

ŽILINSKÁ UNIVERZITA V ŽILINE
FAKULTA RIADENIA A INFORMATIKY



Modelovanie buniek s jadrom pre simulačné modely
mikrofluidických zariadení

DIZERTAČNÁ PRÁCA

2020

Mgr. Alžbeta BOHNIKOVÁ

ŽILINSKÁ UNIVERZITA V ŽILINE
FAKULTA RIADENIA A INFORMATIKY

**Modelovanie buniek s jadrom pre simulačné modely
mikrofluidických zariadení.**

**DIZERTAČNÁ PRÁCA
28360020203007**

Študijný program: Aplikovaná informatika
Študijný odbor: Informatika
Školiace pracovisko: Katedra softvérových technológií
Fakulta riadenia a informatiky
Žilinská univerzita v Žiline
Vedúci práce: doc. RNDr. Katarína Bachratá, PhD.
Školiteľ špecialista: Mgr. Iveta Jančígová, PhD.

Žilina 2020

Mgr. Alžbeta BOHNIKOVÁ

Anotácia

Typ práce:	Dizertačná práca
Akademický rok:	2019/2020
Názov práce:	Modelovanie buniek s jadrom pre simulačné modely mikrofluidických zariadení.
Autor:	Mgr. Alžbeta Bohiniková
Školiteľ:	doc. RNDr. Katarína Bachratá, PhD.
Školiteľ špecialista:	Mgr. Iveta Jančígová, PhD.
Jazyk:	Slovenčina
Počet strán:	166
Počet obrázkov:	51
Počet tabuliek:	20
Počet referencií:	89
Kľúčové slová:	model bunky s jadrom, výpočtový model, mikrofluidické zariadenia, interakcia membrány bunky s jadrom

Pod'akovanie Hlavne by som sa chcela poďakovať mojej školiteľke, doc. RNDr. Kataríne Bachratej, PhD., za jej cenné rady, odborné vedenie a usmernenia, ktoré ma naučili väčšej samostatnosti. Veľká vďaka patrí aj mojej školiteľke špecialistke, Mgr. Ivete Jančigovej, PhD., a celej výskumnej skupine Cell in Fluid, za ich vzácne rady a pomoc s nespočtým množstvom otázok a chybových hlások simulácií.

Abstrakt

BOHINIKOVÁ, Alžbeta: *Modelovanie buniek s jadrom pre simulačné modely mikrofluidických zariadení*. [Dizertačná práca] Žilinská Univerzita v Žiline. Fakulta riadenia a informatiky. Katedra softvérových technológií. - Vedúci diplomovej práce: doc. RNDr. Katarína Bachratá, PhD.- Školiteľ špecialista: Mgr. Iveta Jančígová, PhD.- Žilina, FRI ZU, 2020, 166 s.

Dizertačná práca sa zaoberá rozšírením simulačného nástroja vyvíjaného výskumnou skupinou Cell in Fluid o model bunky s jadrom. Hlavným zameraním práce bolo preto popísať a otestovať rôzne prístupy modelovania buniek s jadrom. Pozornosť sme venovali vzájomnej interakcii medzi bunkovou membránou a jadrom. Modelovali sme ju pomocou vytvorenia fyzických väzieb medzi bodmi membrány a jadra alebo pomocou odpudivých síl medzi skupinami týchto bodov. Navrhnutý simulačný experiment je parametrizovaný a tak ho je možné použiť pre kalibráciu špecifických typov buniek v rôznych škálach geometrií a veľkostí buniek. Dôraz bol kladený aj na prepojenie biologických a simulačných experimentov. Pri vytváraní simulácií popisujeme možné výzvy biologických dát a akým spôsobom s nimi pracovať.

Kľúčové slová: Model bunky s jadrom, výpočtový model, mikrofluidické zariadenia, interakcia membrány bunky s jadrom.

Abstract

BOHINIKOVÁ, Alžbeta: *Modelling of cells with nucleus for simulation models of microfluidic devices*. [Dissertation thesis] - University of Žilina. Faculty of Management Science and Informatics. Department of Software Technology. - Supervisor: doc. RNDr. Katarína Bachratá, PhD.- Supervisor specialist: Mgr. Iveta Jančígová, PhD.- Žilina, FRI ZU, 2020, 166 p.

This dissertation thesis contains an extension of the simulation tool developed by the research group Cell in Fluid. The main focus of the work was therefore to describe and test different approaches to modeling cells with nucleus. We examined the interaction between the cell membrane and the nucleus. We modeled this interaction by creating physical bonds between the points of the membrane and the nucleus or by repulsive forces between groups of these points. The proposed simulation experiment is parameterized, so it can be used to calibrate specific cell types in different scales of channel geometries and cell sizes. Emphasis was also placed on linking biological and simulation experiments. When creating simulations, we describe the possible challenges of biological data and how to work with them.

Key words: Cell model with nucleus, computational model, microfluidic devices, interaction of cell membrane with nucleus.

Obsah

1	Úvod	15
1.1	Motivácia	16
1.2	Ciele práce	17
1.3	Prehľad práce	18
2	Biologické vlastnosti buniek a mikrofluidické zariadenia	19
2.1	Bunky s jadrom - biologické vlastnosti	19
2.2	Bunkové línie	21
2.3	Mikrofluidické zariadenia	22
2.3.1	Prehľad triediacich a zachytávajúcich mikrofluidických zariadení	22
2.3.2	Experimenty merajúce biofyzikálne vlastnosti buniek	26
3	Modelovanie buniek	31
3.1	Rôzne prístupy k modelovaniu buniek	31
3.2	Popis modelu RBC	35
3.2.1	ESPResSo a PyOIF	36
3.2.2	Výpočtové prostriedky	37
3.3	Kalibrácia RBC	37
3.4	Overenie kalibrácie RBC	46
4	Model bunky s jadrom	49
4.1	Simulačný experiment pre testovanie vplyvu parametrov interakcií . .	49
4.1.1	Metódy porovnávania simulačných experimentov	51
4.2	Model bunky s jadrom s použitím viazaných interakcií	52
4.2.1	Testované viazané interakcie	54
4.2.2	Výsledky a porovnanie	55
4.3	Model bunky s jadrom s použitím neviazaných interakcií	59
4.4	Model bunky s jadrom reprezentovaný elastickou guľou	62
4.4.1	Self-cell interakcia membrány bunky	62
4.4.2	Elasticita a Self-cell interakcia	65
4.5	Porovnanie prístupov	66

5	Simulácie biologických experimentov	69
5.1	Dáta z biologických experimentov a ako s nimi pracovať	69
5.1.1	Popis biologického experimentu	70
5.1.2	Mikrofluidické kanáliky so zúžením	72
5.1.3	Spracovanie biologických dát a výpočtový model	73
5.2	Mikrofluidický kanál so zúžením	75
5.2.1	Nastavenie simulácie	76
5.2.2	Vstupný čas v závislosti od sily tekutiny	77
5.2.3	Kalibrácia parametrov	79
5.2.4	Objemový prietok a sila tekutiny	79
5.2.5	Ďalšie typy zúžených kanálikov	80
5.3	Laserové ťaženie buniek	81
5.3.1	Simulácie optického ťaženia	83
5.3.2	Výpočet ťaživej sily	84
5.3.3	Výsledky	88
5.4	Kalibrácia na základe biologických experimentov	90
5.5	Možnosti využitia	94
6	Záver	97
	Publikácie	99
	Literatúra	101
	Prílohy	111
A	Doplňujúce tabuľky	113
A.1	Výsledky kalibrácie RBC bunky	113
A.2	Výsledky simulácií v ktorých bol použitý kvadratický potenciál.	121
A.3	Výsledky simulácií v ktorých bol použitý FENE potenciál.	122
B	Python skripty	123
B.1	Geometria zúženého kanálka	123
B.2	Viazané interakcie - harmonický potenciál	125
B.3	Viazané interakcie - FENE	134
B.4	Viazané interakcie - kvadratický potenciál	143
B.5	Neviazané interakcie	152
B.6	Elastická guľa v zúženom kanáliku	158
C	Použité skratky	165

Zoznam obrázkov

2.1	Stavba eukaryotickej bunky.	19
2.2	Mikrofluidické zariadenie využívajúce optický laser pre zachytávanie CTC buniek.	23
2.3	Mikrofluidické zariadenie využívajúce magnetické guľôčky pre zachytávanie CTC buniek.	24
2.4	Mikrofluidické zariadenie využívajúce zúženie toku pre triedenie CTC buniek.	25
2.5	Mikrofluidické zariadenie využívajúce odbočujúce kanáliky pre triedenie pečeňových buniek.	26
2.6	Mikrofluidický kanál so zúžením	28
3.1	Naťahovanie RBC.	39
3.2	Kontaktná plocha silikových guľôčok.	40
3.3	Rôzne spôsoby naťahovania buniek.	41
3.4	Výsledky naťahovacích simulácií s použitím prstencov.	42
3.5	Výsledky naťahovacích simulácií s použitím bodov.	43
3.6	Výsledky naťahovacích simulácií s použitím čiapočiek.	43
3.7	Časť hyperbolického mikrofluidického kanálika.	47
4.1	Kanálik so zúžením.	50
4.2	Vytváranie väzieb medzi jadrom a membránou.	53
4.3	Zobrazenie väzieb medzi membránou a jadrom.	54
4.4	Výsledky simulácií v ktorých bol použitý harmonický potenciál.	56
4.5	Výsledky simulácií v ktorých bol použitý kvadratický potenciál.	56
4.6	Výsledky simulácií v ktorých bol použitý FENE potenciál.	56
4.7	Porovnanie harmonického a kvadratického potenciálu.	57
4.8	Graf zobrazujúci závislosť D_s od Δr_{max} pri použití FENE potenciálu.	57
4.9	Graf zobrazujúci závislosť D_s od k pri použití harmonického potenciálu.	58
4.10	Grafy zobrazujúce závislosť D_s od k a a pri použití harmonického potenciálu.	58
4.11	Grafy zobrazujúce závislosť D_s od K a a pri použití FENE potenciálu.	59
4.12	Grafy zobrazujúce závislosť D_s od K_0 a a pri použití kvadratického potenciálu.	59
4.13	Výsledky simulácií v ktorých bola použitá soft-sphere interakcia.	61

4.14	Graf zobrazujúci závislosť D_s od a pri použití soft-sphere interakcie.	61
4.15	Grafy zobrazujúce závislosť D_s od a a r_{cut} pri použití soft-sphere interakcie.	61
4.16	Neprirodzené správanie bunky v úzkom kanáli.	62
4.17	Funkcia odpudivej sily pre RBC.	63
4.18	Kanálik so zúžením.	65
4.19	Vplyv nesprávneho nastavenia hodnoty r_{cut} na tvar bunky.	66
4.20	Porovnanie DI bunky pri použití rôznych spôsobov modelovania bunky s jadrom.	68
5.1	Zaznamenávanie biologického experimentu	70
5.2	Chipy použité na experimenty.	71
5.3	Rozmery kanálíka so zúžením.	71
5.4	Porovnanie deformačného indexu (DI) 5 buniek z biologického experimentu.	74
5.5	Pozície jednotlivých buniek z biologického experimentu.	74
5.6	Mikrofluidický kanál so zúžením.	76
5.7	Závislosť vstupného času od sily tekutiny.	78
5.8	Schematické zobrazenie ťahovania pomocou optickej pasce.	82
5.9	Profil mechanického napätia vypočítaný pre rôzne hodnoty maximálneho mechanického napätia.	83
5.10	Porovnanie rôznych prístupov k výpočtu uhlu medzi normálou povrchu bunky a vektorom pôsobenia laserového lúča.	84
5.11	Rozdiely medzi tvarmi RBC po $1\mu s$ simulačného času.	87
5.12	Analyticky vypočítané radiálne tvary bunky.	89
5.13	Porovnanie biologických a simulačných hodnôt ťahovacieho experimentu. V tomto prípade sa dáta nezhodujú.	93
5.14	Porovnanie biologických a simulačných hodnôt ťahovacieho experimentu. V tomto prípade sú dáta podobné.	93
5.15	Porovnanie biologických a simulačných hodnôt ťahovacieho experimentu.	93
5.16	Porovnanie biologických a simulačných hodnôt ťahovacieho experimentu.	93
5.17	Mikrofluidické zariadenie určené na zachytávanie klastrov buniek.	95
5.18	Detail časti mikrofluidického zariadenia zachytávajúceho klastre buniek.	95

Zoznam tabuliek

3.1	Simulačné parametre	42
3.2	Výsledky ťahovacích simulácií pre rôzne nastavenia x -ových súradníc.	44
3.3	Intervaly pre x -ové súradnice pre rôzne diskretizácie	45
3.4	Simulačné parametre	48
4.1	Počet väzieb medzi jadrom a membránou v závislosti od nastavenia uhla α	53
4.2	Dĺžka hrán pri rôznych diskretizáciách buniek, $r = 3.91\mu m$	64
4.3	Dĺžka hrán pri rôznych diskretizáciách buniek, $r = 8\mu m$	64
4.4	Dĺžka hrán pri rôznych diskretizáciách buniek, $r = 12\mu m$	64
5.1	Hodnoty namerané v biologickom experimente [37].	76
5.2	Parametre self-cell interakcie	77
5.3	Parametre simulácie	78
5.4	Parametre simulácie	78
5.5	Fungujúce elastické parametre	79
5.6	Testovanie odchýlok LBgridu s rovnakou silou tekutiny	79
5.7	Testovanie rôznych hodnôt LBgrid s rôznymi hodnotami sily tekutiny	80
5.8	Testovanie vzťahu medzi hodnotami LBgrid a silami tekutiny	80
5.9	Maximálne hodnoty mechanického napätia z [25] použité v simuláciách.	83
5.10	Parametre simulácie	86
5.11	Porovnanie výpočtových časov pri rôznych počtoch iteračných krokov na jeden simulačný krok.	86
5.12	Porovnanie experimentálnych a simulačných dát.	89

1 Úvod

Modelovanie biologických experimentov je pomerne rozsiahla oblasť, ktorej sa v posledných rokoch venuje veľká pozornosť. Vo výskumnej skupine Cell in Fluid sa venujeme najmä modelovaniu krvi. Existuje množstvo rôznych prístupov ku modelovaniu. Otázky, na ktoré má model odpovedať, ovplyvňujú do akého detailu je potrebné vytvárať počítačové simulácie.

V prípadoch, kedy je cieľom modelovať veľké množstvo krvi v kanáloch, ktoré sú rádovo väčšie ako rozmery jednej červenej krvinky sa krv modeluje ako homogénna tekutina. Pri skúmaní sedimentácie krvi niektoré modely využívajú prístup v ktorom sú krvinky modelované ako tuhé objekty. V našom modeli sú červené krvinky modelované ako elastické objekty. Majú definovaný relaxovaný tvar a veľkosť, ktorú si chcú zachovať. Vnútro červenej krvinky považujeme za homogénnu tekutinu, a teda celé správanie červenej krvinky je zachytené v správaní jej membrány.

Jednou z výhod tohto prístupu je možnosť modelovať správanie toku takýchto buniek v rôznych prostrediach, napríklad v kanálikoch s veľkosťami porovnateľnými s veľkosťou krvných ciev, alebo v kanálikoch reprezentujúcich časti mikrofluidických čipov. Model je flexibilný a elastické parametre je možné nastaviť tak, aby simulovali správanie nielen zdravých, ale aj chorých červených krviniek.

V krvi sa však nevyskytujú len červené krvinky, ale aj množstvo ďalších objektov, ktorých odhalenie môže slúžiť pre diagnostiku rôznych ochorení. Hlavným rozdielom medzi RBC a inými bunkami je ich komplikovanejšia vnútorná štruktúra. Najvýraznejší rozdiel je, že červené krvinky neobsahujú jadro. V našej práci sme sa zamerali práve na modelovanie buniek s jadrom. Takýto model môže byť následne využitý pre modelovanie cirkulujúcich tumorových buniek, buniek z bunkových línií, bielych krviniek, červených krviniek infikovaných maláriou a iných buniek s jadrami. Pre tento krok je však dôležitá správna kalibrácia a nastavenie parametrov modelu. V práci sa venujeme aj tejto problematike.

1.1 Motivácia

Doposiaľ, pri snahe modelovať cirkulujúce tumorové bunky pomocou našich simulácií, boli tieto bunky reprezentované ako tuhé gule. Našou hlavnou motiváciou pre vytvorenie modelu bunky s jadrom, ktorý bude elastický, bola práve motivácia modelovať cirkulujúce tumorové bunky.

Pri pohľade na štatistiky zverejnené na stránkach svetovej zdravotníckej organizácie (WHO) môžeme vidieť, že rakovina je na svete na druhom mieste v rebríčku príčin úmrtí. V roku 2018 bola rakovina príčinou smrti u 9,6 milióna pacientov. Jedno zo šiestich úmrtí celosvetovo je spôsobená rakovinou. Medzi najčastejšie typy rakoviny patria rakovina pľúc, prsníka, hrubého čreva, kože a žalúdka.¹

Hlavným znakom rakoviny je tvorba abnormálnych rakovinových buniek, ktoré sa rýchlo delia a rastú. V mnohých prípadoch následne zaútočia na okolité tkanivá a rozšíria sa do vzdialených orgánov. Práve tento proces šírenia sa nazýva metastázovanie.

Metastatické ochorenie je hlavnou príčinou morbiditu a mortality u onkologických pacientov [69]. U veľkého percenta týchto pacientov sa cirkulujúce nádorové bunky (CTC) šíria krvným obehom. Približne 30% až 40% pacientov so solídnyimi nádormi má mikrometastázy prítomné v kostnej dreni, krvi alebo lymfatických uzlinách. Avšak v počiatočných štádiách ich šírenia ich nemožno efektívne odhaliť bežnými klinickými metódami [53].

Včasná diagnostika prítomnosti mikrometastatického ochorenia môže viesť ku lepšej prognóze pacientov v dôsledku skoršej a individualizovanej liečby. Jednou z ciest na detekciu CTC v krvi je použitie mikrofluidických zariadení [8]. Lepšie porozumenie vlastností a správania sa CTC buniek môže byť nápomocné pri vývoji mikrofluidických zariadení s vyššími podielmi zachytenia týchto buniek.

Skúmanie biomechanických vlastností, ktoré môžu byť spojené s metastatickým potenciálom buniek sa zameriavalo hlavne na skúmanie vlastností membrán buniek, avšak ukazuje sa, že podstatnú úlohu zohrávajú aj ďalšie zložky buniek [15]. V ďalších častiach práce priblížime biologické pozadie, ktoré nás motivovalo ku vytvoreniu modelu bunky s jadrom.

¹ [<https://www.who.int/news-room/fact-sheets/detail/cancer>]

1.2 Ciele práce

Táto práca je súčasťou dlhodobého vývoja simulačného nástroja výskumnej skupiny Cell in Fluid. Jedným z cieľov výskumnej skupiny je využitie nášho simulačného nástroja pre optimalizáciu a vývoj mikrofluidických zariadení. Zameranie práce je sústredené na rozšírenie nášho modelu, aby mohol byť využitý pre simulovanie viacerých druhov biologických buniek. Stanovili sme si nasledujúce ciele:

Vytvorenie modelu buniek s jadrom a porovnanie rôznych prístupov a ich implementácia do simulačného modelu

Prvým krokom bolo preskúmanie existujúcich modelov buniek s jadrom a určenie možných aplikácií takýchto modelov pri vývoji mikrofluidických zariadení. Pre vytvorenie modelu bunky s jadrom sme využili existujúci model elastickej membrány. Základnou oblasťou, na ktorú sme sa zamerali, bol spôsob akým je jadro naviazané na membránu bunky. Pozreli sme sa na biologické pozadie tejto väzby a následne sme otestovali rôzne prístupy. Zhodnotili sme ich výhody a nevýhody a taktiež sa zamerali na ich využitie.

Návrh spôsobu kalibrácie elastických parametrov modelu bunky s jadrom a nastavenie parametrov modelu pre simuláciu konkrétneho biologického experimentu

V práci sme sa zamerali na vybranú skupinu biologických experimentov, vhodných pre kalibráciu konkrétnych typov biologických buniek. Dôležitým pre ďalšiu využiteľnosť modelu bolo vytvorenie procesu kalibrácie, ktorý bude využiteľný pre širokú škálu buniek. Zamerali sme sa nielen na simulačný aspekt tohto procesu, ale aj na spôsoby získania biologických dát a na výber vhodného mikrofluidického zariadenia. Takýto prístup umožňuje používateľovi modelu získať celkový postup ako získať nakalibrovaný výpočtový model konkrétneho druhu bunky, ktorý potom ďalej môže využívať.

Využitie modelov buniek s jadrom pre odporúčania pre optimalizáciu reálnych mikrofluidických zariadení

Výhodou výpočtového modelu je otestovať vplyv geometrie mikrofluidického zariadenia na správanie toku buniek. Takýmto spôsobom je možné vylepšiť požadované vlastnosti, a tak ušetriť materiál a čas biológom. Počas doktorandského štúdia sa

nám podarilo nadviazať spoluprácu s laboratóriom, kde sme vykonali biologické experimenty s využitím viacerých mikrofluidických zariadení. Podieľali sme sa na vývoji geometrie mikrofluidického zariadenia určeného na skúmanie deformácie individuálnych buniek. Následne sme vytvorili simuláciu tohto zariadenia a určili aké parametre bude potrebné namerať pre nastavenie parametrov modelu bunky s jadrom.

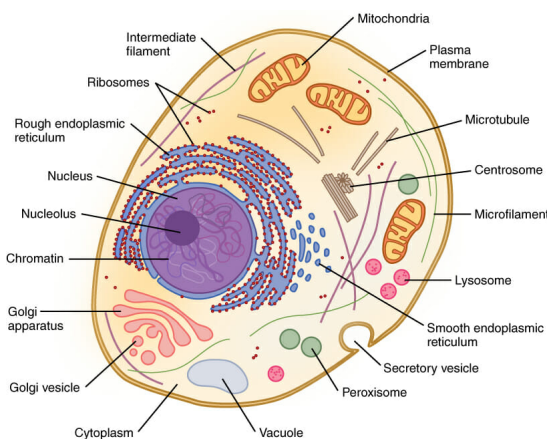
1.3 Prehľad práce

V tejto práci sa venujeme biologickému pozadiu modelu, rôznym spôsobom modelovania používaným v súčasnosti vo výskume buniek s jadrami, a taktiež aj vlastným koncepciám ako model bunky s jadrom navrhnúť. V kapitole 2 sme sa zamerali na popis biologických vlastností buniek s jadrom. Špeciálnu pozornosť sme venovali bunkovým líniam. Následne poskytujeme prehľad mikrofluidických zariadení, ktoré sa využívajú na rôznorodé účely, ako triedenie a zachytávanie odlišných typov buniek, ale aj meranie biofyzikálnych vlastností individuálnych buniek. Kapitola 3 začína prehľadom literatúry o rôznych spôsoboch modelovania buniek. Následne uvádzame popis modelu RBC, vyvíjaného výskumnou skupinou Cell in Fluid, ako aj spôsobu, akým sme sa podieľali na kalibrácii a validácii modelu RBC bunky. Hlavná časť práce sa nachádza v kapitole 4. Tu popisujeme nastavenie simulačného experimentu, pomocou ktorého sme testovali rôzne spôsoby modelovania buniek s jadrom. V častiach 4.2, 4.3 a 4.4 postupne uvádzame detaily jednotlivých prístupov. V časti 4.5 ich porovnávame. Ďalej sa v kapitole 5 venujeme modelovaniu biologických experimentov. Sústredíme sa na dva vybrané experimenty, avšak popisujeme aj zaznamenávanie vlastných biologických experimentov a výzvy spojené so spracovávaním biologických dát. Následne uvádzame možné využitia modelu bunky s jadrom a ďalšie možnosti pokračovania výskumu.

2 Biologické vlastnosti buniek a mikrofluidické zariadenia

2.1 Bunky s jadrom - biologické vlastnosti

Bunky sú základnou štruktúrou, ktorá vytvára všetky živé organizmy. Existuje ich široká škála a nesú rôzne vlastnosti podľa ich využitia. V tejto práci sa zameriavame na bunky s jadrom. Bunky tvorené bunkovou membránou, cytoskeletom, rôznymi organelami, medzi ktoré patrí aj jadro sa nazývajú eukaryotické bunky. Tieto bunky hrajú dôležitú úlohu aj pri procese metastázovania. Celkové správanie bunky závisí od mechanických vlastností jej zložiek.



Obrázok 2.1: Stavba eukaryotickej bunky. Prevzaté z [<https://biologydictionary.net/cell-nucleus/>]

Membrána bunky je tvorená lipidovou dvojvrstvou a oddeľuje bunku od mimobunkového priestoru. Lipidová dvojvrstva je nestlačiteľná, a teda zabezpečuje zachovávanie veľkosti povrchu bunky a taktiež jej odpor voči ohybu. Vnútro bunky je vyplnené okrem už vyššie menovaných organel aj cytosolom (vnútrobunková

2.1. BUNKY S JADROM - BIOLOGICKE VLASTNOSTI

tekutina). Bunková membrána slúži aj na látkovú výmenu, avšak väčšinu času je nepriepustná, a tak zachováva stály objem bunky.

Priestor vnútri bunky je tvorený cytoskeletom. Skladá sa z troch prepojených prvkov - mikrotubúl, mikrofilamentov a stredných filamentov [75]. Sú napojené aj na samotné jadro bunky [15]. Táto dynamická štruktúra zachováva celistvosť bunky a podieľa sa na viacerých bunkových procesoch. Je dôležitá pri pohybe bunky cez tkanivá. Pôsobí proti mechanickému namáhaniu a je podstatná aj pri adhézii. Všetky menované bunkové procesy prebiehajú v bunke aj počas procesu metastázovania. Mikrofilamenty majú najvyššiu odolnosť voči deformácii až do momentu, kým mechanické namáhanie presiahne kritickú hodnotu. Vtedy sa rozpadnú [63]. Stredné mikrofilamenty vedia odolávať deformáciám do nižších hodnôt ako mikrofilamenty. Zabezpečujú celistvosť štruktúry bunky pri šmykovom napätí, keďže sa ani pri vysokých hodnotách tohto napätia nerozpadnú [51]. Mikrotubuly sú prepojené s mikrofilamentami a strednými filamentami, čím stabilizujú cytoskelet. Pri vysokých tlakoch vyvíjaných na bunku sa podieľajú na odpore voči týmto silám [6].

Jadro je najväčšia organela eukaryotickej bunky a je tuhšie ako samotná bunka. Táto informácia bola získaná na základe izolovania samotného jadra z bunky. Je dôležité poznamenať, že jadro v bunke môže mať jemne líšiace sa vlastnosti oproti izolovanému jadru [27]. Celkovo nie je jednoduché získať informácie o mechanike samotného jadra, čo potvrdzuje aj menší počet článkov na túto tému. Ukazuje sa tiež, že z hľadiska deformácie je dôležité nielen jadro, ale aj to akým spôsobom je prepojené s cytoskeletom. Toto prepojenie je kritické najmä pri pohybe bunky v trojrozmernom priestore [22].

Pohyb bunky cez hustý extracelulárny matrix je ovplyvnený minimálne troma vlastnosťami jadra. Medzi tieto patria veľkosť, tuhosť a tvar [82].

V [79] využili nanočastice na sledovanie mikroštruktúr jadra. Zistili, že vnútro jadra je oveľa tuhšie ako cytoplazma a že je viac elastické ako viskózne. Toto poukazuje na to, že vnútro jadra má správanie podobné tuhému objektu.

Rôzne experimenty, ktoré spomíname v časti 2.3.2 ukazujú na to, že za tuhosť jadra zodpovedá jadrová lamina a viskoelastické správanie je pripisované chromatinu vnútri jadra [73]. Jadro je zvyčajne 2- až 10-krát tuhšie ako okolitá cytoplazma [7]. Namerané hodnoty elasticity jadra pomocou Youngovho modulu sa líšia naprieč experimentami od 0,1 do 10 kPa. Toto je najmä z dôvodu rôznorodosti experimentov, kde jedny zanedbávajú vplyv obalu jadra, ďalšie zase zarátavajú aj vlastnosti celej bunky [15].

Mechanické vlastnosti buniek ovplyvňuje aj to, či sa nachádzajú v toku, či prechádzajú bunkovými tkanivami alebo prítomnosť rôznych ochorení v organizme [76]. Jedným z ochorení, ktoré ovplyvňujú mechanické vlastnosti buniek patrí rakovina. S narastajúcim metastatickým potenciálom bunky sa zmäkčuje cytoskelet, a teda aj celá bunka [10]. Ďalšími sú neurodegenerácia, cirhóza pečene, pľúcna fibróza, pľuzgierové kožné ochorenia a viaceré kardiovaskulárnych ochorení [59].

2.2 Bunkové línie

Rôzne štúdie sa zameriavajú na zisťovanie biologických, fyzikálnych a fyziologických vlastností CTC buniek. Je však dôležité myslieť na to, že týchto buniek je pomerne veľké množstvo a líšia sa veľkosťou, tvarom, tuhosťou a mnohými ďalšími vlastnosťami. Taktiež je náročné tieto bunky získať a následne ich udržať v živom stave v laboratórnych podmienkach.

Z týchto dôvodov sa v štúdiách cirkulujúcich rakovinových buniek často používajú **bunkové línie**. Napriek odlišnostiam, oproti bunkám odobratých pacientom, poskytujú pre výskumníkov dobrý model. V článku [74] skúmali 60 rôznych druhov rakovinových bunkových línií. Testovali vyše 70 000 rôznych chemických zlúčenín, medzi ktorými boli aj bežne používané chemoterapeutiká.

Z tohto dôvodu sú častejšie publikované informáciám o bunkových líniách. Pre našu ďalšiu prácu bude prínosné nakalibrovať do modelu vybrané bunkové línie, keďže práve tie, sa často využívajú pri testovaní vyvíjaných mikrofluidických zariadení.

Bunkové línie sú populácie buniek odobraté z viacbunkových organizmov, ktoré by sa za normálnych okolností ďalej nerozmnožovali. Vďaka mutáciám je im ale umožnené deliť sa neustále. To umožňuje pestovať tieto bunky pomerne dlhý čas v in vitro podmienkach. Pôvodne boli bunkové línie vytvárané odizolovaním z rakovinového tkaniva. V súčasnosti je ich však možné vytvárať aj inými biochemickými procesmi, ktoré potlačia prirodzený zánik bunky.

Veľkou výhodou bunkových línií je, že sú cenovo dostupnejšie, poskytujú neobmedzený zdroj materiálu a čisté populácie buniek. Umožňujú tak konzistentnosť testovaných vzoriek, a teda možnosť overiť a zopakovať výsledky experimentov.

2.3 Mikrofluidické zariadenia

V súčasnosti existuje veľké množstvo výskumu v oblasti CTC buniek. Pri vyhľadávaní kľúčového slova "circulating tumor cells" v databáze PubMed nájdeme 7549 článkov za posledných 5 rokov. Napriek tomu, že CTC sa používajú v mnohých klinických štúdiách, sa o nich ešte zďaleka nevie všetko. Výskum CTC buniek nesie množstvo výziev.

Jednou z nich je veľmi nízka koncentrácia CTC buniek. U väčšiny onkologických pacientov sa uvádza množstvo 1 až 10 buniek na 10 mL krvi [3]. Udáva sa, že v 1 mL krvi je okolo 5 miliónov RBC. Rozsiahlou oblasťou výskumu je preto zachytávanie a detekcia CTC. Jednak z dôvodu, že detekcia CTC buniek môže byť použitá ako skorý varovný signál a aj preto, že zachytávanie je dôležité pre skúmanie samotných CTC buniek.

2.3.1 Prehľad triediacich a zachytávajúcich mikrofluidických zariadení

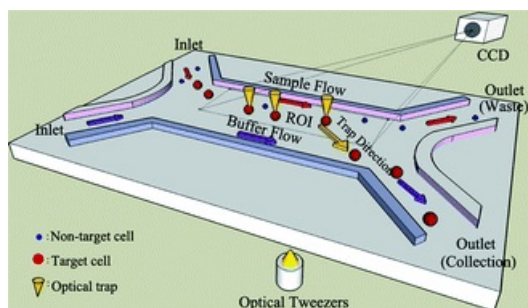
Triedenie a izolácia buniek z komplexných roztokov akým je napríklad krv, je problém, ktorým sa zaoberá biológia a jej príbuzné vedy už pomerne dlho. Prvý komerčný bunkový triedič bol vynájdený v roku 1969 Herzenbergerom [39]. Napriek veľkým pokrokom v tejto oblasti sú však existujúce prístroje stále limitované. Je potrebné ich vylepšiť. Žiadaná je vyššia rýchlosť triedenia, lepšia presnosť, schopnosť spracovať prirodzené biologické vzorky (bez nutnosti zdĺhavej predprípravy vzoriek), schopnosť rozlišovať rôznorodé bunky, triediť viacero typov buniek súčasne, zjednodušenie a zautomatizovanie celkového procesu, zníženie nákladov a zmenšenie veľkosti zariadení [71].

Odpoveď na všetky tieto požiadavky hľadajú výskumníci v mikrofluidických zariadeniach. Ich hlavnou výhodou je, že vedia poskytnúť priestorovú a časovú kontrolu v mikro rozmeroch. Vytváraním matematických modelov týchto zariadení je možné otestovať rôzne dizajny rýchlejšie a oveľa lacnejšie, ako vyrábať každý jeden fyzicky a testovať ho v laboratóriu.

Mikrofluidické čipy sú vo všeobecnosti zložené z jedného alebo viacerých mikrokanálikov vymodelovaných z rôznych druhov materiálov (sklo, silikón, rôzne polyméry ako napríklad PDMS). Na základe techniky použitej na označenie buniek je možné rozdeliť rôzne technológie do troch kategórií: fluorescenčné označovanie, označovanie

pomocou guľôčok, bez označenia [71].

Fluorescenčné označovanie využíva rôzne chemické látky pre rozlišovanie typov buniek. Medzi tieto techniky patria triedenia zakladajúce sa na rôznych mechanizmoch. Jedným z nich je využitie elektrokinetických mechanizmov. Elektrické pole vytvorené v mikrofluidických zariadeniach spôsobuje pohyb buniek, a tým aj ich triedenie. K tomuto deju dochádza, či už sa využíva, že samotné bunky sú čiastočne negatívne nabité a pohybujú sa smerom k pozitívne nabitej elektróde, alebo sa pohybujú vďaka vytvorenému rovnomernému elektrickému poľu, alebo sa do tekutiny vpustia rozpustné ióny. Viac popísané sú tieto techniky v článku [80]. Ďalšia technika využíva akustické tlakové vlny. Pomocou nich je možné upraviť pohyb buniek v mikrofluidických zariadeniach, ako napríklad ich nasmerovať dobre definovaným tokom [48]. Ďalším spôsobom ako dobre manipulovať bunkami je optická manipulácia. Pomocou nasmerovania optických laserov je možné nielen ich presúvať na konkrétne miesta v zariadení, ale aj napríklad ich zachytiť a nechať na ne pôsobiť silu samotného prúdu tekutiny [81].



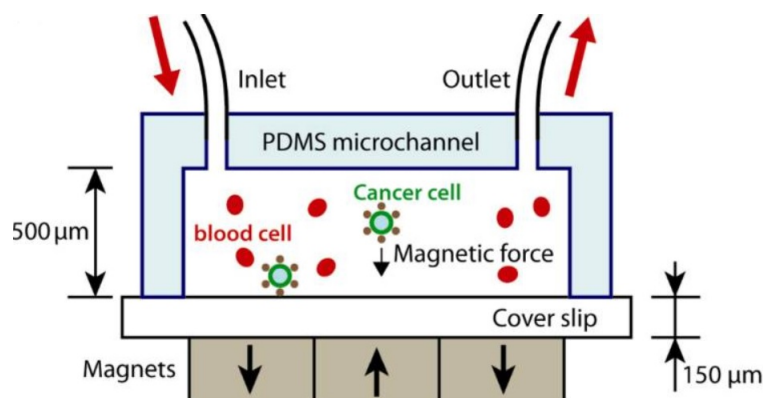
Obrázok 2.2: Mikrofluidické zariadenie využívajúce optický laser pre zachytávanie CTC buniek. Obrázok prevzatý z [81].

V mikrofluidických zariadeniach sa používajú aj mechanické systémy. V nich sa využíva samotný tok, a možnosť jeho prispôbenia pomocou rôznych dizajnov mikrofluidických zariadení [45].

Triedenie buniek založené na použití guľôčok využíva špecifické vlastnosti materiálov, z ktorých sú tieto guľôčky vyrobené. Jednak ich príľnavosť k rôznym typom buniek ale aj ich tvar a veľkosť. Pri týchto technikách nie je potrebné manipulovať s jednotlivými bunkami, a preto je možné rýchlejšie roztriedenie alebo zachytenie buniek, ako pri technikách fluorescenčného označovania. Pripojenie guľôčok na bunky umožňuje aj viac spôsobov manipulácie s nimi, ako napríklad možnosť aplikácie väčších síl. Jednou z techník využívajúcich guľôčky je magnetophoresis,

2.3. MIKROFLUIDICKÉ ZARIADENIA

ktorá využíva trvalé magnety alebo elektromagnetické cievky, aby pôsobili silami na bunky označené magnetickými časticami. Jednu z pomerne úspešných techník na zachytávanie CTC pomocou takýchto guľôčok vyvinul Hoshino. [36]. V ich systéme boli CTC bunky označené nanočasticami oxidu železitého, následne boli oddelené a nakoniec pomocou iných chemických látok bolo overené, že naozaj išlo o bunky pochádzajúce z tumorov. Využitie akustických vln môže byť vylepšené použitím



Obrázok 2.3: Mikrofluidické zariadenie využívajúce magnetické guľôčky pre zachytávanie CTC buniek. Obrázok prevzatý z [36].

elastomérových častíc. Tie sa, na rozdiel od buniek, zlučujú pozdĺž antinód akustickej stojacej vlny. Preto pri použití takýchto častíc, ktoré sú ešte ošetrené látkami priťahujúcimi cieľové bunky je možné zachytiť cieľové bunky tiež na antinódy, a tým ich odfiltrovať od buniek, ktoré nás nezaujímajú. Takýto systém je popísaný v článku [70].

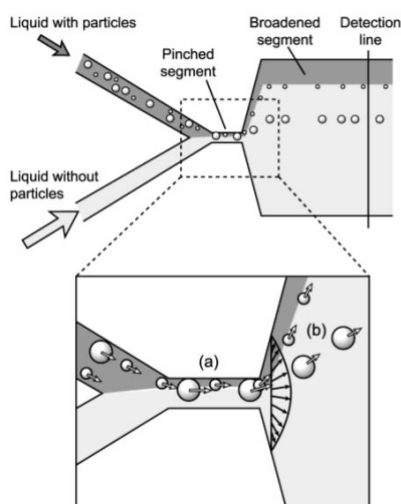
Ďalšia z techník využíva elektrické sily, ktorých pôsobenie je závislé od veľkosti buniek, čo umožňuje nesequenčné triedenie. Avšak mnoho buniek má podobnú veľkosť, a práve tu môžu pomôcť guľôčky naviazané na konkrétny druh buniek. Tie ich veľkostne jasnejšie odlišia od ostatných neoznačených buniek. Konkrétnym príkladom takéhoto využitia je systém z článku [38], kde bunky označené guľôčkami prešli ďalej k odlišnému otvoru ako neoznačené bunky.

Poslednou skupinou techník pre zachytávanie a triedenie sú mikrofluidické zariadenia s **technikami bez označovania buniek**. Základom pre všetky tieto techniky je využívanie iba fyzikálnych vlastností buniek, ako je ich veľkosť, tvar, hustota, elasticita a ich magnetická citlivosť. Keďže tieto techniky si vyžadujú zvyčajne najmenej prípravy z pomedzi všetkých techník sú pomerne populárne.

Týchto techník je pomerne veľké množstvo, spomeniem dve najzaujímavejšie z

hľadiska možnosti využitia nášho modelu. Obidve spadajú ku technikám pasívneho triedenia buniek. Tieto metódy sa spoliehajú na rozdiely v bunkovej morfológii (napríklad veľkosť, tvar, stlačiteľnosť a hustota) a pre triedenie využívajú silu samotného toku, filtráciu a adhéziu samotných buniek.

Triedenie na základe zúženia toku sa využíva v článku [85]. Ako vidíme na obrázku, 2.4 v zúženej časti kanála sú bunky zarovnané pozdĺž jednej zo stien bez ohľadu na to akú majú veľkosť. To je spôsobené rôznou rýchlosťou tekutín v dvoch vstupoch. Pomer rýchlostí týchto tekutín určuje šírku pásu tekutiny, v ktorej sa bunky v zúžení zoradia. Po tom, čo sa kanál opäť rozšíri, bunky sú roztriedené podľa svojej veľkosti na základe profilu ich rýchlosti v momente, keď vystupujú z kanála. Ďalšími faktormi, ktoré ovplyvňujú takéto triedenie, je geometria mikrokanálu, aké široké a dlhé sú jeho jednotlivé časti a pod akým uhlom bunky vstupujú a vystupujú zo zúženia.

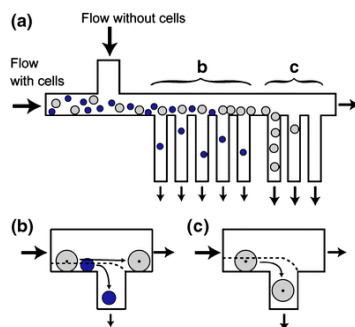


Obrázok 2.4: Mikrofluidické zariadenie využívajúce zúženie toku pre triedenie CTC buniek. Obrázok prevzatý z [85].

Ďalšia technika, nazývaná hydrodynamická filtrácia [86], je tiež vyvinutá autorom predošlého článku. Táto metóda využíva pohyb bunky v rámci toku. Konkrétne využíva, že vzdialenosť stredu bunky od steny mikrokanálu je vždy aspoň jej polomer. Pri tejto metóde sa využívajú mikrokanáliky, ktoré majú viaceré vedľajšie mikrokanáliky, ako môžeme vidieť na obrázku 2.5. Tým, že z hlavného toku cez tieto odbočky odteká časť tekutiny, sú bunky zoradené pri stenách kanála. Na základe ich profilu rýchlosti, ktorý závisí od polohy ich stredu buď do odbočky vojdú, alebo

2.3. MIKROFLUIDICKÉ ZARIADENIA

pokračujú ďalej. Tento princíp Yamada otestoval na rôznych pečňových bunkách [84].



Obrázok 2.5: Mikrofluidické zariadenie využívajúce odbočujúce kanáliky pre triedenie pečňových buniek. Obrázok prevzatý z [84].

2.3.2 Experimenty merajúce biofyzikálne vlastnosti buniek

V tejto časti zhrnieme niektoré z experimentov, pomocou ktorých sa merajú biofyzikálne vlastnosti buniek.

Aspirácia mikropipetou

Táto technika využíva nasávanie bunky do malej sklenenej trubice. Namerané elastické moduly získané takýmto experimentom nám poskytujú dobrú informáciu o celkovej (globálnej) mechanike bunky. Avšak vzhľadom na to, že celá bunka je aspirovaná, môže mať skutočnosť, že CTC má jadro, silný vplyv na výsledky. Táto metóda tak nie je ideálnou, ak by sme ju chceli využiť pre kalibráciu správania iba samotnej membrány, nakoľko by to mohlo viesť k nezrovnalostiam [28]. Dôležité je poznamenať, že aj v článku, kde takúto aspiráciu modelovali [16] zhodnotili že výpočtová náročnosť je príliš veľká. Pre verné zobrazenie priebehu deformácie bunky pri takomto experimente použili diskretizácie bunky s vyše 6000 povrchovými bodmi. Táto technika bola použitá aj na meranie vlastností jadra [12, 27, 11, 65, 34].

Magnetické / optické pinzety

Aplikovaním síl na konkrétne miesta povrchu bunky a na cytoplazmu môžeme získať informácie o schopnostiach buniek natahovať sa. Optické pinzety majú širokú škálu použitia, sily väčšie ako 100 pN sa môžu aplikovať na bunky o veľkostiach od nanometrov po mikróny. Bunky sú zachytené v optickej pasci vytvorenej laserom.

Priestorový posun bunky v takejto pasci je zaznamenaný s vysokou presnosťou [88]. Použitie magnetických pinziat je veľmi podobné používaniu optickej pasce. Avšak, namiesto optického zachytávača sa k povrchu bunky pripojí dvojica magnetických guľôčok. Sila pôsobiaca na guľôčky závisí od sily vonkajšieho magnetického poľa. V tomto prípade môžu byť na častice mikrónových rozmerov aplikované sily väčšie ako 200 pN. Tento typ deformácie buniek spôsobený danými silami v daných miestach buniek môže byť v našom modeli nasimulovaný, a tak nám môže poskytnúť informácie o elasticite modelu bunky v porovnaní so skutočnou bunkou [56, 24].

Metóda indentácie pomocou atómovej silovej mikroskopie (AFM)

Atómová silová mikroskopia je metóda, ktorá môže poskytnúť podrobnejší pohľad na správanie povrchu bunky. Pri veľmi presných meraniach môžeme vytvoriť 3D zobrazenie objektu, ktorý študujeme. AFM sa môže tiež použiť na určenie tuhosti týchto materiálov. Pomocou presnej sily môže AFM merať odpor proti hrotu pri tlačení na bunku. V článku [31] sa AFM používa na stanovenie tuhosti rakovinových buniek s použitím rovnice $F = a[\exp(b\delta) - 1]$, kde F je sila, ktorou hrot tlačí na povrch bunky a δ je ryha spôsobená touto silou, a a a b sú parametre určujúce tuhosť bunky. Pri ryhách s hĺbkou menšou ako $150\mu m$ je vzťah medzi $dF/d\delta$ a δ lineárny, a teda vedeli vypočítať Youngov modul pružnosti (E), ktorý je odvodený od ich vzorca tuhosti $dF/d\delta = 0.0686E\delta$. Na základe ich meraní zistili, že rakovinové bunky sú mäkšie ako normálne bunky [46].

Naťahovanie pomocou mikrodoštičiek

Bunka sa upevní na dve chemicky upravené sklenené mikrodoštičky. Tieto sú potom odťahované od seba iba v rámci jednej priamky a zaznamená sa ich posun. Táto metóda môže byť použitá na porovnanie elasticity modelu a aktuálnej bunky nastavením simulácie napodobňujúcej proces takéhoto naťahovania a následným porovnaním správania simulácie a bunky z laboratórneho experimentu [76].

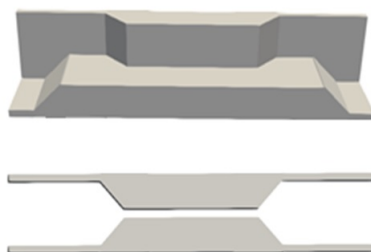
Všetky predchádzajúce techniky majú vysokú kvalitu merania s veľkou presnosťou, ale ich časová náročnosť je pomerne vysoká a zakaždým je testovaná len individuálna bunka. Mikrofluidické zariadenia ponúkajú naopak rýchlejší priebeh samotných experimentov a často aj možnosť otestovať viacero buniek za sebou, bez nutnosti nanovo začínať celý proces. Tieto výhody však prinesú miernu stratu presnosti informácie. Na meranie biofyzikálnych vlastností sú často používané zariadenia s

2.3. MIKROFLUIDICKÉ ZARIADENIA

úzky kanálmi a prekážkami, kde je možné merať prechodné časy a deformačné koeficienty. V úzkych kanáloch sa bunka deformuje nie len vtedy, ak je kanál menší ako samotná bunka, ale aj v dôsledku samotného prúdenia tekutiny.

