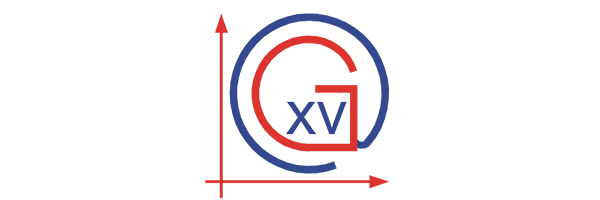
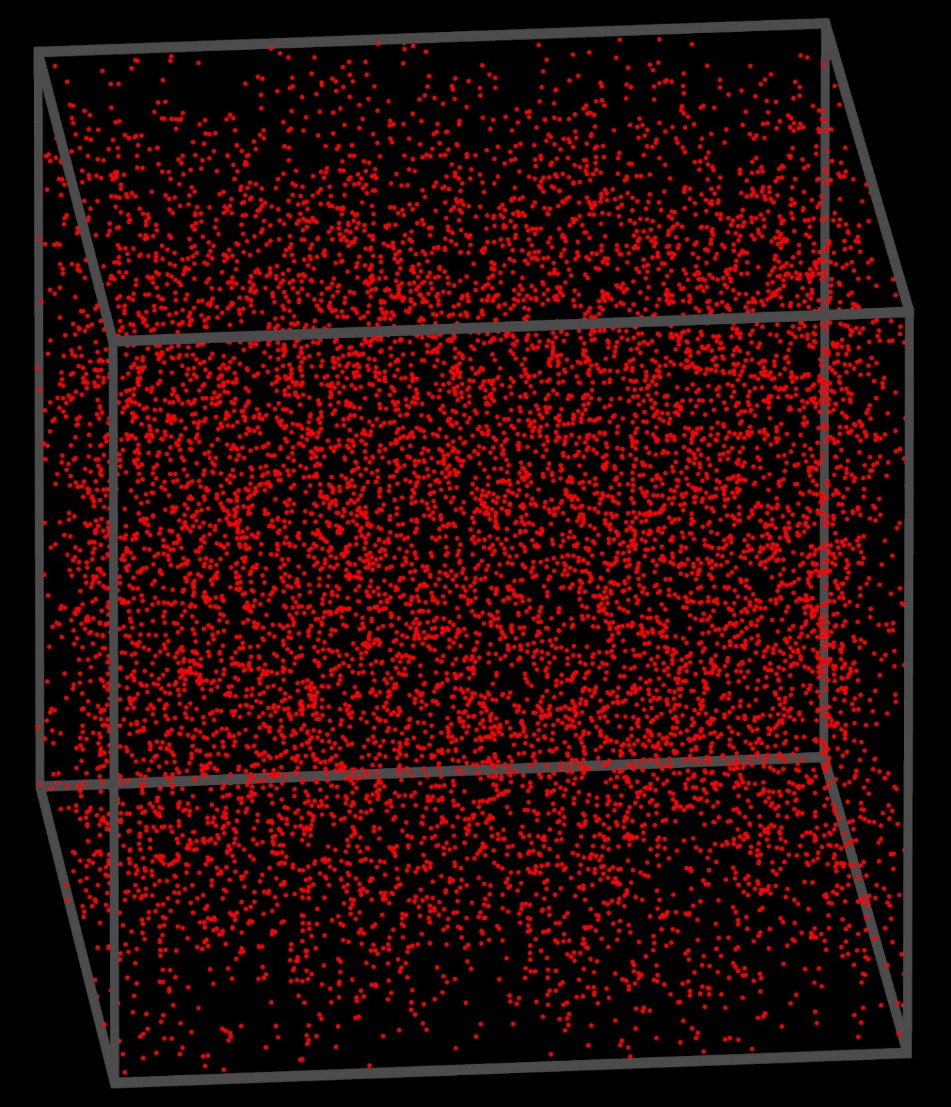
[](https://www.google.hr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwiPj4itnPnZAhWSa1AKHcOLCHcQjRx6BAgAEAU&url=https://www.mioc.hr/wp/?cat%3D71&psig=AOvVaw0bdKIWNjewPLkIL3gJypG_&ust=1521577580794218)

**Autori: Nikola Sočec, Davor Dobrota**

**Mentor: prof. Nikola Dmitrović**

**PLASMA**

**(Particle Localised Action Simulation with Multiple Applications)**



Zagreb, 2020.

