

ŠOLSKI CENTER VELENJE
GIMNAZIJA VELENJE
TRG MLADOSTI 3, 3320 VELENJE
MLADI RAZISKOVALCI ZA RAZVOJ ŠALEŠKE DOLINE

RAZISKOVALNA NALOGA
PROJEKT GRAVITACIJA
Tematsko področje: FIZIKA, RAČUNALNIŠTVO

Avtor:
Luka Jevšenak, 4. letnik

Mentor:
Janez Podhostnik, univ. dipl. inž. fizike
Peter Jevšenak, univ. dipl. inž. fizike

Velenje, 2018

Jevšenak, L. Projekt gravitacija
Raziskovalna naloga, Gimnazija Velenje, 2018

II

Raziskovalna naloga je bila opravljena na Gimnaziji Velenje.

Mentorja: Janez Podhostnik, univ. dipl. inž. fizike, Peter Jevšenak, univ. dipl. inž. fizike

Datum predstavitve:

KLJUČNA DOKUMENTACIJSKA INFORMACIJA

ŠD Gimnazija Velenje, šolsko leto 2017/2018
KG gravitacija / galaksije / računalniške simulacije / algoritem Barnes-Hut
AV JEVŠENAK, Luka
SA PODHOSTNIK, Janez / JEVŠENAK, Peter
KZ 3320 Velenje, SLO, Trg mladosti 3
ZA Gimnazija Velenje
LI 2018
IN **PROJEKT GRAVITACIJA**
TD Raziskovalna naloga
OP
IJ SL
JI sl / en

AI Dinamiko teles, ki tvorijo galaksijo, usmerja gravitacijska interakcija med njimi. Vsako telo z gravitacijsko silo vpliva na vsa ostala telesa in s tem vpliva na njihovo gibanje. Pri sto milijardah zvezd v galaksiji je natančen izračun vseh interakcij in premikov nemogoč tudi z najboljšimi računalniki na svetu, zato se za simuliranje procesov, kot sta vrtenje galaksij ali galaktični trk, uporabljajo različne metode poenostavljanja. Pri obravnavi gravitacijske interakcije med n telesi se je najbolj prijel algoritem Barnes-Hut, ki združuje oddaljenejša telesa v eno samo telo po določenem kriteriju. Za natančen opis gibanja je potrebno vsa telesa postaviti v 3D-prostor in jih v tem prostoru tudi premikati, zato se je veliko časa posvetilo tehnikam programiranja 3D-grafike. Cilj raziskovalne naloge je proučiti dogajanje, ko med vsaj 10 000 telesi po izbrani začetni postavitvi v 3D-prostoru deluje gravitacijska interakcija. To pa je mogoče doseči z računalniškim programom, ki združuje 3D-grafiko in Barnes-Hut algoritem v delujočo celoto. Programiranje v programskem jeziku C++ je potekalo v okolju Visual studio 2015 z vključitvijo programskega vmesnika OpenGL za 3D-grafiko. Po optimizaciji programske kode lahko na zaslonu spremljamo zvezno gibanje 10 000 teles z dovolj veliko dinamike, da je ogled zanimiv tudi v živo. Zaradi doseganja boljših vizualnih učinkov pa so naslednje simulacije potekale s 50 000 telesi, porazdeljenimi naključno po ploščatem disku ali po notranjosti krogle. Čas simulacij se je zaradi tega podaljšal na dobro uro, zato se je dogajanje snemalo z zaslonskim snemalnikom. Iz teh posnetkov so se nato naredili videi z 32-kratno hitrostjo predvajanja. Na videih lahko spremljamo gravitacijsko sesedanje snovi, trk galaktičnega diska s črno luknjo in trk dveh galaksij. Delujoč računalniški program predstavlja dobro osnovo za postopno približevanje realnejšim situacijam glede števila teles in začetnih pogojev, vendar pa je vzporedno s tem potrebno razmišljati o možnostih uporabe hitrejših računalnikov in natančnejših numeričnih metod integracije.

KEY WORD DOCUMENTATION

ND Gimnazija Velenje, 2017/2018
CX gravity / galaxy / computer simulations / Barnes-Hut algorithm
AU JEVŠENAK, Luka
AA PODHOSTNIK, Janez / JEVŠENAK, Peter
PP 3320 Velenje, SLO, Trg mladosti 3
PB Gimnazija Velenje
PY 2018
TI **PROJECT GRAVITY**
DT Research work
NO
LA SL
AL sl / en

AB In a galaxy, every star exerts a gravitational force on all other stars, thus having an impact on their motion. There are hundreds of billions of stars in a single galaxy and the exact calculation of their motion is impossible even with the best computers in the world. For this reason, in computer simulations of galactic rotations and collisions various methods need to be used to reduce the number of calculations. To calculate gravitational interaction between a large number of stars, the Barnes-Hut algorithm is most commonly used. It groups together distant stars and approximates them into a single body. For a detailed description of the gravitational interactions between the stars, a 3D world was created on a personal computer to enable following their movements during the simulation. The initial goal of this research work was to study the motion of at least 10,000 stars, set in a 3D environment, after gravitational forces come into effect. To make that possible, a special computer programme in C++ programming language that integrates 3D graphics (API OpenGL) with the Barnes-Hut algorithm was created. After the programme code was optimized, the simulation ran fast and smooth enough that it was possible to follow the motions of 10,000 stars on the computer screen. In order to achieve even better visual effects, the number of stars in the following simulations was increased to 50,000. Consequently, the time of the simulation increased too, so a screen recorder was used to provide recordings, which were converted into videos with 32x speed of the original. In these videos it is possible to observe stars spiralling towards the centre of mass, a black hole deforming a galactic disc and a collision of two galaxies. The developed computer programme represents a good basis for gradual future improvements of such simulations by further increasing the number of stars and by taking into account more realistic initial conditions. However, in order for this to be achieved, more accurate methods of numerical integration as well as faster computers will be necessary.

KAZALO

1	UVOD	1
2	PREGLED OBJAV	2
2.1	Gravitacija	2
2.2	Programiranje v 3D	2
2.3	Barnes-Hut algoritem	3
3	METODE DELA.....	4
4	REZULTATI.....	6
4.1	Izboljševanje hitrosti simulacij.....	6
4.2	Potek simulacij	7
4.2.1	Razporeditev teles po disku.....	7
4.2.2	Razporeditev teles po krogli.....	9
4.2.3	Disk in črna luknja.....	11
4.2.4	Trk diska in krogle.....	14
5	RAZPRAVA	15
6	ZAKLJUČEK.....	17
7	POVZETEK	17
8	ZAHVALA.....	19
9	PRILOGA.....	19
10	VIRI IN LITERATURA.....	20

KAZALO SLIK

Slika 1: Razdelitev prostora na podprostore v 2D	4
Slika 2: Drevesna shema algoritma v 2D	4
Slika 3a, b: Disk s 50 000 teles, pogled od zgoraj in pogled s strani	5
Slika 4a, b: Praznina v sredini in sesedanje diska	7
Slika 5a, b: Obroč in štiri lokalne zgoščine	8
Slika 6a, b: Postopno sesedanje v enotno zgoščino	8
Slika 7a, b, c: Gravitacijsko sesedanje diska	8
Slika 8a, b: Začasna razširitev in ponovna zgostitev v središču	9
Slika 9a, b: Začetna krogelna postavitvev	9
Slika 10a, b: Vrh in dno krogle padeta skupaj	10
Slika 11a, b: Iz jedra se razvijeta manjša kraka	10
Slika 12a, b: Največja zgostitev	10
Slika 13a, b: Močna spiralna kraka	11
Slika 14a, b: Končna oblika	11
Slika 15a, b: Približevanje črne luknje disku in trenutek trka	12
Slika 16a, b: Uničenje jedra diska	12
Slika 17a, b: Pogled z oddaljenejšje perspektive	12
Slika 18a, b: Približevanje črne luknje jedru diska	13
Slika 19a, b: Polžasta oblika in nekaj preostalih lokalnih zgoščin	13
Slika 20: Črna luknja potegne telesa z roba diska v »slonji rillec«	13
Slika 21a, b: Začetna postavitvev in približevanje (rdeče) krogle jedru (belega) diska	14
Slika 22a, b: Bližja perspektiva stika obeh jeder in začetek pospeševanja v smeri urinih kazalcev	14
Slika 23a, b: Rdeča spirala od daleč in blizu	15
Slika 24: Izvor napak pri računanju tira telesa	16

1 UVOD

Astronomi so v zadnjih desetletjih priskrbeli osupljive slike zvezd, meglic, galaksij, galaktičnih trkov, mogočnih curkov snovi in svetlobe, ki jih poganjajo črne luknje. Na teh slikah je ujet le trenutek procesa, ki lahko traja na milijone let. Strokovnjaki si zato pomagajo z računalniškimi simulacijami, da dobijo čim boljši pogled v celotno dogajanje procesa, ki jih zanima. V procesih, kot je trk dveh galaksij, sodeluje nepredstavlljivo veliko število teles. Samo v eni galaksiji je lahko več 100 milijard zvezd, prav vsaka zvezda pa z gravitacijsko silo deluje na vse ostale zvezde in s tem vpliva na njihovo gibanje. Tako je natančen izračun vseh interakcij in premikov tudi z najsodobnejšimi računalniki nemogoč.

Kakšne metode se potem uporabljajo, da se simulacije vseeno zanesljivo izvajajo?

Ali so tega sposobni že običajni osebni računalniki?

Ali lahko sam z računalnikom ustvarim svojo galaksijo ali celo galaktični trk v tridimenzionalnem prostoru?

Kljub pomislekom, da odgovori na zgornja vprašanja presegajo moje trenutne zmožnosti in znanje, sem se odločil sprejeti izziv in izdelati raziskovalno nalogo s ciljem napisati računalniški program, kjer bi gibanje množice teles usmerjala gravitacijska interakcija med njimi.

