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

Extended Finite Element Methods for Approximation of Singularities

Doctoral Thesis

Study programme: P3901 – Applied Sciences in Engineering
Study branch: 3901V055 – Applied Sciences in Engineering

Author: **Ing. Pavel Exner**
Supervisor: doc. Mgr. Jan Březina, Ph.D.

Liberec

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Rozšířené metody konečných prvků pro aproximaci singularit

Disertační práce

Studijní program: P3901 – Aplikované vědy v inženýrství
Studijní obor: 3901V055 – Aplikované vědy v inženýrství

Autor práce: **Ing. Pavel Exner**
Vedoucí práce: doc. Mgr. Jan Březina, Ph.D.

Liberec

Declaration

I hereby certify that I have been informed that Act 121/2000, the Copyright Act of the Czech Republic, namely Section 60, Schoolwork, applies to my dissertation in full scope. I acknowledge that the Technical University of Liberec (TUL) does not infringe my copyrights by using my dissertation for TUL's internal purposes.

I am aware of my obligation to inform TUL on having used or licensed to use my dissertation in which event TUL may require compensation of costs incurred in creating the work at up to their actual amount.

I have written my dissertation myself using literature listed therein and consulting it with my supervisor and my tutor.

I hereby also declare that the hard copy of my dissertation is identical with its electronic form as saved at the IS STAG portal.

Date: 26.2.2019

Signature: Exner

Abstrakt

Tato doktorská práce je zaměřena na řešení problému proudění podzemní vody v porézním prostředí, které je ovlivněno přítomností vrtů či studní. Model proudění je sestaven na základě konceptu redukce dimenzí, který je hojně využíván při modelování rozpukaného porézního prostředí, především granitů. Vrty jsou modelovány jako 1d objekty, které protínají blok horniny. Propojení těchto domén v redukovaném modelu způsobuje singularity v řešení v okolí vrtů. Vrty i porézní médium jsou síťovány nezávisle na sobě, což vede k výpočetním sítím kombinujícím elementy různých dimenzí.

Jádrem doktorské práce je pak vývoj specializované metody konečných prvků pro výše popsaný model. Pro umožnění propojení sítí různých dimenzí a pro zpřesnění aproximace singularit v okolí vrtů je použita rozšířená metoda konečných prvků (XFEM), v rámci níž jsou navrženy nové typy obohacení konečně-prvkové aproximace. Metoda XFEM je nejprve aplikována v modelu pro tlak, dále je navrženo obohacení pro rychlost a metoda je použita ve smíšeném modelu, jehož řešením jsou rychlost i tlak.

Doktorská práce se dále detailně věnuje numerickým aspektům v metodě XFEM, a to především adaptivním kvadraturám, volbě velikosti obohacené oblasti nebo podmíněnosti výsledného lineárního systému. Vlastnosti navržené XFEM metody a optimální konvergence jsou ověřeny na sérii numerických experimentů. Praktickým výstupem doktorské práce je implementace metody XFEM jako součásti open-source softwaru Flow123d.

Klíčová slova: Rozšířená metoda konečných prvků (XFEM), singularita, sítě kombinovaných dimenzí, Darcyho proudění, rozpukané porézní prostředí

Abstract

In this doctoral thesis, a model of groundwater flow in porous media influenced by wells (boreholes, channels) is developed. The model is motivated by the reduced dimension approach which is being often used in fractured porous media problems, especially in granite rocks. The wells are modeled as lower dimensional 1d objects and they intersect the surrounding bulk rock domains. The coupling between the wells and the rock then causes a singular behavior of the solution in the higher dimensional domains in the vicinity of the cross-sections. The domains are discretized separately resulting in an incompatible mesh of combined dimensions.

The core contribution of this work is in the development of a specialized finite element method for such model. Different Extended finite element methods (XFEM) are studied and new enrichments are suggested to better approximate the singularities and to enable the coupling of the wells with the higher dimensional domains. At first, the XFEM is applied in a pressure model, later an enrichment for velocity is suggested and the XFEM is used in a mixed model, solving both velocity and pressure.

Different numerical aspects of the XFEM is studied in details, including an adaptive quadrature strategy, a proper choice of the enrichment zone or a conditioning of the resulting linear system. The properties of the suggested XFEM are validated on a set of numerical tests and the optimal convergence rate is demonstrated. The XFEM is implemented as a part of the open-source software Flow123d.

Keywords: Extended Finite Element Method (XFEM), Singularity, Meshes of combined dimensions, Darcy flow, Fractured porous media

Dedicated to the memory of my late father, Zdeněk Exner,
whom I still turn to for a guidance through my life,
even though we cannot share the days in this world anymore.

Acknowledgements

This work would not have been created without the great support of my colleagues, family and friends and therefore I would like to express the deepest appreciation to all of them.

Especially, I would like to thank my supervisor, doc. Mgr. Jan Březina, Ph.D., for his leadership and all the support he provided me during my doctorate and also before. His advice and his optimism helped me several times to get past a problem when the solution was too blurred for my eyes.

I also greatly appreciate the support from my colleague, Mgr. Jan Stebel, Ph.D., who never refused to answer any of my questions however silly or complicated they were.

I am grateful to Prof. Barbara Wohlmuth for inviting me to the Technical University of Munich, for her hospitality during my research stay there and for the great opportunity to expand my horizons with the team at M2.

I owe a huge debt to my family and friends for spending all the extra hours at work instead of being with them. I would like to express my sincere gratitude to my mother, Hana Exnerová, in particular, for always supporting me and believing in me.

Finally, I thank with love to my soul mate, Markéta Tůmová. She has been my best friend and great companion, supported, encouraged, entertained and helped me through this exciting life period.

Contents

List of Figures

List of Tables

List of Graphs

List of Algorithms

List of Abbreviations and Symbols

Acronyms

DFN	Discrete Fracture Network
FVM	Finite Volume Method
FDM	Finite Difference Method
FEM	Finite Element Method
MHFEM	Mixed Hybrid Finite Element Method
PDE	Partial Differential Equation
PUM	Partition of Unity Method
XFEM	Extended Finite Element Method
GFEM	Generalized Finite Element Method
SGFEM	Stable Generalized Finite Element Method
CG	Conjugate Gradients
FLOP	Floating Point Operation
AABB	Axes Aligned Bounding Boxes
BIH	Boundary Interval Hiearchy
IC,IP	Intersection Corner, Intersection Polygon
MLMC	Multilevel Monte Carlo
TUL	Technical University of Liberec

Spaces

$\mathbb{R}$	space of real numbers
$\mathbb{N}$	space of unsigned integers
$\mathbb{P}^k$	space of polynomials of order k
$\mathbb{RT}^k$	space of Raviart-Thomas shape functions of order k
C^∞	space of smooth functions
L_2	Lebesque space
$H^k, H(\mathrm{div})$	Hilbert spaces
V	test space of the primary variable
Q	test space of the secondary variable
V', Q'	dual spaces
Λ	test space of Lagrange multipliers
V_h, Q_h, Λ_h	discrete test spaces

d	dimension: 1d, 2d, 3d
Ω_d	domain of dimension d
Γ_{dN}	part of boundary of Ω_d with Neumann boundary condition
Γ_{dD}	part of boundary of Ω_d with Dirichlet boundary condition
$\mathcal{T}_d$	mesh consisting of elements of dimension d
$\mathcal{F}_d$	faces of elements in $\mathcal{T}_d$
h	mesh parameter
T_d^i, F_d^i	element and face of $\mathcal{T}_d$
$\boldsymbol{x}_i$	node
$\mathcal{I}_{dE}, \mathcal{I}_{dF}, \mathcal{I}_{dN}$	indices of elements, faces and nodes in dimension d
$\mathcal{J}_{dE}, \mathcal{J}_{dN}$	indices of enriched elements and nodes in dimension d
Ω_C^w	domain (cylinder) of a well w
Ω_1^w	reduced domain (line) of a well w
Γ_w^m	interface boundary between well w and aquifer m
$\mathcal{W}$	index set of wells
$\mathcal{M}$	index set of aquifers
p_d	pressure in d-dimensional domain
$\boldsymbol{u}_d$	velocity in d-dimensional domain
λ_d	Lagrange multipliers in d-dimensional domain
$\boldsymbol{n}$	(outward) normal unit vector
δ_d	cross-section of d-dimensional domain
f_d	source term in d-dimensional domain
g_{dD}	Dirichlet boundary condition in d-dimensional domain
g_{dN}	Neumann boundary condition in d-dimensional domain
σ_w, σ_w^m	permeability coefficient between well and aquifer
r_w, ρ_w, R_w	distance function, radius and enrichment radius of a well w
Z_w	enrichment zone of a well w
$s_w, \boldsymbol{s}_w$	scalar and vector global enrichment function
$a(\cdot,\cdot)$	bilinear form
$l(\cdot)$	linear form
$\lVert\cdot\rVert_V$	norm associated to a space V
$\langle\cdot\rangle$	average operator
$\{\cdot\}$	fluctuation operator
π_T, π_T^{RT}, π_h	interpolation operators
$B^T, \mathbf{B}^T$	transposed operator, matrix
π_p	Plücker coordinates
$\pi_p \odot \pi_q$	permuted inner product
τ_K	topological position in element K
S_d	simplex of dimension d
ϵ	geometrical tolerance

1 Introduction

Mathematical modeling plays a very important role in science and also our daily lives throughout many different fields of knowledge. Using modern finite element methods (FEM), we are able to simulate, investigate and predict various phenomena of both nature and industrial character. Recent advances in the field of FEM enable us to solve models of various scales, dimensions, to overcome numerical problems and also to couple interfering features together.

A large set of problems with finite element models, that people nowadays deal with, is connected with insufficient accuracy in cases where the model includes large and very small scale phenomena at once. In our case we are interested in modeling of underground processes. We can imagine a simulation of groundwater flow in a large domain (hundreds of meters or even kilometers) which can be significantly influenced by thin fractures in the porous media or artificial wells and boreholes (several centimeters in diameter). These disturbances bring discontinuities and singularities into the model which are hard to capture at the geometric level and even harder to approximate with the standard polynomial finite elements at the discretization level. There are several ways one can follow to increase the accuracy of the standard FEM in such models.

Adaptive meshes can be used in such cases, but it can cost a lot of computational power to build a very fine mesh, and then solve the problem with increasing number of degrees of freedom. It requires very robust meshing algorithms when complex geometries are in question. There can be other constraints on the mesh generation in specific applications which the meshing tool must obey. To work optimally, the FEM demands mesh elements of a particular quality – there are different criteria for edges, angles between the edges or radii of inscribed and circumscribed balls. In some approaches, the presence of hanging nodes might require a special treatment or might be entirely unwanted. In the reduced dimension concept, which is described in details later, the so called compatible meshes might be required. Some methods for modeling time dependent problems, such as an opening of fractures in mechanics or interfaces in two-phase flows, require remeshing at each time step which further amplifies the demands on the meshing tools. All these aspects make the generation of computational meshes hard or even close to impossible.

Alternatively a reduced dimension concept is often used and models combining different dimensions are being developed. In this approach the geometry is decomposed in objects of different dimensions (e.g. 2d fractures, 1d wells, 0d point sources) and the meshes of the objects (computational domains) are created independently. Later the modeled processes must be coupled between the domains of different dimensions. The coupling concepts are mostly available at the continuous level but their implementation at the discrete level is non-trivial and problems often appear there. Two types of meshes are used in these models: compatible and incompatible. A compatible mesh is the one, where the intersections of the computational domains are aligned to nodes and sides of elements. On such a mesh, the standard FEM with proper coupling terms can be applied relatively easily, however the construction of the compatible mesh might be problematic. On the other hand, incompatible meshes with arbitrary intersections are easier to construct, but bring a whole new set of problems in the coupling.

Finally, there are the so called Extended finite element methods (XFEM); or Partition of unity methods (PUM) or Generalized finite element methods (GFEM). These names are often used interchangeably if one talks generally about these types of methods. The XFEM enables us to take advantage of an a priori knowledge of the model solution character such as discontinuity or singularity of searched quantities. The key aspect of the XFEM is that it allows to locally incorporate non-polynomial functions, like a jump or a singular function, into the finite element solution in places, where these features are expected to appear. This way the standard finite element approximation space is extended (enriched) and it is able to approximate the small scale phenomena more accurately. See an example of XFEM usage in 1d-2d coupling with point intersections in Figure 1.1.

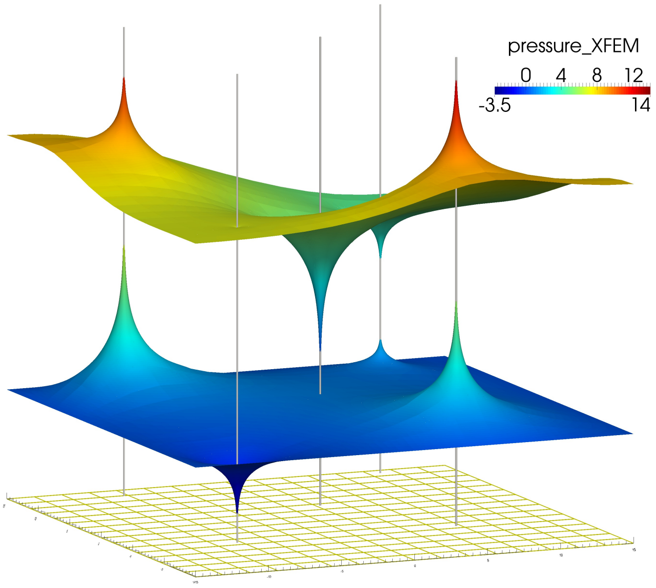


Figure 1.1: An XFEM example in 1d-2d coupling with singularities at intersection points. Distribution of pressure in 2 aquifers (horizontal planes) intersected by 5 wells (vertical lines) is displayed. The model is computed on a coarse mesh without refinement (visible at the bottom).

This thesis is aimed at further development of the XFEM and its usage in reduced dimensional models. We intend to create a model upon an incompatible mesh, where the elements of different dimensions intersect arbitrarily, and then use the XFEM to glue the modeled processes in different dimensions back together. Apparently this would be a very ambitious goal, therefore we narrow our work to non-planar 1d-2d and 1d-3d coupling and resolving arisen singularities.

Our research scope is focused at modeling of the underground processes in porous media and in this work we address ourselves to a groundwater flow model in particular. We look for a model that fits in the reduced dimension concept in the software Flow123d, which is being developed at the Technical University of Liberec (TUL), and we put our effort into implementing the model with XFEM as an additional feature of this software.

1.1 Flow123d

The practical outcome of this thesis is a new model and its solver implemented as a part of the software Flow123d. Flow123d is a simulator of underground water flow, solute and heat transport in fractured porous media. The software is developed as an open source code under the versioning system Git where the eponymous project can be found, or one can reach it via the web page

`http://flow123d.github.io`

It has been developed at the TUL approximately since 2007. The latest stable version 2.2.1 was released in 2018 [1]. The main feature of the software is the ability to compute on complex compatible meshes of combined dimensions, where the continuum models and discrete fracture network models can be coupled. Only the coupling of co-dimension 1 is considered, i.e. coplanar 1d-2d and 2d-3d cases. This thesis extends the groundwater flow model to non-planar 1d-2d and 1d-3d coupling.

Figure 1.2: Logo of Flow123d.

Current version includes mixed-hybrid solver for steady and unsteady Darcy flow, non-linear solver for unsaturated flow (Richards equation), finite volume method and discontinuous Galerkin method for solute transport of several substances and heat transfer model. Using the operator splitting technique, models for various local processes are supported with the solute transport simulation including dual porosity, sorption, decays and first order reactions.

The experimental branch of Flow123d based on version 3.0.x, which is about to be officially released, includes a module for computation of intersections of incompatible meshes, a rock mechanics model and also the XFEM model based on this thesis.

1.2 Aims of Thesis

We are motivated by several applications where our approach can provide a new solution of problems with singular behavior and where it extends possibilities of currently available models and methods. Apart from the already mentioned groundwater flow, in which we focus in this work, there are various other applications where our model might be employed or to which it might be extended. We list a few that are closely related: enhanced geothermal systems and extraction of geothermal energy, oil and gas extraction, CO_2 sequestration, safety assessment of an underground nuclear waste repository, transport of substances in human body or transport in fractured porous media in general.

Considering what we have preceded in the introduction, we see an open space for our research in further development of reduced dimensional FEM models, especially of co-dimension 2, in combination with the XFEM and incompatible meshes, where the XFEM provides a way for coupling the domains of different dimensions together and to significantly improve the finite element approximation accuracy.

The research aims of this work can be summarized in two major points

- Propose suitable XFEM enrichments for singular pressure and velocity in Darcy flow model. If possible, emphasize a good approximation of the velocity field.
- Suggest new data structures and algorithms for the realization phase in the software Flow123d.

The accuracy of the velocity field, in particular the fluxes across the elements edges, is in the main focus in most applications, e.g. when the flow model is coupled with a transport equation.

We briefly sketch the steps that we take to successfully reach these goals. We inspire ourselves by the work of Gracie and Craig [2, 3], and we start from a similar 1d-2d model with XFEM resolving point singularities in pressure. We compare different XFEMs [4, 5, 6, 7], investigate the properties of the methods and we suggest some improvements.

Then we transform the model to the mixed-hybrid form based on [8, 9]. We suggest an enrichment for velocity and a discretization with mixed finite elements. We implement this XFEM enrichment in Flow123d and we run several numerical experiments. We further extend these results into 1d-3d coupled model.

To be able to apply the XFEM on the incompatible meshes, it is necessary to determine the intersections between the meshes of different dimensions. We do not restraint us only to 1d-2d and 1d-3d cases, but we suggest a robust algorithm for finding intersections of simplicial meshes in arbitrary 3d space. Considering a broader scope, this algorithm can be used also for fractures and cracks in other applications.

Regarding the implementation and the code design, several non-standard techniques are required in comparison to standard FEM codes. Efficient data structure must be provided for the enriched finite elements. Handling of the additional degrees of freedom of the enrichment must be considered. An accurate integration on the intersected (and enriched) elements is needed to assembly the integrals correctly. Solving the final linear algebraic system requires an efficient numerical method. A reasonable output post-processing must be developed, so that the XFEM solution (piecewise non-polynomial solution) can be visualized properly. Each of these sub-problems is not trivial and requires a new approach or at least extending the existing one.

1.3 Document Structure

The text is divided into five major parts. In Chapter 2, the reduced dimension concept is described in details and the meshes of combined dimensions are defined. The models of groundwater flow, coupling non-planar 1d-2d and 1d-3d domains, are formulated and put in the context of the software Flow123d and other works.

The background research on various XFEMs is provided in Chapter 3. It is then followed by Chapter 4 where a well-aquifer pressure model is studied and in which the singular enrichments are applied by the means of the XFEM. Various aspects and properties of the model and the method are studied, different enrichment strategies are compared.

Next, a mixed-hybrid model for pressure and velocity is formulated in Chapter 5. A new enrichment for velocity is proposed in the mixed form. The coupled models both for non-planar 1d-2d and 1d-3d are defined and followed by several numerical tests demonstrating the properties of the new enrichment in XFEM.

Chapter 6 is dedicated to the intersection algorithms. The concept of Plücker coordinates is introduced and algorithms for various intersection cases of simplicial elements is described. Numerical experiments are provided.

The thesis is closed with the conclusion in the last Chapter 7 which is followed by the list of author's publications and the bibliography.

2 Reduced Dimensional Models

The goal of this chapter is to introduce several concepts of combining models of different spatial dimension and to put this work into the context of the so called *reduced dimensional models*. At the end we define a well aquifer model, combining 1d-2d and 1d-3d dimensions, which is to be solved in this work. Also the relationship to the long-term research at the Technical University in Liberec is described.

The realistic models of underground processes, namely the groundwater flow and transport, geomechanics and thermal processes, have to deal with a complex of geological formations containing various small scale features such as fractures and wells. Although the scale of these features can differ from the size of the computational domain by several orders, they influence significantly the global behavior of the system. In the past years, two different approaches for incorporating these features into numerical models have been developed.

The equivalent continuum concept includes the fractures and wells in the model by changing the global properties of the modeled volume (increased/decreased hydraulic conductivity, porosity or other rock properties). This approach smears local effects of the disruptions but the model can still provide a valid approximation for an initial global view of the system. For example, instead of having a rock with a complex network of conductive fractures, we use an equivalent continuum with increased conductivity, and the domain can be simply discretized without taking the fractures into account. The concept is also suitable due to the lack of detailed data which people suffer from in most applications. The microscopic effects are neglected by applying the representative elementary volume (homogenization per elements of mesh in FVM, FDM or FEM), the macroscopic effects (larger fractures) can be modeled by equivalent continuum. Since the underlying computational mesh does not need to represent the small scale objects, the equivalent continuum models can be computationally cheap, but they cannot capture the local effects accurately.

The other approach considers the disruptions in a discrete sense, keeping its sizes and properties explicitly in the model. This brings more demanding work at the geometric level, where the disruptions have to be represented so that a suitable computational mesh can be built. A significant simplification in this concept is to reduce the dimension of the local phenomena, if possible, and to represent these as lower dimensional objects. This means at the geometrical level to neglect the thickness of a thin fracture and consider a plane, or neglect the cross-section of a narrow channel and consider a line instead. The quantities are then averaged over the complementary dimensions. This approach is referred as *reducing dimensions* or *reduced dimensional model*. See Figure 2.1 for illustration.

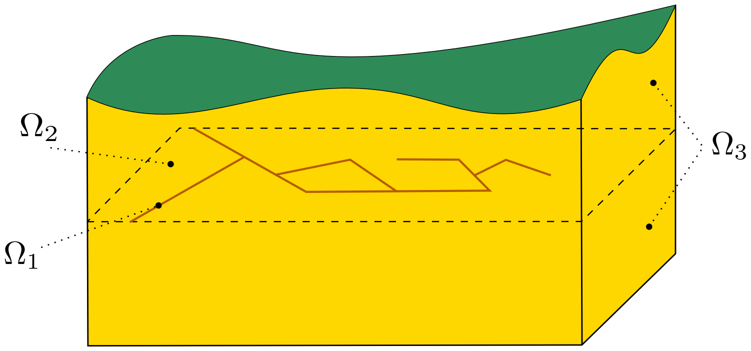


Figure 2.1: Scheme of a problem with domains of multiple dimensions, (taken from the Flow123d documentation [10]).

Within the scope of geology or hydrogeology, the reduced dimension concept is used in the Discrete Fracture Network (DFN) models. There is a world-wide community around the DFN problematic, see e.g. [11, 12] for an overview. The people are dealing with a wide range of topics on DFN, from the definition and measurement of the geological parameters, stochastic generation of DFN and its discretization for FVM/FEM methods, to solving multi-physic problems on such geometries. We may differentiate between two types of fracture network models. One is strictly considering only the fracture network itself, without considering the bulk domain between fractures. Such model can predict for example the so called *preferential paths* in concentration transport problems, assuming the transport through the bulk domain is negligible. The other model takes the processes running in the bulk domain into account, so it needs to solve the communication between the fracture and bulk domains in addition, which puts an extra effort on discretization, coupling conditions and solution of the underlying linear system. Some upscaling techniques regarding DFN are also being developed, where the idea is to have a very complex DFN at the most refined level and only the most influential parts of the DFN (or none) at the coarsest level.

Having the geometry of a reduced dimensional model prepared, the space discretization then follows, leading to the meshes of combined dimensions, i.e. composed of elements of different dimension. This approach, called mixed-dimensional analysis in the mechanics [13], is also studied in the groundwater context, see e.g. [14, 15, 16, 17, 18] and already adopted by some groundwater simulation software, e.g FeFlow [19] and Flow123d [1].

The works mentioned before deal with modeling fractures, however they are not concerned neither about 1d-3d coupling nor point intersections in non-planar 1d-2d coupling. These topics are also studied extensively by different groups, see especially [20, 21] and [22, 23]. Their applications are mainly addressed to the medical problematic and they analyze the transport of medicaments through the vascular system and to the surrounding tissue. The veins can be viewed as 1d channels and the tissue is modeled as a porous media, and thus the flow is governed by the same type of equations as in the groundwater simulations.

In this work we build and expanse a reduced dimensional model on top of the existing one being developed at TUL. It combines both the equivalent continuum and the DFN approach including the communication fractures with surrounding bulk domains. Since the numerical solution is based on FEM, we are interested in how the fractures and bulk are geometrically discretized. Therefore we describe the meshes of combined dimensions below.

2.1 Mesh of Combined Dimensions

The motivation to model the small scale disruptions in the bulk domain as lower dimensional objects has been stated in the introduction of the chapter. Now we fix the idea of a mesh of combined dimensions and we give a more precise description.

An open set $\Omega_3 \subset \mathbb{R}^3$ represents a continuous approximation of a porous medium, i.e. the bulk domain. In the same manner, a set of 2d manifolds $\Omega_2 \subset \mathbb{R}^3$ is considered, representing the 2d fractures and a set of 1d manifolds $\Omega_1 \subset \mathbb{R}^3$ representing the 1d channels in an arbitrary 3d space. See the Figure 2.1.

The domains Ω_2 and Ω_1 are polytopic (i.e. polygonal and piecewise linear, respectively). Following the documentation of Flow123d [10] and its notation, for every dimension $d = 1, 2, 3$, a triangulation $\mathcal{T}_d$ of the open set Ω_d is introduced that consists of

$$\begin{aligned} \text{elements } T_d^i, &\qquad i \in \mathcal{I}_{d,E} = \{1, \dots, N_{d,E}\}, \\ \text{faces } F_d^i, &\qquad i \in \mathcal{I}_{d,F} = \{1, \dots, N_{d,F}\}, \\ \text{nodes } \boldsymbol{x}_i, &\qquad i \in \mathcal{I}_{d,N} = \{1, \dots, N_{d,N}\}. \end{aligned} \tag{2.1}$$

If not specified, the simplicial meshes are used, so the elements are lines, triangles and tetrahedra. By faces F_d^i we understand $(d-1)$-dimensional boundary objects of elements T_d^i, i.e. end points F_1^i of T_1^i, triangle sides F_2^i of T_2^i, tetrahedron faces F_3^i of T_3^i. We note that in Chapter 4, the quadrilateral elements are considered in contrast to the rest of the work due to the use of a specific FEM library code.

The elements in *a compatible mesh* must satisfy the following compatibility conditions

$$T_{d-1}^i \cap \mathcal{T}_d \subset \mathcal{F}_d, \qquad \text{where } \mathcal{F}_d = \bigcup_{k \in \mathcal{I}_{d,E}} \partial T_d^k \tag{2.2}$$

and

$$T_{d-1}^i \cap \mathcal{F}_d = \begin{cases} T_{d-1}^i, \\ \emptyset, \end{cases} \tag{2.3}$$

for every $i \in \mathcal{I}_{d-1,E}$, and $d = 2, 3$. That is, the $(d-1)$-dimensional elements are either between d-dimensional elements and match their faces or they poke out of Ω_d. Under these conditions, a compatible 1d line in 3d space is formed only as an intersection of two 2d planes. Direct compatibility in case of co-dimension 2, i.e. for 1d and 3d coupling, is not considered.

A creation of a compatible mesh can be challenging on the algorithms, due to the compatibility conditions. The requirements on the element quality can be also limiting, since the intersection of domains can be very arbitrary. Therefore we look for models not requiring the compatible meshes.

An incompatible mesh, also referred as non-conforming or non-matching, does not have to satisfy the compatibility equations above. The creation of the mesh is much simpler when it can be done independently per domain. On the other hand, there are three main challenges at the next levels of the model solution. First, the intersections of the mesh elements must be computed efficiently and the intersection data must be represented in such a way that it can be used simply and cheaply in the FEM code. Second, an efficient numerical integration must be available on the intersecting polygons. Third, stable coupling conditions between dimensions must be provided. Part of this work, Chapter 6, is dedicated to a development of algorithms for computation of intersections of incompatible meshes.

2.2 Well-Aquifer Model

A well-aquifer model that we solve in our work is defined in this section. We follow the reduced dimension concept and introduce a simplified underground water flow model.

We consider wells (or boreholes, channels) to be straight narrow tubes (cylinders) Ω_C^w of radius ρ_w; we use the index $w \in \mathcal{W} = \{1 \dots W\}$ when referring to a particular well. Using the reduced dimensional concept, the wells are modeled as lower dimensional objects, i.e. 1d manifolds Ω_1^w which represent the central lines of cylinders Ω_C^w. The 1d manifold Ω_1^w can be parameterized using a mapping $\nu_w : [0,1] \to \Omega_1^w$:

$$\Omega_1^w = \{\boldsymbol{x} \in \mathbb{R}^3 : \boldsymbol{x} = \nu_w(t),\ t \in [0,1]\}. \tag{2.4}$$

Later in the text we suppose that functions in Ω_1^w depend on the relevant parameter t. We define a lateral surface of a cylinder Ω_C^w

$$\partial\Omega_C^w = \{\boldsymbol{x} \in \mathbb{R}^3 : \boldsymbol{x} = \nu_w(t) + \boldsymbol{r},\ \boldsymbol{r} = \rho_w \boldsymbol{n}_{\Omega_1^w}(t),\ t \in [0,1]\}, \tag{2.5}$$

where $\boldsymbol{n}_{\Omega_1^w}(t)$ is the unit normal vector to the line Ω_1^w at point t.

The domain of a well Ω_1^w has a boundary (obviously consisting of two end points) $\Gamma_1 = \Gamma_{1D} \cup \Gamma_{1N}$, where pressure (Dirichlet boundary condition) or flux (Neumann boundary condition) is prescribed. If possible, and apparent from context, we denote the domains of all wells by

$$\Omega_1 = \bigcup_{w \in \mathcal{W}} \Omega_1^w, \quad \Omega_C = \bigcup_{w \in \mathcal{W}} \Omega_C^w. \tag{2.6}$$

A steady groundwater flow governed by Darcy's law is considered in aquifers. The same is considered also in wells for simplicity: we can imagine the wells filled with different porous material, or in some tests we fix pressure in wells to a constant value. Without any coupling terms, we would write for every dimension d

$$\frac{1}{\delta_d}\boldsymbol{K}_d^{-1}\boldsymbol{u}_d + \nabla p_d = 0 \qquad \text{in } \Omega_d \tag{2.7}$$

$$\operatorname{div} \boldsymbol{u}_d = \delta_d f_d \qquad \text{in } \Omega_d \tag{2.8}$$

where $\delta_d \boldsymbol{u}_d$ $[ms^{-1}]$ is the unknown Darcian velocity, p_d $[m]$ is the unknown pressure. Parameter δ_d is the complement measure of the domain: thickness $[m]$ in 2d, cross-section $[m^2]$ in 1d and $\delta_3 = 1$ $[-]$ for consistency. Since the wells are considered as cylinders, the cross-section δ_1 is constant per well: $\delta_1(\boldsymbol{x}) = \delta_1^w = \pi\rho_w^2$ for all $\boldsymbol{x} \in \Omega_1^w$. The quantity $\boldsymbol{u}_d$ $[m^{d-4}s^{-1}]$ itself can be seen as flux density, i.e. flux through Ω_d with complementary dimension $\delta_d = 1$. The conductivity tensor $\boldsymbol{K}_d$ $[ms^{-1}]$ is generally 3×3 matrix, symmetric and positive definite, possibly representing the anisotropy of the media. In the source term, f_d $[s^{-1}]$ denotes the water source density.

We note that we use the same operators div and ∇ in all dimensions $d = 1, 2, 3$ for the sake of notation simplicity, although in 1d we understand these as follows

$$\operatorname{div} \boldsymbol{u}_1 = \operatorname{div}_t \boldsymbol{u}_1 = \frac{\partial}{\partial t}\boldsymbol{u}_1(t), \tag{2.9}$$

$$\nabla p_1 = \nabla_t p_1 = \frac{\partial}{\partial t} p_1(t), \tag{2.10}$$

with t being the 1d parameter in (2.4).

Two models are presented below, the first one supposes further simplifications and includes 1d-2d coupling between dimensions, the other one couples 1d and 3d domains together.

2.2.1 1d-2d Model

We consider a system of aquifers separated by aquitards (based on Gracie and Craig [2, 3]). The aquifers are underground horizontal layers of permeable rock containing water in its pores. These layers often have negligible flow in the vertical direction so they can be modeled as 2d planes by means of the reduced dimensional modeling. There can be several aquifers on top of each other, separated by layers of very low permeability which are called aquitards. In contrast to Gracie and Craig, we suppose the aquitards to be impermeable, so the aquifers can communicate between each other only through the wells. See the scheme of well-aquifer model in Figure 2.2. On the other hand, we add an artificial volume source term on aquifers. This allows us to better study the impact of the prescribed source on the solution which is better suited to the numerical experiments we are going to present.

The model is defined as a complex multi-aquifer system to follow our implementation and to see the differences we made in comparison to Gracie and Craig. However, before solving the complex model, we constrain ourselves to a single aquifer in a major part of the work.

Let Ω_2^m be the domain of an aquifer with index $m \in \mathcal{M} = \{1 \dots M\}$. We denote the aquifers' boundary $\partial\Omega_2^m = \Gamma_{ext}^m \cup \Gamma_{int}^m$, where we call $\Gamma_{ext}^m = \Gamma_{2D}^m \cup \Gamma_{2N}^m$ the exterior boundary, consisting of two parts where we prescribe Dirichlet and Neumann boundary condition, respectively. The interior boundary Γ_{int}^m is created by an union of all the cross-sections of the aquifer domain Ω_2^m and wells lateral surfaces

$$\Gamma_{int}^m = \bigcup_{w \in \mathcal{W}} \Gamma_w^m, \quad \Gamma_w^m = \Omega_2^m \cap \partial\Omega_C^w. \tag{2.11}$$

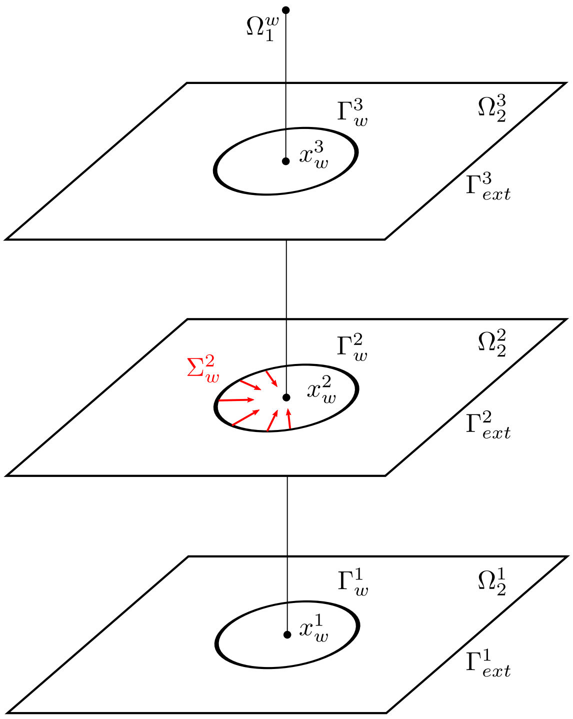


Figure 2.2: Well-aquifer model scheme.

Where possible, we again use the simplified notation

$$\Omega_2 = \bigcup_{m \in \mathcal{M}} \Omega_2^w, \quad \Gamma_\# = \bigcup_{m \in \mathcal{M}} \Gamma_\#^m \tag{2.12}$$

with # being *ext*, *int*, $2D$ or $2N$.

On Γ_w^m, we define the following decomposition of an arbitrary function $q \in C(\overline{\Omega}_2^m)$ on average and fluctuation parts, in the same manner as it can be found in [23],

$$q = \langle q \rangle_w^m + \{q\}_w^m \quad \text{on } \Gamma_w^m, \tag{2.13}$$

where

$$\langle q \rangle_w^m = \frac{1}{|\Gamma_w^m|} \int_{\Gamma_w^m} q \, \mathrm{d}s. \tag{2.14}$$

It holds for the fluctuation term from the definition (2.14)

$$\int_{\Gamma_w^m} \{q\}_w^m \, \mathrm{d}s = \int_{\Gamma_w^m} q - \langle q \rangle_w^m \, \mathrm{d}s = 0. \tag{2.15}$$

The average term is linear, since it holds

$$\langle q_1 + q_2 \rangle_w^m = \langle q_1 \rangle_w^m + \langle q_2 \rangle_w^m. \tag{2.16}$$

The point where the reduced domain Ω_1^w intersects with aquifer Ω_2^m is denoted $\boldsymbol{x}_w^m = \nu_w(t^m)$. In order to prescribe a communication term in Ω_1^w, we use a Dirac measure at the point t^m

$$\delta_t(t-t^m) = \begin{cases} 1 & t = t^m \\ 0 & t \neq t^m \end{cases}, \qquad \int_0^1 q(t)\delta_t(t-t^m)\,\mathrm{d}t = q(t^m), \quad q \in C(\overline{\Omega}_1^w). \tag{2.17}$$

Then we can write the entire problem:

Problem 2.2.1. Find $[\boldsymbol{u}_1,\, \boldsymbol{u}_2]$ and $[p_1,\, p_2]$ satisfying

$$\delta_d^{-1}\boldsymbol{K}_d^{-1}\boldsymbol{u}_d + \nabla p_d = 0 \qquad \text{in } \Omega_d,\ d = 1,2, \tag{2.18a}$$

$$\operatorname{div}\boldsymbol{u}_2 = \delta_2 f_2 \qquad \text{in } \Omega_2, \tag{2.18b}$$

$$\operatorname{div}\boldsymbol{u}_1 = \delta_1 f_1 + \sum_{m\in\mathcal{M}} |\Gamma_w^m|\,\Sigma_w^m \delta_t(t-t^m) \qquad \text{in } \Omega_1^w,\ \forall w \in \mathcal{W}, \tag{2.18c}$$

$$\langle -\delta_2 \boldsymbol{K}_2 \nabla p_2 \cdot \boldsymbol{n} \rangle_w^m = \Sigma_w^m \qquad \forall w \in \mathcal{W},\ \forall m \in \mathcal{M}, \tag{2.18d}$$

$$\{p_2\}_w^m = g_w^m \qquad \forall w \in \mathcal{W},\ \forall m \in \mathcal{M}, \tag{2.18e}$$

$$p_d = g_{dD} \qquad \text{on } \Gamma_{dD},\ d = 1,2, \tag{2.18f}$$

$$\delta_d \boldsymbol{K}_d \nabla p_d \cdot \boldsymbol{n} = g_{dN} \qquad \text{on } \Gamma_{dN},\ d = 1,2 \tag{2.18g}$$

where

$$\Sigma_w^m = \delta_2(\boldsymbol{x}_w^m)\sigma_w^m\big(\langle p_2\rangle_w^m - p_1(\boldsymbol{x}_w^m)\big).$$

One can see the dimensional coupling terms highlighted in the blue color. The term Σ_w^m is a constant representing the flux density from the aquifer Ω_2^m to the well Ω_1^w through their intersection Γ_w^m. The flux is proportional to the pressure difference with the permeability coefficient $\sigma_w^m > 0$ $[s^{-1}]$ between Ω_2^m and Ω_1^w. If the pressure difference is negative, the flux is in opposite direction. We can then read: the average outward flux from the aquifer over Γ_w^m in (2.18d) acts as a positive source term in the well in (2.18c).

The equation (2.18e) prescribes the fluctuation of pressure on Γ_w^m, which is not defined by the coupling terms. In reality, wells are very narrow in comparison to the scale of aquifers and so the changes in pressure on Γ_w^m are very small. Thus we consider $g_w^m \approx \rho_w$ and we neglect the effects of g_w^m as shown later.

The last two equations are the boundary conditions, the first one (2.18f) prescribes the boundary pressure g_{dD}, the later one (2.18g) sets the inward flux g_{dN}. The sign of the flux in (2.18g) is chosen such that it is consistent with the software Flow123d.

The problematic place of a reduced dimensional model, if done in the simplest way, would be that after the reduction of wells, the 1d-2d intersections are degenerated to points. Then the term in the flux coupling condition (2.18d) would rather appear as a point source on the right hand side of (2.18b) creating strong singularities in the solution. However, keeping the real scale of the intersections in the model by means of (2.18d), we cut off the singularity peaks at finite height and get rid of the singular solution part running to infinity. Still there is left the problem of capturing the intersections and adjacent steep gradients in the solution by the numerical discretization. This is the same motivation found in models in [2, 3, 23] and ours, although treated very differently at the numerical level in each of the works.

2.2.2 1d-3d Model

In this model the aquifer domain is not reduced to a plane but it is represented as a 3d domain. This would be a necessary consideration when the aquifer cannot be properly modeled in layers. Let us denote the domain Ω_3, we do not have the index m here. The boundary $\partial\Omega_3 = \Gamma_{ext} \cup \Gamma_{int}$, where the exterior boundary consists of two parts $\Gamma_{ext} = \Gamma_{3D} \cup \Gamma_{3N}$, analogically to the 1d-2d model.

The interior boundary Γ_{int} is created by an union of all the cross-sections of the domain Ω_3 and wells lateral surfaces

$$\Gamma_{int} = \bigcup_{w\in\mathcal{W}} \Gamma_w, \quad \Gamma_w = \Omega_3 \cap \partial\Omega_C^w. \tag{2.19}$$

On Γ_w we define the average decomposition $w = \langle g\rangle_w + \{g\}_w$ in the similar manner as in the 1d-2d case. However, the average is computed over a circle edge, perpendicular to Ω_1^w, with its center at a point $\nu_w(t) \in \Omega_1^w$

$$\langle g\rangle_w(t) = \frac{1}{2\pi\rho_w}\int_0^{2\pi} g(t + \rho_w \boldsymbol{n}_{\Omega_1^w}(t,\theta))\rho_w \,\mathrm{d}\theta. \tag{2.20}$$

with $\boldsymbol{n}_{\Omega_1^w}(t,\theta)$ being the unit normal vector of Ω_1^w at point $\nu_w(t)$ in the direction determined by the angle θ.

Then we can write the entire problem:

Problem 2.2.2. Find $[\boldsymbol{u}_1, \boldsymbol{u}_3]$ and $[p_1, p_3]$ satisfying

$$\begin{aligned}
\delta_d^{-1}\boldsymbol{K}_d^{-1}\boldsymbol{u}_d + \nabla p_d &= 0 && \text{in } \Omega_d,\ d = 1,3, && (2.21a)\\
\operatorname{div}\boldsymbol{u}_3 &= \delta_3 f_3 && \text{in } \Omega_3, && (2.21b)\\
\operatorname{div}\boldsymbol{u}_1 &= \delta_1 f_1 + \Sigma_w && \text{in } \Omega_1^w,\ \forall w \in \mathcal{W}, && (2.21c)\\
\langle -\delta_3\boldsymbol{K}_3\nabla p_3 \cdot \boldsymbol{n}\rangle_w &= \Sigma_w && \text{in } \Omega_1^w,\ \forall w \in \mathcal{W}, && (2.21d)\\
\{p_3\}_w &= g_w && \forall w \in \mathcal{W}, && (2.21e)\\
p_d &= g_{dD} && \text{on } \Gamma_{dD},\ d = 1,3, && (2.21f)\\
\delta_d\boldsymbol{K}_d\nabla p_d \cdot \boldsymbol{n} &= g_{dN} && \text{on } \Gamma_{dN},\ d = 1,3 && (2.21g)
\end{aligned}$$

where

$$\Sigma_w(\boldsymbol{x}) = \delta_3\sigma_w(\boldsymbol{x})\big(\langle p_3\rangle_w(\boldsymbol{x}) - p_1(\boldsymbol{x})\big) \quad \boldsymbol{x} \in \Omega_1^w.$$

One can see the dimensional coupling terms highlighted in the blue color. The term Σ_w, similarly to 1d-2d case, represents the flux from the aquifer Ω_3 to the well Ω_1^w. The flux is proportional to the pressure difference with the (variable) permeability coefficient $\sigma_w(\boldsymbol{x}) > 0$ $[s^{-1}]$ between Ω_3 and Ω_1^w. However, in contrast to the 1d-2d case, $\Sigma_w(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega_1^w$, varies through the well domain. Analogically we can then read: the average outward flux from the aquifer at point $\boldsymbol{x} \in \Omega_1^w$ in (2.21d) acts as a positive source term in the well in (2.21c). The last two equations are the boundary conditions, the first one (2.21f) prescribes the boundary pressure g_{dD}, the later one (2.21g) sets the inward flux g_{dN} in the same manner as in 1d-2d case.

3 Extended Finite Element Method

In this chapter we provide a background research on the XFEM, describing the fundamentals and giving an overview on the evolution of the XFEM. Different enrichment methods and various types of enrichments are discussed.

3.1 Basic Concept

The main feature of this method is the extension of a space of polynomial shape functions of the finite element method with a special function, that enables to better approximate some local effects. This is called *enrichment*. The special function (enrichment function) is often non-polynomial and describes discontinuity or singularity, where polynomials leak accuracy. We can meet these two terms: *extrinsic enrichment*, which adds the enrichment functions to a basis and *intrinsic enrichment*, which replaces some basis functions with the enrichment ones. We are interested in mesh-based methods with extrinsic enrichment and we shall not discuss any mesh-free alternatives.

An overview article by Fries and Belytschko [5] concludes the history and early development of the XFEM. The origins from the Partition of Unity Method (Babuška and Melenk e.g. in [24]) and the Generalized Finite Element Method (e.g. by Strouboulis in [25]) and the early XFEM (e.g. by Belytschko and Moës in [26]) are commented in details there.

The most frequent application of the XFEM is in mechanics, where people take advantage of the XFEM especially in modeling cracks growth without requiring remeshing the computational mesh in every time step. Discontinuous functions (Heaviside or signum function) are used to capture the jump in stress or strain at the crack, singular functions are applied in the vicinity of the crack tips. The exact function for the selected enrichment comes either directly from the nature of the phenomenon, or can be obtained as the solution of a local auxiliary problem. This function is most often referred as the *global enrichment function*.

The XFEM is mainly perceived as a method for local enrichment, which means that the enrichment is applied only in a small subdomain – several elements of the computational mesh close to the local phenomenon. The XFEM solution with a single enrichment is sought in the form

$$u(\boldsymbol{x}) = \sum_{\alpha \in \mathcal{I}} a_\alpha v_\alpha(\boldsymbol{x}) + \sum_{\alpha \in \mathcal{J}} b_\alpha \phi_\alpha(\boldsymbol{x}) \tag{3.1}$$

where a_α are the standard FE degrees of freedom and b_α are the degrees of freedom coming from the enrichment, v_α are the standard FE basis shape functions. We denote the index sets $\mathcal{I}$ and $\mathcal{J}$ that contain all indices of the standard and enriched degrees of freedom, respectively. The *local enrichment function* ϕ_α in (3.1) is typically defined as

$$\phi_\alpha(\boldsymbol{x}) = N_\alpha(\boldsymbol{x})L(\boldsymbol{x}), \quad \alpha \in \mathcal{J} \tag{3.2}$$

where L is the actual enrichment function and N_α is a function of the partition of unity $\sum_\alpha N_\alpha(\boldsymbol{x}) = 1$, creating the localized degrees of freedom for the enrichment. Typically, N_α are linear FE basis functions, having support points at the mesh nodes, $N_\alpha(\boldsymbol{x}_\alpha) = 1$, so one is often talking about enriching nodes and calling these *enriched nodes*. Different functions which make the partition of unity might be used, but linear functions are the most common. An example of such local enrichment is shown in Figure 3.1. One can see the enriched nodes, denoted by red dots, inside the chosen enrichment zone Z, defined by the green boundary line. All nodes of the dark elements are enriched, so N_α in (3.2) creates the whole partition of unity there. In contrast to that, the light elements are enriched only partially and the sum of N_α in the enrichment is smaller than one.

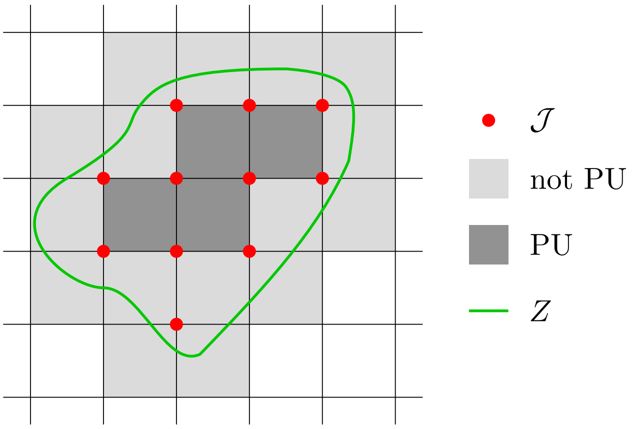


Figure 3.1: Local enrichment on a quadrilateral mesh.

