



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MATEMATIKY

INSTITUTE OF MATHEMATICS

**SIMULACE PROUDĚNÍ TEKUTINY OKOLO PŘEKÁŽEK
LATTICE BOLTZMANNOVOU METODOU**

SIMULATION OF FLUID FLOW AROUND OBSTACLES BY LATTICE BOLTZMANN METHOD

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Ing. BSc František Prinz

VEDOUCÍ PRÁCE

SUPERVISOR

Mgr. Jitka Zatočilová, Ph.D.

BRNO 2020

Zadání diplomové práce

Ústav:	Ústav matematiky
Student:	Ing. BSc František Prinz
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Matematické inženýrství
Vedoucí práce:	Mgr. Jitka Zatočilová, Ph.D.
Akademický rok:	2019/20

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Simulace proudění tekutiny okolo překážek Lattice Boltzmannovou metodou

Stručná charakteristika problematiky úkolu:

Lattice Boltzmannova metoda, která je používána pro numerické modelování proudění tekutin, byla původně odvozena z metody buněčných automat. Později bylo ukázáno, že ji lze odvodit i z kinetické teorie a z Boltzmannovy transportní rovnice. Při odvozování LBM hraje důležitou roli Chapmanův–Enskogův rozvoj díky němuž lze ukázat konvergenci (či aspoň asymptotické přiblížení) získaného řešení k řešení určitých parciálních diferenciálních rovnic, kterými jsou nejčastěji Navierovy–Stokesovy rovnice. Velkou výhodou této metody je relativně jednoduchá implementace a možnost paralelizace výpočtu.

Cíle diplomové práce:

Popis a odvození Lattice Boltzmannovy metody.

Naprogramování Lattice Boltzmannovy metody a její otestování na vybraných úlohách, především pak při simulaci proudění tekutiny okolo překážek v kanálu.

Srovnání dosažených výsledků s výsledky získanými jinou numerickou metodou.

Seznam doporučené literatury:

KRÜGER, T., KUSUMAATMAJA, H., KUZMIN, A., SHARDT, O., SILVA, G., VIGGEN, E. M. The lattice Boltzmann method: principles and practice. New York, NY: Springer Berlin Heidelberg, 2016. ISBN 978-3-319-44647-9.

GUO, Z., SHU, Ch. Lattice Boltzmann method: and its applications in engineering. Hackensack, NJ: World Scientific, c2013.

MOHAMAD, A. A. Lattice Boltzmann method. Fundamentals and engineering applications with computer codes. Springer, London, 2011. ISBN: 978-0-85729-454-8.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2019/20

V Brně, dne

L. S.

prof. RNDr. Josef Šlapal, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Tato práce se zabývá Lattice Boltzmannovou metodou (LBM). Jedná se o mezoskopickou metodu popisující pohyb částic v tekutině pomocí Boltzmannovy rovnice, kde figuruje pravděpodobnostní rozdělovací funkce. Pomocí Chapman-Enskogova rozvoje lze ukázat za využití Hermitových polynomů propojení této rovnice s Navier-Stokesovými rovnicemi zachování makroskopických veličin. Diskretizací rychlosti, prostoru a času je odvozena Lattice Boltzmannova rovnice a příslušný numerický algoritmus. Ten je realizován na úlohách proudění dvou-rozměrné kavity a obtékání překážek. V obou případech byly vypočtené hodnoty rychlostí porovnávány s metodou konečných objemů (FVM) za dosažení hodnot relativních odchylek v řádu jednotek %.

KLÍČOVÁ SLOVA

Rozdělovací funkce, Boltzmannova rovnice, Chapman-Enskogův rozvoj, Hermitovy polynomy, diskretizace, kolizní operátor BGK, rychlostní set D2Q9.

ABSTRACT

The task of this diploma thesis is the Lattice Boltzmann method (LBM). LBM is a mesoscopic method describing the particle motion in a fluid by the Boltzmann equation, where the distribution function is involved. The Chapman-Enskog expansion shows the connection with the macroscopic Navier-Stokes equations of conservation laws. In this process the Hermite polynomials are used. The Lattice Boltzmann equation is derived by the discretisation of velocity, space and time which is concluding to the numerical algorithm. This algorithm is applied at two problems of fluid flow: the two-dimensional square cavity and a flow around obstacles. In both cases were the results of velocities compared to results calculated by finite volume method (FVM). The relative errors are in order of multiple 1 %.

KEYWORDS

Distribution function, Boltzmann equation, Chapman-Enskog analysis, Hermite polynomials, discretisation, collision operator BGK, velocity set D2Q9.

PRINZ, František. *Simulace proudění tekutiny okolo překážek Lattice Boltzmannovou metodou*. Brno, 2019, 114 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav matematiky. Vedoucí práce: Mgr. Jitka Zatočilová, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Simulace proudění tekutiny okolo překážek Lattice Boltzmannovou metodou“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora

PODĚKOVÁNÍ

Nejprve bych rád poděkoval vedoucí diplomové práce paní Mgr. Jitce Zatočilové, Ph.D. za věnovaný čas, cenné připomínky a návrhy k práci a zejména za trpělivost.

Dále patří velký dík všem kolegům na odboru termomechaniky a techniky prostředí za realizování simulace metodou konečných objemů pomocí softwaru starCCM+ a za další tipy a podněty k práci.

Také bych chtěl poděkovat moji rodině, která mně po celou dobu mého studia podporovala. V neposlední řadě bych chtěl zmínit mimořádnou situaci na jaře letošního akademického roku způsobenou celosvětovou koronavirovou krizí, načež došlo k přechodu na distanční výuku a odložení termínů odevzdání této práce asi o měsíc, což mi dodalo dostatek času se práci věnovat.

Brno

.....

podpis autora

Obsah

Úvod	17
I Teorie	19
1 Rovnice zachování makroskopických veličin	21
2 Kinetická teorie	25
2.1 Kolizní operátor	27
2.1.1 Kolizní invarianty	31
2.2 Entropie, H-teorém	32
2.2.1 Rovnovážný stav	33
2.3 Operátor BGK	34
2.4 Propojení s rovnicemi zachování makroskopických veličin	35
2.5 Přejchod do bezrozměrné soustavy	36
3 Hermitovy polynomy	39
4 Chapman Enskogův rozvoj	43
5 Gauss-Hermitova kvadrurní formule	49
5.1 Diskrétní rychlostní sety	52
6 Časová a prostorová diskretizace	59
7 Počáteční a okrajové podmínky	63
7.1 Okrajové podmínky v LBM	64
7.2 Počáteční podmínky v LBM	68
II Aplikace	71
8 O simulaci obecně	75
8.1 Přejvod do bezrozměrných jednotek a volba parametrů	75
8.2 Algoritmus výpočtu	78
9 Kavita	79
9.1 Fyzikální popis problému	79
9.2 Simulace	79
9.3 Výsledky	79

10 Obtékání překážek	85
10.1 Fyzikální popis problému	85
10.2 Simulace	85
10.3 Výsledky	86
Závěr	93
Literatura	95
Seznam použitých zkratk a symbolů	97
Seznam příloh	99
A Zdrojové kody	101
A.1 Kavita	101
A.2 Obtékání překážek	106

Seznam obrázků

2.1	Částice v kvádru, černě vyznačené mají kladnou rychlost ξ_x (převzato z [8]).	26
2.2	Grafické znázornění kolize dvou tuhých koulí o průměru d (převzato z [12]).	29
2.3	Grafické znázornění kolize částice s rychlostí $-\mathbf{g}$ v daném prostorovém segmentu kolidující s částicí ve středu výseče (převzato z [12]).	30
5.1	Grafické znázornění rychlostního setu D1Q3 (převzato z [8]).	53
5.2	Grafické znázornění rychlostního setu D2Q9, (převzato z [8]).	54
5.3	Znázornění vybraných rychlostních trojrozměrných setů, (převzato z [8]).	56
7.1	Druhy uzlů v LBM na příkladu proudění v trubce, (převzato z [8], upraveno).	64
7.2	Rozdíl mezi typy okrajových uzlů link-wise (a) a wet-node (b) (převzato z [8]).	65
7.3	Zjednodušené znázornění metody bounce-back.	66
7.4	Porovnání techniky fullway (a) a halfway (b), (převzato z [8], upraveno).	67
9.1	Znázornění rychlostního pole kavity v čase 0,15 s.	80
9.2	Znázornění rychlostního pole kavity v čase 0,3 s.	81
9.3	Znázornění rychlostního pole kavity v čase 0,6 s.	82
9.4	Znázornění rychlostního pole kavity v čase 1,5 s.	83
9.5	Verifikace modelu pro x složku rychlosti na řezu $y = 0,10$ m s daty z [1].	84
9.6	Verifikace modelu pro y složku rychlosti na řezu $x = 0,10$ m s daty z [1].	84
10.1	Grafické znázornění celkové geometrie.	85
10.2	Grafické znázornění geometrie - detail překážek, rozměry v mm.	86
10.3	Znázornění rychlostního pole, absolutní rychlost.	87
10.4	Znázornění rychlostního pole, detail předních překážek, absolutní rychlost.	87
10.5	Znázornění rychlostního pole, detail zadních překážek, absolutní rychlost.	88
10.6	Srovnání rychlostí u_x na příčném řezu $x = 0,05$ m.	89
10.7	Srovnání rychlostí u_x na příčném řezu $x = 0,075$ m.	90
10.8	Srovnání rychlostí u_x na příčném řezu $x = 0,1$ m.	90
10.9	Srovnání rychlostí u_y na příčném řezu $x = 0,05$ m.	91
10.10	Srovnání rychlostí u_y na příčném řezu $x = 0,075$ m.	91
10.11	Srovnání rychlostí u_y na příčném řezu $x = 0,1$ m.	92

Seznam tabulek

5.1	Běžně používané rychlostní sety.	57
7.1	Opačné rychlosti pro diskrétní set D2Q9	66

Úvod

Lattice Boltzmannova metoda (LBM) je mezoskopická numerická metoda umožňující modelování proudění tekutin. Spočívá v diskretizaci Boltzmannovy rovnice, která je základní rovnicí kinetické teorie plynu a popisuje pravděpodobnostní pohyb částic, ze kterých se tekutina skládá.

Předchůdce LBM jsou mřížkové buněčné automaty pro plyny (LGA) vyvinuté trojicí Hardy, Pomeau a de Pazzis v roce 1973, které popisovaly proudění částic plynu pomocí Boolovských proměnných dodržující daná pravidla. Tyto modely však nesplňovaly fyzikální zákony jako například Galileův princip relativity a vykazovaly mnoho numerických či statistických šumů. Výrazný posun přinesl rok 1986, kdy trojice vědců Frisch, Hasslacher a Pomeau odvodila při použití šestiúhelníkové mřížky konvergenci k Navier-Stokesovým rovnicím [10]. V 90. letech došlo k dalšímu zjednodušení rovnice zavedením linearizovaného kolizního operátoru Bhatnagar-Gross-Krook (BGK) a k odvození Lattice Boltzmannovy rovnice, která se v této podobě používá k simulacím dodnes. S rozvojem digitální techniky se výpočetní možnosti zvyšovaly a počátkem 21. století vznikají první softwary (např.: Open LB [7], Palabos). Metoda se úspěšně používá v mnoha aplikacích fyziky, chemie a různých inženýrských oborech.

LBM lze použít pro simulace slabě stlačitelného proudění do Machova čísla 0,30 na úrovni Large Eddy Simulations (LES). Výhodou oproti klasickým CFD řešičům jako např. metoda konečných objemů (FVM) je relativně jednoduchá paralelizace výpočtu na vícejádrových procesorech či grafických kartách díky lokálním vlastnostem Lattice Boltzmannovy rovnice, která umožňuje rychlý a efektivní výpočet [6]. Další výhodou je snadná tvorba geometrie, kdy numerická síť je rovnoměrná a oproti ostatním CFD simulátorům ji lze vygenerovat automaticky v rámci výpočetního softwaru. Umožňuje dokonce výpočty porózních médií či simulace proudění tekutin vícero složek nebo fází, tudíž je použitelná pro řadu dalších aplikací.

Práce je rozdělena na dvě části. V první teoretické části je Lattice Boltzmannova metoda odvozena. Nejprve je uvedena Boltzmannova rovnice s příslušným kolizním operátorem. Pro numerické řešení je Boltzmannův kolizní operátor aproximován operátorem BGK, který byl používán u simulací v této práci. Chapman Enskogovým rozvojem je ukázáno propojení s Navier-Stokesovými pohybovými rovnicemi, popisující tekutiny z makroskopického hlediska [8]. Boltzmannova rovnice je následně diskretizována volbou rychlostního setu, poté také prostorově a časově. To vede na Lattice Boltzmannovu rovnici, která je vhodná k numerické implementaci.

Druhá část práce se zabývá aplikací metody na konkrétních problémech. Po úvodních zásadách implementace Lattice Boltzmannovy metody jsou uvedeny simulace dvou fyzikálních aplikací: problém dvourozměrné kavity a obtékání překážkové mříže. Výsledky obou simulací jsou porovnány a analyzovány se simulací provedenou klasickou metodou konečných objemů.

Část I

Teorie

1 Rovnice zachování makroskopických veličin

Dříve než bude uvedeno odvození samotné Boltzmannovy rovnice, zmíníme se nejprve o zákonech zachování makroskopických veličin. Zákony zachování označují neměnnost určitých fyzikálních veličin při časovém vývoji uvažovaného systému. Jedná se o zákon zachování hmotnosti, hybnosti a energie. V tomto textu se omezíme na výsledné vztahy, detailní odvození lze najít např. v [5]. Nadále budeme uvažovat třírozměrný prostor, pro ostatní prostory lze rovnice odvodit analogicky. Všechny vektory, nebude-li uvedeno jinak, mají tedy 3 složky a zavedeme pro ně následující značení:

$$\mathbf{u} = (u_1, u_2, u_3)^T.$$

Budeme také používat termín kontrolní objem V , který označuje objem části tekutiny tvořené stále stejnými částicemi.

Zákon zachování hmotnosti říká, že hmota nemůže vzniknout ani zaniknout, tedy že hmotnost tekutiny v kontrolním objemu V se nemění. Matematicky lze uvedené zapsat vztahem nazývaným rovnice kontinuity, který platí pro libovolný bod \mathbf{x} kontrolního objemu V v libovolném čase t :

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \quad (1.1)$$

kde $\rho(\mathbf{x}, t)$ je hustota, $\mathbf{u}(\mathbf{x}, t)$ je makroskopická rychlost tekutiny a

$$\nabla = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3} \right)^T.$$

Dalším zákonem je **zákon zachování hybnosti**. Hybnost $\mathbf{p}(\mathbf{x}, t)$ je definovaná vztahem:

$$\mathbf{p} = \rho \mathbf{u}.$$

Hmota zůstává v klidu nebo rovnoměrném přímočarém pohybu, nepůsobí-li na ni žádné síly. Působící síly vyvolají změnu hybnosti dle druhého Newtonova zákona. Zákon zachování hybnosti pak lze formulovat následovně:

Časová změna hybnosti tekutiny v kontrolním objemu V je rovna síle, která na kontrolní objem působí, zapsáno v diferenciálním tvaru:

$$\frac{d\mathbf{p}}{dt} = \mathbf{f}, \quad (1.2)$$

kde $\mathbf{f}(\mathbf{x}, t)$ je výsledná síla působící na kapalinu v daném bodě vztahovaná na jednotku objemu. Vztah (1.2) lze upravit na tvar zvaný Cauchyho rovnice dynamické rovnováhy:

$$\frac{D(\rho \mathbf{u})}{Dt} = \text{div} \boldsymbol{\sigma} + \mathbf{F}, \quad (1.3)$$

kde

$$\boldsymbol{\sigma} = \sigma_{ij}, \quad i, j = 1, 2, 3,$$

je tenzor napětí, kterým jsou charakterizovány smykové a normálové síly v tekutině, $F(\mathbf{x}, t)$ jsou objemové síly, div je divergence a $\frac{D}{Dt}$ je materiálová derivace, pro kterou platí:

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla.$$

Existuje více modelů, které tření tekutin popisují, my se zde omezíme na klasický model Newtonovské tekutiny, který je charakterizován přímou úměrností rychlosti deformace a tenzoru napětí, což odpovídá lineárním závislostem na prostorových derivacích rychlostí [2]. K tomu je potřeba také připočíst tlakové působení okolní kapaliny $p(\mathbf{x}, t)$ a tenzor napětí pak vypadá následovně:

$$\sigma_{ij} = -p\delta_{ij} + \eta \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \zeta \delta_{ij} \frac{\partial u_k}{\partial x_k}, \quad i, j, k = 1, 2, 3, \quad (1.4)$$

kde η a ζ jsou první a druhý koeficient viskozity a δ_{ij} je Kroneckerovo delta a použili jsme (u indexu k) Einsteinovu sumační konvenci, která, nebude-li uvedeno jinak, bude používána i nadále. Koeficienty viskozity obecně závisí na termodynamických veličinách. Uvažujeme-li tyto koeficienty časově i prostorově konstantní, dostaneme známé Navier-Stokesovy pohybové rovnice:

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \eta \left(\frac{\partial^2 u_i}{\partial x_j \partial x_j} + \frac{\partial^2 u_j}{\partial x_i \partial x_j} \right) + \zeta \frac{\partial^2 u_j}{\partial x_i \partial x_j} + F_i, \quad i, j = 1, 2, 3. \quad (1.5)$$

Pokud tenzor napětí σ_{ij} v (1.4) rozepíšeme na dvě složky následujícím způsobem

$$\sigma_{ij} = \sigma'_{ij} - p\delta_{ij}, \quad (1.6)$$

kde:

$$\sigma'_{ij} = \eta \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right) + \eta_B \delta_{ij} \frac{\partial u_k}{\partial x_k} \quad i, j, k = 1, 2, 3, \quad (1.7)$$

tak objemová viskozita η_B je dána vztahem

$$\eta_B = \frac{2\eta}{3} + \zeta.$$

Po dosazení (1.6) a (1.7) do (1.3) dostáváme Navier-Stokesovy pohybové rovnice ve tvaru:

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \eta \left(\frac{\partial^2 u_i}{\partial x_j \partial x_j} \right) + (\eta_B + \eta/3) \frac{\partial^2 u_j}{\partial x_i \partial x_j} + F_i. \quad (1.8)$$

U jednoatomových plynů objemová viskozita η_B je nulová. V případě plynů proudících nízkými rychlostmi a v případě kapalin lze uvažovat nestlačitelné proudění, pro které platí:

$$\frac{D\rho}{Dt} = 0. \quad (1.9)$$

Dosazením (1.9) do rovnice kontinuity (1.1) dostaneme rovnici kontinuity pro nestlačitelné proudění ve tvaru

$$\nabla \cdot \mathbf{u} = 0$$

a stejně tak Navier-Stokesovy pohybové rovnice (1.5) se pro nestlačitelné proudění zjednoduší na tvar

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \eta \Delta \mathbf{u} + \mathbf{F}, \quad (1.10)$$

kde Laplaceův operátor Δ je definovaný následovně:

$$\Delta = \frac{\partial^2}{\partial x_i \partial x_i}.$$

Zanedbáním veškerého tření uvnitř tekutiny, tj. pro $\eta = 0$, $\xi = 0$ dostaneme nestlačitelné Eulerovy pohybové rovnice:

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \nabla \mathbf{u} \cdot \mathbf{u} = -\nabla p + \mathbf{F}.$$

Posledním ze zákonů zachování, který zde uvedeme, je **zákon zachování energie**. Ten obecně vyjadřuje, že energie nemůže vzniknout ani zaniknout, jen se přeměnit z jedné formy na druhou. Kontrolní objem V nyní představuje uzavřený systém, který si s okolím může vyměňovat energii ve formě tepla nebo práce vykonané působícími silami. Platí tedy, že časová změna celkové energie tekutiny v kontrolním objemu je součtem výkonu sil, kterými obklopující prostředí na tekutinu v kontrolním objemu působí, a výkonu tepla, které je tekutině v kontrolním objemu dodáváno.

Celkovou energii $E(\mathbf{x}, t)$ vztaženou na jednotku objemu můžeme rozdělit na měrnou vnitřní energii $e(\mathbf{x}, t)$ a měrnou kinetickou energii $\mathbf{u}^2/2$, neboli:

$$E = \rho e + \frac{1}{2} \rho \mathbf{u}^2.$$

Zákon zachování energie lze vyjádřit následující rovnicí

$$\frac{\partial E}{\partial t} + \frac{\partial(u_i E)}{\partial x_i} = \frac{\partial(u_j \sigma_{ij})}{\partial x_i} + F_i u_i - \frac{\partial q_i}{\partial x_i} + \rho q, \quad (1.11)$$

kde q_i je plošná hustota tepelného toku a q je měrná objemová hustota tepelného zdroje. Pro celkový popis proudění dané tekutiny je ještě třeba užít stavových rovnic a konstitutivních vztahů. Budeme uvažovat stavovou rovnici ideálního plynu

$$p = \rho RT, \quad (1.12)$$

kde $T(\mathbf{x}, t)$ je teplota tekutiny a R plynová konstanta. Dále kalorimetrickou rovnicí ideálního plynu vyjadřující závislost vnitřní energie na teplotě

$$e = \frac{3}{2} RT \quad (1.13)$$

a Fourierův zákon

$$q_i = -\lambda \frac{\partial T}{\partial x_i}, \quad i = 1, 2, 3, \quad (1.14)$$

kde λ je součinitel tepelné vodivosti.

Nyní přejdeme ke kinetické teorii a Boltzmannově rovnici, která, jak uvidíme, je v souladu s výše uvedenými vztahy.

2 Kinetická teorie

Kinetická teorie vytváří propojení mezi makroskopickými veličinami tekutin (jako je tlak, teplota, hustota atd.) charakterizujícími tekutinu jako kontinuum a mikroskopickým složením tekutiny (tedy atomy či molekuly) a jejich vlastní kinetikou. V dalším se omezíme na tzv. zředěný jednoatomový plyn, což je model tekutiny, pro který platí:

$$t_c \ll t_{mfp},$$

kde t_c je čas srážky částice s jinou částicí a t_{mfp} střední doba pohybu mezi dvěma srážkami. Čas srážky je tedy o dost kratší než čas volného pohybu bez kolizí, což snižuje pravděpodobnost tří a vícečásticových srážek, které v uvedeném modelu nebudeme uvažovat. Mezi jednoatomovými plyny dochází k elastickým srážkám, kdy žádná energie se nepřemění na vnitřní energii atomů (např. rotační či vibrační). Celková energie při srážkách je tedy zachována. Uvedený model může být použit také pro víceatomové plyny při nízkých hustotách.

Základní myšlenkou kinetické teorie tekutin je rozdělení částic v prostoru a v čase. Protože počet částic, které se v kontrolním objemu nachází, je řádově 10^{23} , není možné počítat pohyb každé částice zvlášť, protože by to bylo jednak časově nemožné, ale také proto, že aktuální stav daného systému, tedy pozice a rychlosti každé částice, nelze zjistit. Z toho důvodu se vychází z pravděpodobnostních údajů. Pro identifikaci každé částice z kinematického hlediska je dostačující, známe-li její polohu a rychlost. Tudíž je výhodné pracovat v tzv. fázovém prostoru souřadnic \mathbf{x} a příslušných rychlostí $\boldsymbol{\xi}$ ¹. Pokud budeme uvažovat jednoatomový plyn a trojrozměrný prostor, tak pravděpodobnostní rozdělení částic ve fázovém prostoru je dáno distribuční funkcí

$$P(\mathbf{x}, \boldsymbol{\xi}, t).$$

Střední počet částic nacházející se v malém okolí bodu \mathbf{x} a mající rychlost v malém okolí dané rychlosti $\boldsymbol{\xi}$ rychlostního podprostoru v čase t je dán vztahem

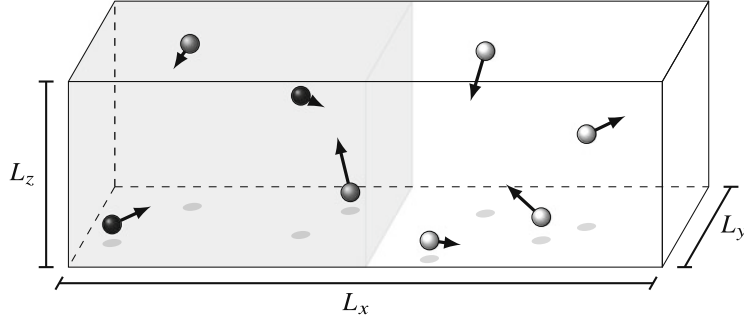
$$P(\mathbf{x}, \boldsymbol{\xi}, t) d\mathbf{x} d\boldsymbol{\xi}.$$

Nyní definujeme střední hustotu částic ve fázovém prostoru nazývanou také rozdělovací funkce f následujícím způsobem:

$$f(\mathbf{x}, \boldsymbol{\xi}, t) = Nm_0P(\mathbf{x}, \boldsymbol{\xi}, t),$$

kde N je počet částic v systému a m_0 hmotnost jedné částice. Uvažujeme, že všechny částice mají stejnou hmotnost. Rozdělovací funkce f má v případě trojrozměrného prostoru

¹Místo rychlosti $\boldsymbol{\xi}$ lze použít také hybnosti \mathbf{p} , pro odvození Boltzmannovy rovnice je výhodnější pracovat s rychlostmi.



Obr. 2.1: Částice v kvádru, černě vyznačené mají kladnou rychlost ξ_x (převzato z [8]).

následující rozměr:

$$[f] = \text{kg} \times \frac{1}{\text{m}^3} \times \frac{1}{(\text{m/s})^3}$$

a udává tedy střední celkovou hmotnost všech částic nacházejících se v bodě \mathbf{x} a pohybujících se rychlostí $\boldsymbol{\xi}$ v čase t . Umožňuje proto určit celkové hmotnosti částic s danými rychlostmi. Chceme-li například spočítat hmotnost částic v levé polovině kvádru o objemu $V = L_x \times L_y \times L_z$, viz obr. 2.1, které mají kladnou složku rychlosti ξ_x , integrujeme rozdělovací funkci f přes daný objem a kladnou poloosu ξ_x (integrály podle souřadnic y , z , ξ_y a ξ_z uvažujeme přes celé \mathbb{R}):

$$\int_{x=0}^{L_x/2} \int_{\xi_x > 0} f \, d\boldsymbol{\xi} \, d\mathbf{x}.$$

Výpočtem rychlostních momentů f lze určit také další veličiny jako hustotu ρ , hybnost \mathbf{p} , celkovou energii E a vnitřní energii částic e :

$$\rho(\mathbf{x}, t) = \int f(\mathbf{x}, \boldsymbol{\xi}, t) \, d\boldsymbol{\xi}, \quad (2.1)$$

$$\rho(\mathbf{x}, t) \mathbf{u}(\mathbf{x}, t) = \int \boldsymbol{\xi} f(\mathbf{x}, \boldsymbol{\xi}, t) \, d\boldsymbol{\xi}, \quad (2.2)$$

$$E(\mathbf{x}, t) = \frac{1}{2} \int \boldsymbol{\xi} \cdot \boldsymbol{\xi} f(\mathbf{x}, \boldsymbol{\xi}, t) \, d\boldsymbol{\xi}, \quad (2.3)$$

$$\rho e(\mathbf{x}, t) = \frac{1}{2} \int (\boldsymbol{\xi} - \mathbf{u}) \cdot (\boldsymbol{\xi} - \mathbf{u}) f(\mathbf{x}, \boldsymbol{\xi}, t) \, d\boldsymbol{\xi}, \quad (2.4)$$

kde $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ je makroskopická rychlost tekutiny.

