

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1675

**ANIMACIJA MEĐUDJELOVANJA KRUTIH
TIJELA I FLUIDA**

Milan Vukušić

Zagreb, rujan 2007.

Sadržaj:

1.	Uvod	1
2.	Pozadina dinamike fluida.....	2
3.	Pozadina računalne dinamike fluida	4
3.1.	MAC mreža	4
3.2.	Konačne razlike	6
4.	MAC metoda.....	9
4.1.	Odabir Δt	9
4.2.	Pomicanje čestica	10
4.3.	Identifikacija ćelija sa i bez fluida	11
4.4.	Uvjeti kontinuiteta i pomicanja granica	11
4.5.	Rješavanje Navier – Stokes-ovih jednadžbi	16
4.6.	Projekcija tlaka.....	17
5.	Rješavanje sustava linearnih jednadžbi.....	18
5.1.	Postavljanje matrica.....	18
5.2.	Granični uvjeti	21
5.2.1.	Neumann-ov granični uvjet	21
5.2.2.	Dirichlet-ov granični uvjet	23
5.3.	Raspršena struktura matrice	24
5.4.	Metoda konjugiranog gradijenta.....	27
6.	Kruti fluidi.....	29
6.1.	Domene Krutih fluida	29
6.2.	Ograničenja dinamike krutih tijela	30

6.3.	Osnovne jednađbe krutih fluida	31
6.4.	Implementacija Krutih fluida	32
6.4.1.	Rješavanje Navier-Stokes-ovih jednađbi: $u^n \rightarrow u^*$	33
6.4.2.	Izrađunavanje sila krutih tijela: $u^* \rightarrow \hat{u}$	33
6.4.3.	Ograničavanje brzina na brzine krutih tijela: $\hat{u} \rightarrow u^{n+1}$	35
6.5.	Pomicanje rađunske domene	37
7.	Programska podrška	40
8.	Rezultati	43
9.	Zaključak	47
10.	Popis slika	48
11.	Literatura	50
12.	Sažetak	51
13.	Abstract	52

1. Uvod

Jednadžbe gibanja koje upravljaju kretanjem fluida zovu se *Navier-Stokes-ove* jednadžbe. U računalnoj animaciji postoji bogata povijest rješavanja tih jednadžbi a ovo poglavlje će pokušati ostvariti tradicionalne ciljeve snabdijevanja čitatelja sa svim potrebnim informacijama da napravi vlastiti simulator fluida. Prilikom pisanja simulatora fluida programer mora donijeti mnoge odluke, te odluke tiču se jednostavnosti objašnjenja, primjene i popularnosti u računalnoj literaturi. Ovaj rad opisuje jedinstvenu primjenu simulatora fluida, posebno MAC metodu sa projekcijama tlaka uz primjenu nestlačivosti, kao i metodu za simuliranje međudjelovanja između krutih tijela i fluida nazvanu Kruti fluidi [1]. Seminar počinje opisima jednadžbi koje upravljaju kretanjem fluida, a zatim opisuje jedan način da se te jednadžbe odjele tako da ih se može rješavati na zasebnoj računalnoj domeni. Slijedeći korak je metoda Krutih fluida koja je dobila svoj naziv iz načina na koji su predstavljena kruta tijela. Naime kruta tijela su predstavljena Navier-Stokes-ovim jednadžbama uz odgovarajuća ograničenja kako bi dobiveni rezultati odgovarali dinamici krutih tijela.

2. Pozadina dinamike fluida

U slijedećoj raspravi, vektor \mathbf{u} predstavlja brzinu fluida. Tlak (skalar) bit će predstavljen sa p , a gustoća fluida sa ρ (gustoća je također skalar, ali za njezinu vrijednost većinom uzimamo 1, iz razloga što je gustoća vode veoma blizu 1). Skalar ν predstavlja kinematički viskozitet fluida.

Jednadžbe gibanja viskoznog nestlačivog fluida su Navier-Stokes-ove jednadžbe.

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

$$\mathbf{u}_t = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \nu \nabla^2 \mathbf{u} - \frac{1}{\rho} \nabla p + \mathbf{f} \quad (2)$$

Jednadžba (1) nam govori da vektor brzine ima divergenciju 0. To u pojednostavljenom značenju glasi; količina fluida koja uđe u neko određeno infinitezimalno područje jednaka je količini fluida koja iz tog područja izađe. To je očuvanje mase za nestlačive fluide. U stvarnosti niti jedan fluid nije nestlačiv, ali nestlačivost se pretpostavlja za fluide koji se gibaju malim brzinama, zato jer je stlačivost zanemariva i takva pretpostavka omogućava vrlo prikladno rješenje za tlak (opisano nešto dalje). Eksplozije, let mlaznih zrakoplova na nadzvučnim brzinama i slične situacije proračunavaju se pomoću drugih formula.

Jednadžba (2) opisuje očuvanje momenta i sastoji se od nekoliko dijelova. Čitajući s lijeva na desno jednadžba objašnjava da je trenutna promjena brzine fluida na zadanoj poziciji jednaka sumi četiriju uvjeta; advekciji fluida, difuziji, promijeni tlaka i vanjskim silama. Vektor \mathbf{u}_t je derivacija brzine fluida po vremenu. Za parcijalnu derivaciju koristit će se primjerene oznake (i.e. $\mathbf{u}_t = \frac{\partial \mathbf{u}}{\partial t}$). Izraz za advekciju $(\mathbf{u} \cdot \nabla)\mathbf{u}$ koristi se za određivanje smjera u kojem okolni fluid potiskuje infinitezimalno područje unutar tog fluida. Rijeka brzog toka je sredina u kojoj prevladava advekcija, kada se u takvu rijeku ulije mala količina vode, ta ista

bit će brzo odnesena riječnim tokom. Izraz za difuziju $\nabla \cdot (\nu \nabla \mathbf{u})$, opisuje koliko brzo se poništi razlika u brzinama unutar fluida koji okružuje zadanu točku. Vrijednost ν je kinematički viskozitet fluida, i povišanjem te vrijednosti, smanjuje se vrijeme koje je potrebno da se poništi razlika u brzinama. Pri konstantnom viskozitetu izlučuje se ν i tako se dobiva poznatiji oblik te formule koji glasi $\nu \nabla^2 \mathbf{u}$. Treći izraz $\frac{1}{\rho} \nabla p$ je izraz za razliku u tlaku i opisuje kako se mali dijelovi fluida potiskuju iz područja višeg tlaka u područja nižeg tlaka. Posljednji vektor \mathbf{f} opisuje vanjske sile po jedinici mase, koje djeluju na fluid u cijelosti. U većini slučajeva \mathbf{f} je samo gravitacija, ali također može biti rotacija Zemlje, vjetar ili bilo koje drugo, od strane korisnika, određeno vektorsko polje.

Da bi netko razumio Navier-Stokes-ove jednačbe, mora također razumjeti i matematičke znakove korištene unutar njih.

Diferencijalni vektor ∇ (što se izgovara "del"), koristi se sa skalarom p da bi se dobio vektor za razliku tlakova:

$$\nabla p = (p_x, p_y, p_z) \quad (3)$$

Divergencija $\nabla \cdot$ je diferencijalna operacija na vektoru, npr. $\mathbf{u} = (u, v, w)$, čiji produkt je skalar:

$$\nabla \cdot \mathbf{u} = u_x + v_y + w_z \quad (4)$$

Laplace-ov operator, ∇^2 , je skalarni diferencijalni operator i koristi se kao dio jednačbe konstantnog viskoziteta. Laplace-ov operator na skalar u je:

$$\nabla \cdot (\nabla u) = u_{xx} + u_{yy} + u_{zz} \quad (5)$$

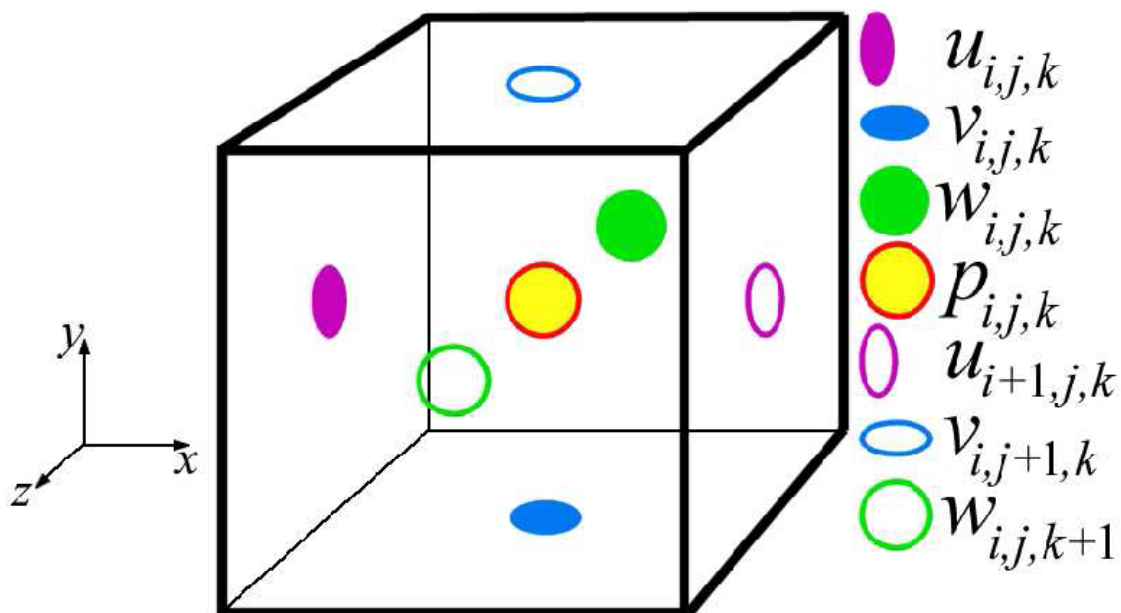
3. Pozadina računalne dinamike fluida

Računalo, očito ne može riješiti Navier-Stokes-ove jednadžbe za beskonačan broj točki unutar kontinuirane domene, da bi se riješile jednadžbe (1) i (2), treba se odabrati diskretna domena. Ovo poglavlje opisuje diskretnu domenu za računanje, znanu kao MAC mreža i numeričku metodu poznatu kao konačne razlike (*finite differences*), da bi se Navier-Stokes-ove jednadžbe rastavile na djeliće veličine jednog bita koje računalo može riješiti. Prvo se opisuje MAC mreža. MAC mreža dobila je naziv prema Marker-And-Cell metodi simulacije fluida sa slobodnim površinama koja je prvi put opisana 1965 od strane Harlow-a i Welch-a [2]. MAC pristup je dobro opisan u nekoliko drugih radova, ali u nadi da ovaj rad bude zaseban i da se popune neke praznine oko 3D dijelova prijašnjih opisa, jedna metoda bit će opisana ovdje. Prvo ćemo definirati MAC mrežu i pozadinu konačnih razlika potrebnu da se shvati Marker-And-Cell postupak.

3.1. MAC mreža

Ćelije u prostoru simulacije su jednako velike kocke ili "*voxels*", sa stranicama dužine Δx . Dva tipa varijabli mogu biti pohranjeni u ćeliji; skalarna vrijednost ili vektor. Sve skalarne vrijednosti smještaju se u centar ćelija koji se ponekad naziva i čvorište ćelije. Sve vektorske vrijednosti smještaju se u razmaknutom rasporedu, prema kojem se x komponenta smješta na lijevu stranicu kocke, y komponenta na donju stranicu, a z komponenta na stražnju stranicu kocke. Te vrijednosti se ponekad nazivaju čvorišta stranica. Simulator fluida opisan u ovom tekstu sprema dvije vrijednosti u svaku ćeliju: tlak i brzinu. Skalarna vrijednost tlaka p sprema se u čvorište ćelije, a vektor brzine

$\mathbf{u} = (u, v, w)$ rastavlja se na komponente i svaka od njih se smješta u odgovarajuće čvorište stranice. Jedna ćelija na lokaciji i, j, k je prikazana na slici 1. Trebalo bi spomenuti da mnogi drugi izvori koji opisuju MAC metodu koriste polovinsko notiranje (*half index notation*). Ako netko poželi prevesti notiranje korišteno u ovom radu na polovinsko notiranje, tada samo treba oduzeti $\frac{1}{2}$ na odgovarajućem čvorištu stranice (npr. x komponenta brzine $u_{i,j,k}$ u našoj notaciji je $u_{i-1/2,j,k}$ u polovinskoj notaciji).



Slika 1: Ćelija u MAC mreži.

Često su potrebne vrijednosti za varijable koje se ne nalaze na čvorištima ćelija ili stranica, da bi se dobile te vrijednosti koristi se trilinearna interpolacija. Čvorište ćelije koje se nalazi na lokaciji i, j, k u MAC mreži, u prostoru simulacije nalazi se na lokaciji $(i\Delta x, j\Delta x, k\Delta x)$. Indeksiranje je bitno kada se uzimaju u obzir interpolirane vrijednosti sa lokacija MAC mreže. Vrijednost varijable α na interpoliranoj lokaciji (x, y, z) , notira se $\alpha(x, y, z)$. Jedan specifičan primjer toga je vektor brzine u čvorištu ćelije,

$$\mathbf{u}(i\Delta x, j\Delta x, k\Delta x) = \left(\frac{u_{i,j,k} + u_{i+1,j,k}}{2}, \frac{v_{i,j,k} + v_{i,j+1,k}}{2}, \frac{w_{i,j,k} + w_{i,j,k+1}}{2} \right), \quad (6)$$

gdje je svaka komponenta vektora spremljena na istoj lokaciji. To se razlikuje od brzine spremljene na lokaciji i, j, k u MAC mreži koja glasi

$$\mathbf{u}_{i,j,k} = (u_{i,j,k}, v_{i,j,k}, w_{i,j,k}), \quad (7)$$

i čije su komponente spremljene na tri različite lokacije $(i\Delta x - \Delta x/2, j\Delta x, k\Delta x)$, $(i\Delta x, j\Delta x - \Delta x/2, k\Delta x)$ i $(i\Delta x, j\Delta x, k\Delta x - \Delta x/2)$.