Mikrofluidické zariadenie s úzkym kanálom

V článku [37], bolo cieľom nájsť rozdiely medzi zhubnými a nezhubnými bunkami. Mikrofluidické zariadenie vytvorené pre tento účel pozostáva z priameho kanála a zásobníka na každom konci. Úzka štrbina tohto kanála má dĺžku $150\mu\text{m}$ a jej prierez je štvorec s veľkosťou $10 \times 10\mu\text{m}$. Tvar kanála môžeme vidieť na obrázku 2.6. V experimente použili bunkovú líniu MCF-10A (tá predstavuje zdravé bunky pre



Obrázok 2.6: Mikrofluidický kanál so zúžením

porovnanie rozdielov) a líniu MCF-7 (táto predstavuje rakovinovú bunku). Rozdiel medzi týmito dvoma bunkami bol meraný tromi parametrami: **vstupný čas** - čas potrebný na to, aby bunka kompletne vstúpila do zúženej časti, **čas prechodu** - čas od momentu keď bola bunka kompletne vnútri, až kým kompletne celá nevyšla von a **index predĺženia**. Ten je definovaný ako $EI = D_L/D$, kde D_L je dĺžkou deformovanej bunky vnútri zúženej časti a D je dĺžkou počiatočného priemeru bunky pred vstupom do úzkeho kanála. Najvýraznejší rozdiel medzi rakovinovými a nerakovinovými bunkami zaznamenali v hodnotách vstupného času.

Tento konkrétny experiment sme popísali do väčšieho detailu, pretože sme na jeho základe vytvorili simuláciu. Bližšie je popísaná v časti 5.2. Ďalšie experimenty, kde boli využité mikrofluidické zariadenia pre skúmanie fyzikálnych vlastností sú: [66], kde sledovali aký vplyv na prechod zariadením má obal jadra neutrofilu (najčastejší typ bielej krvinky). V [14] navrhli kanálik s veľmi malými zúženiami 2 až $5\mu\text{m}$, ktoré malo simulovať prostredie, ktorým musia bunky migrovať. Zaznamenali, že pohyb buniek cez takéto zúženia trval niekoľko hodín.

Naťahovanie optickým laserom

Článok [26] rozoberá možnosti využitia tuhosti buniek ako rozlišovacieho znaku, ktorý možno využiť pre testovanie zhubnej transformácie buniek. Pri experimente používajú dva laserové lúče na vytvorenie optickej pasce, ktorá deformuje jednotlivé bunky. Testovali viacero tipov buniek ako myšie fibroblasty a ľudské epiteliálne bunky z normálneho až rakovinového a metastatického stavu. Použili imortalizované bunkové línie (BALB / 3T3, SV-T2, MCF-7, MCF-10, MDA-MB-231). Pre nás je najzaujímavejšia MCF-7 bunková línia. Sila každého z použitých laserov na zachytenie buniek bola $P = 100 \text{ mW}$ a pre natiahnutie buniek sa zvýšila na $P = 600 \text{ mW}$, pričom konce laserových vlákien boli od seba vzdialené $130 \mu\text{m}$. Priemer optickej pasce mal pri koncoch laserové vlákna dĺžku $6,2 \mu\text{m}$, a v strede, kde bola bunka bol priemer pasce $12 \mu\text{m}$. Táto metóda je pomerne rýchla a poskytuje dostatočné množstvo údajov pre zmysluplné štatistiky o testovaných bunkách. Záverom ich meraní bolo, že vedľa pomerne z malého množstva buniek rozlišovať medzi zhubnými a nezhubnými bunkami a tiež, že rakovinové bunky MCF-7 sa deformovali viac (30%) v porovnaní s normálnymi bunkami MCF-10 (10%).

3 Modelovanie buniek

3.1 Rôzne prístupy k modelovaniu buniek

Modelovanie buniek s jadrom je pomerne komplexný problém, ktorým sa v posledných rokoch začalo zaoberať viac výskumníkov. Dôvodom je najmä veľká rôznorodosť medzi veľkosťami a tvarmi rôznych buniek a ich jadier. Získať informácie o mechanických vlastnostiach takýchto buniek je náročné, keďže výskumný zámer biológov a tých, čo robia simulácie je často rôzny. Simulačné modely môžu pomôcť aj s odpoveďami na niektoré otázky o vplyve konkrétnych súčastí buniek na celkovú tuhosť buniek.

V tejto časti ponúkame výber z rôznych prístupov k modelovaniu buniek dostupných v literatúre. Zamerali sme sa na informácie týkajúce sa hlavne spôsobov modelovania buniek s jadrami, ako sú napríklad CTC alebo WBC. Do prehľadu sme nezahrnuli modely zaoberajúce sa bunkovými tkanivami alebo bunkami, v ktorých sa nachádza viacero jadier. Našu pozornosť sme upriamili na diskkrétne metódy, keďže aj model výskumnej skupiny Cell in Fluid spadá do tejto kategórie.

Jedna spoločná vlastnosť, ktorú majú všetky prístupy k modelovaniu buniek je, že všetky sa pozerajú na deformáciu buniek na makroskopickom stupni. Existuje aj druhý spôsob, akým by bolo možné tieto modely vytvárať, a to pozrieť sa na mechanické vlastnosti najmenších zložiek buniek a postupne z nich vybudovať model až na úroveň samotnej bunky. Vybudovať takýto model má v sebe však niekoľko prekážok. Jednou je veľké množstvo výpočtovej sily, ktorá by bola potrebná a druhou je fakt, že zatiaľ neexistuje dostatočné množstvo biologických experimentov, ktoré by poskytli potrebné teoretické vedomosti pre vytvorenie takéhoto modelu [51].

Prístupy k modelovaniu buniek sa dajú rozdeliť do viacerých skupín podľa rôznych kritérií. Existujú 2D a 3D modely. Dôvodom pre toto rozdelenie je, že hoci sú bunky 3D objekty, výstupy z biologických experimentov sú väčšinou videozáznamy z mikro-

3.1. RÔZNE PRÍSTUPY K MODELOVANIU BUNIEK

fluidických zariadení, ktoré sú pod mikroskopom pripojeným na vysoko rýchlostnú kameru. Záznam je v 2D, a teda odtiaľ pochádza motivácia modelovať v 2D. Ďalším kritériom pre rozdelenie je miera presnosti, ktorá sa od modelu vyžaduje. Použitie konkrétneho prístupu modelovania závisí aj od výskumnej otázky. Niektoré modely sa zameriavajú do detailu molekulárnou štruktúrou jednotlivých častí bunky, ako napríklad v [4], aby mohli odpovedať na otázku, či sa zmena na úrovni nukleotidov prejaví na správaní aktínových mikrofilamentov a následne tak ovplyvní celkové správanie bunky. Ďalším faktorom je aj, akú výpočtovú silu majú výskumníci k dispozícii. Podľa toho musia prispôsobiť spôsob akým získavajú odpovede na svoje otázky. V prípade, že má niekto superpočítač [64], nebude sa zaoberať nutnosťou paralelizácie, a zase naopak v prípade nižšej výpočtovej sily je nutné optimalizovať možnosti simulačného nástroja.

Jednotlivé prístupy používajú rôzne kombinácie metód pre modelovanie buniek, tekutiny a ich vzájomnej interakcie. Tieto prístupy vieme rozdeliť do troch skupín: mesh-based, particle-based a prístupy, ktoré sú ich kombináciou. Pri mesh-based prístupe sú objekty modelované pomocou povrchových bodov jednotlivých objektov, ktoré sú pospájané a vytvárajú rôzne typy sietí, ktoré sú často trojuholníkové. Tento prístup budeme označovať MB. Pri particle-based prístupe sú objekty modelované pomocou molekúl, ktoré zoskupujú viacero atómov, pričom množstvo atómov v molekule sa líši v závislosti od požadovaného stupňa detailu. Tento prístup označíme PB. Každá z týchto kategórií má niekoľko podkategórií. Medzi MB metódy patria metóda konečných prvkov (FEM - Finite Element Method), vnorená metóda konečných prvkov (IFEM - Immersed Finite Element Method) a metóda konečného objemu (VOF - Volume of Fluid)[32]. Medzi PB metódy patria SPH - Smoothed Particle Hydrodynamics a DPD - Dissipative Particle Dynamics. DPD zlučuje viaceré molekuly do častíc, ktoré sa pohybujú spojitým priestorom [54]. Pri SPH sú tekutiny reprezentované diskretnými časticami a vlastnosti častíc sú potom vyhladené funkciou jadra nad časticami v určitom polomere [35]. Pri použití PB metód sa tekutina väčšinou tiež modeluje týmto spôsobom. V prístupoch, kde modelujú bunky pomocou MB metód sa pre modelovanie tekutiny často používa Lattice Boltzmann metóda (LBM)[1]. Priestor, ktorým sa tekutina pohybuje je rozdelený mriežkou diskretných bodov. Častice tekutiny cez ňu putujú, a tak posúvajú informácie o počte a rýchlosti častíc prechádzajúcich cez každý z diskretných bodov a tie ich následne uchováva. V kombinácii s LBM metódou sa pre vzájomnú interakciu medzi tekutinou a bunkami často používa metóda vnorenej hranice (IBM

- Immersed Boundary Method) [57]. Nevýhodou PB prístupu je, že je výpočtovo náročnejší v porovnaní s MB metódami. Dôvodom je potreba veľkého počtu častíc pre dosiahnutie modelovania porovnateľnej presnosti s MB metódami. Jeho výhodou je použitie pre komplexné štruktúry [87].

Prvým krokom k modelu bunky s jadrom bolo u mnohých vytvorenie modelu RBC, ktorej štruktúra je jednoduchšia ako iných buniek keďže nemá ani jadro, ani cytoskelet. Prístup niektorých spočíval v úprave parametrov modelu RBC, no neobsahoval pridanie ďalšej vnútornej štruktúry. V [29] využili takto modelovanú CTC bunku pre modelovanie jej správania pri prechode filtračným systémom využívajúcim samoorganizujúce sa magnetické guľôčky. Bunka bola modelovaná pomocou pružinového MB prístupu, tekutina pomocou LBM metódy a interakcia medzi tekutinou a bunkami pomocou metódy vnorených hraníc. V [49] sa zase pozerali na deformáciu buniek pri prechode rôznym typom filtračných geometrií. V ich simuláciách modelovali CTC a WBC ako kvapky tekutiny držiace si svoj tvar vďaka povrchovému napätiu a kontaktnému uhlu. Pre simulácie tekutiny využili komerčný systém ANSYS Fluent a na modelovanie buniek použili metódu VOF.

Nasledujúce práce sa zaoberali skúmaním buniek predovšetkým v toku a modelovali bunky s jadrom pomocou upravenia elastických vlastností membrány. V [77] sledovali tok RBC a CTC v mikrocievach. Pre modelovanie CTC využili FEM. Modelovali ich ako guľovité kapsule. Tekutinu modelovali pomocou LBM metódy a pre interakciu medzi tekutinou a bunkou použili IBM metódu. V [21] sa zaoberali margináciou WBC. Tieto tiež patria medzi bunky s jadrom. Pre ich modelovanie využili model, ktorý použili aj pre RBC. Používajú mezoskopickú PB techniku, nazývanú tiež smoothed dissipative particle dynamics (metóda kombinujúca SHP a DPD metódu). V článku [83] skúmali adhéziu CTC na steny mikrociev. Použili DPD metódu. Membrána bunky bola modelovaná pomocou strunovej siete. Adhézia CTC buniek bola získaná vytváraním spojení medzi stenou a povrchom bunky. Podľa [42] je možné CTC a WBC modelovať podobným spôsobom, keďže majú podobné správanie, čo sa týka marginácie a príľnavosti. CTC modelovali pomocou IFEM. Jedná sa o rovnaký spôsob, aký použili na modelovanie všetkých typov buniek v simulácii.

Pridanie jadra a iných vnútorných štruktúr bunky bolo zrealizované viacerými spôsobmi do rôznej miery presnosti. Biologické pozadie pre jednotlivé vnútorné štruktúry popisujeme v časti 2.1. V [61] vytvorili 2D model pre skúmanie správania cirkulujúcich buniek v toku a ich následnej adhézie k povrchu mikrociev, v ktorých sa pohybujú. Bunky boli modelované pomocou IBCell[60]. IBCell používa

3.1. RÔZNE PRÍSTUPY K MODELOVANIU BUNIEK

metódu vnorených hraníc (IB). Na bunku pôsobí tok, ale aj odpudivo-priťahujúce sily medzi bunkou a stenou mikrocievy. Bunka sa skladá z jadra a bunkovej membrány (bunkový aktínový kortex). Obidve tieto štruktúry sú modelované pomocou hustej siete pružín. Endoteliálna stena (mikrocievy) je tiež modelovaná pomocou pomerne tuhých lineárnych pružín. Medzi bunkou a stenou prebieha interakcia za pomoci vytvorenia spojov medzi nimi v prípade, že sa bunka nachádza v malej vzdialenosti od steny. V [89] modelovali krvné doštičky vo veľkom detaile a využili na to prepojenie DPD a CGMD. DPD bola použitá pre modelovanie viskózneho toku a CGMD bola použitá pre modelovanie membrány, cytoskeletu a cytoplazmy krvnej doštičky. Tieto dve metódy sú na rôznych škálach. Prechod medzi rôznymi priestorovými jednotkami je zabezpečený pomocou rozhrania hybridného silového poľa na membráne bunky. Väčším problémom je prechod z nanosekúnd na mikrosekundy, ktorý predstavuje možné problémy tohto prístupu z pohľadu výpočtových možností. V [50] popisujú mezoskopický model eukaryotickej bunky, ktorý pozostáva z bunkovej membrány, jadra a cytoskeletu. Ide o DPD metódu. Model bol kalibrovaný pre bunkové línie tkaniva prsníka MCF-7 a MCF10A. Pre kalibráciu elastických parametrov použili mikropipetovú aspiráciu a na overenie použili mikrofluidické experimenty v toku. Ich model využívali na skúmanie vplyvu jednotlivých zložiek bunky na celkové viskoelastické vlastnosti bunky počas biologických experimentov. V [20] skúmali, akým spôsobom sa mení vnútorná bunková štruktúra počas procesu adhézie k mimobunkovému prostrediu za prítomnosti pôsobenia trakčných síl. Vytvorili model bunky skladajúci sa zo základných zložiek eukaryotickej bunky, ktoré sme popísali v časti 2.1. Obsahuje kortikálny cytoskelet, obal jadra, mikrotubuly, aktínové mikrofilamenty a stredné filamenty. Cytoskelet a obal jadra boli modelované pomocou triangulácie ich povrchu (bolo použitých 1806 vrcholov). Guľa predstavujúca cytoskelet mala polomer $10\mu\text{m}$ a jadro malo tretinový polomer. Aktínové filamenty boli rovné filamenty vygenerované náhodne tak, aby ich konce boli zakotvené vo vrcholoch cytoskeletu. Stredné filamenty boli vytvorené tak, aby spájali príslušný bod z obalu jadra s bodom na cytoskelete. Mikrotubuly boli dynamicky vytvárané pre stabilizáciu bunky počas adhézie k substrátu.

V týchto štúdiách používali rôzne veľkosti pre jednotlivé typy buniek. Voľba veľkosti bola niekedy podložená biologickými meraniami, a niekedy len zvolená s ohľadom na skúmaný fenomén. Dôležité je poznamenať, že aj namerané veľkosti biologických buniek majú pomerne veľkú variáciu. Niektoré práce spomínajú proces kalibrácie, ale v mnohých boli vytvorené tuhšie a mäkšie typy buniek bez biolo-

gického porovnania, s cieľom porovnať vplyv či už na tok, alebo margináciu rôznych typov buniek v simulačnom prostredí. Ak bol vykonaný kalibračný proces, je zameraný na konkrétny typ buniek kvôli veľkej variancii medzi vlastnosťami rôznych typov buniek. V štúdiách, kde boli elastické parametre kalibrované, ide o kalibráciu parametrov pre konkrétny typ buniek. Proces kalibrácie nie je širšie popísaný ani v jednej z prác. Dôležité je tiež poznamenať, že keď je nájdená sada elastických parametrov pre konkrétnu bunku, neznamená to, že táto sada je jedinečná a najlepšia.

Každý z prístupov má svoje výhody a nevýhody. Dôležité je zakaždým zvážiť otázku, na ktorú chceme modelovaním odpovedať. V prípade simulácií s množstvom buniek, kde je cieľom skúmať správanie rôznych buniek v toku a ich prípadnú separáciu, môže byť výhodné použiť jednoduchšie modely, ktoré sú výpočtovo menej náročné. Naopak, ak je cieľom pozrieť sa na veľké deformácie individuálnych buniek, ako napríklad v [58], tak je potrebné ísť do väčšieho detailu.

Náš prístup k modelovaniu elastických objektov v tekutine využíva strunový model založený na MB metóde. Pre modelovanie tekutiny využívame LBM metódu a pre interakciu medzi tekutinou a bunkami IBM metódu.

3.2 Popis modelu RBC

Základným kameňom modelu pre bunky s jadrom bude model elastickej membrány vytvorený výskumnou skupinou Cell in Fluid. Doposiaľ bol využívaný najmä pre modelovanie RBC buniek. Podrobne je všetko popísané v knihe [9]. V tejto časti len zhrnieme dôležité informácie pre ďalšie časti tejto práce.

V modeli je bunková membrána reprezentovaná pomocou triangulácie jej povrchu. Jej elastické správanie zabezpečuje päť elastických modulov.

Modul pružnosti, značený k_s , popisuje tuhosť bunky. V momente, keď je časť membrány naťahovaná, tento modul zabezpečuje jej snahu zabrániť tejto zmene. Môžeme si predstaviť, že hrany našej triangulácie sú pružiny, ktoré si chcú zachovať svoju pokojovú dĺžku.

Modul ohybu, značený k_b , sa tiež podieľa na zachovaní tvaru bunky. Predstavuje však inú zložku, jeho mechanizmus zodpovedá za zachovanie uhlov medzi susednými trojuholníkmi, čím si bunka zachováva svoj relaxovaný tvar.

Modul zachovania lokálnej plochy, značený k_{al} , už podľa názvu napovedá, že jeho úlohou je zachovať plochu jednotlivých trojuholníkov v našej triangulácii. Tento modul do istej miery vykonáva podobné mechanizmy ako modul pružnosti.

Modul zachovania celkovej plochy, značený k_{ag} , patrí medzi moduly zabezpečujúce globálne vlastnosti bunky. Bunka si v reálnom svete zachováva svoj povrch, čiže tento modul nekopíruje žiadny merateľný mechanizmus bunky na rozdiel od predošlých modulov, kde sa dá pozorovať a merať napríklad rýchlosť vrátenia sa do pôvodného stavu pri pôsobení rôznych síl.

Posledným modulom je **modul zachovania objemu**, značený k_v , ktorý zodpovedá za to, že bunka si udržiava svoj konštantný objem. Opäť, tento globálny modul nie je špecificky viazaný na konkrétny biofyzikálny mechanizmus. Bunka si v realite udržiava viac-menej stály objem a ku väčším zmenám dochádza len vo veľmi špecifických a ojedinelých situáciách.

Pri vytváraní postupu pre kalibráciu buniek s jadrom sme sa najprv podieľali na opakovanej kalibrácii RBC buniek. Podrobnejšie popisujeme proces kalibrácie RBC bunky a jej následnej validácie v časti 3.3.

Na vytvorenie požadovanej geometrie objektov je možné použiť viacero techník [33]. Existujú voľne dostupné nástroje pre vytváranie rôznych typov sietí. Medzi ne patrí aj Gmsh, ktorý bol použitý na vytvorenie rôznych diskretizácií gúľ použitých v našich simuláciách [23]. Ďalšie použité diskretizácie boli vytvorené v rámci bakalárskej práce [55].

3.2.1 ESPResSo a PyOIF

Všetky výpočty prezentované v tejto práci boli vytvorené pomocou Pythonovského modulu PyOIF v ESPResSe. ESPResSo je open-source software, ktorý sa využíva na rôznorodé simulácie mäkkých látok. Zameriava sa hlavne na molekulárnu dynamiku, avšak vie efektívne simulovať aj mikroskopické elastické objekty. Naša výskumná skupina implementovala do ESPResSa softwareový balík Object-in-Fluid. Výpočtový modul je naprogramovaný v C++. Pre spúšťanie simulácií, ukladanie a spracovanie údajov ako aj pre vykresľovanie používame Python skriptovacie rozhranie

[9].¹

3.2.2 Výpočtové prostriedky

Na začiatkové testovanie skriptov a spracovávanie dát zo simulácií bol použitý osobný notebook. Na zvyšné výpočty bol použitý Snorlax a Klaster ZU.

Snorlax

Počítač bol financovaný v rámci projektu Slovenskej agentúry na podporu výskumu a vývoja (projekt číslo APVV-15-0751).

Procesory: AMD Threadripper 2950X - 16/32T-Core, 3.5GHz (180W), Boost 4.4GHz

RAM: 32 GB

SSD: 1 TB NVME

HDD: 8 TB

GPU: GTX 1080 Ti

Klaster ZU

Klaster bol financovaný v rámci projektu „Slovenská infraštruktúra pre vysokovýkonné počítanie“ (ITMS 26210120002).

46 × výpočtových uzlov

Procesory: 2 × 6 jadrový Intel(R) Xeon(R) CPU L5640 @ 2.27GHz

RAM: 96GB

HDD: 2 × 500GB

2 × výpočtové uzly

Procesory: 2 × 10 jadrový Intel(R) Xeon(R) CPU E7- 8860 @ 2.27GHz

RAM: 256GB

HDD: 2 × 300GB

+ 2 × grafická karta NVIDIA GF100 [Tesla M2070]

3.3 Kalibrácia RBC

Naše prvé kroky k vytvoreniu kalibračného postupu pre bunku s jadrami viedli cez kalibráciu a validáciu modelu RBC. Niektoré elastické moduly modelu bunky súvisia s merateľnými hodnotami, ktoré sú zaznamenávané pri biologických experimentoch.

¹ [<http://cellinfluid.fri.uniza.sk/resources-espresso/>]

3.3. KALIBRÁCIA RBC

Interakcia medzi jednotlivými zložkami bunkovej membrány môže byť modelovaná za pomoci mechaniky kontinua, kde je membrána reprezentovaná ako izotropný tenký plát. Elastické parametre tohto modelu sú práve tie, ktoré sú často odvodené z meraní v biologických experimentoch. Na základe týchto informácií boli odvodené počiatočné hodnoty využité pri kalibračnom procese pre RBC bunky.

Odvodenie vzťahov medzi kontinuálnym modelom membrány a strunovým modelom využívaným v simulačnom nástroji sa nachádza v článku[41]. Tu uvádzame len výsledné prepočty. Všetky sú analyticky odvodené len pre 2D pravidelný šesťuholníkový výrez triangulácie membrány. Preto sú tieto hodnoty parametrov len odrazovým mostíkom pre hodnoty elastických parametrov nášho modelu.

V biologických experimentoch je možné nájsť hodnoty nasledujúcich modulov. **Modul šmykovej deformácie**, μ_0 , ktorý je aktivovaný keď na povrch telesa pôsobí sila rovnobežne so sledovaným prierezom. Ďalší je **modul kompresie/expansionie plochy**, K , ktorý súvisí s deformáciou spôsobenou pri stláčaní/naťahovaní sledovaného prierezu do všetkých smerov v rámci roviny tohto prierezu. Posledným je **modul ohybu**, k_c . Ten, na rozdiel od predošlých dvoch modulov nesúvisí s deformáciami v rámci jednej roviny, ale s deformáciou mimo tejto roviny. Samotná RBC sa totiž deformuje aj ohýbaním.

Pre výpočet medzi parametrami modelu (k_s - modul pružnosti, k_b - modul ohybu, k_{al} - modul zachovania lokálnej plochy) a hodnotami z biologických experimentov (μ_0 , K a k_c) boli využité nasledujúce vzťahy:

$$\mu_0 = k_s \frac{\sqrt{3}}{4} \quad (3.1)$$

$$K = k_s \frac{\sqrt{3}}{2} + \frac{k_{al}}{2} \quad (3.2)$$

$$k_b = \sqrt{3}k_c \quad (3.3)$$

Biologické hodnoty pre tieto moduly sa líšia v rámci dostupnej literatúry. Dôvody sú rôznorodé, niektoré z nich sú spomenuté v časti 2.3.2. Pre kalibračný proces je preto dôležité brať do úvahy, čo presne bolo v biologickom experimente namerané, aká chyba môže byť zahrnutá v biologických meraniach a akým spôsobom využiť získané biologické merania pre simulácie.

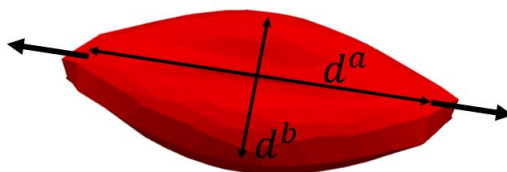
Pre kalibráciu elastických parametrov bol zvolený naťahovací experiment [13].

V tomto experimente sú na RBC upevnené silikové mikrogulôčky na dvoch protiľahlých stranách bunky. Použitím optickej pinzety je potom bunka naťahovaná za tieto gulôčky.

Mnoho simulačných experimentov bolo kalibrovaných práve pomocou Daovho experimentu. Zaujímavá je otázka ako je možné dáta z biologického experimentu využiť pre kalibráciu, a akým spôsobom vôbec nastaviť v simulácii naťahovanie.

V experimente sú postupne aplikované rôzne sily a zaznamenávajú sa dve dĺžky. Pozdĺžne predĺženie d_a v smere naťahovania a priečne zúženie d_b kolmo na smer naťahovania, ako môžeme vidieť na obrázku 3.1.

Zaujímavé argumenty boli ukázané v článku [72], kde porovnávali rôzne modely a ako sa tieto líšia, keď sa porovnáva viac než d_b a d_a naťahovanej bunky.



Obrázok 3.1: Naťahovanie RBC.

Jedným z dôležitých nastavení simulácie bolo určenie akým spôsobom sa bude bunka naťahovať.

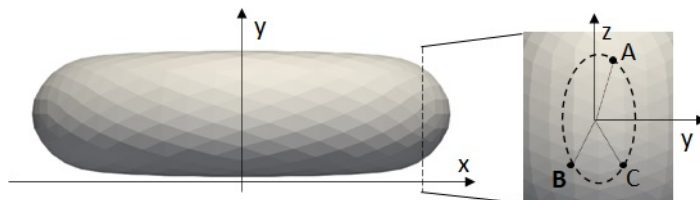
Nastavenie prstencov

V nastavení simulácie nie je potrebné modelovať silikové gulôčky, pretože sme schopní modelovať správanie týchto gulôčok pomocou aplikácie príslušných síl na konkrétne body trojuholníkovej siete. To nám šetrí výpočtový čas, pretože modelovanie gulôčok by si vyžadovalo viac bodov triangulácie objektov a interakcií v simulácii. Na druhej strane sa musíme rozhodnúť, kde presne by sme mali aplikovať naťahovaciu silu, aby sme dosiahli rovnaký účinok ako v biologickom experimente. Na tento účel sme navrhli tri možnosti.

1. **Bod.** Pravdepodobne najjednoduchší spôsob je aplikovať danú naťahovaciu silu na dva najkrajnejšie body triangulácie povrchu bunky. Jeden pre každú silikovú gulôčku. Jeden bod s najmenšou x -ovou súradnicou (pre ľavú silikovú gulôčku) a druhý bod s najväčšou x -ovou súradnicou (pre pravú silikovú gulôčku), pretože sme si zvolili naťahovanie pozdĺž x -ovej osi.

3.3. KALIBRÁCIA RBC

2. **Prstenec.** Druhou možnosťou je vytvoriť podobnú kontaktnú plochu ako v biologickom experimente. Pri upevnení silikovej guľôčky na bunku má ich spoločná plocha tvar kruhu. V [13] uvádzajú, že priemer tejto kontaktnej plochy je približne $2\mu m$. Pre aproximáciu tejto kruhovej plochy sme z bodov membrány vytvorili oválny prstenec tak, aby zodpovedal priemeru kontaktnej plochy silikových guľôčok. Body môžu byť definované nastavením intervalu hodnôt pre ich x -ové súradnice. Priemer kontaktnej plochy sa potom vypočíta ako dvojnásobok priemeru vzdialeností zvolených bodov od stredu tohto oválu. Keďže body na prstenci hľadáme pre danú hodnotu x , tak používame ako stred oválu bod $[0, 0]$ v rovine danej hodnotou x . Na obrázku 3.2 môžeme vidieť príklad pre určenie priemeru.



Obrázok 3.2: Kontaktná plocha silikových guľôčok.

Majme tri body $A[x_a, y_a, z_a]$, $B[x_b, y_b, z_b]$ a $C[x_c, y_c, z_c]$, ktorých x -ová súradnica je v správnom intervale. Potom priemer kontaktnej plochy spočítame ako:

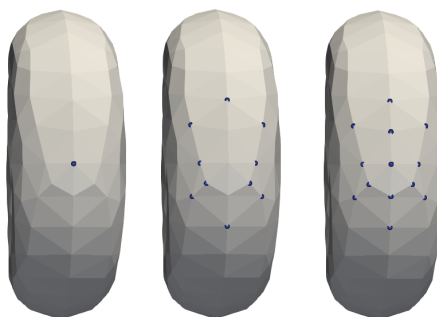
$$C_d = 2 \frac{\sqrt{(y_a^2 + z_a^2)} + \sqrt{(y_b^2 + z_b^2)} + \sqrt{(y_c^2 + z_c^2)}}{3}. \quad (3.4)$$

Úlohou tu preto bolo nájsť správny interval x -ových súradníc, aby sa dosiahol $2\mu m$ priemer kontaktnej plochy. Keďže ide o diskretný model, nechali sme priestor pre miernu odchýlku z dôvodu zachovania symetrie medzi ľavou a pravou stranou. Ako fungujúce sme označili tie priemery, ktoré boli v rozmedzí od $1,7 \mu m$ do $2,2\mu m$.

Potom, čo máme výber bodov tvoriacich kontaktnú plochu, aplikujeme na ne naťahovaciu silu. Silu aplikovanú na každú zo silikových guľôčok rozpočítame rovnomerne na všetky body prstenca.

3. **Čiapočka.** Posledná možnosť vyplynula zo skutočnosti, že nájdenie správneho intervalu pre x -ové hodnoty pre prstenec bolo možno príliš komplikované vzhľadom na úlohu, ktorú sme riešili. Chceli sme otestovať, či bude rozdiel medzi

napínaním prstenca a napínaním čiapočky, kde potrebujeme iba jednu hraničnú hodnotu a nie interval x -ových súradníc. Toto kritérium sme nastavili tak, aby priemer kontaktnej plochy bol väčší ako $1,7 / \mu\text{m}$. Naťahovacia sila bola aplikovaná rovnomerne na všetky body, ktoré sa nachádzajú v oblasti čiapočky (pravej aj ľavej).



Obrázok 3.3: Rôzne spôsoby naťahovania buniek. Zľava doprava postupne: Bod. Prstenec. Čiapočka. Vyznačené sú body, do ktorých je umiestnená naťahovacia sila. Naťahovanie je symetrické na oboch stranách bunky.

Porovnanie rôznych prístupov k naťahovaniu

Vyššie uvedené tri metódy naťahovania sme porovnali spustením naťahovacej simulácie pre každú z nich. Ako počiatočnú možnosť sme vyberali bunku s 374 diskretizačnými bodmi. Pre každú možnosť bola spustená sada piatich simulácií s parametrami nastavenými na hodnoty uvedené v tabuľke 3.1 a aplikovaná naťahovacia sila bola postupne: 0.016, 0.047, 0.088, 0.130, 0.192 nN .

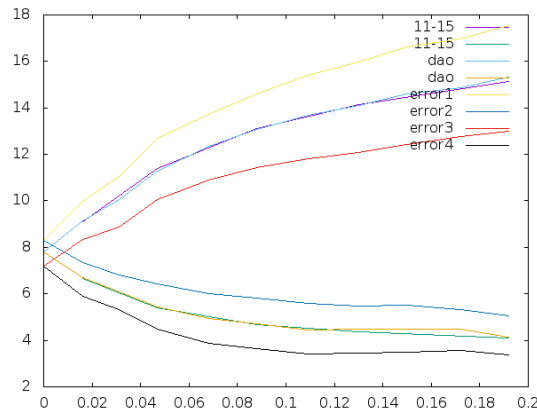
Pre každú simuláciu sme zaznamenali dĺžku bunky v smere naťahovania (axiálna dĺžka) a kolmo na smer naťahovania (pričná dĺžka). Takto sme získali desať hodnôt pre každý prístup. Tieto hodnoty sú vynesené v grafe spolu s hodnotami meraní z biologického experimentu [13] na obrázkoch 3.4, 3.5 a 3.6.

Z obrázkov vidíme, že prístup naťahovania za body nie je veľmi užitočný. Aj pri veľmi nízkych hodnotách naťahovacej sily je axiálna dĺžka bunky ďaleko od biologických hodnôt. Tento prístup môžeme vyradiť.

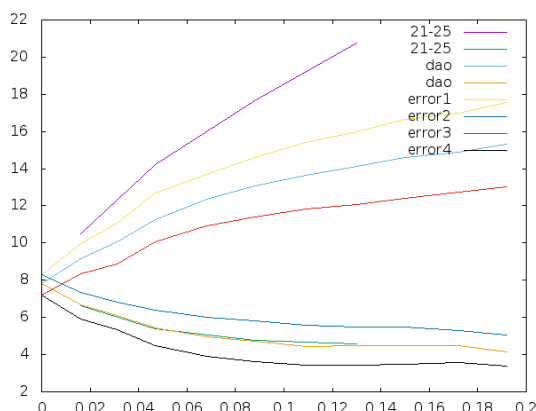
Pri porovnaní „prstenca“ a „čiapočky“ sa javí, že poskytujú celkom dobré a podobné výsledky axiálnych a pričných dĺžok bunky. Na rozlíšenie ich kvality sme vypočítali normalizované stredné kvadratické chyby z biologických údajov. Pre každú hodnotu sa najprv vypočítala normalizovaná chyba - rozdiel simulačných a experimentálnych hodnôt vydelený rozsahom odchýlok z experimentálnych meraní:

Tabuľka 3.1: Simulačné parametre

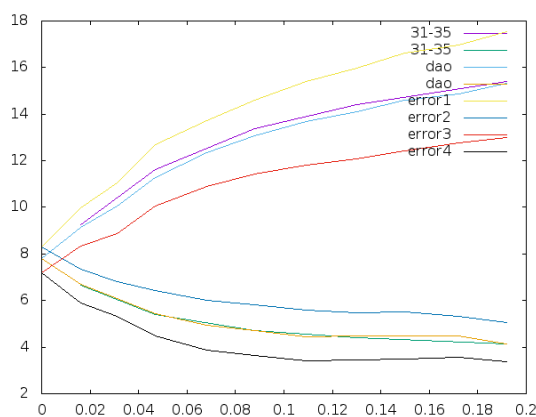
časový krok	10^{-7} s
lbgrid	10^{-6} m
k_s	$4 \cdot 10^{-6} N/m$
k_b	$8 \cdot 10^{-12} N$
k_{al}	$3.9 \cdot 10^{-5} N/m$
k_{ag}	$7 \cdot 10^{-4} N/m$
k_v	$9 \cdot 10^2 N/m^2$
polomer RBC	$3,91 \cdot 10^{-6} m$
koeficient trenia	$3,39 \cdot 10^{-9} N \cdot s \cdot m^{-1}$
hustota tekutiny	$10^3 kg \cdot m^{-3}$
viskozita tekutiny	$1.5 \cdot 10^{-6} m^2/s$



Obrázok 3.4: Výsledky naťahovacích simulácií s použitím prstencov. Na x -ovej osi sú vynesené hodnoty sily naťahovania v nN a na y -ovej osi sú dĺžky v μm . Krivky reprezentujúce biologické hodnoty z [13] sú označené ako "dao". Horná "dao" krivka označuje pozdĺžne predĺženie bunky d_a v smere naťahovania pri rôznych silách naťahovania. Dolná "dao" krivka označuje priečne zúženie bunky d_b pri rôznych silách naťahovania. Odchýlky biologických hodnôt sú označené ako "error". Máme hornú a dolnú hranicu pre pozdĺžne predĺženie aj pre priečne zúženie. Krivky označené ako 11-15 sú výsledky zo simulácií. Opäť horná krivka pre pozdĺžne predĺženie a dolná krivka pre priečne zúženie bunky.



Obrázok 3.5: Výsledky ťaťahovacích simulácií s použitím bodov. Na x -ovej osi sú vynesené hodnoty sily ťaťahovania v nN a na y -ovej osi sú dĺžky v μm . Krivky reprezentujúce biologické hodnoty z [13] sú označené ako "dao". Horná "dao" krivka označuje pozdĺžne predĺženie bunky d_a v smere ťaťahovania pri rôznych silách ťaťahovania. Dolná "dao" krivka označuje priečne zúženie bunky d_b pri rôznych silách ťaťahovania. Odchýlky biologických hodnôt sú označené ako "error". Máme hornú a dolnú hranicu pre pozdĺžne predĺženie aj pre priečne zúženie. Krivky označené ako 21-25 sú výsledky zo simulácií. Opäť horná krivka pre pozdĺžne predĺženie a dolná krivka pre priečne zúženie bunky.



Obrázok 3.6: Výsledky ťaťahovacích simulácií s použitím čiapočiek. Na x -ovej osi sú vynesené hodnoty sily ťaťahovania v nN a na y -ovej osi sú dĺžky v μm . Krivky reprezentujúce biologické hodnoty z [13] sú označené ako "dao". Horná "dao" krivka označuje pozdĺžne predĺženie bunky d_a v smere ťaťahovania pri rôznych silách ťaťahovania. Dolná "dao" krivka označuje priečne zúženie bunky d_b pri rôznych silách ťaťahovania. Odchýlky biologických hodnôt sú označené ako "error". Máme hornú a dolnú hranicu pre pozdĺžne predĺženie aj pre priečne zúženie. Krivky označené ako 31-35 sú výsledky zo simulácií. Opäť horná krivka pre pozdĺžne predĺženie a dolná krivka pre priečne zúženie bunky.

3.3. KALIBRÁCIA RBC

Tabuľka 3.2: Výsledky natahovacích simulácií pre rôzne nastavenia x -ových súradníc.

	sim11-15	sim41-45	sim51-55	sim61-65	sim71-75
x_{min}	0.9	0.91	0.86	0.86	0.92
x_{max}	0.98	0.99	0.99	0.98	0.99
diameter	1.87	1.79	2.11	2.21	1.70
No. points	10	12	18	16	10
error	1.78%	2.60%	8.31%	10.38%	6.41%

$$err_{n_i} = \frac{\|d_{sim} - d_{bio}\|}{err_{max} - err_{min}}. \quad (3.5)$$

Kde n_i označuje postupne jednotlivé error, $i \in 1, 2, 3, \dots, 10$. V simuláciách sme namerali 10 hodnôt - 5 pozdĺžnych predĺžení a 5 priečných skrátení, tieto sú označené d_{sim} . Každá z týchto nameraných hodnôt má príslušnú biologickú hodnotu, ktorú sme označili d_{bio} . Každá z biologických hodnôt má danú hornú a dolnú odchýlku, ktoré sme označili ako err_{max} a err_{min} . Výsledná chyba simulácie, označená ako err_{final} , bola vypočítaná ako RMSE (root-mean-square-error) odchýlka:

$$err_{final} = \sqrt{\frac{\sum_{i=1}^{10} err_{n_i}^2}{10}}. \quad (3.6)$$

Chyba pri natahovaní pomocou prstencov bola 1.7% a pri natahovaní pomocou čiapočky bola 4%. Na základe týchto výsledkov sme sa rozhodli použiť prístup využívajúci prstence.

Určenie hodnôt x -ových súradníc

Potom čo sme vybrali najlepší z prístupov, otestovali sme, ako presne je potrebné nastaviť intervaly pre x -ové súradnice. Testovali sme sadu 5 intervalov. Všetko pre bunku s 374 diskretizačnými bodmi. Výsledky sú uvedené v tabuľke 3.2.

Na základe týchto výsledkov sme dospeli k záveru, že na dosiahnutie presnejších simulácií sa oplatí nastaviť prstence čo najpresnejšie na priemer $2 \mu m$. Zistili sme, že aj malé zmeny v nastavenom intervale môžu viesť k veľkým rozdielom v nameraných axiálnych a priečných dĺžkach buniek.

V našich simuláciách sa používajú aj ďalšie diskretizácie pre membránu bunky. Pretože medzi hodnotami intervalu pre bunky s 374 diskretizačnými bodmi boli významné rozdiely, kalibrovali sme aj správne hodnoty pre ďalšie diskretizácie používané v našich simuláciách. Výsledky sú uvedené v tabuľke 3.3.

Tabuľka 3.3: Intervaly pre x -ové súradnice pre rôzne diskretizácie

diskretizácia	482	642	1002
x min	0.92	0.92	0.905
x max	0.98	0.97	0.98
priemer kontaktnej plochy	1.88	1.90	2.05
Počet bodov prstenca	12	12	18

Porovnali sme tri rôzne prístupy, ako simulovať naťahovanie bunky silikovými guľôčkami pomocou lasera. Ako sme očakávali, bodový prístup nepriniesol užitočné výsledky. Zaujímavý bol rozdiel medzi čiapočkou a prstencom. Na prvý pohľad by sa dalo určiť, že tieto dva prístupy sú rovnako efektívne. Kvantitatívne meranie normalizovanej priemernej štvorcovej chyby však poskytlo podrobnejšie porovnanie.

Na základe týchto výsledkov sme zistili, že pre najpresnejšie výsledky v experimente naťahovania optických pinziet je prstencový prístup najlepší z troch tu testovaných prístupov. Pri tomto prístupe sú sily rovnomerne rozložené a pôsobia na body na prstencoch. Prstence majú oválny tvar. Rozmery prstencov sú porovnateľné s rozmermi kontaktnej plochy silikových guľôčok z biologického experimentu. V tabuľke 3.3 uvádzame fungujúce hodnoty intervalov pre ďalšie diskretizácie RBC.

Výsledky kalibrácie

So správne nastavenými prstencami sme následne spustili veľké množstvo simulácií pre rôzne kombinácie jednotlivých parametrov. Pre každú sadu parametrov sme spustili sadu 5 simulácií kde bola bunka naťahovaná rôznymi silami. Konkrétne išlo o hodnoty 16, 47, 88, 130 a 192 pN z Daovho experimentu. Používali sme bunku s polomerom 3,91 μm a s postupne 374, 642 a 1002 bodmi triangulácie.

Ako sme už spomenuli, hodnoty elastických modulov pre bunky sa líšia naprieč literatúrou. V prehľadovom článku [78], kde sú zhrnuté hodnoty z viacerých článkov používajúcich rôzne techniky píše, že hodnota pre modul šmykovej deformácie je $\mu_0 = 5.5 \pm 3.3 \mu N/m$. V tom istom článku je aj hodnota pre modul ohybu $k_c = 1.15 \pm 0,9 \times Nm$. Tieto hodnoty dávajú pomerne očakávateľné výsledky pri prepočte do našich parametrov. Využitím vzorcov 3.1 a 3.3 získame $k_s = 0.0127$ a $k_b = 0.0002$. Už problematickejšie je to so získaním hodnoty modulu expanzie/kompresie lokálnej plochy. V tom istom prehľadovom článku [78] píše o hodnotách rádovo vyšších v porovnaní s hodnotami μ_0 . Tento problém je hlbšie rozpísaný v [9]. Pre naše potreby budeme brať hodnotu K (modul kompresie/expanzie plochy, pozri vzorec 3.2) dvojnásobnú oproti hodnote μ_0 . Tento vzťah bol experimentálne overený v [47].

3.4. OVERENIE KALIBRÁCIE RBC

Tým pádom teda dostávame $k_{al} = 0.028$. Keďže hodnoty pre k_{ag} a k_v nesúvisia priamo s mechanikou správania sa bunky, dávame najnižšie ktoré fungujú. Ako štartovacie sme pre ne zvolili teda hodnoty ktoré boli používané doposiaľ, $k_{ag} = 0.9$ a $k_v = 0.5$.

Najväčší vplyv na správanie bunky mali zmeny v module pružnosti a ohybu. Preto sme sa rozhodli zamerať sa na tieto dva parametre a jemnejšie otestovať ich okolie.

Výstupom bola oblasť kombinácií parametrov, ktoré fungujú. Dôvodom prečo nebola vybraná jedna najlepšia hodnota sady parametrov je fakt, že nech je kalibrácia robená s najväčšou možnou presnosťou, stále ide o jeden konkrétny druh deformácie bunky v danom biologickom experimente. Pri ťažovaní modul ohybu nemá až tak veľký vplyv. To však nemusí platiť všeobecne pre správanie bunky. Môže to byť spôsobené napríklad tým, že pri ťažovacom experimente bunka nie je až tak ohýbaná, a tým pádom sa nemá ako prejavíť proces, ktorý ma pôsobiť proti sile ohýbania. Detailné výsledky kalibrácie sú v prílohe A.

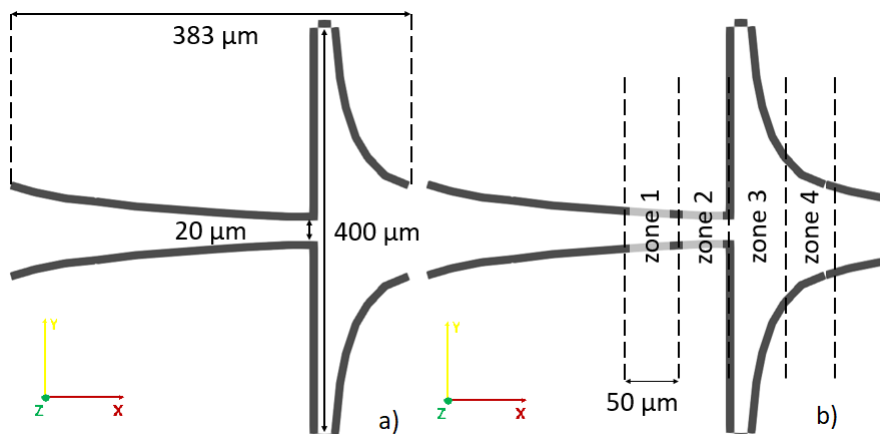
3.4 Overenie kalibrácie RBC

Pre účely overenia kalibrácie sme si vybrali laboratórny experiment od Rodriguesa [62]. Popisovaný biologický experiment využíva mikrofluidický kanál s viacerými zúženiami a rozšíreniami cez ktorý boli púšťané RBC a WBC. Následne bol meraný ich deformačný index (DI). Mikrofluidické zariadenie, cez ktoré tieto bunky púšťali bolo zložené z dvoch častí. Prvá slúžila na oddelenie buniek od krvi a ich následné nasmerovanie do druhej časti zariadenia. Táto pozostávala z viacerých rovnakých paralelných mikrokanálov. Každý z týchto mikrokanálov bol zložený z troch rovnakých, sériovo uložených, hyperbolických komôr. Jednu z nich môžeme vidieť na obrázku 3.7

Počas toho ako bunky prechádzali cez kanál bola zaznamenaná ich dĺžka v smere osi x (L_{max}) a šírka v smere osi y (L_{min}) v každej zo štyroch zón. Následne z nich bol vypočítaný deformačný index, pomocou vzorca:

$$DI = \frac{L_{max} - L_{min}}{L_{max} + L_{min}} \quad (3.7)$$

Pre porovnanie výsledkov laboratórneho experimentu a simulácie bolo potrebné nastaviť všetky príslušné parametre simulácie tak, aby zodpovedali podmienkam v



Obrázok 3.7: Jedna z komôr mikrofluidického zariadenia modelovaného pre overenie kalibrácie RBC. Obrázok z nášho článku [43].

experimente. Elastické koeficienty bunky boli nastavené podľa kalibrácie. Tvar a veľkosť kanála podľa rozmerov zariadenia z biologického experimentu. Pri experimente bol použitý roztok Dextran 40 a hematokrit bol 2%. Hustota tohto roztoku bola teda 10^3kg/m^3 a viskozita bola $4 \cdot 10^{-6} \text{m}^2/\text{s}$, pri teplote $22 \pm 1^\circ\text{C}$, pri ktorej boli vykonané všetky merania. Okrem týchto parametrov bolo nutné v simulácii nastaviť externú silu tekutiny.

