TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Numerical methods for nonlinear flow in fractured porous medium

Diplomová práce

Studijní program: N3901 Aplikované vědy v inženýrství
Studijní obor: Aplikované vědy v inženýrství

Autor práce: **Bc. Sabina Bednářová**
Vedoucí práce: doc. Mgr. Jan Stebel, Ph.D.
Ústav nových technologií a aplikované informatiky

Liberec 2021

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Zadání diplomové práce

Numerical methods for nonlinear flow in fractured porous medium

Jméno a příjmení: **Bc. Sabina Bednářová**
Osobní číslo: M19000176
Studijní program: N3901 Aplikované vědy v inženýrství
Studijní obor: Aplikované vědy v inženýrství
Zadávající katedra: Ústav nových technologií a aplikované informatiky
Akademický rok: **2020/2021**

Zásady pro vypracování:

1. Nastudujte
   - problematiku proudění tekutin v porézním prostředí při velkých rychlostech, zejména Forchheimerovo zobecnění Darcyova zákona;
   - dimenzionální redukci Darcyovy-Forchheimerovy (DF) rovnice pro oblast obsahující puklinu;
   - matematickou teorii smíšené hybridní formulace pro eliptické rovnice;
   - metody řešení nelineárních algebraických soustav vzniklých diskretizací parciálních diferenciálních rovnic.
2. Odvoďte smíšenou hybridní formulaci DF rovnice v prostorové oblasti zahrnující protínající se pukliny. Aplikujte na ni tři zvolené linearizační metody.
3. Naimplementujte linearizované úlohy do výpočetního kódu Flow123d, ověřte na úloze s analytickým řešením a porovnejte konvergenci a výpočetní náročnost jednotlivých linearizací na jednoduché úloze sestávající ze dvou puklin.
4. Vytvořte výpočetní model benchmarkové úlohy s DF modelem proudění v puklinách.

Rozsah grafických prací: dle potřeby dokumentace
Rozsah pracovní zprávy: 40 – 50 stran
Forma zpracování práce: tištěná/elektronická
Jazyk práce: Angličtina

Seznam odborné literatury:

[1] I. Berre et al.: Verification benchmarks for single-phase flow in three-dimensional fractured porous media. arXiv preprint arXiv:2002.07005, 2020.
[2] F. Brezzi, M. Fortin: Mixed and Hybrid Finite Element Methods, Springer-Verlag, 1991.
[3] N. Frih, J. E. Roberts, A. Saada: Modeling fractures as interfaces: a model for Forchheimer fractures, Comput Geosci 12:91–104, 2008.
[4] C. T. Kelley: Iterative Methods for Linear and Nonlinear Equations.
North Carolina State University, 1995.
[5] I. S. Pop, W.-A. Yong: On the existence and uniqueness of a solution for an elliptic problem, Studia Univ. Babes-Bolyai Math. 45:97-107, 2000.
[6] R. W. Zimmermann, A. H. Al-Yaarubi, C. C. Pain and C. A. Grattoni: Non-linear regimes of fluid flow in rock fractures, Int. J.Rock Mech. Min., 41, 163–169, 2004.

Vedoucí práce: doc. Mgr. Jan Stebel, Ph.D.
Ústav nových technologií a aplikované informatiky

Datum zadání práce: 19. října 2020
Předpokládaný termín odevzdání: 17. května 2021

L.S.

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

Ing. Josef Novák, Ph.D.
vedoucí ústavu

V Liberci dne 19. října 2020

Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracovala samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Jsem si vědoma toho, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědoma povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má diplomová práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědoma následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

17. května 2021 Bc. Sabina Bednářová

Abstract

The main goal of this Master Thesis is the formulation, approximation and numerical solution of the Darcy-Forchheimer equation for non-linear flow in fractured porous media. For the numerical solution, three linearization methods have been chosen; Picard iterations, Newton's method and L-scheme method. All three of them have been implemented into the software *Flow123d*. The methods have been compared and verified on a simple analytical solution, and on a problem of two perpendicular fractures. Finally, one linearization method was chosen and tested on a benchmark case.

Keywords: Darcy-Forchheimer equation, non-linear partial differential equation, mixed hybrid approximation, finite element method, iterative methods, Picard iterations, Newton's method, L-scheme method, Flow123d, benchmark

Abstrakt

Hlavním cílem této diplomové práce je formulace, aproximace a numerické řešení Darcyovy-Forchheimerovy rovnice. Pro numerické řešení byly vybrány tři linearizační metody; Picardova iterace, Newtonova metoda a metoda L-schéma. Všechny tři linearizační metody byly naimplementovány do softwaru *Flow123d*. Metody byly nejdříve otestovány na jednoduchém analytickém řešení a na problému dvou kolmých puklin. Nakonec byla vybrána jedna linearizační metoda a otestována na benchmarkovém problému.

Klíčová slova: Darcyova-Forchheimerova rovnice, nelineární parciální diferenciální rovnice, smíšená hybridní formulace, metoda konečných prvků, iterační metody, Picardova iterace, Newtonova metoda, metoda L-schéma, Flow123d, benchmark

Acknowledgements

I would like to specially thank my supervisor doc. Mgr. Jan Stebel, Ph.D., for leading my work, for being of great help and support and for all the consultations and advice he has given me. I am also thankful to Ing. Pavel Exner, Ph.D., who quickly offered his help in some last minute arrangements.

I would also like to thank my family and my friends, who have always been there for me and supported me all the way.

Contents

List of Figures

List of Tables

List of Abbrevations

D-F	Darcy - Forchheimer
LHS	left hand side
PDE	partial differential equation
RHS	right hand side

List of Symbols

β	Darcy - Forchheimer constant
δ	cross-section coefficient
d	representative diameter for fractured or porous media
f	source
Γ	intersection of fractures
h	water level (in Chapter 1), hydraulic head (in Chapter 3)
$\mathbb{K}$	tensor of hydraulic conductivity
L	L-scheme constant
l	length of column
λ	Langrange multiplier
$\mathbf{n}$	normal vector
n	porosity
ν	kinematic viscosity
Ω	domain
p	pressure
Re	Reynolds number
S	cross-section of column
$\mathbf{u}$	Darcy velocity
$\mathbf{v}$	flux velocity
Q	volume flow

Introduction

Fluid flow through porous media has been becoming an important part of today's world. There are numerous examples of porous media playing critical roles in technology. Hydrology (and hydrogeology), which deals with the movement of water through the subsurface geological system, is one of the most important technologies that rely on the properties of porous media [1]. Some of the examples of applications are subsurface applications, including nuclear waste disposal [5], CO_2 storage [3], geothermal energy recovery [4] and hydraulic fracturing.

Porous media flow is usually described by the Darcy equation, where there is a linear relationship between the flux and the pressure gradient. However, it is well established that this relationship becomes nonlinear at sufficiently large Reynolds numbers. For these cases, the flow is supposed to follow the Forchheimer equation, where we find an extra term that depends quadratically on the flow rate [11].

For several years, the Darcy and Forchheimer equations have been used to describe flow in porous media and fractures. The Forchheimer regime is more likely to occur in fractures, as the fractures have a significant impact on the flow. With the Forchheimer term present in the equation, the equation becomes non-linear. Solution of such problem then becomes much more involved and its efficiency depends on the chosen linearization strategy.

This work consists of 3 chapters. In the first one we introduce the Darcy-Forchheimer equation. In the second chapter, we make the essential steps to derive the mixed-hybrid formulation from the D-F equation. We also describe 3 chosen linearization methods, explain the implementation into the software *Flow123d* and we present the results of our test problems. The last chapter focuses on the comparison of results from a chosen benchmark problem and our results.

1 Fluid flow in a porous medium at high velocities

Porous materials can be found in almost every aspect of everyday life, in technology and in nature. Essentially all solid and semi-solid materials are porous to some degree, with the exception of metals and some very dense rocks. In order for a medium to be defined as porous, a material or structure must have these two attributes [1]:

1. It must contain spaces, so-called voids or pores, free of solids, embedded in the solid or semi-solid matrix. The pores contain fluid, such as air, water, oil or a mixture of different fluids.

2. It must be permeable to a variety of fluids, i.e., fluids should be able to penetrate through one face of a sample of material and emerge on the other side.

The texture of a porous material is quite complicated. For example, a rock consists of mineral grains of various shapes and sizes and its pore structure is extremely complex. The most important factors of the pore structure are how much space there is between these grains and what their shapes are. That is because the spaces between these grains serve to either mainly transport fluids forming connecting pores, or to store the fluids forming storage pores. Basically, there are two types of porous materials based on the space between the grains: porous non-permeable (unconnected pore spaces) and porous permeable (connected pore spaces). However, it is quite impracticable to think about a rock in this microscopic measure. For the purpose of maths and physics, we describe the medium macroscopically and perform model homogenisation based on a representative elementary volume. We can only perform this if the size of pores is much smaller than the size of the medium. Physical quantity describing a rock through its pores is called porosity and is defined as:

$$n = \frac{V_{pores}}{V_{medium}}, \ 0 < n < 1, \tag{1.1}$$

where V_{pores} stands for the volume of pore space and V_{medium} stands for the volume

of the whole medium. Value of porosity in natural and commonly found rocks is usually $n < 0.4$.

The flow through a porous medium was famously described in 1855 by Henri Darcy [2], a French hydraulic engineer. He performed an experiment in order to understand the rates of water flow through sand layers. Darcy's experiments consisted of a vertical column filled with sand, with a water inlet at one end and an outlet at the other. The water pressure was controlled at the inlet and outlet ends of the column using reservoirs with constant water levels. Volume flow through the column is described as:

$$Q = \frac{S}{l} K(h_1 - h_2), \tag{1.2}$$

where Q is the volume flow, S is the cross-section of the column, l is the length of the column, K is conductivity and h_1, respectively h_2 are the water levels as described in Figure 1.1.

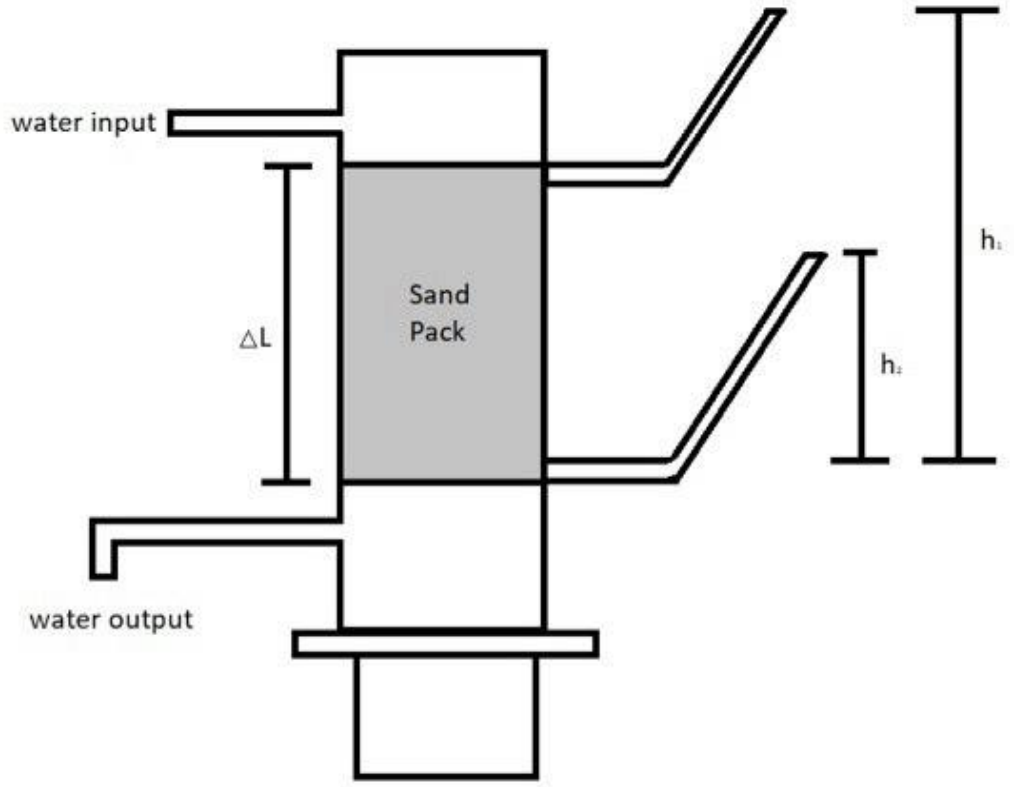


Figure 1.1: Darcy's experiment

It was Darcy's findings that laid the foundation for the modern science of hydrogeology. Darcy's law, refined by Morris Muskat [7], in the absence of gravitational

forces and in a homogeneously permeable medium, is given by a simple proportionality relationship between the instantaneous flux $\boldsymbol{u}$ (also called the Darcy velocity), the conductivity tensor $\mathbb{K}$ of the medium and the hydraulic gradient ∇p over a given distance, in the form:

$$\boldsymbol{u} = -\frac{Q}{S} = -\mathbb{K} \cdot \nabla p. \tag{1.3}$$

The "Darcy velocity" is not the velocity at which the fluid is travelling through the pores. Such velocity is called the flux velocity $\boldsymbol{v}$ and the relationship between the Darcy velocity $\boldsymbol{u}$ and flux velocity $\boldsymbol{v}$ is described by porosity n:

$$\boldsymbol{v} = \frac{\boldsymbol{u}}{n}. \tag{1.4}$$

As we can see from the equation, the Darcy velocity is always smaller than the actual flux velocity.