**PLASMA**

**(Particle Localised Action Simulation with Multiple Applications)**

**Autori**

Nikola Sočec

Davor Dobrota

**Mentor**

Prof. Nikola Dmitrović

XV. Gimnazija Zagreb, Jordanovac 8

**U Zagrebu, 2020**

**Sadržaj**

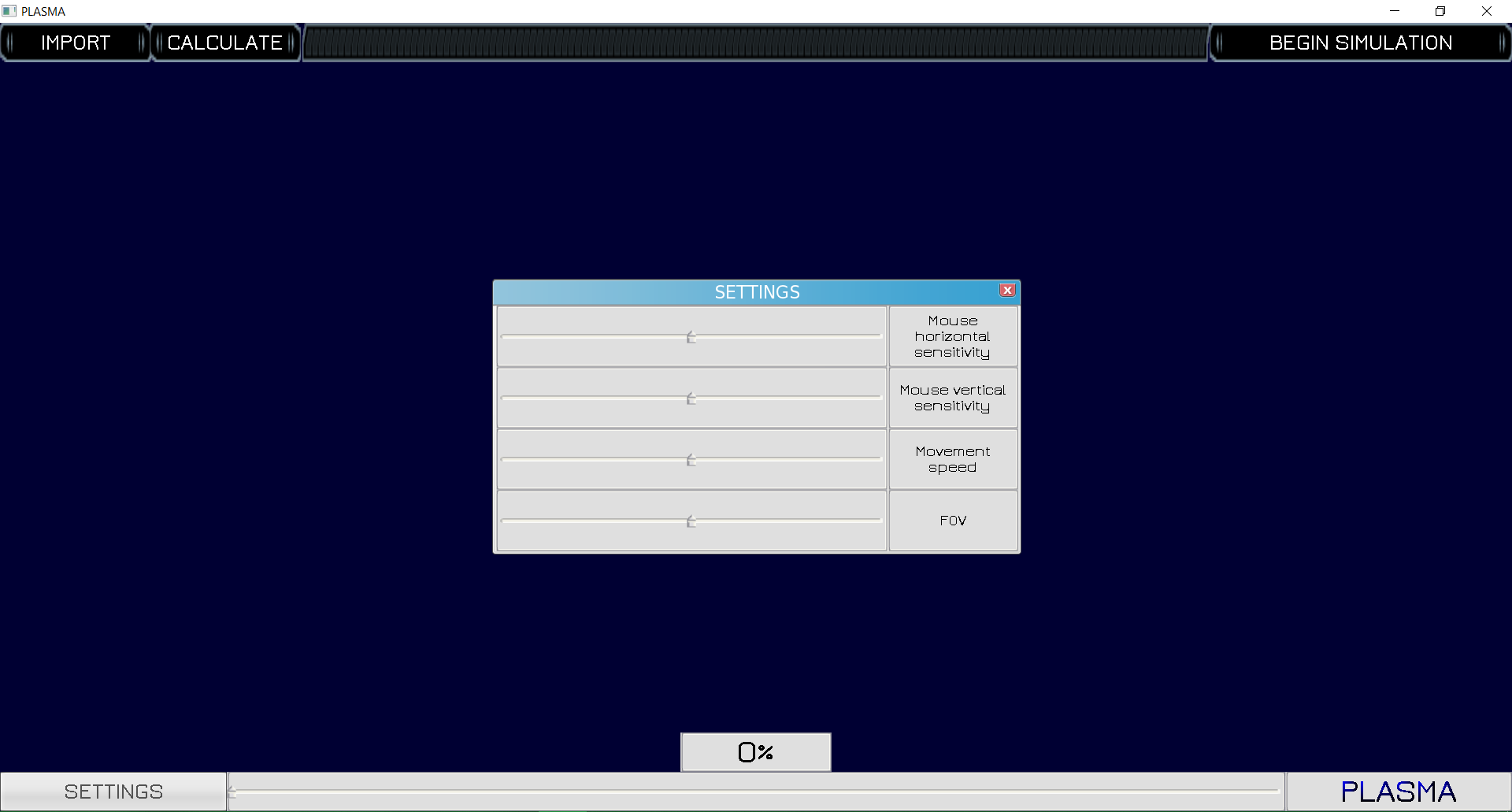
1. **Uvod** 3
2. **Uporaba programa** 4
3. **Fizikalna pozadina** 6
   1. **Elektromagnetizam i polje zavojnice** 6
   2. **Osnovni parametri plazmenih sustava** 19
4. **Rješenje problema** 26
   1. **Općenito o rješenju** 26
   2. **3D ENGINE** 26
   3. **Simulation engine** 27
   4. **GUI** 28
5. **Tehnički podatci sustava** 28
6. **Reference** 29
7. **Uvod**

PLASMA je aplikacija za računanje i prikazivanje sustava čestica plazme ili običnog plina. Sustav je nastao iz potrebe za razumijevanjem sustava plazme na mikroskopskoj razini, a za tu ulogu nije pronađen lako dostupno rješenje. Ova aplikacija je povezana s našim istraživanjem koje se dotiče područja plazme i utjecaja raznih čimbenika na njezino ponašanje. Uz to, postoje mnoge primjene za ovakvu vrstu aplikacije, kao što je modeliranje fuzijskih reaktora i Sunca, a kod simulacije čestica plina postoje primjene u hidraulici, balistici i drugim područjima. Prikaz je trodimenzionalan i omogućeno je kretanje kroz prostor simulacije. Za računanje simulacije postoji nekoliko parametara koji odlučuju kako će se čestice ponašati. Detaljan opis fizike iza simulacije se nalazi u kasnijem poglavlju.

**A picture containing indoor

Description automatically generated**

Slika 1. Jedan trenutak simulacije



2

3

4

5

6

7

8

1

9

10

11

Slika 2. Prikaz grafičkog sučelja

1. **Uporaba programa**

POPIS ELEMENATA GRAFIČKOG SUČELJA

1. gumb za učitavanje podataka izračunatih vanjskim modulom
2. gumb za početak računanja podataka potrebnih za simulaciju
3. pokazivač za vremenski status računanja
4. gumb za pokretanje i zaustavljanje simulacije
5. brojčani indikator vremenskog statusa simulacije
6. traka za navigiranje u simulaciji po vremenu
7. gumb za pokretanje sučelja postavki
8. traka za regulaciju širine vidnog polja kamere
9. traka za regulaciju brzine kretanja kamere po prostoru
10. traka za regulaciju vertikalne brzine rotacije kamere
11. traka za regulaciju horizontalne brzine rotacije kamere

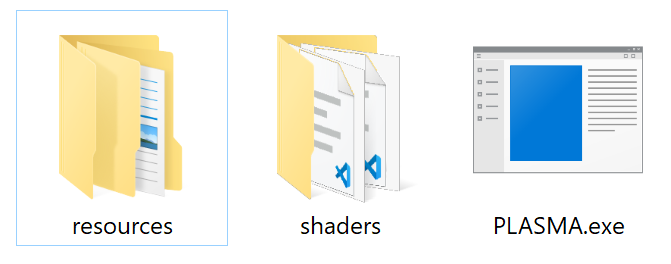
POPIS KONTROLA POMOĆU TIPKOVNICE I MIŠA

* tipkama WASD ili strelicama se kamera pomiče naprijed, lijevo, nazad i desno.
* tipkama SHIFT i SPACE se kamera pomiče dolje i gore.
* mišem se kamera okreće u svim smjerovima.
* tipkom Q se grafičko sučelje skriva i pokazuje.
* klikom miša na sredinu ekrana se ulazi u mod upravljanja kamerom
* tipkom E se izlazi iz moda upravljanja kamerom

PRIMJER KORIŠTENJA PROGRAMA

U ovom primjeru je cilj izračunati simulaciju jednog sustava i prikazati ju u grafičkom sučelju. Prvo je potrebno datoteke PLASMA.exe i PLASMA\_simulation.exe staviti svaki u svoj direktorij. PLASMA.exe ovisi još o nekoliko .dll datoteka koje se također moraju nalaziti u istom direktoriju kao PLASMA.exe. Uz .dll datoteke, potrebno je u direktoriju postaviti direktorije priložene u izvornom kodu: GUI/shaders i GUI/resources. PLASMA\_simulation.exe ovisi o datoteci input.dat koja se mora nalaziti u istom direktoriju kao PLASMA\_simulation.exe. Kada su programi spremni za pokretanje, potrebno je pripremiti datoteku input.dat prema priloženoj slici.

