

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 2989

**FIZIKALNO TEMELJENA SIMULACIJA
POVRŠINE VODE**

Niko Mikuličić

Zagreb, lipanj 2013.

Sadržaj

1.	Uvod	1
2.	Fizikalna osnova modela vodene površine	2
2.1.	Vizualna analiza vodene površine	2
2.2.	Navier-Stokesova jednadžba.....	3
3.	Modeli fluida	5
3.1.	Eulerov fluid	5
3.2.	Lagrangeov fluid.....	6
3.3.	Boltzmanova rešetka.....	6
3.4.	Odabir metode.....	7
4.	Hidrodinamika zaglađene čestice	8
4.1.	Zaglađujuće funkcije.....	9
4.2.	Matematički model kretanja fluida temeljen na sustavu čestica.....	12
4.2.1.	Gustoća	13
4.2.2.	Unutarnje sile.....	13
4.2.3.	Vanjske sile	16
4.3.	SPH pseudokod.....	19
5.	Implementacija	20
5.1.	Struktura programa	20
5.2.	Parametri simulacije	21
5.3.	Nalaženje susjednih čestica	22
5.4.	Leap-Frog integracija.....	23
5.5.	Obrada sudara	24
5.5.1.	Detekcija sudara	24
5.5.2.	Reakcija na sudar.....	26
5.6.	Iscrtavanje	26
6.	Rezultati.....	28
6.1.	Utjecaj pojedinih parametara	29
6.2.	Stabilnost	30
6.3.	Interakcija s korisnikom.....	31
7.	Budući razvoj	32
8.	Zaključak	33
9.	Literatura	34
10.	Sažetak.....	35
11.	Abstract	35

1. Uvod

Simulacija vodenih masa već je niz godina popularna tema u računalnoj grafici. Budući da vodu nalazimo svugdje u svakodnevnom životu, s napretkom računala postalo je neizbjegno simulirati njeno ponašanje i na računalima. Kako je vodena površina dinamična, odnosno u stalnom gibanju koje nije intuitivno predvidivo, realni prikaz vode ne može se ostvariti klasičnim tehnikama crtanja pojedinih slika i prikazivanja u obliku animacije, već su za to potrebni matematički modeli i velika snaga računanja.

Danas simulacije vodenih masa imaju raznoliku primjenu u filmskoj industriji, industriji reklama, video igrama, ali i u znanosti. Svako od navedenih područja primjene postavlja različite zahtjeve na karakteristike modela vode. Filmska industrija zahtijeva realno gibanje vode i visoku razinu detalja dok industrija reklama zahtijeva visoku kontrolu nad vodom. Video igre zahtijevaju prikaz i interakciju s vodenim površinama u stvarnom vremenu, dok znanost očekuje što precizniji prikaz stvarnog gibanja vode bez obzira na potrebno vrijeme izračunavanja.

Dok *offline* simulacije mogu proizvesti nevjerojatne vizualne efekte, postići istu razinu realnosti kod simulacija u stvarnom vremenu (*real-time*) još uvijek ostaje izazov. To je zbog toga što simulacija u stvarnom vremenu, za razliku od *offline* simulacije, mora zadovoljiti još nekoliko dodatnih zahtjeva: brzina izračunavanja mora biti u realnom vremenu (neki stvarnim vremenom smatraju 15fps, dok drugi zahtijevaju bar 60fps), simulacija mora biti apsolutno stabilna neodređeno dugo vremena, a ne samo par sekundi potrebnih za scenu u filmu, mora omogućavati interakciju s korisnikom, te raditi na prosječnom korisničkom računalu (dok se *offline* simulacije izračunavaju na moćnim računalima ili više njih). Iz navedenih razloga, simulacije u stvarnom vremenu nemaju jednaku mogućnost prikaza scena s velikim količinama vode i s jednakom razinom detalja kao i *offline* simulacije, ali uz određena ograničenja i aproksimacije mogu se postići poprilično realni rezultati.

Cilj ovog rada je fizikalno temeljena simulacija vodene površine u stvarnom vremenu. U sljedećem poglavlju osvrnut ćemo se na vizualni izgled vodene površine, ograničiti se na pojedine efekte koje je potrebno ostvariti te ćemo obraditi fizikalnu pozadinu dinamike fluida kao pretpostavku za izradu modela vodenih površina. U poglavlju 3 provedena je analiza postojećih modela vodenih površina te je odabrana

metoda za ostvarenje modela koji najviše odgovara cilju ovog rada. U 4. poglavlju je detaljno razrađen algoritam odabrane metode čija je implementacija potom predstavljena u poglavlju 5. U poglavlju 6 prikazani su rezultati izrađenog programskog produkta te su u poglavlju 7 dane smjernice za daljnji rad na ovom zadatku.

2. Fizikalna osnova modela vodene površine

Prije nego što se upustimo u tumačenje fizikalne pozadine dinamike fluida, prvo ćemo napraviti sveobuhvatnu vizualnu analizu vodene površine. Iako mnogo toga navedenog nije sastavni dio ovog rada, korisno je napraviti osvrt zbog mogućih budućih unapređenja razvijenog programskog produkta.

2.1. Vizualna analiza vodene površine

Vizualna analiza vodene površine daje nam informaciju o tome koje efekte na vodenoj površini možemo primijetiti, a samim time i koje efekte vodene površine ima smisla implementirati da bi voda izgledala realno. Od najuočljivijih efekata možemo izdvojiti: prskanje, mjeđuriće, pjenu i deformacije površine.

Budući da su prskanja, mjeđurići i pjena kaotični po svojoj prirodi, njih se vremenski efikasno može implementirati stohastičkim metodama. Za *real-time* implementaciju najveći problem predstavljaju deformacije vodene površine budući da su za njihov realan prikaz potrebne determinističke i računalno skupe metode (Yuksel, 2010).

Ovisno o tome koliku količinu vode simuliramo i u kojoj skali, pojedini efekti vodene površine imat će veći ili manji značaj. Budući da su ostale deformacije površine jako nestabilne i vrlo se brzo pretvaraju u valove, ukoliko simuliramo velike količine vode najvažnije će nam biti pravilno simulirati površinske valove. S druge strane, ukoliko simuliramo malu količinu vode i pritom je uvećamo, najvažniji će biti površinski efekti poput mjeđurića i sl.

Općenito postoje dvije vrste valova (Yuksel, 2010): ambijentalni i interaktivni valovi. Ambijentalni valovi uglavnom nastaju djelovanjem vjetra i možemo ih smatrati ponašanjem sustava u mirovanju. S druge strane, interaktivni valovi nastaju djelovanjem lokalizirane sile na vodenu površinu. Dok ambijentalne valove možemo izračunati prije pokretanja same simulacije, interaktivne valove, budući da nastaju djelovanjem korisnika, moramo simulirati u stvarnom vremenu.

Zbog jednostavnosti, u ovom radu ćemo se ograničiti na interaktivne valove dok ćemo ostale efekte vodene površine ostaviti mogućem dalnjem razvoju na ovom zadatku. U nastavku razmatramo Navier-Stokesovu jednadžbu kao temelj dinamike fluida.

2.2. Navier-Stokesova jednadžba

Kretanje nestlačivih fluida u vremenu opisujemo sljedećim jednadžbama:

$$\nabla \cdot \mathbf{u} = 0 \quad (2.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} - \frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \quad (2.2)$$

gdje simbol \mathbf{u} označava brzinu fluida, ρ gustoću, p tlak, a ν viskoznost fluida. Simbol \mathbf{f} odnosi se na vanjske sile koje djeluju na fluid kao npr. gravitacijska sila. Različita značenja operatora ∇ koji se koristi u jednadžbama prikazana su u tablici 2.1.

Tablica 2.1 Značenja operatora ∇ korištenog u Navier-Stokesovoj jednadžbi i jednadžbi kontinuiteta

Operator	Definicija
Gradijent	$\nabla p = \left(\frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}, \frac{\partial p}{\partial z} \right)$
Divergencija	$\nabla \cdot \mathbf{u} = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z}$
Usmjerena derivacija	$\mathbf{u} \cdot \nabla = u_x \frac{\partial}{\partial x} + u_y \frac{\partial}{\partial y} + u_z \frac{\partial}{\partial z}$
Laplasijan	$\nabla^2 \mathbf{u} = \frac{\partial^2 \mathbf{u}}{\partial x^2} + \frac{\partial^2 \mathbf{u}}{\partial y^2} + \frac{\partial^2 \mathbf{u}}{\partial z^2}$

Jednadžbu (2.1) nazivamo jednadžbom kontinuiteta i ona govori da za svaki volumen fluida, količina fluida koja ulazi u taj volumen je jednaka količini fluida koja izlazi iz njega, odnosno da je masa fluida očuvana tijekom gibanja. Kad je ta jednadžba zadovoljena, kažemo da je fluid nestlačiv.

Jednadžba (2.2) zove se Navier-Stokesova jednadžba. Ona predstavlja zakon o očuvanju količine gibanja fluida te se direktno izvodi iz drugog Newtonovog zakona.

Sastoje se od nekoliko komponenti koje opisuju promjene u brzini fluida na određenoj poziciji: advekcija, tlak, viskoznost i vanjske sile (Chrisman, 2008).

Advekcija: $-(\mathbf{u} \cdot \nabla) \mathbf{u}$. Ova komponenta čini Navier-Stokesovu jednadžbu nelinearnom diferencijalnom jednadžbom. Advekcija je prijenos neke veličine kroz vektorsko polje. Ta veličina može biti vektor poput pozicije čestice ili njene brzine, može biti skalar poput temperature, a može biti i samo vektorsko polje.

Tlak: $-\frac{1}{\rho} \nabla p$, odnosno gradijent tlaka utječe na brzinu u smislu da usmjerava fluid s područja višeg na područje nižeg tlaka.

Viskoznost: $\nu \nabla^2 \mathbf{u}$, je komponenta zadužena za simuliranje viskoznosti fluida. Što je veća viskoznost, to čestice svojom brzinom više utječu na brzine susjednih čestica, odnosno brzine susjednih čestica više teže prema istoj vrijednosti. To ima za posljedicu da se kod vrlo viskoznih fluida brzine brzo rasprše među susjednim česticama što dovodi do puno bržeg umirivanja fluida nego kod nisko viskoznih fluida.

Vanjske sile: \mathbf{f} , je posljednja komponenta Navier-Stokesove jednadžbe i sadrži vanjske sile koje utječu na fluid. Vanjske sile mogu biti lokalne ili globalne. Lokalne se primjenjuju nad pojedinim područjem fluida, dok globalne djeluju nad cijelim fluidom kao npr. gravitacija.

Zanimljivo je napomenuti da iako je Navier-Stokesova jednadžba postavljena prije više od sto pedeset godina, njeni općenito rješenje u zatvorenoj formi još uvijek nije pronađeno. S razvojem računalne tehnologije razvijene su brojne metode za aproksimaciju Navier-Stokesove jednadžbe, te je uspostavljena nova znanstvena disciplina pod nazivom računalna dinamika fluida (*Computational Fluid Dynamics*). Mnogi algoritmi u fizikalno temeljenoj simulaciji fluida preuzeti su upravo iz te znanstvene discipline. Kao interesantan primjer posljedice aproksimacije Navier-Stokesove jednadžbe možemo uzeti vremensku prognozu. Iako simulira tok fluida kroz vrijeme, zbog nužnosti aproksimacije ona nikad nije potpuno točna.