However, the Darcy law only applies to cases with slow, viscous flow, with a linear relationship between the flux and the pressure gradient, for which inertial effects are negligible. The Darcy equation is typically used for small Reynolds number, indicating clearly laminar flow in a porous or fractured medium. Reynolds number is a dimensionless number describing the behaviour of fluid flow. For flow through porous media the Reynolds number is expressed as:

$$Re = \frac{vd}{\nu} \tag{1.5}$$

where v is specific discharge (not the flow velocity), d is the representative diameter for porous media (for fractured media it would be the mean aperture of the fracture) and ν is the kinematic viscosity of the fluid.

However, for higher flow rates cases where the Reynolds number is typically greater than 1, the results of the Darcy law do not match the results of experiments; the flow rate predicted by the Darcy law is too high. At high flowrates, inertial effects (i.e., kinetic energy) become dominant, and we expect a pressure drop that is proportional to the velocity squared [12]. For this reason the so-called Darcy-Forchheimer equation was introduced, which contains a quadratic rate-dependent term which slows down the flow:

$$\mathbb{K}^{-1}\boldsymbol{u} + \beta|\boldsymbol{u}|\boldsymbol{u} + \nabla p = 0. \tag{1.6}$$

The term with β is the non-linear Forchheimer term. When $\beta = 0$, (1.6) reduces to the Darcy equation (1.3). When the flow is sufficiently rapid, the Forchheimer term

gives better results at the approximation of the relation between the gradient of the pressure and the flow rate [11].

Fractures are common in porous media and have significant impact on flow and transport. The permeability of the fracture, in particular, may be much higher than that of the host rock. As a result, while flow in the host rock can be well defined by Darcy's linear law, flow in the fractures can possibly indicate non-linear effects. Flow and transport issues in a porous medium containing fractures are encountered in a variety of energy and environmental applications, including carbon sequestration, geothermal energy, and ground-water pollution.

The Forchheimer term solves issues that the standard Darcy equation had for flows with high Reynolds number, however, the solution of such equation is rather difficult. Ideally, after time and space discretization of the problem, we want to have a system of linear algebraic equations (preferably with a symmetric positive definitive matrix). The direct discretization of the Darcy-Forchheimer equation would lead to a system of nonlinear equations. Solving a nonlinear system of equations requires appropriate linearization methods. The linearization methods that can be applied in order to solve the Darcy-Forchheimer equation will be introduced and studied in the following chapters of this work.

2 Numerical solution of Darcy-Forchheimer equations in a domain with fractures

The content of the chapter is the discretization and linearization of the Darcy-Forchheimer equation in a region formed by intersecting fractures. Generally, there can be many fractures with different geometries, but for simplicity of presentation, we show discretization on a simple area formed by two perpendicular fractures. The generalization to the case of any number of fractures and their intersection will be straightforward. The problem is discretized using the mixed hybrid formulation [9], [10] and linearized using 3 different linearization methods: Picard iterations, Newton's method and L-scheme.

Let Ω represent a union of two planar fractures Ω_i in $\mathbb{R}^3$ with boundaries $\partial\Omega_i$, $i = 1, 2$, where $\Omega_1 = (0, 1) \times (0, 1) \times \{0.5\}$ and $\Omega_2 = \{0.5\} \times (0.5, 1.5) \times (0, 1)$. The boundary $\partial\Omega_1 \cup \partial\Omega_2$ is decomposed into two parts Γ_D and Γ_N, $\boldsymbol{n}$ is the outer unit normal to the boundary and $\Gamma = \Omega_1 \cap \Omega_2$ represents the intersection of fractures.

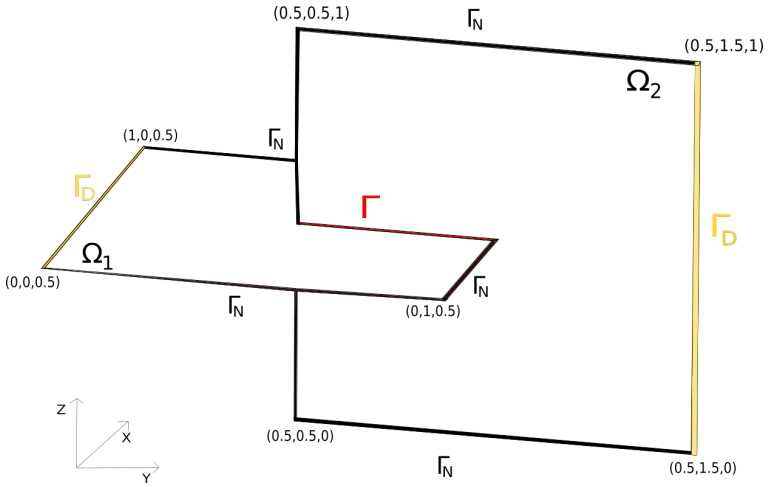


Figure 2.1: Geometry of the problem

The flow through Ω is described by the Darcy-Forchheimer equation with a scaling coefficient δ representing the cross-section of the fracture [11] to model a 3D

fracture as a 2D interface, and continuity equation for an incompressible fluid:

$$\left.\begin{aligned} \mathbb{K}^{-1}\boldsymbol{u} + \delta^{-1}\beta|\boldsymbol{u}|\boldsymbol{u} + \delta\nabla p = 0 \\ \nabla \cdot \boldsymbol{u} = f \end{aligned}\right\} \text{in } \Omega \setminus \Gamma, \tag{2.1}$$

where the unknowns are the pressure p and the velocity of the fluid $\boldsymbol{u}$. ∇ is the tangent gradient in Ω_1, respectively Ω_2, δ is a cross-section coefficient of the fracture and f is a scalar source term. For simplicity, we assume $\delta = 1$.

Let us denote $p_i := p_{|\Omega_i}$, $\boldsymbol{u}_i := \boldsymbol{u}_{|\Omega_i}$ (and similar for other functions) and let $\boldsymbol{n}_i$ be the outer unit normal to $\partial\Omega_i$, $i = 1, 2$. We define unit normal vectors $\boldsymbol{n}_i^+ = -\boldsymbol{n}_i^-$ on the set Γ lying in the plane Ω_i. Their orientation is arbitrarily chosen, but fixed. The function $\boldsymbol{u}_i$ has two values (traces) on Γ - we will denote these by $\boldsymbol{u}_i^+$ and $\boldsymbol{u}_i^-$, so that $\boldsymbol{u}_i^+$ is the trace from that side of Ω_i, which lies against the direction of the vector $\boldsymbol{n}_i^+$.

The boundary conditions are defined as follows:

$$p = p_D \text{ on } \Gamma_D, \tag{2.2}$$

$$\boldsymbol{u} \cdot \boldsymbol{n} = u_N \text{ on } \Gamma_N, \tag{2.3}$$

where p_D, u_N is a given sufficiently smooth pressure trace and flux, respectively. We consider the continuity of the normal flux and of the pressure at the intersection of fractures, i.e. the following conditions hold:

$$\left.\begin{aligned} &\textstyle\sum_{i=1,2}\sum_{*\in\{+,-\}} \boldsymbol{u}_i^* \cdot \boldsymbol{n}_i^* = 0 \\ &\text{p is continuous} \end{aligned}\right\} \text{on } \Gamma. \tag{2.4}$$

We declare $(\boldsymbol{u}, p)$ as the classical solution of the problem (2.1)-(2.4), if p is a continuous function in $\overline{\Omega}$ and continuously differentiable in $\Omega_1 \setminus \Gamma$ and $\Omega_2 \setminus \Gamma$, $\boldsymbol{u}$ is continuous in $\overline{\Omega} \setminus \Gamma$ and equations (2.1) - (2.4) are satisfied at each point of the relevant set.

2.1 Weak formulation

In what follows we shall introduce a numerical scheme for (2.1)-(2.4) based on the mixed hybrid formulation. We start by deriving the weak formulation. The continuity of the normal fluxes on Γ is enforced by Lagrange multipliers ($\lambda = p_{|\Gamma}$) [6]

and flux balance on Γ.

Let $L^p(\Omega_i)$ denote the Lebesgue space:

$$L^p(\Omega_i) = \{q : \Omega_i \to \mathbb{R}; \int_{\Omega_i} |q|^p dx < \infty\}$$

with the scalar product $(q_1, q_2)_{\Omega_i} = \int_{\Omega_i} q_1 q_2 dx$, $i = 1, 2$.

We define the space for velocity as:

$$V_{u_N} := \{\boldsymbol{v} : \Omega \to \mathbb{R}^3;\ \boldsymbol{v}_i \in (L^3(\Omega_i))^3,\ \nabla \cdot \boldsymbol{v}_i \in L^2(\Omega_i),\ i = 1, 2,\\ \boldsymbol{v} \cdot \boldsymbol{n} = u_N \text{ on } \Gamma_N,\ \ v_{1z} = 0,\ v_{2x} = 0\}$$

and the corresponding space for test functions:

$$V := \{\boldsymbol{v} : \Omega \to \mathbb{R}^3;\ \boldsymbol{v}_i \in (L^3(\Omega_i))^3,\ \nabla \cdot \boldsymbol{v}_i \in L^2(\Omega_i),\ i = 1, 2,\\ \boldsymbol{v} \cdot \boldsymbol{n} = 0 \text{ on } \Gamma_N,\ \ v_{1z} = 0,\ v_{2x} = 0\}.$$

We also define the space for pressure:

$$Q := \{q : \Omega \to \mathbb{R};\ q_{|i} \in L^2(\Omega_i),\ i = 1, 2\}$$

and the space for λ:

$$\Lambda = L^2(\Gamma).$$

To pass from the strong formulation (2.1)-(2.4) to weak formulation, we need to multiply the equations (2.1) by test functions, integrate them and apply the Green theorem. Let us focus on the third term from the first equation in (2.1).

