



TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

Transport processes in fractured porous media

Habilitation thesis

Jan Březina

Liberec 2018





Abstract

This habilitation thesis summarizes author's theoretical work related to development of the Flow123d simulator. This includes especially methods and algorithms for solving Darcy flow problems in saturated and unsaturated fractured porous media.

A model with semi-discrete fractures called *mixed dimension model* is derived at the beginning. Then the abstract model for advection-diffusion equation is applied to the Darcy flow.

The mixed-hybrid formulation of the Darcy flow mixed dimension problem is presented followed by its discretization using Raviart-Thomas finite elements. An analytical solution to a test single fracture problem is supplied which allows verification of the model's implementation. Finally, the BDDC method is applied to obtain a scalable solver of the linear systems arising from the problem's discretization.

Subsequently, new developments for the non-conforming mixed meshes are presented. Four methods with common strategy are used to introduce a coupling between equations living on the intersecting finite element meshes of different dimension. Further a family of efficient algorithms for computing mesh intersections is presented.

Final chapter is devoted to the Richards' equation and modification of the mixed-hybrid scheme in order to satisfy discrete maximum principle. This is of particular importance for the Richards' equation where short time steps are often necessary which leads to strong oscillations for the schemes that violate DMP.





Acknowledgement

All work summarized in this thesis, in particular the reprinted papers, is result of collaboration with my colleagues and other researchers. I gratefully remember work with Tomáš Vogel, Milena Císlerová, Michal Dohnal, and Jaromír Dušek on solvers of the Richards' equation which resulted in the reprinted publication in VZJ.

Most of the work is related to the development of the Flow123d simulator. In the first place, I want to thank to Jan Stebel my closest colleague and key contributor to the Flow123d project. However many other people have contributed to this project through its existence. I want to mention especially Jakub Šístek, David Flanderka, Jan Hybš, Pavel Exner, Jiří Kopal, and Jiří Hnídek. My thanks go also to the colleagues who use and test Flow123d: Milan Hokr, Aleš Balvín, Jiřina Královcová, Jakub Říha, and Josef Chudoba. Their constructive and valuable feedback helps to constantly improve the simulator and make it useful for the real applications. At this place I want to acknowledge the pioneering work of Otto Severýn at early days of the simulator and I thank to Jiří Maryška for the leadership and unflagging search for applications of the simulator.

Last and most importantly I'm grateful to my wife Pavlína for her support, extreme patience and understanding for my work.





Contents

1	Introduction	1
2	Continuum-fracture model	3
2.1	Continuum-fracture model for advection-diffusion processes	3
2.2	Meshes of Mixed Dimension	5
2.3	Analysis of continuum-fracture model	6
3	Numerical methods for conforming mixed meshes	17
3.1	Darcy Flow Model	17
3.1.1	Coupling on mixed meshes	18
3.1.2	Boundary conditions	19
3.1.3	Water balance	20
3.2	Mixed formulations with conforming fractures	21
3.2.1	Mixed formulations on mixed meshed	21
3.3	Mixed-hybrid finite element method	23
3.4	Test problem with analytical solution	24
3.5	Scalable BDDC solver for conforming mixed mesh	39
4	Numerical methods for non-conforming mixed meshes	67
4.1	Mixed-Hybrid Method on Non-conforming Mixed Meshes	67
4.1.1	Direct P^0 Method	68
4.1.2	Direct P^1 Method	68
4.1.3	Mortar Like P^0 Method	69
4.1.4	Mortar Like P^1 Method	70
4.1.5	Coupling of codimension 0	70
4.2	Intersections of non-conforming mixed meshes	71
5	Unsaturated Darcy flow on mixed meshes	87
5.1	Richards' equation and instability of Raviart-Thomas elements	87
5.2	Mixed finite elements	88
5.3	Comparison of mixed and primary discretization	89
5.4	Discrete maximum principle	90
5.5	Lumped Mixed-Hybrid Method	92
5.6	Fully coupled dual permeability model	93
	Bibliography	103





Chapter 1

Introduction

Modeling of the groundwater flow and associated transport processes is important for a wide range of applications. In particular the long term safety of underground radioactive waste deposits depends on slow water flow in target geological formations which should ensure a slow transport and a large dilution of the leaking contaminants. For the deposits located in granitoids, the compact rock have extremely small permeability. However, in practice, a network of faults and fractures of very different scale is presented, causing a higher effective permeability and formation of preferential flow paths through the faults of the largest scale. The water moves slowly through the micro-scale fractures on the major part of the rock however it moves rapidly on the small part of the rock occupied by the macro-scale faults. This multi-scale nature have significant impact on the transport times and dilution of the contaminants. This implies that explicit description of the fractures is indispensable for a numerical model used for groundwater flow predictions for the nuclear waste deposits. The models of fractured porous media can also be applied in design of enhanced geothermal systems, in remediation technologies, for description of root zones in the soil and others.

Although the volumetric flow through the large scale fractures can dominate the flow through the bulk volume, the cross-section, even of the largest fractures, is small compared to the diameter of the whole domain. This difference of scales can be treated by intensive mesh refinement in vicinity of the fractures but at a price of significantly increased number of mesh elements. An alternative are models based on *mixed meshes*. In this case the governing equations are integrated across the fracture's opening and are discretized by the elements of lower dimension.

The research covered by this thesis is focused mainly on the Darcy flow with exception of Chapter 2, which discuss derivation of a continuum-fracture model for a general advection-diffusion equation. Included paper provides both theoretical and numerical analysis of the model's error for small fracture cross-sections.

Subsequent Chapter 3 is devoted to the application of these general results in the case of Darcy flow. A mixed-hybrid formulation of the resulting problem is presented as well as its discretization using mixed meshes and RT_0 finite elements. Two papers are included in this chapter. The first is a preprint of just finished work dealing with derivation of the analytical solution to a test Darcy flow problem with a fracture. The second work presents application of the BDDC method to the linear systems arising from the mixed-hybrid finite element method on mixed meshes.

In Chapter 3, only the *conforming mixed meshes* are treated. That is with assumption of the fracture elements laying on faces of the bulk elements. Fracture coupling models for *non-conforming mixed meshes* are disused in Chapter 4. The reprinted paper presents a family of efficient algorithms based on Plücker coordinates for computing intersections of the elements in

non-conforming meshes .

The final Chapter 5 discuss a solution to some peculiarities of the unsteady Darcy flow in particular unsaturated flow described by the Richards' equation when the mixed-formulation is used. We show that the discrete maximum principle is violated for these numerical schemes and we introduce the lumped mixed-hybrid scheme and demonstrate its stabilization effect. The included paper provides application of a fully coupled dual permeability Richards' model to for modeling infiltration into a soil.

All numerical methods, algorithms and theoretical concepts covered in this work with exception of the last paper has been implemented as part of the simulator Flow123d [21]. This inevitably assumes contribution of the other developers and colleagues. For the sake of clarity the reprinted papers are marked by the continuous line at the margin.



Chapter 2

Continuum-fracture model

Realistic modeling of subsurface water flow has to deal with highly heterogeneous and multi-scale nature of the hydraulic properties of the real rock. The water moves slowly on the majority of the rock volume through microscopic pores and fractures while it moves rapidly on the small part of the rock volume occupied by larger fractures that forms preferential flow paths. These paths may be highly localized and the volumetric flow rate in them may be comparable or even dominating flow rate in the bulk volume. However, a cross-section of the larger fractures is still very small compared to the length scale of the whole domain, thus one has to refine computational mesh along the fractures in order to render them properly, which can lead to the meshes that are intractably larger, especially if the network of fractures is dense enough.

To overcome these difficulties, we integrate the flow equations across the aperture of the fractures and derive a system of coupled equations living on domains of different dimension which we shall call a *mixed dimension model*.

2.1 Continuum-fracture model for advection-diffusion processes

The equations modeling a physical process on a manifold as well as its coupling to the model in the surrounding continuum has to be derived from the model on the 3d continuum. This section presents such a procedure for the case of the abstract advection-diffusion process inspired by the paper [13]. This abstract approach can be applied to advection-diffusion processes of different nature, in particular to Darcy flow, solute transport, and heat transfer.

Let us consider a fracture as a strip domain

$$\Omega_f \subset [0, \delta] \times \mathbf{R}^{d-1}$$

for $d = 2$ or $d = 3$ and surrounding continuum domains

$$\Omega_1 \subset (-\infty, 0) \times \mathbf{R}^{d-1}, \Omega_2 \subset (\delta, \infty) \times \mathbf{R}^{d-1}.$$

Further, we denote by γ_i , $i = 1, 2$ the fracture faces common with domains Ω_1 and Ω_2 respectively. By x , \mathbf{y} we denote normal and tangential coordinate of a point in Ω_f . We consider the normal vector $\mathbf{n} = \mathbf{n}_1 = -\mathbf{n}_2 = (1, 0, 0)^\top$. An advection-diffusion process is given by equations:

$$\partial_t w_i + \operatorname{div} \mathbf{j}_i = f_i \quad \text{on } \Omega_i, \quad i = 1, 2, f, \quad (2.1)$$

$$\mathbf{j}_i = -\mathbb{A}_i \nabla u_i + \mathbf{b}_i w_i \quad \text{on } \Omega_i, \quad i = 1, 2, f, \quad (2.2)$$

$$u_i = u_f \quad \text{on } \gamma_i, \quad i = 1, 2, \quad (2.3)$$

$$\mathbf{j}_i \cdot \mathbf{n} = \mathbf{j}_f \cdot \mathbf{n} \quad \text{on } \gamma_i, \quad i = 1, 2, \quad (2.4)$$

where $w_i = w_i(u_i)$ is the conservative quantity and u_i is the principal unknown, \mathbf{j}_i is the flux of w_i , f_i is the source term, \mathbb{A}_i is the diffusivity tensor and \mathbf{b}_i is the velocity field. We assume that the tensor \mathbb{A}_f is symmetric positive definite with one eigenvector in the direction \mathbf{n} . Consequently the tensor has the form:

$$A_f = \begin{pmatrix} a_n & 0 \\ 0 & \mathbb{A}_t \end{pmatrix}$$

Furthermore, we assume that $\mathbb{A}_f(x, \mathbf{y}) = \mathbb{A}_f(\mathbf{y})$ is constant in the normal direction.

Our next aim is to integrate equations on the fracture Ω_f in the normal direction and obtain their approximations on the surface $\gamma = \Omega_f \cap \{x = \delta/2\}$ running through the middle of the fracture. For the sake of clarity, we will not write subscript f for quantities on the fracture. To make the following procedure mathematically correct we have to assume that functions $\partial_x w$, $\partial_x \nabla_{\mathbf{y}} u$, $\partial_x \mathbf{b}_{\mathbf{y}}$ are continuous and bounded on Ω_f . Here and later on $\mathbf{b}_x = (\mathbf{b} \cdot \mathbf{n}) \mathbf{n}$ is the normal part of the velocity field and $\mathbf{b}_{\mathbf{y}} = \mathbf{b} - \mathbf{b}_x$ is the tangential part. The same notation will be used for normal and tangential part of the field \mathbf{q} .

We integrate (2.1) over the fracture opening $[0, \delta]$ and use approximations to get

$$\partial_t(\delta W) - \mathbf{j}_2 \cdot \mathbf{n}_2 - \mathbf{j}_1 \cdot \mathbf{n}_1 + \operatorname{div} \mathbf{J} = \delta F, \quad (2.5)$$

where for the first term, we have used mean value theorem, first order Taylor expansion, and boundedness of $\partial_x w$ to obtain approximation:

$$\int_0^\delta w(x, \mathbf{y}) \, dx = \delta w(\xi_{\mathbf{y}}, \mathbf{y}) = \delta W(\mathbf{y}) + O(\delta^2 |\partial_x w|),$$

where

$$W(\mathbf{y}) = w(\delta/2, \mathbf{y}) = w(u(\delta/2, \mathbf{y})) = w(U(\mathbf{y})).$$

Next two terms in (2.5) come from the exact integration of the divergence of the normal flux \mathbf{j}_x . Integration of the divergence of the tangential flux $\mathbf{j}_{\mathbf{y}}$ gives the fourth term, where we introduced

$$\mathbf{J}(\mathbf{y}) = \int_0^\delta \mathbf{j}_{\mathbf{y}}(x, \mathbf{y}) \, dx.$$

In fact, this flux on γ is scalar for the case $d = 2$. Finally, we integrate the right-hand side to get

$$\int_0^\delta f(x, \mathbf{y}) \, dx = \delta F(\mathbf{y}) + O(\delta^2 |\partial_x f|), \quad F(\mathbf{y}) = f(\delta/2, \mathbf{y}).$$

Due to the particular form of the tensor \mathbb{A}_f , we can separately integrate tangential and normal part of the flux given by (2.2). Integrating the tangential part and using approximations

$$\int_0^\delta \nabla_{\mathbf{y}} u(x, \mathbf{y}) \, dx = \delta \nabla_{\mathbf{y}} u(\xi_{\mathbf{y}}, \mathbf{y}) = \delta \nabla_{\mathbf{y}} U(\mathbf{y}) + O(\delta^2 |\partial_x \nabla_{\mathbf{y}} u|)$$

and

$$\int_0^\delta (\mathbf{b}_{\mathbf{y}} w)(x, \mathbf{y}) \, dx = \delta \mathbf{B}(\mathbf{y}) W(\mathbf{y}) + O(\delta^2 |\partial_x (\mathbf{b}_{\mathbf{y}} w)|)$$

where

$$\mathbf{B}(\mathbf{y}) = \mathbf{b}_{\mathbf{y}}(\delta/2, \mathbf{y}),$$

we obtain

$$\mathbf{J} = -\mathbb{A}_t \delta \nabla_{\mathbf{y}} U + \delta \mathbf{B} W + O(\delta^2(|\partial_x \nabla_{\mathbf{y}} u| + |\partial_x(\mathbf{b}_{\mathbf{y}} w)|)). \quad (2.6)$$

So far, we have derived equations for the state quantities U and \mathbf{J} on the fracture manifold γ . In order to get a well posed problem, we have to prescribe two conditions for boundaries γ_i , $i = 1, 2$. To this end, we perform integration of the normal flux \mathbf{j}_x , given by (2.2), separately for the left and right half of the fracture. Similarly as before we use approximations

$$\int_0^{\delta/2} \mathbf{j}_x dx = (\mathbf{j}_1 \cdot \mathbf{n}_1) \frac{\delta}{2} + O(\delta^2 |\partial_x \mathbf{j}_x|)$$

and

$$\int_0^{\delta/2} \mathbf{b}_x w dx = (\mathbf{b}_1 \cdot \mathbf{n}_1) \tilde{w}_1 \frac{\delta}{2} + O(\delta^2 |\partial_x \mathbf{b}_x| |w| + \delta^2 |\mathbf{b}_x| |\partial_x w|)$$

and their counter parts on the interval $(\delta/2, \delta)$ to get

$$\mathbf{j}_1 \cdot \mathbf{n}_1 = -\frac{2a_n}{\delta} (U - u_1) + \mathbf{b}_1 \cdot \mathbf{n}_1 \tilde{w}_1 \quad (2.7)$$

$$\mathbf{j}_2 \cdot \mathbf{n}_2 = -\frac{2a_n}{\delta} (U - u_2) + \mathbf{b}_2 \cdot \mathbf{n}_2 \tilde{w}_2 \quad (2.8)$$

where \tilde{w}_i can be any convex combination of w_i and W . Equations (2.7) and (2.8) have meaning of a semi-discretized flux from domains Ω_i into fracture. In order to get a stable numerical scheme, we introduce a kind of upwind already on this level using a different convex combination for each flow direction:

$$\begin{aligned} \mathbf{j}_i \cdot \mathbf{n}_i = & -\sigma_i (U - u_i) \\ & + [\mathbf{b}_i \cdot \mathbf{n}_i]^+ (\xi w_i + (1 - \xi) W) \\ & + [\mathbf{b}_i \cdot \mathbf{n}_i]^- ((1 - \xi) w_i + \xi W), \quad i = 1, 2 \end{aligned} \quad (2.9)$$

where $\sigma_i = \frac{2a_n}{\delta}$ is the transition coefficient and the parameter $\xi \in [\frac{1}{2}, 1]$ can be used to interpolate between upwind ($\xi = 1$) and central difference ($\xi = \frac{1}{2}$) scheme. Equations (2.5), (2.6), and (2.9) describe the general form of the advection-diffusion process on the fracture and its communication with the surrounding continuum which we shall later apply to individual processes.

2.2 Meshes of Mixed Dimension

The results of the previous section can be extrapolated to the general case of coupled 1d channels, 2d fractures and 3d continuum in a 3d ambient space. Let $\Omega_3 \subset \mathbf{R}^3$ be an open set representing continuous approximation of porous and fractured medium. Similarly, we consider a set of 2d manifolds $\Omega_2 \subset \overline{\Omega}_3$, representing the 2d fractures and a set of 1d curves $\Omega_1 \subset \overline{\Omega}_2$ representing the 1d channels or preferential paths (see Fig 2.1). We assume that Ω_2 and Ω_1 are polytopic (i.e. polygonal and piecewise linear, respectively). For every dimension $d = 1, 2, 3$, we introduce a triangulation \mathcal{T}_d of the open set Ω_d that consists of finite elements T_d^i , $i = 1, \dots, N_E^d$. The elements are simplices, i.e. lines, triangles and tetrahedra, respectively. The union of the meshes $\mathcal{T} = \mathcal{T}_d$ is called *mixed mesh*.

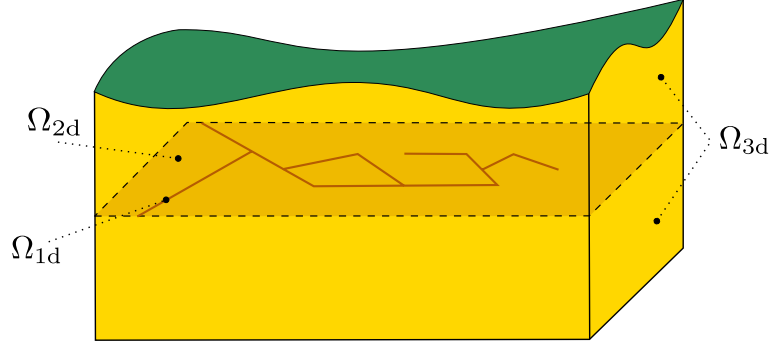


Figure 2.1: Scheme of a problem with domains of multiple dimensions.

Further more we have to distinguish the conforming and the non-conforming mixed meshes. The conforming mixed mesh must satisfy the compatibility conditions:

$$T_{d-1}^i \cap T_d \subset \mathcal{F}_d, \quad \text{where } \mathcal{F}_d = \bigcup_k \partial T_d^k \quad (2.10)$$

and

$$T_{d-1}^i \cap \mathcal{F}_d \text{ is either } T_{d-1}^i \text{ or } \emptyset \quad (2.11)$$


for every $i \in \{1, \dots, N_E^{d-1}\}$, $j \in \{1, \dots, N_E^d\}$, and $d = 2, 3$. That is, the $(d-1)$ -dimensional elements are either between d -dimensional elements and match their sides or they poke out of Ω_d . On the other hand the non-conforming mesh is union of arbitrary intersecting submeshes \mathcal{T}_d .

For complex geometries with many fractures it could be difficult to obtain a conforming mesh or it can lead to massive locale refinement and/or to meshes with elements of low quality (with very acute angles). On the other hand the calculation on non-conforming meshes needs: (i) algorithms to detect element intersections at least for 1d-2d, 1d-3d, 2d-2d, 2d-3d cases and (ii) a method to prescribe coupling between equations living on intersecting elements.

2.3 Analysis of continuum-fracture model

Application of the continuum-fracture model to the Darcy flow for a general mixed dimension domain is presented in Section 3.1. The following paper presents both theoretical and numerical analysis of the error between original continuum model and reduced continuum-fracture model. Theoretical error estimates are presented for the weak solution of the stationary Darcy flow problem and numerical approximation of the error in dependence of the mesh resolution and the fracture opening is studied.

Analysis of Model Error for a Continuum-Fracture Model of Porous Media Flow

Jan Březina^() and Jan Stebel

Technical University of Liberec, Studentská 1402/2, 46117 Liberec, Czech Republic
{jan.brezina,jan.stebel}@tul.cz

Abstract. The Darcy flow problem in fractured porous media is considered. The fractures are treated as lower dimensional objects coupled with the surrounding continuum. Error estimates for the weak solution to such continuum-fracture model in comparison to the weak solution of the full model are derived. Validity of the estimates is inspected on one simple and one quasi-realistic case numerically.

Keywords: Darcy flow · Fractured media · Reduced model · Error estimate

1 Introduction

Deep subsurface deposits in a plutonic rock represent one of possible solutions for the final storage of radioactive waste. The primary reason is small hydraulic permeability of the bulk rock and thus slow migration of a possible leakage due to the ground water flow. On the other hand, granitoid formations contain fractures that may form a network of preferential paths with low volumetric water flow rate but with high velocity. The preferential paths pose a risk of fast transport of small amount of contaminant but in potentially dangerous concentrations. The large scale effect of the small scale fractures is challenging for numerical simulations since direct discretization requires highly refined computational mesh. One possible solution is to model fractures as lower dimensional objects and introduce their coupling with the surrounding continuum. A model for the saturated flow in the system matrix-fracture was formally derived in [7] by integrating the equations across the fracture. It was justified by an error estimate $O(\max\{h, \delta\})$, h being the mesh size and δ the fracture width, which holds inside the fracture for the solution of a particular mixed finite element approximation. The approach was then generalized by others e.g. to the case of curved fractures with variable width [1], non-matching grids [3] or to other equations or systems [4–6]. While most papers aim at the analysis or numerical solution of the continuum-fracture model, the precise statement declaring the relation of the original and reduced problem on the continuous level is, to our knowledge, missing. The presented estimates hold for the pressure gradient, which in turn controls the error in the velocity field which is required for the practical solute transport problems.

© Springer International Publishing Switzerland 2016
T. Kozubek et al. (Eds.): HPCSE 2015, LNCS 9611, pp. 152–160, 2016.
DOI: 10.1007/978-3-319-40361-8_11

TECHNICAL UNIVERSITY OF LIBEREC | Faculty of Mechatronics, Informatics and Interdisciplinary Studies | Studentská 1402/2 | 46117 Liberec 1 | Czech Republic
phone: +420 485 353 624 | jan.brezina@tul.cz | www.fm.tul.cz | ID: 467 47 885 | VATIN: CZ 467 47 885



In this paper we shall study the Darcy flow model, namely

$$\left. \begin{aligned} \operatorname{div} \mathbf{q} &= f && \text{in } \Omega, \\ \mathbf{q} &= -\mathbb{K} \nabla p && \text{in } \Omega, \\ p &= p_0 && \text{on } \partial\Omega, \end{aligned} \right\} \quad (1)$$

where \mathbf{q} is the Darcy flux, f is the source density, \mathbb{K} is the hydraulic conductivity tensor, p is the piezometric head and p_0 is the piezometric head on the boundary. In what follows, $\Omega \subset \mathbf{R}^d$, $d = 2, 3$ will be a bounded domain with Lipschitz boundary (see Fig. 1, left), divided into the fracture

$$\Omega_f := \Omega \cap ((-\delta/2, \delta/2) \times \mathbf{R}^{d-1})$$

with thickness $\delta > 0$, and the surrounding set $\Omega_m := \Omega \setminus \overline{\Omega}_f$, called the matrix. The fracture interacts with the matrix on the interfaces

$$\gamma_1 := \Omega \cap (\{-\delta/2\} \times \mathbf{R}^{d-1}) \text{ and } \gamma_2 := \Omega \cap (\{\delta/2\} \times \mathbf{R}^{d-1}).$$

Normal vectors on these interfaces are denoted \mathbf{n}_i , $i = 1, 2$ with the orientation out of Ω_m . Further, we introduce the reduced geometry (see Fig. 1, right) where the fracture is represented by the manifold $\gamma := \Omega \cap (\{0\} \times \mathbf{R}^{d-1})$ in its center. For a point $\mathbf{x} \in \mathbf{R}^d$, we shall write $\mathbf{x} = (x, \mathbf{y})^\top$, $\mathbf{y} \in \mathbf{R}^{d-1}$. For functions defined in Ω_f we define the tangent gradient along the fracture

$$\nabla_{\mathbf{y}} v := (0, \partial_{y_1} v, \dots, \partial_{y_{d-1}} v)^\top,$$

and the average of v across the fracture:

$$\bar{v} := \frac{1}{\delta} \int_{-\delta/2}^{\delta/2} v(x, \cdot) dx.$$

We shall study the relation of (1) to the so-called *continuum-fracture model* on the reduced geometry:

$$\left. \begin{aligned} -\operatorname{div}(\mathbb{K} \nabla p_m) &= f && \text{in } \Omega_m, \\ p_m &= p_0 && \text{on } \partial\Omega \cap \partial\Omega_m, \\ -\mathbb{K} \nabla p_m \cdot \mathbf{n}_i &= q_i(p_m, p_f) && \text{on } \gamma_i, \ i = 1, 2, \\ -\operatorname{div}(\delta \mathbb{K} \nabla_{\mathbf{y}} p_f) &= \delta \bar{f} + \sum_{i=1}^2 q_i(p_m, p_f) && \text{in } \gamma, \\ p_f &= p_0 && \text{on } \bar{\gamma} \cap \partial\Omega. \end{aligned} \right\} \quad (2)$$

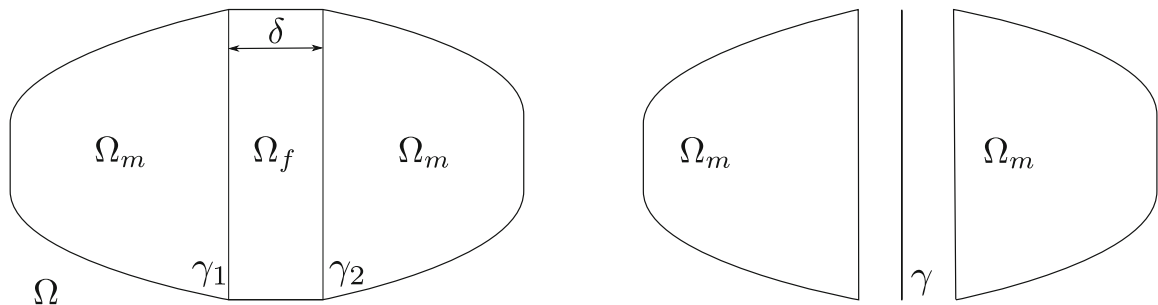


Fig. 1. The domain of the full model (left) and the reduced geometry (right).

The fluxes q_1, q_2 between the fracture and the matrix are given as follows:

$$q_i(v, w) := \frac{2\mathbb{K}_{|\gamma} \mathbf{n}_i \cdot \mathbf{n}_i}{\delta} (v|_{\gamma_i} - w|_{\gamma}), \quad i = 1, 2,$$

where v and w are defined on $\overline{\Omega}_m$ and $\overline{\gamma}$ respectively. Our goal is to justify (2) as an approximation of (1) in the case of small δ . In particular, we shall prove that

$$\bar{u} - u_f \approx \delta \quad \text{and} \quad u|_{\Omega_m} - u_m \approx \delta^{3/2}$$

in a suitable sense.

The organization of the paper is as follows. In the next section we formulate and prove the main theoretical result on the error analysis. Then, in Sect. 3 we show numerical results which confirm the error estimates.

2 Asymptotic Properties of Continuum-Fracture Model

In what follows we assume that \mathbb{K} is uniformly positive definite, bounded in Ω and has the following form:

$$\mathbb{K} = \begin{cases} \mathbb{K}_m & \text{in } \Omega_m, \\ \mathbb{K}_f = \begin{pmatrix} k_x & 0 \\ 0 & \mathbb{K}_y \end{pmatrix} & \text{in } \Omega_f, \end{cases}$$

where $\mathbb{K}_f(x, \mathbf{y}) = \mathbb{K}_f(\mathbf{y})$. Further we consider right hand side $f \in L^2(\Omega)$. As the problem is linear, we can set $p_0 \equiv 0$ without loss of generality.

2.1 Weak Formulation

The ongoing analysis will be done in the framework of weak solutions. By $L^q(B)$ we denote the Lebesgue space on a measurable set B endowed with the norm $\|\cdot\|_{q,B}$, $q \in [1, \infty]$, $H^1(B)$ is the Sobolev space and $H_0^1(B)$ its subspace of functions with vanishing trace. We say that $p \in H_0^1(\Omega)$ is the weak solution of (1) if for every $v \in H_0^1(\Omega)$:

$$\int_{\Omega} \mathbb{K} \nabla p \cdot \nabla v = \int_{\Omega} f v. \quad (3)$$

Introducing the space

$$H_{bc}^1(\Omega_m) := \{v \in H^1(\Omega_m); v|_{\partial\Omega_m \cap \partial\Omega} = 0\},$$

we analogously define the weak solution of (2) as the couple $(p_m, p_f) \in H_{bc}^1(\Omega_m) \times H_0^1(\gamma)$ that satisfies

$$\begin{aligned} \int_{\Omega_m} \mathbb{K}_m \nabla p_m \cdot \nabla v_m + \delta \int_{\gamma} \mathbb{K}_y \nabla_y p_f \cdot \nabla_y v_f \\ + \sum_{i=1}^2 \int_{\gamma_i} q_i(p_m, p_f) (v_m|_{\gamma_i} - v_f) = \int_{\Omega_m} f v_m + \delta \int_{\gamma} \bar{f} v_f \end{aligned} \quad (4)$$

for all $(v_m, v_f) \in H_{bc}^1(\Omega_m) \times H_0^1(\gamma)$.

Let us remark that under the above assumptions on \mathbb{K} and f , problems (3) and (4) have unique solutions.

2.2 Error Analysis of Asymptotic Model

Let $\sigma(\mathbb{A})$ denote the spectrum of a matrix \mathbb{A} . We use the following notation:

$$\underline{K}_m := \inf_{\mathbf{x} \in \Omega_m} \sigma(\mathbb{K}_m(\mathbf{x})), \quad \underline{K}_y := \inf_{\mathbf{x} \in \Omega_f} \sigma(\mathbb{K}_y(\mathbf{x})),$$

$$\underline{k}_x := \inf_{\mathbf{y} \in \gamma} k_x(\mathbf{y}), \quad \bar{k}_x := \sup_{\mathbf{y} \in \gamma} k_x(\mathbf{y}).$$

The main result of this section is the following error estimate.

Theorem 1. *Let $\delta > 0$, and assume in addition that the unique solution to (3) satisfies*

$$\partial_x^2 p \in L^q(\Omega_f) \text{ for some } q \in [2, \infty].$$

Then there is a constant $C := C(\Omega, \gamma) > 0$ independent of δ , \mathbb{K} and f such that

$$\|\nabla_y(\bar{p} - p_f)\|_{2,\gamma} \leq C \sqrt{\frac{\bar{k}_x}{\underline{K}_y}} \|\partial_x^2 p\|_{q,\Omega_f} \delta^{1-\frac{1}{q}}, \quad (5a)$$

$$\|\nabla(p - p_m)\|_{2,\Omega_m} \leq C \sqrt{\frac{\bar{k}_x}{\underline{K}_m}} \|\partial_x^2 p\|_{q,\Omega_f} \delta^{\frac{3}{2}-\frac{1}{q}}, \quad (5b)$$

$$\sum_{i=1}^2 \|\bar{p} - p|_{\gamma_i} + p_m|_{\gamma_i} - p_f\|_{2,\gamma} \leq C \sqrt{\frac{\bar{k}_x}{\underline{k}_x}} \|\partial_x^2 p\|_{q,\Omega_f} \delta^{2-\frac{1}{q}}, \quad (5c)$$

where (p_m, p_f) is the solution of (4).

Proof. For any $\varepsilon \in (0, \delta/2)$ we define the sets

$$\Omega_{f\varepsilon} := \{(x, \mathbf{y}) \in \Omega; -\delta/2 + \varepsilon < x < \delta/2 - \varepsilon\},$$

$$\Omega_{f\varepsilon}^- := \{(x, \mathbf{y}) \in \Omega_f; x < -\delta/2 + \varepsilon\},$$

$$\Omega_{f\varepsilon}^+ := \{(x, \mathbf{y}) \in \Omega_f; x > \delta/2 - \varepsilon\}$$

and an auxiliary operator $\Pi_\varepsilon : L^2(\Omega_m) \times L^2(\gamma) \rightarrow L^2(\Omega)$:

$$\Pi_\varepsilon(v_m, v_\gamma)(x, \mathbf{y}) := \begin{cases} v_m(x, \mathbf{y}) & \text{in } \Omega_m, \\ v_\gamma(0, \mathbf{y}) & \text{in } \Omega_{f\varepsilon}, \\ \frac{1}{\varepsilon}(x + \frac{\delta}{2})v_\gamma(0, \mathbf{y}) - \frac{1}{\varepsilon}(x + \frac{\delta}{2} - \varepsilon)v_m(-\frac{\delta}{2}, \mathbf{y}) & \text{in } \Omega_{f\varepsilon}^-, \\ -\frac{1}{\varepsilon}(x - \frac{\delta}{2})v_\gamma(0, \mathbf{y}) + \frac{1}{\varepsilon}(x - \frac{\delta}{2} + \varepsilon)v_m(\frac{\delta}{2}, \mathbf{y}) & \text{in } \Omega_{f\varepsilon}^+. \end{cases}$$

Note that Π_ε maps $H_{bc}^1(\Omega_m) \times H_0^1(\gamma)$ into $H_0^1(\Omega)$. We use $v_\varepsilon := \Pi_\varepsilon(v_m, v_f)$, $v_m \in H_{bc}^1(\Omega_m)$, $v_f \in H_0^1(\gamma)$ as a test function in (3):

$$\begin{aligned} \int_{\Omega_m} \mathbb{K}_m \nabla p \cdot \nabla v_m + \int_{\Omega_{f\varepsilon}} \mathbb{K}_y \nabla_y p \cdot \nabla_y v_f + \int_{\Omega_f \setminus \Omega_{f\varepsilon}} k_x \partial_x p \partial_x v_\varepsilon \\ + \int_{\Omega_f \setminus \Omega_{f\varepsilon}} \mathbb{K}_y \nabla_y p \cdot \nabla_y v_\varepsilon = \int_{\Omega_m} f v_m + \int_{\Omega_{f\varepsilon}} f v_f + \int_{\Omega_f \setminus \Omega_{f\varepsilon}} f v_\varepsilon. \end{aligned} \quad (6)$$

Next we shall perform the limit $\varepsilon \rightarrow 0+$. Due to continuity of the integral we have:

$$\int_{\Omega_{f\varepsilon}} \mathbb{K}_y \nabla_y p \cdot \nabla_y v_f \rightarrow \int_{\Omega_f} \mathbb{K}_y \nabla_y p \cdot \nabla_y v_f = \delta \int_\gamma \mathbb{K}_y \nabla_y \bar{p} \cdot \nabla_y v_f, \quad (7)$$

$$\int_{\Omega_f \setminus \Omega_{f\varepsilon}} \mathbb{K}_y \nabla_y p \cdot \nabla_y v_\varepsilon \rightarrow 0, \quad (8)$$

$$\int_{\Omega_{f\varepsilon}} f v_f \rightarrow \int_{\Omega_f} f v_f = \delta \int_\gamma \bar{f} v_f, \quad (9)$$

$$\int_{\Omega_f \setminus \Omega_{f\varepsilon}} f v_\varepsilon \rightarrow 0, \quad \varepsilon \rightarrow 0+. \quad (10)$$

The remaining term can be rewritten as follows:

$$\begin{aligned} \int_{\Omega_f \setminus \Omega_{f\varepsilon}} k_x \partial_x p \partial_x v_\varepsilon &= \frac{1}{\varepsilon} \int_{\Omega_{f\varepsilon}^-} k_x \partial_x p (v_f - v_{m|\gamma_1}) - \frac{1}{\varepsilon} \int_{\Omega_{f\varepsilon}^+} k_x \partial_x p (v_f - v_{m|\gamma_2}) \\ &\rightarrow \sum_{i=1}^2 (-1)^{1+i} \int_\gamma k_x \partial_x p|_{\gamma_i} (v_f - v_{m|\gamma_i}), \quad \varepsilon \rightarrow 0+. \end{aligned} \quad (11)$$

Let $\mathbf{y} \in \mathbf{R}^{d-1}$ be fixed and define

$$P(x) := \frac{1}{\delta} \int_{-\delta/2}^x p(t, \mathbf{y}) dt.$$

Using the Taylor expansion

$$\begin{aligned} P(x) &= P(-\delta/2) + (x + \frac{\delta}{2}) P'(-\delta/2) + \frac{(x + \frac{\delta}{2})^2}{2} P''(-\delta/2) \\ &\quad + \frac{(x + \frac{\delta}{2})^2}{2} \int_{-\delta/2}^{\xi(x, \mathbf{y})} P'''(t) dt, \quad \xi(x, \mathbf{y}) \in (-\delta/2, x), \end{aligned} \quad (12)$$

we can show that

$$\bar{p}(\mathbf{y}) = P(\delta/2) = p(-\delta/2, \mathbf{y}) + \frac{\delta}{2} \partial_x p(-\delta/2, \mathbf{y}) + \frac{\delta}{2} \int_{-\delta/2}^{\xi(\delta/2, \mathbf{y})} \partial_x^2 p(t, \mathbf{y}) dt.$$

By a similar argument we obtain:

$$\bar{p}(\mathbf{y}) = p(\delta/2, \mathbf{y}) - \frac{\delta}{2} \partial_x p(-\delta/2, \mathbf{y}) + \frac{\delta}{2} \int_{\eta(-\delta/2, \mathbf{y})}^{\delta/2} \partial_x^2 p(t, \mathbf{y}) dt, \quad \eta(x, \mathbf{y}) \in (x, \delta/2).$$

From this we can deduce that

$$\partial_x p|_{\gamma_i} = (-1)^{1+i} \left(\frac{2}{\delta} (\bar{p} - p|_{\gamma_i}) - \delta g_i \right), \quad i = 1, 2, \quad (13)$$

where

$$|g_i(\mathbf{y})| \leq \frac{1}{\delta} \int_{-\delta/2}^{\delta/2} |\partial_x^2 p(\cdot, \mathbf{y})|. \quad (14)$$

Summing up, (6)–(13) yields:

$$\begin{aligned} & \int_{\Omega_m} \mathbb{K}_m \nabla p \cdot \nabla v_m + \delta \int_{\gamma} \mathbb{K}_{\mathbf{y}} \nabla_{\mathbf{y}} \bar{p} \cdot \nabla_{\mathbf{y}} v_f + \sum_{i=1}^2 \int_{\gamma_i} q_i(p, \bar{p})(v_m|_{\gamma_i} - v_f) \\ &= \int_{\Omega_m} f v_m + \delta \int_{\gamma} \bar{f} v_f + \delta \sum_{i=1}^2 \int_{\gamma} k_x g_i (v_m|_{\gamma_i} - v_f). \end{aligned} \quad (15)$$

Now we use $v_m := p - p_m$, $v_f := \bar{p} - p_f$ as test functions in (15) and (4), and subtract the resulting identities. We obtain:

$$\begin{aligned} & \int_{\Omega_m} \mathbb{K}_m \nabla(p - p_m) \cdot \nabla(p - p_m) + \delta \int_{\gamma} \mathbb{K}_{\mathbf{y}} \nabla_{\mathbf{y}}(\bar{p} - p_f) \cdot \nabla_{\mathbf{y}}(\bar{p} - p_f) \\ &+ \sum_{i=1}^2 \int_{\gamma} \frac{2k_x}{\delta} |p|_{\gamma_i} - p_m|_{\gamma_i} - \bar{p} + p_f|^2 = \delta \sum_{i=1}^2 \int_{\gamma} k_x g_i (p|_{\gamma_i} - p_m|_{\gamma_i} - \bar{p} + p_f). \end{aligned} \quad (16)$$

Using Hölder's and Young's inequality we can estimate the right hand side of (16):

$$\begin{aligned} & \delta \sum_{i=1}^2 \int_{\gamma} k_x g_i (p|_{\gamma_i} - p_m|_{\gamma_i} - \bar{p} + p_f) \\ & \leq \frac{\delta^{\frac{3}{2}}}{\sqrt{2}} \sum_{i=1}^2 \int_{\gamma} \sqrt{k_x} |g_i| \sqrt{\frac{2k_x}{\delta}} |p|_{\gamma_i} - p_m|_{\gamma_i} - \bar{p} + p_f| \\ & \leq \frac{\delta^3}{4} \bar{k}_x \sum_{i=1}^2 \|g_i\|_{2,\gamma}^2 + \frac{1}{2} \sum_{i=1}^2 \int_{\gamma} \frac{2k_x}{\delta} |p|_{\gamma_i} - p_m|_{\gamma_i} - \bar{p} + p_f|^2. \end{aligned} \quad (17)$$

From (14) and Hölder's inequality it follows that

$$\|g_i\|_{2,\gamma}^2 \leq \delta^{-\frac{2}{q}} |\gamma|^{\frac{q-2}{q}} \|\partial_x^2 p\|_{q,\Omega_f}^2. \quad (18)$$

Finally, (16), (17), (18) and the uniform positive definiteness of \mathbb{K} yields:

$$\begin{aligned} & \underline{K}_m \|\nabla(p - p_m)\|_{2,\Omega_m}^2 + \delta \underline{K}_{\mathbf{y}} \|\nabla_{\mathbf{y}}(\bar{p} - p_f)\|_{2,\gamma}^2 \\ &+ \frac{1}{\delta} \bar{k}_x \sum_{i=1}^2 \|\bar{p} - p|_{\gamma_i} + p_m|_{\gamma_i} - p_f\|_{2,\gamma}^2 \leq \frac{\bar{k}_x}{2} |\gamma|^{\frac{q-2}{q}} \|\partial_x^2 p\|_{q,\Omega_f}^2 \delta^{3-\frac{2}{q}}, \end{aligned} \quad (19)$$

from which the estimates (5) follow.

