

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINARSKI RAD

HIDRODINAMIKA ZAGLAĐENIH ČESTICA

Leon Stjepan Uroić

Zagreb, siječanj, 2025.

Sadržaj

1. Uvod	2
2. Teorijska podloga	4
2.1. SPH diskretizacija	4
2.2. Diskretizacija diferencijskih operatora	6
2.3. Simulacija slabo stlačivog fluida	7
2.4. Osnovna SPH simulacija slabo stlačivih fluida	8
3. Implementacija SPH simulacije vode	9
3.1. Inicijalizacija	9
3.2. Određivanje susjedstva	10
3.3. Osvježavanje pozicija i brzina čestica	13
3.4. Jezgrene funkcije	14
3.5. Rubovi domene	15
3.6. Konačan paralelan algoritam	16
3.7. Prikaz čestica	16
4. Rezultati	19
5. Zaključak i daljnji razvoj	20
Literatura	21

1. Uvod

Hidrodinamika izglađenih čestica (SPH, *engl. Smoothed Particle Hydrodynamics*) metoda je za simulaciju dinamike fluida i drugih problema u mehanici kontinuma. Izvorno razvijena je krajem 1970-ih za astrofizičke primjene [1]. SPH se od tada razvio u svestrani alat za modeliranje složenih ponašanja fluida. Za razliku od Eulerovih pristupa koji polja u prostoru diskretiziraju pomoću fiksne mreže, SPH je Lagrangeova metoda. Diskretizaciju polja provodi se skupom čestica za koja nose svojstva poput mase, brzine, gustoće i tlaka. Ove čestice međusobno djeluju putem jezgrene funkcije (*engl. kernel function*), što omogućuje aproksimaciju jednadžbi kontinuma, poput Navier-Stokesovih jednadžbi. Priroda SPH-a metode čini ju posebno pogodnom za probleme koji uključuju nepravilne granice, fragmentaciju i ekstremne deformacije, gdje metode temeljene na mreži mogu imati poteškoća.

U računalnoj dinamici fluida (CFD, *engl. Computational Fluid Dynamics*) postoji nekoliko numeričkih metoda, od kojih svaka ima svoje prednosti i ograničenja. Metode temeljene na mreži, poput metode konačnih volumena [2] (FVM, *engl. Finite Volume Method*) i metode konačnih elemenata [2] (FEM, *engl. Finite Element Method*), naširoko se koriste zbog svoje točnosti i učinkovitosti u strukturiranim domenama. Međutim, ove se metode često suočavaju s izazovima u rješavanju problema s pomičnim granicama, slobodnim površinama ili velikim topološkim promjenama. Metode temeljene na česticama, kao što su SPH i polu-implicitne metode pokretnih čestica (MPS, *engl. Moving Particle Semi-implicit*), nude alternativu eliminirajući potrebu za fiksnom mrežom. SPH se posebno ističe zbog svoje jednostavnosti u implementaciji i sposobnosti da prirodno obuhvati fenomene poput prskanja, lomljenja valova i pucanja materijala. Druge metode, poput Lattice Boltzmannove metode [3] (LBM, *engl. Lattice Boltzmann Method*), pružaju drugačiji pristup simuliranjem strujanja fluida na temelju mikroskopskih inte-

rakcija čestica, ali su često ograničene na specifične režime protoka i rubne uvjete.

SPH metoda pronašla je primjenu u širokom rasponu područja, što pokazuje njezinu svestranost i robusnost. U astrofizici, SPH se koristi za modeliranje formiranja galaksija, sudara zvijezda i planetarne dinamike [4]. U inženjerstvu primjenjuje se za simulaciju interakcija fluid i struktura, poput proboga brana [5], pljuskanja u spremnicima [6] i podvodnih eksplozija [7]. SPH se također koristi u biomehanici za proučavanje protoka krvi, deformacije tkiva i simulacije kirurških zahvata [8]. Prilagodljivost SPH metode paralelnom računanju dodatno je proširila njezinu upotrebu u simulacijama velikih razmjera, čineći je moćnim alatom kako za akademska istraživanja tako i za industrijske primjene.

Unatoč svojim prednostima, SPH metoda nije bez izazova. Problemi poput nume-ričke nestabilnosti, implementacije rubnih uvjeta i visokih računalnih troškova i dalje su aktivna područja istraživanja [9].

Ovaj rad ima za cilj istražiti trenutačno stanje SPH, njegove teorijske temelje i praktične primjene, istovremeno naglašavajući tekuće izazove i buduće smjerove istraživanja. Rad prvo iznosi teorijske temelje SPH metode, a potom dokumentira implementaciju jednostavnog SPH modela za simulaciju nestlačivih fluida.

2. Teorijska podloga

U ovom poglavlju navodimo temeljne koncepte hidrodinamike zaglađenih česticama. Poseban naglasak stavljen je na simulaciju slabo stlačivih fluida zato što se implementacija fokusira na simulaciju vode. Velikim dijelom ovo poglavlje prati izvrsne uvode u područje [10], [11] i [12].

2.1. SPH diskretizacija

SPH metoda je primarno metoda diskretizacije polja u prostoru i diferencijskih operatora u prostoru poput gradijenta, divergencije, rotacija... Kao osnovu za navedu diskretizaciju polja koristimo Diracov delta identitet 2.1 On nalaže da konvolucija svake kontinuirane funkcije $A(\mathbf{x})$ s kompaktnim nosačem i $Dirac - \delta$ funkcije identičan je samoj funkciji A , to jest

$$A(\mathbf{r}) = (A * \delta)(\mathbf{r}) = \int A(\mathbf{r}') \delta(\mathbf{r} - \mathbf{r}') d\mathbf{v}' \quad (2.1)$$

gdje je $d\mathbf{v}'$ diferencijalna element volumena. Konačan cilj nam je integral u jednadžbi 2.1 zamijeniti sumom. Pošto je $\delta(\mathbf{x})$ generalizirana funkcija i nije ju moguće diskretizirati, zamijenit ćemo ju kontinuiranom aproksimacijom. Funkciju s kojom aproksimiramo $\delta(\mathbf{x})$ nazivamo jezgrom. Jezgra će dakle generalno biti funkcija $W : \mathbb{R}^d \times \mathbb{R}^+ \rightarrow \mathbb{R}$ koja kao argument prima poziciju u prostoru i radijus djelovanja, a vraća realan broj. Aproksimacijom Dirac delta funkcije više ne vrijedi identitet 2.1 već vrijedi aproksimacija istog

$$A(\mathbf{r}) \approx (A * W)(\mathbf{r}) = \int A(\mathbf{r}') W(\mathbf{r} - \mathbf{r}', h) d\mathbf{v}' \quad (2.2)$$

