

Simulacija tečnosti u dve dimenzije

Daniel Silađi
Gimnazija "Jovan Jovanović Zmaj"
Novi Sad

14. oktobar 2012

Apstrakt

U ovom radu je opisan fizički i informatički aspekt simulacije fluida (tečnosti) pomoću metode *Smoothed Particle Hydrodynamics* (SPH). Iako se opisan algoritam odnosi na dve dimenzije, on se lako može generalizovati na tri (pa čak i proizvoljan broj) dimenzija. Dobijeni program simulira međusobnu interakciju do oko 10000 čestica, kao i njihove sudare sa statičnim zidovima pri brzini od 60 koraka u sekundi (*60fps*) na Intelovom četvorojezgarnom Core i7 procesoru.

1 Uvod

Simulacije fluida se javljaju na raznim mestima, i to, sa jedne strane u vidu vremenske prognoze, aerodinamički oblikovanih automobila, aviona i raketa, a sa druge strane kao specijalni efekti u filmovima i kompjuterskim igrima.

Prve korake u ovoj oblasti su napravili Claude-Louis Navier i George Gabriel Stokes 1822, postavivši Navier-Stoksove jednačine. One čine osnovu mnogih modela atmosfere, okeana, vodovoda, krvotoka, a koriste se i u ispitivanju aerodinamičnosti aviona i automobila. Ipak, ove jednačine imaju jedan veliki nedostatak: ne zna se da li imaju rešenje za proizvoljno stanje fluida u 3 dimenzije. Štaviše, to pitanje predstavlja jedan od sedam milenijumskih problema Clayovog instituta za matematiku. Zbog toga se i danas radi na pronalaženju što bržih (za izračunavanje), ili što preciznijih aproksimacija ovih jednačina, koje nam garantuju da će rešenje postojati. Prve

kompjuterske simulacije su bile delo stručnjaka iz NASA-e, ustanove koja je u tom trenutku jedina imala kompjutere dovoljno jake da simuliraju vazduh u realnom vremenu, pa makar to bilo u dve dimenzije. Bitan pomak u rešavanju Navier-Stokesovih jednačina u 3D načinjen je u [3], oslanjajući se na [5].

Osnovu ovog rada čini metod Smoothed Particle Hydrodynamics (SPH), otkriven 1977, nezavisno u [4] i [6]. U oba rada SPH je iskorišćen za modeliranje zvezda, i zbog toga nije pravljen da radi u realnom vremenu. Tek u [7] je dat pojednostavljeni algoritam, pogodan za izvršavanje u realnom vremenu.

Cilj ovog rada je simuliranje tečnosti u dve dimenzije, pomoću metode Smoothed particle hydrodynamics, pri čemu je teren za simulaciju zadat od strane korisnika.

2 Teorijske osnove

2.1 Osnovni pojmovi iz dinamike fluida

U ovom delu je dat kratak pregled matematičkih i fizičkih pojmova koji se pojavljuju u Navier-Stokesovim jednačinama i jednačinama SPH.

Gustina, ρ predstavlja masu jedinične zapremine neke supstance, odnosno $\rho = \frac{m}{V}$.

Pritisak, p je skalarna veličina koja je brojno jednaka sili koja deluje normalno na jediničnu površinu, odnosno $p = \frac{F}{S}$

Viskozitet je mera unutrašnjeg trenja između slojeva fluida. Karakteriše ga koeficijent viskoziteta, η .

Površinski napon je težnja tečnosti da zauzme što manju slobodnu površinu. Karakteriše ga koeficijent površinskog napona, σ .

Gradijent neke skalarne funkcije je vektorsko polje gde vektor u svakoj tački pokazuje u smeru najvećeg porasta, i ima intenzitet jednak tom porastu. Matematički, za funkciju $f(x, y, z)$:

$$\nabla f = \frac{\partial f}{\partial x} \mathbf{i} + \frac{\partial f}{\partial y} \mathbf{j} + \frac{\partial f}{\partial z} \mathbf{k} \quad (1)$$

pri čemu se $\mathbf{i}, \mathbf{j}, \mathbf{k}$ jedinični vektori u tri dimenzije.