Ta računaska mreža ima dimenzije $I \times J \times K$ i indeksirana je sa $i \in [0, I-1]$, $j \in [0, J-1]$, $k \in [0, K-1]$. Kao što je već spomenuto, ćelije na vanjskom rubu domene su granične ćelije, tako da fluid može postojati samo u ćelijama gdje je $1 \leq i \leq I-2$, $1 \leq j \leq J-2$, i $1 \leq k \leq K-2$. Veličina simulacijskog prostora se određuje širinom u metrima, tada je $\Delta x = \text{širina}/I$.

3.2. Konačne razlike

Konačne razlike se koriste da bi se mogle riješiti Navier-Stokes-ove jednačbe u MAC mreži. Diskretna konačna razlika gradijenta, jednačba (3) za razliku tlaka u ćeliji i, j, k , je vektor, a svaka od komponenata se rješava na odgovarajućem čvorištu stranice.

$$(\nabla p)_{i,j,k} = \left(\frac{p_{i,j,k} - p_{i+1,j,k}}{\Delta x}, \frac{p_{i,j,k} - p_{i,j+1,k}}{\Delta x}, \frac{p_{i,j,k} - p_{i,j,k+1}}{\Delta x} \right). \quad (8)$$

Konačna razlika divergencije, jednačba (4) brzine na ćeliji i, j, k je skalar

$$(\nabla \cdot \mathbf{u})_{i,j,k} = \frac{u_{i+1,j,k} - u_{i,j,k} + v_{i,j+1,k} - v_{i,j,k} + w_{i,j,k+1} - w_{i,j,k}}{\Delta x}. \quad (9)$$

i pohranjena je u čvorištu ćelije.

Konačne razlike prikazane gore imaju točnost prvog stupnja. Konačne razlike koriste Taylor-ov red da bi se približno odredio rezultat. Točnost određivanja je uvjetovana brojem stupnjeva do kojeg se razvija Taylor-ov red. Točnost prvog stupnja znači da se koristi samo jedan stupanj. Ako se red razvije do drugog stupnja, određivanje bi imalo točnost drugog stupnja. Konačna razlika Laplace-ovog operatora ima točnost drugog stupnja. Kada se to primjeni, Laplace-ov operator glasi:

$$\nabla^2 u_{i,j,k} = \frac{u_{i-1,j,k} + u_{i,j-1,k} + u_{i,j,k-1} - 6u_{i,j,k} + u_{i+1,j,k} + u_{i,j+1,k} + u_{i,j,k+1}}{\Delta x}. \quad (10)$$

Do sada smo usmjerili raspravu samo na prostorne izračune, ali vremenski izračuni su također potrebni da bi se odredilo \mathbf{u}_t . Uzmimo u obzir jednostavnu linearnu jednadžbu.

$$\mathbf{u}_t = \lambda \mathbf{u}, \quad (11)$$

gdje je $\lambda < 0$ neka konstanta. Točno rješenje gornje jednadžbe $e^{\lambda t}$ propada. Euler-ova verzija te linearne jednadžbe glasi:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \lambda \mathbf{u}^n. \quad (12)$$

Superscript notiranje se koristi za predstavljanje vremena. Trenutno vrijeme se označava sa n , tako da vrijednost \mathbf{u} određenom vremenskom koraku glasi \mathbf{u}^n . Vremenski korak Δt predstavlja razmak između koraka n i koraka $n + 1$. Ove jednadžbe koriste se za eksplicitne vremenske izračune, što govori da je desna strana jednadžbe (12) već otprije poznata. Takve jednadžbe je lako riješiti, ali one iziskuju korištenje kriterija za stabilnost koji ograničava veličinu Δt .

Stabilnost je veoma važno područje numeričkih izračuna, ali kada bi se više posvetili toj temi, skrenuli bi od cilja ovog poglavlja. Ovo poglavlje je pokušaj da čitatelji osjete što je stabilnost bez obraćanja mnogo pažnje Von Neumann-ovoj analizi o stabilnosti.

Promotrite jednadžbu (12) s novim rasporedom

$$\mathbf{u}^{n+1} = \mathbf{u}^n(1 + \Delta t\lambda) \quad (13)$$

Sada promotrite izraz nazvan pseudo-prirast $\tilde{\mathbf{G}}$, koji se koristi za analizu vremenskih izračuna:

$$\tilde{\mathbf{G}} = \frac{\mathbf{u}^{n+1}}{\mathbf{u}^n}. \quad (14)$$

Ideja je da $|\tilde{\mathbf{G}}|$ ostane malen, ali kada se primjeni na jednadžbu (13) on izluči

$$\tilde{\mathbf{G}}^{FE} = 1 + \Delta t\lambda, \quad (15)$$

tako da pseudo-prirast raste proporcionalno sa $|\lambda|$ i Δt . Prema tome, da bi se očuvala stabilnost prema Euler-ovim jednadžbama, vremenski korak mora postati manji svaki put kada λ postane veći.

$$\Delta t \approx \frac{1}{\lambda}. \quad (16)$$

Jedan način da se izbjegne ovaj problem sa stabilnošću je da se koriste implicitni oblici poput obrnutih Euler-ovih jednadžbi za vremenske izračune.

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \lambda \mathbf{u}^{n+1}. \quad (17)$$

Kada iz jednadžbe (17) izlučimo pseudo-prirast dobijemo:

$$\tilde{\mathbf{G}}^{BE} = \frac{1}{1 - \Delta t\lambda}, \quad (18)$$

i bez obzira na povećanje $|\lambda|\Delta t$, $|\tilde{\mathbf{G}}|$ ostaje malen.

4. MAC metoda

Marker-And-Cell metoda ima dvije glavne komponente: ćelije, opisane u poglavlju 3.1. u kojima se prati brzina i tlak fluida, i veliku zbirku marker-čestica u tekućini koji označavaju koje ćelije su napunjene fluidom i prenose brzinu na prazne ćelije. Jedan vremenski pomak u simulacijama fluida se računa u nekoliko koraka, što ćemo sada spomenuti a kasnije će biti opisano s više detalja:

1. Izabire se primjerena veličina koraka Δt
2. Čestice se pomiču prema trenutnim vektorima brzine \mathbf{u}
3. Svaka ćelija se označava i puni sa fluidom, ili ostaje prazna s obzirom na to dali određena ćelija sadrži čestice, i određuju se brzine za ćelije koje su sada pune fluida, a bile su prazne.
4. Određuju se granični uvjeti za sve čvrste predmete i stranice ćelija sa čije jedne strane je praznina a sa druge se nalazi fluid.
5. Vrijednosti brzina za ćelije koje sadrže fluid se ažuriraju prema jednadžbama (1) i (2).

Gore navedeni koraci se ponavljaju za svaki vremenski pomak u simulaciji.

4.1. Odabir Δt

Prvi korak je pronaći primjereni Δt . Informacije unutar MAC mreže ne mogu putovati brže od toga. Da bi rješenje bilo dosljedno sa realnim rješenjem, mora se uvesti ograničenje poznato kao Courant-Friedrichs-Lewy (CFL) uvjet. CFL

uvjet ističe da vremenski korak mora biti dovoljno malen da informacije ne bi putovale preko nekoliko ćelija istovremeno. CFL uvjet glasi

$$\Delta t < \frac{\Delta x}{\max(|u|, |v|, |w|)}, \quad (19)$$

i to je prvo ograničenje koje stavljamo na Δt . Drugo ograničenje uvodi se zbog jednadžbe za konstantan viskozitet $\nu \nabla^2 \mathbf{u}$. To ograničenje za stabilnost u 3D glasi

$$\Delta t < \frac{\Delta x^2}{6\nu}, \quad (20)$$

U ovoj jednadžbi ν ima istu ulogu kao i λ u jednadžbi (16).

4.2. Pomicanje čestica

Nakon odabira vremenskog koraka, čestice se pomiču pomoću Runge-Kutta tehnike drugog stupnja. Poznate kao izmijenjeni Euler ili Midpoint metoda. Čestica kreće iz lokacije $\mathbf{x} = (x, y, z)$ i pomiče se u dva koraka:

$$\begin{aligned} \mathbf{x}^{n+1/2} &= \mathbf{x}^n + \Delta t \mathbf{u}(\mathbf{x}^n) \\ \mathbf{x}^{n+1} &= \mathbf{x}^n + \frac{\Delta t}{2} [\mathbf{u}(\mathbf{x}^n) + \mathbf{u}(\mathbf{x}^{n+1/2})]. \end{aligned} \quad (21)$$

Ako se lokacija $\mathbf{x}^{n+1/2}$ nalazi unutar prazne ćelije, tada se drugi korak ne izračunava, zato jer se izvan ćelija sa fluidom ne mogu dobiti pouzdane brzine. Koristi se tehnika drugog stupnja, iz razloga što je to najviši stupanj koji se može postići kada se koristi linearna interpolacija da se dobiju vrijednosti brzina.

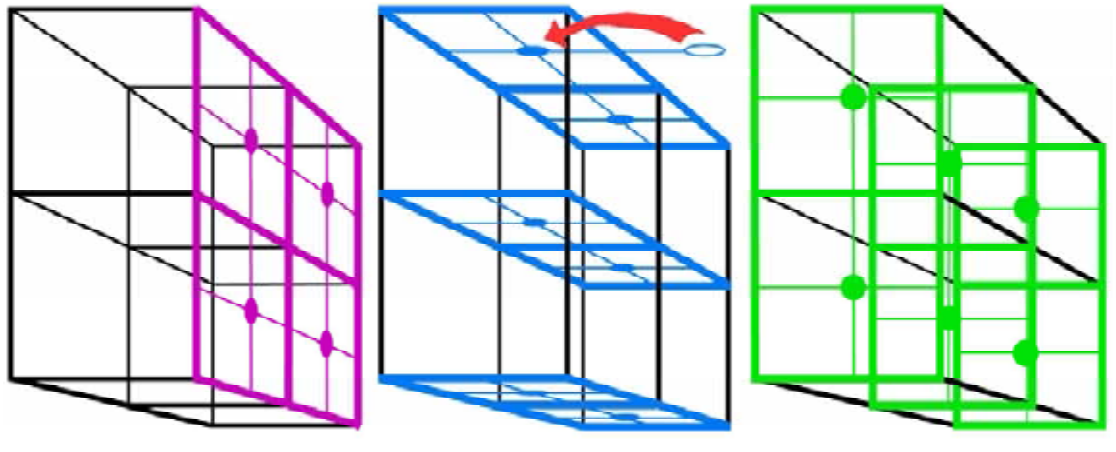
4.3. Identifikacija ćelija sa i bez fluida

Kada se ažuriraju pozicije čestica za označavanje, one se koriste za identifikaciju ćelija sa i bez fluida. Ako ćelija sa fluidom ne sadrži čestice, tada ona postaje prazna ćelija, a ako ćelija koja je bila prazna sada sadrži najmanje jednu česticu, ona postaje ćelija sa fluidom. Kada se to dogodi moraju se odrediti brzine za do tada prazne ćelije. Svaka čestica za označavanje ima pridruženu brzinu, a kada se pronađe nova ćelija sa fluidom, brzine na stranicama se dobiju tako da se sve čestice za označavanje sakupe u čvorištima stranica. Ako se u blizini čvorišta ne nalazi niti jedna čestica za označavanje, koristi se brzina prve najbliže čestice. Položaji čestica u blizini površine daju vrlo detaljan oblik slobodnim površinama, mnogo detaljniji od onog kojeg daju same ćelije. To omogućuje MAC metodi da stvara detaljne površine fluida, a u isto vrijeme koristi grubu mrežu ćelija. Činjenica da čestice daju visoku razlučivost površinama fluida je vrlo važna i biti će objašnjena u slijedećim poglavljima.

4.4. Uvjeti kontinuiteta i pomicanja granica

Stranice ćelija kod kojih se fluid nalazi i s jedne i s druge strane opisane su u poglavlju 3.3.5. Ovaj dio se bavi stranicama ćelija koje s jedne strane imaju fluid a s druge čvrste zidove ili prazan prostor.

U MAC mreži bilo koja od ćelija može predstavljati nepremostivu zapreku za fluid. Konkretno, šest strana simulacijske mreže tretira se kao čvrsti zid. Raspravljat ćemo o tome kako se odnositi prema tom sloju graničnih ćelija. Prema drugima graničnim ćelijama se odnosi gotovo isto. Promotrite četiri ćelije na lijevoj strani MAC mreže sa čije desne strane se nalaze ćelije sa fluidom. Te ćelije su prikazane na tri načina na slici 2. Pri tome imajte na umu da brzina u ima tri komponente u , v i w .



Slika 2: Čvrste granične ćelije na lijevoj strani MAC mreže.

Prikaz na lijevoj strani označava čvorišta stranica, ljubičasto, koja sadrže u komponente brzine. Poput svih takvih čvorišta koja imaju s jedne strane čvrste granice, a s druge ćelije s fluidom, te ljubičaste vrijednosti su podešene na 0, tako da fluid ne bi prolazio kroz čvrste granice.

Središnji prikaz označava čvorišta, plavo, koja sadrže v komponentu brzine, a desni prikaz označava čvorišta stranica, zeleno, koja sadrže w komponentu brzine. Ta čvorišta sadrže tangencijalnu brzinu zida. Sva ta čvorišta imaju čvrste granice sa njihove obje strane. Postoje tri načina da se podese vrijednosti za takva čvorišta. Prvi način je da se njihove vrijednosti jednostavno podese na 0. Druga dva načina su poznata kao uvjeti klizanja; promotrite crvenu strelicu koja pokazuje smjer od čvorišta (prikazanog kao plavi oval) koje je okruženo fluidom, prema čvorištu koje ima čvrste granice s obje strane. Jednostavno kopiranje vrijednosti sa jednog na drugo čvorište bi stvorilo uvjete slobodnog klizanja, u kojima bi voda poprskana po zidu nastavila dalje uz zid bez usporavanja. Kopiranje negativne vrijednosti bi stvorilo takozvane uvjete bez klizanja gdje je tangencijalna brzina fluida koji se sastane sa granicom jednaka 0. Kako se Δx približava 0, uvjet bez klizanja je točniji, ali uvjet slobodnog klizanja održava animaciju živahnom. Zadaća animatora je koji će od ta dva uvjeta odabrati za animaciju.

