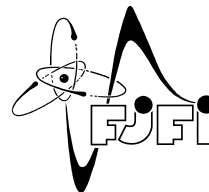




ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta jaderná a fyzikálně inženýrská



Numerické řešení dvoufázového nemísivého filtračního proudění s kapilaritou a jeho implementace na GPU

Numerical solution of two-phase immiscible flow in porous media with capillarity and its implementation on GPU

Výzkumný úkol

Autor: **Bc. Jakub Klinkovský**
Vedoucí práce: **Ing. Radek Fučík, Ph.D.**
Akademický rok: 2015/2016

- Zadání práce -

Poděkování:

Děkuji svému školiteli Ing. Radku Fučíkovi, Ph.D. za pečlivost, ochotu a vstřícnost při vedení mé výzkumné práce. Dále děkuji Ing. Tomáši Oberhuberovi, Ph.D. za vývoj knihovny TNL a neocenitelné rady při implementaci paralelního řešiče.

Čestné prohlášení:

Prohlašuji, že jsem tuto práci vypracoval samostatně a uvedl jsem všechnu použitou literaturu.

V Praze dne 2. června 2016

Jakub Klinkovský

<i>Název práce:</i>	Numerické řešení dvoufázového nemísivého filtračního proudění s kapilaritou a jeho implementace na GPU
<i>Autor:</i>	Bc. Jakub Klinkovský
<i>Obor:</i>	Matematické inženýrství
<i>Druh práce:</i>	Výzkumný úkol
<i>Vedoucí práce:</i>	Ing. Radek Fučík, Ph.D., Katedra matematiky, Fakulta jaderná a fyzikálně inženýrská, České vysoké učení technické v Praze
<i>Abstrakt:</i>	<p>Tato práce se zabývá numerickým řešením dvoufázového nemísivého proudění v porézním prostředí a implementací masivně paralelního řešiče využívajícího moderní architektury GPU. Pro numerické řešení zformulované úlohy sestavíme semi-implicitní numerické schéma založené na hybridní metodě smíšených konečných prvků a metodě konečných objemů. Pro zlepšení numerických vlastností schématu je využita upwindová stabilizace a metoda mass-lumping. Schéma je implementováno pro paralelní architektury GPU s využitím platformy CUDA a knihovny TNL. Experimentální analýzou řádu konvergence pro testovací úlohy se známým semi-analytickým řešením ověříme správnost implementace a konvergenci numerického schématu. Chování různých modelů kapilarity a variant numerického schématu při řešení advekčně-difúzních úloh v heterogenním prostředí ověříme pomocí úlohy s referenčním řešením publikovaným v literatuře. Pro zvolenou testovací úlohu porovnáme efektivitu paralelního výpočtu na GPU a provedeme podrobnou analýzu efektivity paralelizovaného algoritmu.</p>
<i>Klíčová slova:</i>	Dvoufázové proudění, hybridní metoda smíšených konečných prvků, mass-lumping, upwind, paralelizace, GPU

Obsah

Úvod	6
1 Dvoufázové proudění v porézním prostředí	8
1.1 Porézní prostředí	8
1.2 Fáze	9
1.3 Saturace	9
1.3.1 Reziduální saturace	10
1.4 Zákon zachování hmoty	10
1.5 Darcyho zákon	11
1.6 Kapilární tlak	11
1.7 Relativní permeabilita	12
1.8 Rovnice dvoufázového proudění	13
2 Obecná formulace	14
3 Numerické schéma	16
3.1 Variační formulace	16
3.1.1 Prostor \mathbf{RT}_0	16
3.1.2 Aproximace rychlostí	18
3.1.3 Aproximace skalárních veličin	19
3.2 Sestavení globální soustavy rovnic	21
3.2.1 Diskretizace okrajových podmínek	21
3.2.2 Redukce soustavy	21
3.3 Algoritmus výpočtu	22
3.4 Výpočet prvků lokálních matic	23
3.4.1 Výpočet matic $\mathbf{B}_{i,j,K}$ v \mathbb{R}^1	23
3.4.2 Výpočet matic $\mathbf{B}_{i,j,K}$ v \mathbb{R}^2	24
3.4.3 Projekce vektorů $\mathbf{w}_{i,K,E}$ do prostoru $\mathbf{RT}_0(K)$	24
4 Paralelní implementace na GPU	26
4.1 Paralelizace numerického schématu	26
4.1.1 Lokální inverze	26
4.1.2 Sestavení globální matice	27
4.1.3 Řešení soustav lineárních algebraických rovnic s řádkou maticí	27
4.2 Detaily implementace	29
5 Numerické výsledky	30
5.1 McWhorterova–Sunadova úloha	30
5.2 Difúzní úloha v heterogenním prostředí	31
5.3 Numerická analýza difúzních úloh	33
5.3.1 Experimentální analýza konvergence	33

5.3.2	Efektivita paralelního výpočtu	37
5.4	Bariérový efekt	41
5.5	Obtékání překážky	45
5.6	Shrnutí numerických výsledků	48
	Závěr	49
A	Parametry porézních materiálů	50
	Literatura	52

Úvod

Problematika studia procesů v porézním prostředí je velmi bohatá a modelování proudění je jedním ze základních prostředků analýzy a predikce těchto procesů. Matematické modely slouží jako podklad v řadě praktických aplikací, od průmyslových aplikací jako těžba ropy, ukládání CO_2 nebo radioaktivních odpadů, přes ekologické aplikace jako např. ochrana zdrojů pitné vody a simulace šíření kontaminace v podzemí, až po armádní aplikace jako např. detekce min. V této práci se budeme zabývat numerickým řešením dvoufázového nemísivého proudění, což je zjednodušený matematický model, který však poskytuje množství úloh zajímavých z teoretického pohledu i z hlediska numerického řešení. Pro studium proudění v porézním prostředí existuje mnoho numerických schémat založených na metodách konečných diferencí, konečných objemů nebo konečných prvků, přičemž každá z metod má své výhody a nevýhody. V této práci popíšeme pokročilé numerické schéma založené na hybridní metodě smíšených konečných prvků a semi-implicitním přístupu k časové diskretizaci. Použití smíšené hybridní metody smíšených konečných prvků má oproti klasickým metodám řadu výhod: umožňuje použití obecných nestrukturovaných sítí, vede na soustavu s pozitivně definitní maticí a lze ji dobře paralelizovat. Linearizace v čase pomocí metody zamrzlých koeficientů vedoucí na semi-implicitní schéma má oproti plně implicitním metodám výhody v lepší efektivitě.

Numerické simulace komplexních úloh z oblasti modelování proudění vyžadují obrovský výpočetní výkon moderních počítačů. Běžným způsobem, jak dosáhnout požadovaného výsledku v kratším čase, je paralelizace použitých algoritmů a rozptřeni zátěže mezi více výpočetních jednotek. Tradiční architektury využívající vícejádrová CPU jsou omezena relativně malým počtem výpočetních jednotek a zejména datovou propustností operační paměti. V posledních letech se rapidně rozvíjí oblast obecných výpočtů na GPU, což jsou pomocné výpočetní jednotky určené primárně pro výpočty spojené s počítačovou grafikou, které si získaly oblibu díky mnohonásobně vyššímu výkonu oproti běžným procesorům. Dnešní GPU disponují typicky tisíci výpočetními jednotkami, které jsou oproti moderním sofistikovaným CPU podstatně primitivnější, ale jako celek poskytují mnohonásobně vyšší výpočetní výkon. Aby bylo možné efektivně využít potenciál této masivně paralelní architektury, jsou GPU vybavena speciální pamětí, která je řádově rychlejší oproti běžné operační paměti. Z programátorského hlediska je vývoj efektivního kódu pro GPU netriviální zejména z hlediska dobrého rozptřeni řešené úlohy mezi všechny výpočetní jednotky a efektivního využití hierarchie pamětí.

Cílem této práce je implementace masivně paralelního řešiče pro numerické řešení dvoufázového nemísivého proudění v porézním prostředí pomocí hybridní metody smíšených konečných prvků. Výsledný řešič bude určen primárně pro paralelní architekturu GPU a pro účely srovnání také pro architekturu dnešních CPU. K implementaci kódu v jazyce C++ využijeme platformu CUDA a knihovnu TNL. Správnost implementace a konvergenci numerického schématu ověříme pomocí testovacích úloh se známým přesným řešením nebo zjednodušených úloh s referenčním řešením publikovaným v literatuře. Dále porovnáme efektivitu paralelního řešiče vzhledem k jeho základní sériové verzi a provedeme podrobnou analýzu paralelizovaného algoritmu.

Struktura práce

V první kapitole popíšeme danou problematiku ve fyzikálním kontextu, zavedeme základní pojmy nezbytné pro popis porézního prostředí a vícefázového proudění ovlivněného kapilaritou a zformulujeme soustavu parciálních diferenciálních rovnic popisujících dvoufázové proudění v porézním prostředí.

Druhá kapitola představuje mezistupeň mezi konkrétní formulací úlohy dvoufázového proudění v porézním prostředí a popisem numerického schématu. Je zde popsána soustava parciálních diferenciálních rovnic s obecnými koeficienty doplněná o počáteční a okrajové podmínky a konkrétní volba nelineárních koeficientů pro danou úlohu.

Třetí kapitola se věnuje popisu numerického schématu založeného na kombinaci hybridní metody smíšených konečných prvků a metody konečných objemů pro prostorovou diskretizaci, Eulerovy metody pro časovou diskretizaci a semi-implicitním přístupem metody zamrzlých koeficientů pro linearizaci diskrétního schématu v čase. V poslední části kapitoly je popsán algoritmus výpočtu numerického řešení a použití metody mass-lumping pro zlepšení numerických vlastností schématu.

Ve čtvrté kapitole je popsána paralelizace numerického schématu, včetně paralelizace algoritmu pro řešení soustav lineárních algebraických rovnic, a implementace řešiče pro paralelní architekturu GPU s využitím platformy CUDA a knihovny TNL.

V poslední kapitole jsou prezentovány výsledky numerických simulací. Pro ověření správnosti implementace a konvergence numerického schématu jsou využity testovací úlohy se známým semi-analytickým řešením. Experimentální analýza řádu konvergence je provedena pro úlohy v 1D a ve 2D. Dále jsou popsány dvě advekčně-difúzní úlohy simulující bariérový efekt a na zvolené testovací úloze je provedena podrobná analýza efektivity výpočtu.

Kapitola 1

Dvoufázové proudění v porézním prostředí

V první kapitole popíšeme fyzikální kontext a matematický model problematiky dvoufázového nemísivého proudění v porézním prostředí. Postupně zavedeme základní pojmy potřebné pro popis porézního prostředí a jeho vlastností, veličiny potřebné pro popis vícefázového proudění a uvedeme rovnice popisující toto proudění. V závěru kapitoly zformulujeme soustavu parciálních diferenciálních rovnic, kterou se budeme zabývat v dalších kapitolách.

Definice pojmů jsou převzaty z [1] a [9], kde lze najít podrobnější popis této problematiky.

1.1 Porézní prostředí

Porézním prostředím rozumíme takové prostředí, které je tvořené primárně pevnými látkami (*skeletem*), ale které obsahuje prázdné dutiny, neboli *póry*. Typickými příklady porézního prostředí jsou zemina složená ze šterku, písku, jílu a organické hmoty, molitan, nebo rostlinná a živočišná tkáň. Póry mohou být vyplněny jednou nebo více tekutinami (např. vodou, vzduchem), přičemž předpokládáme následující vlastnosti [1]:

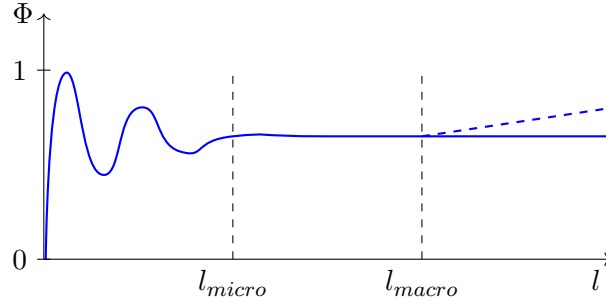
- póry tvoří propojenou síť,
- póry jsou dostatečně velké ve srovnání s rozměry molekul tekutin vyskytujících se v systému,
- póry jsou dostatečně malé na to, aby proudění tekutiny bylo řízeno adhezními silami na rozhraní tekutiny a skeletu.

Při splnění těchto předpokladů lze uvažovat matematický model proudění tekutin v porézním prostředí popsany v této práci.

Na úlohy proudění v porézním prostředí můžeme nahlížet z různých měřítek. V *makroskopickém měřítku* pozorujeme jen některé vlastnosti porézního prostředí, např. rozdílnou průměrnou hrubost zrn písku, ale nejsme schopni rozlišit vnitřní strukturu materiálu. V *mikroskopickém měřítku* jsme schopni rozeznat jednotlivé póry, přičemž tekutiny stále považujeme za kontinuum. Teprve v *molekulárním měřítku* již můžeme pozorovat jednotlivé molekuly tekutin vyplňujících póry. Proudění tekutin lze popsat v mikroskopickém měřítku pomocí soustavy Navierových–Stokesových rovnic s příslušnými okrajovými podmínkami. Porézní prostředí má ale obvykle složité geometrické uspořádání, které není známo na mikroskopické úrovni, navíc okrajových podmínek by bylo příliš mnoho na to, aby bylo možné provést numerickou simulaci.

Pro popis procesů v porézním prostředí z makroskopického hlediska je nutné zavést makroskopické veličiny, které popisují mikrostrukturu prostředí. Jednou z těchto veličin je *porozita*, která vyjadřuje podíl objemu pórů v části uvažované oblasti vzhledem k celkovému objemu této části. Uvažujme oblast $\Omega \subset \mathbb{R}^d$, kde $d = 1, 2$ nebo 3 , vyplněnou porézním prostředím a definujme funkci

$$\gamma(\mathbf{x}) = \begin{cases} 1 & \text{pokud bod } \mathbf{x} \text{ leží v prázdném prostoru,} \\ 0 & \text{pokud bod } \mathbf{x} \text{ leží v prostoru vyplněném skeletem,} \end{cases} \quad \forall \mathbf{x} \in \Omega, \quad (1.1)$$



Obrázek 1.1: Závislost porozity Φ na průměru l referenční oblasti Ω_0 .

příčemž rozhodování, zda bod x leží v prázdném prostoru nebo v prostoru vyplněném skeletem, provádíme na mikroskopické úrovni. Průměrováním funkce γ dospějeme k makroskopické veličině Φ zvané *porozita*. Uvažujme tedy podoblast $\Omega_0 \subset \Omega$ obsahující bod x_0 . Pak veličina Φ je dána vztahem

$$\Phi(x_0) = \frac{1}{\int_{\Omega_0} 1 d\mathbf{x}} \int_{\Omega_0} \gamma(\mathbf{x}) d\mathbf{x}, \quad \forall x_0 \in \Omega. \quad (1.2)$$

Hodnota porozity Φ v daném bodě x_0 závisí na velikosti oblasti Ω_0 , přes kterou průměrujeme. Na obr. 1.1 je zobrazena závislost porozity Φ na průměru l oblasti Ω_0 . Pro velmi malé hodnoty l vykazuje Φ výrazné oscilace způsobené nespojitostí funkce γ . Při překročení určité hodnoty l_{micro} se hodnota porozity ustálí na konstantní hodnotě a pro hodnoty $l > l_{macro}$ případné makroskopické nehomogenity opět destabilizují hodnotu Φ . Oblasti Ω_0 , pro které existují hodnoty l_{micro} a l_{macro} takové, že pro $l_{micro} \ll l \ll l_{macro}$ hodnota Φ nezávisí na průměru oblasti l , označujeme jako *reprezentativní objem*, zkráceně REV¹. Pokud pro uvažované porézní prostředí nelze pro nějaký bod $x_0 \in \Omega$ určit reprezentativní objem, nelze ke studování proudění v tomto prostředí použít makroskopické metody.

1.2 Fáze

Fází rozumíme chemicky homogenní část zkoumaného systému, která se od ostatních částí systému liší svými fyzikálními vlastnostmi, a která je od ostatních částí systému oddělena jednoznačně určenou hranicí.[1] Podle definice jsou tedy fáze na mikroskopické úrovni nemísitelné. *Pevnou fází* rozumíme pevné látky, které tvoří skelet porézního prostředí. Obsahuje-li uvažovaný systém kromě pevné fáze pouze jednu další fázi, hovoříme o *jednofázovém systému*. Pokud systém obsahuje dvě nebo více dalších fází, jedná se o *vícefázový systém*.

1.3 Saturace

Abychom mohli určit rozložení fází v systému z makroskopického hlediska, definujeme vhodnou makroskopickou veličinu podobným způsobem, jak byla definována porozita. Zkoumejme reprezentativní objem $\Omega_0 \subset \Omega$ tvořený porézním materiálem vyplněným několika fázemi. V mikroskopickém měřítku je v každém bodě $x \in \Omega_0$ přítomen buď skelet porézního prostředí, nebo právě jedna z fází. Lze tedy

¹Representative Elementary Volume

definovat indikátorovou funkci γ_α fáze α

$$\gamma_\alpha(\mathbf{x}, t) = \begin{cases} 1 & \text{pokud bod } \mathbf{x} \text{ leží v čase } t \text{ v prostoru vyplněném fází } \alpha, \\ 0 & \text{jinak,} \end{cases} \quad \forall \mathbf{x} \in \Omega, \quad (1.3)$$

Pomocí funkce γ_α definujeme *saturaci* fáze α pomocí vztahu

$$S_\alpha(\mathbf{x}_0, t) = \frac{\int_{\Omega_0} \gamma_\alpha(\mathbf{x}, t)}{\int_{\Omega_0} \gamma(\mathbf{x}, t)}, \quad (1.4)$$

kde $\mathbf{x}_0 \in \Omega_0$. Veličina S_α je bezrozměrná makroskopická veličina vyjadřující poměr objemu obsazeného fáze α ku objemu pórů v reprezentativním objemu Ω_0 . Přímo z definice (1.4) plynou následující vlastnosti:

$$0 \leq S_\alpha \leq 1, \quad (1.5a)$$

$$\sum_{\alpha} S_\alpha = 1. \quad (1.5b)$$

1.3.1 Reziduální saturace

Je známo, že smáčivou fázi (vodu) nelze z porézního prostředí mechanicky úplně vytlačit. Tento fakt lze matematicky popsat zavedením tzv. *reziduální saturace* smáčivé fáze $S_{w,r}$, která vyjadřuje minimální množství fáze, které bude v porézním prostředí vždy přítomno. Pro popis opačných procesů lze definovat také reziduální saturaci nesmáčivé fáze $S_{n,r}$.

Pro práci s reziduálními saturacemi se dále zavádí *efektivní saturace* S_α^e fáze α ,

$$S_\alpha^e = \frac{S_\alpha - S_{\alpha,r}}{1 - \sum_{\beta} S_{\beta,r}}, \quad (1.6)$$

která postihuje pouze objemový podíl fáze, který lze mechanicky vytlačit. Přímo z definice plyne, že efektivní saturace má stejné vlastnosti jako saturace (1.5).

1.4 Zákon zachování hmoty

Zákony zachování hrají důležitou roli při popisu zachovávajících se veličin. Přírodním požadavkem při popisu proudění několika nemísitelných fází je zachování hmoty každé z fází:

$$\frac{\partial(\phi S_\alpha \rho_\alpha)}{\partial t} + \nabla \cdot (\rho_\alpha \mathbf{v}_\alpha) = f_\alpha, \quad (1.7)$$

kde α je index dané fáze a veličiny vystupující v rovnici mají následující význam:

$\Phi(\mathbf{x})$	$[-]$	Porozita prostředí. Dále předpokládáme, že porozita nezávisí na čase, což odpovídá případu, kdy je skelet porézního prostředí nedeformovatelný.
$S_\alpha(\mathbf{x}, t)$	$[-]$	Saturace fáze α .
$\rho_\alpha(\mathbf{x}, t)$	$[\text{kg} \cdot \text{m}^{-3}]$	Hustota fáze α .
$\mathbf{v}_\alpha(\mathbf{x}, t)$	$[\text{m} \cdot \text{s}^{-1}]$	Průsaková rychlost, neboli Darcyho rychlost fáze α .
$f_\alpha(\mathbf{x}, t)$	$[\text{kg} \cdot \text{m}^{-3} \cdot \text{s}^{-1}]$	Člen charakterizující zdroje a propady.

1.5 Darcyho zákon

Makroskopickou průsakovou rychlost \mathbf{v}_α každé fáze lze určit pomocí Darcyho zákona [7]:

$$\mathbf{v}_\alpha = -\frac{1}{\mu_\alpha} \mathbf{K}_\alpha (\nabla p_\alpha - \rho_\alpha \mathbf{g}), \quad (1.8)$$

kde nové veličiny mají následující význam:

$p_\alpha(\mathbf{x}, t)$	[Pa]	Tlak fáze α .
μ_α	[Pa · s]	Dynamická viskozita fáze α . Předpokládáme, že μ_α je konstantní.
$\mathbf{K}_\alpha(\mathbf{x})$	[m ²]	Propustnost neboli permeabilita porézního prostředí vzhledem k fázi α , obecně se jedná o symetrický, pozitivně definitní tenzor 2. řádu.
\mathbf{g}	[m · s ⁻²]	Vektor gravitačního zrychlení.

Darcyho zákon byl odvozen pro jednofázové proudění. Při rozšíření na vícefázové proudění se předpokládá závislost tenzoru \mathbf{K}_α nejen na vlastnostech prostředí, ale také na saturaci fáze α . Pokud jsou póry vyplněny více fázemi, nemůže fáze α využívat k proudění celý prostor pórů a výsledná průsaková rychlost \mathbf{u}_α je proto nižší, než když jsou póry vyplněny pouze jednou fází.

Při modelování se předpokládá, že tenzor \mathbf{K}_α lze rozložit jako součin $\mathbf{K}_\alpha = k_{r,\alpha}(S_\alpha) \mathbf{K}$, přičemž tenzor *absolutní permeability* (neboli vnitřní propustnosti) \mathbf{K} [m²] závisí pouze na vlastnostech prostředí a *relativní permeabilita* $k_{r,\alpha}$ [–] na saturaci fáze α a implicitně také na vlastnostech prostředí. Relativní permeabilita představuje škálovací faktor splňující $0 \leq k_{r,\alpha}(S_\alpha) \leq 1$. Různé modely relativní permeability jsou popsány v sekci 1.7.

Darcyho zákon lze psát ve tvaru

$$\mathbf{v}_\alpha = -\frac{k_{r,\alpha}}{\mu_\alpha} \mathbf{K} (\nabla p_\alpha - \rho_\alpha \mathbf{g}). \quad (1.9)$$

Pro přehlednost se ještě zavádí značení $\lambda_\alpha = \frac{k_{r,\alpha}}{\mu_\alpha}$, kde λ_α [Pa⁻¹ · s⁻¹] se nazývá *mobilita* fáze α .

1.6 Kapilární tlak

Ve zbytku této práce se omezíme na popis proudění dvou fází v porézním prostředí. Dvoufázové proudění v porézním prostředí je silně ovlivněno kapilaritou. Působení povrchového napětí na ostrém rozhraní mezi fázemi v pórech vyvolává rozdíl mezi mikroskopickými tlaky jednotlivých fází. Na fázovém rozhraní definujeme *mikroskopický kapilární tlak* π_c jako rozdíl mezi tlakem nesmáčivé fáze π_n a tlakem smáčivé fáze π_w

$$\pi_c = \pi_n - \pi_w. \quad (1.10)$$

V důsledku definice smáčivosti je kapilární tlak vždy nezáporná veličina, $\pi_c \geq 0$. Dále bude vždy index n označovat nesmáčivou fázi (např. vzduch) a index w smáčivou fázi (např. voda).

Na makroskopické úrovni definujeme *makroskopický kapilární tlak* p_c přirozeně pomocí stejného vztahu jako v mikroskopickém případě

$$p_c = p_n - p_w, \quad (1.11)$$

kde makroskopické tlaky p_w a p_n lze definovat např. průměrováním příslušného mikroskopického tlaku přes reprezentativní objem (REV). Makroskopický kapilární tlak obecně závisí na saturacích fází, teplotě a rozměrech a uspořádání pórů. Dále uvažujeme pouze závislost kapilárního tlaku na saturacích, $p_c = p_c(S_w)$. Pro matematický popis této závislosti se nejčastěji používají modely van Genuchtena a Brookse a Coreyho.

