VIZUALIZACIJA I PRIPREMA SIMULACIJA ZA DUALSPHYSICS RJEŠAVAČ

Grgić, Daniel

Master's thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:190:930825

Rights / Prava: Attribution 4.0 International/Imenovanje 4.0 međunarodna

Download date / Datum preuzimanja: **2024-05-18**

Repository / Repozitorij:

Repository of the University of Rijeka, Faculty of Engineering

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij strojarstva

Diplomski rad

VIZUALIZACIJA I PRIPREMA SIMULACIJA ZA DUALSPHYSICS RJEŠAVAČ

Rijeka, rujan 2023.

Daniel Grgić

0152209100

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij strojarstva

Diplomski rad

VIZUALIZACIJA I PRIPREMA SIMULACIJA ZA DUALSPHYSICS RJEŠAVAČ

Mentor: Prof. dr. sc. Zoran Čarija

Rijeka, rujan 2023.

Daniel Grgić

0152209100

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA DIPLOMSKE ISPITE

Rijeka, 14. ožujka 2023.

Zavod: **Zavod za mehaniku fluida i računarsko inženjerstvo**
Predmet: **Vizualizacija i priprema računalnih simulacija**
Grana: **2.15.04 mehanika fluida**

ZADATAK ZA DIPLOMSKI RAD

Pristupnik: **Daniel Grgić (0152209100)**
Studij: Sveučilišni diplomski studij strojarstva
Modul: Računarska mehanika i inženjerstvo

Zadatak: **Vizualizacija i priprema simulacija za DualSPHysics rješavač / Visualization and Preparation of Computer Simulations for DualSPHysics solver**

Opis zadatka:

U sklopu rada dati će se teoretski pregled metode hidrodinamike izglađenih čestica rješavača (engl. Smoothed Particle Hydrodynamics, SPH) te opis rješavača DualSPHysics. Kreirat će se grafičko sučelje za pripremu simulacija za DualSPHysics rješavač koje će omogućiti korisniku jednostavnije zadavanje postavki numeričkih simulacija i napravit će se vizualizacija rješenja dobivenih simulacijom.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Daniel Grgić

Zadatak uručen pristupniku: 13. ožujka 2023.

Mentor:

Prof. dr. sc. Zoran Čarija

Predsjednik povjerenstva za diplomski ispit:

Izv. prof. dr. sc. Igor Bonefačić

IZJAVA

kojom ja, Daniel Grgić, student Tehničkog fakulteta u Rijeci, kao autor završnog rada pod naslovom „Vizualizacija i priprema simulacija za DualSPHysics rješavač“ izjavljujem da sam završni rad izradio samostalno, oslanjanjem na radove navedene u popisu literature.

Potpis studenta: _____

SADRŽAJ

1. UVOD 1
2. SPH metoda 2
2.1. Lagrangeova mreža 3
2.2 Općenito o SPH metodi i njene mogućnosti 4
3. OPĆENITO O DUALSPHYSICS 6
3.1. Sofveri i njihove instalacije 6
3.1.1. DualSPHysics 6
3.1.2. Visual Studio Code 7
3.1.3. XML 7
3.1.4. Python 8
3.1.5. FreeCAD 8
3.1.6. Spyder 9
3.2. Pregledavanje mogućnosti i tipova zadataka koje nudi DualSPHysics 9
3.2.1. Glavni primjeri 9
3.2.2. Chrono primjeri 10
3.2.3. Moordyin primjeri 10
3.2.4. Spajanje valova primjer 10
3.2.5. Inlet outlet primjeri 11
3.2.6. Modificirani dinamički granični uvijeti 11
3.2.7. Višefazni tekući plin 11
3.2.8. Višefazno strujanje Ne-Newtonovih fluida 12
3.3. Pokretanje DualSPHysics 12
3.3.1. Struktura datoteke 12
3.3.2. Pregled tijeka rada DualSPHysics 13
3.3.3. Pretprocesiranje 14
3.3.4. Postupak procesiranja 15
3.3.5. Postupak poslije procesiranja 16
3.4. Kompajliranje u DualSPHysics 18
4. Općenito o FreeCAD-u 19
4.1. Postavljanje DualSPHysics u FreeCAD 20
5. Opis problema 21
5.1. Pregled primjera strujanja fluida oko cijevi 21

5.1.1 Opis primjera strujanja fluida oko cijevi 21
5.1.2 Korišteni softveri za rješavanje problema 22
5.2 Mogućnost automatizacije 22
6. Metodologija 24
6.1 Objašnjenje XML datoteke 24
6.1.1 Konstantni parametri 24
6.1.2 Izrada geometrije 25
6.1.3 Definiranje pokreta i brzine 27
6.1.4 Definicija osnovnih parametara za ulaz i izlaz 28
6.1.5 Definicija ostalih parametara 30
6.2 Mogućnost automatizacije pomoću pythona 30
7. Rezultati 36
7.1 Laminarno strujanje fluida oko cilindra kružnog poprečnog presjeka 36
7.2 Turbulentno strujanje fluida oko cilindra kvadratnog poprečnog presjeka 44
7.3 Prednosti i mane FreeCAD softvera 50
8. Zaključak 51
9. LITERATURA 52
SAŽETAK i
Abstract ii

1. UVOD

Mehanika fluida je grana fizike koja se bavi gibanjima i silama koje djeluju na fluid. Mehanika fluida se dijeli na statiku fluida koja proučava zakone fluida u stanju mirovanja. Kinematika fluida se bavi zakonima gibanja fluida te dinamika fluida se bavi silama koje se pojavljuju na fluidu i gibanjima koje nastaju zbog djelovanje te sile te interakcijom između čvrstih tijela i fluida. Kinematika fluida opisuje ponašanje fluida u vremenu i prostoru, te ne razmatra uzroke gibanja i sile koje djeluju u fluidu. Neke od sila koje mogu djelovati na fluid su gravitacijske, tlačne, smične, inercijske i mnoge druge te glavni cilj mehanike fluida je određivanje tih sila.

Kako bi uspješno savladali probleme koji nastaju prilikom rada i analiziranja fluida, razvijaju se razne numeričke metode. Razvojem računala u zadnjih 20-ak godina paralelno se razvijaju i metode koje olakšavaju razne proračune i simulacije na fluidima. Cijeli razlog korištenja numeričkih metoda je ta što je puno brža i na kraju krajeva jeftinija nego da se rade eksperimentalne metode koje mogu biti izrazito skupe. Navedene metode ne daju u potpunosti točna rješenja no mogu približno prikazati kako će se fluid ponašati.

Strujanju fluida moguće je pristupiti na dva načina pa se stoga razvijaju dvije metode u računalnoj dinamici fluida. Prva metoda je Eulerov pristup (metoda kontrolnih volumena, metoda konačnih razlika) koji je danas puno istraženiji i prilagođeniji rješavanju raznih problema. Drugi pristup je Lagrangeov pristup gdje je koordinatni sustav pomičan, tj. giba se s česticom fluida, stoga se može reći da je ovaj pristup puno prikladniji za promatranje kinematike fluida.

U ovom radu će se ukratko opisati i koristiti metoda kontrolne mase koja je još uvijek slabo korištena i rasprostranjena u području računalnih simulacija strujanja fluida. Metoda kontrolne mase se temelji na Lagrangeovom pristupu čiji se protok fluida sastoji od velikog broja čestica gdje će se proučavati svojstva, smjer gibanja i brzine pojedinačne čestice. Jedan od rješavača koji će pomoći pri korištenju ove metode je DualSPHysics. DualSPHysics ne zahtijeva izradu numeričke mreže, što je i glavna karakteristika ove metode.

Ciljevi ovog rada su ispitati mogućnosti vizualizacije i mogućnosti automatizacije rješavača. Prvi dio rada će sadržavati teorijski prikaz temeljnih pojmova dok će u drugom djelu rada biti omogućen razgled rješenja i mogućnosti automatizacije DualSPHysics rješavača.

2. SPH METODA

Za rješavanje problema u inženjerstvu i znanosti, danas je neizbježno koristiti računala za numeričke simulacije koje pomažu u izradi simulacija i proračuna. Prve numeričke metode su temeljene na kreiranju domene koja je diskretizirana u konačni skup kontrolnih volumena ili ćelija. Diskretizirana domena naziva se numerička mreža. Numeričke metode temeljene na mreži, danas su naširoko u primjeni u različitim područjima računalne dinamike fluida i računalne mehanike čvrstog tijela.[11]

Za rješavanje problema u mehanici fluida, koriste se brojne metode, no najpopularnije su:

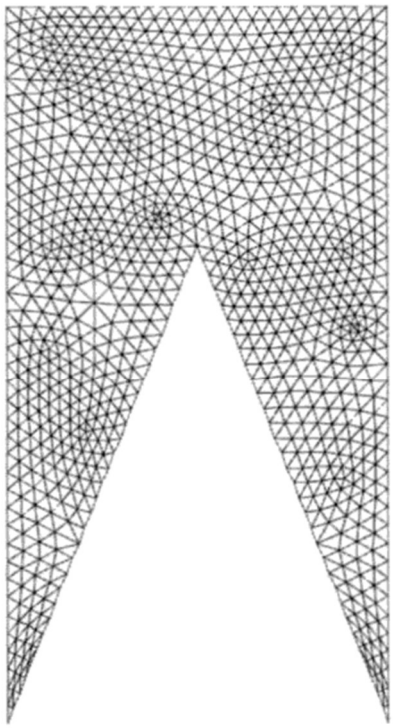
- Metoda konačnih razlika (eng. *Finite difference method*)
- Metoda konačnih elemenata (eng. *Finite element method*)
- Spektralne metode (eng. *Spectral method*)
- Metoda konačnih volumena (eng. Finite volume methods)

Unatoč uspjesima koje su postigle prethodno navedene metode, sve imaju jedan zajednički problem, a to su poteškoće u radu sa složenim mrežama. Kako se cijela formulacija temelji na mreži potreban je dugotrajan i skup proces generiranja kvalitetne mreže. No i postizanjem najkvalitetnije mreže, doći će do poteškoća u rješavanju problema sa slobodnim površinama, deformabilnom granicom i sličnim problemima. Stoga se u zadnjih par godina veliki broj znanstvenika usmjerio na računalne metode nove generacije u kojima se ne koristi mreža, za koje se očekuje da će biti superiornije u odnosu na metode s mrežama.[11]

U ovom radu usredotočit će se na metodu hidrodinamike izglađenih čestica, skraćeno SPH (eng. *Smoothed particle hydrodynamics*). Razlog zbog čega se i posvećuje ovoj metodi je ta što se SPH metoda može nositi s problemima slobodnih površina, deformabilnih granica, pomičnih sučelja te nošenje s problemima kod iznimno velikih deformacija. Ona je ujedno jedna od najstarijih metoda bez upotreba mreže te se približava svojoj zreloj fazi. Uz konstantna poboljšanja dostigla je prihvatljivu razinu za praktične inženjerske primjene. Koristi se za sve raspone od mikro do makro skale pa čak i do astronomske skale. Osim toga SPH procesor je uključen u brojne softverske pakete što donosi još jedan razlog zašto se za izradu simulacija koristi ovaj rješavač.[11]

2.1. Lagrangeova mreža

SPH koristi lagrangeovu metodu koja funkcionira na način da dijeli tekućinu u skup diskretnih pokretnih elemenata koji se nazivaju čestice ili ćelije. U lagrangeovom opisu, za pojedinačne čestice fluida, brzina, položaj i tako dalje je opisan funkcijom vremena. Slika 2.1 prikazuje primjer lagrangeove mreže koja je vezana uz određene čestice i prati njihovo stanje kroz cijeli računski proces.[11]



Slika 2.1 Lagrangeova mreža/ćelije pripremljena za simulaciju. Trokutaste ćelije i cijela mreža stanica se pomiče s materijalom[11]

Budući da svaki čvor mreže slijedi putanju materijala u točki mreže, to relativno kretanje spojnih čvorova može rezultirati širenjem, kompresijom i deformacijom mrežne ćelije. Kada se materijal deformira, mreža se u skladu s tim deformira.[11]

Glavne prednosti metoda temeljene na ovoj mreži su:

- Kod je puno brži nego kod metoda koje koriste mrežu
- Kako je mreža fiksirana na materijal, lako se može pratiti i dobiti cijela vremenska povijest svih varijabli polja u materijalnoj točki

- U proračunu neke čvorove mreže se mogu se postaviti duž granice i sučelja materijala. Rubni uvjet na slobodnim površinama se automatski nameće, prati i određuje jednostavno kretanje ovih čvorova mreže
- Nepravilne i komplicirane geometrije mogu se jednostavno tretirati upotrebom nepravilne mreže
- Kako nisu potrebne dodatne mreže izvan problemske domene, metode temeljene na ovoj mreži su puno računalno učinkovitija.[11]

Jedan od nedostataka ove metode koja primjenjuje ovu mrežu je ta što je teško primjenjiva na jako iskrivljenim geometrijama, kada je previše iskrivljena točnost formulacije postaje ugrožena. Osim toga vremenski korak, koji je kontroliran najmanjom veličinom elementa, može postati premalen da bi bio učinkovit za vremenski marš, što može dovesti i do kvara izračuna.[11]

2.2 Općenito o SPH metodi i njene mogućnosti

U SPH metodi, stanje sustava je predstavljeno skupom čestica, koje posjeduju pojedinačna materijalna svojstva i kreću se prema važećim jednadžbama očuvanja. Od svog izuma osim za rješavanje astrofizičkih problema u trodimenzionalnom prostoru, SPH je opsežno proučavan i proširen na dinamički odgovor s čvrstoćom materijala, kao i na dinamičke tokove fluida s velikim deformacijama.[11]

SPH metoda ima svoje prednosti u odnosu na tradicionalne numeričke metode temeljene na mreži, a jedna od njih je prilagodljivost. Ova prilagodljivost se postiže u ranoj fazi aproksimacije varijable polja koja se izvodi u svakom vremenskom koraku na temelju trenutnog lokalnog skupa proizvoljno raspoređenih čestica što dovodi do toga da ne utječe na formulaciju SPH. Stoga se prirodno može nositi s problemima s iznimno velikim deformacijama, što i je najatraktivnija karakteristika SPH metode. SPH aproksimacija ne zahtijeva unaprijed definiranu mrežu za pružanje bilo kakvog povezivanja čestica u procesu računanja što i predstavlja najveću prednost u odnosu na tradicionalne FEM i FDM metode.[11]

Već prethodno spomenuto SPH metoda se sve više koristi za modeliranje gibanja fluida. Ono što nam omogućava ova metoda, a druge ne nude je očuvanje mase bez dodatnog izračuna jer same čestice predstavljaju masu. Drugo SPH računa tlak iz procjene doprinosa susjednih čestica, a ne rješavanjem linearnih sustava jednadžbi. Zadnje najbitnije, ova metoda stvara slobodne površine za dvofazne tekućine koje izravno djeluju budući da čestice predstavljaju gušću tekućinu (obično voda), a prazan prostor lakšu tekućinu (obično zrak).[11]

Zbog sve veće popularnosti SPH metode, postoje razni izvrsni softveri otvorenog koda temeljeni na ovoj metodi. Oni su širom rasprostranjeni u akademskim krugovima i industrijama, pa se među brojnim mogu istaknuti:[11]

- SWIFT i GADGET koji se koriste za simulacije kozmologije i astrofizike
- PYSPH koji se koristi za simulacije strujanja stlačivog i nestlačivog fluida, elastičnu dinamiku, dinamiku krutog tijela, jednadžbe plitke vode i neke druge probleme
- SPlisHSPlasH koji služi za rješavanje tlaka i algoritma višefaznog toka
- DualSPHysics o kojem će biti pisano u ovom radu, koristi se za simulacije protoka kroz slobodne površine, višefaznih protoka i rješava probleme sprezanje fluida i strukture.
- I tako dalje.[11]

3. OPĆENITO O DUALSPHYSICS

DualSPHysics se temelji na modelu hidrodinamike izglađenih čestica zvanom SPHysics. Kod je razvijen za slučaj proučavanja protoka kroz slobodne površine gdje Eulerova metoda može biti teško primjenjiva. DualSPHysics je skup C++ i CUDA koda dizajniran za inženjerske probleme iz stvarnog života gdje se izvode simulacije s milijunima čestica na CPU-u ili GPU-u.[1]

DualSPHysics potječe iz SPHysicsa koji je model otvorenog koda. FORTRAN kod se koristio u SPHysics koji je pomagao za različite probleme loma valova, ponašanja pri urušavanju brane, interakcije s obalnim strukturama i nekim drugih problemima. Iako SPHysics omogućava rješavanje i simulaciju problema u visokoj razlučivosti, on se slabo koristi za primjenu na stvarne inženjerske probleme. Njegov problem je pretjerano dugo vrijeme izvođenja računala, te se on ne primjenjuje na velikim domenama. Za njegovo korištenje potrebna su razna hardverska ubrzanja te paralelno računanje kako bi bio korisniji za neke stvarne inženjerske probleme.[2]

3.1. Sofveri i njihove instalacije

U ovom poglavlju bit će opisani svi softveri koji su potrebni za obradu ovog rada i postupci instalacije istih kako bi se omogućila priprema XML datoteka u kojima će se mijenjati parametri s ciljem postizanja različitih rezultata za različite parametre. Osim toga bit će opisan postupak instalacije FreeCAD-a koji će se koristiti za potrebe vizualizacije.

3.1.1. DualSPHysics

DualSPHysics je model otvorenog koda koji se može vrlo lako preuzeti sa službene stranice. DualSPHysics sam po sebi nije softver kojem je potrebna instalacija kao što će to biti neki softveri nadalje objašnjeni. To je jedan paket koji sadrži razne primjere, gdje je cijeli problem zapisan u XML formatu.[2]

Osim toga u tom paketu se nalaze SH source datoteke(skripte) i bat datoteke koje je potrebno pokrenuti na CPU ili GPU. SH datoteke se otvaraju na Linux računalnom sustavu, dok bat datoteke se otvaraju na računalnom sustavu windows. Osnovna razlika u pokretanju na CPU i GPU je ta što CPU za neka osobna računala može biti jako spor jer za velike projekte CPU se koristi kod paralelizacije te omogućava korištenje više procesora. GPU je odličan za pokretanje na osobnim računalima jer može ubrzati numeričke modele smanjujući računalne troškove.[3]

3.1.2. Visual Studio Code

Visual Studio code je jedan vrlo lagan i moćan softver koji se koristi za obradu koda, pokretanje lokalnog servera i tako dalje. Otvorenog je koda, dostupan za windows, macOS i Linux. Podržava JavaScript, TypeScript i Node.js no osim toga moguće je okruženje ovog softvera proširiti i s nekim drugim jezicima kao što su C++, C#, Java, Python, PHP i tako dalje.[4]

Sama instalacija nije kompliciрana, pa se tako može samo preuzeti u ovisnosti od sustava cijeli zip file u kojem se nalazi instalacija. Nakon pokretanja postupka instalacije potrebno je obaviti par koraka koji su vrlo jednostavni kako bi se dovršila instalacija. [4]

Visual studio code se najviše koristio u ovom slučaju za obrađivanje podataka u xml formatu te i samu provjeru podataka, a ujedno i za pisanje nekih osnovnih funkcija u pythonu, dok se za pisanje aplikacije u pythonu pomoću kojeg će se omogućiti mijenjanje parametara najviše koristio spyder koji će biti objašnjen naknadno.

Kao što je već prethodno navedeno, za bolje i lakše pisanje koda bilo je potrebno i poželjno instalirati neke ekstenzije, a to su sljedeće:

- VScode-runner: ekstenzija koja omogućava pokretanja raznih programskih jezika kao što je C, C++, Java, Python, Javascript i tako dalje
- VScode-icons: unosi prave ikone u VS code
- Run Terminal Command: ekstenzija koja nam omogućava da unutar ovog softvera pokrenemo command Palette
- Python: ekstenzija koja je proširena verzije pythona koja nam dolazi u osnovnom paketu VScode-runner
- Prettier: Raščlanjuje kod po bojama i puno toga drugog radi lakšeg snalaženja u kodu

3.1.3. XML

XML je skraćenica za Extensible Markup Language koji se u zadnje vrijeme jako puno koristi za u raznim aplikacijama osim za web i za razne druge softvere. Koristi se za upravljanje, prikazivanje i organiziranje podataka. Za razliku od plain-text verzije, XML je puno veći, no cilj XML-a nije smanjenje veličine datoteke, već dobivanje jednostavne strukture podataka.[5]

XML je platforma i jezik koji se može koristiti na raznim sustavima. Može se jako često koristiti u raznim načinima te nam daje fleksibilnost pri prenošenju podataka, za komunikaciju između računalnih programa, za komunikaciju između servera i stranice i tako dalje.[5]

Za web aplikacije, korištenjem XML-a moguće je smanjiti vrijeme očitavanje stranice. Najbitnija činjenica je ta što su informacije unutar XML-a strukturirane kao drvo sa samo jednim korijenom

i ispod korijena veze koje se nazivaju roditelj/dijete. Iz tog razloga lako je pristupiti nekom podatku bez da se očitava cijela datoteka već samo ono što je potrebno. U nastavku je moguće vidjeti osnovna pravila:[5]

- Svaki početni tag mora imati svoj završni tag ili biti samo-zatvarajući tag
- Tag se ne smije preklapati
- XML dokument može imati samo jedan korijen
- Nazivi elemenata moraju poštivati način imenovanja u XML-u
- XML je osjetljiv na velika i mala slova
- XML će zadržati razmake u vašem tekstu

Ako se prate ova pravila prilikom izrade XML-a svaki XML raščlanjivač će biti u mogućnosti za čitanjem podataka iz dokumenta.