The most straightforward choice of $L(\boldsymbol{x})$ is the global enrichment function itself but it is not optimal. It can suffer with a lack of convergence, a large approximation error at some elements or an ill-conditioning of the linear system. All these problems were intensively studied and lead to different XFEM methods: the so called Corrected XFEM [4] and the Stable Generalized Finite Element Method (SGFEM) [6, 7], which both became standard for XFEM in general. These are discussed in more details later on.

Considering more than one enrichment is straightforward. The approximation will then change into the following form

$$u(\boldsymbol{x}) = \sum_{\alpha \in \mathcal{I}} a_\alpha v_\alpha(\boldsymbol{x}) + \sum_{e \in \mathcal{E}} \sum_{\alpha \in \mathcal{J}^e} b_{e\alpha} \phi_{e\alpha}(\boldsymbol{x}), \tag{3.3}$$

with

$$\phi_{e\alpha}(\boldsymbol{x}) = N_\alpha(\boldsymbol{x}) L_e(\boldsymbol{x}), \quad e \in \mathcal{E},\ \alpha \in \mathcal{J}^e. \tag{3.4}$$

The enrichment e is enriching all the nodes in the index set $\mathcal{J}^e$ with degrees of freedom $b_{e\alpha}$ corresponding to the local enrichment functions $\phi_{e\alpha}$. Note that there can be multiple functions $\phi_{e\alpha}$ with support on the same element. If these functions are close to each other in some measure in the approximation space, then the respective degrees of freedom can become linearly dependent which results in ill-conditioning of the linear system (this issue is mainly addressed by SGFEM).

3.1.1 Global Enrichment Functions

To fix the idea of an enrichment, we list the most common global enrichment functions that one meets in applications.

Discontinuity

A discontinuity at a crack or an interface is one of the phenomena where XFEM is applied. We talk about a *strong* discontinuity, if the quantity of interest is discontinuous (pressure in fluid mechanics, stress in mechanics, potential in electrostatics etc.). The term *weak* discontinuity is then used when the derivative of the quantity of interest is discontinuous (fluid velocity, strain or damage, electric current etc.). See Figure 3.2 for comparison of the two types.

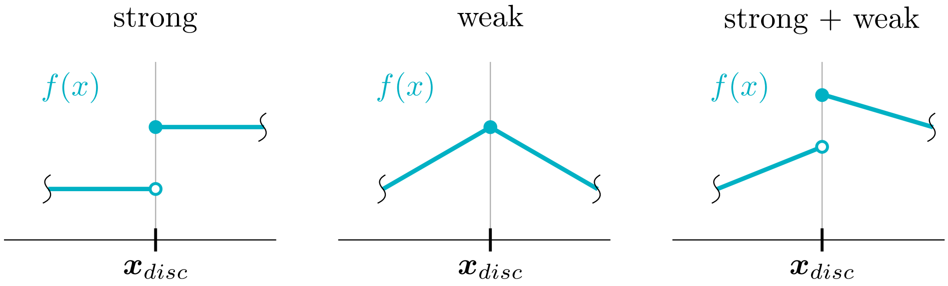


Figure 3.2: Types of discontinuities – strong, weak and their combination.

The discontinuity is commonly described by a *level-set* function – a signed function, that assigns positive values to one part of the domain Ω, negative values to the other part and is zero at the discontinuity/interface Γ_{disc}, for example

$$\gamma(\boldsymbol{x}) = \pm \min \left\| \boldsymbol{x} - \boldsymbol{x}_{disc} \right\| \quad \forall \boldsymbol{x}_{disc} \in \Gamma_{disc}, \ \forall \boldsymbol{x} \in \Omega. \tag{3.5}$$

The global enrichment function then takes the form of a signum or a Heaviside function

$$s(\boldsymbol{x}) = \operatorname{sign}(\gamma(\boldsymbol{x})), \tag{3.6}$$

$$s(\boldsymbol{x}) = \mathrm{H}(\gamma(\boldsymbol{x})). \tag{3.7}$$

for a strong discontinuity, or

$$s(\boldsymbol{x}) = \left| \gamma(\boldsymbol{x}) \right|. \tag{3.8}$$

for a weak discontinuity, respectively.

The discontinuity location Γ_{disc} does not always have to be known explicitly and/or it can be time dependent, $\Gamma_{disc}(t)$. In that situation the signed distance function $\gamma(\boldsymbol{x}, t)$ can be obtained as a solution of auxiliary problem, or be a part of the searched solution (e.g. in the article [27] where the moving interface of two-phase flow is a solution of a transport equation).

Singularity

Another problematic phenomenon that damages the finite element approximation is singular or high gradient behavior often present in the solution. This kind of enrichment has its application at crack tips, re-entrant corners, point sources in 2d or line sources in 3d, see Figure 3.3. The last mentioned are important in the reduced dimensional models where they play the role in the coupling between dimensions; these are of our main interest.

A function with its singularity concentrated in a single point, denoted $\boldsymbol{x}_e$ for singularity (enrichment) e, has a symmetric radial character. Therefore the polar coordinates in 2d are commonly used and the global enrichment function $s(r, \theta)$ is defined in this coordinate system with the origin at $\boldsymbol{x}_e$. The actual form of $s(r, \theta)$ is based on the solution of a local auxiliary problems, e.g. the solution of a simple 2d Laplace problem with a single point source is $\log(r)$ dependent. Other singular enrichment functions are typically r^α, $\alpha \in \mathbb{R}$ dependent. More examples not only in mechanics, namely

$$r^{3/2},\ r^{1/2},\ r^{-1/2},\ r^{-1},\ r^{-3/2},\ r^{-2},$$

are collected in the Natarajan's thesis [28], in the overview table 2.3-2.5.

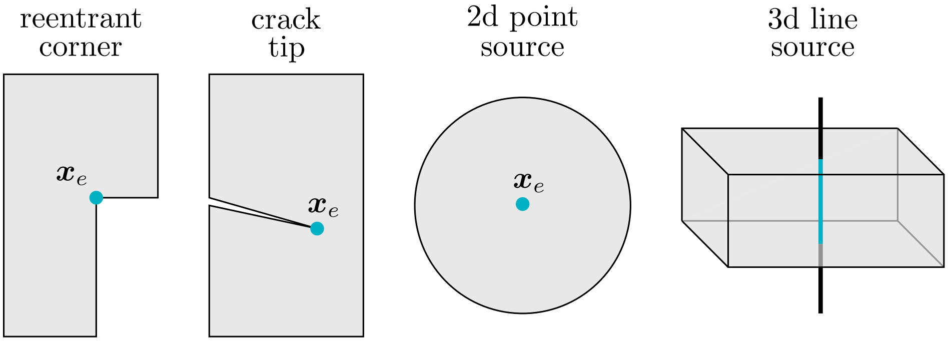


Figure 3.3: Examples of singularities.

In three dimensional space the spherical or cylindrical coordinate systems can be applied when the singularity is concentrated in a point or on a line, respectively. For example a crack front or a line source in 3d has the same strength of the singularity as in 2d, thus there is no need for an enrichment in the length axis of the cylindrical coordinates.

High gradient in the solution occurs also in the vicinity of discontinuities. A demonstrative examples are boundary (or interior) layers in some convection dominated transport models. Abbas et al. in the article [29] described an XFEM with a set of regularized (smooth) step functions that can capture arbitrary gradients and applied that method on linear advection-diffusion equation.

3.1.2 Enrichment Zone

Depending on the enrichment type (discontinuous or singular), the enrichment zone is chosen adequately. In case of discontinuity, the effect is local on the elements intersected by Γ_{disc}, thus only these elements are to be enriched. The enrichment zone Z is then defined implicitly by those intersected elements, see the left part of Figure 3.4.

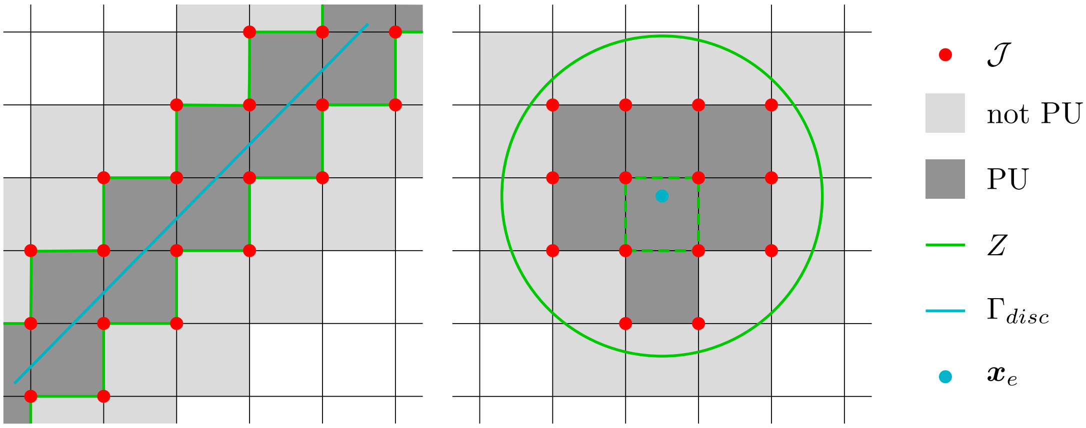


Figure 3.4: Enrichment zones in case of a discontinuity (on the left) and a singularity (on the right). In case of the singularity, the large green circle denotes the fixed (geometrical) enrichment zone; the topological enrichment would only enrich the nodes of the element denoted by the green dashed line.

The situation in case of singular enrichments is trickier. Typically the singular effects reach not only to the element where the singularity is located but also in its surroundings, where steep gradients are present. A proper enrichment zone in the vicinity of the singularity must be chosen so the method converges optimally and/or gives satisfying approximation error. If it is too small, the method can loose its approximation properties and its convergence rate might decrease. If the enrichment zone is too large, a high amount of degrees of freedom can be added unnecessarily and the method can become too computationally expensive. In some cases, the additional degrees of freedom can become almost linearly dependent which then causes ill-conditioning of the linear system.

This topic is originally addressed in [30], where the fixed enrichment zone size is suggested for standard XFEM, including some ideas for integral quadrature improvements and decreasing the ill-conditioning. In the same time, authors of [31] have the same aim and introduce the terms *topological* and *geometrical* enrichment. The former one is a kind of enrichment where only the element that includes the singularity is enriched. The later one defines a fixed enrichment zone in the same manner as it is in the reference above. In [4], the geometrical enrichment is also viewed as necessary to reach the optimal convergence, on the other hand it is mentioned that for a crack branching or multiple cracks, the enrichment zone is preferred as small as possible, i.e., topological enrichment is applied while sacrificing the optimal convergence rate. Both types of singular enrichment zones are shown on the right side of Figure 3.4.

Since the singularities of our interest have a radial character, we shall specify the size of the geometrical enrichment zone by so called *enrichment radius* R_e. We think this aspect of singular enrichments does not receive appropriate attention in XFEM literature, since the size of the enrichment zone can influence the solution significantly and is often given fixed and without any explanation. There is no general recipe available, up to our best knowledge, except the two following hints.

Gupta and Duarte in [32] provide an a priori estimate for the enrichment radius $R_e > Ch^{-2p}$ for a 2d linear elasticity crack problem, where h is the mesh parameter and p is the FE polynomial order. The estimate derivation is dependent on the problem specific bilinear form and the enrichment function $\boldsymbol{s}(r,\theta) = \sqrt{r}\boldsymbol{f}(\theta)$. The main idea behind is the following: to achieve the optimal convergence of order p, choose R_e such that the restriction of the solution to a narrow band of unenriched elements adjacent to the enrichment zone belongs to the Hilbert space of order $p+1$. According to the estimate, the minimum size of the enrichment zone is not fixed and it decreases with the mesh refinement. Therefore, if a fixed large enough geometrical enrichment is adopted, an optimal convergence rate will be achieved regardless of the value of the constant C. On the other hand, the constant C is not specified precisely, so for a practical application (with a given mesh), the estimate does not answer the proper value of R_e. The authors further state in their conclusion that the geometrical enrichment zone is necessary to obtain optimal convergence rate in this type of problems and that large enrichment zones can lead to ill-conditioning of the underlying linear system, which is in agreement with our experience in our models.

The other reference of choosing the enrichment zone is by us in [33]. There we suggest an a posteriori analyses of the enrichment radius for the Poisson equation. We consider splitting the solution onto a regular and a singular part and we measure the approximation error of the singular part in the unenriched part of the domain. This error should be then balanced with the approximation error of the regular part of the solution. This matter is discussed later, giving more details in section 4.4.3.

3.2 Enrichment Methods

We now discuss different choices of the local enrichment functions $L(\boldsymbol{x})$ and their particular aspects and properties. All the presented XFEM methods considered below are using the standard linear finite element shape functions $N_\alpha(\boldsymbol{x})$, $N_\alpha(x_\alpha) = 1$, $\alpha \in \mathcal{I} = \{1 \dots N\}$, associated with the node x_α of the triangulation, as the partition of unity.

3.2.1 Corrected XFEM

The Corrected XFEM was introduced by T. P. Fries in [4] and put in broad context of different XFEMs in the detailed overview [5]. He recognizes the *reproducing* and *blending* elements. The former are the elements where all the nodes are enriched, therefore the complete partition of unity is present and the enrichment function can be reproduced exactly. The later are the elements which have only some of its nodes enriched.

In the article, it is shown that the blending elements suffer from two drawbacks – lack of a partition of unity fails to reproduce enrichment function exactly; unwanted terms show up in assembly on these elements which can significantly increase the approximation error.

To overcome these drawbacks, the corrected XFEM introduces the *ramp function*, built from the linear basis functions,

$$
\begin{aligned}
G(\boldsymbol{x}) &= \sum_{\alpha \in \mathcal{J}} N_\alpha(\boldsymbol{x}) && (3.9) \\
&= \begin{cases} 0 & \text{on unenriched elements,} \\ 1 & \text{on reproducing elements (all the nodes are enriched),} \\ \mathit{ramp} & \text{on blending elements (some of the nodes are enriched).} \end{cases}
\end{aligned}
$$

and modifies the enrichment function into the form

$$
\begin{aligned}
L(\boldsymbol{x}) &= G(\boldsymbol{x}) s_(\boldsymbol{x}), && (3.10) \\
\phi_\alpha(\boldsymbol{x}) &= N_\alpha(\boldsymbol{x}) L(\boldsymbol{x}), \quad \alpha \in \mathcal{J}^*. && (3.11)
\end{aligned}
$$

Note the set $\mathcal{J}^*$ which is slightly bigger than $\mathcal{J}$. It includes the nodes on the blending elements which were previously unenriched, see Figure 3.5. Thus $\mathcal{J} \subset \mathcal{J}^*$.

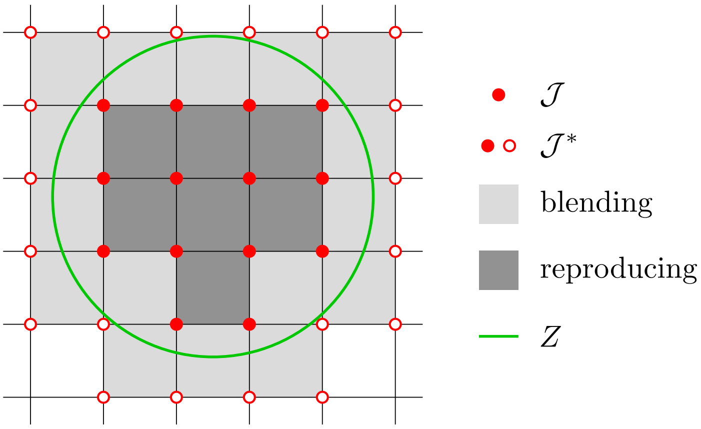


Figure 3.5: Radial enrichment zone on a quadrilateral mesh – blending and reproducing elements.

The ramp function keeps the properties on standard and reproducing elements (is constant 0 and 1, respectively), but on blending elements it creates smooth transition between standard and enriched approximation. Due to additional DoFs the modified enrichment function can be reproduced exactly and there is no lack of partition of unity. The approximation of the original enrichment function is of course worse but it does not damage the convergence of the method.

In the same work, i.e. [4], author further suggest the *shifted* enrichment functions in order to preserve the property of the standard FE approximation at nodes $h(\boldsymbol{x}_\alpha) = a_\alpha$: the value at the node is equal to the corresponding degree of freedom. The enrichment functions must be then zero at the nodes which is satisfied in the form

$$L_\alpha(\boldsymbol{x}) = G(\boldsymbol{x})\left[s(\boldsymbol{x}) - s(\boldsymbol{x}_\alpha)\right], \quad \alpha \in \mathcal{J}^*. \tag{3.12}$$

The property of the shifted formulation enables us to prescribe Dirichlet boundary condition such that $a_\alpha = h_D(\boldsymbol{x}_\alpha)$.

It has been also shown in many cases that both ramp function and shifting are needed to obtain optimal convergence rate. In [34], authors analyze a more general form of a ramp function (calling the method a weighted XFEM) and compare different alternatives of shifting on crack and dislocation problems. The methods described above can be then seen as special types of the weighted XFEM.

3.2.2 Stable Generalized FEM

Babuška and Banerjee developed a different enrichment strategy based on GFEM, for which they created a theoretical framework in [6] and called it the Stable Generalized Finite Element Method. In [7], the authors elaborate the problem in 2d and show the application of the SGFEM on elastic fracture model. The proposed method is supposed to overcome the common problem of the ill-conditioning of the stiffness matrix coming out of enrichment methods.

Particularly the ill-conditioning is observed when using the linear shape functions as the PU which results in some cases in almost linearly dependent degrees of freedom. We also often see that the ill-conditioning is sensitively dependent on the underlying mesh, i.e., the position of a discontinuity or a singularity with respect to element nodes or edges. This can then lead to much worse conditioned stiffness matrix than the one of the FEM and consequently to the loss of accuracy of the solution of the associated linear system. An example is given in [6] where the condition number is increasing with h^{-4} in case of GFEM, compared with the growth with h^{-2} in case of standard FEM for the second order problem.

The target property of the SGFEM is to retain the convergence rate of the XFEM while keeping the conditioning of the associated linear system close to the FEM. Beside that, a good approximation of the enrichment function on blending elements comes naturally without any special treatment. The approximation also holds the property of standard FE approximation at nodes $h(\boldsymbol{x}_\alpha) = a_\alpha$ (same as in case of shifting in the Corrected XFEM).

The enrichment function is defined as the subtraction of the global enrichment function and its interpolation

$$L_\alpha|_T(\boldsymbol{x}) = s(\boldsymbol{x}) - \pi_T(s)(\boldsymbol{x}), \quad \alpha \in \mathcal{J}. \tag{3.13}$$

for any element T of the mesh that have at least one enriched node. The interpolation π_T is built using the linear shape functions associated with nodes $\mathcal{I}_N(T)$ of the element T

$$\pi_T(s)(\boldsymbol{x}) = \sum_{\beta \in \mathcal{I}_N(T)} N_\beta(\boldsymbol{x}) s(\boldsymbol{x}_\beta). \tag{3.14}$$

Notice that there are no additional enriched nodes on blending elements, like in $\mathcal{J}^e$ in (3.10) and (3.12), and no ramp function is involved. Since the interpolant is built using the regular FE shape functions, the additional computational costs lie in the evaluation of $s(\boldsymbol{x}_\beta)$, $\beta \in \mathcal{J}$, i.e., the enrichment function value at all enriched nodes.

Later in [35], the robustness of SGFEM regarding the relative position of a fracture and a mesh is addressed. It is stated that GFEM or XFEM is stable (SGFEM) when the following 3 conditions hold:

(a) it yields the optimal order of convergence $O(h)$,

(b) its conditioning is not worse than that of the FEM, i.e., the order of condition number of the stiffness matrix of the GFEM is not worse than $O(h^2)$, and

(c) its conditioning is robust with respect to the position of the mesh relative to the features of the problem.

It is shown in [35] that the method as defined in [6, 7], does not satisfy all the conditions for a particular choice of discontinuous enrichment in linear elasticity. To regain the optimal convergence, the enrichment spaces is enlarged by so-called linear Heaviside functions and linear singular functions, which resolve the approximation error of discontinuities in particular. Instead of using just piecewise constant Heaviside function H, two *linear* functions are added per node to the approximation:

$$\left\{ H, H\frac{x - x_i}{h_i}, H\frac{y - y_i}{h_i} \right\} \tag{3.15}$$

where i is an index of a node and h_i is the diameter of the largest element sharing $\boldsymbol{x}_i$. These functions can then approximate discontinuity in an arbitrary direction and also capture possible discontinuity of the gradient (if a weak discontinuity is also present). Similarly the singular enrichments at the tip of the crack can be adapted. Since we are not dealing with discontinuities in our work, we do not discuss this matter in any more details.

To meet the last condition, a local orthogonalization technique (LOT) is introduced. This technique orthogonalizes the local basis of the enrichment spaces with respect to the problem specific inner product, and keeps its approximation properties. Although LOT is applied on the specific problem, it is a general idea and might be also suitable for cases, where more than a single enrichment are present on an element.

The same group of authors in [36] study the SGFEM application to a 3d fracture mechanics problem. The transition from 2d to 3d is demonstrated not to be trivial. Two different enrichment basis for displacement are tested – vector and scalar. Also the topological versus the geometrical enrichment in 3d is compared in the sense of conditioning and convergence. Finally, they show that only one of their proposed methods has at least the first two properties listed above, regarding the specific 3d problem.

3.3 XFEM in Flow Problems on Meshes of Combined Dimensions

There is much less to be found on the usage of the XFEM in the field of flow modeling, especially regarding the dimensions coupling, than in mechanics. Apart from the references on various types of the XFEM in previous sections, we can name some works regarding flow and the XFEM.

Extensive work has been done in the area of modeling 1d-2d fractured domains with mixed finite elements by D'Angelo, Fumagalli and Scotti [37, 16, 38]. In there, the XFEM is used to incorporate additional degrees of freedom of zero order Raviart-Thomas basis functions on the intersected elements to allow the discontinuity in velocity. The trick is, that the same basis functions are used but with different support determined by the geometry of the discontinuity. Thus no step function is actually used to enrich the finite element space, like we would do in standard XFEM approach described in Section 3.1.1. This approach however is not suitable for the approximation of a singularity, since enriching the FE space by additional Raviart-Thomas basis functions would not help to capture neither the geometry of the singularity nor the adjacent steep gradients. The authors further analyze the stability and convergence properties of their approach, in particular the inf-sup condition. They also deal with various types of the discontinuity-element cross-sections and with branching of the fractures. The model is formulated in 1d-2d at the moment.

Another work regarding flow in porous media in coplanar 1d-2d fractured system is by Schwenck [39, 40]. The XFEM is applied in the primary formulation for pressure, the discontinuous enrichment is used. The approximation around fracture tips and model of fractures intersections is proposed.

Next, there are several publications on multi-phase fluid flow governed by Navier-Stokes equations using the XFEM for approximation on the interfaces in 2d ([41, 27]). However, these are a bit further from our needs and application.

4 Pressure Model with Singularities

Now we apply the XFEM on the well-aquifer pressure model and we study different enrichment methods and singular enrichments. Several important numerical aspects are inspected, mainly an adaptive quadrature and a choice of an enrichment radius with respect to the convergence of the methods.

At first the problem for the primary unknown is defined and its weak form is derived. Then XFEM discretizations for several enrichment methods are described for this particular singular problem. Later several numerical experiments are computed and the results are investigated in detail. The results presented here are partially covered in our article [33].

4.1 Coupled 1d-2d Model (Primary Weak Form)

We rearrange Problem 2.2.1 and we substitute for $\boldsymbol{u}_d$ from the Darcy's law and solve the problem with pressure p_d as the primary unknown quantity. Thus we obtain the following

Problem 4.1.1. Find $[p_1,\, p_2]$ satisfying

$$\operatorname{div}\left(-\delta_2 \boldsymbol{K}_2 \nabla p_2\right) = \delta_2 f_2 \qquad \text{in } \Omega_2, \tag{4.1a}$$

$$\operatorname{div}\left(-\delta_1 \boldsymbol{K}_1 \nabla p_1\right) = \delta_1 f_1 + \sum_{m\in\mathcal{M}} |\Gamma_w^m|\, \Sigma_w^m \delta_t(t-t^m) \qquad \text{in } \Omega_1^w,\ \forall w \in \mathcal{W}, \tag{4.1b}$$

$$\langle -\delta_2 \boldsymbol{K}_2 \nabla p_2 \cdot \boldsymbol{n} \rangle_w^m = \Sigma_w^m \qquad \forall w \in \mathcal{W},\ \forall m \in \mathcal{M}, \tag{4.1c}$$

$$\{p_2\}_w^m = g_w^m \qquad \forall w \in \mathcal{W},\ \forall m \in \mathcal{M}, \tag{4.1d}$$

$$p_d = g_{dD} \qquad \text{on } \Gamma_{dD},\ d = 1,2, \tag{4.1e}$$

$$\delta_d \boldsymbol{K}_d \nabla p_d \cdot \boldsymbol{n} = g_{dN} \qquad \text{on } \Gamma_{dN},\ d = 1,2 \tag{4.1f}$$

where

$$\Sigma_w^m = \delta_2(\boldsymbol{x}_w^m)\sigma_w^m \big(\langle p_2 \rangle_w^m - p_1(\boldsymbol{x}_w^m)\big).$$

We add the following assumptions and notation. We consider the hydraulic conductivity tensor $\boldsymbol{K}_d$ to be an invertible positive definite $d \times d$ matrix, for which we denote

$$\underline{k_d} = \inf_{\boldsymbol{x}\in\Omega_d} \lambda_{\min}(\boldsymbol{K}_d), \qquad \overline{k_d} = \sup_{\boldsymbol{x}\in\Omega_d} \lambda_{\max}(\boldsymbol{K}_d), \qquad 0 < \underline{k_d} \leq \overline{k_d}. \tag{4.2}$$

We denote the minimal and maximal cross-section δ_d of the domains Ω_d

$$\underline{\delta_2} = \inf_{\boldsymbol{x}\in\Omega_2} \delta_2(\boldsymbol{x}), \quad \overline{\delta_2} = \sup_{\boldsymbol{x}\in\Omega_2} \delta_2(\boldsymbol{x}), \qquad 0 < \underline{\delta_2} \le \overline{\delta_2}, \tag{4.3}$$

$$\underline{\delta_1} = \min_{w\in\mathcal{W}}(\pi\rho_w^2), \quad \overline{\delta_1} = \max_{w\in\mathcal{W}}(\pi\rho_w^2), \qquad 0 < \underline{\delta_1} \le \overline{\delta_1}. \tag{4.4}$$

We consider the standard Sobolev spaces $H^1(\Omega_d)$, $d = 1, 2$ and

$$H_0^1(\Omega_d) = \big\{q_d \in H^1(\Omega_d); q_d|_{\Gamma_{dD}} = 0\big\}.$$

which takes into account the Dirichlet boundary condition. Next we define the trial space V and the test space V_0:

$$V = H^1(\Omega_1) \times H^1(\Omega_2), \tag{4.5}$$

$$V_0 = H_0^1(\Omega_1) \times V_0^1(\Omega_2^1) \times \cdots \times V_0^M(\Omega_2^M), \tag{4.6}$$

with

$$\text{with } V_0^m(\Omega_2^m) = \big\{q_2 \in H_0^1(\Omega_2^m) : \{q_2\}_w^m = 0, \forall w \in \mathcal{W}\big\} \quad \forall m \in \mathcal{M}. \tag{4.7}$$

The spaces V, V_0 are equipped with the norm

$$\|q\|_V = \left(\|q_1\|^2_{H^1(\Omega_1)} + \|q_2\|^2_{H^1(\Omega_2)}\right)^{1/2}. \tag{4.8}$$

We denote the weak solution $p = [p_1, p_2] \in V$ and the test functions $q = [q_1, q_2] \in V_0$.

Below the weak form of Problem 4.1.1 is derived. Due to the choice of V_0^m in (4.7), the well edge integral is simplified:

$$\int_{\Gamma_w^m} p_2 q_2 \,\mathrm{d}s = \int_{\Gamma_w^m} \langle p_2\rangle_w^m \langle q_2\rangle_w^m \,\mathrm{d}s + \int_{\Gamma_w^m} \{p_2\}_w^m \{q_2\}_w^m \,\mathrm{d}s = |\Gamma_w^m|\, \langle p_2\rangle_w^m \langle q_2\rangle_w^m \,\mathrm{d}s. \tag{4.9}$$

We now apply the standard Galerkin method. We multiply the equations (4.1a) and (4.1b) by test functions q_d, integrate by parts over Ω_d using (4.1c)-(4.1f) and the assumption (4.9), then we sum up the two equations together to get

$$\begin{aligned} a(p,q) = &\int_{\Omega_2} \delta_2 \boldsymbol{K}_2 \nabla p_2 \cdot \nabla q_2 \,\mathrm{d}\boldsymbol{x} + \int_{\Omega_1} \delta_1 \boldsymbol{K}_1 \nabla p_1 \cdot \nabla q_1 \,\mathrm{d}\boldsymbol{x} \\ &+ \sum_{\substack{w\in\mathcal{W}\\ m\in\mathcal{M}}} |\Gamma_w^m|\, \delta_2 \sigma_w^m (\langle p_2\rangle_w^m - p_1(\boldsymbol{x}_w^m))(\langle q_2\rangle_w^m - q_1(\boldsymbol{x}_w^m)) \\ = &\int_{\Omega_2} \delta_2 f_2 q_2 \,\mathrm{d}\boldsymbol{x} + \int_{\Omega_1} \delta_1 f_1 q_1 \,\mathrm{d}\boldsymbol{x} + \int_{\Gamma_{2N}} g_{2N} q_2 \,\mathrm{d}s + \int_{\Gamma_{1N}} g_{1N} q_1 \,\mathrm{d}s \\ &= l(q). \end{aligned} \tag{4.10}$$

Considering the Dirichlet boundary conditions, we split the solution into three parts $p = p_0 + p_w + p_D$, while $p_0 \in V_0$ and $p_w, p_D \in V$ are functions chosen such that they satisfy (4.1d) on the wells edges and (4.1e) on the exterior boundary, respectively. Finally, we define the weak problem

Problem 4.1.2. Find $p \in V$ such that

$$a(p_0, q) = l(q) - a(p_w, q) - a(p_D, q) \qquad \forall q \in V_0. \tag{4.11}$$

for given $f_d \in L_2(\Omega_d)$ and $g_{dN} \in L_2(\Gamma_{dN})$, $d = 1, 2$, and $p_w, p_D \in V$ fixed.

Next we discuss some properties of the defined problem and provide several results to show the existence of a unique solution eventually. To this end we shall need two following results about compact mappings from [42], p. 103, Corollary 6.3 and Theorem 6.2:

Lemma 4.1.1. *Let Ω_1 be a closed interval in $\mathbb{R}^1$. The identity mapping $I : H^1(\Omega_1) \to C(\overline{\Omega}_1)$ is compact.*

Lemma 4.1.2. *Let $\Omega_2 \subset \mathbb{R}^2$ be a domain with Lipschitz boundary. The mapping $Z \in [H^1(\Omega_2) \to L_2(\partial\Omega_2)]$, which define traces, is compact.*

At first we bound the bilinear form of Problem 4.1.2.

Lemma 4.1.3. *The bilinear form a, defined in (4.10), is continuous on V_0:*

$$|a(p, q)| \leq \alpha_2 \|p\|_V \|q\|_V \quad \forall p, q \in V_0, \tag{4.12}$$

Proof. To bound the average terms in the form a, we define an auxiliary smooth function $\psi \in C^\infty(\overline{\Omega})$ such that

$$\nabla\psi \cdot \boldsymbol{n} = \begin{cases} 1 & \text{on } \Gamma_{int} \\ 0 & \text{on } \Gamma_{ext}. \end{cases}$$

Then we have the upper bound

$$\begin{aligned} \sum_{\substack{w\in\mathcal{W}\\ m\in\mathcal{M}}} \langle q\rangle_w^m &= \sum_{\substack{w\in\mathcal{W}\\ m\in\mathcal{M}}} \int_{\Gamma_w^m} q(\nabla\psi\cdot\boldsymbol{n})\,\mathrm{d}s = \int_{\partial\Omega_2} q(\nabla\psi\cdot\boldsymbol{n})\,\mathrm{d}s = \int_\Omega \big(q\Delta\psi + \nabla q\cdot\nabla\psi\big)\,\mathrm{d}\boldsymbol{x} \\ &\leq \|q\|_{L_2(\Omega)}\|\Delta\psi\|_{L_2(\Omega)} + \|\nabla q\|_{L_2(\Omega)}\|\nabla\psi\|_{L_2(\Omega)} \\ &\leq C_{avg}(\partial\Omega_2, \Omega_2)\Big(\|q\|_{L_2(\Omega)} + \|\nabla q\|_{L_2(\Omega)}\Big) \end{aligned} \tag{4.13}$$

Further, due to Lemma 4.1.1, $q_1 \in H_0^1(\Omega_1)$ is continuous and bounded on Ω_1, and therefore $|q_1(\boldsymbol{x}_w^m)| \leq \|q\|_{H^1(\Omega_1)}$. Then we obtain the final bound

$$\begin{aligned} |a(p,q)| &\leq \overline{\delta_2}\overline{k_2}\,\|\nabla p_2\|_{L_2(\Omega_2)}\|\nabla q_2\|_{L_2(\Omega_2)} + \overline{\delta_1}\overline{k_1}\,\|\nabla p_1\|_{L_2(\Omega_1)}\|\nabla q_1\|_{L_2(\Omega_1)} \\ &\quad + \sum_{\substack{w\in\mathcal{W}\\ m\in\mathcal{M}}} \overline{\delta_2}\,|\Gamma_w^m|\,\sigma_w^m\,|\langle p_2\rangle_w^m - p_1(\boldsymbol{x}_w^m)|\,|\langle q_2\rangle_w^m - q_1(\boldsymbol{x}_w^m)| \\ &\leq C\Big(\|p_2\|_{H^1(\Omega_2)} + \|p_1\|_{H^1(\Omega_1)}\Big)\Big(\|q_2\|_{H^1(\Omega_2)} + \|q_1\|_{H^1(\Omega_1)}\Big) \\ &\leq \alpha_2\|p\|_V\|q\|_V \quad \forall p, q \in V_0, \end{aligned} \tag{4.14}$$

with a constant $\alpha_2(\overline{\delta_2}, \overline{k_2}, \overline{\delta_1}, \overline{k_1}, \sigma_w^m, |\Gamma_w^m|, C_{avg})$. ☐

Next we want to show that the bilinear form a is coercive on V_0. We state the following lemma

Lemma 4.1.4. *Let there be a Dirichlet boundary condition prescribed on any non-empty subset of* Γ_{dD}, $d = 1, 2$. *Then the bilinear form* a, *defined in* (4.10), *is coercive on* V_0*:*

$$a(q, q) \geq \alpha_1 \|q\|_V^2 \quad \forall q \in V_0. \tag{4.15}$$

Proof. We take similar steps as in the proof of the Poincaré's inequality by contradiction in [43], p. 290, (also in [42], p. 9). Suppose that (4.15) does not hold. Then $\forall n \in \mathbb{N}$ exists a function $q^n \in V_0$ satisfying

$$\|q^n\|_V^2 > n \cdot a(q^n, q^n). \tag{4.16}$$

Without loss of generality, let us consider $\|q^n\|_V = 1$. Then (4.16) implies

$$a(q^n, q^n) < \frac{1}{n}\|q^n\|_V^2 = \frac{1}{n}. \tag{4.17}$$

Space V_0 is a Hilbert space which is reflexive, then there exists a weakly convergent subsequence $\{q^{n_k}\}_{k=1}^{\infty} \subset \{q^n\}_{n=1}^{\infty}$ and a function $q \in V_0$ such that

$$q^{n_k} \rightharpoonup q \quad \text{in } V_0. \tag{4.18}$$

Next the bilinear form a can be bounded below

$$\begin{aligned} a(q^{n_k}, q^{n_k}) \geq{} & \underline{\delta_2}\,\underline{k_2}\|\nabla q_2^{n_k}\|_{L_2(\Omega_2)}^2 + \underline{\delta_1}\,\underline{k_1}\|\nabla q_1^{n_k}\|_{L_2(\Omega_1)}^2 \\ & + \sum_{\substack{w \in \mathcal{W} \\ m \in \mathcal{M}}} |\Gamma_w^m|\,\underline{\delta_2}\sigma_w^m(\langle q_2^{n_k}\rangle_w^m - q_1^{n_k}(\boldsymbol{x}_w^m))^2. \end{aligned} \tag{4.19}$$

Since all terms on the right hand side of (4.19) are non-negative, each term must converge to zero to satisfy (4.17). The weak lower semi-continuity

$$\|\nabla q\|_{L_2(\Omega_d)} \leq \liminf_{k \to \infty} \|\nabla q^{n_k}\|_{L_2(\Omega_d)}, \qquad d = 1, 2 \tag{4.20}$$

and (4.19) with $k \to \infty$ then implies that

$$\nabla q_1 = \nabla q_2 = 0. \tag{4.21}$$

As a consequence of (4.18) and Lemma 4.1.1, we have a point convergence

$$q_1^{n_k}(\boldsymbol{x}_w^m) \to q_1(\boldsymbol{x}_w^m) \qquad \forall w \in \mathcal{W}, \forall m \in \mathcal{M}. \tag{4.22}$$

As a consequence of (4.18) and Lemma 4.1.2, we have a strong convergence

$$q_2^{n_k}|_{\Gamma_w^m} \to q_2|_{\Gamma_w^m} \implies \langle q_2^{n_k}\rangle_w^m \to \langle q_2\rangle_w^m \qquad \forall w \in \mathcal{W}, \forall m \in \mathcal{M} \tag{4.23}$$

Finally for the coupling term, we have the following pressure equality, reminding that $\sigma_w^m > 0$,

$$q_2|_{\Gamma_w^m} = \langle q_2\rangle_w^m = q_1(\boldsymbol{x}_w^m) \qquad \forall w \in \mathcal{W}, \forall m \in \mathcal{M}. \tag{4.24}$$

We see that pressure is constant in each aquifer and in each well. Due to the coupling term, pressure must be equal between each pair well-aquifer and thus pressure is identical everywhere. Finally, if we prescribe Dirichlet boundary condition on any nonempty subset of Γ_{dD}, we have $q = 0$, since $q \in V_0$, and $\|q\|_V = 0$ which is in contradiction with the assumption. $\square$

The existence and uniqueness of the solution can be shown via the Lax-Milgram theorem (e.g. in [42], p. 29) due to the ellipticity of the bilinear form a and the boundedness of the problem, using the results of Lemmas 4.1.3 and 4.1.4:

Theorem 4.1.5. *Let $a : V \times V \to \mathbb{R}$ be a continuous coercive bilinear form on V_0, i.e.*

$$|a(p_0, q)| \le \alpha_2 \|p_0\|_V \|q\|_V \qquad \forall p_0, q \in V_0, \tag{4.25}$$

$$a(q, q) \ge \alpha_1 \|q\|_V^2 \qquad \forall q \in V_0, \tag{4.26}$$

with the constants α_2 from (4.14) and α_1 from (4.15).
Then there exists a unique solution $p \in V$ to Problem (4.1.2) for any $f_d \in L_2(\Omega_d)$ and $g_{dN} \in L_2(\Gamma_{dN})$, $d = 1, 2$ and $p_w, p_D \in V$ fixed, such that

$$\|p_0\|_V \le \frac{1}{\alpha_1}\|l\|_{V'} + \frac{\alpha_2}{\alpha_1}\left(\|p_w\|_V + \|p_D\|_V\right). \tag{4.27}$$

We point out a practical consequence of Theorem 4.1.5 and the argumentation in the proof of coercivity of a. One can observe that the uniqueness of the solution can be attained by fixing the pressure at the top of a single well (e.g. a pumping well where the pressure can be measured), possibly with Neumann boundary condition at the rest of the boundary. This might be useful when there is limiting amount of data on aquifer and no flow boundary condition is considered.

Eventually, we add a practical assumption and we approximate the pressure fluctuation $g_w^m \approx 0$. This assumption is based on the wells being very small in diameter, $\rho_w << |\Omega_2^m|$, and so that pressure is changing minimally along Γ_w^m. Then we put $a(p_w, q) \approx 0$ in Problem (4.1.2) and thus we effectively neglect the given boundary condition for the pressure fluctuation in (4.1d). The practical impact of this assumption is the reduction of input data, since in real cases it is apparently impossible to measure g_w^m. In [23], a similar assumption is taken and the fluctuation term in the weak form is neglected

$$\int_{\Gamma_w^m} \{p_2\}_w^m \{q_2\}_w^m \,\mathrm{d}s \simeq 0. \tag{4.28}$$

Effects of this assumption are studied in details there and an error estimate is provided.

4.2 Discretization

We now choose a particular XFEM discretization, using the methods described in Section 3.2, and apply it to the weak form (4.10).

We omit the upper index m, denoting aquifer, till the end of this section for the sake of the simplicity of notation. Extension to the multi-aquifer system is straightforward. In fact in case of the pressure model, we suppose the aquifers to be parallel 2d planes, defined in the xy axes at different levels of z-axis. The wells are considered perpendicular to the aquifers (i.e. in z-axis) and all have a constant radius. Therefore we use the same triangulation for every aquifer in our implementation and therefore the discrete spaces are also identical. This simplification is of solely technical nature and does not mean any loss of generality of the used discretization schemes. The XFEM is used to couple the 1d and 2d domains and to better approximate the singular solution in the aquifer, so only the finite element space of the 2d domain is extended.

Now we look for the finite dimensional space V_h of the continuous space V introduced in the weak form (4.10), so that $V_h \subset V$. We build the discrete space on domains of each dimension

$$V_h = V_{1h} \times V_{2h}, \qquad V_{2h} = V_{2h}^{reg} \oplus V_{2h}^{enr} \tag{4.29}$$

where V_{2h} consists of the regular part and the enriched part which we define later below.

We denote a patch $\omega_{d\alpha}$ which is the union of elements sharing a node $\boldsymbol{x}_\alpha \in \mathcal{T}_d$

$$\omega_{d\alpha} = \{T_d^i : \boldsymbol{x}_\alpha \in T_d^i, i \in \mathcal{I}_{dE}\}, \quad \alpha \in \mathcal{I}_{dN}.$$

Next, let $N_{d\alpha}(\boldsymbol{x})$ be the standard linear finite element shape functions associated with nodes $\boldsymbol{x}_\alpha$, having support on $\bar{\omega}_{d,\alpha}$ and satisfying the Kronecker delta property $N_{d\alpha}(\boldsymbol{x}_\beta) = \delta_{\alpha\beta}$, $\alpha, \beta \in \mathcal{I}_{dN}$. Then we can define

$$V_{dh}^{reg} = \operatorname{span}\{N_\alpha\}, \quad \alpha \in \mathcal{I}_{dN},\ d = 1, 2, \tag{4.30}$$

understanding that $V_{1h} = V_{1h}^{reg}$ in our problem. Similarly we define the enriched space

$$V_{2h}^{enr} = \operatorname{span}\{N_{2\alpha} L_{\alpha w}\}, \quad \alpha \in \mathcal{J}_{2N}^w,\ w \in \mathcal{W}, \tag{4.31}$$

where $\mathcal{J}_{2N}^w \subset \mathcal{I}_{2N}$ denotes the indices of enriched nodes in $\mathcal{T}_2$ by the well w, and the functions $N_{2\alpha} L_{\alpha w}$ are the local enrichment functions as in (3.4), while $N_{2\alpha}$ are playing the role of PU. Thus we can write the enriched solution in the form

$$p_{2h}(\boldsymbol{x}) = \sum_{\alpha \in \mathcal{I}_{2N}} a_\alpha N_{2\alpha}(\boldsymbol{x}) + \sum_{w \in \mathcal{W}} \sum_{\alpha \in \mathcal{J}_{2N}^w} b_{\alpha w} N_{2\alpha}(\boldsymbol{x}) L_{\alpha w}(\boldsymbol{x}), \tag{4.32}$$

where a_α are the standard degrees of freedom and $b_{\alpha w}$ are the degrees of freedom coming from the enrichment of the well w.

We point out that the selected discrete space $V_{2h}^{reg} \not\subset V$ and thus $V_{2h} \not\subset V$. This is due to the fact that the linear shape functions cannot have zero fluctuations on

Γ_w^m, which contradicts the property of the continuous test space in (4.7). In this view, our discretization is non-conforming.

From the physical perspective, the pressure in the vicinity of the well is above all governed by the large difference between the pressure inside the well and in the aquifer. Then the fluctuations of the linear shape functions will be small, influenced mainly by the regular part of the solution and the size of elements. If the elements intersecting the well are large, then the fluctuations will be even less significant. The fluctuations might increase when element size closes up to the scale of the well, but then this is not the case we are the most interested in and the singular XFEM enrichment is unnecessary. As it will be observed later in the numerical experiments, the non-conformity does not corrupt the convergence of the method and does not increase the approximation error in the problems of our interest.

We now look at the choice of the enrichment function $L_{\alpha w}$ in our particular case. Below we select a proper global enrichment function and then define $L_{\alpha w}$ according to one of the XFEM.

4.2.1 Enrichment Function

The enrichment function can be obtained from the solution of a local problem on the neighborhood of the well w. Let Ω be an annulus with center $\boldsymbol{x}_w$, inner radius ρ_w and arbitrary outer radius $D \gg \rho_w$. The solution of the Laplace equation $-\Delta p = 0$ on Ω with any radially symmetric boundary conditions (e.g. in [43], Section 2.2.1) has a form

$$p = a \log(r_w) + b, \tag{4.33}$$

where r_w is a distance function

$$r_w(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{x}_w\| = \sqrt{(x - x_w)^2 + (y - y_w)^2}. \tag{4.34}$$

Thus the pressure head would go to infinity while closing to the center of the well. Keeping in mind the radius of the well ρ_w and the local solution (4.33), we introduce a (global) enrichment function

$$s_w(\boldsymbol{x}) = \begin{cases} \log(r_w(\boldsymbol{x})) & r_w > \rho_w, \\ \log(\rho_w) & r_w \leq \rho_w, \end{cases} \tag{4.35}$$

see Figure 4.1, and its gradient

$$\nabla s_w(\boldsymbol{x}) = \begin{cases} \frac{1}{r_w^2(\boldsymbol{x})}(\boldsymbol{x} - \boldsymbol{x}_w) & r_w > \rho_w, \\ 0 & r_w \leq \rho_w. \end{cases} \tag{4.36}$$

In case of more aquifers in the model, it is natural to use the same s_w on each aquifer since it depends only on $r_w(x, y)$ and the wells have constant center $\boldsymbol{x}_w = [x_w, y_w]$ and radius ρ_w along the z-axis.