Brooks a Corey

V modelu Brookse a Coreya [4] je závislost kapilárního tlaku na saturacích určena vztahem

$$p_c(S_w) = p_d(S_w^e)^{-\frac{1}{\lambda}}, \quad (1.12)$$

kde p_d [Pa] a λ [–] jsou parametry prostředí. Parametr p_d představuje vstupní tlak pro dané prostředí a λ souvisí s rozložením velikosti pórů.

Van Genuchten

Ve van Genuchtenově modelu [17] je závislost kapilárního tlaku na saturacích určena vztahem

$$p_c(S_w) = \frac{1}{\alpha} \left((S_w^e)^{-\frac{1}{m}} - 1 \right)^{\frac{1}{n}}, \quad (1.13)$$

kde parametr α [Pa⁻¹] charakterizuje vstupní tlak a parametry m [–] a n [–] popisující strukturu pórů jsou obvykle svázány vztahem $m = 1 - \frac{1}{n}$.

1.7 Relativní permeabilita

Permeabilita prostředí vzhledem k fázi α byla zavedena v sekci 1.5 vztahem $\mathbf{K}_\alpha = k_{r,\alpha}(S_\alpha)\mathbf{K}$. Závislostí $k_{r,\alpha} = k_{r,\alpha}(S_\alpha)$ modelujeme skutečnost, že průchodu fáze α prostředím brání kromě skeletu také přítomnost ostatních fází. V případě dvoufázového proudění lze funkce $k_{r,w}$ a $k_{r,n}$ popisující závislost relativní permeability na saturaci odvodit z příslušného modelu kapilarity [19].

Burdine

Modelu Brookse a Coreya odpovídá Burdineův model [5, 6]:

$$k_{r,w}(S_w) = (S_w^e)^{\frac{2+3\lambda}{\lambda}}, \quad (1.14a)$$

$$k_{r,n}(S_n) = (S_n^e)^2 \left(1 - (1 - S_n^e)^{\frac{2+\lambda}{\lambda}} \right), \quad (1.14b)$$

kde p_d a λ jsou parametry Brooksova a Coreya modelu.

Mualem

Modelu van Genuchtena odpovídá Mualemův model [26]:

$$k_{r,w}(S_w) = (S_w^e)^{\frac{1}{2}} \left(1 - (1 - (S_w^e)^{\frac{1}{m}})^m \right)^2, \quad (1.15a)$$

$$k_{r,n}(S_n) = (S_n^e)^{\frac{1}{3}} \left(1 - (1 - S_n^e)^{\frac{1}{m}} \right)^{2m}, \quad (1.15b)$$

kde m a n jsou parametry van Genuchtenova modelu.

1.8 Rovnice dvoufázového proudění

Shrnutím předchozích sekcí dostáváme následující soustavu rovnic popisující dvoufázové nemísivé proudění v porézním prostředí:

$$\Phi \frac{\partial(S_w \rho_w)}{\partial t} + \nabla \cdot (\rho_w \mathbf{v}_w) = f_w, \quad (1.16a)$$

$$\mathbf{v}_w = -\lambda_w \mathbf{K}(\nabla p_w - \rho_w \mathbf{g}), \quad (1.16b)$$

$$\Phi \frac{\partial(S_n \rho_n)}{\partial t} + \nabla \cdot (\rho_n \mathbf{v}_n) = f_n, \quad (1.16c)$$

$$\mathbf{v}_n = -\lambda_n \mathbf{K}(\nabla p_n - \rho_n \mathbf{g}), \quad (1.16d)$$

$$p_c = p_n - p_w, \quad (1.16e)$$

$$S_w + S_n = 1, \quad (1.16f)$$

kde neznámé funkce jsou saturace $S_\alpha = S_\alpha(\mathbf{x}, t)$ a tlaky $p_\alpha = p_\alpha(\mathbf{x}, t)$ fází $\alpha \in \{w, n\}$ pro $\mathbf{x} \in \Omega$ a $t > 0$. Pro nestlačitelné fáze je soustava (1.16) kompletní, v případě stlačitelných fází by bylo potřeba předepsat ještě stavové rovnice určující závislost hustoty fáze na tlaku.

Kapitola 2

Obecná formulace

Úlohu (1.16) budeme řešit pomocí numerického schématu pro obecnou soustavu n parciálních diferenciálních rovnic ve tvaru

$$\sum_{j=1}^n N_{i,j} \frac{\partial Z_j}{\partial t} + \nabla \cdot \left[m_i \left(- \sum_{j=1}^n \mathbf{D}_{i,j} \nabla Z_j + \mathbf{w}_i \right) \right] = f_i \quad (2.1)$$

pro $i = 1, \dots, n$, kde $\mathbf{Z} = [Z_1, \dots, Z_n]^T$ je vektor neznámých funkcí závislých na poloze $\mathbf{x} \in \Omega \subset \mathbb{R}^d$ a čase $t \in (0, T)$, kde $d = 1, 2$ nebo 3 označuje dimenzi prostoru a $T > 0$ určuje konečný čas. Soustava (2.1) je doplněna o počáteční podmínku

$$Z_j(\mathbf{x}, 0) = Z_j^{ini}(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega, \quad j = 1, \dots, n, \quad (2.2)$$

a okrajové podmínky pro všechna $t \in (0, T)$,

$$Z_j(\mathbf{x}, t) = Z_j^{\mathcal{D}}(\mathbf{x}, t), \quad \forall \mathbf{x} \in \Gamma_j^{\mathcal{D}} \subset \partial\Omega, \quad j = 1, \dots, n, \quad (2.3a)$$

$$\mathbf{v}_i(\mathbf{x}, t) \cdot \mathbf{n}_{\partial\Omega}(\mathbf{x}) = v_i^{\mathcal{N}}(\mathbf{x}, t), \quad \forall \mathbf{x} \in \Gamma_i^{\mathcal{N}} \subset \partial\Omega, \quad i = 1, \dots, n, \quad (2.3b)$$

kde \mathbf{v}_i označuje rychlostní člen

$$\mathbf{v}_i = - \sum_{j=1}^n \mathbf{D}_{i,j} \nabla Z_j + \mathbf{w}_i, \quad (2.4)$$

a rozklad hranice $\partial\Omega$ pro všechna $j = 1, \dots, n$ splňuje

$$\Gamma_j^{\mathcal{D}} \cup \Gamma_j^{\mathcal{N}} = \partial\Omega, \quad (2.5a)$$

$$\Gamma_j^{\mathcal{D}} \cap \Gamma_j^{\mathcal{N}} = \emptyset. \quad (2.5b)$$

Koeficienty vystupující v soustavě (2.1) mají následující význam: tenzor $\mathbf{N} = [N_{i,j}]_{i,j=1}^n$ představuje matici útlumu, $\mathbf{m} = [m_i]_{i=1}^n$ je nezáporný vektor koeficientů mobility, $\mathbf{D} = [\mathbf{D}_{i,j}]_{i,j=1}^n$ je tenzor nulových nebo symetrických a pozitivně definitních tenzorů difúze, přičemž předpokládáme, že pro každé i existuje alespoň jeden index j takový, že $\mathbf{D}_{i,j}$ je nenulový, $\mathbf{w} = [\mathbf{w}_i]_{i=1}^n$ představuje vektor externích konzervativních sil a $\mathbf{f} = [f_i]_{i=1}^n$ představuje vektor zdrojových členů.

Volba koeficientů a neznámých v obecné rovnici (2.1) závisí na konkrétní úloze a její formulaci. Pro úlohu dvoufázového proudění v porézním prostředí (1.16) zvolíme s ohledem na použité numerické schéma za primární neznámé funkce fázové tlaky, $Z_1 = p_w$ a $Z_2 = p_n$. Po vyjádření zbylých neznámých

pomocí fázových tlaků a provedení úprav získáme následující volbu koeficientů:

$$\mathbf{N} = \begin{pmatrix} -\Phi\rho_w \frac{dS_w}{dp_c} + \Phi S_w \frac{\partial \rho_w}{\partial p_w} & \Phi\rho_w \frac{dS_w}{dp_c} \\ \Phi\rho_n \frac{dS_w}{dp_c} & -\Phi\rho_n \frac{dS_w}{dp_c} + \Phi S_n \frac{\partial \rho_n}{\partial p_n} \end{pmatrix}, \quad (2.6a)$$

$$\mathbf{m} = \begin{pmatrix} \rho_w \frac{\lambda_w}{\lambda_t} \\ \rho_n \frac{\lambda_n}{\lambda_t} \end{pmatrix}, \quad (2.6b)$$

$$\mathbf{D} = \begin{pmatrix} \lambda_t \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \lambda_t \mathbf{K} \end{pmatrix}, \quad (2.6c)$$

$$\mathbf{w} = \begin{pmatrix} \lambda_t \rho_w \mathbf{K} \mathbf{g} \\ \lambda_t \rho_n \mathbf{K} \mathbf{g} \end{pmatrix}, \quad (2.6d)$$

$$\mathbf{f} = \begin{pmatrix} f_w \\ f_n \end{pmatrix}, \quad (2.6e)$$

kde $\lambda_t = \lambda_w + \lambda_n$. Pro volbu koeficientů \mathbf{m} a \mathbf{D} byl využit postup z článku [21]. Pokud uvažujeme fázi α jako nestlačitelnou, bude výraz $\frac{\partial \rho_\alpha}{\partial p_\alpha}$ nulový (tj. hustota nezávisí na tlaku) a koeficient \mathbf{N} se dále zjednoduší.

Kapitola 3

Numerické schéma

V této kapitole popíšeme numerické schéma pro řešení soustavy (2.1). Schéma je založeno na kombinaci hybridní metody smíšených konečných prvků (MHFEM¹) a metody konečných objemů (FVM²) pro prostorovou diskretizaci, Eulerovy metody pro časovou diskretizaci a semi-implicitním přístupem metody zamrzlých koeficientů pro linearizaci v čase. Podobnou formulaci numerického schématu pro řešení rovnic dvoufázového proudění lze nalézt v [13, 14, 21].

Uvažujeme diskretizaci prostorové oblasti $\Omega \subset \mathbb{R}^d$ pomocí numerické sítě \mathcal{K}_h tvořené konečným počtem elementů K , kde $h > 0$ označuje průměr největšího elementu. Předpokládáme, že síť je konformní, tj. průnik dvou různých elementů je buď prázdný, vrchol, hrana nebo stěna. Pro aplikaci numerického schématu se v této práci omezíme na ekvidistantní strukturované sítě tvořené v \mathbb{R}^1 úsečkami a v \mathbb{R}^2 obdélníky. Schéma je však obecnější a dobře použitelné např. na nestrukturovaných sítích tvořených trojúhelníky v \mathbb{R}^2 . V následujících sekcích značíme počty všech vrcholů, hran a elementů sítě \mathcal{K}_h po řadě N_V , N_E a N_K . Množinu vnějších, resp. vnitřních hran sítě \mathcal{K}_h označíme jako \mathcal{E}_h^{ext} , resp. \mathcal{E}_h^{int} . Pro množinu hran příslušných danému elementu $K \in \mathcal{K}_h$ zavedeme symbol \mathcal{E}_K .

3.1 Variační formulace

Předpokládáme, že vektorové funkce \mathbf{v}_i jsou pro všechna $i = 1, \dots, n$ prvky funkčního prostoru $\mathbf{H}(\text{div}, \Omega)$, kde

$$\mathbf{H}(\text{div}, \Omega) = \left\{ \boldsymbol{\omega} \in [L^2(\Omega)]^d \mid \nabla \cdot \boldsymbol{\omega} \in L^2(\Omega) \right\}. \quad (3.1)$$

Smíšenou variační formulaci úlohy (2.1), (2.4) lze formálně odvodit vynásobením rovnice (2.1) funkcí $\varphi \in V \equiv L^2(\Omega)$ a rovnice (2.4) funkcí $\boldsymbol{\omega} \in \mathbf{W} \equiv \mathbf{H}(\text{div}, \Omega)$ a integrací obou rovnic přes oblast Ω . Pro nalezení přibližného řešení $Z_i \in V$, $\mathbf{v}_i \in \mathbf{W}$, $i = 1, \dots, n$ variační úlohy pomocí metody konečných prvků přejdeme k vhodným konečněrozměrným podprostorům $V_h \subset V$, $\mathbf{W}_h \subset \mathbf{W}$. V této práci použijeme prostory nejnižšího řádu, prostor po částech konstantních funkcí $D_0(\mathcal{K}_h)$ a Raviartův–Thomasův–Nédélecův prostor $\mathbf{RT}_0(\mathcal{K}_h)$.

3.1.1 Prostor \mathbf{RT}_0

Při definici Raviartova–Thomasova–Nédélecova prostoru $\mathbf{RT}_0(\mathcal{K}_h)$ využijeme diskretizaci oblasti Ω pomocí numerické sítě \mathcal{K}_h . Nejprve na každém elementu $K \in \mathcal{K}_h$ definujeme lokální prostor $\mathbf{RT}_0(K)$ generovaný bazickými funkcemi $\boldsymbol{\omega}_{K,E} \in \mathbf{H}(\text{div}, K)$, $E \in \mathcal{E}_K$. Konkrétní podoba prostoru $\mathbf{RT}_0(K)$ závisí na dimenzi prostoru a na geometrii elementu K , obecně je však výhodné volit bazické funkce

¹Mixed Hybrid Finite Element Method

²Finite Volume Method

$\omega_{K,E}$ tak, aby splňovaly podmínky

$$\nabla \cdot \omega_{K,E}(\mathbf{x}) = \frac{1}{|K|_d}, \quad \forall E \in \mathcal{E}_K, \forall \mathbf{x} \in K, \quad (3.2a)$$

$$\omega_{K,E}(\mathbf{x}) \cdot \mathbf{n}_{K,F} = \frac{1}{|F|_{d-1}} \delta_{EF}, \quad \forall E, F \in \mathcal{E}_K, \forall \mathbf{x} \in F, \quad (3.2b)$$

kde $\mathbf{n}_{K,F}$ označuje jednotkový vektor vnější normály na hraně $F \in \mathcal{E}_K$ vzhledem k elementu K , $|\cdot|_d$ značí d -rozměrnou Lebesgueovu míru (definujeme $|E|_0 = 1$ pro všechna $E \in \mathcal{E}_K$) a δ_{EF} představuje Kroneckerovo delta. Z podmínek (3.2) dále plyne

$$\int_K \nabla \cdot \omega_{K,E}(\mathbf{x}) = 1, \quad \forall E \in \mathcal{E}_K, \quad (3.3a)$$

$$\int_F \omega_{K,E}(\mathbf{x}) \cdot \mathbf{n}_{K,F} = \delta_{EF}, \quad \forall E, F \in \mathcal{E}_K. \quad (3.3b)$$

Pomocí lokálních prostorů $\mathbf{RT}_0(K)$ definujeme prostor $\mathbf{RT}_0(\mathcal{K}_h)$ tak, aby se jednalo o podprostor prostoru $\mathbf{H}(\text{div}, \Omega)$. Podle definice platí $\mathbf{RT}_0(K) \subset \mathbf{H}(\text{div}, K)$. Definujeme tedy prostor

$$\mathbf{RT}_0^0(\mathcal{K}_h) = \left\{ \omega \in [L^2(\Omega)]^d \mid \forall K \in \mathcal{K}_h, \omega|_K \in \mathbf{H}(\text{div}, K) \right\}. \quad (3.4)$$

Prostor $\mathbf{RT}_0^0(\mathcal{K}_h)$ však není podprostorem prostoru $\mathbf{H}(\text{div}, \Omega)$. Ve smyslu tvrzení 1.2 v knize [3, str. 95] navíc po funkcích $\omega \in \mathbf{RT}_0^0(\mathcal{K}_h)$ požadujeme spojitost normálových stop na vnitřních hranách sítě. Tuto vlastnost můžeme ekvivalentně vyjádřit pomocí bilanční podmínky, kterou použijeme pro definici prostoru $\mathbf{RT}_0(\mathcal{K}_h)$:

$$\mathbf{RT}_0(\mathcal{K}_h) = \left\{ \omega \in [L^2(\Omega)]^d \mid \forall K \in \mathcal{K}_h, \omega|_K \in \mathbf{RT}_0(K); \forall E \in \mathcal{E}_h^{\text{int}}, E \in \mathcal{E}_{K_1} \cap \mathcal{E}_{K_2}, \right. \\ \left. \int_E \omega|_{K_1} \cdot \mathbf{n}_{K_1,E} + \int_E \omega|_{K_2} \cdot \mathbf{n}_{K_2,E} = 0 \right\}. \quad (3.5)$$

Bazické funkce v \mathbb{R}^1

V \mathbb{R}^1 jsou všechny elementy $K \in \mathcal{K}_h$ úsečky definované dvojicí vrcholů $V_1, V_2 \in \mathcal{V}_K, V_1 < V_2$. Zde tedy $\mathcal{V}_K = \mathcal{E}_K$ a bazické funkce prostoru $\mathbf{RT}_0(K)$ volíme ve tvaru

$$\omega_{K,V_\iota} = \frac{1}{|K|_1} (x - V_\iota), \quad (3.6)$$

pro všechna $\iota \in \{1, 2\}$ a $x \in K = [V_1, V_2]$.

Bazické funkce v \mathbb{R}^2

V \mathbb{R}^2 pro obdélníkový element $K = (0, h_x) \times (0, h_y)$ volíme bazické funkce prostoru $\mathbf{RT}_0(K)$ ve tvaru

$$\omega_{K,E_1}(\mathbf{x}) = \frac{1}{|K|_2} \begin{pmatrix} x_1 - h_x \\ 0 \end{pmatrix}, \quad \omega_{K,E_2}(\mathbf{x}) = \frac{1}{|K|_2} \begin{pmatrix} x_1 \\ 0 \end{pmatrix}, \\ \omega_{K,E_3}(\mathbf{x}) = \frac{1}{|K|_2} \begin{pmatrix} 0 \\ x_2 - h_y \end{pmatrix}, \quad \omega_{K,E_4}(\mathbf{x}) = \frac{1}{|K|_2} \begin{pmatrix} 0 \\ x_2 \end{pmatrix}, \quad (3.7)$$

kde x_1, x_2 značí složky vektoru $\mathbf{x} \in K \subset \mathbb{R}^2$ a E_1, E_2, E_3, E_4 značí postupně levou, pravou, dolní a horní hranu elementu K .

3.1.2 Aproximace rychlostí

Vyjádřením aproximace funkcí \mathbf{v}_i na elementu $K \in \mathcal{K}_h$ pomocí bazických funkcí prostoru $\mathbf{RT}_0(K)$ získáme

$$\mathbf{v}_i|_K = \sum_{E \in \mathcal{E}_K} v_{i,K,E} \boldsymbol{\omega}_{K,E}. \quad (3.8)$$

Integrací normálové stopy $\mathbf{v}_i|_K \cdot \mathbf{n}_{K,E}$ vzhledem k elementu K přes hranu $E \in \mathcal{E}_K$ a s využitím vlastností bazických funkcí (3.3) dostaneme

$$\int_E \mathbf{v}_i|_K \cdot \mathbf{n}_{K,E} = \int_E \sum_{F \in \mathcal{E}_K} v_{i,K,F} \boldsymbol{\omega}_{K,F} \cdot \mathbf{n}_{K,E} = v_{i,K,E}. \quad (3.9)$$

Koeficienty $v_{i,K,E}$ tedy odpovídají numerickému toku funkce \mathbf{v}_i přes hranu $E \in \mathcal{E}_K$ ve směru vnější normály vzhledem k elementu K . Jelikož předpokládáme, že funkce \mathbf{v}_i patří do prostoru $\mathbf{H}(\text{div}, \Omega)$, musí splňovat podmínku spojitosti normálových stop na vnitřních hranách sítě. Na hraně E oddělující elementy K_1 a K_2 tedy musí platit

$$v_{i,K_1,E} + v_{i,K_2,E} = 0. \quad (3.10)$$

Tato podmínka odpovídá předpokladu nulových zdrojových členů na hraně E a představuje tedy lokální bilanci veličiny \mathbf{v}_i . Standardnímu zákonu zachování hmoty však odpovídá lokální bilance toku $\mathbf{q}_i = m_i \mathbf{v}_i$. Problém nastává, pokud je mobilita m_i rovna nule, což může nastat například při modelování vícefázového proudění, kdy se příslušná fáze v daném místě nevyskytuje. Degenerace mobility způsobí, že bilanční podmínka typu (3.10) pro tok \mathbf{q}_i by nebyla jednoznačně řešitelná. Proto ke konstrukci numerického schématu využijeme pouze bilanci veličiny \mathbf{v}_i a mobilitu m_i diskretizujeme jiným způsobem v následující sekci.

Dále potřebujeme vyjádřit koeficienty $v_{i,K,E}$ pomocí diskretizovaných neznámých proměnných Z_j . Definujme parciální rychlosti $\mathbf{v}_{i,j}$ a předpokládejme, že $\mathbf{v}_{i,j}$ patří do prostoru $\mathbf{H}(\text{div}, \Omega)$:

$$\mathbf{v}_{i,j} = -\mathbf{D}_{i,j} \nabla Z_j. \quad (3.11)$$

Aproximaci $\mathbf{v}_{i,j}$ v lokálním prostoru $\mathbf{RT}_0(K)$ můžeme zapsat pomocí

$$\mathbf{v}_{i,j}|_K = \sum_{E \in \mathcal{E}_K} v_{i,j,K,E} \boldsymbol{\omega}_{K,E}, \quad (3.12)$$

kde $\boldsymbol{\omega}_{K,E}$ označuje bazickou funkci prostoru $\mathbf{RT}_0(K)$. Předpokládáme, že tenzor $\mathbf{D}_{i,j}$ je buď nulový nebo pozitivně definitní. Pokud je $\mathbf{D}_{i,j}$ na elementu K nulový, je i projekce $\mathbf{v}_{i,j}|_K$ nulová, a tedy i koeficienty $v_{i,j,K,E}$ musí být nulové. Pokud je $\mathbf{D}_{i,j}$ na K pozitivně definitní, můžeme rovnici (3.11) vynásobit jeho inverzí a požadovat splnění výsledné rovnosti ve slabém smyslu:

$$\sum_{E \in \mathcal{E}_K} v_{i,j,K,E} \int_K \boldsymbol{\omega}_{K,F}^T \mathbf{D}_{i,j}^{-1} \boldsymbol{\omega}_{K,E} = - \int_K \boldsymbol{\omega}_{K,F}^T \nabla Z_j, \quad \forall F \in \mathcal{E}_K, \quad (3.13)$$

kde $\omega_{K,F}$ označuje bazickou funkci prostoru $\mathbf{RT}_0(K)$. Dále s využitím Greenovy formule a vlastností (3.3) pro úpravu pravé strany (3.13) dostaneme

$$-\int_K \omega_{K,F}^T \nabla Z_j = -\int_{\partial K} Z_j \omega_{K,F}^T \mathbf{n}_{\partial K} + \int_K Z_j \nabla \cdot \omega_{K,F} = -Z_{j,F} + Z_{j,K}, \quad (3.14)$$

kde $Z_{j,K}$ a $Z_{j,F}$ značí střední hodnotu veličiny Z_j na elementu K resp. na hraně $F \in \mathcal{E}_K$. Použili jsme označení $Z_{j,F}$ namísto $Z_{j,K,F}$, jelikož předpokládáme spojitost veličiny Z_j na hranicích všech elementů. To je v souladu s globálním pohledem na variační formulaci řešené úlohy, viz [3, kap. 5.1.2].