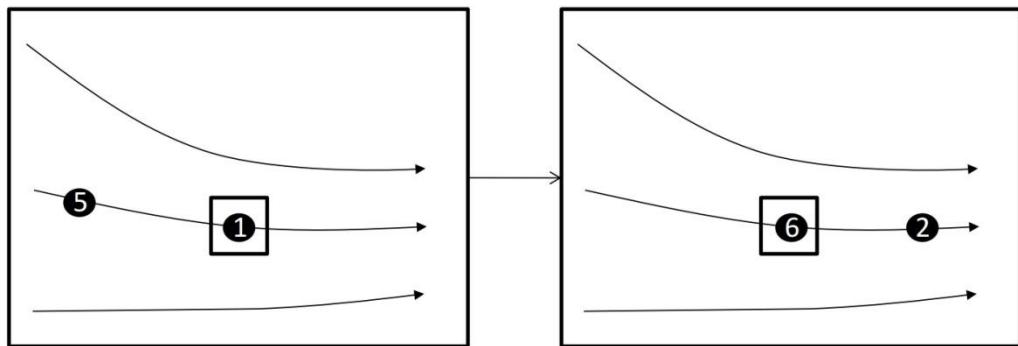
Navier-Stokesova jednadžba zajedno s jednadžbom kontinuiteta preduvjet su za izradu modela vodenih površina. U sljedećem poglavljju razmotrit ćemo postojeće fizikalno temeljene modele nestlačivih fluida te ćemo navesti pojedine metode koje se koriste za njihovo računalno ostvarivanje.

3. Modeli fluida

Postoje tri različita pogleda na fluid i s posljedično tri različita modela fluida (Tan, 2009): Eulerov fluid, Lagrangeov fluid, Boltzmanova rešetka. Iako su navedeni pogledi naizgled različiti, vladajuće jednadžbe su još uvijek Navier-Stokesove jednadžbe ili se mogu svesti na njih, te se može pokazati da su modeli zapravo matematički ekvivalentni.

3.1. Eulerov fluid

Eulerov pogled gleda na fluid kao na poravnatu rešetku čelija po kojima je fluid diskretiziran. Promjene u fluidu poput tlaka, brzine, gustoće, itd. bilježe se kao svojstva fiksiranih čelija. Na slici 3.1 možemo vidjeti da protjecanjem vremena čelija ostaje na istom mjestu. Ono što se mijenja su svojstva i količina fluida u njoj.



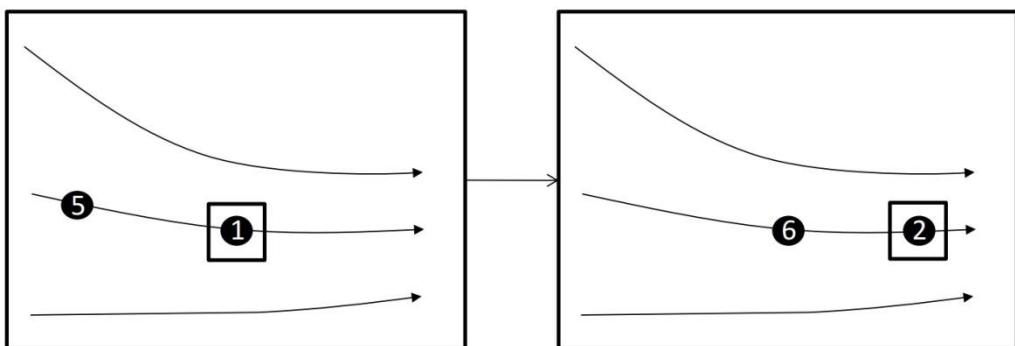
Slika 3.1 Eulerov pogled na fluid

Negativna posljedica ovog modela je što je fluid ograničen na rešetku. On ne može postojati izvan nje i zbog toga je teško simulirati realističan tok fluida u situacijama kad se npr. spremnik dotičnog fluida slomi. Kao odgovor na to javile su se adaptivne rešetke ali to ne rješava ostale probleme. Veličina rešetke je usko povezana s količinom zauzete radne memorije računala gdje je ovisnost kvadratna ili čak kubna što ovu metodu čini loše skalabilnom. Također, ovaj model je i računski, odnosno vremenski vrlo zahtjevan.

Iako zbog navedenih razloga nije prigodan za interaktivne simulacije, ovaj model se široko koristi u *offline* simulacijama. Jednostavni razlozi za to što pruža veliku razinu detalja što za posljedicu ima nevjerojatnu realističnost za koju je katkad teško reći je li simulacija ili stvarnost.

3.2. Lagrangeov fluid

Za razliku od Eulerovog, Lagrangeov pogled gleda na fluid kao na niz čestica koje u potpunosti definiraju fluid, što znači da se čestice gibaju s fluidom. Svaka čestica ima informaciju o svojoj masi, brzini i poziciji te se promjene u fluidu (tlak, gustoća, itd.) izračunavaju za svaku česticu posebno. Na slici 3.2 možemo vidjeti da fokus više nije na fiksiranim pozicijama u fluidu kao kod Eulerovog pogleda, već je fokus na česticama koje se gibaju zajedno s fluidom.



Slika 3.2 Lagrangeov pogled na fluid

Pozitivna strana ovakvog pogleda na fluid je to što su čestice slobodne gibati se kroz prostor, odnosno fluid nije prostorno ograničen kao što je slučaj s Eulerovim fluidom. Također, iz Lagrangeovog fluida lako je dobiti detalje površine fluida kao što su prskanja, pjena i mjehurići. Ovaj model općenito zahtijeva manje računalnih resursa od ostalih pa je zato posebno popularan kod interaktivnih simulacija.

S negativne strane, budući da je fluid definiran česticama postoje poteškoće u rekonstrukciji glatke površine što za posljedicu ima da ovaj model ne može simulirati fluid s jednakom razine detalja kao i ostali modeli. Nadalje, ne može se garantirati nestlačivost. Iako postoje metode koje rješavaju problem nestlačivosti, one zahtijevaju dulje vrijeme izračunavanja.

3.3. Boltzmanova rešetka

Za razliku od prethodno navedenih modela koji se zasnivaju direktno na Navier-Stokesovim jednadžbama, ovaj model gleda na fluid s mikroskopske razine. Glavna ideja je napraviti pojednostavljene kinetičke modele koji objedinjuju mikroskopske i mezoskopske fizikalne procese na taj način da makroskopska svojstva koja se pritom očituju zadovoljavaju željene makroskopske jednadžbe. Jednostavnije rečeno, ovaj model

modelira mikroskopske i mezoskopske fizikalne procese koji na makroskopskoj razini konvergiraju prema Navier-Stokesovoj jednadžbi.

Kao i kod Eulerovog fluida, ovaj model podrazumijeva da je područje fluida podijeljeno na ćelije i to sa sobom povlači lošu skalabilnost. Osim toga, vremenski korak mora biti mali kako bi simulacija ostala stabilna, no valja napomenuti da se vladajuće jednadžbe uglavnom rješavaju brže nego što vremenski korak to ograničava, pa to uglavnom ne predstavlja problem.

S druge strane, neki fenomeni fluida koji se teško modeliraju klasičnim Eulerovim ili Lagrangeovim fluidom prirodno su ukomponirani u jednadžbe Boltzmanove rešetke. Nadalje, algoritmi koji implementiraju ovaj model lako se paraleliziraju i implementiraju na grafičkoj kartici.

3.4. Odabir metode

Za svaki od navedenih modela razvijeno je niz metoda koje implementiraju takav pogled na fluid. U početku razmatranja različitih metoda za ostvarenje interaktivne simulacije vodene površine najviše mi se svidjela metoda Boltzmanove rešetke jer ima relativno malo nedostataka, a i smatrao sam da je s razlogom postala popularna u posljednje vrijeme. Ipak, povučen radom *Position Based Fluids* (Macklin, 2013), odlučio sam se za tu metodu. Nakon što sam je gotovo programski ostvario, shvatio sam da podešavanje parametara da bi simulacija funkcionalala nije nimalo lagan posao, a budući da je predložena metoda nova i tek se ima objaviti, nije dovoljno detaljno razrađena i dokumentirana pa nisam imao gdje potražiti odgovore na svoja pitanja.

Na posljetku sam se odlučio za metodu hidrodinamike zaglađene čestice (*Smoothed Particle Hydrodynamics*) (Kelager, 2006) koja je već dugo vremena popularna i stoga dobro dokumentirana. Osim toga, budući da ta metoda implementira Lagrangeov fluid baš kao i metoda implementirana u radu *Position Based Fluids* (Macklin, 2013), nije bilo teško prilagoditi već ostvareni kod novoj metodi. Također, mogao sam biti siguran da će dobiti interaktivnu simulaciju budući da je to glavna prednost Lagrangeovog fluida.

4. Hidrodinamika zaglađene čestice

Hidrodinamika zaglađene čestice (*Smoothed Particle Hydrodynamics*, SPH) je interpolacijska metoda inicijalno razvijena za simulaciju dinamike astrofizičkih plinova kao npr. strukturiranje svemira, formiranje galaksija, supernova i solarnih sustava. Vrijednost veličine $A(\mathbf{r})$ na poziciji \mathbf{r} dobije se interpolacijom vrijednosti veličina okolnih točaka u prostoru, odnosno možemo pisati:

$$A_I(\mathbf{r}) = \int A(\mathbf{r}') W(\mathbf{r} - \mathbf{r}', h) d\mathbf{r}' \quad (4.1)$$

gdje je $W(\mathbf{r}, h)$ radijalno simetrična funkcija zaglađivanja (*smoothing kernel*) s duljinom zaglađivanja h (*smoothing length*). Može se reći da interpolacija koristi funkciju zaglađivanja da proširi vrijednost veličine s dane pozicije na njenu okolinu. U praksi je poželjno da je funkcija zaglađivanja parna ($W(\mathbf{r}, h) = W(-\mathbf{r}, h)$) i normalizirana ($\int W(\mathbf{r}) d\mathbf{r} = 1$). Također je poželjno da teži prema delta funkciji kad h teži u nulu što možemo interpretirati kao: kad duljina zaglađivanja teži u nulu, okolne točke ne utječu na vrijednost promatrane veličine i izraz $A_I(\mathbf{r})$ postaje jednak $A(\mathbf{r})$ što znači da interpolirana vrijednost veličine u nekoj točki postaje jednaka vrijednosti promatrane veličine u toj točki. Zato ćemo h promatrati kao radijus unutar kojeg ostale točke utječu na interpoliranu vrijednost. Ukoliko je razlika pozicija dviju točaka veća od h , odnosno $|\mathbf{r} - \mathbf{r}'| > h$, funkcija zaglađivanja će međusobni utjecaj evaluirati u nula.

Budući da metoda SPH ostvaruje Lagrangeov model fluida, fluid je reprezentiran nizom čestica gdje svaka čestica j ima informaciju o svojoj poziciji \mathbf{r}_j , brzini \mathbf{v}_j , masi m_j i gustoći ρ_j . Vrijednost neke veličine na danoj poziciji dobije se interpolacijom vrijednosti A_j susjednih čestica koristeći interpolacijsku sumu izvedenu iz integralne forme (4.1):

$$A_s(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h) \quad (4.2)$$

gdje se koeficijent mase po gustoći pojavljuje jer svaka čestica predstavlja volumen $V_j = \frac{m_j}{\rho_j}$. Budući da derivacija sume (4.2) utječe samo na zaglađujuću funkciju, njen gradijent možemo pisati:

$$\nabla A_s(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} \nabla W(\mathbf{r} - \mathbf{r}_j, h) \quad (4.3)$$

i Laplasijan:

$$\nabla^2 A_s(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} \nabla^2 W(\mathbf{r} - \mathbf{r}_j, h) \quad (4.4)$$

Kod SPH metode, stabilnost, brzina i vjernost simulacije jako ovisi o izgledu zaglađujuće funkcije $W(\mathbf{r}, h)$ i stoga ona mora biti pažljivo dizajnirana.