Máme tedy soustavu částic a jejich pravděpodobnostní rozdělení v čase t a chceme určit, jak se bude vyvíjet v čase. Pokud nedochází ke kolizím, změní se rozdělení částic pro dané \mathbf{x} a $\boldsymbol{\xi}$ pouze konvekcí a působením vnějších sil \mathbf{F} :

$$f(\mathbf{x} + \boldsymbol{\xi} \, dt, \boldsymbol{\xi} + \frac{\mathbf{F}}{\rho} \, dt, t + dt) = f(\mathbf{x}, \boldsymbol{\xi}, t).$$

Ovšem uvažujeme-li srážky mezi částicemi, změní se uvedený vztah o aditivní člen $\Omega(f, f)$, nazývaný kolizní operátor, který tyto srážky zohledňuje:

$$f(\mathbf{x} + \boldsymbol{\xi} dt, \boldsymbol{\xi} + \frac{\mathbf{F}}{\rho} dt, t + dt) - f(\mathbf{x}, \boldsymbol{\xi}, t) = \frac{df}{dt} = \Omega(f, f). \quad (2.5)$$

V předchozí rovnici lze první člen nahradit Taylorovým rozvojem f podle času t v čase t :

$$f(\mathbf{x} + \boldsymbol{\xi} dt, \boldsymbol{\xi} + \frac{\mathbf{F}}{\rho} dt, t + dt) = f(\mathbf{x}, \boldsymbol{\xi}, t) + \frac{\partial f}{\partial t} dt + \frac{\partial f}{\partial x_i} \frac{\partial x_i}{\partial t} dt + \frac{\partial f}{\partial \xi_i} \frac{\partial \xi_i}{\partial t} dt + O(dt^2), \quad (2.6)$$

kde platí:

$$\begin{aligned} \frac{\partial x_i}{\partial t} &= \xi_i, \\ \frac{\partial \xi_i}{\partial t} &= \frac{F_i}{\rho}. \end{aligned}$$

Zanedbáme-li v (2.6) členy $O(dt^2)$ a dosadíme-li tento vztah do rovnice (2.5), dostaneme **Boltzmannovu rovnici**:

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x_i} \xi_i + \frac{\partial f}{\partial \xi_i} \frac{F_i}{\rho} = \Omega(f, f). \quad (2.7)$$

2.1 Kolizní operátor

Nyní se zaměříme na kolizní operátor, který udává dynamiku daného systému. Nejprve nastíníme odvození obecného kolizního operátoru (detailní odvození lze najít například v [12]). Jak se ukáže, tento kolizní operátor je pro numerickou implementaci značně složitý, proto bude později aproximován.

Jak již bylo zmíněno, uvažujeme elastické srážky částic, tedy takové, kde nedochází k přeměně kinetické energie na vnitřní tepelnou a zachovává se tedy celková energie i hybnost systému. Také se omezíme pouze na dvoučásticové srážky, protože vícečásticové srážky jsou v plynu málo pravděpodobné.

Uvažujme nyní srážku dvou částic o stejné hmotnosti m_0 s rychlostmi $\boldsymbol{\xi}, \boldsymbol{\xi}_*$ před srážkou a $\boldsymbol{\xi}', \boldsymbol{\xi}'_*$ po srážce (na částice nepůsobí žádné další síly, ke změně rychlostí tedy dochází jen během srážky). Z platnosti zákonů zachování hmotnosti, hybnosti a energie dostáváme:

$$m_0 + m_0 = m_0 + m_0, \quad (2.8)$$

$$m_0 \boldsymbol{\xi} + m_0 \boldsymbol{\xi}_* = m_0 \boldsymbol{\xi}' + m_0 \boldsymbol{\xi}'_*, \quad (2.9)$$

$$\frac{1}{2} m_0 \boldsymbol{\xi}^2 + \frac{1}{2} m_0 \boldsymbol{\xi}_*^2 = \frac{1}{2} m_0 \boldsymbol{\xi}'^2 + \frac{1}{2} m_0 \boldsymbol{\xi}'_*^2. \quad (2.10)$$

Z těchto vztahů lze vyjádřit rychlosti částic $\boldsymbol{\xi}'$, $\boldsymbol{\xi}'_*$ po srážce v závislosti na rychlostech $\boldsymbol{\xi}$, $\boldsymbol{\xi}_*$ před srážkou v této podobě:

$$\begin{aligned}\boldsymbol{\xi}' &= \boldsymbol{\xi} - \mathbf{k} \cdot (\boldsymbol{\xi} - \boldsymbol{\xi}_*) \mathbf{k}, \\ \boldsymbol{\xi}'_* &= \boldsymbol{\xi}_* + \mathbf{k} \cdot (\boldsymbol{\xi} - \boldsymbol{\xi}_*) \mathbf{k},\end{aligned}\tag{2.11}$$

kde \mathbf{k} je jednotkový vektor, který udává směr vychýlení částic při kolizi. Odečtením rovnic v (2.11) od sebe dostaneme:

$$\mathbf{g}' = \mathbf{g} - 2(\mathbf{k} \cdot \mathbf{g}) \mathbf{k},\tag{2.12}$$

kde $\mathbf{g} = \boldsymbol{\xi} - \boldsymbol{\xi}_*$ je relativní rychlost částic před srážkou a $\mathbf{g}' = \boldsymbol{\xi}' - \boldsymbol{\xi}'_*$ je relativní rychlost částic po srážce. Po vynásobení rovnosti (2.12) vektorem \mathbf{k} dostaneme:

$$\mathbf{k} \cdot \mathbf{g} = -\mathbf{k} \cdot \mathbf{g}'.\tag{2.13}$$

Rovnost velikostí vektorů relativních rychlostí

$$|\mathbf{g}| = |\mathbf{g}'|\tag{2.14}$$

odvodíme vynásobením rovnice (2.12) postupně \mathbf{g}' a \mathbf{g} a následným sečtením těchto rovnic. Dále zavedeme kolizní úhel Θ :

$$\mathbf{k} \cdot \mathbf{g} = g \cos \Theta,$$

kde $g = |\mathbf{g}|$. Z (2.12), (2.13) a (2.14) dále vyplývá vyjádření \mathbf{k} v závislosti na předkolizních a pokolizních relativních rychlostech:

$$\mathbf{k} = \frac{\mathbf{g} - \mathbf{g}'}{|\mathbf{g} - \mathbf{g}'|}.\tag{2.15}$$

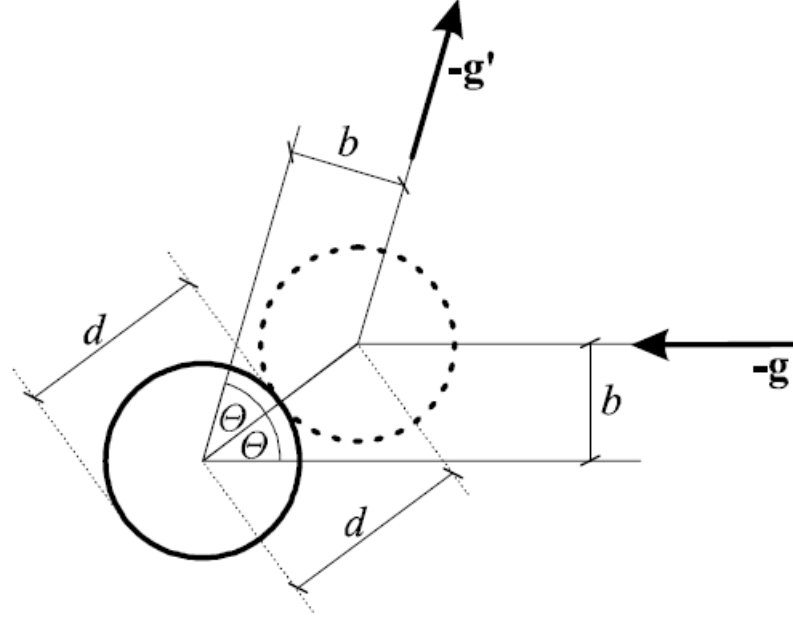
Pomocí výše uvedených vztahů (2.11) až (2.15) lze určit také rychlosti před srážkou v závislosti na rychlostech po srážce:

$$\begin{aligned}\boldsymbol{\xi} &= \boldsymbol{\xi}' - \mathbf{k} \cdot (\boldsymbol{\xi}' - \boldsymbol{\xi}'_*) \mathbf{k}, \\ \boldsymbol{\xi}_* &= \boldsymbol{\xi}'_* + \mathbf{k} \cdot (\boldsymbol{\xi}' - \boldsymbol{\xi}'_*) \mathbf{k}.\end{aligned}\tag{2.16}$$

Vztahy jsou stejné jako (2.11), což znamená, že je proces časově invariantní. Další vlastností, kterou využijeme, je jednotkový jakobián transformace $(\boldsymbol{\xi}, \boldsymbol{\xi}_*) \rightarrow (\boldsymbol{\xi}', \boldsymbol{\xi}'_*)$ předkolizních na pokolizní rychlosti, tedy:

$$\det \mathbf{J}_{(\boldsymbol{\xi}', \boldsymbol{\xi}'_*)}(\boldsymbol{\xi}, \boldsymbol{\xi}_*) = 1.\tag{2.17}$$

Dvoučásticová srážka je znázorněna na obr. 2.2, kde jsou částice znázorněny jako tuhé koule. Vidíme, že kolizní úhel Θ zůstává během kolize nezměněn. Stejně tak kolizní para-



Obr. 2.2: Grafické znázornění kolize dvou tuhých koulí o průměru d (převzato z [12]).

metr b udávající vzdálenost obou částic kolmou na relativní rychlosti. Vektor \mathbf{k} má směr stejný jako spojnice středů částic v okamžiku srážky.

Mezi parametry \mathbf{g} , \mathbf{k} a Θ existuje závislost, která je dána interakcí, která mezi částicemi při srážce působí. Pravděpodobnost, že částice bude rozptýlena pod úhlem z intervalu $(\Theta, \Theta + d\Theta)$ je dána veličinou zvanou diferenciální účinný průřez σ , která je určena vztahem:

$$\sigma \sin \Theta d\Theta = b db. \quad (2.18)$$

Nyní se vrátíme zpět k rozdělovací funkci $f(\mathbf{x}, \boldsymbol{\xi}, t)$ a budeme uvažovat její časovou změnu vyvolanou srážkami. Každá kolize $(\boldsymbol{\xi}, \boldsymbol{\xi}_*) \rightarrow (\boldsymbol{\xi}', \boldsymbol{\xi}'_*)$ vyvolá pokles rozdělovací funkce, který označíme jako S_- , kdežto inverzní kolize $(\boldsymbol{\xi}', \boldsymbol{\xi}'_*) \rightarrow (\boldsymbol{\xi}, \boldsymbol{\xi}_*)$ vyvolá nárůst rozdělovací funkce, který označíme jako S_+ . Celková změna f za jednotku času se rovná koliznímu operátoru a je tedy

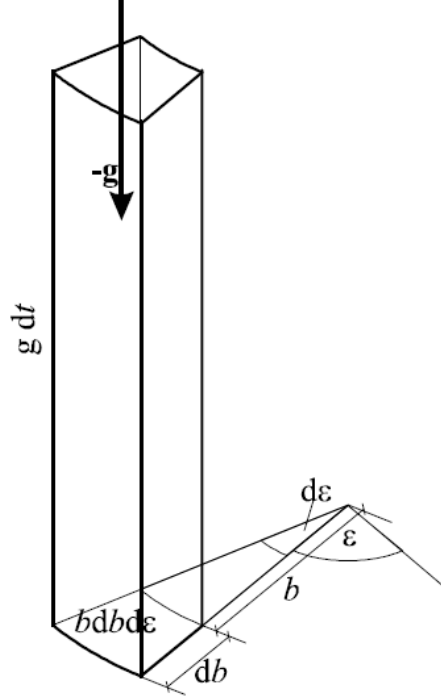
$$\Omega(f, f) = S_+ - S_-. \quad (2.19)$$

Počet částic s rychlostí $\boldsymbol{\xi}_*$, které by mohly kolidovat s částicemi s rychlostí $\boldsymbol{\xi}$, které se vyskytují v objemu $V = g dt db d\varepsilon$ (viz obr. 2.3) je

$$f(\mathbf{x}, \boldsymbol{\xi}_*, t) d\boldsymbol{\xi}_* g dt b d\varepsilon. \quad (2.20)$$

Pravděpodobnost výskytu částice s rychlostí $\boldsymbol{\xi}$ v bodě \mathbf{x} je

$$f(\mathbf{x}, \boldsymbol{\xi}, t) d\boldsymbol{\xi}. \quad (2.21)$$



Obr. 2.3: Grafické znázornění kolize částice s rychlostí $-\mathbf{g}$ v daném prostorovém segmentu kolidující s částicí ve středu výseče (převzato z [12]).

Pokles rozdělovací funkce za jednotkový čas u dané částice spočteme vynásobením obou pravděpodobností² (2.20) a (2.21). S využitím vztahu (2.18) a integrací přes všechny rychlosti $\boldsymbol{\xi}_*$, všechny kolizní úhly Θ v intervalu $(0, \pi/2)$ a všechny kolizní roviny ε v intervalu $(0, 2\pi)$ tak dostáváme:

$$S_- d\boldsymbol{\xi} = \iint f(\mathbf{x}, \boldsymbol{\xi}, t) f(\mathbf{x}, \boldsymbol{\xi}_*, t) g \sigma d\omega d\boldsymbol{\xi}_* d\boldsymbol{\xi}, \quad (2.22)$$

kde

$$d\omega = \sin \Theta d\Theta d\varepsilon.$$

Pro inverzní kolize dostaneme analogicky vyjádření S_+ ve tvaru:

$$S_+ d\boldsymbol{\xi}' = \iint f(\mathbf{x}, \boldsymbol{\xi}', t) f(\mathbf{x}, \boldsymbol{\xi}'_*, t) g' \sigma' d\omega d\boldsymbol{\xi}'_* d\boldsymbol{\xi}'. \quad (2.23)$$

Nyní využijeme identity (2.14). Dále ze zachování kolizního úhlu před i po srážce lze odvodit také $\sigma = \sigma'$ (viz [12]). Stejně tak lze ukázat, že platí rovnost $d\boldsymbol{\xi}'_* d\boldsymbol{\xi}' = d\boldsymbol{\xi}_* d\boldsymbol{\xi}$, vyplývající z transformace z pokolizních do předkolizních rychlostí za využití jednotkového

²Využíváme předpoklad molekulárního chaosu, tedy že jsou oba jevy navzájem nezávislé a příslušná f lze násobit.

jakobiánu (2.17). Dosazením výše uvedeného do (2.23) dostáváme:

$$S_+ d\boldsymbol{\xi} = \iint f(\boldsymbol{x}, \boldsymbol{\xi}', t) f(\boldsymbol{x}, \boldsymbol{\xi}_*, t) g\sigma d\omega d\boldsymbol{\xi}_* d\boldsymbol{\xi}. \quad (2.24)$$

Užitím vztahů (2.19), (2.22) a (2.24) dostaneme výsledný kolizní operátor:

$$\Omega(f, f) = \iint (f'f'_* - ff_*)g\sigma d\omega d\boldsymbol{\xi}_*, \quad (2.25)$$

kde

$$f = f(\boldsymbol{x}, \boldsymbol{\xi}, t), \quad f' = f(\boldsymbol{x}, \boldsymbol{\xi}', t), \quad f_* = f(\boldsymbol{x}, \boldsymbol{\xi}_*, t), \quad f'_* = f(\boldsymbol{x}, \boldsymbol{\xi}'_*, t).$$

Dosazením kolizního operátoru (2.25) do Boltzmannovy rovnice (2.7) dostaneme nelineární integro-diferenciální rovnici:

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x_i} \xi_i + \frac{\partial f}{\partial \xi_i} \frac{F_i}{\rho} = \iint (f'f'_* - ff_*)g\sigma d\omega d\boldsymbol{\xi}_*.$$

Pro jednodušší implementaci do simulací bude tento kolizní operátor (2.25) dále zjednodušen. Než tak učiníme, budeme se ještě zabývat kolizními invarianty.

2.1.1 Kolizní invarianty

Definice: Funkce $\phi(\boldsymbol{x}, \boldsymbol{\xi}, t) : \mathbb{R}^3 \times \mathbb{R}^3 \times (0, T) \rightarrow \mathbb{R}$ se nazývá kolizní invariant, když

$$\int \Omega(f, f)\phi d\boldsymbol{\xi} = 0, \quad (2.26)$$

pro skoro všechna $(\boldsymbol{x}, t) \in \mathbb{R}^3 \times (0, T)$ a všechny měřitelné funkce f .

Lemma: Následující identita

$$\begin{aligned} & \int \Omega(f, f)\phi d\boldsymbol{\xi} = \\ & = \frac{1}{4} \iiint (\phi(\boldsymbol{x}, \boldsymbol{\xi}, t) + \phi(\boldsymbol{x}, \boldsymbol{\xi}_*, t) - \phi(\boldsymbol{x}, \boldsymbol{\xi}', t) - \phi(\boldsymbol{x}, \boldsymbol{\xi}'_*, t))(f'f'_* - ff_*)g\sigma d\omega d\boldsymbol{\xi} d\boldsymbol{\xi}_*, \end{aligned} \quad (2.27)$$

platí pro všechna $(\boldsymbol{x}, t) \in \mathbb{R}^3 \times (0, T)$ a pro všechny funkce ϕ a f takové, že:

$$\iiint |(f'f'_* - ff_*)g\sigma\phi| d\omega d\boldsymbol{\xi}_* d\boldsymbol{\xi} < \infty \quad \forall (\boldsymbol{x}, t) \in \mathbb{R}^3 \times (0, T).$$

Důkaz: Viz [3].

Protože kolizní operátor je nenulový, tak z uvedeného lemmatu vyplývá, že musí být nulový výraz

$$\phi(\boldsymbol{x}, \boldsymbol{\xi}, t) + \phi(\boldsymbol{x}, \boldsymbol{\xi}_*, t) - \phi(\boldsymbol{x}, \boldsymbol{\xi}', t) - \phi(\boldsymbol{x}, \boldsymbol{\xi}'_*, t),$$

jak plyne z (2.26). Kolizní invariant proto musí splňovat následující rovnost:

$$\phi(\mathbf{x}, \boldsymbol{\xi}, t) + \phi(\mathbf{x}, \boldsymbol{\xi}_*, t) = \phi(\mathbf{x}, \boldsymbol{\xi}', t) + \phi(\mathbf{x}, \boldsymbol{\xi}'_*, t).$$

Porovnáním se zákony zachování (2.8), (2.9) a (2.10) dostaneme tzv. fundamentální kolizní invarianty:

$$\phi_0 = 1 \quad \phi_1(\boldsymbol{\xi}) = \boldsymbol{\xi}, \quad \phi_2(\boldsymbol{\xi}) = |\boldsymbol{\xi}|^2. \quad (2.28)$$

Lze dokázat, viz [4], že další kolizní invarianty pro odvozený kolizní operátor, které by nebyly lineární kombinací těchto tří fundamentálních kolizních invariantů, neexistují, jejich počet tedy odpovídá počtu zákonů zachování.

Protože integrál je lineární operátor, musí být také každá lineární kombinace kolizních invariantů kolizní invariant, tedy platí

$$\phi(\mathbf{x}, \boldsymbol{\xi}, t) = a + \mathbf{b} \cdot \boldsymbol{\xi} + c |\boldsymbol{\xi}|^2, \quad (2.29)$$

kde $a = a(\mathbf{x}, t) \in \mathbb{R}$, $\mathbf{b} = \mathbf{b}(\mathbf{x}, t) \in \mathbb{R}^3$ a $c = c(\mathbf{x}, t) \in \mathbb{R}$.

Dále se zaměříme na vývoj systému v čase, který je charakterizován veličinou zvanou entropie.

2.2 Entropie, H-teorém

Termodynamická veličina měrná entropie s udává míru neuspořádanosti dané soustavy. 2. termodynamický zákon říká, že entropie celého uzavřeného systému v závislosti na čase je neklesající funkce a je definována přes následující vztah:

$$ds = \frac{dq}{T},$$

kde q je měrné teplo a T termodynamická teplota. Pro uzavřený systém lze napsat 2. zákon termodynamiky ve tvaru

$$\frac{d}{dt} \int s d\mathbf{x} \geq 0.$$

V případě rovnosti se jedná o děje vratné, jinak o děje nevratné. Z nárůstu entropie poznáme vývoj systému v čase. Podobnou veličinu lze odvodit také ve fázovém prostoru pro rozdělovací funkci.

Dosadíme-li $\phi = \ln(f)$ do rovnice (2.27) a využijeme-li vlastností logaritmů, dostaneme:

$$\int \Omega(f, f) \phi d\boldsymbol{\xi} = \frac{1}{4} \iiint (f' f'_* - f f_*) \ln \left(\frac{f f_*}{f' f'_*} \right) g \sigma d\omega d\boldsymbol{\xi}_* d\boldsymbol{\xi}.$$

Protože platí

$$(X - Y) \ln \left(\frac{Y}{X} \right) \leq 0, \quad \text{pro } \forall X, Y > 0$$

a protože g , σ a rozdělovací funkce f v (2.25) jsou nezáporné, je integrand nekladný a integrál nekladného integrandu je též nekladný, proto platí následující nerovnost:

$$\int \ln(f) \Omega(f, f) d\xi \leq 0. \quad (2.30)$$

Nyní definujeme veličinu

$$H(t) = \int f \ln(f) d\xi, \quad (2.31)$$

pro kterou platí nerovnost zvaná **H-teorém**:

$$\frac{dH}{dt} \leq 0. \quad (2.32)$$

Vztah (2.32) lze odvodit zderivováním vztahu (2.31) podle t , užitím nerovnice (2.30), dosazením kolizního operátoru za $\frac{df}{dt}$ z Boltzmannovy rovnice a faktu, že skalár je kolizní invariant dle (2.28).

Veličina H vypovídá tedy o míře neuspořádanosti systému jako entropie. Pro ideální plyn existuje jednoduchý přepočít mezi veličinou H a entropií:

$$\rho s = -RH.$$

Znaménko mínus dokládá, že entropie v uzavřeném systému na rozdíl od veličiny H je neklesající funkce.

2.2.1 Rovnovážný stav

Nyní se zaměříme na rovnovážný stav, tedy stav maximální entropie a minimálního H . Rovnost v H -teorému nastane jen tehdy, pokud, v souladu s (2.26), je funkce $\ln(f)$ kolizní invariant. To bude, vyjádříme-li f pomocí (2.29) ve tvaru:

$$f_M(\mathbf{x}, \boldsymbol{\xi}, t) = \exp(a + \mathbf{b} \cdot \boldsymbol{\xi} + c |\boldsymbol{\xi}|^2) = A \exp(-\alpha |\boldsymbol{\xi} - \mathbf{u}|^2),$$

kde byla provedena substituce

$$a = -\alpha \mathbf{u}^2 + \ln A, \quad \mathbf{b} = 2\alpha \mathbf{u}, \quad c = -\alpha,$$

v níž A, α jsou konstanty a \mathbf{u} makroskopická rychlost plynu. Pokud je plyn v klidu, pak $\mathbf{u} = 0$. Funkci f_M nazveme Maxwellovo rozdělení. Je to rovnovážné rozdělení, tedy takové, že nedochází ke změnám rozložení rychlostí částic v prostoru a čase. Kolizní operátor je

tedy roven 0 a proto platí následující rovnice, jak se lze přesvědčit dosazením do Boltzmannovy rovnice:

$$\frac{\partial f_M}{\partial t} + \xi_i \cdot \frac{\partial f_M}{\partial x_i} = 0.$$

Neznámé konstanty A, α určíme z požadavku na korespondenci s makroskopickými veličinami a tak, využijeme-li vztahů (2.1) až (2.4) a stavových rovnic (1.12) a (1.13), dostaneme:

$$\begin{aligned} \alpha &= \frac{3}{4e} = \frac{1}{2RT}, \\ A &= \rho \left(\frac{4}{3}\pi e \right)^{-3/2} = \rho (2\pi RT)^{-3/2}. \end{aligned}$$

Výsledné Maxwell-Boltzmannovo rovnovážné rozdělení pak má následující podobu:

$$\begin{aligned} f^{eq} &= \rho \left(\frac{1}{2\pi RT} \right)^{3/2} \exp \left(-\frac{1}{2RT} (\boldsymbol{\xi} - \mathbf{u})^2 \right) = \\ &= \rho \left(\frac{3}{4\pi e} \right)^{3/2} \exp \left(\frac{-3(\boldsymbol{\xi} - \mathbf{u})^2}{4e} \right). \end{aligned} \quad (2.33)$$

Jak již bylo zmíněno, Boltzmannova rovnice (2.7) s odvozeným kolizním operátorem (2.25) je příliš složitá jak pro analytické, tak i numerické řešení. Byly proto odvozeny různé aproximace, které kolizní operátor zjednodušují. Jednou z aproximací je operátor BGK, v němž figuruje právě uvedené Maxwell-Boltzmannovo rovnovážné rozdělení.

2.3 Operátor BGK

Všechny zjednodušené kolizní operátory Ω_{ap} by měly splňovat některé důležité vlastnosti obecného kolizního operátoru:

1. Kolizní invarianty zůstanou nezměněné, což odpovídá zákonům zachování hmotnosti, hybnosti a energie:

$$\begin{aligned} \int \Omega_{ap} d\boldsymbol{\xi} &= 0, \\ \int \boldsymbol{\xi} \Omega_{ap} d\boldsymbol{\xi} &= \mathbf{0}, \\ \int \boldsymbol{\xi}^2 \Omega_{ap} d\boldsymbol{\xi} &= 0. \end{aligned} \quad (2.34)$$

2. Platí H-Teorém (2.32).
3. Rovnovážný stav odpovídá Maxwell-Boltzmannovu rozdělení (2.33).

V této práci bude používán nejznámější kolizní operátor BGK pojmenovaný podle jeho autorů - Bhatnagara, Grosse a Krooka:

$$\Omega_B(f, f) = -\frac{1}{\tau}(f - f^{eq}). \quad (2.35)$$

τ je relaxační konstanta, která udává tendenci přechodu systému do rovnovážného stavu f^{eq} . Operátor BGK výše uvedené podmínky splňuje, jak je ukázáno např. v [12].

2.4 Propojení s rovnicemi zachování makroskopických veličin

Nyní ukážeme, že z mezoskopického popisu tekutiny vyjádřeného Boltzmannovou rovnicí (2.7) lze odvodit zákony zachování makroskopických veličin, které byly zmíněny v úvodní kapitole (detailnější odvození lze nalézt např. v [4]). Boltzmannovu rovnici vynásobíme jednotlivými kolizními invarianty ϕ_j , $j = 0, 1, 2$, které jsou uvedeny v (2.28). Následně rovnici zintegrujeme přes rychlostní podprostor:

$$\frac{\partial}{\partial t} \int \phi_j f \, d\boldsymbol{\xi} + \frac{\partial}{\partial x_i} \int \xi_i \phi_j f \, d\boldsymbol{\xi} + \frac{F_i}{\rho} \int \phi_j \frac{\partial f}{\partial \xi_i} \, d\boldsymbol{\xi} = \int \phi_j \Omega(f, f) \, d\boldsymbol{\xi}. \quad (2.36)$$

Pravá strana rovnice je nulová, jak plyne z (2.26). Dosadíme-li pro $j = 0$, $\phi_0 = 1$ a uplatníme-li vztahy (2.1), (2.2) a

$$\int \frac{\partial f}{\partial \xi_i} \, d\boldsymbol{\xi} = 0,$$

dostaneme rovnici kontinuity (1.1), tedy

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho u_i) = 0.$$

Pokud dosadíme další kolizní invariant pro $j = 1$, $\phi_1 = \boldsymbol{\xi}$ do (2.36) a druhý člen levé strany rovnice přepíšeme za pomocí rozkladu mikroskopické rychlosti $\boldsymbol{\xi}$ na makroskopickou rychlost \boldsymbol{u} a relativní mikroskopickou rychlost \boldsymbol{v} , tedy

$$\boldsymbol{\xi} = \boldsymbol{u} + \boldsymbol{v}$$

a následně využijeme vztahů

$$\int v_i f \, d\boldsymbol{\xi} = 0$$

a pro silový člen

$$\int \xi_i \frac{\partial f}{\partial \xi_j} d\xi = - \int \frac{\partial \xi_i}{\partial \xi_j} f d\xi = \rho \delta_{ij},$$

kde bylo využito integrace per partes. Dostáváme Cauchyho rovnice dynamické rovnováhy (1.3)

$$\frac{\partial}{\partial t}(\rho u_j) + \frac{\partial}{\partial x_i}(\rho u_i u_j) = \frac{\partial \sigma_{ij}}{\partial x_i} + F_j, \quad (2.37)$$

kde

$$\sigma_{ij} = - \int v_i v_j f d\xi. \quad (2.38)$$

Konečně užitím posledního kolizního invariantu $\phi_2 = \xi^2$ dostaneme po složitějších výpočtech, viz [4], rovnici zachování celkové energie (1.11):

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x_i}(E u_i) = F_i u_i + \frac{\partial(\sigma_{ij} u_j)}{\partial x_j} - \frac{\partial q_i}{\partial x_i}, \quad (2.39)$$

kde

$$q_i = \frac{1}{2} \int v_i v_j v_j f d\xi. \quad (2.40)$$

Abychom dostali Navier-Stokesovy rovnice, musí platit pro σ_{ij} v (2.38) rovnost (1.4) a pro q_i v (2.40) Fourierův zákon (1.14). Pro obecnou rozdělovací funkci f však výrazy nelze určit. Avšak rozdělením f podle jednotlivých řádů perturbace lze integrály zjednodušit a vyjádřit, jak bude ukázáno v kapitole o Chapman Enskogově rozvoji.