3 Numerical Experiments

In this section we present computational results that demonstrate the relevance of the continuum-fracture model in the discrete setting, in particular we study the dependence of the error between the full and the reduced model on the fracture thickness δ . For the numerical computation we used the mixed-hybrid FEM implemented in the code Flow123d [2].

For each δ , the solution to the continuum-fracture model computed on a sequence of meshes with different step h was compared either against the analytical solution of the full model or against a reference solution of the full model on a sufficiently fine mesh. This approach allows to distinguish the discretization error and the error of the reduced model.

3.1 Test 1 - Analytical Solution and Virtual Fracture

As the first, we consider a problem with the fixed constant conductivity $\mathbb{K} = \mathbb{I}$ admitting the exact solution $p^\delta(x, y) = e^x \sin y$ in the domain $\Omega := (-1, 1) \times (0, 1)$ with a virtual fracture $\Omega_f^\delta := (-\delta/2, \delta/2) \times (0, 1)$. In Fig. 2, we display the L^2 norm of the error in the pressure and in the velocity separately for the matrix and for the fracture domain. Numerical solution to the reduced model using $h \in \{0.04, 0.02, 0.01, 0.005, 0.0025\}$ is compared to the analytical solution. Estimated orders of convergence are consistent with the estimate (5), namely the convergence in the matrix domain is faster then in the fracture domain. All numerically estimated orders of convergence are higher then predicted by Theorem 1 since the solution is perfectly regular. Let us also note that $\|\partial_x^2 p^\delta\|_{\infty, \Omega_f}$, which stands on the right hand side of (5), is close to 1 for all δ . Unfortunately, the discretization error of the velocity on the matrix domain is only of the first order with respect to h which makes the numerical estimate of the order of convergence with respect to δ less precise.

3.2 Test 2 - Highly Permeable Fracture

In the second test we investigated more realistic case, with $\mathbb{K}_m = 1$ and $\mathbb{K}_f = 100$. We prescribed harmonic Dirichlet boundary condition $p^\delta(x, y) = \cos(x) \cosh(y)$ on the boundary of the domain $\Omega := (0, 2) \times (-0.5, 0.5)$ with the fracture along the line $x = 1$. Actually, we exploited symmetry of the problem, computing only on the top half and prescribing the homogeneous Neumann condition on the bottom boundary. In this case, no analytical solution is available, so for each δ we solved the full model on a mesh locally refined around the fracture and especially around its endpoint $[1, 0.5]$, where the solution exhibits singularities. The corresponding continuum-fracture model was solved on a sequence of meshes for $h \in \{0.2, 0.1, 0.05, 0.02, 0.01, 0.005, 0.0025\}$.

As in the previous case, Fig. 3 displays the L^2 -norm of the error in the pressure and in the velocity separately for the matrix and for the fracture domain. However, unlike in the previous case, the error is not consistent with the theoretical estimate. The error in the matrix domain is only of the order 1 with

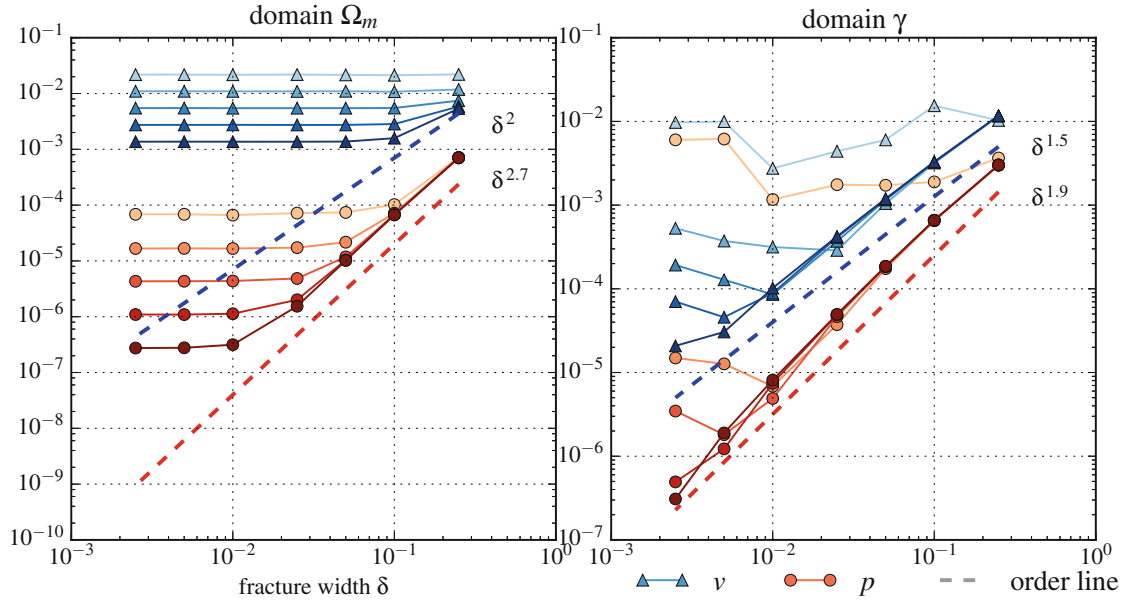


Fig. 2. Test 1 - virtual fracture $K_m = K_f = 1$, L^2 norm of the error in the pressure (p) and in the velocity (v), comparison against analytical solution. Each line connects values for the same h , darker colors correspond to smaller h .

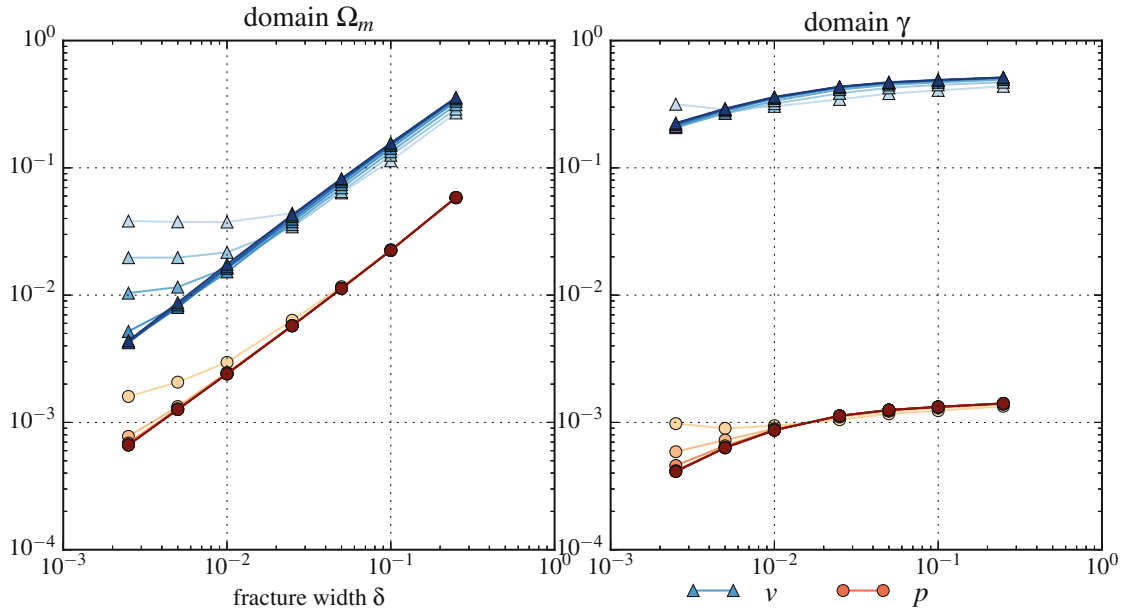


Fig. 3. Test 2 - permeable fracture, $K_m = 1$, $K_f = 100$, L^2 norm of the error in the pressure (p) and in the velocity (v), comparison against solution to the full model on refined mesh. Each line connects values for the same h , darker colors correspond to smaller h .

respect to δ and the error in the fracture domain stagnates. The main reason for this situation is the high and possibly unbounded value of the norm $\|\partial_x^2 p^\delta\|_{\infty, \Omega_f}$ as well as the norm $\|\partial_x^2 p^\delta\|_{2, \Omega_f}$, whose numerical estimates uniformly grow with decreasing h . On the other hand even if the norm of the second derivative is

not bounded, we observe relatively good approximation of the pressure in both domains and good approximation of the velocity in the matrix domain.

4 Conclusions

We analyzed theoretically and numerically the error of the continuum-fracture model for the Darcy flow in a domain containing a fracture. The obtained error rates are related to the regularity of the solution to the full model and are in agreement with the result of [7]. Unfortunately, in applications the solution exhibits singularities at the intersection of the fracture boundary with the domain boundary which may lead to inaccuracy in the continuum-fracture model. Refinement or dedicated model may be necessary in order to obtain descent flux error at the fracture boundary.

Acknowledgement. This work was supported by the Ministry of Education, Youth and Sports under the project LO1201 in the framework of the targeted support of the “National Programme for Sustainability I” and the OPR & DI project Centre for Nanomaterials, Advanced Technologies and Innovation CZ.1.05/2.1.00/01.0005. Computational resources were provided by the MetaCentrum under the program LM2010005 and the CERIT-SC under the program Centre CERIT Scientific Cloud, part of the Operational Program Research and Development for Innovations, Reg. no. CZ.1.05/3.2.00/08.0144.

References

1. Angot, P., Boyer, F., Hubert, F.: Asymptotic and numerical modelling of flows in fractured porous media. *ESAIM Math. Model. Numer. Anal.* **43**(02), 239–275 (2009)
2. Březina, J., Stebel, J., Exner, P., Flanderka, D.: Flow123d 2011–2015. <http://flow123d.github.com>
3. Frih, N., Martin, V., Roberts, J.E., Saâda, A.: Modeling fractures as interfaces with nonmatching grids. *Comput. Geosci.* **16**(4), 1043–1060 (2012)
4. Fumagalli, A., Scotti, A.: A reduced model for flow and transport in fractured porous media with non-matching grids. In: Cangiani, A., Davidchack, R.L., Georgoulis, E., Gorbun, A.N., Levesley, J., Tretyakov, M.V. (eds.) *Numerical Mathematics and Advanced Applications 2011*, pp. 499–507. Springer, Heidelberg (2013)
5. Ganis, B., Girault, V., Mear, M., Singh, G., Wheeler, M.: Modeling fractures in a poro-elastic medium. *Oil Gas Sci. Technol.-Rev. dIFP Energies nouvelles* **69**(4), 515–528 (2014)
6. Lesinigo, M., D’Angelo, C., Quarteroni, A.: A multiscale Darcy-Brinkman model for fluid flow in fractured porous media. *Numer. Math.* **117**(4), 717–752 (2011)
7. Martin, V., Jaffré, J., Roberts, J.E.: Modeling fractures and barriers as interfaces for flow in porous media. *SIAM J. Sci. Comput.* **26**(5), 1667 (2005)



Chapter 3

Numerical methods for conforming mixed meshes

In this chapter we present the Darcy flow mixed dimension model and its discretization using mixed-hybrid finite elements in particular RT_0 elements. We present the model and the discretization as it is implemented in the Flow123d simulator [21]. Section 3.1 describes the model at continuous level. The discretization is covered in Section 3.3. Two papers follows: first devoted to the derivation of the analytical solution to the test fracture problem, second dealing with application of BDDC method for scalable parallel solution of the stationary problems.

3.1 Darcy Flow Model

This section presents a formulation for steady and unsteady Darcy flow problems on a mixed dimension domain. We apply general theory from Chapter 2 and provide physical context and realistic boundary conditions.

Let us start with the simplest model for the velocity of the steady or unsteady flow in porous and fractured medium given by the Darcy flow:

$$\mathbf{w} = -\mathbb{K}\nabla H \quad \text{in } \Omega_d, \text{ for } d = 1, 2, 3. \quad (3.1)$$

Here and later on, we drop the dimension index d of the quantities if it can be deduced from the context. In (3.1), \mathbf{w} [ms^{-1}] is the superficial velocity, \mathbb{K}_d is the conductivity tensor, and H [m] is the piezometric head. The velocity \mathbf{w}_d is related to the flux \mathbf{q}_d [$\text{m}^{4-d}\text{s}^{-1}$] through

$$\mathbf{q}_d = \delta_d \mathbf{w}_d,$$

where δ_d [m^{3-d}] is the cross section coefficient, in particular $\delta_3 = 1$, δ_2 [m] is the thickness of a fracture, and δ_1 [m^2] is the cross-section of a channel. The flux $\mathbf{q}_d \cdot \mathbf{n}$ is the volume of the liquid (water) that passes through a unit square ($d = 3$), unit line ($d = 2$), or through a point ($d = 1$) per one second. The conductivity tensor is given by the product $\mathbb{K}_d = k_d \mathbb{A}_d$, where $k_d > 0$ [ms^{-1}] is the hydraulic conductivity and \mathbb{A}_d is the 3×3 dimensionless anisotropy tensor which has to be symmetric and positive definite. The piezometric-head H_d is related to the pressure head h_d through

$$H_d = h_d + z \quad (3.2)$$

assuming that the gravity force acts in the negative direction of the z -axis. Combining these relations, we get the Darcy law in the form:

$$\mathbf{q} = -\delta k \mathbb{A} \nabla (h + z) \quad \text{in } \Omega_d, \text{ for } d = 1, 2, 3. \quad (3.3)$$

Next, we employ the continuity equation for a saturated porous medium and apply abstract mixed dimension model from Chapter 2, we obtain

$$\partial_t(\delta_d S_d h_d) + \operatorname{div} \mathbf{q}_d = F_d \quad \text{in } \Omega_d, \text{ for } d = 1, 2, 3, \quad (3.4)$$

where $S_d [\text{m}^{-1}]$ is the storativity and $F_d [\text{m}^{3-d}\text{s}^{-1}]$ is the source term. In our setting the principal unknowns of the system (3.3, 3.4) are the pressure head h_d and the flux \mathbf{q}_d .

The storativity (or the volumetric specific storage) $S_d > 0$ can be expressed as

$$S_d = \gamma_w(\beta_r + \vartheta\beta_w), \quad (3.5)$$

where $\gamma_w [\text{kgm}^{-2}\text{s}^{-2}]$ is the specific weight of water, $\vartheta [-]$ is the porosity, β_r is compressibility of the bulk material of the pores (rock) and β_w is compressibility of the water, both with units $[\text{kg}^{-1}\text{ms}^{-2}]$. For steady problems, we set $S_d = 0$ for all dimensions $d = 1, 2, 3$. The source term F_d on the right hand side of (3.4) consists of the volume density of the water source $f_d [\text{s}^{-1}]$ and flux from the from the higher dimension. Precise form of F_d slightly differs for every dimension and will be discussed presently. In Ω_3 we simply have $F_3 = f_3 [\text{s}^{-1}]$.

For unsteady problems one has to specify an initial condition in terms of the initial pressure head $h_d^0 [\text{m}]$ or the initial piezometric head $H_d^0 [\text{m}]$.

3.1.1 Coupling on mixed meshes

In the set $\Omega_2 \cap \Omega_3$ the fracture is surrounded by at most one 3d surface from every side. On $\partial\Omega_3 \cap \Omega_2$ we prescribe a boundary condition of the Robin type:

$$\begin{aligned} \mathbf{q}_3 \cdot \mathbf{n}^+ &= q_{32}^+ = \sigma_3(h_3^+ - h_2), \\ \mathbf{q}_3 \cdot \mathbf{n}^- &= q_{32}^- = \sigma_3(h_3^- - h_2), \end{aligned} \quad (3.6)$$

where $\mathbf{q}_3 \cdot \mathbf{n}^{+/-} [\text{ms}^{-1}]$ is the outflow from Ω_3 , $h_3^{+/-}$ is a trace of the pressure head in Ω_3 , h_2 is the pressure head in Ω_2 , and $\sigma_3 [\text{s}^{-1}]$ is the transition coefficient, compare with (2.7 – 2.8), given by:

$$\sigma_3 = \sigma_{32} \frac{2\mathbb{K}_2 : \mathbf{n}_2 \otimes \mathbf{n}_2}{\delta_2}.$$

Here \mathbf{n}_2 is the unit normal to the fracture (sign does not matter). On the other hand, the sum of the interchange fluxes $q_{32}^{+/-}$ forms a volume source in Ω_2 . Therefore $F_2 [\text{ms}^{-1}]$ on the right hand side of (3.4) is given by

$$F_2 = \delta_2 f_2 + (q_{32}^+ + q_{32}^-). \quad (3.7)$$

The communication between Ω_2 and Ω_1 is similar. However, in the 3d ambient space, a 1d channel can join multiple 2d fractures $1, \dots, n$. Therefore, we have n independent outflows from Ω_2 :

$$\mathbf{q}_2 \cdot \mathbf{n}^i = q_{21}^i = \sigma_2(h_2^i - h_1),$$

where $\sigma_2 [\text{ms}^{-1}]$ is the transition coefficient integrated over the width of the fracture i :

$$\sigma_2 = \sigma_{21} \frac{2\delta_2^2 \mathbb{K}_1 : \mathbf{n}_1^i \otimes \mathbf{n}_1^i}{\delta_1}.$$

Here \mathbf{n}_1^i is the unit normal to the channel that is tangential to the fracture i . Sum of the fluxes forms a part of F_1 [m^2s^{-1}]:

$$F_1 = \delta_1 f_1 + \sum_{i=1}^n q_{21}^i. \quad (3.8)$$

We remark that the direct communication between 3d and 1d (e.g. model of a well) is not supported yet. The transition coefficients σ_{32} [–] and σ_{21} [–] are independent scaling parameters which represent the ratio of the crosswind and the tangential conductivity in the fracture. For example, in the case of impermeable film on the fracture walls one may choose $\sigma_{32} < 1$.

3.1.2 Boundary conditions

In order to obtain unique solution we have to prescribe boundary conditions. Currently we consider a disjoint decomposition of the boundary

$$\partial\Omega_d = \Gamma_d^D \cup \Gamma_d^{TF} \cup \Gamma_d^{Sp} \cup \Gamma_d^{Ri}$$

where we support the following types of boundary conditions:

Dirichlet boundary condition

$$h_d = h_d^D \text{ on } \Gamma_d^D,$$

where h_d^D [m] is the boundary pressure head. Alternatively one can prescribe the boundary piezometric head H_d^D [m] related to the pressure head through (3.2).

Total flux boundary condition (combination of Neumann and Robin type)

$$-\mathbf{q}_d \cdot \mathbf{n} = \delta_d (q_d^N + \sigma_d^R (h_d^R - h_d)) \text{ on } \Gamma_d^{TF},$$

where q_d^N [ms^{-1}] is the surface density of the water inflow, h_d^R [m] is the boundary pressure head and σ_d^R [s^{-1}] is the transition coefficient. As before one can also prescribe the boundary piezo head H_d^R to specify h_d^R .

Seepage face condition is used to model a surface with possible springs:

$$h_d \leq h_d^S \quad \text{and} \quad -\mathbf{q}_d \cdot \mathbf{n} \leq \delta_d q_d^N \quad (3.9)$$

while the equality holds in at least one inequality. The switch pressure head h_d^S [m] can alternatively be given by switch piezometric head.

The first inequality in (3.9) with the default value $h_d^S = 0$ disallows non-zero water height on the surface, the later inequality with default value $q_d^N = 0$ allows only outflow from the domain (i.e. spring). In practice one may want to allow given water height h_d^S or given infiltration (e.g. precipitation-evaporation) q_d^N .

River boundary condition models free water surface with bedrock of given conductivity. We prescribe:

$$-\mathbf{q}_d \cdot \mathbf{n} = \delta_d (\sigma_d^R (H_d - H_d^D) + q_d^N), \quad \text{for } H_d \geq H_d^S, \quad (3.10)$$

$$-\mathbf{q}_d \cdot \mathbf{n} = \delta_d (\sigma_d^R (H_d^S - H_d^D) + q_d^N), \quad \text{for } H_d < H_d^S, \quad (3.11)$$

where H_d is piezometric head. The parameters of the condition are given by similar fields of other boundary conditions: the transition coefficient of the bedrock σ_d^R [s^{-1}], the piezometric head of the water surface given as boundary piezometric head H_d^D [m], the head of the bottom of

the river given as the switch piezometric head H_d^S [m]. The boundary flux q_d^N is zero by default, but can be used to express approximation of the seepage face condition (see discussion below). The piezometric heads H_d^S and H_d^R may be alternatively given by pressure heads h_d^S and h_d^R , respectively.

The physical interpretation of the condition is as follows. For the water level H_d above the bottom of the river H_d^S the infiltration is given as Robin boundary condition with respect to the surface of the river H_d^D . For the water level below the bottom the infiltration is given by the water column of the river and transition coefficient of the bedrock.

The river could be used to approximate the seepage face condition in the similar way as the Robin boundary condition with large σ can approximate Dirichlet boundary condition. We rewrite the condition as follows

$$-\mathbf{q}_d \cdot \mathbf{n} = \delta_d (\sigma_d^R (h_d - h_d^D) + q_d^N), \quad \text{for } -\mathbf{q}_d \cdot \mathbf{n} \geq \delta_d (\sigma_d^R (h_d^S - h_d^D) + q_d^N), \quad (3.12)$$

$$-\mathbf{q}_d \cdot \mathbf{n} = \delta_d (\sigma_d^R (h_d^S - h_d^D) + q_d^N), \quad \text{for } h_d < h_d^S. \quad (3.13)$$

Now if we take $h_d^S = h_d^D$, we obtain

$$-\mathbf{q}_d \cdot \mathbf{n} = \delta_d (\sigma_d^R (h_d - h_d^S) + q_d^N), \quad \text{for } -\mathbf{q}_d \cdot \mathbf{n} \geq \delta_d q_d^N, \quad (3.14)$$

$$-\mathbf{q}_d \cdot \mathbf{n} = \delta_d q_d^N, \quad \text{for } h_d < h_d^S, \quad (3.15)$$

where the first equation approximates $h_d = h_d^S$ if σ_d^R is sufficiently large.

3.1.3 Water balance

The equation (3.4) represents a conservation law for the volume of the liquid. In particular integrating over Ω_d for every $d = 1, 2, 3$, we obtain balance of the total liquid volume V :

$$V(t) = V(0) + \int_0^t s(\tau) d\tau + \int_0^t f(\tau) d\tau,$$

for any time point t in the computational interval $[0, T]$. Other variables are: the total liquid volume $[m^3]$,

$$V(t) := \sum_{d=1}^3 \int_{\Omega^d} (\delta S h)(t, \mathbf{x}) d\mathbf{x},$$

the total of volume sources $[m^3 s^{-1}]$ in time t ,

$$s(t) := \sum_{d=1}^3 \int_{\Omega^d} F(t, \mathbf{x}) d\mathbf{x},$$

and the boundary flux $[m^3 s^{-1}]$ of the liquid at time t ,

$$f(t) := - \sum_{d=1}^3 \int_{\partial\Omega^d} \mathbf{q}(t, \mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) d\mathbf{x}.$$

Flow123d simulator can compute the volume, the source total and the on every geometrical region for every output time. In addition the cumulative flux and source can be provided. These information are essential for usage for real world problems.

3.2 Mixed formulations with conforming fractures

Discretization of the mixed dimension Darcy flow problem in Flow123d use mixed-hybrid formulation using zero order Raviart-Thomas elements. The formulation we shall describe follows and extends ideas and results from the previous works [6], [10], [11], [19] from the origin of the Flow123d simulator.

3.2.1 Mixed formulations on mixed meshed

The unsteady Darcy flow problem introduced in Section 3.1 allows straight forward mixed-hybrid formulation. We shall present mixed-hybrid formulation to a steady problem as a single time step of the problem already discretized in time.

Let $\mathcal{P}_d = \{\Omega_d^i\}$, $i \in I_d$, $d = 2, 3$ be a decomposition of Ω_d into disjoint open sets such that lower dimension domain is on its boundary:

$$\Gamma_d = \bigcup_{i \in I_d} \partial\Omega_d^i = \partial\Omega_d \cup \Omega_{d-1} \cup \Gamma_d^I, \quad (3.16)$$

where the decomposition on the right hand side contains outer boundary, fracture internal interfaces and proper internal interfaces, respectively.

Further we denote

$$\Omega_d^u = \bigcup_{i \in I_d} \Omega_d^i$$

Assuming that all indices are unique, we denote a common index set $I = I_1 \cup I_2 \cup I_3$. We introduce the velocity spaces:

$$\mathbf{V}_d = \prod_{i \in I_d} \mathbf{H}(\text{div}, \Omega_d^i), \quad \mathbf{V} = \mathbf{V}_1 \times \mathbf{V}_2 \times \mathbf{V}_3. \quad (3.17)$$

then the pressure spaces:

$$Q_d = L^2(\Omega_d), \quad Q = Q_1 \times Q_2 \times Q_3. \quad (3.18)$$

the pressure trace spaces:

$$\dot{Q}_d = \{\dot{q} \in H^{1/2}(\Gamma_d) : \dot{q} = 0 \text{ on } \Gamma_d^D\}, \quad \dot{Q} = \dot{Q}_1 \times \dot{Q}_2 \times \dot{Q}_3. \quad (3.19)$$

and a common pressure spaces:

$$\overline{Q}_d = Q_d \times \dot{Q}_d, \quad \overline{Q} = Q \times \dot{Q}. \quad (3.20)$$

For the components of $\mathbf{u} \in \mathbf{V}$ and $p \in \overline{Q}$, we shall use notation $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)$ and $p = (p_1, p_2, p_3, \dot{p}_1, \dot{p}_2, \dot{p}_3)$ respectively. On these spaces we shall define mixed-hybrid solution similarly as in [6] or [7], but using the language of the book [4] by Brezzi and Fortin.

Definition 3.2.1. We say that the pair $(\mathbf{u}, p) \in \mathbf{V} \times \overline{Q}$ is a mixed-hybrid solution of the single time step problem $P(t, \mathcal{P})$ at the time t on the partitioning \mathcal{P} if it satisfies a saddle point problem

$$a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = \langle G, \mathbf{v} \rangle \quad \forall \mathbf{v} \in \mathbf{V}, \quad (3.21)$$

$$b(\mathbf{u}, q) - c(p, q) = \langle F, q \rangle \quad \forall q \in Q, \quad (3.22)$$

where the bilinear forms on the left hand side are

$$a(\mathbf{u}, \mathbf{v}) = \sum_{d=1}^3 \sum_{i \in I_d} \int_{\Omega_d^i} \frac{1}{\delta_d} \mathbb{K}_d^{-1} \mathbf{u}_d \cdot \mathbf{v}_d dx, \quad (3.23)$$

$$b(\mathbf{u}, q) = - \sum_{d=1}^3 \sum_{i \in I_d} \int_{\Omega_d^i} q_d \operatorname{div} \mathbf{u}_d dx + \sum_{d=1}^3 \sum_{i \in I_d} \int_{\partial \Omega_d^i} \hat{q}(\mathbf{u}_d \cdot \mathbf{n}) ds, \quad (3.24)$$

and the composed term c :

$$c(\bar{p}, \bar{q}) = c_f(\bar{p}, \bar{q}) + c_t(\bar{p}, \bar{q}) + c_R(\hat{p}, \hat{q}) \quad (3.25)$$

$$c_f(\bar{p}, \bar{q}) = \sum_{d=2,3} \sum_{i \in I_d} \int_{\Omega_{d-1}} \sigma_d(p_{d-1} - \hat{p}|_{T_i})(q_{d-1} - \hat{q}|_{T_i}) ds$$

$$c_t(\bar{p}, \bar{q}) = \sum_{d=1}^3 \sum_{i \in I_d} \int_{\Omega_d^i} \frac{\delta_d S_d}{\tau} p_d q_d dx,$$

$$c_R(\hat{h}, \hat{q}) = \sum_{d=1}^3 \sum_{i \in I_d} \int_{\Gamma_d^{TF}} \sigma_d^R p_d \hat{q}_d ds,$$

where c_f , c_t , c_R arise from fracture coupling, time term and the Robin boundary condition respectively. The linear functionals on the right-hand side have the form

$$\langle G, \mathbf{v} \rangle = - \sum_{d=1}^3 \sum_{i \in I_d} \int_{\partial \Omega_d} p_d^D (\mathbf{v} \cdot \mathbf{n}) ds, \quad (3.26)$$

$$\begin{aligned} \langle F, q \rangle = & - \sum_{d=1}^3 \int_{\Omega_d} \delta_d f_d q_d dx, \\ & - \sum_{d=1}^3 \sum_{T \in \mathcal{T}_d} \int_{\Gamma_d^{TF}} q_d^N \hat{q}_d + \sigma_d^R h_d^R \hat{q}_d ds \\ & - c_t(\bar{p}_{-1}, \bar{q}). \end{aligned} \quad (3.27)$$

where \bar{p}_{-1} is the pressure at previous time level and $p_d^D \in H^{1/2}(\Gamma_d)$ is an extension of the Dirichlet condition $p_d^D \in H^{1/2}(\Gamma_d^D)$.

Few remarks:

- The pressure trace \hat{p} is from the space \hat{Q} in particular it is zero on the Dirichlet boundary. The full pressure trace is therefore $\hat{p}_d + \tilde{P}_d$.
- We consider the Neumann boundary flux q_d^N positive for an inflow in the contrast to usual notation but consistently with the sign of the source f_d .
- We need trace \hat{p}_2 on Ω_1 in the definition of the bilinear form $c_f(p, q)$, which is available because of the compatibility condition (3.16).
- For every fracture triangular element $K \in \mathcal{T}_2$ we have two terms in c_f and two traces \hat{p}_2 one for each of two neighboring tetrahedral elements. For a segment elements $K \in \mathcal{T}_1$ the number of the terms can be even larger (for 3d ambient space), according to the number of neighboring triangular elements.

It can be shown using the general theory of mixed formulations [4] and same arguments as in Theorem 1 from the paper [22] reproduced in Section 3.5

3.3 Mixed-hybrid finite element method

Spatial discretization of the mixed-hybrid formulation from Definition 3.2.1 is straight forward replacement of the function spaces by their finite dimensional subspaces. We assume a mesh \mathcal{T} and submeshes \mathcal{T}_d as introduced in Section 2.2. Let us denote $\mathbf{V}_d^h(T_d) \subset \mathbf{H}(\text{div}, T_d)$ the space of Raviart-Thomas functions of zero order (RT_0) on an element $T_d \in \mathcal{T}_d$. Identifying the subdomains Ω_d^i with T_d^i we can choose following approximation of the function spaces \mathbf{V} and \mathbf{Q} . We set

$$\mathbf{V}^h = \mathbf{V}_1^h \times \mathbf{V}_2^h \times \mathbf{V}_3^h, \quad \mathbf{V}_d^h = \prod_{T_d \in \mathcal{T}_d} \mathbf{V}_d^h(T_d).$$

For the pressure space we use piecewise functions:

$$\mathbf{Q}^h = Q_1^h \times Q_2^h \times Q_3^h, \quad Q_d^h = \prod_{T \in \mathcal{T}_d} P^0(T) \subset Q_d. \quad (3.28)$$

Approximation is a bit more elaborated as we need a single degree of freedom (DOF) per interior edge but two DOFs per edge for fractures and possibly more DOFs per edge for channels. We begin with the local spaces for every element $T_d \in \mathcal{T}$:

$$\dot{\mathbf{Q}}^h(T_d) = \left\{ \dot{q} \in L^2(\partial T_d) : \dot{q} = \mathbf{v} \cdot \mathbf{n}|_{\partial T_d}, \mathbf{v} \in \mathbf{V}_d^h \right\}. \quad (3.29)$$

Due to properties of the RT_0 functions this is equivalent to piecewise constant functions per side of the element T_d . For RT_0 elements this gives a constant function for every element face. Further we identify DOFs on the proper interior faces Γ_d^I :

$$\dot{\mathbf{Q}}_d^h = \left\{ \dot{q} \in \prod_{T \in \mathcal{T}_d} \dot{\mathbf{Q}}^h(T) : \dot{q}|_{\partial K} = \dot{q}|_{\partial L} \text{ on } \partial K \cap \partial L \subset \Gamma_d^I \forall K, L \in \mathcal{T}_d \right\}. \quad (3.30)$$

Finally we set $\dot{\mathbf{Q}}^h = \dot{\mathbf{Q}}_1^h \times \dot{\mathbf{Q}}_2^h \times \dot{\mathbf{Q}}_3^h$. Note that the space $\dot{\mathbf{Q}}^h$ do not conform to the space $\dot{\mathbf{Q}}$ as it is less regular. However this is compensated by the higher regularity of \mathbf{V}^h .

Rest of the formulation is the same. We are looking for a triplet (\mathbf{u}, p, \dot{p}) from $\mathbf{V}^h \times \mathbf{Q}^h \times \dot{\mathbf{Q}}^h$ which satisfies the saddle-point problem:

$$a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = \langle g, \mathbf{v} \rangle, \quad \forall \mathbf{v} \in \mathbf{V}^h, \quad (3.31)$$

$$b(\mathbf{u}, q) - c(p, \dot{p}, q, \dot{q}) = \langle f, (q, \dot{q}) \rangle, \quad \forall q \in \mathbf{Q}, \dot{q} \in \dot{\mathbf{Q}}, \quad (3.32)$$

with the same forms as in Definition 3.2.1.

