\mathcal{F},11} & & & & & & & & \\
& B_{\mathcal{F},22} & & -C_{\mathcal{F},12} & & & & -\widetilde{C}_{22} & \\
& & B_{\mathcal{F},33} & & -C_{\mathcal{F},23} & & & & -\widetilde{C}_{33}
\end{bmatrix}. \tag{39}$$

Suppose for a moment that we solve a problem only on domain Ω_d, $d \in 1,2,3$ (i.e. $\Omega_i = \emptyset$ for $i \neq d$). If no natural boundary conditions are imposed, there is a single -1 entry on each row of B_{dd}^T and a single $+1$ entry on each row of $B_{\mathcal{F},dd}^T$. Since Ω_d is a simply connected set, the matrix $\overline{B}_{dd}^T = [\, B_{dd}^T \; B_{\mathcal{F},dd}^T \,]$ has a nontrivial nullspace of constant vectors. Enforcing natural boundary condition on a part of Ω_d changes the $+1$ value on the corresponding row of matrix $B_{\mathcal{F},dd}^T$ to 0, in which case $\overline{B}_{dd}^T$ has only a trivial nullspace, i.e. full column rank (see e.g. [33, Section IV.1] or [26, Lemma 3.2]).

The nullspace becomes more complicated for domains with fractures, in which case Ω_d typically has more simply connected components separated by fractures (cf. Fig. 1). Let us denote them Ω_k^c, $k = 1, \ldots, n_c$, regardless of the dimension. In particular, Ω_k^c, $k = 1, \ldots, n_{ci}$ will be components without natural condition boundary, i.e. $\partial\Omega_k^c \cap \partial\Omega_N = \emptyset$, while for $k = n_{ci}+1, \ldots, n_c$ we get components with prescribed natural boundary condition. We also denote $\overline{\chi}_k \in \overline{Q}$ the characteristic vector of the component Ω_k^c that takes value $+1$ for degrees of freedom associated with elements and faces of the component Ω_k^c. With such notation, the basis of the nullspace of the whole matrix

$$\overline{B}^T = \begin{bmatrix}
B_{11}^T & & & B_{\mathcal{F},11}^T & & \\
& B_{22}^T & & & B_{\mathcal{F},22}^T & \\
& & B_{33}^T & & & B_{\mathcal{F},33}^T
\end{bmatrix} \tag{40}$$

consists of characteristic vectors $\overline{\chi}_k$, $k = 1, \ldots n_{ci}$ and has dimension n_{ci}.

Next, we show that matrix $\overline{C}$ is not only symmetric positive semi-definite, as seen from (38), but also positive definite on null $\overline{B}^T$. To this end, take $\overline{p} \in$ null $\overline{B}^T$, a vector that is piecewise constant on components, having value $\overline{p}_{k,d}$ on the component Ω^c_k of dimension d for $k = 1, \dots, n_{ci}$, and value $\overline{p}_{k,d} = 0$ for other components. Then $\overline{p}^T \overline{C}\,\overline{p} = 0$ implies $\overline{p} = \overline{p}_{k,d} = 0$. Indeed, every component Ω^c_k of dimension $d = 2, 3$ has some component Ω^c_j of dimension $d - 1$ on its boundary, and therefore all $\overline{p}_{k,d}$ have the same value, cf. (38). This value is zero, since there is at least one component with natural boundary condition.

Applying the congruence transformation,

$$\begin{bmatrix} A & \overline{B}^T \\ \overline{B} & -\overline{C} \end{bmatrix} = \begin{bmatrix} I & \\ \overline{B}A^{-1} & I \end{bmatrix} \begin{bmatrix} A & \\ & -\left(\overline{B}A^{-1}\overline{B}^T + \overline{C}\right) \end{bmatrix} \begin{bmatrix} I & A^{-1}\overline{B}^T \\ & I \end{bmatrix}. \tag{41}$$

Matrix A is symmetric positive definite (SPD) from (29) and therefore $\overline{B}A^{-1}\overline{B}^T$ is SPD on range $\overline{B}$, which is the orthogonal complement of the nullspace of $\overline{B}^T$. Thus $\overline{B}A^{-1}\overline{B}^T + \overline{C}$ is SPD on whole $\overline{Q}$. From the Sylvester law of inertia, the number of positive, negative and zero eigenvalues is preserved by the congruence transformation. Since the block diagonal matrix on the right-hand side of (41) has only (strictly) positive and (strictly) negative eigenvalues, the matrix on the left-hand side also must be nonsingular, and problem (37) has a unique solution.

□

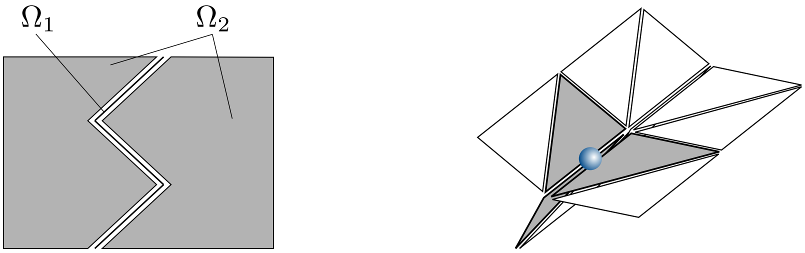

Figure 1. Example of a two-dimensional problem: even single fracture gives rise to two components of the two-dimensional mesh (left); example of three (shaded) triangles sharing a single Lagrange multiplier in 3D (right).

4. ITERATIVE SUBSTRUCTURING

For our purposes of combining meshes with different spatial dimensions, we define *substructures* as subsets of finite elements in the mesh rather than parts of a physical domain, cf. [9].

To begin, let us define the combined triangulation $\mathcal{T}_{123}$ as the union of triangulations for each spatial dimension, i.e., $\mathcal{T}_{123} = \mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3$. The triangulation $\mathcal{T}_{123}$ is subsequently divided into substructures $\Omega^i,\ i = 1, \dots, N_S$. Note that in general a substructure can contain finite elements of different dimensions. We define the interface Γ as the set of degrees of freedom shared by more than one substructure. Note that $\Gamma \subset \Lambda$ for the current setting. Thus, let us split the Lagrange multipliers λ into two subsets. First, we denote by λ_Γ the set of multipliers corresponding to the interface Γ. The remaining multipliers, corresponding to substructure interiors, will be denoted by λ_I. The interface Γ^i of substructure Ω^i is defined as a subset of Γ corresponding to $\partial\Omega^i$. Next, let Λ^i_Γ be defined as the space of Lagrange multipliers corresponding to Γ^i, $i = 1, \dots, N_S$, and define a space

$$\Lambda_\Gamma = \Lambda^1_\Gamma \times \cdots \times \Lambda^{N_S}_\Gamma. \tag{42}$$

The substructure problems are obtained by assembling contributions of finite elements in each Ω^i,

$$\begin{bmatrix} A^i & B^{iT} & B^{iT}_{\mathcal{F},I} & B^{iT}_{\mathcal{F},\Gamma} \\ B^i & -C^i & -C^{iT}_{\mathcal{F},I} & -C^{iT}_{\mathcal{F},\Gamma} \\ B^i_{\mathcal{F},I} & -C^i_{\mathcal{F},I} & -\widetilde{C}^i_{II} & -\widetilde{C}^{iT}_{\Gamma I} \\ B^i_{\mathcal{F},\Gamma} & -C^i_{\mathcal{F},\Gamma} & -\widetilde{C}^i_{\Gamma I} & -\widetilde{C}^i_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{u}^i \\ p^i \\ \lambda^i_I \\ \lambda^i_\Gamma \end{bmatrix} = \begin{bmatrix} g^i \\ f^i \\ 0 \\ 0 \end{bmatrix}, \qquad i = 1, \dots, N_S, \tag{43}$$

where the blocks A^i are block diagonal with blocks corresponding to finite element matrices, and the blocks $\overline{C}^i \neq 0$ only if the substructure Ω^i contains coupling of elements of different dimensions. We note that the global problem (37) could be obtained from (43) by further assembly at the interface.

In *the iterative substructuring* (see e.g. [9]), we first reduce the problem to substructure interfaces. In our context we can eliminate normal fluxes, pressure unknowns, and Lagrange multipliers at interiors of substructures, and we can define the substructure Schur complements $S^i : \Lambda^i_\Gamma \mapsto \Lambda^i_\Gamma$, $i = 1, \dots, N_S$, formally as

$$S^i = \widetilde{C}^i_{\Gamma\Gamma} + \begin{bmatrix} B^i_{\mathcal{F},\Gamma} & -C^i_{\mathcal{F},\Gamma} & -\widetilde{C}^i_{\Gamma I} \end{bmatrix} \begin{bmatrix} A^i & B^{iT} & B^{iT}_{\mathcal{F},I} \\ B^i & -C^i & -C^{iT}_{\mathcal{F},I} \\ B^i_{\mathcal{F},I} & -C^i_{\mathcal{F},I} & -\widetilde{C}^i_{II} \end{bmatrix}^{-1} \begin{bmatrix} B^{iT}_{\mathcal{F},\Gamma} \\ -C^{iT}_{\mathcal{F},\Gamma} \\ -\widetilde{C}^{iT}_{\Gamma I} \end{bmatrix}. \tag{44}$$

However, in an implementation of a Krylov subspace iterative method, we only need to compute the matrix-vector product $S^i \lambda^i_\Gamma$ for a given vector λ^i_Γ. Therefore, the matrix is not constructed explicitly, and the multiplication is obtained as follows.

