

SVEUČILIŠTE U ZAGREBU  
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

DIPLOMSKI RAD br. 1723

**FIZIKALNA SIMULACIJA SUDARA  
KONKAVNIH KRUTIH TIJELA**

Igor Popovski

Zagreb, svibanj 2008.



# Sadržaj:

1.	Uvod .....	1
2.	Prikaz tijela .....	2
2.1.	Polje udaljenosti.....	2
3.	Detekcija sudara .....	6
3.1.	Određivanje predznaka.....	6
3.2.	Udaljenost točke od trokuta .....	8
4.	Dinamika krutog tijela .....	12
4.1.	Pozicija i orijentacija .....	12
4.2.	Linearna i kutna brzina .....	13
4.3.	Brzina točke na tijelu.....	15
4.4.	Orijentacija pomoću kvaterniona.....	15
4.5.	Masa tijela i centar mase .....	16
4.6.	Moment inercije .....	17
5.	Korak integracije .....	19
6.	Rješavanje sudara.....	22
7.	Reakcija na sudar.....	24
7.1.	Statičko i dinamičko trenje .....	26
7.2.	Trenje prilikom kotrljanja i vrtnje .....	27
8.	Rješavanje kontakata .....	29
8.1.	Koeficijent elastičnosti .....	29
8.2.	Propagirajući model .....	30
8.3.	Kontaktni graf.....	30
8.4.	Propagiranje šoka.....	33
9.	Zglobovi .....	36
9.1.	Translacijski zglobovi.....	36

9.1.1.	Translacijski zglob s ograničena tri stupnja slobode .....	36
9.1.2.	Translacijski zglob s ograničena dva stupnja slobode.....	38
9.1.3.	Translacijski zglob s ograničenim jednim stupnjem slobode .....	40
9.2.	Rotacijski zglobovi .....	40
9.2.1.	Rotacijski zglob s ograničena tri stupnja slobode.....	41
9.2.2.	Rotacijski zglob s ograničena dva stupnja slobode .....	42
9.2.3.	Rotacijski zglob s ograničenim jednim stupnjem slobode .....	43
10.	Hibridni model .....	44
11.	Implementacija .....	45
12.	Rezultati .....	47
13.	Zaključak .....	56
14.	Popis slika .....	57
15.	Literatura .....	59
	Sažetak .....	60
	Abstract.....	61

## 1. Uvod

Unatrag zadnjih deset godina postignut je veliki napredak u grafičkoj industriji. Sa sve realnijim grafičkim prikazom virtualnih okruženja sve se više traži i realnija fizikalna simulacija. Simulacija krutih tijela naveliko je prisutna u filmovima pogotovo kad je potrebno simulirati veliki broj tijela koje bi bilo previše komplikirano ručno animirati pa je potrebno primijeniti odgovarajuću fizikalnu simulaciju. Kako je računalna snaga danas dovoljno napredovala i u računalnim igrama se već puno koristiti fizikalna simulacija.

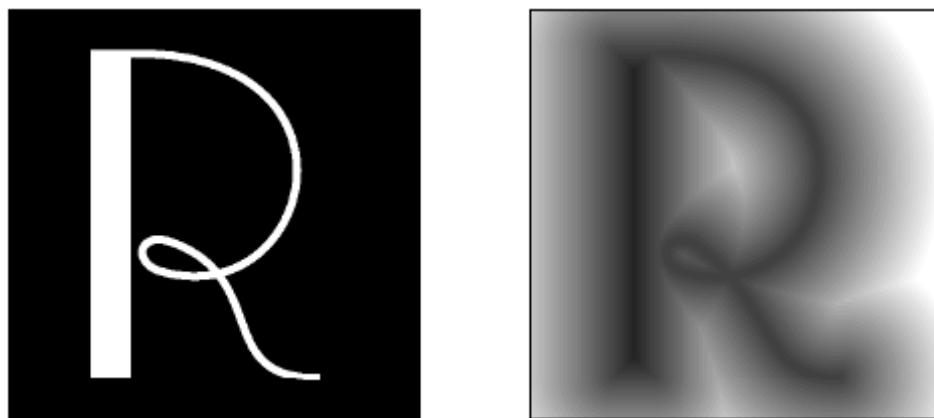
Većina rada je zasnovana na radu "Nonconvex Rigid Bodies with Stacking" [1]. Prvo je opisan način detekcije sudara između konkavnih tijela. Zatim je prikazan postupak simulacije gibanja tijela. Sudari i kontakti između tijela opisani su pomoću impulsa s time da su se pokušali ispraviti neki od nedostataka koji se javljaju kod impulsne metode. Simulirano je statičko i dinamičko trenje, te trenje prilikom kotrljanja i vrtnje. Veliki problem kod simulacije većeg broja tijela je kada ima mnogo tijela naslaganih jedno na drugo. Taj problem je riješen konstruiranjem kontaktnog grafa predloženog u [1]. Na kraju je još prikazano i kako se impulsi također mogu iskoristiti za modeliranje zglobova i njihovo kombiniranje sa sudarima i kontaktima.

## 2. Prikaz tijela

Kruta tijela se na računalu najčešće prikazuju pomoću mreže trokuta. Za svako tijelo definirati će se lista vrhova, trokuta i bridova. Većina programskih alata za modeliranje podržava zapis objekta kao mreže poligona pri čemu se u datoteku zapisuju liste vrhove i trokuta, međutim bez liste bridova. Budući da će za detekciju sudara konkavnih tijela biti potrebno znati i pojedine bridove svakog tijela potrebno je posebno izračunati i bridove. Bridovi se odrede tako da se za svaki trokut u listi traži ima li neki drugi trokut dva vrha koji se podudaraju s vrhovima prvog trokuta. Osim toga mnogi programski alati pri zapisu liste vrhova iste vrhove zapisuju više puta za različite trokute, tako da je često potrebno izbaciti duple vrhove.

### 2.1. Polje udaljenosti

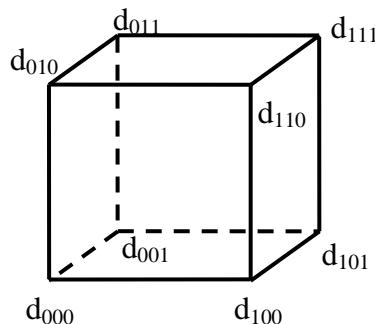
Osim mreže trokuta za svako tijelo definirat će se i polje udaljenosti. Polje udaljenosti za svaku točku u prostoru određuje udaljenost te točke od tijela. Ovisno o kompleksnosti tijela biti će potrebna različita veličina polja. Polje se može spremiti na jednoliku mrežu ili na oktalno stablo, ovisno o tome što je usko grlo u simulaciji, memorija ili procesor. Budući da će se u ovoj simulaciji simulirati uglavnom maleni objekti memorija neće biti usko grlo pa će se koristiti jednolika mreža.



Slika 1. 2D polje udaljenosti za simbol R [2]

Polje udaljenosti definirat će se samo unutar omeđujućeg volumena tijela poravnatog s koordinatnim osima (AABB, axis aligned bounding box). Za svaku

točku mreže unutar AABB potrebno je odrediti udaljenost te točke od površine tijela. Pri tome je potrebno odrediti je li točka unutar tijela ili izvan, za točke unutar tijela udaljenost će se označiti negativnom vrijednošću a za točke izvan tijela udaljenost će imati pozitivnu vrijednost. Određivanje udaljenosti detaljnije je objašnjeno u poglavlju o detekciji sudara. Udaljenost određene točke ( $x$   $y$   $z$ ) od tijela odredit će se trilinearnom interpolacijom 8 najbližih susjednih točaka, pri čemu je koordinate  $x, y, z$  potrebno transformirati u koordinatni sustavu ćelije i normalizirati na vrijednost [0 1].