4.1. Zaglađujuće funkcije

U ovom radu su se za izračun različitih veličina koristile tri zaglađujuće funkcije za koje se utvrdilo da dobro obavljaju svoju ulogu, a to su: *Poly6*, *Spiky* i *Viscosity*. Prva je *Poly6* zaglađujuća funkcija prikazana na slici 4.1 i jednaka je:

$$W_{poly6}(\mathbf{r}, h) = \begin{cases} \frac{315}{64\pi h^9} (h^2 - |\mathbf{r}|^2)^3, & 0 \leq |\mathbf{r}| \leq h \\ 0, & \text{inache} \end{cases} \quad (4.5)$$

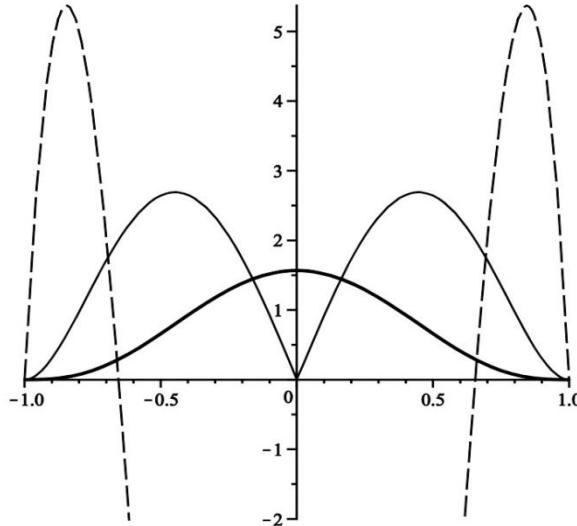
s gradijentom:

$$\nabla W_{poly6}(\mathbf{r}, h) = -\mathbf{r} \frac{945}{32\pi h^9} (h^2 - |\mathbf{r}|^2)^2 \quad (4.6)$$

i Laplasijanom:

$$\nabla^2 W_{poly6}(\mathbf{r}, h) = -\frac{945}{32\pi h^9} (h^2 - |\mathbf{r}|^2)(3h^2 - 7|\mathbf{r}|^2). \quad (4.7)$$

Njene glavne prednosti su što zadržava izgled Gaussove zvonolike krivulje te uvijek koristi izraz $|\mathbf{r}|^2$ pa se može izbjegći skupo računanje drugog korijena. Ova funkcija je korištena za sve osim za proračun sila koje nastaju posljedicom tlaka i viskoznosti.



Slika 4.1 Poly6 zaglađujuća funkcija za $x = \mathbf{0}, y = \mathbf{0}, h = 1$. Podebljana linija je funkcija $W(\mathbf{r}, h)$, tanka linija je apsolutna vrijednost gradijenta $|\nabla W(\mathbf{r}, h)|$, a isprekidana Laplasijan $\nabla^2 W(\mathbf{r}, h)$.

Problem s tlačnim silama je što gradijent Poly6 funkcije teži prema nuli kad su čestice jako blizu, odnosno odbojne sile nestaju onda kad bi trebale biti najjače. Zbog navedenog razloga dizajnirana je *Spiky* zaglađujuća funkcija prikazana na slici 4.2 koja iznosi:

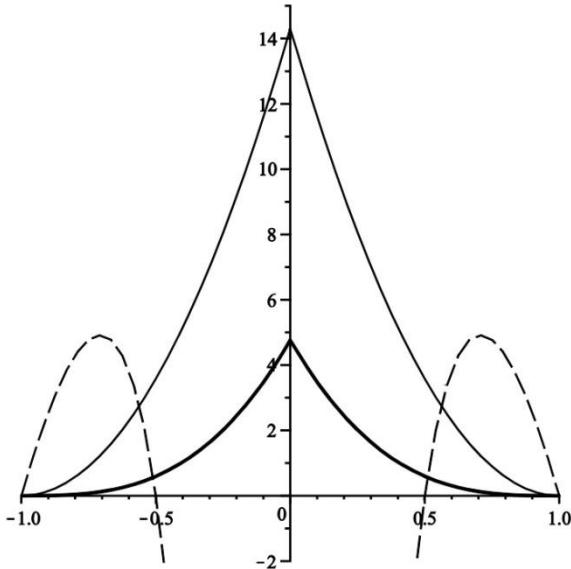
$$W_{spiky}(\mathbf{r}, h) = \frac{15}{\pi h^6} \begin{cases} (h - |\mathbf{r}|)^3, & 0 \leq |\mathbf{r}| \leq h \\ 0, & \text{inače} \end{cases} \quad (4.8)$$

s gradijentom:

$$\nabla W_{spiky}(\mathbf{r}, h) = -\frac{45}{\pi h^6} \frac{\mathbf{r}}{|\mathbf{r}|} (h - |\mathbf{r}|)^2 \quad (4.9)$$

i Laplasijanom:

$$\nabla^2 W_{spiky}(\mathbf{r}, h) = -\frac{90}{\pi h^6} \frac{1}{|\mathbf{r}|} (h - |\mathbf{r}|)(h - 2|\mathbf{r}|). \quad (4.10)$$



Slika 4.2 Spiky zaglađujuća funkcija za $\mathbf{x} = \mathbf{0}$, $y = 0$, $\mathbf{h} = \mathbf{1}$. Podebljana linija je funkcija $W(\mathbf{r}, \mathbf{h})$, tanka linija je absolutna vrijednost gradijenta $|\nabla W(\mathbf{r}, \mathbf{h})|$, a isprekidana Laplasijan $\nabla^2 W(\mathbf{r}, \mathbf{h})$.

Ova funkcija je korištena za izračun tlačne sile između čestica, i za razliku od *Poly6* funkcije ima svojstvo da absolutna vrijednost gradijenta raste kako se čestice približavaju, odnosno odbojne sile su jače što su čestice bliže.

S viskoznim silama je problem što kako se čestice približavaju Laplasijan *Poly6* funkcije vrlo brzo postaje negativan. To za posljedicu ima da čestica koja je brža od svoje okoline može biti dodatno ubrzana rezultantnom viskoznom silom, dok bi zapravo trebala biti usporena. Zbog toga je dizajnirana *Viscosity* zaglađujuća funkcija koja je prikazana na slici 4.3 i iznosi:

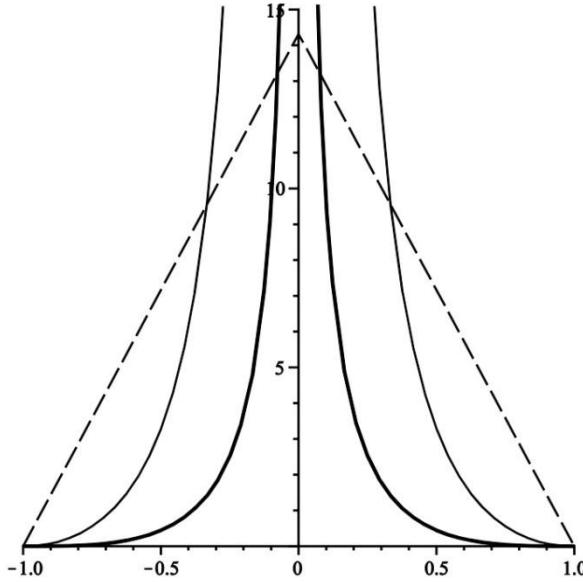
$$W_{viscosity}(\mathbf{r}, h) = \frac{15}{2\pi h^3} \begin{cases} -\frac{|\mathbf{r}|^3}{2h^3} + \frac{|\mathbf{r}|^2}{h^2} + \frac{h}{2|\mathbf{r}|} - 1, & 0 \leq |\mathbf{r}| \leq h \\ 0, & \text{inache} \end{cases} \quad (4.11)$$

s gradijentom:

$$\nabla W_{viscosity}(\mathbf{r}, h) = \frac{15}{2\pi h^3} \mathbf{r} \left(-\frac{3|\mathbf{r}|}{2h^3} + \frac{2}{h^2} - \frac{h}{2|\mathbf{r}|^3} \right) \quad (4.12)$$

i Laplasijanom:

$$\nabla^2 W_{viscosity}(\mathbf{r}, h) = \frac{45}{\pi h^6} (h - |\mathbf{r}|). \quad (4.13)$$



Slika 4.3 Viscosity zaglađujuća funkcija za $x = \mathbf{0}, y = \mathbf{0}, h = \mathbf{1}$. Podebljana linija je funkcija $W(\mathbf{r}, h)$, tanka linija je apsolutna vrijednost gradijenta $|\nabla W(\mathbf{r}, h)|$, a isprekidana Laplasijan $\nabla^2 W(\mathbf{r}, h)$.

Laplasijan ove funkcije je uvijek pozitivan pa se problem viskoznih sila koji se javlja s Poly6 zaglađujućom funkcijom ovdje nećejavljati.

4.2. Matematički model kretanja fluida temeljen na sustavu čestica

Kako bi se čestice ponašale kao fluid, one moraju približno zadovoljavati jednadžbe (2.1) i (2.2). Budući da riječ o Lagrangeovom fluidu gdje svaka čestica ima konstantnu masu i gdje je broj čestica konstantan, očuvanje mase je automatski garantirano pa jednadžbu (2.1) možemo izostaviti. Nadalje, izraz advekcije u jednadžbi (2.2) označava promjenu količine fluida koju promatramo u nekom trenutku. Ukoliko promatramo fluid kao Eulerov s fokusom na jednu ćeliju, osim što se s vremenom mijenjaju svojstva fluida koji se nalazi u toj ćeliji, mijenja se i sama količina fluida. Kod Lagrangeovog fluida to nije slučaj budući da promatramo česticu koja reprezentira uvijek istu količinu fluida. Posljedica toga je da ukoliko koristimo Lagrangeov pogled na fluid, izraz advekcije možemo izbaciti iz jednadžbe za očuvanje količine gibanja (Auer, 2008) nakon čega ostaje jednadžba:

$$\frac{d\mathbf{u}}{dt} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \quad (4.14)$$

Parcijalna derivacija zamijenjena je običnom jer sada svaka veličina ovisi samo o vremenu t . Jednadžbu možemo pomnožiti s gustoćom i zapisati je u kontekstu jedne čestice:

$$\rho(\mathbf{r}_i) \frac{d\mathbf{u}_i}{dt} = \mathbf{F}_i = -\nabla p(\mathbf{r}_i) + \mu \nabla^2 \mathbf{u}(\mathbf{r}_i) + \rho(\mathbf{r}_i) \mathbf{f}(\mathbf{r}_i) \quad (4.15)$$

gdje je parametar $\mu = \nu \rho$ dinamička viskoznost i uzimamo je kao konstantu. Desna strana jednadžbe predstavlja unutarnje i vanjske sile koje djeluju na česticu te možemo pisati $\mathbf{F}_i = \mathbf{f}_i^{unutarnje} + \mathbf{f}_i^{vanjske}$, gdje je \mathbf{F}_i ukupna sila koja djeluje na česticu i . Za akceleraciju čestice sada vrijedi:

$$\mathbf{a}_i = \frac{d\mathbf{u}_i}{dt} = \frac{\mathbf{F}_i}{\rho(\mathbf{r}_i)} \quad (4.16)$$

Dakle, kako bismo mogli izračunati poziciju na kojoj će se čestica nalaziti u sljedećem trenutku i brzinu koju će pritom imati, što je u konačnici cilj, nužno je izračunati akceleraciju čestice. Da bismo mogli izračunati akceleraciju potrebno je znati gustoću fluida na poziciji promatrane čestice te unutarnje i vanjske sile koje djeluju na nju.

