

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2880

**FIZIKALNO TEMELJENA SIMULACIJA MATERIJALA
METODOM MATERIJALNE TOČKE**

Tin Blažević

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 2880

**FIZIKALNO TEMELJENA SIMULACIJA MATERIJALA
METODOM MATERIJALNE TOČKE**

Tin Blažević

Zagreb, lipanj 2022.

Zagreb, 11. ožujka 2022.

DIPLOMSKI ZADATAK br. 2880

Pristupnik: **Tin Blažević (0036507355)**

Studij: Računarstvo

Profil: Računarska znanost

Mentorica: prof. dr. sc. Željka Mihajlović

Zadatak: **Fizikalno temeljena simulacija materijala metodom materijalne točke**

Opis zadatka:

Proučiti fizikalne osnove metode materijalne točke. Proučiti metodu materijalne točke u simulacijama dinamike fluida. Razraditi dvodimenzionalni i trodimenzionalni implementacijski model proučene metode. Razraditi i ostvariti simulacijski model uz ostvarivanje animiranog prikaza rezultata. Načiniti testiranje na nizu primjera. Analizirati i ocijeniti ostvarene rezultate. Diskutirati upotrebljivost ostvarenih rezultata kao i moguća proširenja. Izraditi odgovarajući programski proizvod. Koristiti programski alat Unity. 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: 27. lipnja 2022.

SADRŽAJ

1. Uvod	1
2. Fizikalno temeljene simulacije	2
2.1. Numeričke simulacije	2
2.1.1. Podjela metoda	3
2.2. Navier-Stokesove jednačbe	4
2.3. Hidrodinamika zaglađenih čestica	4
2.4. Stabilni fluid	7
3. Metoda materijalne točke	12
3.1. Eksplicitna integracija kroz vrijeme	17
3.1.1. APIC prijenos	17
3.1.2. Ažuriranje deformacijskog gradijenta	18
3.1.3. Ažuriranje stanja	18
3.1.4. Akumuliranje sile	18
3.1.5. Algoritam metode materijalne točke za eksplicitni postupak	19
3.1.6. Pseudokod algoritma, čestice i čvora rešetke	21
4. Modeli materijala	26
4.1. Model vode	27
4.1.1. Pseudokod čestice vode	28
4.2. Modeli hiperelastičnog materijala	29
4.2.1. Neo-Hookeov model	29
4.2.2. Fiksni korotacijski model	29
4.2.3. Pseudokod hiperelastične čestice	31
4.3. Model snijega	32
4.3.1. Pseudokod čestice snijega	33
4.4. Model pijeska	35

4.4.1.	Projekcija na Drucker-Prager stožac	35
4.4.2.	Stvrdnjavanje pijeska prema Drucker-Prager modelu	37
4.4.3.	Pseudokod čestice pijeska	38
5.	Sudari s krutim tijelima	40
5.1.	Funkcije udaljenosti s predznakom	40
5.2.	Uvjeti na granici	41
5.3.	Trenje na površini krutog objekta	42
5.4.	Pseudokod jednostavnog krutog SDF objekta	42
6.	Implementacija	44
6.1.	Simulacijski model	44
6.2.	Upute za korištenje	44
6.3.	Vizualizacija rezultata	47
7.	Rezultati	48
8.	Zaključak	52
	Literatura	54

1. Uvod

Simuliranje fizikalnih ponašanja deformabilnih objekata i fluida važna je tema u području računalne grafike. Izazov je osmisliti postupak koji će ujediniti niz složenih ponašanja uz visoku preciznost. Računalna animacija zahtijeva vrhunske vizualne rezultate i prilagodljivost parametara.

Velike deformacije, odvajanje te spajanje elemenata, sudari i sprega između više materijala neki su od fenomena koje metoda materijalne točke pruža bez previše modifikacija. Složenije varijante u modele uključuju i toplinu.

Ovaj rad za cilj ima razviti razumijevanje metode materijalne točke krenuvši osnovnog pojma numeričke simulacije i kratkim pregledom dviju tradicionalnih postupaka, nakon čega slijedi teorijska podloga i formulacija eksplicitnog algoritma metode materijalne točke. Definirani su i modeli više materijala - vode, hiperelastičnog materijala, snijega i pijeska. Dodatno, opisuje se interakcija s ostalim, krutim objektima u sceni korištenjem funkcija udaljenosti s predznakom. Za opisane koncepte pruža se odgovarajući pseudokod, koji je u sklopu ovog rada implementiran koristeći programski alat Unity. Vizualni rezultati dobiveni su tehnikom praćenja zrake unutar alata Blender.

2. Fizikalno temeljene simulacije

2.1. Numeričke simulacije

Numerička simulacija korištenjem računala postala je važan pristup rješavanju složenih praktičnih problema u inženjerstvu i znanosti. Numerička simulacija prevodi bitne aspekte fizikalnog problema u diskretni matematički opis koji se rekreira i rješava na računalu gdje se virtualno otkrivaju i promatraju fenomeni prema potrebi. Bez tradicionalnog pristupa uvođenja slojeva pretpostavki i aproksimacija, moderni numerički pristupi problem savladavaju direktno, u svim detaljima i bez previše pretpostavki, koristeći pomoć rastuće računalne moći. Numerička simulacija nudi alternativu znanstvenim istraživanjima, skupim i vremenski zahtjevnim eksperimentima u laboratoriju ili na lokaciji. Numerički alati mogu biti korisniji za dobivanje potpunijih informacija koje je inače teško mjeriti. Numerička simulacija uz pomoć računala ima važnu ulogu u provjeri teorija i nudi uvid u rezultate eksperimenata te asistira u interpretaciji i otkrivanju novih fenomena. Dodatno, služi kao most između eksperimentalnih modela i teoretskih predviđanja.

Postupak stvaranja numeričke simulacije prema (Liu i Liu (2003)) je:

1. **promatranje fizikalnog fenomena** - pojednostavljivanje i izvlačenje bitnog
2. **matematički model** - jednačbe upravljanja (obične diferencijalne jednačbe, parcijalne diferencijalne jednačbe)
3. **diskretizacija domene** - rešetka, ćelije, čvorovi ili čestice
4. **numerički algoritmi** - početni uvjeti i uvjeti na granici (za granicu se uzima da pripada idealnom krutom tijelu), numerička diskretizacija, aproksimacija diferencijalnih jednačbi ili algebarsko rješavanje jednačbi
5. **programska implementacija** - računalna preciznost, brzina i memorija, robustnost i lakoća korištenja

6. numerička simulacija

Obično numerička simulacija problema iz domene računalne dinamike fluida obuhvaća sljedeće faktore:

- 1. jednadžbe upravljanja**
- 2. uvjeti na granici i početni uvjeti**
- 3. diskretizacija domene**
- 4. numerička diskretizacija**
- 5. numeričko rješavanje resultantnih algebarskih jednadžbi ili običnih diferencijalnih jednadžbi**

2.1.1. Podjela metoda

Kada se promatra gibanje kontinuuma kao što je fluid ili deformabilno tijelo, postoje dvije vrste pristupa, Lagrangeov i Eulerov. Pretpostavka kontinuuma znači da je za svaku točku domene definirana vrijednost atributa, koja se može dobiti interpolacijom vrijednosti iz susjednih točaka iz prethodno diskretizirane domene.

Lagrangeov pristup tretira kontinuum kao oblak čestica koje se gibaju kao fluid ili tijelo. Svaka točka fluida ili tijela promatra se kao odvojena čestica s vlastitim fizikalnim atributima. Eulerov pristup znatno je drugačiji i obično se koristi za fluide. Umjesto praćenja svake čestice, promatraju se fiksne točke u prostoru i kako se atributi fluida, kao što su gustoća, brzina i temperatura, u njima mijenjaju s vremenom. Numerički, Lagrangeov pristup odgovara korištenju sustava čestica, poznatom konceptu u računalnoj grafici, s korištenjem dodatne ili bez rešetke koja povezuje čestice, a Eulerov pristup odgovara korištenju fiksne rešetke predstavljene višedimenzionalnim nizom čiji elementi ne mijenjaju svoje pozicije u vremenu. Eulerov pristup služi za jednostavnije računanje prostornih derivacija kao što su gradijent tlaka i sila viskoznosti u odnosu na računanje istih u oblaku čestica.

Prije formulacije metode materijalne točke, koja koristi hibridnu formulaciju modela fluida, korisno je prvo promotriti dvije čiste metode i njihove ključne koncepte koje koristi i metoda materijalne točke. Prije samih metoda, potrebno je definirati Navier-Stokesove jednadžbe za čije rješavanje su te metode osmišljene.

2.2. Navier-Stokesove jednadžbe

Navier-Stokesove jednadžbe opisuju gibanje viskoznog fluida. Fluidi se opisuju skalarnim poljem gustoće ρ , skalarnim poljem tlaka p , a tradicionalna oznaka za vektorsko polje brzine je $\vec{u}$. Prva jednadžba formulira zakon očuvanja mase, a u tradicionalnom zapisu za nestlačivi fluid glasi:

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

Druga jednadžba formulira zakon očuvanja količine gibanja:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla p = \nu \nabla \cdot \nabla \vec{u} + \vec{g}. \quad (2.2)$$

Gdje ν predstavlja kinematičku viskoznost i govori koliko je unutarnje trenje čestica fluida, a $\vec{g}$ su vanjske sile od kojih je najjednostavniji primjer gravitacija.

Jednadžba (2.1) govori da je gradijent brzine jednak nuli. Naime kada bi gradijent brzine bio različit od nule, s obzirom na to da brzina ne postoji bez neke mase koja se giba, postojale bi točke koje djeluju kao njeni izvori i ponori. U izvorima bi masa naizgled nastajala, a u ponorima nestajala što očito nije slučaj u stvarnosti. Ovu jednadžbu je trivijalno zadovoljiti korištenjem sustava čestica ali znatno je složenije ako se koristi Eulerov prikaz fluida. Jednadžba (2.2) govori da se brzina mijenja u vremenu u ovisnosti o samoj brzini, tlaku, viskoznosti i vanjskim silama.

Ovaj oblik jednadžbi ne koristi se direktno nego predstavlja jednadžbe upravljanja - specifikaciju koja mora biti zadovoljena. Različite metode kreću od jednadžbi u ovom obliku ali ih diskretiziraju koristeći vlastite pretpostavke o modelu.

2.3. Hidrodinamika zaglađenih čestica

Metoda hidrodinamike zaglađenih čestica (eng. *Smoothed Particle Hydrodynamics*; SPH) izvorno je osmišljena za primjene u astrofizici Gingold i Monaghan (1977) i promatranje gibanja nebeskih tijela, a kasnije je postala popularna za simulaciju fluida. Formulacija SPH dijeli često se dijeli u dva ključna dijela. Prvi dio je aproksimacija jezgrenom funkcijom, a drugi je aproksimacija česticama. Jezgrena aproksimacija funkcije je integral umnoška proizvoljne funkcije i zadane jezgrene funkcije koja se često naziva i funkcija zaglađivanja ili potporna funkcija. Aproksimacija česticama je suma svih jezgrenih aproksimacija neke čestice i njenih susjeda. Drugim riječima, radi se o modelu čestica-čestica Lagrangeove prirode gdje jezgrena funkcija opisuje jačinu utjecaja neke čestice na druge čestice u prostoru oko sebe.

Kako bi neka funkcija bila korištena kao jezgrena za model SPH, mora zadovoljavati sljedeće uvjete Liu i Liu (2010):

- Mora biti normalizirana na svojoj domeni:

$$\int_{\Omega} W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' = 1, \quad (2.3)$$

gdje je W funkcija, Ω njena domena, $\mathbf{x}$ središte čestice, a $\mathbf{x}'$ točka u domeni funkcije.

- Trebala bi biti kompaktna - za dovoljnu udaljenost od čestice treba se evaluirati u nulu:

$$W(\mathbf{x} - \mathbf{x}') = 0, \text{ za } |\mathbf{x} - \mathbf{x}'| > \kappa h, \quad (2.4)$$

gdje je h radijus zaglađivanja, a κ faktor skaliranja. $|\mathbf{x} - \mathbf{x}'| \leq \kappa h$ definira S - jezgrenu ili potpurnu domenu funkcije.

- Mora biti nenegativna na svojoj domeni:

$$W(\mathbf{x} - \mathbf{x}') \geq 0, \forall \mathbf{x}' \in \Omega. \quad (2.5)$$

- Vrijednost funkcije treba se monotono smanjivati udaljavanjem od čestice:

$$\begin{aligned} & \forall \mathbf{x}_1', \mathbf{x}_2' \in \Omega \\ & |\mathbf{x} - \mathbf{x}_1'| > |\mathbf{x} - \mathbf{x}_2'| \Rightarrow \\ & W(\mathbf{x} - \mathbf{x}_1', h) < W(\mathbf{x} - \mathbf{x}_2', h). \end{aligned} \quad (2.6)$$

- Trebala bi zadovoljavati svojstvo Diracove delta funkcije kada radijus zaglađivanja teži u nulu:

$$\lim_{h \rightarrow 0} W(\mathbf{x} - \mathbf{x}', h) = \delta(\mathbf{x} - \mathbf{x}'). \quad (2.7)$$

- Trebala bi biti parna, odnosno simetrična:

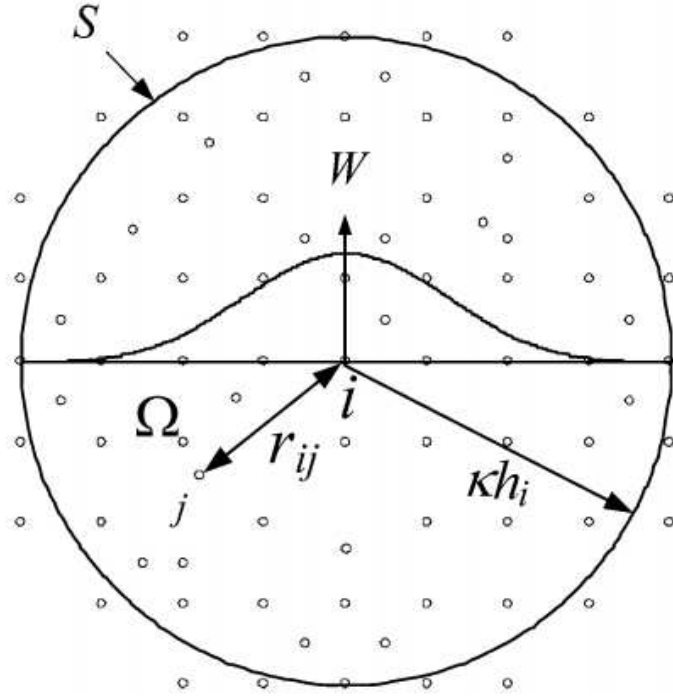
$$W(\mathbf{x} - \mathbf{x}', h) = W(-\mathbf{x} + \mathbf{x}', h). \quad (2.8)$$

- Trebala bi biti dovoljno glatka radi bolje aproksimacije.

Slika 2.1 prikazuje SPH jezgrenu funkciju sa središtem u čestici i s odgovarajućim simbolima.

U nastavku je dana SPH formulacija iz Bridson i Müller-Fischer (2007). Kreće se od Navier-Stokesovih jednažbi zapisanih u nešto drugačijem obliku u odnosu na (2.1) i (2.2):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \quad (2.9)$$



Slika 2.1: Grafički prikaz SPH jezgrene funkcije. Izvor: Liu i Liu (2010).

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \mu \nabla^2 \mathbf{v} + \rho \mathbf{g}. \quad (2.10)$$

Popularni izbor jezgrene funkcije je *poly6* jezgra jer se udaljenost pojavljuje samo u obliku kvadrata i ne koriste se korijeni:

$$W_{poly6}(r) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3, & 0 \leq r \leq h \\ 0, & \text{inače} \end{cases}. \quad (2.11)$$

Zakon očuvanja mase (2.9) automatski je zadovoljen jer se unutar samog postupka ne mijenja broj niti masa pojedinih čestica. Izraz $\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}$ iz (2.10) može se zamijeniti s $\frac{D\mathbf{v}}{Dt}$ jer se čestice kroz vrijeme gibaju s fluidom, odnosno advekcijski izraz $\mathbf{v} \cdot \nabla \mathbf{v}$ može se izostaviti u slučaju sustava čestica. Za preostale članove iz (2.10) primjenjuje se SPH pravilo na pojedinačne članove izraza nakon čega je potrebno rezultat učiniti simetričnim, jer jednaka sila mora vrijediti od čestice i prema čestici j i obrnuto. Za glađena polja A_s proizvoljnih atributa A_i čestica računaju se uz sljedeće pravilo:

$$A_s(\mathbf{x}) = \sum_j m_j \frac{A_j}{\rho_j} W(|\mathbf{x} - \mathbf{x}'|), \quad (2.12)$$

čiji je gradijent jednak:

$$\nabla A_s(\mathbf{x}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(|\mathbf{x} - \mathbf{x}'|). \quad (2.13)$$

Primjena SPH pravila na izraz $-\nabla p$ iz (2.10) uz korekciju radi simetrije daje izraz za silu tlaka:

$$\mathbf{f}_i^{tlak} = - \sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(|\mathbf{x} - \mathbf{x}'|). \quad (2.14)$$

Tlak nije atribut čestica pa je za evaluaciju (2.14) potrebna jednadžba idealnog plina:

$$p = k(\rho - \rho_0), \quad (2.15)$$

gdje je k konstanta plina ovisna o temperaturi, a ρ_0 gustoća, odnosno pritisak okoline.

Primjena SPH pravila na izraz $\mu \nabla^2 \mathbf{v}$ iz (2.10) uz korekciju daje izraz za silu viskoznosti:

$$\mathbf{f}_i^{viskoznost} = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W(|\mathbf{x} - \mathbf{x}'|). \quad (2.16)$$

Moguća interpretacija (2.16) je da česticu i promatramo u njenom referentnom okviru gdje ju okolina ubrzava u svom relativnom smjeru.

Na kraju se bez primjene SPH pravila dodaju vanjske sile kao što su gravitacija, sile uslijed sudara ili sile koje su rezultat interakcije korisnika sa simulacijom. Iz mase čestice i akumulirane sile računa se akceleracija iz čega se dobivaju nova brzina i pozicija čestice (plu-implicitni Eulerov postupak numeričke integracije).

