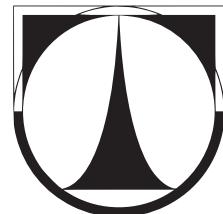


TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií



DIPLOMOVÁ PRÁCE

Liberec 2013

Pavel Exner

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: N3901 - Aplikované vědy v inženýrství

Studijní obor: 3901T025 - Přírodovědné inženýrství

Metody rozkladu jednotky pro approximaci bodových zdrojů vody v porézním prostředí

**Partition of unity methods for approximation of
point water sources in porous media**

Diplomová práce

Autor:

Bc. Pavel Exner

Vedoucí práce:

Mgr. Jan Březina, Ph.D.

Zadání

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

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

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

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum: 25. června 2013

Podpis:

Poděkování

Mé poděkování patří především Mgr. Janu Březinovi, Ph.D., vedoucímu diplomové práce, za odborné vedení a také za čas a zájem, který mi věnoval.

Dále děkuji Mgr. Janu Stebelovi, Ph.D. za cenné rady a pomoc v pochopení některých souvislostí.

Děkuji také prof. Dr. Ing. Jiřímu Maryškovi, CSc. za podporu a podnětné připomínky.

Bc. Pavel Exner

Abstrakt

V modelech proudění podzemních vod jsou často zahrnuty rozsáhlé oblasti, vůči kterým se zdroje jeví jako bodové. Příkladem může být území s hydrogeologickými vrty či území, kde probíhala důlní činnost. Rozloha oblasti vzhledem k rozměrům vrtů se může lišit o několik řádů. Metoda konečných prvků nedokáže dostatečně přesně approximovat tlak a proudové pole v blízkém okolí vrtů.

Hlavním cílem této práce je použití moderní metody rozšířených konečných prvků (XFEM), která dokáže kompenzovat chybu approximace lineárními konečnými prvky a poskytnout přesnější řešení než metoda konečných prvků.

V práci je popsán model porézního proudění ve vícevodňovém systému s vrty, je představena jeho matematická formulace a ukázána existence jednoznačného slabého řešení metody konečných prvků. Dále je model řešen pomocí metody lineárních konečných prvků na adaptivně zjemňované síti. Následně je předvedeno použití metody XFEM a její porovnání s předchozí metodou na několika úlohách.

Klíčová slova:

vícevodňový systém s vrty, approximace bodových zdrojů, metoda konečných prvků, FEM, metoda rozšířených konečných prvků, XFEM, Deal II

Abstract

Large areas are often included in models of an underground water flow where sources compared to the dimensions of areas seem to be point sources. Good examples of these are areas with hydrogeological wells or areas where mining is going on (or was). There can be a difference of several orders between dimensions of an area and a well. The finite element method cannot approximate an accurate solution of a pressure and flow properly.

The main aim of this paper is an application of a modern method of extended finite elements (XFEM) which can compensate an error of an approximation by linear finite elements and provides us more accurate solution than the finite element method.

A multiaquifer model of a flow in a porous media with wells is described in the paper. Its mathematical formulation is introduced and the existence of the unique weak solution of the finite element method is shown. Next the model is solved by a linear finite element method on an adaptively refined grid. Finally the use of XFEM is introduced and it is compared with other methods in several numerical tests.

Keywords:

multiaquifer system with wells, approximation of point sources, finite element method, FEM, extended finite element method, XFEM, Deal II

Obsah

Zadání	2
Prohlášení	3
Poděkování	4
Abstrakt	5
Obsah	7
Seznam obrázků	9
Seznam tabulek	9
Seznam použitých značení	10
1 Úvod	11
2 Matematický model	13
3 Metoda konečných prvků	17
3.1 Odvození slabé formulace	17
3.2 Diskretizace	24
3.3 Maticový zápis	25
4 Metoda rozšířených konečných prvků	27
4.1 Obohacující funkce	28
4.2 Diskretizace	29
4.3 Maticový zápis	30
4.4 Adaptivní integrace na obohacených elementech	32
4.5 Vrty jako zdroje	35
4.5.1 Odvození slabé formulace	35
4.5.2 Výhody modelu se zdroji	36
5 Implementace	37
5.1 Základní struktura tříd	37
5.2 FEM model s adaptivně zjemňovanou sítí	41
5.2.1 Tvorba sítě	41
5.2.2 Vyhledání elementů s vrty	41
5.2.3 Sestavení matice systému	42
5.2.4 Adaptivita	42

5.3	XFEM model	44
5.3.1	Realizace obohacení	44
5.3.2	Sestavení matice systému	45
5.4	Model s jednou zvodní a jedním vrtem	46
6	Numerické testy	48
6.1	Úloha s jedním vrtem na kruhové oblasti	48
6.2	Úloha s jedním vrtem na čtvercové oblasti	52
6.3	Poloměr obohacení	54
6.4	Úloha s dvěma vrty	57
6.5	Úloha s více vrty	58
7	Závěr	60
Seznam použité literatury		63
A	Metoda hraničních prvků	65
A.1	Odvození slabé formulace	65
A.2	Diskretizace	66
B	Obrazová příloha	67
C	Zdrojový kód	68
C.1	function AdaptiveIntegration::integrate	68

Seznam obrázků

1	Schéma systému zvodní s vrty	13
2	Model proudění ve vrtu	15
3	Obohacující funkce $\phi(\mathbf{x})$	28
4	Adaptivní zjemnění elementu	33
5	Mapování elementu	34
6	Struktura tříd	38
7	Lokálně zjemňovaná síť	43
8	Volné uzly	43
9	Průběh tlaku v kruhové zvodni s jedním vrtem	49
10	Konvergance metod FEM a XFEM	51
11	Průběh chyby řešení XFEM	52
12	Chyby aproximace v okolí vrstu	54
13	Dekomponované řešení XFEM	55
14	Chyba řešení XFEM	56
15	Špatné obohacení	57
16	Řešení modelu se dvěma vrty pomocí XFEM	58
17	Porovnání sítí metod FEM a XFEM	59
18	Řešení modelu s pěti vrty pomocí XFEM	59
19	Různé poloměry obohacení	67

Seznam tabulek

1	Porovnání XFEM a FEM metod s analytickým řešením	50
2	Změna velikosti σ	53
3	Změna velikosti obohacené oblasti na kruhu	55
4	Porovnání různých zjemnění sítě pro XFEM metodu	58
5	Porovnání metod FEM a XFEM na úloze s pěti vrty	58
6	Změna velikosti obohacené oblasti na čtverci	68

Seznam použitých značení

FEM	– Finite Element Method (Metoda konečných prvků)
XFEM	– Extended Finite Element Method (Rozšířená metoda FEM)
∇	– nabla, operátor gradientu $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)$
Δ	– Laplaceův operátor $\nabla \cdot \nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$
\mathbb{N}, \mathbb{R}^n	– prostor přirozených čísel, prostor reálných čísel dimenze n
$\partial\Omega, \bar{\Omega}$	– hranice množiny Ω , uzávěr množiny Ω
$L^2(\Omega)$	– prostor všech funkcí integrovatelných s 2. mocninou na Ω
\mathbf{n}	– normálový vektor v \mathbb{R}^2 případně \mathbb{R}^3 , $\mathbf{n} = (n_x, n_y, n_z)$
\mathbf{x}	– bod prostoru \mathbb{R}^2 případně \mathbb{R}^3 , $\mathbf{x} = [x, y, z] = [x_1, x_2, x_3]$
$(\cdot)^T$	– transpozice matice, vektoru,
$\ \cdot\ _V$	– norma vektoru na prostoru V
$\ \cdot\ _2 \ \cdot\ _{1,2}$	– norma vektoru na prostoru $L^2(\Omega)$ a $W^{1,2}(\Omega)$
$a(\cdot, \cdot)$	– bilineární forma $a : X \times X \rightarrow \mathbb{R}$
Ω^m	– oblast m -té zvodně
B_w^m	– průnik oblasti m -té zvodně a w -tého vrtu
∂B_w^m	– hranice w -tého vrtu na m -té zvodni
h^m / \tilde{h}^m	– tlaková/piezometrická výška na m -té zvodni
H_w^m	– stupeň volnosti pro tlak ve w -tém vrtu na úrovni m -té zvodně
φ_i^m	– i -tá bázová funkce na m -té zvodni
ϕ_w^m	– rozšířená bázová funkce od vrtu w na m -té zvodni
α	– stupně volnosti klasické metody FEM
β	– stupně volnosti rozšířené metody XFEM

1 Úvod

V mnoha oblastech lidského počínání se dnes setkáváme s potřebou zkoumat některé procesy, hledat a popsat charakteristické vlastnosti komplikovaných systémů, nalézt parametry, na kterých závisí jejich chování, a umět toto chování predikovat. Uvedeme standardní definici modelování jako výzkumné techniky (převzatou z teorie automatizace¹):

„Podstatou modelování ve smyslu výzkumné techniky je náhrada zkoumaného systému jeho modelem (přesněji: systémem, který jej modeluje), jejímž cílem je získat pomocí pokusů s modelem informaci o původním systému.“

Jednou z takto popsaných výzkumných technik je matematické modelování, které transformuje model do matematického zápisu. Může být velmi užitečným nástrojem, obzvlášť v situacích, ve kterých si například nemůžeme dovolit reálný experiment provést, nebo to ani není možné.

V přírodě se setkáváme v zásadě se spojitými veličinami, někdy však musíme zaházet až do mikroskopických rozměrů, abychom potvrdili spojitý charakter. Naopak v některých makroskopických měřítkách se setkáváme s téměř skokovým chováním nebo se v našem matematickém popisu mohou objevit singularity, které nemají v původním systému fyzikální opodstatnění. Právě takové úlohy, ve kterých se snažíme zachytit lokální nespojitost jinak spojité veličiny, mohou být velmi problematické.

V této práci se zabýváme výpočtem modelu podzemního proudění vody v oblasti s vrty – bodovými zdroji. Příkladem mohou být úlohy proudění v oblastech, na kterých dochází nebo docházelo k těžbě, nebo oblasti s hustou sítí vertikálních hydrogeologických vrtů, ve kterých se setkáme s problémem aproximace tlaku v blízkém okolí vrtů. Oblasti bývají oproti velikosti vrtů natolik rozsáhlé, že se vrty jeví téměř jako bodové zdroje, neboť jejich průměr bývá o několik řádů menší než rozměr oblasti (vrty o rozměrech několika centimetrů a naproti tomu modelovaná oblast v rozsahu desítek metrů až kilometrů).

Průběh přesného řešení tlaku v okolí vrtu má logaritmický charakter a při bodové představě vrtu se blíží v tomto bodě řešení v absolutní hodnotě nekonečnu. Nejčastěji používaná metoda konečných prvků nedokáže dostatečně přesně approximovat tlak

¹Dohoda o chápání pojmu simulace systémů. Automatizace, 1986, roč. 29, č. 12, s. 299–300.

v okolí takové singularity (viz článek [GR10], který se zabývá modelem proudění v úloze úložiště CO₂ a použitím rozšířených konečných prvků). Pro zlepšení aproximace je zapotřebí výrazného zjednodušení výpočetní sítě v okolí vrutu, čímž dochází ke zvýšení složitosti výpočtu a také problémům s konvergencí řešiče systému lineárních rovnic, neboť roste velikost úlohy a může se zhoršovat podmíněnost matice popisující tento systém.

Matematický model ustáleného saturovaného podzemního proudění vody popisuje eliptická parciální diferenciální rovnice. Naším úkolem je implementovat a porovnat několik metod výpočtu aproximace tlaku pomocí konečných prvků.

Navazujeme na článek R.Gracie a R. Craiga [GR10]. Řešíme velmi podobnou úlohu (rozdíly v matematickém modelu uvedeme v kapitole 2) a naším cílem je použít moderní metodu rozšířených konečných prvků ve vlastní implementaci a reprodukovat výsledky demonstrované v uvedeném článku. Navíc chceme vyzkoušet modifikaci modelu, ve které si vrty představujeme jiným způsobem než jako okrajové podmínky. Dále chceme spočítat model pomocí metody konečných prvků s adaptivně zjemňovnou sítí a pomocí metody hraničních prvků a porovnat dosažené výsledky. provedeme vlastní implementaci všech metod v jazyce C++ za pomoci knihovny Deal II.

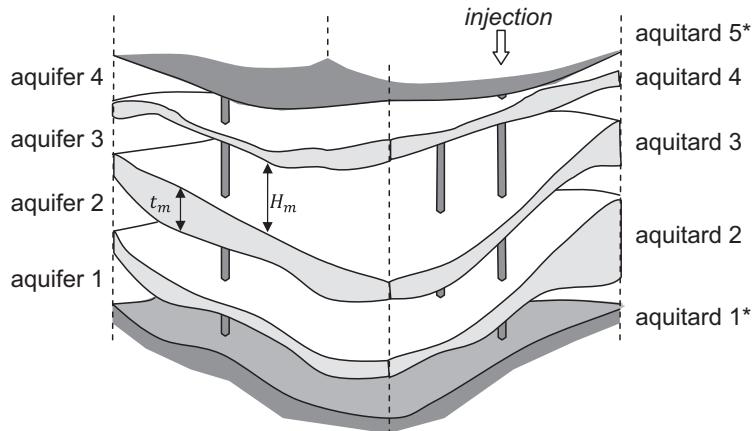
Text je rozdělen do několika hlavních kapitol. Nejprve v kapitole 2 popíšeme matematický model, zavedeme všechna potřebná značení a popíšeme rovnice, které budeme řešit. Dále se pak v kapitole 3 budeme věnovat řešení pomocí metody konečných prvků, odvodíme slabou formulaci a ukážeme, že existuje právě jedno slabé řešení. Poté v kapitole 4 představíme metodu rozšířených konečných prvků, definujeme obohacení a rozebereme detailně postup výpočtu touto metodou. Kapitola 5 je věnována popisu implementace, seznámení s knihovnou Deal II a vysvětlení některých algoritmů. V kapitole 6 demonstrujeme výsledky na několika testovacích úlohách a ukážeme výhody a nevýhody použitých metod. V závěrečné kapitole 7 shrneme dosažené výsledky a nastíníme směr dalšího možného vývoje.

Součástí textu je také příloha, ve které jsou některé doplňující obrázky (příloha B), zdrojové kódy (příloha C). Také mezi přílohy byla umístěna teoretická část k metodě hraničních prvků, kterou se nepodařilo odladit a dovést do plně funkčního stavu, aby bylo možné ji použít pro srovnání s ostatními metodami (příloha A).

2 Matematický model

Modelovat budeme ustálené saturované proudění podzemní vody v oblasti tvořené vrstvami – zvodněmi (spojitá tělesa vody, kterými může docházet k transportu hmot, též akvifer z anglického aquifer). Vzorem nám bude model popisovaný v [GR10], dokumentovaný obrázkem 1. V našem případě budeme uvažovat zvodně oddělené od sebe zcela nepropustnou vrstvou.

Zvodně mezi sebou tedy komunikují pouze prostřednictvím vrtů. Proudění v jedné zvodni uvažujeme jako proudění v ploše. Třírozměrný problém je tak zjednodušen na sadu dvourozměrných problémů na jednotlivých vrstvách, mezi kterými dokážeme definovat tok skrz vrty. Na okraji plochy zvodní uvažujeme znalost tlaku (pro Dirichletovu okrajovou podmínku) nebo nepropustnost, tzn. že zde bude možné zavést homogenní Neumannovu okrajovou podmínku.



obr. 1: Schéma systému zvodní s vrty popisovaný v [GR10].

Oblast je perforována vrty. Do některých z nich bude vháněna kapalina pod daným tlakem, ostatní vrty budou volné, tedy v jejich ústí bude tlak roven atmosférickému. Při formulaci problému nebudeme vrty uvažovat jako bodové zdroje, ale přiřadíme jim nenulový poloměr. Tlak na průřezu vrtu uvažujeme konstantní, tj. ve vrtu řešíme jednorozměrnou úlohu. V případě přítomnosti pouze homogenní Neumannovy okrajové podmínky znalost tlaku na ústí injektovaných vrtů a atmosférický tlak ve volných vrtech zafixují řešení (budou mít v podstatě funkci Dirichletovy okrajové podmínky jednodimenzionální úlohy na vrtu).

Zavedeme nyní značení oblastí. Mějme M zvodní (2D oblastí definované tloušťky) indexovaných $m = 1, \dots, M$ od nejnižše položené až po vrchní zvodeň. Tyto zvodně

protíná W vrtů indexovaných $w = 1, \dots, W$. Označme Ω^m oblast m -té zvodně s hranicí $\partial\Omega^m$, dále B_w^m průnik m -té zvodně s w -tým vrtem a ∂B_w^m jeho hranici (kružnice). Pak budeme potřebovat značení pro oblast celé zvodně bez vrtů, které do zvodně zasahují, tedy $\Theta^m = \Omega^m - \bigcup_{w=1}^W B_w^m$.

Proudění kapaliny v plně saturovaném porézním prostředí popisují dvě rovnice

$$\mathbf{u}^m = -\mathbf{T}^m \nabla \tilde{h}^m \quad \text{na } \Theta^m \text{ pro } m = 1, \dots, M \quad (2.1)$$

$$\nabla \cdot \mathbf{u}^m = f^m \quad \text{na } \Theta^m \text{ pro } m = 1, \dots, M \quad (2.2)$$

Rovnice (2.1) vyjadřuje Darcyho zákon a (2.2) je rovnicí kontinuity pro nestlačitelnou kapalinu. Index m v obou rovnicích označuje vztah veličin k m -té zvodni. V (2.1) značí \mathbf{T}^m [$\text{m}^2 \text{s}^{-1}$] dvou-dimenzionální tenzor transmisivity, \mathbf{u}^m [ms^{-1}] je horizontální tok zvodní, $\tilde{h}^m = \frac{p}{\rho g} + z$ [m] je piezometrická výška. Funkce f^m [ms^{-1}] v (2.2) vyjadřuje hustotu zdrojů. Pokud zavedeme označení pro tloušťku zvodně t^m [m], můžeme psát také $\mathbf{T}^m = \mathbf{K}^m t^m$, kde \mathbf{K}^m [ms^{-1}] je tenzor horizontální hydraulické vodivosti zvodně, a $f^m = q^m t^m$, kde q^m [s^{-1}] je objemová hustota zdrojů. V izotropním prostředí přejde \mathbf{K}^m a tedy i \mathbf{T}^m ve skalárni funkci, kterou pro zjednodušení budeme brát konstantní, tedy K^m a T^m .

Pokud dosadíme Darcyho zákon (2.1) do rovnice kontinuity (2.2), popisuje rozložení tlaku ve zvodních Poissonova rovnice

$$-T^m \Delta h^m = f^m \quad \text{na } \Theta^m \text{ pro } m = 1, \dots, M, \quad (2.3)$$

kde jsme mohli piezometrickou výšku na vodorovné zvodni při $z = \text{konst} = 0$ nahradit funkcí tlakové výšky $h^m = \frac{p}{\rho g}$ na m -té zvodni, protože rovnice (2.3) je lineární.

Definujme nyní okrajové podmínky na hranici oblasti. Hranici oblasti Θ^m rozdělíme na tři části Γ_N^m , Γ_D^m a Γ_B^m , přičemž platí pro vnější hranici celé zvodně $\partial\Omega^m = \Gamma_N^m \cup \Gamma_D^m$ a pro hranici oblasti $\partial\Theta^m = \Gamma_N^m \cup \Gamma_D^m \cup \Gamma_B^m$, kde $\Gamma_B^m = \bigcup_{w=1}^W \partial B_w^m$. Na hranici Γ_N^m volíme pro jednoduchost homogenní Neumannovu okrajovou podmínu, tedy nulový tok ze zvodně

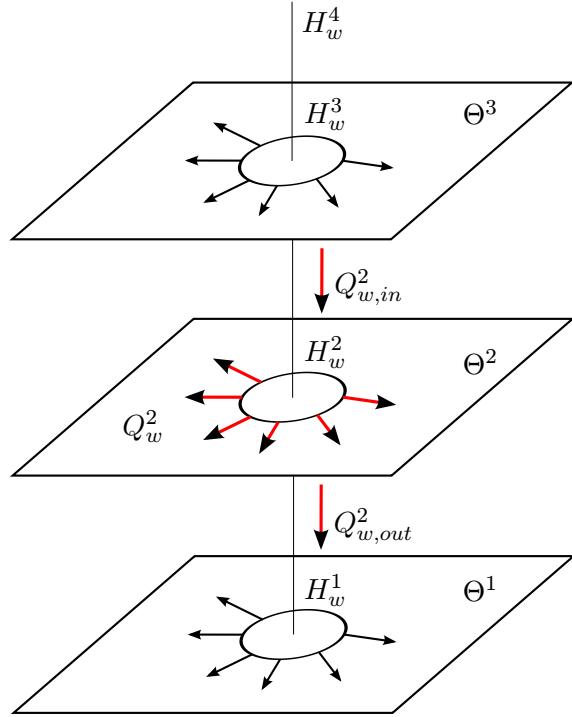
$$\mathbf{u}^m \cdot \mathbf{n}^m = 0 \quad \text{na hranici } \Gamma_N^m, \quad (2.4)$$

tam, kde uvažujeme hranici zvodní s okolím nepropustnou. V (2.4) je \mathbf{n} vnější normálna hranice². Tam, kde je známý tlak, volíme Dirichletovu okrajovou podmínu

$$h^m = h_D^m \quad \text{na hranici } \Gamma_D^m, \quad h_D^m(\mathbf{x}) \in C^1(\overline{\Theta^m}), \quad h_D^m|_{\Gamma_B^m} = 0. \quad (2.5)$$

²zvodně modelujeme jako plochy, tudíž vektory \mathbf{n} a \mathbf{x} zde mají pouze dvě složky

Funkce h_D^m v (2.5) je spojitá na $\overline{\Theta^m}$. Protože vrty nikdy nebudeme uvažovat na hranici zvodní, můžeme v (2.5) zvolit za h_D^m takové funkce, které směrem od hranice dovnitř oblasti rychle klesají k nule a pak předpokládat $h_D^m|_{\Gamma_B^m} = 0$.



obr. 2: Model proudění ve vrtu.

Věnujme se nyní komunikaci mezi vrty a zvodněmi. Definujme hustotu toku ze zvodně do vrtu výrazem

$$\sigma_w^m (h^m(\mathbf{x}) - H_w^m) \quad (2.6)$$

kde σ_w^m [ms^{-1}] je koeficient propustnosti mezi w -tým vrttem a m -tou zvodní, H_w^m je tlaková výška ve vrtu w na úrovni m -té zvodně a $h^m(\mathbf{x})$ je tlaková výška na hranici ∂B_w^m . Podél vrty budeme řešit jednodimenzionální úlohu danou rovnicemi bilance toku z vrty do zvodní

$$\begin{aligned} Q_w^m &= Q_{w,in}^m - Q_{w,out}^m, \quad \text{kde} \\ Q_w^m &= - \int_{\partial B_w^m} \sigma_w^m (h^m(\mathbf{x}) - H_w^m) d\mathbf{x} \quad \text{je tok do zvodně}, \\ Q_{w,in}^m &= -c_w^{m+1} (H_w^m - H_w^{m+1}) \quad \text{je tok z horní zvodně}, \\ Q_{w,out}^m &= -c_w^m (H_w^{m-1} - H_w^m) \quad \text{je tok do spodní zvodně}, \\ &\forall m = 1, \dots, M \text{ a } \forall w = 1, \dots, W. \end{aligned} \quad (2.7)$$

Koeficienty c_w^m [m^2s^{-1}] vyjadřují vodivosti vrtů mezi zvodněmi (můžeme si představit, že vrt nebude prázdný, ale vyplněn porózním materiélem). Z [GR10] jsme použili vztah $c_w^m = k_w^m \pi r_w^2 / t^m$, kde k_w^m je vertikální hydraulická vodivost ve vrtu a r_w je poloměr vrtu. Poznamenejme také, že členy H_w^0 a H_w^{M+1} v (2.7) mají funkci Dirichletovy okrajové podmínky jednodimenzionální úlohy na vrtu. Pokud je např. na celé hranici $\partial\Omega^m = \Gamma_N^m$ všech zvodní, zajišťuje H_w^M jednoznačnost řešení, což později ukážeme. Pro volné vrty předpokládáme atmosférický tlak, čili $H_w^{M+1} = 0$. V článku [GR10] není rovnice (2.7) řešena explicitně, ale je zde zavedena střední hodnota tlaku na hranici vrtu, pro kterou máme ekvivalent ve stupni volnosti H_w^m .