Divergencija vektorskog polja $\mathbf{v}(x, y, z) = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}$ je skalarna funkcija

$$\operatorname{div} \mathbf{v} = \nabla \cdot \mathbf{v} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \quad (2)$$

Laplaceov operator funkcije f je divergencija gradijenta te funkcije.

$$\nabla^2 f = \nabla \cdot \nabla f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} \quad (3)$$

Za neku tačku t , ona predstavlja meru koliko se vrednost f u tačkama na sferi sa centrom u t menja u odnosu na $f(t)$, ako poluprečnik sfere raste.

2.2 Navier-Stokesova jednačina

U opštem slučaju, Navier-Stokesova jednačina izgleda ovako:

$$\rho \left(\frac{\delta \mathbf{v}}{\delta t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \nabla \cdot \mathbf{T} + \mathbf{F} \quad (4)$$

Ona u stvari predstavlja drugi Newtonov zakon za kretanje fluida. Takođe, za potrebe ovog rada je dovoljna pojednostavljena verzija jednačine, koja se odnosi na Newtonovske fluide, nestišljivog toka. Njeno izvođenje sledi.

Krenimo od drugog Newtonovog zakona,

$$\mathbf{F} = m \mathbf{a} \quad (5)$$

Zamenimo ubrzanje sa materijalnim izvodom brzine po vremenu ($\mathbf{a} = \frac{D\mathbf{v}}{Dt}$), a masu sa gustinom (posmatramo jediničnu zapreminu):

$$\mathbf{F} = \rho \frac{D\mathbf{v}}{Dt} = \rho \left(\frac{\delta \mathbf{v}}{\delta t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) \quad (6)$$

Sa druge strane, sile koje deluju na fluid možemo da podelimo na unutrašnje (viskozitet, površinski napon,...) i spoljašnje (gravitacija,...). Za početak, neka je gravitaciona sila $\rho \mathbf{g}$ (opet, ovo nije sila u strogom smislu te reči, već sila po jedinici zapremine)

$$\rho \mathbf{g} + \mathbf{F}_{\text{unutrašnje}} = \rho \frac{D\mathbf{v}}{Dt} = \rho \left(\frac{\delta \mathbf{v}}{\delta t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) \quad (7)$$

Što se tiče unutrašnjih sila, radi pojednostavljivanja jednačina (pa samim tim i njihovog rešavanja), u daljem tekstu će se koristiti dve pretpostavke:

1. Fluid je Newtonovski
2. Fluid ima nestišljiv tok

Činjenica da je fluid Newtonovski nam govori da je viskoznost konstantna, odnosno ne zavisi od tangencijalnog napona u fluidu, a iz definicije nestišljivog toka znamo da je divergencija polja brzina jednaka nuli ($\nabla \cdot \mathbf{v} = 0$). Zato, unutrašnje sile možemo podeliti na one izazvane razlikom u pritiscima (normalni napon), i na viskozne sile izazvane razlikom u brzinama (tangencijalni napon)[1]. U ovom slučaju, sile izazvane razlikom pritisaka možemo modelirati negativnim gradijentom pritiska ($-\nabla p$), a viskozne sile sa $\eta \nabla \cdot \nabla \mathbf{v} = \eta \nabla^2 \mathbf{v}$, i tako dobijamo konačnu Navier-Stokesovu jednačinu za Newtonovske fluide nestišljivog toka:

$$\underbrace{\rho}_{\text{gustina}} \left(\underbrace{\frac{\delta \mathbf{v}}{\delta t}}_{\text{ubrzanje delića fluida}} + \underbrace{\mathbf{v} \cdot \nabla \mathbf{v}}_{\text{konvektivno ubrzanje}} \right) = \underbrace{-\nabla p}_{\text{gradijent pritiska}} + \underbrace{\eta \nabla^2 \mathbf{v}}_{\text{viskozitet}} + \underbrace{\mathbf{F}}_{\text{spoljšnje sile}} \quad (8)$$