V preteklosti sem se že ukvarjal z gravitacijo, tako sem v programskem jeziku C++ simuliral gravitacijsko polje teles nepravilnih oblik, simuliral sem gibanje zvezd v dvozvezdijah in proučeval stabilnost orbit planetov v takih sistemih. Pot k novemu cilju pa sem začel z namestitvijo knjižnic OpenGL, ki služijo kot razširitev programskega jezika C++ za programiranje grafike v tridimenzionalnem (3D) prostoru. Potrebno se je bilo naučiti kako 3D-objekt ustvariti in premakniti v 3D-prostoru, kako pogledati nanj iz drugega zornega kota, kako ga predstaviti na 2D-zaslону ... Pri vsem tem so v veliko pomoč transformacijske matrike, ki jih obravnava linearna algebra. Naslednja faza je bila usvojitev algoritma Barnes-Hut. Ta algoritem uspešno zmanjšuje število potrebnih izračunov v simulacijah z velikim številom teles na račun združevanja oddaljenejših teles po določenem kriteriju. Posnetki simulacij na spletu (youtube: iskalni niz »Barnes-Hut«) vsebujejo različna števila teles, večinoma med deset tisoč in milijon. Glede na to, da se z omenjenimi problemi, programi in metodami srečujem prvič, sem si kriterij za uspešnost svojega dela postavil na spodnjo mejo. Zadovoljen bom, če bom na zaslonu svojega osebnega računalnika lahko opazoval zvezno gibanje (vsaj) 10 000 teles hkrati.

Moj namen ni natančno simuliranje procesov, ki potekajo v vesolju, saj je gravitacija vidne snovi le redkokdaj edini odločilni dejavnik od začetka do konca. Na tvorbo, vrtenje in razporeditev galaksij odločilno vpliva skrivnostna temna snov, znaten del mase galaksij pa predstavljajo tudi ogromni oblaki prahu in plina, ki jih ne moremo smatrati za točkasta telesa tako kot zvezde. Z izdelavo naloge bi predvsem rad pridobil veliko novega in uporabnega matematičnega in računalniškega znanja, motivacija za delo pa je priti do rešitve zastavljenega fizikalnega problema: proučiti dogajanje, ko med n telesi po izbrani začetni postavitvi v 3D-prostoru deluje samo gravitacijska interakcija.

2 PREGLED OBJAV

2.1 Gravitacija

Gravitacijsko silo med telesi podaja Newtonov gravitacijski zakon:

$$F_g = \frac{Gm_1m_2}{r^2},$$

kjer je $G = 6,67 \cdot 10^{-11} \frac{\text{Nm}^2}{\text{kg}^2}$ gravitacijska konstanta, m_1 in m_2 masi točkastih teles ali masi krogel, v katerih je gostota odvisna samo od razdalje od središča in r razdalja med njunima središčema. Gravitacijska sila je vedno privlačna in leži na zveznici točkastih teles ali na zveznici težišč krogel. Lahko pa na gravitacijo pogledamo tudi drugače: masno telo M (točkasto ali krogla) spremeni lastnosti prostora, ki tako postane gravitacijsko polje. Označimo ga s težnim pospeškom

$$\vec{g} = \frac{GM}{r^2} \frac{\vec{r}}{|\vec{r}|} \quad (1),$$

pri čemer je \vec{r} vektor od izbrane točke do telesa M . Pri obravnavi sistema teles, ki se privlačijo z gravitacijsko silo, je potencialna energija prvega telesa v polju drugega telesa enaka:

$$W_p = -G \frac{m_1m_2}{r} \quad (2).$$

Enak izraz velja za skupno potencialno energijo obeh teles [1].

2.2 Programiranje v 3D

Grafični procesor lahko izrisuje le tri elemente: točke, črte in trikotnike. Točko predstavimo v 3D prostoru s krajevnim vektorjem, črte z dvema krajevnima vektorjema, ki določata začetek in konec črte, trikotnik pa predstavimo s tremi krajevnimi vektorji, ki določajo vsa tri oglišča. Vse objekte, ki se pojavijo na zaslonu, konstruiramo s temi tremi elementi. Kocko tako predstavimo z dvanajstimi pravokotnimi trikotniki, po dva za vsako ploskev.

Če želimo ustvariti svet z več kot enim objektom, objektov ni smiselno konstruirati v koordinatnem sistemu sveta. Veliko priročneje je, da vsak objekt posebej opišemo v ločenem koordinatnem sistemu in ga nato premaknemo v svetovni koordinatni sistem. Za to poskrbimo z matriko *model matrix*, ki preslika krajevne vektorje objekta iz ločenega v svetovni koordinatni sistem. Tako ustvarjamo naš 3D-svet. Ker pa želimo na zaslonu prikazati le tisti del, ki ga naša kamera vidi, potrebujemo še dve matriki. Matrika *view matrix* preslika svet v koordinatni sistem, ki ima izhodišče v kameri. Matrika *projection matrix* pa nastavi zorni kot kamere in projicira 3D-vektorje v 2D-koordinatni sistem, ki ga lahko prikažemo na zaslonu.

Uporaba ločenih koordinatnih sistemov je smiselna iz več razlogov. Objekt – predmet lahko ustvarimo v nam najbolj ugodni legi, kar lahko precej olajša delo. Predmet, ki se pojavi večkrat v svetu, samo prenesemo na prava mesta. To storimo v treh korakih. Najprej določimo velikost predmeta, ga zavrtimo v pravi položaj in ga prestavimo. Konstrukcija predmeta in konstrukcija transformacijskih matrik potekata na centralnem procesorju računalnika. Podatke nato naložimo na grafični spomin in svoje delo začne opravljati grafični procesor, ki je specializiran

za izvajanje operacij med matrikami, zato računalnik dela hitreje, saj je grafični procesor (GPU) v teh nalogah precej učinkovitejši kot univerzalni centralni procesor (CPU).

OpenGL je aplikacijski programski vmesnik (API) za izrisovanje 2D- in 3D-vektorske grafike. Sestavlja ga set funkcij, ki omogočajo risanje scen v 3D-svetu z optimizirano uporabo grafičnega procesorja. Vmesnik deluje neodvisno od izbire programskega jezika in je podprt na več platformah [2].

2.3 Barnes-Hut algoritem

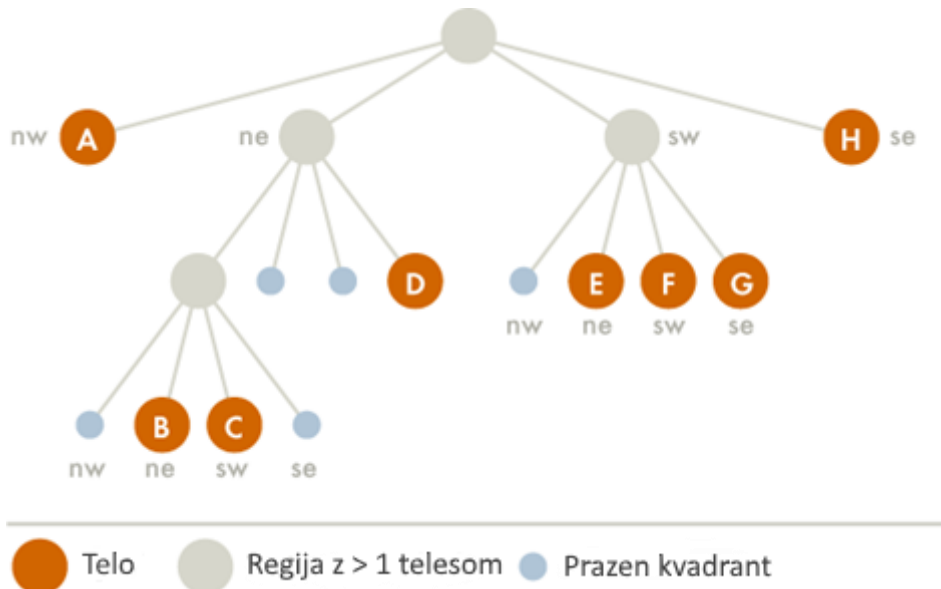
Josh Barnes in Piet Hut sta se domislila učinkovite sheme, kako različna telesa, ki so dovolj blizu skupaj, združiti v eno telo. To je ključna ideja, kako pospešiti algoritme z množico teles, ki vplivajo druga na drugo. Če smo v točki prostora dovolj daleč stran od gruče teles, lahko dejanski gravitacijski vpliv gruče (približno) izenačimo z gravitacijskim vplivom telesa, ki ima skupno maso gruče in se nahaja v njenem težišču.

V 3D-prostoru, recimo v kocki s stranico a , imamo razporejenih n teles. Kocko razdelimo na 8 manjših enako velikih kock s stranico $a/2$ in vsaka taka kocka predstavlja nov podprostor. Vsak podprostor, v katerem se nahaja več teles, rekurzivno delimo naprej na 8 nadaljnjih podprostorov, dokler se vsa telesa ne razporedijo vsako v svoj podprostor. Delitev v dveh dimenzijah poteka tako, kot je prikazano na sliki 1. Algoritem gradi drevesno strukturo, kjer podprostori predstavljajo vozlišča, iz katerih se spušča 8 vej do novih vozlišč en nivo nižje (4 veje v 2D, slika 2). Zapolnjena vozlišča na zunanjih položajih drevesa predstavljajo posamezna telesa, vozlišča znotraj drevesa pa predstavljajo skupino teles pod njim in hranijo lego težišča in skupno maso.

Pri izračunu težnega pospeška na izbrano telo se postopek začne pri korenu drevesa (prvi vozlel – cel prostor) in se nato spušča po nivojih. Za vsak vozlel se preveri, ali je težišče skupine, ki jo predstavlja, dovolj daleč od izbranega telesa. Če je pogoj izpolnjen, se ves del drevesa pod vozliščem obravnava kot eno telo s skupno maso v težišču te skupine. Kriterij, ali je vozlišče dovolj oddaljeno od telesa, določa kvocient s/d , kjer je s značilna dimenzija podprostora, ki ga vozlišče predstavlja (običajno stranica kocke), d pa razdalja med izbranim telesom in težiščem skupine teles pod vozliščem. Kvocient nato primerjamo z referenčno vrednostjo θ . Če je $s/d < \theta$, potem je vozlišče dovolj oddaljeno za združevanje. S spreminjanjem parametra θ vplivamo na hitrost in natančnost simulacij. Za gravitacijsko interakcijo se običajno vzame vrednost $\theta = 0,5$. Če θ spuščamo proti nič, potem je združevanja vse manj, čas simulacij se podaljšuje, povečuje pa se natančnost.



