

Simulacija tečnosti u dve dimenzije

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 10 000 čestica, kao i njihove sudare sa statičnim zidovima pri brzini od 60 koraka u sekundi (60 fps) na Intelovom četvorojezgarnom Core i7 procesoru.

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 igrama.

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 dimen-

zije. Bitan pomak u rešavanju Navier-Stokesovih jednačina u 3D načinili su u Foster i Metaxas (1996), oslanjajući se na rad Harlow i Welch (1965).

Osnovu ovog rada čini metod Smoothed Particle Hydrodynamics (SPH), kojeg su nezavisno otkrili Gingold i Monaghan i Lucy (Gingold i Monaghan 1977; Lucy 1977). U oba rada SPH je iskorišćen za modeliranje zvezda, i zbog toga nije pravljen da radi u realnom vremenu. Tek su Müller i saradnici (2003) dali 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.

Teorijske osnove

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 $\rho = m/V$.

Pritisak, p , je skalarna veličina koja je brojno jednaka sili koja deluje normalno na jediničnu površinu, odnosno $p = F/S$.

Viskoznost je mera unutrašnjeg trenja između slojeva fluida. Karakteriše ga koeficijent viskoznosti, 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 $f(x, y, z)$:

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

Daniel Siladi (1995), Novi Sad, Fruškogorska 15, učenik 2. razreda Gimnazije „Jovan Jovanović Zmaj” u Novom Sadu

pri čemu su \mathbf{i} , \mathbf{j} i \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:

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

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.

Navier-Stokesova jednačina

U opštem slučaju, Navier-Stokesova jednačina izgleda ovako:

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

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 (tako dobijamo konkretan izraz za tenzor tangencijalnog napona, \mathbf{T}). Njeno izvođenje sledi.

Krenimo od drugog Newtonovog zakona:

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

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

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

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 (opet, ovo nije sila u strogom smislu te reči, već sila po jedinici zapremine):

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

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) (Auer 2009). U ovom slučaju, sile izazvane razlikom pritisaka možemo modelirati negativnim gradijentom pritiska ($-\nabla p$), a viskozne sile sa $\eta \cdot \nabla \cdot \nabla \mathbf{v} = \eta \cdot \nabla^2 \mathbf{v}$, i tako dobijamo konačnu Navier-Stokesovu jednačinu za Newtonovske fluide nestišljivog toka:

$$\rho \cdot \left(\frac{\delta \mathbf{v}}{\delta t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \eta \nabla^2 \mathbf{v} + \mathbf{F} \quad (8)$$

gde izraz u zagradi predstavlja ukupno ubrzanje, pri čemu je $\frac{\delta \mathbf{v}}{\delta t}$ ubrzanje delića fluida, a $\mathbf{v} \cdot \nabla \mathbf{v}$ konvektivno ubrzanje; ∇p je gradijent pritiska, $\eta \nabla^2 \mathbf{v}$ viskoznost a sa \mathbf{F} su označene spoljašnje sile.

Smoothed Particle Hydrodynamics (SPH)

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 veru 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 .

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

Jasno je da svaka čestica učestvuje u $A_{\text{interpolirano}}$ srazmerno svojoj zapremini $V_i = m_i/\rho_i$ 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 $\mathbf{d}_i = |\mathbf{r} - \mathbf{r}_i|$) čestice u tački \mathbf{r}_i . Značaj

parametra h je samo u tome što funkcija W za vrednosti d_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 je 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 Navier-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 :

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

i analogno za $\nabla^2 A$:

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

Sada imamo sve potrebne „alate“ da izvedemo SPH jednačine za ostale članove Navier-Stokesove jednačine.

Pritisak

Koristeći SPH interpolaciju, izraz za „sile pritiska“ \mathbf{F}_{p_i} izgleda ovako:

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

Na žalost, 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 je predložena jednostavna metoda (Desbrun i Gascuel 1996) kako da se prethodna jednačina simetrizuje: gustina ρ_j se zameni sa aritmetičkom sredinom gustina ρ_i i ρ_j :

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

Ostaje pitanje kako izračunati pritiske kod čestica i i j , a odgovor na njega, inspirisan osnovnom jednačinom gasnog stanja za idealni gas pri konstantnoj temperaturi, nalazi se u radu Müller *et al.* (2003):

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

ρ_0 je konstanta nazvana „gustina ostatka“ (rest density), i neće uticati na rezultujuće sile (koje se zasnivaju na razlici pritiska), ali će zato (pozitivno) uticati na numeričku stabilnost simulacije (Müller *et al.* 2003).