Navedena jezgrena funkcija W treba ispuniti sljedeće uvjete:

$$\int_{\mathbb{R}^d} W(\mathbf{r}', h) dv' = 1 \quad (\text{uvjet normalizacije})$$

$$\lim_{h' \rightarrow 0} W(\mathbf{r}, h') = \delta(\mathbf{r}) \quad (\text{Dirac-}\delta \text{ uvjet})$$

$$W(\mathbf{r}, h') \geq 0 \quad (\text{pozitivnost})$$

$$W(\mathbf{r}, h') = W(-\mathbf{r}, h') \quad (\text{simetričnost})$$

$$W(\mathbf{r}, h') = 0 \quad \text{za} \quad ||\mathbf{r}|| \geq \hat{h} \quad (\text{kompaktnost nosača})$$

Prva dva uvjeta su nužna kako bi vrijedila aproksimacija 2.2. Uvjet pozitivnosti osigurava nam kasnije izračunate vrijednosti imaju fizikalnog smisla. Neke jezgre ne zadovoljavaju ovaj uvjet, ali onda je moguće dobiti negativne mase, gustoće itd. Uvjet simetričnosti osigurava točnost prvog reda aproksimacije kao što je vidljivo u ?? . Kompaktnost nosača jezgre nije nužna no omogućava značajno efikasniju implementaciju. Ovaj uvjet ustvari govori da svaka čestica djeluje isključivo lokalno na svoje susjede unutar nekog radijusa \hat{h} .

Analizu pogreške koju smo uveli aproksimacijom $Dirac - \delta$ funkcije jezgrom možemo provesti dekompozicijom 2.2 na Taylorov red

$$(A * W)(\mathbf{r}) = \int \left[A(\mathbf{r}) + \nabla A|_{\mathbf{r}} \cdot (\mathbf{r}' - \mathbf{r}) + \frac{1}{2} (\mathbf{r}' - \mathbf{r}) \cdot \nabla \nabla A|_{\mathbf{r}} (\mathbf{r}' - \mathbf{r}) + \mathcal{O}((\mathbf{r}' - \mathbf{r})^3) \right] W(\mathbf{r} - \mathbf{r}', h) dv'$$

$$= A(\mathbf{r}) \cdot \int W(\mathbf{r} - \mathbf{r}', h) dv' + \nabla A|_{\mathbf{r}} \cdot \int (\mathbf{r} - \mathbf{r}') W(\mathbf{r} - \mathbf{r}') dv' + \mathcal{O}((\mathbf{r} - \mathbf{r}')^2)$$

Sada je lako uočiti da je 2.2 aproksimacija prvog reda ako je integral jezgre jednak jedan što nam upravo osigurava uvjet normalizacije. Štoviše to je aproksimacija drugog reda ako je jezgra simetrična što smo opet naveli kao uvjet za jezgrenu funkciju.

Preostalo nam je još diskretizirati formulu 2.2 zamjenom integrala sumom po dis-

kretnim uzorcima.

$$A(\mathbf{r}) \approx (A * W)(\mathbf{r}) = \int \frac{A(\mathbf{r}')}{\rho(\mathbf{r}')} W(\mathbf{r} - \mathbf{r}', h) \rho(\mathbf{r}') d\mathbf{r}' \approx \sum_{j \in \mathcal{F}} A_j \frac{m_j}{\rho_j} W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (2.3)$$

Navedenom diskretizacijom nažalost gubimo garanciju točnosti nultog reda. U praksi ovo nije preveliki problem. Ako baš želimo, moguće je vratiti točnost nultog normalizacijom pa čak i prvog reda rješavanjem linearne sustava. Više o ovoj temi moguće je pronaći u [11].

2.2. Diskretizacija diferencijskih operatora

Gradijent polja lako je izvesti iz formule 2.3

$$\nabla A_i \approx \sum_{j \in \mathcal{F}} A_j \frac{m_j}{\rho_j} \nabla W_{ij} \quad (2.4)$$

gdje je $W_{ij} = W(\mathbf{r}_i - \mathbf{r}_j, h)$. Razvojem direktno izvedenog gradijent Taylorovim redom pokazuje se da u općem slučaju ovako dobiven gradijent ne posjeduje niti točnost nultog reda. Kako bi to popravili potrebno je oduzeti prvi izraz greške iz Taylorovog reda. Rezultat te razlike jednak je

$$\nabla A_i \approx \sum_{j \in \mathcal{F}} \frac{m_j}{\rho_j} (A_j - A_i) \nabla W_{ij} \quad (2.5)$$

. Uz gore navedenu formulu predložena je još simetrična formula [13] za izračun gradijenta

$$\nabla A_i \approx \rho_i \sum_{j \in \mathcal{F}} m_j \left(\frac{A_i}{\rho_i^2} + \frac{A_j}{\rho_j^2} \right) \nabla W_{ij} \quad (2.6)$$

. Iako je nije točna do prvog reda, velika prednost je što očuva linearu i kutnu količinu gibanja što je važan kriterij za robustnost simulacije.

Na sličan način diskretizaciju Laplaceovog operatora možemo izravno dobiti iz 2.3

$$\nabla^2 A_i \approx \sum_{j \in \mathcal{F}} A_j \frac{m_j}{\rho_j} \nabla^2 W_{ij} \quad (2.7)$$

. Opet ovakav izvod loše aproksimira Laplaceov operator. Poboljšana verzija prezenti-

rana je u [14], a ovdje ju navodimo u jednadžbi 2.8 .

$$\nabla^2 A_i \approx - \sum_{j \in \mathcal{F}} \frac{m_j}{\rho_j} (A_j - A_i) \frac{2||\nabla W_{ij}||}{||\mathbf{r}_i - \mathbf{r}_j||} \quad (2.8)$$

2.3. Simulacija slabo stlačivog fluida

Jednadžba kontinuiteta 2.9 opisuje evoluciju gustoće objekta kroz vrijeme.

$$\frac{D\rho}{Dt} = -\rho(\nabla \cdot \mathbf{u}) \quad (2.9)$$

Izraz $\frac{D(\cdot)}{Dt}$ označava materijalnu derivaciju. U Lagrangeovom koordinatnom sustavu koji koristimo u ostatku rada, materijalna derivacije jednostavno je $\frac{\delta(\cdot)}{\delta t}$. Za nestlačive fluide izraz 2.9 daje nam ograničenje 2.10 .

$$\frac{D\rho}{Dt} = 0 \iff \nabla \cdot \mathbf{u} \quad (2.10)$$

O kretanju nestlačivog fluida govore nam poznata Navier-Stokes jednadžba 2.11 .