Pre jej nastavenie bolo potrebné získať informáciu o rýchlosti tekutiny v mikrofluidickom kanáli. V článku boli uvedené rýchlosti šmykovej deformácie. Rýchlosť tekutiny bola vypočítaná pomocou vzorca udávajúceho vzťah medzi rýchlosťou šmykovej deformácie (σ), rýchlosťou tekutiny (v) a hydraulickým priemerom kanála ($D_h = \frac{2wh}{w+h}$, kde w je dĺžka a h je výška kanála).

$$\sigma = \frac{v}{D_h}, \quad (3.8)$$

Výsledkom boli štyri rýchlosti pre každú zo štyroch zón kanála. Postupne od prvej zóny boli rýchlosti $0.0481; 0.0535; 0.0094$ a 0.0049ms^{-1} . Externú silu sme nastavili tak, aby sa zhodovala s maximálnou rýchlosťou, ktorú dosiahla tekutina v druhej zóne. V tabuľke 3.4 sú zhrnuté všetky parametre nastavené v simulácii.

S takto nastavenou simuláciou bola spustená sada 25 simulácií, kde v každej bola počiatočná pozícia bunky mierne odchylená od stredu a natočená pod iným uhlom. Rôzne pozície boli vytvorené pre simulovanie situácie rôznych vstupov do kanála, ktoré sa diali v biologickom experimente. Vzhľadom na to, že v experimente zaznamenávali informácie len o bunkách prechádzajúcich stredom, stačilo spustiť len

3.4. OVERENIE KALIBRÁCIE RBC

Tabuľka 3.4: Simulačné parametre

časový krok	10^{-7} s
lbgrid	10^{-6} m
externá sila tekutiny	$1,5 \cdot 10^6$ N/m ³
k_s	$8 \cdot 10^{-6}$ N/m
k_b	$3 \cdot 10^{-12}$ N
k_{al}	$3 \cdot 10^{-6}$ N/m
k_{ag}	$9 \cdot 10^{-4}$ N/m
k_v	$5 \cdot 10^2$ N/m ²
polomer RBC	$3,91 \cdot 10^{-6}$ m
koeficient trenia	$3,39 \cdot 10^{-9}$ N · s · m ⁻¹
hustota tekutiny	10^3 kg · m ⁻³
viskozita tekutiny	$4 \cdot 10^{-6}$ m ² /s

malé vychýlenia od stredu.

Prvou metrikou, použitou pre porovnanie biologického a simulačného experimentu bol rýchlostný profil bunky pri prechode kanálom. V zúženej časti, čiže v zónach 1 a 2, sa profily takmer úplne zhodovali s reálnym experimentom. V zónach 3 a 4 pri daných parametroch hustoty a viskozity tekutiny nebolo možné definovať takú externú silu tekutiny, aby bol tok v simulácii zhodný s tokom v laboratórnom experimente vo všetkých zónach kanálu. Cieľom pri tomto simulačnom experimente bolo predovšetkým porovnať deformačný index a nie správanie sa tekutiny. Jedným z dôvodov prečo nebola dosiahnutá rovnaká rýchlosť mohlo byť iné pochopenie vstupných hodnôt vzorca použitého pre výpočet rýchlosti tekutiny z hodnôt rýchlosti šmykovej deformácie. Toto sa odzrkadlilo aj na výsledkoch deformačného indexu. Záverom tohto experimentu bolo, že nie je možné porovnať hodnotu deformačného indexu v tretej a štvrtej zóne, pretože rýchlosť tekutiny simulácie a experimentu sú rôzne. Naopak, v prvých dvoch zónach, kde sa rýchlosti zhodujú môžeme porovnať DI. Priemerná hodnota DI zo simulácie sa líši od hodnoty DI z experimentu o 9,7% v prvej zóne a 5,1% v druhej zóne. Čím sme úspešne validovali nakalibrované elastické koeficienty RBC. Viac podrobností je v našom článku [43].

4 Model bunky s jadrom

Rôzne štúdie sa zameriavajú na zisťovanie biologických, fyzikálnych a fyziologických vlastností CTC buniek. Prehľad vybraných štúdií sme prezentovali v časti 2.3.2. Je však dôležité myslieť na to, že týchto buniek je pomerne veľké množstvo a líšia sa veľkosťou, tvarom, tuhosťou a mnohými ďalšími vlastnosťami. Taktiež je náročné tieto bunky získať a následne ich udržať v živom stave v laboratórnych podmienkach. Vytvorenie katalógu všetkých typov týchto buniek, dokonca len výberu tých najčastejšie sa vyskytujúcich, by prevyšovalo rozsah takejto práce.

Naším cieľom je preto vytvoriť model bunky s jadrom, ktorý bude možné prispôbiť vždy danej bunke, ktorú je potrebné modelovať. Pre správne nastavenie parametrov popisujeme kalibračný proces. V tejto časti sme sa zamerali na popisanie troch prístupov k modelovaniu buniek s jadrom. Prvé dva spôsoby využívajú vloženie gule predstavujúcej jadro do membrány elastickej bunky. Samotné správanie membrány a jadra sa riadi rovnakou mechanikou ako RBC, dôležité je nastaviť správne elasticke parametre. Najväčšou výzvou je spôsob akým udržať jadro v bunke. Pre tento účel využívame viazané a neviazané interakcie dostupné v ESPResSe. Porovnávame viacero spôsobov v rámci jednotlivých typov interakcií, aby sme mohli vybrať najefektívnejší spôsob. Tretí prístup modelovania bunky s jadrom spočíva vo využití elastickej sféry s upravenými parametrami.

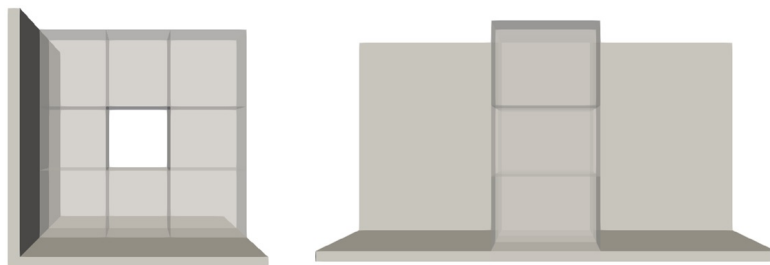
4.1 Simulačný experiment pre testovanie vplyvu parametrov interakcií

Podľa informácií z biologických experimentov vieme, že by bunky mali byť schopné prejsť určitými zúženiami v mikrofluidických zariadeniach a zúženiami in vivo. Na základe rozmerov takýchto zúžení sme navrhli kanál na testovanie tejto schopnosti buniek. Našou hypotézou bolo, že čím menšie bude zúženie kanálu, tým väčšie bude

4.1. SIMULAČNÝ EXPERIMENT PRE TESTOVANIE VPLYVU PARAMETROV INTERAKCIÍ

napätie na deformáciu bunky, a tým viac budeme môcť pozorovať vplyv konkrétneho typu interakcie a jej parametrov na deformáciu bunky a celkovú funkčnosť tohto prístupu. Pod funkčnosťou rozumieme hlavne schopnosť bunky s jadrom prejsť zúžením pri daných nastaveniach modelu.

Aby sme otestovali našu hypotézu, vytvorili sme simuláciu s prúdením tekutiny cez jednoduchý kanál so zúžením. Kanál bol tvorený úzkym kvádom s prierezom $40 \mu m \times 40 \mu m$ a dĺžkou $80 \mu m$. Zúženie bolo uprostred kanála, s dĺžkou $20 \mu m$ a prierez bol štvorec s rôznymi veľkosťami ($13 \mu m \times 13 \mu m$, $15 \mu m \times 15 \mu m$, $17 \mu m \times 17 \mu m$ a $19 \mu m \times 19 \mu m$). Geometriu kanáliku môžeme vidieť na obrázku 4.1. Skript pre vytvorenie takejto geometrie je uvedený v prílohe B.1.



Obrázok 4.1: Kanálik so zúžením. Rozmery kanáliku $40 \mu m \times 40 \mu m \times 80 \mu m$, rozmery zúženia $13 \mu m \times 13 \mu m \times 20 \mu m$.

Aby sme mohli porovnať jednotlivé prístupy nastavenia modelovania interakcie bunky s jadrom, nastavili sme vo všetkých simulačných experimentoch model membrány a jadra rovnako. Rovnaký bol aj tok tekutiny. Menili sme len parametre ovplyvňujúce jednotlivé typy interakcií.

Pre modelovanie membrány a jadra sme zvolili elastickú guľu. Keďže informácie o veľkosti jadra v 3D suspenziách sú takmer neexistujúce, zvolili sme jadro s polovičným polomerom oproti veľkosti polomeru bunky. Častejšie je možné v literatúre nájsť meranie jadra v 2D priestore, ako napríklad v [68]. Avšak plocha jadra v porovnaní s plochou bunky v takomto preparáte slúži len ako odhad pre rozmer v 3D.

Polomer jadra bol nastavený na $3 \mu m$ a polomer membrány na $7.5 \mu m$. Pre membránu bola použitá triangulácia gule s 304 bodmi a pre jadro s 169 bodmi. Elastické parametre jadra boli nasledujúce : $k_s = 1.4$, $k_b = 1.4$, $k_{al} = 1.4$, $k_g = 2.5$, $k_v = 3.5$. Boli nastavené tak, aby vytvorili dostatočne tuhú guľu. Parametre membrány boli: $k_s = 0.2$, $k_b = 0.16$, $k_{al} = 0.2$, $k_g = 0.7$, $k_v = 0.9$. Tie boli nastavené oveľa nižšie, čím spôsobili, že membrána bola pružnejšia ako jadro. Tok v kanáliku bol nastavený

pomocou *Fluid force density* = 0.005.

4.1.1 Metódy porovnávania simulačných experimentov

Hlavným cieľom pri týchto simulačných experimentoch bolo zistiť, za akých podmienok prejde bunka s polomerom $15\mu m$ zúžením, ktorého prierez je menší ako táto bunka. Okrem toho či prejde, nás zaujímal aj to, ako veľmi sa deformuje a ako veľmi sa jadro pohybuje vnútri bunky. Pri prvotných simuláciách sme si tiež všimli nebiologické správanie jadra pri prechode zúžením s prierezom $13\mu m \times 13\mu m$. Zatiaľ čo membrána musela spomaliť kvôli nutnosti deformácie, jadro, ktorého rozmery boli menšie ako daný prierez sa deformovať nemuselo. A pokiaľ boli sily udržiavajúce jadro v bunke príliš slabé, vplyv tekutiny spôsobil, že sa jadro dostalo von z membrány bunky. Takýto jav však u reálnych biologických buniek nenastáva. Preto sme potrebovali metódu, ktorá bude sledovať či sa jadro udržiava vnútri bunky.

V simuláciách sme preto sledovali nasledujúce štyri ukazovatele:

- **Prechod cez kanálik.**

Pokiaľ bunka nebola pri danom nastavení schopná prejsť kanálikom, znamenalo to, že daná sada parametrov je nevyhovujúca. Pri prvotných testovaniach sme mali zopár prípadov, keď bunka prešla kanálikom raz, ale pri druhom prechode sa buď zasekla na vstupe do zúžením, alebo sa jadro z membrány bunky uvoľnilo. Dôvodom mohlo byť, že pri spustení simulácie máme vycentrovaný stred bunky aj jadra a po prvom prejení kanáliku to už tak byť nemusí. Stále by však mala byť schopná prejsť ním znovu. Preto sme testovali, či bunky vedia prejsť kanálikom v simulačných experimentoch dvakrát.

- **Deformačný index**

Deformačný index sme počítali ako $DI = \frac{L_x}{L_y}$, kde L_x je dĺžka bunky pozdĺž x -ovej osi (smer toku tekutiny) a L_y je dĺžka bunky pozdĺž y -ovej osi (smer šírky kanála). Bol vypočítaný pre jadro aj pre membránu.

- **Vzdialenosť stredov jadra a stredov membrány bunky.**

V simuláciách vieme sledovať pohyb bunky na základe polohy jej stredov v čase. Aby sme videli, akým spôsobom vplýva geometria a teda nutnosť sa deformovať v prípade zúžením na pohyb jadra v bunke, pozorovali sme závislosť vzdialeností stredov jadra a bunky od polohy stredov bunky.

4.2. MODEL BUNKY S JADROM S POUŽITÍM VIAZANÝCH INTERAKCIÍ

Geometria tohto kanálika udržuje bunky v jeho strede, takže bunka vykonáva len minimálny pohyb v smere osi z a y . Pre výpočet vzdialenosti sme teda brali do úvahy len x -ové súradnice. Keďže nás zaujímala aj vzájomná poloha týchto stredov, vzdialenosť D_c sme počítali ako rozdiel stredu bunky x_{Cb} a stredu jadra x_{Cj} . V prípade $D_c > 0$ vieme, že je jadro vychýlené do zadnej časti bunky a naopak, ak $D_c < 0$ jadro je vychýlené do prednej časti bunky.

Pôvodne sme túto metódu využívali na sledovanie, či jadro uniká z bunky. Ak $|D_c| > 4,5\mu m$, tak sme vyhlásili, že jadro z bunky uniklo. Avšak toto tvrdenie platí len v prípade nezdeformovaných gúľ. Preto sme vytvorili nasledujúcu metódu.

- **Únik jadra z bunky.**

Bunky v simuláciách sa pohybujú v smere osi x . V každom časovom kroku vieme získať bod membrány a jadra, ktorý má najväčšiu x -ovú súradnicu. Nazvime ho "špic". Podobne ako pri vzdialenosti stredov sa aj pri vzdialenosti špicov pozrieme len na rozdiel ich x -ových súradníc. Vzdialenosť špicov je potom $D_s = x_{sb} - x_{sj}$, kde x_{sb} je x -ová súradnica špicu bunky a x_{sj} je x -ová súradnica špicu jadra. V prípade, že $D_s < 0$, vieme, že jadro sa nenachádza v bunke.

4.2 Model bunky s jadrom s použitím viazaných interakcií

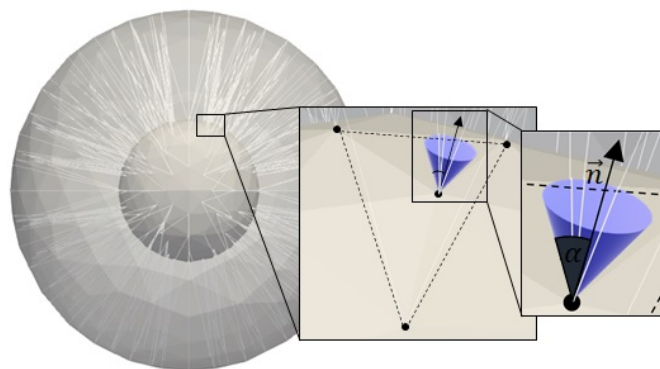
Základom pri použití prístupu s viazanými interakciami je najprv **vytvoriť väzby** medzi membránou bunky a jadrom. Potrebujeme, aby tieto väzby boli ňahavacieho typu, a aby tak umožnili jadru mierny pohyb vnútri bunky. Zároveň by mali zabrániť jadru dostať sa mimo membránu bunky.

Pri vytváraní týchto väzieb sme brali do úvahy aj dĺžky hrán triangulácie bunky a jadra, keďže príliš veľké rozdiely v ich dĺžkach by mohli viesť k numerickým chybám. Dôležitosť správnej diskretizácie priestoru tekutiny, času a rozmerov buniek je popísaná v rôznej literatúre, napríklad aj v [9]. Zachovanie porovnateľných dĺžok hrán jadra a membrány bunky prirodzene viedlo k tomu, že povrch jadra bol tvorený menším počtom bodov ako povrch membrány. Aby bolo jadro centrovane v bunke je potrebné pospájať body jadra a body membrány rovnomerne. Nastáva tak situácia, že jeden bod z povrchu jadra je spojený s viacerými bodmi na povrchu membrány.

Tabuľka 4.1: Počet väzieb medzi jadrom a membránou v závislosti od nastavenia uhla α .

a	10	15	20
α	18°	12°	9°
počet väzieb	467	222	112

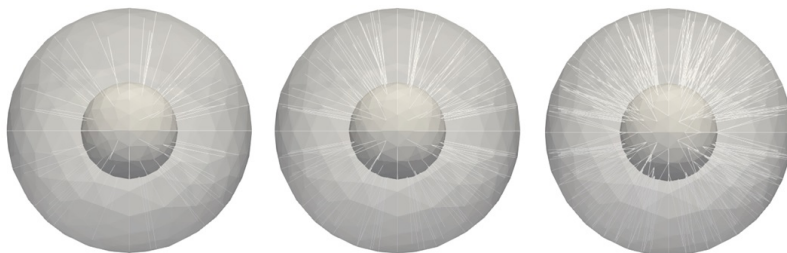
Počet takýchto spojení bol potom tiež jeden z faktorov, ktoré ovplyvňujú pohyb jadra v rámci bunky.



Obrázok 4.2: Vytváranie väzieb medzi jadrom a membránou. Môžeme vidieť uhol α . Uhly medzi všetkými väzbami a normálovým vektorom n musia byť menšie ako α .

To, s koľkými bodmi na membráne bol spojený, sme definovali pomocou uhla α . Pre každý bod na jadre bol spočítaný uhol medzi možnou väzbou a jeho "**normálovým vektorom**". Možná väzba je spojnica bodu na sieti jadra s bodom na sieti membrány. Za normálový vektor bodu na povrchu jadra pokladáme normálový vektor plochy, ktorá je definovaná pomocou trojuholníka tvoreného tromi najbližšími susedmi tohto bodu. Pre daný bod na jadre boli vytvorené tie väzby, ktorých uhol s normálovým vektorom bol menší ako uhol α , tak ako môžeme vidieť na obrázku 4.2. Hodnota pre tento uhol je nastavená na začiatku simulácie pomocou parametra, ktorý sme si označili a . V našich simuláciách sme testovali tri rôzne hodnoty tohto parametra. V tabuľke 4.1 uvádzame uhol pre príslušné nastavenie a a taktiež počet vytvorených väzieb. Tieto väzby môžeme vidieť v obrázku 4.3.

Pri definovaní viazaných interakcií v Espresso je potrebné urobiť tri kroky. Najprv je potrebné definovať interakciu, pridať ju do systému a aplikovať ju na dané body. Podrobné informácie je možné nájsť v [17].



Obrázok 4.3: Zobrazenie väzieb medzi membránou a jadrom. Postupne zľava doprava, väzby pri uhle $\alpha = 9^\circ, 12^\circ, 18^\circ$.

4.2.1 Testované viazané interakcie

V Espresso sa nachádza viacero typov interakcií, ktoré sa dajú použiť. Možné je pridať aj vlastné typy interakcií, to však vyžaduje ešte nultý krok, kde sa táto funkcia musí zdefinovať ešte v zdrojových kódach Espresso.

Interakcie sú rozdelené podľa toho, či fungujú na základe vzdialenosti medzi dvoma bodmi, na základe uhlu určeného tromi bodmi alebo na základe uhlu medzi rovinami definovanými 4 bodmi. Keďže väzby medzi jadrom a membránou sú tvorené spojnicami dvoch bodov, zamerali sme sa na možnosti viazaných interakcií závislých na vzdialenosti dvoch bodov.

V našom prípade je táto vzdialenosť **dĺžka väzby** medzi bodom na jadre a bodom na membráne. Budeme ju označovať r . Parameter r_0 bude označovať **relaxovanú dĺžku** pre danú väzbu. Táto dĺžka sa mierne líši pre rôzne väzby a je zadaná na začiatku simulácie pri ich vytváraní.

- **Harmonický potenciál** - symetrická interakcia medzi dvoma časticami. Parameter k určuje rozsah sily pôsobenia tejto funkcie. Testovali sme hodnoty $[0.00625, 0.0125, 0.025, 0.05, 0.1, 0.2]$.

$$V(r) = \frac{1}{2}k(r - r_0)^2 \quad (4.1)$$

Možné je tiež nastaviť prahovú hodnotu dĺžky väzby, r_{cut} . Pokiaľ nastane situácia, kde $r > r_{cut}$ tak je väzba zlomená a simulácia vyhlási error. V simuláciách sme túto možnosť nevyužili.

- **FENE** (finite extention nonlinear elastic) - interakcia, ktorá vytvorí väzbu simulujúcu správanie elastickej gumičky. Parameter K určuje silu tejto väzby, Δr_0 určuje maximálnu dĺžku natiahnutia. Pre K sme testovali hodnoty $[0.00625,$

0.0125, 0.025, 0.05, 0.1, 0.2], pre Δr_0 hodnoty [8, 10, 6, 12, 18] Potenciál tejto väzby $V(r)$ diverguje pri dĺžkach $(r_0 - \Delta r_{max})$ a $(r_0 + \Delta r_{max})$ a je daný funkciou:

$$V(r) = -\frac{1}{2}K\Delta r_{max}^2 \ln \left[1 - \left(\frac{r - r_0}{\Delta r_{max}} \right)^2 \right] \quad (4.2)$$

- **Kvadratický potenciál** - veľmi podobný harmonickému potenciálu.

$$V(r) = \frac{1}{2}K_0 (r - r_0)^2 + \frac{1}{4}K_1 (r - r_0)^4 \quad (4.3)$$

Hodnoty K_0 a K_1 určujú rozsah sily pôsobenia funkcie $V(r)$. Pre hodnotu $K_1 = 0$ sú harmonický a kvadratický potenciál zhodné. Testovali sme hodnoty $K_0 = [0.00625, 0.0125, 0.025, 0.05, 0.1, 0.2]$ a $K_1 = [0.001, 0.0001, 0.00001]$.

Skripty k príslušným simuláciám sú v prílohe B.2, B.3 a B.4.

4.2.2 Výsledky a porovnanie

Pre každý typ potenciálu sme vytvorili simuláciu pre každú kombináciu menovaných parametrov. Prvým krokom bolo vyhodnotiť, ktoré sady sú fungujúce. Za fungujúce sme považovali tie, pri ktorých bunka prešla kanálikom dvakrát a jadro sa nachádzalo vo vnútri membrány počas celej doby. Vyhodnotenie sád parametrov môžeme postupne vidieť v obrázkoch 4.4, 4.5 a 4.6. Hodnota 1 indikuje, že bunka prešla kanálikom a jadro sa v nej udržalo. Hodnota 0 označuje nevhodné parametre pri ktorých sa bunka buď zasekla, alebo z nej jadro odišlo. Pre kvadratický a FENE potenciál sú uvedené len výsledky pre prierez zúženia $13 \times 13 \mu m$, keďže v ňom boli najviac testované nastavenia parametrov. Tabuľky so všetkými hodnotami sú v prílohe A.2 a A.3.

Pri porovnávaní kvadratického a harmonického potenciálu si môžeme všimnúť, že nastavenia parametrov k a k_0 je to isté. Znamená to, že jediný rozdiel medzi simuláciami, ktorých výsledky sú v prvom riadku na obrázku 4.4 a prvých troch riadkoch na obrázku 4.5 je člen $\frac{1}{4}K_1 (r - r_0)^4$ s rôznymi hodnotami parametra k_1 . Vidíme však, že úpravou tohto parametra získavame viac fungujúcich sád parametrov. Celkovo, keď porovnáme všetky výsledky, kvadratický potenciál má najviac fungujúcich sád parametrov. V porovnaní s harmonickým potenciálom má totiž rýchlejšie stúpanie pri väčších dĺžkach väzieb. V prípade, že jadro uniklo, znamená to, že sily, ktoré ho udržiavali vnútri neboli dostatočné v porovnaní so silou toku. Rozdiel medzi

4.2. MODEL BUNKY S JADROM S POUŽITÍM VIAZANÝCH INTERAKCIÍ

prierez	a/k	0.00625	0.0125	0.025	0.05	0.1	0.2
13	10	1	1	1	1	0	0
13	15	0	0	1	1	1	0
13	20	0	0	0	1	0	0
15	10	1	1	1	1	1	1
15	15	0	1	1	1	1	1
15	20	0	0	1	1	1	1
17	10	1	1	1	1	1	1
17	15	0	1	1	1	1	1
17	20	0	0	1	1	1	1
19	10	1	1	1	1	1	1
19	15	0	1	1	1	1	1
19	20	0	0	1	1	1	1

Obrázok 4.4: Výsledky simulácií v ktorých bol použitý harmonický potenciál.

prierez	a	K1/K0	0.00625	0.0125	0.025	0.05	0.1	0.2
13	10	0.00001	1	1	1	1	0	0
13	10	0.0001	1	1	1	1	1	1
13	10	0.001	1	1	1	1	1	1
13	15	0.00001	1	1	1	1	1	1
13	15	0.0001	1	1	1	1	0	0
13	15	0.001	1	1	1	1	1	1
13	20	0.00001	1	1	1	1	1	1
13	20	0.0001	1	1	1	1	1	1
13	20	0.001	1	1	1	1	0	0

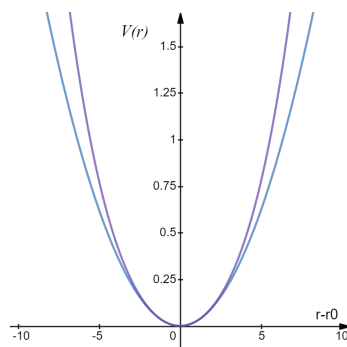
Obrázok 4.5: Výsledky simulácií v ktorých bol použitý kvadratický potenciál.

prierez	a	r_{max}/K	0.00625	0.0125	0.025	0.05	0.1	0.2
13	10	6	1	1	1	0	0	0
13	10	8	1	1	1	1	0	0
13	10	10	1	1	1	0	0	0
13	10	12	1	1	1	0	0	0
13	15	6	1	1	1	1	1	0
13	15	8	1	1	1	1	1	0
13	15	10	1	1	1	1	1	0
13	15	12	1	1	1	1	1	0
13	20	6	0	1	1	1	0	0
13	20	8	1	1	1	1	0	0
13	20	10	1	1	1	1	0	0
13	20	12	0	0	1	1	0	0

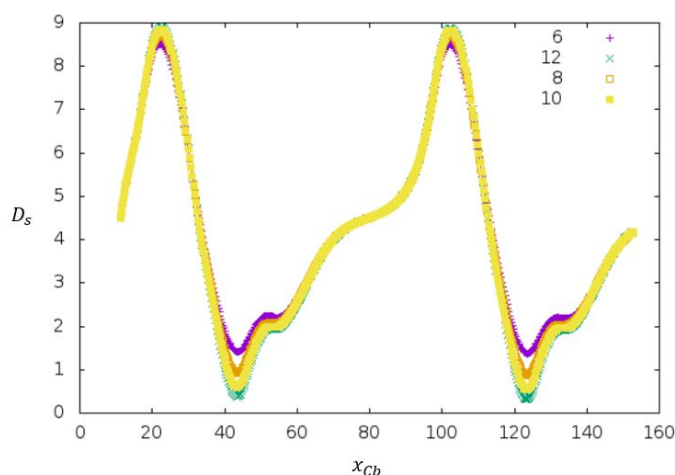
Obrázok 4.6: Výsledky simulácií v ktorých bol použitý FENE potenciál.

kvadratickým a harmonickým potenciálom môžeme vidieť na obrázku 4.7. Výsledky zo simulácií používajúcich FENE potenciál sa javia ako najtuhšie. Hoci sme očakávali, že parameter Δr_{max} bude mať najsilnejší vplyv na pôsobenie tohto potenciálu, na obrázku 4.8 môžeme vidieť, že jeho vplyv je len minimálny.

Cieľom týchto simulácií je analyzovať, akým spôsobom sa bunky správajú v závislosti od nastavenia parametrov. Vzhľadom na širšiu škálu parametrov, ktoré



Obrázok 4.7: Porovnanie harmonického a kvadratického potenciálu pre $k = k_0 = 0.05$ a $k_1 = 0.001$.

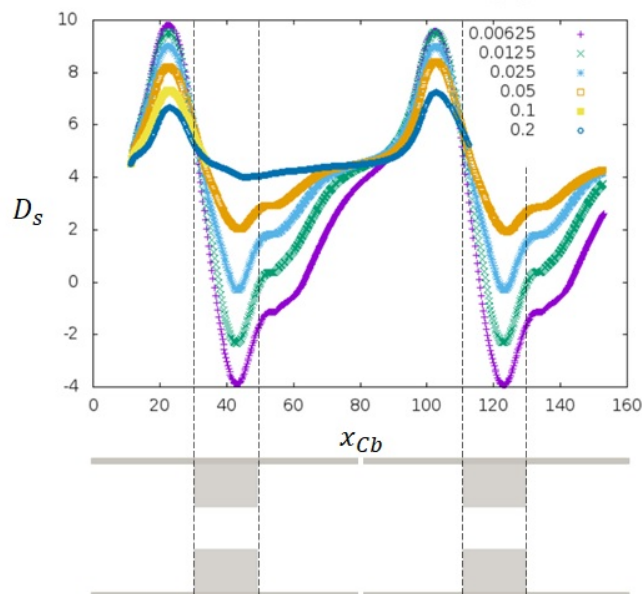


Obrázok 4.8: Graf zobrazujúci závislosť D_s (vzdialenosť špicov bunky a jadra) od zmeny parametra Δr_{max} pri použití FENE potenciálu, $a = 20$, $k = 0,025$ bunka prechádza zúžením $13 \mu m \times 13 \mu m$.

fungovali, je z nich možné vybrať hodnoty pre nastavenie tuhších ale aj mäkkších buniek. Pre získanie takejto informácie sa však potrebujeme pozrieť na samotný pohyb jadra v rámci bunky a taktiež na deformačný profil bunky v priebehu prechodu kanálikom. Pre detekciu úniku z jadra sme používali vzdialenosť špicov bunky a jadra D_s . Na obrázku 4.9 môžeme vidieť akým spôsobom ovplyvňuje pohyb jadra parameter k . Čím nižšia hodnota, tým väčší pohyb jadra a bunka sa vie jednoduchšie zdeformovať pri vstupe do kanáliku.

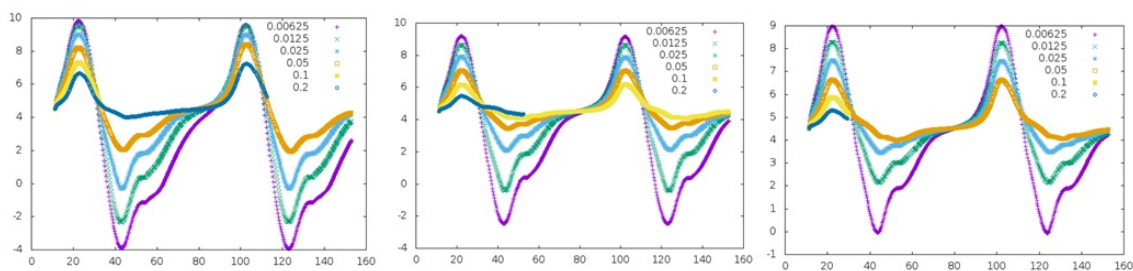
Na obrázkoch 4.10 môžeme vidieť rozdiel medzi rôznym nastavením hodnoty uhlu α , zľava doprava sa zväčšuje, pribúdajú väzby. Aj toto sú výsledky z najmenšieho prierezu. Spolu so zvyšovaním hodnoty k a α je bunka tuhšia. Tento mechanizmus môžeme sledovať na vývoji modrej krivky na jednotlivých grafoch v obrázku.

4.2. MODEL BUNKY S JADROM S POUŽITÍM VIAZANÝCH INTERAKCIÍ



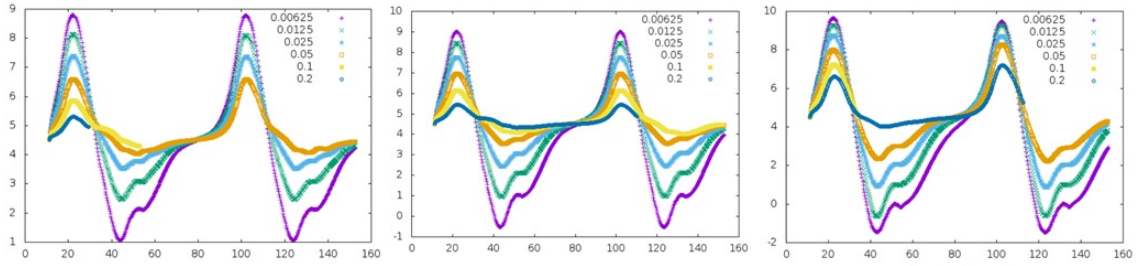
Obrázok 4.9: Graf zobrazujúci závislosť D_s (vzdialenosť špicov bunky a jadra) od zmeny parametra k pri použití harmonického potenciálu, $a = 20$, bunka prechádza zúžením $13 \mu m \times 13 \mu m$. Pod grafom vidíme pohľad zvrchu na štruktúru kanálika. Prerušované čiary vyznačujú v ktorých miestach sa začína a končí zúženie.

Naopak čím je menej väzieb, a teda slabšia sila udržiavajúca jadro v membráne, tým je väčšia pravdepodobnosť, že jadro z membrány odíde. Môžeme vidieť, že pri hodnote $k = 0,025$ jadro jemne prechádza von. Naopak pri $a = 10$ a $k = 0,025$, graf vpravo, vidíme, že minimálna vzdialenosť počas celej simulácie medzi špicmi je viac ako 3.

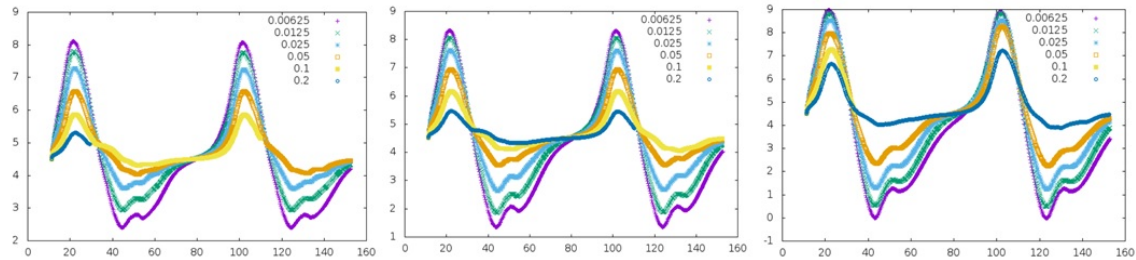


Obrázok 4.10: Grafy zobrazujúce závislosť D_s (vzdialenosť špicov bunky a jadra) od zmeny parametra k pri použití harmonického potenciálu. Postupne zľava doprava je hodnota $a = 20, 15, 10$, bunka prechádza zúžením $13 \mu m \times 13 \mu m$.

Podobný efekt závislosti medzi uhlom a celkovou tuhosťou bunky môžeme vidieť aj pri použití potenciálu FENE a kvadratického potenciálu. Obrázky 4.11 a 4.12.



Obrázok 4.11: Grafy zobrazujúce závislosť D_s (vzdialenosť špicov bunky a jadra) od zmeny parametra K pri použití FENE potenciálu. Postupne zľava doprava je hodnota $a = 20, 15, 10$, bunka prechádza zúžením $13 \mu m \times 13 \mu m$.



Obrázok 4.12: Graf zobrazujúci závislosť D_s (vzdialenosť špicov bunky a jadra) od zmeny parametra K_0 pri použití kvadratického potenciálu. Postupne zľava doprava je hodnota $a = 20, 15, 10$, bunka prechádza zúžením $13 \mu m \times 13 \mu m$.

Celkovo výsledky ukazujú, že keď je zúženie menšie, stres na bunku je väčší, a teda rovnaká sada parametrov vedie k rôznemu pohybu jadra v bunke. Z popísaných prístupov je najvhodnejší kvadratický potenciál. Avšak pri dobrom nastavení parametrov fungujú aj ostatné. Výhodou kvadratického potenciálu je, že umožňuje jemnejšie odškálovať aké správanie bunky chceme dosiahnuť. Vieme získať aj pomerne elastickú bunku, v ktorej sa jadro stále drží vnútri.

4.3 Model bunky s jadrom s použitím neviazaných interakcií

Ďalším zo spôsobov ako je možné udržať jadro vnútri buniek je využiť neviazané interakcie dostupné v ESPResSo. Tento spôsob interakcie jadra s bunkou je založený na pôsobení odpudivej sily medzi dvoma typmi častíc. V simuláciách sú body triangulácie povrchu danej bunky a jadra označené rôznymi identifikačnými číslami, a tak je možné nastaviť interakciu medzi týmito dvoma množinami bodov. Tento typ in-

4.3. MODEL BUNKY S JADROM S POUŽITÍM NEVIAZANÝCH INTERAKCIÍ

terakcie sa v simuláciách využíva napríklad pri interakcii medzi stenou simulačného kanálíka a bunkou. Espresso ponúka veľké množstvo týchto neviazaných interakcií. Väčšina z nich je však viazaná na konkrétne fyzikálne javy ovplyvňujúce správanie sa rôznych druhov molekúl. Pre potreby interakcie medzi bunkou a jadrom sú vhodné nasledujúce dva prístupy:

- Soft-sphere interakcia je definovaná pomocou funkcie

$$V(r) = \begin{cases} a(r - r_{offset})^{-n} & r < r_{cut} \\ 0 & r \geq r_{cut}. \end{cases} \quad (4.4)$$

Kde r je vzdialenosť bodov. Parametre a a n určujú škálu odpudivej sily. r_{offset} určuje vzdialenosť bodov pri ktorej začne pôsobiť a r_{cut} určuje hraničnú vzdialenosť pri ktorej prestane pôsobiť.

- Membrane-collision je veľmi podobná soft-sphere interakcii, využíva však aj informáciu o vzájomnej pozícii dvoch interagujúcich sietí. Definovaná je ako:

$$V(r) = \begin{cases} a \frac{1}{1 + e^{n(r - r_{offset})}} & r < r_{cut} \\ 0 & r \geq r_{cut}. \end{cases} \quad (4.5)$$

Pre siete, ktoré sa prekryli je hodnota tejto funkcie definovaná ako $V(-d)$. Táto interakcia je používaná pre interakcie jednotlivých buniek v toku, a v prípade prekrytia dvoch buniek umožňuje, že sa budú od seba odpudzovať a nenastane ešte väčšie prekrytie.

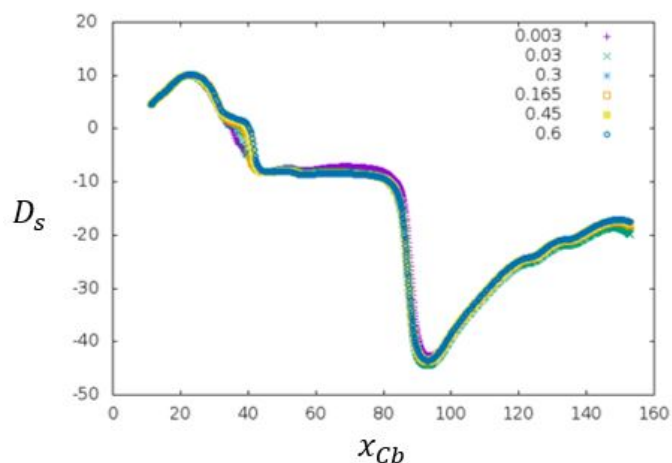
Pre udržanie jadra v bunke sa ukázala ako vhodná iba soft-sphere interakcia. Výsledky simulácií spustených pre rôzne kombinácie parametrov môžeme vidieť na obrázku 4.13. Pri všetkých týchto simuláciách bola hodnota $r_{offset} = 0$ a $n = 3$.

V testovaných sadách sme nezískali žiadnu, ktorá by udržala jadro v bunke počas prechodu zúžením $13 \times 13 \mu m$. Na obrázku 4.14 môžeme vidieť, že parameter a má len minimálny vplyv na vzdialenosť špicu bunky a jadra. Iná situácia nastáva pri zúžení $15 \mu m \times 15 \mu m$. Vplyv parametra a môžeme vidieť na grafoch v obrázku 4.15. Postupne zľava doprava je možné sledovať závislosť D_s na parametri $r_{cut} = 4.5, 4, 3.5 \mu m$.

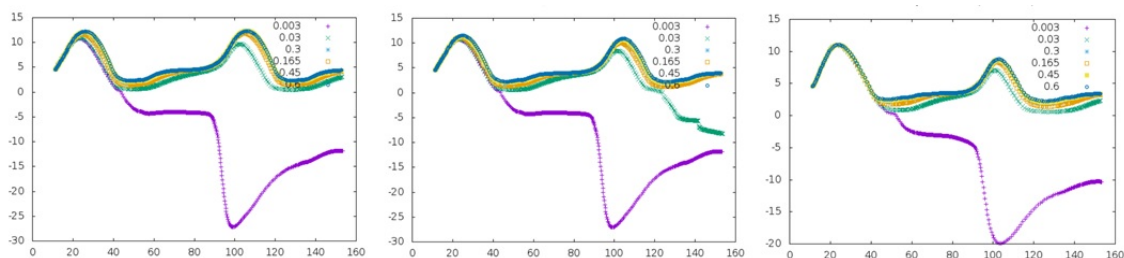
Celkovo tento prístup funguje dobre vo väčších zúženiach. No pre menšie zúženia je prístup s viazanými interakciami vhodnejší.

prierez	r_{cut}/a	0.003	0.03	0.165	0.3	0.45	0.6
13	3.5	0	0	0	0	0	0
13	4	0	0	0	0	0	0
13	4.5	0	0	0	0	0	0
15	3.5	0	0	1	1	1	1
15	4	0	0	1	1	1	1
15	4.5	0	1	1	1	1	1
17	3.5	0	1	1	1	1	1
17	4	0	1	1	1	1	1
17	4.5	0	1	1	1	1	1
19	3.5	0	1	1	1	1	1
19	4	0	1	1	1	1	1
19	4.5	0	1	1	1	1	1

Obrázok 4.13: Výsledky simulácií v ktorých bola použitá soft-sphere interakcia.



Obrázok 4.14: Graf zobrazujúci závislosť D_s (vzdialenosť špicov bunky a jadra) od zmeny parametra a pri použití soft-sphere interakcie, $r_{cut} = 3.5\mu m$, bunka prechádza zúžením $13\mu m \times 13\mu m$.



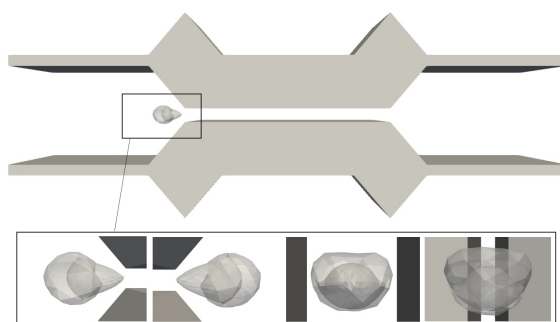
Obrázok 4.15: Grafy zobrazujúce závislosť D_s (vzdialenosť špicov bunky a jadra) od zmeny parametra a pri použití soft-sphere interakcie. Hodnota r_{cut} je postupne (zľava doprava) 4.5, 4, 3.5. Bunka prechádza zúžením $15\mu m \times 15\mu m$.

4.4 Model bunky s jadrom reprezentovaný elasticou guľou

Modelovať bunku s jadrom je možné aj pomocou samotnej elastickej gule predstavujúcej membránu bunky. Prítomnosť jadra je potom zahrnutá pomocou nastavenia elastických parametrov.

Výhodou pri tomto prístupe je kratší výpočtový čas a menšie množstvo parametrov simulácie. Využitelnosť takéhoto prístupu sa preto ukáže najmä pri simuláciách s veľkým množstvom buniek.

Pri spúšťaní RBC zúženými kanálikmi sme odhalili neprirodzené správanie buniek. Keď bunka prešla úzkou časťou, membrána sa zdeformovala až do takej miery, že membrána prešla sama cez seba. Toto je vidieť na obrázku 4.16.



Obrázok 4.16: Neprirodzené správanie bunky v úzkom kanáli. Bližší pohľad zhora, zdola, zozadu a spredu.

Pri prechode cez úzke kanály membrána prechádzala sama cez seba. Použitím novej odpudivej sily, ktorá pôsobí vždy keď sa dva body bunky priblížia k sebe príliš, sa problém vyriešil pre červené krvinky (RBC). Rovnaký problém však nastával aj pri simuláciách elastických guľôčok. Keďže druhov a veľkostí buniek je veľké množstvo, poskytujeme v tejto časti prehľad o závislostiach medzi parametrami odpudivej sily a veľkosťou, počtom diskretizačných bodov a elasticitou bunky.

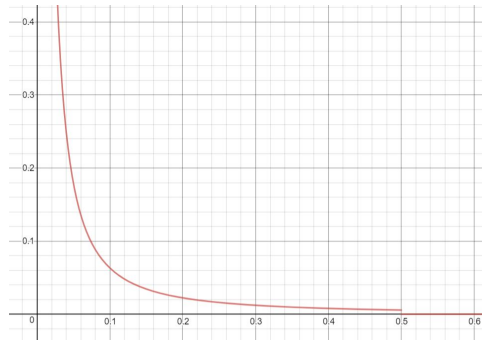
4.4.1 Self-cell interakcia membrány bunky

Prekryv membrány bol vyriešený pridaním interakcie self-cell. Ako názov napovedá ide o interakciu bodov membrány so sebou. Táto interakcia používa rovnaké mechanizmy aké sa používajú pri interakcii bunky a steny. Ide o zabudovanú funkciu

ESPResSo nazývanú interakcia soft-sphere. Podrobnosti je možné nájsť v dokumentácii ESPResSo [17]. Sila tejto interakcie je vypočítaná pomocou vzorca 4.4

Interakcia self-cell bola nastavená pre nakalibrovanú RBC s priemerom $7.92 \mu m$. RBC bunky ktoré používame v našich simuláciách majú povrch $132.94 \mu m$ čo veľmi blízko korešponduje s biologicky nameranými veľkosťami povrchu RBC, v [18] je veľkosť $135 \mu m^2$. Membrána je v simuláciách reprezentovaná trojuholníkovou sieťou. Aby sme správne nastavili parametre pre funkciu odpudivej sily analyzovali sme dĺžky jednotlivých hrán trojuholníkov. Maximálna dĺžka je $0.72 \mu m$, minimálna $0.36 \mu m$ a priemerná je $0.49 \mu m$. Po otestovaní viacerých možností boli vybraté nasledujúce fungujúce parametre.

$$V(r) = 0.002(r - 0)^{-1.5} \quad (4.6)$$



Obrázok 4.17: Funkcia odpudivej sily pre RBC.

Hoci cirkulujúce rakovinové bunky môžu mať aj iné tvary, v literatúre sa často vyskytujú bunkové línie a ich tvar je podľa literatúry prevažne guľovitý. Nezabúdame však na to, že ich veľkosť a mechanické vlastnosti môžu byť rôzne. Naším cieľom v tejto časti je preto zistiť ako je potrebné meniť parametre funkcie $V(r)$ v závislosti od meniacich sa vlastností guľí, ktoré budeme používať. Dôležitým faktorom je aj množstvo diskretizačných bodov, keďže toto ovplyvňuje vzdialenosti bodov na sieti.

Predpokladáme, že veľkosť bunky nemusí byť nutne najvýznamnejšia. Dôležitejšia by mala byť dĺžka hrany medzi dvoma susednými bodmi, keďže interakcia by mala ovplyvňovať správanie susedných bodov.