The submatrix A of the resulting linear system S which corresponds to the form $a(\cdot, \cdot)$ is blockwise diagonal with small blocks corresponding to individual elements. It can be cheaply inverted so we can form the Schur complement $S \setminus A$. The (q, q) submatrix A_1 of the first Schur complement is again diagonal allowing to form the second Schur complement $S \setminus A \setminus A_1$ with unknowns only from $\dot{\mathbf{Q}}$. Second Schur complement is negative definite and can be effectively solved using a CG method with ICC preconditioner. For large problems one can use balanced domain decomposition by constraints where Schur complements are formed locally per every processor. This is described in Section 3.5

3.4 Test problem with analytical solution

This section presents a preprint of a paper where we derive an analytical solution to a test Darcy flow problem on a square domain with a single horizontal fracture. This analytical solution is used as part of the test suite of Flow123d and is used to test optimal convergence rate of the implementation. It is used to test implementation of the formulation on conforming mixed meshes from the previous section as well as various coupling methods for non-conforming mixed meshes which will be discussed in Chapter 4.



Analytical solution to a single fracture test problem

Jan Březina ^{*}; Pavel Burda [†]

April 13, 2018

Abstract

An analytical solution to a test Darcy flow problem on a square domain with a single horizontal fracture is derived in terms of Fourier series. Distinct pressure variables p_1 and p_2 are considered for the fracture and the matrix respectively coupled together by a Robin type condition. Two cases are treated: the conductive fracture with p_2 continuous and symmetric about the fracture and the barrier fracture, generalization with separate p_2 and parameters for upper and lower part. The analytical solution was verified against the numerical solution using second order finite differences.

1 Introduction

Fractures, cracks, fissures, faults and other discontinuities are ubiquitous in real rock formations especially granitoids. Fractures present a challenge for subsurface water flow modeling as they have small volume but large permeability with potentially large impact on the pressure and velocity fields. Alternatively the filled fractures can behave as barriers.

Fractures occurs on wide range of scales. Small scale fractures could be homogenized, while fractures with scales comparable to the dimensions of a domain should be captured explicitly. The large scale fractures are still very thin and a direct discretization with any mesh based method (FE, FV, DG, ...) requires high level of refinement which become costly especially for large number of fractures. A solution to this problem is usage of a *mixed mesh* combining the elements of different dimension. It has been shown that for the advection-diffusion equation the original 3d (or 2d) problem can be approximated by the system of 3d-2d (or 2d-1d) problems coupled by a Robin type condition (see e.g. [5], [4], [1]).

For the purpose of testing implementations of the mixed mesh approach in combination with different numerical methods it is necessary to have a suitable test problem with known solution. A common approach is to plug any function into the equations and prescribe resulting right hand side. This approach is convenient to test correctness of the implementation but often turns out to do

^{*}Technical University in Liberec, Studentská 2, Liberec, Czech Republic (jan.brezina@tul.cz).

[†]Czech Technical University, (pavel.burda@fs.cvut.cz)



not capture peculiarities of the real solution and thus may be unsuitable for tests of convergence of the used method.

To this end we propose a test Darcy flow problem consisting of the square domain (matrix) and the single horizontal fracture cutting the domain into upper and lower part. Two cases are considered: the *conductive fracture* and the more general *barrier fracture*. The conductive fracture assumes symmetry about the fracture yet keeping separate pressure for the matrix and for the fracture. This case is relatively simple to solve but have very limited practical usage. The barrier fracture case is a generalization where the problem parameters as well as the matrix pressure are treated independently for the upper and the lower part of the matrix domain. Both problems are introduced in Section 2. The solution to the conductive fracture problem is derived in Section 3 the same approach with necessary modifications and extensions is used to derive the solution to the barrier fracture problem in Section 4. In Section 6, both solutions are verified against a numerical solution provided by the second order finite difference scheme on a regular grid. Conclusions are summarized in the final Section 7

2 Test problems

Main result of the paper is derivation of the strong analytical solution for two test problems with coupling between continuum and a fracture. A Darcy flow is considered on a square 2D domain $\Omega_2 = (-1, 1) \times (-1, 1)$ with a horizontal fracture $\Omega_1 = (x, 0) : x \in (-1, 1)$. The fracture splits Ω_2 into upper and lower part Ω_2^+ and Ω_2^- , respectively. The stationary Darcy flow is driven by the same equation on all three domains:

$$-k_2 \Delta p_2^+(x, y) = 0 \quad \text{on } \Omega_2^+, \quad (1)$$

$$-k_2 \Delta p_2^-(x, y) = 0 \quad \text{on } \Omega_2^-, \quad (2)$$

$$-k_1 p_1''(x) = f(x) \quad \text{on } \Omega_1. \quad (3)$$

Where p_d , $d = 1, 2$ is the pressure and f is the communication term that will be specified later. We consider positive constant conductivities k_2 , k_1 on Ω_2 and Ω_1 , respectively.

The homogeneous Neumann condition is set on the left and the right side of Ω_2 , while the Dirichlet condition is set on the top and bottom and at the tips of the fracture. We denote:

$$\Gamma_2^N = \{(x, y) : x \in \{-1, 1\}, y \in (-1, 1), y \neq 0\},$$

$$\Gamma_2^D = \{(x, y) : x \in (-1, 1), y \in \{-1, 1\}\},$$

$$\Gamma_1 = \partial\Omega_1 = \{(-1, 0), (1, 0)\}.$$

Then we prescribe following boundary conditions

$$\partial_x p_2(x, y) = 0 \quad \text{on } \Gamma_2^N \quad (4)$$

$$p_2(x, y) = P_2 \quad \text{on } \Gamma_2^D \quad (5)$$

$$p_1(x) = P_1 \quad \text{on } \Gamma_1. \quad (6)$$

In order to complete the problem we must prescribe boundary conditions on the fracture and specify the source term f . Here we distinguish two cases: the *conductive fracture* and the *barrier fracture*.

Conductive fracture. In this case we assume a fracture with similar or higher conductivity then in the continuum, $k_1 \geq k_2$. In such case we can assume continuity of the pressure across the fracture. However we keep difference between p_2 and p_1 on the fracture. We set:

$$p_2^+ = p_2^- \quad \text{on } \Omega_1, \quad (7)$$

$$-k_2(-\partial_y p_2^+ + \partial_y p_2^-) = f(x) \quad \text{on } \Omega_1, \quad (8)$$

$$f(x) = 2\sigma(p_2(x, 0) - p_1(x)), \quad (9)$$

where $\sigma \geq 0$ is a coupling parameter, usually $\sigma \approx k_1/\delta$ with δ standing for the fracture cross-section. Solution of this case is discussed in Section 3.

Barrier fracture. Other case is a fracture with significantly smaller conductivity compared to the surrounding continuum. In this case the pressure p_2 is discontinuous across the fracture and we have two independent boundary conditions for each side of the fracture. We also distinguish conductivities k_2^+ , k_2^- and boundary pressure P_2^+ , P_2^- for the upper and lower domains Ω_2^+ , Ω_2^- , respectively. The coupling on the fracture is prescribed by the boundary conditions:

$$-k_2^+ \nabla p_2^+ \cdot \mathbf{n}^+(x, 0) = k_2^+ \partial_y p_2^+(x) = f^+(x) \quad \text{on } \Gamma^+, \quad (10)$$

$$-k_2^- \nabla p_2^- \cdot \mathbf{n}^-(x, 0) = -k_2^- \partial_y p_2^-(x) = f^-(x) \quad \text{on } \Gamma^-, \quad (11)$$

with Γ^+ and Γ^- denoting boundary of Ω_2^+ and Ω_2^- , respectively, collocated with Ω_1 . The communication term is:

$$f(x) = f^+(x) + f^-(x), \quad f^{+/-} = \sigma^{+/-}(p_2^{+/-}(x, 0) - p_1(x)), \quad (12)$$

where σ^+ and σ^- are positive coupling parameters for upper and lower side of the fracture respectively. Solution to this system is discussed in Section 4.

3 Conductive fracture

We shall derive an analytical solution to the system (1 – 6) with the coupling conditions (7 – 9). Symmetry of the problem in both x and y direction allows us to solve equivalent reduced problem on $\tilde{\Omega}_2 = (0, 1) \times (0, 1)$ and $\tilde{\Omega}_1$. We impose the symmetry in x direction by homogeneous Neumann condition at $x = 0$. All equations are preserved with exception of the half flux on Γ^+ . The equivalent system reads:

$$-k_2 \Delta p_2(x, y) = 0 \quad \text{on } \tilde{\Omega}_2 \quad (13)$$

$$-k_1 p_1''(x) = f(x) \quad \text{on } \tilde{\Omega}_1 \quad (14)$$

with boundary conditions:

$$p_2(x, 1) = P_2, \quad k_2 \partial_y p_2(x, 0) = \frac{f(x)}{2} = \sigma(p_2(x, 0) - p_1(x)) \quad (15)$$

for $x \in (0, 1)$,

$$\partial_x p_2(0, y) = \partial_x p_2(1, y) = 0 \quad (16)$$

for $y \in (0, 1)$ and finally

$$p_1'(0) = 0, \quad p_1(1) = P_1. \quad (17)$$

Proposition 1 *Let the real parameters $P_2, P_1, k_1 > 0, k_2 > 0, \sigma \geq 0$ be given. Then the solution $p_1(x)$ and $p_2(x, y)$ to the system (13–17) can be expressed in form of Fourier series:*

$$p_2(x, y) = P_2 + B_0(y - 1) - 2B_0 \sum_{n=1}^{\infty} a_n \cos(\pi n x) \sinh(\pi n(1 - y)) \quad , \quad (18)$$

$$p_1(x) = P_2 - B_0 + u_0 \cosh(x/k) - 2B_0 \sum_{n=1}^{\infty} u_n \cos(\pi n x) \quad (19)$$

where we denote:

$$k = \sqrt{\frac{k_1}{2\sigma}},$$

and use following coefficients:

$$a_n = \frac{(-1)^n k_2}{k_2 \pi n \cosh(\pi n) (1 + (kn\pi)^2) + \sigma (kn\pi)^2 \sinh(\pi n)} \quad , \quad (20)$$

$$u_0 = -\frac{k_2 B_0}{\sigma k \sinh(1/k)}. \quad (21)$$

$$u_n = \frac{a_n \sinh(n\pi)}{1 + (kn\pi)^2} \quad , \quad (22)$$

and constants:

$$B_0 = \frac{P_2 - P_1}{1 + 2U + \frac{k_2}{\sigma k} \frac{\cosh(1/k)}{\sinh(1/k)}} \quad (23)$$

$$U = \sum_{n=1}^{\infty} (-1)^n u_n. \quad (24)$$

3.1 Separation of variables for 2D equation

We shall apply the separation of variables (see e.g. [3]) to the equation (13).

Considering a solution in form:

$$p_2(x, y) = X(x)Y(y) \quad (25)$$

the equation (13) gives us two equations:

$$\frac{X''}{X} = -\frac{Y''}{Y} = L \quad ,$$

L being a real constant. Applying homogeneous Neumann boundary conditions (16) we get possible solutions $X(x)$ in the form:

$$X_n(x) = \tilde{A}_n + \tilde{B}_n \cos(n\pi x) \quad \text{for } n = 0, 1, \dots \quad (26)$$

where \tilde{A}_n, \tilde{B}_n are arbitrary real constants and L is quantized to values:

$$L = -n^2\pi^2.$$

Next we plug L in the equation for $Y(y)$ to get all solutions:

$$\begin{aligned} Y_0(y) &= \tilde{C}_0 + \tilde{D}_0 y, \\ Y_n(y) &= \tilde{C}_n e^{n\pi y} + \tilde{D}_n e^{-n\pi y} \quad \text{for } n = 1, \dots \end{aligned} \quad (27)$$

where \tilde{A}_n, \tilde{B}_n for $n = 0, 1, \dots$ are arbitrary real constants.

Combining (25), (26), (27) we can write the solution p_2 as:

$$p_2(x, y) = A_0 + B_0 y + \sum_{n=1}^{\infty} (C_n + \cos(n\pi x)) \frac{1}{2} (A_n e^{n\pi(y-1)} + B_n e^{-n\pi(y-1)}). \quad (28)$$

Then the Dirichlet condition (15a) yields:

$$P_2 = A_0 + B_0 + \sum_{n=1}^{\infty} (C_n + \cos(n\pi x)) \frac{A_n + B_n}{2}.$$

for all $x \in (0, 1)$ and thus:

$$A_0 + B_0 = P_2, \quad \text{and} \quad A_n + B_n = 0.$$

Then (28) can be simplified to the final form (18) where we yet have to determine B_0 and coefficients a_n .

3.2 Pressure on fracture

Next step is a solution to the equation on fracture, i.e. (14). Substituting for f using (15) and then for p_2 from (18) we arrive at:

$$-k^2 p_1''(x) + p_1(x) = P_2 - B_0 - 2B_0 \sum_{n=1}^{\infty} a_n \sinh(n\pi) \cos(n\pi x). \quad (29)$$

with $k = \sqrt{k_1/(2\sigma)}$.

Solution to:

$$-aP''(x) + P(x) = \cos(n\pi x)$$

is

$$P(x) = \frac{\cos(n\pi x)}{1 + an^2\pi^2}.$$

Using linearity of the equation we obtain the general form of p_1 as:

$$p_1(x) = c^+ e^{x/k} + c^- e^{-x/k} + P_2 - B_0 - 2B_0 \sum_{n=1}^{\infty} u_n \cos(n\pi x). \quad (30)$$

with u_n given by (22). Then the boundary conditions (17) yield:

$$\begin{aligned} P_1 &= p_1(1) = c^+ e^{1/k} + c^- e^{-1/k} + P_2 - B_0 - 2B_0 U, \\ 0 &= p_1'(0) = \frac{1}{k}(c^+ - c^-) \end{aligned}$$

with U given by (24). Solving for c^+ and c^- we get:

$$c^+ = c^- = \frac{P_1 - P_2 + B_0(1 + 2U)}{2 \cosh(1/k)}.$$

Then (30) gives

$$p_1(x) = P_2 - B_0 + u_0 \cosh(x/k) - 2B_0 \sum_{n=1}^{\infty} u_n \cos(\pi n x) \quad (31)$$

which is (19) but with

$$u_0 = \frac{P_1 - P_2 + B_0 + 2B_0 U}{\cosh(1/k)}. \quad (32)$$

3.3 Fracture coupling

Last relation that we have to consider is the boundary condition (15b). To this end we plug in the relations (18), (31) for p_2 , p_1 and we group the terms for remaining unknowns on the left hand side. After straight forward manipulation we obtain:

$$\frac{\mathcal{A}_0}{2} + \sum_{n=1}^{\infty} \mathcal{A}_n \cos(n\pi x) = -u_0 \cosh(x/k). \quad (33)$$

with

$$\mathcal{A}_0 = \frac{2k_2 B_0}{\sigma} \quad (34)$$

$$\mathcal{A}_n = 2B_0 \left[\frac{k_2}{\sigma} n\pi \cosh(n\pi) - \frac{\sinh(n\pi)}{1 + (kn\pi)^2} + \sinh(n\pi) \right] a_n \quad (35)$$

for $n = 1, \dots$. The left hand side of (33) is the Fourier series, thus we determine remaining unknowns B_0 , A_n by computing Fourier of the function on the right hand side and comparing coefficients.

For the zero term we have:

$$\mathcal{A}_0 = -2u_0 \int_0^1 \cosh(x/k) dx = -2u_0 k \sinh(1/k). \quad (36)$$

which compared to (34) gives us (21).

For other terms, we integrate by parts to get:

$$\mathcal{A}_n = -2u_0 \int_0^1 \cosh(x/k) \cos(n\pi x) dx = -\frac{(-1)^n 2u_0 k}{1 + (kn\pi)^2} \sinh(1/k). \quad (37)$$

We plug (21) into the result and compare it with (35) to obtain formula (20) for coefficients a_n . The only remaining unknown is B_0 which we determine by comparing (32) and (21). Straight forward calculation leads to (23).

3.4 Evaluation procedure

Formulas in Proposition 1 are not suitable for practical calculations since hypergeometric functions are evaluated for large n . To remedy this issue, we factor out and cancel $e^{\pi n}$. Actual evaluation then takes following procedure:

1. Set finite N for truncated series.
2. Compute \tilde{a}_n for $n = 1, \dots, N$ using:

$$\tilde{a}_n = \frac{(-1)^n k_2}{k_2 \pi n (1 - e^{-2\pi n}) (1 + (kn\pi)^2) + \sigma (kn\pi)^2 (1 + e^{-2\pi n})},$$

3. Compute related \tilde{u}_n coefficients:

$$\tilde{u}_n = \frac{\tilde{a}_n (1 - e^{-2\pi n})}{1 + (kn\pi)^2},$$

4. Evaluate truncated sum:

$$U = \sum_{n=1}^N (-1)^n u_n.$$

5. Compute remaining parameters u_0, B_0 using (21), (23).
6. Evaluate p_2 for any given point (x, y) in Ω_2 by:

$$p_2(x, y) = P_2 + B_0(y - 1) - 2B_0 \sum_{n=1}^N \tilde{a}_n \cos(\pi n x) \left(e^{-\pi n |y|} - e^{\pi n (|y| - 2)} \right).$$

7. Evaluate p_1 for any given x in Ω_1 by:

$$p_1(x) = P_2 - B_0 + u_0 \cosh(x/k) - 2B_0 \sum_{n=1}^N u_n \cos(\pi n x).$$

Convergence rates of the sums are reasonable. For a_n we have:

$$\tilde{a}_n \approx (-1)^n n^{-3}$$

and for two consecutive terms:

$$\tilde{a}_n + \tilde{a}_{n+1} \approx n^{-3} - (n+1)^{-3} \approx n^{-4}.$$

Series for p_2 converge exponentially for $y \neq 0$. Error for $y = 0$ is of order N^{-3} with exception of $x = 1$ where $\cos(\pi n)$ cancels alternation in a_n and error is N^{-2} .

Even better is convergence of u_n sums, since:

$$u_n \approx (-1)^n n^{-5}, \quad u_n + u_{n+1} \approx n^{-6}.$$

Thus for calculation of p_1 we have error N^{-5} (for $x = 1$ only N^{-4}). The error of U is only N^{-4} as the sign alternation cancels with alternation of a_n . Practical behavior of errors follows these estimates.

4 Barrier fracture

The case with discontinuous p_2 across the fracture is solved by the same approach as the continuous case, but the derivation is slightly more technical. Considering the y axis symmetry of the problem (1 – 6) and the coupling (10 – 12), we can consider the same system of equations on reduced domains $\Omega_2^+ = (0, 1) \times (0, 1)$, $\Omega_2^- = (0, 1) \times (-1, 0)$, $\Omega_1 = (0, 1)$. To impose the symmetry we consider homogeneous Neumann boundary condition on the y axis $\{(0, y)\}$.

Repeating arguments from Section 3.1 we obtain Fourier expansion for p_2^+ and p_2^- similar to (18):

$$p_2^+(x, y) = P_2^+ + B_0^+(y - 1) - \bar{u}_0 \sum_{n=1}^{\infty} a_n^+ \cos(\pi n x) \sinh(\pi n (1 - y)), \quad (38)$$

$$p_2^-(x, -y) = P_2^- + B_0^-(y - 1) - \bar{u}_0 \sum_{n=1}^{\infty} a_n^- \cos(\pi n x) \sinh(\pi n (1 - y)). \quad (39)$$

with some variable \bar{u}_0 to be specified later.

The equation for p_1 on Ω_1 can be converted to the form similar to (29):

$$-\bar{k} p_1''(x) + p_1(x) = \bar{P}_2 - \bar{B}_0 - \bar{u}_0 \sum_{n=1}^{\infty} \bar{a}_n \cos(n\pi x) \sinh(n\pi) \quad (40)$$

where we have used averaged variables:

$$\begin{aligned} \bar{\sigma} &= (\sigma^+ + \sigma^-), & \bar{k} &= \sqrt{k_1/\bar{\sigma}}, & \bar{P}_2 &= \frac{\sigma^+ P_2^+ + \sigma^- P_2^-}{\bar{\sigma}}, \\ \bar{B}_0 &= \frac{\sigma^+ B_0^+ + \sigma^- B_0^-}{\bar{\sigma}}, & \bar{a}_n &= \frac{\sigma^+ a_n^+ + \sigma^- a_n^-}{\bar{\sigma}}. \end{aligned}$$

Then we can repeat steps from Section 3.2 to get p_1 expansion that closely follows (31):

$$p_1(x) = \bar{P}_2 - \bar{B}_0 + \bar{u}_0 \cosh(x/\bar{k}) - \bar{u}_0 \sum_{n=1}^{\infty} \bar{u}_n \cos(\pi n x) \quad (41)$$

where we finally identify \bar{u}_0 as:

$$\bar{u}_0 = \frac{P_1 - \bar{P}_2 + \bar{B}_0 - \bar{u}_0 \bar{U}}{\cosh(1/\bar{k})}, \quad (42)$$

and we set:

$$\bar{u}_n = \frac{\bar{a}_n \sinh(n\pi)}{1 + (\bar{k}n\pi)^2}, \quad \bar{U} = - \sum_{n=1}^{\infty} (-1)^n \bar{u}_n. \quad (43)$$

4.1 Coupling

As in Section 3.3, we plug expansion of p_2^+ , p_2^- , and p_1 into (10) and (11). We group the terms to get Fourier expansion on the left hand side and a known function of x on the right hand side. In particular on Γ^+ we have:

$$\frac{\mathcal{A}_0^+}{2} + \sum_{n=1}^{\infty} \mathcal{A}_n^+ \cos(n\pi x) = -\bar{u}_0 \cosh(x/\bar{k}). \quad (44)$$

with

$$\frac{\mathcal{A}_0^+}{2} = \frac{k_2^+ B_0^+}{\sigma^+} - P_2^+ + \bar{P}_2 + B_0^+ - \bar{B}_0 \quad (45)$$

$$\mathcal{A}_n^+ = \bar{u}_0 a_n^+ \left(n\pi \frac{k_2^+}{\sigma^+} \cosh(n\pi) + \sinh(\pi n) \right) - \bar{u}_0 \bar{u}_n \quad (46)$$

We reuse the calculation of Fourier coefficients of the right hand side from (36) and (37) to obtain:

$$\mathcal{A}_0^+ = -2\bar{u}_0 \bar{k} \sinh(1/\bar{k}), \quad (47)$$

$$\mathcal{A}_n^+ = -\frac{(-1)^n 2\bar{u}_0 \bar{k}}{1 + (\bar{k}n\pi)^2} \sinh(1/\bar{k}). \quad (48)$$

Analogous relations hold for the lower side on Γ^- .

Now we combine (46) and (48), we cancel \bar{u}_0 , and we substitute for \bar{u}_n using (43). Performing the same operation for the \mathcal{A}_n^- we obtain a system to determine a_n^+ , a_n^- :

$$\begin{pmatrix} X_n^{00} & X_n^{01} \\ X_n^{01} & X_n^{11} \end{pmatrix} \begin{pmatrix} a_n^+ \\ a_n^- \end{pmatrix} = \begin{pmatrix} \sigma^+ y_n \\ \sigma^- y_n \end{pmatrix} \quad (49)$$

where

$$y_n = -\frac{(-1)^n 2\bar{\sigma}\bar{k}}{1 + (\bar{k}n\pi)^2} \sinh(1/\bar{k}), \quad (50)$$

$$X_n^{00} = \bar{\sigma}n\pi k_2^+ \cosh(n\pi) + \bar{\sigma}\sigma^+ \sinh(\pi n) - (\sigma^+)^2 \frac{\sinh(n\pi)}{1 + (\bar{k}n\pi)^2}, \quad (51)$$

$$X_n^{11} = \bar{\sigma}n\pi k_2^- \cosh(n\pi) + \bar{\sigma}\sigma^- \sinh(\pi n) - (\sigma^-)^2 \frac{\sinh(n\pi)}{1 + (\bar{k}n\pi)^2}, \quad (52)$$

$$X_n^{01} = -\sigma^- \sigma^+ \frac{\sinh(n\pi)}{1 + (\bar{k}n\pi)^2}, \quad (53)$$

$$(54)$$

The system matrix is strictly diagonally dominant providing $k_2^{+/-}$, $\sigma^{+/-}$, k_1 are positive. There for coefficients $a_n^{+/-}$ have the sign same as the right hand side, that is $(-1)^{n+1}$. This sign alternation cancels with signs in series for \bar{U} (43) which makes \bar{U} always positive.

Now we express \bar{u}_0 from (42):

$$\bar{u}_0 = \frac{P_1 - \bar{P}_2 + \bar{B}_0}{\cosh(1/\bar{k}) + \bar{U}} \quad (55)$$

and we plug it into (47) which we compare to (45). Taking the same procedure for \mathcal{A}_0^- we obtain a system for B_0^+ , B_0^- :

$$\begin{pmatrix} A^{00} & A^{01} \\ A^{01} & A^{11} \end{pmatrix} \begin{pmatrix} B_0^+ \\ B_0^- \end{pmatrix} = \begin{pmatrix} \bar{\sigma}\sigma^+(P_2^+ - P_b) \\ \bar{\sigma}\sigma^-(P_2^- - P_b) \end{pmatrix} \quad (56)$$

where

$$P_b = (1 - T)\bar{P}_2 + TP_1, \quad (57)$$

$$A^{00} = \bar{\sigma}k_2^+ + \bar{\sigma}\sigma^+ + (T - 1)(\sigma^+)^2, \quad (58)$$

$$A^{11} = \bar{\sigma}k_2^- + \bar{\sigma}\sigma^- + (T - 1)(\sigma^-)^2, \quad (59)$$

$$A^{01} = (T - 1)\sigma^+\sigma^- \quad (60)$$

denoting

$$T = \frac{\bar{k} \sinh(1/\bar{k})}{\cosh(1/\bar{k}) + \bar{U}}.$$

The system matrix is strictly diagonally dominant as long as

$$\frac{k_2^{+/-}}{\sigma^{+/-}} + 1 > 1 > T - 1.$$

Since \bar{U} is positive we conclude:

$$T < \bar{k} \tanh(1/\bar{k}) < 1$$

and the system matrix is always strictly diagonally dominant.

5 Evaluation procedure

Evaluation of the solution is performed in following steps:

1. Choose number of terms N in the truncated sums.
2. Solve systems (49) to obtain a_n^+ , a_n^- for $n = 1, \dots, N$.
3. Compute \bar{u}_n and the sum \bar{U} according to (43).
4. Solve system (56) for B_0^+ , B_0^- .
5. Compute \bar{u}_0 using (55).
6. Evaluate p_1 or p_2^+ , p_2^- by truncated summation of the series (41), (38), (39), respectively.

6 Verification by finite differences

In order to verify correctness of the analytical solution we have implemented a finite difference solution for both the conductive fracture and the barrier fracture. We use classical second order differences:

$$f''(x) = \frac{-2f(x) + f(x-h) + f(x+h)}{h^2} + O(h^2),$$

$$f'(x) = \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h} + O(h^2)$$

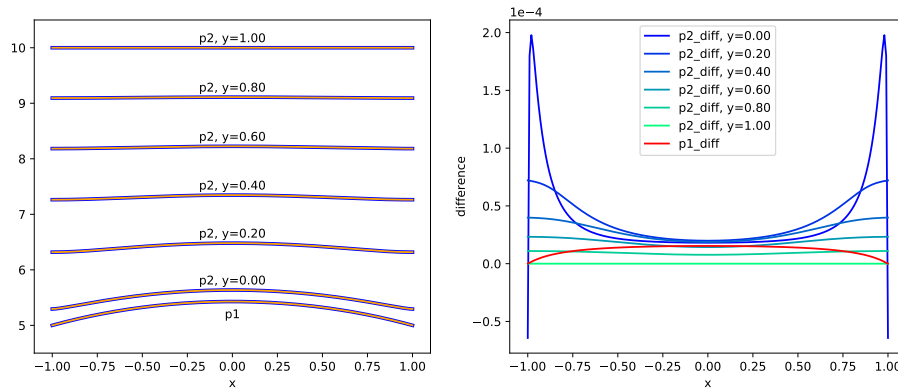


Figure 1: Conductive fracture case. *Left*: Match between the analytical and the numerical solution. *Right*: Difference of the analytical and the numerical solution.

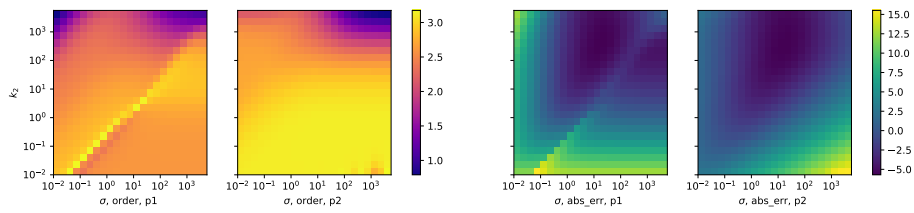


Figure 2: Convergence rate p and absolute error exponent c as a function of k_2 and σ . *Left*: Order of convergence p for p_1 , p_2 . *Right*: Absolute error exponent c for p_1 , p_2 .

where h is the mesh step.

Figure 6 presents good match between the numerical solution ($h = 0.01$) and the analytical solution (truncation after 1000 terms). Problem parameters were: $\sigma = 20$, $k_1 = 10$, $k_2 = 1$, $P_2 = 10$, $P_1 = 5$. Estimate for the sum remainder is 10^{-5} . The magnitude of the error is of order 10^{-4} . For the whole technically feasible range $h \in (0.1, 0.001)$ we have observed perfect second order convergence.

This is displayed at Figure 6 which contains results of a parametric study of the convergence rate as a function of parameters $k_1 \in [0.01, 0.1, 1, 10, 100]$ and σ passing through the same set of values. The pressures $P_2 = 10$, $P_1 = 5$ and the conductivity $k_2 = 1$ are kept constant. For a fixed pair k_1, σ we have estimated the rate of convergence by computing the finite difference solution for a sequence of mesh steps 0.1, 0.05, 0.025, 0.0125. For every mesh step the L^2 errors ϵ_1 and ϵ_2 were approximated by the midpoint rule for p_1 and p_2 respectively. The order of convergence p and the absolute error exponent c were determined by the fit:

$$\log_2(\epsilon) \approx -c + p \log_2(h)$$

Clearly the optimal second order convergence or better is preserved for majority of the parameter space. The drop to the linear convergence for combination of small k_1 and large σ is due to extreme derivatives of p_1 close to the endpoints $-1, 1$. This behavior can not be resolved by the used regular grid. Also the absolute error exponent is well above 0 so the error have a small magnitude.

In the similar fashion we have tested the analytical solution to the barrier case. Shape of the solution and its match with the finite difference approximation is shown in Figure 6. The problem parameters were: $k_1 = 0.5$, $k_2^+ = 5$, $k_2^- = 2$, $\sigma^+ = 20$, $\sigma^- = 10$, $P_1 = 0$, $P_2^+ = 10$, $P_2^- = -10$, mesh step $h = 0.01$. Notice the larger pressure gradient in lower part Ω^- with smaller conductivity $k_2^- = 2$, also the gap between $p_1(x)$ and $p_2(x, 0)$ is larger due to smaller value of $\sigma^- = 10$.

The convergence rate for varying parameters is shown in Figure 6. In particular we have used $\sigma^+ = 3 * \sqrt{\sigma^-}$, $\sigma^- = \sigma$ and $k_2^+ = k_2$, $k_2^- = \sqrt{k_2}$ with both σ and k_2 iterating through the list $[0.01, 0.1, 1, 10, 100]$. Remaining parameters were fixed on following values: $k_1 = 1$, $P_2^+ = 10$, $P_2^- = -10$, $P_1 = 0$. Overall convergence rate is close to or beyond second order as in the previous case we see problems of the finite differences to resolve high gradient in p_1 for high k_2 and σ . Absolute error is also small for small k_2 which is natural since the solution (both p_1 and p_2) tends to be constant in x .

7 Conclusion

The analytical solution in form of Fourier series has been derived for the symmetrical conductive fracture problem as well as for the general barrier fracture problem. The solution was modified for practical calculations and convergence rate of the series were estimated theoretically and confirmed numerically. It was realized that summing 100 terms provides precision about 10^{-6} which would be enough for most of applications. Adaptive summation can do even better. Analytical solution was successfully verified against a finite difference solution.

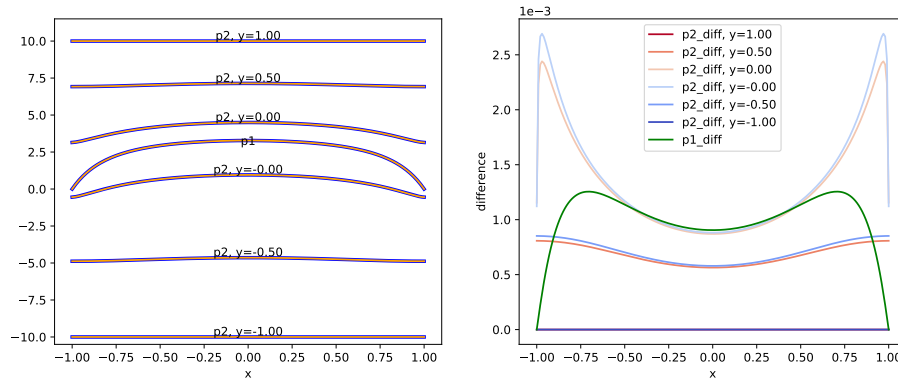


Figure 3: Barrier fracture case. *Left*: Match between the analytical and the numerical solution. *Right*: Difference of the analytical and the numerical solution.

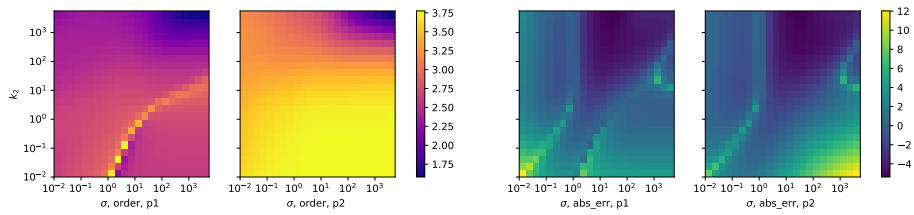


Figure 4: Convergence rate p and absolute error exponent c as a function of k_2 and σ . *Left*: Order of convergence p for p_1 , p_2 . *Right*: Absolute error exponent c for p_1 , p_2 .

The analytical solution is used as part of the test suit of the software Flow123d [2] to test various methods of coupling for both conforming and non-conforming mixed meshes.

References

- [1] Jan Březina and Jan Stebel. Analysis of model error for a continuum-fracture model of porous media flow. In *High Performance Computing in Science and Engineering*, number 9611 in Lecture Notes in Computer Science, pages 152–160. Springer International Publishing, 2015.
- [2] Jan Březina, Jan Stebel, Pavel Exner, and Jan Hybš. Flow123d. <http://flow123d.github.com>, repository: <http://github.com/flow123d/flow123d>, 2011–2016.
- [3] Lawrence C. Evans. *Partial Differential Equations*. American Mathematical Society.
- [4] Alessio Fumagalli and Anna Scotti. Numerical modelling of multiphase subsurface flow in the presence of fractures. *Communications in Applied and Industrial Mathematics*, 3(1), 2011.
- [5] Vincent Martin, Jérôme Jaffré, and Jean E. Roberts. Modeling Fractures and Barriers as Interfaces for Flow in Porous Media. 26(5):1667.



3.5 Scalable BDDC solver for conforming mixed mesh

The balanced domain decomposition by constraints provides efficient and scalable method to solve large systems. Its usage for systems formed for mixed mesh problems was challenging as for the technical complexity as for the significantly different conductivities on the fractures and on the matrix. Several new scaling methods have been proposed and tested in order to compensate for the discontinuities on domain interfaces.

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

Jakub Šístek^{1,*}, Jan Březina² and Bedřich Sousedík³

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 one-dimensional or two-dimensional elements inside three-dimensional domains. In this set-up, the global problem and 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. Copyright © 2015 John Wiley & Sons, Ltd.

Received 27 September 2013; Revised 9 February 2015; Accepted 14 April 2015

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 (CO₂) sequestration or nuclear waste disposal. In order to set up a reliable numerical model, one needs to have 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 Stokes–Brinkman equation, which reduces to either the Stokes or Darcy model in certain parameter limits, for example, within the multiscale mixed finite element framework [1]. In this paper, we consider the latter case and apply the Darcy 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

*Correspondence to: Jakub 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



and their intersections should be considered as two-dimensional (2D) and one-dimensional flows (1D), respectively. Because of the quite complex structure of the domains, the discretisation is performed using 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 because of the mixing of spatial dimensions, large jumps in permeability coefficients and the presence of elements of considerably different sizes, and so they are challenging for iterative solvers as well.

The matrices have a saddle-point structure

$$\begin{bmatrix} A & \overline{B}^T \\ \overline{B} & -\overline{C} \end{bmatrix}, \quad (1)$$

where A is symmetric positive definite on the kernel of \overline{B} and \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, [3–6], 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 [10–12]; 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 [19–22]. A closely related BDDC preconditioner for vector field problems discretised with the Raviart–Thomas finite elements has been studied in [23].