A screenshot of a cell phone

Description automatically generated

Slika 4. direktorij datoteke PLASMA\_simulation.exe

Slika 3. Direktorij datoteke plazma

A close up of text on a black background

Description automatically generated

Slike 5. datoteka input.dat

Nakon što je datoteka input.dat pripremljena, potrebno je pokrenuti aplikaciju PLASMA\_simulation.exe i pričekati da završi. Kad računanje završi, treba pokrenuti aplikaciju PLASMA.exe. Rezultati računanja će biti spremljeni u datoteci calc\_data.pcd koja se nalazi u istom direktoriju kao PLASMA\_simulation.exe. Klikom na gumb IMPORT (element 1) otvorit će se explorer.exe prozor u kojem je potrebno odabrati datoteku calc\_data.pcd. Klikom na gumb Open u explorer prozoru nakon što je datoteka označena, datoteka će biti učitana. Nakon toga preostaje jedino kliknuti na gumb BEGIN SIMULATION (element 4) i vizualizacija simulacije će biti pokrenuta.

1. **Fizikalna pozadina**
   1. **Elektromagnetizam i polje zavojnice**

Plazmu, odnosno radije plazmene sustave, teško je jednoznačno odrediti zbog njihove vrlo velike raznolikosti. Mnogi njihovi parametri pokrivaju desetke redova veličine (primjerice gustoća nabijenih čestica) što proizvodi vrlo različita svojstva pri različitim vrijednostima parametara. No zajedničko svojstvo svih plazmenih sustava jest da središnju ulogu u interakciji, unutar ali i izvan sustava, preuzima elektromagnetska sila. Plazmu stoga najopćenitije definirati kao kvazineutralni plin sastavljen od neutralnih i nabijenih čestica koji pokazuje skupno međudjelovanje zasnovano na elektromagnetskim silama. U jednom termodinamičkom plinskom sustavu, većina međudjelovanja je zasnovana na parnom međudjelovanju, preko sudara, i u nešto manjoj mjeri Van der Waalsovim potencijalom (opada proporcionalno 1/r6), dok u plazmenom sustavu bez ikakvog vanjskog utjecaja prevladava Coulombova sila. Uzimajući ovo u obzir, prirodno je započeti od *Maxwellovih jednadžbi*, temeljnih jednadžbi elektromagnetizma, u diferencijalnom i integralnom obliku.

Nekada se radi poopćenja umjesto *jakost* *električnog polja* [Vm-1]i *gustoća magnetskog toka* [T] koriste termini *gustoća električnog toka* [Cm-2] i *jakost magnetskog polja* [Am-1] kako bi se otklonile konstante i , ali i dodali dielektrični i magnetski parametri ovisni o materijalu (tzv. *vezani parametri*). Tada vrijedi

Gdje predstavlja *magnetizaciju* materijala [Am-1] (određivanje magnetskih svojstava) dok *polarizacijska gustoća* materijala [Cm-2] (određivanje dielektričnih svojstava). je *gustoća struje* [Am-2] (izvor magnetskog polja) dok je *gustoća naboja* [Cm-3].

Navedene relacije daju opći opis elektromagnetske sile, no nama su ipak potrebne nešto specifičnije jednadžbe, u prvome redu *Coulombov zakon* (rješenje jednadžbe (2a) za točkasti naboj)i *Poissonova jednadžba* te *Biot-Savartov* *zakon* zajedno s električnim i magnetskim potencijalima.

je *električni potencijal*, a je *vektorski potencijal magnetskog polja* [Tm], dok je *vektor pomaka* [m], *infinitezimalni element volumena u kojemu teče gustoća struje* , *električna struja* [A], a *infinitezimalni pomak kroz koji teče struja* . Nadalje, bitna je relacija između električnog polja i vektorskog potencijala koja slijedi iz jednadžbi (1c) i (5b) koja govori da je električno polje jednakog smjera kao i vektorski potencijal

Kako ćemo u simulaciju uključiti i utjecaje vanjskih komponenata potrebno je razmotriti kakvo polje stvaraju najčešći izvori vanjskog električnog polja – zavojnice. U našem slučaju raspisujemo polje kružne zavojnice pravokutnog presjeka.

Slika 6. prikaz vektora i kutova na petlji













x

y

z

Zavojnicu možemo promatrati kao mnogo kružnih petlji s različitim radijusima i pozicijama središta, ili još jednostavnije kao jednu veliku konduktivnu površinu kroz koju teče određena gustoća struje. Ovako se odvajamo od koncepta navoja i površinu dijelimo na male segmente proizvoljne veličine. Prvo je potrebno pronaći jednadžbu kojom možemo opisati magnetsko polje i električno polje jednog takvog malog segmenta koji ćemo pojednostavljeno tretirati kao petlju radiusa (koji je za naše potrebe vektor) za bilo koju točku u prostoru, a zatim jednadžbu relativno jednostavno proširimo na cijelu površinu.

S obzirom na kružni oblik petlje korisno je koristiti se polarnim koordinatnim sustavom u kojemu su koordinate definirane udaljenošću od središta te dvama kutovima. Središte petlje postavljamo u ishodište polarnog sustava. Uzimamo proizvoljnu točku u prostoru danu koordinatama i do nje povlačimo vektor . Polarne koordinate jednostavno je prevesti u kartezijanske koordinate i iz toga saznati iznos svake zasebne komponente magnetskog polja. Uvodimo kut koji predstavlja kut između polupravca koji izlazi iz središta petlje i prolazi kroz mali segment na kružnici čiji doprinos promatramo, te polupravca kojeg definira kut α sa središtem. Navedeno se može bolje vidjeti iz sheme prikazane na Slici 5. Za točku koju promatramo primjenjujemo jednadžbu (5a) pojednostavljenu za jednu petlju.

Raspisujemo pripadne vektore na kartezijanske koordinate. U ovoj fazi izvoda zanemarit ćemo kut zato što sam kut isključivo služi razdvajanju x i y komponenti polja. Raspisujemo vektore kartezijanski

Ovo su svi identiteti koji su potrebni za daljnji izračun. Uvrštavamo ih u jednadžbu (17c). Kako više ne integriramo po duljini već po kutu prikladno je zamijeniti s granicama i

Izraz za vektorski potencijal je već sređen, samo je potrebno uočiti da zbog simetrije integrala, gubi se dio u kojemu je zapisan odnosno preostaje samo komponenta s . Sređujemo dalje izraz za magnetsko polje petlje, množeći vektorski komponente dobivamo

Slično kao kod vektorskog potencijala izraz sa se gubi. Pa nam preostaju konačne jednadžbe magnetskog i električnog polja.

