

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2163

# **SIMULACIJA FLUIDA TEHNIKOM SPH**

Jurij Kos

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2163

# **SIMULACIJA FLUIDA TEHNIKOM SPH**

Jurij Kos

Zagreb, lipanj 2020.

## DIPLOMSKI ZADATAK br. 2163

Pristupnik: **Jurij Kos (0036491447)**  
Studij: Računarstvo  
Profil: Računarska znanost  
Mentor: prof. dr. sc. Željka Mihajlović

Zadatak: **Simulacija fluida tehnikom SPH**

### Opis zadatka:

Proučiti fizikalne osnove dinamike fluida. Proučiti tehniku hidrodinamike zaglađujućih čestica SPH (engl. Smoothed Particle Hydrodynamics) u simulacijama dinamike fluida. Posebice obratiti pažnju na površinsku napetost fluida. Razraditi trodimenzionalni implementacijski model uz mogućnost djelovanja vanjske sile u simulaciji dinamike. Proučiti mogućnosti ostvarivanja prikaza implicitno dobivenih površina postavljenim modelom. Razraditi i ostvariti simulacijski model uz ostvarivanje animiranog prikaza rezultata. Diskutirati utjecaj različitih parametara. Načiniti ocjenu rezultata i implementiranih algoritama. Izraditi odgovarajući programski proizvod. Koristiti programski jezik C++, programsko grafičko sučelje OpenGL i razvojno programsko okruženje VisualStudio. Rezultate rada načiniti dostupne putem Interneta. Radu priložiti algoritme, izvorne kodove i rezultate uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 30. lipnja 2020.



## Sadržaj

Uvod .....	1
1. Fizikalni opis fluida .....	2
1.1. Fizikalna obilježja fluida .....	2
1.1.1. Gustoća .....	2
1.1.2. Viskoznost .....	2
1.1.3. Tlak .....	3
1.1.4. Hipoteza kontinuuma .....	3
1.2. Navier-Stokesove jednačbe .....	3
1.3. Modeli fluida .....	4
1.3.1. Čestične metode bez mreže (engl. Meshfree Particle Methods) .....	4
1.3.2. Metode s rešetkom .....	5
2. Hidrodinamika zaglađujućih čestica .....	6
2.1. Svojstva SPH-a .....	6
2.2. Formulacija SPH-a .....	7
2.3. Jezgrena zaglađujuća funkcija .....	8
2.4. Pseudokod algoritma .....	9
2.4.1. Gustoća .....	9
2.4.2. Sile .....	9
2.4.3. Integracija .....	11
2.5. Površinska napetost .....	11
3. Algoritam pokretnih kocki (engl. Marching Cubes) .....	13
3.1. Triangulacija kocke .....	14
3.2. Računanje normala .....	16
4. Implementacija .....	17

4.1.	SPH.....	18
4.1.1.	Čestica (engl. Particle).....	18
4.1.2.	Vremenski korak simulacije .....	18
4.1.3.	Pronalazak susjednih čestica – ubrzanje algoritma .....	19
4.2.	Konverzija podataka iz SPH čestica u trodimenzionalnu rešetku .....	19
4.3.	Algoritam pokretnih kocki .....	20
4.4.	Phongov model osvjetljenja.....	20
4.5.	Pokretanje programa.....	22
4.6.	Rezultati.....	23
	Zaključak .....	27
	Literatura .....	28
	Sažetak.....	29
	Summary.....	30

# Uvod

Od zraka koji udišemo, preko oceana koji prekrivaju dvije trećine našeg planeta pa sve do unutrašnjih slojeva planeta fluidi se nalaze svuda oko nas i zaslužni su za neke od najljepših i najimpresivnijih fenomena. Za opis gibanja fluida koristimo diferencijalne jednačbe. Za složenije sustave kakve nalazimo u prirodi jednačbe nemaju analitičko rješenje pa je potrebno numeričkim postupcima i aproksimacijama doći do zadovoljavajućih rezultata. Postoji velik broj metoda koje se koriste za simulaciju fluida. Optimalnu metodu odabiremo ovisno o tome za što planiramo koristiti simulaciju i koliko precizne rezultate trebamo postići. Kod animacija gdje nam je potrebno dobiti prikaz gibanja fluida kroz nekoliko sekundi, a možemo si dozvoliti da je računalo izrađuje satima, odabiremo drugačiju metodu nego kod računalne igre gdje je glavni kriterij izvođenje u stvarnom vremenu. Širina i složenost ovog područja privukla je velik broj programera i istraživača što je u kombinaciji s naglim rastom performansa računala omogućilo velik napredak ovog područja kroz zadnjih dvadeset godina. Današnje animacije izgledaju toliko dobro da ih je ponekad teško razlikovati od stvarnosti.

Tema ovog diplomskog rada je simulacija fluida korištenjem tehnike hidrodinamike zaglađujućih čestica (engl. Smoothed Particle Hydrodynamics - SPH) i prikaz površine fluida mrežom trokuta dobivenom korištenjem algoritma pokretnih kocki (engl. Marching Cubes). U prvom poglavlju su opisana osnovna svojstva fluida, nakon čega slijedi detaljni opis algoritama SPH i marching cubes. Zatim se opisuje implementacija navedenih algoritama za koje su korišteni programski jezik c++ i aplikacijsko programsko sučelje OpenGL. Nadalje, opisan je utjecaj različitih parametara na simulaciju te su razrađena moguća poboljšanja i dana usporedba s drugim metodama.

# 1. Fizikalni opis fluida

Fluidi su tvari koje struje, odnosno neprekidno se deformiraju pod djelovanjem smičnog naprezanja. S obzirom na to mijenjaju li volumen pod utjecajem vanjskih sila, fluide dijelimo na stlačive i nestlačive. Gustoća, tlak, viskoznost i temperatura su svojstva kojima opisujemo fluide i o kojima ovisi njihovo gibanje. Gibanje fluida opisano je Navier-Stokesovom jednadžbom.

## 1.1. Fizikalna obilježja fluida

Za simulaciju gibanje fluida potrebno je poznavati gustoću i viskoznost te stanje opisano temperaturom i tlakom mu kojem se trenutno nalazi.

### 1.1.1. Gustoća

Gustoća je svojstvo definirano kao omjer mase i volumena. Ovo svojstvo neovisno je o veličini promatranog uzorka kod homogenih materijala, dok se kod nehomogenih materijala definira kao promjena mase po jedinici volumena. Mjerna jedinica za gustoću prema SI sustavu je  $\frac{kg}{m^3}$ . Gustoća ovisi o tlaku i temperaturi. Pri povećanju tlaka gustoća se povećava, dok se pri povećanju temperature smanjuje. S obzirom da tekućine smatramo nestlačivima, utjecaj tlaka na njihovu gustoću je zanemariv.

### 1.1.2. Viskoznost

Fluid se opire vanjskim smičnim silama putem viskoznih naprezanja koja se javljaju kao reakcija na brzinu deformacije. U newtonovskim fluidima viskozna naprezanja su proporcionalna brzini deformacije fluida. Koeficijent koji povezuje brzinu deformacije fluida i viskozna naprezanja naziva se dinamička viskoznost fluida i označava se s  $\mu$ . Viskoznost je svojstvo fluida koje opisuje njegov otpor tečenju, a zavisi o temperaturi fluida. Kod tekućina viskoznost pada s porastom temperature, a kod plinova viskoznost raste s porastom temperature.



### 1.1.3. Tlak

Tlak je fizikalna veličina koja opisuje djelovanje sile u smjeru okomitom na površinu. Tlak je određen omjerom komponente sile okomite na površinu i površine na koju se raspoređuje sila. Kod tekućina, tlak se javlja u tri slučaja. Hidrostatski tlak je tlak u unutrašnjosti tekućine, koji nastaje zbog težine tekućine i raste linearno s visinom stupca tekućine iznad promatrane točke. Hidrodinamički tlak je tlak koji nastaje zbog strujanja u fluidu. Tlak u fluidu koji nastaje djelovanjem izvana (npr. pumpa) naziva se hidrauličkim tlakom. Zbroj ovih tlakova je jednak u svakoj točki idealnog fluida što je opisano Bernullijevom jednačinom  $\frac{1}{2}\rho v^2 + \rho gh + p = \text{konstanta}$ , koja proizlazi iz zakona o očuvanje energije.

### 1.1.4. Hipoteza kontinuuma

Kontinuum je matematički model prema kojem materija zadržava svoja fizikalna svojstva pri smanjivanju volumena u diferencijal. Hipoteza kontinuuma omogućuje primjenu integralnog i diferencijalnog računa u mehanici fluida.