Slika 2. Interpolacija udaljenosti

Udaljenost je određena sljedećom formulom:

$$\begin{aligned}\phi = & (1 - z) \cdot ((1 - y) \cdot (d_{000} \cdot (1 - x) + d_{100} \cdot x) + y \cdot (d_{010} \cdot (1 - x) + d_{110} \cdot x)) \\ & + z \cdot ((1 - y) \cdot (d_{001} \cdot (1 - x) + d_{101} \cdot x) + y \cdot (d_{011} \cdot (1 - x) + d_{111} \cdot x))\end{aligned}\quad (2.1)$$

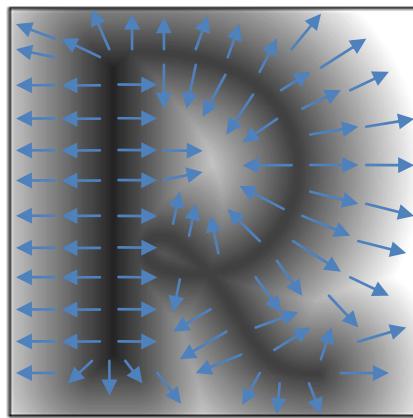
Velika prednost korištenja polja udaljenosti je pri računanju normale. Kako točke unutar tijela imaju negativnu, a točke izvana pozitivnu vrijednost normala se može definirati kao gradijent udaljenosti u traženoj točki:

$$N = \nabla \phi \quad (2.2)$$

Time dobivamo aproksimaciju normale u cijelom prostoru za razliku od normale definirane na samoj površini objekta čime se ubrzava postupak detekcije sudara i određivanja reakcije na sudar.

Gradijent skalarnog polja je vektorsko polje koje pokazuje u kojem smjeru najviše raste skalarno polje i njegova veličina odgovara povećanju u tom smjeru. Gradijent se definira kao:

$$\nabla \phi = \left( \frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y}, \frac{\partial \phi}{\partial z} \right) \quad (2.3)$$



Slika 3. Skalarno polje i njemu odgovarajuće vektorsko polje

Potrebno je izračunati parcijalne derivacije udaljenosti u smjerovima  $x$ ,  $y$  i  $z$ .

Kao i kod određivanja udaljenosti normala će se također odrediti interpolacijom.

Parcijalna derivacija  $\frac{\partial \phi}{\partial x}$  odredi se tako da se prvo izračunaju parcijalne derivacije

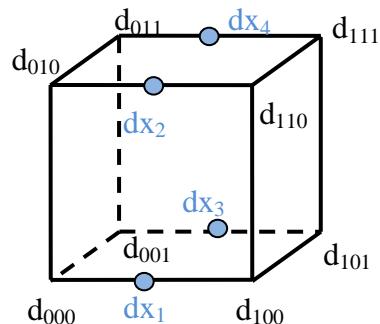
točaka koje se nalaze u sredini bridova paralelnih sa  $x$ .

$$dx_1 = \left( \frac{\partial \phi}{\partial x} \right)_1 = \frac{d_{100} - d_{000}}{cellSize} \quad (2.4)$$

gdje je:

$cellSize$  duljina stranice kocke celije

S time da valja napomenuti da prilikom implementacije nije potrebno dijeliti sa  $cellSize$  jer nas ne zanima točna veličina gradijenta već samo smjer. Ukupna derivacija po  $x$  izračuna se bilinearnom interpolacijom između četiri brida.



Slika 4. Interpolacija normale

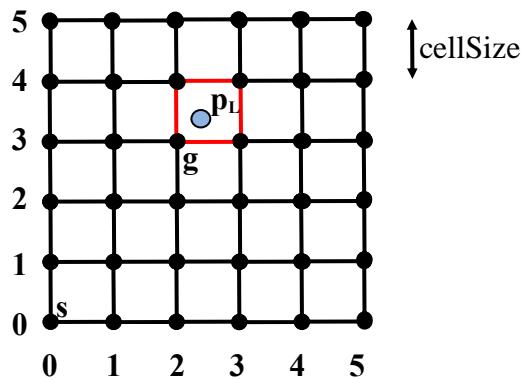
Vrijednost  $x$  normale sudara je:

$$\frac{\partial \phi}{\partial x} = (1-z) \cdot ((1-y) \cdot dx_2 + y \cdot dx_1) + z \cdot ((1-y) \cdot dx_4 + y \cdot dx_3) \quad (2.5)$$

Na sličan način se odrede i preostale parcijalne derivacije:  $\frac{\partial \phi}{\partial x}$  i  $\frac{\partial \phi}{\partial y}$  koje određuju

normalu  $N = \left( \frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y}, \frac{\partial \phi}{\partial z} \right)$ .

Određivanje ćelije u kojoj se nalazi točka te koordinate točke u koordinatnom sustavu ćelije prikazano je na sljedećoj slici:



Slika 5. 2D prikaz određivanja ćelije u kojoj se nalazi točka te njezine koordinate u koordinatnom sustavu ćelije

Formula za određivanje indeksa prve točke ćelije  $g = (g_x, g_y, g_z)$  u kojoj se nalazi točka je:

$$g = \left\lfloor \frac{p_L - s}{cellSize} \right\rfloor \quad (2.6)$$

gdje su:

$p_L$  koordinate točke u lokalnom sustavu tijela

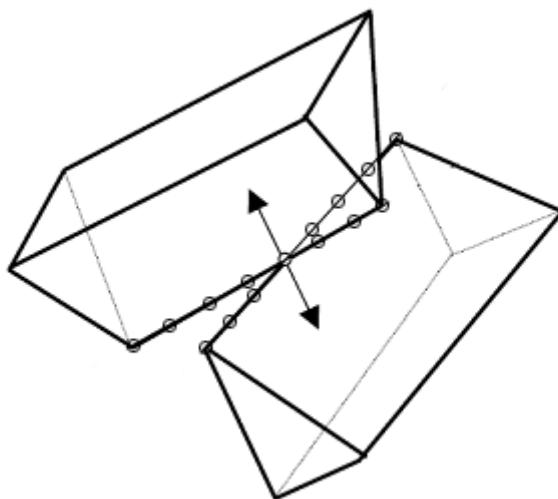
$s$  koordinate prve točke mreže u lokalnom sustavu tijela

Koordinate točke u lokalnom koordinatnom sustavu ćelije  $p_G$  određe se prema sljedećoj formuli:

$$p_G = \frac{(p_L - s)}{cellSize} - g \quad (2.7)$$

### 3. Detekcija sudara

Testiranje je li došlo do sudara između dva tijela određuje se na način da se za svaki vrh testira je li unutar ili izvan tijela. Na taj način se neće detektirati svi sudari poput sudara između dva brida kada se vrhovi brida nalaze izvan tijela s kojim se traži sudar. Kada se tijelo sastoji od velikog broja trokuta greška zbog zanemarenih sudara između bridova biti će mala, no ukoliko se traži sudar za tijelo s malim brojem trokuta greška će biti prevelika da bi zanemarili sudare između bridova. Na primjer, za detekciju sudara između dvije kocke koje imaju samo 12 trokuta greška je prevelika pa je potrebno detektirati sudare između bridova. Sudari između bridova se detektiraju na način da se na svaki brid doda određeni broj točaka, ovisno o tome koliko preciznu simulaciju želimo (za sudar između dvije kocke iste veličine dovoljno je dodati dva do tri vrha). Za svaku točku na bridu vrši se detekcija sudara i od svih tih točaka za točku sudara se uzima samo ona točka koja se nalazi najdublje unutar drugog tijela.