2.5 Přejchod do bezrozměrné soustavy

V mechanice tekutin se využívá zákonů podobnosti. Ty říkají, že dané proudění nezávisí na konkrétních hodnotách příslušných veličin, ale na tzv. podobnostních číslech. Příkladem je Reynoldsovo číslo

$$Re = \frac{x_0 u_0}{\nu_0},$$

kde x_0 je charakteristická délka, u_0 charakteristická rychlost a ν_0 charakteristická kinematická viskozita. Reynoldsovo číslo dává do souvislosti setrvačné síly a viskozitu. Zákon podobnosti říká, že dva systémy jsou si dynamicky podobné, pokud mají shodnou geometrii a rovnají se příslušná Reynoldsova čísla. Tento poznatek umožňuje například provádět experimenty na zmenšených modelech, kdy se dosáhne stejných výsledků jen jsou výsledné veličiny zmenšeny v poměru, který je daný charakteristickými hodnotami veličin. Důvodem tohoto přístupu u LBM je zejména rychlostní omezení, které je třeba dodržet, aby

metoda konvergovala a byla stabilní. Převedením daného problému do konkrétní bezrozměrné soustavy se příslušné podmínky konvergence a stability totiž značně zjednoduší. Proto přistoupíme k přepočtu z rozměrové Boltzmannovy rovnice na bezrozměrnou. Rozměrové veličiny budeme značit se stříškou $\hat{}$, charakteristické rozměry, kterými budeme přeskálovávat, dolním indexem $_0$. Veličiny bez stříšky a dolního indexu $_0$ od teď uvažujeme jako bezrozměrné:

$$\begin{aligned} \hat{\rho} &= \rho_0 \rho, & \hat{\xi}_i &= u_0 \xi_i, & \hat{\mathbf{x}}_i &= x_0 \mathbf{x}_i, & \hat{t} &= \frac{x_0}{u_0} t, & \hat{\nu} &= \nu_0 \nu \\ \frac{\partial}{\partial \hat{\mathbf{x}}_i} &= \frac{1}{x_0} \frac{\partial}{\partial \mathbf{x}_i}, & \frac{\partial}{\partial \hat{t}} &= \frac{u_0}{x_0} \frac{\partial}{\partial t}. \end{aligned} \quad (2.41)$$

Parametr ρ_0 značí charakteristickou hustotu. Protože rozdělovací funkce \hat{f} udává hmotnost částic na jednotku objemu a rychlosti, je rozměr následující:

$$[\hat{f}] = \frac{\text{kg s}^3}{\text{m}^6},$$

tudíž volíme:

$$\hat{f} = \frac{\rho_0}{u_0^3} f. \quad (2.42)$$

Kolizní operátor $\hat{\Omega}(f, f)$ má stejný rozměr jako $\frac{df}{dt}$, tedy

$$[\hat{\Omega}(f, f)] = \frac{\text{kg s}^2}{\text{m}^6},$$

a proto volíme bezrozměrné $\Omega(f, f)$ následovně:

$$\hat{\Omega}(f, f) = \frac{\rho_0}{x_0 u_0^2} \Omega(f, f). \quad (2.43)$$

Nakonec ještě uvedeme vztah pro sílu:

$$\hat{\mathbf{F}} = \frac{\rho_0 u_0^2}{x_0} \mathbf{F}. \quad (2.44)$$

Dosazením vztahů (2.41) - (2.44) do rovnice (2.7) a podělením celé rovnice výrazem $\frac{\rho_0}{x_0 u_0^2}$ dostaneme formálně stejnou rovnici, jen nyní v bezrozměrných jednotkách:

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial \mathbf{x}_i} \xi_i + \frac{\partial f}{\partial \xi_i} \frac{\mathbf{F}_i}{\rho} = \Omega(f, f).$$

Pro dosažení operátoru BGK (2.35) převedeme rovnovážné rozdělení f^{eq} (2.33) také do bezrozměrných jednotek stejným způsobem, který jsme použili u f :

$$\hat{f}^{eq} = \frac{\rho_0}{u_0^3} f^{eq}.$$

Pak dostaneme, že

$$f^{eq} = \frac{\rho}{(2\pi\theta)^{3/2}} \exp\left(-\frac{1}{2\theta}(\boldsymbol{\xi} - \mathbf{u})^2\right), \quad (2.45)$$

kde θ je bezrozměrná teplota ve tvaru:

$$\theta = \frac{\hat{R}\hat{T}}{u_0^2}.$$

Relaxační parametr $\hat{\tau}$, který vystupuje v BGK operátoru, přeškálujeme následovně:

$$\hat{\tau} = \frac{x_{mfp}}{u_0} \tau,$$

kde x_{mfp} je střední volná dráha letu částice. Předpokládáme, že hodnota relaxačního parametru řádově odpovídá právě střední době mezi dvěma srážkami částice. Protože nyní Boltzmannovu rovnici nemůžeme pokrátit charakteristickým rozměrem x_0 , dostaneme Boltzmannovu rovnici v tomto tvaru:

$$\frac{x_{mfp}}{x_0} \left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial \mathbf{x}_i} \boldsymbol{\xi}_i + \frac{\partial f}{\partial \boldsymbol{\xi}_i} \frac{\mathbf{F}_i}{\rho} \right) = -\frac{1}{\tau} (f - f^{eq}). \quad (2.46)$$

Zavedeme Knudsonovo číslo

$$Kn = \frac{x_{mfp}}{x_0},$$

které sehraje klíčovou roli při Chapman Enskogově rozvoji v jedné z dalších podkapitol.

3 Hermitovy polynomy

Aby bylo možné Boltzmannovu rovnici vyřešit numericky, je třeba jednotlivé integrály, kterými jsou dány makroskopické momenty, diskretizovat. Ukazuje se, že vhodné k tomu jsou Hermitovy polynomy, které tvoří úplný ortogonální systém a je možné na ně jednotlivé členy rozdělovací funkce f promítnout. Uvažujeme prostor funkcí $L^2((a, b))$ a váhovou funkci $\omega(x) : (a, b) \rightarrow \mathbb{R}$, že

$$\int_a^b P(x)\omega(x) dx < \infty \quad (3.1)$$

pro libovolný polynom $P(x)$. Pak lze definovat skalární součin následovně:

$$\langle f, g \rangle := \int_a^b f(x)g(x)\omega(x) dx. \quad (3.2)$$

Hermitův polynom n -tého stupně je dán předpisem

$$H^{(n)}(x) = (-1)^n \frac{1}{\omega(x)} \frac{d^n}{dx^n} \omega(x), \quad (3.3)$$

s váhovou funkcí

$$\omega(x) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right). \quad (3.4)$$

Interval, na kterém jsou splněny podmínky (3.1) a (3.2), je celé \mathbb{R} . Hermitovy polynomy jsou navzájem ortogonální, tedy platí:

$$\langle H^{(n)}, H^{(m)} \rangle = n! \delta_{nm}.$$

Protože oblast řešení Boltzmannovy rovnice je vícerozměrná, uijeme Hermitových polynomů pro d dimenzí, které jsou definovány následovně:

$$\mathbf{H}^{(n)}(\mathbf{x}) = (-1)^n \frac{1}{\omega(\mathbf{x})} \nabla_{\alpha_1 \dots \alpha_n}^n \omega(\mathbf{x}) \quad (3.5)$$

s příslušnou váhovou funkcí

$$\omega(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}} \exp\left(\frac{-\mathbf{x}^2}{2}\right), \quad (3.6)$$

přičemž

$$\nabla_{\alpha_1 \dots \alpha_n}^n = \frac{\partial^n}{\partial \alpha_1 \dots \partial \alpha_n}.$$

Každé $\alpha_i, i = 1, 2, \dots, n$, značí některou souřadnici d -dimenzionálního prostoru, pro $d = 2$ typicky $\alpha_i \in \{x, y\} = \{x_1, x_2\}$, pro $d = 3$ pak $\alpha_i \in \{x, y, z\} = \{x_1, x_2, x_3\}$. Ale protože uvažujeme pouze spojité a hladké funkce, lze aplikovat Schwarzovu větu o smíšených derivacích, které pak nezávisí na pořadí, ale pouze na dané kombinaci α_1 až α_n . $\mathbf{H}^{(n)}(\mathbf{x})$ jsou tenzory n -tého řádu o d^n složkách. Pro $d = 2$ dosazením do vzorce (3.5) dostaneme pro $n = 0, 1, 2$:

$$\begin{aligned}\mathbf{H}^{(0)}(\mathbf{x}) &= 1, \\ \mathbf{H}^{(1)}(\mathbf{x}) &= (H_x^{(1)}, H_y^{(1)})^T = (x, y)^T, \\ \mathbf{H}^{(2)}(\mathbf{x}) &= \begin{pmatrix} H_{xx}^{(2)} & H_{xy}^{(2)} \\ H_{yx}^{(2)} & H_{yy}^{(2)} \end{pmatrix} = \begin{pmatrix} x^2 - 1 & xy \\ xy & y^2 - 1 \end{pmatrix}.\end{aligned}$$

Protože xy a yx jsou různé permutace téže kombinace, na základě Schwarzovy věty platí:

$$H_{xy}^{(2)} = H_{yx}^{(2)}.$$

V trojrozměrném prostoru lze Hermitovy polynomy spočítat obdobným způsobem, pro $n = 0, 1, 2$ dostáváme:

$$\begin{aligned}\mathbf{H}^{(0)}(\mathbf{x}) &= 1, \\ \mathbf{H}^{(1)}(\mathbf{x}) &= (H_x^{(1)}, H_y^{(1)}, H_z^{(1)})^T = (x, y, z)^T, \\ \mathbf{H}^{(2)}(\mathbf{x}) &= \begin{pmatrix} H_{xx}^{(2)} & H_{xy}^{(2)} & H_{xz}^{(2)} \\ H_{yx}^{(2)} & H_{yy}^{(2)} & H_{yz}^{(2)} \\ H_{zx}^{(2)} & H_{zy}^{(2)} & H_{zz}^{(2)} \end{pmatrix} = \begin{pmatrix} x^2 - 1 & xy & xz \\ yx & y^2 - 1 & yz \\ zx & zy & z^2 - 1 \end{pmatrix}.\end{aligned}$$

Ortogonalita platí také ve vícerozměrných prostorech, tedy:

$$\langle \mathbf{H}_\alpha^{(n)}, \mathbf{H}_\beta^{(m)} \rangle = \int \omega(\mathbf{x}) \mathbf{H}_\alpha^{(n)}(\mathbf{x}) \mathbf{H}_\beta^{(m)}(\mathbf{x}) d\mathbf{x} = \prod_{i=1}^d n_i! \delta_{nm}^{(2)} \delta_{\alpha\beta}^{(n+m)}.$$

kde $\delta_{\alpha\beta}^{(n+m)} = 1$ jen tehdy, když vektor $\alpha = (\alpha_1, \dots, \alpha_n)$ je permutací vektoru $\beta = (\beta_1, \dots, \beta_m)$, jinak je rovno nule. Čísla n_i udávají počet derivací podle jednotlivých rozměrů v α . Lze ukázat také úplnost těchto funkcí v prostoru $L_2(\mathbb{R}^n)$, tudíž Hermitovy polynomy tvoří ortogonální bázi Hilbertova prostoru.

Každou funkci $g(\mathbf{x}) \in L_2$ lze tedy napsat jako součet Hermitových polynomů vynásobených příslušnými koeficienty

$$g(\mathbf{x}) = \sum_{n=0}^{\infty} \mathbf{a}_\alpha^{(n)} \cdot \mathbf{H}_\alpha^{(n)}(\mathbf{x}), \quad (3.7)$$

kde každý koeficient $\mathbf{a}_\alpha^{(n)}$ je tensor n -tého řádu vypočtený následovně:

$$\mathbf{a}_\alpha^{(n)} = \frac{\langle g(\mathbf{x}), \mathbf{H}_\alpha^{(n)}(\mathbf{x}) \rangle}{\|\mathbf{H}_\alpha^{(n)}(\mathbf{x})\|^2} = \frac{1}{\prod_{i=1}^d n_i!} \int \omega(\mathbf{x}) g(\mathbf{x}) \mathbf{H}_\alpha^{(n)}(\mathbf{x}) d\mathbf{x},$$

kde

$$\|\mathbf{H}_\alpha^{(n)}(\mathbf{x})\|^2 = \langle \mathbf{H}_\alpha^{(n)}(\mathbf{x}), \mathbf{H}_\alpha^{(n)}(\mathbf{x}) \rangle.$$

Vzhledem k tomu, že hodnoty $\mathbf{a}_\alpha^{(n)}$ závisí pouze na kombinaci daného multiindexu α , mají všechny permutace příslušného koeficientu stejnou hodnotu. Lze dokázat (viz [9]), že tyto počty permutací s opakováním jsou takové, že se výsledné vztahy zjednoduší na

$$g(\mathbf{x}) = \omega(\mathbf{x}) \sum_{n=0}^{\infty} \frac{1}{n!} \mathbf{a}_\alpha^{(n)} \cdot \mathbf{H}_\alpha^{(n)}(\mathbf{x}), \quad (3.8)$$

kde

$$\mathbf{a}_\alpha^{(n)} = \int g(\mathbf{x}) \mathbf{H}_\alpha^{(n)} d\mathbf{x}.$$

U vztahů (3.8), resp. (3.7) ve skalárním součinu však sčítáme jen přes odlišné kombinace multiindexu α , což budeme uvažovat i nadále.

Obdobně můžeme zapsat rovnovážné rozdělení (2.45) pomocí Hermitových polynomů v závislosti na rychlostech $\boldsymbol{\xi}$, tedy diskretizovat rychlostní podprostor:

$$f^{eq}(\rho, \mathbf{u}, \theta, \boldsymbol{\xi}) = \omega(\boldsymbol{\xi}) \sum_{n=0}^{\infty} \frac{1}{n!} \mathbf{a}^{eq,(n)} \cdot \mathbf{H}^{(n)}(\boldsymbol{\xi}),$$

$$\mathbf{a}^{eq,(n)}(\rho, \mathbf{u}, \theta) = \int f^{eq}(\rho, \mathbf{u}, \theta, \boldsymbol{\xi}) \mathbf{H}^{(n)}(\boldsymbol{\xi}) d\boldsymbol{\xi}.$$

Porovnáním výrazů (3.4) a (2.45) si lze všimnout podobnosti obou výrazů a funkci f^{eq} lze napsat pomocí váhové funkce $\omega(\mathbf{x})$ (3.6) ve tvaru:

$$f^{eq} = \frac{\rho}{\theta^{d/2}} \omega\left(\frac{\boldsymbol{\zeta} - \mathbf{u}}{\sqrt{\theta}}\right).$$

Po provedení substituce $\boldsymbol{\eta} = (\boldsymbol{\zeta} - \mathbf{u})/\sqrt{\theta}$ dostaneme analyticky řešitelný integrál:

$$\mathbf{a}_\alpha^{eq,(n)} = \rho \int \omega(\boldsymbol{\eta}) \mathbf{H}_\alpha^{(n)}(\sqrt{\theta}\boldsymbol{\eta} + \mathbf{u}) d\boldsymbol{\eta} \quad (3.9)$$

a pro koeficienty $a_{\alpha}^{eq,(n)}$ vzhledem k Hermitově bázi dostaneme následující hodnoty:

$$\begin{aligned} a^{eq,(0)} &= \rho, \\ a_{\alpha}^{eq,(1)} &= \rho u_{\alpha}, \\ a_{\alpha\beta}^{eq,(2)} &= \rho(u_{\alpha}u_{\beta} + (\theta - 1)\delta_{\alpha\beta}). \end{aligned} \quad (3.10)$$

Vidíme, že tyto koeficienty Hermitovy báze jsou identické nebo podobné vztahům pro výpočet momentů rozdělovací funkce f a vyjadřují hmotnost, hybnost a energii:

$$\begin{aligned} a^{eq,(0)} &= \int f^{eq} d\boldsymbol{\xi} = \rho = \int f d\boldsymbol{\xi} = a^{(0)}, \\ a_{\alpha}^{eq,(1)} &= \int f^{eq} \xi_{\alpha} d\boldsymbol{\xi} = \rho u_{\alpha} = \int f \xi_{\alpha} d\boldsymbol{\xi} = a_{\alpha}^{(1)}, \\ \frac{a_{\alpha\alpha}^{eq,(2)} + 3\rho}{2} &= \int f^{eq} \boldsymbol{\xi}^2 d\boldsymbol{\xi} = E = \int f \boldsymbol{\xi}^2 d\boldsymbol{\xi} = \frac{a_{\alpha\alpha}^{(2)} + 3\rho}{2}. \end{aligned} \quad (3.11)$$

Proto stačí u Hermitova rozvoje brát v úvahu pouze Hermitovy polynomy do řádu dva, které nám garantují, že fyzika, tedy zákony zachování, budou dodrženy. Tím je opodstatněno využití právě Hermitových polynomů. Pokud polynomy vyšších řádů zanedbáme, dostáváme aproximaci

$$f^{eq}(\rho, \mathbf{u}, \theta, \boldsymbol{\xi}) \approx \omega(\boldsymbol{\xi}) \sum_{n=0}^N \frac{1}{n!} \mathbf{a}^{(n),eq} \cdot \mathbf{H}^{(n)}(\boldsymbol{\xi}), \quad (3.12)$$

$$f(\rho, \mathbf{u}, \theta, \boldsymbol{\xi}) \approx \omega(\boldsymbol{\xi}) \sum_{n=0}^N \frac{1}{n!} \mathbf{a}^{(n)} \cdot \mathbf{H}^{(n)}(\boldsymbol{\xi}), \quad (3.13)$$

kde volíme $N = 2$. Vyčíslíme-li jednotlivé polynomy a dosadíme rovnovážné rozdělení, lze f^{eq} vyjádřit v tomto tvaru:

$$f^{eq} \approx \omega(\boldsymbol{\xi}) \rho (1 + \xi_{\alpha} u_{\alpha} + (u_{\alpha} u_{\beta} + (\theta - 1) \delta_{\alpha\beta}) (\xi_{\alpha} \xi_{\beta} - \delta_{\alpha\beta})) =: \omega(\boldsymbol{\xi}) \mathbf{Q}(\boldsymbol{\xi}) \quad (3.14)$$

Využití Hermitovy báze a zanedbání bazových prvků vyšších řádů vede tedy ke zjednodušení rozdělovací funkce v rovnovážném rozdělení. Z důvodu diskretizace je třeba zjednodušit ale i nevlastní integrály pro rozdělovací funkci v (3.13) přes jednotlivé souřadnice rychlostního podprostoru. Na to budou použity Gauss-Hermitovy kvadraturní formule. Dříve však bude odvozena linearizovaná rovnice přes Chapman Enskogův rozvoj.

4 Chapman Enskogův rozvoj

Nyní abychom dostali úplné propojení mezi makroskopickými veličinami v Navier- Stokesových pohybových rovnicích a v rovnici energie, je třeba určit výrazy (2.38) a (2.40). To se provede pomocí poruchové teorie postupem odvozeným matematiky S. Chapmanem a D. Enskogem. Za tím účelem rozepíšeme rozdělovací funkci f do řádů perturbace podle Knudsonova čísla Kn . Celý postup provedeme pro rovnici s kolizním operátorem BGK, který bude při simulacích používán. Při použití jiných kolizních operátorů bude postup analogický.

Zavedeme zmenšující parametr poruchové teorie ε , který odpovídá Knudsonovu číslu

$$\varepsilon \rightarrow Kn,$$

podle kterého budeme jednotlivé proměnné rozepisovat do různých řádů. Je třeba podotknout, že parametr ε slouží jen jako čistě formální značka, díky které můžeme v dalším postupu odlišit členy různých řádů. Na konci úvah pak musíme formálně položit $\varepsilon = 1$. Rozepíšeme-li tedy f do řádů perturbace, dostaneme:

$$f = \sum_{i=0}^N f^{(i)} \varepsilon^i. \quad (4.1)$$

Protože se jedná o expanzi kolem rovnovážného stavu $f^{(eq)}$, odpovídá nulový řád perturbace právě Maxwell-Boltzmannovu rozdělení, tedy

$$f^{(0)} = f^{(eq)}.$$

Dosadíme-li (4.1) do (2.46) dostaneme:

$$\varepsilon \left(\frac{\partial}{\partial t} + \frac{\partial}{\partial x_i} \xi_i + \frac{\partial}{\partial \xi_i} \frac{F_i}{\rho} \right) (f^{(0)} + \varepsilon f^{(1)} + \varepsilon^2 f^{(2)} + \dots) = -\frac{1}{\tau} (\varepsilon f^{(1)} + \varepsilon^2 f^{(2)} + \dots). \quad (4.2)$$

V závislosti na požadované přesnosti rovnic proudění a konkrétním problému, který chceme řešit, volíme maximální stupeň poruchy. Pokud zvolíme jen rovnovážné rozdělení, tedy

$$f \approx f^{(0)} = f^{(eq)},$$

dostaneme právě Eulerovy pohybové rovnice. V našem případě budeme simulovat proudění, ve kterém budeme uvažovat i tření, proto bereme v úvahu i následující řád rozvoje

$$f \approx f^{(eq)} + \varepsilon f^{(1)},$$

pro který, jak později uvidíme, dostaneme Navier-Stokesovy pohybové rovnice. Stejným způsobem rozvineme také tenzor napětí $\boldsymbol{\sigma}$ a vektor přenosu tepla \mathbf{q} :

$$\begin{aligned}\boldsymbol{\sigma} &= \boldsymbol{\sigma}^{(0)} + \varepsilon \boldsymbol{\sigma}^{(1)}, \\ \mathbf{q} &= \mathbf{q}^{(0)} + \varepsilon \mathbf{q}^{(1)}.\end{aligned}$$

Jednotlivé složky rozvoje jsou definované stejně jako celkové veličiny v (2.38) a (2.40), (viz [13]):

$$\sigma_{ij}^{(n)} = - \int v_i v_j f^{(n)} d\boldsymbol{\xi}, \quad (4.3)$$

$$q_i^{(n)} = \frac{1}{2} \int v_i \mathbf{v}^2 f^{(n)} d\boldsymbol{\xi}. \quad (4.4)$$

Abychom určili $\sigma_{ij}^{(1)}$ a $q_i^{(1)}$, tak potřebujeme znát $f^{(1)}$, které dostaneme z (4.2) porovnáním členů u koeficientu ε :

$$f^{(1)} = -\tau \left(\frac{\partial}{\partial t} + \xi_i \frac{\partial}{\partial x_i} + \frac{F_i}{\rho} \frac{\partial}{\partial \xi_i} \right) f^{(0)}.$$

Nyní promítneme tento vztah do Hermitovy báze přenásobením Hermitovými polynomy a integrací přes rychlostní podprostor

$$\int f^{(1)} \mathbf{H}_\alpha^{(n)} d\boldsymbol{\xi} = \int \left(-\tau \left(\frac{\partial}{\partial t} + \xi_i \frac{\partial}{\partial x_i} + \frac{F_i}{\rho} \frac{\partial}{\partial \xi_i} \right) f^{(0)} \mathbf{H}_\alpha^{(n)} \right) d\boldsymbol{\xi}, \quad (4.5)$$

což lze přepsat na rovnici

$$a_\alpha^{(1)(n)} = -\tau \left(\frac{\partial}{\partial t} a_\alpha^{(0)(n)} + \frac{\partial}{\partial x_i} \left(a_{i\alpha}^{(0)(n+1)} + a_{\alpha,i}^{(0)(n-1)} \right) - \frac{\mathbf{F}_i}{\rho} a_{\alpha,i}^{(0)(n-1)} \right), \quad (4.6)$$

kde platí:

$$\begin{aligned}a_\alpha^{(1)(n)} &= \int f^{(1)} \mathbf{H}_\alpha^{(n)} d\boldsymbol{\xi}, \\ a_\alpha^{(0)(n)} &= \int f^{(eq)} \mathbf{H}_\alpha^{(n)} d\boldsymbol{\xi} = a_\alpha^{(eq)(n)}, \\ a_{i\alpha}^{(0)(n+1)} &= \int f^{(eq)} \mathbf{H}_{i\alpha}^{(n+1)} d\boldsymbol{\xi}, \\ a_{\alpha,i}^{(0)(n-1)} &= \int f^{(eq)} \frac{\partial \mathbf{H}_\alpha^{(n)}}{\partial \xi_i} d\boldsymbol{\xi}.\end{aligned}$$

Koeficienty $a_\alpha^{(0)(n)}$ je postačující vyjádřit do třetího řádu, přičemž pro $n = 0,1,2$, tyto koeficienty $a_\alpha^{(0)(n)}$ odpovídají koeficientům (3.10) a pro $n = 3$ lze tyto koeficienty nalézt v [9]. Výpočet $a_\alpha^{(1)(n)}$ v (4.6) je technicky poměrně náročný, přičemž vyžaduje vyřešit řadu integrálů a vypočítat koeficienty vůči Hermitově bázi, které jsou tenzory až 4. řádu. Celý postup lze nalézt např. v [9], v této práci se omezíme pouze na důležité body postupu.

Parciální derivace koeficientů v (4.6) podle času vypočteme užitím složených derivací podle zachovávajících veličin:

$$\frac{\partial a_{\alpha}^{(0)(n)}}{\partial t} = \frac{\partial a_{\alpha}^{(0)(n)}}{\partial \rho} \frac{\partial \rho}{\partial t} + \frac{\partial a_{\alpha}^{(0)(n)}}{\partial \mathbf{p}} \frac{\partial \mathbf{p}}{\partial t} + \frac{\partial a_{\alpha}^{(0)(n)}}{\partial \epsilon} \frac{\partial \epsilon}{\partial t}, \quad (4.7)$$

kde $\epsilon = \rho e$. Za parciální derivace hustoty, hybnosti a energie podle času můžeme do (4.7) dosadit z rovnice kontinuity, Eulerovy pohybové rovnice hybnosti a rovnice energie. Po výpočtu zbývajících derivací jednotlivých koeficientů v (4.7), které zde neuvádíme, ale které lze nalézt např. v [9] můžeme vyčíst koeficienty $a_{\alpha}^{(1)(n)}$ a dosadit je do vztahu (4.6), čímž dostaneme:

$$\begin{aligned} a^{(1)(0)} &= 0, \\ a_i^{(1)(1)} &= 0, \\ a_{ij}^{(1)(2)} &= -\tau \rho \theta \Lambda_{ij}, \\ a_{ijk}^{(1)(3)} &= -\tau \rho \theta (\Lambda_{ij} \xi_k + \Lambda_{ik} \xi_j + \Lambda_{jk} \xi_i) + \left(\delta_{ij} \frac{\partial \theta}{\partial x_k} + \delta_{ik} \frac{\partial \theta}{\partial x_j} + \delta_{jk} \frac{\partial \theta}{\partial x_i} \right), \end{aligned}$$

kde

$$\Lambda_{ij} = \frac{\partial \xi_i}{\partial x_j} + \frac{\partial \xi_j}{\partial x_i} - \frac{2}{3} \frac{\partial \xi_k}{\partial x_k} \delta_{ij}.$$

Nyní můžeme dopočítat také odchylky tensoru napětí, za využití Hermitových polynomů:

$$\begin{aligned} \sigma_{ij}^{(n)} &= - \int v_i v_j f^{(n)} d\boldsymbol{\xi} = - \int (\xi_i - u_i)(\xi_j - u_j) f^{(n)} d\boldsymbol{\xi} = \\ &= - (a_{ij}^{(n)(2)} + \delta_{ij} a^{(n)(0)} - u_j a_i^{(n)(1)} - u_i a^{(n)(1)} + u_i u_j a^{(n)(0)}). \end{aligned}$$

Pro $n = 0, 1$ dostaneme:

$$\begin{aligned} \sigma_{ij}^{(0)} &= -p \delta_{ij} = -\rho \theta \delta_{ij}, \\ \sigma_{ij}^{(1)} &= \frac{2}{3} \rho e \tau \Lambda_{ij} = \rho \theta \tau \Lambda_{ij}. \end{aligned} \quad (4.8)$$

Stejným způsobem určíme také vektor tepelného toku

$$\begin{aligned} q_i^{(n)} &= \frac{1}{2} \int (\xi_i - u_i)(\xi_j - u_j)^2 f^{(n)} d\boldsymbol{\xi} = \\ &= a_{ijj}^{(n)(3)} - 2u_j a_{ij}^{(n)(2)} - u_i a_{jj}^{(n)(2)} + u_j^2 a_i^{(n)(1)} + 2u_i u_j a_j^{(n)(1)} - (u_i u_j^2 + 5u_i) a^{(n)(0)}, \end{aligned}$$

kde pro $n = 0, 1$ dostaneme:

$$\begin{aligned} q_i^{(0)} &= 0, \\ q_i^{(1)} &= -\frac{5}{2} \rho \theta \tau \frac{\partial \theta}{\partial x_i} = -\frac{10}{9} \rho e \tau \frac{\partial e}{\partial x_i}. \end{aligned} \quad (4.9)$$

Pokud nyní dosadíme za $\sigma_{ij} = \sigma_{ij}^{(0)} + \sigma_{ij}^{(1)}$ dle (4.8) do rovnice (2.37) dostaneme Navier-Stokesovy pohybové rovnice. Podobně po dosazení $q_i = q_i^{(0)} + q_i^{(1)}$ do rovnice (2.39) nám dá rovnici zachování energie.