Slika 1: Razdelitev prostora na podprostore v 2D



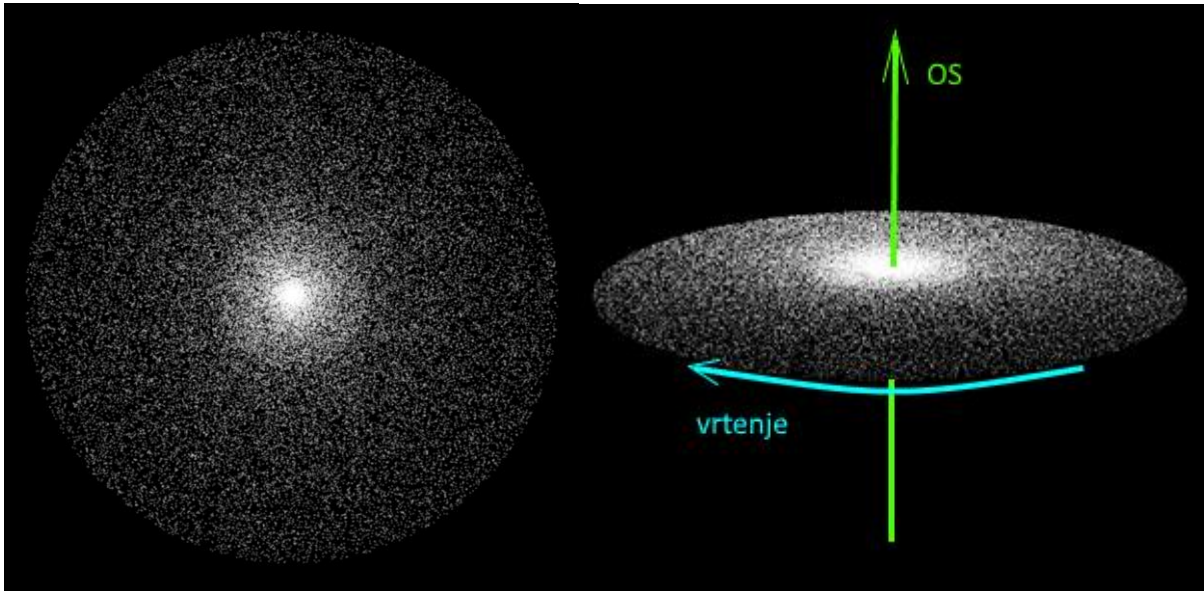
Slika 2: Drevesna shema algoritma v 2D

Pri računanju vseh medsebojnih interakcij med n telesi je potrebno opraviti n^2 izračunov. Pri deset tisoč telesih to pomeni sto milijonov izračunov za en korak. Algoritem Barnes-Hut za isti problem potrebuje $n \cdot \log_2 n$ izračunov. Pri deset tisoč telesih torej le 133 000 izračunov, kar je 750-krat manj [3, 4].

3 METODE DELA

Program je napisan v programskem jeziku C++ v okolju Visual studio 2015. Program ima več faz. Najprej moramo odpreti grafično okno na zaslonu računalnika in v 3D-prostor, ki ga okno predstavlja, postaviti telesa. Najbolj smiselni začetni postavitvi glede na obliko galaksij se mi zdita razporeditev teles znotraj kroga ali znotraj krogle. Razporeditev teles po krogu lahko predstavlja disk spiralne galaksije (v nadaljevanju disk), razporeditev teles po krogli pa

kroglasto kopico ali eliptično galaksijo. Telesa sem običajno naključno porazdelil po ploskvi znotraj diska oziroma po notranjosti krogle. To sem storil tako, da sem naključno izbral dve števili med 0 in 1. Prvo sem pomnožil s polmerom, drugo pa z 2π . Tako sem dobil polarni koordinati točke na disku, kamor sem postavil telo. Dobljeno porazdelitev, kjer gostota teles pada od sredine proti robu, prikazuje slika 3a. Podoben, a ustrezno prilagojen postopek sem uporabil tudi pri krogli [5].



Slika 3a, b: Disk s 50 000 teles, pogled od zgoraj in pogled s strani

Telesa lahko v začetnem položaju mirujejo, lahko pa se jim določi začetno hitrost. Telesom, razporejenim po disku, sem določil začetno hitrost tako, da se glede na navpično os skozi središče vsa telesa na začetku vrtijo v smeri urinih kazalcev (slika 3b). Za vsako telo leži vektor hitrosti v ravnini diska in je pravokoten na zveznico med izbranim telesom in središčem diska. Po velikosti pa sem preizkušal različne možnosti, tako enako hitrost za vsa telesa kot enakomerno in neenakomerno naraščanje hitrosti od sredine proti robu.

Skupaj z začetnim položajem določimo tudi maso telesa. Vsem telesom sem najprej določil isto maso, kasneje pa sem jo določal naključno, sem pa omejil razmerje med največjo in najmanjšo maso na 10. Izvajal sem tudi simulacije, pri katerih sem dodatno v prostor (v sredino ali še boljje na rob) postavil telo z zelo veliko maso (recimo toliko kot vsa ostala telesa skupaj) po analogiji supermasivnih črnih lukenj.

Med drugim lahko med začetne pogoje štejem odločitve, ali med simulacijo dopustimo trke med telesi in s tem »lepljenje« teles ali ne. Najpreprostejša možnost je, da so vsa telesa točkasta in do trkov ne prihaja. Realno, če pogledamo trk dveh galaksij, do trkov praktično ne prihaja, ker so razdalje med zvezdami zelo velike v primerjavi z velikostjo zvezd. Če pa pogledamo formacijo sončnega sistema, so trki v tem procesu nujni. Tako sem preveril tudi možnost, da se dve telesi, ki se približata na neko minimalno razdaljo, zlepi. Pri tem se masi seštejeta, hitrost novega telesa pa se določi z ohranitvijo gibalne količine.

Ko določimo vse začetne pogoje, program izvaja algoritem Barnes-Hut. Program dopušča globino do 20 nivojev, kar pomeni do 8^{19} vozlišč. Pri ciljnem številu nekaj deset tisoč teles taka globina več kot zadošča. Za vsako telo se izračuna težni pospešek \vec{g} , ki je vektorska vsota posameznih težnih pospeškov, ki jih povzročajo vsa ostala telesa in se izračunajo po enačbi 1.

Bližnja telesa so obravnavana samostojno, bolj oddaljena pa algoritem združuje. Iz znanih začetne lege, začetne hitrosti in težnega pospeška, ki se v (kratkem) časovnem intervalu Δt obravnava kot konstanten, se izračunata nova lega telesa

$$\vec{r} = \vec{r}_0 + \vec{v}_0 \cdot \Delta t \quad (3)$$

in hitrost v naslednjem koraku

$$\vec{v} = \vec{v}_0 + \vec{g} \cdot \Delta t \quad (4).$$

Začetne polarne koordinate teles se pretvorijo v kartezične in v kartezičnem koordinatnem sistemu potekajo vsi nadaljnji izračuni. Nove lege teles se sproti nalagajo v grafični spomin, na zaslonu se točke prestavijo v novo lego in lahko se začne nov korak. Človeški vid smatra gibanje za zvezno, če se slika osveži vsaj 12-krat v sekundi, pri nižjih frekvencah osveževanja pa si lahko pomagamo s snemalnikom zaslona in potem posnetek hitreje zavrtimo. Proces, ki so predmet simulacij, v vesolju trajajo na milijone let. Da sem pospešil potek dogajanja v simulacijah, sem vzel točkasta kilogramska telesa na metriških razdaljah in povečal gravitacijsko konstanto G za faktor sto tisoč. Med izvajanjem simulacij je omogočeno spreminjati perspektivo s tipkovnico in z miško. Objektu se lahko približamo, ga zavrtimo in nanj pogledamo s katere koli strani.

4 REZULTATI

4.1 Izboljševanje hitrosti simulacij

Pri pisanju programa sem imel v mislih predvsem pravilnost delovanja. Ko pa so prve simulacije potekale zelo počasi že pri postavitvi nekaj sto teles, sem bil prisiljen razmišljati tudi o optimizaciji postopkov. Okolje Visual studio 2015 omogoča pregled nad tem, koliko časa porabi računalnik za izvedbo posameznih delov programa. Tako sem lahko učinkovito iskal programsko kodo, ki je po nepotrebnem upočasnjevala izvajanje simulacij. Ugotovil sem, da se porabi veliko časa, ko podprogrami za svoje interno delovanje podvajajo spomin za globalne spremenljivke. Z uporabo kazalcev namesto spremenljivk sem uspel hitrost simulacij zvišati skoraj za faktor sto. Posledično sem lahko povečal število teles v simulacijah na deset tisoč. Začel sem z razporeditvijo teles iste mase po disku, kot kaže slika 3a. Sprva sem spremljal dinamično dogajanje na zaslonu, podrobnejša analiza pa je pokazala, da dogajanje ni realno. Že pri začetni hitrosti nič je na stotine teles v prvih sekundah simulacije pobegnilo iz sistema z veliko hitrostjo. Kasneje se je dogajanje pričakovano umirilo in formirala se je osrednja zgoščina. Ugotovil sem, da pobegla telesa prihajajo iz pasu ob centralni zgoščini. Ko se ta telesa približujejo središču, se težni pospešek hitro zmanjšuje proti nič. Ker pa sem imel v simulaciji nastavljen predolg časovni korak Δt , ko naj bi pospešek smatrali za konstanten, so ta telesa dobila nerealno veliko hitrost, ki jih je posledično odnesla iz sistema. Zato sem Δt zmanjšal za 10-krat, vendar sem s tem za isti faktor upočasnil simulacije. Te so sedaj potekale natančneje in brez opaznih nelogičnosti, na zaslonu pa sem lahko spremljal dinamiko od 5 do 10 slik na sekundo, odvisno od razporeditve teles. Diagnostika računalnika (*task manager* v operacijskem sistemu Windows) med simulacijami je pokazala, da je procesorska enota obremenjena samo 25 odstotno. Ugotovil sem, da se je s problemom ukvarjalo samo eno izmed štirih jeder procesorja. V navodilih za C++ (<http://www.cplusplus.com/reference/>) sem našel napotke, kako vzporedno med jedra procesorja porazdeliti delo. Ustrezne funkcije so že del standardnih knjižnic. Po uspešni implementaciji novih znanj se je hitrost simulacij potrojila. Računalnik