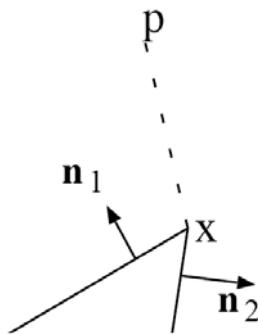
Slika 6. Sudar između dva brida

#### 3.1. Određivanje predznaka

Nalazi li se točka unutar konkavnog tijela ili izvan njega može se odrediti na mnogo načina. Jedan od jednostavnijih načina je bacanjem zrake iz točke i određivanja sjecišta zrake s mrežom poligona. Ukoliko je zraka presjekla mrežu neparan broj puta točka je unutar tijela. Problem kod bacanja zrake je kada zraka

siječe više trokuta u istoj točci (kada siječe bridove ili vrhove) pa je potrebno ta sjecišta brojati kao jedno. Zbog toga će se u ovom radu za određivanje predznaka koristiti metoda predstavljena u [3].

Prvo je potrebno odrediti točku na površini tijela  $x$  koja je najbliža točki  $p$  za koju želimo odrediti predznak. Normala  $n$  u točki  $x$  jednaka je normali poligona kojem pripada (osim za točke na bridovima i vrhovima). Sada je jednostavno za odrediti je li točka unutar ili izvan tijela a to je skalarnim produktom normale  $n$  i vektora  $r = p - x$ . Ukoliko je produkt pozitivan točka je izvan tijela a inače je u tijelu. Međutim za bridove i vrhove normala nije definirana te se lako može pokazati da postoje slučajevi kod kojih je skalarni produkt s nekim normalama poligona koji se sijeku u vrhu pozitivan dok je s drugim negativan. Primjer takvog slučaja vidi se na slici 7.



*Slika 7. Skalarni umnožak  $r = p - x$  s jednom normalom daje pozitivan rezultat a s drugom negativan.*

Potrebno je odrediti normalu kod koje će skalarni produkt dati ispravan rezultat. Takva normala nazvat će se *pseudo normala*. Pseudo normala bi se mogla izračunati kao srednja vrijednost svih normala poligona u vrhu, kao kod računanja normale prilikom Gouraud-ovog sjenčanja, no i tada se mogu pronaći slučajevi kod kojih skalarni produkt ne daje ispravan rezultat. Ispravna normala odredit će se zbrojem svih normala poligona pri čemu će težina svake normale ovisiti o veličini kuta pod kojim određeni poligon sudjeluje u vrhu. Tako za normalu na bridu između dva poligona  $i$  i  $j$  vrijedi:

$$n_\alpha = \pi_i \cdot n_i + \pi_j \cdot n_j \quad (3.1)$$

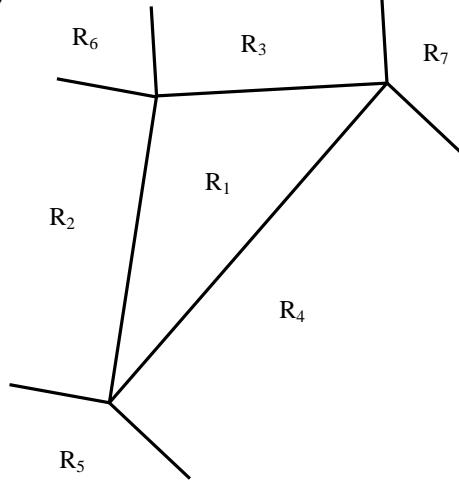
Dok za normalu u vrhu vrijedi:

$$n_\alpha = \sum_i \alpha_i \cdot n_i \quad (3.2)$$

Za trokut određen vrhovima  $P_1$ ,  $P_2$ ,  $P_3$  koji vrhom  $P_1$  sudjeluje u određivanju normale težina  $\alpha_i$  normale  $n_i$  jednaka je kutu između vektora  $P_1P_2$  i  $P_1P_3$ .

$$\cos \alpha_i = \frac{P_1P_2 \cdot P_1P_3}{|P_1P_2||P_1P_3|} \quad (3.3)$$

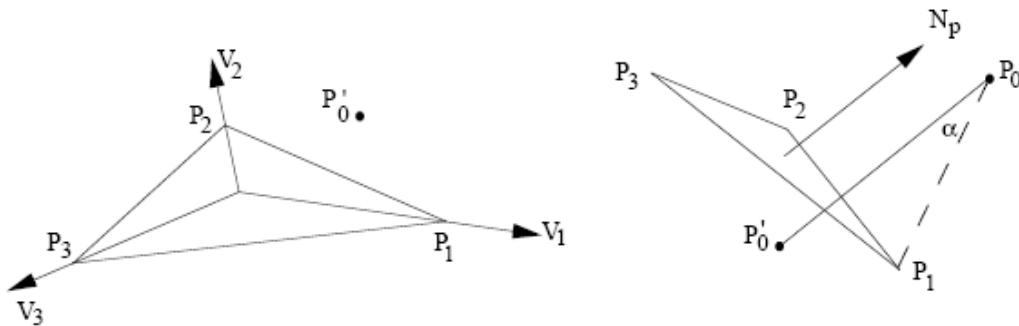
Prilikom određivanja udaljenosti točke od tijela potrebno je odrediti udaljenost točke od svakog trokuta, udaljenost od tijela jednak je najmanjoj od svih tih udaljenosti. Osim udaljenosti potrebno je odrediti i točku na trokutu koja je najbliža za traženu točku te normalu. Da bi se to odredilo potrebno je pronaći regiju u koju se projicira tražena točka a to će se sve odrediti prilikom određivanja udaljenosti točke od trokuta (Slika 8.).



Slika 8. Sedam različitih Voronoi-evih regija ravnine u kojoj se nalazi trokut

## 3.2. Udaljenost točke od trokuta

Udaljenost točke od trokuta nije jednostavno za izračunati kao što bi se moglo pomisliti na prvi pogled. To je zbog toga što postoji mnogo različitih slučajeva. Točka može biti najbliža ravnini trokuta, najbliža bridu trokuta ili najbliža vrhu trokuta. Metoda određivanja udaljenosti koja se koristi u ovom radu dobro je opisana u [4].