Při označení $B_{i,j,K,E,F} = \int_K \omega_{K,E}^T \mathbf{D}_{i,j}^{-1} \omega_{K,F}$ můžeme rovnost (3.13) zapsat ve tvaru

$$\sum_{E \in \partial K} v_{i,j,K,E} B_{i,j,K,E,F} = Z_{j,K} - Z_{j,F}. \quad (3.15)$$

Koeficienty $B_{i,j,K,E,F}$ tvoří na elementu K pozitivně definitní matici $\mathbf{B}_{i,j,K} = [B_{i,j,K,E,F}]_{E,F \in \mathcal{E}_K}$ a označme její inverzi $\mathbf{b}_{i,j,K}$. Inverzí vztahu (3.15) získáme

$$v_{i,j,K,E} = b_{i,j,K,E} Z_{j,K} - \sum_{F \in \mathcal{E}_K} b_{i,j,K,E,F} Z_{j,F}, \quad (3.16)$$

kde $b_{i,j,K,E} = \sum_{F \in \mathcal{E}_K} b_{i,j,K,E,F}$. Pomocí (3.16) již můžeme vyjádřit koeficienty $v_{i,K,E}$ pomocí $Z_{j,K}$ a $Z_{j,F}$:

$$v_{i,K,E} = \sum_{j=1}^n \left(b_{i,j,K,E} Z_{j,K} - \sum_{F \in \mathcal{E}_K} b_{i,j,K,E,F} Z_{j,F} \right) + w_{i,K,E}, \quad (3.17)$$

kde $w_{i,K,E}$ označuje koeficient projekce vektoru \mathbf{w}_i do směru $\omega_{K,E}$ v prostoru $\mathbf{RT}_0(K)$:

$$\mathbf{w}_i|_K = \sum_{E \in \mathcal{E}_K} w_{i,K,E} \omega_{K,E}. \quad (3.18)$$

3.1.3 Aproximace skalárních veličin

Variační aproximace řešení i -té rovnice (2.1) v prostorech nejnižšího řádu odpovídá použití metody konečných objemů s centrem konečného objemu v těžišti elementu (CFVM³). Máme konečněrozměrný prostor $V_h = D_0(\mathcal{K}_h)$ funkcí po částech konstantních na elementech $K \in \mathcal{K}_h$, jehož bazické funkce jsou

$$\varphi_K(\mathbf{x}) = \begin{cases} 1 & \text{pokud } \mathbf{x} \in K, \\ 0 & \text{jinak.} \end{cases} \quad (3.19)$$

Aproximace neznámé Z_j pak nabývá tvaru

$$Z_j(\mathbf{x}, t) \approx \sum_{K \in \mathcal{K}_h} Z_{j,K}(t) \varphi_K(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega, \quad \forall t \in (0, T). \quad (3.20)$$

Koeficienty $Z_{j,K}$ vystupující v (3.20) závisí na čase a mají význam střední hodnoty veličiny Z_j na elementu $K \in \mathcal{K}_h$.

³Cell-Centered Finite Volume Method

Vynásobením i -té rovnice (2.1) bazickou funkcí φ_K a integrací přes oblast Ω dostaneme

$$\sum_{j=1}^n \int_K N_{i,j} \frac{\partial Z_j}{\partial t} + \int_K \nabla \cdot (m_i \mathbf{v}_i) = \int_K f_i. \quad (3.21)$$

První integrál na levé straně a integrál na pravé straně rovnice (3.21) nahradíme středními hodnotami příslušných veličin. Druhý integrál na levé straně rovnice (3.21) upravíme pomocí Greenovy formule, čímž dostaneme

$$\sum_{j=1}^n N_{i,j,K} \frac{dZ_{j,K}(t)}{dt} + \frac{1}{|K|_d} \sum_{E \in \mathcal{E}_K} m_{i,E}^{upw}(t) \underbrace{\int_E \mathbf{v}_i(\mathbf{x}, t)|_K \cdot \mathbf{n}_{K,E}(\mathbf{x})}_{v_{i,K,E}(t)} = f_{i,K}(t), \quad (3.22)$$

kde $\mathbf{n}_{K,E}$ označuje jednotkový vektor vnější normály na hraně E vzhledem k elementu K a $m_{i,E}^{upw}(t)$ představuje zatím blíže neurčenou hodnotu veličiny m_i na hraně E , která je v případě vnitřní hrany společná pro oba elementy sousedící s hranou E . Volbu koeficientů $m_{i,E}^{k,upw}$ upřesníme později.

Pro časovou diskretizaci schématu využijeme rozdělení intervalu $[0, T]$ na malé úseky s množinou hraničních bodů \mathcal{T} ,

$$\mathcal{T} = \{0 = t_0 < t_1 < \dots < t_M = T\}, \quad (3.23)$$

přičemž délky časových kroků označíme jako $\tau_k = t_{k+1} - t_k$. Dále budeme hodnoty časově závislých funkcí vyčíslěných na časové hladině $t_k \in \mathcal{T}$ značit pomocí horního indexu k .

Nelineární semi-diskrétní schéma (3.22) diskretizujeme v čase pomocí zpětné Eulerovy metody a linearizujeme pomocí metody zamrzlých koeficientů. Tím získáme rovnice diskrétního schématu

$$\sum_{j=1}^n N_{i,j,K} \frac{Z_{j,K}^{k+1} - Z_{j,K}^k}{\tau_k} + \frac{1}{|K|_d} \sum_{E \in \mathcal{E}_K} m_{i,E}^{k,upw} v_{i,K,E}^{k+1} = f_{i,K}^{k+1}, \quad \forall i = 1, \dots, n, \quad (3.24)$$

platné pro všechny elementy $K \in \mathcal{K}_h$ a časové hladiny $k = 0, \dots, M-1$.

Nyní již můžeme upřesnit volbu koeficientů $m_{i,E}^{k,upw}$. Pro zajištění stability schématu pro advekční úlohy použijeme stabilizační techniku *upwind* [23] a společnou hodnotu veličiny m_i na hraně E volíme dle směru numerického toku funkce v_i přes tuto hranu na stejné časové hladině:

$$m_{i,E}^{k,upw} = \begin{cases} m_{i,K_1}^k & \text{pokud } v_{i,K_1,E}^k \geq 0, \\ m_{i,K_2}^k & \text{jinak,} \end{cases} \quad \forall E \in \mathcal{E}_h^{int}, E \in \mathcal{E}_{K_1} \cap \mathcal{E}_{K_2}, \quad (3.25a)$$

$$m_{i,E}^{k,upw} = \begin{cases} m_{i,K_1}^k & \text{pokud } v_{i,K_1,E}^k \geq 0, \\ m_{i,E}^{\mathcal{D}}(t_k) & \text{jinak,} \end{cases} \quad \forall E \in \mathcal{E}_h^{ext}, E \in \mathcal{E}_{K_1}, \quad (3.25b)$$

kde m_{i,K_l}^k označuje střední hodnotu m_i na elementu $K_l \in \mathcal{K}_h$ a $m_{i,E}^{\mathcal{D}}$ představuje střední hodnotu m_i na hraně $E \in \mathcal{E}_h^{ext}$ předepsanou dle Dirichletovy okrajové podmínky (3.28). Zde je nutné podotknout, že pro korektnost volby upwindové hodnoty na hranici oblasti $\partial\Omega$ je potřeba znát hodnotu upwindované veličiny kromě Dirichletovy části hranice $\Gamma_i^{\mathcal{D}}$ také na části Neumannovy hranice $\Gamma_i^{\mathcal{N}*} \subset \Gamma_i^{\mathcal{N}}$, kde tekutina vtéká dovnitř oblasti, tedy

$$\Gamma_i^{\mathcal{N}*} = \left\{ \mathbf{x} \in \Gamma_i^{\mathcal{N}} \mid \mathbf{v}_i(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}) < 0 \right\}. \quad (3.26)$$

3.2 Sestavení globální soustavy rovnic

Pro spojení rovnic (3.17) a (3.24) potřebujeme vyjádřit hodnoty koeficientů $v_{i,K,E}$ v čase $t = t_{k+1}$. Nelineární vztah (3.17) linearizujeme v čase pomocí metody zamrzlých koeficientů. Při výpočtu hodnot $v_{i,K,E}^{k+1}$ využijeme hodnoty $b_{i,K,E,F}^k$ a $w_{i,K,E}^k$ z předchozí časové hladiny, čímž získáme lineární vztah pro $Z_{i,K}^{k+1}$ a $Z_{i,E}^{k+1}$:

$$v_{i,K,E}^{k+1} = \sum_{j=1}^n \left(b_{i,j,K,E}^k Z_{j,K}^{k+1} - \sum_{F \in \mathcal{E}_K} b_{i,j,K,E,F}^k Z_{j,F}^{k+1} \right) + w_{i,K,E}^k, \quad (3.27)$$

Proto diskretní schéma (3.24) označujeme jako semi-implicitní.

Dále směřujeme k sestavení soustavy rovnic pro neznámé $Z_{i,K}^{k+1}$, $Z_{i,E}^{k+1}$. K tomu nám poslouží bilanční podmínka (3.10) pro rychlosti \mathbf{v}_i platná pro vnitřní hrany $E \in \mathcal{E}_h^{int}$ nezávisle na časové hladině. Potřebné vztahy pro okrajové hrany $E \in \mathcal{E}_h^{ext}$ získáme diskretizací okrajových podmínek.

3.2.1 Diskretizace okrajových podmínek

Při diskretizaci okrajových podmínek předpokládáme, že na všech vnějších hranách $E \in \mathcal{E}_h^{ext}$ je dána podmínka právě jednoho typu, tedy pro všechna $j = 1, \dots, n$ platí buď $E \subset \Gamma_j^D$ nebo $E \subset \Gamma_j^N$. Tento předpoklad lze chápat jako diskretizaci podmínky (2.5).

Na Dirichletově části hranice Γ_j^D dle (2.3a) předepisujeme přímo hodnotu funkce Z_j , diskretizací na okrajové hraně E získáme přímo střední hodnotu $Z_{j,E}^k$:

$$Z_{j,E}^k = \frac{1}{|E|_{d-1}} \int_E Z_j^D(\mathbf{x}, t_k) \stackrel{\text{ozn}}{=} Z_{j,E}^D(t_k). \quad (3.28)$$

Diskretizací Neumannovy podmínky (2.3b) na okrajové hraně $E \in \mathcal{E}_K$ získáme hodnotu koeficientu $v_{i,K,E}$, který má podle (3.9) význam toku funkce \mathbf{v}_i přes hranu E ve směru vnější normály:

$$v_{i,K,E}^k = \int_E v_i^N(\mathbf{x}, t_k) \stackrel{\text{ozn}}{=} v_{i,E}^N(t_k). \quad (3.29)$$

3.2.2 Redukce soustavy

Spojením rovnic (3.24), (3.27), bilance rychlostí na vnitřních hranách (3.10) a okrajových podmínek (3.28) a (3.29) pro $i = 1, \dots, n$ získáme soustavu $n(N_K + N_E)$ lineárních rovnic pro stejný počet neznámých $Z_{j,K}^{k+1}$, $Z_{j,E}^{k+1}$, $j = 1, \dots, n$. Tato soustava je však pro řešení zbytečně velká – můžeme totiž využít její blokové struktury a pomocí metody hybridizace snížit počet rovnic v úloze.

Z rovnic (3.24) a (3.27) lze vyjádřit neznámé hodnoty $Z_{j,K}^{k+1}$ pomocí $Z_{j,E}^{k+1}$ ve tvaru

$$\mathbf{Z}_K^{k+1} = \sum_{F \in \mathcal{E}_K} \mathbf{Q}_K^{-1} \mathbf{R}_{K,F} \mathbf{Z}_F^{k+1} + \mathbf{Q}_K^{-1} \mathbf{R}_K, \quad (3.30)$$

kde vektor $\mathbf{Z}_K^{k+1} \in \mathbb{R}^n$ obsahuje neznámé hodnoty $Z_{j,K}^{k+1}$, $j = 1, \dots, n$ a prvky matic \mathbf{Q}_K a $\mathbf{R}_{K,F}$ a vektoru \mathbf{R}_K jsou následující:

$$[\mathbf{Q}_K]_{i,j} = \frac{|K|_d}{\tau_k} N_{i,j,K} + \sum_{E \in \mathcal{E}_K} m_{i,E}^{k,upw} b_{i,j,K,E}, \quad (3.31)$$

$$[\mathbf{R}_{K,F}]_{i,j} = \sum_{E \in \mathcal{E}_K} m_{i,E}^{k,upw} b_{i,j,K,E,F}, \quad (3.32)$$

$$[\mathbf{R}_K]_i = |K|_d f_{i,K} + \frac{|K|_d}{\tau_k} \sum_{j=1}^n N_{i,j,K} Z_{j,K}^k - \sum_{E \in \mathcal{E}_K} m_{i,E}^{k,upw} w_{i,K,E}. \quad (3.33)$$

Hodnoty Z_K^{k+1} vyjádřené pomocí (3.30) dosadíme do (3.10) a spolu s okrajovými podmínkami (3.28) a (3.29) získáme soustavu lineárních algebraických rovnic pro neznámé $Z_{j,F}^{k+1}$, kterou symbolicky zapíšeme v maticovém tvaru

$$\mathbf{A} \mathbf{Z}^{k+1} = \mathbf{b}, \quad (3.34)$$

kde $\mathbf{Z}^{k+1} = [[Z_{j,F}^{k+1}]_{j=1}^n]_{F \in \mathcal{E}_h}$ je vektor neznámých. Neznámé v soustavě (3.34) nazýváme *stupni volnosti* (DOF⁴), počet neznámých je $N_{DOF} = nN_E$. Matice soustavy \mathbf{A} je za uvedených předpokladů řádká a regulární. Řešení $\mathbf{Z}^{k+1} = \mathbf{A}^{-1} \mathbf{b}$ spolu s explicitním vztahem (3.30) umožňuje získat všechny hodnoty potřebné pro přechod na následující časovou hladinu $t = t_{k+1}$.

3.3 Algoritmus výpočtu

Soustava (3.34) spolu s explicitním vztahem (3.30) udávají, jak spočítat aproximaci řešení \mathbf{Z}^{k+1} v čase t_{k+1} pomocí aproximace \mathbf{Z}^k z předchozí časové hladiny t_k . Abychom získali aproximaci řešení v čase $t = T$ pomocí počátečního stavu v čase t_0 , musíme iteračně určit aproximaci ve všech časových hladinách $t_k \in \mathcal{T}$. Podrobný algoritmus výpočtu numerického řešení pak vypadá takto:

Algoritmus 1

1. Zvolíme $h > 0$ a oblast $\Omega \subset \mathbb{R}^d$, $d = 1$ nebo 2 pokryjeme strukturovanou konformní sítí \mathcal{K}_h . Časový interval diskretizujeme pomocí množiny \mathcal{T} .
2. Načteme vstupní data charakterizující úlohu, tj. data popisující počáteční a okrajové podmínky a koeficienty rovnice (2.1) v diskrétní podobě.
3. Položíme $k = 0$, $t = t_0 = 0$.
4. Inicializujeme koeficienty $N_{i,j,K}$, $m_{i,K}$, $\mathbf{D}_{i,j,K}$, $w_{i,K,E}$, $f_{i,K}$ pro všechna $i, j = 1, \dots, n$, $K \in \mathcal{K}_h$ a $E \in \mathcal{E}_K$.
5. Střední hodnoty mobility na hranách $m_{i,E}^{upw}$ inicializujeme pro vnitřní hrany $E \in \mathcal{E}_h^{int}$ ze vztahu $m_{i,E}^{upw} = \frac{1}{2}(m_{i,K_1} + m_{i,K_2})$, kde K_1 a K_2 jsou elementy oddělené hranou E , a na vnějších hranách použijeme informace z okrajových podmínek.
6. Dokud $t_k < T$, opakujeme následující kroky:
 - 6.1. Aktualizujeme lokální matice $\mathbf{b}_{i,j,K}$, $\mathbf{R}_{K,F}$ a vektory \mathbf{R}_K pro všechna $i, j = 1, \dots, n$, $K \in \mathcal{K}_h$, $F \in \mathcal{E}_K$.

⁴Degrees of Freedom

- 6.2. Sestavíme lokální matice \mathbf{Q}_K a vypočteme inverze $\mathbf{Q}_K^{-1} \mathbf{R}_{K,F}$, $\mathbf{Q}_K^{-1} \mathbf{R}_K$ pro všechny $K \in \mathcal{K}_h$, $F \in \mathcal{E}_K$.
- 6.3. Sestavíme matici \mathbf{A} a vektor pravé strany \mathbf{b} v soustavě (3.34).
- 6.4. Řešíme soustavu (3.34) pro nN_E neznámých $Z_{i,E}^{k+1}$, $i = 1, \dots, n$, $E \in \mathcal{E}_h$.
- 6.5. Dle (3.30) vypočteme $Z_{i,K}^{k+1}$ pro všechna $i = 1, \dots, n$, $K \in \mathcal{K}_h$.
- 6.6. Aktualizujeme koeficienty $N_{i,j,K}$, $m_{i,K}$, $\mathbf{D}_{i,j,K}$, $w_{i,K,E}$, $f_{i,K}$ pro všechna $i, j = 1, \dots, n$, $K \in \mathcal{K}_h$, $E \in \mathcal{E}_K$.
- 6.7. Dle (3.25) aktualizujeme $m_{i,E}^{upw}$ pro všechna $i = 1, \dots, n$, $E \in \mathcal{E}_h$.
- 6.8. Přejdeme na následující časovou hladinu, $k \rightarrow k + 1$.

7. V čase $t_k = T$ získáme numerické řešení dané úlohy.

3.4 Výpočet prvků lokálních matic

Popis numerického schématu zakončíme podrobnostmi týkajícími se výpočtu prvků lokálních matic $\mathbf{B}_{i,j,K} = [B_{i,j,K,E,F}]_{E,F \in \mathcal{E}_K}$ a vektorů $\mathbf{W}_{i,K} = [w_{i,K,E}]_{E \in \mathcal{E}_K}$ definovaných pro všechna $i, j = 1, \dots, n$ a $K \in \mathcal{K}_h$ v sekci 3.1.2.

3.4.1 Výpočet matic $\mathbf{B}_{i,j,K}$ v \mathbb{R}^1

Uvažujme referenční element $K = [E_1, E_2] \subset \mathbb{R}^1$ o objemu $|K|_1 = E_2 - E_1 = h_x$, kde $E_1, E_2 \in \mathcal{E}_K = \mathcal{V}_K$. V jednorozměrném případě lze tenzor $\mathbf{D}_{i,j}$ reprezentovat jedinou skalární hodnotou $D_{i,j}$, přičemž střední hodnotu $D_{i,j}$ na elementu K označíme jako $D_{i,j,K}$. Přímým výpočtem definičních integrálů $B_{i,j,K,E,F} = D_{i,j,K}^{-1} \int_K \omega_{K,E}^T \omega_{K,F}$ pro $E, F \in \mathcal{E}_K$ dostaneme

$$\mathbf{B}_{i,j,K} = \frac{1}{6} D_{i,j,K}^{-1} h_x \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}. \quad (3.35)$$

Inverzní matice $\mathbf{b}_{i,j,K} = \mathbf{B}_{i,j,K}^{-1}$ má tvar

$$\mathbf{b}_{i,j,K} = \frac{D_{i,j,K}}{h_x} \begin{pmatrix} 4 & 2 \\ 2 & 4 \end{pmatrix}. \quad (3.36)$$

V praxi se však ukazuje (viz sekci 5.4), že tato volba koeficientů $b_{i,j,K,E,F}$ může v simulaci úloh v heterogenním prostředí způsobovat nežádoucí oscilace. Řešením je použít metodu *mass-lumping* [35] a integrály definující prvky matice $\mathbf{B}_{i,j,K}$ spočítat pouze přibližně pomocí numerické integrace

$$\int_K \xi(\mathbf{x}) d\mathbf{x} \approx \frac{|K|}{4} \sum_{i=1}^4 \xi(\mathbf{x}_i), \quad (3.37)$$

kde body \mathbf{x}_i umístíme do vrcholů elementu K . Takto získaná matice $\mathbf{B}_{i,j,K}^{(\ell)}$ je diagonální a má tvar

$$\mathbf{B}_{i,j,K} \approx \mathbf{B}_{i,j,K}^{(\ell)} = \frac{1}{2} D_{i,j,K}^{-1} h_x \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \quad (3.38)$$

Matici $\mathbf{b}_{i,j,K} = \mathbf{B}_{i,j,K}^{-1}$ pak aproximujeme inverzí matice $\mathbf{B}_{i,j,K}^{(\ell)}$, tedy

$$\mathbf{b}_{i,j,K} \approx 2 \frac{D_{i,j,K}}{h_x} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \quad (3.39)$$

3.4.2 Výpočet matic $\mathbf{B}_{i,j,K}$ v \mathbb{R}^2

Uvažujme referenční element $K = (0, h_x) \times (0, h_y) \subset \mathbb{R}^2$ o objemu $|K|_2 = h_x h_y$ a stejně jako v sekci 3.1.1 označíme symboly E_1, E_2, E_3, E_4 postupně levou, pravou, dolní a horní hranu elementu K .

Prvky matice $\mathbf{B}_{i,j,K}$ určíme pro případ, kdy tenzor $\mathbf{D}_{i,j} = D_{i,j} \mathbf{I}$ popisuje izotropní prostředí, přičemž střední hodnotu $D_{i,j}$ na elementu K označíme jako $D_{i,j,K}$. Přímým výpočtem definičních integrálů $B_{i,j,K,E,F} = D_{i,j,K}^{-1} \int_K \boldsymbol{\omega}_{K,E}^T \boldsymbol{\omega}_{K,F}$ pro $E, F \in \mathcal{E}_K$ dostaneme

$$\mathbf{B}_{i,j,K} = \frac{1}{6} D_{i,j,K}^{-1} \begin{pmatrix} 2\frac{h_x}{h_y} & -\frac{h_x}{h_y} & 0 & 0 \\ -\frac{h_x}{h_y} & 2\frac{h_x}{h_y} & 0 & 0 \\ 0 & 0 & 2\frac{h_y}{h_x} & -\frac{h_y}{h_x} \\ 0 & 0 & -\frac{h_y}{h_x} & 2\frac{h_y}{h_x} \end{pmatrix}. \quad (3.40)$$

Inverzní matice $\mathbf{b}_{i,j,K} = \mathbf{B}_{i,j,K}^{-1}$ má tvar

$$\mathbf{b}_{i,j,K} = D_{i,j,K} \begin{pmatrix} 4\frac{h_y}{h_x} & 2\frac{h_y}{h_x} & 0 & 0 \\ 2\frac{h_y}{h_x} & 4\frac{h_y}{h_x} & 0 & 0 \\ 0 & 0 & 4\frac{h_x}{h_y} & 2\frac{h_x}{h_y} \\ 0 & 0 & 2\frac{h_x}{h_y} & 4\frac{h_x}{h_y} \end{pmatrix}. \quad (3.41)$$

V článku [35] je ukázáno, že výsledná matice soustavy pro MHFEM není při použití obdélníkových elementů M-matice, což může způsobit oscilace v numerickém řešení. Proto stejně jako v \mathbb{R}^1 použijeme metodu mass-lumping a matici $\mathbf{B}_{i,j,K}$ aproximujeme pomocí matice $\mathbf{B}_{i,j,K}^{(\ell)}$, jejíž prvky spočítáme pomocí numerické integrace (3.37). Matice $\mathbf{B}_{i,j,K}^{(\ell)}$ je diagonální a má tvar

$$\mathbf{B}_{i,j,K} \approx \mathbf{B}_{i,j,K}^{(\ell)} = \frac{1}{2} D_{i,j,K}^{-1} \begin{pmatrix} \frac{h_x}{h_y} & 0 & 0 & 0 \\ 0 & \frac{h_x}{h_y} & 0 & 0 \\ 0 & 0 & \frac{h_y}{h_x} & 0 \\ 0 & 0 & 0 & \frac{h_y}{h_x} \end{pmatrix}. \quad (3.42)$$

Matici $\mathbf{b}_{i,j,K} = \mathbf{B}_{i,j,K}^{-1}$ pak aproximujeme inverzí matice $\mathbf{B}_{i,j,K}^{(\ell)}$, tedy

$$\mathbf{b}_{i,j,K} \approx 2D_{i,j,K} \begin{pmatrix} \frac{h_y}{h_x} & 0 & 0 & 0 \\ 0 & \frac{h_y}{h_x} & 0 & 0 \\ 0 & 0 & \frac{h_x}{h_y} & 0 \\ 0 & 0 & 0 & \frac{h_x}{h_y} \end{pmatrix}. \quad (3.43)$$

3.4.3 Projekce vektorů $w_{i,K,E}$ do prostoru $\mathbf{RT}_0(K)$

Pro výpočet koeficientů $w_{i,K,E}$ vynásobíme rovnici (3.18) bazickou funkcí $\boldsymbol{\omega}_{K,F} \in \mathbf{RT}_0(K)$ a integrujeme přes element K :

$$\int_K \boldsymbol{\omega}_{K,F}^T w_i|_K = \sum_{E \in \mathcal{E}_K} w_{i,K,E} \int_K \boldsymbol{\omega}_{K,F}^T \boldsymbol{\omega}_{K,E}. \quad (3.44)$$

Označme integrál na pravé straně rovnice (3.44) jako $B_{K,E,F} = \int_K \boldsymbol{\omega}_{K,F}^T \boldsymbol{\omega}_{K,E}$ a lokální matici $\mathbf{B}_K = [B_{K,E,F}]_{E,F \in \mathcal{E}_K}$. Všimněme si, že platí $\mathbf{B}_{i,j,K} = \mathbf{B}_K$, pokud $\mathbf{D}_{i,j}|_K = \mathbf{I}$, a označme inverzi matice \mathbf{B}_K jako $\mathbf{b}_K = [b_{K,E,F}]_{E,F \in \mathcal{E}_K}$. Inverzí vztahu (3.44) dostaneme

$$w_{i,K,E} = \sum_{F \in \mathcal{E}_K} b_{K,E,F} G_{K,F}, \quad (3.45)$$

kde $G_{K,F} = \int_K \boldsymbol{\omega}_{K,F}^T \mathbf{w}_i|_K$. Při označení $\mathbf{G}_K = [G_{K,F}]_{F \in \mathcal{E}_K}$ a $\mathbf{W}_{i,K} = [w_{i,K,E}]_{E \in \mathcal{E}_K}$ můžeme rovnost (3.45) zapsat ve vektorovém tvaru

$$\mathbf{W}_{i,K} = \mathbf{b}_K \mathbf{G}_K. \quad (3.46)$$

Složky vektorů \mathbf{G}_K vypočteme pro případ, kdy $\mathbf{w}_i|_K \equiv \mathbf{w}_{i,K}$ je konstantní na elementu $K \in \mathcal{K}_h$. Podobně jako v předchozích sekcích uvažujeme nejprve referenční element $K = [E_1, E_2] \subset \mathbb{R}^1$ o objemu $|K|_1 = E_2 - E_1 = h_x$, kde $E_1, E_2 \in \mathcal{E}_K = \mathcal{V}_K$. Výpočtem definičních integrálů dostaneme

$$\mathbf{G}_K = \frac{1}{2} h_x (\mathbf{w}_{i,K})_x \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \quad (3.47)$$

kde $(\mathbf{w}_{i,K})_x$ označuje první složku vektoru $\mathbf{w}_{i,K}$.