Když řešíme úlohu na oblasti Θ^m (bez vrtů), můžeme nahlížet na komunikaci mezi vrty a zvodněmi jako na Newtonovu okrajovou podmíncu

$$\mathbf{u}^m \cdot \mathbf{n}^m = \sum_{w=1}^W \sigma_w^m (h^m(\mathbf{x}) - H_w^m) \quad \text{na hranici } \Gamma_B^m = \bigcup_{w=1}^W \partial B_w^m, \quad (2.8)$$

kde se na pravé straně objevuje suma hustoty toků (výraz v sumě je shodný s (2.6)) přes hranice jednotlivých vrtů. Tento výraz se objeví ve slabé formulaci v hraničním integrálu.

Definice 2.1. Předpokládejme, že hranice oblasti $\partial\Theta^m$ je lipschitzovká (viz str. 341 v [Rek99]). Označme

$$\mathbf{h}_0 = (h_0^1, \dots, h_0^M, 0, \dots, 0, H_1^1, \dots, H_W^M, 0, \dots, 0) \quad (2.9)$$

$$\mathbf{h}_D = (h_D^1, \dots, h_D^M, H_1^0, \dots, H_W^0, 0, \dots, 0, H_1^{M+1}, \dots, H_W^{M+1}) \quad (2.10)$$

Pak nazveme řešení

$$\mathbf{h} = \mathbf{h}_0 + \mathbf{h}_D \in (C^2(\Theta) \cap C^1(\overline{\Theta}))^M \times \mathbb{R}^{(M+2) \times W} \quad (2.11)$$

které splňuje soustavu (2.3) a (2.7) a vyhovuje okrajovým podmínkám (2.4) a (2.5), silným řešením problému.

Pro výpočet popsaného modelu jsme vybrali několik metod – klasickou metodu konečných prvků, metodu hraničních prvků a metodu rozšířených konečných prvků. Popišme si nyní jejich formulace a vlastnosti.

3 Metoda konečných prvků

Řešme nyní výše popsanou úlohu a hledejme řešení pomocí metody konečných prvků (z angličtiny FEM – Finite Element method). Na okraji zvodní uvažujeme výše zavedené okrajové podmínky (2.4) a (2.5). Tok mezi vrty a zvodněmi budeme uvažovat jako komunikaci přes hranici, a proto ji budeme formulovat jako Newtonovu okrajovou podmínsku (2.8).

3.1 Odvození slabé formulace

Definujme Hilbertovy prostory

$$\begin{aligned} W^{1,2}(\Theta) &= \{h \in L^2(\Theta, \mathbb{R}); \nabla h \in L^2(\Theta, \mathbb{R}^2)\} \\ W_0^{1,2}(\Theta) &= \{h \in W^{1,2}(\Theta); h|_{\Gamma_D} = 0\} \end{aligned} \quad (3.1)$$

a normu na těchto prostorzech

$$\|h\|_{1,2,\Theta} = \|h\|_{W^{1,2}(\Theta)} = \sqrt{\|h\|^2 + \|\nabla h\|^2}. \quad (3.2)$$

Pokud bude z kontextu zřejmé, o jaký prostor se v (3.2), budeme dále vynechávat značení prostoru, tedy $\|h\|_{1,2}$. Stejně tak budeme značit $\|h\| = \|h\|_2$ normu prostoru L^2 tam, kde jsou prostor a typ normy zřejmé z kontextu. Pro zjednodušení zápisu zavedeme následující množiny indexů

$$\mathcal{M} = \{1, \dots, M\}, \quad (3.3)$$

$$\mathcal{M}_w = \{0, \dots, M+1\}, \quad (3.4)$$

$$\mathcal{W} = \{1, \dots, W\}. \quad (3.5)$$

Hledejme řešení ve tvaru $\mathbf{h} = \mathbf{h}_0 + \mathbf{h}_D$, takové že

$$\mathbf{h}_0 = (h_0^1, \dots, h_0^M, 0, \dots, 0, H_1^1, \dots, H_W^M, 0, \dots, 0) \quad (3.6)$$

$$\mathbf{h}_D = (h_D^1, \dots, h_D^M, H_1^0, \dots, H_W^0, 0, \dots, 0, H_1^{M+1}, \dots, H_W^{M+1}) \quad (3.7)$$

kde jednotlivé funkce $h_0^m \in W_0^{1,2}(\Theta^m)$, $h_D^m \in W^{1,2}(\Theta^m)$ pro všechna $m \in \mathcal{M}$ a konstanty $H_w^m \in \mathbb{R}$ pro všechna $m \in \mathcal{M}_w$ a všechna $w \in \mathcal{W}$. Zavedeme dále v (3.8) prostory V a V_D spolu s normou

$$V = W_0^{1,2}(\Theta)^M \times \mathbb{R}^{(M+2) \times W} \quad (3.8)$$

$$V_D = W^{1,2}(\Theta)^M \times \mathbb{R}^{(M+2) \times W} \quad (3.9)$$

$$\|\mathbf{h}\|_V = \sqrt{\sum_{m \in \mathcal{M}} \|h^m\|_{1,2}^2 + \sum_{m \in \mathcal{M}_w} \sum_{w \in \mathcal{W}} |\bar{H}_w^m|^2}, \quad (3.10)$$

takže potom můžeme psát $\mathbf{h}_0 \in V$ a $\mathbf{h}_D \in V_D$.

Zvolme testovací funkce

$$\mathbf{v} = (v^1, \dots, v^M, \bar{v}_1^0, \dots, \bar{v}_W^0, \bar{v}_1^1, \dots, \bar{v}_W^{M+1}) \in V \quad (3.11)$$

tak, že konstanty $\bar{v}_w^0 = \bar{v}_w^{M+1} = 0$ položíme rovné nule pro všechna $w \in \mathcal{W}$.

Přistupme nyní k vlastnímu odvození slabé formulace. Upravme nejprve rovnici (2.3), popisující proudění v m -té zvodni, vynásobme ji testovací funkcí v^m a integrujme ji přes Θ^m . Dostaneme

$$\int_{\Theta^m} -T^m \Delta h^m v^m \, d\mathbf{x} = \int_{\Theta^m} f^m v^m \, d\mathbf{x}. \quad (3.12)$$

Levou stranu rovnice (3.12) následně upravíme pomocí Greenovy věty (viz věta 8.7 v [Rek99]) do tvaru

$$\int_{\Theta^m} T^m \nabla h^m \cdot \nabla v^m \, d\mathbf{x} - \int_{\partial\Omega^m} T^m \nabla h^m \cdot \mathbf{n} v^m \, d\mathbf{x} = \int_{\Theta^m} f^m v^m \, d\mathbf{x}. \quad (3.13)$$

Do hraničního integrálu v (3.13) dosadíme okrajové podmínky. Člen na hranici Γ_D je nulový, protože je zde testovací funkce nulová, člen na hranici Γ_N je nulový, protože Neumannova podmínka je nulová. Výsledkem této úpravy je rovnice

$$\int_{\Theta^m} T^m \nabla h^m \cdot \nabla v^m \, d\mathbf{x} + \sum_{w \in \mathcal{W}} \int_{\partial B_w^m} \sigma_w^m (h^m - H_w^m) v^m \, d\mathbf{x} = \int_{\Theta^m} f^m v^m \, d\mathbf{x} \quad (3.14)$$

Tak jako jsme zavedli $\mathbf{h} = \mathbf{h}_0 + \mathbf{h}_D$, platí i na jedné zvodni $h^m = h_0^m + h_D^m$, což nyní dosadíme do (3.14). Protože navíc $h_D^m|_{\Gamma_B} = 0$, viz (2.5), dostáváme

$$\begin{aligned} \int_{\Theta^m} T^m \nabla h_0^m \cdot \nabla v^m \, d\mathbf{x} + \sum_{w \in \mathcal{W}} \int_{\partial B_w^m} \sigma_w^m (h_0^m - H_w^m) v^m \, d\mathbf{x} &= \\ &= \int_{\Theta^m} f^m v^m \, d\mathbf{x} - \int_{\Theta^m} T^m \nabla h_D^m \cdot \nabla v^m \, d\mathbf{x} \end{aligned} \quad (3.15)$$

Zbývá nám upravit soustavu rovnic na vrtech složenou z členů v (2.7). Vynásobíme rovnice testovacími funkcemi \bar{v}_w^m (jsou konstanty, proto mohou být zahrnuty v integrálu) a zároveň dosadíme za h^m , jako jsme to udělali v (3.15), čímž dostaneme

$$\begin{aligned} \int_{\partial B_w^m} \sigma_w^m (h_0^m - H_w^m) \bar{v}_w^m \, d\mathbf{x} &= \\ &= c_w^{m+1} (H_w^m - H_w^{m+1}) \bar{v}_w^m - c_w^m (H_w^{m-1} - H_w^m) \bar{v}_w^m. \end{aligned} \quad (3.16)$$

Definice 3.1. Pokud $\mathbf{h} \in V_D$ splňuje rovnice (3.15) a (3.16) pro všechny funkce $\mathbf{v} \in V$, pak nazveme \mathbf{h} slabým řešením.

Poznamenejme, že v definici 3.1 závisí \mathbf{h}_0 na volbě \mathbf{h}_D . Lze ovšem ukázat, že \mathbf{h} už je na \mathbf{h}_D nezávislé.

Nyní se budeme zabývat řešitelností úlohy. Odečtěme rovnici (3.16) od (3.15) a zároveň vysčítějme soustavy přes indexy m, w . Získáme tím jedinou rovnici (3.17), kterou budeme moci vyjádřit pomocí bilineární formy $a(\cdot, \cdot) : V \times V \rightarrow \mathbb{R}$ a funkcionálu $F(\cdot) : V \rightarrow \mathbb{R}$.

$$a(\mathbf{h}_0, \mathbf{v}) = F(\mathbf{v}), \quad (3.17)$$

kde

$$\begin{aligned} a(\mathbf{h}_0, \mathbf{v}) &= \sum_{m \in \mathcal{M}} \int_{\Theta^m} T^m \nabla h_0^m \cdot \nabla v^m \, d\mathbf{x} + \sum_{\substack{m \in \mathcal{M} \\ w \in \mathcal{W}}} \int_{\partial B_w^m} \sigma_w^m (h_0^m - H_w^m) (v^m - \bar{v}_w^m) \, d\mathbf{x} + \\ &+ \sum_{\substack{m \in \mathcal{M}_w \\ w \in \mathcal{W}}} [c_w^{m+1} (H_w^m - H_w^{m+1}) \bar{v}_w^m - c_w^m (H_w^{m-1} - H_w^m) \bar{v}_w^m] \end{aligned} \quad (3.18)$$

$$F(\mathbf{v}) = \sum_{m \in \mathcal{M}} \left[\int_{\Theta^m} f^m v^m \, d\mathbf{x} - \int_{\Theta^m} T^m \nabla h_D^m \cdot \nabla v^m \, d\mathbf{x} \right] \quad (3.19)$$

Zdali existuje řešení rovnice (3.17) a jestli je takové jediné, to nám říká Laxova-Milgramova věta 3.1 (věta 33.1 v [Rek99]).

Věta 3.1. (Laxova-Milgramova) Nechť je V Hilbertův prostor se skalárním součinem (h, v) a nechť $a(h, v)$ je bilineární forma, definovaná pro $h, v \in V$ a taková, že existují konstanty $C_1 > 0$ a $\alpha > 0$ nezávislé na h a v tak, že pro každé $h, v \in V$ platí

$$|a(h, v)| \leq C_1 \|h\|_V \|v\|_V \quad \text{podmínka omezenosti}, \quad (3.20)$$

$$a(v, v) \geq \alpha \|v\|_V^2 \quad \text{podmínka V-ellipticity}. \quad (3.21)$$

Pak lze každý lineární funkcionál F , omezený na V (tedy i spojitý na V), vyjádřit ve tvaru

$$F(v) = a(z, h), \quad v \in V, \quad (3.22)$$

kde z je prvek prostoru V , jednoznačně určený funkcionálem F . Přitom je

$$\|z\| \leq \frac{\|F\|}{\alpha}. \quad (3.23)$$

Připomeňme, že funkcionál ve větě 3.1 je omezený, existuje-li takové číslo C_2 , že pro všechny prvky $v \in V$ platí

$$|F(v)| \leq C_2 \|v\|, \quad (3.24)$$

a norma $\|F\|$ je rovna nejmenšímu číslu C_2 , pro které platí (3.24), viz definice 8.19 v [Rek99].

Abychom tedy ukázali, že existuje jediné řešení rovnice (3.17), musíme dle věty 3.1 ověřit, že jsou splněny nerovnice

$$|a(\mathbf{h}_0, \mathbf{v})| \leq C_1 \|\mathbf{h}_0\|_V \|\mathbf{v}\|_V \quad (3.25)$$

$$a(\mathbf{v}, \mathbf{v}) \geq \alpha \|\mathbf{v}\|_V^2 \quad (3.26)$$

$$|F(\mathbf{v})| \leq C_2 \|\mathbf{v}\|_V. \quad (3.27)$$

K tomu potřebujeme několik nerovností, které nám pomohou v dílčích krocích. Nerovnost (3.28) se nazývá trojúhelníková, nerovnost (3.29) je Hölderova, obě najdeme ve větě 3.1 v [Rek99].

$$|a + b| \leq |a| + |b| \quad (3.28)$$

$$|(u, v)| \leq \|u\| \|v\| \quad (3.29)$$

Dále užijeme některé důležité věty funkcionální analýzy.

Věta 3.2. (O stopách) Nechť Ω je oblast s lipschitzovskou hranicí. Pak existuje, a to právě jeden, omezený lineární operátor T_r , který zobrazuje prostor $W^{1,2}(\Omega)$ do prostoru $L^2(\partial\Omega)$ tak, že $u(x) \in C^\infty(\bar{\Omega})$ je $T_r u = u|_{\partial\Omega}$

Z věty o stopách 3.2, citována z [Rek99] – věta 30.1, a omezenosti operátoru T_r dále plyne, že existuje konstanta $C_{TR} > 0$ taková, že

$$\|T_r u\|_{L^2(\partial\Omega)} \leq C_{TR} \|u\|_{W^{1,2}(\Omega)}. \quad (3.30)$$

Věta 3.3. (Zobecněná Friedrichsova nerovnost) Nechť Ω je oblast s lipschitzovskou hranicí Γ , Γ_1 její část mající kladnou Lebesgueovu míru, $m\Gamma_1 > 0$. Pak existuje konstanta $C_F > 0$, závislá jen na dané oblasti a na Γ_1 , taková, že pro každou funkci $u \in W^{1,2}(\Omega)$ platí

$$C_F \|u\|_{W^{1,2}(\Omega)}^2 \leq \|\nabla u\|_{L^2(\Omega)}^2 + \|T_r u\|_{\Gamma_1}^2 \quad (3.31)$$

Větu 3.3 nalezneme v [Rek99] – věta 30.3. Pokud ztotožníme v našem případě na m -té zvodni $\Gamma_1 = \Gamma_D^m$, kde víme, že $h_0^m|_{\Gamma_D^m} = 0$, pak dostaneme nerovnost (3.31) ve tvaru

$$C_F \|h_0^m\|_{1,2,\Theta^m} \leq \|\nabla h_0^m\|_{2,\Theta^m} \quad (3.32)$$

Ověření omezenosti formy $a(\mathbf{h}_0, \mathbf{v})$ (3.25)

Začneme ověřením omeznosti bilineární formy. Předpokládejme, že fyzikální konstanty jsou v reálné úloze takto omezené: $T^m > 0$ pro všechna $m \in \mathcal{M}$, $\sigma_w^m \geq 0$ a $c_w^m \geq 0$ pro všechna $m \in \mathcal{M}_w$ a $w \in \mathcal{W}$. Zároveň tyto konstanty vzhledem k fyzikální podstatě úlohy můžeme omezit i shora, proto zavedeme T, σ a c takové, že

$$T^m \leq T \quad \forall m \in \mathcal{M}, \quad (3.33)$$

$$\sigma_w^m \leq \sigma \quad \forall m \in \mathcal{M}_w \text{ a } \forall w \in \mathcal{W}, \quad (3.34)$$

$$c_w^m \leq c \quad \forall m \in \mathcal{M}_w \text{ a } \forall w \in \mathcal{W}. \quad (3.35)$$

Aplikujme nyní trojúhelníkovou nerovnost (3.28) na pravou stranu (3.25) a použijme výše zavedené konstanty. Dostaneme

$$\begin{aligned} |a(\mathbf{h}_0, \mathbf{v})| &\leq T \left| \sum_{m \in \mathcal{M}} \int_{\Theta^m} \nabla h_0^m \cdot \nabla v^m \, d\mathbf{x} \right| + \\ &\quad + \sigma \left| \sum_{\substack{m \in \mathcal{M} \\ w \in \mathcal{W}}} \int_{\partial B_w^m} (h_0^m - H_w^m) (v^m - \bar{v}_w^m) \, d\mathbf{x} \right| + \\ &\quad + c \left| \sum_{\substack{m \in \mathcal{M}_w \\ w \in \mathcal{W}}} [(H_w^m - H_w^{m+1}) \bar{v}_w^m - (H_w^{m-1} - H_w^m) \bar{v}_w^m] \right|. \end{aligned} \quad (3.36)$$

Vidíme, že je nyní možné omezit zvlášť každý z členů na pravé straně (3.36). Na první ze členů použijeme Hölderovu nerovnost (3.29) a získáme

$$\begin{aligned} T \left| \sum_{m \in \mathcal{M}} \int_{\Theta^m} \nabla h_0^m \cdot \nabla v^m \, d\mathbf{x} \right| &\leq T \sum_{m \in \mathcal{M}} \int_{\Theta^m} |\nabla h_0^m| |\nabla v^m| \, d\mathbf{x} \leq \\ &\leq T \sum_{m \in \mathcal{M}} \|\nabla h_0^m\|_2 \|\nabla v^m\|_2 \leq T \sum_{m \in \mathcal{M}} \|h_0^m\|_{1,2} \|v^m\|_{1,2} \end{aligned} \quad (3.37)$$

Ve druhém členu (3.36) roznásobíme součin v integrálu, použijeme opět trojúhelníkovou (3.28) a Hölderovu nerovnost (3.29) a navíc větu o stopách 3.2, čímž získáme

$$\begin{aligned} \sigma \sum_{\substack{m \in \mathcal{M} \\ w \in \mathcal{W}}} \int_{\partial B_w^m} (|h_0^m| |v^m| + |h_0^m| |\bar{v}_w^m| + |H_w^m| |v^m| + |H_w^m| |\bar{v}_w^m|) \, d\mathbf{x} &\leq \\ &\leq \sigma \sum_{\substack{m \in \mathcal{M} \\ w \in \mathcal{W}}} \left(C_{TR}^2 \|h_0^m\|_{1,2} \|v^m\|_{1,2} + C_{TR} |\partial B_w^m| \|h_0^m\|_{1,2} |\bar{v}_w^m| + \right. \\ &\quad \left. + C_{TR} |\partial B_w^m| |H_w^m| \|v^m\|_{1,2} + |\partial B_w^m| |H_w^m| |\bar{v}_w^m| \right). \end{aligned} \quad (3.38)$$

Třetí člen (3.36) je pouze sumou součinů konstantních funkcí, takže je možné jej omezit jejich absolutními hodnotami. Pak dostaneme

$$c \sum_{\substack{m \in \mathcal{M}_w \\ w \in \mathcal{W}}} |H_w^m| |\bar{v}_w^m| \quad (3.39)$$

Když nyní sloučíme úpravy (3.37), (3.38) a (3.39), a dosadíme zpět do (3.36) získáme

$$\begin{aligned} |a(\mathbf{h}_0, \mathbf{v})| &\leq C_1 \left(\sum_{m \in \mathcal{M}} \|h_0^m\|_{1,2} \|v^m\|_{1,2} + \right. \\ &\quad \left. + \sum_{\substack{m \in \mathcal{M} \\ w \in \mathcal{W}}} \left(\|h_0^m\|_{1,2} |\bar{v}_w^m| + |H_w^m| \|v^m\|_{1,2} \right) + \sum_{\substack{m \in \mathcal{M}_w \\ w \in \mathcal{W}}} |H_w^m| |\bar{v}_w^m| \right), \end{aligned} \quad (3.40)$$

kde konstanta $C_1 > 0$ závisí pouze na konstantách $T, \sigma, c, W, C_{TR}, |\partial B_w^m|$ (není nutné ji přesně vyčíslovat). Jednotlivé členy v závorce v (3.40) jsou obsaženy v součinu norm, viz (3.8), což lze ukázat jejich jednoduchým roznásobením. Pokud tedy levou stranu (3.40) rozšíříme o zbývající členy v normě (všechny jsou nezáporné), dostaneme po úpravách vyhovující tvar nerovnice

$$|a(\mathbf{h}_0, \mathbf{v})| \leq C_1 \|\mathbf{h}_0\|_V \|\mathbf{v}\|_V \quad (3.41)$$

Konstanta C_1 v (3.41) zřejmě není nejmenší, důležité však je, že taková existuje. Tím jsme ukázali, že platí nerovnice (3.25).

Ověření ellipticity formy $a(\mathbf{v}, \mathbf{v})$ (3.26)

Nyní budeme vyšetřovat ellipticitu bilineární formy $a(\mathbf{v}, \mathbf{v})$. Do (3.18) dosadíme za \mathbf{h}_0 funkci \mathbf{v} . Třetí člen (3.18) můžeme potom přeskladat a dostaneme tvar

$$\begin{aligned} a(\mathbf{v}, \mathbf{v}) &= \sum_{m \in \mathcal{M}} \int_{\Theta^m} T^m |\nabla v^m|^2 \, d\mathbf{x} + \sum_{\substack{m \in \mathcal{M} \\ w \in \mathcal{W}}} \int_{\partial B_w^m} \sigma_w^m (v^m - \bar{v}_w^m)^2 \, d\mathbf{x} - \\ &\quad - c_w^1 \bar{v}_w^1 (\bar{v}_w^0 - \bar{v}_w^1) + \dots + c_w^m (\bar{v}_w^{m-1} - \bar{v}_w^m)^2 + \dots + c_w^{M+1} \bar{v}_w^M (\bar{v}_w^M - \bar{v}_w^{M+1}). \end{aligned} \quad (3.42)$$

Je zřejmé, že všechny členy jsou nezáporné. Pokud hranice alespoň jedné zvodně s Dirichletovou okrajovou podmínkou je nenulové míry, potom je důkaz nerovnosti (3.26) přímočarý za použití Friedrichsovy nerovnosti (3.32).

$$\int_{\Theta^m} T^m |\nabla v^m|^2 \, d\mathbf{x} = T^m \|\nabla v^m\|_2^2 \geq C_F \|v^m\|_{1,2}^2 \quad (3.43)$$

Z (3.43) víme, že alespoň na jedné zvodni existuje $C_F > 0$, což nám stačí, abychom věděli, že minimálně $\alpha \geq C_F$.

V případě, že na hranici všech zvodní je zadána pouze Neumannova okrajová podmínka, provedeme důkaz analogický důkazu Friedrichsovy nerovnosti sporem (viz kapitola 2.1.1 v [RJJ⁺09]).