Porovnáním (4.8) s (1.7) dostaneme vztah mezi prvním koeficientem viskozity η a relaxačním parametrem τ

$$\eta = \frac{2}{3}\rho e\tau = p\tau = \tau\rho\theta \quad (4.10)$$

a pro objemovou viskozitu dostaneme

$$\eta_B = 0,$$

což odpovídá teorii ideálních jednoatomových plynů. Pokud dosadíme za vektor tepelného toku \mathbf{q} v energetické rovnici Fourierův zákon (1.14) a porovnáme s (4.9), dostaneme následující vztah mezi tepelnou vodivostí λ a relaxačním parametrem τ :

$$\lambda = \frac{5}{3}\rho e\tau = \frac{5}{2}\rho\tau\theta,$$

kde jednotlivé přepočty dostaneme využitím stavové a kalorimetrické rovnice.

Tím jsme ukázali, že rozvoj Boltzmannovy rovnice s BGK operátorem do prvního řádu perturbace vykazuje stejné fyzikální chování jako Navier-Stokesovy rovnice. Tím je ospravedlněno využití aproximace v podobě BGK operátoru. Oproti makroskopickému popisu tekutiny jsme zde určili hodnoty koeficientů viskozity a tepelné vodivosti právě pomocí relaxačního parametru τ , který v podstatě popisuje síly na mikroskopické úrovni, které mezi částicemi působí. V případě makroskopického popisu jsou hodnoty viskozity a tepelné vodivosti vztahy empirickými.

Pokud bychom uvažovali i další řád perturbace, tedy

$$f \approx f^{(0)} + f^{(1)} + f^{(2)},$$

dostali bychom Burnettův model, který je aplikovatelný, pokud se pohybujeme ve vyšších hodnotách Kn , což se vyskytuje například u šíření akustických vln o vysokých frekvencích. V postupu, který jsme provedli, jsme promítli rovnovážnou funkci dle (3.12) až do čtvrtého řádu rozvoje, tj. pro $N = 4$. Pokud uvedený řád snížíme, dostaneme zjednodušené modely, které lze aplikovat za speciálních podmínek. Pro $N = 3$ dostáváme model pro izotermické stlačitelné proudění, tedy takové, kde nedochází k přenosu tepla. Řád můžeme snížit ještě o jeden, pak dostáváme pro $N = 2$ model popisující izotermické slabě stlačitelné proudění, které lze použít zhruba do Machova čísla

$$Ma = \frac{u_0}{c_s} < 0,3,$$

kde c_s je rychlost zvuku.

Protože v aplikaci, která byla v rámci diplomové práce simulována, se jedná o proudění

bez přenosu tepla při nízkých rychlostech splňující podmínku Machova čísla, budeme využívat pouze rozvoj do $N = 2$ a od teď dál se právě na tento model omezíme. V další kapitole přistoupíme k diskretizaci rychlostního podprostoru, která proběhne pomocí Gauss-Hermitovy kvadraturní formule, čímž se integrály v (3.13) a (3.12) změní v konečné sumy.

5 Gauss-Hermitova kvadraturační formule

Gauss-Hermitova kvadraturační formule je speciálním případem Gaussovy kvadraturační formule, která umožňuje numerickou integraci:

$$\int_{-\infty}^{\infty} f(x)\omega(x)dx = \sum_{i=0}^n w_i f(x_i) + E(f) = \mathcal{Q}(f) + E(f),$$

kde $\omega(x)$ je váhová funkce, x_i jsou vhodně zvolené uzly na integračním intervalu, w_i jsou příslušné koeficienty, $\mathcal{Q}(f)$ je kvadraturační formule a $E(f)$ je chyba kvadraturační formule. Váhová funkce $\omega(x)$ je pro Gauss-Hermitovu kvadraturační formuli dána předpisem:¹

$$\omega(x) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right).$$

Algebraický řád neboli stupeň přesnosti kvadraturační formule lze vyjádřit celým číslem q , které udává nejvyšší stupeň polynomu, který kvadraturační formule integruje přesně. Platí tedy $E(x^j) = 0$ pro $j = 0, 1, \dots, q$ a $E(x^{q+1}) \neq 0$. Pokud kvadraturační formule má algebraický řád alespoň N a $\{P_n\}_{n=0,1,\dots}$ tvoří ortogonální systém polynomů stupně nejvýše N vzhledem k váhové funkci ω , tak tato kvadraturační formule má stupeň přesnosti $2N + 1$, právě tehdy, když její uzly jsou kořeny polynomu P_{N+1} . Uzly i koeficienty u Gauss-Hermitovy formule jsou voleny tak, aby bylo dosaženo stupně přesnosti $2N + 1$. Chceme-li tedy polynom $P_N(x)$ stupně N zintegrovat Gauss-Hermitovou kvadraturační formuli

$$\int_{-\infty}^{\infty} \omega(x)P_N(x) dx = \sum_{i=0}^q w_i P_N(x_i) \quad (5.1)$$

přesně, musí být dle výše uvedeného splněna podmínka $\frac{N-1}{2} \leq q$, přičemž uzly x_i , kde $i = 0, 1, \dots, q$ jsou kořeny Hermitova polynomu $H^{(q+1)}(x)$. Například pro polynom stupně $n = 3$, dostáváme $q = 1$ a uzly x_0 a x_1 , které jsou právě kořeny polynomu $H^{(2)}(x)$. Váhy jednotlivých uzlů pro Gauss-Hermitovu kvadraturační formuli jsou dány předpisem

$$w_i = \frac{(q+1)!}{((q+1)H^{(q)}(x_i))^2}. \quad (5.2)$$

Uvedený postup nyní rozšíříme na vícerozměrné funkce, abychom jej mohli aplikovat na rozdělovací funkci f . Mějme obecně polynom $P_N(\mathbf{x})$ stupně N v d dimenzích

$$P_N(\mathbf{x}) = \sum_{N_1+\dots+N_d \leq N} a_{N_1 \dots N_d} x_1^{N_1} \dots x_d^{N_d},$$

¹Je vícero možností, jak lze váhovou funkci pro Gauss-Hermitovu formuli zadefinovat. Jednotlivé varianty se liší o konstanty.

kde $N_i, i = 1, 2, \dots, d$, jsou přirozená čísla. Exponenciální funkce v $\omega(\mathbf{x})$ a také jednotlivé členy polynomů umožňují separaci členů polynomů na součin funkcí závislých na právě jedné proměnné. Výraz (5.1) ve více dimenzích lze rozepsat na součin integrálů po složkách ve tvaru:

$$\int \omega(\mathbf{x}) P_N(\mathbf{x}) d\mathbf{x} = \int \omega(\mathbf{x}) \sum a_{N_1 \dots N_d} x_1^{N_1} \dots x_d^{N_d} d\mathbf{x} = \sum a_{N_1 \dots N_d} \prod_{j=1}^d \int \omega(x_j) x_j^{N_j} dx_j. \quad (5.3)$$

Jednotlivé integrály v (5.3) vyřešíme dle (5.1) a dostaneme:

$$\sum a_{N_1 \dots N_d} \prod_{j=1}^d \int \omega(x_j) x_j^{N_j} dx_j = \sum a_{N_1 \dots N_d} \prod_{j=1}^d \sum_{i=0}^{q_j} w_{i,j} x_{i,j}^{N_j}, \quad (5.4)$$

kde $x_{i,j}$ je i -tý uzel dosažený do polynomu x_j a $w_{i,j}$ je jeho příslušná váha. Součin můžeme roznásobit na jednotlivé členy a dostaneme uzly v prostoru \mathbf{x}_i s příslušnými váhami w_i . Potřebujeme-li pro polynom jedné proměnné stupně $N = 2q+1$ právě $q+1$ uzlů pro exaktní integraci (kořeny polynomu $\mathbf{H}^{(q+1)}(x)$), odpovídá tomu $(q+1)^d$ bodů v d dimenzích. Výraz (5.4) tedy lze upravit na výsledný tvar:

$$\int \omega(\mathbf{x}) P_N(\mathbf{x}) d\mathbf{x} = \sum_{i=0}^{(q+1)^d - 1} w_i P_N(\mathbf{x}_i).$$

Uvedené poznatky nyní můžeme aplikovat na výpočet koeficientů rozdělovací funkce rovnovážného rozdělení (3.14), které spočítáme projekcí:

$$a^{eq,(n)} = \int f^{(eq)} \mathbf{H}^{(n)}(\boldsymbol{\xi}) d\boldsymbol{\xi} = \int \omega(\boldsymbol{\xi}) Q(\boldsymbol{\xi}) \mathbf{H}^{(n)}(\boldsymbol{\xi}) d\boldsymbol{\xi}.$$

Tento integrál nyní můžeme vypočítat pomocí Gauss-Hermitovy kvadrurní formule. Protože $Q(\boldsymbol{\xi})$ i $\mathbf{H}^{(n)}(\boldsymbol{\xi})$ uvažujeme jako polynomy nejvýše 2. stupně, je polynom vzniklý jejich součinem nejvýše stupně $N = 4$, stačí tak uvažovat $q \geq (N - 1)/2 = 1,5 \approx 2$, tedy kořeny ξ_i polynomů $\mathbf{H}^{(q+1)}(\boldsymbol{\xi})$ nejméně třetího řádu s příslušnými váhami w_i :

$$a^{eq,(n)} = \sum_{i=0}^{(q+1)^d - 1} w_i Q(\boldsymbol{\xi}_i) \mathbf{H}^{(n)}(\boldsymbol{\xi}_i).$$

Stejný postup můžeme provést také u rozdělovací funkce f , kterou nejprve zapíšeme pomocí Hermitových polynomů až do druhého řádu dle (3.13):

$$f(\mathbf{x}, \boldsymbol{\xi}, t) \approx \omega(\boldsymbol{\xi}) \sum_{n=0}^2 \frac{1}{n!} \mathbf{a}^{(n)}(\mathbf{x}, t) \cdot \mathbf{H}^{(n)}(\boldsymbol{\xi}).$$

Výpočet jednotlivých koeficientů probíhá analogicky, jen rozšíříme integrand tak, abychom mohli použít kvadraturní formuli (5.1):

$$\begin{aligned} \mathbf{a}^{(n)}(\mathbf{x}, t) &= \int f(\mathbf{x}, \boldsymbol{\xi}, t) \mathbf{H}^{(n)}(\boldsymbol{\xi}) d\boldsymbol{\xi} = \int f(\mathbf{x}, \boldsymbol{\xi}, t) \frac{\omega(\boldsymbol{\xi})}{\omega(\boldsymbol{\xi})} \mathbf{H}^{(n)}(\boldsymbol{\xi}) d\boldsymbol{\xi} \approx \\ &\approx \sum_{i=0}^{(q+1)^d-1} \frac{w_i}{\omega(\boldsymbol{\xi}_i)} f(\mathbf{x}, \boldsymbol{\xi}_i, t) \mathbf{H}^{(n)}(\boldsymbol{\xi}_i) \end{aligned}$$

Ke každé diskrétní rychlosti $\boldsymbol{\xi}_i$ dostaneme tedy příslušnou rozdělovací funkci diskrétní v rychlostním podprostoru

$$f_i(\mathbf{x}, t) = \frac{w_i}{\omega(\boldsymbol{\xi}_i)} f(\mathbf{x}, \boldsymbol{\xi}_i, t)$$

a taktéž u rovnovážného rozdělení

$$f_i^{eq} = w_i Q(\mathbf{u}, \theta, \boldsymbol{\xi}_i).$$

Jak bylo ukázáno Chapman-Enskogovým rozvojem, u izotermních slabě stlačitelných problémů nám stačí polynomy nejvýše druhého stupně. Místo spojité funkce (3.14) dostaneme konečný počet hodnot závisících na $\boldsymbol{\xi}_i$, pro $i = 0, 1, \dots, m$

$$f_i^{eq} = w_i \rho \left(1 + \xi_{i\alpha} u_\alpha + \frac{1}{2} (u_\alpha u_\beta + (\theta - 1) \delta_{\alpha\beta}) (\xi_{i\alpha} \xi_{i\beta} - \delta_{\alpha\beta}) \right), \quad (5.5)$$

kteří však ještě spojitě závisí na \mathbf{x} a t . Hodnota m závisí na volbě rychlostního setu, jak bude ukázáno v další kapitole. Bez újmy na obecnosti můžeme pro izotermní model zvolit $\theta = 1$, protože teplotu budeme uvažovat v průběhu simulace za konstantní. To vede na další zjednodušení výrazu v (5.5):

$$f_i^{eq} = w_i \rho \left(1 + \xi_{i\alpha} u_\alpha + \frac{1}{2} u_\alpha u_\beta (\xi_{i\alpha} \xi_{i\beta} - \delta_{\alpha\beta}) \right). \quad (5.6)$$

Stejně tak pokud dosadíme do Boltzmannovy rovnice (2.7) diskrétní rychlosti $\boldsymbol{\xi}_i$ a příslušné f_i a f_i^{eq} , dostaneme ke každému i danou rovnici a tedy soustavu $m+1$ diferenciálních rovnic

$$\left(\frac{\partial f_i}{\partial t} + \frac{\partial f_i}{\partial x_\alpha} \xi_{i\alpha} + \frac{\partial f_i}{\partial \xi_{i\alpha}} \frac{F_\alpha}{\rho} \right) = -\frac{1}{\tau} (f_i - f_i^{eq}). \quad (5.7)$$

Makroskopické momenty v (3.11) se změjí v sumy:

$$\begin{aligned}\rho &= \sum_{i=0}^m f_i = \sum_{i=0}^m f_i^{eq}, \\ \rho \boldsymbol{\xi}_i &= \sum_{i=0}^m f_i \boldsymbol{\xi}_i = \sum_{i=0}^m f_i^{eq} \boldsymbol{\xi}_i, \\ E &= \sum_{i=0}^m f_i \boldsymbol{\xi}_i^2 = \sum_{i=0}^m f_i^{eq} \boldsymbol{\xi}_i^2.\end{aligned}\tag{5.8}$$

Nyní přistoupíme k volbě rychlostního setu.

5.1 Diskrétní rychlostní sety

Podle toho, jaké fyzikální jevy u daného problému předpokládáme a jakou přesnost požadujeme, volíme stupeň polynomu u kvadraturní formule² a tak dostaneme různé diskretizace rychlostí a příslušné rychlostní sítě. Nejpoužívanější z nich budou uvedeny v této kapitole. V této práci bude uveden výpočet rychlostního setu pomocí Gauss-Hermitovy kvadraturní formule na příkladu sítě, která bude použita v numerické simulaci v dalších kapitolách.

Pro popis konkrétního rychlostního setu je běžné používat následující značení

$$DdQq,$$

kde d značí počet prostorových dimenzí a q počet rychlostních směrů. Pro náš izotermický slabě stlačitelný problém ve dvou dimenzích postačí rozvoj do Hermitových polynomů druhého řádu, což vede na rychlostní set D2Q9, který nyní bude odvozen:

Rychlostní set D2Q9 lze vytvořit rozšířením jednorozměrného rychlostního setu D1Q3 do další dimenze. Uvažujeme vztah (5.1). Protože však při výpočtu druhých rychlostních momentů integrujeme polynomy nejvýše čtvrtého řádu³, musíme tomu přizpůsobit i kvadraturní pravidlo. Budeme sčítat tedy přes uzly, které jsou kořeny Hermitova polynomu třetího stupně

$$H^{(3)}(\xi') = \xi'^3 - 3\xi',$$

které jsou

$$\begin{aligned}\xi'_0 &= 0, \\ \xi'_{1,2} &= \pm\sqrt{3}.\end{aligned}$$

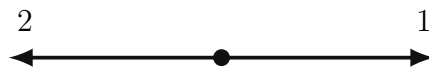
²Existují i další způsoby, jak vytvořit diskrétní rychlostní síť.

³Vektor přenosu tepla \mathbf{q} , který je třetím rychlostním momentem, je v izotermním případě nulový.

Příslušné váhy spočítáme dle vzorce (5.2) a dostaneme:

$$\begin{aligned}w'_0 &= 2/3, \\w'_{1,2} &= 1/6.\end{aligned}$$

Máme tedy jednorozměrný rychlostní set D1Q3 s jednou nulovou rychlostí a dvěma navzájem opačnými rychlostmi o velikosti $\sqrt{3}$, viz obr. 5.1.



Obr. 5.1: Grafické znázornění rychlostního setu D1Q3 (převzato z [8]).

Chceme-li přejít do dalšího rozměru, tak využijeme vztahu (5.4), tedy že lze kvadratickou formuli počítat po souřadnicích a výsledné hodnoty určit součinem. Máme-li tedy dva sety D1Q3 pro oba souřadnicové směry, dostaneme uvažované rychlosti jako všechny možné dvouprvkové kombinace daných rychlostí a příslušné váhy jako součin vah dílčích rychlostí. Nastanou tři početně odlišné případy:

Pokud volíme v obou směrech nulovou rychlost, dostaneme

$$\boldsymbol{\xi}_0 = (0,0)^T$$

a příslušnou váhu

$$w_0 = w_0'^2 = \frac{4}{9}.$$

Zkombinujeme-li nenulovou rychlost s nulovou rychlostí, dostaneme horizontální a vertikální směry

$$\begin{aligned}\boldsymbol{\xi}_{1,3} &= (0, \pm\sqrt{3})^T, \\ \boldsymbol{\xi}_{2,4} &= (\pm\sqrt{3}, 0)^T.\end{aligned}$$

Všechny mají stejné váhy o hodnotě

$$w_i = w_0'w_1' = \frac{1}{9}, \quad i \in \{1,2,3,4\}.$$

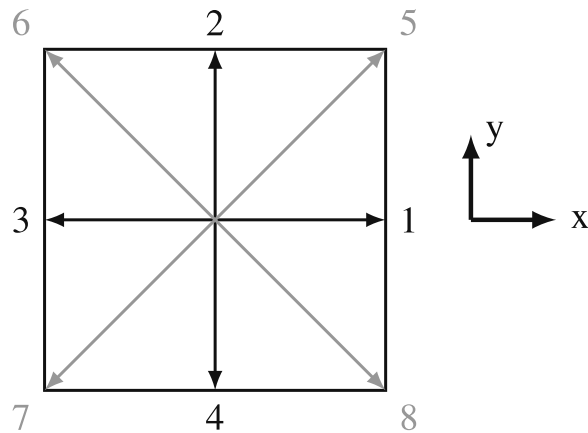
Konečně uhlopříčné směry dostaneme, volíme-li na obou osách nenulové rychlosti

$$\begin{aligned}\xi_{5,6} &= (\pm\sqrt{3}, \sqrt{3})^T, \\ \xi_{7,8} &= (\pm\sqrt{3}, -\sqrt{3})^T,\end{aligned}$$

s příslušnými váhami

$$w_i = w_2'^2 = \frac{1}{36} \quad i \in \{5,6,7,8\}.$$

Výsledný set D2Q9 tedy obsahuje nulovou rychlost, dále všechny pravoúhlé a úhlopříčné směry, což odpovídá devíti rychlostem. Grafické znázornění je na obr. 5.2.



Obr. 5.2: Grafické znázornění rychlostního setu D2Q9, (převzato z [8]).

Protože většina rychlostních setů obsahuje násobky $\sqrt{3}$, je výhodné zavést následující substituci

$$\mathbf{c}_i = \frac{\xi_i}{\sqrt{3}},$$

která bude používána i nadále. V dalších kapitolách se ukáže, že daná hodnota odpovídá rychlosti zvuku:

$$c_s = \frac{1}{\sqrt{3}}.$$

Provedením této substituce dostaneme u našeho rychlostního setu celočíselné rychlosti:

$$\begin{aligned} \mathbf{c}_0 &= (0,0)^T, \\ \mathbf{c}_{1,3} &= (\pm 1, 0)^T, \\ \mathbf{c}_{2,4} &= (0, \pm 1)^T, \\ \mathbf{c}_{5,6} &= (\pm 1, 1)^T, \\ \mathbf{c}_{7,8} &= (\pm 1, -1)^T. \end{aligned}$$

Stejným způsobem se upraví také diskrétní rovnovážné funkce (5.6) a dostaneme:

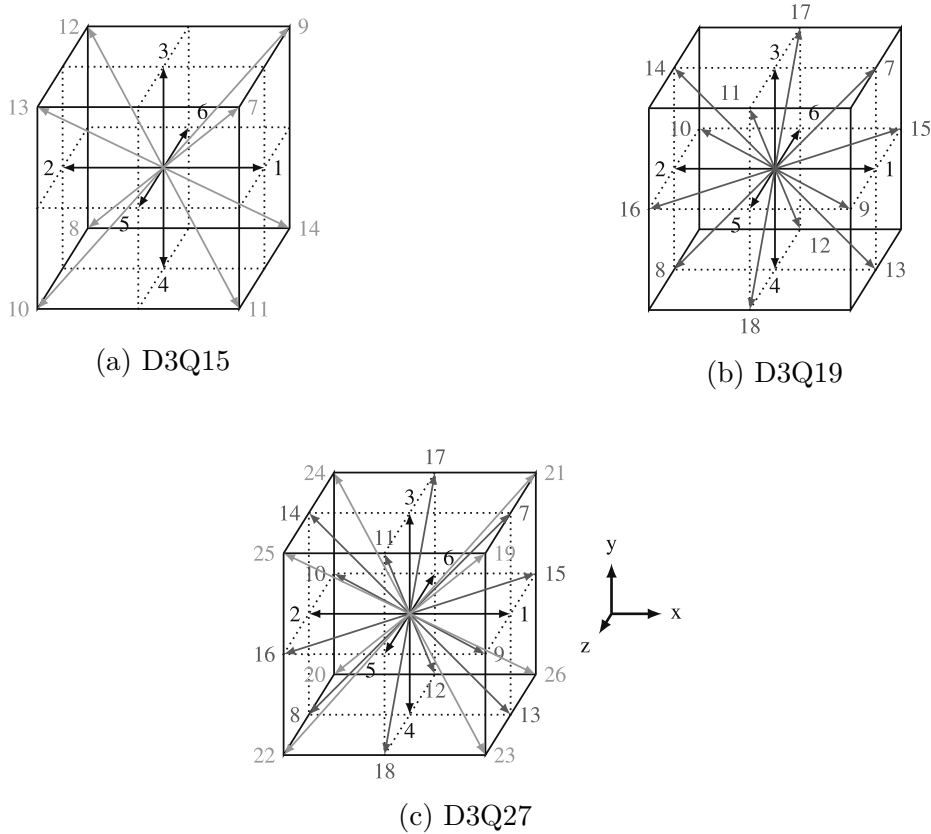
$$f_i^{eq} = w_i \rho \left(1 + \frac{c_{i\alpha} u_\alpha}{c_s^2} + \frac{1}{2c_s^4} u_\alpha u_\beta (c_{i\alpha} c_{i\beta} - \delta_{\alpha\beta}) \right), \quad i = 0, 1, \dots, 8. \quad (5.9)$$

Za rychlosti v (5.9) lze dosadit hodnoty vypočtené pro set D2Q9 a dostaneme ke každé diskrétní rychlosti c_i také diskrétní rovnovážnou funkci:

$$\begin{aligned} f_0^{eq} &= \frac{2\rho}{9} (2 - 3\mathbf{u}^2), \\ f_1^{eq} &= \frac{\rho}{18} (2 + 6u_x + 9u_x^2 - 3\mathbf{u}^2), & f_2^{eq} &= \frac{\rho}{18} (2 + 6u_y + 9u_y^2 - 3\mathbf{u}^2), \\ f_3^{eq} &= \frac{\rho}{18} (2 - 6u_x + 9u_x^2 - 3\mathbf{u}^2), & f_4^{eq} &= \frac{\rho}{18} (2 - 6u_y + 9u_y^2 - 3\mathbf{u}^2), \\ f_5^{eq} &= \frac{\rho}{36} (1 + 3u_x + 3u_y + 9u_x u_y + 3\mathbf{u}^2), & f_6^{eq} &= \frac{\rho}{36} (1 - 3u_x + 3u_y - 9u_x u_y + 3\mathbf{u}^2), \\ f_7^{eq} &= \frac{\rho}{36} (1 - 3u_x - 3u_y + 9u_x u_y + 3\mathbf{u}^2), & f_8^{eq} &= \frac{\rho}{36} (1 + 3u_x - 3u_y - 9u_x u_y + 3\mathbf{u}^2). \end{aligned}$$

Místo rovnovážné funkce spojité v rychlostním podprostoru máme nyní konečný počet diskrétních rychlostí c_i s příslušnými hodnotami rozdělovací funkce f_i a $f_i^{(eq)}$. Přestože rozdělovací funkce je pouze pravděpodobnostním vyjádřením pohybu částic, lze si nyní pohyb částic zjednodušeně představit jako skupiny částic o hmotnosti f_i , které se pohybují právě danou diskrétní rychlostí c_i . Tyto skupiny bývají v literatuře nazývány populace. Tento termín bude používán i dále v této práci.

Pro úplnost se zmíníme i o dalších běžně používaných rychlostních setech. Například rozšířením rychlostního setu D2Q9 o další dimenzi stejným způsobem, jak bylo uvedeno výše, dostaneme trojrozměrný set D3Q27, znázorněn na obr. 5.3c. Existuje i mnoho dalších rychlostních setů řešících složitější fyziku, které vykazují různé stupně přesnosti a stability. Zvolíme-li například Hermitovy polynomy vyšších řádů, dostaneme sety s větším počtem rychlostí. Dá se ukázat, pomocí teorie reprezentací konečných grup (viz [11]), že všechny sety musí splňovat podmínky invariance vůči rotační symetrii krychle (jedná se tedy o takové transformace, co zobrazí všech 8 vektorů $(\pm 1, \pm 1, \pm 1)^T$ identicky na sebe). Tomu odpovídají právě 4 různé báze: počátek, 12 vektorů spojující středy hran s počátkem, 8 vektorů spojující vrcholy s počátkem a 6 vektorů spojující středy stěn s počátkem. Báze samotné, případně jejich kombinace, pak tvoří různé rychlostní sety, jako např.: D3Q15 (viz obr. 5.3a) nebo D3Q19 (viz obr. 5.3b).



Obr. 5.3: Znázornění vybraných rychlostních trojrozměrných setů, (převzato z [8]).

Nejčastěji používané sety, které byly nyní zmíněny až do tří dimenzí s příslušnými váhami a rychlostmi jsou shrnuté v tabulce 5.1.

