

**Универзитет у Крагујевцу**  
**Факултет инжењерских наука**



**Назив студијског програма: Машинско инжењерство**

**Ниво студија: Мастер академске студије**

**Модул: Информатика у инжењерству**

**Предмет: Рачунска динамика флуида**

**Број индекса: 308/2010**

**Тијана Р. Ђукић**

**Моделирање солид-флуид интеракције  
применом LB методе**

**мастер рад**

**Комисија за преглед и одбрану:**

**1. проф. Ненад Филиповић – ментор**

**2. \_\_\_\_\_**

**3. \_\_\_\_\_**

**Датум одбране: \_\_\_\_\_**

**Оцена: \_\_\_\_\_**

У оквиру овог мастер рада кандидат треба да изведе основне једначине на којима се базира lattice Boltzmann метода – дискретна метода која се примењује у рачунској динамици флуида. Такође, на основу развијених једначина, кандидат треба да развије софтвер на бази ове методе, који ће моћи нумерички да симулира струјања флуида. У оквиру софтвера треба имплементирати и могућност спрегнутог кретања солида и флуида, моделирањем солид-флуид интеракције. Након анализе постојећих метода које се примењују у симулацијама кретања честица кроз крвне судове, треба упоредити резултате развијеног софтвера са резултатима који су приказани у литератури и проверити тачност рада програма.

#### Препоручена литература:

- [1] Filipović, N., Kojić, M., Decuzzi, P., Ferrari, M., "Dissipative Particle Dynamics simulation of circular and elliptical particles motion in 2D laminar shear flow," *Microfluidics and Nanofluidics*, vol. 10, no. 5, pp. 1127-1134, 2010.
- [2] Malaspinas, O. P., *Lattice Boltzmann Method for the Simulation of Viscoelastic Fluid Flows*, Switzerland, PhD dissertation, 2009.
- [3] Wu, J., Shu, C., "Particulate flow simulation via a boundary condition-enforced immersed boundary-lattice Boltzmann scheme," *Commun. Comput. Phys.*, vol. 7, no. 4, pp. 793-812, 2010.

Крагујевац, 01.10.2011.

ментор:

др Ненад Филиповић, ред. проф.

## Резиме рада

Поред стандардних метода рачунске динамике флуида, постоје и друге, дискретне методе, као што је *lattice Boltzmann* (скраћено LB) метода. Ова метода посматра флуид као скуп фиктивних честица и проучавањем динамике њиховог кретања, моделира струјање флуида на макроскопском нивоу. Комбиновањем LB методе и IBM (енг. *Immersed Boundary Method*) приступа добијена је методологија за симулирање солид-флуид интеракције и развијен је посебан софтвер у коме је имплементиран цео концепт. У овом раду приказани су резултати симулација неколико проблема из области биомедицине добијени применом овог софтвера. Добијени резултати су упоређени са резултатима из литературе и добијено је добро поклапање решења, што потврђује да се развијени LB солвер може успешно користити за симулације струјања флуида, са комплексним граничним доменом и уз спрегнуто кретање честица кроз домен флуида. Такође, у овом раду су принципи GPU програмирања примењени на LB солвер. Програм је паралелизован и на специјализованом Tesla уређају постигнуто је убрзање од чак 20 пута у односу на класичну верзију. Применом паралелизованог LB солвера, све симулације кретања солидних честица у домену флуида се могу пратити интерактивно.

**Кључне речи:** симулације струјања флуида, солид-флуид интеракција, софтвер на бази LB методе, GPU програмирање, паралелизација

## Abstract

In addition to conventional methods of computational fluid dynamics, there are also discrete methods, such as lattice Boltzmann (abbreviated LB) method. This method observes fluid as a set of fictitious particles and by studying the dynamics of their movements, the fluid flow is modeled on the macroscopic level. By combining LB method and IBM (Immersed Boundary Method) approach a methodology for simulating solid-fluid interaction is obtained. A special software is developed to implement this concept. This Thesis presents results of simulations of several problems in the field of biomedicine that were obtained using the mentioned software. The obtained numerical results are in very good agreement with the data found in literature. This confirms that the developed LB solver can be successfully used for fluid flow simulations with complex boundaries and with coupled solid particle movements through the domain. Also, in this Thesis the principles of GPU computing are applied on LB solver. The software is parallelized and on a specialized Tesla device it achieved the acceleration of 20 times compared to the classic version. By applying the parallelized LB solver, all simulations of solid particle movements in fluid domain can be observed interactively.

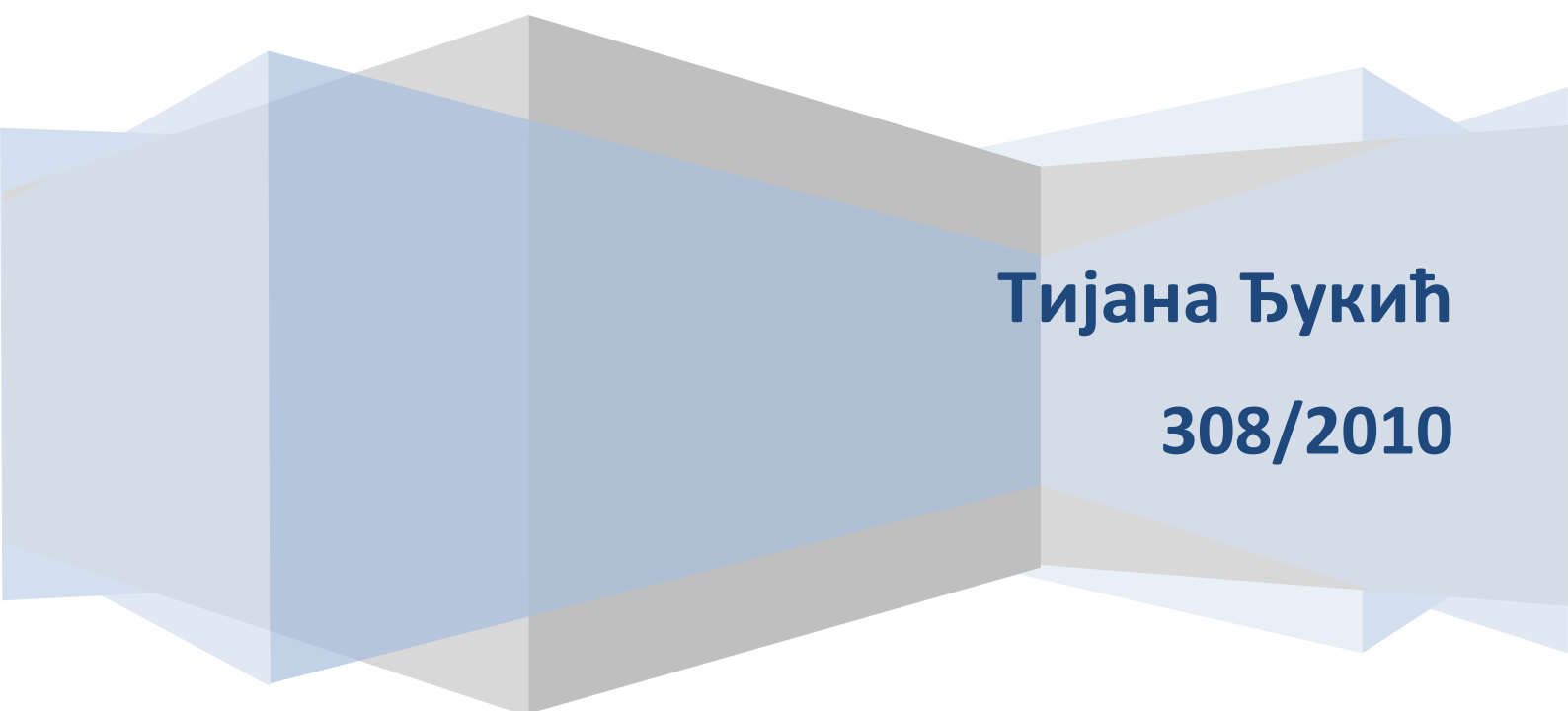
**Key words:** fluid flow simulation, solid-fluid interaction, software based on LB method, GPU computing, parallelization

Факултет инжењерских наука, Крагујевац

# Мастер рад

Моделирање солид-флуид интеракције  
применом LB методе

ментор: др Ненад Филиповић,  
редовни професор



Тијана Ђукић

308/2010

|                                                             |           |
|-------------------------------------------------------------|-----------|
| Садржај.....                                                | I         |
| Листа слика .....                                           | III       |
| Листа табела .....                                          | VI        |
| Списак коришћених ознака .....                              | VII       |
| Коришћене скраћенице .....                                  | X         |
| <b>1 Увод.....</b>                                          | <b>1</b>  |
| <b>2 Извођење континуалне BGK једначине.....</b>            | <b>5</b>  |
| 2.1 Boltzmann-ова једначина .....                           | 6         |
| 2.2 Дефинисање макроскопских величина .....                 | 8         |
| 2.3 Дефинисање закона одржања .....                         | 10        |
| 2.4 Пројектовање на Hermite-ову базу .....                  | 11        |
| 2.5 Chapman-Enskog растављање .....                         | 15        |
| <b>3 Дискретизација BGK једначине .....</b>                 | <b>21</b> |
| 3.1 Дискретизација поља брзина .....                        | 21        |
| 3.2 Временска и просторна дискретизација .....              | 26        |
| <b>4 Детаљи имплементације LB методе.....</b>               | <b>31</b> |
| 4.1 Алгоритам програма .....                                | 32        |
| 4.2 Гранични услови у LB методи .....                       | 35        |
| 4.2.1 Периодични гранични услов .....                       | 35        |
| 4.2.2 Одбојни гранични услов .....                          | 36        |
| 4.2.3 Задавање брзине и притиска као граничних услова ..... | 37        |
| <b>5 Моделирање солид-флуид интеракције.....</b>            | <b>42</b> |

|                                                                               |           |
|-------------------------------------------------------------------------------|-----------|
| 5.1 Слабо спрезање .....                                                      | 43        |
| 5.2 Јако спрезање .....                                                       | 47        |
| <b>6 Симулације кретања честица - Примери.....</b>                            | <b>54</b> |
| 6.1 Правoliniјско стационарно струјање флуида између два паралелна зида ..... | 56        |
| 6.2 Опструјавање цилиндра .....                                               | 58        |
| 6.3 Симулација ротације елиптичне честице .....                               | 60        |
| 6.4 Симулација кретања елиптичне честице .....                                | 63        |
| 6.5 Кретање сфере у флуиду са смичућим профилем брзине .....                  | 65        |
| 6.6 Таложење честице у вискозном флуиду .....                                 | 68        |
| 6.7 Кретање сфере кроз артерију са стенозом .....                             | 72        |
| 6.8 Кретање две сфере кроз артерију са стенозом .....                         | 75        |
| 6.9 Кретање сфере кроз бифуркацију .....                                      | 77        |
| <b>7 Паралелизација LB солвера .....</b>                                      | <b>80</b> |
| 7.1 Програмирање GPU уређаја .....                                            | 81        |
| 7.2 NVIDIA Tesla .....                                                        | 82        |
| 7.3 Паралелизација софтвера на бази lattice Boltzmann методе .....            | 82        |
| 7.3.1 Позивање и извршавање кернел функција .....                             | 84        |
| 7.3.1 Манипулација меморијским ресурсима .....                                | 86        |
| 7.4 Поређење брзине рада стандардне и паралелизоване верзије LB солвера ..... | 89        |
| <b>8 Закључак .....</b>                                                       | <b>93</b> |
| <b>Додатак А.....</b>                                                         | <b>96</b> |
| <b>Литература.....</b>                                                        | <b>98</b> |

## Листа слика

---

|                                                                                                                 |    |
|-----------------------------------------------------------------------------------------------------------------|----|
| Слика 4.1 - Дводимензионална мрежа и дискретизована функција расподеле [41].....                                | 31 |
| Слика 4.2 - Алгоритам програма базираног на LB методи .....                                                     | 33 |
| Слика 4.3 - Функција расподеле пре почетка временског корака.....                                               | 34 |
| Слика 4.4 - Функција расподеле после моделирања судара (првог корака) .....                                     | 34 |
| Слика 4.5 - Функција расподеле после пропагације (другог корака) .....                                          | 34 |
| Слика 4.6 - Периодични гранични услов.....                                                                      | 35 |
| Слика 4.7 - Илустрација bounce-back граничног услова .....                                                      | 36 |
| Слика 4.8 - Информације о функцији расподеле у граничном чвору .....                                            | 37 |
| Слика 5.1 - Размена информација при решавању проблема солид-флуид интеракције<br>применом слабог спрезања ..... | 43 |
| Слика 5.2 - Слабо спрезање солида и флуида – LB метода .....                                                    | 44 |
| Слика 5.3 - Алгоритам слабог спрезања (LB метода) .....                                                         | 46 |
| Слика 5.4 – Међусобни утицај тачака домена флуида и солида .....                                                | 48 |
| Слика 5.5 - Алгоритам јаког спрезања (LB и <i>Immersed boundary</i> метода).....                                | 53 |
| Слика 6.1 - Дводимензионални домен .....                                                                        | 54 |
| Слика 6.2 - Пример 1 – Шема геометрије струјања .....                                                           | 56 |
| Слика 6.3 - Пример 1 - Поље брзине .....                                                                        | 57 |
| Слика 6.4 - Пример 1 - Поље притиска .....                                                                      | 57 |
| Слика 6.5 - Пример 1 - Упоредни приказ брзине .....                                                             | 58 |
| Слика 6.6 - Пример 2 - Шема геометрије струјања .....                                                           | 58 |
| Слика 6.7 - Пример 2 - Поље брзине .....                                                                        | 59 |
| Слика 6.8 - Пример 2 - Упоредни приказ вучне силе .....                                                         | 60 |
| Слика 6.9 - Пример 3 - Шема геометрије струјања .....                                                           | 61 |

|                                                                                               |    |
|-----------------------------------------------------------------------------------------------|----|
| Слика 6.10 - Пример 3 - Поље брзине .....                                                     | 61 |
| Слика 6.11 - Пример 3 - Упоредни приказ промене угла ротације .....                           | 62 |
| Слика 6.12 - Пример 3 - Упоредни приказ промене угаоне брзине .....                           | 63 |
| Слика 6.13 - Пример 4 - Шема геометрије струјања .....                                        | 63 |
| Слика 6.14 - Пример 4 - Путања елиптичне честице за $St=1.25$ .....                           | 64 |
| Слика 6.15 - Пример 4 - Путања елиптичне честице за различите вредности Стоксовог броја ..... | 65 |
| Слика 6.16 - Пример 5 - Шема геометрије струјања .....                                        | 66 |
| Слика 6.17 - Пример 5 - Поље брзине .....                                                     | 66 |
| Слика 6.18 - Пример 5 - Поређење решења - промена позиције у правцу осе $y$ .....             | 67 |
| Слика 6.19 - Пример 5 - Поређење решења – Промена $x$ компоненте брзине .....                 | 67 |
| Слика 6.20 - Пример 5 - Поређење решења - Промена $y$ компоненте брзине.....                  | 68 |
| Слика 6.21 - Пример 6 - Шема геометрије струјања .....                                        | 69 |
| Слика 6.22 - Пример 6 - Поље брзине .....                                                     | 70 |
| Слика 6.23 - Пример 6 - Поређење решења - промена позиције у правцу осе $y$ .....             | 71 |
| Слика 6.24 - Пример 6 - Поређење решења - Промена $y$ компоненте брзине.....                  | 71 |
| Слика 6.25 - Пример 6 - Поређење решења - Промена Рејнолдсовог броја.....                     | 72 |
| Слика 6.26 - Пример 6 - Поређење решења - Промена кинетичке енергије .....                    | 72 |
| Слика 6.27 - Пример 7 - Шема геометрије струјања .....                                        | 73 |
| Слика 6.28 - Пример 7 - Поље брзине .....                                                     | 74 |
| Слика 6.29 - Пример 7 - Путања честице кроз артерију са стенозом .....                        | 74 |
| Слика 6.30 - Пример 7 - Поређење решења - Промена $x$ компоненте брзине.....                  | 74 |
| Слика 6.31 - Пример 7 - Поређење решења - Промена $y$ компоненте брзине.....                  | 75 |
| Слика 6.32 - Пример 8.1 - Поље брзине .....                                                   | 76 |
| Слика 6.33 - Пример 8.1 - Путање честица .....                                                | 76 |
| Слика 6.34 - Пример 8.2 - Путање честица .....                                                | 77 |
| Слика 6.35 - Пример 8.2 - Поље брзине .....                                                   | 77 |
| Слика 6.36 - Пример 9 - Шема геометрије струјања .....                                        | 78 |
| Слика 6.37 - Пример 9 - Поље брзине .....                                                     | 79 |
| Слика 6.38 - Пример 9 - Путања честице.....                                                   | 79 |

|                                                                                                        |    |
|--------------------------------------------------------------------------------------------------------|----|
| Слика 7.1 - Алгоритам рада класичног LB солвера.....                                                   | 83 |
| Слика 7.2 - Симулација судара и пропагација - класична и паралелна верзија.....                        | 84 |
| Слика 7.3 - Илустрација мреже блокова и нити .....                                                     | 85 |
| Слика 7.4 - Алгоритам извршавања паралелизованог LB солвера .....                                      | 88 |
| Слика 7.5 - Упоредни приказ времена извршавања различитих верзија LB солвера .....                     | 89 |
| Слика 7.6 - Упоредни приказ времена извршавања за различит број чворова.....                           | 91 |
| Слика 7.7 - Дијаграм зависности убрзања паралелизованог LB солвера у зависности од броја чворова ..... | 92 |

## Листа табела

---

|                                                    |    |
|----------------------------------------------------|----|
| Табела 6.1. Подаци за LB солвер за пример 1 .....  | 57 |
| Табела 6.2. Подаци за LB солвер за пример 2 .....  | 59 |
| Табела 6.3. Подаци за LB солвер за пример 3 .....  | 61 |
| Табела 6.4. Подаци за LB солвер за пример 5 .....  | 66 |
| Табела 6.5. Подаци за LB солвер за пример 6 .....  | 69 |
| Табела 6.6. Подаци за LB солвер за пример 7 .....  | 73 |
| Табела 6.7. Подаци за LB солвер за пример 9 .....  | 78 |
| Табела 7.1. Карактеристике NVIDIA Tesla C1060..... | 82 |

## Списак коришћених ознака

---

|                                        |   |                                             |
|----------------------------------------|---|---------------------------------------------|
| $a_{\alpha}^{(n)}$                     | - | коефицијент експанзије степена $n$          |
| $\alpha$                               | - | векторски индекс                            |
| $c$                                    | - | микроскопска брзина                         |
| $c_i$                                  | - | дискретизована микроскопска брзина          |
| $c_l$                                  | - | брзина светлости у јединицама мреже         |
| $c_s$                                  | - | брзина звука у јединицама мреже             |
| $\zeta$                                | - | растојање између два чвора мреже            |
| $\delta_{\alpha\beta}$                 | - | <i>Kronecker</i> -ов делта симбол           |
| $\delta(\mathbf{r})$                   | - | <i>Dirac</i> -ова делта функција            |
| $\delta \mathbf{u}$                    | - | корекција брзине флуида                     |
| $\delta \mathbf{u}_B(\mathbf{X}_B, t)$ | - | корекција брзине солида                     |
| $\Delta s_l$                           | - | дужина лука граничног домена                |
| $\epsilon$                             | - | унутрашња енергија                          |
| $\mathbf{F}(s, t)$                     | - | сила у граничним тачкама                    |
| $F(v)$                                 | - | израз за силу                               |
| $\mathbf{F}^{col}$                     | - | сила интеракције између две честице         |
| $F_i$                                  | - | дискретизован израз за силу                 |
| $\mathbf{F}_R$                         | - | сила којом флуид делује на солид            |
| $f$                                    | - | функција вероватноће расподеле маса честица |
| $f^{(0)}$                              | - | равнотежна функција расподеле               |
| $f^{(1)}$                              | - | ванравнотежна функција расподеле            |
| $f^N$                                  | - | скраћена функција расподеле                 |

|                               |   |                                                       |
|-------------------------------|---|-------------------------------------------------------|
| $f_i$                         | - | дискретизована функција расподеле                     |
| $f_i^{(0)}$                   | - | дискретизована равнотежна функција расподеле          |
| $f_i^{(1)}$                   | - | дискретизована ванравнотежна функција расподеле       |
| $\overline{f_i}$              | - | дискретизована функција расподеле у нумеричкој шеми   |
| $\mathbf{G}$                  | - | убрзање Земљине теже                                  |
| $\mathbf{g}$                  | - | вектор спољашњих сила                                 |
| $\mathcal{H}_\alpha^{(n)}$    | - | <i>Hermite</i> -ов полином степена $n$                |
| $\mathcal{H}_{i\alpha}^{(n)}$ | - | дискретизовани <i>Hermite</i> -ов полином степена $n$ |
| $h$                           | - | растојање између 2 тачке фиксне мреже                 |
| $\mathbf{I}$                  | - | јединична матрица                                     |
| $I_0$                         | - | момент инерције честице                               |
| $j$                           | - | количина кретања                                      |
| $\mathbf{K}$                  | - | вектор количине кретања                               |
| $k_B$                         | - | <i>Boltzmann</i> -ова константа                       |
| $L$                           | - | карактеристична дужина посматраног домена             |
| $l$                           | - | средњи слободни пут                                   |
| $l_x, l_y$                    | - | карактеристичне дужине домена                         |
| $\mathbf{M}_R$                | - | момент силе којом флуид делује на солид               |
| $m$                           | - | маса честице                                          |
| $\mu_s$                       | - | динамичка вискозност                                  |
| $\nabla$                      | - | оператор дивергенције                                 |
| $N$                           | - | резолуција LB мреже                                   |
| $N_L$                         | - | број граничних ( <i>Lagrange</i> -ових) тачака        |
| $\rho$                        | - | густина                                               |
| $\rho_f$                      | - | густина флуида                                        |
| $\rho_p$                      | - | густина честице                                       |
| $\Omega$                      | - | оператор судара                                       |

|                      |   |                                                                     |
|----------------------|---|---------------------------------------------------------------------|
| $\omega$             | - | угаона брзина                                                       |
| $\omega(\mathbf{v})$ | - | тежинска функција                                                   |
| $\omega_i$           | - | дискретни тежински коефицијент <i>Gauss-Hermite</i> -ове квадратуре |
| $\mathbf{P}$         | - | тензор напона                                                       |
| $p$                  | - | притисак                                                            |
| $\mathbf{q}$         | - | флукс унутрашње енергије                                            |
| $\mathbf{S}$         | - | тензор деформације                                                  |
| $T$                  | - | температура у Келвинима                                             |
| $t$                  | - | време                                                               |
| $\Delta t$           | - | временски корак                                                     |
| $\tau$               | - | време релаксације                                                   |
| $\bar{\tau}$         | - | модификовано време релаксације                                      |
| $\theta$             | - | величина која је једнака $k_B T / m$                                |
| $\mathbf{u}$         | - | поље средње брзине                                                  |
| $\mathbf{u}^*$       | - | привремена брзина флуида                                            |
| $u_{\perp}$          | - | пројекција вектора брзине на вектор нормале $\mathbf{n}$            |
| $u_{lb}$             | - | карактеристична брзина струјања                                     |
| $\mathbf{v}$         | - | вектор брзине                                                       |
| $\mathbf{v}_C$       | - | брзина центра честице                                               |
| $\mathbf{v}_i$       | - | апсцисе <i>Gauss-Hermite</i> -ове квадратуре                        |
| $\mathbf{X}_B(s, t)$ | - | координате граничних ( <i>Lagrange</i> -ових) тачака                |
| $\mathbf{X}_R$       | - | координате центра масе честице                                      |
| $\mathbf{x}$         | - | вектор положаја                                                     |

*Коришћене скраћенице*

|             |   |                                                                                                    |
|-------------|---|----------------------------------------------------------------------------------------------------|
| <i>BGK</i>  | - | <i>Bhatnagar-Gross-Krook</i> модел                                                                 |
| <i>CUDA</i> | - | <i>Compute Unified Device Architecture</i>                                                         |
| <i>GBps</i> | - | <i>Gigabytes per second</i>                                                                        |
| <i>GPU</i>  | - | <i>Graphics Processing Unit</i>                                                                    |
| <i>Kn</i>   | - | <i>Knudsen</i> -ов број                                                                            |
| <i>LB</i>   | - | <i>lattice Boltzmann</i> метода                                                                    |
| <i>Ma</i>   | - | <i>Mach</i> -ов број                                                                               |
| <i>MB</i>   | - | <i>Megabytes</i>                                                                                   |
| <i>Re</i>   | - | <i>Reynolds</i> -ов број                                                                           |
| <i>SRT</i>  | - | апроксимација једноструког времена релаксације (енг. <i>single relaxation time approximation</i> ) |
| <i>St</i>   | - | <i>Stokes</i> -ов број                                                                             |

# 1 УВОД

---

Проблем симулације струјања флуида се може нумерички решити применом широког спектра метода – методом коначних елемената, коначних разлика, коначних запремина. Суштина свих метода је решавање Навије-Стоксових једначина и добијање поља промене брзине и притиска. Поред ових стандардних метода рачунске динамике флуида, постоје и друге, дискретне методе, као што је *lattice Boltzmann* (скраћено LB) метода. Ова метода посматра флуид као скуп фиктивних честица и проучавањем динамике кретања ових честица, моделира се струјање флуида на макроскопском нивоу. Кретање се моделира у два одвојена корака – најпре се моделирају судари честица, а потом и њихова даља пропагација у дефинисаном домену. Највеће предности ове методе су једноставна имплементација и могућност паралелизације. Такође, велика предност је што нема потребе за решавањем диференцијалних једначина или система једначина.

LB метода се успешно примењује за симулације струјања вишекомпонентног флуида у порозним срединама и кроз микро и нано канале [1,2], али и у моделирању крви, укључујући симулације кретања ћелија канцера [3], кретање леукоцита у крвним судовима [4], моделирање интеракције леукоцита и еритроцита [5,6,7], као и у многим другим областима биомедицине.

Последњих година моделирање интеракције солида и флуида је једна веома интересантна област, која је тема многих научних радова. Исправно симулирање струјања флуида, са солидним честицама које се крећу кроз домен флуида, представља данас један од највећих изазова у области рачунске динамике флуида. Постоји неколико приступа у моделирању кретања солида кроз флуид. Један је коришћење координатне трансформације [8], други је коришћење покретне мреже коначних елемената [9], трећи је примена тзв. ALE формулације (енг. *Arbitrary Lagrangian Eulerian*) [10]. Сваки од ових приступа има своје предности и недостатке. У литератури се користи још један приступ моделирању солид-флуид интеракције који је примењен у овом раду – IBM (енг. *Immersed Boundary Method*).

Комбиновање LB методе и IBM приступа се показало као ефикасан начин за симулирање солид-флуид интеракције. IB методу је први развио *Peskin* [11] и користио је за симулације струјања крви кроз срце. Ова метода користи фиксану мрежу за домен флуида и скуп *Lagrange*-ових тачака за моделирање границе домена солида. Основна идеја је да се граница солида третира као деформабилна, са великом крутошћу. Свака деформација границе доводи до појаве силе која тежи да врати границу у почетну позицију. Ова сила се преноси на чворове фиксане мреже и на тај начин солид делује на флуид и обрнуто. Решавањем система Навије-Стоксових једначина са спољашњом силом за домен флуида може се симулирати постојање солида уроњеног у флуид. Објављен је велики број радова у којима су применом IBM приступа за моделирање интеракције солида и флуида симулирани различити проблеми у области динамике флуида [12,13,14]. У овом раду су приказани резултати симулација, добијени применом посебно развијеног софтвера на бази LB методе, са додатно имплементираним IB методом за симулирање солид-флуид интеракције. Добијени резултати су упоређени са резултатима из литературе и добијено је добро поклапање решења, што потврђује да се развијени LB солвер може успешно користити за симулације струјања флуида, са комплексним граничним доменом и уз спрегнуто кретање честица кроз домен флуида.

Модерне компјутерске методе прорачуна и симулације различитих појава, као што је симулација струјања флуида могу бити јако комплексне. Рачунарски захтеви оваквих апликација су веома високи и потребни су велики рачунарски ресурси да би се извршиле такве симулације. Технологија данашњих персоналних (CPU) рачунара је напредовала током година, али како поменуте апликације захтевају све више и више рачунарских ресурса, модерни рачунари то не могу да испрате на одговарајући начин. Предвиђа се да ће чак два или три пута већа рачунарска снага бити потребна за будуће апликације [15]. Природно решење су компјутерски кластери (енг. *computer clusters*). Али, ова врста система је веома скупа, захтева додатан простор за складиштење, проветравање, климатизацију и електричну струју за напајање.

Апликације које приказују неке графичке елементе приступају симултано великом скупу података. Овакве апликације обављају исте операције на милионима (или чак и већем броју) тачака (пиксела) истовремено. NVIDIA је светски познат произвођач графичких картица, које омогућавају рад оваквих графичких апликација. Временом су графичке картице компаније NVIDIA постале све програмабилније, што је довело до тога да NVIDIA представи свој први GPU уређај (енг. *Graphics Processing Unit*). У почетку, GPU уређаји су се користили за приказивање, рендеровање и растеризовање слика. Почетком овог века, ови уређаји су почели да се користе у широком спектру апликација опште намене [16], од једноставних израчунавања, линеарне алгебре, решавања система једначина, до знатно сложенијих алгоритама. Многи научници из

различитих области, пре свега из области биомедицинског инжењеринга, електромагнетизма, геодезије и других су почели да користе GPU уређаје за компјутерске симулације, што је довело до драстичног убрзања рада њихових програма.

С обзиром да модерне графичке картице, односно GPU уређаји, немају већину наведених недостатака компјутерских кластера, а имају добре перформансе, GPU уређаји представљају добру алтернативу за апликације високих захтева са аспекта рачунарских ресурса. Програмирање GPU уређаја спада у област рачунарства која се назива *high performance computing*. Због велике популарности овог вида програмирања и широке примене, уведен је и нови појам – програмирање GPU уређаја опште намене (енг. *General Purpose GPU programming - GPGPU*). GPU уређаји извршавају стотине или чак хиљаде нити (енг. *threads*) симултано и самим тим се убрзавају прорачуни и научне симулације. Још једна додатна предност је у томе што GPU уређај ради као копроцесор главног рачунара. На овај начин додатни рачунарски ресурси су доступни персоналном рачунару, а при томе се не користе суперкомпјутери или скупи кластер системи. GPU уређај и персонални рачунар заједно чине ефикасан систем, а при томе такође није потребно уносити велике измене у постојеће компјутерске програме, захваљујући посебној архитектури, о којој ће бити речи у овом раду. Програми који су радили и по неколико дана или сати, уз помоћ GPU уређаја раде и до неколико десетина пута брже. На тај начин симулације многих проблема се могу пратити интерактивно.

Постоји много примера успешне примене GPU уређаја, а већина је публикована у специјализованим часописима. Неке од примена су у области биологије [17], симулација струјања флуида [18], молекуларне динамике [19], хемијских реакција [20], процесирања и анализе медицинских слика [21], моделирања земљотреса применом коначних елемената [22] и другим. У овом раду су принципи GPU програмирања примењени на симулацију струјања флуида, конкретно на имплементацију *lattice Boltzmann* методе. Већ поменути софтвер развијен на бази LB методе је паралелизован и на специјализованом Tesla уређају постигнуто је убрзање од чак 20 пута у односу на класичну верзију софтвера, што знатно олакшава и убрзава симулације струјања флуида. Такође, применом паралелизованог LB солвера, све симулације кретања солидних честица у домену флуида се могу пратити интерактивно.

Рад је организован на следећи начин. У Поглављу 2 дате су теоријске основе *lattice Boltzmann* методе, укључујући и појединости везане за извођење основних једначина, као и доказ да се ове једначине могу свести на познату Навије-Стоксову једначину и једначину континуитета. У Поглављу 3 описан је поступак дискретизације основних једначина и дат је коначни облик једначина које се користе при имплементацији LB

методе. Дефинисање граничних услова у LB методи и други детаљи имплементације су тема Поглавља 4, док су у Поглављу 5 описани приступи моделирању солид-флуид интеракције коришћени у LB солверу. У Поглављу 6 су приказани примери и резултати добијени применом развијеног софтвера. Паралелизација софтвера је описана у Поглављу 7. Поглавље 8 даје кратак закључак целог рада.

## 2 Извођење континуалне BGK једначине

---

Метода која је предмет овог рада, *lattice Boltzmann* метода, припада класи проблема такозваних *Cellular Automata* (CA). То практично значи да се физички систем посматра на идеализован начин, тако да су време и простор дискретизовани, односно цео простор је сачињен од великог броја идентичних ћелија [23]. Дефинише се посебна функција пропагације, која зависи од стања суседних ћелија и идентичног је облика за све ћелије. Стање свих ћелија се ажурира синхронизовано, кроз низ итерација, односно у дискретним временским корацима.

Посебна врста CA, тзв. *lattice-gas automata* (LGA) [24] описује динамику честица које се крећу и сударају у дискретном просторно-временском домену. Применом овог модела се уз одређена ограничења може добити решење Навије-Стоксове једначине, што значи да се може користити за симулирање понашања флуида. Предности ове методе су лака имплементација, стабилност решења, лако задавање граничних услова и природна паралелизација. Међутим, модел има и доста недостатака, као што је статистичка грешка, посебне процедуре за осредњавање да би се добиле макроскопске величине и слично.

Једначине на којима се заснива *lattice Boltzmann* метода се могу извести на основу једначина LGA модела. Циљ развоја нове LB методе је био управо да се отклоне поменути недостаци LGA модела и да се тако побољшани модел може лакше примењивати за моделирање струјања флуида. LGA модел се дефинише мрежом ћелија (енг. *lattice* - решетка), а сваки чвор мреже је дефинисан једном променљивом *Boolean* типа. За разлику од LGA модела, код LB модела је теорија са *Boolean* бројева проширена на реалне бројеве [25], тако да се у сваком чвору мреже дефинише реални број, односно честична функција расподеле. На тај начин је постигнуто да LB метода има већу нумеричку тачност и буде ефикаснија.