## 1.2. Navier-Stokesove jednačine

Navier-Stokesove jednačine opisuju gibanje viskoznih fluida, a proizlaze iz primjene drugog Newtonovog zakona na gibanje fluida. Zasnovane su na pretpostavci da je fluid kontinuum.

$$\nabla \cdot \mathbf{u} = 0, \quad (1)$$

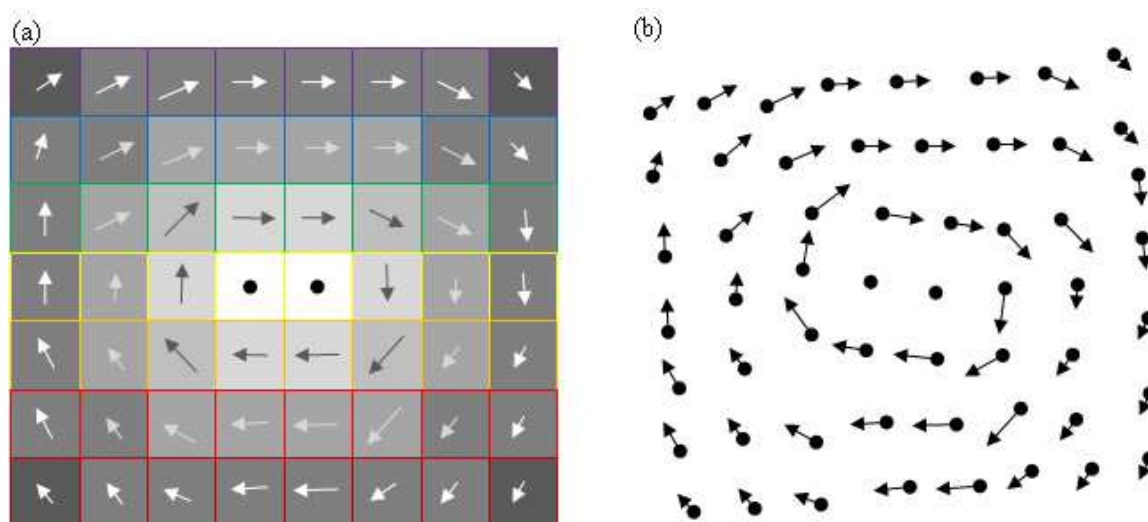
$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \mu \nabla^2 \mathbf{u} + \rho \mathbf{g}, \quad (2)$$

Druga jednačina je drugi Newtonov zakon ( $F=ma$ ) primijenjen na fluide. Jednačinu dijelimo s volumenom. Lijeva strana jednačine je izraz koji bi odgovarao umnošku gustoće i akceleracije. S desne strane jednačine su redom sila koja nastaje zbog razlike tlaka u fluidu, viskozna sila i gravitacijska sila koje djeluju po jediničnom volumenu. Jednačina 2 je zakon očuvanja količine gibanja, ali za potpuni opis toka fluida potrebne su dodatne informacije koje dobivamo iz rubnih uvjeta ili očuvanja mase i jednačine stanja. Prva jednačina je izvedena iz činjenice da gustoća kod nestlačivih fluida ostaje konstantna u vremenu pa je

divergencija brzine jednaka nuli. Ovom jednađbom je opisano očuvanje mase i ona vrijedi samo za nestlačive fluide. U radu se bavimo nestlačivim fluidima.

### 1.3. Modeli fluida

Kada proučavamo gibanja fluida ili deformabilnih čvrstih tijela postoje dva pristupa praćenja njihova kretanja, Eulerov i Lagrangeov (slika 1). Kontinuum možemo tretirati kao sustav čestica u prostoru čije gibanje i svojstva pratimo kroz vrijeme ili tako da pratimo fiksne točke u prostoru i promatramo kako se mijenjaju veličine kojima opisujemo fluid poput gustoće, brzine, temperature i tlaka. Prvi je Lagrangeov pristup koji često nazivamo materijalnim opisom fluida, a drugi je Eulerov pristup koji nazivamo prostornim opisom fluida.



Slika 1 Eulerov i Lagrangeov model fluida

#### 1.3.1. Čestične metode bez mreže (engl. Meshfree Particle Methods)

Dio kontinuuma ili diskretan fizički objekt prikazujemo pomoću konačnog broja diskretnih čestica. Prednosti metoda iz ove skupine su jednostavna paralelizacija i trivijalno očuvanje mase jer česticama masa ostaje konstanta za vrijeme izvođenja programa. Metode iz ove grupe se koriste kod simulacija gdje su granice sustava pomične i fluid se u nekim područjima nalazi kraće vrijeme. Njihova prednost u tom slučaju je što koriste istu količinu memorije neovisno o prostoru koji simuliramo, dok metode koje koriste mrežu trebaju veliku količinu memorije za pokrivanje cijelog prostora. Kod simulacije fluida, najčešće korištena metoda iz ove grupe je hidrodinamika zaglađujućih čestica. Dobivanje površine fluida

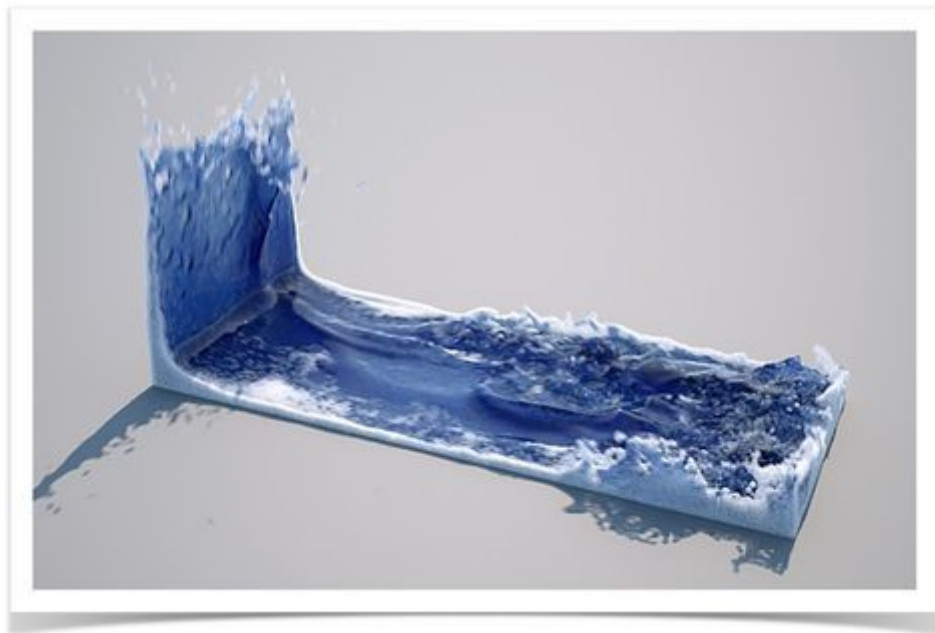
korištenjem ovih metoda je nešto složenije. Jedno od mogućih rješenja je algoritam pokretnih kocki. Algoritam pokretnih kocki daje dobre rezultate u kombinaciji s metodom hidrodinamike zaglađujućih čestica.

### **1.3.2. Metode s rešetkom**

Postoje dvije vrste metoda koje koriste rešetku, metode s Eulerovom rešetkom i metode s Lagrangeovom rešetkom. Metode s Eulerovom rešetkom koriste rešetku fiksiranu u prostoru. Rešetka ostaje nepromjenjena za čitavo vrijeme simulacije pa ne dolazi do numeričkih problema kao u metodama s Lagrangeovom rešetkom koja prati deformacije materijala. Metode s Lagrangeovom rešetkom se rijetko koriste kod simulacije fluida, dok su metode s Eulerovom rešetkom vrlo često korištene. Mane metoda s Eulerovom rešetkom su zahtjevno postavljanje rešetke koje bi odgovaralo složenim tijelima i potreba za rešetkom puno većom od tijela.

## 2. Hidrodinamika zaglađujućih čestica

Metoda hidrodinamike zaglađujućih čestica (engl. Smoothed Particle Hydrodynamics, SPH) je najpopularniji pristup za reprezentaciju fluida česticama. SPH su razvili Lucy, Gingold i Monaghan 1977. za rješavanje problema u astrofizici, ali se ubrzo pokazala pogodnom za probleme u mehanici fluida. Ova metoda se koristi i u komercijalnim programima za simulaciju fluida među kojima je najpoznatiji RealFlow (slika 2).



Slika 2 RealFlow

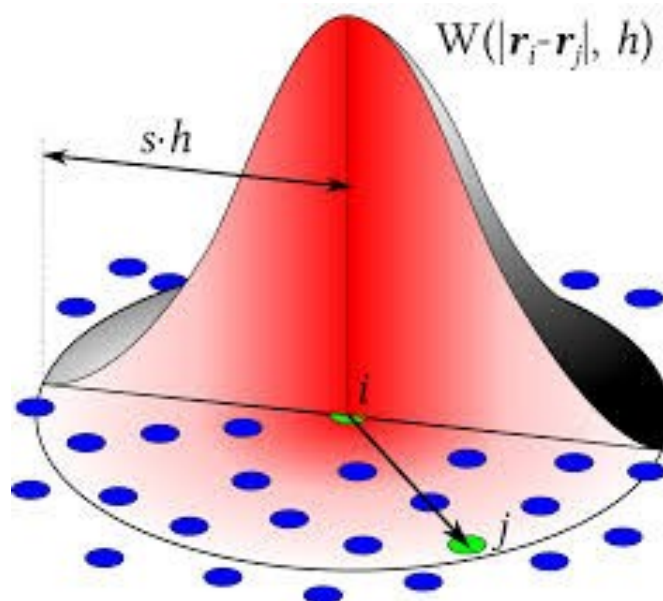
### 2.1. Svojstva SPH-a

SPH je interpolacijska metoda za sustave čestica. U svakoj točki prostora možemo dobiti vrijednost za neko svojstvo fluida tako da zbrojimo utjecaj čestica koje se nalaze u blizini. Tako pretvaramo konačni skup čestica u kontinuirano polje. Ovo je svojstvo iznimno važno jer smo ograničeni računalnim resursima koji su konačni, a prema hipotezi kontinuuma čestica ima infinitezimalni volumen što je istovjetno tome da imamo beskonačno mnogo čestica. SPH je metoda koja volumen fluida dijeli na čestice pa svaka čestica predstavlja mali dio volumena. Svaka čestica djeluje na susjedstvo koje se nalazi unutar njenog radijusa zaglađivanja, a utjecaj izračunavamo pomoću jezgrenih funkcija. Sljedeće ideje se koriste kako bismo zadovoljili opisana svojstva.