We are interested in the 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, because of 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 that 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 the Darcy law and the equation of continuity written as

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

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

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

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

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 because \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, for example, 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

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

equipped with the standard norm. Let $\mathbf{V} \subset \mathbf{H}(\Omega; \text{div})$ 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, for example, 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

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

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

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 that 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

$$\mathbf{V}^i = \{\mathbf{v} \in \mathbf{H}(T^i; \text{div}) : \mathbf{v} \in RT_0(T^i)\}, \quad (9)$$

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

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

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

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

$$= - \int_{\partial\Omega_N} p_N \mathbf{v} \cdot \mathbf{n} \, ds - \sum_{i=1}^{N_E} \int_{T^i} v_z \, dx, \quad \forall \mathbf{v} \in \mathbf{V}^{-1}, \quad (12)$$

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

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

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

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

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, 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 $T_d^i, i = 1, \dots, N_E^d$ and satisfies the compatibility conditions

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

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

for every $i \in \{1, \dots, N_E^{d-1}\}, j \in \{1, \dots, N_E^d\}$, 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 $\Omega_d, d = 1, 2, 3$, completed by a slight modification of the Darcy law (2):

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

where \mathbf{u}_d stands for the velocity integrated over the cross-section for $d = 1, 2$, that is, the units of $\mathbf{u}_3, \mathbf{u}_2$ and \mathbf{u}_1 are $\text{m s}^{-1}, \text{m}^2 \text{s}^{-1}$ and $\text{m}^3 \text{s}^{-1}$, respectively. In addition, $\delta_3 = 1, \delta_2$ 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

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

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

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

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

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,

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

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,

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

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

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



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

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

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:

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

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

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

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

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

with

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

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

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

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

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

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

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

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

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

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

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 $\overline{C} = \begin{bmatrix} C & C_{\mathcal{F}}^T \\ C_{\mathcal{F}} & \tilde{C} \end{bmatrix}$ 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 2D and 1D 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) and (20) and $p_2^k = \lambda_2^k$ in (22).

Assuming δ_d is 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 \tilde{C} is symmetric positive semi-definite because

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

The following theorem is a standard result, for example, [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, that is, $\partial\Omega_{N,d} \neq \emptyset$ 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,

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

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 $B_{\mathcal{F},dd}^T$. Because Ω_d is a simply connected set, the matrix $\overline{B}_{dd}^T = \begin{bmatrix} B_{dd}^T & B_{\mathcal{F},dd}^T \end{bmatrix}$ 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, that is, full column rank (see, e.g. [33, Section IV.1] or [26, Lemma 3.2]).

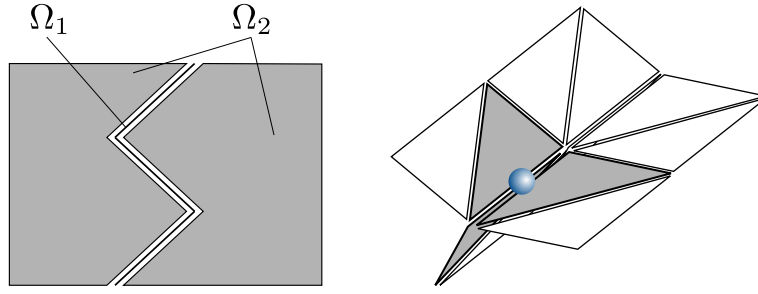


Figure 1. Example of a two-dimensional problem, even single fracture gives rise to two components of the two-dimensional mesh (left); example of three (shaded) triangles sharing a single Lagrange multiplier in three dimension (right).

The nullspace becomes more complicated for domains with fractures, in which case Ω_d typically has more simply connected components separated by fractures (cf. Figure 1). Let us denote them $\Omega_k^c, k = 1, \dots, n_c$, regardless of the dimension. In particular, $\Omega_k^c, k = 1, \dots, n_{ci}$ will be components without natural condition boundary, that is, $\partial\Omega_k^c \cap \partial\Omega_N = \emptyset$, while for $k = n_{ci} + 1, \dots, n_c$, we obtain components with prescribed natural boundary condition. We also denote $\bar{\chi}_k \in \bar{Q}$ 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

$$\bar{B}^T = \begin{bmatrix} B_{11}^T & & B_{\mathcal{F},11}^T & & \\ & B_{22}^T & & B_{\mathcal{F},22}^T & \\ & & B_{33}^T & & B_{\mathcal{F},33}^T \end{bmatrix} \quad (40)$$

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

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

Applying the congruence transformation, we obtain

$$\begin{bmatrix} A & \bar{B}^T \\ \bar{B} & -\bar{C} \end{bmatrix} = \begin{bmatrix} I & \\ \bar{B}A^{-1} & I \end{bmatrix} \begin{bmatrix} A & \\ & -(\bar{B}A^{-1}\bar{B}^T + \bar{C}) \end{bmatrix} \begin{bmatrix} I & A^{-1}\bar{B}^T \\ & I \end{bmatrix}. \quad (41)$$

Matrix A is symmetric positive definite from (29), and therefore, $\bar{B}A^{-1}\bar{B}^T$ is symmetric positive definite on $\text{range } \bar{B}$, which is the orthogonal complement of the nullspace of \bar{B}^T . Thus, the Schur complement $\bar{B}A^{-1}\bar{B}^T + \bar{C}$ is symmetric positive definite on whole \bar{Q} . From the Sylvester law of inertia, the number of positive, negative and zero eigenvalues is preserved by the congruence transformation. Because 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. \square

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, that is, $\mathcal{T}_{123} = \mathcal{T}_1 \cup \mathcal{T}_2 \cup \mathcal{T}_3$. The triangulation \mathcal{T}_{123} is subsequently divided into

substructures Ω^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_Γ^i be defined as the space of Lagrange multipliers corresponding to Γ^i , $i = 1, \dots, N_S$, and define a space

$$\Lambda_\Gamma = \Lambda_\Gamma^1 \times \dots \times \Lambda_\Gamma^{N_S}. \quad (42)$$

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

$$\begin{bmatrix} A^i & B^{iT} & B_{\mathcal{F},I}^{iT} & B_{\mathcal{F},\Gamma}^{iT} \\ B^i & -C^i & -C_{\mathcal{F},I}^{iT} & -C_{\mathcal{F},\Gamma}^{iT} \\ B_{\mathcal{F},I}^i & -C_{\mathcal{F},I}^i & -\tilde{C}_{II}^i & -\tilde{C}_{\Gamma I}^i \\ B_{\mathcal{F},\Gamma}^i & -C_{\mathcal{F},\Gamma}^i & -\tilde{C}_{\Gamma I}^i & -\tilde{C}_{\Gamma\Gamma}^i \end{bmatrix} \begin{bmatrix} \mathbf{u}^i \\ p^i \\ \lambda_I^i \\ \lambda_\Gamma^i \end{bmatrix} = \begin{bmatrix} \mathbf{g}^i \\ f^i \\ 0 \\ 0 \end{bmatrix}, \quad i = 1, \dots, N_S, \quad (43)$$

where the blocks A^i are block diagonal with blocks corresponding to finite element matrices and the blocks $\tilde{C}^i \neq 0$ 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 $S^i : \Lambda_\Gamma^i \mapsto \Lambda_\Gamma^i$, $i = 1, \dots, N_S$, formally as

$$S^i = \tilde{C}_{\Gamma\Gamma}^i + \begin{bmatrix} B_{\mathcal{F},\Gamma}^i & -C_{\mathcal{F},\Gamma}^i & -\tilde{C}_{\Gamma I}^i \end{bmatrix} \begin{bmatrix} A^i & B^{iT} & B_{\mathcal{F},I}^{iT} \\ B^i & -C^i & -C_{\mathcal{F},I}^{iT} \\ B_{\mathcal{F},I}^i & -C_{\mathcal{F},I}^i & -\tilde{C}_{II}^i \end{bmatrix}^{-1} \begin{bmatrix} B_{\mathcal{F},\Gamma}^{iT} \\ -C_{\mathcal{F},\Gamma}^{iT} \\ -\tilde{C}_{\Gamma I}^{iT} \end{bmatrix}. \quad (44)$$

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

Algorithm 2

Given $\lambda_\Gamma^i \in \Lambda_\Gamma^i$, determine $S^i \lambda_\Gamma^i \in \Lambda_\Gamma^i$ in the following two steps:

- (1) solve a local *Dirichlet problem*

$$\begin{bmatrix} A^i & B^{iT} & B_{\mathcal{F},I}^{iT} \\ B^i & -C^i & -C_{\mathcal{F},I}^{iT} \\ B_{\mathcal{F},I}^i & -C_{\mathcal{F},I}^i & -\tilde{C}_{II}^i \end{bmatrix} \begin{bmatrix} \mathbf{w}^i \\ q^i \\ \mu_I^i \end{bmatrix} = - \begin{bmatrix} B_{\mathcal{F},\Gamma}^{iT} \\ -C_{\mathcal{F},\Gamma}^{iT} \\ -\tilde{C}_{\Gamma I}^{iT} \end{bmatrix} \lambda_\Gamma^i, \quad (45)$$

- (2) perform two sparse matrix–vector multiplications

$$S^i \lambda_\Gamma^i \leftarrow - \left(-\tilde{C}_{\Gamma\Gamma}^i \lambda_\Gamma^i + \begin{bmatrix} B_{\mathcal{F},\Gamma}^i & -C_{\mathcal{F},\Gamma}^i & -\tilde{C}_{\Gamma I}^i \end{bmatrix} \begin{bmatrix} \mathbf{w}^i \\ q^i \\ \mu_I^i \end{bmatrix} \right). \quad (46)$$

Next, let $\hat{\Lambda}_\Gamma$ 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 $\lambda_\Gamma^i \in \Lambda_\Gamma^i$ and the vector of the global degrees of freedom $\lambda_\Gamma \in \hat{\Lambda}_\Gamma$ are related by

$$\lambda_\Gamma^i = R^i \lambda_\Gamma, \quad i = 1, \dots, N_S, \quad (47)$$

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 $\hat{S} : \hat{\Lambda}_\Gamma \rightarrow \hat{\Lambda}_\Gamma$ can be obtained as

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

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

$$\widehat{S} \lambda_\Gamma = \widehat{b}, \quad (49)$$

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

$$b^i = \begin{bmatrix} B_{\mathcal{F},\Gamma}^i & -C_{\mathcal{F},\Gamma}^i & -\tilde{C}_{\Gamma I}^i \end{bmatrix} \begin{bmatrix} A^i & B^{iT} & B_{\mathcal{F},I}^{iT} \\ B^i & -C^i & -C_{\mathcal{F},I}^{iT} \\ B_{\mathcal{F},I}^i & -C_{\mathcal{F},I}^i & -\tilde{C}_{II}^i \end{bmatrix}^{-1} \begin{bmatrix} g^i \\ f^i \\ 0 \end{bmatrix}. \quad (50)$$

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 $\lambda_\Gamma \rightarrow \widehat{S} \lambda_\Gamma$ 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 the explicit Schur complement with respect to the whole block of Lagrange multipliers λ and they show that because of 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 the 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, that is, $\partial\Omega_{N,d} \neq \emptyset$ 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) and (41), let us introduce a matrix $S = \overline{B} A^{-1} \overline{B}^T + \overline{C}$. The matrix S is symmetric positive definite by Theorem 1 (and its proof). Applying another congruence transformation to S and denoting the rows corresponding to the interface Lagrange multipliers by subscript Γ and the interior by I , we obtain

$$S = \begin{bmatrix} S_{II} & S_{\Gamma I}^T \\ S_{\Gamma I} & S_{\Gamma\Gamma} \end{bmatrix} = \begin{bmatrix} I & \\ S_{\Gamma I} S_{II}^{-1} & I \end{bmatrix} \begin{bmatrix} S_{II} & \\ S_{\Gamma\Gamma} - S_{\Gamma I} S_{II}^{-1} S_{\Gamma I}^T & \end{bmatrix} \begin{bmatrix} I & S_{II}^{-1} S_{\Gamma I}^T \\ & I \end{bmatrix}. \quad (51)$$

Because the matrix on the left-hand side is symmetric positive definite, both diagonal blocks S_{II} and $S_{\Gamma\Gamma} - S_{\Gamma I} S_{II}^{-1} S_{\Gamma I}^T$ 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 $\widehat{S} = S_{\Gamma\Gamma} - S_{\Gamma I} S_{II}^{-1} S_{\Gamma I}^T$. \square

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 (Remark 5).

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

$$\hat{\Lambda}_\Gamma \subset \tilde{\Lambda}_\Gamma \subset \Lambda_\Gamma. \quad (52)$$

Remark 5

In three spatial dimensions, several triangular elements can be connected at a single Lagrange multiplier in a star-like configuration (cf. Figure 1). A similar statement holds for line elements considered in 2D or 3D 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 earlier, 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 , for example, a corner corresponds to a single 1 entry at a row and an arithmetic average to several 1s at a row. The *coarse basis functions* Φ_Γ^i , 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

$$\begin{bmatrix} A^i & B^{iT} & B_{\mathcal{F},I}^{iT} & B_{\mathcal{F},\Gamma}^{iT} & 0 \\ B^i & -C^i & -C_{\mathcal{F},I}^{iT} & -C_{\mathcal{F},\Gamma}^{iT} & 0 \\ B_{\mathcal{F},I}^i & -C_{\mathcal{F},I}^i & -\tilde{C}_{II}^i & -\tilde{C}_{\Gamma I}^i & 0 \\ B_{\mathcal{F},\Gamma}^i & -C_{\mathcal{F},\Gamma}^i & -\tilde{C}_{\Gamma I}^i & -\tilde{C}_{\Gamma\Gamma}^i & D^{iT} \\ 0 & 0 & 0 & D^i & 0 \end{bmatrix} \begin{bmatrix} X^i \\ Z^i \\ \Phi_I^i \\ \Phi_\Gamma^i \\ L^i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ I \end{bmatrix}, \quad i = 1, \dots, N_S, \quad (53)$$

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

$$S_{CC}^i = \begin{bmatrix} X^{iT} & Z^{iT} & \Phi_I^{iT} & \Phi_\Gamma^{iT} \end{bmatrix} \begin{bmatrix} A^i & B^{iT} & B_{\mathcal{F},I}^{iT} & B_{\mathcal{F},\Gamma}^{iT} \\ B^i & -C^i & -C_{\mathcal{F},I}^{iT} & -C_{\mathcal{F},\Gamma}^{iT} \\ B_{\mathcal{F},I}^i & -C_{\mathcal{F},I}^i & -\tilde{C}_{II}^i & -\tilde{C}_{\Gamma I}^i \\ B_{\mathcal{F},\Gamma}^i & -C_{\mathcal{F},\Gamma}^i & -\tilde{C}_{\Gamma I}^i & -\tilde{C}_{\Gamma\Gamma}^i \end{bmatrix} \begin{bmatrix} X^i \\ Z^i \\ \Phi_I^i \\ \Phi_\Gamma^i \end{bmatrix} = -L^i. \quad (54)$$

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

$$\mu_C^i = R_C^i \mu_C. \quad (55)$$



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

$$S_{CC} = \sum_{i=1}^{N_S} R_C^{iT} S_{CC}^i R_C^i. \quad (56)$$

Finally, let us define the scaling operators

$$W^i : \Lambda_\Gamma^i \rightarrow \Lambda_\Gamma^i, \quad i = 1, \dots, N_S, \quad (57)$$

which are given as diagonal matrices of weights that satisfy

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

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 $M_{BDDC} : r_\Gamma \in \widehat{\Lambda}_\Gamma \rightarrow \lambda_\Gamma \in \widehat{\Lambda}_\Gamma$ is defined in the following steps:

- (1) Compute the local residuals

$$r_\Gamma^i = W^i R^i r_\Gamma, \quad i = 1, \dots, N_S. \quad (59)$$

- (2) Compute the *substructure corrections* $\eta_{\Gamma\Delta}^i$ by solving the local *Neumann problems*

$$\begin{bmatrix} A^i & B^{iT} & B_{\mathcal{F},I}^{iT} & B_{\mathcal{F},\Gamma}^{iT} & 0 \\ B^i & -C^i & -C_{\mathcal{F},I}^{iT} & -C_{\mathcal{F},\Gamma}^{iT} & 0 \\ B_{\mathcal{F},I}^i & -C_{\mathcal{F},I}^i & -\tilde{C}_{II}^i & -\tilde{C}_{\Gamma I}^{iT} & 0 \\ B_{\mathcal{F},\Gamma}^i & -C_{\mathcal{F},\Gamma}^i & -\tilde{C}_{\Gamma I}^i & -\tilde{C}_{\Gamma\Gamma}^{iT} & D^{iT} \\ 0 & 0 & 0 & D^i & 0 \end{bmatrix} \begin{bmatrix} x^i \\ z^i \\ \eta_{I\Delta}^i \\ \eta_{\Gamma\Delta}^i \\ l^i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ r_\Gamma^i \\ 0 \end{bmatrix}, \quad i = 1, \dots, N_S. \quad (60)$$

- (3) Compute the *coarse correction* η_C^i by collecting the coarse residual

$$r_C = \sum_{i=1}^{N_S} R_C^{iT} \Phi_\Gamma^{iT} r_\Gamma^i, \quad (61)$$

solving the *global coarse problem*

$$S_{CC} \eta_C = r_C \quad (62)$$

and distributing the coarse correction

$$\eta_{\Gamma C}^i = \Phi_\Gamma^i R_C^i \eta_C, \quad i = 1, \dots, N_S. \quad (63)$$

- (4) Combine and average the corrections

$$\lambda_\Gamma = - \sum_{i=1}^{N_S} R^{iT} W^i (\eta_{\Gamma\Delta}^i + \eta_{\Gamma C}^i). \quad (64)$$

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. The substructure corrections in (60), due to (44), can be written equivalently as

$$\begin{bmatrix} -S^i & D^{iT} \\ D^i & 0 \end{bmatrix} \begin{bmatrix} \eta_{\Gamma\Delta}^i \\ l^i \end{bmatrix} = \begin{bmatrix} r_\Gamma^i \\ 0 \end{bmatrix} \quad (65)$$

and the construction of coarse basis functions Φ_Γ^i in (53) as

$$\begin{bmatrix} -S^i & D^{iT} \\ D^i & 0 \end{bmatrix} \begin{bmatrix} \Phi_\Gamma^i \\ L^i \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}. \quad (66)$$

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

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

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

$$M_{BDDC} = R^T W \tilde{S}^{-1} W R, \quad (68)$$

where

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

The first term in \tilde{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

$$\text{null } S^i \perp \text{null } D^i. \quad (70)$$

In order to satisfy Assumption 7, we must prescribe enough coarse degrees of freedom as constraints along with the Robin boundary conditions (19), (20) and (22) at each fracture within substructure Ω^i . Because constraints 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 \tilde{S}^{-1} in preconditioner M_{BDDC} is symmetric and positive definite on the space $\tilde{\Lambda}_\Gamma$.

Proof

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

$$\tilde{\Lambda}_\Gamma = \tilde{\Lambda}_{\Gamma\Delta} \oplus \tilde{\Lambda}_{\Gamma C}. \quad (71)$$

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

$$\Lambda_\Gamma^i = \text{null } D^i \oplus \text{range } \Phi_\Gamma^i, \quad (72)$$

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

$$S^i \Phi_\Gamma^i = D^{iT} L^i, \quad (73)$$

$$D^i \Phi_\Gamma^i = I. \quad (74)$$



From (73),

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

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

$$\chi_\Delta^{iT} S^i \xi_C^i = 0. \quad (76)$$

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

$$\text{null } D^i \cap \text{range } \Phi_\Gamma^i = 0. \quad (77)$$

Finally, from Assumption 7 and (72),

$$\text{null } S^i \subset \text{range } \Phi_\Gamma^i. \quad (78)$$

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 $\zeta_\Delta^i \in \text{null } D^i$, $\zeta_C^i \in \text{range } \Phi_\Gamma^i$ 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 because of Assumption 7. If we define, in addition, a matrix Q^i with orthonormal columns forming a basis of $\text{null } D^i$, that is,

$$\text{range } Q^i = \text{null } D^i, \quad Q^{iT} Q^i = I, \quad (79)$$

we have

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

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

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

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

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

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

Because 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 \tilde{S}^{-1} , and the desired positive definiteness is obtained through the change of sign in front of the braces in (69). \square

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 $M_{BDDC} \hat{S}$ satisfies

$$\kappa \leq \omega = \max_{\lambda_\Gamma \in \tilde{\Lambda}_\Gamma} \frac{\|RR^T W \lambda_\Gamma\|_S^2}{\|\lambda_\Gamma\|_S^2}. \quad (83)$$

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

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



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

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; and (ii) it is not clear whether in the 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 $\lambda_{\Gamma,j}^i$ is given by the inverse counting function as

$$W_{jj}^i = \frac{1}{\text{card}(\mathcal{I}_j)}, \quad (85)$$

where $\text{card}(\mathcal{I}_j)$ is the number of substructures in the set \mathcal{I}_j of indices of substructures to which $\lambda_{\Gamma,j}^i$ belongs. For 2D or 3D meshes without fractures, $W_{jj}^i = 1/2$ for the Raviart–Thomas elements. However, because several 2D 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

$$W_{jj}^i = \frac{\rho_i}{\sum_{k=1}^{\text{card}(\mathcal{I}_j)} \rho_k}, \quad (86)$$

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 that takes into account the material coefficient of the element to which the Lagrange multiplier $\lambda_{\Gamma,j}^i$ corresponds. In our case, we use $\rho_i = d/\text{tr}(\mathbb{k}^{-1})$, 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

$$W_{jj}^i = \tilde{C}_{\Gamma,jj}^i + \frac{1}{A_{kk}^i}, \quad (87)$$

where the index k corresponds to the row in block A^i of the element face to which the Lagrange multiplier $\lambda_{\Gamma,j}^i$ 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 the 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 the Darcy 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 3D elements of porous media are combined with 2D elements modelling planar fractures, which may be in turn connected in 1D elements for channels. The 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.

[‡] <http://flow123d.github.io>

[§] <http://users.math.cas.cz/~sistek/software/bddcml.html>



Although the original system (37) is indefinite, system (49) is symmetric positive definite, which allows the use of the 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). Because both are then strictly negative definite, the product $M_{BDDC}\hat{S}$ 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.

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 geoenvironment 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 geoenvironment 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 $\|r^{(k)}\| / \|\hat{b}\| < 10^{-7}$. 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 2D and 3D finite elements, respectively. For this reason, block \bar{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

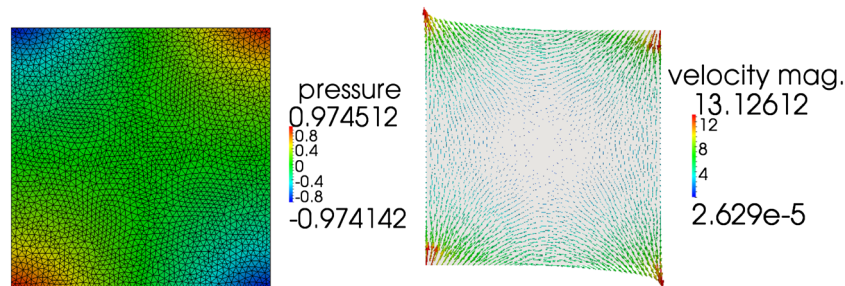


Figure 2. Example of solution to the model square problem containing only two-dimensional elements, plot of pressure head with mesh (left) and velocity vectors (right).

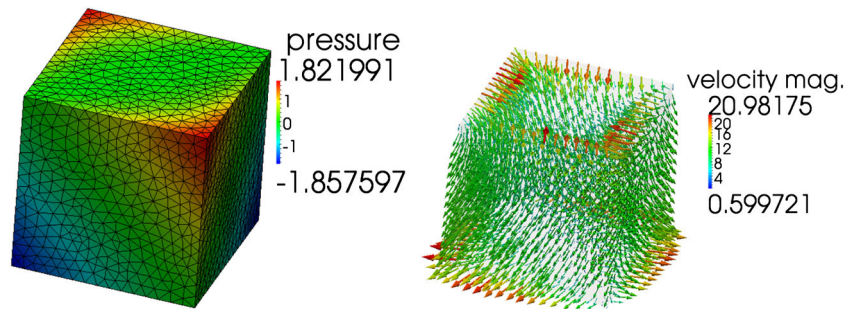


Figure 3. Example of solution to the model cube problem containing only three-dimensional elements, plot of pressure head with mesh (left) and velocity vectors (right).

problems, and the problems do not contain any jumps in material coefficients. In Figures 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 Figure 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.

Table I. Weak scaling test for the 2D square problem, each substructure problem contains approximately 100 000 unknowns.

N	$n(\times 10^3)$	$n/N(\times 10^3)$	n_Γ	n_f	n_c	Its.	Cond.	Time (s)		
								Set-up	PCG	Solve
2	207	103	155	1	2	7	1.37	8.3	1.6	9.9
4	440	110	491	5	10	8	1.60	12.2	2.2	14.4
8	822	103	1 236	13	26	9	1.78	11.0	2.5	13.5
16	1 783	111	2 816	33	66	8	1.79	14.3	2.7	17.0
32	3 332	104	5 940	74	148	9	1.79	12.1	3.3	15.4
64	7 201	113	13 048	166	332	9	1.85	14.8	4.4	19.2

Its., resulting number of preconditioned conjugate gradient (PCG) iterations; Cond., the approximate condition number from the Lanczos sequence in PCG.

Table II. Weak scaling test for the 3D cube problem, each substructure problem contains approximately 100 000 unknowns.

N	$n(\times 10^3)$	$n/N(\times 10^3)$	n_Γ	n_f	n_c	Its.	Cond.	Time (s)		
								Set-up	PCG	Solve
2	217	108	884	1	3	11	2.88	11.7	2.3	14.0
4	437	109	2 315	6	18	12	3.04	11.7	2.5	14.2
8	945	118	5 677	21	63	15	12.00	15.4	4.0	19.3
16	1 647	103	12 773	56	168	16	6.58	12.9	4.0	17.0
32	3 393	106	29 824	132	401	18	10.10	15.4	5.2	20.6
64	6 108	95	59 617	307	931	19	16.58	13.7	6.3	20.0

Its., resulting number of preconditioned conjugate gradient (PCG) iterations; Cond., the approximate condition number from the Lanczos sequence in PCG.

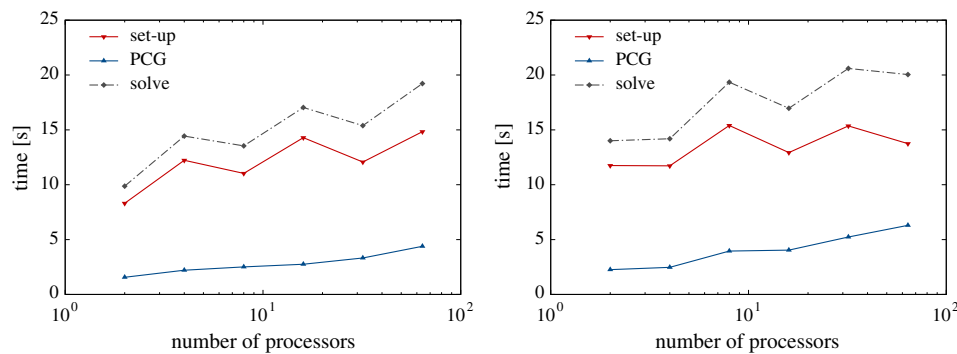


Figure 4. Weak scaling test for the two-dimensional square problem (left) and the three-dimensional cube problem (right), approximately 100 000 unknowns per core. Computational times separately for set-up and preconditioned conjugate gradient (PCG) phases and their sum (solve).

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 the Darcy 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 Figure 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 2D and 3D settings and dominates the overall time.

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 \mathbf{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 Figure 5, the computational mesh and the resulting pressure head and velocity fields are presented. This scaling 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 Figure 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

$$s_{np} = \frac{16 t_{16}}{t_{np}}, \quad (88)$$

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 seems 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.

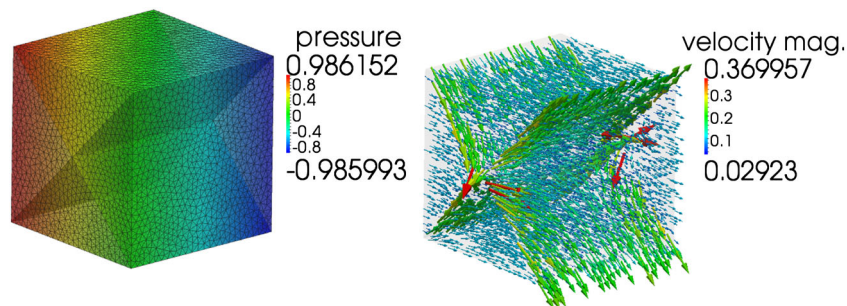


Figure 5. Example of solution to the model cube problem containing one-dimensional, two-dimensional and three-dimensional elements, plot of pressure head with mesh and fractures (left) and velocity vectors (right).

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

N	$n/N(\times 10^3)$	$n_\Gamma(\times 10^3)$	n_f	n_c	Its.	Cond.	Time (s)		
							Set-up	PCG	Solve
16	912	47	53	159	26	59.3	171.6	84.5	256.2
32	456	65	126	380	48	2091.0	90.1	109.8	200.0
64	228	86	301	914	81	1436.1	36.8	77.1	114.0
128	114	116	689	2076	109	2635.8	14.3	43.1	57.4
256	57	151	1436	4365	164	1700.5	6.7	31.2	38.0
512	28	196	3021	9244	254	42614.5	4.0	26.9	30.9

Its., resulting number of preconditioned conjugate gradient (PCG) iterations; Cond., the approximate condition number from the Lanczos sequence in PCG.

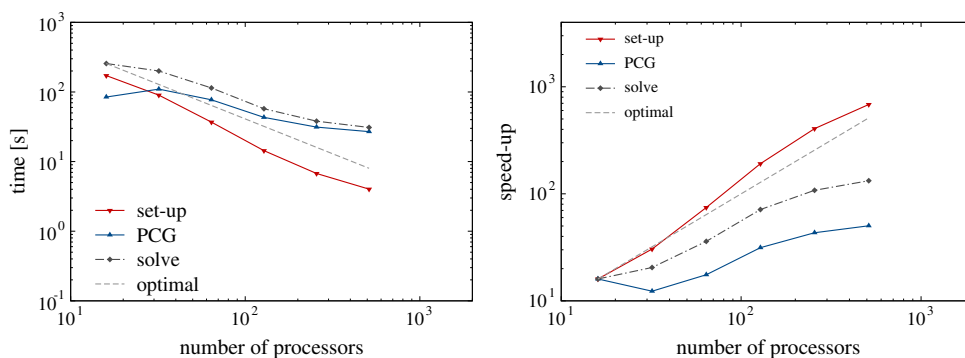


Figure 6. Strong scaling test for the cube problem with one-dimensional, two-dimensional and three-dimensional elements and 14.6 million unknowns, computational time (left) and speed-up (right) separately for set-up and preconditioned conjugate gradient (PCG) phases, and their sum (solve).

8.2. Results for geoenvironmental 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 2D 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 the 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 Figure 7. The problem contains vertical 2D fractures visualised in Figure 8. The maximal hydraulic conductivity within the fractures is $6.3 \cdot 10^4 \text{ m s}^{-1}$, while the minimal conductivity of the outer material is $6.0 \cdot 10^{-3} \text{ m s}^{-1}$, the transition coefficient $\sigma_3 = 1 \text{ s}^{-1}$ and the effective thickness of fractures $\delta_2 = 0.1 \text{ m}$.

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 Figure 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 Figure 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 is to accommodate

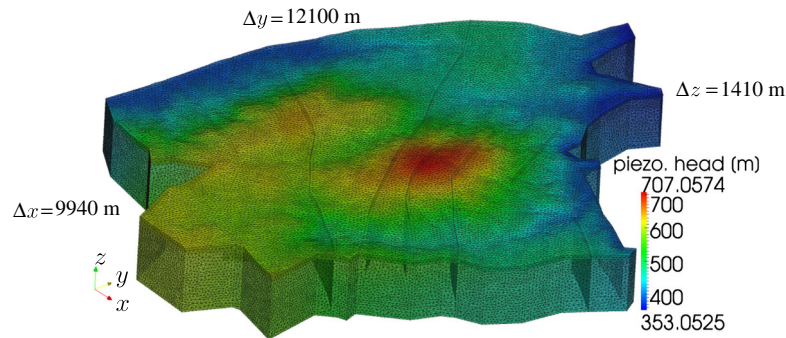


Figure 7. The problem of the *Melechov locality* containing two-dimensional and three-dimensional elements; mesh contains 2.1 million elements and 15 million unknowns. Plot of the piezometric head. Data by courtesy of Jiřina Královcová.

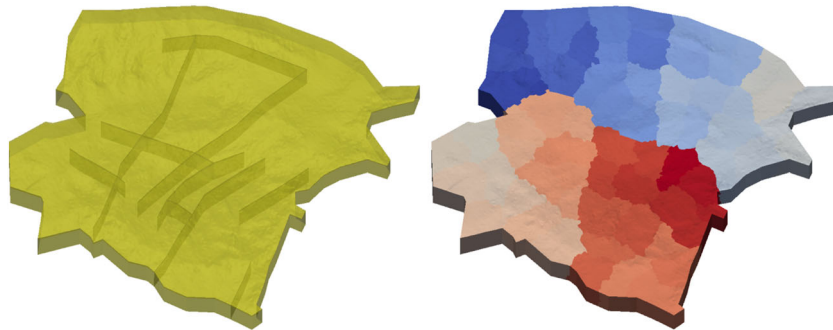


Figure 8. The problem of the *Melechov locality*: the system of fractures (left) and an example division into 64 substructures (right).

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

N	$n/N(\times 10^3)$	$n_\Gamma(\times 10^3)$	n_f	n_c	Its.	Cond.	Time (s)		
							Set-up	PCG	Solve
16	934	36	32	96	40	53.0	131.4	144.1	275.6
32	467	54	76	228	70	878.3	47.5	112.9	160.4
64	233	82	186	561	67	202.4	17.4	50.2	67.7
128	117	116	528	1592	69	237.6	7.9	23.1	31.1
256	58	155	1235	3747	96	5577.0	4.0	14.7	18.8
512	29	207	2699	8256	106	1658.1	2.2	8.3	10.5
1024	15	271	5711	17581	119	11554.5	2.1	7.0	9.2

Its., resulting number of preconditioned conjugate gradient (PCG) iterations; Cond., the approximate condition number from the Lanczos sequence in PCG.

water pipes, which supply the city by drinking water from a reservoir in the mountains. However, 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 Figure 10. The system of fractures and an example division into 256 substructures are visualised in Figure 11. The hydraulic conductivity of the fractures is 10^{-7} m s^{-1} , while the conductivity of the outer material is $10^{-10} \text{ m s}^{-1}$, the transition coefficient $\sigma_3 = 1 \text{ s}^{-1}$ and the effective thickness of fractures $\delta_2 = 1.1 \text{ m}$.

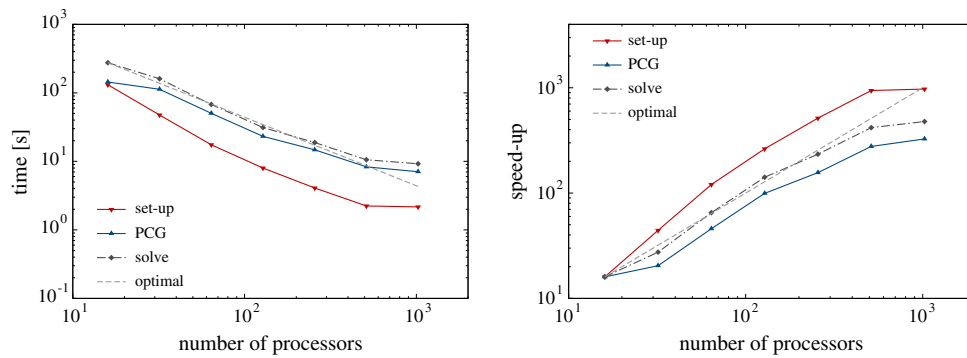


Figure 9. Strong scaling test for the problem of the *Melechov locality* containing two-dimensional and three-dimensional elements and 15 million unknowns, computational time (left) and speed-up (right) separately for set-up and preconditioned conjugate gradient (PCG) phases, and their sum (solve).

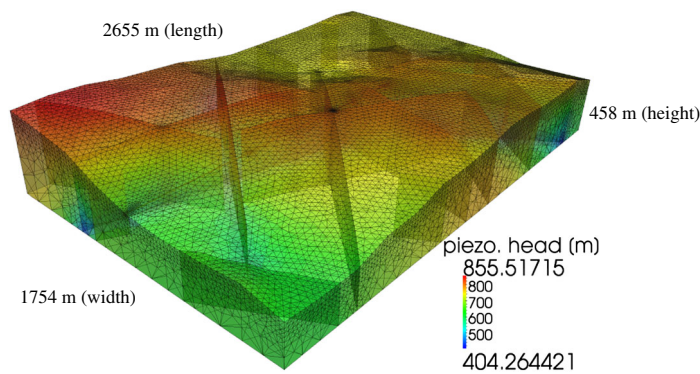


Figure 10. The *Bedřichov tunnel* problem containing two-dimensional and three-dimensional elements; mesh contains 1.1 million elements and 7.8 million unknowns. Plot of the piezometric head. Data by courtesy of Dalibor Frydrych.