If $\boldsymbol{v} \in V$, then:

$$\begin{aligned}(\nabla p, \boldsymbol{v})_{\Omega_i} &= (p_i \boldsymbol{n}_i, \boldsymbol{v}_i)_{\partial\Omega_i} + \sum_{*\in\{+,-\}} (p\boldsymbol{n}_i^*, \boldsymbol{v}_i^*)_\Gamma - (p, \nabla \cdot \boldsymbol{v})_{\Omega_i}\\ &= (p_D \boldsymbol{n}_i, \boldsymbol{v}_i)_{\Gamma_D \cap \partial\Omega_i} + (p_i, \underbrace{\boldsymbol{v}_i \cdot \boldsymbol{n}_i}_{=0})_{\Gamma_N \cap \partial\Omega_i}\\ &+ \sum_{*\in\{+,-\}} (p\boldsymbol{n}_i^*, \boldsymbol{v}_i^*)_\Gamma - (p, \nabla \cdot \boldsymbol{v})_{\Omega_i}.\end{aligned}$$

For the first equation in (2.1) we then get:

$$\sum_{i=1,2}\Big((\mathbb{K}^{-1}\boldsymbol{u}+\beta|\boldsymbol{u}|\boldsymbol{u},\boldsymbol{v})_{\Omega_i}-(p,\nabla\cdot\boldsymbol{v})_{\Omega_i}+(p_D,\boldsymbol{v}_i\cdot\boldsymbol{n}_i)_{\Gamma_D\cap\partial\Omega_i} + \sum_{*\in\{+,-\}}(p,\boldsymbol{v}_i^*\cdot\boldsymbol{n}_i^*)_\Gamma\Big)=0.$$

The weak formulation of (2.1)-(2.4) reads:
Find $(\boldsymbol{u},p,\lambda)\in V_{u_N}\times Q\times\Lambda$ such that:

$$\begin{aligned}\forall\boldsymbol{v}\in \mathrm{V}:\sum_{i=1}^{2}\Bigg(&\left(\mathbb{K}_i^{-1}\boldsymbol{u}_i+\beta|\boldsymbol{u}_i|\boldsymbol{u}_i,\boldsymbol{v}\right)_{\Omega_i}-(p_i,\nabla\cdot\boldsymbol{v}_i)_{\Omega_i}+\\ &\sum_{*\in\{+,-\}}(\lambda,\boldsymbol{v}_i^*\cdot\boldsymbol{n}_i^*)_\Gamma+(p_D,\boldsymbol{v}_i\cdot\boldsymbol{n}_i)_{\Gamma_D\cap\partial\Omega}\Bigg)=0,\\ \forall \mathrm{q}\in \mathrm{Q}:\sum_{i=1}^{2}(\nabla\cdot\boldsymbol{u}_i,q_i)_{\Omega_i}&=\sum_{i=1}^{2}(f_i,q_i)_{\Omega_i},\\ \forall\mu\in\Lambda:\sum_{i=1}^{2}\sum_{*\in\{+,-\}}&\left(\boldsymbol{u}_i^*\cdot\boldsymbol{n}_i^*,\mu\right)_\Gamma=0.\end{aligned}\tag{2.5}$$

2.2 Mixed hybrid formulation

In this section, we still describe the equations in infinite space. The finite-dimensional discretization will be done in following section. We consider triangulations $\epsilon_{h,i}$ of the domains Ω_i, i=1,2, and define: the set of all elements $\epsilon_h := \epsilon_{h,1}\cup\epsilon_{h,2}$, the set of element faces $S_h := \bigcup_{e\in\epsilon_{h,1}\cup\epsilon_{h,2}}\partial e\setminus\Gamma_D$ and the discretization parameter $h = max_{e\in\epsilon_h}\{diam\ e\}$ [9]. For any $e\in\epsilon_h$ we denote $p^e := p_{|e}$, $\boldsymbol{u}^e := \boldsymbol{u}_{|e}$ (and similarly for other functions).

We introduce the function space:

$$H(div,\epsilon_h)=\{\boldsymbol{v}:\Omega\to\mathbb{R}^3; v_{1z}=0, v_{2x}=0, \boldsymbol{v}_{|e}\in(L^3(e))^3; \nabla\cdot\boldsymbol{v}_{|e}\in(L^2(e))^2, \forall e\in\epsilon_h\}.$$

In the mixed-hybrid formulation of the problem (2.1)-(2.4), we replace the space V_{u_N} by $H(div,\epsilon_h)$, ie. the velocity can be discontinuous across element boundaries. To recover the continuity of the flux $\boldsymbol{u}\cdot\boldsymbol{n}$, the Lagrange multiplier is extended to S_n.

The mixed-hybrid formulation of (2.1)-(2.4) reads:

Find $(\boldsymbol{u}, p, \lambda) \in H(div, \epsilon_h) \times Q \times L^2(S_h)$ such that:

$$\begin{aligned}\forall \boldsymbol{v} \in H(\mathrm{div}, \epsilon_h) : \sum_{e \in \epsilon_h} \Bigg(& \left(\mathbb{K}^{-1}\boldsymbol{u}^e + \beta|\boldsymbol{u}^e|\boldsymbol{u}^e, \boldsymbol{v}^e\right)_e - (p^e, \nabla \cdot \boldsymbol{v}^e)_e + \\ & (\lambda, \boldsymbol{v}^e \cdot \boldsymbol{n}^e)_{\partial e \setminus \Gamma_D} + (p_D^e, \boldsymbol{v}^e \cdot \boldsymbol{n}^e)_{\partial e \cap \Gamma_D} \Bigg) = 0, \\ \forall q \in Q : \sum_{e \in \epsilon_h} (\nabla \cdot \boldsymbol{u}^e, q^e)_e &= \sum_{e \in \epsilon_h} (f^e, q^e)_e, \\ \forall \mu \in L^2(S_h) : \sum_{e \in \epsilon_h} \left(\boldsymbol{u}^e \cdot \boldsymbol{n}^e, \mu\right)_{\partial e \setminus \Gamma_D} &= (u_N, \mu)_{\Gamma_N}.\end{aligned} \tag{2.6}$$

2.3 Finite-dimensional approximation

The formulation (2.6) is based on a discretization of the set Ω. In this section we further discretize the equations using the lowest order Raviart-Thomas element [9] for the velocity and piecewise constant approximation of the pressure and its trace.

The velocity function $\boldsymbol{u}$ will be approximated by a discontinuous function linear on every element $e \in \epsilon_h$. We define the Raviart-Thomas space $RT^0(e)$:

$$RT^0(e) = \{\boldsymbol{v}^e; \boldsymbol{v}^e(x) = \sum_{j=1}^{3} \nu_j \boldsymbol{v}_j^e(x), x \in e\}$$

with linearly independent basis functions $\boldsymbol{v}_j^e$ such that if $e \subset \Omega_1$, then $v_{jz}^e = 0$, if $e \subset \Omega_2$, then $v_{jx}^e = 0$, $j = 1, .., 3$, and such that the functionals:

$$\mathcal{F}_j^e(\boldsymbol{v}_i^e) := \int_{f_j^e} \boldsymbol{n}_j^e \cdot \boldsymbol{v}_i^e dS$$

satisfy: $\mathcal{F}_j^e(\boldsymbol{v}_i^e) = \delta_{ij}, i, j = 1, ..., 3$.

Here f_j^e denotes the jth face of the element e and $\boldsymbol{n}_j^e$ is its outer normal vector. We also define a space of vector functions linear on each element

$$RT^0_{\epsilon_h} = \{\boldsymbol{v}_h : \Omega \to \mathbb{R}^3; \boldsymbol{v}_h^e \in RT^0(e), \forall e \in \epsilon_h\}.$$

Let us denote the space of scalar functions constant on an element e by M_e^0, and similarly the space of scalar functions constant on a face f by M_f^0. Next, we introduce a space for piece-wise constant functions on the elements in $_h$:

$$M^0_{\epsilon_h} = \{q_h : \Omega \to \mathbb{R}^3; q_h^e \in M_e^0, \forall e \in \epsilon_h\}.$$

Similarly the space

$$M^0_{S_h} = \{\mu_h : S_h \to \mathbb{R}; \mu_h^f \in M_f^0, \forall f \in S_h\}$$

consists of functions constant on every face of S_h. Furthermore, let $p_{D,h}$ and $u_{N,h}$ be the local L^2 projections of p_D and u_N, respectively, onto piece-wise constants:

$$\int_f (p_{D,h} - p_D) = 0; \forall f \in \bigcup_{e \in \epsilon_h} \partial e \cap \Gamma_D,$$

$$\int_f (u_{N,h} - u_N) = 0; \forall f \in \bigcup_{e \in \epsilon_h} \partial e \cap \Gamma_N.$$

Then we introduce the following approximation of the mixed-hybrid formulation: Find $(\boldsymbol{u}_h, p_h, \lambda_h) \in RT^0_{\epsilon_h} \times M^0_{\epsilon_h} \times M^0_{S_h}$:

$$\begin{aligned}
\forall \boldsymbol{v}_h \in RT^0_{\epsilon_h} : \sum_{e \in \epsilon_h} \Bigg(& \left(\mathbb{K}^{-1}\boldsymbol{u}_h + \beta|\boldsymbol{u}_h|\boldsymbol{u}_h, \boldsymbol{v}_h\right)_e - (p_h, \nabla \cdot \boldsymbol{v}_h)_e + \\
& (\lambda_h, \boldsymbol{v}_h^e \cdot \boldsymbol{n}^e)_{\partial e \setminus \Gamma_D} + (p_{D,h}, \boldsymbol{v}_h \cdot \boldsymbol{n}^e)_{\partial e \cap \Gamma_D} \Bigg) = 0, \\
\forall q_h \in M^0_{\epsilon_h} : & \sum_{e \in \epsilon_h} (\nabla \cdot \boldsymbol{u}_h, q_h)_e = (f_h, q_h)_e. \\
\forall \mu_h \in M^0_{S_h} : & \sum_{e \in \epsilon_h} \left(\boldsymbol{u}_h^e \cdot \boldsymbol{n}^e, \mu_h\right)_{\partial e \setminus \Gamma_D} = (u_N, \mu_h)_{\Gamma_N} \ .
\end{aligned} \tag{2.7}$$

2.4 Linearization of Forchheimer term

The set of equations (2.7) is a nonlinear problem with the unknowns $(\boldsymbol{u}_h, p_h, \lambda_h)$. In this section, we introduce three methods for solving the problem (2.7); Picard iteration, Newton's method and L-scheme. The Picard method is commonly used to solve nonlinear equations. It is easy to program and computationally inexpensive, but it has been known to fail or converge slowly. The Newton's method is more complex and expensive (on a per-iteration basis) than Picard iterations [13]. It is robust and has a quadratic rate of convergence, however its convergence is only local, which means that for the scheme to be convergent, the initial guess of the solution for the iterations must be near enough to the actual solution. To address this, we propose a fixed-term iteration scheme called L-scheme, a robust quasi-Newton method with a parameter $L > 0$ mimicking the Jacobian [14]. The L-scheme method is linearly convergent but it has the interesting property of unconditional

convergence, meaning that its convergence is independent of the choice of initial solution guess [14].

The article [13] discusses Picard and Newton's method for Richard's equation, however the behaviour of mentioned methods is generally valid.

2.4.1 Picard iterations

We introduce a Picard iteration with k as an iteration counter. Let $\boldsymbol{u}_n^k, k = 0, 1, ...$ be the approximation of the unknown $\boldsymbol{u}_h$ at k-th step. The solution using Picard iterations states:

$$\mathbb{K}^{-1}\boldsymbol{u}_n^{k+1} + \beta\left|\boldsymbol{u}_n^k\right|\boldsymbol{u}^{k+1} + \nabla p_n^k = 0. \tag{2.8}$$

Written in the mixed-hybrid formulation after the discretization of the problem: Find $(\boldsymbol{u}_h, p_h, \lambda_h) \in RT^0_{\epsilon_h} \times M^0_{\epsilon_h} \times M^0_{S_h}$ such that:

$$\begin{aligned}
\forall \boldsymbol{v}_h \in RT^0_{\epsilon_h} : \sum_{e\in\epsilon_h} \Bigg(& \left(\mathbb{K}^{-1}\boldsymbol{u}_h^{k+1} + \beta\left|\boldsymbol{u}_n^k\right|\boldsymbol{u}_h^{k+1}, \boldsymbol{v}_h\right)_e - \left(p_h^{k+1}, \nabla\cdot\boldsymbol{v}_h\right)_e + \\
& (\lambda_h^e, \boldsymbol{v}_h^e\cdot\boldsymbol{n})_{\partial e\setminus\Gamma_D} + (p_{D,h}, \boldsymbol{v}_h\cdot\boldsymbol{n})_{\partial e\cap\Gamma_D}\Bigg) = 0, \\
\forall q_h \in M^0_{\epsilon_h} : & \sum_{e\in\epsilon_h} (\nabla\cdot\boldsymbol{u}_h^{k+1}, q_h)_e = (f_h, q_h)_e, \\
\forall \mu_h \in M^0_{S_h} : & \sum_{e\in\epsilon_h} \left(\boldsymbol{u}_h^{k+1,e}\cdot\boldsymbol{n}^e, \mu_h\right)_{\partial e\setminus\Gamma_D} = (u_N, \mu_h)_{\Gamma_N}\ .
\end{aligned} \tag{2.9}$$

The iteration process is typically stopped when the change in the solution is below a given tolerance or when the residual of the equation is small enough.