Dále uvažujme referenční element $K = (0, h_x) \times (0, h_y) \subset \mathbb{R}^2$ o objemu $|K|_2 = h_x h_y$ a stejně jako v předchozí sekci označíme symboly E_1, E_2, E_3, E_4 postupně levou, pravou, dolní a horní hranu elementu K . Výpočtem definičních integrálů dostaneme

$$\mathbf{G}_K = \frac{1}{2} \begin{pmatrix} -h_x (\mathbf{w}_{i,K})_x \\ h_x (\mathbf{w}_{i,K})_x \\ -h_y (\mathbf{w}_{i,K})_y \\ h_y (\mathbf{w}_{i,K})_y \end{pmatrix}, \quad (3.48)$$

kde $(\mathbf{w}_{i,K})_x$ a $(\mathbf{w}_{i,K})_y$ označují první resp. druhou složku vektoru $\mathbf{w}_{i,K}$.

Kapitola 4

Paralelní implementace na GPU

V této kapitole popíšeme paralelní implementaci numerického schématu pro řešení úlohy (2.1) popsaného v předchozí kapitole. Nejprve je popsána paralelizace algoritmu pro výpočet numerického řešení popsaného v sekci 3.3, včetně paralelizace algoritmu pro řešení soustav lineárních algebraických rovnic. Druhá část kapitoly se již věnuje popisu implementace masivně paralelního řešiče na GPU s využitím platformy CUDA [27] a šablonové knihovny TNL¹, kterou vyvíjíme na Katedře matematiky FJFI pod vedením Ing. Tomáše Oberhubera, Ph.D.

4.1 Paralelizace numerického schématu

Při paralelizaci algoritmu pro výpočet řešení dané úlohy je nejprve nutné rozdělit úlohu na co největší počet malých, na sobě nezávislých kroků, které lze zpracovat současně. Kroky algoritmu popsaného v sekci 3.3 jsou na sobě závislé a je tedy nutné je provést v uvedeném pořadí, u většiny ale můžeme využít nezávislosti skrytých elementárních kroků.

První tři kroky představují inicializaci, kterou provedeme sériovým zpracováním v konstantním čase. Inicializace jednotlivých koeficientů v krocích 4. a 5. jsou již nezávislé pro různé hodnoty uvedených indexů i, j, K, E . Hlavním problémem je paralelizace časové smyčky v kroku 6.. Kroky 6.1. až 6.8. je nutné provádět v uvedeném pořadí, přechody na následující krok tedy představují hlavní synchronizační bariéry. V krocích 6.1., 6.5., 6.6. a 6.7. probíhají výpočty dle explicitních vztahů, které jsou navzájem nezávislé pro různé hodnoty uvedených indexů i, j, K, E , a proto je snadné tyto kroky paralelizovat. Paralelizace zbývajících kroků 6.2., 6.3. a 6.4. je komplikovanější, popíšeme je tedy podrobněji.

4.1.1 Lokální inverze

Krok 6.2. sestává z výpočtu prvků lokálních matic \mathbf{Q}_K a následného výpočtu inverzí $\mathbf{Q}_K^{-1}\mathbf{R}_{K,F}$ a $\mathbf{Q}_K^{-1}\mathbf{R}_K$ pro všechny elementy $K \in \mathcal{K}_h$ a hrany $F \in \mathcal{E}_K$. Prvky všech matic \mathbf{Q}_K můžeme vypočítat navzájem nezávisle, ale pro výpočet inverzí musíme pracovat s maticí \mathbf{Q}_K jako celkem. Tyto dvě fáze můžeme oddělit globální synchronizační bariérou, nejprve masivně paralelně určit prvky matic \mathbf{Q}_K a po synchronizaci vlákna příslušná stejnému elementu sloučit a vypočítat inverze $\mathbf{Q}_K^{-1}\mathbf{R}_{K,F}$ a $\mathbf{Q}_K^{-1}\mathbf{R}_K$. V praxi je však výhodnější využít toho, že matice \mathbf{Q}_K mají malé rozměry (v případě dvoufázového proudění pouze 2×2) a navíc je nebudeme dále potřebovat, což nám umožní vyhnout se pomalému zápisu matic do globální paměti. Proto hned od začátku vytvoříme pouze jedno vlákno na každý element a v každém vlákně sériově sestavíme lokální matici \mathbf{Q}_K uloženou v registrech a vypočteme inverze. Výpočet inverzí $\mathbf{Q}_K^{-1}\mathbf{R}_{K,F}$, $\mathbf{Q}_K^{-1}\mathbf{R}_K$ je ekvivalentní řešení soustavy s maticí \mathbf{Q}_K a několika pravými stranami, což provedeme pomocí LU rozkladu matice \mathbf{Q}_K .

¹Template Numerical Library

4.1.2 Sestavení globální matice

Matice soustavy (3.34) je řídká a její rozměry jsou typicky velké, proto je pro její reprezentaci potřeba použít speciální formát. V této práci použijeme formát *sliced-ellpack* [25], jehož implementace optimalizovaná pro GPU je dostupná v knihovně TNL.

Pro sestavení matice \mathbf{A} v soustavě (3.34) jsou možné dva přístupy. První možnost je procházet síť po elementech, na každém elementu napočítat všechny příslušné členy a přičíst je k prvkům matice. Tento postup ale není vhodný pro paralelní implementaci, protože vlákna zpracovávající dva sousední elementy K_1, K_2 by zapisovala do stejné oblasti v paměti příslušející společné hraně $E \in \mathcal{E}_{K_1} \cap \mathcal{E}_{K_2}$. Proto použijeme druhý přístup, kde síť procházíme po hranách. Při tom využíváme toho, že každý řádek matice má pevně daný počet nenulových prvků, při použití obdélníkových elementů mohou nastat tyto možnosti:

- Řádek příslušející vnitřní hraně $E \in \mathcal{E}_h^{int}$ a i -té rovnici soustavy (2.1) má $7n$ nenulových prvků, pro každé $j = 1, \dots, n$ počítáme příspěvky hran elementů oddělených hranou E .
- Řádek příslušející okrajové hraně $E \in \mathcal{E}_h^{ext}$ a i -té rovnici soustavy (2.1), pro kterou je na hraně E předepsaná Neumannova podmínka, má $4n$ nenulových prvků.
- Řádek příslušející okrajové hraně $E \in \mathcal{E}_h^{ext}$ a i -té rovnici soustavy (2.1), pro kterou je na hraně E předepsaná Dirichletova podmínka, má pouze 1 nenulový prvek.

Dále je vhodné využít toho, že řádky příslušející hraně $E \in \mathcal{E}_h$ a různým rovnicím soustavy (2.1) lze zpracovat nezávisle. Matici tedy přesněji řečeno neplníme po hranách numerické sítě, ale po řádcích.

4.1.3 Řešení soustav lineárních algebraických rovnic s řídkou maticí

Řešení soustav lineárních algebraických rovnic s řídkou maticí $\mathbf{A} \in \mathbb{R}^{N,N}$ v kroku 6.4. představuje z hlediska implementace nejnáročnější část algoritmu. Přímé metody lze obecně velmi těžko paralelizovat, proto spoléháme na iterační metody. V této práci použijeme metodu GMRES² [28, 29], což je jedna z nejpoužívanějších iteračních metod pro řešení soustav s obecnou maticí.

Metoda GMRES je založena na Arnoldiho algoritmu pro konstrukci ortonormální báze Krylovova podprostoru $\mathcal{K}_m = \text{span}\{\mathbf{v}_1, \mathbf{A}\mathbf{v}_1, \dots, \mathbf{A}^{m-1}\mathbf{v}_1\}$, jehož standardní varianta popsaná v [28, str. 148] vypadá následovně:

Algoritmus 2

1. Zvol $\mathbf{v}_1 \in \mathbb{R}^N$ tak, že $\|\mathbf{v}_1\| = 1$.
2. Pro $i = 1, \dots, m$:
 - 2.1. $\mathbf{w}_i := \mathbf{A}\mathbf{v}_i$
 - 2.2. Pro $k = 1, \dots, i$:
 - 2.2.1. $h_{ki} = \mathbf{w}_i^T \mathbf{v}_k$
 - 2.2.2. $\mathbf{w}_i := \mathbf{w}_i - h_{ki}\mathbf{v}_k$
 - 2.3. $h_{i+1,i} = \|\mathbf{w}_i\|$. Pokud $h_{i+1,i} = 0$, skonči.
 - 2.4. $\mathbf{v}_{i+1} = \frac{1}{h_{i+1,i}}\mathbf{w}_i$

Algoritmus 2 využívá v kroku 2.2. modifikovaný Gramův–Schmidtův ortogonalizační proces, což je ze své podstaty čistě sekvenční algoritmus, jelikož pořadí prováděných kroků zajišťuje numerickou stabilitu. V praxi ani přesnost algoritmu 2 nemusí být dostatečná a je potřeba provést reortogonalizaci

²Generalized Minimal Residual method

opakováním kroku 2.2. dvakrát [18]. Algoritmus 2 však lze paralelizovat implicitně použitím paralelních algoritmů pro výpočet základních operací lineární algebry, jako např. součet vektorů nebo skalární součin.

Alternativním řešením je použití jiného ortogonalizačního procesu, který by mohl být vhodnější pro paralelizaci. Nahrazení modifikovaného Gramova–Schmidtova procesu v algoritmu 2 ortogonalizací pomocí Householderových transformací bylo poprvé navrženo v článku [32]. Touto modifikací získáme algoritmus 3, který opět přebíráme z [28, str. 149]:

Algoritmus 3

1. Zvol nenulový vektor $\mathbf{z}_1 \in \mathbb{R}^N$.
2. Pro $i = 1, \dots, m + 1$:
 - 2.1. Urči Householderův vektor $\mathbf{y}_i \in \mathbb{R}^N$ takový, že

$$(\mathbf{y}_i)_j = 0 \text{ pro } j = 1, \dots, i - 1 \text{ a } (\mathbf{P}_i \mathbf{z}_i)_j = 0 \text{ pro } j = i + 1, \dots, N, \text{ kde } \mathbf{P}_i = \mathbf{I} - t_i \mathbf{y}_i \mathbf{y}_i^T,$$

$$t_i = \frac{2}{\|\mathbf{y}_i\|^2}.$$
 - 2.2. $\mathbf{h}_{i-1} = [(\mathbf{P}_i \mathbf{z}_i)_j]_{j=1}^{m+1}$
 - 2.3. $\mathbf{v}_i = \mathbf{P}_1 \dots \mathbf{P}_i \mathbf{e}_i$
 - 2.4. Pokud $i \leq m$, vypočti $\mathbf{z}_{i+1} = \mathbf{P}_i \dots \mathbf{P}_1 \mathbf{A} \mathbf{v}_i$.

Z hlediska paralelizace jsme si zatím nijak nepomohli, jelikož Householderovy transformace v krocích 2.3. a 2.4. algoritmu 3 je třeba aplikovat sekvenčně. Můžeme však využít kompaktní WY reprezentaci [30] součinu Householderových matic, která byla v nedávné době aplikována v paralelních algoritmech pro moderní hardware [22, 34]. Podobná technika byla navržena také v [32].

Pro modifikaci algoritmu 3 pomocí kompaktní WY reprezentace označme $\mathbf{Y}_i = [\mathbf{y}_1, \dots, \mathbf{y}_i] \in \mathbb{R}^{N,i}$, $\mathbf{T}_1 = t_1 \in \mathbb{R}^{1,1}$ a rekurzivně definujme horní trojúhelníkovou matici $\mathbf{T}_i \in \mathbb{R}^{i,i}$,

$$\mathbf{T}_i = \begin{pmatrix} \mathbf{T}_{i-1} & -t_i \mathbf{T}_{i-1} \mathbf{Y}_{i-1} \mathbf{y}_i^T \\ 0 & t_i \end{pmatrix}. \quad (4.1)$$

Součiny Householderových matic $\mathbf{P}_1 \dots \mathbf{P}_i$ a $\mathbf{P}_i \dots \mathbf{P}_1$ lze vyjádřit pomocí matic \mathbf{Y}_i a \mathbf{T}_i následujícím způsobem:

$$\mathbf{P}_1 \dots \mathbf{P}_i = \mathbf{I} - \mathbf{Y}_i \mathbf{T}_i \mathbf{Y}_i^T, \quad (4.2a)$$

$$\mathbf{P}_i \dots \mathbf{P}_1 = \mathbf{I} - \mathbf{Y}_i \mathbf{T}_i^T \mathbf{Y}_i^T. \quad (4.2b)$$

Tím dostáváme následující algoritmus:

Algoritmus 4

1. Zvol nenulový vektor $\mathbf{z}_1 \in \mathbb{R}^N$.
2. Pro $i = 1, \dots, m + 1$:
 - 2.1. Pro \mathbf{z}_i vypočti \mathbf{y}_i a t_i stejně jako v algoritmu 3.
 - 2.2. Aktualizuj matice \mathbf{Y}_i a \mathbf{T}_i pomocí t_i , \mathbf{y}_i , \mathbf{T}_{i-1} a \mathbf{Y}_{i-1} .
 - 2.3. $\mathbf{h}_{i-1} = [(\mathbf{P}_i \mathbf{z}_i)_j]_{j=1}^{m+1}$
 - 2.4. $\mathbf{v}_i = (\mathbf{I} - \mathbf{Y}_i \mathbf{T}_i \mathbf{Y}_i^T) \mathbf{e}_i$
 - 2.5. Pokud $i \leq m$, vypočti $\mathbf{z}_{i+1} = (\mathbf{I} - \mathbf{Y}_i \mathbf{T}_i^T \mathbf{Y}_i^T) \mathbf{A} \mathbf{v}_i$.

Výhoda algoritmu 4 oproti algoritmu 3 spočívá v nahrazení sekvenční aplikace Householderových transformací výpočtem pevného počtu součinů plné matice a vektoru, což je operace, kterou lze dobře paralelizovat. Podstatné vlastnosti uvedených algoritmů jsou uvedeny v následující tabulce s údaji převzatými z [22, 28]:

	MGS	HO	CWY
Počet operací	$O(2m^2N)$	$O(4m^2N)$	$O(4m^2N)$
Počet synchronizací	$O(m^2)$	$O(m^2)$	$O(m)$
Paměťové nároky	$O((m+1)N)$	$O(2(m+1)N)$	$O(2(m+1)N)$
Ztráta ortogonalit	$O(\epsilon\kappa(\mathbf{A}))$	$O(\epsilon)$	$O(\epsilon)$

Zde v MGS označuje ortogonalizaci pomocí modifikovaného Gramova–Schmidtova procesu, HO označuje Householderovy transformace a CWY označuje kompaktní WY reprezentaci. V posledním řádku tabulky ϵ označuje strojovou přesnost a $\kappa(\mathbf{A})$ označuje číslo podmíněnosti matice \mathbf{A} . Z hlediska paměťové náročnosti ortogonalizace pomocí Householderových transformací a kompaktní WY reprezentace ve skutečnosti není potřeba ukládat vektory \mathbf{v}_i , které lze dopočítat dle [28, str. 159], čímž dostaneme náročnost $O((m+1)N)$. To je však také čistě sekvenční algoritmus, navíc v praxi je škoda nevyužít dostatek operační paměti a globální paměti GPU.

V praxi provádíme výpočty v krocích 2.4. a 2.5. algoritmu 4 „zprava doleva“, aby se co nejrychleji redukoval objem zpracovávaných dat. Při výpočtu na GPU rozlišujeme výpočty vyžadující $O(N)$ operací, které provádíme na GPU, a výpočty vyžadující $O(m)$ nebo $O(m^2)$ operací, které provádíme sekvenčně na CPU. Aby to bylo možné, ukládáme matice \mathbf{Y}_i v globální paměti GPU a matice \mathbf{T}_i v operační paměti systému. V krocích 2.3. a 2.4. se nám však hodí přistupovat k prvkům matice \mathbf{Y}_i při výpočtech na CPU, proto pro umožnění rychlého přístupu duplikujeme prvních $m+1$ řádků matic \mathbf{Y}_i také v operační paměti.

Konvergenci iteračních metod pro řešení soustav lineárních algebraických rovnic je obecně možné urychlit použitím vhodného předpokládání. Vývoj předpokládání pro paralelní architektury je však velmi komplikovaná úloha. V rámci knihovny TNL máme k dispozici pouze Jacobiho předpokládání, pokročilejší metody budou předmětem dalšího výzkumu.

4.2 Detaily implementace

Paralelní řešič pro uvedenou úlohu je implementován v jazyce C++ s využitím platformy CUDA, která umožňuje využít moderní GPU pro obecné výpočty. Mapování vláken na výpočetní jednotky a správa paměti jsou v CUDA abstrahovány pomocí rozšíření jazyka C++, takže z programátorského hlediska je programování pro GPU intuitivní a jednoduché. Vytvořit efektivní kód je však podstatně složitější, optimalizace zpravidla využívají podrobných znalostí dané hardwarové architektury. Oproti vícejádrovým CPU je vývoj optimálního kódu pro GPU komplikovanější zejména z hlediska efektivního využití hierarchie pamětí a dobrého rozprostření řešené úlohy mezi tisíce výpočetních jednotek, které jsou oproti moderním sofistikovaným CPU podstatně primitivnější, ale jako celek poskytují mnohonásobně vyšší výpočetní výkon.

Pro porovnání efektivity paralelního řešiče je potřeba mít k dispozici také sekvenční řešič pro stejnou úlohu. Hlavní výhodou použité knihovny TNL je poskytnutí jednotného rozhraní pro vývoj sekvenčních a paralelních algoritmů, takže ve výsledném programu lze v závislosti na zvoleném parametru spustit výpočet buď na CPU nebo na GPU. Pro výběr algoritmů jsou využity šablony jazyka C++, čímž odpadá dodatečná režie za běhu programu. Z knihovny TNL kromě tohoto rozhraní využíváme také zejména numerické sítě, formát sliced-ellpack pro reprezentaci řídkých matic a iterační řešič GMRES.

Kapitola 5

Numerické výsledky

Poslední kapitola této práce se zabývá výsledky numerických simulací. Hlavním cílem této kapitoly je ověřit konvergenci numerického schématu pro řešení soustavy (2.1) popsaného v kapitole 3 a porovnat efektivitu paralelního výpočtu na GPU vůči sekvenčnímu výpočtu na CPU.

Nejprve jsou popsány dvě difúzní úlohy se známými semi-analytickými řešeními, která jsou použita pro ověření konvergence numerického schématu pomocí experimentální analýzy konvergence. Dále se věnujeme advekčně-difúzním úlohám v heterogenním prostředí a simulaci tzv. bariérového efektu, nejprve v 1D a poté ve 2D, kde již hraje významnou roli umístění překážky v oblasti. Poslední podkapitola se věnuje analýze efektivity výpočtu a porovnání paralelního řešiče vzhledem k jeho základní sériové verzi.

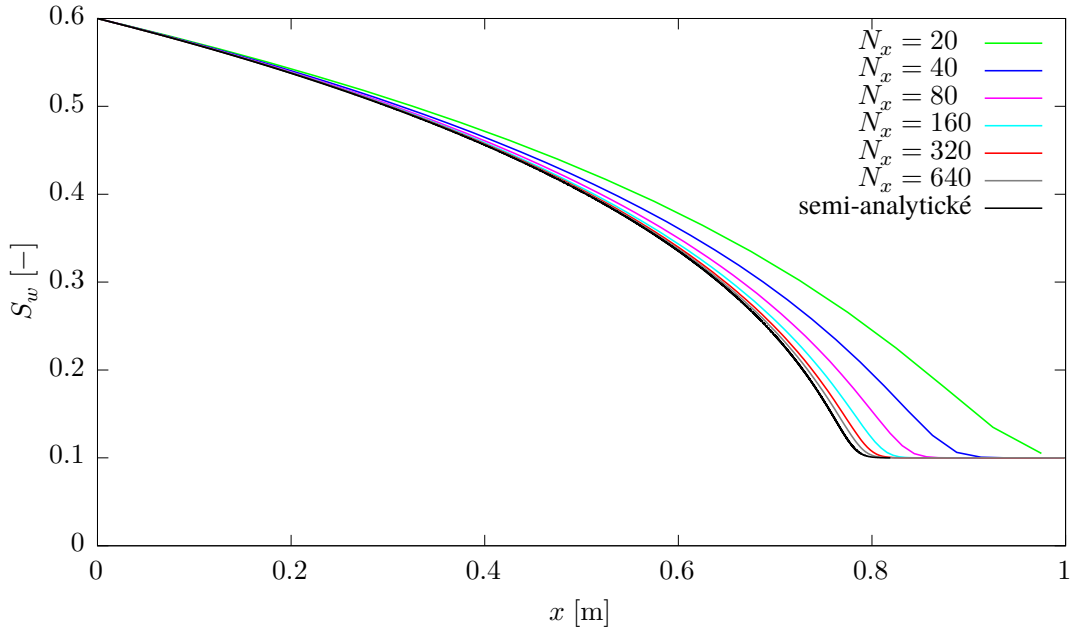
Výsledky prezentované v této kapitole byly získány pomocí vlastního řešiče popsaného v kapitole 4, referenční řešení byla převzata z citovaných zdrojů. Pro zpracování výsledků a vizualizaci byly použity programy GNU Parallel [31] a Gnuplot [33].