Slika 9. Određivanje udaljenosti točke od trokuta

Normala  $N_p$  od trokuta  $P_1P_2P_3$  se može izračunati kao:

$$N_p = P_1P_2 \times P_1P_3 \quad (3.4)$$

Kut između normale  $N_p$  i  $P_1P_0$  je:

$$\cos \alpha = \frac{P_1P_0 \cdot N_p}{|P_1P_0| |N_p|} \quad (3.5)$$

Udaljenost točke  $P_0$  do njene projekcije na ravnini trokuta  $P_0'$  je:

$$|P_0P_0'| = |P_0P_1| \cos \alpha \quad (3.6)$$

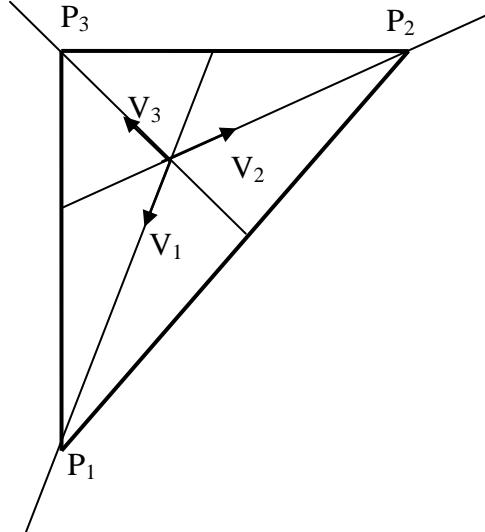
Vektor  $P_0P_0'$  iznosi:

$$P_0P_0' = -|P_0P_0'| \frac{N_p}{|N_p|} \quad (3.7)$$

Sada se može odrediti projicirana točka  $P_0'$ :

$$P_0' = P_0 + P_0P_0' \quad (3.8)$$

Da bi se odredilo je li projicirana točka unutra ili izvan trokuta i ako je izvan kojem je bridu ili vrhu najbliža potrebno je odrediti položaj točke u odnosu na tri vektora  $V_1, V_2, V_3$  koji predstavljaju simetrale kutova:



Slika 10. Određivanje položaja točke u odnosu na simetrale kutova

$$V_1 = \frac{P_2 P_1}{|P_2 P_1|} + \frac{P_3 P_1}{|P_3 P_1|} \quad V_2 = \frac{P_3 P_2}{|P_3 P_2|} + \frac{P_1 P_2}{|P_1 P_2|} \quad V_3 = \frac{P_1 P_3}{|P_1 P_3|} + \frac{P_2 P_3}{|P_2 P_3|} \quad (3.9)$$

Ako je  $f_1 = (V_1 \times P_1 P_0) \cdot N_p$ , i  $f_1 > 0$  kažemo da se  $P_0'$  nalazi iznad vektora  $V_1$  (ili da se  $P_0'$  nalazi suprotno od smjera kazaljke na satu u odnosu na  $V_1$ ). Na isti način se mogu i izračunati  $f_2$  i  $f_3$ . Koristeći  $f_1$ ,  $f_2$  i  $f_3$  može se odrediti položaj točke  $P_0'$  u odnosu na  $V_1$ ,  $V_2$  i  $V_3$ . Ako se  $P_0'$  nalazi ispod  $V_2$  i iznad  $V_1$  točka je unutar trokuta ako je:

$$(P_0' P_1 \times P_0' P_2) \cdot N_p \geq 0 \quad (3.10)$$

Slijedi da je udaljenost točke od trokuta jednaka  $|P_0 P_0'|$ .

Ukoliko je izraz (3.10) neistinit točka je izvan trokuta i potrebno je odrediti je li najbliža bridu ili vrhu.

Smjer od  $P_0'$  (projekcija  $P_0$  na ravninu) prema  $P_0''$  (projekcija  $P_0'$  na brid) jednak je:

$$R = (P_0' P_2 \times P_0' P_1) \times P_1 P_2 \quad (3.11)$$

Kut između vektora  $R$  i  $P_0' P_1$  je:

$$\cos \beta = \frac{P_0' P_1 \cdot N_p}{|P_0' P_1| \|N_p\|} \quad (3.12)$$

Udaljenost točke  $P_0'$  do njene projekcije na bridu je:

$$|P_0'P_0''| = |P_0'P_1| \cos \beta \quad (3.13)$$

Vektor  $P_0'P_0''$  iznosi:

$$P_0'P_0'' = |P_0'P_0''| \frac{R}{|R|} \quad (3.14)$$

Sada se može odrediti projicirana točka na bridu  $P_0''$

$$P_0'' = P_0' + P_0'P_0'' \quad (3.15)$$

Neka je:

$$t = \pm \frac{P_0'' - P_1}{P_2 - P_1} \quad (3.16)$$

Gdje je predznak negativan ukoliko je  $P_0''P_1 < 0$ .

Ako je  $0 \leq t \leq 1$ ,  $P_0''$  je između  $P_1$  i  $P_2$ , znači da je točka  $P_0''$  najbliža bridu  $P_1P_2$ .

Udaljenost  $P_0'$  od brida  $P_1P_2$  je  $|P_0'P_0''|$  izračunata u (3.13) a udaljenost  $P_0$  od brida  $P_1P_2$  je  $\sqrt{|P_0'P_0''|^2 + |P_0P_0'|^2}$ .

Ako je  $t < 0$ ,  $P_0''$  je najbliža točki  $P_1$ , i udaljenost  $P_0$  od trokuta je  $|P_1 - P_0|$ .

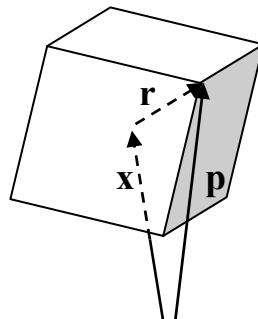
Ako je  $t > 1$ ,  $P_0''$  je najbliža točki  $P_2$ , i udaljenost  $P_0$  od trokuta je  $|P_2 - P_0|$ .

Na sličan način se izračuna udaljenost točke od trokuta za slučaj kada se  $P_0'$  nalazi ispod  $V_3$  i iznad  $V_2$ , te slučaj kada se  $P_0'$  nalazi ispod  $V_1$  i iznad  $V_3$ .

## 4. Dinamika krutog tijela

Simulacija gibanja krutog tijela slična je simulaciji gibanja čestica. Kod simulacije čestica stanje čestice određeno je pozicijom  $x$  i brzinom  $v$ . Promjena stanja kroz vrijeme određena je promjenom pozicije  $\dot{x} = v$  te promjenom brzine  $\dot{v} = \frac{F}{m}$ . Novo stanje odrediti će se pomoću jednog od mnogih postupaka integriranja. Na sličan način će se simulirati i kruta tijela osim što će stanje tijela sadržavati više podataka i promjenu stanja će biti nešto teže za odrediti.

### 4.1. Pozicija i orijentacija



Slika 11. Pozicija i orijentacija tijela

Pozicija čestice u prostoru u vremenu  $t$  određena je vektorom  $x$ . Kako kruto tijelo zauzima u prostoru određeni volumen njegov prikaz je nešto složeniji. Za poziciju tijela uzet će se jedna fiksna točka u prostoru, pri tome se za tu točku odredi da se u koordinatnom sustavu tijela nalazi na poziciji  $(0,0,0)$ . Kako bi se pojednostavile mnoge operacije za tu točku se uzima centar mase tijela. Osim pozicije tijela, za razliku od čestica kruta tijela se mogu i rotirati tako da je potrebno odrediti njihovu orijentaciju. Orijentacija tijela se može predstaviti  $3 \times 3$  matricom rotacije. Tako da ukoliko je  $r$  pozicija točke u koordinatnom sustavu tijela pozicija točke u globalnim koordinatama će biti rezultat rotiranja točke oko ishodišta i zatim translatiranja točke:

$$p = x + R \cdot r \quad (4.1)$$

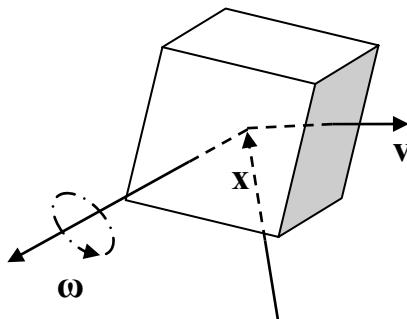
gdje su:

$R$  3x3 matrica rotacije

$x$  pozicija tijela

Prilikom iscrtavanja tijela koristit će se formula (4.1) za određivanje nove pozicije svakog vrha tijela. Dakle kod određivanja nove pozicije i orijentacije tijela prilikom simulacije nije potrebno računati i novu poziciju svakog vrha tijela već će se to raditi samo prilikom iscrtavanja budući da postoji grafička podrška za brzo računanje s matricama.