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f}_{ext} \quad (2.11)$$

Ukratko Navier-Stokes jednadžbu moguće je izvesti iz zakona očuvanja linearne količine gibanja koji je generalizacija drugog Newtonovog zakona. Lijeva strana jednadžbe jednak je umnošku gustoće s materijalnom derivacijom brzine po vremenu. Desna strana predstavlja zbroj sila koje djeluju na svaku materijalnu česticu. Prvi pribrojnik je sila uzrokovana pritiskom, drugi je sila viskoznosti, a treći vanjska sila koja djeluje na svaku česticu.

Ako želimo strogo održati nestlačivost fluida, pritisak p možemo shvatiti kao Lagrangeov multiplikator koji određujemo tako da je 2.10 zadovoljena. Ako stroga nestlačivost nije nužna, pritisak možemo izračunati pomoću tako zvane *jednadžbe stanja*. Jedan često korišten primjer je varijacija zakona idealnog plina

$$p(\rho) = k(\rho - \rho_0) \quad (2.12)$$

gdje je ρ_o gustoća fluida u mirovanju.

2.4. Osnovna SPH simulacija slabo stlačivih fluida

Napokon ovdje izvodimo jednostavan model za simulaciju slabo stlačivih fluida SPH metodom. Prvi korak bit će izračun gustoće svake čestice. Formula za izračun gustoće sada trivijalno slijedi iz 2.3 .

$$\rho_i \approx \sum_{j \in \mathcal{F}} \rho_j \frac{m_j}{\rho_j} W(\mathbf{r}_i - \mathbf{r}_j, h) = \sum_{j \in \mathcal{F}} m_j W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (2.13)$$

Kada izračunamo gustoću, tlak čestice lako je dobiti iz jednadžbe stanja 2.12 Silu koju uzrokuje tlak na svaku materijalnu česticu tada dobivamo trivijalno iz simetrične formule za izračun gradijenta polja 2.6 .

$$\mathbf{f}_p \approx \rho_i \sum_{j \in \mathcal{F}} m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \nabla W_{ij} \quad (2.14)$$

Silu viskoznosti na isti način dobijemo iz 2.8 .

$$\mathbf{f}_v \approx -\mu \sum_{j \in \mathcal{F}} \frac{m_j}{\rho_j} (\mathbf{u}_j - \mathbf{u}_i) \frac{2||\nabla W_{ij}||}{||\mathbf{r}_i - \mathbf{r}_j||} \quad (2.15)$$

Na kraju iznosimo pseudokod koji iznosi generalne korake koje je potrebno izvesti za svaki korak simulacije.

Algoritam 1: Osnovna SPH simulacija slabo stlačivog fluida

za svaku česticu i čini

izračunaj gustoću ρ_i prema 2.13

kraj

za svaku česticu i čini

izračunaj silu tlaka f_i^p prema 2.14 koristeći 2.12

izračunaj silu viskoznosti f_i^v prema 2.15

$\mathbf{f}_i^{ext} \leftarrow G\rho_i$

$\mathbf{F}_i \leftarrow \mathbf{f}_i^p + \mathbf{f}_i^v + \mathbf{f}_i^{ext}$

kraj

za svaku česticu i čini

integriraj promjenu brzine $\frac{D\mathbf{u}_i}{Dt} = \frac{\mathbf{F}_i}{\rho_i}$

integriraj promjenu pozicije čestice $\frac{D\mathbf{r}_i}{Dt} = \mathbf{u}_i$

kraj

3. Implementacija SPH simulacije vode

U ovom poglavlju dokumentirana je jednostavna implementacija SPH modela za simulaciju vode u stvarnom vremenu. Navedena implementacija napisana je u Rust programskom jeziku i WebGpu aplikacijskom programskom sučelju. Kako bi se simulacija vode u tri prostorne dimenzije izvršavala u stvarnom vremenu, cijeli postupak je napisan u sjenčarima za računanje.

Sam programski kod dostupan je u repozitoriju <https://github.com/leon3428/sploosh>. Ovo poglavlje prati navedeni programski kod gotovo identičnim redoslijedom kojim se izvodi u svakom koraku simulacije.

3.1. Inicijalizacija

Čestice su na početku simulacije raspoređene u obliku kocke kao što je prikazano na slici 3.1.. Veličina te kocke određena je brojem čestica i radijusom zaglađivanja h . Čestice unutar kocke želimo pravilno rasporediti tako da je gustoća svih čestica približno jednak ciljnoj gustoći. Kada to ne bi bio slučaj, čestice u simulaciji bi prilikom pokretanja simulacije "eksplodirale". Iako bi možda intuitivno čestice postavili tako da su razmaknute za radius zaglađivanja kako ne bi međusobno djelovale, ovakav pristup rezultirat će izrazito ne uniformnim poljem gustoće. Čestice su zato razmagnute za konstantu sažimanje $s = 0.5$ radijusa gustoće. Ovaj odabir je donekle proizvoljan kao i većina parametara simulacije. Na poziciju svake čestice dodaje se još uniformno distribuiran šum $n \sim \mathcal{U}(-\frac{h}{12}, \frac{h}{12})$ na svaku komponentu. Bez šuma simulacija ne djeluje realistično zbog pretjerano jednolike raspodjele čestica nakon kolizije s rubom domene što nije nešto što opažamo u stvarnom svijetu. Ovdje je još važno primijetiti da broj čestica i radius zagla-

đivanja definiraju volumen fluida koji će biti prisutan u simulaciji. Taj volumen iznosi