3.1.4. Python

Python je jedan od najboljih objektno-orijentiranih, interpretiranih i interaktivnih programskih jezika. Često je uspoređivan s nekim drugim jezicima kao što je Perl, Ruby, C#, Java i tako dalje. Osim što je kao programski jezik jako snažan, on daje vrlo jasnu sintaksu. Koristi module, klase i dinamičke tipove podataka visoke razine. Python osim što se koristi kao glavni jezik u nekim aplikacijama može se koristiti i kao proširenje nekih aplikacija koje su napisane u drugim jezicima kako bi se postiglo jednostavnije skriptiranje ili sučelja za automatizaciju.[6]

3.1.5. FreeCAD

FreeCAD je aplikacija za 3D parametarsko modeliranje. Najviše se koristi za dizajniranje u strojarstvu no može se koristiti i za neke druge stvari gdje je potrebna preciznost i kontrola nad poviješću modeliranja.[7]

Da bi bili u mogućnosti koristiti DualSPHysics, potrebno je koristiti softver model koji je ugrađen u FreeCAD-u te se on naziva DesignSPHysics. Taj nam model omogućava kreiranje raznih simulacije gdje se koristi neki fluid. Daje mogućnost kreiranja 2D i 3D scene gdje nakon izrade cijele simulacije omogućava izvođenje datoteka u XML formatu.[7]

Pomoću DesignSPHysics možemo kreirati granice, fluide, razne plutajuće objekte i tako dalje. Na kraju daje grafički prikaz procesa kako bi osoba koje modelira bila u mogućnosti vizualno vidjeti kako se fluid ponaša u nekim uvjetima.[7]

Sama instalacija je jako jednostavna, potrebno je prvo preuzeti FreeCAD te nakon njene instalacije samo ubaciti taj modul na predviđeno mjesto. Više o mogućnostima ovog softvera bit će objašnjeno u 4 poglavlju.

3.1.6. Spyder

Spyder je moćno znanstveno okruženje napisano u Pythonu koji je dizajniran od strane znanstvenika, inženjera i analitičara podataka. To je softver koji se nalazi u skupu platforme Anaconda, pa za mogućnost preuzimanja spyder-a potrebno je prvo preuzeti platformu anaconda pa potom instalirati softver spyder. Ovaj softver sadrži jedinstvene kombinacije naprednog uređivanja, analize, otklanjanja pogrešaka i funkcionalnosti koji nam omogućava istraživanje podataka, interaktivna izvođenja, duboke preglede i prekrasne mogućnosti vizualizacije znanstvenih paketa.[10]

Osim toga spyder nudi mogućnost korištenja raznih popularnih znanstvenih paketa kao što su NumPy, SciPy, Pandas, IPython, QtConsole, Matplotlib, SymPy i mnoge druge. Osim ugrađenih značajki ovaj softver je moguće proširiti i raznim drugih paketima s treće strane.[10]

3.2. Pregledavanje mogućnosti i tipova zadataka koje nudi DualSPHysics

U nastavku je moguće vidjeti demonstraciju nekih primjera koji se nalaze u paketu DualSPHysics. Neki od tih primjera su lom brane, valovi na plaži, strujanje fluida oko cijevi i puno drugih. Svaki primjer posjeduje .bat scripte koje je moguće pokrenuti u Windowsu te -sh datoteke koje se pokreću na Linux. Osim toga svaki posjeduje XML datoteku u kojoj su definirani razni parametri, pa su tako neki od njih granice, razna svojstva fluida, geometrija i tako dalje.[2]

Nama će najzanimljiviji slučaj biti opstrujavanja fluida oko cijevi, te će taj slučaj naknadno biti i više objašnjen.

3.2.1. Glavni primjeri

DualSPHysics daje razne tipove zadataka no za slučaj glavnih primjera razlikujemo sljedeće primjere:

- Lom brane
- Periodičnost
- Pokretni kvadrat
- Vanjske sile
- Kretanje tanka s fluidom

- Izrada valova
- I tako dalje

3.2.2. Chrono primjeri

Chrono je biblioteka za modeliranje interakcija mehanizma i tekućine. Neki od primjera su:

- Njihalo
- Opruga
- Fleksibilna vrata
- Postavljanje tijela u vodu te modeliranje mehanizma valova na obalu
- Zipline
- Lom brane s kockama
- Turbina
- Vodenica
- I tako dalje

3.2.3. Moordyin primjeri

Moordyin je biblioteka koja uglavnom proučava ponašanje nekog krutog tijela u fluidu, pa tako jedan od primjera su:

- MooredBox
- Moored valovi 2D
- Moored valovi 3D
- Povezane kutije
- Stepenasti spremnik

3.2.4. Spajanje valova primjer

Ova biblioteka se bavi širenjem valova koji koriste višeslojne klipne valove i tehnike zone opuštanja. Osim toga koristi se također za samostalnu generaciju valova i struje. Neki od primjera korištenja ovog modela su:

- Nailaženje običnog vala na plažu
- Regularni valovi generirani pomoću zone opuštanja
- Neregularni valovi generirani pomoću zone relaksacije
- 2D strujanje

- 3D strujanje oko cilindra

3.2.5. Inlet outlet primjeri

Ovaj tip primjera se bavi slučajevima gdje je potrebno implementirati simulacije korištenja otvorenih rubnih uvjeta. Pomoću ovih primjera moguće je testirati sposobnost ulaznih i izlaznih površina da točno reproduciraju protok fluida. Simulira se nekoliko tipova tokova, uključujući protok, ograničene i slobodne površine, protok bez gravitacije, stvaranje valova i tako dalje. Neki od primjera su:

- Opstrujavanje fluida oko cilindra
- Strujanje otvorenog kanala u dvodimenzionalnom kanalu
- Obrnuti tok
- Valovi 2D
- Kutija s 4 ulaza 3D
- Strujanje fluida oko broda

3.2.6. Modificirani dinamički granični uvijeti

Skraćenica za ovu biblioteku je mDBC, te predstavlja novi tip modeliranja. Neki tipovi zadataka koji se modeliraju ovim postupkom su sljedeći:

- Stillwedge
- Poiseuille
- Sloshing
- Lom brane
- Opstrujavanje oko cilindra
- I tako dalje

3.2.7. Višefazni tekući plin

Ova biblioteka nam omogućava praćenje plina u zraku kako se ponaša za slučajeve protoka fluida kroz neki prostor. To se radi tako da se na neke prethodne primjere doda dio koda u xml kako bi obilježili i ovaj slučaj.

DualSPHysics nam tako nudi neke od sljedećih primjera gdje se prati plin zajedno sa protokom fluida:

- Lom brane 2D
- Sloshing
- Površinski tlak
- Lom brane 3D

3.2.8. Višefazno strujanje Ne-Newtonovih fluida

Ova biblioteka nam nudi mogućnost praćenja Ne-Newtonovih formulacija i višefaznih tokova na reološkim modelima. Neki od primjera su:

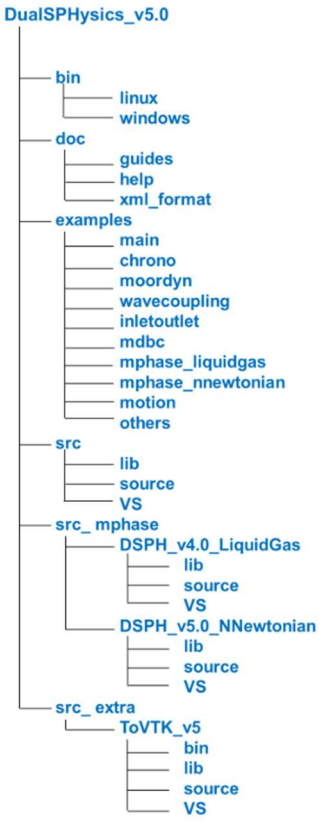
- Lom brane s 3 različita fluida
- 3D slučaj loma brane s 2 različita Ne-Newtonova fluida
- Miješanje dva različita fluida
- Poiseuille

3.3. Pokretanje DualSPHysics

Pokretanje DualSPHysics primjera je moguće odraditi na dva načina. Pomoću .bat koji se pokreće na windows sustavima te pomoću .sh datoteke koje se pokreće na linux sustavima. Osim toga moguće je pokrenuti te datoteke na GPU(grafička kartica) ili na CPU(procesor). Postoje razne specifikacije što se tiče grafičkih kartica, kakve moraju biti da bi se mogle pokrenuti, no najbitnije je posjedovanje Nvidia CUDA-enabled GPU kartice i instaliranu zadnju verziju driver-a. U nastavku će biti objašnjeno sve vezano za pokretanje ovih primjera na računalu.[2]

3.3.1. Struktura datoteke

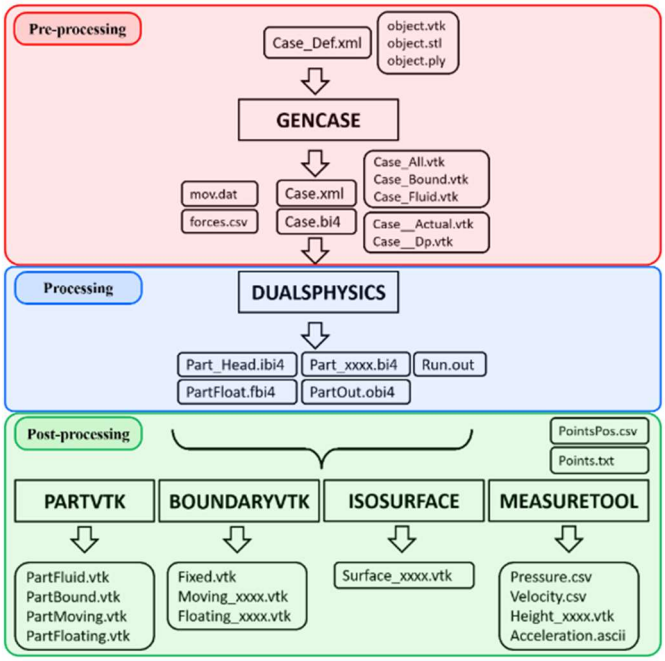
Na slici 3.1. moguće je vidjeti kako glavne mape unutar ovog paketa su bin(sadrži binarne izvršne datoteke za linux i windows, te se nalaze još neke biblioteke potrebne za kod), doc(sadrži mapu guides koja ima par pdf kao uputstva, help mapa sadrži datoteke s opisom parametara te xml_format mapa koja sadrži brojne xml datoteke s raznim oznakama i formatima koje se mogu koristiti kao ulazne datoteke), examples mapa sadrži razne primjere koji su prethodno objašnjeni u poglavlju 3.2., src mapa koja sadrži izvorne datoteke, src_mphase mapa koja sadrži datoteke koje su obvezne za primjere s tekućim plinom i Ne-Newtonovim fluidom i src_extra mapa sadrži osim izvornih datoteka i datoteke s nazivom ToVTK5 koja se koristi kao pomoć pri učitavanju podataka o česticama iz .bi4 tipa podataka te također posjeduje datoteke koje pomažu u stvaranju .vtk i .csv datoteka.[2]



Slika 3.1 Struktura datoteke DualSPHysics v5.0.[2]

3.3.2. Pregled tijeka rada DualSPHysics

Slika 3.2. prikazuje tijek rada s reprezentativnim primjerima ulaznih i izlaznih datoteka.

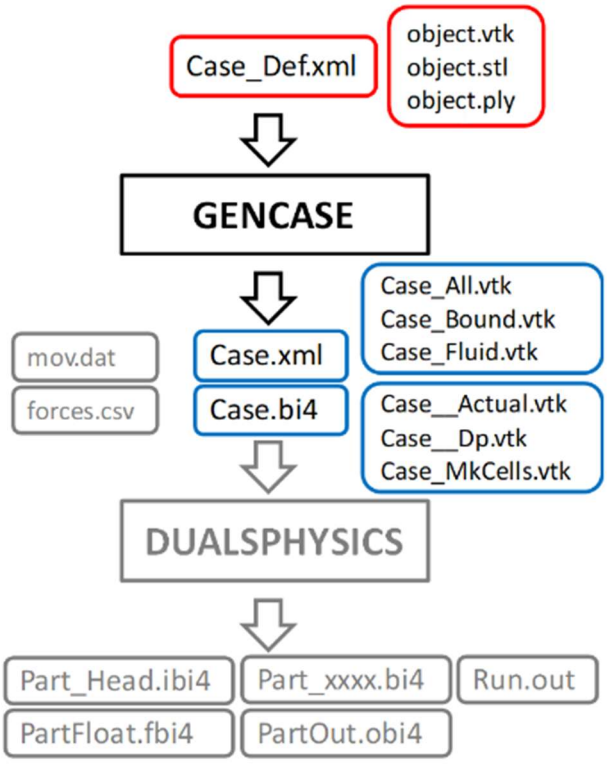


Slika 3.2 Tijek rada DualSPHysics[2]

Cijelo proces se sastoji od 3 faze. Prva faza je pre-processing u kojoj se koristi alat GenCase za pretprocesiranje. U toj se fazi stvara početno stanje čestica(položaj, brzina i gustoća) i definira se različiti SPH parametri za simulaciju. Nakon što se kreira početno stanje DualSPHysics počinje s izvršavanjem. SPH solver se koristi za rješavanje interakcija fluid-fluid, fluid-kruto i čvrsto-kruto tijelo kako bi se mogla definirati vremenska evolucija sustava. Nakon što je simulacija gotova, moguće je koristiti razne alate za naknadnu obradu kao što je vizualizacija simulacije ili za izračune raznih veličina od interesa.[2]

3.3.3. Pretprocesiranje

Za postupak prije procesiranja koristi se program GenCase koji definira početnu konfiguraciju simulacije, opis kretanja pojedinih pokretnih objekata, a isto tako i opis parametara izvođenja u DualSPHysics. Svi ovi podatci se nalaze u XML formatu. Nakon što se pokrene program GenCase stvaraju se dvije izlazne datoteke: Case.xml i Case.bi4.[2]



Slika 3.3 Shematski prikaz postupka prije procesiranja[2]

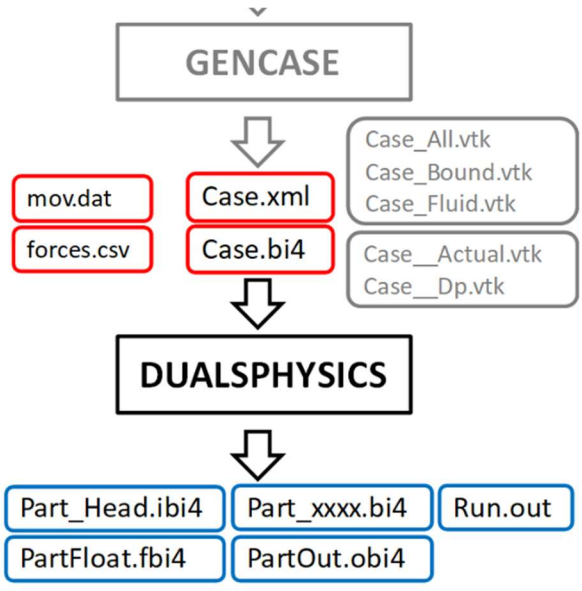
Na slici je moguće vidjeti crveni okvir koji predstavlja ulazne datoteke te plavi okvir koji predstavlja izlazne datoteke. Case.xml datoteka koja se nalazi u plavom okviru sadrži sve parametre za konfiguraciju sustava te za njeno izvođenje. Neki od parametara su duljina izglađivanja, referentna gustoća, gravitacija, koeficijent za izračun tlaka, brzina zvuka, broj čestica sustava, definicije kretanja granice, svojstva tijela koja se gibaju i tako dalje. Case.bi4 je dokument

koji sadrži početno stanje čestice(broj čestica, položaj, brzina i gustoća). Dobivena geometrija čestice koja se nalazi u datotekama case_all.vtk, case_bound.vtk i case_fluid.vtk moguće je provjeriti pomoću paraview. Case_MKCells.vtk sadrže ravnine koje predstavlja skup svih tih čestica te je ova datoteka puno lakša za vizualizaciju u slučaju ogromnih datoteka.[2]

XML datoteka kojom se stvara geometrija i gdje se definiri svi parametri bit će objašnjeno naknadno.[2]

3.3.4. Postupak procesiranja

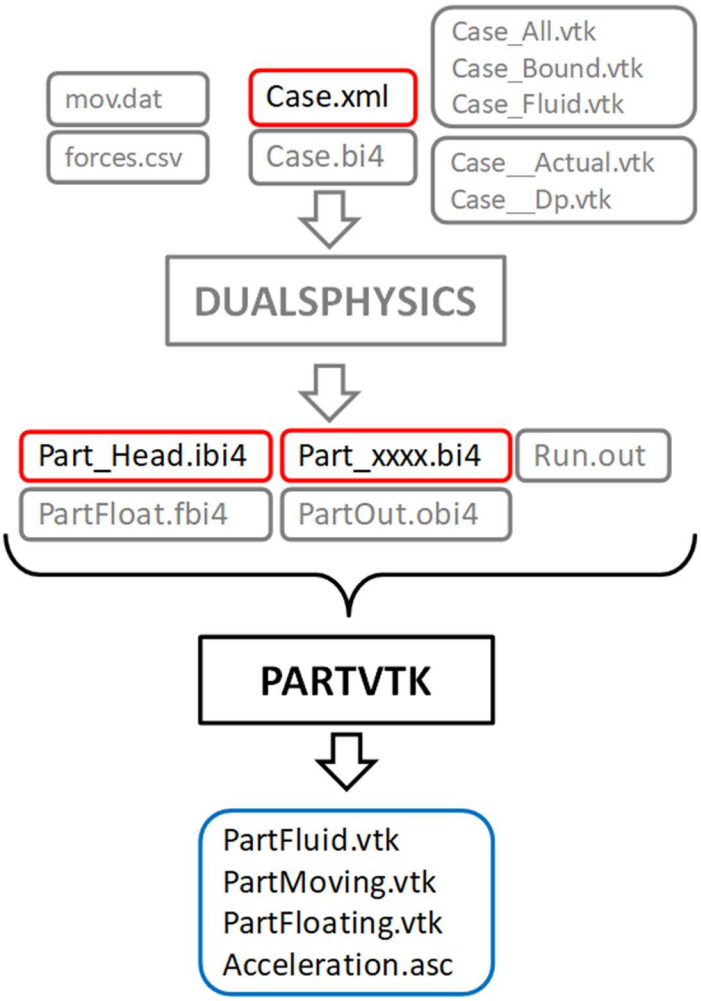
Glavni kod koji izvodi SPH simulacija se naziva DualSPHysics. Potrebne ulazne datoteke za pokretanje DualSPHysics je jedna XML datoteka (Case.xml) i jedna binarna datoteka (Case.bi4) koje se nalaze na slici 3.4. u crvenom okviru. Već je prethodno spomenuto da se u xml datoteci nalaze svi parametri konfiguracije sustava, ključne varijable, broj čestica i tako dalje. Case.bi4 je binarna datoteka koja sadrži podatke o česticama. Na izlazu se dobiju binarne datoteke koje sadrže informacije u česticama u različitim trenutcima (jedan primjer: ako smo stavili vrijeme trajanje simulacije 1 sekundu, a definirani vremenski razmak između svakog trenutka dt=0.1 sekunda, na izlazu se dobije 10 različitih binarnih datoteka) te se one nalaze pod nazivom Part0000.bi4, Part0001.bi4 i tako dalje ovisno o broju zabilježenih trenutaka. Osim tih datoteka još se dobije datoteka sa isključenim česticama PartOut.obi4, datoteka s osnovnim informacijama za izvršavanje, datoteka s informacijama o poslije procesorskim postupcima Part_Head.ibi4, datoteka s informacijama o plutajućim objektima PartFloat.fbi4 i tekstualna datoteka sa zapisnikom izvršenja Run.out. Sve izlazne datoteke se nalaze na slici 3.4. u plavom okviru.[2]



Slika 3.4 Shematski prikaz postupka procesa[2]

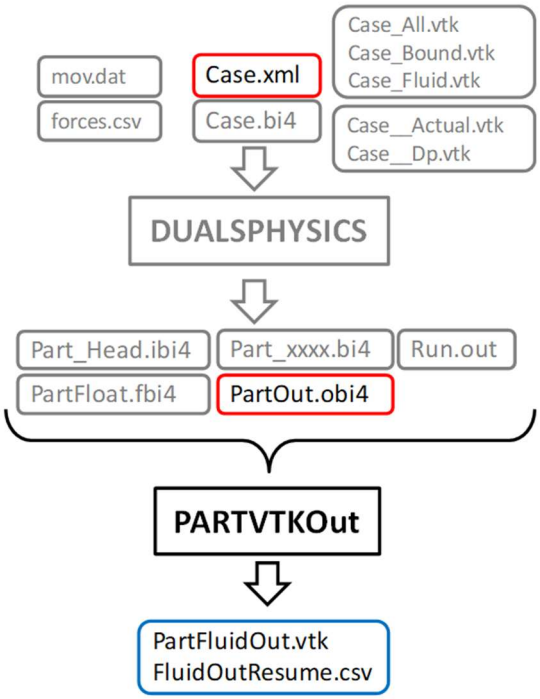
3.3.5. Postupak poslije procesiranja

Sa slike 3.5. moguće je vidjeti kako alat PartVTK u postupku poslije procesa pretvara binarne datoteke u različite formate koje se mogu vizualizirati. Pomoću PartVTK moguće je vizualizirati izlazne podatke o česticama.[2]



Slika 3.5 Shematski prikaz procesa poslije procesiranja za vizualizaciju izlaznih podataka o česticama[2]

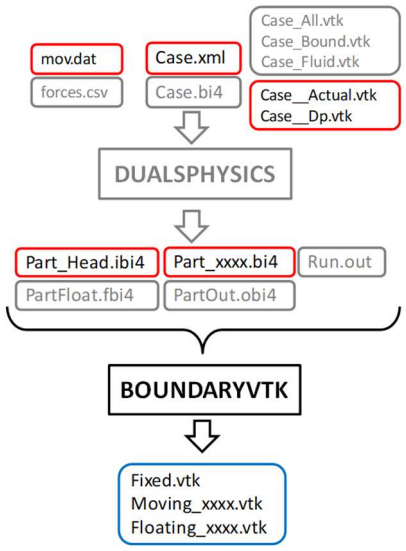
Na prethodnoj slici je vidljivo da su Case.xml, Part_Head.ibi4 i Part_xxxx.bi4 ulazne datoteke koje prolaze kroz PartVTK koji nam na izlazu daje PartFluid.vtk, PartMoving.vtk, PartFloating.vtk i Acceleration.asc.