2.4.2 Newton's method

The non-linear problem (2.7) can be written in the algebraic form [17] as:

$$\boldsymbol{F}(\boldsymbol{x}) = 0, \tag{2.10}$$

where $\boldsymbol{F} : \mathbb{R}^N \to \mathbb{R}^N$ and the vector $\boldsymbol{x} \in \mathbb{R}^N$ is the vector of degrees of freedom. The Newton method solves a linearized problem:

$$(\nabla\boldsymbol{F}(\boldsymbol{x}_k))\Delta\boldsymbol{x} = \boldsymbol{F}(\boldsymbol{x}_k) \tag{2.11}$$

where $\nabla \boldsymbol{F}(\boldsymbol{x}_k)$ is a Jacobian matrix and $\Delta \boldsymbol{x}$ is the vector of increments which defines a new approximation as: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta \boldsymbol{x}$.

Using the Newton method, we generate a sequence of approximations $\{(\boldsymbol{u}_h^k, p_h^k, \lambda_h^k)\}_{k=1}^{\infty}$ as follows. Let $(\boldsymbol{u}_h^k, p_h^k, \lambda_h^k) \in RT^0(\epsilon_h) \times M^0(\epsilon_h) \times M^0(S_h)$ be given. Then we define $(\boldsymbol{u}_h^{k+1}, p_h^{k+1}, \lambda_h^{k+1}) := (\boldsymbol{u}_h^k - \boldsymbol{\delta u}, p_h^k - \delta p, \lambda_h^k - \delta\lambda)$, where $(\boldsymbol{\delta u}, \delta p, \delta\lambda)$ is the solution of the linearized problem:

Find $(\boldsymbol{\delta u}, \delta p, \delta\lambda) \in RT^0_{\epsilon_h} \times M^0_{\epsilon_h} \times M^0_{S_h}$ such that $\forall (\boldsymbol{v}_h, q_h, \mu_h) \in RT^0_{\epsilon_h} \times M^0_{\epsilon_h} \times M^0_{S_h}$:

$$
\begin{aligned}
\sum_{e\in\epsilon_h} \Bigg(\Bigg(\mathbb{K}^{-1}\boldsymbol{\delta u} + \beta \left(\frac{(\boldsymbol{\delta u}\cdot \boldsymbol{u}_h^k)\boldsymbol{u}_h^k}{\left|\boldsymbol{u}_h^k\right|} + \left|\boldsymbol{u}_h^k\right|\boldsymbol{\delta u} \right), \boldsymbol{v}_h \Bigg)_e + (\delta\lambda, \boldsymbol{v}_h^e \cdot \boldsymbol{n}^e)_{\partial e\setminus\Gamma_D} \\
-(\delta p, \nabla\cdot \boldsymbol{v}_h)_e \Bigg) &= \sum_{e\in\epsilon_h} \Bigg(\left(\mathbb{K}^{-1}\boldsymbol{u}_h^k + \beta\left|\boldsymbol{u}_h^k\right|\boldsymbol{u}_h^k, \boldsymbol{v}_h\right)_e + (\lambda_h^k, \boldsymbol{v}_h^e\cdot\boldsymbol{n}^e)_{\partial e\setminus\Gamma_D} \\
&\quad - \left(p_h^k, \nabla\cdot\boldsymbol{v}_h\right)_e + (p_{D,h}, \boldsymbol{v}_h\cdot\boldsymbol{n})_{\partial e\cap\Gamma_D} \Bigg), \\
\sum_{e\in\epsilon_h} (\nabla\cdot\boldsymbol{\delta u}, q_h)_e &= \sum_{e\in\epsilon_h} (\nabla\cdot\boldsymbol{u}_h^k, q_h)_e - (f, q_h)_e, \\
\sum_{e\in\epsilon_h} (\boldsymbol{\delta u}^e\cdot\boldsymbol{n}^e, \mu_h)_{\partial e\setminus\Gamma_D} &= \sum_{e\in\epsilon_h} (\boldsymbol{u}_h^{k,e}\cdot\boldsymbol{n}^e, \mu_h)_{\partial e\setminus\Gamma_D} - (u_N, \mu_h)_{\Gamma_N}.
\end{aligned}
\tag{2.12}
$$

2.4.3 L-scheme method

The L-scheme method is a cross between Picard iterations and Newton's method. As mentioned before, it adds a new coefficient L, which approximates the Jacobian from Newton's method. Another way to describe it is looking at the new term with coefficient L; the difference between solutions mimics time discretization of a time-depend term. The L-scheme method reads:

Given initial approximation $\boldsymbol{u}_h^0$ and stabilization parameter $L > 0$, find for $k = 0, 1, 2, ...$ the functions $(\boldsymbol{u}_h^{k+1}, p_h^{k+1}, \lambda_h^{k+1}) \in RT_{\epsilon_h}^0 \times M_{\epsilon_h}^0 \times M_{S_h}^0$:

$$\begin{aligned}
\forall \boldsymbol{v}_h \in RT_{\epsilon_h}^0 : \sum_{e \in \epsilon_h} \Bigg(& \left(\mathbb{K}^{-1}\boldsymbol{u}_h^{k+1} + \beta\left|\boldsymbol{u}_h^k\right|\boldsymbol{u}_h^k + L\left(\boldsymbol{u}_h^{k+1} - \boldsymbol{u}_h^k\right), \boldsymbol{v}_h\right)_e - \left(p_h^{k+1}, \nabla \cdot \boldsymbol{v}_h\right)_e \\
& + (\lambda_h^{k+1}, \boldsymbol{v}_h^e \cdot \boldsymbol{n}^e)_{\partial e \setminus \Gamma_D} + (p_{D,h}, \boldsymbol{v}_h \cdot \boldsymbol{n})_{\partial e \cap \Gamma_D} \Bigg) = 0, \\
\forall q_h \in M_{\epsilon_h}^0 : & \sum_{e \in \epsilon_h} (\nabla \cdot \boldsymbol{u}_h^{k+1}, q_h)_e = (f_h, q_h)_e, \\
\forall \mu_h \in M_{S_h}^0 : & \sum_{e \in \epsilon_h} \left(\boldsymbol{u}_h^{k+1,e} \cdot \boldsymbol{n}, \mu_h\right)_{\partial e \setminus \Gamma_D} = (u_N, \mu_h)_{\Gamma_N} \ .
\end{aligned} \tag{2.13}$$

2.5 Implementation into Flow123d

In the following section we describe the implementation of the Darcy-Forchheimer equation and its three linearization methods into a software called *Flow123d* [15]. *Flow123d* is an open source simulator of underground water flow, solute and heat transport in fractured porous media. Computation on complex meshes consisting of simplicial elements of various dimensions is a unique feature of this program. As a result, we can use a combination of continuum and discrete fracture network models. Before any changes that we made, two water flow models had been available: the water flow model for a saturated medium based on the Darcy law and the model for partially saturated medium described by the Richards equation. Both models use the mixed-hybrid finite element method for the space discretization and the implicit Euler method for the time discretization. Since in this work we focus on saturated Darcy-Forchheimer flow, other parts of *Flow123d* will not be discussed but can be read in the User's manual [15]. *Flow123d* is coded in C/C++ using PETSc library for linear algebra. The input file is either YAML or JSON file.

The structure of *Flow123d* is quite complex, as the software can be used to solve more problems than just fluid flow in fractured media. For our work, we were only moving ourselves in the part called *flow*, which governs everything connected to fluid flow, including Richards equation. The two main files that we have been working on are called `assembly_mh.cc` and `darcy_flow_mh.cc`.

Since the Darcy equation had already been implemented before we started working on the D-F equation, the first rather simple step was to add the term with β in

order to make the D-F equation. The parameter β is an input parameter and when not given, the equation remains a Darcy equation.

Next we implemented the Picard method. When the user assigns $\beta \neq 0$ to a certain physical region, the solver switches to non-linear regime. This switch happens when the linear system is assembled in the method `DarcyMH::assembly_linear_system()`.

The main changes made in the code for Picard iterations are adding the respective term with β (see 2.4.1) into the already existing class `AssemblyMH`. Since the default value of β is zero, no other changes were needed in this class.

The implementation of Newton's method was a lot more complicated, as a whole new linear system and therefore a whole new class `AssemblyMH_Newton` had to be implemented. We took advantage of our previously implemented Picard iterations and the process of solving the D-F equation using Newton's method looks like:

1. Assemble linear system for Picard iterations and set its residual as the RHS for Newton's method
2. Assemble the matrix for the Newton method
3. Solve the linear system for Newton's method
4. Update solution

This time some changes in the solving process itself were needed, as well as making sure the user is able to switch between the linearization methods. We did that by implementing a new key called `solver_type` in the non-linear solver context, for which the default setting is Picard iterations. The user specifies the key in the YAML file. We implemented an `int` called `ns_type` that is used for switching between user specified methods (see the contructor of `DarcyMH`).

The implementation of the L-scheme method was a lot easier than the implementation of Newton's method, which is its advantage. Since the method is a cross between the Picard and Newton's method, a simple addition of respected terms from 2.4.3 into `AssemblyMH` was needed. This class uses the previously mentioned switch `ns_type` to switch between an assembly for Picard iterations or L-scheme method in its methods where we assembly the RHS and LHS. Besides that, we also needed to define the coefficient L, the same way as we defined the coefficient β. The detailed changes consist of over 700 new lines of code and can be seen on GitHub [16].

The coefficient L needs to be defined by the user in the YAML file, otherwise it's set to 0 and the L-scheme method will not function correctly.

For Newton's method the user input in the YAML file looks like:

```
flow_equation:  !Flow_Darcy_MH
  nonlinear_solver:
    solver_type:  Newton
    linear_solver:  !Petsc
  input fields:
    - region:  fracture
      beta:  1
```

The parameter β is specified for all the regions that are nonlinear, and for the regions that would remain linear (domain), we don't need to specify any value. For Picard iterations, the only change needed is to write `solver_type:  Picard` and for L-scheme, the set up would look like:

```
flow_equation:  !Flow_Darcy_MH
  nonlinear_solver:
    solver_type:  Lscheme
    linear_solver:  !Petsc
  input fields:
    - region:  fracture
      beta:  1
      l_param:  1
```

2.6 Analytical solution of the Darcy-Forchheimer equation

To verify the correct implementation of the methods, an analytical solution was derived for the problem in the square domain $(0,1) \times (0,1)$. Assuming a constant velocity field $\boldsymbol{u} = (u_x, u_y)$, the pressure can be expressed as:

$$p = x(-\mathbb{K}^{-1}u_x - \beta|\boldsymbol{u}|u_x) + y(-\mathbb{K}^{-1}u_y - \beta|\boldsymbol{u}|u_y). \tag{2.14}$$

The calculated pressure was prescribed as a boundary condition, and the resulting velocity was compared with the input data of the analytical solution. The performance of the linearization methods was measured in the overall number of iterations needed to reach the stopping criteria, mainly because the computational complexity of 1 iteration is similar in all of the three methods. Stopping criteria used for nonlinear solver is the residual tolerance `tolerance =` 10^{-6} (the maximum number of

iterations was always set very high, so the solver would never stop because of reaching that point), and for linear solver we used the residual tolerance `r_tol` = 10^{-10} and absolute residual tolerance `a_tol` = 10^{-10}. All calculations were performed on a computer with an Intel (R) Core (TM) i5-8250U CPU @ 1.60GHz 1.80GHz and 8GB of RAM. We present the results of the comparison on two model settings:

1. $\boldsymbol{u} = (2, 2), \mathbb{K} = \mathbb{I}$

   For a visual representation of the solution of this problem, see Figure 2.2.

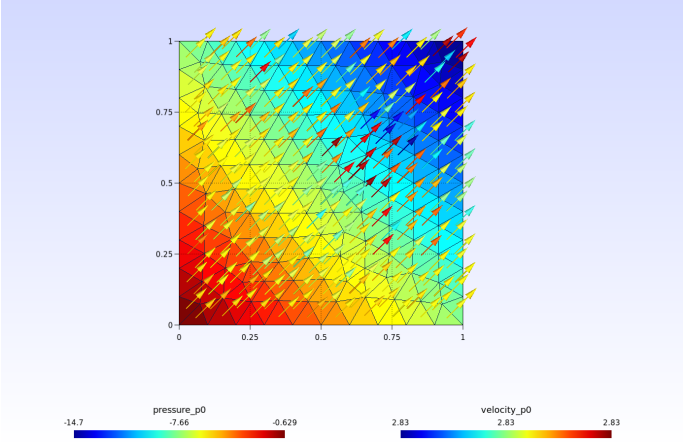


Figure 2.2: Visualisation of results for problem 1, $\beta = 1$, Picard iterations

We considered 18 values of β from 0.1 to 5, the other parameters remained fixed. The resulting number of iterations for each method can be seen in the Table 2.1 and Figures 2.3-2.4.