4.2.1. Gustoća

Uvrštavajući izraz za gustoću u općenitu SPH interpolacijsku sumu danu jednadžbom (4.2) lako dobijemo izraz za gustoću na poziciji promatrane čestice:

$$\rho_i = \rho(\mathbf{r}_i) = \sum_j m_j W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (4.17)$$

gdje se sumira po svim susjednim česticama, odnosno po česticama unutar radiusa h .

4.2.2. Unutarnje sile

Unutarnje sile su sile koje nastaju unutar samog fluida, a to su tlačne i viskozne sile. U jednadžbi (4.15), prvi izraz na desnoj strani jednadžbe odgovara tlačnim silama, a drugi viskoznim.

4.2.2.1 Tlak

Ukoliko se izraz za tlačne sile uvrsti u općenitu SPH interpolacijsku sumu danu jednadžbom (4.2) dobijemo jednadžbu:

$$\mathbf{f}_i^{tlak} = -\nabla p(\mathbf{r}_i) = -\sum_{j \neq i} p_j \frac{m_j}{\rho_j} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (4.18)$$

gdje je p_j tlak na poziciji \mathbf{r}_j čestice j . Možemo primijetiti da je tlačna sila dobivena u (4.18) asimetrična. Ukoliko promatramo samo dvije čestice i tlačne sile između njih, možemo vidjeti da tlačna sila na poziciji pojedine čestice ovisi samo o vrijednostima na poziciji druge čestice. Budući da se vrijednosti veličina mijenjaju s obzirom na poziciju, dvije čestice koje su u međusobnoj interakciji imat će različite tlačne sile kojom djeluju jedna na drugu. Budući da je poželjno da je SPH izraz za izračunavanje pojedine veličine simetričan, koristit ćemo jednadžbu:

$$\mathbf{f}_i^{tlak} = -\rho_i \sum_{j \neq i} \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) m_j \nabla W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (4.19)$$

Kao što je već napomenuto u poglavlju 4.1, prilikom izračunavanja tlačnih sila za funkciju zaglađivanja koristit ćemo *Spiky* zaglađujuću funkciju, odnosno njen gradijent.

Izraz p_j za tlak na poziciji čestice j je još uvijek nepoznat. Standardno se za njegovo izračunavanje koristi jednadžba idealnog plina:

$$pV = nRT \quad (4.20)$$

gdje je $V = \frac{1}{\rho}$ volumen po jedinici mase, n količina čestica u plinu izražen u molovima, R univerzalna konstanta plina, te T njegova temperatura. Ukoliko prepostavimo da promatramo fluid u kojem se broj čestica i temperatura ne mijenjaju, desnu stranu jednadžbe (4.20) možemo zamijeniti konstantom i pisati:

$$pV = k \quad (4.21)$$

$$p \frac{1}{\rho} = k$$

$$p = k\rho \quad (4.22)$$

Takov model tlaka rezultirat će samo odbojnim silama među česticama, što je ispravno u slučaju plina koji se širi kroz prostor. Ukoliko želimo modelirati tekućine, potrebno je uvesti i privlačne sile među česticama, odnosno koheziju. U te svrhe dodaje se tlak mirovanja p_0 za kojeg po jednadžbi (4.22) vrijedi:

$$p_0 = k\rho_0 \quad (4.23)$$

Za ukupni tlak onda, po jednadžbi (4.21), možemo pisati:

$$(p + p_0)V = k$$

Uvrštavanjem (4.23) dobije se:

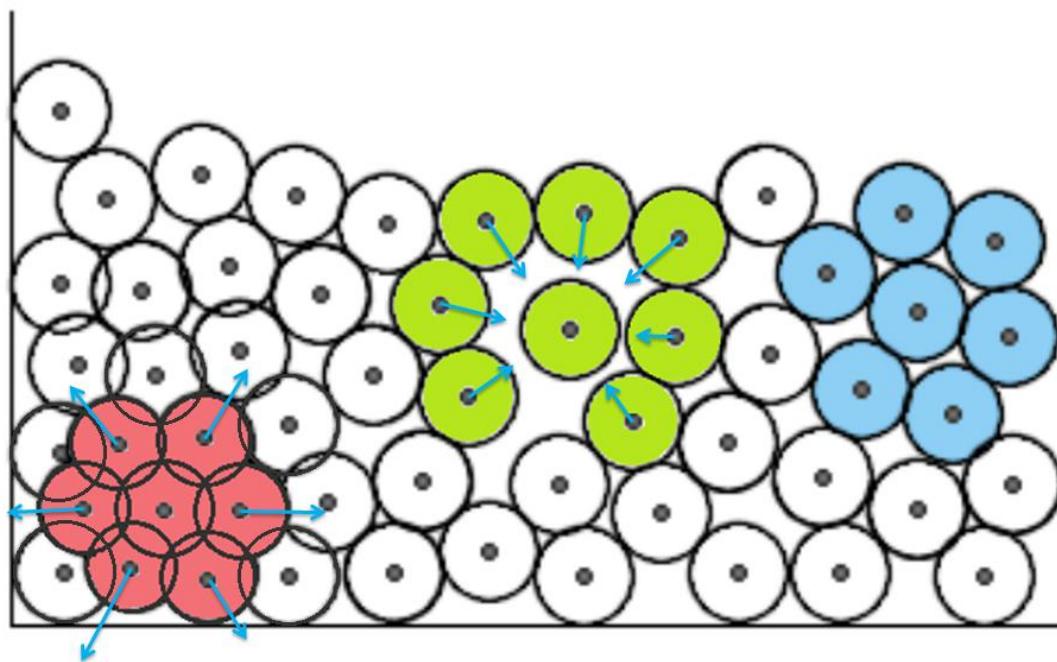
$$(p + k\rho_0)V = k$$

te konačno uz dodavanje pojedinih oznaka:

$$p_j = k(\rho_j - \rho_0) \quad (4.24)$$

gdje su p_j i ρ_j tlak i gustoća na poziciji čestice j , a ρ_0 gustoća mirovanja promatranog fluida.

Jednadžba (4.19) uz jednadžbu (4.24) modelira i privlačne i odbojne sile među česticama. Na slici 4.4 prikazane su čestice fluida s različitim tlačnim silama. U zelenom području čestice imaju gustoću manju od gustoće mirovanja fluida što uzrokuje negativan tlak odnosno privlačne sile među njima. Crveno područje prikazuje dio fluida gdje su čestice preblizu jedna drugoj, odnosno imaju gustoću veću od gustoće mirovanja fluida što rezultira pozitivnim tlakom i odbojnim silama među njima. Plavo područje označava čestice koje su raspoređene tako da im je gustoća jednaka gustoći mirovanja fluida. To je gustoća kojoj teže sve čestice i u tim uvjetima tlačne sile su jednake nula.



Slika 4.4 Prikaz tlačnih sila u fluidu

4.2.2.2 Viskoznost

Ukoliko se izraz za viskozne sile uvrsti u SPH interpolacijsku sumu danu jednadžbom (4.2) dobije se:

$$\mathbf{f}_i^{viskoznost} = -\mu \nabla^2 \mathbf{u}(\mathbf{r}_i) = \mu \sum_{j \neq i} \mathbf{u}_j \frac{m_j}{\rho_j} \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (4.25)$$

gdje je μ dinamička viskoznost. Slično kao i s tlačnim silama, izraz dobiven za viskozne sile je asimetričan jer brzina varira od čestice do čestice. Umjesto jednadžbe (4.25) koristit ćemo jednadžbu:

$$\mathbf{f}_i^{viskoznost} = \mu \sum_{j \neq i} (\mathbf{u}_j - \mathbf{u}_i) \frac{m_j}{\rho_j} \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (4.26)$$

što je moguće budući da viskozne sile zapravo ovise o razlici brzina čestica (Kelager, 2006). U skladu s poglavljem 4.1, za izračunavanje viskoznih sila koristit ćemo *Viscosity* zaglađujuću funkciju, odnosno njen Laplasijan.

4.2.3. Vanjske sile

Vanjske sile nastaju izvan fluida i mogu biti globalne ili lokalne. Lokalne vanjske sile djeluju samo na dio fluida, dok globalne djeluju na cijeli fluid. U vanjske sile spadaju gravitacija, površinska napetost te sile koje nastaju interakcijom korisnika s fluidom. Neke vanjske sile poput gravitacije i interakcijskih sila mogu se primjenjivati direktno na čestice bez korištenja SPH metode, a neke, poput površinske napetosti, ovise o susjednim česticama i stoga se za njihovo izračunavanje treba koristiti SPH.

4.2.3.1 Gravitacija

Gravitacijska sila djeluje jednakom na sve čestice i za njeno izračunavanje koristimo jednadžbu:

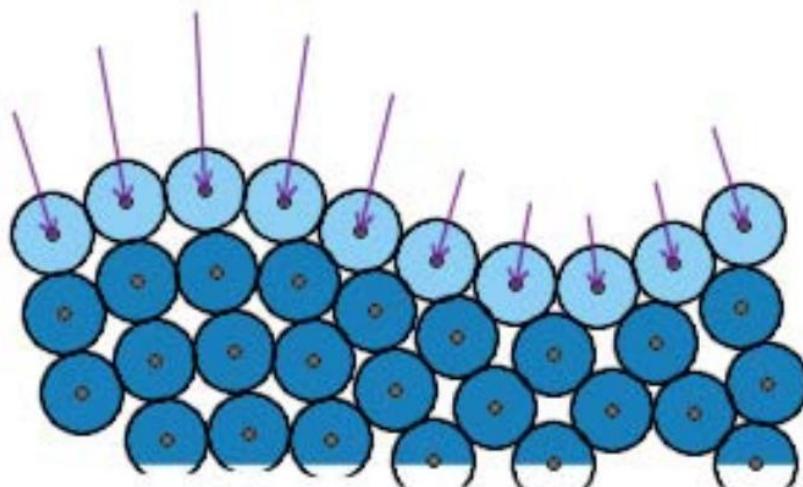
$$\mathbf{f}_i^{gravitacija} = \rho_i \mathbf{g} \quad (4.27)$$

gdje je \mathbf{g} gravitacijska akceleracija.

4.2.3.2 Površinska napetost

Površinska napetost je sila koja nastaje kao posljedica kohezivnih sila između čestica na površini fluida. Ona nije uključena u Navier-Stokesove jednadžbe već se smatra graničnim uvjetom. Molekule u unutrašnjosti fluida su sa svih strana okružene istovrsnim

molekulama i stoga su privlačne sile među njima izbalansirane. S druge strane, molekule na površini fluida nemaju istovrsne molekule sa svih strana i stoga se čvršće vežu za one molekule fluida koje su u njihovoј blizini. Takvo ponašanje fluida nazivamo površinska napetost. Sile površinske napetosti djeluju u smjeru normale na površinu prema unutrašnjosti fluida i nastoje minimizirati površinu koju fluid zauzima. Slika 4.5 prikazuje način na koji sile površinske napetosti djeluju na fluid. Možemo primjetiti da su sile jače na lijevoj strani slike gdje je površina fluida konveksna, odnosno slabije na desnoj strani slike gdje je površina fluida konkavna. To je zato što za konkavnu površinu fluida možemo reći da se nalazi "više u fluidu" i stoga je pojedina čestica više okružena susjednim česticama nego što je slučaj s česticama na konveksnom dijelu površine. Posljedično se i privlačne sile među česticama na konkavnom dijelu bolje balansiraju pa su stoga i sile površinske napetosti na tom dijelu manje.