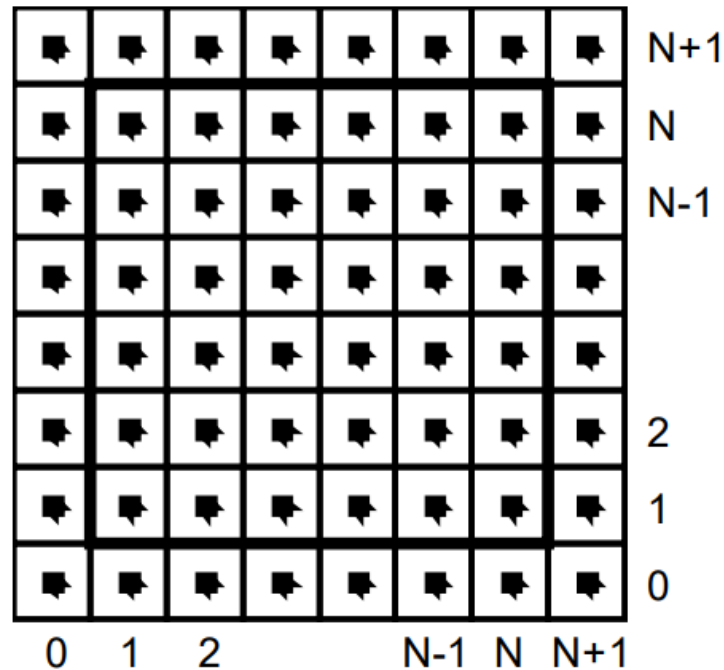
Za potrebe izvođenja u stvarnom vremenu, moguće je koristiti stablo za particioniranje prostora kako bi traženje susjednih čestica bilo složenosti $O(\log N)$. Više kompromis nego optimizacija, u izračunu se može uzeti maksimalno $k \in \mathbb{Z}$, $k > 0$ čestica umjesto svih u susjedstvu za potrebe u stvarnom vremenu, npr. u računalnim igrama.

2.4. Stabilni fluid

U ovom poglavlju ukratko je opisana metoda iz Stam (2003), formalnije definirana u Stam (1999). Često se naziva i metodom stabilnog fluida. Radi se o jednostavnoj Eulerovoj metodi koja žrtvuje preciznost ali zauzvrat ima mogućnost izvođenja u stvarnom vremenu bez paralelizacije zbog čega je stekla veliku popularnost u području računalne animacije i računalnim igrama dok je računalna moć bila ograničena. Unaprijeđene inačice algoritma implementirane su u programu Maya (kasnije Autodesk Maya). Izvorni kod u jeziku C za dvije dimenzije dostupan je u Stam (2003). Algoritam se lako može proširiti i na tri dimenzije.

Prostor se dijeli na uniformnu rešetku veličine N i dodaju se rubne ćelije za provjeru uvjeta na granici koje se ne iscrtavaju (slika 2.2). Rubne ćelije služe za računanje

sudara s granicom domene. Potrebno je onoliko slojeva rubnih ćelija koliko u postupku jednoj ćeliji treba susjeda iz pojedinog smjera za računanje idućeg stanja. Rešetka se može promatrati kao da su u središtima ćelija nepomične čestice za koje su definirani atributi.

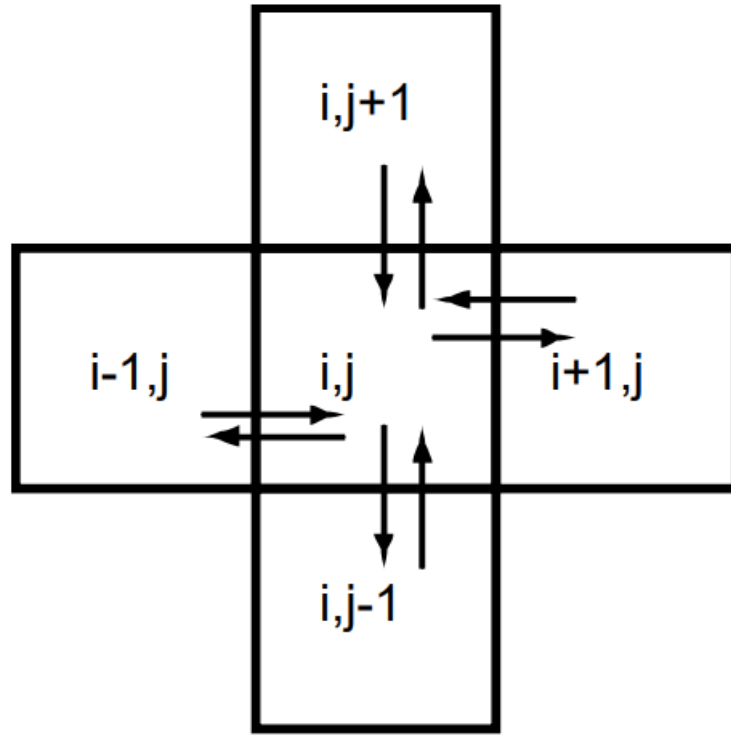


Slika 2.2: Nepomične čestice unutar središta ćelija uniformne rešetke s dodanim ćelijama na rubu. Izvor: Stam (2003).

U kontekstu Eulerovim metoda, za attribute ćelija se ne koristi izraz masa nego gustoća. Svaka ćelija ima svoj volumen i promjenjivu masu pa je prirodnije koristiti gustoću. Gibanje gustoće kreće od početno distribuirane gustoće na koju djeluju vanjski izvori, difuzija i na kraju se vrši pomak u smjeru nove brzine. Izvorni rad definira postupak uz implicitni Eulerov postupak integracije, koji je stabilan i za velike vremenske korake potrebne za izvršavanje u stvarnom vremenu.

Difuzija je spontano raspoređivanje gustoće iz jedne ćelije u susjedne (slika 2.3), odnosno jedna ćelija teži prema prosjeku svojih susjeda. Raspisano za sve ćelije u rešetci, dobiva se linearni sustav. U izvornom radu za rješavanje linearnog sustava koristi se Gauss-Seidelova metoda uz 20 iteracija. Dodatno, jačina difuzije gustoće se kontrolira parametrom koji ujedno služi za prigušivanje rješenja čime se količina gustoće sveukupno smanjuje i zakon očuvanja mase ne vrijedi ako se za jačinu difuzije uzme faktor veći od nule, ali zato je postupak garantirano stabilan.

Gauss-Seidelova metoda iterativni je postupak koji služi za rješavanje linearnih

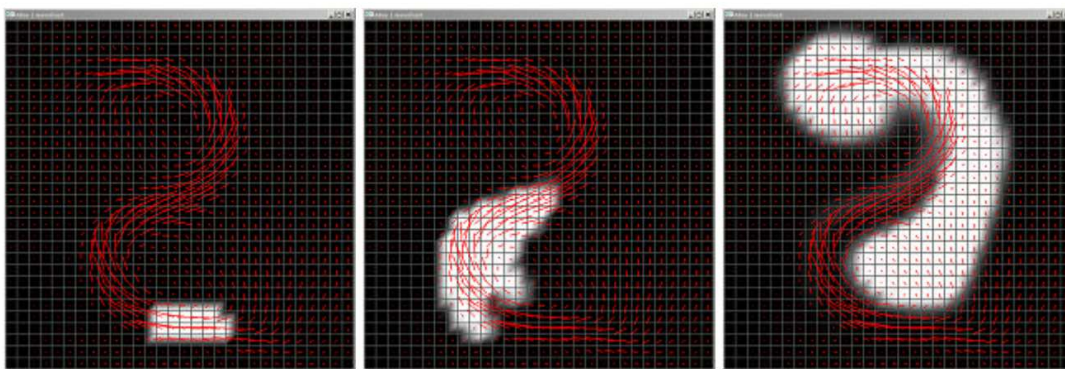


Slika 2.3: Difuzija između susjednih ćelija. Izvor: Stam (2003).

sustava oblika $\mathbf{Ax} = \mathbf{b}$. Matrica $\mathbf{A}$ se dekomponira u oblik $\mathbf{A} = \mathbf{L}_* + \mathbf{U}$, gdje je $\mathbf{L}_*$ donja trokutasta matrica, a $\mathbf{U}$ strogo gornja trokutasta. Matrice $\mathbf{L}_*$ i $\mathbf{U}$ direktno su zapisane u matrici $\mathbf{A}$ i nije ih potrebno posebno računati. Rješenje sustava tada ima oblik $\mathbf{x}^{k+1} = \mathbf{L}_*^{-1}(\mathbf{b} - \mathbf{U}\mathbf{x}^k)$. Formula po elementima glasi $x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^n a_{ij}x_j^k \right)$, $i = 1, 2, \dots, n$. Zbog člana $\sum_{j=1}^{i-1} a_{ij}x_j^{k+1}$ elementi rješenja moraju se računati slijedno i algoritam se ne može paralelizirati. Poznati su dovoljni ali ne i nužni uvjeti za konvergenciju postupka. Postupak može konvergirati i kada ti uvjeti nisu zadovoljeni. Stoga algoritam opisan u ovom poglavlju unutar Gauss-Seidelovog rješavača prigušuje rješenje u svakom koraku kako bi konvergencija bila garantirana.

Moguća alternativa je Jacobijeva metoda koja se može paralelizirati. Jacobijeva metoda matricu $\mathbf{A}$ dekomponira u $\mathbf{A} = \mathbf{D} + \mathbf{L} + \mathbf{U}$ gdje je $\mathbf{D}$ dijagonalna matrica, $\mathbf{L}$ donja trokutasta i $\mathbf{U}$ gornja trokutasta matrica. Rješenje ima oblik $\mathbf{x}^{k+1} = \mathbf{D}^{-1}(\mathbf{b} - (\mathbf{L} + \mathbf{U})\mathbf{x}^k)$, a formula po elementima je $x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij}x_j^k \right)$, $i = 1, 2, \dots, n$.

Nakon difuzije slijedi advekcija - gibanje atributa u smjeru brzine (slika 2.4). Drugi nazivi za advekciju su konvekcija i transport. Ideja je promatrati središta ćelija kao čestice koje se gibaju i gledati gdje će završiti pa ih vratiti nazad u rešetku ali nije baš jasno kako to implementirati na jednostavan i efikasan način jer takav postupak nije

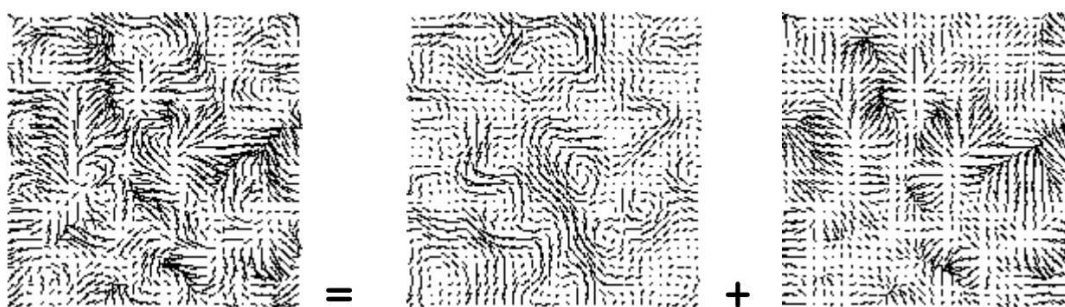


Slika 2.4: Advekcija gustoće u smjeru vektorskog polja brzine. Izvor: Stam (2003).

stabilan (eksplicitni Eulerov postupak). Ovaj algoritam umjesto toga promatra čestice koje bi u jednom vremenskom koraku gibanjem u smjeru brzine završile u središtu ćelija iz trenutnog koraka. Stoga su potrebne dvije rešetke, jedna u kojoj su vrijednosti atributa iz prethodnog koraka i jedna u koju se spremaju trenutne vrijednosti. Pozicija čestica koja se dobije promatra se na rešetci iz prethodnog koraka i količina atributa koju ta čestica nosi dobije se linearnom interpolacijom četiri ćelije najbliže toj točki. Sama pozicija tih čestica dobiva se rješavanjem linearnog sustava Gauss-Seidelovom metodom na isti način kao i za difuziju.

Na brzinu djeluju vanjske sile, viskozna difuzija i samo-advekcija. Viskozna difuzija ista je kao ona za gustoću ali za jačinu difuzije i prigušivanje rješenja koristi drugi parametar - faktor viskoznosti, odnosno viskoznost smanjuje ukupnu količinu gibanja s vremenom. Samo-advekcija ista je kao za gustoću, a znači da se vektorsko polje brzine giba u smjeru samog sebe. S obzirom na to da je brzina vektorsko polje, za razliku od gustoće koja je skalarno, potreban je dodatan korak koji se zove projekcija i popravljajući dobiveni rezultat. Koristeći Helmholtz-Hodgeovu dekompoziciju, svako dovoljno glatko vektorsko polje može se rastaviti u zbroj dva posebna polja - jedno gradijentno (bez vrtloženja) i drugo konzervativno (nestlačivo, bez divergencije), kao što je prikazano na slici 2.5. Gradijentno polje može se promatrati kao visinska mapa, dok je u konzervativnom polju visina u svakoj točki jednaka pa izgleda spljošteno u odnosu na izvorno - otuda naziv projekcija. Operacija projekcije uklanja gradijent iz vektorskog polja brzine kako je specificirano u (2.1), a vizualni rezultat su vrtlozi unutar fluida. Pokazuje se da je potrebno oduzeti gradijent tlaka kako bi se dobilo traženo vektorsko polje brzine koje čuva masu, a taj gradijent se dobiva rješavanjem Poissonove jednadžbe.

U jednom koraku simulacije prvo se izvršavaju sve operacije nad poljem brzine



Slika 2.5: Helmholtz-Hodgeova dekompozicija. Rastav vektorskog polja s lijeve strane jednako-
 sti u zbroj konzervativnog (lijevo) i gradijentnog polja (desno). Izvor: Stam (2003).

gdje se obavlja difuzija pa advekcija po pojedinačnim prostornim komponentama. Nakon svake operacije koja mijenja brzinu potrebno je obaviti projekciju jer se ne može nastaviti s neispravnim rezultatom. Na kraju se obavlja difuzija i razmjena gustoće među ćelijama. Nakon svake velike operacije potrebno je provjeriti uvjete na granici i korigirati rezultat. Uvjeti su da horizontalna komponenta brzine uz vertikalne zidove treba biti nula i obrnuto, vertikalna komponenta brzine treba biti nula uz horizontalne zidove. Uvjeti na granici su detaljnije razrađeni u poglavlju 5.

3. Metoda materijalne točke

Potreba za računalnom efikasnošću, složenom promjenom topologije i numeričkom stabilnošću doveli su inženjere do hibridnih Lagrange-Eulerovih metoda. Metoda materijalne točke ili metoda točke materijala (eng. *Material Point Method*, MPM) kombinira Lagrangeov sustav čestica s Eulerovim Kartezijanskim rešetkama. Metoda se javlja kao generalizacija PIC (eng. *Particle In Cell*) i FLIP (eng. *Fluid Implicit Particle*) metoda na mehaniku čvrstih materijala. FLIP metoda je poznata kao koristan alat za simulaciju tekućina. Metoda materijalne točke predstavljena je relativno nedavno.

Metoda materijalne točke pokazala se učinkovitom za simulaciju niza fizikalnih fenomena, uključujući tekućine, hiperelastične materijale, snijeg, pijesak, ali i složenije fenomene poput kaljenja i smrzavanja. Metoda implicitno podržava sudare među česticama i razdvajanje čestica materijala kroz pozadinsku Eulerovu rešetku.

Strože govoreći, metoda materijalne točke spada pod Lagrangeove metode, ali koristi Eulerovu rešetku za računanje derivacija. To uklanja potrebu za povezivanjem čestica Lagrangeovom rešetkom radi računanja derivacija, koja bi se zapetljala uslijed velikih deformacija u odnosu na početnu konfiguraciju. Ovo omogućava simuliranje šireg spektra materijala u odnosu na čisti Lagrangeov pristup. Eulerov aspekt omogućava prirodno tretiranje topoloških promjena i sudara, kako među česticama, tako i s drugim objektima, međutim gubi se dio preciznosti.

Prednosti MPM metode kao hibridnog pristupa u odnosu na tradicionalne Lagrangeove i Eulerove postupke su sljedeći:

- MPM omogućava fizikalno ispravnu i preciznu diskretizaciju fizikalnih zakona.
- Uvjeti na granici, sudari s čvrstim tijelima i vanjske sile mogu se primjenjivati i na rešetci i na česticama.
- Automatski sudari i kontakt između čestica.
- Automatska ponašanja rastavljanja ili lomljenja i spajanja. Ovo je korisno za fluide i zrnate materijale.
- Automatske interakcije između više materijala mogu se postići davanjem razli-

čitih svojstava česticama ili korištenjem različitih konstitutivnih modela - jednadžbi koje opisuju materijal.

- MPM se može koristiti za simulaciju Lagrangeovih sila baziranih na rešetci bez gubitaka ostalih prednosti. Ovo omogućava interakciju objekata koji su bazirani na česticama i objekata baziranih na rešetci, a sudari se obavljaju automatski.

Kinematika se odnosi na proučavanje gibanja nastalih u materijalima kontinuuma. U središtu pozornosti je promjena oblika, odnosno deformacija. Deformacija se može promatrati lokalno ili globalno, u različitim koordinatnim sustavima. Opisivanje gibanja i deformacije izuzetno je bitno za izvod jednadžbi upravljanja i odziva materijala.