Slika 3.6 Shematski prikaz procesa poslije procesiranja za vizualizaciju izlaznih podataka o česticama koje su isključene iz simulacije[2]

Slika 3.6 nam pokazuje prikaz kako PartVTKOut koristi Case.xml i PartOut.obi4 kao ulazne datoteke kako bi se dobile izlazne datoteke PartFluidOut.vtk i FluidOutResume.csv koje će se koristiti za vizualizaciju izlaznih podataka o česticama koje su isključene iz simulacije.[2]

Da bi se postigla mogućnost vizualizacije granica formirane od graničnih čestica, potrebno je koristiti BoundaryVTK kod. Moguće je vidjeti na slici 3.7. kako su u ovom slučaju mov.dat, Case.xml, Case_Actual.vtk, Case_Dp.vtk, Part_Head.ibi4 i Part_xxxx.bi4 ulazne datoteke, a na izlazu se dobiju datoteke Fixed.vtk, Mooving_xxxx.vtk i Floating_xxxx.vtk.[2]



Slika 3.7 Shematski prikaz procesa poslije procesiranja za vizualizaciju grupe čestica koje čine granicu[2]

3.4. Kompajliranje u DualSPHysics

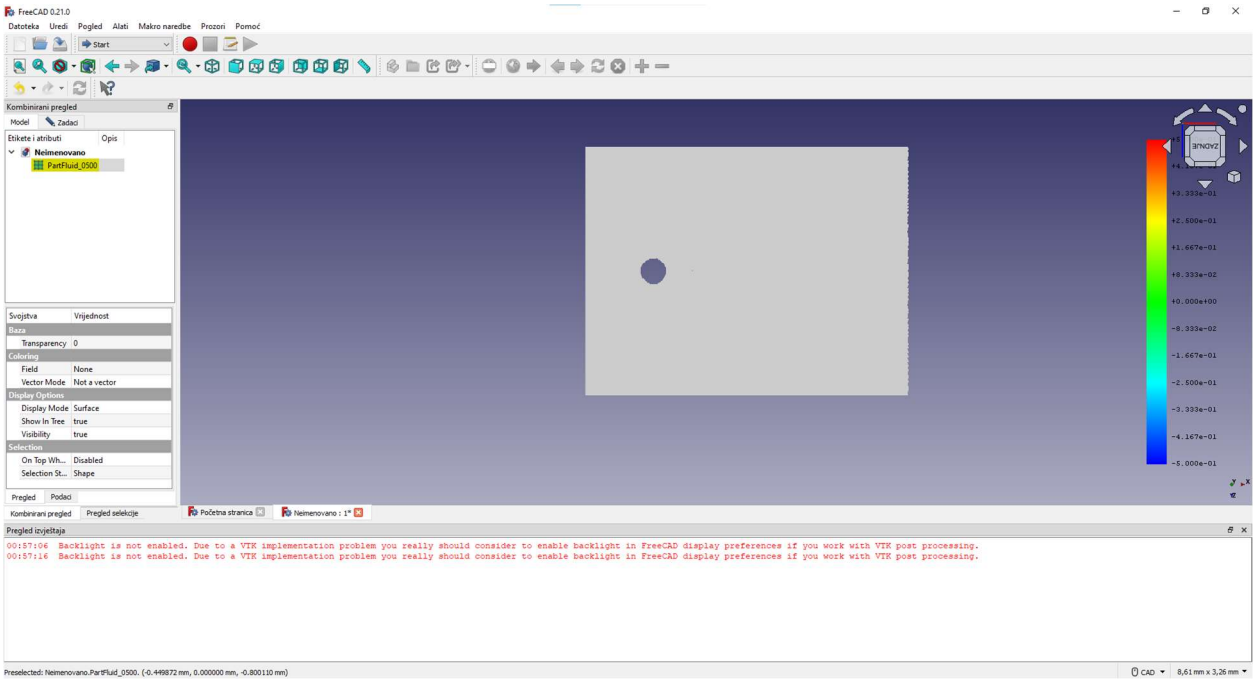
Kod je moguće kompajlirati za CPU ili za CPU&GPU. Kako bi proces kompajliranja bio uspješan na računalu moraju biti instalirani C++ i CUDA prevodioci. Većina izvornog koda je zajednička i za CPU i GPU koja omogućava pokretanje bez GPU s omogućenim CUDA prevodiocem, koristeći samo CPU implementaciju.[2]

Da bi bili u mogućnosti pokrenuti DualSPHysics na GPU pomoću izvršne datoteke potrebno je posjedovanje Nvidia CUDA GPU kartice te mora biti instaliran zadnja verzija driver-a. Za uspješno kompajliranje na računalu moraju biti instalirani programski jezik CUDA i nvcc kompajler.[2]

Proces kompajliranja je moguće napraviti na windowsu koristeći VisualStudio pomoću Chrono ili MoorDyn+ alata. Na linuxu se također koristi Chrono ili MoorDyn+ alati. Osim toga moguće je proces kompajliranja provesti pomoću cross-platform (program koji može raditi na više platformi ili radnih okruženja, uključuje razne hardverske i softverske sustave) koristeći CMAKE. To je nova metoda koja je podržana za verzije v4.0 DualSPHysics i novije.[2]

4. OPĆENITO O FREECAD-U

FreeCAD je aplikacija otvorenog koda za parametrično 3D modeliranje. Zadatak ovog softvera je projektiranje objekata iz stvarnog života. Parametarsko modeliranje predstavlja vrstu modeliranja gdje se oblik nekog 3D objekta kojeg se dizajnira kontrolira parametrima, ako se uzme na primjer dizajniranje neke kocke, ona se može kontrolirati uz pomoć tri parametra, a to su dužina, visina i širina. Kao i u većini softvera za modeliranje pa tako i ovdje ti se parametri mogu mijenjati u bilo kojem trenutku nakon što je neki objekt stvoren.[9]



Slika 4.1 Prikaz FreeCAD sučelja

FreeCAD nije dizajniran samo za određenu vrstu posla ili za izradu samo određene vrste objekata. Ovaj softver omogućava široku primjenu te dopušta korisnicima proizvodnju modela svih veličina od najmanjih kao što su neke elektroničke komponente pa sve do velikih modela kao što su zgrade.[9]

U nastavku će biti objašnjeno pojedine stvari koje su potrebne napraviti za uspješno postavljanje simulacije u FreeCAD-u uz pretpostavku prethodnog poznavanja 3D modeliranje i dizajniranja.

FreeCAD je jako fleksibilan softver koji omogućava korisniku da radi ono što mu je potrebno. Daje slobodu za gradnju čega god je potrebno (mogućnost parametarskog modeliranja u stvarnoj veličini), prebacivanje iz 2D u 3D i obrnuto, pristupačan, fleksibilan i integriran softver (radi na brojnim platformama, jako prilagodljiv i proširiv softver, ima mogućnost čitanja i pisanja u raznim formatima datoteka) te osim toga nudi i broje druge moderne alate kao što je analiza konačnih elemenata (FEA), eksperimentalni CFD, namjenski BIM, Geodata ili CAM/CNC radne stolove, module za simulacije robota koji nam daje u uvid kretanje robota i tako dalje.[7]

4.1. Postavljanje DualSPHysics u FreeCAD

Kako ovaj softver ne prepoznaje odmah u početku DualSPHysics potrebno je implementirati ovaj modul. Taj se postupak sastoji od dva koraka. Prvi korak je instalacija DesignSPHysics paketa u markronaredbama. To se radi na sljedeći način. Prvo se otvara izbornik Macro i odlazi se u makronaredbe. Na novom prozoru koji se otvorio nalazi se dole desno dugme „Preuzmi“. Potom se otvara novi prozor u kojem se nude svi mogući moduli koje Freecad nudi, u tražilicu je potrebno upisati DesignSPHysics te potom i instalirati taj modul.[8]

Drugi korak je instaliranje paketa za razvoj. To se pak radi tako da se klonira repozitorij sa github-a koju DualSPHysics nudi za DesignSPHysics, potom se kloniranoj mapi mijenja ime u DesignSPHysics. Ista ta mapa se mora kopirati u mapu Mod koja se nalazi na mjestu gdje je instaliran FreeCAD. Zadnje još potrebno je kopiranje datoteke DesignSPHysics.FCMacro koja se nalazi u kloniranoj mapi te zalijepiti u mapu FreeCAD/Macro kako bi imali mogućnost korištenja ove makronaredbe.[8]

5. OPIS PROBLEMA

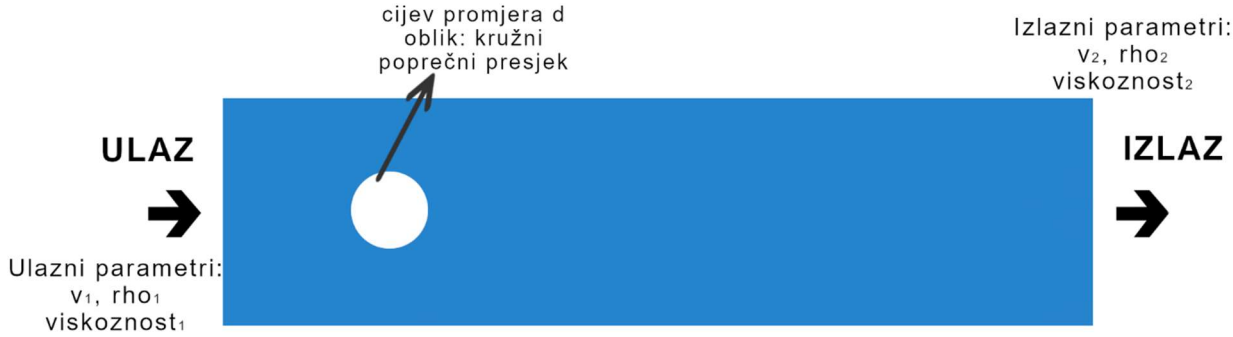
Cilj ovog poglavlja je detaljnije objasniti primjer strujanja fluida oko cilindra kroz otvorene granice. Kao što je prethodno teoretski objašnjeno, zbog lagrangeove metode gdje se prati čestica, a ne kontrolni volumen, omogućava se kvalitetno praćenje svake čestice zbog čega se lako mogu dobiti rezultate gustoće, brzine, sila, tlaka i mnoge druge parametre za iste te čestice u svakom trenutku. Kako bi prethodno navedeno prikazali istinitom, u rješenjima će se prikazati rezultati nekih različitih vrsta simulacija koje se dobiju jako brzo i jednostavno primjenom ove metode, dok kada bi se koristila Eulerova formula ili neka druga metoda, proračun bi trajao jako dugo, te upitno je dali bi rješenje ikad konvergiralo i dalo neke rezultate.

5.1. Pregled primjera strujanja fluida oko cijevi

Primjer koji će se obrađivati je strujanje fluida oko cijevi. Taj se primjer može naći u examples/inletoutlet. Spada u tip zadataka inletoutlet jer se sastoji od protoka fluida gdje se mora definirati ulaz, a potom i izlaz sa svim ostalim parametrima, za slučajeve kada se treba implementirati simulaciju s korištenjem otvorenih rubnih uvjeta, koje će naknadno biti objašnjeno.[2]

5.1.1 Opis primjera strujanja fluida oko cijevi

Ovaj slučaj se može promatrati i kao 2D i kao 3D slučaj, zbog jednostavnosti i ne posjedovanja snažne računalne opreme, obrađivat će se algoritam u simulaciji 2D protoka fluida oko cilindra.



Slika 5.1 Opstrujavanje fluida oko cijevi

Na slici 5.1. moguće je vidjeti fluidnu domenu označeno plavom bojom i cilindar kružnog poprečnog presjeka nekog promjera d koji će se moći mijenjati. Osim toga bit će moguće mijenjati oblik i sve druge parametre. Cilindar je diskretiziran dinamičkim graničnim česticama i okružen je vodom. Tekućina se inicijalizira s konstantnom brzinom od 1 m/s te nultom y-brzinom.

Osim toga bitno je napomenuti kako ovaj zadatak ima lijevu i desnu površinu otvorenih rubnih uvjeta koji predstavljaju ulaz i izlaz u fluidnu domenu, te ostale površine su definirane kao čvrste granice zatvorenih rubnih uvjeta. Svaki ulaz i izlaz opisan je raznim parametrima kao što je viskoznost, gustoća, brzina, tlak i tako dalje, no to će u sljedećem poglavlju prilikom objašnjenja XML datoteke biti detaljnije opisano.

Cilj ovog seminara je objašnjenje strujanja tog fluida oko cijevi, to jest što će se dogoditi pri promjeni parametara, kako se ponaša fluid kad se parametri mijenjaju. To su sve neke stvari koje je potrebno objasniti i opisati. Kako se danas uglavnom prije same izvedbe obavlja neka računalna simulacija, a ovakve vrste protoka su jako česte u stvarnom životu, potrebno je ovakav tip primjera proći kako bi se lakše mogla koristiti primjena ovog zadatka za neke stvarne probleme.

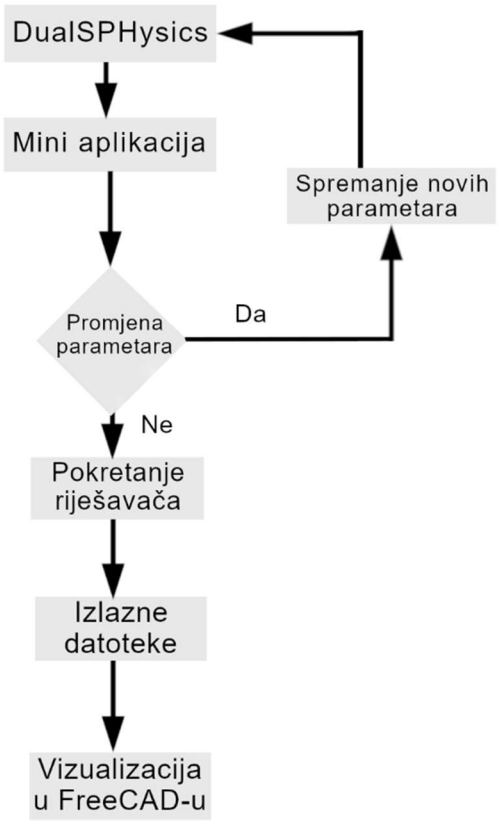
5.1.2 Korišteni softveri za rješavanje problema

Kako se već prethodno spomenuto pregledava se metoda hidrodinamike izglađenih čestica koja će se opisati pomoću rješavača DualSPHysics. Kako će se koristiti mogućnost mijenjanja parametara numeričke simulacije koristit će se Visual Studio Code za pregled parametara i Spyder alat koji se nalazi u sklopu programa Anaconda za pisanje python koda s kojim će se omogućiti automatizacija u primjeni ovog tipa zadatka. Za vizualizaciju će se koristiti FreeCAD. Cijeli proces će biti moguće pokrenuti na 2 načina, na CPU i GPU. CPU je proces koji je puno sporiji dok GPU taj proces obavi 90% brže nego što to obavi CPU, stoga sve simulacije će se obavljati na GPU.

5.2 Mogućnost automatizacije

DualSPHysics je softver koji nudi brojna rješenja. No jedan veliki problem koji ovaj softver ima je taj što ga je jako teško koristiti. Potrebno je ogromno znanje i razumijevanje kako ovaj softver radi, te zbog kompleksnosti često se kao takav izbjegava za korištenje. Osim svoje kompleksnosti, to je softver koji ne posjeduje paraview s kojim se omogućuje vizualizacija svih rješenja nakon proračuna.

Glavni cilj ovog rada je prikazati mogućnost automatizacije gdje će korisnik biti u mogućnosti koristiti softver, a da ne mora posjedovati znanje u radu s XML datotekama, ne mora posjedovati iskustva u radu sa svim parametrima koje ovaj softver navodi u svojim dokumentima. Cilj je da se pokretanjem zasebne mini aplikacije korisniku omogući mijenjanje ulaznih parametara prije nego što se krene s proračunom i omogući lakša vizualizacija, pogotovo što je potrebno poznavati rad sa FreeCAD softverom koji također nije jednostavan za koristiti u radu sa simulacijama.



Slika 5.2 Slijed rada aplikacije

Slikom 5.2 prikazan je slijed rada koju aplikacija nudi. Prvo je potrebno postojanje DualSPHysics rješavača i njegovi primjeri, potom posjedovanje aplikacije koja nudi mogućnost promjere parametara. U slučaju odabira mogućnosti za mijenjanjem ulaznih parametara potrebno je spremiti sve te nove vrijednosti u XML datoteku. Nakon što se promjene svi parametri potrebno je pokrenuti rješavač koji se također obavlja sve pomoću aplikacije. Na kraju se dobije kao rezultat izlazne datoteke koje se koriste za vizualizaciju u FreeCAD softveru.

U sljedećem poglavlju bit će prikazan kod mini aplikacije u pythonu koji će pomoći u radu sa DualSPHysics riješavačem.

6. METODOLOGIJA

Strujanje fluida oko cilindra promjera D=0.2m diskretiziran je dinamičkim graničnim česticama, gdje će fluid za sve sljedeće primjere biti voda, no postoji mogućnost i mijenjanja tipa fluida, a to se radi tako da mu se mijenja vrijednost gustoće i viskoznosti. Tekućina se inicijalizira s konstantnom brzinom u smjeru osi x U = 1 m/s, dok je brzina u smjeru osi y jednaka 0. Kako se radi o 2D slučaju sve u smjeru osi z je zanemarivo. Reynoldsov broj se računa za ovakav slučaj na sljedeći način:[2]

$$Re = U * \frac{D}{nu} \tag{6.1}$$

gdje je:

Re – Reynoldsov broj

U – brzina fluida [m/s]

D – promjer cilindra [m]

nu – kinetička viskoznost [m^2/s]

Reynoldsov broj nam daje odgovor na to dali ovaj sustav struji turbulentno ili laminarno, a to je moguće provjeriti tako da usporedimo s graničnom vrijednošću reynoldsovog broja, tako što ako je Re manji od granične vrijednosti onda je to laminarno strujanje, te ako je veći od granične vrijednosti tada je to turbulentno strujanje.

6.1 Objašnjenje XML datoteke

U nastavku će biti opisan svaki dio XML datoteke u kojem su definirani svi parametri i geometrija za primjer strujanja fluida oko cilindra.

6.1.1 Konstantni parametri

Da bi simulacija bila uspješna, za svaku je potrebno definirati razne parametre, geometriju, te također definirati koje izlazne datoteke se dobiju kao rezultat proračuna i mnogo toga drugog. Postoje parametri koji se mijenjaju tijekom simulacije no postoje i parametri koji su konstantni.

```
<constantsdef>
    <gravity x="0" y="0" z="0" comment="Gravitational acceleration" units_comment="m/s^2" />
    <rhop0 value="1000" comment="Reference density of the fluid" units_comment="kg/m^3" />
    <hswl value="0" auto="true" comment="Maximum still water level to calculate speedofsound using coefsound" units_comment="metres (m)" />
    <gamma value="7" comment="Politropic constant for water used in the state equation" />
    <speedsystem value="0" auto="true" comment="Maximum system speed (by default the dam-break propagation is used)" />
    <coefsound value="0" comment="Coefficient to multiply speedsystem" />
    <speedsound value="15" auto="false" comment="Speed of sound to use in the simulation (by default speedofsound=coefsound*speedsystem)" />
    <hdp value="2" comment="Alternative option to calculate the smoothing length (h=hdp*dp)" />
    <cflnumber value="0.2" comment="Coefficient to multiply dt" />
</constantsdef>
<mkconfig boundcount="240" fluidcount="9" />
```

Slika 6.1 Parametri koji su konstantne

Na slici 6.1 se nalazi dio koda iz XML datoteke koji nam prikazuje parametre koji su konstantni te se ne mogu promijeniti, pa su tako neki od njih su gravitacija koja ovdje iznosi 0, referentna gustoća fluida koja iznosi 1000 kg/m^3, brzina zvuka koja se koristi u simulaciji je 15 m/s.

Mkconfig nije parametar koji utječe na vrstu fluida, to je parametar pomoću kojeg se radi diskretizacija domene pa se tako koristi za redoslijed stvaranja objekata. U ovom slučaju definirano je 240 oznaka za granične čestice i 10 oznaka za čestice tekućine.