Z fyzikálních zákonů lze odvodit další obecné podmínky, které musí každý rychlostní set splňovat. Vycházejí z rotační isotropie rychlostní sítě, tedy nezávislosti fyzikálních vlastností na prostorovém směru. Podle zvoleného nejvyššího řádu Chapman Enskogova rozvoje dostáváme různý počet podmínek. V případě rozvoje až do prvního řádu, čemuž odpovídají Navier-Stokesovy rovnice, dostaneme následující podmínky, viz [13]:

$$\begin{aligned}
 \sum_i w_i &= 1, \\
 \sum_i w_i c_{i\alpha} &= 0, \\
 \sum_i w_i c_{i\alpha} c_{i\beta} &= c_s^2 \delta_{\alpha\beta}, \\
 \sum_i w_i c_{i\alpha} c_{i\beta} c_{i\gamma} &= 0, \\
 \sum_i w_i c_{i\alpha} c_{i\beta} c_{i\gamma} c_{i\mu} &= c_s^4 (\delta_{\alpha\beta} \delta_{\gamma\mu} + \delta_{\alpha\gamma} \delta_{\beta\mu} + \delta_{\alpha\mu} \delta_{\beta\gamma}), \\
 \sum_i w_i c_{i\alpha} c_{i\beta} c_{i\gamma} c_{i\mu} c_{i\varepsilon} &= 0.
 \end{aligned} \tag{5.10}$$

Označení	Rychlosti \mathbf{c}_i	Počet	$ \mathbf{c}_i $	w_i
D1Q3	(0)	1	0	2/3
	(± 1)	2	1	1/6
D2Q9	(0,0)	1	0	4/9
	($\pm 1, 0$), (0, ± 1)	4	1	1/9
	($\pm 1, \pm 1$)	4	$\sqrt{2}$	1/36
D3Q15	(0,0,0)	1	0	2/9
	($\pm 1, 0, 0$), (0, $\pm 1, 0$), (0,0, ± 1)	6	1	1/9
	($\pm 1, \pm 1, \pm 1$)	8	$\sqrt{3}$	1/72
D3Q19	(0,0,0)	1	0	1/3
	($\pm 1, 0, 0$), (0, $\pm 1, 0$), (0,0, ± 1)	6	1	1/18
	($\pm 1, \pm 1, 0$), ($\pm 1, 0, \pm 1$), (0, $\pm 1, \pm 1$)	12	$\sqrt{2}$	1/36
D3Q27	(0,0,0)	1	0	8/27
	($\pm 1, 0, 0$), (0, $\pm 1, 0$), (0,0, ± 1)	6	1	2/27
	($\pm 1, \pm 1, 0$), ($\pm 1, 0, \pm 1$), (0, $\pm 1, \pm 1$)	12	$\sqrt{2}$	1/54
	($\pm 1, \pm 1, \pm 1$)	8	$\sqrt{3}$	1/216

Tab. 5.1: Běžně používané rychlostní sety.

Mimo uvedený postup tvorby jednotlivých rychlostních setů pomocí Hermitových polynomů lze hledat takové rychlostní sety, které vyhovují uvedeným rovnicím (5.10).

V této práci, jak již bylo zmíněno, bude používán pouze rychlostní set D2Q9.

6 Časová a prostorová diskretizace

Lattice Boltzmannova rovnice (5.7) je diskretizována rychlostně, lze dosadit pouze q hodnot rychlostí v závislosti na zvoleném rychlostním setu. Rozdělovací funkce f však ještě je spojitá v prostoru a čase. Aby bylo možné rovnici numericky řešit, je třeba provést také prostorovou a časovou diskretizaci. Podstatným rozdílem oproti metodě konečných objemů, kde je standardně nestrukturovaná síť z různě velkých polygonů (trojúhelníků, čtyřúhelníků, jehlanů, apod.), je u LBM prostorová mříž uniformní, skládá se ze stejně velkých čtverečků nebo krychliček. Uniformní síť je žádoucí z toho důvodu, aby se při každém časovém kroku populace částic f_i nacházely právě na uzlech sítě. Následující postup je převzatý z [8]. Uvažujme rozměr prostorové buňky Δx a časový krok diskretizace Δt , platí následující vztah:

$$c_{i\alpha} = \frac{\Delta x}{\Delta t}.$$

Nyní přistoupíme k časové a prostorové diskretizaci LBM, přičemž se omezíme na případ bez působících sil, Lattice-Boltzmannova rovnice (5.7) se zredukuje na tvar:

$$\frac{\partial f_i}{\partial t} + c_{i\alpha} \frac{\partial f_i}{\partial x_\alpha} = -\frac{f_i - f_i^{eq}}{\tau}. \quad (6.1)$$

Tato rovnice patří mezi hyperbolické rovnice prvního řádu. Jednou z metod, jak rovnici vyřešit, je metoda charakteristik (neboli metoda trajektorií). Tato metoda spočívá v tom, že rovnici (6.1) lze parametrizovat na trajektorie podle parametru ζ , na kterém závisí všechny další nezávislé proměnné, tedy $f_i = f_i(\mathbf{x}_i(\zeta), t(\zeta))$, a rovnici můžeme napsat jako obyčejnou diferenciální rovnici ve tvaru

$$\frac{df_i}{d\zeta} = -\frac{f_i(\zeta) - f_i^{eq}(\zeta)}{\tau},$$

kde

$$\frac{dt}{d\zeta} = 1, \quad \frac{dx_\alpha}{d\zeta} = c_{i\alpha} \quad (6.2)$$

a

$$f_i^{eq}(\zeta) = f_i^{eq}(\rho(\zeta), \mathbf{u}(\zeta)) = f_i^{eq}(\rho(\mathbf{x}(\zeta), t(\zeta)), \mathbf{u}(\mathbf{x}(\zeta), t(\zeta))).$$

Počáteční podmínky dostaneme z podmínek (6.2) integrací:

$$\begin{aligned} t_0 &= t(\zeta = 0), \\ \mathbf{x}_0 &= \mathbf{x}(\zeta = 0). \end{aligned}$$

Homogenní řešení rovnice pak je

$$f_{i,hom} = C(\zeta) \exp\left(\int_{\zeta_0}^{\zeta} -\frac{1}{\tau} d\zeta'\right).$$

Z nehomogenní části rovnice spočítáme $C(\zeta)$ ve tvaru:

$$C(\zeta) = \int_{\zeta_0}^{\zeta} \frac{1}{\tau} \exp\left(\int_{\zeta_0}^{\zeta'} \frac{1}{\tau} d\zeta''\right) f_i^{eq}(\zeta') d\zeta' + C.$$

Po dosazení $C(\zeta)$ do $f_{i,hom}$ dostaneme obecné řešení:

$$f_i(\zeta) = e^{-(\zeta-\zeta_0)/\tau} \left(C + \frac{1}{\tau} \int_{\zeta_0}^{\zeta} e^{\zeta'/\tau} f_i^{eq}(\zeta') d\zeta' \right). \quad (6.3)$$

Proměnnou ζ napíšeme jako $\zeta = \zeta_0 + \Delta t$ a vrátíme se k závislostem na \mathbf{x} a t a za integrační konstantu v (6.3) dosadíme $C = f_i(\mathbf{x}, t)$. To vede na rovnici:

$$f_i(\mathbf{x} + c_i \Delta t, t + \Delta t) = e^{-\Delta t/\tau} \left(f_i(\mathbf{x}, t) + \frac{1}{\tau} \int_t^{t+\Delta t} e^{(t'-t)/\tau} f_i^{(eq)}(\mathbf{x} + c_i(t' - t), t') dt' \right). \quad (6.4)$$

Pro počítačovou implementaci je třeba integrál (6.4) vyřešit. Pokud použijeme obdélníkové pravidlo s levým krajním bodem

$$\int_t^{t+\Delta t} g(t') dt' \approx g(t) \Delta t$$

a pokud nahradíme exponenciální funkci v (6.4) Taylorovou řadou

$$e^{-\Delta t/\tau} = \sum_{i=0}^{\infty} \frac{1}{i!} \left(-\frac{\Delta t}{\tau} \right)^i = 1 - \frac{\Delta t}{\tau} + O(\Delta t^2) \approx 1 - \frac{\Delta t}{\tau},$$

tak dostaneme:

$$f_i(\mathbf{x} + \mathbf{x}_i \Delta t, t + \Delta t) = \left(1 - \frac{\Delta t}{\tau} \right) f_i(\mathbf{x}, t) + \frac{\Delta t}{\tau} f_i^{(eq)}(\mathbf{x}, t) + O(\Delta t^2).$$

To je standardní Lattice Boltzmannova rovnice s kolizním operátorem BGK, jedná se však o aproximaci prvního řádu. Ukážeme ale, že pro aproximaci druhého řádu dostaneme stejnou rovnici. K diskretizaci integrálu uijeme lichoběžníkového pravidla:

$$\int_t^{t+\Delta t} g(t') dt' \approx (g(t) + g(t + \Delta t)) \Delta t / 2.$$

Dosazením do (6.4) a dosazením Taylorova rozvoje za exponenciální funkci až do druhého řádu dostaneme následující vztah:

$$f_i(\mathbf{x} + \mathbf{x}_i \Delta t, t + \Delta t) = \left(1 - \frac{\Delta t}{\tau} + \frac{\Delta t^2}{2\tau^2}\right) f_i(\mathbf{x}, t) + \frac{\Delta t}{2\tau} \left(f_i^{(eq)}(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) + \left(1 - \frac{\Delta t}{\tau}\right) f_i^{(eq)}(\mathbf{x}, t) \right) + O(\Delta t^3). \quad (6.5)$$

Zavedeme-li substituci

$$\bar{f}_i = f_i - \frac{(f_i - f_i^{(eq)})\Delta t}{2\tau},$$

tak přepočtením zjistíme, že nové populace \bar{f}_i mají stejné hustoty i hybnosti, tedy že:

$$\begin{aligned} \rho &= \sum_i \bar{f}_i = \sum_i f_i, \\ \rho \mathbf{u} &= \sum_i \bar{f}_i \mathbf{c}_i = \sum_i f_i \mathbf{c}_i \end{aligned}$$

a dosazením do rovnice (6.5) dostaneme rovnici

$$\bar{f}_i(\mathbf{x} + \mathbf{x}_i \Delta t, t + \Delta t) = \bar{f}_i(\mathbf{x}, t) - \frac{\bar{f}_i(\mathbf{x}, t) - f_i^{(eq)}(\mathbf{x}, t)}{\bar{\tau}} + O(\Delta t^3).$$

Zde jsme již dosadili standardně v implementacích používaný časový krok $\Delta t = 1$ a použili jsme upravený koeficient relaxace:

$$\bar{\tau} = \tau + \frac{\Delta t}{2}. \quad (6.6)$$

Vidíme tedy, že Lattice Boltzmannova rovnice v novém tvaru je také aproximací druhého řádu, kde jsme také ukázali, že nultý a první rychlostní moment jsou stejné jako u standardního f_i , což nás opravňuje \bar{f}_i používat. To uděláme i nadále, jen \bar{f}_i budeme dále psát bez nadtržítka.

Pro účely algoritmizace je zvykem rozdělit proces zachycený v rovnici do více kroků - na kolizi a proudění. Toto implementaci zjednoduší a zároveň je touto úpravou velmi názorně vidět, co se v průběhu jednoho kroku Δt s populacemi částic děje. Jednotlivé kroky si nyní více rozepíšeme:

1.) Kolize

Zavedeme postkolizní stav populací, který budeme značit s hvězdičkou $f_i^*(\mathbf{x}, t)$. Kolizní krok je pak dán předpisem:

$$f_i^*(\mathbf{x}, t) = \left(1 - \frac{\Delta t}{\tau}\right) f_i(\mathbf{x}, t) + \frac{\Delta t}{\tau} f_i^{(eq)}(\mathbf{x}, t).$$

2.) Proudění

Proudění na rozdíl od kolize již není lokální, tudíž se přenáší na okolní uzly. Populace částic f_i^* , jejichž hmotnosti se při kolizi změnila, se pohybují rychlostí \mathbf{c}_i , při dané prostorové diskretizaci, a za čas Δt změni pozici o $\Delta t \mathbf{c}_i$. Dostaneme následující rovnici:

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i^*(\mathbf{x}, t).$$

Nyní přejdeme na počáteční a okrajové podmínky Lattice Boltzmannovy metody.

7 Počáteční a okrajové podmínky

Abychom dostali konkrétní řešení soustavy parciálních diferenciálních rovnic, je třeba ještě specifikovat počáteční a okrajové podmínky, díky kterým lze vyčíslit integrační konstanty a dojít tak ke konkrétnímu řešení. Následující postup je převzatý z [8].

Počáteční podmínky nám udávají stav tekutiny na začátku simulace v čase t_0 ve všech místech tekutiny \mathbf{x} dané domény. Uvažujeme-li izotermní proudění, postačuje zadat rychlost a tlak:

$$\begin{aligned}\mathbf{u}(\mathbf{x}, t_0) &= \mathbf{u}_0(\mathbf{x}), \\ p(\mathbf{x}, t_0) &= p_0(\mathbf{x}).\end{aligned}$$

Okrajové podmínky popisují chování tekutiny na okrajích simulované oblasti, musí být zadány v každém čase simulace, aby bylo možné vypočítat další časový krok. Body, kde je zadána okrajová podmínka, budeme značit \mathbf{x}_B , \mathbf{n} pak bude značit příslušný normálový vektor. Nechť ϕ je nějaká veličina, pro kterou okrajové podmínky zadáváme. Obecně můžeme okrajovou podmínku napsat ve tvaru

$$b_1 \frac{\partial \phi}{\partial n} |_{\mathbf{x}_B, t} + b_2 \phi(\mathbf{x}_B, t) = b_3,$$

kde podle volby konstant b_1 , b_2 a b_3 rozlišujeme tři druhy okrajových podmínek:

1. Dirichletovy okrajové podmínky: $b_1 = 0, b_2 \neq 0$,
2. Neumannovy okrajové podmínky: $b_1 \neq 0, b_2 = 0$,
3. Robinovy okrajové podmínky: $b_1 \neq 0, b_2 \neq 0$.

V hydrodynamice se nejčastěji používají první dvě z uvedených okrajových podmínek. Dirichletova podmínka udává rychlost tekutiny na okraji a můžeme ji zapsat ve tvaru

$$\mathbf{u}(\mathbf{x}_B, t) = \mathbf{U}_B(\mathbf{x}_B, t).$$

Pokud se na okraji oblasti nachází stěna, která nepropouští žádnou tekutinu dovnitř ani ven, tak platí podmínka:

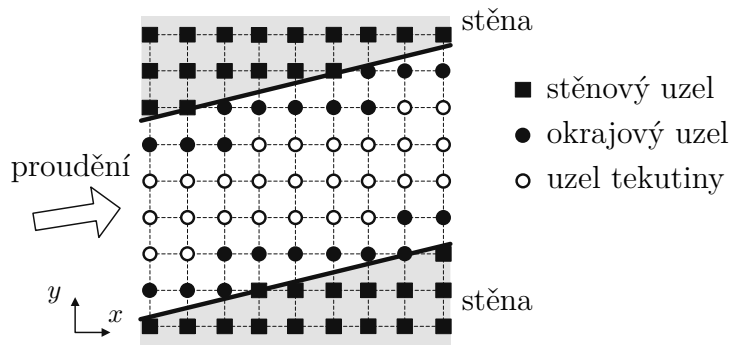
$$(\mathbf{u} - \mathbf{U}_B) \cdot \mathbf{n} = 0.$$

Pokud navíc uvažujeme drsné stěny, které zaručují, že relativní rychlost tekutiny vůči stěně je nulová, dostaneme tzv. no-slip podmínku:

$$(\mathbf{u} - \mathbf{U}_B) \cdot \mathbf{t} = 0,$$

kde \mathbf{t} je vektor tečny k hranici domény.

Neumannovu podmínku použijeme, chceme-li definovat třecí síly na hranici, a to v podobě



Obr. 7.1: Druhy uzlů v LBM na příkladu proudění v trubce, (převzato z [8], upraveno).

tensoru napětí:

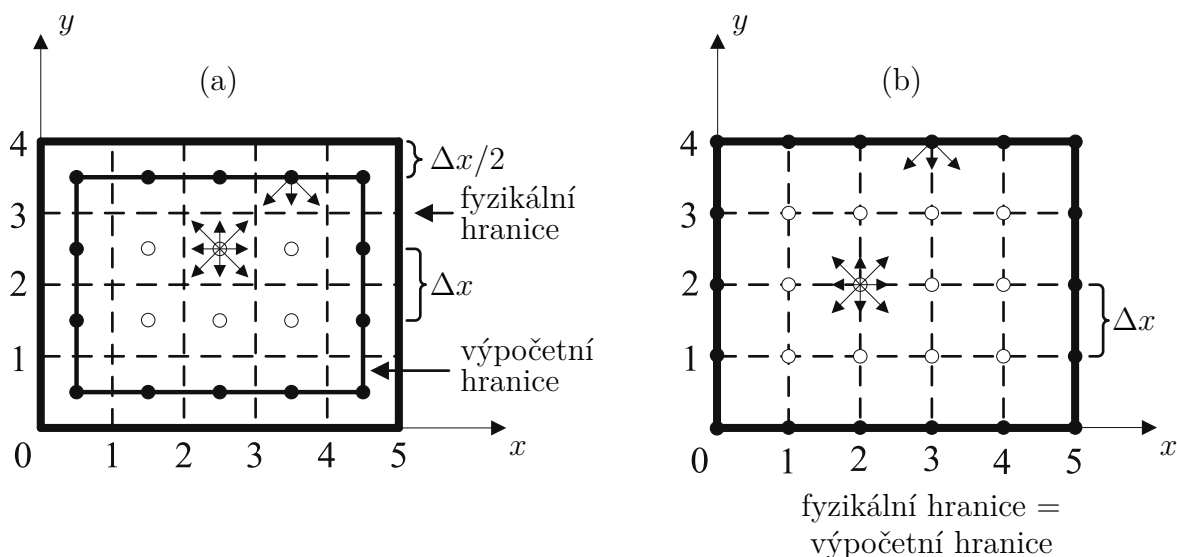
$$\mathbf{n} \cdot \boldsymbol{\sigma}(\mathbf{x}_B, t) = \mathbf{T}_B(\mathbf{x}_B, t).$$

Všechny podmínky můžeme libovolně kombinovat a rozdělit tak okraj oblasti na části s různými okrajovými podmínkami. Zároveň je třeba dodat, že pokud zkoumáme přesnost a řád aproximace numerické metody, je třeba uvažovat i řád aproximace okrajových podmínek. Pokud jsou nižšího řádu, tak se tyto nepřesnosti přenesou na celou simulovanou oblast, jak se dá ukázat na lineárním jednoduchém příkladu Couettova proudění či na kvadratickém Poiseuillově proudění (viz [8]).

7.1 Okrajové podmínky v LBM

Protože v LBM se nepočítá s makroskopickými veličinami, je formulace okrajových podmínek odlišná. Uvažujeme-li uzly v doméně, kterou chceme simulovat, můžeme je rozdělit na uzly v tekutině a uzly mimo oblast proudění. Okrajové uzly jsou ty, kde bude hrát roli okrajová podmínka. Jednotlivé typy uzlů jsou znázorněny na příkladu dvoudimenzionálního proudění v trubce na obr. 7.1. Okrajové podmínky spočívají v určení populací f_i sousedící s okrajovými uzly. Je třeba určit populace f_i okrajových uzlů s takovými směry, které by byly v bodu algoritmu proudění dány sousedními uzly tekutiny. Tyto sousední uzly však jsou již mimo tekutinu a nelze je tedy určit předpisem daným v kroku proudění. Stanovení těchto hodnot je právě úkolem okrajových podmínek. Ty nesmí změnit fyzikální vlastnosti proudění, tedy musí zůstat v platnosti vztahy (5.8). Základní rozdíl je nyní v tom, že díky mnoha populacím f_i je více stupňů volnosti a tedy i voleb, jak se daných momentů dopočítat a dostáváme celou řadu metod, jak okrajové podmínky stanovit. Omezíme se na ty nejjednodušší, které budou poté v námi uvedené aplikaci použité.

Další rozdělení okrajových podmínek vyplývá z toho, jestli se výpočetní uzly nacházejí přímo na fyzikálních hranicích, tedy přímo na stěně či na vtoku nebo výtoku. Pokud splý-



Obr. 7.2: Rozdíl mezi typy okrajových uzlů link-wise (a) a wet-node (b) (převzato z [8]).

vají přímo s hranicemi, tak jsou pojmenovány wet-node¹,. Pokud jsou uzly umístěny na středech spojnic uzlů z případu wet-node, nazývají se link-wise², rozdíl obou metod je znázorněn na obr. 7.2. Také se v tomto textu omezíme pouze na takové okrajové podmínky, které jsou složeny z úseček rovnoběžných se souřadnicovými osami. Křivočaré hranice by totiž vyžadovaly použití složitějších metod okrajových podmínek.

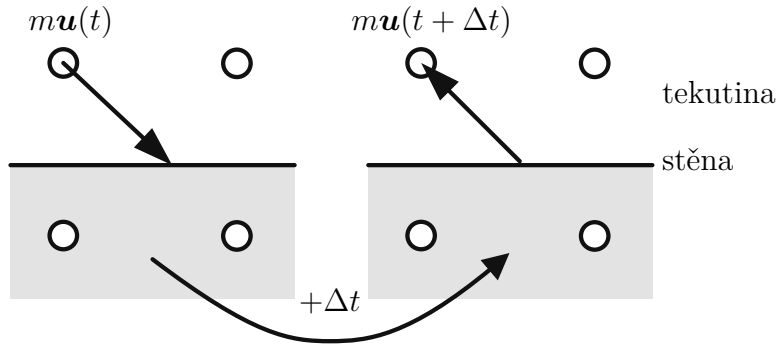
Okrajové podmínky můžeme dále rozdělit na otevřené a stěnové. Otevřené podmínky se týkají vtoků a výtoků do domény, stěnové podmínky uzlů nacházejících se na stěně. Lze je formulovat periodicky, kdy očekáváme, že na výtoku jsou stejné rychlosti jak na vtoku do oblasti. V aplikacích v rámci této práce však nebudou použity, proto nebudou detailněji popisovány a přesuneme se k neznámějším okrajovým podmínkám v Lattice Boltzmannově metodě.

Stěnová podmínka Bounce-Back

Patří mezi nejstarší a nejpoužívanější okrajové podmínky, vzhledem k její jednoduché implementaci. Lze ji použít pro stacionární i pohybující se stěny. Odpovídá totiž okrajové podmínce no-slip, která je nejčastěji používána na hranicích tekutiny a stěny. Princip lze ukázat, když si představíme jednotlivé populace jako částice, které naráží do zdi a odráží se zpět (obr. 7.3). V dalším časovém kroku tedy mají opačnou rychlost. Tím dostaneme nulovou relativní tečnou složku rychlosti při stěně, čímž je podmínka no-slip charakterizována. Samotnou ekvivalenci lze dokázat Chapman Enskogovým rozvojem těchto okrajových podmínek, viz [8].

¹Český ekvivalent pro dané výrazy neexistuje, proto budeme používat anglické názvy.

²U základních modelů okrajových podmínek přesně uprostřed mezi uzly.



Obr. 7.3: Zjednodušené znázornění metody bounce-back, (převzato z [8], upraveno). Na prvním obrázku je částice pohybující se směrem ke stěně v čase t . Na druhém obrázku v dalším časovém kroku $t + \Delta t$ je částice na téže pozici, ale má opačnou rychlost.

Metoda bounce-back patří do třídy link-wise. Hranice se tedy umísťují mezi výpočetní uzly, pokud možno přesně doprostřed, z důvodu vyšší přesnosti. To budeme uvažovat i my. Z hlediska délky výpočtu se rozlišují dva základní způsoby aplikace této techniky. Tzv. fullway bounce-back, kde uvažujeme že částice urazí celou cestu od hraničního uzlu po stěnový uzol a zase zpět. Uvažujeme-li tedy, že částice urazí vzdálenost mezi dvěma uzly Δx za čas Δt , trvá tento proces čas $2\Delta t$. Naproti tomu při halfway bounce-back částice letí jen po fyzikální hranici, kde se odrazí s opačnou rychlostí zpět. Zde urazí jen poloviční dráhu, což zabere polovinu času, a to Δt .

U časově stacionárních problémů lze zvolit oba dva modely, jednodušší počítačovou implementaci umožňuje metoda fullway. Pokud ovšem uvažujeme závislost na čase, dochází u fullway bounce-back varianty ke zpoždění na hranicích, jak lze vidět na obr. 7.4, které se přenáší samozřejmě do celé simulované oblasti a způsobuje větší časovou chybu metody. Protože nás časová závislost zajímá, budeme implementovat metodu halfway bounce-back. V případě stacionárních stěn lze napsat metodu následovně. Uvažujeme částici v hraničním bodě \mathbf{x}_b v čase t s rychlostí \mathbf{c}_i , která má kladnou složku ve směru kolmém k hranici. Po kolizi v čase $t + \Delta t$ bude mít tedy částice rychlost $\mathbf{c}_{\bar{i}} = -\mathbf{c}_i$. Pro populace f_i tedy platí:

$$f_{\bar{i}}(\mathbf{x}_b, t) = f_i^*(\mathbf{x}_b, t). \quad (7.1)$$

Vztahy mezi \bar{i} a i závisí na konkrétním rychlostním setu, který je pro daný problém používán. Pro náš set D2Q9 dostáváme:

i	1	2	3	4	5	6	7	8
\bar{i}	3	4	1	2	7	8	5	7

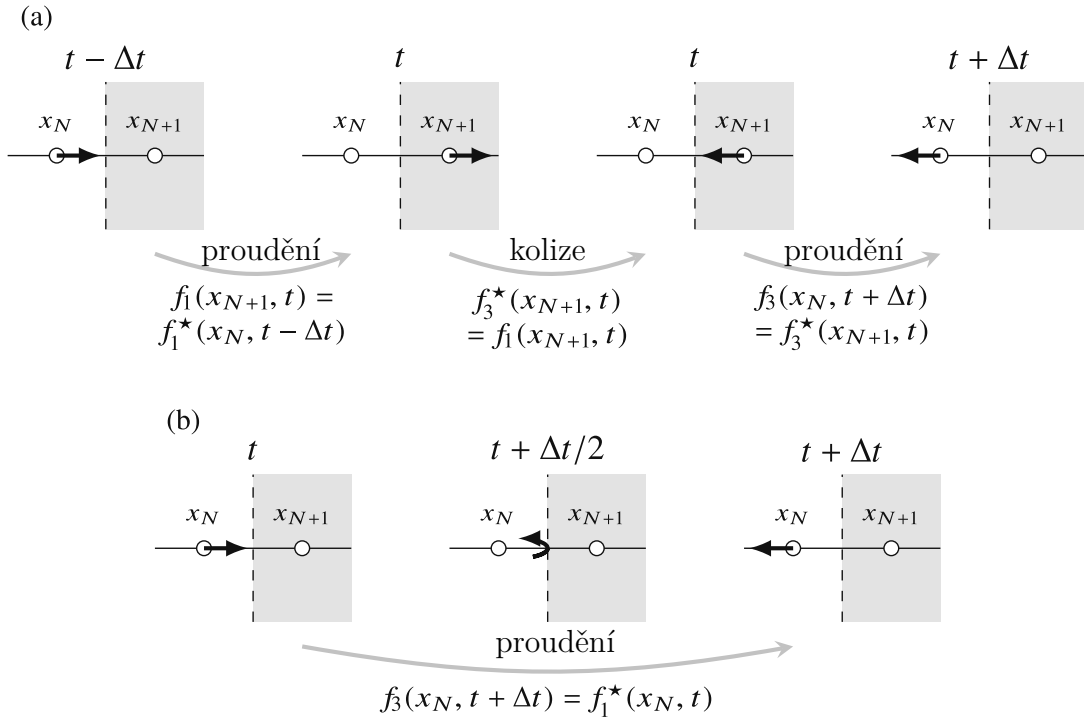
Tab. 7.1: Opačné rychlosti pro diskretní set D2Q9

Na každé stěně tedy máme tři rovnice předepisující změnu f_i . V rozích domény se zkombinují pravidla z obou dvou stěn, dostaneme tak 5 rovnic.

Podmínka pro pohybující se stěny vypadá podobně, jen je třeba zohlednit relativní rychlosti tak, aby byl zachován Galileův princip invariance (při transformaci do inerciálního systému, ve kterém je stěna v klidu, dostaneme klasické bounce-back pravidlo):

$$f_i(\mathbf{x}_b, t + \Delta t) = f_i^*(\mathbf{x}_b, t) - 2w_i\rho_w \frac{\mathbf{c}_i \cdot \mathbf{u}_w}{c_s^2}, \quad (7.2)$$

kde dolní index w značí vlastnosti na stěně v bodě $\mathbf{x}_w = \mathbf{x}_b + \frac{1}{2}\mathbf{c}_i\Delta t$ a w_i jsou příslušné váhy. U standardního nestlačitelného LBM lze za hustotu ρ_w dosadit jedničku, u stlačitelných modelů je třeba hustotu určit, například nahrazením průměrnou hustotou celého systému. Dosadíme-li $\mathbf{u}_w = 0$ do (7.2), dostaneme standardní model bounce-back.