## 4.2. Linearna i kutna brzina



Slika 12. Linearna i kutna brzina tijela

Sljedeće što je potrebno odrediti je promjena pozicije i orijentacije tijekom vremena, dakle potrebno je odrediti  $\dot{x}$  i  $\dot{R}$ . Promjena pozicije je jednostavna, budući da je  $x$  pozicija centra mase u globalnim koordinatama  $\dot{x}$  biti će jednako definiranoj linearnej brzini tijela  $v$ .

$$\dot{x} = v \quad (4.2)$$

Kutna brzina može se zapisati pomoću vektora  $\omega$ . Smjer  $\omega$  određuje os oko koje se tijelo vrti. Duljina vektora  $|\omega|$  određuje koliko brzo se tijelo vrti (kutna brzina). Za linearnu brzinu odnos između  $x$  i  $v$  jednak je  $v = \dot{x}$ . Postavlja se pitanje koji je odnos između  $R$  i  $\omega$ . Kako stupci od  $R$  predstavljaju smjerove koordinatnih osi lokalnog sustava tijela, znači da stupci od  $\dot{R}$  opisuju brzinu promjene smjera koordinatnih osi. Promjena smjera vektora određena je sa:

$$\dot{r} = \omega \times r \quad (4.3)$$

Znamo da je smjer  $x$  osi lokalnog koordinatnog sustava tijela zapisan u prvom stupcu matrice  $R$ .

$$\begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix}$$

Na isti način se odrede i ostala dva stupca, te se dobije odnos između  $R$  i  $\omega$ .

$$\dot{R} = \left[ \omega \times \begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix}, \omega \times \begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix}, \omega \times \begin{bmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{bmatrix} \right] \quad (4.4)$$

Kako bi se izračun  $\dot{R}$  donekle pojednostavio vektorski produkt ćemo definirati na drugi način. Vektorski produkt  $a \times b$  jednak je:

$$a \times b = \begin{bmatrix} a_y b_z - b_y a_z \\ -a_x b_z + b_x a_z \\ a_x b_y - b_x a_y \end{bmatrix} \quad (4.5)$$

Za određeni vektor  $a$  definira se matrica vektorskog produkta kao:

$$\tilde{a} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \quad (4.6)$$

Sada se  $\dot{R}$  može zapisati kao:

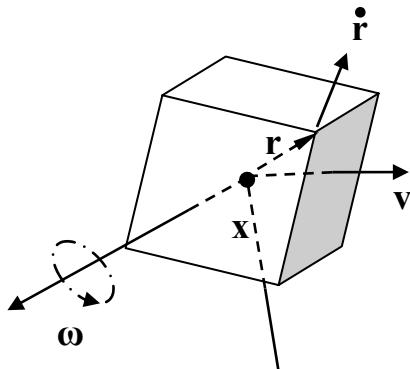
$$\dot{R} = \left[ \tilde{\omega} \cdot \begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix}, \tilde{\omega} \cdot \begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix}, \tilde{\omega} \cdot \begin{bmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{bmatrix} \right] \quad (4.7)$$

Prema pravilima množenja matrica dobijemo da je promjena orijentacije jednaka:

$$\dot{R} = \tilde{\omega} \cdot R \quad (4.8)$$

Matrica vektorskog produkta će se osim za računanje promjene orijentacije dosta često koristiti kod računanja reakcije na sudar.

### 4.3. Brzina točke na tijelu



Slika 13. Brzina točke na tijelu

Prilikom rješavanja sudara između tijela često će biti potrebno znati brzinu određene točke na tijelu. Derivacijom izraza (4.1) i uvrštavanjem izraza (4.8) te malo raspisivanja dobije se brzina točke na tijelu:

$$\dot{p} = v + \omega \times r \quad (4.9)$$

### 4.4. Orijentacija pomoću kvaterniona

Osim  $3 \times 3$  matrica rotacije orijentacija tijela može se predstaviti i kvaternionima. Kvaternioni su bolji za prikaz orijentacije zbog mnogih razloga poput bolje interpolacije između dviju orijentacija. No za simulaciju krutih tijela glavni razlog za korištenje kvaterniona je zbog greške zaokruživanja koja se akumulira tijekom rotacije. Konstantnim ažuriranjem orijentacije pomoću formule (4.8) numerička greška će se nakupiti unutar koeficijenata matrice tako da matrica rotacije neće više predstavljati isključivo matricu rotacije već će sadržavati i efekte smika i skaliranja. Greška se može ispraviti ortonormiranjem matrice prilikom svakog ažuriranja no bolji način je korištenje kvaterniona. Kako kvaternioni koriste četiri parametra za prikaz tri stupnja slobode dok matrica koristi devet parametra očito je da će greška biti mnogo manja. Greška kod kvaterniona se može zanemariti, no ukoliko je potrebno greška se može ispraviti normaliziranjem kvaterniona na jediničnu duljinu. Za integriranje orijentacije će se koristiti kvaternioni dok će matrica rotacije još uvijek biti potrebna za određene izračune poput računanja momenta inercije tijela u globalnim koordinatama, također kutna brzina  $\omega$  će još

uvijek biti prikazana pomoću vektora. Za derivaciju kvaterniona  $q = (s, \mathbf{v})$  se može pokazati da vrijedi [5]:

$$\dot{q} = \frac{1}{2} \omega q \quad (4.10)$$

gdje je:

$\omega$  skraćeni zapis za kvaternion  $\omega = (0, \text{kutnaBrzina})$

Za pretvorbu iz kvaterniona  $q = (s, \mathbf{v})$  u matricu rotacije  $R$  vrijedi [6]:

$$R = \begin{bmatrix} 1 - 2v_y^2 - 2v_z^2 & 2v_xv_y - 2sv_z & 2v_xv_z + 2sv_y \\ 2v_xv_y + 2sv_z & 1 - 2v_x^2 - 2v_z^2 & 2v_yv_z + 2sv_x \\ 2v_xv_z - 2sv_y & 2v_yv_z + 2sv_x & 1 - 2v_x^2 - 2v_y^2 \end{bmatrix} \quad (4.11)$$

## 4.5. Masa tijela i centar mase

Za realnu simulaciju krutih tijela potrebno je znati masu tijela i centar mase. Masa tijela se jednostavno može ručno zadati za svako tijelo, također centar mase se može intuitivno odrediti za jednostavna tijela poput kugle ili kocke, međutim za kompleksna konkavna tijela centar će biti potrebno izračunati. Iako postoje metode koje računaju svojstva tijela direktno iz mreže trokuta poput [7] kako već imamo izračunato polje udaljenosti svojstva tijela se mogu izračunati na jednostavniji način slično kao u [8]. Masa tijela  $M$  i centar mase  $x_{cm}$  mogu se izračunati pomoću zadane gustoće tijela  $\delta$ . Za ukupnu masu tijela vrijedi:

$$M = \sum_{i=1}^N m_i \quad (4.12)$$

gdje je:

$m_i$  masa pojedine čestice tijela

Kako za tijelo imamo izračunato polje udaljenosti masa tijela će se izračunati zbrajanjem masa pojedinih ćelija unutar polja udaljenosti, s time da se zbrajam samo one ćelije koje se nalaze unutar tijela. Vrijedi:

$$m_i = \delta \cdot d^3 \quad (4.13)$$

gdje je:

$d$  duljina stranice čelije

Centar mase može izračunati prema sljedećoj formuli:

$$x_{cm} = \frac{\sum_{i=1}^N m_i r_i}{M} \quad (4.14)$$

gdje je:

$r_i$  udaljenost centra pojedine čelije od ishodišta koordinatnog sustava

## 4.6. Moment inercije

Kao što masa tijela utječe na translaciju tijela moment inercije (moment tromosti) utječe na rotaciju tijela. Da bi ispravno odredili kutnu brzinu tijela prilikom određivanja reakcije na sudar potrebno je odrediti moment inercije tijela. Kao i centar mase moment inercije se ne može intuitivno odrediti te ga je potrebno izračunati. Za inerciju materijalne točke vrijedi:

$$I_i = m_i \begin{bmatrix} r_{iy}^2 + r_{iz}^2 & -r_{ix}r_{iy} & -r_{ix}r_{iz} \\ -r_{iy}r_{ix} & r_{ix}^2 + r_{iz}^2 & -r_{iy}r_{iz} \\ -r_{iz}r_{ix} & -r_{iz}r_{iy} & r_{ix}^2 + r_{iy}^2 \end{bmatrix} \quad (4.15)$$

gdje je:

$r_i$  udaljenost točke od centra rotacije

Na sličan način kao što se masa tijela računa zbrajanjem mase čelija koje se nalaze unutar tijela tako će se i moment inercije izračunati zbrajanjem momenta inercije pojedinih čelija. Čelije su oblika kocke pa je potrebno odrediti moment inercije kocke. Za moment inercije kocke vrijedi:

$$I_{kocka} = \frac{2}{3} m \left( \frac{d}{2} \right)^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.16)$$

gdje su:

$m$  masa kocke

$d$  duljina stranice kocke

Može se pokazati da je moment inercije tijela oko zadane točke jednak zbroju momenta inercije oko težišta i momenta inercije čestice oko zadane točke. Slijedi da je moment inercije tijela  $I$  jednak:

$$I = \sum_{i=1}^N (I_{kocka} + I_i) \quad (4.17)$$

Na prvi pogled pomislilo bi se da je potrebno računati prethodnu sumu svaki put kod promjene rotacije objekta. To bi bilo računski prezahtjevno tako da će se moment inercije izračunati samo jedanput na početku programa dok tijelo još nije rotirano a ispravan moment inercije će se prilikom svake promjene orientacije računati prema sljedećoj formuli [5]:

$$I = R \cdot I_{init} \cdot R^T \quad (4.18)$$

gdje su:

$I_{init}$  početni moment inercije tijela

$R$  3x3 matrica orientacije

U programu će se zbog brzine spremati inverzija momenta inercije pa se u tom slučaju moment inercije može izračunati na sljedeći način [5]:

$$I^{-1} = R \cdot I_{init}^{-1} \cdot R^T \quad (4.19)$$

## 5. Korak integracije

Za svaki vremenski korak potrebno je izračunati novo stanje objekta na temelju proteklog vremena. Jednadžbe gibanja krutog tijela su sljedeće:

$$\dot{x} = v \quad \dot{q} = \frac{1}{2} \omega q \quad (5.1)$$

$$\dot{v} = \frac{F}{m} \quad \dot{\omega} = I^{-1} \tau \quad (5.2)$$

gdje je:

$\tau$  ukupni moment sile koji djeluje na tijelo

Postoji mnogo različitih preciznih postupaka numeričke integracije. No kako se u ovom algoritmu neće tražiti točno vrijeme sudara sasvim dovoljan postupak će biti i jednostavan Eulerov postupak. Tako da za nove pozicije objekata vrijedi:

$$v_{n+1} = v_n + \dot{v}_n \Delta t \quad \omega_{n+1} = \omega_n + \dot{\omega}_n \Delta t \quad (5.3)$$

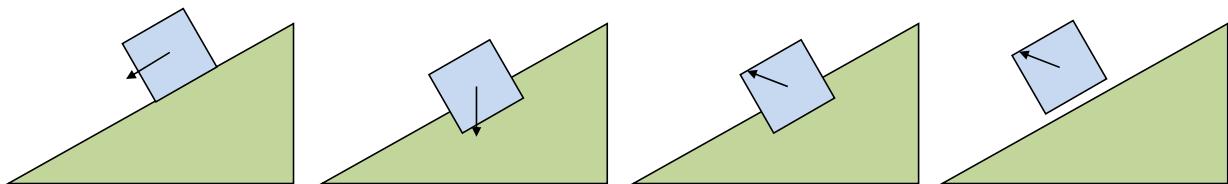
$$x_{n+1} = x_n + v_{n+1} \Delta t \quad q_{n+1} = q_n + \frac{1}{2} \omega q_n \Delta t \quad (5.4)$$

Dva su osnovna pristupa rješavanja sudara i kontakata, pomoću impulsa i pomoću ograničenja. Postupak rješavanja sudara pomoću impulsa radi na način da se odredi sudar između dva tijela te se obično pronađe par najbližih točaka i na njih se primjeni impuls. Zbog primjenjenog impulsa mogu se dogoditi novi sudari pa je potrebno ponovo primjenjivati nove impulse sve dok se svi sudari ne riješe. Izračun impulsa je vrlo brz pa je metoda vrlo efikasna. Problem nastaje kada se tijela naslanjaju jedno na drugo te tada međusobno imaju puno kontakata istovremeno, međutim samo će jedan kontakt biti riješen u jednom vremenskom koraku. Rezultat toga će biti vibriranje tijela. Ukoliko bi se htjela velika preciznost svi kontakti bi se trebali riješiti u isto vrijeme. To se može riješiti pomoću ograničenja. Za svaki kontakt koji se dogodi definira se određeno ograničenje penetracije, određeni uvjet koji mora biti zadovoljen. Sva ta ograničenja se mogu zapisati pomoću skupa linearnih jednadžbi. Rješavanje tih jednadžbi može biti jako složeno pogotovo kada ima mnogo tijela koja su istovremeno u kontaktu. Pa je tako u većini slučajeva osim kad je potrebna velika preciznost bolje rješavati kontakte pomoću impulsa.

Sudar se definira kao dodir između dva tijela pri čemu tijela međusobno imaju određenu relativnu brzinu, dok je kontakt dodir između dva tijela pri čemu tijela nemaju brzinu već djeluju jedno na drugo samo određenom silom. U [1] je predložen novi algoritam kojim se nastoje smanjiti mnogi problemi vezani uz simulaciju pomoću impulsa. Uobičajeni postupak simulacije krutih tijela je sljedeći:

- odredi nove pozicije i brzine tijela
- riješi sudare
- riješi kontakte

Pri tome se za rješavanje sudara i rješavanja kontakta koriste različiti algoritmi. Pri rješavanju kontakta bi se primjerice izračunao impuls kao i za sudar te bi se na novu brzinu određenu pomoću impulsa nadodala još mala brzina ovisno o tome koliko je objekt penetrirao unutar drugi objekt kako bi ih se razdvojilo. Na mjesto kontakta se može postaviti i opruga tako da ovisno o dubini u kojoj se neko tijelo nalazi unutar drugog toliko na njega djeluje veća sila. Problem koji se tada javlja najbolje se vidi prilikom simulacije kocke koje klizi niz kosinu. Neka je koeficijent elastičnosti  $\varepsilon = 1$  i koeficijent trenja dovoljno velik da bi tijelo trebalo stajati mirno na kosini. Pod utjecajem sile teže kocka dobije brzinu i počne propadati kroz kosinu. Prilikom detekcije sudara budući da sada postoji mala brzina između tijela i kosine detektira se sudar te će zbog  $\varepsilon = 1$  kocka promijeniti smjer gibanja tj. odbiti će se od kosine. Nakon toga slijedi detekcija kontakata gdje se neće ništa detektirati budući da se kocka udaljava od kosine. Kocka će kasnije ponovo pasti na kosinu i nastaviti će se nepravilno odbijati od kosine umjesto da klizi niz kosinu. Zbog istog razloga se javlja i efekt da tijelo koje stoji na podlozi nepravilno vibrira umjesto da mirno stoji.