Tu poskytujeme prehľad dĺžok hrán pre rôzne diskretizácie povrchu gule a pre rôzne veľkosti polomerov guľí. Rozhodli sme sa pre polomery $3.91 \mu m$, aby sme

4.4. MODEL BUNKY S JADROM REPREZENTOVANÝ ELASTICKOU GULOU

Tabuľka 4.2: Dĺžka hrán pri rôznych diskretizáciách buniek, $r = 3.91\mu m$

Počet bodov	169	304	482	514	642	1922	2050
Počet trojuholníkov	334	604	960	1024	1280	3840	4096
Povrch	186.87	189.87	190.9	190.53	191.2	191.8	191.72
min. hrana	0.76	0.49	0.64	0.47	0.54	0.32	0.23
max. hrana	1.65	1.22	0.75	1.18	0.643	0.38	0.6
priemerná hrana	1.15	0.86	0.68	0.69	0.59	0.34	0.34
vypočítaná hrana	1.15	0.86	0.68	0.66	0.59	0.34	0.33

Tabuľka 4.3: Dĺžka hrán pri rôznych diskretizáciách buniek, $r = 8\mu m$

Počet bodov	169	304	482	514	642	1922	2050
Počet trojuholníkov	334	604	960	1024	1280	3840	4096
Povrch	787.29	794.86	799.17	797.61	800.42	802.97	802.58
min. hrana	1.56	1.01	1.3	0.95	1.11	0.65	0.48
max. hrana	3.38	2.5	1.53	2.41	1.32	0.77	1.21
priemerná hrana	2.36	1.76	1.39	1.4	1.21	0.7	0.7
vypočítaná hrana	2.36	1.75	1.39	1.35	1.20	0.70	0.67

mali guľu porovnateľnej veľkosti ako je RBC, a $8\mu m$ a $12\mu m$, keďže to je rozsah veľkostí, ktoré sme si na základe literatúry vymedzili na skúmanie.

Pre každú kombináciu sme analyticky vypočítali aká by mala byť priemerná dĺžka hrany pri danom polomere gule, počte trojuholníkov a fakte, že trojuholníky by mali v ideálnej triangulácii byť rovnostranné. Toto číslo nám dáva porovnávaciu hodnotu, na základe ktorej môžeme zistiť, ktoré diskretizácie sú lepšie. Ako vidíme v tabuľkách 4.2, 4.3 a 4.4, ideálna dĺžka hrany vypočítaná z analytického modelu gule a priemerná dĺžka hrán z jednotlivých diskretizácií sú porovnateľné. Vhodnosť danej diskretizácie môže byť determinovaná na základe porovnávania rozdielu medzi minimálnou a maximálnou dĺžkou hrán v danej diskretizácii.

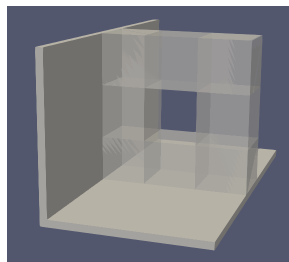
Tabuľka 4.4: Dĺžka hrán pri rôznych diskretizáciách buniek, $r = 12\mu m$

Počet bodov	169	304	482	514	642	1922	2050
Počet trojuholníkov	334	604	960	1024	1280	3840	4096
Povrch	1760.2	1788.4	1798.1	1794.6	1800.9	1809.6	1805.8
min. hrana	2.34	1.51	1.95	1.43	1.66	0.98	0.72
max. hrana	5.07	3.75	2.29	3.62	1.98	1.15	1.83
priemerná hrana	3.53	2.64	2.09	2.1	1.81	1.05	1.06
vypočítaná hrana	3.54	2.63	2.09	2.02	1.81	1.04	1.01

4.4.2 Elasticita a Self-cell interakcia

Aby sme mohli pozorovať vplyv elasticity definovali sme si tri sady elastických parametrov. Prvá sada bola "elastická" a parametre sme nastavili na hodnoty elastických parametrov zdravých RBC. Druhá sada boli "stredne elastické" parametre, kde parameter pre natáhanie, ohýbanie a zachovanie relaxovanej plochy boli desať násobne vyššie ako pri "elastickej" sade. "Tuhou" sme nazvali sadu, ktorá mala desať násobne vyššie parametre ako "stredne elastická" sada.

Aby sme otestovali fungovanie self-cell interakcie, vytvorili sme simulácie, kde bunka prechádzala cez zúženie v rovnom kanáliku. Rozhodli sme sa použiť zúženia, ktoré boli čiastočne menšie ako samotná bunka. Pre polomer $3,92 \mu m$ sme použili zúženie s prierezom s rozmermi $7 \times 7 \mu m$. Pre $8 \mu m$ polomer to bolo zúženie s prierezom $15 \times 15 \mu m$ a pre $12 \mu m$ bol prierez $23 \times 23 \mu m$. Kanálik je zobrazený v obrázku 4.18.

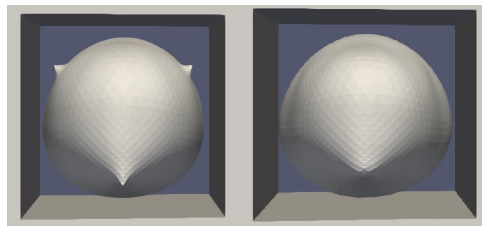


Obrázok 4.18: Kanálik so zúžením.

Spustili sme simulácie pre všetky kombinácie rozmerov bunky, diskretizácie a troch sád elastických parametrov, ktoré sme spomenuli vyššie s hodnotami parametru self-cell interakcie r_cut nastavenému na minimálnu, maximálnu a priemernú dĺžku hrany danej bunky. S parametrami $a = 0.002$ a $n = 1.5$. Taktiež pre porovnanie sme spustili tieto simulácie bez self-cell interakcie.

Pre diskretizácie s 482, 642 a 1922 bodmi fungovali všetky nastavenia pre r_cut pomerne dobre. V prípade, kde bol r_cut nastavený na maximálnu dĺžku sa na bunke pri zúžení vytvorili len veľmi malé špice. Znatelnejší bol rozdiel pri zvyšných štyroch diskretizáciách. Pre hodnoty r_cut väčšie ako minimálna dĺžka hrany danej bunky sa vytvorili viditeľné špičky na bunke, jeden príklad vidíte na obrázku 4.19.

Po správnom nastavení parametra r_cut , sme ďalej spustili simulácie s rôznymi nastaveniami pre a a n , aby sme dosiahli rovnaký tvar krivky, aká bola nakalibrovaná pre RBC, viď 4.17 pre normalizované $r = r/r_{cut}$. Tieto výsledky boli veľmi podobné



Obrázok 4.19: Vplyv nesprávneho nastavenia hodnoty r_{cut} na tvar bunky. Na ľavej strane môžeme pozorovať špice, ktoré sa vytvorili. Na pravej strane s lepšie nastaveným r_{cut} je membrána hladšia.

výsledkom so simulácií s nastaveniami $a = 0.002$ and $n = 1.5$.

V simuláciách s elastickými, stredne elastickými a tuhými bunkami sa bunky deformovali na základe nastavenia elasticity. Výsledky však ukazujú, že elasticita nemá žiadny vplyv na nastavenie parametrov self-cell interakcie. Membrána bunky nevytvorila viditeľné špice a preto celkový výsledok nasvedčuje, že parametre funkcie $V(r)$ boli nastavené správne a ich zmena neponúkla zlepšenie kvality správania membrány bunky.

Dĺžky hrán diskretizácie membrány bunky sú najdôležitejším faktorom pri nastavovaní funkcie $V(r)$ self-cell interakcie. V prípade používania pravidelnejších triangulácií buniek (kde minimálna, maximálna a priemerná dĺžka hrany sa líšia minimálne) je najvhodnejšie nastaviť hodnotu r_{cut} na dĺžku priemernej dĺžky hrany danej bunky.

Očakávali sme, že bude potrebné upraviť a a n takým spôsobom, aby hodnoty $V(r)$ boli rovnaké pri normalizovaní vzdialeností bodov (r) pomocou priemernej alebo minimálnej dĺžky danej diskretizácie. Z výsledkov simulácií sme zistili, že to nie je potrebné. Pre náš rozsah veľkostí guľí stačí použiť parametre $a = 0,002$ a $n = 1,5$. V prípade väčších buniek s hranami s rádovo väčšími dĺžkami ako sú tu uvedené, odporúčame otestovať aj väčšie hodnoty parametrov a porovnať výsledky pred konečným nastavením self-cell interakcie.

4.5 Porovnanie prístupov

Aby sme mohli porovnať rozdiely medzi modelovaním bunky s jadrom a modelovaním iba samotnej membrány bunky, spustili sme sadu simulácií s rovnakými parametrami kanálíka a elastických koeficientov membrány aké boli použité v častiach 4.2 a 4.3. Hoci sme otestovali viacero parametrov pre self-cell interakciu membrány,

tak podobne ako pri použití soft-sphere interakcie v časti 4.3, žiadna zo sád neprešla zúžením $13 \times 13 \mu m$. Pri ostatných rozmeroch zúžení už prešli všetky.

V obrázku 4.20 zobrazujeme sadu grafov pre simulácie kanálik so zúžením $15 \times 15 \mu m$. Na x -ovej osi sú uvedené x -ové súradnice stredu buniek, na y -ovej osi je hodnota DI bunky.

Výsledky simulácií používajúcich viazané interakcie sú zobrazené vo vrchnej časti obrázku. Postupne zľava doprava harmonický, kvadratický a FENE potenciál. Pri týchto simuláciách bola hodnota parametrov nasledujúca: $a = 20$, $k_1 = 0.0001$, $\Delta r_{max} = 8$. Rôzne krivky v grafoch predstavujú rozdielne nastavenie parametra k pre harmonický potenciál, parametra k_0 pre kvadratický potenciál a K pre FENE potenciál.

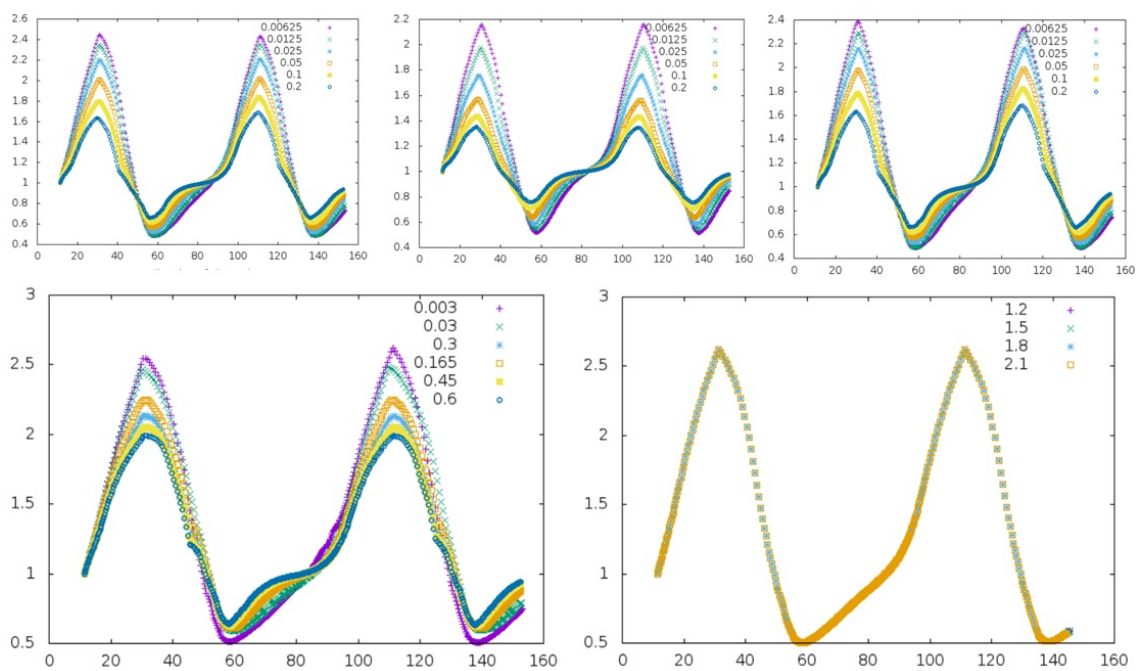
V spodnej časti obrázku sú na ľavej strane zobrazené hodnoty DI pre simulácie využívajúce neviazanú soft-sphere interakciu. V týchto simuláciách boli nastavené nasledujúce hodnoty parametrov: $r_{offset} = 0$, $n = 3$, $r_{cut} = 4.5$. V grafe sú zobrazené krivky vývoja DI pri rôznych hodnotách parametra a . Na pravej strane je potom uvedený graf pre simulácie modelujúce membránu bunky s použitím self-cell interakcie. Self-cell interakcia je nastavená pomocou parametra $a = 0.001$ a n , ktorý má postupne hodnoty 1.2, 1.5, 1.8, 2.1. Vidíme však, že n v týchto simuláciách nemalo žiadny efekt.

Pri porovnaní maximálnych dosiahnutých DI indexov vidíme pri jednotlivých simuláciách vplyv prítomnosti jadra na elasticitu bunky. Všetky zobrazené simulácie využívajúce viazané interakcie nedosahujú hodnotu $DI = 2.5$, ktorú dosahuje samotná membrána bez jadra. Dôvodom je, že aj najslabšie nastavená interakcia medzi membránou a jadrom spôsobuje zvýšenie tuhosti celkovej bunky.

Pri neviazaných interakciách (ľavý dolný graf), si môžeme všimnúť, že fialová krivka sa zhoduje s krivkou DI pre membránu bunky (pravý dolný graf). Z výsledkov uvedených v časti 4.3 vieme, že pri tejto simulácii jadro uniklo z membrány. To vysvetľuje prečo jadro nemá vplyv na deformáciu tejto membrány.

Nech už je použitá akákoľvek implementácia väzby jadra v bunke, vidíme, že čím je pevnejšia tým menej sa bunky deformujú. Táto analýza citlivosti jednotlivých parametrov a prístupov posluží v budúcnosti pri nastavovaní parametrov pre konkrétne typy buniek.

4.5. POROVNANIE PRÍSTUPOV



Obrázok 4.20: Porovnanie DI bunky pri použití rôznych spôsobov modelovania bunky s jadrom. Bunka prechádza zúžením $15 \mu\text{m} \times 15 \mu\text{m}$.

5 Simulácie biologických experimentov

5.1 Dáta z biologických experimentov a ako s nimi pracovať

Získavanie dát z experimentov pre proces kalibrácie je pomerne náročné. Existujú dve možnosti, buď získať dáta z publikovanej práce alebo nadviazať spoluprácu s laboratóriom uskutočňujúcim experimenty. Ako bolo ukázané v predchádzajúcich častiach, získavanie informácií z už publikovanej práce, nesie so sebou viacero výziev. V prvom rade je možné, že niektoré dáta zaujímavé z pohľadu simulačných experimentov budú nedostupné, pokiaľ neboli zaujímavé z pohľadu biologického experimentu. Medzi ďalšie určite patrí istá miera straty informácie, keďže v biologických štúdiách sú zaznamenávané dáta z opakovaných experimentov, kde si experimentátori vyberú konkrétne bunky, ktoré ich zaujímali, tieto bunky nemusia a ani nie sú nutne identické a teda získané dáta sú len informáciou o priemernom správaní buniek a nie o konkrétnej bunke konkrétnych rozmerov. Preto porovnávanie simulačných výsledkov s takýmito dátami nesie so sebou nutnosť aproximácie hodnôt, ktoré chceme porovnávať. Pri spolupráci s laboratóriom je možné ovplyvniť aké informácie sú zaznamenávané. Tu však prichádzajú iné typy výziev. V rámci výskumného pobytu v Laboratory of Interfacial plus Microscale Phenomena from IN+ - Center for Innovation, Technology and Policy Research na Instituto Superior Técnico, Universidade de Lisboa boli zaznamenané experimenty s mikročasticami prechádzajúcimi cez mikrofluidické kanáliky. Následne boli uskutočnené a zaznamenané ďalšie experimenty s RBC prechádzajúcimi cez rôzne mikrofluidické zariadenia v spolupráci s Metrics research center na University of Minho. V rámci tejto práce bolo nutné riešiť otázky toho čo presne je potrebné zaznamenať a akým spôsobom

5.1. DÁTA Z BIOLOGICKÝCH EXPERIMENTOV A AKO S NIMI PRACOVAŤ

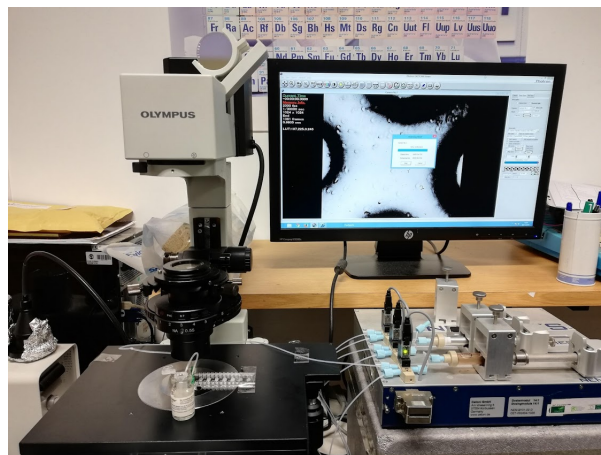
následne spracovať videá z týchto experimentov.

V tejto časti zhrnieme náplň tohto výskumného pobytu a aký prínos a poučenie je z neho možné vyvodiť pre kalibračný proces buniek s jadrami.

5.1.1 Popis biologického experimentu

Všetky experimenty s bunkami boli vykonané v laboratóriu na University of Minho. Bunková suspenzia použitá v experimentoch bola suspenzia ľudských červených krviniek (RBC) v Dextráne 40 (Dx40) s hematokritom (Hedct) 0,2 %. Krv od zdravých jedincov bola získaná z Instituto Português do Sangue e da Transplantação (IPST). RBC boli oddelené od celej krvi odstredením (1500 rpm po dobu 15 minút) a potom bola plazma odstránená odsátím. RBC sa potom premyli fyziologickým roztokom chloridu sodného a proces sa opakoval ešte dvakrát. Nakoniec sa RBC riedili Dx40, aby sa dosiahla požadovaná koncentrácia.

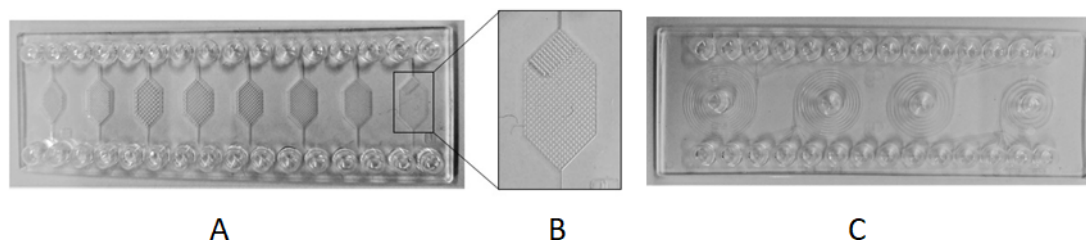
Prietok pracovnej tekutiny sa analyzoval pomocou systému tvoreného inverzným mikroskopom (IX71, Olympus) a vysokorýchlostnou kamerou (FASTCAM SA3, Photron). Mikrofluidické zariadenia sa umiestnili na stolček invertovaného mikroskopu. Na vytvorenie konštantného prietoku sa použila injekčná pumpa (CetoniNEMESYS Syringe Pump). Celý systém je zobrazený na obrázku 5.1.



Obrázok 5.1: Zaznamenávanie biologického experimentu

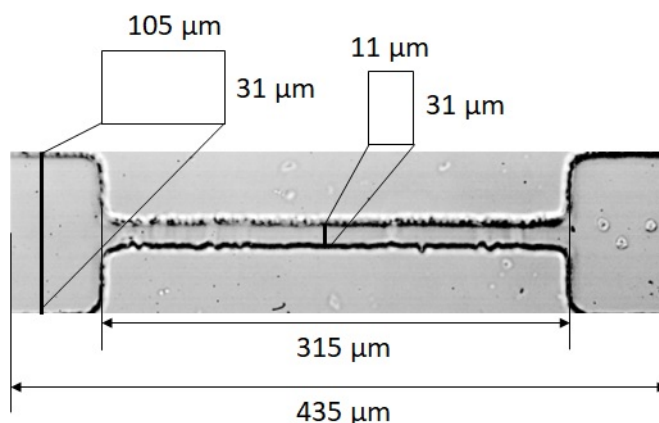
Uskutočnené boli experimenty na 4 druhoch mikrofluidických čipoch. Dva z čipov, špirálový (obrázok 5.2, C) a pilierový čip (obrázok 5.2, A a detail B), boli objednané z microfluidic Chipshop. Zvyšné dva boli vyrobené v Metrics research center na University of Minho. Tieto dva čipy obsahovali viacero mikrofluidických

kanálikov. Typovo však boli rovnaké - dlhý rovný kvádrový kanál so zúžením v strede. Líšili sa svojou dĺžkou, výškou a veľkosťou zúženia.



Obrázok 5.2: Chipy použité na experimenty.

Tieto čipy boli vyrobené mäkkou litografickou technikou. Na výrobu hlavných foriem sa v AutoCAdE nakreslili geometrie, aby sa vytvorila fotomaska s vysokým rozlíšením. Forma bola potom vyrobená na kremíkovej doštičke s ultratenkým fotorezistom (SU-8 50; Kayaku MicroChem, Japonsko). Prepolymer polydimethylsiloxane (PDMS) bol pripravený zmiešaním komerčného prepolymeru a vytvrdzovacieho činidla (Silpot 184; Dow Corning, USA) v hmotnostnom pomere 10: 1. Zmes bola odplynená vo vákuu a PDMS bola naliata do základnej formy SU-8 z fotorezistu a vytvrdená v peci asi 2 hodiny pri 70°C. Potom, čo bola hlavná aj PDMS forma ochladená na izbovú teplotu, bol PDMS odlúpnutý z hlavnej formy. Vstupné a výstupné porty sa vyrábajú pomocou špičiek na pipety. Nakoniec bol PDMS premytý etanolom a upevnený odstrednením na čisté sklíčko. Mikrokanály sa vyrábali v troch rôznych výškach: 31 μ m, 25 μ m a 12 μ m. Geometria mikrokanála s najvyššou výškou je uvedená na obrázku 5.3.



Obrázok 5.3: Rozmery kanálka so zúžením.

5.1.2 Mikrofluidické kanálky so zúžením

Tento typ mikrofluidických kanálikov bol vyrobený s cieľom študovať deformáciu jednotlivých buniek. Pre začiatok sme sledovali deformáciu zdravej RBC. Analýza týchto dát sa nachádza v ďalšej časti tejto kapitoly.

Do budúca je plánovaná ďalšia spolupráca. V súčasnosti čakáme na dostupnosť ďalších typov buniek. Jeden z predpokladaných typov buniek sú červené krvinky nakazená parazitom malárie *Plasmodium falciparum*, čo spôsobuje, že bunky sú tuhšie oproti zdravým [2]. Vzhľadom na to, že tieto parazity sa nachádzajú vnútri bunky, bude možné pre modelovanie takejto biologickej situácie využiť náš model, podobne ako v práci "Modeling of hemodynamics arising from malaria infection" [40].

Získanie meraní o ďalších typoch buniek s jadrami pre validáciu nášho modelu je ovplyvnené limitáciami spôsobenými s náročnosťou produkcie bunkových suspenzií. Bunky sú náchylné k zániku počas prípravy suspenzie. Taktiež je náročné pracovať s 3D suspenziami, keďže bunky prirodzene adherujú ku stenám mikrofluidických kanálikov [30]. Toto správanie je odstraňované rôznymi ošetreniami stien kanálikov, ktoré však sťažujú proces získania meraní a taktiež môžu spôsobiť zmeny vo vlastnostiach buniek. Najst' správnu kombináciu všetkých týchto parametrov pri realizovaní biologického experimentu je preto náročné a získanie dát pre výpočtový model si vyžaduje dlhší čas.

Náročná však vie byť aj výroba mikrofluidických zariadení. Metrics research center na University of Minho už vytvorilo podobné geometrie [5]. Cieľom v súčasnosti je znížiť výšku kanálikov a tak dosiahnuť tok buniek v jednej vrstve, čo by malo umožniť lepšie zaostrenie videozáznamov. Pri väčšej výške sa totiž v kanáliku môžu nachádzať bunky vo viacerých vrstvách. Pri zaznamenávaní videa z mikroskopu sa však zaostruje iba na jednu z týchto vrstiev, čo často vedie k menej čistým videám, ktoré sú ťažšie na spracovanie. Výroba kanálikov z nižšou výškou je však zdlhavejšia, keďže tieto častejšie kolabujú a tak je nutné skúšať rôzne spôsoby ich výroby.

Počas výskumného pobytu bolo zaznamenaných zhruba 40 videí. V ďalšej časti sa zameriame na jedno z týchto videí a na spôsob akým sme modelovali tento experiment.

5.1.3 Spracovanie biologických dát a výpočtový model

Na obrázku 5.5 môžeme vidieť 12 rôznych buniek, ktoré prešli zúženým kanálikom počas jedného záznamu z biologického experimentu. V tomto prípade, ale aj v prípade iných simulácií je potrebné nastaviť správne tok, a ak máme nakalibrované potrebné elastické parametre, môžeme pozorovať zhodu v správaní bunky v biologickom experimente a v simulačnom experimente.

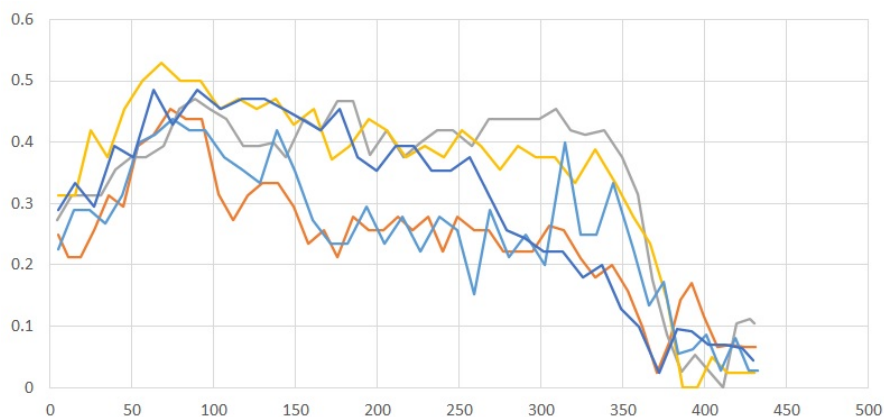
Zo záznamu experimentu sme získali informácie o rýchlosti jednotlivých buniek, o čase prechodu zúžením a o deformáciách buniek. Vzhľadom na nižšiu kvalitu videa, nebolo možné použiť automatické spracovanie obrazu. Bolo nutné ručne anotovať všetky bunky. Použili sme ImageJ software, kde sme každej bunke opísali obdĺžnik každých $8\mu m$, tak aby sa jednotlivé obdĺžniky pre jednu bunku neprekrývali. Nazvali sme strany tohto opísaného obdĺžnika a , v smere x -ovej osi, a b , v smere y -ovej osi. Deformačný index bunky sme potom vypočítali ako $DI = a/b$.

Profil rýchlosti buniek bol porovnateľný. Priemerná rýchlosť, ktorú dosahovali v zúžení bola $4,5mm/s$. Tento fakt je spôsobený konštrukciou kanálíka. Bunka je ťahaná tokom a musí sa prispôbiť jeho rýchlosti. Z toho vyplýva, že aj čas prechodu zúžením bol porovnateľný pre jednotlivé bunky, a to $0,07s$. Čas prechodu vie byť zaujímavým ukazovateľom pre bunky s rôznym stupňom tuhosti. Tento triediaci faktor popisujeme v časti 5.2. Avšak napriek zhode rýchlosti a tomu, že išlo o bunky rovnakého typu, zaznamenali sme rozdiely v tom ako sa bunky deformovali. Na obrázku 5.4 vidíme vývoj deformácie. Hoci v niektorých oblastiach kanálíka sa DI jednotlivých buniek zhodujú (napríklad okolo oblasti výstupu zo zúženia kanálíka, hodnota $375\mu m$ na x -ovej osi), v iných sa ich hodnota líši aj dvojnásobne.

Je potrebné vyhodnotiť, s akými hodnotami DI chceme porovnávať simulačné výsledky a ako má vyzerat' simulácia, ktorá sa zhoduje s biologickým experimentom. Jedno z riešení je porovnať simuláciu len na základe profilu rýchlosti bunky. Na vytvorenie prietoku bola použitá injekčná pumpa ovládaná počítačovým softwarom a vieme, že prietok suspenzie v skúmanom zázname bol $2.5\mu L/min$. Po spustení simulácie s daným prietokom sme získali výsledky s hodnotami rýchlosti aj trojnásobne vyššími ako mali biologické bunky.

Dôvodom tejto nehody sú malé straty tekutiny v miestach napojenia injekčnej striekačky a vstupu do mikrofluidického zariadenia, a prípadné vzduchové bubliny. Pre správne nastavenie toku v simulácii je preto nutné využiť informáciu o rýchlosti bunky. Nastavením hodnoty fluid force density na 0.0003 sme získali rovnaký profil

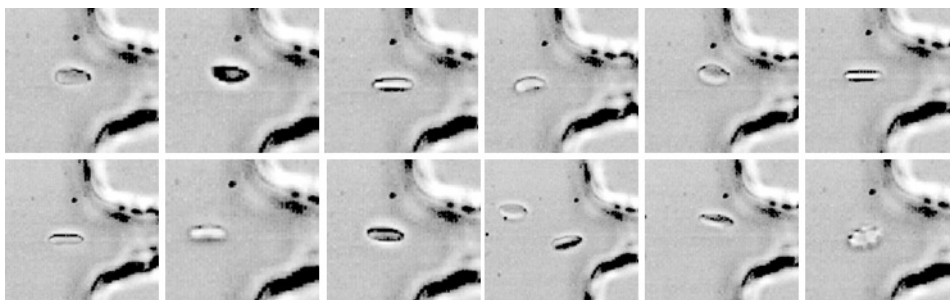
5.1. DÁTA Z BIOLOGICKÝCH EXPERIMENTOV A AKO S NIMI PRACOVAŤ



Obrázok 5.4: Porovnanie deformačného indexu (DI) 5 buniek z biologického experimentu. Na x -ovej osi je uvedená hodnota x -ovej súradnice stredu bunky (v μm) a na osi y jej DI.

rýchlosti.

Pre porovnanie simulácie a biologického experimentu potrebujeme DI aby sme mohli povedať, že pri rovnakom nastavení toku sa nakalibrovaná RBC bunka správa rovnako ako priemerná zdravá bunka z experimentu. V obrázku 5.5 vidíme rozdiely medzi jednotlivými bunkami. Jednak rôzne natočenie, ale aj viditeľné rozdiely v štruktúre buniek. Tieto rozdiely sú pravdepodobne dôvodom odlišných profilov DI. Aby sme mohli vytvoriť porovnanie, potrebujeme väčšiu štatistickú vzorku ako 12 buniek.



Obrázok 5.5: Pozície jednotlivých buniek z biologického experimentu.

Dĺžka záznamu a teda aj počet buniek je ovplyvnený limitáciami použitej techniky. Viac buniek môžeme získať ak vieme dlhšie zaznamenávať video, alebo ak znížime FPS, alebo ak zrýchlime tok. Rýchlosť toku musí byť nastavená tak, aby bolo možné zaznamenať pohyb buniek na kameru. Zaostrenie kamery určuje akú veľkosť zariadenia vieme zaznamenávať. Napríklad detailnejší obraz vieme získať ak

sa zameriame len na časť zariadenia, ale potom nevieme určiť niektoré parametre, napríklad čas prechodu bunky.

Tento experiment bol začiatkom spolupráce s laboratóriom na University of Minho. Okrem nadviazania spolupráce bolo prínosom tejto práce pripravenie simulácií a ich otestovanie pre RBC. Simulácie sa môžu následne využiť po získaní ďalších dát pri spolupráci v budúcnosti. Pri zmene výšky kanálíka, stačí zmeniť jeden z parametrov pythonovského skriptu. Taktiež miesto RBC je možné použiť model bunky s jadrom pri biologických experimentoch s inými typmi buniek.

5.2 Mikrofluidický kanál so zúžením

Pri hľadaní vhodných biologických experimentov pre kalibráciu nášho modelu sme vybrali experiment od Houa [37]. V tomto experimente skúmali deformovateľnosť buniek rakoviny prsníka prechádzajúcich cez úzky mikrofluidický kanál. Kanál má podobné rozmery ako krvné kapiláry, aby napodobnil prostredie in vivo. Aby sme dosiahli podobné správanie modelu bunky a skutočnej bunky v experimente, potrebovali sme nastaviť prietok tekutiny podľa experimentálnych údajov. Informácie o toku boli zadané pomocou rozdielu tlakov. Keďže neboli publikované kompletne údaje o rozmeroch kanálíka, museli sme prietokový objem aproximovať na základe dostupných informácií. Hoci takto získané výsledky nám nevedeli poskytnúť kvantitatívne zaujímavé výsledky, slúžili na prvotné získanie sád fungujúcich parametrov. Na základe článku sme vedeli aké veľkosti buniek a akou rýchlosťou by mali byť schopné prejsť cez kanálik daných veľkostí.

Cieľom experimentu bolo nájsť rozdiely medzi malígnymi a nemalígnymi bunkami. Mikrofluidické zariadenie vytvorené na tento účel pozostáva z úzkeho kanála a zásobníka na každom konci. Úzka časť tohto kanála má dĺžku $150 \mu m$ a jeho prierez je štvorec s veľkosťou $10 \times 10 \mu m$. Tvar kanála je znázornený na obrázku 5.6.

V experimente boli použité bunkové línie *MCF-10A* (zdravé bunky použité ako referenčný rámec) a línia *MCF-7* (ktorá predstavuje rakovinové bunky). Rozdiel medzi týmito dvoma bunkami bol meraný tromi parametrami: **vstupný čas** - čas potrebný na úplné vstúpenie bunky do úzkej časti kanála, **čas prechodu** - čas od okamihu, keď je bunka kompletne vnútri úzkej časti až kým ju neopustí, a **elongation index (EI)** (index predĺženia). Index predĺženia je definovaný ako $EI = D_L/D$, kde D_L je dĺžka deformovanej bunky vo vnútri úzkej časti, a D je počiatočný priemer bunky pred vstupom do úzkej časti. Najvýraznejší rozdiel medzi rakovinovými a

5.2. MIKROFLUIDICKÝ KANÁL SO ZÚŽENÍM



Obrázok 5.6: Mikrofluidický kanál so zúžením.

zdravými bunkami namerali v čase vstupu. Veľkosti buniek použité v článku sa po-

Tabuľka 5.1: Hodnoty namerané v biologickom experimente [37].

Typ bunky	čas vstupu	EI	Priemerná rýchlosť ($\mu m/s$)
MCF-7	0,433	1,281	177,3
MCF-10A	1 698	1 231	187,0

hybovali v rozmedzí od 15 do 23 μm . Pre túto kalibráciu sme vybrali veľkosť bunky 16 μm , pretože je to veľkosť použitá v iných článkoch a mohla by sa použiť pri overovaní našej kalibrácie.

Pre túto veľkosť bunky bol index predĺženia 1,15 μm a priemerná rýchlosť bunky v zúžení bola 175 μm . Vstupný čas bol pre túto veľkosť kratší v porovnaní s priemerom, ktorý je 0,15 sekúnd. Tieto hodnoty sme použili pri hľadaní elastických parametrov pre model rakovinej bunky prsníka s polomerom 16 μm . Najväčšiu pozornosť sme venovali vstupnému času, keďže zvyšné namerané hodnoty významne nerozlišovali medzi zdravými a rakovinovými bunkami.

5.2.1 Nastavenie simulácie

V tomto simulačnom experimente bunky prešli relatívne úzkym kanálom, čo malo za následok veľké zaťaženie bunky. Pri spúšťaní prvých sád simulácií, v mieste kanáliku, kde bola rýchlosť tekutiny najväčšia, sa bunka neprirodzene zdeformovala. Bunková membrána prešla cez seba, čo by sa v biologických experimentoch nestalo. Na obrázku 4.16 vidíme, ako vyzerá táto situácia.

Preto bolo potrebné do simulácií pridať interakciu medzi membránou samotnou. Túto interakciu sme nazvali **self-cell interakcia** a jej úlohou je kontrolovať, ako blízko sú body membrány. Ak sú príliš blízko a existuje riziko, že membrána prejde cez seba, začne pôsobiť odpudivá sila.

Rovnaká mechanika sa používa na odpudenie dvoch buniek od seba. Odpudivá sila je funkciou vzdialenosti dvoch bodov, definovaná ako 5.1.

$$V(d) = ad^{-n}, d < d_{cut}, \quad (5.1)$$

kde d je vzdialenosť dvoch bodov na membráne, d_{cut} je prahová hodnota, pri ktorej začína odpudzujúca sila pôsobiť, a je škálovací parameter a n určuje, aká prudká je reakcia, keď sa dva body blížia k sebe. Je potrebné skontrolovať aby sa susedné body neodpuďovali keď sú v relaxovanej vzdialenosti. Výsledné parametre sú v tabuľke 5.2.

Tabuľka 5.2: Parametre self-cell interakcie

a	n	d_{cut}
0,2	1	6

Následne sme nastavili všetky potrebné koeficienty tekutín tak, aby prietok tekutiny aproximoval prietok z biologického experimentu, na základe poskytnutých informácií. Príslušné hodnoty sily tekutiny sú uvedené v tabuľke 5.8. Ako východiskový bod sme použili simuláciu, pri ktorej je naša bunka CTC nastavená s elastickým parametrom použitým pre bunky RBC. Tabuľka 5.3 zobrazuje všetky parametre potrebné na spustenie simulácie. Táto simulácia však mala pomerne dlhý výpočtový čas, preto sme sa rozhodli nájsť rýchlejšie alternatívy. Naším cieľom bolo spustiť sadu simulácií, kde sme vždy zmenili len jeden z elastických parametrov a na základe hodnôt času vstupu bunky do úzkej časti kanála sme určili, aký rozsah parametrov je funkčný pre modelovanie CTC buniek.

5.2.2 Vstupný čas v závislosti od sily tekutiny

Na získanie údajov o závislosti medzi vstupným časom a silou tekutiny sme spustili niekoľko simulácií. Možno ich vidieť v tabuľke 5.3.

Tieto dáta sme potom fitovali exponenciálnou funkciou vo forme $F(x) = ae^{bx}$. Kde $F(x)$ označuje vstupný čas, x je veľkosť sily tekutiny a parametre a a b určujú tvar krivky.

5.2. MIKROFLUIDICKÝ KANÁL SO ZÚŽENÍM

Tabuľka 5.3: Parametre simulácie

Časový krok	$10^{-7} s$
LBgrid	$10^{-6} m$
fluid force density	$2,85 \cdot 10^4 N/m^3$
k_s	$8 \cdot 10^{-6} N/m$
k_b	$3 \cdot 10^{-12} Nm$
k_{al}	$3 \cdot 10^{-6} N/m$
k_{ag}	$9 \cdot 10^{-4} N/m$
k_v	$5 \cdot 10^2 N/m^2$
Polomer CTC	$8 \cdot 10^{-6} m$
Parameter šmykového trenia	$3,56 \cdot 10^{-9} Nsm^{-1}$
Hustota tekutiny	$10^3 kg \cdot m^{-3}$
Viskozita tekutiny	$1,5 \cdot 10^{-6} m^2/s$

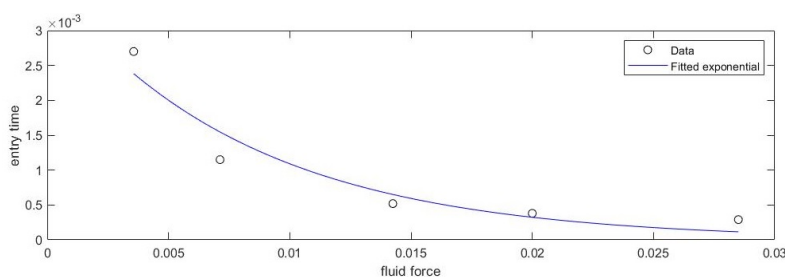
Tabuľka 5.4: Parametre simulácie

Sim	x	ks	kb	kal	F(x)
1003	0,0285	0,008	0,003	0,003	0,00029
004	0,02	0,008	0,003	0,003	0,00038
007	0,01425	0,008	0,003	0,003	0,00052
008	0,007125	0,008	0,003	0,003	0,00115
009	0,003563	0,008	0,003	0,003	0,0027

Príslušné hodnoty parametrov a a b boli nájdené pomocou funkcie Matlab lsqcurvefit [52], ktorá využíva metódu najmenších štvorcov. Problém optimalizácie je definovaný ako

$$\min_x \sum_i (F(x, xdata_i) - ydata_i)^2, \quad (5.2)$$

kde v našom prípade $xdata$ je sila tekutiny a $ydata$ je pozorovaný vstupný čas pre dané hodnoty sily tekutiny. Výsledky boli $a = 0,0037$ a $b = -121,4321$. Vidíme ich aj na obrázku 5.7.



Obrázok 5.7: Závislosť vstupného času od sily tekutiny.

Takže pri sile tekutiny = 0,0000285, čo zodpovedá toku tekutiny z biologického experimentu, bol čas vstupu do bunky s elastickými koeficientmi, ktoré sme stanovili, 0,003687s. Čo je rýchlejšie ako hodnota 0,15s, ktorú namerali počas biologického experimentu. Získali sme tak štartovaciu hodnotu pre bunku s elastickými koeficientami používanými pre RBC.

5.2.3 Kalibrácia parametrov

Ďalším krokom bola kalibrácia piatich elastických parametrov. Postupne sme menili vždy iba jeden parameter a testovali sme hranice kým simulácia počítala. Výsledkom bolo, že dva globálne parametre, parameter zachovania objemu a parameter globálnej plochy, nemali veľký vplyv na vstupný čas. Nastavili sme ich teda na $kag = 0,9$ a $. Výsledky pre ďalšie tri parametre nám dali intervaly, ktoré pracovali pre simulácie. Uvádzame ich v tabuľke 5.5.$

Tabuľka 5.5: Fungujúce elastické parametre

	ks	kb	kal
min	0,004	0,0015	0,0015
max	1,6	0,6	1,5

5.2.4 Objemový prietok a sila tekutiny

Pri nastavovaní prietoku v simuláciách je jedným z dôležitých krokov nastavenie správneho prietoku tekutiny. V práci [44] sme overili lineárnu závislosť medzi hodnotou sily tekutiny a objemovým prietokom. Pomocou sady simulácií sme si overili ako funguje nastavenie sily tekutiny (názov tohto parametra v simuláciách je fluid force density). Testovaním rôznych diskretizácií tekutiny s rovnakou silou tekutiny sme získali rôzne hodnoty objemových prietokov, ako vidíme v tabuľke 5.6. Dôvod je

Tabuľka 5.6: Testovanie odchýlok LBgridu s rovnakou silou tekutiny

LBgrid	fluid force density	objemový prietok
1	0,1	67
2	0,1	34,25
4	0,1	19,375

ten, že v našich simuláciách parameter "fluid force density" upravuje silu pôsobiacu na každý bod diskretizácie, a teda čím je menšia hodnota LBgridu, tým viac bodov

5.2. MIKROFLUIDICKÝ KANÁL SO ZÚŽENÍM

je v diskretizácii, a to sa premieta do väčšej sily pôsobiacej celkovo na tekutinu. Objemový prietok je teda väčší pri rovnakej úrovni sily tekutiny. Pretože je niekedy užitočné použiť jemnejší LBgrid, rozhodli sme sa overiť, či existuje lineárna závislosť medzi zmenou LBgridu a hodnotou sily tekutiny pri nastavení rovnakého prietoku v mikrofluidickom kanáli.

Overovali sme hypotézu, že keď je mriežka LB gridu dvakrát jemnejšia, objemový prietok bude pri rovnakej sile tekutiny dvakrát väčší. Ak teda chceme spustiť simuláciu s rovnakým tokom tekutiny v mikrofluidickom zariadení, stačí nastaviť správnu silu tekutiny pri rôznych hodnotách LB gridu. Túto hypotézu sme testovali na súbore troch simulácií. Preskúmali sme, či keď $LBgrid = 1$, a sila tekutiny je x , potom s hodnotou $LBgrid = 2$ bude sila tekutiny $2x$ a s hodnotou $LBgrid = 4$ bude sila tekutiny $4x$, v prípade, že chceme dosiahnuť rovnaký tok tekutiny. Výsledky simulácií sú v tabuľke 5.7. Z výsledkov vidíme, že naša hypotéza sa potvrdila pre

Tabuľka 5.7: Testovanie rôznych hodnôt LBgrid s rôznymi hodnotami sily tekutiny

Lb grid	sila tekutiny	objemový prietok
1	0,01	6,9
2	0,02	6,92
4	0,04	7,68

prechod z $LBgrid = 1$ na $LBgrid = 2$. V prípade, že chceme nastaviť rovnaký objemový prietok pri $LBgrid = 2$, je nutné zvýšiť silu tekutiny použitú pri $LBgrid = 1$ dvakrát. Hypotéza sa nepotvrdila pri prechode z $LBgrid = 1$ na $LBgrid = 4$. Silu tekutiny je potrebné zvýšiť 3,5 krát, a nie 4 krát k dosiahnutiu rovnakého toku. Toto sa overilo spustením ďalšej sady troch simulácií. Výsledky sú v tabuľke 5.8

Tabuľka 5.8: Testovanie vzťahu medzi hodnotami LBgrid a silami tekutiny

LB grid	fluid force density	objemový prietok
1	0,0000285	0,0189
2	0,0000588	0,0188
4	0,0001	0,0186

5.2.5 Ďalšie typy zúžených kanálikov

Tento simulačný experiment bol jeden z prvých, ktorý sme v našej práci zvolili ako vhodný pre kalibráciu. Z dôvodu prístupu iba k publikovaným dátam, bolo nutné

aproximovať niektoré rozmery kanálika na základe informácií uvedených v článku. Z uvedeného dôvodu preto nie je možné uskutočniť podrobnú kalibráciu na týchto biologických dátach. Vytvorené simulácie a dizajn geometrie tohto zariadenia sú vhodné na účely kalibrácie.

Prínosom tohto simulačného experimentu bolo objavenie nebiologického správania sa modelu a odstránenie tohto správania pomocou novej interakcie membrány bunky (self-cell). Taktiež sme získali hodnoty intervalov pre parametre modelu, pri ktorých bunka vedela prejsť daným mikrofluidickým zariadením. Tieto poskytujú začiatočný bod pre presnejšiu kalibráciu.

V prípade získania presnejších dát, ako sú napríklad všetky rozmery kanálika a informácia o toku tekutiny, z podobného biologického experimentu, je možné využiť skripty z vyššie popísaného experimentu. Geometria mikrofluidického kanálika, ako aj samotný skript so simulačným experimentom, sú parametrizované a je ich možné využiť aj pre iné rozmery. Následne je nutné správne nastaviť tok tekutiny, čo je popísané v predošlých častiach. Potom pre kalibráciu konkrétneho typu bunky je možné použiť kalibračný postup popísaný v časti 5.4.

Mikrofluidické zariadenia s rovnými kanálíkmi a rôznymi typmi zúžení patria medzi najviac dostupné pre väčšinu výskumníkov. Zaznamenávanie experimentov, kde je cieľom pustiť suspenziu buniek cez takéto zariadenie, taktiež patrí medzi viac dostupné, ak porovnáваме všetky typy experimentov, ktoré skúmajú deformačné vlastnosti buniek. Výhoda je teda na strane dostupnosti. Avšak, dostupnosť je vyvážená stratou detailnejšej informácie. Pri zložitejších experimentoch, ktoré si vyžadujú náročnejšie dostupné technológie zase môžeme získať presnejšiu informáciu o mechanických vlastnostiach bunky. Ako druhý typ biologického experimentu vhodného na kalibráciu bunky sme preto zvolili laserové ňaťahovanie, ktoré patrí medzi menej dostupné techniky, avšak umožňuje jemnejšiu kalibráciu.