1. Nekorištenje mreže (engl. meshfree) – Domena problema se prikazuje pomoću skupa raspodijeljenih čestica.
2. Integralna reprezentacija funkcije – Za aproksimaciju funkcije polja koristi se metoda integralne reprezentacije za što se često susreće i naziv jezgrena aproksimacija (engl. kernel approximation)
3. Kompaktna podrška – Jezgrena aproksimacija se dodatno aproksimira koristeći čestice tako da integral zamijenimo sumom utjecaja svih čestica. Taj postupak se naziva čestična aproksimacija
4. Adaptivnost – Čestična aproksimacija se izvodi svaki vremenski korak i rezultat ovisi o trenutnom rasporedu čestica
5. Lagrangian – Za sve izraze koji se koji su povezani s funkcijama polja u parcijalnim diferencijalnim jednačbama izvode se čestične aproksimacije kako bismo izveli skup običnih diferencijalnih jednačbi koje su ovisne samo o vremenu
6. Dinamičnost – Diferencijalne jednačbe se rješavaju pomoću eksplicitnih integracijskih algoritama kako bi se postiglo brzo napredovanje simulacije i dobila povijest vrijednosti svih varijabli za sve čestice.

## 2.2. Formulacija SPH-a

Prema metodi SPH skalarnu veličinu  $A$  na poziciji  $\mathbf{r}$  dobivamo sumacijom doprinosa svih čestica unutar radijusa zaglađivanja. Utjecaj ovisi o udaljenosti čestica (slika 3).



Slika 3 Utjecaj susjednih čestica

$$A_s(\mathbf{r}) = \int A(\mathbf{r}')W(\mathbf{r} - \mathbf{r}', h)dr' \approx \sum_j A_j \frac{m_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h), \quad (3)$$

U jednadžbi 3 srednji izraz je integralna reprezentacije funkcije, a izraz s desne strane je dobiven čestičnom aproksimacijom koju koristi SPH za računanje vrijednosti određenog svojstva na poziciji  $\mathbf{r}$ .  $m_j$  predstavlja masu čestice  $j$ ,  $\rho_j$  njenu gustoću,  $A_j$  vrijednost  $j$ -te čestice za veličinu koju želimo interpolirati u točki  $\mathbf{r}$ , a  $W$  je zaglađujuća jezgrena funkcija. Zbog linearnosti, derivacije imaju utjecaj samo na zaglađujuću jezgrenu funkciju što omogućuje jednostavno računanje gradijenta i laplasijana. Računanje gradijenta i laplasijana veličine  $A$  prikazano je u izrazima 4 i 5.

$$\nabla A_s(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} \nabla W(\mathbf{r} - \mathbf{r}_j, h), \quad (4)$$

$$\nabla^2 A_s(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} \nabla^2 W(\mathbf{r} - \mathbf{r}_j, h), \quad (5)$$

### 2.3. Jezgrena zaglađujuća funkcija

Preciznost, brzina i točnost SPH-a ovise o izboru zaglađujuće jezgrene funkcije koja mora zadovoljiti sljedeća svojstva:

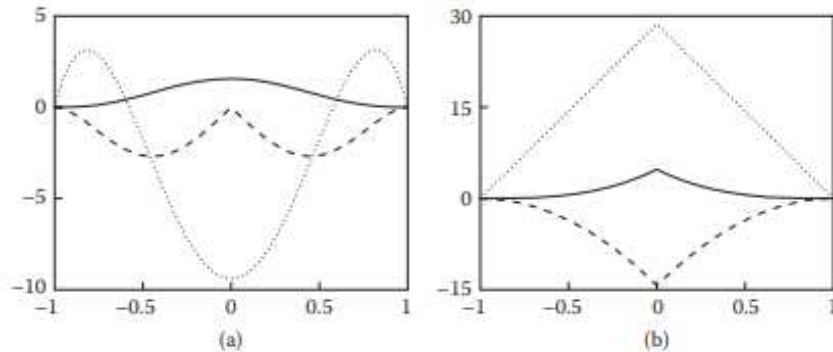
1. Parna funkcija –  $f(x) = f(-x)$
2. Normalizirana –  $\int W(\mathbf{r} - \mathbf{r}', h)dr' = 1$
3. U blizini nule se ponaša kao Diracova delta funkcija –  $\lim_{h \rightarrow 0} W(\mathbf{r} - \mathbf{r}', h) = \delta(\mathbf{r} - \mathbf{r}')$
4. Iznosi nula izvan područja zaglađivanja  $W(\mathbf{r} - \mathbf{r}', h) = 0$  when  $|\mathbf{r} - \mathbf{r}'| > h$
5. Pozitivna vrijednost funkcije –  $W(\mathbf{r} - \mathbf{r}', h) \geq 0$
6. Vrijednost funkcije se mora monotono smanjivati s povećanjem udaljenosti od čestice

U radu su korištene jezgrene funkcije prema radu[1]. Funkcije su navedene u obliku koji se koristi za 3D.

$$W_{poly6} = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3, & 0 \leq r \leq h \\ 0, & r \geq h \end{cases}, \quad (6)$$

$$W_{spiky} = \frac{15}{\pi h^6} \begin{cases} (h - r)^3, & 0 \leq r \leq h \\ 0, & r \geq h \end{cases}, \quad (7)$$

Standardna jezgrena funkcija (engl. standard kernel) i šiljata jezgrena funkcija (engl. spiky kernel) prikazane na slici 4. Puna, isprekidana i točkasta linija redom prikazuju funkciju, njenu prvu i drugu derivaciju.



Slika 4 Jezgrene funkcije

## 2.4. Pseudokod algoritma

SPH se izvodi kroz sljedeća četiri koraka za svaku česticu:

1. Računanje gustoće
2. Računanje sila
3. Integracija
4. Rješavanje sudara

### 2.4.1. Gustoća

Gustoću računamo za svaku česticu uvrštavanjem u osnovnu formulaciju i koristimo standardnu jezgrenu funkciju  $W_{poly6}$ .

$$\rho(\mathbf{r}_i) = \sum_j m_j \frac{\rho_j}{\rho_j} W(\mathbf{r}_i - \mathbf{r}_j, h) = \sum_j m_j W(\mathbf{r}_i - \mathbf{r}_j, h), \quad (8)$$

### 2.4.2. Sile

Prema Navier-Stokesovoj jednadžbi, na fluid djeluju sile uzrokovane razlikom tlaka u fluidu, viskozna sila i vanjske sile.

### 2.4.2.1 Tlak

Prema radu u kojem je predstavljeno korištenje metode SPH za simuliranje fluida, tlak se računa korištenjem jednadžbe stanja plina  $p = k(\rho - \rho_0)$ . 2007. je objavljen rad pod nazivom „Weakly compressible SPH for free surface flows“ u kojem se za računanje tlaka koristi Taitova jednadžba. Korištenjem Taitove jednadžbe dolazi do puno manje varijacije gustoće što nam daje bolje rezultate simulacije i omogućuje veći vremenski korak kod integracije.

$$p = B \left( \left( \frac{\rho}{\rho_0} \right)^\gamma - 1 \right), \quad (9)$$

$$B = \frac{\rho_0 c_s^2}{\gamma}, \quad (10)$$

$\rho_0$  predstavlja ciljanu gustoću,  $\gamma$  je konstanta koja se naziva eksponentom jednadžbe stanja i iznosi 7, a  $c_s$  je brzina zvuka u fluidu.

### 2.4.2.2 Sila uzrokovana razlikom tlaka

Zbog razlike tlaka, fluid će se kretati iz područja višeg tlaka u područje nižeg tlaka pa kod računanja te sile koristimo gradijent tlaka prema jednadžbi 11. U radu se koristi simetrična verzija računanja sile koja zadovoljava 3. Newtonov zakon.

$$F_i^{pressure} = -m^2 \sum_j \left( \frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla W(r_i - r_{j,h}), \quad (11)$$

### 2.4.2.3 Viskozna sila

Viskozna sila je trenje koje se pojavljuje između slojeva fluida zbog razlike u njihovim brzinama.

$$F_i^{viscosity} = \mu m^2 \sum_j \frac{v_j - v_i}{\rho_j} \nabla^2 W(r_i - r_j, h), \quad (12)$$

### 2.4.2.4 Vanjske sile

Vanjske sile se računaju izravno bez korištenja SPH. Primjer je gravitacijska sila koju računamo s  $F_{grav} = mg$ .



### 2.4.3. Integracija

Nakon što se izračunaju sve sile računa se nova brzina i pozicija čestice. Moguće je koristiti bilo koju metodu numeričke integracije, a najčešće se koristi ona najjednostavnija, Eulerova metoda.