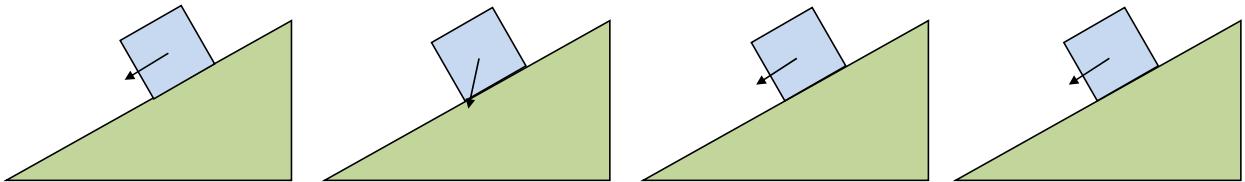
Slika 14. Gibanje kocke na kosini pri standardnom koraku integracije

Postoji mnogo metoda rješavanja tog problema. Jedan od jednostavnijih načina kojim se može riješiti taj problem je da se za određivanje je su li tijela u sudaru ili u kontaktu zada određeni prag pa ukoliko je relativna brzina tijela manja od određenog praga tijela su u kontaktu dok su inače u sudaru.

Prednost predloženog algoritma je razdvajanje rješavanja sudara i kontakata bez upotrebe praga. Sudari su vezani uz brzinu jer tijela prilikom sudara mijenjaju brzinu dok su kontakti vezani uz silu i akceleraciju jer da nema akceleracije kontakte ne bi trebalo uopće rješavati. Zato ima smisla rješavati sudare prije nego što se na tijela primjeni sila, tj. akceleracija, a kontakte nakon primjene sile. Upravo je to predloženo u [1]. Algoritam je sljedeći:

- detekcija i rješavanje sudara
- određivanje novih brzina
- detekcija i rješavanje kontakata
- određivanje novih pozicija

Kako se prvo vrši detekcija sudara sva tijela u mirovanju imaju brzinu jednaku nuli ili blizu nule tako da se prilikom detekcije sudara ukoliko je  $\varepsilon = 1$  ne dobije više elastični sudar. Nakon detekcije sudara odrede se nove brzine tijela pod utjecajem gravitacije. Sada slijedi rješavanje kontakta kod kojeg se ispravno zaustavi tijelo tako da se kod određivanja nove pozicije tijelo ostaje na svome mjestu.



Slika 15. Gibanje kocke na kosini pri novom koraku integracije

Prilikom detekcije sudara i kontakata mora se voditi računa o tome da se isti algoritam koristi za detekciju sudara kao i za detekciju kontakata, inače bi se moglo dogoditi da se ne detektira sudar ali da se detektira kontakt pri čemu će se primiti neispravan impuls, neelastični umjesto elastičnog.

## 6. Rješavanje sudara

Kako se u simulaciji na računalu koristi diskretno vrijeme kod simulacije velikog broja tijela doći će do istovremenog sudara više tijela. Kada se istovremeno sudari veliki broj tijela vrlo je teško riješiti sudare. Ispravno bi bilo odrediti točno vrijeme sudara između svakog tijela te riješiti sve sudare prema redoslijedu kojim su se dogodili. No to bi zahtjevalo previše procesorskog vremena. Umjesto vraćanja simulacije u vrijeme sudara svi sudari se mogu riješiti istovremeno. Na taj način se neće dobiti fizikalno ispravno rješenje (tj. isto rješenje koje bi se dobilo da su se sudari rješavali po redoslijedu kojem su se dogodili), no ionako naš fizikalni model krutog tijela nije savršen tako da ni ne možemo dobiti skroz ispravno fizikalno rješenje. Greška koja će se dobiti zbog istovremenog rješavanja sudara može se pripisati nesavršenosti tijela, tj. mikro-strukturi materijala. Dobiti će se jedno od mnogih fizički uvjerljivih rješenja, a što je i bio cilj.

Sudari se detektiraju tako da se tijela privremeno pomaknu na mjesto sljedećeg vremenskog koraka, te se provjerava koja se tijela međusobno preklapaju. Na isti način se detektiraju i kontakti između tijela, tako da ukoliko nema sudara tijela se trebaju pomaknuti na isto mjesto na koje su se pomakli tijekom detekcije sudara. To će se postići tako da se nove brzine (brzine koje dobijemo u sljedećem vremenskom koraku zbog utjecaja sile) koriste pri određivanju novih pozicija tijela i kod detekcije sudara i kod detekcije kontakta. Dok će se stare brzine koristiti za rješavanje sudara a nove brzine za rješavanje kontakata.

Neka su trenutna pozicija i brzina objekta  $x$  i  $v$ . Nova pozicija objekta koja će se koristiti za detekciju sudara je:

$$x' = x + \Delta t(v + \Delta t g) \quad (6.1)$$

Impuls za rješavanje sudara primijeniti će se na brzinu  $v$ .

Nova pozicija objekta koja će se koristiti za detekciju kontakta je:

$$x' = x + \Delta t v' \quad (6.2)$$

Impuls za rješavanje kontakta primijeniti će se na brzinu  $v' = v + \Delta t g$ .

Pri tome se  $v'$  određuje nakon rješavanja sudara tako da se vidi da ukoliko nema sudara nove pozicije će biti jednake kao i kod detekcije sudara.

Nakon rješavanja svih sudara tijela imaju nove brzine tako da se mogu pojaviti novi sudari između tijela koja prethodno nisu bila u sudaru, tako da se cijeli postupak ponovi nekoliko puta. Pri tome se redoslijed po kojem se detektiraju sudari između tijela svaki put promijeni na način da se dva nasumično odabranim tijelima u listi zamijene mjesta.

Za svaki pronađeni par tijela u sudaru odrede se svi vrhovi tijela koje se nalaze unutar drugog tijela te po jedna točka sa svakog ruba koja se nalazi najdublje unutar drugog tijela. Od svih tih vrhova te točaka sa rubova uzme se točka koja se nalazi najdublje unutar nekog od tijela. Pomoću nje se odredi reakcija na sudar, tj. nova brzina tijela. Pomoću te nove brzine ponovo se odredi pozicija objekta. Sada je moguće da više nema sudara između tijela, ukoliko još uvijek ima sudara postupak se ponavlja sve dok se sve točke jednog tijela više ne nalaze unutar drugog tijela ili se udaljavaju od njega. Kako bi se ubrzalo rješavanje sudara postupak se neće ponavljati sve dok se sve točke razdvajaju nego dok se sve točke barem jedanput ne razdvajaju.