Kako bismo dobili natrag rastavljenu horizontalnu komponentu (spajanje u koraku (17f)) Uvodimo kut koji rastavljamo horizontalne komponente i polja i potencijala, pa dobivamo konačne jednadžbe

Iako ove jednadžbe izgledaju dovoljno „lijepo“ ipak imaju jedan veliki nedostatak, a taj je da se ne mogu integrirati na konvencionalan način, pripadaju domeni eliptičnih integrala. Što znači da će biti potrebno napraviti dobru aproksimaciju vrijednosti ovoga integrala i pronaći način za evaluaciju pogreške. No to ostavimo za kasnije, proširimo izraze (19) na cijelu zavojnicu, tako da podijelimo njezin presjek, koji tretiramo kao homogeni vodljivi materijal, na male elemente koje aproksimiramo kao petlje. Kroz navedeni vodljivi presjek teče određena gustoća struje , radije nego specifična struja . Neka je debljina tog presjeka, a duljina (prikaz na Slici 6.)











z

x









Slika 2. presjek vodljive površine

Za ovakvu zavojnicu gustoća struje kroz površinu je konstantna i iznosi

Početni kut i početna udaljenost uzima se od središta zavojnice.

Vidimo da će se radius petlje označen sa mijenjati u intervalu od te da se razmak od dna do vrha zavojnice mijenja u intervalu . Promjena visine utječe na i

Sada možemo konstruirati konačni izraz za magnetsko polje i potencijal vodljive površine

Uvrštavanjem i sređivanjem dobivamo konačne izraze za polje i potencijal

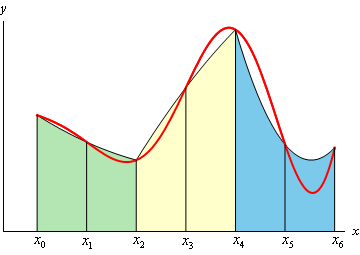
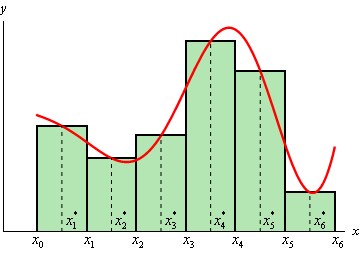
Kao što smo već prije odredili, ovi integrali nemaju antiderivat, već ih je potrebno aproksimirati numeričkim metodama. Razmotrili smo i drugačiji pristup računu, u kojemu smo pokušali umjesto podjele na petlje podijeliti zavojnicu na vodljive segmente, pa zatim računati doprinos svakog segmenta. No ovakav način je doveo do reduciranja samo jednog sloja integrala (umjesto 3 dimenzije aproksimacije, potrebne su samo dvije), a proizveo je vrlo ružan i teško izračunljiv izraz. Stoga je ovakav izraz, s razmjerno jednostavnim unutarnjim izrazima, optimalan za paralelizaciju i izračun preko procesora ili grafičke kartice.

Samo integriranje sa jako malim inkrementima , i je vrlo zahtjevan za računalo, proizvodi velike pogreške i jako je vezan uz preciznost varijable u koju pohranjujemo vrijednosti. Ovo je posljedica „*overflowa*“, gubitka preciznosti zbog repetitivnog zbrajanja vrijednosti u istu varijablu što dovodi do potrebe uporabe povećane preciznosti. Naime, računala koriste dva osnovna tipa realnih brojeva, *float* (od engl. *single precision floating point*) i *double* (od engl. *double precision floating point*). Glavna razlika je veličina varijable, float zauzima 32 bita (ili 4B) dok double 64 bita (8B) te posljedično preciznost, float je precizan na 6 značajnih znamenki i tolerira zbrajanja prije gubitka preciznosti dok je double precizan na 15 značajnih znamenki i tolerira zbrajanja prije overflowa. Ima još jedna ključna razlika, a to je brzina računanja s tim tipovima podataka. Dok procesor računa samo 2 puta sporije, grafička kartica računa punih 32 puta sporije. Stoga je potrebno smanjiti overflow koji nastaje zbrajanjem u varijablu koja sadrži vrijednosti polja, a to možemo učiniti smanjivanjem broja inkremenata. No tako nešto smanjuje preciznost, pa nam treba bolja metoda aproksimacije ovog volumnog integrala.

Prva optimizacija potječe od same prirode kosinusne funkcije, odnosno njene parnosti jer vrijedi

Pa uočavamo da možemo jednostavno razmatrati integral u granicama i zatim pomnožiti s 2 bez da gubimo preciznost. Ovo se lijepo vidi i na Grafu 1. koji prikazuje ovisnosti vrijednosti funkcije o parametru za proizvoljne vrijednosti , i .

Ovaj graf nam također govori da se vrijednosti funkcije mogu brzo mijenjati, odnosno imati velike vrijednosti prvog i drugog derivata. Ovo je važna informacija, jer najbolji način reduciranja broja potrebnih poznatih vrijednosti točaka funkcije koje su potrebne za integriranje je aproksimacija tog područja polinomom. Postoji nekoliko metoda aproksimacije, *metoda središnje točke*, *Simpsonovo pravilo* i *Booleovo pravilo* (sve tri metode su zapravo specifični slučajevi *Newton-Cotesovih formula*). Grafički prve dvije metode je prikazan ispod.



Slika 7. Prikaz dviju metode aproksimacije integrala za proizvoljnu krivulju

Jednadžbe Simpsonove i Booleove aproksimacije integrala za mali interval neke funkcije u kojemu su zadane 3 ili 5 točaka razmaknute za fiksnu vrijednost te njihove apsolutne pogreške za vrijednost iz intervala unutar kojeg aproksimiramo.

Uočavamo da s obzirom na male očekivane vrijednosti i velike moguće nagibe funkcije da nam je pogodnije koristiti Booleovu metodu aproksimacije. No kao što uočavamo ovo postavlja jednu prepreku, našu funkciju moramo podijeliti na n blokova, a zatim u svakome od njih odabrati 5 točaka od kojih će se unutarnje rubne preklapati odnosno ponoviti dva puta, jednom u svakome intervalu. Kako bismo računalu olakšali posao, paralelizirati želimo svako računanje funkcije te izbaciti dvostruka računanja. Ovo lijepo možemo ostvariti stvaranjem liste težinskih vrijednosti. Nakon pomnog razmatranja zaključeno je da je integral povoljno podijeliti na 12 dijelova, a da ikakav veći broj ne utječe znatnije na preciznost. Pa onda naš integral postaje

Iako smo funkciju inicijalno definirali samo za z komponentu polja, isto vrijedi i za x i y komponente. Dakle sada smo reducirali unutarnji integral na sumu

Ovakav oblik sa definiranom listom odličan je za paralelizaciju računa, a vrijednosti ove liste zovu se *težinske vrijednosti*, dok je broj ispred *koeficijent* kojim se množi nakon sumiranja.