U mehanici kontinuuma, deformacija se prikazuje korištenjem prostora materijala (bez deformacije) $\mathbf{X}$, prostora svijeta (s deformacijom) $\mathbf{x}$ i preslikavanjem iz lokalnog u globalni prostor $\phi(\mathbf{X}, t)$. Drugim riječima, preslikavanje $\phi(\mathbf{X}, t) : \Omega^0 \rightarrow \Omega^t$ opisuje gibanje svake točke materijala kroz vrijeme u odnosu na početnu poziciju $\mathbf{X}$

$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t) = \phi(\mathbf{X}, t). \quad (3.1)$$

Oznaka Ω^0 predstavlja točke materijala u početnoj konfiguraciji, a Ω^t u trenutku t . Ovo preslikavanje istovjetno je konceptu matrice preslikavanja u računalnoj grafici. Objekt u nekom početnom stanju može biti skaliran, rotiran i translatican, a ukupno gibanje se može prikazati umnoškom tih matrica. Ta konačna matrica je upravo deformacijska mapa. Dalje, brzina neke točke materijala $\mathbf{X}$ u trenutku t dana je jednadžbom

$$\mathbf{V}(\mathbf{X}, t) = \frac{\partial \phi}{\partial t}(\mathbf{X}, t) \quad (3.2)$$

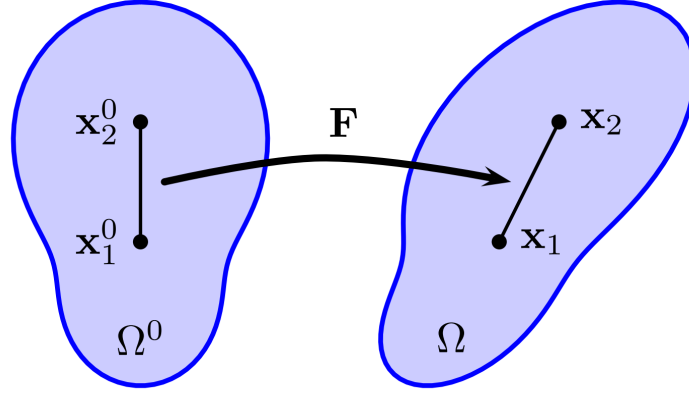
dok je akceleracija

$$\mathbf{A}(\mathbf{X}, t) = \frac{\partial^2 \phi}{\partial t^2}(\mathbf{X}, t). \quad (3.3)$$

Brzina $\mathbf{V}$ i akceleracija $\mathbf{A}$ definirani izrazima 3.2 i 3.3 zasnivaju se na Lagrangeovom pogledu. To znači da se mjere nad pojedinačnim česticama. Čestica ima svoju masu i volumen.

Prva derivacija, Jakobijan preslikavanja ϕ od velike je važnosti i označava se kao $\mathbf{F}$. $\mathbf{F}$ se naziva deformacijskim gradijentom, $\mathbf{F} = \frac{\partial \phi}{\partial \mathbf{X}}$. Deformacijski gradijent predstavlja lokalnu deformaciju materijala - odnos između susjednih točaka, slika 3.1. Neka su $\mathbf{x}_1^0$ i $\mathbf{x}_2^0$ dvije točke u početnoj konfiguraciji materijala, a $\mathbf{x}_1$ i $\mathbf{x}_2$ u trenutnoj konfiguraciji. Tada vrijedi $(\mathbf{x}_2 - \mathbf{x}_1) = \mathbf{F}(\mathbf{x}_2^0 - \mathbf{x}_1^0)$.

Često je korištena i determinanta od $\mathbf{F}$, oznake J , $J = \det(\mathbf{F})$. J predstavlja omjer trenutnog i početnog volumena. Za kruta gibanja (translacija i rotacija) $\mathbf{F}$ je



Slika 3.1: Grafički prikaz deformacijskog gradijenta. Izvor: Jiang et al. (2016)

rotacijska matrica i vrijedi $J = 1$. $J > 1$ znači da se volumen povećao, a $J < 1$ da se volumen smanjio.

Kod jednadžbi upravljanja i zakona očuvanja količine gibanja, veliku ulogu imaju stres i naprezanje materijala. Stres i naprezanje povezani su konstitutivnom relacijom, koja se naziva i konstitutivnim modelom ili funkcijom gustoće energije naprezanja. Slično kao što u elementi u strujnim krugovima imaju svoju strujno-naponsku karakteristiku koja modelira odziv elementa, tako konstitutivna relacija daje odziv materijala karakteriziran stresom i naprežanjem.

Relacija stresa i naprezanja često se iskazuje preko deformacijskog gradijenta i prvog Piola-Kirchhoff stresa $\mathbf{P}$ zbog prirodnih oznaka u prostoru materijala. Neka je $\Psi(\mathbf{F})$ funkcija gustoće elastične energije. Rezultat funkcije $\Psi(\mathbf{F})$ je skalar, a dizajnirana je da kažnjava odstupanje od krute deformacije (translacija i rotacija). Tada vrijedi

$$\mathbf{P} = \frac{\partial \Psi}{\partial \mathbf{F}}. \quad (3.4)$$

$\mathbf{P}$ je matrica istih dimenzija kao $\mathbf{F}$. U inženjerskoj literaturi često se koristi i Cauchyjev stres $\boldsymbol{\sigma}$, koji je povezan s 3.4 preko relacije

$$\boldsymbol{\sigma} = \frac{1}{J} \mathbf{P} \mathbf{F}^\top = \frac{1}{\det(\mathbf{F})} \frac{\partial \Psi}{\partial \mathbf{F}} \mathbf{F}^\top. \quad (3.5)$$

Ponašanje materijala definirano je interakcijom preslikavanja ϕ i stresa $\boldsymbol{\sigma}$ ili $\mathbf{P}$. Relacija $\frac{\partial \Psi}{\partial \mathbf{F}}$ predstavlja promjenu elastične energije uzrokovanu promjenom deformacijskog gradijenta. Različiti materijali imat će drugačije definiranu funkciju Ψ čime se u konačnici definira njihov odziv $\frac{\partial \Psi}{\partial \mathbf{F}}$.

Ponašanje nekih materijala nije u potpunosti elastično nego ima i plastičnu komponentu. Na primjer, tanka metalna žica namotana u spiralu predstavlja plastičnu deformaciju, a titranje te opruge je elastično. Deformacijski gradijent se može razložiti na elastični i plastični dio

$$\mathbf{F} = \mathbf{F}_E \mathbf{F}_P. \quad (3.6)$$

Plastični dio deformacijskog gradijenta $\mathbf{F}_P$ predstavlja dio povijesti materijala koji je zaboravljen. Metalna žica namotana u spiralu ponašat će se kao da je oduvijek bila opruga. Međutim, lagana kompresija dovodi do naprezanja i elastične deformacije. Opruga pamti ovu deformaciju i pokušava ju poništiti. Puna povijest metalne žice sadržana je u namotavanju u oblik opruge $\mathbf{F}_P$ i kompresiji $\mathbf{F}_E$. Elastični odziv je funkcija jedino $\mathbf{F}_E$ komponente. Lokalna deformacija sadržana u $\mathbf{F}_P$ je trajna i predstavlja novi oblik mirovanja. $\mathbf{F}_P$ može utjecati na druge parametre modela materijala, u kontekstu zrnatih materijala radi se o stvrdnjavanju. Uz plastičnost, $\Psi(\mathbf{F})$ naziva se funkcijom gustoće elasto-plastične potencijalne energije, a $\frac{\partial \Psi}{\partial \mathbf{F}}$ naziva se elasto-plastičnom konstitutivnom relacijom.

Jednadžbe kojima se upravlja promjenom deformacije $\phi(\mathbf{X})$ su zakon očuvanja mase

$$\frac{D\rho}{Dt} = 0, \quad (3.7)$$

zakon očuvanja količine gibanja

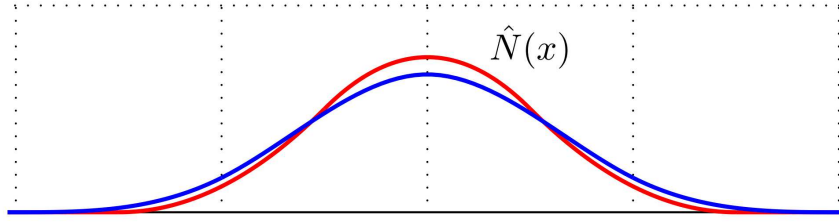
$$\rho \frac{D\mathbf{v}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g}, \quad (3.8)$$

i konstitutivna relacija

$$\boldsymbol{\sigma} = \frac{1}{J} \frac{\partial \Psi}{\partial \mathbf{F}_E} \mathbf{F}_E^\top. \quad (3.9)$$

Diskretizacija jednadžbi upravljanja u kontekstu metode materijalne točke složen je postupak i u nastavku će biti dan samo konačni rezultat. U nastavku je dana formulacija metode materijalne točke iz (Jiang et al. (2016)).

Metoda materijalne točke zahtijeva C^1 kontinuitet interpolacijske funkcije radi izbjegavanja nestabilnosti uslijed prijelaza čestica između ćelija. Interpolacijske funkcije koje se koriste u praksi su jednodimenzionalni kvadratni ili kubni B-splajnovi (slika 3.2) i njihov dijadski produkt. Smatra se da kubni daje bolje vizualne rezultate ali ponešto je skuplji za izračun. U literaturi i programskim matematičkim knjižicama za dijadski produkt može se naći i naziv vanjski produkt - kada dva vektora množenjem daju matricu.



Slika 3.2: Kvadratni (crveno) i kubni (plavo) splajn. Izvor: Jiang et al. (2016)

Kvadratni B-splajn dan je formulom

$$N_{kvadratna}(x) = \begin{cases} \frac{3}{4} - |x|^2, & 0 \leq |x| < \frac{1}{2} \\ \frac{1}{2}(\frac{3}{2} - |x|)^2, & \frac{1}{2} \leq |x| < \frac{3}{2} \\ 0, & \frac{3}{2} \leq |x| \end{cases} \quad (3.10)$$

a kubni formulom

$$N_{kubna}(x) = \begin{cases} \frac{1}{2}|x|^3 - |x|^2 + \frac{2}{3}, & 0 \leq |x| < 1 \\ \frac{1}{6}(2 - |x|)^3, & 1 \leq |x| < 2 \\ 0, & 2 \leq |x| \end{cases} \quad (3.11)$$

Tada je jezgrena funkcija definirana kao

$$N_i(\mathbf{x}_p) = N\left(\frac{1}{h}(x_p - x_i)\right)N\left(\frac{1}{h}(y_p - y_i)\right)N\left(\frac{1}{h}(z_p - z_i)\right), \quad (3.12)$$

Gdje je h udaljenost između čvorova rešetke, i čvor, p čestica, a $\mathbf{x} = (x, y, z)$ odgovarajuća pozicija. Gradijent jezgre definira se formulom

$$\nabla N_i(\mathbf{x}_p) = \begin{pmatrix} \frac{1}{h}N'\left(\frac{1}{h}(x_p - x_i)\right)N\left(\frac{1}{h}(y_p - y_i)\right)N\left(\frac{1}{h}(z_p - z_i)\right) \\ N\left(\frac{1}{h}(x_p - x_i)\right)\frac{1}{h}N'\left(\frac{1}{h}(y_p - y_i)\right)N\left(\frac{1}{h}(z_p - z_i)\right) \\ N\left(\frac{1}{h}(x_p - x_i)\right)N\left(\frac{1}{h}(y_p - y_i)\right)\frac{1}{h}N'\left(\frac{1}{h}(z_p - z_i)\right) \end{pmatrix} \quad (3.13)$$

Gdje je $'N(x)$ prva derivacija funkcije $N(x)$. Umjesto $N_i(\mathbf{x}_p)$ koristi se skraćeni zapis w_{ip} , a umjesto ∇w_{ip} koristi se zapis ∇w_{ip} . Zapis $N(\mathbf{x})$ i $\nabla N(\mathbf{x})$ označava da se radi o funkciji, a w_{ip} i ∇w_{ip} je funkcija evaluirana za udaljenost između čvora i i čestice p .

3.1. Eksplicitna integracija kroz vrijeme

Najjednostavnija metoda integracije jednadžbi upravljanja je eksplicitnim Eulerovim postupkom. Postupak zahtijeva mali vremenski korak zbog manje stabilnosti u odnosu na implicitni postupak. Međutim, implicitni postupak zahtijeva rješavanje nelinearnog sustava, što je znatno kompliciranije a može biti i računski zahtjevno. U slučajevima osjetljivih materijala, kao što su snijeg i pijesak, potreban je poprilično malen ili čak adaptivni vremenski korak.

3.1.1. APIC prijenos

Za prijenose između čestica i rešetke postoji više metoda, ali APIC (eng. *Affine Particle In Cell*) ima poželjna svojstva i veliku stabilnost (Jiang et al. (2017)). Autori APIC metode pokazuju da njihova metoda bolje čuva kutnu količinu gibanja u odnosu na ostale, a nudi i veću stabilnost postupka u odnosu na linearne kombinacije PIC i FLIP metoda.

Prijenos s čestica na rešetku je

$$m_i = \sum_p w_{ip} m_p, \quad (3.14)$$

$$m_i \mathbf{v}_i = \sum_p w_{ip} m_p (\mathbf{v}_p + \mathbf{B}_p (\mathbf{D}_p)^{-1} (\mathbf{x}_i - \mathbf{x}_p)), \quad (3.15)$$

Gdje je $\mathbf{B}_p$ akumulirani dijadski produkt brzina čvorova iz prošlog koraka i odgovarajućih težina, demonstriran kasnije u pseudokodu. $\mathbf{D}_p$ je matrica dana izrazom

$$\mathbf{D}_p = \sum_i w_{ip} (\mathbf{x}_i - \mathbf{x}_p) (\mathbf{x}_i - \mathbf{x}_p)^\top \quad (3.16)$$

i služi za očuvanje afinog gibanja. Odgovarajući prijenos nazad s rešetke na čestice je

$$\mathbf{v}_p = \sum_i w_{ip} \mathbf{v}_i \quad (3.17)$$

$$\mathbf{B}_p = \sum_i w_{ip} \mathbf{v}_i (\mathbf{x}_i - \mathbf{x}_p)^\top. \quad (3.18)$$

Matrica $\mathbf{D}_p$ poprima jednostavan oblik kada se koriste kvadratni ($\mathbf{D}_p = \frac{h^2}{4} \mathbf{I}$) i kubni ($\mathbf{D}_p = \frac{h^2}{3} \mathbf{I}$) B-splajn za interpolaciju. Tada je i inverz matrice trivijalan. U literaturi je moguće naći umnožak $\mathbf{B}_p (\mathbf{D}_p)^{-1}$ zamijenjen jednom matricom $\mathbf{C}_p$.

3.1.2. Ažuriranje deformacijskog gradijenta

Na početku svakog koraka, lokacije čvorova na rešetki su nedeformirane. Deformacijski gradijent se ažurira korištenjem gibanja na rešetci pravilom

$$\mathbf{F}_p^{n+1} = \left(\mathbf{I} + \Delta t \sum_i \mathbf{v}_i^{n+1} (\nabla w_{ip})^\top \right) \mathbf{F}_p^n, \quad (3.19)$$

gdje $\sum_i \mathbf{v}_i^{n+1} (\nabla w_{ip})^\top$ odgovara gradijentu brzine $\nabla \mathbf{v}_p^{n+1}$.

3.1.3. Ažuriranje stanja

Integracija polu-implicitnim Eulerovim postupkom nad rešetkom implicira

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \Delta t \mathbf{f}_i(\mathbf{x}_i^n) / m_i, \quad (3.20)$$

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \Delta t \mathbf{v}_i^{n+1}. \quad (3.21)$$

3.1.4. Akumuliranje sile

Ukupna potencijalna elastična energija dana je izrazom

$$e = \sum_p V_p^0 \Psi_p(\mathbf{F}_p), \quad (3.22)$$

gdje je V_p^0 početni volumen čestice p . Elastična sila nad čvorovima je tada negativni gradijent ukupne potencijalne elastične energije evaluirana u pozicijama čvorova

$$\mathbf{f}_i(\mathbf{x}) = -\frac{\partial e}{\partial \mathbf{x}_i}(\mathbf{x}) = -\sum_p V_p^0 \left(\frac{\partial \Psi_p}{\partial \mathbf{F}}(\mathbf{F}_p(\mathbf{x})) \right) (\mathbf{F}_p^n)^\top \nabla w_{ip}^n. \quad (3.23)$$

U slučaju polu-implicitnog Eulerovog postupka, izraz za silu je

$$\mathbf{f}_i^n = \mathbf{f}_i(\mathbf{x}_i^n) = -\sum_p V_p^0 \left(\frac{\partial \Psi_p}{\partial \mathbf{F}}(\mathbf{F}_p^n) \right) (\mathbf{F}_p^n)^\top \nabla w_{ip}^n, \quad (3.24)$$

koji potpuno ovisi o trenutnim, poznatim vrijednostima. Alternativno, izraz za silu se može zapisati korištenjem Cauchyjevog stresa

$$\mathbf{f}_i^n = \mathbf{f}_i(\mathbf{x}_i^n) = -\sum_p V_p^n \boldsymbol{\sigma}_p^n \nabla w_{ip}^n. \quad (3.25)$$

3.1.5. Algoritam metode materijalne točke za eksplicitni postupak

U nastavku je dan potpuni algoritam metode materijalne točke za integraciju polu-implicitnim Eulerovim postupkom. Čestice se na početku ispravno inicijaliziraju - moraju imati definiranu masu i volumen, imaju početnu poziciju i brzinu, a deformacijski gradijent je matrica identiteta. Potrebno je postaviti i ostale konstitutivne parametre modela, a matrice B_p i A_p su nul-matrice. Početni volumen čestica može se odrediti iz volumena početnog oblika (npr. kugla ili kvadar) i ukupnog broja čestica, ali preporučuje se napraviti prijenos mase s rešetke na čestice i izračun volumena (samo jednom, kod početne konfiguracije). Prvi način rezultira jednolikim česticama, dok drugi unosi laganu varijaciju. Masa se u prvom slučaju računa iz početnog volumena i gustoće materijala.

1. **Prijenos informacija s čestica na rešetku (eng. *Particle to Grid*); P2G).** Prema APIC metodi, prijenos atributa čestica p u ćeliju i dan je izrazima 3.14 i 3.15.
2. **Izračun brzine na rešetci.** Korištenjem izraza:

$$\mathbf{v}_i = \frac{m_i \mathbf{v}_i}{m_i}. \quad (3.26)$$

Za ćelije koje imaju masu jednaku 0, m_i i $\mathbf{v}_i$ ručno se postavljaju u 0. Ispostavlja se da nije potrebna klasična usporedba za realne brojeve je li masa ispod nekog praga, nego treba uspoređivati direktno s 0.0. U praksi će dijeljenje malom masom biti balansirano malenim težinskim faktorom kod vraćanja informacija nazad na čestice. Postavljanje nekakvog praga za masu bi čak moglo dovesti do gubitaka količine gibanja.