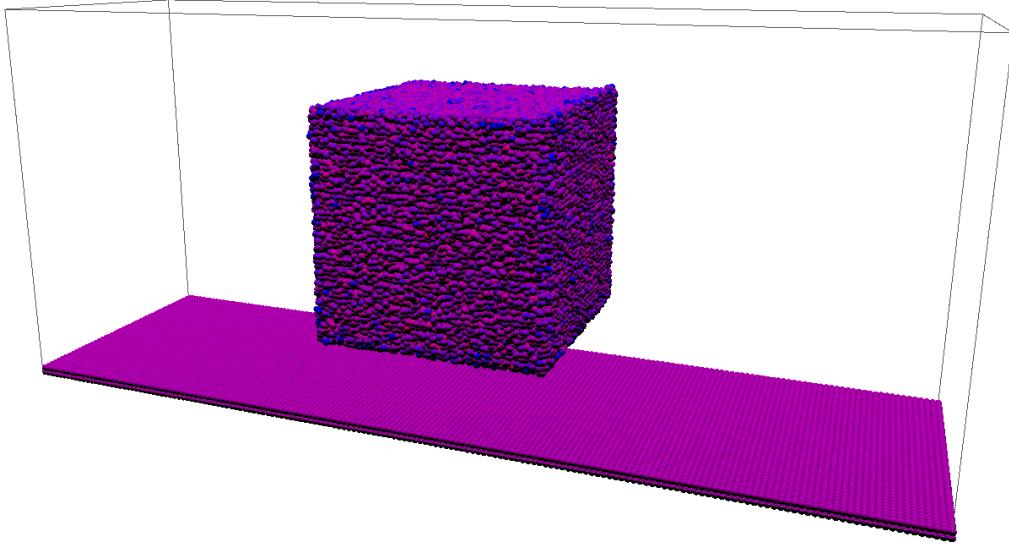
$$V = (sh\sqrt[3]{N})^3 = s^3h^3N \quad (3.1)$$

gdje je N broj čestica, a s konstanta sažimanja.

Ciljna gustoća opet je proizvoljno postavljena. Prilikom izbora važno je samo voditi računa da vrijednosti na koje utječe ciljna gustoća padnu u dobar raspon vrijednosti za prikaz s brojevima s pomicnim zarezom. Dobra i često korištena vrijednost je $\rho_0 = 200$.

Masu čestica sada je moguće odrediti provođenjem postupka izračuna gustoće opisanog u 4 za proizvoljnu masu. Masu zatim normaliziramo tako da je gustoća čestica na početku jednaka ciljnoj. Masu je moguće i jednostavnije postaviti jer znamo ciljnu gustoću i volumen fluida. Ova metoda nije toliko precizna kao prethodna, ali je jednostavnija u implementacijskom smislu i pokazala se kao dovoljno dobra u praksi.

Početna brzina svih čestica jednaka je nuli.



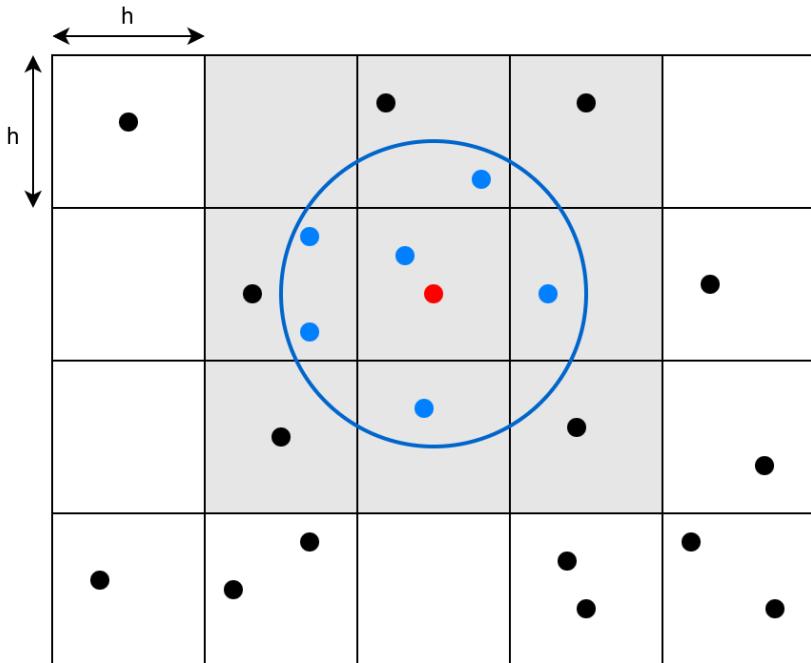
Slika 3.1. Prikaz čestica nakon inicijalizacije. Boja čestice predstavlja gustoću gdje je crvenom bojom označena veća gustoća od ciljne, a plavom manja. Svrha čestica na dnu domene opisana je u 3.5. .

3.2. Određivanje susjedstva

Uvjet kompaktnosti nosača jezgrene funkcije naveden u uvjet normalizacije osigurava da čestice djeluju samo na svoje lokalno susjedstvo i da na njih same djeluje smo su-

sjedne čestice. Efikasno određivanje susjedstva svake čestice glavna je prepreka u efikasnoj implementaciji SPH simulacije. Trivijalan algoritam sa složenošću $\mathcal{O}(N^2)$ nije nam prihvatljiv.

Pronalazak susjeda ubrzat ćeemo korištenjem strukture podataka za prostornu pretragu preuzetu iz [15]. Svaka čestica djeluje na čestice unutar radiusa zaglađivanja. Ako domenu podijelimo na čelije s dužinom stranice h , susjadi svake čestice mogu se nalaziti samo u istoj ili susjednim čelijama kao što je prikazano na slici 3.2.. Prilikom pronalaska susjeda više neće biti potrebno ispitati sve čestice nego samo one u susjednim čelijama. Dakle struktura podataka treba efikasno podržavati upite tipa koje se čestice nalaze u određenoj čeliji.



Slika 3.2. Podjela 2D domene na čelije. Čestica čije susjede pronalazimo označena je crvenom bojom, a njeni susjedi označeni su plavom bojom. Čelije koje ispitujemo osjenčane su sivom bojom.

Pseudokod paralelnog algoritma konstrukcije strukture podataka za prostornu pretragu prikazan je u 2. Algoritam će izgraditi tri niza: niz ključeva K , niz vrijednost V i niz indeks I . Prvi korak je popunjavanje nizova ključeva i vrijednosti. Za svaku česticu zapisujemo indeks čelije u kojem se čestica nalazi u niz ključeva. Taj indeks čelije možemo dobiti koristeći razne funkcijama raspršenog adresiranja. Konkretno u trenutnoj implementaciji indeks čelije računa se kao $c_{ind} = c_z + c_y n_z + c_x n_y n_z$ gdje su c_x, c_y i c_z redni brojevi čelije, a n_x, n_y i n_z broj čelija po svakoj dimenziji. U niz vrijednosti upisujemo

indeks čestice. Ova dva niza u sljedećem koraku sortiramo uzlazno prema ključevima. Time smo sve čestice koje pripadaju istoj čeliji smjestili u jedan kontinuiran blok u nizovima. Za efikasnu pretragu potrebno je još odrediti gdje počinje blok koji odgovara svakoj čeliji. To obavlja treći korak algoritma koji će u niz I upisati na mjesto svakog indeksa čelije od koje pozicije u nizovima ključeva i vrijednosti se nalaze čestice u toj čeliji. Ukupna složenost algoritma na CRCW PRAM-u uz pretpostavku da koristimo paralelni radix sort kao npr. [16] iznosi $\mathcal{O}(w\log(N))$ gdje je w broj bitova ključa, a N broj čestica.