Sada preostaje primijeniti sličnu metodu na cijelu vodljivu površinu, odnosno sumirati dobivene doprinose od svake posebne petlje. Ova funkcija očekujemo da je nešto „glađa“ odnosno da nema veliki derivat, pa ćemo opet za maksimalnu preciznost primijeniti Booleovo pravilo no ovaj put umjesto da kreiramo listu težinskih vrijednosti, cilj nam je dobiti matricu. Ovakvu matricu dobivamo tako da množimo vrijednosti liste L na sljedeći način: element matrice je dan kao umnožak vrijednosti i vrijednosti . Ovo slijedi iz činjenice da smo površinu pretvorili u 2D plohu u trodimenzionalnoj funkciji i želimo saznati volumen koji ova ploha zatvara s ravninom koju definiraju x i y osi. Liste i su slične listi samo nemaju 13 elemenata i možemo ih zapisati kao

Vrijednosti i su broj ponavljanja središnjeg djela liste, koji mora uvijek biti serija ponavljanja liste . Ovo ponavljanje sugerira da će se i element matrice ponavljati, a da će dio biti različit za rubne vrijednosti matrice. Element matrice koji se ponavlja će biti

Kako ova matrica ne radi na rubovima definiramo dvije liste koje pokrivaju te slučajeve

Sada definirajmo kako će se ponašati težinska vrijednost točke za određene vrijednosti i

Operator ima ulogu *ostatka u dijeljenju*, a je logički *operator ili*. Uvodimo dvije vrijednosti koje služe za definiranje preciznosti i koje predstavljaju koliko inkremenata uzimamo po segmentu površine u jednom i u drugome smjeru, odnosno je *broj inkremenata po radiusu* (), a *broj inkremenata po duljini* (. U jednadžbi (26) napisan je konačni izraz za numeričku aproksimaciju ovog volumnog integrala. Optimalan izbor ovih inkremenata je ključan za najbolji omjer brzine računanja naspram preciznosti. Kao općenito pravilo određivanja optimalne preciznosti možemo uzeti sljedeće:

1. omjer duljine () i debljine () onda neka omjer i bude jednak njemu,

2. za dobru preciznost vrijedi valja ih uzeti tako da vrijedi nejednakost ,

3. Broj inkremenata i moraju biti djeljivi s 4 kako bi metoda radila optimalno

Dakle za izračunati prosječnu točku potrebno je napraviti operacija uvrštavanja u funkciju s različitim vrijednostima , i . Kako se radi samo o z komponenti polja, potrebno je napraviti isto i za horizontalnu komponentu (koji zatim preko rastavljamo na xi y komponente). Pa je ukupno za naći polje potrebno operacija. U prošlogodišnjem radu smo također računali magnetsko polje, samo je korištena Simpsonova metoda u poveznici s metodom srednje točke. Tada je za isti račun bilo potrebno kao minimum i čak za dobru preciznost. Dakle nova metoda računa je u teoriji 23 puta brža i za razliku od stare ne ovisi o broju namotaja. Razmotrimo da li je i preciznija.

Preciznost možemo provjeravati na 2 načina: 1. preko redukcije formule samo za z os, 2. preko usporedbe sa vrijednostima izračunatim na znatno većoj preciznosti. Jednadžba za polje za postaje samo

Uzimamo specifikacije , , i teče struja od 1A Uzmimo preciznost i .

Prosječna pogreška je svega odnosno što je granica preciznosti float tipa podataka koji je precizan na pa možemo reći da je tolika i preciznost našeg računa. Stara metoda je imala pogrešku što znači da je ova nova metoda čak 30 000 puta preciznija ako se računa samo za z os.

Razmotrimo dalje što se događa ako uspoređujemo za punu kružnicu radiusa , odnosno umjesto da variramo udaljenost, variramo kut . Ovdje očekujemo veće pogreške kada su vrijednosti manje jer je referentno računato s povećanom preciznošću i double tipom. Uzimamo iste podatke i udaljenost

Vidimo da greška ipak nije onoliko zanemariva kao kada samo razmatramo z os no da nije jasno vidljiva na grafu, jer iznosi svega u prosjeku što je znatno bolje od stare metode koja je imala pogrešku od u prosjeku, a bila je 20 puta sporija uz gotovo 20 puta veću pogrešku. Dakle ova nova metoda je jako veliko unaprjeđenje i preciznosti i brzine računa. Preostaje još razmotriti preciznost vektorskog potencijala.

Uočavamo da je pogreška za vektorski potencijal veća nego za polje, ali je konzistentnija i zato je nešto bolji pokazatelj preciznosti same metode. Može se sa velikim samopouzdanjem reći da je pogreška uvijek manja od .

U prošlogodišnjem radu smo također napravili alat za grafičko prikazivanje magnetskog polja zavojnice. Razvojem ove metode program je znatno unaprijeđen i glede brzine (iako nije ubrzanje od 23 puta, 70 000 točaka/s je znatno bolje od 9 000 točaka/s) i glede preciznosti izračuna (stara metoda je proizvodila manju asimetriju u slikama, a uvodila je i potrebu filtriranja ekstremnijih vrijednosti uz rub zavojnice, a nova metoda ovakvih problema nema). Navedena brzina je za jaku grafičku karticu, dok su za procesor očekivane brzine od 250 točaka /s po svakom aktivnom *threadu* (prosječan procesor ih ima 8). Ispod je prikazano magnetsko polje i vektorski potencijal zavojnice koju smo prethodno opisali. Crvena boja prikazuje regije s najvećim poljem, a plava s najmanjim, skala boja je logaritamska kako polje izvan ne bi postalo odmah zanemarivo u usporedbi.

Ovime smo dobili karakteristične matrice i liste koje su potrebne za efikasan izračun polja zavojnice te smo razmotrili njihovu preciznost.

* 1. **Osnovni parametri plazmenih sustava**

Formalna definicija plazme da je u prethodnom poglavlju, a njena općenitost sugerira da se radi o jako širokome spektru sustava sa različitim parametrima. Ovdje ćemo pokušati izvesti i objasniti neke od najvažnijih, ključnih za analizu sustava, ako ih je moguće pouzdano izmjeriti, to jest.