Међутим, постоји још један начин извођења једначина LB методе, а то је директно извођење из континуалне *Boltzmann*-ове једначине. Управо тај приступ је описан у наставку овог поглавља. Најпре је дефинисана полазна једначина, а потом су извршене потребне трансформације да би се добио облик погодан за даљу дискретизацију и који се може свести на систем једначина које се користе у рачунској динамици флуида – једначину континуитета и Навије-Стоксову једначину.

## 2.1 Boltzmann-ова једначина

Као што је већ поменуто у уводном делу поглавља, *Lattice Boltzmann* метода не мора бити изведена јединствено из *LGA* методе. Постоји једноставнији поступак извођења, који се иначе чешће користи у пракси и може се срести у литератури [26,27]. Тај поступак укључује трансформацију *Boltzmann*-ове једначине, која важи за континуум.

*Boltzmann*-ова једначина је парцијална диференцијална једначина која описује понашање и кретање честица у простору на основу функције расподеле  $f$ . Ова функција расподеле је дефинисана тако да вредност  $f(\mathbf{x}, \mathbf{v}, t)$  представља вероватноћу да се честица налази унутар елемента  $d\mathbf{x} d\mathbf{v}$  око тачке  $(\mathbf{x}, \mathbf{v})$ , у тренутку  $t$ . При томе се са  $\mathbf{x}$  и  $\mathbf{v}$  означавају вектор положаја и вектор брзине честице, респективно.

Увођење овакве функције је било неопходно јер се у оквиру једног домена посматра велики број честица, које интерагују кроз међусобне сударе. Када би се посматрала свака честица појединачно, то би значило праћење флуида на микроскопском нивоу. Са једне стране, то би било практично немогуће, а са друге стране, за проблеме динамике флуида далеко је битније посматрати промене у систему на макроскопском нивоу. Због тога се у симулацијама LB методом користи поменута функција расподеле, која се другачије назива и функција расподеле маса честица. Може се још дефинисати тако да

$$f(\mathbf{x}, \mathbf{v}, t) d^3\mathbf{x} d^3\mathbf{v} \quad (2.1)$$

представља укупну масу честица које се налазе у запремини  $d^3\mathbf{x}$  око тачке  $\mathbf{x}$  и које имају брзину унутар  $d^3\mathbf{v}$  око брзине  $\mathbf{v}$  у тренутку  $t$ . Оваква дефиниција важи ако мали елемент  $d^3\mathbf{x} d^3\mathbf{v}$  садржи велики број честица, што као што је већ поменуто овде јесте случај.

Ако се узму у обзир спољашње силе, преко вектора спољашњих сила  $\mathbf{g}$ , може се написати једначина баланса функције расподеле - *Boltzmann*-ова једначина:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{x}} + \frac{\mathbf{g}}{m} \cdot \frac{\partial f}{\partial \mathbf{v}} = \Omega \quad (2.2)$$

где је са  $\Omega$  означен интеграл судара или оператор судара (енг. *collision operator*), који представља промену функције расподеле услед судара честица.

*Boltzmann*-ова једначина је нелинеарна интегро-диференцијална једначина, која је тешко решива, ако су у питању велике запремине и велики број честица.

Оператор судара  $\Omega$  представља квадратну функцију од  $f$  (пошто се разматрају само бинарни судари – претпоставка је да се у једном тренутку у неком простору мреже сударају само две честице истовремено). Како је ова функција јако комплексног облика, уводи се поједностављени модел, који су предложили *Bhatnagar*, *Gross* и *Krook* [28]. Уведена је претпоставка да је ефекат судара честица такав да доводи флуид „ближе“ равнотежном стању. Овај модел је познат као апроксимација једноструког времена релаксације (енг. *single relaxation time approximation*) или *Bhatnagar-Gross-Krook* (BGK) модел. Оператор  $\Omega$  је дефинисан на следећи начин:

$$\Omega = -\frac{1}{\tau}(f - f^{(0)}) \quad (2.3)$$

где је  $\tau$  време релаксације (просечно време које протекне између 2 судара), а  $f^{(0)}$  равнотежна функција расподеле, тзв. *Maxwell-Boltzmann*-ова функција расподеле.

*Maxwell-Boltzmann*-ова функција расподеле вероватноће је преузета из статистичке механике, где се користи да би се одредиле брзине молекула. Иначе, овај принцип се односи на гасове, али је већ напоменуто да се LB метода заснива на принципима који важе за гасове и да се увођењем одређених ограничења примењују у динамици флуида. Коришћење једне статистичке методе за одређивање неких величина је неопходно зато што су димензије честица које чине флуид и чије се кретање и интеракције разматрају инфинитезимално мале. Ако би требало пратити сваку честицу појединачно, у великом просторном домену, то би захтевало велики број операција, који чак ни савремени рачунари не би могли да ураде у неко разумно време. Такође, у оваквим случајевима долази до изражаја ХајзеЛВергов принцип неодређености – како се повећава број честица у посматраном просторном домену, није могуће довољно тачно одредити позицију и брзину сваке од честица. А пошто је овакав вид праћења неопходан, уводи се статистичка репрезентација која испуњава наведене услове. Извођење *Maxwell-Boltzmann*-ове функције расподеле је овде изостављено, а може се наћи у литератури [29,30]. Коначан израз равнотежне функције расподеле гласи:

$$f^{(0)}(\mathbf{x}, \mathbf{v}, t) = \frac{\rho(\mathbf{x}, t)}{(2\pi \theta(\mathbf{x}, t))^{D/2}} \exp\left(-\frac{(\mathbf{u}(\mathbf{x}, t) - \mathbf{v})^2}{2 \theta(\mathbf{x}, t)}\right) \quad (2.4)$$

У овом изразу:

$D$  је број физичких димензија ( $D = 3$  за тродимензионални простор)

$\theta = k_B T/m$

$k_B = 1,38 \cdot 10^{-23} \frac{J}{K}$  је *Boltzmann*-ова константа

$T$  је апсолутна температура у Келвинима (K)

$m$  је маса честице (у даљем тексту, сматра се да је  $m = 1$ )

Коначно, BGK модел *Boltzmann*-ове једначине, који важи за континуум, дефинисан је следећом једначином:

$$\frac{\partial f}{\partial t} + \mathbf{v} \frac{\partial f}{\partial \mathbf{x}} + \mathbf{g} \frac{\partial f}{\partial \mathbf{v}} = -\frac{1}{\tau} (f - f^{(0)}) \quad (2.5)$$

## 2.2 Дефинисање макроскопских величина

LB метода описује флуид на молекуларном нивоу, а карактеристике флуида са аспекта континуума су имплицитно садржане у самом моделу. Функција расподеле, као главна карактеристика струјања посматраног на микроскопском нивоу и основна величина која се одређује у LB симулацијама, користи се за одређивање макроскопских величина које дефинишу струјање флуида. Густина и количина кретања се одређују израчунавањем следећих интеграла:

$$\rho(\mathbf{x}, t) = \int d\mathbf{v} f(\mathbf{x}, \mathbf{v}, t) \quad (2.6)$$

$$\mathbf{j}(\mathbf{x}, t) = \rho \mathbf{u} = \int d\mathbf{v} \mathbf{v} f(\mathbf{x}, \mathbf{v}, t) \quad (2.7)$$

Интеграција се врши на целом домену брзине. У горњим изразима,  $\mathbf{u}$  представља поље средње брзине.

Остале макроскопске величине се рачунају као функције вишег степена, па се ради прегледности не пишу у функцији средње брзине, већ преко одступања од средње брзине. Због тога се и уводи појам микроскопске брзине, као поље одступања од средње брзине.

$$\mathbf{c}(\mathbf{x}, \mathbf{v}, t) = \mathbf{v} - \mathbf{u}(\mathbf{x}, t) \quad (2.8)$$

Сада се густина унутрашње енергије (енг. *internal energy density*) може изразити на следећи начин:

$$\rho\epsilon = \frac{1}{2} \int d\mathbf{c} \, \mathbf{c}^2 f(\mathbf{x}, \mathbf{v}, t) \quad (2.9)$$

где је  $\mathbf{c}^2 = \|\mathbf{c}\|^2$

У складу са једначином која важи за једноатомски гас, и у случају флуида који се разматра на овај начин, унутрашња енергија се следећом једначином може повезати са температуром:

$$\epsilon = \frac{D}{2} \theta \quad (2.10)$$

Односно:

$$\rho\epsilon = \frac{D}{2} \rho\theta \quad (2.11)$$

Поред ових величина, потребно је дефинисати још две величине које фигуришу у законима одржања, који ће детаљније бити разматрани у наставку поглавља. Једна од тих величина је тензор напона, који се дефинише на следећи начин:

$$\mathbf{P} = \int d\mathbf{c} \, \mathbf{c} \mathbf{c} f \quad (2.12)$$

где је  $\mathbf{c} \mathbf{c}$  тензорски производ  $\mathbf{c}$  и  $\mathbf{c}$ .

Друга величина је флукс унутрашње енергије  $\mathbf{q}$ :

$$\mathbf{q} = \frac{1}{2} \int d\mathbf{c} \, \mathbf{c} \mathbf{c}^2 f \quad (2.13)$$

На основу тензора напона, могуће је одредити и хидростатички притисак (који је повезан и са унутрашњом енергијом):

$$p = \frac{\mathbf{P}_{ii}}{D} = \frac{2\rho\epsilon}{D} \quad (2.14)$$

Коришћењем израза (2.11), долази се до једначине стања идеалног гаса:

$$p = \rho\theta \quad (2.15)$$

### 2.3 Дефинисање закона одржања

Применом посебне процедуре осредњавања коју је описао *Malaspinas* [27], могуће је из једначине (2.5) добити познате законе одржања. Најпре се може добити закон одржања масе (једначина континуитета):

$$\frac{\partial}{\partial t}(\rho) + \frac{\partial}{\partial x}(\rho \mathbf{u}) = 0 \quad (2.16)$$

Затим се може добити закон одржања количине кретања:

$$\frac{\partial}{\partial t}(\rho \mathbf{u}) + \frac{\partial}{\partial x}(\rho \mathbf{u} \mathbf{u}) - \frac{\partial \mathbf{P}}{\partial x} = \rho \mathbf{g} \quad (2.17)$$

Односно, ако се примени једначина континуитета, закон одржања количине кретања гласи:

$$\rho \frac{d\mathbf{u}}{dt} - \frac{\partial \mathbf{P}}{\partial x} = \rho \mathbf{g} \quad (2.18)$$

где  $\frac{d}{dt}$  представља тотални диференцијал.

Трећи једначина, која дефинише закон одржања енергије, гласи:

$$\frac{\partial}{\partial t} \left( \rho \epsilon + \frac{1}{2} \rho \mathbf{u}^2 \right) + \frac{\partial}{\partial x} \cdot \left[ \mathbf{u} \left( \rho \epsilon + \frac{1}{2} \rho \mathbf{u}^2 \right) + \mathbf{P} \cdot \mathbf{u} + \mathbf{q} \right] = 0 \quad (2.19)$$

Ако се примене закони одржања количине кретања и једначина континуитета, добија се сређени израз закона одржања енергије:

$$\rho \frac{d\epsilon}{dt} + \mathbf{P} \frac{\partial}{\partial x} \mathbf{u} + \frac{\partial}{\partial x} \mathbf{q} = 0 \quad (2.20)$$

На овај начин су добијени закони одржања масе, количине кретања и енергије. Да би овај систем једначина био потпун, неопходно је дефинисати конститутивне релације за тензор напона и флуks унутрашње енергије.

Један од начина да се добију потребне једначине је да се једначина (2.7) пројектује на *Hermite*-ову базу (енг. *Hermite polynomials basis projection*) и да се онда анализирају добијени изрази. Ову процедуру је први предложио и спровео *Grad* [31,32]. У овом раду неће бити описани сви детаљи овог поступка, већ само основне једначине, које су потребне да би се разумео концепт LB методе. Такође, каснија извођења наведена у овом раду се надовезују на неке величине дефинисане у поменутом поступку, тако да је ради бољег разумевања неопходно укратко објаснити овај поступак, а детаљи се могу наћи у литератури [26,33].

## 2.4 Пројектовање на *Hermite*-ову базу

Пројекција функције расподеле брзина на *Hermite*-ову полиномску базу дата је следећом једначином:

$$f(\mathbf{x}, \mathbf{v}, t) = \omega(\mathbf{v}) \sum_{n=0}^{\infty} \frac{1}{n!} \mathcal{H}_{\alpha}^{(n)}(\mathbf{v}) a_{\alpha}^{(n)}(\mathbf{x}, t) \quad (2.21)$$

Напомена: У наставку рада биће коришћена индексна нотација која се користи и у литератури, ради усаглашености и лакшег праћења поступка извођења.

У изразу (2.21):

$\omega(\mathbf{v})$  је тежинска функција

$\mathcal{H}_{\alpha}^{(n)}$  и  $a_{\alpha}^{(n)}$  представљају *Hermite*-ов полином степена  $n$  и коефицијент експанзије степена  $n$ , респективно

$\alpha$  је векторски индекс;  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ , при томе важи Einstein-ова конвенција о сабирању по поновљеним индексима.

*Rodrigues*-ова формула дефинише *Hermite*-ов полином степена  $n$ , у  $D$ -димензионом простору:

$$\mathcal{H}^{(n)} = \frac{(-1)^n}{\omega(\mathbf{v})} \frac{\partial^n}{\partial v_{\alpha}} \omega(\mathbf{v}) \quad (2.22)$$

Тежинска функција  $\omega(\mathbf{v})$  се дефинише:

$$\omega(\mathbf{v}) = \frac{1}{(2\pi)^{D/2}} \exp\left(-\frac{v^2}{2}\right) \quad (2.23)$$

Коефицијент експанзије се дефинише:

$$a_{\alpha}^{(n)}(\mathbf{x}, t) = \int d\mathbf{v} f(\mathbf{x}, \mathbf{v}, t) \mathcal{H}_{\alpha}^{(n)}(\mathbf{v}) \quad (2.24)$$

Да би се добиле жељене макроскопске једначине, поступком који је предложио *Grad*, довољно је одредити 5 величина:  $\rho$ ,  $\rho \mathbf{u}$ ,  $\rho \epsilon$ ,  $\mathbf{P}$  и  $\mathbf{q}$ . Због тога се претходно дефинисани коефицијенти рачунају до степена 4.

На основу формуле (2.22), добијају се *Hermite*-ови полиноми до степена 4:

$$\mathcal{H}^{(0)} = 1 \quad (2.25)$$

$$\mathcal{H}_{\alpha}^{(1)} = v_{\alpha} \quad (2.26)$$

$$\mathcal{H}_{\alpha\beta}^{(2)} = v_{\alpha} v_{\beta} - \delta_{\alpha\beta} \quad (2.27)$$

$$\mathcal{H}_{\alpha\beta\gamma}^{(3)} = v_{\alpha} v_{\beta} v_{\gamma} - (\delta_{\alpha\beta} v_{\gamma} + \delta_{\alpha\gamma} v_{\beta} + \delta_{\beta\gamma} v_{\alpha}) \quad (2.28)$$

$$\begin{aligned} \mathcal{H}_{\alpha\beta\gamma\delta}^{(4)} = & v_{\alpha} v_{\beta} v_{\gamma} v_{\delta} - (\delta_{\alpha\beta} v_{\gamma} v_{\delta} + \delta_{\alpha\gamma} v_{\beta} v_{\delta} + \delta_{\alpha\delta} v_{\beta} v_{\gamma} + \\ & + \delta_{\beta\gamma} v_{\alpha} v_{\delta} + \delta_{\beta\delta} v_{\alpha} v_{\gamma} + \delta_{\gamma\delta} v_{\alpha} v_{\beta}) + \\ & + (\delta_{\alpha\beta} \delta_{\gamma\delta} + \delta_{\alpha\gamma} \delta_{\beta\delta} + \delta_{\alpha\delta} \delta_{\beta\gamma}) \end{aligned} \quad (2.29)$$

Ако се у општу једначину за коефицијент експанзије замене макроскопске величине (дефинисане једначинама (2.6), (2.7) и (2.12)) добијају се следеће једначине за коефицијенте до степена 4:

$$a^{(0)} = \rho \quad (2.30)$$

$$a_{\alpha}^{(1)} = \rho u_{\alpha} \quad (2.31)$$

$$a_{\alpha\beta}^{(2)} = P_{\alpha\beta} + \rho(u_{\alpha} u_{\beta} - \delta_{\alpha\beta}) \quad (2.32)$$

$$a_{\alpha\beta\gamma}^{(3)} = Q_{\alpha\beta\gamma} + u_{\alpha} a_{\beta\gamma}^{(2)} + u_{\beta} a_{\alpha\gamma}^{(2)} + u_{\gamma} a_{\alpha\beta}^{(2)} + (1 - D)\rho u_{\alpha} u_{\beta} u_{\gamma} \quad (2.33)$$

$$a_{\alpha\beta\gamma\delta}^{(4)} = R_{\alpha\beta\gamma\delta} - (P_{\alpha\beta}\delta_{\gamma\delta} + P_{\alpha\gamma}\delta_{\beta\delta} + P_{\alpha\delta}\delta_{\beta\gamma} + P_{\beta\gamma}\delta_{\alpha\delta} + P_{\beta\delta}\delta_{\alpha\gamma} + P_{\gamma\delta}\delta_{\alpha\beta}) + (\delta_{\alpha\beta}\delta_{\gamma\delta} + \delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}) \quad (2.34)$$

где су  $\mathbf{Q}$  и  $\mathbf{R}$  дати изразима:

$$\mathbf{Q} = \int d\mathbf{c} \, \mathbf{c} \mathbf{c} \mathbf{c} \mathbf{c} f \quad (2.35)$$

$$\mathbf{R} = \int d\mathbf{c} \, \mathbf{c} \mathbf{c} \mathbf{c} \mathbf{c} \mathbf{c} f \quad (2.36)$$

а притом су  $\mathbf{c} \mathbf{c} \mathbf{c}$  и  $\mathbf{c} \mathbf{c} \mathbf{c} \mathbf{c}$  тензорски производи  $\mathbf{c}$  са самим собом.

Величина дефинисана једначином (2.35) назива се тензор флуksа унутрашње енергије и повезана је са већ дефинисаним флуksом унутрашње енергије  $\mathbf{q}$  релацијом:

$$q_\alpha = Q_{\alpha\beta\beta} \quad (2.37)$$

Као што се може видети из једначине (2.21), пројекцијом на *Hermite*-ову базу функција расподеле се развија у бесконачан ред. Међутим, већ је наведено да у макроскопским једначинама дефинисаним у поглављу 2.3 фигурише само 5 величина. Стога се може извести закључак да се могу занемарити чланови после одређеног степена. Тиме се функција расподеле „скраћује“, односно губе се одређене информације које је оригинална функција расподеле садржала. Али, може се показати да те информације заправо и нису неопходне за симулације струјања флуида.

У складу са овим закључцима, функција расподеле се може записати на следећи начин:

$$f(\mathbf{x}, \mathbf{v}, t) \cong f^N(\mathbf{x}, \mathbf{v}, t) = \omega(\mathbf{v}) \sum_{n=0}^N \frac{1}{n!} \mathcal{H}_\alpha^{(n)}(\mathbf{v}) a_\alpha^{(n)}(\mathbf{x}, t) \quad (2.38)$$

при чему се сви чланови реда вишег од  $N$  занемарују.

У литератури је показано да ако се узме да је  $N = 4$ , из BGK једначине се могу извести Навије-Стоксове једначине за стишљив термални флуид [27]. Слично, ако се изврши скраћивање до реда 3, добијају се идентичне једначине одржања масе и количине кретања, као у претходном случају. Али, губитак информација из функције расподеле доводи до тога да једначина одржања енергије није више идентична и више не може

довољно тачно да симулира топлотна струјања. Оваква формулација ипак сасвим реално може да репрезентује изотермална струјања стишљивог флуида.

Скраћивањем реда до степена 2 добија се најчешће коришћен *lattice Boltzmann* модел. Овај модел важи само за изотермалне и слабо стишљиве флуиде. Најчешће се користи због тога што репрезентује и најчешће коришћен облик Навије-Стоксових једначина у рачунској динамици флуида. У наставку овог рада извођење ће бити спроведено управо за овај модел – за изотермално струјање нестишљивог флуида.

Сада се може дефинисати и равнотежна функција расподеле, развијена у ред:

$$f^0 = \omega \sum_{n=0}^N \frac{1}{n!} \mathcal{H}_\alpha^{(n)} a_{0\alpha}^{(n)} \quad (2.39)$$

где су коефицијенти  $a_{0\alpha}^{(n)}$  дефинисани једначином:

$$a_{0\alpha}^{(n)} = \int dv f^{(0)} \mathcal{H}_\alpha^{(n)} \quad (2.40)$$

У овом изразу (и у свим наредним изразима)  $N = 2$ , као што је већ наглашено.

Коефицијенти  $a_0^{(n)}$  се могу лако одредити применом Гаусове интеграције и заменом израза за макроскопске величине:

$$a_0^{(0)} = \rho \quad (2.41)$$

$$a_{0\alpha}^{(1)} = \rho u_\alpha \quad (2.42)$$

$$a_{0\alpha\beta}^{(2)} = \rho u_\alpha u_\beta + \rho(\theta - 1)\delta_{\alpha\beta} \quad (2.43)$$

Како се у овом моделу температура не узима у обзир (разматра се струјање нестишљивог, изотермалног флуида), закон одржања енергије нема никакав утицај и узима се да је  $\theta = 1$ . Тако да коефицијент  $a_{0\alpha\beta}^{(2)}$  има облик:

$$a_{0\alpha\beta}^{(2)} = \rho u_\alpha u_\beta \quad (2.44)$$

Заменом вредности коефицијената  $a_0^{(n)}$ , као и *Hermite*-ових полинома дефинисаних у претходном поглављу, добија се коначни израз за функцију расподеле:

$$f^{(0)} = \omega_i \rho \left( 1 + \mathbf{v} \cdot \mathbf{u} + \frac{1}{2} \mathcal{H}_{\alpha\beta}^{(2)} u_\alpha u_\beta \right) \quad (2.45)$$

## 2.5 Chapman-Enskog растављање

До сада су изведени сви потребни изрази и сада се они користе за пројектовање целе BGK једначине на *Hermite*-ову базу, а потом се коришћењем *Chapman-Enskog* растављања величине у добијеној једначини растављају на сабирке, тако да се добију потребни изрази за флуks унутрашње енергије и тензор напона.

BGK једначина, као што је већ изведено, гласи:

$$\frac{\partial f}{\partial t} + \mathbf{v} \frac{\partial f}{\partial \mathbf{x}} + \mathbf{g} \frac{\partial f}{\partial \mathbf{v}} = -\frac{1}{\tau} (f - f^{(0)}) \quad (2.46)$$

Д а би се ова једначина успешно пројектовала на *Hermite*-ову базу, потребно је трансформисати парцијалне изводе по координатама и брзинама. После трансформације, добијају се следећи изрази:

$$\int d\mathbf{v} \mathcal{H}_\alpha^{(n)} \mathbf{v} \nabla_\mathbf{x} f = \nabla_\mathbf{x} a_\alpha^{(n)} + (\nabla_\mathbf{x} a_\alpha^{(n-1)} + perm) \quad (2.47)$$

$$F(\mathbf{v}) \equiv \int d\mathbf{v} (\mathbf{g} \cdot \nabla_\mathbf{v}) (\mathcal{H}_\alpha^{(n)} f) = (\mathbf{g} a_\alpha^{(n-1)} + perm) \quad (2.48)$$

где  $F$  представља израз за силу, „*perm*“ означава све пермутације индекса, а  $\nabla$  је оператор дивергенције.

Ако се најпре претпостави да је функција расподеле једнака равнотежној вредности функције расподеле, BGK једначина добија облик:

$$\frac{\partial f^{(0)}}{\partial t} + \mathbf{v} \frac{\partial f^{(0)}}{\partial \mathbf{x}} - F^{eq} = 0 \quad (2.49)$$

где је  $F^{eq}$  дефинисано:

$$F^{eq} = (\mathbf{g} a_{0\alpha}^{(n-1)} + perm) \quad (2.50)$$

Коришћењем једначина (2.47) и (2.48), пројекција једначине (2.49) на *Hermite*-ову базу са полиномима  $n$  има облик:

$$\frac{\partial a_0^{(n)}}{\partial t} + \nabla_x a_0^{(n+1)} + \left( \nabla_x a_\alpha^{(n-1)} + perm \right) - \left( g a_{0\alpha}^{(n-1)} + perm \right) = 0 \quad (2.51)$$

Заменом вредности коефицијената  $a$  за  $n = 0$  и  $n = 1$ , добијају се две основне једначине рачунске динамике флуида – закони одржања масе и количине кретања:

$$\frac{\partial}{\partial t}(\rho) + \frac{\partial}{\partial x}(\rho u) = 0 \quad (2.52)$$

$$\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho I + \rho uu) = \rho g \quad (2.53)$$

Почетна апроксимација у овој анализи (да је  $f = f^{(0)}$ ) није прихватљива за коначно извођење. Наиме, применом ове апроксимације добијају се *Euler*-ове једначине, а циљ извођења је да се добију Навије-Стоксове једначине. Зато се дефиниција функције расподеле  $f$  мења и сматра се да  $f$  није једнако  $f^{(0)}$ , већ је врло близу те вредности, односно, важи једнакост:

$$f = f^{(0)} + f^{(1)}, \text{ при чему је } f^{(1)} \ll f^{(0)} \quad (2.54)$$

Сада је потребно одредити „ред величине“  $f^{(1)}$ . Уводе се две нове величине -  $L$  је карактеристична дужина посматраног домена, а  $c_s$  је карактеристична брзина честица, која се још назива и брзина звука у јединицама мреже (енг. *lattice units*). Карактеристична брзина се може изразити на следећи начин:

$$c_s \sim \sqrt{k_B T / m} = \sqrt{\theta} \quad (2.55)$$

Лева страна једначине (2.46) се сада може апроксимирати:

$$\frac{df}{dt} \sim \frac{c_s}{L} f = -\frac{1}{\tau} (f - f^{(0)}) \quad (2.56)$$

Заменом израза (2.65) у ову једначину, добија се:

$$f^{(0)} \sim -\frac{L}{c_s \tau} f^{(1)} \quad (2.57)$$

при чему је узет у обзир већ наведени услов да је  $f^{(1)} \ll f^{(0)}$ .

Са  $l = c_s \tau$  означен је средњи слободни пут честица (просечни пут који честица пређе између 2 судара). *Knudsen*-ов број дефинише се као  $Kn = \frac{l}{L}$  – однос средњег слободног пута честица и карактеристичне дужине. Јасно је да  $f^{(1)}$  мора бити пропорционално *Knudsen*-овом броју. Зато је битно нагласити да целокупни приступ и *BGK* модел заправо важе у области малог *Knudsen*-овог броја.

Ако се даље рачуна  $f^{(1)}$  из следеће једначине:

$$\frac{\partial f^{(0)}}{\partial t} + \mathbf{v} \frac{\partial f^{(0)}}{\partial \mathbf{x}} - F^{eq} = -\frac{1}{\tau} f^{(1)} \quad (2.58)$$

и ако се поново користи пројекција на *Hermite*-ову базу, добија се:

$$-\frac{1}{\tau} a_1^{(n)} = \frac{\partial a_0^{(n)}}{\partial t} + \nabla_{\mathbf{x}} a_0^{(n+1)} + \left( \nabla_{\mathbf{x}} a_{\alpha}^{(n-1)} + perm \right) - \left( \mathbf{g} a_{0\alpha}^{(n-1)} + perm \right) \quad (2.59)$$

где је  $a_{1\alpha}^{(n)}$  дато изразом:

$$a_{1\alpha}^{(n)} = \int d\mathbf{v} f^{(1)} \mathcal{H}_{\alpha}^{(n)} \quad (2.60)$$

Као што се може видети из једначина (2.41) – (2.44), коефицијенти  $a_0^{(n)}$  представљају функције густине  $\rho$  и количине кретања  $\mathbf{j} = \rho \mathbf{u}$ . Ако се узму у обзир једначине одржања масе и количине кретања (2.52) и (2.53), парцијални извод коефицијента  $a_0^{(n)}$  по времену се може трансформисати на следећи начин:

$$\begin{aligned} \frac{\partial a_0^{(n)}}{\partial t} &= \frac{\partial a_0^{(n)}}{\partial \rho} \frac{\partial \rho}{\partial t} + \frac{\partial a_0^{(n)}}{\partial \mathbf{j}} \frac{\partial (\rho \mathbf{u})}{\partial t} = \\ &= -\frac{\partial a_0^{(n)}}{\partial \rho} \frac{\partial (\rho \mathbf{u})}{\partial \mathbf{x}} + \frac{\partial a_0^{(n)}}{\partial \mathbf{j}} \cdot \left( \rho \mathbf{g} - \frac{\partial}{\partial \mathbf{x}} (\rho \mathbf{I} + \rho \mathbf{u} \mathbf{u}) \right) \end{aligned} \quad (2.61)$$

На основу већ наведених израза за коефицијенте  $a_0^{(n)}$  (једначине (2.41) – (2.44)), лако је одредити парцијалне изводе по  $\rho$  и  $\mathbf{j}$ :

$$\frac{\partial a_0^{(0)}}{\partial \rho} = 1 ; \quad \frac{\partial a_{0\alpha}^{(1)}}{\partial \rho} = 0 ; \quad \frac{\partial a_{0\alpha\beta}^{(2)}}{\partial \rho} = -u_{\alpha} u_{\beta} - \delta_{\alpha\beta} \quad (2.62)$$

$$\frac{\partial a_0^{(0)}}{\partial j_{\alpha}} = 0 ; \quad \frac{\partial a_{0\alpha}^{(1)}}{\partial j_{\beta}} = \delta_{\alpha\beta} ; \quad \frac{\partial a_{0\alpha\beta}^{(2)}}{\partial j_{\gamma}} = \delta_{\alpha\gamma} u_{\beta} + \delta_{\beta\gamma} u_{\alpha} \quad (2.63)$$

С обзиром на сва ова претходна извођења, једноставно је сада одредити све коефицијенте  $a_1^{(n)}$ :

$$a_1^{(0)} = a_{1\alpha}^{(1)} = a_{1\alpha\alpha}^{(2)} = 0 \quad (2.64)$$

$$a_{1\alpha\beta}^{(2)} = -2\tau\rho S_{\alpha\beta} \quad (2.65)$$

где је тензор деформације дефинисан као:

$$S_{\alpha\beta} = \frac{1}{2}(u_{\beta,\alpha} + u_{\alpha,\beta}) \quad (2.66)$$

Напомена: У горе наведеном изразу  $u_{\alpha,\beta}$  означава парцијални извод  $\frac{\partial u_\alpha}{\partial x_\beta}$ .

Треба обратити посебну пажњу на израчунавање коефицијента  $a_{1\alpha\beta}^{(2)}$ . Ако се изводи поступно, израз за коефицијент  $a_{1\alpha\beta}^{(2)}$  гласи:

$$-\frac{1}{\tau}a_{1\alpha\beta}^{(2)} = 2\rho S_{\alpha\beta} - (\rho u_\alpha u_\beta u_\gamma)_{,\gamma} = 2\rho S_{\alpha\beta} + \mathcal{O}(Ma^3) \quad (2.67)$$

тако да се практично занемарују чланови вишег реда.

У овом изразу,  $Ma$  представља *Mach*-ов број, који се дефинише као однос карактеристичне брзине флуида и „брзине звука“  $c_s$ .

$$Ma = \frac{u}{c_s} \quad (2.68)$$

Коришћењем апроксимације да је у оквиру LB симулација *Mach*-ов број мали, чланови вишег реда у једначини (2.67) се могу занемарити и на тај начин се добија већ написани израз (2.65).

Као последица апроксимације да је струјање у области малих *Mach* бројева, добија се коначан израз за ванравнотежну функцију расподеле  $f^{(1)}$  (енг. “*off-equilibrium*” part):

$$f^{(1)} = -\tau\rho\mathcal{H}_{i\alpha\beta}^{(2)}S_{\alpha\beta} \quad (2.69)$$

Да би се добио потпун систем једначина, и даље је потребно одредити тензор напона  $\mathbf{P}$  и флукс  $\mathbf{q}$ . Већ је наглашено да се у овом раду разматра струјање изотермалног

нестисљивог флуида, па се због тога утицај температуре занемарује. Самим тим више није потребно одређивати флукс  $\mathbf{q}$ , јер је он и иначе изостављен из система једначина.

Применом истог принципа растављања, који је већ коришћен приликом израчунавања функције расподеле, тензор напона се може

Primenom istog principa *Chapman-Enskog* rastavljanja, koji je već korišćen prilikom izračunavanja funkcije raspodele, tenzor napona se može rastaviti na 2 komponente:

$$P = P^{(0)} + P^{(1)} \quad (2.70)$$

На основу једначине (2.12), трансформацијама се добија:

$$P_{\alpha\beta}^{(n)} = \int d\mathbf{v} \left( \mathcal{H}_{\alpha\beta}^{(2)} - \mathcal{H}_{\alpha}^{(1)} u_{\beta} - \mathcal{H}_{\beta}^{(1)} u_{\alpha} + (\delta_{\alpha\beta} + u_{\alpha} u_{\beta}) \mathcal{H}^{(0)} \right) f^{(n)} \quad (2.71)$$

Заменом  $n = 0$ , добија се:

$$P_{\alpha\beta}^{(0)} = \rho \delta_{\alpha\beta} \quad (2.72)$$

Заменом  $n = 1$ , добија се:

$$P_{\alpha\beta}^{(1)} = \int d\mathbf{v} \mathcal{H}_{\alpha\beta}^{(2)} f^{(1)} \quad (2.73)$$

На основу већ одређене вредности  $f^{(1)}$  (дефинисане једначином (2.69)), добија се:

$$P_{\alpha\beta}^{(1)} = -2\tau\rho S_{\alpha\beta} \quad (2.74)$$

Укупна вредност тензора напона износи:

$$P_{\alpha\beta} = \rho \delta_{\alpha\beta} - 2\mu_s S_{\alpha\beta} \quad (2.75)$$

где је  $\mu_s = \rho\tau$  динамичка вискозност.