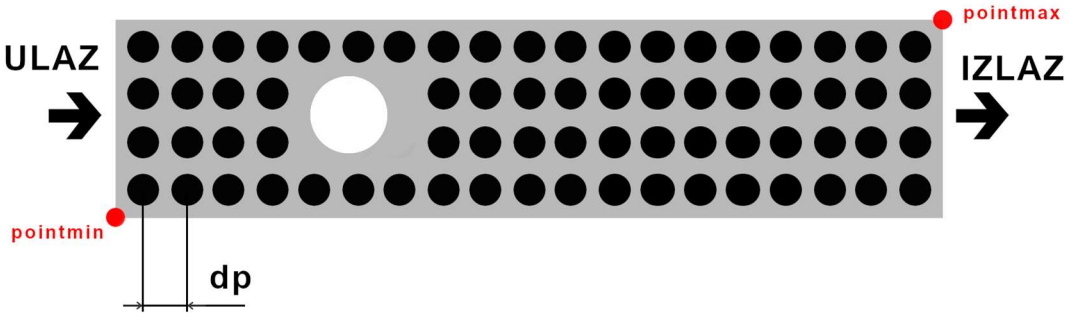
6.1.2 Izrada geometrije

Kako bi bili u mogućnosti napraviti simulaciju potrebno je definirati geometriju.

```
<geometry>
    <definition dp="0.01">
        <pointmin x="-1.2" y="0" z="-2" />
        <pointmax x="2.2" y="0" z="2" />
    </definition>
```

Slika 6.2 Definicija domene gdje će čestice biti kreirane

Na slici 6.2 moguće je vidjeti definicija čestica u zadanoj domeni. Pa tako dp predstavlja razmak između svake čestice te njenim mijenjanjem, podešava se i ukupan broj čestica za cijelu domenu. Pointmin i pointmax predstavljaju dimenzije domene gdje će čestice biti kreirane što je sve i vidljivo na slici 6.3.[1]



Slika 6.3 Prikaz primjera slaganja čestica u domeni

Nakon što se definira način slaganja čestica tj. kako će se obaviti diskretizacija, potrebno je nadalje kreirati sve granice.

```
<mainlist>
    <setshapemode>dp | bound</setshapemode>
    <setdrawmode mode="full" />
    <!-- FLUID DOMAIN -->
    <setmkfluid mk="0" />
    <drawbox>
        <boxfill>solid</boxfill>
        <point x="-0.510" y="-0.1" z="-1.000" />
        <size x="2.520" y="0.2" z="2.000" />
    </drawbox>
    <!-- OPEN BOUNDARIES -->
    <setmkfluid mk="1" />
    <drawbox>
        <boxfill>left</boxfill>
        <point x="-0.510" y="-0.1" z="-1.000" />
        <size x="2.520" y="0.2" z="2.000" />
    </drawbox>
    <setmkfluid mk="2" />
    <drawbox>
        <boxfill>right</boxfill>
        <point x="-0.510" y="-0.1" z="-1.000" />
        <size x="2.520" y="0.2" z="2.000" />
    </drawbox>
    <!-- CYLINDER -->
    <setmkbound mk="0" />
    <drawcylinder radius="0.1">
        <point x="0" y="-0.1" z="0" />
        <point x="0" y="0.1" z="0" />
    </drawcylinder>
    <!-- SOLID BOUNDARY -->
    <setmkbound mk="1" />
    <drawbox>
        <boxfill>solid</boxfill>
        <point x="-0.600" y="-0.1" z="-1.030" />
        <size x="2.700" y="0.2" z="0.030" />
    </drawbox>
    <setmkbound mk="2" />
    <drawbox>
        <boxfill>solid</boxfill>
        <point x="-0.600" y="-0.1" z="1.000" />
        <size x="2.700" y="0.2" z="0.030" />
    </drawbox>
    <!-- END -->
    <shapeout file="" />
</mainlist>
</commands>
```

Slika 6.4 Geometrija za laminarno gibanje

Slika 6.4 nam prikazuje XML kod u kojem je upisana geometrija cijelog sustava strujanja fluida oko cijevi. Prva bitna stavka je setshapemode koja se koristi za definiranje operacije crtanja potrebne za stvaranje VTK datoteka. Setdrawmode indicira na koji način će se stvarati točke u kojima će biti generirane čestice. Mode full naredba predstavlja mogućnost da se kombinira površine i čvrsto tijelo.[1]

Nadalje dolaze setovi sličnih naredbi koje se ponavljaju 6 puta. Prvi set predstavlja kreiranje domene fluida, za fluid će se postavljati mk=0 pa se unutar naredbe setmkfluid i upisuje tako. Kasnije će se pojaviti i naredba setmkbound koja nam predstavlja granicu. Nakon naredbe setmkfluid dolazi naredba drawbox pomoću koje se i kreira domena. Unutar iste te naredbe nalazi se 3 nove naredbe. Prva naredba je boxfill koja će definirati gdje će se nalaziti površine vode (u ovom slučaju se nalaze na svih 6 površina domene jer je definirano vrijednost solid). Nakon toga potrebno je definirati veličinu te domene, a to se radi tako da se koristi naredba point i size. Pomoću naredbe point naredbe se definira od koje točke počinje domena(potrebno je definirati atribute x, y i z) te naredba size nam definira veličinu te domene. Kako je u ovom slučaju vrijednost atributa x = -0.510, a size ima vrijednost x = 2.520, to znači da će druga točka na drugoj strani domene biti na vrijednosti x = 2.01. Isto vrijedi i za os y i z.[1]

Osim definiranje domene fluida potrebno je definirati još dvije površine koje će predstavljati ulaz i izlaz. To su površine koje predstavljaju otvoreni rubni uvjet. Jedina razlika u odnosu na definiranje cijele domene fluida je ta što za lijevu površinu se koristi vrijednost mk = 1 i definira se boxfill : left, dok druga površina poprima vrijednost mk = 2 i boxfill : right.[1]

Nakon što je definirana domena fluida i otvoreni rubni uvjeti, potrebno je definirati još i zatvorene rubne uvjete te cilindar oko kojeg će strujati fluid. Sve se radi na sličan način no jedina razlika je ta što se više ne korist setmkfluid naredba nego setmkbound što i inicira na to da se mora kreirati zatvoreni rubni uvjeti(granica koja je čvrsta koja nije otvorena), te za definiranje cilindra koristi se naredba drawcylinder koja je opisana polumjerom i s dvije točke za definiranje pozicije unutar fluidne domene. Zadnja definirana naredba shapeout je naredba koja se koristi za kreiranje VTK datoteka (poligona) zatvorenih rubnih uvjet.[1]

6.1.3 Definiranje pokreta i brzine

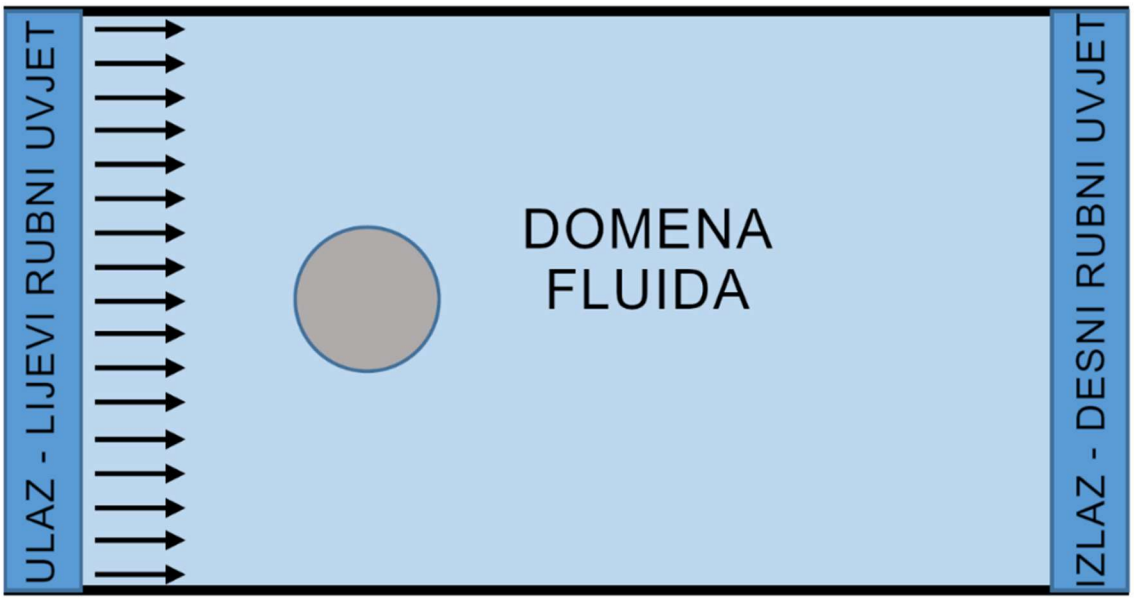
Nakon uspješnog definiranja konstantnih vrijednosti i geometrije, potrebno je definirati vrijednosti brzine i pokreta, tj. dati objašnjenje DualSPHysics-u kako će se neki fluid kretati ponašati i tako dalje.

```
<initials>
    <velocity mkfluid="0" x="1.0" y="0" z="0" />
</initials>
<motion>
    <objreal ref="1">
        <begin mov="1" start="0" finish="100" />
        <mvrect id="1" duration="100">
            <vel x="1" y="0" z="0" units_comment="m/s" />
        </mvrect>
    </objreal>
    <objreal ref="2">
        <begin mov="1" start="0" finish="100" />
        <mvrect id="1" duration="100">
            <vel x="1" y="0" z="0" units_comment="m/s" />
        </mvrect>
    </objreal>
</motion>
```

Slika 6.5 Definicija inicijalne brzine i opis kretanja granica

Na slici 6.5 može se vidjeti naredba initials koja definira neko posebno ponašanje za neku skupinu čestica tekućina. U ovom slučaju definirana je brzina svih čestica fluida u smjeru osi x = 1.0 m/s. U slučaju potrebe za mijenjanjem brzine potrebno je ovdje i na svakom mjestu gdje je definirana brzina napraviti promjene, pa je tako sljedeća promjena potrebna i u motion naredbi.[1]

Motion je naredba pomoću koje se definira opis kretanja granica. Na slici 6.5 moguće je vidjeti dva poziva naredbe objreal. Objreal je naredba pomoću koje se pozivaju svi elementi kreirani u geometriji te se u ovom slučaju pozivaju samo otvoreni rubni uvjeti. Begin naredba definira što će se gibati dok mvrect definira kakvo će gibanje biti. Unutar mvrect potrebno je definirati brzinu tog gibanja.



Slika 6.6 Prikaz rubnih uvjeta i domene fluida

Na slici 6.6 moguće je vidjeti prethodno objašnjeno, kako izgleda domena fluida, te gdje se nalaze otvoreni rubni izlazi. Smjer kretanja fluida je u smjeru osi x, tj. od lijevog do desnog rubnog uvjeta. Otvoreni rubni uvjet predstavlja vrstu rubnog uvjeta koji daje mogućnost da fluid prolazi kroz taj rub tj. da se ne nalazi neka prepreka. U računalnim simulacijama to je potrebno definirati kako bi program znao o čemu se radi.

6.1.4 Definicija osnovnih parametara za ulaz i izlaz

Sljedeći korak pri pisanju koda u XML datoteci je definiranje osnovnih parametara koje dolaze za ulaz i izlaz kao što je vidljivo na slici 6.7.

Memoryresize je naredba koja prati i kontrolira koliko se često veličina nizova čestica povećava kako bi se mogla izbjeći česta promjena lokacije. Naredba deterlimit definira tip ekstrapolacije dodijeljene zonama ulaza i izlaza.[1]

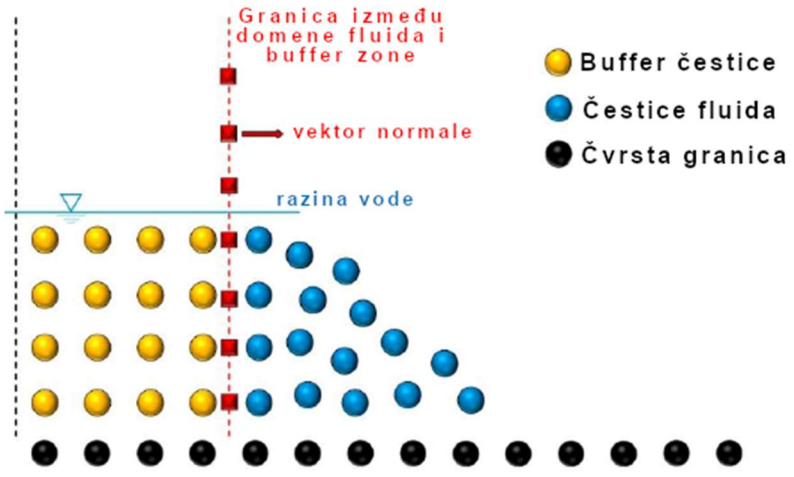
Nakon definiranja memoryresize i deterlimit, dolazi dio koda koji direktno opisuje što će se događati na ulaznim i izlaznim zonama. Tri naredbe koje moraju biti definirane su refilling, inputtreatment i layers. Refilling naredba definira način na koji će se zona ulaza ili izlaza održati ispunjenom česticama kada uđu u fluidno područje te se potom pretvaraju u fluid. Za konkretni

slučaj odabire se opcija 0 (value = 0) koja govori da će ulaz i izlaz biti ispunjeni česticama do vrha. Za svaku česticu koja uđe u prostor fluida, bit će zamjenjena česticom na ulazu.[1]

Čestice koje se nalaze na ulazu i izlazu još se zovu Buffer čestice, a ulaz i izlaz se još nazivaju i buffer zona. Naredba inputtreatment definira ponašanje fluidnih čestica koje ulaze u buffer zonu tj. u zonu ulaza ili izlaza. Za ulaz je potrebno odabrati opciju broj dva (value = 2) što predstavlja da će se sve čestice fluida koje ulaze u ulaznu zonu biti isključene iz simulacije. Ova opcija je prikladna samo za ulaz dok je za izlazno zonu prikladnija opcija 1 (value = 1) koja govori da sve čestice tekućina koje ulaze u buffer zonu (u ovom slučaju u zonu izlaza) se pretvaraju u buffer čestice.[1]

```
<inout>
    <memoryresize size0="2" size="4" comment="Initial memory resize (size0) and the following
                                             memory resizes according to initial inlet/outlet particles (default=2 and 4)" />
    <determlimit value="1e+3" comment="Use 1e-3 for first_order or 1e+3 for zeroth_order (default=1e+3)" />

    <inoutzone>
        <refilling value="0" comment="Refilling mode. 0:Simple full, 1:Simple below zsurf,
                                      2:Advanced for reverse flows (very slow) (default=1)" />

        <inputtreatment value="2" comment="Treatment of fluid entering the zone. 0:No changes, 1:Convert fluid (necessary for outlet),
                                           2:Remove fluid (recommended for inlet)" />

        <layers value="4" comment="Number of inlet/outlet particle layers" />
        <zone2d comment="Input zone for 2D simulations">
            <particles mkfluid="1" direction="right" />
        </zone2d>
        <imposevelocity mode="0" comment="Imposed velocity 0:fixed value, 1:variable value, 2:Extrapolated value (default=0)">
            <velocity v="1.0" comment="Uniform velocity" units_comment="m/s" />
        </imposevelocity>
        <imposerhop mode="2" comment="Outlet rhop 0:Imposed fixed value, 1:Hydrostatic, 2:Extrapolated from ghost nodes (default=0)" />
    </inoutzone>

    <inoutzone>
        <refilling value="0" comment="Refilling mode. 0:Simple full, 1:Simple below zsurf,
                                      2:Advanced for reverse flows (very slow) (default=1)" />

        <inputtreatment value="1" comment="Treatment of fluid entering the zone. 0:No changes,
                                           1:Convert fluid (necessary for outlet), 2:Remove fluid (recommended for inlet)" />
        <layers value="4" comment="Number of inlet/outlet particle layers" />
        <zone2d comment="Input zone for 2-D simulations">
            <particles mkfluid="2" direction="left" />
        </zone2d>
        <imposevelocity mode="2" comment="Imposed velocity 0:fixed value, 1:variable value, 2:Extrapolated value (default=0)" />
        <imposerhop mode="2" comment="Outlet rhop 0:Imposed fixed value, 1:Hydrostatic, 2:Extrapolated from ghost nodes (default=0)" />
    </inoutzone>

</inout>
```

Slika 6.7 Definiranje parametara za otvorene rubne uvjete



Slika 6.8 Prikaz slojeva čestica u buffer i fluidnoj zoni

Naredba Layers definira broj slojeva buffer čestica u buffer zoni. Na slici 6.8 moguće je vidjeti kako postoje četiri sloja čestica. Osim toga za opaziti je isto da se ovdje radi o ulazu. Vidljivo je da će se buffer čestice pretvarati u čestice fluida.[1]

Osim toga slikom 6.8 objašnjen je postupak rada DualSPHysics softvera, tj. opisano je kako se čestice ponašaju prilikom protoka fluida. Ova metoda vrijedi za sve primjere koje nam nudi ovaj softver.

Sljedeća jako bitna naredba je imposevelocity koja definira brzinu kretanja fluida unutar sustava. Brzinu je potrebno definirati i za ulaz i za izlaz pa tako za ulaz se odabire mod 0. To predstavlja da je brzina jednolika pa je stoga potrebno definirati još jedno naredbu velocity pomoću koje navodimo iznos jednolike brzine. Za izlaz se odabire mod dva što predstavlja da će se brzina na izlazu ekstrapolirati. Za ovaj slučaj nije potrebna definicija iznosa brzine na izlazu jer će se ona ekstrapolacijom izračunati. Ovo se i preporučuje za izlaz kako bi se izbjeglo moguće greške.[1]

Zadnja bitna naredba je imposerhop, koja definira vrijednost gustoće na ulazu i izlazu. Za obje zone u ovom slučaju se odabire mod dva koja postavlja vrijednost gustoće svih čestica prema ekstrapolacijskim vrijednostima iz ghost nodova.[1]

6.1.5 Definicija ostalih parametara

Za kraj je potrebno definirati još neke parametre kako bi simulacija bila uspješno provedena. Pa su to sljedeći: parametar SavePosDouble se korsiti za spremanje pozicija čestica korištenjem duple preciznosti. Nadalje jedna od bitnijih parametara je Visco kojim se definira vrijednost viskoznosti. Za slučaj laminarnog strujanja vrijednost je 0.01. TimeMax predstavlja ukupno vrijeme simulacije dok TimeOut predstavlja nakon koliko vremena će se dobiti nova izlazna datoteka. U ovom slučaju vrijeme trajanje simulacije je 1 sekunda i TimeOut je 0.02 sekunde. PartsOutMax predstavlja broj čestica koje mogu biti isključene iz domene prije završetka simulacije. RhopOutMin i RhopOutMax predstavljaju kada će doći do isključenja čestice prema gustoći. Na kraju ostaje naredba simulationdomain sa naredbama posmin i posmax unutar te naredbe. Ona definira veličinu domene koja će se koristiti za simulaciju.[1]

6.2 Mogućnost automatizacije pomoću pythona

Python je programski kod pomoću kojeg će se moći napraviti jednostavnije mijenjanje parametara u xml datoteci i brojne druge stvari. Za samo pisanje python koda koristio se softver Spyder koji daje mogućnost korištenja konzole gdje će se cijeli proces obavljati.

Cilj ovog poglavlja je opisati mogućnost automatizacije na primjeru strujanja fluida oko cilindra, kako bi se cijeli proračun, mogućnost vizualizacije te mijenjanje ulaznih parametara ubrzao kako bi se za kraće vrijeme moglo dobiti više rezultata.

Prvi korak automatizacije je definiranje početnih vrijednosti u slučaju nekih grešaka, kako bi korisnik bio u mogućnosti vratiti vrijednost parametara na prvobitno definirane vrijednosti.

Kao što je prethodno navedeno dio koda sa slike 6.10 će se koristiti u slučaju kada se kod u XML datoteci toliko promjeni da se više ne znaju one početne vrijednosti. Jedan zadatak koji nudi ovaj manji program omogućit će vraćanje svih podataka na tu neku početnu vrijednost.

Za postizanje mogućnosti rada sa XML datotekom, prvo potrebno je datoteku uvesti u python aplikaciju. Na slici 6.9 vidljiv je dio koda koji koji predstavlja uvoz navedene datoteke.

```
6
7   #importanje xml file
8   import sys,  xml.etree.ElementTree as ET
9   tree = ET.parse('CaseFlowCylinder_Re020_Def.xml')
10  root = tree.getroot()
11
```

Slika 6.9 Uvoz xml datoteke u pythonu

Cijela automatizacija se sastoji od čitanja, mijenjanja i vraćanje prvobitnih vrijednosti parametara te pokretanja procesa proračuna.