Prva skupina parametara su oni koji se odnose na *gustoće čestica* [m-3] pojedinih vrsta u sustavu (pod pojam vrsta smatramo čestice koje su slične jedna drugoj po masi i električnom naboju). Njih označavamo sa . Kako je plazma kvazineutralan plin, to nam govori da mora postojati balans između negativnih i pozitivnih čestica (iako i na ovo postoje iznimke, primjerice elektronska zraka) pa možemo pretpostaviti da je . Ovime smo napravili još jednu pretpostavku, da su *čestice nositelji negativnog naboja elektroni*, a *nositelji pozitivnog naboja pozitivni ioni*. Ova pretpostavka je točna za većinu sustava s kojima se možemo susresti (primjerice vatra i plazma kugla). No plazma obično nije sačinjena samo od nabijenih čestica, već redovito prevladavaju *neutralne čestice*, *atomi i molekule*, a njihovu gustoću uzimamo da je i obično je nekoliko redova veličine veća od što sugerira da je većina sustava slabo ionizirana (koeficijent ionizacije ). *Koeficijent ionizacije* dan je kao

Važno je primijetiti da je kod tog koeficijenta prisutna koncentracija iona. To čini ovu vrijednost pouzdanijom jer redovito jedan ion otpušta više elektrona pri ionizaciji.

Sljedeća važna mjera je termodinamičke prirode, a to je *temperatura sustava*. Ova temperatura nije dana jednoznačno, već je karakteristična za svaku vrstu i predstavlja prosječnu brzinu čestica u skladu s molekularno-kinetičkim modelom. Ovo sugerira da i temperatura svake vrste može biti različita od temperatura drugih vrsta, i to je redovito slučaj. Obično elektroni posjeduju nekoliko redova veću temperaturu od ostatka sustava (ovo se lijepo vidi u emisiju svijetlosti pri gorenju, gdje elektroni imaju redovito temperature od nekoliko tisuća Kelvina). Iz teorije idealnog plina dobivamo srednju kvadratnu brzinu za pojedine vrste čestica

Oznaka je *stupanj slobode* pojedine vrste, za atome i ione je on obično 3, dok se za elektrone obično uzima da je 1. Čestice, iako se gibaju kaotično ipak donekle prate dvije raspodjele, *Maxwellovu* i *Boltzmannovu*. Boltzmannova raspodjela jednostavno govori da će čestice težiti zauzimanju stanja minimalne potencijalne energije pri nekoj temperaturi. Kako je u našem slučaju dominantan elektrostatski potencijal i čestice se očito odbijaju (žele doći u stanje minimalne potencijalne energije) vrijedi

Maxwellova rapodjela je za sustav sa tri razine slobode dana kao

Iz prirode ove raspodjele vidimo da najvjerojatnija brzina nije ona iz jednadžbe (18c) već da je

Sljedeće bitno svojstvo je *elektronska plazmena frekvencija* i razmotriti ćemo je na plazmi vodikova tipa, u kojoj je koncentracija iona i elektrona jednaka te imaju suprotne naboje. Kako bismo pojednostavili stvari, uzmimo da je gibanje iona zanemarivo u usporedbi s gibanjem elektrona jer vrijedi . Ako je raspodjela na početku malo pomaknuta, dobivamo harmonijski oscilator bez prigušenja. To slijedi iz činjenice da kada se elektroni vrate u ravnotežno stanje, električno polje postaje 0, ali elektroni i dalje imaju brzinu te se nastavljaju gibati u suprotnome smjeru dok im polje koje nastaje asimetrijom naboja ne promijeni opet smjer. Za elektron vrijedi

Promatranjem doprinosa polju, iz Gaussovog zakona dobivamo da vrijedi da vrijedi

Uvrštavanjem dobivamo jednadžbu harmonijskog oscilatora

Iz ovoga slijedi da je tražena elektronska frekvencija

Sljedeća važna vrijednost je *Debyjeva duljina*. Koristeći Boltzmannovu raspodjelu dobivamo da je raspodjela naboja po potencijalu jednaka

Uvrštavanjem u Poissonovu jednadžbu dobivamo

Ako uzimamo da da je elektrostatksa energija manja od termičke, možemo pretpostaviti da je

Jednadžba postaje

Gdje je Debyjeva duljina i njen izraz je

Debyjevu duljinu možemo gledati kao karakterističnu udaljenost oko koje se pri uvođenju naboja u plazmu stvara poremećaj raspodjele naboja. Pa za potencijal u ovome poremećaju dobivamo

Iznimna važnost ovih dviju parametara leži u činjenici da daju osjećaj za dimenzija sustava, specifično na kolikim vremenskim intervalima i duljinama možemo očekivati raspad znatnih kolektivnih svojstava. Vidimo da će se svojstva električnog polja znatnije promijeniti u vremenu reda veličine , a u tome vremenu čestice će proći duljinu jednaku . Navedeni poremećaj se zove *elektrostatski val* ili, kako se često naziva *Langmuirov val*.

Sljedeće svojstvo koje predstavlja interes je električna vodljivost plazmenog sustava. Električnu vodljivost smo definirali u uvodu, da je ona proporcionalna električnom polju, no za plazmene sustave se ona redovito zna tretirati kao tenzor (zbog tenzorske prirode gustoće struje u volumenu), no za naše potrebe neka ostane obična skalarna vrijednost. Ova vodljivost se zove *Spitzerova klasična vodljivost*, a temelji se na analizi vjerojatnosti sudara. Izvod je poprilično dugačak i zahtijeva kompliciranu analizu vjerojatnosti sudara no zapišimo nekoliko ključnih koraka u izvodu

Gdje je

je dosta komplicirano dobiti vrlo precizno no analizom se dobilo

Član se naziva *Coulombov logaritam* i za većinu sustava iznosi između 10 i 20. Ako uvrstimo aproksimaciju u jednadžbu (21b) dobivamo približan oblik Spitzerove vodljivosti

Razmotrivši neke temeljne parametre koji ovise o električnom polju, pogledajmo kakav utjecaj ima magnetsko polje. Njegov najbitniji utjecaj potječe zasigurno od *Lorentzove sile*

Uočavamo da je navedena sila centripetalne prirode pa vrijedi

Pa vrijedi da je radijus po kojemu će se čestica gibati jednak

se još naziva i *Larmorovim radijusom* i govori koliki je radijus po kojemu se gibaju nabijene čestice u plazmi u prisutnosti magnetskog polja. Ove čestice također imaju frekvenciju kruženja

Valja naglasiti da su *ciklotronske frekvencije* različite za ione i elektrone zbog njihovih različitih masa (a nekad i naboja). Važno svojstvo koje proizlazi iz ovog kruženja je pojava magnetskog momenta, odnosno čestica koja kruži dobiva magnetski moment okomit na smjer svog gibanja pa vrijedi

Važno je naglasiti da je ovaj fenomen dijamagnetičan, jer navedena struja stvara polje suprotno vanjskome magnetskome.