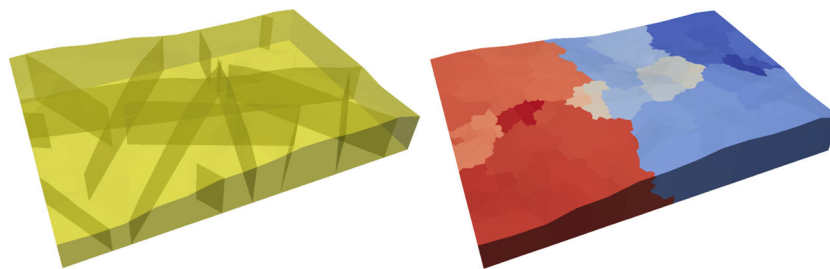


Figure 11. The *Bedřichov tunnel* problem: the system of fractures (left) and an example division into 256 substructures (right).

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 (Figure 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 Figure 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

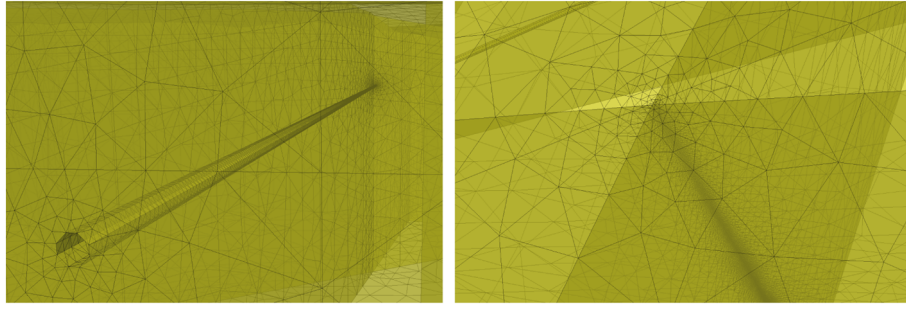


Figure 12. Example of difficulties in the mesh of the *Bedřichov tunnel* problem: detail of the tunnel geometry with fine elements (left) and enforced refinement at an intersection of fractures (right).

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

N	$n/N(\times 10^3)$	$n_\Gamma(\times 10^3)$	n_f	n_c	Its.	Cond.	Time (s)		
							Set-up	PCG	Solve
32	245	20	106	322	112	1514.1	110.3	144.0	254.3
64	123	28	192	597	63	117.7	42.2	36.0	78.3
128	61	45	413	1293	75	194.4	13.4	16.8	30.3
256	31	72	902	2791	119	526.7	4.2	10.9	15.1
512	15	110	2009	6347	137	1143.4	1.8	7.1	9.0
1024	8	155	4575	14725	173	897.0	1.6	8.0	9.7

Its., resulting number of preconditioned conjugate gradient (PCG) iterations; Cond., the approximate condition number from the Lanczos sequence in PCG.

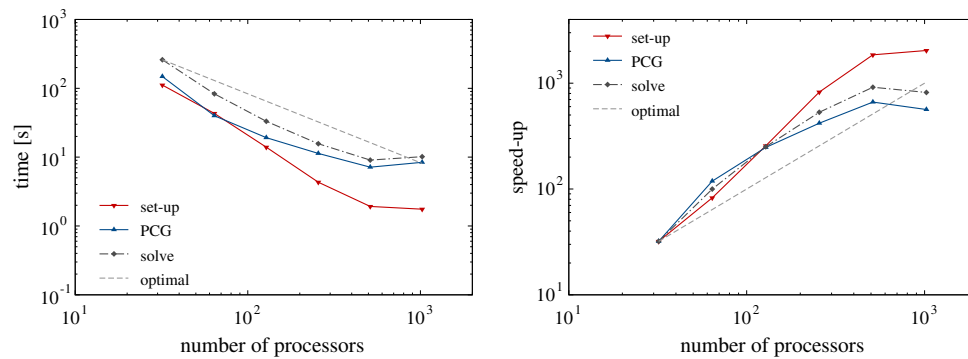


Figure 13. Strong scaling test for the problem of the *Bedřichov tunnel* containing two-dimensional and three-dimensional elements and 7.8 million unknowns, computational time (left) and speed-up (right) separately for set-up and PCG phases, and their sum (solve).

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 the 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, for example, in elasticity analysis [21]. This is also the default option for *BDDCML*, in which selection

Table VI. Effect of using corners.

N	n/N ($\times 10^3$)	Without corners						With corners				
		Its.	Cond.	Time (s)			Its.	Cond.	Time (s)			
				Set-up	PCG	Solve			Set-up	PCG	Solve	
32	245	131	1789.6	107.5	175.0	282.5	112	1514.1	110.3	144.0	254.3	
64	123	70	122.2	40.3	40.4	80.7	63	117.7	42.2	36.0	78.3	
128	61	96	208.5	10.9	21.6	32.6	75	194.4	13.4	16.8	30.3	
256	31	139	541.9	3.7	12.5	16.2	119	526.7	4.2	10.9	15.1	
512	15	197	1418.9	1.4	10.0	11.4	137	1143.4	1.8	7.1	9.0	
1024	8	312	3779.4	1.0	14.5	15.6	173	897.0	1.6	8.0	9.7	

Problem of the *Bedřichov tunnel* containing two-dimensional and three-dimensional elements, size of the global problem is $n = 7.8$ million unknowns. In column ‘Without corners’, no additional corners are selected in balancing domain decomposition by constraints. In column ‘With corners’, additional corners are selected. Its., resulting number of preconditioned conjugate gradient (PCG) iterations; Cond., the approximate condition number from the Lanczos sequence in PCG.

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

N	n/N ($\times 10^3$)	n_Γ ($\times 10^3$)	n_f	n_c	Arithmetic avg.		Mod. ρ -scal.		Diagonal scal.	
					Its.	Cond.	Its.	Cond.	Its.	Cond.
32	245	20	106	322	637	9811.7	110	1467.8	112	1514.1
64	123	28	192	597	618	10254.1	62	115.1	63	117.7
128	61	45	413	1293	2834	1.0e+11	206	401641.4	75	194.4
256	31	72	902	2791	799	11172.9	117	512.9	119	526.7
512	15	110	2009	6347	883	15449.6	136	1160.1	137	1143.4
1024	8	155	4575	14725	n/a	2.5e+10	504	99023.6	173	897.0

Its., resulting number of preconditioned conjugate gradient (PCG) iterations; Cond., the approximate condition number from the Lanczos sequence in PCG.

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



9. CONCLUSION

A parallel solver for the mixed-hybrid finite element formulation based on the Darcy 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 geoen지니어ing 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 the Darcy law, although the hydraulic conductivity of the porous media is considered by orders of magnitude 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 the Robin boundary conditions. These coupling terms lead to a modification of the usual saddle-point matrix of the system, in which a new nonzero 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.

ACKNOWLEDGEMENTS

The authors are grateful to Jiřina Královcová and Dalibor Frydrych for providing the geoen지니어ing models. The research was supported by the Czech Science Foundation under the project GA CR 14-02067S, by the Academy of Sciences of the Czech Republic through RVO:67985840 and by the 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).



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-Verlag: New York, 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: Oxford, 2005.
9. Toselli A, Widlund OB. *Domain Decomposition Methods—Algorithms and Theory*, Springer Series in Computational Mathematics, vol. 34. Springer-Verlag: Berlin, Heidelberg, 2005.
10. Cros JM. A preconditioner for the Schur complement domain decomposition method. In *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 Journal of Scientific Computing* 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 Institute of Mathematical Sciences, Department of Computer Science, 2013.
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. In *Mathematical Aspects of Finite Element Methods*, vol. 606, Galligani I, Magenes E (eds)., Lecture Notes in Mathematics. 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. In *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*. Courier Corporation, 1988.
32. Chen Z, Huan G, Ma Y. *Computational Methods for Multiphase Flows in Porous Media*. SIAM: Philadelphia, 2006.
33. Brezzi F, Fortin M. *Mixed and Hybrid Finite Element Methods*. Springer-Verlag: New York, 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. *Mathematical Computations and Simulation* 2012; **82**(10):1788–1798.
36. Mandel J, Dohrmann CR, Tezaur R. An algebraic theory for primal and dual substructuring methods by constraints. *Applied Numerical Mathematics* 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. In *Proceedings of Programs and Algorithms of Numerical Mathematics 16, Dolní Maxov, Czech Republic, June 3–8, 2012*. Institute of Mathematics AS CR: Prague, 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, Souček B, Šístek J. Adaptive BDDC in three dimensions. *Mathematics and Computers in Simulation* 2012; **82**(10):1812–1831.



Chapter 4

Numerical methods for non-conforming mixed meshes

Non-conforming coupling between elements can overcome difficulties with meshing the mixed meshes. On the other hand the solver has to deal with two new issues: calculation of element intersections and prescribe a suitable approximation of the coupling equations (2.7–2.8) on these intersections. The first section of this chapter discuss some approaches to the second problem while the second and the last section presents a collection of algorithms for efficient calculation of the mesh intersections.

4.1 Mixed-Hybrid Method on Non-conforming Mixed Meshes

We can classify intersections between pair of elements $K \in \mathcal{T}_{d_K}$ and $L \in \mathcal{T}_{d_L}$ by the *codimension* which we define as absolute difference of the element's dimensions, $|d_K - d_L|$.

Codimension 0. Continuity on the 2d-2d intersections in 3d ambient space and 1d-1d intersections in 2d ambient space must be enforced unless we assume a single continuous submesh per dimension.

Codimension 1. Continuum-fracture interaction, in particular conditions (2.7 – 2.8) need to be approximated on 2d-3d and 1d-2d intersections in 3d and 2d ambient space respectively.

Codimension 2. Coupling for 3d-1d and 2d-1d intersections in 3d ambient space have singular character and require specific methods, e.g. XFEM.

We will focus on the coupling for the codimension 1 intersections. Proposed methods are a generalization of the Mortar method which can be used for enforcing continuity for codimension 0 intersections. We further restrict our interest to the case of conductive fractures where we can assume continuity of the pressure across the fracture. The general case requires local enrichment of the finite element space by the jump functions to represent discontinuities. We also completely omit the codimension 2 case.

Assuming continuity of the pressure across the fracture we can merge duplicated DOFs of the trace pressure \tilde{p} on fractures and replace the form c_f by an approximation. We start with change of the discrete spaces \tilde{Q}_d^h which we redefine as the space of piecewise constant functions on edges:

$$\tilde{Q}_d^h = \left\{ \tilde{q} \in \prod_{T \in \mathcal{T}_d} \tilde{Q}^h(T) : \tilde{q}|_{\partial K} = \tilde{q}|_{\partial L} \text{ on } \partial K \cap \partial L \subset \Gamma_d \forall K, L \in \mathcal{T}_d \right\}. \quad (4.1)$$



Now we can replace the coupling form c_f in (3.25) by its approximation:

$$c_F(p, \hat{p}, q, \hat{q}) = \sum_{d=\{1,2\}} \sum_{K \in \mathcal{T}_d} \int_T \sigma_d(R(\hat{p}_d) - T(\hat{p}_{d+1}))(R(\hat{\phi}_d) - T(\hat{\phi}_{d+1})),$$

where R is a reconstruction operator of the pressure p and T is the trace approximation. The reconstruction operator maps p_d and \hat{p}_d DOFs to a suitable pressure function on the Ω_d domain, while the trace approximation maps p_{d+1} and \hat{p}_{d+1} DOFs to a function on the Ω_d that approximates the trace of the \bar{p}_{d+1} pressure on Ω_d . Several different methods can be obtained by different choice of operators R and T . In the following, we shall present four methods: P^0 and P^1 *direct* methods and P^0 and P^1 *mortar like* methods. In order to do not break solving the linear system by the two consecutive Schur complements, we construct R and T operators that use only \hat{p}_d and \hat{p}_{d+1} DOFs respectively. Both direct methods first construct a \bar{p}_{d+1} function from p_{d+1} and \hat{p}_{d+1} DOFs and then evaluates its trace on Ω_d . The integration in c_F must be split into integrals over intersections of the element $K \in \mathcal{T}_d$ with elements $L \in \mathcal{T}_{d+1}$. The mortar like methods projects the \bar{p}_{d+1} function into the image space of the R operator. The integration in c_F is evaluated over whole elements $K \in \mathcal{T}_d$. As we will see not all of these methods actually works. On the other hand one other methods can be constructed within this general framework.

In order to proceed we first introduce necessary notation. Since the integration in c_F is performed over individual elements $K \in \mathcal{T}_d$, we fix a single element K and introduce all notations and definitions of R and T operators only for this single element K . We denote by I_L intersection of K with an element L from \mathcal{T}_{d+1} and we also use $\delta_L = |I_L|$ for its d -dimensional measure. We also denote by $\delta_K = \sum \delta_L$ the measure of the element K . Further we denote by \hat{p}_T a set of the trace pressure DOFs of an arbitrary element T and by $\hat{p}_{T,i}$ the DOF on a single side $i \in \partial T$ of this element. Finally, we introduce an average element pressure for any element T of any dimension d as:

$$\bar{p}_T = \sum_{i \in \partial T} \frac{\hat{p}_{T,i}}{d+1}.$$

4.1.1 Direct P^0 Method

The direct P_0 method just imposes coupling between individual pairs of intersecting elements using their average pressures:

$$R(\hat{p}_d)|_K = \bar{p}_K, \quad T(\hat{p}_{d+1})|_{I_L} = \bar{p}_L.$$

This kind of coupling was proposed already at the beginning of the Flow123d project (see [17]), but using directly elementwise pressures p_K, p_L instead their approximation by the averages of the trace pressures. However this leads to unphysical locking phenomena for the high values of σ_d . In this case every pair of intersecting elements acts as an independent constraint prescribing equality of the pressures which cause overconstrained problem. In our case the overconstraining is relaxed by the usage of averages however this is still not enough to avoid locking in the 3d case.

4.1.2 Direct P^1 Method

In order to relax the overconstrained problem and also use higher order method we first interpret the trace pressures as DOFs of the non-conforming P^1 space (piecewise linear with continuity in face barycenters). Let us denote $\mathbf{S}_i, i = 0, \dots, d$ the edge barycenters of the element K of

dimension d . We find a basis $\phi_i(\mathbf{x})$, $i = 0, \dots, d$ of the space of linear functions on K that is orthogonal to the functionals $\Phi_j(\phi) = \phi(S_j)$, $j = 0, \dots, d$. With \check{p}_i , $i = 0, \dots, d$ denoting the trace pressure DOFs on the element K , we introduce the operator \mathcal{P} that creates the non-conforming linear function:

$$\mathcal{P}_K[\check{p}](\mathbf{x}) = \sum_i \check{p}_i \phi_i(\mathbf{x}).$$

Finally, on the intersection I_L we define R and T operators as:

$$R(\check{p})|_K = \mathcal{P}_K[\check{p}_K], \quad T(\check{p})|_L = \mathcal{P}_L[\check{p}_L].$$

Using a space of linear functions eliminates the locking for the case of 1d fracture in a 2d continuum. However when applied to the 2d-3d case a kind of cross-locking still persists as can be seen in Figure 4.1.2. In order to simulate natural groundwater flow we have considered a cube domain with vertical fractures and a flow driven by the differences in the piezometric head prescribed on the top. No flow boundary conditions have been applied on the rest of the boundary. The conductivities 10^{-5} on fractures and 10^{-8} has been chosen. Enforcing a near equilibrium between the fracture and the continuum pressure by setting $\sigma = 1$ we observe the cross-locking phenomena as the piezometric head fails decay with depth as can be observed in reference solution obtained using the conforming mixed mesh.

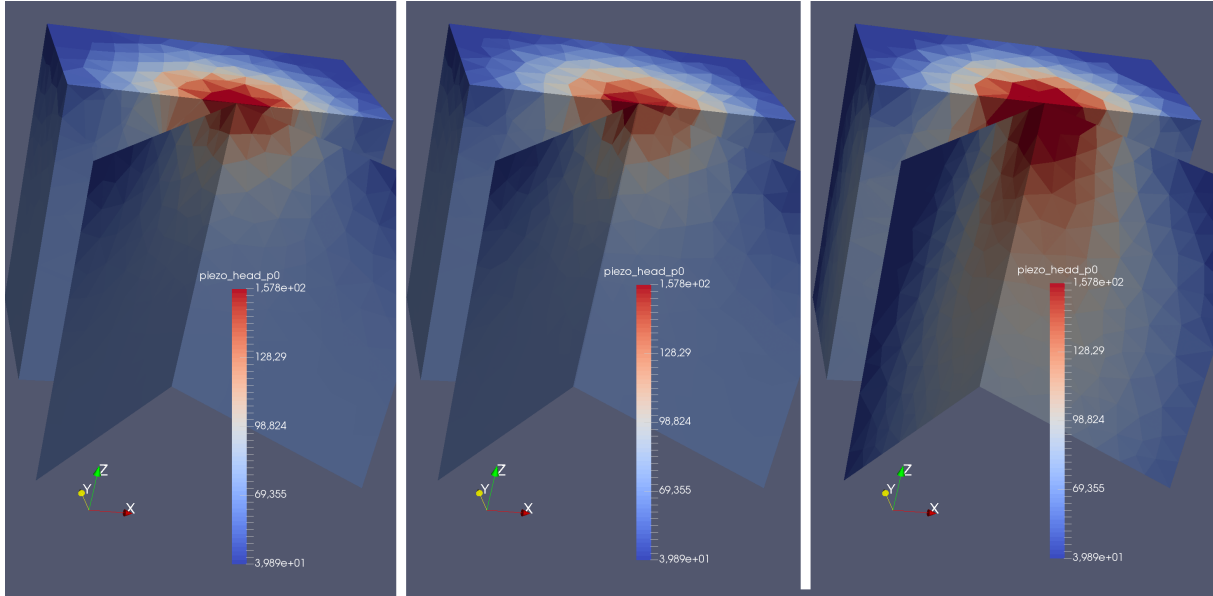


Figure 4.1: Comparison of the conforming (left), non-conforming mortar P^0 (center), and non-conforming direct P^1 (right) coupling. The direct P^1 method exhibits a cross-locking as the piezo head tends to remain constant in vertical direction.

4.1.3 Mortar Like P^0 Method

The locking phenomena is well known in context of mortar methods that are used for non-conforming domain decomposition, see e.g. [8] and [15]. The idea of the mortar methods is to glue solutions on two sides of an interface by introduction of a *mortar space* on the interface,

projecting both solutions to the mortar space and penalize the difference of these projection. If the mortar space is not too rich the locking is avoided and the optimal convergence is obtained.

In the similar way select a suitable mortar space Q_d^m on the fracture and we set R and T operators to be projections to this space. In particular for a P^0 method we use piecewise constant functions Q_d as the mortar space and operators:

$$R(\hat{p}_d)|_K = \bar{p}_K, \quad T(\hat{p}_{d+1})|_K = \frac{1}{\delta_K} \sum_L \delta_L \bar{p}_L.$$

Comparison of the method to the conforming coupling is depicted on Figure 4.1.2. Unlike the direct P^1 coupling it is in good agreement with the reference solution using the conforming coupling.

4.1.4 Mortar Like P^1 Method

Similarly to the previous method we introduce the mortar space as the space of discontinuous piecewise linear functions:

$$Q_d^m = \prod_{L \in \mathcal{T}_d} P^1(L).$$

the R operator is same as in the direct P^1 method:

$$R(\hat{p})|_K = \mathcal{P}_K[\hat{p}_K].$$

As the trace approximation we use the L^2 projection of the traces of \mathcal{P}_L functions on the element K . In particular,

$$T(\hat{p}_{d+1})[\mathbf{x}] = \sum_{i=0}^d \lambda_i \hat{q}_K^i(\mathbf{x})$$

where λ_i are DOFs of the projection to $P^1(K)$ using the non-conforming bases with support points on faces of K . For fixed K the vector λ is a solution to the local system:

$$\mathbb{L}\lambda = \mathbf{b}, \quad L_{ij} = \int_K \hat{q}_K^i \hat{q}_K^j dx, \quad b_i = \sum_L \int_{I_L} \mathcal{P}_L(\hat{p}_L) \hat{q}_K^i dx.$$

4.1.5 Coupling of codimension 0

For codimension 0, we first introduce a numbering \mathcal{S}_d of d dimensional manifolds (2d or 1d fractures), for every intersection line $I_{i,j}$ of two manifolds $i, j \in \mathcal{S}_d$ we define the manifold with smaller number as a *master* manifold, while the other as a *slave* manifold. The intersection curve $I_{i,j}$ of manifolds S_i and S_j , $i < j$ is decomposed into segments corresponding to the elements of the master manifold, i.e.

$$I_{i,j} = \cup_{K \in S_i} I_{K,S_j}$$

With such a notation at our disposal we can write the coupling term as:

$$c_{F,0,d}(h, \hat{h}, q, \hat{q}) = \sum_{I_{i,j}, i < j} \sum_{T \in S_i} \int_{I_{T,S_j}} \sigma_d(R(\hat{h}_i) - T(\hat{h}_j))(R(\hat{\phi}_i) - T(\hat{\phi}_j)),$$

where R is the trace approximation on the master element while T is the trace approximation of the slave manifold, mapping the local discrete spaces of all intersecting slave elements to the discrete space of the master element. Any of the trace approximation operators described in the previous chapters can be used.

4.2 Intersections of non-conforming mixed meshes

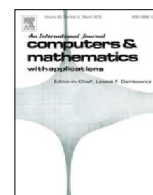
All methods for coupling equations on non-conforming meshes depends on efficient algorithms for computing intersections of such meshes. Following paper deals with this problem presenting a new family of efficient algorithms based on Plücker coordinates. This approach is attractive in combination with a front tracking approach where we can reuse some calculations for neighboring elements.





Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Fast algorithms for intersection of non-matching grids using Plücker coordinates



Jan Březina*, Pavel Exner

Technical University of Liberec, Faculty of Mechatronics, Studentská 1402/2, 461 17 Liberec 1, Czech Republic

ARTICLE INFO

Article history:

Available online 27 February 2017

Keywords:

Non-matching grid
Non-conforming mesh
Mesh intersection
Mixed-dimensional mesh
Plücker coordinates
Advancing front method

ABSTRACT

The XFEM and Mortar methods can be used in combination with non-matching or non-conforming grids to deal with problems on complex geometries. However the information about the mesh intersection must be provided. We present algorithms for intersections between 1d and 2d unstructured multi component simplicial meshes and their intersections with a background unstructured 3d mesh. A common algorithm based on the advancing front technique is used for the efficient selection of candidate pairs among simplicial elements. Bounding interval hierarchy (BIH) of axes aligned bounding boxes (AABB) of elements is used to initialize the front tracking algorithm. The family of element intersection algorithms is built upon a line–triangle intersection algorithm based on the Plücker coordinates. These algorithms combined with the advancing front technique can reuse the results of calculations performed on the neighboring elements and reduce the number of arithmetic operations. Barycentric coordinates on each of the intersecting elements are provided for every intersection point. Benchmarks of the element intersection algorithms are presented and three variants of the global intersection algorithm are compared on the meshes raising from hydrogeological applications.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

The grid intersection algorithms are crucial for several techniques that try to overcome some limitations of the classical finite element method. The Chimera method [1], also called overset grid, and similar Niche method [2] allow solution of the problems with changing geometry as in the fluid–structure problems. The Mortar method [3] allows domain decomposition, independent meshing of domains, and supports sliding boundaries. However our primal motivation is usage of XFEM methods [4] and non-matching meshes of mixed dimension in groundwater models.

The realistic models of groundwater processes including the transport processes and geomechanics have to deal with a complex nature of geological formations containing various small scale features as fractures (or fractured zones) and wells. Although of small scale, these features may have significant impact on the global behavior of the system and their representation in the numerical model is imperative. For example the fractures may form preferential paths which allow much faster transport that cannot be fully captured by equivalent continuum. One possible approach is to model fractures and wells as lower dimensional objects and introduce their coupling with the surrounding continuum. The discretization then leads to the meshes of mixed dimensions, i.e. composed of elements of different dimension. This approach called mixed-dimensional analysis in the mechanics [5] is also studied in the groundwater context, see e.g. [6–8] and already adopted by some groundwater simulation software, e.g. FeFlow [9] and Flow123d [10]. Nevertheless as the complexity of the geometry

* Corresponding author.

E-mail addresses: jan.brezina@tul.cz (J. Březina), pavel.exner@tul.cz (P. Exner).<http://dx.doi.org/10.1016/j.camwa.2017.01.028>

0898-1221/© 2017 Elsevier Ltd. All rights reserved.



increases (e.g. when lots of fractures are randomly generated) the compatible meshing becomes painful or even impossible. In order to avoid these difficulties we may discretize the continuum and every fracture and well independently, getting a non-matching (or incompatible) mesh of mixed dimensions and then apply XFEM to represent jumps of the solution on the fractures or singularities at the wells. The prerequisite for such approach is a fast and robust algorithm for calculating intersections of individual meshes. Although it is (currently) out of our interest, the non-matching mesh approach allows a time evolving network of fractures necessary e.g. in modeling of hydraulic fracturing. We consider a composed mesh \mathcal{T} consisting of simplicial meshes \mathcal{T}_i of dimensions $d_i \in \{1, 2, 3\}$, $i = 1, \dots, N_{\mathcal{T}}$ in the 3d ambient space. We assume that every mesh \mathcal{T}_i is a connected set with no self intersection. Further we assume only single 3d mesh \mathcal{T}_1 . The mesh intersection problem is to find all pairs of elements $L \in \mathcal{T}_i, K \in \mathcal{T}_j, i \neq j$ that have non-empty intersection and to compute that intersection. The mesh intersection problem consists of the two parts: the first, generating a set of candidate pairs (K, L) ; the second, computing the intersection of a particular pair.

According to our knowledge, there are lots of works using non-matching grids, yet only few of them discuss algorithms how to compute their intersections. Gander and Japhet [11] present the PANG algorithm for 2d–2d and 3d–3d intersections that can be used e.g. for mesh overlapping methods. They use the advancing front technique to get candidate pairs in linear time. The algorithm is part of the library DUNE [12]. Massing, Larson, and Logg [2] present an algorithm for 2d–3d intersections as part of their implementation of the Niche method which is part of the project Dolfin [13]. They use axes aligned bounding boxes of elements (AABB) and bounding interval hierarchy (BIH) to get intersection candidate pairs of elements, the GTS library [14] is used for 2d–3d intersections. Finally, there is the work of Elsheikh and Elsheikh [15] presenting an algorithm for 2d–2d mesh union operation which includes calculation and imprinting of the intersection curves. They exploit the binary space partitioning for searching of the initial intersection and the advancing front method for the intersection curve tracking.

In this paper we present a new approach to the mesh intersection problem based on the Plücker coordinates, further developing the algorithm of Platis and Theoharis [16] for ray–tetrahedron intersections. Presented intersection algorithms for pairs of simplicial elements of different dimensions are based on Plücker coordinates. These algorithms are combined with the advancing front method which allows us to reuse Plücker coordinates and their products among neighboring elements and reduce the number of arithmetic operations.

The paper is organized as follows. In Section 2 the algorithms for 1d–2d, 1d–3d and 2d–3d intersections of simplices are described. In Section 3 we discuss our implementation of the advancing front technique and usage of AABB and BIH for its initialization. Finally, in Section 4, we present benchmarks and comparison of individual algorithms.

2. Element intersections

In this section, we present algorithms for computing the intersection of a pair of simplicial elements of a different dimension in the 3d ambient space. In particular we address intersection algorithms for 1d–2d, 1d–3d, 2d–3d pairs of elements. The fundamental idea is to compute intersection of 1d–2d simplices using the Plücker coordinates and reduce all other cases to this one. We have implemented the case 2d–2d as well, however the treatment of all special cases is quite technical and not fully tested yet.

We denote by S_i a simplicial element with $i + 1$ vertices (of dimension i). We call vertices, edges, faces and simplices itself the n -faces and we denote by M_i the set of all n -faces of the simplex S_i . In general, an intersection can be a point, a line segment or a polygon called *intersection polygon* (IP) in common. The intersection polygon is represented as a list of its corners called *intersection corners* (IC). The IP data structure keeps also reference to the intersecting simplices. A data structure of a single IC consists of:

- The barycentric coordinate \mathbf{w}_K of IC on K .
- The dimension d_K of the smallest dimension n -face the IC lies on, e.g. IC on an edge has $d_K = 1$ although it also lies on the connected faces.
- the local index i_K of that n -face on K ,

for each intersecting element K of the pair. The pair $\tau_K = (d_K, i_K)$ is called the *topological position* of the IC on K . Moreover, as every IC is a result of a permuted inner product of some Plücker coordinates (see Section 2.2), we store the sign of the product as well.

2.1. Plücker coordinates

Plücker coordinates represent a line in 3d space. The definition properties and more general context from computational geometry can be found e.g. in [17] or [18]. Considering a line p , given by a point \mathbf{A}_p and its directional vector \mathbf{u}_p , the Plücker coordinates of p are defined as

$$\pi_p = (\mathbf{u}_p, \mathbf{v}_p), \quad \mathbf{v}_p = \mathbf{u}_p \times \mathbf{A}_p.$$

This representation is independent of the choice of \mathbf{A}_p since $\mathbf{u}_p \times (\mathbf{A}_p + t\mathbf{u}) = \mathbf{u}_p \times \mathbf{A}_p$. Further, having two lines p and q with Plücker coordinates π_p and π_q , we denote a permuted inner product by

$$\pi_p \odot \pi_q = \mathbf{u}_p \cdot \mathbf{v}_q + \mathbf{u}_q \cdot \mathbf{v}_p.$$

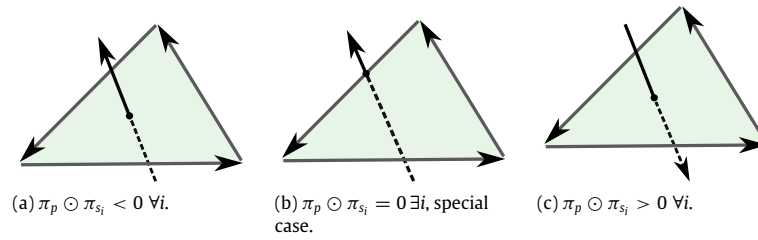


Fig. 1. Different relative positions of the line p and a triangle with sides s_i , $i = 0, 1, 2$. Dashed parts are behind the triangle. Signs of the permuted inner products depend on orientation of lines, the line p is coplanar with a side in the case (b).

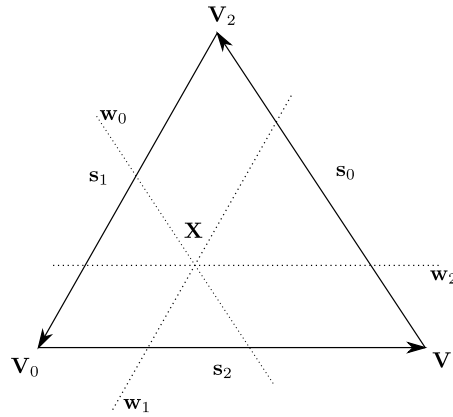


Fig. 2. Notation for Lemma 2.1.

The sign of the permuted inner product is non-zero if p and q are skew lines and is positive if q is oriented counterclockwise and negative if q is oriented clockwise looking in the p direction. This can be used to determine relative position of the line p and the triangle. This is demonstrated in Fig. 1. The permuted inner products of triangle sides with the line p have common sign, cases (a) and (c), if and only if the line intersects the triangle inside. If any $\pi_p \odot \pi_{s_i}$ is zero, as in the case (b), it means that the lines p and s_i are coplanar.

2.2. Intersection line–triangle (1d–2d)

Let us consider a line p with parametric equation

$$\mathbf{X} = \mathbf{A} + t\mathbf{u}, \quad (1)$$

on which a line segment S_1 is defined by $t \in [0, 1]$ and a triangle S_2 given by vertices $(\mathbf{V}_0, \mathbf{V}_1, \mathbf{V}_2)$ with oriented sides $s_i = (\mathbf{V}_j, \mathbf{V}_k)$, $j = (i + 1) \bmod 3$, $k = (i + 2) \bmod 3$, see Fig. 2.

Lemma 2.1. The permuted inner products $\pi_p \odot \pi_{s_i}$, $i = 0, 1, 2$ have the same non-zero sign if and only if there is an intersection point \mathbf{X} on the p and inside the triangle S_2 . The barycentric coordinates of \mathbf{X} on S_2 are

$$w_i = \frac{\pi_p \odot \pi_{s_i}}{w}, \quad w = \sum_{i=0}^2 \pi_p \odot \pi_{s_i}. \quad (2)$$

Proof. Using the barycentric coordinates, the intersection point can be expressed as $\mathbf{X} = \mathbf{V}_0 + w_1\mathbf{s}_2 - w_2\mathbf{s}_1$. The line p has Plücker coordinates $(\mathbf{u}, \mathbf{u} \times \mathbf{X})$ since these are invariant to change of the initial point. Combining these two expressions and substituting for $\mathbf{V}_0 - \mathbf{V}_2 = \mathbf{s}_1$, we get for side \mathbf{s}_1

$$\pi_p \odot \pi_{s_1} = \mathbf{u} \cdot (\mathbf{s}_1 \times \mathbf{V}_2) + \mathbf{s}_1 \cdot (\mathbf{u} \times [\mathbf{V}_0 + w_1\mathbf{s}_2 - w_2\mathbf{s}_1]) = -w_1\mathbf{u} \cdot (\mathbf{s}_1 \times \mathbf{s}_2).$$

Since $\mathbf{s}_0 + \mathbf{s}_1 + \mathbf{s}_2 = 0$ we have $\mathbf{s}_1 \times \mathbf{s}_2 = \mathbf{s}_2 \times \mathbf{s}_0 = \mathbf{s}_0 \times \mathbf{s}_1$ and thus

$$\pi_p \odot \pi_{s_i} = -w_i\mathbf{u} \cdot (\mathbf{s}_1 \times \mathbf{s}_2), \quad (3)$$

$$\sum_{i=0}^2 \pi_p \odot \pi_{s_i} = -\mathbf{u} \cdot (\mathbf{s}_1 \times \mathbf{s}_2). \quad (4)$$

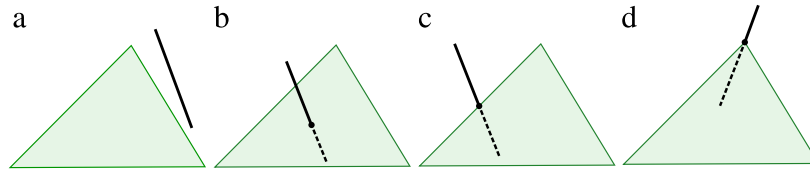


Fig. 3. Some possible cases of the 1d–2d algorithm.

The result (2) then follows directly from combination of (3) and (4). The point \mathbf{X} is inside S_2 if and only if $w_i > 0$ for all $i = 0, 1, 2$. \square

Having the barycentric coordinates of \mathbf{X} on S_2 , we can compute also its local coordinate on p from its parametric form:

$$X_i = A_i + tu_i, \quad \text{for } i = 1, 2, 3. \quad (5)$$

We use i with maximal $|u_i|$ for practical computation.

The calculation of the intersection proceeds as follows:

1. Compute or reuse Plücker coordinates and permuted inner products: $\pi_p, \pi_{s_i}, \pi_p \odot \pi_{s_i}$, for $i = 1, 2, 3$.
2. If the total w of the products is less than $\epsilon L_1 L_2^2$ jump to the coplanar case in the step 8.
3. Compute barycentric coordinates w_i , $i = 1, 2, 3$ using (2).
4. If any w_i is less than $-\epsilon$ (see definition below), there is no intersection, return empty IP. Fig. 3, (a).
5. If all w_i are greater than ϵ , we set $\tau_{S_2} = (2, 0)$ for the IC. Fig. 3, (b).
6. If one w_i is less than ϵ , intersection is on the edge s_i , we set $\tau_{S_2} = (1, i)$. Fig. 3, (c).
7. If two w_i are less than ϵ , intersection is at the vertex V_i , we set $\tau_{S_2} = (0, i)$. Fig. 3, (d).
8. If all w_i are less than ϵ , the line is coplanar with the triangle, both objects are projected to the plane $x_i = 0$ where i is the index of the maximal component of the triangle's normal vector. Every pair p, s_i is checked for an intersection on S_2 boundary either inside s_i or at a vertex V_i , setting the topological info τ_{S_2} to $(1, i)$ or $(0, i)$, respectively. The ICs (at most two of them) obtained in this coplanar case will be called *degenerate* and will later need special treatment.
9. For each IC the barycentric coordinates $(1 - t, t)$ on the line p are computed according to (5).
10. If $t \in (-\epsilon, \epsilon)$ or $t \in (1 - \epsilon, 1 + \epsilon)$, we set the end point of S_1 : $\tau_{S_1} = (0, 0)$ or $\tau_{S_1} = (0, 1)$, respectively.
11. If $t \notin (-\epsilon, 1 + \epsilon)$, the IC is eliminated.

In order to make the test in the step 2 independent of the scale of elements, we use characteristic lengths L_1 and L_2 of S_1 and S_2 respectively. Further, the algorithm depends on the parameter ϵ used as a common tolerance parameter for detection of ICs with special positions. First, it is used in the sign check for permuted inner products, second, it is used for position check on the line. Only these two kinds of geometric predicates are used through the all intersection algorithms. Currently, we use just fixed value $\epsilon = 10^{-9}$. This value is close to the machine epsilon (10^{-16}) of the double precision arithmetic, while far enough to keep precision of the further calculations. Let us note that the algorithm is susceptible to the loss of significance due to cancellation during evaluation of the products. Nevertheless, the algorithm works on all realistic meshes we deal with.