Slika 4.5 Sile površinske napetosti djeluju u smjeru normale na površinu prema unutrašnjosti fluida

Da bismo izračunali silu površinske napetosti koja djeluje na pojedinu česticu fluida, prvo je potrebno izračunati normalu na površinu fluida na poziciji promatrane čestice. U tu svrhu uvodimo još jednu veličinu - obojanost c . Obojanost je veličina za koju vrijedi $c = 1$ točno na poziciji promatrane čestice te $c = 0$ inače. Uvrštavanje te veličine u SPH interpolacijsku sumu danu jednadžbom (4.2) dobije se:

$$c_i = c(\mathbf{r}_i) = \sum_j c_j \frac{m_j}{\rho_j} W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (4.28)$$

odnosno, budući da vrijedi $c_j = 1$:

$$c_i = \sum_j \frac{m_j}{\rho_j} W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (4.29)$$

Gradijent obojanosti ∇c daje nam dvije vrste informacija: njegova duljina postaje velika samo blizu površine fluida što nam pomaže utvrditi koje čestice se nalaze na površini, te ima smjer okomit na površinu usmjeren prema unutrašnjosti fluida što ga činim pogodnim izborom za normalu na površinu fluida. Stoga pišemo:

$$\mathbf{n}_i = \nabla c(\mathbf{r}_i) = \sum_j \frac{m_j}{\rho_j} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (4.30)$$

gdje je \mathbf{n}_i oznaka za traženu normalu na površinu na poziciji čestice i .

Zakrivljenost površine fluida zapravo je mjera za jačinu sile površinske napetosti i može se izraziti uz pomoć Laplasijana obojanosti:

$$\kappa_i = -\frac{\nabla^2 c_i}{|\nabla c_i|} = -\frac{\nabla^2 c_i}{|\mathbf{n}_i|} \quad (4.31)$$

gdje je predznak negativan kako bi zakrivljenost bila pozitivna za područja gdje je površina konveksna.

Koristeći vektor normale na površinu za smjer u kojem djeluje sila površinske napetosti, te zakrivljenost površine fluida kao jačinu te sile, za silu površinske napetosti konačno možemo pisati:

$$\mathbf{f}_i^{površinska\ napetost} = \sigma \kappa_i \mathbf{n}_i = -\sigma \frac{\mathbf{n}_i}{|\mathbf{n}_i|} \nabla^2 c_i \quad (4.32)$$

gdje je σ koeficijent površinske napetosti koji ovisi o tome koji fluidi tvore površinu, npr. voda i zrak. Prilikom izračuna normale na površinu izrazom (4.30) i Laplasijana obojanosti u jednadžbi (4.31) korištena je *Poly6* zaglađujuća funkcija u skladu s poglavljem 4.1.

Za čestice koje su udaljenije od površine fluida vrijedi $|\mathbf{n}_i| \rightarrow 0$, što uzrokuje numeričku nestabilnost simulacije. To se može spriječiti tako da se prilikom izračunavanja sile površinske napetosti za česticu i , izraz (4.32) evaluira samo ako vrijedi:

$$|\mathbf{n}_i| \geq l \quad (4.33)$$

gdje je l pozitivna konstanta koja se zadaje na početku simulacije.

Treba napomenuti da u stvarnosti površinska napetost nastaje na granici između dva fluida (npr. voda i zrak) i stoga se sile na granici fluida djelomično balansiraju posljedicom toga što s druge strane postoje molekule neke druge vrste. U navedenom modelu

zanemarena je mogućnost postojanja drugog fluida i stoga ovakav model sila površinske napetosti može rezultirati nešto manje realnom simulacijom.

4.2.3.3 Interakcijske sile

Budući da je cilj napraviti interaktivnu simulaciju vodene površine, prepostavka je da korisnik može na neki način utjecati na površinu vode. To se može napraviti na različite načine. Može mu se dozvoliti promjena pojedinih parametara fluida poput gustoće mirovanja, viskoznosti i gravitacije, ili mu se može dozvoliti da upravljanjem okolinom posredno utječe na ponašanje vode. Tipični primjeri bi bili dodavanje pojedinih objekata koji bi se sudarali s vodom ili pomicanje spremnika u kojem se nalazi voda. Također se mogu dodati pojedine sile koje bi simulirale stvaranje valova ili utjecaj vjetra na vodenu površinu.

4.3. SPH pseudokod

Sad kad imamo napisane sve potrebne jednadžbe za opis ponašanja fluida, odnosno čestica koje ga sačinjavaju, možemo napisati općeniti pseudokod SPH simulacijske petlje:

```
1  |  dok traje simulacija:
2  |    za svaku česticu u fluidu:
3  |      resetiraj gustoću i sile
4  |      pronađi susjedne čestice
5
6      za svaku česticu u fluidu:
7          za svaku česticu u susjednim česticama:
8              računaj gustoću           // izraz (4.17)
9              računaj tlak            // izraz (4.24)
10
11     //računa unutarnje sile fluida
12     za svaku česticu u fluidu:
13         za svaku česticu u susjednim česticama:
14             računaj tlačnu силу      // izraz (4.19)
15             računaj viskoznu силу   // izraz (4.26)
16
17     //računa vanjske sile fluida
18     za svaku česticu u fluidu:
19         računaj gravitacijsku силу // izraz (4.27)
20         za svaku česticu u susjednim česticama:
21             računaj površinsku napetost // izraz (4.32)
22
23     //pomiče čestice
24     za svaku česticu u fluidu:
25         računaj akceleraciju        // izraz (4.16)
26         računaj novu poziciju i brzinu
27         pomakni česticu
```

Napisani pseudokod u velikoj mjeri je već objašnjen zakomentiranim izrazima i prethodnim poglavljima. Kao još uvijek neobjašnjeni dijelovi pseudokoda ostaju linije 4 i 26, no njih ćemo detaljnije obraditi u poglavlju koje slijedi.

5. Implementacija

Za implementaciju predstavljenog algoritma koristio sam programski jezik C++ koji je praktički postao standard u svijetu računalne grafike, grafičko programsko sučelje *OpenGL* i njegovo proširenje *FreeGLUT*, te matematičku biblioteku *glm OpenGL Mathematics*.

5.1. Struktura programa

Program se sastoji od šest klasa: *Particle*, *WaterContainer*, *Wall*, *Renderer*, *RendererSmoothPoint*, *RendererSolidSphere* te od glavnog programa *main.cpp*.

Tijekom izvršavanja programa najviše surađuju modul *main.cpp* i klasa *Particle*. Modul *main.cpp* implementira funkcije za obradu korisničkih akcija poput pritiska miša ili tipke, stvara čestice i spremnik vode te pokreće animaciju i nad svakom česticom poziva funkcije klase *Particle* u skladu sa pseudokodom prikazanim u poglavlju 4.3.

Klasa *Particle* sadrži varijable koje čuvaju poziciju, brzinu, akceleraciju, masu, gustoću, tlak i vrijednost pojedinih sila koje djeluju na česticu. Svaka od navedenih veličina ima implementiranu metodu koja izračunava njenu vrijednost.

Klasa *WaterContainer* sadrži referencu na više objekata klase *Wall* koji imaju podatke o vrhovima pojedinog zida. Te dvije klase zajedno čine zatvoreni sustav spremnika vode i nose podatke poput njegove duljine, visine i širine što je potrebno za detekciju kolizije. Implementacija detekcije kolizije i reakcije na nju detaljnije je obrađena u poglavlju 5.5.

Klase *RendererSmoothPoint* i *RendererSolidSphere* implementiraju apstraktну klasu *Renderer* koja pruža sučelje za iscrtavanje vode. Takva organizacija omogućava dinamičku promjenu načina iscrtavanja te lako dodavanje novih načina. Načini iscrtavanja vode dodatno su objašnjeni u poglavlju 5.6.

5.2. Parametri simulacije

U ovom radu spomenuto je nekoliko parametara čija se vrijednost zadaje prije pokretanja simulacije i ne mijenja tijekom izvršavanja programa. Stabilnost i vjernost simulacije uvelike ovise o vrijednostima tih parametara i stoga se te vrijednosti moraju pažljivo odabratи. Budući da to nije nimalo lagan posao, vrijednosti parametara korištenih u ovom radu preuzete su iz rada *Lagrangian Fluid Dynamics Using Smoothed Particle Hydrodynamics* (Kelager, 2006). Tablica 5.1 prikazuje parametre i njihove vrijednosti. Za te vrijednosti parametara, simulacija rezultira ponašanjem fluida koje nalikuje ponašanju vode. Promjenom nekog od parametara mijenjaju se svojstva fluida kojeg simuliramo, pa se na taj način mogu simulirati i neke druge tekućine ili plinovi.

Tablica 5.1 Parametri simulacije i njihove vrijednosti

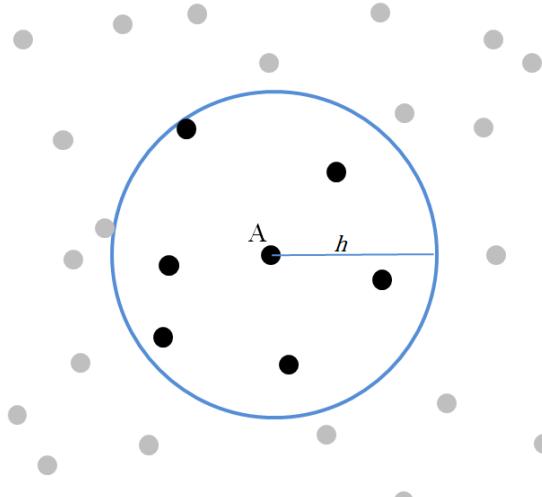
Naziv	Simbol	Vrijednost	Jedinica
Gravitacijska akceleracija	\mathbf{g}	$[0, -9.81, 0]^T$	$\frac{\text{m}}{\text{s}^2}$
Vremenski korak	Δt	0.01	s
Gustoća mirovanja	ρ_0	998.29	$\frac{\text{kg}}{\text{m}^3}$
Masa čestice	m	0.02	kg
Viskoznost	μ	3.5	$\text{Pa} \cdot \text{s}$
Površinska napetost	σ	0.0728	$\frac{\text{N}}{\text{m}}$
Prag normale na površinu	l	7.065	–
Konstanta plina	k	3	J
Restitucija	c_R	0	–
Radius zaglađivanja	h	0.0457	m

Od navedenih parametara do sad još nisu spomenuti vremenski korak i restitucija. Vremenski korak je parametar koji određuje koliko je vremena prošlo u svijetu simulacije između dvije iteracije simulacijske petlje i on služi tome da možemo odrediti gdje će se čestica nalaziti u sljedećem trenutku i koliku će brzinu imati što je detaljnije obrađeno u poglavljju 5.4. Koeficijent restitucije je parametar koji koristimo pri sudaru čestice vode s nekim drugim tijelom i detaljnije je objašnjen u poglavljju 5.5.2.

5.3. Nalaženje susjednih čestica

Linija 4 pseudokoda opisanog u poglavlju 4.3 definira jedan naizgled lagan problem, no on može znatno utjecati na brzinu izvođenja simulacije.

Susjedstvo čestica možemo definirati na sljedeći način: čestica B je susjedna čestici A ako i samo ako se nalazi unutar radiusa zaglađivanja h promatrano s pozicije čestice A. Slika 5.1 prikazuje jednostavno susjedstvo u skladu s navedenom definicijom. Čestice koje se nalaze unutar nacrtane kružnice smatraju se susjedima čestici A. Valja napomenuti da je čestica A ujedno i sama sebi susjed. To će biti bitno pri izračunavanju gustoće jer čestica mora doprinositi gustoći na svojoj poziciji. S izračunom sila to nije slučaj jer čestica ne djeluje silom na sebe pa će se u tom slučaju morati vršiti provjera je li susjedna čestica istovjetna promatranoj čestici.