In contrast to global enrichment methods, the XFEM and the SGFEM apply the enrichment functions only locally. Since the enrichment function is radial, it is

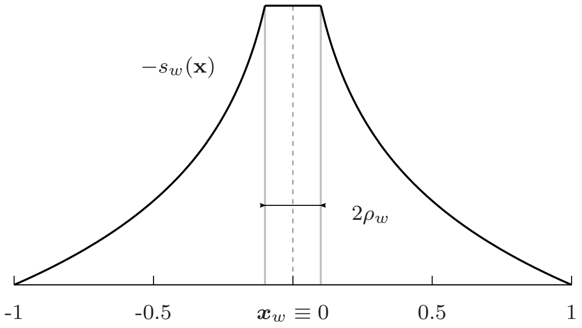


Figure 4.1: The logarithmic enrichment function.

natural to consider the enriched domain $Z_w = B_{R_w}(\boldsymbol{x}_w)$ of the well w given by the enrichment radius R_w. Thus for each well w all the nodes of $\mathcal{T}_2$ in Z_w are enriched and we denote the enriched element indices by

$$\mathcal{J}^w_{2N} = \{\alpha \in \mathcal{I}_{2N} : \boldsymbol{x}_\alpha \in Z_w,\, w \in \mathcal{W}\}. \tag{4.37}$$

We also define the index set of enriched elements, i.e. elements in which at least one node is enriched

$$\mathcal{J}^w_{2E} = \{i \in \mathcal{I}_{2E} : \boldsymbol{x}_\alpha \in T^i_2,\, \alpha \in \mathcal{J}^w_{2N},\, w \in \mathcal{W}\}. \tag{4.38}$$

on elements T^i_2 with enriched nodes $\boldsymbol{x}_\alpha \in T^i_2$, $i \in \mathcal{I}_{2E}$,

We define the following enrichment schemes, according to the section 3.2, which we later compare in numerical tests. At first we use the simplest scheme

$$L_{\alpha w} = s_w(\boldsymbol{x}), \quad \alpha \in \mathcal{J}^w_{2N}, w \in \mathcal{W}. \tag{4.39}$$

which we refer to as **standard XFEM** later in this section. For the Corrected XFEM, we start with the ramp function G, see (3.10), having the local enrichment function in the form

$$L_{\alpha w} = G_w(\boldsymbol{x}) s_w(\boldsymbol{x}), \quad \alpha \in \mathcal{J}^{*w}_{2N}, w \in \mathcal{W}. \tag{4.40}$$

We call this scheme **ramp function XFEM**. Next the Corrected XFEM can also include the shifting, see (3.12),

$$L_{\alpha w} = G_w(\boldsymbol{x}) \left[s_w(\boldsymbol{x}) - s_w(\boldsymbol{x}_\alpha)\right], \quad \alpha \in \mathcal{J}^{*w}_{2N}, w \in \mathcal{W}, \tag{4.41}$$

which we refer to as the **shifted XFEM**. We remind that in the Corrected XFEM, the index set of the enriched nodes is extended due to the ramp function such that every node of elements that have non-zero intersection with the area Z_w is enriched:

$$\mathcal{J}^{*w}_{2N} = \{\alpha \in \mathcal{I}_{2N} : \boldsymbol{x}_\alpha \in T^j_2,\, j \in \mathcal{J}^w_{2E},\, w \in \mathcal{W}\}, \tag{4.42}$$

see Figure 3.5 above. Finally, we have the **SGFEM** scheme, see (3.13),

$$L_{\alpha w}|_{T^j_2} = \left[s_w(\boldsymbol{x}) - \pi_{T^j_2}(s_w)(\boldsymbol{x})\right], \quad \alpha \in \mathcal{J}^w_{2N},\, j \in \mathcal{J}^w_{2E},\, w \in \mathcal{W}, \tag{4.43}$$

on enriched elements T^j_2, with the interpolation operator

$$\pi_{T^j_2}(s_w)(\boldsymbol{x}) = \sum_{\beta \in \mathcal{I}_N(T^j_2)} s_w(\boldsymbol{x}_\beta) N_\beta(\boldsymbol{x}). \tag{4.44}$$

4.2.2 Integration on Enriched Elements

In order to compute the entries of the system matrix, we need to integrate the expressions containing the enrichment functions. These of course can be non-polynomial, like they are in our case. The standard quadrature rules are not appropriate any more, for they are constructed to integrate precisely only polynomials up to a given degree. The higher requirements on the integration precision are the price for using enrichment functions and a coarse mesh.

There are two aspects which the integration must handle properly:

- the steep gradient of enrichment shape functions in the vicinity of the singularity,
- the singularity cut-off edge geometry.

One of the approaches to deal with these requirements is an adaptive quadrature. The element is further subdivided to split the integration area into much smaller pieces (subelements), which better align with the well edge. On the subelements, standard quadrature rules are applied, possibly of higher order. This way the integration is more accurate, increasing the number of quadrature points per element, but not bringing any more degrees of freedom into the linear system. In this section we discuss the adaptivity rules for the element subdivision, suggest an improvement and compare our strategy to the original one from which we have started, developed in [2].

Instability of Adaptive Quadrature

Gracie and Craig in [2] refine only subelements that cross the boundary of the well, using at most 12 refinements. This catches nicely the well edge but it works only when the well is placed at the node of an element or near the center of an element. When the well is placed near the edge of an element, there can be a large difference in the size of neighboring subelements, see Figure 4.2a. Although the integrand is computed precisely enough on the element with the well inside, the quadrature points on the neighboring elements (where the integrand has still large derivatives) are placed very sparsely and the integration error is large.

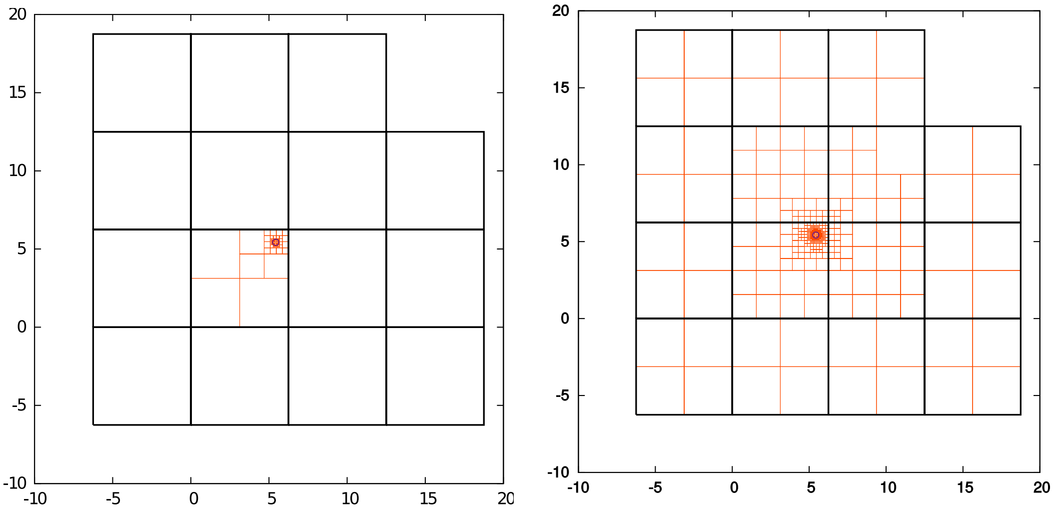
In order to overcome this instability of the adaptive quadrature, we have made an asymptotic analysis of the integration error presented in the next section.

Estimate of Quadrature Error

Let us assume only one well of radius ρ_w situated at the origin. In the case of elliptic equation, the term with the strongest singularity is

$$f(r_w) = (\nabla \log r_w)^2 \approx r_w^{-2} \tag{4.45}$$

which is also the worst term to integrate independently of the particular XFEM enrichment variant. Consider a square subelement S with a side δ and let us denote r_S its distance from the origin. We want to estimate the error of the 2d tensor



(a) refinement due to Gracie and Craig

(b) improved refinement

Figure 4.2: Comparison of the original and improved refinement techniques. Black lines denote enriched elements edges, red lines denote adaptive refinement (subelements edges) and the well edge is blue.

product Gauss quadrature rule of order n (n times n points) on the square S. Let us denote $\Pi^n f$ the projection of the integrand f to the space of polynomials that are integrated exactly. We were not able to find error estimates for 2d quadratures in the literature and deriving them would be extremely technical. However, we can make an observation that among the squares of the same r_S, the quadrature error is the highest for the squares lying on one of the axis. Assume without loss of generality a square on the X-axis. Since $|y| < \delta \ll |x|$, the monomials of $\Pi^n f$ containing y are negligible and we get the quadrature of order n in the radial (X-axis) direction. On the other hand for the square on the diagonal of the axes, the bilinear terms of a 2d quadrature effectively enhance its order to $2n$ in the radial (diagonal) direction. Due to the radial nature of the integrand, we can estimate the quadrature error on the square S by the error on the square S' with the same r_S laying on the X-axis, then we can neglect y monomials and use the error estimates for the 1d quadrature:

$$\int_S |f - \Pi^n f| \, \mathrm{d}\boldsymbol{x} \le \int_{S'} |f - \Pi^n f| \, \mathrm{d}\boldsymbol{x} \le \delta E^n((r_S, r_S + \delta)). \tag{4.46}$$

The E_n is the error of 1d Gauss quadrature of the order n (n quadrature points) over the interval $(r_w, r_w + \delta)$

$$E_n = \frac{\delta^{2n+1}(n!)^4}{(2n+1)((2n)!)^3} f^{(2n)}(\xi_n) \tag{4.47}$$

for some $\xi_n \in (r_w, r_w + \delta)$, see e.g. [44]. The expression $f^{(2n)}$ denotes a derivative of order $2n$ of a function f. Regarding the integrand (4.45), we have

$$\left| f^{(2n)}(r_w) \right| = (2n+1)! r_w^{-(2n+2)}. \tag{4.48}$$

Finally, we get an estimate for the quadrature error on a single subelement:

$$\int_S |f - \Pi^n f| \, \mathrm{d}\boldsymbol{x} \leq \alpha_n \left(\frac{\delta}{r_S}\right)^{2n+2}, \qquad \alpha_n = \left(\frac{(n!)^2}{(2n)!}\right)^2. \tag{4.49}$$

This estimate implies that we have to ensure $\delta < r_S$ in order to get a decent quadrature error and possibly to employ a higher order quadrature.

A Priori Adaptive Quadrature Rules

Let us denote r_{min} the minimum and r_{max} the maximum distance from a particular subelement S to the singularity center. Based on the analysis presented above, we propose following adaptive quadrature rules:

1. If $r_{max} < \rho_w$, the subelement is not refined and the quadrature is zero.
2. If $r_{min} < \rho_w < r_{max}$ and $\delta > 2^{-10}h$, the subelement is refined.
3. If $r_{min} < \rho_w < r_{max}$ and $\delta \leq 2^{-10}h$, 3 × 3 Gauss quadrature is used. The weights at quadrature points lying inside the well are set to zero.
4. If $r_{min} > \rho_w$ and $\delta > r_{min}/2$, the subelement is refined.
5. If $r_{min} > \rho_w$ and $\delta \leq r_{min}/2$, 3 × 3 Gauss quadrature is used.

These rules ensure $\delta < r_{min}/2$ outside the well, where the integrand is smooth. Subelements intersecting the well's boundary are refined using at most 10 refinement levels, since the integrand is discontinuous there and we cannot employ estimates from the previous section. The maximum number of levels is chosen so that we get the similar total number of quadrature points as in the quadrature used by Gracie and Craig in [2]. Using the proposed rules, the elements that do not contain the well are refined as well, see Figure 4.2b.

By implementing this approach, all the variants of the XFEM enrichment methods converge with optimal order. as the results of the numerical tests will show in section 4.4.

Quadrature in Polar Coordinates

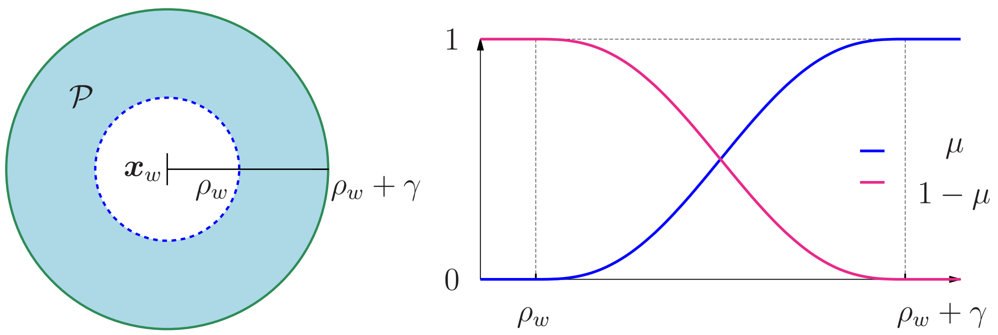
The idea for the usage of the quadrature in polar coordinates originates from the radial character of the terms that are integrated. The motivation is to reduce the number of quadrature points (refinement levels) of the previously described quadrature due to precise representation of the well edge and/or increase of accuracy.

We define a circular neighborhood of a well, i.e. a band of width γ (see Figure 4.3a),

$$\mathcal{P} = \{\boldsymbol{x} : \rho_w < |\boldsymbol{x} - \boldsymbol{x}_w| < \rho_w + \gamma\}. \tag{4.50}$$

Next we define a smooth step function (see Figure 4.3b)

$$\mu(z) = -2z^3 + 3z^2, \quad z = \frac{r_w - \rho_w}{\gamma}. \tag{4.51}$$



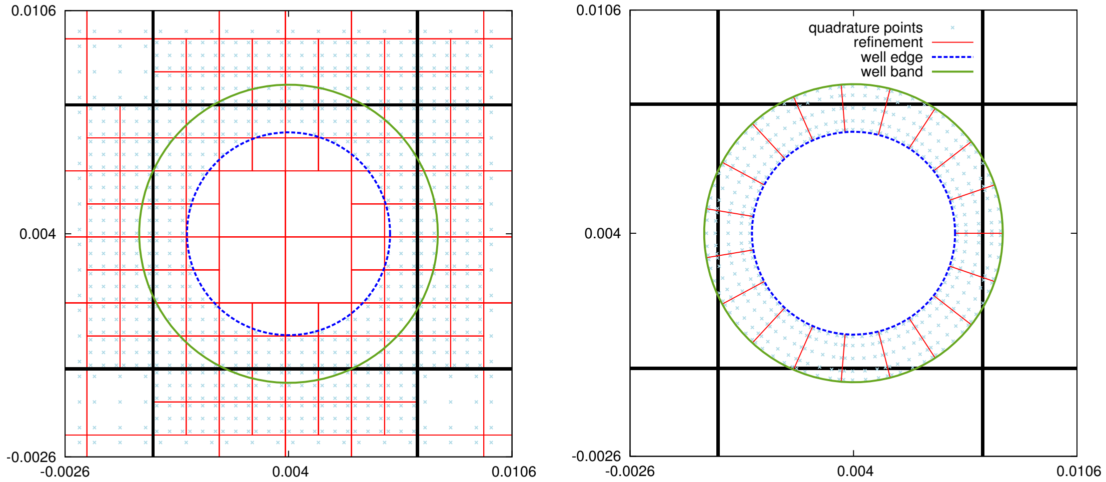
(a) well neighborhood $\mathcal{P}$ (b) smooth step function

Figure 4.3: Smooth step function μ on the well neighborhood $\mathcal{P}$.

We use μ as a partition of unity to divide an integral into 2 parts

$$\begin{aligned}\int_S v(\mathbf{x})\,\mathrm{d}\mathbf{x} &= \int_S \mu(r_w)v(\mathbf{x})\,\mathrm{d}\mathbf{x} + \int_S (1-\mu(r_w))v(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ &= \int_S \mu(r_w)v(\mathbf{x})\,\mathrm{d}\mathbf{x} + \int_0^{2\pi}\int_{\rho_w}^{\rho_w+\gamma} (1-\mu(r_w))v(\mathbf{x})r_w\,\mathrm{d}r\,\mathrm{d}\phi. \qquad (4.52)\end{aligned}$$

We see that the first integral vanishes when closing to the well edge, while the second grows. This corresponds to the character of the solution which is more radial in the vicinity of the well rather than far away from it. Therefore we transform the second integral into polar coordinates $\mathbf{x} \longleftrightarrow (r_w, \phi)$. The first integral is computed using the previously described quadrature, but with much less refinement levels,



(a) $\int_\Omega \mu(r_w)v(\mathbf{x})\,\mathrm{d}\mathbf{x}$ (b) $\int_0^{2\pi}\int_{\rho_w}^{\rho_w+\gamma}(1-\mu(r_w))v(\mathbf{x})r_w\,\mathrm{d}r\,\mathrm{d}\phi$

Figure 4.4: Distribution of quadrature points of the two integral parts.

since the well edge is now represented precisely (see Figure 4.4 with an example of quadrature points distribution).

The adaptive integration in polar coordinates has been implemented and tested. However, the results of are not satisfying as expected. The quadrature accuracy is very dependent on the choice of the band width around the well, position of the well relative to nodes of the mesh and also to the size of nearby elements. When playing long enough with the setting, an optimal convergence rate can be reached, but no general and robust setting has been found. The quadrature points generated in the band also demand more efficient mapping onto elements and a reference element for the shape functions to be evaluated. The problem would need a deeper investigation and we leave it open for now. The previously described quadrature is stable and accurate enough and that is sufficient for our purposes.

4.3 Single Aquifer Analytic Solution

We define a simple problem where analytical solution is available and on which we later test and compare the enrichment methods. To this end, we restrict ourselves to a single aquifer model. In contrast to our article [33], the set of problems with analytical solution has been broaden to include multiple wells and to support arbitrary regular function in the source term.

We rewrite Problem 4.1.1, considering a single aquifer and wells having one end point at the cross-section with the aquifer $\boldsymbol{x}_w$ and the other end point at the surface level denoted $\boldsymbol{x}_D^w$. Let us suppose the parameters of the model $\delta_2, \boldsymbol{K}_2, \boldsymbol{K}_1$ to be constants, and the hydraulic conductivity $\boldsymbol{K}_2 = K_2$, $\boldsymbol{K}_1 = K_1$ to be scalars. Zero source term $f_1 = 0$ is considered in the wells, so the flux there is governed only by the Dirichlet boundary condition at $\boldsymbol{x}_D^w$ and the flux to/from aquifer which forms the Neumann boundary condition at $\boldsymbol{x}_w$. Therefore the flux is constant and the pressure is linear in the wells:

$$-\delta_1^w K_1^w \nabla p_1 = -\frac{\delta_1^w K_1^w}{\boldsymbol{x}_D^w - \boldsymbol{x}_w}\big(g_{1D}^w - p_1(\boldsymbol{x}_w)\big) \qquad \text{in } \Omega_1^w,\ \forall w \in \mathcal{W} \tag{4.53}$$

The denominator $(\boldsymbol{x}_D^w - \boldsymbol{x}_w)$ is in fact the length of the well $|\Omega_1^w|$. The constant $\delta_1^w K_1^w / |\Omega_1^w|$ can be seen as the permeability of the well c_w between aquitards in the original work [33].

Then we have a simplified problem:

Problem 4.3.1. Find $[p_1,\, p_2]$ satisfying

$$\begin{aligned}
-K_2 \Delta p_2 &= f_2 && \text{in } \Omega_2 && (4.54a)\\
-K_2 \langle \nabla p_2 \cdot \boldsymbol{n} \rangle_w &= \sigma_w\big(\langle p_2 \rangle_w - p_1(\boldsymbol{x}_w)\big) && \forall w \in \mathcal{W} && (4.54b)\\
p_2 &= g_{2D} && \text{on } \Gamma_{ext} = \Gamma_{2D}, && (4.54c)\\
p_1(\boldsymbol{x}_D^w) &= g_{1D}^w && \forall w \in \mathcal{W} && (4.54d)\\
-\frac{\delta_1^w K_1^w}{|\Omega_1^w|}\big(g_{1D}^w - p_1(\boldsymbol{x}_w)\big) &= |\Gamma_w|\, \delta_2 \sigma_w \big(\langle p_2 \rangle_w - p_1(\boldsymbol{x}_w)\big) && \forall w \in \mathcal{W} && (4.54e)
\end{aligned}$$

We now look for the analytic solution $[p_1, p_2]$ of Problem 4.3.1. Well pressure is fully determined by finding the value $p_1(\boldsymbol{x}_w)$. Considering multiple singularities in the domain, we can obtain the solution using the superposition principle. The solution is searched in the form

$$p_2 = p_{sin} + p_{reg} = \sum_{j \in \mathcal{W}} a_j \log r_j + p_{reg}, \tag{4.55}$$

where we split p_2 into a singular and a regular part. Function p_{reg} is considered to be a smooth function which can be well approximated by standard finite elements (with optimal convergence rate). Having the pressure in (4.55), we also derive its gradient and average terms

$$\nabla p_2 = \sum_{j \in \mathcal{W}} a_j \frac{\boldsymbol{r}_j}{r_j^2} + \nabla p_{reg}, \tag{4.56}$$

$$\langle p_2 \rangle_i = a_i \log \rho_i + \sum_{\substack{j \in \mathcal{W} \\ j \neq i}} a_j \langle \log r_j \rangle_i + \langle p_{reg} \rangle_i, \tag{4.57}$$

$$\langle \nabla p_2 \cdot \boldsymbol{n}_i \rangle_i = -a_i \frac{1}{\rho_i} + \sum_{\substack{j \in \mathcal{W} \\ j \neq i}} a_j \langle \frac{\boldsymbol{r}_j \cdot \boldsymbol{n}_i}{r_j^2} \rangle_i + \langle \nabla p_{reg} \cdot \boldsymbol{n}_i \rangle_i, \tag{4.58}$$

The singular part p_{sin} solves (4.54a) with zero right hand side since logarithm is a harmonic function. The source f_2 is sought such that the chosen regular part p_{reg} is the solution of (4.54a), i.e.

$$f_2 = -K_2 \Delta p_{reg}. \tag{4.59}$$

To find the unknowns a_w and $p_1(\boldsymbol{x}_w)$, we need to solve a system of linear equations which we obtain by substituting (4.55)-(4.58) into the two conditions (4.54b) and (4.54e). We then assemble the $2W \times 2W$ linear system $\boldsymbol{M}\boldsymbol{a} = \boldsymbol{b}$ where

$$M_{ij} = \delta_{ij} K_2 \langle \frac{\boldsymbol{r}_j \cdot \boldsymbol{n}_i}{r_j^2} \rangle_i - \sigma_i \langle \log r_j \rangle_i \qquad i, j = 1 \dots W, \tag{4.60a}$$

$$M_{ii} = |\Gamma_i| \, \delta_2 \sigma_i + \frac{\delta_1^i K_1^i}{\boldsymbol{x}_D^i - \boldsymbol{x}_i} \qquad i = W+1 \dots 2W, \tag{4.60b}$$

$$M_{ij} = \sigma_i \qquad i = 1 \dots W, \tag{4.60c}$$

$$j = W+1 \dots 2W, \tag{4.60d}$$

$$M_{ij} = - |\Gamma_i| \, \delta_2 \sigma_i \langle \log r_j \rangle \qquad i = W+1 \dots 2W, \tag{4.60e}$$

$$j = 1 \dots W, \tag{4.60f}$$

$$b_i = K_2 \langle \nabla p_{reg} \cdot \boldsymbol{n}_i \rangle_i + \sigma_i \langle p_{reg} \rangle_i \qquad i = 1 \dots W, \tag{4.60g}$$

$$b_i = |\Gamma_i| \, \delta_2 \sigma_i \langle p_{reg} \rangle_i + \frac{\delta_1^i K_1^i}{\boldsymbol{x}_D^i - \boldsymbol{x}_i} g_{1D}^i \qquad i = W+1 \dots 2W. \tag{4.60h}$$

We evaluate all the averages $\langle \cdot \rangle_i$ in the linear system numerically, using simple composite midpoint rule integration with 1000 equidistant intervals on Γ_i. This way we obtain pseudo-analytic solution of the multi well problem, which is accurate enough to be used for measuring the error of our model. The analytical solution (4.55) is then used to define the boundary function $g_{2D} = p_2|_{\Gamma_D}$ in (4.54c).

4.4 Numerical Tests

In this section we suggest several numerical tests and provide comparison of different enrichments as suggested in Section 4.2.1.

The implementation has been done in C++ language using the Deal II [45], version 8.0, an open source finite element library. This library provides a well-documented code for high range of operations one needs for solving partial differential equation: mesh generation, quadratures, different scalar and vector finite elements, linear algebra, h and p adaptivity, error estimators, postprocessing and output to various formats. However, it does not support any enrichment techniques like XFEM in the version 8.0.0, and so it does not in the actual version 9.0.0, released in May, 2018, up to our best knowledge.

We use as much as possible of the relevant library code, although the enrichment functions, well intersections, distribution and handling of the enriched degrees of freedom, adaptive quadrature and some output routines must have been implemented on our own. The source is available on GitHub:

`https://github.com/Paulie14/xfem_project`.

4.4.1 Test Cases with Single Well

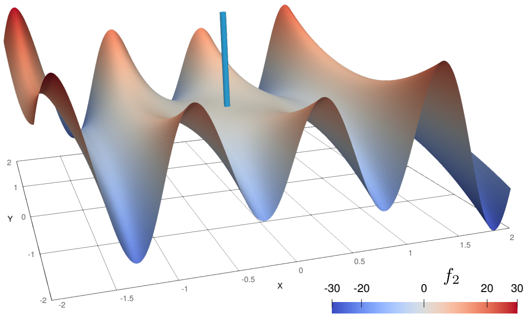
Let us now define four test cases on which we investigate the behavior of the methods, mainly their convergence properties. We suggest four analytic solutions which differ by the regular part p_{reg} and the corresponding source term f_2. Thus the quality of the approximation of both the singular and the regular part can be later inspected to make a conclusion whether it behaves as expected. The list of p_{reg} and f_2 follows:

- case 1: $p_{reg} = 0,\ f_2 = 0$,
- case 2: $p_{reg} = U \sin(\omega x)(r_w - \rho_w)^2$,

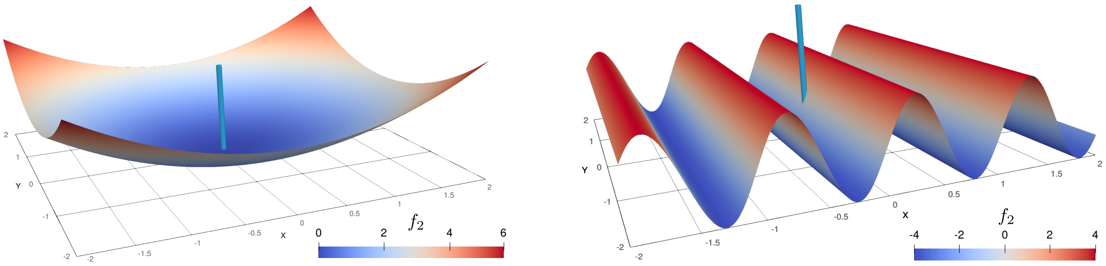
$$f_2 = -K_2 U \Bigg[\left(4 - 2\frac{\rho_w}{r_w}\right) \sin(\omega x) \\ + 4(r_w - \rho_w)\frac{\boldsymbol{r} \cdot \boldsymbol{e}_1}{r_w} \omega \cos(\omega x) - (r_w - \rho_w)^2 \omega^2 \sin(\omega x) \Bigg],$$

- case 3: $p_{reg} = U r_w^2,\ f_2 = -4K_2 U$,
- case 4: $p_{reg} = U \sin(\omega x),\ f_2 = K_2 U \omega^2 \sin(\omega x)$.

The source term f_2 is visualized in Figure 4.5. Test case 1 includes only the singularity and has zero source term, therefore the approximation of the singularity itself is in the focus. In Test case 2, the part of the searched solution p_{reg} has both its value and gradient zero on the edge of the well Γ_w. In Test case 3, p_{reg} has constant value and gradient on Γ_w, and finally p_{reg} is fluctuating on Γ_w in Test case 4. Thus from test cases 1 to 4 we are going from the simplest to the most complex problem in the view of how the approximation is constructed in the vicinity of the well.



(a) test case 2



(b) test case 3

(c) test case 4

Figure 4.5: Source term f_2 visualization.

We now list the input data common to all test cases. The permeability between the aquifer and the well is set to 10^5. The value of the parameter c_w (well permeability), explained in (4.53) is set to 10^{10}, so the pressure in the well is nearly constant. The aquifer domain is a square $\Omega = [-l_\Omega, l_\Omega] \times [-l_\Omega, l_\Omega]$, with l_Ω defined for each test case individually. As we shall see below, the approximation of the singularity by means of the standard FEM shape functions is better when a node of $\mathcal{T}_2$ is inside the well cross-section. Therefore the center of the singularity $\boldsymbol{x}_w$ is placed such that we do not take advantage of this feature and solve the worse scenario. See the common geometry of the test cases in Figure 4.6.

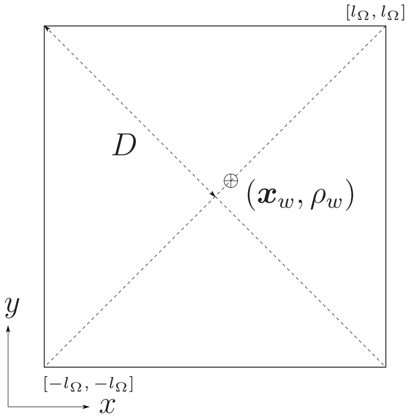


Figure 4.6: Geometry of the aquifer domain common to all the test cases.

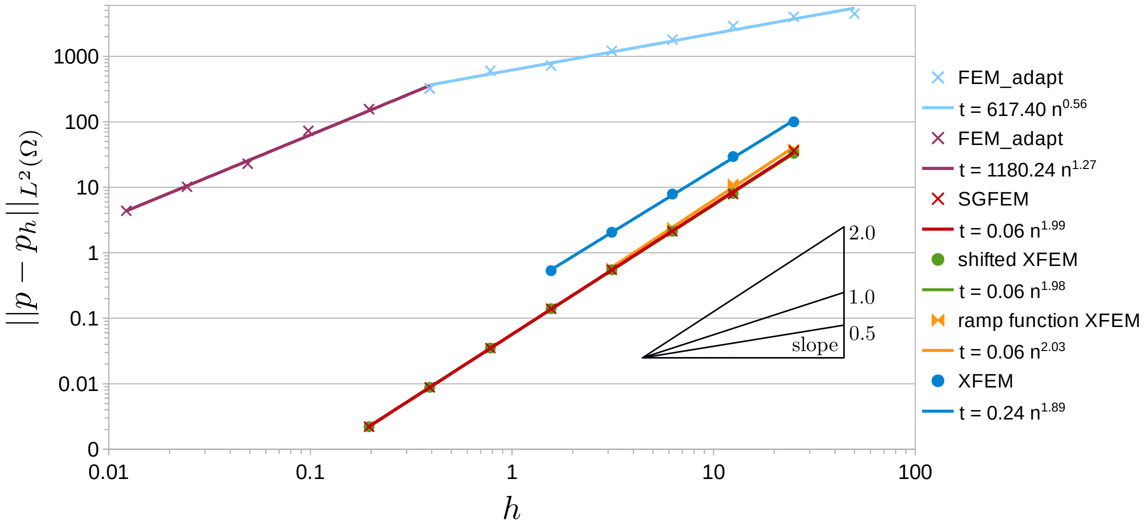
We are interested in the behavior of the enrichment in the 2d domain. Thus we measure the convergence of the methods only inside the aquifer. The error of the solution is measured in L_2 norm, which is evaluated using higher order quadrature on the unenriched elements and adaptive quadrature in the enriched area, as defined in Section 4.2.2.

Test Case 1

Let us start with Test case 1 and present the results achieved in [33]. The parameters are gathered in Table 4.1. Convergence of all used methods is measured on a set of uniformly refined meshes, see the convergence rate in Graph 4.1. At first, we solve the problem by the standard FEM using h adaptivity. Kelly's error estimator from Deal II library is used to determine 30% of the elements with the highest error which are refined in the next step. The element size h for the convergence graph is determined as the size of the smallest element in the vicinity of the well. From the graph of "FEM_adapt" it is apparent that the convergence is slow until the size of the elements reaches the scale of the well cross-section and one of the mesh nodes gets inside. Therefore the graph is divided into two parts with different convergence orders 0.56 and 1.27.

l_Ω	$\boldsymbol{x}_w$	ρ_w	R_w	g^w_{1D}
100	[5.43,5.43]	0.2	50	100

Table 4.1: Input data for Test case 1.



Graph 4.1: Convergence of the L_2 norm of the approximation error in Test case 1.

Next we look at the convergence of the enrichment methods in the XFEM. The standard XFEM pushes the error down by three orders of magnitude. Its convergence rate is nearly optimal with the order closing to 2.0. On the other hand we have no results on finer meshes due to the conditioning of the linear system which deteriorates rapidly for $h < 2$ and our conjugate gradient (CG) solver does not converge. The same problem arises using the ramp function XFEM. It deals better with the error on blending elements but the ill-conditioning of the system matrix still corrupts the computation. We discuss the conditioning of the system a bit more in the next subsection 4.4.2.

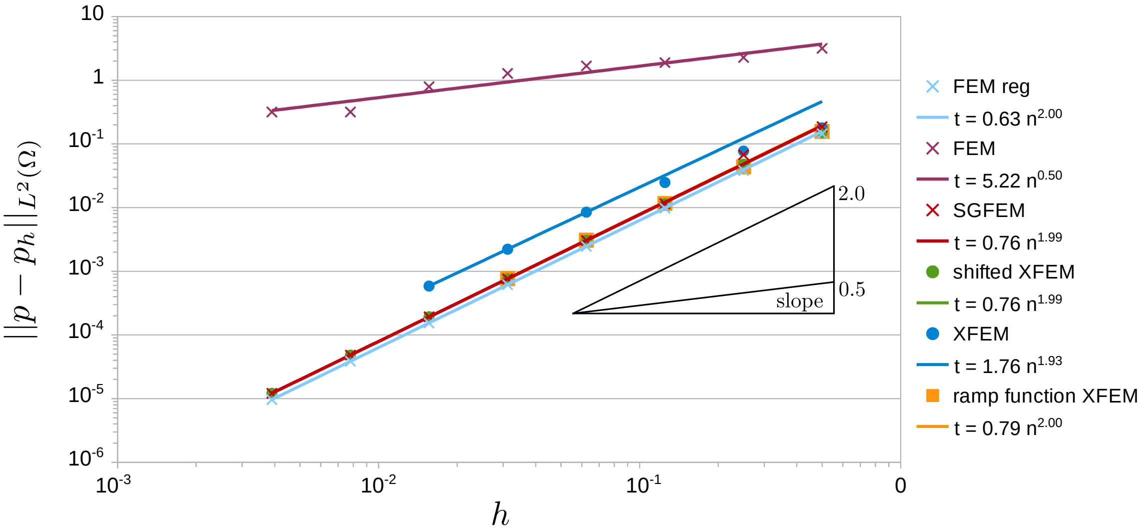
The shifted XFEM and the SGFEM behave nearly the same way and give the best results as expected. We alert the reader that the convergence graphs of ramp function XFEM, shifted XFEM and SGFEM overlap in Graph 4.1. We only mention that the SGFEM saves small amount of degrees of freedom on blending elements in comparison with the shifted and the ramp function XFEM. The order of convergence in L_2 norm closing to 2.0 is optimal.

Test Case 2

Next we present the results in Test case 2, the convergence graph is in Graph 4.2. The parameters are gathered in Table 4.2. We expect the standard FE shape functions to optimally approximate the regular part of the solution and the shape functions of the enrichment to capture the singularity. Therefore we solve the problem on

l_Ω	$\boldsymbol{x}_w$	ρ_w	R_w	g^w_{1D}	ω	U
2	[0.004,0.004]	0.003	0.3	4	6	4

Table 4.2: Input data for Test case 2.



Graph 4.2: Convergence of the L_2 norm of the approximation error in Test case 2. The FEM$_{reg}$ data comes from the problem without the well solved by standard FEM and with optimal convergence order 2.0.

the aquifer domain only with the source term f_2 but omitting the singularity to have a reference solution – an approximation of p_{reg}. We plot this solution labeled as FEM$_{reg}$ in the convergence graph. Since the source term f_2 is a smooth and bounded function, the FEM is supposed to converge optimally with order 2.0 which is confirmed in the graph. The error in XFEM is then expected to be of the same magnitude as the error FEM$_{reg}$.

Using the standard FEM approximation, see the line FEM in Graph 4.2, the convergence order is low 0.5, because the size of the elements is still far from ρ_w and it cannot capture the singularity at all.

The XFEM behave similarly as in the Test case 1. The error of the enrichment methods is very close to the error FEM$_{reg}$, except standard XFEM where the error in the blending elements is significant. The ill-conditioning of the linear system again disallows computing on finer meshes in case of the standard and ramp function XFEM. The shifted XFEM and the SGFEM converge optimally and the error is dominated by the error of the regular part.

Test Case 3

The application of the standard and ramp function XFEM suffers from the same weaknesses, as we have shown above, in both Test case 3 and 4. Due to the fact that these two methods do not perform well in our problems, we do not deal with them any further and we do not present obtained results. The parameters are gathered in Table 4.3.

l_Ω	$\boldsymbol{x}_w$	ρ_w	R_w	g_{1D}^w	U
2	[0.004,0.004]	0.003	0.3	4	0.7

Table 4.3: Input data for Test case 3.

Table 4.4 shows the convergence of the shifted XFEM and SGFEM for Test case 3. In the column FEM$_{reg}$ the optimal convergence with the order 2.0 of the standard FEM on the regular problem is displayed as a reference. We see that both shifted XFEM and SGFEM converge also with the optimal convergence order 2.0. The error of both enrichment methods is a bit higher than the error of FEM$_{reg}$, we would further diminish this difference by enlarging the enrichment radius. Looking at the graph in Figure 4.7 where the error distribution is plotted, we see that the error is concentrated outside the enrichment radius circle. It is caused due to the insufficient approximation of the singularity by the standard FE shape functions. We also see from this figure that the approximation quality of the quadrilateral FEs on structured mesh is direction dependent.

		FEM$_{reg}$		shifted XFEM		SGFEM	
i	h	$\|\|p - p_h\|\|_{L_2(\Omega_2)}$	order	$\|\|p - p_h\|\|_{L_2(\Omega_2)}$	order	$\|\|p - p_h\|\|_{L_2(\Omega_2)}$	order
0	0.5000	2.45e-01	–	2.73e-01	–	2.35e-01	–
1	0.2500	6.12e-02	2.17	7.89e-02	1.79	6.70e-02	1.81
2	0.1250	1.53e-02	1.95	1.60e-02	2.30	1.60e-02	2.07
3	0.0625	3.82e-03	1.99	4.08e-03	1.97	4.08e-03	1.97
4	0.0312	9.56e-04	2.00	1.02e-03	2.00	1.02e-03	2.00
5	0.0156	2.39e-04	2.00	2.58e-04	1.98	2.58e-04	1.98
6	0.0078	5.97e-05	2.00	6.44e-05	2.00	6.44e-05	2.00
7	0.0039	1.49e-05	2.00	1.61e-05	2.00	1.61e-05	2.00

Table 4.4: Convergence table of the shifted XFEM and SGFEM in Test case 3.

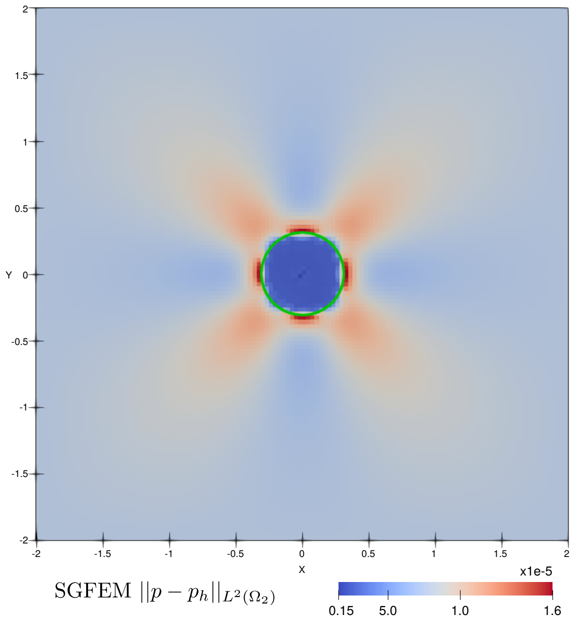


Figure 4.7: L_2 error distribution in Test case 3 at the refinement level level 4 ($h = 0.0312$). A green circle represents the enrichment radius R_w. There is no visible difference between shifted XFEM and SGFEM, so only SGFEM is plotted.

Test Case 4

The last case of our test series demonstrates the necessity of the averaged terms in the model formulation. The parameters are gathered in Table 4.5.

The convergence of the considered methods is shown in Table 4.6, in the same manner as in Test case 3. We see that the convergence order of shifted XFEM and SGFEM is optimal up to the refinement level 6 and the error corresponds to the

l_Ω	$\boldsymbol{x}_w$	ρ_w	R_w	g_{1D}^w	ω	U
2	[0.004,0.004]	0.003	0.3	4	6	4

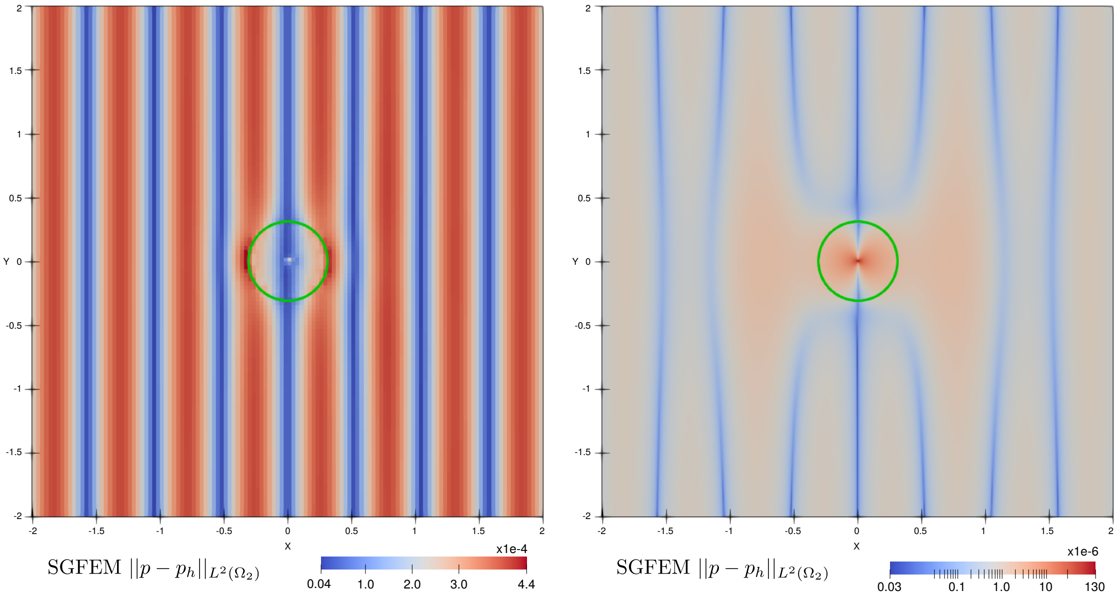
Table 4.5: Input data for Test case 4.

error of the regular problem in FEM$_{reg}$ column.

		FEM$_{reg}$		shifted XFEM		SGFEM	
i	h	$\lVert p - p_h \rVert_{L_2(\Omega_2)}$	order	$\lVert p - p_h \rVert_{L_2(\Omega_2)}$	order	$\lVert p - p_h \rVert_{L_2(\Omega_2)}$	order
0	0.5000	1.01e+01	–	9.30e+00	–	9.92e+00	–
1	0.2500	2.25e+00	2.17	2.21e+00	2.07	2.24e+00	2.15
2	0.1250	5.84e-01	1.95	5.75e-01	1.94	5.80e-01	1.95
3	0.0625	1.47e-01	1.99	1.46e-01	1.98	1.47e-01	1.98
4	0.0312	3.70e-02	2.00	3.66e-02	2.00	3.67e-02	2.00
5	0.0156	9.24e-03	2.00	9.15e-03	2.00	9.17e-03	2.00
6	0.0078	2.31e-03	2.00	2.41e-03	1.93	2.39e-03	1.94
7	0.0039	5.78e-04	2.00	1.05e-03	1.20	1.01e-03	1.24

Table 4.6: Convergence table of the shifted XFEM and SGFEM in Test case 4.

Then the error of the enrichment methods does not decrease as expected at the refinement level 7 and partially 6. To explain this behavior we plot the error distribution at two different refinement levels in Figure 4.8. On the left side, the error in the regular part is dominating. On the right side, the error of the singular part dominates in the center and it spreads in x direction. Closer look at the solution on the well edge shows that the pressure on the well edge is incorrect. The problem has been solved several times for different ρ_w (smaller and larger) and this error always shows up when there is an element of the mesh with all its nodes inside the well-aquifer intersection. We have a suspicion that the elimination of the degrees of freedom of such elements from the linear system is done incorrectly in our implementation. On the finer meshes where ρ_w is larger than the element size, neglecting the fluctuation term on the well edge (4.28) might actually increase the error. Since our model is not describing the pressure in the aquifer inside the well-aquifer intersection (the pressure is constant and equal $p_1(\boldsymbol{x}_w)$ according to the reduced 1d model of the well), and we do not aim in reality to compute on such fine meshes, we do not pursue solving this error at the moment.



(a) refinement level 4

(b) refinement level 7

Figure 4.8: L_2 error distribution in Test case 4 at the refinement level level 4 and 7. A green circle represents the enrichment radius R_w. Pay attention to the logarithmic scale in the right figure.

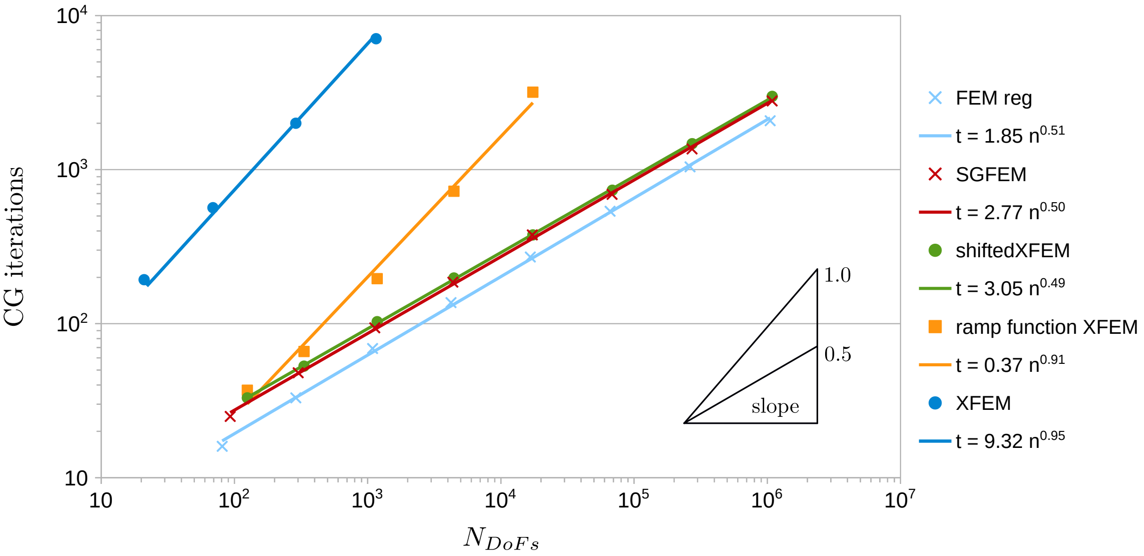
Test Cases Summary

We solved four single well-aquifer problems with different solutions on the well-aquifer intersection and different source terms. We showed that the enrichment methods can sufficiently approximate the singularity and that the L_2 error of the approximation can be pushed as low as the error of the regular part of the solution. The optimal convergence order 2.0 has been reached for all the enrichment methods. The standard XFEM displayed significant error on the blending elements in contrast to the ramp function XFEM. However, both suffered from ill-conditioning of the linear system. The most promising results were obtained using the shifted XFEM and SGFEM, which behaved very similarly in our test cases.

Regarding the order convergence 1.8 presented by Gracie and Craig [2], we obtained similar convergence order around 1.7-1.8 in our experiments using the original adaptive quadrature. Although, the order could be lower depending on the position of the well to the nodes of the mesh. We do not experience this behavior with our adaptive quadrature and the convergence order is always close to the optimum of 2.0.

4.4.2 Conditioning of System Matrix

A problem with ill-conditioning of the linear system coming from the XFEM methods is mentioned in Section 3.1 and are encountered also in our test cases in the section above. We do not inspect the matrices of the linear system in details, but we use some general results on the conjugate gradients method and the Laplace equation to have an insight on this problem.



Graph 4.3: Graph of dependence of the CG iteration count on the number of degrees of freedom. Measured on both problems with no serious distinction observed.