Table 2.1: Number of iterations for all 3 linearization methods for selected values of β, problem 1.

beta	**Picard iterations**	**Newton's method**	**L-scheme method**
0.5	31	5	5
1	56	6	6
2	107	7	6
4	214	8	6

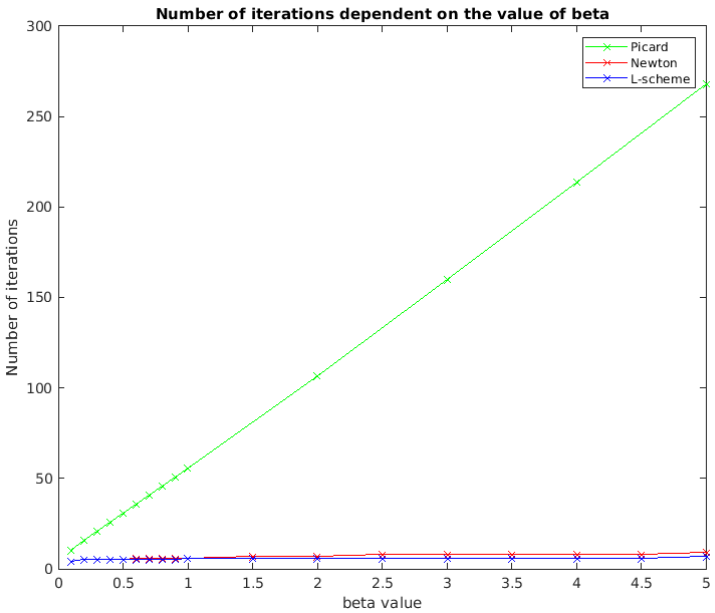


Figure 2.3: Number of iterations dependent on the value of coefficient β, problem 1.

As we can see, the number of iterations for Picard's method rises linearly and is already from the beginning much higher than for the Newton and L-scheme method. For a strongly non-linear problem it performs quite slow and requires a high number of iterations. A detailed comparison of iteration count for the Newton and L-scheme method is depicted in Figure 2.4.

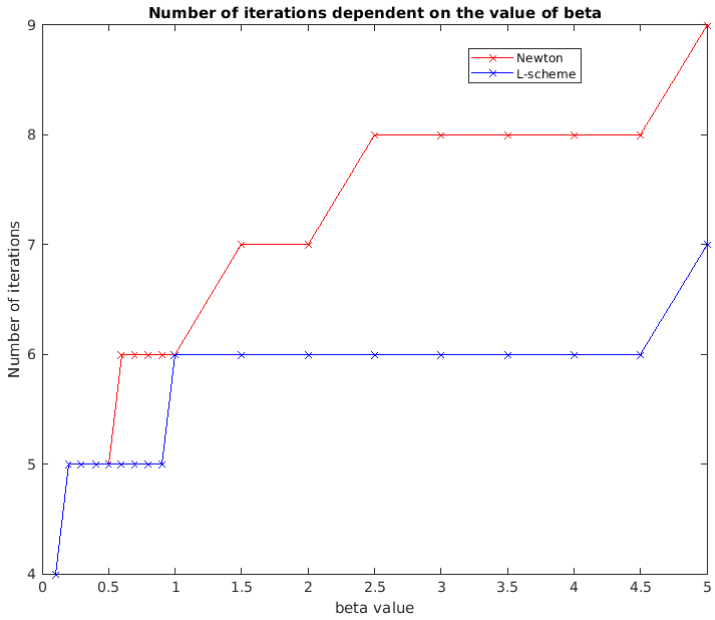


Figure 2.4: Number of iterations dependent on the value of coefficient β for Newton's and L-scheme method, problem 1.

The number of iterations of these two methods grows very slowly, even for a strongly non-linear problem. The L-scheme method performs better than the Newton's method, but it's performance is very closely connected to the right choice of coefficient the L. For each value of β , the best L coefficient was chosen. The results can be seen in Table 2.2.

Table 2.2: The value of coefficient L for each different value of coefficient β (problem 1 and 2).

β **value**	**L value (problem 1)**	**L value (problem 2)**
0.1	0.55	0.8
0.2	1.1	1.8
0.3	1.7	2.7
0.4	2.3	3.5
0.5	2.8	4.5
0.6	3.3	5.5
0.7	3.9	6
0.8	4.7	7
0.9	5.1	8
1	5.5	9
1.5	8.5	13
2	11	18
2.5	14	22
3	17	27
3.5	20	31
4	23	36
4.5	25	40
5	27	45

To briefly demonstrate the importance of the value of L coefficient, let us fix the value of $\beta = 1$ and other parameters in problem 1 and change the value of L, from 1 to 10 with a uniform step of 0.5.

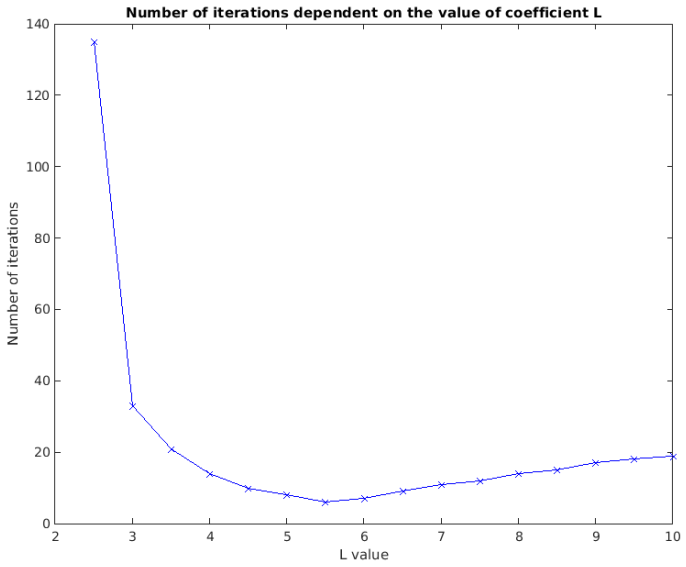


Figure 2.5: Number of iterations depending on the value of coefficient L, problem 1.

As we can see in Figure 2.5, the value of L is crucial to the number of iterations in which the solution converges. For values of L below 2.5 the problem did not even converge, as the residual was increasing instead of decreasing. The lowest number of iterations needed to solve the problem was at $L = 5.5$ and with the increase of the L coefficient, the number of iterations was increasing as well. However, further theoretical study of the choice of L coefficient is not provided in this work.

2. $\boldsymbol{u} = (4, -2), \mathbb{K} = \mathbb{I}$

   For a visual representation of the solution of this problem, see Figure 2.6.

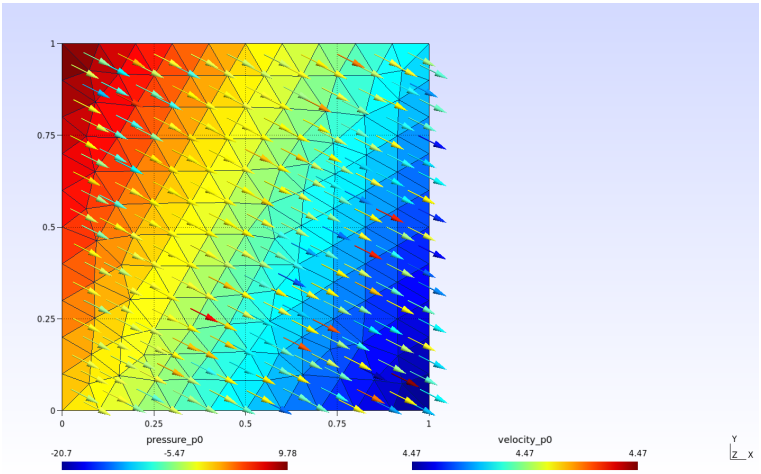


Figure 2.6: Visualisation of results for problem 2, $\beta = 1$, L-scheme method

Again, we consider 18 values of β from 0.1 to 5, just like in problem 1, and the other parameters remained fixed.

Table 2.3: Number of iterations for all 3 linearization methods for $\beta = 1$, problem 2.

beta	**Picard iterations**	**Newton's method**	**L-scheme method**
0.5	40	6	5
1	88	7	6
2	173	8	6
4	X	9	6

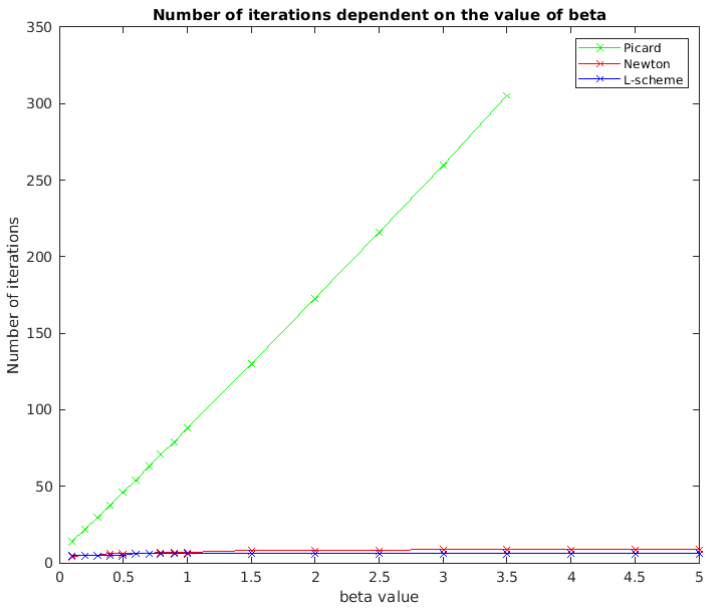


Figure 2.7: Number of iterations dependent on the value of coefficient β, problem 2.

The Picard method performed similarly as in problem 1, linearly rising with the coefficient β. However, because of higher velocity magnitude, it did not converge for $\beta > 3.5$ and the residual of the equations was oscillating.

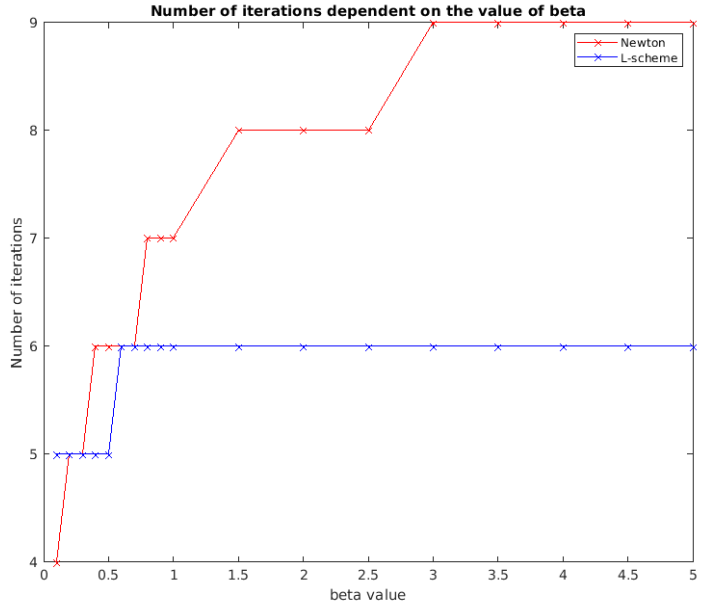


Figure 2.8: Number of iterations dependent on the value of coefficient β for Newton's and L-scheme method, problem 2.

The values of L can be seen in Table 2.2. We can observe that on the same

problem with just higher velocity magnitude, the value of the coefficient is also higher.

In other aspects, the three methods behaved very similarly to problem 1.

2.7 Solution of the model of two perpendicular fractures

In this section, we present the numerical solution in the domain consisting of two perpendicular fractures defined in the beginning of the section 2 and show how the linearization methods behave for different values of parameter β. Other parameters that were fixed include: hydraulic conductivity $\mathbb{K} = \mathbb{I}$, cross section of fractures $cs = 1e-3$, hydraulic head on the inlet 1 and hydraulic head on the outlet 0; which means that the fluid flow is controlled by the pressure drop. The stopping criteria were the same as in section 2.6.