5.1 McWhorterova–Sunadova úloha

McWhorterova–Sunadova úloha představuje jednorozměrnou čistě difúzní úlohu v homogenním prostředí, pro kterou je známé semi-analytické řešení [12, 15, 16, 24]. V jednorozměrném případě zkoumáme oblast $\Omega = (0, +\infty)$ vyplněnou porézním materiálem s homogenními vlastnostmi a nestlačitelnými fázemi se zadanou počáteční saturací smáčivé fáze $S_w(x, 0) = S_{ini}$ v celé oblasti. Na vstupní hraně ($x = 0$) předepíšeme Dirichletovu podmínku pro obě fáze, pro smáčivou fázi předepíšeme saturaci $S_w(0, t) = S_0 > S_{ini}$ a pro nesmáčivou fázi tlak $p_n(0, t) = p_{n,0}$. Případ $S_0 > S_{ini}$ odpovídá vtlačení smáčivé fáze do oblasti zaplněné (převážně) nesmáčivou fází. Díky nestlačitelnosti je však role obou fází stejná až na materiálové konstanty, pro opačný případ $S_0 < S_{ini}$ tedy dostaneme ekvivalentní úlohu odpovídající vtlačení nesmáčivé fáze. V této úloze neuvažujeme vliv gravitace a pro výpočet numerického i semi-analytického řešení použijeme model kapilarity podle Brookse a Coreya (1.12).

Pro numerické řešení této úlohy se omezíme na omezenou oblast Ω a na pravé straně oblasti předepíšeme nulovou Neumannovu podmínku pro obě fáze. Pro porovnání se semi-analytickým řešením na polonekonečné oblasti je potřeba zvolit konečný čas simulace tak, aby vtlačená fáze v průběhu simulace nedorazila na druhý konec oblasti. Úlohu lze dále rozšířit do vyšších dimenzí, např. v \mathbb{R}^2 uvažujeme obdélníkovou oblast s jednou vstupní stranou, na které předepíšeme Dirichletovy podmínky a na zbývajících stranách předepíšeme nulové Neumannovy podmínky.

Úlohu řešíme nejprve v \mathbb{R}^1 na oblasti $\Omega = (0, L_x)$, kde $L_x = 1$ m. Oblast je vyplněna pískem C, jehož parametry jsou uvedeny v tab. A.1. Zvolíme hodnoty okrajové saturace $S_0 = 0,6$, počáteční saturace $S_{ini} = 0,1$ a referenční tlak $p_{n,0} = 10^5$ Pa. Jako smáčivou fázi uvažujeme vodu ($\rho_w = 1000 \text{ kg m}^{-3}$, $\mu_w = 8,9 \cdot 10^{-4} \text{ kg m}^{-1} \text{ s}^{-1}$) a jako nesmáčivou fázi vzduch s parametry $\rho_n = 1,2047 \text{ kg m}^{-3}$, $\mu_n = 1,87 \cdot 10^{-5} \text{ kg m}^{-1} \text{ s}^{-1}$. Porovnání semi-analytického řešení [11] s numerickými řešeními na různých sítích v čase $T = 60\,000$ s je zobrazeno na obr. 5.1. Dále řešíme úlohu rozšířenou do \mathbb{R}^2 na



Obrázek 5.1: Numerické řešení McWhorterovy–Sunadovy úlohy v 1D na sítích o N_x elementech v čase $T = 6 \cdot 10^4$ s, časový krok je volen $\tau = 250$ s na nejhrubší síti $N_x = 20$ a pro jemnější síť $\tau \propto h^2$.

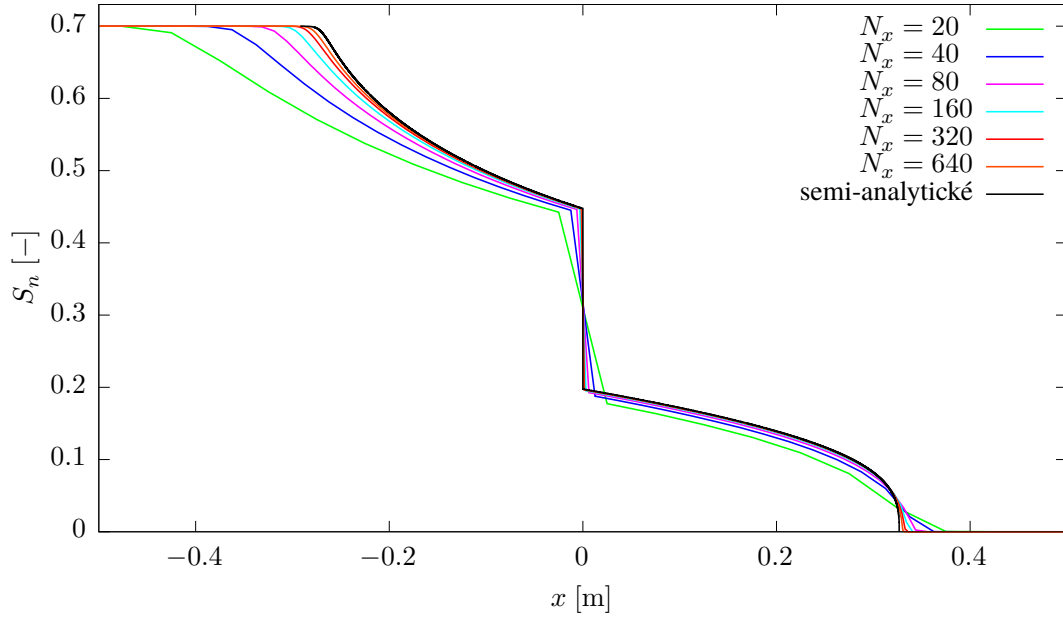
oblasti $\Omega = (0, L_x) \times (0, L_y) \subset \mathbb{R}^2$, kde $L_x = 1$ m a $L_y = 0,25$ m. Numerické řešení rozšířené úlohy je dle očekávání konstantní podél osy y a řez ve směru osy x odpovídá řešení jednorozměrné úlohy.

5.2 Difúzní úloha v heterogenním prostředí

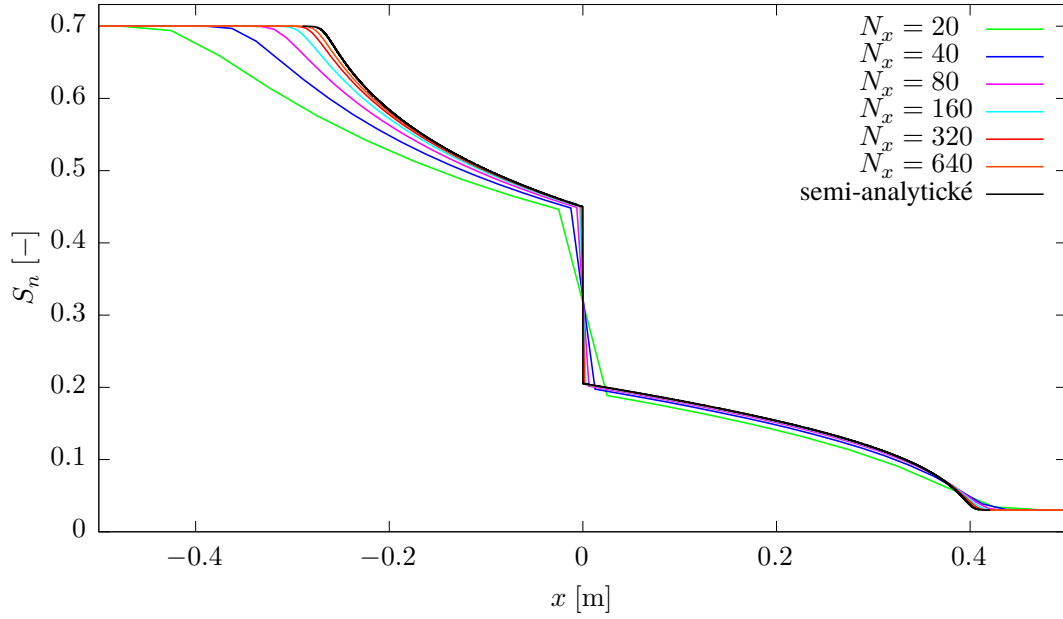
Druhou testovací úlohou je difúzní úloha v heterogenním prostředí s jediným materiálovým rozhraním. Oblast Ω je složena ze dvou homogenních podoblastí oddělených rozhraním v $x = 0$, levá oblast $\Omega_1 = (L_1, 0)$ je vyplněna pískem A a pravá oblast $\Omega_2 = (0, L_2)$ je vyplněna pískem B. Vlastnosti materiálů jsou uvedeny v tab. A.1; písek B je jemnější než písek A. Jako smáčivou fázi uvažujeme vodu ($\rho_w = 1000 \text{ kg m}^{-3}$, $\mu_w = 8,9 \cdot 10^{-4} \text{ kg m}^{-1} \text{ s}^{-1}$) a jako nesmáčivou fázi vzduch s parametry $\rho_n = 1,2047 \text{ kg m}^{-3}$, $\mu_n = 18,7 \cdot 10^{-6} \text{ kg m}^{-1} \text{ s}^{-1}$. Stejně jako v předchozí úloze i zde neuvažujeme vliv gravitace a pro výpočet numerického i semi-analytického řešení použijeme model kapilarity podle Brookse a Coreyho (1.12).

Pro numerické řešení úlohy zvolíme následující počáteční a okrajové podmínky. Zvolíme referenční tlak pro smáčivou fázi $p_w(\mathbf{x}, 0) = 10^5$ Pa, v čase $t = 0$ konstantní v Ω , a tlak nesmáčivé fáze dopočítáme dle zadané saturace. V jednotlivých podoblastech předepíšeme počáteční saturace $S_w(\mathbf{x}, 0) = S_1$ na Ω_1 a $S_w(\mathbf{x}, 0) = S_2$ na Ω_2 . Na hranici $x = L_1$ pro obě fáze předepíšeme Dirichletovu podmínku souhlasnou s počáteční podmínkou, $p_w^D(L_1, t) = 10^5$ Pa a $S_w^D(L_1, t) = S_1$, na protější hranici $x = L_2$ předepíšeme nulovou Neumannovu podmínku $v_w^N(L_2, t) = v_n^N(L_2, t) = 0 \text{ m s}^{-1}$. Pro toto nastavení úlohy je známo semi-analytické řešení, které je popsáno v [8, 16]. Podobně jako v případě McWhorterovy–Sunadovy úlohy je i tato úloha rozšiřitelná do vyšších dimenzí.

Řešíme dvě varianty této úlohy lišící se počáteční podmínkou. V první variantě volíme $S_1 = 0,3$ a $S_2 = 1$, ve druhé variantě $S_1 = 0,3$ a $S_2 = 0,97$. Rozměry oblasti volíme dle ostatních parametrů symetricky, $L_2 = -L_1 = 0,5$ m. Porovnání semi-analytického řešení [10] s numerickými řešeními na různých sítích v čase $T = 10000$ s je zobrazeno na obr. 5.2.



(a) Počáteční podmínka $S_n(\mathbf{x}, 0) = 0,7$ v levé podoblasti a $S_n(\mathbf{x}, 0) = 0$ v pravé podoblasti.



(b) Počáteční podmínka $S_n(\mathbf{x}, 0) = 0,7$ v levé podoblasti a $S_n(\mathbf{x}, 0) = 0,03$ v pravé podoblasti.

Obrázek 5.2: Numerické řešení difúzní úlohy v heterogenním prostředí v 1D na sítích o N_x elementech v čase $T = 10^4$ s, časový krok je volen $\tau = 250$ s na nejhrubší síti $N_x = 20$ a pro jemnější síť $\tau \propto h^2$.

5.3 Numerická analýza difúzních úloh

5.3.1 Experimentální analýza konvergence

Pro srovnání numerických řešení s příslušným semi-analytickým řešením použijeme experimentální analýzu řádu konvergence (EOC¹). Chybu numerického řešení E_h na síti \mathcal{K}_h definujeme pomocí saturace S_w , případně S_n jako rozdíl mezi přesným řešením S_α a numerickým řešením $S_{\alpha,h}$

$$E_h = S_w - S_{w,h} = S_n - S_{n,h}. \quad (5.1)$$

Pro analýzu řádu konvergence volíme v celé simulaci konstantní časový krok $\tau = t_k - t_{k-1}$. Přesnost časové diskretizace (3.24) je prvního řádu, dále zkoumáme řád přesnosti prostorové diskretizace v normách L^1 , L^2 a L^∞ . Předpokládáme, že L^ν norma chyby E_h má tvar

$$\|E_h\|_\nu = C_1\tau + C_2h^{\alpha_\nu}, \quad (5.2)$$

kde C_1 a C_2 jsou konstanty. Řád konvergence α_ν lze určit experimentálně pomocí analýzy numerických řešení na sítích s různými parametry h , přičemž pro omezení chyby časové diskretizace zmenšujeme časový krok úměrně jisté mocnině parametru sítě h , $\tau \propto h^\beta$. Pro $\beta < \alpha_\nu$ bude při zjemňování sítě chyba numerického řešení $\|E_h\|_\nu$ dána převážně chybou časové diskretizace, pro $\beta > \alpha_\nu$ naopak chybou prostorové diskretizace. Pro chyby numerických řešení $\|E_{h_1}\|_\nu$, $\|E_{h_2}\|_\nu$ na sítích s parametry h_1 resp. h_2 definujeme experimentální řád konvergence v normě L^ν pomocí

$$\text{EOC}_\nu = \frac{\ln\|E_{h_1}\|_\nu - \ln\|E_{h_2}\|_\nu}{\ln h_1 - \ln h_2}. \quad (5.3)$$

Při určování chyby numerického řešení E_h interpolujeme numerické řešení na síť semi-analytického řešení, která je mnohem jemnější oproti síti numerického řešení.

Numerická řešení úloh popsaných v sekcích 5.1 a 5.2 spočítáme v 1D na sítích o rozměrech $N_x = 20$, $N_x = 40$, $N_x = 80$, $N_x = 160$, $N_x = 320$ a $N_x = 640$ elementů a ve 2D na sítích s $N_x \times N_y$ elementy, přičemž konkrétní hodnoty N_x jsou stejné jako v 1D a platí $N_x = 4N_y$. V obou úlohách jsou tedy pro danou síť rozměry elementů $h = h_x = h_y = \frac{1}{N_x}$. Časový krok volíme v obou úlohách stejně, $\tau = 250$ s na nejhrubší síti $N_x = 20$ a při každém přechodu na jemnější síť je časový krok $4 \times$ menší oproti předchozí síti, tedy $\tau \propto h^2$. EOC a chyby numerických řešení úlohy 5.1 jsou uvedeny v tab. 5.1 a 5.2 a pro úlohu 5.2 v tab. 5.3 a 5.4.

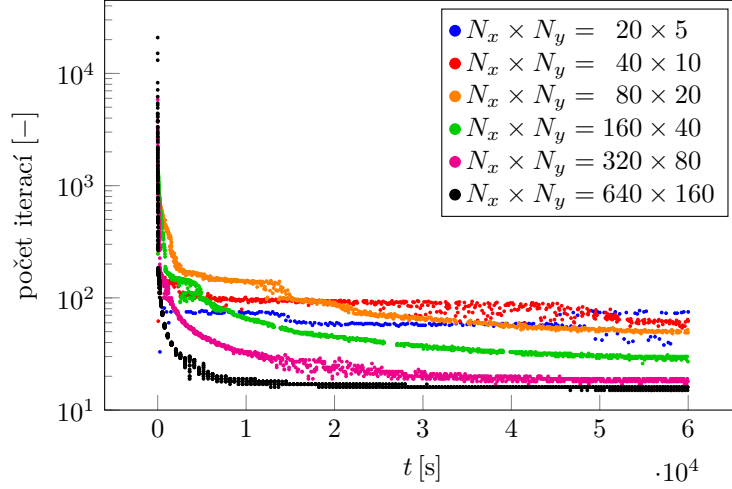
Soustavy lineárních rovnic jsou řešeny pomocí iterační metody GMRES(m) s restartováním po m iteracích a s využitím Jacobiho předpodmínění. Výsledky tedy závisí na volbě parametru m a na zastavovacím kritériu. Při řešení soustavy $\mathbf{A}\mathbf{x} = \mathbf{b}$ pomocí iterační metody s předpodmíněním \mathbf{M} konstruujeme posloupnost vektorů $\mathbf{x}^{(k)}$ konvergující k přesnému řešení \mathbf{x} . Metoda končí, pokud

$$\frac{\|\mathbf{M}^{-1}(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b})\|}{\|\mathbf{M}^{-1}\mathbf{b}\|} < \varepsilon, \quad (5.4)$$

kde ε je zvolený parametr. Pro dosažení uvedených výsledků byla na základě experimentální analýzy závislosti chyby E_h na zastavovacím kritériu ε zvolena hodnota $\varepsilon = 10^{-11}$.

Pro volbu parametru m byla použita následující úvaha. Dle grafu počtu iterací v průběhu simulace na obr. 5.3 je patrné, že z hlediska konvergence metody je nejproblematictější začátek simulace, kdy v prvních několika krocích potřebuje GMRES ke konvergenci tisíce iterací. To je způsobeno nespojitým

¹Experimental Order of Convergence



Obrázek 5.3: Počet iterací potřebných pro konvergenci metody $\text{GMRES}(m)$ v průběhu výpočtu McWhorterovy–Sunadovy úlohy ve 2D na síti s $N_x \times N_y$ elementy. Restartovací parametr $m = 100$.

počátečním nastavením úlohy, kdy iterační řešič musí překonat velký rozdíl mezi počátečním nástřelem z předchozí časové hladiny a přesným řešením soustavy. Po vyrovnaní rozdílu mezi okrajovou a počáteční podmínkou, přibližně od času $t = 1\,000$ s, již stačí pouze desítky iterací. Z hlediska výpočetní náročnosti by bylo výhodné volit časový krok adaptivně, ale při analýze konvergence předpokládáme, že časový krok je konstantní. Časový krok je tedy nutné volit dostatečně malý, aby iterační řešič GMRES na začátku simulace zkonvergoval, ale také dostatečně velký, aby celá simulace skončila v rozumném čase. Konvergenci v problematické části lze urychlit (z hlediska počtu iterací) zvětšením restartovacího parametru m , ale za cenu větších nároků na dostupnou paměť a rostoucí množství práce při konstrukci ortonormální báze Krylovova podprostoru. Pokud však metoda zkonverguje ještě před restartováním, nemá zvětšování m žádný vliv na dobu výpočtu. Jelikož tento případ nastává ve většině případů v průběhu simulace, je zvětšení parametru m přijatelnou investicí, která může na jemných sítích dokonce zajistit konvergenci metody. Pro řešení úloh 5.1 a 5.2 byla použita hodnota $m = 100$.

V případě McWhorterovy–Sunadovy úlohy jsou hodnoty EOC v tab. 5.1 a 5.2 při zmenšování časového kroku $\tau \propto h^2$ ve všech normách rovny přibližně 1. Chyba časové diskretizace tedy byla potlačena a lze tedy říci, že při řešení difúzní úlohy v homogenním prostředí je konvergence numerického schématu prvního řádu.

Hodnoty EOC pro difúzní úlohu v heterogenním prostředí s počáteční podmínkou $S_n(\mathbf{x}, 0) = 0$ na Ω_2 jsou uvedeny v tab. 5.3 a 5.4. EOC v L^1 normě vychází opět přibližně 1, ale v L^2 normě pouze 0,8. V L^∞ normě se chyba redukuje velmi pomalu, protože přesné řešení S_n na obr. 5.2a má v bodě $x = 0,327$ m nekonečnou derivaci v důsledku volby počáteční podmínky $S_n(\mathbf{x}, 0) = 0$ na Ω_2 . Změnou počáteční podmínky např. na $S_n(\mathbf{x}, 0) = 0,03$ na Ω_2 se přesné řešení S_n vyhladí (viz obr. 5.2b) a konvergenci bude možno pozorovat i v normě L^∞ . Dle výsledků uvedených v tab. 5.5 a 5.6 je v tomto případě konvergence prvního řádu ve všech normách, stejně jako v McWhorterově–Sunadově úloze.

Výsledky prezentované v tab. 5.1 až 5.6 byly získány výpočtem na GPU metodou GMRES využívající modifikovaný Gramův–Schmidtův proces s reortogonalizací, která je v kombinaci s Jacobiho předpokládáním nutná pro zajištění konvergence. Použití ortogonalizace pomocí kompaktní WY reprezentace dává prakticky stejné výsledky jako modifikovaný Gramův–Schmidtův proces s reortogonalizací. Podobně se i výsledky výpočtu stejné metody na CPU a GPU mírně liší, což je způsobeno použitím paralelních algoritmů např. pro výpočet skalárních součinů a pro $\varepsilon \rightarrow 0$ se také tento rozdíl blíží k 0.

N_x	$\ E_h\ _1$	EOC ₁	$\ E_h\ _2$	EOC ₂	$\ E_h\ _\infty$	EOC _∞
20	$3,571 \cdot 10^{-02}$		$5,396 \cdot 10^{-02}$		$1,646 \cdot 10^{-01}$	
40	$1,980 \cdot 10^{-02}$	0,851	$3,323 \cdot 10^{-02}$	0,699	$1,142 \cdot 10^{-01}$	0,528
80	$1,075 \cdot 10^{-02}$	0,881	$1,936 \cdot 10^{-02}$	0,779	$7,427 \cdot 10^{-02}$	0,621
160	$5,653 \cdot 10^{-03}$	0,927	$1,064 \cdot 10^{-02}$	0,863	$4,439 \cdot 10^{-02}$	0,743
320	$2,915 \cdot 10^{-03}$	0,956	$5,623 \cdot 10^{-03}$	0,921	$2,446 \cdot 10^{-02}$	0,860
640	$1,485 \cdot 10^{-03}$	0,973	$2,900 \cdot 10^{-03}$	0,955	$1,292 \cdot 10^{-02}$	0,921
1280	$7,535 \cdot 10^{-04}$	0,979	$1,477 \cdot 10^{-03}$	0,973	$6,640 \cdot 10^{-03}$	0,960
2560	$3,640 \cdot 10^{-04}$	1,050	$7,252 \cdot 10^{-04}$	1,026	$3,315 \cdot 10^{-03}$	1,002

Tabulka 5.1: EOC v normách L^1 , L^2 a L^∞ pro McWhorterovu–Sunadovu úlohu 5.1 v 1D. Časový krok je volen $\tau = 250$ s na nejhrubší síti $N_x = 20$ a pro jemnější síť $\tau \propto h^2$.

$N_x \times N_y$	$\ E_h\ _1$	EOC ₁	$\ E_h\ _2$	EOC ₂	$\ E_h\ _\infty$	EOC _∞
20×5	$8,929 \cdot 10^{-03}$		$2,698 \cdot 10^{-02}$		$1,646 \cdot 10^{-01}$	
40×10	$4,949 \cdot 10^{-03}$	0,851	$1,662 \cdot 10^{-02}$	0,699	$1,142 \cdot 10^{-01}$	0,528
80×20	$2,686 \cdot 10^{-03}$	0,882	$9,682 \cdot 10^{-03}$	0,779	$7,427 \cdot 10^{-02}$	0,621
160×40	$1,413 \cdot 10^{-03}$	0,927	$5,321 \cdot 10^{-03}$	0,863	$4,438 \cdot 10^{-02}$	0,743
320×80	$7,274 \cdot 10^{-04}$	0,958	$2,808 \cdot 10^{-03}$	0,922	$2,445 \cdot 10^{-02}$	0,860
640×160	$3,675 \cdot 10^{-04}$	0,985	$1,441 \cdot 10^{-03}$	0,963	$1,287 \cdot 10^{-02}$	0,926
1280×320	$1,782 \cdot 10^{-04}$	1,044	$7,137 \cdot 10^{-04}$	1,013	$6,495 \cdot 10^{-03}$	0,986

Tabulka 5.2: EOC v normách L^1 , L^2 a L^∞ pro McWhorterovu–Sunadovu úlohu 5.1 ve 2D. Časový krok je volen $\tau = 250$ s na nejhrubší síti $N_x \times N_y = 20 \times 5$ a pro jemnější síť $\tau \propto h^2$.