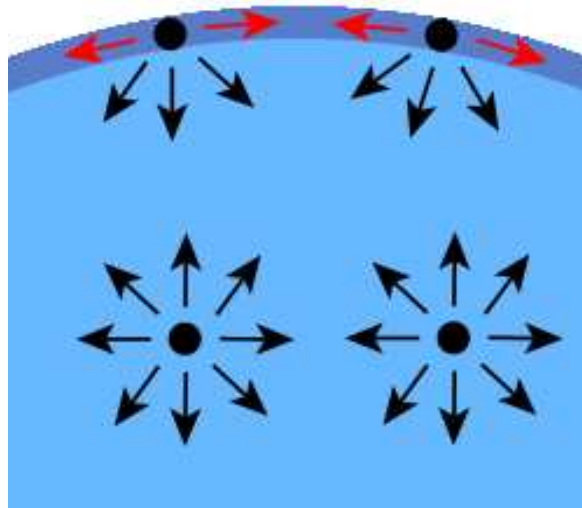
$$a_{n+1} = \frac{F_{total}}{m}, \quad (13)$$

$$v_{n+1} = v_n + a_{n+1}\Delta t, \quad (14)$$

$$x_{n+1} = x_n + v_{n+1}\Delta t, \quad (15)$$

### 2.5. Površinska napetost

Površinska napetost fenomen je kojim opisujemo tendenciju tekućine da minimizira slobodnu površinu. Površinsku napetost uzrokuju kohezijske sile, međumolekularne sile između čestica iste tvari. Svaku molekulu u unutrašnjosti fluida privlače susjedne molekule iz svih smjerova i rezultantna sila iznosi 0. Uz slobodnu površinu molekule nemaju susjede pa rezultantna sila djeluje prema unutrašnjosti fluida (slika 5).



Slika 5 Površinska napetost

Modele za računanje površinske napetosti kod simulacije fluida tehnikom SPH dijelimo u dvije osnovne skupine: modeli koji računaju kohezijske sile i modeli koji koriste izraz za minimizaciju zakrivljenosti površine. Dodavanje samo kohezijskih sila uzrokuje proizvoljne konfiguracije ovisne o početnom stanju. Problem kod drugog tipa modela je stvaranje grupa

čestica, odnosno raspadanje jedne kapljice na nekoliko manjih. U diplomskom radu koristim model opisan u radu „Versatile Surface Tension and Adhesion for SPH Fluids”. Kod ovog modela se ne javljaju prethodni problemi i dobivamo rezultate koji odgovaraju ponašanju tekućina u prirodi. Površinska napetost se sastoji od dvije komponente: kohezije i sile koja minimizira površinu vanjskog ruba fluida. Kohezija je privlačna međumolekularna sila koje djeluje između susjednih čestica i računamo je prema izrazu 16.

$$\mathbf{F}_{cohesion\ i \rightarrow j} = -\gamma m_i m_j C(|\mathbf{r}_i - \mathbf{r}_j|) \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|}, \quad (16)$$

$$C(r) = \frac{32}{\pi h^9} \begin{cases} (h-r)^3 r^3, & 2r > h \cap r \leq h \\ 2(h-r)^3 r^3 - \frac{h^6}{64}, & r > 0 \cap 2r \leq h, \\ 0, & \text{inače} \end{cases} \quad (17)$$

$\gamma$  predstavlja konstantu kojom opisujemo površinsku napetost u ovisnosti o vrsti fluida. Izraz 17 za udaljenosti manje od udaljenosti mirovanja poprima negativne vrijednosti pa sila među molekulama postaje odbojna. Silu koja smanjuje površinu vanjskog ruba računamo prema izrazima 18 i 19.

$$\mathbf{n}_i = t \sum_j \frac{m_j}{\rho_j} \nabla W(|\mathbf{r}_i - \mathbf{r}_j|), \quad (18)$$

$$\mathbf{F}_{curvature\ i \rightarrow j} = -\gamma m_i (\mathbf{n}_i - \mathbf{n}_j), \quad (19)$$

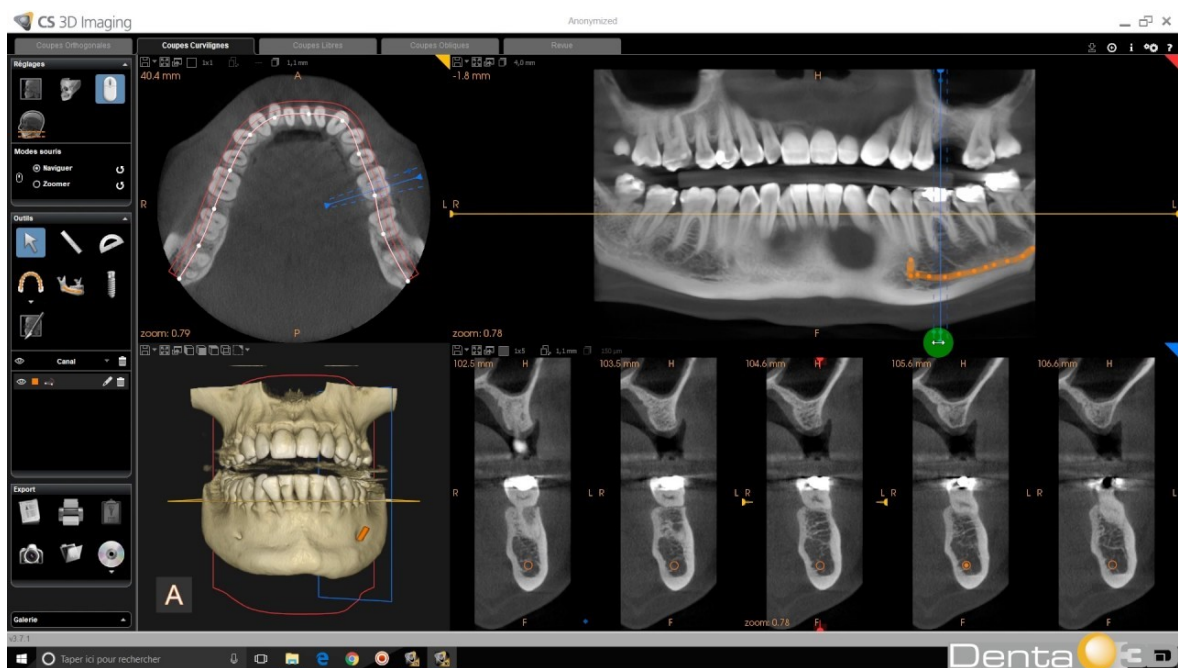
SPH predstavlja fluid pomoću makroskopskih čestica zbog čega dolazi do pojave da se manje čestica lakše razdvaja. Taj problem rješavamo korištenjem simetričnog korekcijskog faktora.

$$K_{ij} = \frac{2\rho_0}{\rho_i + \rho_j}, \quad (20)$$

$$\mathbf{F}_{st\ i \rightarrow j} = K_{ij} (\mathbf{F}_{cohesion\ i \rightarrow j} + \mathbf{F}_{curvature\ i \rightarrow j}), \quad (21)$$

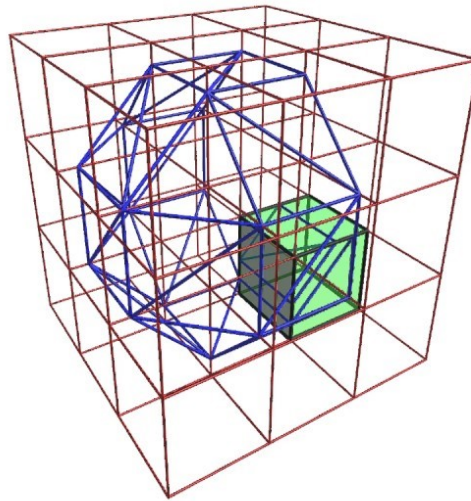
### 3. Algoritam pokretnih kocki (engl. Marching Cubes)

Algoritam pokretnih kocki koristi se za dobivanje izopovršina iz trodimenzionalnog skalarnog polja. Izopovršine se prikazuju mrežom trokuta. Predstavili su ga William Lorensen i Harvey Cline 1987. godine u svom radu za konferenciju o računalnoj grafici SIGGRAPH. Algoritam pokretnih kocki razvijen je za efikasnu vizualizaciju podataka dobivenih korištenjem računalne tomografije i magnetske rezonance (slika 6). Ideja je podijeliti skalarno polje na sitne kocke u čijim vrhovima računamo vrijednost skalarnog polja. Na temelju tih vrijednosti ucrtavamo plohe koje odvajaju vrhove s različitim vrijednostima (slika 7). Zatim je potrebno povezati sve plohe i u svakom vrhu trokuta odrediti normalu koja se koristi za sjenčanje modela.