Řekněme, že (3.26) neplatí. Pak $\forall n \in \mathbb{N} \exists \mathbf{v}^n \in V$ takové, že $\|\mathbf{v}^n\|_V > n \cdot a(\mathbf{v}^n, \mathbf{v}^n)$. Bez újmy na obecnosti zvolme $\|\mathbf{v}^n\|_V = 1$. Prostor V složený z Hilbertových prostorů na Θ^m je také Hilbertův, a proto je také reflexivní. Z toho vyplývá, že posloupnost \mathbf{v}^n je omezená a lze k ní najít podposloupnost slabě konvergentní

$$\mathbf{v}^{n_k} \rightharpoonup \mathbf{v} \quad \text{ve } V.$$

Z věty o kompaktním vnoření $V \hookrightarrow L^2$ plyne silná konvergence

$$\mathbf{v}^{n_k} \rightarrow \mathbf{v} \quad \text{ve } L^2.$$

Protože všechny členy bilineární formy $a(\mathbf{v}, \mathbf{v})$ jsou nezáporné, omezíme ji zdola $a(\mathbf{v}, \mathbf{v}) \geq T \|\nabla \mathbf{v}\|_2^2$. Potom dostaneme

$$T \|\nabla \mathbf{v}^{n_k}\|_2^2 \leq a(\mathbf{v}^{n_k}, \mathbf{v}^{n_k}) < \frac{1}{n_k} \|\mathbf{v}^n\|_V = \frac{1}{n_k}.$$

Za předpokladu slabé zdola polospojitosti normy $\|\nabla \mathbf{v}\|_2 \leq \liminf_{k \rightarrow \infty} \|\mathbf{v}^{n_k}\|_2$,

$$\nabla \mathbf{v}^{n_k} \rightarrow \nabla \mathbf{v} = 0 \quad \text{v } L^2 \implies \mathbf{v}^{n_k} \rightarrow \mathbf{v} \quad \text{ve } V, \quad \mathbf{v} = \text{konst.}$$

Tlak ve zvodni je tedy konstantní, to znamená, že tok zvodní musí být nulový. Tok přes hranici vrtu bude nulový pouze pokud $\sigma_w^m = 0$, pak ale nebude žádná komunikace mezi vrtem a zvodní, nebo pokud $v^m = \bar{v}_w^m$. Potom ale musí být i tok vrty nulový, proto si musí být rovny všechny konstanty $\bar{v}_w^i = \bar{v}_w^j$, $\forall i, j \in \mathcal{M}_w$. Protože předpokládáme $\bar{v}_w^0 = \bar{v}_w^{M+1} = 0$, potom

$$\mathbf{v} = \text{konst.} = 0 \implies \text{spor s } \|\mathbf{v}\|_V = 1.$$

Tím je proveden důkaz elipticity (3.26).

Ověření omezenosti funkcionálu $F(\mathbf{v})$ (3.27)

Zbývá ověřit omezenost funkcionálu $F(\mathbf{v})$. Použijeme stejné úpravy jako při ověřování omezenosti bilineární formy (3.25), tedy trojúhelníkovou nerovnost (3.28) a Hölderovu nerovnost (3.29), a dostaneme

$$\begin{aligned}
 |F(\mathbf{v})| &= \left| \sum_{m \in \mathcal{M}} \left[\int_{\Theta^m} f^m v^m \, d\mathbf{x} - \int_{\Theta^m} T^m \nabla h_D^m \cdot \nabla v^m \, d\mathbf{x} \right] \right| \leq \\
 &\leq \sum_{m \in \mathcal{M}} \|f^m\|_2 \|v^m\|_{1,2} + T \sum_{m \in \mathcal{M}} \|h_D^m\|_{1,2} \|v^m\|_{1,2} \leq \\
 &\leq \sum_{m \in \mathcal{M}} (\|f^m\|_2 + T \|h_D^m\|_{1,2}) \|v^m\|_{1,2} \leq C_2 \|\mathbf{v}\|_V, \quad (3.44)
 \end{aligned}$$

kde konstanta C_2 závisí na velikostech $\|f^m\|_2, T, \|h_D^m\|_{1,2}$. Tím je ukázána platnost nerovnice (3.27) a zároveň proveden poslední článek důkazu existenze a jednoznačnosti řešení.

Věta 3.4. Nechť je bilineární forma $a(\mathbf{h}_0, \mathbf{v})$, definovaná v (3.18), omezena s konstantou C_1 , tedy je splněna nerovnice (3.25). Nechť je dále $a(\mathbf{h}_0, \mathbf{v})$ V -eliptická s konstantou α , tedy platí nerovnice (3.26), a lineární funkcionál $F(\mathbf{v})$, definovaný v (3.19), je omezený s konstantou C_2 . Potom jsou splněny všechny předpoklady Laxovy-Milgramovy věty 3.1 a existuje jediné řešení rovnice (3.18).

3.2 Diskretizace

Zvolme n -dimenzionální podprostor $W_{h,0}^{1,2} \subset W_0^{1,2}$ a jeho konečně-prvkovou bázi $\{\varphi_1, \dots, \varphi_N\}$ a zavedeme na m -té zvodni approximaci

$$h_0^m = \sum_{j=1}^N \alpha_j^m \varphi_j^m. \quad (3.45)$$

Aproximovaný bázový prostor $W_{h,0}^{1,2}$ ztotožníme s testovacím. Pro testovací funkce \bar{v}_w^m volíme bázové vektory $\mathbf{e}^{mw} = (0, \dots, 1, \dots, 0)$ prostoru $R^{M \times W}$, tedy v rovnici (3.16) pro m -tou zvodeň a w -tý vrt je $\bar{v}_w^m = 1$ a všechny ostatní jsou rovny nule. Dosazením approximovaného řešení a testovacích funkcí do odvozených rovnic (3.16) a (3.15) získáme $(M \cdot W)$ rovnic pro hranice vrtů

$$\begin{aligned}
 \int_{\partial B_w^m} \sigma_w^m \left(\sum_{j=1}^N \alpha_j^m \varphi_j^m - H_w^m \right) \bar{v}_w^m \, d\mathbf{x} &= \\
 &= c_w^{m+1} (H_w^m - H_w^{m+1}) \bar{v}_w^m - c_w^m (H_w^{m-1} - H_w^m) \bar{v}_w^m \quad (3.46)
 \end{aligned}$$

a pro M zvodní ($M \cdot N$) rovnic

$$\begin{aligned} & \sum_{j=1}^N \alpha_j^m \int_{\Theta^m} T^m \nabla \varphi_j^m \cdot \nabla \varphi_i^m d\mathbf{x} + \sum_{j=1}^N \alpha_j^m \sum_{w \in \mathcal{W}} \int_{\partial B_w^m} \sigma_w^m \varphi_j^m \varphi_i^m d\mathbf{x} - \\ & - \sum_{w \in \mathcal{W}} H_w^m \int_{\partial B_w^m} \sigma_w^m \varphi_i^m d\mathbf{x} = \int_{\Theta^m} f^m \varphi_i^m d\mathbf{x} - \int_{\Theta^m} T^m \nabla h_D^m \cdot \nabla \varphi_i^m d\mathbf{x}. \quad (3.47) \end{aligned}$$

Počet rovnic v soustavě (3.46) a (3.47) dává dohromady $(M \cdot N + M \cdot W)$, což odpovídá počtu stupňů volnosti $\{\alpha_1^1 \dots \alpha_N^1, \alpha_1^2 \dots \alpha_N^M, H_1^1 \dots H_W^1, H_1^2 \dots H_W^M\}$.

3.3 Maticový zápis

Doposud jsme popisovali komplexní model proudění podzemní vody zvodněmi a jeho řešení klasickou metodou konečných prvků. V implementaci pro zjednodušení omezíme model na jednu zvodeň – chceme porovnávat chování metod v okolí vrstu a nezáleží nám také na komplexnosti implementovaného modelu. Přepišme rovnice (3.47) a (3.46) pro tento případ, tzn. $m = 1$ nebudeme psát, $c_w^2 H_w^2$ má význam Dirichletovy okrajové podmínky na vrtech, do podloží nic neodteká a tudíž položíme $c_w = 0$. Označíme indexy $i, j = 1, \dots, N$ bázové funkce a stupně volnosti na zvodni, H_w jsou stupně volnosti pro tlaky ve vrtech. Rovnici (3.46) navíc násobíme -1 , abychom zajistili symetrii soustavy, kterou zapíšeme

$$\begin{pmatrix} \mathbf{A} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{D} \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{q} \\ \mathbf{d} \end{pmatrix}, \quad (3.48)$$

kde jsou jednotlivé prvky rovny

$$\begin{aligned} A_{ij} &= T \int_{\Theta} \nabla \varphi_j \cdot \nabla \varphi_i d\mathbf{x} + \sum_{w \in \mathcal{W}} \sigma_w \int_{\partial B_w} \varphi_j \varphi_i d\mathbf{x}, \\ C_{wj} &= -\sigma_w \int_{\partial B_w} \varphi_j d\mathbf{x}, \\ D_{ww} &= \int_{\partial B_w} \sigma_w dB_w + c_w^2, \\ q_i &= \int_{\Theta} f \varphi_i d\mathbf{x} - \int_{\Theta} T \nabla h_D \cdot \nabla \varphi_i d\mathbf{x}, \\ d_w &= c_w^2 H_w^2. \end{aligned} \quad (3.49)$$

Matice mají rozměry $\mathbf{A}^{N \times N}$, $\mathbf{C}^{W \times N}$ a $\mathbf{D}^{W \times W}$ a vektory \mathbf{q}^N a \mathbf{d}^W . Poznamenejme, že tyto matice jsou řídké, \mathbf{A} je navíc symetrická a \mathbf{D} je diagonální. Stupně volnosti jsme sloučili do vektorů $\mathbf{a} = (\alpha_1, \dots, \alpha_N)$ a $\mathbf{p} = (H_1, \dots, H_W)$.

Pokud bychom chtěli model rozšířit na více zvodní, bude soustava blokově diagonální. Přepišme rovnici (3.46) tak, abychom viděli lépe prvky matice.

$$\begin{aligned} -c_w^{m+1}H_w^{m+1} - \sum_{j=1}^N \alpha_j \int_{\partial B_w^m} \sigma_w^m \varphi_j^m d\mathbf{x} + \\ + H_w^m \left(\int_{\partial B_w^m} \sigma_w^m d\mathbf{x} + c_w^{m+1} + c_w^m \right) - c_w^m H_w^{m-1} = 0 \quad (3.50) \end{aligned}$$

Soustava bude navíc doplněna členy z rovnice (3.50), odkud tedy přibudou bloky $\mathbf{E}^m = \text{diag}(-c_1^m, \dots, -c_W^m)$, které reprezentují komunikaci uvnitř vrtů (blok \mathbf{E}^m dává do vztahu tlaky ve vrtech mezi zvodněmi m a $m-1$). K prvkům matice \mathbf{C}^m přičteme konstanty propustnosti ve vrtu, čili $C_{wj}^m = -\sum_{w \in \mathcal{W}} \left[\sigma_w^m \int_{\partial B_w^m} \varphi_j^m d\mathbf{x} + c_w^{m+1} + c_w^m \right]$, a matice systému bude mít tvar

$$\begin{pmatrix} \mathbf{A}^M & (\mathbf{C}^M)^T & & & \dots \\ \mathbf{C}^M & \mathbf{D}^M & \mathbf{E}^M & & \dots \\ & & \mathbf{A}^{M-1} & (\mathbf{C}^{M-1})^T & \dots \\ & & \mathbf{E}^M & \mathbf{C}^{M-1} & \mathbf{D}^{M-1} & \mathbf{E}^{M-1} & \dots \\ \vdots & \vdots & \vdots & \vdots & & \ddots \end{pmatrix} \begin{pmatrix} \mathbf{a}^M \\ \mathbf{p}^M \\ \mathbf{a}^{M-1} \\ \mathbf{p}^{M-1} \\ \vdots \end{pmatrix} = \begin{pmatrix} \mathbf{q}^M \\ \mathbf{d} \\ \mathbf{q}^{M-1} \\ 0 \\ \vdots \end{pmatrix}. \quad (3.51)$$

Pokud nejsou přítomny další zdroje, je vektor pravé strany při homogenní Neumannově okrajové podmínce roven $(0, \mathbf{d}, 0, 0, \dots, 0)$. To znamená, že je nulový až na řádky, ve kterých se objevuje tlak v ústí vrtů. Nakonec vektor hledaných stupňů volnosti ve zvodních a vrtech je roven $(\mathbf{a}^M, \mathbf{p}^M, \mathbf{a}^{M-1} \mathbf{p}^{M-1}, \dots, \mathbf{a}^1, \mathbf{p}^1)$.

4 Metoda rozšířených konečných prvků

Názvy metod XFEM (Extended FEM), GFEM (Generalized FEM) a PUM (Partition of Unity Method, popsal Babuška a Melenk [BM97]) se v dnešní době používají téměř jako synonyma. Dohromady je spojuje myšlenka rozšíření prostoru pro hledání řešení parciální diferenciální rovnice, kde je rozšíření – obohacení – realizováno pomocí rozkladu jednotky (partition of unity). Přehledné srovnání těchto metod shrnuli Fries a Belytschko v [FB10]. Ve stručnosti, metody rozkladu jednotky přinesly možnost obohacení prostoru řešení a to velice obecně pro síťové i bezsíťové metody. To umožňuje využít při řešení parciální diferenciální rovnice apriorní znalost o řešení nebo o jeho charakteru a zahrnout ji do modelu při tvorbě prostoru bázových a testovacích funkcí. Tuto znalost můžeme být schopni získat vypočtením řešení lokálního problému. Počáteční motivací metody GFEM bylo spojení PUM a FEM s důrazem na zachování vlastností FEM. Obě metody zatím uvažovaly globální obohacení a až XFEM s použitím znalostí předchozích metod přináší obohacení pouze lokální oblasti. Dnes se metody vzájemně prolínají, a proto se jejich názvy často považují za rovnocenné.

Typickými příklady, kde se XFEM metoda používá, jsou úlohy na výpočet deformace a rozložení napětí v mechanice. Problematická místa jsou ta, kde vlivem namáhání roste napětí a jeho gradient, případně dochází i k trhlinám a napětí se skokově mění. Zároveň se může velikost a poloha trhlin měnit, což v případě FEM (narozdíl od XFEM) nutně znamená měnit výpočetní síť. Takové nespojitosti lze dobře approximovat pomocí obohacení skokovými funkcemi jako jsou $\text{sgn}(x)$ nebo Heavisidova funkce. Dalšími příklady úloh mohou být mísení nestlačitelných kapalin, kde se na jejich rozhraní skokově mění viskozita a důsledkem je nespojitosť v polích tlaku a rychlosti; model tenké membrány, kde se nespojité mění rychlostní pole. Výhodou XFEM oproti klasické konečně-prvkové metodě jsou také nižší nároky na tvorbu sítě, protože vrcholy a hrany elementů nemusí lícovat s místy nespojitosťí a není potřeba zjemňovat síť v těchto problematických místech.

Při konstrukci našeho XFEM modelu budeme vycházet z článku [GR10] R. Gracie a R. Craiga, kde je popsán vícezvodňový model proudění a jeho řešení pomocí metody rozšířených konečných prvků. V řešeném modelu je problémem singularita v bodě zdroje – v místě vrtu – a velké gradienty tlaku v jejím okolí. Analytické řešení jednoho zdroje v nekonečné ploše je logaritmicky závislé na vzdálenosti od zdroje,

proto se řešení v bodě zdroje blíží nekonečnu. Prostor testovacích funkcí obohatíme logaritmickými funkcemi s omezeným definičním oborem (bez bodu singularity) a lokální approximace zavedeme pomocí metody rozkladu jednotky. Podrobně tyto funkce popíšeme v příští podkapitole.

4.1 Obohacující funkce

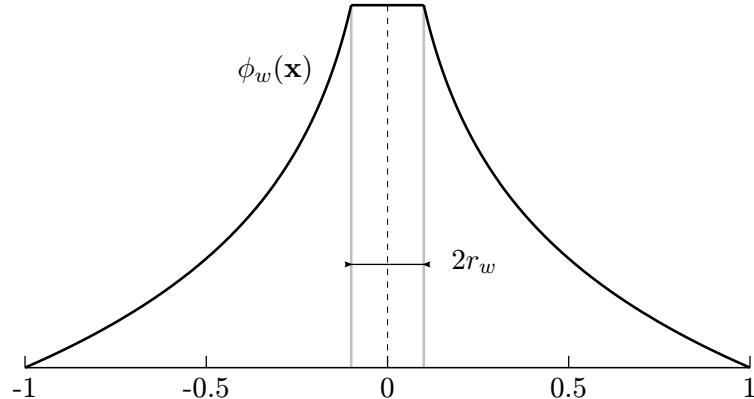
Pro vystihnutí logaritmické singularity v místech vrtů rozšíříme prostor bázových (a testovacích) funkcí o funkci přirozeného logaritmu.

$$\phi_w(\mathbf{x}) = \begin{cases} \log(r_w(\mathbf{x})), & r_w > R_w \\ \log(R_w), & r_w \leq R_w \end{cases} \quad (4.1)$$

kde vzdálenost r_w bodu \mathbf{x} od středu vrtu \mathbf{x}_w určíme pomocí Euklidovy normy

$$r_w(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_w\| = \sqrt{(x - x_w)^2 + (y - y_w)^2}. \quad (4.2)$$

Uvnitř oblasti vrtu je obohacující funkce omezena na konstantní hodnotu, jinak



obr. 3: Obohacující funkce $\phi(\mathbf{x})$.

bychom obsáhli i singulární bod ($\lim_{r \rightarrow \infty} \log(r) = -\infty$ ve středu vrtu). Gradient funkce ϕ vypočteme pomocí pravidel derivování složené funkce

$$\nabla \phi_w(\mathbf{x}) = \nabla \log(r_w(\mathbf{x})) = \frac{1}{r_w(\mathbf{x})} \nabla \|\mathbf{x} - \mathbf{x}_w\| = \frac{\mathbf{x} - \mathbf{x}_w}{r_w^2(\mathbf{x})} \quad \text{pro } r_w > R_w. \quad (4.3)$$

Funkce ϕ_w je konstantní pro $r_w \leq R_w$, tudíž gradient bude v těchto bodech roven $\nabla \phi_w(\mathbf{x}) = 0$.

4.2 Diskretizace

Vyjdeme z dříve odvozené slabé formulace (3.16) a (3.15), nepoužijeme však dále diskrétní approximaci (3.45), ale řešení budeme hledat v jiném tvaru. Diskretizaci lineránimi prvky provedeme shodně jako u klasické FEM v kapitole 3.2, tj. rozdělením sítě na elementy a přechodem ke konečně-prvkové bázi funkcí $\{\varphi_1, \dots, \varphi_N\}$ prostoru $W_{h,0}^{1,2}(\Omega^m)$.

Při diskretizaci je nutné určit, které elementy budou obohacené. Zvolíme určité okolí vrtu o poloměru r_{enr} . Všechny uzly sítě, které spadají do tohoto okolí budou obohacené. Obohacený element je každý takový, na kterém je alespoň jeden obohacený uzel. Parametr r_{enr} je potřeba volit vhodně tak, aby obohacená oblast dobře postihla místa s velkými gradienty, ale zároveň také, aby při větším r_{enr} zbytečně nenarostl počet obohacených stupňů volnosti.

Popišme nejprve obecně princip diskretizace v metodě XFEM. Označme množinu indexů všech uzlů sítě \mathcal{N} a množinu indexů obohacených uzlů \mathcal{N}_e . Řešení hledáme ve tvaru

$$h(\mathbf{x}) = \underbrace{\sum_{j \in \mathcal{N}} \alpha_j \varphi_j(\mathbf{x})}_{\text{FEM}} + \underbrace{\sum_{k \in \mathcal{N}_e} \beta_k \phi(\mathbf{x}) \varphi_k(\mathbf{x})}_{\text{enrichment}}, \quad (4.4)$$

kde α_j jsou stupně volnosti konečných prvků všech uzlů a β_k jsou přidané stupně volnosti obohacených uzlů. Fries popisuje ve svém článku [Fri07] problémy na tzv. rozmývacích (blending) elementech, tedy na elementech, na kterých jsou obohacené pouze některé uzly. V matici se potom objeví nechtěné členy, které se projeví zhoršením podmíněnosti matice výsledného systému a snížením rychlosti konvergence. Zároveň Fries nabízí řešení v podobě modifikované XFEM, ve které se do členu obohacení zavede funkce rampy

$$g(\mathbf{x}) = \sum_{u \in \mathcal{N}_e} \varphi_u(\mathbf{x}) = \sum_{u \in \mathcal{N}} g_u \varphi_u(\mathbf{x}), \quad (4.5)$$

což je součet bázových funkcí φ_u příslušných obohaceným uzlům. Totéž lze zapsat také jako součet přes všechny bázové funkce s použitím vah $g_u = 1$ pro obohacený uzel a nulovými jinde, viz druhá rovnost v (4.5). Funkce rampy slouží jako tzv. rozmývací funkce, která vytváří hladší přechod mezi klasickou a obohacenou částí řešení. Tvar řešení modifikované metody XFEM vidíme v (4.6).

$$h(\mathbf{x}) = \sum_{j \in \mathcal{N}} \alpha_j \varphi_j(\mathbf{x}) + \sum_{k \in \mathcal{N}_e} g(\mathbf{x}) \beta_k \phi(\mathbf{x}) \varphi_k(\mathbf{x}). \quad (4.6)$$

Je zřejmé, že na neobohacené oblasti řešení approximují pouze klasické konečné prvky, na zcela obohacené oblasti je $g = 1$ a formulace je shodná s (4.4). Na rozmývacích elementech funkce g narůstá od nuly do jedné, čímž plynule zesiluje část obohacení. Modifikovanou metodu používá také již citovaný článek od R. Gracie.

Vraťme se k našemu modelu. Každý vrt přispívá vlastní obohacovací funkcí, proto se objeví v řešení suma přes vrty $w \in \mathcal{W}$, kde $\mathcal{W} = \{1, \dots, W\}$. Zavedeme navíc indexovou množinu \mathcal{N}_w , která obsahuje indexy všech obohacených uzlů (přidaných stupňů volnosti) od w -tého vrstu. Takové řešení hledáme opět pro každou zvoden

$$h^m(\mathbf{x}) = \underbrace{\sum_{j \in \mathcal{N}} \alpha_j^m \varphi_j^m(\mathbf{x})}_{\text{FEM}} + \underbrace{\sum_{w \in \mathcal{W}} \sum_{k \in \mathcal{N}_w} g_w^m(\mathbf{x}) \beta_{w,k}^m \phi_w^m(\mathbf{x}) \varphi_k^m(\mathbf{x})}_{\text{enrichment}}. \quad (4.7)$$

Funkci ϕ_w^m můžeme obecně volit různou v jednotlivých zvodeních, ale nemáme k tomu žádný důvod, proto nemusíme psát index zvodeně m . Funkce g_w^m, φ_j^m mohou být také obecně rozdílné v různých zvodeních, závisí totiž na vysíťování zvodení. Pro zjednodušení však také můžeme předpokládat, že sítě jsou pře všechny zvodeně shodné, poloměr obohacení zvolíme jednotný v rámci vrstu a pak nemusíme dále v této kapitole psát index m .

Zavedeme nyní testovací funkce

$$\begin{aligned} \varphi_i & \quad \text{pro } i \in \mathcal{N} \\ z_{w,k} = g_w \phi_w \varphi_k & \quad \text{pro } k \in \mathcal{N}_w, w \in \mathcal{W}. \end{aligned} \quad (4.8)$$

S takto zavedeným řešením v (4.7), testovacími funkcemi (4.8) a indexovými množinami můžeme systém zapsat pomocí matic, jak nyní ukážeme.