Algorithm 2
Given $\lambda^i_\Gamma \in \Lambda^i_\Gamma$, determine $S^i \lambda^i_\Gamma \in \Lambda^i_\Gamma$ in the following two steps:

1. solve a local *Dirichlet problem*

$$\begin{bmatrix} A^i & B^{iT} & B^{iT}_{\mathcal{F},I} \\ B^i & -C^i & -C^{iT}_{\mathcal{F},I} \\ B^i_{\mathcal{F},I} & -C^i_{\mathcal{F},I} & -\widetilde{C}^i_{II} \end{bmatrix} \begin{bmatrix} \mathbf{w}^i \\ q^i \\ \mu^i_I \end{bmatrix} = - \begin{bmatrix} B^{iT}_{\mathcal{F},\Gamma} \\ -C^{iT}_{\mathcal{F},\Gamma} \\ -\widetilde{C}^{iT}_{\Gamma I} \end{bmatrix} \lambda^i_\Gamma, \tag{45}$$

2. perform two sparse matrix-vector multiplications

$$S^i \lambda^i_\Gamma \longleftarrow - \left(-\widetilde{C}^i_{\Gamma\Gamma} \lambda^i_\Gamma + \begin{bmatrix} B^i_{\mathcal{F},\Gamma} & -C^i_{\mathcal{F},\Gamma} & -\widetilde{C}^i_{\Gamma I} \end{bmatrix} \begin{bmatrix} \mathbf{w}^i \\ q^i \\ \mu^i_I \end{bmatrix} \right). \tag{46}$$

Next, let $\widehat{\Lambda}_\Gamma$ be the space of global degrees of freedom, such that the values of degrees of freedom from Λ_Γ coincide on Γ. The vectors of the local substructure degrees of freedom $\lambda^i_\Gamma \in \Lambda^i_\Gamma$ and the vector of the global degrees of freedom $\lambda_\Gamma \in \widehat{\Lambda}_\Gamma$ are related by

$$\lambda^i_\Gamma = R^i \lambda_\Gamma, \quad i = 1, \dots, N_S, \tag{47}$$

where R^i are the restriction operators. More specifically, each R^i is a 0-1 matrix such that every row contains one entry equal to one, and $R^i R^{iT} = I$. The global Schur complement $\widehat{S} : \widehat{\Lambda}_\Gamma \to \widehat{\Lambda}_\Gamma$ can be obtained as

$$\widehat{S} = \sum_{i=1}^{N_S} R^{iT} S^i R^i. \tag{48}$$

Equation (48) represents the formal assembly of substructure Schur complements into the global Schur complement. The global Schur complement system, which we would like to solve iteratively, reads

$$\widehat{S} \lambda_\Gamma = \widehat{b}, \tag{49}$$

where vector $\widehat{b} = \sum_{i=1}^{N_S} R^{iT} b^i$ is obtained from substructure reduced right-hand sides

$$b^i = \left[\begin{array}{ccc} B^i_{\mathcal{F},\Gamma} & -C^i_{\mathcal{F},\Gamma} & -\widetilde{C}^i_{\Gamma I} \end{array} \right] \left[\begin{array}{ccc} A^i & B^{iT} & B^{iT}_{\mathcal{F},I} \\ B^i & -C^i & -C^{iT}_{\mathcal{F},I} \\ B^i_{\mathcal{F},I} & -C^i_{\mathcal{F},I} & -\widetilde{C}^i_{II} \end{array} \right]^{-1} \left[\begin{array}{c} g^i \\ f^i \\ 0 \end{array} \right]. \tag{50}$$

In our implementation, we factor and store the matrices from (45). The factors are then used to compute the vectors b^i in (50), and especially in Algorithm 2, which is used in connection to formula (48) to evaluate $\lambda_\Gamma \to \widehat{S}\lambda_\Gamma$ within each iteration of a Krylov subspace iterative method. This algorithm allows a straightforward parallel implementation. After an approximate solution of (49) is found, solution in interiors of substructures, including all primal variables, is recovered from (43) using the factors from (45).

Remark 3
There are other ways to derive the interface problem (49). The authors of [6, 34] use a mixed-hybrid formulation with Lagrange multipliers introduced only at the interface Γ as their starting point. While problem (49) is equivalent to the interface problems considered in [6, 34], the substructure problems therein have a different structure from (43). In particular, there are no blocks corresponding to λ^i_I, and the matrices corresponding to A^i are no longer block diagonal element-wise. Next, the authors of [28, 30] construct explicit Schur complement with respect to the whole block of Lagrange multipliers λ and they show that, due to the special structure of A, the complement is both sparse and reasonably cheap to construct. If this was performed substructure by substructure, this could be again seen as an intermediate step in obtaining problem (49) by additional elimination of the interior Lagrange multipliers λ_I. However, this would again lead to different substructure problems based on explicit local Schur complements.

The following result allows an application of the BDDC method to problems with fractures.

Theorem 4
Let natural boundary conditions (4) be prescribed at a certain part of the boundary, i.e. $\partial\Omega_{N,d} \neq \emptyset$ for at least one $d \in \{1, 2, 3\}$. Then the matrix $\widehat{S}$ in (49) is symmetric positive definite.

Proof
Using the notation of (40)–(41), let us introduce a matrix $\mathcal{S} = \overline{B} A^{-1} \overline{B}^T + \overline{C}$. The matrix $\mathcal{S}$ is symmetric positive definite by Theorem 1 (and its proof). Applying another congruence transformation to $\mathcal{S}$ and denoting the rows corresponding to the interface Lagrange multipliers by subscript $_\Gamma$ and the interior by $_I$, we obtain

$$\mathcal{S} = \begin{bmatrix} \mathcal{S}_{II} & \mathcal{S}^T_{\Gamma I} \\ \mathcal{S}_{\Gamma I} & \mathcal{S}_{\Gamma\Gamma} \end{bmatrix} = \begin{bmatrix} I & \\ \mathcal{S}_{\Gamma I}\mathcal{S}^{-1}_{II} & I \end{bmatrix} \begin{bmatrix} \mathcal{S}_{II} & \\ & \mathcal{S}_{\Gamma\Gamma} - \mathcal{S}_{\Gamma I}\mathcal{S}^{-1}_{II}\mathcal{S}^T_{\Gamma I} \end{bmatrix} \begin{bmatrix} I & \mathcal{S}^{-1}_{II}\mathcal{S}^T_{\Gamma I} \\ & I \end{bmatrix}. \tag{51}$$

Since the matrix on the left-hand side is SPD, both diagonal blocks $\mathcal{S}_{II}$ and $\mathcal{S}_{\Gamma\Gamma} - \mathcal{S}_{\Gamma I}\mathcal{S}^{-1}_{II}\mathcal{S}^T_{\Gamma I}$ are also symmetric positive definite from the Sylvester law of inertia. It remains to note that the Schur complement $\widehat{S}$ in (49) is symmetric positive definite because $\widehat{S} = \mathcal{S}_{\Gamma\Gamma} - \mathcal{S}_{\Gamma I}\mathcal{S}^{-1}_{II}\mathcal{S}^T_{\Gamma I}$. □

Theorem 4 allows us to use the conjugate gradient method for the iterative solution of (49). In the next section, we describe the BDDC method used as a preconditioner for $\widehat{S}$.

5. THE BDDC PRECONDITIONER

In this section, we formulate the BDDC method for the solution of (49). The algorithm can be viewed as a generalisation of [6]. However, we follow the original description from [11], which better reflects our implementation.

One step of BDDC provides a two-level preconditioner for the conjugate gradient method applied to solving problem (49). It is characterised by the selection of certain *coarse degrees of freedom* based on primary degrees of freedom at interface Γ. The main coarse degrees of freedom in this paper are arithmetic averages over *faces*, defined as subsets of degrees of freedom shared by the same two substructures. In addition, *corner* coarse degrees of freedom, defined as any selected Lagrange multiplier at the interface, are used. Substructure *edges*, defined as subsets of degrees of freedom shared by several substructures, may also appear (see Remark 5).

The BDDC method introduces *constraints* which enforce continuity of functions from Λ_Γ at coarse degrees of freedom among substructures. This gives raise to the space $\widetilde{\Lambda}_\Gamma$, which is given as the subspace of Λ_Γ of functions which satisfy these continuity constraints. In particular,

$$\widehat{\Lambda}_\Gamma \subset \widetilde{\Lambda}_\Gamma \subset \Lambda_\Gamma. \tag{52}$$

Remark 5
In three spatial dimensions, several triangular elements can be connected at a single Lagrange multiplier in a star-like configuration (cf. Fig. 1). A similar statement holds for line elements considered in two- or three-dimensional space. This fact may lead to the presence of substructure *edges* and even *vertices* (defined as degenerate edges consisting of a single degree of freedom), and we may prescribe also edge averages as constraints. As mentioned above, we also select *corners* as coarse degrees of freedom. While essentially any degree of freedom at the interface Γ can be *a corner*, we select them by the face-based algorithm [21]. This algorithm considers all vertices as corners, and, in addition, it selects three geometrically well distributed degrees of freedom from the interface between two substructures sharing a face into the set of *corners*. Although considering corners is not the standard practice with RT_0 finite elements, in our experience, corners improve convergence for numerically difficult problems, as can be observed for the engineering applications presented in Section 8.

We now proceed to the formulation of operators used in the BDDC method. The choice of constraints determines the construction of matrices D^i. Each row of D^i defines one coarse degree of freedom at substructure Ω^i, e.g., a corner corresponds to a single 1 entry at a row and an arithmetic average to several 1's at a row. The *coarse basis functions* Φ^i_Γ, one per each substructure coarse degree of freedom, are computed by augmenting the matrices from (43) with D^i, and solving the augmented systems with multiple right-hand sides

$$\begin{bmatrix} A^i & B^{iT} & B^{iT}_{\mathcal{F},I} & B^{iT}_{\mathcal{F},\Gamma} & 0 \\ B^i & -C^i & -C^{iT}_{\mathcal{F},I} & -C^{iT}_{\mathcal{F},\Gamma} & 0 \\ B^i_{\mathcal{F},I} & -C^i_{\mathcal{F},I} & -\widetilde{C}^i_{II} & -\widetilde{C}^{iT}_{\Gamma I} & 0 \\ B^i_{\mathcal{F},\Gamma} & -C^i_{\mathcal{F},\Gamma} & -\widetilde{C}^i_{\Gamma I} & -\widetilde{C}^i_{\Gamma\Gamma} & D^{iT} \\ 0 & 0 & 0 & D^i & 0 \end{bmatrix} \begin{bmatrix} X^i \\ Z^i \\ \Phi^i_I \\ \Phi^i_\Gamma \\ L^i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ I \end{bmatrix}, \quad i = 1, \dots, N_S, \tag{53}$$

where I is the identity matrix, and X^i, Z^i, and Φ^i_I are auxiliary matrices not used any further. As shown in [35], the *local coarse matrix* S^i_{CC} is obtained as a side product of solving (53) as

$$S^i_{CC} = \left[X^{iT}\ Z^{iT}\ \Phi^{iT}_I\ \Phi^{iT}_\Gamma\right] \begin{bmatrix} A^i & B^{iT} & B^{iT}_{\mathcal{F},I} & B^{iT}_{\mathcal{F},\Gamma} \\ B^i & -C^i & -C^{iT}_{\mathcal{F},I} & -C^{iT}_{\mathcal{F},\Gamma} \\ B^i_{\mathcal{F},I} & -C^i_{\mathcal{F},I} & -\widetilde{C}^i_{II} & -\widetilde{C}^{iT}_{\Gamma I} \\ B^i_{\mathcal{F},\Gamma} & -C^i_{\mathcal{F},\Gamma} & -\widetilde{C}^i_{\Gamma I} & -\widetilde{C}^i_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} X^i \\ Z^i \\ \Phi^i_I \\ \Phi^i_\Gamma \end{bmatrix} = -L^i. \tag{54}$$