Obr. 7.4: Porovnání techniky fullway (a) a halfway (b), (převzato z [8], upraveno).

Výhodou metody bounce-back je stabilita při nízkých časech relaxačního parametru τ , dokonce i pro $\tau \rightarrow \Delta t/2$ a také platnost zákona zachování hmotnosti na stacionárních stěnách, což všechny techniky okrajových podmínek nespĺňují. Nevýhodou je, že ji lze použít jen u schodovitých okrajových podmínek, které jsou rovnoběžné s výpočetní sítí a také že dosahuje aproximace druhého řádu jen tehdy, pokud se nachází hranice přesně uprostřed mezi uzly. S BGK kolizním operátorem je ekvivalence s podmínkou no-slip závislá na viskozitě, což lze odstranit použitím jiných kolizních operátorů (např. TRT, MRT).

Otevřené okrajové podmínky

Otevřené okrajové podmínky slouží k ošetření vstupů a výstupů domény. I zde existuje mnoho technik, jak k těmto okrajovým podmínkám přistoupit, protože může docházet k odrazu tlakových vln zpět do domény. My se zde omezíme pouze na uvedení rozšíření metody bounce-back.

Na vtocích a výtocích jsou okrajové podmínky dány buďto rychlostním, nebo tlakovým profilem. V případě rychlostního profilu, který se používá zpravidla na vtoku do domény, kde je rychlost známa, využijeme stejné podmínky jako u pohybujících se stěn (7.2), jen nyní rychlost \mathbf{u}_w bude mít nejenom tečnou, ale také normálovou složku.

Podmínka tlaku se využívá zejména na výstupu z domény, kde vlivem změny proudění opoti vstupu dojde k změnám rychlostního profilu, který není apriori znám. Zde se využije technika anti-bounce-back, kdy dojde k otočení znaménka oproti (7.1) a přičtení dalšího členu:

$$f_i^*(\mathbf{x}_b, t + \Delta t) = -f_i^*(\mathbf{x}_b, t) + 2w_i\rho_w \left(1 + \frac{(\mathbf{c}_i \cdot \mathbf{u}_w)^2}{2c_s^4} - \frac{\mathbf{u}_w^2}{2c_s^2} \right). \quad (7.3)$$

Rychlost \mathbf{u}_w na výtokové hranici není pochopitelně známá, ale dá se určit extrapolací:

$$\mathbf{u}_w = \mathbf{u}(\mathbf{x}_b) + 1/2(\mathbf{u}(\mathbf{x}_b) - \mathbf{u}(\mathbf{x}_{b+1})),$$

kde b a $b + 1$ jsou první a druhý nejbližší uzel v tekutině ve směru kolmém na hranici.

7.2 Počáteční podmínky v LBM

U Lattice Boltzmannovy metody se ukazuje, že počáteční podmínky mají na průběh simulace nemalý vliv. Počáteční podmínky spočívají ve znalosti rychlosti $\mathbf{u}(\mathbf{x}, t = 0) = \mathbf{u}_0(\mathbf{x})$ a tlaku $p(\mathbf{x}, t = 0) = p_0(\mathbf{x})$, případně hustoty $\rho(\mathbf{x}, t = 0) = \rho_0(\mathbf{x})$, v čase $t = 0$. Pokud neznáme hustotu, ale známe tlakové pole $p_0(\mathbf{x})$, můžeme dopočítat hustotu následujícím způsobem:

$$\rho_0(\mathbf{x}) = \bar{\rho} + \frac{p_0(\mathbf{x}) - \bar{p}}{c_s^2},$$

kde $\bar{\rho}$ je průměrná hustota a \bar{p} průměrný tlak.

Ze znalosti hustoty lze dopočítat rovnovážné stavy $f_i^{eq}(\rho_0, \mathbf{u}_0)$. Ukazuje se však, že pokud při spuštění simulace použijeme pouze rovnovážné stavy, bude aproximace pouze 1. řádu, tedy lineární. Je třeba přidat také nerovnovážné stavy, které lze určit v závislosti na tensoru napětí z Chapman Enskogova rozvoje, kde dostaneme závislost na tensoru napětí³

$$\sigma'_{ij} = \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k}.$$

³Zpravidla se volí podmínky, kde je divergence nulová a poslední člen v tensoru σ'_{ij} zmizí.

V případě použití BGK operátoru, lze použít aproximaci

$$f_i^{neq} \simeq -w_i \frac{\tau \rho}{c_s^2} Q_{i\alpha\beta} \frac{\partial u_\beta}{\partial x_\alpha},$$

kde $Q_{i\alpha\beta} = c_{i\alpha} c_{i\beta} - c_s^2 \delta_{\alpha\beta}$.

Počáteční podmínky rozdělovací funkce pak dostaneme ve tvaru

$$f_i = f_i^{eq} + f_i^{neq}.$$

Můžeme si všimnout, že v případě jednoduchých situací jako např. kapalina v klidu či pohybující se konstantní rychlostí, nerovnovážná část počáteční podmínky vypadne. Závěrem lze říct, že LBM se hodí zejména pro nestacionární proudění i po delších časech, kde se ukazuje, že závislost na počátečních podmínkách je zanedbatelná. Naopak u stacionárních proudění je závislost podstatná a v některých případech může trvat dlouhou dobu než simulace ke stacionárnímu stavu dokonverguje.

Část II

Aplikace

V této části bude Lattice Boltzmannova metoda, která byla teoreticky odvozena v první části této práce, aplikována na konkrétních problémech. Nejprve však algoritmus výpočtu popíšeme, přičemž v rámci tohoto popisu budou uvedeny i simulační zásady volby parametrů a také omezení, které jsou důsledkem aproximací provedených při odvozování a také zaručují konvergenci simulace. Celý postup pak bude aplikován na příkladu dvourozměrné kavity, pro který byl za účelem názornějšího pochopení metody vytvořen kód v jazyce C++ a verifikován na dostupných experimentálních datech. Hlavní aplikací pak je simulace obtékání překážek v dvourozměrné trubici provedené pomocí volně dostupné C++ knihovny openLB, která umožňuje paralelizaci výpočtů na vícero jádrech a dosažení reálných časů simulace. Výsledná data pak budou porovnána s řešičem konečných objemů implementovaným komerčním softwarem starCCM+.

8 O simulaci obecně

8.1 Převod do bezrozměrných jednotek a volba parametrů

Dříve než můžeme spustit simulaci, je třeba zvolit hodnoty všech parametrů k simulaci potřebných. Je třeba se rozhodnout v jakém diskretním rychlostním setu bude výpočet probíhat, jaký druh počátečních a okrajových podmínek bude zvolen, s ohledem na typ úlohy.

Jak bylo uvedeno výše, výpočet probíhá v bezrozměrných jednotkách, do kterých je třeba daný problém převést. Problém popisují charakteristické rozměry a hodnoty veličin, ze kterých lze vypočítat bezrozměrná podobnostní čísla. Rozměrové veličiny budeme v dalším textu značit se stříškou. Jsou to především charakteristická délka \hat{l} , charakteristická rychlost \hat{u}^1 , kinematická viskozita $\hat{\nu}$, hustota $\hat{\rho}$. Přepočet probíhá dle konverzních faktorů, které budeme označovat $\hat{C}_?$, kde otazník značí příslušnou veličinu (např. pro délku \hat{C}_l). Jak bylo ukázáno v kapitole 2.5, mezi konverzními faktory existují závislosti, počet nezávislých konverzních faktorů odpovídá počtu používaných základních jednotek (v tomto případě třem - m, s a kg). V obou soustavách dle zákona podobnosti musí podobnostní čísla nabývat stejných hodnot, v našem případě udává charakter proudění především Reynoldsovo číslo

$$Re = \frac{\hat{l}\hat{u}}{\hat{\nu}} = \frac{lu}{\nu}. \quad (8.1)$$

Rovnost obou Re platí, pokud

$$\frac{\hat{C}_l\hat{C}_u}{\hat{C}_\nu} = 1. \quad (8.2)$$

U simulací je také rozhodující rozměr prostorové výpočetní mřížky Δx a velikost časového kroku Δt . Zpravidla se v bezrozměrném systému volí

$$\Delta x = \Delta t = 1. \quad (8.3)$$

Střední hustotu volíme $\rho_0 = 1$. V závislosti na dané geometrii problému volíme rozměr buňky výpočetní mřížky $\Delta\hat{x}$, čímž je dán konverzní faktor:

$$\hat{C}_l = \frac{\Delta\hat{x}}{\Delta x}. \quad (8.4)$$

V modelu D2Q9 je rychlost zvuku $c_s = \sqrt{1/3} \approx 0,577$. Z Chapman Enskogova rozvoje pro diskretní Lattice Boltzmannovu rovnici vyplývá požadavek slabé stlačitelnosti, tedy

¹U vektorových veličin se volí jen jedna složka charakteristické veličiny a jeden konverzní faktor, kterým se všechny složky přeškálují.

$u_{max} \ll c_s$, viz [8]. Tomu je třeba uzpůsobit u . Zpravidla se volí $u \leq 0,1$, aby bylo dosaženo dostatečně přesných výsledků. Protože Reynoldsovo číslo musí být v obou soustavách stejné, lze vypočítat viskozitu v bezrozměrných jednotkách dle:

$$\nu = \frac{lu}{Re}. \quad (8.5)$$

Při volbě BGK operátoru však existuje závislost mezi relaxačním parametrem τ a viskozitou ν vyplývající z (4.10) a (6.6)

$$\nu = c_s^2 \left(\tau - \frac{1}{2} \right), \quad (8.6)$$

v rozměrových jednotkách pak dostaneme:

$$\hat{\nu} = \hat{c}_s^2 \left(\hat{\tau} - \frac{1}{2} \right) \frac{\Delta \hat{x}^2}{\Delta \hat{t}}. \quad (8.7)$$

Vyjádříme τ z (8.6), kde bylo dosazeno (8.5),

$$\tau = \frac{lu}{Rec_s^2} + \frac{1}{2} \quad (8.8)$$

a z převodu rychlostí vypočteme také konverzní faktor času a časový krok:

$$\begin{aligned} \hat{C}_t &= \frac{\hat{C}_l u}{\hat{u}}, \\ \Delta \hat{t} &= \hat{C}_t \Delta t. \end{aligned} \quad (8.9)$$

Konverzní faktor hustoty dostaneme standardně:

$$\hat{C}_\rho = \frac{\hat{\rho}}{\rho_0}. \quad (8.10)$$

Další veličina, kterou je třeba dopočít, je tlak. Vyjdeme ze stavové rovnice pro izotermní proudění v tomto tvaru:

$$p = c_s^2 \rho, \quad (8.11)$$

kde $c_s = \sqrt{RT}$.

V izotermním modelu není třeba uvažovat rovnici energie, ale jen rovnici kontinuity a hybnosti, kde se tlak p nevyskytuje, pouze jeho změny ∇p . Hustotu rozepíšeme na střední hodnotu ρ_0 a její odchylky ρ' :

$$\rho = \rho_0 + \rho' \quad (8.12)$$

a stavovou rovnici aplikujeme právě na odchylky od středních hodnot, viz [8]:

$$p' = c_s^2 \rho'. \quad (8.13)$$

Pro rozměrový tlak pak platí:

$$\hat{p} = \hat{p}_0 + p' \hat{C}_p, \quad (8.14)$$

kde

$$\hat{C}_p = \frac{\hat{C}_\rho \hat{C}_l^2}{\hat{C}_t^2}. \quad (8.15)$$

Hodnota skutečného tlaku \hat{p}_0 může být tedy libovolná a nemá vliv na izotermní proudění. Ostatní odvozené veličiny a jejich konverzní faktory se určí rozkladem do základních jednotek a výpočtem z konverzních faktorů pro čas (resp. rychlost), délku a hustotu (příp. hmotnost).

Z uvedeného vyplývá, že nelze zvolit všechny tři hlavní parametry simulace, Δx , Δt , τ nezávisle na sobě, což proces nastavení parametrů může mírně zkomplikovat.

U simulace uživatel zpravidla požaduje, aby byla přesná, stabilní a zároveň netrvala příliš dlouho. Výše uvedené může být ovlivněno nastavením následujících parametrů (převzato z [8]):

1. Přesnost

- τ by nemělo být moc větší než 1
- $u_{max} < 0,1$
- Δx by mělo být dostatečně malé, aby obsáhlo všechny rysy daného proudění (např. zahrnutí vlivu malých překážek, víry malých rozměrů apod.)
- Δt by mělo být dostatečně malé, aby nevznikla příliš velká chyba způsobená diskretizací

2. Stabilita

- $u_{max} < 0,4$
- $\tau > 0,5^2$
- Reynoldsovo číslo mřížky: $Re_g = u_{max} \Delta x / \nu \propto O(10)$, aby byly simulovány případné lokální víry

3. Rychlost simulace

- Δx a Δt by neměly být příliš malé (velikost paměti $\propto \Delta x^{-d}$, výpočetní čas $\propto \Delta t^{-1} \Delta x^{-d}$, kde d je počet dimenzí)

Uvedená pravidla se však můžou lišit v závislosti na typu řešeného problému, uvažované geometrii a v praxi bývají využívány pouze orientačně, čehož se budeme držet i v této práci. Také postup pro přepočítání do bezrozměrného systému, který byl uveden, není jediný. Podstatná je rovnost Reynoldsových čísel (8.1) pro obě jednotkové soustavy a pro zdárnou simulaci také splnění výše uvedených podmínek.

²Pro $\tau < 0,55$ musí platit $\tau > 1/2 + u_{max}/8$, tedy s klesajícím τ klesá maximální povolená rychlost.

Samotný algoritmus simulace se skládá z následujících částí:

8.2 Algoritmus výpočtu

1. Volba parametrů viz předchozí kapitola.
2. Inicializace a načtení počátečních podmínek ($f = f_i^{eq}$ pro $\rho = 1$ a $\mathbf{u} = 0$).
3. Výpočet f_i^{eq} .
4. Kolize.
5. Proudění, zahrnutí okrajových podmínek.
6. Výpočet makroskopických momentů ρ a \mathbf{u} v každém bodě výpočetní mřížky.
7. Ukládání dat dle potřeby.
8. Zvýšení časového kroku o Δt a návrat k bodu 3. dokud není dosaženo požadovaného času.

Tento postup bude demonstrován na jednoduchém příkladu izotermního proudění čtvercové kavity, ke kterému byl napsán zdrojový kód v jazyce c++ (viz příloha) a byla provedena simulace. Posloužila pro názorné pochopení algoritmu. Zároveň tento příklad byl zvolen s ohledem na dostatek dostupných dat, se kterými lze výsledky simulace Lattice Boltzmannovou metodou srovnat.

9 Kavita

9.1 Fyzikální popis problému

Uvažujeme dvourozměrnou čtvercovou nádobu plně naplněnou kapalinou. Horní stěna se pohybuje rychlostí $\hat{u}_w = 6$ m/s zleva doprava, která roztáčí kapalinu uvnitř, ostatní stěny jsou v klidu. Rozměry stěny nádoby jsou $\hat{x} = \hat{y} = 0,2$ m. Kapalina má kinematickou viskozitu $\hat{\nu} = 0,0012$ m²/s. Na stěnách uvažujeme, že kapalina proudí stejnou rychlostí jako stěna, tedy okrajové podmínky typu no-slip. Cílem je namodelovat proudění kapaliny uvnitř nádoby.

9.2 Simulace

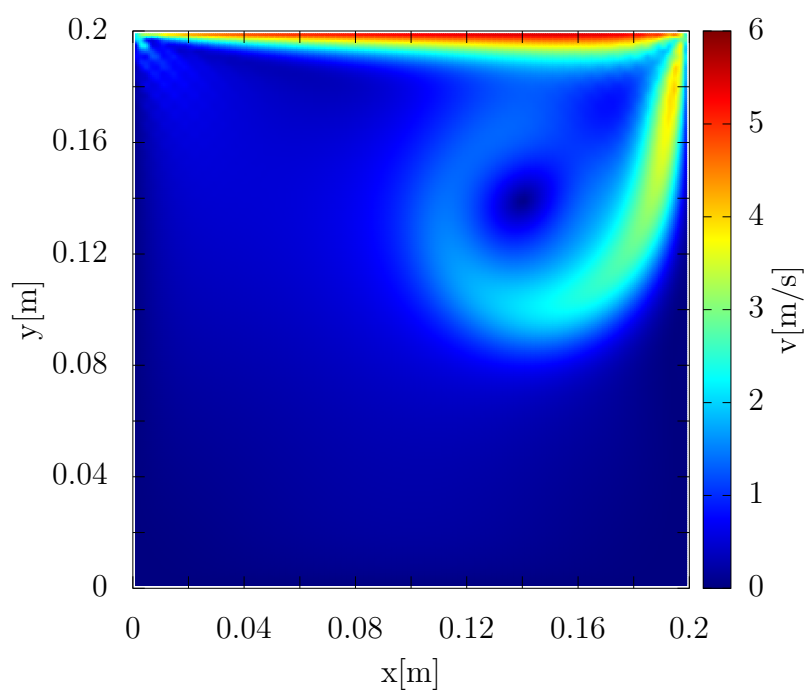
Vypočteme Reynoldsovo číslo, za charakteristické rozměry volíme následující parametry:

$$Re = \frac{\hat{l}\hat{u}}{\hat{\nu}} = \frac{\hat{x}\hat{u}_w}{\hat{\nu}} = 1000 \quad (9.1)$$

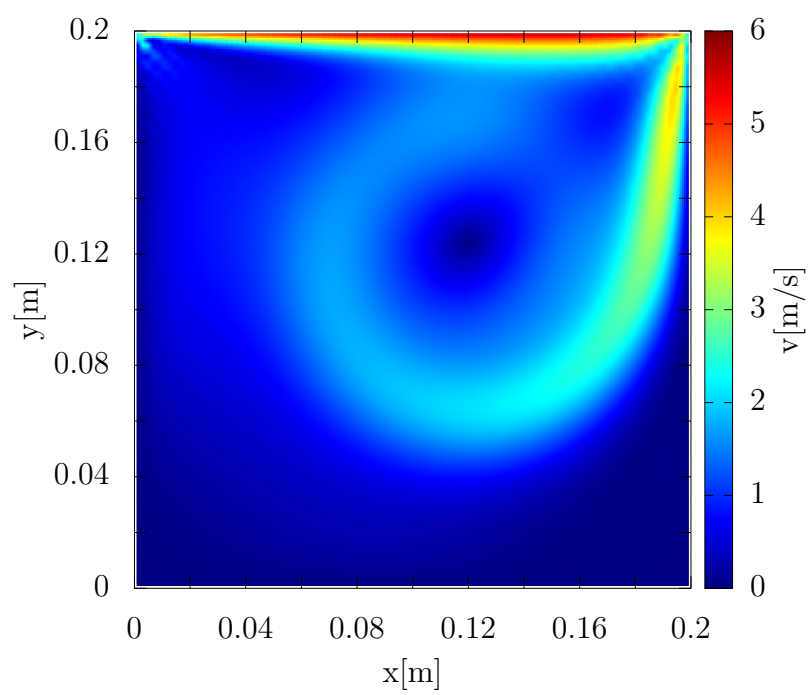
Jedná se o izotermní problém bez vedení tepla, použijeme operátor BGK a diskretizační rychlostní set D2Q9. Na stěnách volíme okrajové podmínky typu bounce back (7.1) a u horní stěny, která se pohybuje, okrajové podmínky (7.2). Volíme výpočetní síť 100x100, čímž dostaneme $\Delta\hat{x} = \Delta\hat{y} = 0,002$ m. Konverzní faktor pak je $\hat{C}_l = 0,002$ m. Rychlost kapaliny při horní stěně, jak se ukáže také ze simulace, bude maximální v celé doméně po celý čas simulace, proto ji lze zvolit dle podmínek $u_{max} = 0,1$. Konverzní faktor pro rychlost tedy dostaneme $\hat{C}_u = 60$ m/s. Nyní lze spočítat konverzní faktor pro čas, který odpovídá časovému kroku dle (8.9) a dostaneme $\Delta\hat{t} = 3,33 \cdot 10^{-5}$ s. Hodnotu relaxačního parametru určíme dle vzorce (8.8) a dostaneme $\tau = 0,53$. Zbývá určit počet časových kroků simulace. Volíme konečný čas $\hat{t}_0 = 1,5$ s. Podělením \hat{t}_0 převodním faktorem pro čas $\hat{C}_t = \Delta\hat{t}$ zjistíme, že čas simulace odpovídá přibližně 45000 časových kroků. Zdrojový kód simulace včetně komentářů je v příloze A.1.

9.3 Výsledky

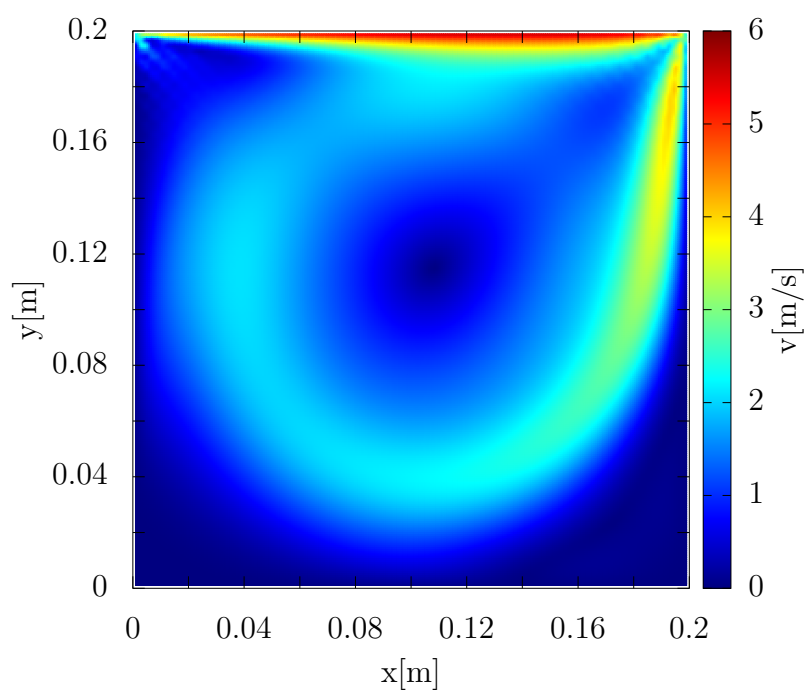
Program ukládá požadované výsledky, v tomto případě rychlosti a hustoty v daných bodech. Na obr. 9.1 až 9.4 jsou znázorněny rychlostní pole pomocí programu Gnuplot v časech 0,15 s, 0,3 s, 0,6 s a 1,5 s. Na těchto obrázcích je názorně vidět, že v pravém horním rohu domény lze pozorovat tvorbu pravotočivého víru, který se postupně zvětšuje a přesouvá se doprostřed nádoby. Zároveň zhruba od času 1 s již nedochází k velkým změnám rychlostního pole.



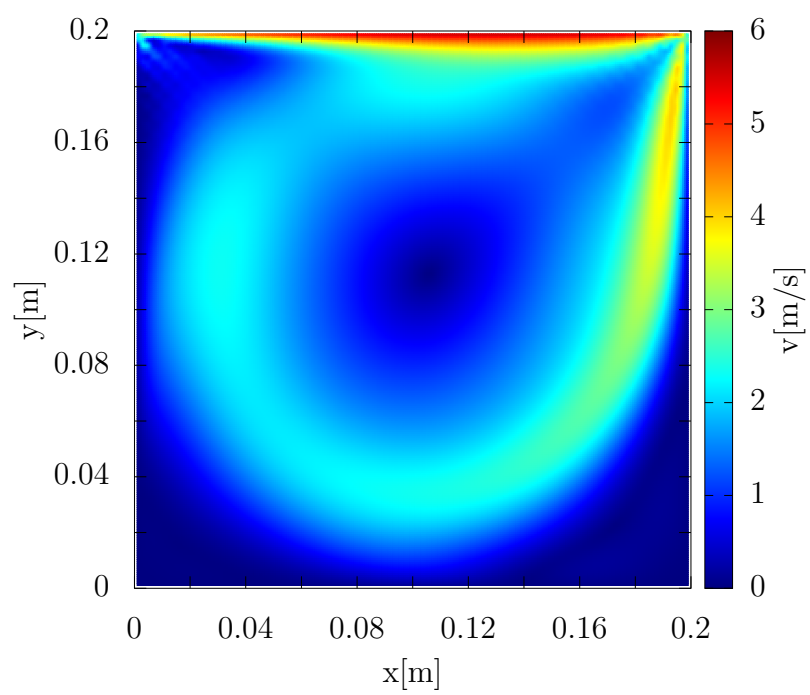
Obr. 9.1: Znázornění rychlostního pole kavity v čase 0,15 s.



Obr. 9.2: Znázornění rychlostního pole kavity v čase 0,3 s.

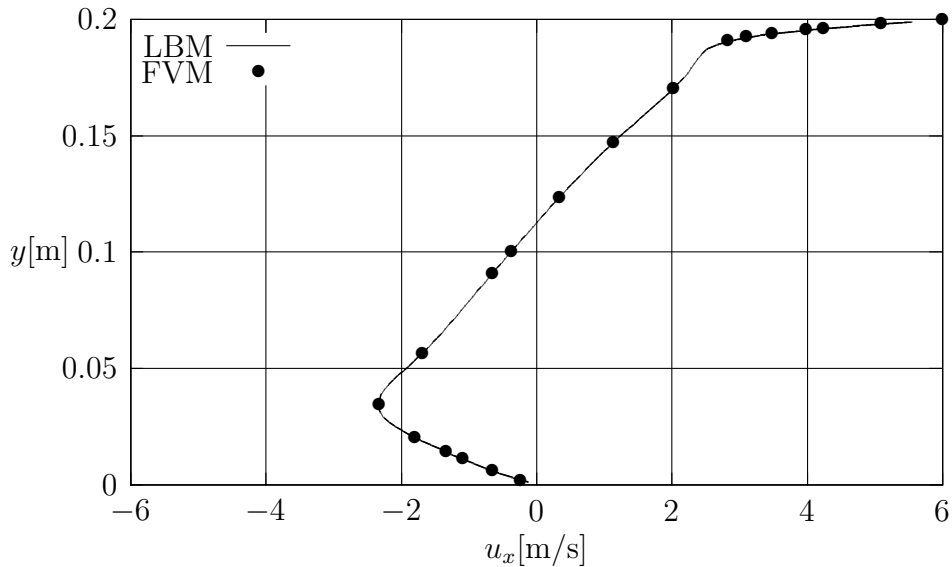


Obr. 9.3: Znázornění rychlostního pole kavity v čase 0,6 s.

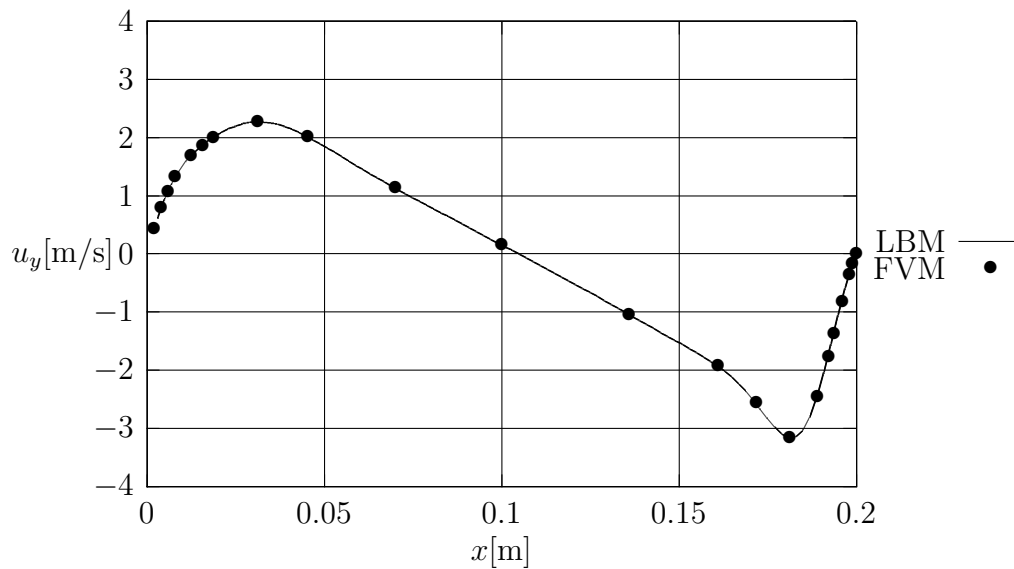


Obr. 9.4: Znázornění rychlostního pole kavity v čase 1,5 s.