3 Smoothed particle hydrodynamics

Osnovu SPH-a čini konačan broj čestica jednakih masa, od kojih svaka predstavlja jedan deo (V_i) zapremine datog fluida. Glavna prepreka u svim Lagrangeovskim algoritmima za simuliranje fluida leži u činjenici da je za fizički potpuno verno simulaciju potrebno simulirati praktično neograničen broj čestica. SPH taj problem prevazilazi tako što vrednost neke fizičke veličine A u nekoj tački \mathbf{r} interpolira iz diskretnog skupa tačaka (na pozicijama \mathbf{r}_i) za koje smo već izračunali vrednost A_i .

$$A_{\text{interpolirano}}(\mathbf{r}) = \sum_i A_i V_i W(\mathbf{r} - \mathbf{r}_i, h) = \sum_i A_i \frac{m_i}{\rho_i} W(\mathbf{r} - \mathbf{r}_i, h) \quad (9)$$

Jasno je da svaka čestica učestvuje u $A_{\text{interpolirano}}$ srazmerno svojoj zapremini $V_i = \frac{m_i}{\rho_i}$ i vrednosti funkcije W na udaljenosti $\mathbf{r} - \mathbf{r}_i$ od date tačke \mathbf{r} (h je konstanta o kojoj će uskoro biti reči).

Funkcija $W(\mathbf{r}, h)$ je tzv. kernel za poravnavanje (smoothing kernel) sa h kao radijusom poravnavanja i služi da "rasporedi" uticaj A_i u okolini (odnosno na udaljenosti $|\mathbf{r} - \mathbf{r}_i|$) čestice u tački \mathbf{r}_i . Značaj parametra h je samo u

tome što funkcija W za vrednosti $|\mathbf{r} - \mathbf{r}_i|$ veće od h ima vrednost 0, određujući preciznost simulacije tako što na česticu u \mathbf{r} deluju samo čestice koje su joj dovoljno blizu, tj. imaju dovoljno velik uticaj na nju. Takođe, radi efikasnosti i jednostavnosti simulacije, kernel se obično bira da bude simetričan ($W(\mathbf{r}, h) = W(-\mathbf{r}, h)$) i normalizovan ($\int W(\mathbf{r}) d\mathbf{r} = 1$).

Na primer, gustina neke čestice se pomoću gorenavedene formule može izraziti na sledeći način:

$$\rho_j = \sum_i m_i \frac{\rho_i}{\rho_i} W(\mathbf{r}_j - \mathbf{r}_i, h) = \sum_i m_i W(\mathbf{r}_j - \mathbf{r}_i, h) \quad (10)$$

Da bismo mogli da izrazimo ostale članove Naver Stokesove jednačine, potrebni su nam oblici izraza za SPH interpolaciju za ∇A i $\nabla^2 A$. Na sreću, oni se ne razlikuju mnogo od SPH izraza za interpolaciju same veličine A :

$$\nabla A_{\text{interpolirano}}(\mathbf{r}) = \sum_i A_i \frac{m_i}{\rho_i} \nabla W(\mathbf{r} - \mathbf{r}_i, h) \quad (11)$$

i analogno za $\nabla^2 A$:

$$\nabla^2 A_{\text{interpolirano}}(\mathbf{r}) = \sum_i A_i \frac{m_i}{\rho_i} \nabla^2 W(\mathbf{r} - \mathbf{r}_i, h) \quad (12)$$

Sada imamo sve potrebne "alate" da izvedemo SPH jednačine za ostale članove Navier Stokesove jednačine.