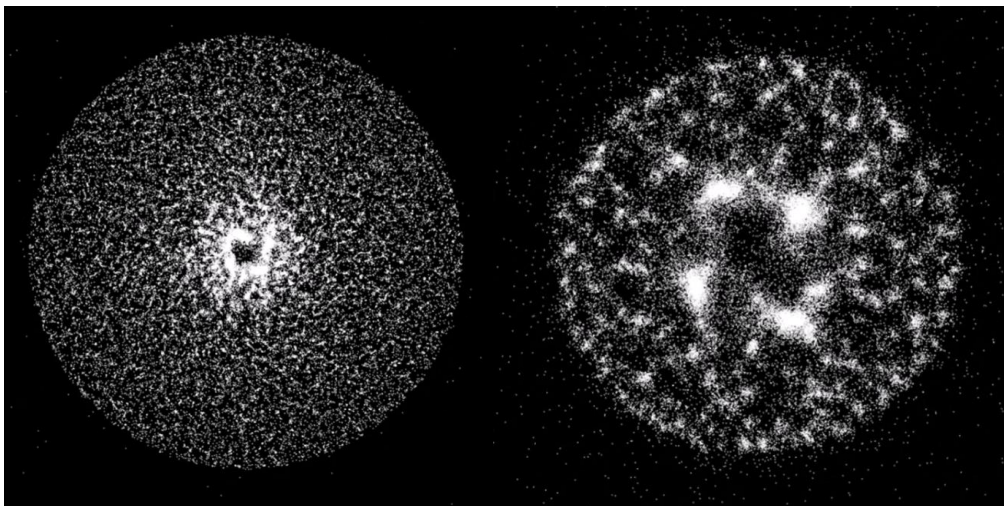
lahko eno jedro obremeni do konca, pri obremenitvi vseh štirih pa si še vedno pusti malo rezerve. Na hitrost verjetno vpliva tudi časovno usklajevanje dela med jedri. Hitrost se je zato povečala za faktor tri in ne za štiri. Končni rezultat je, da pri deset tisoč telesih v disku procesor (Intel i5) zmore 26 slik na sekundo (FPS), pri deset tisoč telesih znotraj krogle z enakim polmerom pa 16 slik na sekundo. Pri disku gre hitreje najbrž zato, ker so telesa porazdeljena bolj na gosto in je več združevanja (Barnes-Hut algoritem, poglavje 3.2). Uporaba snemalnika zaslona hitrost simulacij zniža za približno dvajset odstotkov.

4.2 Potek simulacij

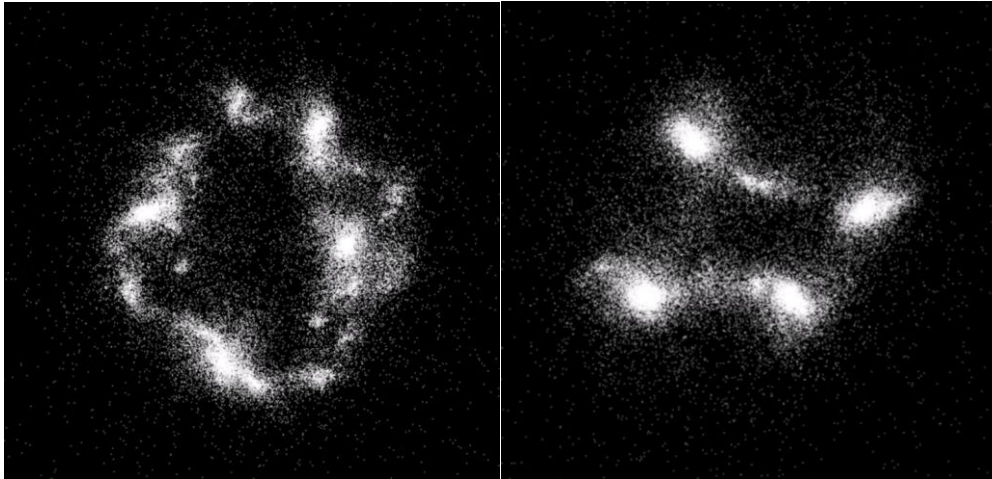
Začetni pogoji ponujajo praktično neskončno različnih možnosti pri simuliranju. Primeri v nadaljevanju po moji oceni predstavljajo izbor najbolj karakterističnih, najzanimivejših in tudi najzahtevnejših simulacij. V vseh primerih je teles vsaj petdeset tisoč. Večje število teles pomeni, da je kakovost slike po prenosu zaslona na papir bistveno boljše in omogoča vsaj približno predstavo o trenutni razporeditvi. Simulacije so bile zato počasne, v večini primerov jih je bilo treba pustiti teči več kot eno uro, celotno dogajanje pa sem posnel s snemalnikom zaslona. Videi, ki predvajajo hitre posnetke simulacij, so v prilogi na DVD-ju.

4.2.1 Razporeditev teles po disku

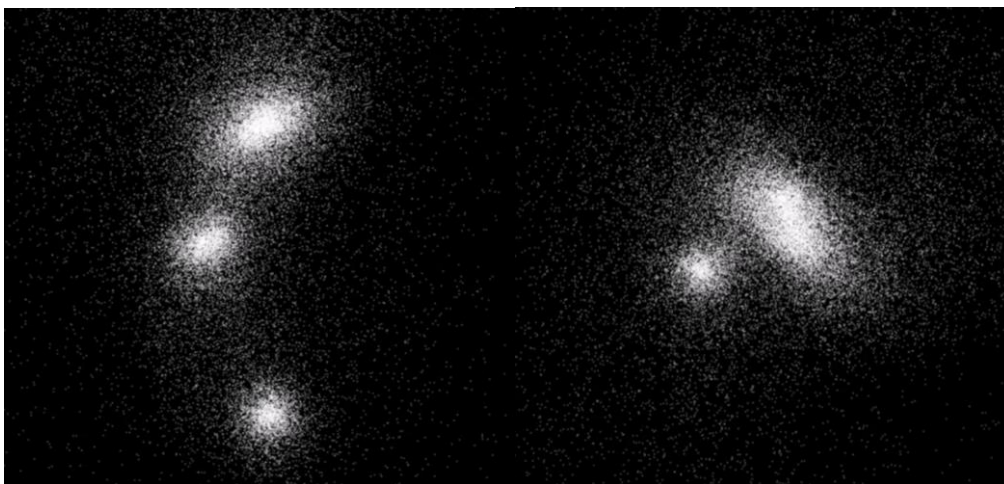
50 000 teles istih mas sem razporedil po disku tako, kot je prikazano na sliki 3a. Velikost hitrosti je enaka za vsa telesa, disk pa se vrti v smeri urinih kazalcev (poglavje 3, slika 3b). Rezultat takega pogoja je, da pri telesih blizu središča, kjer je težnost majhna, prevlada centrifugalna sila in ta telesa se začnejo premikati po spirali navzven (praznina v središču), medtem ko pri telesih na obrobju diska prevlada težnost in se začnejo premikati po spirali navznoter (disk se manjša, sliki 4a, b). Blizu krožnice s polovičnim polmerom prvotnega diska se telesa zgostijo v neenakomeren obroč (slika 5a), čez čas prevladajo štiri večje zgoščine (slika 5b), ki se postopoma ena po ena sesedajo v osrednjo zgoščino eliptične oblike (sliki 6a, b).



Slika 4a, b: Praznina v sredini in sesedanje diska

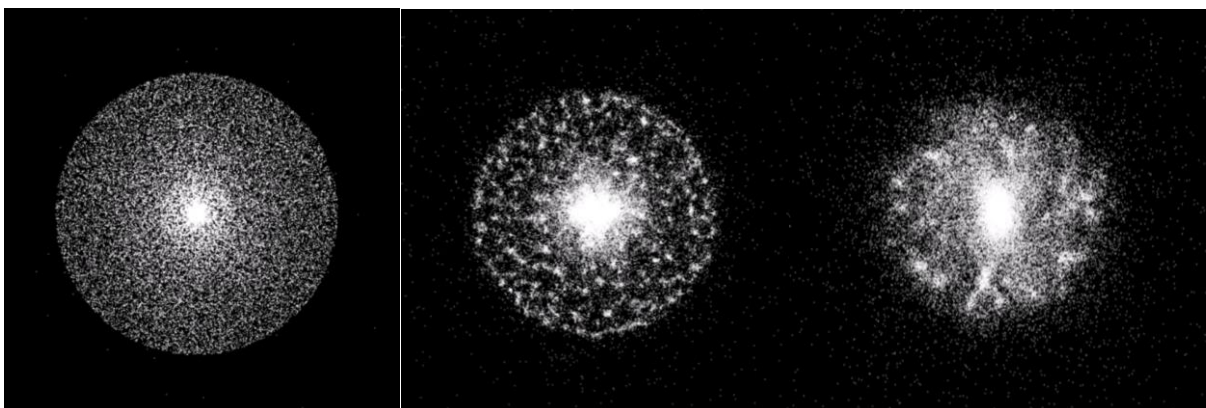


Slika 5a, b: Obroč in štiri lokalne zgoščine

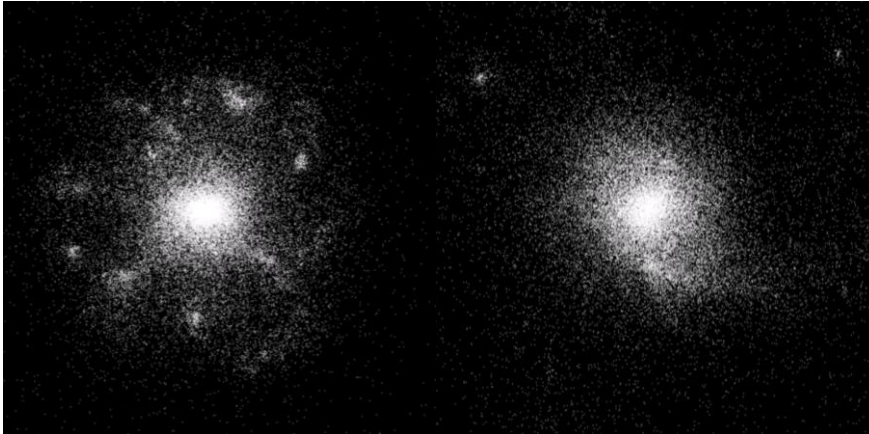


Slika 6a, b: Postopno sesedanje v enotno zgoščino

V naslednji simulaciji sem spremenil velikost hitrosti. Ta sedaj ni povsod enaka, ampak se linearno spreminja od nič v sredini do maksimalne vrednosti na robu. Na robu diska je ista hitrost kot v prvem primeru. Disk se hitro seseda proti sredini, s tem pa se večja hitrost teles zaradi ohranitvenih zakonov (energija, vrtilna količina, slike 7a, b, c). Centrifugalna sila zaustavi sesedanje, nekaj snovi se začasno oddalji, nato pa se počasi ponovno pridruži večini v centralni zgoščini (sliki 8a, b).