N_x	$\ E_h\ _1$	EOC ₁	$\ E_h\ _2$	EOC ₂	$\ E_h\ _\infty$	EOC _∞
20	$3,020 \cdot 10^{-02}$		$4,623 \cdot 10^{-02}$		$1,292 \cdot 10^{-01}$	
40	$1,616 \cdot 10^{-02}$	0,902	$2,708 \cdot 10^{-02}$	0,771	$1,018 \cdot 10^{-01}$	0,343
80	$8,162 \cdot 10^{-03}$	0,985	$1,520 \cdot 10^{-02}$	0,833	$6,739 \cdot 10^{-02}$	0,596
160	$4,190 \cdot 10^{-03}$	0,962	$8,503 \cdot 10^{-03}$	0,838	$4,266 \cdot 10^{-02}$	0,660
320	$2,165 \cdot 10^{-03}$	0,953	$4,716 \cdot 10^{-03}$	0,850	$3,775 \cdot 10^{-02}$	0,177
640	$1,121 \cdot 10^{-03}$	0,950	$2,589 \cdot 10^{-03}$	0,865	$2,949 \cdot 10^{-02}$	0,356
1280	$5,757 \cdot 10^{-04}$	0,961	$1,435 \cdot 10^{-03}$	0,851	$2,664 \cdot 10^{-02}$	0,147
2560	$2,832 \cdot 10^{-04}$	1,023	$8,263 \cdot 10^{-04}$	0,797	$2,424 \cdot 10^{-02}$	0,136

Tabulka 5.3: EOC v normách L^1 , L^2 a L^∞ pro difúzní úlohu v heterogenním prostředí 5.2 v 1D. Časový krok je volen $\tau = 250$ s na nejhrubší síti $N_x = 20$ a pro jemnější síť $\tau \propto h^2$.

$N_x \times N_y$	$\ E_h\ _1$	EOC ₁	$\ E_h\ _2$	EOC ₂	$\ E_h\ _\infty$	EOC _∞
20×5	$7,551 \cdot 10^{-03}$	0,902 0,985 0,962 0,950 0,942	$2,311 \cdot 10^{-02}$	0,771 0,833 0,838 0,851 0,868	$1,292 \cdot 10^{-01}$	0,343 0,596 0,660 0,179 0,369
40×10	$4,040 \cdot 10^{-03}$		$1,354 \cdot 10^{-02}$		$1,018 \cdot 10^{-01}$	
80×20	$2,040 \cdot 10^{-03}$		$7,600 \cdot 10^{-03}$		$6,739 \cdot 10^{-02}$	
160×40	$1,048 \cdot 10^{-03}$		$4,252 \cdot 10^{-03}$		$4,266 \cdot 10^{-02}$	
320×80	$5,422 \cdot 10^{-04}$		$2,358 \cdot 10^{-03}$		$3,768 \cdot 10^{-02}$	
640×160	$2,823 \cdot 10^{-04}$		$1,292 \cdot 10^{-03}$		$2,918 \cdot 10^{-02}$	

Tabulka 5.4: EOC v normách L^1 , L^2 a L^∞ pro difúzní úlohu v heterogenním prostředí 5.2 ve 2D. Časový krok je volen $\tau = 250$ s na nejhrubší síti $N_x \times N_y = 20 \times 5$ a pro jemnější síť $\tau \propto h^2$.

N_x	$\ E_h\ _1$	EOC ₁	$\ E_h\ _2$	EOC ₂	$\ E_h\ _\infty$	EOC _∞
20	$2,787 \cdot 10^{-02}$	0,903 0,996 0,979 0,971 0,971 0,985 1,000	$4,382 \cdot 10^{-02}$	0,768 0,833 0,857 0,897 0,935 0,964 0,983	$1,236 \cdot 10^{-01}$	0,330 0,621 0,631 0,788 0,882 0,937 0,974
40	$1,491 \cdot 10^{-02}$		$2,572 \cdot 10^{-02}$		$9,831 \cdot 10^{-02}$	
80	$7,472 \cdot 10^{-03}$		$1,444 \cdot 10^{-02}$		$6,392 \cdot 10^{-02}$	
160	$3,791 \cdot 10^{-03}$		$7,971 \cdot 10^{-03}$		$4,127 \cdot 10^{-02}$	
320	$1,933 \cdot 10^{-03}$		$4,280 \cdot 10^{-03}$		$2,391 \cdot 10^{-02}$	
640	$9,865 \cdot 10^{-04}$		$2,238 \cdot 10^{-03}$		$1,297 \cdot 10^{-02}$	
1280	$4,985 \cdot 10^{-04}$		$1,148 \cdot 10^{-03}$		$6,778 \cdot 10^{-03}$	
2560	$2,492 \cdot 10^{-04}$		$5,806 \cdot 10^{-04}$		$3,452 \cdot 10^{-03}$	

Tabulka 5.5: EOC v normách L^1 , L^2 a L^∞ pro modifikovanou difúzní úlohu v heterogenním prostředí v 1D s počáteční podmínkou $S_n(x, 0) = 0,03$ na Ω_2 . Časový krok je volen $\tau = 250$ s na nejhrubší síti $N_x = 20$ a pro jemnější síť $\tau \propto h^2$.

$N_x \times N_y$	$\ E_h\ _1$	EOC ₁	$\ E_h\ _2$	EOC ₂	$\ E_h\ _\infty$	EOC _∞
20×5	$6,968 \cdot 10^{-03}$	0,903 0,996 0,979 0,970 0,967	$2,191 \cdot 10^{-02}$	0,768 0,833 0,857 0,897 0,934	$1,236 \cdot 10^{-01}$	0,330 0,621 0,631 0,787 0,881
40×10	$3,726 \cdot 10^{-03}$		$1,286 \cdot 10^{-02}$		$9,831 \cdot 10^{-02}$	
80×20	$1,868 \cdot 10^{-03}$		$7,218 \cdot 10^{-03}$		$6,392 \cdot 10^{-02}$	
160×40	$9,480 \cdot 10^{-04}$		$3,985 \cdot 10^{-03}$		$4,127 \cdot 10^{-02}$	
320×80	$4,840 \cdot 10^{-04}$		$2,141 \cdot 10^{-03}$		$2,391 \cdot 10^{-02}$	
640×160	$2,476 \cdot 10^{-04}$		$1,120 \cdot 10^{-03}$		$1,298 \cdot 10^{-02}$	

Tabulka 5.6: EOC v normách L^1 , L^2 a L^∞ pro difúzní úlohu v heterogenním prostředí ve 2D s počáteční podmínkou $S_n(x, 0) = 0,03$ na Ω_2 . Časový krok je volen $\tau = 250$ s na nejhrubší síti $N_x \times N_y = 20 \times 5$ a pro jemnější síť $\tau \propto h^2$.

5.3.2 Efektivita paralelního výpočtu

Efektivitu paralelního výpočtu analyzujeme pro McWhorterovu–Sunadovu úlohu 5.1. Budeme zkoumat doby sériového výpočtu na CPU a paralelního výpočtu na GPU na sítích s různým počtem elementů. Parametry CPU a GPU použitých k výpočtům jsou uvedeny v tab. 5.7. Znamé semi-analytické řešení pro danou úlohu umožňuje porovnání efektivity řešiče při nastavení vedoucím k přibližně stejným výsledkům na CPU i GPU. Parametry metody GMRES jsou voleny stejně jako při experimentální analýze konvergence (tedy $\varepsilon = 10^{-11}$ a $m = 100$), takže níže uvedené výsledky odpovídají výsledkům prezentovaným v sekci 5.3.1.

	CPU	GPU
Název	Intel Core i7-5820K	NVIDIA Tesla K40c
Počet jader★	6	2880
Frekvence (boost)♠	3.3 GHz (3.6 GHz)	745 MHz (875 MHz)
Cache♣	15 MB	1.5 MB
Velikost paměti	64 GB	12 GB
Teoretická propustnost paměti	17 GB/s	288 GB/s

★ Výpočty na CPU vždy využívají pouze 1 jádro.

♠ Maximální frekvence dosažitelná pomocí technologie Intel Turbo Boost resp. NVIDIA GPU Boost.

♣ L3 cache v případě CPU, L2 cache v případě GPU.

Tabulka 5.7: Hardware použitý k porovnání efektivity řešiče.

Hlavním faktorem pro srovnání rychlosti sériového a paralelního výpočtu je urychlení, počítané jako doba výpočtu na CPU dělená dobou výpočtu na GPU. Urychlení zřejmě závisí na velikosti diskrétní úlohy, kterou popisujeme pomocí veličiny $N_{DOF} = nN_E$ vyjadřující počet tzv. stupňů volnosti (DOF) a odpovídající počtu rovnic v soustavě (3.34). V jednorozměrném případě je $N_E = N_x + 1$, ve 2D je $N_E = 2N_xN_y + N_x + N_y$.

Nejprve zkoumáme efektivitu základní verze řešiče využívající metodu GMRES s modifikovaným Gramovým–Schmidtovým ortogonalizačním procesem (MGS). Jak již bylo zmíněno v sekci 5.3.1, pro zajištění konvergence je potřeba provádět reortogonalizaci opakováním kroku 2.2. v algoritmu 2. Doby výpočtu McWhorterovy–Sunadovy úlohy jsou uvedeny v tab. 5.8 pro původní jednorozměrnou formulaci

N_x	Čas [h : mm : ss] (CPU)	Čas [h : mm : ss] (GPU)	Urychlení [–] (CPU/GPU)
20	0,1	6,0	0,01
40	0,4	53,4	0,01
80	3,1	5 : 39	0,01
160	16,9	14 : 12	0,02
320	1 : 03	23 : 29	0,04
640	4 : 31	39 : 28	0,11
1280	23 : 55	1 : 15 : 26	0,32
2560	2 : 27 : 36	2 : 54 : 52	0,84

Tabulka 5.8: Doby výpočtu McWhorterovy–Sunadovy úlohy 5.1 v 1D na sítích s N_x elementy. V metodě GMRES byl použit algoritmus MGS.

$N_x \times N_y$	N_{DOF}	Čas [h : mm : ss] (CPU)	Čas [h : mm : ss] (GPU)	Urychlení [–] (CPU/GPU)
20×5	450	0,8	23,8	0,03
40×10	1 700	15,1	2 : 48	0,09
80×20	6 600	4 : 02	11 : 25	0,35
160×40	26 000	34 : 48	25 : 08	1,38
320×80	103 200	4 : 40 : 49	59 : 42	4,70
640×160	411 200	41 : 54 : 17	4 : 38 : 18	9,03
1280×320	1 641 600	526 : 43 : 10	37 : 47 : 27	13,94

(a) V metodě GMRES byl použit algoritmus MGS.

$N_x \times N_y$	N_{DOF}	Čas [h : mm : ss] (CPU)	Čas [h : mm : ss] (GPU)	Urychlení [–] (CPU/GPU)
20×5	450	0,9	3,8	0,25
40×10	1 700	19,7	23,1	0,85
80×20	6 600	5 : 02	2 : 05	2,42
160×40	26 000	44 : 33	6 : 29	6,86
320×80	103 200	6 : 19 : 33	27 : 10	13,97
640×160	411 200	53 : 37 : 44	3 : 02 : 24	17,64
1280×320	1 641 600	nepočítáno	31 : 15 : 11	–

(b) V metodě GMRES byl použit algoritmus CWY.

$N_x \times N_y$	N_{DOF}	Čas [h : mm : ss] (CPU, MGS)	Čas [h : mm : ss] (GPU, CWY)	Urychlení [–] (CPU/GPU)
20×5	450	0,8	3,8	0,20
40×10	1 700	15,1	23,1	0,66
80×20	6 600	4 : 02	2 : 05	1,94
160×40	26 000	34 : 48	6 : 29	5,36
320×80	103 200	4 : 40 : 49	27 : 10	10,33
640×160	411 200	41 : 54 : 17	3 : 02 : 24	13,78
1280×320	1 641 600	526 : 43 : 10	31 : 15 : 11	16,85

(c) Srovnání algoritmu MGS na CPU a CWY na GPU.

Tabulka 5.9: Srovnání dob výpočtu McWhorterovy–Sunadovy úlohy 5.1 ve 2D na sítích s $N_x \times N_y$ elementy při použití ortogonalizačních algoritmů MGS a CWY v metodě GMRES.

a v tab. 5.9a pro úlohu rozšířenou do 2D. Pro malý počet stupňů volnosti není možné udržet všech 2880 výpočetních jader použitého GPU aktivních po dostatečně dlouhou dobu, projevuje se častá synchronizace a dodatečná režie a k urychlení výpočtu tak nedojde. V 1D je tedy na všech zkoumaných sítích rychlejší výpočet na CPU, urychlení paralelního výpočtu by se mělo projevit na ještě větších sítích. Při výpočtu na 2D sítích se teprve od přibližně 20 000 stupňů volnosti rychlost sériového a paralelního výpočtu srovná a dále s rostoucím počtem stupňů volnosti roste také urychlení. Na nejmenější použité síti $N_x \times N_y = 1280 \times 320$ bylo dosaženo urychlení 13,94.

Dále provedeme podrobnější analýzu efektivity paralelní implementace a určíme nejnáročnější část výpočtu, jejíž optimalizace by měla být předmětem dalšího vývoje řešiče. V tab. 5.10 jsou uvedeny doby trvání hlavních kroků algoritmu popsaného v sekci 3.3 při výpočtu na dvou nejmenších sítích $N_x \times N_y = 640 \times 160$ a $N_x \times N_y = 1280 \times 320$ pro CPU a GPU. Pro všechny uvedené kroky je paralelní implementace na GPU rychlejší oproti sériové CPU verzi, nicméně celková doba výpočtu je dána především řešením soustavy lineárních algebraických rovnic s řídkou maticí v kroku 6.4.. Nejmenší urychlení je u kroku 6.1., kde navíc došlo k zajímavému poklesu urychlení při přechodu na jemnější síť, což by mohlo být předmětem dalších optimalizací, nicméně vliv na celkovou dobu výpočtu je zanedbatelný. Kroky 6.2., 6.3. a 6.5. jsou urychleny přibližně 10×, což je blízko celkové hodnotě urychlení. Zbývající kroky 6.6. a 6.7. jsou urychleny mnohanásobně nadprůměrně, což pozorovatelně

Krok	Symbolický popis výpočtu	Čas [s] (CPU)	Čas [s] (GPU)	Urychlení [–] (CPU/GPU)
6.1.	$\mathbf{b}_{i,j,K}, \mathbf{R}_{K,F}, \mathbf{R}_K$	2 099 (1%)	235 (1%)	8,92
6.2.	$\mathbf{Q}_K^{-1} \mathbf{R}_{K,F}, \mathbf{Q}_K^{-1} \mathbf{R}_K$	2 538 (2%)	237 (1%)	10,73
6.3.	\mathbf{A}	12 568 (8%)	1 152 (7%)	10,91
6.4.	$\mathbf{Z}^{k+1} = \mathbf{A}^{-1} \mathbf{b}$	123 447 (82%)	14 626 (88%)	8,44
6.5.	$\mathbf{Z}_{i,K}^{k+1}$	1 071 (1%)	116 (1%)	9,20
6.6.	$N_{i,j,K}, m_{i,K}, \mathbf{D}_{i,j,K}, w_{i,K,E}, f_{i,K}$	6 040 (4%)	46 (0%)	130,07
6.7.	$m_{i,E}^{upw}$	3 095 (2%)	106 (1%)	29,09
Komunikace a režie řízení		1 (0%)	180 (1%)	0,01
Celková doba výpočtu		150 857 (100%)	16 699 (100%)	9,03

(a) $N_x \times N_y = 640 \times 160$

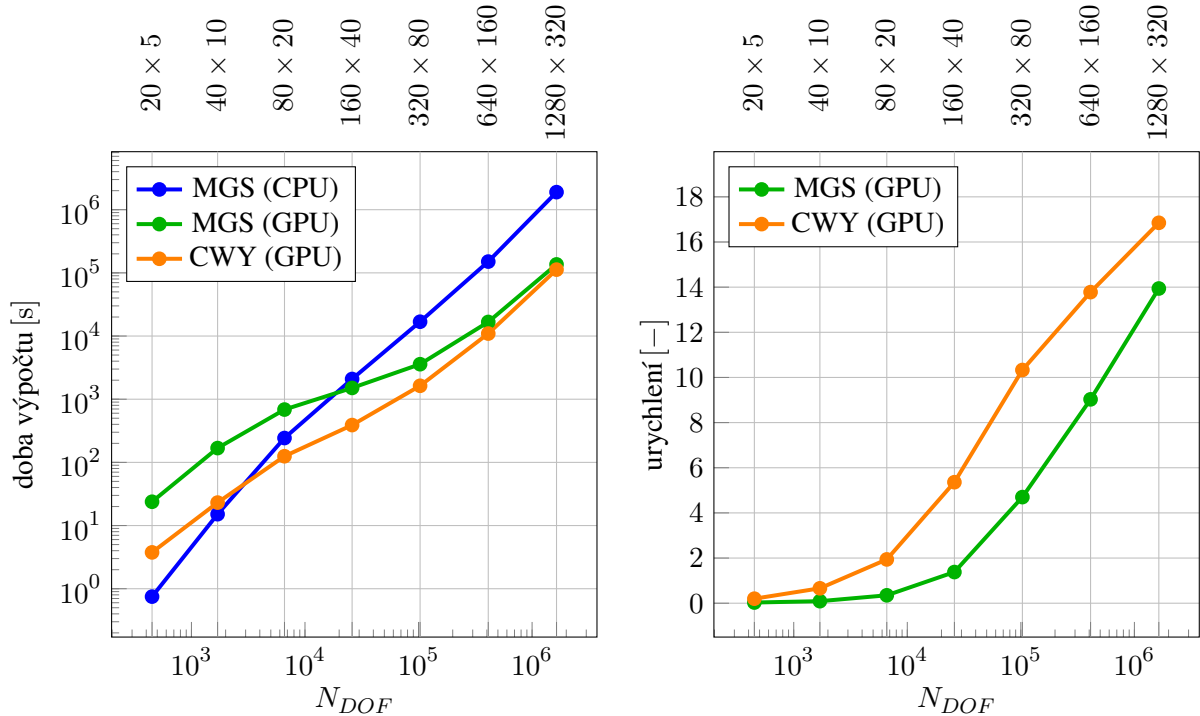
Krok	Symbolický popis výpočtu	Čas [s] (CPU)	Čas [s] (GPU)	Urychlení [–] (CPU/GPU)
6.1.	$\mathbf{b}_{i,j,K}, \mathbf{R}_{K,F}, \mathbf{R}_K$	34 801 (2%)	4 103 (3%)	8,48
6.2.	$\mathbf{Q}_K^{-1} \mathbf{R}_{K,F}, \mathbf{Q}_K^{-1} \mathbf{R}_K$	41 850 (2%)	3 637 (3%)	11,51
6.3.	\mathbf{A}	208 975 (11%)	17 065 (13%)	12,25
6.4.	$\mathbf{Z}^{k+1} = \mathbf{A}^{-1} \mathbf{b}$	1 441 347 (76%)	105 505 (78%)	13,66
6.5.	$\mathbf{Z}_{i,K}^{k+1}$	17 668 (1%)	1 562 (1%)	11,31
6.6.	$N_{i,j,K}, m_{i,K}, \mathbf{D}_{i,j,K}, w_{i,K,E}, f_{i,K}$	100 244 (5%)	594 (0%)	168,84
6.7.	$m_{i,E}^{upw}$	51 300 (3%)	1 322 (1%)	38,80
Komunikace a režie řízení		4 (0%)	2 258 (2%)	0,00
Celková doba výpočtu		1 896 190 (100%)	136 047 (100%)	13,94

(b) $N_x \times N_y = 1280 \times 320$

Tabulka 5.10: Rozbor efektivity paralelní implementace. První sloupec odkazuje na jednotlivé kroky algoritmu popsaného v sekci 3.3, druhý sloupec obsahuje symbolický popis obsahu jednotlivých kroků, třetí a čtvrtý sloupec obsahuje dobu trvání daného kroku při výpočtu na CPU resp. GPU (v závorce je uvedeno procentuální zastoupení daného kroku na celkové době výpočtu) a poslední sloupec obsahuje urychlení daného kroku na GPU oproti výpočtu na CPU.

zvyšuje hodnotu celkového urychlení oproti dominantnímu kroku 6.4.. Jedinou položkou, která je při výpočtu na GPU pomalejší než na CPU, je komunikace a dodatečná režie řízení výpočtu. Celkově tedy můžeme konstatovat, že algoritmus popsáný v sekci 3.3 lze dobře masivně paralelizovat a jeho efektivita je dána především způsobem řešení soustavy lineárních algebraických rovnic s řádkou maticí v kroku 6.4..

Prvním krokem ke zlepšení efektivity řešiče je použití modifikace metody GMRES pomocí Householderových transformací a kompaktní WY reprezentace, která byla popsána v sekci 4.1.3. Podobně jako v případě algoritmu MGS srovnáme efektivitu algoritmu CWY, tentokrát již pouze ve 2D. Z hodnot v tab. 5.9b je patrné, že při sekvenčním výpočtu je algoritmus MGS rychlejší než CWY, ale při paralelním výpočtu je to naopak. V tab. 5.9c proto dále srovnáme nejrychlejší sekvenční algoritmus a nejrychlejší paralelní algoritmus, tedy MGS na CPU a CWY na GPU. Grafy pozorované závislosti doby výpočtu na počtu stupňů volnosti a urychlení obou paralelních algoritmů vzhledem k sériovému algoritmu MGS jsou zobrazeny na obr. 5.4. Je patrné, že paralelní algoritmus CWY škáluje mnohem lépe než MGS, na hrubých sítích je přibližně šestkrát rychlejší a na jemných sítích je rozdíl také výrazný (na nejjemnější síti až 6 hodin). Celkově byl na nejjemnější použité síti $N_x \times N_y = 1280 \times 320$ paralelní výpočet pomocí algoritmu CWY $16,85\times$ rychlejší oproti sekvenčnímu výpočtu pomocí algoritmu MGS.

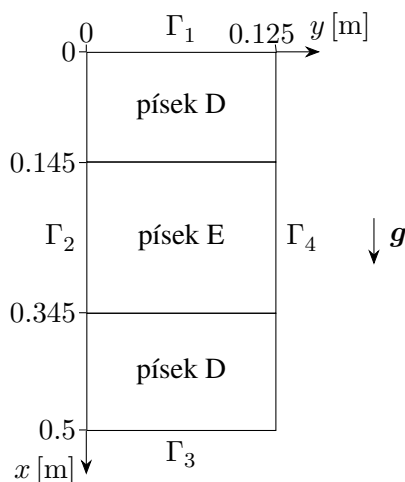


Obrázek 5.4: Závislost doby výpočtu numerického řešení a urychlení na počtu stupňů volnosti $N_{DOF} = nN_E$. Urychlení je počítáno vzhledem k algoritmu MGS na CPU.

5.4 Bariérový efekt

V následující úloze ověříme schopnost numerického schématu správně zachytit chování při prostupu nesmáčivé fáze přes materiálové rozhraní. Bariérový efekt je jev, kdy při prostupu nesmáčivé fáze z hrubého písku do jemného dojde k akumulaci nesmáčivé fáze před rozhraním a teprve po dosažení vstupního tlaku p_d pro jemnější písek začne nesmáčivá fáze prosakovat přes rozhraní.

Geometrická konfigurace úlohy je znázorněna na obr. 5.5. Horní a dolní část oblasti Ω jsou vyplněny pískem D, který je hrubší než písek E vyplňující prostřední část oblasti. Parametry použitých písků jsou uvedeny v tab. A.2. Na počátku simulace v čase $t = 0$ je celá oblast zaplněna vodou ($\rho_w = 1000 \text{ kg m}^{-3}$, $\mu_w = 10^{-3} \text{ kg m}^{-1} \text{ s}^{-1}$) a horní hranicí Γ_1 začne prosakovat nesmáčivá fáze ($\rho_n = 1400 \text{ kg m}^{-3}$, $\mu_n = 10^{-3} \text{ kg m}^{-1} \text{ s}^{-1}$) rychlostí o velikosti $|v_n^N(0, t)| = 3,57 \cdot 10^{-5} \text{ m s}^{-1}$, přičemž rychlost prosakování vody je nulová na Γ_1 . Na spodní hranici Γ_3 je udržován konstantní tlak vody $2 \cdot 10^5 \text{ Pa}$ a saturace $S_w = 1$. V případě rozšíření jednorozměrné úlohy do \mathbb{R}^2 přibudou hranice Γ_2 a Γ_4 , které jsou nepropustné pro obě fáze. V úloze uvažujeme vliv gravitace ve směru osy x .