Brzina, gustoća i viskoznost fluida, vrijeme trajanja simulacije, oblik i dimenzije cilindra te udaljenost između čestica su parametri s čijim se vrijednostima dozvoljava upravljanje pomoću python aplikacije. Slika 6.10 prikazuj dio koda koji se izvršava nakon učitavanja XML datoteke te prikazuje pozivanje prethodno navedenih .

```
27  brzina = root.findall('.//velocity')[1].attrib['v']
28  gustocareff = root.findall('.//rhop0')[0].attrib['value']
29  gustocamax = root.find(".//*[@key='RhopOutMax']").attrib['value']
30  gustocamin = root.find(".//*[@key='RhopOutMin']").attrib['value']
31  timemax = root.find(".//*[@key='TimeMax']").attrib['value']
32  dimenzija = tree.findall('.//drawcylinder')[0].attrib['radius']
33  viskoznost = root.find(".//*[@key='Visco']").attrib['value']
34  oblik = root.find('.//mainlist')[9].tag
35  dp = root.find('.//definition').attrib['dp']
```

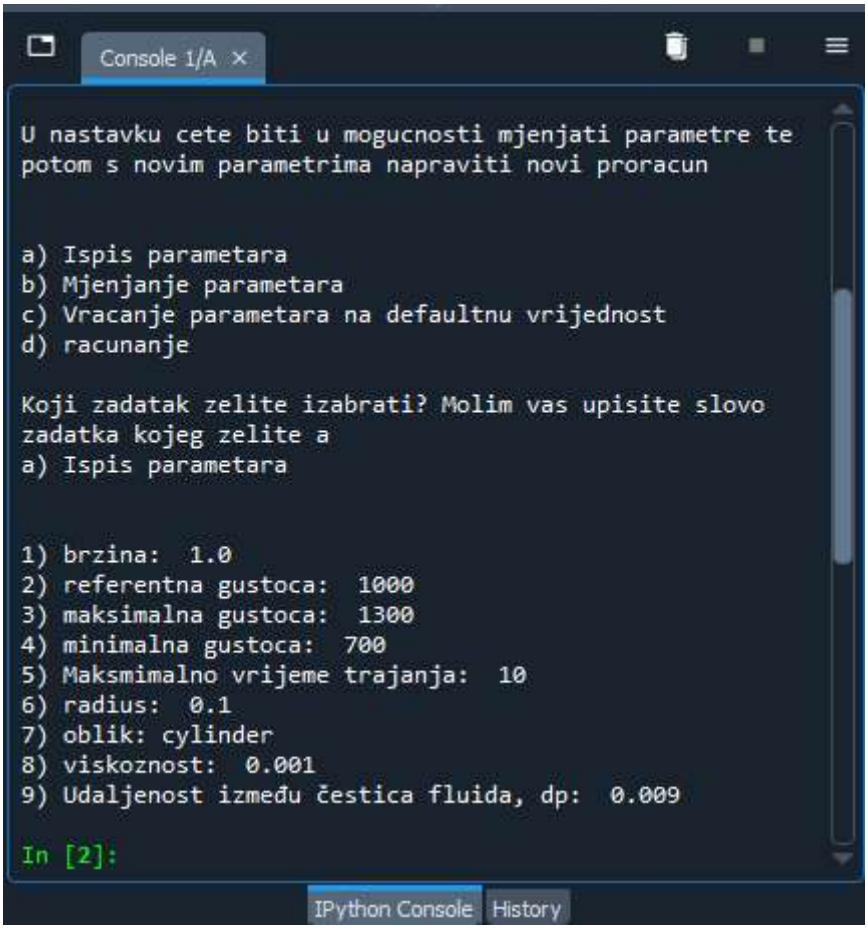
Slika 6.10 Poziv vrijednosti parametara u pythonu

Prilikom pokretanja aplikacije u konzoli, aplikacija nudi korisniku da odabere jednu od četiri mogućnosti. Slika 6.11 prikazuje dio koda koji ovisi o izboru tipa zadatka koji je korisnik odabrao.

```
239 DR1 = input("Koji zadatak zelite izabrati? Molim vas upisite slovo zadatka kojeg zelite ")
240
241 match DR1:
242     case "a":
243         print("a) Ispis parametara \n\n")
244         isp()
245     case "b":
246         print("b) Mjenjanje parametara \n\n")
247         mje()
248     case "c":
249         print("c) Vratiti parametre na defaultnu vrijednost \n\n")
250         vra()
251     case "d":
252         print("d) Računanje \n\n")
253         rac()
254
255     # If an exact match is not confirmed, this last case will be used if provided
256     case _:
257         print( "Niste unijeli ispravan zadatak, molim vas da ponovite")
```

Slika 6.11 Mogućnost odabira 4 tipa zadatka

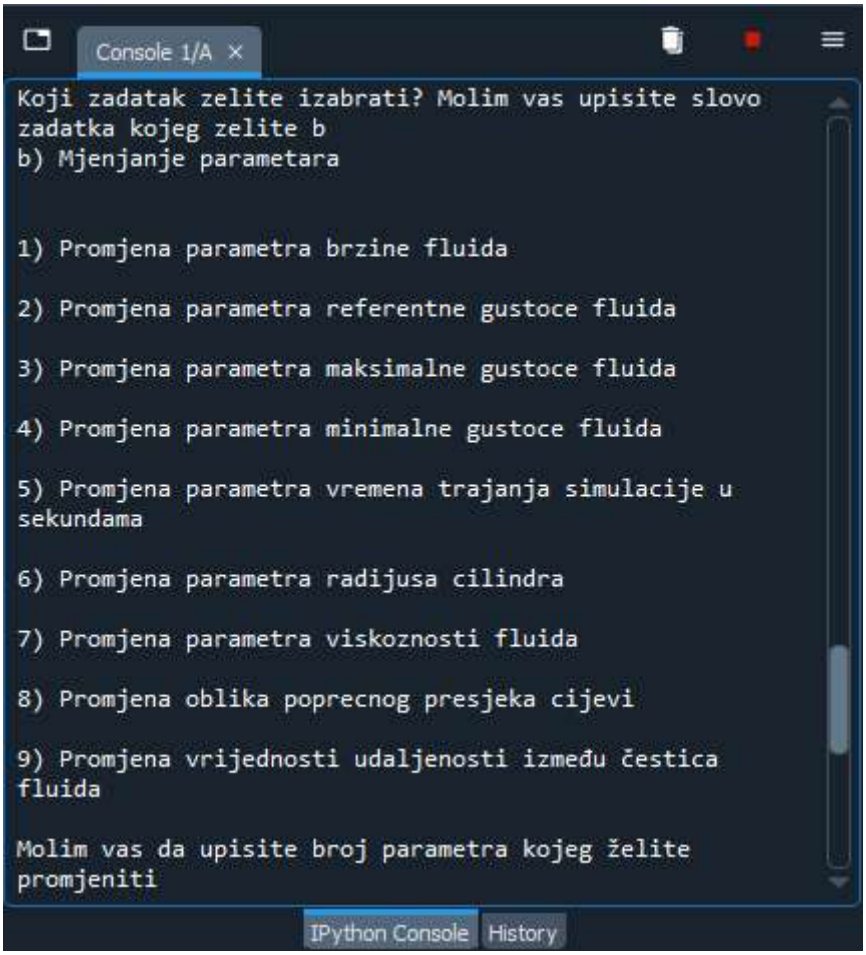
U slučaju ako korisnik odabere zadatak „a“ u konzoli se ispišu svi parametri s kojima aplikacija upravlja. Na slici 6.12 se može vidjeti kako ispis parametara izgleda.



Slika 6.12 Ispis svih parametara

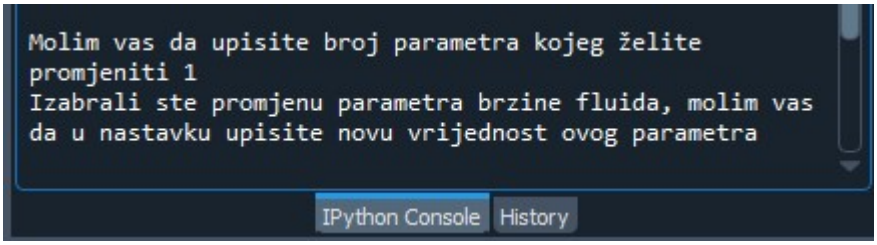
Cilj ovog zadatka je uvid u vrijednosti parametara, kako bi korisnik bio u mogućnosti pokrenuti proračun s tim da zna sa sigurnošću da su svi parametri željene vrijednosti.

Odabirom zadatka „b“ sa slike 6.11, omogućava se mijenjanje parametara. Slika 6.13 prikazuje python konzolu u kojem se nudi mogućnost odabira željenog parametra kojem se treba promijeniti vrijednost.



Slika 6.13 Mogućnost odabira 4 tipa zadatka

Nakon odabira jednog od svih ponuđenih parametara, korisnika se traži unos nove vrijednosti za željeni parametar što je i prikazano slikom 6.14.



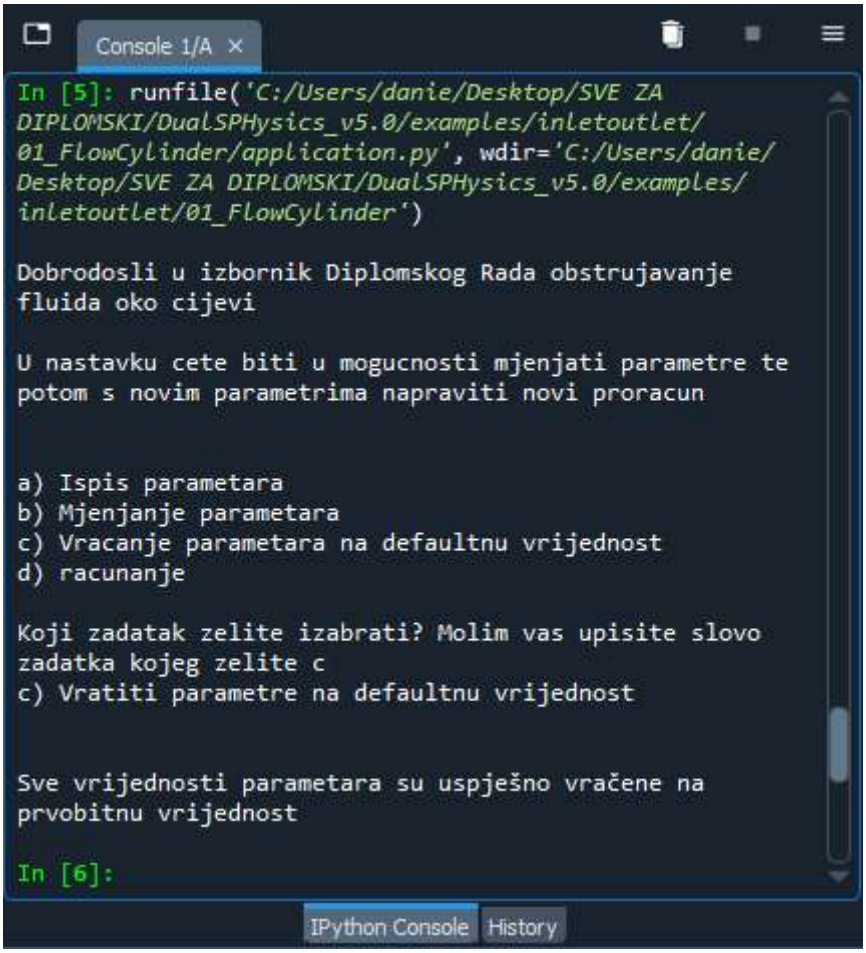
Slika 6.14 Unos nove vrijednosti parametra brzine

Za prikazani slučaj odabrala se mogućnost mijenjanja brzine, gdje je nadalje potrebno samo upisati novu brojčanu vrijednost.

Cilj zadatka „b“ sa slike 6.11 je omogućavanje mijenjanje ulaznih parametara koje je potrebno postaviti prije pokretanja proračuna. Cijeli razlog zbog kojeg se i daje mogućnost mijenjanja parametara je taj da korisnik može napraviti više različitih simulacija sa različitim parametrima

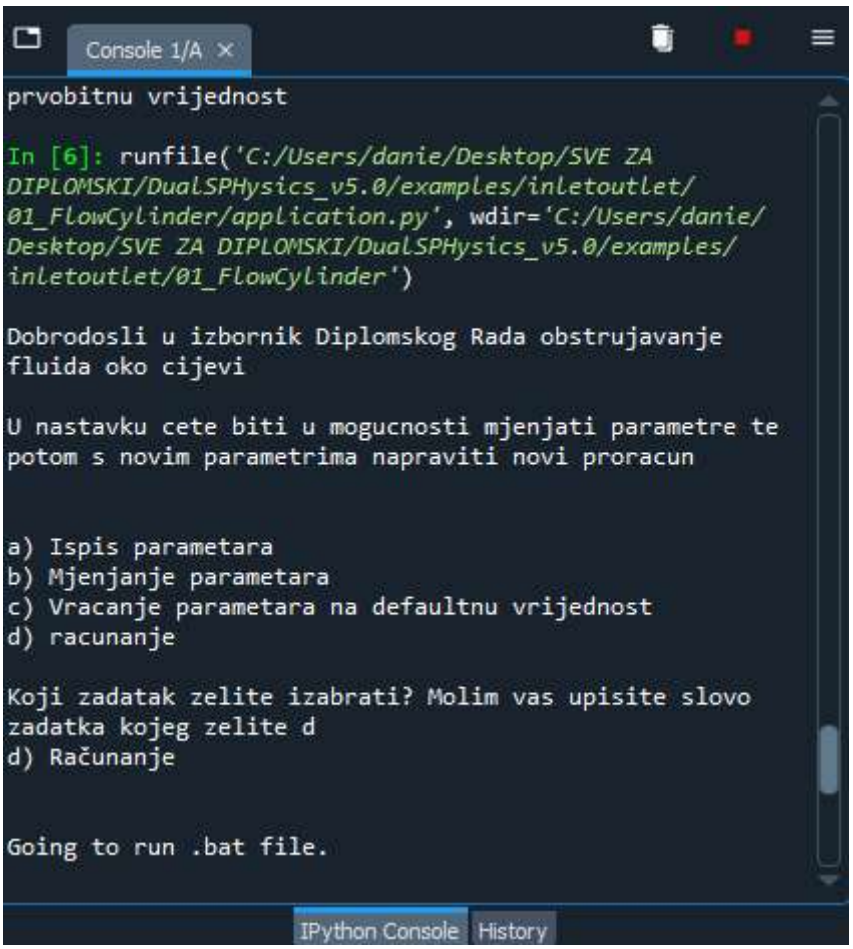
kako bi bio u mogućnosti istražiti i usporediti dobivene rezultate. Osim toga cilj je ubrzati proces promjene parametara tako što u ovom slučaju python aplikacija obavlja brzu promjenu vrijednosti umjesto korisnika.

Ponovnim pregledom slike 6.11 moguće je vidjeti kako sljedeći zadatak pod „c“ predstavlja slučaj vračanja prvobitnih vrijednosti parametara. Slika 6.15 prikazuje python konzolu za slučaj ako se odabere ovaj zadatak.



Slika 6.15 Vračanje vrijednosti parametara na prvobitno zadane vrijednosti

Zadnji zadatak sa slike 6.11 predstavlja proces pokretanja proračuna pomoću DualSPHysics rješavača, gdje slika 6.16 predstavlja prikaz tog zadatka u python konzoli.



Slika 6.16 Pokretanje .bat datoteke pomoću python aplikacije

Nakon što proračun završi, kao rezultat se dobiju izlazne datoteke koje će se koristiti za vizualizaciju u FreeCAD-u.

Cilj ovog zadatka je omogućiti korisniku lakše i brže pokretanje proračuna.

7. REZULTATI

Nakon što je kod u potpunosti objašnjen, sljedeći korak je pokretanje simulacije i prikaz rezultata. Za pokretanje simulacije potrebno je pokrenuti .bat datoteku koja poziva xml datoteku prethodno opisano.

Za prikaz rezultata i ponašanja fluida prilikom opstrujavanja oko cijevi koristit će se par primjera, gdje će prvi primjeri opisivati laminarno strujanje oko cilindra kružnog poprečnog presjeka, dok drugi dio primjera će opisivati turbulentno strujanje oko cilindra kvadratnog poprečnog presjeka.

Cilj ovog poglavlja je opisati dobivene rezultate za svaki primjer te usporediti rezultate za iste slučajeve gdje se vrijednost razmaka između čestica mijenja. Prikazat će se i usporediti vrijeme obavljanja proračuna, ukupan broj čestica koji je prošao kroz domenu i mnogi drugi rezultati. Vrijednost parametara brzine od 1 m/s, trajanja simulacije od 10 sekundi i gustoće od 1000 kg/m^3 su kroz sljedećih par primjera ostali ne promijenjeni dok su se ostale vrijednosti mijenjale.

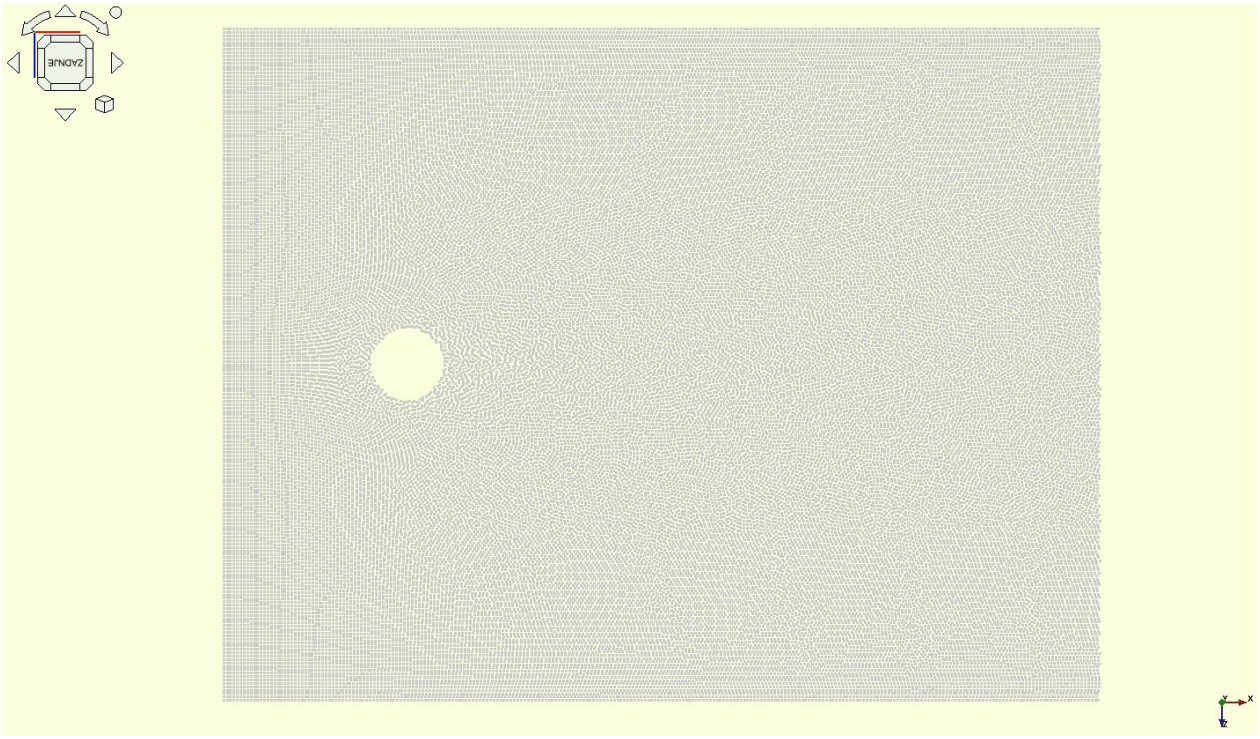
DualSPHysics daje rezultate za svaki vremenski korak koji će iznositi 0.02s za svaki sljedeći primjer, a za prikaz rezultata koristit će se zadnji trenutak.

Za prikaz svih rezultata koristit će se FreeCAD softver koji zahtjeva vtk datoteku u kojoj se nalaze svi potrebni podatci vizualizacije. Nakon prikaza svih rezultata dat će se vlastiti zaključak koliko je FreeCAD softver koristan, a koliko nije za vizualizaciju rješenja.

7.1 Laminarno strujanje fluida oko cilindra kružnog poprečnog presjeka

U nastavku bit će opisan slučaj laminarnog strujanja fluida oko cilindra kružnog poprečnog presjeka. Vrijednost reynoldsovog broja koja nam opisuje koja je vrsta strujanja u pitanju, izračunava se prema formuli (6.1). Reynoldsov broj je bezdimenzijska vrijednost te ona nema mjerne jedinice. Kada se ne bi znalo sa sigurnošću koliko iznosi vrijednost graničnog reynoldsovog broja, moguće je napraviti par simulacije za različite Re te provjeriti radi li se o laminarnom ili turbulentnom strujanju tako što ako postoji vrtloženje i miješanje fluida, onda se tu radi o turbulentnom strujanju, a ako se fluid ne miješa i struji bez vrtloga i samo u jednom smjeru, onda se sa sigurnošću može reći da se radi o laminarnom strujanju.

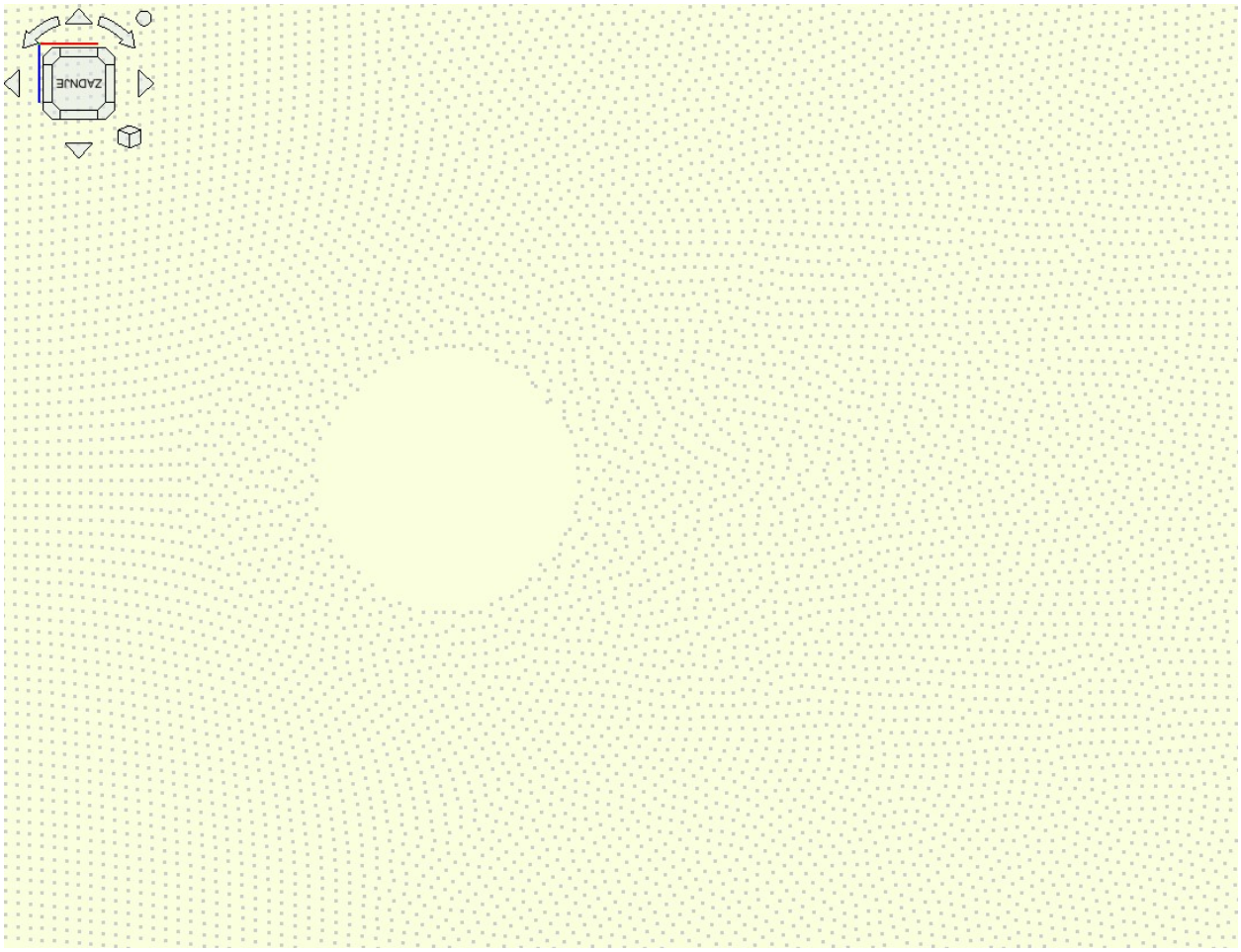
Na slici 7.1 moguće je vidjeti domenu ispunjenom česticama koje su međusobno udaljene jedna o druge u ovom slučaju 0.01m.