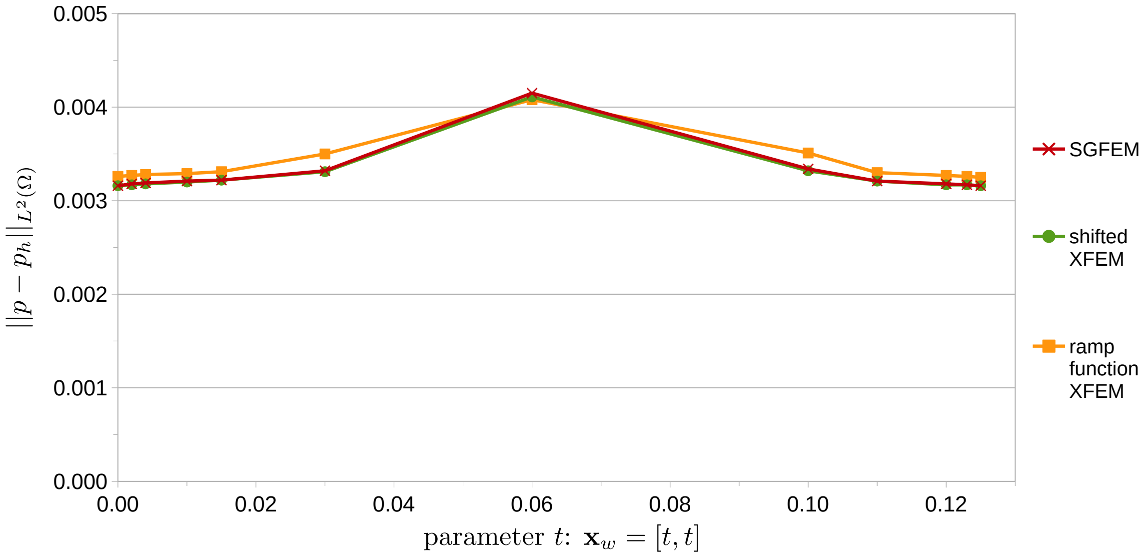
The condition number for matrices resulting from a conforming FEM applied to Laplace equation is $O(h^{-2})$, so the iteration count for the CG without preconditioning is $O(h^{-1}) = O(\sqrt{n})$, where $n = 1/h^2$ is number of degrees of freedom in case of linear finite elements in 2d. With local preconditioning (Jacobi, SOR, ILU) one can usually achieve the number of iterations $O(h^{-0.5})$, cf. [46].

Let us use the data from Test case 3 and look at the iteration count needed by the CG solver in Graph 4.3. The number of iterations of the standard FEM is corresponding to the classic results as mentioned in the paragraph above.

We can see clearly the enormous growth of the number of iterations in case of the standard XFEM and the ramp function XFEM. These problems are generally known and are described for example in the overview of the XFEM in [5]. The usage of enrichment functions can make the approximation space almost linearly dependent from which the ill-conditioning of the system arises.

On the other hand, the SGFEM is proven in [6] to overcome this. They state in the conclusion that the conditioning of the SGFEM system is not worse than that of the standard FEM system. This is exactly what we see in Graph 4.3, where the trends of the SGFEM and the FEM are nearly the same. The behavior of the shifted XFEM is similarly good. We explain this by the fact that the difference between the enrichment functions in shifted XFEM and SGFEM in this particular settings is not that significant.

Another test is considered based on Test case 3, in which the source term is set zero. The model is solved for different positions of the singularity $\boldsymbol{x}_w$ and the effects on the solution and the linear system is observed. The mesh is structured and fixed with $h = 0.125$, centered at point $[0, 0]$ which is an element node. The position of



Graph 4.4: Graph of dependence of the approximation error on the position of the singularity respective to a mesh node.

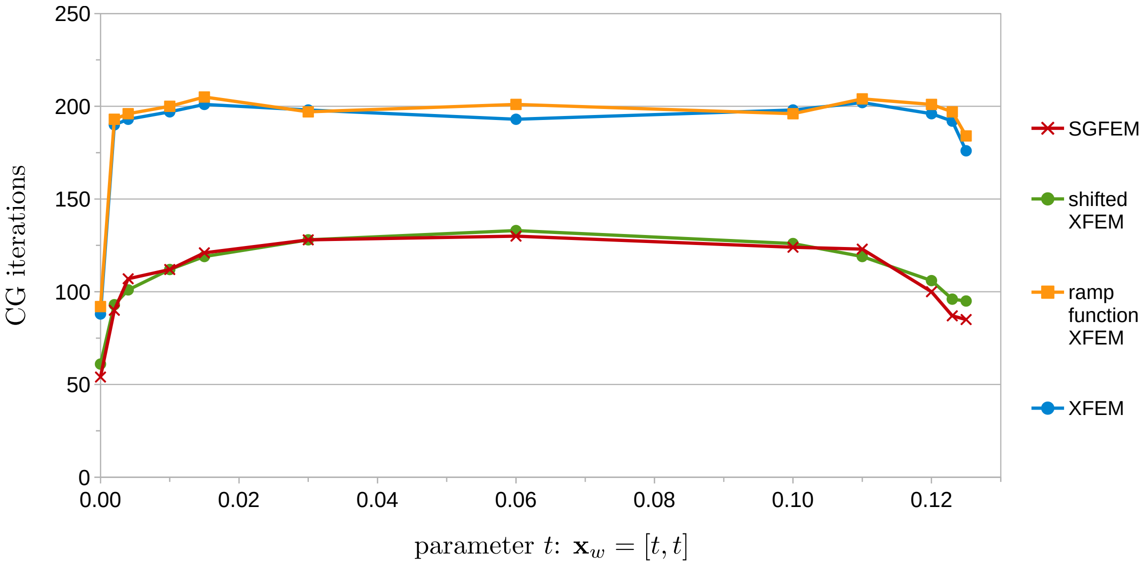
the well (the singularity center) is parameterized by $\boldsymbol{x}_w = [t, t]$ with

$$t \in \{0,\ 0.002,\ 0.004,\ 0.01,\ 0.015,\ 0.03,\ 0.06,\ 0.1,\ 0.11,\ 0.12,\ 0.123,\ 0.125\},$$

so the well moves from one node of the element to the opposite one along the element diagonal. We see the results in two graphs 4.4 and 4.5. The first one displays the approximation error dependence on the relative position of the singularity and the element. The trend is apparent, however the magnitude of the change is not significant. The error of the standard XFEM exhibits the same behavior but it is by one order of magnitude higher, so it does not fit in the figure.

The second figure shows the change in the number of iterations of the CG solver. The trend is again apparent: the CG deals better with the linear system when the singularity closes to an element node, the situation is worse with the singularity in the middle. We do not understand this behavior completely, although we provide the following explanation. The approximation of a singularity by standard FE shape functions is much better when the singularity center is close to a mesh node (we observed this in Test case 1) due to the fact that the magnitude of the singularity is captured by the DoF corresponding to the node. Therefore when moving the singularity center away from the mesh node, the contribution of the enrichment to the system matrix is larger (also observed in the experiments). Since the conditioning (and the number of CG iterations) is sensitive to the enrichment part of the matrix, we observe this trend in Graph 4.5.

We are satisfied with the results of the shifted XFEM and SGFEM, so we do not investigate the properties of the linear system any further at this stage. This area of the problem would need deeper investigation, including a search for a proper preconditioner of the system matrix and looking for an alternative iteration scheme to solve these specific types of linear systems.



Graph 4.5: Graph of dependence of the CG iteration count on the position of the singularity respective to a mesh node.

4.4.3 Enrichment Radius Estimation

Up to now, we set the enrichment radius R_w in the experiments to a fixed value, without providing any explanation of the particular choice. In this section, we study the dependence of the solution error on the enrichment radius R_w. We look for a clue for a particular choice of R_w that is optimal in a specific sense. The first part of this section is devoted to a theoretical analysis, the later part presents a numerical validation of obtained estimates.

Derivation of Theoretical Estimate

Let us consider a general elliptic problem to find $p \in V$ satisfying

$$a(p,q) = l(q), \text{ for } q \in V,$$

where a is bounded elliptic bilinear form: $\|a\| \leq \alpha_2$, $a(q,q) \geq \alpha_1 \|q\|_V^2$, $\alpha_1 > 0$, and l a bounded linear form, $l \in V'$. Suppose, that the problem is to be solved on a domain $\Omega \subset \mathbb{R}^2$ with a single well of radius ρ_w at the origin. Let us assume, that the solution can be split into the singular part $p_{sin}(\boldsymbol{x}) = \log|\boldsymbol{x}|$ and the regular part $p_{reg} = p - p_{sin}$. Let V_h^{reg} be a polynomial finite element subspace of V on a regular mesh of elements with a maximal side length h and let $V_h = V_h^{reg} + V_h^{enr}$ be a such enriched space that p_{sin} can be approximated exactly on the enriched domain Z_w, i.e.

$$\inf_{q \in V_h} \|p_{sin} - q\|_V = \inf_{q \in V_h^{reg}} \|p_{sin}|_{Z'_w} - q\|_V, \quad Z'_w = \Omega \setminus Z_w. \tag{4.61}$$

Using standard error estimate for elliptic PDE (e.g. [47, Theorem 13.1]), we get

$$\|p - p_h\|_V \leq c_a \inf_{q \in V_h} \|p - q\|_V \leq c_a \left(\inf_{q \in V_h^{reg}} \|p_{reg} - q\|_V + \inf_{q \in V_h} \|p_{sin} - q\|_V \right) \tag{4.62}$$

where $c_a = 1 + \alpha_2/\alpha_1$. In the following, we consider $V = H^1(\Omega)$, a square grid and V_h^{reg} formed by bilinear finite elements. Then (4.62) can be further estimated using approximation property of V_h^{reg}:

$$\left\|p - p_h\right\|_{H^1(\Omega)} \leq c_a\left(ch\left|p_{reg}\right|_{H^2(\Omega)} + \left\|p_{sin} - \pi_h p_{sin}\right\|_{H^1(Z'_w)}\right) \tag{4.63}$$

where $\pi_h p_{sin}$ denotes interpolation of p_{sin} in V_h^{reg}. Our next aim is to find tight estimate for the second term. To this end, we calculate H^1 error on a single square element $S_{h,r}$ with side h and distance r from origin. Using parametrization $0 < s, t < 1$, we get

$$\begin{aligned}(p_{sin} - \pi_h p_{sin})(s,t) = \log\sqrt{(r+hs)^2 + (ht)^2} - \Big[&(1-s)(1-t)\log r \\ &+ (1-s)t\log\sqrt{r^2+h^2} + s(1-t)\log(r+h) \\ &+ st\log\sqrt{(r+h)^2+h^2}\Big] \\ = \frac{1}{2}\frac{h^2}{r^2}&\left(t^2 - t - s^2 + s\right) + O\left(\frac{h^3}{r^3}\right)\end{aligned}$$

and

$$\nabla(p_{sin} - \pi_h p_{sin})(s,t) = \frac{h}{r^2}\left(\frac{1}{2} - s, t - \frac{1}{2}\right) + O\left(\frac{h^2}{r^2}\right). \tag{4.64}$$

Assuming $h < r$, we can neglect higher order terms. Then, we obtain by direct integration

$$\left\|p_{sin} - \pi_h p_{sin}\right\|^2_{L_2(S_{h,r})} \approx \frac{1}{4}\frac{h^6}{r^4}\int_0^1\int_0^1 \left(t^2 - t - s^2 + s\right)^2 \,\mathrm{d}s\,\mathrm{d}t = \frac{1}{360}\frac{h^6}{r^4}$$

and

$$\left\|\nabla(p_{sin} - \pi_h p_{sin})\right\|^2_{L_2(S_{h,r})} \approx \frac{2h^4}{r^4}\int_0^1\left(t - \frac{1}{2}\right)^2 \mathrm{d}t = \frac{1}{6}\frac{h^4}{r^4}. \tag{4.65}$$

Thus for the density of squared error we have

$$\frac{1}{|S_{h,r}|}\left\|p_{sin} - \pi_h p_{sin}\right\|^2_{H^1(S_{h,r})} \approx \frac{h^2}{6r^4}$$

which after integration over the unenriched domain gives final estimate

$$\left\|p_{sin} - \pi_h p_{sin}\right\|_{H^1(Z'_w)} \leq \left[\int_0^{2\pi}\int_{R_w}^{\infty}\frac{h^2}{6r^4}r\,\mathrm{d}r\,\mathrm{d}\theta\right]^{1/2} = \sqrt{\frac{\pi}{6}}\frac{h}{R_w}. \tag{4.66}$$

Recalling the estimate (4.62), we can conclude that optimal choice of the enrichment radius is $h/R_w \approx \left\|p_{reg} - \pi_h p_{reg}\right\|_{H^1(\Omega)}$, which balances the error in the regular and the singular part. Larger R_w would not benefit to better overall approximation error and it would lead unnecessarily to higher amount of enriched DoFs. In combination with an a posteriori error analysis, this could give a rule for an automatic determination of the enrichment radius.

Numeric Validation

Here we provide numerical validation of the results above. Our aim is twofold: we first validate the estimate (4.65), secondly we simulate the dependence of the error in L_2 norm on the enrichment radius numerically and compare it with (4.66).

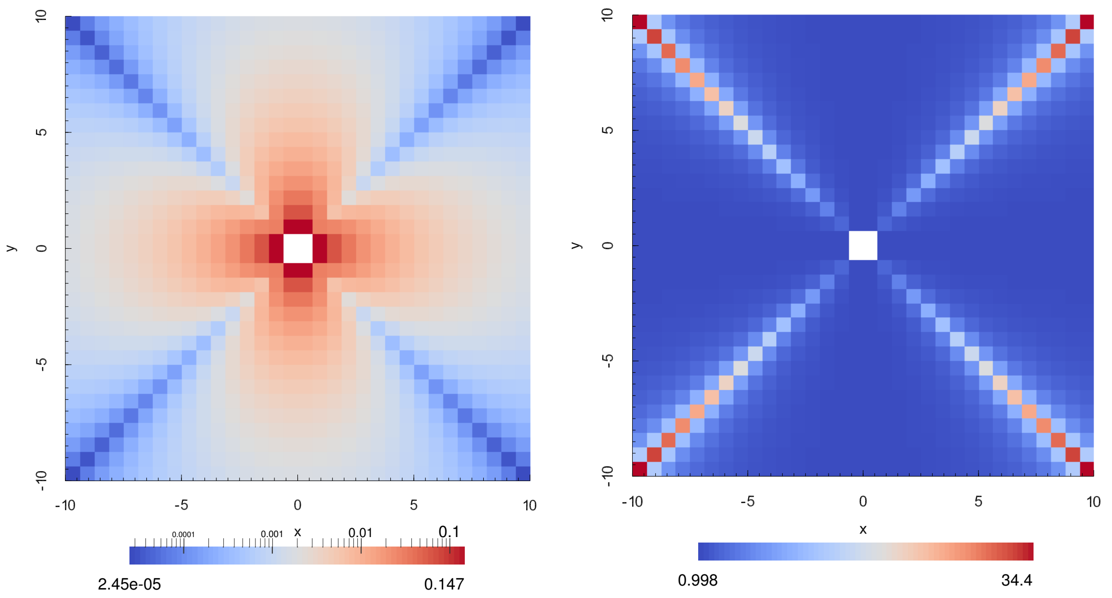
Validity of the estimate (4.65) is verified by calculating the ratio

$$\frac{h^{3/2}r^{-2}12^{-1/2}}{\|p_{sin}-\pi_h p_{sin}\|^2_{H^1(T)}}, \qquad p_{sin}(\boldsymbol{x}) = \log|\boldsymbol{x}| \tag{4.67}$$

on every element T of the sequence of refined meshes using a 5×5 Gaussian quadrature for the estimation of the H^1 norm.

h	min	max
$^{10}/_{8}$	0.97	7.1
$^{10}/_{16}$	0.99	16.4
$^{10}/_{32}$	1.00	34.4
$^{10}/_{64}$	1.00	70.3
$^{10}/_{128}$	1.00	142.0

Table 4.7: Minimal and maximal values of the ratio (4.67) for sequence of refined meshes with element size h. Validation of the estimate (4.65).



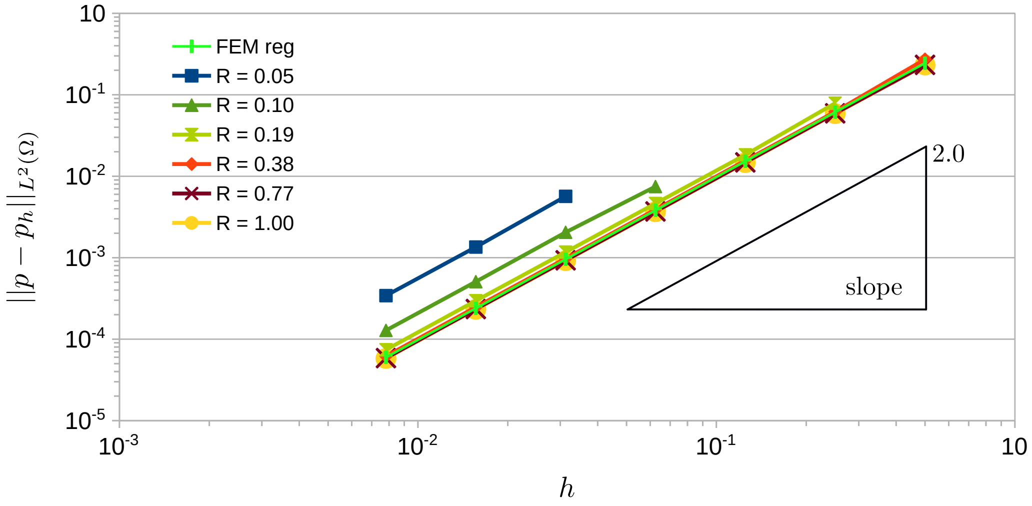
(a) $\|\log\boldsymbol{x}-p_h\|^2_{H^1(T)}$ in log scale

(b) the ratio (4.67)

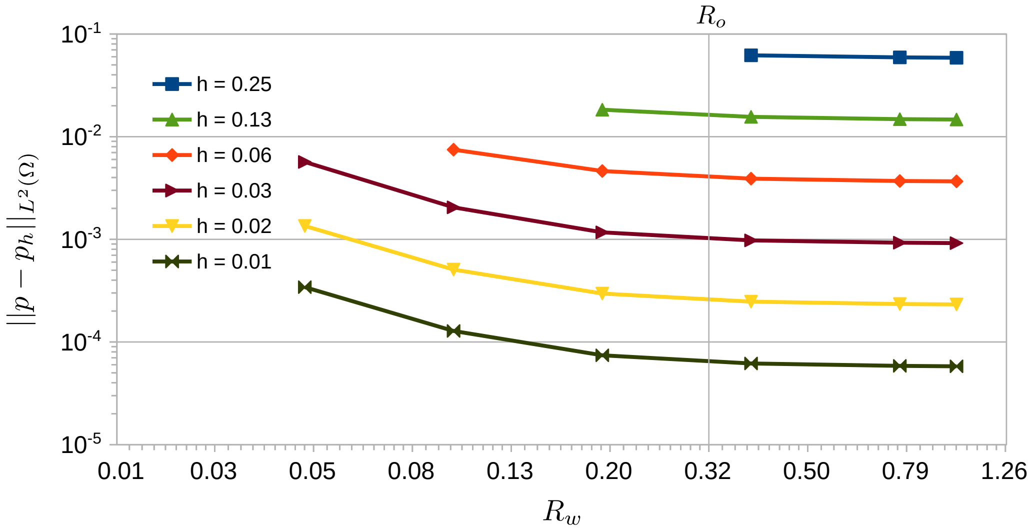
Figure 4.9: Numerical validation of the estimate (4.65). The elements are left out in the center where the log singularity is situated and where the function is cut off.

Table 4.7 reports the minimum and the maximum values of the ratio over all elements of every mesh. The minimum values are close to 1 independently of h

which is in perfect agreement with (4.65). Moreover, the minimum value is attained on the majority of elements, see Figure 4.9b. Both parts of Figure 4.9 demonstrate also higher convergence rate on diagonal elements where the nonlinear term of the bilinear finite elements allows better approximation of the saddle shaped logarithmic surface.



Graph 4.6: Convergence graph for different enrichment radii. The "FEM reg" data comes from the problem without the singularity solved by the standard FEM – it has the optimal convergence order 2.0.



Graph 4.7: Dependence of the error on the enrichment radius for different element sizes h.

Next, we study the influence of the enrichment radius R_w on the global L_2 error. To this end, we solve Test case 3 using the SGFEM for different mesh steps and different values of R_w. Let us remind that $O(h^p)$ convergence of the solution in the H^1 norm translates to the $O(h^{p+1})$ convergence of the solution in the L_2 norm for the linear elliptic problems (cf. [47, Theorem 19.2]). According to the estimates (4.62) and (4.66), we expect $O(h^2)$ convergence of L_2 norm independently of the enrichment radius. This is clearly demonstrated in Graph 4.6. For comparison, we plot also the error of the regular part p_{reg} of the solution solved by standard FEM showing the $O(h^2)$ convergence. As predicted, the total error diminishes with R_w but cannot drop under the error of p_{reg}. We approximate $\|p_{reg} - \pi_h p_{reg}\|_{H^1(\Omega)}$ using a fine mesh and then according to (4.66), we get the optimal value of the enrichment radius

$$R_o \sim \sqrt{\frac{\pi}{6}} h / \|p_{reg} - \pi_h p_{reg}\|_{H^1(\Omega)} \sim 0.32$$

This value roughly matches a point in the plots of the error as a function of R_w in Graph 4.7, from which the error is not decreasing any more.

4.4.4 Test Cases with Multiple Wells

In this section, more complex test cases are solved, including more than one well intersecting the aquifer. We test the enrichment methods, shifted XFEM and SGFEM in particular, whether they still have such good convergence properties when the wells influence each other and when the enrichment zones coincide with each other.

Test Case 5

At first we consider two wells intersecting a square aquifer $\Omega_2 = [0, 10] \times [0, 10]$ at points $\boldsymbol{x}_1 = [4.1, 4.3]$ and $\boldsymbol{x}_2 = [5.7, 5.9]$. The pressure at the top of the wells is set $g^1_{1D} = 150$ and $g^2_{1D} = 100$ respectively. Remaining parameters are set the same for both wells: $\rho_w = 0.003$, $\sigma_w = 100$, $c_w = 10^{10}$ for $w = 1, 2$. The hydraulic conductivity of the aquifer is set $K_2 = 10^{-3}$ and the aquifers thickness is set $\delta_2 = 1$. The source term is sinusoidal as in Test case 4, i.e. $f_2 = K_2 U \omega^2 \sin(\omega x)$, with parameters $\omega = 1$, $U = 80$. The solution is shown in Figure 4.10.

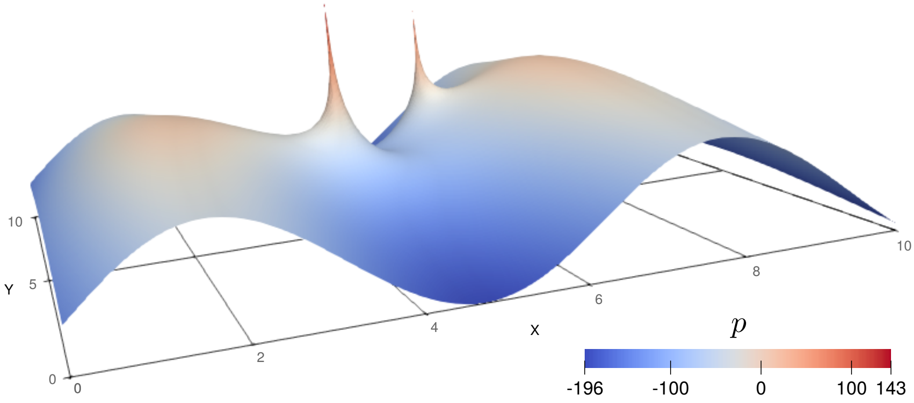
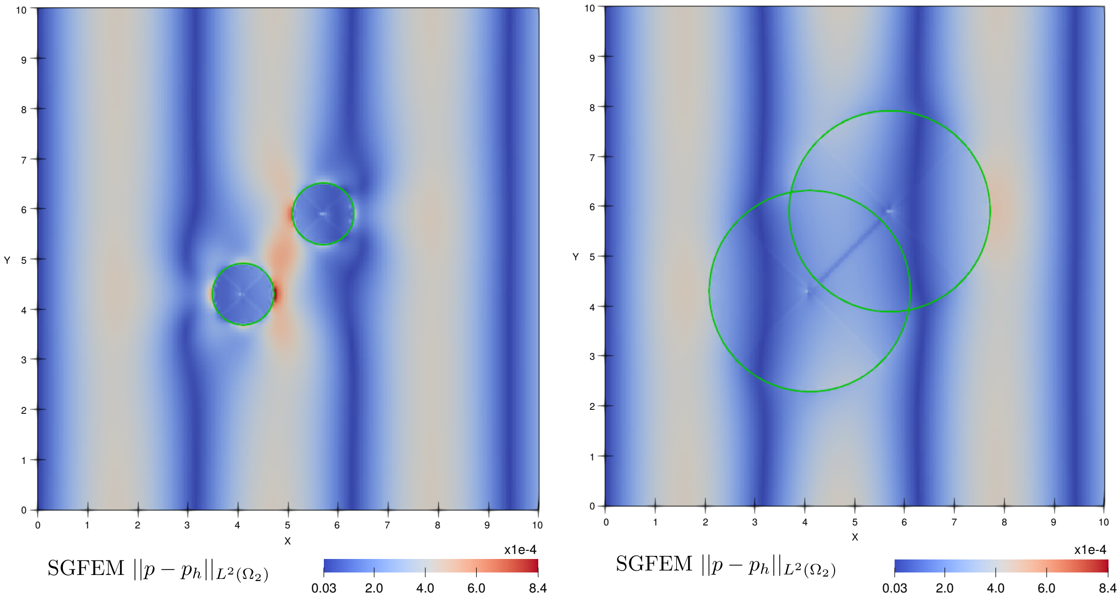


Figure 4.10: Solution of Test case 5.

At first we solve the problem using the enrichment radius $R_w = 0.6$ for both wells. The results are as expected: optimal convergence order is observed and



(a) enrichment radius $R_w = 0.6$ (b) enrichment radius $R_w = 2.0$

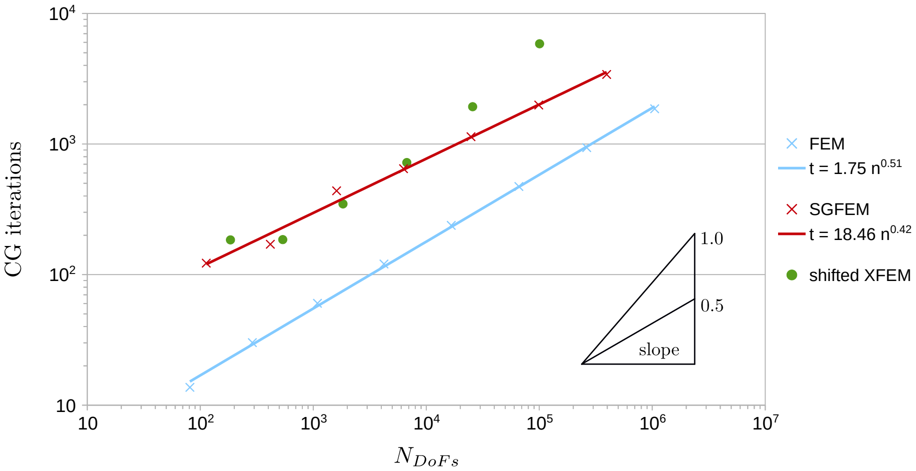
Figure 4.11: L_2 error distribution in Test case 5 for two different R_w, represented by green circles.

CG iterations count increases reasonably as in Section 4.4.2. Next we enlarge the enrichment radius $R_w = 2.0$ such that the two circular enrichment zones overlap and the elements are enriched from both singularities there. We plot the error distribution in Figure 4.11, comparing the difference between $R_w = 0.6$ on the left and $R_w = 2.0$ on the right. We see a higher approximation error accumulating outside the smaller enrichment zones, in case of larger enrichment radii only the error of the regular part is apparent.

		shifted XFEM		SGFEM	
i	h	$\|\|p - p_h\|\|_{L_2(\Omega_2)}$	order	$\|\|p - p_h\|\|_{L_2(\Omega_2)}$	order
0	0.5000	1.01e+02	–	1.06e+02	–
1	0.2500	1.66e+01	2.61	1.78e+01	2.57
2	0.1250	4.22e+00	1.97	4.45e+00	2.00
3	0.0625	1.07e+00	1.99	1.11e+00	2.00
4	0.0312	2.67e-01	1.99	2.78e-01	2.00
5	0.0156	6.72e-02	1.99	6.93e-02	2.00
6	0.0078	–	–	1.72e-02	2.01

Table 4.8: Convergence table of the shifted XFEM and SGFEM in Test case 5, using enrichment radius $R_w = 2.0$.

Table 4.8 displays the optimal convergence order of both shifted XFEM and SGFEM for $R_w = 2.0$. The computation on the most refined mesh is failing in case of the shifted XFEM due to the non-converging CG solver. The number of CG iterations for both enrichment methods and standard FEM is plotted in Graph 4.8. The SGFEM performs as expected, on the other hand the CG iterations for shifted XFEM increase rapidly. This is due to the enrichment of elements by multiple enrichment functions causing worse conditioning of the resulting linear system.



Graph 4.8: Graph of dependence of the CG iteration count on the number of degrees of freedom. Measured on both problems with no serious distinction observed.

Test Case 6

In this test case, five wells are intersecting the aquifer $\Omega_2 = [0, 10] \times [0, 10]$. The well specific data are gathered in Table 4.9. The wells 3 and 5 can be seen as injection wells with a positive flux to the aquifer, the others as pumping wells with a positive flux from the aquifer. The common well parameters are $\rho_w = 0.003$, $c_w = 10^7$. The hydraulic conductivity of the aquifer is set $K_2 = 10^{-3}$ and the aquifers thickness is set $\delta_2 = 1$. The source term is similar to the one in Test case 5, using parameters $\omega = 1$, $U = -8$. The solution is shown in Figure 4.12.

well w	1	2	3	4	5
$\boldsymbol{x}_w$	[2.8,2.5]	[4.9,5.4]	[2.9,7.4]	[7.3,7.8]	[7.4, 2.8]
σ_w	20	10	10	10	20
g_{1D}^w	-150	-30	120	-50	100

Table 4.9: Input data for the wells in Test case 6.

We show the convergence results of the SGFEM in Table 4.10. The problem is solved with two different enrichment radii R_w, the smaller one assures non-overlapping enrichment zones. The convergence order is still satisfying, although in case of the larger R_w the error did not decreased enough at the last refinement level for an unknown reason.

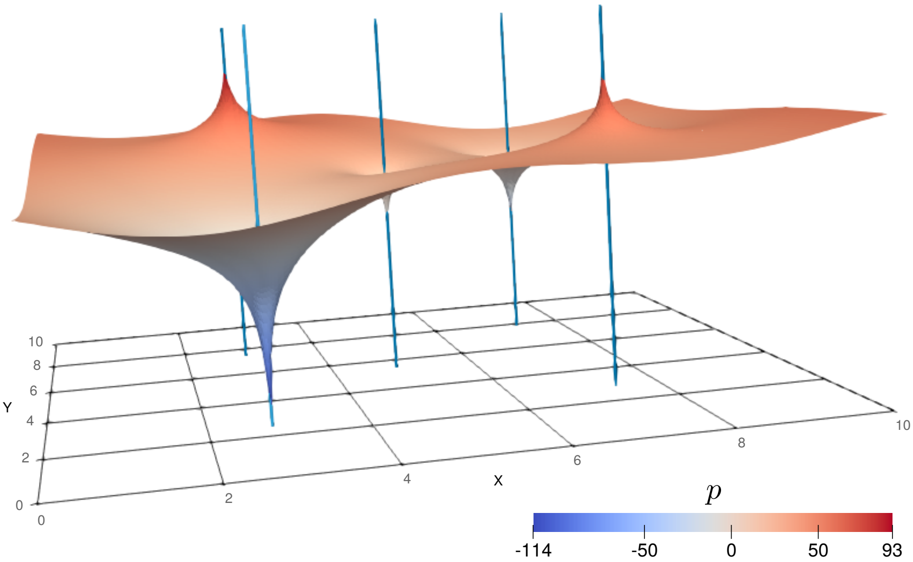


Figure 4.12: Solution of Test case 6.

		$R_w = 0.8$		$R_w = 2.0$	
i	h	$\|p - p_h\|_{L_2(\Omega_2)}$	order	$\|p - p_h\|_{L_2(\Omega_2)}$	order
0	0.5000	5.09e+01	–	5.08e+01	–
1	0.2500	2.30e+00	4.46	1.66e+00	4.93
2	0.1250	5.66e-01	2.03	4.05e-01	2.04
3	0.0625	1.44e-01	1.98	1.03e-01	1.98
4	0.0312	3.61e-02	1.99	2.47e-02	2.06
5	0.0156	9.22e-03	1.97	6.36e-03	1.96
6	0.0078	2.41e-03	1.94	1.77e-03	1.85

Table 4.10: Convergence table of the SGFEM in Test case 6, using two different enrichment radii.

We do not mention the results of shifted XFEM explicitly since it performs nearly the same as SGFEM with the smaller enrichment radius. In case of overlapping enrichment zones, the shifted XFEM suffers with ill-conditioning as in the previous test case. On the other hand, the iteration count for the SGFEM increases still with the same rate independently of the chosen R_w.

4.5 Summary

This Chapter was devoted to the investigation of four different enrichment methods – the XFEM, the Corrected XFEM with the ramp function and shifting, and the SGFEM. The solved problem setting was inspired by the work [2, 3] of R. Gracie and J.R. Craig but our effort was aimed more in understanding the details of the enrichment methods and their comparison rather than in computation of complex problems. Resolving a point singularity in a 2d domain was addressed.

The well-aquifer model was defined in the beginning and its weak form was derived, including the analysis of the weak solution existence. In addition to our article [33], the model was formulated correctly using the averaged term on the well-aquifer cross-section. Then the discretization by means of the XFEM was thoroughly described. The range of the numerical tests was narrowed to problems with a single aquifer for which the pseudo-analytic solution was derived.

All the implemented methods were compared in 4.4.1. Regarding the XFEM, we saw that at least the ramp function must be used in order to optimize the error on the blending elements. The shifted XFEM and the SGFEM converged optimally and did not show any difference in the solution precision. For the last two methods we also showed that the precision is not dependent on the relative position of a singularity to a node.

The issue of a suboptimal convergence order reported in [2] was investigated. We revealed the problem and we suggested a better strategy for the adaptive quadrature. The improvement was confirmed by the numerical tests in 4.4 where we obtained the optimal order of convergence in L_2 norm. An alternative quadrature for integration of enriched functions in polar coordinates was also suggested.

A proper size of the enrichment zone, defined by the enrichment radius, was studied. The error estimate dependent on the enrichment radius was derived and it was numerically validated. Furthermore, the optimal enrichment radius was predicted for the test problem and it corresponded with the computed data.

The ill-conditioning of the system matrix was observed through the increasing iteration count of conjugate gradients solver. The standard XFEM and the ramp function XFEM suffered from ill-conditioning even in the simplest test cases. The shifted XFEM performed well when there was only one singular enrichment per element. Otherwise, when the enrichment zones from multiple singularities were overlapping, only the SGFEM behaved well in terms of system matrix conditioning.

Regarding the three conditions (a)-(c) in Section 3.2.2 for an XFEM to be called the SGFEM, we conclude that our implementation of the SGFEM indeed appears to have these properties, although we do not provide a theoretical proof. The SGFEM yields the optimal convergence rate and its conditioning is not worse than that of the FEM in all presented test cases, both the approximation error and the conditioning is robust with respect to the relative position of the singularity to the mesh nodes. The local orthogonalization technique has not been implemented, but it might be beneficial in case of overlapping enrichment zones. Particularly in Test case 6, it is worth of further experimenting to resolve the slightly suboptimal error decrease at the last refinement level.

5 Mixed Model with Singularities

In this chapter we derive a mixed-hybrid model for velocity and pressure and we develop a mathematical background. The model formulation is built from a simple 2d Dirichlet problem to a fully coupled 1d-2d and 1d-3d problem. We introduce an SGFEM like enrichment for the velocity approximation. At the end we provide results of several numerical experiments to demonstrate the properties and usability of our model, implemented in the software Flow123d.

5.1 Mixed Problems in Groundwater Flow

The theory of mixed and mixed-hybrid finite element method is described in general in the well known book by Brezzi and Fortin [48]. The formulation, its stability and error estimates are studied there.

The actual application of the mixed-hybrid method in the model of flow in fractured porous media with combined mesh dimensions is studied in several articles. In [8] the weak formulation and its discretization using the lowest-order Raviart-Thomas finite elements is written. The linear system is then reduced using the Schur complement (original idea in [49]) and solved efficiently with preconditioned conjugate gradients method. In [9] the theoretic part includes the weak and the discrete formulation and also shows the uniqueness of the discrete solution. Further, the authors discuss application and results of BDDC (Balancing Domain Decomposition by Constraints) method used for solution of the linear system.

In [50] a mortar-like method is used to deal with the discrete coupling between equations on incompatible 1D-2D meshes of different dimensions. The drawback of this approach is that it uses continuous approximation of velocity, so it cannot approximate the discontinuity of the velocity over the fractures accurately enough.

5.2 Mixed Dirichlet Problem

At the beginning, we consider a steady 2d problem with a single singularity present in a domain $\Omega \subset \mathbb{R}^2$. We take Problem 2.2.1 and restrain ourselves to a single aquifer, therefore omitting the index m, and a single well w perpendicular to the aquifer. Further we fix the pressure in the well, so we do not solve the 1d part of the problem and we can focus on the properties of the mixed form in the aquifer domain. We consider the aquifer thickness $\delta_2 = 1$ for simplicity. Due to this assumption we also omit the dimension index further in this section (we consider only $d = 2$).

We let the exterior boundary be $\Gamma_{ext} = \Gamma_D$, where we prescribe the Dirichlet boundary condition, for simplicity. The boundary of the domain $\partial\Omega = \Gamma_D \cup \Gamma_w$ then consists of only two parts: the exterior boundary Γ_D and the interior boundary Γ_w.

Let us now solve

Problem 5.2.1. Find $\boldsymbol{u}$ and p satisfying

$$\begin{aligned} \boldsymbol{K}^{-1}\boldsymbol{u} + \nabla p &= 0 && \text{in } \Omega, && (5.1a)\\ \operatorname{div} \boldsymbol{u} &= f && \text{in } \Omega, && (5.1b)\\ \langle \boldsymbol{u}\cdot\boldsymbol{n}\rangle_w &= \sigma_w(\langle p\rangle_w - P_w) && \text{on } \Gamma_w, && (5.1c)\\ \{p_2\}_w &= g_w && \forall w \in \mathcal{W}, && (5.1d)\\ p &= g_D && \text{on } \Gamma_D. && (5.1e) \end{aligned}$$

We consider similar assumptions on the input data as in the primary form in Section 4.1. In particular, the hydraulic conductivity tensor $\boldsymbol{K}$ is an invertible positive definite 2x2 matrix, for which we denote

$$\underline{k}^{-1} = \inf_{\boldsymbol{x}\in\Omega} \lambda_{\min}(\boldsymbol{K}^{-1}), \quad \bar{k}^{-1} = \sup_{\boldsymbol{x}\in\Omega} \lambda_{\max}(\boldsymbol{K}^{-1}), \qquad 0 < \underline{k}^{-1} \le \bar{k}^{-1}. \tag{5.2}$$

the minimum and maximum eigenvalues of the inverse matrix. The constant $P_w \in \mathbb{R}$ is the fixed pressure inside the well, $\sigma_w \in \mathbb{R}$, $\sigma_w > 0$ is the permeability coefficient between the well and the 2d domain. The condition (5.1c) relates the average of the normal flux over Γ_w in the outward direction to the pressure difference. Due to the fixed pressure P_w, this equation resembles a Robin type boundary condition.

5.2.1 Saddle Point Weak Form

We define the spaces for velocity and pressure, so we can derive the weak form of Problem 5.2.1:

$$\begin{aligned} V &= H(\operatorname{div}, \Omega), && (5.3)\\ Q &= L_2(\Omega). && (5.4) \end{aligned}$$

We test the equations (5.1a) and (5.1b)

$$\begin{aligned} \int_\Omega \boldsymbol{u}\boldsymbol{K}^{-1}\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega} p\,(\boldsymbol{v}\cdot\boldsymbol{n})\,\mathrm{d}s - \int_\Omega p \operatorname{div}\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} &= 0 && \forall \boldsymbol{v} \in V, && (5.5)\\ - \int_\Omega q \operatorname{div}\boldsymbol{u}\,\mathrm{d}\boldsymbol{x} &= -\int_\Omega f q\,\mathrm{d}\boldsymbol{x} && \forall q \in Q. && (5.6) \end{aligned}$$

In the first equation, we split the boundary integral and on the interior boundary we apply the average decomposition

$$\int_{\Gamma_w} p\,(\boldsymbol{v}\cdot\boldsymbol{n})\,\mathrm{d}s = \int_{\Gamma_w} \langle p\rangle_w \langle \boldsymbol{v}\cdot\boldsymbol{n}\rangle_w + \{p\}_w\{\boldsymbol{v}\cdot\boldsymbol{n}\}_w\,\mathrm{d}s. \tag{5.7}$$

Next, the boundary conditions (5.1c)-(5.1e) are applied, to obtain a saddle point problem

Problem 5.2.2. Find $\boldsymbol{u} \in V$ and $p \in Q$ satisfying

$$\begin{aligned} a(\boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{v}, p) &= \langle G, \boldsymbol{v} \rangle_{V' \times V} && \forall \boldsymbol{v} \in V, && (5.8a) \\ b(\boldsymbol{u}, q) &= \langle F, q \rangle_{Q' \times Q} && \forall q \in Q, && (5.8b) \end{aligned}$$

with the bilinear forms $a : V \times V \to \mathbb{R}$, $b : V \times Q \to \mathbb{R}$

$$\begin{aligned} a(\boldsymbol{u}, \boldsymbol{v}) &= \int_\Omega \boldsymbol{u} \boldsymbol{K}^{-1} \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} + \frac{1}{\sigma_w} \langle \boldsymbol{u} \cdot \boldsymbol{n} \rangle_w \langle \boldsymbol{v} \cdot \boldsymbol{n} \rangle_w \, |\Gamma_w| \,, && (5.9a) \\ b(\boldsymbol{v}, p) &= - \int_\Omega p \operatorname{div} \boldsymbol{v} \, \mathrm{d}\boldsymbol{x}, && (5.9b) \\ \langle G, \boldsymbol{v} \rangle_{V' \times V} &= - \int_{\Gamma_D} g_D (\boldsymbol{v} \cdot \boldsymbol{n}) \, \mathrm{d}s - P_w \langle \boldsymbol{v} \cdot \boldsymbol{n} \rangle_w \, |\Gamma_w| - \int_{\Gamma_w} g_w \{ \boldsymbol{v} \cdot \boldsymbol{n} \}_w \, \mathrm{d}s, && (5.9c) \\ \langle F, q \rangle_{Q' \times Q} &= - \int_\Omega f q \, \mathrm{d}\boldsymbol{x}. && (5.9d) \end{aligned}$$

We denote $\langle \cdot, \cdot \rangle_{V' \times V}$ duality between V and its dual space V'. The bilinear forms a, b then define the operators

$$\begin{aligned} A &: V \to V', & \langle A\boldsymbol{u}, \boldsymbol{v} \rangle_{V' \times V} &= a(\boldsymbol{u}, \boldsymbol{v}) & \forall \boldsymbol{u} \in V, \forall \boldsymbol{v} \in V, \\ B &: V \to Q', & \langle B\boldsymbol{v}, q \rangle_{Q' \times Q} &= b(\boldsymbol{v}, q) & \forall \boldsymbol{v} \in V, \forall q \in Q, \\ B^T &: Q \to V', & \langle \boldsymbol{v}, B^T q \rangle_{V \times V'} &= b(\boldsymbol{v}, q) & \forall \boldsymbol{v} \in V, \forall q \in Q, \end{aligned} \qquad (5.10)$$

and (5.8a)-(5.8b) can be rewritten as

$$\begin{aligned} A\boldsymbol{u} + B^T p &= G \quad \text{in } V', && (5.11a) \\ B\boldsymbol{u} &= F \quad \text{in } Q'. && (5.11b) \end{aligned}$$

Lemma 5.2.1. *The bilinear forms a, b in Problem 5.2.2 are continuous.*

Proof. To bound the average terms in the form a, we define auxiliary smooth function $\psi \in C^\infty(\overline{\Omega})$ with boundary values

$$\psi(\boldsymbol{x}) = \begin{cases} 1 & \text{on } \Gamma_w, \\ 0 & \text{on } \Gamma_D. \end{cases} \qquad (5.12)$$

Then for $\boldsymbol{v} \in V$ we have

$$\begin{aligned} \langle \boldsymbol{v} \cdot \boldsymbol{n} \rangle_w = \int_{\Gamma_w} \boldsymbol{v} \cdot \boldsymbol{n} \psi \, \mathrm{d}s &= \int_{\partial\Omega} (\psi \boldsymbol{v}) \cdot \boldsymbol{n} \, \mathrm{d}s \\ &= \int_\Omega \operatorname{div} (\psi \boldsymbol{v}) \, \mathrm{d}\boldsymbol{x} = \int_\Omega \psi \operatorname{div} \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} + \int_\Omega \boldsymbol{v} \cdot \nabla \psi \, \mathrm{d}\boldsymbol{x} \\ &\le \|\psi\|_{L_2(\Omega)} \|\operatorname{div} \boldsymbol{v}\|_{L_2(\Omega)} + \|\nabla \psi\|_{L_2(\Omega)} \|\boldsymbol{v}\|_{L_2(\Omega)} \le C_w \|\boldsymbol{v}\|_{H(\mathrm{div}, \Omega)} \end{aligned} \qquad (5.13)$$

with a constant $C_w(\partial\Omega)$. Then we obtain the following bound

$$\begin{aligned} |a(\boldsymbol{u},\boldsymbol{v})| &\le \bar{k}^{-1}\,\|\boldsymbol{u}\|_{L_2(\Omega)}\|\boldsymbol{v}\|_{L_2(\Omega)} + \sigma_w\,|\Gamma_w|\,\langle\boldsymbol{u}\cdot\boldsymbol{n}\rangle_w\langle\boldsymbol{v}\cdot\boldsymbol{n}\rangle_w \\ &\le \alpha_2\|\boldsymbol{u}\|_{H(\mathrm{div},\Omega)}\|\boldsymbol{v}\|_{H(\mathrm{div},\Omega)} \quad \forall\boldsymbol{u},\boldsymbol{v}\in V, \end{aligned} \tag{5.14}$$

with a constant $\alpha_2\left(\bar{k}^{-1},\sigma_w,\partial\Omega\right)$. The continuity of b is straightforward

$$|b(\boldsymbol{v},p)| \le \|p\|_{L_2(\Omega)}\|\mathrm{div}\,\boldsymbol{v}\|_{L_2(\Omega)} \le \|p\|_{L_2(\Omega)}\|\boldsymbol{v}\|_{H(\mathrm{div},\Omega)} \quad \forall q\in Q,\ \forall\boldsymbol{v}\in V. \tag{5.15}$$

□

The bilinear form a is coercive on kernel space $V_0 = \{\boldsymbol{v}\in V : b(\boldsymbol{v},q) = 0\ \forall q\in Q\}$, i.e. a subspace of V where $\mathrm{div}\,\boldsymbol{v} = 0$,

$$\begin{aligned} a(\boldsymbol{v},\boldsymbol{v}) &\ge \underline{k}^{-1}\|\boldsymbol{v}\|^2_{L_2(\Omega)} + \sigma_w^{-1}\,|\Gamma_w|\,\langle\boldsymbol{v}\cdot\boldsymbol{n}\rangle_w^2 \\ &\ge \alpha_1\|\boldsymbol{v}\|^2_{L_2(\Omega)} = \alpha_1\|\boldsymbol{v}\|^2_{H(\mathrm{div},\Omega)} \quad \forall\boldsymbol{v}\in V_0, \end{aligned} \tag{5.16}$$

with $\alpha_1 = \underline{k}^{-1}$ and having the average term greater than zero.

We now show that our problem is inf-sup stable, i.e. it satisfies the so-called Ladyshenskaja-Babuška-Brezzi (LBB) condition, see [48], p. 42.