Let us define, similarly to (47), operators R^i_C that relate vectors of local coarse degrees of freedom μ^i_C to the vector of global coarse degrees of freedom μ_C as

$$\mu^i_C = R^i_C \mu_C. \tag{55}$$

The *global coarse matrix* S_{CC} is then obtained by assembling the local contributions as

$$S_{CC} = \sum_{i=1}^{N_S} R^{iT}_C S^i_{CC} R^i_C. \tag{56}$$

Finally, let us define the scaling operators

$$W^i : \Lambda^i_\Gamma \to \Lambda^i_\Gamma, \quad i = 1, \dots, N^S, \tag{57}$$

which are given as diagonal matrices of weights that satisfy

$$\sum_{i=1}^{N_S} R^{iT} W^i R^i = I. \tag{58}$$

More details on the selection of diagonal entries in W^i are given in Section 6.

With this selection of spaces and operators, we are ready to formulate the BDDC preconditioner.

Algorithm 6
The BDDC preconditioner $M_{BDDC} : r_\Gamma \in \widehat{\Lambda}_\Gamma \to \lambda_\Gamma \in \widehat{\Lambda}_\Gamma$ is defined in the following steps:

1. Compute the local residuals

$$r^i_\Gamma = W^i R^i r_\Gamma, \quad i = 1, \dots, N_S. \tag{59}$$

2. Compute the *substructure corrections* $\eta^i_{\Gamma\Delta}$ by solving the local *Neumann problems*

$$\begin{bmatrix} A^i & B^{iT} & B^{iT}_{\mathcal{F},I} & B^{iT}_{\mathcal{F},\Gamma} & 0 \\ B^i & -C^i & -C^{iT}_{\mathcal{F},I} & -C^{iT}_{\mathcal{F},\Gamma} & 0 \\ B^i_{\mathcal{F},I} & -C^i_{\mathcal{F},I} & -\widetilde{C}^i_{II} & -\widetilde{C}^{iT}_{\Gamma I} & 0 \\ B^i_{\mathcal{F},\Gamma} & -C^i_{\mathcal{F},\Gamma} & -\widetilde{C}^i_{\Gamma I} & -\widetilde{C}^i_{\Gamma\Gamma} & D^{iT} \\ 0 & 0 & 0 & D^i & 0 \end{bmatrix} \begin{bmatrix} x^i \\ z^i \\ \eta^i_{I\Delta} \\ \eta^i_{\Gamma\Delta} \\ l^i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ r^i_\Gamma \\ 0 \end{bmatrix}, \quad i = 1, \dots, N_S. \tag{60}$$

3. Compute the *coarse correction* η^i_C by collecting the coarse residual

$$r_C = \sum_{i=1}^{N_S} R^{iT}_C \Phi^{iT}_\Gamma r^i_\Gamma, \tag{61}$$

solving the global *coarse problem*

$$S_{CC}\, \eta_C = r_C, \tag{62}$$

and distributing the coarse correction

$$\eta^i_{\Gamma C} = \Phi^i_\Gamma R^i_C \eta_C, \quad i = 1, \dots, N_S. \tag{63}$$

4. Combine and average the corrections

$$\lambda_\Gamma = -\sum_{i=1}^{N_S} R^{iT} W^i \left(\eta^i_{\Gamma\Delta} + \eta^i_{\Gamma C} \right). \tag{64}$$

We note that the factorisations of the matrices from (53) are also used for each solution of (60).

In order to apply the existing BDDC theory for elliptic problems (e.g. [13, 36]) to the proposed preconditioner, we introduce some additional notation and make a few observations. Due to (44), the substructure corrections in (60) can be written equivalently as

$$\begin{bmatrix} -S^i & D^{iT} \\ D^i & 0 \end{bmatrix} \begin{bmatrix} \eta^i_{\Gamma\Delta} \\ l^i \end{bmatrix} = \begin{bmatrix} r^i_\Gamma \\ 0 \end{bmatrix}, \tag{65}$$

and the construction of coarse basis functions Φ^i_Γ in (53) as

$$\begin{bmatrix} -S^i & D^{iT} \\ D^i & 0 \end{bmatrix} \begin{bmatrix} \Phi^i_\Gamma \\ L^i \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}. \tag{66}$$

Next, let us formally write the operators and vectors in the block form

$$\lambda_\Gamma = \begin{bmatrix} \lambda_\Gamma^1 \\ \vdots \\ \lambda_\Gamma^{N_S} \end{bmatrix}, \; R = \begin{bmatrix} R^1 \\ \vdots \\ R^{N_S} \end{bmatrix}, \; W = \begin{bmatrix} W^1 & & \\ & \ddots & \\ & & W^{N_S} \end{bmatrix}, \; S = \begin{bmatrix} S^1 & & \\ & \ddots & \\ & & S^{N_S} \end{bmatrix}. \tag{67}$$

By grouping the steps of Algorithm 6 and using (65), the operator of the BDDC preconditioner can be formally written as

$$M_{BDDC} = R^T W \widetilde{S}^{-1} W R, \tag{68}$$

where

$$\widetilde{S}^{-1} = -\left\{ \operatorname*{diag}_{i=1,\dots,N_S} \left(\begin{bmatrix} I_{\Lambda_\Gamma^i} & 0 \end{bmatrix} \begin{bmatrix} -S^i & D^{iT} \\ D^i & 0 \end{bmatrix}^{-1} \begin{bmatrix} I_{\Lambda_\Gamma^i} \\ 0 \end{bmatrix} \right) + \left(\sum_{i=1}^{N_S} R_C^{iT} \Phi_\Gamma^{iT} \right)^T S_{CC}^{-1} \left(\sum_{i=1}^{N_S} R_C^{iT} \Phi_\Gamma^{iT} \right) \right\}. \tag{69}$$

The first term in $\widetilde{S}^{-1}$ corresponds to substructure corrections and the second term to the coarse correction (steps 2 and 3 of Algorithm 6), and $I_{\Lambda_\Gamma^i}$ is the identity matrix in Λ_Γ^i. From (68) and (69), one can readily see that M_{BDDC} is symmetric.

Assumption 7
Let us assume that

$$\operatorname{null} S^i \perp \operatorname{null} D^i. \tag{70}$$

In order to satisfy Assumption 7, we must prescribe enough coarse degrees of freedom as constraints along with Robin boundary conditions (19)–(20) and (22) at each fracture within substructure Ω^i. Since constrains in D^i are linearly independent, D^{iT} has full column rank. In particular, Assumption 7 is satisfied when arithmetic averages are used on each substructure face (and eventually edge) as constraints.

Lemma 8
The operator $\widetilde{S}^{-1}$ in preconditioner M_{BDDC} is symmetric and positive definite on the space $\widetilde{\Lambda}_\Gamma$.

Proof

The space $\widetilde{\Lambda}_\Gamma$ is decomposed into the substructure spaces and the coarse space,

$$\widetilde{\Lambda}_\Gamma = \widetilde{\Lambda}_{\Gamma\Delta} \oplus \widetilde{\Lambda}_{\Gamma C}. \tag{71}$$

To achieve this splitting, each local space Λ_Γ^i is decomposed into subspaces

$$\Lambda_\Gamma^i = \operatorname{null} D^i \oplus \operatorname{range} \Phi_\Gamma^i, \tag{72}$$

corresponding to the *substructure space* $\widetilde{\Lambda}_{\Gamma\Delta}^i$, and the *coarse space* on substructure Ω^i, $\widetilde{\Lambda}_{\Gamma C}|_{\Omega^i}$, respectively. To analyse this decomposition, let us recall that S^i is a positive semi-definite matrix and write (66) in detail as

$$S^i \Phi_\Gamma^i = D^{iT} L^i, \tag{73}$$

$$D^i \Phi_\Gamma^i = I. \tag{74}$$

From (73),

$$\operatorname{range}(S^i \Phi_\Gamma^i) \subset \operatorname{range} D^{iT} \perp \operatorname{null} D^i, \tag{75}$$

which in turn, similarly to [36, Lemma 8], gives for any $\chi_\Delta^i \in \operatorname{null} D^i$ and $\xi_C^i \in \operatorname{range} \Phi_\Gamma^i$

$$\chi_\Delta^{iT} S^i \xi_C^i = 0. \tag{76}$$

From (74), the matrix Φ_Γ^i has full column rank, and

$$\operatorname{null} D^i \cap \operatorname{range} \Phi_\Gamma^i = 0. \tag{77}$$

Finally, from Assumption 7 and (72),

$$\operatorname{null} S^i \subset \operatorname{range} \Phi_\Gamma^i. \tag{78}$$

Decomposition of the subdomain space (72) implies decomposition of a function $\zeta^i \in \Lambda_\Gamma^i$ to $\zeta^i = \zeta_\Delta^i + \zeta_C^i$, where $\zeta_\Delta^i \in \operatorname{null} D^i$, $\zeta_C^i \in \operatorname{range} \Phi_\Gamma^i$, and $\zeta_\Delta^{iT} S^i \zeta_C^i = 0$ by (76).