Algoritam 2: Paralelan algoritam konstrukcije strukture podataka za prostornu pretragu

```

 $K \leftarrow []$ 
 $V \leftarrow []$ 
 $I \leftarrow []$ 
paralelno za  $i$  od 0 do  $N$  čini
     $c \leftarrow \text{dohvati_ćeliju}(r_i)$ 
     $K[i] \leftarrow \text{indeks_ćelije}(c)$ 
     $V[i] \leftarrow i$ 
kraj
paralelni_sort(K, V)
paralelno za  $i$  od 1 do  $N$  čini
    ako  $K[i] \neq K[i - 1]$  onda
         $I[K[i]] \leftarrow i$ 
    kraj
kraj

```

Susjede svake čestice sada možemo efikasno pronaći uporabom algoritma 3 . Algoritam će prvo izračunati redni broj čelije po svakoj koordinatnoj osi za česticu i s pozicijom r_i . Nakon toga prolazit će po svim susjednim čelijama, njih ukupno dvadeset i sedam. Za svaku susjednu čeliju algoritam provjerava od koje pozicije se čestice u toj čeliji nalaze u nizu ključeva. To čini pomoću prethodno konstruiranog niza I . U konačnoj petlji prolazi se po svim česticama od tog indeksa sve dok indeks čelije na trenutnoj poziciji nije različit od indeksa susjedne čelije. Različita vrijednost označava da smo prošli sve čestice u toj čeliji. Kako bi se uvjerili da je neka čestica stvarno susjed, provodimo još jednu završnu provjeru udaljenosti.

Algoritam 3: Dohvaćanje susjeda pomoću strukture podataka za prostornu pretragu

```

Ulaz:  $\mathbf{r}_i, P[], K[], V[], I[]$ 
 $\mathbf{c} \leftarrow \text{dohvati_ćeliju}(\mathbf{r}_i)$ 
za  $i$  od -1 do 1 čini
    za  $j$  od -1 do 1 čini
        za  $k$  od -1 do 1 čini
             $\mathbf{c}_n \leftarrow \mathbf{c} + [ijk]^T$ 
            ako je  $\mathbf{c}_n$  izvan domene onda
                | nastavi
                kraj
                 $ind_n \leftarrow \text{indeks_ćelije}(\mathbf{c}_n)$ 
                 $l \leftarrow I[ind_n]$ 
                dok  $K[l] = ind_n$  čini
                    |  $l \leftarrow l + 1$ 
                    |  $ind \leftarrow I[l]$ 
                    | ako  $\|P[ind] - \mathbf{r}_i\| \leq h$  onda
                        | | /* čestica ind je susjed
                        | | kraj
                        | kraj
                    | kraj
                | kraj
            | kraj
        | kraj
    | kraj
kraj

```

3.3. Osvježavanje pozicija i brzina čestica

Jednom kada izračunamo ukupnu silu koja djeluje na svaku česticu F_i , potrebno je tu silu i primijeniti. Prema Navier-Stokes jednadžbi 2.11 materijalna derivacija brzine jednaka je ukupnoj sili koja djeluje na česticu podijeljenu s gustoćom. Ono što nas zapravo zanima je pozicija čestice u svakom trenutku. Tu poziciju izračunat ćemo numeričkom integracijom materijalne derivacije brzine. Tipični odabiri su semi-implicitna Eulerova metoda i metoda žabljeg skoka. Među njima u implementaciji se koristi metoda žabljeg skoka zbog svoje veće preciznosti i neznačajno veće implementacijske složenosti. Postupak integracije opisan je s

$$\mathbf{u}_{t+\frac{1}{2}} = \mathbf{u}_{t-\frac{1}{2}} + \mathbf{a}_t \Delta t \quad (3.2)$$

$$\mathbf{r}_{t+1} = \mathbf{r}_t + \mathbf{u}_{t+\frac{1}{2}} \Delta t \quad (3.3)$$

.

Maksimalan vremenski korak za koji je integracija konvergira moguće je procijeniti Courant-Friedrichs-Lowy uvjetom navedenim u 3.4 .

$$\Delta t \leq \lambda \frac{\hat{h}}{\| \mathbf{u}_{max} \|} \quad (3.4)$$

Prema toj heuristici gornja granica koraka integracije proporcionalna je omjerom radijusa zaglađivanja i brzine čestice s najvećom brzinom. Parametar λ heuristički se tipično postavlja na vrijednost $\lambda \approx 0.4$. Ovaj uvjet može se iskoristiti i za dinamičnu prilagodbu koraka integracije tijekom izvođenja makar se on trenutno ne koristi u implementaciji zato što se simulacija izvodi u stvarnom vremenu.

3.4. Jezgrene funkcije

Na stabilnost, točnost i brzinu SPH metode uvelike utječe odabir jezgrena funkcije. Jezgrene funkcije korištene u implementaciji preuzete su iz rada [17] . Za izračun gustoće koristi se poly6 jezgrena funkcija 3.5 .

$$W_{poly6}(r, h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3 & 0 \leq r \leq h \\ 0 & \text{inače} \end{cases} \quad (3.5)$$