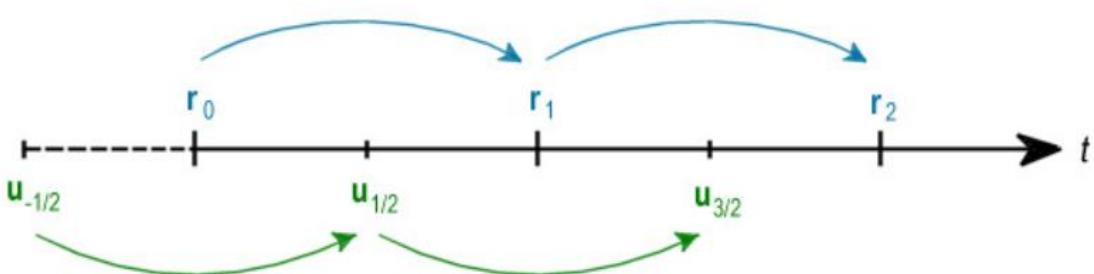


Slika 5.1 Susjedstvo čestice A

Ukoliko listu susjednih čestica gradimo tako da odmah za svaku česticu provjerimo koje čestice se nalaze unutar radiusa h od njene pozicije, jasno je da je riječ o složenosti $O(n^2)$ gdje je n broj čestica kojima simuliramo vodu. Ukoliko vodu simuliramo velikim brojem čestica, zbog kvadratne složenosti pretraživanja susjedstva simulacija postaje jako spora. Ovaj problem rješava se prethodnom izgradnjom *hash* rešetke (Onderik, 2007) koja značajno smanjuje prostor pretraživanja, no u ovom radu implementirana je jednostavnija i sporija metoda bez spomenute rešetke.

5.4. Leap-Frog integracija

Da bi se simulirao tok vode, svaku česticu se pomiče u vremenu koristeći konstantni vremenski korak Δt . Uz pomoć jednadžbe (4.16) izračuna se akceleracija čestice te se njena nova pozicija i brzina izračunava leap-frog integracijom. Leap-Frog integracija je dobila ime po tome što se pozicije i brzine međusobno preskaču, odnosno pozicije se izračunavaju za $t = m\Delta t$, $m \in N_0$, a brzine za $t = \left(m - \frac{1}{2}\right)\Delta t$, $m \in N_0$. Slika 5.2 ilustrira opisani koncept.



Slika 5.2 Shema leap-frog integracije. Horizontalna linija predstavlja vrijeme t .

Kada se prvi put izračuna akceleracija čestice, potrebno je inicijalizirati leap-frog integrator izrazom:

$$\mathbf{u}_{-\frac{1}{2}\Delta t} = \mathbf{u}_0 - \frac{1}{2}\Delta t \mathbf{a}_0 \quad (5.1)$$

gdje su \mathbf{u}_0 i \mathbf{a}_0 brzina i akceleracija čestice u trenutku $t = 0$. Sada se za svaki idući trenutak, brzina i pozicija računaju izrazima:

$$\mathbf{u}_{t+\frac{1}{2}\Delta t} = \mathbf{u}_{t-\frac{1}{2}\Delta t} + \Delta t \mathbf{a}_t \quad (5.2)$$

$$\mathbf{r}_{t+\Delta t} = \mathbf{r}_t + \Delta t \mathbf{u}_{t+\frac{1}{2}\Delta t} \quad (5.3)$$

Leap-Frog integrator je pogodan jer se prilikom računanja sljedeće pozicije čestice koristi njena prosječna brzina između dva trenutka. Dobivenu prosječnu brzinu ne treba miješati s brzinama potrebnim za izračunavanje vrijednosti sila koje djeluju na česticu. Za izračunavanje tih vrijednosti u trenutku t potrebna je brzina čestice \mathbf{u}_t u trenutku t , a ne njena brzina $\mathbf{u}_{t+\frac{1}{2}\Delta t}$ u trenutku $t + \frac{1}{2}\Delta t$. Brzinu čestice u trenutku t možemo dobiti aproksimacijom:

$$\mathbf{u}_t \approx \frac{\mathbf{u}_{t-\frac{1}{2}\Delta t} + \mathbf{u}_{t+\frac{1}{2}\Delta t}}{2} \quad (5.4)$$

Linija 26 pseudokoda opisanog u poglavlju 4.3 koristi leap-frog integraciju za računanje brzine i pozicije čestice u sljedećem trenutku.

5.5. Obrada sudara

Često se prilikom izračuna pozicije čestice u sljedećem trenutku njena pozicija izračuna izvan granica spremnika na koji je voda ograničena. Zbog toga mora postojati mehanizam koji će prepoznavati takve situacije i pravilno ispravljati poziciju i brzinu čestice. Taj mehanizam možemo nazvati obrada sudara i on se također izvršava u liniji 26 pseudokoda zajedno s leap-frog integracijom. Obradu sudara možemo podijeliti na dva koraka: detekcija sudara i reakcija na sudar.

5.5.1. Detekcija sudara

Iako je detekcija sudara u općenitom slučaju vrlo složen i računski zahtjevan problem, ukoliko se ograničimo na određene primitive on se može značajno pojednostavniti. U ovom radu je detekcija sudara napravljena samo za kvadar kao spremnik fluida, no jednadžbe bi ostale gotovo identične u slučaju da se kvadar koristi kao predmet koji se sudara s vodom izvana, a ne iznutra.

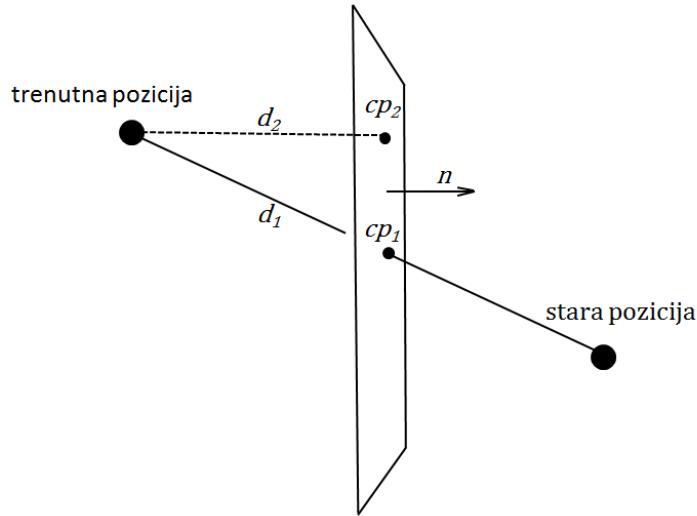
Za kvadar duljine l , širine w i visine h definiramo vektor $\mathbf{v} = [\frac{l}{2}, \frac{w}{2}, \frac{h}{2}]$. Taj vektor zapravo govori koliko su stranice kvadra udaljene od njegovog središta \mathbf{c} . Neka se čestica nalazi na poziciji \mathbf{r} . Komponente vektora \mathbf{r} su udaljenosti čestice od koordinatnih osi u sustavu scene. Zbog jednostavnosti, u ovom radu je spremnik vode centriran u ishodište globalnog sustava scene tj. $\mathbf{c} = [0, 0, 0]$ te su njegove osi poravnate s osima globalnog koordinatnog sustava. Nije implementirana mogućnost translacije i rotacije spremnika te se to ostavlja mogućem budućem radu na ovom zadatku. U tom slučaju se jednadžbe koje slijede moraju poopćiti, ali shvaćanje detekcije kolizije i reakcije ostaje jednako.

Kad je neka čestica na poziciji \mathbf{r} izvan kvadra, neka komponenta vektora \mathbf{r} bit će veća od iste komponente vektora \mathbf{v} . Odnosno, možemo pisati da se dogodila kolizija ukoliko vrijedi:

$$[abs(\mathbf{r}) - \mathbf{v}]_{max} > 0 \quad (5.5)$$

gdje operator \max vraća komponentu vektora s najvećom vrijednošću, a funkcija abs vektor s absolutnim vrijednostima komponenti.

Ako se nakon navedene provjere uspostavi da je došlo do kolizije, potrebno je izračunati mjesto kolizije \mathbf{cp} i normalu ravnine \mathbf{n} u toj točki. Slika 5.3 prikazuje dvije mogućnosti računanja točke kolizije: jednu preciznu i jednu aproksimacijsku. U ovom radu se koristila aproksimacijska točka budući da se brže izračunava, a ne utječe vidljivo na ponašanje vode jer su udaljenosti d_1 i d_2 gotovo jednake za mali vremenski korak Δt .



Slika 5.3 Određivanje točke kolizije

Aproksimacijsku točku kolizije možemo dobiti projekcijom trenutne pozicije \mathbf{r} na ravninu s kojom je došlo do sudara:

$$\mathbf{cp} = \min(\mathbf{v}, \max(-\mathbf{v}, \mathbf{r})) \quad (5.6)$$

gdje su \min i \max funkcije koje vraćaju vektor s minimalnim odnosno maksimalnim vrijednostima uspoređujući dva vektora po komponentama. Udaljenost d je dubina s kojom čestica probila ravninu i ona iznosi:

$$d = |\mathbf{cp} - \mathbf{r}| \quad (5.7)$$

dok se normala na ravninu dobije izrazom:

$$\mathbf{n} = \frac{\mathbf{cp} - \mathbf{r}}{d} \quad (5.8)$$

5.5.2. Reakcija na sudar

Nakon što smo utvrdili da je došlo do sudara, počinje faza reakcije na sudar. Pozicija čestice i se projicira na ravninu sudara što znači da poprima vrijednost točke kolizije:

$$\mathbf{r}_i = \mathbf{cp} \quad (5.9)$$

To znači da ćemo zanemariti vrijeme koje je proteklo nakon trenutka kolizije. To radimo zato što želimo izbjegći pomicanje čestice po prostoru bez da računamo sile koje djeluju na nju. Kad to radimo, smještamo česticu tamo gdje ona ne bi došla prirodnim putem što uglavnom rezultira snažnim titranjem čestica na rubovima sustava jer ju opetovano smještamo na jedno mjesto dok je sile pomicu na drugo.

Nova brzina čestice se računa standardnim izrazom za refleksiju:

$$\mathbf{u}_i = \mathbf{u}_i - 2(\mathbf{u}_i \cdot \mathbf{n})\mathbf{n} \quad (5.10)$$

Taj izraz vrijedi za savršeno elastične sudare. Ukoliko želimo moći upravljati elastičnošću sudara uvodimo novi parametar - koeficijent restitucije c_R . On poprima vrijednost iz intervala $[0,1]$ gdje za $c_R = 0$ imamo savršeno neelastični sudar, a za $c_R = 1$ savršeno elastični sudar. Dakle, vrijedi:

$$\mathbf{u}_i = \mathbf{u}_i - (1 + c_R)(\mathbf{u}_i \cdot \mathbf{n})\mathbf{n} \quad (5.11)$$

U jednadžbi (5.11) se prilikom refleksije računa brzina čestice u sljedećem trenutku odnosno u trenutku koji dolazi nakon što prođe čitav vremenski korak Δt . Budući da smo već napomenuli da ćemo ignorirati vrijeme proteklo nakon kolizije, moramo i novu brzinu računati za trenutak u kojem je došlo do kolizije kako se ne bi dogodilo da se čestica odbije većom brzinom nego što je imala prije sudara. Zato se koristi izraz:

$$\mathbf{u}_i = \mathbf{u}_i - (1 + c_R \frac{d}{\Delta t |\mathbf{u}_i|})(\mathbf{u}_i \cdot \mathbf{n})\mathbf{n} \quad (5.12)$$

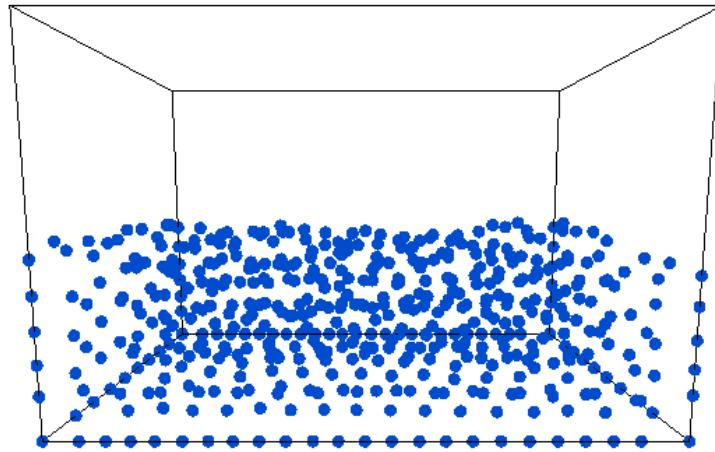
gdje je pretpostavka da je došlo do sudara tj. da je $|\mathbf{u}_i| > 0$.