Do sada smo raspravljali kako se podešavaju brzine za čvorišta stranica koja imaju čvrstu granicu s jedne ili s obje strane. Kasnije ćemo pokazati kako riješiti Navier-Stokes-ovih jednadžbi za čvorišta koja s obje strane imaju fluid, takozvane fluidne stranice, ali prvo moramo opisati kako se podešavaju vrijednosti za čvorišta stranica koja s jedne strane imaju fluid, a s druge su prazne (sa te strane se nalazi zrak), poznate kao površinske stranice. Čelija sa barem jednom površinskom stranicom zvat će se površinska čelija. U daljnjoj raspravi unutarnja stranica je ili fluidna stranica ili stranica kod koje se barem s jedne strane nalazi čvrsto tijelo. Drugim riječima unutarnja stranica je stranica za koju znamo vrijednost rješavanjem graničnih uvjeta ili Navier-Stokes-ovih jednadžbi.

Kod površinskih stranica ne koristimo uvjete klizanja, jer ovdje mora biti zadovoljen uvjet da nema odstupanja brzine unutar čelija. Svaka površinska čelija može imati od jedne do šest strana izloženih zraku i svaki od tih slučajeva je obrađen da ne bi došlo do odstupanja.

Da bi se podesili granični uvjeti za površinsku stranicu u površinskoj čeliji, koristi se uvjet kontinuiteta prikazani jednadžbom (1). Tijekom slijedeće rasprave razmotrit ćemo površinsku čeliju na lokaciji i, j, k . Diskretna verzija uvjeta kontinuiteta korištenjem jednadžbe (9) glasi:

$$\nabla \cdot \mathbf{u} = 0 \rightarrow \tag{22}$$

$$u_{i+1,j,k} - u_{i,j,k} + v_{i,j+1,k} - v_{i,j,k} + w_{i,j,k+1} - w_{i,j,k} = 0,$$

koja uključuje vrijednosti brzina svih šest stranica čelije. Ako postoji točno jedna površinska stranica u čeliji, tada se njezina vrijednost određuje izravno pomoću jednadžbe (22). Npr. ako je desna strana čelije površinska stranica, tada joj se vrijednost određuje izrazom:

$$u_{i+1,j,k} = u_{i,j,k} - v_{i,j+1,k} + v_{i,j,k} - w_{i,j,k+1} + w_{i,j,k}. \tag{23}$$

Kada na ćeliji postoji više od jedne površinske stranice, tada ne postoji jedinstveno rješenje za jednadžbu kontinuiteta. U idealnom slučaju, jednadžbe za nestanak tangencijalnog naprezanja trebaju biti korištene kao dodatne jednadžbe za izračune drugih nepoznanica u sustavu, ali te jednadžbe se ne koriste u MAC metodi opisanoj ovdje. Harlow [3] je istaknuo da kod dvije dimenzije primjena slabijih uvjeta $u_{i+1,j,k} = u_{i,j,k}$ i $v_{i,j+1,k} = v_{i,j,k}$ rezultira dobivanjem zanemarivih tangencijalnih naprezanja, čak i kod vrlo viskoznih fluida. Prilikom odlučivanja načina na koji ću podesiti vrijednosti za površinske ćelije, nastojao sam se držati ovih uputa, ali pri tome paziti da je jednadžba (22) i dalje istinita.

Postoje šezdeset i tri moguće postavke za površinske ćelije. Jednadžba (23) i pet sličnih jednadžbi brine se za slučajeve kada u ćeliji postoji samo jedna površinska stranica. Ako u ćeliji postoji šest površinskih stranica, tada ništa ne radimo i na fluid utječu samo vanjske sile. Ono što slijedi je rasprava o ostalih pedeset i šest slučajeva poredanih po broju površinskih stranica u ćeliji. Između tih slučajeva postoje velike sličnosti, tako da detaljnija rasprava svakog nije potrebna.

Kada na površinskoj ćeliji postoje samo dvije površinske stranice, moramo uzeti u obzir dva slučaja, jedan je kada se sa suprotnih strana ćelije nalazi zrak, u tom slučaju ništa se ne poduzima. To može uzrokovati divergenciju, ali čini se da kapljicama i tankim slojevima fluida omogućuje prirodan slobodan pad. U drugom slučaju kopiramo brzine na površinske stranice sa stranica nasuprot njima i toj vrijednosti dodajemo pola razlike od preostale dvije stranice na svaku površinsku stranicu.

Kada su na ćeliji tri površinske stranice, također postoje dva slučaja koja moramo uzeti u obzir. U slučaju kada su površinske stranice nasuprot unutarnjim stranicama, jednostavno se vrijednosti brzina kopiraju sa unutarnjih stranica na površinske stranice. U drugom slučaju postoji jedna unutrašnja stranica koja je nasuprot površinskoj stranici, a i dvije površinske stranice jedna nasuprot drugoj. U tom slučaju brzinu za površinsku stranicu koja je nasuprot unutrašnje stranice

određujemo kao u prvom slučaju kada je postojala samo jedna stranica u dodiru sa zrakom i pri tome ignoriramo činjenicu da postoje još dvije stranice koje su u dodiru sa zrakom.

Kada ćelija ima četiri površinske stranice također postoje dva slučaja. Prvi je kada su dvije površinske stranice jedna nasuprot drugoj, ali preostale dvije nisu jedna nasuprot drugoj. U tom slučaju na površinske stranice koje nisu jedna nasuprot drugoj upisujemo suprotne vrijednosti i ponovno podijelimo razliku preostale dvije stranice i dodamo je da bi dobili ćeliju bez rasipanja brzine. U slučaju kada svaka površinska stranica ima nasuprot također površinsku stranicu, tada izračunamo rasipanje brzine u ćeliji, podijelimo na četiri dijela i ravnomjerno rasporedimo po površinskim stranicama.

Posljednjih šest slučajeva je kada na ćeliji postoji pet površinskih stranica. U takvom slučaju koristimo se metodom kao kad u ćeliji postoji samo jedna površinska ćelija, a brzina koju određujemo je za onu stranicu nasuprot unutarnje stranice. Moramo istaknuti da u ovom slučaju nema dovoljno informacija za rad, jer imamo jednu jednadžbu a pet nepoznanica. Odluka koju donosimo u ovom slučaju može prouzročiti nepravilnosti u oblicima ili neravnine na površini.

Ponekad kada se interpoliraju vrijednosti brzina blizu površine (kada je potrebna brzina čestice za označavanje), potrebne su vrijednosti izvan fluida. Jedan dobar način da se odrede te brzine je da se koristi produžena brzina da bi se odredile brzine uzduž normale na površinu. Međutim u danom trenutku nemamo podataka o normali, pa je najjednostavnije rješenje kopirati vrijednost brzine koja je najbliže čestici za označavanje.

4.5. Rješavanje Navier – Stokes-ovih jednačnji

Postoje dva važna koraka u rješavanju \mathbf{u}_t . Jedan od problema koji se javljaju prilikom rješavanja jednačnje (2) je taj da nema podataka o tome kako riješiti tlak p . Srećom jednačnja (1) daje dovoljno podataka da bi se dobio odgovarajući tlak. Kako se koristi taj tlak i kako se dobiva rješenje vidjet ćemo u poglavlju 3.4. Za sada ćemo ignorirati tlak i opisati izmijenjenu Euler-ovu metodu za dobivanje približne brzine $\tilde{\mathbf{u}}$.

Jednačnja za približnu brzinu pri konstantnom viskozitetu glasi:

$$\tilde{\mathbf{u}} = \mathbf{u} + \Delta t [-(\mathbf{u} \cdot \nabla) \mathbf{u} + \nu \nabla^2 \mathbf{u} + \mathbf{f}] \quad (24)$$

U jednačnji je izostavljena stavka o tlaku. Za metodu konačnih razlika se koriste slijedeće jednačnje i rješavaju se za svaku dimenziju posebno:

$$\begin{aligned} \tilde{u}_{i,j,k} = & u_{i,j,k} + \frac{\Delta t}{4\Delta x} [(u_{i-1,j,k} + u_{i,j,k})(u_{i-1,j,k} + u_{i,j,k}) - (u_{i,j,k} + u_{i+1,j,k})(u_{i,j,k} + u_{i+1,j,k}) \\ & + (u_{i,j-1,k} + u_{i,j,k})(v_{i-1,j,k} + v_{i,j,k}) - (u_{i,j,k} + u_{i,j+1,k})(v_{i-1,j+1,k} + v_{i,j+1,k}) \\ & + (u_{i,j,k-1} + u_{i,j,k})(w_{i-1,j,k} + w_{i,j,k}) - (u_{i,j,k} + u_{i,j,k+1})(w_{i-1,j,k+1} + w_{i,j,k+1})] \\ & + \frac{\nu \Delta t}{\Delta x^2} [u_{i-1,j,k} + u_{i,j-1,k} + u_{i,j,k-1} - 6u_{i,j,k} + u_{i+1,j,k} + u_{i,j+1,k} + u_{i,j,k+1}] + \Delta t f, \end{aligned} \quad (25)$$

$$\begin{aligned} \tilde{v}_{i,j,k} = & v_{i,j,k} + \frac{\Delta t}{4\Delta x} [(v_{i-1,j,k} + v_{i,j,k})(u_{i,j-1,k} + u_{i,j,k}) - (v_{i,j,k} + v_{i+1,j,k})(u_{i+1,j-1,k} + u_{i+1,j,k}) \\ & + (v_{i,j-1,k} + v_{i,j,k})(v_{i,j-1,k} + v_{i,j,k}) - (v_{i,j,k} + v_{i,j+1,k})(v_{i,j,k} + v_{i,j+1,k}) \\ & + (v_{i,j,k-1} + v_{i,j,k})(w_{i,j-1,k} + w_{i,j,k}) - (v_{i,j,k} + v_{i,j,k+1})(w_{i,j-1,k+1} + w_{i,j,k+1})] \\ & + \frac{\nu \Delta t}{\Delta x^2} [v_{i-1,j,k} + v_{i,j-1,k} + v_{i,j,k-1} - 6v_{i,j,k} + v_{i+1,j,k} + v_{i,j+1,k} + v_{i,j,k+1}] + \Delta t g, \end{aligned} \quad (26)$$

$$\begin{aligned} \tilde{w}_{i,j,k} = & w_{i,j,k} + \frac{\Delta t}{4\Delta x} [(w_{i-1,j,k} + w_{i,j,k})(u_{i,j,k-1} + u_{i,j,k}) - (w_{i,j,k} + w_{i+1,j,k})(u_{i+1,j,k-1} + u_{i+1,j,k}) \\ & + (w_{i,j-1,k} + w_{i,j,k})(v_{i,j,k-1} + v_{i,j,k}) - (w_{i,j,k} + w_{i,j+1,k})(v_{i,j+1,k-1} + v_{i,j+1,k}) \\ & + (w_{i,j,k-1} + w_{i,j,k})(w_{i,j,k-1} + w_{i,j,k}) - (w_{i,j,k} + w_{i,j,k+1})(w_{i,j,k} + w_{i,j,k+1})] \\ & + \frac{\nu \Delta t}{\Delta x^2} [w_{i-1,j,k} + w_{i,j-1,k} + w_{i,j,k-1} - 6w_{i,j,k} + w_{i+1,j,k} + w_{i,j+1,k} + w_{i,j,k+1}] + \Delta t h, \end{aligned} \quad (27)$$

gdje komponente trenutne brzine \mathbf{u} su (u, v, w) , komponente približne brzine $\tilde{\mathbf{u}}$ su $(\tilde{u}, \tilde{v}, \tilde{w})$, a komponente vanjskih sila \mathbf{f} su (f, g, h) . Rješavanje ovih

jednadžbi za svaku stranicu ćelije u MAC mreži sa fluidom na obje strane daje približnu brzinu $\tilde{\mathbf{u}}$. Međutim, u približnoj brzini nema stavke o tlaku, tako da jednadžba (1) nije bila razmatrana za ćelije koje se ne nalaze na površini.

4.6. Projekcija tlaka

Do sada nismo uzeli u obzir tlak zato jer kod približne brzine $\tilde{\mathbf{u}}$ postoji rasipanje (i.e., $\nabla \cdot \tilde{\mathbf{u}} \neq 0$) i slijedeći korak koji moramo poduzeti jest da koristimo rješenje za tlak koje će ukloniti rasipanja iz približne brzine i tako učvrstiti uvjet o nestlačivosti fluida. Ovaj korak je poznat kao korak projiciranja tlaka. Izraz iz jednadžbe (2) koji smo izostavili iz jednadžbe (24) je:

$$-\frac{1}{\rho} \nabla p, \quad (28)$$

i moramo ga koristiti za dobivanje konačne brzine

$$\mathbf{u}^{new} = \tilde{\mathbf{u}} - \frac{\Delta t}{\rho} \nabla p. \quad (29)$$

Također se moramo pobrinuti da konačna brzina bude za nestlačiv fluid, zato koristimo rasipanje iz jednadžbe (29) da bi dobili

$$\nabla \cdot \mathbf{u}^{new} = \nabla \cdot \tilde{\mathbf{u}} - \frac{\Delta t}{\rho} \nabla \cdot (\nabla p) = 0. \quad (30)$$

Preuređenjem jednadžbe (30) dobivamo jednadžbu

$$\Delta t \nabla^2 p = \rho \nabla \cdot \tilde{\mathbf{u}} \quad (31)$$

koja se mora riješiti po tlaku p . Kada to napravimo uvrstimo rješenja u jednadžbu (29) da bi dovršili predviđanje tlaka i tako potvrdili uvjet o nestlačivosti.