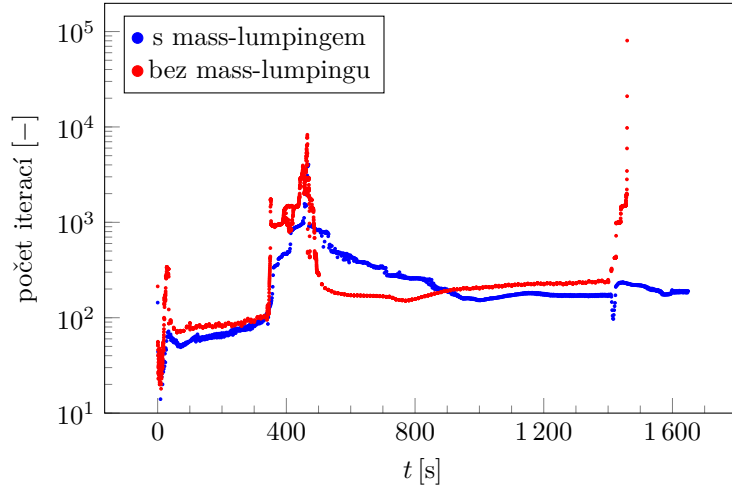


Obrázek 5.5: Geometrická konfigurace úlohy simulující bariérový efekt.

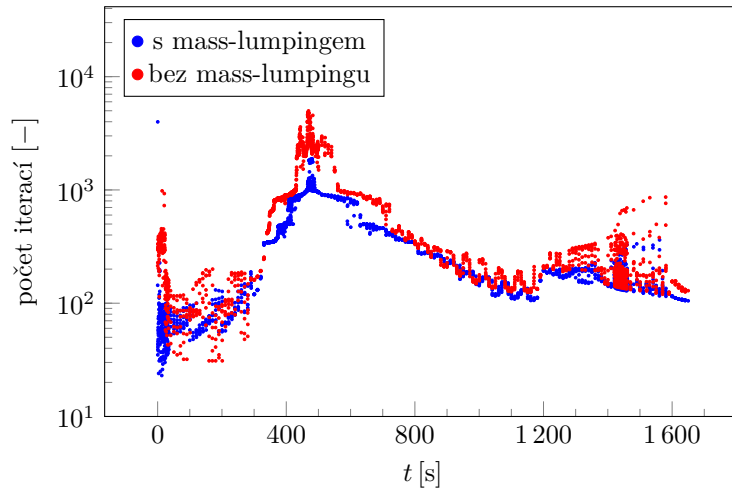
Řešení v čase $T = 1650 \text{ s}$ lze srovnat s výsledky publikovanými v literatuře [9, 19]. Zajímavé je srovnání výsledků dosažených pomocí různých modelů kapilarity a porovnání vlivu metody mass-lumping. Nejprve použijeme model Brookse a Coreyho (1.12) a schéma s původními lokálními maticemi $\mathbf{b}_{i,j,K}$ (viz (3.36) a (3.41)) bez aproximace metodou mass-lumping. Numerické řešení 1D úlohy pro tento případ je zobrazeno na obr. 5.8a. Řešení metodou MHFEM vykazuje výrazné oscilace na materiálových rozhraních, které nelze potlačit zjemněním sítě ani zmenšením časového kroku. Tyto oscilace způsobují těžkosti z hlediska numerického řešení úlohy, a to zejména ve 2D při použití jemných sítí. Například na síti $N_x \times N_y = 400 \times 100$ nezkonverguje metoda GMRES, jakmile nesmáčivá fáze dorazí na druhé rozhraní v $x = 0,345 \text{ m}$. Tato situace je zobrazena na obr. 5.6.

Dále použijeme schéma s lokálními maticemi $\mathbf{b}_{i,j,K}^{(\ell)}$ (viz (3.39) a (3.43)) aproximovanými pomocí metody mass-lumping. Numerické řešení 1D úlohy pro tento případ je zobrazeno na obr. 5.8b. Řešení metodou MHFEM již neobsahuje oscilace a je v souladu s referenčním řešením pomocí plně implicitní VCFVM² metody, které bylo převzato z [9]. Potlačením oscilací také zmizel problém s konvergencí metody GMRES při průchodu nesmáčivé fáze přes materiálová rozhraní. Dle obr. 5.6 pro 2D formulaci

²Vertex-Centered Finite Volume Method



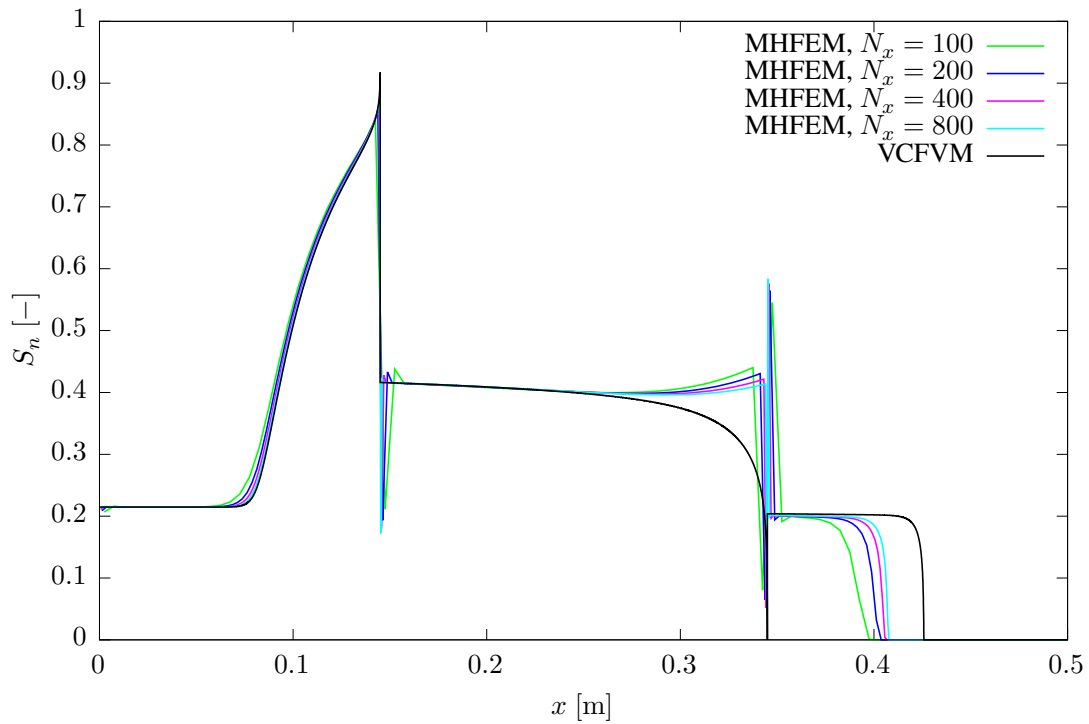
Obrázek 5.6: Počet iterací potřebných pro konvergenci metody $\text{GMRES}(m)$ v průběhu simulace bariérového efektu s využitím modelu Brookse a Coreyho ve 2D na síti s 400×100 elementy. Schéma bez použití mass-lumpingu v čase $t = 1\,459$ s nezkonvergovalo.



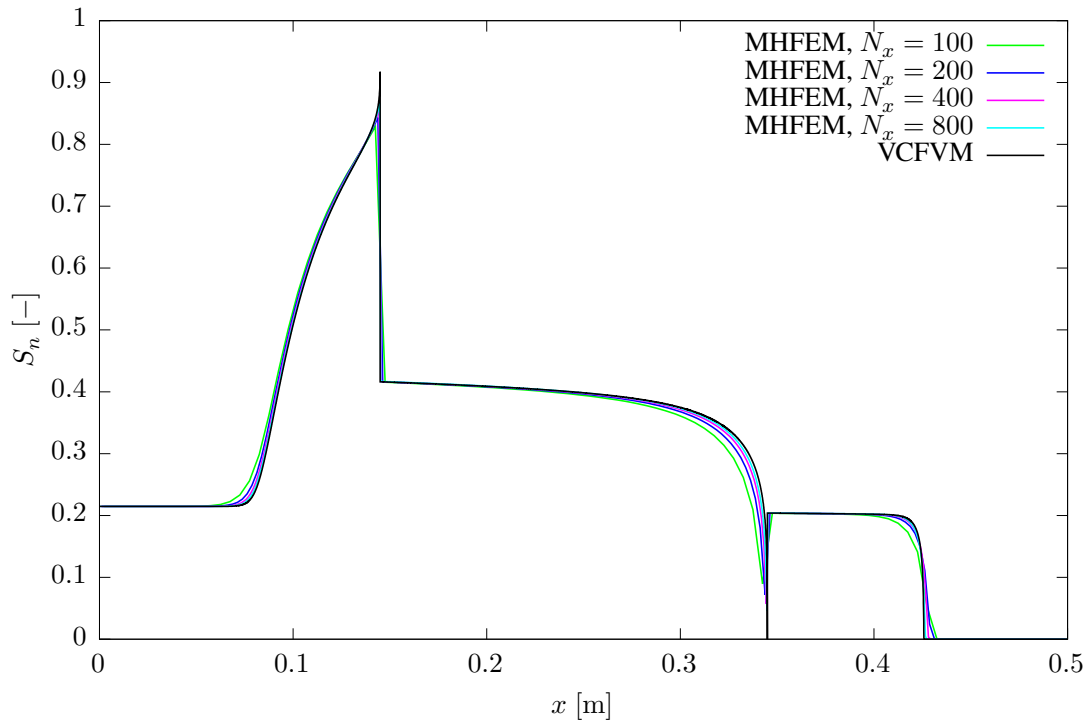
Obrázek 5.7: Počet iterací potřebných pro konvergenci metody $\text{GMRES}(m)$ v průběhu simulace bariérového efektu s využitím modelu van Genuchtena ve 2D na síti s 400×100 elementy.

úlohy sice dojde k nárůstu počtu iterací při průchodu z hrubého písku do jemného, ale při průchodu z jemného písku do hrubého je počet iterací přibližně konstantní.

Při použití van Genuchtenova modelu (1.13) a schématu bez mass-lumpingu opět dojde k oscilacím na materiálových rozhraních, což je patrné z obr. 5.9a pro 1D úlohu. Oproti modelu Brookse a Coreyho postupuje nesmáčivá fáze rychleji a v čase $T = 1\,650$ s již dospěla na protější okraj oblasti, kde dojde k oscilacím v důsledku okrajové podmínky. Oscilace lze opět potlačit použitím metody mass-lumping. Řešení pro tento případ zobrazené na obr. 5.9b je v souladu s řešením publikovaným v literatuře [19]. Také z hlediska rychlosti konvergence metody GMRES je použití mass-lumpingu výhodné, ve 2D případě je na obr. 5.7 patrný rozdíl v počtu iterací metody GMRES v průběhu simulace. Schéma sice zkonvergovalo i bez využití mass-lumpingu, ale za cenu přibližně o 60% většího počtu iterací v průběhu celé simulace.

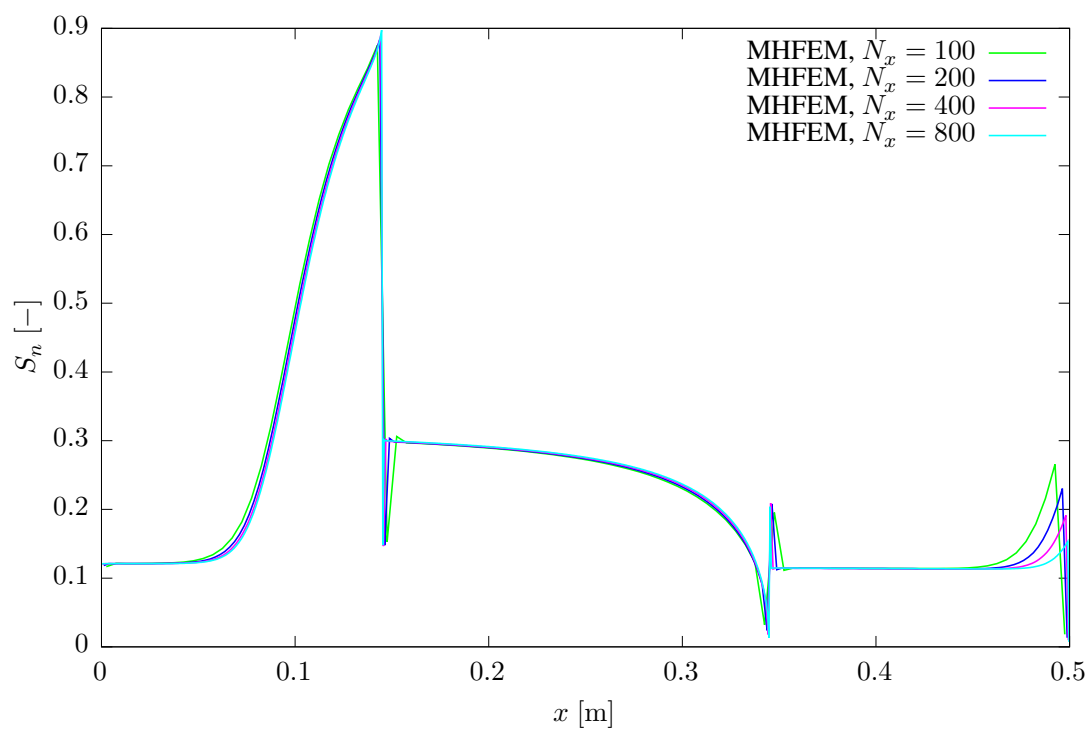


(a) Schéma bez mass-lumpingu

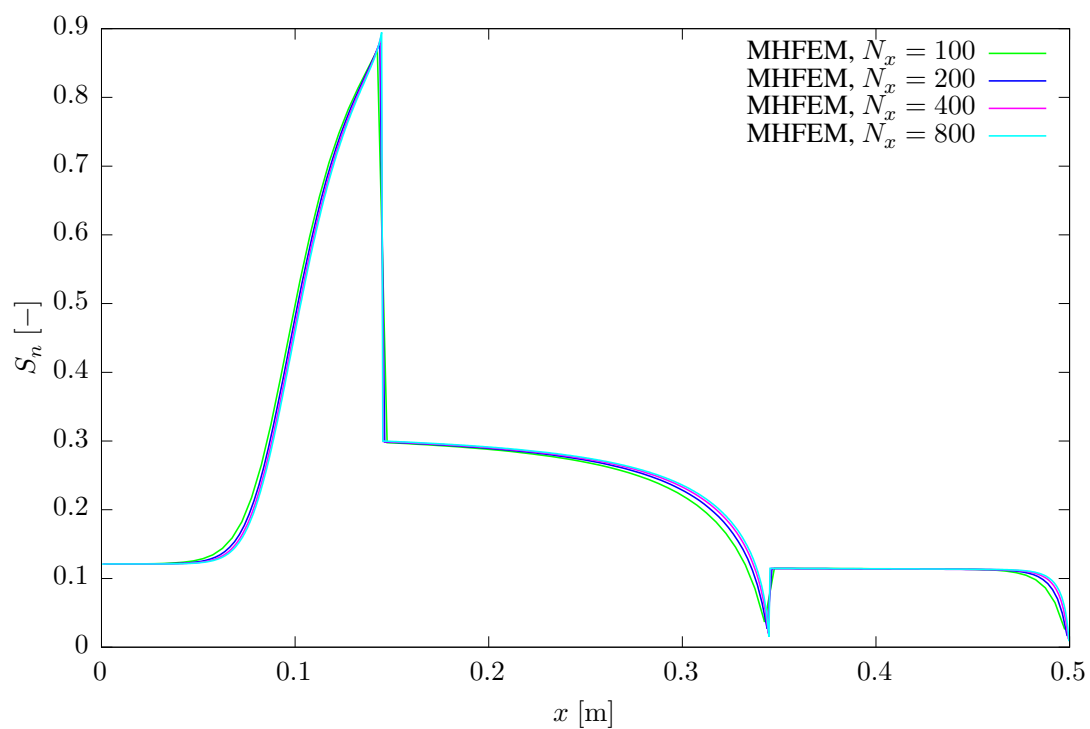


(b) Schéma s mass-lumpinem

Obrázek 5.8: Simulace bariérového efektu v 1D v čase $T = 1\,650$ s při použití modelu Brookse a Coreyho pomocí MHFEM na sítích s N_x elementy. Referenční řešení pomocí VCFVM bylo převzato z [9].



(a) Schéma bez mass-lumpingu



(b) Schéma s mass-lumpingem

Obrázek 5.9: Simulace bariérového efektu v 1D v čase $T = 1\,650$ s při použití modelu van Genuchtena pomocí MHFEM na sítích s N_x elementy.

5.5 Obtékání překážky

Dále budeme řešit čistě dvourozměrnou úlohu v heterogenním prostředí. Uvažujeme oblast Ω znázorněnou na obr. 5.10, uvnitř které je vymezena podoblast Ω_1 . Vnitřní podoblast Ω_1 je vyplněna jemným pískem G a zbývající část oblasti Ω je vyplněna hrubším pískem F. V případě písku G uvažujeme dvě různé varianty G_1 a G_2 (viz tab. A.3), které se liší hodnotami vstupního tlaku p_d tak, že v dané úloze tvoří oblast Ω_1 vzhledem k nesmáčivé fázi buď propustnou nebo nepropustnou překážku.

Na počátku simulace v čase $t = 0$ je celá oblast Ω plně nasycena vodou ($\rho_w = 997,78 \text{ kg m}^{-3}$, $\mu_w = 9,772 \cdot 10^{-4} \text{ kg m}^{-1} \text{ s}^{-1}$) a částí hranice oblasti Γ_1 začne konstantní rychlostí vtékat nesmáčivá fáze ($\rho_n = 1460 \text{ kg m}^{-3}$, $\mu_n = 9 \cdot 10^{-4} \text{ kg m}^{-1} \text{ s}^{-1}$). Části hranice Γ_3 , Γ_4 a Γ_5 jsou nepropustné pro obě fáze a na Γ_2 a Γ_6 předepíšeme nulovou saturaci nesmáčivé fáze a hydrostatický tlak vody. Celkově na jednotlivých úsecích hranice oblasti Ω předepisujeme tyto okrajové podmínky:

$$\mathbf{v}_w \cdot \mathbf{n} = 0, \quad \mathbf{v}_n \cdot \mathbf{n} = -5,13 \cdot 10^{-5} \text{ m s}^{-1}, \quad \text{na } \Gamma_1, \quad (5.5a)$$

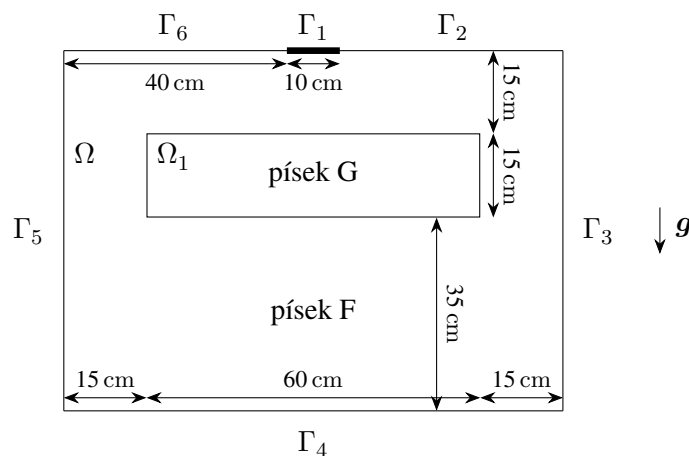
$$\mathbf{v}_w \cdot \mathbf{n} = 0, \quad \mathbf{v}_n \cdot \mathbf{n} = 0, \quad \text{na } \Gamma_3 \cup \Gamma_4 \cup \Gamma_5, \quad (5.5b)$$

$$p_w = p_{ref} - y\rho_w g, \quad S_n = 0, \quad \text{na } \Gamma_2 \cup \Gamma_6, \quad (5.5c)$$

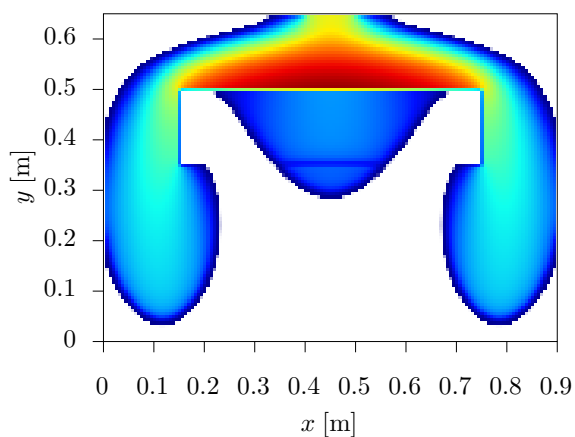
kde $p_{ref} = 10^5 \text{ Pa}$ je referenční hodnota tlaku a y označuje výšku nad spodní stranou oblasti Ω .

Výsledky podobně formulované úlohy byly publikovány v literatuře [2, 9, 19, 20]. V rámci knihovny TNL použité pro implementaci paralelního řešiče jsme omezeni na použití ekvidistantní čtvercové sítě, což by znamenalo nutnost použití velmi jemné sítě pro přesné zarovnání elementů vzhledem k rozměrům vnitřní oblasti Ω_1 a vstupní části hranice Γ_1 . Proto jsme úlohu modifikovali a rozměry zaokrouhlili tak, aby bylo možné použít hrubší ekvidistantní síť. Do budoucna bude předmětem vývoje obecná síť.

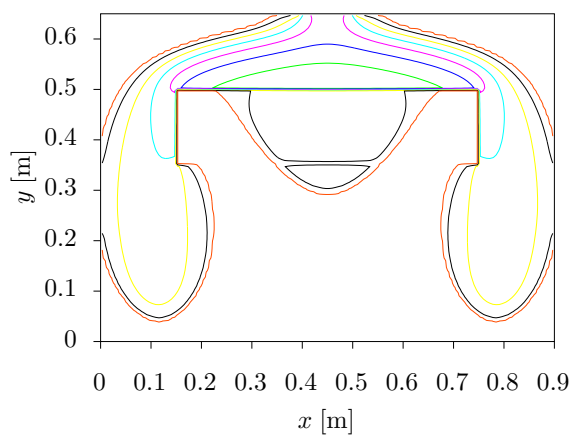
Úlohu řešíme s využitím Brooksova a Coreyho modelu (1.12) a schématu využívajícího metodu mass-lumping. Řešení první varianty s pískem G_1 na sítích s 144×104 , 288×208 a 576×416 elementy jsou zobrazena na obr. 5.11. Ve zvoleném konečném čase $T = 7000 \text{ s}$ projde nesmáčivá fáze přes vnitřní oblast několik centimetrů za její dolní okraj, větší část hmoty nesmáčivé fáze však překážku obteče. Ve druhé variantě s pískem G_2 zobrazené na obr. 5.12 se vnitřní oblast chová jako nepropustná překážka a nesmáčivá fáze ji pouze obtéká. Čelo řešení je dobře zachyceno i na nejhrubší síti, na jemnějších sítích řešení v obou variantách vizuálně konverguje. Přesné srovnání s publikovanými výsledky však zatím není možné kvůli odlišným rozměrům oblasti.



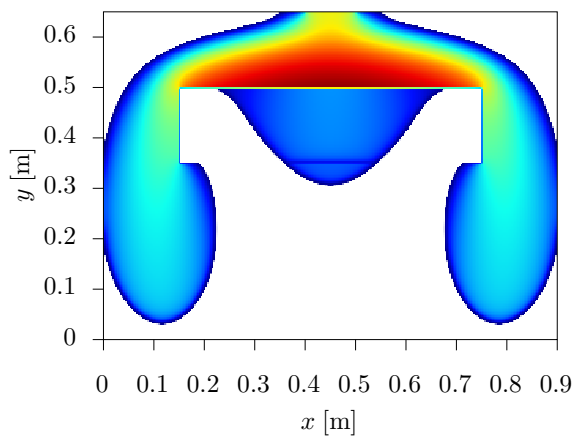
Obrázek 5.10: Geometrická konfigurace úlohy obtékání překážky.