4.3 Maticový zápis

Začneme obdobně jako u modelu řešeného pomocí FEM úlohou na jedné zvodi v kapitole 3.3. Řešenou soustavu můžeme opět zapsat pomocí matic, ovšem nyní bude struktura komplikovanější o prvky, do kterých přispívá obohacení. Pro zjednodušení předpokládejme, že obohacení nezasahuje hranici oblasti Γ_D s Dirichletovou okrajovou podmínkou (je tvořena tedy pouze lineárními konečnými prvky a druhý řádek pravé strany rovnice (4.9) pak bude nulový) a také zdroje f v oblasti jsou mimo oblast obohacení. Tím jsme určili, že prvky q_i a d_w budou shodné s prvky v (3.49).

$$\begin{pmatrix} \mathbf{A} & \mathbf{R}^T & \mathbf{C}_a^T \\ \mathbf{R} & \mathbf{S} & \mathbf{C}_b^T \\ \mathbf{C}_a & \mathbf{C}_b & \mathbf{D} \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{q} \\ \mathbf{0} \\ \mathbf{d} \end{pmatrix} \quad (4.9)$$

Jednotlivé prvky matice v (4.9) jsou rovny

$$\begin{aligned} A_{ij} &= T \int_{\Theta} \nabla \varphi_j \cdot \nabla \varphi_i \, d\mathbf{x} + \sum_w^W \sigma_w \int_{\partial B_w} \varphi_j \varphi_i \, d\mathbf{x} \quad \forall i, j \in \mathcal{N} \\ R_{(w,k),j} &= T \int_{\Theta} \nabla \varphi_j \cdot \nabla z_{w,k} \, d\mathbf{x} + \\ &\quad + \sum_{s \in \mathcal{W}} \sigma_s \int_{\partial B_s} \varphi_j z_{w,k} \, d\mathbf{x} \quad \forall w \in \mathcal{W}, \forall k \in \mathcal{N}_w, \forall j \in \mathcal{N} \\ S_{(w_i, k_i), (w_j, k_j)} &= T \int_{\Theta} \nabla z_{w_j, k_j} \cdot \nabla z_{w_i, k_i} + \\ &\quad + \sum_{s \in \mathcal{W}} \sigma_s \int_{\partial B_s} z_{w_j, k_j} z_{w_i, k_i} \, d\mathbf{x} \quad \forall w_i, w_j \in \mathcal{W}, \forall k_i, k_j \in \mathcal{N}_w, \\ C_{wj}^a &= -\sigma_w \int_{\partial B_w} \varphi_j \, d\mathbf{x} \quad \forall w \in \mathcal{W}, \forall j \in \mathcal{N}, \\ C_{w,(w_j, k_j)}^b &= -\sigma_w \int_{\partial B_w} z_{w_j, k_j} \, d\mathbf{x} \quad \forall w, w_j \in \mathcal{W}, k_j \in \mathcal{N}_w, \\ D_{ww} &= \left[\int_{\partial B_w} \sigma_w \, d\mathbf{x} + c_w^2 \right] \quad \forall w \in \mathcal{W}, \\ q_i &= \int_{\Theta} f \varphi_i \, d\mathbf{x} - \int_{\Theta} T \nabla h_D \cdot \nabla \varphi_i \, d\mathbf{x} \quad \forall i \in \mathcal{N}, \\ d_w &= c_w^2 H_w^2 \quad w \in \mathcal{W}. \end{aligned} \quad (4.10)$$

Dvojice indexů (w, k) v (4.9) jednoznačně určuje řádek (a sloupec) matic – odpovídá stupni volnosti $\beta_{w,k}$ ve vektoru \mathbf{b} a zároveň testovací funkci $z_{w,k}$. Pokud označíme celkový počet obohacených stupňů volnosti $N_e = \sum_{w \in \mathcal{W}} N_w$, můžeme zapsat rozměry matic $\mathbf{A}^{N \times N}$, $\mathbf{S}^{N_e \times N_e}$, $\mathbf{R}^{N \times N_e}$, $\mathbf{C}_a^{N \times W}$, $\mathbf{C}_b^{N_e \times W}$, $\mathbf{D}^{W \times W}$. Poznamenejme, že všechny matice jsou řídké, \mathbf{A} a \mathbf{S} jsou navíc symetrické a \mathbf{D} je diagonální. Stupně volnosti lineárních prvků jsou ve vektoru $\mathbf{a} = (\alpha_1, \dots, \alpha_N)$, stupně volnosti obohacení $\mathbf{b} = (\beta_{1,1}, \dots, \beta_{w,k}, \dots, \beta_{W,N_W})$ a stupně volnosti na vrtech $\mathbf{p} = (H_1, \dots, H_W)$.

Pokud bychom chtěli opět model rozšířit na více zvodní tak jako v 3.3, bude soustava také blokově diagonální a doplněna o ty samé bloky $\mathbf{E}^m = \text{diag}(-c_1^m, \dots, -c_W^m)$. K prvkům matice \mathbf{C}_a^m , \mathbf{C}_b^m budou přičteny navíc konstanty propustnosti ve vrtu

$(c_w^{m+1} + c_w^m)$ a matice systému bude mít tvar

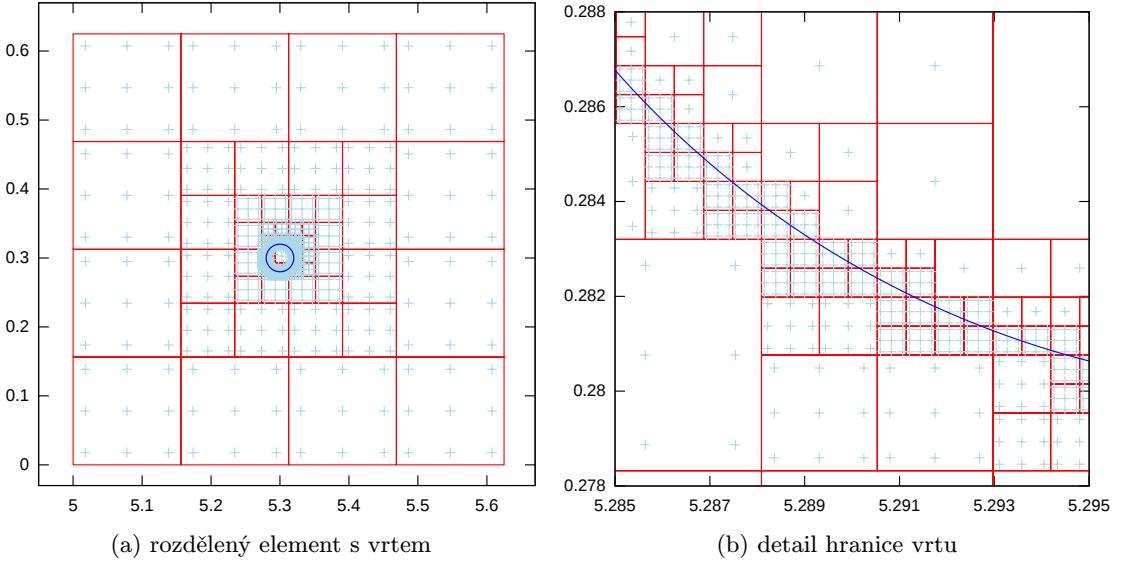
$$\begin{pmatrix} \mathbf{A}^M & (\mathbf{R}^M)^T & (\mathbf{C}_a^M)^T & & \dots \\ \mathbf{R}^M & \mathbf{S}^M & (\mathbf{C}_b^M)^T & & \dots \\ \mathbf{C}_a^M & \mathbf{C}_b^M & \mathbf{D}^M & \mathbf{E}^M & \dots \\ & & & \mathbf{A}^{M-1} & (\mathbf{R}^{M-1})^T & (\mathbf{C}_a^{M-1})^T & \dots \\ & & & \mathbf{R}^{M-1} & \mathbf{S}^{M-1} & (\mathbf{C}_b^{M-1})^T & \dots \\ & & \mathbf{E}^M & \mathbf{C}_a^{M-1} & \mathbf{C}_b^{M-1} & \mathbf{D}^{M-1} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \end{pmatrix} \quad (4.11)$$

Vektor pravé strany bude podobný jako ve FEM metodě, pouze v případě přítomnosti jiných zdrojů nebo Dirichletovy okrajové podmínky přibudou obohacující příspěvky. V implementaci budeme uvažovat obohacení pouze uvnitř oblasti, tzn. okrajová podmínka bude řešena pouze lineráními prvky. Vektor hledaných stupňů volnosti je roven $(\mathbf{a}^M, \mathbf{b}^M, \mathbf{p}^M, \mathbf{a}^{M-1}, \mathbf{b}^{M-1}\mathbf{p}^{M-1}, \dots, \mathbf{a}^1, \mathbf{b}^1, \mathbf{p}^1)$.

4.4 Adaptivní integrace na obohacených elementech

Na obohacených elementech je potřeba přesně integrovat, a to zejména v okolí vrtů v místech velkých gradientů. Dalším důvodem je nespojitost derivace obohacující funkce na obvodu vrtu. Zvolili jsme způsob navržený v článku [GR10]. Pokud na obohaceném elementu neleží vrt, je volena 4-bodová Gaussova kvadratura. Pokud zde vrt leží, je oblast elementu dělena na malé podoblasti (v našem případě čtverce, neboť dělíme čtvercový referenční element). Kritérium pro rozdelení čtverce je průnik s hranicí vrtu. Oblastem mimo vrt a těm, které po posledním dělení (zvoleno 12 úrovní) leží na hranici vrtu, je přidělena 3-bodová Gaussova kvadratura a oblastem uvnitř vrtu 1-bodová Gaussova kvadratura (obohacená funkce zde vrací konstantu, proto zde stačí jedna hodnota). Dělení dokumentuje obr. 4, na kterém vidíme síť čtverců na elementu vzhledem k vyznačenému vrtu. Na podobrázku 4b si potom můžeme všimnout detailu hranice vrtu, kde vidíme i kvadraturní body rozmístěné tak, jak jsme popsali výše.

Kritérium dělení je možné změnit, stačí implementovat pouze jinou rozhodovací funkci pro označování čtverců k rozdelení. Například by bylo zajímavé založit dělení na základě měření rozdílu vypočtené hodnoty integrálu mezi jednotlivými kroky zjemnění a porovnat, zdali takto lépe vystihneme hranici vrtu a zpřesníme integraci.



obr. 4: Adaptivní zjemnění elementu v okolí vrtu v metodě XFEM (červeně dělení, modře hranice vrtu, křížky kvadraturní body).

Byla navržena samostatná třída `AdaptiveIntegration`, ve které je řešeno zjemanování dle popsaného kritéria, mapování na zjemněné čtverce a také samotný výpočet lokální matice obohaceného elementu.

Mapování se projeví při integraci ve formě jakobiánu, ale také přeskálováním gradientů. Mějme obecnou mapovací funkci ve tvaru

$$\mathbf{x} = \mathbf{M}(\hat{\mathbf{x}}) = \mathbf{Z}\hat{\mathbf{x}} + \mathbf{t} \quad (4.12)$$

kde \mathbf{t} je vektor translace a \mathbf{Z} je maticí obecného zobrazení. Jakobiho matice tohoto mapování a její determinant (jakobián) jsou ve dvou dimenzích určeny vztahy

$$\mathbf{J}(\mathbf{M}(\hat{\mathbf{x}})) = \begin{pmatrix} \frac{\partial x_1}{\partial \hat{x}_1} & \frac{\partial x_1}{\partial \hat{x}_2} \\ \frac{\partial x_2}{\partial \hat{x}_1} & \frac{\partial x_2}{\partial \hat{x}_2} \end{pmatrix}, \quad \det \mathbf{J}(\mathbf{M}(\hat{\mathbf{x}})) = \det \mathbf{Z}. \quad (4.13)$$

Pokud počítáme gradient $\nabla_x f(\mathbf{x})$ na mapované oblasti (typicky přesun z reálných souřadnic \mathbf{x} na souřadnice referenčního elementu $\hat{\mathbf{x}}$), musíme přeskálovat gradient na referenčním elementu zobrazovací maticí. To uvidíme nejlépe ve vyjádření po složkách

$$(\nabla_x f(\mathbf{x}))_i = \frac{\partial f(\mathbf{x})}{\partial x_i} = \sum_{k=1}^2 \frac{\partial \hat{f}(\hat{\mathbf{x}})}{\partial \hat{x}_k} \frac{\partial \hat{x}_i}{\partial x_k}, \quad \text{kde } \frac{\partial \hat{x}_i}{\partial x_k} = Z_{ik}^{-1},$$

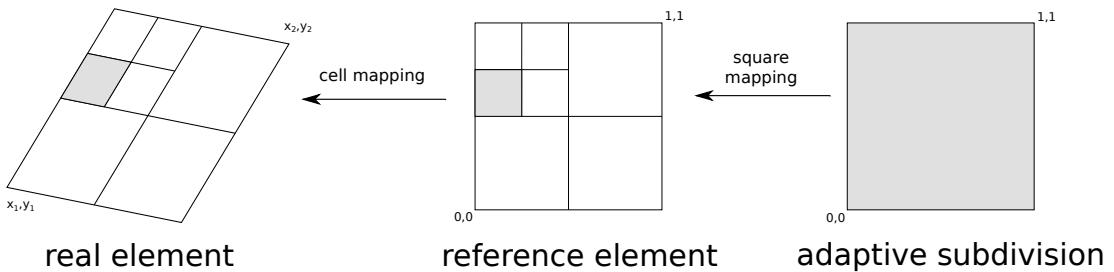
takže můžeme psát

$$\nabla_x f(\mathbf{x}) = \mathbf{Z}^{-1} \nabla_{\hat{x}} \hat{f}(\hat{\mathbf{x}}). \quad (4.14)$$

Mapování mezi referenčním (čtvercovým) elementem a zjemněným čtvercem je jednoduchým případem výše uvedeného mapování

$$\hat{\mathbf{x}} = M_s(\mathbf{x}^*), \quad \mathbf{Z}_s = \begin{pmatrix} d & 0 \\ 0 & d \end{pmatrix}, \quad \mathbf{Z}_s^{-1} = \begin{pmatrix} \frac{1}{d} & 0 \\ 0 & \frac{1}{d} \end{pmatrix}, \quad \det \mathbf{Z}_s = d^2, \quad (4.15)$$

kde d je délka strany čtverce a jakobián je tedy roven ploše daného čtverce d^2 . Vektor translace \mathbf{t} je za předpokladu zachování orientace čtverce roven vzdálenosti levého dolního vrcholu od počátku.



obr. 5: Mapování mezi reálným, referenčním elementem a jednotkovým čtvercem.

Mapování mezi reálným a referenčním elementem označme $M_e(\hat{\mathbf{x}})$. Hodnoty obohacujících funkcí vyhodnocujeme v reálných souřadnicích a lineární bázové funkce vyhodnocujeme na referenčním elementu. Souřadnice kvadraturních bodů získáme přemapováním z čtverce na referenční element a případně dále na reálný element, tudíž do vztahů (4.7) a (4.8) dosazujeme

$$\begin{aligned}\varphi(\mathbf{x}) &= \hat{\varphi}(\hat{\mathbf{x}}) = \varphi(M_s(\mathbf{x}^*)), \\ \phi(\mathbf{x}) &= \phi(M_e(M_s(\mathbf{x}^*))), \\ \nabla_x \varphi(\mathbf{x}) &= \mathbf{Z}_e^{-1} \nabla_{\hat{x}} \hat{\varphi}(M_s(\mathbf{x}^*))\end{aligned}$$

Pokud ještě označíme váhy kvadraturních bodů W_q , dostaneme v rovnici (4.20) plošné intergrály ve tvaru

$$\int_{\Omega^m} (\dots) \det \mathbf{J}(M_e(\hat{\mathbf{x}})) \det \mathbf{J}(M_s(\mathbf{x}^*)) W_q d\mathbf{x}^* = 0 \quad (4.16)$$

Poznamenejme, že pro mapování gradientu v tomto případě nemůžeme použít objekt knihovní třídy `FEValues`, protože ten má předpočítané pouze hodnoty v kvadraturních bodech pro standardní element. K tomuto problému se vrátíme v kapitole o implementaci 5.3.2.

4.5 Vrty jako zdroje

Pokusili jsme se také model přeformulovat a zvolit jiný přístup popisu komunikace mezi vrty. Na vrty můžeme pohlížet jako na zdroje v oblasti Ω^m , které vyjádříme jako tok plochou zvodně v místech vrtů. Potom se na pravé straně rovnice (2.3) objeví nenulový člen

$$-T^m \Delta h^m = -\sum_{w=1}^W \eta_w^m (h^m(\mathbf{x}) - H_w^m) \quad \text{na } \Omega^m \text{ pro } m = 1, \dots, M, \quad (4.17)$$

kde η [s^{-1}] je koeficient propustnosti mezi vrtem a zvodní (vyjadřuje tok $1 \text{ m}^3 \text{s}^{-1}$ plochou 1 m^2 při tlakovém spádu 1 m). Všimněme si, že nyní neodebíráme plochy B_w^m z oblasti zvodně, ale rovnici řešíme na celé Ω^m .

4.5.1 Odvození slabé formulace

Při odvození postupujeme shodně jako v odvození v kapitole 3. Vrty si však nyní představujeme jako plošné zdroje, nikoli zadané jako okrajové podmínky, a řešíme rovnici (4.17) na celé oblasti Ω^m . Prostor funkcí řešení $\mathbf{h}, \mathbf{h}_0, \mathbf{h}_D$ a testovacích funkcí \mathbf{v} vezmeme shodný, tedy $\mathbf{h}_0, \mathbf{v} \in V$ a $\mathbf{h}_D \in V_D$, definovali jsme v (3.8). Vynásobme tedy (4.17) testovací funkcí, integrujme přes oblast Ω^m a dosadme za \mathbf{h} , abychom dostali

$$\begin{aligned} \int_{\Omega^m} T^m \nabla h_0^m \cdot \nabla v^m \, d\mathbf{x} &= -\sum_w^W \int_{B_w^m} \eta_w^m (h_0^m - H_w^m) v^m \, d\mathbf{x} + \\ &\quad + \int_{\Omega^m} f^m v^m \, d\mathbf{x} - \int_{\Omega^m} T^m \nabla h_D^m \cdot \nabla v^m \, d\mathbf{x} \end{aligned} \quad (4.18)$$

Zavedeme pomocnou testovací funkci

$$\mu_w^m(\mathbf{x}) = \begin{cases} 1, & r_w \leq R_w \\ 0, & r_w > R_w, \end{cases} \quad (4.19)$$

která bude nenulová pouze v oblasti vrtu a díky které můžeme sumu integrálů přes vrty v (4.18) přepsat pod integrál přes celou oblast. Tím dostaneme slabou formulaci ve tvaru (4.20).

$$\begin{aligned} \int_{\Omega^m} \left(T^m \nabla h_0^m \cdot \nabla v^m + \sum_w^W \mu_w^m \eta_w^m (h_0^m - H_w^m) v^m \right) \, d\mathbf{x} &= \\ &= \int_{\Omega^m} (f^m v^m - T^m \nabla h_D^m \cdot \nabla v^m) \, d\mathbf{x} \end{aligned} \quad (4.20)$$

V rovnici na vrtu (3.16) nahradíme křivkový integrál vyjadřující tok do zvodně plošným a dostaneme s $\sigma_w^m [s^{-1}]$ rovnici

$$\begin{aligned} \int_{B_w^m} \eta_w^m (h_0^m - H_w^m) \bar{v}_w^m d\mathbf{x} = \\ = c_w^{m+1} (H_w^m - H_w^{m+1}) \bar{v}_w^m - c_w^m (H_w^{m-1} - H_w^m) \bar{v}_w^m \quad (4.21) \end{aligned}$$

4.5.2 Výhody modelu se zdroji

Vzhledem k tomu, že již počítáme plošný integrál na adaptivně zjemněném obohaceném elementu pro člen s gradienty bázových funkcí, stačí navíc napočítat funkční hodnoty a přičíst výraz do lokální matice. Toho nemůžeme využít v případě výpočtu hraničního integrálu, neboť potom počítáme funkční hodnoty v jiných bodech na hranici vrtu. Zjednodušíme tím tedy asemblaci obohacených elementů s vrty. Zároveň nemusíme při vyhledávání obohacených elementů přiřazovat body na hranici vrtu příslušným elementům – pokud vrt zasahuje do více elementů.

Je otázkou, zdali je tento způsob i rychlejší. To záleží na počtu zvolených bodů na hranici a na počtu úrovní, přesněji na počtu bodů adaptivní integrace, který bude pravděpodobně ve více případech vyšší.

Další otázkou je potom při reálné úloze volba parametru η , který číselně ani jednotkou neodpovídá parametru σ . Z testů pak vyšlo najev, že se úlohy v podstatě shodují pro $\eta > 10^4$, pro která již bereme tlak na hranici a uvnitř vrtu za shodný.

5 Implementace

Cílem této kapitoly je představit implementační část diplomové práce. Pro základní práci s konečnými prvky, stavbu sítě, operace s maticemi a vektory, řešení soustav a výstupy modelů byla použita knihovna Deal II [BHK07] ve verzi 7.2.0 (dále v textu myslíme knihovnou právě tuto). Její rozmanité nástroje zjednodušíly rutinní záležitosti a po nastudování knihovních funkcí bylo možné zabývat se plně vývojem výše popsaných modelů. Elementy sítě knihovny vycházejí z jednotkového čtverce, není možné zde použít trojúhelníky či méně tradiční prvky, což pro účely této práce nebylo žádným omezením. Poznamenejme, že knihovna zatím nijak nepodporuje XFEM. Nesporným kladem knihovny je také její perfektně zpracovaná dokumentace systémem Doxygen.

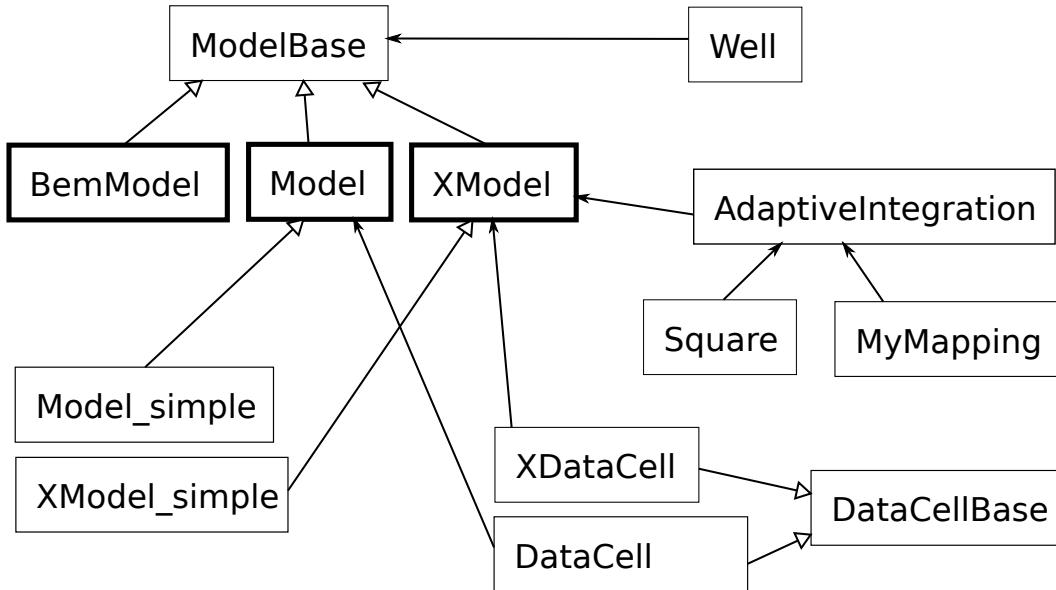
Implementaci provedeme v C++, pro tvorbu a zobrazování sítě použijeme volně dostupný software GMSH [GR09], pro vizualizace a postprocessing vypočtených modelů software Paraview (3.98), též volně dostupný (<http://www.paraview.org/>).