Slika 6 Primjena pokretnih kocki u dentalnoj medicini

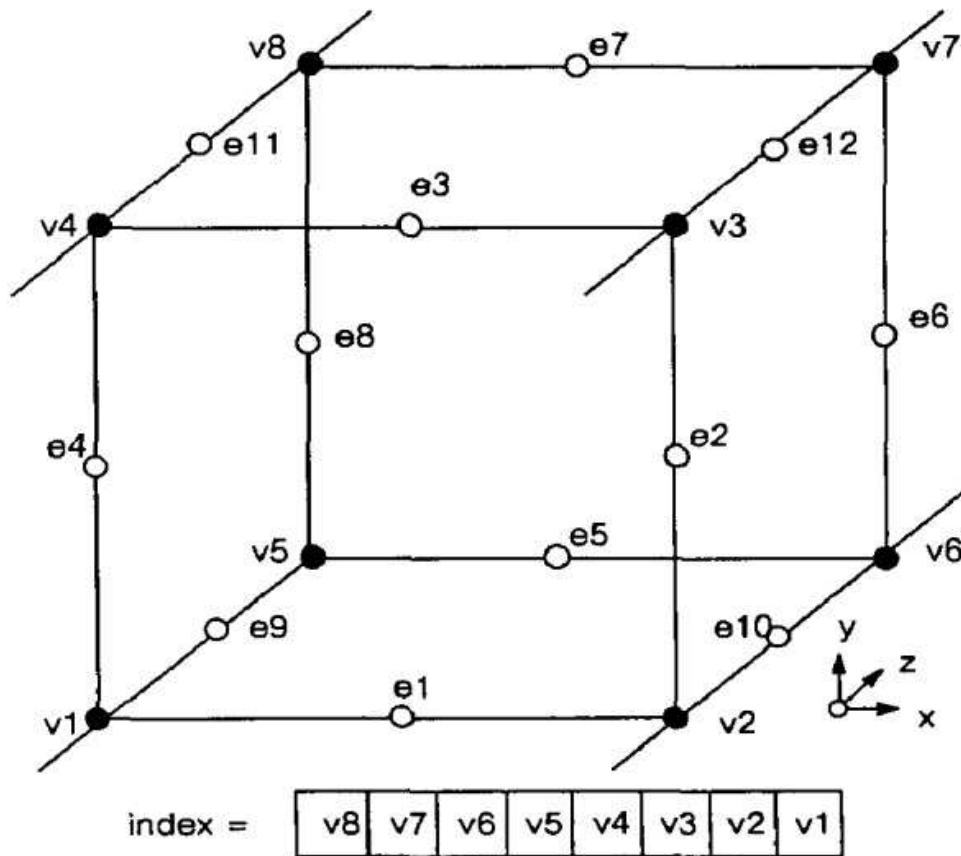
## Implementation



Slika 7 Vizualizacija algoritma pokretnih kocki

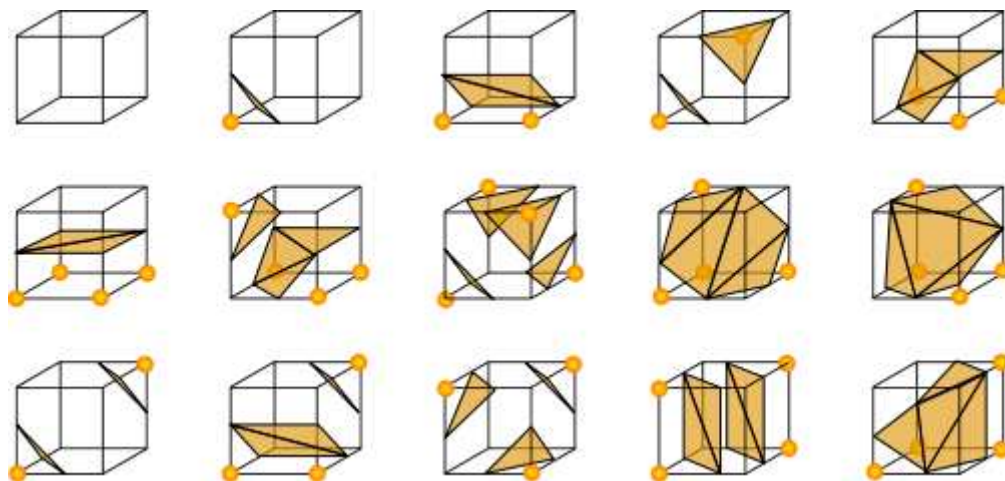
### 3.1. Triangulacija kocke

Algoritam prolazi kroz skalarno polje uzimajući osam susjednih lokacija u svakoj iteraciji. Susjedni vrhovi predstavljaju vrhove imaginarne kocke. Na temelju vrijednosti u vrhovima kocke želimo odrediti kako površina sječe kocku. Ako je vrijednost polja u vrhu kocke veća od ili jednaka izovrijednosti za koju konstruiramo površinu tom vrhu pripisujemo 1, a inače 0. Prolazimo kroz svaki vrh kocke i upisujemo vrijednost 0 ili 1 na odgovarajuću poziciju indeksa. Indeks predstavlja 8-bitni vektor pomoću kojeg određujemo konfiguraciju kocke.



Slika 8 Indeks

Vrijednost 1 predstavlja da se vrh nalazi unutar površine, a 0 izvan. Površina prolazi kroz svaki brid kocke koji se nalazi između vrhova s različitim vrijednostima. S obzirom da kocka ima osam vrhova, a svaki vrh može poprimiti samo dvije vrijednosti, očigledno je da imamo 256 konfiguracija. Koristeći simetrije dolazimo do 15 jedinstvenih konfiguracija koje su prikazane na slici 9.



Slika 9 Konfiguracije kocki

## 3.2. Računanje normala

Potrebno je izračunati vrijednosti normala u svakom vrhu mreže kako bi mogli koristiti Phongov model osvjetljenja. Gradijent je okomit na izopovršinu što znači da je gradijent jednak normalu. Određujemo ga na temelju vrijednosti u susjednim vrhovima i udaljenosti među njima prema sljedećim formulama.

$$n_x = \frac{D(i+1, j, k) - D(i-1, j, k)}{\Delta x}, \quad (22)$$

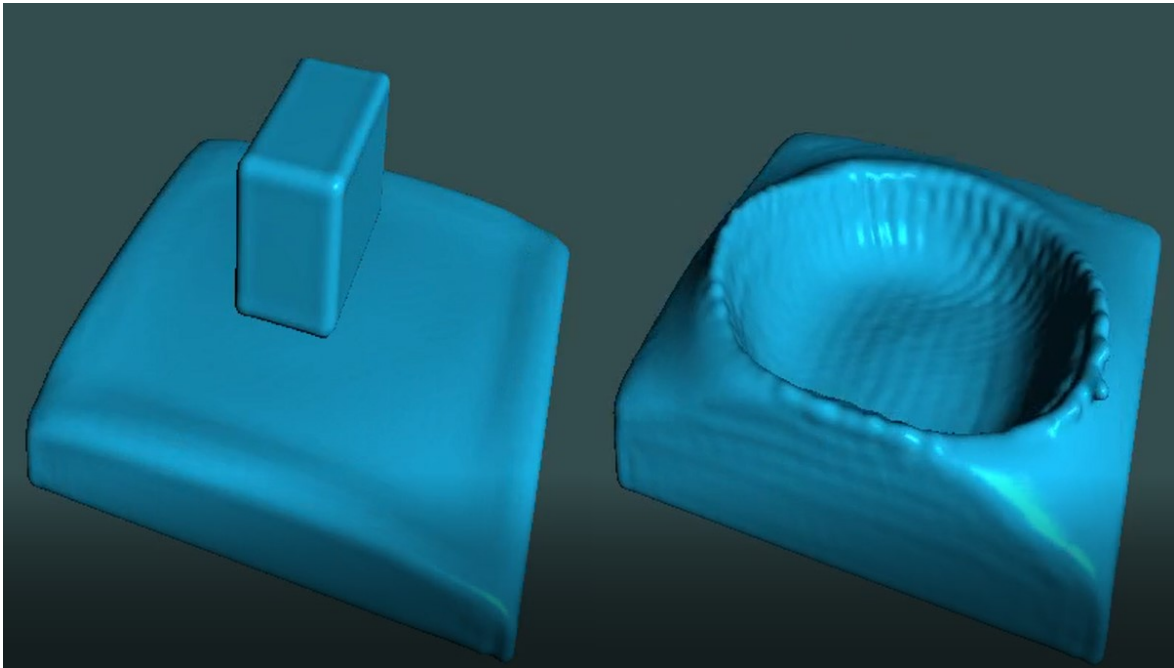
$$n_y = \frac{D(i, j+1, k) - D(i, j-1, k)}{\Delta y}, \quad (23)$$

$$n_z = \frac{D(i, j, k+1) - D(i, j, k-1)}{\Delta z}, \quad (24)$$

Nakon toga gradijent linearno interpoliramo na mjestu presjeka i normaliziramo ga.

## 4. Implementacija

Razvijen je program za simulaciju fluida koji implementira metodu hidrodinamike zaglađujućih čestica i algoritam pokretnih kocki. Korišten je programski jezik c++, aplikacijsko programsko sučelje OpenGL i biblioteke glfw, glew i glm. Glm je matematička biblioteka za grafičke aplikacije koja nam pojednostavljuje rad s vektorima i matricama. Glfw se koristi za upravljanje prozorima, a glew omogućuje korištenje funkcija iz novijih standarda OpenGL-a. Program je pisan unutar razvojnog okruženja Visual Studio. Već kod relativno malog broja čestica imamo prevelik broj operacija za rad u stvarnom vremenu. Zbog toga je rješenje podjeljeno u dva manja projekta kako bi mogli dobiti animaciju fluida takvu da brzina gibanja odgovara brzini gibanja u stvarnosti. Prvi projekt izvršava cijelu simulaciju i sprema rezultate svakog frame-a u file obj formata. Za malo složeniji primjer sa 50000 čestica računanje simulacije od 10 sekundi traje nekoliko sati (slika 10). Drugi projekt se pokreće nakon što smo dobili rezultate simulacije. On učitava fileove u obj formatu i prikazuje rezultate simulacije.