Ако се сада добијени израз за тензор напона замени у једначину дефинисану на почетку извођења (у једначину (2.18)), добија се позната Навије-Стоксова једначина за нестишљив флуид.

Треба напоменути да LB метода притисак  $p$  уводи у систем једначина преко једначине стања идеалног гаса:

$$p = \rho \theta = \rho k_B T / m \quad (2.76)$$

Коначно, систем једначина који се добија целокупним поступком, гласи:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho \mathbf{u}) = 0 \quad (2.77)$$

$$\rho \frac{d\mathbf{u}}{dt} + \frac{\partial}{\partial x}(p\mathbf{I} - 2\mu_s \mathbf{S}) = \rho \mathbf{g} \quad (2.78)$$

## 3 Дискретизација BGK једначине

---

BGK Boltzmann-ова једначина се односи на континуум и подразумева да постоји континуално поље брзина. Овакав облик једначине није погодан за нумеричко решавање. Да би било могуће направити програм који решава нумерички ову једначину, она претходно мора бити дискретизована. Али, дискретизација не може бити произвољно спроведена. Систем једначина мора остати такав да буду задовољене Навије-Стоксове једначине, да би цела метода могла да се користи за симулације струјања флуида.

Поступак дискретизације се спроводи у два корака. Најпре се дискретизује поље брзина, применом *Gauss-Hermite*-овог правила (интеграли се израчунавају као коначне суме, увођењем тежинских коефицијената), а затим се примењује трапезно правило другог реда, у циљу просторне и временске дискретизације. Поступак дискретизације је јако важан и посвећено је цело поглавље тој теми, делом и због саме имплементације, пошто се кроз тај поступак објашњава и изглед мреже, карактеристике функције расподеле и слично.

### 3.1 Дискретизација поља брзина

---

Функција расподеле  $f^N$  ( $N$  је ред скраћивања, у овом раду је разматран случај изотермалног, нестишљивог струјања, тако да је  $N = 2$ ) је изведена у претходном поглављу и она се даље користи да се на једноставан начин дискретизује поље брзина.

*Hermite*-ови коефицијенти експанзије  $a_\alpha^{(n)}$  се могу добити коришћењем  $f^N$ , а не потпуне функције расподеле  $f$ . Такође, показано је да све макроскопске једначине зависе само од ових коефицијената  $a_\alpha^{(n)}$ . У складу са тим, једначина којом се дефинишу ови коефицијенти (једначина (2.24)) може се трансформисати на следећи начин:

$$a_\alpha^{(n)}(x, t) = \int d\mathbf{v} f^N(x, \mathbf{v}, t) \mathcal{H}_\alpha^{(n)}(\mathbf{v}) = \int d\mathbf{v} \omega(\mathbf{v}) r(x, \mathbf{v}, t) \quad (3.1)$$

где је са  $r(\mathbf{x}, \mathbf{v}, t)$  означен полином степена не већег од  $2N$ , дефинисан као функција брзине  $\mathbf{v}$ . Овај интеграл се може израчунати ако се даље трансформише у тежинску суму, коришћењем поступка *Gauss-Hermite* квадратуре [34,35].

У овом конкретном случају, за полином  $r(\mathbf{v})$ , *Gauss*-ова квадратура тежи да што приближније одреди вредност интеграла  $\int \omega(\mathbf{v}) r(\mathbf{v}) d\mathbf{v}$ . То је могуће ако се одабере оптималан скуп апсциса  $v_i, i = 1, 2, \dots, n$ , тако да важи:

$$\int \omega(\mathbf{v}) r(\mathbf{v}) d\mathbf{v} \simeq \sum_{i=1}^n \omega_i r(v_i) \quad (3.2)$$

где је  $\omega_i, i = 1, 2, \dots, n$  скуп константних тежинских коефицијената.

За ово правило се каже да има алгебарски степен тачности  $m$ , ако за било које  $r$ , које је полином максимално  $m$ -тог степена, важи егзактна једнакост горње једначине.

Овај параметар (алгебарски степен тачности  $m$ ) је значајан зато што се помоћу њега дефинише број апсциса које ће се користити за тачно одређивање *Hermite*-ових коефицијената. У зависности од типа струјања које треба да се симулира, потребан је различит број коефицијената. За топлотно струјање стишљивог флуида узима се  $m = 9$ , за изотермално струјање стишљивог флуида узима се  $m = 7$ , а за случај који се разматра у овом раду – изотермално струјање нестишљивог флуида, алгебарски степен тачности је  $m = 5$ . Наравно, што је веће  $m$ , већи су и меморијски захтеви и потребно је више времена за прорачун.

У Додатку А дати су *Gauss-Hermite*-ови тежински коефицијенти и апсцисе за алгебарски степен тачности 5, који се користе за решавање дводимензионалних и тродимензионалних проблема.

Једначина (3.1) се применом описаног поступка своди на:

$$a_{\alpha}^{(n)}(\mathbf{x}, t) = \sum_{i=0}^{q-1} \omega_i r(\mathbf{x}, \mathbf{v}_i, t) = \sum_{i=0}^{q-1} \frac{\omega_i}{\omega(\mathbf{v}_i)} f^N(\mathbf{x}, \mathbf{v}_i, t) \mathcal{H}_{\alpha}^{(n)}(\mathbf{v}_i) \quad (3.3)$$

где су  $\omega_i$  и  $\mathbf{v}_i$  ( $i = 0, \dots, q - 1$ ) тежински коефицијенти и апсцисе *Gauss-Hermite*-ове квадратуре степена мањих или једнаких  $2N$ .

Оваквим израчунавањем интеграла и оваквом дискретизацијом поља брзине, дискретизована функција расподеле  $f^N(\mathbf{x}, \mathbf{v}_i, t)$ ,  $i = 0, \dots, q - 1$ , потпуно одређује

$f^N(x, v, t)$  и све особине величина које се израчунавају на основу  $f^N$  остају непромењене.

Као што се може видети из Додатка А, апсцисе нису целобројне дужине, а пожељно је због лакшег рачунања да то буде случај. Пошто су вредности апсциса ирационални бројеви ( $v_i = \sqrt{3}$ ), неопходно је увођење корекционог фактора да би се они избегли. Дефинише се константа

$$c_l = \frac{\Delta x}{\Delta t} \quad (3.4)$$

која се у литератури назива брзина светлости мреже и узима се да је то јединична вредност [36]. Практично, у имплементацији се вредности  $\Delta x$  и  $\Delta t$  се узимају тако да њихов однос буде једнак јединици. Више детаља о избору параметара и осталим специфичностима имплементације LB методе биће дато у наставку рада.

Са константом  $c_l$  се повезује и друга константа (брзина звука мреже)  $c_s$  и то релацијом:

$$c_s^2 = \frac{c_l^2}{3} \quad (3.5)$$

Већ је поменуто да се  $c_s$  може изразити као:

$$c_s \sim \sqrt{k_B T / m} = \sqrt{\theta} \quad (3.6)$$

и она зависи од избора дискретних брзина. За модел са 9 брзина, за дводимензионални домен, узима се да је  $c_s = 1/\sqrt{3}$ . То је наведено и у Додатку А.

Прорачуном се добијају одређени резултати. Да би они били конзистентни, све брзине вредности апсциса треба помножити константом  $c_s$ , да би се добиле јединичне вредности за апсцисе поља брзина. Уводи се смена за апсцисне осе:

$$c_i = c_s \cdot v_i \quad (3.7)$$

Већ је поменуто да се  $c_s$  често у литератури назива брзина звука мреже (енг. *lattice speed of sound*), али заправо представља само фактор скалирања.

Да би се исправно дискретизовала и функција расподеле, потребно је увести дискретне *Hermite*-ове полиноме:

$$\mathcal{H}_i^{(0)} = 1 \quad (3.8)$$

$$\mathcal{H}_{i\alpha}^{(1)} = v_{i\alpha} \quad (3.9)$$

$$\mathcal{H}_{i\alpha\beta}^{(2)} = v_{i\alpha} v_{i\beta} - c_s^2 \delta_{\alpha\beta} \quad (3.10)$$

Такође, увођење фактора скалирања доводи и до промена израза за коефицијенте експанзије, који сада гласе:

$$a^{(0)} = \rho \quad (3.11)$$

$$a_\alpha^{(1)} = \rho u_\alpha \quad (3.12)$$

$$a_{\alpha\beta}^{(2)} = P_{\alpha\beta} + c_s^2 \rho (u_\alpha u_\beta - \delta_{\alpha\beta}) \quad (3.13)$$

Да би се у потпуности дискретизовала једначина (2.46), треба пројектовати и равнотежну функцију расподеле  $f^{(0)}$  на *Hermite*-ову базу. Коефицијенти експанзије који се односе на равнотежну функцију расподеле су дати следећим једначинама:

$$a_0^{(0)} = \rho \quad (3.14)$$

$$a_{0\alpha}^{(1)} = \rho u_\alpha \quad (3.15)$$

$$a_{0\alpha\beta}^{(2)} = \rho u_\alpha u_\beta + c_s^2 \rho (\theta - 1) \delta_{\alpha\beta} \quad (3.16)$$

Пошто се у овом раду занемарује топлотно струјање, у једначини (3.16) треба узети у обзир да је  $\theta = 1$ :

$$a_{0\alpha\beta}^{(2)} = \rho u_\alpha u_\beta \quad (3.17)$$

Сада дискретизована функција расподеле  $f_i$  добија облик:

$$f_i(\mathbf{x}, t) = \frac{\omega_i}{\omega(\mathbf{v}_i)} f(\mathbf{x}, \mathbf{v}_i, t) \quad (3.18)$$

Дискретизована равнотежна функција расподеле има облик (примењено је скраћивање до реда 2):

$$f_i^{(0)} = \omega_i \sum_{n=0}^2 \frac{1}{c_s^{2n} n!} a_{0\alpha}^{(n)} \mathcal{H}_{i\alpha}^{(n)} = \omega_i \rho \left[ 1 + \frac{\mathbf{v}_i \cdot \mathbf{u}}{c_s^2} + \frac{1}{2c_s^4} \mathcal{H}_{i\alpha\beta}^{(2)} u_\alpha u_\beta \right] \quad (3.19)$$

где су са  $\mathcal{H}_{i\alpha}^{(n)}$  означени дискретизовани *Hermite*-ови полиноми (дефинисани једначинама (3.8)-(3.10)).

Аналогно претходним, може се дискретизовати и израз за силу  $F_i$ , који сада гласи:

$$F_i = \omega_i \sum_{n=1}^2 \frac{1}{n! c_s^2} \mathcal{H}_{i\alpha}^{(n)} a_{0\alpha}^{(n)} = \omega_i \rho \left( \frac{\mathbf{v}_i - \mathbf{u}}{c_s^2} - \frac{(\mathbf{v}_i \cdot \mathbf{u}) \mathbf{v}_i}{c_s^4} \right) \cdot \mathbf{g} \quad (3.20)$$

Ванравнотежни део функције расподеле у дискретизованом облику гласи:

$$f_i^{(1)} = -\omega_i \frac{\tau \rho}{c_s^2} \mathcal{H}_{i\alpha\beta}^{(2)} S_{\alpha\beta} \quad (3.21)$$

Када је у питању тензор притиска, неопходно је узети у обзир фактор скалирања  $c_s$  и у једначини којом се он дефинише:

$$\mathbf{P}^{(0)} = c_s^2 \rho \mathbf{I} \quad (3.22)$$

$$\mathbf{P}^{(1)} = -\frac{\tau}{c_s^4} \mathbf{S} \quad (3.23)$$

На исти начин је потребно изменити и израз за динамичку вискозност и једначину која дефинише везу између притиска флуида и густине флуида:

$$\mu_s = c_s^2 \rho \tau \quad (3.29)$$

$$p = c_s^2 \rho \theta = c_s^2 \rho k_B T / m \quad (3.30)$$

Коначно, може се написати BGK једначина, дискретизована по пољу брзина и она гласи:

$$\frac{\partial f_i}{\partial t} + \mathbf{v}_i \frac{\partial f_i}{\partial \mathbf{x}} = -\frac{1}{\tau} (f_i - f_i^{(0)}) + F_i \quad (3.24)$$

Макроскопске величине се израчунавају преко функције расподеле. У претходном поглављу су дати изрази којима се оне израчунавају интеграљењем по бесконачном домену брзина. Пошто је спроведена дискретизација поља брзина, ове величине се сада могу израчунати као тежинске суме по коначном броју дискретних брзина:

$$\rho = \sum_i f_i \quad (3.25)$$

$$\rho \mathbf{u} = \sum_i \mathbf{v}_i f_i \quad (3.26)$$

$$\rho \epsilon = \frac{1}{2} \sum_i \mathbf{c}_i^2 f_i \quad (3.27)$$

$$\mathbf{P} = \sum_i \mathbf{c}_i \mathbf{c}_i f_i \quad (3.28)$$

где је  $\mathbf{c}_i = \mathbf{v}_i - \mathbf{u}$  дискретна микроскопска брзина, што је новоуведена величина, наспрам већ дефинисане континуалне микроскопске брзине.

### 3.2 Временска и просторна дискретизација

Након дискретизације поља брзина, потребно је дискретизовати конвективни члан једначине (3.24). Поступак временске и просторне дискретизације се може наћи у литератури [37,38,27].

BGK Boltzmann-ова једначина се може написати и у следећем облику:

$$\frac{df_i}{dt} = -\frac{1}{\tau} (f_i - f_i^{eq}) \quad (3.31)$$

где је

$$f_i^{eq} = f_i^{(0)} + \tau F_i \quad (3.32)$$

Интеграцијом једначине (3.31) добија се:

$$\begin{aligned} f_i(\mathbf{x} + \mathbf{v}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) \\ = -\frac{1}{\tau} \int_0^{\Delta t} \left( f_i(\mathbf{x} + \mathbf{v}_i s, t + s) - f_i^{eq}(\mathbf{x} + \mathbf{v}_i s, t + s) \right) ds \end{aligned} \quad (3.33)$$

где је  $\Delta t$  временски корак.

Десна страна ове једначине се може апроксимирати трапезним правилом:

$$\begin{aligned} -\frac{1}{\tau} \int_0^{\Delta t} \left( f_i(\mathbf{x} + \mathbf{v}_i s, t + s) - f_i^{eq}(\mathbf{x} + \mathbf{v}_i s, t + s) \right) ds \\ = -\frac{\Delta t}{2\tau} \left[ f_i(\mathbf{x} + \mathbf{v}_i \Delta t, t + \Delta t) - f_i^{eq}(\mathbf{x} + \mathbf{v}_i \Delta t, t + \Delta t) + f_i(\mathbf{x}, t) \right. \\ \left. - f_i^{eq}(\mathbf{x}, t) \right] + \mathcal{O}(\Delta t^2) \end{aligned} \quad (3.34)$$

Тако да сада једначина (3.33) има облик:

$$\begin{aligned} f_i(\mathbf{x} + \mathbf{v}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) \\ = -\frac{\Delta t}{2\tau} \left[ f_i(\mathbf{x} + \mathbf{v}_i \Delta t, t + \Delta t) - f_i^{eq}(\mathbf{x} + \mathbf{v}_i \Delta t, t + \Delta t) + f_i(\mathbf{x}, t) \right. \\ \left. - f_i^{eq}(\mathbf{x}, t) \right] + \mathcal{O}(\Delta t^2) \end{aligned} \quad (3.35)$$

У даљем тексту се за временски корак узима да је  $\Delta t = 1$ , што је погодно и за нумеричко решавање једначина.

Горе наведени облик једначине је имплицитан, а за решавање је погодније користити експлицитни облик. Због тога се уводи нова величина:

$$\bar{f}_i(\mathbf{x}, t) = f_i(\mathbf{x}, t) + \frac{1}{2\tau} \left( f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t) \right) \quad (3.36)$$

Ако се израз (3.36) замени у једначини (3.35) и добијена једначина потом среди, добија се:

$$\bar{f}_i(\mathbf{x} + \mathbf{v}_i, t + 1) - \bar{f}_i(\mathbf{x}, t) = -\frac{1}{\tau} \left( \bar{f}_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t) \right) \quad (3.37)$$

Овим је добијен експлицитни облик једначине, при чему је уведено модификовано време релаксације:

$$\bar{\tau} = \tau + \frac{1}{2} \quad (3.38)$$

У нумеричким симулацијама се израчунава управо  $\bar{f}_i$  и користи се модификовано време релаксације.

У једначини (3.35) је назначено да се чланови вишег реда занемарују ( $\mathcal{O}(\Delta t^2)$  у поменутој једначини). Наравно, овим занемаривањем је направљена одређена грешка. Да би се кориговала та грешка уведен је корекциони фактор  $\frac{1}{2}$  у изразу (3.38). На тај начин је постигнуто да *lattice Boltzmann* методнема нумеричке нестабилности или грешке, као што је то био случај са претходником (*LGA* методом) или са неким другим методама.

Наиме, на почетку овог поглавља је наглашено да се поступком дискретизације мора очувати својство *BGK* једначине да се из ње могу извести Навије-Стоксове једначине. Уколико би се овако дискретизована *BGK* једначина развила у *Taylor*-ов ред, добио би се исти облик Навије-Стоксових једначина, који је добијен трансформацијом континуалне *BGK* једначине [39]. Тиме је доказано да је дискретизација спроведена исправно. Управо применом овог другог поступка утврђује се колика је поменута грешка занемаривања чланова вишег реда [40].

Још једна предност увођења модификованог времена релаксације јесте и то што омогућава примену експлицитних временских корака, дакле бољу и прикладнију временску дискретизацију. Али, измена времена релаксације, доводи и до измена осталих величина симулације, што је неопходно да би се обезбедила конзистентност модела.

Раније коришћено  $\tau$  треба заменити са  $\bar{\tau} - \frac{1}{2}$  у изразу којим се дефинише вискозност:

$$\mu_s = c_s^2 \rho \left( \bar{\tau} - \frac{1}{2} \right) \quad (3.39)$$

Овако добијена вискозност се назива и „вискозност мреже“ (енг. *lattice viscosity*).

Како је напоменуто да се  $f_i$  сада више не рачуна директно, већ се израчунава  $\bar{f}_i$ , треба сагледати и измене до којих услед ове промене долази у изразима за макроскопске величине.

Ако се узму у обзир чињенице да је:  $\sum_i f_i = \sum_i f_i^{(0)}$  и  $\sum_i F_i = 0$ , израз за густину гласи:

$$\bar{\rho} = \sum_{i=1}^{q-1} \bar{f}_i = \rho \quad (3.40)$$

Дакле, када је густина у питању, нема никаквих измена. Међутим, израз за количину кретања сада гласи:

$$\rho \bar{u} = \sum_i v_i \bar{f}_i = \rho u - \rho g/2 \quad (3.41)$$

Из ове једнакости се види да брзина која се у прорачуну израчунава на основу функције  $\bar{f}_i$  и која је означена са  $\bar{u}$ , не представља „физичку брзину“ струјања, која се може израчунати преко функције расподеле  $f_i$ . Да би се одредила физичка брзина (која је једна од најважнијих карактеристика струјања на макроскопском нивоу), потребно је извршити следеће прерачунавање:

$$u = \bar{u} + \frac{g}{2} \quad (3.42)$$

О овоме треба водити рачуна приликом анализе резултата симулације, односно приликом имплементације LB методе. Треба приметити и да је у случају да не постоји утицај спољашњих сила (када је  $g = 0$ ) брзина иста без обзира на промене у рачунању функције расподеле ( $u = \bar{u}$ ).

Још једна ствар коју треба нагласити јесте да је  $f_i^{eq}$  дефинисано као функција „физичког“ времена релаксације  $\tau$  и физичке брзине. Имајући у виду једначину (3.32) којом је дефинисано  $f_i^{eq}$ , десна страна једначине (3.37) се трансформисати на следећи начин:

$$\begin{aligned} & -\frac{1}{\tau} \left( \bar{f}_i(x, t) - f_i^{(0)}(\rho, u) - \tau F_i(\rho, u) \right) \\ & = -\frac{1}{\tau} \left( \bar{f}_i(x, t) - f_i^{(0)}(\rho, u) \right) + \left( 1 - \frac{1}{2\tau} \right) F_i(\rho, u) \end{aligned} \quad (3.43)$$

Сада се може написати једначина која репрезентује целу LB нумеричку шему и која се користи приликом имплементације ове методе:

$$\bar{f}_i(\mathbf{x} + \mathbf{v}_i, t + 1) - \bar{f}_i(\mathbf{x}, t) = -\frac{1}{\bar{\tau}} \left( \bar{f}_i(\mathbf{x}, t) - f_i^{(0)}(\rho, \mathbf{u}) \right) + \left( 1 - \frac{1}{2\bar{\tau}} \right) F_i \quad (3.44)$$

Када се имплементира LB метода, најчешће се ова једначина раздваја на два корака – сударе (енг. *collision step*) и пропагацију (енг. *propagation step*). Могу се дефинисати две вредности функције расподеле -  $\bar{f}_i^{in}$  и  $\bar{f}_i^{out}$ , које представљају вредности дискретизоване функције расподеле пре и после судара. Наведени кораци се могу описати једначинама:

Судари (енг. *collision*):

$$\bar{f}_i^{out}(\mathbf{x}, t) = \bar{f}_i^{in}(\mathbf{x}, t) - \frac{1}{\bar{\tau}} \left( \bar{f}_i^{in}(\mathbf{x}, t) - f_i^{(0)}(\rho, \mathbf{u}) \right) + \left( 1 - \frac{1}{2\bar{\tau}} \right) F_i \quad (3.45)$$

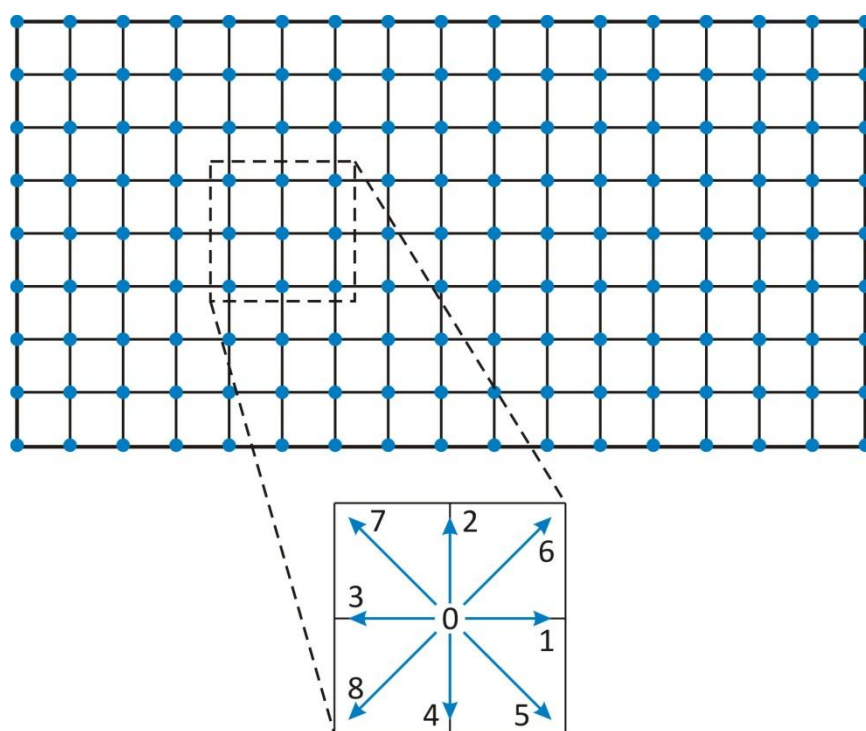
Пропагација (енг. *propagation*):

$$\bar{f}_i^{in}(\mathbf{x} + \mathbf{v}_i, t + 1) = \bar{f}_i^{out}(\mathbf{x}, t) \quad (3.46)$$

Сваки од ова два корака се мора применити на цео систем (на све чворове мреже, односно на све честице), пре него што почне следећи корак. Корак у коме се разматрају судари је локалног карактера (једначине за чворове су међусобно независне и нису спрегнуте), док се у кораку у коме се врши пропагација узима у обзир само тренутно разматрани чвор и неколико најближих суседних чворова. Управо због тога је LB метод јако погодан за паралелизацију.

## 4 Детаљи имплементације LB методе

У оквиру овог Мастер рада развијен је софтвер базиран на LB методи и једначинама које су наведене у претходним поглављима. У овом поглављу ће бити детаљније објашњено како се применом ових једначина симулира струјање флуида и како то практично изгледа на примеру дводимензионалне мреже. Основна идеја имплементације је да се разматрани домен подели на одређени број идентичних елемената који се другачије називају ћелије (енг. *cells*). За сваки чвор овако формиране мреже дефинише се функција расподеле. Како је спроведена дискретизација функције расподеле, поступком који је описан у поглављу 3, за сваки чвор је потребно дефинисати 9 оваквих вредности. Овај тип мреже се најчешће означава са **D2Q9**, јер се ради о дводимензионалном домену, са 9 апсциса. На слици 4.1 приказан је домен у дводимензионалном простору и мрежа ћелија, а такође су за један чвор су приказани смерови дискретизоване функције расподеле.



Слика 4.1 - Дводимензионална мрежа и дискретизована функција расподеле [41]

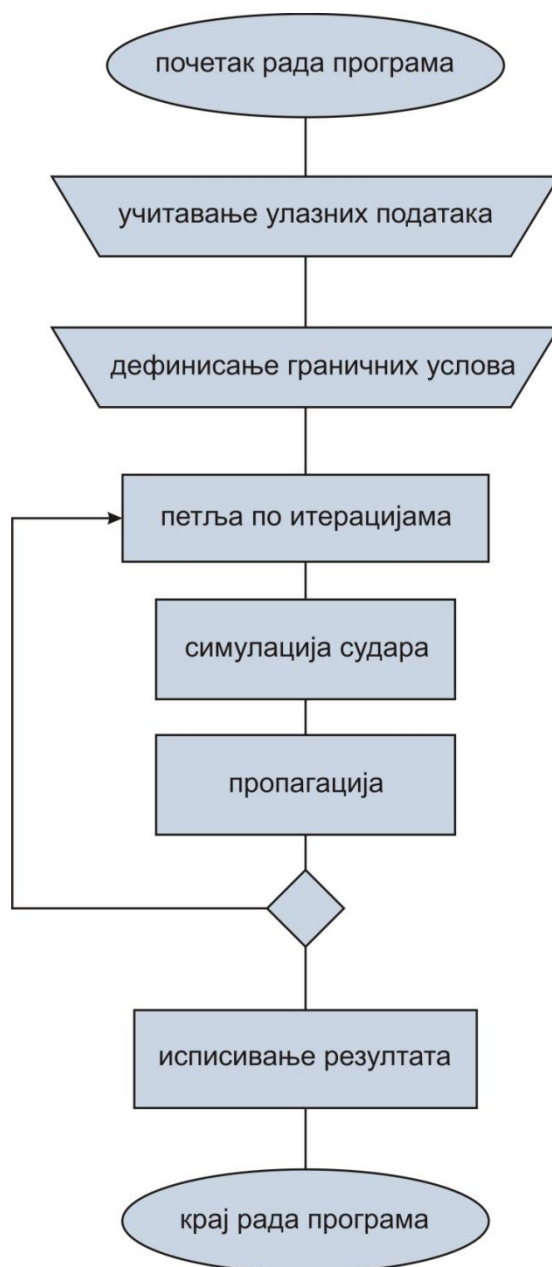
Овако означене компоненте функције расподеле се могу израчунати на основу једначине (3.19). Смерови означени бројевима 1-8 на слици 4.1 означавају апсцисе по којима је извршена дискретизација поља брзина. Апсцисе Gauss-Hermite-ове квадратуре су дефинисани у Додатку А. Као што је наведено у поглављу 3, у питању су ирационални бројеви, па је неопходно извршити корекцију, према једначини (3.7). Корекциони фактор (такозвана брзина звука) за овај тип мреже има вредност  $c_s = \frac{1}{\sqrt{3}}$ . На основу тога се могу дефинисати компоненте вектора који дефинишу смерове апсциса у природном координатном систему и оне су дате у следећој табели:

| Правац | Вектор   |
|--------|----------|
| 0      | (0, 0)   |
| 1      | (1, 0)   |
| 2      | (0, 1)   |
| 3      | (-1, 0)  |
| 4      | (0, -1)  |
| 5      | (1, 1)   |
| 6      | (-1, 1)  |
| 7      | (-1, -1) |
| 8      | (1, -1)  |

Нису све функције расподеле исте тежине, односно исте важности. Због тога се дефинишу и тежине апсциса  $\omega_i$ . Њихове вредности су такође дате у Додатку А.

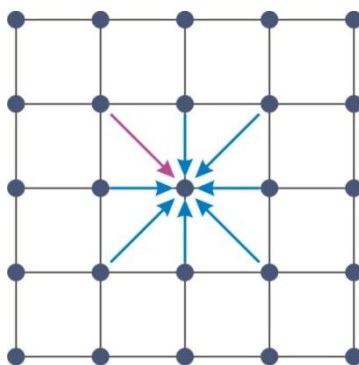
#### 4.1 Алгоритам програма

Програм за симулацију струјања флуида, базиран на LB методи написан је у програмском језику C++. Постоје функције за манипулацију подацима које се односе на чворове појединачно, на целу мрежу (у оквиру ње се разматрају интеракције међу чворовима), као и целе симулације (која контролише све активности). Све ове функције су имплементирани у оквиру неколико класа, тако да постоји класа којом се дефинише чвор, мрежа и симулација. Алгоритам програма је шематски приказан на слици 4.2.

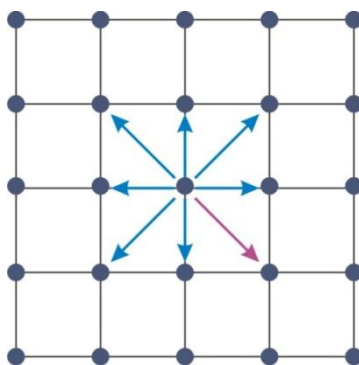


Слика 4.2 - Алгоритам програма базираног на LB методи

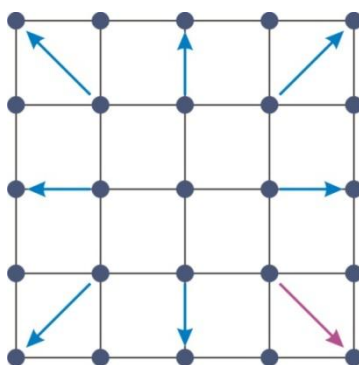
Важно је напоменути да је неопходно да постоји петља по итерацијама, због тога што се проблем решава у великом броју временских корака, док се не достигне стационарно решење. У оквиру те петље се наизменично спроводе два корака који су описани једначинама (3.45) и (3.46). На следећим сликама је приказано како се мења функција расподеле у једном чвору мреже пре почетка једног временског корака (слика 4.3), после моделирања судара (слика 4.4) и после пропагације (слика 4.5).



Слика 4.3 - Функција расподеле пре почетка временског корака



Слика 4.4 - Функција расподеле после моделирања судара (првог корака)



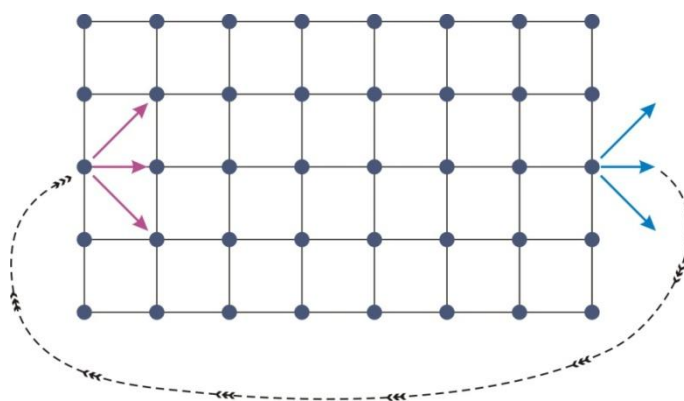
Слика 4.5 - Функција расподеле после пропагације (другог корака)

## 4.2 Гранични услови у LB методи

Сва претходна извођења нису узела у обзир граничне услове. Ипак, да би се добили валидни резултати и да би се исправно симулирало струјање флуида, неопходно је исправно дефинисати граничне услове. У овом раду ће бити споменуто неколико типова граничних услова који су имплементирани и коришћени при решавању примера приказаних у овом раду.

### 4.2.1 Периодични гранични услов

Ово је најједноставнији тип граничних услова. Практично, он се може тумачити као да су улаз и излаз домена спојени. У самом програму, овакав гранични услов се имплементира у оквиру корака пропагације. За све оне чворове који се налазе на граници домена, компоненте функције расподеле које треба да пропагирају ван границе домена се „преусмеравају“ тако да се ове вредности преносе на чворове који се налазе на другој (супротној) граници домена. Ово је илустровано на слици 4.6.



Слика 4.6 - Периодични гранични услов

### 4.2.2 Одбојни гранични услов

Овај гранични услов је јакo једноставан за имплементацију и један је од разлога популарности ове методе. Примењује се када је потребно креирати домene комплексне геометрије, као што је случај када се симулира струјање у порозној средини [41]. Једино је потребно дефинисати одређени чвор мреже као солид, односно препреку, и у таквим чворовима се не рачунају компоненте функције расподеле на исти начин као у осталим чворовима.