5.1 Základní struktura tříd

Pro implementaci modelů a jejich algoritmů jsme navrhli systém několika tříd. Popišme nyní jejich skladbu, vztahy a stručně také jejich metody. Obrázek 6 znázorňuje základní strukturu tříd. Pro detailní přehled tříd a jejich metod je nutné nahlédnout do zdrojových kódů nebo do automaticky generované dokumentace systémem Doxygen.

Pro každý model budeme implementovat vlastní třídu, ale některé funkčnosti můžeme zobecnit pro všechny. Proto vytvoříme abstraktní třídu `ModelBase`, která v podobě virtuálních metod definuje základní strukturu tříd modelů a zároveň implementuje jednoduché metody pro základní nastavení parametrů modelu. Dohromady máme tři základní modely ve třídách `Model`, `Bem_model` a `XModel`, které odpovídají po řadě použitým metodám FEM, BEM a XFEM. Vstupem do modelů jsou geometrie, data na vrtech a fyzikální parametry modelu. Třídy `Model_simple` a `XModel_simple` jsou určeny speciálně pro úlohu s jednou zvodní a jedním vrtem a implementují navíc pouze konkrétní Dirichletovu okrajovou podmínu.

Pro vrty je vytvořena vlastní třída `Well`, která obsahuje základní rozhraní pro data na vrstu – propustnosti, poloměr a střed vrstu a tlak v ústí vrstu. Zároveň umožňuje napočítat body na hranici vrstu, potřebné při asemblaci hraničního integrálu.



obr. 6: Struktura tříd (prázdné šipky znační vztah potomek-předek).

Pro XFEM metodu navíc vrací metodami `xshape_value` a `xshape_grad` hodnoty obohacujících funkcí a jejich gradientů.

Další třídou je `DataCellBase`, která k jednotlivým elementům doplňuje potřebná data. Používáme ji k propojení dat na vrtech se sítí – obsahuje informace, které vrty zasahují do daného elementu, která část hranice vrtu elementem prochází a v případě XFEM metody indexy nových stupňů volnosti. Pro metodu FEM a XFEM jsou odvozeni specifickí potomci této třídy – `DataCell` a `XDataCell`, blíže je popíšeme dále.

Třída `AdaptiveIntegration` implementuje adaptivní integraci na elementu, popsanou v kapitole 4.4.

Výpočet modelu probíhá v několika krocích, které odpovídají virtuálním metodám ve třídě `ModelBase`

- `make_grid`
- `refine_grid`
- `setup_system`
- `assemble_system`
- `assemble_dirichlet`
- `solve`.

Posloupnost těchto metod je volána při spuštění výpočtu modelu v metodě `run`.

Nejprve vytvoříme výpočetní síť. Můžeme ji načíst ze souboru, vytvořit pomocí globálního rovnoměrného zjemňování (v případě FEM a XFEM), pracujeme se základními dvěma tvary oblasti, a to obdélníkem a kruhem. Poznamenejme, že pokud máme síť s volnými uzly, není uložení a načtení takové sítě jednoduchým úkolem. Problém spočívá v tom, že soubory GMSH neobsahují sousednosti elementů. V knihovně Deal II můžeme použít speciální třídu `PersistentTriangulation`, díky které můžeme uložit počáteční hrubou síť a k tomu navíc soubor s příznaky zjemnění/zhrubování elementů po jednotlivých úrovních. Při načítání sítě pak musíme načíst i soubor s příznaky a poté dokážeme síť zrekonstruovat do zvolené úrovně zjemnění.

Pro některé testovací úlohy bude potřebné umožnit spouštění v cyklu, například pro sledování chování při změně parametru. Proto jedním z argumentů některých metod je číslo cyklu. Zároveň můžeme v každém cyklu provést zjemnění sítě metodou `refine_grid`, čehož využíváme především v modelu FEM s adaptivní sítí.

V metodě `setup_system` distribuujeme stupně volnosti a nastavujeme rozměry systémové matice, vektorů pravé strany a řešení. Vytváříme přitom objekt třídy `SparsityPattern`, který je vzorem zaplnění řídké systémové matice (obsahuje jedničky pouze v místech nenulových prvků).

V metodě `assemble_system` provádíme vlastní výpočty integrálů a dalších částí rovnic a vše dosazujeme do příslušných prvků matice a pravé strany. Pokud máme definovanou na části hranice oblasti Dirichletovu krajovou podmínu, voláme ještě před ukončením metody asemblace navíc metodu `assemble_dirichlet`.

Poté, co je celý systém sestaven, tak jej můžeme v metodě `solve` vyřešit pomocí vybrané numerické metody – v našem případě sdružených gradientů (CG) nebo zobecněné metody minimálních reziduí (GMRES). Můžeme také použít také některý z nabízených předpodmiňovačů pro zvýšení stability a rychlosti konvergence výpočtu.

V poslední řadě potřebujeme vypočtené řešení nějakým způsobem vystoupit. Knihovna disponuje několika metodami, které zpracovávají výstup. Je možné řešení zapsat do několika formátů včetně formátu GMSH a VTK, pro jehož vizualizaci a postprocessing používáme software Paraview. V metodě `output_results` ukládáme řešení na výpočetní síti do souboru ve zvoleném formátu. Pokud chceme vystoupit

řešení na jiné síti než výpočetní, předáme přímo objekt triangulace nebo soubor sítě metodě `output_distributed_solution`.

Kód 1: Ukázka použití modelu

```

1  double well_radius = 0.02 ,
2      perm2fer = 1e5 ,
3      perm2tard = 1e10 ,
4      pressure_at_top = 2;
5  Point<2> well_center(0,0),
6      down_left(-10,-10),
7      down_right(10,10);
8
9  Well *well = new Well(well_radius, well_center,
10                         perm2fer, perm2tard);
11 well->set_pressure(pressure_at_top);
12 well->evaluate_q_points(100);
13
14 XModel xmodel(well, "My_XFEM_Model", 1);
15 xmodel.set_area(down_left, up_right);
16 xmodel.set_output_dir("output");
17 xmodel.set_transmissivity(1e-4,0);
18 xmodel.set_refinement(2);
19 xmodel.set_adaptivity(true);
20 xmodel.set_enrichment_radius(2.0);
21
22 for (unsigned int cycle=0; cycle < 5; ++cycle)
23 { xmodel.run(cycle);
24     xmodel.output_results(cycle); }
```

Uvedený kód 1 ukazuje, jak třídu modelu s metodou XFEM použijeme k výpočtu problému s jednou zvodní a jedním vrtem. Nejprve definujeme vlastnosti vrtu, geometrii pomocí protilehlých vrcholů obdélníku (kód 1, ř. 1-8). Poté vytváříme objekt vrtu, předepisujeme tlak na vrtu a počítáme body na hranici vrtu. Dále vytváříme a nastavujeme objekt modelu, parametry jsou zřejmé z názvů metod, a nakonec model pustíme v 5 cyklech (kód 1, ř. 22-24). Protože jsme povolili adaptivitu (ř. 19), tak se v každém cyklu volá metoda `refine_grid` a provede se globální zjemnění sítě.

Nakonec uvedeme několik tříd knihovny, které budeme hlavně používat v implementaci a též zmiňovat dál v textu. Pro práci se sítí budeme používat třídy `Triangulation` a `PersistentTriangulation`, třída `FE_Q` bude reprezentovat vybraný konečný prvek (volíme lineární), třída `DofHandler` se bude starat o indexaci a distribuci stupňů volnosti, třída `QGauss` bude obsahovat kvadraturní body a jejich váhy v referenčním elementu a nakonec třída `FEValues` bude všechny zmíněné objekty spojovat a bude schopna vracet hodnoty a gradienty bázových funkcí, ma-

pování mezi elementem a jeho referenčním protějškem a další. Všechny třídy jsou šablonami s parametrem dimenze, který volíme roven dvěma, neboť síťujeme 2D oblast zvodně.

5.2 FEM model s adaptivně zjemňovanou sítí

V následujících odstavcích popíšeme implementaci virtuálních metod pro model používající metodu lineárních konečných prvků na adaptivně zjemňované síti.

5.2.1 Tvorba sítě

K vytvoření výpočetní sítě jsme zavedli přepínač `grid_create_type`, který nabývá hodnoty `load` pro načtení sítě ze souboru, hodnoty `load_circle` pro načtení kruhové sítě známé velikosti ze souboru, `circle` pro vytvoření kruhové sítě a `rect` pro obdélníkovou síť. Problém načítání sítě s volnými uzly jsme již popsali dříve. Pro práci se sítí plně využíváme možností knihovny, podrobné informace lze najít v dokumentaci tříd `Triangulation` a `PersistentTriangulation`.

5.2.2 Vyhledání elementů s vrty

Při sestavování systémové matice budeme potřebovat vypočítat hraniční integrály přes hranici vrtu (kružnici). V obecném případě může hranice vrtu procházet několika elementy a tím pádem se v různých částech integrace objeví různé bázové funkce. Musíme vyřešit geometrickou úlohu průniku kružnice a elementu, který zde představuje čtyřúhelník.

Vzhledem k tomu, že po hranici vrtu budeme nakonec integrovat, je nasnadě approximovat kružnici pomocí mnohoúhelníku. Počet jeho vrcholů musí být dostatečný, aby vrcholy zasáhly do všech elementů, kterými kružnice skutečně prochází. Informaci, jaký vrt prochází daným elementem a jaké jsou na něm body hranice, uložíme do objektu třídy `DataCell`. Poté využijeme možnost sítě knihovny přiřadit libovolný ukazatel k elementu sítě, a to právě ukazatel na zmíněný objekt.

Vyhledávání elementů provádí metoda `find_well_cell`. Nejprve najde element, ve kterém leží střed daného vrstu, a poté volá rekurzivní metodu `add_data_to_cell`. Ta testuje každý okolní element, vytváří datové objekty a přiřazuje jejich ukazatele elementů. Využijeme opět možnost sítě – přidání příznaku k elementu – a označíme tak elementy, které jsme již prošli. Testem vzdálenosti uzel od středu vrstu pak

zajistíme, že po nalezení elementu se středem vrstu prochází pouze elementy uvnitř a na hranici vrstu, nikoli celou síť.

Výsledkem je potom vektor objektů `DataCell`, ve kterém máme informace do jakých elementů vrty zasahují a k jakým stupňům volnosti přispívají.

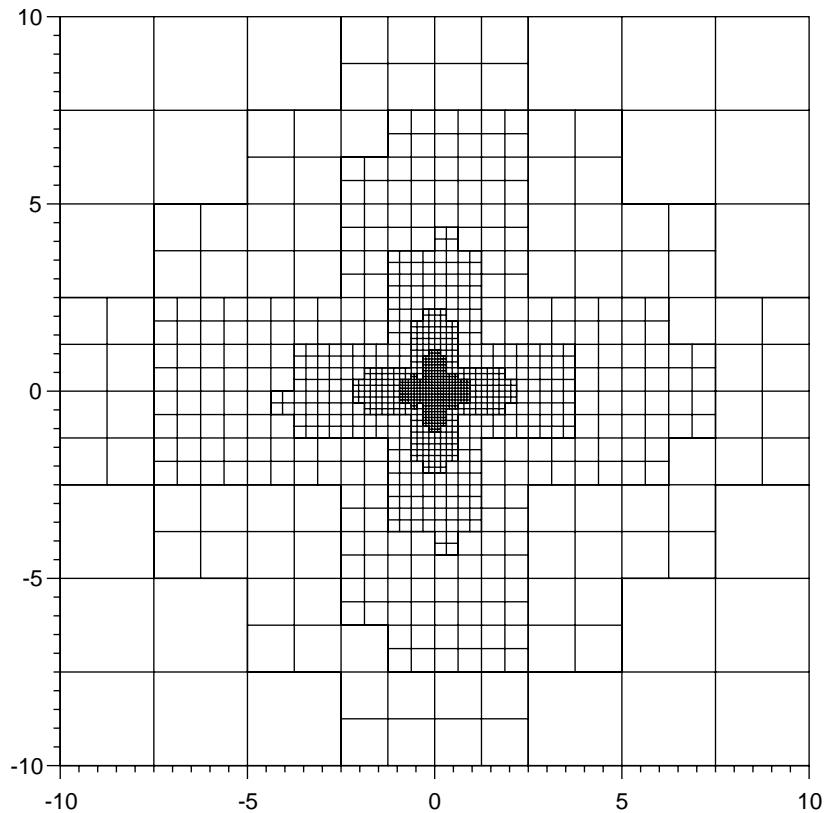
5.2.3 Sestavení matice systému

V metodě `setup` vytváříme tzv. vzor řídké matice soustavy rovnic pomocí objektu `BlockSparsityPattern`, čímž definujeme zaplnění systémové matice, která je pomocí tohoto objektu inicializována. Zároveň inicializujeme i vektor řešení a pravé strany. O rozmištění stupňů volnosti se postará objekt třídy `DofHandler`, o rozmištění příspěvků z rovnic na vrtech se musíme postarat zvlášť. Také se v této metodě řeší distribuce volných uzlů, což popíšeme v následující podkapitole.

5.2.4 Adaptivita

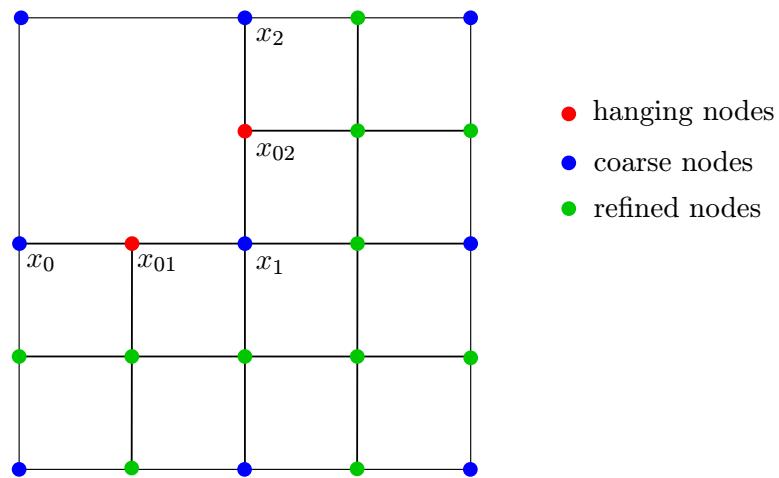
Pro lokální zjemňování sítě využíváme opět prostředky poskytnuté knihovnou. Pro zobecněnou Poissonovu rovnici je zde navržena třída `KellyEstimator` odhadující chybu řešení na základě výpočtu rozdílů gradientů na hranicích elementů. Výstupem metody této třídy je vektor chyb na jednotlivých elementech, který můžeme předat spolu s triangulací třídě `GridRefinement`. Ta na základě nastavených parametrů – procento zjemněných a zhrubovaných elementů – provede lokální změny sítě na určených elementech. Příklad zjemněné sítě na úloze s jedním vrtem vidíme na obrázku 7. Podrobný popis najdeme v dokumentaci knihovny u uvedených tříd.

Při lokálním zjemňování sítě vznikají tzv. volné uzly (hanging nodes) – uzly, které leží ve středu stěn vedlejších nezjemněných elementů, viz obr. 8. Abychom zajistili spojitost řešení i v těchto uzlech, musíme splnit omezující podmínky pro stupně volnosti. Pro volné uzly na obrázku jsou to tyto $x_{01} = \frac{1}{2}(x_0 + x_1)$ a $x_{02} = \frac{1}{2}(x_0 + x_2)$. Dokumentace říká, že omezující podmínky jsou obecně lineární a mohou být případně nehomogenní ve tvaru $x_{i1} = \sum_{j=2}^M a_{ij}x_{ij} + b_i$. Knihovna nabízí řešení tohoto problému ve třídě `ConstraintMatrix`, která na základě znalosti triangulace vytvoří omezující podmínky pro stupně volnosti v okolí volných uzlů. Sestavení globální matice a vektoru pravé strany provedeme stejným způsobem, pouze po jejím naplnění je potřeba odstranit nadbytečné stupně volnosti ve volných uzlech metodou `condense` a po výpočtu je zpět dopočítat pomocí metody `distribute`.



obr. 7: Lokálně zjemňovaná síť v osmi úrovních.

Systém rovnic v našem modelu je rozšířený o rovnice na vrtech a tudíž i na ně je potřeba aplikovat uvedené podmínky pro volné stupně volnosti. Metody třídy `ConstraintMatrix` můžeme použít na celý systém, a proto v našem případě nepřináší volné uzly žádný další problém.



obr. 8: Volné uzly v lokálně zjemněné síti.

Lokální zjemňování a tvorba volných uzlů ovšem naruší vyhledávání elementů

s vrty v metodě `find_well_cells`. Sousední elementy se mohou lišit úrovní zjemnění, a proto musíme při iterování přes sousedy postupovat obezřetněji. Problém iterování přes sousední elementy v síti s volnými uzly je částečně řešen v příkladu Step30 tutoriálu knihovny Deal II. Podstatou je porovnávání úrovně zjemnění sousedních elementů a testování, zdali je hranice elementu dále dělená nebo není.

5.3 XFEM model

Model řešený metodou XFEM je implementován ve třídě `XModel`. Mimo základní poddělené funkce nabízí také nastavení parametru r_{enr} , určujícího velikost obohacené oblasti a popsáno v kapitole 4.2.

Začínáme opět tvorbou sítě tak, jak jsme popsali v modelu s FEM. Ponechali jsme všechny možnosti načtení a vytvoření sítě včetně sítí s volnými uzly. Jelikož ale zatím neumíme řešit metodu XFEM na takové síti, předpokládáme, že oblast obohacení pokrývá pouze elementy shodné úrovně zjemnění a volné uzly se vyskytují jenině mimo tuto oblast. Problém volných uzlů a XFEM popisuje Fries se spoluautory v článku Hanging Nodes and XFEM [FBA⁺11]. Pro další rozvoj bude nepochybně užitečné tento problém také umět vyřešit.

Dále popíšeme realizaci obohacování a sestavení systémové matice.

5.3.1 Realizace obohacení

Obdobně jako u FEM modelu funkce `find_well_cells` realizovala vyhledání elementů přímo komunikujících s vrty, u tohoto modelu funkce `find_enriched_cells` vyhledává elementy k obohacení. Pro každý vrt nalezně element, na němž leží střed vrta, a poté rekursivní funkcí `enrich_cell` hledá přes sousední elementy všechny takové, které splňují podmínu obohacení – uzly ležící ve vzdálenosti do r_{enr} .

Stejně tak jako u předchozího modelu, potřebujeme rozmístit body zdiskretizované hranice vrta mezi elementy, kterými hranice vrta prochází. Pokud použijeme model s vrty branými jako plošné zdroje, problém diskretizace hranice vrta odpadá.

Obohaceným elementům přiřadíme ukazatel na objekt `XDataCell`, obdobně jako v modelu s FEM. Tento objekt navíc obsahuje rozmístění a indexy stupňů volnosti pocházejících z obohacení.

5.3.2 Sestavení matice systému

Inicializace v metodě `setup` probíhá obdobně, jako v předchozím modelu, bloková struktura systémové matice je ovšem komplikovanější, jak jsme ukázali v rovnici (4.9) nebo (4.11).

Sestavení matice v metodě `assemble` je téměř stejné jako u FEM. V cyklu přes elementy však navíc testujeme, zdali je tento element obohacený. Pokud ne, počítá se stejně jako v metodě FEM lokální matice s příspěvkem pouze od gradientů funkcí. V opačném případě se vytváří objekt třídy `AdaptiveIntegration`, která implementuje integraci na obohaceném elementu a generuje vypočtenou lokální matici.

Pokud elementem navíc prochází hranice vrtu, provede se adaptivní zjemnění referenčního elementu na množinu čtverců, popsané třídou `Square`. Mapování bodů mezi čtverci a referenčním elementem zařizuje třída `MyMaping`. Poté se volá metoda vlastního výpočtu integrálů, jejíž zdrojový kód najdeme v příloze C.1 a kterou si nyní popíšeme.

Nejprve probíhá zjišťování stupňů volnosti a inicializace proměnných. Výpočet, který jsme popsali detailně v kapitole 4.4, začíná cyklem přes všechny zjemněné čtverce (ř. 72). Řešíme zde zmíněný problém s mapováním gradientů (nemáme k dispozici matici zobrazení). Abychom využili knihovní funkce, konstruujeme uměle pro každý ze zjemněných čtverců dočasný objekt kvadratury s kvadraturními body příslušného čtverce. Pomocí něho zkonstruujeme dočasný objekt třídy `FEValues` (ř. 81), který v kvadraturních bodech vypočítá jak hodnoty bázových funkcí, tak jejich gradientů.

Následuje cyklus přes kvadraturní body a v každém z nich se vypočítá příspěvek do lokální matice. Na řádku 89 je výpočet jakobiánů a vah zobrazení bodů tak, jak jsme je určili v (4.16). Na řádcích 104-156 počítáme vektor hodnot bázových funkcí `shape_val_vec` a jejich gradientů `shape_grad_vec` – nejprve pro lineární funkce φ_j a poté pro obohacenou část (viz 4.8). Posledním prvkem vektoru je -1, kterou definujeme bázovou funkcí tlaku ve vrtu (mínus je zahrnuto ze záporných členů \mathbf{C}^a a \mathbf{C}^b , viz (4.10)).

Varianty, ve kterých vrty bereme buď jako okrajové podmínky, nebo jako plošné zdroje, vybíráme v kódu pomocí direktivní definice `BC_NEWTON`, nebo `SOURCES`. Vídíme například na ř. 97-99, kde počítáme hodnoty bázové funkce.

Dále plníme lokální matici příspěvky s gradienty na řádcích 161-168, viz prvky

matic $\mathbf{A}, \mathbf{R}, \mathbf{S}$ (4.10). Pak zbývá dopočítat členy z rovnic na vrtech. Pro variantu se zdroji (**SOURCES**) použijeme napočítaný vektor funkčních hodnot v aktuálním kvadraturním bodě a pokud tento bod leží ve vrtu, rovnou přičítáme členy do lokální matici. Pak končíme cyklus přes kvadraturní body a přes zjemněné čtverce a můžeme vrátit kompletní lokální matici.

Při variantě s vrty branými jako okrajové podmínky (**BC_NEWTON**) se musí hraniční integrály počítat zvlášť v iteraci přes body na hranici vrtu. Jakobián je roven délce kružnice – hranice vrtu, body jsou po ní rozmištěny ekvidistantně a jejich váhy odpovídají převrácené hodnotě celkového počtu bodů n_{points} . Součin jakobiánu a vah, počítaný na ř. 209, je tedy pro všechny body konstantní a je roven

$$JxW = \frac{2\pi r_w}{n_{points}} \quad (5.1)$$

Velice podobně jako pro kvadraturní body elementu se spočítají bázové funkce pro body na hranici vrtu a nakonec se přičte příspěvek hraničních integrálů do lokální matice. Tím jsme se dostali na konec popisované metody.

Abychom ověřili správné rozdělení elementu při adaptivní integraci, vytvořili jsme ve třídě **AdaptiveIntegration** navíc výstupní metodu **gnuplot_refinement**, která umí vygenerovat skript pro utilitu **GNUPLOT** a následně zobrazit adaptivní rozdělení elementu spolu s kvadraturními body, viz například obr. 4 v předchozím textu.

5.4 Model s jednou zvodní a jedním vrtem

Pro jednoduché modely s jednou zvodní, jedním vrtem a Dirichletovou okrajovou podmínkou na hranici oblasti jsme vytvořili třídy **XModel_simple** a **Model_simple**, potomky výše popsaných tříd modelů. Obsahují navíc konstruktor s jedním vrtem v argumentu a implementaci virtuální metody **assemble_dirichlet**. Vzhledem k tomu, že v úlohách neuvažujeme obohacení na hraničních uzlech oblasti, dovolili jsme si vyhnout se problému definice okrajové podmínky na obohaceném uzlu. Díky tomu je možno využít standardní metody knihovny pro distribuci okrajové podmínky jak pro model klasické FEM metody, tak i pro XFEM.