3. **Brojanje stupnjeva slobode.** Ovaj korak je bitan zbog efikasnosti. Kao broj stupnjeva slobode uzimaju se ćelije s masom različitom od 0. Ćelije s masom jednakom 0 se ne mijenjaju i ne tretiraju se kao nepoznanice. Za njih nije potrebno provoditi izračune.
4. **Računanje sile na rešetci.** Korištenje formule 3.24.
5. **Ažuriranje brzine i provjera uvjeta na granici.** Polu-implicitni Eulerov postupak za računanje nove brzine za čvorove, izraz 3.20. U ovom koraku dodatno treba provjeriti uvjete na granici i korigirati brzine zbog sudara i trenja. Ovaj dio je detaljnije objašnjen u poglavlju 5.
6. **Ažuriranje deformacijskog gradijenta čestica.** Korištenjem izraza 3.19. Gibanje čvorova je imaginarno i ne mijenja stanje rešetke, samo brzine se spremaju.

7. **Prijenos rešetke na čestice (eng. *Grid to Particle*; G2P).** Prema APIC metodi, prijenos iz ćelija i u česticu p , izrazi 3.17 i 3.18.
8. **Advekcija čestica.** Polu-implicitni Eulerov postupak nad česticama, izraz 3.21. Čestice se mogu ovako advektirati samo kod korištenja APIC metode. Kod korištenja FLIP ili linearne kombinacije PIC i FLIP metoda, na čestice se prenose advektirane pozicije s rešetke $\mathbf{x}_p^{n+1} = \sum_i \mathbf{x}_i^{n+1} w_{ip}$. Oba načina su ekvivalentna kod APIC metode i svejedno je koji se koristi.

3.1.6. Pseudokod algoritma, čestice i čvora rešetke

U ovom dijelu je dan pseudokod za apstraktnu česticu (pseudokod 1), čvor rešetke (pseudokod 2) i korak metode materijalne točke (pseudokod 3). Algoritam sadrži dio za jednostavnu implementaciju adaptivnog vremenskog koraka. Nakon što se prenesu informacije s čestica na rešetku, promatra se maksimalni pomak čvora na rešetci. Omjer najvećeg dozvoljenog pomaka i maksimalnog pomaka na rešetci određuje novi vremenski korak koji se koristi u nastavku algoritma, za integraciju u polu-implicitnom Eulerovom postupku. Ovaj korak nije nužan za većinu materijala, ali je potreban za osjetljive zrnate materijale kao što je pijesak. Dodatno, za ažuriranje pozicija čestica i deformacijskog gradijenta koristi se brzina nakon sudara, a na rešetci se koristi brzina nakon trenja. Korištenje brzine nakon trenja za čestice dovodi do nestabilnog ponašanja kod implicitne integracije. Kod eksplicitne integracije je kod čestica svejedno koristi li se brzina nakon sudara ili nakon trenja.

Pseudokod 1 Čestica

```
1: članske varijable
2:    $V_p \rightarrow$  volumen
3:    $m_p \rightarrow$  masa
4:    $x_p \rightarrow$  pozicija
5:    $v_p \rightarrow$  brzina
6:    $B_p \rightarrow$  za APIC prijenos
7:    $A_p \rightarrow$  inkrement sile
8: end članske varijable
9:
10: procedure DERIVACIJA_ENERGIJE()
11:   //Apstraktno, implementirati konstitutivni model za konkretni materijal
12: end procedure
13:
14: procedure AŽURIRAJ_SILU()
15:   //Apstraktno, implementirati za konkretni materijal
16: end procedure
17:
18: procedure AŽURIRAJ_DEFORMACIJSKI_GRADIJENT( $T, \Delta t$ )
19:   //Apstraktno, implementirati za konkretni materijal
20: end procedure
```

Pseudokod 2 Čvor

```
1: članske varijable
2:    $x_i \rightarrow$  pozicija
3:    $m_i \rightarrow$  masa
4:    $mv_i \rightarrow$  količina gibanja
5:    $v_i \rightarrow$  brzina nakon integracije
6:    $f_i \rightarrow$  sila
7:    $v_i^{poslije\_sudara} \rightarrow$  brzina nakon obrade sudara
8:    $v_i^{poslije\_trenja} \rightarrow$  brzina nakon obrade trenja
9: end članske varijable
10:
11: procedure OBRADI_SUDARE(objekti,  $\Delta t$ )
12:    $v_i^{poslije\_sudara} \leftarrow v_i$ 
13:   for za svaki objekt o do
14:     o.OBRADI_SUDAR( $< v_i^{poslije\_sudara} >, v_i, x_i, \Delta t$ )
15:   end for
16:   OBRADI_TRENJE( )
17: end procedure
18:
19: procedure OBRADI_TRENJE
20:    $v_i^{poslije\_trenja} \leftarrow v_i^{poslije\_sudara}$ 
21:   for za svaki objekt o s registriranim sudarom do
22:     o.OBRADI_TRENJE( $< v_i^{poslije\_trenja} >, v_i^{poslije\_sudara}, v_i, x_i$ )
23:   end for
24: end procedure
```

Pseudokod 3 Korak metode materijalne točke uz eksplicitnu integraciju

```
1: članske varijable
2:   cestice  $\rightarrow$  lista čestica
3:   resetka  $\rightarrow$  višedimenzionalni niz čvorova
4:   N  $\rightarrow$  interpolacijska krivulja
5:   objekti  $\rightarrow$  lista drugih objekata, za sudare
6:   iteracija_po_koraku  $\rightarrow$  pozitivni cijeli broj
7:    $\Delta t_{ciljani}$   $\rightarrow$  pozitivni skalar
8:   ogranici_max_ $\Delta x$   $\rightarrow$  bool zastavica
9:    $\Delta x_{max}$   $\rightarrow$  maksimalni pomak, pozitivni skalar
10: end članske varijable
11:
12: procedure KORAK( )
13:    $\Delta t_{akumulirani} \leftarrow 0$ 
14:   prag  $\leftarrow$  iteracija_po_koraku  $\times$   $\Delta t_{ciljani}$ 
15:   for (i  $\leftarrow 0$ ; (ogranici_max_ $\Delta x$  and  $\Delta t_{akumulirani} < prag$ ) or i  $<$  iteracija_po_koraku
     ; i  $\leftarrow i + 1$ ) do
16:     PRIJENOS_ČESTICA_NA_REŠETKU( )
17:      $\Delta t_{novi} \leftarrow$  ODREDI_NOVI_VREMENSKI_KORAK( )
18:     AŽURIRANJE_STANJA_REŠETKE( $\Delta t_{novi}$ )
19:     PRIJENOS_REŠETKE_NA_ČESTICE( )
20:     AŽURIRANJE_STANJA_ČESTICA( $\Delta t_{novi}$ )
21:     RESETIRAJ_REŠETKU( )
22:      $\Delta t_{akumulirani} \leftarrow \Delta t_{akumulirani} + \Delta t_{novi}$ 
23:   end for
24: end procedure
25:
26: procedure PRIJENOS_ČESTICA_NA_REŠETKU( )
27:   for svaku česticu p do
28:     p.AŽURIRAJ_SILU( )
29:     for svaki čvor i u susjedstvu čestice p do
30:        $m_i \leftarrow m_i + w_{ip}m_p$ 
31:        $m\mathbf{v}_i \leftarrow m\mathbf{v}_i + w_{ip}m_p(\mathbf{v}_p + \mathbf{D}_p^{-1}\mathbf{B}_p(\mathbf{x}_i - \mathbf{x}_p))$ 
32:        $\mathbf{f}_i \leftarrow \mathbf{f}_i + w_{ip}\mathbf{A}_p$ 
33:     end for
```

```

34:   end for
35: end procedure
36:
37: procedure ODREDI_NOVI_VREMENSKI_KORAK()
38:   if  $\neg \text{ogranci\_max\_}\Delta x$  then
39:     return  $\Delta t_{ciljani}$ 
40:   end if
41:    $\Delta x_m \leftarrow 0$ 
42:   for svaki čvor  $i$  do
43:     if  $m_i = 0$  then
44:       preskoci cvor  $i$ 
45:     end if
46:      $\mathbf{v}_i \leftarrow m\mathbf{v}_i/m_i$ 
47:      $\mathbf{a}_i \leftarrow -\mathbf{f}_i/m_i + \text{gravitacija}$ 
48:      $\mathbf{v}_i \leftarrow \mathbf{v}_i + \mathbf{a}_i\Delta t_{ciljani}$ 
49:     i.OBRADI_SUDARE(objekti,  $\Delta t_{ciljani}$ )
50:      $\Delta \mathbf{x} \leftarrow \Delta t_{ciljani} \mathbf{v}_i^{\text{poslije\_sudara}}$ 
51:      $\Delta x_{norma} \leftarrow \|\Delta \mathbf{x}\|$ 
52:     if  $\Delta x_{norma} > \Delta x_m$  then
53:        $\Delta x_m \leftarrow \Delta x_{norma}$ 
54:     end if
55:   end for
56:   return  $\Delta t_{ciljani} \frac{\Delta x_{max}}{\Delta x_m}$ 
57: end procedure
58:
59: procedure AŽURIRANJE_STANJA_REŠETKE( $\Delta t$ )
60:   for svaki čvor  $i$  do
61:     if  $m_i = 0$  then
62:       preskoči čvor  $i$ 
63:     end if
64:      $\mathbf{v}_i \leftarrow m\mathbf{v}_i/m_i$ 
65:      $\mathbf{a}_i \leftarrow -\mathbf{f}_i/m_i + \text{gravitacija}$ 
66:      $\mathbf{v}_i \leftarrow \mathbf{v}_i + \mathbf{a}_i\Delta t$ 
67:     i.OBRADI_SUDARE(objekti,  $\Delta t$ )
68:   end for
69: end procedure

```

```

70:
71: procedure PRIJENOS_REŠETKE_NA_ČESTICE( )
72:   for svaku česticu p do
73:      $\mathbf{v}_p \leftarrow \mathbf{0}$ 
74:      $\mathbf{B}_p \leftarrow \text{Matrica\_Nula}(d, d)$  ▷ d = dimenzija
75:     for svaki čvor i u susjedstvu čestice p do
76:        $\mathbf{v}_p \leftarrow \mathbf{v}_p + w_{ip} \mathbf{v}_i^{\text{poslije\_trenja}}$ 
77:        $\mathbf{B}_p \leftarrow \mathbf{B}_p + \text{VANJSKI\_PRODUKT}(\mathbf{v}_i^{\text{poslije\_trenja}}, \nabla w_{ip})$ 
78:     end for
79:   end for
80: end procedure
81:
82: procedure AŽURIRANJE_STANJA_ČESTICA( $\Delta t$ )
83:   for svaku česticu p do
84:      $\mathbf{x}_p \leftarrow \mathbf{0}$ 
85:      $\mathbf{T} \leftarrow \text{Matrica\_Nula}(d, d)$  ▷ d = dimenzija
86:     for svaki čvor i u susjedstvu čestice p do
87:        $\mathbf{x}_p \leftarrow \mathbf{x}_p + w_{ip}(\mathbf{x}_i + \Delta t w_{ip} \mathbf{v}_i^{\text{poslije\_sudara}})$ 
88:        $\mathbf{T} \leftarrow \mathbf{T} + \text{VANJSKI\_PRODUKT}(\mathbf{v}_i^{\text{poslije\_sudara}}, \nabla w_{ip})$ 
89:     end for
90:     p.AŽURIRAJ_DEFOMACIJSKI_GRADIJENT( $\mathbf{T}, \Delta t$ )
91:   end for
92: end procedure

```

4. Modeli materijala

Kao što je prethodno objašnjeno, implementacija nekog materijala svodi se na definiciju konstitutivnog modela $\frac{\partial \Psi}{\partial \mathbf{F}}(\mathbf{F})$ i konstitutivnih parametara. U ovom poglavlju su izneseni modeli

- vode (Pradhana et al. (2017)),
- hiperelastičnog materijala (Jiang et al. (2016)),
- snijega (Stomakhin et al. (2013)) i
- pijeska (Klár et al. (2016), Pradhana et al. (2017)),

s odgovarajućim pseudokodom za implementaciju. Promatraju se modeli za implementaciju uz polu-implicitni Eulerov postupak. Implicitni modeli su složeniji i zahtijevaju dodatne korake.

4.1. Model vode

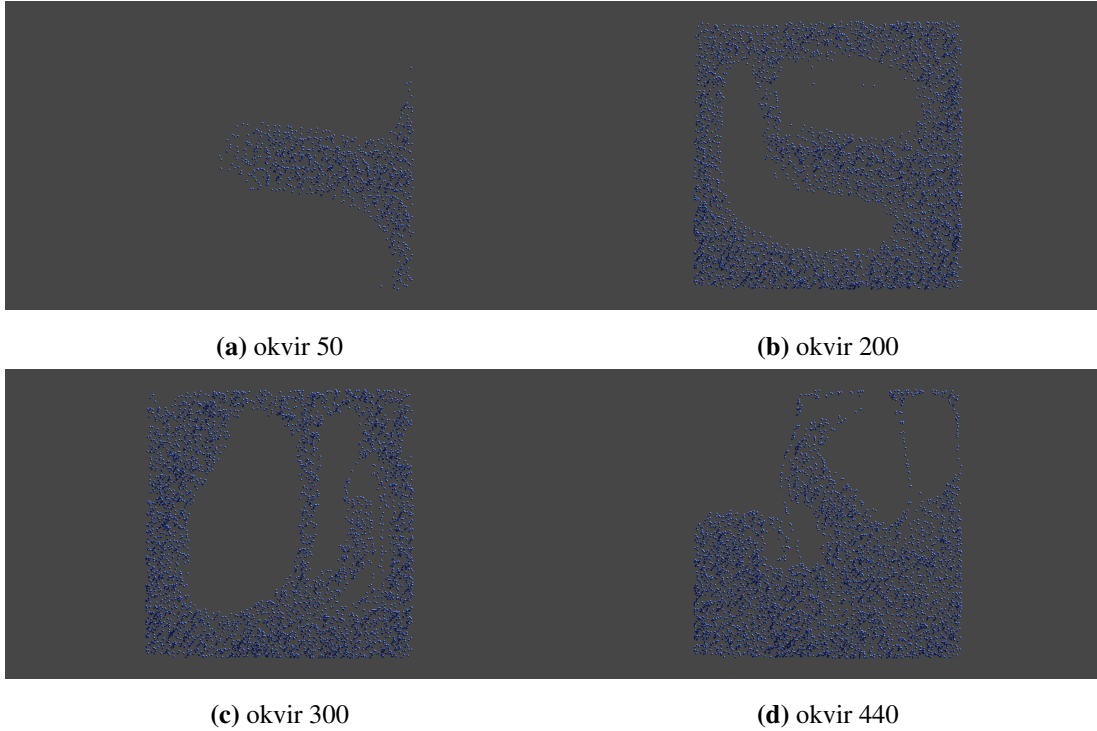
Voda se modelira kao gotovo nestlačiva, preko energije

$$\Psi(J) = -k \left(\frac{J^{1-\gamma}}{1-\gamma} - J \right). \quad (4.1)$$

Njezina derivacija je

$$\frac{\partial \Psi}{\partial J}(J) = -k \left(\frac{1}{J^\gamma} - 1 \right). \quad (4.2)$$

Deformacijski gradijent nije izravno dio modela pa ga ne treba spremati cijelog, nego je dovoljna samo njegova determinanta. Tlak vode izravno je povezan s potencijalnom energijom izrazom $\mathbf{p} = -\frac{\partial \Psi}{\partial J}(J)$. Tlak je dizajniran tako da kruto kažnjava promjenu volumena vode, odnosno promjenu J . k je modul stlačivosti vode, a γ element koji više kruto kažnjava velika odstupanja od nestlačivosti $J = 1$. Implementaciju opisuje pseudokod 4. Konstitutivni parametri nisu strogo zadani nego se prilagođavaju potrebi, najčešće radi postizanja nekakvog umjetničkog efekta. Isto vrijedi za ostale materijale. Slika 4.1 prikazuje tok vode u dvije dimenzije.



Slika 4.1: Tok vode u dvije dimenzije.

4.1.1. Pseudokod čestice vode

Pseudokod 4 Vodena čestica

```
1: nasljeđuje Česticu
2:
3: članske varijable
4:    $J \rightarrow$  determinanta deformacijskog gradijenta
5:    $k \rightarrow$  modul stlačivosti
6:    $\gamma \rightarrow$  jačina otpora prema promjeni volumena
7: end članske varijable
8:
9: procedure DERIVACIJA_ENERGIJE( )
10:   return  $-k\left(\frac{1}{J^\gamma} - 1\right)$ 
11: end procedure
12:
13: procedure AŽURIRAJ_SILU( )
14:    $\mathbf{A}_p \leftarrow \text{DERIVACIJA\_ENERGIJE}() \times V_p \times \mathbf{J} \times \mathbf{I}$ 
15: end procedure
16:
17: procedure AŽURIRAJ_DEFORMACIJSKI_GRADIJENT( $\mathbf{T}, \Delta t$ )
18:    $J \leftarrow (1 + \Delta t \operatorname{tr}(\mathbf{T}))J$ 
19: end procedure
```

4.2. Modeli hiperelastičnog materijala

Elastični materijali imaju linearan odnos stresa i naprežanja. Takvi modeli su precizni u slučaju drveta i metala, gdje su deformacije malene. Hiperelastični materijal koristi složeniju relaciju stresa i naprežanja, funkciju gustoće energije, kako bi se mogao modelirati nelinearan odnos i velike deformacije. Za definiranje modela koristi se polarni i singularni rastav $\mathbf{F} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. $\mathbf{U}$ i $\mathbf{V}$ se tretiraju kao rotacije i polarna dekompozicija $\mathbf{F} = \mathbf{R}\mathbf{S}$ se može izvesti kao $\mathbf{R} = \mathbf{U}\mathbf{V}^\top$ i $\mathbf{S} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^\top$, gdje je $\mathbf{R}$ rotacija najbliža matrici $\mathbf{F}$ a $\mathbf{S}$ je simetrična matrica.