5.6. Isrtavanje

U ovom radu implementirane su dvije metode za isrtavanje čestica vode u klasama *RendererSmoothPoint* i *RendererSolidSphere*.

Prva metoda implementirana je u klasi *RendererSmoothPoint* i koristi *OpenGL*-ove funkcije za isrtavanje točaka. Uz pomoć *glDrawArrays* jednim funkcijskim pozivom na crtanje pošaljemo koordinate svih točaka odnosno pozicija čestica te ih uz pomoć

`glEnable(GL_POINT_SMOOTH)` zagladimo tako da se umjesto kvadrata na poziciji čestice iscrta krug. Slika 5.4 prikazuje rezultat takvog iscrtavanja čestica vode. Ovakav način iscrtavanja dovoljan je za 2D scene, ali zbog toga što su sve točke jednake veličine i što ih se ne može osjenčati, 3D scene zahtijevaju neku drugu metodu.

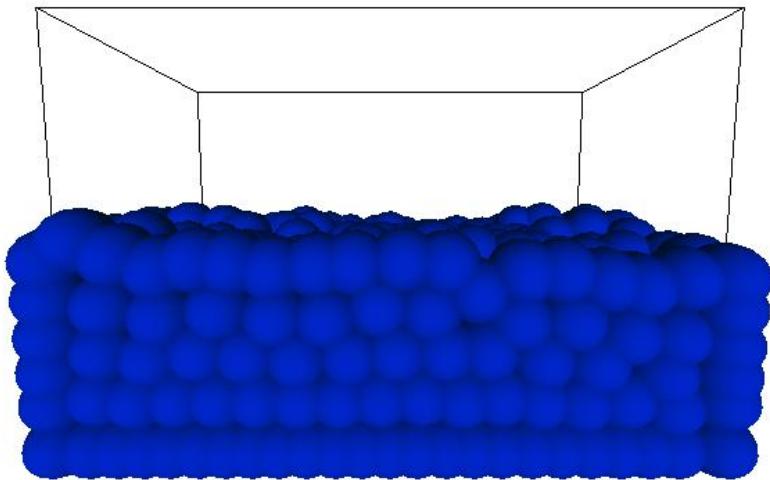


Slika 5.4 IsCRTAVANJE VODE TOČKAMA

Sljedeća metoda implementirana je u klasi *RendererSolidSphere*. Za svaku česticu nacrtajte sferu radijusa r pomoću funkcije *glutSolidSphere* te se zatim translatira na poziciju čestice. Radijus r izračuna se na početku izvršavanja programa izrazom:

$$r = \sqrt[3]{\frac{3m}{4\pi\rho_0}} \quad (5.13)$$

gdje je m masa čestice, a ρ_0 gustoća mirovanja vode. Ova metoda je sporija od prethodne, ali prikazuje čestice vode u 3D. Slika 5.5 prikazuje rezultat ovakvog iscrtavanja.



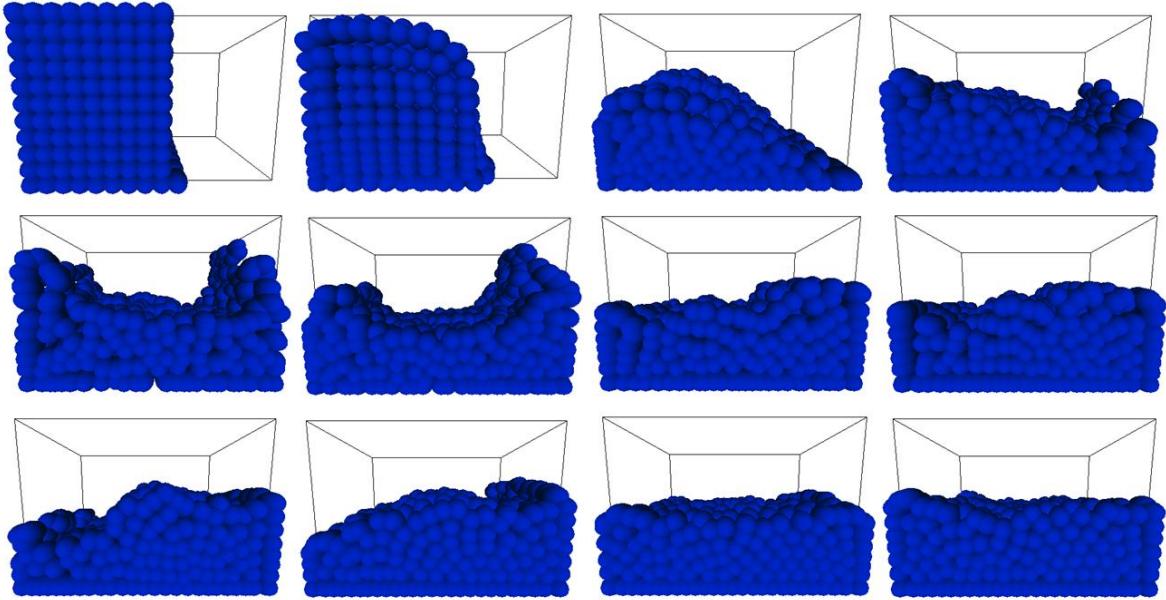
Slika 5.5 IsCRTavanje vode sferama radijusa r uz dodano osvjetljenje

Razvijene su i metode koje fluid diskretiziran česticama interpoliraju u kontinuirani fluid od kojih je najčešća metoda *Marching Cubes* (Lorensen, 1987), ali njenu implementaciju ćemo ostaviti za mogući daljnji rad na ovom zadatku.

6. Rezultati

Nakon implementacije opisanog postupka, dobivena je simulacija koja u realnom vremenu radi s maksimalno 1500 čestica na računalu s procesorom Intel® Core™ i5-2500K s dvije jezgre s taktovima 3.3GHz i 3.6GHz te 8GB radne memorije na operacijskom sustavu Windows7. Optimalna izvedba dobije se korištenjem 500-1000 čestica.

Slika 6.1 prikazuje tipičan početak izvođenja simulacije kad se čestice pravilno generiraju na lijevoj strani spremnika vode. Korišteno je 700 čestica i prikazana je svaka deseta slika od početka simulacije do stanja kad je voda blizu mirovanja. Voda popunjava dno spremnika pod utjecajem gravitacije te posljedično zapljuškuje lijevu i desnu stranu spremnika. Nakon što se voda koja je zapljušnula strane spremnika vratiла, uzrokovala je valove koji su interferirali i stvorili val u sredini posude. Budući da je zapljuškivanje desne strane posude bilo jače, u interferenciji dominira val koji je nastao na desnoj strani te se on propagira na lijevu stranu posude. Ubrzo zatim, voda poprima stanje blizu mirovanja.

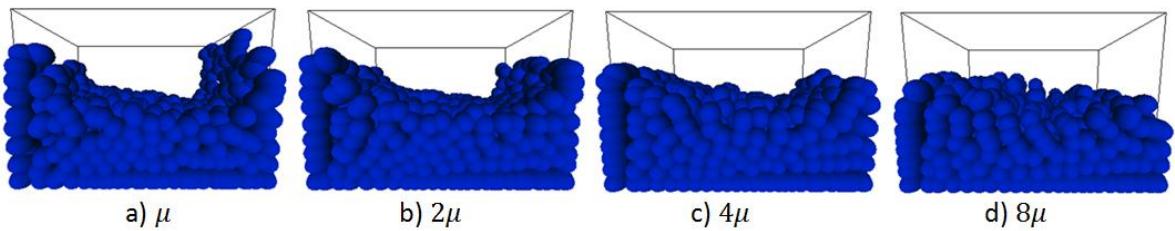


Slika 6.1 Tok vode na početku simulacije

6.1. Utjecaj pojedinih parametara

Budući da parametri imaju veliki utjecaj na realističnost prikaza vodene površine, prikazat ćemo neke primjere i uvjeriti se u njihov utjecaj.

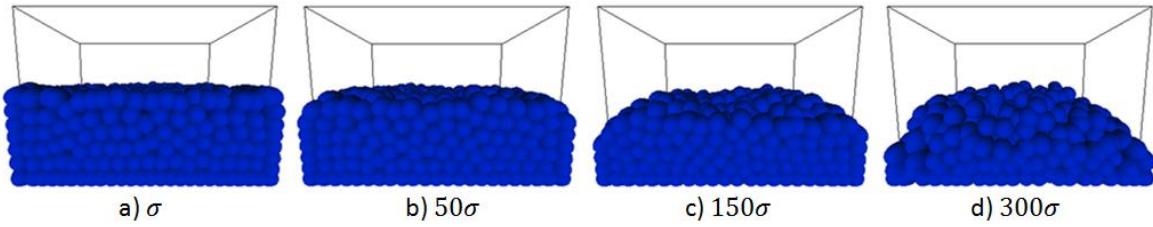
Slika 6.2 ilustrira utjecaj dinamičke viskoznosti μ na ponašanje fluida. Budući da voda sama po sebi ima jako malu viskoznost, smanjivanjem viskoznosti neće se vidjeti značajna promjena u ponašanju vode. Stoga slike s lijeva na desno prikazuju isti trenutak gdje svaka iduća slika ima duplo veću vrijednost parametra μ . Jasno je da povećavanjem viskoznosti dinamičnost vode opada i ona puno brže dolazi u stanje mirovanja.



Slika 6.2 Utjecaj parametra μ na tijek simulacije

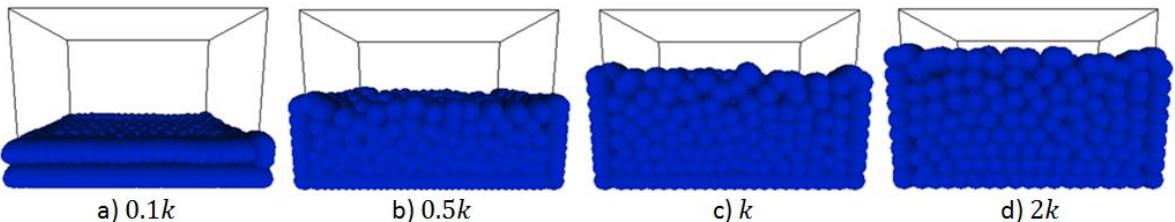
Sljedeći parametar je koeficijent površinske napetosti σ . Iz sličnih razloga kao i za viskoznost, nećemo prikazivati vrijednosti manje od one koju inače koristimo. Slika 6.3 ilustrira utjecaj koeficijenta površinske napetosti na ponašanje fluida. Na slici b) možemo primijetiti kako su se rubovi vode malo zaoblili. Na slici c) zaobljenost je već značajno

izražena, dok se na slici d) već vidi da voda želi zauzeti oblik kugle. Sve slike prikazuju isti trenutak od početka simulacije.