Lemma 5.2.2. *The Problem 5.2.2 satisfies the LBB condition*

$$\sup_{\boldsymbol{v}\in V}\frac{b(\boldsymbol{v},q)}{\|\boldsymbol{v}\|_{H(\mathrm{div},\Omega)}} \ge \beta_1\|q\|_{L_2(\Omega)}. \tag{5.17}$$

Proof. For the given $q\in Q$, we construct a function $\boldsymbol{v}$, satisfying the inequality. This is done through solving an auxiliary Poisson problem

$$\begin{aligned} -\Delta\varphi &= q \quad \text{in } \Omega \\ \varphi &= 0 \quad \text{on } \Gamma_D \\ \nabla\varphi\cdot\boldsymbol{n} &= 0 \quad \text{on } \Gamma_w \end{aligned} \tag{5.18}$$

with homogeneous Dirichlet b. c. on Γ_D and homogeneous Neumann b. c. on Γ_w. From the ellipticity of the artificial problem we get

$$\|\nabla\varphi\|_{L_2(\Omega)} \le \|\varphi\|_{H^1(\Omega)} \le C_F^2\|q\|_{L_2(\Omega)}, \tag{5.19}$$

with $C_F(\Omega,\Gamma_D)$ from Friedrich's inequality. Next we set $\boldsymbol{v} = -\nabla\varphi$ for which it holds $\{\boldsymbol{v}\cdot\boldsymbol{n}\}_w = 0$ and $\mathrm{div}\,\boldsymbol{v} = -\Delta\varphi = q$ for this particular φ. Then we can bound the norm

$$\|\boldsymbol{v}\|^2_{H(\mathrm{div},\Omega)} = \|\boldsymbol{v}\|^2_{L_2(\Omega)} + \|\mathrm{div}\,\boldsymbol{v}\|^2_{L_2(\Omega)} \le (1+C_F^4)\|q\|^2_{L_2(\Omega)}. \tag{5.20}$$

Using it in the LBB condition, we obtain

$$\sup_{\boldsymbol{v}\in V}\frac{b(\boldsymbol{v},q)}{\|\boldsymbol{v}\|_{H(\mathrm{div},\Omega)}} \ge \frac{\|q\|^2_{L_2(\Omega)}}{\|\nabla\varphi\|_{H(\mathrm{div},\Omega)}} \ge \frac{1}{(1+C_F^4)^{1/2}}\|q\|_{L_2(\Omega)} = \beta_1\|q\|_{L_2(\Omega)}, \tag{5.21}$$

and having the constant $\beta_1 = (1+C_F^4)^{-1/2}$. □

Using the Brezzi's theorem ([48], p. 42), we see that we have now all necessary conditions for the existence of the unique solution $\boldsymbol{u}$, p of Problem (5.2.2). The pressure part of the solution p is according to the Brezzi's theorem defined up to an element of $\mathrm{Ker}B^T$. If we use the same construction of $\boldsymbol{v}$ as in the auxiliary Poisson problem (5.18), we see, that

$$\langle \boldsymbol{v}, B^T q \rangle_{V \times V'} = b(\boldsymbol{v}, q) = \int_\Omega q^2 \,\mathrm{d}\boldsymbol{x} \geq 0$$

and so that $b(\boldsymbol{v}, q) = 0 \iff q = 0$ and therefore $\mathrm{Ker}B^T = \{0\}$.

We summarize the results on the continuous spaces in Lemmas 5.2.1 and 5.2.2 into the following theorem

Theorem 5.2.3. *Let a be a continuous bilinear form on $V \times V$ and b be a continuous bilinear form on $V \times Q$, i.e.*

$$\begin{aligned}
|a(\boldsymbol{u}, \boldsymbol{v})| &\leq \alpha_2 \|\boldsymbol{u}\|_{H(\mathrm{div},\Omega)} \|\boldsymbol{v}\|_{H(\mathrm{div},\Omega)} \quad \forall \boldsymbol{u}, \boldsymbol{v} \in V, && (5.22)\\
|b(\boldsymbol{v}, p)| &\leq \beta_2 \leq \|p\|_{L_2(\Omega)} \|\boldsymbol{v}\|_{H(\mathrm{div},\Omega)} \quad \forall q \in Q,\ \forall \boldsymbol{v} \in V, && (5.23)
\end{aligned}$$

with the constants α_2 from (5.14) and $\beta_2 = 1$ from (5.15). Let us suppose that a is coercive on the kernel V_0 of the space V, i.e. there holds

$$a(\boldsymbol{v}, \boldsymbol{v}) \geq \alpha_1 \|\boldsymbol{v}\|^2_{H(\mathrm{div},\Omega)} \quad \forall \boldsymbol{v} \in V_0, \tag{5.24}$$

with the constant α_1 from (5.16), and that there holds the LBB condition

$$\sup_{\boldsymbol{v} \in V} \frac{b(\boldsymbol{v}, q)}{\|\boldsymbol{v}\|_{H(\mathrm{div},\Omega)}} \geq \beta_1 \|q\|_{L_2(\Omega)} \quad \forall q \in Q, \tag{5.25}$$

with β_1 from (5.21).
Then there exists a unique solution $\boldsymbol{u}$, p to Problem (5.2.2) for any $F \in V'$ and $G \in Q'$ such that the solution fulfills the stability estimate

$$\begin{aligned}
\|\boldsymbol{u}\|_{H(\mathrm{div},\Omega)} &\leq \frac{1}{\alpha_1}\|F\|_{V'} + \left(\frac{\alpha_2}{\alpha_1} + 1\right)\frac{1}{\beta_1}\|G\|_{Q'}, && (5.26)\\
\|p\|_{L_2(\Omega)} &\leq \frac{1}{\beta_2}\left(1 + \frac{\alpha_2}{\alpha_1}\right)\|F\|_{V'} + \frac{\alpha_2}{\beta_1^2}\left(1 + \frac{\alpha_2}{\alpha_1}\right)\|G\|_{Q'}. && (5.27)
\end{aligned}$$

5.2.2 Discretization

The discretization applied in the flow model in Flow123d is followed in this section and it is enhanced using an enrichment of the velocity space. In contrast to the first part of this work, simplicial meshes are considered from now on. At the beginning a mixed-hybrid discretization approach is described, then a proper enrichment of the discrete velocity space is introduced.

For the approximation of the $H(\mathrm{div}, \Omega)$ space, we use the lowest order Raviart-Thomas space

$$\mathbb{RT}^0(T^i) = \{\boldsymbol{\psi}_j\}_{j=1}^{n_F}, \tag{5.28}$$

on the element T^i, $i \in \mathcal{I}_E$. The vector basis functions of $\mathbb{RT}^0(T^i)$ are defined in the standard way

$$\boldsymbol{\psi}_j(\boldsymbol{x}) = \frac{|F^j|}{|T^i|}\left(\boldsymbol{x} - \boldsymbol{x}_k\right), \qquad \boldsymbol{x} \in T^i,\ j = 1 \dots n_F, \tag{5.29}$$

with $\boldsymbol{x}_k$ being the opposite node of the face F^j and n_F the number of faces of the element T^i (holds in all dimensions, putting $|F^j| := 1$ in 1d for point faces). Note that j is the index in local faces numbering on the element; we assume this implicitly further in text, if it is apparent from the range, e.g. $1 \dots n_F$. The degrees of freedom are interpreted as the fluxes over the corresponding faces

$$\int_{F_i} \boldsymbol{\psi}_j \cdot \boldsymbol{n} \,\mathrm{d}s = \delta_{ij} \qquad i, j = 1 \dots n_F. \tag{5.30}$$

Let us denote the standard discrete space for velocity

$$\tilde{V}_h = \{\boldsymbol{v}_h \in V : \boldsymbol{v}_h|_{T^i} \in \mathbb{RT}^0(T^i),\ i \in \mathcal{I}_E\}, \tag{5.31}$$

where the fluxes over faces are continuous, and the discrete space for pressure consisting of piecewise constant functions

$$Q_h = \{q_h \in L_2(\Omega) :\ q_h|_{T^i} \in \mathbb{P}^0(T^i),\ i \in \mathcal{I}_E\}. \tag{5.32}$$

One can then read the discrete counterpart of Problem 5.2.2 as follows

Problem 5.2.3. Find $\boldsymbol{u}_h \in \tilde{V}_h$ and $p_h \in Q_h$ satisfying

$$a(\boldsymbol{u}_h, \boldsymbol{v}_h) + b(\boldsymbol{v}_h, p_h) = \langle G, \boldsymbol{v}_h\rangle_{V' \times V} \qquad \forall \boldsymbol{v}_h \in \tilde{V}_h, \tag{5.33a}$$

$$b(\boldsymbol{u}_h, q_h) = \langle F, q_h\rangle_{Q' \times Q} \qquad \forall q_h \in Q_h. \tag{5.33b}$$

For the mixed problem, not all the assumptions in Theorem 5.2.3 hold automatically at the discrete level and the existence of a unique discrete solution is not straightforward. In particular the coercivity of a on $\tilde{V}_{0h}$ and LBB condition on discrete spaces $\tilde{V}_h$ and Q_h are not implied by the continuous inequalities (5.16) and (5.17).

Considering the $\mathbb{RT}^0$ and $\mathbb{P}^0$ finite elements, we have $\operatorname{div} \tilde{V}_h \subset Q_h$, i.e. $\operatorname{div} \tilde{V}_h$ is the space of piecewise constant functions. Thus $\tilde{V}_{0h} \subset V_0$ and the coercivity of a holds:

$$a(\boldsymbol{v}_h, \boldsymbol{v}_h) \geq \left\|\boldsymbol{v}_h\right\|_V^2 \qquad \forall \boldsymbol{v}_h \in \tilde{V}_{0h}. \tag{5.34}$$

Next, we define a local interpolation operator into $\mathbb{RT}^0(T)$ space of an arbitrary element T

$$\pi_T^{RT} \boldsymbol{v} = \sum_{j=1}^{n_F} \left(\int_{F^j} \boldsymbol{v} \cdot \boldsymbol{n} \,\mathrm{d}s\right) \boldsymbol{\psi}_j \qquad \forall \boldsymbol{v} \in V \tag{5.35}$$

which satisfies

$$\int_T \operatorname{div}(\pi_T^{RT}\boldsymbol{v} - \boldsymbol{v}) = 0, \tag{5.36}$$

$$b(\pi_T^{RT}\boldsymbol{v} - \boldsymbol{v}, q)_T = 0 \qquad \forall q \in Q_h, \tag{5.37}$$

since $q|_T \in \mathbb{P}^0$.

According to the proposition 2.8 in ([48], p. 58), we can show the inf-sup stability if we find a continuous interpolation operator $\pi_h : V \to V_h$ which satisfies

$$b(\pi_h\boldsymbol{v} - \boldsymbol{v}, q) = 0 \quad \forall q \in Q_h. \tag{5.38}$$

We define an interpolation operator

$$(\pi_h^{RT}\boldsymbol{v})|_T = \pi_T^{RT}\boldsymbol{v} \tag{5.39}$$

using the local interpolation operator from (5.35). This operator is not continuous on $V = H(\operatorname{div}, \Omega)$ but on smaller space we have

$$\|\pi_h^{RT}\boldsymbol{v}\|_V \leq C_\pi \|\boldsymbol{v}\|_W \qquad \forall \boldsymbol{v} \in W = [H^1(\Omega)]^d \subset H(\operatorname{div}, \Omega) \tag{5.40}$$

and from (5.25)

$$\sup_{\boldsymbol{v}\in W} \frac{(\operatorname{div}\boldsymbol{v}, q)}{\|\boldsymbol{v}\|_W} \geq \beta_1 \|q\|_Q \qquad \forall q \in Q. \tag{5.41}$$

The orthogonality of π_h^{RT} (5.38) comes as a consequence of (5.37). Using this we obtain for all $q_h \in Q_h$

$$\begin{aligned}\sup_{\boldsymbol{v}_h\in\tilde{V}_h} \frac{b(\boldsymbol{v}_h, q_h)}{\|\boldsymbol{v}_h\|_V} \geq \sup_{\boldsymbol{v}_h\in W} \frac{\int_\Omega q_h \operatorname{div}(\pi_h^{RT}\boldsymbol{v}_h)}{\|\pi_h^{RT}\boldsymbol{v}\|_V} \geq \frac{1}{C_\pi} \sup_{\boldsymbol{v}_h\in W} \frac{\int_\Omega q_h \operatorname{div}\boldsymbol{v}}{\|\boldsymbol{v}\|_W} \\ \geq \frac{\beta_1}{C_\pi}\|q_h\|_Q = \beta_{1h}\|q_h\|_Q\end{aligned} \tag{5.42}$$

We summarize the results above into the following lemma.

Lemma 5.2.4. *Let us have spaces V and Q for which the continuous LBB condition (5.25) holds. Let us choose discrete spaces $\tilde{V}_h \subset V$ and $Q_h \subset Q$ and let us have the continuous interpolation operator $\pi_h^{RT} : V \to V_h$, defined in 5.38.*

Due to the coercivity of a on discrete kernel space $\tilde{V}_{0h}$ by (5.34) and discrete inf-sup stability (5.42), the discrete Problem 5.2.3 has unique solution by Brezzi's Theorem 5.2.3.

Considering the hybridization, we define a new discrete space

$$V_h^{reg} = \prod_{i\in\mathcal{I}_E} \mathbb{RT}^0(T^i) \tag{5.43}$$

in which the functions normal components on the element faces are not continuous in contrast to $\tilde{V}_h$. Further we define a space of traces of the normal component of $\boldsymbol{v} \in \tilde{V}_h$ on the element boundaries

$$\begin{aligned} \Lambda(\mathcal{F}) &= \{\mu \in L_2(\mathcal{F} \setminus \partial\Omega) : \mu = (\boldsymbol{v} \cdot \boldsymbol{n})|_{\mathcal{F}\setminus\partial\Omega},\ \boldsymbol{v} \in \tilde{V}_h\} && (5.44) \\ \Lambda_h(\mathcal{F}) &= \{\mu \in \Lambda(\mathcal{F}) : \mu|_{F^i} \in \mathbb{P}^0(F^i),\ \forall i \in \mathcal{I}_F\} && (5.45) \end{aligned}$$

and a space of average traces of the normal component of the flux on the well-aquifer interface

$$\Lambda_h^{enr} = \{\mu_w \in \mathbb{P}^0(\Gamma_w) :\ \mu_w = \langle \boldsymbol{v} \cdot \boldsymbol{n}\rangle_w,\ \boldsymbol{v} \in \tilde{V}_h(\Omega)\}. \tag{5.46}$$

Having the proper spaces, we now derive the mixed-hybrid form. We multiply the equations (5.1a)-(5.1c) by test functions and integrate them elementwise:

$$\begin{aligned} \sum_{i\in\mathcal{I}_E} \left[\int_{T^i} \boldsymbol{u}\boldsymbol{K}^{-1}\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} - \int_{T^i} p \,\mathrm{div}\,\boldsymbol{v}\,\mathrm{d}\boldsymbol{x} + \int_{\partial T^i\setminus\partial\Omega} \lambda(\boldsymbol{v}\cdot\boldsymbol{n})\,\mathrm{d}s + \int_{\Gamma_w} \lambda_w \langle \boldsymbol{v}\cdot\boldsymbol{n}\rangle_w\,\mathrm{d}s \right] & \\ = -\sum_{i\in\mathcal{I}_F} \int_{F^i\cap\Gamma_D} g_D(\boldsymbol{v}\cdot\boldsymbol{n})\,\mathrm{d}s - \int_{\Gamma_w} g_w\{\boldsymbol{v}\cdot\boldsymbol{n}\}_w\,\mathrm{d}s \qquad & \forall \boldsymbol{v} \in V_h^{reg}, \qquad (5.47\text{a}) \end{aligned}$$

$$\begin{aligned} -\sum_{i\in\mathcal{I}_E} \int_{T^i} q\,\mathrm{div}\,\boldsymbol{u}\,\mathrm{d}\boldsymbol{x} = -\sum_{i\in\mathcal{I}_E} \int_{T^i} f\,q\,\mathrm{d}\boldsymbol{x} \qquad & \forall q \in Q_h, && (5.47\text{b}) \\ \sum_{i\in\mathcal{I}_F} \int_{F^i\setminus\partial\Omega} (\boldsymbol{u}\cdot\boldsymbol{n})\mu\,\mathrm{d}s = 0 \qquad & \forall \mu \in \Lambda_h(\mathcal{F}), && (5.47\text{c}) \\ \int_{\Gamma_w} \langle \boldsymbol{u}\cdot\boldsymbol{n}\rangle_w \mu_w\,\mathrm{d}s = \int_{\Gamma_w} \sigma_w(\lambda_w - P_w)\mu_w\,\mathrm{d}s \qquad & \forall \mu_w \in \Lambda_h^{enr}. && (5.47\text{d}) \end{aligned}$$

We now sum up equations (5.47b)-(5.47d), introduce the hybridized bilinear forms and transform Problem 5.2.3 into

Problem 5.2.4. Find $\boldsymbol{u}_h \in V_h^{reg}$ and $\mathring{p}_h = [p_h, \lambda, \lambda_w] \in Q_h \times \Lambda_h(\mathcal{F}) \times \Lambda_h^{enr}$ which satisfy

$$\begin{aligned} a_h(\boldsymbol{u}_h, \boldsymbol{v}_h) + b_h(\boldsymbol{v}_h, \mathring{p}_h) &= \langle G, \boldsymbol{v}_h\rangle_{V_h^{\prime reg}\times V_h^{reg}} && \forall \boldsymbol{v}_h \in V_h^{reg}, && (5.48\text{a}) \\ b_h(\boldsymbol{u}_h, \mathring{q}_h) - c_w(\mathring{p}_h, \mathring{q}_h) &= \langle F, \mathring{q}_h\rangle_{Q_h'\times Q_h} && \forall \mathring{q}_h \in Q_h \times \Lambda_h(\mathcal{F}) \times \Lambda_h^{enr} && (5.48\text{b}) \end{aligned}$$

with the bilinear forms

$$a_h(\boldsymbol{u}_h, \boldsymbol{v}_h) = \sum_{i \in \mathcal{I}_E} \int_{T^i} \boldsymbol{u} \boldsymbol{K}^{-1} \boldsymbol{v} \, \mathrm{d}\boldsymbol{x}, \tag{5.49a}$$

$$b_h(\boldsymbol{v}_h, \mathring{p}_h) = \sum_{i \in \mathcal{I}_E} \left[\int_{T^i} -p_h \operatorname{div} \boldsymbol{v}_h \, \mathrm{d}\boldsymbol{x} + \int_{\partial T^i \setminus \partial\Omega} \lambda (\boldsymbol{v}_h \cdot \boldsymbol{n}) \, \mathrm{d}s \right] \tag{5.49b}$$

$$+ \int_{\Gamma_w} \lambda_w \langle \boldsymbol{v}_h \cdot \boldsymbol{n} \rangle_w \, \mathrm{d}s, \tag{5.49c}$$

$$c_w(\mathring{p}_h, \mathring{q}_h) = \int_{\Gamma_w} \sigma_w \lambda_w \mu_w \, \mathrm{d}s, \tag{5.49d}$$

$$\langle G, \boldsymbol{v}_h \rangle_{V_h'^{reg} \times V_h^{reg}} = \sum_{i \in \mathcal{I}_F} \int_{F^i \cap \Gamma_D} -g_D (\boldsymbol{v}_h \cdot \boldsymbol{n}) \, \mathrm{d}s$$

$$- \int_{\Gamma_w} g_w \{ \boldsymbol{v}_h \cdot \boldsymbol{n} \}_w \, \mathrm{d}s, \tag{5.49e}$$

$$\langle F, \mathring{q} \rangle_{Q_h' \times Q_h} = \sum_{i \in \mathcal{I}_E} \int_{T^i} -f q_h \, \mathrm{d}\boldsymbol{x} - \int_{\Gamma_w} \sigma_w P_w \mu_w \, \mathrm{d}s. \tag{5.49f}$$

The continuity of the normal components on the element faces, which is missing in V_h^{reg}, is forced by the equation (5.47c), so that the solution $\boldsymbol{u}_h \in \tilde{V}_h$. Thus the problems 5.2.3 and 5.2.4 are equivalent.

If we take the equation (5.1a) on a single element T^i (let it be an interior element), multiply it by a test function and integrate by parts, we obtain

$$\int_{T^i} \boldsymbol{u} \boldsymbol{K}^{-1} \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} - \int_{T^i} p \operatorname{div} \boldsymbol{v} \, \mathrm{d}\boldsymbol{x}, + \int_{\partial T^i} p (\boldsymbol{v} \cdot \boldsymbol{n}) \, \mathrm{d}s$$

$$+ \int_{\Gamma_w} \langle p \rangle_w \langle \boldsymbol{v} \cdot \boldsymbol{n} \rangle_w \, \mathrm{d}s + \int_{\Gamma_w} \{p\} \{ \boldsymbol{v} \cdot \boldsymbol{n} \}_w \, \mathrm{d}s = 0. \tag{5.50}$$

Comparing (5.50) with (5.47a) it is apparent from the third term that the Lagrange multipliers $\lambda \in \Lambda_h(\mathcal{F})$ can be interpreted as the trace of pressure p on element faces $\mathcal{F}$. Similarly we see from the fourth term that the Lagrange multiplier $\lambda_w \in \Lambda_h^{enr}$ can be interpreted as the average of trace of pressure p on well-aquifer cross-section. See [51] for details.

The bilinear form c_w in (5.48b) is positive semidefinite. Considering the necessary conditions for unique solvability of Problem 5.2.3, Theorem 1.2, p. 47 in [48], implies the existence of a unique solution of Problem 5.2.4. The mixed-hybrid problem 5.2.4 is conceptually compatible with the model implemented in Flow123d, see [9, 1].

At last, we address the fluctuation term in (5.49e). Comparing it to the primary form in Section 4.2, we see, that this time the fluctuation on Γ_w does not influence the approximation space itself, but it appears on the right hand side. Since the fluctuation function g_w is mostly unknown and very small in real life problems, we neglect it in our model and implementation by setting it zero. As we saw in the test in Section 4.4 already, and as we see in further in the numerical tests in this chapter, it does not influence the convergence rate.

Velocity Enrichment

In addition to the discretization approach described above, we are now interested in finding a proper XFEM enrichment of the velocity space V_h^{reg}. To this end we proceed in a similar manner as in the pressure model in Section 4.2. We build an enrichment space V_h^{enr}, such that the discrete velocity space is

$$V_h = V_h^{reg} \oplus V_h^{enr}. \tag{5.51}$$

The choice of the logarithmic pressure enrichment is explained in Section 4.2.1 and is based on solving a local problem. The global enrichment function for velocity is then the derivative of pressure. We define the velocity global enrichment function

$$\boldsymbol{s}_w(\boldsymbol{x}) = -\frac{1}{S_e}\frac{\boldsymbol{r}_w}{r_w^2}, \qquad \operatorname{div}\boldsymbol{s}_w = 0, \tag{5.52}$$

where we call S_e an effective (lateral) surface

$$S_e = 2\pi\rho_w. \tag{5.53}$$

i.e. a part of Ω_C^w through which the well and aquifer are communicating. The effective surface normalizes the function, so we have the flux

$$\int_{\Gamma_w} \boldsymbol{s}_w \cdot \boldsymbol{n} \,\mathrm{d}s = 1. \tag{5.54}$$

For the global enrichment functions it holds on an element T arbitrarily intersected by a well

$$\begin{aligned} 0 = \int_{T\setminus\Omega_C^w} \operatorname{div}\boldsymbol{s}_w \,\mathrm{d}\boldsymbol{x} &= \int_{\partial(T\setminus\Omega_C^w)} \boldsymbol{s}_w \cdot \boldsymbol{n} \,\mathrm{d}s \\ &= \sum_{j=1}^{n_F} \int_{F^j\setminus\Omega_C^w} \boldsymbol{s}_w \cdot \boldsymbol{n} \,\mathrm{d}s + \int_{\Gamma_w\cap T} \boldsymbol{s}_w \cdot \boldsymbol{n} \,\mathrm{d}s. \end{aligned} \tag{5.55}$$

which implies two special cases

$$\sum_{j=1}^{n_F} \int_{F^j} \boldsymbol{s}_w \cdot \boldsymbol{n} \,\mathrm{d}s = 0 \qquad\qquad T \cap \Gamma_w = \emptyset, \tag{5.56}$$

$$\sum_{j=1}^{n_F} \int_{F^j} \boldsymbol{s}_w \cdot \boldsymbol{n} \,\mathrm{d}s = -\int_{\Gamma_w} \boldsymbol{s}_w \cdot \boldsymbol{n} \,\mathrm{d}s \qquad\qquad T \cap \Gamma_w = \Gamma_w. \tag{5.57}$$

The first case in (5.56) means that $\boldsymbol{s}_w$ has zero divergence on elements not intersecting with the well w (on most of the elements). The second case means that flux of $\boldsymbol{s}_w$ over element faces is equal the flux of $\boldsymbol{s}_w$ from the well, if the intersection lies inside of the element.

Similarly for $\mathbb{RT}^0(T)$ functions it holds

$$\int_{T\setminus\Omega_C^w} \operatorname{div}\boldsymbol{\psi}_j \,\mathrm{d}\boldsymbol{x} = \int_{\partial T} \boldsymbol{\psi}_j \cdot \boldsymbol{n} \,\mathrm{d}s =$$

$$= \begin{cases} 1 & T\cap\Gamma_w = \emptyset, \\ 1 - |F^j|\,|T|^{-1}\, 2\pi\rho_w^2 & T\cap\Gamma_w = \Gamma_w, \\ \int_{F^j\setminus\Omega_C^w} \boldsymbol{\psi}_j\cdot\boldsymbol{n}\,\mathrm{d}s + \int_{\Gamma_w\cap T}\boldsymbol{\psi}_j\cdot\boldsymbol{n}\,\mathrm{d}s & T\cap\Gamma_w \subset \Gamma_w. \end{cases} \tag{5.58}$$

We see in (5.58) that the first two cases, Γ_w outside or fully inside the element, do not imply any significant changes in the numerical computation in system matrix assembly. In the second case, the normal flux of $\boldsymbol{\psi}_j$ over Γ_w is computed analytically. Only the last case, the well intersecting an element face, means numerical evaluation of the face integrals.

Velocity Enrichment with PU We now introduce SGFEM like enrichment for the velocity. At first we mimic the steps taken in the pressure model discretization. Let us suppose a partition of unity

$$\sum_{i=1}^{n_{PU}} N_i(\boldsymbol{x}) = 1 \qquad \forall \boldsymbol{x}\in T. \tag{5.59}$$

The functions N_i will be specified later, but one can consider functions from $\mathbb{P}^1(T)$ to fix the following ideas. Since velocity is being enriched, we interpolate $\boldsymbol{s}_w$ using π_T^{RT} on an element T in the SGFEM:

$$\pi_T^{RT}(\boldsymbol{s}_w)(\boldsymbol{x}) = \sum_{j=1}^{n_F} z_j^w \boldsymbol{\psi}_j(\boldsymbol{x}), \qquad z_j^w = \int_{F^j} (\boldsymbol{s}_w\cdot\boldsymbol{n})\,\mathrm{d}s. \tag{5.60}$$

Following Section 3.2.2 we then obtain the local enrichment function

$$\boldsymbol{\phi}_{iw} = N_i \boldsymbol{L}_w, \qquad i = 1\dots n_{PU},\ \boldsymbol{L}_w = \boldsymbol{s}_w - \pi_T^{RT}\boldsymbol{s}_w. \tag{5.61}$$

However, such $\boldsymbol{\phi}_{iw}$ has non-zero flux over element faces, so it devalues the advantages of the delta property of $\mathbb{RT}^0$ functions (5.30). This property is important to retain the standard implementation of boundary conditions and possibly the coupling scheme for fractures in the mixed-hybrid model in Flow123d. Thus we would like to preserve this property and we suggest to include the PU functions into $\boldsymbol{L}_w$. The local enrichment functions for each $i = 1\dots n_{PU}$ then become

$$\boldsymbol{\phi}_{iw} = \boldsymbol{L}_{iw} = N_i\boldsymbol{s}_w - \sum_{j=1}^{n_F} z_j^{iw}\boldsymbol{\psi}_j, \quad z_j^{iw} = \int_{F^j} N_i(\boldsymbol{s}_w\cdot\boldsymbol{n})\,\mathrm{d}s. \tag{5.62}$$

We compute the divergence of $\boldsymbol{L}_{iw}$ and of the whole partition of unity:

$$\operatorname{div}\boldsymbol{L}_{iw} = \nabla N_i\cdot\boldsymbol{s}_w - \sum_{j=1}^{n_F} z_j^{iw}\operatorname{div}\boldsymbol{\psi}_j, \tag{5.63}$$

$$\sum_{i=1}^{n_{PU}}\operatorname{div}\boldsymbol{L}_{iw} = -\sum_{j=1}^{n_F}\operatorname{div}\boldsymbol{\psi}_j\int_{F^j}\boldsymbol{s}_w\cdot\boldsymbol{n}\,\mathrm{d}s. \tag{5.64}$$

We compute also the fluxes of $\boldsymbol{L}_{iw}$ and of the whole partition of unity:

$$\int_{F_k} \boldsymbol{L}_{iw} \cdot \boldsymbol{n} \,\mathrm{d}s = \int_{F_k} N_i \boldsymbol{s}_w \cdot \boldsymbol{n} \,\mathrm{d}s - \sum_{j=1}^{n_F} z_j^{iw} \int_{F_k} \boldsymbol{\psi}_j \cdot \boldsymbol{n} \,\mathrm{d}s$$

$$= \int_{F_k} N_i \boldsymbol{s}_w \cdot \boldsymbol{n} \,\mathrm{d}s - z_k^{iw} = 0 \qquad \forall k = 1 \dots n_F, \tag{5.65}$$

$$\int_T \sum_{i=1}^{n_{PU}} \operatorname{div} \boldsymbol{L}_{iw} \,\mathrm{d}\boldsymbol{x} = - \sum_{j=1}^{n_F} \int_{Fj} \boldsymbol{s}_w \cdot \boldsymbol{n} \,\mathrm{d}s \int_T \operatorname{div} \boldsymbol{\psi}_j \,\mathrm{d}\boldsymbol{x} = 0 \qquad T \cap \Gamma_w = \emptyset., \tag{5.66}$$

We see from (5.65) that this kind of enrichment does not affect the meaning of the standard degrees of freedom (the fluxes of $\boldsymbol{\psi}_j$ over element faces). The property (5.66) can be checked when filling local element matrices during assembly, and can be a good guide for finding errors.

Finally, we need to choose a partition of unity. Since the hybridization decoupled the elements, the simplest choice is $\mathbb{P}^0$, the enriched velocity then has the form:

$$\boldsymbol{u}_h = \sum_{i=1}^{n_F \cdot N_E} a_i \boldsymbol{\psi}_i + \sum_{w \in \mathcal{W}} \sum_{i \in \mathcal{J}_E^w} b_{iw} \boldsymbol{\phi}_{iw}. \tag{5.67}$$

The enrichment area Z_w is given by the enrichment radius R_w as in the pressure model. The enriched nodes and elements are then defined the same way by index sets $\mathcal{J}_N^w$, $\mathcal{J}_E^w$, respectively, see (4.37) and (4.38). There is a single local enrichment function per element. Choosing $\mathbb{P}^1$ as a partition of unity, the enriched velocity then has the form:

$$\boldsymbol{u}_h = \sum_{i=1}^{n_F \cdot N_E} a_i \boldsymbol{\psi}_i + \sum_{w \in \mathcal{W}} \sum_{i \in \mathcal{J}_N^w} b_{iw} \boldsymbol{\phi}_{iw}. \tag{5.68}$$

There are 3 local enrichment functions per element, in case of 2d.

Unfortunately, we were unsuccessful in implementation of these PU. In case of $\mathbb{P}^0$, the coupling of the enrichment functions over the element faces was too loose, that the solution was utterly broken into pieces. Additional enrichment of the constant Lagrange multipliers, that would strengthen the connections over faces, might be worth trying.

The $\mathbb{P}^1$ partition of unity goes against the hybridization, since we have the regular part disconnected and the enriched part continuous over element faces. We were looking for proper Lagrange multipliers for hybridization of the enriched part, but we were unsuccessful in this matter.

A step back to solving the mixed model 5.2.3 and implementing an enrichment for that problem would be reasonable. However, the non-hybridized model is not implemented currently in Flow123d and it would require significant changes from mesh processing to output routines. Testing the enrichment in the mixed form is one of our future aims.

Velocity Enrichment without PU We simplify the enrichment so that we have only single enrichment function per singularity. On each enriched element T^i, $i \in \mathcal{J}_E^w$, we subtract the interpolation of the global enrichment function, see (5.60),

$$\boldsymbol{\phi}_w(\boldsymbol{x})|_{T^i} = \boldsymbol{L}_w(\boldsymbol{x})|_{T^i} = \boldsymbol{s}_w(\boldsymbol{x}) - \pi_{T^i}^{RT}(\boldsymbol{s}_w)(\boldsymbol{x}). \tag{5.69}$$

For such enrichment it holds:

$$\operatorname{div} \boldsymbol{L}_w = -\sum_{j=1}^{n_F} z_j^w \operatorname{div} \boldsymbol{\psi}_j, \tag{5.70}$$

$$\begin{aligned}\int_{F_k} \boldsymbol{L}_w \cdot \boldsymbol{n} \, \mathrm{d}s &= \int_{F_k} \boldsymbol{s}_w \cdot \boldsymbol{n} \, \mathrm{d}s - \sum_{j=1}^{n_F} z_j^w \int_{F_k} \boldsymbol{\psi}_j \cdot \boldsymbol{n} \, \mathrm{d}s \\ &= \int_{F_k} \boldsymbol{s}_w \cdot \boldsymbol{n} \, \mathrm{d}s - z_k^w = 0 \qquad \forall k = 1 \dots n_F, \end{aligned} \tag{5.71}$$

Finally, we define the enriched velocity in the form:

$$\boldsymbol{u}_h = \sum_{i=1}^{n_F \cdot N_E} a_i \boldsymbol{\psi}_i + \sum_{w \in \mathcal{W}} b_w \boldsymbol{\phi}_w. \tag{5.72}$$

We use this kind of enrichment later on, in our implementation in Flow123d and in all our test cases.

5.2.3 Numerical Verification of LBB Condition.

Proving the discrete LBB condition for enriched mixed finite element spaces is nontrivial. For a mixed problem, there exits a numerical validation technique how to test the inf-sup stability experimentally, e.g. in [52] for problems in incompressible elasticity.

The numerical validation is done at the discrete level, inspecting the eigenvalues of an associated eigenvalue problem. We rewrite equations (5.33) using the matrix notation

$$\mathbf{A}\mathbf{u} + \mathbf{B}^T\mathbf{p} = \mathbf{f}, \tag{5.73}$$

$$\mathbf{B}\mathbf{u} - \mathbf{C}\mathbf{p} = \mathbf{g}. \tag{5.74}$$

The matrix $\mathbf{C}$ is zero in this particular form, although later it represent the coupling terms between dimensions. The Schur complement of $\mathbf{A}$ is derived by elimination of velocity $\mathbf{u}$ from the system:

$$(\mathbf{C} + \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T)\mathbf{p} = \mathbf{B}\mathbf{A}^{-1}\mathbf{f} - \mathbf{g}. \tag{5.75}$$

The equation (5.75) includes only pressure variable. According to [48], p.76, we can construct a generalized eigenvalue problem

$$(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T)\mathbf{v}_i = \eta_i^2 \mathbf{S}\mathbf{v}_i \tag{5.76}$$

where $\mathbf{S}$ is a matrix associated with the scalar product of Q_h and η_i are the singular values of $\mathbf{B}$. The smallest non-zero singular value corresponds to the discrete inf-sup constant $\eta_{min} = \beta_{1h}$ according to [48], p.76.

The generalized eigenvalue problem (5.76) is solved for several refined meshes and the values of η_{min} are obtained. If η_{min} is independent of the refinement, i.e. it is constant for different mesh parameter h, then it indicates the in-sup stability. This of course cannot replace a mathematical proof, however, it might be used for problems where the proof is not available or is hard to obtain.

The problem in our case is that we actually solve the mixed-hybrid form (5.2.4). Therefore, by computing the Schur complement as in (5.75), we obtain an equation

$$(\mathbf{C}_h + \mathbf{B}_h\mathbf{A}_h^{-1}\mathbf{B}_h^T)\mathring{\mathbf{p}} = \mathbf{B}_h\mathbf{A}_h^{-1}\mathbf{f} - \mathbf{g}. \tag{5.77}$$

for $\mathring{\mathbf{p}} = [\mathbf{p}, \boldsymbol{\lambda}, \boldsymbol{\lambda}_w]$, pressure on elements and pressure on element sides/well edges (Lagrange multipliers). We suggest a new generalized eigenvalue problem

$$(\mathbf{B}_h\mathbf{A}_h^{-1}\mathbf{B}_h^T)\mathbf{v}_i = \mathring{\eta}_i^2\mathbf{S}_h\mathbf{v}_i \tag{5.78}$$

where we associate the matrix $\mathbf{S}_h$ with the scalar products of Q_h, $\Lambda_h(\mathcal{F})$, Λ_h^{enr} which are subsets of L_2 spaces on Ω, $\mathcal{F} \setminus \partial\Omega$ and Γ_w, respectively.

However, we are not sure about the relationship between η_{min} and $\mathring{\eta}_{min}$. Apparently, $\mathring{\eta}_{min}$ does not indicate inf-sup stability even for the regular case (without singularities), as we see below in a numerical experiment. We are aware of this shortcoming, the topic shall be pursued further. Better insight might give us the implementation of the mixed problem without hybridization, which is currently unavailable in Flow123d.

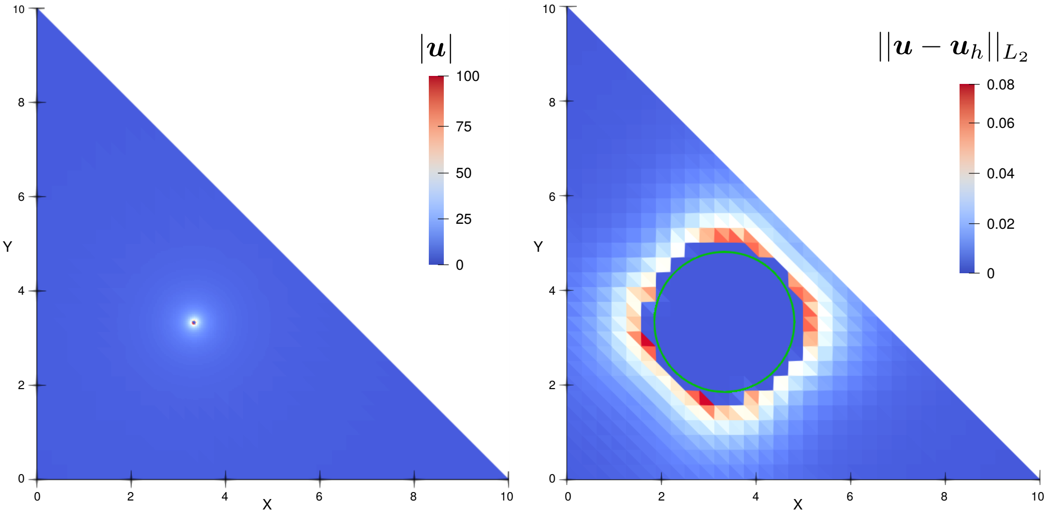
5.2.4 Numerical Test for Dirichlet Problem.

We provide the first numerical results for the Dirichlet problem 5.2.1. We consider a sequence of simple structured meshes of a triangle domain with nodes $[0, 0]$, $[10, 0]$ and $[0, 10]$. The well-aquifer cross-section is inside of an element for all refinement levels of the mesh. The input parameters for the test are gathered in Table 5.1.

$\boldsymbol{K}$	$\boldsymbol{x}_w$	ρ_w	R_w	P_w	ω	U
10^{-3}	[3.33,3.33]	0.03	1.5	100	1	20

Table 5.1: Input data for the Dirichlet problem.

Two cases are computed similarly as in Section 4.4.1, the first setting a zero source term $f = 0$, the second considering the source term $f = U\sin(\omega x)$. The solution error is computed against a pseudo-analytic solution determined according to Section 4.3, while setting the pressure in the well to constant P_w. The pseudo-analytic solution is used to set the Dirichlet boundary condition on the exterior boundary g_D, velocity is derived from the pressure $\boldsymbol{u} = -\boldsymbol{K}\nabla p$.



(a) velocity solution (magnitude) (b) L_2 elementwise error in velocity

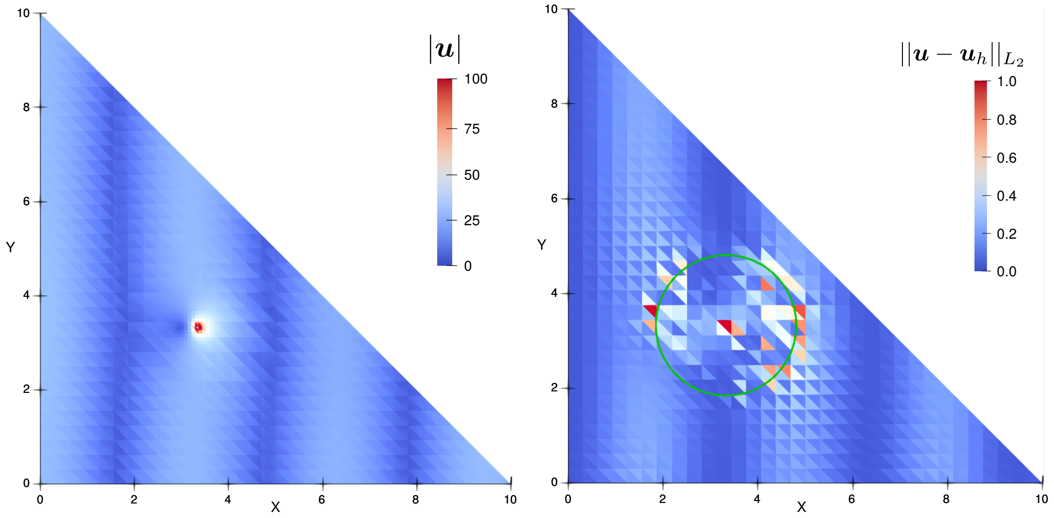
Figure 5.1: Results of Dirichlet problem for $f = 0$. The velocity solution is displayed at refinement level 4, the green circle indicates the enrichment radius R_w.

The results of the Dirichlet problem with zero source term can be seen in Figure 5.1. The singular velocity solution is displayed on the left, while the L_2 elementwise error of velocity is on the right. Similarly to the pressure model, the error is mainly concentrated on the edge of the enriched zone. We can see that the error is higher in $y = x$ direction than in $y = 6.66 - x$ direction which is caused by the shape of the elements in the structured mesh.

The convergence results for both cases are gathered in Table 5.2. As a reference, we provide also the convergence of the regular problem (non-zero source term, without singularity) in the third column. The first column shows the convergence results in this case, the convergence rate is closing to optimal.

		SGFEM, $f = 0$		SGFEM		MHFEM	
i	h	$\|\|\boldsymbol{u} - \boldsymbol{u}_h\|\|_{L_2(\Omega_2)}$	order	$\|\|\boldsymbol{u} - \boldsymbol{u}_h\|\|_{L_2(\Omega_2)}$	order	$\|\|\boldsymbol{u} - \boldsymbol{u}_h\|\|_{L_2(\Omega_2)}$	order
1	0.625	$7.12 \cdot 10^{-3}$	-	$2.42 \cdot 10^{-1}$	-	$2.42 \cdot 10^{-1}$	-
2	0.312	$4.10 \cdot 10^{-3}$	0.80	$1.25 \cdot 10^{-1}$	0.96	$1.25 \cdot 10^{-1}$	0.96
3	0.156	$2.21 \cdot 10^{-3}$	0.89	$6.40 \cdot 10^{-2}$	0.96	$6.41 \cdot 10^{-2}$	0.96
4	0.078	$1.14 \cdot 10^{-3}$	0.96	$3.11 \cdot 10^{-2}$	1.04	$3.11 \cdot 10^{-2}$	1.04
5	0.039	$0.57 \cdot 10^{-3}$	0.96	$1.58 \cdot 10^{-2}$	0.97	$1.56 \cdot 10^{-2}$	0.99

Table 5.2: Convergence table in mixed Dirichlet problem. MHFEM denotes the standard mixed-hybrid finite element method used in regular case.

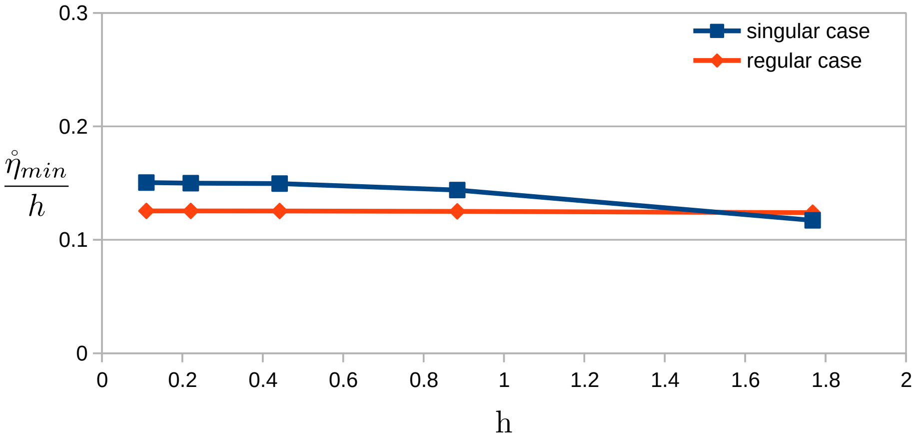


(a) velocity solution (magnitude)

(b) L_2 elementwise error in velocity

Figure 5.2: Results of Dirichlet problem for $f = U\sin(\omega x)$. The velocity solution is displayed at refinement level 4, the green circle indicates the enrichment radius R_w.

Analogically the results of the Dirichlet problem with the sinusoidal source term can be seen in Figure 5.2. The singular velocity solution is displayed on the left, while the L_2 elementwise error of velocity is on the right. The highest error is again concentrated on the edge of the enriched zone but it also appears inside the enrichment zone. The convergence results in this case are shown in the second column of Table 5.2. The convergence rate is again nearly optimal, the magnitude of the error is closing to the error in the regular case.



Graph 5.1: Graph representing numerical test of the LBB condition of Dirichlet problem. The red line is for the case without singularity, the blue line is for the case including the singularity. The smallest eigenvalue $\mathring{\eta}_{min}$ decreases linearly with h.

Finally, we present the inf-sup test results in Graph 5.1 according to (5.78). In both cases, $\mathring{\eta}_{min}$ decreases linearly with h (note the fraction $\mathring{\eta}_{min}/h$ in the graph). This is in contrast to what we expect, in particular in the regular case where the

LBB condition is proved theoretically [9]. However, as we pointed it out in Section 5.2.3, we do not have deep knowledge of the inf-sup indicator in the hybridized case and therefore the result is not trustworthy. On the other hand, the optimal convergence result itself can be viewed as a confirmation of stability of our problem.

5.3 Coupled 1d-2d Model

Let us consider Problem 2.2.1, including multiple wells and aquifers. For the sake of simplicity, we only assume homogeneous Neumann boundary condition $g_{dN} = 0$, if $\Gamma_{dN} \neq \emptyset$. The saddle point problem is derived using the same steps as in derivation of Problem 5.2.4 with an additional treatment of the coupling terms.