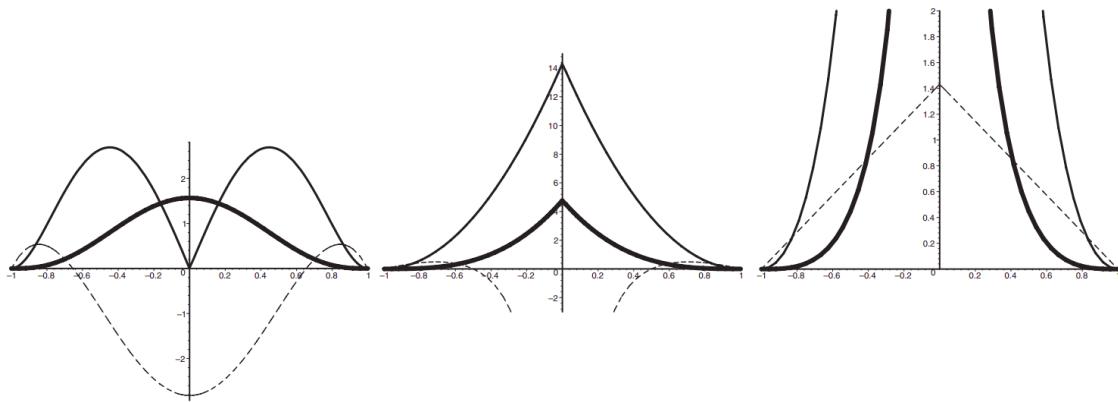
Kada bi ovu jezgenu funkciju koristili i za izračun sile tlaka, čestice koje dođu dovoljno blizu jedna drugoj grupirale bi se u klastere. Specifično, problem je gradijent jezgrene funkcije koji teži ka nuli kako se udaljenost smanjuje. Zato za tu svrhu koristimo jezgranu funkciju 3.6 .

$$W_{spiky}(r, h) = \frac{15}{\pi h^6} \begin{cases} (h - r)^3 & 0 \leq r \leq h \\ 0 & \text{inače} \end{cases} \quad (3.6)$$

Viskoznost je fenomen koji nastaje zbog trenja između čestica. Zato bi viskoznost trebala zaglađivati polje brzine. Poly6 jezgra nema ovo svojstvo jer Laplacian polja brzine za dvije čestice koje su dovoljno blizu postat će negativan. Negativan Laplacian rezultirat će povećanju relativne brzine između čestica što je upravo suprotno od onog što želimo. Zbog navedenog razloga za izračun sile viskoznosti koristi se treća jezgrena funkciju 3.7

$$W_{visc}(r, h) = \frac{15}{2\pi h^3} \begin{cases} -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1 & 0 \leq r \leq h \\ 0 & \text{inače} \end{cases} \quad (3.7)$$

Izgled svih navedenih jezgrenih funkcija kao i njihovih gradijenata i Laplaciana vidljiv je na slici 3.3. .



Slika 3.3. S lijeva na desno: poly6, spiky, viskoznost jezgrena funkcija. Debela linija predstavlja smu jezgrena funkciju, tanka gradijent, a iscrtkana Laplacian.

3.5. Rubovi domene

Kako čestice ne bi izašle iz simulacijske domene, nakon osvježavanja pozicije čestica provodi se provjera. Ako se čestica nalazi izvan, ona se vraća na granicu i vektor njene brzine reflektira se s obzirom na normalu granice domene. Ovime nismo još riješili sve probleme koji nastaju uz granicu. Ako promotrimo dno domene, gdje je ovaj problem posebno naglašen, uočit ćemo da se tamo gomila velik broj čestica. Problem nastaje upravo zbog odbijanja. Čestice se na dnu odbiju od ruba domene, ali nemaju kuda otici zbog čestica iznad njih. Tada se ponovno reflektiraju prema dnu domene itd. Kroz par koraka ovaj postupak svesti će komponentu brzine u smjeru normale granice domene na nulu. Takve čestice tada trajno ostaju na granici.

Jedna od metoda za suočavanje s ovi problemom zasniva se na dodatnim neaktivnim česticama. Neaktivne čestice u procesu inicijalizacije rasporedimo u pravilnu mrežu s nekoliko slojeva kao što je prikazano na 3.1.. U implementaciji korištena su dva slojeva. Tim česticama dodijelimo ciljnu gustoću fluida bez obzira na okolne čestice. Tu gustoću ne mijenjamo za vrijeme trajanja simulacije, ali neaktivne čestice i dalje sudjeluju i ra-

čunanjem gustoća i sila drugih čestica. Poziciju neaktivnih čestica također ne mijenjamo. Ti slojevi neaktivnih čestica zapravo postavljaju idealne uvjete na rubu domene i ne dopuštaju aktivnim česticama da imaju preveliki doticaj s rubom domene.

3.6. Konačan paralelan algoritam

Algoritam 4: Konačan paralelan algoritam za SPH simulaciju slabo stlačivog fluida

za svaki vremenski korak t čini

konstruiraj strukturu podataka za prostorni pretragu prema algoritmu 2

paralelno za svaku česticu i čini

pronađi sve susjede čestice i koristeći algoritam 3

izračunaj gustoću koristeći formulu 2.13 i jezgru 3.5

kraj

paralelno za svaku česticu i čini

pronađi sve susjede čestice i koristeći algoritam 3

izračunaj silu tlaka f_p koristeći formulu 2.14 i jezgru 3.6

izračunaj silu viskoznosti f_v koristeći formulu 2.15 i jezgru 3.7

$$F_i \leftarrow f_p + f_v + f_i^{ext}$$

kraj

paralelno za svaku česticu i čini

integriraj brzinu i poziciju pomoću formule 3.3

kraj

kraj

3.7. Prikaz čestica

Čestice su prikazane tehnikom oglasne ploče. Za svaku česticu instancira se jedan kvadrat s normaliziranim koordinatama vrhova $(-1.0, -1.0)$, $(1.0, -1.0)$, $(-1.0, 1.0)$ i $(1.0, 1.0)$. Taj kvadrat u sjenčaru vrhova se orijentira prema kameri i skalira s obzirom na veličinu čestice koju prikazuje. Orijentacija prema kameri ostvarena je konstrukcijom novog koordinatnog sustava prema 3.11. Točka \mathbf{p}_c predstavlja poziciju kamere u prostoru, a \mathbf{p}_p

poziciju čestice u prostoru.

$$\mathbf{v}_{up} = [0.0, 1.0, 0.0]^T \quad (3.8)$$

$$\mathbf{v}_f = \mathbf{p}_c - \mathbf{p}_p \quad (3.9)$$

$$\mathbf{v}_r = \frac{\mathbf{v}_{up} \times \mathbf{v}_f}{\|\mathbf{v}_{up} \times \mathbf{v}_f\|} \quad (3.10)$$

$$\hat{\mathbf{v}}_{up} = \mathbf{v}_f \times \mathbf{v}_r \quad (3.11)$$

Poziciju svakog vrha kvadrata sada je moguće odrediti kao

$$\mathbf{p}_w = \mathbf{p}_p + p_{nx}s\mathbf{v}_r + p_{ny}s\hat{\mathbf{v}}_{up} \quad (3.12)$$

gdje je s veličina čestice, p_{nx} i p_{ny} normalizirane koordinate kvadrata kao što je prethodno navedeno, a \mathbf{p}_w pozicija vrha u prostoru.