Let us first analyse the substructure corrections. Following [3, Section 3.3], the matrix of (65) is invertible due to Assumption 7. If we define, in addition, a matrix Q^i with orthonormal columns forming a basis of $\operatorname{null} D^i$, i.e.

$$\operatorname{range} Q^i = \operatorname{null} D^i, \qquad Q^{iT} Q^i = I, \tag{79}$$

we have

$$\begin{bmatrix} I_{\Lambda_\Gamma^i} & 0 \end{bmatrix} \begin{bmatrix} -S^i & D^{iT} \\ D^i & 0 \end{bmatrix}^{-1} \begin{bmatrix} I_{\Lambda_\Gamma^i} \\ 0 \end{bmatrix} = -Q^i \left(Q^{iT} S^i Q^i \right)^{-1} Q^{iT}. \tag{80}$$

The matrix $\left(Q^{iT} S^i Q^i \right)^{-1}$ is symmetric positive definite, and consequently, for any $\zeta^i \in \Lambda_\Gamma^i$,

$$-\zeta^{iT} Q^i \left(Q^{iT} S^i Q^i \right)^{-1} Q^{iT} \zeta^i \leq 0 \tag{81}$$

with equality if $\zeta^i = \zeta_C^i \in \operatorname{range} \Phi_\Gamma^i$.

Next, let us turn towards the coarse correction. Formula (54) for S_{CC}^i can be written equivalently as

$$S_{CC}^i = -\Phi_\Gamma^{iT} S^i \Phi_\Gamma^i. \tag{82}$$

Since the term on the right-hand side is just a (negative) Galerkin projection of the positive semi-definite matrix S^i, matrix S_{CC}^i is symmetric negative semi-definite. If at least one substructure is equipped with natural boundary conditions, the matrix S_{CC} assembled by (56) becomes symmetric negative definite, and so is S_{CC}^{-1}.

We have just verified the negative definiteness of the principal parts of $\widetilde{S}^{-1}$, and the desired positive definiteness is obtained through the change of sign in front of the braces in (69). □

In view of Lemma 8, the standard condition number bound follows from [13, Lemma 2].

Theorem 9
The condition number κ of the preconditioned operator $M_{BDDC}\widehat{S}$ satisfies

$$\kappa \leq \omega = \max_{\lambda_\Gamma \in \widetilde{\Lambda}_\Gamma} \frac{\left\| R R^T W \lambda_\Gamma \right\|_S^2}{\left\| \lambda_\Gamma \right\|_S^2}. \tag{83}$$

The norms in (83) are induced by the matrix S defined in (67) for all functions $\lambda_\Gamma \in \widetilde{\Lambda}_\Gamma$.

In addition, in the case of a single mesh dimension in either 2D or 3D, and under the assumption of substructure-wise constant hydraulic conductivities, it has been also derived in [6, Lemma 5.5 and Theorem 6.1] that the condition number bound ω satisfies

$$\omega \leq C \left(1 + \log \frac{H}{h} \right)^2, \tag{84}$$

where H is the characteristic size of geometric substructures. We note that the bound (84) implies that, for a fixed relative subdomain size H/h, the condition number is independent of the problem size.

It is worth emphasising that Theorem 9 is also valid for combined mesh dimensions. However, several simplifications are employed in [6] to obtain (84), which are not satisfied in the set-up considered in this paper. In particular (i) hydraulic conductivity coefficient here is, in general, not substructure-wise constant nor isotropic, (ii) it is not clear whether, in presence of fractures, the interpolation operator onto a conforming mesh introduced in [6] can be constructed and bounded in the H^1 norm.

6. SCALING WEIGHTS IN BDDC

Let us now discuss the choice of entries in the diagonal weight matrices W^i. These matrices play an important role in the BDDC method, both in the theory, cf. Theorem 9 or [6, 13, 37], and in the computations, cf. [19, 38]. Three possible choices are also studied numerically in Section 8.2. The basic choice is presented by the *arithmetic average* taken from values at the neighbouring substructures. In this simplest construction, the entry corresponding to Lagrange multiplier $\lambda^i_{\Gamma,j}$ is given by the inverse counting function as

$$W^i_{jj} = \frac{1}{card(\mathcal{I}_j)}, \tag{85}$$

where $card(\mathcal{I}_j)$ is the number of substructures in the set $\mathcal{I}_j$ of indices of substructures to which $\lambda^i_{\Gamma,j}$ belongs. For 2D or 3D meshes without fractures, $W^i_{jj} = 1/2$ for Raviart-Thomas elements. However, since several two-dimensional fractures can meet in our setting, smaller weights can occasionally appear at such regions.

While arithmetic average is sufficient for problems with homogeneous coefficients, it is well known that for problems with large variations in material properties along the interface, it is necessary to incorporate their values into the (weighted) average to obtain a robust method. This gives rise to the ρ-scaling, for which

$$W^i_{jj} = \frac{\rho_i}{\sum_{k=1}^{card(\mathcal{I}_j)} \rho_k}, \tag{86}$$

where ρ_k is a material characteristic for substructure Ω^k. This choice is robust with respect to jumps in coefficients across the interface, cf. [6, 9]; however, coefficients are assumed constant for each substructure. This requirement is very restrictive for practical computations with quickly varying coefficients, and we employ a generalisation which takes into account the material coefficient of the element to which the Lagrange multiplier $\lambda^i_{\Gamma,j}$ corresponds. In our case, we use $\rho_i = d/\mathrm{tr}\,(\Bbbk^{-1})$, where $d \in \{1, 2, 3\}$ is the dimension of the element T^i. This value can be seen as a representative hydraulic conductivity on the element.

Finally, we propose a modification of the popular scaling by diagonal stiffness [19]. In the usual diagonal stiffness approach, the optimal weight, which is the diagonal entry of the Schur complement, is approximated by the diagonal entry of the original substructure matrix. However, this is not directly applicable to the indefinite system (43), as, in general, matrix C^i contains only seldom nonzeros on the diagonal. For this reason, we approximate the diagonal of the Schur complement as

$$W^i_{jj} = \widetilde{C}^i_{\Gamma\Gamma,jj} + \frac{1}{A^i_{kk}}, \tag{87}$$

where the index k corresponds to the row in block A^i of the element face to which the Lagrange multiplier $\lambda^i_{\Gamma,j}$ belongs.

Using the diagonal stiffness scaling in connection with the standard Lagrange finite elements may lead to poor convergence for problems with rough interface [19, 38], for which the diagonal stiffness can vary quickly even for smooth problems with constant coefficients on uniform meshes. This is a severe issue for practical computations, in which graph partitioners are typically used for creating substructures. However, this issue is not as pronounced for Raviart-Thomas elements, for which only one element contributes to the stiffness on the diagonal at an interface degree of freedom, and thus irregularities caused by changing number of elements contributing to an interface weight cannot occur. On the other hand, an advantage of the diagonal stiffness scaling is the fact, that—unlike the ρ-scaling—it takes into account the shape and relative sizes of elements, which vary considerably in engineering applications, as well as the effect of δ_d introduced in (29) and (36). Unless stated otherwise, scaling (87) is used in the computations presented in Section 8.

7. THE PARALLEL SOLVER

The basis for an efficient parallel implementation of the method described in previous sections was obtained by combining two existing open-source software packages: the finite element package *Flow123d*[†] (version 1.6.5) for underground fluid flow simulations and the BDDC-based solver *BDDCML*[‡] (version 2.0) used for the solution of the resulting system of equations. However, minor changes have been made to both codes to support the specific features, such as the weights (86) and (87).

The *Flow123d* package has been developed for modelling complex behaviour of underground water flow and pollution transport. However, only the simple flow in a fully saturated porous media described by Darcy's law is considered in this paper. To accurately account for fractures in the medium, such as granite rock, the solver allows us to combine finite elements of different dimensions: the three-dimensional elements of porous media are combined with two-dimensional elements modelling planar fractures, which may be in turn connected in one-dimensional elements for channels. Raviart-Thomas elements are consistently used throughout such discretisation. Although the fractures are also modelled as porous media, their hydraulic conductivity is by orders of magnitude higher than that of the main porous material of the domain. In addition, the finite element discretisations are typically not uniform within the domain, and the relative sizes of elements may also vary by orders of magnitude. Both these aspects give rise to very poorly conditioned linear systems, which are very challenging for iterative solvers. The *Flow123d* solver has been developed for over 10 years and it is written in C/C++ programming language with object-oriented design and parallelism through MPI.

The *BDDCML* is a library for solving algebraic systems of linear equations by means of the BDDC method. The package supports the Adaptive-Multilevel BDDC method [22] suitable for very high number of substructures and computer cores, although we only use the standard (non-adaptive two-level) BDDC method from [6, 11] for the purpose of this paper. The *BDDCML* library is typically interfaced by finite element packages, which may provide the division into substructures. This feature is used in our current implementation, in which the division into non-overlapping substructures is constructed within the *Flow123d* using the *METIS* (version 5.0) package [39]. One substructure is assigned to a processor core in the current set-up of the parallel solver, although *BDDCML* is more flexible in this respect. The library performs the selection of additional corners by the face-based algorithm from [21]. The *BDDCML* package is written in Fortran 95 and parallelised through MPI.

The *BDDCML* solver relies on a serial instance of the *MUMPS* direct solver [40] for the solution of each local discrete Dirichlet problem (45) as well as for the solution of each local discrete Neumann problem (60). The coarse problem (62) is solved by a parallel instance of *MUMPS*. The main difference from using *BDDCML* for symmetric positive definite problems is the need to use the LDL^T factorisation of general symmetric matrices for problems (45), which are saddle-point (i.e. indefinite) systems in the present setting.

Although the original system (37) is indefinite, system (49) is symmetric positive definite, which allows the use of the preconditioned conjugate gradient (PCG) method. One step of BDDC is used as the preconditioner within the PCG method applied to problem (49). The matrix of problem (49) is not explicitly constructed in the solver, and only its actions on vectors are computed following (45)–(48).

Remark 10
In our implementation, we change the sign neither in the action of S^i (46) nor in the action of the preconditioner M_{BDDC} (64). Since both are then strictly negative definite, the product $M_{BDDC}\widehat{S}$ is the same as if both signs were changed, and the PCG method runs correctly. In this way, no changes are necessary in an implementation developed for symmetric positive definite problems.

[†] http://flow123d.github.io
[‡] http://users.math.cas.cz/~sistek/software/bddcml.html

8. NUMERICAL RESULTS

In this section, we investigate the performance of the algorithm and its parallel implementation on several benchmark problems in 2D and 3D, and on two geoengineering problems of existing localities in 3D. For the two benchmark problems without fractures we perform weak scaling tests. For the benchmark problem with fractures and for the geoengineering problems, we perform strong scaling tests with the problem size fixed and increasing number of processor cores. In all cases, the PCG iterations are run until the relative norm of residual $\|r^{(k)}\|/\|\widehat{b}\| < 10^{-7}$. If not stated otherwise, the proposed scaling by diagonal stiffness (87) is used within the averaging operator of BDDC.