The mesh for this model contains 3444 elements and is locally refined at the line of intersection of the fractures, where 45 equally large elements were set for the intersection, and on the vertical wall of the second domain, 20 gradually increasing elements were set in the direction from the intersection, see Figure 2.9.

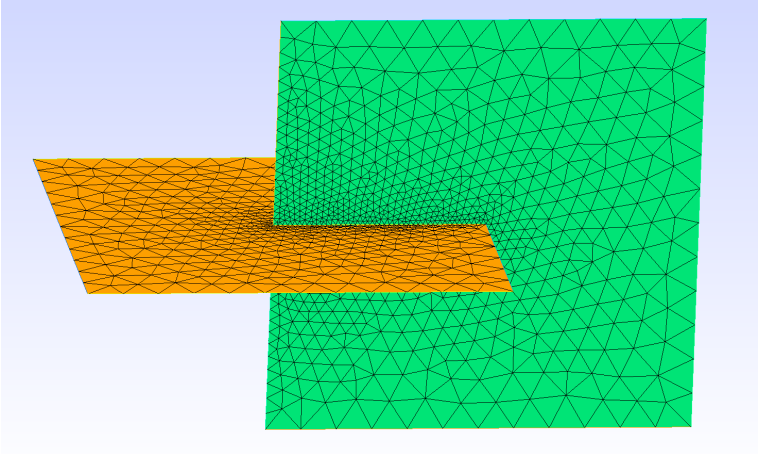


Figure 2.9: Mesh for the problem of two perpendicular fractures.

The problem was solved using the three linearization methods for 21 values of β, ranging from 0.1 to 100.

Table 2.4: Number of iterations for all 3 linearization methods for $\beta = 1$, problem of two fractures.

beta	Picard iterations	Newton's method	L-scheme method
0.1	5	3	5
0.5	10	4	8
1	13	4	10
2	18	5	13
4	25	6	17
8	34	6	22
10	34	7	22
100	113	9	37

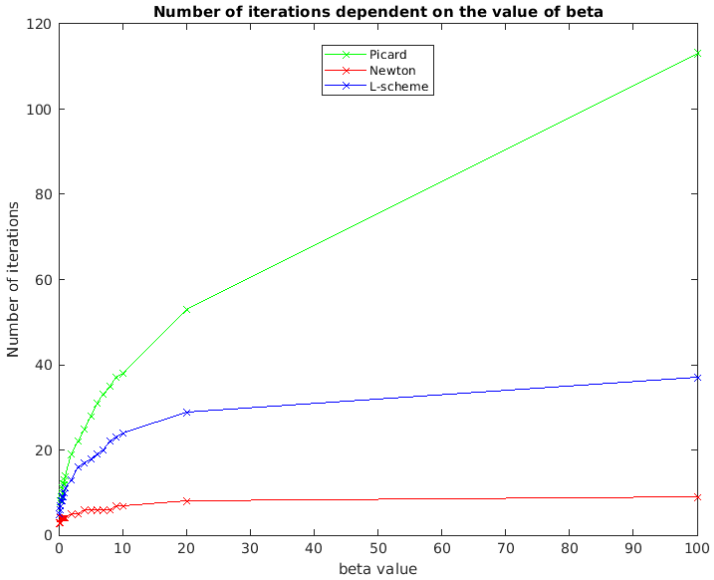


Figure 2.10: Number of iterations dependent on the value of coefficient β, problem of two fractures.

As we can see from in Figure 2.10, Picard method did converge on all the tests, but still had the highest amount of iterations from all the methods. On the other hand, here we can see that Newton's method performed better than L-scheme, and that overall, the number of iterations is way lower than in the testing we did for the two analytical problems. That is most likely because the pressure drop was not sufficient enough to create high velocity magnitude, as we can see in Figure 2.12. The reason behind Newton's method performing better than L-scheme on this kind of problem is most likely because L-scheme doesn't behave that well when the velocity field is singular. We can observe some singularities in Figure 2.12, at the intersection of the two fractures.

Table 2.5: The value of coefficient L for each different value of coefficient β, problem of two fractures.

β **value**	**L value (coarse grid)**	**L value (finer grid)**
0.1	0.1	0.1
0.2	0.25	0.25
0.3	0.4	0.5
0.4	0.45	0.65
0.5	0.6	0.8
0.6	0.7	0.85
0.7	0.75	0.9
0.8	0.9	1
0.9	0.9	1.1
1	1.1	1.2
2	1.6	2
3	2.2	2.4
4	2.6	3.2
5	2.9	3.9
6	3.2	4.2
7	3.5	4.6
8	4.2	5
9	4.6	5.3
10	4.9	5.8
20	7	8.4
100	17.5	20

In Table 2.5 we can look into the values of coefficient L. We can see that the values are overall smaller than in Table 2.2, that is because of the fact that the velocity magnitude is not as high as in the analytical problems.

We have also examined the dependency of L-scheme method on the domain discretization. We refined the mesh, so it would count roughly 3x more elements. Both Picard iterations and Newton's method did not change the number of iterations they needed to solve the equations. L-scheme method did need slightly more then before, however the difference is not immense, see Figure 2.11.

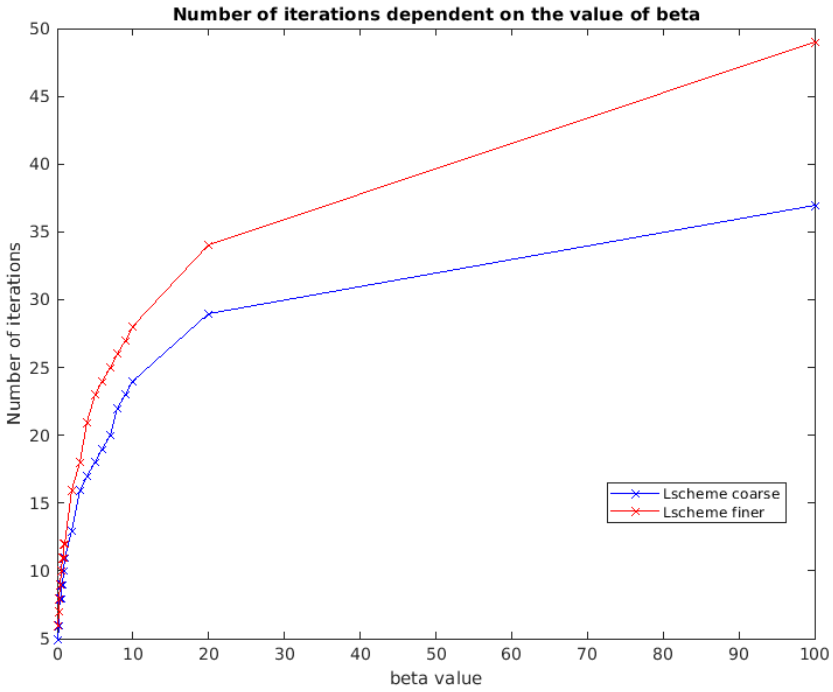


Figure 2.11: Comparison of L-scheme method on finer and coarser grid.

Having a look again in Table 2.5, the value of L for a case with a finer grid is slightly different too, but just like the difference between the iterations, it is not large. For visual representation of the solution, see Figure 2.12.

For this problem, we also measured the computation complexity. It was performed on both grids, with the fixed value $\beta = 1$. We can see that the Picard method is the slowest and Newton's method the fastest, which was to be expected, see Table 2.6.

Table 2.6: Computational complexity for all three methods, $\beta = 1$

Method	**Mesh**	**Computational time [s]**
Picard iterations	coarse	18.8
Newton's method	coarse	10.8
L-scheme method	coarse	16.0
Picard iterations	finer	60.0
Newton's method	finer	41.8
L-scheme method	finer	50.7

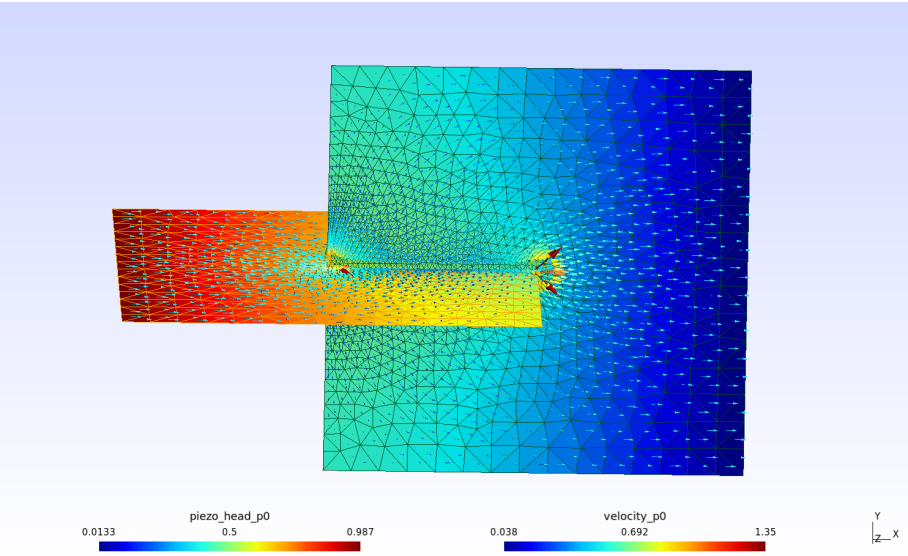


Figure 2.12: Visualisation of the solution for two perpendicular fractures (Newton's method, coarse grid), $\beta = 1$.

3 Solution of computational benchmark problem

In this chapter, we demonstrate our implementation of the D-F equation on a benchmark problem defined in [18] using one chosen linearization method. The article mentions 4 benchmark problems, from which we chose one (Case 2.1.) We chose a 3D problem based on a synthetic network composed of nine, regularly oriented fractures. The benchmark problem itself only describes the flow as Darcian, however, since the fractures are highly conductive compared to the matrix, we will also consider the Darcy-Forchheimer equation in the fractures and their intersections.

The domain is given by the unit cube $\Omega = (0,1) \times (0,1) \times (0,1)$ and contains 9 perpendicular fractures, see Figure 3.1. The geometry of the problem can be downloaded from the repository on Github [19].

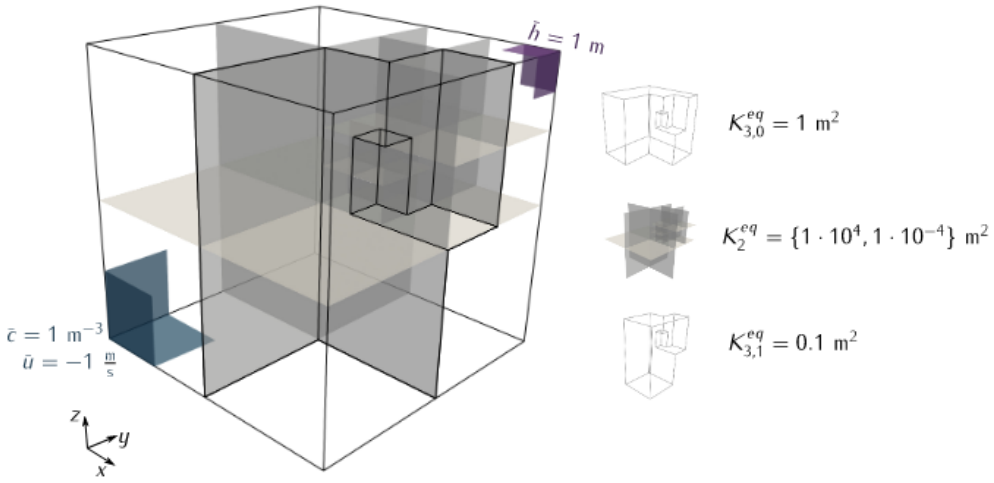


Figure 3.1: Representation of the domain and fractures for benchmark problem [18]. Here the inlet and outlet are coloured blue and purple, respectively.