Jednom pravilno pozicioniran kvadrat u prostoru potrebno je osjenčati kao sferu. Kako bi to postigli za svaki fragment potrebno je odrediti normalu na sferu prema 3.16

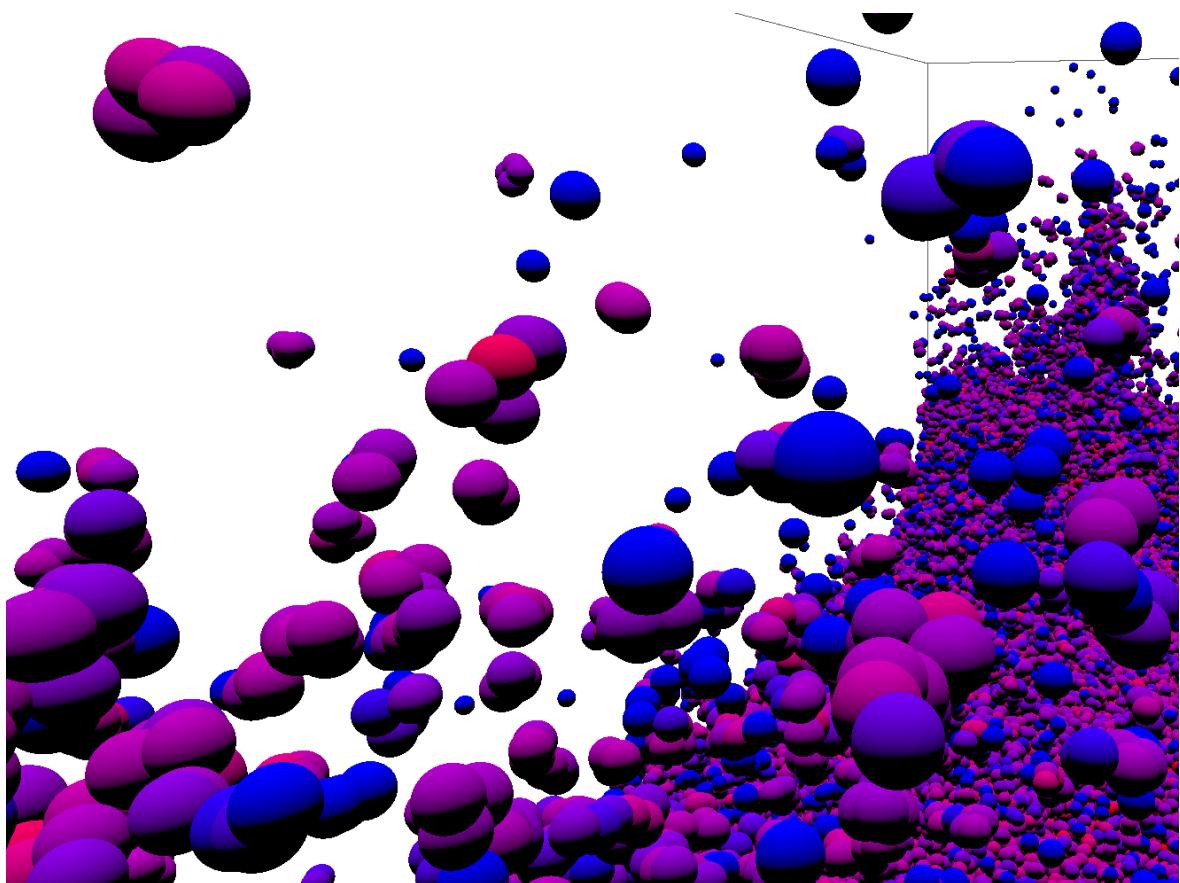
$$u' = 2u - 1 \quad (3.13)$$

$$v' = 2v - 1 \quad (3.14)$$

$$d = \sqrt{u'^2 + v'^2} \quad (3.15)$$

$$\mathbf{n} = [u', v', d]^T \quad (3.16)$$

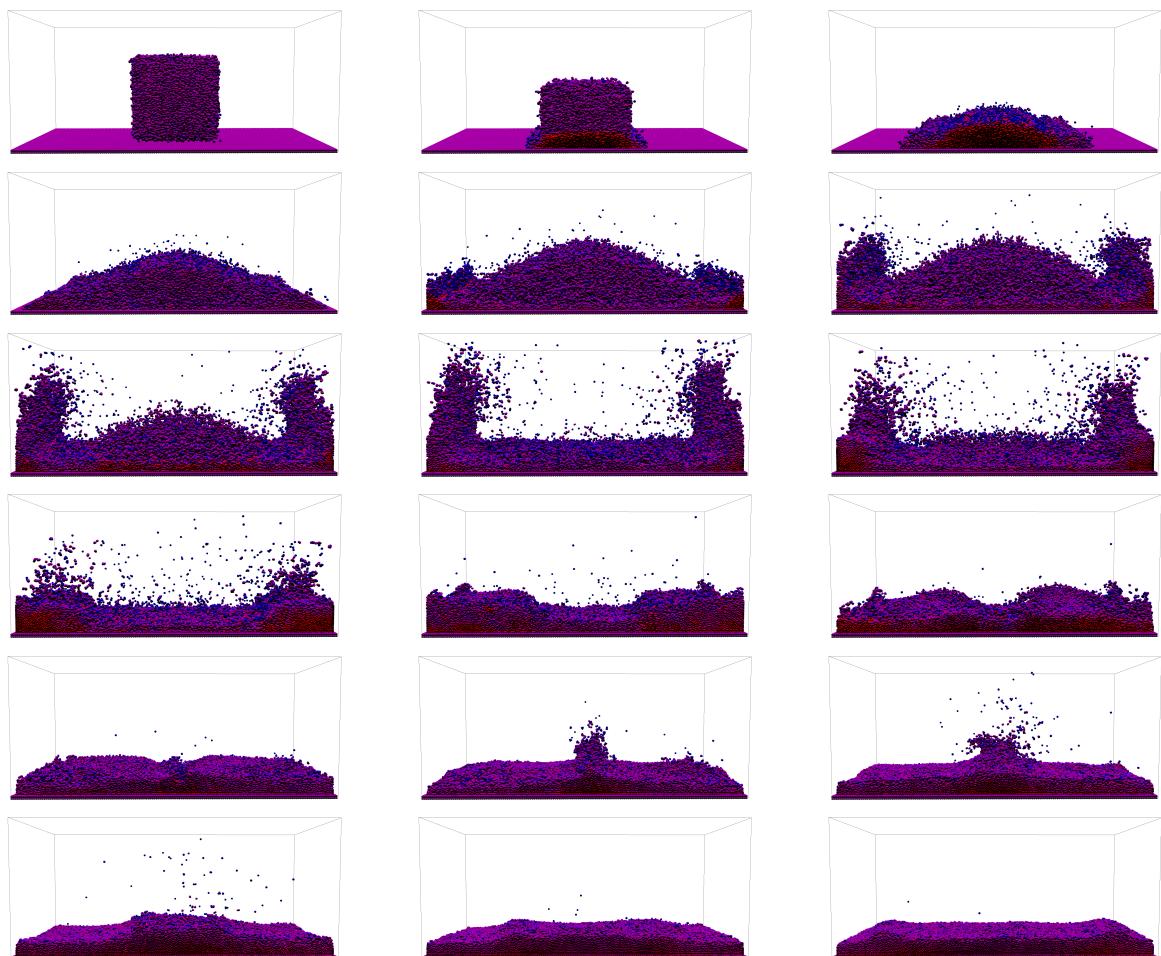
Svetlina svakog fragment određena je ambijentnom i difuznom komponentom osvjetljenja. Za difuznu komponentu korišten je Lambertov model.