Slika 10 Sph i marching cubes

## 4.1. SPH

U metodi SPH fluid je predstavljen konačnim brojem čestica čije pozicije i brzinu ažuriramo u svakoj iteraciji.

### 4.1.1. Čestica (engl. Particle)

Česticu predstavljamo pomoću klase Particle u kojoj čuvamo podatke potrebne za simulaciju poput brzine, pozicije, sile, gustoće i tlaka.

### 4.1.2. Vremenski korak simulacije

U jednoj sekundi se prikazuje 60 slika na računalu. Maksimalan vremenski korak da bi simulacija bila stabilna je značajno manji od  $\frac{1}{60}$  s i ovisi o trenutnom stanju sustava. Zbog toga jedan korak algoritma dijelimo na nekoliko manjih koraka tako da je zbroj njihovog trajanja točno  $\frac{1}{60}$  s kako je prikazano u isječku koda 1.

```
void SPHSolver::updateOneFrame() {
    double remaingTime = timeStep;

    while (true) {
        double maxTimeStep = nextDt();
        if (maxTimeStep >= remaingTime) {
            update(remaingTime);
            break;
        }
        else {
            update(maxTimeStep);
            remaingTime -= maxTimeStep;
        }
    }
}

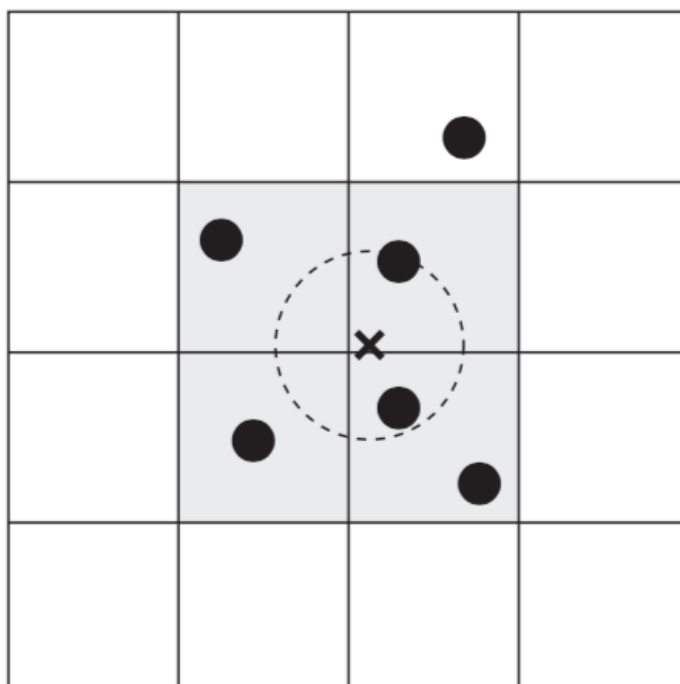
double SPHSolver::nextDt() {
    double dtLimitBySpeed = 0.4 *kernelRadius / speedOfSound;
    double maxForce = 0.0;
    for (auto &p : particles) {
        maxForce = std::max(glm::length(p.force), maxForce);
    }
    double dtLimitByForce = 0.25 * sqrt(kernelRadius * mass / maxForce);
    return std::min(dtLimitBySpeed, dtLimitByForce);
}
```

Isječak koda 1



### 4.1.3. Pronalazak susjednih čestica – ubrzanje algoritma

U svakom koraku za izračun gustoće, tlaka i sila potrebno je izračunati utjecaj susjednih čestica. Vremenski najsloženiji dio algoritma je pronalazak susjeda. Za svaku česticu treba proći kroz sve čestice i odrediti nalaze li se one unutar radijusa zaglađivanja. Asimptotska složenost je  $O(n^2)$ . Traženje susjeda možemo ubrzati tako da postavimo trodimenzionalnu rešetku i svaku česticu upišemo u odgovarajuće polje. Nakon toga za svaku česticu pronalazimo njene susjede tako da provjerimo čestice u susjednom polju. Na slici 11 prikazan je primjer za 2D prostor.



Slika 11 Traženje susjednih čestica u 2D polju

## 4.2. Konverzija podataka iz SPH čestica u trodimenzionalnu rešetku

Za primjenu algoritma pokretnih kocki potrebno je imati vrijednosti na pozicijama koje odgovaraju vrhovima kocke. Vrijednosti u vrhovima trodimenzionalne pravokutne rešetke ovise o gustoći fluida u tim točkama. Ispunjavamo je tako da vrhovi koji se nalaze unutar fluida imaju negativnu vrijednost, a vrhovi izvan fluida pozitivnu. Ispunjavanje rešetke prikazano je u isječku koda 2.

```

void fillGrid(double cutOffDensity, SPHSolver *solver) {
    for (int i = 0; i < values.size(); i++) {
        for (int j = 0; j < values[i].size(); j++) {
            for (int k = 0; k < values[i][j].size(); k++) {
                glm::dvec3 pos = origin + glm::dvec3(i, j, k)*spacing;
                double d = solver->interpolate(pos);
                values[i][j][k] = cutOffDensity - d;
            }
        }
    }
}

```

Isječak koda 2

### 4.3. Algoritam pokretnih kocki

Ulaz u ovaj algoritam je trodimenzionalna rešetka s vrijednostima gustoće, a izlaz je mreža trokuta. Potrebno je za svaku kocku odrediti presjecišta. To radimo tako da odredimo indeks pod nazivom cubeFlag prema isječku koda 3.

```

int cubeFlag = 0;
for (int i = 0; i < 8; i++) {
    if (data[i] < isoValue) {
        cubeFlag = cubeFlag | (1 << i);
    }
}

if (cubeFlag == 0 || cubeFlag == 255) {
    return;
}

```

Isječak koda 3

U programu se nalaze dva polja koje sadrže podatke za svih 256 konfiguracija kocke. Prvo polje sadrži indekse bridova na kojima se nalaze sjecišta, a drugo popis svih trokuta za određenu konfiguraciju. Nakon što se završi izvođenje ovog algoritma upisujemo rezultate u obj file.

### 4.4. Phongov model osvjetljenja

Ovaj model pretpostavlja da se cjelokupno osvjetljenje objekta može opisati kao linearna kombinacija ambijentne, difuzne i zrcalne komponente svjetlosti. Ambijentna komponenta rezultat je interakcije svjetlosti među svim objektima u sceni i kod ovog se modela pretpostavlja da je konstantnog iznosa. Za različite vrste materijala uzimamo različitu konstantu ambijentne refleksije  $k_a$  kojom opisujemo koliko se svjetlosti apsorbira, a koliko

reflektira. Difuzna komponenta dominira kod hrapavih površina i ne ovisi o položaju promatrača. Intenzitet ovisi o kutu pod kojim upada svjetlost u odnosu na normalu te elementarne površine. Zrcalna komponenta ovisi o položaju promatrača i vrsti materijala. Zastupljena je kod glatkih materijala. Implementacija osvjetljenja je riješena unutar fragment shader-a prema isječku koda 4.

```
#version 330 core
layout(location = 0) out vec4 color;

in vec3 fragPos;
in vec3 normal;

uniform vec3 lightPos;
uniform vec3 objectColor;
uniform vec3 lightColor;
uniform vec3 viewPos;

void main()
{
    //ambient
    float ambientStrength = 0.1;
    vec3 ambient = ambientStrength * lightColor;

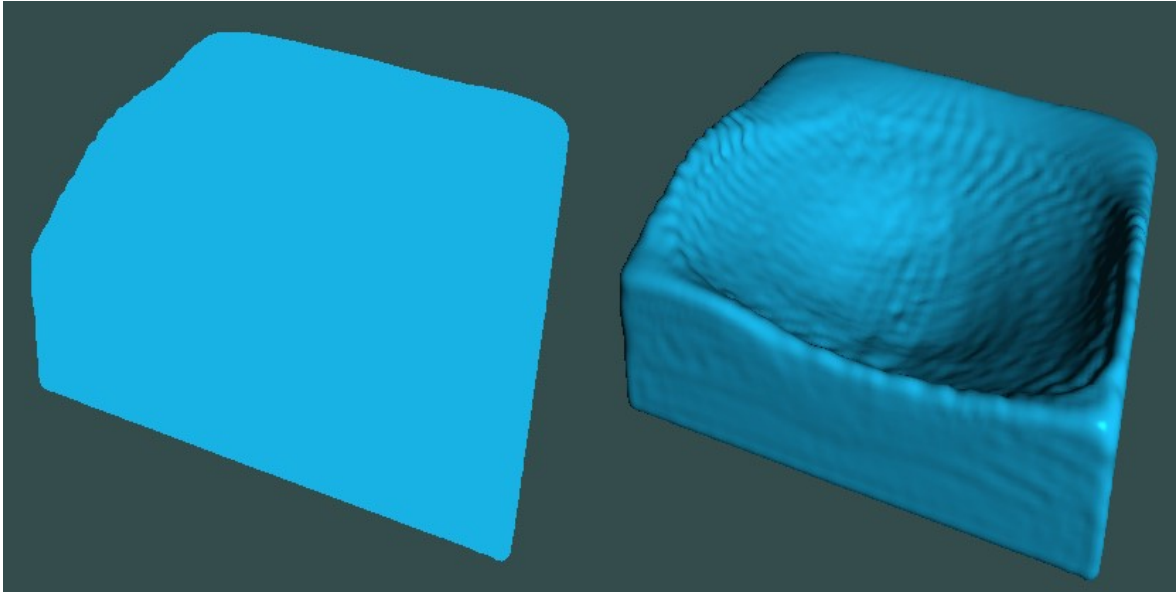
    // diffuse
    vec3 norm = normalize(normal);
    vec3 lightDir = normalize(lightPos - fragPos);
    float diff = max(dot(norm, lightDir), 0.0);
    vec3 diffuse = diff * lightColor;

    // specular
    float specularStrength = 0.5;
    vec3 viewDir = normalize(viewPos - fragPos);
    vec3 reflectDir = reflect(-lightDir, norm);
    float spec = pow(max(dot(viewDir, reflectDir), 0.0), 32);
    vec3 specular = specularStrength * spec * lightColor;

    vec3 result = (ambient + diffuse + specular) * objectColor;
    color = vec4(result, 0.02f);
};
```