Another problem that would deserve further investigation is possible inconsistent result of two different, but logically related predicates. Adaptive-precision evaluation of the geometric predicates was designed by Shewchuk [19] and used for 2d–2d mesh intersections in [15] in order to deal with these inconsistencies. It is a topic for future work to understand dependency between our geometric predicates and decide if the adaptive-precision is the only way to guarantee the correctness of the algorithm even for various corner cases.

The algorithms for 1d–3d and 2d–3d intersections use simpler version of the 1d–2d intersection algorithm, in particular the search for ICs in the coplanar case (step 8) is not necessary, and the test in the last point is not performed.

2.3. Intersection line–tetrahedron (1d–3d)

In this section we consider an intersection of a line segment S_1 , defined by an interval $t \in [0, 1]$ of the line p defined in (1), with a tetrahedron S_3 . The used algorithm is based on the 1d–2d algorithm and closely follows [16]. Our modification takes into account intersection with the line segment and consistently propagates topological position of ICs.

Algorithm 1 first computes line–face intersections for every face of S_3 . Tetrahedron has six edges, so 7 Plücker coordinates and 6 permuted inner products are computed at most. Precomputed coordinates and products are passed into the 1d–2d algorithm which is performed for the whole line p (line 3). If no IC is found, or coplanar case occurs in line–face computation, we continue with the next face. Note, ICs that would be created in coplanar case are to be found as ICs with the other faces, since they lie on edges. Next, IC can be on an edge or at a vertex; then we set the correct topological position and mark the adjacent faces to be skipped, since there cannot be another IC (and coplanar case has been checked already). Finally at the line 11, we append the IC to the result and check whether the maximal amount of ICs has been reached.

After collecting line–tetrahedron ICs, we do the line segment trimming from the line 13 further. If we have only one IC, we check that it actually lies inside S_1 (otherwise, we throw it away). If we have two ICs, and if both lie outside S_1 , we

Algorithm 1: 1d-3d intersection

Input: Line segment S_1 of line p , Tetrahedron S_3 .
Output: List I of ICs sorted along p .

```

1  $I = \{\}$ 
2 for unmarked face  $f$  of  $S_3$  do
3    $L = \text{intersection}(p, f)$ 
4   if  $L$  is none or degenerate then continue
5   if  $L$  is inside the edge  $e$  then
6     set  $\tau_{S_3} = (1, e)$ 
7     mark faces incident to  $e$ 
8   else if  $L$  is at the vertex  $v$  then
9     set  $\tau_{S_3} = (0, v)$ 
10    mark faces incident with  $v$ 
11   append  $L$  to  $I$ 
12   if  $|I| = 2$  then break
13 if  $|I| = 1$  and  $I$  is outside of  $S_1$  then erase  $I$ 
14 else if  $|I| = 2$  then
15   trim intersection with respect to the line segment  $S_1$ 

```

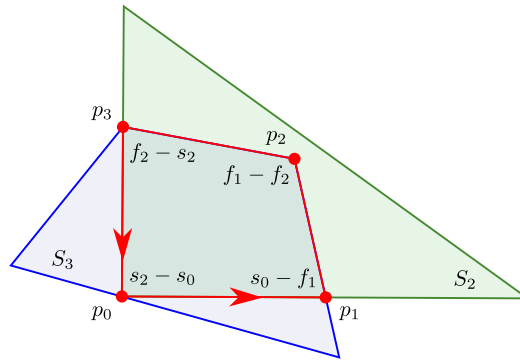


Fig. 4. An example of an intersection 2d-3d, demonstrating the ICs ordering. We see at every intersection polygon corner p_i which n -faces it lies on. Looking at p_0 , the connection table entries are: $F_c[s_2] = p_0$, $F_f[p_0] = s_0$. For the other ICs we have: $F_c[s_0] = p_1$, $F_f[p_1] = f_1$, $F_c[f_1] = p_2$, $F_f[p_2] = f_2$ and $F_c[f_2] = p_3$, $F_f[p_3] = s_2$.

eliminate both of them. If one of the ICs lies out of S_1 , we use the closest end point of the line segment instead and interpolate barycentric coordinates of the IC on S_3 . The topological positions τ_{S_1} and τ_{S_3} are updated as well. The result of the algorithm is 0, 1, or 2 ICs, sorted by the parameter t in the direction of the line p .

2.4. Intersection triangle-tetrahedron (2d-3d)

The intersection of a triangle S_2 and a tetrahedron S_3 is an n -side intersection polygon (IP), $n \leq 7$. The sides of the polygon lie either on sides of S_2 or on faces of S_3 . Thus each vertex (IC) of the polygon can arise either from side-face intersection, or from edge-triangle intersection, or be a vertex of S_2 . To get all ICs, we have to compute at most 12 side-face intersections and at most 6 edge-triangle intersections. However, to this end we only need to compute 9 Plücker coordinates (3 sides, 6 edges) and 18 permuted inner products, one for every side-edge pair. Computation of IP consists of three stages: calculation of side-tetrahedron ICs (Section 2.4.2), calculation of edge-triangle ICs (Section 2.4.3), ordering of ICs (Section 2.4.5).

2.4.1. Successor tables

The intersection corners are appended to the list I as they are computed, however their order on IP is given indirectly by the successor tables $F_c[\cdot]$ and $F_f[\cdot]$. Every side of IP that lies on n -face $x \in M_2 \cup M_3$ is followed by an IC given by $F_c[x]$ and every IC p is followed by the side of IP that lies on the n -face $y = F_f[p] \in M_2 \cup M_3$ (see Fig. 4). After an IC p is computed, we also obtain two n -faces x, y incident with the two IP's sides that are neighboring with the IC. Order of the n -faces x, y have to match the orientation of the IP which is the same as the orientation of S_2 triangle, that is counterclockwise around the interior with normal pointing to us. Having x, y in right order, we set $F_c[x] = u$, $F_f[u] = y$.

This simple approach works well even for most of the degenerated ICs, however in order to deal with some special cases and with duplicity of ICs in vertices, we further mark by a backlink $F_c[y] = u$ the n -faces that succeed some IC but still do

not possess its successors. If y already has the backlink we swap x and y . The result is the *set links* (SL) operation formalized in Algorithm 2.

Algorithm 2: 2d-3d intersection, set links

```

Input:  $n$ -face  $x$ , IC  $p$ ,  $n$ -face  $y$ 
1 if  $F_f[F_c[y]] = y$  then
2   | swap  $x$  and  $y$                                      //  $y$  success an IC already
3  $F_c[x] = u, F_f[u] = y$ 
4 if  $F_c[y]$  is unset then  $F_c[y] = u$ 
  
```

2.4.2. Intersections on sides of triangle

Algorithm 3 computes all ICs on the boundary of S_2 . It passes through the sides of the triangle S_2 computing the line–tetrahedron intersection L for every side s . In the regular case ($|L| = 2$), we process each IC in L (line 4). The IC p is appended to I and successor tables are set using the SL operation. If p is at the vertex of S_2 the links connect the vertex with the S_2 side. In both cases SL is called with the side s as the target n -face since SL correctly swaps n -faces if the side is already used as the target. The vertex ICs are put twice into I and are merged in the final step.

The case $|L| = 1$ can happen only if the boundary of S_2 touches the boundary of S_3 . These ICs will be rediscovered again in Algorithm 4 with better topological information, however this is not the case if the touched edge e of S_3 is coplanar with S_2 and the IC is inside of e . To this end we call SL with e as the target which allows to use backlink and get already computed IC if it is rediscovered later on. The ICs at vertices of S_2 are treated differently, but follows the same idea. The ICs at vertices of S_3 are skipped.

Algorithm 3: 2d-3d intersection, ICs on sides of S_2

```

Input:  $S_2$  and  $S_3$ 
Output: List  $I$  with ICs on sides of  $S_2$ 
1  $F_c(\cdot) = -1, F_f(\cdot) = -1$                                      // Unset links.
2 for side  $s$  of  $S_2$  do
3   |  $L = \text{intersection}(s, S_3)$ 
4   | for  $p$  in  $L$  do
5     |  $p$  lies on  $n$ -face  $x \in M_2$  and  $y \in M_3$ 
6     | if  $|L| = 1$  then
7       | | deal with special case                               // side  $s$  touching  $S_3$ 
8       | append  $p$  to  $I$ 
9       | if  $x$  is the vertex of  $S_2$  then
10        | | set links( $x, p, s$ )
11        | else
12        | | set links( $y, p, x$ )                                //  $x$  is  $s$ 
  
```

2.4.3. Intersections on edges of tetrahedron

Algorithm 4 uses the line–triangle intersection algorithm for the edges of S_3 . First, the intersection $L[e]$ is evaluated for every edge e (line 1). Then we pass through once again skipping the edges with none or degenerate IC. For every intersection corner $p = L[e]$, we first get n -faces that would appear before and after the IC on IP.

The function *edge faces* (line 4) returns the adjacent faces f_0, f_1 to the edge e on which the IC lies (see the situation in Fig. 5). The faces are sorted using the sign of the permuted inner product in 1d–2d intersection. The order of faces matches the order of sides of IP if the sign is negative. If the sign is positive the function *edge faces* returns face pair (f_1, f_0) . If the IC is at the vertex v of S_3 , the function *vertex faces* described later (Algorithm 5) is used. It returns a pair of n -faces (face or edge) adjacent to the IC $L[e]$ at the vertex v of S_3 . Then p is appended to I . If IC p is inside S_2 , the obtained pair of n -faces is directly used to set links (line 16). However, if p is on the boundary of S_2 (n -face x), just one of the faces is used, complemented with x . Presence of the backlink is used to determine the right face.

2.4.4. Vertex faces algorithm

The function in Algorithm 5 gets as a parameter IC p at the vertex v of S_3 which is a special case of a non-degenerate edge–triangle intersection. There are three edges incident with the vertex v , results $s[i]$ of their intersections with S_2 may be one of: no IC, degenerate IC, positive IC and negative IC. Accordingly we say the edge is: without intersection, degenerate,

Algorithm 4: 2d-3d intersection, ICs on edges of S_3

Input: I with ICs on S_2 boundary, partially filled F_f, F_c
Output: all ICs in I , complete F_f, F_c

```

1 for edge  $e$  of  $S_3$  do  $L[e] = \text{intersection}(e, S_2)$ 
2 for edge  $e$  of  $S_3$  with regular  $L[e]$  do
3    $p = L[e]$ 
4   if  $p$  is inside  $e$  then
5      $(f_0, f_1) = \text{edge faces}(e)$ 
6   else  $p$  at the vertex  $v$  of  $S_3$ 
7      $(f_0, f_1) = \text{vertex faces}(v, L)$  // Algorithm 5
8   append  $p$  to  $I$ 
9   if  $p$  is on the boundary of  $S_2$  then
10     $p$  lies on edge or at vertex  $x \in M_3$ 
11    if  $x$  have backlink then
12      set links( $x, p, f_1$ )
13    else
14      set links( $f_0, p, x$ )
15  else
16    set links( $f_0, p, f_1$ )

```

Algorithm 5: 2d-3d intersection, vertex faces

Input: vertex v of S_3 , $L[:]$ intersection results for edges of S_3
Output: pair of n -faces incident with v that is intersected by the plane of S_2

```

1  $e_0, e_1, e_2$  edges incident with  $v$  oriented out of  $v$ 
2  $s[i] = L[e_i]$ , for  $i = 0, 1, 2$ ,
3 if  $s[:]$  contains 1 degenerate edge  $e$  then
4   Let  $f$  be the face opposite to  $e$ .
5   if other two edges  $e_a, e_b$  have different sign then
6      $z = \text{EdgeFaces}(e_a)$ 
7     replace  $g \in z, g \neq f$  with  $e$ , return  $z$ 
8   else return  $(v, e)$ 
9 else if  $s[:]$  contains 1 non-degenerate edge  $e$  then
10  return pair of degenerate edges sorted according to  $\text{EdgeFaces}(e)$ 
11 else if  $s[:]$  contains edge  $e$  with the sign opposite to the other two then
12  return  $\text{EdgeFaces}(e)$ 
13 else  $s[:]$  have all signs same
14  return  $(v, v)$ 

```

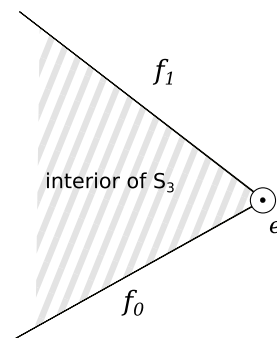


Fig. 5. Order of faces adjacent to the oriented edge e pointing towards us.

positive, or negative. We use these edge indicators to return generalized faces of S_3 preceding and succeeding p on the polygons boundary assuming p is at interior of S_2 . Possible cases are (see also Fig. 6(a)–(e)):

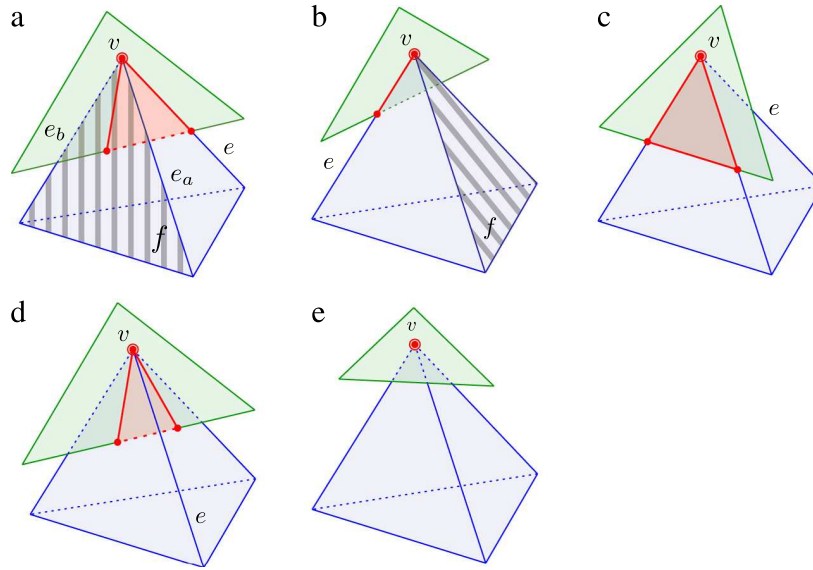


Fig. 6. Possible cases processed in the function *vertex_faces*. Only the main features referred in text are denoted: tetrahedron vertex v , edge e , face f (stripes).

- *Single degenerate IC* (line 3). Let us denote e as the edge with degenerate IC and f the face between the other two edges. The other two (non-degenerate) edges may have either the opposite sign (the plane is cutting S_3 , see Fig. 6, (a)) or the same sign (the plane is touching S_3 at the edge e , see Fig. 6, (b)). In the first case, the call *edge_faces*(e) returns (f_x, f) or (f, f_x) , then the vertex faces function returns (e, f) or (f, e) , respectively. In the second case, there must be another IC on e , either at S_2 boundary or at the other end of e . In both cases the edge e is the common n -face of the two intersection points thus we return (v, e) taking the edge as the target object.
- *Two degenerate ICs* (line 9). A face of S_3 lies in the plane of S_2 , see Fig. 6, (c). Let e be the single non-degenerate edge. We treat the two degenerate edges as faces adjacent to e and return them sorted like the faces given by *edge_faces* of edge e .
- *Single IC has the opposite sign to the other two* (line 11). Let e be the edge of the single IC with the different sign. The plane of S_2 separates e from the other two edges so it goes through the faces adjacent to e , see Fig. 6, (d). The order is determined by the function *edge_faces* called for the edge e .
- *All ICs have the same sign* (line 13). Since S_2 is touching S_3 at the vertex v , Fig. 6, (e), the polygon degenerates into a point and thus no connection information is necessary. We just return (v, v) .

2.4.5. Ordering of intersections

The final stage of the 2d–3d intersection is ordering of ICs. We start with the first IC in I and follow the successor tables until we return back. The ICs are copied into the result vector, skipping the duplicities. Special treatment must be done for degenerate cases with less than 3 ICs as they may not form a cycle.

3. Global mesh intersection algorithm

Having the algorithms for element–element intersections at our disposal we can proceed to the mesh intersection algorithm. We consider the composed mesh \mathcal{T} containing the 3d mesh \mathcal{T}_1 that we shall call a bulk mesh \mathcal{T}_b . Any other mesh \mathcal{T}_i , with dimension $d_i < 3$, $i = 1, \dots, N_{\mathcal{T}}$, we shall call a component mesh. We first compute all component–bulk mesh intersections that is (1d–3d and 2d–3d) using the advancing front algorithm which we shall describe in Sections 3.1, 3.2 and then the 1d–2d and 2d–2d intersections are computed using the bulk mesh to get the intersection pair candidates. This step is described in Section 3.3.

Let us consider a single pair of the component mesh \mathcal{T}_c and the bulk mesh \mathcal{T}_b . Element intersections for this pair of meshes are obtained in two phases: firstly, we look for the first pair (c, b) of the component and the bulk element with a non-empty intersection (the initialization); secondly, we prolong the intersection by investigating neighboring elements (the front tracking).

3.1. Initialization

Given a component element c , we have to find an intersecting bulk element b . If this step is done only few times the optimal way is to iterate over the bulk mesh and test every element for the intersection. This process may be accelerated

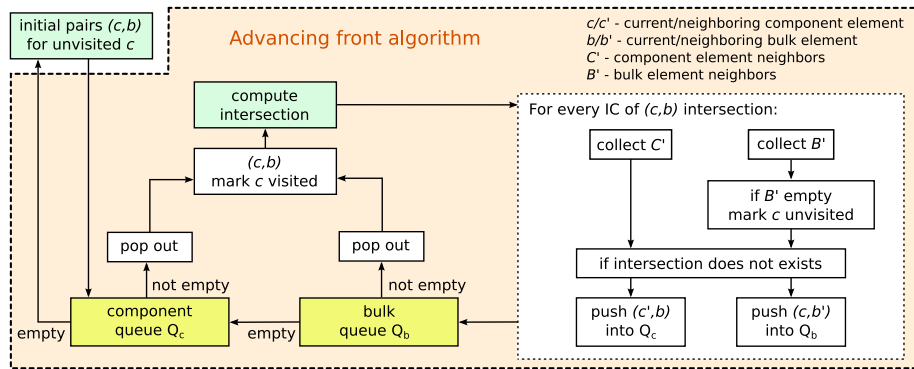


Fig. 7. Advancing front algorithm for 1d–2d and 2d–3d intersections.

using the axis aligned bounding box (AABB) for every element and use intersection of the bounding boxes as a fast indicator for possible intersection of the elements. This step takes time $O(N)$ with respect to the number of elements of the bulk mesh N . If the number of components k is small and if the components are contained inside the bulk mesh, the total time of the initialization may still be linear $O(kN)$. However, for more complex cases we organize the bounding boxes of the bulk mesh into the bounding interval hierarchy (BIH) [20] a data structure in principle equivalent to the R-trees [21,22]. The construction of a BIH takes time $O(N \log(N/n))$ and the search time is $O(\log(N/n))$ where n is the number of the bulk elements in the leaf nodes of the tree.

3.2. Advancing front method

The advancing front algorithm requires the neighboring information for the elements within the component mesh \mathcal{T}_c as well as within the bulk mesh \mathcal{T}_b . It can be viewed as the breadth first search algorithm for a graph where the graph vertices are the intersection polygons and the graph edges are the sides shared by two polygons. Since every side of IP is on the boundary of either the component element or the bulk element, we can distinguish bulk and component edges. Correspondingly we use a *component queue* Q_c and a *bulk queue* Q_b , where we shall place intersection candidate pairs (c, b) . In order to process every pair (c, b) only once, we check if the pair was already processed before it is enqueued into one of the queues. Only if the pair was not yet processed we mark it processed and push it into the queue. Since the number of possible pairs is too big we cannot have a flag array which may allow constant time checks. Therefore, we keep a hash table of the processed pairs which allows the constant check in the average.

The key idea behind the two queues is to compute intersections for a component element with all possible bulk elements at first, and then move to a next neighboring component element. So the bulk queue is emptied before the component queue.

First, we mark all component elements $c \in \mathcal{T}_c$ as unvisited. For every unvisited element $c \in \mathcal{T}_c$, we find some intersection candidate pairs $\{(c, b), b \in \mathcal{T}_b\}$ and into the queue Q_c . Then we increment the *component number* γ , which we use to mark all intersection polygons we shall find until the queue Q_c becomes empty. This way, we shall later know to which component a given IP belongs to, which will become important in Section 3.3. This is from where the front tracking starts, see the top-left corner of the scheme in Fig. 7.

We dequeue the first candidate pair (c, b) from Q_c and compute the IP. If the intersection exists, we look for the new candidate pairs among the neighboring elements (see the big white block in Fig. 7). Therefore, we iterate over ICs of the IP and further exploit their topological position on the component element c and the bulk element b . For every IC one or both of the following cases may happen:

- (a) IC is on the boundary of c and inside b .

We find all the sides S of c incident with the n -face of c on which IC lies. Then we get all component elements C' neighboring with c over any side $s \in S$. And finally, we push all pairs (c', b) , $c' \in C'$ into the component queue. Note that c can have more than one neighbor component elements over the single side s , i.e. branches are allowed.

- (b) IC is inside c and on the boundary of b .

We find all the faces F of b incident with the n -face of b on which IC lies. Then we get all bulk elements B' neighboring with b over any face $f \in F$, analogically to the previous case. Finally, we push the new candidate pairs (c, b') , $b' \in B'$ into the bulk queue. However, if the list B' is empty, which means that the component element c pokes out of the bulk mesh, we mark the element c as unvisited again. This way we have a chance to find possible other intersections of the element c with the bulk mesh in the main loop. Note that every time this happens, the possible further intersection of the current c will be seen as different component with increased component number m (see an example situation in Fig. 8).

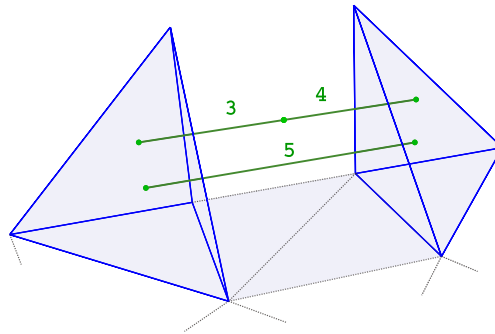


Fig. 8. For a non-convex bulk domain a situation as this may happen. The 1d elements 3, 4, 5 extend out of the bulk mesh. Therefore four initializations are needed, to find all four 1d–3d intersections every one forming an independent component. Advancing front method cannot play any part in this situation.

We see that (c, b') can prolong the intersection over a bulk element face, on the other hand (c', b) may prolong the intersection over the component side. If the IC lies both on the boundaries of c and b , we obtain candidate pairs of both types. Having all ICs processed, we continue emptying the queues. We empty the bulk queue first, trying to fully cover the current component element c before we proceed to the next one.

3.3. Intersections between component meshes

We consider here the situation where components are in the interior of the 3d bulk mesh. After we compute all component–bulk intersections, we use it to easily find all the component–component intersection candidate pairs. If the bulk element intersects more than one component element, then we look for candidate pairs only among these.

Let us start with the description of how we store the intersection results, which will be of great importance here. For each element intersection, we save the following data: references to the component and bulk element, the barycentric coordinates on both and the index of the component. These objects are stored in separate vectors for each pair of dimensions. Further we define a matrix (*intersection map*) which has as many rows as there are elements in the mesh. At each row, we save the references to all other elements, having intersection with the element corresponding to this row, and references to the actual intersection data.

The algorithm for 2d–2d intersections works as follows. We iterate over all 2d–3d intersections, in fact over the bulk elements having some intersections with 2d components. We look at the intersection map at the bulk element row and collect all elements that have 2d–3d intersection with it. Then we create all possible pairs from the collected component elements. Now comes into play the component number γ . If the elements of a single pair have γ equal, then these are part of a single continuous component and we do not compute any intersection. Otherwise we obtain a new candidate pair, for which IP can be computed.

The algorithm for 1d–2d is analogical, only we do not have to check the component number. Note that this way, we do not obtain any intersection in the exterior of bulk mesh. If such problem is of our interest, we find the candidate pairs using the search algorithms as in initialization phase of advancing front method.

4. Benchmarks

In this section, we present numerical results on several benchmark problems. In the beginning, we provide some software development related information. Then we analyze the theoretical number of FLOPS in our element intersection algorithms with other state of art algorithms. Next we compare our algorithms with different initialization phase (candidate pairs search), and using the advancing front method or not. We show the results both on a mesh of a real locality and an artificial mesh.

4.1. Software Flow123d

The implementation of the presented algorithms is part of the open source software Flow123d [10]. The version used for this paper is tagged as *intersections_paper* in the main project repository at GitHub. The C++ is used as principal language. Flow123d provides models for saturated and unsaturated groundwater flow, solute transport with sorption, and heat transfer. Mixed finite elements, finite volumes, and discontinuous Galerkin method are used. All equations are consistently formulated for meshes of mixed dimensions. However, non-matching meshes are at the moment available only in experimental 1d–2d flow model, using mortar like coupling.

Table 1

Raw number of FLOPs used by different intersection algorithms. Second row contains estimated effective number of FLOPs per intersection accounting for reuse of edges through the mesh intersection assuming edges of 2d and 3d elements are used twice (conservative).

Algorithm	1d–2d	1d–3d	2d–3d
Plücker	92	198	426
Plücker (edge reuse)	42	138	264
Haines	51	177	469
Möller and Trumbore	42	168	756

4.2. Theoretical comparison

It proved to be a bit problematic to compare the presented algorithms for element–element intersections to any of the state of art algorithms e.g. from the field of computer graphics. The algorithms for computer graphics applications are specialized for the 1d–2d and 1d–3d cases and they provide different output information than our algorithms. Moreover our implementation is not yet fully optimized to be on par with the fine tuned implementations. Instead, we present a theoretical comparison in terms of estimated number of floating point operations (FLOPS) performed by individual algorithms. As the intersection algorithms work on small data they should not be limited by the memory access, thus such comparison may be realistic.

We consider 3 algorithms for the line–triangle intersections: Plücker algorithm described in Section 2.2, the algorithm based on the plane clipping due to Haines [23], and the minimum storage algorithm due to Möller and Trumbore (MT) [24]. For the later two algorithms we have considered straightforward modifications to make them return qualitatively same output as our algorithms for 1d–2d, 1d–3d, and 2d–3d cases. Estimated numbers of FLOPS for all cases are summarized in Table 1. For the Plücker, we count FLOPS actually made by the implementation of individual intersection algorithms. For Haines and MT, we estimated number of FLOPS in theoretical implementations. In particular we account for reuse of the calculations in Plücker and Haines algorithms. Conclusions from this census are: algorithms based on the Plücker coordinates should be competitive with state of art algorithms in the case of 1d–3d and 2d–3d intersections. The expected performance for the 1d–2d case seems to be poor however these intersections are computed after 1d–3d and 2d–3d so the Plücker coordinates may be reused. Considering this scenario we get quite competitive 45 FLOPS. Similarly we may expect better times in remaining two intersection cases when the Plücker coordinates and their products are reused by neighboring elements.

4.3. Global mesh intersections

The global mesh intersection algorithm for a composed mesh \mathcal{T} presented in Section 3 has been implemented in three variants. First variant uses a full search (FS) over the bulk mesh to get the initial pair for the advancing front algorithm (AF). Second variant uses the BIH to accelerate the initialization of the AF algorithm. Third variant does not use AF at all and relies only on the BIH. In this section, we compare these three variants (FS+AF, BIH+AF, BIH) on one artificial composed mesh and one mesh raising from a real hydrogeological simulation.

The artificial composed mesh consists of a cube and two diagonal rectangular 2d meshes (see Fig. 9). The bulk mesh A sequence of meshes was prepared with increasing number of elements ranging from 33 up to 2000 thousands of elements. The mesh step for the bulk mesh was always about half the mesh step of the component mesh. The number of the bulk–component intersections varies from 0.1 up to 2.0 million. The timing for the three compared variants of the mesh intersection algorithm is shown in Fig. 10. Every algorithm consists of the initialization phase which processes all elements of the mesh and the intersection phase which depends only on the number of elements in the component mesh. In these terms both phases of all three variants exhibit almost linear time complexity. As the number of component meshes is low and they are completely inside of the bulk mesh the FS+AF variant is the fastest in particular due to fast initialization. On the other hand the BIH variant is about two times slower than the BIH+AF variant during the intersection phase. That is roughly related to the average fraction of the non-intersecting 3d element in the bounding box of a 2d element.

Next, we study the performance of the intersection algorithms on a mesh of a real problem, see Fig. 11. The mesh represents a mountain ridge above a water supply tunnel in Bedřichov in the Jizera mountains. The mesh includes a system of geological fractures (Fig. 11(a)). In order to study influence of the component elements poking out of the bulk mesh we also prepared a mesh with the artificially extended fractures (Fig. 11(b)). Each of these meshes contains 28 fractures however some of them are connected in the conforming way so there are 9 separated 2d components and a single 1d component, the tunnel.

The results for both meshes can be seen in Fig. 12, pay attention to the different time scales in the graph. In the first case, we notice that FS+AF and BIH+AF algorithms are nearly twice as fast as *BIHonly*. The fraction of the non-intersecting 3d elements in bounding boxes of the 2d elements is higher as the 3d elements are on average smaller than the 2d elements. Creation of the BIH in the BIH+AF variant pays off and the algorithm performs better than the FS+AF variant. This in the contrast to the cube test case since the number of the component meshes is higher.

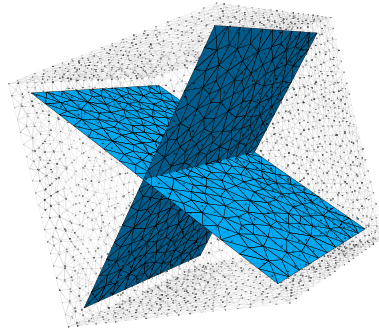


Fig. 9. Artificial mesh — a cube with two perpendicular planes placed on the diagonals of the cube. The planes are also non-matching, therefore can be seen as two independent components.

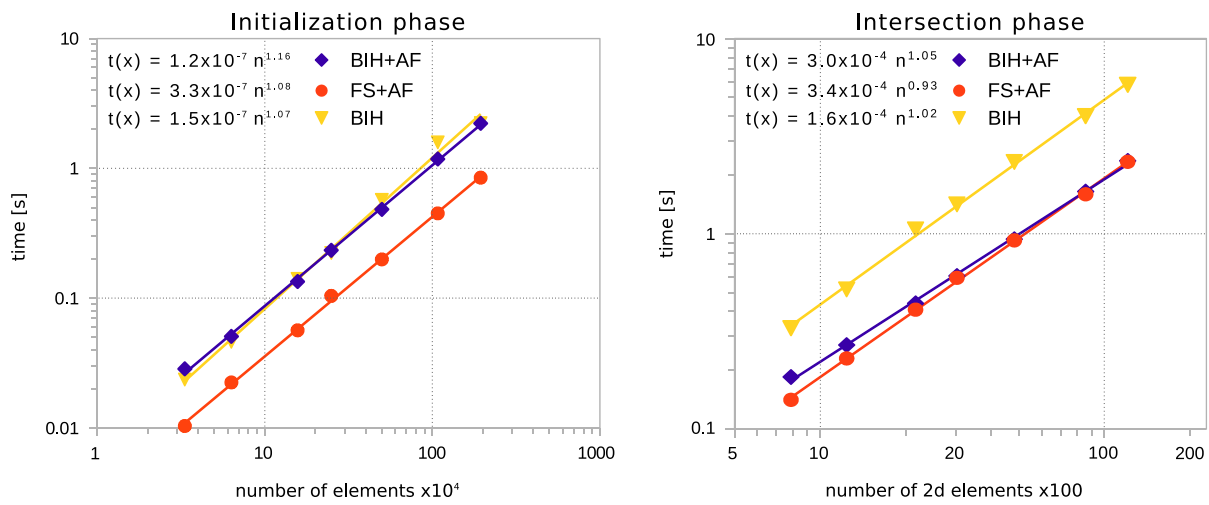


Fig. 10. Time complexity for the initialization phase (left) with respect to total mesh size and the intersection algorithm (right) with respect to the size of the component mesh.

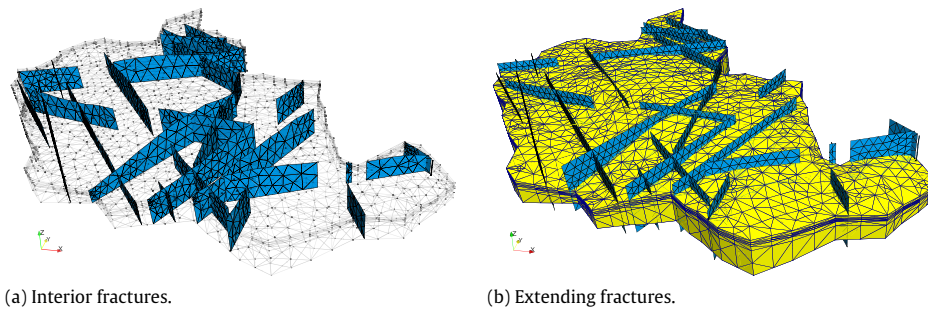


Fig. 11. A mesh of the real locality of Bedřichov in the Jizera mountains. We see fractures inside the bulk mesh in the left figure, fractures are extending the bulk mesh.

In the second case, we observe large blow up for the FS+AF variant. It is caused by the exterior component elements, for which all the bulk elements bounding boxes are iterated before the algorithm concludes there is no intersection. This case is clearly treated by both BIH and BIH+AF variants much better.

5. Conclusions

We present a family of the algorithms for computing intersection polygons for pairs of simplicial elements. The algorithms are based on the Plücker coordinates of the edges which may be reused between elements. A unified algorithm for the intersections of the meshes of a composed mesh was demonstrated. All algorithms were tested and compared on a set of

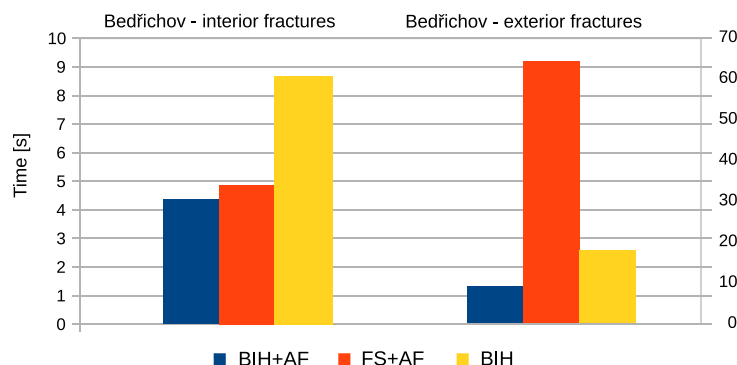


Fig. 12. Comparison of the algorithms on meshes of Bedřichov locality – interior fractures on the left, extending fractures on the right.

benchmark problems. In the near future, we want to perform an optimization of the algorithms and in particular use them in XFEM and Mortar like methods for problems in porous media.

Acknowledgments

The paper was supported in part by the Project OP VaVPI Centre for Nanomaterials, Advanced Technologies and Innovations CZ.1.05/2.1.00/01.0005.

The work of P. Exner was supported by the Ministry of Education of the Czech Republic within the SGS Project No. 21176/115 of the Technical University of Liberec.