4.2.1. Neo-Hookeov model

Neo-Hookeov model je često korišten nelinearni model hiperelastičnosti. Funkcija elastične energije dana je formulom

$$\Psi(\mathbf{F}) = \frac{\mu}{2} \left(\text{tr}(\mathbf{F}^\top \mathbf{F}) - d \right) - \mu \log(J) + \frac{\lambda}{2} \log^2(J), \quad (4.3)$$

gdje je $d = 2$ ili 3 dimenzija problema, a λ i μ su Laméovi parametri, povezani s Youngovim modulom elastičnosti E i Poissonovim omjerom ν relacijama

$$\mu = \frac{E}{2(1 + \nu)}, \lambda = \frac{E\nu}{(1 + \nu)(1 - 2\nu)}. \quad (4.4)$$

Lako je primijetiti da svaki od članova postaje 0 u slučaju krutog gibanja (translacija i rotacija), a za ne-invertirani $\mathbf{F}$ (pozitivna determinanta, $J > 0$) vrijedi $\Psi(\mathbf{F}) \geq 0$. Interpretacija Neo-Hookeovog modela je generalizacija opruge na više dimenzija, gdje Laméovi paramteri određuju krutost opruge. Odziv na deformaciju je jednak

$$\frac{\partial \Psi}{\partial \mathbf{F}}(\mathbf{F}) = \mu(\mathbf{F} - \mathbf{F}^\top) + \lambda \log(J) \mathbf{F}^{-\top}. \quad (4.5)$$

4.2.2. Fiksni korotacijski model

Drugi jednostavan i široko korišten model je fiksni korotacijski model (eng. *fixed corotational model*). Definira se preko singularne dekompozicije $\mathbf{F} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. Zove se fiksnim jer se radi o modifikaciji drugog modela, modela korotacijske linearne elastičnosti (eng. *corotational linear elasticity*). Elastična energija je dana izrazom

$$\Psi(\mathbf{F}) = \hat{\Psi}(\mathbf{\Sigma}(\mathbf{F})) = \mu \sum_{i=1}^d (\sigma_i - 1)^2 + \frac{\lambda}{2} (J - 1)^2, \quad (4.6)$$

gdje je $J = \prod_{i=1}^d \sigma_i$. Raspisivanjem izraza te korištenjem singularnog i polarnog rastava, dobiva se izraz

$$\Psi(\mathbf{F}) = \mu \|\mathbf{F} - \mathbf{R}\|_F^2 + \frac{\lambda}{2} (J - 1)^2, \quad (4.7)$$

čija je derivacija

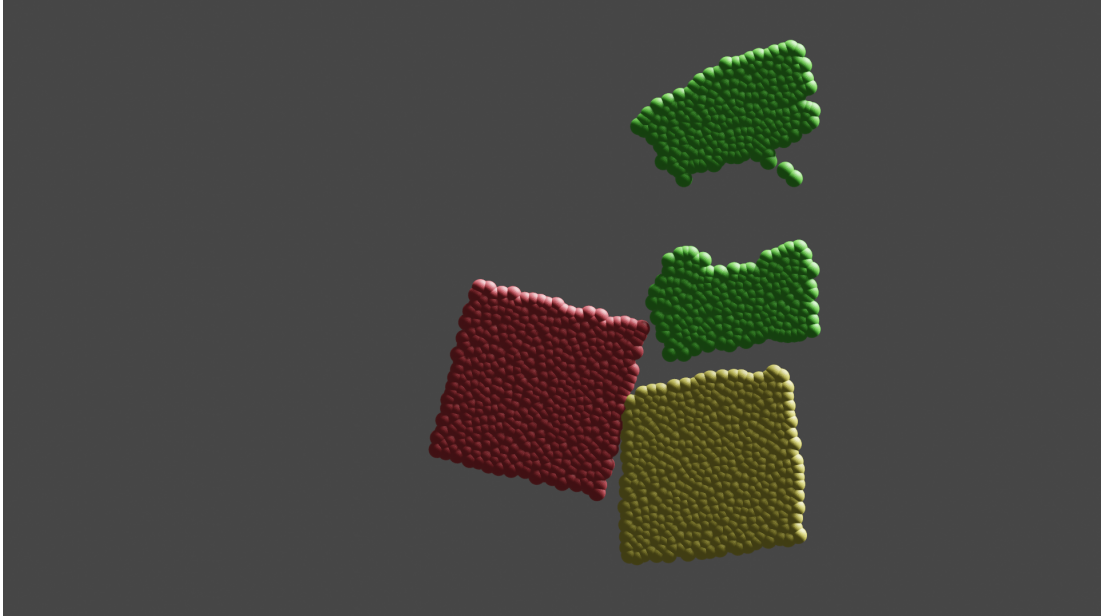
$$\frac{\partial \Psi}{\partial \mathbf{F}}(\mathbf{F}) = 2\mu(\mathbf{F} - \mathbf{R}) + \lambda(J - 1)J\mathbf{F}^{-\top}. \quad (4.8)$$

Očito je da za kruta gibanja vrijedi $\Psi(\mathbf{F}) = 0$. Pseudokod 5 pokazuje kako implementirati ovaj materijal uz fiksni korotacijski model te singularni i polarni rastav koji su pojašnjeni na početku poglavlja. Slika 4.2 prikazuje rezultate sudara više hipere-elastičnih objekata. Crveno tijelo ima 10 puta veće Laméove parametre u odnosu na žuto tijelo, a zeleno tijelo ima 10 puta manje Laméove parametre. Kod zelenog tijela vidljive su velike deformacije i odvajanje čestica, a crveno tijelo je gotovo kruto.



(a) okvir 1

(b) okvir 20



(c) okvir 40

Slika 4.2: Hiperelastični materijal u dvije dimenzije.

4.2.3. Pseudokod hiperelastične čestice

Pseudokod 5 Hiperelastična čestica

```
1: nasljeđuje Česticu
2:
3: članske varijable
4:    $\mathbf{F}_E \rightarrow$  deformacijski gradijent
5:    $\lambda, \mu \rightarrow$  trenutni Laméovi parametri
6:    $\lambda_0, \mu_0 \rightarrow$  početni Laméovi parametri
7: end članske varijable
8:
9: procedure DERIVACIJA_ENERGIJE( )
10:    $J_E \leftarrow \det(\mathbf{F}_E)$ 
11:    $(\mathbf{R}_E, \mathbf{S}_E) \leftarrow \text{POLARNA\_DEKOMPOZICIJA}(\mathbf{F}_E)$ 
12:   return  $2\mu(\mathbf{F}_E - \mathbf{R}_E) + \lambda(J_E - 1)J_E\mathbf{F}_E^{-\top}$ 
13: end procedure
14:
15: procedure AŽURIRAJ_SILU( )
16:    $\mathbf{A}_p \leftarrow \text{DERIVACIJA\_ENERGIJE}() \times V_p \times \mathbf{F}_E^\top$ 
17: end procedure
18:
19: procedure AŽURIRAJ_DEFORMACIJSKI_GRADIJENT( $\mathbf{T}, \Delta t$ )
20:    $\mathbf{F}_E \leftarrow (\mathbf{I} + \Delta t \mathbf{T}) \mathbf{F}_E$ 
21: end procedure
```

4.3. Model snijega

Dinamika snijega je poprilično raznolika pa time i složena. Tradicionalno su se za snijeg koristile specijalizirane metode i simulatori fluida čiji rezultati nalikuju na čvrste materijale. Model opisan u ovome poglavlju obuhvaća čitav spektar ponašanja snijega i uspješno je korišten u animiranim filmovima.

Izraz za energiju gotovo je identičan onome iz poglavlja 4.2.2

$$\Psi(\mathbf{F}_E, \mathbf{F}_P) = \mu(\mathbf{F}_P) \|\mathbf{F}_E - \mathbf{R}_E\|_F^2 + \frac{\lambda(\mathbf{F}_P)}{2} (J_E - 1)^2. \quad (4.9)$$

Razlika je u tome što su λ i μ funkcije plastičnog dijela deformacijskog gradijenta

$$\mu(\mathbf{F}_P) = \mu_0 \exp(\xi(1 - J_P)), \lambda(\mathbf{F}_P) = \lambda_0 \exp(\xi(1 - J_P)), \quad (4.10)$$

gdje je ξ faktor stvrdnjavanja. Konstitutivni model glasi

$$\frac{\partial \Psi}{\partial \mathbf{F}_E}(\mathbf{F}_E, \mathbf{F}_P) = 2\mu(\mathbf{F}_P)(\mathbf{F}_E - \mathbf{R}_E) + \lambda(\mathbf{F}_P)(J_E - 1)J_E \mathbf{F}_E^{-\top}. \quad (4.11)$$

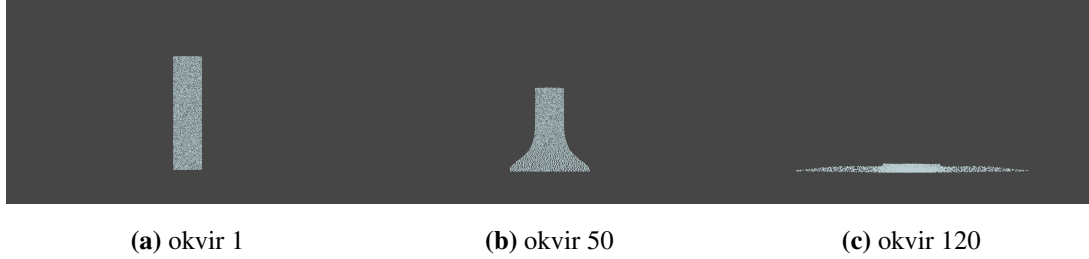
Kod ažuriranja deformacijskog gradijenta, mora se paziti na elastični i plastični dio. Parametri kritične kompresije θ_C i kritičnog rastezanja θ_S definiraju kada se materijal počinje plastično deformirati. Singularne vrijednosti $\mathbf{F}_E$ se ograničavaju na interval $[1 - \theta_C, 1 + \theta_S]$. Ovo utječe i na svojstva materijala preko izraza 4.10.

Za simuliranje različitih vrsta snijega koristi se sljedeća intuicija. θ_C i θ_S definiraju kada se materijal počinje raspadati, veći = grudast, manji = praškasti snijeg. Faktor stvrdnjavanja definira koliko brzo se materijal lomi jednom kada postane plastičan. *Suh* i *praškast* snijeg ima manje parametre kritične kompresije i kritičnog rastezanja, a *mokri* i *grudasti* veće. *Ledeni* snijeg ima veći faktor stvrdnjavanja, a manja vrijednost daje ponašanje nalik na *blato*.

Bitan je i inicijalni broj čestica po ćeliji u početnoj konfiguraciji, 4 do 10 prema Stomakhin et al. (2013). Zbog osjetljivosti projekcije, bitan je i dovoljno malen vremenski korak, 10^{-4} do 10^{-5} sekundi.

Ažuriranje deformacijskog gradijenta počinje definiranjem privremenog elastičnog dijela deformacijskog gradijenta $\hat{\mathbf{F}}_{Ep}^{n+1} = (\mathbf{I} + \Delta t \nabla \mathbf{v}_p^{n+1}) \mathbf{F}_{Ep}^n$, a za plastični dio se uzima $\hat{\mathbf{F}}_{Pp}^{n+1} = \mathbf{F}_{Pp}^n$ i time vrijedi jednakost 3.6 za vrijednosti deformacijskog gradijenta u koraku $n + 1$. Idući je operator projekcije, odnosno popravljavanje rezultata. Računa se singularni rastav $\hat{\mathbf{F}}_{Ep}^{n+1} = \mathbf{U} \Sigma \mathbf{V}^\top$ i nove singularne vrijednosti $\Sigma_{projicirani} = \text{ograniči_na_interval}(\Sigma, [1 - \theta_C, 1 + \theta_S])$. Konačni rezultati dani su formulom $\mathbf{F}_{Ep}^{n+1} = \mathbf{U} \Sigma_{projicirani} \mathbf{V}^\top$ i $\mathbf{F}_{Pp}^{n+1} = (\mathbf{F}_{Ep}^{n+1})^{-1} \mathbf{F}_{Pp}^n$.

Moguće je primijetiti da u implementaciji nije potreban cijeli plastični dio deformacijskog gradijenta nego samo njegova determinanta jer konstitutivni parametri ovise samo o njoj. Pseudokod 6 implementira model snijega uz gornju optimizaciju. slika 4.3 prikazuje raspad stupca snijega.



Slika 4.3: Raspad stupca snijega u dvije dimenzije.

4.3.1. Pseudokod čestice snijega

Pseudokod 6 Čestica snijega

```

1: nasljeđuje Česticu
2:
3: članske varijable
4:    $\mathbf{F}_E, \mathbf{F}_E^{privremeni} \rightarrow$  elastični dio deformacijskog gradijenta
5:    $J_E, J_P \rightarrow$  determinante deformacijskog gradijenta
6:    $\lambda, \mu \rightarrow$  trenutni Laméovi parametri
7:    $\lambda_0, \mu_0 \rightarrow$  početni Laméovi parametri
8:    $\xi, \theta_C, \theta_S \rightarrow$  parametri plastičnosti
9: end članske varijable
10:
11: procedure DERIVACIJA_ENERGIJE( )
12:    $J_E \leftarrow \det(\mathbf{F}_E)$ 
13:    $(\mathbf{R}_E, \mathbf{S}_E) \leftarrow \text{POLARNA\_DEKOMPOZICIJA}(\mathbf{F}_E)$ 
14:   return  $2\mu(\mathbf{F}_E - \mathbf{R}_E) + \lambda(J_E - 1)J_E\mathbf{F}_E^{-\top}$ 
15: end procedure
16:
17: procedure AŽURIRAJ_SILU( )
18:    $\mathbf{A}_p \leftarrow \text{DERIVACIJA\_ENERGIJE}() \times V_p \times \mathbf{F}_E^\top$ 
19: end procedure
20:

```

```

21: procedure AŽURIRAJ_DEFORMACIJSKI_GRADIJENT( $\mathbf{T}$ ,  $\Delta t$ )
22:    $\mathbf{F}_E^{privremeni} \leftarrow (\mathbf{I} + \Delta t \mathbf{T}) \mathbf{F}_E$ 
23:   PLASTIČNOST( )
24: end procedure
25:
26: procedure PLASTIČNOST( )
27:    $(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}) \leftarrow \text{SVD}(\mathbf{F}_E^{privremeni})$ 
28:    $\mathbf{\Sigma}^{projicirani} \leftarrow \text{OGRANIČI\_VRIJEDNOSTI\_NA\_INTERVAL}(\mathbf{\Sigma}, 1 - \theta_C, 1 + \theta_S)$ 
29:    $\mathbf{F}_E \leftarrow \mathbf{U} \mathbf{\Sigma}^{projicirani} \mathbf{V}^\top$ 
30:    $J_E^{prethodni} \leftarrow \det(\mathbf{F}_E^{privremeni})$ 
31:    $J_E \leftarrow \det(\mathbf{F}_E)$ 
32:    $J_P \leftarrow J_P \frac{J_E^{prethodni}}{J_E}$ 
33:   STVRDNJAVANJE( )
34: end procedure
35:
36: procedure STVRDNJAVANJE( )
37:    $faktor \leftarrow \exp(\xi(1 - J_P))$ 
38:    $\lambda \leftarrow \lambda_0 \cdot faktor$ 
39:    $\mu \leftarrow \mu_0 \cdot faktor$ 
40: end procedure

```

4.4. Model pijeska

Dinamika pijeska slična je fluidu i krutom tijelu. Pijesak može zauzeti oblik spremnika u kojem se nalazi ali može podržavati težinu drugog objekta i formirati hrpe. Cilj je rekreirati karakteristični tok čestica pijeska i njihovo grupiranje. Ovako složeno ponašanje izuzetno je teško modelirati. Problem dodatno otežava zahtijevanje ogromnog broja sitnih čestica. Prema (Klár et al. (2016)) to može biti 2 do 32 čestice po ćeliji rešetke, a ukupni broj čestica može dosežati 10^6 do 10^7 . U ovome konkretnome modelu zahtijeva se i izuzetno malen maksimalni vremenski korak u slučaju eksplisitne integracije. Obično se koristi tehnika adaptivnog vremenskog koraka, koja kažnjava velike brzine.

Potencijalna energija dana je izrazom

$$\Psi(\mathbf{F}) = \mu \operatorname{tr}((\ln \Sigma)^2) + \frac{1}{2} \lambda (\operatorname{tr}(\ln \Sigma))^2, \quad (4.12)$$

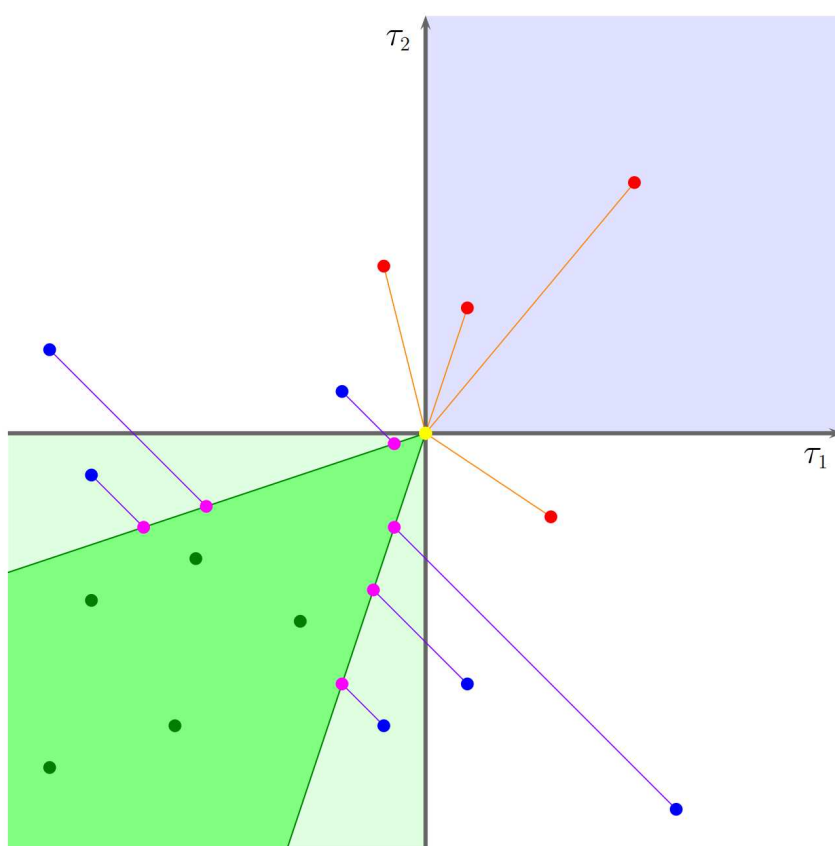
a njezina derivacija u odnosu na deformacijski gradijent izrazom

$$\frac{\partial \Psi}{\partial \mathbf{F}}(\mathbf{F}) = \mathbf{U}(2\mu \Sigma^{-1} \ln \Sigma + \lambda \operatorname{tr}(\ln \Sigma) \Sigma^{-1}) \mathbf{V}^\top. \quad (4.13)$$

Ažuriranje deformacijskog gradijenta počinje jednako kao kod snijega u poglavlju 4.3, definiranjem privremenog elastičnog dijela deformacijskog gradijenta $\hat{\mathbf{F}}_{Ep}^{n+1} = (\mathbf{I} + \Delta t \nabla \mathbf{v}_p^{n+1}) \mathbf{F}_{Ep}^n$, a za plastični dio se uzima $\hat{\mathbf{F}}_{Pp}^{n+1} = \mathbf{F}_{Pp}^n$ i time vrijedi jednakost 3.6 za vrijednosti deformacijskog gradijenta u koraku $n + 1$. Idući je operator projekcije, odnosno popravljavanje rezultata. Računa se singularni rastav $\hat{\mathbf{F}}_{Ep}^{n+1} = \mathbf{U} \Sigma \mathbf{V}^\top$ i nove singularne vrijednosti $(\Sigma_{projicirani}, \Delta q) \leftarrow projekcija(\Sigma, \alpha_p)$. Konačni rezultati dani su formulom $\mathbf{F}_{Ep}^{n+1} = \mathbf{U} \Sigma_{projicirani} \mathbf{V}^\top$ i $\mathbf{F}_{Pp}^{n+1} = (\mathbf{F}_{Ep}^{n+1})^{-1} \mathbf{F}_{Pp}^n$. Na samom kraju materijal se stvrdnjava, *stvrdnjavanje* (Δq) . Operacije projekcije i stvrdnjavanja te njihovi parametri detaljnije su opisani u poglavljima 4.4.1 i 4.4.2.