Let us first define the weak spaces and their discrete counterparts. We denote

$$\tilde{V}(\Omega_d) = \{\boldsymbol{v}_d \in H(\mathrm{div}, \Omega_d) : \ \boldsymbol{v}_d \cdot \boldsymbol{n} = 0 \text{ on } \Gamma_{dN}\} \tag{5.79}$$

the velocity subspace of $H(\mathrm{div})$ that satisfy the zero flux condition on Neumann boundary and its discrete subset

$$\tilde{V}_h(\Omega_d) = \{\boldsymbol{v}_d \in \tilde{V}(\Omega_d) : \ \boldsymbol{v}_d|_{T_d^i} \in \mathbb{RT}^0(T_d^i), \ i \in \mathcal{I}_{dE}\}. \tag{5.80}$$

Considering the hybridization, the disconnected spaces are defined

$$V_h^{reg}(\Omega_d) = \{\boldsymbol{v}_d \in \prod_{i \in \mathcal{I}_{dE}} \mathbb{RT}^0(T_d^i) : \ \boldsymbol{v}_d \cdot \boldsymbol{n} = 0 \text{ on } \Gamma_{dN}\}, \tag{5.81}$$

$$Q_h(\Omega_d) = \prod_{i \in \mathcal{I}_{dE}} \mathbb{P}^0(T_d^i). \tag{5.82}$$

It holds $\tilde{V}_h(\Omega_d) \subset \tilde{V}(\Omega_d)$ and $\tilde{V}_h(\Omega_d) \subset V_h^{reg}(\Omega_d)$. The enriched part of velocity space is defined using the functions from (5.69), including all wells and aquifers

$$V_h^{enr} = \mathrm{span}\{\boldsymbol{\phi}_w^m : \ m \in \mathcal{M}, w \in \mathcal{W}\}. \tag{5.83}$$

Thus we have on the 2d domain the velocity space

$$V_h(\Omega_2) = V_h^{reg}(\Omega_2) \oplus V_h^{enr}. \tag{5.84}$$

In all domains we define the spaces of traces of the normal component of the flux on the element boundaries

$$\Lambda(\mathcal{F}_d) = \{\mu_d \in L_2(\mathcal{F}_d \setminus \partial\Omega_d) : \ \mu_d = (\boldsymbol{v} \cdot \boldsymbol{n})|_{\mathcal{F}_d \setminus \partial\Omega_d}, \ \boldsymbol{v} \in \tilde{V}_h(\Omega_d)\} \tag{5.85}$$

and its discrete subset

$$\Lambda_h(\mathcal{F}_d) = \{\mu_d \in \Lambda(\mathcal{F}_d) : \ \mu_d|_{F_d^i} \in \mathbb{P}^0(F_d^i), \ \forall i \in \mathcal{I}_{dF}\}. \tag{5.86}$$

For handling the coupling terms, let us define a space of average traces of fluxes normal components on well edges

$$\Lambda_h^{enr} = \{\mu_w^m \in \mathbb{P}^0(\Gamma_w^m) : \ \mu_w^m = \langle \boldsymbol{v} \cdot \boldsymbol{n} \rangle_w^m, \ m \in \mathcal{M}, w \in \mathcal{W}, \ \boldsymbol{v} \in V_h(\Omega_2)\}. \tag{5.87}$$

Finally, let us denote the discretization spaces for the 1d-2d coupled problem:

$$V_h = V_h^{reg}(\Omega_1) \times V_h(\Omega_2), \tag{5.88}$$

$$Q_h = Q_h(\Omega_1) \times Q_h(\Omega_2) \tag{5.89}$$

$$\Lambda_h = \Lambda_h(\mathcal{F}_1) \times \Lambda_h(\mathcal{F}_2) \times \Lambda_h^{enr}, \tag{5.90}$$

Problem 5.3.1. Find $\boldsymbol{u}_h = [\boldsymbol{u}_d] \in V_h$ and $p_h = [p_d, \lambda_d, \lambda_w^m] \in Q_h \times \Lambda_h$, $d = 1, 2$, $m \in \mathcal{M}$, $w \in \mathcal{W}$ which satisfy

$$a_h(\boldsymbol{u}_h, \boldsymbol{v}_h) + b_h(\boldsymbol{v}_h, p_h) = \langle G, \boldsymbol{v}_h \rangle_{V_h' \times V_h} \qquad \forall \boldsymbol{v}_h \in V_h, \tag{5.91a}$$

$$b_h(\boldsymbol{u}_h, q_h) - c_w(p_h, q_h) = \langle F, q_h \rangle_{Q_h' \times Q_h} \qquad \forall q_h \in Q_h \times \Lambda_h \tag{5.91b}$$

where the forms are

$$a_h(\boldsymbol{u}_h, \boldsymbol{v}_h) = \sum_{\substack{d=1,2 \\ i \in \mathcal{I}_{dE}}} \int_{T_d^i} \delta_d^{-1} \boldsymbol{u}_d \boldsymbol{K}_d^{-1} \boldsymbol{v}_d \, \mathrm{d}\boldsymbol{x},$$

$$b_h(\boldsymbol{v}_h, p_h) = \sum_{\substack{d=1,2 \\ i \in \mathcal{I}_{dE}}} \left[\int_{T_d^i} -p_d \operatorname{div} \boldsymbol{v}_d \, \mathrm{d}\boldsymbol{x}, + \int_{\partial T_d^i \setminus \partial\Omega} \lambda_d (\boldsymbol{v}_d \cdot \boldsymbol{n}) \, \mathrm{d}s \right]$$

$$+ \sum_{\substack{w \in \mathcal{W} \\ m \in \mathcal{M}}} \int_{\Gamma_w^m} \lambda_w^m \langle \boldsymbol{v}_2 \cdot \boldsymbol{n} \rangle_w^m \, \mathrm{d}s,$$

$$c_w(p_h, q_h) = \sum_{\substack{w \in \mathcal{W} \\ m \in \mathcal{M}}} \int_{\Gamma_w^m} \delta_2(\boldsymbol{x}_w^m) \sigma_w^m \big(p_1(\boldsymbol{x}_w^m) - \lambda_w\big)\big(q_1(\boldsymbol{x}_w^m) - \mu_w\big) \, \mathrm{d}s,$$

$$\langle G, \boldsymbol{v}_h \rangle_{V_h' \times V_h} = \sum_{d=1,2} \sum_{i \in \mathcal{T}_d} \int_{\partial T_d^i \cap \Gamma_{dD}} -g_{dD} (\boldsymbol{v}_d \cdot \boldsymbol{n}) \, \mathrm{d}s,$$

$$\langle F, q_h \rangle_{Q_h' \times Q_h} = \sum_{d=1,2} \sum_{i \in \mathcal{T}_d} \int_{T_d^i} -\delta_d f_d q_d \, \mathrm{d}\boldsymbol{x},$$

The coupling terms are again emphasized in blue color.

We note that the nonzero Neumann boundary condition is also implemented. It technically means changing the right hand side to

$$\langle F, q \rangle = \sum_{d=1,2} \sum_{i \in \mathcal{T}_d} \left(\int_{T_d^i} -\delta_d f_d q_d - \int_{\partial T_d^i \cap \Gamma_{dN}} g_{dN} \mu_d \right) \mathrm{d}s. \tag{5.92}$$

and searching for $\boldsymbol{u}_h = \boldsymbol{u}_{0h} + \boldsymbol{u}_{hN}$, where $\boldsymbol{u}_h, \boldsymbol{u}_{hN}$ are from larger space than V_h, dismissing the zero flux condition, $\boldsymbol{u}_{hN}$ satisfying the Neumann condition and $\boldsymbol{u}_{0h} \in V_h$.

5.4 Coupled 1d-3d Model

Let us consider Problem 2.2.2, including multiple wells. For the sake of simplicity, we only assume homogeneous Neumann boundary condition $g_{dN} = 0$, if $\Gamma_{dN} \neq \emptyset$.

Before we define the saddle point form, we must specify how the enrichment functions are computed in 3d, how the enriched elements are selected and what is the actual support of the enrichment functions.

5.4.1 Enrichment Function in 3d

We define the velocity global enrichment function in the same form as in 2d case in (5.52)

$$\boldsymbol{s}_w(\boldsymbol{x}) = -\frac{1}{S_e}\frac{\boldsymbol{r}_w}{r_w^2}, \qquad S_e = 2\pi\rho_w z, \qquad \mathrm{div}\boldsymbol{s}_w = \frac{1}{r_w^2}. \tag{5.93}$$

where the effective surface S_e is the lateral surface of a cylinder of length z. The global enrichment functions have these properties on an element T:

$$\begin{aligned}\int_{T\setminus\Omega_C^w} \mathrm{div}\,\boldsymbol{s}_w \,\mathrm{d}\boldsymbol{x} &= \int_{\partial T\setminus\Omega_C^w} \boldsymbol{s}_w\cdot\boldsymbol{n}\,\mathrm{d}s = \\ &= \begin{cases} \sum\limits_{j=1}^{n_F} \int_{F^j\setminus\Omega_C^w} \boldsymbol{s}_w\cdot\boldsymbol{n}\,\mathrm{d}s = \int_{\Gamma_w} \boldsymbol{s}_w\cdot\boldsymbol{n}\,\mathrm{d}s & T\cap\Gamma_w = \emptyset, \\ \sum\limits_{j=1}^{n_F} \int_{F^j\setminus\Omega_C^w} \boldsymbol{s}_w\cdot\boldsymbol{n}\,\mathrm{d}s = \int_{\Gamma_w\cap T} \boldsymbol{s}_w\cdot\boldsymbol{n}\,\mathrm{d}s & T\cap\Gamma_w \subset \Gamma_w. \end{cases}\end{aligned} \tag{5.94}$$

Similarly for $\mathbb{RT}^0(T)$ functions it holds

$$\begin{aligned}\int_{T\setminus\Omega_C^w} \mathrm{div}\,\boldsymbol{\psi}_j \,\mathrm{d}\boldsymbol{x} &= \int_{\partial T\setminus\Omega_C^w} \boldsymbol{\psi}_j\cdot\boldsymbol{n}\,\mathrm{d}s = \\ &= \begin{cases} 1 & T\cap\Gamma_w = \emptyset, \\ \int_{F^j\setminus\Omega_C^w} \boldsymbol{\psi}_j\cdot\boldsymbol{n}\,\mathrm{d}s + \int_{\Gamma_w\cap T} \boldsymbol{\psi}_j\cdot\boldsymbol{n}\,\mathrm{d}s & T\cap\Gamma_w \subset \Gamma_w. \end{cases}\end{aligned} \tag{5.95}$$

The evaluation of the global enrichment function $\boldsymbol{s}_w$ in 3d is little bit more involved than in the 2d case. The logarithmic (in pressure) and hyperbolic (in velocity) singularity is concentrated along a 1d line, so r_w in (5.93) is the distance function from the line.

Let us have the direction vector $\boldsymbol{d} = \boldsymbol{b} - \boldsymbol{a}$ of an abscissa, $\boldsymbol{a}, \boldsymbol{b}$ being its endpoints. The abscissa corresponds to an 1d element of $\mathcal{T}_1$. Let us further have an arbitrary point $\boldsymbol{x}$ and compute its shortest distance vector $\boldsymbol{r}_w$ from $\boldsymbol{d}$, see Figure 5.3. The angle α between $(\boldsymbol{x}-\boldsymbol{a})$ and $\boldsymbol{d}$ is

$$\cos(\alpha) = \frac{(\boldsymbol{x}-\boldsymbol{a})\cdot\boldsymbol{d}}{|\boldsymbol{x}-\boldsymbol{a}|\,|\boldsymbol{d}|} = \frac{|\boldsymbol{d}_p|}{|\boldsymbol{x}-\boldsymbol{a}|}.$$

From there we use $|\boldsymbol{d}_p|$ to obtain the projection of $(\boldsymbol{x}-\boldsymbol{a})$ onto $\boldsymbol{d}$

$$\boldsymbol{d}_p = |\boldsymbol{d}_p|\frac{\boldsymbol{d}}{|\boldsymbol{d}|} = \frac{(\boldsymbol{x}-\boldsymbol{a})\cdot\boldsymbol{d}}{|\boldsymbol{d}|^2}\boldsymbol{d}.$$

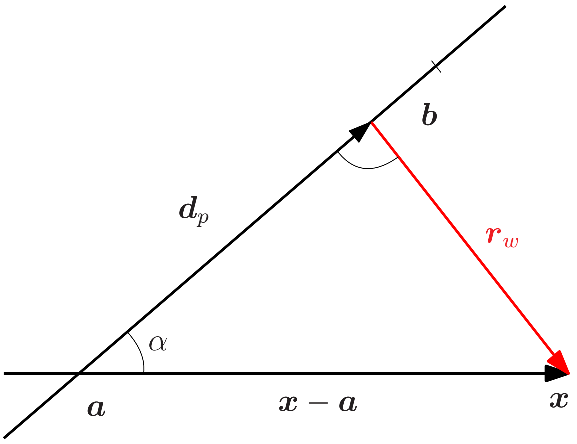


Figure 5.3: Distance vector $\boldsymbol{r}_w$ in 3d.

Then the distance vector is

$$\boldsymbol{r}_w(\boldsymbol{x}) = (\boldsymbol{x}-\boldsymbol{a}) - \boldsymbol{d}_p = (\boldsymbol{x}-\boldsymbol{a}) - \frac{(\boldsymbol{x}-\boldsymbol{a})\cdot\boldsymbol{d}}{|\boldsymbol{d}|^2}\boldsymbol{d}. \tag{5.96}$$

Next let us consider a mesh of a single well Ω_1^w consisting of elements T_1^i. For each element a global enrichment function $\boldsymbol{s}_{iw}$ is defined as in (5.93), where $\boldsymbol{r}_w$ is the distance vector to the particular element T_1^i. For each $\boldsymbol{s}_{iw}$ a cylindrical enrichment zone Z_w^i around the corresponding element T_1^i is determined, using the enrichment radius R_w, see Figure 5.4. The enrichment functions $\boldsymbol{s}_{iw}$ are in fact the same on all the elements, since we assume the wells to be straight (elements lie on a line). However, their enrichment zones differ from one element to another, so the enriched degrees of freedom can capture different singularity strength, if the pressure difference between the well and the bulk differs along the well. The enrichment zones Z_w^i

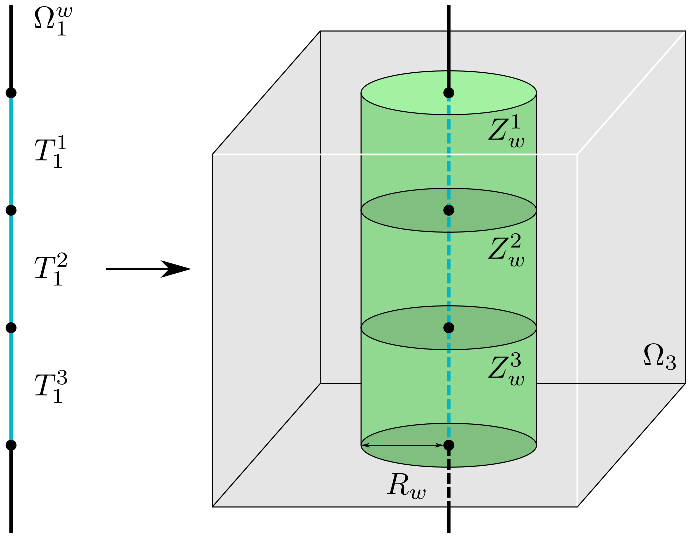


Figure 5.4: Cylindrical enrichment zone in 3d.

can have different enrichment radii, but we do not see any practical advantage for that at the moment.

Finally, we define the enriched velocity solution

$$\boldsymbol{u}_h = \sum_{i=1}^{n_F \cdot N_E} a_i \boldsymbol{\psi}_i + \sum_{w \in \mathcal{W}} \sum_{\substack{i \in \mathcal{I}_{1E} \\ T_1^i \subset \Omega_1^w}} b_{iw} \boldsymbol{\phi}_{iw} \tag{5.97}$$

where the local enrichment function is

$$\boldsymbol{\phi}_{iw}(\boldsymbol{x}) = \boldsymbol{L}_{iw}(\boldsymbol{x}) = \boldsymbol{s}_{iw}(\boldsymbol{x}) - \pi^{RT}_{T_3^j}(\boldsymbol{s}_{iw})(\boldsymbol{x}), \qquad j \in \mathcal{J}_E^w. \tag{5.98}$$

We see in (5.97) that there is one enriched degree of freedom β_{iw} per element of each well.

5.4.2 Saddle Point Problem in 3d

We build the saddle point problem like we did in 1d-2d coupled model, we use the established notation of function spaces in Section 5.3, where possible. Let us redefine the velocity space for 3d domain

$$V_h(\Omega_3) = V_h^{reg}(\Omega_3) \oplus V_h^{enr} \tag{5.99}$$

where the enriched part includes the enrichment functions specified in (5.98):

$$V_h^{enr} = \operatorname{span}\{\boldsymbol{\phi}_{iw} : w \in \mathcal{W},\, i \in \mathcal{I}_{1E}\}. \tag{5.100}$$

In the 3d case the pressure average along the Ω_1^w is computed according to (2.20). Considering a mesh $\mathcal{T}_1$ of the 1d domain, see Figure 5.4, the interior boundary term on cylinder lateral surface Γ_w is:

$$\begin{aligned}
\int_{\Gamma_w} p_3(\boldsymbol{v}_3 \cdot \boldsymbol{n})\,\mathrm{d}s &= \int_0^1 \int_0^{2\pi} p_3(\boldsymbol{v}_3 \cdot \boldsymbol{n})\rho_w \left|\Omega_1^w\right| \,\mathrm{d}\theta\,\mathrm{d}t \\
&= \int_0^1 \int_0^{2\pi} \Big[\langle p_3\rangle_w(t)\langle \boldsymbol{v}_3 \cdot \boldsymbol{n}\rangle_w(t) + \{p_3\}_w(t)\{\boldsymbol{v}_3 \cdot \boldsymbol{n}\}_w(t)\Big]\rho_w \left|\Omega_1^w\right| \,\mathrm{d}\theta\,\mathrm{d}t \\
&\approx 2\pi\rho_w \left|\Omega_1^w\right| \int_0^1 \langle p_3\rangle_w(t)\langle \boldsymbol{v}_3 \cdot \boldsymbol{n}\rangle_w(t)\,\mathrm{d}t \\
&= 2\pi\rho_w \left|\Omega_1^w\right| \sum_{\substack{i \in \mathcal{I}_{1E} \\ T_1^i \subset \Omega_1^w}} \int_{T_1^i} \langle p_3\rangle_w(t)\langle \boldsymbol{v}_3 \cdot \boldsymbol{n}\rangle_w(t)\,\mathrm{d}t. \qquad (5.101)
\end{aligned}$$

In (5.101) we neglected the fluctuation part. We now create a new space of Lagrange multipliers similar to (5.87), i.e. a space of average traces of fluxes normal components on Γ_w:

$$\Lambda^{enr} = \{\mu_w \in L_2(\Gamma_w) :\ \mu_w(t) = \langle \boldsymbol{v} \cdot \boldsymbol{n}\rangle_w(t),\ w \in \mathcal{W},\, \boldsymbol{v} \in V_h(\Omega_3)\}. \tag{5.102}$$

We further denote $\Omega_C^{iw} = \Omega_C^w|_{t\in T_1^i}$, a part of Ω_C^w along the element T_1^i, and define the discrete subset of Λ^{enr}

$$\Lambda_h^{enr} = \{\mu_w \in \Lambda^{enr} : \ \mu_w|_{\Omega_C^{iw}} = \mathbb{P}^0(\Omega_C^{iw}), \ w \in \mathcal{W}\}. \tag{5.103}$$

Using the spaces

$$\begin{aligned} V_h &= V_h^{reg}(\Omega_1) \times V_h(\Omega_3), && (5.104)\\ Q_h &= Q_h(\Omega_1) \times Q_h(\Omega_3) && (5.105)\\ \Lambda_h &= \Lambda_h(\mathcal{F}_1) \times \Lambda_h(\mathcal{F}_3) \times \Lambda_h^{enr}, && (5.106)\end{aligned}$$

we formulate the saddle point problem for 1d-3d coupling

Problem 5.4.1. Find $\boldsymbol{u}_h = [\boldsymbol{u}_d] \in V_h$ and $p_h = [p_d, \lambda_d, \lambda_w] \in Q_h \times \Lambda_h$, $d = 1, 3$, $w \in \mathcal{W}$ which satisfy

$$\begin{aligned} a_h(\boldsymbol{u}_h, \boldsymbol{v}_h) + b_h(\boldsymbol{v}_h, p_h) &= \langle G, \boldsymbol{v}_h\rangle_{V_h'\times V_h} && \forall \boldsymbol{v}_h \in V_h, && (5.107a)\\ b_h(\boldsymbol{u}_h, q_h) - c_w(p_h, q_h) &= \langle F, q_h\rangle_{Q_h'\times Q_h} && \forall q_h \in Q_h \times \Lambda_h && (5.107b)\end{aligned}$$

where the forms are

$$\begin{aligned} a_h(\boldsymbol{u}_h, \boldsymbol{v}_h) &= \sum_{\substack{d=1,3\\ i\in\mathcal{I}_{dE}}} \int_{T_d^i} \delta_d^{-1} \boldsymbol{u}_d \boldsymbol{K}_d^{-1} \boldsymbol{v}_d \,\mathrm{d}\boldsymbol{x},\\ b_h(\boldsymbol{v}_h, p_h) &= \sum_{\substack{d=1,3\\ i\in\mathcal{I}_{dE}}} \left[\int_{T_d^i} -p_d \operatorname{div} \boldsymbol{v}_d \,\mathrm{d}\boldsymbol{x}, + \int_{\partial T_d^i\setminus\partial\Omega} \lambda_d (\boldsymbol{v}_d\cdot\boldsymbol{n}) \,\mathrm{d}s\right]\\ &\quad + \sum_{w\in\mathcal{W}} \left[2\pi\rho_w\, |\Omega_1^w| \sum_{\substack{i\in\mathcal{I}_{1E}\\ T_1^i\subset\Omega_1^w}} \int_{T_1^i} \lambda_w(t) \langle \boldsymbol{v}_3\cdot\boldsymbol{n}\rangle_w(t) \,\mathrm{d}t\right],\\ c_w(p_h, q_h) &= \sum_{w\in\mathcal{W}} \left[2\pi\rho_w\, |\Omega_1^w| \sum_{\substack{i\in\mathcal{I}_{1E}\\ T_1^i\subset\Omega_1^w}} \int_{T_1^i} \sigma_w \big(p_1(t) - \lambda_w(t)\big)\big(q_1(t) - \mu_w(t)\big) \,\mathrm{d}t\right],\\ \langle G, \boldsymbol{v}_h\rangle_{V_h'\times V_h} &= \sum_{d=1,3} \sum_{i\in\mathcal{T}_d} \int_{\partial T_d^i\cap\Gamma_{dD}} -g_{dD} (\boldsymbol{v}_d\cdot\boldsymbol{n}) \,\mathrm{d}s,\\ \langle F, q_h\rangle_{Q_h'\times Q_h} &= \sum_{d=1,3} \sum_{i\in\mathcal{T}_d} \int_{T_d^i} -\delta_d f_d q_d \,\mathrm{d}\boldsymbol{x}. \end{aligned}$$

The coupling terms are again emphasized in blue color. Analogically to the 1d-2d model, see (5.92), nonzero Neumann boundary condition can be prescribed.

5.5 Flow123d implementation

The implementation of the mixed-hybrid model using XFEM is a part of the software Flow123d, version 3.0.x. The code is not however included in the official release yet, but it can be found on GitHub in an experimental branch `PE_xfem_master`. There are some new features in the master development branch at the moment which prevent a fast conflictless merge.

All about Flow123d, including its installation and usage inside Docker, the input file format and running simulations can be found through the official web pages [1], its GitHub pages and its current documentation [10].

Flow123d uses the YAML file format as the main input format. To solve the model with singularities and XFEM, additional keys must be specified. Code 5.1 provides an example input file structure.

Code 5.1: Flow123d input including XFEM.

```
problem: !Coupling_Sequential
  ...
  flow_equation: !Flow_Darcy_MH
    ...
    output_specific:
      compute_errors: true
      python_solution: analytical_solution.py
      output_linear_system: true
      fields:
        - velocity_enr
        - velocity_reg
        - velocity_diff
        - velocity_exact
        - pressure_diff
    use_xfem:
      use_xfem: true
      enrich_velocity: true
      enr_radius: 2.0
      dim: 2
    output_stream:
      ...
      output_mesh:
        max_level: 6
        refine_by_error: false
        error_control_field: pressure_exact
    ...
  ...
```

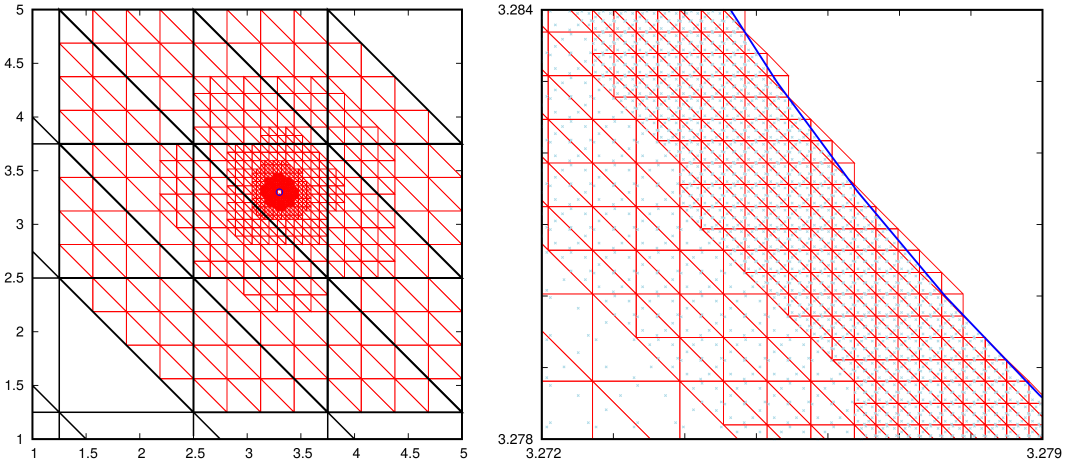
In the first 3 lines the specific problem and equation are selected, Darcy flow with mixed-hybrid solver in our case. On the lines 6-8, it is set that the L_2 norm of the solution is to be computed, the analytic solution is passed through the python script and the linear system is set to be output in the Matlab format. On the lines 9-14, the fields (quantities) that are to be output are selected, these are in a row: the enriched (singular) part of velocity, the regular part of velocity, the elementwise L_2 velocity error, the exact solution of velocity by the Python script and the elementwise L_2 pressure error.

On the lines 15-19, the XFEM is actually switched on. It is set that velocity is to be enriched (the only choice at the moment), the enrichment radius is set to 2.0 and the dimension in which the XFEM is applied is selected (is equal 3 in 3d cases). Finally on the lines 22-25, the refined output mesh can be defined. The output mesh can be refined to visualize the non-polynomial solution properly (described later in Section 5.5.2). The maximal level of the refinement and possibly the adaptive refinement by error can be selected.

The rest of the input file is standard as in other Flow123d simulations. We use similar settings in our test cases where analytic solution is available.

5.5.1 Adaptive quadrature

The adaptive quadrature implemented in Flow123d is based on the one developed for the quadrilateral meshes in Section 4.2.2. The same adaptive quadrature rules are used, only the distance function r is adapted to the simplicial elements. The detail of refined subelements and quadrature points is displayed in Figure 5.5.



(a) adaptive quadrature refinement (b) adaptive quadrature in detail

Figure 5.5: Adaptive quadrature for triangle elements implemented in Flow123d. Black lines denote enriched elements edges, red lines denote adaptive refinement (subelements edges) and the well edge is blue.

The adaptive quadrature is generalized for all three dimensions. This way we can use it in the computation of both integrals over elements and integrals over element faces (e.g. for accurate computation of z_j^w in π_T^{RT} interpolant (5.60)). Such quadrature on a face of tetrahedron is shown in Figure 5.6.

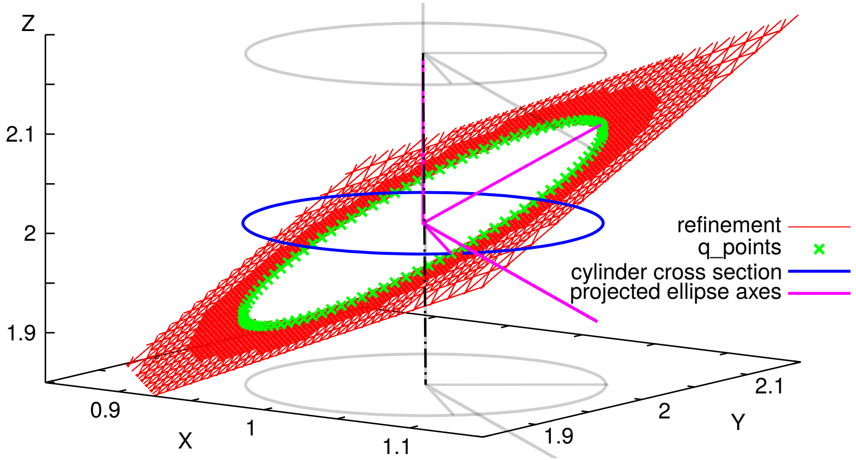


Figure 5.6: Adaptive quadrature on a face of tetrahedron intersected by a well. The well is parallel to z-axis (the gray and blue circles indicate the cylinder), the tetrahedron face is covered by the red subelements. First the face is mapped to the plane of the blue cross-section, then the subelements are constructed adaptively around the circle and mapped back to the face. Finally, the quadrature points are placed in the subelements. The actual cross-section is an ellipse (green points/purple axes).

In 3 dimensions the distance vector (5.96) is used to determine the distance of nodes and elements from the singularity. The subelements are refined using the edge splitting technique which is later described in Section 5.5.2.

5.5.2 Output Mesh

Due to the enrichment, the finite element approximation is not polynomial anymore. If we want to look at the discrete solution, e.g. in Paraview, we have to interpolate it into a space of piecewise constant or linear functions which can be viewed in such software. However, the interpolation on the computational mesh is quite coarse, so we suggested creating a finer mesh specially refined for the output.

Two types of refinement is implemented – uniform and adaptive. In the first case the mesh is uniformly refined in specified number of steps. In the later case the refinement is governed by an error criterion which bounds the measured elementwise L_2 norm of the difference between the (enriched) discrete solution and its constant/linear approximation. The refinement criterion reads as follows

$$\frac{\left\| v_h - \pi_{out} v_h \right\|_T}{\left\| v_h \right\|_T} < \mathrm{tol} \tag{5.108}$$

where π_{out} is the interpolator to a space of constant or linear functions on an arbitrary element T of the output mesh. The relative tolerance tol is specified by the user.

The refinement is meant to be used for the enriched discrete solutions in particular, however in Flow123d the output mesh can be refined according to any scalar quantity present in the model. Thus for the quantity v_h in (5.108) one can use for example the source term, the hydraulic conductivity or the supplied analytic solution.

Element Refinement

The refinement of elements for the output mesh is done using edge splitting technique (the so called red refinement). Since the output mesh is used only for better visualization of non-polynomial quantities, we do not care about any hanging nodes present in the refined mesh.

In 2d case, the refinement of an element is a straightforward process: find the midpoints of all sides, connect them and generate 4 triangles. These triangles are congruent and have equal surface areas. On the other hand, the 3d case is more complicated. After splitting the edges, we obtain 4 tetrahedra at the vertices of the original one. The octahedron that remains in the middle can be subdivided according to one of its three diagonals. Only the choice of the shortest octahedron diagonal leads to a regular tetrahedra decomposition. This algorithm originally comes from Bey [53], further e.g. in [54].

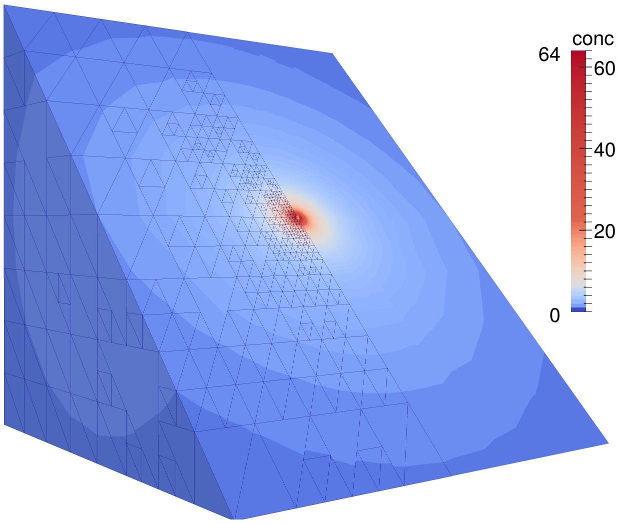


Figure 5.7: An example of an adaptive output mesh refinement in 3d. A singular function $1/r$ is displayed.

We demonstrate the refinement in Figure 5.7 where we used the adaptively refined output mesh to visualize a singular function $1/r$ in a cube. The domain is actually cut by a diagonal plane. One can see the mesh edges in the left half of the cube.

5.6 Numerical Tests in Flow123d

We present several numerical tests in which we demonstrate the properties and the behavior of the suggested enrichments in the mixed-hybrid method. The tests are computed using our implementation in Flow123d. The test setting in 2d is similar to the one used in Section 4.4.1 and 4.4.4. We will again investigate cases on different domains, with a single well or with multiple wells, with a source term or without.

An input file for Flow123d in Yaml format is prepared for each test. The pseudo-analytic solution is passed into Flow123d as a Python script. The solution error is evaluated against the pseudo-analytic solution determined according to Section 4.3. We remind the pressure solution (4.55) from which the velocity solution is derived:

$$
\begin{aligned}
p_2 &= p_{sin} + p_{reg} = \sum_{j \in \mathcal{W}} a_j \log r_j + p_{reg}, && (5.109)\\
\boldsymbol{u}_2 &= -\boldsymbol{K}_2 \nabla p_2 = -\boldsymbol{K}_2 \left[\sum_{j \in \mathcal{W}} a_j \frac{\boldsymbol{r}_j}{r_j^2} + \nabla p_{reg} \right]. && (5.110)
\end{aligned}
$$

The meshes used for computations are simplicial and unstructured. They are prepared by the software Gmsh using the default "MeshAdapt" algorithm. When examining the convergence of the method, we use Linux shell scripts and parameterized ".geo" files for Gmsh to generate series of refined meshes. We output the results into VTK format and visualize them in Paraview software.

5.6.1 Test Cases in 1d-2d

Test Case 1

In the first test case we consider Ω_2 to be a circular shaped domain of radius 5.0. A well is perpendicular to the domain and is intersecting the domain in its center.

Since the enrichment in 2d is of our main interest in this test case, we want to minimize any other effects influencing the approximation error. The strength of the singularity is determined by the pressure difference between the well and the aquifer, so the accuracy of the solution in the vicinity of the well is also dependent on the accuracy of pressure in the well, on $p_1(\boldsymbol{x}_w)$ in particular. A simple way to achieve that is simulating constant pressure inside the well by setting very high conductivity $\boldsymbol{K}_1 = 10^{10}$ and setting a constant Dirichlet boundary condition g_{1D} on both ends of the well.

Let us denote three different settings:

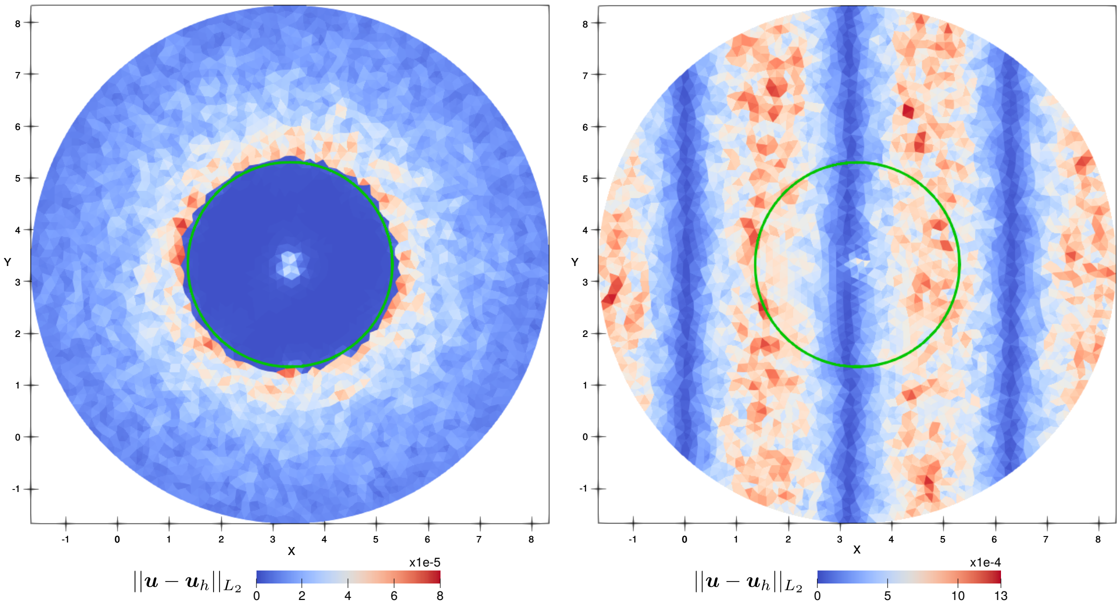
- TC1-a: $f = 0$,
- TC1-b: $f = U \sin(\omega x)$,
- TC1-c: $f = U \sin(\omega x)$, no singularity (regular case)

The first setting considers a zero source term, the second sets the sinusoidal source term. The last setting is the regular problem without the singularity, which we solve with the standard mixed-hybrid finite element method (MHFEM). The input parameters are gathered in Table 5.3.

$\boldsymbol{K}$	$\boldsymbol{x}_w$	ρ_w	σ_w	R_w	g_{1D}	ω	U
10^{-3}	[3.33,3.33]	0.03	10.0	2.0	100	1.0	200

Table 5.3: Input data for Test case 1.

The model is computed on a series of refined meshes and the convergence of our method is examined. The approximation error of velocity is displayed in Figure 5.8. We can see that the error is mainly accumulated outside the edge of the enriched zone in case of the zero source term. Considering the source term, the error of the singular part is inferior to the error of the regular part, as it is apparent in the right subfigure.



(a) TC1-a

(b) TC1-b

Figure 5.8: Results of Test case 1. The elementwise L_2 error in velocity is displayed at refinement level 5. The green circle indicates the enrichment radius R_w.

The convergence results are shown in Table 5.4. There are three convergence columns in the table corresponding to different settings. The one on the left is for the zero source term case, the middle one is for the model including the source term, the last column shows the convergence of the regular problem without the singularity solved with MHFEM. We see that the convergence of velocity is nearly optimal, closing to 1.0, in all three situations. Comparing the magnitude of the

		TC1-a		TC1-b		TC1-c	
i	h	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order
1	1.368	$7.11 \cdot 10^{-3}$	-	$2.42 \cdot 10^{-1}$	-	$2.42 \cdot 10^{-1}$	-
2	0.913	$5.31 \cdot 10^{-3}$	0.72	$1.72 \cdot 10^{-1}$	0.84	$1.72 \cdot 10^{-1}$	0.84
3	0.575	$3.30 \cdot 10^{-3}$	1.03	$9.99 \cdot 10^{-2}$	1.19	$9.99 \cdot 10^{-2}$	1.19
4	0.358	$2.19 \cdot 10^{-3}$	0.87	$6.40 \cdot 10^{-2}$	0.94	$6.41 \cdot 10^{-2}$	0.94
5	0.271	$1.61 \cdot 10^{-3}$	1.11	$4.50 \cdot 10^{-2}$	1.26	$4.51 \cdot 10^{-2}$	1.26
6	0.176	$1.08 \cdot 10^{-3}$	0.92	$3.03 \cdot 10^{-2}$	0.92	$3.02 \cdot 10^{-2}$	0.94

Table 5.4: Convergence table of SGFEM in Test case 1.

approximation error between TC1-b and TC1-c, we see that the singularity is only minimally affecting the error which is the desired result.

Test Case 2

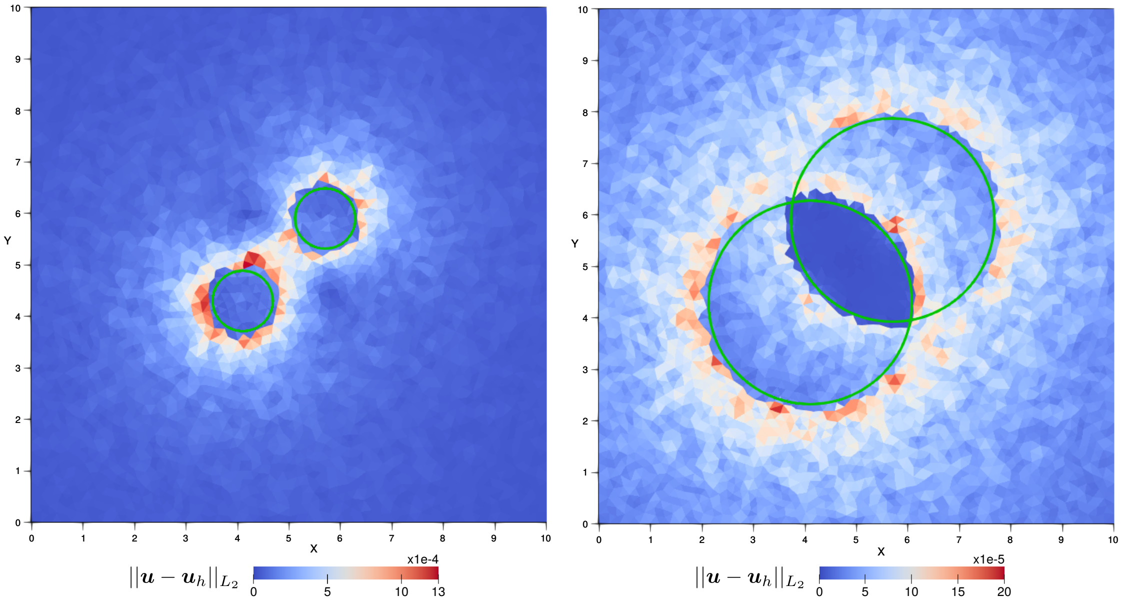
In the second test case we consider a square shaped domain $\Omega_2 = [0,10] \times [0,10]$, with two wells perpendicular to the domain Ω_2. Similarly to the previous case, the source term is set to zero or defined as $f = U \sin(\omega x)$, $U = 80$, $\omega = 1.0$. The pressure inside the wells is fixed to a constant value in the same way as in the previous case by setting high conductivity $\boldsymbol{K}_1$ and a constant Dirichlet boundary condition g_{1D} on both ends of the well. The parameters of the wells are gathered in Table 5.5.

w	$\boldsymbol{x}_w$	ρ_w	σ_w	R_w	g_{1D}
1	[4.1,4.3]	0.03	10.0	2.0	150
2	[5.7,5.9]	0.03	10.0	2.0	100

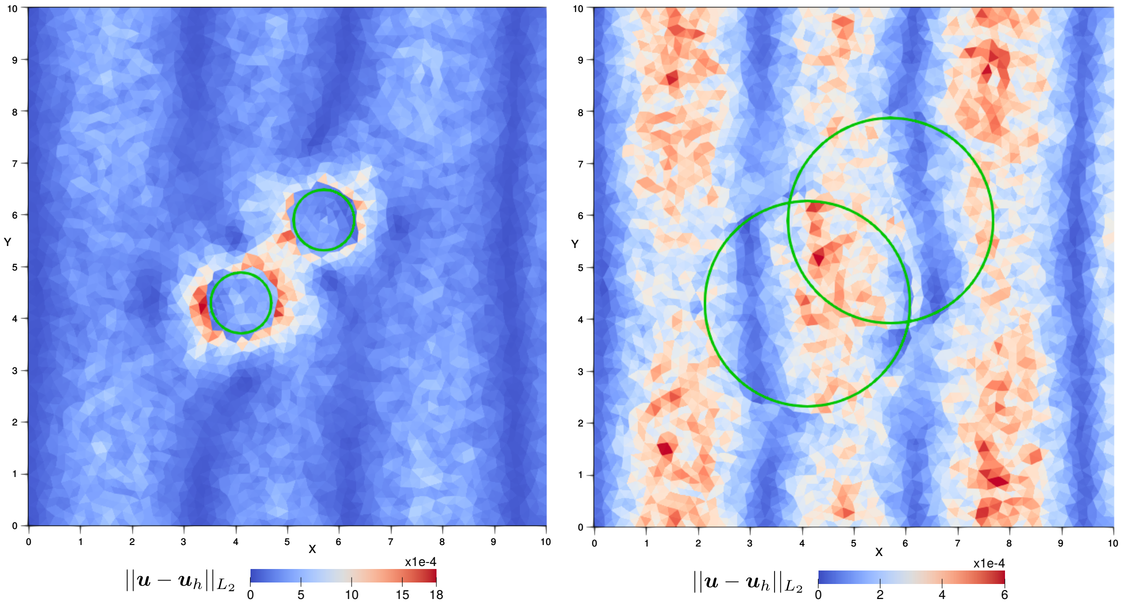
Table 5.5: Input data for the wells in Test case 2.

We solved the model with two different enrichment radii. Let us denote the following five settings:

- TC2-a: $f = 0$, $R_w = 0.6$
- TC2-b: $f = 0$, $R_w = 2.0$
- TC2-c: $f = U \sin(\omega x)$, $R_w = 0.6$
- TC2-d: $f = U \sin(\omega x)$, $R_w = 2.0$
- TC2-e: $f = U \sin(\omega x)$, no singularity (regular case)



(a) TC2-a (b) TC2-b



(c) TC2-c (d) TC2-d

Figure 5.9: Results of Test case 2. The elementwise L_2 error in velocity is displayed at refinement level 5. The green circle indicates the enrichment radius R_w.

Setting the larger enrichment radius leads to enrichment zones overlap.

The approximation error of velocity is displayed in Figure 5.9. We can again observe the error mainly accumulating outside the edge of the enriched zone in TC2-a,b,c. The error near the first well is little higher since the singularity is stronger (there is higher pressure inside the well). In case TC2-b, we see a very small error in the overlap of the enrichment zones since both singular enrichments are active there.

Comparing TC2-c and TC2-d, we see that the larger enrichment radius makes the error in the singular part insignificant in the scale of the error of the regular part.

		TC2-a		TC2-c		TC2-d	
i	h	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order
1	1.368	$6.71 \cdot 10^{-2}$	-	$1.38 \cdot 10^{-1}$	-	$1.10 \cdot 10^{-1}$	-
2	0.913	$3.44 \cdot 10^{-2}$	1.67	$7.80 \cdot 10^{-2}$	1.42	$6.94 \cdot 10^{-1}$	1.13
3	0.575	$2.64 \cdot 10^{-2}$	0.70	$5.50 \cdot 10^{-2}$	0.93	$4.59 \cdot 10^{-2}$	1.10
4	0.358	$1.89 \cdot 10^{-2}$	0.81	$3.85 \cdot 10^{-2}$	0.87	$3.12 \cdot 10^{-2}$	0.93
5	0.271	$1.34 \cdot 10^{-2}$	0.89	$2.62 \cdot 10^{-2}$	0.98	$2.06 \cdot 10^{-2}$	1.06
6	0.176	$9.24 \cdot 10^{-3}$	0.90	$1.79 \cdot 10^{-2}$	0.94	$1.37 \cdot 10^{-2}$	0.99

Table 5.6: Convergence table of SGFEM in Test case 2.

The convergence results are shown in Table 5.6. The three columns in the table correspond to three selected settings in Test case 2. We see that the convergence order of velocity approximation is closing to the optimum in all situations. The overlapping enrichment zones do not corrupt the accuracy of the approximation in TC2-d. Comparing the magnitude of the error in TC2-c and TC2-d columns, we see that the larger enrichment radius leads to a slightly increased accuracy while the convergence order is the same.

The model with TC2-b setting is converging also optimally, but we do not present all the results for the sake of brevity. The larger enrichment radius in TC2-b leads to decrease of the magnitude of the overall error, approximately with factor of 3.0 to the TC2-a setting. Similarly to Test case 1, the regular problem without singularities TC2-e is also computed. The convergence results correspond to the setting TC2-d, the magnitude of the approximation error and the convergence order are almost identical due to the accurate approximation of the singularities.

Test Case 3

The following test case copies the setting of Test case 2, however it includes five wells. The input parameters for the wells are gathered in Table 5.7. The wells 1,2 and 4 can be seen as the pumping wells, the others as the injection wells. Let us again denote 4 different settings with two enrichment radii:

- TC3-a: $f = 0$, $R_w = 0.8$,
- TC3-b: $f = 0$, $R_w = 2.0$,
- TC3-c: $f = U \sin(\omega x)$, $R_w = 0.8$,
- TC3-d: $f = U \sin(\omega x)$, $R_w = 2.0$.

w	$\boldsymbol{x}_w$	ρ_w	σ_w	g_{1D}
1	[2.8,2.5]	0.03	20.0	-150
2	[4.9,5.4]	0.03	10.0	-30
3	[2.9,7.4]	0.03	10.0	120
4	[7.3,7.8]	0.03	10.0	-50
5	[7.4,2.8]	0.03	20.0	100

Table 5.7: Input data for the wells in Test case 3.