8.1. Results for benchmark problems

First, the performance of the solver is investigated on a unit square and a unit cube discretised solely using two-dimensional and three-dimensional finite elements, respectively. For this reason, block $\overline{C}$ in system (37), which is related to combining elements of different dimension, is zero, and the problem reduces to the standard problem (14). The sequence of unstructured meshes is approximately uniform for both problems, and the problems do not contain any jumps in material coefficients. In Figs. 2 and 3, example meshes and the resulting pressure head and velocity fields are presented. While gravity is present in the 3D case, its effect is not considered in the 2D case.

The results of the weak scaling tests are summarised in Tables I and II. To give a better view, the resulting solution times for different problem sizes are also visualised in Fig. 4. In these tables, N denotes the number of substructures and processors, n is the size of the global problem (37), n_Γ is the size of the interface problem (49), n_f denotes number of faces, n_c denotes number of corners, 'its.' stands for resulting number of PCG iterations, and 'cond.' is the approximate condition number computed from the Lanczos sequence in PCG. We report separately the time spent in preconditioner set-up, the time spent by PCG iterations, and the total time for the whole solve.

In these weak scaling tests, the number of unknowns per core is kept approximately constant around 10^5. These weak scaling tests were performed using up to 64 cores of the SGI Altix UV supercomputer at the Supercomputing Centre of the Czech Technical University in Prague. The computer contains twelve Intel Xeon processors, each with six cores at frequency 2.67 GHz. Intel compilers version 12.0 were used.

The numbers of PCG iterations and condition number estimates in Tables I and II confirm the expected numerical scalability of the BDDC method, which is well known for symmetric positive definite problems as well as for Darcy's flow problems [6, 37]. The slight irregularities in the condition number in Table II are probably caused by using non-nested unstructured meshes.

Looking at times in these tables and in Fig. 4, we can see almost optimal scaling, with only mild growth of times with number of cores. The numbers of PCG iterations are higher in 3D, and the time spent in PCG iterations grows proportionally, while the time spent in the set-up phase does not differ considerably between two-dimensional and three-dimensional setting and dominates the overall time.

N	n	n/N	n_Γ	n_f	n_c	its.	cond.	time (sec)		
								set-up	PCG	solve
2	207k	103k	155	1	2	7	1.37	8.3	1.6	9.9
4	440k	110k	491	5	10	8	1.60	12.2	2.2	14.4
8	822k	103k	1.2k	13	26	9	1.78	11.0	2.5	13.5
16	1.8M	111k	2.8k	33	66	8	1.79	14.3	2.7	17.0
32	3.3M	104k	5.9k	74	148	9	1.79	12.1	3.3	15.4
64	7.2M	113k	13.0k	166	332	9	1.85	14.8	4.4	19.2

Table I. Weak scaling test for the 2D square problem, each substructure problem contains approx. 100k unknowns.

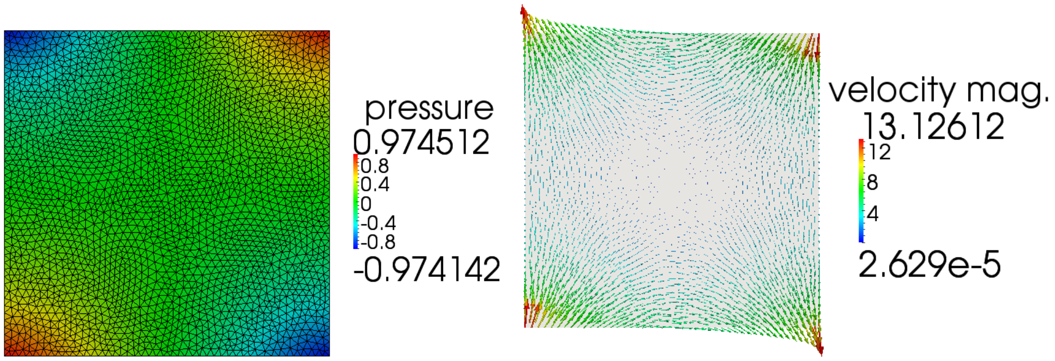

Figure 2. Example of solution to the model square problem containing only 2D elements, plot of pressure head with mesh (left) and velocity vectors (right).

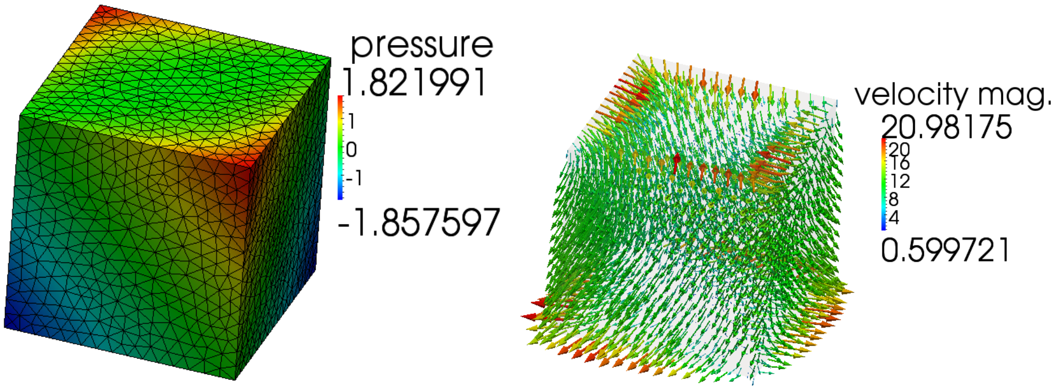

Figure 3. Example of solution to the model cube problem containing only 3D elements, plot of pressure head with mesh (left) and velocity vectors (right).

N	n	n/N	n_Γ	n_f	n_c	its.	cond.	time (sec)		
								set-up	PCG	solve
2	217k	108k	884	1	3	11	2.88	11.7	2.3	14.0
4	437k	109k	2.3k	6	18	12	3.04	11.7	2.5	14.2
8	945k	118k	5.7k	21	63	15	12.00	15.4	4.0	19.3
16	1.6M	103k	12.8k	56	168	16	6.58	12.9	4.0	17.0
32	3.4M	106k	29.8k	132	401	18	10.10	15.4	5.2	20.6
64	6.1M	95k	59.6k	307	931	19	16.58	13.7	6.3	20.0

Table II. Weak scaling test for the 3D cube problem, each substructure problem contains approx. 100k unknowns.

The next benchmark problem is considerably more complicated. It consists again of a unit cube, which now contains four planar fractures aligned with diagonals of a 2D cross-section. These four planar fractures meet at a 1D channel in the centre of the cross-section. Therefore, the problem contains the full possible combination of 3D, 2D and 1D finite elements. The tensor $\mathbb{k}$ is isotropic, thus it is just a scalar multiple of identity. The corresponding scalar value is set to 10, 1, and 0.1 for 1D, 2D, and 3D elements, respectively.

We perform a strong scaling test with this problem, keeping the mesh size fixed with approximately 2.1 million elements and 14.6 million degrees of freedom. In Fig. 5, the computational mesh and the resulting pressure head and velocity fields are presented. This scaling

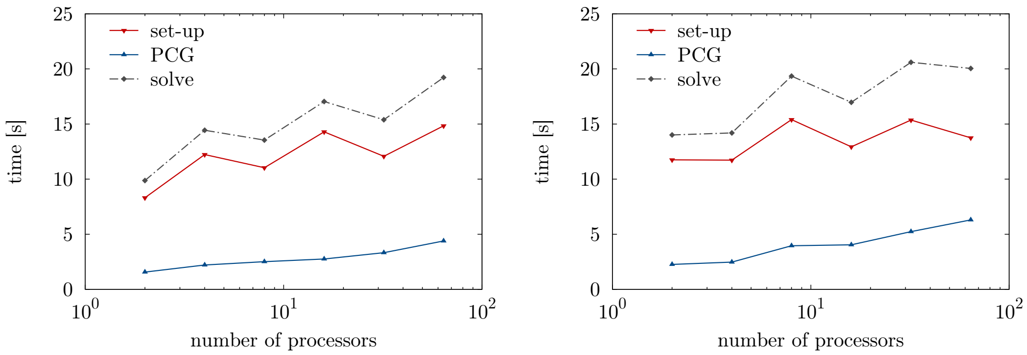

Figure 4. Weak scaling test for the 2D square problem (left), and the 3D cube problem (right), approx. 100k unknowns per core. Computational times separately for set-up and PCG phases, and their sum (solve).

test was computed on the Cray XE6 supercomputer *Hector* at the Edinburgh Parallel Computing Centre. This supercomputer is composed of 2816 nodes, each containing two AMD Opteron Interlagos processors with 16 cores at 2.3 GHz. GNU compilers version 4.6 were employed.

Results of the strong scaling test are summarised in Table III, and the computing times are visualised also in Fig. 6 together with the parallel speed-up. The reference value for computing speed-up is the time on 16 cores, and the speed-up on np processors is computed as

$$s_{np} = \frac{16\, t_{16}}{t_{np}}, \tag{88}$$

where t_{np} is the time on np processors.

We can see that the number of PCG iterations grows with the number of substructures for this problem which is also confirmed by the growing condition number estimate. While the time spent in set-up phase scales very well, the time spent in PCG grows together with the number of iterations. The reason for this growth seem to be related to the larger interface, at which more numerical difficulties appear. This seems to be related to more 1D-2D and 2D-3D connections at the interface and makes this difficult problem a good candidate for using the *Adaptive BDDC* method [22, 41]. However, this will be the subject of a separate study.