4.4.1. Projekcija na Drucker-Prager stožac

Drucker-Prager model plastičnosti zasniva se na Coulombovom modelu trenja između čestica pijeska. Slika 4.4 prikazuje projekciju u dvije dimenzije. U tri dimenzije dozvoljeno područje (tamno zeleno) ima oblik stošca. Na osima su logaritamske vrijednosti singularnih vrijednosti elastičnog dijela deformacijskog gradijenta. U slučaju da se čestica nalazi unutar dozvoljenog područja, projekcija ne radi ništa (Slučaj 1). Točke se moraju projicirati prema stošcu najkraćim mogućim putem. Ako postoji normala od plašta stošca prema čestici, obavlja se projekcija na plašt stošca (Slučaj 3). Inače se točke projiciraju u vrh stošca (Slučaj 2). Parametar projekcije je α_p .



Slika 4.4: Projekcija na Drucker-Prager stožac u dvije dimenzije. Izvor: Klár et al. (2016)

Projekcija se definira uz singularni rastav $\mathbf{F}_{Ep} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. Neka je $\boldsymbol{\epsilon} = \ln \mathbf{\Sigma}$ i

$$\hat{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon} - \frac{\text{tr}(\boldsymbol{\epsilon})}{d} \mathbf{I}, \delta\gamma = \|\hat{\boldsymbol{\epsilon}}\|_F + \frac{d\lambda + 2\mu}{2\mu} \text{tr}(\boldsymbol{\epsilon})\alpha, \quad (4.14)$$

gdje je d dimenzija problema, a $\delta\gamma$ količina plastične deformacije. Ako je $\delta\gamma \leq 0$ onda je $\mathbf{F}_{Ep}$ već unutar stošca i projekcija vraća $\mathbf{\Sigma}$. Ako je $\|\hat{\boldsymbol{\epsilon}}\|_F = 0$ ili $\text{tr}(\boldsymbol{\epsilon})$ onda se mora projicirati u vrh stošca i vraća se $\mathbf{I}$. Inače se projicira na plašt stošca i vraća se $\exp(\mathbf{H})$, gdje je

$$\mathbf{H} = \boldsymbol{\epsilon} - \delta\gamma \frac{\hat{\boldsymbol{\epsilon}}}{\|\hat{\boldsymbol{\epsilon}}\|_F} \quad (4.15)$$

4.4.2. Stvrdnjavanje pijeska prema Drucker-Prager modelu

Model stvrdnjavanja je takav da plastična deformacija povećava trenje između čestica pijeska. Količina stvrdnjavanja ovisi o količini korekcije prilikom projekcije. U slučaju 1 nije bilo plastičnosti i $\Delta q = 0$. U slučaju 2, sav stres je uklonjen pa je $\Delta q = \|\boldsymbol{\epsilon}\|_F$. U slučaju 3, količina plastičnosti je $\Delta q = \delta\gamma$. U svakom slučaju vrijedi $\Delta q \geq 0$. Ažuriranje stvrdnjavanja dalje se definira preko izraza

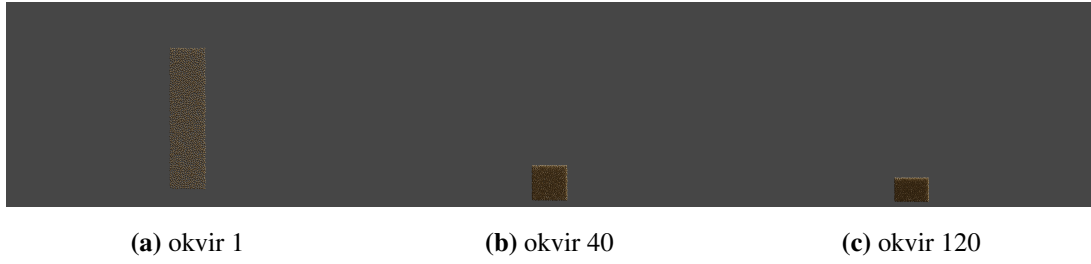
$$q_p^{n+1} = q_p^n + \Delta q \quad (4.16)$$

$$\phi = h_0 + (h_1 q_p^{n+1} - h_3) \exp(-h_2 q_p^{n+1}) \quad (4.17)$$

$$\alpha_p^{n+1} = \sqrt{\frac{2}{3}} \frac{2 \sin \phi}{3 - \sin \phi} \quad (4.18)$$

q_p^n je stanje tvrdoće, ϕ se naziva i kutem trenja, a unutarnji koeficijent trenja je $\tan \phi$ i 4.17 modelira monotono rastuću funkciju s asimptotskim maksimumom. Moguće vrijednosti ϕ su u intervalu $[0, \frac{\pi}{2})$, a $\phi = 0$ daje ponašanje fluida. Moguće vrijednosti parametara stvrdnjavanja zadovoljavaju uvjete $h_0 > h_3 \geq 0$ i $h_1, h_2 \geq 0$.

Moguće je primijetiti da za implementaciju ne treba plastični dio gradijenta niti njegova determinanta, već je sva plastičnost definirana preko q i α . Početna vrijednost q^0 je 0, a početna vrijednost α^0 se dobiva stvrdnjavanjem uz inkrement $\Delta q = 0$. Pseudokod 7 daje opisanu implementaciju uz dio o korekciji volumena iz (Klár et al. (2016)). Uvodi se dodatan parametar V_c koji prati logaritamsku promjenu volumena i dodatno utječe na projekciju. Slika 4.5 prikazuje rezultate dobivene simuliranjem stupca pijeska. Inicijalno dolazi do velike kompresije i stvrdnjavanja pijeska. Čestice kontinuirano gube volumen, ali to se događa i u referentnoj implementaciji (Klár et al. (2016)) zbog asimptotskog maksimuma u izrazu 4.17.



Slika 4.5: Stup pijeska u dvije dimenzije.

4.4.3. Pseudokod čestice pijeska

Pseudokod 7 Čestica pijeska

```

1: nasljeđuje Česticu
2:
3: članske varijable
4:    $\mathbf{F}_E, \mathbf{F}_E^{privremeni} \rightarrow$  elastični dio deformacijskog gradijenta
5:    $\lambda, \mu \rightarrow$  Laméovi parametri
6:    $q, \alpha \rightarrow$  parametri plastičnosti
7:    $h_0, h_1, h_2, h_3 \rightarrow$  parametri plastičnosti, u radijanima
8:    $V_c \rightarrow$  praćenje logaritamske promjene volumena
9: end članske varijable
10:
11: procedure DERIVACIJA_ENERGIJE( )
12:    $(\mathbf{U}, \Sigma, \mathbf{V}) \leftarrow \text{SVD}(\mathbf{F}_E)$ 
13:   return  $\mathbf{U}(2\mu\Sigma^{-1} \ln \Sigma + \lambda \text{tr}(\ln \Sigma)\Sigma^{-1})\mathbf{V}^\top$ 
14: end procedure
15:
16: procedure AŽURIRAJ_SILU( )
17:    $\mathbf{A}_p \leftarrow \text{DERIVACIJA_ENERGIJE}() \times V_p \times \mathbf{F}_E^\top$ 
18: end procedure
19:
20: procedure AŽURIRAJ_DEFORMACIJSKI_GRADIJENT( $\mathbf{T}, \Delta t$ )
21:    $\mathbf{F}_E^{privremeni} \leftarrow (\mathbf{I} + \Delta t \mathbf{T})\mathbf{F}_E$ 
22:   PLASTIČNOST( )
23: end procedure
24:
25: procedure PLASTIČNOST( )
26:    $(\mathbf{U}, \Sigma, \mathbf{V}) \leftarrow \text{SVD}(\mathbf{F}_E^{privremeni})$ 

```

```

27:   $(\Sigma^{projicirani}, \Delta q) \leftarrow \text{PROJEKCIJA}(\Sigma)$ 
28:   $\mathbf{F}_E \leftarrow \mathbf{U} \Sigma^{projicirani} \mathbf{V}^\top$ 
29:   $J_E^{prethodni} \leftarrow \det(\mathbf{F}_E^{privremeni})$ 
30:   $J_E \leftarrow \det(\mathbf{F}_E)$ 
31:   $V_c \leftarrow V_c - \ln(J_E J_E^{prethodni})$ 
32:   $\text{STVRDNJAVANJE}(\Delta q)$ 
33: end procedure
34:
35: procedure PROJEKCIJA( $\Sigma$ )
36:    $\epsilon \leftarrow \ln \Sigma + \frac{V_c}{d} \mathbf{I}$ 
37:    $\hat{\epsilon} \leftarrow \epsilon - \frac{\text{tr}(\epsilon)}{d} \mathbf{I}$ 
38:   if  $\|\hat{\epsilon}\|_F^2 = 0$  or  $\text{tr}(\epsilon)$  then
39:     return  $(\mathbf{I}, \|\epsilon\|_F)$ 
40:   end if
41:    $\delta\gamma \leftarrow \|\hat{\epsilon}\|_F + \frac{d\lambda+2\mu}{2\mu} \text{tr}(\epsilon)\alpha$ 
42:   if  $\delta\gamma \leq 0$  then
43:     return  $(\Sigma, 0)$ 
44:   end if
45:    $\mathbf{H} \leftarrow \epsilon - \delta\gamma \frac{\hat{\epsilon}}{\|\hat{\epsilon}\|_F}$ 
46:   return  $(\exp(\mathbf{H}), \delta\gamma)$ 
47: end procedure
48:
49: procedure STVRDNJAVANJE( $\Delta q$ )
50:    $q \leftarrow q + \Delta q$ 
51:    $\phi \leftarrow h_0 + (h_1 q - h_3) \exp(-h_2 q)$ 
52:    $\alpha \leftarrow \sqrt{\frac{2}{3} \frac{2 \sin \phi}{3 - \sin \phi}}$ 
53: end procedure

```

▷ d = dimenzija

5. Sudari s krutim tijelima

Rezolucija sudara podijeljena je u dva dijela - prvo se detektira sudar i korigira brzina, nakon čega se primjenjuje trenje. Odvojenost je bitna za implicitne postupke, gdje je teže primijeniti trenje. Za eksplicitne postupke odvojenost nije bitna ali je dobra praksa. Također, nužno je da se rezolucija sudara obavlja nad čvorovima rešetke, inače $\mathbf{x}_p^{n+1}$ i $\mathbf{F}_p^{n+1}$ mogu postati desinkronizirani što se manifestira laganim probijanjem čestica u krute objekte. Nakon rezolucije sudara nad čvorovima, opcionalno se može napraviti rezolucija sudara nad česticama, ali to uglavnom nije potrebno.

5.1. Funkcije udaljenosti s predznakom

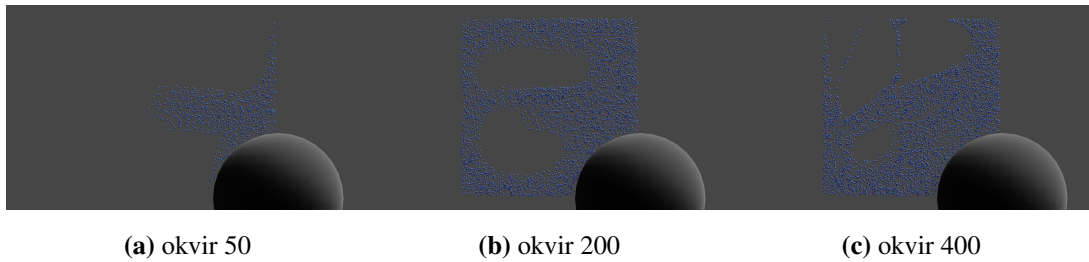
Najjednostavniji način za prikazivanje drugih objekata je korištenjem funkcija udaljenosti s predznakom $\phi(\mathbf{x})$ (eng. *signed distance functions*, SDF). Time detekcija sudara postaje trivijalna, $\phi(\mathbf{x}_i) > 0$ znači da nema sudara, a $\phi(\mathbf{x}_i) \leq 0$ da se dogodio sudar. U slučaju sudara računa se normala $\mathbf{n} = \nabla\phi(\mathbf{x}_i)$. Dakle, za reprezentaciju nekog objekta potrebna je funkcija $\phi(\mathbf{x})$ i funkcija koja vraća normalu $\nabla\phi(\mathbf{x})$ na njegovu površinu. Jednostavniji primjeri ovakvih objekata su ravnina i sfera.

Ravnina. Za definiciju ravnine potrebna je točka na ravnini $\mathbf{x}_R$ i normala ravnine $\mathbf{n}_R$. Tada je funkcija udaljenosti od ravnine s predznakom $\phi(\mathbf{x}) = \mathbf{n}_R \cdot (\mathbf{x} - \mathbf{x}_R)$, a normala na ravninu je $\nabla\phi(\mathbf{x}) = \mathbf{n}_R$.

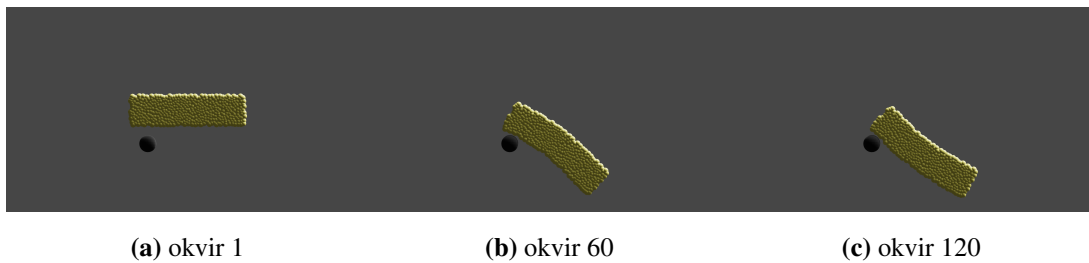
Sfera. Za definiciju sfere potrebno je njezino središte $\mathbf{x}_S$ i radijus r_S . Tada je funkcija udaljenosti od sfere s predznakom $\phi(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_S\|_F - r_S$, a normala na sferu je $\nabla\phi(\mathbf{x}) = \frac{\mathbf{x} - \mathbf{x}_S}{\|\mathbf{x} - \mathbf{x}_S\|_F}$.

Gornja dva primjera su pojednostavljeni objekti koji se ne gibaju kroz vrijeme i utječu na MPM čestice, ali čestice ne utječu na njih. Takvo ponašanje zove se jednosmjerna spreaga (eng. *one-way coupling*). Definiranje SDF objekata s masom i količinom gibanja radi postizanja dvosmjerne sprege (eng. *two-way coupling*) moguće je korištenjem tradicionalnih tehnika za simuliranje krutih, neelastičnih tijela kakve se primjerice koriste u računalnim igrama.

Slika 5.1 prikazuje utjecaj krute kugle na tok vode, jednako definiranome kao na slici 4.1. Sudar hiperelastičnog objekta i manje kugle prikazan je na slici 5.2.



Slika 5.1: Utjecaj SDF kugle na tok vode.



Slika 5.2: Sudar hiperelastičnog tijela i SDF kugle.

5.2. Uvjeti na granici

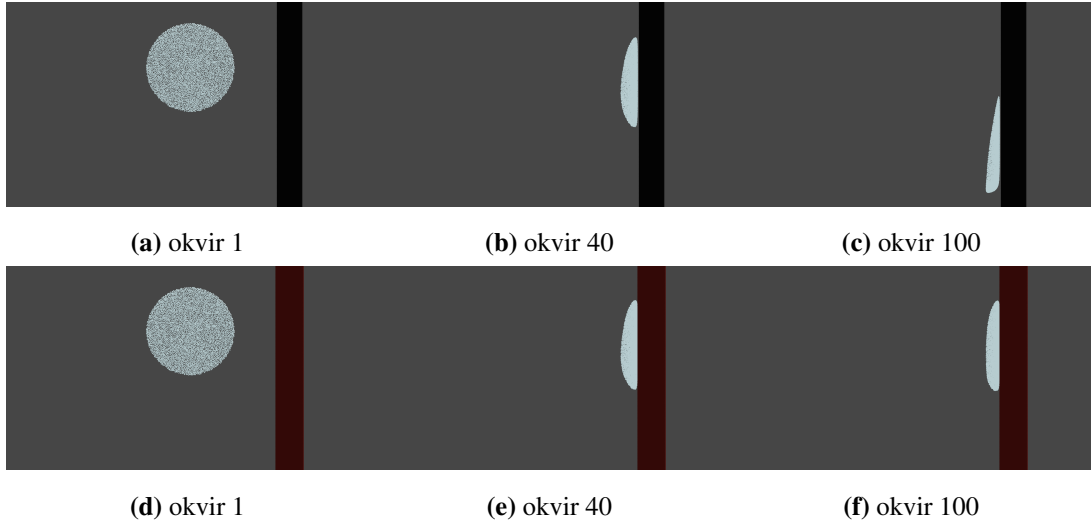
Jednom kada se detektira sudar, na njega treba reagirati tako da se zadovolje postavljena fizikalna ograničenja - uvjeti na granici. Tip površine naziva se prema ograničenjima postavljenima za tu površinu. Površina može biti ljepljiva, klizajuća i separirajuća.