Valja spomenuti još nekoliko osnovnih parametara koji nam daju informacije o tendencijama gibanja plazme. Prvi takav je *plazmeni parametar* *.* On predstavlja omjer tlaka plazme i magnetskog tlaka

Navedeno je bitno jer ako je parametar znatno manji od 1, valja uzimati u obzir samo gibanje duž silnica magnetskoga polja i njegov gradijent. Drugi parametar je *Reynoldsov magnetski broj* dan kao

Gdje je *karakteristična dimenzija sustava*, *brzina plazme*, a je *magnetska difuzivnost* jednaka

On mjeri odnos konvektivnog gibanja magnetoplazme i difuzije magnetskog polja, a ako je prevladava konvektivno gibanje, odnosno plazma se ne može gibati okomito na magnetsko polje nego jedino uzduž silnica. Ovdje također postaje bitan i *Alfvenov broj* koji je omjer dinamičkog i magnetskog tlaka

Važnost ovog parametra ogleda se u činjenici da ako kinetičko gibanje nadilazi magnetsko, da ono povlači silnice za sobom, a u drugome slučaju magnetsko polje kontrolira smjer toka. Konačno, važan nam je omjer dviju prethodno spomenutih frekvencija, ciklotronske i elektronske, a on je bitan zbog nastanka različitih valova i nestabilnosti u sustavu.

Razmotrili smo u uvodu kako se ponaša polje zavojnice koja utječe na sustav te osnovne zakone elektromagnetizma. No također treba razmotriti kako čestice utječu jedna na drugu. Zakon ove interakcije je jednostavno zapisati u slučaju da uzimamo da svaka čestica djeluje na svaku drugu česticu pa će ukupnno djelovanje na česticu biti samo

Gdje je ukupno električno polje kojim sve druge čestice djeluju na jednu, ali i ukupno vanjsko električno polje (stacionarno ili promjenjivo u vremenu). Električna interakcija između dvije čestice dana je Coulombovim zakonom

Druga stvar koja utječe na dinamiku sustava je magnetsko polje kojim čestice djeluju jedne na drugu

Zaključujemo da će nam vektor udaljenosti između čestica biti korisni za razne aspekte simulacije i da se više puta koristi. Stoga kreiramo sustav 4 matrice u koje ćemo pohranjivati vrijednosti , , i . Najefikasniji način za pronaći sve udaljenosti između dviju čestica je tako da definiramo udaljenost od čestice s indeksom do one s indeksom . Važno je primijetiti simetriju, a ta je da je Također valja definirati da se u simulaciji nalazi čestica. Tada matrice dobivaju oblik

Iz navedenog uočavamo da komponente vektora (i magnitudu) udaljenosti računamo u ukupnoj složenosti . Sljedeće primjenjujemo formulu (24b) na sve parove čestica, u 3 zasebne matrice pohranjujemo sve električne interakcije te formulu (24c) u još 3 matrice, oblika jednakog kao one za udaljenost. Kada dobijemo sve navedene vrijednosti, koristimo formulu (22) u proširenom obliku

Definiramo tri varijable , i koje predstavljaju akceleraciju čestica u svakome smjeru i pohranjene su unutar čestice. Ona se izračunava iz formule (26) koju primjenjujemo prvo na svaku česticu u zadanom redu, odnosno sumiramo sve vrijednosti , i kako bismo dobili ukupnu silu na česticu, a zatim je dijelimo s masom navedene čestice (ona ne mora biti ista za svaku česticu, kao ni naboj), dobivajući akceleraciju. Promjena brzine i položaja dana je kao

Navedeni podatci se zapisuju u varijable pohranjene u čestici, i u sljedećem inkrementu one postaju izvorne vrijednosti. Unutar klase Particle nalaze se sljedeće varijable

* Particle
  + float mass - masa čestice
  + float charge - naboj čestice
  + float radius - atomski radijus čestice
  + float inherentMagneticMoment - intrinzični magnetski moment čestice
  + float ionisationEnergy - energija ionizacije čestice
  + float positionX, positionY, positionZ - pozicija čestice u zadnjem frameu
  + float velocityX, velocityY, velocityZ - brzina čestice u zadnjem frameu
  + float accX, accY, accZ - akceleracija čestice u zadnjem frameu
  + float EPFieldX, EPFieldY, EPFielkZ - električna polja od čestica na česticu u frameu
  + float EFieldX, EFieldY, EFieldZ - vanjska električna polja na česticu u zadnjem frameu
  + float EPotential - električni potencijal čestice u zadnjem frameu
  + float BPFieldX, BPFieldY, BPFieldZ - magnetsko polje od čestica na česticu u frameu
  + float BFieldX, BFieldY, BfieldZ - vanjsko magnetsko polje na česticu u frameu
  + float mMomentX, mMomentY, mMomentZ - magnetski moment u frameu

Navedeni električni potencijal dobivamo na isti način kao i ostala polja, samo nam je potrebna samo jedna matrica kako se radi o skalarnoj veličini. Električni potencijal dan je kao

Sve čestice započinju sa nasumičnom prostornom raspodjelom i posjeduju termičku brzinu danu kao

Važno je naglasiti da je ovo ukupna brzina, i da je nasumično raspoređujemo po komponentama te tako dobivamo nasumični smjer početne brzine, a za navedeno koristimo marsenne twister.

Konačno dolazimo do teme sudara. Sudare možemo podijeliti u 3 kategorije: sudare sa stranicama kutije u koju smo stavili plazmu, sudare među česticama s manjom energijom, i kinetičke ionizacije. Sudari sa stranicama kutije su relativno jednostavni, čestica se odbija s istom energijom kojom je upala, odnosno mijenjamo smjer komponente brzine koja je okomita na navedenu stranicu. Sudari s česticama nižih energija (manjim od energije ionizacije) su nešto kompliciraniji.

Do njih dolazi kada je udaljenost između dvije čestice manja od zbroja njihovih radijusa. Prvo nalazimo komponentu brzine koja je u smjeru vektora koji povezuje središta. To činimo tako da skalarno množimo vektor brzine i udaljenosti te podijelimo s magnitudom udaljenosti. Tada smo slučaj reducirali na jednu dimenziju i možemo ga općenito riješiti pomoću zakona očuvanja energije i gibanja.

Rješavanje dobivamo općenite izraze za nove brzine i

Navedeni parametri uz dodatak pribrajanja utjecaja vanjskih polja na sustav, čine jezgru mehanike simulacije.