3.1 Pritisak

Koristeći SPH interpolaciju, izraz za "sile pritiska" izgleda ovako:

$$\mathbf{F}_{\text{pritisak}_i} = -\nabla p(\mathbf{r}_i) = -\sum_j p_j \frac{m_j}{\rho_j} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (13)$$

Nažalost, ovakav izraz za silu pritiska nije simetričan, tj. sila kojom čestica i deluje na česticu j je različita od sile kojom čestica j deluje na česticu i . Zato, u [2] je predložena jednostavna metoda kako da se prethodna jednačina simetrizuje: gustina ρ_j se zameni sa aritmetičkom sredinom gustina ρ_i i ρ_j :

$$\mathbf{F}_{\text{pritisak}_i} = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (14)$$

Jedino ostaje kako izračunati pritiske kod čestica i i j , a odgovor na njega je dat u [7], inspirisan osnovnom jednačinom gasnog stanja za idealni gas pri konstantnoj temperaturi:

$$p = k(\rho - \rho_0) \quad (15)$$

ρ_0 je konstanta nazvana "gustina ostatka" (rest density), i neće uticati na rezultujuće sile (koje se zasnivaju na razlici pritisaka), ali će zato (pozitivno) uticati na numeričku stabilnost simulacije [7].

3.2 Viskozne sile

Analogno izrazu za sile pritiska, dobijamo:

$$\mathbf{F}_{\text{viskozitet}_i} = \eta \nabla^2 \mathbf{v}(\mathbf{r}_i) = \eta \sum_j \mathbf{v}_j \frac{m_j}{\rho_j} \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (16)$$

Ni ova jednačina ne daje simetrične sile između čestica i i j , pa je balansiramo zamenjivanjem \mathbf{v}_j sa $\mathbf{v}_j - \mathbf{v}_i$, odnosno $\mathbf{v}_i - \mathbf{v}_j$:

$$\mathbf{F}_{\text{viskozitet}_i} = \eta \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h) \quad (17)$$

Ova smena je sa fizičke tačke gledišta sasvim opravdana, jer viskozne sile zavise od relativnih brzina slojeva tečnosti, a ne njihove apsolutne vrednosti.

3.3 Površinski napon

Radi vernije simulacije tečnosti, potrebno je uvesti i silu površinskog napon, koja je odgovorna za razne pojave uključujući "barice" koje nastaju pri prosipanju tečnosti na neku podlogu, kapilarne efekte, ... Prirodno, da bismo izračunali površinski napon, potrebno je prvo odrediti koje čestice su najbliže površini. To se postiže metodom "polja boja" (colour field), koja u tačkama gde se nalaze čestice ima vrednost 1, a u ostalim 0. Naravno, ovakvo polje je nepogodno za rad, pa se zato koristi njegova poravnata varijanta, slična ostalim SPH formulama:

$$c_S(\mathbf{r}) = \sum_i 1 \cdot \frac{m_i}{\rho_i} W(\mathbf{r} - \mathbf{r}_i, h) \quad (18)$$

Sada se vidi da će ovakvo polje imati velik gradijent blizu površine (mnogo prelaza iz 1 u 0), i mali u unutrašnjosti tečnosti (uglavnom vrednosti blizu

1). Pošto u površinskom naponu učestvuje i zakrivljenost slobodne površine tečnosti, potrebno je izraziti i taj ugao zakrivljenosti:

$$\kappa = \frac{-\nabla^2 c_S}{|\nabla c_S|} \quad (19)$$

Kombinujući ove jednačine, dobijamo:

$$\mathbf{F}_{\text{pov. napon}} = \sigma \kappa \nabla c_S = -\sigma \nabla c_S \frac{\nabla^2 c_S}{|\nabla c_S|} \quad (20)$$

3.4 Dodatne napomene

Kao što je već nagovešteno na početku dela 2.2, pošto se u SPH posmatraju nestišljive čestice, možemo zanemariti advekciju, odnosno $\nabla \cdot \mathbf{v} = 0$, Navier Stokesova jednačina za ovakve čestice je:

$$\rho \frac{\partial \mathbf{v}}{\partial t} = -\nabla p + \eta \nabla^2 \mathbf{v} + \mathbf{F} \quad (21)$$

4 Interakcija sa čvrstim telima

Interakcija sa čvrstim telima (u daljem tekstu: zidovima) je rešena na dva načina: SPH i sudar čvrstih tela.