5. Rješavanje sustava linearnih jednadžbi

Jednadžba (24) može izgledati jako komplicirano kada se pretvori u oblik jednadžbi za konačnu razliku (25 – 27), ali sve sa desne strane jednadžbe je poznato i rješenje se dobiva direktno. Kod jednadžbe (31) postoji razlika. Iako je desna strana te jednadžbe poznata, lijeva sadrži Laplace-ov operator ∇^2 . Taj operator može biti predstavljen jednadžbom (10), tako da za svaku od N ćelija u MAC mreži postoji jedna nepoznanica za tlak i jedna jednadžba za tu nepoznanicu. Jednadžbe za tlak su linearne i u kombinaciji sa N nepoznanica tvore linearni sustav. Takav sustav može biti predstavljen matricom i za njegovo rješavanje koristi se linearna algebra.

5.1. Postavljanje matrica

U ovom djelu opisat ćemo kako napraviti linearni sustav unutar matrice koji će riješiti jednadžbu (31). Zapis koji će biti korišten za matricu je

$$\mathbf{Ax} = \mathbf{b}. \quad (32)$$

U linearnom sustavu sa N jednadžbi i N nepoznanica, \mathbf{A} je matrica veličine $N \times N$ gdje svaki red predstavlja jednadžbu; \mathbf{x} je stupac veličine N sa unosom za svaku nepoznanicu, a \mathbf{b} je stupac veličine N koji predstavlja konstantne poznate vrijednosti koje mogu biti preseljene na desnu stranu jednadžbe. U ovom tekstu N predstavlja količinu ćelija ispunjenih fluidom. Vrijednosti tlaka moraju biti pronađene za svaku od tih ćelija. Da bi prikazali jednadžbu (31) pomoću linearnog sustava u jednadžbi (32), neka je

$$\mathbf{A} \equiv -\Delta x^2 \nabla^2 \quad (33)$$

negativni Laplace-ov operator skaliran za negativno od širine mreže na kvadrat. Negativna vrijednost se koristi iz razloga da sustav bude pozitivno definiran, a sve množimo sa Δx^2 , da bi unosi u matrici bile cjelobrojne vrijednosti, tako da cijeli postupak bude jednostavniji. Neka je

$$\mathbf{x} \equiv p \quad (34)$$

nepoznata vrijednost tlaka koju trebamo pronaći, a neka je

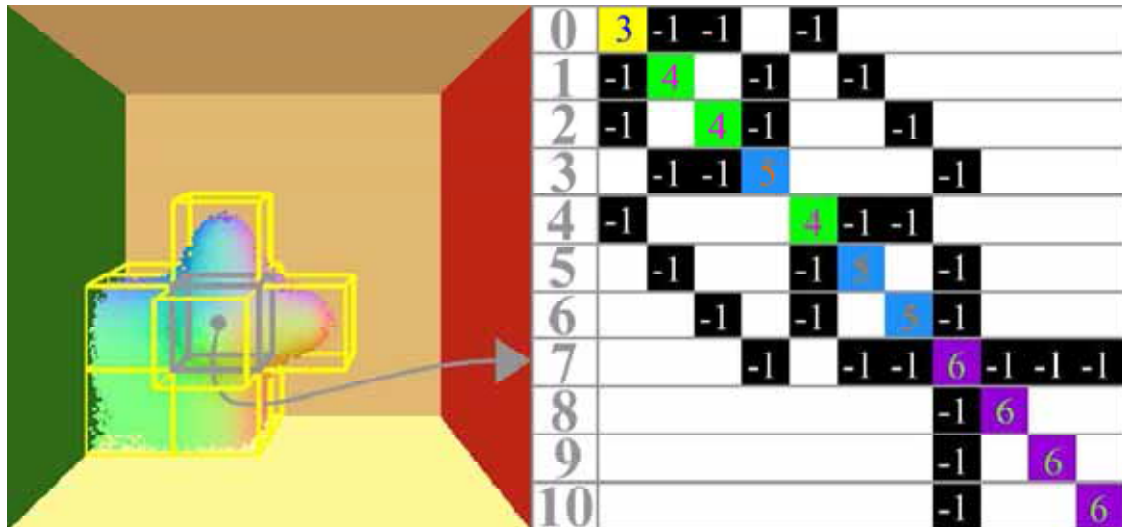
$$\mathbf{b} \equiv \frac{\Delta x^2 \rho \nabla \tilde{\mathbf{u}}}{\Delta t} \quad (35)$$

divergencija svake ćelije skalirana sa Δx^2 , gustoćom fluida i inverznom vrijednosti vremenskog koraka.

Poznate konstante, \mathbf{b} , u jednadžbi (35) se rješavaju izravno, zbog toga što su vrijednosti brzina zamijenjene sa približnim brzinama, jednadžba (24) i uvjetima kontinuiteta, jednadžba (22), na površinskim ćelijama. Sa tim na umu, rješenje desne strane jednadžbe (32) je trivijalno. Slijedi rasprava o lijevoj strani jednadžbe, koja je mnogo zanimljivija.

Oblik konačnih razlika za lijevu stranu jednadžbe za ćeliju okruženu fluidom sa svih strana je slična jednadžbi (10), ali umjesto x -komponente brzine na stranici ćelije koristi se vrijednost tlaka iz središta ćelije. Slijedeće prijelazne jednadžbe iz jednadžbi za konačne razlike u unos za redak matrice odgovaraju takvoj ćeliji.

$$\begin{aligned} -\Delta x^2 \nabla^2 p_{i,j,k} &= -\Delta x^2 \frac{p_{i-1,j,k} + p_{i,j-1,k} + p_{i,j,k-1} - 6p_{i,j,k} + p_{i+1,j,k} + p_{i,j+1,k} + p_{i,j,k+1}}{\Delta x^2} \\ &= -p_{i-1,j,k} - p_{i,j-1,k} - p_{i,j,k-1} + 6p_{i,j,k} - p_{i+1,j,k} - p_{i,j+1,k} - p_{i,j,k+1} \\ &= (-1 \cdots -1 \cdots -1 \cdots + 6 \cdots -1 \cdots -1 \cdots -1) p_{i,j,k} \\ &= (\mathbf{Ax})_{i,j,k}. \end{aligned} \quad (36)$$



Slika 4: Slika 3 sa označenom jedinom ćelijom sa fluidom koja je sa svih strana okružena ćelijama sa fluidom.

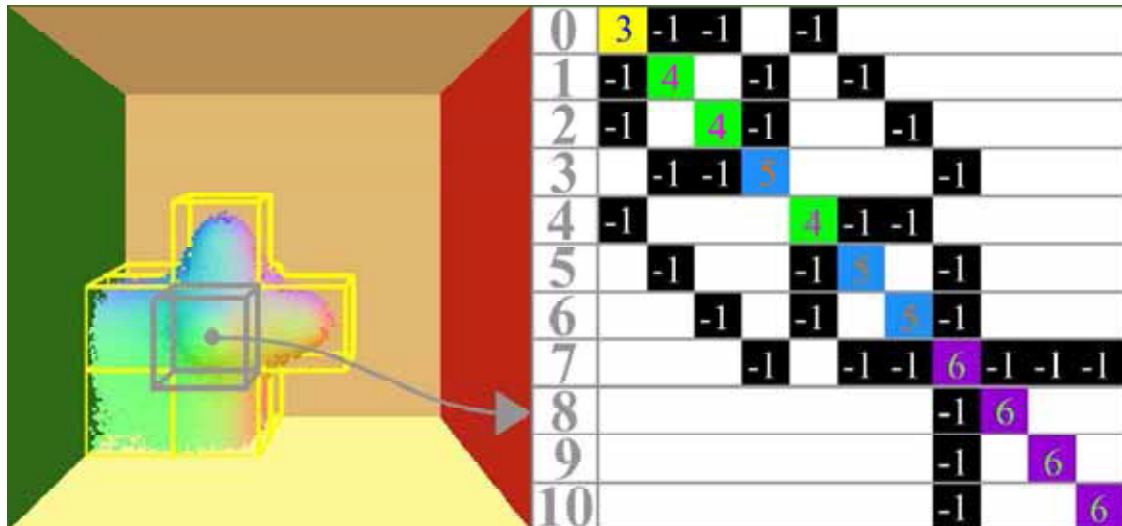
5.2. Granični uvjeti

Postoje dva tipa graničnih uvjeta o kojima ćemo raspravljati: Dirichlet-ov koji se postavlja prema poznatim vrijednostima i Neumann-ov koji se postavlja prema vrijednosti izvedenoj između granične ćelije i ćelije unutar matrice.

5.2.1. Neumann-ov granični uvjet

Za matricu projekcije tlaka A , postavit će se Neumann-ov granični uvjet vrijednosti 0 za svaku ćeliju sa čvrstim tijelom. To znači da nema promjene u tlaku između ćelija sa čvrstim tijelima i onih punjenih fluidom.

konstantu jednostavno preselimo na stranu poznatog vektora **b** i tako je opet izostavimo iz retka u matrici **A**.



Slika 6: Slika 3 sa ćelijom sa fluidom koja strši u zrak označenom sivom bojom. Ova ćelija je posebna jer je okružena sa praznim ćelijama (ispunjenih zrakom) sa svih strana osim jedne. Dirichlet-ovi granični uvjeti moraju biti postavljeni za sve okolne prazne ćelije.

5.2.2. Dirichlet-ov granični uvjet

Tlak zraka u simulaciji je 0, pa je tako prema Dirichlet-ovom graničnom uvjetu tlak za svaku ćeliju ispunjenu zrakom također 0. Pogledajmo ćeliju označenu sivom bojom na slici 6, znamo da tlak u pet ćelija koje su ispunjene zrakom i okružuju istaknutu ćeliju, mora biti postavljen na 0. Indeksi tih praznih ćelija ne unose se u matricu. Označena ćelija ima indeks 8, a ćelija iza nje ima indeks 7, tako da jednačba matrice za ćeliju glasi:

$$-p_{lijevo} - p_{ispod} - p_7 + 6p_8 - p_{ispred} - p_{iznad} - p_{desno} \quad (39)$$

Ako zamijenimo vrijednosti pet praznih ćelija u jednačbi (39) sa 0, dolazimo do jednačbe predstavljene u osmom retku matrice na slici 6. Ako je

potrebno uvrstiti neku drugu vrijednost za tlak zraka, tada se jednostavno te vrijednosti nakon uvrštavanja u jednadžbu (39) prebace na stranu poznatog vektora \mathbf{b} .

Ćelije sa fluidom u MAC mreži mogu imati bilo kakvu kombinaciju graničnih uvjeta, ali u matrici \mathbf{A} mora biti zastupljena barem jedna prazna ćelija, ili će matrica biti singularna i neće se moći jedinstveno invertirati. Drugi način da se ovo objasni je da, ako ne postoji ćelija ispunjena zrakom, tada ne možemo podijeliti jednadžbu (32) sa \mathbf{A} (*i.e.*, $\mathbf{x} = \mathbf{b}/\mathbf{A}$), da bi dobili rješenje za \mathbf{x} . U jednom primjeru to je riješeno na način da su Dirichlet-ovi granični uvjeti postavljeni za sve ćelije na vrhu simulacije, čime je postignuto da je simulacija odozgo u dodiru sa zrakom.

Nakon postavljanja \mathbf{A} pomoću opisanih graničnih uvjeta i izračunavanja vektora poznatih vrijednosti \mathbf{b} , spremni smo riješiti vektor \mathbf{x} u našem sustavu linearnih jednadžbi. To ćemo izvesti, ne pomoću dijeljenja sa \mathbf{A} , nego pomoću tehnike nazvane konjugativni gradijent (*conjugate gradient* - CG). Prije opisa CG-a, opisat ćemo raspršenu strukturu matrice \mathbf{A} .

5.3. Raspršena struktura matrice

Matrica \mathbf{A} je raspršena matrica, što znači da se sastoji od uglavnom nula. Da bi simulacija radila kako treba, ne možemo pohraniti sve te unose sa nulama. Npr. neka se simulacija sastoji od mreže veličine $40 \times 40 \times 40$ i gotovo je u cijelosti ispunjena fluidom, tako da broj ćelija s fluidom iznosi $N = 64000$. Vrijednosti sa pomičnim zarezom (floating point) u \mathbf{A} , \mathbf{x} i \mathbf{b} bit će spremljeni kao tip "double" koji je veličine 8 byte-a. Za pohranu tih vrijednosti potrebno je 0,5 MB prostora (MB=1,048,576 byte-a). Međutim imajmo na umu da je matrica veličine $N \times N$, pa ako pohranimo "double" vrijednost za svaki unos, to će zauzeti 244 GB memorije. Kao što je spomenuto prije, većina tih unosa su nule, pa ako

pronađemo način da zanemarimo te unose, matrica će stati u memoriju. Ako zanemarimo sve granične uvjete, maksimalan iznos unosa koji nisu nula je $7N$. To upućuje na stvaranje $O(N)$ sheme pomoću koje ćemo pohraniti i pristupati matrici.

Postoji mnogo tipova matrica sa raspršenom strukturom i isto tako mnogo načina za njihovu pohranu. Prilikom odabira sheme za pohranu moramo na umu imati nekoliko stvari, kao što su: operacije koje će biti izvođene s matricom (množenje matrice s matricom ili vektorom), tip matrice koja se pohranjuje (strukturirana ili ne strukturirana) i jednostavnost programiranja (korištenje manjeg prostora za pohranu može zahtijevati komplicirane kontrolne petlje).

```
typedef struct {
    byte numOff // broj nedijagonalnih elemenata
    double diag // vrijednost dijagonalnog elementa
    double offEntries[6] // vrijednosti nedijagonalnih
                        // elemenata
    int offIndices[6] // polje indeksa nedijagonalnih
                    // elemenata
} matrix-row
```

Pseudo kod 1: Matrix-row struktura podataka korištena za raspršenu matricu A.

Svaki redak unutar matrice **A** predstavlja jednu jednadžbu, pa je logično da shemu spremanja temeljimo na strukturi retka. Koristimo strukturu matrica-red prikazanu pseudo kodom 1, koja ima veličinu od 81 byte-a. Prema tome prostor za pohranu matrice iznosi $O(N)$ kada pohranimo takvu strukturu za svaku od N ćelija, a u gornjem primjeru on iznosi 4,95 MB, što je sasvim zadovoljavajuće. Sada ćemo pobliže pogledati svaki unos u matrix-row strukturi i razmotriti alternative koje se mogu iskoristiti na tim mjestima.