Vytvořili jsme pomocnou třídu **Comparing**, v níž jsme implementovali porovnávací metody **L2_norm_diff**. Knihovna Deal II má k dispozici funkci porovnávající spočítané řešení a analytickou funkci, což můžeme využít. Vstupem jsou objekt

DofHandler, vektor řešení na příslušné síti, analytická funkce, výstupní vektor difference, zvolená kvadratura a typ normy, ve které chceme rozdíl počítat.

Pro úlohy, ve kterých budeme srovnávat dvě různá diskrétní řešení, jsme vytvořili vlastní podobný výpočet. Potřebujeme dvě řešení spočítaná na shodné síti a tuto síť. Počítáme potom přes všechny elementy approximaci L^2 normy rozdílu různých řešení u a v , čili $\|u - v\|_{L^2(\Omega)} \sqrt{\int (u - v)^2}$.

6 Numerické testy

V této kapitole uvedeme výsledky výpočtů a porovnání vybraných modelových úloh, jejichž matematické formulace a způsoby implementace jsme doposud popsali. Abychom mohli posoudit vlivy parametrů a chování modelů, je potřeba volit úlohy nekomplikované, především na jednoduchých geometrií. Proto se omezíme na úlohy s jednou zvodní na obdélníkové, nebo kruhové oblasti.

Na úloze s jedním vrtem na kruhové zvodni ukážeme srovnání s analytickým řešením, ukážeme vliv velikosti obohacené oblasti v metodě XFEM a ověříme některé fyzikální předpoklady. V ostatních úlohách nebudeme mít k dispozici analytické řešení, proto zde jako referenční budeme uvažovat model vypočtený metodou FEM na velmi zjemněné síti.

Budeme srovnávat i čas výpočtu modelu. Do měření zahrneme ovšem pouze metody `setup`, `assemble` a `solve` bez tvorby sítě a výstupu řešení. Výsledky vystupujeme na jemných sítích (jiných než výpočetních), abychom mohli modely dobře porovnat. Pro model počítaný pomocí FEM můžeme distribuovat vektor řešení na výstupní síť pomocí optimalizovaných tříd knihovny a dokonce dokážeme velice snadno interpolovat řešení na jemnější síť (za předpokladu, že síť vzešly ze stejné hrubé sítě). Vzhledem k tomu, že pro metodu XFEM nemáme takové nástroje k dispozici a při výstupu řešení je navíc potřeba dopočítat a sloučit neobohacenou i obohacenou část, trvá výstup daleko déle. Nebylo však naším cílem tento netriviální technický problém řešit a optimalizovat.

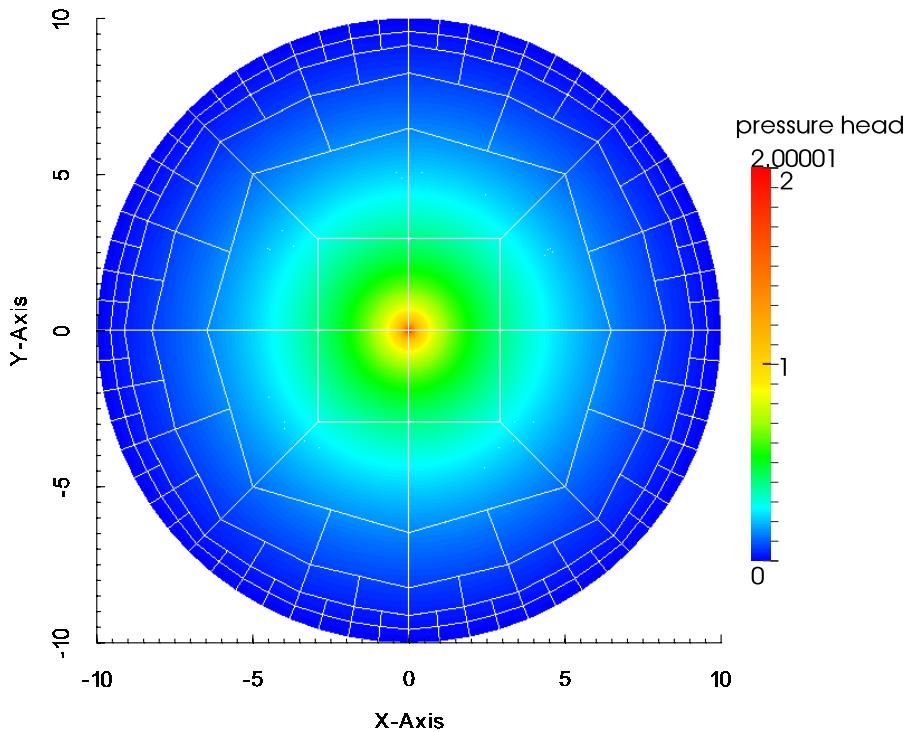
Některé výstupy uvádíme v obrazové formě v rámci přílohy, všechny zbylé nekomentované, nebo pouze zmíněné lze najít na přiloženém disku v adresáři `output`. Veškeré výsledky jsou reprodukovatelné zvolením požadovaných úloh ve zdrojovém kódu souboru `main.cc`, zkompilováním a spuštěním hlavního programu.

6.1 Úloha s jedním vrtem na kruhové oblasti

Pro úlohu na kruhové oblasti o poloměru R , s jediným vrtem v jejím středu, můžeme určit analytické řešení ve tvaru $u(r) = a \log(r) + b$, kde r je vzdálenost od středu vrstu. Konstanty a, b můžeme odvodit ze známých hodnot na okraji $u(R) = 0$ a v místě vrstu $u(r_w) = P_w$ a po úpravách dostaneme předpis funkce $u(r)$ ve tvaru

$$u(r) = a \log\left(\frac{r}{R}\right), \quad \text{kde } a = \frac{P_w}{\log \frac{r_w}{R}} \quad (6.1)$$

Potom jsme schopni porovnávat chybu vypočtených modelů proti analytickému řešení. Mějme tedy konkrétní úlohu na kruhové oblasti s poloměrem $R = 10$ m a vrtem ve středu $[0, 0]$, o poloměru $r_w = 0.02$ m a s tlakovou výškou 2.0 m. Zvolíme koeficient propustnosti $\sigma = 10^5 \text{ ms}^{-1}$ a transmisivitu zvodně $T = 10^{-4} \text{ m}^2\text{s}^{-1}$. Na okraji volíme nulovou Dirichletovu podmínku – připomínáme, že v modelu XFEM předpokládáme takovou síť, ve které všechny hraniční elementy jsou neobohacené, a tudíž okrajová podmínka je realizována pouze klasickými konečnými prvky na hranici.



obr. 9: Průběh tlaku v kruhové zvodni s jedním vrtem. Bílými čarami znázorněna výpočetní síť. Škála tlaku vyznačena barevně dle stupnice.

Technickým problémem v tomto případě je konstrukce kruhové sítě pomocí knihovny. Kružnice je approximována lineárně, tudíž pro různá zjemnění bychom mohli dostat sítě s různými hranicemi. Proto jsme nejprve vytvořili síť se zjemněnou pevnou hranicí, která byla výchozí pro všechny modely (viz obr. 9). Protože v implementaci XFEM neuvažujeme v síti volné uzly, zjemňujeme rovnoměrně pouze oblast obohacení, na neobohacené části mohou být volné uzly, které se pak řeší shodně jako u modelu FEM. Adaptivní zjemňování v modelu FEM funguje bez problému, jak jsme popsali v předchozích kapitolách.

V tabulce 1 jsou výsledky – chyba, počítaná jako L^2 norma rozdílu přesného řešení a approximovaného, počet stupňů volnosti a časová náročnost. XFEM úlohu

jsme spočítali v 5 iteracích, neboť jsme tím dosáhli menší chyby, než se poté podařilo pomocí FEM.

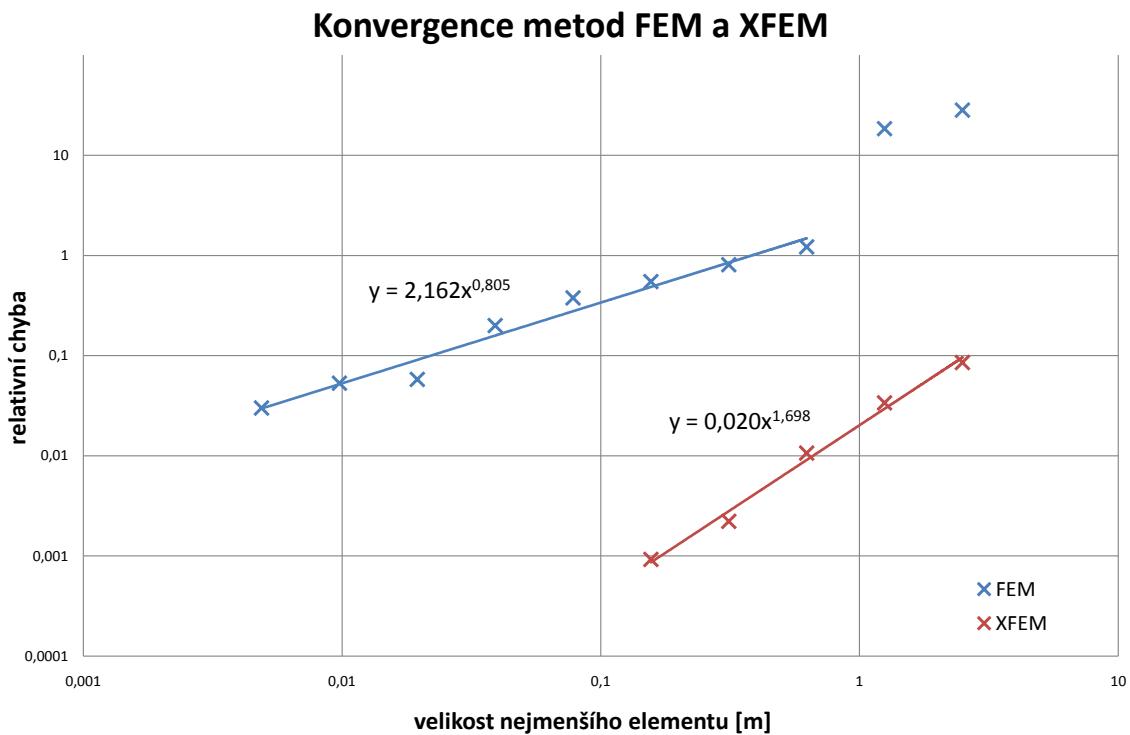
i	Chyba _{XFEM}	Chyba _{FEM}	n _{XFEM}	n _{FEM}	čas _{XFEM}	čas _{FEM}
0	3.44e-01	1.14e+01	258	249	0.740	0.300
1	1.36e-01	7.42e+00	690	421	1.480	0.600
2	4.29e-02	4.91e+00	1414	747	3.130	1.080
3	8.95e-03	3.26e+00	2818	1369	7.700	2.060
4	3.73e-03	2.21e+00	5366	2581	22.880	4.150
5	-	1.52e+00	-	4911	-	9.130
6	-	8.05e-01	-	9009	-	19.480
7	-	2.34e-01	-	16349	-	45.660
8	-	2.14e-01	-	24129	-	89.170
9	-	1.21e-01	-	36942	-	181.220
10	-	4.62e-02	-	61193	-	434.320

Tabulka 1: Porovnání XFEM a FEM metod s analytickým řešením ($n_{XFEM,FEM}$ označuje počet stupňů volnosti). Uvedené časy jsou v sekundách.

Je vidět, že při srovnatelném počtu stupňů volnosti je metoda XFEM výpočetně náročnější, jak ukazují časy výpočtu. Avšak pro dosažení stejné úrovně chyby potřebuje metoda FEM s lineárními prvky mnohonásobně více elementů v okolí vrtu, čímž neúměrně narůstá paměťová i časová náročnost.

Graf v obrázku 10 ukazuje konvergenci metod k přesnému řešení. Velikost nejmenšího elementu odpovídá elementu v blízkosti vrtu. Blížíme se výsledkům uvedeným v článku [GR10], kde uvádí řád konvergence XFEM metody $O(h^{1.8})$, který se blíží optimálnímu rádu konvergence lineárních konečných prvků $O(h^2)$. Upozorněme, že v článku řešili úlohu na čtvercové oblasti s pravidelnou sítí (tedy méně komplikovanou), a analytické řešení approximovali konečným součtem Besselových funkcí.

Řád konvergence FEM metody $O(h^{0.4})$, uváděný též v článku [GR10], není dle očekávání optimální. Dále uvádí, že konvergence FEM se vylepšuje, pokud je střed vrtu umístěn v uzlu sítě. To je v naší úloze splněno a v našem testu jsme dosáhli vyššího rádu konvergence $O(h^{0.8})$. Navíc jsme s použitím adaptivního zjemňování mohli elementy zmenšit až pod úroveň rozměru vrtu a ušetřili místo v paměti a výpočetní čas.

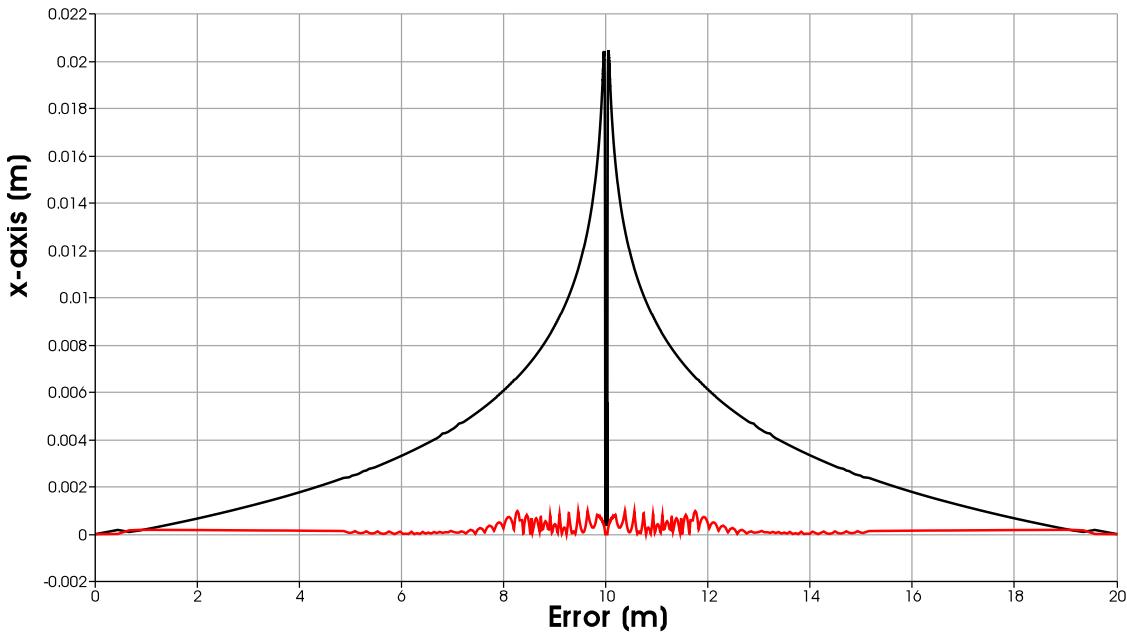


obr. 10: Graf konvergencie metod FEM (modrá) a XFEM (červená). Body jsme proložili mocninnou funkcií, pričemž první dvě najväčšie chyby řešení pomocí FEM jsme vyniechali. Môžeme odhadnúť rád konvergencie metody FEM $O(h^{0.8})$ a metody XFEM $O(h^{1.7})$.

Poznamenejme, že toto srovnání trochu znevýhodňuje metodu FEM, protože elementy sítě nedokážú dobre vystihnout hranici vrtu. Pokud bychom meli možnosť navrhnuť lepsí síť, ve ktoré by hranice elementov lepe zachycovaly hranici vrtu, pravdepodobne bychom dosahli rychlejsí konvergencie s mené hustými sítami. Zjednovaní sítie až na úroveň velikosti vrtu je ale presne ten problém, ktorému se chceme pri řešení reálnej úlohy vyhnout. Jedna z nejpodstatnejších výhod metody XFEM je absence obdobných požadavkov na vysítovanie oblasti. Pokud bychom řešili úlohu s desítkami vrtu a požadovali obdobnou prenos, nebude úloha metodou FEM reálne řešiteľná.

Nelze vyloučit, že nějakou část chyby nevnášíme do řešení chybou v implementaci. Môžeme však vyloučit niektorá podezrenia v miestach, kde môže mať rozhodujúci vliv presnosť numerických výpočtov. Počet diskretných bodov pre hranici vrtu je dostatečný – vyzkoušeno 500 a 1000 bez rozdielu v řešení, a i pri 50 bodech zůstávají chyby rádově shodné. Stojně tak počet úrovni zjemnenia pre adaptívnu integraci obohaceného elementu se jeví ako dostatečný.

Na obrázku 11 vidíme absolutní chybu approximovaných řešení v řezu osou x. FEM metoda s lineárními konečnými prvky se dopouští tím větší chyby, čím větší gradient se snaží postihnout. V metodě XFEM je mimo oblast obohacení chyba způsobená pouze approximací lineárními konečnými prvky, v oblasti obohacení je největší na rozmývacích elementech a směrem k hranici vrtu se mírně změňuje.



obr. 11: Průběh chyby řešení FEM (černá) a XFEM (červená) v řezu osou x. Obohacenou oblast XFEM metody vidíme v intervalu $< 8, 12 >$.

V kapitole 4.5 jsme popsali formulaci s vrty jako plošnými zdroji a zmínili jsme její výhody promítnuté do implementace. Zůstala nám k vyřešení otázka, zdali se jednodušší implementace (vzhledem k tomu, že odpadá výpočet křivkového integrálu) projeví i v časové úspoře při výpočtu. Mezi časy naměřenými na této testovací úloze jsme nepozorovali významný rozdíl, ten se projevil v určité míře až při úlohách s více vrty a na jemnějších sítích. Aproximace řešení byla stejně kvalitní jako u modelu s vrty branými jako okrajové podmínky a chyba se lišila oproti výsledkům v tabulce 1 pouze v rámci posledních desetinných míst.

6.2 Úloha s jedním vrttem na čtvercové oblasti

V této úloze ověříme předpokládané fyzikální chování při změně parametrů – koeficientu σ a transmisivity T . Čtvercovou oblast volíme proto, abychom měli jednodušší síť a rychlejší výpočet. Zároveň již nemáme v úmyslu porovnávat řešení

s analytickým, které by bylo pro čtvercovou oblast komplikovanější.

Mějme čtvercovou zvodeň s rozměry 20×20 m. Vrt umístíme do středu $[0, 0]$. Síť pro XFEM metodu je 5-krát globálně zjemněná, čili je rozdělena na 1024 elementů.

Koeficient propustnosti σ

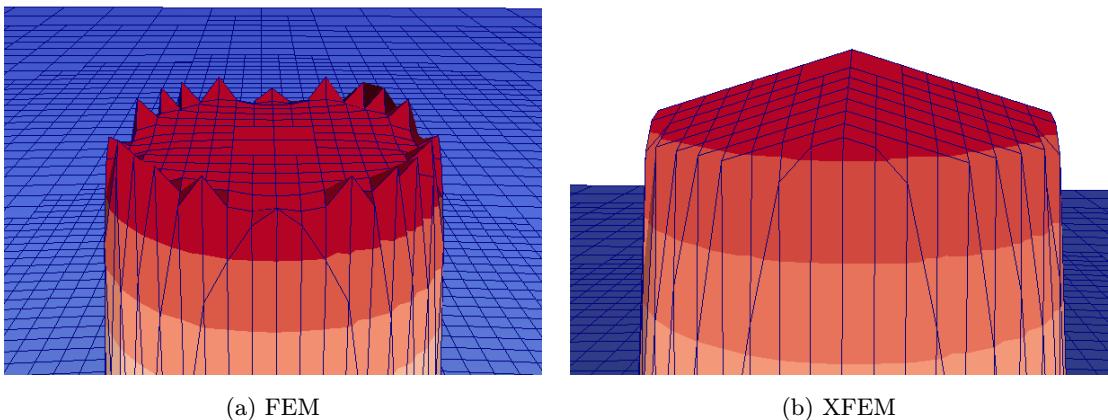
Při zvětšujícím se parametru σ bychom měli pozorovat bližší hodnoty tlaku na hranici vrtu a tlaku ve vrtu. Výsledky shrnuje tabulka 2, ve které uvádíme maximální tlak v oblasti (v této úloze s jedním vrtem odpovídá tlaku ve vrtu). Vidíme, že modely počítané metodami FEM a XFEM s vrty branými jako okrajové podmínky si odpovídají.

V tabulce jsme pro XFEM model s vrty jako zdroji použili $\eta = \sigma$. Protože tato konstanta má fyzikálně jiný význam, jsou i hodnoty odlišné (nemáme k dispozici jiný model pro porovnání).

$\sigma[\text{ms}^{-1}], \eta[\text{s}^{-1}]$	FEM	XFEM-Newton	XFEM-zdroje
1e-04	0.224	0.222	0.003
1e-02	1.859	1.857	0.223
1e+00	2.007	2.004	1.860
1e+02	2.012	2.000	2.004
1e+04	2.012	2.000	2.000
1e+06	2.001	2.000	2.000

Tabulka 2: Změna velikosti σ a maximální hodnota tlaku v okolí vrtu pro jednotlivé metody – FEM, XFEM s vrty jako okrajovými podmínkami, XFEM s vrty jako plošnými zdroji.

Hodnoty přesahující zadaný tlak ve vrtu v metodě FEM způsobuje špatná aproximace tlaku na okraji vrtu, kde lineární funkce musí vyrovnat skokovou změnu gradientu, jak dokumentuje obrázek 12a. U metody XFEM může nastat chyba, pokud vrt zasahuje do více elementů a zároveň lineární část není konstantní. Takovou chybu aproximace ukazuje obrázek 12b, ve kterém se uzel síť nachází uprostřed vrtu a lokální lineární část aproximace vytváří onu špičku. Nepozorovali jsme rozdíl mezi obohacením při variantě s vrty branými jako plošné zdroje.

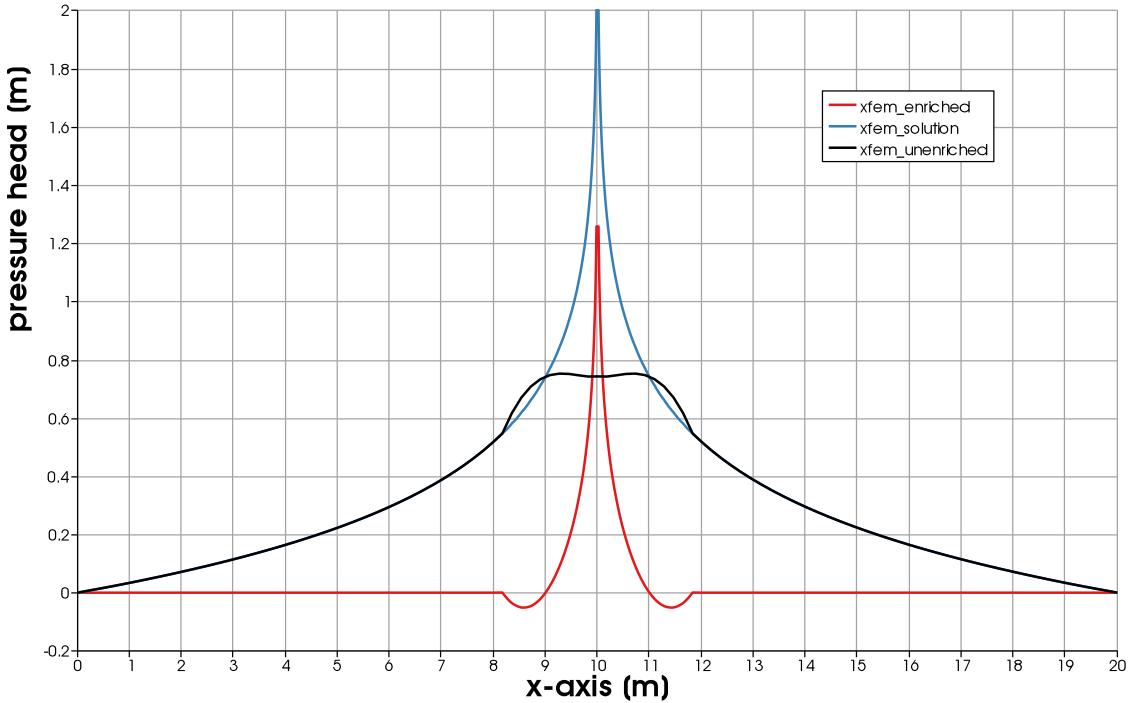


obr. 12: Chyby aproximace v okolí vrstu (pro $\sigma = 1.0$). V případě metody XFEM je problém shodný pro obě řešené varianty – vrty realizované jako okrajové podmínky, nebo plošné zdroje.