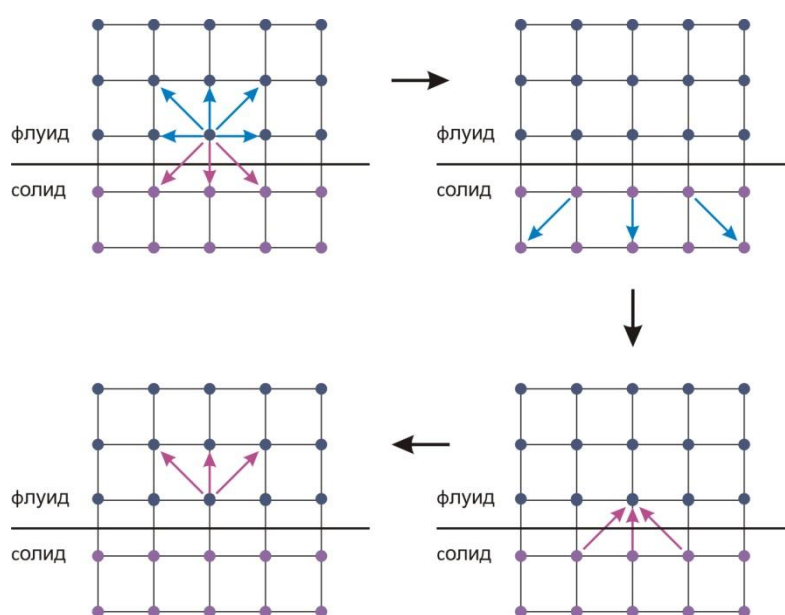
Ако се са  $opp(i)$  означи правац супротан правцу  $i$  у мрежи, вредности апсциса у тим правцима су повезане једначином:

$$v_{opp(i)} = -v_i \quad (4.1)$$

У чворовима мреже који су означени као солид чворови и који се третирају применом одбојног граничног услова (енг. *bounce-back*) вредности функције расподеле се израчунавају применом следеће формуле:

$$f_i = f_{opp(i)} \quad (4.2)$$

На слици 4.7 је приказан поступак примене *bounce-back* граничног услова, уз претпоставку да је граница солид-флуид домена постављена на половини мреже између солид и флуид чвора.

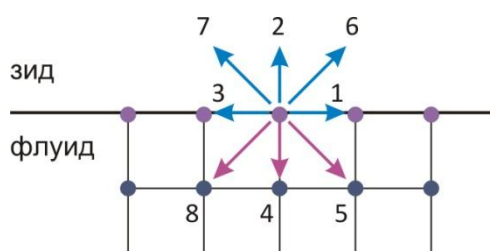


Слика 4.7 - Илустрација bounce-back граничног услова

Ипак, *bounce-back* гранични услов има и својих недостатака, који су разматрани у литератури [42,43]. Кључни недостатак је што у случају лоше имплементације може да дође до нестабилности и нарушавања једначине континуитета. Детаљном анализом је утврђено да се применом овог граничног услова задржава тачност другог реда, само када се граница солида и флуида налази између самих чворова мреже [44], као што је илустровано на слици 4.7. Ако је граница између два домена сачињена од правих линија, онда је пожељно користити овакав приступ. Међутим, када је у питању комплексна геометрија, криволинијска или када се посматра солид унутар флуида и интеракција солида са флуидом, ефикасније је користити нешто другачији приступ о коме ће бити речи у наредном поглављу.

#### 4.2.3 Задавање брзине и притиска као граничних услова

Уопштено када су гранични услови у питању, у оним чворовима који се налазе на граници домена, после корака пропагације постојаће одређене информације у функцији расподеле који потичу од зида, односно које немају физички карактер. С обзиром да се пропагацијом информације преносе са једног чвора на њему суседне чворове, јасно је да би те информације за граничне чворове требало да дођу од чвора унутар зида. Како се ти чворови не симулирају, ове информације о функцији расподеле које недостају морају на неки начин бити израчунате. Такође мора да се води рачуна о томе да цео концепт о ком је дискутовано у поглављу 2 буде задовољен и у чворним тачкама. Дискусија о очувању једначине континуитета може се наћи у литератури [45].



Слика 4.8 - Информације о функцији расподеле у граничном чвору

Као што се може видети на слици 4.8, у граничном чвору су познате вредности дискретизоване функције расподеле  $f_i$ , за  $i = 0,1,2,3,6,7$ , док су за  $i = 4,5,8$  вредности непознате. Али, управо зато се за граничне чворове дефинишу вредности макроскопских величина – брзине и притиска.

Када су у питању проблеми симулације струјања флуида, најчешће се задаје вредност брзине и притиска (то је тзв. *Dirichlet*-ов гранични услов) или се задају изводи ових

величина, односно флуксеви одређених величина (то је тзв. *Neumann*-ов гранични услов). Треба напоменути да се у LB методи уместо притиска практично задаје густина, пошто су ове две величине повезане једначином стања (2.76). На основу ових макроскопских вредности је могуће израчунати вредности компоненти функције расподеле и то тако да на крају буду задовољене једначине (3.25) и (3.26).

Постоји неколико типова граничних услова за брзине и притиске који се могу наћи у литератури – гранични услов које је предложио *Inamuro* и други [43], *Zou/He* приступ [46], регуларизована метода [47] и метода коначних разлика, базирана на идеји *Skordos*-а [48]. Сви ови гранични услови су такозвани „мокри“ гранични услови, јер се сматра да се чворови на које се односе налазе у флуиду, инфинитезимално близу зида. То је супротно од *bounce-back* граничног услова, који се сматра „сувим“ јер се чворови на које се он односи налазе у солиду (унутар зида). Последица ове чињенице је да је након рачунања непознатих компоненти функције расподеле неопходно спровести корак судара на чворовима који се сматрају граничним, док то код *bounce-back*-а није потребно.

У наставку ће бити описан регуларизовани гранични услов (енг. *regularized boundary condition*). Сматра се да је директно задата брзина или притисак (густина), тако да се разматра *Dirichlet*-ов гранични услов. Наравно, како се задаје само једна макроскопска величина, најпре је неопходно утврдити једначину којом су ове две величине повезане. Поступак извођења који је овде наведен може се наћи и у литератури [27,46].

Ако се са  $\mathbf{n}$  дефинише вектор нормале на зид, тако да смер вектора буде ка спољашњости домена, могу се дефинисати и следеће три величине:

$$\rho_0 = \sum_{i \in I_0} f_i \quad (4.3)$$

где је  $I_0 = \{i | v_i \cdot n = 0\}$

$$\rho_- = \sum_{i \in I_-} f_i \quad (4.4)$$

где је  $I_- = \{i | v_i \cdot n < 0\}$

$$\rho_+ = \sum_{i \in I_+} f_i \quad (4.5)$$

где је  $I_+ = \{i | v_i \cdot n > 0\}$

У случају који је приказан на слици 4.8, ове величине би добиле вредности:

$$\begin{aligned}\rho_0 &= f_0 + f_1 + f_3 \\ \rho_+ &= f_2 + f_6 + f_7 \\ \rho_- &= f_4 + f_5 + f_8\end{aligned}\tag{4.6}$$

Укупна густина се рачуна као:

$$\rho = \rho_0 + \rho_+ + \rho_-\tag{4.7}$$

Ако се са  $u_\perp$  дефинише пројекција вектора брзине на вектор нормале  $\mathbf{n}$

$$u_\perp = \mathbf{u} \cdot \mathbf{n}\tag{4.8}$$

на основу једначине (3.26) и претходно наведених једначина (4.3)-(4.5), следи:

$$u_\perp = \frac{\rho_+ - \rho_-}{\rho}\tag{4.9}$$

Комбинацијом новодобијене једнакости и једначине (4.7) добија се веза између брзине и густине:

$$u_\perp = \frac{2\rho_+ + \rho_0}{\rho} - 1\tag{4.10}$$

Како су вредности  $\rho_0$  и  $\rho_+$  познате за сваки чвор мреже, на основу задате вредности брзине лако се може израчунати густина и обрнуто. Треба узети у обзир да је ова релација изведена уз претпоставку да су у питању прави зидови. Ако то није случај, потребно је додатно разматрање и нешто другачије извођење [49]. У примерима који су део овог рада задају се брзине и притисци само на правим зидовима, тако да може бити коришћена релација (4.10).

Приступ који је коришћен у овом раду је базиран на томе да се не одређују само непознате компоненте функције расподеле, већ се одређују све компоненте, на основу задатих вредности макроскопских величина. Најједноставнији начин би био да се све компоненте замене својим равнотежним вредностима. Међутим, иако би у том случају било могуће израчунати вредности брзине и густине, то није довољно да би резултат био задовољавајући. Да би се добила тражена тачност, неопходно је да се добије и тачна вредност тензора напона. То практично значи да вредност добијена применом једначине (2.73) мора да се слаже са хидродинамичком вредношћу (која је већ

дефинисана једначином (2.75), али је овде поновљена, ради лакшег праћења поступка):

$$P_{\alpha\beta} = \rho\delta_{\alpha\beta} - 2\mu_s S_{\alpha\beta} \quad (4.11)$$

Када је у питању регуларизована метода, коју су развили *Latt* и *Chopard* [47], најпре се одређује вредност тензора  $\mathbf{P}^{(1)}$ , на основу већ познатих компоненти функције расподеле. Сматра се да непознате компоненте ванравнотежног дела функције расподеле добијају вредности компоненти функције расподеле за супротан смер (слично као код *bounce-back* граничног услова), применом једначине (4.2). То се може описати на следећи начин:

$$f_i^{(1)} = f_{opp(i)}^{(1)} \quad (4.12)$$

Укупна вредност непознатих компоненти тада гласи:

$$f_i = f_i^{(0)}(\rho, \mathbf{u}) + f_{opp(i)} - f_{opp(i)}^{(0)} \quad (4.13)$$

Треба напоменути да се ова једначина користи само привремено, за израчунавање тензора  $\mathbf{P}^{(1)}$ , а не у оквиру коначног облика граничног услова. Тензор  $\mathbf{P}^{(1)}$  се израчунава применом једначине (2.73), која је овде поновљена:

$$P_{\alpha\beta}^{(1)} = \int d\mathbf{v} \mathcal{H}_{\alpha\beta}^{(2)} f^{(1)} \quad (4.14)$$

Ако се има у виду симетрија тензора  $\mathcal{H}_{i\alpha\beta}^{(2)}$  и  $\mathbf{P}^{(1)}$  и ако се примени једначина (4.11), једначина (2.69) се може написати у нешто другачијем облику:

$$f^{(1)} = -\frac{\rho\tau\omega_i}{c_s^2} \mathcal{H}_{i\alpha\beta}^{(2)} S_{\alpha\beta} = \frac{\omega_i}{2c_s^4} \mathcal{H}_{i\alpha\beta}^{(2)} P_{\alpha\beta}^{(1)} \quad (4.15)$$

Применом ове једначине може се одредити ванравнотежни део функције расподеле на основу тензора  $\mathbf{P}^{(1)}$ . Ова процедура се назива регуларизација, јер се апсолутно (без нумеричке грешке) поштује једначина (4.11).

Коначно, вредност свих компоненти функције расподеле се сада може одредити на следећи начин:

Напомена: у овој једначини су са  $h_i$  обележене нове вредности функције расподеле, због поједностављења ознака.

$$h_i = f_i^{(0)}(\rho, \mathbf{u}) + \frac{\omega_i}{2c_s^4} \mathcal{H}_{i\alpha\beta}^{(2)} P_{\alpha\beta}^{(1)} \quad (4.16)$$

Након овако израчунатих компоненти функције расподеле, на граничне чворове се примењују и корак моделирања судара и корак пропагације, као и на све остале чворове. Једино је било потребно извршити корекцију због непознатих компоненти које „долазе“ од чворова у зиду. Очигледно је да се имплементацијом овог граничног услова тачно реконструишу не само вредности брзине и густине (притиска), већ и тензора напона.

## 5 Моделирање солид-флуид интеракције

---

Постоје одређени проблеми код којих је неопходно да се истовремено решавају два или више физичких система који међусобно интерагују. Такав пример постоји и у области моделирања струјања флуида, када се унутар домена флуида налазе честице, односно солиди. При томе флуид делује одређеним силама на солид и проузрокује кретање или деформације, а важи и обрнуто – солид утиче на струјање флуида. То практично значи да солид и флуид формирају један спрегнути механички систем. Да би се симулирао један овакав систем потребно је разматрати оба домена истовремено, односно моделирати солид-флуид интеракцију.

Када се решавају овакви мултидисциплинарни проблеми могуће су две алтернативе: писање новог солвера који ће истовремено решавати оба домена или кориговање већ постојећих програма за анализу солида и флуида појединачно. Програми за решавање солида и флуида су веома различити, због саме проблематике коју решавају, тако да је тешко уклопити их да раде заједно.

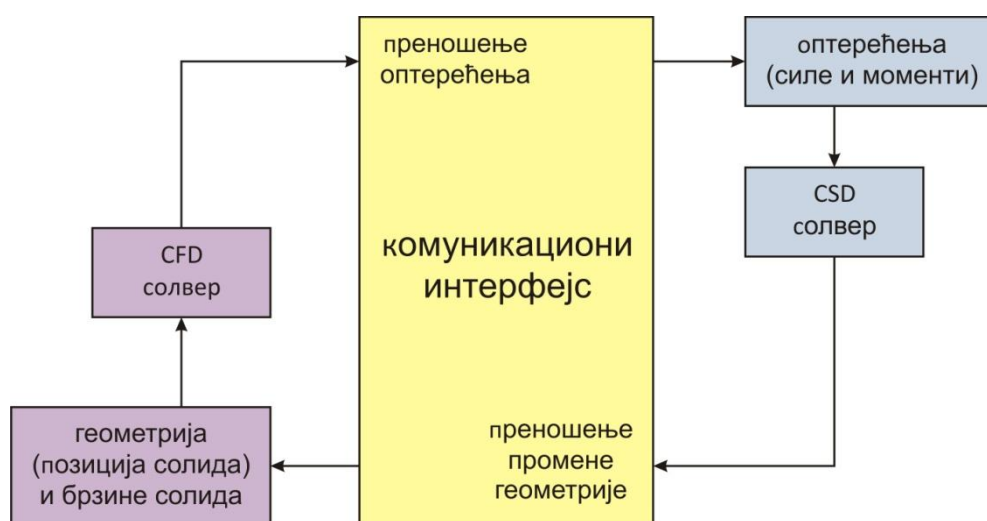
Постоје два приступа моделирању солид-флуид интеракције: слабо и јако спрезање. У случају слабог спрезања домени за солид и флуид се решавају појединачно, а сви потребни параметри добијени из солвера за један домен се прослеђују солверу за други домен. Посебан спољашњи програм управља решавањем оба проблема и обезбеђује правилну размену информација. У случају јаког спрезања оба домена се моделирају истовремено, као да је у питању један механички систем. За одређене проблеме, лакше је користити слабо спрезање, због једноставније имплементације. Међутим, постоје и неки недостаци, пре свега проблем временске интеграције. Пошто су физичке карактеристике солида и флуида различите, није увек могуће користити исти временски корак за нумеричко решавање. Са друге стране, јако спрезање је применљивије када је потребно прецизно моделирати кретање солида унутар флуидног домена. Али и овај приступ има своје недостатке. Солвер који истовремено решава оба домена је знатно комплекснији и ради спорије, што је очекивано с обзиром да се повећава број једначина у систему једначина.

У овом раду су коришћена оба приступа. У одређеним примерима је коришћена имплементација LB методе са slabим спрезањем, а у другом делу поглавља са

примерима коришћено је јако спрезање, уз примену *Immersed boundary* методе. У овом поглављу ће стога бити дате теоријске основе оба приступа.

### 5.1 Слабо спрезање

Слабо спрезање се састоји у томе да се наизменично решавају домен солида и домен флуида. Главна предност ове методе је што се могу користити већ постојећи солвери за солид и флуид, са врло мало измена. За домен солида се користи *CSD* солвер (енг. *computational solid dynamics* – *CSD*), а за домен флуида други, *CFD* солвер (енг. *computational fluid dynamics* – *CFD*). Сва израчунавања потребна за оба домена су међусобно независна, али између ова два програма мора да постоји комуникација, пошто је неопходно да се омогући размена потребних информација у сваком временском кораку. Овакав приступ се може имплементирати и ако се струјање флуида симулира *LB* методом, али и другим методама, као што је на пример метода коначних елемената, што је описано у литератури [50].

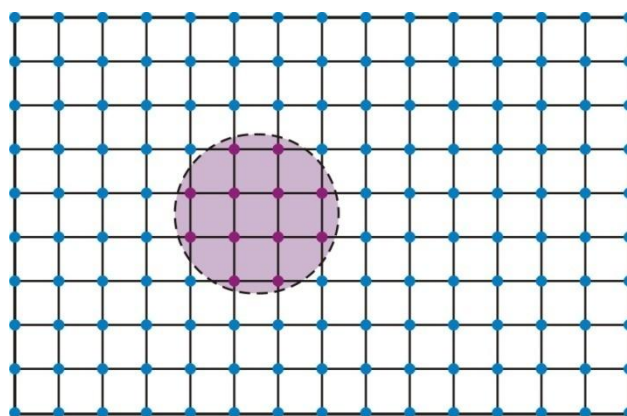


Слика 5.1 - Размена информација при решавању проблема солид-флуид интеракције применом слабог спрезања

Суштина приступа је приказана блок дијаграмом на слици 5.1. *CFD* солверу се прослеђују гранични услови за текућу геометрију додирних површина солида и флуида. Како се у овом раду разматрају честице које се налазе у флуиду, онда су у питању позиције честица. Са друге стране, *CSD* солверу се прослеђују оптерећења (у

овом случају силе) којима флуид делује на додирне површине. На тај начин се остварује интеракција између ова два солвера.

Приступ моделирању солид-флуид интеракције slabим спрезањем у LB методи се састоји у томе да се унутар домена флуида одређени чворови обележе као солид чворови и да се сматрају фиксираним. У оквиру овог рада слабо спрезање је имплементирано тако што је солид чворовима додељена *bounce-back* динамика, односно сматрано је да се на том месту налази граница под-домена. На слици 5.2 је илустрован овакав приступ. Унутар домена флуида је означен под-домен, који симулира солид.



Слика 5.2 - Слабо спрезање солида и флуида - LB метода

Честица (која представља солид) је фиксирана унутар флуида, па се најпре решава поље флуида, док се не достигне стационарно стање. На крају овог поступка могуће је одредити силе којима флуид делује на солид. Када је у питању LB метода, потребно је применити закон одржања количине кретања да би се одредила тражена сила. Наиме, за сваки чвор солида познат је вектор брзине који би се добио пропагацијом функције расподеле. Вектор брзине се може одредити применом једначине (3.41). Закон одржања количине кретања гласи:

$$\frac{d\mathbf{K}}{dt} = \mathbf{F} \quad (5.1)$$

Вектор количине кретања за чвор  $n$  у итерацији  $i$  и у итерацији  $i + 1$  се може записати на следећи начин:

$${}^i\mathbf{K}_n = m {}^i\mathbf{u}_n \quad (5.2)$$

$${}^{i+1}\mathbf{K}_n = m {}^{i+1}\mathbf{u}_n \quad (5.3)$$

Притом треба напоменути да с обзиром да се ради о чвору који се третира као *bounce-back* гранични чвор, у складу са једначином (4.1) мора да важи:

$$^{i+1}\mathbf{u}_n = -\ ^i\mathbf{u}_n \quad (5.4)$$

Сада се једначина (5.1) може трансформисати:

$$\frac{m^{i+1}\mathbf{u}_n - m^i\mathbf{u}_n}{\Delta t} = \mathbf{F}_n \quad (5.5)$$

Ако се узме у обзир да је претпоставка у LB методи да је маса честица јединична ( $m = 1$ ) и да је  $\Delta t = 1$ , добија се:

$$\mathbf{F}_n = 2\ ^i\mathbf{u}_n \quad (5.6)$$

Укупна вредност силе којом флуид (после достизања стационарног стања) делује на солид износи:

$$\mathbf{F}_R = 2 \sum_n \mathbf{u}_n \quad (5.7)$$

Ако је солид унутар флуида кружна или елиптична честица, као што је случај са примерима који су презентовани у наставку овог рада, на основу сила које су одређене за сваки чвор појединачно, може се одредити и момент којим флуид делује на солид:

$$\mathbf{M}_R = \sum_n \mathbf{r}_n \times \mathbf{F}_n \quad (5.8)$$

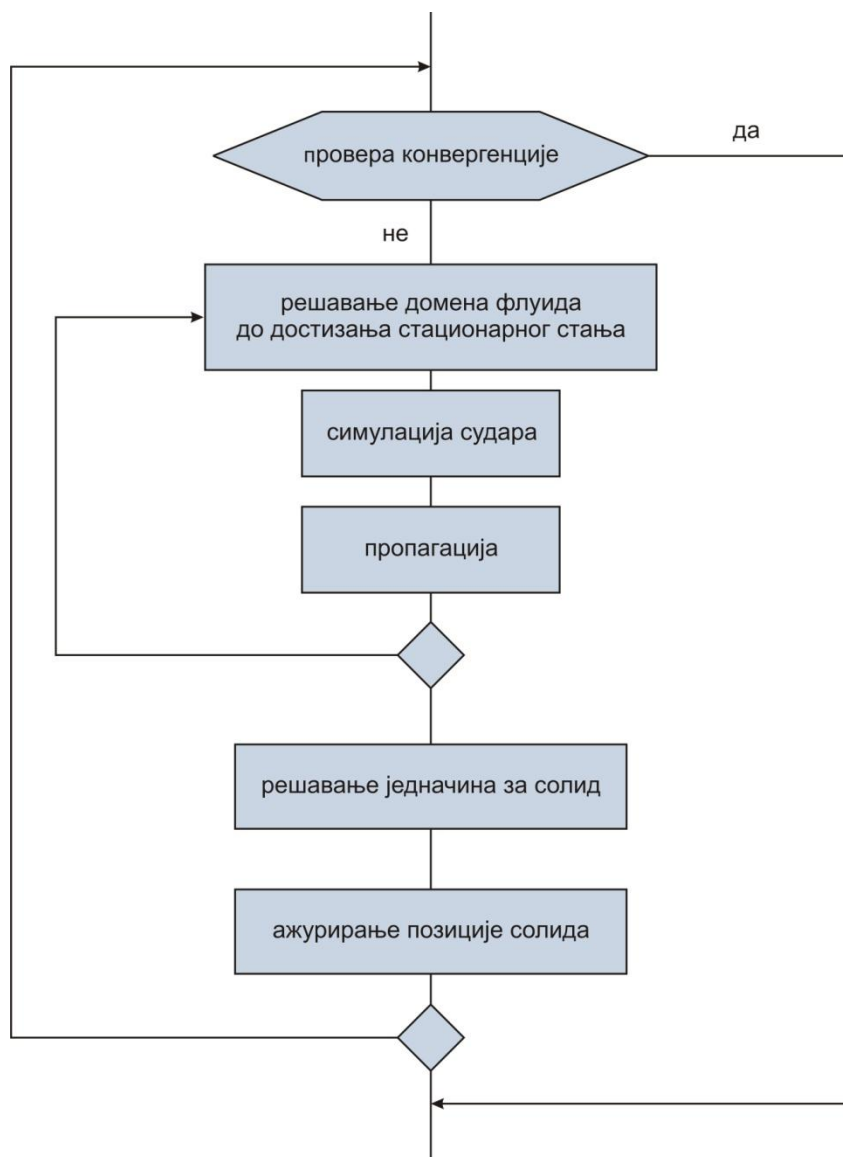
Пошто су одређени вектор силе и момента силе за солид, постављају се једначине кретања:

$$m \frac{d\mathbf{v}_C}{dt} = \mathbf{F}_R \quad (5.9)$$

$$I_0 \frac{d\boldsymbol{\omega}}{dt} = \mathbf{M}_R \quad (5.10)$$

где су са  $m$  и  $I_0$  означени маса и момент инерције честице, респективно, док је са  $\mathbf{v}_C$  означена брзина центра, а са  $\boldsymbol{\omega}$  означена угаона брзина.

Интеграцијом једначина (5.9) и (5.10) добијају се координате центра честице и угао оријентације. На основу ових података ажурира се позиција солида у домену флуида и поступак се понавља, док се не достигне коначно решење – или конвергенција поља брзина или док солид не дође до краја домена флуида, што зависи од примера до примера. На слици 5.3 је шематски приказан описани поступак.



Слика 5.3 - Алгоритам слабог спрезања (LB метода)

## 5.2 Јако спрезање

---

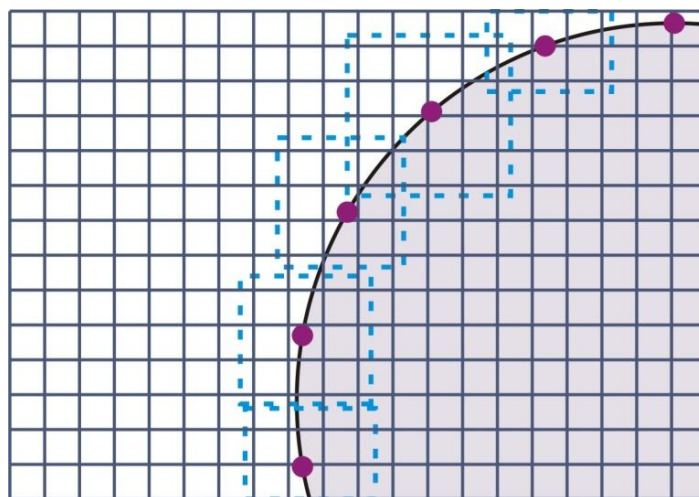
Основна идеја јаког спрезања је да се комплетан систем једначина заједно решава у сваком временском кораку. На тај начин се обезбеђује да се све величине (како за солид, тако и за флуид) мењају истовремено. Уопштено, да би се применом овог приступа решавали спрегнути проблеми, потребно је креирање специфичног солвера. Међутим, у случају симулације струјања флуида применом LB методе, постоји једна метода која даје доста добре резултате када је у питању симулација солид-флуид интеракције, конкретно проблем кретања честица кроз флуид, што је и тема овог рада [12]. То је *Immersed boundary* метода (*IBM*), коју је прво осмислио и представио *Peskin* [11]. Овај метод користи фиксану Декартову (картезијанску – енг. *cartesian*) мрежу помоћу које се представља домен флуида. Мрежа флуида се састоји од *Euler*-ових тачака. Овакав опис одговара LB репрезентацији домена флуида, па је јасно да се овакав приступ може лако применити када се струјање флуида симулира LB методом. Што се тиче солида, *IBM* представља солид као издвојен део флуида, са границом која се представља скупом *Lagrange*-ових тачака. Основна идеја је да се физичка граница између два домена третира као деформабилна граница, велике крутости [14]. Флуид делује на солид, односно на граничну површину преко силе, која тежи да деформише границу. Међутим, у самој граничној области се јавља сила која тежи да задржи солид у истом облику. Ове две силе морају да буду у равнотежи. Практично, сила којом флуид делује на солид применом закона акције и реакције делује и на флуид, који се налази у околини границе, а преноси се применом посебне дискретне делта функције. Цео солид-флуид домен се решава применом Навије-Стоксових једначина, са спољашњом силом. Постоји неколико начина да се одреди ова сила међусобног дејства солида и флуида. Неки од њих су директно задавање силе (енг. *direct forcing term*) [51,52], побољшано задавање силе [53], пеналти формулација (енг. *penalty formulation*) [12], метод размене количине кретања (енг. *momentum exchange method*) [13]. Већина ових начина има један проблем, а то је што се подразумева да се ова сила интеракције одређује унапред. На тај начин постоји ризик да не буде апсолутно задовољен услов да нема проклизавања, а у неким случајевима може се десити и да није задовољена једначина континуитета, што доводи до грешке у симулацији [54]. Да би се избегли овакви проблеми, развијен је нови начин одређивања силе интеракције солида и флуида, који су предложили *Shi* и други [55]. Овај приступ је коришћен и у овом раду приликом имплементације јаког спрезања и он ће бити детаљно описан у наставку.

За симулацију струјања флуида користи се добро познати систем једначина – Навије-Стоксова једначина и једначина континуитета (једначине (2.77) и (2.78)). Као што је већ напоменуто, ефекат солида који је уроњен у флуид се симулира преко граничне површине, односно преко спољашње запреминске силе која делује на флуид у околини граничне површине. Та сила се може изразити на следећи начин:

$$\mathbf{g}(\mathbf{x}, t) = \int_{\Gamma} \mathbf{F}(s, t) \delta(\mathbf{x} - \mathbf{X}_B(s, t)) ds \quad (5.11)$$

где су са  $\mathbf{X}_B(s, t)$  означене координате *Lagrange*-ових тачака које се налазе на граничној површини,  $\mathbf{F}(s, t)$  је сила у граничним тачкама, којом флуид делује на солид, док је  $\delta(\mathbf{x} - \mathbf{X}_B(s, t))$  *Dirac*-ова делта функција.

Гранична површина солида уроњеног у флуид се представља скупом *Lagrange*-евих тачака, са координатама  $\mathbf{X}_B^l(s, t)$  ( $l = 1, 2, \dots, m$ ), где је  $m$  број тачака на граничној површини. Све величине које су потребне за симулацију интеракције између солида и флуида се рачунају интерполацијом преко тачака на граничној површини. Пошто постоји могућност различите дискретизације домена флуида и солида, проблем интерполације са различитом дискретизацијом се решава тако што се за једну граничну тачку која припада солиду разматра утицај већег броја тачака из фиксиране мреже домена флуида, као што је приказано на слици 5.4.



Слика 5.4 - Међусобни утицај тачака домена флуида и солида

Брзина у *Lagrange*-овим тачкама се интерполира на основу околних *Euler*-ових тачака које се налазе у флуиду (илустровано на слици 5.4), што се може представити следећом једначином:

$$\frac{\partial \mathbf{X}_B(s, t)}{\partial t} = \mathbf{u}(\mathbf{X}_B(s, t), t) = \int_{\Omega} \mathbf{u}(\mathbf{x}, t) \delta(\mathbf{x} - \mathbf{X}_B(s, t)) d\mathbf{x} \quad (5.12)$$

Основна једначина LB методе (једначина (3.44)), која се користи за симулације струјања флуида остаје непромењена, без обзира што се сада не симулира само домен флуида, већ и солид уроњен у флуид:

$$\bar{f}_i(\mathbf{x} + \mathbf{v}_i, t + 1) - \bar{f}_i(\mathbf{x}, t) = -\frac{1}{\bar{\tau}} \left( \bar{f}_i(\mathbf{x}, t) - f_i^{(0)}(\rho, \mathbf{u}) \right) + \left( 1 - \frac{1}{2\bar{\tau}} \right) F_i \quad (5.13)$$

У овој једначини треба обратити пажњу да израз за силу  $F_i$  не сме да се занемари, јер се управо помоћу тог израза узима у обзир дејство солида на флуид. Да би се исправно имплементирало дејство спољашње силе у LB методи, мора се водити рачуна да остану задовољене све једначине (да и даље важи систем једначина изведен у Поглављу 2). Детаљну анализу поступка укључивања спољашње силе спровели су Guo и други [56] и на основу њихових закључака је имплементирана сила и у овом раду. Израз за силу има следећи облик (то је заправо једначина (3.20) из Поглавља 3, која је овде поновљена ради лакшег праћења извођења):

$$F_i = \omega_i \rho \left( \frac{\mathbf{v}_i - \mathbf{u}}{c_s^2} - \frac{(\mathbf{v}_i \cdot \mathbf{u}) \mathbf{v}_i}{c_s^4} \right) \cdot \mathbf{g}(\mathbf{x}, t) \quad (5.14)$$

Такође је важно нагласити да се дејство спољашње силе мора узети у обзир и приликом рачунања физичког поља брзине струјања, тако да се физичка брзина израчунава применом једначине:

$$\rho \mathbf{u} = \sum_i \mathbf{v}_i \bar{f}_i + \frac{\mathbf{g}}{2} \quad (5.15)$$

У оквиру *IBM* методе у сваком временском кораку се усклађује брзина флуида и брзина солида, тако да буде задовољен услов једнакости брзина флуида и солида на граничној површини. Приступ који је коришћен у овом раду обезбеђује једнакост брзина увођењем корекције брзине флуида у граничним тачкама [55].

Ако се са  $\mathbf{u}^*$  означи привремена брзина флуида:

$$\mathbf{u}^* = \frac{1}{\rho} \sum_i \mathbf{v}_i \bar{f}_i \quad (5.16)$$

и ако се корекција брзине флуида означи са  $\delta \mathbf{u}$ :

$$\delta \mathbf{u} = \frac{1}{2\rho} \mathbf{g}(\mathbf{x}, t) \quad (5.17)$$

тада једначина (5.15) постаје:

$$\mathbf{u} = \mathbf{u}^* + \delta \mathbf{u} \quad (5.18)$$

Корекција брзине флуида је непозната величина, која се одређује на основу корекција брзине солида у граничним тачкама:

$$\delta \mathbf{u}(\mathbf{x}, t) = \int_{\Gamma} \delta \mathbf{u}_B(\mathbf{X}_B, t) \delta(\mathbf{x} - \mathbf{X}_B(s, t)) ds \quad (5.19)$$

У овој једначини са  $\delta \mathbf{u}_B(\mathbf{X}_B, t)$  је означена корекција брзине солида у граничним тачкама.

*Dirac*-ова делта функција је апроксимирана на следећи начин:

$$\delta(\mathbf{x} - \mathbf{X}_B(s, t)) = D_{ij}(\mathbf{x}_{ij} - \mathbf{X}_B^l) = \delta(x_{ij} - X_B^l) \delta(y_{ij} - Y_B^l) \quad (5.20)$$

Вредност  $\delta(r)$  је дефинисана у литератури [11]:

$$\delta(r) = \begin{cases} \frac{1}{4h} \left( 1 + \cos\left(\frac{\pi r}{2}\right) \right), & |r| \leq 2 \\ 0, & |r| > 2 \end{cases} \quad (5.21)$$

где је са  $h$  означено растојање између 2 тачке фиксне мреже (у овом раду, пошто је у питању LB метода,  $h = 1$ ).