Unos numoff mora pohraniti broj nedijagonalnih elemenata u retku i ima veličinu od jednog byte-a, zato jer znamo da može sadržavati vrijednosti između 0 i 6. Unos diag sprema vrijednosti na dijagonali retka matrice u obliku 8 byte vrijednosti s pomičnim zarezom, a unos offEntries je polje koje sadrži nedijagonalne vrijednosti. Ako unutra retka ima manje od 6 ne dijagonalnih

uvjeta, tada se ne pristupa zadnjim unosima `offEntries`, ali oni svejedno zauzimaju memoriju. Isti je slučaj i za `offIndices` unose. Oni sadrže cjelobrojne vrijednosti koje su indeksi unutar mreže ćelija s fluidom, kojima odgovaraju nedijagonalni unosi. To je potrebno zato jer pri množenju matrice s vektorom, moramo znati indeks vektora kojem odgovara unos u stupcu matrice.

Koristeći strukturu `matrix-row` da predstavlja matricu A , dok su p i q dva polja realnih vrijednosti, možemo pomoću algoritma navedenog u pseudo kodu 2 izračunati i pohraniti umnožak matrice i vektora.

Naravno, postoje i drugi načini za pohranu matrice. Postoje načini pomoću kojih se može iskoristiti simetrija matrice i tako uštedjeti prostor, ali u našem slučaju postoji način koji će uštedjeti još više prostora.

```
for(int n = 0; n < N; n++){
    q[n] = A[n].diag * p[n]
    for(int m = 0; m < A[n].numOff; m++){
        q[n] += A[n].offEntries[m] * p[ A[n].offIndices[m] ]
    }
}
```

Pseudo kod 2: Algoritam za rješavanje umnoška matrice i vektora, i.e. $q = Ap$.

Svi nedijagonalni unosi različiti od nule unutar matrice za tlak iznose -1, tako da njihovu vrijednost ne treba spremati jer je ona poznata. Nadalje, dijagonalni uvjeti nemaju veću vrijednost od 6, tako da je potrebno samo tri bita da se pohrani broj nedijagonalnih unosa i vrijednosti na dijagonali. Također, umjesto indeksa mreže ćelija s fluidom možemo pohraniti pomak (offset). Postoji šest načina da se MAC mreža prevede u mrežu indeksa ćelija s fluidom, i to tako da se prolazi kroz različite dimenzije MAC mreže različitim redoslijedom: *ijk*, *ikj*, *jki*, *jik*, *kij* ili *kji*. Ako se u prvi plan stavi najveća dimenzija, tada je najveća vrijednost pomaka koju možemo pohraniti u mreži ćelija s fluidom jednaka umnošku dva najmanja unosa. Dakle, prema ovoj shemi potrebno je 6 bita da bi se za svaku ćeliju s fluidom pohranio broj ne dijagonalnih uvjeta i nedijagonalne vrijednosti, a za svaki nedijagonalni uvjet je potrebno $\lceil \log_2(\min(IJ; IK; JK)) \rceil + 1$ bita. Onaj +1 bit je iz razloga što vrijednost pomaka može biti negativna.

Nećemo opisivati shemu za pohranu u detalje, zato jer je njezina primjena vježba premještanja bitova (bit shifting) i takav kod nije prenosiv. Sada kada smo opisali strukturu raspršene matrice, opisat ćemo algoritam za koji je takva struktura napisana.

5.4. Metoda konjugiranog gradijenta

Najjednostavniji način za rješavanje linearnog sustava $\mathbf{Ax} = \mathbf{b}$, je taj da izravno pronađemo \mathbf{A}^{-1} i izračunamo $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. Nažalost svojstva raspršene matrice \mathbf{A} ne dopuštaju računanje \mathbf{A}^{-1} iz jednostavnog razloga što \mathbf{A}^{-1} može biti potpuna matrica i množenje takve matrice sa \mathbf{b} postaje operacija koja ne može biti izvedena točno zbog veličine računa i pogrešaka prilikom računanja.

Jedna od opcija je da se koristi CG metoda, koja minimizira:

$$\frac{1}{2}\mathbf{x}^T\mathbf{Ax} - \mathbf{x}^T\mathbf{b}. \quad (40)$$

Ako uzmemo da je $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ dobijemo minimalnu vrijednost jednadžbe (40), koja iznosi $-\mathbf{b}^T\mathbf{A}^{-1}\mathbf{b}/2$. Prema tome, minimiziranje jednadžbe (40) i rješavanje $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ su jednaki problemi.

CG algoritam je prikazan pseudo kodom 3. Tijekom svake petlje algoritma, uzima se novo privremeno rješenje (spremljeno u \mathbf{x}). Odabire se takav vektor \mathbf{p} , da se pomicanjem po njemu za iznos α svaki puta više približimo rješenju. Sa svakim novim ponavljanjem odabire se takav vektor koji je okomit na vektore iz prethodnih ponavljanja. Sve dok je \mathbf{A} simetrično pozitivno definirana (a takva je ako sadrži barem jednu ćeliju ispunjenu zrakom), algoritam će pronaći rješenje. Za većinu simulacija koristimo $\varepsilon = 10^{-8}$, a $\text{iter}^{\text{max}} = 1000$. Početna pretpostavka pohranjena u \mathbf{x} je prethodno rješenje za tlak.

```

// N je duljina svih polja. A je polje tipa matrix-row, b
// je polje poznatih vrijednosti. U x polju su inicijalno
// spremljene početne vrijednosti; svakim prolaskom kroz
// petlju u x se spremaju nova privremena rješenja, i na
// kraju x sadrži rezultat. Polje r sadrži diferencijalne
// vrijednosti, a p i q sadrže međuvrijednosti.  $\alpha$ ,  $\beta$ ,  $\rho$ ,
//  $\rho^{\text{old}}$ , i  $b^{\text{norm}}$  su skalari.  $\text{iter}^{\text{max}}$  je maksimalan broj
// iteracija petlje. Kriterij  $\text{sqrt}(\rho) \leq \epsilon b^{\text{norm}}$ , služi za
// provjeru konvergencije i zaustavljanje gdje je  $\epsilon$ 
// apsolutna tolerancija

\rho = 0.0
bnorm = sqrt(b•b)
r = b - Ax // Pseudo kod 2
for(int i = 0; i < itermax; i++){
    \rhoold = \rho
    \rho = r•r
    if (\rho == 0.0 || sqrt(\rho) <= \epsilonbnorm)
        break; // Provjera konvergencije
    if(i == 0){
        p = r
    } else {
        \beta = \rho / \rhoold
        p *= \beta
        p += r
    }
    q = Ap // Pseudo kod 2
    \alpha = \rho / (p • q)
    x += \alphap
    r -= \alphaq
}

```

Pseudo kod 3: Algoritam konjugiranog gradijenta.

6. Kruti fluidi

Da bi efikasno simulirali međudjelovanje krutih tijela i fluida, kruta tijela ćemo tretirati kao da su fluidi uz promjenu brzina fluida unutar krutih tijela. Koristi ćemo tehniku poznatu kao Kruti fluidi, koja je dobila ime po načinu na koji se mijenjaju (ograničavaju) brzine unutar fluida da budu brzine krutih tijela.

6.1. Domene Krutih fluida



Slika 7: Žuto područje je domena fluida, plavo područje je domena krutih tijela; primijetimo da drveni blokovi koji ne dodiruju fluid ne spadaju u domenu krutih tijela.

Dva su dijela računске domene, vidi sliku 7. Dio koji sadrži samo fluide je \mathbb{F} , a unija ćelija koje sadrže kruta tijela je domena krutih tijela \mathbb{R} . Domene \mathbb{F} i \mathbb{R} su disjunktne i zajedno tvore računsku domenu $\mathbb{C} = \mathbb{F} \cup \mathbb{R}$. Granica koja razdvaja \mathbb{F} i \mathbb{R} je $\partial\mathbb{R}$.

6.2. Ograničenja dinamike krutih tijela

Kao što smo u prošlom poglavlju vidjeli postoje dva dijela računске domene. U ovom poglavlju ćemo opisati po čemu se razlikuje domena krutih tijela \mathbb{R} od domene fluida \mathbb{F} . Navier-Stokes-ove jednačbe rješavamo u obje domene, ali na \mathbb{R} primjenjujemo i ograničenje krutosti pomoću operatora deformacije $D[\]$.

Kada simuliramo kretanje krutog tijela, najlakše ga je predstaviti kao translaciju i rotaciju oko središta mase krutog tijela. Predstavljajući kretanje svih točaka unutar krutog tijela na ovaj način skrivamo kompleksnost dinamike krutog tijela. Svi simulatori dinamike krutog tijela implicitno provode krutost tijela ograničavajući njegovo kretanje na translacije i rotacije oko središta mase.

S druge strane, metoda Krutih fluida, rješava dinamiku gibanja krutih tijela pomoću Navier-Stokes-ovih jednačbi (1, 2). Navier-Stokes-ove jednačbe dopuštaju i deformacije u brzinama fluida tako da se krutost krutog tijela mora eksplicitno primijeniti pomoću Lagrange-ovog multiplikatora. Uvjet krutosti je vrlo sličan uvjetu nestlačivosti. Uvjet krutosti je ipak jače ograničenje jer ne dozvoljava ni divergenciju ni deformaciju. Krutost tijela uvjetuje da za svaku točku y_j unutar krutog tijela vrijedi sljedeća relacija:

$$\dot{\mathbf{y}}_j = \mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}_j \quad (41)$$

za neku konstantnu vrijednost \mathbf{v} i $\boldsymbol{\omega}$. U danoj jednačbi $\dot{\mathbf{y}}_j$ je brzina u točki y_j , \mathbf{r}_j je vektor koji pokazuje od središta mase krutog tijela, \mathbf{x} , do y_j , \mathbf{v} je translacijska brzina, a $\boldsymbol{\omega}$ je kutna brzina oko \mathbf{x} duž osi $\boldsymbol{\omega}/|\boldsymbol{\omega}|$ i vrijednošću $|\boldsymbol{\omega}|$.

Uvjet krutosti se može izraziti pomoću operatora deformacije, $D[\]$, definiranog za vektorsko polje $\mathbf{u} = (u, v, w)$ sa

$$\begin{aligned}
D[\mathbf{u}] &= \frac{1}{2} [\nabla \mathbf{u} + \nabla \mathbf{u}^T] \\
&= \frac{1}{2} \begin{bmatrix} 2u_x & (v_x + u_y) & (w_x + u_z) \\ (v_x + u_y) & 2v_y & (w_y + v_z) \\ (w_x + u_z) & (w_y + v_z) & 2w_z \end{bmatrix}
\end{aligned} \tag{42}$$

Ovaj 3 x 3 simetrični tenzor mjeri prostornu deformaciju od \mathbf{u} . Ograničenje

$$D[\mathbf{u}] = 0 \text{ u } \mathbb{R} \tag{43}$$

garantira da je gibanje unutar \mathbb{R} doista i gibanje krutog tijela. Ne govori nam kakvo je to gibanje, ali znamo da mora vrijediti jednadžba (41), tj. ako znamo pozicije točaka gdje je \mathbf{u} definirano i njihov relativan položaj prema centru mase krutog tijela, onda postoje translacijska brzina \mathbf{v} i kutna brzina ω koje možemo izračunati.

6.3. Osnovne jednadžbe krutih fluida

U ovom poglavlju predstaviti ćemo osnovne jednadžbe koje čine samu srž metode krutih fluida. Jednadžba očuvanja momenta je definirana kao

$$\mathbf{u}_t = -(\mathbf{u} \cdot \nabla) \mathbf{u} + \nu \nabla^2 \mathbf{u} - \frac{1}{\rho_f} \nabla p + \mathbf{f} \text{ u } \mathbb{F} \tag{44}$$

u domeni fluida i

$$\mathbf{u}_t = -(\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla \cdot \Pi - \frac{1}{\rho_r} \nabla p + \mathbf{f} \text{ u } \mathbb{R} \tag{45}$$

u domeni krutih tijela, gdje je ρ_f gustoća fluida, a ρ_r gustoća krutog tijela. Izraz za difuziju je uklonjen iz jednadžbe (45) zato što uvjet krutosti eliminira Newton-ovu viskoznu disipaciju, ali zato je tu dodatni član zbog deformacijskog naprezanja unutar krutog tijela koji je potreban da bi zadržali krutost. Implicitno definiramo Π kao taj dodatni član deformacijskog naprezanja.

Budući da je ograničavanje deformacije koje primjenjujemo na domeni krutog tijela jače od ograničavanja divergencije, možemo primijeniti uvjet ograničavanja divergencije u cijeloj domeni:

$$\nabla \cdot \mathbf{u} = 0 \text{ u } \mathbb{C}. \quad (46)$$

Grafični uvjeti između krutih tijela i fluida definirani su sa

$$\mathbf{u} = \mathbf{u}_i \text{ i } (2\rho_f\nu\mathbf{D}[\mathbf{u}] - p\mathbf{1}) \cdot \mathbf{n} = \mathbf{t} \text{ u } \partial\mathbb{R} \quad (47)$$

gdje je $\mathbf{1}$ jedinični tenzor, \mathbf{u}_i i \mathbf{n} su brzina i normala na $\partial\mathbb{R}$, i \mathbf{t} je sila kojom fluid djeluje na tijelo koja je zbroj projiciranog viskozne napetosti i tlaka. Možemo napisati sličan uvjet za silu kojom kruto tijelo djeluje na fluid koja mora biti jednaka i suprotna sili \mathbf{t} . Ipak, nećemo morati direktno primjenjivati grafične uvjete iz jednažbe (47) jer će oni biti približno obuhvaćeni metodom projekcije opisanom kasnije u tekstu.