Viskozne sile

Analogno izrazu za sile pritiska, za viskozne sile \mathbf{F}_{v_i} dobijamo:

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

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 $\mathbf{v}_i - \mathbf{v}_j$:

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

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

Površinski napon

Radi vernije simulacije tečnosti, potrebno je uvesti i silu površinskog napona, koja je odgovorna za razne pojave uključujući „barice“ koje nastaju pri prosipanju tečnosti na neku podlogu, kapilarne efekte, itd. 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:

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

Dodatne napomene

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

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

Interakcija sa čvrstim telima

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

I1: 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.

I2: 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.

Tehnički 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 sa-

Pseudokod

```
while simulacija-radi
  for i = 1 to broj-čestica
    for j = 1 to broj-čestica
      // Prema jednačini (10) izračunati gustine koje čestica[i] i čestica[j] dodaju jedna drugoj
    for j = 1 to broj-zidova
      // Prema I1 izračunati dodatnu gustinu za česticu koju zid[j] dodaje čestici[i]
  for i = 1 to broj-čestica
    for j = 1 to broj-čestica
      // Prema jednačinama (14), (17), (18), (11), (12) izračunati dodatne sile-pritiska, viskozne-sile,
      // gradijent-polja-boja, Laplaceov-operator-polja-boja koje čestica[i] i čestica[j] dodaju jedna drugoj
    for j = 1 to broj-zidova
      // Prema I1 izračunati dodatne sile pritiska i viskoziteta koje zid[j] dodaje čestici[i]
  for i = 1 to broj-čestica
    // Prema (20) izračunati silu-površinskog-napona za česticu[i]
    // Prema (5) izračunati ukupnu-silu, ubrzanje, novu-brzinu i pomeraj
    // Prema I2 odbiti česticu[i] od prepreka
  Is crtati čestice na ekran
```

čuvaju 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 za simulaciju, su sačuvani u JSON formatu, za čije je učitavanje korišćena biblioteka Boost.

Rezultati

Napravljeni program simulira do oko 10 hiljada čestica proizvoljne tečnosti (ili gasa), zadate parametrima. Simulacija radi na 60 fps na Intelovom četvorjezrgarnom 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.

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.

Zaključak

Iako simulacija daje vizuelno 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

Auer S. 2009. Realtime particle-based fluid simulation. Master thesis, Technische Universität München, Fakultät für Informatik, Boltzmannstr. 3, D-85748 Garching, Nemačka.

Desbrun M., Gascuel M. P. 1996. Smoothed particles: A new paradigm for animating highly deformable bodies. U *Computer Animation and Simulation '96* (Proceedings of EG Workshop on Animation and Simulation). New York: Springer, str. 61–76.

Foster N., Metaxas D. 1996. Realistic animation of liquids. *Graph. Models Image Process.*, **58** (5): 471.

Gingold R. A., Monaghan J. J. 1977. Smoothed particle hydrodynamics – Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, **181**: 375.

Harlow F. H., Welch J. E. 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, **8** (12): 2182.

Lucy L. B. 1977. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, **82**: 1013.

Müller M., Charypar D., Gross M. 2003. Particle-based fluid simulation for interactive applications. U *Proceedings of the 2003 ACM SIGGRAPH / Eurographics symposium on Computer animation, SCA '03*, Aire-la-Ville: Eurographics Association, str. 154-159.

Danijel Siladi

Simulation of Liquids in Two Dimensions

This paper describes the physical and algorithmical aspect of fluid (liquid) simulation, using the method of Smoothed Particle Hydrodynamics (SPH). Although the algorithm applies to two dimensions, it can easily be generalized to an arbitrary number of dimensions. The created program successfully simulates 10 000 particles, and their collision with static walls at 60 frames per second, using an Intel quad-core i7 processor. 