Ако се разматра дводимензионални проблем, користећи једначину (5.20), корекција брзина флуида (једначина 5.19) се може записати:

$$\delta \mathbf{u}(\mathbf{x}_{ij}, t) = \sum_l \delta \mathbf{u}_B^l(\mathbf{X}_B^l, t) D_{ij}(\mathbf{x}_{ij} - \mathbf{X}_B^l) \Delta s_l \quad (5.22)$$

где је са  $\Delta s_l$  означена дужина лука граничног домена (растојање између две граничне тачке).

Да би био задовољен услов да на граници између солида и флуида нема проклизавања, брзина у домену флуида у граничним тачкама мора бити једнака брзини солида у тој истој граничној тачки.

$$\mathbf{u}_B^l(\mathbf{X}_B^l, t) = \sum_{i,j} \mathbf{u}(\mathbf{x}_{ij}, t) D_{ij}(\mathbf{x}_{ij} - \mathbf{X}_B^l) \Delta x \Delta y \quad (5.23)$$

Користећи једначине (5.18) и (5.22), једначина (5.23) се може трансформисати на следећи начин:

$$\begin{aligned} \mathbf{u}_B^l(\mathbf{X}_B^l, t) &= \sum_{i,j} \mathbf{u}^*(\mathbf{x}_{ij}, t) D_{ij}(\mathbf{x}_{ij} - \mathbf{X}_B^l) \Delta x \Delta y \\ &+ \sum_{i,j} \sum_l \delta \mathbf{u}_B^l(\mathbf{X}_B^l, t) D_{ij}(\mathbf{x}_{ij} - \mathbf{X}_B^l) \Delta s_l D_{ij}(\mathbf{x}_{ij} - \mathbf{X}_B^l) \Delta x \Delta y \end{aligned} \quad (5.23)$$

Ако се уведу смене (ради поједностављивања израза):

$$\delta_{ij}^B = D_{ij}(\mathbf{x}_{ij} - \mathbf{X}_B^l) \Delta s_l \quad (5.24)$$

$$\delta_{ij} = D_{ij}(\mathbf{x}_{ij} - \mathbf{X}_B^l) \Delta x \Delta y \quad (5.25)$$

тада се једначина (5.23) може записати у матричној форми:

$$\mathbf{A} \cdot \mathbf{X} = \mathbf{B} \quad (5.26)$$

где су матрице дефинисане на следећи начин:

$$\mathbf{X} = \{\delta \mathbf{u}_B^1, \delta \mathbf{u}_B^2, \dots, \delta \mathbf{u}_B^m\}^T \quad (5.27)$$

$$\mathbf{A} = \begin{bmatrix} \delta_{11} & \delta_{12} & \dots & \delta_{1n} \\ \delta_{21} & \delta_{22} & \dots & \delta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{m1} & \delta_{m2} & \dots & \delta_{mn} \end{bmatrix} \begin{bmatrix} \delta_{11}^B & \delta_{12}^B & \dots & \delta_{1m}^B \\ \delta_{21}^B & \delta_{22}^B & \dots & \delta_{2m}^B \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{n1}^B & \delta_{n2}^B & \dots & \delta_{nm}^B \end{bmatrix} \quad (5.28)$$

$$\mathbf{B} = \begin{pmatrix} \mathbf{u}_B^1 \\ \mathbf{u}_B^2 \\ \vdots \\ \mathbf{u}_B^m \end{pmatrix} - \begin{bmatrix} \delta_{11} & \delta_{12} & \dots & \delta_{1n} \\ \delta_{21} & \delta_{22} & \dots & \delta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{m1} & \delta_{m2} & \dots & \delta_{mn} \end{bmatrix} \begin{pmatrix} \mathbf{u}_1^* \\ \mathbf{u}_2^* \\ \vdots \\ \mathbf{u}_n^* \end{pmatrix} \quad (5.29)$$

у овим изразима  $m$  представља број *Lagrange*-ових тачака, а  $n$  представља број *Euler*-ових тачака који се налазе довољно близу *Lagrange*-ових тачака тако да се могу користити у интерполацији применом *Dirac*-ове делта функције.

Решавањем система једначина (5.26) добијају се непознате корекције брзина за солид у граничним тачкама  $\delta \mathbf{u}_B^l$ . Када су ове вредности познате, лако се могу одредити корекције брзина за флуид, применом једначине (5.22), а потом и брзине флуида, применом једначине (5.18).

Помоћу прираштаја (корекција) брзина, могу се врло једноставно одредити силе којима флуид делује на солид и обрнуто. Једначина (5.17) даје везу између силе и прираштаја брзине, па пошто су брзине сада познате, може се одредити спољашња сила:

$$\mathbf{g}(x_{ij}, t) = 2\rho\delta\mathbf{u}(x_{ij}) \quad (5.30)$$

Овако израчуната спољашња сила (за сваку тачку фиксираних мреже за домен флуида) се укључује у једначину (5.14), а потом и у једначину (5.13) и на тај начин се у симулацији струјања флуида узима у обзир утицај солида.

Сила којом флуид делује на солид у једној граничној тачки може се израчунати на следећи начин:

$$\mathbf{F}(\mathbf{X}_B^l) = 2\rho\delta\mathbf{u}_B^l \quad (5.31)$$

Укупна сила дејства флуида се изражава преко силе и момента силе:

$$\mathbf{F}_R = - \sum_l \mathbf{F}(\mathbf{X}_B^l) \Delta s_l \quad (5.32)$$

$$\mathbf{M}_R = - \sum_l (\mathbf{X}_B^l - \mathbf{X}_R) \times \mathbf{F}(\mathbf{X}_B^l) \Delta s_l \quad (5.33)$$

где је са  $\mathbf{X}_R$  означен центар масе посматраног солида. Знак минус у обе једначине је последица трећег Њутновог закона (закона акције и реакције).

Уколико је потребно разматрати и утицај гравитационе силе на солид, тада се на укупну силу додаје још један члан:

$$\mathbf{F}_R = \left(1 - \frac{\rho_f}{\rho_p}\right) m \mathbf{G} - \sum_l \mathbf{F}(\mathbf{X}_B^l) \Delta s_l \quad (5.34)$$

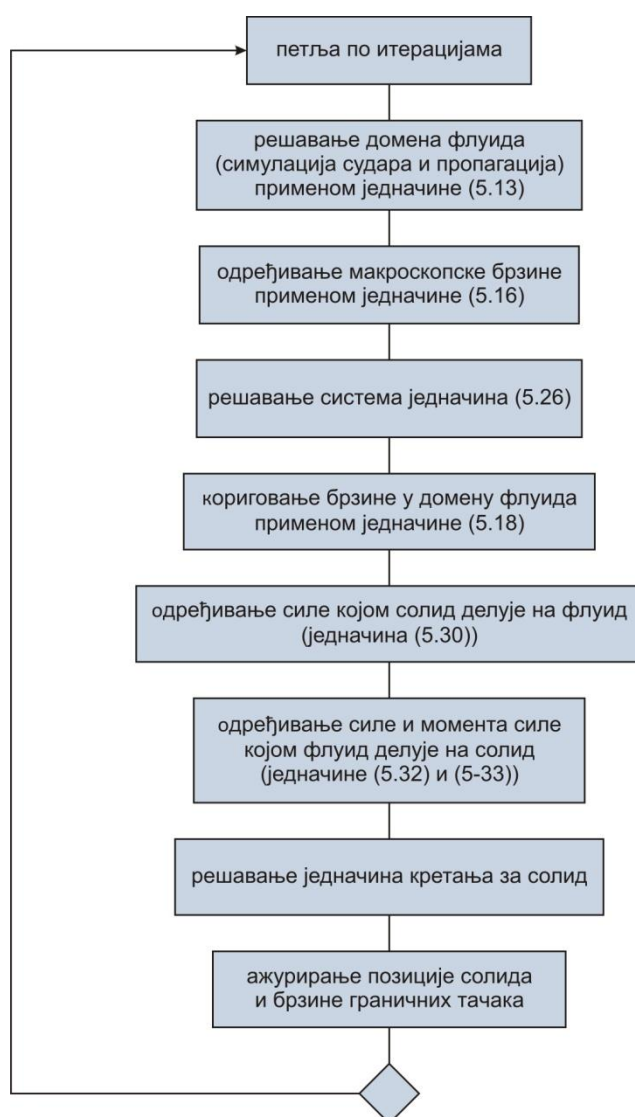
У овом изразу са  $m$  је означена маса солида, са  $\mathbf{G}$  је обележено убрзање Земљине теже, док  $\rho_f$  и  $\rho_p$  представљају густину флуида и честице, респективно.

Када су одређени сила и момент силе за солид, слично као и код слабог спрезања, могу се поставити једначине кретања за солид:

$$m \frac{dv_c}{dt} = F_R \quad (5.35)$$

$$I_0 \frac{d\omega}{dt} = M_R \quad (5.36)$$

Интеграцијом ових једначина добија се позиција и оријентација солида. На основу ових података ажурира се позиција солида у домену флуида и поступак се понавља, задати број итерација. На слици 5.5 је шематски приказан поступак јаког спрезања у солид-флуид интеракцији применом *Immersed boundary* методе.



Слика 5.5 - Алгоритам јаког спрезања (LB и *Immersed boundary* метода)

## 6 Симулације кретања честица - Примери

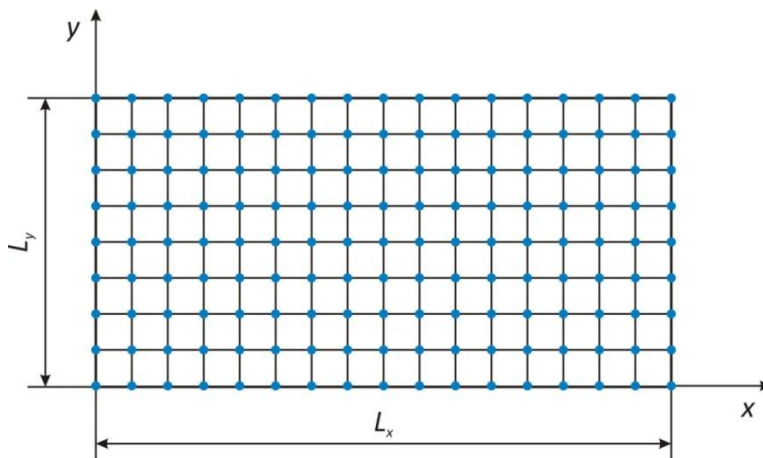
Као део истраживачког рада који је претходио овом дипломском раду, развијен је софтвер који представља нумеричку имплементацију LB методе, чија су теоријске основе дате у претходним поглављима. У овом поглављу ће бити приказани примери помоћу којих је тестиран овај софтвер. Резултати који су добијени су поређени са аналитичким решењима, или са решењима која су добијена применама других метода у литератури.

Сви примери се односе на дводимензионалне (раванске) проблеме, тако да је за прорачун коришћена мрежа са ознаком **D2Q9**, која је описивана у уводном делу Поглавља 4. За сваки пример је потребно дефинисати неколико карактеристичних величина. Најпре, потребно је дефинисати резолуцију  $N$ , која представља број тачака дуж једног правца (најчешће је то правац  $y$  осе). Домен се дефинише преко карактеристичних дужина  $l_x$  и  $l_y$ , док се „стварне“ дужине, означене са  $L_x$  и  $L_y$  изражавају на следећи начин:

$$L_x = l_x \cdot N \quad (6.1)$$

$$L_y = l_y \cdot N \quad (6.2)$$

На слици 6.1 је приказан један домен и означене су координатне осе и поменуте дужине.



Слика 6.1 - Дводимензионални домен

У LB методи се примењује један посебан принцип рачунања, тако да се све макроскопске величине које су значајне за симулацију струјања флуида дефинишу у тзв. систему јединица мреже (енг. *system of lattice units*). Тако да је најпре потребно добити вредности ових величина у бездимензионом облику, а након тога трансформисати бездимензионе величине у реалне вредности за конкретан пример.

Постоје три макроскопске величине помоћу којих се све остале величине изражавају. То су време  $t$  [s], дужина  $l$  [m] и густина  $\rho$   $\left[\frac{kg}{m^3}\right]$ . Свака друга макроскопска величина  $x$  (на пример, брзина или притисак) се изражавају у функцији ове три величине.

$$x = F(t, l, \rho) \quad (6.3)$$

Да би се извршила конверзија из система јединица мреже у бездимензионе величине, потребно је одредити фактор скалирања. Такође, вредности макроскопских величина у бездимензионом облику се помоћу другог фактора скалирања трансформишу у физичке макроскопске величине.

$$x_p = x_d \cdot F_p = x_{lb} \cdot F_D \cdot F_p \quad (6.4)$$

где су са  $x_p$ ,  $x_d$  и  $x_{lb}$  означене вредности физичке величине у стандардном систему јединица, у бездимензионом облику и у систему јединица мреже, респективно, док су са  $F_D$  и  $F_p$  означени фактори скалирања.

У склопу дефинисања карактеристичних величина, дефинише се и карактеристична брзина (у јединицама мреже – енг. *LB units*), означена са  $u_{lb}$ . Одговарајућим избором ове вредности обезбеђује се да буде задовољен услов да струјање мора бити у области малог *Mach*-овог броја, односно флуид се задржава у области мале стишљивости. То практично значи да вредност  $u_{lb}$  мора да задовољи неједнакост:

$$u_{lb} < \frac{1}{c_s} \quad (6.5)$$

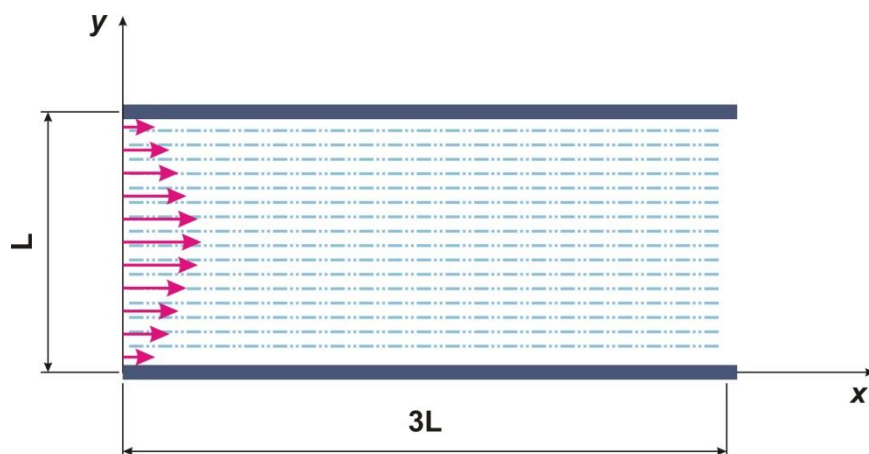
Кинематска вискозност флуида и Рејнолдсов број су повезани релацијом:

$$Re = \frac{UL}{\nu} \quad (6.6)$$

Најчешће се у симулацијама задаје Рејнолдсов број, па се применом једначине (6.6) одређује вискозност флуида у систему јединица мреже, да би се потом одредило време релаксације на основу једначине (3.39).

### 6.1 Праволинијско стационарно струјање флуида између два паралелна зида

Један од најједноставнијих примера праволинијског струјања вискозног (нестишљивог) флуида је стационарно струјање између два паралелна зида. Овај пример се користи за прву проверу тачности различитих солвера за симулацију струјања флуида. На слици 6.2 приказана је геометрија за овај проблем. Сматра се да су зидови бесконачне дужине у правцу осе  $x$ . Претпоставља се да су брзине у правцу осе  $y$  једнаке нули. Као гранични услов се задаје да је брзина на граници са зидом једнака нули (у овом случају, брзина се задаје применом регуларизованог граничног услова). Почетни услови се могу бити задати на два начина: разлика притисака на улазу и излазу или профил брзине на улазу и услов истицања флуида на излазу.



Слика 6.2 - Пример 1 - Шема геометрије струјања

Када се задаје пад притиска, користи се следећа једначина којом се изражава промена притиска дуж  $x$  осе у бездимензионом облику:

$$\frac{\partial p}{\partial x} = -\frac{8}{Re} \quad (6.7)$$

Када се задаје профил брзине (тзв. параболични профил брзине, енг. *Poiseuille flow*), користи се следећа једначина којом се изражава брзина дуж  $y$  осе у бездимензионом облику:

$$u_x(y) = 4U(y - y^2) \quad (6.8)$$

У овом случају, почетни услови су задати преко профила брзине на улазу. Такође, могуће је укључити и периодични гранични услов по  $x$  оси.

За овакво струјање се може одредити тачно аналитичко решење Навије-Стоксових једначина. Расподела брзине у правцу осе  $y$  дата је следећом једначином:

$$u = -\frac{1}{2\mu} \frac{\partial p}{\partial x} \left( \left( \frac{L}{2} \right)^2 - y^2 \right) \quad (6.9)$$

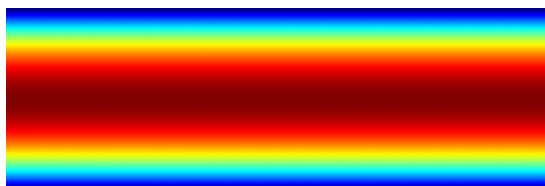
где је  $L$  растојање између зидова, а  $\frac{\partial p}{\partial x}$  је пад притиска дуж осе  $x$ .

Подаци који су дефинисани у LB солверу приказани су табеларно:

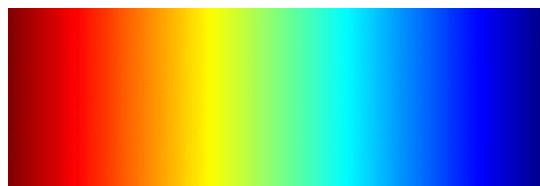
Табела 6.1. Подаци за LB солвер за пример 1

|          |      |
|----------|------|
| $N$      | 60   |
| $l_x$    | 3    |
| $l_y$    | 1    |
| $u_{lb}$ | 0.02 |
| $Re$     | 10   |

На сликама 6.3 и 6.4 приказани су поље брзине и притиска добијени прорачуном.

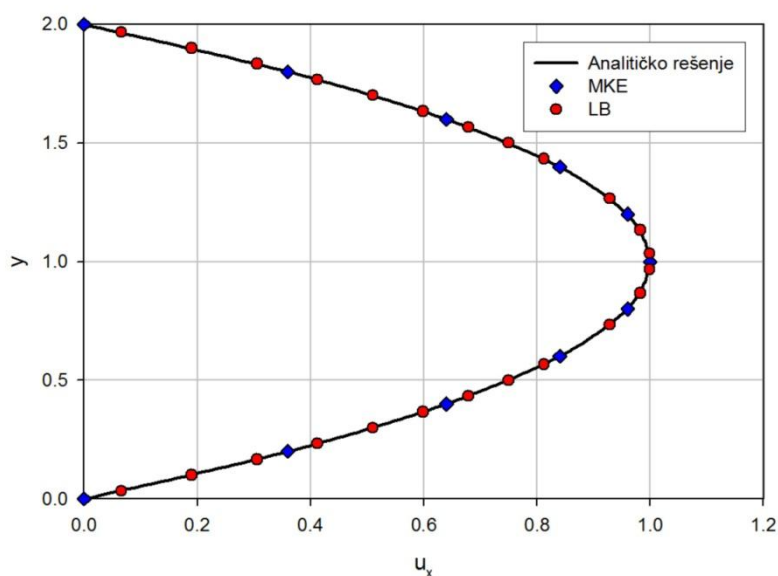


Слика 6.3 - Пример 1 - Поље брзине



Слика 6.4 - Пример 1 - Поље притиска

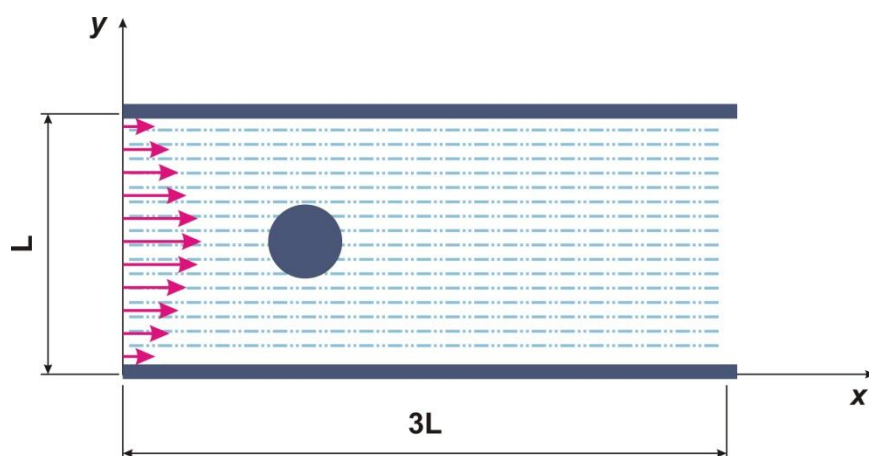
Резултати добијени за профил брзине упоређени су са аналитичким решењем, као и решењима који су добијени применом методе коначних елемената [57] и добијено је апсолутно поклапање решења, што је приказано на дијаграму 6.5.



Слика 6.5 - Пример 1 - Упоредни приказ брзине

## 6.2 Опструјавање цилиндра

Кружна честица мирује у домену флуида. Почетни и гранични услови су дефинисани као у претходном примеру. Позиција честице се дефинише координатама центра ( $x_C, y_C$ ). Геометрија струјања је приказана на слици 6.6. Потребно је одредити вучну силу којом флуид делује на честицу. Могуће је користити два приступа за моделирање солид-флуид интеракције (оба су описана у Поглављу 5). Овде су приказани резултати добијени применом слабог спрезања.



Слика 6.6 - Пример 2 - Шема геометрије струјања

Вучна сила којом флуид делује на кружну честицу приликом ламинарног струјања одређена је аналитички [58] и дефинисана је једначином:

$$F = 6\pi\mu RV \left(1 + \frac{3}{8}Re\right) \quad (6.10)$$

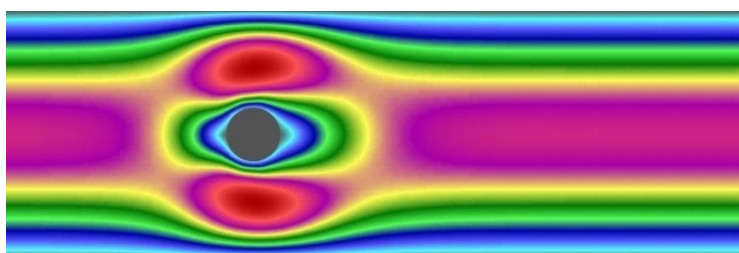
где је  $R$  полупречник честице.

Подаци који су дефинисани у LB солверу приказани су табеларно:

Табела 6.2. Подаци за LB солвер за пример 2

|          |      |
|----------|------|
| $N$      | 50   |
| $l_x$    | 3    |
| $l_y$    | 1    |
| $u_{lb}$ | 0.02 |
| $Re$     | 10   |
| $x_C$    | 25   |
| $y_C$    | 25   |

На слици 6.7 је приказано поље брзине добијено нумеричким прорачуном.



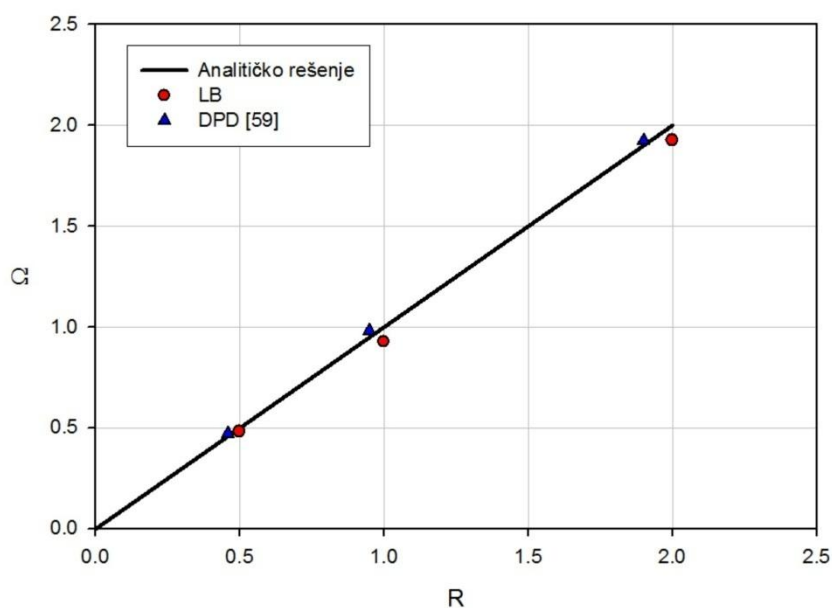
Слика 6.7 - Пример 2 - Поље брзине

Резултати који су добијени применом LB солвера су поређени са овако дефинисаним аналитичким решењем, али и са решењима добијеним применом DPD (*Dissipative Particle Dynamics*) методе [59,60]. Упоредни приказ резултата је дат на слици 6.8.

Треба обратити пажњу да зависно променљива на овом дијаграму представља следећи израз:

$$\Omega = F/6\pi\mu V \left(1 + \frac{3}{8}Re\right) \quad (6.11)$$

Овај однос је, као што се може видети из једначине (6.10) једнак полупречнику честице  $R$ .

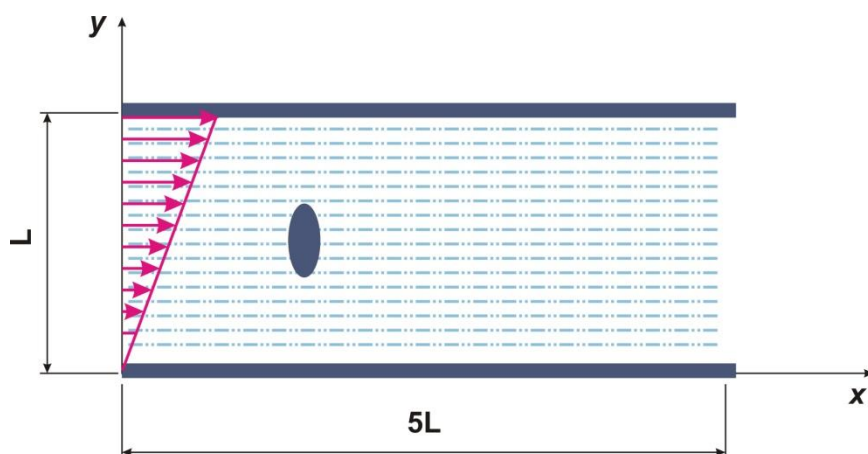


Слика 6.8 - Пример 2 - Упоредни приказ вучне силе

Као што се може видети са слике 6.8, добија се доста добро поклапање решења са аналитичким решењем, за различите вредности полупречника честице.

### 6.3 Симулација ротације елиптичне честице

Елиптична честица се налази у домену флуида. Што се тиче домена флуида, као почетни услов се задаје профил брзине на улазу, само што профил брзине више није параболичан као у претходним примерима, већ је у питању смичући профил брзине (енг. *shear flow*). Гранични услов је да је брзина доњег зида (за  $y = 0$ ), док је брзина горњег зида једнака  $u_{lb}$  (за  $y = L_y$ ). Позиција честице се дефинише координатама центра честице ( $x_C, y_C$ ), а како је у питању елиптична честица, дефинишу се и полуосе  $a$  и  $b$ . Честица је фиксирана (не може слободно да се креће), али може слободно да ротира. Солид-флуид интеракција је моделирана применом слабог спрезања. С обзиром да је за овај пример потребно решавати једначине кретања солида, приликом дефинисања величина потребних за симулацију, дефинише се и густина честице (па се на основу тога одређују маса и момент инерције). Геометрија за овај пример је приказана на слици 6.9.



Слика 6.9 - Пример 3 - Шема геометрије струјања

Однос полуоса елиптичне честице је дефинисан релацијом:

$$\gamma = \frac{b}{a} = 0.5 \quad (6.12)$$

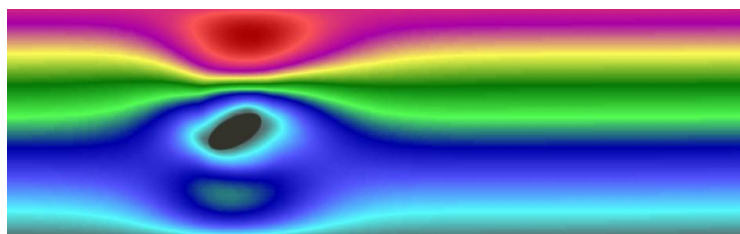
Почетни угао који већа полуоса елипсе заклапа са вертикалом је  $\frac{\pi}{2}$ .

Подаци који су дефинисани у LB солверу приказани су табеларно:

Табела 6.3. Подаци за LB солвер за пример 3

|          |             |
|----------|-------------|
| $N$      | 50          |
| $l_x$    | 5           |
| $l_y$    | 1           |
| $u_{lb}$ | 0.01        |
| $Re$     | 10          |
| $\rho_p$ | $1.25 \rho$ |
| $x_c$    | 25          |
| $y_c$    | 25          |
| $a$      | 10          |

На слици 6.10 приказано је добијено поље брзине.



Слика 6.10 - Пример 3 - Поље брзине

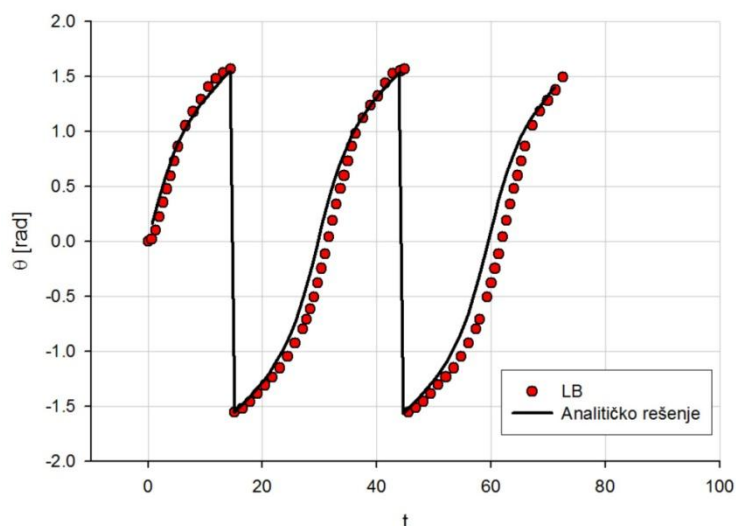
Аналитичко решење за овај проблем одредио је *Jeffery* [61]. Угаона брзина  $\omega$ , изражена у функцији угла  $\theta$  може се представити следећом једначином:

$$\omega = \frac{1}{2} \left( 1 + \frac{1 - \gamma^2}{1 + \gamma^2} \cos 2\theta \right) \quad (6.13)$$

Угао ротације се може представити једначином:

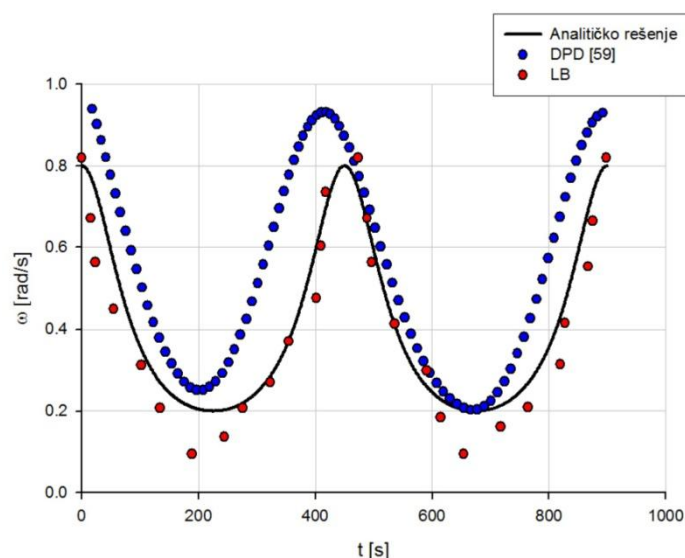
$$\theta = \arctan \left[ \frac{1}{\gamma} \tan \left( \frac{\gamma t}{1 + \gamma^2} \right) \right] \quad (6.14)$$

Промена угла  $\theta$  у зависности од времена је приказана на слици 6.11. Поређено је аналитичко решење са решењем које је добијено нумеричком симулацијом, применом LB солвера и може се видети да се добија доста добро поклапање решења.



Слика 6.11 - Пример 3 - Упоредни приказ промене угла ротације

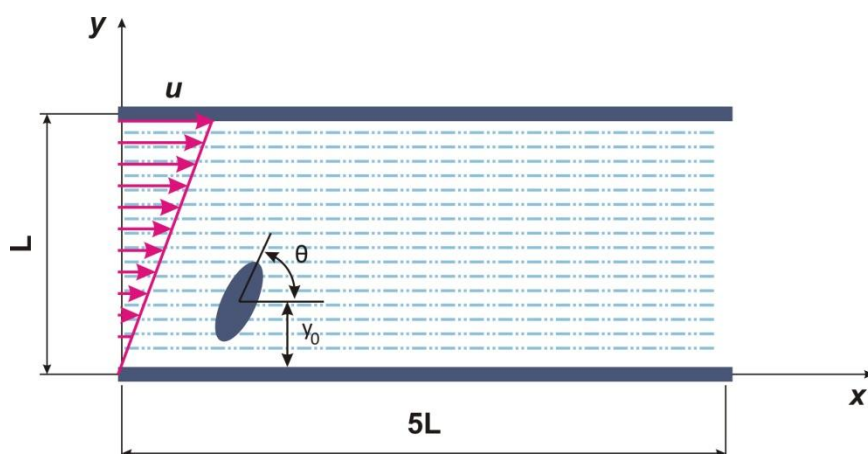
На слици 6.12 је приказана промена угаоне брзине са временом. Поређено је аналитичко решење са резултатима која су добијена применом LB солвера, као и са резултатима који су добијени применом DPD методе и преузети су из литературе [59,60].