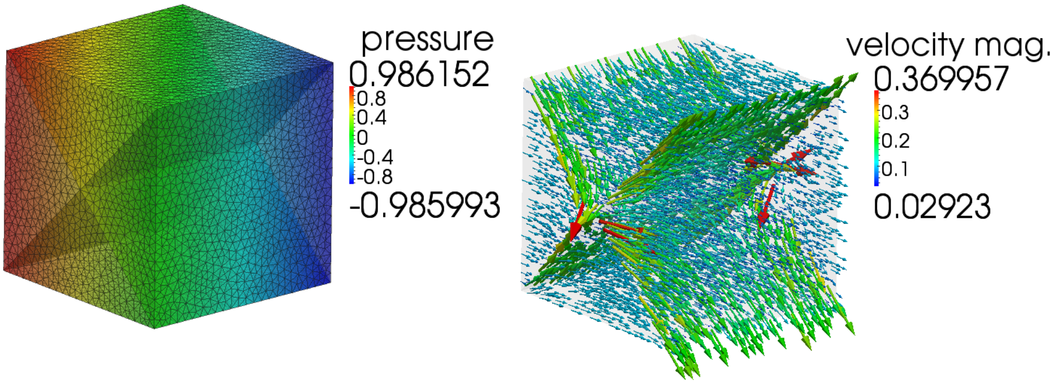

Figure 5. Example of solution to the model cube problem containing 1D, 2D, and 3D elements, plot of pressure head with mesh and fractures (left) and velocity vectors (right).

N	n/N	n_Γ	n_f	n_c	its.	cond.	time (sec)		
							set-up	PCG	solve
16	912k	47k	53	159	26	59.3	171.6	84.5	256.2
32	456k	65k	126	380	48	2091.0	90.1	109.8	200.0
64	228k	86k	301	914	81	1436.1	36.8	77.1	114.0
128	114k	116k	689	2076	109	2635.8	14.3	43.1	57.4
256	57k	151k	1436	4365	164	1700.5	6.7	31.2	38.0
512	28k	196k	3021	9244	254	42614.5	4.0	26.9	30.9

Table III. Strong scaling test for the cube problem with 1D, 2D, and 3D elements, size of the global problem is n = 14.6M unknowns.

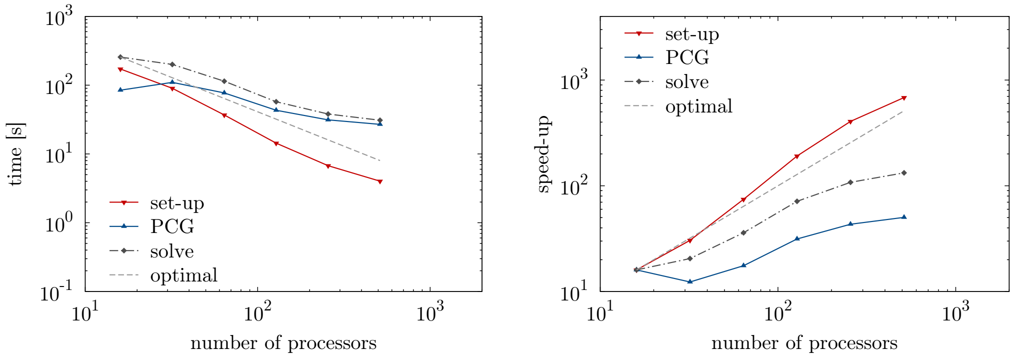

Figure 6. Strong scaling test for the cube problem with 1D, 2D, and 3D elements and 14.6M unknowns, computational time (left) and speed-up (right) separately for set-up and PCG phases, and their sum (solve).

8.2. *Results for geoengineering problems*

The performance of the algorithm and its parallel implementation has been investigated on two engineering problems of underground flows within real geologic locations. For both problems, the porous medium is fractured granite rock, with the fractures modelled by two-dimensional elements.

The first problem is the *Melechov locality*, which models one of the candidate sites for a nuclear waste deposit to be build within the Czech Republic in future. The goal is to model the underground flow and estimate the speed at which an eventual radioactive pollution would spread. The computational mesh contains 2.1 million finite elements resulting in 15 million unknowns. The geometry of the problem with the resulting distribution of piezometric head and the finite element mesh is presented in Fig. 7. The problem contains vertical two-dimensional fractures visualised in Fig. 8. The maximal hydraulic conductivity within the fractures is $6.3{\cdot}10^4$ ms^{-1}, while the minimal conductivity of the outer material is $6.0{\cdot}10^{-3}$ ms^{-1}, the transition coefficient σ_3 = 1 s^{-1}, and the effective thickness of fractures δ_2 = 0.1 m.

We perform a strong scaling test for this problem, keeping the problem size fixed and increasing the number of substructures and computing cores. An example of division into 64 substructures is presented in Fig. 8. The scaling test was computed on the *Hector* supercomputer.

Table IV summarises the results of this test. We can still see some growth of the number of iterations with the number of substructures, which is however much milder than the growth observed for the unit cube with fractures in Table III. Correspondingly, the times reported in Table IV and visualised in Fig. 9 show an optimal scaling of the solver over a large range of core counts.

The second engineering model is the locality around the *Bedřichov tunnel*. The main purpose of this 2.1 km long tunnel near the city of Liberec in the north of the Czech Republic to accommodate water pipes, which supply the city by drinking water from a reservoir in the mountains. However,

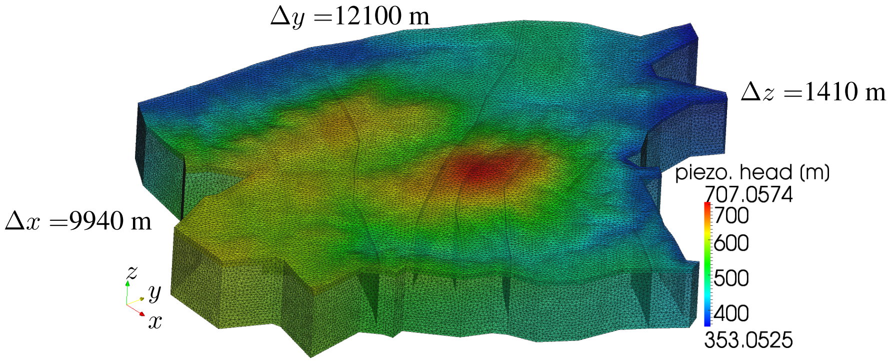

Figure 7. The problem of the *Melechov locality* containing 2D and 3D elements, mesh contains 2.1M elements and 15M unknowns. Plot of the piezometric head. Data by courtesy of Jiřina Královcová.

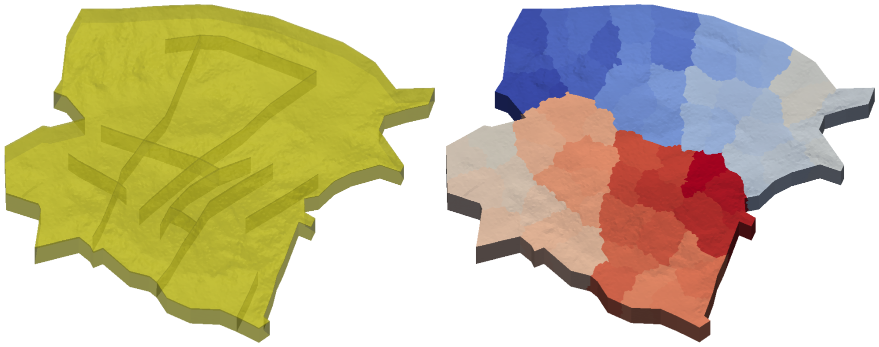

Figure 8. The problem of the *Melechov locality*; the system of fractures (left) and an example division into 64 substructures (right).

N	n/N	n_Γ	n_f	n_c	its.	cond.	time (sec)		
							set-up	PCG	solve
16	934k	36k	32	96	40	53.0	131.4	144.1	275.6
32	467k	54k	76	228	70	878.3	47.5	112.9	160.4
64	233k	82k	186	561	67	202.4	17.4	50.2	67.7
128	117k	116k	528	1592	69	237.6	7.9	23.1	31.1
256	58k	155k	1235	3747	96	5577.0	4.0	14.7	18.8
512	29k	207k	2699	8256	106	1658.1	2.2	8.3	10.5
1024	15k	271k	5711	17581	119	11554.5	2.1	7.0	9.2

Table IV. Strong scaling test for the problem of the *Melechov locality* containing 2D and 3D elements, size of the global problem is n = 15M unknowns.

this locality is also a valuable site for experimental geological measurements performed inside the tunnel.

The model aims at describing the flow in the granite rock surrounding the tunnel. The computational mesh consists of 1.1 million elements leading to 7.8 million unknowns. The mesh with the plot of resulting piezometric head is presented in Fig. 10. The system of fractures and an example division into 256 substructures are visualised in Fig. 11. The hydraulic conductivity of the fractures is 10^{-7} ms^{-1}, while the conductivity of the outer material is 10^{-10} ms^{-1}, the transition coefficient σ_3 = 1 s^{-1}, and the effective thickness of fractures δ_2 = 1.1 m.

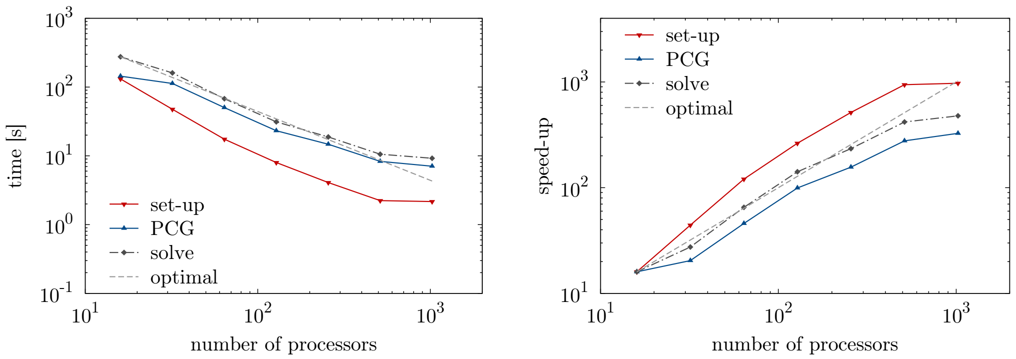

Figure 9. Strong scaling test for the problem of the *Melechov locality* containing 2D and 3D elements and 15M unknowns, computational time (left) and speed-up (right) separately for set-up and PCG phases, and their sum (solve).

Although the mesh contains fewer finite elements than the one of the *Melechov locality* model, this problem is considerably more complicated. This is caused mainly by the presence of relatively very small and irregularly shaped finite elements in the vicinity of the tunnel and near the cross-sections of fractures (see Fig. 12) generated by the mesh generator.