Ljepljiva. Ljepljiva površina je najjednostavnija i zahtijeva da se sudareni čvor ne nastavi gibati nego ostane prlijepljen za površinu. $\mathbf{v}_i^{poslije_udara}$ postaje nul-vektor.

Klizajuća. Klizajuća površina dozvoljava gibanje po površini i nikakvu penetraciju unutar površine objekta. Brzina se mora korigirati tako da se čvor u trenutnom koraku odbije u odnosu na normalu površine.

Separirajuća. Separirajuća površina je nešto složenija od klizajuće ali funkcionira na istom principu. Čvor koji se sudari s objektom ne smije dalje penetrirati u njegovu površinu ako je već unutar nje, ali inače se može slobodno gibati.

Sva tri slučaja su implementirana u pseudokodu 8 prema (Klár et al. (2016)). Usporedba ljepljive i klizajuće površine na grudu snijega dana je na slici 5.3.



Slika 5.3: Gruda snijega uz klizajuću (gore) i ljepljivu površinu (dolje).

5.3. Trenje na površini krutog objekta

Primjena trenja ne ovisi o tipu uvjeta na granici nego o njihovim utjecajima na brzinu. U slučaju eksplicitnog postupka, $\mathbf{v}_i$ i $\mathbf{v}_i^{\text{poslije_sudara}}$ su dostupni, a $\Delta\mathbf{v}_i = \mathbf{v}_i^{\text{poslije_sudara}} - \mathbf{v}_i$ je promjena brzine uslijed sudara. Ovo je morao biti rezultat impulsa sile $\mathbf{j} = m_i \Delta\mathbf{v}_i$. Normala $\mathbf{n}$ je poznata i služi za rastav brzine na normalni dio $v_{in} = \mathbf{n} \cdot \mathbf{v}_i^{\text{poslije_sudara}}$ i tangencijalni dio $\mathbf{v}_{it} = \mathbf{v}_i^{\text{poslije_sudara}} - \mathbf{n}v_{in}$. Smjer tangente je tada $\mathbf{t} = \frac{\mathbf{v}_{it}}{\|\mathbf{v}_{it}\|}$. Coulombov zakon trenja ograničava ukupno trenje na $\mu\|\mathbf{j}\|$ gdje je μ koeficijent trenja. Ako vrijedi $\|\mathbf{v}_{it}\| \leq \frac{\mu}{m_i}\|\mathbf{j}\|$, onda trenje treba samo ukloniti tangencijalnu komponentu brzine i vrijedi $\mathbf{v}_i^{\text{poslije_trenja}} = \mathbf{n}v_{in}$. Inače vrijedi $\mathbf{v}_i^{\text{poslije_trenja}} = \mathbf{v}_i^{\text{poslije_sudara}} - \frac{\mu}{m_i}\|\mathbf{j}\|\mathbf{t}$.

Implementacija je dana u pseudokodu 8. Konkretni objekti trebaju implementirati jedino dijelove vezane uz funkcije udaljenosti s predznakom. Svi objekti koriste istu implementaciju rezolucije sudara i primjene trenja.

5.4. Pseudokod jednostavnog krutog SDF objekta

Pseudokod 8 SDF objekt

- 1: **članske varijable**
- 2: $\text{tip} \rightarrow$ tip površine
- 3: $\mu \rightarrow$ koeficijent trenja
- 4: **end članske varijable**
- 5:

```

6: procedure SDF( $x$ )
7:   //Apstraktno, implementirati za konkretni objekt
8: end procedure
9:
10: procedure SDF_NORMALA( $x$ )
11:   //Apstraktno, implementirati za konkretni objekt
12: end procedure
13:
14: procedure OBRADI_SUDAR( $\langle v^{poslije\_sudara} \rangle, v, x, \Delta t$ )
15:    $d \leftarrow \text{SDF}(x)$ 
16:   if  $d > 0$  then
17:     return
18:   end if
19:   if  $tip = ljepljiva$  then
20:      $v^{poslije\_sudara} \leftarrow 0$ 
21:     return
22:   end if
23:    $n \leftarrow \text{SDF\_NORMALA}(x)$ 
24:    $x^{sljedeci} \leftarrow x + \Delta t v$ 
25:    $d^{sljedeci} \leftarrow \text{SDF}(x^{sljedeci})$ 
26:    $\Delta d \leftarrow d^{sljedeci} - d$ 
27:   if  $tip = klizajuci$  or ( $tip = separirajuci$  and  $\Delta d < 0$ ) then
28:      $v^{poslije\_sudara} \leftarrow v - \frac{\Delta d}{\Delta t} n$ 
29:   end if
30: end procedure
31:
32: procedure OBRADI_TRENJE( $\langle v^{poslije\_trenja} \rangle, v^{poslije\_sudara}, v, x$ )
33:    $n \leftarrow \text{SDF\_NORMALA}(x)$ 
34:    $v_t \leftarrow v^{poslije\_sudara} - n(n \cdot v_i)$ 
35:    $t \leftarrow \frac{v_t}{\|v_t\|}$ 
36:    $v^{poslije\_trenja} \leftarrow v_i - \min\left(\|v_t\|, \mu\left\|v_i^{poslije\_sudara} - v_i\right\|\right)t$ 
37: end procedure

```

6. Implementacija

6.1. Simulacijski model

Simulacijski model implementiran je koristeći programski alat Unity inačice 2021.3 LTS. Unutar Unityja koristi se jezik C#. Za operacije s vektorima i matricama, uključujući odgovarajuće strukture podataka, korištena je knjižica Math.NET Numerics (Math.NET Project). Za instalaciju potrebnog paketa potreban je NuGet upravitelj paketima za Unity (McCarthy (2022)). Razvijen je jednodretveni simulacijski model.

Simulator je implementiran prema pseudokodu 3, čvor rešetke prema pseudokodu 2 i apstraktna čestica prema pseudokodu 1. Model vode implementiran je prema pseudokodu 4, a model hiperelastičnog materijala prema pseudokodu 5. Model snijega implementiran je prema pseudokodu 6, a model pijeska prema pseudokodu 7. Granice domene i kruti objekti implementirani su tako što nasljeđuju apstraktni objekt iz pseudokoda 8. Granice domene su implementirane kao ravnine, a dodani su i kruti objekti u obliku kugle.

6.2. Upute za korištenje

Program je potrebno otvoriti u odgovarajućoj verziji Unityja. Ako postoje problemi s knjižicom Math.Net Numerics, potrebno je ponovo učitati paket NuGet za Unity i kroz njegovo sučelje reinstalirati Math.NET Numerics.

U scenu je potrebno dodati prazan objekt na kojeg se dodaje skripta *MPMController.cs*, koju treba povezati s objektom *MPMParticleRenderer* koji služi za rudimentarno iscrtavanje rezultata. Dane su scene korištene za dobivanje rezultata ovog rada.

Unutar skripte, kroz grafičko sučelje postavljaju se parametri simulacije. Početni parametri su ukupno trajanje simulacije (broj okvira), putanja za spremanje rezultata, broj okvira po sekundi i broj iteracija algoritma po jednom okviru. Broj okvira po sekundi i broj iteracija po okviru impliciraju ciljani vremenski korak. Čestice je moguće

spremati po materijalu ili po simuliranom tijelu (skupu čestica) koje se dodaje u simulaciju. Implementirani su odvojeni dvodimenzionalni i trodimenzionalni simulator.

Zatim se određuje veličina domene, vrsta uvjeta na granici i koeficijent trenja. Postoji i opcija adaptivnog vremenskog koraka koja je na početku isključena. Za tu opciju potrebno je specificirati maksimalni dopušteni pomak i tada ciljani vremenski korak predstavlja najveći mogući vremenski korak jedne iteracije. Prije definiranja skupova čestica još je potrebno odrediti interpolacijsku krivulju, a početno je odabrana kubna.

Moguće je definirati i dodatne krute objekte u obliku ravnine ili kugle, zadati im poziciju, koeficijent trenja i parametre potrebne za specifični objekt. Moguće je mijenjati i konstitutivne parametre materijala. Slika 6.1 ujedno pokazuje i korištene vrijednosti konstitutivnih parametara.

▼ Material Parameters	
Water RHO	1
Water K	50
Water GAMMA	4
Water Particle Volume	1
Water Particle Mass	1
Elastic RHO	400
Elastic E	140000
Elastic V	0.2
Snow RHO	400
Snow E	140000
Snow V	0.2
Snow THETA_C	0.025
Snow THETA_S	0.0075
Snow KSI	10
Sand RHO	2200
Sand E	35370000
Sand V	0.2

Slika 6.1: Parametri materijala

Za skupove čestica potrebno je definirati ime pod kojim će se spremati rezultati, nakon čega slijede parametri kao što su indeks okvira kada se skup ubacuje, trajanje

u broju okvira (za tokove čestica ili ponavljano ubacivanje nekog tijela), ukupni broj čestica, pozicija, oblik, veličina, tip materijala te modifikatori mase i Laméovih parametara. Podržani su oblici kugle i kvadra. Slika 6.2 prikazuje primjer jednog takvog skupa čestica.

The screenshot shows a software interface titled 'Simulation Entries' with a counter '2' in the top right. A dropdown menu is set to 'water-box'. Below it, various parameters are listed with input fields:

Entry Name	water-box		
Start Frame	0		
Duration	1		
Frame Pause Between	0		
Total Particle Count	2000		
Material Type	Water		
Shape	Box		
Position	X 10	Y 10	Z 10
Initial Velocity	X 0	Y -15	Z 0
Radius	3		
X Size	10		
Y Size	5		
Z Size	10		
Lambda Modifier	1		
Mu Modifier	1		
Mass Modifier	0.1		

Slika 6.2: Parametri jednog skupa čestica

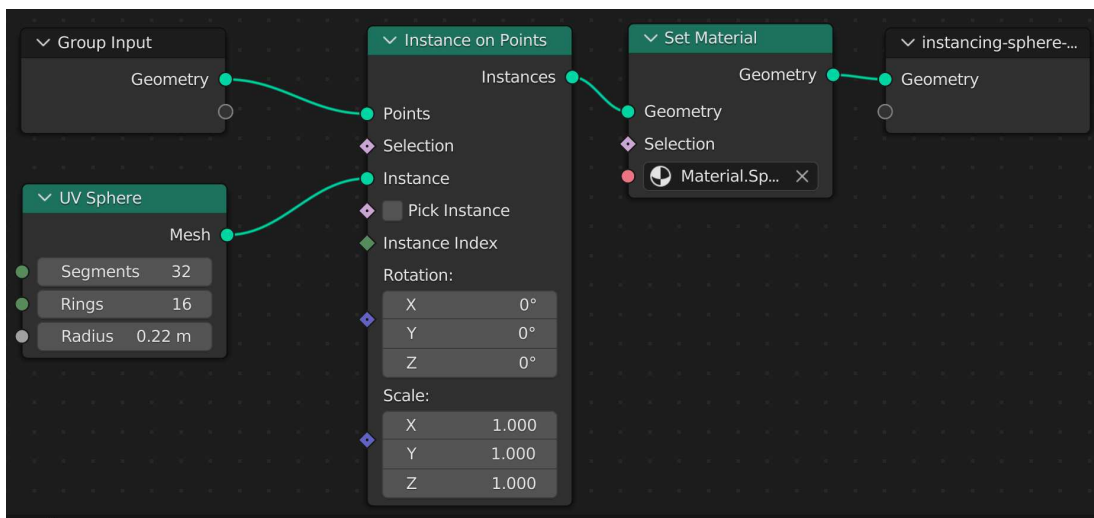
Za dobivanje početne konfiguracije čestica unutar zadanog oblika koristi se uzorkovanje Poissonovog diska, koje daje nasumično ali ravnomjerno raspoređene čestice. U dvodimenzionalnom slučaju korištena je skripta (Schlomoff (2014)). Skripta je lako proširiva na tri dimenzije. Glavna razlika je korištenje dva kuta za pretraživanje prostora.

Na kraju izračuna novog okvira, čestice iz simulatora se demultipleksiraju po materijali ili početnom skupu kojem pripadaju i njihove pozicije se spremaju u Wavefront (.OBJ) formatu na zadanoj putanji kao niz trodimenzionalnih vrhova. Uz to se spremaju i podaci o tome koji kruti objekti su u sceni, kolika je veličina rešetke i informacije za svaki okvir - trenutni broj čestica, trajanje izračuna i kinetička energija čestica. Datoteke sa spremljenim pozicijama čestica moguće je koristiti u bilo kojem drugom programu.

6.3. Vizualizacija rezultata

Unutar Unityja implementirano je rudimentarno iscrtavanje koje koristi Phongov model osvjetljenja. Korišten je Unity API za instanciranje geometrije (Unity Technologies (2022)). Primjer iz službene dokumentacije moguće je koristiti uz minimalne izmjene. Dodane su mogućnosti iscrtavanja više materijala različitih veličina čestica.

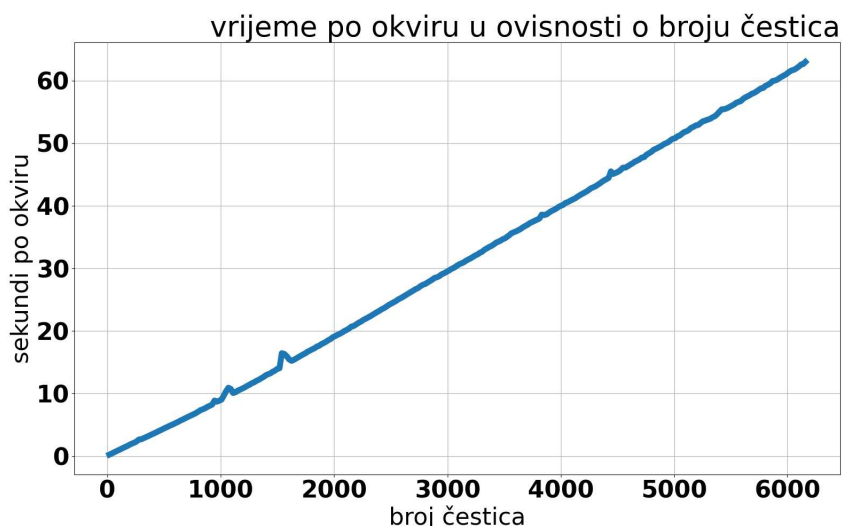
Za iscrtavanje tehnikom praćenja zrake korišten je alat Blender inačice 3.1. Rad s nizovima datoteka koje predstavljaju jedan objekt kroz više okvira olakšava dodatak Stop motion OBJ (Jensen (2021)). Nakon učitavanja niza datoteka kroz dodatak, niz datoteka se tretira kao svaki drugi model. Rezultati u ovom radu ostvareni su instanciranjem geometrije pomoću geometrijskih čvorova (slika 6.3). Na stvorenu geometriju primjenjuje se osnovni *Principled BSDF* materijal uz različitu boju, ovisno o tipu simuliranog materijala kojeg predstavlja.



Slika 6.3: Čvorovi za instanciranje geometrije unutar Blendera

7. Rezultati

Eksperimenti su izvedeni na stolnom računalu s AMD Ryzen 5 3400G procesorom. Ovisnost trajanja izračuna jednog okvira o broju čestica u slučaju vode u dvije dimenzije prikazana je na slici 7.1. Ciljani vremenski korak je impliciran kroz 30 okvira u sekundi i 33 iteracije algoritma po okviru, što daje 1.01×10^{-3} sekundi po iteraciji. Linearna ovisnost je očekivana i vidljivo je da 1000 čestica zahtijeva oko dodatnih 10 sekundi izračuna. Implementirana je optimizacija prijenosa između čestica i čvorova rešetke tako što se za pojedinu česticu ne prolazi kroz sve ćelije nego samo susjedne, a susjedne su one koje su unutar radijusa funkcije zaglađivanja. U tri dimenzije se radi o susjedstvu većem za red veličine u odnosu na dvije dimenzije pa je vrijeme po okviru proporcionalno veće. Međutim, zbog navedene optimizacije veličina domene nema pretjeran utjecaj na vrijeme izračuna.

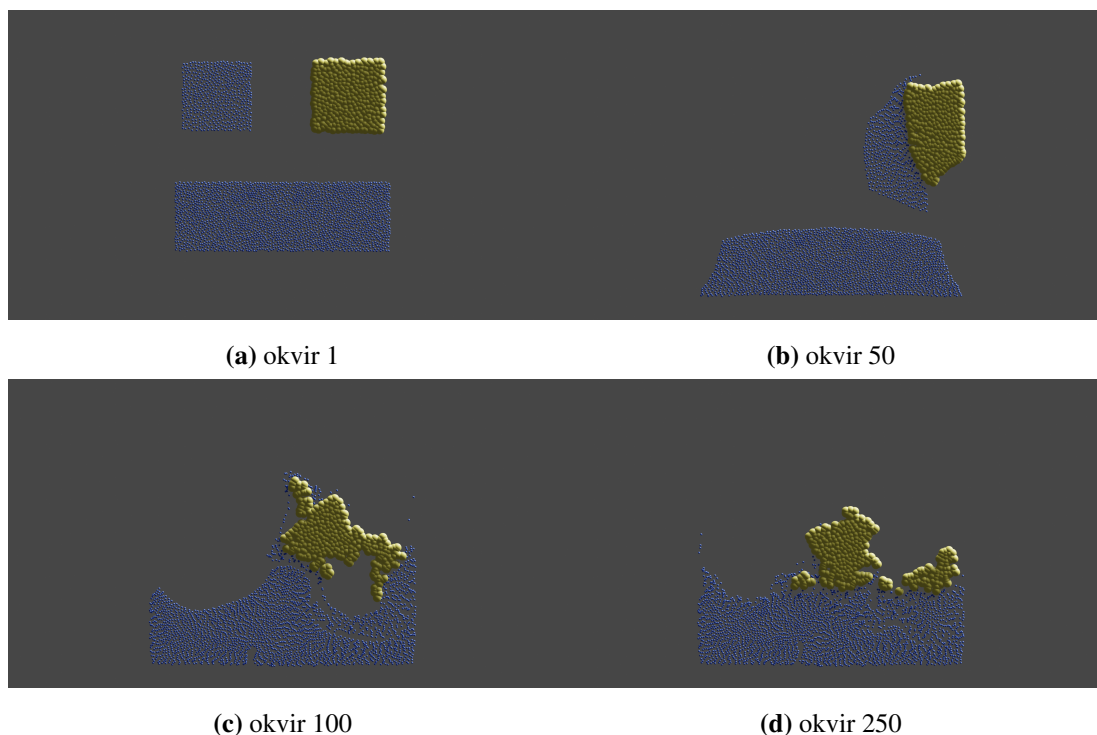


Slika 7.1: Vrijeme izračuna pojedinog okvira u ovisnosti o broju čestica vode.