Co se týče hodnoty transmisivity, tak pokud nedefinujeme v oblasti žádné další zdroje f , můžeme rovnici (2.3) transmisivitou vydělit a model je na její volbě zcela nezávislý. To se nám potvrdilo v jednom z testů, ve kterém jsme měnili transmisivitu řádově od $10^{-12} \text{ m}^2\text{s}^{-1}$ do $10^2 \text{ m}^2\text{s}^{-1}$. Jinak se ovšem chová model řešený metodou XFEM, ve kterém uvažujeme vrty jako plošné zdroje. Potom vydělením rovnice (4.17) transmisivitou, dostáváme v rovnici poměr konstant η/T . Pokud je tento poměr dostatečně velký (cca $> 10^5$), je již silně vynucována rovnost tlaků ve vrstu h a H a model se chová stejně jako model XFEM se silnou Newtonovou okrajovou podmínkou. Transmisivita tedy zeslabuje vynucení rovnosti tlaku ve vrstu s okolním tlakem ve zvodni.

6.3 Poloměr obohacení

Vezměme shodnou úlohu, jako při testování konvergence v 6.1 a věnujme se otázce, jak volit velikost obohacené oblasti. Výpočty budeme provádět pouze s XFEM, ve které vrty uvažujeme jako okrajové podmínky, síť oblasti 6-krát globálně zjemníme. Na obrázku 13 demonstrujeme řešení v řezu rovinou vrstu. Vyznačili jsme zvlášť neobohacenou (černá), obohacenou (červená) část řešení a jejich součet (modrá). Poloměr oblasti obohacení jsme volili 2.0. Je patrné, že lineární prvky approximují řešení v oblasti s menším gradientem tlaku, na rozmyvacích elementech jsou postupně omezovány a chybějící část je kompenzována obohacujícími funkcemi.



obr. 13: Dekomponované řešení XFEM. Neobohacená část (černá), obohacená část (červená) a jejich součet (modrá).

Stejně tak jako při zjemňování sítě, roste s větším poloměrem obohacení počet přidaných stupňů volnosti a tedy i náročnost výpočtu. Očekáváme, že při zvětšování počtu obohacených elementů budeme zpřesňovat řešení. Volíme postupně poloměr obohacení od 0.2 do 2.0. Výsledky shrnuje tabulka 6, ve které vidíme vývoj relativní chyby řešení, vzhledem k volbě r_{enr} .

i	Poloměr obohacení r_{enr}	Rel. chyba
0	0.2	1.487e-01
1	0.5	2.233e-02
2	1.0	1.394e-03
3	1.5	9.045e-04
4	2.0	9.182e-04

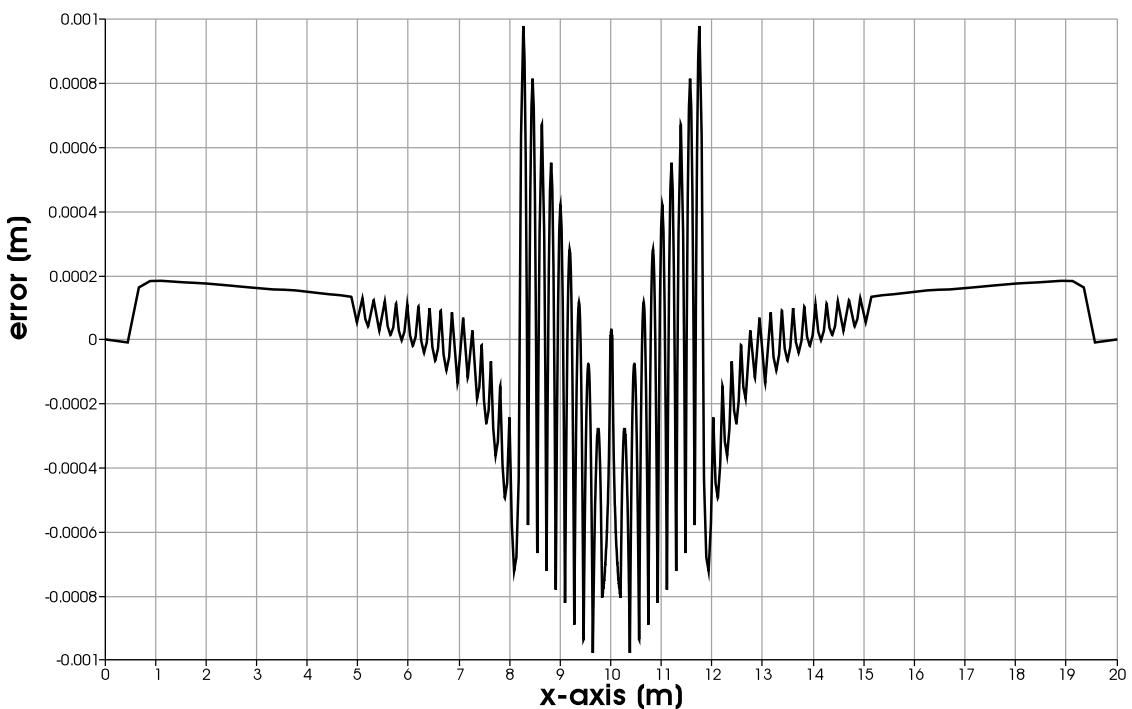
Tabulka 3: Změna velikosti obohacené oblasti a vývoj chyby.

Relativní chyba dle očekávání klesá, ovšem pouze do poloměru oblasti obohacení $r_{enr} = 1.5$. Důvod zřejmě souvisí s chybou, kterou jsme popsali a ukázali na obr. 14.

Pro další porovnání uvádíme v příloze B na obrázcích 19a-19d řešení na čtvercové

síti s poloměry obohacení 0.5, 1.5, 2.0, 3.0, 4.0. Relativní chyby řešení jsou v tomto případě vztaženy k řešení metodou FEM a jsou uvedeny v tabulce 6, taktéž v příloze.

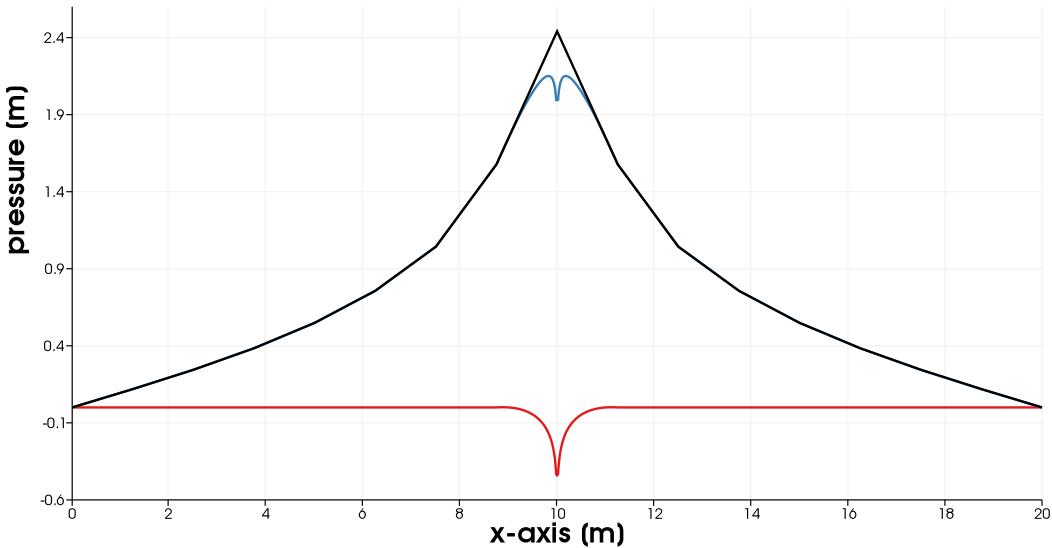
Dokumentujme ještě obrázkem 14 rozdíl přesného řešení a řešení získaného metodou XFEM. Vidíme, že chyba je největší (v absolutní hodnotě) v blízkém okolí vrtu a na přechodu z obohacené oblasti do neobohacené. Bohužel se nám zatím nepodařilo porozumět, proč chyba od hranice obohacené oblasti směrem k vrtu více neklesá a proč approximované řešení tímto způsobem kmitá okolo přesného řešení. Poznamenejme, že „vlnky“ v grafu (obr. 14) pokračují směrem k okraji oblasti, ovšem zde již nebyla výstupní síť natolik zjemněná, aby je dokázala viditelně postihnout.



obr. 14: Rozdíl přesného řešení a řešení XFEM.

Potřeba vhodného nastavení poloměru obohacení je zřejmá, ovšem není vůbec jednoduché navrhnut jeho automatickou volbu. Pro potřeby modelových úloh jsme hodnotu r_{enr} zatím omezili zdola nejprve poloměrem vrtu a poté největší vzdáleností mezi středem vrtu a vrcholem elementu, na kterém vrt leží. Tím dosáhneme toho, že alespoň jeden element je zcela obohacený. Dokumentuje to první volba poloměru obohacení v testu, která je příliš malá a poloměr je tedy přenastaven na 0.26

Upozorněme nyní na jeden problém, na který jsme během testu narazili. Pokud nebyla zavedena podmínka popsaná v minulém odstavci, dostali jsme zcela ne-správné deformované řešení, jak dokumentuje obrázek 15. Problémem je Newtonova



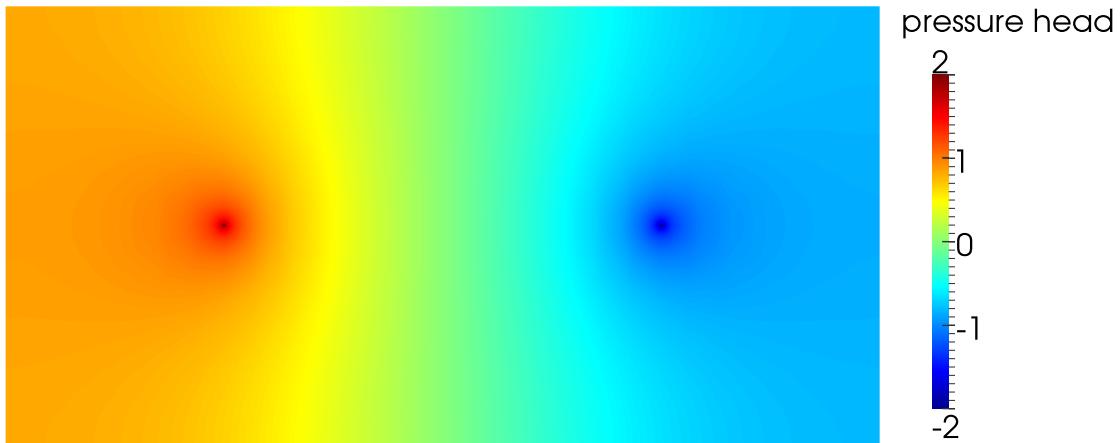
obr. 15: Dekomponované řešení XFEM, špatné obohacení – malý poloměr r_{enr}

okrajová podmínka, která při větším podílu přestupového koeficientu a transmisivity $\frac{\sigma}{T}$ (dle provedených pokusů řádově 10^9), vyžaduje tím striktnější shodu tlaku na hranici vrtu s tlakem ve vrtu. Abychom tohoto dosáhli, potřebujeme, aby lineární část řešení byla v místě vrtu v ideálním případě konstantní (dobře vidět na 13). Toto nelze zajistit na pouze částečně obohaceném elementu, neboť zde vynucení konstantnosti a rovnosti tlaku ve vrtu jdou proti sobě.

6.4 Úloha s dvěma vrty

Budeme počítat úlohu na obdélníkové oblasti s protilehlými vrcholy $[-10, -5], [10, 5]$, kde vrty s tlakovými výškami 2.0 m a -2.0 m umístíme do bodů $[-5, 0]$ a $[5, 0]$. Na okraji uvažujeme pouze homogenní Neumannovu okrajovou podmínku. Parametry jsme zvolili $T = 10^{-4} \text{ m}^2\text{s}^{-1}$ a $\sigma = 10^5 \text{ ms}^{-1}$. Nyní můžeme porovnávat pouze approximovaná řešení, proto vypočítáme úlohu metodou FEM na co nejjemnější síti a s tímto řešením budeme poté XFEM řešení porovnávat. Průběh řešení XFEM vidíme na obrázku 16.

V tabulce 4 jsou shrnutý výsledek výpočtu metodou XFEM pro tři různé úrovně zjemnění – 64, 256, 1024 elementů. Pro srovnání, výpočet modelu pomocí FEM na referenční zjemněné síti s 30440 elementy trval 125 s. Paměťová a časová náročnost je v případě metody XFEM výrazně menší. Chybou obou variant metody XFEM jsou téměř shodné, pro nejjemnější síť trvá výpočet o více než 1 sekundu méně.



obr. 16: Řešení modelu se dvěma vrty pomocí XFEM.

Zjemnění	Chyba XFEM Newton	Chyba XFEM Zdroje	Čas XFEM Newton	Čas XFEM Zdroje
3	7.5e-3	7.1e-3	2.22	2.5
4	5.0e-3	5.0e-3	4.51	4.7
5	4.1e-3	4.1e-3	11.64	10.2

Tabulka 4: Porovnání různých zjemnění sítě pro XFEM metodu. Relativní chyby obou variant metody XFEM jsou relativní k řešení metodou FEM.

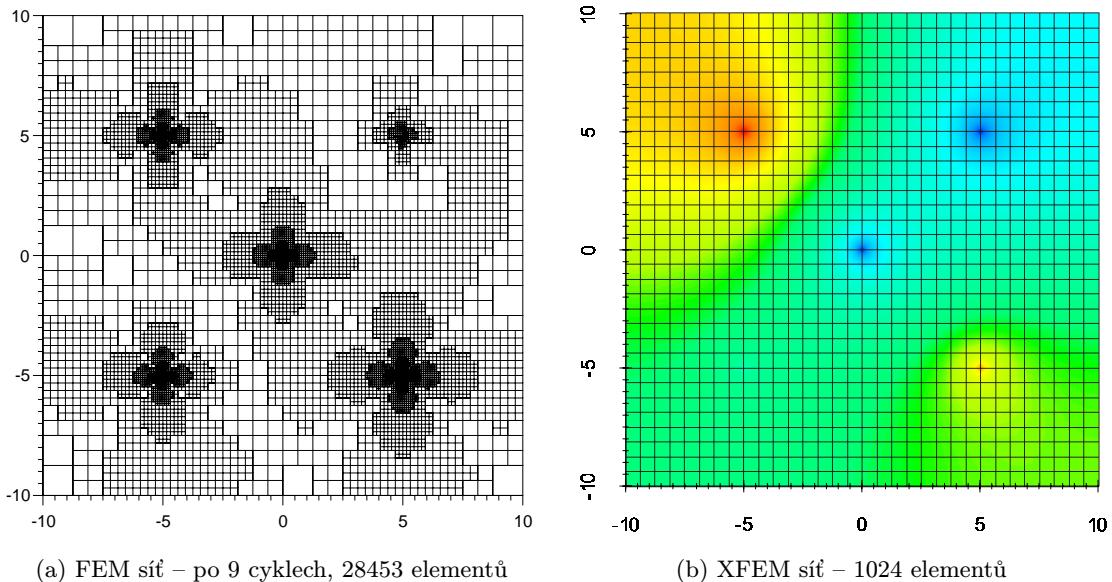
6.5 Úloha s více vrty

Na úloze s pěti vrty ukážeme především to, jak při klasické metodě konečných prvků zesložitujeme výpočetní síť adaptivním zjemňováním. Každý z vrtů vyžaduje malé elementy ve svém okolí, proto celkový počet elementů rychle narůstá (viz obr. 17a a 17b). Naproti tomu pro metodu XFEM jsme síť pouze 5-krát globálně zjemnili (dostaneme 1024 elementů).

	FEM	XFEM-Newton	XFEM-Zdroje
čas	171.95	17.02	16.30
rel. rozdíl	-	8.8e-3	8.8e-3

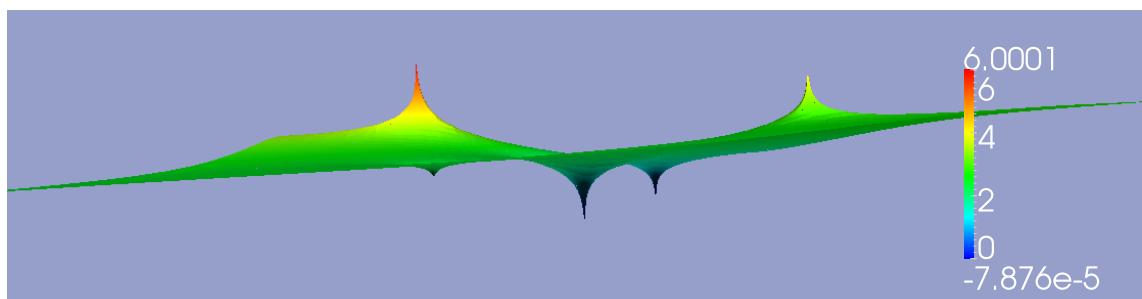
Tabulka 5: Porovnání metod FEM a XFEM na úloze s pěti vrty. XFEM počítáme oběma variantami – vrty jako Newtonova okrajová podmínka a vrty jako plošné zdroje.

Vrty jsou umístěny na souřadnicích $[-5, -5]$, $[0, 0]$, $[-5, 5]$, $[5, 5]$ a $[5, -5]$, s tlakovými výškami po řadě 0, 0, 4, 2 a 5 m. Transmisivitu jsme nastavili $T = 10^{-4} \text{ m}^2\text{s}^{-1}$ a koeficient propustnosti $\sigma = 10^5 \text{ ms}^{-1}$. Na obr. 18 vidíme vypočítané řešení pomocí XFEM metody ve formě 3D zobrazení tlakové výšky ve zvodni (rozsah hodnot – barev – odpovídá i obr. 17b).



obr. 17: Porovnání sítí metody FEM a XFEM (vrty jako okrajové podmínky).

Můžeme srovnat v tabulce 5 časy výpočtů jednotlivými metodami a relativní rozdíl řešení metody XFEM a metody FEM. Vidíme, že časy metody XFEM jsou přibližně 10-krát kratší. Výsledky obou variant metody XFEM jsou téměř totožné, pouze varianta s vrty branými jako plošné zdroje je o necelou sekundu rychlejší. Rozdíl mezi řešením metodou FEM a XFEM (oběma variantami) je pod 1%.



obr. 18: Řešení modelu s pěti vrty pomocí XFEM (vrty jako okrajové podmínky).

7 Závěr

Navázali jsme na článek [GR10] a nejprve popsali obdobný model pro ustálené saturované proudění podzemní vody ve zvodňovém systém s vrty.

V druhé části práce jsme rozebrali použité metody k výpočtu úlohy ustáleného saturovaného proudění podzemní vody v systému zvodní s vrty. Pro metodu konečných prvků jsme dokázali existenci a jednoznačnost řešení pomocí Laxovy-Milgramovy věty.

Implementovali jsme metodu lineárních konečných prvků v jazyce C++ pomocí knihovny Deal II. Využili jsme možností knihovny a vylepšili jsme výpočet o adaptivní zjemňování sítě na základě měření chyby gradientů na rozhraní elementů.

V další části jsme se seznámili s principem metody XFEM a její modifikací s funkcí rampy na rozmývacích elementech. Popsali jsme pomocí XFEM shodný model a navíc jsme navrhli variantu, ve které vrty nechápeme jako okrajové podmínky, ale jako plošné zdroje, a ukázali její výhody. Obě varianty metody se podařilo implementovat opět pomocí knihovny Deal II.

Metodě hraničních prvků (popsaná v příloze A), která je v principu odlišná od předchozích dvou metod, jsme mohli věnovat pouze omezené množství času. Pokusili jsme se o implementaci opět s pomocí knihovny Deal II, která obsahuje některé vhodné třídy i pro tuto metodu. Výpočet však dává špatné (nesmyslné) výsledky a chybu se zatím v kódu nepodařilo odhalit. Je možné, že bychom pomocí této metody na malých úlohách (např. numerický test 6.4), ve kterých ale nemáme k dispozici analytické řešení, dostali přesnější approximaci řešení než pomocí FEM. Tím bychom získali další referenční data pro porovnání s metodou XFEM, což bylo důvodem, proč jsme se metodou hraničních prvků zabývali.

Nakonec jsme vypočítali několik jednoduchých úloh, na kterých jsme mohli porovnávat metody mezi sebou. V úloze na kruhové oblasti, ve které jsme měli k dispozici analytické řešení, se ukazuje velmi dobrá approximace pomocí XFEM a rychlá konvergence. Naopak metoda s lineárními konečnými prvky konverguje suboptimálně, především proto, že ani na adaptivní síti nedokáže dobře zachytit hranici vrty.

Předpokladem pro rychlé řešení reálné úlohy je co nejjednodušší síť, nejlépe nezávislá na rozmístění zdrojů – vrty. V metodě FEM je kvalita approximace řešení úzce spojena s kvalitou sítě, tím máme na mysli především dostatečně dobré zachycení

hranice vrtu pro okrajovou podmítku, kterou na této hranici v modelu předepisujeme. Naopak při použití metody XFEM postačí rovnoměrně zjemněná a podstatně hrubší síť pro výpočet aproximace řešení při stejné nebo spíše menší chybě.

V úlohách s více vrty jsme porovnávali i časovou náročnost výpočtu. Nasazení metody XFEM se ukazuje v tomto směru velmi výhodné, neboť při použití menší a jednodušší sítě vzniká po diskretizaci také menší systém lineárních rovnic, který vyřešíme v kratším čase. Paměťová a časová náročnost metody XFEM je dána samozřejmě počtem elementů sítě, ale také volbou oblasti obohacení. Čím více elementů obohatíme, tím více přidáme stupňů volnosti. Volba v určitém smyslu optimálního poloměru obohacené oblasti v řešeném modelu je jedním z úkolů, kterými je možné dále se zabývat.

V kapitole 4.4 jsme popsali princip adaptivní integrace na obohaceném elementu, kterým prochází vrt. Způsob dělení elementu a rozdělovací kritérium jsme implementovali dle návrhu v [GR10], ale je možné je snadno nahradit. Bylo by zajímavé vyzkoušet i jiné přístupy a porovnat, jestli nejsme schopni jimi dosáhnout lepších výsledků. Jednou z možností je například měření rozdílu velikosti integrálu na jednotlivých čtvercích mezi dvěma úrovněmi zjemnění a na základě tohoto rozdílu poté volit další dělení.