Slika 3.4. Prikaz čestica

4. Rezultati

Izgled simulacije tijekom vremena prikazan je na slici 4.1.. Prikazana simulacija koristi sto tisuća čestica(brojeći neaktivne). Boja svake čestice prikazuje gustoću gdje crvena boja označava premalu gustoći, a plava preveliku. Za izvođenje jednog korak simulacije na RTX 3080 grafičkoj potrebno je oko 2ms.



Slika 4.1. Simulacija vode tokom vremena.

5. Zaključak i daljnji razvoj

Hidrodinamika zaglađenih čestica pokazala se kao moćna i svestrana metoda za simulaciju fluida, ali i rješavanje drugih problema u mehanici kontinuuma. Kroz rad pokazali smo kratak pregled teorijskih temelja na kojim počiva metoda i dokumentirali jednostavnu implementaciju. Iako rezultati već sada izgledaju kao razumna simulacija vode postoje brojna područja koje je moguće poboljšati.

Trenutačni model ne uzima površinsku napetost vode u obzir što dovodi do ne realističnog ponašanja vode. Modele ostalih sila također je moguće poboljšati. Simulacija također ne uspijeva održati konstantu gustoću fluida što je još jedan izvor problema. Moguće je poboljšati i interakciju s rubovima domene. Osim naveden i drugih nedostataka model moguće je poboljšati i samu implementaciju. Trenutni sjenčari za računanje ne pristupaju memoriji na optimalan način što značajno usporava računanje gustoća i sila.

Literatura

- [1] L. B. Lucy, “A numerical approach to the testing of the fission hypothesis”, *Astronomical Journal*, vol. 82, Dec. 1977, p. 1013-1024., sv. 82, str. 1013–1024, 1977.
- [2] J. Boudet, “Finite volume methods”, *Computational Fluid Dynamics*, str. 1–24, 2011.
- [3] S. Chen i G. D. Doolen, “Lattice boltzmann method for fluid flows”, *Annual review of fluid mechanics*, sv. 30, br. 1, str. 329–364, 1998.
- [4] V. Springel, “The cosmological simulation code gadget-2”, *Monthly notices of the royal astronomical society*, sv. 364, br. 4, str. 1105–1134, 2005.
- [5] A. Crespo, M. Gómez-Gesteira, i R. A. Dalrymple, “Modeling dam break behavior over a wet bed by a sph technique”, *Journal of waterway, port, coastal, and ocean engineering*, sv. 134, br. 6, str. 313–320, 2008.
- [6] S. Zheng, F. Gao, Z. Zhang, H. Liu, i B. Li, “Topology optimization on fuel tank rib structures for fuel sloshing suppression based on hybrid fluid–solid sph simulation”, *Thin-Walled Structures*, sv. 165, str. 107938, 2021.
- [7] M. Liu, G. Liu, K. Lam, i Z. Zong, “Smoothed particle hydrodynamics for numerical simulation of underwater explosion”, *Computational mechanics*, sv. 30, str. 106–118, 2003.
- [8] M. Müller, S. Schirm, i M. Teschner, “Interactive blood simulation for virtual surgery based on smoothed particle hydrodynamics”, *Technology and Health Care*, sv. 12, br. 1, str. 25–31, 2004.

- [9] C. Zhang, Y.-j. Zhu, D. Wu, N. A. Adams, i X. Hu, “Smoothed particle hydrodynamics: Methodology development and recent achievement”, *Journal of Hydrodynamics*, sv. 34, br. 5, str. 767–805, 2022.
- [10] D. Koschier, J. Bender, B. Solenthaler, i M. Teschner, “Smoothed particle hydrodynamics techniques for the physics based simulation of fluids and solids”, *arXiv preprint arXiv:2009.06944*, 2020.
- [11] D. J. Price, “Smoothed particle hydrodynamics and magnetohydrodynamics”, *Journal of Computational Physics*, sv. 231, br. 3, str. 759–794, 2012.
- [12] J. J. Monaghan, “Smoothed particle hydrodynamics”, *Reports on progress in physics*, sv. 68, br. 8, str. 1703, 2005.
- [13] ——, “Smoothed particle hydrodynamics”, In: *Annual review of astronomy and astrophysics. Vol. 30 (A93-25826 09-90)*, p. 543-574., sv. 30, str. 543–574, 1992.
- [14] L. Brookshaw, “A method of calculating radiative heat diffusion in particle simulations”, *Publications of the Astronomical Society of Australia*, sv. 6, br. 2, str. 207–210, 1985.
- [15] S. Green, “Particle simulation using cuda”, *NVIDIA whitepaper*, sv. 6, str. 121–128, 2010.
- [16] L. Ha, J. Krüger, i C. T. Silva, “Fast four-way parallel radix sorting on gpus”, u *Computer Graphics Forum*, sv. 28, br. 8. Wiley Online Library, 2009., str. 2368–2378.
- [17] M. Müller, D. Charypar, i M. Gross, “Particle-based fluid simulation for interactive applications”, u *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation.* Citeseer, 2003., str. 154–159.