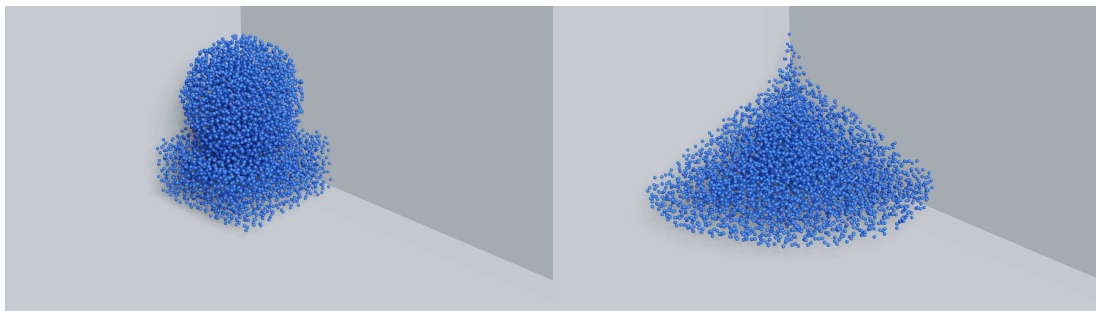
Voda je najjednostavniji materijal od implementiranih jer ne koristi posebne matrice operacije. Ostali materijali imaju zahtjevnije izračune pa je i njihovo vrijeme po okviru nešto veće. Primjerice, za scenu s 1200 čestica hiperelastičnog materijala i

isti vremenski korak, potrebno je oko 14.5 sekundi po okviru. Za scenu s 4000 čestica snijega i vremenski korak 1×10^{-4} sekundi potrebno je oko 464 sekunde po okviru. Vrijeme po okviru za pijesak varira zbog adaptivnog vremenskog koraka, koji uslijed velikih brzina može usporiti izračun jednog okvira za jedan do dva reda veličine.

U nastavku su prikazani dodatni rezultati, scena s više materijala i simulacija u tri dimenzije. Slika 7.2 prikazuje kombinaciju više materijala u jednoj sceni. Voda ima postavljenu znatno veću masu u odnosu na hiperelastični materijal zbog čega dolazi do velikog naprezanja i raspada hiperelastičnog materijala. Na slici 7.3 prikazana je voda u tri dimenzije, a na slici 7.4 hiperelastični materijal u tri dimenzije.

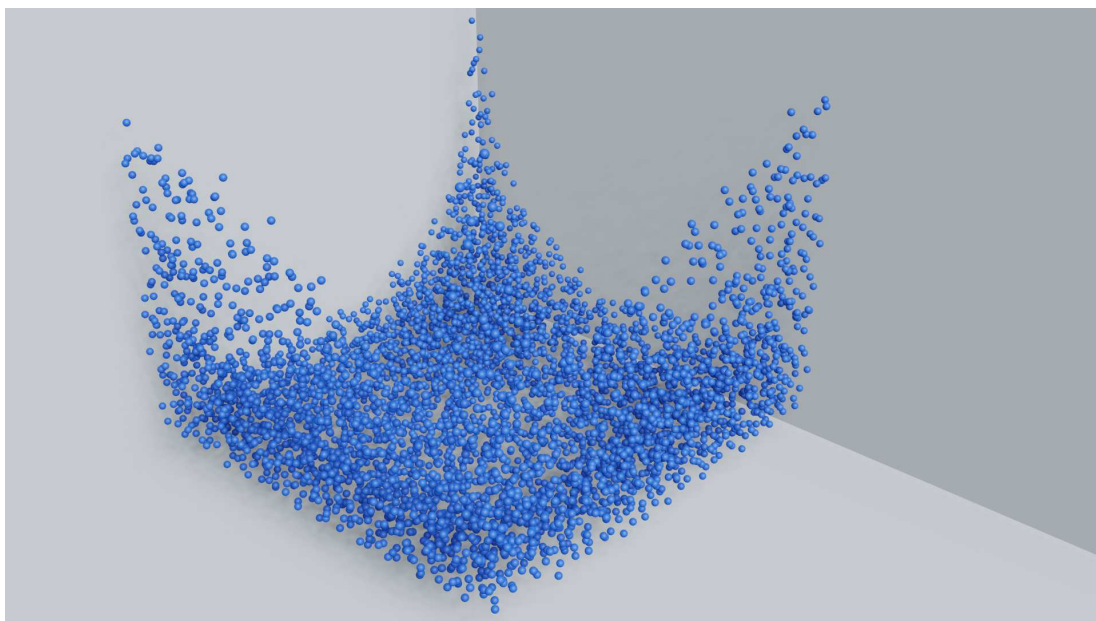


Slika 7.2: Kombinacija vode i hiperelastičnog materijala.

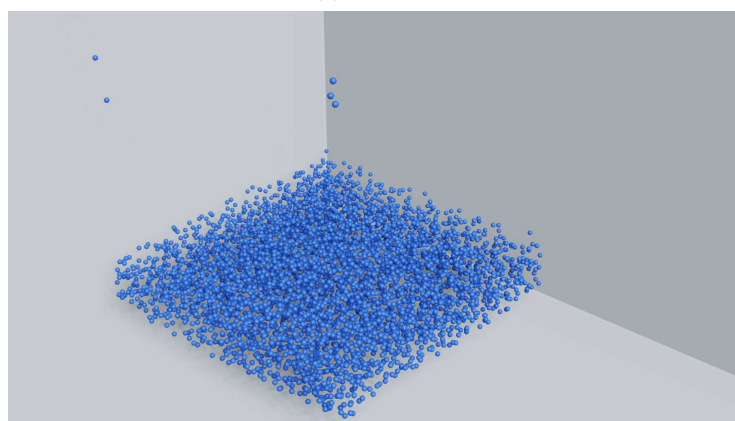


(a) okvir 1

(b) okvir 15



(c) okvir 50



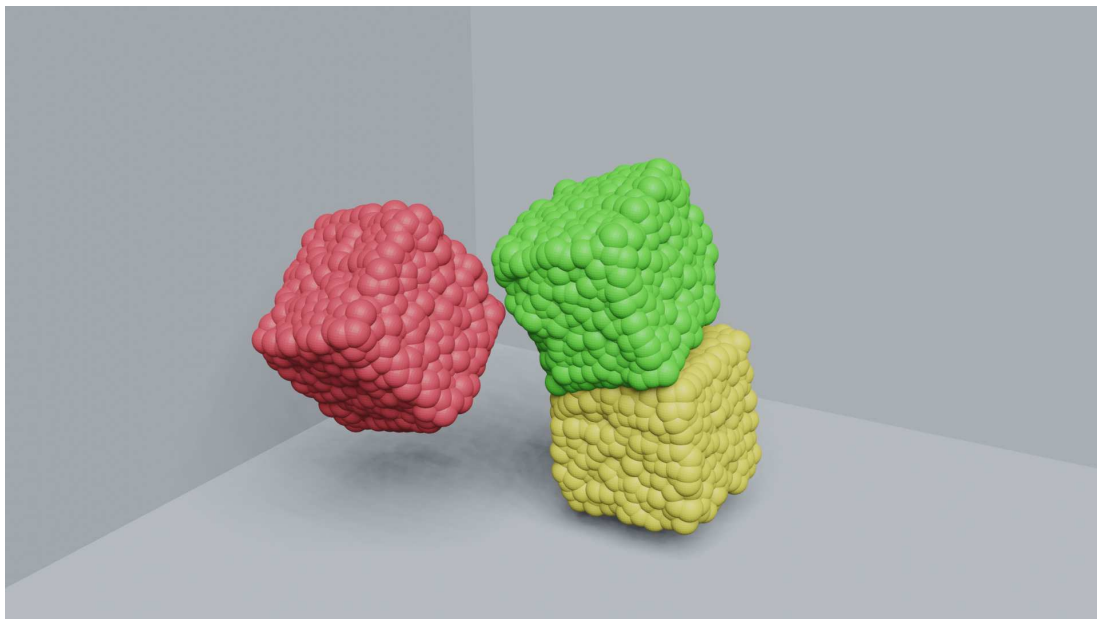
(d) okvir 300

Slika 7.3: Voda u tri dimenzije.



(a) okvir 1

(b) okvir 15



(c) okvir 50

Slika 7.4: Hiperelastični materijal u tri dimenzije.

8. Zaključak

U sklopu rada ukratko je opisana teorijska podloga metode materijalne točke. Predstavljen je algoritam metode materijalne točke za eksplicitnu integraciju. Algoritam je jednostavan za implementaciju i daje zadovoljavajuće rezultate. Idući korak bio bi razvoj višedretvenog simulatora ili korištenje GPGPU tehnika (eng. General Purpose GPU). Prije implementacije implicitnog simulatora, treba razmotriti njegove prednosti i što se njime dobiva. Iako podnose znatno manje vremenske korake, rješavanje sustava koji su rezultat implicitne integracije nosi svoju cijenu izračuna. Za početak, simulator bi valjalo prebaciti u C++. Naime, radi se o velikoj količini objekata koji su u slučaju jezika C# raspršeni u memoriji zbog čega priručna memorija nije efikasno iskorištena. C++ pruža izravno korištenje stoga i mogućnost kontinuiranih nizova objekata na hrpi.

Model vode najjednostavniji je od definiranih materijala, ali uspijeva obuhvatiti karakteristično ponašanje vode. Model hiperelastičnog materijala zasniva se na složenijim matričnim operacijama i singularnom rastavu, ali daje odlične rezultate. Hiperelastični materijal pokazuje ponašanje sustava čestica povezanih oprugama, ali i mogućnost složenijeg fenomena trganja materijala. Naime, radi se o generalizaciji opruge na više dimenzija. Modeli hiperelastičnosti služe kao podloga za druge materijale i složenije fenomene. Jedan od tih materijala je i snijeg, koji uvodi pojam plastičnosti i promjenu tvrdoće usred kompresije i rastezanja. Model pijeska najsloženiji je među definiranim, zbog operacije projekcije na Drucker-Prager stožac. Osjetljivost operacija u modelu pijeska zahtijeva malene vremenske korake koji se postižu tehnikom adaptivnog vremenskog koraka.

Interakcija s drugim objektima u sceni olakšana je korištenjem funkcija udaljenosti s predznakom. Ravnina definirana preko takve funkcije može se koristiti za granice domene. Za implementaciju neograničene domene, odnosno skalabilne rešetke, potrebne su strukture podataka za rad s prostorno rijetkim podacima, na primjer (Academy Software Foundation (2022)). Unutar tri dimenzije do izražaja dolazi prokletstvo dimenzionalnosti i potreba za značajno većim brojem čestica kako bi se popunio prostor. Dvosmjerna sprega između krutih objekata i simuliranih materijala može se imple-

mentirati proširivanjem definiranih koncepata i korištenjem tradicionalnih tehnika za sudare između krutih objekata. Ključan je zakon očuvanja količine gibanja.

Vizualni rezultati dobiveni korištenjem Blender zadovoljavajući su i iscrtavanje pojedinog okvira traje svega par sekundi za nekoliko tisuća čestica, što je znatno manje od vremena potrebnog za simulaciju. Idući korak bio bi iscrtavanje implicitno definiranih površina, a ne pojedinih čestica. Također, korisno bi bilo ubacivanje oblaka točaka, dobivenih od nekog složenog trodimenzionalnog objekta, koji se mogu generirati u nekom vanjskom alatu.

LITERATURA

Academy Software Foundation. OpenVDB. GitHub repository, <https://github.com/AcademySoftwareFoundation/openvdb>, 2022. Posljednje pristupano: 2022-06-26.

Robert Bridson i Matthias Müller-Fischer. Fluid simulation: Siggraph 2007 course notes. U *ACM SIGGRAPH 2007 Courses*, SIGGRAPH '07, stranica 1–81, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781450318235. doi: 10.1145/1281500.1281681. URL <https://doi.org/10.1145/1281500.1281681>.

R. A. Gingold i J. J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. 181:375–389, Studeni 1977. doi: 10.1093/mnras/181.3.375.

Justin Jensen. Stop motion OBJ. GitHub repository, <https://github.com/neverhood311/Stop-motion-OBJ>, 2021. Posljednje pristupano: 2022-06-20.

C. Jiang, C. Schroeder, J. Teran, A. Stomakhin, i A. Selle. The material point method for simulating continuum materials. U *ACM SIGGRAPH 2016 Courses*, SIGGRAPH '16, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342896. doi: 10.1145/2897826.2927348. URL <https://doi.org/10.1145/2897826.2927348>.

C. Jiang, C. Schroeder, i J. Teran. An angular momentum conserving affine-particle-in-cell method. *Journal of Computational Physics*, 338:137–164, lipanj 2017. doi: 10.1016/j.jcp.2017.02.050. URL <https://doi.org/10.1016%2Fj.jcp.2017.02.050>.

G. Klár, T. Gast, A. Pradhana, C. Fu, C. Schroeder, C. Jiang, i J. Teran. Drucker-prager elastoplasticity for sand animation. *ACM Trans. Graph.*, 35(4), srpanj 2016.

ISSN 0730-0301. doi: 10.1145/2897824.2925906. URL <https://doi.org/10.1145/2897824.2925906>.

G. R. Liu i M. B. Liu. *Smoothed Particle Hydrodynamics: A Meshfree Particle Method*. World Scientific, 2003. ISBN 9789812564405. URL https://books.google.hr/books?id=_cwFMmEQvZQC.

M. Liu i G. R. Liu. Smoothed particle hydrodynamics (sph): an overview and recent developments. *Archives of Computational Methods in Engineering*, 17:25–76, 03 2010. doi: 10.1007/s11831-010-9040-7. URL <https://doi.org/10.1007/s11831-010-9040-7>.

Math.NET Project. Math.NET Numerics. GitHub repository, <https://github.com/mathnet/mathnet-numerics>. Posljednje pristupano: 2022-06-20.

Patrick McCarthy. NuGet for Unity. GitHub repository, <https://github.com/GlitchEnzo/NuGetForUnity>, 2022. Posljednje pristupano: 2022-06-20.

A. Pradhana, T. Gast, G. Klár, C. Fu, J. Teran, C. Jiang, i K. Museth. Multi-species simulation of porous sand and water mixtures. *ACM Trans. Graph.*, 36(4), srpanj 2017. ISSN 0730-0301. doi: 10.1145/3072959.3073651. URL <https://doi.org/10.1145/3072959.3073651>.

Gregory Schlomoff. Poisson-disc sampling in Unity. Devlog, <http://gregschlom.com/devlog/2014/06/29/Poisson-disc-sampling-Unity.html>, 2014. Posljednje pristupano: 2022-06-20.

J. Stam. Stable fluids. U *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, stranice 121–128, USA, 1999. ACM Press/Addison-Wesley Publishing Co. ISBN 0201485605. doi: 10.1145/311535.311548. URL <https://doi.org/10.1145/311535.311548>.

J. Stam. Real-time fluid dynamics for games. Svibanj 2003. URL <http://graphics.cs.cmu.edu/nsp/course/15-464/Spring11/papers/StamFluidforGames.pdf>.

A. Stomakhin, C. Schroeder, L. Chai, J. Teran, i A. Selle. A material point method for snow simulation. *ACM Trans. Graph.*, 32(4), srpanj 2013. ISSN 0730-0301. doi:

10.1145/2461912.2461948. URL <https://doi.org/10.1145/2461912.2461948>.

Unity Technologies. Unity Scripting Reference - Graphics.DrawMeshInstancedIndirect. Unity Documentation, <https://docs.unity3d.com/ScriptReference/Graphics.DrawMeshInstancedIndirect.html>, 2022. Posljednje pristupano: 2022-06-20.

Fizikalno temeljena simulacija materijala metodom materijalne točke

Sažetak

Cilj ovog rada je izgradnja razumijevanja metode materijalne točke. Na početku je predstavljena osnovna teorijska podloga iza metode materijalne točke. Dana je formulacija algoritma za simulaciju koraka metode materijalne točke koji koristi eksplisitu integraciju. Objašnjena je i jednostavna tehnika adaptivnog vremenskog koraka. Definirano je više modela materijala - voda, hiperelastični materijal, snijeg i pijesak. Objašnjen je jednostavan način za definiranje i interakciju s krutim objektima u sceni, korištenjem funkcija udaljenosti s predznakom. Pokazani su različiti uvjeti na granici. Izneseni su pseudokodovi potrebni za implementaciju opisanih koncepata. Izrađen je simulacijski model unutar programskog alata Unity. Vizualni rezultati dobiveni su korištenjem tehnike praćenja zrake nad rezultatima simulacije unutar alata Blender.

Ključne riječi: metoda materijalne točke, eksplicitno rješavanje, simuliranje fluida i materijala, računalna animacija, funkcije udaljenosti s predznakom, deformabilni i kruti objekti, voda, hiperelastični materijal, snijeg, pijesak

Physically Based Simulation of Materials Using the Material Point Method

Abstract

The goal of this paper is building the understanding of the material point method. The basic theoretical background required for understanding the material point method is presented. Formulation of an algorithm for explicit time integration is given. A simple technique to implement an adaptive time step is presented. Various material models are defined - water, hyperelastic material, snow and sand material models. A simple way to define and interact with rigid objects is explained, based on signed distance functions. Different border conditions are explained. Pseudocode required for the implementation of discussed topics is given. A simulation model is made using the Unity engine. Visual results are made by raytracing the simulation results using Blender.

Keywords: material point method, explicit solver, fluid and material simulation, computer animation, signed distance functions, deformable and solid objects, water, hyperelastic material, snow, sand