Při implementaci jsme se setkali s problémem volných uzlů v síti. Řešení pro metodu FEM jsme popsali v kapitole 5.2.4, kde jsme zmínili i nástroje knihovny Deal II, které nám v tomto směru pomohly. V modelu řešeném metodou XFEM jsme předpokládali, že volné uzly mohou být síti pouze mimo obohacenou oblast. Řešení problematiky volných uzlů na obohacených elementech je popsáno například v článku [FBA⁺11], podle kterého by bylo zřejmě možné přizpůsobit implementaci i námi řešeného modelu.

Pro další postup bude nutné urychlit výstupní procedury. Zmínili jsme, že pro porovnávání modelů vystupujeme data na jemných sítích, vytvořených většinou adaptivním zjemňováním sítě v metodě FEM. Ve výstupní funkci XFEM metody bude zapotřebí optimalizovat mapování bodů mezi výstupní a výpočetní sítí a také vlastní výpočet řešení v těchto bodech.

V práci jsme se vypořádali s celou řadou teoretických i implementačních problémů. Použití principu XFEM a obohacení prostoru bázových funkcí ve smíšené formulaci vyžaduje ověření podmínek, za kterých je možno smíšenou formulaci pro-

vést a zkonstruovat potřebné prostory funkcí. To nakonec nebylo v rozsahu této práce možné a zvládnutelné, proto jsme tento bod zadání zanechali otevřený pro další práci.

Seznam použité literatury

- [BHK07] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II – a general purpose object oriented finite element library. *ACM Trans. Math. Softw.*, 33(4):24/1–24/27, 2007.
- [BM97] I. Babuška and J.M. Melenk. The partition of unity method. *International Journal for Numerical Methods in Engineering*, 40, 1997.
- [FB10] Thomas Peter Fries and Ted Belytschko. The generalized/extended finite element method: An overview of the method and its applications. *International Journal for Numerical Methods in Engineering*, 84:253–304, 2010.
- [FBA⁺11] T.P. Fries, A. Byfut, A. Alizada, K.W. Cheng, and A. Schröder. Hanging nodes xfem. *International Journal for Numerical Methods in Engineering*, 86:404–430, 2011.
- [Fri07] Thomas Peter Fries. A corrected xfem approximation without problems in blending elements. *International Journal for Numerical Methods in Engineering*, 75:503–532, 2007.
- [GR09] C. Geuzaine and J.-F. Remacle. GMSH: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009. <http://geuz.org/gmsh/>.
- [GR10] Robert Gracie and James R.Craig. Modelling well leakage in multilayer aquifer systems using the extended finite element method. *Finite elements in Analysis and Design*, Elsevier B.V., 46:504–513, 2010.
- [HP01] Professor Peter Hunter and Associate Professor Andrew Pullan. Fem/bem notes. *Finite elements in Analysis and Design*, Elsevier B.V., pages 43–59, 2001.
- [Rek99] Karel Rektorys. *Variační metody v inženýrských problémech a v problémech matematické fyziky*. Academia Praha, 1999.

- [RJJ⁺09] M. Rokyta, O. John, J. Málek, M. Pokorný, and J. Stará. Skripta – Úvod do moderní teorie parciálních diferenciálních rovnic. <http://ww.karlin.mff.cuni.cz/~malek/new/images/Skripta.pdf>, 2009. pages 15-22.

A Metoda hraničních prvků

V této příloze popíšeme teoretický základ pro metodu hraničních prvků (v angličtině BEM – Boundary element method), kterou se bohužel nepodařilo v rámci diplomové práce odladit do funkčního stavu. Tato metoda se liší od FEM použitím odlišné bázové/testovací funkce a odvozením integrální rovnice, kterou tvoří pouze hraniční integrály, čímž je snížena dimenze problému (z 3D na 2D síť, v našem případě z 2D na 1D síť). Pro diskretizaci jsou poté použity hraniční konečné prvky. Sestavení soustavy lineárních rovnic vede k nesymetrické neřídké matici, což je velká nevýhoda oproti FEM, na druhou stranu je však soustava podstatně menší. Řešit budeme touto metodou prozatím pouze Laplaceovu rovnici $\Delta u = 0$.

A.1 Odvození slabé formulace

V článku s poznámkami k metodám FEM a BEM [HP01] je odvozeno fundamentální řešení rovnice $\Delta u = 0$ (tzv. Greenova funkce), které použijeme jako testovací funkci, je zde odvozena integrální rovnice a popsána její diskretizace. Uvedeme zde stěžejní poznatky.

Mějme rovnici

$$\Delta u = 0 \quad \text{na } \Omega. \quad (\text{A.1})$$

Fundamentálním řešením této rovnice je řešení $\Delta\omega + \delta(\xi - x, \eta - y) = 0$ na oblasti \mathbb{R}^2 (neuvážujeme oblast Ω a původní okrajové podmínky). Převodem do polárních souřadnic, užitím vlastností Dirackovy funkce δ a vyřešením upravené rovnice získáme fundamentální řešení, tj. naši testovací funkci (A.2) a její směrovou derivaci (A.3), kterou budeme také potřebovat.

$$\omega = -\frac{1}{2\pi} \log r \quad (\text{A.2})$$

$$\frac{\partial\omega}{\partial n} = \frac{\partial\omega}{\partial r} \frac{dr}{dn} = -\frac{1}{2\pi} \frac{1}{r} \nabla r \cdot \mathbf{n}, \quad (\text{A.3})$$

V (A.2) a (A.3) je $r = \sqrt{(\xi - x)^2 + (\eta - y)^2}$ vzdálenost od bodu se singularitou. V článku [HP01] je provedeno podrobné odvození integrální rovnice (A.4) pomocí dvojté aplikace Greenovy věty na Laplaceovu rovnici (A.1) a dosazením (A.2) za testovací funkci.

$$c(P)u(P) + \int_{\Gamma} u \frac{\partial\omega}{\partial n} d\Gamma = \int_{\Gamma} \frac{\partial u}{\partial n} \omega d\Gamma, \quad (\text{A.4})$$

kde P je bodem hranice a konstanta $c(P)$ je dána jeho polohou takto

$$c(P) = \begin{cases} 1 & P \text{ je vnitřním bodem} \\ \frac{1}{2} & P \text{ leží na hladké hranici} \\ \frac{\alpha}{2\pi} & P \text{ leží na nehladké hranici, } \alpha \text{ je vnitřní úhel.} \end{cases}$$

A.2 Diskretizace

Hranici Γ oblasti Ω zdiskretizujeme na množinu elementů Γ_j a zavedeme approximaci konečnými prvky shodně s [HP01]

$$u_j = \sum_{k=1}^{N_j} \alpha_{j,k} \varphi_k, \quad q_j = \frac{\partial u_j}{\partial n} = \sum_{k=1}^{N_j} \beta_{j,k} \varphi_k, \quad (\text{A.5})$$

u_j, q_j jsou hodnoty u a q na j -tém elementu, $\alpha_{j,k}$ a $\beta_{j,k}$ jsou stupně volnosti na j -tém elementu a odpovídají hodnotám u a q v k -tému uzlu elementu ($P_{j,k}$). Dosazením (A.5) do (A.4) dostaneme

$$c(P)u(P) + \sum_{j=1}^N \sum_{k=1}^{N_j} \alpha_{j,k} \int_{\Gamma_j} \varphi_k \frac{\partial \omega}{\partial n} d\Gamma = \sum_{j=1}^N \sum_{k=1}^{N_j} \beta_{j,k} \int_{\Gamma_j} \varphi_k \omega d\Gamma. \quad (\text{A.6})$$

Rovnici (A.6) můžeme vyjádřit pro každý bod na hranici Γ . Pokud to provedeme pro všechny uzlové body hraniční sítě, dostaneme soustavu N rovnic ve tvaru

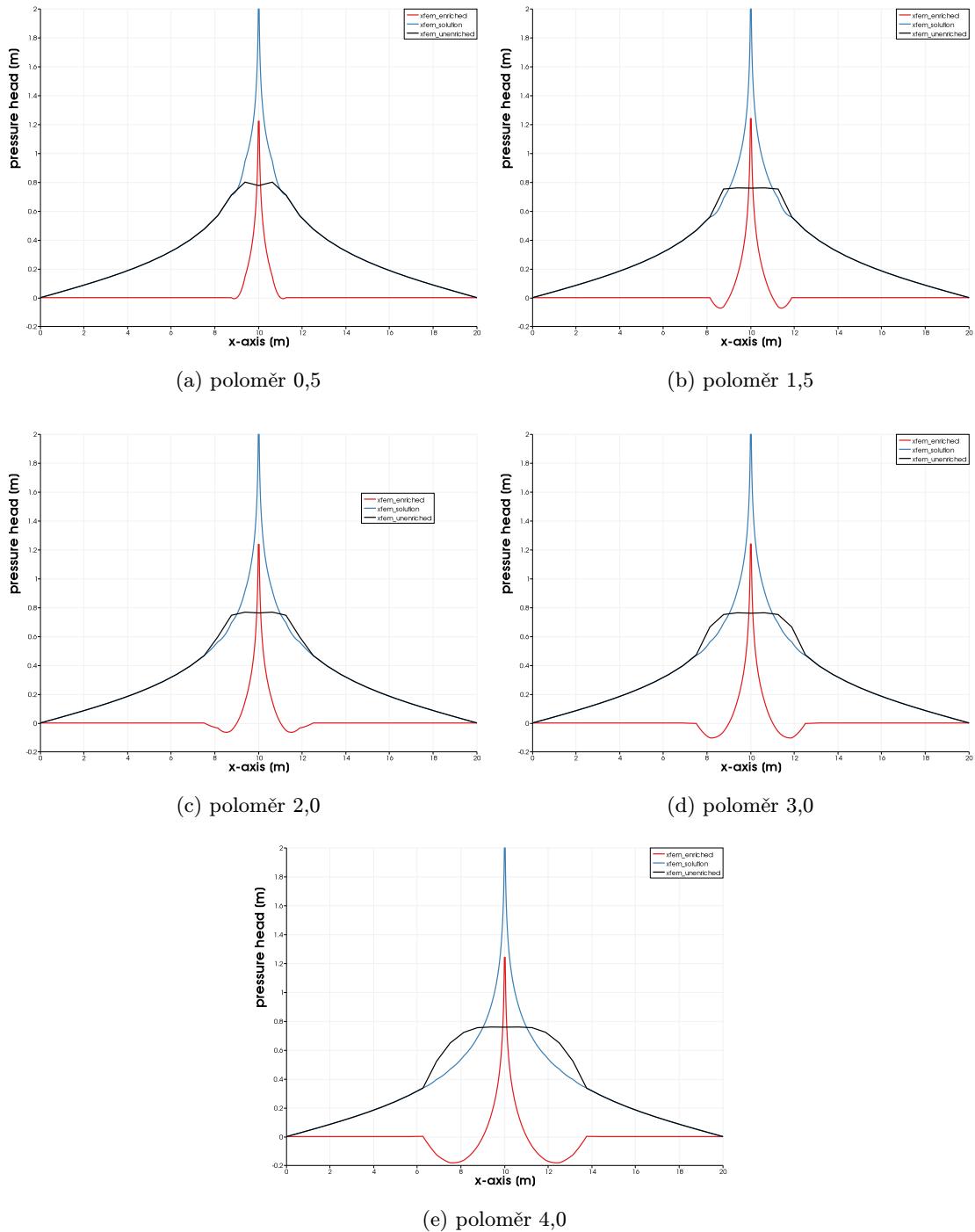
$$c_i u_i + \sum_{j=1}^N \sum_{k=1}^{N_j} \alpha_{j,k} \int_{\Gamma_j} \varphi_k \frac{\partial \omega_i}{\partial n} d\Gamma = \sum_{j=1}^N \sum_{k=1}^{N_j} \beta_{j,k} \int_{\Gamma_j} \varphi_k \omega_i d\Gamma. \quad (\text{A.7})$$

Rovnice (A.7) má zatím $2N$ neznámých. Abychom ji mohli vyřešit, musíme v každém bodě definovat buď hodnotu $u_{j,k}$ nebo $q_{j,k}$. Potom se v každém uzlovém bodě převeze člen se známými hodnotami na pravou stranu a dostaneme soustavu lineárních rovnic ve tvaru $\mathbf{Ax} = \mathbf{b}$, kterou musíme vyřešit. Připomeňme, že matice A je zaplněná a nesymetrická.

V naší úloze je tedy hledanou veličinou tlaková výška, proto položíme $u = h$. Poté na hranici Γ_N , kde máme definovanou homogenní Neumannovu okrajovou podmínku, položíme $q_{j,k} = 0$. Na hranici s Dirichletovou okrajovou podmínkou Γ_D položíme $h_{j,k} = h_D(P_{j,k})$. Na hranici vrtu ∂B_w dále položíme $h_{j,k} = H_w$, tedy rovnou hodnotě tlaku ve vrtu. Hodnotu h v jakémkoli bodě P oblasti Ω dostaneme ze vztahu

$$h(P) = \sum_{j=1}^N \sum_{k=1}^{N_j} \beta_{j,k} \int_{\Gamma_j} \varphi_k \omega(P) d\Gamma - \sum_{j=1}^N \sum_{k=1}^{N_j} \alpha_{j,k} \int_{\Gamma} \varphi_k \frac{\partial \omega(P)}{\partial n} d\Gamma. \quad (\text{A.8})$$

B Obrazová příloha



obr. 19: Úloha s jedním vrtem a různé poloměry obohacení. Na obrázcích 19a-19e je znázorněné rozložené řešení metody XFEM (modrá) na obohacenou část (červená) a neobohacenou část (černá).

i	Polomér obohacení r_{enr}	Chyba
0	0.5	0.009
1	1.5	0.018
2	2.0	0.016
3	3.0	0.017
4	4.0	0.018

Tabulka 6: Změna velikosti obohacené oblasti a vývoj chyby. Tabulka odpovídá grafickému výstupu na obr. 19.

C Zdrojový kód

V následující příloze nalezneme zdrojový kód některých metod programu, na které se odkazujeme v textu. Pro kompletní implementaci a popisy je nutné nahlédnout do zdrojových souborů.

C.1 function AdaptiveIntegration::integrate

```

1 void Adaptive_integration::integrate(
2     FullMatrix<double> &cell_matrix ,
3     Vector<double> &cell_rhs ,
4     std::vector<unsigned int> &local_dof_indices ,
5     const double &transmisivity )
6 {
7     double cell_jakobian = cell_mapping.jakobian();
8     cell_matrix = 0;
9     cell_rhs = 0;
10
11    // number of wells with q-points inside the cell
12    unsigned int n_wells_inside = 1;
13    //number of wells affecting the cell
14    unsigned int n_wells = xdata->n_wells();
15    unsigned int dofs_per_cell = fe->dofs_per_cell;
16
17    //local dof indices (both enriched and unenriched)
18    local_dof_indices.clear();
19
20    //getting unenriched local dofs indices
21    local_dof_indices.resize(dofs_per_cell);
22    cell->get_dof_indices(local_dof_indices);
23
24    //weights of enriched nodes = 1, else 0 (for every well)
25    std::vector<double> node_weights(n_wells * dofs_per_cell ,0);
26
27    //getting weights of nodes of the cell , enriched dof indices
28    for(unsigned int w = 0; w < n_wells; w++)
29    {
30        for(unsigned int i = 0; i < dofs_per_cell; i++)
31        {
32            if(xdata->global_enriched_dofs(w)[ i ] != 0)
33            {
34                node_weights[w*dofs_per_cell + i] = 1; //enriched
35                local_dof_indices.push_back(
36                    xdata->global_enriched_dofs(w)[ i ] );

```

```

37     }
38   }
39 }
40
41 unsigned int n_dofs = local_dof_indices.size();
42
43 for(unsigned int w = 0; w < n_wells; w++)
44 {
45   if(xdata->get_q_points(w).size() > 0)
46   {
47     n_wells_inside++;
48     //one more for well testing function
49     local_dof_indices.push_back(
50       xdata->get_well_dof_indices(w));
51   }
52
53 //initialized with zeros
54 cell_matrix = FullMatrix<double>(n_dofs+n_wells_inside ,
55                                     n_dofs+n_wells_inside);
56 //initialized with zeros
57 cell_rhs = Vector<double>(n_dofs+n_wells_inside);
58
59 //temporary for shape values and gradients
60 Tensor<1,2> xshape_grad;
61 double xshape, jacobian, jxw = 0;
62
63 //vector of quadrature points on the unit square
64 std::vector<Point<2>> q_points;
65 //vector of quadrature points mapped to unit cell
66 std::vector<Point<2>> q_points_mapped;
67
68 //temporary vectors for both shape and xshape values ,gradients
69 std::vector<Tensor<1,2>> shape_grad_vec(n_dofs);
70 std::vector<double> shape_val_vec(n_dofs+n_wells_inside ,0);
71
72 for(unsigned int s=0; s < squares.size(); s++)
73 {
74   //unit square quadrature points
75   q_points = squares[s].gauss->get_points();
76   q_points_mapped = squares[s].gauss->get_points();
77   //mapped from unit square to unit cell
78   squares[s].mapping.map_unit_to_real(q_points_mapped);
79
80   Quadrature<2> temp_quad(q_points_mapped);
81   FEValues<2> temp_fe_values(*fe,temp_quad,
82     update_values | update_gradients | update_jacobians );
83   temp_fe_values.reinit(cell);
84
85   jacobian = squares[s].mapping.jakobian();
86
87   for(unsigned int q=0; q < q_points.size(); q++)
88   {
89     jxw = jacobian * temp_fe_values.jacobian(q).determinant() *
90           squares[s].gauss->get_weights()[q];
91
92     // filling shape values and shape gradients at first
93     for(unsigned int i = 0; i < dofs_per_cell; i++)
94     {
95       shape_grad_vec[i] = temp_fe_values.shape_grad(i,q);
96
97 #ifdef SOURCES
98   shape_val_vec[i] = temp_fe_values.shape_value(i,q);
99 #endif
100  }
101
102 // filling xshape values and xshape gradients next
103 //index in the vector of values and gradients
104 unsigned int index = dofs_per_cell;
105 for(unsigned int w = 0; w < n_wells; w++) //W
106 {
107   Well * well = xdata->get_well(w);
108   //gradient of xfem function needn't be mapped

```

```

109         xshape = well->xshape_value(
110             mapping->transform_unit_to_real_cell(
111                 cell , q-points_mapped[q]));
112         xshape_grad = well->xshape_grad(
113             mapping->transform_unit_to_real_cell(
114                 cell , q-points_mapped[q]));
115
116     for(unsigned int k = 0; k < dofs_per_cell; k++) //Ne
117     {
118         if(xdata->global_enriched_dofs(w)[k] != 0)
119         {
120             #ifdef SOURCES
121                 shape_val_vec[index] = 0;
122             #endif
123                 shape_grad_vec[index] = 0;
124                 for(unsigned int l = 0; l < dofs_per_cell; l++) //N
125                 {
126                     #ifdef SOURCES
127                         shape_val_vec[index] +=
128                             node_weights[w*dofs_per_cell + l] *
129                             temp_fe_values.shape_value(l,q) *
130                             temp_fe_values.shape_value(k,q) *
131                             xshape;
132                     #endif
133
134                     //gradients of shape functions need to be mapped
135                     //scale_to_unit means inverse scaling
136                     shape_grad_vec[index] +=
137                         node_weights[w*dofs_per_cell + l] *
138                         ( temp_fe_values.shape_grad(l,q) *
139                           temp_fe_values.shape_value(k,q) *
140                           xshape
141                           +
142                           temp_fe_values.shape_value(l,q) *
143                           temp_fe_values.shape_grad(k,q) *
144                           xshape
145                           +
146                           temp_fe_values.shape_value(l,q) *
147                           temp_fe_values.shape_value(k,q) *
148                           xshape_grad
149                         );
150                     } //for l
151                     index++;
152                 } //if
153             } //for k
154             //testing function of the well (minus from equation)
155             shape_val_vec[index] = -1.0;
156         } //for w
157
158         //filling cell matrix now
159         //additions to matrix A,R,S
160
161     for(unsigned int i = 0; i < n_dofs; i++)
162         for(unsigned int j = 0; j < n_dofs; j++)
163         {
164             cell_matrix(i,j) += transmisivity *
165                             shape_grad_vec[i] *
166                             shape_grad_vec[j] *
167                             jxw;
168         }
169
170     #ifdef SOURCES
171         for(unsigned int w = 0; w < n_wells; w++)
172         {
173             //this condition tests if the quadrature point lies
174             //within the well (testing function omega)
175             if(xdata->get_well(w)->points_inside(
176                 mapping->transform_unit_to_real_cell(
177                     cell , q-points_mapped[q])))
178             {
179                 for(unsigned int i = 0; i < n_dofs+n_wells_inside; i++)
180                 {

```

```

181         for(unsigned int j = 0; j < n_dofs+n_wells_inside; j++)
182     {
183         cell_matrix(i,j) +=
184             xdata->get_well(w)->get_perm2aquifer() *
185                 shape_val_vec[i] *
186                 shape_val_vec[j] *
187                 jxw;
188     } //for j
189 } //for i
190 } //if w
191 } //for w
192 #endif
193
194 } //for q
195 } //for s
196
197
198 #ifdef BCNEWTON
199 unsigned int n_w_dofs=0;
200
201 for(unsigned int w = 0; w < n_wells; w++)
202 {
203     if(xdata->get_q_points(w).size() > 0)
204     {
205         Well * well = xdata->get_well(w);
206         // jacobian = radius of the well;
207         // weights are the same all around
208         jxw = 2 * MPI * well->get_radius() /
209             well->get_q_points().size();
210
211         // value of enriching function is constant
212         // all around the well edge
213         xshape = well->xshape_value(well->get_q_points()[0]);
214
215         shape_val_vec.clear();
216         node_weights.clear();
217         node_weights.resize(dofs_per_cell,0);
218
219         unsigned int n_enriched_dofs=0;
220         for(unsigned int i = 0; i < dofs_per_cell; i++)
221         {
222             if(xdata->global_enriched_dofs(w)[i] != 0)
223             {
224                 node_weights[i] = 1; //enriched
225                 n_enriched_dofs++;
226             }
227         }
228
229         n_w_dofs = dofs_per_cell+n_enriched_dofs+1;
230         //unenriched, enriched, well
231         shape_val_vec.resize(n_w_dofs,0);
232
233         //cycle over quadrature points inside the cell
234         for (unsigned int q=0; q<xdata->get_q_points(w).size(); ++q)
235         {
236             Point<2> q_point = *(xdata->get_q_points(w)[q]);
237             //transforming the quadrature point to unit cell
238             Point<2> unit_point =
239                 mapping->transform_real_to_unit_cell(cell, q_point);
240
241             // filling shape values at first
242             for(unsigned int i = 0; i < dofs_per_cell; i++)
243                 shape_val_vec[i] = fe->shape_value(i, unit_point);
244
245             // filling xshape values next
246             //index in the vector of values and gradients
247             unsigned int index = dofs_per_cell;
248
249             for(unsigned int k = 0; k < dofs_per_cell; k++) //Ne
250             {
251                 if(xdata->global_enriched_dofs(w)[k] != 0)
252

```

```

253     {
254         shape_val_vec[index] = 0; // giving zero
255         for(unsigned int l = 0; l < dofs_per_cell; l++) //N
256     {
257         shape_val_vec[index] +=
258             node_weights[l] *
259                 fe->shape_value(l, unit_point) *
260                 fe->shape_value(k, unit_point) *
261                 xshape;
262         } //for l
263         index++;
264     } //if
265 } //for k
266
267 //testing function of the well
268 shape_val_vec[index] = -1.0;
269
270 for (unsigned int i=0; i < n_w_dofs; ++i)
271     for (unsigned int j=0; j < n_w_dofs; ++j)
272     {
273         cell_matrix(i, j) += (well->get_perm2aquifer() *
274                             shape_val_vec[i] *
275                             shape_val_vec[j] *
276                             jxw );
277     } //end of iteration over q-points
278 } //if
279 } //for w
280 #endif
281 }

```