6.4. Implementacija Krutih fluida

Jednažbe (43 – 47) su osnovne jednažbe za sve pokretne objekte u ovoj simulaciji, i fluida i krutih tijela. Jednažbe rješavamo u tri koraka. prvo riješimo Navier-Stokes-ove jednažbe (1, 2) za cijelu domenu $\mathbb{C} = \mathbb{F} \cup \mathbb{R}$. Tijekom prvog koraka kruta tijela tretiramo kao da su fluidi. Zatim izračunamo sile krutih tijela koje potječu od kolizija i relativne gustoće i dodamo ih ćelijama koje se nalaze unutar krutih tijela. I na kraju ograničimo brzine na tim lokacijama da budu brzine krutih tijela. Ta tri koraka pomiču simulaciju u vremenu iz $\mathbf{u}^n \rightarrow \mathbf{u}^{n+1}$, prolazeći pritom kroz dva međustanja \mathbf{u}^* i $\hat{\mathbf{u}}$.

6.4.1. Rješavanje Navier-Stokes-ovih jednadžbi: $\mathbf{u}^n \rightarrow \mathbf{u}^*$

Prvi korak je rješavanje Navier-Stokes-ovih jednadžbi. Taj problem smo detaljno opisali u 3. poglavlju. Sastoji se od dva dijela. Prvi dio je izračunavanje približne brzine (jednadžbe 25 – 27). Drugi korak je projiciranje tlaka kako bi se dobile brzine bez divergencije. Pri tome treba paziti na dva uvjeta za vremenski korak Δt . Prvi je tzv. CFL uvjet (jednadžba 19) a drugi (jednadžba 20) nastaje zbog korištenja Euler-ove metode za izračun difuzije $\nu \nabla^2 \mathbf{u}$. Nakon završetka ovog koraka imamo polje brzina \mathbf{u}^* u \mathbb{C} bez divergencije, ali to nisu konačne brzine zato što u obzir nisu uzete sile kolizije i sile koje nastaju zbog relativne gustoće niti su brzine u \mathbb{R} ograničene na kretanja krutih tijela.

6.4.2. Izračunavanje sila krutih tijela: $\mathbf{u}^* \rightarrow \hat{\mathbf{u}}$

Tijekom jednog vremenskog koraka, manipulator krutih tijela primjenjuje sile kolizije dok ažurira njihove pozicije. Te sile moraju biti uključene u polje brzine kako bi pravilno prenesle moment između domena krutih tijela i fluida.

Za svaku silu kolizije, \mathbf{F}_j , koja se primjeni na neko od N krutih tijela, čuvamo sumu akceleracija za pojedino tijelo kreiranih tijekom vremenskog koraka:

$$\mathbf{A}_c = \sum_j \frac{\mathbf{F}_j}{M_i}, \quad (48)$$

gdje je $i \in \{1, 2, \dots, N\}$, a M_i je masa krutog tijela na koji je sila primijenjena.

Analogno, za svaku silu primijenjenu u točki \mathbf{p}_j , čuvamo sumu kutnih akceleracija koje ta sila kreira oko središta mase datog tijela:

$$\alpha_c = \sum_j \mathbf{I}_i^{-1} [(\mathbf{p}_j - \mathbf{x}_i) \times \mathbf{F}_j], \quad (49)$$

gdje je \mathbf{I}_i moment inercije i -tog krutog tijela, u njegovom trenutnom položaju, a \mathbf{x}_i je centar mase tog tijela.

Sile koje nastaju zbog relativne gustoće se isto moraju uračunati. Relativna gustoća krutog tijela je omjer njegove gustoće naspram gustoće fluida koji ga okružuje, ρ_r/ρ_f . Ako je relativna gustoća veća od 1, tijelo će potonuti, a ako je manja od 1 tijelo će plutati. Što je relativna gustoća veća to fluid sve teže pomiče kruto tijelo. Relativna gustoća i sile kolizije u \mathbb{R} su sadržane u slijedećem izrazu:

$$\mathbf{S} = \rho_r \mathbf{A}_c + \mathbf{r}_i \times \rho_r \alpha_c - (\rho_r - \rho_f) \left[\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + (\mathbf{u}^* \cdot \nabla) \mathbf{u}^* - \mathbf{f} \right], \quad (50)$$

gdje su vektori $\mathbf{r}_i = \mathbf{y}_i - \mathbf{x}_i$ usmjereni od centra mase i -tog tijela prema točkama mreže, \mathbf{y}_i , unutar domene datog krutog tijela \mathbb{R}_i . Rješenje jednadžbe (50) je direktno jer su sve varijable sa desne strane poznate.

Koristeći \mathbf{S} izračunamo novo polje vektora brzina:

$$\hat{\mathbf{u}} = \mathbf{u}^* + w \frac{\Delta t}{\rho_r} \mathbf{S}, \quad (51)$$

gdje je w broj između 0 i 1 koji predstavlja postotak volumena ćelije zauzet krutim tijelom.

Dobivena brzina $\hat{\mathbf{u}}$ ima svojstvo očuvanja momenta, ali još uvijek to nije brzina krutog tijela tako da moramo završiti još jedan korak.

6.4.3. Ograničavanje brzina na brzine krutih tijela: $\hat{\mathbf{u}} \rightarrow \mathbf{u}^{n+1}$

Zadnji korak je pronalazak nepoznate sile \mathbf{R} koja održava krutost tijela kako bi brzine stvarno i bile brzine krutog tijela. Konačna brzina je onda:

$$\mathbf{u}^{n+1} = \hat{\mathbf{u}} + \frac{\Delta t}{\rho_r} \mathbf{R} \mathbf{u} \mathbb{R}, \quad (52)$$

ali još uvijek ne znamo kako izračunati \mathbf{R} .

Jednadžba (52) je projekcija koja osigurava krutost na sličan način kao što projekcija tlaka, jednadžba (29), osigurava uvjet divergencije. Ograničenje primijenjeno projekcijom tlaka je $\nabla \cdot \mathbf{u} = 0$, a kao Lagrange-ov multiplikator za osiguranje ograničenja je korišten tlak p . Da bi našli jednadžbu za p uzeli smo divergenciju jednadžbe (29) i uvrstili je u jednadžbu (31). Ograničenje koje moramo primijeniti sa projekcijom krutosti je jednadžba (43), a Lagrange-ov multiplikator je \mathbf{R} , tako da primjenom jednažbe (43) na jednadžbi (52) dobivamo jednadžbu za \mathbf{R} :

$$\mathbf{D}[\mathbf{u}^{n+1}] = \mathbf{D} \left[\hat{\mathbf{u}} + \frac{\Delta t}{\rho_r} \mathbf{R} \right] = 0 \quad (53)$$

koja nam govori da je $\hat{\mathbf{u}} + \Delta t / \rho_r \mathbf{R}$ traženo gibanje krutog tijela. Alternativno možemo rastaviti $\hat{\mathbf{u}}$ u \mathbb{R} na dva dijela:

$$\hat{\mathbf{u}} = \hat{\mathbf{u}}_R + \hat{\mathbf{u}}', \quad (54)$$

gdje je $\hat{\mathbf{u}}_R$ tražena brzina krutog tijela, a

$$\hat{\mathbf{u}}' = - \frac{\Delta t}{\rho_r} \mathbf{R} \quad (55)$$

nastaje zbog napetosti unutar \mathbb{R} i uvjetuje krutost tijela.

Traženo rješenje jednadžbi (53, 55) za \mathbf{R} i $\hat{\mathbf{u}}'$ mora očuvati moment i zbog toga ga možemo direktno izračunati. Ako napišemo jednadžbu (41) kao uniju širom pojedinog krutog tijela dobijemo slijedeći izraz:

$$\hat{\mathbf{u}}_R = \bigcup_i (\hat{\mathbf{v}}_i + \hat{\omega}_i \times \mathbf{r}_i) \quad (56)$$

za neki $\hat{\mathbf{v}}_i$ i $\hat{\omega}_i$. Pošto moment mora biti očuvan, $\hat{\mathbf{v}}_i$ i $\hat{\omega}_i$ za pojedino kruto tijelo dobijemo direktnim integriranjem $\hat{\mathbf{u}}$ unutar traženog krutog tijela \mathbb{R}_i pomoću jednadžbi:

$$M_i \hat{\mathbf{v}}_i = \int_{\mathbb{R}_i} \rho_i \hat{\mathbf{u}} dy_i, \quad (57)$$

$$\mathbf{I}_i \hat{\omega}_i = \int_{\mathbb{R}_i} \mathbf{r}_i \times \rho_i \hat{\mathbf{u}} dy_i, \quad (58)$$

gdje M_i , \mathbf{I}_i i ρ_i su masa, moment inercije i gustoća i -tog krutog tijela, a dy_i je volumen ćelije ispunjen krutim tijelom:

$$dy_i = \Delta x^3 w. \quad (59)$$

Jednadžbe (57, 58) se izračunavaju zbrajanjem adekvatnih izraza za svaku ćeliju koja je potpuno ili djelomično unutar i -te domene krutog tijela \mathbb{R}_i .

Zato što jednadžba (52) mora očuvati moment, možemo koristeći jednadžbe (57, 58) direktno izračunati brzinu krutog tijela $\hat{\mathbf{u}}_R$. Nakon toga tu brzinu raspodijelimo da bi dobili konačno rješenje:

$$\mathbf{u}^{n+1} = (1 - w)\hat{\mathbf{u}} + w\hat{\mathbf{u}}_R, \quad (60)$$

koje očuva krutost tijela i moment unutar \mathbb{R} .

6.5. Pomicanje računске domene

Metoda Krutih fluida koristi diskretnu računsku mrežu sa konstantnim razmakom gdje su vektori brzina na stranicama ćelija, a tlak u sredini. To je MAC mreža koju smo opisali ranije u tekstu. Domene fluida i krutih tijela se pomiču svaki vremenski korak, tako da prije nego što možemo riješiti jednađbe moramo odrediti novu računsku domenu i identificirati ćelije u \mathbb{F} , kao i ćelije u \mathbb{R} . Moramo odrediti i prostor koji niti \mathbb{F} niti \mathbb{R} ne mogu zauzimati – nepomične granice. Da bi pomakli računsku domenu, \mathbb{C} , moramo pomaknuti i \mathbb{F} i \mathbb{R} . Pomicanje domene fluida je već opisano ranije u tekstu, tako da ćemo odmah prijeći na pomicanje domene krutih tijela.

Algoritam krutih fluida pronalazi novi \mathbb{R} pomicanjem krutih tijela. Kako će se pomicati kruta tijela, detektirati kolizije sa drugim tijelima i granicama je tema sama za sebe i jedan od mogućih pristupa je opisan u [4]. Prije nego pomičemo kruta tijela moramo odrediti koja od njih dodiruju fluid a koja ne. Samo tijela koja su u kontaktu sa fluidom spadaju u domenu krutih tijela \mathbb{R} i njihove brzine se dobivaju integriranjem jednađbi 57 i 58. Proučimo te jednađbe malo bolje za jedno kruto tijelo u domeni \mathbb{R} , sa masom M , gustoćom ρ , momentom inercije \mathbf{I} i centrom mase \mathbf{x} :

$$M\mathbf{v} = \int_{\mathbb{R}} \rho \mathbf{u} dy, \quad (61)$$

$$\mathbf{I}\omega = \int_{\mathbb{R}} \mathbf{r} \times \rho \mathbf{u} dy, \quad (62)$$

gdje je $\mathbf{r} = \mathbf{y} - \mathbf{x}$, a \mathbf{y} je trenutna točka u \mathbb{R} koja se integrira. Svrha ove dvije jednađbe je da se pronađe trenutna brzina translacije \mathbf{v} i kutna brzina ω , uz danu brzinu fluida \mathbf{u} . Nekoliko koraka je potrebno da bi izračunali diskretne integrale iz jednađbi 61 i 62. Prvo je potrebno za svaki komad krutog tijela koji se nalazi unutar ćelije odrediti volumen i centar mase. Slijedeći korak je

izračunavanje diskretnog moment inercije i mase, i za vrijeme skupljanja tih podataka rješavaju se jednačbe. Pogledajmo malo detaljnije te korake.

U poglavlju 5.1 smo rekli da je \mathbb{R} skupina ćelija koje sadrže dijelove krutih tijela. Zapravo, kada pomaknemo kruto tijelo, uz listu ćelija držimo još neke podatke. Dva važna podatka koja moramo znati za svaku ćeliju su volumen krutog tijela koji je sadržan u toj ćeliji i centar mase tog dijela krutog tijela. Kako odrediti te podatke ovisi o složenosti krutog tijela. Metoda koju smo koristili je slijedeća. Za svaki vrh ćelije smo odredili da li je unutar ili izvan tijela te na temelju omjera broja vrhova izvan i unutar tijela te njihovih položaja smo odredili i volumen i centar mase. Za jednostavnija kruta tijela poput kvadra ili kugle određivanje da li je neka točka unutar ili izvan tijela je jednostavna matematička operacija, dok je za kompliciranije oblike tijela pripadnost potrebno odrediti metodom praćenja zrake.