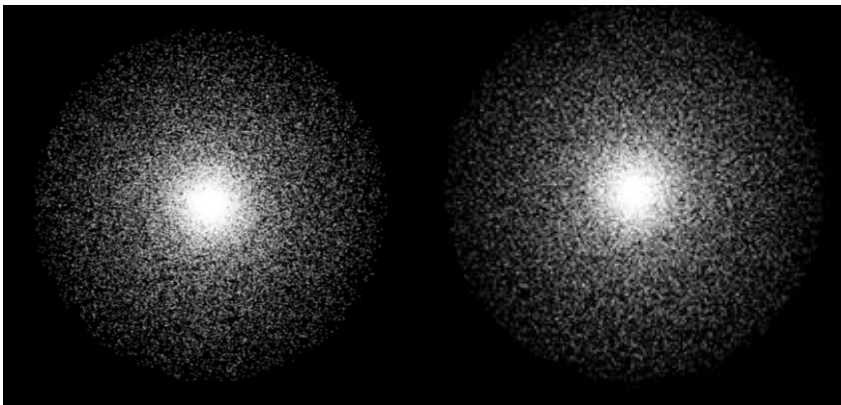
Slika 7a, b, c: Gravitacijsko sesedanje diska



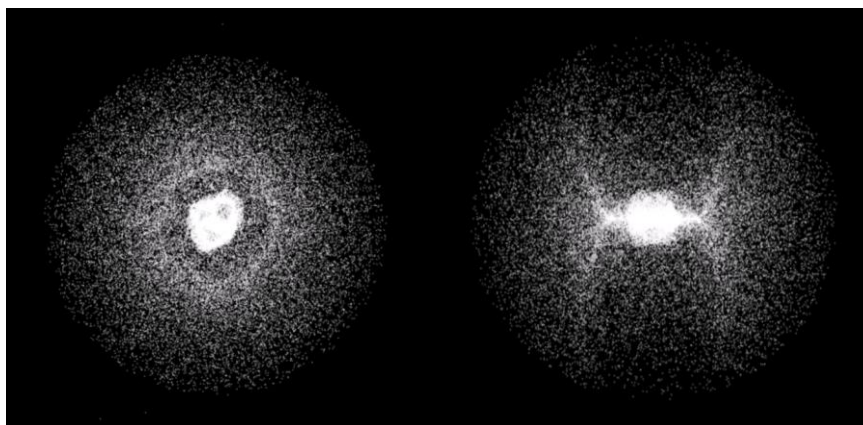
Slika 8a, b: Začasna razširitev in ponovna zgostitev v središču

4.2.2 Razporeditev teles po krogli

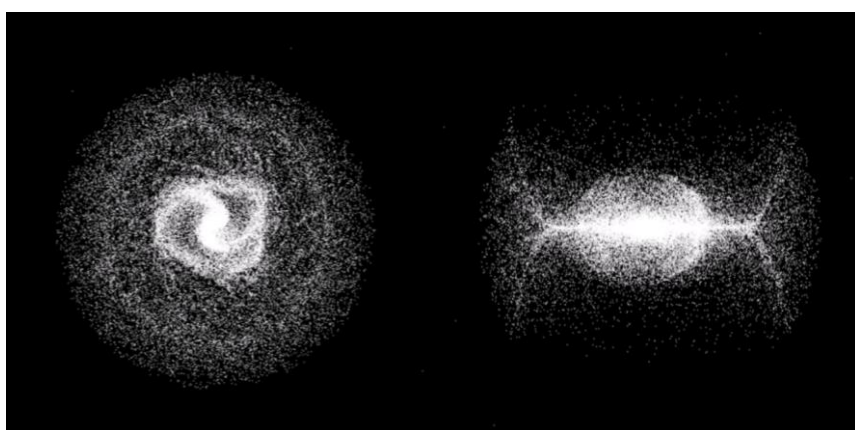
50 000 teles istih mas in istih velikosti hitrosti z naključno razporeditvijo po notranjosti krogle sem pognal v taki smeri, da se pri pogledu od zgoraj na kroglo vsa telesa vrtijo v isto smer okoli navpične osi skozi središče. Slike od 9 do 14 so združene v pare, kjer leva slika prikazuje pogled na dogajanje od zgoraj, desna slika pa pogled v približno istem trenutku s strani. Pogled na začetno postavitev je skoraj enak z obeh perspektiv (sliki 9a, b), razlike so malenkostne zaradi naključne razdelitve. Po zagonu simulacije opazimo, da se telesa v sredini najprej združijo v jedro (slika 10a), pogled s strani pa odkrije, da se tudi zgornji in spodnji del krogle sesedeta proti jedru (slika 10b). Telesa daleč od osi (ob ekvatorju) se gravitaciji močnejše upirajo, vseeno pa se doseže največja zgostitev na slikah 12a in 12b. V nadaljevanju lahko opazujemo vrtenje dveh spiralnih krakov (slika 13a), dokler ne zbledita v končni kroglasti obliki.



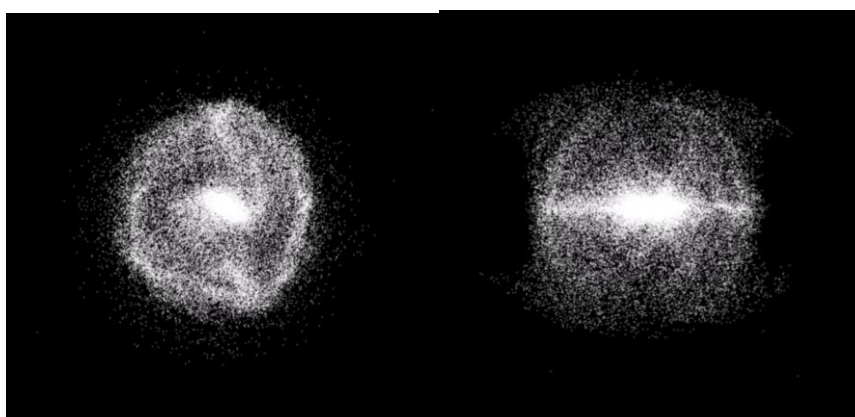
Slika 9a, b: Začetna krogelna postavitev



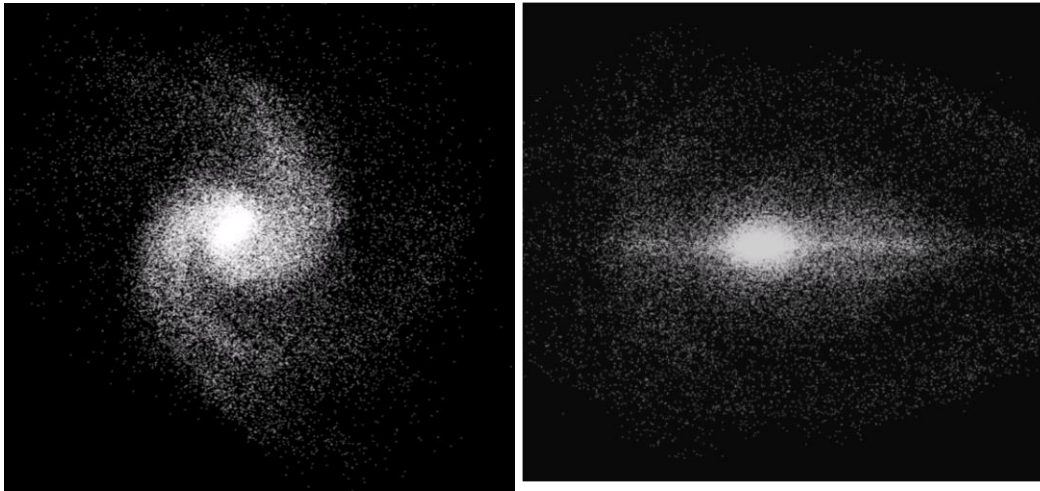
Slika 10a, b: Vrh in dno krogle padeta skupaj



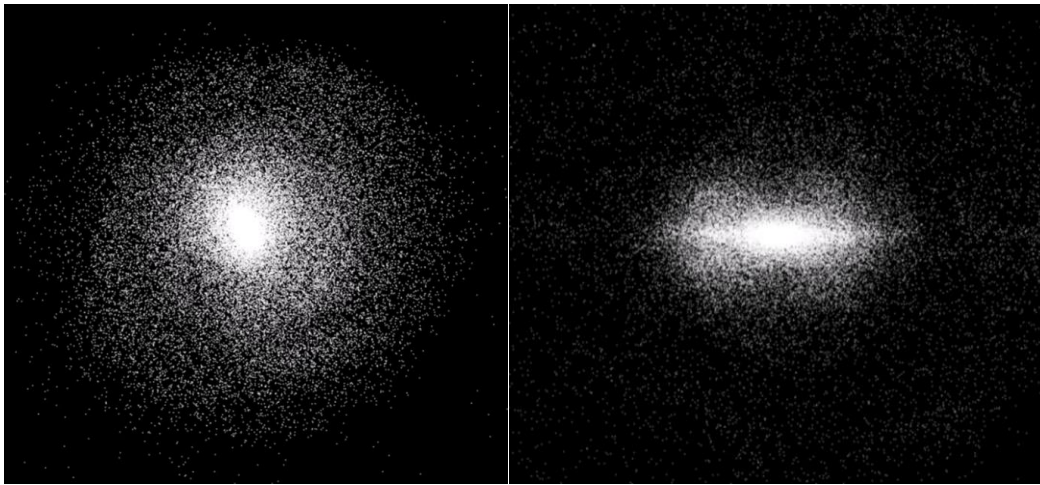
Slika 11a, b: Iz jedra se razvijeta manjša kraka