Слика 6.12 - Пример 3 - Упоредни приказ промене угаоне брзине

#### 6.4 Симулација кретања елиптичне честице

У овом примеру разматрано је кретање елиптичне честице у флуиду. Сви параметри су потпуно исти као у примеру 3. Гранични и почетни услови су такође исти. На слици 6.13 приказана је шема геометрије. Битно је нагласити да почетни угао који већа полуоса елиптичне честице заклапа са вертикалом износи  $\theta_0 = \frac{\pi}{2}$ , а почетна позиција честице по  $y$  оси износи  $y_0 = 1.32a$ , где је  $a$  већа полуоса елиптичне честице. Солид-флуид интеракција је моделирана применом слабог спрезања.



Слика 6.13 - Пример 4 - Шема геометрије струјања

Велики утицај на начин кретања честице има Стоксов број, који се дефинише на следећи начин:

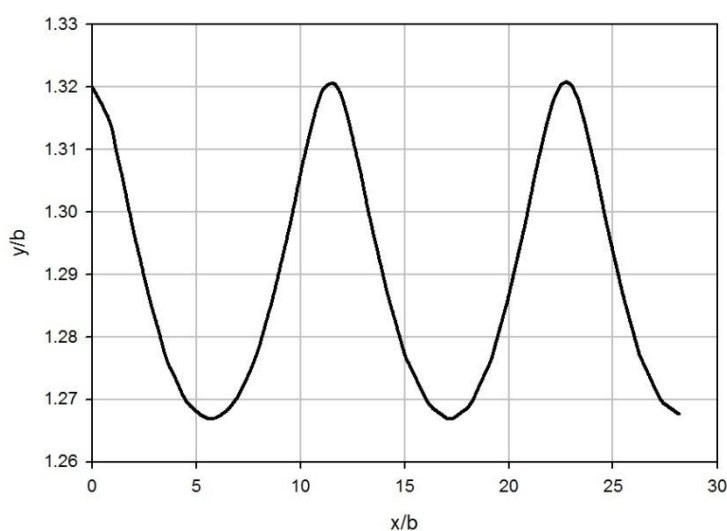
$$St = \frac{\rho_p b^2 S}{\mu} \quad (6.15)$$

где је  $\rho_p$  густина честице.

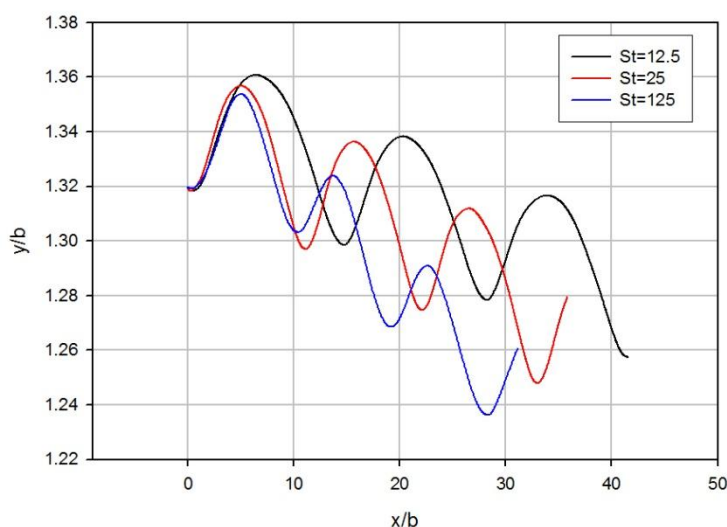
Веза између Стоксовог броја и Рејнолдсовог броја је дата једначином:

$$St = \frac{\rho_p}{\rho} Re \quad (6.16)$$

То практично значи да уколико се Рејнолдсов број и густина флуида не мењају, а густина честице се мења, мењаће се Стоксов број, а самим тим и начин кретања честице кроз флуид. Када је Стоксов број једнак 1.25 честица се креће осцилаторно, око неке константне вредности, што је приказано на слици 6.14. На слици 6.15 је приказано како изгледају путање честице за различите вредности Стоксовог броја – што је Стоксов број већи, честица брже пада ка доњем зиду.



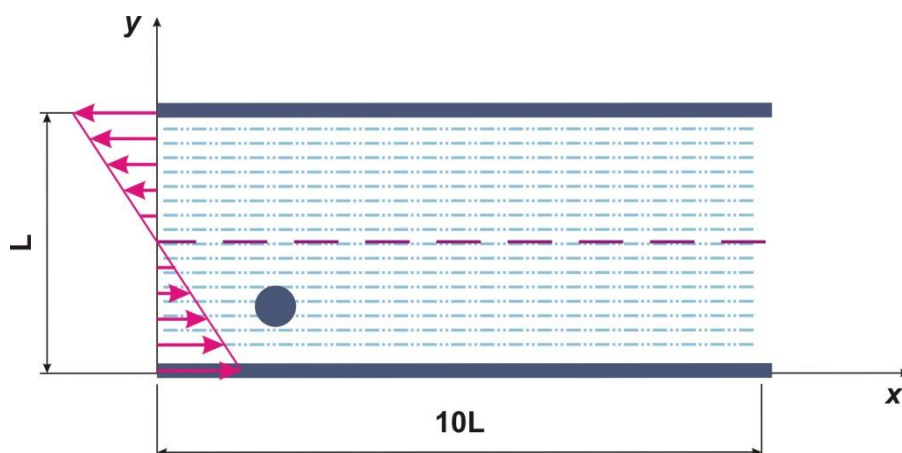
Слика 6.14 - Пример 4 - Путања елиптичне честице за  $St=1.25$



Слика 6.15 - Пример 4 - Путања елиптичне честице за различите вредности Стоксовог броја

### 6.5 Кретање сфере у флуиду са смичућим профилем брзине

Кружна честица може слободно да се креће у домену флуида. Флуид се као и у свим до сада наведеним примерима симулира применом LB методе. Као гранични услов се задаје да је брзина горњег и доњег зида једнака  $u_{lb}$ , с тим што се зидови крећу у супротним смеровима. Као почетни услов се задаје профил брзине на улазу, такозвани двоструки смичући профил брзине. Шема геометрије је приказана на слици 6.16. У овом примеру је за симулацију солид-флуид интеракције коришћен принцип јаког спрезања. У овом случају, сматра се да су густине солида и флуида једнаке. Почетна позиција честице се дефинише координатама центра  $(x_c, y_c)$ , а одређује се и полупречник честице  $r$ . Треба напоменути да је почетна позиција честице дефинисана тако да се она налази на растојању  $\frac{L}{4}$  од доњег зида. Поред стандардних величина које се дефинишу у LB симулацијама, када се користи принцип јаког спрезања, мора се дефинисати и број граничних (*Lagrange*-ових) тачака означен са  $N_L$ .



Слика 6.16 - Пример 5 - Шема геометрије струјања

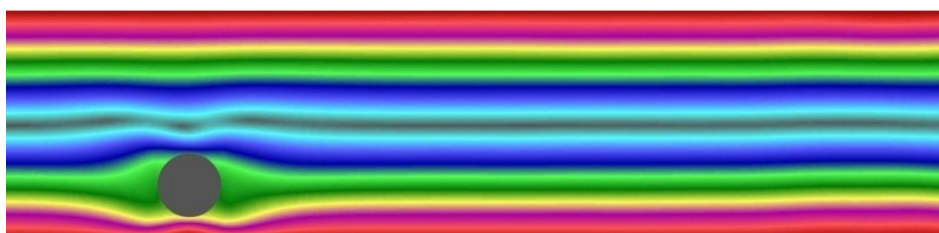
Подаци који су дефинисани у LB солверу приказани су табеларно:

Табела 6.4. Подаци за LB солвер за пример 5

|          |        |
|----------|--------|
| $N$      | 80     |
| $l_x$    | 10     |
| $l_y$    | 1      |
| $u_{lb}$ | 0.0375 |
| $Re$     | 40     |
| $x_C$    | 50     |
| $y_C$    | 20     |
| $r$      | 10     |
| $N_L$    | 50     |

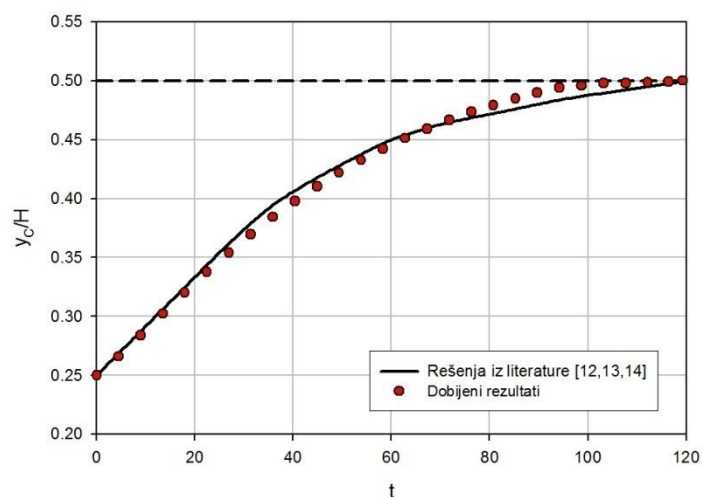
Овај проблем су прво разматрали Feng и други [62] (и симулирали кретање применом методе коначних елемената) и дошли су до закључка да без обзира како се честица постави, она увек тежи да дође до средине канала. Касније су овај исти проблем проучавали и други и симулирали применом других метода, укључујући и LB методу и различите приступе за симулирање солид-флуид интеракције. У овом раду су поређени резултати који су презентовани у литератури [12,13,14].

Слика 6.17 приказује поље брзине у току кретања честице.

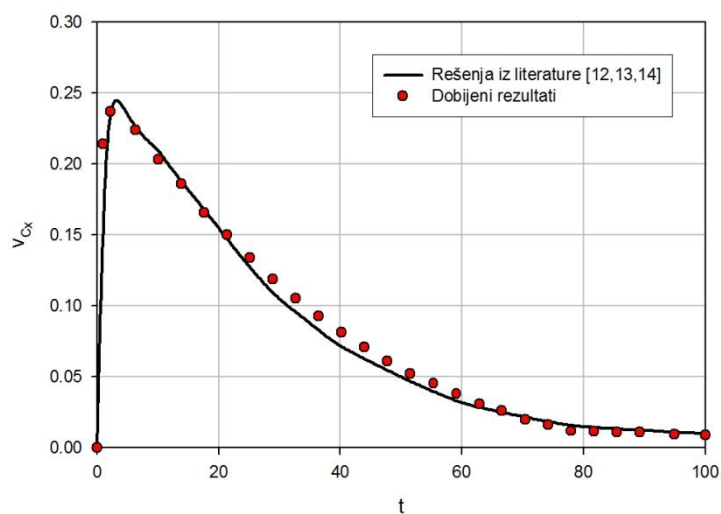


Слика 6.17 - Пример 5 - Поље брзине

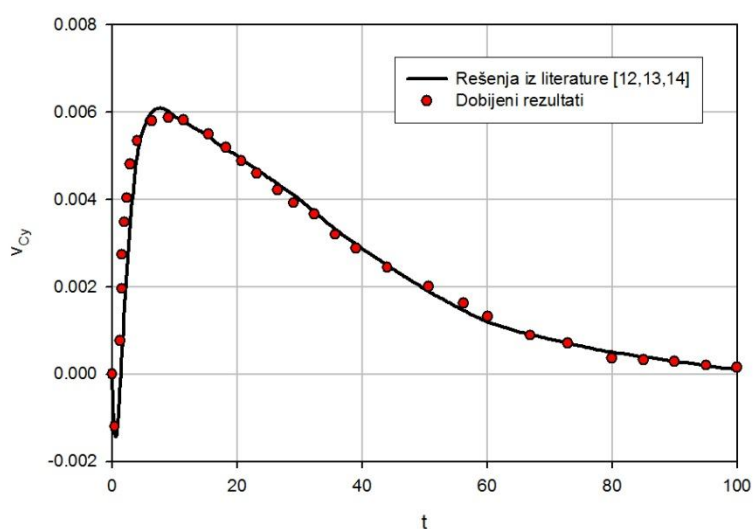
На сликама 6.18, 6.19 и 6.20 упоређени су резултата добијени применом развијеног LB солвера и резултати из литературе, када су у питању промена позиције честице у правцу у осе, брзине у правцу оса  $x$  и  $y$ . На сва три дијаграма, на обе осе су наносене бездимензионе величине добијене у симулацијама.



Слика 6.18 - Пример 5 - Поређење решења - промена позиције у правцу осе  $y$



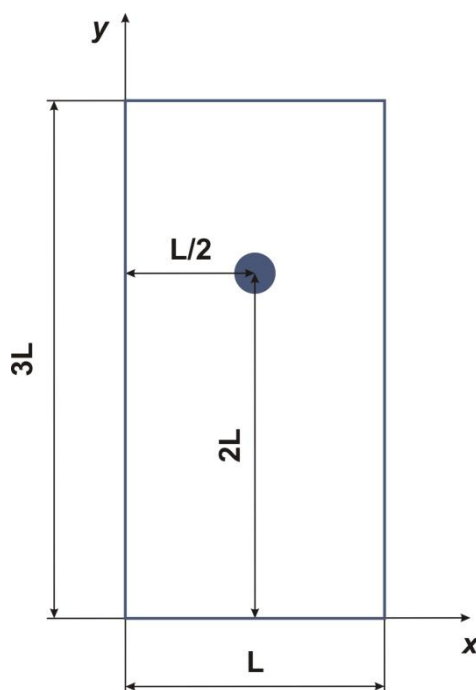
Слика 6.19 - Пример 5 - Поређење решења - Промена  $x$  компоненте брзине



Слика 6.20 - Пример 5 - Поређење решења - Промена у компоненте брзине

### 6.6 Таложење честице у вискозном флуиду

Овај пример илуструје кретање честице под дејством гравитационе силе. На почетку симулације, кружна честица и флуид су у стању мировања (почетна брзина честице, као и брзина флуида у свим чворовима мреже су једнаки нули). Честица се налази на средини домена флуида по хоризонталној оси, а на  $\frac{2}{3}$  висине по вертикалној оси (као што је приказано на слици 6.21). На честицу делује гравитациона сила, која узрокује њено кретање, односно продирање честице кроз флуид. Солид-флуид интеракција је симулирана применом јаког спрезања. Дејство гравитационе силе је имплементирано применом једначине (5.34).



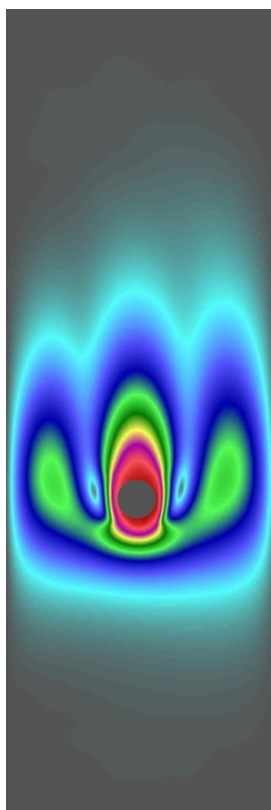
Слика 6.21 - Пример 6 - Шема геометрије струјања

И овај проблем је често разматран у литератури, тако да су сви параметри потребни за LB симулацију подешени тако да се лако могу упоредити добијени резултати са резултатима из литературе. Подаци који су дефинисани у LB солверу приказани су табеларно:

Табела 6.5. Подаци за LB солвер за пример 6

|          |             |
|----------|-------------|
| $N$      | 100         |
| $l_x$    | 3           |
| $l_y$    | 1           |
| $u_{lb}$ | 0           |
| $Re$     | 40          |
| $\rho_p$ | $1.25 \rho$ |
| $x_c$    | 50          |
| $y_c$    | 200         |
| $r$      | 6.25        |
| $N_L$    | 50          |

На слици 6.22 приказано је поље брзине флуида у току симулације (док честица продире кроз вискозни флуид).



Слика 6.22 - Пример 6 - Поље брзине

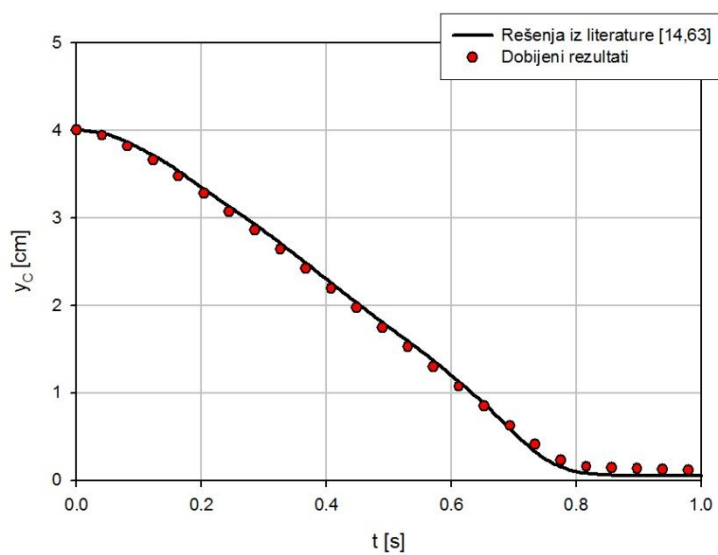
Резултати добијени нумеричким прорачуном у LB солверу су упоређени са резултатима које су објавили Wu и други [14] (прорачун је рађен применом LB методе), као и Wan и Turek [63] (прорачун је рађен применом коначних елемената). Поређена су промена четири величине – позиције честице у правцу осе  $y$ ,  $y$  компоненте брзине, Рејнолдсовог броја и кинетичке енергије. При томе се Рејнолдсов број у одређеном тренутку времена израчунава на следећи начин:

$$Re = \frac{\rho_p r \sqrt{v_{cx}^2 + v_{cy}^2}}{\mu} \quad (6.17)$$

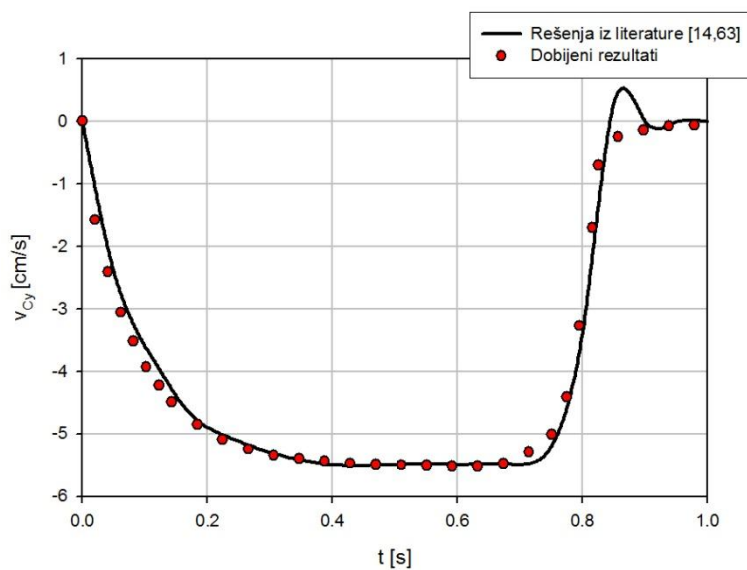
Израз за кинетичку енергију у одређеном тренутку времена гласи:

$$E_t = \frac{1}{2} m (v_{cx}^2 + v_{cy}^2) \quad (6.18)$$

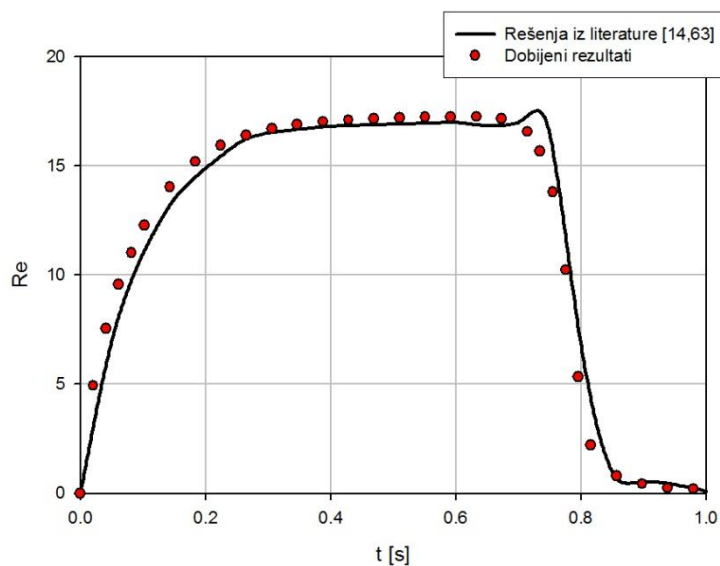
Као што се види са слика 6.23 – 6.26, добијено је добро поклапање резултата.



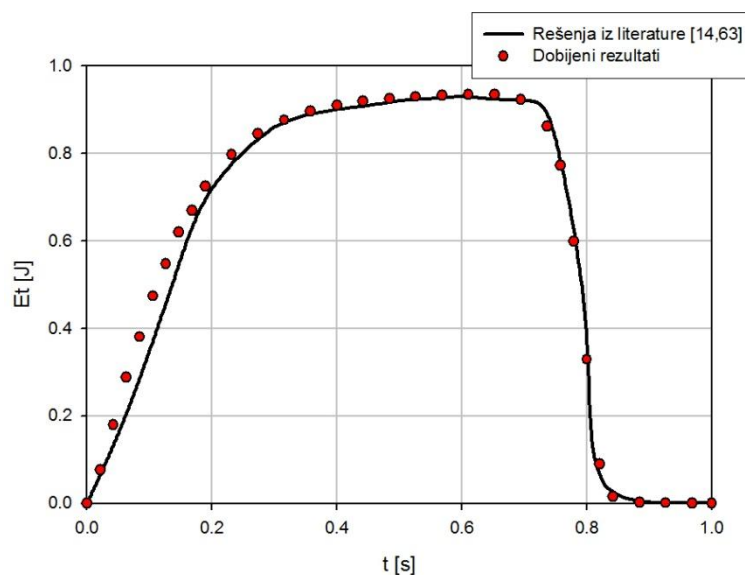
Слика 6.23 - Пример 6 - Поређење решења - промена позиције у правцу осе  $y$



Слика 6.24 - Пример 6 - Поређење решења - Промена у компоненте брзине



Слика 6.25 - Пример 6 - Поређење решења - Промена Рејнолдсовог броја

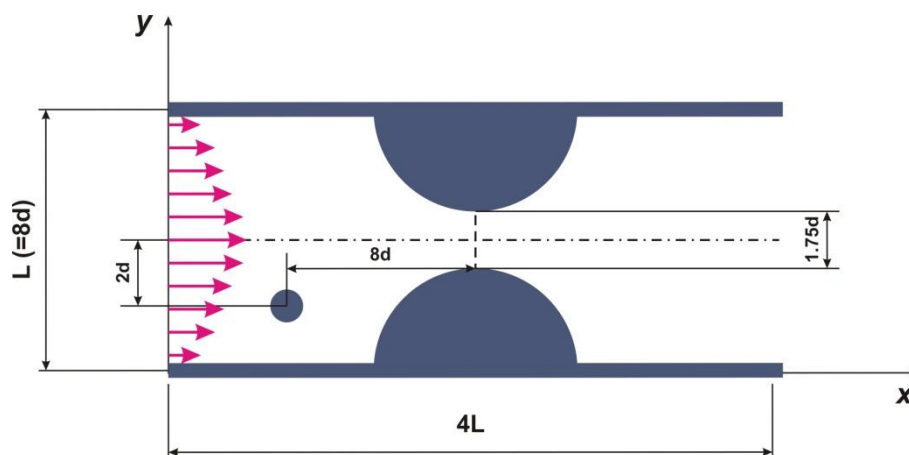


Слика 6.26 - Пример 6 - Поређење решења - Промена кинетичке енергије

### 6.7 Кретање сфере кроз артерију са стенозом

Ефикасност принципа јаког спрезања у солид-флуид интеракцији посебно долази до изражаја када домен флуида има комплексне границе. Један такав пример је моделирање кретања честице у крвном суду (артерији) са стенозом (сужењем). Li и други [64] су већ разматрали струјање флуида у артерији са већом или мањом

стенозом, а Wu и Shu су у једном од радова [14] симулирали управо кретање честица кроз овакав крвни суд. Због тога ће у овом раду бити упоређени резултати из литературе са онима који се добијају применом LB солвера.



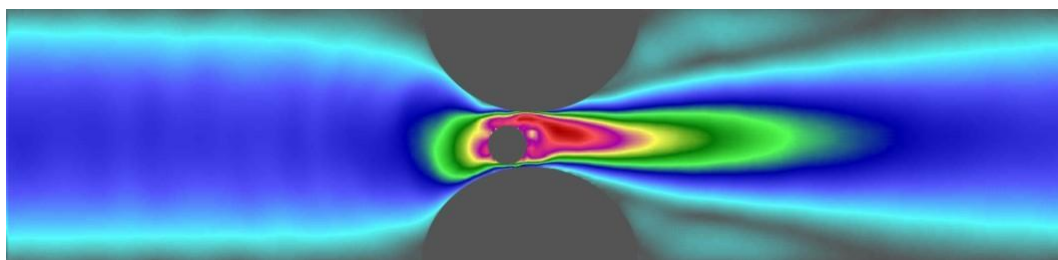
Слика 6.27 - Пример 7 - Шема геометрије струјања

Сви потребни параметри, што се тиче геометрије се могу видети на слици 6.27, а подаци који су дефинисани у LB солверу приказани су у табели 6.6. Што се тиче домена флуида, као почетни услов се задаје разлика притисака на улазу и излазу, а гранични услов је да је брзина горњег и доњег зида једнака нули. Стеноза (и уопште комплексне границе) се могу имплементирати на два начина – применом *Bounce-back* граничног услова (који је објашњен у Поглављу 4) или применом *Immersed boundary* методе (слично као што се ради и солид-флуид интеракција). Овде је коришћен први начин, што доводи до нешто мање тачности, али је нумерички ефикаснији, због мањег броја операција израчунавања.

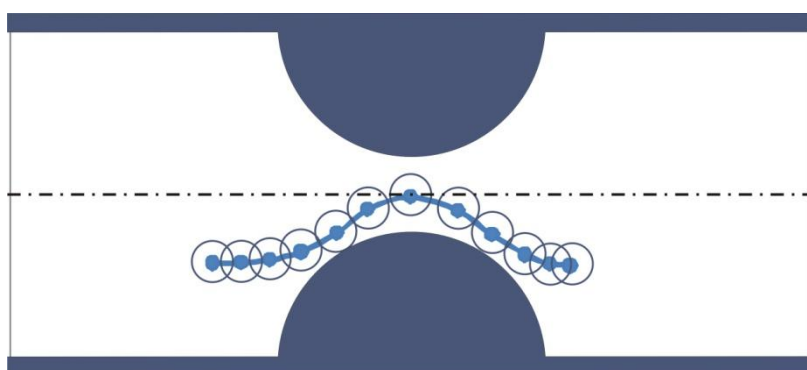
Табела 6.6. Подаци за LB солвер за пример 7

|          |             |
|----------|-------------|
| $N$      | 64          |
| $l_x$    | 4           |
| $l_y$    | 1           |
| $u_{lb}$ | 0.01        |
| $Re$     | 40          |
| $\rho_p$ | $1.25 \rho$ |
| $x_C$    | 24          |
| $y_C$    | 48          |
| $r$      | 4           |
| $N_L$    | 50          |

Поље брзина у тренутку проласка честице кроз сужење приказано је на слици 6.28, док је на слици 6.29 скицирана путања честице (у неколико корака), од почетног положаја, па све до неког крајњег положаја са друге стране, у односу на сужење.

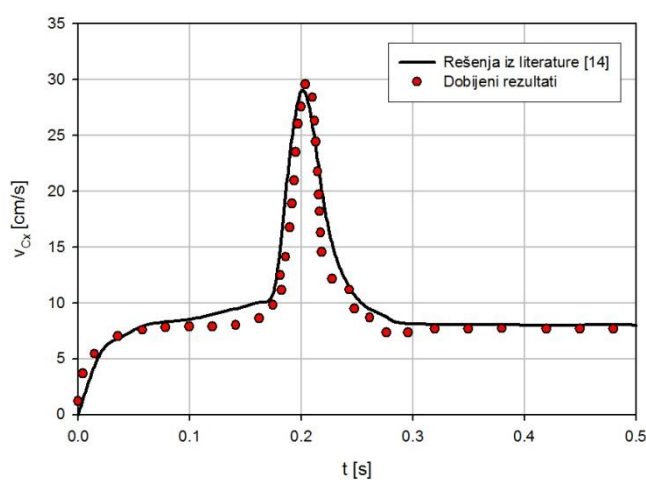


Слика 6.28 - Пример 7 - Поље брзине

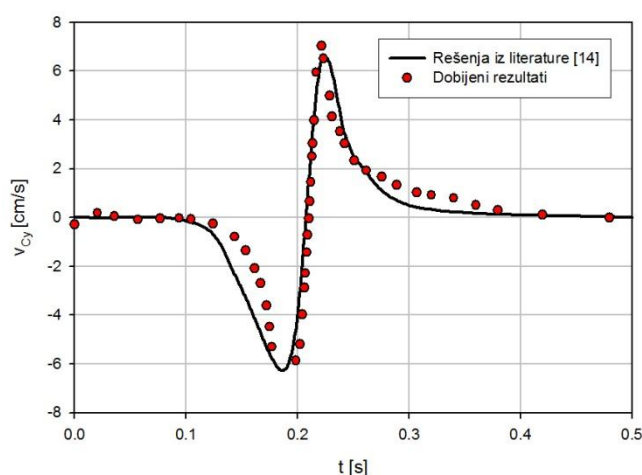


Слика 6.29 - Пример 7 - Путања честице кроз артерију са стенозом

Упоредни дијаграми  $x$  и  $y$  компоненте брзине за овај пример (решења добијена применом LB солвера и решења која су добили Wu и Shu [14]) су приказана на сликама 6.30 и 6.31.



Слика 6.30 - Пример 7 - Поређење решења - Промена  $x$  компоненте брзине



Слика 6.31 - Пример 7 - Поређење решења - Промена у компоненте брзине

### 6.8 Кретање две сфере кроз артерију са стенозом

Овај пример има потпуно исту поставку као пример 7 и урађен је по угледу на сличан пример објављен у раду [14]. Једина разлика у односу на пример 7 јесте што се уместо једне честице, моделира кретање две. Такође, потребно је имплементирати и силу међусобног дејства, којом честице делују једна на другу. У овом случају, сила којом честица  $i$  делује на честицу  $j$  дефинише се на следећи начин:

$$\mathbf{F}^{col} = \begin{cases} 0 & , X_R^{i,j} > r_i + r_j + \zeta \\ 2.4\varepsilon \cdot \left[ 2 \left( \frac{r_i + r_j}{X_R^{i,j}} \right)^{14} - \left( \frac{r_i + r_j}{X_R^{i,j}} \right)^8 \right] \cdot \frac{\mathbf{X}_R^i - \mathbf{X}_R^j}{(r_i + r_j)^2} & , X_R^{i,j} \leq r_i + r_j + \zeta \end{cases} \quad (6.19)$$

где је са  $\zeta$  означено растојање између два чвора мреже (у овом случају  $\zeta = 1$ ),  $r_i$  и  $r_j$  су полупречници честица,  $\mathbf{X}_R^i$  и  $\mathbf{X}_R^j$  су вектори положаја центара маса честица, а  $\varepsilon$  и  $X_R^{i,j}$  су дефинисани следећим једначинама:

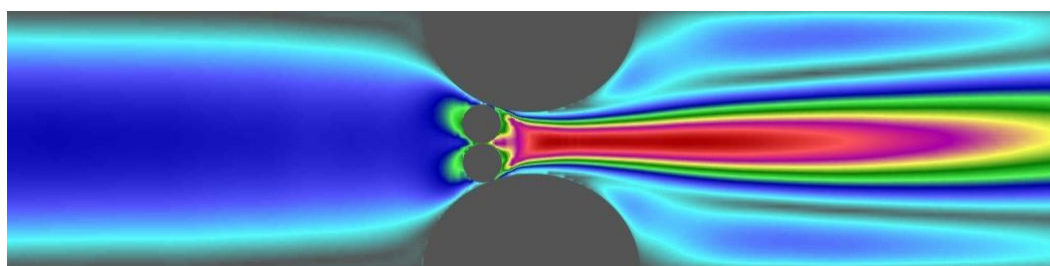
$$\varepsilon = \left( \frac{2r_i r_j}{r_i + r_j} \right)^2 \quad (6.20)$$

$$X_R^{i,j} = |\mathbf{X}_R^i - \mathbf{X}_R^j| \quad (6.21)$$

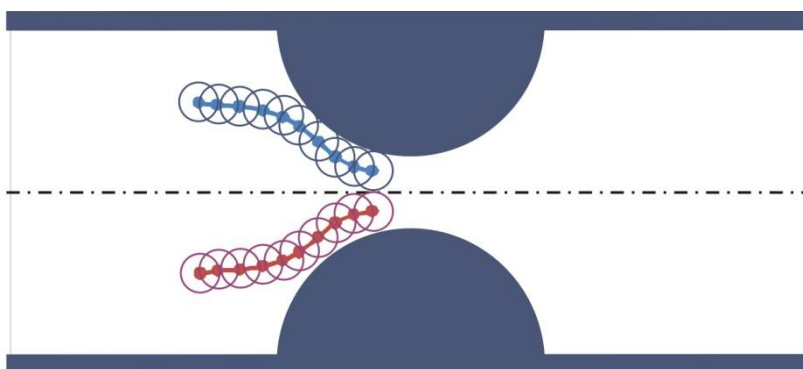
Када се израчуна вредност силе интеракције честица, она се додаје у израз за укупну силу која делује на сваку од честица појединачно (једначина 5.34) и онда се даље решавају једначине кретања. Укупна вредност силе сада гласи:

$$\mathbf{F}_R = \left(1 - \frac{\rho_f}{\rho_p}\right) m \mathbf{G} + \mathbf{F}^{col} - \sum_l \mathbf{F}(\mathbf{X}_B^l) \Delta s_l \quad (6.22)$$

С обзиром да је слободан простор артерије (који није захваћен стенозом) мањи од  $2d$  (као што је приказано на слици 6.27), очигледно је да не могу истовремено обе честице да прођу кроз сужење. Уколико се обе честице поставе у почетни положај такав да се налазе симетрично у односу на хоризонталну осу симетрије, оне ће се кретати ка сужењу, али ће остати заглављене на улазу и неће моћи даље да се крећу. На слици 6.32 је приказано поље брзине, а на слици 6.33 су скициране путање честица у том случају.