References

- [1] F. Brezzi, J.-L. Lions, O. Pironneau, Analysis of a chimera method, *C. R. Acad. Sci., Paris I* 332 (7) (2001) 655–660. [http://dx.doi.org/10.1016/S0764-4442\(01\)01904-8](http://dx.doi.org/10.1016/S0764-4442(01)01904-8).
- [2] A. Massing, M.G. Larson, A. Logg, Efficient implementation of finite element methods on nonmatching and overlapping meshes in three dimensions, *SIAM J. Sci. Comput.* 35 (1) (2013) C23–C47. <http://dx.doi.org/10.1137/11085949X>.
- [3] F.B. Belgacem, The mortar finite element method with lagrange multipliers, *Numer. Math.* 84 (2) (1999) 173–197. <http://dx.doi.org/10.1007/s002110050468>.
- [4] T.-P. Fries, T. Belytschko, The extended/generalized finite element method: An overview of the method and its applications, *Internat. J. Numer. Methods Engrg.* 84 (3) (2010) 253–304. <http://dx.doi.org/10.1002/nme.2914>.
- [5] S. Bournival, J.-C. Cuillière, V. François, A mesh-geometry based approach for mixed-dimensional analysis, in: *Proceedings of the 17th International Meshing Roundtable*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 299–313. http://dx.doi.org/10.1007/978-3-540-87921-3_18.
- [6] V. Martin, J. Jaffré, J.E. Roberts, Modeling fractures and barriers as interfaces for flow in porous media, *SIAM J. Sci. Comput.* 26 (5) (2005) 1667–1691. <http://dx.doi.org/10.1137/S1064827503429363>.
- [7] A. Fumagalli, A. Scotti, Numerical modelling of multiphase subsurface flow in the presence of fractures, *Commun. Appl. Ind. Math.* 3(1) <http://dx.doi.org/10.1685/journal.aim.380>.
- [8] J. Březina, J. Stebel, Analysis of model error for a continuum-fracture model of porous media flow, in: *High Performance Computing in Science and Engineering*, in: *Lecture Notes in Computer Science*, no. 9611, Springer International Publishing, 2015, pp. 152–160. http://dx.doi.org/10.1007/978-3-319-40361-8_11.
- [9] M.G. Trefry, C. Muffels, FEFLOW: A finite-element ground water flow and transport modeling tool, *Ground Water* 45 (5) (2007) 525–528. <http://dx.doi.org/10.1111/j.1745-6584.2007.00358.x>.
- [10] J. Březina, J. Stebel, P. Exner, J. Hybš, Flow123d, <http://flow123d.github.com>, repository: <http://github.com/flow123d/flow123d> (2011–2016).
- [11] M.J. Gander, C. Japhet, Algorithm 932: PANG: Software for nonmatching grid projections in 2d and 3d with linear complexity, *ACM Trans. Math. Software* 40 (1) (2013) 1–25. <http://dx.doi.org/10.1145/2513109.2513115>.
- [12] P. Bastian, M. Droske, C. Engwer, R. Klöforn, T. Neubauer, M. Ohlberger, M. Rumpf, Towards a unified framework for scientific computing, in: *Domain Decomposition Methods in Science and Engineering*, in: *Lecture Notes in Computational Science and Engineering*, no. 40, Springer, Berlin Heidelberg, 2005, pp. 167–174. http://dx.doi.org/10.1007/3-540-26825-1_13.
- [13] A. Logg, G.N. Wells, J. Hake, Dolfin: a c++/python finite element library, in: *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 173–225. http://dx.doi.org/10.1007/978-3-642-23099-8_10.
- [14] Gts, gnu triangulated surface library, software package, <http://gts.sourceforge.net/>.
- [15] A.H. Elsheikh, M. Elsheikh, A reliable triangular mesh intersection algorithm and its application in geological modelling, *Eng. Comput.* 30 (1) (2012) 143–157. <http://dx.doi.org/10.1007/s00366-012-0297-3>.
- [16] N. Platis, T. Theoharis, Fast ray-tetrahedron intersection using plucker coordinates, *J. Graph. Tools* 8 (4) (2003) 37–48. <http://dx.doi.org/10.1080/10867651.2003.10487593>.
- [17] L. Dorst, D. Fontijne, S. Mann, *Geometric Algebra for Computer Science*, first ed., Morgan Kaufmann, Amsterdam, San Francisco, 2007.
- [18] M. Joswig, T. Theobald, *Polyhedral and Algebraic Methods in Computational Geometry*, Springer, 2013, URL <http://www.springer.com/us/book/9781447148166>.
- [19] J. Richard Shewchuk, Adaptive precision floating-point arithmetic and fast robust geometric predicates, *Discrete Comput. Geom.* 18 (3) (1997) 305–363. <http://dx.doi.org/10.1007/PL00009321>.
- [20] C. Wächter, A. Keller, Instant ray tracing: The bounding interval hierarchy, in: *Proceedings of the 17th Eurographics Conference on Rendering Techniques*, EGSR '06, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2006, pp. 139–149. <http://dx.doi.org/10.2312/EGWR/EGSR06/139-149>.
- [21] A. Guttman, R-trees: A dynamic index structure for spatial searching, *SIGMOD Rec.* 14 (2) (1984) 47–57. <http://dx.doi.org/10.1145/971697.602266>.

- [22] B. Nam, A. Sussman, A comparative study of spatial indexing techniques for multidimensional scientific datasets, in: Proceedings of the 16th International Conference on Scientific and Statistical Database Management, SSDBM'04, IEEE Computer Society, 2004, p. 171. <http://dx.doi.org/10.1109/SSDBM.2004.1>.
- [23] E. Haines, Fast ray-convex polyhedron intersection, in: Graphics Gems II, Academic Press, San Diego, 1991, pp. 247–250.
- [24] T. Möller, B. Trumbore, Fast, minimum storage ray-triangle intersection, J. Graph. Tools 2 (1) (1997) 21–28. <http://dx.doi.org/10.1080/10867651.1997.10487468>.





Chapter 5

Unsaturated Darcy flow on mixed meshes

Single phase unsaturated Darcy flow is modeled by the Richards' equation, where the water content and the conductivity are strongly non-linear functions of the pressure. In particular the equation degenerate from a non-linear parabolic equation to an elliptic equation as the pressure tends to zero, i.e. to the saturated state. On the other hand the existence and uniqueness as well as basic regularity properties were proved by ALT, and LUCKHAUS [3] for a wide range of realistic constitutive relations. More over a monotonicity of the equation was proved in [18]. This is in contradiction to the observed fingering phenomena and suggests that phenomenological constitutive relations are not realistic at least in some corner cases.

As we show in Section 5.1 the standard discretization of the source term (including the time term) is unstable due to violation of the discrete maximum principle. Next Section 5.5 presents application of lumping technique to maintain stability. In the final Section 5.6, we present the dual permeability model (due to GERKE and GENUCHTEN [5] and its solution by fully coupled solver. This solver was also used to show instability of RT elements in Section 5.1, but without fractures. The dual permeability model presents use a dual continuum model where the matrix and the fractures are represented by two artificial phasis coupled by a Robin type relation similar to (2.7 – 2.8) but with non-linear conductivity. The duap permeability model can be viewed as a non-equilibrium homogenization of the network of conductive fractures.

5.1 Richards' equation and instability of Raviart-Thomas elements

A standard model for the water flow in a partially saturated porous medium is Richards' equation which can be written as the system:

$$\partial_t \theta(h) + \operatorname{div}(\mathbf{v}) = f \quad \text{in } (0, T) \times \Omega, \quad (5.1)$$

$$\mathbf{v} = -k(h) \nabla(h + z) \quad \text{in } (0, T) \times \Omega. \quad (5.2)$$

The unknowns are the pressure head h and the water velocity \mathbf{v} while the other involved quantities are the density of volume water sources f , the z -coordinate, assumed to be in opposite direction to the gravity force, the water content θ and the hydraulic conductivity k , where θ and k are given nonlinear function of h . Both equations are considered on the domain $\Omega \subset \mathbf{R}^N$ and during the time interval $(0, T)$. Through this work we consider the Dirichlet boundary condition h_D on

$\Gamma_D \subset \partial\Omega$, the homogeneous Neumann condition $\mathbf{v} = 0$ on the remaining part of the boundary, and the initial condition h_0 for the pressure head.

The characteristic functions $\theta(h)$ and $K(h)$ are empirical. We assume the most common Mualem – van Genuchten model [1], [2]:

$$\theta(h) = \theta_r + (\theta_s - \theta_r)\tilde{\theta}(h), \quad (5.3)$$

$$\tilde{\theta}(h) = (1 + (\alpha h)^n)^{-m}, \quad m = 1 - 1/n \quad (5.4)$$

$$k(h) = k_s \tilde{\theta}^{0.5} \left(1 - (1 - \tilde{\theta}^{1/m})^m\right)^2, \quad (5.5)$$

where θ_r , θ_s , n , α , and k_s are suitable soil parameters.

System (5.1 – 5.2) represents a quasilinear degenerated parabolic-elliptic equation. The existence and uniqueness of the solution as well as some regularity properties were proved by ALT, and LUCKHAUS [3]. When solving Richards' equation numerically, we want to obtain a discrete velocity field which satisfies a discrete version of the continuity equation (5.1) up to the given tolerance of the nonlinear solver. This is important for a subsequent simulation of the water transport. That is why mixed or mixed-hybrid finite elements are used by many authors, e.g. [9], [12].

Motivated by these works, we want to develop a simulator that can solve coupled Richards' equations on domains of different dimension. Since the solution of Richards' equation evolve substantially only around a small wetting front region, adaptivity is crucial to achieve reasonable performance. To meet these two requirements, we have decided to try C++ finite element library DEAL II [16]. The library allows to produce a dimension independent code with h , p , and hp versions of adaptivity and provides a rich palette of finite elements. The only but fundamental restriction of the library is that elements has to be topologically equivalent to hypercubes. However, during tests of our code we have observed serious oscillations of the solution. Aim of this paper is to present these observations and give an explanation of this behavior.

The paper is organized as follows. First, the mixed discretization is described. Then, in Section 3, we make its comparison with a primary discretization and we demonstrate the presence of instabilities. In the last section, we derive a condition under which the mixed scheme obeys a discrete maximum principle in 1D and we discuss some similar results.

5.2 Mixed finite elements

In order to derive mixed formulation of the system (5.1 – 5.2), we multiply the first equation by a scalar test function ϕ , while in the second equation we divide by k , test by a vector valued function $\boldsymbol{\psi}$ and integrate by parts in the pressure term. Finally, we are looking for a solution $h \in L^2(\Omega)$, $\mathbf{v} \in H(\text{div}, \Omega)$ which satisfies

$$\int_{\Omega} k^{-1}(h)(\mathbf{v} \cdot \boldsymbol{\psi}) - \int_{\Omega} h \text{div } \boldsymbol{\psi} = \int_{\Omega} z \text{div } \boldsymbol{\psi} - \int_{\partial\Omega} (h_D + z) \boldsymbol{\psi} \cdot \mathbf{n}, \quad (5.6)$$

$$- \int_{\Omega} \partial_t \theta(h) \phi - \int_{\Omega} \phi \text{div } \mathbf{v} = - \int_{\Omega} f \phi \quad (5.7)$$

for all $\boldsymbol{\psi} \in H(\text{div}, \Omega)$ and $\phi \in L^2(\Omega)$, where $H(\text{div}, \Omega)$ is a space of vector valued L^2 -function with divergence in $L^2(\Omega)$.

Next, we consider a decomposition $\mathcal{T} = \{K_i\}$ of the domain $\Omega \subset \mathbf{R}^N$ into lines ($N=1$), quadrilaterals ($N=2$) or hexahedrons ($N=3$). On this computational grid we use Raviart-Thomas

finite elements RT_0 for discretization of the velocity and piecewise constant finite elements P_0 for discretization of the pressure head. More specifically, we consider discrete solution in a form

$$\mathbf{v}(t, \mathbf{x}) = \sum_i \tilde{u}_i(t) \boldsymbol{\psi}_i(\mathbf{x}), \quad h(t, \mathbf{x}) = \sum_i \tilde{h}_i(t) \phi_i(\mathbf{x}), \quad (5.8)$$

where \tilde{u} and \tilde{h} are unknown coefficient vectors. The backward Euler is used for temporal discretization. A fully implicit scheme is necessary to avoid oscillations on the saturated part of the domain where the equation becomes elliptic. Finally, we obtain a nonlinear system of equations which we solve by simple Pickard iterations. Resulting linear system for the solution \tilde{h}^k, \tilde{u}^k in iteration k of time t_n reads

$$A(h^{k-1})\tilde{u}^k + B\tilde{h}^k = F \quad (5.9)$$

$$B^T \tilde{u}^k + D(h^{k-1})\tilde{h}^k = G(h^{k-1}) \quad (5.10)$$

with

$$\begin{aligned} A_{i,j}(h^{k-1}) &= \sum_{K \in \mathcal{T}} \int_K k^{-1}(h^{k-1})(\boldsymbol{\psi}_i \cdot \boldsymbol{\psi}_j), \\ B_{i,j} &= - \sum_{K \in \mathcal{T}} \int_K \phi_i \operatorname{div} \boldsymbol{\psi}_j, \\ D_{i,j}(h^{k-1}) &= \sum_{K \in \mathcal{T}} \int_K -\frac{\theta'(h^{k-1})}{dt} \phi_i \phi_j \\ F_i &= \sum_{K \in \mathcal{T}} \int_K z \operatorname{div} \boldsymbol{\psi}_i - \int_{K \cap \Gamma_D} (z + h_D) \boldsymbol{\psi}_i \cdot \mathbf{n}, \\ G_i(h^{k-1}) &= \sum_{K \in \mathcal{T}} \int_K -\frac{\theta'(h^{k-1})h^{k-1}}{dt} \phi_i + \frac{\theta(h^{k-1}) - \theta^0}{dt} \phi_i, \end{aligned}$$

where h^{k-1} is the actual discrete pressure head field given by according to (5.8) and θ^0 is the water content field from the previous time t_{n-1} . Before solving system (5.9 – 5.10), we use the last pressure head \tilde{h}^{k-1} to resolve equation (5.9) and compute a residuum r^{k-1} of the equation (5.10). Iterations are stopped, when l^2 norm of the residuum drops under the prescribed tolerance. Then the residuum is subtracted from the actual water content which forms θ^0 for the next time step. This way we achieve a perfect conservation of the total water content over the whole domain.

5.3 Comparison of mixed and primary discretization

The described mixed finite element approximation with the lowest element order $d = 0$ (MFE) have been compared with a mature one dimensional solver based on the primary linear finite element (FE) approximation of the pressure. The later solver was thoroughly tested against experimental data in cooperation with Vogel et al. [20].

The setting of the one dimensional infiltration test problem was as follows: a vertical domain $(-5, 0)$ [m], the constant initial pressure head $h_0 = -150$ [m], the Dirichlet boundary condition $h_D = 1$ [m] on the top and the homogeneous Neumann condition on the bottom. The parameters of the soil model were $n = 1.14$, $\alpha = 0.1$ [m⁻¹], $\theta_r = 0.01$, $\theta_s = 0.480$, $k_s = 2$ [mh⁻¹]. This setting leads to a steep wetting front during the initial phase, thus we have to use short time steps. The wetting front goes from the top to the bottom so that the pressure head should be monotonous

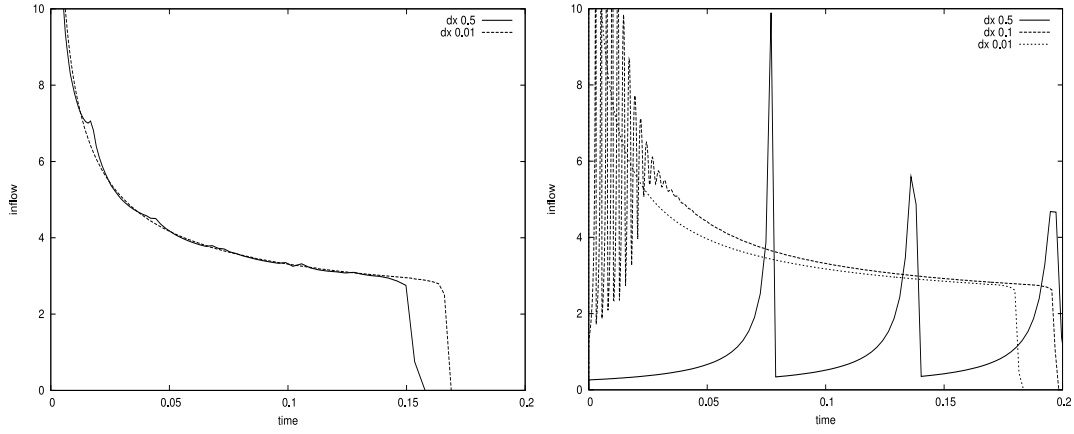


Figure 5.1: Infiltration velocity on the tof of the vertical 1D domain. The stable FE scheme (left) and the unstable MFE scheme (right)

in time and space, increasing from -150 up to $1 + z$. The velocity should be always negative. The MFE code was run on meshes with steps 0.01 , 0.1 , and 0.5 the FE code was run only for steps 0.01 and 0.5 . All simulations have started with the time step 10^{-6} and the time step is enlarged if the number of nonlinear iterations drops under 3 .

Figure 1 shows the infiltration velocity on the top of the domain up to the full saturation of the whole domain. For the fine mesh step 0.01 the results are comparable. The infiltration computed by the MFE code takes just a little bit longer compared to the FE code. On the other hand, for the coarser meshes, the MFE code produces terrible oscillations while the FE code still provides satisfactory results. The oscillations are not only in time but also in space and they get worse with shorter time steps or larger mesh steps. Values of the pressure head leave the valid interval $[-150, 1]$ and positive values of the velocity appear.

5.4 Discrete maximum principle

Maximum principle for elliptic PDEs states that a solution of the equation

$$\operatorname{div}(-\tilde{k}\nabla h) + \tilde{c}h = \tilde{f} \quad \text{on } \Omega, \quad h = \tilde{g} \quad \text{on } \partial\Omega, \quad (5.11)$$

with $\tilde{k} > 0$, $\tilde{c} \geq 0$, is non-negative provided \tilde{f} and \tilde{g} are non-negative. If a similar property holds for a discrete problem, we say that it obeys the discrete maximum principle (DMP).

In view of the previous section it seems that the MFE scheme violates DMP for the short time steps. To show this, we shall analyze one linear step, i.e. system (5.9 – 5.10), which can be viewed as the discretization of the linear elliptic problem (5.11) with $\tilde{k} = k(h)$, $\tilde{c} = \theta'(h)/dt$, and suitable positive \tilde{f} . We consider one dimensional domain with grid points $x_1 < x_2 < \dots < x_n$ and the lowest order elements $d = 0$. Further, we use equivalent mixed-hybrid discretization of (5.11). On every element $K_i = (x_i, x_{i+1})$ the discrete solution is represented by the pressure head h_i in the center of element, by the traces $\hat{h}_i^{1,2}$ on element boundary, and by the velocity $\mathbf{v}_i = u_i^1 \psi^1 + u_i^2 \psi^2$. The velocity is linear combination of discontinuous RT_0 base functions

$$\psi_i^1(x) = \frac{x_{i+1} - x}{x_{i+1} - x_i}, \quad \psi_i^2(x) = \frac{x - x_i}{x_{i+1} - x_i}$$

where coefficients $u_i^{1,2}$ are the outer normal fluxes from the element i . Proceeding similarly as in

the case of mixed formulation we obtain a discrete version of (5.11):

$$\sum_{j=1,2} \tilde{k}_i^{-1} u_i^j \int_{K_i} \psi_i^m \psi_i^j = h_i - \mathring{h}_i^m \quad \text{for } m = 1, 2 \quad (5.12)$$

$$\tilde{c}_i h_i |K_i| + u_i^1 + u_i^2 = \tilde{f}_i |K_i| \quad (5.13)$$

$$u_i^2 = -u_{i+1}^1, \quad \mathring{h}_i^2 = \mathring{h}_{i+1}^1. \quad (5.14)$$

We denote $\mathring{h}_i = \mathring{h}_i^2 = \mathring{h}_{i+1}^1$. The integral in (5.12) evaluates to $|K_i|/3$ and $-|K_i|/6$ for $m = j$ and $m \neq j$, respectively. On the Dirichlet boundary x_n we set $\mathring{h}_n^1 = h_D$. Then, eliminating h_i and $u_i^{1,2}$ from the system, we obtain an equation for \mathring{h}_i :

$$a_{i-1} \mathring{h}_{i-1} + (b_{i-1} + b_i) \mathring{h}_i + a_i \mathring{h}_{i+1} = c_{i-1} + c_i \quad (5.15)$$

where

$$a_i = \frac{2\tilde{k}_i}{|K_i|} - \frac{\alpha_i \alpha_i}{\beta_i}, \quad b_i = \frac{4\tilde{k}_i}{|K_i|} - \frac{\alpha_i \alpha_i}{\beta_i}, \quad c_i = \frac{\alpha_i |K_i| \tilde{f}_i}{\beta_i}, \quad (5.16)$$

$$\alpha_i = \frac{6\tilde{k}_i}{|K_i|}, \quad \beta_i = |K_i| \tilde{c}_i + 2\alpha_i. \quad (5.17)$$

Equation (5.15) is one row of a linear system $A\mathring{h} = c$, where vector c is non-negative provided \tilde{f}_i and h_D are non-negative. In order to obtain a non-negative solution \mathring{h} , the matrix A has to have positive inverse. This holds if A is so called M -matrix, that is a matrix with positive diagonal entries, non-positive off diagonal entries, and positive row sums. In our case this is equivalent to $a_i \leq 0$, $b_i > 0$, and $a_i + b_i > 0$. The later two inequalities are always true, while the first one holds only if

$$\frac{|K_i|^2}{6} \leq \frac{\tilde{k}_i}{\tilde{c}_i} = dt \frac{k(h_i)}{\theta'(h_i)}. \quad (5.18)$$

For positive \tilde{f} and \tilde{g} , this condition implies positive nodal pressures \mathring{h}_i . Then the elemental pressures h_i are also positive since

$$h_i = \frac{|K_i| \tilde{f}_i + \alpha_i (\mathring{h}_i^1 + \mathring{h}_i^2)}{\beta_i}.$$

Our numerical experiments reveals that oscillations of the solution appears exactly on that elements where the condition (5.18) does not hold. Thus to get stable scheme one has to adapt the element size $|K_i|$ according to the condition. However, the limit of the right hand side as $h_i \rightarrow -\infty$ is zero, at least for the soil model (5.3 – 5.5). It means that $|K_i|$ should be very small on the whole dry part of the domain where the solution is mainly constant which is highly ineffective. Moreover, mixed elements on 2D quadrilaterals or 3D hexahedrons never leads to M -matrix even for $c_i = 0$.

In the paper due to Younes, Ackerer, and Lehmann [14] authors prove stability conditions similar to (5.18) for mixed-hybrid elements on triangular and tetrahedral meshes. We can conclude that the mixed scheme for the Richards' equation is stable only for large time steps and therefore is not suitable for a robust solver. However, one can try to modify the mixed scheme to make it more stable. In fact two such modifications were already proposed in [14].

5.5 Lumped Mixed-Hybrid Method

The instabilities demonstrated in previous section for the simple 1d elements are even worse for higher dimensions. One possible solution is the diagonalization of the method (lumped mixed-hybrid method, LMH) proposed in [14]. This method use a simple modification of the saddle point problem (3.31 – 3.32) replacing the time derivative of the element pressures by the time derivative of the weighted average of the trace pressures. This is done by replacing the from c_t by:

$$\tilde{c}_t(p, \mathring{p}, q, \mathring{q}) = \sum_{d=1}^3 \sum_{T \in \mathcal{T}_d} \sum_{i=1}^{d+1} \alpha_{T,i} |T| \frac{\delta_d S_d}{\tau} (\mathring{p}|_{S_{T,i}} \mathring{q}|_{S_{T,i}}),$$

and the source term in (3.27) by

$$\sum_{d=1}^3 \sum_{T \in \mathcal{T}_d} \sum_{i=1}^{d+1} \alpha_{T,i} |T| \delta_d f_d \mathring{q}|_{S_{T,i}},$$

where $|T|$ is volume of the element T , $S_{T,i}$ is the i -th side of T , and $\mathring{h}|_{S_{T,i}}$ is the degree of freedom on the side $S_{T,i}$.

The wights $\alpha_{T,i}$ for the trace pressures are chosen as follows. For a single element T of dimension d , we denote A the local matrix on T corresponding to the form $a(\cdot, \cdot)$ in (3.23). Then the optimal choice of the weights $\alpha_{T,i}$, $i = 0, \dots, d$ is:

$$\alpha_{T,i} = \frac{\lambda_i}{\lambda}, \quad \lambda_i = \sum_{j=0}^d A_{ij}^{-1}, \quad \lambda = \sum_{i=0}^d \lambda_i$$

however the unconditioned satisfaction of DMP is guarantied even for the simple average:

$$\alpha_{T,i} = \frac{1}{d+1}.$$

As this modification adds the source term to the balance of side fluxes, the side fluxes on a face shared by two elements K and L do not sum to zero anymore. To fix this, we have to perform a postprocessing of the fluxes after the linear system is solved. Assuming that the side i of the element K and the side j of the element L corresponds to the common edge e , we set corrected flux $\tilde{u}_{K,i}$ as:

$$\tilde{u}_{K,i} = v_{K,i} + \alpha_{K,i} |K| F_e, \quad F_e = \int_e \delta_d f_d \mathring{q}_e - \frac{\delta_d S_d}{\tau} (\mathring{p}_e - \mathring{p}_e^0) \mathring{q}_e$$

where \mathring{p}_e , \mathring{p}_e^0 , \mathring{q}_e are the trace pressure, the trace pressure in previous time step and the trace test function on the edge e respectively.

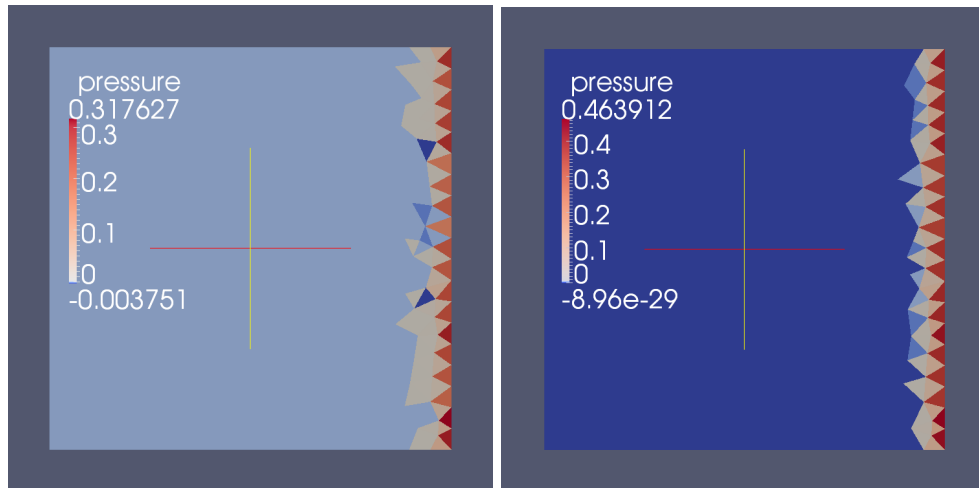


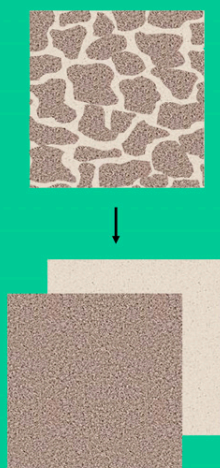
Figure 5.2: Comparison of MH (left) and LMH scheme (right), $\tau = 10^{-4}$.

Figure 5.2 shows a comparison of the results using conventional MH scheme and LMH scheme. At the MH scheme one can observe oscillations in the wavefront where the minimum value is significantly less than zero.

5.6 Fully coupled dual permeability model

The paper [20] reprinted in this section presents a dual permeability model a fractured soil and its solution by a fully coupled finite element solver. The region of soil parameters is identified where the full coupling provides better results than the sequential coupling. Both coupling techniques are compared to the real measurements.

Tomas Vogel*
Jan Brezina
Michal Dohnal
Jaromir Dusek



The first-order transfer term, controlling the soil water exchange between the preferential flow domain and the soil matrix, is parameterized with the emphasis on interfacial resistance. This parameterization is used as a framework for the evaluation of the performance of two basic approaches to numerical coupling of the governing flow equations.

T. Vogel, M. Dohnal, and J. Dusek, Faculty of Civil Engineering, Czech Technical Univ. in Prague, Thakurova 7, 166 29 Prague, Czech Republic; and J. Brezina, Faculty of Mechatronics, Informatics and Interdisciplinary Studies, Technical Univ. of Liberec, Studentska 2, 461 17 Liberec, Czech Republic.
*Corresponding author (vogel@fsv.cvut.cz).

Vadose Zone J. 9:260–267
doi:10.2136/vzj2009.0091
Received 30 June 2009.
Published online 3 May 2010.

© Soil Science Society of America
5585 Guilford Rd., Madison, WI 53711 USA.
All rights reserved. No part of this periodical may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher.

Physical and Numerical Coupling in Dual-Continuum Modeling of Preferential Flow

Dual-continuum models are useful for describing flow in porous systems with significant local pressure disequilibrium between slow moving water, contained in the porous matrix, and fast moving water in preferential pathways. The formation and intensity of preferential flow depends on the contrast between the hydraulic properties of the two flow domains as well as on the properties of their interface. In this study, we focused on both physical coupling of the flow domains through the mass transfer term and numerical coupling of the respective governing equations. The set of governing equations was alternatively solved using a sequentially coupled (SC) approach and a fully coupled (FC) approach. The SC approach was shown to be computationally more efficient for strongly developed preferential flow in systems with high interfacial resistance; however, it becomes numerically unstable for weak preferential flow associated with low interfacial resistance. The FC approach is a computationally more expensive yet numerically more robust alternative, capable of simulating a complete class of intermediate flow regimes ranging from strongly preferential flow in a dual-continuum system to nonpreferential flow in a single-continuum system. To illustrate the performance of the numerical coupling approaches in conjunction with the effect of different interfacial resistances, we present a simple example problem involving one-dimensional near-saturated flow in a vertical soil column.

Abbreviations: FC, fully coupled; PF, preferential flow; SC, sequentially coupled; SM, soil matrix.

When seeking a reasonably realistic description of the field-scale movement of water and solutes, the assumption that soil is a single-continuum system is often inadequate. During rainstorms or intensive irrigation events, water and chemicals can move at relatively large velocities in macropores or other structural elements, causing local disequilibrium conditions in pressure heads and solute concentrations. Preferential flow related to soil structure has been widely reported in soils containing worm holes, root channels, and interaggregate fissures (e.g., Bouma, 1981; Beven and Germann, 1982). Other types of preferential flow have been associated with textural differences rather than structural aspects (e.g., Cislerova et al., 2002), water repellency, and other conditions (e.g., Snehota et al., 2008).

In the last two decades, several dual-continuum models (e.g., Gerke and van Genuchten, 1993a; Jarvis, 1994; Ray et al., 1997; Vogel et al., 2000) have been developed to simulate preferential flow in soils conceptualized as having two pore domains (see Fig. 1): the soil matrix domain (SM domain) and the preferential flow domain (PF domain). Dual-continuum models have been commonly referred to as dual-porosity or dual-permeability models (although these two terms are not always considered synonymous). In the dual-continuum models, a Darcian flow assumption for the movement of water and a Fickian dispersion assumption for the transport of solutes have been assumed to be valid for each of the two pore domains separately, allowing local interdomain disequilibrium. Detailed discussions of various forms and applications of dual-continuum models can be found in, e.g., Gerke (2006), Jarvis (2007), and Köhne et al. (2009).

A crucial component of dual-continuum models is the mass transfer term governing the exchange of water between the PF and SM domains through their interface (PF–SM interface). Several empirical and semiempirical expressions are used to represent mass transfer in current models. Further research is needed, however, to establish more adequate and computationally feasible relationships and to develop the experimental methodologies needed to determine the additional constitutive parameters. Generally speaking, water communication between the two flow domains is a transient, nonlinear process. For this

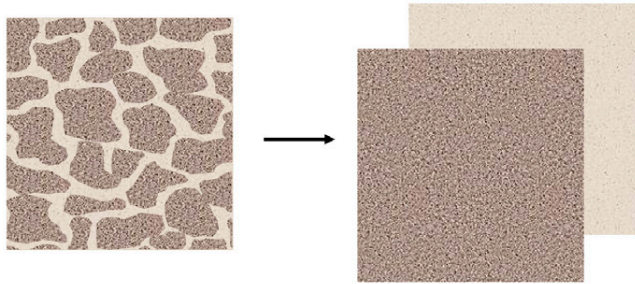


Fig. 1. The dual-continuum model: an internally structured porous medium is decomposed into two flow domains, the soil matrix domain (SM domain) and the preferential flow domain (PF domain).

reason, Zimmerman et al. (1993) suggested a nonlinear ordinary differential equation to evaluate the fracture–matrix transfer for fractured rock formations. More recently, Lewandowska et al. (2004) implemented a nonlinear exchange term assuming an additional local flow equation in a double-porosity homogenization approach. Köhne et al. (2004) proposed a second-order transfer term. In spite of that, the first-order algebraic approximation of the interdomain transfer (e.g., Gerke and van Genuchten, 1993b, 1996) is still considered a reasonably adequate and computationally highly efficient assumption.

As far as the numerical coupling of the dual-continuum system is concerned, Gerke and van Genuchten (1993a) solved the dual set of governing equations as a fully coupled system (FC approach). Tseng et al. (1995) presented a partitioned solution procedure performed sequentially for the two flow domains (SC approach). They showed that the proposed single-pass scheme is stable and computationally more efficient than the FC approach. The SC approach was also implemented in the HYDRUS-1D code (Šimunek and van Genuchten, 2008).

To allow comparison of the two numerical coupling approaches (FC and SC) on an equal basis, we implemented the FC approach in a numerical solver that was originally based on the SC approach. In this study, both approaches were compared through solving a simple example problem, designed to test their performance under different flow regimes, ranging from strongly preferential flow in a dual-continuum system to nonpreferential flow in a single-continuum system.

The main objectives of this study were twofold: (i) to improve the existing dual-continuum model in terms of computational efficiency and numerical stability, and (ii) to identify the most relevant and easy-to-interpret set of parameters that could be used to characterize the flow regime in a dual-continuum system. The latter objective is especially important in situations in which preferential flow is observed macroscopically but more detailed microscopic information about the physical properties of the porous system is unavailable.

Dual-Continuum System

The dual-continuum system consists of the PF domain and the SM domain separated by the PF–SM interface. Variably saturated flow of water in a dual-continuum system is described by a dual set of Richards' equations (e.g., Gerke and van Genuchten, 1993a). These equations are coupled through a soil water transfer term, which allows a dynamic exchange of water between the PF domain and the SM domain. In case of one-dimensional flow, the dual set of governing equations can be written in the following form:

$$\frac{\partial w_f \theta_f}{\partial t} = \frac{\partial}{\partial z} \left[w_f K_f \left(\frac{\partial h_f}{\partial z} + 1 \right) \right] - w_f S_f - \Gamma_w \quad [1]$$

$$\frac{\partial w_m \theta_m}{\partial t} = \frac{\partial}{\partial z} \left[w_m K_m \left(\frac{\partial h_m}{\partial z} + 1 \right) \right] - w_m S_m + \Gamma_w \quad [2]$$

where subscript *m* denotes the SM domain, subscript *f* denotes the PF domain, *h* is the soil water pressure head (m), *K* is the unsaturated hydraulic conductivity (m s^{-1}), θ is the volumetric soil water content (dimensionless), *S* is the intensity of the local root water uptake (s^{-1}), Γ_w is the soil water transfer term (s^{-1}) defined as the volume of fluid moving from the PF domain to the SM domain per unit volume of bulk soil per unit time, and w_m and w_f are the volumetric fractions of the pore space occupied by the respective flow domains ($w_m + w_f = 1$) (all variables are also defined in Appendix 1).

The composite properties of the bulk soil are related to the respective domain-specific properties through simple summation rules:

$$\theta_s = w_f \theta_{fs} + w_m \theta_{ms} \quad [3]$$

$$K_s = w_f K_{fs} + w_m K_{ms} \quad [4]$$

where θ_s is the saturated water content and K_s is the saturated hydraulic conductivity.

Gerke and van Genuchten (1993b) suggested the following first-order approximation of the transfer term:

$$\Gamma_w = \alpha_w (h_f - h_m) \quad [5]$$

in which α_w denotes the first-order soil water transfer coefficient ($\text{m}^{-1} \text{s}^{-1}$). They showed that in structured soils,

$$\alpha_w \propto \frac{\beta}{a^2} K_a \quad [6]$$

where β is a dimensionless geometry factor related to the shape of soil aggregates, *a* is the average soil aggregate radius (m), and K_a is the unsaturated hydraulic conductivity at or near the PF–SM interface (m s^{-1}).

In this study, the first-order transfer coefficient is parameterized in a slightly different way. First, the interfacial conductivity is

expressed as a product of the saturated conductivity, K_{as} , and the relative unsaturated conductivity, K_{ar} . The former of the two conductivities is then incorporated into the coefficient α_{ws} , so that

$$\alpha_w = \alpha_{ws} K_{ar}(h_f, h_m) \quad [7]$$

where α_{ws} ($m^{-1} s^{-1}$) is the value of soil water transfer coefficient at saturation (when $K_{ar} = 1$).

Next, we assume that the transfer coefficient is inversely proportional to the interfacial resistance, r_{ws} (s). The interfacial resistance operates at the microscopic scale, but can be upscaled to the macroscopic level by utilizing the concept of specific interfacial area:

$$\alpha_{ws} = \frac{\tau}{r_{ws}} \quad [8]$$

The specific interfacial area τ (m^{-1}) is defined as the area of the PF–SM interface per bulk volume of soil.

The magnitude of r_{ws} clearly depends on the magnitude of the interfacial conductivity K_{as} , but it is also proportional to a characteristic length, λ (m), associated with the mass transfer. This length could be thought of as the distance separating the two flow domains, or better as an effective distance across which the pressure head difference $h_f - h_m$ operates. The definition formula for the interfacial resistance combines the effect of K_{as} and λ :

$$r_{ws} = \frac{\lambda}{K_{as}} \quad [9]$$

Note that the present formulation transforms to a form consistent with the original concept of Gerke and van Genuchten (1993b), as given by Eq. [6], in the case where λ is equal to the aggregate radius a , and $\tau = \beta/a$. These two conditions can be combined in a more compact way as $\lambda\tau = \beta$. The relationship between λ , τ , and β for different interfacial geometries was studied in detail by Gerke and van Genuchten (1996).

At saturation, the interfacial conductivity K_{as} is closely related to the saturated conductivity of the soil matrix, K_{ms} ; however, under the presence of increased interfacial resistance, e.g., due to soil aggregate coating, it may be significantly lower (e.g., Gerke and Köhne, 2002). Therefore, we can expect that $K_{as} \leq K_{ms}$.

The presence of coating at the PF–SM interface may also reduce λ because the soil water pressure difference operates at much smaller distances, related to the thickness of the aggregate coating rather than the size of the aggregates. Thus, the characteristic length is bound by the condition $\lambda \leq a$.