The domain is divided into two sections, $\Omega_{3,0}$ and $\Omega_{3,1}$:

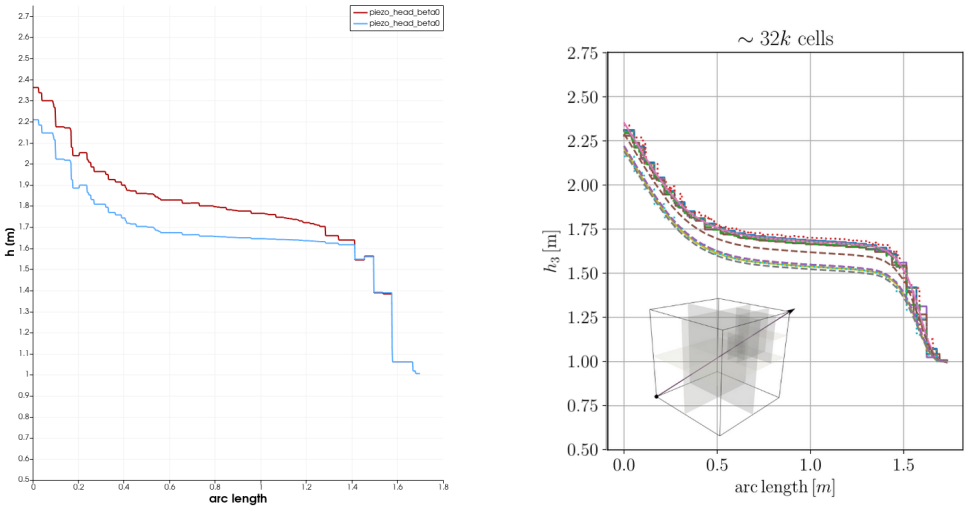
$$\begin{aligned}
\Omega_{3,0} &= \Omega_3 \setminus \Omega_{3,1} \\
\Omega_{3,1} &= \{(x,y,z) \in \Omega_3 : x > 0.5 \text{ m} \wedge y < 0.5 \text{ m}\} \\
&\quad \cup \{(x,y,z) \in \Omega_3 : x > 0.75 \text{ m} \wedge 0.5 \text{ m} < y < 0.75 \text{ m} \wedge z > 0.5 \text{ m}\} \\
&\quad \cup \{(x,y,z) \in \Omega_3 : 0.625 \text{ m} < x < 0.75 \text{ m} \wedge 0.5 \text{ m} < y < 0.625 \text{ m} \\
&\quad \wedge 0.5 \text{ m} < z < 0.75 \text{ m}\}
\end{aligned}$$

which vary in the value of conductivity $\mathbb{K}_{3,0}$ and $\mathbb{K}_{3,1}$, respectively. The fractures are marked as Ω_2 and the value of conductivity for fractures $\mathbb{K}_2$. The boundary $\partial\Omega$ is divided into three sections, each corresponding to a chosen boundary condition. First, $\partial\Omega_h = \{(x,y,z) \in \partial\Omega : x,y,z > 0.875 \text{ m}\}$ is a part of the boundary on which we impose the Dirichlet condition $h = 1$ m, where h stands for the hydraulic head. Second, we set a total flux boundary condition on $\partial\Omega_{in} = \{(x,y,z) \in \partial\Omega : x,y,z < 0.25 \text{ m}\}$, $u_N = 1$ m/s. On the remaining parts of $\partial\Omega$ we impose no-flow conditions. It is worth noting that different notation for conductivity was used in the article. For us $K^{eq}_{3,0} = \mathbb{K}_{3,0}$ and similarly for other indexes. The article uses the notation of $\mathbb{K}$ for a bit different conductivity, and the conversion between $\mathbb{K}$ and K^{eq} is explained in the article [18] in equations 6*a* and 6*b*.

In order to be able to compare our results with the benchmark results, we also created a transport problem on the same domain. The transport problem uses the so called tracer, a solute that is prescribed by a given concentration on the inlet boundary $c = 1 \text{ m}^{-3}$. A complete overview of the parameters used in this benchmark case is given in Table 3.2. The transport problem allows us to verify the velocity, as we can observe the mean concentration over time in the matrix. The transport problem consists of only advective flow (no diffusion was considered) and the equations used for this problem had already been implemented in *Flow123d* before we started working on D-F equation. More about the transport equation can be read in the User's manual [15].

The results were collected on two variants of the benchmark problem: one where we considered only steady Darcian flow, just like in the article, and one where we considered steady Darcy flow in the domain and steady Darcy-Forchheimer flow in the fractures. We compared these two together with the results from benchmark. The mesh created for this problem used approximately 32k cells. For the non-linear problem, we chose Newton method for linearization and the value of $\beta = 1$. This choice has no particular physical reason and it is only a numerical test.

First, we compared the results of hydraulic head over line the line $(0,0,0) - (1,1,1)$, see Figure 3.2. The hydraulic head for the case of only Darcy flux has lower values than the one with the D-F flow, however, by the end of the line, the values are the same. That is because we prescribed a Dirichlet boundary condition for the outlet, which is where the abscissa ends, so both cases had to have the same value. The results from the article correspond to our results of $\beta = 0$. The benchmark problem also compared different numerical methods (see that Figure 3.2 (b) contains a lot of them). For the comparison, we went with a case that had the same number of mesh cells as ours, where the difference in solution between numerical methods is negligible.



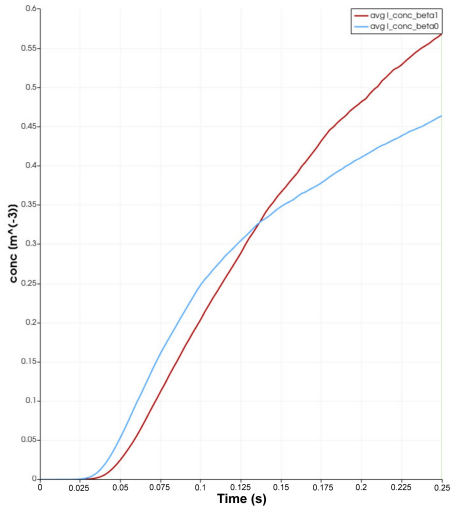
(a) Results obtained by us for $\beta = 0$ and $\beta = 1$.

(b) Benchmark results.

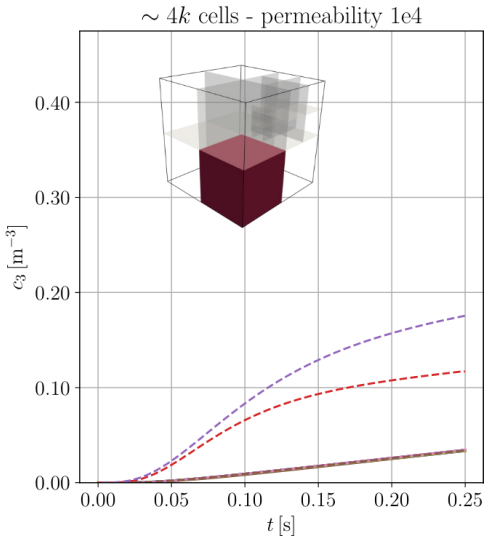
Figure 3.2: Comparison of our results and the results from the article [18] for hydraulic head over the line (0,0,0)-(1,1,1).

Second, we compared mean matrix concentration over time in three subregions of the domain.This concerns the solution of the transport equation. Looking at the first comparison in Figure 3.3, we can see that the time at which the concentration begins to develop in that region is the same for our results (linear case) and the benchmark results. They differ in the value of average concentration. It is also worth noting that the benchmark mesh only contained around 4k cells, but ours contained 32k cells. Another inaccuracy may be caused by 0D dimensional elements. The intersection volume is defined by physical points in the `geo` file. Unfortunately, physical points that are not part of the boundary are not yet supported by *Flow123d*. However, the points are just points, and the difference should not be significant. Looking at the

difference between the results for linear and non-linear case, we can see that there is a certain retardation in the development of the concentration for the nonlinear case. That is because the Forchheimer term in the equation was introduced to cases with high velocities and its impact is the decrease of the flow velocity.



(a) Results obtained by us for $\beta = 0$ and $\beta = 1$.



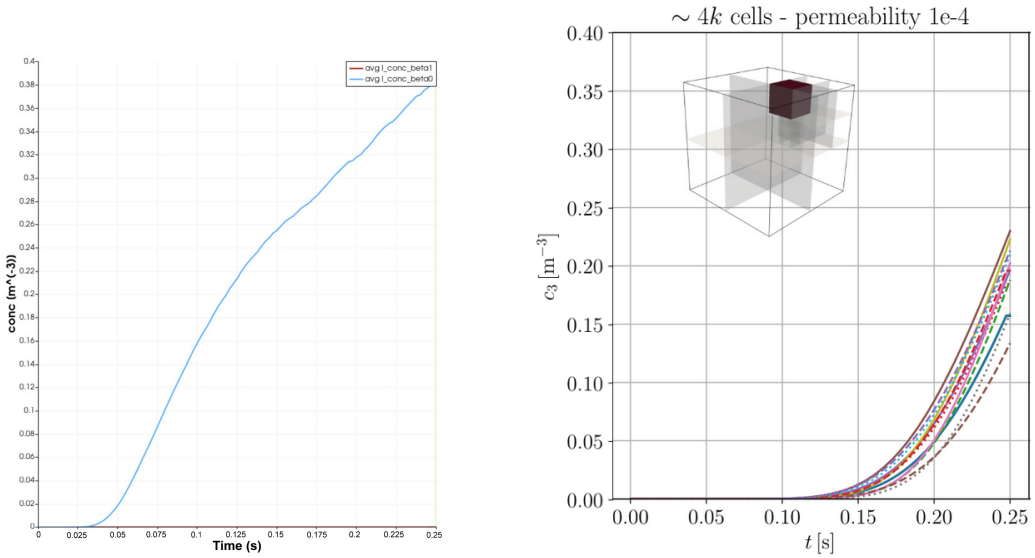
(b) Benchmark results.

Figure 3.3: Comparison of our results and the results from the article [18] for mean matrix concentration over time in the first region.

In the second region (see Figure 3.4), the average concentration in the top middle region in the nonlinear case was almost zero. That is because at the end time $t = 0.25$ s, there was almost no tracer in the region yet, as the flow is slower than in the linear case. Almost identical behaviour can be seen in the next comparison (see Figure 3.5). The concentration for only the non-linear case in the second and third region can be seen in Figure 3.6. For that reason, we decided to run the transport equation again, but this time with $t = 1$ s, $\Delta t = 0.01$ s and with a locally refined mesh at the intersections of fractures. The mesh was now using 120k elements. The results can be seen in Figure 3.7. The regions through which we calculated the average concentration remain the same.

In the first region, which is the closest region of the three to inlet, we can see a very small retardation in the nonlinear case compared to the linear case. The concentrations settle at a very similar value, the settlement for the non-linear case happens slightly earlier than for the linear case. In the second region, the retardation is way more prominent and we can see that the non-linear concentration curve is way

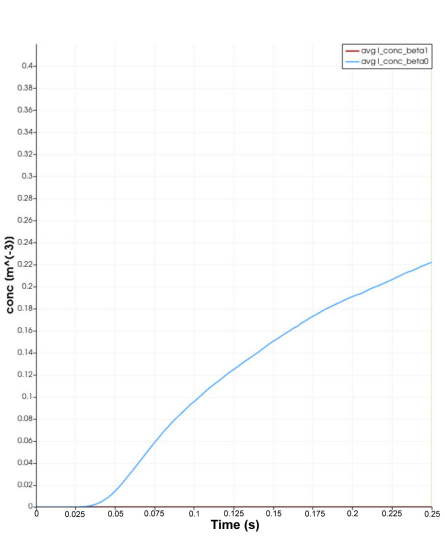
steeper than for the linear case. Again, they settle at a very similar value. The reason behind the concentration settling at a very similar value as in the first region is that the second region is a part of $\Omega_{3,0}$, a part of domain with the higher conductivity. In the last region, we can see the biggest retardation, with the concentration for the non-linear case being very low. That is because this region is the furthest away from the inlet, so the impact of the Forchheimer term can be observed very well here and because the region is a part of the domain $\Omega_{3,1}$ where the conductivity is lower than in the fractures or the domain $\Omega_{3,0}$.



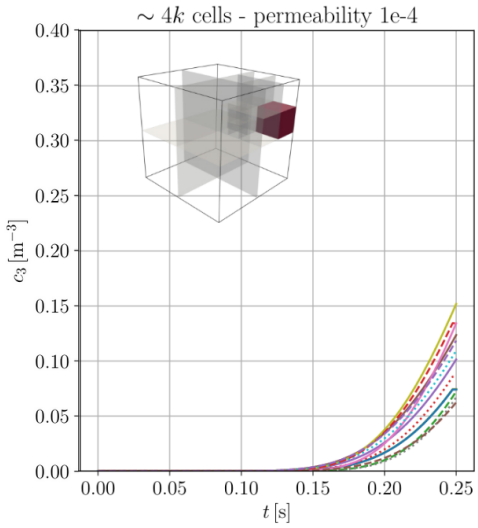
(a) Results obtained by us for $\beta = 0$ and $\beta = 1$.