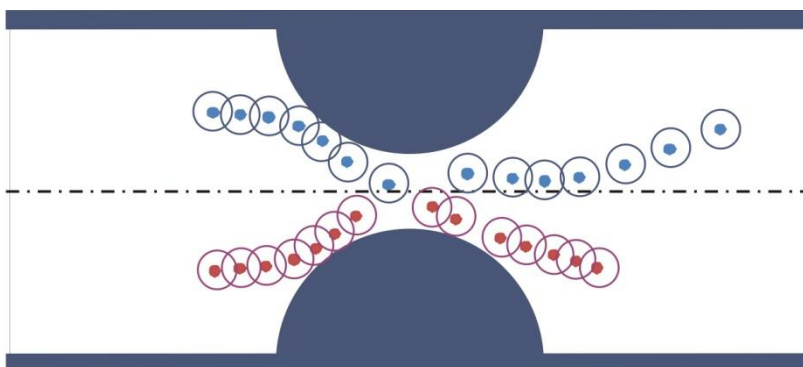


Слика 6.32 - Пример 8.1 - Поље брзине

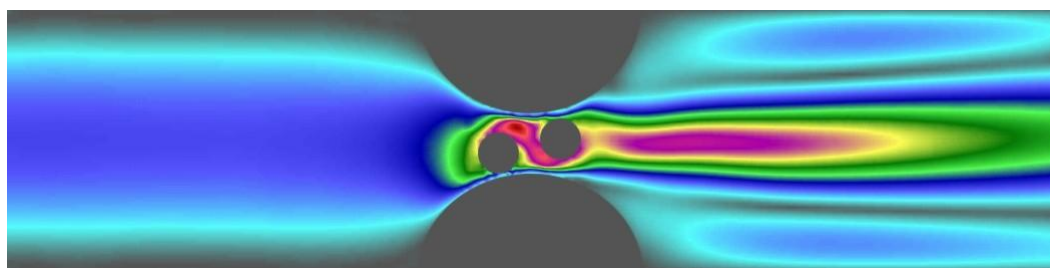


Слика 6.33 - Пример 8.1 - Путање честица

Уколико се међутим једна од честица помери за неко веома мало растојање (на пример за само 1 чвор мреже), честице ће успети да се размимоиђу у сужењу, да успешно прођу и наставе даље кретање, што је приказано шематски на слици 6.34, док је на слици 6.35 приказано поље брзине у тренутку мимоилажења.



Слика 6.34 - Пример 8.2 - Путање честица

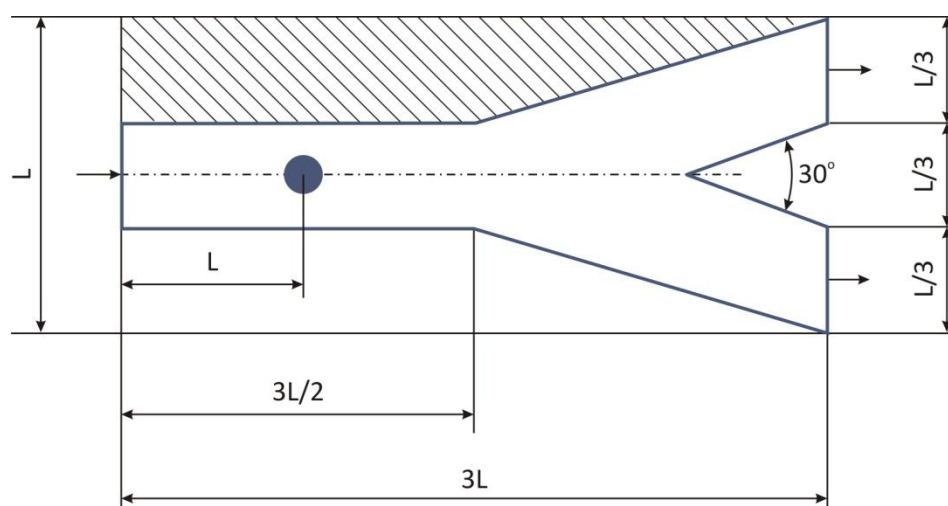


Слика 6.35 - Пример 8.2 - Поље брзине

### *6.9 Кретање сфере кроз бифуркацију*

---

Ово је још један пример кретања честица кроз домен флуида са комплексном границом. Ради се о крвном суду са бифуркацијом (рачвањем). Солид-флуид интеракција је моделирана применом принципа јаког спрезања. Геометрија домена, као и сви потребни параметри када је геометрија у питању приказани су на слици 6.36.



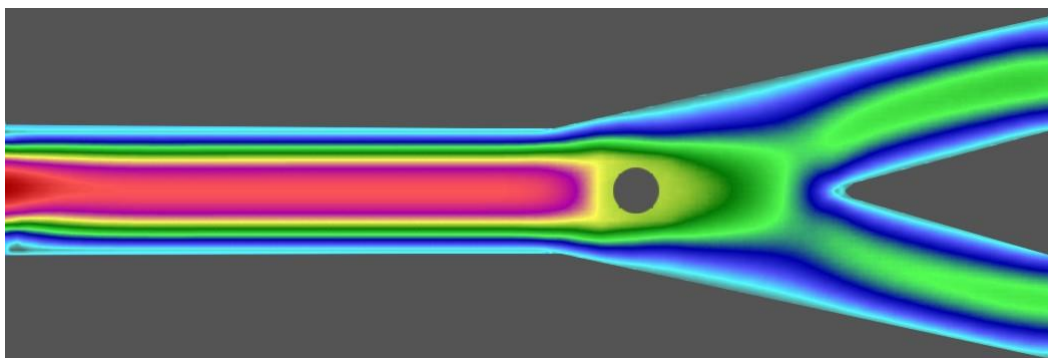
Слика 6.36 - Пример 9 - Шема геометрије струјања

Гранични услов је да брзина на свим зидовима буде једнака нули. Пошто границе домена нису правилног облика, могуће је користити два приступа за моделирање зидова - *Bounce-back* гранични услов или *Immersed boundary* метода. Овде је (као и примерима 7 и 8) коришћен први начин, због боље ефикасности, а смањење нумеричке тачности не утиче драстично на решење. Као почетни услов се користи задати профил брзине (параболични профил брзине), као и услов истицања флуида у излазним гранама. Остали подаци дефинисани у LB солверу приказани су табеларно:

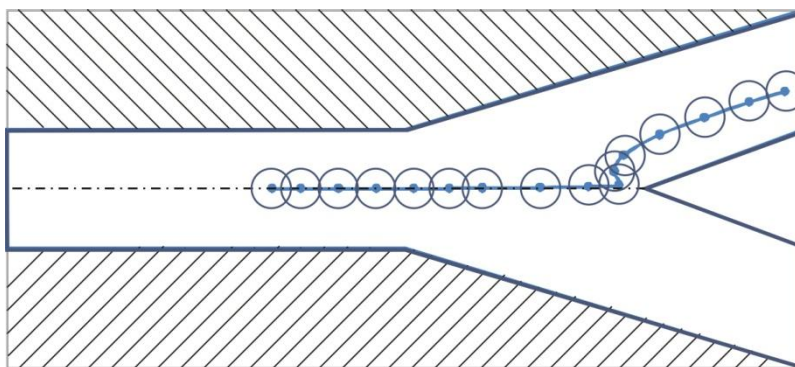
Табела 6.7. Подаци за LB солвер за пример 9

|          |             |
|----------|-------------|
| $N$      | 60          |
| $l_x$    | 3           |
| $l_y$    | 1           |
| $u_{lb}$ | 0.01        |
| $Re$     | 40          |
| $\rho_p$ | $1.25 \rho$ |
| $x_c$    | 60          |
| $y_c$    | 30          |
| $r$      | 3           |
| $N_L$    | 20          |

Када честица почне да се креће, она наилази на бифуркацију, на њу делује јака одбојна сила, она се одбија од зида и наставља да се креће горњом граном, до краја домена. На слици 6.37 приказано је поље брзине у току кретања честице кроз главни део крвног суда, док је на слици 6.38 шематски приказана путања честице.



Слика 6.37 - Пример 9 - Поље брзине



Слика 6.38 - Пример 9 - Путaња честице

## 7 Паралелизација LB солвера

---

Оперативни систем савременог персоналног (енг. *Central Processing Unit - CPU*) рачунара извршава неколико задатака у исто време. Док се рачунар користи, оперативни систем прати све активности, комуницира са екстерним уређајима у позадини и прослеђује информације док корисник ради истовремено у неколико програма. Иако процесор рачунара ради на неколико независних задатака, брзина процесора је довољно велика, тако да изгледа као да се сви ови задаци извршавају истовремено. Стандардна архитектура рачунарског процесора је таква да рачунар може извршавати одређени број задатака истовремено. Ипак, ефикасност рада се значајно смањује када треба паралелно анализирати проблем који је подељен на одређени број подскупова. Ово је управо тип проблема који решавају GPU уређаји (енг. *Graphics Processing Unit*).

GPU уређаји су посебно прилагођени решавању проблема који се могу другачије описати као процесирање огромне количине аритметичких операција над неким (такође великим) скупом података. Другим речима, када треба исти низ операција извршити на великом броју елемената, паралелно, са малим бројем меморијских операција и великим бројем аритметичких операција – пожељно је користити GPU уређаје.

Програм који обавља процесирање оваквог скупа података би се на стандардном рачунару извршавао секвенцијално, док се на GPU уређају подаци преносе у меморију уређаја и деле се на хиљаде нити (енг. *threads*) које се паралелно процесирају. Апликације у којима се користе дугачки низови или матрице великих димензија су згодне за паралелизацију применом овог принципа и у таквим случајевима се израчунавања драстично убрзавају.

У овом раду су поменути принципи GPU програмирања примењени на симулацију струјања флуида, конкретно на имплементацију *lattice Boltzmann* методе. Ова метода је иначе јако погодна за паралелизацију, због своје природе и теоријских принципа на којима је изведена, што је већ обрађено у Поглављима 2 и 3. Развијени LB солвер, помоћу кога су добијени сви резултати приказани у Поглављу 6, је паралелизован и сам поступак паралелизације и добијено убрзање ће бити тема овог Поглавља.

### 7.1 Програмирање GPU уређаја

---

Основни проблем који је постојао у раним фазама коришћења GPU уређаја је био што су ови уређаји били прилагођени за рад искључиво са графичким апликацијама. То је практично значило да су програмери своје апликације морали да прилагођавају раду са троугловима и полигонима, што је понекад било јако компликовано. Али, касније су се појавиле посебне софтверске архитектуре, као што је CUDA (енг. *Compute Unified Device Architecture*) [65], које омогућавају далеко лакше програмирање GPU уређаја уз помоћ програмских језика вишег нивоа, као што су C и C++.

NVIDIA је представила CUDA архитектуру још у новембру 2006. године, као програмски језик за програмирање њихових GPU уређаја. CUDA заправо представља софтвер потребан за програмирање хардвера – GPU-а. Од 2006. године архитектура је неколико пута унапређивана и постоји неколико верзија. За паралелизацију у овом раду коришћена је верзија 4.0.

CUDA заправо проширује програмски језик C, са одређеним скупом нових резервисаних речи помоћу којих се GPU уређају задају команде. CUDA програмирање омогућава покретање кода на две различите платформе, на тзв. *host* систему (који представља CPU процесор) и на тзв. *device* систему (који се састоји од једног или више GPU уређаја). Велика предност ове архитектуре је и то што се програмирање може вршити у *Microsoft Visual Studio* окружењу, као да су у питању стандардне апликације.

Задатак програмера је да измени постојећи програм и да раздвоји делове програма који имају велики број израчунавања од остатка апликације и да у том делу програма податке проследи GPU уређају. Остатак програма се као и раније извршава на CPU. Приликом извршавања паралелизоване верзије програма, GPU ради као копроцесор главног процесора (CPU-а). Велики скупови података (над којима се врше аритметичке операције) се прослеђује GPU уређају, а CPU организује и предаје потребне информације.

Да би се извршио део програма на GPU уређају, програмер мора да дефинише специјалне функције, који се називају кернели (енг. *kernels*). Свака оваква функција се позива из главног дела програма и извршава се на мрежи нити (енг. *grid of threads*). Приликом сваког позива кернел функције, потребно је дефинисати број нити и димензију мреже и наравно проследити одговарајуће аргументе.

## 7.2 NVIDIA Tesla

Већина савремених графичких картица, који су део конфигурације данашњих персоналних рачунара, могу се користити као GPU уређај. Ипак, NVIDIA у својој понуди има и специјализоване GPU уређаје – Tesla серију уређаја. Они су оптимизовани за извршавање комплексних научних симулација. Основна разлика између класичне графичке картице и Tesla уређаја је у томе што Tesla нема могућност приказа слике на екрану. За паралелизацију LB солвера и тестирање брзине рада, која је део овог рада, коришћен је рачунар у који су уграђена 3 GPU уређаја Tesla C1060, чије су карактеристике приказане у табели 7.1.

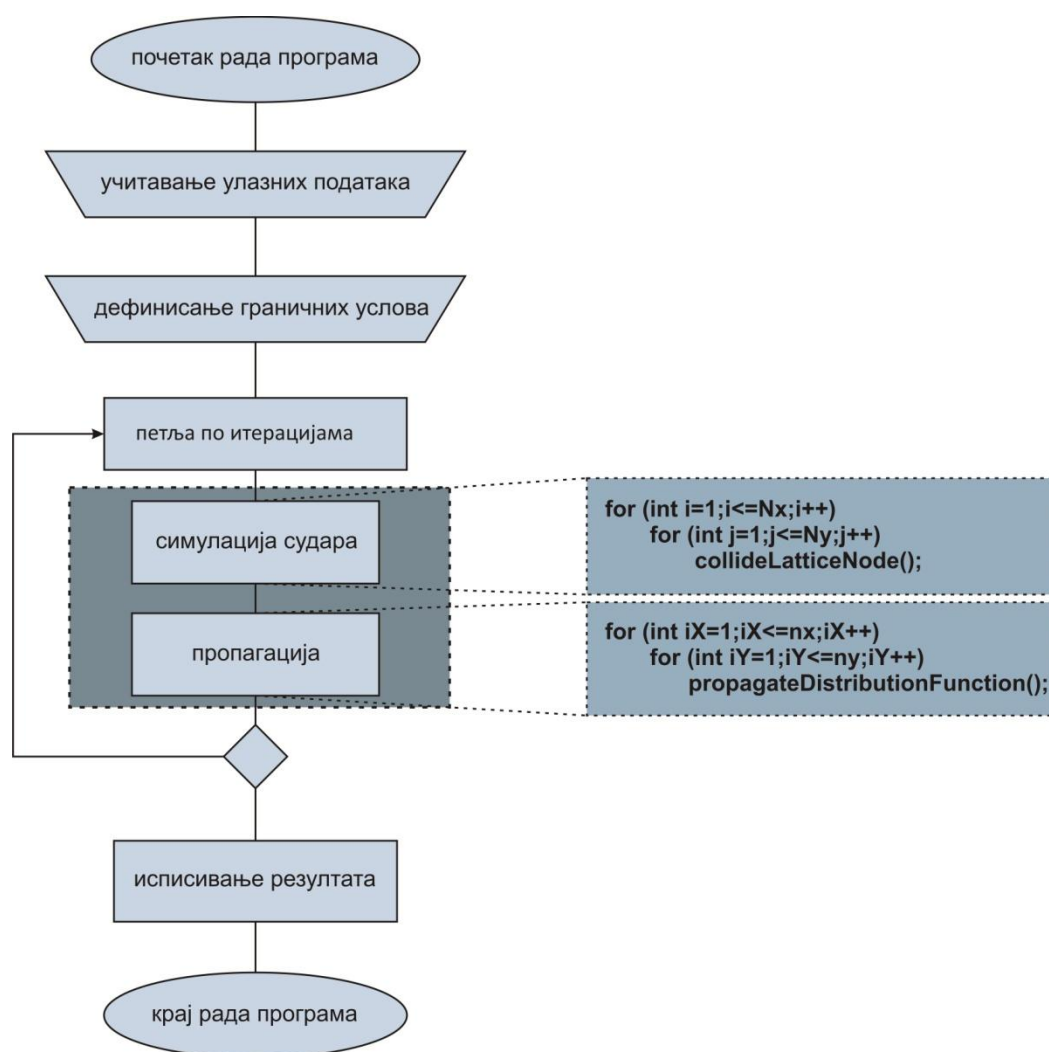
Табела 7.1. Карактеристике NVIDIA Tesla C1060

|                                                                   |                   |
|-------------------------------------------------------------------|-------------------|
| CUDA верзија уређаја<br>(енг. <i>compute capability</i> )         | 1.3               |
| Укупна расположива глобална меморија                              | 4096 MB           |
| Број мултипроцесора                                               | 30                |
| Број CUDA језгара (8 по мултипроцесору)                           | 240               |
| Радни такт GPU уређаја                                            | 1.30 GHz          |
| Такт меморије                                                     | 800 MHz           |
| Укупна локална (регистарска) меморија на располагању једног блока | 16384 bytes       |
| Максималан број нити у једном блоку                               | 512               |
| Максималне димензије блока                                        | 512 x 512 x 64    |
| Максималне димензије мреже блокова                                | 65535 x 65535 x 1 |

## 7.3 Паралелизација софтвера на бази *lattice Boltzmann* методе

Имплементација *lattice Boltzmann* методе подразумева да се у одређеном броју итерација (или док се не испуни неки услов) понављају наизменично корак судара и корак пропагације (дефинисани једначинама (3.45) и (3.46)). Као што је већ наведено у

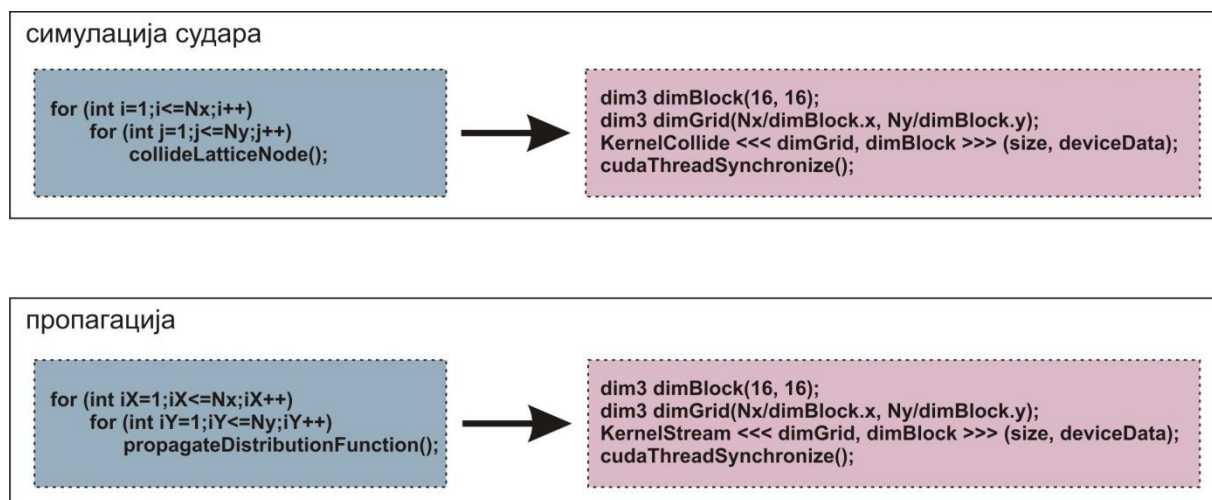
Поглављу 3, сваки од ова два корака се мора применити на све чворове мреже, пре него што почне следећи корак. Како су једначине корака судара међусобно независне, овај корак је потпуно локалног карактера. У оквиру пропагације узима у обзир само тренутно посматрани чвор и неколико најближих суседних чворова. Све ово указује на то да постоји велики скуп података, који се може поделити на мање подскупове и на њима треба извршити одређени број (углавном аритметичких) операција. При томе су ти подскупови међусобно независни и операције се могу извршавати паралелно. Управо због тога је LB метод јако погодан за паралелизацију. На слици 7.1 је приказан алгоритам рада класичног (серијског) LB солвера, са означеним делом који се може паралелизовати. Такође су наведени делови кода помоћу којих се имплементирају ова два корака.



Слика 7.1 - Алгоритам рада класичног LB солвера

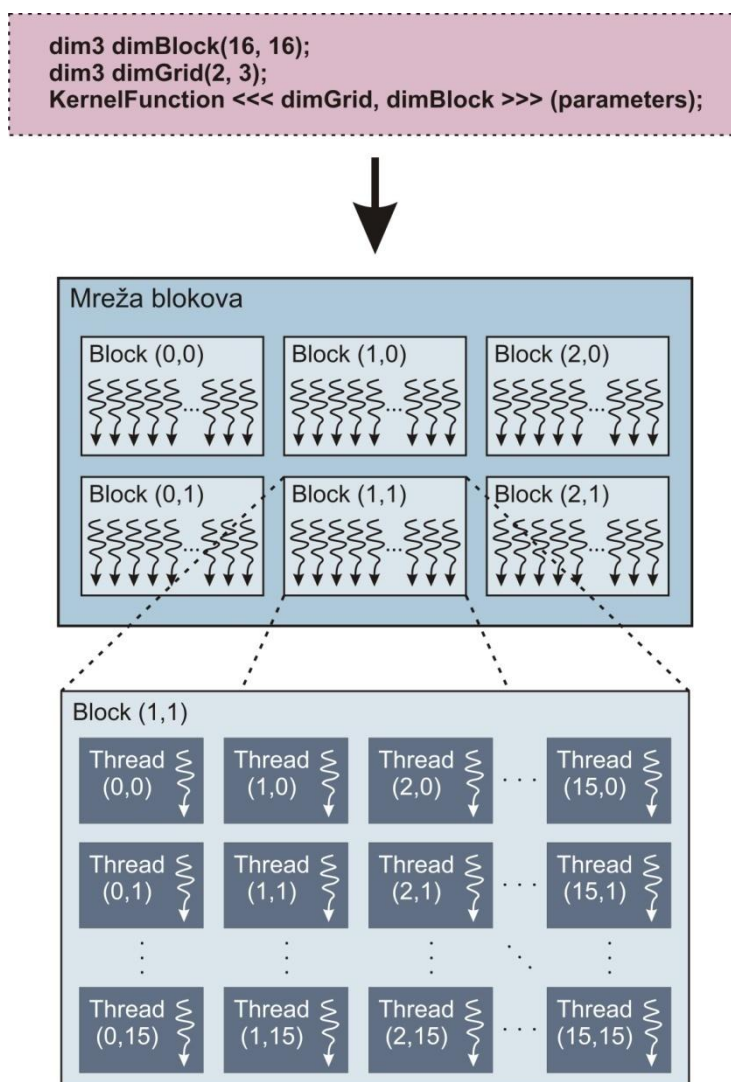
### 7.3.1 Позивање и извршавање кернел функција

С обзиром да је већ изведен закључак да су симулација судара и пропација локалног карактера и да их треба паралелизовати, потребно је испрограмирати две кернел функције које ће се позивати уместо стандардних функција које се користе у класичној верзији LB солвера. На слици 7.2 је приказано како изгледају позиви ових функција у класичној и паралелној верзији.



Слика 7.2 - Симулација судара и пропација - класична и паралелна верзија

Када се позове кернел функција, она на  $N$  различитих нити извршава паралелно подпрограм који је написан у оквиру тела кернел функције. Због тога је потребно дефинисати како су те нити распоређене. GPU уређаји имају такву хардверску архитектуру да садрже одређени број тзв. мултипроцесора. Један мултипроцесор има 8 CUDA језгара на којима се извршавају кернели. Пре покретања кернела мора се дефинисати распоред нити, да би GPU уређај могао да одреди колико ресурса треба да ангажује. Променљивом `dimGrid` (видети слику 7.2) дефинише се мрежа блокова, а променљивом `dimBlock` дефинише се број нити у оквиру сваког блока. Наравно, постоје извесна ограничења када су у питању димензије мреже блокова и самог блока и то зависи од самог GPU уређаја. Када је у питању Tesla C1060 која је коришћена за паралелизацију LB солвера, блок може имати највише 512 нити, а максимална димензија мреже блокова је 65.535 нити. Више детаља се може видети у табели 7.1. На слици 7.3 илустрован је изглед мреже блокова и нити у оквиру једног блока, за један позив кернел функције.



Слика 7.3 - Илустрација мреже блокова и нити

Да би се унутар кернел функције приступало елементима низа или матрице, потребно је знати индекс сваке нити. Свака нит приликом покретања кернел функције добија свој дефинисани идентификациони број (ID) и на основу њега се одређује тачно место те нити у низу или матрици. Сваки блок има свој ID број којим је одређена позиција тог блока у мрежи блокова и постоји системски дефинисана променљива ***blockIdx*** која памти овај број. Системски дефинисана променљива ***threadIdx*** памти позицију нити у оквиру одговарајућег блока. Помоћу ове две променљиве одређује се глобални индекс нити и помоћу њега се приступа одговарајућем елементу матрице (или низа).

### 7.3.1 Манипулација меморијским ресурсима

Када се испрограмирају два кернела потребна за симулацију судара и пропагацију, они се позивају уместо еквивалентних класичних (серијских) функција. Међутим, постоји још једна важна ставка о којој се мора водити рачуна када се врши паралелизација постојећих програма, а то је манипулација меморијом. Наиме, CUDA архитектура је тако конципирана да CPU (другачије назван *host*, у преводу „домаћин“) и GPU (другачије назван *device*, у преводу „уређај“) имају одвојене DRAM меморије (што у суштини и јесте случај, ако се посматра са хардверске стране), које се називају *host* и *device* меморија, респективно. CUDA омогућава програму да приступа овим меморијама независно, али при томе постоје одређена ограничења. Део програма који се извршава на главном процесору (CPU) може користити само *host* меморију, док кернел функције могу приступити само *device* меморији. Због тога је неопходно имплементирати експлицитно алоцирање и деалоцирање меморије за оба система, као и трансфер података између ова два типа меморије. Постоје посебне CUDA функције које обезбеђују пренос података са CPU на GPU и обрнуто.

Такође, *device* меморија је организована у једну хијерархијску структуру, тако да нити могу да приступају подацима из меморије на више нивоа. Свака нит има своју приватну локалну меморију, којој само та нит има приступ. Сваки блок нити има своју дељиву меморију (енг. *shared memory*) која је доступна свим нитима у оквиру тог блока. И на крају, све нити имају приступ глобалној меморији уређаја.

Најједноставнији начин паралелизације LB солвера би био да се сви потребни подаци о чворовима (на првом месту подаци о функцији расподеле) чувају у меморији главног рачунара (у *host* меморији), а да се пре сваког позива кернел функције изврши пренос података у *device* меморију, па да се ти подаци поново врате назад после завршетка рада кернел функције. Међутим, овај приступ је потпуно погрешан, због тога што је трансфер података између ова два типа меморије изузетно „скупа“ операција. Максимални пропусни опсег *device* меморије која се користи са самог GPU уређаја износи 102.4 GBps (енг. *Gigabytes per second*) када је у питању Tesla C1060, док пропусни опсег трансфера између *host* и *device* меморије износи око 8 GBps [66]. То значи да је потребно минимизовати трансфере између ова два типа меморије кад год је то могуће.

У овом конкретном случају, то значи да се подаци о чворовима (функције расподеле и све остало) иницијализују на CPU, потом се ти подаци само једном прекопирају у *device* меморију и током извршавања програма се више не преносе. Кернел функције

приступају подацима у *device* меморији и мењају их по потреби. За приказивање резултата (креирање поља брзине и притиска) се евентуално назад у *host* меморију прослеђују само макроскопске величине, које се рачунају помоћу посебне кернел функције. Треба имати у виду да се најчешће поље брине и притиска не приказују за сваку итерацију LB солвера, већ на 100, 1000 или више итерација, зависи од самог проблема који се симулира. На овај начин је драстично смањена количина података која се преноси између меморија, што додатно убрзава рад програма.

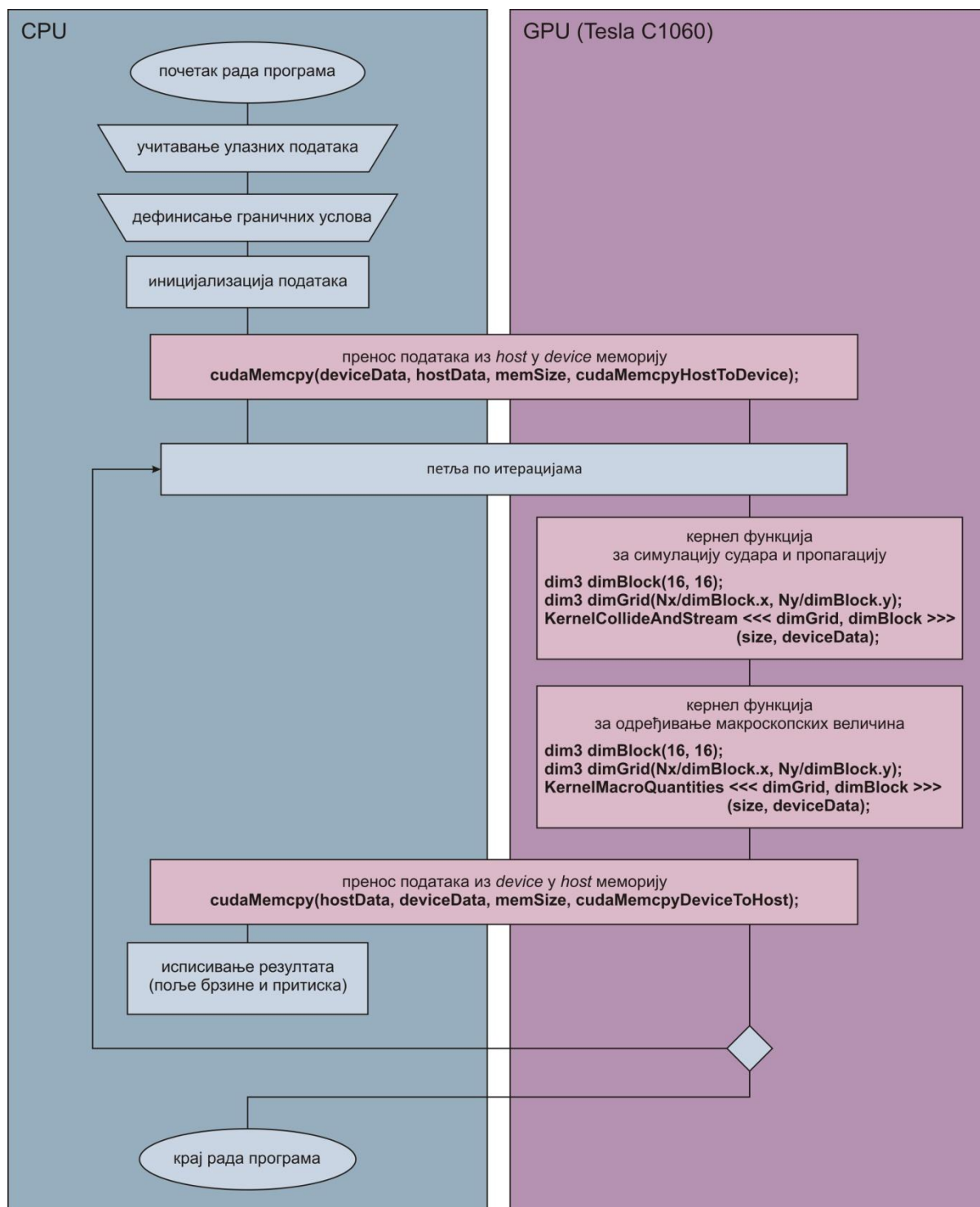
Приликом имплементације кернел функција потребно је водити рачуна о још неким аспектима манипулације меморијом. Већ је поменуто да постоји хијерархија у меморији GPU уређаја. Није исто време потребно да би једна нит приступила податку из локалне (регистарске), дељиве или глобалне меморије. Најбржи је приступ локалној, затим дељивој, а најспорији је приступ глобалној меморији [65,66] (у неким случајевима је и до 150 пута спорији од регистарске меморије). Зато треба избегавати приступ глобалној меморији кад год је то могуће.

Ако се ове чињенице имају у виду, када је у питању паралелизација LB солвера, далеко је погодније уместо позива две кернел функције за симулацију судара и пропагацију, имплементирати једну функцију која спаја ова два корака. То ништа не мења у самој теоријској поставци, пошто се та два корака и даље извршавају одвојено. Ово је само питање имплементације. Вредности функције расподеле  $\tilde{f}_i^{out}(x, t)$  из једначина (3.45) и (3.46) (после завршеног корака судара, а истовремено и почетка корака пропагације) се не записују у глобалну *device* меморију и потом поново учитавају, већ се памте у регистарској меморији сваке нити појединачно. Да би се обезбедило да се заврши корак судара пре почетка пропагације за све чворове (односно за све нити кернел функције), користи се системска CUDA функција за синхронизацију свих нити `__syncthreads()`. Ова функција обезбеђује да све нити заврше са извршавањем свих претходних операција, пре наставка извршавања кернел функције.