Správnost výsledků a také zdrojového kódu vytvořené simulace byla verifikována s dostupnými daty [1]. Byly pro to vybrány hodnoty rychlostí na střední vertikální a horizontální ose nádoby. Porovnané hodnoty jsou graficky vykresleny na obr. 9.5 a 9.6. V obou případech jsou odchylky zanedbatelné.



Obr. 9.5: Verifikace modelu pro x složku rychlosti na řezu $y = 0,10$ m s daty z [1]



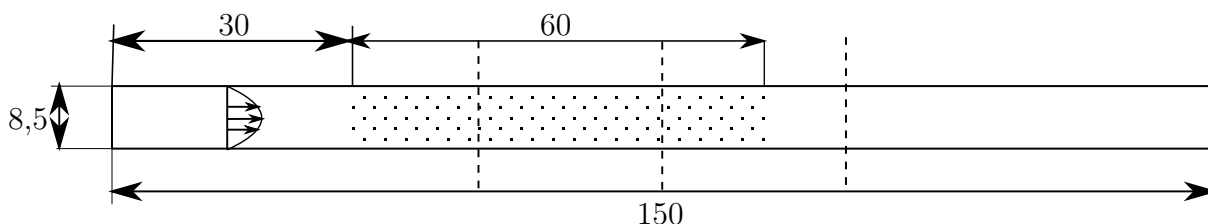
Obr. 9.6: Verifikace modelu pro y složku rychlosti na řezu $x = 0,10$ m s daty z [1]

Protože uvedený výpočet probíhal pouze na jednom jádře procesoru a hlavní simulace práce - obtékání překážek - proběhne na větší výpočetní síti a po delší čas, bude pro další simulace využíván dostupný software openLB, který paralelní výpočty umožňuje.

10 Obtékání překážek

10.1 Fyzikální popis problému

Nyní přistoupíme k popisu druhého problému, který byl v rámci této práce simulován. Uvažujeme obdélníkový kanál o šířce $\hat{y} = 8,5$ mm a délce $\hat{x} = 150$ mm, viz obr. 10.1.



Obr. 10.1: Grafické znázornění celkové geometrie, rozměry v mm, čárkované čáry znázorňují srovnávací řezy, viz dále.

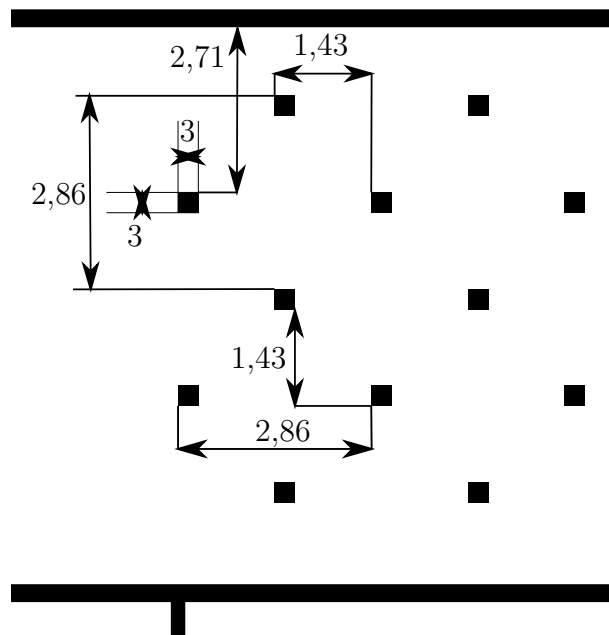
Uprostřed domény jsou umístěny čtvercové překážky o šířce strany $\hat{a} = 0,3$ mm v řadách střídavě po dvou a třech překážkách. Jednotlivé řady jsou vůči sobě posunuty, aby došlo ke střídavému rozmístění viz obr. 10.2. Do domény proudí zleva plyn o kinematické viskozitě $\hat{\nu} = 1,5 \cdot 10^{-5}$ Pa s⁻¹ s Poiseuillovým parabolickým profilem (viz [14]) o maximální rychlosti $\hat{u} = 1$ m s⁻¹. Horní a dolní stěnu uvažujeme nepropustnou s ideálním třením odpovídající okrajové podmínce no-slip. Stejně podmínky předpokládáme také na povrchu jednotlivých překážkách. Na pravé straně proudí tekutina ven z domény.

10.2 Simulace

Doménu rozdělíme na uzly o vzdálenosti $\Delta\hat{x} = 0,03$ mm. Dohromady dostáváme tak přes 1 000 000 výpočetních uzlů. Použijeme i zde rychlostní set D2Q9. Na stěnách a na překážkách uvažujeme okrajovou podmínku bounce-back (7.1) a na vstupu do domény okrajovou podmínku (7.2). Na výtoku volíme tlakovou podmínku (7.3). Maximální rychlost v průběhu celé simulace předpokládáme $\hat{u}_{max} = 2$ m s⁻¹. Provedením stejného postupu převodu soustavy do bezrozměrných jednotek jako v předchozí kapitole dostáváme následující hodnoty simulačních parametrů:

$$\begin{aligned}\Delta\hat{t} &= 7,5 \cdot 10^{-7} \text{ s,} \\ \tau &= 0,5375, \\ u_{max} &= 0,05.\end{aligned}$$

Budeme simulaci provádět až do času $\hat{t}_0 = 3$ s. Simulace proběhne pomocí softwaru openLB, kde je implementován výpočet na více jádrech procesoru najednou a tím se



Obr. 10.2: Grafické znázornění geometrie - detail překážek, rozměry v mm.

doba výpočtu simulace značně zkrátí (zdrojový kod lze nalézt v příloze A.2). Pro hladký průběh je vhodné volit postupné zvyšování rychlosti na vstupu do domény. Čas dosažení maximální rychlosti na vtoku stanovíme na $\hat{t}_{in} = 0,1$ s.

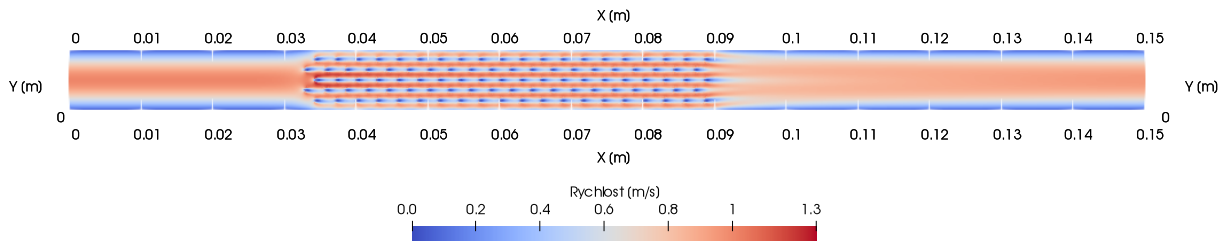
Z výsledků simulace se ukáže, že dané proudění je stacionární a s rostoucím časem se rychlostní pole nemění. Proudění má laminární charakter, jak vyplývá také z hodnoty Reynoldsova čísla pro obtékání překážek:

$$Re = \frac{\hat{a}\hat{u}}{\hat{\nu}} = 20. \quad (10.1)$$

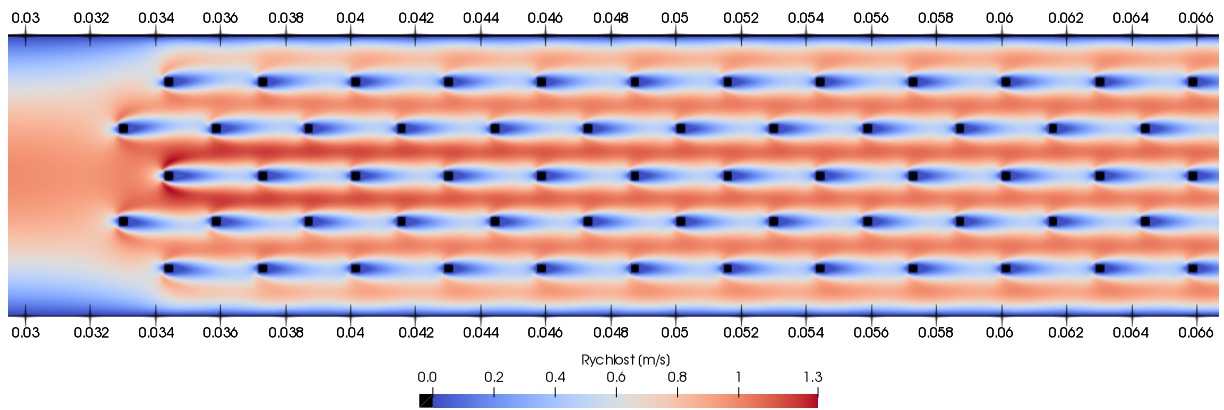
10.3 Výsledky

Na obr. 10.3 je znázorněno rychlostní pole celé domény po ustálení. Můžeme vidět, že při obtékání překážek je rychlost za jednotlivými překážkami nižší. Zároveň na podélných řezech domény, kde se překážky nenachází, se rychlost příliš nemění. Za poslední řadou překážek lze vyzorovat rychlostní stíny, kde jsou rychlosti nižší a se zvětšující se vzdáleností od překážek se rychlost zvyšuje a postupně se obnovuje Poiseullovu proudění. Maximální rychlost lze pozorovat na druhé řadě okolo prostřední překážky, což odpovídá teoretickým předpokladům, protože uprostřed kanálu je proudění nejrychlejší.

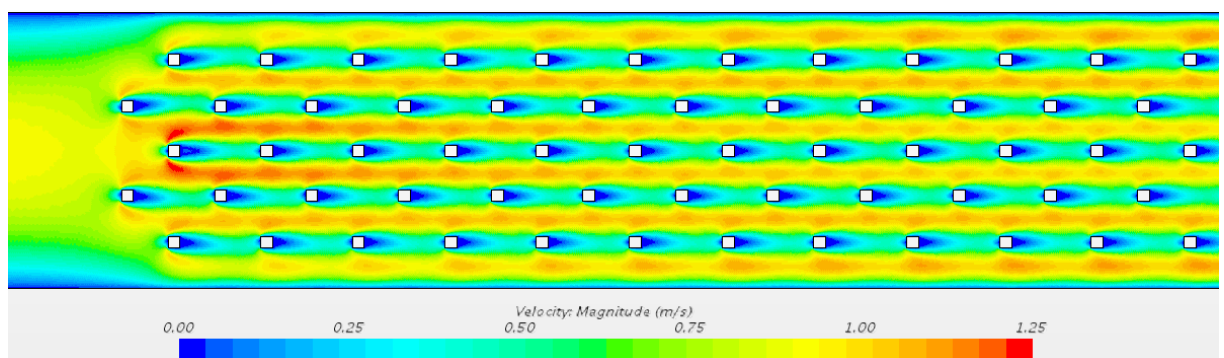
Pro validaci výsledků byl tento problém také simulován metodou konečných objemů (FVM) v prostředí starCCM+. Na obr. 10.4 a 10.5 jsou znázorněna rychlostní pole překážkové části oběma metodami. Jak je vidět z obrázku, proudění má v obou případech



Obr. 10.3: Znázornění rychlostního pole, absolutní rychlost.

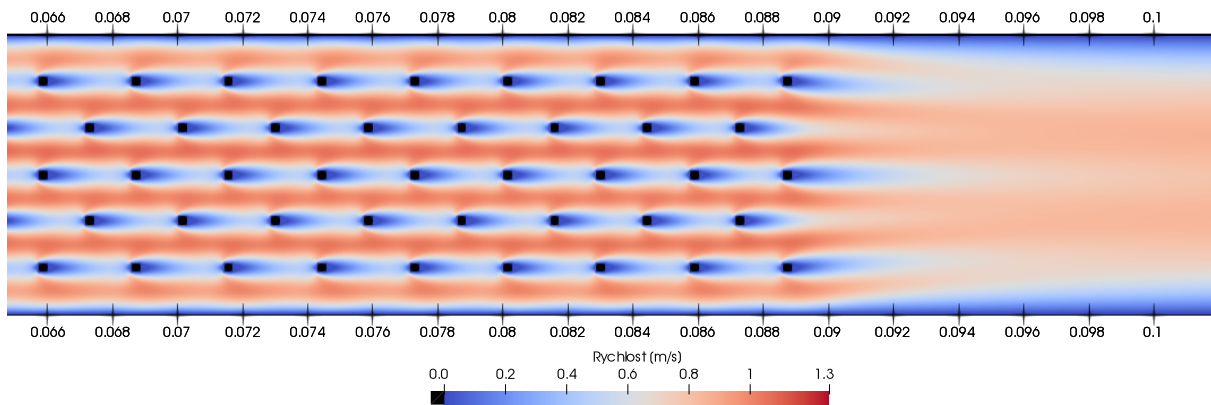


(a) LBM

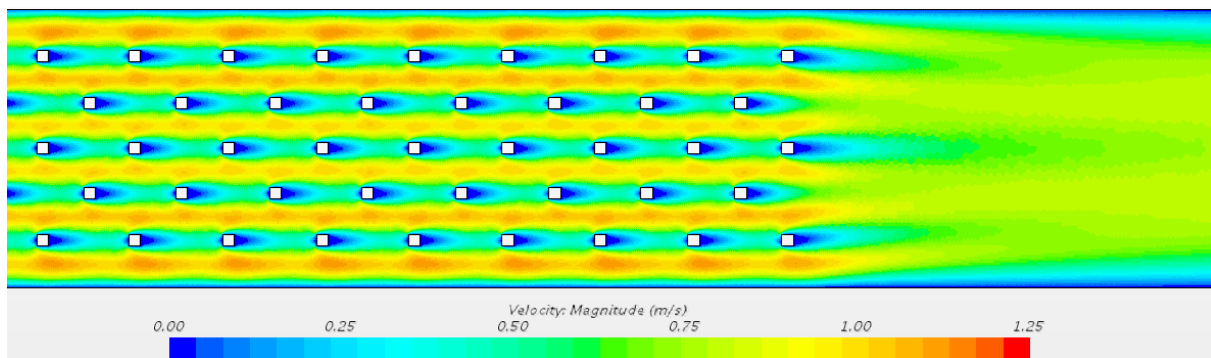


(b) FVM

Obr. 10.4: Znázornění rychlostního pole, detail předních překážek, absolutní rychlost.



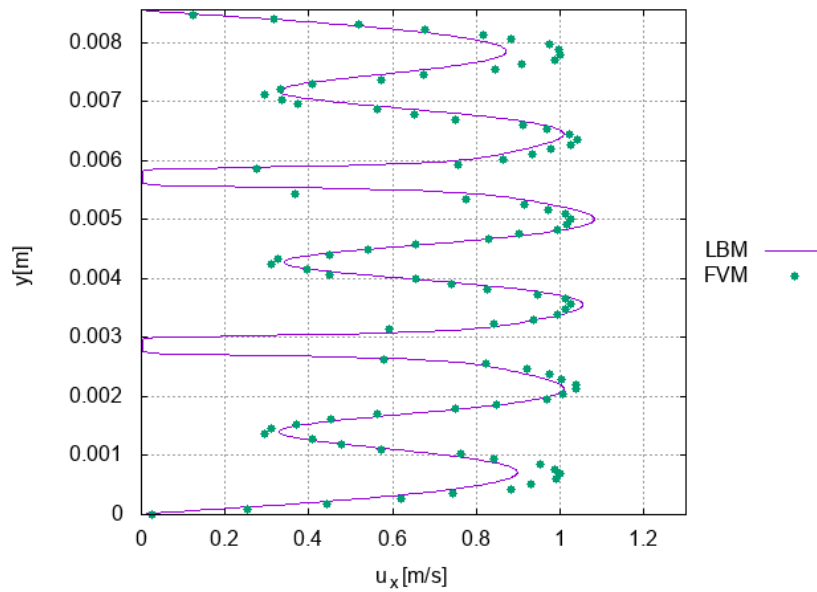
(a) LBM



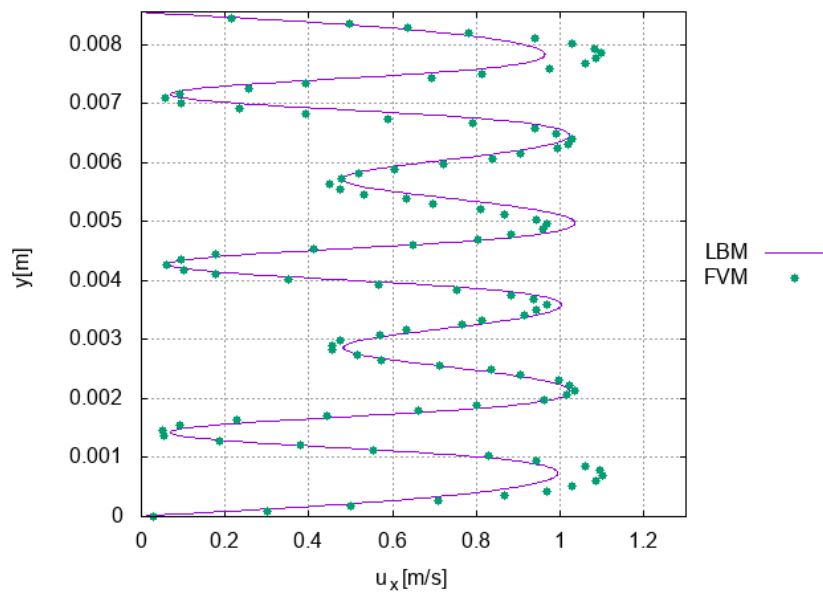
(b) FVM

Obr. 10.5: Znáznornění rychlostního pole, detail zadních překážek, absolutní rychlost.

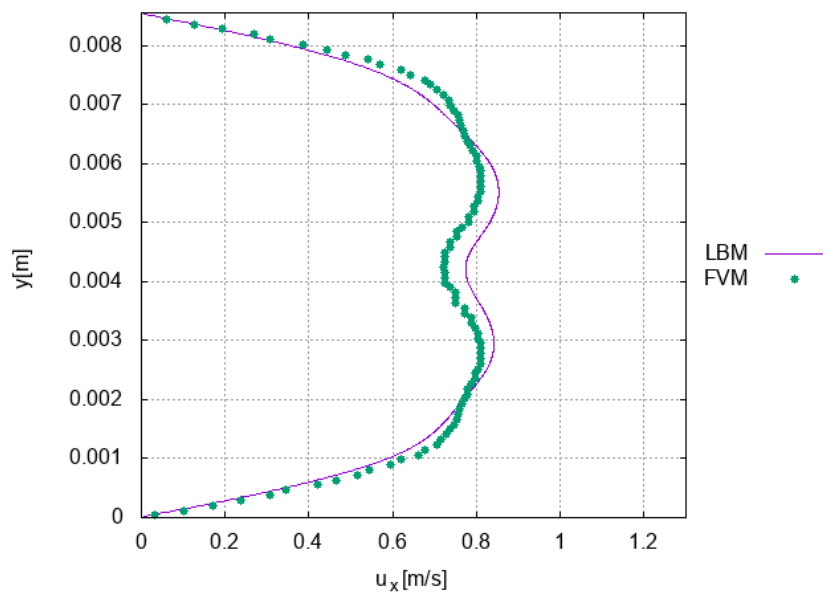
laminární charakter a z grafu nelze vypořádat signifikantnější rozdíly mezi výsledky obou metod. Pro detailnější analýzu byly zvoleny příčné řezy doménou o x -ových souřadnicích 0,05 m, 0,075 m a 0,1 m, viz obr. 10.1, na kterých porovnáme jednotlivé složky rychlostí u_x a u_y . Porovnání výsledku z openLB s vybranými body ze simulace pomocí konečných objemů je znázorněno na obr. 10.6 až 10.11. Jak z grafů vyplývá, hodnoty rychlostí v obou simulacích se příliš neliší, řádově o jednotky %, což je akceptovatelná odchylka, protože se již jedná o složitější problém s komplexnější geometrií. Případná další analýza vyžaduje různé volby simulačních parametrů obou metod a případné srovnání s experimentem, což je nad rámec této práce.



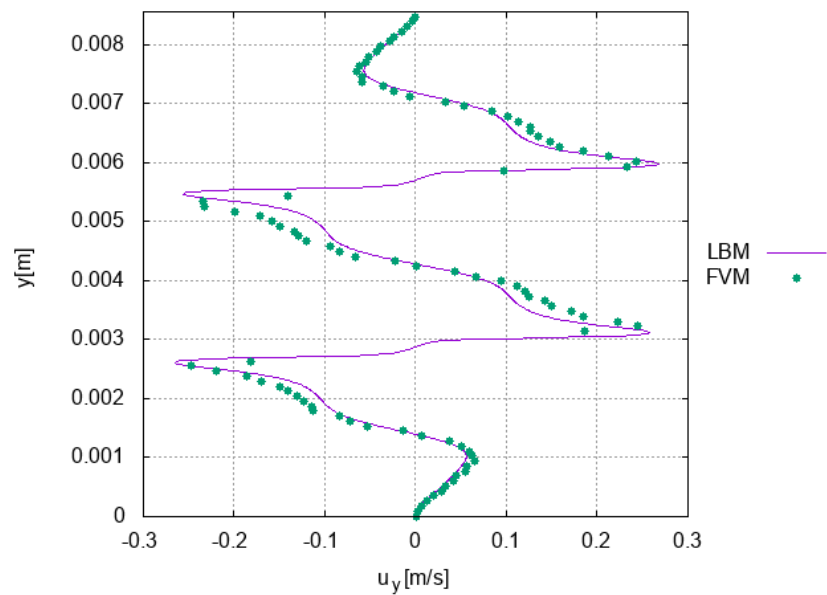
Obr. 10.6: Srovnání rychlostí u_x na příčném řezu $x = 0,05$ m.



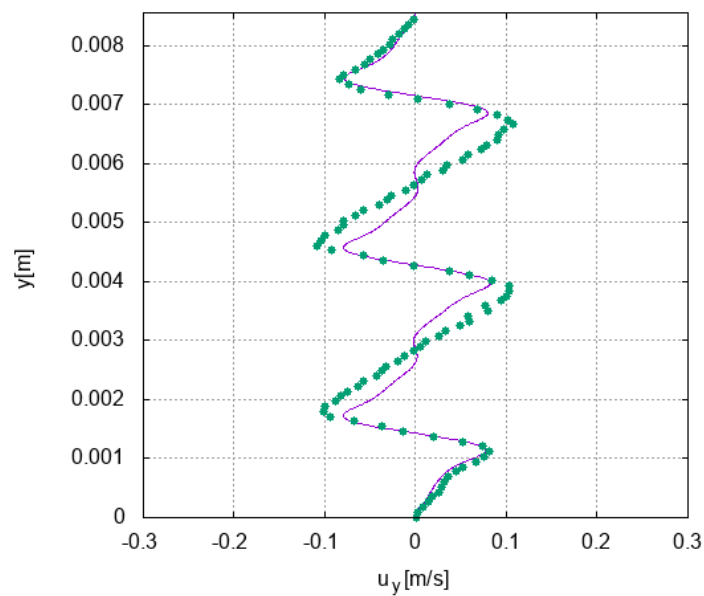
Obr. 10.7: Srovnání rychlostí u_x na příčném řezu $x = 0,075$ m.



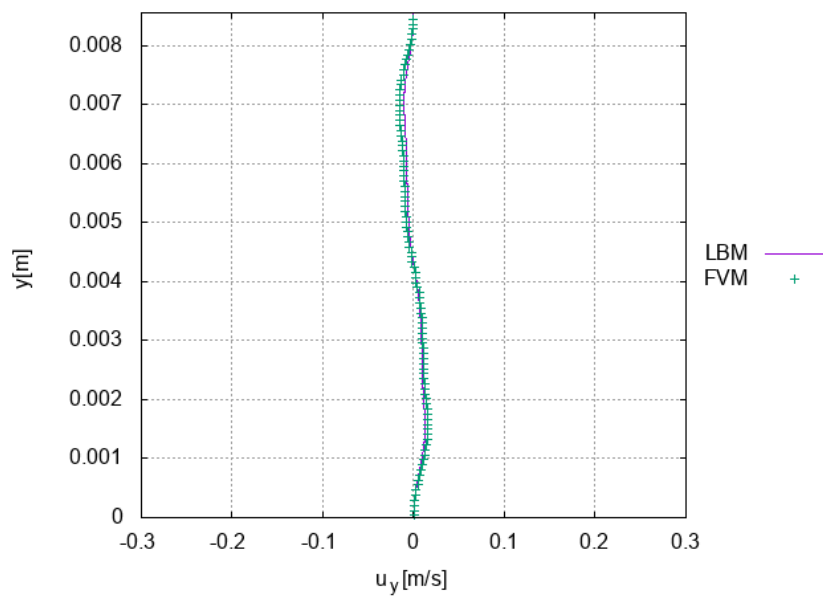
Obr. 10.8: Srovnání rychlostí u_x na příčném řezu $x = 0,1$ m.



Obr. 10.9: Srovnání rychlostí u_y na příčném řezu $x = 0,05$ m.



Obr. 10.10: Srovnání rychlostí u_y na příčném řezu $x = 0,075$ m.



Obr. 10.11: Srovnání rychlostí u_y na příčném řezu $x = 0,1$ m.

Závěr

Cílem této práce bylo popsat a odvodit Lattice Boltzmannovu metodu a následně ji implementovat na úlohách proudění tekutin a výsledky srovnat s jinou numerickou metodou. V teoretické části byly nejprve uvedeny makroskopické rovnice zákonů zachování hmotnosti, hybnosti a energie. Boltzmannova rovnice popisuje pravděpodobnostní pohyb částic pomocí rozdělovací funkce. Integrací této rovnice bylo ukázáno propojení s rovnicemi zákonů zachování. Obecný kolizní operátor byl z důvodu obtížné implementace nahrazen aproximací v podobě operátoru BGK. Pro dosažení konkrétního modelu, v našem případě se jednalo o Navier-Stokesův model, byla Boltzmannova rovnice rozepsána pomocí Chapman Enskogovy poruchové teorie až do prvního řádu a za využití Hermitových polynomů byly příslušné řády vyčísleny. Výsledkem odvození byly již zmíněné Navier-Stokesovy rovnice. Za účelem numerické implementace byla provedena diskretizace Boltzmannovy rovnice. Rychlostní prostor byl diskretizován za pomoci Gauss-Hermitovy kvadraturní formule, takže integrály byly nahrazeny konečnými sumami. Při diskretizaci prostoru a času bylo využito metody trajektorií a obdélníkové a lichoběžníkové aproximace integrálů. To vedlo na Lattice Boltzmannovu rovnici a příslušný výpočetní algoritmus.

Na závěr teoretické části byly popsány počáteční a okrajové podmínky s důrazem na typy podmínek, které byly v této práci v numerických aplikacích použity.

V praktické části této práce jsou nejprve uvedeny obecné zásady simulace Lattice Boltzmannovou metodou - podmínky přeskálování do bezrozměrné soustavy, podmínky stability a přesnosti. Poté jsou touto metodou vyřešeny dvě úlohy.

V prvním simulovaném problému se jednalo o proudění kavity v dvourozměrné čtvercové doméně s pohybující se horní stěnou. Problém byl naprogramován v jazyce C++ a následně byly výsledky srovnány s dostupnými daty získanými metodou konečných objemů. Ukázalo se, že sekvenční výpočet není časově příliš efektivní, proto u hlavního problému této práce - obtékání překážek - byl využit software openLB umožňující paralelní výpočet na více jádrech procesoru. Tento problém byl nasimulován Lattice Boltzmannovou metodou a také metodou konečných objemů. Výsledná rychlostní pole byla v obou případech laminární a hodnoty se příliš nelišily. V rámci detailnější analýzy byly porovnávány rychlosti na vybraných řezech doménou. Relativní chyby byly v řádu jednotek %, což vzhledem ke komplexnější geometrii dané úlohy lze považovat za uspokojivé. Pro další validaci by bylo možné měnit simulační parametry obou modelů, případně uvedenou úlohu realizovat také experimentálně.