(b) Benchmark results.

Figure 3.4: Comparison of our results and the results from the article [18] for mean matrix concentration over time in the second region.

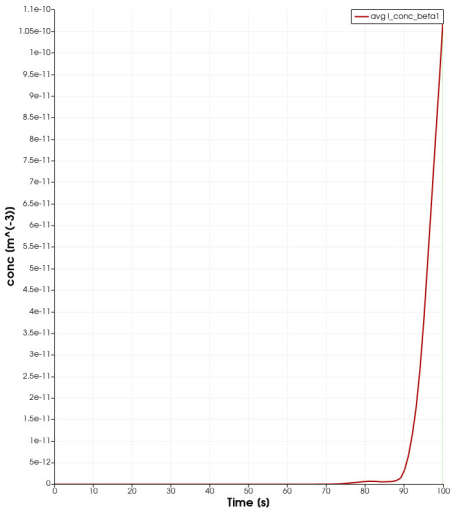


(a) Results obtained by us for $\beta = 0$ and $\beta = 1$.

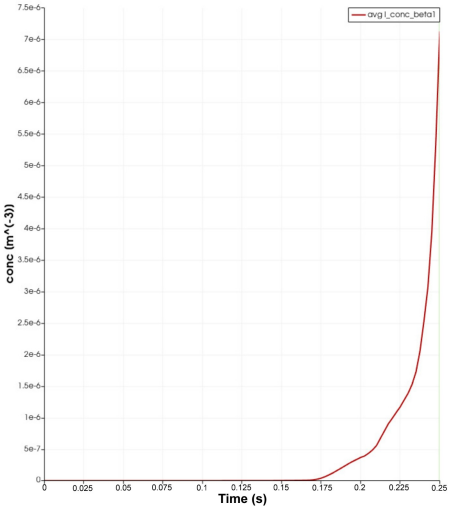


(b) Benchmark results.

Figure 3.5: Comparison of our results and the results from the article [18] for mean matrix concentration over time in the third region.



(a) Results for the non-linear case, second region.



(b) Results for the non-linear case, third region.

Figure 3.6: Comparison of the results of mean matrix concentration for the non-linear case in second and third region.

For convenience, we have included the benchmark case's computational time, see Table 3.1.

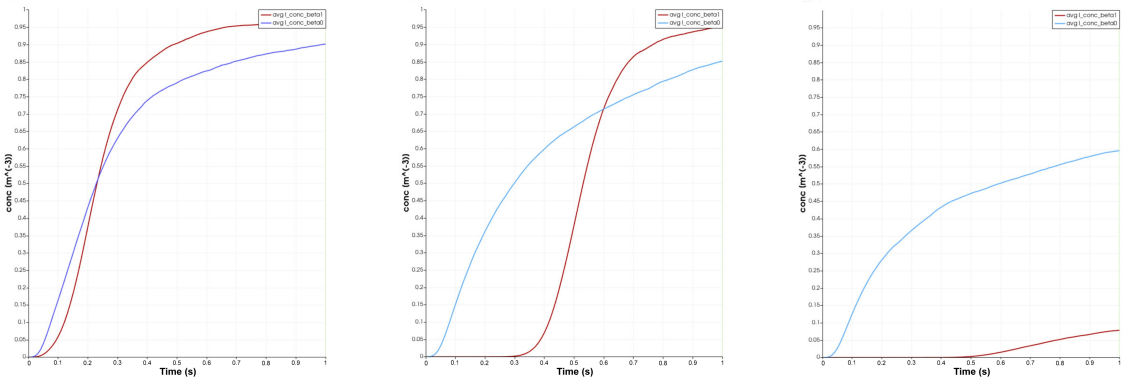


Figure 3.7: Average matrix concentration over time, t = 1 s.

Table 3.1: Computational complexity of the benchmark problem, flow equation, Newton's method, $\beta = 1$

Type	**Mesh**	**Computational time [s]**
Non-linear problem	$32k$	793
Linear problem	$32k$	1.02
Non-linear problem	$120k$	2803
Linear problem	$120k$	2.03

Table 3.2: Parameters used to calculate the benchmark problem.

Parameter	**Value**	**Units**
Matrix hydraulic conductivity $\mathbb{K}_3\|_{\Omega_{3,0}}$	$\mathbb{I}$	m/s
Matrix hydraulic conductivity $\mathbb{K}_3\|_{\Omega_{3,1}}$	$1 \times 10^{-1}\mathbb{I}$	m/s
Fracture hydraulic conductivity $\mathbb{K}_2$	$1 \times 10^{4}\mathbb{I}$	m/s
Intersection hydraulic conductivity $\mathbb{K}_1$	$1 \times 10^{4}\mathbb{I}$	m/s
Fracture cross-sectional length δ_2	1×10^{-4}	m
Intersection cross-sectional area δ_1	1×10^{-8}	m
Matrix porosity n_3	1×10^{-1}	-
Fracture porosity n_2	9×10^{-1}	-
Intersection porosity n_1	9×10^{-1}	-
Total simulation time	2.5×10^{-1}	s
Time step Δt	2.5×10^{-3}	s
Non-linear solver tolerance	10^{-6}	-
Maximum number of iterations	100	-
Linear solver residual tolerance	10^{-15}	-
Linear solver absolute tolerance	10^{-15}	-

Conclusion

In this Master Thesis, we solved the non-linear Darcy-Forchheimer equation in domains containing fractures. In the second chapter, we derived the weak formulation, the mixed hybrid formulation and the finite-dimensional approximation. Three linearization methods have been proposed; Picard iterations, Newton's method and L-scheme method. The Picard method is computationally very cheap and it is easy to implement, however it has proven to converge very slowly or even fail. For that reason, we proposed Newton's method, which has a quadratic convergence, however is hard to implement. To address these problems, last method was proposed: L-scheme method. This method is not that well known yet, but it has interesting properties, such as unconditional convergence.

The most challenging part was the implementation into already existing software *Flow123d*. We implemented over 700 lines of code, extending the software and adding the Darcy-Forchheimer equation, as well as all of the proposed linearization methods. The process of implementing and debugging new additions into already existing software is always difficult, especially without any previous experience with its structure.

The methods have been tested and compared on two simple analytical problems of the Darcy-Forchheimer equation, and also on a slightly more advanced problem of two intersecting fractures. The main measurement for the results of the three methods was the number of iterations they needed to converge, as the computational complexity of one iteration is almost identical for all of them. From these tests, the L-scheme and Newton's method came out as the better options, with L-scheme performing slightly better at the simplest problems. However, the L-scheme method highly depends on the chosen value of its parameter L, it performs slightly worse on finer meshes and might have problems when the velocity field has singularities.

Finally, in the last chapter, we compare our results to benchmark results. The benchmark case consisted of a cube and nine perpendicular fractures, making it more accurate to real world problems than prior experiments, however it still is

not completely realistic. We used the same geometry and parameters as in the benchmark case, and computed two variants of the problem. We started with a linear problem (using only Darcy equation) and because the fractures are highly conductive compared to the domain, we also computed the non-linear problem (using Darcy-Forchheimer equation). For the lineatization method, we chose Newton's method. We compared these two problems together as well as the results from the benchmark. We confirmed that the Forchheimer term has a great impact on the flow; it slows it down sufficiently.

The relevance of the selection of the parameter L and its effect on the L-scheme method's convergence may be the subject of subsequent research. Following a thorough examination of the optimal L parameter, future research could compare all three linearization methods on a benchmark problem.

Bibliography

[1] KANTZAS, Apostolos, Jonathan BRYAN a Saeed TAHERI. *Fundamentals of Fluid Flow in Porous Media* [online]. [cit. 2021-01-24]. Available from: https://perminc.com/resources/fundamentals-of-fluid-flow-in-porous-media/

[2] Darcy, H. (1856). *Les fontaines publiques de la ville de Dijon.* Paris: Dalmont.

[3] SMITH, James, Sevket DURUCAN, Anna KORRE a Ji-Quan SHI. *Carbon dioxide storage risk assessment: Analysis of caprock fracture network connectivity.* International Journal of Greenhouse Gas Control. 2011, 5(2), 226-240. ISSN 1750-5836. Dostupné z: doi:https://doi.org/10.1016/j.ijggc.2010.10.002

[4] TESTER, Jefferson W., Brian J. ANDERSON, Anthony S. BATCHELOR, et al. *The Future of Geothermal Energy: Impact of Enhanced Geothermal Systems (EGS) on the United States in the 21st Century.* MIT, 2006.

[5] TSANG, Chin-Fu, Ivars NERETNIEKS a Yvonne TSANG. *Hydraulic Issues associated with nuclear waste repositories.* Water Resources Research. 2015, (51), 6923-6972.

[6] FUMAGALLI, Alessio, Eirik KIELEGAVLEN a Stefano SCIALO. *Conforming, non-conforming and non-matching discretization couplings in discrete fracture network simulations.* Journal of Computational Physics. 2018, (376), 694-712. ISSN 0021-9991. Dostupné z: doi:https://doi.org/10.1016/j.jcp.2018.09.048.

[7] MUSKAT, Morris. *The Flow of Homogeneous Fluids Through Porous Media.* New York: McGraw-Hill, 1937. ISBN 978-0934634168.

[8] H.-D. CHEN, A. *Poroelasticity.* Springer, 2016. ISBN 978-3-319-25200-1.

[9] J. Maryška, M. Rozložník, M. Tůma. *Mixed-hybrid finite element approximation of the potential fluid flow problem.* Journal of Computational and Applied Mathematics 63(1–3), pp. 383-392, 1995.

[10] BREZZI, Franco a Michel FORTIN. *Mixed and Hybrid Finite Element Methods.* Springer-Verlag, 1991. ISBN 9783540975823.

[11] N. Frih, J. E. Roberts, A. Saada. *Modeling fractures as interfaces: a model for Forchheimer fractures, Comput Geosci* 12:91–104, 2008.

[12] ZIMMERMAN, Robert W., Azzan AL-YAARUBI, Chris C. PAIN a Carlos A. GRATTONI. *Non-linear regimes of fluid flow in rock fractures.* International Journal of Rock Mechanics and Mining Sciences. May: Elsevier, 2004, (41), 163-169. Dostupné z: doi:https://doi.org/10.1016/j.ijrmms.2004.03.036

[13] PANICONI, Claudio a Mario PUTTI. *A comparison of Picard and Newton iteration in the numerical solution of multidimensional variably saturated flow problems.* Water Resources Research: Subsurface Hydrology. 1994, 12(30), 3357-3374. Dostupné z: doi:https://doi.org/10.1029/94WR02046

[14] MITRA, Koondanibha a Iuliu Sorin POP. *A modified L-scheme to solve nonlinear diffusion problems.* Computers Mathematics with Applications. 6(77), 1722-1738. ISSN 0898-1221. Dostupné z: doi:https://doi.org/10.1016/j.camwa.2018.09.042

[15] User guide and input reference. *Flow123d [online].* Liberec: Technical University of Liberec, Faculty of mechatronics, informatics and interdisciplinary studies, 2018 [cit. 2020-08-28].
Dostupné z: ¿œßhttps://flow123d.github.io/

[16] *GitHub [online].* [cit. 2020-08-28]. Dostupné z: ¿œßhttps://github.com/Sabina-Bednarova/flow123d/releases/tag/DP-Bednarova

[17] KELLEY, C. T. *Iterative Methods for Linear and Nonlinear Equations.* North Carolina State University, 1995. ISBN 978-0-89871-352-7.

[18] BERRE, Inga, Wietese M. BOON, Bernd FLEMISCH, et al. *Verification benchmarks for single-phase flow in three-dimensional fractured-porous media.* Advances in Water Recourses. 2021, 147. Dostupné z: doi:10.1016/j.advwatres.2020.103759

[19] *Data accompanying "Verification benchmarks for single-phase flow in three-dimensional fractured porous media."* [online]. [cit. 2021-5-7]. Dostupné z: ¿œßhttps://git.iws.uni-stuttgart.de/benchmarks/fracture-flow-3d.git