5.3 Laserové ňaťahovanie buniek

Aj v procese kalibrácie RBC buniek sme využívali biologický experiment, ktorý využíval laser pre ňaťahovanie bunky. Išlo však o inú techniku ako popisujeme v tejto časti. Na bunku boli upevnené silikové guľôčky a laser bol aplikovaný na ne. Jedným z dôvodov prečo takýto typ ňaťahovania nie je používaný pre bunky s jadrom, ako sú napríklad bunkové línie, je, že môžu byť tuhšie ako RBC bunky, čo si vyžaduje použitie väčšej sily ako je možné vyvinúť na upevnené silikové guľôčky.

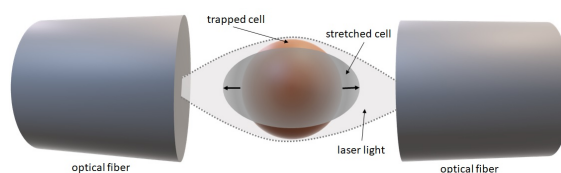
5.3. LASEROVÉ NAŤAHOVANIE BUNIEK

Guck [25] využíva pre naťahovanie buniek optické pasce, nie je preto nutné upevňovať nič na samotné bunky. Toto je výhodou, keďže aj takýto zásah môže skresľovať získané informácie o elasticite buky.

Mikrofluidické zariadenie používané v tomto biologickom experimente vytvára pomocou dvoch rovnako silných laserov optickú pascu. Bunky prúdiace v toku týmto zariadením sa do nej zachytia a následne, za pomoci zvyšujúcej sily lasera, sú bunky naťahované bez nutnosti priamej manipulácie s nimi.

Pomocou optického naťahovača ("optical stretcher") je možné merať viskoelastické vlastnosti buniek s veľkou mierou senzitivity, ktorá umožňuje rozlišovať medzi rôznymi typmi buniek. Táto funkčnosť je využiteľná najmä pre diagnostické účely. Informácie získané o deformácii individuálnych buniek sú výborné aj pre kalibráciu výpočtového modelu, konkrétne elastických parametrov.

Mechanika optickej pasce je založená na pôsobení dvoch laserových lúčov. V experimente s optickou pascou je bunka zachytená medzi dvoma protiľahlými nezaostrenými laserovými lúčmi. Pretože lasery sú identické, celková sila pôsobiaca na bunku je nulová. Povrchové sily sú však aditívne a naťahujú bunku pozdĺž osi laserových lúčov ako je to znázornené na obrázku 5.8.



Obrázok 5.8: Schematické zobrazenie naťahovania pomocou optickej pasce.

Pomocou tejto techniky sa natiahlo niekoľko typov buniek a objektov. RBC bola vybraná ako počiatočný testovací objekt, pretože jej zloženie je menej zložitá ako zloženie eukaryotických buniek. Bolo to hlavne kvôli skutočnosti, že RBC nemá jadro. RBC sú vyplnené homogénnym hemoglobínom a môžu byť osmoticky opuchnuté, aby vytvorili guľový tvar. Použitím membránovej teórie vypočítali [25] relatívnu deformáciu pre RBC. To využili pre overenie nastavení lasera. Okrem RBC boli natiahnuté aj fibroblasty a bunkové línie rakoviny prsníka. Tieto údaje nám môžu pomôcť s kalibráciou bunky s jadra.

Počas experimentu bola RBC zachytená s laserovým svetelným výkonom 5 mW v každom lúči. Následne boli bunky naťahované so zvyšujúcimi sa hodnotami svetelného výkonu, postupne 15 mW, 30 mW, 50 mW, 80 mW a 110 mW.

5.3.1 Simulácie optického naťahovania

Aby sme simulovali vplyv lasera na povrch bunky, musíme vypočítať silu naťahovania aplikovanú na každý z bodov na bunkovej membráne. Na základe teórie vysvetlenej v článku [25] vieme, že je možné vypočítať profil mechanického napätia pôsobiaceho na bunku σ_r (pri daných parametroch vzdialenosti medzi laserovými lúčmi a polomerom laserových lúčov) pomocou rovnice:

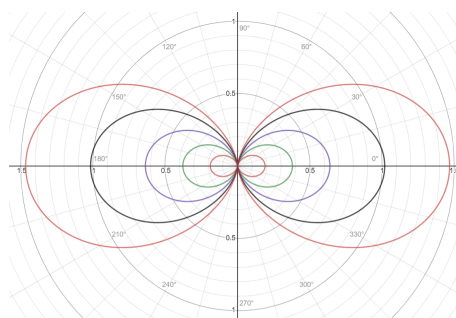
$$\sigma_r(\theta) = \sigma_0 \cos^2(\theta), \quad (5.3)$$

kde σ_0 je maximálne mechanické napätie pozdĺž osi lasera a θ je uhol medzi povrchom bunky a normálov sily pôsobiaceho lasera.

V tabuľke 5.9 uvádzame hodnoty maximálneho mechanického napätia pre naťahovanie RBC pri daných hodnotách svetelného výkonu a daných experimentálnych podmienkach, ktoré vypočítali v [25].

Tabuľka 5.9: Maximálne hodnoty mechanického napätia z [25] použité v simuláciách.

Svetelný výkon [mW]	Maximálne mechanické napätie [N/m ²]
15	0.19
30	0.38
50	0.64
80	1.02
110	1.47



Obrázok 5.9: Profil mechanického napätia vypočítaný pre rôzne hodnoty maximálneho mechanického napätia.

Pretože maximálne mechanické napätie zohľadňuje všetky parametre biologického experimentu, je to jediný parameter, ktorý musíme nastaviť v našich simuláciách.

5.3. LASEROVÉ NAŤAHOVANIE BUNIEK

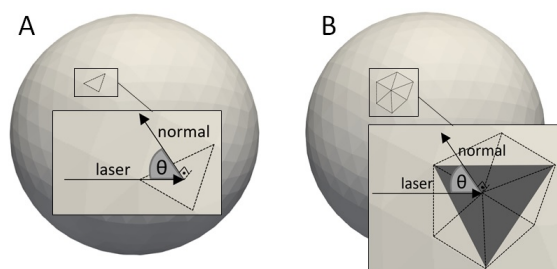
Aby sme vypočítali stresový profil bunky, musíme vypočítať uhol θ pre každý z bodov membrány. Na obrázku 5.9 vidíme hodnoty profilu napätia v závislosti od uhla a hodnoty maximálneho napätia. Najväčší profil mechanického napätia sa vyskytuje pri najsilnejšom svetelnom výkone pozdĺž osi lúča.

5.3.2 Výpočet ňaťahovacej sily

Porovnávame dva možné prístupy k výpočtu ňaťahovacej sily. Spoločné pre oba prístupy je skutočnosť, že na natiahnutie bunky potrebujeme vyvinúť sily na každý bod na membráne bunky. Tieto sily sa musia prepočítať opakovane. Pretože sa bunka deformuje, mení sa jej tvar a mení sa aj uhol medzi normálnym vektorom povrchu bunky a vektorom laserového lúča.

Pre účely simulácie sme rozdelili body na bunke do dvoch skupín - ľavé a pravé, v závislosti od ich polohy od stredu bunky pozdĺž x -ovej osi. Toto oddelenie je spôsobené skutočnosťou, že orientácia laseru je pre tieto dve skupiny odlišná. Na ľavé body pôsobí laser s orientáciou smerujúcou v smere $(1, 0, 0)$ a na pravé s laserovou orientáciou $(-1, 0, 0)$.

Naše dva prístupy sa rozlišujú v spôsobe výpočtu uhlu medzi laserom a povrchom membrány bunky. Porovnanie týchto dvoch prístupov môžeme vidieť na obrázku 5.10.



Obrázok 5.10: Porovnanie rôznych prístupov k výpočtu uhlu medzi normálou povrchu bunky a vektorom pôsobenia laserového lúča.

Prístup A

Pri použití prístupu A sa najskôr vezme každý trojuholník membrány, vypočíta normálový vektor pre tento trojuholník a potom vypočíta uhol medzi normálou a vektorom laserového lúča. Týmto spôsobom získame mechanické napätie pôsobiace na celý trojuholník. Následne je každému bodu tohto trojuholníka pripísaná rovnaká časť tejto sily - jedna tretina mechanického napätia aplikovaného na daný

trojuholník. Pretože každý bod membrány je súčasťou niekoľkých trojuholníkov, celková sila pôsobiaca na akýkoľvek daný bod je súčtom všetkých čiastkových síl z každého trojuholníka.

Prístup B

Pri použití prístupu B sa vypočíta normálny vektor pre každý bod membrány. Na získanie normálneho vektora v danom bode však musíme najprv získať rovinu. Pri tomto prístupe teda hľadáme dotykovú rovinu k povrchu v danom bode membrány. Na definovanie roviny potrebujeme tri body. A keďže máme trojuholníkovú sieť, hľadáme medzi susednými bodmi daného bodu tri umiestnené tak, aby sa normála zarovnala s daným bodom. Potom sa ťažhová sila aplikovaná na tento bod vypočíta podľa vzorca pre profil mechanického napätia. Nie sú potrebné ďalšie kroky. Tento prístup je možno menej intuitívny, implementácia v Pythone je však jednoduchšia ako pri prístupe A. Jednoduchosť použitia tohto prístupu spočíva v dostupnosti funkciu na nájdenie troch najlepších susedov daného bodu, ktorá je zahrnutá v PyOIF.

Porovnaním týchto dvoch prístupov sme dosiahli porovnateľné výsledky. Hoci počas procesu ťažhovania sme mohli pozorovať určité rozdiely vo vývoji tvaru buniek, bunky sa stabilizovali v porovnateľných tvaroch s rozdielmi menšími ako $0.01\mu\text{m}$. A keďže použitie prístupu A má prirodzenejšie prepojenie s fyzikálnymi procesmi prebiehajúcimi počas laserového ťažhovania, pretože laser ovplyvňuje povrch a nie diskkrétne body, rozhodli sme sa použiť tento prístup v simuláciách.

Implementácia softvéru

Všetky simulácie sú spúšťané v ESPResSo. Náš modul PyOIF je pythonovský modul pomocou ktorého je možné simulovať elastické objekty v toku. Výpočty všetkých interakcií medzi tekutinou a elastickými objektami a medzi jednotlivými bodmi na membráne objektov sú prepočítavané v každom časovom kroku na C++ jadre. Komunikácia medzi Pythonom a C++ prebieha vďaka Cython. Všetky parametre simulácie sú uvedené v tabuľke 5.10.

Implementácia nového typu interakcie v jadre C++ je pomerne časovo náročná. Preto sme sa najprv rozhodli otestovať oba naše prístupy v Pythone. Vytvorili sme skript, kde sa sily prepočítavajú v každom simulačnom kroku. Jeden výpočtový krok v Pythone budeme nazývať *simulačný krok* a krok v jadre C++ *iteračný krok*. Výpočty interakcií medzi tekutinou a objektami a jednotlivými bodmi na povrchu objektov bežia v každom iteračnom kroku. Zmeny však môžu byť dosť malé a ak chceme analyzovať našu simuláciu nemusíme nutne pozeráť na výstup z každého

Tabuľka 5.10: Parametre simulácie

časový krok	10^{-7} s
lbgrid	10^{-6} m
k_s	$1 \cdot 10^{-5} N/m$
k_b	$5.4 \cdot 10^{-12} N$
k_{al}	$2 \cdot 10^{-5} N/m$
k_{ag}	$7 \cdot 10^{-4} N/m$
k_v	$9 \cdot 10^2 N/m^2$
polomer RBC	$3,3 \cdot 10^{-6} m$
koeficient trenia	$3,39 \cdot 10^{-9} N \cdot s \cdot m^{-1}$
hustota tekutiny	$10^3 kg \cdot m^{-3}$
viskozita tekutiny	$1.5 \cdot 10^{-6} m^2/s$

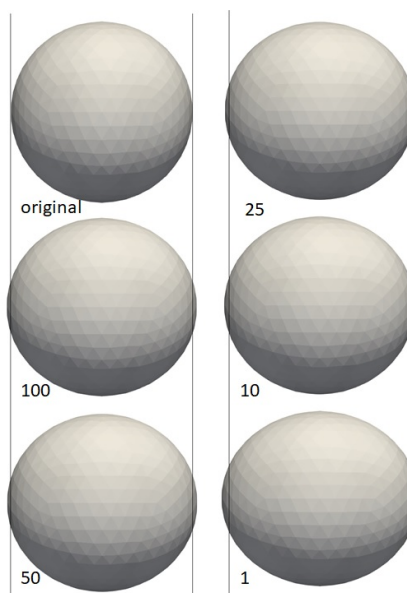
iteračného kroku. Jeden simulačný krok sme zvyčajne v našich simuláciách nastavili na 100 iteračných krokov.

Ak však prepočítavame naťahovacie sily pôsobiace na bunku v optickej pasci, prirodzeným prístupom je ich prepočítavať v každom iteračnom kroku. Pretože v skutočnom biologickom experimente sa bunka neustále mení vďaka pôsobeniu svetelnej sily, a tak sa aj neustále mení vplyv laserového lúča na jej povrch. V tabuľke 5.11 prezentujeme rozdiel vo výpočtovom čase simulácií. Čas bol nameraný v momente keď bol tvar bunky stabilizovaný. Hlavnou príčinou zvyšujúceho času bola komunikácia medzi Pythonom a C++.

Tabuľka 5.11: Porovnanie výpočtových časov pri rôznych počtoch iteračných krokov na jeden simulačný krok.

Počet iteračných krokov	Výpočtový čas
1	15.28 min
10	5.12 min
25	3.5 min
50	3.42 min
100	3.38 min

Rozdiely medzi simuláciami sú nielen vo výpočtovom čase, ale aj v deformácii bunky. Porovnanie týchto rozdielov je uvedené na obrázku 5.11.



Obrázok 5.11: Rozdiely medzi tvarmi RBC po $1\mu s$ simulačného času. Čísla označujú počet iteračných krokov na jeden simulačný krok v simulácii. Horná ľavá bunka označená *original* je bunka v svojom relaxovanom tvare.

Dôvod je celkom jasný. Ak napríklad porovnáme simuláciu s krokom simulácie pozostávajúcim z 1 iteračného kroku so simuláciou so 100 iteračnými krokmi. Porovnávame ich po rovnakom počte iteračných krokov. Rozdiel vyplýva z malej, ale stálej zmeny sily pôsobiacej na body membrány. Za 100 iteračných krokov sa v simulácii s 1 iteračným krokom na 1 simulačný krok prepočíta sila pôsobiaca na bunku 100 krát. Pri simulácii kde máme 100 iteračných krokov na jeden simulačný krok sa táto sila prepočíta len raz a rovnaká sila pôsobí 100 iteračných krokov.

Zaujímavejší ako priebeh deformácie bunky bol pre nás výsledný tvar po stabilizácii bunky. Bunku sme označili za stabilizovanú v prípade, že zmena dĺžky bunky bola menšia ako 0,0001. Ak sme sa zamerali len na výsledné tvary buniek, rozdiely medzi jednotlivými nastaveniami pre počet iteračných krokov na jeden simulačný krok boli zanedbateľne malé. Prístup pre modelovanie laserového natiahovania cez Pythonovský modul je funkčný v prípade záujmu o výsledný tvar buniek. Implementácia interakcie laserového natiahovania v C++ by podľa našich odhadov zlepšila najmä výpočtový čas. Zníženie výpočtového času by bolo spôsobené predovšetkým znížením nutnosti komunikovať medzi C++ a Pythonovským modulom.

Taktiež v prípade, že by nás zaujímala vyššia presnosť celkového priebehu natiahovania membrány bunky v čase, bolo by výhodnejšie mať implementované laserové

naťahovanie v C++.

5.3.3 Výsledky

Predchádzajúce časti popisovali všetky podrobnosti o nastavení simulácií. Spustili sme simuláciu pre každú hodnotu maximálneho mechanického napätia. V každej simulácii sme zaznamenali dĺžku a šírku bunky. Dĺžka sa vypočítala ako vzdialenosť medzi dvoma najextrémnejšími bodmi pozdĺž osi laserového lúča (smer x v našej simulácii), jeden s najmenšou a druhý s najväčšou hodnotou x . Šírka bola vypočítaná podobným spôsobom. Bola to vzdialenosť medzi dvoma najextrémnejšími bodmi pozdĺž osi kolmej na laserový lúč (v osi simulácia z -os). Simulácia sa skončila, keď sa dĺžka bunky nezmenila o viac ako 0,0001. Pre každú simuláciu bola zaznamenaná konečná dĺžka a šírka. Výsledky simulácií sme porovnali s teoreticky odvodenou dĺžkou a šírkou gule s danými rozmermi.

Na ich výpočet používame výrazy popisujúce radiálne (u_r) a meridiálne (u_θ) deformácie membrány. Tieto boli odvodené na základe teórie membrán v [25].

$$u_r(\theta) = \frac{\rho^2 \sigma_0}{4Eh} [(5 + \nu) \cos^2(\theta) - 1 - \nu] \quad (5.4)$$

$$u_\theta(\theta) = \frac{\rho^2 \sigma_0 (1 + \nu)}{2Eh} \cos(\theta) \sin(\theta) \quad (5.5)$$

Dĺžka a hrúbka sa potom vypočítali odpočítaním týchto deformácií od pôvodného radiálneho a meridiálneho tvaru bunky, ktorý bol v oboch prípadoch kruh s polomerom 3,3. Pokiaľ ide o parametre, σ_0 je maximálne mechanické napätie, θ je v tomto prípade polárny uhol, r je radiálny smer, ν je Poissonove číslo s hodnotou $\nu = 0.5$ (normálna hodnota pre biologické bunky) a $Eh = (3.9 \pm 1.4) \times 10^{-5} \text{ Nm}^{-1}$.

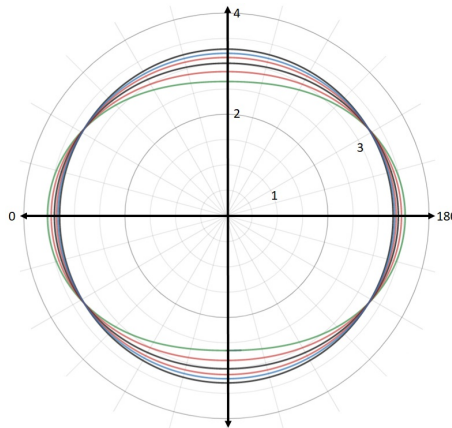
Výsledné deformované tvary pre radiálnu deformáciu sú zobrazené v obrázku 5.12. Porovnanie výsledkov medzi simulačnými a teoretickými dátami pre dĺžky a šírky bunky je uvedené v tabuľke 5.12.

Okrem porovnania dĺžky a hrúbky sme porovnali aj celkový výsledný tvar deformácie simulačného radiálneho tvaru s teoreticky odvodeným tvarom. Keďže sú však rozdiely pomerne malé, ako vidíme z tabuľky 5.12, ide skôr o kvalitatívne porovnanie. Prekrytie simulačných a teoretických tvarov nie je dokonalé, ale malé rozdiely môžu byť spôsobené malými číselnými odchýlkami, ktoré sa vyskytujú pri každej interakcii medzi Pythonom a C ++. Tieto výsledky ukazujú, že prístup k

Tabuľka 5.12: Porovnanie experimentálnych a simulačných dát.

σ_0	dĺžka _t	dĺžka _s	šírka _t	šírka _s
0	6.6	6.6	6.6	6.6
0.19	6.66	6.69	6.43	6.51
0.38	6.72	6.77	6.27	6.36
0.64	6.81	6.88	6.04	6.17
1.02	6.93	7.02	5.71	5.89
1.47	7.08	7.17	5.31	5.59

modelovaniu optického natáhovania buniek je funkčný a existuje možnosť zlepšenia.



Obrázok 5.12: Analyticky vypočítané radiálne tvary bunky pre každú z hodnôt maximálneho mechanického napätia.

V tejto časti sme poskytli stručný prehľad o mechanike techniky optických pascí použitej pre natáhovanie buniek v biologickom experimente. Vysvetlili sme náš prístup k simulácii natáhovania buniek, najmä sme sa zamerali na výpočet natáhovacích síl aplikovaných na jednotlivé body triangulácie povrchu bunky. Ukázali sme dva možné výpočty týchto síl a vysvetlili sme ich výhody a nevýhody. Zvolili sme prístup výpočtu uhlu medzi laserom a bunkou, ktorý využíva normálový vektor trojuholníka siete a nie normálový vektor pre konkrétny bod na sieti bunky. Tento prístup bol zvolený vďaka prirodzenejšiemu vysvetleniu fyzikálneho javu prebiehajúceho počas natáhovania. Implementácia tohto mechanizmu v Pythone nám umožnila overiť fungovanie tejto metódy na nakalibrovanej RBC. Údaje o dĺžke bunky sa lepšie zhodujú s teoretickými hodnotami v porovnaní s údajmi o šírke bunky. Malý rozdiel medzi simulačnými a experimentálnymi dátami možno vysvetliť numerickými chybami akumulujúcimi sa početnými prechodmi medzi Pythonom a

C ++.

5.4 Kalibrácia na základe biologických experimentov

Ako sme vyššie spomenuli, vykonávanie experimentov s mikrofluidickými zariadeniami je náročná práca, hoci aj pri použití čo najjednoduchších geometrií kanálikov. Náročnosť sa potom len znásobuje v prípade použitia napríklad zariadení určených na triedenie buniek, ktoré sme spomínali v časti 2.3. Preto je našim cieľom poskytnúť výpočtový model, s ktorým je možné najprv otestovať rôzne typy geometrií a biologický experiment vykonať už len pre vybrané rozmery geometrie.

Náš model má množstvo parametrov, a vďaka tomu je ho možné prispôbiť pre rôzne druhy buniek. Buniek s jadrom, ktoré sa používajú v biologických experimentoch je veľké množstvo a informácie o meraní mechanických vlastností jednotlivých buniek nie sú dostupné pre všetky.

Preto nami navrhovaný kalibračný proces zahŕňa aj návrh mikrofluidického zariadenia, pomocou ktorého je možné tieto mechanické vlastnosti namerať. Odporúčame využiť mikrofluidický kanálik so zúžením, ktorý sme opísali v časti 5.1.1. Rozmery je potrebné upraviť podľa rozmerov veľkosti bunky. Ideálne je zvoliť zúženie s prierezom, ktorý bude mať šírku jemne menšiu ako je priemer bunky. Ak by bolo zúženie príliš malé, mohla by sa v ňom bunka zaseknúť. Ako sme už spomínali v predošlej časti, výšku kanáliku je vhodné zvoliť najnižšiu ako sa podarí vyrobiť pre zjednodušenie zaostrenia mikroskopu. Je vhodné zvoliť výšku aspoň z časti väčšiu ako je priemer bunky. To umožní inicializovať bunku v simulácii v nezdeformovanom stave. Pokiaľ by výška kanálika bola menšia ako priemer bunky, bolo by nutné najprv zbehnúť prípravnú simuláciu. V nej by bola bunka pustená kanálikom, ktorý sa z väčších rozmerov zúži na požadované rozmery daného mikrofluidického zariadenia. Následne by bol uložený takto zdeformovaný tvar bunky a ten by bolo možné použiť do simulácií biologického experimentu. Tento prípravný krok zbytočne navyšuje prácu a pokiaľ je možné ovplyvniť rozmery kanálika, je vhodné myslieť aj na simuláciu.

Pri zaznamenávaní experimentu odporúčame získať dáta aspoň o 100 bunkách, ktoré prešli celým kanálom. Následne je potrebné získať informáciu o priemernom rýchlostnom profile buniek, čase prechodu kanálikom a deformačnom indexe. Ako

sme videli pri dátach z Daovho ňaťahovacieho experimentu [13] použitých pri kalibrácii RBC, je dobré vyhodnotiť aj odchýlky v biologických dátach.

Následne pri nastavení toku v simulácií, odporúčame vzhľadom na možné straty pri prechode tekutiny zo striekačky do zariadenia, využiť informáciu o rýchlostnom profile buniek z biologického experimentu.

Pre modelovanie bunky je možné využiť jeden z prístupov spomínaných v kapitole 4. Kalibrácia modelu spočíva s nastavením jednotlivých elastických parametrov. Odporúčame nasledujúci postup, nech už je použitý ktorýkoľvek z prístupov k modelovaniu bunky s jadrom.

Určenie hraníc.

V prípade použitia buniek s priemerom okolo $15 \mu m$ je možné použiť ako začiatočný interval kalibrácie hodnoty parametrov uvedených v kapitole 4. V prípade použitia iných rozmerov odporúčame najprv vytvoriť kratšie kanáliky, aké popisujeme v kapitole 4. Následne v nich otestovať parametre, ktoré fungovali pre bunku s priemerom $15 \mu m$. Pri použití rovnakej diskretizácie by mali dobre fungovať parametre súvisiace s tvorbou väzieb. Pri použití inej diskretizácie bude potrebné zmeniť aj tieto parametre. Dobrým spôsobom úpravy je pozrieť sa na priebeh funkcie potenciálu príslušnej väzby. Ak napríklad vzdialenosť bodov membrány a jadra je $3 \mu m$ (o $\frac{1}{3}$ menšia ako priemerná relaxovaná dĺžka väzby) a potenciál má vtedy hodnotu 1, tak fungujúca sada parametrov pri väčšej bunke by dávala potenciál s hodnotou 1 keď sa vzdialenosť bodov membrány a jadra zníži o tretinu. Takýmto spôsobom vieme získať hrubý odhad pre fungujúce sady parametrov pomerne rýchlo, keďže kanáliky z kapitoly 4 sú kratšie ako kanálik použitý pri biologickom experimente. Výroba takýchto kanálikov, hlavne rozdielna výška kanálika, je však náročnejšia a preto pre biologický experiment odporúčame kanálik s jednotnou výškou.

Hrubá kalibrácia.

Pre každý parameter sme teda získali interval parametrov, ktorý fungoval pre kratšie kanáliky. Tieto intervaly si rozdelíme na niekoľko častí (odporúčame 4) a spustíme simulácie už v geometrii navrhutej podľa biologického experimentu. Pre každú sadu získame kvantitatívne hodnoty. Ako napríklad DI a vstupný čas, za ktorý sa bunka dostane do zúženia celá (táto informácia je zaujímavá v prípade, že zúženie je menšie ako priemer bunky a bunka sa v biologickom experimente na chvíľu zachytí pri vstupe doň). Pre každú sadu si vypočítame percentuálnu odchýlku od hodnôt

5.4. KALIBRÁCIA NA ZÁKLADE BIOLOGICKÝCH EXPERIMENTOV

z biologického experimentu. Je vhodné rozdeliť kanálik na niekoľko úsekov a pre každý úsek vyhodnotiť priemernú hodnotu DI a tú potom porovnávať s priemernou hodnotou DI zo simulácie. Následne vyberieme tie intervaly hodnôt parametrov, ktoré majú najmenšie odchýlky. Jeden zo spôsobov počítania odchýlok je popísaný v časti 3.3. Ďalší zo spôsobov popisujeme neskôr v tejto časti.

Vymedzenie intervalov fungujúcich sád parametrov.

Pre získanie finálnej sady parametrov je vhodné pozrieť sa na citlivosť jednotlivých parametrov. Pre porovnanie citlivosti parametrov si vždy zafixujeme všetky okrem jedného. Pokiaľ sa odchýlka od biologických dát príliš nemení, môžeme hodnotu daného parametra zafixovať. Pre zvyšné parametre potom rozdelíme intervaly z predošlého kroku na jemnejšie časti. Opäť spustíme všetky kombinácie parametrov a vypočítame odchýlky od biologických hodnôt. Následne tak získame hodnoty elastických parametrov, ktoré môžeme používať v ďalších simuláciách danej bunky aj v iných mikrofluidických zariadeniach.

Spôsob porovnania experimentálnych a simulačných dát

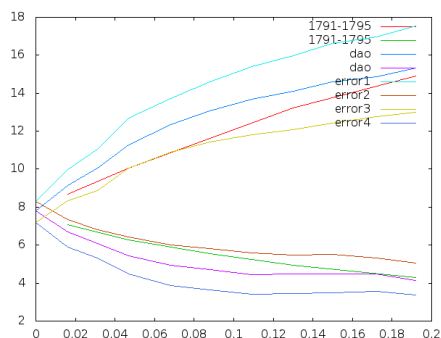
Pri kalibrácii RBC sme mali viacero spôsobov vyhodnocovania vhodnosti elastických parametrov. V tejto časti popisujeme jeden z nich, ktorý sme nepoužili pre záverečné vyhodnotenie, ale môže byť užitočný pri vyhodnocovaní zhody medzi DI nameraným z biologických dát a zo simulačných dát v zúženom mikrofluidickom kanáliku. Keďže DI je počítaný ako pomer medzi stranami obdĺžnika opísaného bunke, môžeme miesto samotného DI porovnávať tieto dve dĺžky samostatne, podobne ako pri RBC.

Na prvý pohľad sa zdá, že je pomerne ľahké určiť, ktorá sada parametrov je dobrá, a ktorá nefungujúca. Avšak dve rôzne sady parametrov môžu vyústiť aj do takto podobných grafov 5.15 a 5.16. Tu už určovanie vhodnosti na základe grafu nie je možné.

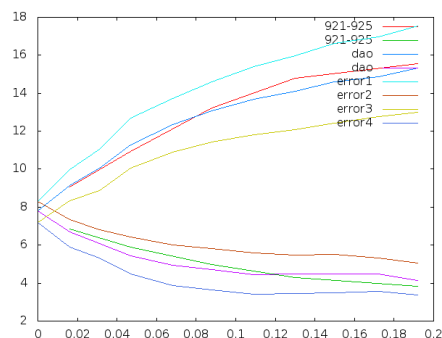
Jeden zo spôsobov vyhodnocovania tejto zhodnosti využíva štvorcovú chybu definovanú ako

$$SE = \sum_i ((d_i^a - e_i^a)^2 + (d_i^b - e_i^b)^2), \quad (5.6)$$

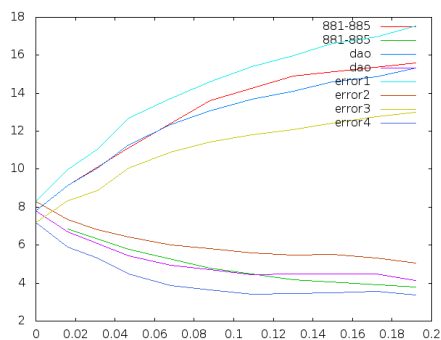
kde (d_i^a) je dĺžka pozdĺžneho predĺženia v smere natiahovania a (d_i^b) je dĺžka priečného zúženia kolmo na smer natiahovania, e_i^a a e_i^b sú príslušné hodnoty z biologického experimentu.



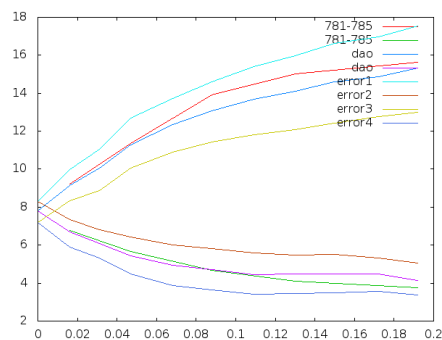
Obrázok 5.13: Porovnanie biologických a simulačných hodnôt naťahovacieho experimentu. V tomto prípade sa dáta nezhodujú.



Obrázok 5.14: Porovnanie biologických a simulačných hodnôt naťahovacieho experimentu. V tomto prípade sú dáta podobné.



Obrázok 5.15: Porovnanie biologických a simulačných hodnôt naťahovacieho experimentu.



Obrázok 5.16: Porovnanie biologických a simulačných hodnôt naťahovacieho experimentu.

Rozhodli sme sa využiť informáciu o rozptyle hodnôt a v našom postupe sme si vďaka nej určili "pásky" rôznej zhody. Jednotlivé hodnoty rozptylu budeme značiť r_i^{aH} pre hornú a r_i^{aD} pre dolnú hranicu dĺžky (d_i^a a rovnako potom r_i^{bH} a r_i^{bD} sú hranice pre dĺžku (d_i^b).

Ak sa všetkých 10 hodnôt nachádzalo v oblasti do 25% vzdialenosti od biologických hodnôt boli príslušné sady parametrov vyhlásené za dobré. Do 60% boli fungujúce a zvyšné boli nefungujúce.

Dobré sady teda spĺňali nasledujúce nerovnosti:

$$|d_i^a - e_i^a| \leq \frac{r_i^{aH} - r_i^{aD}}{4} \quad (5.7)$$

$$|d_i^b - e_i^b| \leq \frac{r_i^{bH} - r_i^{bD}}{4} \quad (5.8)$$

Tento postup je možné ďalej upraviť použitím zvlášť kritérií pre hornú a dolnú oblasť okolo biologických hodnôt. Napríklad je možné zvoliť nasledujúce dve nerovnosti ako kritérium pre dobré sady:

$$|d_i^a - e_i^a| \leq \min\left\{\frac{r_i^{aH} - e_i^a}{2}, \frac{e_i^a - r_i^{aD}}{2}\right\} \quad (5.9)$$

$$|d_i^b - e_i^b| \leq \min\left\{\frac{r_i^{bH} - e_i^b}{2}, \frac{e_i^b - r_i^{bD}}{2}\right\} \quad (5.10)$$

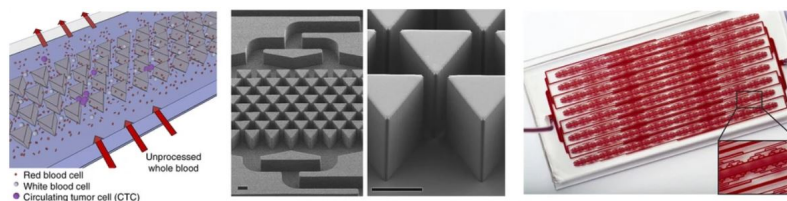
5.5 Možnosti využitia modelu pre vývoj mikrofluidických zariadení

S nakalibrovaným modelom je následne možné uskutočniť ďalšie simulácie, ktoré môžu slúžiť napríklad k optimalizácii rozmerov mikrofluidického zariadenia. Napríklad pri zariadení zobrazenom na obrázku 2.4 je možné skúmať uhol vstupných kanálikov, ich šírku ale aj rôznu rýchlosť toku. Následne, na základe výsledkov z takto rôzne nastavených simulácií, je možné vybrať najvhodnejšiu geometriu zariadenia.

Zaujímavým využitím je aj skúmanie správania CTC buniek v krvnom toku v cievach s rôznymi veľkosťami. Simulácie môžu umožniť skúmanie závislosti správania CTC buniek od rýchlosti toku a hematokritu. Podobne ako v štúdií [77].

Okrem individuálnych buniek sa však v toku môžu nachádzať aj zoskupenia buniek. Ako sme spomínali v časti 1.1, prítomnosť CTC buniek v toku krvi je indikátorom začínajúcej metastázovanej rakoviny. Prítomnosť klastrov týchto buniek poukazuje na agresívnejší typ takejto rakoviny [19]. Detekcia takýchto klastrov vo vzorke krvi je preto užitočným nástrojom pre klinickú prax. Jedným z mikrofluidických zariadení, ktoré boli vytvorené pre tento účel popisujú v článku [67]. Zariadenie je zobrazené na obrázku 5.17. Model bunky s jadrom by mohol byť využitý pre optimalizáciu rozložení prekážok v tomto mikrofluidickom kanáliku.

Model klastra by mohol byť vytvorený pomocou spojenia viacerých buniek s jadrom. Spojenia by mali podobný charakter ako väzby, ktoré sme popísali v časti 4.2. Bude potrebné otestovať viacero funkcií, ktoré by mohli simulovať dynamiku

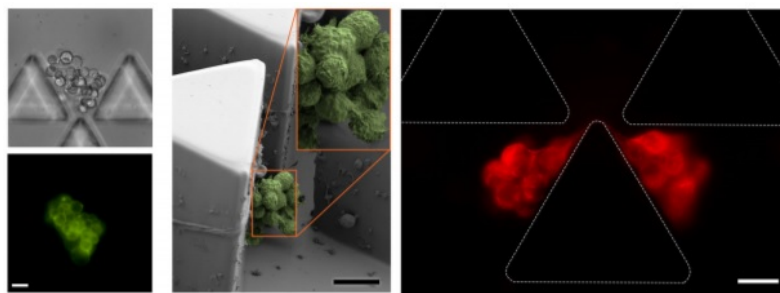


Obrázok 5.17: Mikrofluidické zariadenie určené na zachytávanie klastrov buniek. Prevzaté z [67].

týchto spojov. Taktiež koľko takýchto spojov bude potrebné vytvoriť. Mohol by sa aplikovať podobný postup ako v časti 4.2.

Pre začiatok by v simuláciách mohlo byť skúmané správanie klastra tvoreného dvoma bunkami v časti mikrofluidického zariadenia. Táto simulácia by mohla obsahovať prekážky rozložené podobne ako vidíme na obrázku 5.18. Pomocou sérií simulácií by mohli byť otestované faktory vplyvajúce na správanie pohybu klastra.

- Skúmanie vplyvu rýchlosti toku.
- Skúmanie vplyvu pozície klastra.
- Skúmanie vplyvu rozloženia buniek v rámci klastra.



Obrázok 5.18: Detail časti mikrofluidického zariadenia zachytávajúceho klastre buniek. Prevzaté z [67].

Taktiež by bolo možné sledovať mieru zachytenia buniek pri rôznych rozostaveniach troch trojuholníkových prekážok.

Neustále napredovanie v oblasti mikrofluidických zariadení, snaha zlepšovať zdravotnú starostlivosť a minimalizovať invazívnosť nutných diagnostických vyšetrení vedie k veľkej využiteľnosti nášho modelu.

6 Záver

V našej práci sme sa zamerali na rozšírenie výpočtového nástroja vyvíjaného výskumnou skupinou Cell in Fluid. Rozšírenie tohto nástroja o model bunky s jadrom umožňuje modelovanie širokého spektra buniek, čím poskytuje ďalšie možnosti využitia výpočtového modelu, ktoré popisujeme v časti 5.5.

Pre vytvorenie modelu bunky s jadrom sme najprv preskúmali dostupné informácie o biologických a mechanických vlastnostiach buniek. Tieto poznatky sme zhrnuli v časti 2.1. Dôležitým bolo aj získanie prehľadu o už existujúcich spôsoboch modelovania takýchto buniek. V časti 3.1 sme sa pozreli na rôznorodé prístupy. Zamerali sme sa na diskkrétne metódy, ktorých pohľad na deformáciu je makroskopický. Popísali sme modely, ktoré boli detailnejšie v popise štruktúry bunky a teda aj výpočtovo náročnejšie. Zahrnuli sme aj modely, ktoré naopak štruktúru bunky s jadrom zjednodušovali a získali tak výpočtovú výhodu. V prístupe, ktorý sme následne vybrali a rozpracovali v tejto práci sme zvolili strednú cestu.

V kapitole 4 detailne popisujeme všetky navrhované metodiky. Pre otestovanie fungovania jednotlivých prístupov sme vytvorili geometriu mikrofluidického kanálíka, na základe poznatkov z existujúcich biologických experimentov. Prehľad rôznych typov mikrofluidických zariadení uvádzame v časti 2.3.

Pri vytváraní geometrie kanálíka sme sa zamerali na vytvorenie stresového prostredia pre bunku, v ktorom bola jej deformácia podmienkou prechodu. Prípadné nedostatky jednotlivých prístupov tak mohli byť odhalené. Pythonovské skripty ktoré sme využívali pri týchto simulačných experimentoch sú priložené v prílohe B. Tieto skripty majú veľké množstvo parametrov, a tak môžu byť ľahko upravené pre potreby testovania konkrétnej biologickej bunky v schematicky podobnom mikrofluidickom zariadení. Dôvodom pre takúto parametrizáciu bola možnosť využiť tieto skripty pre kalibráciu rôznych druhov buniek.

Poznatky z kalibrácie RBC bunky, ktorú popisujeme v časti 3.3, sme využili pre vytvorenie kalibračného postupu pre model bunky s jadrom. Dôležitou súčasťou

kalibračného procesu je získanie dát z vhodného biologického experimentu. Veľká časť kapitoly 5 je venovaná práve tejto problematike. Poznatky, ktoré sme získali pri realizácii biologických experimentov popísaných v časti 5.1, sme využili pri popise kalibračného procesu v časti 5.4.

Počas doktorandského štúdia sa nám podarilo nadviazať spoluprácu s laboratóriom, kde prebehnú experimenty s bunkami nakazenými maláriou. V súčasnosti čakáme na zrealizovanie týchto experimentov. Zatiaľ sme pripravili prvotné simulácie. Ďalším krokom bude nakalibrovať RBC, ktorá je infikovaná parazitom malárie, pomocou predpripravenej simulácie. Podľa dostupnosti iných druhov buniek, bude možné uskutočniť ďalšie experimenty a následnú kalibráciu elastických parametrov modelu bunky s jadrom.

Počas výskumného pobytu v spomínanom laboratóriu sme vykonali aj experimenty s ďalšími typmi mikrofluidických zariadení. Jeden z čipov obsahoval špirálový triediaci kanálik. Po získaní maláriových nakazených RBC bude možné vytvoriť zmiešanú suspenziu týchto buniek so zdravými RBC. Pomocou simulácií môže byť následne upravená geometria špirál v týchto čipoch pre lepšie výsledky triedenia, ktoré budú špecificky nastavené na odlíšenie zdravých a chorých RBC.