The results of a strong scaling test are summarised in Table V. As before, the times are also plotted in Fig. 13. Although the number of iterations is not independent of the number of substructures, the growth is still small. Consequently, the computing times, and especially the time for set-up, scale very well over a large range of numbers of substructures. The observed super-optimal scaling may be related to faster factorisation of the smaller local problems by the direct solver for indefinite matrices.

Table VI summarises an experiment performed to analyse the effect of using corners in the construction of the coarse space in BDDC. As has been mentioned in Section 5, using Raviart-Thomas finite elements does not lead to 'natural' corners as cross-points shared by several substructures. On the other hand, the notion of corners was generalised to any selected interface degree of freedom, at which continuity of functions from the coarse space is required. Such generalisation is important for the well-posedness of the local problems for unstructured meshes e.g. in elasticity analysis [21]. This is also the default option for *BDDCML*, in which selection of corners is performed at each face between two substructures. Adding corners improves the approximation properties of the coarse space at the cost of increasing the size of the coarse problem. Table VI compares the convergence for variable number of substructures without and with constraints at corners. Column 'with corners' in Table VI corresponds to results in Table V, and it is repeated for comparison. We can see that while the effect of corners on convergence is small for smaller number of substructures, the improvement of the coarse problem and the approximation of BDDC becomes more significant for higher numbers of cores. Looking at times in Table VI, the additional time spent in the set-up phase due to higher number of constraints when using corners is compensated by the lower number of PCG iterations, resulting in lower overall times. Thus, using additionally selected corners appears beneficial for complicated engineering problems like this one.

In the final experiment, we compare the effect of different averaging techniques on the convergence of BDDC. In Table VII, results of the strong scaling test for arithmetic averaging (85), the modified ρ-scaling (86), and the proposed scaling by diagonal stiffness (87) are summarised. The final column corresponds to the results from Table V, which are repeated here for comparison.

Table VII suggests, that while the simple arithmetic averaging does not lead to satisfactory convergence for this problem, the modified ρ-scaling and the diagonal scaling mostly lead to similar convergence. However, while the former provides slightly better convergence for several cases, it also leads to irregularities for certain divisions, for which the BDDC method with this averaging

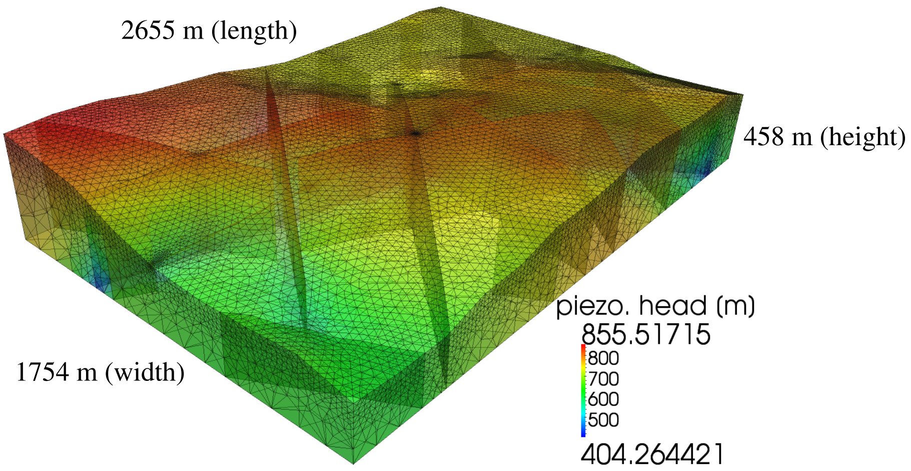

Figure 10. The *Bedřichov tunnel* problem containing 2D and 3D elements, mesh contains 1.1M elements and 7.8M unknowns. Plot of the piezometric head. Data by courtesy of Dalibor Frydrych.

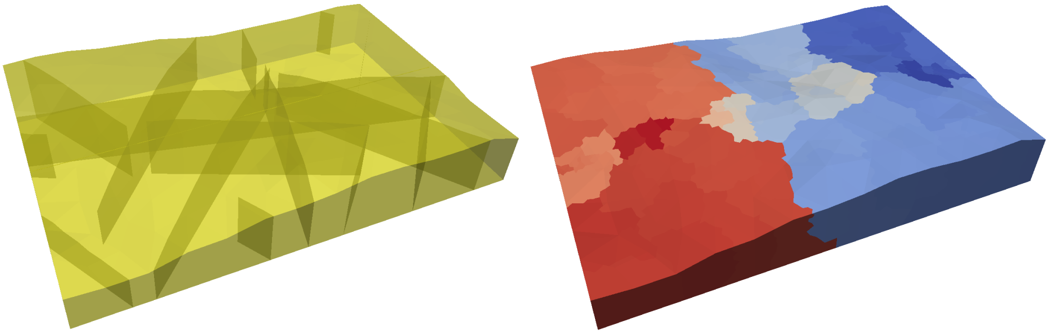

Figure 11. The *Bedřichov tunnel* problem; the system of fractures (left) and an example division into 256 substructures (right).

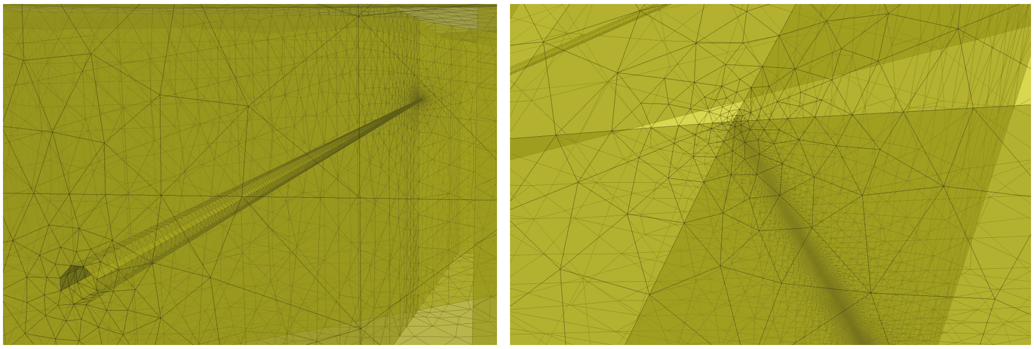

Figure 12. Example of difficulties in the mesh of the *Bedřichov tunnel* problem; detail of the tunnel geometry with fine elements (left) and enforced refinement at an intersection of fractures (right).

converges rather poorly. Therefore, the proposed scaling (87) can be recommended as the most robust choice among the three tested options.

N	n/N	n_Γ	n_f	n_c	its.	cond.	time (sec)		
							set-up	PCG	solve
32	245k	20k	106	322	112	1514.1	110.3	144.0	254.3
64	123k	28k	192	597	63	117.7	42.2	36.0	78.3
128	61k	45k	413	1293	75	194.4	13.4	16.8	30.3
256	31k	72k	902	2791	119	526.7	4.2	10.9	15.1
512	15k	110k	2009	6347	137	1143.4	1.8	7.1	9.0
1024	8k	155k	4575	14725	173	897.0	1.6	8.0	9.7

Table V. Strong scaling test for the problem of the *Bedřichov tunnel* containing 2D and 3D elements, size of the global problem is n = 7.8M unknowns.

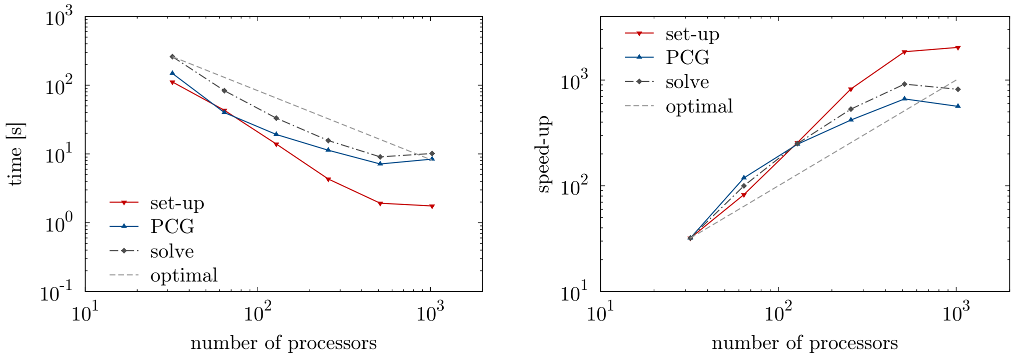

Figure 13. Strong scaling test for the problem of the *Bedřichov tunnel* containing 2D and 3D elements and 7.8M unknowns, computational time (left) and speed-up (right) separately for set-up and PCG phases, and their sum (solve).

N	n/N	without corners					with corners				
		its.	cond.	time (sec)			its.	cond.	time (sec)		
				set-up	PCG	solve			set-up	PCG	solve
32	245k	131	1789.6	107.5	175.0	282.5	112	1514.1	110.3	144.0	254.3
64	123k	70	122.2	40.3	40.4	80.7	63	117.7	42.2	36.0	78.3
128	61k	96	208.5	10.9	21.6	32.6	75	194.4	13.4	16.8	30.3
256	31k	139	541.9	3.7	12.5	16.2	119	526.7	4.2	10.9	15.1
512	15k	197	1418.9	1.4	10.0	11.4	137	1143.4	1.8	7.1	9.0
1024	8k	312	3779.4	1.0	14.5	15.6	173	897.0	1.6	8.0	9.7

Table VI. Effect of using corners. Problem of the *Bedřichov tunnel* containing 2D and 3D elements, size of the global problem is n = 7.8M unknowns. In column ‘without corners’, no additional corners are selected in BDDC. In column ‘with corners’, additional corners are selected.

9. CONCLUSION

A parallel solver for the mixed-hybrid finite element formulation based on Darcy’s law has been presented. The software combines an existing package *Flow123d* developed for problems in geophysics with *BDDCML*, a parallel library for solution of systems of linear algebraic equations by the BDDC method.