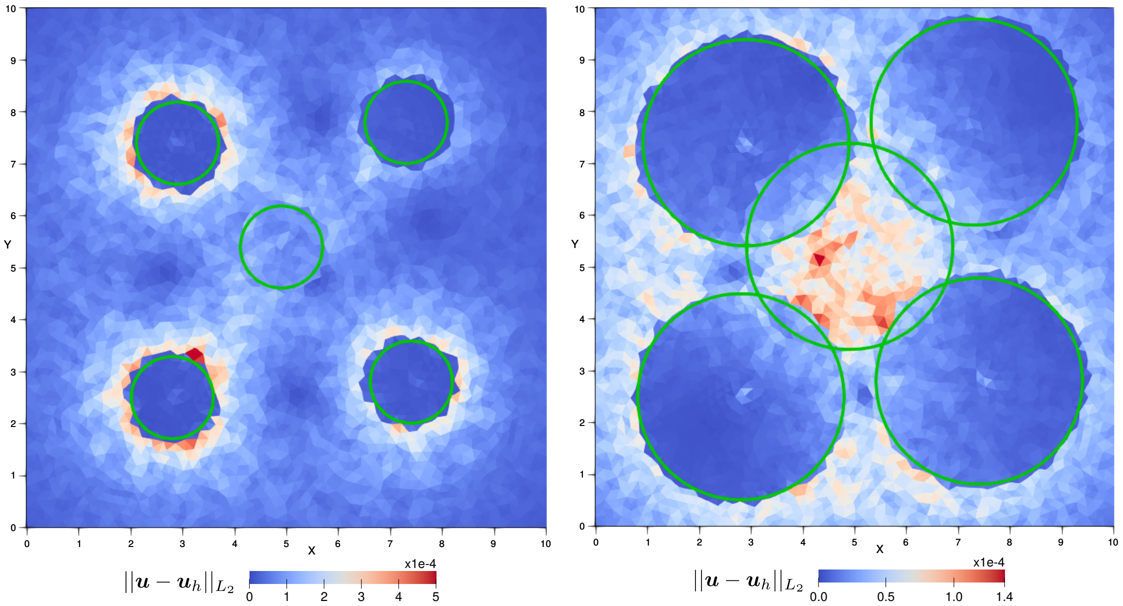
The distribution of the velocity error is shown in Figure 5.10. We can see in setting TC3-a that the error is again concentrated on the edges of the enrichment zones of the strongest singularities. In setting TC3-b, the error is largest in the central area, which is affected by all the singularities, however it is covered only by the enrichment zone of the well 2. The solution is well approximated inside the four enrichment zones of the wells 1,3,4,5, we see only small effects of the other singularities there.

In the subfigures for settings TC3-c and TC3-d, we see the same behavior of the error as in previous test case. The error of the regular part is significant in TC3-d while the error of the singular part is not apparent.

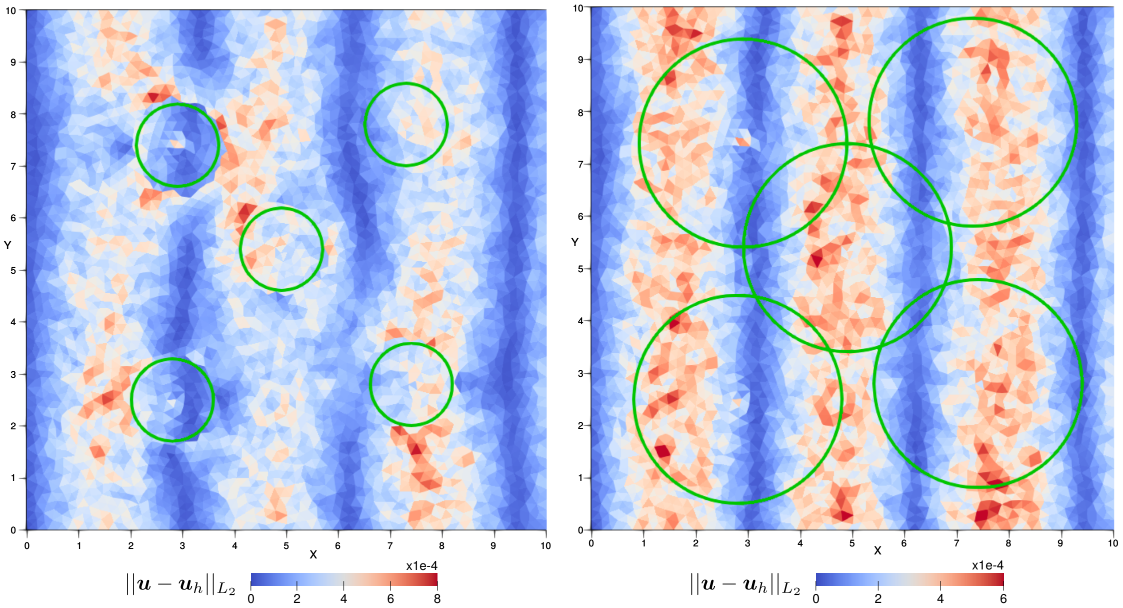
The convergence for the selected settings is displayed in Table 5.8. We see a nearly optimal convergence order in all three columns. It is again apparent that the larger enrichment radius pushes the approximation error down a bit but it does not effect the convergence order.

		TC3-a		TC3-c		TC3-d	
i	h	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order
1	1.412	$2.82 \cdot 10^{-2}$	-	$1.19 \cdot 10^{-1}$	-	$1.16 \cdot 10^{-1}$	-
2	0.946	$2.11 \cdot 10^{-2}$	0.72	$7.67 \cdot 10^{-2}$	1.10	$7.33 \cdot 10^{-2}$	1.13
3	0.650	$1.43 \cdot 10^{-2}$	1.03	$5.06 \cdot 10^{-2}$	1.11	$4.87 \cdot 10^{-2}$	1.09
4	0.431	$1.08 \cdot 10^{-2}$	0.68	$3.50 \cdot 10^{-2}$	0.89	$3.32 \cdot 10^{-2}$	0.93
5	0.292	$7.34 \cdot 10^{-3}$	1.00	$2.31 \cdot 10^{-2}$	1.06	$2.19 \cdot 10^{-2}$	1.06
6	0.193	$5.05 \cdot 10^{-3}$	0.91	$1.56 \cdot 10^{-2}$	0.96	$1.47 \cdot 10^{-2}$	0.98

Table 5.8: Convergence table of SGFEM in Test case 3.



(a) TC3-a (b) TC3-b



(c) TC3-c (d) TC3-d

Figure 5.10: Results of Test case 3. The elementwise L_2 error in velocity is displayed at refinement level 5. The green circle indicates the enrichment radius R_w.

5.6.2 Test Cases in 1d-3d

We now present several test cases in 3 dimensions. The geometries of the test cases are analogical to the previous test cases in 2d, but extruded to the third dimension in z coordinate.

Test Case 4

In the first 3d test case, the 3d domain Ω_3 is a cylinder with the bottom base in xy plane with the center at $[3.33, 3.33, 0]$, of height 2.0 and radius 5.0. A well is intersecting the cylinder along its vertical axis. Similarly to the 2d case, we set high conductivity $\boldsymbol{K}_1$ inside the well and a constant Dirichlet boundary condition g_{1D} on both ends of the well to have constant pressure there. A homogeneous Neumann boundary condition for normal flux is applied on both bases of the cylinder. The flow is then governed by the well and the Dirichlet boundary condition on the lateral surface of the cylinder. Therefore we can still use the pseudo-analytic solution from Section 4.3.

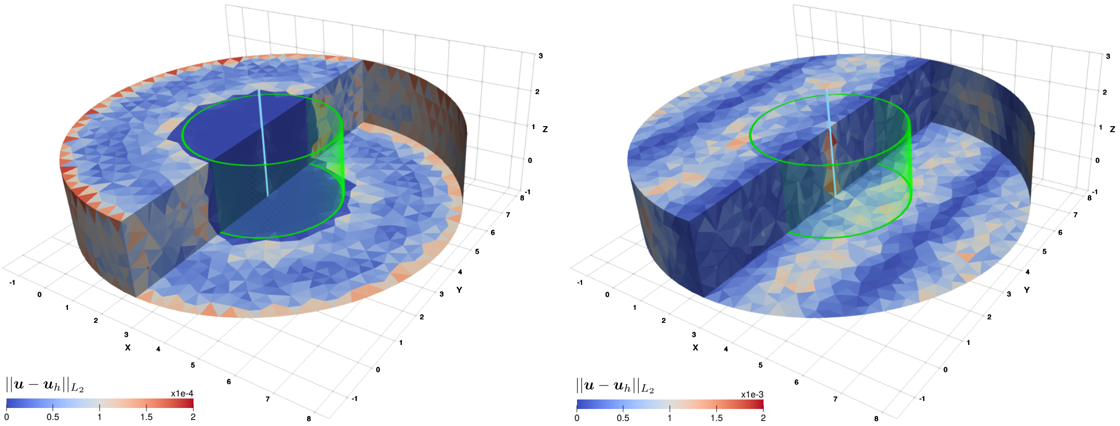
Let us have three different settings:

- TC4-a: $f = 0$,
- TC4-b: $f = U \sin(\omega x)$,
- TC4-c: $f = U \sin(\omega x)$, no singularity (regular case).

We use the same parameters for the nonzero source terms as before. The the input parameters are gathered in Table 5.9.

$\boldsymbol{K}$	ρ_w	σ_w	R_w	g_{1D}	ω	U
10^{-3}	0.03	10.0	2.0	100	1.0	80

Table 5.9: Input data for Test case 4.



(a) TC4-b

(b) TC4-d

Figure 5.11: Results of Test case 4. The elementwise L_2 error in velocity is displayed at refinement level 5. The green cylinder indicates the enrichment zone.

The approximation error of velocity is displayed in Figure 5.11. We can see that higher error is accumulated outside the edge of the enriched zone in TC4-a. However,

		TC4-a		TC4-b		TC4-c	
i	h	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order
1	1.348	$1.16 \cdot 10^{-2}$	-	$1.81 \cdot 10^{-1}$	-	$1.81 \cdot 10^{-1}$	-
2	1.063	$1.10 \cdot 10^{-2}$	0.20	$1.50 \cdot 10^{-1}$	0.79	$1.50 \cdot 10^{-1}$	0.80
3	0.845	$8.89 \cdot 10^{-3}$	0.95	$1.12 \cdot 10^{-1}$	1.27	$1.12 \cdot 10^{-2}$	1.28
4	0.632	$7.90 \cdot 10^{-3}$	0.40	$8.90 \cdot 10^{-2}$	0.79	$8.88 \cdot 10^{-2}$	0.79
5	0.544	$7.27 \cdot 10^{-3}$	0.56	$7.36 \cdot 10^{-2}$	1.28	$7.34 \cdot 10^{-2}$	1.28
6	0.408	$6.67 \cdot 10^{-3}$	0.30	$5.45 \cdot 10^{-2}$	1.04	$5.47 \cdot 10^{-2}$	1.02

Table 5.10: Convergence table in Test case 4.

we observe a significant error also along the edges of the cylinder bases where the two boundaries with the Dirichlet and the homogeneous Neumann boundary condition are adjacent. The velocity shape functions on the elements with sides on both types of boundaries are obviously unable to approximate the solution correctly there. This error on the boundary also slows down the convergence rate, as it can be seen in Table 5.10. Considering the source term in TC4-b, the error of the singular part is inferior to the error of the regular part, as it is apparent in the right subfigure. The error along the edges of the cylinder bases is negligible in contrast to TC4-a.

Test Case 5

In the second 3d test case, the domain Ω_3 is a block with the bottom base in xy plane, with height 2.0. Two wells intersecting the block are perpendicular to its base. The parameters of the wells are specified in Table 5.11.

w	$\boldsymbol{x}_w$	ρ_w	σ_w	g_{1D}
1	[4.1,4.3]	0.03	10.0	150
2	[5.7,5.9]	0.03	10.0	100

Table 5.11: Input data for the wells in Test case 5.

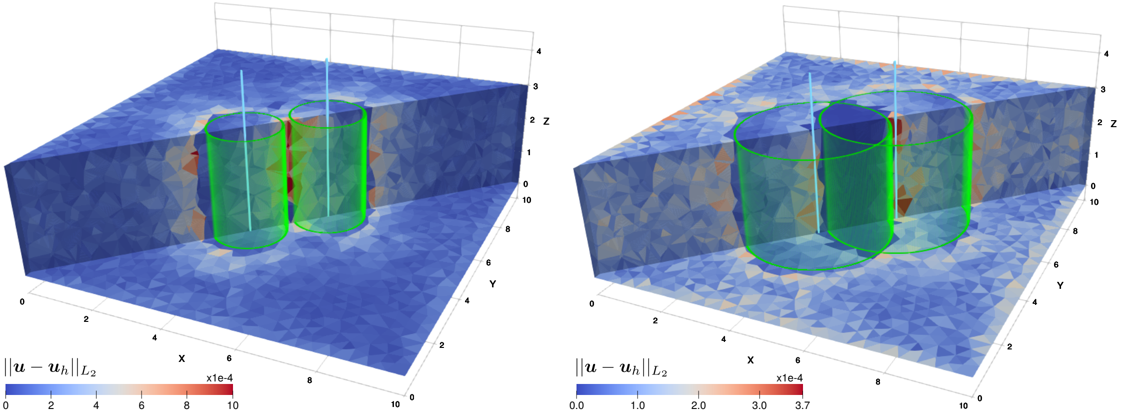
Analogically to the previous test case, the constant pressure inside the wells is enforced and zero normal flux is prescribed on both bases of the block.. We use again the pseudo-analytic solution to determine the approximation error.

Let us have four different settings:

- TC5-a: $f = 0$, $R_w = 1.0$,
- TC5-b: $f = 0$, $R_w = 2.0$,

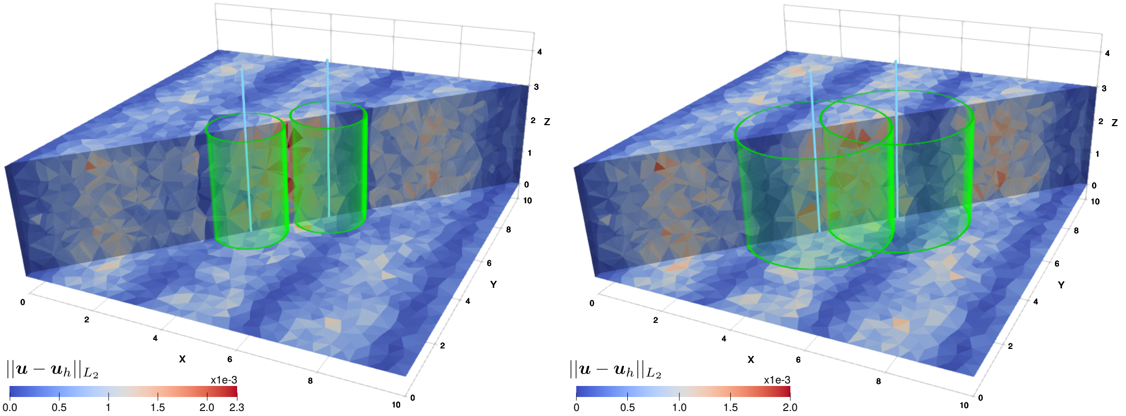
- TC5-c: $f = U\sin(\omega x)$, $R_w = 1.0$,
- TC5-d: $f = U\sin(\omega x)$, $R_w = 2.0$.

We use the same parameters for the nonzero source terms as before: $U = 80$, $\omega = 1.0$.



(a) TC5-a

(b) TC5-b



(c) TC5-c

(d) TC5-d

Figure 5.12: Results of Test case 5. The elementwise L_2 error in velocity is displayed at refinement level 5. The green cylinders indicate the enrichment zone.

We show the distribution of the error in Figure 5.12 and the convergence results in Table 5.12. In case of settings TC5-a and TC5-b, a significant error can be again observed on the edges of the block bases as in the previous case. Due to this error the convergence rate is suboptimal. On the other hand for the settings TC5-c and TC5-d, the results are satisfying and the approximation error behaves similarly as in the 2 cases. The error of the regular part is dominating as it can be seen in the subfigures. The convergence order is optimal.

The well 1 causes a stronger singularity due to the higher pressure difference, therefore the error outside the enrichment zone of the well is also higher. This is apparent especially in the subfigure TC5-b. We can also observe a very low error inside the enrichment zones overlap.

		TC5-a		TC5-c		TC5-d	
i	h	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order
1	1.404	$5.70 \cdot 10^{-2}$	-	$2.49 \cdot 10^{-1}$	-	$2.43 \cdot 10^{-1}$	-
2	1.190	$5.36 \cdot 10^{-2}$	0.37	$2.07 \cdot 10^{-1}$	1.12	$2.04 \cdot 10^{-1}$	1.05
3	0.878	$4.29 \cdot 10^{-2}$	0.73	$1.55 \cdot 10^{-1}$	0.95	$1.53 \cdot 10^{-2}$	0.93
4	0.681	$3.38 \cdot 10^{-2}$	0.95	$1.19 \cdot 10^{-1}$	1.02	$1.18 \cdot 10^{-2}$	1.04
5	0.515	$2.83 \cdot 10^{-2}$	0.63	$9.22 \cdot 10^{-2}$	0.92	$9.03 \cdot 10^{-2}$	0.96
6	0.416	$2.37 \cdot 10^{-2}$	0.83	$7.51 \cdot 10^{-2}$	0.96	$7.35 \cdot 10^{-2}$	0.96

Table 5.12: Convergence table in Test case 5.

Test Case 6

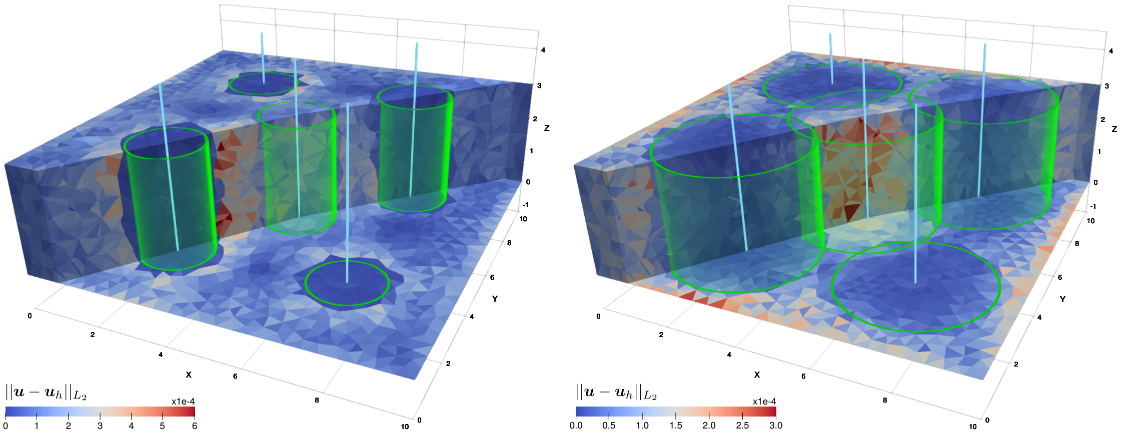
This test case is analogical to the previous test case, however it includes 5 wells. The wells are perpendicular to the xy plane and they have the same parameters as in the 2d Test Case 3. The following settings are used:

- TC6-a: $f = 0$, $R_w = 1.0$,
- TC6-b: $f = 0$, $R_w = 2.0$,
- TC6-c: $f = U\sin(\omega x)$, $R_w = 1.0$,
- TC6-d: $f = U\sin(\omega x)$, $R_w = 2.0$,
- TC6-e: $f = U\sin(\omega x)$, no singularity.

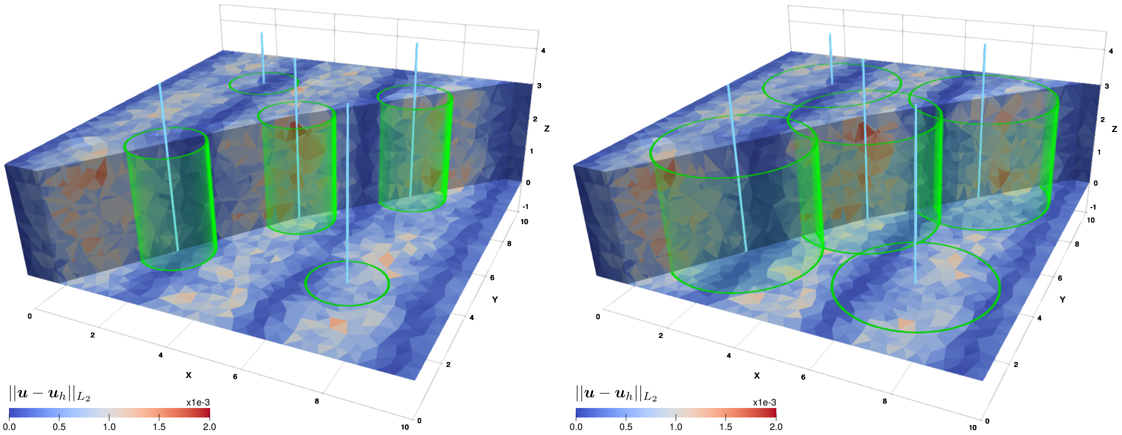
		TC6-c		TC6-d		TC6-e	
i	h	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2(\Omega_2)}$	order
1	1.404	$2.52 \cdot 10^{-1}$	-	$2.51 \cdot 10^{-1}$	-	$2.49 \cdot 10^{-1}$	-
2	1.190	$2.15 \cdot 10^{-1}$	0.96	$2.11 \cdot 10^{-1}$	1.04	$2.10 \cdot 10^{-1}$	1.03
3	0.878	$1.62 \cdot 10^{-1}$	0.93	$1.59 \cdot 10^{-1}$	0.93	$1.59 \cdot 10^{-2}$	0.92
4	0.681	$1.25 \cdot 10^{-1}$	1.01	$1.23 \cdot 10^{-1}$	1.01	$1.23 \cdot 10^{-2}$	1.02
5	0.515	$9.55 \cdot 10^{-2}$	0.98	$9.39 \cdot 10^{-2}$	0.98	$9.31 \cdot 10^{-2}$	0.99
6	0.416	$7.75 \cdot 10^{-2}$	0.97	-	-	$7.54 \cdot 10^{-2}$	0.98

Table 5.13: Convergence table in Test case 6.

The distribution of the error is displayed in Figure 5.13 and the convergence results are summarized in Table 5.13. Setting the zero source term in the settings



(a) TC6-a (b) TC6-b



(c) TC6-c (d) TC6-d

Figure 5.13: Results of Test case 6. The elementwise L_2 error in velocity is displayed at refinement level 5. The green cylinders indicate the enrichment zone.

TC6-a,b, the optimal convergence rate is again corrupted by the error on the edges of the bases of the block, although in TC6-a it is not that significant. Considering the model with the settings TC6-c,d, we see that the singularities are well approximated and the dominant error is in the regular part of the solution. The optimal convergence is comparable to the regular case TC6-e.

For the setting TC6-d, the algebraic solver failed due to insufficient memory on the finest mesh. Since most of the elements are enriched from at least one well, there are several rows relatively full which causes loss of sparsity and higher memory consumption.

Test Case 7

In this test case we want to demonstrate a more general setting including 10 wells of different tilt. All the wells are set with the following parameters: $\sigma_w = 100,\ \boldsymbol{K}_1 = 100,\ \rho_w = 0.03,\ R_w = 0.8$. Instead of fixing pressure in the wells, we prescribe fluxes by the means of the inhomogeneous Neumann boundary condition at the top of the wells. The fluxes are gathered in the first row of Table 5.14, the second row contains the fluxes scaled by the cross-section, i.e. flux density for Flow123d input. The rock block conductivity is set to 10^{-2}, a Dirichlet boundary condition $g_{3D} = 25(3-z)$ is applied on the sides and a homogeneous Neumann boundary condition for normal fluxes is set at the bases.

w	1	2	3	4	5	6	7	8	9	10
g^w_{1N}	1.414	0.848	0.565	1.414	1.131	-1.131	0.848	0.848	-1.131	0.565
$\frac{g^w_{1N}}{\delta_1}$	500	300	200	500	400	-400	300	300	-400	200

Table 5.14: Prescribed fluxes at the top of the wells in Test case 7.

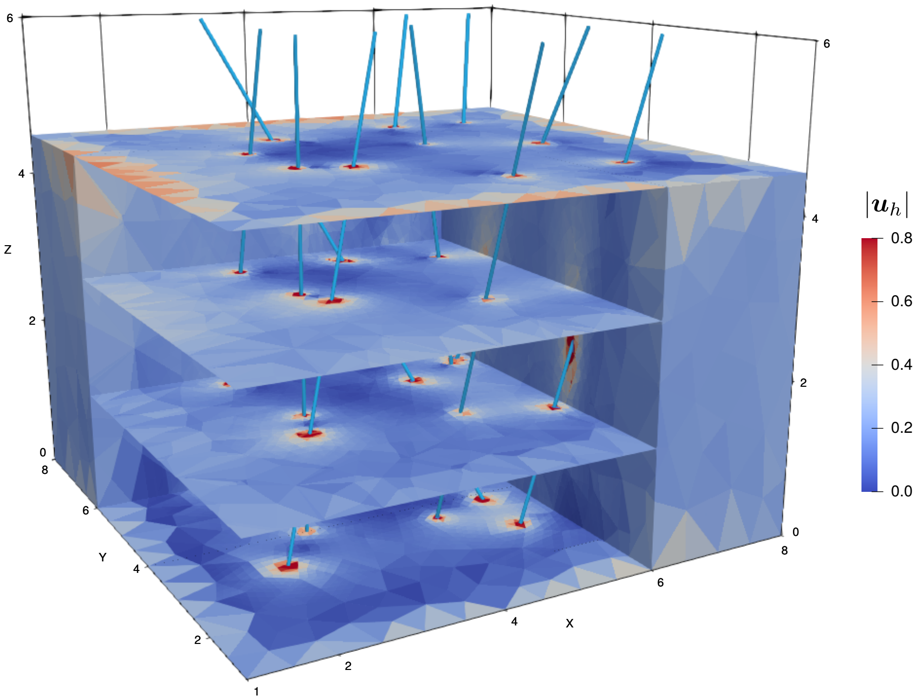


Figure 5.14: Results of Test case 7. Ten wells are represented symbolically by the blue tubes. The block is clipped so that we can see inside. The magnitude of velocity is displayed.

An analytical solution is not available in this case, we can only inspect the discrete solution qualitatively, see the solution of velocity Figure 5.14. The block is meshed regularly, the element size can be noticed along the edges of the lower base. The refined solution in the vicinity of the wells is due to the refined output mesh.

We calculate the water balance over the domain boundaries. There is zero flux on the bases of the block and also at the bottom of the wells. The sum of the fluxes at the top of the wells is equal 5.372, while the flux over the block sides is -5.376, i.e. the difference between what flows into and out of the system is 0.004. Thus, we can conclude, that the communication between the wells and the block is well approximated in terms of the water balance.

5.7 Summary

In this chapter we developed an XFEM in a mixed-hybrid form for approximation of singularities in both 2d and 3d problems. The model was formulated following the concept of the software Flow123d and using the Lagrange multipliers for dimensional coupling. We used what we have learned in Chapter 4 and transferred our experience into creating an SGFEM like enrichment of velocity in the vicinity of the singularities to improve the approximation. This enrichment was applied in the discretization together with the lowest order Raviart-Thomas FE. The properties of such enrichment were described. The numerical validation of the inf-sup stability was discussed, however this topic is left open for further research.

Regarding the reduced dimensional modeling, our approach enables coupling of non-planar 1d-2d domains and coupling of co-dimension 2 in 3d. The possibility to compute such problems on incompatible meshes is a significant advantage, as it was demonstrated even in the small artificial test cases.

The coupled models were implemented as a part of the software Flow123d. Some of the technical aspects of the code extensions were discussed, the input file, the adaptive integration and the output mesh refinement in particular. The solution of the algebraic linear system was not addressed. A direct solver or Krylov subspace iterative methods with some standard preconditioners were used. This topic would definitely deserve more attention in future, in order to solve larger problems efficiently.

At last, an extensive set of numerical tests was performed and convergence results were presented. Although pressure converges suboptimally, because its FE approximation is not enriched, velocity error reaches optimal order of convergence. Solutions with different enrichment radii and overlapping enrichment zones were compared in most of the test cases. The last test case simulated a more complex problem with multiple wells intersecting the rock in different angles.

6 Mesh Intersection Algorithms

In the previous chapter we studied the application of XFEM in a model where the reduced dimension concept and incompatible meshes are used. So far, we have not mentioned how the intersections of the meshed domains of different dimensions are computed. In our case these are the intersections of non-planar 1d-2d meshes and 1d-3d meshes. Even though we consider only straight 1d domains in our well-aquifer models, where the computation of the intersections is not overly demanding, we look at this topic from a more general perspective and dive deeper into intersection algorithms.

The motivation to solve the intersections of meshes more generally originates from our needs and future vision. The concept of the compatible meshes provides a great tool up some geometry complexity, however the compatible meshing becomes painful or even impossible as the number of separate domains (fractures and wells) increases. There are approaches available for computations on incompatible meshes, especially of codimension 1, including XFEM of course. The Chimera method [55], also called overset grid, and similarly Nitche method [56] allow solution of the problems with a changing geometry, e.g. in fluid-structure problems. The Mortar method [57, 17] allows domain decomposition, independent meshing of domains, and supports sliding boundaries. A Mortar like method [50] is also experimentally implemented in Flow123d.

Recently, the Multilevel Monte Carlo (MLMC) [58] has been developed extensively and it has become attractive also in underground processes simulations, where it is used in uncertainty quantification analysis in DFN [59, 60]. One of the principles in DFN with MLMC is a generation of the fractures randomly and there the construction of incompatible meshes is indisputably more convenient. Similarly in problems with time evolving fracture network, e.g. in modeling of hydraulic fracking, a fast and robust mesh construction is required, which is not generally satisfied in case of compatible meshes.

The prerequisite for any of these applications is a fast and robust algorithm for calculating intersections of the individual meshes. The content of this chapter includes the results published in our article [61].

6.1 Introduction to Mesh Intersection

We consider a composed mesh $\mathcal{T}$ consisting of simplicial meshes $\mathcal{T}_d$, $d \in \{1, 2, 3\}$ in the 3d ambient space. For the purpose of this chapter, we further denote a submesh

$\mathcal{T}^i \subset \mathcal{T}$, $i = 1 \dots N_{\mathcal{T}}$, a connected set of elements of the same dimension with no self intersection. We assume only a single 3d mesh $\mathcal{T}^1 = \mathcal{T}_3$. 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 a non-empty intersection and to compute that intersection. The solution of the mesh intersection problem consists of two major steps: first, generating a set of candidate pairs (K, L); second, computing the intersection of each pair.

According to our knowledge, there are lots of works using incompatible meshes, yet only a few of them discuss algorithms how to compute their intersections. Gander and Japhet [62] 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 a linear time. The algorithm is a part of the library DUNE [63]. Massing, Larson, and Logg [56] present an algorithm for 2d-3d intersections included in their implementation of the Nitche method which is a part of the project Dolfin [64]. They use the axes aligned bounding boxes of elements (AABB) and bounding interval hierarchy (BIH) to get intersection candidate pairs of elements, while the GTS library [65] is used for 2d-3d intersections. Finally, there is the work of Elsheikh and Elsheikh [66] 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.

Our approach to the mesh intersection problem is based on the Plücker coordinates, further developing the algorithm of Platis and Theoharis [67] for ray-tetrahedron intersections. The Plücker coordinates play important role in the algorithms for intersection of simplicial elements. These algorithms are combined with the advancing front method which allows us to reuse Plücker coordinates and their products among neighboring elements and to reduce the number of arithmetic operations.

At first, the algorithms for 1d-2d, 1d-3d, 2d-3d and 2d-2d intersections of simplices are described. Then we discuss our implementation of the advancing front technique and the usage of AABB and BIH for its initiation. Finally, several benchmarks are presented and the individual algorithms are compared.

6.2 Element Intersections

In this section, we present algorithms for computing the intersection of a pair of simplicial elements of a different dimension in a 3d ambient space. The fundamental idea is to compute intersection of 1d-2d simplices using the Plücker coordinates and to reduce all the other cases to this one.

We denote S_i a simplicial element with $i+1$ vertices (of dimension i). We call vertices, edges, faces and simplices themselves the n-faces and we denote 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 a reference to the intersecting simplices. A data structure of a single IC consists of:

- the barycentric coordinate $\boldsymbol{w}_K$ of IC on the element K,
- the dimension d_K of the lowest dimensional n-face the IC is lying on, e.g. an IC on an edge have $d_K = 1$ 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, since every IC is a result of a permuted inner product of some Plücker coordinates (defined below in Section 6.2.2), we store the sign of the product as well.

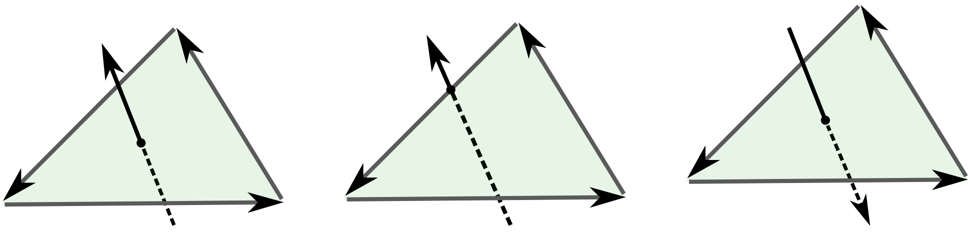
6.2.1 Plücker Coordinates

Plücker coordinates represent a line in a 3d space. The definition, properties and the more general context from computational geometry can be found e.g. in [68] or [69]. Considering a line p, given by a point $\boldsymbol{A}_p$ and its directional vector $\boldsymbol{u}_p$, the Plücker coordinates of p are defined as

$$\pi_p = (\boldsymbol{u}_p, \boldsymbol{v}_p), \quad \boldsymbol{v}_p = \boldsymbol{u}_p \times \boldsymbol{A}_p. \tag{6.1}$$

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

$$\pi_p \odot \pi_q = \boldsymbol{u}_p \cdot \boldsymbol{v}_q + \boldsymbol{u}_q \cdot \boldsymbol{v}_p. \tag{6.2}$$


(a) $\pi_p \odot \pi_{s_i} < 0 \; \forall i$

(b) $\pi_p \odot \pi_{s_i} = 0 \; \exists i$, special case

(c) $\pi_p \odot \pi_{s_i} > 0 \; \forall i$

Figure 6.1: Different relative positions of a line p and a triangle with sides s_i, $i = 0, 1, 2$. The dashed parts of p are hidden behind the triangle. The signs of the permuted inner products depend on the orientation of lines, the line p is coplanar with the side in case (b).

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 the relative position of the line p and the triangle. This is demonstrated in Figure 6.1. The permuted inner products of the triangle sides with the line p have a 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.

6.2.2 Intersection Line-Triangle (1d-2d)

Let us consider a line p with parametric equation

$$\boldsymbol{X} = \boldsymbol{A} + t\boldsymbol{u}, \tag{6.3}$$

on which a line segment S_1 is defined by $t \in [0,1]$ and a triangle S_2 given by vertices $(\boldsymbol{V}_0, \boldsymbol{V}_1, \boldsymbol{V}_2)$ with oriented sides $s_i = (\boldsymbol{V}_j, \boldsymbol{V}_k)$, $j = (i+1) \bmod 3$, $k = (i+2) \bmod 3$, see Figure 6.2.

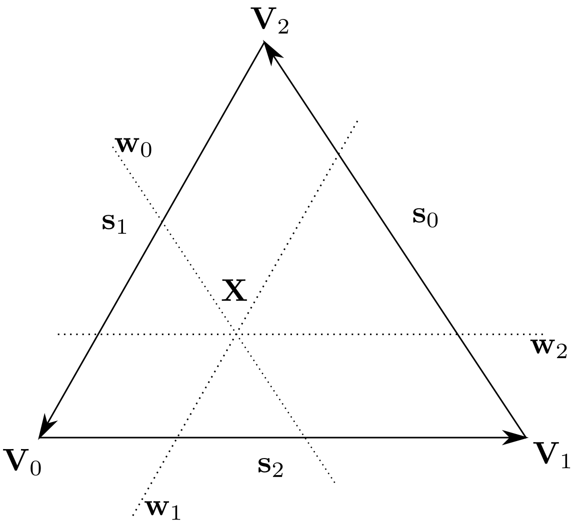


Figure 6.2: Notation for Lemma 6.2.1.

Lemma 6.2.1. *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 $\boldsymbol{X}$ on p and inside the triangle S_2. The barycentric coordinates of $\boldsymbol{X}$ on S_2 are*

$$w_i = \frac{\pi_p \odot \pi_{s_i}}{w}, \quad w = \sum_{i=0}^{2} \pi_p \odot \pi_{s_i}. \tag{6.4}$$

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

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

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

$$\begin{aligned} \pi_p \odot \pi_{s_i} &= -w_i \boldsymbol{u} \cdot (\boldsymbol{s}_1 \times \boldsymbol{s}_2), && (6.5) \\ \sum_{i=0}^{2} \pi_p \odot \pi_{s_i} &= -\boldsymbol{u} \cdot (\boldsymbol{s}_1 \times \boldsymbol{s}_2). && (6.6) \end{aligned}$$

The result (6.4) then follows directly from the combination of (6.5) and (6.6). The point $\boldsymbol{X}$ is inside S_2 if and only if $w_i > 0$ for all $i = 0, 1, 2$. □

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

$$X_i = A_i + t u_i, \ \text{for } i = 1, 2, 3 \tag{6.7}$$

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

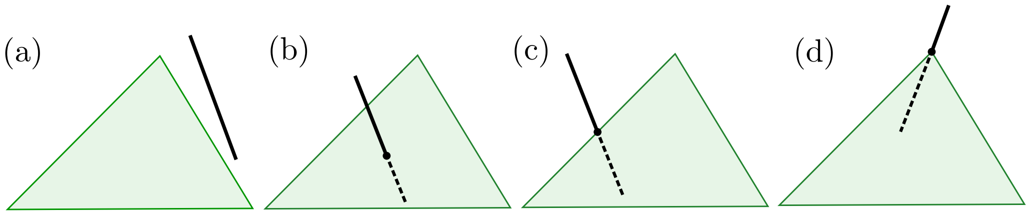


Figure 6.3: Some possible cases of the 1d-2d algorithm.

All the possible intersections of p and S_2 are displayed in Figure 6.3. Considering a given geometric tolerance ϵ, the calculation of the intersection proceeds as follows:

1. Compute or reuse the Plücker coordinates and the permuted inner products: π_p, π_{s_i}, $\pi_p \odot \pi_{s_i}$, for $i = 1, 2, 3$.
2. If the total w of the products is less then $\epsilon L_1 L_2^2$, jump to the coplanar case in the step 8. L_1 and L_2 are defined bellow.
3. Compute the barycentric coordinates w_i, $i = 1, 2, 3$ using (6.4).
4. If any w_i is less than $-\epsilon$, there is no intersection. Then an empty IP is returned. See the case (a) in Figure 6.3.
5. If all w_i are greater than ϵ, we set $\tau_{S_2} = (2, 0)$ for the IC. Figure 6.3, (b).
6. If one w_i is less than ϵ, the intersection lies on the edge s_i and we set $\tau_{S_2} = (1, i)$. Figure 6.3, (c).
7. If two w_i are less than ϵ, the intersection is at the vertex $\boldsymbol{V}_i$, we set $\tau_{S_2} = (0, i)$. Figure 6.3, (d).

8. If all w_i are less than ϵ, then 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 $\boldsymbol{V}_i$, setting the topological position τ_{S_2} to $(1, i)$ or $(0, i)$, respectively. The ICs (at most two) obtained in this coplanar case will be called *degenerate* and will later need a special treatment.

9. For each IC the barycentric coordinates $(1 - t, t)$ on the line p are computed according to (6.7).

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 the elements, we use the characteristic lengths L_1 and L_2 of S_1 and S_2 respectively. Further, the algorithm depends on the parameter ϵ which is used as a common tolerance parameter for detection of ICs with special positions. First, it is used in the sign check for the permuted inner products, second, it is used for the topological check on the line. Only these two kinds of geometric predicates are used through all the intersection algorithms. Currently, we use a 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 the evaluation of the products. Nevertheless, the algorithm works on all realistic meshes we deal with.

Other 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 Schewchuk [70] and used for 2d-2d mesh intersections in [66] in order to deal with these inconsistencies. It is a topic for future work to understand the dependency between our geometric predicates and to 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 tests in the last two steps 10-11 are not performed.

6.2.3 Intersection Line-Tetrahedron (1d-3d)

Let us now consider an intersection of a line segment S_1, defined by a parameter $t \in [0, 1]$ of the line p defined in (6.3), with a tetrahedron S_3. The algorithm is based on the 1d-2d algorithm and closely follows [67]. Our modification takes into account the intersection with the line segment and consistently propagates topological positions of ICs.

Algorithm 6.1: 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     $L$ = intersection12($p, f$)
 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 $S_1$
```

Algorithm 6.1 first computes line-face intersections for every face of S_3. A tetrahedron has six edges, so 7 Plücker coordinates and 6 permuted inner products are computed at most. The precomputed coordinates and products are passed into the 1d-2d algorithm which is performed for the infinite line p (line 3), i.e. steps 8, 10, 11 in Section 6.2.2 are not performed. If no IC is found or a coplanar case occurs in the line-face computation (line 4), we continue to 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, an IC can be on an edge (line 5) or at a vertex (line 8); 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 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 its barycentric coordinates on S_3. The topological positions τ_{S_1} and τ_{S_3} are updated as well. The result of Algorithm 6.1 is 0, 1, or 2 ICs, sorted by the parameter t in the direction of the line p.

6.2.4 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 the sides of S_2 or on the faces of S_3. Thus each vertex (IC) of the polygon can arise either from a side-face intersection, or from an 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. The computation of IP consists of three stages: a calculation of side-tetrahedron ICs, a calculation of edge-triangle ICs, an ordering of ICs.

The intersection corners are appended to the list I as they are computed, however their final order in IP is given indirectly by the *successor tables* $F_c[:]$ and $F_f[:]$. Every side of IP that lies on an n-face $x \in M_2 \cup M_3$ is followed by an IC given by $F_c[x]$. 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 an example in Figure 6.4). After an IC p is computed, we also obtain two n-faces x, y incident with the two IP sides that are connected by 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 $F_c[x] = u$, $F_f[u] = y$.

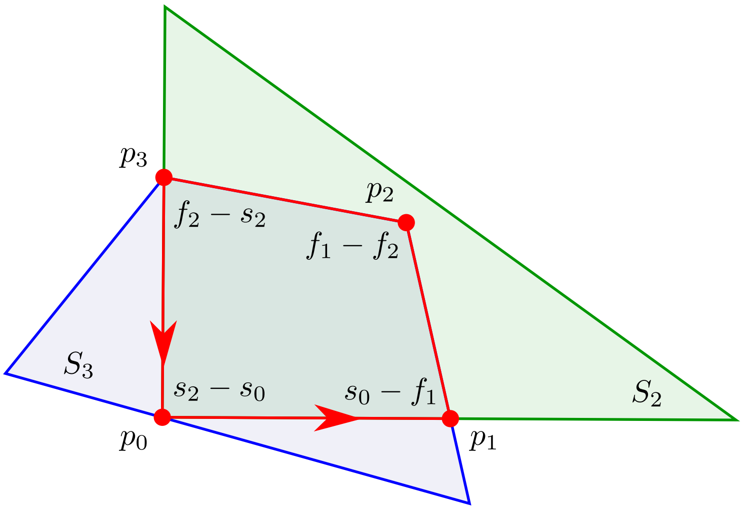


Figure 6.4: An example of an intersection 2d-3d, demonstrating the ICs ordering. We see at every intersection polygon corner p_i which n-faces it lies on. Looking at p_0, the connection table entries are: $F_c[s_2] = p_0$, $F_f[p_0] = s_0$. For the other ICs we have: $F_c[s_0] = p_1$, $F_f[p_1] = f_1$, $F_c[f_1] = p_2$, $F_f[p_2] = f_2$ and $F_c[f_2] = p_3$, $F_f[p_3] = s_2$.

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 at vertices, we further mark by a *backlink* $F_c[y] = u$ the n-faces that success some IC but still do not possess its successors. If y already have the backlink we swap x and y. The result is the *set links* (SL) operation formalized in Algorithm 6.2.

Algorithm 6.2: 2d-3d intersection, set links

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

Intersections on Sides of Triangle

Algorithm 6.3 computes all ICs on the boundary of S_2. It iterates over 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 swap n-faces if the side is already used as the target. The vertex ICs are added 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 6.4 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 the backlink and get the already computed IC if it is rediscovered later on. The ICs at vertices of S_2 are treated differently, but it follows the same idea. The ICs at vertices of S_3 are skipped.

Algorithm 6.3: 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  $F_c(:) = -1$,  $F_f(:) = -1$                          // Unset links.
2  for side $s$ of $S_2$ do
3      $L$ = intersection13 ($s, S_3$)                        // 6.1
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($x$, $p$, $s$)                        // 6.2
11         else
12             set links($y$, $p$, $x$)                      // $x$ is $s$
```

Intersections on Edges of Tetrahedron

Algorithm 6.4 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 while skipping the edges with none or a degenerate IC. For every intersection corner $p = L[e]$, we first get n-faces that would appear before and after the IC in IP.

Algorithm 6.4: 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 $L[e]$ = intersection12 ($e, S_2$)
 2 for edge $e$ of $S_3$ with regular $L[e]$ do
 3     $p = L[e]$
 4     if $p$ is inside $e$ then
 5         ($f_0$, $f_1$) = edge faces ($e$)
 6     else $p$ at the vertex $v$ of $S_3$
 7         ($f_0$, $f_1$) = vertex faces ($v$,$L$)              // 6.5
 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($x$, $p$, $f_1$)                      // 6.2
13         else
14             set links($f_0$, $p$, $x$)
15     else
16         set links($f_0$, $p$, $f_1$)
```

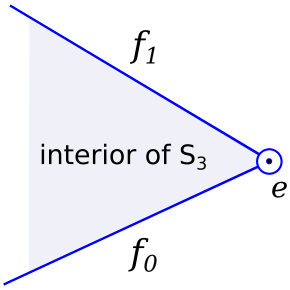


Figure 6.5: The order of faces adjacent to the oriented edge e pointing towards us.

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 Figure 6.5). 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 6.5) 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 the 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 correct face.

Algorithm 6.5: 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         $z =$ edge faces($e_a$)                              // Figure 6.5
 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 edge faces($e$)
11 else if  $s[:]$ contains edge $e$ with the sign opposite to the other two then
12     return edge faces($e$)
13 else $s[:]$ have all signs same
14     return  $(v, v)$
```

The function *vertex faces* in Algorithm 6.5 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, a degenerate IC, an IC with a positive or negative permuted inner product sign. Accordingly we say the edge is: without intersection, degenerate, 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 the interior of S_2. Possible cases are (see also Figure 6.6 a-e):

- **Single degenerate IC** (line 3**).** Let us denote e the edge with a 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 Figure 6.6, (a)) or the same sign (the plane is touching S_3 at the edge e, see Figure 6.6, (b)). In the first case, the call *edge faces*(e) returns (f_x, f) or (f, f_x), then the *vertex faces* 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.