Publikácie

- [1] (AFD) Bohiniková, A.: 'Methods for measuring biophysical properties of circulating tumor cells', MIST 2018 : Mathematics in Science and Technologies, Rajecká Lesná, Slovakia, January 3-7, 2018 pp. 7-11, CreateSpace Independent Publishing Platform, 2018, ISBN 978-1987603392

- [2] (AFC) Kovalčíková, K., Bohiniková, A., Slavík, M., Mazza Guimaraes, I., Cimrák, I.: 'Red Blood Cell Model Validation in Dynamic Regime', Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Volume 10813 LNBI, 2018), IWBBIO 2018: 6th International Work-Conference on Bioinformatics and Biomedical Engineering, Granada, Spain, April 25-27, 2018, pp. 259-269, 2018, ISBN 978-331978722-0

- [3] (AFC) Kovalčíková, K., Slavík, M., Bachratá, K., Bachratý, H., Bohiniková, A.: 'Volumetric flow rate in simulations of microfluidic devices', EPJ Web of Conferences 2018 (Volume 180, 4 June 2018, Article number 02046), EFM 2017: 12th International Conference on Experimental Fluid Mechanics, Mikulov, Czech Republic, November 21-24, 2017, ISSN 21016275

- [4] (AFD) Bohiniková, A.: 'Elastic spheres in narrow channels discretization and self-cell interaction', MIST 2019 : Mathematics in Science and Technologies, Rajecká Lesná, Slovakia, January 14-18, 2019 pp. 11-16, Independently published, 2019, ISBN 978-1794002180

- [5] (AFC) Bohiniková, A., Bachratá, K.: 'Simulation Approaches to Stretching of Red Blood Cells', Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) IWBBIO 2019: 7th International Work-Conference on Bioinformatics and Biomed-

- ical Engineering, Granada, Spain, May 6-8, 2019, pp. 309-317, 2019, ISBN 978-3-030-17935-9
- [6] (AFC) Bohiniková, A.: 'Computational Modelling of Elastic Cells with Nucleus', CMBE 2019 : 6th International Conference on Computational & Mathematical Biomedical Engineering, Sendai, Japan, June 10-12, 2019 pp. 622-625, ISBN 978-0-9562914-5-5
- [7] (AFC) Bohiniková, A., Bachratá, K.: 'Calibration of Circulating Tumor Cell's Model in Narrow Flow', EPJ Web of Conferences 2019 (Volume 213, 28 June 2019, Article number 02006), EFM 2018: 13th International Conference on Experimental Fluid Mechanics, Prague, Czech Republic, November 13-16, 2018, ISSN 2267-1242
- [8] (AFC) Bohiniková, A., Bachratá, K.: 'Computational Model of Optical Stretcher', IDT 2019: International Conference on Information and Digital Technologies, Zilina, Slovakia, June 25-27, 2019, IEEE, pp. 38-42, 2019, ISBN 978-1-7281-1401-9
- [9] (AFC) Bohiniková, A., Maia, I., Smiešková, M., Bugáňová, A., Moita, A. S., Cimrák, I., Lima, R. A.: 'Assessment of Computational Cell Model Benefits for Optimization of Microfluidic Devices', In: Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies - (Volume 1). - 2020. - ISBN 978-989-758-398-8. - S. 280-287.
- [10] (ADC) Jančígová, I., Kovalčíková, K., Bohiniková, A., Cimrák, I.: 'Spring-network model of red blood cell: From membrane mechanics to validation', In: International Journal for Numerical Methods in Fluids. - (2020), s. 1-26.
- [11] (AFH) Bohiniková, A.: 'Understanding data from biological experiments', In: 3rd Workshop on Modelling of Biological Cells, Fluid Flow and Microfluidics 2020, 09.02.2020-13.02.2020, Vrátna, Slovensko, Kindle Direct Publishing, 2020. - ISBN 9781653079599, s. 7-8

Literatúra

- [1] Aidun, C.K., Clausen, J.R.: ‘Lattice-boltzmann method for complex flows’, *Annual review of fluid mechanics*, vol. 42, pp. 439–472, 2010
- [2] Aingaran, M., Zhang, R., et al.: ‘Host cell deformability is linked to transmission in the human malaria parasite plasmodium falciparum’, *Cellular microbiology*, vol. 14, no. 7, pp. 983–993, 2012
- [3] Alix-Panabières, C., Pantel, K.: ‘Challenges in circulating tumour cell research’, *Nature Reviews Cancer*, vol. 14, no. 9, p. 623, 2014
- [4] Bidone, T.C., Kim, T., et al.: ‘Multiscale impact of nucleotides and cations on the conformational equilibrium, elasticity and rheology of actin filaments and crosslinked networks’, *Biomechanics and modeling in mechanobiology*, vol. 14, no. 5, pp. 1143–1155, 2015
- [5] Boas, L., Faustino, V., et al.: ‘Assessment of the deformability and velocity of healthy and artificially impaired red blood cells in narrow polydimethylsiloxane (pdms) microchannels’, *Micromachines*, vol. 9, no. 8, p. 384, 2018
- [6] Brangwynne, C.P., MacKintosh, F.C., et al.: ‘Microtubules can bear enhanced compressive loads in living cells because of lateral reinforcement’, *J Cell Biol*, vol. 173, no. 5, pp. 733–741, 2006
- [7] Caille, N., Thoumine, O., et al.: ‘Contribution of the nucleus to the mechanical properties of endothelial cells’, *Journal of Biomechanics*, vol. 35, pp. 177–187, 2002
- [8] Cimrák, I., Gusenbauer, M., Schrefl, T.: ‘Modelling and simulation of processes in microfluidic devices for biomedical applications’, *Computers & Mathematics with Applications*, vol. 64, no. 3, pp. 278–288, 2012

- [9] Cimrak, I., Jancigova, I.: *Computational Blood Cell Mechanics: Road Towards Models and Biomedical Applications*, Chapman & Hall/CRC Mathematical and Computational Biology, 2018
- [10] Cross, S.E., Jin, Y.S., et al.: ‘Nanomechanical analysis of cells from cancer patients’, *Nature nanotechnology*, vol. 2, no. 12, p. 780, 2007
- [11] Dahl, K.N., Engler, A.J., et al.: ‘Power-law rheology of isolated nuclei with deformation mapping of nuclear substructures’, *Biophysical journal*, vol. 89, no. 4, pp. 2855–2864, 2005
- [12] Dahl, K.N., Kahn, S.M., et al.: ‘The nuclear envelope lamina network has elasticity and a compressibility limit suggestive of a molecular shock absorber’, *Journal of cell science*, vol. 117, no. 20, pp. 4779–4786, 2004
- [13] Dao, M., Lim, C., Suresh, S.: ‘Mechanics of the human red blood cell deformed by optical tweezers’, *Journal of the Mechanics and Physics of Solids*, vol. 51, no. 11–12, pp. 2259 – 2280, 2003, ISSN 0022-5096, proceedings of a Symposium on Dynamic Failure and Thin Film Mechanics, honoring Professor L.B. Freund
- [14] Davidson, P.M., Sliz, J., et al.: ‘Design of a microfluidic device to quantify dynamic intra-nuclear deformation during cell migration through confining environments’, *Integrative Biology*, vol. 7, no. 12, pp. 1534–1546, 2015
- [15] Denais, C., Lammerding, J.: ‘Nuclear mechanics in cancer’, *Advances in experimental medicine and biology*, vol. 773, p. 435, 2014
- [16] Discher, D.E., Boal, D.H., Boey, S.K.: ‘Simulations of the erythrocyte cytoskeleton at large deformation. ii. micropipette aspiration’, *Biophysical Journal*, vol. 75, no. 3, pp. 1584–1597, 1998
- [17] ESPResSo: ‘Espresso documentation’, [<http://espressomd.org/html/doc/index.html>], 2019
- [18] Evans, E., Fung, Y.C.: ‘Improved measurements of the erythrocyte geometry’, *Microvascular research*, vol. 4, no. 4, pp. 335–347, 1972
- [19] Fabisiewicz, A., Grzybowska, E.: ‘Ctc clusters in cancer progression and metastasis’, *Medical Oncology*, vol. 34, no. 1, p. 12, 2017

-
- [20] Fang, Y., Lai, K.W.: ‘Modeling the mechanics of cells in the cell-spreading process driven by traction forces’, *Physical Review E*, vol. 93, no. 4, p. 042404, 2016
- [21] Fedosov, D.A., Gompper, G.: ‘White blood cell margination in microcirculation’, *Soft matter*, vol. 10, no. 17, pp. 2961–2970, 2014
- [22] Friedl, P., Wolf, K., Lammerding, J.: ‘Nuclear mechanics during cell migration’, *Current opinion in cell biology*, vol. 23, no. 1, pp. 55–64, 2011
- [23] Geuzaine, C., Remacle, J.F.: ‘Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities’, *International journal for numerical methods in engineering*, vol. 79, no. 11, pp. 1309–1331, 2009
- [24] Gosse, C., Croquette, V.: ‘Magnetic tweezers: micromanipulation and force measurement at the molecular level’, *Biophysical journal*, vol. 82, no. 6, pp. 3314–3329, 2002
- [25] Guck, J., Ananthakrishnan, R., et al.: ‘The optical stretcher: a novel laser tool to micromanipulate cells’, *Biophysical journal*, vol. 81, no. 2, pp. 767–784, 2001
- [26] Guck, J., Schinkinger, S., et al.: ‘Optical deformability as an inherent cell marker for testing malignant transformation and metastatic competence’, *Biophysical journal*, vol. 88, no. 5, pp. 3689–3698, 2005
- [27] Guilak, F., Tedrow, J.R., Burgkart, R.: ‘Viscoelastic properties of the cell nucleus’, *Biochemical and biophysical research communications*, vol. 269, no. 3, pp. 781–786, 2000
- [28] Guo, Q., Park, S., Ma, H.: ‘Microfluidic micropipette aspiration for measuring the deformability of single cells’, *Lab on a Chip*, vol. 12, no. 15, pp. 2687–2695, 2012
- [29] Gusenbauer, M., Cimrak, I., et al.: ‘A tunable cancer cell filter using magnetic beads: cellular and fluid dynamic simulations’, *arXiv preprint arXiv:1110.0995*, 2011
- [30] Halldorsson, S., Lucumi, E., et al.: ‘Advantages and challenges of microfluidic cell culture in polydimethylsiloxane devices’, *Biosensors and Bioelectronics*, vol. 63, pp. 218–231, 2015

- [31] Hayashi, K., Iwata, M.: ‘Stiffness of cancer cells measured with an afm indentation method’, *Journal of the mechanical behavior of biomedical materials*, vol. 49, pp. 105–111, 2015
- [32] Hirt, C.W., Nichols, B.D.: ‘Volume of fluid (vof) method for the dynamics of free boundaries’, *Journal of computational physics*, vol. 39, no. 1, pp. 201–225, 1981
- [33] Ho-Le, K.: ‘Finite element mesh generation methods: a review and classification’, *Computer-aided design*, vol. 20, no. 1, pp. 27–38, 1988
- [34] Hochmuth, R.M.: ‘Micropipette aspiration of living cells’, *Journal of biomechanics*, vol. 33, no. 1, pp. 15–22, 2000
- [35] Hosain, M.L., Fdhila, R.B.: ‘Literature review of accelerated cfd simulation methods towards online application’, *Energy Procedia*, vol. 75, pp. 3307–3314, 2015
- [36] Hoshino, K., Huang, Y.Y., et al.: ‘Microchip-based immunomagnetic detection of circulating tumor cells’, *Lab on a Chip*, vol. 11, no. 20, pp. 3449–3457, 2011
- [37] Hou, H.W., Li, Q., et al.: ‘Deformability study of breast cancer cells using microfluidics’, *Biomedical microdevices*, vol. 11, no. 3, pp. 557–564, 2009
- [38] Hu, X., Besette, P.H., et al.: ‘Marker-specific sorting of rare cells using dielectrophoresis’, *Proceedings of the national academy of sciences*, vol. 102, no. 44, pp. 15757–15761, 2005
- [39] Hulett, H.R., Bonner, W.A., et al.: ‘Cell sorting: automated separation of mammalian cells as a function of intracellular fluorescence’, *Science*, vol. 166, no. 3906, pp. 747–749, 1969
- [40] Imai, Y., Kondo, H., et al.: ‘Modeling of hemodynamics arising from malaria infection’, *Journal of biomechanics*, vol. 43, no. 7, pp. 1386–1393, 2010
- [41] Jančígová, I., Kovalčíková, K., et al.: ‘Spring-network model of red blood cell: From membrane mechanics to validation’, *International Journal for Numerical Methods in Fluids*, 2020

-
- [42] King, M.R., Phillips, K.G., et al.: ‘A physical sciences network characterization of circulating tumor cell aggregate transport’, *American Journal of Physiology-Cell Physiology*, vol. 308, no. 10, pp. C792–C802, 2015
- [43] Kovalčíková, K., Bohiniková, A., et al.: ‘Red blood cell model validation in dynamic regime’, *International Conference on Bioinformatics and Biomedical Engineering*, pp. 259–269, Springer, 2018
- [44] Kovalčíková, K., Slavík, M., et al.: ‘Volumetric flow rate in simulations of microfluidic devices+’, *European Physical Journal Web of Conferences*, vol. 180, 2018
- [45] Krüger, J., Singh, K., et al.: ‘Development of a microfluidic device for fluorescence activated cell sorting’, *Journal of micromechanics and microengineering*, vol. 12, no. 4, p. 486, 2002
- [46] Kuznetsova, T.G., Starodubtseva, M.N., et al.: ‘Atomic force microscopy probing of cell elasticity’, *Micron*, vol. 38, no. 8, pp. 824–833, 2007
- [47] Lenormand, G., Hénon, S., et al.: ‘Direct measurement of the area expansion and shear moduli of the human red blood cell membrane skeleton’, *Biophysical Journal*, vol. 81, no. 1, pp. 43–56, 2001
- [48] Lenshof, A., Magnusson, C., Laurell, T.: ‘Acoustofluidics 8: applications of acoustophoresis in continuous flow microsystems’, *Lab on a Chip*, vol. 12, no. 7, pp. 1210–1223, 2012
- [49] Li, H., Chen, J., et al.: ‘The optimization of a microfluidic ctc filtering chip by simulation’, *Micromachines*, vol. 8, no. 3, p. 79, 2017
- [50] Lykov, K., Nematbakhsh, Y., et al.: ‘Probing eukaryotic cell mechanics via mesoscopic simulations’, *PLoS computational biology*, vol. 13, no. 9, p. e1005726, 2017
- [51] Lykov, K., Pivkin, I.V.: ‘Computational models of eukaryotic cells in health and disease’, *Handbook of Materials Modeling: Applications: Current and Emerging Materials*, pp. 1–13, 2018
- [52] MathWorks: ‘Matlab documentation’, [<https://www.mathworks.com/help/optim/ug/lsqcurvefit.html>], 2018

- [53] Mego, M.: ‘Cirkulujúce a diseminované nádorové bunky’, *Onkológia (Bratisl.)*, vol. 5, no. 6, pp. 356–359, 2010
- [54] Moeendarbary, E., Ng, T., Zangeneh, M.: ‘Dissipative particle dynamics: introduction, methodology and complex fluid applications—a review’, *International Journal of Applied Mechanics*, vol. 1, no. 04, pp. 737–763, 2009
- [55] Mučka, F.: ‘Triangulácie pre modelovanie pohybu trojrozmerných elastických objektov v kvapaline pomocou softvéru espresso’, 2015
- [56] Neuman, K.C., Nagy, A.: ‘Single-molecule force spectroscopy: optical tweezers, magnetic tweezers and atomic force microscopy’, *Nature methods*, vol. 5, no. 6, p. 491, 2008
- [57] Peskin, C.S.: ‘The immersed boundary method’, *Acta numerica*, vol. 11, pp. 479–517, 2002
- [58] Pivkin, I.V., Peng, Z., et al.: ‘Biomechanics of red blood cells in human spleen and consequences for physiology and disease’, *Proceedings of the National Academy of Sciences*, vol. 113, no. 28, pp. 7804–7809, 2016
- [59] Ramaekers, F.C., Bosman, F.T.: ‘The cytoskeleton and disease’, *The Journal of Pathology: A Journal of the Pathological Society of Great Britain and Ireland*, vol. 204, no. 4, pp. 351–354, 2004
- [60] Rejniak, K.A.: ‘An immersed boundary framework for modelling the growth of individual cells: an application to the early tumour development’, *Journal of theoretical biology*, vol. 247, no. 1, pp. 186–204, 2007
- [61] Rejniak, K.A.: ‘Investigating dynamical deformations of tumor cells in circulation: predictions from a theoretical model’, *Frontiers in oncology*, vol. 2, p. 111, 2012
- [62] Rodrigues, R.O., Pinho, D., et al.: ‘A simple microfluidic device for the deformability assessment of blood cells in a continuous flow’, *Biomedical microdevices*, vol. 17, no. 6, p. 108, 2015
- [63] Rosin, C., Erkamp, M., et al.: ‘Exploring the stability limits of actin and its suprastructures’, *Biophysical Journal*, vol. 108, no. 2, p. 521a, 2015

-
- [64] Rossinelli, D., Tang, Y.H., et al.: ‘The in-silico lab-on-a-chip: petascale and high-throughput simulations of microfluidics at cell resolution’, *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, p. 2, ACM, 2015
- [65] Rowat, A., Lammerding, J., Ipsen, J.H.: ‘Mechanical properties of the cell nucleus and the effect of emerin deficiency’, *Biophysical journal*, vol. 91, no. 12, pp. 4649–4664, 2006
- [66] Rowat, A.C., Jaalouk, D.E., et al.: ‘Nuclear envelope composition determines the ability of neutrophil-type cells to passage through micron-scale constrictions’, *Journal of Biological Chemistry*, vol. 288, no. 12, pp. 8610–8618, 2013
- [67] Sarioglu, A.F., Aceto, N., et al.: ‘A microfluidic device for label-free, physical capture of circulating tumor cell clusters’, *Nature methods*, vol. 12, no. 7, p. 685, 2015
- [68] Schulze, H.G., Konorov, S.O., et al.: ‘Label-free imaging of mammalian cell nucleoli by raman microspectroscopy’, *Analyst*, vol. 138, no. 12, pp. 3416–3423, 2013
- [69] Seyfried, T.N., Huysentruyt, L.C.: ‘On the origin of cancer metastasis’, *Critical reviews in oncogenesis*, vol. 18, no. 1-2, p. 43, 2013
- [70] Shields IV, C.W., Johnson, L.M., et al.: ‘Elastomeric negative acoustic contrast particles for capture, acoustophoretic transport, and confinement of cells in microfluidic systems’, *Langmuir*, vol. 30, no. 14, pp. 3923–3927, 2014
- [71] Shields IV, C.W., Reyes, C.D., López, G.P.: ‘Microfluidic cell sorting: a review of the advances in the separation of cells from debulking to rare cell isolation’, *Lab on a Chip*, vol. 15, no. 5, pp. 1230–1249, 2015
- [72] Sigüenza, J., Mendez, S., Nicoud, F.: ‘How should the optical tweezers experiment be used to characterize the red blood cell membrane mechanics?’, *Biomechanics and modeling in mechanobiology*, vol. 16, no. 5, pp. 1645–1657, 2017
- [73] Stephens, A.D., Banigan, E.J., et al.: ‘Chromatin and lamin a determine two different mechanical response regimes of the cell nucleus’, *Molecular biology of the cell*, vol. 28, no. 14, pp. 1984–1996, 2017

- [74] Stinson, S., Alley, M., et al.: ‘Morphological and immunocytochemical characteristics of human tumor cell lines for use in a disease-oriented anticancer drug screen.’, *Anticancer research*, vol. 12, no. 4, pp. 1035–1053, 1992
- [75] Suresh, S.: ‘Biomechanics and biophysics of cancer cells’, *Acta Materialia*, vol. 55, no. 12, pp. 3989–4014, 2007
- [76] Suresh, S., Spatz, J., et al.: ‘Connections between single-cell biomechanics and human disease states: gastrointestinal cancer and malaria’, *Acta biomaterialia*, vol. 1, no. 1, pp. 15–30, 2005
- [77] Takeishi, N., Imai, Y., et al.: ‘Flow of a circulating tumor cell and red blood cells in microvessels’, *Physical Review E*, vol. 92, no. 6, p. 063011, 2015
- [78] Tomaiuolo, G.: ‘Biomechanical properties of red blood cells in health and disease towards microfluidics’, *Biomicrofluidics*, vol. 8, no. 5, p. 051501, 2014
- [79] Tseng, Y., Lee, J.S., et al.: ‘Micro-organization and visco-elasticity of the inter-phase nucleus revealed by particle nanotracking’, *Journal of cell science*, vol. 117, no. 10, pp. 2159–2167, 2004
- [80] Voldman, J.: ‘Electrical forces for microscale cell manipulation’, *Annu. Rev. Biomed. Eng.*, vol. 8, pp. 425–454, 2006
- [81] Wang, X., Chen, S., et al.: ‘Enhanced cell sorting and manipulation with combined optical tweezer and microfluidic chip technologies’, *Lab on a Chip*, vol. 11, no. 21, pp. 3656–3662, 2011
- [82] Wolf, K., Te Lindert, M., et al.: ‘Physical limits of cell migration: control by ecm space and nuclear deformation and tuning by proteolysis and traction force’, *J Cell Biol*, vol. 201, no. 7, pp. 1069–1084, 2013
- [83] Xiao, L., Liu, Y., et al.: ‘Effects of flowing rbcs on adhesion of a circulating tumor cell in microvessels’, *Biomechanics and modeling in mechanobiology*, vol. 16, no. 2, pp. 597–610, 2017
- [84] Yamada, M., Kano, K., et al.: ‘Microfluidic devices for size-dependent separation of liver cells’, *Biomedical microdevices*, vol. 9, no. 5, pp. 637–645, 2007

- [85] Yamada, M., Nakashima, M., Seki, M.: ‘Pinched flow fractionation: continuous size separation of particles utilizing a laminar flow profile in a pinched microchannel’, *Analytical chemistry*, vol. 76, no. 18, pp. 5465–5471, 2004
- [86] Yamada, M., Seki, M.: ‘Hydrodynamic filtration for on-chip particle concentration and classification utilizing microfluidics’, *Lab on a Chip*, vol. 5, no. 11, pp. 1233–1239, 2005
- [87] Ye, T., Phan-Thien, N., Lim, C.T.: ‘Particle-based simulations of red blood cells—a review’, *Journal of biomechanics*, vol. 49, no. 11, pp. 2255–2266, 2016
- [88] Zhang, H., Liu, K.K.: ‘Optical tweezers for single cells’, *Journal of the Royal Society interface*, vol. 5, no. 24, pp. 671–690, 2008
- [89] Zhang, P., Zhang, N., et al.: ‘A multiple time stepping algorithm for efficient multiscale modeling of platelets flowing in blood plasma’, *Journal of computational physics*, vol. 284, pp. 668–686, 2015

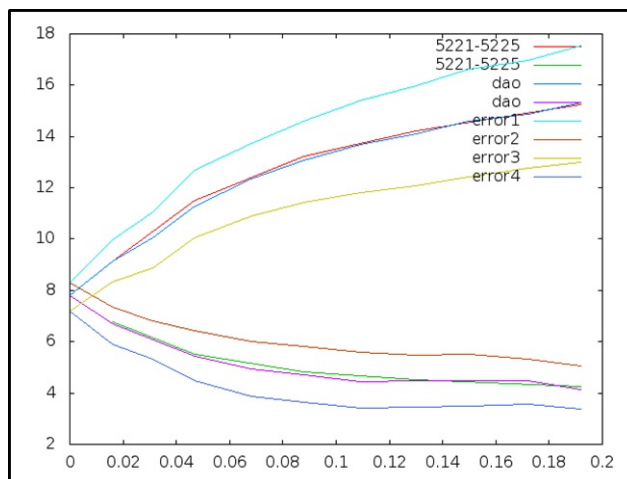
Prílohy

A Doplnujúce tabuľky

A.1 Výsledky kalibrácie RBC bunky

V časti 3.3 sme popísali proces kalibrácie RBC bunky. V tejto prílohe sú tabuľky, v ktorých je možné nájsť rôzne sady parametrov podľa toho aký typ bunky je potrebné v simulácii použiť. Keďže sme počas kalibrácie narazili na jemné rozdiely medzi rôznymi diskretizáciami povrchu bunky, uvádzame výsledky pre diskretizácie s 374, 642 a 1002 bodmi. Pre vybrané sady parametrov sú znázornené aj grafy vývoja tvaru bunky počas naťahovania, pre lepšiu predstavu aké správanie bunky daná sada parametrov spôsobuje.

Príklad takéhoto grafu môžeme vidieť tu.



Popis ku grafom

- **Osi:** Na x -ovej osi je vynesená ťahová sila v pN, ktorou je bunka deformovaná. Pri naťahovaní bunky sa predlžuje jej axiálny priemer (d_a) v smere naťahovacej sily a skrakuje jej priečný priemer (d_b). Na osi y je veľkosť axiálneho a

priečného priemeru týchto buniek v μm . Naťahovanie bunky je znázornené na obrázku 3.1.

- Červená a zelená krivka (v tomto príklade krivky označené ako "5221-5225" v legende) sú výsledkom numerického experimentu.
- Červená krivka bola porovnávaná s tmavomodrou krivkou (označená ako "dao" v legende). Táto tmavomodrá krivka je výsledkom biologického experimentu. Predstavuje dĺžku d_a bunky počas jej napínania. Svetlomodrá krivka nad ňou ("error 1") a žltá krivka pod ňou ("error 3") predstavuje rozsah hodnôt z biologického experimentu.
- Zelená krivka bola porovnávaná s fialovou krivkou ("dao"), ktorá je výsledkom z biologického experimentu. Táto zelená krivka predstavuje veľkosť d_b+ bunky počas naťahovania. Tmavočervená krivka nad ňou a modrá krivka pod ňou ("error 2" a "error 4") predstavujú rozsah hodnôt z biologického experimentu.

Bunka s 374 diskretizačnými bodmi

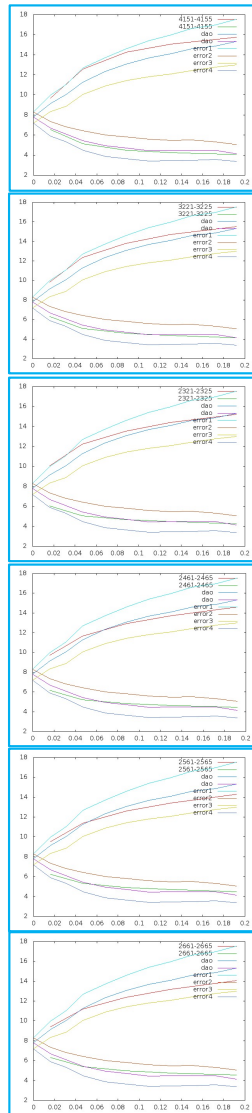
V tabuľke sú uvedené hodnoty odchýlky simulačných výsledkov od biologických výsledkov v ňafahovacom experimente pre rôzne sady parametrov simulácií. Šípky ukazujú ako sa mení tuhosť bunky v závislosti od zmeny parametrov.

Najnižšie hodnoty sú zelené a hodnoty prevyšujúce 0,8 sú šedé.

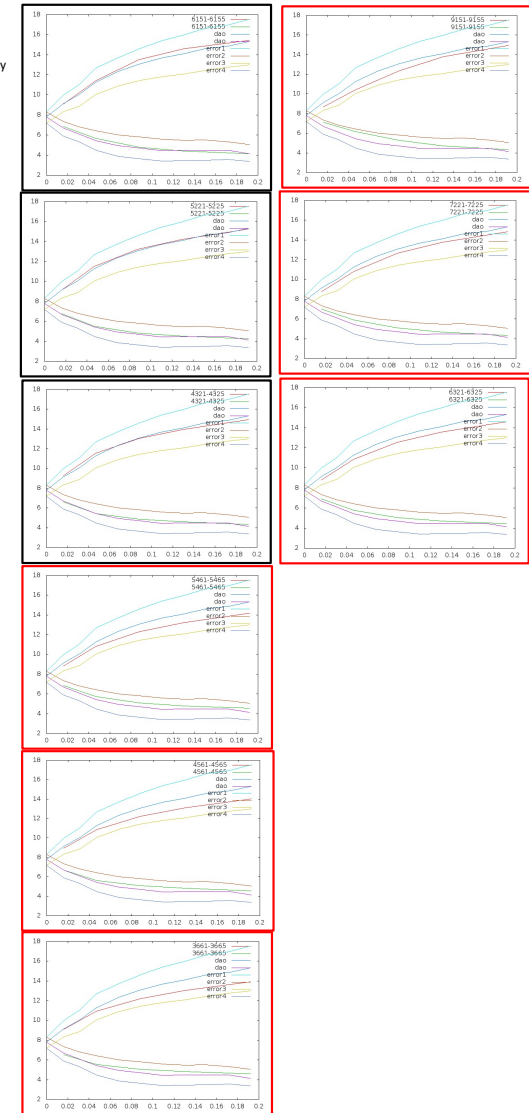
Napríklad, vyberme si bunku s hodnotou parametra $ks=0.008$. Táto bude tuhšia ako bunka s hodnotou parametra $ks=0.002$ v prípade že ostatné parametre sú zafixované.

Zvýraznili sme niekoľko sád parametrov (vyznačené hrubým orámovaním) ku ktorým sme uviedli aj korešpondujúce grafy. Môžeme vidieť ako vzerá tvar krivky ňafahovania v "mäkkšej" zóne tabuľky ňafavo a "tuhšej" zóne tabuľky napravo.

V hornej časti tabuľky môžeme pozorovať závislosť tvaru krivky od zmeny ks , zvyšné parametre sú zafixované. V dolnej časti tabuľky je viditeľná závislosť tvaru krivky od zmeny parametra kal , zvyšné parametre sú zafixované.



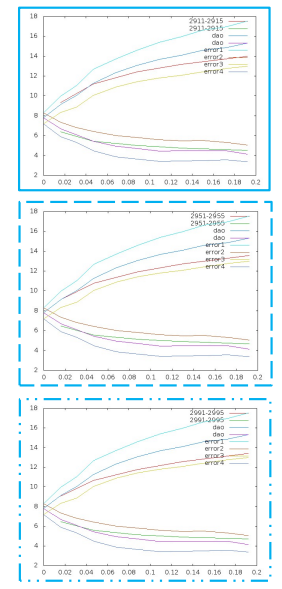
		Mäkkšie bunky				Tuhšie bunky				
		k_s								
kb	kal/ks	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009
0.001	0.01	4.4721	1.2078	0.8449	0.5809	0.3621	0.2217	0.2104	0.309	0.4313
0.001	0.02	1.6327	1.1613	0.8037	0.5343	0.3266	0.1989	0.2162	0.3288	0.454
0.001	0.03	1.6016	1.134	0.7771	0.5124	0.3067	0.1876	0.2225	0.3416	0.4685
0.001	0.04	1.5843	1.1146	0.7615	0.4978	0.2935	0.1814	0.2274	0.3507	0.4786
0.001	0.05	1.569	1.1017	0.7497	0.4867	0.2838	0.1775	0.2319	0.3577	0.4863
0.001	0.06	1.558	1.0914	0.7401	0.4779	0.2764	0.1749	0.2358	0.3634	0.4924
0.001	0.07	1.5489	1.0825	0.732	0.4707	0.2701	0.1729	0.2392	0.3682	0.4975
0.001	0.08	1.541	1.0748	0.7249	0.4643	0.2648	0.1716	0.2423	0.3724	0.5017
0.001	0.09	1.5341	1.0676	0.7187	0.4587	0.2601	0.1705	0.2452	0.3761	0.5056
0.003	0.01	4.4721	0.776	0.4857	0.2599	0.106	0.1689	0.31	0.4467	0.5715
0.003	0.02	4.4721	0.7245	0.4403	0.2191	0.0977	0.1993	0.3434	0.4783	0.601
0.003	0.03	1.035	0.6959	0.4146	0.1987	0.1018	0.2197	0.3639	0.4977	0.6195
0.003	0.04	1.0136	0.6772	0.3983	0.1859	0.109	0.2342	0.3781	0.5111	0.6323
0.003	0.05	0.9985	0.6633	0.3866	0.1775	0.1161	0.2448	0.3884	0.5211	0.642
0.003	0.06	0.9865	0.6528	0.3776	0.1716	0.1221	0.2533	0.3967	0.529	0.6495
0.003	0.07	0.9773	0.6444	0.3701	0.1672	0.1274	0.2601	0.4034	0.5355	0.6557
0.003	0.08	0.9696	0.6373	0.3642	0.1636	0.132	0.266	0.4091	0.5409	0.661
0.003	0.09	0.963	0.6312	0.3591	0.1608	0.1362	0.2711	0.4141	0.5457	0.6655
0.005	0.01	4.4721	0.5569	0.3047	0.1289	0.1525	0.2907	0.4302	0.5575	0.673
0.005	0.02	0.7916	0.5045	0.2664	0.128	0.1933	0.3337	0.469	0.5926	0.7053
0.005	0.03	0.7585	0.478	0.2501	0.1371	0.2216	0.3601	0.4928	0.6145	0.7258
0.005	0.04	0.7385	0.4623	0.2419	0.1495	0.2411	0.3779	0.5091	0.6298	0.7401
0.005	0.05	0.7253	0.4516	0.2379	0.16	0.2552	0.391	0.5214	0.6413	0.751
0.005	0.06	0.7146	0.4439	0.2354	0.168	0.2661	0.4012	0.5309	0.6504	0.7595
0.005	0.07	0.7067	0.4382	0.2339	0.1746	0.2749	0.4096	0.5387	0.6577	0.7665
0.005	0.08	0.7002	0.4331	0.2328	0.1804	0.2823	0.4166	0.5453	0.6639	0.7724
0.005	0.09	0.6948	0.4291	0.2321	0.1854	0.2888	0.4226	0.5509	0.6694	0.7775
0.007	0.01	4.4721	0.4293	0.2266	0.1659	0.2634	0.3982	0.5258	0.6433	0.751
0.007	0.02	0.6286	0.3857	0.2167	0.2047	0.315	0.4439	0.5671	0.6807	0.7853
0.007	0.03	0.5993	0.3689	0.2213	0.2345	0.3454	0.4719	0.5928	0.7044	0.8074
0.007	0.04	0.5826	0.3604	0.2313	0.2545	0.3661	0.4912	0.6107	0.7211	0.823
0.007	0.05	0.5713	0.3573	0.2382	0.2694	0.3815	0.5055	0.624	0.7336	0.8349
0.007	0.06	0.5644	0.3548	0.2441	0.2811	0.3935	0.5169	0.6345	0.7437	0.8443
0.007	0.07	0.5588	0.3528	0.2494	0.2908	0.4032	0.526	0.6431	0.7517	0.8519
0.007	0.08	0.5539	0.3517	0.2541	0.2989	0.4114	0.5337	0.6503	0.7585	0.8584
0.007	0.09	0.5502	0.351	0.2583	0.3059	0.4186	0.5403	0.6564	0.7643	0.8639
0.009	0.01	4.4721	0.3575	0.2231	0.2458	0.3578	0.483	0.6022	0.7126	0.8143
0.009	0.02	0.5282	0.3339	0.2476	0.2985	0.4109	0.531	0.6456	0.7519	0.8501
0.009	0.03	0.5053	0.3321	0.2701	0.3317	0.4432	0.5608	0.6729	0.777	0.8735
0.009	0.04	0.4947	0.3339	0.2875	0.3547	0.4655	0.5815	0.692	0.7948	0.8902
0.009	0.05	0.4884	0.3372	0.301	0.3716	0.4821	0.5969	0.7063	0.8082	0.9027
0.009	0.06	0.4845	0.3406	0.312	0.3849	0.4949	0.6089	0.7176	0.8187	0.9128
0.009	0.07	0.4815	0.3439	0.3212	0.3957	0.5054	0.6187	0.7268	0.8274	0.921
0.009	0.08	0.4797	0.3467	0.3289	0.4048	0.5142	0.6269	0.7345	0.8348	0.9278
0.009	0.09	0.4783	0.3495	0.3356	0.4126	0.5216	0.634	0.7411	0.841	0.9338
0.011	0.01	4.4721	0.324	0.2562	0.3196	0.4337	0.5528	0.6657	0.7704	0.8673
0.011	0.02	0.468	0.3257	0.302	0.3787	0.4893	0.6029	0.7106	0.8112	0.9045
0.011	0.03	0.4554	0.3377	0.334	0.415	0.5234	0.634	0.7393	0.8376	0.9289
0.011	0.04	0.4512	0.3491	0.3569	0.4399	0.5471	0.6559	0.7595	0.8562	0.9463
0.011	0.05	0.45	0.3591	0.3742	0.4583	0.5646	0.6721	0.7745	0.8703	0.9595
0.011	0.06	0.4505	0.3674	0.3877	0.4726	0.5782	0.6848	0.7864	0.8814	0.97
0.011	0.07	0.4515	0.3747	0.3988	0.4842	0.5893	0.6952	0.7961	0.8906	0.9786
0.011	0.08	0.4527	0.381	0.4081	0.494	0.5985	0.7038	0.8042	0.8982	0.9859
0.011	0.09	0.4537	0.3865	0.4161	0.5022	0.6064	0.7112	0.8113	0.9048	0.9921



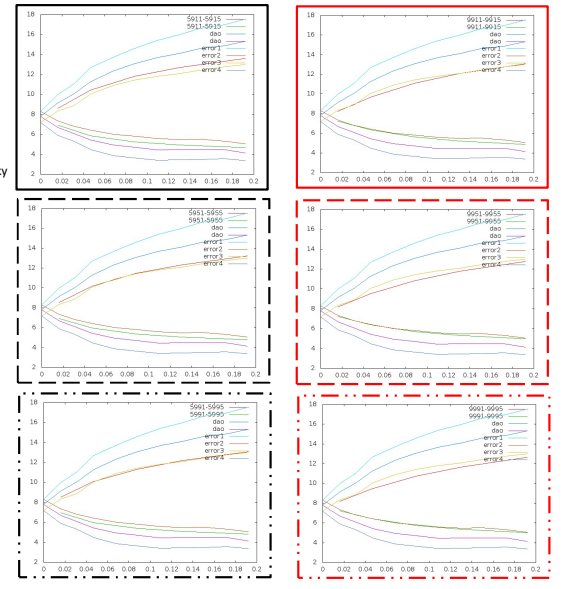
← Mäkkisie bunky k_s Tuhšie bunky →

kb	kal/ks	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009
0.013	0.01	4.4721	0.3178	0.3016	0.3847	0.498	0.6122	0.7202	0.8204	0.9131
0.013	0.02	0.4355	0.3421	0.3601	0.4476	0.5558	0.664	0.7665	0.8622	0.9512
0.013	0.03	0.4341	0.3662	0.3978	0.486	0.5913	0.6964	0.7961	0.8894	0.9766
0.013	0.04	0.4379	0.385	0.4239	0.5125	0.6158	0.719	0.817	0.909	0.9946
0.013	0.05	0.4427	0.4001	0.4434	0.532	0.6341	0.7361	0.8329	0.9236	1.0084
0.013	0.06	0.4473	0.4122	0.4585	0.5471	0.6484	0.7493	0.8453	0.9354	1.0194
0.013	0.07	0.4518	0.4221	0.4707	0.5593	0.66	0.7601	0.8555	0.9449	1.0284
0.013	0.08	0.456	0.4306	0.481	0.5695	0.6697	0.7693	0.864	0.9529	1.036
0.013	0.09	0.4595	0.4379	0.4897	0.5783	0.6779	0.777	0.8713	0.9598	1.0425
0.015	0.01	4.4721	0.3285	0.3491	0.4421	0.5536	0.6639	0.7676	0.8641	0.9537
0.015	0.02	0.4235	0.3716	0.4156	0.5078	0.6128	0.7167	0.8151	0.9069	0.9926
0.015	0.03	0.4331	0.4047	0.4568	0.5479	0.6498	0.7502	0.8456	0.9351	1.0185
0.015	0.04	0.4445	0.4289	0.4851	0.5755	0.6753	0.7737	0.8673	0.9551	1.0372
0.015	0.05	0.4545	0.4473	0.506	0.5958	0.6942	0.7912	0.8835	0.9703	1.0514
0.015	0.06	0.4633	0.4618	0.5222	0.6116	0.7091	0.805	0.8965	0.9823	1.0628
0.015	0.07	0.4707	0.4737	0.5353	0.6243	0.7211	0.8162	0.907	0.9922	1.0721
0.015	0.08	0.4774	0.4836	0.5463	0.635	0.7311	0.8257	0.9158	1.0005	1.08
0.015	0.09	0.4832	0.4921	0.5555	0.644	0.7396	0.8339	0.9235	1.0077	1.0868
0.017	0.01	4.4721	0.3494	0.3952	0.4929	0.6026	0.7095	0.81	0.9033	0.9899
0.017	0.02	0.4254	0.4068	0.4667	0.5609	0.6634	0.7636	0.8583	0.9469	1.0294
0.017	0.03	0.4456	0.4463	0.5104	0.6024	0.7011	0.7979	0.8895	0.9755	1.0558
0.017	0.04	0.4632	0.474	0.5403	0.6308	0.7274	0.822	0.9118	0.9962	1.0752
0.017	0.05	0.4779	0.4948	0.5623	0.6519	0.7469	0.8401	0.9285	1.0118	1.0898
0.017	0.06	0.4897	0.511	0.5794	0.6683	0.7623	0.8543	0.9418	1.0243	1.1014
0.017	0.07	0.4999	0.5242	0.5932	0.6816	0.7747	0.8658	0.9526	1.0345	1.1111
0.017	0.08	0.5084	0.5352	0.6046	0.6926	0.7851	0.8757	0.9618	1.043	1.1191
0.017	0.09	0.5158	0.5446	0.6143	0.7019	0.7941	0.8839	0.9696	1.0503	1.1261

Mäkkisie bunky ↑
 k_b
↓ Tuhšie bunky



↑ Mäkkisie bunky
 k_{at}
↓ Tuhšie bunky



kb	ka/ks	Mäkkie bunky				Tuhšie bunky				
		0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009
0.011	0.001	4.4721	0.3383	0.1813	0.2721	0.4228	0.5634	0.6889	0.8013	0.9022
0.011	0.002	4.4721	0.3103	0.1875	0.2899	0.4377	0.5755	0.6988	0.8097	0.9094
0.011	0.003	0.5916	0.2913	0.197	0.3064	0.4513	0.5865	0.708	0.8175	0.9162
0.011	0.004	0.5523	0.2783	0.2082	0.3215	0.4638	0.5968	0.7166	0.8248	0.9225
0.011	0.005	0.5225	0.2704	0.2202	0.3355	0.4753	0.6062	0.7246	0.8316	0.9285
0.011	0.01	0.4468	0.269	0.2373	0.3399	0.5212	0.6445	0.7572	0.85	0.9353
0.011	0.02	0.4165	0.3098	0.3498	0.4585	0.5789	0.6938	0.8002	0.8982	0.988
0.011	0.03	0.4195	0.345	0.394	0.4984	0.6132	0.7241	0.8275	0.9228	1.0107
0.011	0.04	0.4278	0.3707	0.4233	0.5248	0.6364	0.7448	0.8462	0.94	1.0267
0.011	0.05	0.4365	0.3898	0.4443	0.5438	0.6532	0.7599	0.86	0.9528	1.0386
0.011	0.06	0.4441	0.4048	0.4602	0.5582	0.6661	0.7716	0.8706	0.9627	1.0479
0.011	0.07	0.451	0.4167	0.4728	0.5695	0.6762	0.7808	0.8791	0.9706	1.0554
0.011	0.08	0.457	0.4264	0.4829	0.5789	0.6846	0.7884	0.8861	0.9771	1.0615
0.011	0.09	0.4622	0.4346	0.4914	0.5866	0.6915	0.7946	0.8919	0.9825	1.0667
0.013	0.001	4.4721	0.2775	0.1924	0.3282	0.4826	0.621	0.7434	0.8525	0.9505
0.013	0.002	4.4721	0.2555	0.2093	0.3479	0.4981	0.6333	0.7535	0.8609	0.9578
0.013	0.003	0.5134	0.2429	0.227	0.3656	0.5122	0.6446	0.7628	0.8688	0.9646
0.013	0.004	0.4759	0.2377	0.2444	0.3819	0.5251	0.655	0.7715	0.8762	0.9709
0.013	0.005	0.4481	0.2373	0.261	0.3967	0.5369	0.6647	0.7795	0.8831	0.9769
0.013	0.01	0.4386	0.2669	0.287	0.4551	0.5844	0.704	0.8129	0.9114	1.0121
0.013	0.02	0.3825	0.3368	0.4115	0.526	0.6437	0.7545	0.8568	0.9506	1.0369
0.013	0.03	0.4022	0.3837	0.4596	0.5677	0.6795	0.7859	0.8846	0.9758	1.0598
0.013	0.04	0.4212	0.4158	0.4913	0.5955	0.7038	0.8074	0.904	0.9935	1.0761
0.013	0.05	0.4373	0.4389	0.5139	0.6155	0.7215	0.8231	0.9182	1.0066	1.0884
0.013	0.06	0.4503	0.4567	0.5311	0.6307	0.7349	0.8352	0.9293	1.0168	1.0979
0.013	0.07	0.461	0.4706	0.5446	0.6427	0.7456	0.8448	0.9381	1.025	1.1056
0.013	0.08	0.4701	0.4821	0.5555	0.6526	0.7542	0.8526	0.9453	1.0318	1.112
0.013	0.09	0.4778	0.4915	0.5646	0.6608	0.7615	0.8592	0.9513	1.0375	1.1173
0.015	0.001	4.4721	0.2319	0.221	0.3803	0.5357	0.672	0.7917	0.8981	0.9936
0.015	0.002	4.4721	0.218	0.2445	0.4007	0.5515	0.6845	0.8019	0.9066	1.0008
0.015	0.003	0.4482	0.2149	0.2667	0.4193	0.5658	0.6959	0.8113	0.9146	1.0076
0.015	0.004	0.4128	0.219	0.2872	0.4361	0.579	0.7065	0.82	0.922	1.014
0.015	0.005	0.3877	0.227	0.3062	0.4516	0.5912	0.7163	0.8282	0.9289	1.02
0.015	0.01	0.3455	0.2343	0.351	0.5121	0.6398	0.7641	0.8619	0.9571	1.0453
0.015	0.02	0.3711	0.3749	0.469	0.5855	0.7006	0.808	0.9066	0.9972	1.0804
0.015	0.03	0.4071	0.4294	0.5196	0.6287	0.7377	0.8402	0.9351	1.0228	1.1036
0.015	0.04	0.4354	0.4655	0.5531	0.6577	0.7627	0.8623	0.955	1.0408	1.1202
0.015	0.05	0.4572	0.4914	0.5769	0.6785	0.781	0.8785	0.9697	1.0543	1.1328
0.015	0.06	0.4744	0.5109	0.5948	0.6943	0.795	0.8911	0.9811	1.0648	1.1426
0.015	0.07	0.4884	0.5263	0.6091	0.7068	0.8061	0.9011	0.9902	1.0732	1.1505
0.015	0.08	0.4998	0.5388	0.6206	0.717	0.8151	0.9093	0.9977	1.0802	1.157
0.015	0.09	0.5096	0.5492	0.6302	0.7255	0.8227	0.9161	1.004	1.086	1.1625
0.017	0.001	4.4721	0.2005	0.2564	0.4278	0.5831	0.7177	0.835	0.9391	1.0322
0.017	0.002	4.4721	0.1973	0.2834	0.4491	0.5992	0.7302	0.8452	0.9476	1.0394
0.017	0.003	0.3929	0.205	0.3082	0.4682	0.6139	0.7418	0.8547	0.9556	1.0462
0.017	0.004	0.3605	0.2184	0.3306	0.4855	0.6273	0.7525	0.8635	0.963	1.0526
0.017	0.005	0.3394	0.2343	0.351	0.5013	0.6397	0.7624	0.8718	0.9699	1.0585
0.017	0.01	0.3204	0.3134	0.4296	0.5634	0.689	0.8031	0.9057	0.999	1.0839
0.017	0.02	0.3763	0.4166	0.5216	0.6387	0.7513	0.8556	0.951	1.0387	1.1193
0.017	0.03	0.4257	0.476	0.5743	0.6832	0.7891	0.8884	0.9801	1.0647	1.1429
0.017	0.04	0.4613	0.515	0.6089	0.7129	0.8149	0.9112	1.0003	1.083	1.1597
0.017	0.05	0.4876	0.5427	0.6337	0.7345	0.8338	0.9278	1.0154	1.0968	1.1724
0.017	0.06	0.5081	0.5637	0.6525	0.7509	0.8484	0.9408	1.027	1.1076	1.1824
0.017	0.07	0.5242	0.5802	0.6673	0.7639	0.8599	0.9511	1.0366	1.1163	1.1904
0.017	0.08	0.5375	0.5936	0.6794	0.7746	0.8694	0.9597	1.0419	1.1234	1.1971
0.017	0.09	0.5486	0.6046	0.6894	0.7834	0.8772	0.9667	1.0508	1.1294	1.2027
0.019	0.001	4.4721	0.1826	0.2934	0.4715	0.6265	0.7591	0.8745	0.9764	1.0676
0.019	0.002	4.4721	0.1919	0.3226	0.4931	0.6427	0.7717	0.8848	0.9849	1.0748
0.019	0.003	0.3458	0.2098	0.3488	0.5128	0.6575	0.7834	0.8943	0.9928	1.0816
0.019	0.004	0.3175	0.2313	0.3723	0.5305	0.6711	0.7943	0.9031	1.0002	1.088
0.019	0.005	0.3015	0.2532	0.3939	0.5467	0.6836	0.8043	0.9113	1.0072	1.0939
0.019	0.01	0.3093	0.3469	0.4753	0.6099	0.7337	0.8452	0.9456	1.0364	1.1193
0.019	0.02	0.3924	0.4587	0.57	0.6867	0.797	0.8984	0.9915	1.0765	1.1548
0.019	0.03	0.4521	0.5217	0.6242	0.7322	0.8357	0.932	1.0209	1.1029	1.1786
0.019	0.04	0.493	0.5625	0.6599	0.7627	0.8621	0.9551	1.0416	1.1215	1.1956
0.019	0.05	0.5229	0.5917	0.6855	0.7849	0.8814	0.9723	1.0569	1.1355	1.2085
0.019	0.06	0.5456	0.6138	0.7051	0.8018	0.8963	0.9855	1.069	1.1465	1.2186
0.019	0.07	0.5619	0.63	0.7203	0.8152	0.9081	0.9953	1.0769	1.1535	1.2269
0.019	0.08	0.5782	0.645	0.7328	0.8263	0.9178	1.0048	1.0865	1.1626	1.2336
0.019	0.09	0.5904	0.6567	0.7431	0.8354	0.9259	1.0121	1.0931	1.1687	1.2393

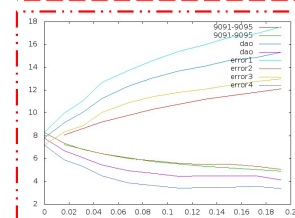
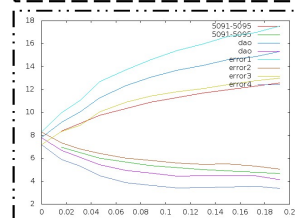
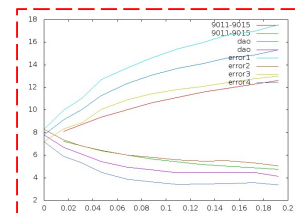
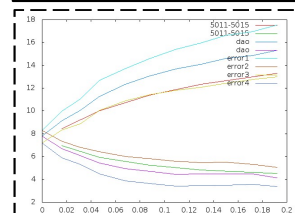
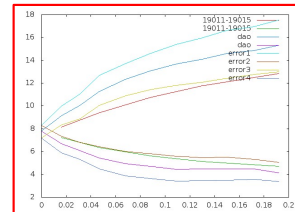
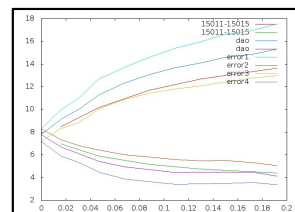
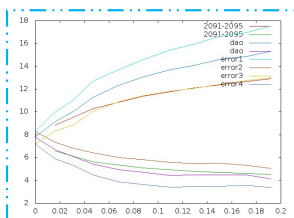
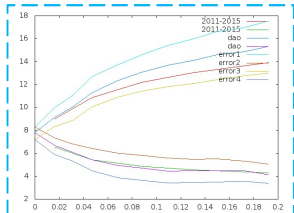
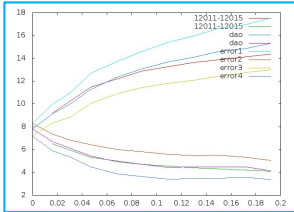
Mäkkie bunky

Mäkkie bunky

Tuhšie bunky

k_b

Tuhšie bunky



Bunka s 1002 diskretizačnými bodmi

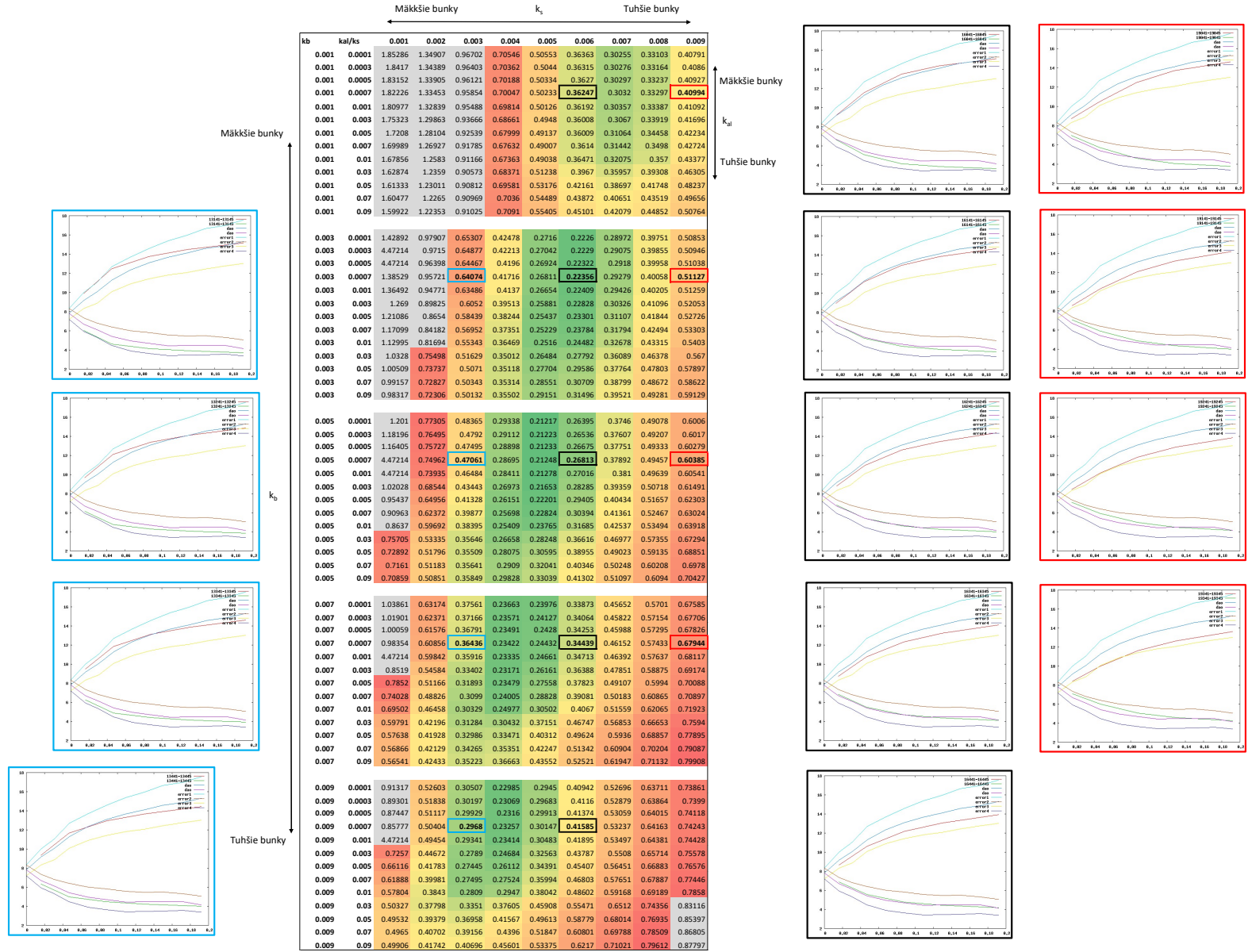
V tabuľke su uvedené hodnoty odchýlky simulačných výsledkov od biologických výsledkov v ňaťahovacom experimente pre rôzne sady parametrov simulácií. Šípky ukazujú ako sa mení tuhosť bunky v závislosti od zmeny parametrov.