Aplikaci obtékání překážek kolem překážek by bylo možné dále simulovat při vyšších rychlostech turbulentního proudění, případně zvolit jiný kolizní operátor nebo nechat na proudění působit také objemové síly.

Literatura

- [1] *ACENUMERICS, Collection of numerical projects*, [online]. [cit. 30.4.2020]. Dostupné z <https://www.acenumerics.com/>.
- [2] *BRDIČKA, Miroslav, Ladislav SAMEK a Bruno SOPKO. Mechanika kontinua*. Vyd. 4., rev. a upr. Praha: Academia, 2011. Gerstner. ISBN 978-80-200-2039-0.
- [3] *CALOGERO, S. Lectures notes on Boltzmann's equation.*, [online]. [cit. 25.5.2020]. Dostupné z: http://www.math.chalmers.se/~calogero/Boltzman_notes.pdf.
- [4] *CERCIGNANI, Carlo. The Boltzmann Equation and Its Applications. Applied Mathematical Sciences*, vol 67. Springer, New York, NY, 1988.
- [5] *ČERMÁK, Libor. Výpočtová dynamika tekutin*. Brno, 2019, 137s. Učební texty vysokých škol.
- [6] *JOUBERT, Gerhard; LEATHER, Hugh; PARSONS, Mark; PETERS, Frans; SAWYER, Christopher. Parallel Computing: On the Road to Exascale*. Netherlands : IOS Press, 2016. 872 s. (Advances in Parallel Computing; Vol. 27).
- [7] *KRAUSE, Mathias J. OpenLB – Open Source Lattice Boltzmann Code*. 2006-2020, [online].[cit. 21.03.2020]. Dostupné z: www.openlb.cz
- [8] *KRÜGER, Timm & Co. The lattice Boltzmann method: principles and practice*. New York, NY: Springer Berlin Heidelberg, 2016. ISBN 9783319446479.
- [9] *KURISCAK, Pavel. Diplomová práce - Simulace proudění neneutronovských tekutin pomocí lattice Boltzmannovy metody*. Praha. 2011, 88s. Univerzita Karlova v Praze.
- [10] *PERUMAL D. Arumuga, ANOOP K. Dass. A Review on the development of lattice Boltzmann computation of macro fluid flows and heat transfer*. Alexandria Engineering Journal. (2015) 54, 955 – 971
- [11] *RUBINSTEIN, Robert and LI-SHI Luo. Theory of the lattice Boltzmann equation: Symmetry properties of discrete velocity sets*. Phys. Rev. E 77, 036709.
- [12] *STRUCHTUP, H. Macroscopic Transport Equations for Rarefied Gas Flows*.Springer-Verlag, Berlin Heidelberg, 2005.
- [13] *VIGGEN, Erlend Magnus. The lattice Boltzmann method: Fundamentals and acoustics. Disertační práce*. Trondheim, 2014.
- [14] *WIKIPEDIA, the free encyklopedia, Hagen–Poiseuille equation*. [online].[cit. 1.6.2020]. Dostupné z: https://en.wikipedia.org/wiki/Hagen%E2%80%93Poiseuille_equation.

Seznam použitých zkratek a symbolů

u	makroskopická rychlost
ρ	hustota
t	čas
∇	nabla
F	síly objemové
x	polohový vektor
p	hybnost
V	kontrolní objem
f	měrná síla
div	divergence
$\frac{D}{Dt}$	materiálová derivace
σ, σ_{ij}	tensor napětí
p	tlak
η	1. koeficient viskozity
η_B	objemová viskozita
ζ	2. koeficient viskozity
δ_{ij}	Kroneckerovo delta
E	energie celková
e	energie měrná vnitřní
T	teplota
R	plynová konstanta
q	plošná hustota tepelného toku
λ	součinitel tepelné vodivosti
P	distribuční funkce
f	rozdělovací funkce
f_i	rozdělovací funkce diskrétní
ξ	mikroskopická rychlost
N	počet částic
m_0	hmotnost jedné částice
$\Omega(f, f)$	kolizní operátor
g	vektor relativní rychlosti
k	jednotkový vektor
J	jakobián
det	determinant
Θ	kolizní úhel
ε	kolizní úhel, parametr poruchové teorie
φ	kolizní invariant
\mathbb{R}	obor reálných čísel
s	entropie

H	Boltzmannova entropie
$f^{eq}, f^{(eq)}$	rovnovážné rozdělení
$f^{neq}, f^{(neq)}$	nerovnovážná část rozdělovací funkce
$\Omega_B(f, f)$	kolizní operátor BGK
\mathbf{v}	relativní mikroskopická rychlost
Re	Reynoldsovo číslo
Kn	Knudsenovo číslo
$\langle f, g \rangle$	skalární součin
H, \mathbf{H}	Hermitovy polynomy
\mathbf{a}	koefficienty k Hermitovým polynomům
ω	váhová funkce
ω_i	váhy
$\mathbb{Q}(f)$	kvadraturní formule
\mathbf{c}_i	mikroskopická rychlost
Δx	rozměr prostorové buňky
Δt	časový krok
\mathbf{n}	normálový vektor
\mathbf{t}	vektor tečny
ν	kinematická viskozita
C	konverzní faktor
d	počet rozměrů
\cdot	skalární součin
T	transpozice
Δ	operátor Laplace
$\nabla \cdot$	divergence

Seznam příloh

A Zdrojové kody	101
A.1 Kavita	101
A.2 Obtékání překážek	106

A Zdrojové kody

A.1 Kavita

```
#include <iostream>
#include <array>
#include <cmath>
#include <vector>
#include <iostream>
#include <string>
#include <fstream>
#include <cstdio>
#include <ctime>

using namespace std;

//skalární součin
double dot(vector <double> a, vector <double> b)
{
    double res = 0;
    if (a.size() != b.size())
        cout << "Error scalar product!Different size" << endl;
    for (int i = 0; i < a.size(); i++)
        res += a[i] * b[i];
    return res;
}

//suma komponent vektoru
double sum(vector <double> vec)
{
    double res = 0;
    for (int i = 0; i < vec.size(); i++)
        res += vec[i];
    return res;
}

//násobení skalárem
vector <double> mult(vector<double> vec, double a)
{
    for (int i = 0; i < vec.size(); i++)
        vec[i] = a * vec[i];
    return vec;
}

int main()
```

```

{
clock_t start;
double duration;
start = std::clock();
//D2Q9 rychlostní set
const int k0{ 9 }; //number of directions o f_k
const vector < double > w = { 4. / 9., 1. / 9., 1. / 9., 1. / 9.,
    1. / 9., 1. / 36., 1. / 36., 1. / 36., 1. / 36. };
//Volba parametrů
double alpha = 0.01;
double omega = 1. / (3.*alpha + 0.5); //1/tau

int x0 = 100; //počet uzlů na šířku domény
int y0 = 100; //počet uzlů na výšku domény
int t0 = 45000; //odpovídá 1,5s
double velo_conv = 60.; //přepočet rychlosti
double length_conv = 0.2; //přepočet délky
double rho_num = 1.0; //hustota
vector < double > u_w = { 0.1, 0. }; //rychlost horní stěny
vector < vector < double > > c = { //rychlosti c_k
    vector < double > { 0, 0 },
    vector < double > { 1, 0 },
    vector < double > { 0, 1 },
    vector < double > { -1, 0 },
    vector < double > { 0, -1 },
    vector < double > { 1, 1 },
    vector < double > { -1, 1 },
    vector < double > { -1, -1 },
    vector < double > { 1, -1 },
};
double c_s = (1. / sqrt(3.));
//Inicializace vektorů rychlosti, hustoty a rozdělovací funkce
vector < double > ones(9, 1.);
vector < vector < vector < double > > > v(x0, vector < vector < double > > (y0,
    vector < double > (2, 0))); //rychlost [x][y][složka]
vector < vector < double > > rho(x0, vector < double > (y0, 0));
    //density [x][y]
vector < vector < double > > vabs(x0, vector < double > (y0, 0));
    //absvelocity [x][y]
vector < vector < vector < double > > > feq(x0,
    vector < vector < double > > (y0, vector < double > (k0, 0)));
    //rozdělovací funkce v rovnováze [x][y][složka]
vector < vector < vector < double > > > f(x0, vector < vector < double > > (y0,
    vector < double > (k0, 0))); //rozdělovací funkce [x][y][složka]

```

```

vector <vector<vector<double>>> fcoll(x0,
    vector<vector<double>>(y0, vector<double>(k0, 0)));
    //rozdělovací funkce [x][y][složka]
//Počáteční podmínky
for (int x = 0; x < x0; x++)
    for (int y = 0; y < y0; y++)
    {
        rho[x][y] = rho_num;
        v[x][y] = { 0.,0. };
        for (int k = 0; k < k0; k++)
            f[x][y][k] = w[k] * rho[x][y] * (1. + dot(c[k], v[x][y]) /
                (c_s*c_s) + 0.5 * pow(dot(c[k], v[x][y]), 2) / pow(c_s,
                    4) - 0.5 * dot(v[x][y], v[x][y]) / pow(c_s, 2));
    }
//Krok algoritmu
for (int t = 0; t <= t0; t++)
{
    cout << "Step " << t << " running!" << endl;
//Výpočet rovnovážné funkce feq v každém bodě výpočetní mříže
    for (int x = 0; x < x0; x++)
        for (int y = 0; y < y0; y++)
            for (int k = 0; k < k0; k++)
                feq[x][y][k] = w[k] * rho[x][y] * (1 + dot(c[k], v[x][y])
                    / pow(c_s, 2) + 0.5 * pow(dot(c[k], v[x][y]), 2) /
                    pow(c_s, 4) - 0.5 * dot(v[x][y], v[x][y]) / pow(c_s,
                        2));
//Kolize
    for (int x = 0; x < x0; x++)
        for (int y = 0; y < y0; y++)
            for (int k = 0; k < k0; k++)
            {
                fcoll[x][y][k] = f[x][y][k] * (1 - omega) + feq[x][y][k] *
                    omega;
            }
//Proudění
    for (int x = 0; x < x0; x++)
        for (int y = 0; y < y0; y++)
        {
            f[x][y][0] = fcoll[x][y][0];
        }
    for (int x = 0; x < x0 - 1; x++)
        for (int y = 0; y < y0; y++)
        {
            f[x + 1][y][1] = fcoll[x][y][1];
        }
}

```

```

for (int x = 0; x < x0 - 1; x++)
  for (int y = 0; y < y0 - 1; y++)
  {
    f[x + 1][y + 1][5] = fcoll[x][y][5];
  }
for (int x = 0; x < x0; x++)
  for (int y = 0; y < y0 - 1; y++)
  {
    f[x][y + 1][2] = fcoll[x][y][2];
  }
for (int x = 0; x < x0 - 1; x++)
  for (int y = 1; y < y0; y++)
  {
    f[x][y][6] = fcoll[x + 1][y - 1][6];
  }
for (int x = 0; x < x0 - 1; x++)
  for (int y = 0; y < y0; y++)
  {
    f[x][y][3] = fcoll[x + 1][y][3];
  }
for (int x = 0; x < x0 - 1; x++)
  for (int y = 0; y < y0 - 1; y++)
  {
    f[x][y][7] = fcoll[x + 1][y + 1][7];
  }
for (int x = 0; x < x0; x++)
  for (int y = 0; y < y0 - 1; y++)
  {
    f[x][y][4] = fcoll[x][y + 1][4];
  }
for (int x = 1; x < x0; x++)
  for (int y = 0; y < y0 - 1; y++)
  {
    f[x][y][8] = fcoll[x - 1][y + 1][8];
  }
//Okrajové podmínky

//Dolní stěna
for (int x = 0; x < x0; x++)
{
  f[x][0][5] = fcoll[x][0][7];
  f[x][0][2] = fcoll[x][0][4];
  f[x][0][6] = fcoll[x][0][8];
}
//Horní stěna

```



```

for (int x = 0; x < x0; x++)
{
    f[x][y0 - 1][7] = fcoll[x][y0 - 1][5] - 2 * w[5] * rho[x][y0 -
        1] * dot(c[5], u_w) / pow(c_s, 2);
    f[x][y0 - 1][4] = fcoll[x][y0 - 1][2] - 2 * w[2] * rho[x][y0 -
        1] * dot(c[2], u_w) / pow(c_s, 2);
    f[x][y0 - 1][8] = fcoll[x][y0 - 1][6] - 2 * w[6] * rho[x][y0 -
        1] * dot(c[6], u_w) / pow(c_s, 2);
}
//Levá stěna
for (int y = 0; y < y0; y++)
{
    f[0][y][5] = fcoll[0][y][7];
    f[0][y][1] = fcoll[0][y][3];
    f[0][y][8] = fcoll[0][y][6];
}
//Pravá stěna
for (int y = 0; y < y0; y++)
{
    f[x0 - 1][y][7] = fcoll[x0 - 1][y][5];
    f[x0 - 1][y][3] = fcoll[x0 - 1][y][1];
    f[x0 - 1][y][6] = fcoll[x0 - 1][y][8];
}
//Výpočet makroskopických veličin
for (int x = 0; x < x0; x++)
{
    for (int y = 0; y < y0; y++)
    {
        rho[x][y] = sum(f[x][y]);
        double num = 0;
        for (int i = 0; i < k0; i++)
            num += f[x][y][i] * c[i][0];

        v[x][y][0] = 1. / rho[x][y] * num;

        num = 0;
        for (int i = 0; i < k0; i++)
            num += f[x][y][i] * c[i][1];
        v[x][y][1] = 1. / rho[x][y] * num;
        vabs[x][y] = sqrt(pow(v[x][y][0], 2) + pow(v[x][y][1], 2));
    }
}
//Ukládání dat v daných časech
if (t % (t0 / 20) == 0)
{

```

```

ofstream myfile;
string name4 = "data_" + to_string(t) + ".csv";
myfile.open(name4);
for (int y = y0 - 1; y >= 0; y--)
{
    for (int x = 0; x < x0; x++)
    {
        myfile << (x + 0.5) / x0 * length_conv << ";" << (y + 0.5)
            / y0 * length_conv << ";" << velo_conv * vabs[x][y] <<
            ";" << velo_conv * v[x][y][0] << ";" << velo_conv *
            v[x][y][1] << ";" << rho[x][y] << "\n";
    }
}
myfile.close();
}
}
//Doba výpočtu simulace.
duration = (std::clock() - start) / (double)CLOCKS_PER_SEC;
std::cout << "The operation lasted " << duration << " seconds." <<
    '\n';
std::cout << endl;
}

```

A.2 Obtékání překážek

```

/* Lattice Boltzmann sample, written in C++, using the OpenLB
 * library
 *
 * Copyright (C) 2006, 2007, 2012 Jonas Latt, Mathias J. Krause
 * E-mail contact: info@openlb.net
 * The most recent release of OpenLB can be downloaded at
 * <http://www.openlb.net/>
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU General Public License
 * as published by the Free Software Foundation; either version 2
 * of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 */

```

```

* You should have received a copy of the GNU General Public
* License along with this program; if not, write to the Free
* Software Foundation, Inc., 51 Franklin Street, Fifth Floor,
* Boston, MA 02110-1301, USA.
*/

#include "olb2D.h"
#ifndef OLB_PRECOMPILED // Unless precompiled version is used,
#include "olb2D.hh" // include full template code
#endif

#include <cmath>
#include <iostream>
#include <fstream>

using namespace olb;
using namespace olb::descriptors;

typedef double T;
#define DESCRIPTOR D2Q9<>

// Parameters for the simulation setup , geometry, time
const T lx0 = 0.06; //překážková část
const T lxout = 2*lx0;
const T lxmesh = 0.5*lx0; //začátek překážek
const T lxall = lxout+lxmesh; //délka kanálu
const int N = 2000; // resolution of the model
const int M = 50; // resolution of the model - 1/M is the
lattice velocity
const T maxPhysT = 3.; // čas simulace v SI
const T a = 0.0003; //strana čtverečku
const int pocet_rad = 20; //počet dvojřad
const int pocet_new = 2;
const T vert_rozt = lx0/(pocet_rad+1); //vertikální rozteč
const T hor_rozt = lx0/(pocet_rad+1); //horizontální rozteč
const T ly0_new = 3*vert_rozt; //šířka kanálu

// Stores geometry information in form of material numbers
SuperGeometry2D<T> prepareGeometry (UnitConverter<T, DESCRIPTOR>
const& converter)

```

```

{
    OstreamManager clout(std::cout, "prepareGeometry");
    clout << "Prepare Geometry ..." << std::endl;

    // set number of cuboids/blocks
#ifdef PARALLEL_MODE_MPI
    const int noOfCuboids = singleton::mpi().getSize();
#else
    const int noOfCuboids = 6;
#endif

    // setup channel
    Vector<T, 2> extendChannel(lxall, ly0_new);
    Vector<T, 2> originChannel(0, 0);
    std::shared_ptr<IndicatorF2D<T>> channel =
        std::make_shared<IndicatorCuboid2D<T>>(extendChannel,
        originChannel);

    CuboidGeometry2D<T>* cuboidGeometry = new
        CuboidGeometry2D<T>(*(channel),
        converter.getConversionFactorLength(), noOfCuboids);
    HeuristicLoadBalancer<T>* loadBalancer = new
        HeuristicLoadBalancer<T>(*cuboidGeometry);
    // Instantiation of a superGeometry
    SuperGeometry2D<T> superGeometry(*cuboidGeometry, *loadBalancer,
        2); // 2 = overlap ( default)
    // 1 fluid, 2 walls, 3 inflow, 4 outflow 5 obstacles
    // material numbers from zero to 2 inside geometry defined by
    // indicator
    superGeometry.rename(0, 2);
    superGeometry.rename(2, 1, 1, 1); // change 2 to 1, if overlap x
        1 and y 1 (boundaries)

//generování překážek
for (int j=1; j<=pocet_rad; j++)
    for (int i = 1; i <= pocet_new; i++)
    {
        Vector<T, 2> extend(a, a);
        Vector<T, 2> pos(lxmesh+j*hor_rozt, i * vert_rozt-a/2.);
        IndicatorCuboid2D<T> cubed1(extend, pos);
        superGeometry.rename(1, 5, cubed1);
        Vector<T, 2> pos2(lxmesh +(j+0.5) * hor_rozt, (i - 0.5) *
            vert_rozt-a/2.);
    }
}

```

```

IndicatorCuboid2D<T> cubed2(extend, pos2);
superGeometry.rename(1, 5, cubed2);
if (i == pocet_new)
{
    Vector<T, 2> pos2(lxmesh+(j+0.5) * hor_rozt, (i + 0.5) *
        vert_rozt-a/2.);
    IndicatorCuboid2D<T> cubed2(extend, pos2);
    superGeometry.rename(1, 5, cubed2);
}
}

Vector<T,2> extendBC( 0,ly0_new );//nahrazeno za ly0
Vector<T,2> originBC;
IndicatorCuboid2D<T> inflow( extendBC, originBC );

// Set material number for inflow
superGeometry.rename( 2,3,1,inflow );//replaces 2 to 3 if
neighbourhood in direction inflow is 1
originBC[0] = lxmesh+lxout;
IndicatorCuboid2D<T> outflow( extendBC, originBC );
// Set material number for outflow
superGeometry.rename( 2,4,1,outflow );
// Removes all not needed boundary voxels outside the surface
superGeometry.clean();
// Removes all not needed boundary voxels inside the surface
superGeometry.innerClean();
superGeometry.checkForErrors();
superGeometry.getStatistics().print();
clout << "Prepare Geometry ... OK" << std::endl;
return superGeometry;
}

// Set up the geometry of the simulation
void prepareLattice( UnitConverter<T,DESCRIPTOR> const& converter,
    SuperLattice2D<T,DESCRIPTOR>& sLattice,
    Dynamics<T,DESCRIPTOR>& bulkDynamics,
    sOnLatticeBoundaryCondition2D<T,DESCRIPTOR>& bc,
    SuperGeometry2D<T>& superGeometry ) {
OstreamManager clout( std::cout, "prepareLattice" );
clout << "Prepare Lattice ..." << std::endl;

auto bulkIndicator = superGeometry.getMaterialIndicator({1, 3,
    4});

// Material=0 -->do nothing

```

```

sLattice.defineDynamics( superGeometry , 0,
    &instances::getNoDynamics<T,DESCRIPTOR>() );
// Material=1 -->bulk dynamics
// Material=3 -->bulk dynamics (inflow)
// Material=4 -->bulk dynamics (outflow)
sLattice.defineDynamics( bulkIndicator , &bulkDynamics );
// Material=2 -->bounce back
sLattice.defineDynamics( superGeometry , 2,
    &instances::getBounceBack<T,DESCRIPTOR>() );

sLattice.defineDynamics( superGeometry , 5,
    &instances::getBounceBack<T, DESCRIPTOR>());

// Setting of the boundary conditions
bc.addVelocityBoundary( superGeometry , 3,
    converter.getLatticeRelaxationFrequency() );
bc.addPressureBoundary( superGeometry , 4,
    converter.getLatticeRelaxationFrequency() );

// Initial conditions
AnalyticalConst2D<T,T> ux( 0. );
AnalyticalConst2D<T,T> uy( 0. );
AnalyticalConst2D<T,T> rho( 1. );
AnalyticalComposed2D<T,T> u( ux,uy );

//Initialize all values of distribution functions to their local
    equilibrium
sLattice.defineRhoU( bulkIndicator , rho, u );
sLattice.iniEquilibrium( bulkIndicator , rho, u );

// Make the lattice ready for simulation
sLattice.initialize();

clout << "Prepare Lattice ... OK" << std::endl;
}

// Generates a slowly increasing inflow for the first iTMaxStart
    timesteps
void setBoundaryValues( UnitConverter<T,DESCRIPTOR> const&
    converter ,
                                SuperLattice2D<T,DESCRIPTOR>& sLattice , int
                                iT ,
                                SuperGeometry2D<T>& superGeometry ) {

```

```

OstreamManager clout( std::cout, "setBoundaryValues" );

// time for smooth start-up
int iTmaxStart = converter.getLatticeTime( maxPhysT*0.03333333 );
int iTupdate = 5;

if ( iT%iTupdate == 0 && iT<= iTmaxStart ) {
    // Smooth start curve, sinus
    //SinusStartScale<T,int> StartScale( iTmaxStart, T(1) );
    // Smooth start curve, polynomial
    PolynomialStartScale<T,int> StartScale( iTmaxStart, T( 1 ) );
    // Creates and sets the Poiseuille inflow profile using
    // functors HOW THIS WORKS????
    int iTvec[1] = {iT};
    T frac[1] = {};
    StartScale( frac, iTvec );
    T maxVelocity =
        converter.getCharLatticeVelocity() * 0.3333 * frac[0];
    //clout <<frac[0] <<std::endl;
    T distance2Wall = converter.getConversionFactorLength() / 2.;
    Poiseuille2D<T> poiseuilleU( superGeometry, 3, maxVelocity,
        distance2Wall );
    // define lattice speed on inflow
    sLattice.defineU( superGeometry, 3, poiseuilleU ); //zadává na
        supergeometry u materiálu 3 poiseuilleův profil
}
}

// write data to terminal and file system
void getResults( SuperLattice2D<T,DESCRIPTOR>& sLattice,
    UnitConverter<T,DESCRIPTOR> const& converter, int
    iT,
    SuperGeometry2D<T>& superGeometry, Timer<T>& timer,
    SuperPlaneIntegralFluxVelocity2D<T>& velocityFlux,
    SuperPlaneIntegralFluxPressure2D<T>& pressureFlux )
    {
OstreamManager clout( std::cout, "getResults" );
SuperVTMwriter2D<T> vtmWriter( "wires" );

if ( iT==0 ) {
    // Writes geometry, cuboid no. and rank no. to file system
    SuperLatticeGeometry2D<T,DESCRIPTOR> geometry( sLattice,
        superGeometry );
    SuperLatticeCuboid2D<T,DESCRIPTOR> cuboid( sLattice );
    SuperLatticeRank2D<T,DESCRIPTOR> rank( sLattice );
}
}

```

```

    vtmWriter.write( geometry );
    vtmWriter.write( cuboid );
    vtmWriter.write( rank );
    vtmWriter.createMasterFile();
}

// Writes every 0.05 seconds
if ( iT%converter.getLatticeTime( 0.05 )==0 ) {
    SuperLatticePhysVelocity2D<T,DESCRIPTOR> velocity( sLattice ,
        converter );
    SuperLatticePhysPressure2D<T,DESCRIPTOR> pressure( sLattice ,
        converter );
    SuperLatticePhysDissipation2D<T, DESCRIPTOR>
        disipation( sLattice , converter );
    vtmWriter.addFunctor( velocity );
    vtmWriter.addFunctor( pressure );
    // write vtk to file system
    vtmWriter.write( iT );
    sLattice.communicate();
    SuperEuklidNorm2D<T,DESCRIPTOR> normVel( velocity );
    BlockReduction2D2D<T> planeReduction( normVel, 600,
        BlockDataSyncMode::ReduceOnly );

    // write output as JPEG
    heatmap::plotParam<T> jpeg_param;
    heatmap::write( planeReduction , iT , jpeg_param );
}

// Writes every 0.1 simulated
if ( iT%converter.getLatticeTime( 0.01 )==0 ) {
    // velocityFlux.print();
    // pressureFlux.print();

    // write to terminal
    timer.update( iT );
    timer.printStep();
    // Lattice statistics console output
    sLattice.getStatistics().print( iT , converter.getPhysTime( iT ) );
}
}

int main( int argc , char* argv [] ) {
    // === 1st Step: Initialization ===
    olbInit( &argc , &argv );
}

```



```

singleton::directories().setOutputDir( "./tmp/" ); // set output
    directory
OstreamManager clout( std::cout, "main" );

UnitConverter<T,DESCRIPTOR> converter(
    (T) 1x0/N, // physDeltaX: spacing between two lattice
        cells in __m__
    (T) 0.00000075, // physDeltaT: time step in __s__
    (T) a, // charPhysLength: reference length of
        simulation geometry
    (T) 2., // charPhysVelocity: maximal/highest expected
        velocity during simulation in __m / s__
    (T) 0.000015, // physViscosity: physical kinematic viscosity
        in __m^2 / s__
    (T) 1., // physDensity: physical density in __kg /
        m^3__
    (T) 101325 //charPressure in __Pa__
);

// Prints the converter log as console output
converter.print();
// Writes the converter log in a file
converter.write("wires");

// === 2nd Step: Prepare Geometry ===
// Instantiation of a superGeometry
SuperGeometry2D<T> superGeometry( prepareGeometry( converter ) );

// === 3rd Step: Prepare Lattice ===
SuperLattice2D<T,DESCRIPTOR> sLattice( superGeometry );//produce
    from supergeometry
//Dynamics
BGKdynamics<T,DESCRIPTOR> bulkDynamics (
    converter.getLatticeRelaxationFrequency(),
    instances::getBulkMomenta<T,DESCRIPTOR>()
);

// choose between local and non-local boundary condition
sOnLatticeBoundaryCondition2D<T,DESCRIPTOR> sBoundaryCondition(
    sLattice );
createLocalBoundaryCondition2D<T,DESCRIPTOR> ( sBoundaryCondition
    );

```

```

prepareLattice( converter , sLattice , bulkDynamics ,
               sBoundaryCondition , superGeometry );

// instantiate reusable functors
SuperPlaneIntegralFluxVelocity2D<T> velocityFlux( sLattice ,
           converter , superGeometry , {19.,1.} , {0.,1.} );
SuperPlaneIntegralFluxPressure2D<T> pressureFlux( sLattice ,
           converter , superGeometry , {19.,1.} , {0.,1.} );

// === 4th Step: Main Loop with Timer ===
clout << "starting simulation..." << std::endl;
Timer<T> timer( converter.getLatticeTime( maxPhysT ) ,
               superGeometry.getStatistics().getNvoxel() );
timer.start();

for ( int iT = 0; iT < converter.getLatticeTime( maxPhysT ); ++iT
    ) {
    // === 5th Step: Definition of Initial and Boundary Conditions
    ===
    setBoundaryValues( converter , sLattice , iT , superGeometry );
    // === 6th Step: Collide and Stream Execution ===
    sLattice.collideAndStream();
    //clout << "Step " << iT << " done!" << std::endl;
    // === 7th Step: Computation and Output of the Results ===
    getResults( sLattice , converter , iT , superGeometry , timer ,
                velocityFlux , pressureFlux );
}

timer.stop();
timer.printSummary();
}

```