Slika 12a, b: Največja zgoštev



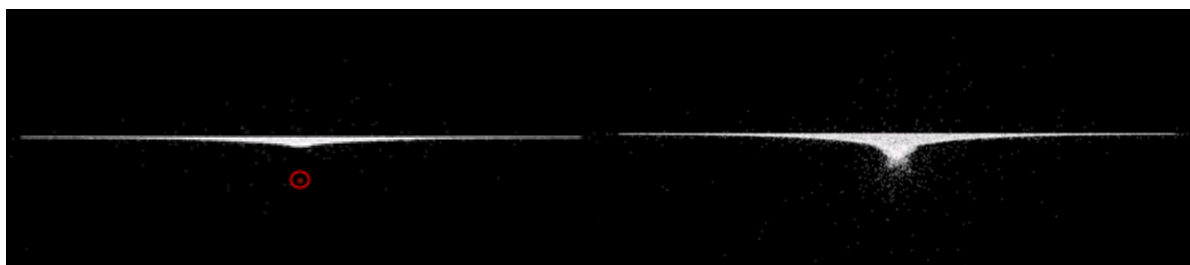
Slika 13a, b: Močna spiralna kraka



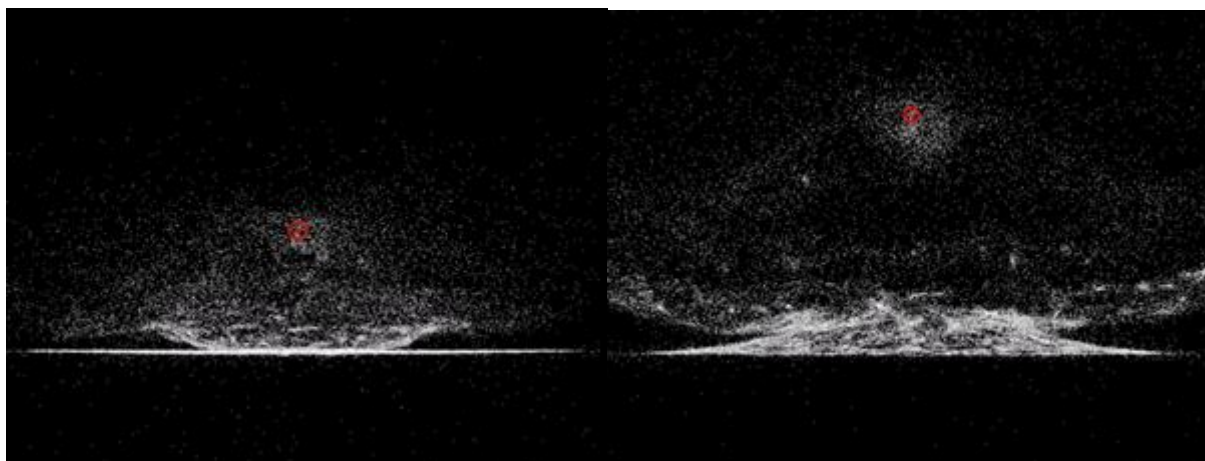
Slika 14a, b: Končna oblika

4.2.3 Disk in črna luknja

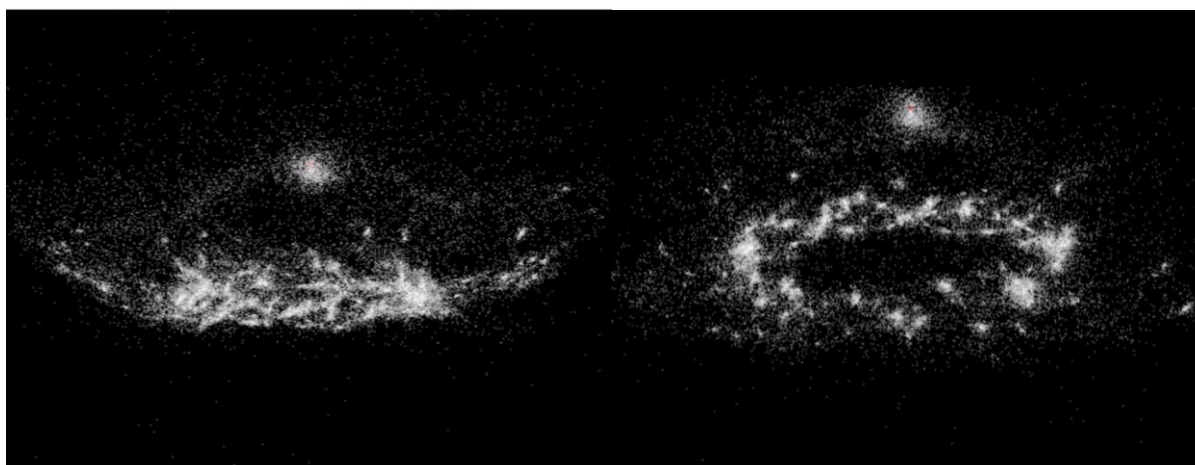
Zanimalo me je, kolikšno hitrost bi morala imeti telesa v disku, da bi vsaj nekaj časa obdržal svojo značilno obliko. Preizkusil sem različne možnosti, kot najbolj zadovoljivo pa sem izbral naraščanje hitrosti od sredne proti robu po funkciji, ki je sorazmerna s kvadratnim korenom iz radija. Proti disku s 50 000 teles sem nato iz bližnje okolice poslal točkasto telo z maso 20 000 teles, ki predstavlja (supermasivno) črno luknjo. V obeh primerih, ko črna luknja potuje skozi sredino diska v navpični ali vodoravni smeri, pride do popolnega uničenja oblike diska. Na tisoče teles se razprši po prostoru, okrog črne luknje pa se ustvari zgoščina hitro krožečih zvezd. Lega črne luknje je na večini slik od 15 do 20 označena z rdečim krožcem. Slike od 15 do 17 prikazujejo primer, ko se črna luknja v navpični smeri približa sredini diska, ki leži v vodoravni ravnini. Slike od 15a do 17a prikazujejo pogled s strani, pri sliki 17b pa je »kamera« nekoliko dvignjena nad ravnino diska. Črna luknja že med približevanjem upogne disk proti sebi, tako da ta dobi obliko plitkega lijaka. Po trku se večina teles z območja jedra diska razbeži navzgor in navzven, v sredini pa zeva velika praznina, lepo vidna na sliki 17b. V nadaljevanju simulacije črna luknja z rojem ujetih zvezd odtava proč, preostala telesa pa se brez osrednje zgoščine, ki bi jih držala skupaj, porazgubijo po prostoru.



Slika 15a, b: Približevanje črne luknje disku in trenutek trka

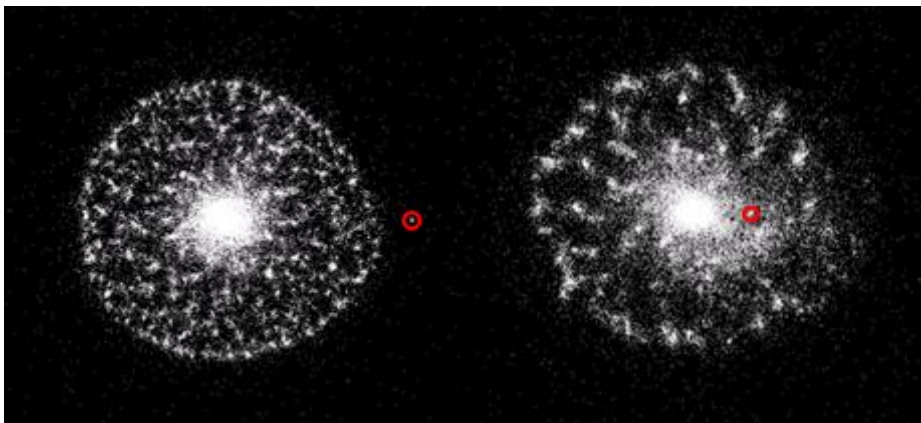


Slika 16a, b: Uničenje jedra diska

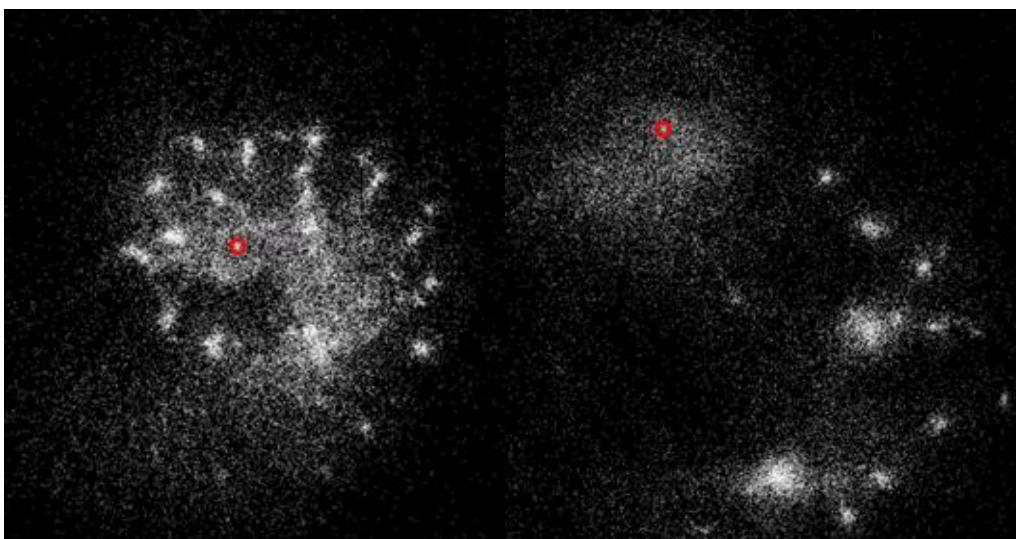


Slika 17a, b: Pogled z oddaljenejši perspektive. Črna luknja je rdeča pika sredi zgoščine na vrhu slike

Slike 18a, b in 19a, b prikazujejo primer, ko leži črna luknja v ravnini diska in z leve strani slike potuje proti sredini diska. Slike prikazujejo pogled od zgoraj. Na sliki 18b še opazimo izrazito jedro diska, na sliki 19a pa to ne obstaja več. Črna luknja je telesa iz jedra razpršila v mogočno polžasto obliko in od diska ostane samo nekaj nepovezanih lokalnih zgoščin.