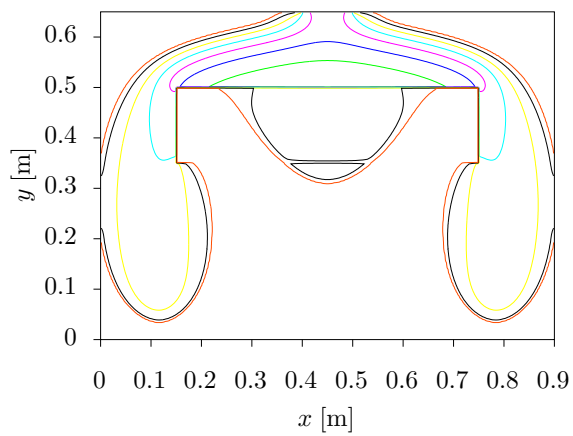
(a) 144×104 elementů



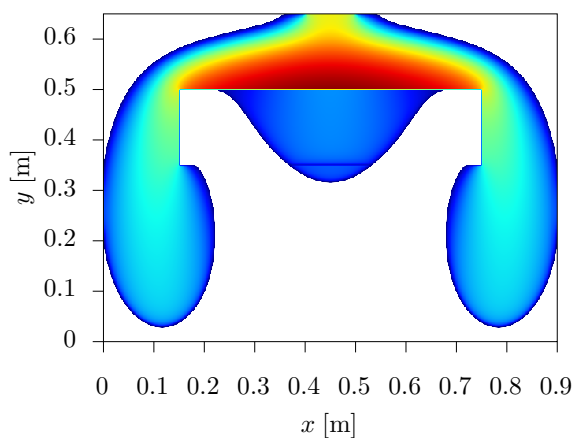
(b) 144×104 elementů



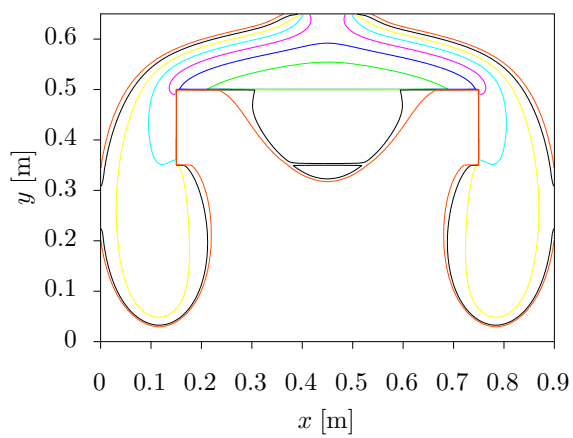
(c) 288×208 elementů



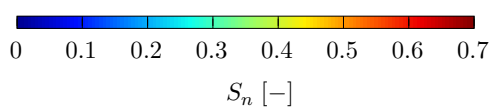
(d) 288×208 elementů



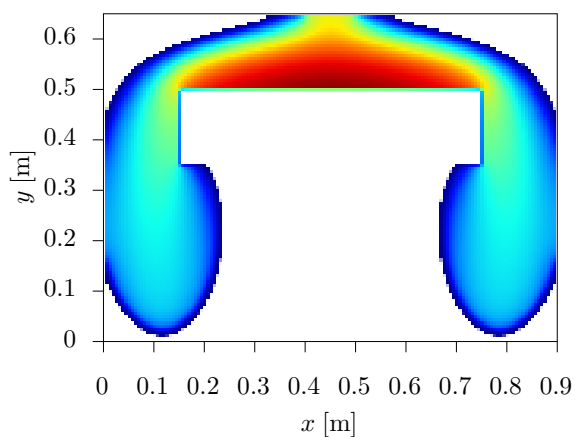
(e) 576×416 elementů



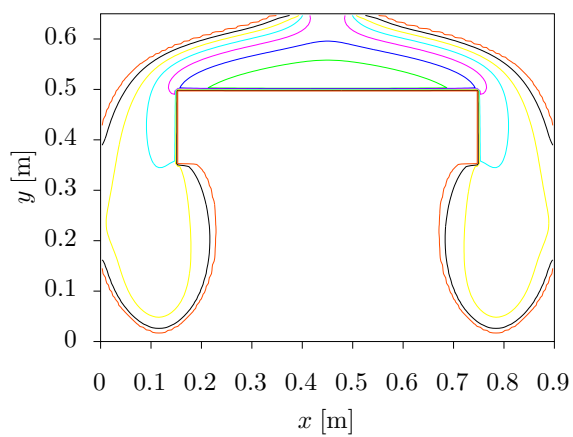
(f) 576×416 elementů



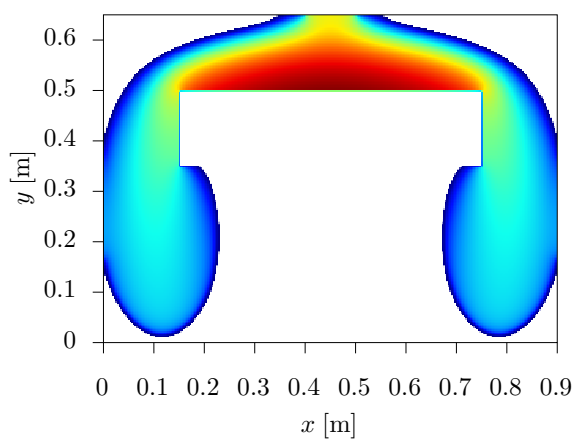
Obrázek 5.11: Řešení úlohy obtékání propustné překážky v čase $T = 7000$ s.



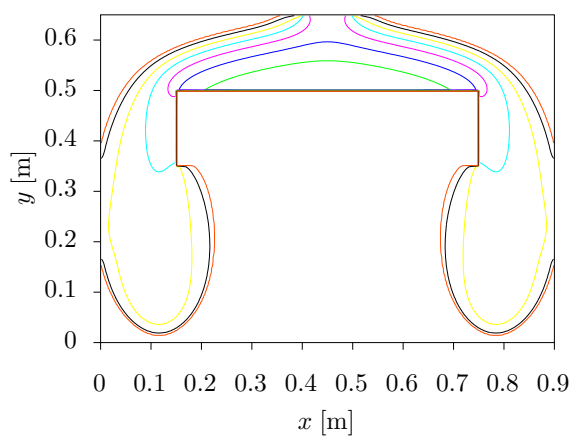
(a) 144×104 elementů



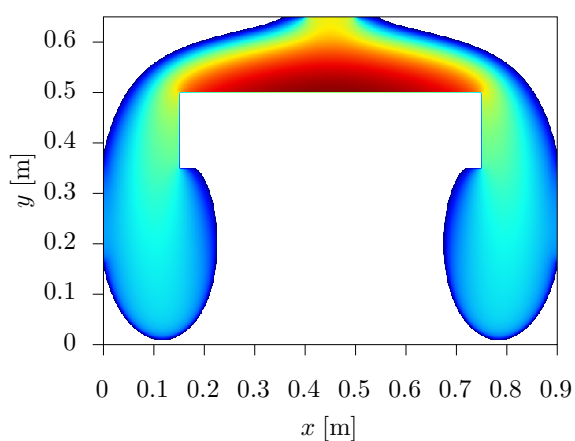
(b) 144×104 elementů



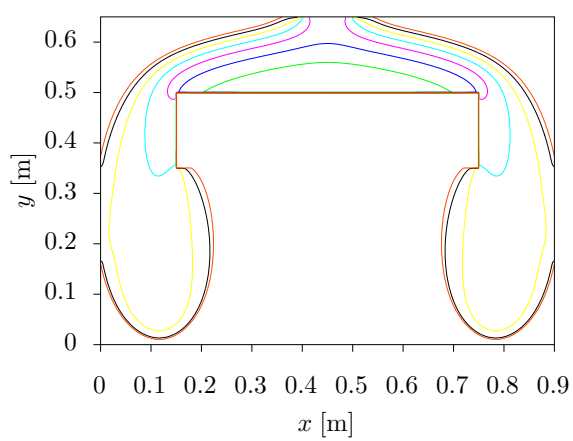
(c) 288×208 elementů



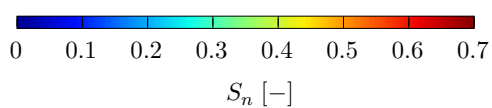
(d) 288×208 elementů



(e) 576×416 elementů



(f) 576×416 elementů



Obrázek 5.12: Řešení úlohy obtékání nepropustné překážky v čase $T = 7000$ s.

5.6 Shrnutí numerických výsledků

V této kapitole bylo popsáno několik testovacích úloh, které byly použity k ověření konvergence numerického schématu a porovnání efektivity paralelní implementace na GPU. Pomocí experimentální analýzy konvergence pro difúzní úlohy v homogenním a heterogenním prostředí se známým semi-analytickým řešením bylo ukázáno, že použité schéma konverguje s přibližně prvním řádem přesnosti. Výsledky advekčně-difúzní úlohy v heterogenním prostředí simulující bariérový efekt odpovídají výsledkům publikovaným v literatuře, a to s využitím modelů kapilarity dle Brookse a Coreya i dle van Genuchtena. V této úloze se ukázala nutnost použití metody mass-lumping pro potlačení oscilací při přechodu nesmáčivé fáze přes materiálová rozhraní.

Efektivita paralelní implementace byla podrobně zkoumána na příkladu McWhorterovy–Sunadovy úlohy, pro ostatní úlohy očekáváme v závislosti na velikosti numerické sítě a časového kroku přibližně stejné výsledky. Urychlení paralelní implementace na GPU oproti sekvenčnímu výpočtu na CPU bylo prokázáno pro dostatečně velké sítě, na nejmenější síti s 1280×320 elementy (tj. 1 641 600 stupňů volnosti) je urychlení $14\times$ při použití metody GMRES s ortogonalizací pomocí modifikovaného Gramova–Schmidtova procesu s reortogonalizací. Dále byla navržena modifikace metody GMRES využívající ortogonalizaci pomocí Householderových transformací a kompaktní WY reprezentaci, přičemž urychlení paralelní implementace je $17\times$ oproti sekvenční variantě modifikovaného Gramova–Schmidtova procesu. Paralelizace řešiče na GPU tedy umožňuje řešit velké úlohy řádově rychleji oproti CPU.

Vzhledem k náročnosti výpočtu dané úlohy na CPU již není reálně možné porovnat doby výpočtu se stejným nastavením řešiče na ještě jemnější síti. Velkou nepříjemností je nutnost volit časový krok konstantní po celou dobu simulace. Efektivitu výpočtu jsme zkoumali jen na příkladě difúzní úlohy, kde bylo možné provést sekvenční výpočet i na poměrně jemných sítích. Advekčně-difúzní úlohy mají z pohledu volby časového kroku ještě horší vlastnosti, pro zpřístupnění velmi jemných sítí i pro řešení úloh z praxe je potřeba implementovat adaptivní volbu časového kroku, přičemž poměr mezi dobami výpočtu na CPU a GPU při stejném nastavení řešiče by měl zůstat přibližně stejný jako v případě McWhorterovy–Sunadovy úlohy s konstantním časovým krokem. Další možností, jak zrychlit výpočet, je použití efektivnějšího předpodmínění. V této práci jsme pro veškeré výpočty používali Jacobiho předpodmínění, které je sice triviální pro paralelizaci, ale není dostatečně účinné. Jedním z dalších úkolů proto bude implementace efektivnějšího předpodmínění pro zkrácení celkové doby výpočtu.

Závěr

V této práci jsme se zabývali matematickým modelováním dvoufázového nemísivého proudění v porézním prostředí a implementací paralelního řešiče na GPU. V první kapitole věnující se popisu problematiky zkoumaného problému je zformulována soustava parciálních diferenciálních rovnic popisující dvoufázové proudění v porézním prostředí a jsou zde popsány běžně užívané modely kapilarity.

Numerické schéma je odvozeno pro soustavu parciálních diferenciálních rovnic s obecnými koeficienty, kterou lze použít i pro vyjádření obecných úloh vícefázového kompozičního proudění. Pro sestavení semi-implicitního schématu jsme využili kombinaci hybridní metody smíšených konečných prvků a metody konečných objemů. Pro zlepšení numerických vlastností schématu byla dále použita upwindová stabilizace pro advekční úlohy a metoda mass-lumping, která se ukazuje jako velmi důležitá při řešení úloh v heterogenním prostředí. Schéma odvozené v obecné prostorové dimenzi bylo v práci použito pro řešení úloh v 1D a ve 2D pomocí strukturované obdélníkové sítě.

Numerické schéma bylo použito pro implementaci masivně paralelního řešiče na GPU s využitím platformy CUDA a knihovny TNL. Moderní architektury GPU disponují tisíci výpočetními jednotkami a rychlou pamětí. Efektivní využití potenciálu dané architektury bylo netriviální zejména z hlediska efektivního využití hierarchie pamětí. Díky použití knihovny TNL bylo možné vytvořit program umožňující spustit výpočet jak na CPU, tak i masivně paralelně na GPU, což je velmi užitečné pro porovnání urychlení a analýzu efektivity výpočtu.

V páté kapitole bylo popsáno několik testovacích úloh, které byly použity k ověření konvergence numerického schématu a porovnání efektivity paralelní implementace na GPU. Pomocí experimentální analýzy konvergence pro difúzní úlohy v homogenním a heterogenním prostředí se známým semi-analytickým řešením bylo ukázáno, že použité schéma konverguje s přibližně prvním řádem přesnosti. Výsledky advekčně-difúzní úlohy v heterogenním prostředí simulující bariérový efekt odpovídají výsledkům publikovaným v literatuře, a to s využitím modelů kapilarity dle Brookse a Coreya i dle van Genuchtena. V této úloze se ukázala nutnost použití metody mass-lumping pro potlačení oscilací při přechodu nesmáčivé fáze přes materiálová rozhraní.

Efektivita paralelní implementace byla podrobně zkoumána na příkladu McWhorterovy–Sunadovy úlohy. Urychlení paralelní implementace na GPU oproti sekvenčnímu výpočtu na CPU bylo prokázáno pro dostatečně velké sítě. Dle očekávání zabírá největší část výpočetního času řešení soustav lineárních algebraických rovnic s řídkou maticí pomocí metody GMRES. Standardní implicitně paralelizovaná verze metody GMRES využívající modifikovaný Gramův–Schmidtův ortogonalizační proces je na nejjemnější síti s 1280×320 elementy (tj. 1 641 600 stupňů volnosti) $14\times$ rychlejší oproti sekvenční variantě. Navržená modifikace metody GMRES využívající ortogonalizaci pomocí Householderových transformací a kompaktní WY reprezentace poskytuje lepší možnosti paralelizace, na síti 1280×320 je paralelní výpočet $17\times$ rychlejší oproti stejnému referenčnímu sekvenčnímu algoritmu. Paralelizace řešiče na GPU tedy umožňuje řešit velké úlohy řádově rychleji oproti CPU.

Hlavním cílem do budoucna bude další optimalizace iteračního řešiče, např. s využitím účinnějšího předpokládání. Celkový výpočetní čas by také bylo možné zkrátit pomocí adaptivní volby časového kroku. V neposlední řadě bude jedním z budoucích cílů využít obecnou formulaci numerického schématu pro implementaci řešiče pro obecný model vícefázového kompozičního proudění v porézním prostředí.

Dodatek A

Parametry porézních materiálů

Parametr			Písek A	Písek B	Písek C
Porozita	Φ	$[-]$	0,448	0,418	0,343
Vnitřní propustnost	K	$[m^2]$	$1,631 \cdot 10^{-11}$	$1,437 \cdot 10^{-11}$	$5,168 \cdot 10^{-12}$
Reziduální saturace	S_{wr}	$[-]$	0,265	0,037	0,04
Reziduální saturace	S_{nr}	$[-]$	0	0	0
Brooksův a Coreyho parametr	λ	$[-]$	4,660	5,323	2,857
Brooksův a Coreyho parametr	p_d	$[Pa]$	3 450,18	4 041,72	4 605,8
van Genuchtenův parametr	n	$[-]$	—	11,53	6,64
van Genuchtenův parametr	α	$[Pa^{-1}]$	—	$2,14 \cdot 10^{-4}$	$1,71 \cdot 10^{-4}$

Tabulka A.1: Parametry písků s označeními A, B a C. Hodnoty byly převzaty z [9].

Parametr			Písek D	Písek E
Porozita	Φ	$[-]$	0,4	0,39
Vnitřní propustnost	K	$[m^2]$	$5,04 \cdot 10^{-10}$	$5,26 \cdot 10^{-11}$
Reziduální saturace	S_{wr}	$[-]$	0,08	0,10
Reziduální saturace	S_{nr}	$[-]$	0	0
Brooksův a Coreyho parametr	λ	$[-]$	3,86	2,49
Brooksův a Coreyho parametr	p_d	$[Pa]$	370	1 324
van Genuchtenův parametr	n	$[-]$	8,06	5,34
van Genuchtenův parametr	α	$[Pa^{-1}]$	$2,25 \cdot 10^{-3}$	$5,81 \cdot 10^{-4}$

Tabulka A.2: Parametry písků s označeními D a E. Hodnoty byly převzaty z [9].

Parametr			Písek F	Písek G ₁	Písek G ₂
Porozita	Φ	[-]	0,4	0,39	0,39
Vnitřní propustnost	K	[m ²]	$6,64 \cdot 10^{-11}$	$3,32 \cdot 10^{-11}$	$3,32 \cdot 10^{-11}$
Reziduální saturace	S_{wr}	[-]	0,10	0,12	0,12
Reziduální saturace	S_{nr}	[-]	0	0	0
Brooksův a Coreyho parametr	λ	[-]	2,7	2,0	2,0
Brooksův a Coreyho parametr	p_d	[Pa]	755	1 163,5	1 466,1
van Genuchtenův parametr	n	[-]	—	—	—
van Genuchtenův parametr	α	[Pa ⁻¹]	—	—	—

Tabulka A.3: Parametry písků s označeními F, G₁ a G₂. Hodnoty byly převzaty z [9].

Literatura

- [1] P. Bastian, *Numerical computation of multiphase flow in porous media*, Habilitation Thesis, Univeristät Kiel, 1999.
- [2] P. Bastian and R. Helmig, *Efficient fully-coupled solution techniques for two-phase flow in porous media: parallel multigrid solution and large scale computations*, *Advances in Water Resources* 23.3 (1999), pages 199–216.
- [3] F. Brezzi and M. Fortin, *Mixed and hybrid finite elements methods*, Springer series in computational mathematics, Springer-Verlag, 1991, ISBN: 9781461278245.
- [4] R. H. Brooks and A. T. Corey, *Hydraulic properties of porous media and their relation to drainage design*, *Transactions of the ASAE* 7.1 (1964), pages 26–0028.
- [5] N. T. Burdine, *Relative permeability calculations from pore size distribution data*, *Journal of Petroleum Technology* 5.03 (1953), pages 71–78.
- [6] N. T. Burdine, L. S. Gournay, and P. P. Reichertz, *Pore size distribution of petroleum reservoir rocks*, *Journal of Petroleum Technology* 2.07 (1950), pages 195–204.
- [7] H. Darcy, *Les fontaines publiques de la ville de Dijon*, Victor Dalmont, Paris, 1856.
- [8] C. J. van Duijn and M. J. De Neef, *Self-similar profiles for capillary diffusion driven flow in heterogeneous porous media*, Centrum voor Wiskunde en Informatica (1996).
- [9] R. Fučík, *Advanced Numerical Methods for Modelling Two-Phase Flow in Heterogeneous Porous Media*, Dissertation Thesis, FNSPE, CTU in Prague, 2010.
- [10] R. Fučík, *Online implementation of exact solution for two-phase flow in porous medium with a material discontinuity*, URL: <http://mmg.fjfi.cvut.cz/~fucik/exact1d>.
- [11] R. Fučík, *Online implementation of McWhorter and Sunada exact solution*, URL: <http://mmg.fjfi.cvut.cz/~fucik/exact1d>.
- [12] R. Fučík, T. H. Illangasekare, and M. Beneš, *Multidimensional self-similar analytical solutions of two-phase flow in porous media*, *Advances in Water Resources* 90 (2016), pages 51–56.
- [13] R. Fučík and J. Mikyška, *Discontinuous Galerkin and Mixed-Hybrid Finite Element Approach to Two-Phase Flow in Heterogeneous Porous Media with Different Capillary Pressures*, *Procedia Computer Science* 4 (2011), pages 908–917.
- [14] R. Fučík and J. Mikyška, *Mixed-hybrid finite element method for modelling two-phase flow in porous media*, *Journal of Math-for-Industry* 3.2 (2011), pages 9–19.
- [15] R. Fučík, J. Mikyška, M. Beneš, and T. H. Illangasekare, *An improved semi-analytical solution for verification of numerical models of two-phase flow in porous media*, *Vadose Zone Journal* 6.1 (2007), pages 93–104.
- [16] R. Fučík, J. Mikyška, M. Beneš, and T. H. Illangasekare, *Semianalytical solution for two-phase flow in porous media with a discontinuity*, *Vadose Zone Journal* 7.3 (2008), pages 1001–1007.
- [17] M. T. van Genuchten, *A closed-form equation for predicting the hydraulic conductivity of unsaturated soils*, *Soil science society of America journal* 44.5 (1980), pages 892–898.

- [18] L. Giraud, J. Langou, and M. Rozloznik, *The loss of orthogonality in the Gram-Schmidt orthogonalization process*, Computers & Mathematics with Applications 50.7 (2005), pages 1069–1075.
- [19] R. Helmig, *Multiphase flow and transport processes in the subsurface: a contribution to the modeling of hydrosystems*. Springer-Verlag, 1997.
- [20] R. Helmig and R. Huber, *Comparison of Galerkin-type discretization techniques for two-phase flow in heterogeneous porous media*, Advances in Water Resources 21.8 (1998), pages 697–711.
- [21] H. Hoteit and A. Firoozabadi, *Numerical modeling of two-phase flow in heterogeneous permeable media with different capillarity pressures*, Advances in Water Resources 31.1 (2008), pages 56–73.
- [22] H. Ishigami, K. Kimura, and Y. Nakamura, “Implementation and performance evaluation of new inverse iteration algorithm with Householder transformation in terms of the compact WY representation”, *Proc*, 2011, pages 775–781.
- [23] R. J. LeVeque, *Finite volume methods for hyperbolic problems*, volume 31, Cambridge university press, 2002.
- [24] D. B. McWhorter and D. K. Sunada, *Exact integral solutions for two-phase flow*, Water Resources Research 26 (1990), pages 399–413.
- [25] A. Monakov, A. Lokhmotov, and A. Avetisyan, *Automatically tuning sparse matrix-vector multiplication for GPU architectures*, High Performance Embedded Architectures and Compilers, Springer, 2010, pages 111–125.
- [26] Y. Mualem, *A new model for predicting the hydraulic conductivity of unsaturated porous media*, Water resources research 12.3 (1976), pages 513–522.
- [27] NVIDIA, *CUDA Toolkit Documentation, version 7.5*, 2015, URL: <http://docs.nvidia.com/cuda/index.html>.
- [28] Y. Saad, *Iterative methods for sparse linear systems*, SIAM, 2003, ISBN: 0898715342.
- [29] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on scientific and statistical computing 7.3 (1986), pages 856–869.
- [30] R. Schreiber and C. Van Loan, *A storage-efficient WY representation for products of Householder transformations*, SIAM Journal on Scientific and Statistical Computing 10.1 (1989), pages 53–57.
- [31] O. Tange, *GNU Parallel - The Command-Line Power Tool*, The USENIX Magazine 36.1 (Feb. 2011), pages 42–47, DOI: 10.5281/zenodo.16303, URL: <http://www.gnu.org/s/parallel>.
- [32] H. F. Walker, *Implementation of the GMRES method using Householder transformations*, SIAM Journal on Scientific and Statistical Computing 9.1 (1988), pages 152–163.
- [33] T. Williams, C. Kelley, and many others, *Gnuplot 5.0: an interactive plotting program*, Jan. 2015, URL: <http://gnuplot.sourceforge.net/>.
- [34] Y. Yamamoto and Y. Hirota, *A parallel algorithm for incremental orthogonalization based on the compact WY representation*, JSIAM Letters 3 (2011), pages 89–92.
- [35] A. Younes, P. Ackerer, and F. Lehmann, *A new mass lumping scheme for the mixed hybrid finite element method*, International journal for numerical methods in engineering 67.1 (2006), pages 89–107.