4.1 SPH način

Iako ova metoda nije opisana nigde u literaturi, do nje se prirodno dolazi primenom SPH logike: zidovi se posmatraju kao da su i sami napravljeni od statičnih čestica velike gustine. Naravno, ovakav način predstavljanja nije zgodan ni za ljudsku, ni za kompjutersku upotrebu. Zato, ljudi (korisnici) unose linije kao parove tačaka, a program implicitno pravi nove čestice u blizini podnožja normale iz čestice na zid. Zatim se te čestice zida normalno koriste u SPH jednačinama za računanje gustina, i sila koje deluju na čestice tečnosti. Prirodno, nisu nam bitne sile koje deluju na čestice zida, jer je pretpostavka da je zid statičan.

4.2 Sudar čvrstih tela

Sa druge strane, ako i posle primene prethodnog algoritma postoji opasnost da čestica prođe kroz zid, ovaj primitivni algoritam će je odbiti kao gumenu lopticu, reflektovanjem vektora njene brzine u odnosu na zid. Ovo je poslednji deo svakog koraka simulacije, i izvršava se sve dok tokom simuliranog vremena (dt) čestica ima u blizini zid kroz koji bi prošla, a da nije prethodno bila odbijena od njega.

4.3 Pseudokod

```
1  while SIMULACIJA-RADI
2      for  $i = 1$  to BROJ-ČESTICA
3          for  $j = 1$  to BROJ-ČESTICA
4              // Prema jednačini (10)
5              izračunati gustine koje  $\check{c}estica[i]$  i  $\check{c}estica[j]$  dodaju jedna drugoj
6          for  $j = 1$  to BROJ-ZIDOVA
7              // Prema 4.1
8              izračunati dodatnu gustinu za česticu koju  $zid[j]$  dodaje  $\check{c}estici[i]$ 
9      for  $i = 1$  to BROJ-ČESTICA
10         for  $j = 1$  to BROJ-ČESTICA
11             // Prema jednačinama (14), (17), (18), (11), (12)
12             izračunati dodatne sile-pritiska, viskozne-sile,
13             gradijent-polja-boja, Laplaceov-operator-polja-boja
14             koje  $\check{c}estica[i]$  i  $\check{c}estica[j]$  dodaju jedna drugoj
15         for  $j = 1$  to BROJ-ZIDOVA
16             // Prema 4.1
17             izracunati dodatne sile pritiska i viskoziteta
18             koje  $zid[j]$  dodaje  $\check{c}estici[i]$ 
19     for  $i = 1$  to BROJ-ČESTICA
20         // Prema (20)
21         izračunati silu-površinskog-napona za česticu  $i$ 
22         // Prema (5)
23         izračunati ukupnu-silu, ubrzanje, novu-brzinu i pomeraj
24         // Prema 4.2
25         odbiti  $\check{c}esticu[i]$  od prepreka
26     Is crtati čestice na ekran
```