Поред паралелизације корака судара и пропагације, било је неопходно паралелизовати и функције које су везане за солид-флуид интеракцију. Уколико би те функције остале непромењене, то би поново узроковало трансфере података између *host* и *device* меморије, што није добро решење. Због тога су и поменуте функције паралелизоване. У оквиру њих постоји доста стандардних матричних операција (множење матрице вектором, одузимање и множење матрица, решавање система једначина). За извршавање ових операција нису писане појединачне кернел функције, већ су коришћене већ дефинисане функције, у оквиру посебне CUDA библиотеке која је специјализована за линеарну алгебру – CUBLAS (скраћено од *CUDA Basic Linear Algebra Subroutines*) [67]. На тај начин је програмерски део посла скраћен, а поред тога,

додатна предност функција у оквиру CUBLAS библиотеке је и што су оне додатно оптимизоване да оптимално користе ресурсе GPU уређаја.

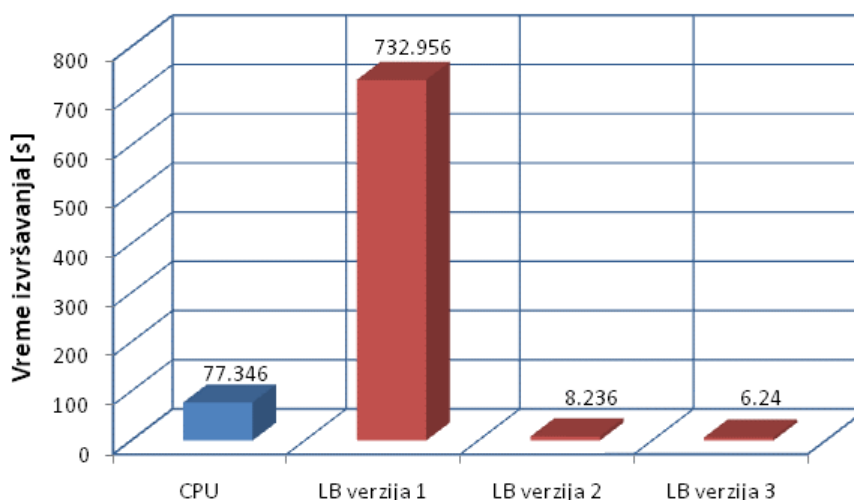
На слици 7.4 приказан је алгоритам извршавања паралелизованог LB солвера.



Слика 7.4 - Алгоритам извршавања паралелизованог LB солвера

### 7.4 Поређење брзине рада стандардне и паралелизоване верзије LB солвера

У наставку ће бити приказано убрзање које се постиже када се користи паралелизована верзија LB солвера. Најпре ће на једном примеру бити илустрована важност правилне манипулације меморијом у CUDA програмима. Као тест пример је коришћен пример 1 из Поглавља 6, резолуција је постављена на 100 (што значи да је број чворова 20.000), а број итерација је постављен на 20.000. Одређивано је време извршавања стандардне верзије LB солвера, као и време извршавања три различите верзије паралелизованог LB солвера. Прва верзија после сваке итерације (после сваког позива кернел функције) врши трансфер података из *device* у *host* меморију и обрнуто. Друга верзија је оптимизована са аспекта трансфера меморије – уведена је додатна кернел функција која одређује макроскопске величине и само се ове вредности преносе у *host* меморију, због приказивања резултата. У трећој верзији је минимизован приступ глобалној *device* меморији, односно обједињени су кораци симулације судара и пропагације у једну кернел функцију, као што је већ објашњено у Поглављу 7.3. Упоредни приказ времена извршавања (у секундама) дат је на слици 7.5.



Слика 7.5 - Упоредни приказ времена извршавања различитих верзија LB солвера

Са слике 7.5 је очигледно да лоша манипулација меморијом може да има веома велики утицај на време извршавања. Прва верзија LB солвера се извршава скоро 10

пута спорије од класичне верзије, што је наравно неприхватљиво. Правилна манипулација меморијом (као у другој и трећој верзији) је довела до убрзања од скоро 10 пута у односу на класичну верзију, што је очекивано убрзање. Ако се упореди време извршавања друге и треће верзије, види се да трећа верзија ради брже, због оптималнијег приступа глобалној меморији, али је разлика мање драстична, у односу на прву верзију.

Побољшање перформанси постојећег програма односно убрзање који се постиже паралелизацијом зависи од више фактора, али један од главних је проценат програма који се не може паралелизовати. Amdahl-ов закон дефинише максимално убрзање које се може постићи ако се паралелизује одређени део програмског кода [68]. Максимално убрзање се може одредити применом следеће формуле:

$$S = \frac{1}{(1 - P) + \frac{P}{N}} \quad (7.1)$$

где је  $P$  удео програма који се паралелизује (најлакше се може одредити ако се подели време извршавања дела који се паралелизује са укупним временом рада класичног програма), а  $N$  је број процесора.

У случају LB солвера, може се сматрати да је број процесора једнак броју CUDA језгара, дакле 240. Међутим, ако се мало детаљније анализира израз (7.1), постаје јасно да што је већи број процесора, однос  $\frac{P}{N}$  је мањи и постаје величина нижег реда, која се може занемарити. Тако да са порастом броја процесора ова величина заправо губи на важности. Може се сматрати да ово тврђење важи и у овом случају.

Део LB солвера који се не паралелизује укључује иницијализацију података, као и приказивање резултата. Због тога се може рећи да је удео програма који се не паралелизује око 5% од укупног времена рада програма, што значи да је  $P = 0.95$ .

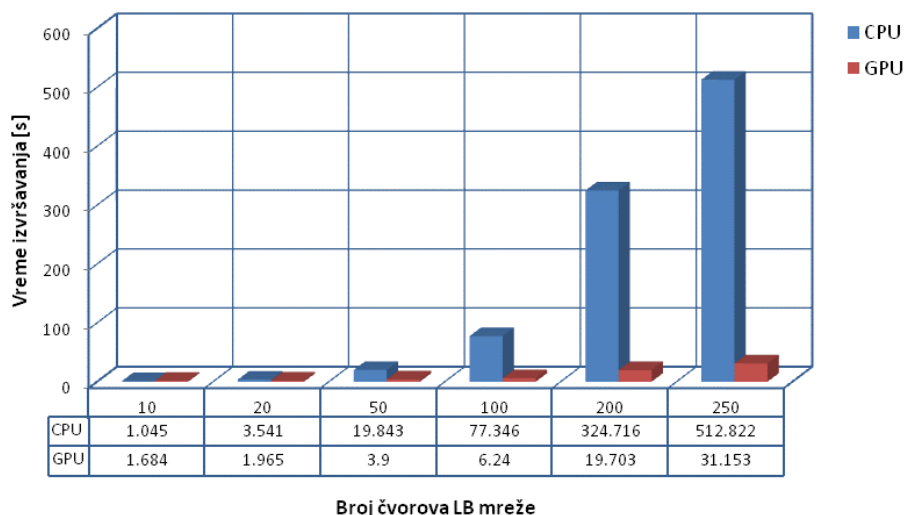
Ако се замене вредности за  $P$  и  $N$ , добија се:

$$S = 18,53 \quad (7.2)$$

Дакле, процењено максимално убрзање које се може постићи паралелизацијом LB солвера (и применом једног Tesla C1060 GPU уређаја) је 18,5.

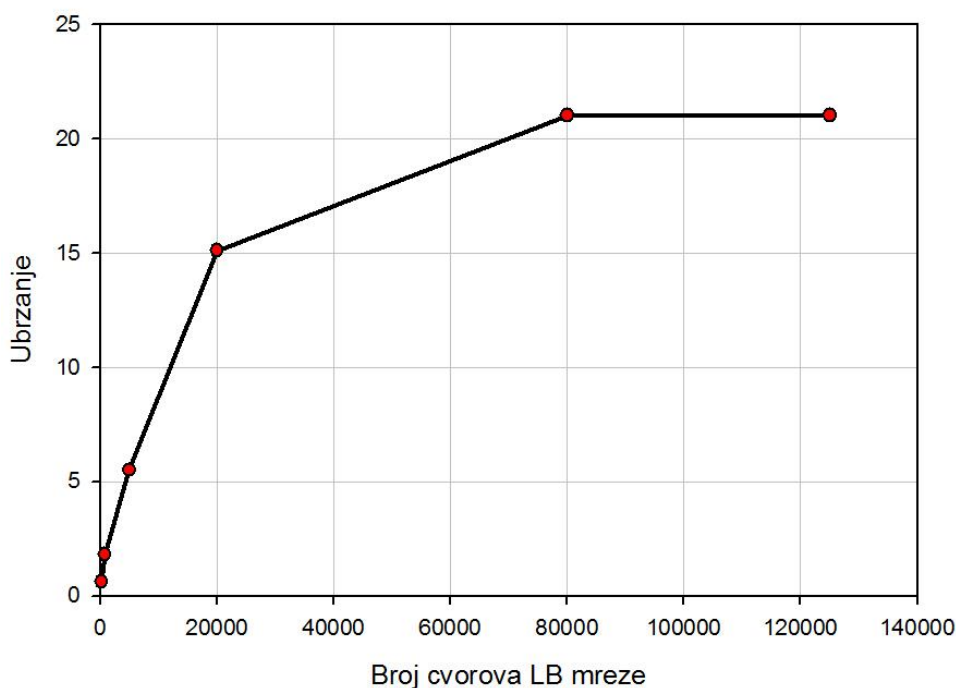
Када се пореди брзина рада паралелизованог LB солвера у односу на класични солвер, потребно је имати у виду и утицај броја чворова LB мреже, односно димензија

матрица над којима се врше операције. Што је број чворова већи, убрзање постигнуто применом GPU уређаја је веће. Време извршавања обе верзије солвера је приказано на дијаграму 7.6.



Слика 7.6 - Упоредни приказ времена извршавања за различит број чворова

На слици 7.7 је приказано како се убрзање паралелизованог LB солвера у односу на класични солвер мења у зависности од броја чворова LB мреже. Како се број чворова повећава, расте и убрзање. Уколико се убрзање одређује на основу времена извршавања целог програма, максимална вредност износи 16.5 пута (за 80.000 чворова). Даље повећање броја чворова нема утицаја на убрзање.



Слика 7.7 - Дијаграм зависности убрзања паралелизованог LB солвера у зависности од броја чворова

Реално достигнута вредност убрзања од 16,5 пута је нешто мања од предвиђених 18,5 пута, што се може објаснити чињеницом да иако минимализовани, трансфери података између *host* и *device* меморије постоје и потребно је одређено време да се они изврше, а то време није директно утрошено на извршавање операција које се иначе извршавају у класичној верзији LB солвера. На слици 7.7 су приказане вредности убрзања израчунате на основу измереног времена извршавања само оног дела програма у коме се спровode операције израчунавања (петља по итерацијама, без приказивања резултата). У том случају, убрзање које се постиже је и до 21, што је чак и веће од предвиђене вредности.

## 8 Закључак

---

Када се упореди *lattice Boltzmann* метода са другим класичним методама рачунске динамике флуида, очигледно је да се LB метода знатно разликује од осталих. Главна разлика је у начину на који се описује флуид. Код класичних метода, непознате величине које се добијају решавањем Навије-Стоксове једначине су макроскопске величине – поља брзине и притиска, која су просторно и временски променљива. Са друге стране, LB метода симулира динамику такозване честичне функције расподеле, која је изведена из кинетичке теорије гасова и која представља вероватноћу постојања честица у неком простору. Карактеристике флуида са аспекта континуума се могу изразити на основу функције расподеле, која се израчунава у сваком временском кораку у оквиру симулације.

Моделирање солид-флуид интеракције и уопште симулација кретања солидних честица кроз домен флуида је последњих година јако често обрађивана тема у савременим радовима из области примене рачунске динамике флуида у биомедицини. Постоји неколико приступа, а један од њих, IBM (енг. *Immersed Boundary Method*) је коришћен у овом раду. Овај приступ је посебно развијен да би се лакше симулирали проблеми солида са комплексним граничним доменом који се крећу кроз домен флуида.

Спрезање домена солида и флуида које је примењено у овом раду користи све предности LB и IB методе, с обзиром да се домени солида и флуида решавају применом две независне мреже – фиксиране *Euler*-ове мреже за флуид и покретног скупа *Lagrange*-ових тачака за солид. Захваљујући једноставности третирања домена флуида, решавање једначина за флуид применом LB методе је поједностављено, док се применом независне и произвољне *Lagrange*-ове мреже лако симулира комплексна граница домена солида. Решавање домена флуида (решавање Навије-Стоксове једначине, са дејством спољашње силе) је потпуно идентично као да нема уроњеног солида, а интеракција са солидом се симулира увођењем спољашње силе, која се одређује тако да брзина у граничним тачкама буде једнака нули (да нема пенетрације

флуида уз границу солида). Ова сила се директно израчунава преко корекција брзине у граничним тачкама. Ипак, с обзиром да се све величине везане за интеракцију рачунају интерполацијом преко тачака на граничној површини, применом *Dirac*-ове делта функције, постоји одређена грешка у тачности израчунатих вредности. У том смислу је битно одабрати одговарајућу густину обе мреже, да би се избегле грешке у рачунању.

У овом раду су приказани резултати добијени применом посебно развијеног софтвера на бази LB методе, са имплементираним спрезањем домена солида и флуида. Добијени резултати су упоређени са резултатима који су добијени применом бројних поменутих метода и који су објављени у литератури. Симулирано је укупно 8 различитих проблема који су значајни у области биомедицине, као што је на пример кретање честица кроз бифуркацију или артерију са стенозом. Поређење са другим резултатима из литературе и добро поклапање решења доказују да се развијени софтвер може успешно примењивати за симулације струјања нестишљивог флуида.

Са друге стране, чини се да програми за научне симулације захтевају све више рачунарских ресурса. Персонални рачунари више не могу да прате те повећане захтеве. NVIDIA је својом серијом Tesla GPU уређаја понудила добру алтернативу за решавање овог новонасталог проблема. Применом GPU уређаја постојећи компликовани и захтевни програми се могу драстично убрзати. Захваљујући посебној технологији коју је развила NVIDIA, такозваној CUDA (енг. *Compute Unified Device Architecture*) архитектури, прилагођавање програма за рад са GPU уређајима је значајно поједностављено. У овом раду је применом CUDA архитектуре већ поменути LB солвер паралелизован и упоређена је брзина рада класичне и паралелизоване верзије. На специјализованом Tesla уређају постигнуто је убрзање од чак 20 пута, што представља велику уштеду времена. Симулације струјања флуида и солид-флуид интеракције, за које је раније требало и по неколико сати, сада се извршавају за неколико минута, што је свакако значајно. Кретање солидних честица се применом паралелизованог софтвера може пратити интерактивно.

Оно што је евидентно је да је LB метода последњих година постала веома интересантна и применљива у симулацијама струјања флуида, пре свега са комплексним доменом флуида и са солидним честицама које се крећу унутар домена. Управо зато је тема овог рада моделирање солид-флуид интеракције применом LB методе. Уз не тако велике измене у методологији, развијени LB солвер се може даље проширити тако да симулира кретања деформабилних честица. Такође, треба напоменути да иако су сви проблеми разматрани у овом раду били равански и струјање флуида дводимензионално, све једначине које су наведене су универзалне и IBM приступ важи и за тродимензионални домен. То значи да се цео концепт може без

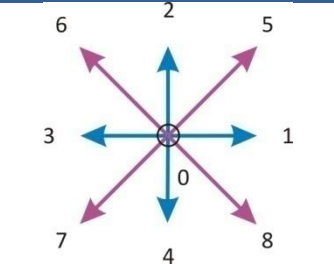
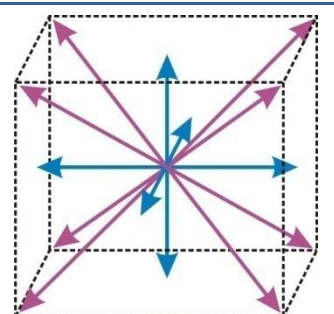
већих проблема применити и за тродимензионалне проблеме. Због једноставне имплементације солид-флуид интеракције, али и значајно смањене брзине рада која је постигнута паралелизацијом, развијени софтвер има велики потенцијал када је реч о практичној примени у симулацијама реалних проблема у биоинжењерингу.

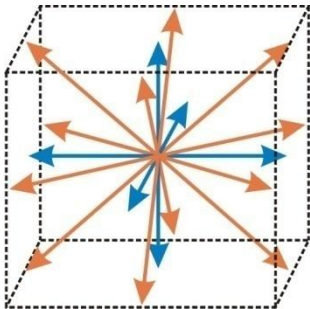
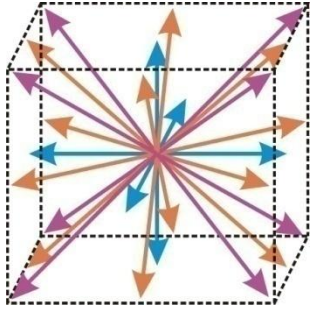
## Додатак А

Структура мреже са  $q$  различитих праваца брзина у  $d$ -димензионом простору се означава као  $DdQq$  мрежа (енг. *lattice*). За симулације струјања изотермалног нестишљивог флуида користи се неколико 2D и 3D структура мреже и они су овде наведени. За сваки тип је дата слика која илуструје правце брзина. Такође, дати су и одговарајући коефицијенти – тежински коефицијенти  $\omega_i$ , коефицијенти за дефинисање апсцисних оса  $v_i$ .

Треба напоменути да је за све наведене структуре коефицијент означен као „брзина звука мреже“  $c_s$  исти и износи:

$$c_s^2 = \frac{1}{3}$$

|                                                                                                         |                                                                                                                                                                                                                                                                               |
|---------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>D2Q9</b></p>   | $\omega_i = \begin{cases} 4/9 & i = 0 \\ 1/9 & i = 1, 2, 3, 4 \\ 1/36 & i = 5, 6, 7, 8 \end{cases}$ $v_i = \begin{cases} (0, 0) & i = 0 \\ (\pm\sqrt{3}, 0) & i = 1, 2, 3, 4 \\ (0, \pm\sqrt{3}) & i = 5, 6, 7, 8 \end{cases}$                                                |
| <p><b>D3Q15</b></p>  | $\omega_i = \begin{cases} 2/9 & i = 0 \\ 1/9 & i = 1 - 6 \\ 1/72 & i = 7 - 14 \end{cases}$ $v_i = \begin{cases} (0, 0, 0) & i = 0 \\ (\pm\sqrt{3}, 0, 0) & i = 1 - 6 \\ (0, 0, \pm\sqrt{3}) & i = 7 - 14 \\ (\pm\sqrt{3}, \pm\sqrt{3}, \pm\sqrt{3}) & i = 7 - 14 \end{cases}$ |

|                                                                                    |                                                                                                                                                                                                                                                                                                                     |
|------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $D3Q19$                                                                            | $\omega_i = \begin{cases} 1/3 & i = 0 \\ 1/18 & i = 1 - 6 \\ 1/36 & i = 7 - 18 \end{cases}$                                                                                                                                                                                                                         |
|   | $v_i = \begin{cases} (0,0,0) & i = 0 \\ (\pm\sqrt{3}, 0, 0) & \\ (0, \pm\sqrt{3}, 0) & i = 1 - 6 \\ (0, 0, \pm\sqrt{3}) & \\ (\pm\sqrt{3}, \pm\sqrt{3}, 0) & \\ (0, \pm\sqrt{3}, \pm\sqrt{3}) & i = 7 - 18 \\ (\pm\sqrt{3}, 0, \pm\sqrt{3}) & \end{cases}$                                                          |
| $D3Q27$                                                                            | $\omega_i = \begin{cases} 8/27 & i = 0 \\ 2/27 & i = 1 - 6 \\ 1/216 & i = 7 - 14 \\ 1/54 & i = 15 - 26 \end{cases}$                                                                                                                                                                                                 |
|  | $v_i = \begin{cases} (0,0,0) & i = 0 \\ (\pm\sqrt{3}, 0, 0) & \\ (0, \pm\sqrt{3}, 0) & i = 1 - 6 \\ (0, 0, \pm\sqrt{3}) & \\ (\pm\sqrt{3}, \pm\sqrt{3}, \pm\sqrt{3}) & i = 7 - 14 \\ (\pm\sqrt{3}, \pm\sqrt{3}, 0) & \\ (0, \pm\sqrt{3}, \pm\sqrt{3}) & i = 15 - 26 \\ (\pm\sqrt{3}, 0, \pm\sqrt{3}) & \end{cases}$ |

- [1] C.Y. Lim, C. Shu, X. D. Niu, and Y. T. Chew, "Application of the lattice Boltzmann Method to simulate microchannel flows," *Physics of Fluids*, vol. 14, no. 7, pp. 2299–2308., 2002.
- [2] S. Chen and Z. Tian, "Simulation of microchannel flow by lattice Boltzmann method," *Physica A*, vol. 388, no. 23, pp. 4803-4810, 2009.
- [3] H. Hatzikirou, L. Brusch, C. Schaller, M. Simon, and A. Deutsch, "Prediction of traveling front behavior in a lattice-gas cellular automaton model for tumor invasion," *Computers and Mathematics with Applications*, vol. 59, no. 7, pp. 2326-2339, 2010.
- [4] A. Sequeira, A.M. Artoli, A.S. Silva-Herdade, and C. Saldanha, "Leukocytes dynamics in microcirculation under shear-thinning blood flow," *Computers and Mathematics with Applications*, vol. 58, no. 5, pp. 1035-1044, 2009.
- [5] C. Migliorini et al., "Red blood cells augment leukocyte rolling in a virtual blood vessel," *Biophysical Journal*, vol. 83, no. 4, pp. 1834-1841., 2002.
- [6] C. Sun and L.L. Munn, "Lattice-Boltzmann simulation of blood flow in digitized vessel networks," *Computers and Mathematics with Applications*, vol. 55, no. 7, pp. 1594-1600., 2008.
- [7] C. Sun, C. Migliorini, and L.L. Munn, "Red blood cells initiate leukocyte rolling in postcapillary expansions: A lattice Boltzmann analysis," *Biophysical Journal*, vol. 85, no. 1, pp. 208-222, 2003.
- [8] D. Mateescu, A. Mekanik, and M.P. Paidoussis, "Analysis of 2-D and 3-D unsteady annular flows with oscillating," *Journal of Fluids and Structures*, vol. 10, pp. 57–77, 1996.
- [9] R. Li, T. Tang, and P.W. Zhang, "A moving mesh finite element algorithm for singular problems in two and three space," *Journal of Computational Physics*, vol. 177, pp. 365–393, 2002.

- [10] H.H. Hu, N.A. Patankar, and M.Y. Zhu, "Direct numerical simulations of fluid–solid systems using arbitrary Lagrangian–," *Journal of Computational Physics*, vol. 169, pp. 427–462, 2001.
- [11] C. S. Peskin, "Numerical analysis of blood flow in the heart," *Journal of Computational Physics*, vol. 25, no. 3, pp. 220–252, 1977.
- [12] Z. Feng and E. Michaelides, "The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problem," *Journal of Computational Physics*, vol. 195, no. 2, pp. 602–628, 2004.
- [13] X. D. Niu, C. Shu, Y. T. Chew, and Y. Peng, "A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows," *Phys. Lett. A*, vol. 354, no. 3, pp. 173–182, 2006.
- [14] J. Wu and C. Shu, "Particulate flow simulation via a boundary condition-enforced immersed boundary-lattice Boltzmann scheme," *Commun. Comput. Phys.*, vol. 7, no. 4, pp. 793–812, 2010.
- [15] M. Schmeisser et al., "Parallel, distributed and GPU computing technologies in single-particle electron microscopy," *Acta Cryst. D65*, vol. 65, no. 7, pp. 659–671, 2009.
- [16] R. Strzodka, M. Doggett, and A. Kolb, "Scientific computation for simulations on programmable graphics hardware," *Simulation Modeling Practice and Theory*, vol. 13, no. 8, pp. 667–680, 2005.
- [17] W. Liu, B. Schmidt, G. Voss, and W. Muller-Wittig, "Streaming Algorithms for Biological Sequence Alignment on GPUs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 9, pp. 1270–1281, 2007.
- [18] M. Bernaschi, M. Fatica, S. Melchionna, S. Succi, and E. Kaxiras, "A flexible high-performance lattice Boltzmann GPU code for the simulations of fluid flows in complex geometries," *Concurrency and Computation: Practice and Experience*, vol. 22, no. 1, pp. 1–14, 2010.
- [19] J.A. Anderson, C.D. Lorenz, and A. Travesset, "General purpose molecular dynamics simulations fully implemented on graphics processing units," *Journal of Computational Physics*, vol. 227, no. 10, pp. 5342–5359, 2008.
- [20] H. Li and L. Petzold, "Efficient parallelization of the stochastic simulation algorithm for chemically reacting systems on the GPU," *The International Journal of High*

- Performance Computing Applications*, vol. 24, no. 2, pp. 107–116, 2010.
- [21] D. Vintache, B. Humbert, and D. Brasse, "Iterative Reconstruction for Transmission Tomography on GPU Using Nvidia CUDA," *Tsinghua Science & Technology*, vol. 15, no. 1, pp. 11-16, 2010.
- [22] D. Komatitsch, D. Michéa, and G. Erlebacher, "Porting a high-order finite-element earthquake modeling application to NVIDIA graphics cards using CUDA," *Journal of Parallel and Distributed Computing*, vol. 69, no. 5, pp. 451-460, 2009.
- [23] S. Wolfram, *Cellular Automaton Fluids 1: Basic Theory.*: J. Stat. Phys., 3/4:471–526, 1986.
- [24] D. H. Rothman and S. Zaleski, *Lattice Gas Cellular Automata. Simple models of Complex Hydrodynamics*. England: Cambridge University Press, 1997.
- [25] G. McNamara and G. Zanetti, "Use of the Boltzmann equation to simulate lattice-gas automata," *Phys. Rev. Lett.*, vol. 61, no. 20, pp. 2332-2335, 1988.
- [26] T. Đukić, *Lattice Boltzmann metod, Završni rad*. Kragujevac: Mašinski fakultet, 2010.
- [27] O. P. Malaspinas, *Lattice Boltzmann Method for the Simulation of Viscoelastic Fluid Flows*. Switzerland: PhD dissertation, 2009.
- [28] P. L. Bhatnagar, E. P. Gross, and M. Krook, "A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems," *Phys. Rev. E*, vol. 77, no. 5, pp. 511-525, 1954.
- [29] A. H. Carter, *Classical and Statistical Thermodynamics*. New Jersey: Prentice–Hall, Inc., 2001.
- [30] E. H. Kennard, *Kinetic Theory of Gases*. New York: McGraw-Hill, 1938.
- [31] H. Grad, "Note on the N-dimensional Hermite polynomials," *Communications on Pure and Applied Mathematics*, vol. 2, no. 4, pp. 325–330, 1949.
- [32] H. Grad, "On the kinetic theory of rarefied gases," *Communications on Pure and Applied Mathematics*, vol. 2, no. 4, pp. 331–407, 1949.
- [33] X. B. Nie, X. Shan, and H. Chen, "Galilean invariance of lattice Boltzmann models," *Europhys. Lett.*, vol. 83, no. 3, pp. 169-181, 2008.

- [34] V. I. Krylov, *Approximate Calculation of Integrals*. New York: Macmillan, 1962.
- [35] P.J. Davis and P. Rabinowitz, *Methods of Numerical Integration*. New York, 1984.
- [36] X. He and L.-S. Luo, "Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation," *Phys. Rev. E*, vol. 56, no. 6, pp. 6811–6817, 1997.
- [37] X. He, X. Shan, and G. D. Doolean, "Discrete Boltzmann equation model for non-ideal gases," *Phys. Rev. E*, vol. 57, no. 1, pp. R13 - R16, 1998.
- [38] P. J. Dellar, "Bulk and shear viscosities in lattice Boltzmann equations," *Phys. Rev. E*, vol. 64, no. 3, p. 031203, 2001.
- [39] A. J. Wagner, *A Practical Introduction to the Lattice Boltzmann Method*. North Dakota State University: PhD dissertation, 2008.
- [40] S. Chapman and T. G. Cowling, *The Mathematical Theory of Non-Uniform Gases.*: Cambridge University Press, 1995.
- [41] M.C. Sukop and D.T. Jr. Thorne, *Lattice Boltzmann Modeling - An Introduction for Geoscientists and Engineers*. Heidelberg: Springer, 2006.
- [42] M.A. Gallivan, D.R. Noble, J.G. Georgiadis, and R.O. Buckius, "An evaluation of the bounce-back boundary condition for lattice Boltzmann simulations," *Int J Num Meth Fluids*, vol. 25, no. 3, pp. 249-263, 1997.
- [43] T. Inamuro, M. Yoshina, and F. Ogino, "A non-slip boundary condition for lattice Boltzmann simulations," *Phys. Fluids*, vol. 7, no. 12, pp. 2928-2930, 1995.
- [44] I. Ginzbourg and D. d 'Humières, "Local second-order boundary method for lattice Boltzmann models," *J. Statist. Phys.*, vol. 84, no. 5-6, pp. 927-971, 1996.
- [45] B. Chopard and A. Dupuis, "A mass conserving boundary condition for lattice Boltzmann models," *Int. J. Mod. Phys. B*, vol. 17, no. 1/2, pp. 103-108, 2003.
- [46] Q. Zou and X. He, "On pressure and velocity boundary conditions for the lattice Boltzmann BGK model," *Phys. Fluids*, vol. 9, no. 6, pp. 1592-1598, 1997.
- [47] J. Latt and B. Chopard, "Lattice Boltzmann method with regularized non-equilibrium distribution functions," *Math. Comp. Sim.*, vol. 72, no. 1, pp. 165-168, 2006.

- [48] P.A. Skordos, "Initial and boundary conditions for the lattice Boltzmann method," *Phys. Rev. E*, vol. 48, no. 6, pp. 4823-4842, 1993.
- [49] J. Latt., B. Chopard, O. Malaspinas, M. Deville, and A. Michler, "Straight velocity boundaries in the lattice Boltzmann method," *Phys. Rev. E*, vol. 77, no. 5, p. 056703, 2008.
- [50] N. Filipović, *Numeričko rešavanje spregnutih problema deformabilnog tela solida i strujanja fluida*. Mašinski fakultet Kragujevac: PhD dissertation, 1999.
- [51] E. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof, "Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations," *J. Comput. Phys.*, vol. 161, no. 1, pp. 35-60, 2000.
- [52] Z. Feng and E. Michaelides, "Proteus: A direct forcing method in the simulations of particulate flows," *Journal of Computational Physics*, vol. 202, no. 1, pp. 20-51, 2005.
- [53] M. Uhlmann, "An immersed boundary method with direct forcing for the simulation of particulate flows," *J. Comput. Phys.*, vol. 209, no. 2, pp. 448-476, 2005.
- [54] C. Shu, N. Y. Liu, and Y. T. Chew, "A novel immersed boundary velocity correction-lattice Boltzmann method and its application to simulate flow past a circular cylinder," *J. Comput. Phys.*, vol. 226, no. 2, pp. 1607-1622, 2007.
- [55] J. Wu and C. Shu, "Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its application," *J. Comput. Phys.*, vol. 228, no. 6, pp. 1963-1979, 2009.
- [56] Z. Guo, C. Zheng, and B. Shi, "Discrete lattice effects on the forcing term in the lattice Boltzmann method," *Phys. Rev. E*, vol. 65, no. 4, p. 046308, 2002.
- [57] M. Kojić, R. Slavković, M. Živković, and N. Grujović, *Metod konačnih elemenata 1*. Kragujevac: Mašinski fakultet Kragujevac, 1998.
- [58] A. T. Chwang and T. Y. Wu, "Hydromechanics of low-Reynolds-number flow, Part 4, Translation of spheroids," *J. Fluid Mech.*, vol. 75, no. 4, pp. 677-689, 1976.
- [59] N. Filipović, M. Kojić, P. Decuzzi, and M. Ferrari, "Dissipative Particle Dynamics simulation of circular and elliptical particles motion in 2D laminar shear flow," *Microfluidics and Nanofluidics*, vol. 10, no. 5, pp. 1127-1134, 2010.

- [60] N. Filipović, V. Isailović, T. Đukić, M. Ferrari, and M. Kojić, "Multi-scale modeling of circular and elliptical particles in laminar shear flow," *IEEE Trans Biomed Eng*, vol. 59, no. 1, pp. 50-53, 2012.
- [61] G. B. Jeffery, "The motion of ellipsoidal particles immersed in a viscous fluid," *Proc. R. Soc. Lond. A*, vol. 102, no. 715, pp. 161-180, 1922.
- [62] J. Feng, H. H. Hu, and D. D. Joseph, "Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid. Part 2, Couette and Poiseuille flows," *J. Fluid Mech.*, vol. 277, pp. 271-301, 1994.
- [63] D. Wan and S. Turek, "Direct numerical simulation of particulate flow via multigrid FEM techniques and the fictitious boundary method," *Int. J. Numer. Meth. Fluids*, vol. 51, no. 5, pp. 531-566, 2006.
- [64] H. Li, H. Fang, Z. Lin, S. Xu, and S. Chen, "Lattice Boltzmann simulation on particle suspensions in a two-dimensional symmetric stenotic artery," *Phys. Rev. E*, vol. 69, no. 3, p. 031919, 2004.
- [65] NVIDIA. (2011, May) CUDA Programming guide, Version 4.0. [Online].  
[http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA\\_C\\_Programming\\_Guide.pdf](http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf)
- [66] NVIDIA. (2011, May) CUDA C Best practices guide, Version 4.0. [Online].  
[http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA\\_C\\_Best\\_Practices\\_Guide.pdf](http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Best_Practices_Guide.pdf)
- [67] NVIDIA. (2012, January) CUBLAS Library User Guide. [Online].  
[http://developer.download.nvidia.com/compute/DevZone/docs/html/CUDALibraries/doc/CUBLAS\\_Library.pdf](http://developer.download.nvidia.com/compute/DevZone/docs/html/CUDALibraries/doc/CUBLAS_Library.pdf)
- [68] G. Amdahl, "Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities," in *AFIPS Conference Proceedings (30)*, 1967, pp. 483–485.