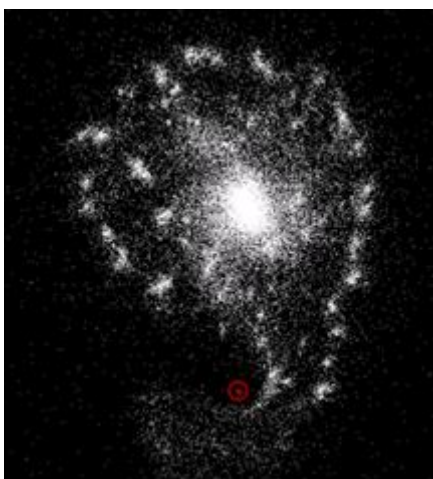


Slika 18a, b: Približevanje črne luknje jedru diska



Slika 19a, b: Polžasta oblika in nekaj preostalih lokalnih zgoščin

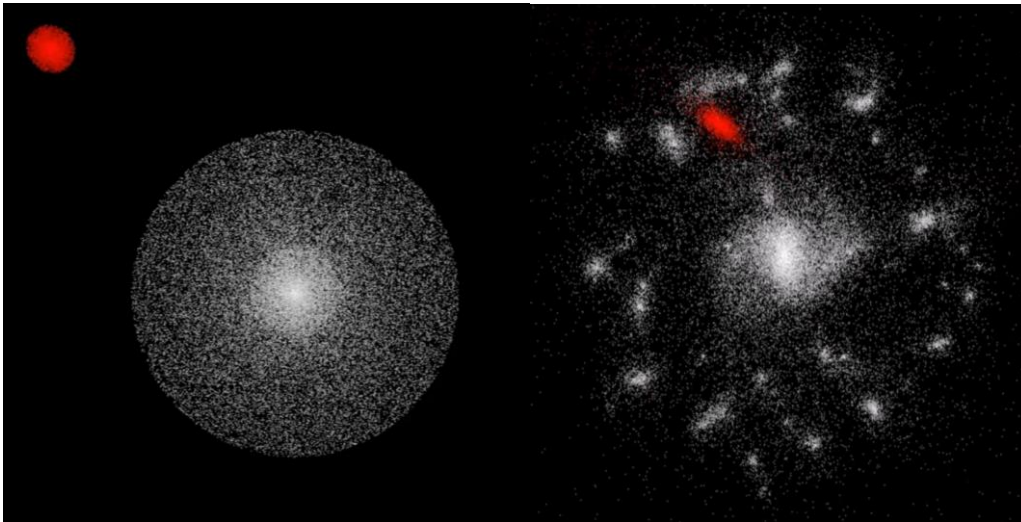
Črno luknjo sem poslal še mimo diska tako, da se je nekaj časa gibala po njegovem robu (slika 20). V tem primeru jedro »preživi«, črna luknja pa na disku vseeno pusti močan pečat.



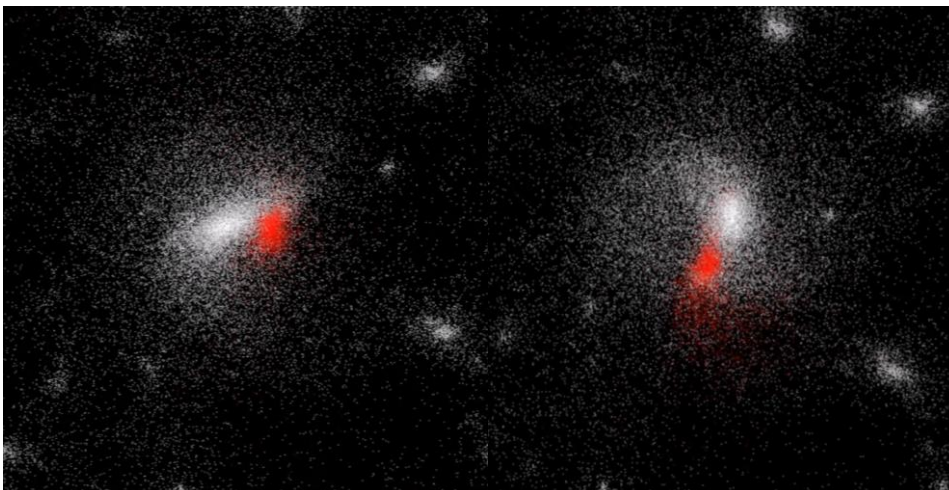
Slika 20: Črna luknja potegne telesa z roba diska v »slonji rilec«

4.2.4 Trk diska in krogle

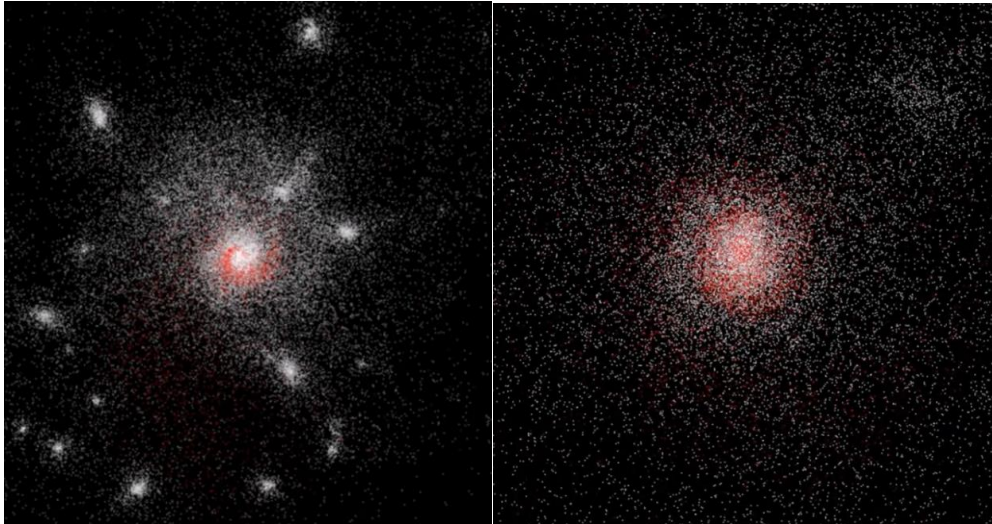
V 3D-sceni sem v sredino postavil disk s 50 000 teles, na obrobje pa sem postavil še 10 000 teles, razporejenih v kroglo. Telesa v krogli sem obarval rdeče. Začetna postavitev je vidna na sliki 21a, ki prikazuje pogled od zgoraj na ravnino diska. Središče krogle leži v isti ravnini. S takšno postavitvijo sem se hotel približati trku dveh galaksij, recimo večje spiralne in manjše eliptične. Telesa iz diska in krogle sem obarval različno, da bi iz simulacij lažje ocenil, na kak način poteka trk in kako se razporedijo telesa iz krogle v novi združbi. Jedro krogle se sprva počasi približuje jedru diska (slika 21b), po stiku obeh jeder (slika 22a) pa dogajanje postane bolj dinamično. Jedri se začneta pospešeno vrteti v smeri urinih kazalcev. Vedno večja centrifugalna sila začne razprševati šibkeje vezana telesa in obe jedri dobita širok rep (slika 22b). Ker pa jedro krogle vsebuje bistveno manj teles od jedra diska, se prej razprši med gibanjem po spirali (slika 23a).



Slika 21a, b: Začetna postavitev in približevanje (rdeče) krogle jedru (belega) diska



Slika 22a, b: Bližja perspektiva stika obeh jeder in začetek pospeševanja v smeri urinih kazalcev



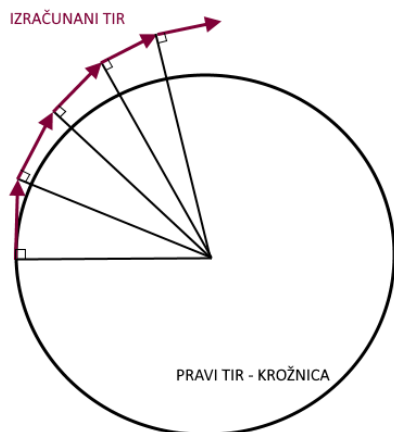
Slika 23a, b: Rdeča spirala od daleč in blizu

Slika 23b prikazuje približno jedro. Na sliki je še vidna spirala, ki pa čez čas izgine, ko se bela in rdeča telesa pomešajo v novo nastalo elipsoidno zgoščino.

5 RAZPRAVA

V simulacijah prihaja do primerov, ko dve telesi v paru z veliko hitrostjo v nasprotnih smereh pobegneta iz ustvarjenega 3D-prostora. To se zgodi takrat, ko se telesi zelo približata in se pospešek v kratkem času (krajšem od časovnega koraka Δt , poglavje 3) močno spremeni. Posledično je izračunana hitrost teles napačna (enačba 3). Telesi lahko dobijo nerealno veliko hitrost, ki ju odnese iz prostora. Število pobegov se zmanjša, če se skrajša Δt , vendar se s tem dodatno upočasnijo simulacije. V predstavljenih primerih je ta pobeg teles prisoten, vendar v omejenem številu, ki ga ocenjujem na 1 % glede na število vseh teles v simulaciji. Nekaj testnih simulacij sem pognal s štirikrat krajšim Δt , vendar je bil razvoj globalne slike praktično enak, zato sem Δt vrnil na prvotno vrednost.

Za zastavljeni problem nisem našel nobene referenčne situacije, kjer bi natančno poznal tako začetne pogoje kot končni rezultat. Tako ne morem zanesljivo vedeti, kako točne so predstavljene simulacije. Vemo, da simulacije predstavljajo približek resničnega dogajanja, zato je vsak naslednji računski korak nekoliko manj natančen. Računalniški program računa premike teles po najenostavnejši metodi numerične integracije (enačbi 3 in 4), za katero je značilno, da zaradi nenatančnosti povečuje skupno energijo sistema teles. Slika 24 prikazuje, kako program računa gibanje telesa po krožnici. Po izračunih se telo dejansko giblje po spirali navzven, a če je časovni korak kratek, je to opazno šele po večjem številu obhodov. Posledica takega gibanja je povečevanje potencialne energije telesa. Zato sem se odločil, da preverim, kako je s skupno energijo vseh teles med simulacijami. Skupna energija je vsota potencialne in kinetične energije in je teoretično ves čas konstantna. Potencialno energijo sem računal po enačbi 2, kinetično energijo pa po enačbi $W_k = \frac{mv^2}{2}$. Potencialna energija je negativna, kinetična pa pozitivna in dokler je skupna energija negativna, je sistem gravitacijsko vezan. Za preizkus sem vzel disk z 10 000 telesi, začetna hitrost teles pa je naraščala od sredine proti robu sorazmerno s kvadratnim korenem iz radija.