Kada smo izračunali volumen i centar mase za svaku ćeliju koja sadržava kruto tijelo, diskretni integral je jednostavno suma svih volumena pomnožena sa gustoćom krutog tijela. Moment inercija je malo kompliciraniji za izračunat. Svaka od $n \in \{1, 2, \dots, N\}$ ćelija okupiranih krutim tijelom ima masu m_n i centar mase \mathbf{x}_n . Uz pomoć centra mase krutog tijela \mathbf{x} izračunamo pozicioni vektor $\mathbf{r}_n = \mathbf{x}_n - \mathbf{x} = (x_n, y_n, z_n)$, koji je usmjeren od centra mase krutog tijela do centra mase dijela krutog tijela koji se nalazi unutar ćelije n . Diskretni moment inercije \mathbf{I}_n je:

$$\mathbf{I}_n = m_n \begin{bmatrix} y_n^2 + z_n^2 & -x_n y_n & -x_n z_n \\ -x_n y_n & x_n^2 + z_n^2 & -y_n z_n \\ -x_n z_n & -y_n z_n & x_n^2 + y_n^2 \end{bmatrix} \quad (63)$$

```

M = 0.0
I = 0.0
Mv = 0.0
Iomega = 0.0
for(n = 0; n < N; n++)
{
  m[n] = volume[n] * rho
  r[n] = x[n] - x
  M += m[n]
  Mv += m[n] * u[n]
  I[n][0,0] = r[n].y * r[n].y + r[n].z * r[n].z
  I[n][0,1] = -r[n].x * r[n].y
  I[n][0,2] = -r[n].x * r[n].z
  I[n][1,0] = -r[n].x * r[n].y
  I[n][1,1] = r[n].x * r[n].x + r[n].z * r[n].z
  I[n][1,2] = -r[n].y * r[n].z
  I[n][2,0] = -r[n].x * r[n].z
  I[n][2,1] = -r[n].y * r[n].z
  I[n][2,2] = r[n].x * r[n].x + r[n].y * r[n].y
  I += I[n]
  Iomega += m[n] * Vector_product(r[n], u[n])
}
v = Mv / M
omega = Iomega / I /*matrix inversion with LU decomposition*/

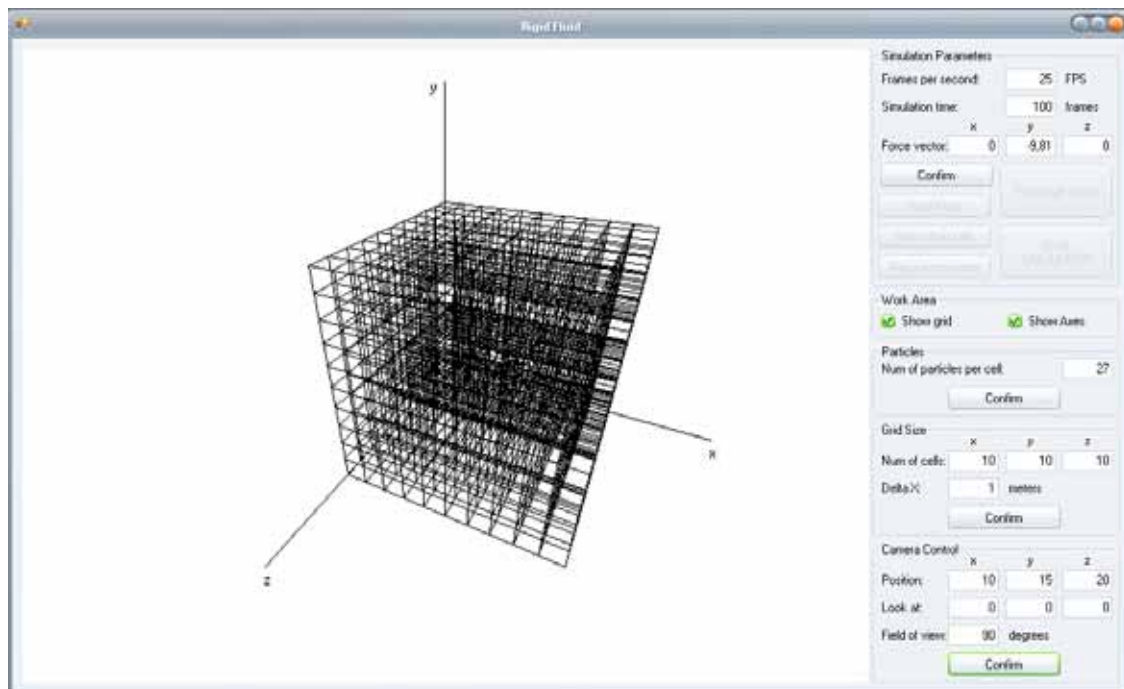
```

Pseudo kod 4: Algoritam za računanje diskretnih integrala u jednadžbama (61, 62)

Zadnja dva reda pseudo koda 4 računaju brzine krutog tijela koje se onda koriste za pomicanje krutih tijela. Tijekom jednog vremenskog koraka zbrajamo sve sile koje nastaju zbog kolizije krutih tijela (za svako tijelo posebno) i spremamo ih u A_c i α_c . Te varijable predstavljaju akceleracije krutog tijela nastale zbog kolizije i koristimo ih u jednadžbi (50).

7. Programska podrška

Program za simulaciju međudjelovanja krutih tijela i fluida je napisan u programskom jeziku C#.



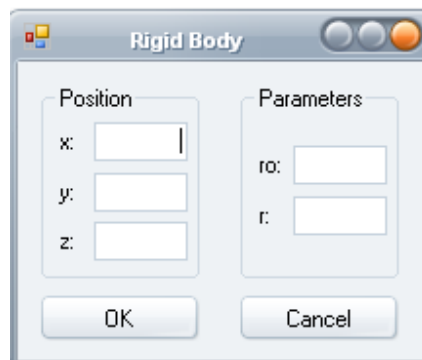
Slika 8: Programsko sučelje

Programsko sučelje se sastoji od dva glavna dijela. Lijeva strana programa služi za prikaz scene dok se na desnoj strani nalaze alati za upravljanje simulacijom, slika 8. Ti alati odvojeni su u pet grupa koje su redom:

- simulation parameters; u ovoj grupi se nalaze parametri za određivanje trajanja simulacije, broj sličica u sekundi, vektora vanjske sile koja djeluje u simulaciji, kao i tipke za određivanje parametara krutog tijela, ćelija ispunjenih fluidom i pokretanje simulacije
- work area; u ovoj grupi imamo dvije opcije koje služe kao pomoć pri određivanju pogleda na scenu

- particles; ova grupa sadrži samo jedan parametar – broj čestica po ćeliji
- grid size; u ovoj grupi biramo broj ćelija po svakoj dimenziji kao i parametar Δx
- camera control; ova grupa sadrži parametre za kontrolu kamere: pozicija kamere, točka pogleda, i širina pogleda.

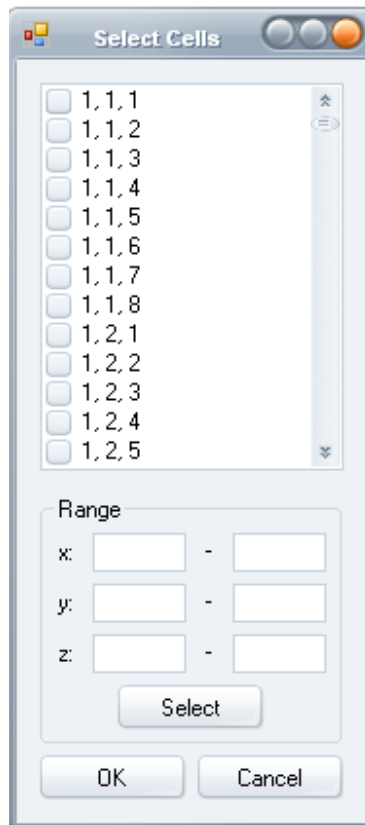
Nakon pokretanja programa prvi korak je određivanje parametara mreže, broj ćelija i njihova veličina Δx . Zatim se zada broj čestica po ćeliji te parametri kamere. Svaki od ovih koraka se mora potvrditi pritiskom na tipku *Confirm*. Sljedeći korak je određivanje vremena simulacije, vremenskog koraka (vremenski korak se određuje brojem sličica u sekundi) i vanjske sile koja djeluje u sustavu. Kada se ti parametri potvrde tipkom *Confirm* omogućuje se odabir parametara krutog tijela, slika 9.



Slika 9: Parametri krutog tijela

Ti parametri su pozicija (x, y, z) , relativna gustoća (ρ) i radijus (r) . Sljedeći korak je odabir ćelija sa fluidom, slika 10.

Odabir je moguć pojedinačnim selektiranjem ćelija ili odabirom po rasponu, dozvoljene vrijednosti su od 1 do $(N - 2)$ gdje N predstavlja broj ćelija u smjeru osi x , y ili z . Ćelije sa indeksima 0 i $N - 1$ su predstavljaju čvrste granice prostora simulacije.



Slika 10: Odabir ćelija sa fluidom

Nakon što su svi parametri određeni pritiskom na tipku *Prepare simulation* inicijaliziramo simulaciju i omogućimo njeno pokretanje. Simulaciju možemo pokretati za jedan vremenski korak, *Run single frame*, ili za zadano trajanje, *Run simulation*. Izlaz iz programa su slike čiji se parametri određuju formatom slike u datoteci „bitmap.bmp“. Glavni parametar je razlučivost slike i o njemu znatno ovisi prikaz izlaznih datoteka jer jedna čestica zauzima jedan piksel.

8. Rezultati

Implementirana metoda krutih fluida daje realistične i fizikalno ispravne rezultate. U nastavku ćemo prikazati neke od rezultata dobivenih ovom metodom.

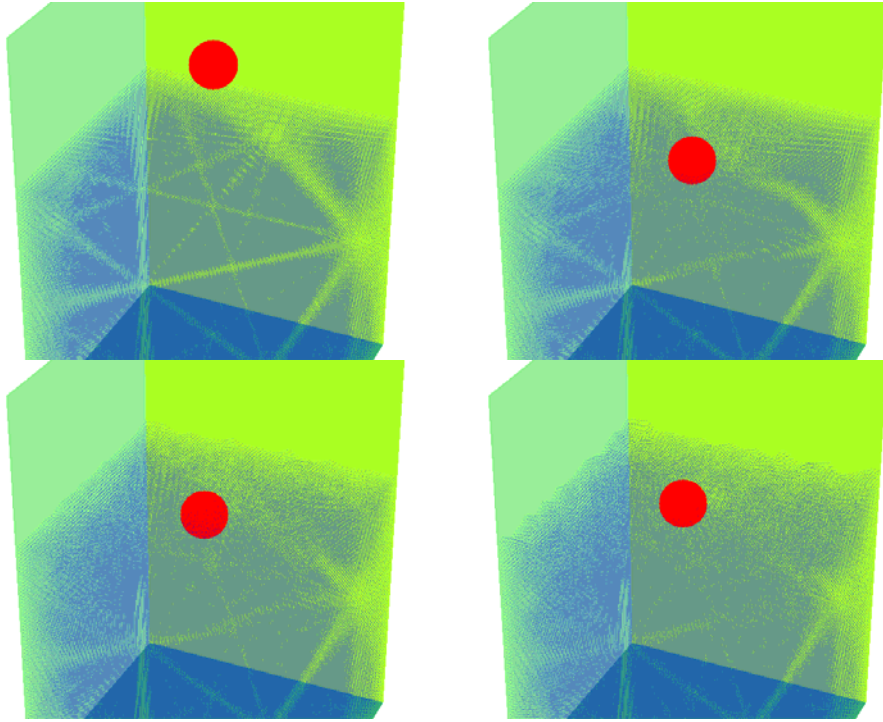
Za sve simulacije vrijede sljedeći parametri:

Δx	0,2
Sličica u sekundi	25
Vrijeme simulacije	100
Vanjska sila (x,y,z)	0, -9.81, 0
Radius krutog tijela	0.3

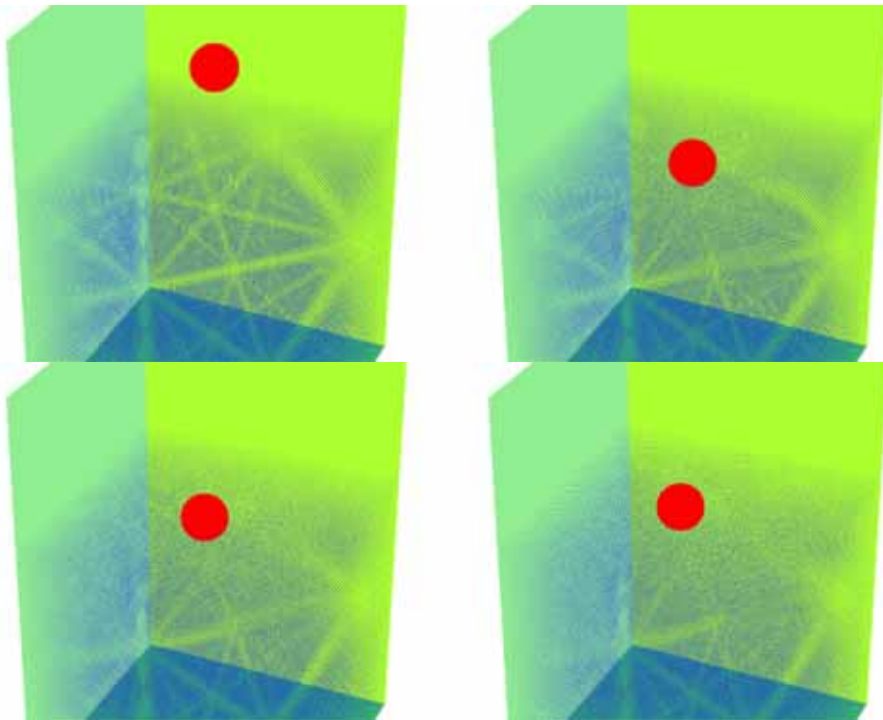
Ostali parametri po simulaciji:

Simulacija broj	Mreža lxJxK	Broj čestica po ćeliji	Položaj krutog tijela (x,y,z)	Relativna gustoća	Vrijeme izvođenja (min:sek)
1	20x30x20	64	2, 5, 2	0.7	7:13
2	20x30x20	27	2, 5, 2	0.7	4:23
3	20x30x20	64	2, 5, 2	2	6:55
4	20x30x20	27	2, 5, 2	2	4:24
5	20x20x20	64	3.3, 0.6, 2	0.7	3:36
6	20x20x20	64	3.3, 0.6, 2	2.6	3:35