Slika 7.1 Domena ispunjena česticama za slučaj strujanja fluida oko cilindra kružnog poprečnog presjeka

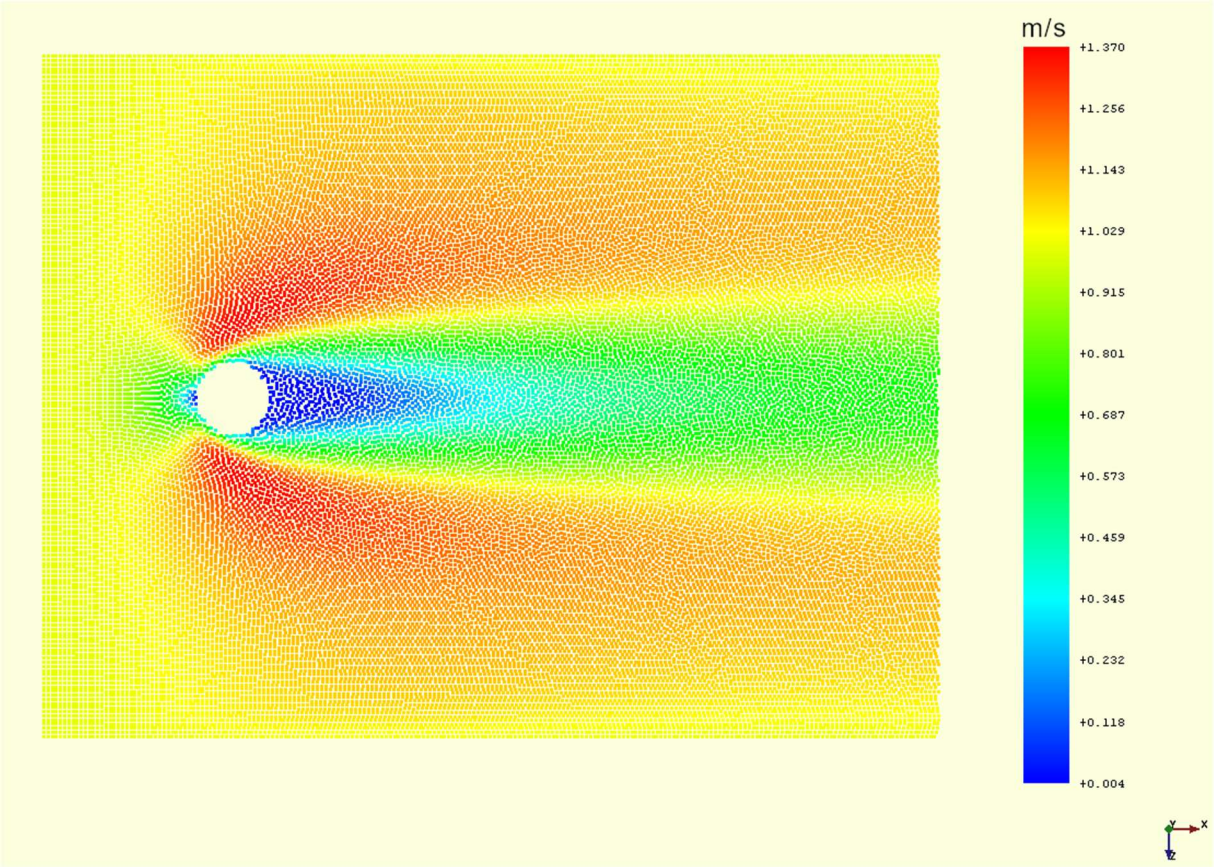
Čestice prikazane na slici 7.1 predstavljaju čestice fluida koje se gibaju kroz domenu. Broj čestica ovisi o dimenziji dp koja predstavlja udaljenost između čestica čija je vrijednost definirana u xml datoteci. Što je dp manji broj čestica je veći što omogućava kvalitetniji prikaz. Isto tako smanjenjem vrijednosti dp, vrijeme potrebno rješavaču da obavi proračun se povećava ali paralelno s tim kvaliteta vizualizacije opada.

Moguće je vidjeti da raspored čestica nije pravilno raspoređen u stupcima i redcima. Slika 7.2 prikazuje nepravilnu raspodjelu čestica oko cilindra. Razlog tome je to što nailazi prepreka pa DualSPHysics diskretizira čestice na najbolji način koji će dati najbolja rješenja s najmanje grešaka.

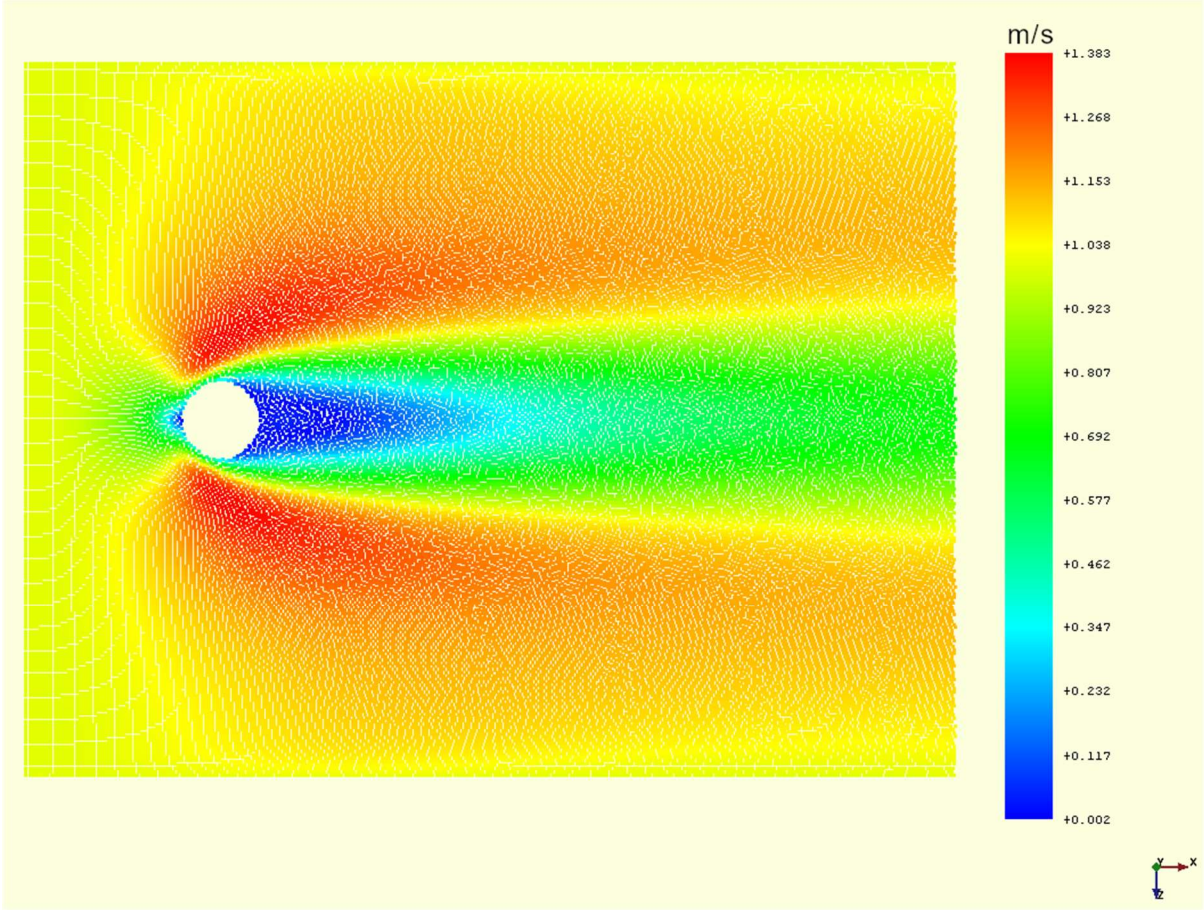


Slika 7.2 Raspored čestica oko cilindra kružnog poprečnog presjeka

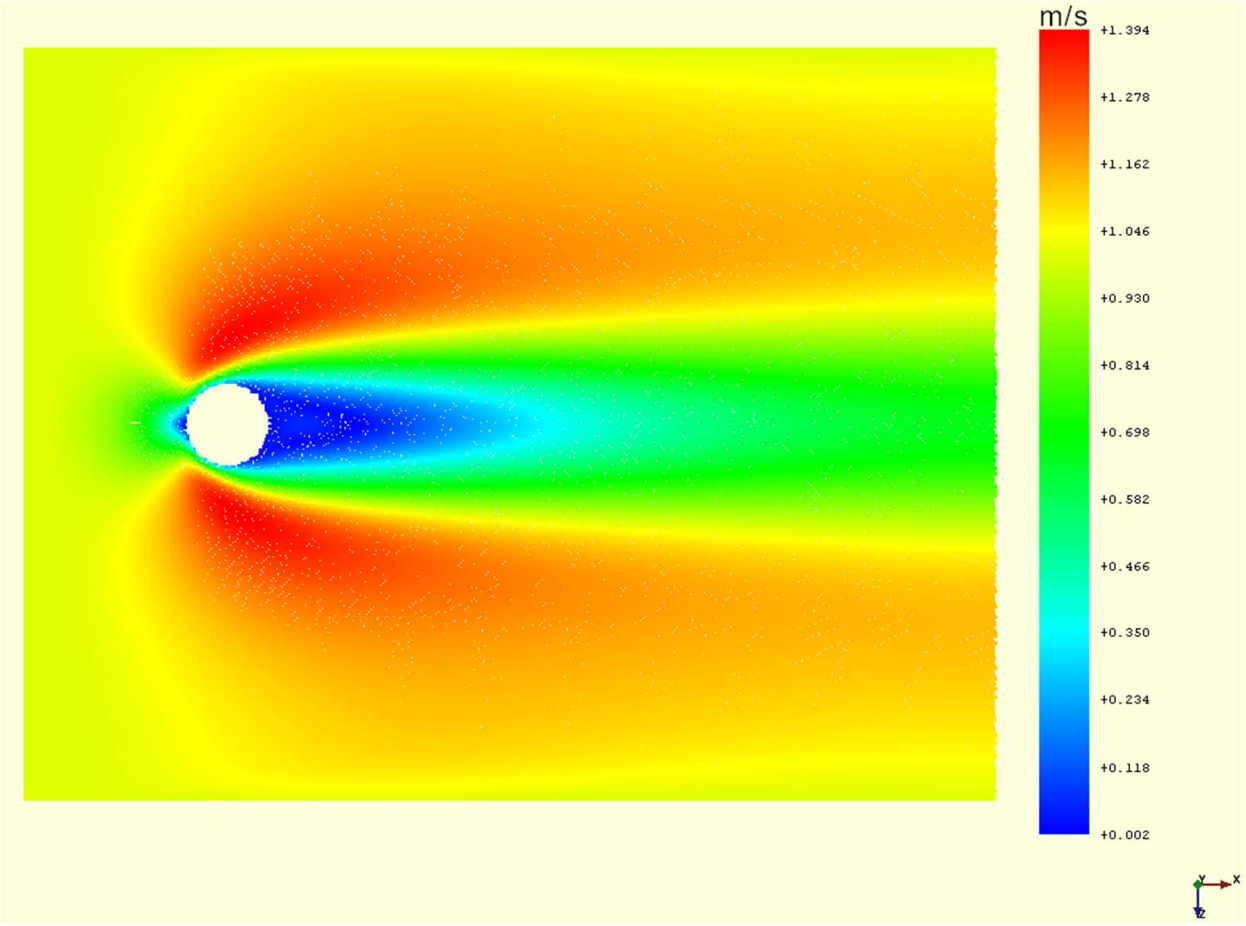
Za slučaj laminarnog strujanja fluida oko cilindra kružnog poprečnog presjeka koristila se viskoznost u vrijednosti od 0.01 m^2/s. Polumjer cilindra iznosi 0.1m. U nastavku će se opisati i usporediti tri tipa strujanja, gdje se za svaki slučaj mijenja vrijednost udaljenosti između čestica (dp). U jednom primjeru vrijednost udaljenosti iznosi 0.008m, u drugom primjeru iznosi 0.01m dok u trećem primjeru ta vrijednost udaljenosti čestica iznosi 0.012m.



Slika 7.3 Prikaz vrijednosti brzina laminarnog strujanja za slučaj dp=0.012m



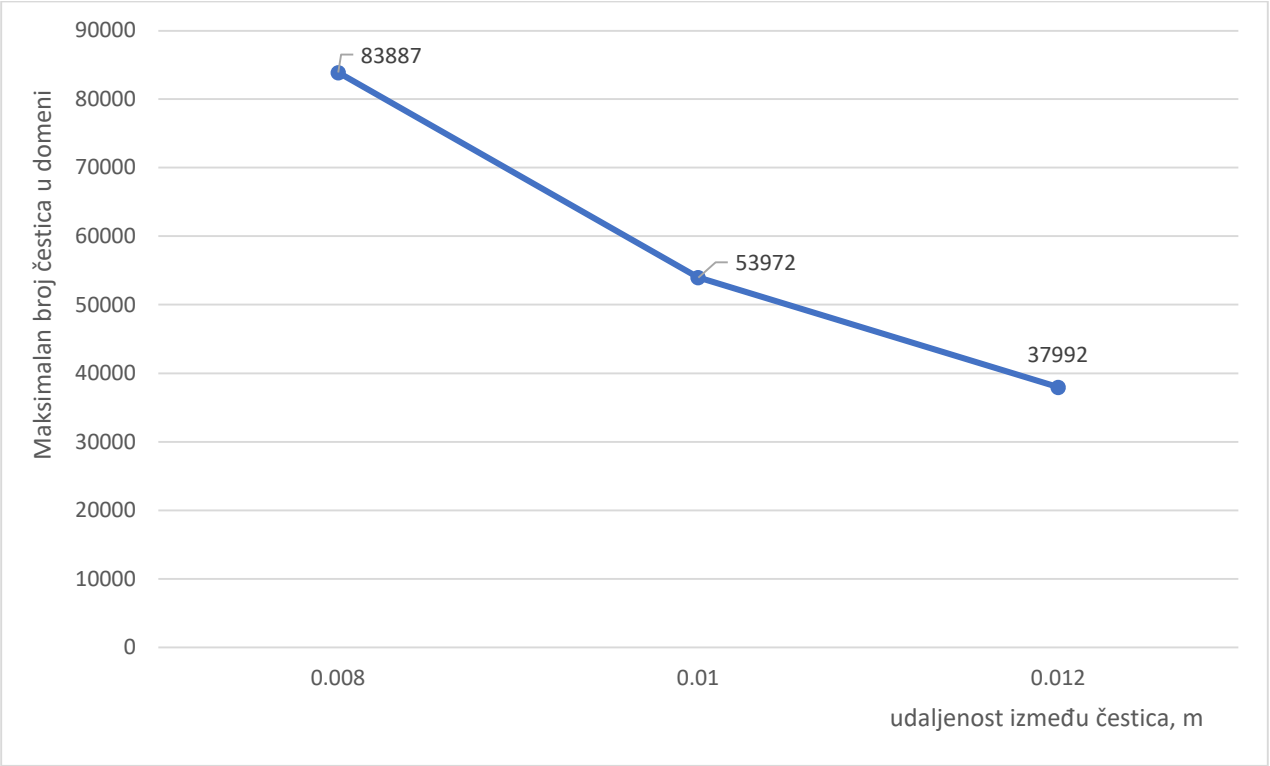
Slika 7.4 Prikaz vrijednosti brzine laminarnog strujanja za slučaj dp=0.01



Slika 7.5 Prikaz vrijednosti brzine laminarnog strujanja za slučaj dp=0.008

Slika 7.3, 7.4 i 7.5 prikazuju strujanje fluida oko cilindra u trenutku t=10 sekundi. Prvo što je za primijetiti je kvaliteta rezolucije. Kao što je prethodno već spomenuto razmak između čestica predstavlja koliki će biti ukupan broj čestica u domeni. Vidljivo je na slici 7.5 kada je vrijednost dp=0.008 da se dobije najkvalitetnija rezolucija dok na slici 7.3 kvaliteta je najgora. Osim toga bitno je uočiti da vrijednosti najvećih brzina nije ista za sva 3 slučaja iako su svi parametri ostali isti, samo broj čestica je drugačiji.

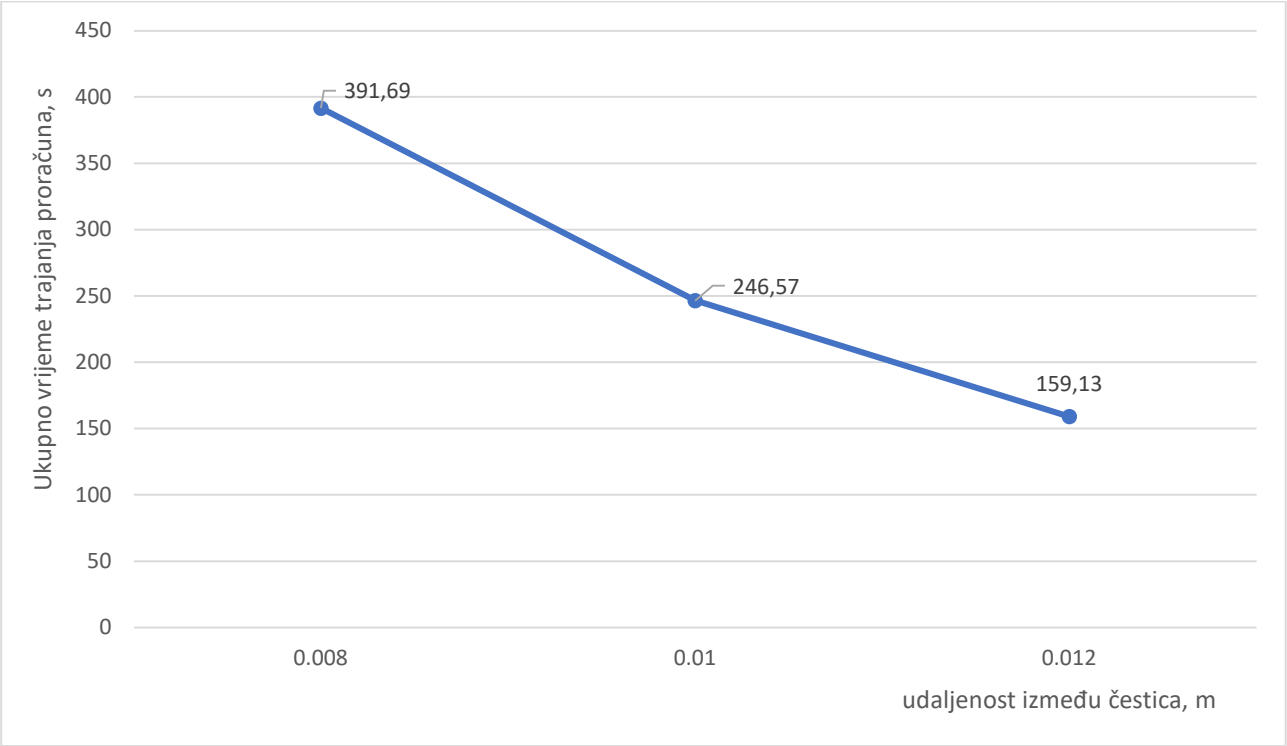
Može se također zaključiti da nema miješanja i vrtloženja već kontinuirano gibanje u jednom smjeru što dokazuje da je ova vrsta strujanja laminarno strujanje. Korištenjem formule 6.1 vrijednost Re iznosi 20 i stoga se može reći da je ova vrijednost reynoldsovog broja definitivno ispod granične vrijednosti.