Slika 24: Izvor napak pri računanju tira telesa

Največjo hitrost sem omejil na tako vrednost, da se je disk po zagonu simulacije začel gravitacijsko sesedati, podobno kot prikazujejo slike od 7a do 8b. Program sem priredil tako, da je na vsako sekundo izpisal vrednost skupne kinetične in skupne potencialne energije. Po zagonu sem pustil simulacijo teči tako dolgo, da se je slika na zaslonu ustalila in smo lahko spremljali samo še dinamiko teles, ki jo je narekovala nastala osrednja zgoščina. Analiza vrednosti je pokazala, da se je skupna vrednost energije v tem času povečala za 35 %. Pri ogledu simulacije se to kaže na tak način, da se telesa postopoma izmikajo težnosti osrednje zgoščine in odtavajo ven iz 3D prostora. Število teles v simulaciji se zato zmanjšuje in če bi pustili simulacijo teči dovolj dolgo časa (recimo 10 ur), bi najbrž lahko dočakali razpad osrednje zgoščine in odhod vseh teles iz prostora. Največja dinamika v predstavljenih primerih se dogaja kmalu po zagonu simulacij, ko je sprememba skupne energije še majhna, zato so tudi poskusne simulacije s krajšim časovnim korakom potekale na enak način. Razlike bi se pokazale verjetno po daljšem časovnem obdobju, ampak do tedaj se najbolj zanimiv del dogajanja že odvije. Obstajajo tudi natančnejše metode numeričnega integriranja, ki pa so matematično in računalniško zahtevnejše in jih v tej fazi še nisem preučil.

Poleg Δt na natančnost izvajanja simulacij vpliva še parameter θ v algoritmu Barnes-Hut. V predstavljenih primerih je njegova vrednost vedno priporočljivih 0,5. V nadaljevanju bi bilo dobro preveriti, kako spremembe tega parametra vplivajo na potek in natančnost simulacij. Pravzaprav je idej, kaj vse bi še lahko počel, ogromno. Lahko bi se poglobil v zgradbo in kroženje teles v galaksiji in poskusil ustvariti stabilno spiralno galaksijo. Potem bi lahko simuliral galaktične trke v treh dimenzijah, dodajal črne luknje, temno snov. Glavno oviro za te načrte predstavlja hitrost osebnih računalnikov. Smiselno bi bilo precej povečati število teles v simulacijah, recimo na več sto tisoč, jih razporediti v galaksijo in potem poiskati za to postavitev optimalna parametra θ in Δt . To pa pomeni, da bi simulacije trajale zelo dolgo, verjetno dneve, in pojavile bi se lahko nove težave na primer s snemalnikom zaslona, ki ni prilagojen za tako dolgotrajno snemanje.

Poizkusil sem tudi z združevanjem teles. Telesi, ki se dovolj približata drugo drugemu, sem združil v novo telo s skupno maso. A ta sprememba ni bistveno vplivala ne na število pobeglih teles ne na potek simulacij. V nekaj testnih simulacijah sem preizkusil tudi različne mase teles. Razmerje med največjo in najmanjšo maso sem omejil na 10. Tudi te simulacije niso bistveno odstopale od predstavljenih simulacij s telesi z isto maso, vendar je tu še možnost za nadaljnje raziskovanje.

6 ZAKLJUČEK

Mislil sem, da sem nalogo, ki sem si jo zadal, uspešno opravil. Pri deset tisoč telesih v gravitacijski interakciji je dinamika osveževanja slike na zaslonu dovolj visoka, da lahko proces brez težav spremljamo »v živo«. Pri petdeset tisoč telesih pa sem uporabljal snemalnik zaslona *Icecream Screen Recorder* ter nato v programu *Movie maker* naredil videoposnetek z 32-kratno hitrostjo predvajanja. Ocenjujem, da je petdeset tisoč teles optimalno število za simulacije z osebnim računalnikom glede na simulacijski čas in kakovost slike. Napisan računalniški program predstavlja dobro osnovo za postopno približevanje realnejšim situacijam kar se tiče števila teles in začetnih pogojev, vendar pa je vzporedno s tem potrebno razmišljati o možnostih uporabe hitrejših računalnikov in natančnejših numeričnih metod integracije.

Pri izdelavi naloge sem pridobil ogromno novih znanj in programerskih veščin. Z matematičnega področja sem moral usvojiti znaten del linearne algebre, da sem razumel, kako deluje 3D grafika. Pri računalništvu sem usvojil tehniko ustvarjanja v 3D-grafiki ter vrsto praktičnih znanj tako v zvezi s programskim jezikom C++ kot z delovanjem računalnika. Pri fiziki sem spoznal metodo Barnes-Hut, ki zmanjšuje število potrebnih izračunov za hitrejši potek simulacij, ob njihovem spremljanju pa sem poglobil tudi svoje razumevanje gravitacije. Izmed predstavljenih primerov me je navdušilo gravitacijsko sesedanje teles, porazdeljenih po krogli, najbolj pa me je presenetilo popolno uničenje, ki ga za sabo pusti črna luknja po prehodu skozi disk.

7 POVZETEK

Galaksije so osnovni gradnik v strukturi vidnega vesolja. Dinamiko teles, ki tvorijo galaksijo, usmerja gravitacijska interakcija med njimi. Gravitacija je s fizikalnega in matematičnega stališča dobro poznana, a enormno število teles v galaksiji vseeno povzroča probleme. Vsako telo z gravitacijsko silo vpliva na vsa ostala telesa in s tem vpliva na njihovo gibanje. Pri sto milijardah zvezd v galaksiji je natančen izračun vseh interakcij in premikov nemogoč tudi z najboljšimi računalniki na svetu, zato se za simuliranje procesov, kot je vrtenje galaksij ali galaktični trk, uporabljajo različne metode poenostavljanja. Pri obravnavi gravitacijske interakcije med n telesi se je najbolj prijel algoritem Barnes-Hut, ki združuje oddaljenejša telesa po določenem kriteriju. Za natančen opis je potrebno vsa telesa postaviti v 3D-prostor in jih v tem prostoru tudi premikati. Vse to pa bi radi spremljali na 2D-zaslonu iz določene perspektive. Zato se je veliko časa posvetilo tehnikam programiranja 3D-grafike. Vsa 3D-računalniška grafika temelji na transformiranju vektorjev z matrikami, zato je bilo nekaj matematičnih osnov linearne algebre zelo dobrodošlih za lažje razumevanje.

Cilj raziskovalne naloge je proučiti dogajanje, ko med vsaj 10 000 telesi po izbrani začetni postavitvi v 3D-prostoru deluje samo gravitacijska interakcija. To pa je mogoče doseči z računalniškim programom, ki združuje 3D-grafiko in Barnes-Hut algoritem v delujočo celoto. Programiranje v programskem jeziku C++ je potekalo v okolju Visual studio 2015 z vključitvijo programskega vmesnika OpenGL za 3D-grafiko. Po prvih uspešno izvedenih simulacijah na manjšem številu teles se je začelo obdobje optimiziranja programske kode za povečanje hitrosti. Postopno se je s pomočjo diagnostike računalnika odkrivalo, kje prihaja do nepotrebnih zastojev, zadnji pospešek pa je omogočila maksimalna obremenitev vseh štirih jeder

centralnega procesorja računalnika. To je prineslo željen rezultat: zvezno gibanje 10 000 teles z dovolj veliko dinamike, da je ogled simulacije zanimivo spremljati tudi v živo.

Zaradi doseganja boljših vizualnih učinkov so naslednje simulacije potekale s 50 000 telesi, porazdeljenimi naključno po ploščatem disku ali po notranjosti krogle. Čas simulacij se je zaradi tega podaljšal na dobro uro, zato se je dogajanje snemalo z zaslonskim snemalnikom. Iz teh posnetkov so se nato naredili videi z 32-kratno hitrostjo predvajanja. Na videih lahko spremljamo tvorbo lokalnih zgoščin, ki nato gravitirajo proti osrednji zgoščini – jedru, iz katerega se občasno vijejo spiralni kraki. Nadalje se je izvedel trk diska s točkastim telesom z veliko maso, ki znaša 40 % mase diska. To telo bi lahko predstavljalo črno luknjo. Centralni trk v vodoravni ali v navpični smeri pomeni popolno uničenje diska in na tisoče teles se razprši po prostoru. Le okoli črne luknje se ujame roj hitro krožečih teles. Za konec se je izvedel še poskus trka dveh galaksij: diska s 50 000 telesi bele barve in krogle z 10 000 telesi rdeče barve. Jedro krogle je za sabo pustilo lepo vidno rdečo spiralo, preden so se telesa obeh barv pomešala med seboj v združeno jedro. Dopustitev trkov dveh posameznih teles in njuno združitev v novo telo skupne mase ter testne simulacije z različnimi masami teles niso bistveno odstopale od predstavljenih simulacij s telesi z isto maso, vendar je tu še možnost nadaljnega raziskovanja.

Delujoč računalniški program predstavlja dobro osnovo za postopno približevanje realnejšim situacijam glede števila teles in začetnih pogojev, vendar pa je vzporedno s tem potrebno razmišljati o možnostih uporabe hitrejših računalnikov in natančnejših numeričnih metod integracije.

8 ZAHVALA

Za pomoč pri izdelavi raziskovalne naloge se zahvaljujem svojima mentorjema. Zahvaljujem se tudi dr. Tomažu Zwitteru za nekaj koristnih nasvetov glede začetnih pogojev v simulacijah. Zahvala gre tudi profesorici Polonci Glojek in Nataši Makovecki za lektoriranje. Za konec se zahvaljujem še svoji družini za podporo.

9 PRILOGA

K nalogi je priložen DVD z osmimi videoposnetki simulacij. Iz teh videoposnetkov formata mp4 so vzete slike v poglavjih od 4.2.1 do 4.2.4.

Sesedanje diska z različnimi začetnimi hitrostmi, predstavljenimi v poglavju 4.2.1, prikazujeta videoposnetka *disk enakomerna hitrost* in *disk linearne hitrosti*.

Sesedanje teles razporejenih po krogli iz poglavja 4.2.2 prikazujeta videoposnetka *krogla od zgoraj* in *krogla od strani*.

Srečanja diska s črno luknjo iz poglavja 4.2.3 prikazujejo videoposnetki *črna luknja navpično*, *črna luknja vodoravno* in *črna luknja ob disku*.

Trk diska s kroglo iz poglavja 4.2.4 prikazuje videoposnetek *galaktični trk*.

10 VIRI IN LITERATURA

1. Mohorič, A., Babič, V. 2014. Fizika 2. Mladinska knjiga Založba, d. d., Ljubljana.
2. <https://learnopengl.com/> (28. 10. 2017).
3. Wikipedia, Barnes - Hut simulation
https://en.wikipedia.org/wiki/Barnes%E2%80%93Hut_simulation (30. 11. 2017).
4. Ventimiglia, T., Wayne, K. The Barnes-Hut Algorithm. 2012.
<http://arborjs.org/docs/barnes-hut> (30. 11. 2017).
5. Naključna porazdelitev teles po notranjosti krogle
<http://mathworld.wolfram.com/SpherePointPicking.html> (4. 12. 2017).