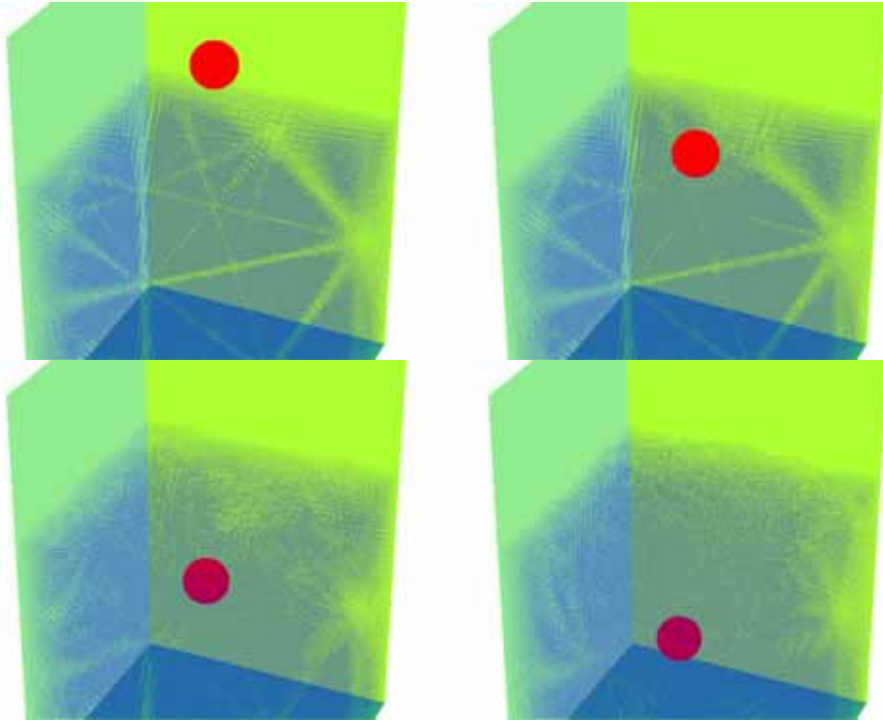
Testno računalo: Athlon 64 3200+ @ 2.0 GHz, 1 GB RAM-a



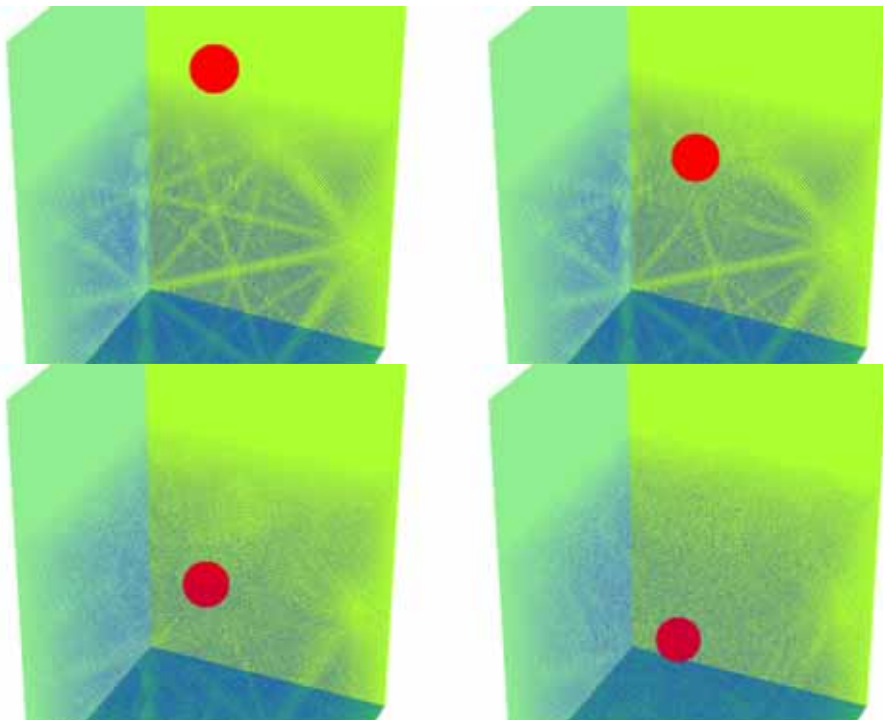
Slika 11: Simulacija broj 1



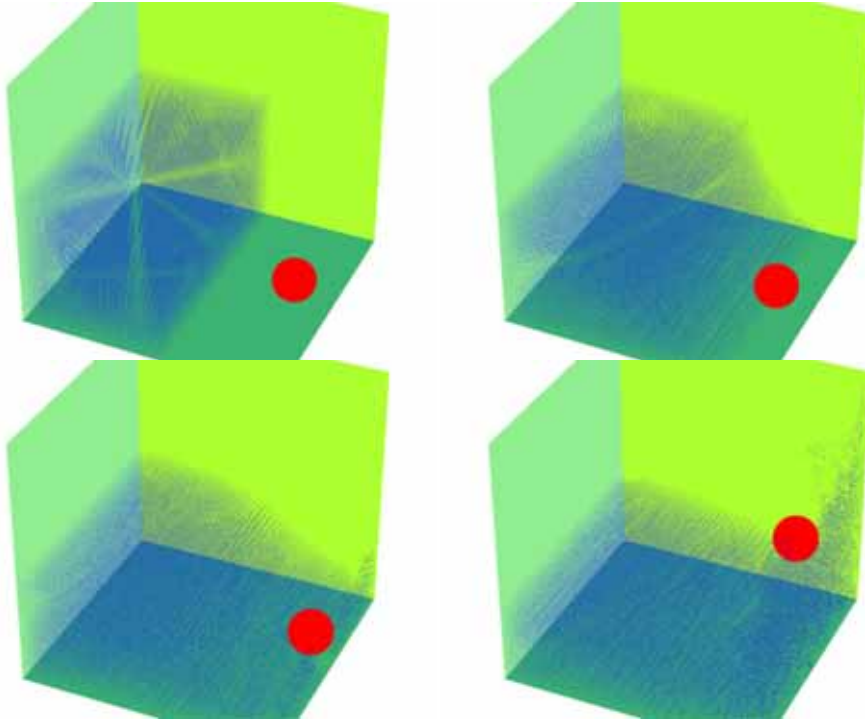
Slika 12: Simulacija broj 2



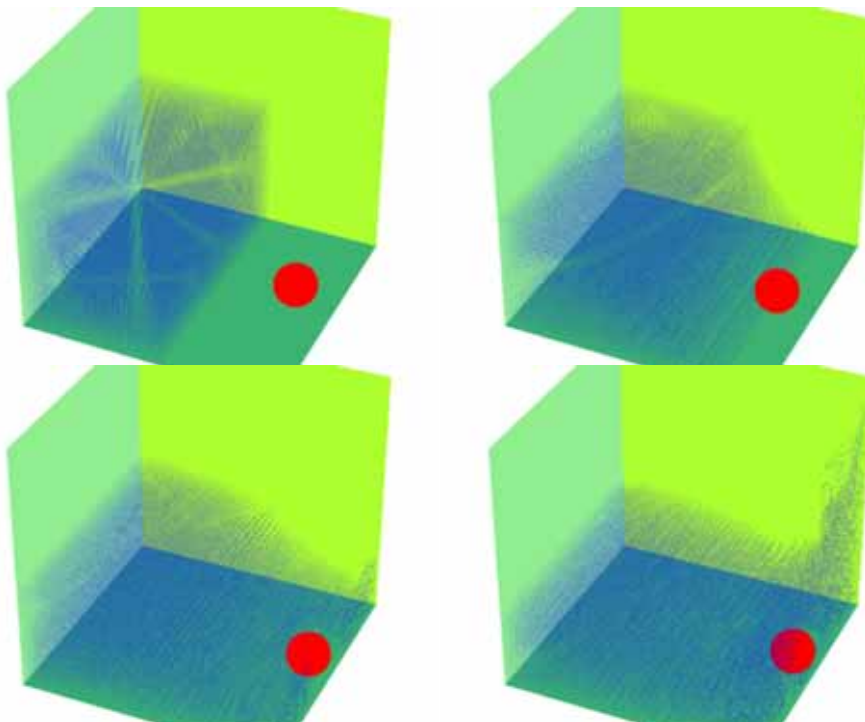
Slika 13: Simulacija broj 3



Slika 14: Simulacija broj 4



Slika 15: Simulacija broj 5



Slika 16: Simulacija broj 6

9. Zaključak

U ovom radu predstavili smo metodu krutih fluida kao način simuliranja međudjelovanja krutih tijela i fluida. Glavna snaga krutih fluida leži u efikasnom baratnju krutim tijelima, zahtijevajući relativno malo dodatnog računskog vremena na vrijeme potrošeno na simulaciju samog fluida. To smo postigli korištenjem distribuiranih Lagrange-ovih multiplikatora za ograničavanje krutosti tijela, kao i metodom rastavljanja operatora gdje smo prvo izračunali brzine toka bez divergencije u računskoj domeni, a zatim smo u domeni krutih tijela brzine ograničili na brzine krutih tijela. U određenim situacijama ta tehnika ne daje dobre rezultate, uzmimo za primjer okomitu cijev koja se sužava i lopticu koja je zapela u cijevi i predstavlja čep, u takvom bi slučaju voda prošla kroz lopticu, te je u tim i sličnim situacijama potrebno istovremeno obaviti i ograničavanje toka bez divergencije i ograničavanje krutosti tijela. Kruta tijela koja zauzimaju vrlo malo ćelija su isto problematična za simulaciju. Tanka stjenka, tanja od širine jedne ćelije ne može spriječiti fluid pri prolasku kroz nju te je u takvim slučajevima potrebno koristiti finiju mrežu. Jedno od mogućih unaprjeđenja je korištenje implicitnih metoda za rješavanje difuzije fluida i semi-Lagrangian metode za advekciju fluida koje su stabilne i za velike vremenske korake [5].

10. Popis slika

Slika 1: Čelija u MAC mreži.	5
Slika 2: Čvrste granične ćelije na lijevoj strani MAC mreže.	12
Slika 3: Čelije sa fluidom u MAC mreži (lijevo) i pripadajuća matrica projekcija tlaka A (desno), i indeksi pripadajuće mreže ćelija sa fluidom (sredina). Bijeli unosi u matrici su nule.	20
Slika 4: Slika 3 sa označenom jedinom ćelijom sa fluidom koja je sa svih strana okružena ćelijama sa fluidom.	21
Slika 5: Slika 3 sa stražnjom-donjom-lijevom ćelijom sa fluidom označenom sivo. Ova kutna ćelija je posebna jer ima tri stranice koje graniče sa čvrstim nepremostivim ćelijama. Neumann-ovi granični uvjeti moraju biti postavljeni na tim čvrstim preprekama.	22
Slika 6: Slika 3 sa ćelijom sa fluidom koja strši u zrak označenom sivom bojom. Ova ćelija je posebna jer je okružena sa praznim ćelijama (ispunjenih zrakom) sa svih strana osim jedne. Dirichlet-ovi granični uvjeti moraju biti postavljeni za sve okolne prazne ćelije.	23
Slika 7: Žuto područje je domena fluida, plavo područje je domena krutih tijela; primijetimo da drveni blokovi koji ne dodiruju fluid ne spadaju u domenu krutih tijela.	29
Slika 8: Programsko sučelje	40
Slika 9: Parametri krutog tijela	41
Slika 10: Odabir ćelija sa fluidom	42
Slika 11: Simulacija broj 1	44
Slika 12: Simulacija broj 2	44

Slika 13: Simulacija broj 3	45
Slika 14: Simulacija broj 4	45
Slika 15: Simulacija broj 5	46
Slika 16: Simulacija broj 6	46

11. Literatura

1. Carlson, M; Mucha, J. P; Turk, G: „Rigid Fluid: Animating the Interplay Between Rigid Bodies and Fluid“, ACM SIGGRAPH 2004 conference proceedings
2. Harlow, F. H; Welch, J. E: „Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface“, Physics of Fluids, vol. 8, pp. 2182–2189, Prosinac 1965.
3. Welch, J. E; Harlow, F. H; Shannon, J. P; Daly, B. J: “THE MAC METHOD a computing technique for solving viscous, incompressible, transient fluid-flow problems involving free surfaces”, Report LA-3425, Los Alamos Scientific Laboratory, 1965.
4. Guendelman, E; Bridson, R; Fedkiw, R. P: „Nonconvex rigid bodies with stacking“, ACM Transactions on Graphics, vol. 22, pp. 871–878, Srpanj 2003.
5. Stam, J: „Stable fluids“, Proceedings of SIGGRAPH 99, Computer Graphics Proceedings, Annual Conference Series, pp. 121–128, Kolovoz 1999.
6. Patankar, N. A: „A formulation for fast computations of rigid particulate flows“, Center for Turbulence Research Annual Research Briefs 2001, pp. 185–196, 2001.
7. Foster, N; Metaxas, D: „Realistic animation of liquids“, Graphical Models and Image Processing, vol. 58, no. 5, pp. 471–483, 1996.

12. Sažetak

(Animacija međudjelovanja krutih tijela i fluida)

U ovom radu je predstavljena metoda Krutih fluida, tehnika animiranja međudjelovanja krutih tijela i nestlačivih fluida. Korišteni su Lagrangeov-i multiplikatori da bi osigurali međudjelovanje koje kreira realistična kretanja i za kruta tijela i za fluide za vrijeme njihova međudjelovanja. Metoda je nazvana Kruti fluidi zato što tretiramo kruta tijela kao da su fluidi. Krutost takvih tijela je postignuta ograničavanjem brzina unutar njih tako da one bude brzine krutih tijela. Više krutih tijela različite gustoće (drvo, olovo, i sl.) se mogu kombinirati u istoj simulaciji.

Ključne riječi: MAC (marker and cell), kruti fluidi, računalna dinamika fluida

13. Abstract

(Animating the interplay between rigid bodies and fluid)

We present the Rigid Fluid method, a technique for animating the interplay between rigid bodies and viscous incompressible fluid. We use Lagrange multipliers to ensure two-way coupling that generates realistic motion for both the solid objects and the fluid as they interact with one another. This method is called the rigid fluid method because the simulator treats the rigid objects as if they were made of fluid. The rigidity of such an object is maintained by identifying the region of the velocity field that is inside the object and constraining those velocities to be rigid body motion. Many solid objects of different densities (e.g., wood or lead) can be combined in the same animation.

Keywords: MAC (marker and cell), rigid fluid, computational fluid dynamics