Također su ugrađeni alati za dobivanje signifikantnih raspodjela unutar sustava, ponajprije Maxwellove raspodjele, Boltzmannove raspodjele, te nekoliko raspodjela u manjim elementima volumena sustava (tijekom više frameova) među kojima su najbitnije raspodjela naboja, brzine i potencijala.

1. **Rješenje problema**
   1. **Općenito o rješenju**

Rješenje problema se sastoji od dva glavna dijela: računanje stanja sustava u svakom trenutku i vizualizacija sustava. Sukladno tome, kod je razdijeljen u dva glavna dijela: GUI (grafičko sučelje) i simulation engine (računanje sustava). Sav kod je napisan u jeziku C++, a korišteni alati su WinAPI (program je napravljen za Windows), OpenGL (za 3D prikazivanje), Assimp (za učitavanje 3D modela), CEGUI (za elemente grafičkog sučelja) i GLFW (radi uz OpenGL, za stvaranje prozora i dobivanja unosa od periferije). Za implementaciju   
GUI-a je bilo potrebno razviti alat koji se nalazi između glavnog koda i OpenGL-a, trenutno nazvan 3D engine. Projekt je implementiran kao skupina C++ classova tako da svaki pokriva određenu funkcionalnost. Tematski su podijeljeni na tri dijela: 3D engine, GUI i Simulation engine.

* 1. **3D ENGINE**

3D engine je skupina modula koji služe za stvaranje prozora programa, primanje unosa od periferije, učitavanje i prikazivanje 3D modela te sadrži glavnu petlju programa. Distribuira se u obliku statičkog librarya.

**POPIS MODULA**

* Runtime
  + stvara prozor programa
  + sadrži glavnu petlju programa
  + omogućuje registriranje korisnikove funkcije koja će se pozivati za svaku iteraciju petlje
  + prikazuje registrirane objekte za svaku iteraciju petlje
  + omogućuje upravljanje opcijama
    - Vsync – petlja iterira u frekvenciji monitora
    - micanje kamere mišem i tipkovnicom
      * promjena horizontalne i vertikalne brzine okretanja kamere
      * promjena brzine micanje kamere kroz prostor
    - boja pozadine
* Renderer
  + omogućuje prikazivanje jednog ili više objekata (predviđeno je da to automatski radi modul Runtime, ali je omogućeno ručno prikazivanje objekata)
  + omogućuje upravljanje opcijama
    - pozicija kamere u prostoru
    - rotacija kamere u prostoru
    - FOV – širina vidnog polja kamere
    - maksimalna udaljenost do koje kamera može vidjeti
* ModelHandler
  + omogućuje učitavanje 3D modela iz datoteke
  + svi učitani modeli su imenovani, tako da nema potrebe učitati isti model više puta
* ObjectHandler
  + omogućuje stvaranje i spremanje 3D objekata u memoriju
  + 3D objekt ima svoju poziciju, boju, ime učitanog modela i druge parametre
* EventHandler
  + omogućuje primanje unosa od periferije
  + sadrži funkcije za registraciju korisnikovih funkcija koje će se pozivati u slučaju unosa
  + pokrivene vrste unosa
    - pritisak tipke
    - unos slova
    - izlazak miša iz prozora
    - pomak miša
    - pritisak miša
    - pomak kotačića miša
    - promjena veličine prozora
  1. **Simulation engine**

Simulation engine trenutno ima samo jedan aktivan modul, SimulationEngine. U ovom modulu se odvijaju sve kalkulacije potrebne za simulaciju. Trenutno se parametri kalkulacije unose iz datoteke input.dat koja se nalazi u istom direktoriju kao .exe, a rezultati se spremaju u datoteku calc\_data.pcd (Plasma Calculation Data). U budućnosti će ovaj modul biti u obliku statičkog librarya i pozivat će se direktno iz GUI koda. Računanje trenutno nije u potpunosti ubrzano grafičkom karticom i to je veliki prioritet za budućnost.

**POPIS PARAMETARA KALKULACIJE (primjer datoteke input.dat)**

Particle ammount = 200 - broj čestica u simulaciji

Space scale = 1.0e-7 - dimenzije 3D kubičnog prostora u metrima

Initial temperature = 1000 - početna temperatura čestica u kelvinima

Time increment = 5.0e-14 - vremenski interval između okvira (trenutaka) simulacije u sekundama

Number of frames = 600 - ukupan broj okvira za izračunati

Particle charge = 1 - naboj čestica u višekratnicima e (elementarni naboj)

Particle mass = 1.007825 - masa čestica u višekratnicima u (unificirana atomska jedinica)

Magnetic field = 0 - homogeno magnetsko polje prostora u Teslama

Electric field = 0 - homogeno električno polje prostora u Voltima po metru

* 1. **GUI**

Ova skupina modula predstavlja srce PLASME. Ovdje su sadržani moduli koji upravljaju elementima sučelja, koji procesiraju unos iz periferije i koji prikazuju simulaciju. U budućnosti će ovo biti jedina skupina modula koja će se pokretati u .exe formatu, dok će sve ostale skupine biti u obliku statičkog librarya.

**POPIS MODULA**

* GUIHandler
  + pokreće CEGUI sustav
  + prima unos od periferije te ga distribuira
    - CEGUI sustavu (jer ga ne prima sam)
    - korisniku koji može definirati funkcije koje se pozivaju u slučaju unosa
  + omogućuje
    - dohvaćanje CEGUI prozora po imenu
    - primjenu korisnikove funkcije na prozor određenog imena
    - primjenu korisnikove funkcije na sve prozore
* SimulationRenderer
  + priprema i prikazuje simulaciju
  + omogućuje
    - učitavanje već izračunate simulacije iz datoteke
    - započinjanje i pauziranje prikaza simulacije
    - omogućuje namještanje simulacije na određeni trenutak
    - u budućnosti početak računanja simulacije

1. **Tehnički podatci sustava**

|  |  |
| --- | --- |
| OS | WINDOWS 10 |
| CPU | INTEL I7 8xxx ili bolje |
| GPU | NVIDIA GTX 1060 ili bolje |

1. **Reference**

[1] Bojan Vršnak: Temelji fizike plazme

[2] Biot-Savart-Laplaceov zakon za magnetno polje  
 <https://en.wikipedia.org/wiki/Biot%E2%80%93Savart_law> (09.02.2019.)

[3] Maxwellove jednadžbe   
 <https://en.wikipedia.org/wiki/Maxwell%27s_equations> (09.02.2019.)

[4] Metode aproksimacije integrala  
<http://tutorial.math.lamar.edu/Classes/CalcII/ApproximatingDefIntegrals.aspx> (09.02.2019.)