Slika 7.6 Promjena maksimalnog broja čestica u domeni u ovisnosti o dp za slučaj laminarnog strujanja oko cilindra kružnog poprečnog presjeka

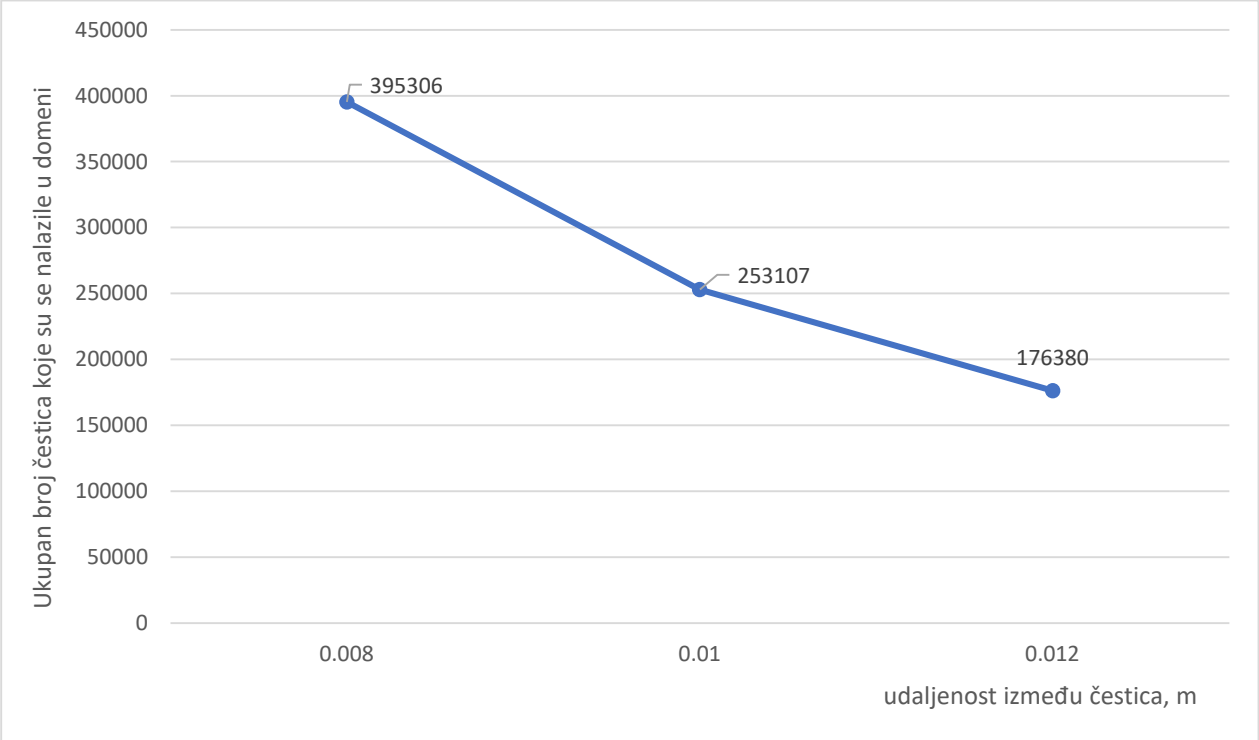
Promjena maksimalnog broja čestica u domeni u ovisnosti o dp za slučaj laminarnog strujanja oko cilindra kružnog poprečnog presjeka

Na slici 7.6. moguće je vidjeti graf gdje za najmanje vrijednosti dp, postižu se maksimalne vrijednosti broja čestica koji se može pojaviti u domeni. Broj čestica u domeni će biti minimalan onda kada je dp najveći.



Slika 7.7 Promjena ukupnog vremena trajanja proračuna u ovisnosti dp za slučaj laminarnog strujanja oko cilindra kružnog poprečnog presjeka

Na slici 7.7 moguće je vidjeti graf promjene ukupnog vremena trajanja proračuna u ovisnosti o dp za slučaj laminarnog strujanja oko cilindra kružnog poprečnog presjeka. To jest pokazuje koliko je vremenski bilo potrebno da DualSPHysics na GPU obradi cijeli proračun. Što je i za zaključiti da što je manja udaljenost između čestica, tj. što je veći broj čestica u domeni, potrebno je više vremena da se završi proračun



Slika 7.8 Promjena ukupnog broja čestica koje su se nalazile u domeni u ovisnosti o dp za slučaj laminarnog strujanja oko cilindra kružnog poprečnog presjeka

Slika 7.8 prikazuje graf promjene ukupnog broja čestica koje su se nalazile u domeni u ovisnosti o dp za slučaj laminarnog strujanja oko cilindra kružnog poprečnog presjeka

Vidljiva je promjena broja čestica koje su se nalazile u domeni čija vrijednost ovisi o dp. Slično kao i kod slike 7.6, što je udaljenost između čestica manja, veći je broj čestica u domeni, sukladno tome što je udaljenost veća broj čestica je manji.

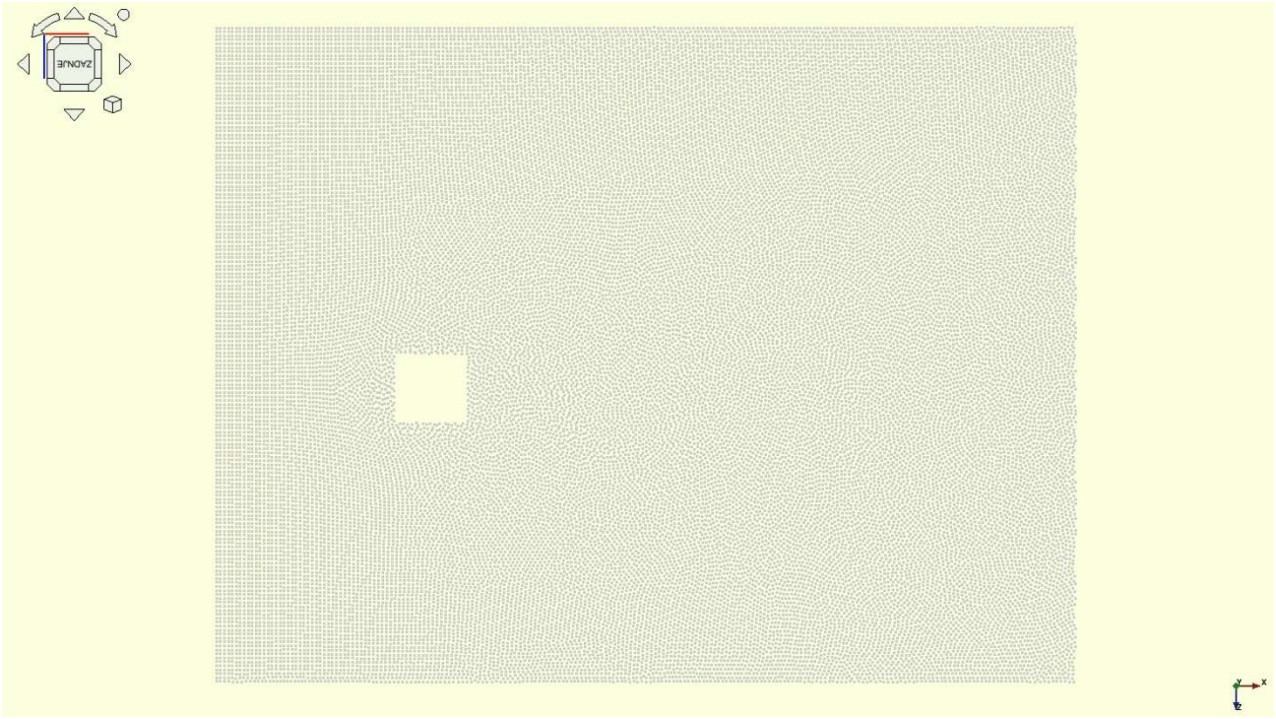
Za zaključiti je da količina čestica uveliko utječe na rezolucija, a isto tako i na vrijednosti brzina. Osim toga u sva tri grafikona se dobiju funkcije koje izgledaju slično, što ukazuje na to da proporcionalnim povećavanjem dp, vrijeme potrebno za proračun, ukupan broj čestica te maksimalan broj čestica se proporcionalno smanjuje.

7.2 Turbulentno strujanje fluida oko cilindra kvadratnog poprečnog presjeka

U ovom poglavlju bit će opisan slučaj turbulentnog strujanja fluida oko cilindra kvadratnog poprečnog presjeka. Za slučaj kada se ne zna granična vrijednost reynoldsovog broja, potrebno je napraviti par simulacija te kad dođe slučaj da u fluidu ima vrtloženja i miješanja, može se reći kako u tom primjeru fluid struji turbulentno.

Za prikaz rezultata koristit će se 3 primjera, te će bit prikazan na sličan princip kako su rezultati prikazani kod laminarnog strujanja fluida oko cilindra kružnog poprečnog presjeka.

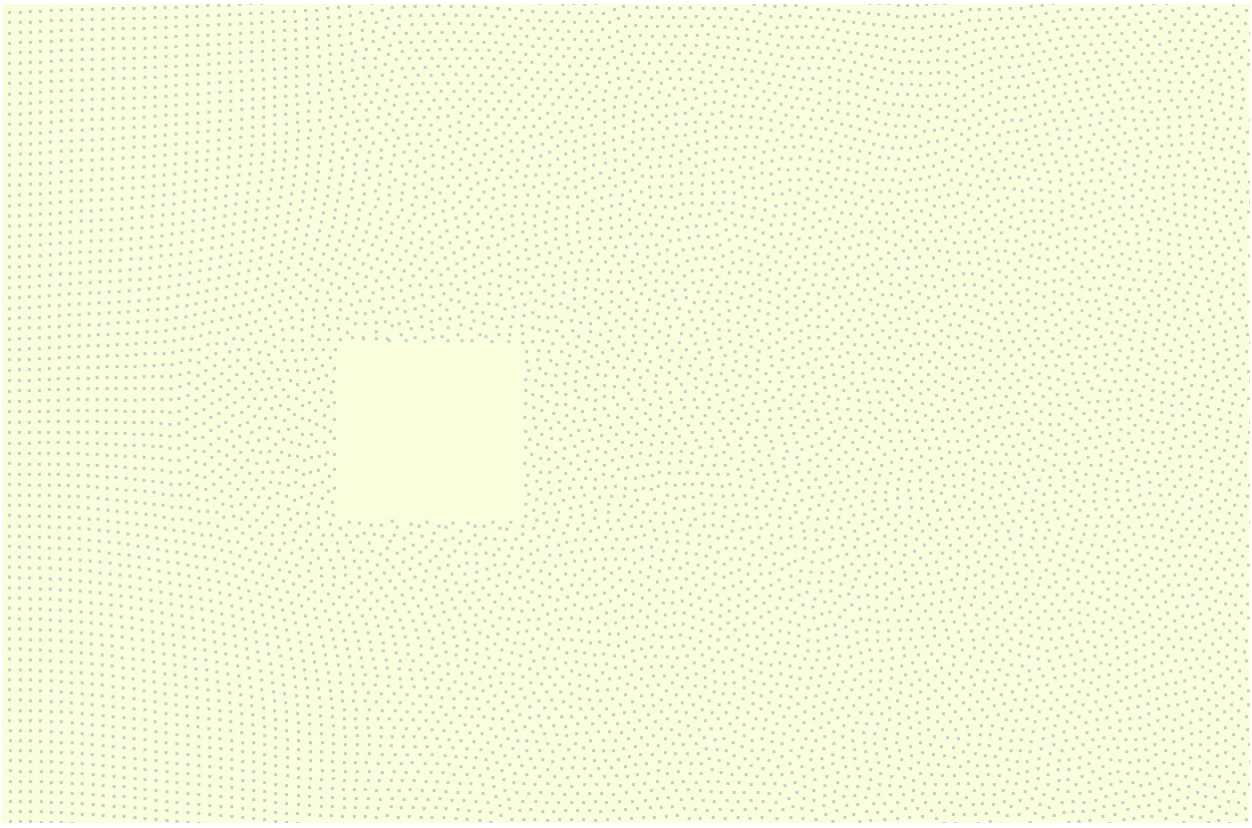
Na slici 7.6 moguće je vidjeti domenu ispunjenom česticama koje su međusobno udaljene jedna o druge u ovom slučaju 0.012m.



Slika 7.9 Domena ispunjena česticama za slučaj strujanja fluida oko cilindra kvadratnog poprečnog presjeka

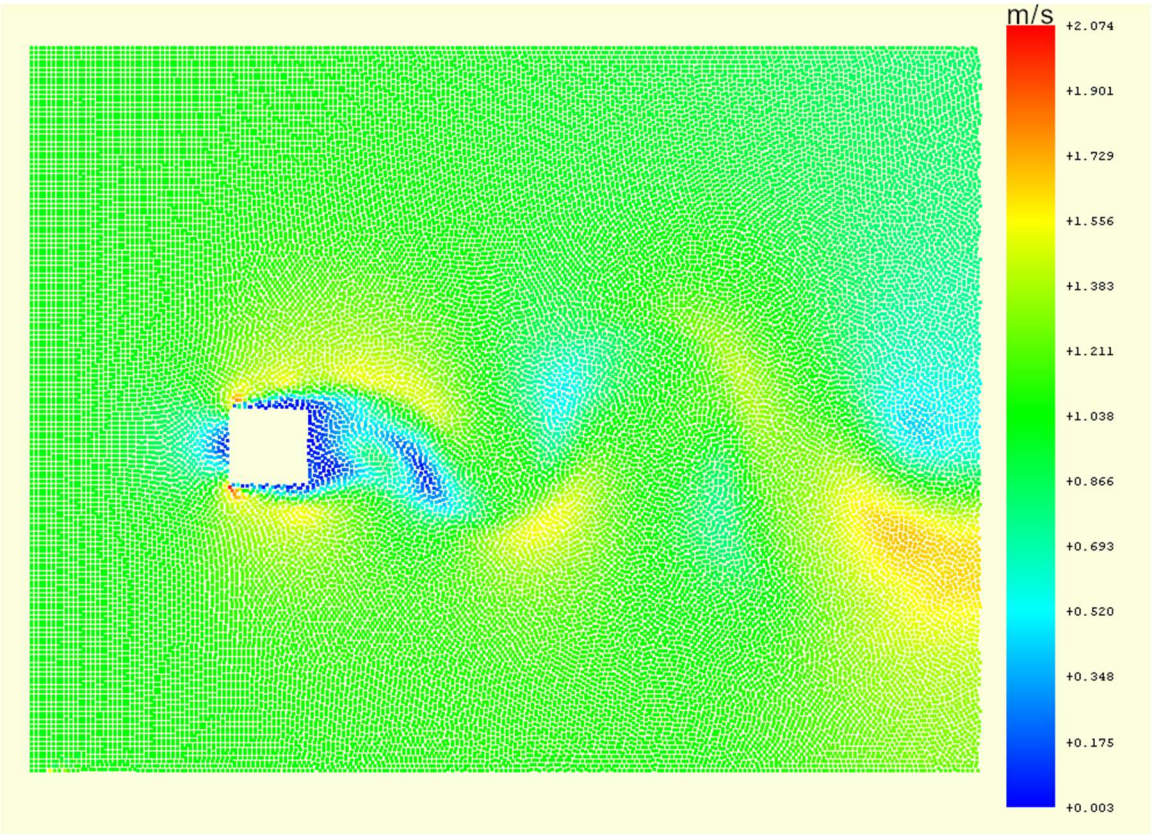
Kao što je prethodno već definirano što je veći razmak između čestica, manji će biti broj čestica. Za ovaj slučaj sa slike 7.9 vidljivo je kako je mreža puno rjeđa što bi rezultiralo u simulacijama lošijom razlučivosti.

Osim toga vidljivo je kako i ovdje raspored čestica nije pravilan što je vidljivo na slici 7.10.

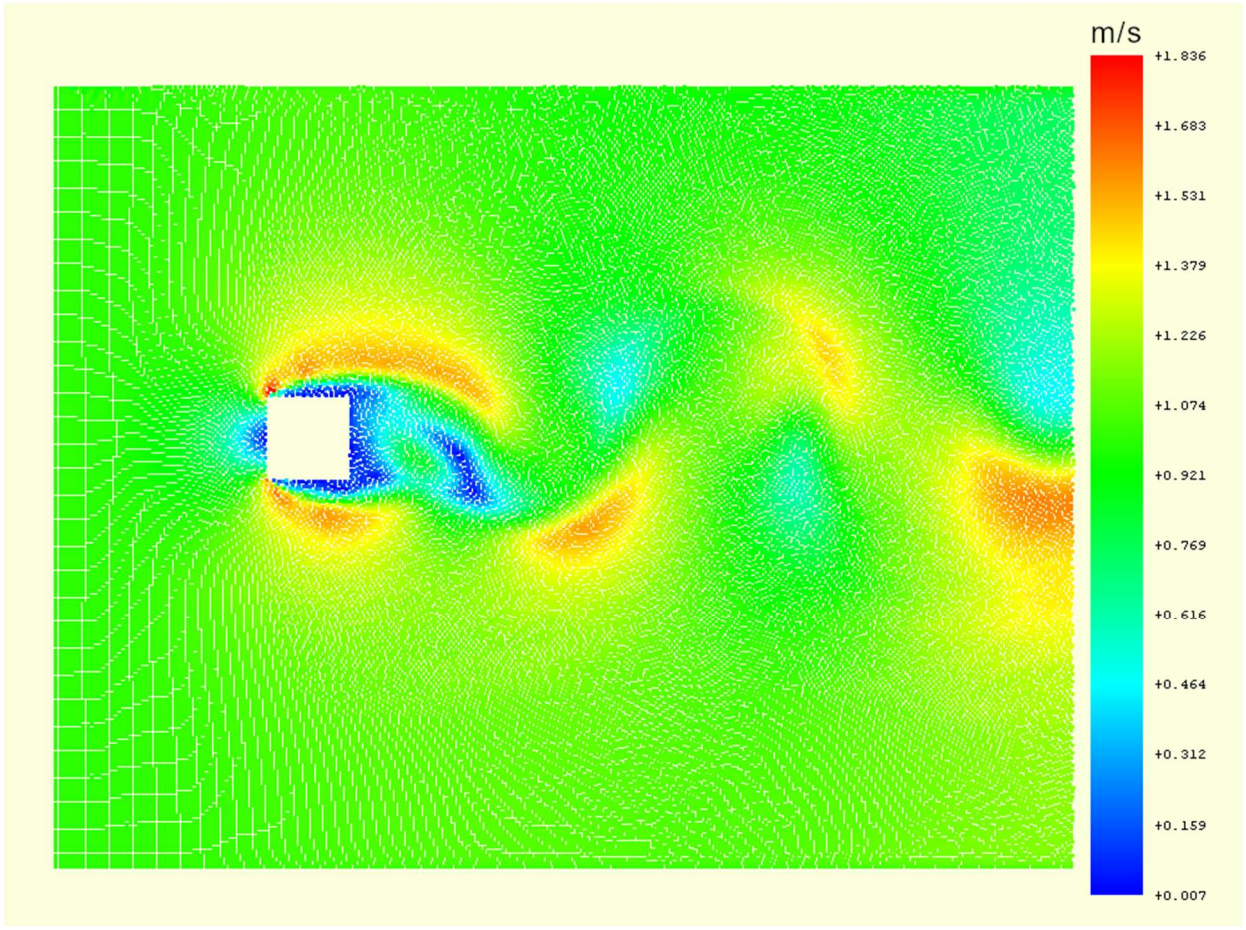


Slika 7.10 Raspored čestica oko cilindra kvadratnog poprečnog presjeka

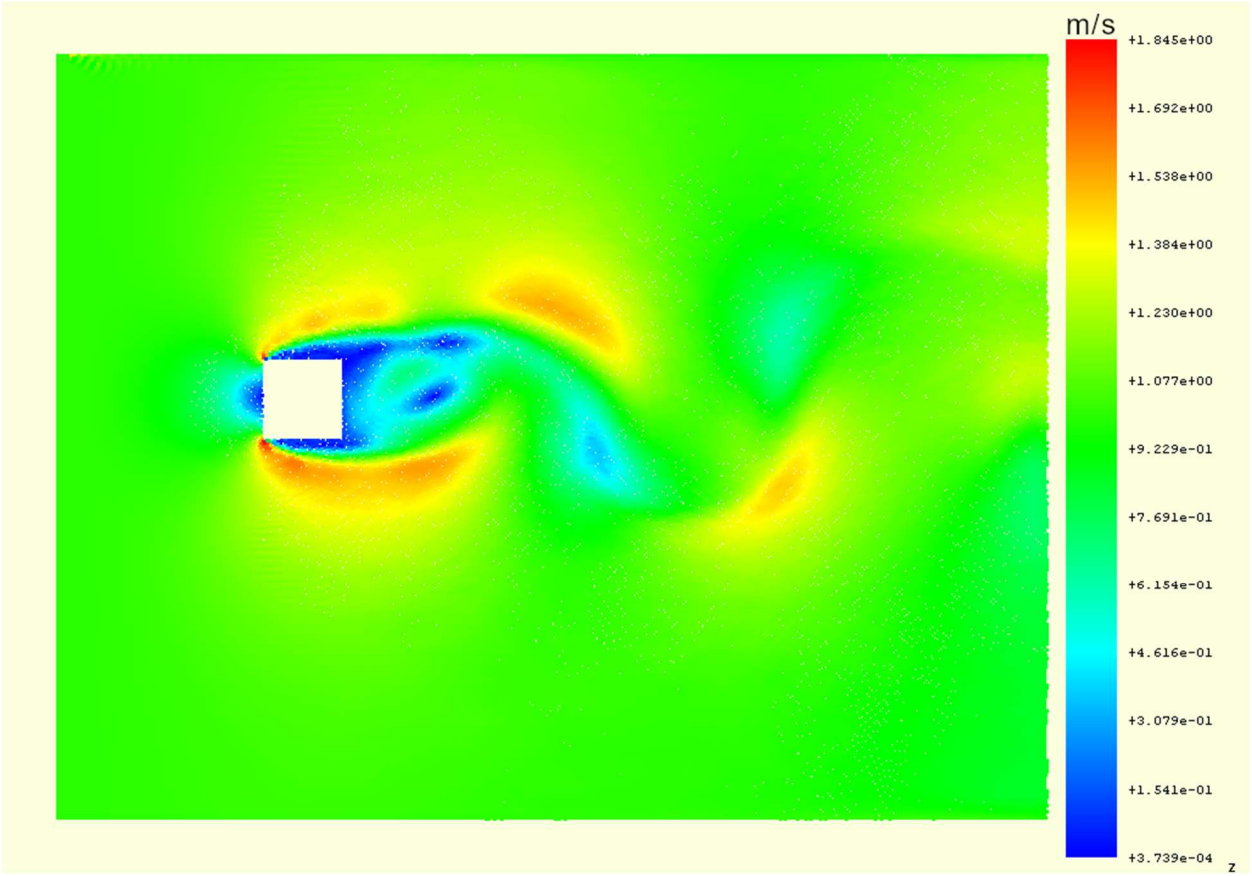
Za slučaj turbulentnog strujanja fluida oko cilindra kvadratnog poprečnog presjeka koristila se viskoznost u vrijednosti od 0.001 m^2/s. Dužina jedne stranice iznosi 0.2m.