In geoengineering applications, the mathematical model is applied to geometries with the presence of fractures. In the present approach, the flow in these fractures is also modelled by Darcy’s law, although the hydraulic conductivity of the porous media is considered by orders of magnitude

N	n/N	n_Γ	n_f	n_c	arithmetic avg.		mod. ρ-scal.		diagonal scal.	
					its.	cond.	its.	cond.	its.	cond.
32	245k	20k	106	322	637	9811.7	110	1467.8	112	1514.1
64	123k	28k	192	597	618	10254.1	62	115.1	63	117.7
128	61k	45k	413	1293	2834	1.0e+11	206	401641.4	75	194.4
256	31k	72k	902	2791	799	11172.9	117	512.9	119	526.7
512	15k	110k	2009	6347	883	15449.6	136	1160.1	137	1143.4
1024	8k	155k	4575	14725	n/a	2.5e+10	504	99023.6	173	897.0

Table VII. Comparison of different averaging techniques for the *Bedřichov tunnel* containing 2D and 3D elements, size of the global problem is n = 7.8M unknowns.

higher. These fractures are modelled by finite elements of a lower dimension. In the discretised model, 1D, 2D and 3D finite elements are coupled together through Robin boundary conditions. These coupling terms lead to a modification of the usual saddle-point matrix of the system, in which a new non-zero block appears on the diagonal.

The BDDC method is employed for the solution of the resulting system of linear algebraic equations. BDDC is based on iterative substructuring, in which the problem is first reduced to the interface among substructures. The Schur complement is not built explicitly. Instead, only multiplications by the matrix are performed through solving a discrete Dirichlet problem on each substructure. In the setting of the mixed-hybrid problem, the interface is built only as a subset of the block of Lagrange multipliers, while remaining unknowns belong to interiors of substructures. Although the original problem is symmetric indefinite, the system reduced to the interface is symmetric positive definite. This is also shown to hold for the case with fractures in the present paper. Consequently, the PCG method is used for the solution of the reduced problem. However, unlike the symmetric positive definite problems, a direct solver for symmetric indefinite matrices needs to be used for the factorisation and repeated solution of local problems on substructures.

One step of the BDDC method is used as the preconditioner for the PCG method run on the interface problem. A modification of the diagonal stiffness scaling has been introduced. It is motivated by difficult engineering problems, for which it performs significantly better than other two applicable choices—the arithmetic averaging and the modified ρ-scaling. Arithmetic averages over faces between substructures are used as the basic constraints defining the coarse space. In addition, corners are selected from unknowns at the interface using the face-based algorithm. While corners are not required by the theory, they are shown to improve both the convergence and the computational times for complicated problems.

The performance of the resulting solver has been investigated on three benchmark problems in 2D and 3D. Both weak and strong scaling tests have been performed. On benchmark problems with single mesh dimension, the expected optimal convergence independent of number of substructures has been achieved. Correspondingly, the resulting parallel scalability has been nearly optimal for the weak scaling tests up to 64 computer cores.

The strong scaling tests were presented for a benchmark problem of a unit cube and for two engineering problems containing large variations in element sizes and hydraulic conductivities, using up to 1024 computer cores and containing up to 15 million degrees of freedom. The convergence for the unit cube problem with all three possible dimensions of finite elements slightly deteriorated by using more substructures, and this translated to sub-optimal parallel performance. However, for the two engineering applications, in which only 3D and 2D elements are combined, the BDDC method has also maintained good convergence properties with the growing number of substructures, resulting in optimal, or even super-optimal parallel scalability of the solver. It has been also shown that the proposed modification of the diagonal stiffness scaling plays an important role in achieving such independence for the challenging engineering problems presented in the paper.

ACKNOWLEDGEMENT

The authors are grateful to Jiřina Královcová and Dalibor Frydrych for providing the geoengineering models.

REFERENCES

1. Gulbransen AF, Hauge VL, Lie KA. A multiscale mixed finite-element method for vuggy and naturally-fractured reservoirs. *SPE Journal* 2010; **15**(2):395–403.
2. Martin V, Jaffré J, Roberts JE. Modeling fractures and barriers as interfaces for flow in porous media. *SIAM Journal on Scientific Computing* 2005; **26**(5):1667–1691.
3. Benzi M, Golub GH, Liesen J. Numerical solution of saddle point problems. *Acta Numerica* 2005; **14**:1–137.
4. Dohrmann CR, Lehoucq RB. A primal-based penalty preconditioner for elliptic saddle point systems. *SIAM Journal on Numerical Analysis* 2006; **44**(1):270–282.
5. Tu X. A BDDC algorithm for mixed formulation of flow in porous media. *Electronic Transactions on Numerical Analysis* 2005; **20**:164–179.
6. Tu X. A BDDC algorithm for flow in porous media with a hybrid finite element discretization. *Electronic Transactions on Numerical Analysis* 2007; **26**:146–160.
7. Vassilevski PS. *Multilevel block factorization preconditioners: matrix-based analysis and algorithms for solving finite element equations*. Springer, 2008.
8. Elman HC, Silvester DJ, Wathen AJ. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Numerical Mathematics and Scientific Computation, Oxford University Press, 2005.
9. Toselli A, Widlund OB. *Domain Decomposition Methods—Algorithms and Theory*, *Springer Series in Computational Mathematics*, vol. 34. Springer-Verlag, 2005.
10. Cros JM. A preconditioner for the Schur complement domain decomposition method. *Domain Decomposition Methods in Science and Engineering*, Herrera I, Keyes DE, Widlund OB (eds.). National Autonomous University of Mexico (UNAM), México, 2003; 373–380. 14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico, January 6–12, 2002.
11. Dohrmann CR. A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci. Comput.* 2003; **25**(1):246–258.
12. Fragakis Y, Papadrakakis M. The mosaic of high performance domain decomposition methods for structural mechanics: Formulation, interrelation and numerical efficiency of primal and dual methods. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**:3799–3830.
13. Mandel J, Sousedík B. BDDC and FETI-DP under minimalist assumptions. *Computing* 2007; **81**:269–280.
14. Sousedík B, Mandel J. On the equivalence of primal and dual substructuring preconditioners. *Electronic Transactions on Numerical Analysis* 2008; **31**:384–402.
15. Li J, Widlund OB. BDDC algorithms for incompressible Stokes equations. *SIAM Journal on Numerical Analysis* 2006; **44**(6):2432–2455.
16. Tu X, Li J. A balancing domain decomposition method by constraints for advection-diffusion problems. *Communications in Applied Mathematics and Computational Science* 2008; **3**(1):25–60.
17. Tu X. Three-level BDDC in three dimensions. *SIAM Journal on Scientific Computing* 2007; **29**(4):1759–1780.
18. Mandel J, Sousedík B, Dohrmann CR. Multispace and multilevel BDDC. *Computing* 2008; **83**(2-3):55–85.
19. Klawonn A, Rheinbach O, Widlund OB. An analysis of a FETI-DP algorithm on irregular subdomains in the plane. *SIAM Journal on Numerical Analysis* 2008; **46**(5):2484–2504.
20. Mandel J, Sousedík B. Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**(8):1389–1399.
21. Šístek J, Čertíková M, Burda P, Novotný J. Face-based selection of corners in 3D substructuring. *Mathematics and Computers in Simulation* 2012; **82**(10):1799–1811.
22. Sousedík B, Šístek J, Mandel J. Adaptive-Multilevel BDDC and its parallel implementation. *Computing* 2013; **95**(12):1087–1119.
23. Oh DS, Widlund OB, Dohrmann CR. A BDDC algorithm for Raviart-Thomas vector fields. *Technical Report TR-951*, Courant Institue of Mathematical Sciences Feb 2013. Department of Computer Science.
24. Sousedík B. Nested BDDC for a saddle-point problem. *Numerische Mathematik* 2013; **125**(4):761–783.
25. Tu X. A three-level BDDC algorithm for a saddle point problem. *Numerische Mathematik* 2011; **119**(1):189–217.
26. Maryška J, Rozložník M, Tůma M. Mixed-hybrid finite element approximation of the potential fluid flow problem. *Journal of Computational and Applied Mathematics* 1995; **63**:383–392.
27. Oden J, Lee J. Dual-mixed hybrid finite element method for second-order elliptic problems. *Mathematical Aspects of Finite Element Methods*, *Lecture Notes in Mathematics*, vol. 606, Galligani I, Magenes E (eds.). Springer Berlin Heidelberg, 1977; 275–291.
28. Maryška J, Rozložník M, Tůma M. Schur complement systems in the mixed-hybrid finite element approximation of the potential fluid flow problem. *SIAM Journal on Scientific Computing* 2000; **22**(2):704–723.
29. Maryška J, Severýn O, Vohralík M. Numerical simulation of fracture flow with a mixed-hybrid FEM stochastic discrete fracture network model. *Computational Geosciences* 2005; **8**:217–234.
30. Březina J, Hokr M. Mixed-hybrid formulation of multidimensional fracture flow. *Proceedings of the 7th international conference on Numerical methods and applications*, NMA'10, Springer-Verlag: Berlin, Heidelberg, 2011; 125–132.
31. Bear J. *Dynamics of Fluids in Porous Media*. Dover, 1988.
32. Chen Z, Huan G, Ma Y. *Computational Methods for Multiphase Flows in Porous Media*. SIAM, 2006.
33. Brezzi F, Fortin M. *Mixed and Hybrid Finite Element Methods*. Springer-Verlag, 1991.
34. Cowsar LC, Mandel J, Wheeler MF. Balancing domain decomposition for mixed finite elements. *Mathematics of Computation* 1995; **64**(211):989–1015.

35. Pultarová I. Preconditioning of the coarse problem in the method of balanced domain decomposition by constraints. *Math. Comput. Simulation* 2012; **82**(10):1788–1798.
36. Mandel J, Dohrmann CR, Tezaur R. An algebraic theory for primal and dual substructuring methods by constraints. *Appl. Numer. Math.* 2005; **54**(2):167–193.
37. Mandel J, Dohrmann CR. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numerical Linear Algebra with Applications* 2003; **10**(7):639–659.
38. Čertíková M, Šístek J, Burda P. On selection of interface weights in domain decomposition methods. *Proceedings of Programs and Algorithms of Numerical Mathematics 16, Dolní Maxov, Czech Republic, June 3–8, 2012*. Institute of Mathematics AS CR, 2013; 35–44.
39. Karypis G, Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing* 1998; **20**(1):359–392.
40. Amestoy PR, Duff IS, L'Excellent JY. Multifrontal parallel distributed symmetric and unsymmetric solvers. *Computer Methods in Applied Mechanics and Engineering* 2000; **184**:501–520.
41. Mandel J, Sousedík B, Šístek J. Adaptive BDDC in three dimensions. *Mathematics and Computers in Simulation* 2012; **82**(10):1812–1831.