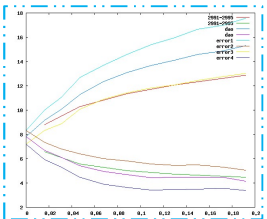
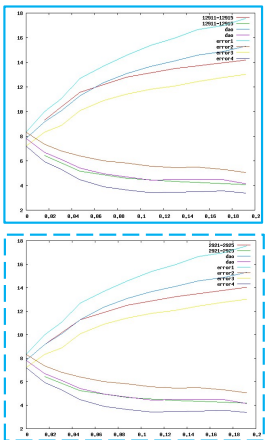
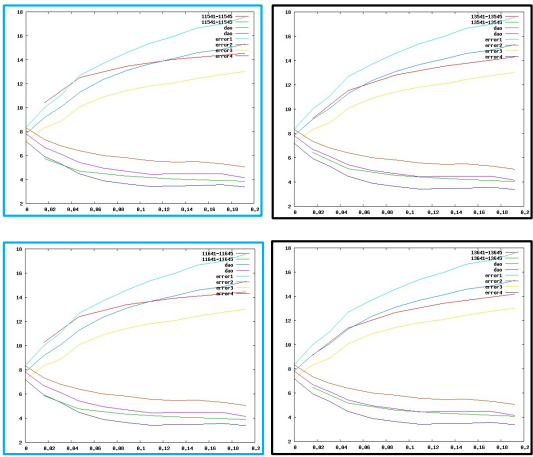
Najnižšie hodnoty sú zelené a hodnoty prevyšujúce 0,8 sú šedé.

Napríklad, vyberme si bunku s hodnotou parametra $k_s=0.008$. Táto bude tuhšia ako bunka s hodnotou parametra $k_s=0.002$ v prípade že ostatné parametre sú zafixované.

Zvýraznili sme niekoľko sád parametrov (vyznačené hrubým orámovaním) ku ktorým sme uviedli aj korešpondujúce grafy. Môžeme vidieť ako vzerá tvar krivky ňaťahovania v "mäkkšej" zóne tabuľky naľavo a "tuhšej" zóne tabuľky napravo.

V hornej časti tabuľky môžeme pozorovať závislosť tvaru krivky od zmeny k_s , zvyšné parametre sú zafixované. V dolnej časti tabuľky je viditeľná závislosť tvaru krivky od zmeny parametra k_{al} , zvyšné parametre sú zafixované.





Mäkkisie bunky

k_s

Tuhšie bunky

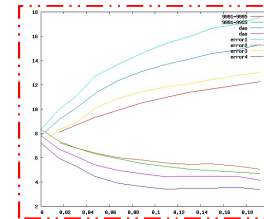
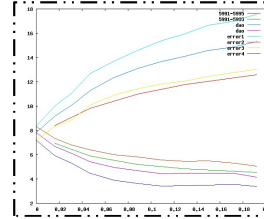
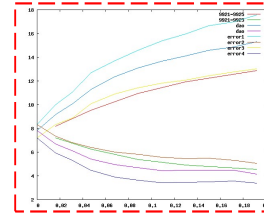
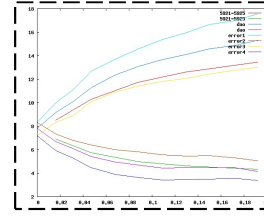
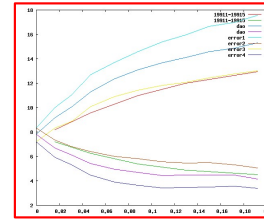
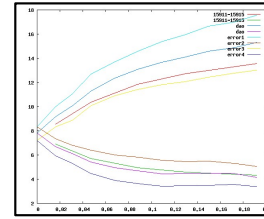
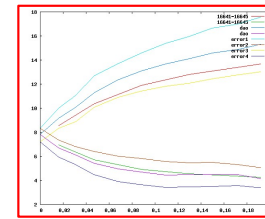
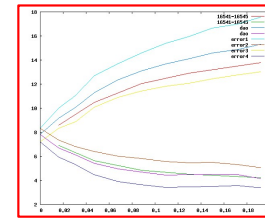
Mäkkisie bunky k_s Tuhšie bunky

kb	kal/ks	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009
0.011	0.0001	4.47214	0.44366	0.26282	0.25314	0.35224	0.47231	0.58788	0.69469	0.79294
0.011	0.0003	0.79097	0.43665	0.26147	0.25543	0.35498	0.47464	0.58981	0.69629	0.79429
0.011	0.0005	0.77249	0.43011	0.26033	0.25779	0.35768	0.47694	0.5917	0.69787	0.79562
0.011	0.0007	0.75587	0.42369	0.25929	0.26011	0.36034	0.4792	0.59356	0.69942	0.79894
0.011	0.001	4.47214	0.41525	0.25831	0.2636	0.36425	0.48251	0.5963	0.70171	0.79887
0.011	0.003	0.62629	0.37517	0.25846	0.26555	0.38816	0.50264	0.61301	0.71581	0.81077
0.011	0.005	0.56578	0.35469	0.26626	0.30784	0.40861	0.51991	0.62759	0.72815	0.82121
0.011	0.007	0.52826	0.34466	0.27739	0.327	0.42652	0.53504	0.6403	0.73887	0.83054
0.011	0.01	0.49416	0.34046	0.29551	0.35184	0.44907	0.55429	0.65659	0.75277	0.84273
0.011	0.03	0.45278	0.37781	0.38409	0.44812	0.5356	0.62911	0.72114	0.80895	0.89222
0.011	0.05	0.462	0.41182	0.42976	0.49365	0.57697	0.66569	0.75332	0.83761	0.91755
0.011	0.07	0.47358	0.43461	0.45755	0.52084	0.60194	0.68814	0.77321	0.8553	0.93363
0.011	0.09	0.48315	0.45104	0.47653	0.53963	0.61891	0.70341	0.78703	0.8678	0.9448
0.013	0.0001	4.47214	0.37884	0.24467	0.29008	0.40646	0.52801	0.64148	0.74536	0.84073
0.013	0.0003	0.70632	0.37245	0.24504	0.29327	0.40944	0.53045	0.64347	0.74702	0.84212
0.013	0.0005	0.68791	0.36683	0.24571	0.29644	0.41237	0.53285	0.64543	0.74864	0.84349
0.013	0.0007	0.67108	0.36166	0.24656	0.29958	0.41525	0.53521	0.64736	0.75024	0.84485
0.013	0.001	4.47214	0.35451	0.24807	0.30421	0.41958	0.53867	0.6502	0.7526	0.84684
0.013	0.003	0.54568	0.32533	0.26323	0.33319	0.44529	0.55981	0.66754	0.76711	0.85915
0.013	0.005	0.49104	0.31574	0.28256	0.35843	0.46728	0.57803	0.68271	0.77988	0.87
0.013	0.007	0.45933	0.31569	0.30214	0.38035	0.48618	0.59377	0.69607	0.79129	0.87964
0.013	0.01	0.43499	0.32436	0.32918	0.4084	0.51028	0.61399	0.71321	0.80586	0.89244
0.013	0.03	0.43374	0.4028	0.43966	0.51431	0.60284	0.69361	0.78188	0.86578	0.94514
0.013	0.05	0.46125	0.45017	0.49244	0.56399	0.64722	0.73289	0.81646	0.89674	0.97269
0.013	0.07	0.48244	0.47997	0.5239	0.59375	0.67429	0.75701	0.83813	0.91612	0.99019
0.013	0.09	0.49843	0.50049	0.54541	0.614	0.69273	0.77376	0.8532	0.92975	1.00264
0.015	0.0001	4.47214	0.32761	0.24561	0.33127	0.45664	0.57829	0.68954	0.79105	0.88372
0.015	0.0003	4.47214	0.32245	0.24785	0.335	0.45978	0.5808	0.69158	0.79274	0.88521
0.015	0.0005	0.61599	0.31807	0.25013	0.33867	0.46286	0.58327	0.69358	0.7944	0.88661
0.015	0.0007	4.47214	0.31418	0.25261	0.34228	0.466	0.58569	0.69556	0.79604	0.88799
0.015	0.001	4.47214	0.30916	0.25648	0.34773	0.47044	0.58926	0.69846	0.79845	0.89002
0.015	0.003	0.47898	0.29368	0.28446	0.38013	0.49733	0.61097	0.71627	0.81331	0.90273
0.015	0.005	0.43198	0.29689	0.31208	0.40787	0.52031	0.62966	0.73185	0.82635	0.91388
0.015	0.007	0.40838	0.30774	0.33733	0.43157	0.54013	0.64617	0.7456	0.83791	0.92383
0.015	0.01	0.39548	0.32875	0.37016	0.46169	0.56521	0.6671	0.76325	0.85311	0.93691
0.015	0.03	0.43678	0.43945	0.49442	0.57436	0.66232	0.75035	0.83521	0.91582	0.99208
0.015	0.05	0.48004	0.49643	0.55209	0.62712	0.70921	0.79164	0.87183	0.94878	1.02163
0.015	0.07	0.50922	0.53095	0.58632	0.65879	0.73785	0.81744	0.89513	0.96943	1.04039
0.015	0.09	0.52996	0.55452	0.6096	0.68039	0.75751	0.83535	0.91127	0.98397	1.05372
0.017	0.0001	4.47214	0.28877	0.26029	0.37288	0.50279	0.62373	0.7333	0.83215	0.92272
0.017	0.0003	4.47214	0.28541	0.26402	0.37694	0.50604	0.62629	0.73537	0.83386	0.92416
0.017	0.0005	0.5539	0.28261	0.26779	0.38093	0.50922	0.62881	0.73741	0.83569	0.92558
0.017	0.0007	4.47214	0.28012	0.27158	0.38485	0.51235	0.63129	0.73942	0.83735	0.92698
0.017	0.001	4.47214	0.27743	0.27734	0.39073	0.51705	0.63494	0.74238	0.8398	0.92905
0.017	0.003	0.42385	0.278	0.31427	0.42546	0.54491	0.65722	0.76051	0.85493	0.94189
0.017	0.005	0.38608	0.29424	0.34734	0.45469	0.56855	0.67653	0.77633	0.86834	0.95327
0.017	0.007	0.37257	0.31509	0.37626	0.47958	0.58901	0.69343	0.79028	0.88033	0.96355
0.017	0.01	0.37213	0.34662	0.41258	0.51119	0.61488	0.71489	0.80864	0.89584	0.97712
0.017	0.03	0.45388	0.48058	0.5464	0.62871	0.71545	0.80095	0.88303	0.96087	1.03439
0.017	0.05	0.50963	0.54425	0.60748	0.68389	0.76447	0.84424	0.92145	0.99521	1.06519
0.017	0.07	0.54481	0.58229	0.64392	0.71713	0.79452	0.87139	0.94593	1.01724	1.08518
0.017	0.09	0.56912	0.60795	0.66868	0.74004	0.8151	0.89002	0.96282	1.03273	1.09929
0.019	0.0001	4.47214	0.26137	0.28361	0.41345	0.54568	0.66561	0.77307	0.8703	0.95862
0.019	0.0003	4.47214	0.25991	0.28838	0.41773	0.549	0.66821	0.77517	0.87203	0.96007
0.019	0.0005	4.47214	0.25885	0.29312	0.42193	0.55226	0.67077	0.77723	0.87373	0.96151
0.019	0.0007	4.47214	0.25847	0.29781	0.42605	0.55545	0.67328	0.77926	0.87541	0.96293
0.019	0.001	4.47214	0.25874	0.30476	0.43207	0.56023	0.67698	0.78226	0.87789	0.96502
0.019	0.003	0.37855	0.27582	0.34774	0.46837	0.58865	0.69973	0.80085	0.89322	0.97811
0.019	0.005	0.35204	0.30384	0.38439	0.49857	0.61272	0.71919	0.81701	0.90674	0.98966
0.019	0.007	0.34928	0.33286	0.41564	0.52441	0.6338	0.73632	0.83128	0.91892	1.00002
0.019	0.01	0.36244	0.37229	0.45447	0.55702	0.66035	0.7584	0.84977	0.9349	1.0138
0.019	0.03	0.47918	0.52267	0.59491	0.67822	0.76354	0.8469	0.92642	1.00163	1.07286
0.019	0.05	0.54436	0.59114	0.65867	0.73551	0.81439	0.89176	0.96623	1.03736	1.10503
0.019	0.07	0.58406	0.63174	0.69695	0.77016	0.84552	0.91974	0.99158	1.06023	1.12578
0.019	0.09	0.61134	0.65901	0.72262	0.79397	0.86711	0.93931	1.00938	1.07634	1.14064

Mäkkisie bunky

k_{cl}

Tuhšie bunky



A.2 Výsledky simulácií v ktorých bol použitý kvadratický potenciál.

prierez	a	K1/K0	0.00625	0.0125	0.025	0.05	0.1	0.2
13	10	0.00001	1	1	1	1	0	0
13	10	0.0001	1	1	1	1	1	1
13	10	0.001	1	1	1	1	1	1
13	15	0.00001	1	1	1	1	1	1
13	15	0.0001	1	1	1	1	0	0
13	15	0.001	1	1	1	1	1	1
13	20	0.00001	1	1	1	1	1	1
13	20	0.0001	1	1	1	1	1	1
13	20	0.001	1	1	1	1	0	0
15	10	0.00001	1	1	1	1	1	1
15	10	0.0001	1	1	1	1	1	1
15	10	0.001	1	1	1	1	1	1
15	15	0.00001	1	1	1	1	1	0
15	15	0.0001	0	1	1	1	1	1
15	15	0.001	0	1	1	1	1	1
15	20	0.00001	0	1	1	1	1	1
15	20	0.0001	1	1	1	1	1	0
15	20	0.001	0	1	1	1	1	1
17	10	0.00001	0	1	1	1	1	1
17	10	0.0001	0	1	1	1	1	1
17	10	0.001	1	1	1	1	1	0
17	15	0.00001	1	1	1	1	1	1
17	15	0.0001	1	1	1	1	1	1
17	15	0.001	1	1	1	1	1	1
17	20	0.00001	1	1	1	1	0	0
17	20	0.0001	0	0	1	1	1	1
17	20	0.001	0	0	1	1	1	1
19	10	0.00001	0	0	1	1	1	1
19	10	0.0001	1	1	1	1	0	0
19	10	0.001	0	0	1	1	1	1
19	15	0.00001	0	0	1	1	1	1
19	15	0.0001	0	0	1	1	1	1
19	15	0.001	1	1	1	1	0	0
19	20	0.00001	0	1	1	1	1	1
19	20	0.0001	0	1	1	1	1	1
19	20	0.001	0	1	1	1	1	1

A.3 Výsledky simulácií v ktorých bol použitý FENE potenciál.

prierez	a	r_{max}/K	0.00625	0.0125	0.025	0.05	0.1	0.2
13	10	6	1	1	1	0	0	0
13	10	8	1	1	1	1	0	0
13	10	10	1	1	1	0	0	0
13	10	12	1	1	1	0	0	0
13	15	6	1	1	1	1	1	0
13	15	8	1	1	1	1	1	0
13	15	10	1	1	1	1	1	0
13	15	12	1	1	1	1	1	0
13	20	6	0	1	1	1	0	0
13	20	8	1	1	1	1	0	0
13	20	10	1	1	1	1	0	0
13	20	12	0	0	1	1	0	0
15	10	6	1	1	1	1	1	1
15	10	8	0	1	1	1	1	1
15	10	10	0	1	1	1	1	1
15	10	12	0	1	1	1	1	1
15	15	6	0	1	1	1	1	1
15	15	8	0	1	1	1	1	1
15	15	10	0	1	1	1	1	1
15	15	12	0	1	1	1	1	1
15	20	6	0	0	1	1	1	1
15	20	8	0	0	1	1	1	1
15	20	10	0	0	1	1	1	1
15	20	12	0	0	1	1	1	1
17	10	6	1	1	1	1	1	1
17	10	8	1	1	1	1	1	1
17	10	10	1	1	1	1	1	1
17	10	12	1	1	1	1	1	1
17	15	6	0	1	1	1	1	1
17	15	8	0	1	1	1	1	1
17	15	10	0	1	1	1	1	1
17	15	12	0	1	1	1	1	1
17	20	6	0	0	1	1	1	1
17	20	8	0	0	1	1	1	1
17	20	10	0	0	1	1	1	1
17	20	12	0	0	1	1	1	1
19	10	6	1	1	1	1	1	1
19	10	8	1	1	1	1	1	1
19	10	10	1	1	1	1	1	1
19	10	12	1	1	1	1	1	1
19	15	6	1	1	1	1	1	1
19	15	8	0	1	1	1	1	1
19	15	10	0	1	1	1	1	1
19	15	12	0	1	1	1	1	1
19	20	6	0	1	1	1	1	1
19	20	8	0	0	1	1	1	1
19	20	10	0	0	1	1	1	1
19	20	12	0	0	1	1	1	1

B Python skripty

B.1 Geometria zúženého kanála

```
1 import espressomd
2 import object_in_fluid as oif
3
4 from espressomd import shapes
5
6 # GEOMETRY
7 # rectangular channel with size boxX*boxY*boxZ, the width of the
   borders is set in width
8 # and narrow rectangle part in the middle with cross-section ay*az
   and length ax
9 boxX = 80.0
10 boxY = 40.0
11 boxZ = 40.0
12 width = 2.0
13
14 def FillBoundaries(boundaries, vtk_directory, ax, ay, az):
15
16     # center of the channel
17     cx = boxX / 2
18     cy = boxY / 2
19     cz = boxZ / 2
20
21     # size of obstacles that create the narrow part
22     # horizontal
23     ah = ax
24     bh = 2 * width + boxY
25     ch = width + (boxZ - az) / 2
26     # vertical
27     av = ax
```

```
28     bv = width + (boxY - ay) / 2
29     cv = 2 * width + boxZ
30
31     # corners for obstacles
32     # A
33     A_x = cx - ax / 2
34     A_y = -width
35     A_z = -width
36     # B
37     B_x = cx - ax / 2
38     B_y = (boxY + ay) / 2
39     B_z = -width
40     # C
41     C_x = cx - ax / 2
42     C_y = -width
43     C_z = (boxZ + az) / 2
44
45     wallBottom = shapes.Rhomboid(corner=[-width, -width, -width], a
=[boxX+2*width, 0.0, 0.0], b=[0.0, boxY+2*width, 0.0], c=[0.0,
0.0, width], direction=1)
46     wallTop = shapes.Rhomboid(corner=[-width, -width, boxZ], a=[
boxX+2*width, 0.0, 0.0], b=[0.0, boxY+2*width, 0.0], c=[0.0,
0.0, width], direction=1)
47     wallBack = shapes.Rhomboid(corner=[-width, boxY, -width], a=[
boxX+2*width, 0.0, 0.0], b=[0.0, width, 0.0], c=[0.0, 0.0, boxZ
+2*width], direction=1)
48     wallFront = shapes.Rhomboid(corner=[-width, -width, -width], a
=[boxX+2*width, 0.0, 0.0], b=[0.0, width, 0.0], c=[0.0, 0.0,
boxZ+2*width], direction=1)
49     horizBottom = shapes.Rhomboid(corner=[A_x, A_y, A_z], a=[ah, 0,
0], b=[0, bh, 0], c=[0, 0, ch], direction = 1)
50     horizTop = shapes.Rhomboid(corner=[C_x, C_y, C_z], a=[ah, 0,
0], b=[0, bh, 0], c=[0, 0, ch], direction = 1)
51     verticLeft = shapes.Rhomboid(corner=[A_x, A_y, A_z], a=[av, 0,
0], b=[0, bv, 0], c=[0, 0, cv], direction = 1)
52     verticRight = shapes.Rhomboid(corner=[B_x, B_y, B_z], a=[av, 0,
0], b=[0, bv, 0], c=[0, 0, cv], direction = 1)
53
54     oif.output_vtk_rhomboid(rhom_shape=wallBottom, out_file=
vtk_directory + "/wallBottom.vtk")
55     oif.output_vtk_rhomboid(rhom_shape=wallTop, out_file=
vtk_directory + "/wallTop.vtk")
```



```

56     oif.output_vtk_rhomboid(rhom_shape=wallBack, out_file=
vtk_directory + "/wallBack.vtk")
57     oif.output_vtk_rhomboid(rhom_shape=wallFront, out_file=
vtk_directory + "/wallFront.vtk")
58     oif.output_vtk_rhomboid(rhom_shape=horizBottom, out_file=
vtk_directory + "/horizBottom.vtk")
59     oif.output_vtk_rhomboid(rhom_shape=horizTop, out_file=
vtk_directory + "/horizTop.vtk")
60     oif.output_vtk_rhomboid(rhom_shape=verticLeft, out_file=
vtk_directory + "/verticLeft.vtk")
61     oif.output_vtk_rhomboid(rhom_shape=verticRight, out_file=
vtk_directory + "/verticRight.vtk")
62
63     boundaries.append(wallBottom)
64     boundaries.append(wallTop)
65     boundaries.append(wallBack)
66     boundaries.append(wallFront)
67     boundaries.append(horizBottom)
68     boundaries.append(horizTop)
69     boundaries.append(verticLeft)
70     boundaries.append(verticRight)
71
72     return 0

```

B.2 Viazané interakcie - harmonický potenciál

```

1  import espressomd
2  import object_in_fluid as oif
3
4  import Boundaries_15
5  from Boundaries_15 import *
6
7  from espressomd import lb
8  from espressomd import lbboundaries
9  from espressomd import shapes
10 from espressomd import interactions
11 from espressomd.interactions import HarmonicBond
12
13 import numpy as np
14 import os, sys
15 import random
16

```

PRÍLOHY

```
17 # this is a script to try the nucleus with bonded interactions
18
19 #####
20 #INPUT DATA
21 #####
22 large_number = 10000000.0
23
24 # cell
25 membrane_radius = float(sys.argv[1])
26 membrane_noNodes = int(sys.argv[2])
27 membrane_ks = float(sys.argv[3])
28 membrane_kb = float(sys.argv[4])
29 membrane_kal = float(sys.argv[5])
30 membrane_kag = float(sys.argv[6])
31 membrane_kv = float(sys.argv[7])
32 membrane_kvisc = float(sys.argv[8])
33 membrane_mass = float(sys.argv[9])
34 # nucleus
35 nucleus_radius = float(sys.argv[10])
36 nucleus_noNodes = int(sys.argv[11])
37 nucleus_ks = float(sys.argv[12])
38 nucleus_kb = float(sys.argv[13])
39 nucleus_kal = float(sys.argv[14])
40 nucleus_kag = float(sys.argv[15])
41 nucleus_kv = float(sys.argv[16])
42 nucleus_kvisc = float(sys.argv[17])
43 nucleus_mass = float(sys.argv[18])
44 # fluid force
45 fluid_force = float(sys.argv[19])
46 # M_force is the force applied to each point of the cell (M as
    membrane)
47 M_force = float(sys.argv[20])
48 sim_no = sys.argv[21]
49 # harmonic bond parameter
50 par_k = float(sys.argv[22])
51 angle_par = float(sys.argv[23])
52 # parameters of the constriction
53 par_ax = float(sys.argv[24])
54 par_ay = float(sys.argv[25])
55 par_az = float(sys.argv[26])
56
57 #potrebujes mat pricinok output a v nom priecinok data a vtk, do
```

```
    data sa ti ukladaju datove subory, do vtk sa ti ukladaju vtkacka
58 directory_data = "output/data/sim"+str(sim_no)
59 os.makedirs(directory_data)
60 directory_vtk = "output/vtk/sim"+str(sim_no)
61 os.makedirs(directory_vtk)
62 vtk_directory = directory_vtk+"/vtk"
63 os.makedirs(vtk_directory)
64
65 #CELL
66 originX = 3*membrane_radius/2
67 originY = boxY/2
68 originZ = boxZ/2
69 membrane_partMass = membrane_mass/membrane_noNodes
70 membrane_stretch = float(membrane_radius)
71 #NUCLEUS
72 nucleus_partMass = nucleus_mass/nucleus_noNodes
73 nucleus_stretch = float(nucleus_radius)
74
75 #INPUT FILES
76 membraneNodes = "input/sphere" + str(membrane_noNodes) + "nodes.dat"
77 membraneTriangles = "input/sphere" + str(membrane_noNodes) + "
    triangles.dat"
78 nucleusNodes = "input/sphere" + str(nucleus_noNodes) + "nodes.dat"
79 nucleusTriangles = "input/sphere" + str(nucleus_noNodes) + "
    triangles.dat"
80
81 #LBFLUID
82 LBgrid = 1.0
83 density = 1.025
84 viscosity = 1.5
85 time_step = 0.1
86
87 #FRICTION
88 # surface is the refence sphere with r = 4
89 surface = 4*np.pi*4**2
90 surface_sph = 4*np.pi*membrane_radius**2
91 friction = (393.0/membrane_noNodes)*np.sqrt(surface_sph/surface)
    *((5.6-1.82)/(6.0/1.025-1.5)*(viscosity-1.5)+(10-1.82)/(6-1.025)
    *(density-1.025)+1.82)
92 #print ("surface_sph:" + str(surface_sph))
93 #print ("friction_sph:" + str(friction))
```

```
94
95 system = espressomd.System(box_l=[boxX, boxY, boxZ])
96 system.cell_system.skin = 0.2
97 system.time_step = time_step
98
99 #ITERATION PARAMETERS
100 cycle = 0
101 maxCycle = 500000000000
102 integr_steps = 100
103
104 #####
105 #initialization
106 #####
107
108 # creating templates
109 membrane_type = oif.OifCellType(nodes_file=membraneNodes,
110     triangles_file=membraneTriangles, check_orientation=False,
111     system=system, ks=membrane_ks, kb=
112     membrane_kb, kal=membrane_kal, kag=membrane_kag,
113     kv=membrane_kv, kvisc=membrane_kvisc,
114     resize=[membrane_stretch, membrane_stretch, membrane_stretch],
115     normal=True)
116 nucleus_type = oif.OifCellType(nodes_file=nucleusNodes,
117     triangles_file=nucleusTriangles, check_orientation=False,
118     system=system, ks=nucleus_ks, kb=
119     nucleus_kb, kal=nucleus_kal, kag=nucleus_kag,
120     kv=nucleus_kv, kvisc=nucleus_kvisc,
121     resize=[nucleus_stretch, nucleus_stretch, nucleus_stretch],
122     normal=True)
123
124 # creating the cell
125 membrane = oif.OifCell(cell_type = membrane_type, particle_type=0,
126     origin=[originX,originY,originZ], particle_mass=
127     membrane_partMass)
128 nucleus = oif.OifCell(cell_type = nucleus_type, particle_type=1,
129     origin=[originX,originY,originZ], particle_mass=nucleus_partMass
130 )
131
132 #info about the cell membrane and nucleus membrane
133 surface_sph = membrane.mesh.surface()
134 min_hrana_sph = membrane.mesh.min_edge_length()
135 max_hrana_sph = membrane.mesh.max_edge_length()
```

```

124 aver_hrana_sph = membrane.mesh.aver_edge_length()
125 out_file = open(directory_data + "/sphere_info_" + str(sim_no) + ".
    dat", "a")
126 out_file.write ("surface min_edge max_edge ave_edge \n")
127 out_file.write(str(surface_sph)+" "+str(min_hrana_sph)+" "+str(
    max_hrana_sph)+" "+str(aver_hrana_sph))
128 out_file.close()
129
130 surface_nuc = nucleus.mesh.surface()
131 min_hrana_nuc = nucleus.mesh.min_edge_length()
132 max_hrana_nuc = nucleus.mesh.max_edge_length()
133 aver_hrana_nuc = nucleus.mesh.aver_edge_length()
134 out_file = open(directory_data + "/nucleus_info_" + str(sim_no) + "
    .dat", "a")
135 out_file.write ("surface min_edge max_edge ave_edge \n")
136 out_file.write(str(surface_nuc)+" "+str(min_hrana_nuc)+" "+str(
    max_hrana_nuc)+" "+str(aver_hrana_nuc))
137 out_file.close()
138
139
140 small_angle = np.pi/angle_par
141 countbonds = 0
142 average_dlzka = 0
143 max_dlzka = - large_number
144 min_dlzka = large_number
145
146 lines = []
147 pairs = []
148 for p in nucleus.mesh.points:
149     neighbors = nucleus.mesh.neighbors[p.id]
150     normal = neighbors.outer_normal()
151     lengthNormal = oif.norm(normal)
152     pPos = p.get_pos()
153     for pMembrane in membrane.mesh.points:
154         pMembranePos = pMembrane.get_pos()
155         tmpConnection = pMembranePos - pPos
156         tmpDist = oif.norm(tmpConnection)
157         angle = np.arccos(np.dot(tmpConnection,normal)/(tmpDist *
            lengthNormal))
158         if angle < small_angle:
159             harmonicInter = HarmonicBond(k=par_k, r_0=tmpDist)
160             system.bonded_inter.add(harmonicInter)

```

```
161         p.part.add_bond((harmonicInter, pMembrane.part_id))
162         tmpelist = [pPos[0], pPos[1], pPos[2], pMembranePos[0],
pMembranePos[1], pMembranePos[2]]
163         tmppair = [p.id, pMembrane.id]
164         lines.append(tmpelist)
165         pairs.append(tmppair)
166         countbonds += 1
167
168 #initial info about the bonds , number, min_length, max_length,
average
169 for pair in pairs:
170     pPos = nucleus.mesh.points[pair[0]].get_pos()
171     pMembranePos = membrane.mesh.points[pair[1]].get_pos()
172     tmpConnection = pMembranePos - pPos
173     dlzka = oif.norm(tmpConnection)
174     average_dlzka = average_dlzka + dlzka
175     if dlzka > max_dlzka:
176         max_dlzka = dlzka
177     if dlzka < min_dlzka:
178         min_dlzka = dlzka
179 average_dlzka = average_dlzka / countbonds
180 out_file = open(directory_data + "/bonds_initial_info_" + str(
sim_no) + ".dat", "a")
181 out_file.write ("NoBonds min_edge max_edge ave_edge \n")
182 out_file.write(str(countbonds)+" "+str(min_dlzka)+" "+str(max_dlzka)
)+" "+str(average_dlzka))
183 out_file.close()
184 average_dlzka = 0
185 max_dlzka = - large_number
186 min_dlzka = large_number
187
188
189 # cell-wall interactions
190 system.non_bonded_inter[0,10].soft_sphere.set_params(a = 0.0001, n
= 1.2, cutoff = 0.1, offset = 0.0)
191 system.non_bonded_inter[1,10].soft_sphere.set_params(a = 0.0001, n
= 1.2, cutoff = 0.1, offset = 0.0)
192
193
194 # fluid
195 #lbf = espressomd.lb.LBFluid(agrid = LBgrid, dens = density, visc =
viscosity, tau = time_step, fric = friction , ext_force_density
```

```
    = [fluid_force, 0.0, 0.0])
196 #system.actors.add(lbf)
197
198 # fluid
199 lb_params = {'agrid': LBgrid, 'dens': density, 'visc': viscosity, '
    tau': time_step,
200             'ext_force_density': [fluid_force, 0, 0]}
201 lbf = espressomd.lb.LBFluid(**lb_params)
202 system.actors.add(lbf)
203
204 system.thermostat.set_lb(LB_fluid=lbf, gamma=friction)
205
206 boundaries = list()
207 FillBoundaries(boundaries, vtk_directory, par_ax, par_ay, par_az)
208 for boundary in boundaries:
209     system.lbboundaries.add(lbboundaries.LBBoundary(shape=boundary)
    )
210     system.constraints.add(shape=boundary, particle_type=10,
    penetrable=False)
211
212 #force on individual mesh points
213 for it in membrane.mesh.points:
214     it.set_force((M_force,0.0,0.0))
215
216 out_file_sph = open(directory_data + "/sphere_" + str(sim_no) + ".
    dat", "a")
217 out_file_sph.write ("cycle time velX velY velZ posX posY posZ sizeX
    sizeY sizeZ defI")
218 out_file_sph.write("\n")
219 out_file_sph.close()
220
221 out_file_nuc = open(directory_data + "/nucleus_" + str(sim_no) + ".
    dat", "a")
222 out_file_nuc.write ("cycle time velX velY velZ posX posY posZ sizeX
    sizeY sizeZ defI")
223 out_file_nuc.write("\n")
224 out_file_nuc.close()
225
226 out_file_diff = open(directory_data + "/difference_" + str(sim_no)
    + ".dat", "a")
227 out_file_diff.write ("posX diffMaxX defI")
228 out_file_diff.write("\n")
```

PRÍLOHY

```
229 out_file_diff.close()
230
231 out_file = open(directory_data + "/bonds_info_" + str(sim_no) + ".
    dat", "a")
232 out_file.write ("cycle min_edge max_edge ave_edge \n")
233 out_file.close()
234
235 #####
236             #MAIN LOOP
237 #####
238
239 membrane.output_vtk_pos_folded(file_name=directory_vtk + "/"
    membrane_0.vtk")
240 nucleus.output_vtk_pos_folded(file_name=directory_vtk + "/nucleus_0
    .vtk")
241 oif.output_vtk_lines(lines=lines, out_file=directory_vtk + "/"
    lines_0.vtk" )
242
243
244 #for i in range(1,maxCycle):
245 while cycle < maxCycle:
246     system.integrator.run(steps=integr_steps)
247     # vtk file for membrane
248     membrane.output_vtk_pos_folded(file_name=directory_vtk + "/"
    membrane_" + str(cycle//integr_steps) + ".vtk")
249     # dat file for membrane
250     out_file_sph = open(directory_data + "/sphere_" + str(sim_no) +
    ".dat", "a")
251     sph_velocity = membrane.get_velocity()
252     sph_position = membrane.get_origin()
253     sph_pos_bounds = membrane.pos_bounds()
254     sph_x_size = sph_pos_bounds[1] - sph_pos_bounds[0]
255     sph_y_size = sph_pos_bounds[3] - sph_pos_bounds[2]
256     sph_z_size = sph_pos_bounds[5] - sph_pos_bounds[4]
257     # out_file_sph.write ("cycle time velX velY velZ posX posY posZ
    sizeX sizeY sizeZ")
258     out_file_sph.write(
259         str(cycle) + " " + str(cycle*time_step) + " " + str(
    sph_velocity) + " " + str(sph_position) + " " + str(
260         sph_x_size) + " " + str(sph_y_size) + " " + str(
    sph_z_size) + " " + str(sph_x_size/sph_y_size))
261     out_file_sph.write("\n")
```



```

262     out_file_sph.close()
263     # vtk file for nucleus
264     nucleus.output_vtk_pos_folded(file_name=directory_vtk + "/"
nucleus_" + str(cycle//integr_steps) + ".vtk")
265     # dat file for nucleus
266     out_file_nuc = open(directory_data + "/nucleus_" + str(sim_no)
+ ".dat", "a")
267     nuc_velocity = nucleus.get_velocity()
268     nuc_position = nucleus.get_origin()
269     nuc_pos_bounds = nucleus.pos_bounds()
270     nuc_x_size = nuc_pos_bounds[1] - nuc_pos_bounds[0]
271     nuc_y_size = nuc_pos_bounds[3] - nuc_pos_bounds[2]
272     nuc_z_size = nuc_pos_bounds[5] - nuc_pos_bounds[4]
273     out_file_nuc.write(
274         str(cycle) + " " + str(cycle*time_step) + " " + str(
nuc_velocity) + " " + str(nuc_position) + " " + str(
275         nuc_x_size) + " " + str(nuc_y_size) + " " + str(
nuc_z_size) + " " + str(nuc_x_size/nuc_y_size))
276     out_file_nuc.write("\n")
277     out_file_nuc.close()
278     lines = []
279     out_file_diff = open(directory_data + "/difference_" + str(
sim_no) + ".dat", "a")
280     out_file_diff.write(str(sph_position[0]) + " " + str(
sph_pos_bounds[1]-nuc_pos_bounds[1]) + " " + str(sph_x_size/
sph_y_size))
281     out_file_diff.write("\n")
282     out_file_diff.close()
283     for pair in pairs:
284         pPos = nucleus.mesh.points[pair[0]].get_pos()
285         pMembranePos = membrane.mesh.points[pair[1]].get_pos()
286         tmpConnection = pMembranePos - pPos
287         dlzka = oif.norm(tmpConnection)
288         average_dlzka = average_dlzka+dlzka
289         if dlzka > max_dlzka:
290             max_dlzka = dlzka
291         if dlzka < min_dlzka:
292             min_dlzka = dlzka
293         tmplist = [pPos[0], pPos[1], pPos[2], pMembranePos[0],
pMembranePos[1], pMembranePos[2]]
294         lines.append(tmplist)
295     average_dlzka = average_dlzka/countbonds

```

```
296     oif.output_vtk_lines(lines=lines, out_file=directory_vtk + "/"
297     lines_" + str(cycle//integr_steps) + ".vtk")
297     out_file = open(directory_data + "/bonds_info_" + str(sim_no) +
298     ".dat", "a")
298     out_file.write(str(cycle) + " " + str(min_dlzka) + " " + str(
299     max_dlzka) + " " + str(average_dlzka))
299     out_file.write("\n")
300     out_file.close()
301     if sph_pos_bounds[1] > 2 * boxX:
302         cycle = maxCycle
303     cycle = cycle + integr_steps
304     max_dlzka = - large_number
305     min_dlzka = large_number
306     average_dlzka = 0
307     # print "time: ", str(i*time_step*integr_steps)
```