#### Isječak koda 4

Osvjetljenje je vrlo bitno jer tek s njim dobijemo osjećaj trodimenzionalnosti što se vidi na slici 11 gdje je dana usporedba istog modela bez osvjetljenja i s osvjetljenjem.



Slika 12 Model fluida bez osvjetljenja i s osvjetljenjem

## 4.5. Pokretanje programa

Prevođenjem dobivamo dvije izvršne datoteke SPHSimulation.exe i SPH3D.exe. Pokrećemo ih iz komandne linije i potrebno je navesti nekoliko argumenata. Izvođenjem SPHSimulation.exe dobivamo rezultate simulacije zapisane u obj fileove. U obj fileovima se nalaze modeli koji prikazuju površinu fluida u različitim trenucima. Svaki model, jedna slika koju prikazujemo je spremljena u jedan obj file. Jedna sekunda animacije je prikazana pomoću 60 slika. Sljedeće argumente koristimo kod pokretanja SPHSimulation.exe:

- -f [ime\_file-a] ili --filename [ime\_file-a] - Naziv fileova u koji se upisuju rezultati. Nazivu se dodaje redni broj modela i ekstenzija .obj. Defaultna vrijednost je mesh.
- -n [broj] ili --number [broj] - Ovaj broj označava koliko ćemo modela dobiti izvršavanjem programa
- -e [redni\_broj] ili --example [redni\_broj] - Redni broj primjera za koji želimo dobiti rezultate izvođenja

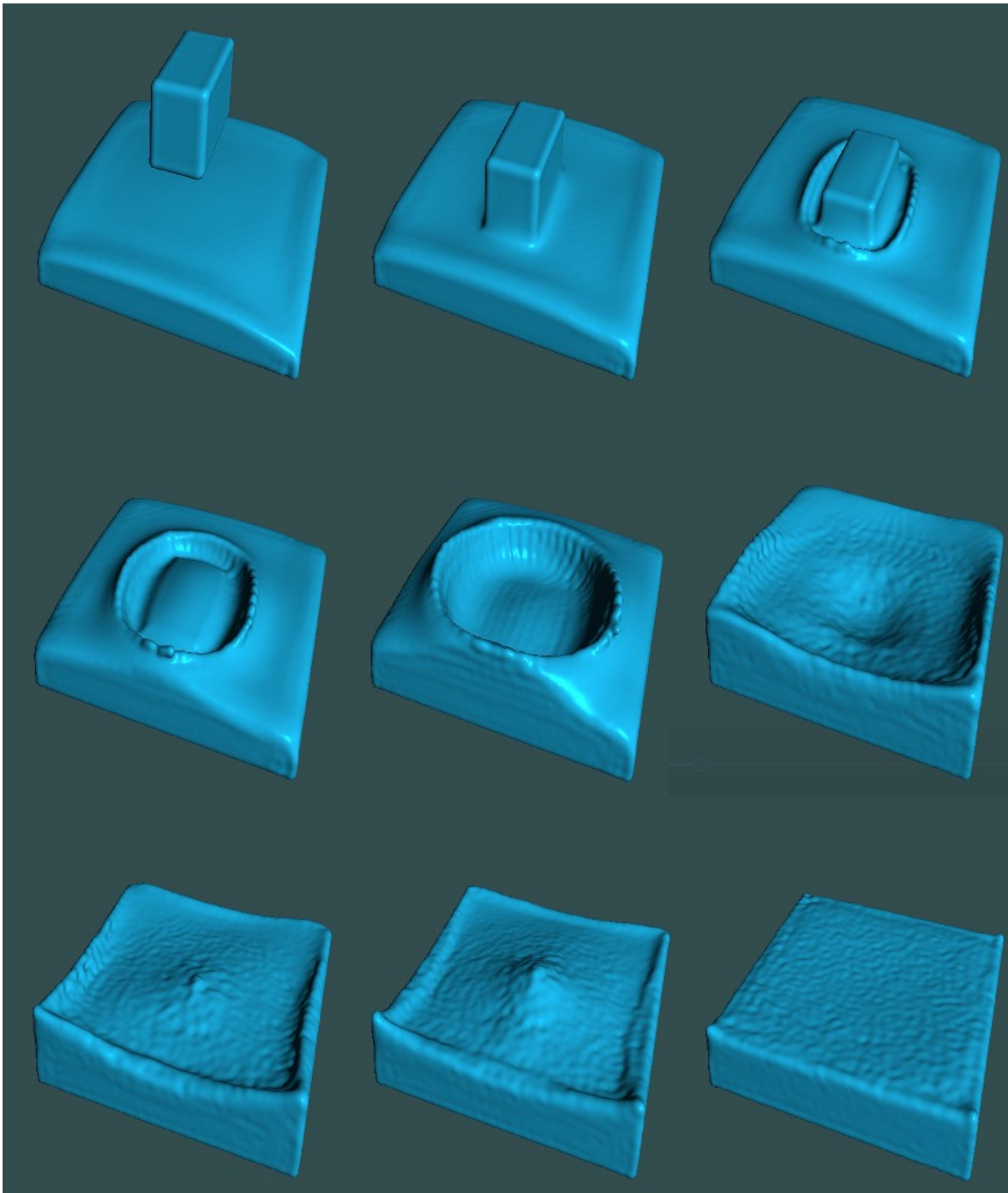
Pokretanjem programa SPH3D.exe na ekranu se pojavljuje animacija koja prikazuje gibanje fluida. Argumenti koje je moguće zadati kod pokretanja:

- -f [putanja] ili --filename [putanja] - Putanja do fileova u kojima se nalaze modeli fluida.
- -n [broj] --number [broj] - Broj modela koji prikazuju površinu fluida

- -p --preload - Dodavanjem ove zastavice se učitavaju se svi objekti u memoriju prije prikaza.