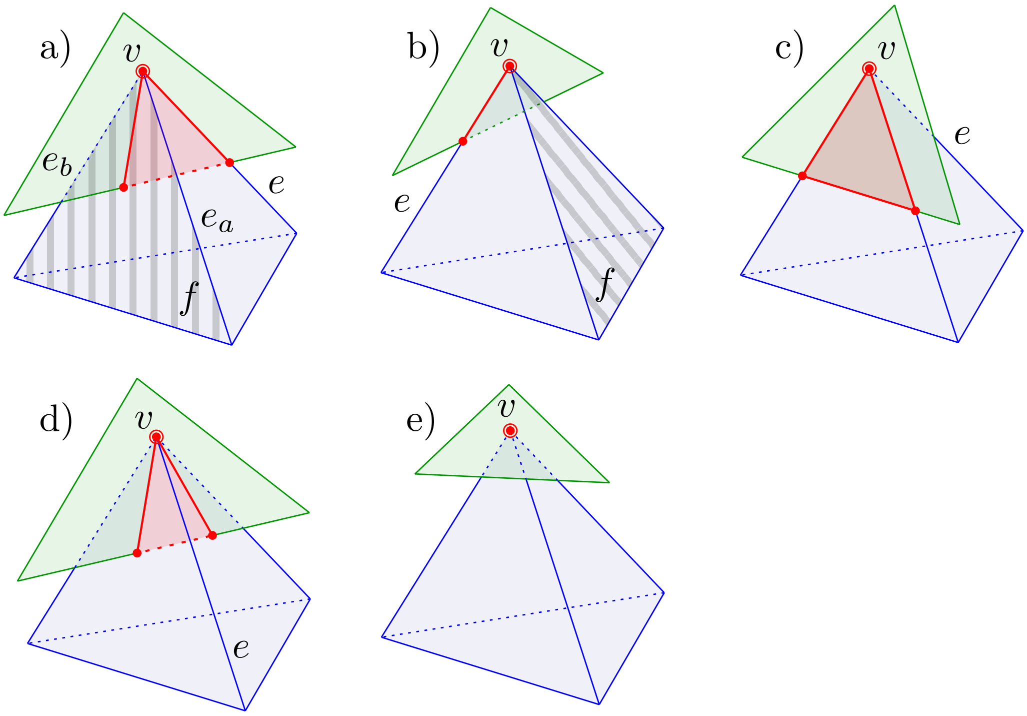


Figure 6.6: Possible cases processed in the function *vertex faces*. Only the main features referred in the text are denoted: tetrahedron vertex v, edge e, face f (stripes).

- **Two degenerate ICs** (line 9)**.** A face of S_3 lies in the plane of S_2, see Figure 6.6, (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 the 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 Figure 6.6, (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, Figure 6.6, (e), the polygon degenerates into a point and thus no connection information is necessary. We just return (v, v).

Ordering of Intersections

The final stage of the 2d-3d intersection is the ordering of ICs. We start with the first IC in I and follow the successor tables until we return back to the first IC. The ICs are copied into a resulting vector, skipping all the duplicities. Special action must be taken in degenerate cases with less than 3 ICs, since they cannot form a cycle.

6.2.5 Intersection Triangle-Triangle (2d-2d)

Let us now consider two triangles S_2^a and S_2^b, whose intersection consists of at most 2 ICs, i.e we do not assume the triangles to overlap when lying in the same plane (we do not need this in our applications). Each IC can arise from a side-triangle intersection, while we determine this intersection for each of 6 possible combinations. We have to compute 6 Plücker coordinates for the sides of both triangles and 9 permuted inner products, one per every side-side pair. In case of two ICs no sorting is performed, there is no need for that.

Algorithm 6.6: 2d-2d intersection

```
   Input: Triangles $S_2^a$ and $S_2^b$.
   Output: List $I$ of ICs.
 1 $I = \{\}$
 2 for unmarked side $s^a$ of $S_2^a$ do
 3     $L$ = intersection12($s^a, S_2^b$)
 4     for $p$ in $L$ do
 5         if $p$ is vertex $v^a$ of $S^a$ then
 6             set $\tau_{S_2^a} = (0, v^a)$
 7             mark sides of $S_2^a$ incident to $v^a$
 8         if $p$ is vertex $v^b$ of $S^b$ then mark sides of $S_2^b$ incident to $v^b$
 9         if $p$ lies on a side $s^b$ of $S^b$ then mark side $s^b$ of $S_2^b$
10     append $L$ to $I$
11     if $|I| = 2$ then return $I$
12 for unmarked side $s^b$ of $S_2^b$ do
13     $L$ = intersection12($s^b, S_2^a$)
14     for $p$ in $L$ do
15         if $p$ is vertex $v^b$ of $S^b$ then
16             set $\tau_{S_2^b} = (0, v^b)$
17             mark sides of $S_2^b$ incident to $v^b$
18     append $L$ to $I$
19     if $|I| = 2$ then return $I$
```

The computation of the triangle-triangle intersection is sketched in Algorithm 6.6. We use the 1d-2d algorithm from Section 6.2.2 as we did before, but including the degenerate case and full topology information (in contrast to the 1d-3d case). We solve all the possible side-triangle pairs in cycles at lines 4, 12, however we mark potential sides to be skipped in some special cases. We also end the computations when two ICs are computed already, lines 11, 19.

Looking at the first cycle over the sides of the first triangle, the special case from the line segment s^a perspective is an IC at one of its end points. Then the topology position must be updated to the triangle vertex, line 6. Also the other side connected to the vertex is marked so it can be skipped later. The topological position of the IC from the triangle S_2^b perspective is already known from 1d-2d algorithm. Additionally, we can mark the sides of S_2^b, if the IC is at its vertex, or mark the side s^b, if the IC lies on it.

If the algorithm reaches into the second cycle over the sides of the triangle S_2^b, we do not have to check the special cases on the triangle S_2^a, because these would have been found already. The only thing to check is whether the IC is at the end point of s_b and update its topological position in S_2^b.

6.3 Global Mesh Intersection Algorithm

Having the algorithms for all the element-element intersections at our disposal we can proceed to the mesh intersection algorithm. We consider a composed mesh $\mathcal{T}$ containing a 3d mesh $\mathcal{T}^1$ that we call a bulk mesh $\mathcal{T}_b$. Any other mesh $\mathcal{T}^i$, of lower dimension, $i = 2 \dots N_{\mathcal{T}}$, we call a component mesh. We first compute all component-bulk mesh intersections, i.e. 1d-3d and 2d-3d, using the advancing front algorithm which is described in Sections 6.3.1, 6.3.2. Then the 1d-2d and 2d-2d intersections are computed using the bulk mesh results to get the intersection pair candidates. This step is described in Section 6.3.3.

Let us now 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: first, the first pair (c, b) of the component and the bulk elements with a non-empty intersection is searched (the initiation); second, the intersection is prolonged by investigating neighboring elements (the front tracking).

6.3.1 Initiation

Given a component element c, the intersecting bulk element b has to be found. If this step is performed only a few times, the optimal way is to iterate over the bulk mesh and to test its every element. This process may be accelerated by computing the AABB for every element, then the intersection of the bounding boxes can be used as a fast indicator for possible element intersection. 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 initiation 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) [71], a data structure in principle equivalent to the R-trees [72], [73]. 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.

6.3.2 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 a breadth first search algorithm for a graph where the graph vertices are the intersection polygons and the graph edges are the IP sides shared by two IPs. Since every side of an IP is on the boundary of either a component element or a bulk element, we can distinguish bulk and component edges. Correspondingly we use a *component queue* Q_c and a *bulk queue* Q_b in which we place intersection candidate pairs (c, b). In order to process every pair (c, b) only once, we check whether 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 would allow constant time checks. Therefore, we keep a hash table of the processed pairs which allows the constant check on 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 the 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 push them into the queue Q_c. Then we increment the *component number* γ, which we use to mark all intersection polygons we find until the queue Q_c is empty. This way, we later know to which component a given IP belongs to, which will become important in Section 6.3.3. This is from where the front tracking starts, see the top-left corner of the scheme in Figure 6.7.

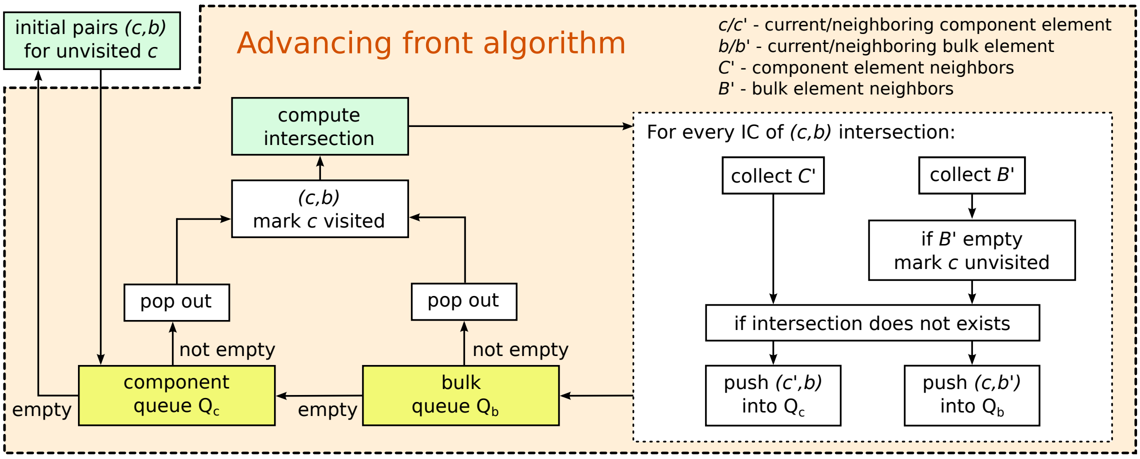


Figure 6.7: Advancing front algorithm for 1d-2d and 2d-3d intersections.

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 Figure 6.7). 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.**

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 on a single side s, i.e. branches are allowed.

(b) **IC is 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 extends out of the bulk mesh, we mark the element c as unvisited again. This way we have a chance to find possible other intersection 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 γ (see an example situation in Figure 6.8).

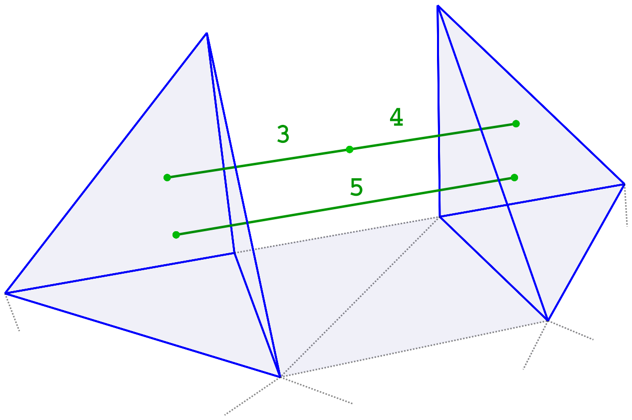


Figure 6.8: For a non-convex bulk domain a situation as this may happen. The 1d elements 3, 4, 5 extend out of the bulk mesh. Therefore, four initiations are needed to find all four 1d-3d intersections, each one of them forming an independent component. Advancing front method cannot play any part in 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.

6.3.3 Intersections Between Component Meshes

We consider here a situation where the component meshes 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 component number γ. 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, i.e. over the bulk elements having some intersections with 2d components. We look at the intersection map at the bulk element b row and collect all the 2d elements that have intersection with b. Then we create all possible pairs from the collected component elements. Now the component number γ comes into play. If the elements of a single pair have γ equal, then these are neighboring elements 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 the bulk mesh. If such problem is of our interest, we find the candidate pairs using the search algorithms as in the initiation phase of the advancing front method.

6.4 Benchmarks

In this section, we present numerical results on several benchmark problems. The theoretical number of floating point operations (FLOPs) is analyzed in our element intersection algorithms with other state of the art algorithms. Next, a comparison of our algorithms is provided, including different initiation phases (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.

The implementation of the algorithms is a part of the software Flow123d. We note, that internally in Flow123d, the open source C++ library Armadillo [74] is used for fast vector and matrix operations. This library is specialized for linear algebra in C++, with a high-level syntax deliberately similar to Matlab.

6.4.1 Theoretical Comparison

It proved to be a bit problematic to compare the presented algorithms for element-element intersections to any of the state of the 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 the estimated number of FLOPs performed by individual algorithms. Since 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 6.2.2, the algorithm based on the plane clipping due to Haines [75], and the minimum storage algorithm due to Möller and Trumbore (MT) [76]. For the later two algorithms we consider straightforward modifications to make them return a qualitatively same output as our algorithms do in 1d-2d, 1d-3d, and 2d-3d cases. The estimated numbers of FLOPs for all cases are summarized in Table 6.1. For the Plücker algorithm, we count FLOPs actually made by the implementation of individual intersection algorithms. For Haines and MT, we estimate the number of FLOPs in theoretical implementations. In particular, we consider reusing of the calculations in Plücker and Haines algorithms. Conclusions from this census are: algorithms based on the Plücker coordinates should be competitive with the state of the art algorithms in 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 results in the remaining two intersection cases when the Plücker coordinates and their products are reused by neighboring elements.

algorithm	1d-2d	1d-3d	2d-3d
Plücker	92	198	426
Plücker (edge reuse)	45	138	264
Haines	51	177	469
Möller and Trumbore	42	168	756

Table 6.1: Raw number of FLOPs used by different intersection algorithms. Second row contains the estimated effective number of FLOPs per intersection, accounting for reusing the computed Plücker data over neighboring elements, while assuming data on edges of 2d and 3d elements are used twice (conservative).

6.4.2 Global Mesh Intersections

The global mesh intersection algorithm for a composed mesh $\mathcal{T}$ presented in Section 6.3 is implemented in three variants, see Table 6.2. The first variant FS+AF uses a full search (FS) over the bulk mesh, i.e. it uses only the unordered array of AABB of elements, to get the initial pair for the advancing front algorithm (AF). The second variant uses the BIH on top of AABB to accelerate the initiation of the AF algorithm. The third variant does not use AF at all and relies on the search through BIH only.

We now compare these three variants on an artificial composed mesh and on a mesh raising from a real hydrogeological simulation. The 2d-3d intersections are considered only, the other types are not computed in the tests. Regarding the time

	FS+AF	**BIH+AF**	**BIH**
init.	AABB	BIH(AABB)	BIH(AABB)
AF	•	•	—

Table 6.2: Three variants of the global mesh algorithm.

results, the algorithms were run on a personal laptop with Intel Core i7-3630QM, 8 × 2.4 GHz CPU, 8 GB RAM, Ubuntu 16.04 LTS 64-bit system, with Flow123d compiled with O3 optimization.

Let us start with the artificial case. The composed mesh consists of a cube and two diagonal rectangular 2d meshes (see Figure 6.9). A sequence of meshes is prepared with an increasing number of elements ranging from 33 up to 2000 thousands. The mesh step of the bulk mesh is about half the mesh step of the component mesh in each realization. The number of the bulk-component intersections varies from 0.1 up to 2.0 millions.

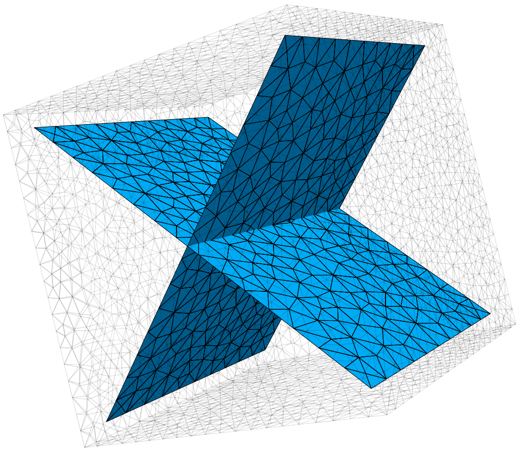
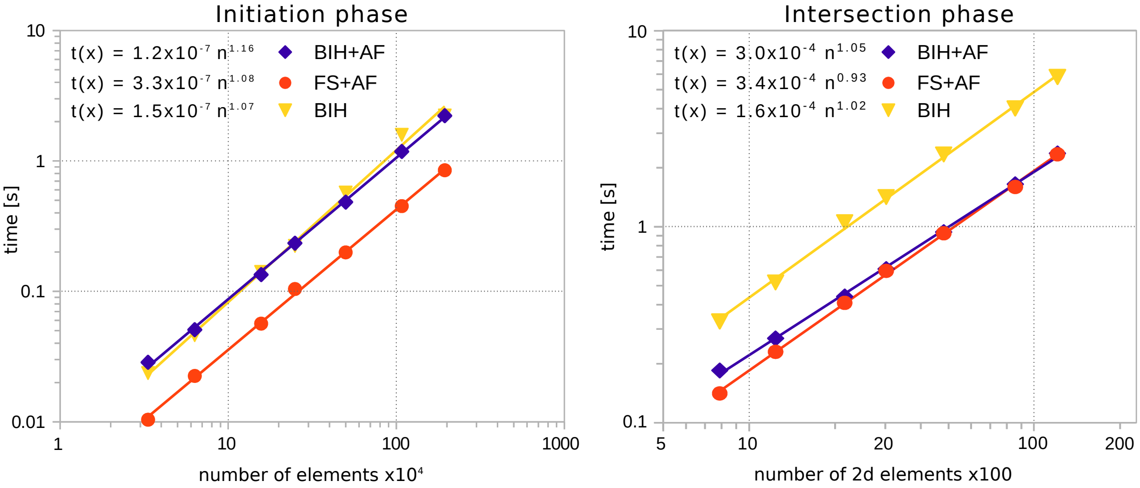


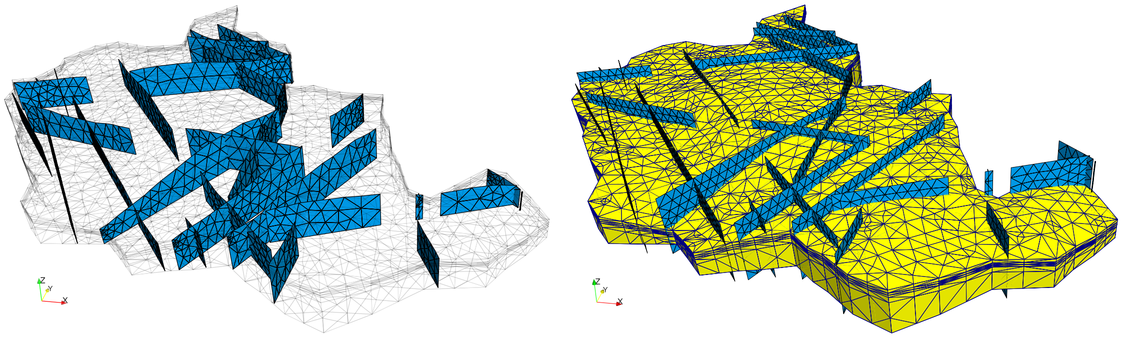
Figure 6.9: The artificial mesh: a cube with two perpendicular planes placed along its diagonals. The planes are also incompatible, therefore they can be seen as two independent components.

The time consumption of the three compared variants of the mesh intersection algorithm is shown in Graph 6.1. Every algorithm consists of the initiation phase which processes all the elements of the mesh and the intersection phase which depends only on the number of elements in the component mesh. Under these terms both phases of all three variants exhibit almost linear time complexity. Due to the low number of the component meshes which are completely covered by the bulk mesh, the FS+AF variant is the fastest one, in particular in its initiation phase. 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.



Graph 6.1: The time complexity for the initiation phase (left) with respect to the total mesh size and the intersection algorithm (right) with respect to the size of the component mesh.

Next, we study the performance of the intersection algorithms on a mesh of a real problem, see Figure 6.10. 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 (Figure 6.10a). In order to study the influence of the component elements outside of the bulk mesh, we also prepare a mesh with artificially extended fractures (Figure 6.10b). Each of these meshes contains 28 fractures, however some of them are compatibly connected so there are 9 separated 2d components and a single 1d component (the supply tunnel).



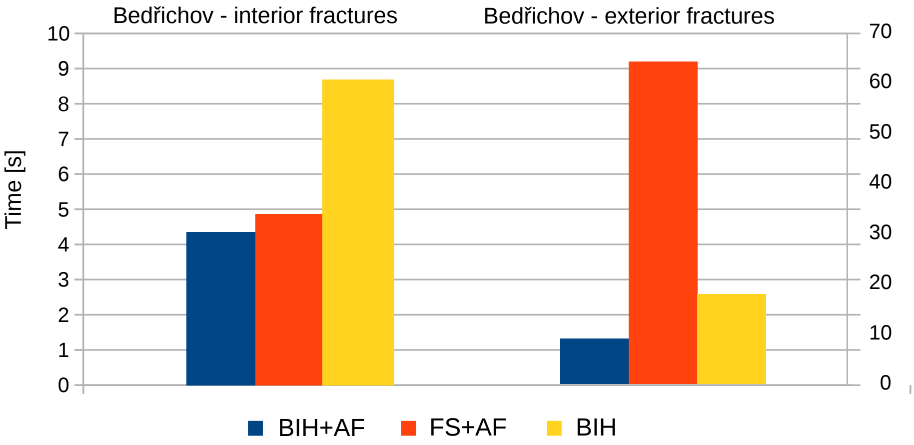
(a) interior fractures

(b) extending fractures

Figure 6.10: A mesh of the real locality of Bedřichov in the Jizera mountains. We see the fractures inside the bulk mesh on the left, the fractures are extending from the bulk mesh on the right.

The results for both meshes can be seen in Graph 6.2, pay attention to the different time scales on each side of the graph. In the first case, we notice that FS+AF and BIH+AF algorithms are nearly twice as fast as BIH. The fraction of the non-intersecting 3d elements in the bounding boxes of the 2d elements is higher as the 3d elements are smaller than the 2d elements on average. Creating the BIH in the BIH+AF variant pays off and the algorithm performs better than the FS+AF variant. This is in contrast to the cube test case since the number of the component meshes is higher.

In the second case, we observe a 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 that there is no intersection. The better performance of both BIH and BIH+AF variants is evident in this case.



Graph 6.2: Comparison of the algorithms on meshes of Bedřichov locality – interior fractures on the left, extending fractures on the right.

6.5 Summary

Algorithms for computing intersections of pairs of simplicial elements were suggested in this chapter. The algorithms are based on computation of the Plücker coordinates of the elements edges which may be reused between the elements of a mesh. The algorithms were theoretical compared by their complexity in terms of FLOPs count estimates with other state of the art algorithms. From this point of view, the suggested algorithms seem to be very competitive with the others, especially when the Plücker coordinates and the permuted inner products can be reused.

Three variants of the global mesh intersection algorithm for a composed mesh were proposed. In two of them a specialized advancing front algorithm was used. All three algorithms were tested and compared on benchmark problems, indicating which ones are the most suitable in different cases. Looking back at the results, the variant BIH+AF is recommended in general, while the FS+AF might be more efficient when there are few component meshes inside a bulk mesh.

The algorithm is a part of the software Flow123d and the results were published in the article [61]. It is currently used in the experimental Mortar model for flow and in the XFEM models presented in Chapter 5. The obtained intersections indicate the positions of the singularities and the AABB of the elements are also used when the enrichment zone is determined.

7 Conclusion

Based on the research of the related works and experience gained at the conferences (in particular MAMERN VI '15, X-DMS '15-17 and CMWR '18), we are convinced that we dedicated our efforts to a very interesting and hot topic with wide range of applications. We are also not aware of any closely connected work to this topic in the Czech Republic which puts us in a pioneer position, in the Czech scientific environment at least.

In the first part of this work, a reduced dimension concept was described and a model for coupling groundwater flow in non-planar 1d-2d and 1d-3d domains intersecting each other was suggested. The drawbacks of several approaches in FE approximation in such models were discussed leading us to our solution: incompatible meshing of the domains and using the XFEM to couple them back together and to improve FE approximation of arisen singularities by proper discrete space enrichments.

We provided an extensive study of the currently available XFEM [5, 6] in Chapter 3 and we addressed the singular enrichments in particular. Then in Chapter 4 we created a model simulating pressure in a well-aquifer system, inspired by [2, 3]. We studied different types of enrichments and compared them in terms of convergence rate, linear system conditioning and sensitivity to mesh – singularity alignment. In the view of our numerical results, we found the SGFEM to be the most promising method for singularity approximation in our model. Apart from that we focused on implementation aspects of the XFEM and we improved adaptive quadrature rules for an accurate integration on enriched elements. We also investigated the optimal enrichment zone of singular enrichments and we verified our model on a set of numerical test cases. Our early work was summarized in [33].

Chapter 5 was dedicated to the XFEM application in a mixed problem in order to extend the possibilities of the groundwater model in [9, 1]. The mixed-hybrid form was carefully derived for both non-planar 1d-2d and 1d-3d case. We suggested a new singular SGFEM like enrichment of the standard Raviart-Thomas finite elements and we applied it to the velocity discrete space. We implemented the model as an experimental part of the software Flow123d and we provided a set of numerical tests. Since velocity is important in the attached processes (e.g. transport), we put a stress on the velocity precision and we traced the velocity convergence rate. The optimal order of convergence was observed in both 1d-2d and 1d-3d tests, which included single and multiple wells, overlapping enrichment zones and non-zero source prescribed.

The difficulty of the suggested vector enrichment is that it shares a single degree of freedom per singularity over its whole enrichment zone. This is a source of two problems. First, the system matrix has some non-sparse rows, which can then lead to a loss of sparsity when applying a preconditioner based on elimination. Second, any heterogeneity, e.g. in conductivity, inside the enrichment zone cannot be captured by a single singular enrichment function. We tried to find proper elementwise enrichment functions, similarly to the ones used in the pressure model, however we struggled with the hybridized form, where such enrichment functions must be accompanied by some corresponding Lagrange multipliers. So far we were unsuccessful in performing a numerical test of the inf-sup stability of the mixed-hybrid form, this problem had to be left open. We intend to further study the suggested elementwise enrichment functions and the inf-sup test in the mixed problem without hybridization, however such model is not available in Flow123d yet.

A necessary prerequisite for computations on incompatible meshes is the ability to determine the intersections of the different meshed domains. In Chapter 6 we developed a fast and robust algorithm to compute such intersections on simplicial meshes. Although we do need to solve only the non-planar 1d-2d and 1d-3d cases in our singular models, we extended the algorithms also to higher dimensions, namely 2d-2d, 2d-3d. New models in these cases are in focus of our future work and further development of the software Flow123d. We exploited the properties of the Plücker coordinates [67, 69] in the element-element intersection algorithms which provide not only the coordinates but also additional topological information. This was then used in the global mesh intersection algorithms together with other modern techniques such as BIH of axes aligned bounding boxes and advancing front tracing. The suggested algorithms were shown to be competitive to other works [76, 75]. The global mesh intersection algorithms were tested in Flow123d on an artificial and a real case benchmarks and they exhibit linear time complexity. The results were summarized in [61]. Possible further improvements include a deeper study on the precision of the used geometric predicates, e.g. regarding the adaptivity in [70], and thorough code optimization in Flow123d.

The work was also consulted during the author's traineeship at the Technical University of Munich at the Department of Numerical Mathematics lead by Prof. Barbara Wohlmuth. Mainly the theoretical aspects of the work and new ideas were discussed. We got also familiarized with a different approach for problems with Dirac delta sources [22, 23] as a coupling method for inclusions.

The goals of this thesis as set in the introduction were fulfilled to a great extent. We studied the XFEM intensively and researched its usage in singular problems. Apart from the created pressure model, we managed to suggest a new velocity enrichment in the mixed-hybrid form and implement a working model in Flow123d. The model was derived and formulated in detail. A lot of technical work was done while preparing all the building blocks for the XFEM in the software. Eventually, we left open several issues which were addressed above.

As we already pointed out, the future work may concern a study of the vector enrichments in the mixed form. Extensions of the discretization for 1d objects that are not straight might be of interest. A specialized iterative method can be suggested in order to solve the linear algebraic system efficiently, including a proper preconditioner. Finally, some processes attached to the groundwater model may be considered using the velocity solution. These processes, namely transport of substances, poroelasticity or heat transfer, then may require similar kind of enrichment for scalar/vector quantities of interest.

Author's Publications

Publications

M. Hokr, J. Březina, J. Královcová, J. Říha, P. Exner, I. Hančilová, A. Balvín, Multidimensional Model of Flow and Transport in Fractured Rock for Support of Czech Deep Geological Repository Program, *Proceedings of 2nd International Discrete Fracture Network Engineering Conference, Seattle* (2018), ARMA-DFNE-18-1386, American Rock Mechanics Association.
`https://www.onepetro.org/conference-paper/ARMA-DFNE-18-1386`.

J. Březina, P. Exner, Fast algorithms for intersection of non-matching grids using Plücker coordinates, *Computers and Mathematics with Applications* 74 (2017) pp. 174-187. ISSN 0898-1221. `doi.org/10.1016/j.camwa.2017.01.028`.

P. Exner, J. Březina, Partition of unity methods for approximation of point water sources in porous media, *Applied Mathematics and Computation* 273 (2016) pp. 21-32. ISSN 0096-3003. `doi:10.1016/j.amc.2015.09.048`.

P. Exner, J. Březina, Adaptive integration of singularity in partition of unity methods, in proceedings of *Seminar on Numerical Analysis 2015*, Institute of Geonics AS CR, Ostrava, (2015) pp. 29-32. ISBN 978-80-86407-55-5.
`http://www.ugn.cas.cz/actually/event/2015/sna/sna-sbornik.pdf`.

P. Exner, Partition of unity methods for approximation of point water sources in porous media, in proceedings of *Seminar on Numerical Analysis 2014*, Institute of Computer Science AS CR, Prague, (2014) pp. 29-32. ISBN 978-80-87136-16-4.
`http://www.cs.cas.cz/sna2014/sbornik.pdf`.

Software

J. Březina, J. Stebel, D. Flanderka, P. Exner, J. Hybš, software Flow123d, Technical University of Liberec, (2013–2019). `http://flow123d.github.com`.

Bibliography

[1] J. Březina, J. Stebel, D. Flanderka, P. Exner, J. Hybš, Flow123d (2011–2019). URL: `http://flow123d.github.com`

[2] R. Gracie, J. R. Craig, Modelling well leakage in multilayer aquifer systems using the extended finite element method, *Finite Elements in Analysis and Design* 46 (6) (2010) pp. 504–513. ISSN 0168-874X. `doi:10.1016/j.finel.2010.01.006`.

[3] J. R. Craig, R. Gracie, Using the extended finite element method for simulation of transient well leakage in multilayer aquifers, *Advances in Water Resources* 34 (9) (2011) pp. 1207–1214. ISSN 0309-1708. `doi:10.1016/j.advwatres.2011.04.004`.

[4] T.-P. Fries, A corrected XFEM approximation without problems in blending elements, *International Journal for Numerical Methods in Engineering* 75 (5) (2008) pp. 503–532. ISSN 1097-0207. `doi:10.1002/nme.2259`.

[5] T.-P. Fries, T. Belytschko, The extended/generalized finite element method: An overview of the method and its applications, *International Journal for Numerical Methods in Engineering* 84 (3) (2010) pp. 253–304. ISSN 1097-0207. `doi:10.1002/nme.2914`.

[6] I. Babuška, U. Banerjee, Stable generalized finite element method (SGFEM), *Computer Methods in Applied Mechanics and Engineering* 201–204 (2012) pp. 91–111. ISSN 0045-7825. `doi:10.1016/j.cma.2011.09.012`.

[7] V. Gupta, C. A. Duarte, I. Babuška, U. Banerjee, A stable and optimally convergent generalized FEM (SGFEM) for linear elastic fracture mechanics, *Computer Methods in Applied Mechanics and Engineering* 266 (2013) pp. 23–39. ISSN 0045-7825. `doi:10.1016/j.cma.2013.07.010`.

[8] J. Březina, M. Hokr, Mixed-Hybrid Formulation of Multidimensional Fracture Flow, in: Numerical Methods and Applications, no. 6046 in Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010, pp. 125–132. ISBN 978-3-642-18465-9 978-3-642-18466-6. `doi:10.1007/978-3-642-18466-6_14`.

[9] J. Šístek, J. Březina, B. Sousedík, BDDC for mixed-hybrid formulation of flow in porous media with combined mesh dimensions, *Numerical Linear Algebra with Applications* 22 (6) (2015) pp. 903–929. ISSN 10705325. `doi:10.1002/nla.1991`.

[10] J. Březina, et. al., Flow123d, version 2.2.1 – user guide and input reference (2018). URL: `https://flow.nti.tul.cz/packages/2.2.1_release/flow123d_2.2.1_doc.pdf`

[11] L. Jing, O. Stephansson, 10 - discrete fracture network (dfn) method, in: L. Jing, O. Stephansson (Eds.), Fundamentals of Discrete Element Methods for Rock Engineering, Vol. 85 of *Developments in Geotechnical Engineering*, Elsevier, 2007, pp. 365 – 398. `doi:https://doi.org/10.1016/S0165-1250(07)85010-3`.

[12] Q. Lei, J.-P. Latham, C.-F. Tsang, The use of discrete fracture networks for modelling coupled geomechanical and hydrological behaviour of fractured rocks, *Computers and Geotechnics* 85 (2017) pp. 151 – 176. ISSN 0266-352X. `doi:https://doi.org/10.1016/j.compgeo.2016.12.024`.

[13] 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. ISBN 978-3-540-87921-3. `doi:10.1007/978-3-540-87921-3_18`.

[14] V. Martin, J. Jaffré, J. 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 1095-7197. `doi:10.1137/S1064827503429363`.

[15] A. Fumagalli, A. Scotti, Numerical modelling of multiphase subsurface flow in the presence of fractures, *Communications in Applied and Industrial Mathematics* 3 (1). ISSN 2038-0909. `doi:10.1685/journal.caim.380`.

[16] C. D'Angelo, A. Scotti, A mixed finite element method for Darcy flow in fractured porous media with non-matching grids, *ESAIM: Mathematical Modelling and Numerical Analysis* 46 (2) (2012) pp. 465–489. ISSN 0764-583X, 1290-3841. `doi:10.1051/m2an/2011148`.

[17] G. Pichot, J. Erhel, J.-R. de Dreuzy, A generalized mixed hybrid mortar method for solving flow in stochastic discrete fracture networks, *SIAM Journal on scientific computing* 34 (1) (2012) pp. B86–B105. `doi:10.1137/100804383`.

[18] 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. HPCSE 2015, Vol. 9611 of *Lecture Notes in Computer Science*, Springer International Publishing, Cham, 2016, pp. 152–160. ISBN 978-3-319-40361-8. `doi:10.1007/978-3-319-40361-8_11`.

[19] M. G. Trefry, C. Muffels, FEFLOW: A finite-element ground water flow and transport modeling tool, *Ground Water* 45 (5) (2007) pp. 525–528. ISSN 1745-6584. `doi:10.1111/j.1745-6584.2007.00358.x`.

[20] C. D'Angelo, A. Quarteroni, On the Coupling of 1d and 3d Diffusion-reaction equations: Application to Tissue Perfusion Problems, *Mathematical Models and Methods in Applied Sciences* 18 (08) (2008) pp. 1481–1504. ISSN 0218-2025, 1793-6314. `doi:10.1142/S0218202508003108`.

[21] L. Cattaneo, P. Zunino, Numerical Investigation of Convergence Rates for the FEM Approximation of 3d-1d Coupled Problems, in: Numerical Mathematics and Advanced Applications - ENUMATH 2013, no. 103 in Lecture Notes in Computational Science and Engineering, Springer International Publishing, 2015, pp. 727–734. ISBN 978-3-319-10704-2 978-3-319-10705-9. `doi:10.1007/978-3-319-10705-9_72`.

[22] T. T. Köppl, Multi-scale modeling of flow and transport processes in arterial networks and tissue, Ph.D. thesis, Technical University of Munich, Germany (2015).

[23] T. Köppl, E. Vidotto, B. Wohlmuth, P. Zunino, Mathematical modeling, analysis and numerical approximation of second-order elliptic problems with inclusions, *Mathematical Models and Methods in Applied Sciences* 28 (05) (2018) pp. 953–978. `doi:10.1142/S0218202518500252`.

[24] I. Babuška, J. M. Melenk, The partition of unity method, *International Journal for Numerical Methods in Engineering* 40 (4) (1997) pp. 727–758. ISSN 1097-0207. `doi:10.1002/(SICI)1097-0207(19970228)40:4<727::AID-NME86>3.0.CO;2-N`.

[25] T. Strouboulis, K. Copps, I. Babuška, The generalized finite element method: an example of its implementation and illustration of its performance, *International Journal for Numerical Methods in Engineering* 47 (8) (2000) pp. 1401–1417. ISSN 1097-0207. `doi:10.1002/(SICI)1097-0207(20000320)47:8<1401::AID-NME835>3.0.CO;2-8`.

[26] N. Moës, J. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing, *International Journal for Numerical Methods in Engineering* 46 (1) (1999) pp. 131–150. `doi:10.1002/(SICI)1097-0207(19990910)46:1<131::AID-NME726>3.0.CO;2-J`.

[27] H. Sauerland, T.-P. Fries, The stable XFEM for two-phase flows, *Computers & Fluids* 87 (2013) pp. 41–49. ISSN 0045-7930. `doi:10.1016/j.compfluid.2012.10.017`.

[28] S. Natarajan, Enriched finite element methods: Advances and applications, Ph.D. thesis, Cardiff University, United Kingdom (Jun. 2011).

[29] S. Abbas, A. Alizada, T.-P. Fries, The xfem for high-gradient solutions in convection-dominated problems, *International Journal for Numerical Methods in Engineering* 82 (8) (2010) pp. 1044–1072. `doi:10.1002/nme.2815`.

[30] P. Laborde, J. Pommier, Y. Renard, M. Salaün, High-order extended finite element method for cracked domains, *International Journal for Numerical Methods in Engineering* 64 (3) (2005) pp. 354–381. ISSN 1097-0207. `doi:10.1002/nme.1370`.

[31] E. Béchet, H. Minnebo, N. Moës, B. Burgardt, Improved implementation and robustness study of the x-fem for stress analysis around cracks, *International Journal for Numerical Methods in Engineering* 64 (8) (2005) pp. 1033–1056. ISSN 1097-0207. `doi:10.1002/nme.1386`.

[32] V. Gupta, C. A. Duarte, On the enrichment zone size for optimal convergence rate of the generalized/extended finite element method, *Computers and Mathematics with Applications* 72 (3) (2016) pp. 481 – 493. ISSN 0898-1221. `doi:https://doi.org/10.1016/j.camwa.2016.04.043`.

[33] P. Exner, J. Březina, Partition of unity methods for approximation of point water sources in porous media, *Applied Mathematics and Computation* 273 (2016) pp. 21–32. ISSN 0096-3003. `doi:http://dx.doi.org/10.1016/j.amc.2015.09.048`.

[34] G. Ventura, R. Gracie, T. Belytschko, Fast integration and weight function blending in the extended finite element method, *International Journal for Numerical Methods in Engineering* 77 (1) (2009) pp. 1–29. ISSN 1097-0207. `doi:10.1002/nme.2387`.

[35] Q. Zhang, I. Babuška, U. Banerjee, Robustness in stable generalized finite element methods (sgfem) applied to poisson problems with crack singularities, *Computer Methods in Applied Mechanics and Engineering* 311 (2016) pp. 476 – 502. ISSN 0045-7825. `doi:https://doi.org/10.1016/j.cma.2016.08.019`.

[36] V. Gupta, C. Duarte, I. Babuška, U. Banerjee, Stable gfem (sgfem): Improved conditioning and accuracy of gfem/xfem for three-dimensional fracture mechanics, *Computer Methods in Applied Mechanics and Engineering* 289 (2015) pp. 355 – 386. ISSN 0045-7825. `doi:https://doi.org/10.1016/j.cma.2015.01.014`.

[37] A. Fumagalli, Numerical modelling of flows in fractured porous media by the XFEM method, Ph.D. thesis, Polytechnic University of Milan, Italy (2012).

[38] A. Fumagalli, A. Scotti, An Efficient XFEM Approximation of Darcy Flows in Arbitrarily Fractured Porous Media, *Oil & Gas Science and Technology – Revue d'IFP Energies nouvelles* 69 (4) (2014) pp. 555–564. ISSN 1294-4475, 1953-8189. `doi:10.2516/ogst/2013192`.

[39] N. Schwenck, An XFEM-based model for fluid flow in fractured porous media, Ph.D. thesis, University of Stuttgart, Germany (2015).

[40] N. Schwenck, B. Flemisch, R. Helmig, B. I. Wohlmuth, Dimensionally reduced flow models in fractured porous media: crossings and boundaries, *Computational Geosciences* 19 (6) (2015) pp. 1219–1230. ISSN 1573-1499. `doi:10.1007/s10596-015-9536-1`.

[41] P. Díez, R. Cottereau, S. Zlotnik, A stable extended FEM formulation for multi-phase problems enforcing the accuracy of the fluxes through Lagrange multipliers, *International Journal for Numerical Methods in Engineering* 96 (5) (2013) pp. 303–322. ISSN 1097-0207. `doi:10.1002/nme.4554`.

[42] J. Nečas, Direct Methods in the Theory of Elliptic Equations, Springer Monographs in Mathematics, Springer-Verlag, Berlin Heidelberg, 2012. ISBN 978-3-642-10454-1, 978-3-642-10455-8. `doi:10.1007/978-3-642-10455-8`.

[43] L. Evans, Partial Differential Equations: Second Edition, Graduate studies in mathematics, American Mathematical Society, Providence, R.I., 2010. ISBN 9780821849743.

[44] D. Kahaner, C. B. Moler, S. Nash, G. E. Forsythe, Numerical methods and software, Prentice Hall, Englewood Cliffs, New Jersey, 07632, 1989. ISBN 0-13-627258-4.

[45] W. Bangerth, R. Hartmann, G. Kanschat, Deal.II—a general-purpose object-oriented finite element library, *ACM Trans. Math. Softw.* 33 (4). ISSN 0098-3500. `doi:10.1145/1268776.1268779`.

[46] A. Ern, J.-L. Guermond, Evaluation of the condition number in linear systems arising in finite element approximations, *ESAIM: Mathematical Modelling and Numerical Analysis* 40 (1) (2006) pp. 29–48. `doi:10.1051/m2an:2006006`.

[47] P. G. Ciarlet, Basic error estimates for elliptic problems, in: Finite Element Methods (Part 1), Vol. 2 of *Handbook of Numerical Analysis*, Elsevier, 1991, pp. 17–351. ISBN 1570-8659. `doi:10.1016/S1570-8659(05)80039-0`.

[48] F. Brezzi, M. Fortin (Eds.), Mixed and Hybrid Finite Element Methods, Vol. 15 of *Springer Series in Computational Mathematics*, Springer, New York, 1991. ISBN 978-1-4612-7824-5 978-1-4612-3172-1.

[49] 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) (1995) pp. 383–392. ISSN 0377-0427. `doi:10.1016/0377-0427(95)00066-6`.

[50] J. Březina, Mortar-like mixed-hybrid methods for elliptic problems on complex geometries, in: Conference proceedings of Algoritmy 2012, Slovak University of Technology in Bratislava, 2012, pp. 200–208. ISBN 978-80-227-3742-5.

[51] L. C. Cowsar, J. Mandel, M. F. Wheeler, Balancing domain decomposition for mixed finite elements, *Mathematics of Computation* 64 (211) (1995) pp. 989–1015. ISSN 00255718, 10886842. `doi:10.2307/2153480`.

[52] D. Chapelle, K. Bathe, The inf-sup test, *Computers and Structures* 47 (4) (1993) pp. 537–545. ISSN 0045-7949. `doi:10.1016/0045-7949(93)90340-J`.

[53] J. Bey, Simplicial grid refinement: on freudenthal's algorithm and the optimal number of congruence classes, *Numerische Mathematik* 85 (1) (2000) pp. 1–29. ISSN 0945-3245. `doi:10.1007/s002110050475`.

[54] J. Brandts, S. Korotov, M. Křížek, A geometric toolbox for tetrahedral finite element partitions, in: Efficient Preconditioned Solution Methods for Elliptic Partial Differential Equations, Bentham Science Publishers Ltd., 2011, pp. 103–122. ISBN 9781608056101. `doi:10.2174/978160805291211101010103`.

[55] F. Brezzi, J.-L. Lions, O. Pironneau, Analysis of a chimera method, *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics* 332 (7) (2001) pp. 655–660. ISSN 0764-4442. `doi:10.1016/S0764-4442(01)01904-8`.

[56] A. Massing, M. G. Larson, A. Logg, Efficient implementation of finite element methods on nonmatching and overlapping meshes in three dimensions, *SIAM Journal on Scientific Computing* 35 (1) (2013) pp. C23–C47. ISSN 1064-8275, 1095-7197. `doi:10.1137/11085949X`.

[57] F. B. Belgacem, The mortar finite element method with lagrange multipliers, *Numerische Mathematik* 84 (2) (1999) pp. 173–197. ISSN 0945-3245. `doi:10.1007/s002110050468`.

[58] M. B. Giles, Multilevel Monte Carlo methods, *Acta Numerica* 24 (2015) pp. 259–328. ISSN 0962-4929, 1474-0508. `doi:10.1017/S096249291500001X`.

[59] S. Berrone, C. Canuto, S. Pieraccini, S. Scialò, Uncertainty quantification in discrete fracture network models: Stochastic fracture transmissivity, *Computers and Mathematics with Applications* 70 (4) (2015) pp. 603 – 623. ISSN 0898-1221. `doi:https://doi.org/10.1016/j.camwa.2015.05.013`.

[60] S. Berrone, C. Canuto, S. Pieraccini, S. Scialo, Uncertainty quantification in discrete fracture network models: Stochastic geometry, *Water Resources Research* 54 (2) (2018) pp. 1338–1352. `doi:10.1002/2017wr021163`.

[61] J. Březina, P. Exner, Fast algorithms for intersection of non-matching grids using plücker coordinates, *Computers & Mathematics with Applications* 74 (2017) pp. 174 – 187. ISSN 0898-1221. `doi:10.1016/j.camwa.2017.01.028`.

[62] M. J. Gander, C. Japhet, Algorithm 932: PANG: Software for nonmatching grid projections in 2d and 3d with linear complexity, *ACM Transactions on Mathematical Software* 40 (1) (2013) pp. 1–25. ISSN 00983500. `doi:10.1145/2513109.2513115`.

[63] P. Bastian, M. Droske, C. Engwer, R. Klöfkorn, T. Neubauer, M. Ohlberger, M. Rumpf, Towards a unified framework for scientific computing, in: Domain Decomposition Methods in Science and Engineering, Vol. 40 of *Lecture Notes in Computational Science and Engineering*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 167–174. ISBN 978-3-540-26825-3. `doi:10.1007/3-540-26825-1_13`.

[64] 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. ISBN 978-3-642-23099-8. `doi:10.1007/978-3-642-23099-8_10`.

[65] Gts, gnu triangulated surface library, software package., `http://gts.sourceforge.net/.`

[66] A. H. Elsheikh, M. Elsheikh, A reliable triangular mesh intersection algorithm and its application in geological modelling, *Engineering with Computers* 30 (1) (2012) pp. 143–157. ISSN 1435-5663. `doi:10.1007/s00366-012-0297-3`.

[67] N. Platis, T. Theoharis, Fast ray-tetrahedron intersection using plucker coordinates, *Journal of graphics tools* 8 (4) (2003) pp. 37–48. `doi:10.1080/10867651.2003.10487593`.

[68] L. Dorst, D. Fontijne, S. Mann, Chapter 12 - applications of the homogeneous model, in: Geometric Algebra for Computer Science (Revised Edition), The Morgan Kaufmann Series in Computer Graphics, Morgan Kaufmann, Boston, 2009, pp. 327 – 354. ISBN 978-0-12-374942-0. `doi:https://doi.org/10.1016/B978-0-12-374942-0.00017-8`.

[69] M. Joswig, T. Theobald, Plücker coordinates and lines in space, in: Polyhedral and Algebraic Methods in Computational Geometry, Springer London, London, 2013, pp. 193–207. ISBN 978-1-4471-4817-3. `doi:10.1007/978-1-4471-4817-3_12`.

[70] J. Richard Shewchuk, Adaptive precision floating-point arithmetic and fast robust geometric predicates, *Discrete & Computational Geometry* 18 (3) (1997) pp. 305–363. ISSN 1432-0444. `doi:10.1007/PL00009321`.

[71] 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. ISBN 3-905673-35-5. `doi:10.2312/EGWR/EGSR06/139-149`.

[72] A. Guttman, R-trees: A dynamic index structure for spatial searching, *SIGMOD Rec.* 14 (2) (1984) pp. 47–57. ISSN 0163-5808. `doi:10.1145/971697.602266`.

[73] 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, pp. 171–. ISBN 0-7695-2146-6. `doi:10.1109/SSDBM.2004.1`.

[74] C. Sanderson, R. Curtin, Armadillo: a template-based C++ library for linear algebra, *The Journal of Open Source Software* 1 (2). `doi:10.21105/joss.00026`.

[75] E. Haines, V.1 - fast ray – convex polyhedron intersection, in: Graphics Gems II, Morgan Kaufmann, San Diego, 1991, pp. 247 – 250. ISBN 978-0-08-050754-5. `doi:10.1016/B978-0-08-050754-5.50053-0`.

[76] T. Möller, B. Trumbore, Fast, minimum storage ray-triangle intersection, *Journal of Graphics Tools* 2 (1) (1997) pp. 21–28. ISSN 1086-7651. `doi:10.1080/10867651.1997.10487468`.