Slika 6.3 Utjecaj parametra σ na tijek simulacije

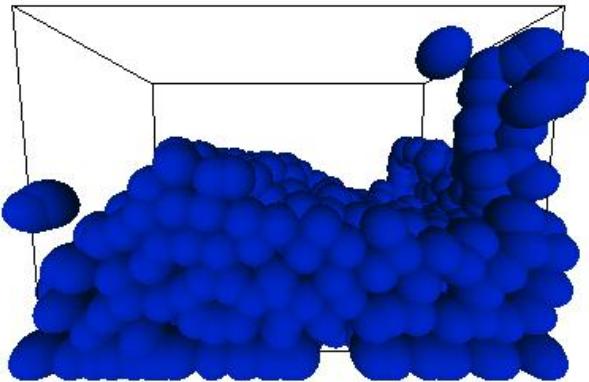
Konstanta plina k je vrlo osjetljiv parametar i što je veći to bolje osigurava nestlačivost fluida. Nažalost, kako povećavamo k tako moramo smanjivati vremenski korak Δt da bi simulacija ostala stabilna stoga nestlačivost dolazi s cijenom interaktivnosti. Na slici 6.4 prikazan je utjecaj vrijednosti konstante plina na tijek simulacije. Budući da nestlačivost zapravo nije osigurana, može se reći da voda u stanju mirovanja skakuće stoga je na sljedećim slikama prikazan minimalan volumen vode za iznos konstante plina kao mjere za nestlačivost. Veličina spremnika je stvaran volumen koji bi voda trebala zauzimati. Kako nemamo svojstvo nestlačivosti možemo vidjeti da ni na jednoj slici voda ne zauzima svoj stvarni volumen, ali možemo vidjeti i da što je veća konstanta plina to se zauzeti volumen više približava stvarnome.



Slika 6.4 Utjecaj parametra k na tijek simulacije

6.2. Stabilnost

U situacijama kad se voda pušta da pada s visine redovito dolazi do nestabilnosti uzrokovane prevelikim tlakom na dnu spremnika. Kao posljedica, čestice na dnu spremnika neko vrijeme jako titraju, a zatim izlete velikom brzinom iz vode. Slika 6.5 ilustrira opisano ponašanje. Nakon kratkog vremena tlak među česticama se izbalansira i voda uskoro poprimi stanje mirovanja.

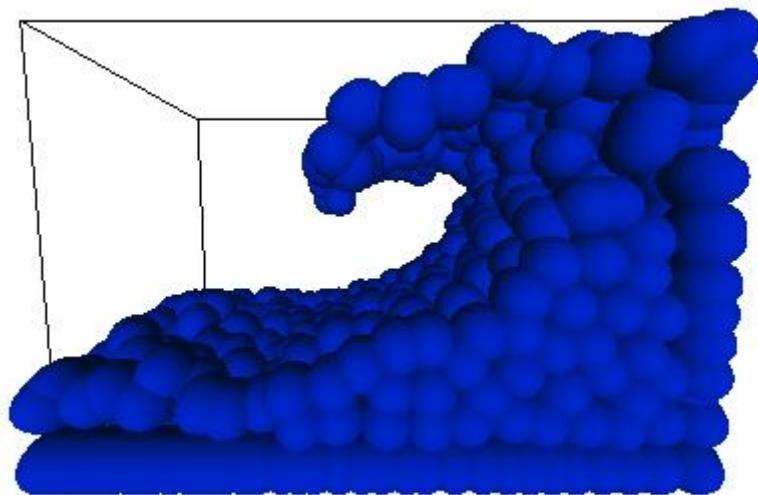


Slika 6.5 Nestabilnost uzrokovana prejakinim tlakom na dnu spremnika

Osim toga, reakcija na sudar očito nije riješena u potpunosti ispravno pa čestice koje izlete katkad ostanu na gornjoj strani spremnika. Ukoliko se voda približi takvoj čestici, čestica će se priključiti ostatku vode.

6.3. Interakcija s korisnikom

Implementirani su i osnovni načini interakcije s korisnikom. Strelicama gore, dolje, lijevo i desno primjenjuju se sile koje djeluju na cijeli fluid i usmjeravaju ga u željenom smjeru. Na taj način se mogu proizvesti interesantni vizualni efekti poput slamanja valova kao što je prikazano na slici 6.6.



Slika 6.6 Slomljeni val

Na lijevi klik miša stvara se sila koja djeluje odozgo na manje područje i stvara vrtložno gibanje fluida. Postoji još mogućnost dodavanja novih čestica u obliku kapi koje padaju u fluid na desni klik miša.

7. Budući razvoj

Postoji niz kategorija u kojima bi se ovaj rad mogao dalje unaprijediti, od brzine simulacije, realističnijeg prikaza pa do povećanja mogućnosti interakcije s vodom.

Prvo što je potrebno napraviti je povećati broj čestica koje se mogu simulirati u realnom vremenu. Veći broj čestica znači veći volumen vode što znači složenije i zanimljivije gibanje vodene površine. Najlakši način za omogućiti simulaciju više čestica u realnom vremenu je ubrzati pretragu susjednih čestica. Trenutno se pretraživanje susjednih čestica obavlja u složenosti $O(n^2)$ i prvo poboljšanje moglo bi se napraviti koristeći već spomenutu *hash* rešetku. Nadalje, u pojedinim blokovima simulacijske petlje čestice su međusobno neovisne jedna o drugoj što otvara mogućnost da se njihove veličine izračunavaju višedretveno što bi značajno ubrzalo program. Također, trenutno se program u cijelosti izvršava na CPU i kad bi se prilagodio za GPU i iskoristio mogućnosti *pixel shadera* koji može obaviti milijarde floating point operacija u sekundi paralelno, sigurno bi došlo do značajnih ubrzanja.

Kao drugo moguće poboljšanje bilo bi povećanje realističnosti. Trenutno se voda prikazuje kao niz čestica. Napredniji prikaz vode uključivao bi interpolaciju čestica u kontinuirani fluid. Od metoda koje ostvaruju takav prikaz vode već smo spomenuli *Marching Cubes*. Dodatno bi se mogla ostvariti refleksija i refrakcija svjetla na fluidu te kaustika što bi rezultiralo fotorealističnim prikazom vode. Osim navedenog, mogli bi se prikazati još neki efekti vodene površine poput pjene što bi dodatno doprinijelo realističnosti simulacije.

Jedan od važnijih aspekata simulacije je i interakcija korisnika s vodom. Trenutno je interakcija ograničena na dodavanje pojedinih sila i čestica no kad bi se napravila općenitija detekcija kolizije, mogla bi se ostvariti interakcija vode s raznolikim objektima i njen tok ne bi morao biti ograničen na statičan spremnik konkretne vrste već bi se ona mogla kretati kroz komplikiraniji prostor.

8. Zaključak

Područje fizikalno temeljene simulacije fluida vrlo je opsežno i interesantno područje, ali i matematički iznimno zahtjevno te je bilo potrebno pročitati dosta literature da se djelomično upoznam s vladajućim pojmovima i mogućnostima u tom području. Budući da se u računalnoj grafici još uvijek nije uspjela postići interaktivnost s velikim količinama vodenih površina i prihvatljivom realističnošću, nove metode će se sigurno razvijati i samo je pitanje vremena kad ćemo razvijenu tehnologiju vidjeti i u računalnim igrama.

Realističnost i stabilnost simulacije vode jako ovise o parametrima i stoga oni moraju biti pažljivo odabrani. Želimo imati što realističniju simulaciju, ali ne želimo riskirati da se zbog toga, ukoliko se voda tretira na nepredviđen način, simulacija počne neobično ponašati. Parametri su međusobno ovisni što dodatno otežava pronalaženje odgovarajućih parametara pa mogu samo zaključiti da je moj prvotni pokušaj da ih samostalno podesim s razlogom završio neuspješno.

Metoda SPH je relativno jednostavna metoda za simulaciju tekućina, ali posjeduje jedan veliki nedostatak - ne garantira nestlačivost. Budući da nestlačivost igra veliku ulogu u ponašanju vode, kad bih ponovo radio rad na neku sličnu temu pokušao bih s nekom naprednijom verzijom SPH metode koja garantira nestlačivost poput *Predictive-Corrective Incompressible SPH* (Solenthaler, 2009).

9. Literatura

- Kelager, M.: Lagrangian Fluid Dynamics Using Smoothed Particle Hydrodynamics, Diplomski rad, University of Copenhagen, 2006.
- Auer, S.: Realtime Particle-Based Fluid Simulation, Magistarski rad, Technische Universität München, Fakultät für Informatik, 2008.
- Macklin M., Müller, M.: Position Based Fluids, To appear at ACM Transactions on Graphics, Volume 32, No. 4, 2013.
- Yuksel, C.: Real-Time Water Waves with Wave Particles, Doktorska disertacija, Texas A&M University, 2010.
- Tan J., Yang X.: Physically-based Fluid Animation: A Survey, Science in China Series F: Information Sciences, Volume 52., 2009.
- Chrisman C.: A Comparison of Grid-Based Techniques for Navier-Stokes Fluid Simulation in Computer Graphics, Magistarski rad, University of California, San Diego, 2008.
- Solenthaler B., Pajarola R.: Predictive-Corrective Incompressible SPH, ACM Transactions on Graphics, Volume 28, 2009.
- Onderik J., Ďurkovič R.: Efficient Neighbor Search for Particle-based Fluids, Journal of the Applied Mathematics, Statistics and Informatics (JAMSI), Volume 2, No. 3, 2007.
- Lorensen W.E., Cline H.E.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm, Computer Graphics, Volume 21, No. 4, 1987.

10. Sažetak

Naslov: Fizikalno temeljena simulacija površine vode

U ovom radu ukratko je opisan značaj fizikalno temeljene simulacije površine vode te njena primjena u industriji igara, filmova, ali i u znanosti. Dana je usporedba standardnih pogleda na fluid te su predstavljene prednosti i mane svakog od njih. U konačnici je za implementaciju odabran Lagrangeov fluid jer je pogodan za simulaciju u realnom vremenu. Za implementaciju je odabrana metoda zaglađene čestice (*Smoothed Particle Hydrodynamics*) koja je već dugo vremena popularna metoda za simuliranje Langrangeovog fluida i stoga jako dobro dokumentirana. Njen algoritam je detaljno razrađen i predstavljene su sve važne jednadžbe za njegovu uspješnu implementaciju. Algoritam je potom implementiran te su detalji implementacije dodatno objašnjeni. Na kraju rada dan je pogled u budućnost i moguća poboljšanja ostvarenog programskog produkta.

Ključne riječi: voda, fizikalno temeljena simulacija fluida, Lagrangeov fluid, SPH, simulacija u stvarnom vremenu

11. Abstract

Title: Physically based simulation of water surface

In this thesis a short introduction was given on importance of physically based simulation of water surface and its application in industry of video games, films and also in science. A comparison was given between different views on a fluid and their positive and negative sides were presented. Because of its real-time advantage, a Lagrangian fluid was chosen for implementation using *Smoothed Particle Hydrodynamics* method which has been popular for a while and is therefore well documented. Its algorithm was presented in detail covering all the important equations necessary for a successful implementation. The algorithm was then implemented and implementation details were additionally explained. At the end of the thesis, possible future improvements on the software were given as a guidance for a further work on this topic.

Keywords: water, physically based fluid simulation, Lagrangian fluid, Smoothed Particle Hydrodynamics, real-time simulation