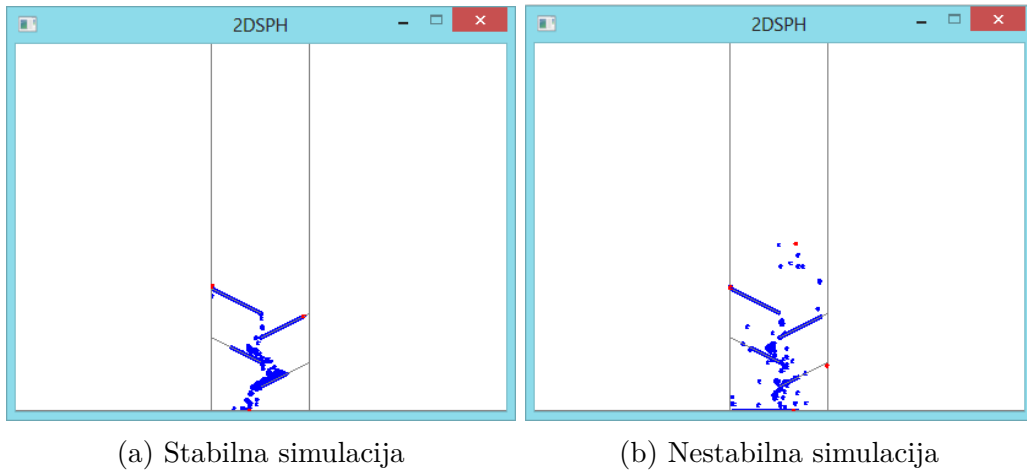

4.4 Tehički detalji

Naivna implementacija goreopisanog algoritma je naravno daleko od optimalne. Zato je korišćeno nekoliko "trikova" za optimizaciju. Jasno je da u navedenom algoritmu najviše vremena odlazi na operacije koje se odnose na svaki par čestica ($O(n^2)$ vremena). Primećeno je da od mogućih n^2 parova čestica veliki broj njih uopšte ne utiče jedna na drugu, jer su one na dovoljno velikoj udaljenosti da bi vrednost funkcije za poravnavanje bila 0. Da bi se uštedelo na vremenu, čestice su podeljene u ćelije, te se onda samo proverava interakcija čestica u istim i susednim ćelijama. Potom, kada se na početku jednog koraka odrede čestice koje će međusobno interagovati se sačuvaju u jednu listu koja se onda koristi do kraja tog koraka.

Program je pisan u programskom jeziku C++. Za grafiku je korišćena biblioteka SDL, koja se oslanja na OpenGL grafički sistem. Teren za simulaciju, kao i ostali parametri simulaciju su sačuvani u JSON formatu, za čije je učitavanje korišćena biblioteka Boost.

5 Rezultati

Napravljeni program simulira do oko 10000 čestica proizvoljne težnosti (ili gasa), zadate parametrima. Simulacija radi na 60fps, na Intelovom četvorojezgarnom Core i7 procesoru. Čestice vizuelno zadovoljavajuće reaguju međusobno, kao i sa zidovima, sve dok simulacija ne postane nestabilna. Tada čestice počinju da se kreću haotično, i zakon održanja energije prividno prestaje da važi.



Slika 1: Slike programa u radu

Uzrok nestabilnosti još nije određen, ali verovatno leži u lošem izboru parametara simulacije. Eksperimenti pokazuju da je za to najzaslužniji parametar "gasna konstanta", iz jednačine (15). Ukoliko je on previše mali, čestice se uopšte neće odbijati, a ukoliko je prevelik, simulacija postaje nestabilna. U ovom konkretnom slučaju, vrednost $k = 250$ se pokazala skoro optimalnom.

6 Zaključak

Iako simulacija daje vizualno prihvatljivi izlaz, zbog nepredvidivosti trenutka gubitka stabilnosti, još uvek nije pogodna za neku praktičnu upotrebu.

Dalji pravac razvoja bi verovatno išao u smeru povećanja stabilnosti simulacije, kao i implementacija nekog naprednijeg algoritma za renderovanje fluida.

Literatura

- [1] S. Auer. Realtime particle-based fluid simulation. Master's thesis, Technische Universität München, Fakultät für Informatik, 2009.
- [2] M. Desbrun and M. paule Gascuel. Smoothed particles: A new paradigm for animating highly deformable bodies. In *In Computer Animation and Simulation '96 (Proceedings of EG Workshop on Animation and Simulation)*, pages 61–76. Springer-Verlag, 1996.
- [3] N. Foster and D. Metaxas. Realistic animation of liquids. *Graph. Models Image Process.*, 58(5):471–483, Sept. 1996.
- [4] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics - Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389, Nov. 1977.
- [5] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8(12):2182–2189, 1965.
- [6] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, 82:1013–1024, Dec. 1977.
- [7] M. Müller, D. Charypar, and M. Gross. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '03, pages 154–159, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.