Slika 7.11 Prikaz vrijednosti brzina turbulentnog strujanja za slučaj dp=0.012m



Slika 7.12 Prikaz vrijednosti brzina turbulentnog strujanja za slučaj dp=0.01

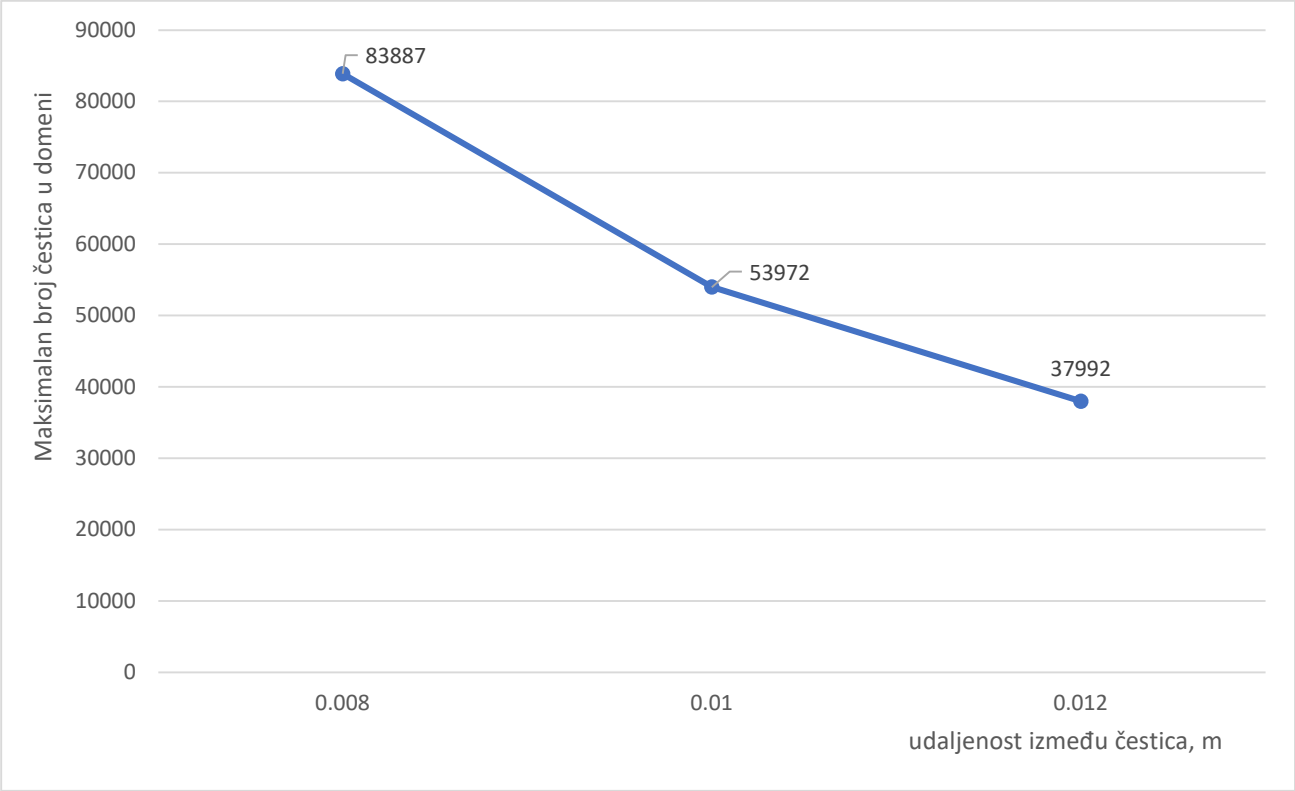


Slika 7.63 Prikaz vrijednosti brzina turbulentnog strujanja za slučaj dp=0.008m

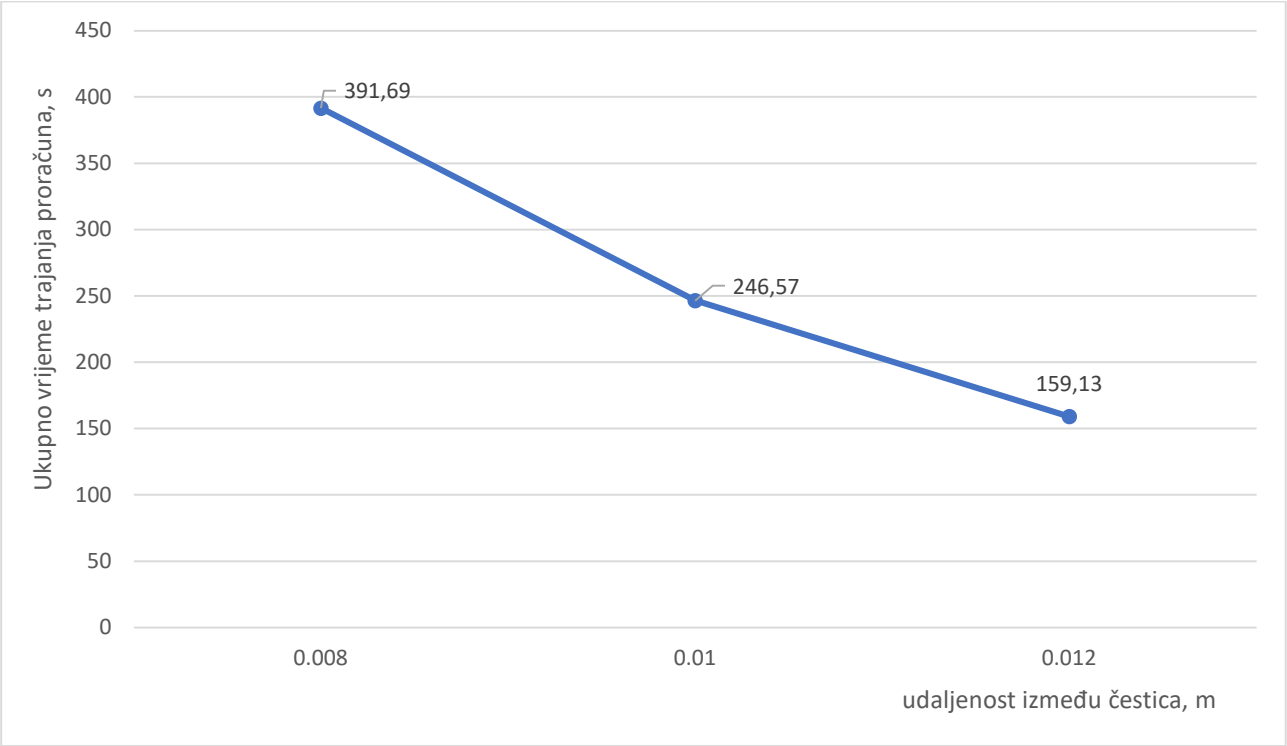
Slika 7.11, 7.12 i 7.13 prikazuju strujanje fluida oko cilindra u trenutku t=10 sekundi. Ovdje je za primijetiti kako fluid kod velike gustoće mreže u potpunosti drugačije struji nego što je to slučaj kada je mreža rjeđa.

Za zaključit je, a i vidljivo je iz prethodnih slika kako dolazi do miješanja i vrtloženja fluida. To ukazuje na to da je ovaj tip strujanja turbulentan

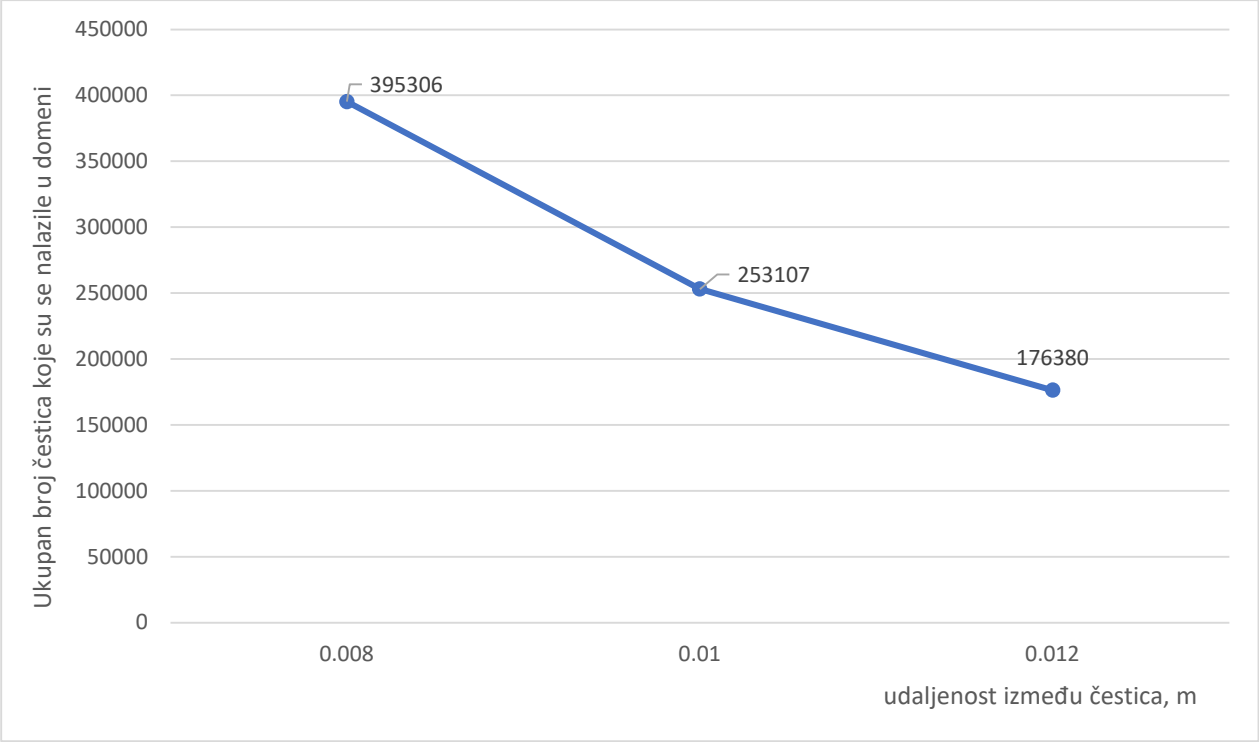
Nakon vizualnog prikaza rezultata strujanja, potrebno je kao i u prethodnom primjeru pronaći karakteristične podatke koje opisuju proračun te ih spremiti u grafikone i na kraju usporediti.



Slika 7.14 Promjena maksimalnog broja čestica u domeni u ovisnostio dp za slučaj turbulentnog strujanja oko cilindra kvadratnog poprečnog presjeka



Slika 7.15 Promjena ukupnog vremena trajanja proračuna u ovisnosti o dp za slučaj turbulentnog strujanja oko cilindra kvadratnog poprečnog presjeka

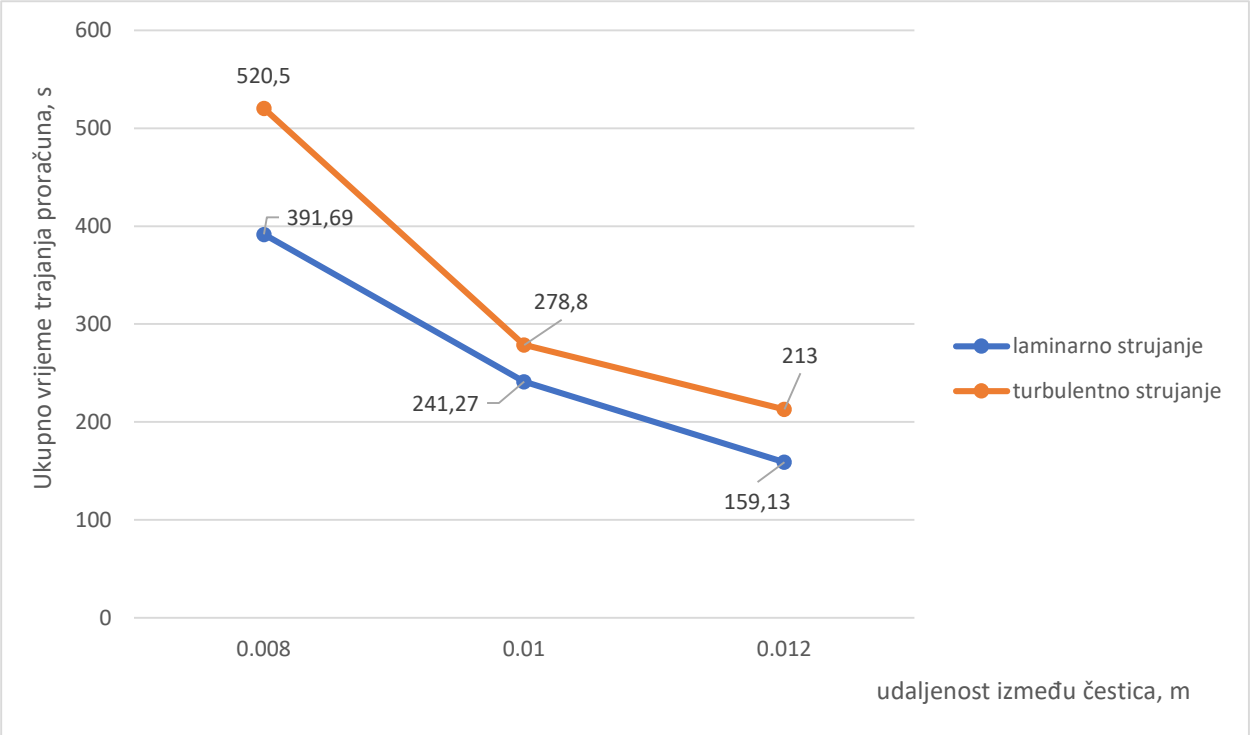


Slika 7.16 Promjena ukupnog broja čestica koje su se nalazile u domeni u ovisnosti o dp za slučaj turbulentnog strujanja oko cilindra kvadratnog poprečnog presjeka

Vidljivo je kako graf sa slike 7.6 i 7.14 isto kao i graf sa slike 7.8 i 7.16 identični. Razlog je taj što vrijednost dp je ostao isti za sve primjere i u laminarnog i u turbulentnog strujanju, a vrijednosti

dp nisu dovoljno male da bi mogle stvoriti novu česticu na mjestu gdje se pojavljuje rupa koje nastaju zbog promjene oblika.

Grafikon koji predstavlja ukupno vrijeme proračuna simulacije u ovisnosti o dp je za turbulentno strujanje malo drugačiji, to jest rješavaču je potrebno malo duže da obradi slučajeve kada se pojavi turbulentno strujanje.



Slika 7.17 Promjena ukupnog vremena trajanja proračuna u ovisnosti o dp za slučaj turbulentnog i laminarnog strujanja oko cilindra

Na slici je moguće vidjeti kako je vrijeme trajanja simulacije za laminarno strujanje puno kraće. Razlog je u tome što rješavač ne mora rješavati kompleksne kretnje čestica kod laminarnog strujanja, već čestice se uglavnom pravocrtnog kretanja koje se lako računaju, dok kod turbulentnog strujanja čestice se mogu naći i u vrtlogu što komplicira proračun.

7.3 Prednosti i mane FreeCAD softvera

FreeCAD softver ne daje puno mogućnosti prilikom vizualizacije u usporedbi sa Ansys ili paraview softverima koje koriste metode s mrežama. Iako FreeCAD nudi prikaz tlakova i gustoće, kvaliteta tih prikaza nije zadovoljavajući, dok softveri koji koriste mreže, lako mogu prikazati sva rješenja na ploči u obliku vektora, strujnica, raznih boja te prikazati sva moguća rješenja od vrijednosti tlakova, brzina, gustoće i svega ostalog.

Glavni razlog zašto FreeCAD u nekim slučajevima dominira je taj što je on besplatan dok softveri kao što je ansys su skupi teško prihvatljive privatnim ljudima. Iako je FreeCAD u razvoju, puno zastajkuje iza prethodno navedenih softvera.

Za zaključit je da FreeCAD softver nije loše koristiti u slučajevima nekih malih projekata ili kada se ima potrebe za učenjem, dok za ozbiljnije velike projekte, gdje se ne smije dozvoliti greška, još uvijek vladaju softveri kao što su Ansys, Paraview i tako dalje.

8. ZAKLJUČAK

U radu je prikazana primjena računalne dinamike fluida na metodi hidrodinamike izglađenih čestica. Za prikaz rješenja koristilo se par primjera zadataka, laminarno strujanje fluida oko cilindra kružnog poprečnog presjeka i turbulentno strujanje fluida oko cilindra kvadratnog poprečnog presjeka.

Za potrebe izvođenja proračuna koristio se DualSPHysics rješavač, koji nudi razne primjere, gdje se među brojnima i nalazi primjer strujanja fluida oko cilindra. Jedan od ciljeva ovog rada je prikazati mogućnost vizualizacije rješenja, pa se za te potrebe koristio FreeCAD softver. Kako DualSPHysics rješavač nema grafičko korisničko sučelje, u sklopu programa FreeCAD postoji modul koji omogućava grafičku podršku prilikom izrade simulacija. Upravo iz tog razloga je navedeni softver korišten za ispitivanje mogućnosti vizualizacije.

DualSPHysics je sam po sebi kompleksan pogotovo za početnike, pa je cilj ovog rada bio omogućiti lakše korištenje rješavača bez potrebe prethodnog poznavanja kako radi rješavač. Da bi se omogućilo tako nešto bilo je potrebno napraviti manju aplikaciju u pythonu koja omogućava automatizirano i lakše mijenjanje ulaznih parametara koji odlaze u rješavač i FreeCAD. Postoji još jako puno mjesta za napredak gdje se može korisniku omogućiti rad s rješavačem DualSPHysics. S obzirom da navedeni rješavač ne zahtijeva izradu numeričkih mreža može predstavljati dobro rješenje za probleme koji zahtijevaju dugotrajnu izradu kompleksnih mreža. Navedeni rješavač se kontinuirano nadograđuje te je za očekivati da će s većom korisničkom podrškom, grafičkim sučeljem i mogućnostima vizualizacije biti sve zastupljeniji alat.

9. LITERATURA

[1] „From fluid dynamics to multiphysics problems“, s interneta, https://dual.sphysics.org/, 17.08.2023.

[2] Alejandro J. C. Crespo: „DualSPHysics“, s interneta, https://github.com/DualSPHysics/DualSPHysics/wiki/, 21.08.2023.

[3] Alejandro J. C. Crespo; Jose M. Dominguez; Moncho Gomez-Gesteira: „Optimization strategies for CPU and GPU implementations of a smoothed particle hydrodynamics method“, s interneta, https://www.sciencedirect.com/science/article/abs/pii/S001046551200358X, 25.08.2023.

[4] „Getting started“, s interneta, https://code.visualstudio.com/docs, 26.08.2023.

[5] Cagle, K.; Dix, C.; Hunter, D.; Kovack, R.; Pinnock, J.; Rafter, J.: „Beginning XML, 2nd Edition“, Wiley Publishing, Inc., pp. 11-22, 2003.

[6] „The Python Wiki“, s interneta, https://wiki.python.org/moin/FrontPage, 29.08.2023.

[7] „Getting started“, s interneta, https://wiki.freecad.org/Getting_started, 30.08.2023.

[8] „DesignSPHysics“, s interneta, https://github.com/DualSPHysics/DesignSPHysics, 01.09.2023.

[9] Verma, G.; Weber, M.: „FreeCAD 0.20 Black Book“, CADCAMCAE WORKS, pp. 1-10, 2022.

[10] „The Scientific Python Development Environment“, s interneta, https://anaconda.org/anaconda/spyder, 03.09.2023.

[11] Liu, G. R.; Liu, M. B.: „Smoothed Particle Hydrodynamics: A Meshfree Particle Method“, World Scientific Publishing Co. Pte. Ltd, pp. 1-31, 2003.
https://books.google.hr/books?id=_cwFMmEQvZQC&printsec=frontcover&hl=hr&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

SAŽETAK

U ovom radu se razmatra i ispituje mogućnost računalne simulacije dinamike fluida koja se temelji na metodi hidrodinamike izglađenih čestica. DualSPHysics omogućava korištenje ove metode bez potrebe izrade numeričke mreže što ujedno olakšava i ubrzava proces simulacije. Domena se sastoji od velikog broja čestica gdje se prate stanja svake čestice što i karakterizira navedenu metodu. DualSPHysics je rješavač koji ne nudi mogućnost vizualizacije, stoga je za tu potrebu korišten FreeCAD softver.

U radu se također ispituje i mogućnost automatizacije rješavača s ciljem omogućavanja njegova sslakšeg korištenja. Opisano je strujanje fluida oko cilindra, te su za taj primjer prikazani rezultati gdje su mijenjani ulazni parametri kao što su brzina, oblik cilindra, viskoznost, veličina cilindra i tako dalje.

Ključne riječi: DualSPHysics, SPH, FreeCAD, strujanje fluida oko cijevi, python, XML, laminarno strujanje, turbulentno strujanje, automatizirana promjena ulaznih parametara

ABSTRACT

In this paper, the possibility of computer simulation of fluid dynamics based on the Smoothed-particle hydrodynamics method is considered and tested. DualSPHysics enables the use of this method without the need to create numerical grid, which at the same time facilitates and speeds up the simulation process. The domain consists a large number of particles where the states of each particle are monitored, which is what caracterizes the mentioned method. DualSPHysics is a solver that does not offer the possibility of visualization, therefore FreeCAD software was used for this purpose. The paper also examines the possibility of automating the solver with the aim of making it easier to use. The fluid flow around the cylinder is described, and for this example the results are shown where the parameters such as speed, shape of cylinder, viscosity, size of cylinder and so on are changed.

Keywords: DualSPHysics, SPH, FreeCAD, fluid flow around a pipe, python, XML, laminar flow, turbulent flow, automated parameter change