B.3 Viazané interakcie - FENE

```
1 import espressomd
2 import object_in_fluid as oif
3
4 import Boundaries_15
5 from Boundaries_15 import *
6
7 from espressomd import lb
8 from espressomd import lbboundaries
9 from espressomd import shapes
10 from espressomd import interactions
11 from espressomd.interactions import FeneBond
12
13 import numpy as np
14 import os, sys
15 import random
16
17 # this is a script to try the nucleus with bonded interactions
18
19 #####
20 #INPUT DATA
21 #####
22 large_number = 10000000.0
23
24 # cell
```

```
25 membrane_radius = float(sys.argv[1])
26 membrane_noNodes = int(sys.argv[2])
27 membrane_ks = float(sys.argv[3])
28 membrane_kb = float(sys.argv[4])
29 membrane_kal = float(sys.argv[5])
30 membrane_kag = float(sys.argv[6])
31 membrane_kv = float(sys.argv[7])
32 membrane_kvisc = float(sys.argv[8])
33 membrane_mass = float(sys.argv[9])
34 # nucleus
35 nucleus_radius = float(sys.argv[10])
36 nucleus_noNodes = int(sys.argv[11])
37 nucleus_ks = float(sys.argv[12])
38 nucleus_kb = float(sys.argv[13])
39 nucleus_kal = float(sys.argv[14])
40 nucleus_kag = float(sys.argv[15])
41 nucleus_kv = float(sys.argv[16])
42 nucleus_kvisc = float(sys.argv[17])
43 nucleus_mass = float(sys.argv[18])
44 # fluid force
45 fluid_force = float(sys.argv[19])
46 # M_force is the force applied to each point of the cell (M as
    membrane)
47 M_force = float(sys.argv[20])
48 sim_no = sys.argv[21]
49 # fene bond parameter
50 par_k = float(sys.argv[22])
51 angle_par = float(sys.argv[23])
52 par_max = float(sys.argv[24])
53 # parameters of the constriction
54 par_ax = float(sys.argv[25])
55 par_ay = float(sys.argv[26])
56 par_az = float(sys.argv[27])
57
58 #potrebujes mat pricinok output a v nom priecinok data a vtk, do
    data sa ti ukladaju datove subory, do vtk sa ti ukladaju vtkacka
59 directory_data = "output/data/sim"+str(sim_no)
60 os.makedirs(directory_data)
61 directory_vtk = "output/vtk/sim"+str(sim_no)
62 os.makedirs(directory_vtk)
63 vtk_directory = directory_vtk+"/vtk"
64 os.makedirs(vtk_directory)
```

```
65
66 #CELL
67 originX = 3*membrane_radius/2
68 originY = boxY/2
69 originZ = boxZ/2
70 membrane_partMass = membrane_mass/membrane_noNodes
71 membrane_stretch = float(membrane_radius)
72 #NUCLEUS
73 nucleus_partMass = nucleus_mass/nucleus_noNodes
74 nucleus_stretch = float(nucleus_radius)
75
76 #INPUT FILES
77 membraneNodes = "input/sphere" + str(membrane_noNodes) + "nodes.dat"
78
79 membraneTriangles = "input/sphere" + str(membrane_noNodes) + "
80 triangles.dat"
81
82 nucleusNodes = "input/sphere" + str(nucleus_noNodes) + "nodes.dat"
83
84 nucleusTriangles = "input/sphere" + str(nucleus_noNodes) + "
85 triangles.dat"
86
87
88 #LBFLUID
89 LBgrid = 1.0
90 density = 1.025
91 viscosity = 1.5
92 time_step = 0.1
93
94 #FRICTION
95 # surface is the refence sphere with r = 4
96 surface = 4*np.pi*4**2
97 surface_sph = 4*np.pi*membrane_radius**2
98 friction = (393.0/membrane_noNodes)*np.sqrt(surface_sph/surface)
99
100 *((5.6-1.82)/(6.0/1.025-1.5)*(viscosity-1.5)+(10-1.82)/(6-1.025)
101 *(density-1.025)+1.82)
102 #print ("surface_sph:" + str(surface_sph))
103 #print ("friction_sph:" + str(friction))
104
105
106 system = espressomd.System(box_l=[boxX, boxY, boxZ])
107 system.cell_system.skin = 0.2
108 system.time_step = time_step
109
110 #ITERATION PARAMETERS
111 cycle = 0
```

```
102 maxCycle = 500000000000
103 integr_steps = 100
104
105 #####
106 #initialization
107 #####
108
109 # creating templates
110 membrane_type = oif.OifCellType(nodes_file=membraneNodes,
    triangles_file=membraneTriangles, check_orientation=False,
111     system=system, ks=membrane_ks, kb=
    membrane_kb, kal=membrane_kal, kag=membrane_kag,
112     kv=membrane_kv, kvisc=membrane_kvisc,
    resize=[membrane_stretch, membrane_stretch, membrane_stretch],
    normal=True)
113 nucleus_type = oif.OifCellType(nodes_file=nucleusNodes,
    triangles_file=nucleusTriangles, check_orientation=False,
114     system=system, ks=nucleus_ks, kb=
    nucleus_kb, kal=nucleus_kal, kag=nucleus_kag,
115     kv=nucleus_kv, kvisc=nucleus_kvisc,
    resize=[nucleus_stretch, nucleus_stretch, nucleus_stretch],
    normal=True)
116
117 # creating the cell
118 membrane = oif.OifCell(cell_type = membrane_type, particle_type=0,
    origin=[originX,originY,originZ], particle_mass=
    membrane_partMass)
119 nucleus = oif.OifCell(cell_type = nucleus_type, particle_type=1,
    origin=[originX,originY,originZ], particle_mass=nucleus_partMass
    )
120
121 #info about the cell membrane and nucleus membrane
122 surface_sph = membrane.mesh.surface()
123 min_hrana_sph = membrane.mesh.min_edge_length()
124 max_hrana_sph = membrane.mesh.max_edge_length()
125 aver_hrana_sph = membrane.mesh.aver_edge_length()
126 out_file = open(directory_data + "/sphere_info_" + str(sim_no) + ".
    dat", "a")
127 out_file.write ("surface min_edge max_edge ave_edge \n")
128 out_file.write(str(surface_sph)+" "+str(min_hrana_sph)+" "+str(
    max_hrana_sph)+" "+str(aver_hrana_sph))
129 out_file.close()
```

```
130
131 surface_nuc = nucleus.mesh.surface()
132 min_hrana_nuc = nucleus.mesh.min_edge_length()
133 max_hrana_nuc = nucleus.mesh.max_edge_length()
134 aver_hrana_nuc = nucleus.mesh.aver_edge_length()
135 out_file = open(directory_data + "/nucleus_info_" + str(sim_no) + ".dat", "a")
136 out_file.write("surface min_edge max_edge ave_edge \n")
137 out_file.write(str(surface_nuc)+" "+str(min_hrana_nuc)+" "+str(max_hrana_nuc)+" "+str(aver_hrana_nuc))
138 out_file.close()
139
140
141 small_angle = np.pi/angle_par
142 countbonds = 0
143 average_dlzka = 0
144 max_dlzka = - large_number
145 min_dlzka = large_number
146
147 lines = []
148 pairs = []
149 for p in nucleus.mesh.points:
150     neighbors = nucleus.mesh.neighbors[p.id]
151     normal = neighbors.outer_normal()
152     lengthNormal = oif.norm(normal)
153     pPos = p.get_pos()
154     for pMembrane in membrane.mesh.points:
155         pMembranePos = pMembrane.get_pos()
156         tmpConnection = pMembranePos - pPos
157         tmpDist = oif.norm(tmpConnection)
158         angle = np.arccos(np.dot(tmpConnection,normal)/(tmpDist * lengthNormal))
159         if angle < small_angle:
160             feneInter = FeneBond(k=par_k, d_r_max=par_max, r_0=tmpDist)
161             system.bonded_inter.add(feneInter)
162             p.part.add_bond((feneInter, pMembrane.part_id))
163             tmpList = [pPos[0], pPos[1], pPos[2], pMembranePos[0], pMembranePos[1], pMembranePos[2]]
164             tmppair = [p.id, pMembrane.id]
165             lines.append(tmpList)
166             pairs.append(tmppair)
```

```
167         countbonds += 1
168
169 #initial info about the bonds , number, min_length, max_length,
    average
170 for pair in pairs:
171     pPos = nucleus.mesh.points[pair[0]].get_pos()
172     pMembranePos = membrane.mesh.points[pair[1]].get_pos()
173     tmpConnection = pMembranePos - pPos
174     dlzka = oif.norm(tmpConnection)
175     average_dlzka = average_dlzka + dlzka
176     if dlzka > max_dlzka:
177         max_dlzka = dlzka
178     if dlzka < min_dlzka:
179         min_dlzka = dlzka
180 average_dlzka = average_dlzka / countbonds
181 out_file = open(directory_data + "/bonds_initial_info_" + str(
    sim_no) + ".dat", "a")
182 out_file.write("NoBonds min_edge max_edge ave_edge \n")
183 out_file.write(str(countbonds)+" "+str(min_dlzka)+" "+str(max_dlzka)
    )+" "+str(average_dlzka))
184 out_file.close()
185 average_dlzka = 0
186 max_dlzka = - large_number
187 min_dlzka = large_number
188
189
190 # cell-wall interactions
191 system.non_bonded_inter[0,10].soft_sphere.set_params(a = 0.0001, n
    = 1.2, cutoff = 0.1, offset = 0.0)
192 system.non_bonded_inter[1,10].soft_sphere.set_params(a = 0.0001, n
    = 1.2, cutoff = 0.1, offset = 0.0)
193
194
195 # fluid
196 #lbf = espressomd.lb.LBFluid(agrid = LBgrid, dens = density, visc =
    viscosity, tau = time_step, fric = friction , ext_force_density
    = [fluid_force, 0.0, 0.0])
197 #system.actors.add(lbf)
198
199 # fluid
200 lb_params = {'agrid': LBgrid, 'dens': density, 'visc': viscosity, '
    tau': time_step,
```

```
201         'ext_force_density': [fluid_force, 0, 0]}
202 lbf = espressomd.lb.LBFluid(**lb_params)
203 system.actors.add(lbf)
204
205 system.thermostat.set_lb(LB_fluid=lbf, gamma=friction)
206
207 boundaries = list()
208 FillBoundaries(boundaries, vtk_directory, par_ax, par_ay, par_az)
209 for boundary in boundaries:
210     system.lbboundaries.add(lbboundaries.LBBoundary(shape=boundary)
211                             )
212     system.constraints.add(shape=boundary, particle_type=10,
213                             penetrable=False)
214
215 #force on individual mesh points
216 for it in membrane.mesh.points:
217     it.set_force((M_force,0.0,0.0))
218
219 out_file_sph = open(directory_data + "/sphere_" + str(sim_no) + ".
220                   dat", "a")
221 out_file_sph.write ("cycle time velX velY velZ posX posY posZ sizeX
222                   sizeY sizeZ defI")
223 out_file_sph.write("\n")
224 out_file_sph.close()
225
226 out_file_nuc = open(directory_data + "/nucleus_" + str(sim_no) + ".
227                   dat", "a")
228 out_file_nuc.write ("cycle time velX velY velZ posX posY posZ sizeX
229                   sizeY sizeZ defI")
230 out_file_nuc.write("\n")
231 out_file_nuc.close()
232
233 out_file_diff = open(directory_data + "/difference_" + str(sim_no)
234                   + ".dat", "a")
235 out_file_diff.write ("posX diffMaxX defI")
236 out_file_diff.write("\n")
237 out_file_diff.close()
238
239 out_file = open(directory_data + "/bonds_info_" + str(sim_no) + ".
240                   dat", "a")
241 out_file.write ("cycle min_edge max_edge ave_edge \n")
242 out_file.close()
```



```
235
236 #####
237             #MAIN LOOP
238 #####
239
240 membrane.output_vtk_pos_folded(file_name=directory_vtk + "/"
    membrane_0.vtk")
241 nucleus.output_vtk_pos_folded(file_name=directory_vtk + "/nucleus_0
    .vtk")
242 oif.output_vtk_lines(lines=lines, out_file=directory_vtk + "/"
    lines_0.vtk )
243
244
245 #for i in range(1,maxCycle):
246 while cycle < maxCycle:
247     system.integrator.run(steps=integr_steps)
248     # vtk file for membrane
249     membrane.output_vtk_pos_folded(file_name=directory_vtk + "/"
    membrane_" + str(cycle//integr_steps) + ".vtk")
250     # dat file for membrane
251     out_file_sph = open(directory_data + "/sphere_" + str(sim_no) +
    ".dat", "a")
252     sph_velocity = membrane.get_velocity()
253     sph_position = membrane.get_origin()
254     sph_pos_bounds = membrane.pos_bounds()
255     sph_x_size = sph_pos_bounds[1] - sph_pos_bounds[0]
256     sph_y_size = sph_pos_bounds[3] - sph_pos_bounds[2]
257     sph_z_size = sph_pos_bounds[5] - sph_pos_bounds[4]
258     # out_file_sph.write ("cycle time velX velY velZ posX posY posZ
    sizeX sizeY sizeZ")
259     out_file_sph.write(
260         str(cycle) + " " + str(cycle*time_step) + " " + str(
    sph_velocity) + " " + str(sph_position) + " " + str(
261         sph_x_size) + " " + str(sph_y_size) + " " + str(
    sph_z_size) + " " + str(sph_x_size/sph_y_size))
262     out_file_sph.write("\n")
263     out_file_sph.close()
264     # vtk file for nucleus
265     nucleus.output_vtk_pos_folded(file_name=directory_vtk + "/"
    nucleus_" + str(cycle//integr_steps) + ".vtk")
266     # dat file for nucleus
267     out_file_nuc = open(directory_data + "/nucleus_" + str(sim_no)
```

```
+ ".dat", "a")
268     nuc_velocity = nucleus.get_velocity()
269     nuc_position = nucleus.get_origin()
270     nuc_pos_bounds = nucleus.pos_bounds()
271     nuc_x_size = nuc_pos_bounds[1] - nuc_pos_bounds[0]
272     nuc_y_size = nuc_pos_bounds[3] - nuc_pos_bounds[2]
273     nuc_z_size = nuc_pos_bounds[5] - nuc_pos_bounds[4]
274     out_file_nuc.write(
275         str(cycle) + " " + str(cycle*time_step) + " " + str(
nuc_velocity) + " " + str(nuc_position) + " " + str(
276         nuc_x_size) + " " + str(nuc_y_size) + " " + str(
nuc_z_size) + " " + str(nuc_x_size/nuc_y_size))
277     out_file_nuc.write("\n")
278     out_file_nuc.close()
279     lines = []
280     out_file_diff = open(directory_data + "/difference_" + str(
sim_no) + ".dat", "a")
281     out_file_diff.write(str(sph_position[0]) + " " + str(
sph_pos_bounds[1]-nuc_pos_bounds[1]) + " " + str(sph_x_size/
sph_y_size))
282     out_file_diff.write("\n")
283     out_file_diff.close()
284     for pair in pairs:
285         pPos = nucleus.mesh.points[pair[0]].get_pos()
286         pMembranePos = membrane.mesh.points[pair[1]].get_pos()
287         tmpConnection = pMembranePos - pPos
288         dlzka = oif.norm(tmpConnection)
289         average_dlzka = average_dlzka+dlzka
290         if dlzka > max_dlzka:
291             max_dlzka = dlzka
292         if dlzka < min_dlzka:
293             min_dlzka = dlzka
294         tmplist = [pPos[0], pPos[1], pPos[2], pMembranePos[0],
pMembranePos[1], pMembranePos[2]]
295         lines.append(tmplist)
296         average_dlzka = average_dlzka/countbonds
297         oif.output_vtk_lines(lines=lines, out_file=directory_vtk + "/"
lines_" + str(cycle//integr_steps) + ".vtk")
298         out_file = open(directory_data + "/bonds_info_" + str(sim_no) +
".dat", "a")
299         out_file.write(str(cycle) + " " + str(min_dlzka) + " " + str(
max_dlzka) + " " + str(average_dlzka))
```

```

300     out_file.write("\n")
301     out_file.close()
302     if sph_pos_bounds[1] > 2 * boxX:
303         cycle = maxCycle
304     cycle = cycle + integr_steps
305     max_dlzka = - large_number
306     min_dlzka = large_number
307     average_dlzka = 0
308     # print "time: ", str(i*time_step*integr_steps)

```

B.4 Viazané interakcie - kvadratický potenciál

```

1  import espressomd
2  import object_in_fluid as oif
3
4  import Boundaries_15
5  from Boundaries_15 import *
6
7  from espressomd import lb
8  from espressomd import lbboundaries
9  from espressomd import shapes
10 from espressomd import interactions
11 from espressomd.interactions import QuarticBond
12
13 import numpy as np
14 import os, sys
15 import random
16
17 # this is a script to try the nucleus with bonded interactions
18
19 #####
20 #INPUT DATA
21 #####
22 large_number = 10000000.0
23
24 # cell
25 membrane_radius = float(sys.argv[1])
26 membrane_noNodes = int(sys.argv[2])
27 membrane_ks = float(sys.argv[3])
28 membrane_kb = float(sys.argv[4])
29 membrane_kal = float(sys.argv[5])
30 membrane_kag = float(sys.argv[6])

```

PRÍLOHY

```
31 membrane_kv = float(sys.argv[7])
32 membrane_kvisc = float(sys.argv[8])
33 membrane_mass = float(sys.argv[9])
34 # nucleus
35 nucleus_radius = float(sys.argv[10])
36 nucleus_noNodes = int(sys.argv[11])
37 nucleus_ks = float(sys.argv[12])
38 nucleus_kb = float(sys.argv[13])
39 nucleus_kal = float(sys.argv[14])
40 nucleus_kag = float(sys.argv[15])
41 nucleus_kv = float(sys.argv[16])
42 nucleus_kvisc = float(sys.argv[17])
43 nucleus_mass = float(sys.argv[18])
44 # fluid force
45 fluid_force = float(sys.argv[19])
46 # M_force is the force applied to each point of the cell (M as
    membrane)
47 M_force = float(sys.argv[20])
48 sim_no = sys.argv[21]
49 # quadratic bond parameter
50 par_k0 = float(sys.argv[22])
51 angle_par = float(sys.argv[23])
52 par_k1 = float(sys.argv[24])
53 # parameters of the constriction
54 par_ax = float(sys.argv[25])
55 par_ay = float(sys.argv[26])
56 par_az = float(sys.argv[27])
57
58
59 #potrebujes mat pricinok output a v nom priecinok data a vtk, do
    data sa ti ukladaju datove subory, do vtk sa ti ukladaju vtkacka
60 directory_data = "output/data/sim"+str(sim_no)
61 os.makedirs(directory_data)
62 directory_vtk = "output/vtk/sim"+str(sim_no)
63 os.makedirs(directory_vtk)
64 vtk_directory = directory_vtk+"/vtk"
65 os.makedirs(vtk_directory)
66
67 #CELL
68 originX = 3*membrane_radius/2
69 originY = boxY/2
70 originZ = boxZ/2
```

```
71 membrane_partMass = membrane_mass/membrane_noNodes
72 membrane_stretch = float(membrane_radius)
73 #NUCLEUS
74 nucleus_partMass = nucleus_mass/nucleus_noNodes
75 nucleus_stretch = float(nucleus_radius)
76
77 #INPUT FILES
78 membraneNodes = "input/sphere" + str(membrane_noNodes) + "nodes.dat"
79
80 membraneTriangles = "input/sphere" + str(membrane_noNodes) + "
81 triangles.dat"
82
83 nucleusNodes = "input/sphere" + str(nucleus_noNodes) + "nodes.dat"
84
85 nucleusTriangles = "input/sphere" + str(nucleus_noNodes) + "
86 triangles.dat"
87
88
89 #LBFLUID
90 LBgrid = 1.0
91 density = 1.025
92 viscosity = 1.5
93 time_step = 0.1
94
95 #FRICTION
96 # surface is the refence sphere with r = 4
97 surface = 4*np.pi*4**2
98 surface_sph = 4*np.pi*membrane_radius**2
99 friction = (393.0/membrane_noNodes)*np.sqrt(surface_sph/surface)
100
101 *((5.6-1.82)/(6.0/1.025-1.5)*(viscosity-1.5)+(10-1.82)/(6-1.025)
102 *(density-1.025)+1.82)
103
104 #print ("surface_sph:" + str(surface_sph))
105 #print ("friction_sph:" + str(friction))
106
107
108 system = espressomd.System(box_l=[boxX, boxY, boxZ])
109 system.cell_system.skin = 0.2
110 system.time_step = time_step
111
112 #ITERATION PARAMETERS
113 cycle = 0
114 maxCycle = 500000000000
115 integr_steps = 100
116
117 #####
118 #initialization
```

PRÍLOHY

```
108 #####
109
110 # creating templates
111 membrane_type = oif.OifCellType(nodes_file=membraneNodes,
112     triangles_file=membraneTriangles, check_orientation=False,
113     system=system, ks=membrane_ks, kb=
114     membrane_kb, kal=membrane_kal, kag=membrane_kag,
115     kv=membrane_kv, kvisc=membrane_kvisc,
116     resize=[membrane_stretch, membrane_stretch, membrane_stretch],
117     normal=True)
118
119 nucleus_type = oif.OifCellType(nodes_file=nucleusNodes,
120     triangles_file=nucleusTriangles, check_orientation=False,
121     system=system, ks=nucleus_ks, kb=
122     nucleus_kb, kal=nucleus_kal, kag=nucleus_kag,
123     kv=nucleus_kv, kvisc=nucleus_kvisc,
124     resize=[nucleus_stretch, nucleus_stretch, nucleus_stretch],
125     normal=True)
126
127 # creating the cell
128 membrane = oif.OifCell(cell_type = membrane_type, particle_type=0,
129     origin=[originX,originY,originZ], particle_mass=
130     membrane_partMass)
131 nucleus = oif.OifCell(cell_type = nucleus_type, particle_type=1,
132     origin=[originX,originY,originZ], particle_mass=nucleus_partMass
133 )
134
135 #info about the cell membrane and nucleus membrane
136 surface_sph = membrane.mesh.surface()
137 min_hrana_sph = membrane.mesh.min_edge_length()
138 max_hrana_sph = membrane.mesh.max_edge_length()
139 aver_hrana_sph = membrane.mesh.aver_edge_length()
140 out_file = open(directory_data + "/sphere_info_" + str(sim_no) + ".
141     dat", "a")
142 out_file.write ("surface min_edge max_edge ave_edge \n")
143 out_file.write(str(surface_sph)+" "+str(min_hrana_sph)+" "+str(
144     max_hrana_sph)+" "+str(aver_hrana_sph))
145 out_file.close()
146
147 surface_nuc = nucleus.mesh.surface()
148 min_hrana_nuc = nucleus.mesh.min_edge_length()
149 max_hrana_nuc = nucleus.mesh.max_edge_length()
150 aver_hrana_nuc = nucleus.mesh.aver_edge_length()
```

```
136 out_file = open(directory_data + "/nucleus_info_" + str(sim_no) + ".dat", "a")
137 out_file.write("surface min_edge max_edge ave_edge \n")
138 out_file.write(str(surface_nuc)+" "+str(min_hrana_nuc)+" "+str(max_hrana_nuc)+" "+str(aver_hrana_nuc))
139 out_file.close()
140
141
142 small_angle = np.pi/angle_par
143 countbonds = 0
144 average_dlzka = 0
145 max_dlzka = - large_number
146 min_dlzka = large_number
147
148 lines = []
149 pairs = []
150 for p in nucleus.mesh.points:
151     neighbors = nucleus.mesh.neighbors[p.id]
152     normal = neighbors.outer_normal()
153     lengthNormal = oif.norm(normal)
154     pPos = p.get_pos()
155     for pMembrane in membrane.mesh.points:
156         pMembranePos = pMembrane.get_pos()
157         tmpConnection = pMembranePos - pPos
158         tmpDist = oif.norm(tmpConnection)
159         angle = np.arccos(np.dot(tmpConnection,normal)/(tmpDist * lengthNormal))
160         if angle < small_angle:
161             quarticInter = QuarticBond(k0=par_k0, k1=par_k1, r=tmpDist, r_cut=30.0)
162             system.bonded_inter.add(quarticInter)
163             p.part.add_bond((quarticInter, pMembrane.part_id))
164             tmplist = [pPos[0], pPos[1], pPos[2], pMembranePos[0], pMembranePos[1], pMembranePos[2]]
165             tmppair = [p.id, pMembrane.id]
166             lines.append(tmplist)
167             pairs.append(tmppair)
168             countbonds += 1
169
170 #initial info about the bonds , number, min_length, max_length, average
171 for pair in pairs:
```

PRÍLOHY

```
172     pPos = nucleus.mesh.points[pair[0]].get_pos()
173     pMembranePos = membrane.mesh.points[pair[1]].get_pos()
174     tmpConnection = pMembranePos - pPos
175     dlzka = oif.norm(tmpConnection)
176     average_dlzka = average_dlzka + dlzka
177     if dlzka > max_dlzka:
178         max_dlzka = dlzka
179     if dlzka < min_dlzka:
180         min_dlzka = dlzka
181 average_dlzka = average_dlzka / countbonds
182 out_file = open(directory_data + "/bonds_initial_info_" + str(
183     sim_no) + ".dat", "a")
184 out_file.write("NoBonds min_edge max_edge ave_edge \n")
185 out_file.write(str(countbonds)+" "+str(min_dlzka)+" "+str(max_dlzka)
186     )+" "+str(average_dlzka))
187 out_file.close()
188 average_dlzka = 0
189 max_dlzka = - large_number
190 min_dlzka = large_number
191
192 # cell-wall interactions
193 system.non_bonded_inter[0,10].soft_sphere.set_params(a = 0.0001, n
194     = 1.2, cutoff = 0.1, offset = 0.0)
195 system.non_bonded_inter[1,10].soft_sphere.set_params(a = 0.0001, n
196     = 1.2, cutoff = 0.1, offset = 0.0)
197
198 # fluid
199 #lbf = espressomd.lb.LBFluid(agrid = LBgrid, dens = density, visc =
200     viscosity, tau = time_step, fric = friction , ext_force_density
201     = [fluid_force, 0.0, 0.0])
202 #system.actors.add(lbf)
203
204 # fluid
205 lb_params = {'agrid': LBgrid, 'dens': density, 'visc': viscosity, '
206     tau': time_step,
207     'ext_force_density': [fluid_force, 0, 0]}
208 lbf = espressomd.lb.LBFluid(**lb_params)
209 system.actors.add(lbf)
210
211 system.thermostat.set_lb(LB_fluid=lbf, gamma=friction)
```



```
207
208 boundaries = list()
209 FillBoundaries(boundaries, vtk_directory, par_ax, par_ay, par_az)
210 for boundary in boundaries:
211     system.lbboundaries.add(lbboundaries.LBBoundary(shape=boundary)
212     )
213     system.constraints.add(shape=boundary, particle_type=10,
214     penetrable=False)
215
216 #force on individual mesh points
217 for it in membrane.mesh.points:
218     it.set_force((M_force,0.0,0.0))
219
220 out_file_sph = open(directory_data + "/sphere_" + str(sim_no) + ".
221     dat", "a")
222 out_file_sph.write ("cycle time velX velY velZ posX posY posZ sizeX
223     sizeY sizeZ defI")
224 out_file_sph.write("\n")
225 out_file_sph.close()
226
227 out_file_nuc = open(directory_data + "/nucleus_" + str(sim_no) + ".
228     dat", "a")
229 out_file_nuc.write ("cycle time velX velY velZ posX posY posZ sizeX
230     sizeY sizeZ defI")
231 out_file_nuc.write("\n")
232 out_file_nuc.close()
233
234 out_file_diff = open(directory_data + "/difference_" + str(sim_no)
235     + ".dat", "a")
236 out_file_diff.write ("posX diffMaxX defI")
237 out_file_diff.write("\n")
238 out_file_diff.close()
239
240 out_file = open(directory_data + "/bonds_info_" + str(sim_no) + ".
241     dat", "a")
242 out_file.write ("cycle min_edge max_edge ave_edge \n")
243 out_file.close()
244
245 #####
246 #MAIN LOOP
247 #####
```

```
241 membrane.output_vtk_pos_folded(file_name=directory_vtk + "/"
    membrane_0.vtk")
242 nucleus.output_vtk_pos_folded(file_name=directory_vtk + "/nucleus_0
    .vtk")
243 oif.output_vtk_lines(lines=lines, out_file=directory_vtk + "/"
    lines_0.vtk" )
244
245
246 #for i in range(1,maxCycle):
247 while cycle < maxCycle:
248     system.integrator.run(steps=integr_steps)
249     # vtk file for membrane
250     membrane.output_vtk_pos_folded(file_name=directory_vtk + "/"
    membrane_" + str(cycle//integr_steps) + ".vtk")
251     # dat file for membrane
252     out_file_sph = open(directory_data + "/sphere_" + str(sim_no) +
    ".dat", "a")
253     sph_velocity = membrane.get_velocity()
254     sph_position = membrane.get_origin()
255     sph_pos_bounds = membrane.pos_bounds()
256     sph_x_size = sph_pos_bounds[1] - sph_pos_bounds[0]
257     sph_y_size = sph_pos_bounds[3] - sph_pos_bounds[2]
258     sph_z_size = sph_pos_bounds[5] - sph_pos_bounds[4]
259     # out_file_sph.write ("cycle time velX velY velZ posX posY posZ
    sizeX sizeY sizeZ")
260     out_file_sph.write(
261         str(cycle) + " " + str(cycle*time_step) + " " + str(
    sph_velocity) + " " + str(sph_position) + " " + str(
262         sph_x_size) + " " + str(sph_y_size) + " " + str(
    sph_z_size) + " " + str(sph_x_size/sph_y_size))
263     out_file_sph.write("\n")
264     out_file_sph.close()
265     # vtk file for nucleus
266     nucleus.output_vtk_pos_folded(file_name=directory_vtk + "/"
    nucleus_" + str(cycle//integr_steps) + ".vtk")
267     # dat file for nucleus
268     out_file_nuc = open(directory_data + "/nucleus_" + str(sim_no)
    + ".dat", "a")
269     nuc_velocity = nucleus.get_velocity()
270     nuc_position = nucleus.get_origin()
271     nuc_pos_bounds = nucleus.pos_bounds()
272     nuc_x_size = nuc_pos_bounds[1] - nuc_pos_bounds[0]
```

```
273     nuc_y_size = nuc_pos_bounds[3] - nuc_pos_bounds[2]
274     nuc_z_size = nuc_pos_bounds[5] - nuc_pos_bounds[4]
275     out_file_nuc.write(
276         str(cycle) + " " + str(cycle*time_step) + " " + str(
nuc_velocity) + " " + str(nuc_position) + " " + str(
277         nuc_x_size) + " " + str(nuc_y_size) + " " + str(
nuc_z_size) + " " + str(nuc_x_size/nuc_y_size))
278     out_file_nuc.write("\n")
279     out_file_nuc.close()
280     lines = []
281     out_file_diff = open(directory_data + "/difference_" + str(
sim_no) + ".dat", "a")
282     out_file_diff.write(str(sph_position[0]) + " " + str(
sph_pos_bounds[1]-nuc_pos_bounds[1]) + " " + str(sph_x_size/
sph_y_size))
283     out_file_diff.write("\n")
284     out_file_diff.close()
285     for pair in pairs:
286         pPos = nucleus.mesh.points[pair[0]].get_pos()
287         pMembranePos = membrane.mesh.points[pair[1]].get_pos()
288         tmpConnection = pMembranePos - pPos
289         dlzka = oif.norm(tmpConnection)
290         average_dlzka = average_dlzka+dlzka
291         if dlzka > max_dlzka:
292             max_dlzka = dlzka
293         if dlzka < min_dlzka:
294             min_dlzka = dlzka
295         tmlist = [pPos[0], pPos[1], pPos[2], pMembranePos[0],
pMembranePos[1], pMembranePos[2]]
296         lines.append(tmlist)
297         average_dlzka = average_dlzka/countbonds
298         oif.output_vtk_lines(lines=lines, out_file=directory_vtk + "/"
lines_" + str(cycle//integr_steps) + ".vtk")
299         out_file = open(directory_data + "/bonds_info_" + str(sim_no) +
".dat", "a")
300         out_file.write(str(cycle) + " " + str(min_dlzka) + " " + str(
max_dlzka) + " " + str(average_dlzka))
301         out_file.write("\n")
302         out_file.close()
303         if sph_pos_bounds[1] > 2 * boxX:
304             cycle = maxCycle
305         cycle = cycle + integr_steps
```

```
306     max_dlzka = - large_number
307     min_dlzka = large_number
308     average_dlzka = 0
309     # print "time: ", str(i*time_step*integr_steps)
```

B.5 Neviazané interakcie

```
1 import espressomd
2 import object_in_fluid as oif
3
4 import Boundaries_15
5 from Boundaries_15 import *
6
7 from espressomd import lb
8 from espressomd import lbboundaries
9 from espressomd import shapes
10 from espressomd import interactions
11 #from espressomd.interactions import HarmonicBond
12
13 import numpy as np
14 import os, sys
15 #import random
16
17 # this is a script to try the nucleus with non-bonded interactions
18
19 #####
20 #INPUT DATA
21 #####
22 #large_number = 10000000.0
23
24 # cell
25 membrane_radius = float(sys.argv[1])
26 membrane_noNodes = int(sys.argv[2])
27 membrane_ks = float(sys.argv[3])
28 membrane_kb = float(sys.argv[4])
29 membrane_kal = float(sys.argv[5])
30 membrane_kag = float(sys.argv[6])
31 membrane_kv = float(sys.argv[7])
32 membrane_kvisc = float(sys.argv[8])
33 membrane_mass = float(sys.argv[9])
34 # nucleus
35 nucleus_radius = float(sys.argv[10])
```

```
36 nucleus_noNodes = int(sys.argv[11])
37 nucleus_ks = float(sys.argv[12])
38 nucleus_kb = float(sys.argv[13])
39 nucleus_kal = float(sys.argv[14])
40 nucleus_kag = float(sys.argv[15])
41 nucleus_kv = float(sys.argv[16])
42 nucleus_kvisc = float(sys.argv[17])
43 nucleus_mass = float(sys.argv[18])
44 # fluid force
45 fluid_force = float(sys.argv[19])
46 # M_force is the force applied to each point of the cell (M as
    membrane)
47 M_force = float(sys.argv[20])
48 sim_no = sys.argv[21]
49 # parameters for membrane / nucleus interaction
50 inter_a = float(sys.argv[22])
51 inter_n = float(sys.argv[23])
52 inter_cutoff = float(sys.argv[24])
53 # parameters of the constriction
54 par_ax = float(sys.argv[25])
55 par_ay = float(sys.argv[26])
56 par_az = float(sys.argv[27])
57
58 #potrebujes mat pricinok output a v nom priecinok data a vtk, do
    data sa ti ukladaju datove subory, do vtk sa ti ukladaju vtkacka
59 directory_data = "output/data/sim"+str(sim_no)
60 os.makedirs(directory_data)
61 directory_vtk = "output/vtk/sim"+str(sim_no)
62 os.makedirs(directory_vtk)
63 vtk_directory = directory_vtk+"/vtk"
64 os.makedirs(vtk_directory)
65
66 #CELL
67 originX = 3*membrane_radius/2
68 originY = boxY/2
69 originZ = boxZ/2
70 membrane_partMass = membrane_mass/membrane_noNodes
71 membrane_stretch = float(membrane_radius)
72 #NUCLEUS
73 nucleus_partMass = nucleus_mass/nucleus_noNodes
74 nucleus_stretch = float(nucleus_radius)
75
```

```
76 #INPUT FILES
77 membraneNodes = "input/sphere" + str(membrane_noNodes) + "nodes.dat"
78 membraneTriangles = "input/sphere" + str(membrane_noNodes) + "
    triangles.dat"
79 nucleusNodes = "input/sphere" + str(nucleus_noNodes) + "nodes.dat"
80 nucleusTriangles = "input/sphere" + str(nucleus_noNodes) + "
    triangles.dat"
81
82 #LBFLUID
83 LBgrid = 1.0
84 density = 1.025
85 viscosity = 1.5
86 time_step = 0.1
87
88 #FRICTION
89 # surface is the refence sphere with r = 4
90 surface = 4*np.pi*4**2
91 surface_sph = 4*np.pi*membrane_radius**2
92 friction = (393.0/membrane_noNodes)*np.sqrt(surface_sph/surface)
    *((5.6-1.82)/(6.0/1.025-1.5)*(viscosity-1.5)+(10-1.82)/(6-1.025)
    *(density-1.025)+1.82)
93 #print ("surface_sph:" + str(surface_sph))
94 #print ("friction_sph:" + str(friction))
95
96 system = espressomd.System(box_l=[boxX, boxY, boxZ])
97 system.cell_system.skin = 0.2
98 system.time_step = time_step
99
100 #ITERATION PARAMETERS
101 cycle = 0
102 maxCycle = 500000000000
103 integr_steps = 100
104
105 #####
106 #initialization
107 #####
108
109 # creating templates
110 membrane_type = oif.OifCellType(nodes_file=membraneNodes,
    triangles_file=membraneTriangles, check_orientation=False,
111 system=system, ks=membrane_ks, kb=
```

```
membrane_kb, kal=membrane_kal, kag=membrane_kag,
112         kv=membrane_kv, kvisc=membrane_kvisc,
resize=[membrane_stretch, membrane_stretch, membrane_stretch],
normal=True)
113 nucleus_type = oif.OifCellType(nodes_file=nucleusNodes,
triangles_file=nucleusTriangles, check_orientation=False,
114         system=system, ks=nucleus_ks, kb=
nucleus_kb, kal=nucleus_kal, kag=nucleus_kag,
115         kv=nucleus_kv, kvisc=nucleus_kvisc,
resize=[nucleus_stretch, nucleus_stretch, nucleus_stretch],
normal=True)
116
117 # creating the cell
118 membrane = oif.OifCell(cell_type = membrane_type, particle_type=0,
origin=[originX,originY,originZ], particle_mass=
membrane_partMass)
119 nucleus = oif.OifCell(cell_type = nucleus_type, particle_type=1,
origin=[originX,originY,originZ], particle_mass=nucleus_partMass
)
120 system.non_bonded_inter[0,1].soft_sphere.set_params(a = inter_a, n
= inter_n, cutoff = inter_cutoff, offset = 0.0)
121
122 #info about the cell membrane and nucleus membrane
123 surface_sph = membrane.mesh.surface()
124 min_hrana_sph = membrane.mesh.min_edge_length()
125 max_hrana_sph = membrane.mesh.max_edge_length()
126 aver_hrana_sph = membrane.mesh.aver_edge_length()
127 out_file = open(directory_data + "/sphere_info_" + str(sim_no) + ".
dat", "a")
128 out_file.write ("surface min_edge max_edge ave_edge \n")
129 out_file.write(str(surface_sph)+" "+str(min_hrana_sph)+" "+str(
max_hrana_sph)+" "+str(aver_hrana_sph))
130 out_file.close()
131
132 surface_nuc = nucleus.mesh.surface()
133 min_hrana_nuc = nucleus.mesh.min_edge_length()
134 max_hrana_nuc = nucleus.mesh.max_edge_length()
135 aver_hrana_nuc = nucleus.mesh.aver_edge_length()
136 out_file = open(directory_data + "/nucleus_info_" + str(sim_no) + "
.dat", "a")
137 out_file.write ("surface min_edge max_edge ave_edge \n")
138 out_file.write(str(surface_nuc)+" "+str(min_hrana_nuc)+" "+str(
```

```
max_hrana_nuc)+" "+str(aver_hrana_nuc))
139 out_file.close()
140
141 # cell-wall interactions
142 system.non_bonded_inter[0,10].soft_sphere.set_params(a = 0.0001, n
    = 1.2, cutoff = 0.1, offset = 0.0)
143 system.non_bonded_inter[1,10].soft_sphere.set_params(a = 0.0001, n
    = 1.2, cutoff = 0.1, offset = 0.0)
144
145 # fluid
146 #lbf = espressomd.lb.LBFluid(agrid = LBgrid, dens = density, visc =
    viscosity, tau = time_step, fric = friction , ext_force_density
    = [fluid_force, 0.0, 0.0])
147 #system.actors.add(lbf)
148
149 # fluid
150 lb_params = {'agrid': LBgrid, 'dens': density, 'visc': viscosity, '
    tau': time_step,
151              'ext_force_density': [fluid_force, 0, 0]}
152 lbf = espressomd.lb.LBFluid(**lb_params)
153 system.actors.add(lbf)
154
155 system.thermostat.set_lb(LB_fluid=lbf, gamma=friction)
156
157 boundaries = list()
158 FillBoundaries(boundaries, vtk_directory, par_ax, par_ay, par_az)
159 for boundary in boundaries:
160     system.lbboundaries.add(lbboundaries.LBBoundary(shape=boundary)
    )
161     system.constraints.add(shape=boundary, particle_type=10,
    penetrable=False)
162
163 #force on individual mesh points
164 for it in membrane.mesh.points:
165     it.set_force((M_force,0.0,0.0))
166
167 out_file_sph = open(directory_data + "/sphere_" + str(sim_no) + ".
    dat", "a")
168 out_file_sph.write ("cycle time velX velY velZ posX posY posZ sizeX
    sizeY sizeZ defI")
169 out_file_sph.write("\n")
170 out_file_sph.close()
```



```
171
172 out_file_nuc = open(directory_data + "/nucleus_" + str(sim_no) + ".
    dat", "a")
173 out_file_nuc.write ("cycle time velX velY velZ posX posY posZ sizeX
    sizeY sizeZ defI")
174 out_file_nuc.write("\n")
175 out_file_nuc.close()
176
177 out_file_diff = open(directory_data + "/difference_" + str(sim_no)
    + ".dat", "a")
178 out_file_diff.write ("posX diffMaxX defI")
179 out_file_diff.write("\n")
180 out_file_diff.close()
181
182 #####
183             #MAIN LOOP
184 #####
185
186 membrane.output_vtk_pos_folded(file_name=directory_vtk + "/"
    membrane_0.vtk")
187 nucleus.output_vtk_pos_folded(file_name=directory_vtk + "/nucleus_0
    .vtk")
188
189 #for i in range(1,maxCycle):
190 while cycle < maxCycle:
191     system.integrator.run(steps=integr_steps)
192     # vtk file for membrane
193     membrane.output_vtk_pos_folded(file_name=directory_vtk + "/"
    membrane_" + str(cycle//integr_steps) + ".vtk")
194     # dat file for membrane
195     out_file_sph = open(directory_data + "/sphere_" + str(sim_no) +
    ".dat", "a")
196     sph_velocity = membrane.get_velocity()
197     sph_position = membrane.get_origin()
198     sph_pos_bounds = membrane.pos_bounds()
199     sph_x_size = sph_pos_bounds[1] - sph_pos_bounds[0]
200     sph_y_size = sph_pos_bounds[3] - sph_pos_bounds[2]
201     sph_z_size = sph_pos_bounds[5] - sph_pos_bounds[4]
202     # out_file_sph.write ("cycle time velX velY velZ posX posY posZ
    sizeX sizeY sizeZ")
203     out_file_sph.write(
204         str(cycle) + " " + str(cycle*time_step) + " " + str(
```

```

sph_velocity) + " " + str(sph_position) + " " + str(
205     sph_x_size) + " " + str(sph_y_size) + " " + str(
sph_z_size) + " " + str(sph_x_size/sph_y_size))
206     out_file_sph.write("\n")
207     out_file_sph.close()
208     # vtk file for nucleus
209     nucleus.output_vtk_pos_folded(file_name=directory_vtk + "/"
nucleus_" + str(cycle//integr_steps) + ".vtk")
210     # dat file for nucleus
211     out_file_nuc = open(directory_data + "/nucleus_" + str(sim_no)
+ ".dat", "a")
212     nuc_velocity = nucleus.get_velocity()
213     nuc_position = nucleus.get_origin()
214     nuc_pos_bounds = nucleus.pos_bounds()
215     nuc_x_size = nuc_pos_bounds[1] - nuc_pos_bounds[0]
216     nuc_y_size = nuc_pos_bounds[3] - nuc_pos_bounds[2]
217     nuc_z_size = nuc_pos_bounds[5] - nuc_pos_bounds[4]
218     out_file_nuc.write(
219         str(cycle) + " " + str(cycle*time_step) + " " + str(
nuc_velocity) + " " + str(nuc_position) + " " + str(
220         nuc_x_size) + " " + str(nuc_y_size) + " " + str(
nuc_z_size) + " " + str(nuc_x_size/nuc_y_size))
221     out_file_nuc.write("\n")
222     out_file_nuc.close()
223     lines = []
224     out_file_diff = open(directory_data + "/difference_" + str(
sim_no) + ".dat", "a")
225     out_file_diff.write(str(sph_position[0])+ " " + str(
sph_pos_bounds[1]-nuc_pos_bounds[1]) + " " + str(sph_x_size/
sph_y_size))
226     out_file_diff.write("\n")
227     out_file_diff.close()
228     if sph_pos_bounds[1] > 2 * boxX:
229         cycle = maxCycle
230     cycle = cycle + integr_steps
231     # print "time: ", str(i*time_step*integr_steps)

```

B.6 Elastická guľa v zúženom kanáliku

```

1 import espressomd
2 import object_in_fluid as oif
3

```

```
4 import Boundaries_15
5 from Boundaries_15 import *
6
7 from espressomd import lb
8 from espressomd import lbboundaries
9 from espressomd import shapes
10 from espressomd import interactions
11 from espressomd.interactions import HarmonicBond
12
13 import numpy as np
14 import os, sys
15 import random
16
17 # this is a script to try the nucleus with bonded interactions
18
19 #####
20 #INPUT DATA
21 #####
22 large_number = 10000000.0
23
24 # cell
25 membrane_radius = float(sys.argv[1])
26 membrane_noNodes = int(sys.argv[2])
27 membrane_ks = float(sys.argv[3])
28 membrane_kb = float(sys.argv[4])
29 membrane_kal = float(sys.argv[5])
30 membrane_kag = float(sys.argv[6])
31 membrane_kv = float(sys.argv[7])
32 membrane_kvisc = float(sys.argv[8])
33 membrane_mass = float(sys.argv[9])
34 # fluid force
35 fluid_force = float(sys.argv[10])
36 # M_force is the force applied to each point of the cell (M as
    membrane)
37 M_force = float(sys.argv[11])
38 sim_no = sys.argv[12]
39 # self cell interaction parameter
40 par_a = float(sys.argv[13])
41 par_n = float(sys.argv[14])
42 cut = float(sys.argv[15])
43 # parameters of the constriction
44 par_ax = float(sys.argv[16])
```

PRÍLOHY

```
45 par_ay = float(sys.argv[17])
46 par_az = float(sys.argv[18])
47
48 #potrebujes mat pricinok output a v nom priecinok data a vtk, do
    data sa ti ukladaju datove subory, do vtk sa ti ukladaju vtkacka
49 directory_data = "output/data/sim"+str(sim_no)
50 os.makedirs(directory_data)
51 directory_vtk = "output/vtk/sim"+str(sim_no)
52 os.makedirs(directory_vtk)
53 vtk_directory = directory_vtk+"/vtk"
54 os.makedirs(vtk_directory)
55
56 #CELL
57 originX = 3*membrane_radius/2
58 originY = boxY/2
59 originZ = boxZ/2
60 membrane_partMass = membrane_mass/membrane_noNodes
61 membrane_stretch = float(membrane_radius)
62
63 #INPUT FILES
64 membraneNodes = "input/sphere" + str(membrane_noNodes) + "nodes.dat"
    "
65 membraneTriangles = "input/sphere" + str(membrane_noNodes) + "
    triangles.dat"
66
67 #LBFLUID
68 LBgrid = 1.0
69 density = 1.025
70 viscosity = 1.5
71 time_step = 0.1
72
73 #FRICTION
74 # surface is the refence sphere with r = 4
75 surface = 4*np.pi*4**2
76 surface_sph = 4*np.pi*membrane_radius**2
77 friction = (393.0/membrane_noNodes)*np.sqrt(surface_sph/surface)
    *((5.6-1.82)/(6.0/1.025-1.5)*(viscosity-1.5)+(10-1.82)/(6-1.025)
    *(density-1.025)+1.82)
78 #print ("surface_sph:" + str(surface_sph))
79 #print ("friction_sph:" + str(friction))
80
81 system = espressomd.System(box_l=[boxX, boxY, boxZ])
```

```
82 system.cell_system.skin = 0.2
83 system.time_step = time_step
84
85 #ITERATION PARAMETERS
86 cycle = 0
87 maxCycle = 500000000000
88 integr_steps = 100
89
90 #####
91 #initialization
92 #####
93
94 # creating templates
95 membrane_type = oif.OifCellType(nodes_file=membraneNodes,
96     triangles_file=membraneTriangles, check_orientation=False,
97     system=system, ks=membrane_ks, kb=
98     membrane_kb, kal=membrane_kal, kag=membrane_kag,
99     kv=membrane_kv, kvisc=membrane_kvisc,
100     resize=[membrane_stretch, membrane_stretch, membrane_stretch],
101     normal=True)
102
103 # creating the cell
104 membrane = oif.OifCell(cell_type = membrane_type, particle_type=0,
105     origin=[originX,originY,originZ], particle_mass=
106     membrane_partMass)
107
108 #info about the cell membrane
109 surface_sph = membrane.mesh.surface()
110 min_hrana_sph = membrane.mesh.min_edge_length()
111 max_hrana_sph = membrane.mesh.max_edge_length()
112 aver_hrana_sph = membrane.mesh.aver_edge_length()
113 out_file = open(directory_data + "/sphere_info_" + str(sim_no) + ".
114     dat", "a")
115 out_file.write ("surface min_edge max_edge ave_edge \n")
116 out_file.write(str(surface_sph)+" "+str(min_hrana_sph)+" "+str(
117     max_hrana_sph)+" "+str(aver_hrana_sph))
118 out_file.close()
119
120 # cell-wall interactions
121 system.non_bonded_inter[0,10].soft_sphere.set_params(a = 0.0001, n
122     = 1.2, cutoff = 0.1, offset = 0.0)
```

PRÍLOHY

```
115 # self-cell interactions
116 system.non_bonded_inter[0,0].soft_sphere.set_params(a = par_a, n =
    par_n, cutoff = cut, offset = 0.0)
117
118 # fluid
119 lb_params = {'agrid': LBgrid, 'dens': density, 'visc': viscosity, '
    tau': time_step,
120             'ext_force_density': [fluid_force, 0, 0]}
121 lbf = espressomd.lb.LBFluid(**lb_params)
122 system.actors.add(lbf)
123
124 system.thermostat.set_lb(LB_fluid=lbf, gamma=friction)
125
126 boundaries = list()
127 FillBoundaries(boundaries, vtk_directory, par_ax, par_ay, par_az)
128 for boundary in boundaries:
129     system.lbboundaries.add(lbboundaries.LBBoundary(shape=boundary)
    )
130     system.constraints.add(shape=boundary, particle_type=10,
    penetrable=False)
131
132 #force on individual mesh points
133 for it in membrane.mesh.points:
134     it.set_force((M_force,0.0,0.0))
135
136 out_file_sph = open(directory_data + "/sphere_" + str(sim_no) + ".
    dat", "a")
137 out_file_sph.write ("cycle time velX velY velZ posX posY posZ sizeX
    sizeY sizeZ defI")
138 out_file_sph.write("\n")
139 out_file_sph.close()
140
141 #####
142             #MAIN LOOP
143 #####
144
145 membrane.output_vtk_pos_folded(file_name=directory_vtk + "/"
    membrane_0.vtk")
146
147 #for i in range(1,maxCycle):
148 while cycle < maxCycle:
149     system.integrator.run(steps=integr_steps)
```

```

150     # vtk file for membrane
151     membrane.output_vtk_pos_folded(file_name=directory_vtk + "/"
membrane_" + str(cycle//integr_steps) + ".vtk")
152     # dat file for membrane
153     out_file_sph = open(directory_data + "/sphere_" + str(sim_no) +
".dat", "a")
154     sph_velocity = membrane.get_velocity()
155     sph_position = membrane.get_origin()
156     sph_pos_bounds = membrane.pos_bounds()
157     sph_x_size = sph_pos_bounds[1] - sph_pos_bounds[0]
158     sph_y_size = sph_pos_bounds[3] - sph_pos_bounds[2]
159     sph_z_size = sph_pos_bounds[5] - sph_pos_bounds[4]
160     # out_file_sph.write ("cycle time velX velY velZ posX posY posZ
sizeX sizeY sizeZ")
161     out_file_sph.write(
162         str(cycle) + " " + str(cycle*time_step) + " " + str(
sph_velocity) + " " + str(sph_position) + " " + str(
163         sph_x_size) + " " + str(sph_y_size) + " " + str(
sph_z_size) + " " + str(sph_x_size/sph_y_size))
164     out_file_sph.write("\n")
165     out_file_sph.close()
166     if sph_pos_bounds[1] > 2 * boxX:
167         cycle = maxCycle
168     cycle = cycle + integr_steps
169     # print "time: ", str(i*time_step*integr_steps)

```


C Použité skratky

AFM	A tomic F orce M icroscopy atómová silová mikroskopia
CGMD	C oarse- G ained M olecular D ynamics technika modelovania založená na zoskupovaní atómov do väčších molekúl
CPU	C entral P rocessing U nit procesor základnej jednotky
CTC	C irculating T umor C ell cirkulujúca rakovinová bunka / bunky
DI	D eformation I ndex deformačný index
DPD	D issipative P article D ynamics stochastická simulačná technika na simuláciu dynamických a reologických vlastností jednoduchých a komplexných tekutín
ESPResSo	E xtensible S imulation P ackage for R esearch on S oft M atter software určený pre modelovanie molekulárnej dynamiky
FEM	F inite E lement M ethod metóda konečných prvkov
FENE	F inite E xtention N onlinear E lastic ohraničená nelineárna elastická väzba
FPS	F rames P er S econd obrazová frekvencia
GPU	G raphics P rocessing U nit grafický procesor

HDD	H ard D isk D rive pevný disk
IBM	I mmersed B oundary M ethod metóda vnorených hraníc
IFEM	I mmersed F inite E lement M ethod vnorená metóda konečných prvkov
LBM	L attice B oltzmann M ethod lattice-Boltzmannova metóda
MB	M esh B ased models modely používajúce siete pre diskretizáciu povrchu objektov
PB	P article B ased models modely používajúce molekuly pre diskretizáciu objektov
PyOIF	P ython O bject I n F luid Pythonovský modul slúžiaci na modelovanie elastických objektov
RAM	R andom A ccess M emory pamäť s priamym prístupom
RBC	R ed B lood C ells červené krvinky
SPH	S moothed P article H ydrodynamics hydrodynamika hladkých častíc
SSD	S olid S tate D rive mechanika s nepohyblivým médiom
VOF	V olume O f F luid metóda konečného objemu
WBC	W hite B lood C ells biele krvinky
WHO	W orld H ealth O rganization svetová zdravotnícka organizácia