It is of interest to note that the degree of hydraulic communication between the PF and SM flow domains can be characterized by two dimensionless numbers: $\lambda\tau$ and K_{as}/K_{ms} . The two numbers uniquely determine the magnitude of the transfer coefficient α_{ws} , provided that τ and K_{ms} are known a priori. These numbers

complement the other two dimensionless numbers, which characterize the contrast between the properties of the PF and SM domains, i.e., w_f and K_{fs}/K_{ms} .

The dual system of governing equations is numerically solved by the finite element method. The numerical solver is implemented in the variably saturated flow and transport model S1D (see Ray et al. [2004] or Vogel et al. [2007] for recent applications of the code). The S1D code is a follow-up version of the single-continuum model HYDRUS 5.0 (Vogel et al., 1996).

Sequentially Coupled Approach

With the SC approach, the governing flow equations are solved separately for each of the two flow domains at each time level. The first-order transfer term, which mediates the exchange of water between the two domains, is evaluated at each time level using the soil water pressure information from the previous time level, as shown in

$$\Gamma_w^j = \alpha_{ws} K_{ar}^{j-1} (h_f^{j-1} - h_m^{j-1}) \quad [10]$$

in which the relative hydraulic conductivity of the PF–SM interface is determined as

$$K_{ar}^{j-1} = \begin{cases} \min[K_{fr}(h_f^{j-1}), K_{mr}(h_m^{j-1})] & \text{for } h_f^{j-1} \geq h_m^{j-1} \\ \min[K_{fr}(h_m^{j-1}), K_{mr}(h_f^{j-1})] & \text{for } h_f^{j-1} < h_m^{j-1} \end{cases} \quad [11]$$

where K_{mr} and K_{fr} are the relative hydraulic conductivities of the SM domain and the PF domain, respectively. Equations [10] and [11] are evaluated at each nodal point of the space discretization.

Fully Coupled Approach Variational Formulation

Alternatively to the SC approach, described above, the set of two governing equations can be solved as a fully coupled system. To derive a joined variational formulation, we multiply Eq. [1] and [2] by the test functions ϕ_f and ϕ_m , integrate by parts, and add together as follows:

$$\begin{aligned} & \int_{\Omega} w_f \frac{\partial \theta_f}{\partial t} + w_m \frac{\partial \theta_m}{\partial t} dz \\ & + \int_{\Omega} w_f \alpha_w (h_f - h_m) \phi_f + w_m \alpha_w (h_m - h_f) \phi_m dz \\ & + \int_{\Omega} w_f K_f \frac{\partial h_f}{\partial z} \frac{\partial \phi_f}{\partial z} + w_m K_m \frac{\partial h_m}{\partial z} \frac{\partial \phi_m}{\partial z} dz \\ & = \int_{\Omega} w_f \left(K_f \frac{\partial \phi_f}{\partial z} - S_f \phi_f \right) + w_m \left(K_m \frac{\partial \phi_m}{\partial z} - S_m \phi_m \right) dz \end{aligned} \quad [12]$$

for every $(\phi_m, \phi_f) \in H_0^1(\Omega) H_0^1(\Omega)$

where Ω is a domain with Lipschitz boundary and $H_0^1(\Omega)$ is the space of weakly differentiable functions with zero trace on the Dirichlet boundary (see discussion of boundary conditions below).

Discretization

The variational formulation Eq. [12] is discretized by the finite element method in the very same way as in the SC approach. Namely, ϕ_f and ϕ_m are approximated by piecewise linear functions, while θ_f , θ_m , b_f and b_m are approximated by piecewise constant functions in the first two integrals of Eq. [12] and by piecewise linear functions in the remaining terms. The resulting system of ordinary differential equations is solved by an implicit Euler scheme using Pickard iterations to solve the nonlinear system at each time level. The linear system solved at each iteration reads

$$(\mathbf{T} + \mathbf{D} + \mathbf{A})\mathbf{h} = \mathbf{b} \quad [13]$$

where \mathbf{T} is a diagonal matrix corresponding to the time term, i.e., the first integral in Eq. [12], \mathbf{D} is a block diagonal matrix of the interdomain communication corresponding to the second integral in Eq. [12], \mathbf{A} is a three-diagonal matrix corresponding to the third integral, the vector \mathbf{b} is related to the right-hand side of Eq. [12] plus part of the time difference term due to the previous time, and \mathbf{h} is the vector of unknown nodal values of the pressure head.

Composition of the Linear System

For each nodal point i , we denote its base function by ϕ^i . Using either $(\phi_m, \phi_f) = (\phi^i, 0)$ or $(\phi_m, \phi_f) = (0, \phi^i)$ as test functions in Eq. [12], we obtain rows $2i - 1$ and $2i$ in our linear system, so that the odd indices are used for the SM domain and even indices for the PF domain.

The matrix \mathbf{T} is diagonal, composed from the values

$$T^{2i-1,2i-1} = \frac{w^{2i-1} F^i C^{2i-1}}{\Delta t} \quad [14]$$

$$T^{2i,2i} = \frac{w^{2i} F^i C^{2i}}{\Delta t} \quad [15]$$

where F_i is the integral of the base function ϕ^i , $w^{2i-1} = w_m$ and $w^{2i} = w_f$ are the volumetric fractions of the domains in the node i , C^{2i-1} and C^{2i} are the soil water capacities computed from the respective pressure heads $\tilde{h}^{2i-1} = \tilde{h}_m^i$ and $\tilde{h}^{2i} = \tilde{h}_f^i$ at the previous iteration, and Δt is the time step.

The interdomain communication matrix \mathbf{D} consists of 2×2 blocks:

$$D^{2i-1,2i-1} = F^i \alpha_w^i \quad [16]$$

$$D^{2i-1,2i} = -F^i \alpha_w^i$$

$$D^{2i-1,2i} = -F^i \alpha_w^i \quad [17]$$

$$D^{2i,2i} = F^i \alpha_w^i$$

where α_w^i are nodal values of the soil water transfer coefficient.

The last matrix \mathbf{A} is tridiagonal with values

$$\begin{aligned} A^{2i-1,2i-3} &= K_m^- + K_m^+ \\ A^{2i-1,2i-1} &= -K_m^- \end{aligned} \quad [18]$$

$$\begin{aligned} A^{2i-1,2i+1} &= -K_m^+ \\ A^{2i,2i-2} &= K_f^- + K_f^+ \\ A^{2i,2i} &= -K_f^- \\ A^{2i,2i+2} &= -K_f^+ \end{aligned} \quad [19]$$

where K^+ and K^- are integrals over the neighboring elements of the node i , in particular

$$K_m^- = \frac{w^{2i-3} K_{ar}^{2i-3} + w^{2i-1} K_{ar}^{2i-1}}{2(z^i - z^{i-1})} \quad [20]$$

$$K_m^+ = \frac{w^{2i+1} K_{ar}^{2i+1} + w^{2i-1} K_{ar}^{2i-1}}{2(z^{i+1} - z^i)}$$

$$K_f^- = \frac{w^{2i-2} K_{ar}^{2i-2} + w^{2i} K_{ar}^{2i}}{2(z^i - z^{i-1})} \quad [21]$$

$$K_f^+ = \frac{w^{2i+2} K_{ar}^{2i+2} + w^{2i} K_{ar}^{2i}}{2(z^{i+1} - z^i)}$$

The right-hand-side vector \mathbf{b} consists of a source term and a gravity term, which have already been presented on the right-hand side of the variational formulation Eq. [12], but in addition it also contains part of the time difference due to the water content in the previous time. Finally, because we have assumed that the water uptake takes place only in the SM domain, the sink term in the PF domain is set equal to zero. Then the formulas for the \mathbf{b} vector are

$$\begin{aligned} b^{2i-1} &= K_m^+ - K_m^- \\ &+ w^{2i-1} F^i \left(-S_m^i + \frac{C^{2i-1} \tilde{h}^{2i-1}}{\Delta t} - \frac{\tilde{\theta}^{2i-1} - \hat{\theta}^{2i-1}}{\Delta t} \right) \end{aligned} \quad [22]$$

$$b^{2i} = K_f^+ - K_f^- + w^{2i} F^i \left(\frac{C^{2i} \tilde{h}^{2i}}{\Delta t} - \frac{\tilde{\theta}^{2i} - \hat{\theta}^{2i}}{\Delta t} \right) \quad [23]$$

where S_m^i is the nodal value of the sink term in node i , $\tilde{\mathbf{h}}$ is the pressure head vector at the last iteration, $\tilde{\boldsymbol{\theta}}$ is the corresponding water content vector, and $\hat{\boldsymbol{\theta}}$ is the water content vector at the previous time step.

Compared with the SC approach, the FC approach is numerically more stable with respect to the high values of α_{ws} . This fact allowed us to relax the artificial restriction, built into the SC code, that forced α_{ws} to be zero when the target domain of the interdomain flux was saturated.

Boundary Conditions

The simulator supports a number of boundary conditions of different complexity, in particular the conditions for simulating the precipitation and evapotranspiration processes. A more detailed description of these conditions, specifically the procedure for redirecting the infiltration excess water from the SM to the PF domain during extreme rainfall events, was given in Dusek et al. (2008).

From the point of view of a numerical solution, all types of boundary conditions lead to either a Dirichlet or a Neumann boundary condition. In the case of a Neumann boundary condition, we simply add a given flux to the right-hand side of Eq. [13], while at the nodes with a Dirichlet boundary condition, we use the known value of the pressure head to rewrite the corresponding row of the linear system as a flux equation. This makes it possible to incorporate the algorithm for handling the infiltration excess water directly into the linear system and results in a reduced number of iterations. On the other hand, it has the effect of breaking the symmetry of the matrix, which could be undesirable if an iterative method is preferred for solving the linear system. For a direct solver, as used in the present version of the code, this does not represent a problem.

Example Problem

To illustrate the performance of the numerical coupling approaches in conjunction with the effect of different interfacial resistances, we present a simple example problem involving one-dimensional infiltration in a vertical soil column, conceptualized as a dual-continuum system. The upper boundary of the initially dry soil column is exposed to ponded infiltration, which lasts sufficiently long to lead to the outflow of soil water from the lower boundary of the column.

The example problem is formulated as follows:

- Geometry: one-dimensional homogeneous vertical soil column 100 cm high.
- SM domain: loamy sand (soil hydraulic parameters are shown in Table 1); $K_{ms} = 1 \text{ cm h}^{-1}$.
- PF domain: sand (see Table 1); $K_{fs} = 100 \text{ cm h}^{-1}$; $w_f = 0.05$.
- Specific interfacial area: $\tau = 1 \text{ cm}^{-1}$.
- Initial conditions: equilibrium with groundwater table at the depth of 5 m below the soil surface.
- Upper boundary conditions: ponded infiltration (Dirichlet type); depth of ponding = 2 cm.
- Lower boundary conditions: free drainage (Neumann type).

The soil hydraulic parameters were generated using the ROSETTA database (Schaap et al., 2001). The PF domain is represented by sand, to imitate the situation in which the preferential pathways are filled with relatively coarse particles eroded from the matrix. The values of the saturated hydraulic conductivities computed by ROSETTA were respected only in their order of magnitude. The

Table 1. Soil residual and saturated volumetric water content (θ_r and θ_s , respectively) and fitted parameters α and n of hydraulic conductivity (ROSETTA, Schaap et al., 2001).

Parameter	θ_r	θ_s	α	n
			cm^{-1}	
Soil matrix domain	0.049	0.390	0.035	1.75
Preferential flow domain	0.053	0.375	0.035	3.18

value of the specific interfacial area τ was chosen arbitrarily, however, within the expected range for structured soils.

The example problem was alternatively solved by applying the SC approach and the FC approach. In addition, the problem was solved by the conventional single-continuum approach, in which the soil hydraulic properties were defined as composite properties of the two flow domains (see Eq. [3] and [4]).

The simulations of soil water flow in the soil column were repeated with different values of interfacial resistance r_{ws} ranging from 1 h (high interdomain communication) to 10,000 h (low interdomain communication).

Altogether, 11 simulation scenarios were executed, involving one single-continuum approach and two dual-continuum approaches—each applied with five different interfacial resistances.

Results and Discussion

Impact of Interfacial Resistance

The results of the different simulation scenarios are presented in Fig. 2, 3, 4, and 5. Figure 2 indicates a relatively weak dependence of the simulated infiltration rates on the interfacial resistance. On the other hand, the outflow rates (see Fig. 3) differed significantly when different values of r_{ws} were applied.

The simulated responses of soil water pressure at the bottom of the soil column, as shown in Fig. 4, reveal significant local disequilibrium between the flow domains for strongly preferential scenarios ($r_{ws} = 100 \text{ h}$ and higher). The disequilibrium started with the arrival of the moisture front and lasted for the rest of the simulation period. Pressure responses for weak preferential flow scenarios ($r_{ws} = 1$ and 10 h) were much more synchronized between the two domains. The local pressure became reequilibrated soon after the arrival of the moisture front.

As shown in Fig. 4, both numerical coupling methods provide consistent results for strongly preferential scenarios. While the FC approach was numerically stable for all scenarios, the SC approach failed to converge for the weak preferential flow scenarios.

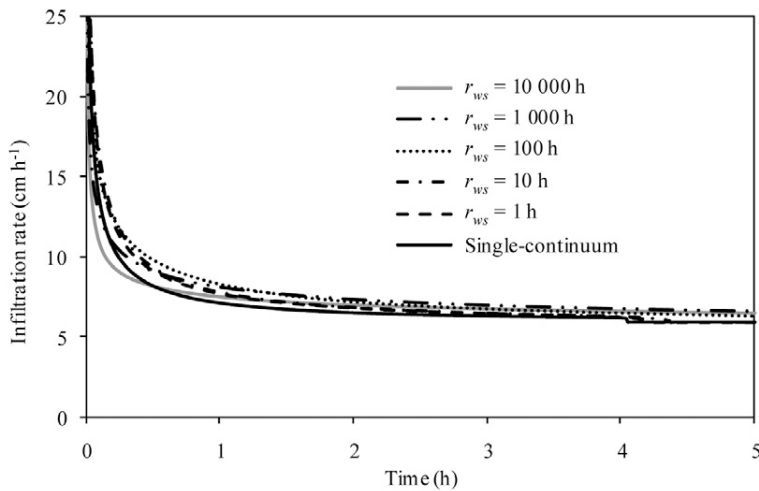


Fig. 2. Infiltration rates at the soil surface during ponded infiltration computed for different values of interfacial resistance r_{ws} (the dual-continuum scenarios were simulated with the fully coupled approach).

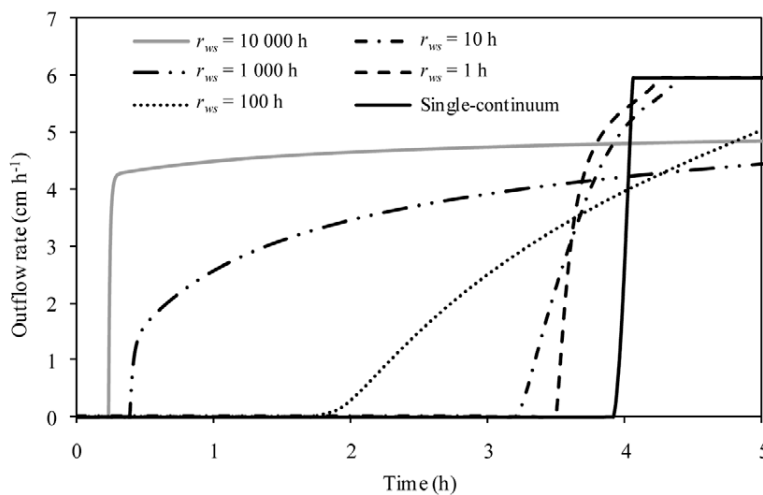


Fig. 3. Outflow rates at the bottom of the soil column computed for different values of interfacial resistance r_{ws} (the dual-continuum scenarios were simulated with the fully coupled approach).

Figure 5 shows the interdomain soil water transfer rates Γ_w at $t = 1$ h computed by S1D for different input values of r_{ws} . Positive values of Γ_w indicate transfer from preferential pathways to the soil matrix. The figure shows highly concentrated transfer rates along the advancing moisture front when low interfacial resistances are considered (for $r_{ws} = 1$ and 10 h).

Figures 2, 3, and 4 can also be used to compare the results of the dual-continuum scenarios with the single-continuum scenario. The outflow rates in Fig. 3 as well as the pressure heads in Fig. 4 indicate that for the weak preferential flow scenarios the dual-continuum results are, in fact, not that much different from those computed by the single-continuum model. In other words, the preferential flow effects become increasingly significant when the interfacial

resistance is higher than about two orders of magnitude compared with the lowest considered resistance (in our case, for $r_{ws} = 100$ h and higher).

Since the ratio K_{as}/K_{ms} is expected to be ≤ 1 , the lowest possible interfacial resistance is equal to λ/K_{ms} (cf. Eq. [9]). Let us assume that the characteristic length λ in our example problem is ≥ 1 cm. The lowest possible resistance is then $r_{ws} = 1$ h (as $K_{ms} = 1$ cm h⁻¹). In other words, the value of 1 h is a reasonable choice for the lowest possible r_{ws} in our system as long as $\lambda \geq 1$ cm is a reasonable estimate of the characteristic length.

The highest value of the interfacial resistance, $r_{ws} = 10,000$ h, was chosen as an extreme case to represent systems with a low interfacial conductivity K_{as} or a large characteristic length λ (e.g., in soils with a sparse network of preferential pathways).

When τ is known and K_{ar} is evaluated by Eq. [11] or a similar procedure, r_{ws} represents a single-valued parameter that determines the rate of the soil water transfer between the PF and SM domains. This makes r_{ws} useful not only for comparing the responses of dual-continuum systems with different level of interdomain communication (as shown in this study), but also as an important parameter in experimental studies in which the hydraulic properties of structured soils are estimated by inverse modeling.

Preferential flow in structured soils, represented by a dual-continuum system, is often thought of as being, to a large extent, controlled by the contrast in hydraulic conductivity between the soil matrix and preferential pathways. The presented parameterization concept stresses the role of the interfacial resistance as an equally important factor. The resistance is responsible for local disequilibrium, which is essential for supporting persistent preferential flow conditions. Low interfacial resistance in a dual-continuum system leads to negligible preferential flow, even when large conductivity contrasts between the flow domains exist.

Stability Issues

The SC version of the S1D code failed to converge for the weak preferential flow scenarios ($r_{ws} < 100$ h) but was stable for the strongly preferential scenarios. This is related to the basic assumption of the SC approach—that the soil matrix locally absorbs or releases a reasonably small amount of water within any time step of the numerical solution, so that the flow equations can be solved in sequence and the interdomain transfer of water, which is evaluated at the end of the time step, can be approximated by the transfer rate calculated from the pressure difference at the

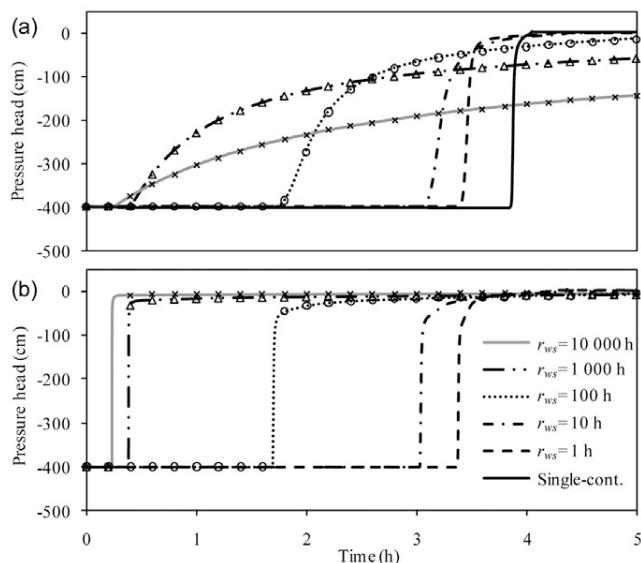


Fig. 4. Soil water pressure head at the depth of 100 cm computed for different values of interfacial resistance r_{ws} : (a) soil matrix domain, (b) preferential flow domain. Lines denote the fully coupled approach, while symbols are used for the sequentially coupled (SC) approach (missing symbols for the smallest interfacial resistances indicate non-convergence of the SC approach).

beginning of the time step (cf. Eq. [10]). Low interfacial resistances lead to large interdomain transfers (see Fig. 5) with adverse effects on convergence.

Although the FC approach permits much lower values of the interfacial resistance, it is paid off by worse conditioning of the nonlinear problem. In practice, this leads to a higher number of linear iterations, sometime even to oscillations and nonconvergence. In particular, this may happen when the time step is too small, so that large linear steps may lead to overshooting. This type of instability can probably be avoided by using a suitable line search method that has been proven to be globally convergent (Deuffhard, 2004). When used to solve our example problem, the FC scheme converged for all selected scenarios.

The computational efficiency of the numerical codes based on the SC and FC approaches is compared in Table 2.

Conclusions

The parameterization of the first-order transfer coefficient, controlling the exchange of water between the PF and SM domains, was reformulated to allow a more straightforward interpretation of the hydraulic function of the PF–SM interface. This parameterization was built around the notion of interfacial resistance. It was shown that the preferential flow effects become increasingly significant, and therefore worth modeling by a dual- instead of a single-continuum approach, when the interfacial resistance is higher than the value of λ/K_{ms} by about two orders of magnitude

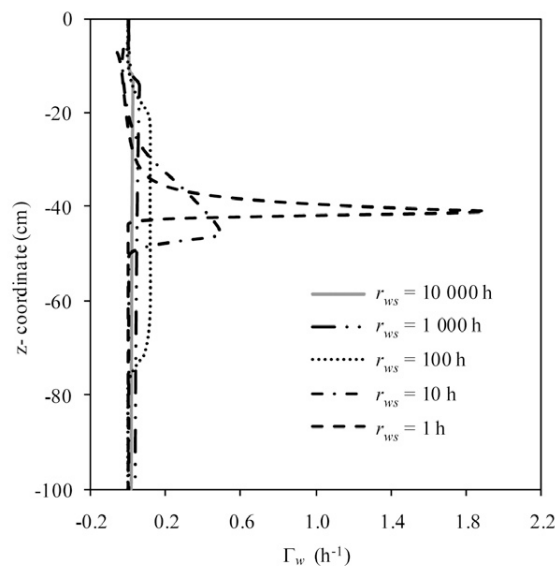


Fig. 5. Interdomain soil water transfer rates (Γ_w) at time $t = 1$ h computed for different values of interfacial resistance r_{ws} (simulated with the fully coupled approach).

(for a system with $\lambda\tau \sim 1$). The flow regime in a dual-continuum system can be characterized by four dimensionless numbers: w_p , K_{fs}/K_{ms} , $\lambda\tau$, and K_{as}/K_{ms} .

With respect to the alternative methods for numerical coupling, the SC approach is relatively simple to program and leads to a computationally efficient solution of dual-continuum problems, provided that the time steps are reasonably small and the amount of water exchanged locally between the two flow domains in a time step is limited by a relatively high interfacial resistance. The FC approach is more numerically robust. Unlike the SC approach, it allows study of a complete class of intermediate flow regimes ranging from strongly preferential flow in a dual-continuum system to nonpreferential flow in a single-continuum system. This is achieved at the expense of somewhat decreased computational efficiency.

Table 2. Model performance (Intel Core2 6700 CPU, 2.69 GHz, 6 GB RAM) for the sequentially coupled (SC) approach in the soil matrix (SM) and preferential flow (PF) domains and the fully coupled (FC) approach.

Interfacial resistance	Cumulative number of iterations			Elapsed time	
	SC in SM domain	SC in PF domain	FC	SC	FC
h				s	
10,000	3,354	4,324	5,318	2.39	3.46
1,000	2,930	4,128	4,646	2.04	3.18
100	2,503	4,302	3,501	1.92	2.39
10	NC†	NC	4,597	–	2.96
1	NC	NC	11,196	–	6.76

† NC, failure to converge.

Appendix: List of Symbols

a	average soil aggregate radius, m
h_f	pressure head in the PF domain, m
h_m	pressure head in the SM domain, m
K_a	unsaturated hydraulic conductivity of the PF–SM interface, m s^{-1}
K_{ar}	relative unsaturated hydraulic conductivity of the PF–SM interface
K_{as}	saturated hydraulic conductivity of the PF–SM interface, m s^{-1}
K_{fr}	relative hydraulic conductivity of the PF domain
K_{fs}	saturated hydraulic conductivity of the PF domain, m s^{-1}
K_{mr}	relative hydraulic conductivity of the SM domain
K_{ms}	saturated hydraulic conductivity of the SM domain, m s^{-1}
K_s	saturated hydraulic conductivity of the bulk soil, m s^{-1}
r_{ws}	interfacial resistance at saturation, s
S_f	intensity of the root water uptake in the PF domain, s^{-1}
S_m	intensity of the root water uptake in the SM domain, s^{-1}
w_f	volumetric fraction of the pore space occupied by the PF domain
w_m	volumetric fraction of the pore space occupied by the SM domain
α_w	soil water transfer coefficient, $\text{m}^{-1} \text{s}^{-1}$
α_{ws}	soil water transfer coefficient at saturation, $\text{m}^{-1} \text{s}^{-1}$
β	geometry factor related to the shape of soil aggregates
Γ_w	soil water transfer term, s^{-1}
λ	characteristic length associated with the interdomain transfer of soil water, m
θ_f	soil water content in the PF domain
θ_m	soil water content in the SM domain
θ_s	saturated soil water content of the bulk soil
τ	specific interfacial area, m^{-1}

Acknowledgments

The research was supported by the EU FP6 project AquaTerra (no. 505428) under the thematic priority "Sustainable development, global change and ecosystems." Additional support was provided by the Ministry of Education of the Czech Republic (MSM 6840770002) and by the Czech Science Foundation (526/08/1016 and 205/09/P567).

References

- Beven, K.J., and P. Germann. 1982. Macropores and water flow in soils. *Water Resour. Res.* 18:1311–1325.
- Bouma, J. 1981. Soil morphology and preferential flow along macropores. *Agric. Water Manage.* 3:235–250.
- Cislerova, M., T. Vogel, J. Votrubova, and A. Robovska. 2002. Searching below thresholds: Tracing the origins of preferential flow within undisturbed soil samples. p. 265–274. *In* D. Smiles et al. (ed.) *Environmental mechanics: Water, mass and energy transfer in the biosphere*. Geophys. Monogr. Ser. 129. Am. Geophys. Union, Washington, DC.
- Deuflhard, P. 2004. Newton methods for nonlinear problems: Affine invariance and adaptive algorithms. Ser. Comput. Math. 35. Springer-Verlag, Berlin.
- Dusek, J., H.H. Gerke, and T. Vogel. 2008. Surface boundary conditions in two-dimensional dual-permeability modeling of tile drain bromide leaching. *Vadose Zone J.* 7:1241–1255.
- Gerke, H.H. 2006. Preferential flow descriptions for structured soils. *J. Plant Nutr. Soil Sci.* 169:382–400.
- Gerke, H.H., and J.M. Köhne. 2002. Estimating hydraulic properties of soil aggregate skins from sorptivity and water retention. *Soil Sci. Soc. Am. J.* 66:26–36.
- Gerke, H.H., and M.Th. van Genuchten. 1993a. A dual-porosity model for simulating the preferential movement of water and solutes in structured porous media. *Water Resour. Res.* 29:305–319.
- Gerke, H.H., and M.Th. van Genuchten. 1993b. Evaluation of a first-order water transfer term for variably saturated dual-porosity models. *Water Resour. Res.* 29:1225–1238.
- Gerke, H.H., and M.Th. van Genuchten. 1996. Macroscopic representation of structural geometry for simulating water and solute movement in dual-porosity media. *Adv. Water Resour.* 19:343–357.
- Jarvis, N.J. 1994. The MACRO model Version 3.1: Technical description and sample simulation. Rep. Diss. Dep. Soil Sci. Swed. Univ. Agric. Sci. 19:51.
- Jarvis, N.J. 2007. A review of non-equilibrium water flow and solute transport in soil macropores: Principles, controlling factors and consequences for water quality. *Eur. J. Soil Sci.* 58:523–546.
- Köhne, J.M., S. Köhne, and J. Šimunek. 2009. A review of model applications for structured soils: A) Water flow and tracer transport. *J. Contam. Hydrol.* 104:4–35.
- Köhne, J.M., B.P. Mohanty, J. Šimunek, and H.H. Gerke. 2004. Evaluation of a second-order water transfer term for variably saturated dual-permeability flow models. *Water Resour. Res.* 40:W07409, doi:10.1029/2004WR003285.
- Lewandowska, J., A. Szymkiewicz, K. Burzynski, and M. Vauclin. 2004. Modeling of unsaturated water flow in double-porosity soils by the homogenization approach. *Adv. Water Resour.* 27:283–296.
- Ray, C., T.R. Ellsworth, A.J. Valocchi, and C.W. Boast. 1997. An improved dual porosity model for chemical transport in macroporous soils. *J. Hydrol.* 193:270–292.
- Ray, C., T. Vogel, and J. Dusek. 2004. Modeling depth-variant and domain specific sorption and biodegradation in dual-permeability media. *J. Contam. Hydrol.* 70:63–87.
- Schaap, M.G., F.J. Leij, and M.Th. van Genuchten. 2001. ROSETTA: A computer program for estimating soil hydraulic parameters with hierarchical pedotransfer functions. *J. Hydrol.* 251:163–176.
- Šimunek, J., and M.Th. van Genuchten. 2008. Modeling nonequilibrium flow and transport processes using HYDRUS. *Vadose Zone J.* 7:782–797.
- Snehota, M., M. Sobotkova, and M. Cislerova. 2008. Impact of the entrapped air on water flow and solute transport in heterogeneous soil: Experimental set-up. *J. Hydrol. Hydromech.* 56:247–256.
- Tseng, P.H., A. Sciortino, and M.Th. van Genuchten. 1995. A partitioned solution procedure for simulating water flow in a variably saturated dual-porosity medium. *Adv. Water Resour.* 18:335–343.
- Vogel, T., H.H. Gerke, R. Zhang, and M.Th. van Genuchten. 2000. Modeling flow and transport in a two-dimensional dual-permeability system with spatially variable hydraulic properties. *J. Hydrol.* 238:78–89.
- Vogel, T., K. Huang, R. Zhang, and M.Th. van Genuchten. 1996. The HYDRUS code for simulating one-dimensional water flow, solute transport, and heat movement in variably-saturated media, version 5.0. Res. Rep. 140. U.S. Salinity Lab., Riverside, CA.
- Vogel, T., L. Lichner, J. Dusek, and A. Cipakova. 2007. Dual-continuum analysis of a cadmium tracer field experiment. *J. Contam. Hydrol.* 92:50–65.
- Zimmerman, R.W., G. Chen, T. Hadgu, and G.S. Bodvarsson. 1993. A numerical dual-porosity model with semianalytical treatment of fracture matrix flow. *Water Resour. Res.* 29:2127–2137.





Bibliography

- [1] Yechezkel Mualem. “A new model for predicting the hydraulic conductivity of unsaturated porous media”. *Water Resources Research* 12.3 (1976), PP. 513–522. DOI: 197610.1029/WR012i003p00513.
- [2] M. Th. van Genuchten. “A closed-form equation for predicting the hydraulic conductivity of unsaturated soils”. *Soil Science Society of America Journal* 44.5 (1980), pp. 892–898. DOI: 10.2136/sssaj1980.03615995004400050002x.
- [3] Hans Wilhelm Alt and Stephan Luckhaus. “Quasilinear elliptic-parabolic differential equations”. *Mathematische Zeitschrift* 183.3 (1983), pp. 311–341. ISSN: 0025-5874. DOI: 10.1007/BF01176474.
- [4] Franco Brezzi and Michel Fortin. *Mixed and Hybrid Finite Element Methods*. en. Springer Series in Computational Mathematics. New York: Springer-Verlag, 1991. ISBN: 978-1-4612-7824-5.
- [5] H. H. Gerke and M. T. van Genuchten. “Evaluation of a First-Order Water Transfer Term for Variably Saturated Dual-Porosity Flow Models”. *Water Resources Research* 29.4 (1993), PP. 1225–1238. DOI: 199310.1029/92WR02467.
- [6] J. Maryška, M. Rozložník, and M. Tůma. “Mixed-Hybrid Finite Element Approximation of the Potential Fluid Flow Problem”. *Journal of Computational and Applied Mathematics*. Proceedings of the International Symposium on Mathematical Modelling and Computational Methods Modelling 94 63.1 (Nov. 1995), pp. 383–392. ISSN: 0377-0427. DOI: 10.1016/0377-0427(95)00066-6.
- [7] T. Arbogast and M. Wheeler. “A Nonlinear Mixed Finite Element Method for a Degenerate Parabolic Equation Arising in Flow in Porous Media”. *SIAM Journal on Numerical Analysis* 33.4 (Aug. 1996), pp. 1669–1687. ISSN: 0036-1429. DOI: 10.1137/S0036142994266728.
- [8] Fakher Ben Belgacem. “The Mortar Finite Element Method with Lagrange Multipliers”. en. *Numerische Mathematik* 84.2 (1999), pp. 173–197. ISSN: 0029-599X, 0945-3245. DOI: 10.1007/s002110050468.
- [9] Luca Bergamaschi and Mario Putti. “Mixed Finite Elements and Newton-Type Linearizations for the Solution of Richards’ Equation”. en. *International Journal for Numerical Methods in Engineering* 45.8 (July 1999), pp. 1025–1046. ISSN: 1097-0207. DOI: 10.1002/(SICI)1097-0207(19990720)45:8<1025::AID-NME615>3.0.CO;2-G.
- [10] Jiří Maryška, Miroslav Rozložník, and Miroslav Tůma. “Schur Complement Systems in the Mixed-Hybrid Finite Element Approximation of the Potential Fluid Flow Problem”. *SIAM Journal on Scientific Computing* 22.2 (Jan. 2000), pp. 704–723. ISSN: 1064-8275. DOI: 10.1137/S1064827598339608.



- [11] Jiří Maryška, Otto Severýn, and Martin Vohralík. “Mixed-Hybrid FEM Discrete Fracture Network Model of the Fracture Flow”. en. *Computational Science — ICCS 2002*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Apr. 2002, pp. 794–803. ISBN: 978-3-540-43594-5. DOI: 10.1007/3-540-47789-6_82.
- [12] M. Bause and P. Knabner. “Computation of variably saturated subsurface flow by adaptive mixed hybrid finite element methods”. *Advances in Water Resources* 27.6 (June 2004), pp. 565–581. ISSN: 0309-1708. DOI: 10.1016/j.advwatres.2004.03.005.
- [13] Vincent Martin, Jérôme Jaffré, and Jean E. Roberts. “Modeling Fractures and Barriers as Interfaces for Flow in Porous Media”. *SIAM Journal on Scientific Computing* 26.5 (2005), pp. 1667–1691. ISSN: 10648275. DOI: 10.1137/S1064827503429363. (Visited on 04/18/2012).
- [14] Anis Younes, Philippe Ackerer, and François Lehmann. “A New Mass Lumping Scheme for the Mixed Hybrid Finite Element Method”. en. *International Journal for Numerical Methods in Engineering* 67.1 (July 2006), pp. 89–107. ISSN: 1097-0207. DOI: 10.1002/nme.1628.
- [15] T. Arbogast et al. “A Multiscale Mortar Mixed Finite Element Method”. *Multiscale Modeling & Simulation* 6.1 (Jan. 2007), pp. 319–346. ISSN: 1540-3459. DOI: 10.1137/060662587.
- [16] W. Bangerth, R. Hartmann, and G. Kanschat. “deal.II – A general-purpose object-oriented finite element library”. *ACM Trans. Math. Softw.* 33.4 (2007), p. 24. DOI: 10.1145/1268776.1268779.
- [17] Jan Šembera et al. “A Novel Approach to Modelling of Flow in Fractured Porous Medium”. eng. *Kybernetika* 43.4 (2007), pp. 577–588. ISSN: 0023-5954.
- [18] Tomáš F. F. et al. “On the incompatibility of Richards’ equation and finger-like infiltration in unsaturated homogeneous porous media”. *Water Resources Research* 45 (Mar. 2009), 12 PP. DOI: 200910.1029/2008WR007062.
- [19] Jan Březina and Milan Hokr. “Mixed-Hybrid Formulation of Multidimensional Fracture Flow”. en. *Numerical Methods and Applications*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Aug. 2010, pp. 125–132. ISBN: 978-3-642-18465-9 978-3-642-18466-6. DOI: 10.1007/978-3-642-18466-6_14.
- [20] Tomas Vogel et al. “Physical and Numerical Coupling in Dual-Continuum Modeling of Preferential Flow”. *Vadose Zone Journal* 9.2 (May 2010), pp. 260–267. DOI: 10.2136/vzj2009.0091.
- [21] Jan Březina et al. *Flow123d*. repository: <http://github.com/flow123d/flow123d>. 2011–2016. URL: <http://flow123d.github.com>.
- [22] Jakub Šístek, Jan Březina, and Bedřich Sousedík. “BDDC for Mixed-Hybrid Formulation of Flow in Porous Media with Combined Mesh Dimensions”. en. *Numerical Linear Algebra with Applications* 22.6 (Dec. 2015), pp. 903–929. ISSN: 1099-1506. DOI: 10.1002/nla.1991.