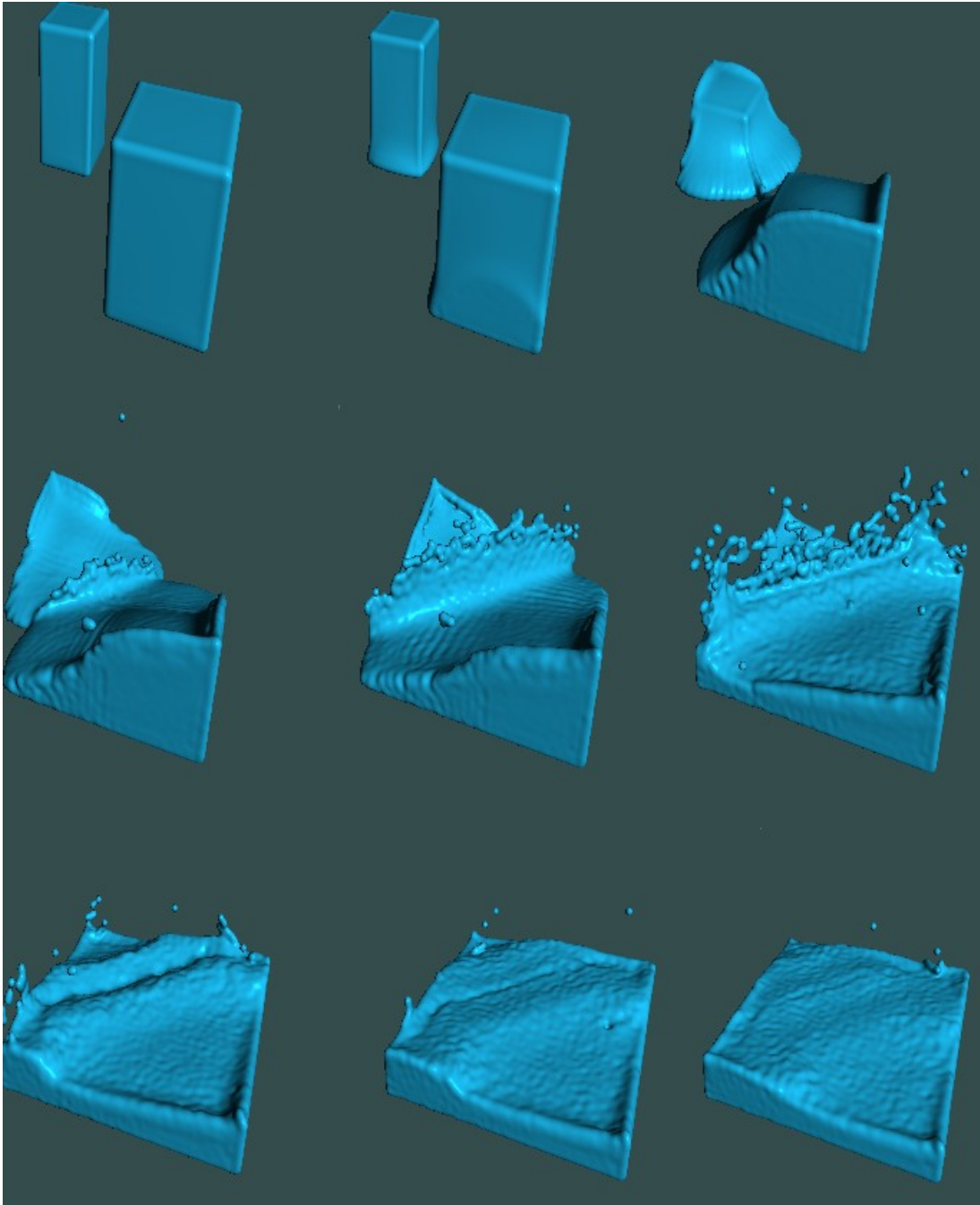
## 4.6. Rezultati

Rezultati su prikazani sekvencom slika koje prikazuju gibanje fluida na nekoliko primjera.

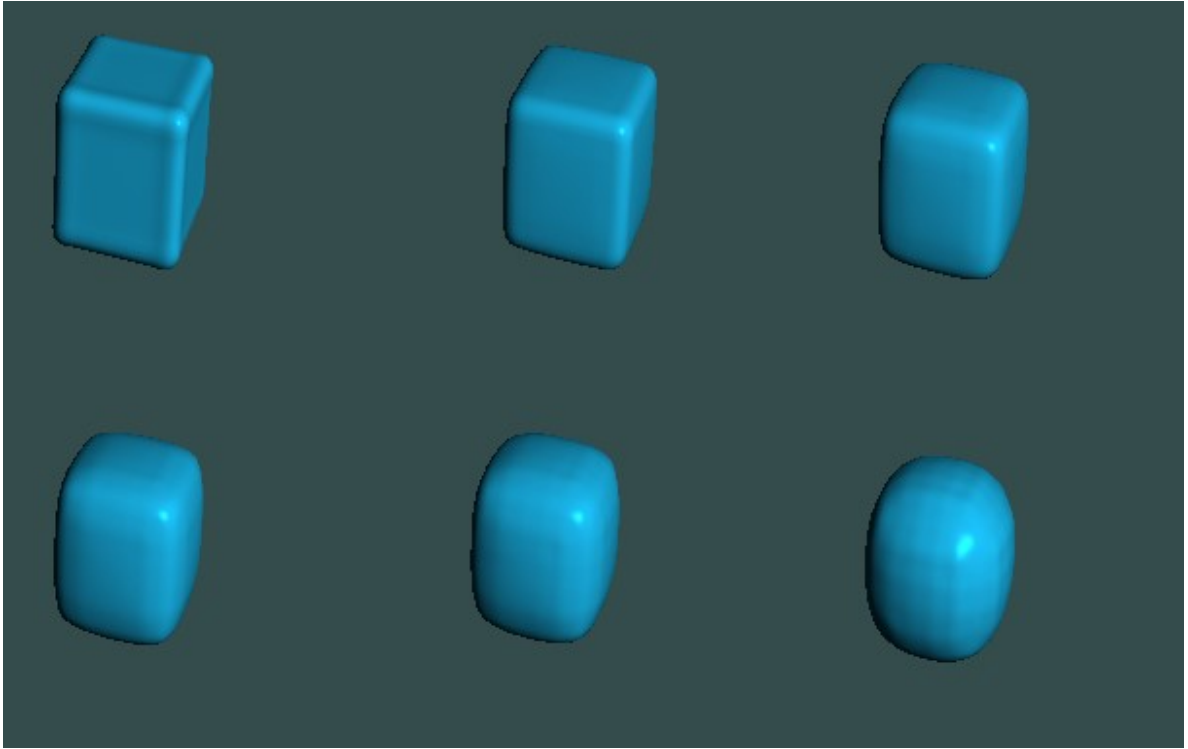


Slika 13 Primjer fluida bez uračunate površinske napetosti

Na slici 13 prikazan je primjer bez površinske napetosti. Slika 15 prikazuje kako se zbog površinske napetosti kocka pretvara u kuglu u području bez gravitacije. Na slici 14 uračunata je površinska napetost, a za viskoznost se koristi manji koeficijent, 0.005 umjesto 0.01. Primjeri su pokretani na računalu Dell Inspiron 5570 s procesorom Intel i7-8850U i 8GB ram-a. Izračun jedne slike u prosjeku traje 120.1s za primjer sa slike 13, 42.2s za primjer sa slike 14 te 5.1 sekunda za primjer sa slike 15.



Slika 14 Primjer fluida s površinskom napetosti



Slika 15 Površinska napetost – transformacija iz kocke u kuglu





## Zaključak

Simulacija fluida je područje računalne grafike s vrlo širokom primjenom od računalnih igara i specijalnih efekata do inženjerskih problema poput toka fluida u cijevima ili kretanja zraka oko krila aviona. Izvođenje tih simulaciju je iznimno složeno i današnja računala nemaju dovoljnu snagu da bi ih mogla izvršavati u stvarnom vremenu. To je jedan od glavnih razloga što se to područje i danas toliko proučava i napreduje. Za razvoj kvalitetnih simulacija potrebno je uz znanja o dinamici fluida i vrlo dobro poznavanje računarstva. Znanja iz računalne grafike nam pomažu da prikaz bude uvjerljiv, a za efikasno pisanje algoritama potrebna su nam znanja o strukturama podataka i paralelizaciji. Gibanje fluida je opisano Navier-Stokesovim jednažbama. Postoje mnoge metode za simulaciju fluida, a razlika među metodama je način na koji one aproksimiraju i rješavaju Navier-Stokesove jednažbe. Površinska napetost djeluje samo na slobodnim površinama pa se ne pojavljuje u Navier-Stokesovim i dodajemo je kao posebnu silu. U ovom radu je opisana metoda hidrodinamike zaglađujućih čestica i algoritam pokretnih kocki za dobivanje površine iz diskretnog skalarnog polja.

## Literatura

- [1] Matthias Muller, David Charypar, Markus Gross, Particle-Based Fluid Simulation for Interactive Applications, SIGGRAPH Symposium on Computer Animation(2003)
- [2] Markus Becker, Matthias Teschner Weakly compressible SPH for free surface flows, 2007
- [3] Nadir Akinici, Gizem Akinici, Matthias Teschner, Versatile Surface Tension and Adhesion for SPH Fluids, 2014
- [4] Doyub Kim, Fluid Engine Development, 2017
- [5] G. R. Liu, M. B. Liu, Smoothed Particle Hydroynamics a meshfree particle method, 2003
- [6] Zdravko Virag, Mario Savar, Ivo Džijan Mehanika fluida I i II: predavanja, 2017.
- [7] <https://software.intel.com/en-us/articles/fluid-simulation-for-video-games-part-1>  
Michael J. Gourlay, datum pristupanja 17.06.2020
- [8] William E. Lorensen, Harvey E. Cline, Marching Cubes: A high resolution 3D surface construction algorithm

# Sažetak

Naslov: Simulacija fluida tehnikom SPH

Ovaj rad se bavi simulacijom nestlačivih fluida. Na početku su opisana osnovna obilježja fluida. Detaljno su opisane metoda hidrodinamike zaglađujućih čestica i algoritam pokretnih kocki. Opisani su modeli površinske napetosti i Phongov model osvjetljenja. Implementirana je 3D simulacija koristeći programski jezik c++ i OpenGL. U radu su opisani ključni dijelovi implementacije. Prikazani su dobiveni rezultati za različite vrijednosti viskoznosti i površinske napetosti.

Ključne riječi: SPH, pokretne kocke, mehanika fluida, hidrodinamika zaglađujućih čestica, Lagrangeov fluid, simulacija

# Summary

Title: Fluid simulation using SPH

This paper presents methods for simulations of incompressible fluids. Firstly, basic fluid properties have been described. Smoothed particle hydrodynamics method and marching cubes algorithm are described in more details. Surface tension models and Phong lighting model are presented. 3D implementation was implemented using the programming language c++ and OpenGL. Key parts of the implementation are described in the paper. Results for different values of viscosity and surface tension are presented.

Keywords: SPH, marching cubes, fluid mechanics, smoothed particle hydrodynamics, Lagrange fluid, simulation