## 7. Reakcija na sudar

Sve sile koje djeluju na kruto tijelo mogu se obuhvatiti jednom silom  $F$  koja djeluje na centar mase tijela i jednog momenta sile  $M$ . Neka je  $r$  pozicija centra mase i  $m$  masa tijela. Prema osnovnim fizikalnim zakonima vrijedi [9]:

$$F = ma_1 \quad (7.1)$$

$$M = r_1 \times F = I_1 \alpha_1 + \omega_1 \times I_1 \omega_1 \quad (7.2)$$

gdje su:

$a_1$  linearna akceleracija tijela

$\omega_1$  kutna akceleracija tijela

Izraz  $\omega_1 \times I_1 \omega_1$  iz jednadžbe (7.2) sadrži sile koje djeluju na tijelo pod utjecajem momenta inercije koje su zanemarive u trenutku sudara. Uzimajući u obzir da su  $m_1$ ,  $I_1$  i  $r_1$  konstante gornje jednadžbe se mogu integrirati tako da se dobije:

$$j = m_1 \Delta v_1 \quad (7.3)$$

$$r_1 \times j_1 = I_1 \Delta \omega_1 \quad (7.4)$$

gdje je:

$$j \text{ impuls sudara} \quad j(t) = \int_0^t f(\tau) d\tau$$

Sada se mogu napisati promjene brzine i kutne brzine pod utjecajem impulsa:

$$\Delta v_1 = \frac{1}{m} j \quad (7.5)$$

$$\Delta \omega_1 = I_1^{-1} r_1 \times j \quad (7.6)$$

Brzina točke na tijelu je:

$$u_1 = v_1 + \omega_1 \times r_1 \quad (7.7)$$

Kombiniranjem jednadžbi (7.5), (7.6) i (7.7) dobije se promjena brzine točke na tijelu:

$$\Delta u_1 = \frac{1}{m_1} j + (I^{-1} r_1 \times j) \times r_1 = \left( \frac{1}{m_1} 1 - \tilde{r}_1 I_1^{-1} \tilde{r}_1 \right) j \quad (7.8)$$

gdje je:

$\tilde{r}$  matrica vektorskog produkta od  $r$

$1$  jedinična matrica veličine  $3 \times 3$

Na isti način se može dobiti promjena brzine točke na drugom tijelu, jedina razlika je da se  $j$  zamijeni s  $-j$ .

$$\Delta u_2 = - \left( \frac{1}{m_2} 1 - \tilde{r}_2 I_2^{-1} \tilde{r}_2 \right) j \quad (7.9)$$

Sada se mogu napisati brzine točaka nakon sudara:

$$u'_1 = u_1 + K_1 j \quad (7.10)$$

$$u'_2 = u_2 - K_2 j \quad (7.11)$$

gdje je:

$K = \left( \frac{1}{m} 1 - \tilde{r} I^{-1} \tilde{r} \right)$   $3 \times 3$  matrica i naziva se matrica sudara.

Iz (7.10) i (7.11) odredi se relativna brzina točaka nakon sudara:

$$u'_{rel} = u_{rel} + K_T j \quad (7.12)$$

gdje je:

$K_T$  zbroj  $K_1$  i  $K_2$

Množenjem gornje jednadžbe s normalom dobije se:

$$u'_{rel,n} = u_{rel,n} + N^T K_T N j_n \quad (7.13)$$

Relativna brzina između točaka nakon sudara u slučaju bez trenja jednaka je:

$$u'_{rel,n} = -\varepsilon u_{rel,n} \quad (7.14)$$

gdje je:

$\varepsilon$  koeficijent elastičnosti

Uvrštavanjem (7.14) u (7.13) dobije se konačna vrijednost impulsa:

$$\mathbf{j}_n = \frac{(-\varepsilon \mathbf{u}_{rel,n} - \mathbf{u}_{rel,n})}{N^T K_T N} \quad (7.15)$$

$$\mathbf{j} = \mathbf{j}_n N \quad (7.16)$$

Iz (7.5) i (7.6) slijedi da će brzine tijela nakon sudara biti:

$$\mathbf{v}' = \mathbf{v} \pm \frac{\mathbf{j}}{m} \quad (7.17)$$

$$\boldsymbol{\omega}' = \boldsymbol{\omega} \pm I^{-1}(\mathbf{r} \times \mathbf{j}) \quad (7.18)$$

## 7.1. Statičko i dinamičko trenje

Postoje dvije vrste trenja: statičko i dinamičko. Statičko trenje se odnosi na trenje kod kojeg tijelo miruje na podlozi dok se kod dinamičkog trenja tijelo kliže po podlozi. Svako tijelo ima svoj određeni koeficijent trenja  $\mu$  koji opisuje odnos sile trenja i sile kojom tijelo pritišće podlogu. Kod određivanja reakcije na sudar uzima se najveći od ta dva moguća koeficijenta. Prvo se odredi impuls s pretpostavkom da je statičko trenje dovoljno veliko tako da se tijela ne mogu micati. Dakle, za brzinu nakon sudara u smjeru tangente vrijedi  $\mathbf{u}'_{rel,t} = 0$ . Iz čega slijedi da će relativna brzina nakon sudara biti:

$$\mathbf{u}'_{rel} = -\varepsilon \mathbf{u}_{rel,n} N \quad (7.19)$$

Sada se može izračunati impuls  $\mathbf{j}$  uvrštavanjem (7.19) u (7.12):

$$\mathbf{j} = K^{-1}(-\varepsilon \mathbf{u}_{rel,n} N - \mathbf{u}_{rel}) \quad (7.20)$$

Prema Columb-ovom zakonu za silu trenja vrijedi:

$$F_{tr} \leq \mu N \quad (7.21)$$

Znači da ukoliko je sila kojom djelujemo na tijelo u smjeru tangente na podlogu veća od  $\mu N$  tijelo će se početi gibati i na njega će djelovati dinamičko trenje. Analogno tome za izračunati impuls sudara u (7.20) komponenta u smjeru

tangente mora biti veća od  $\mu j_n$  kako bi u trenutku sudara na tijelo djelovalo dinamičko trenje.

$$|j - (j \cdot N)N| \leq \mu \cdot j_n \quad (7.22)$$

Ukoliko vrijedi (7.22) izračunati impuls  $j$  u (7.20) je ispravan te se može primijeniti na rješavanje sudara. Inače je potrebno odrediti impuls uzimajući u obzir kinetičko

trenje. Neka je  $T = \frac{u_{rel,t}}{|u_{rel,t}|}$  smjer tangente sudara, impuls sudara zbog trenja biti će

u smjeru obrnuto smjera tangente:

$$j_t = -\mu j_n T \quad (7.23)$$

Ukupni impuls je zbroj impulsa sudara bez trenja i impulsa zbog dinamičkog trenja:

$$j = j_n N - \mu j_n T \quad (7.24)$$

Uvrštavanjem (7.24) i (7.14) u (7.12) pomnoženo s normalom i dobijemo impuls:

$$j_n = \frac{-(\varepsilon + 1)u_{rel,n}}{N^T K_T (N - \mu T)} \quad (7.25)$$

Iz (7.24) i (7.25) dobijemo konačni impuls sudara:

$$j = j_n (N - \mu T) \quad (7.26)$$

## 7.2. Trenje prilikom kotrljanja i vrtnje

Kada se točke kontakta ne mogu micati pod utjecajem statičkog impulsa, tijelo još uvijek ima slobodu da se kotrlja i vrti. Trenje ovisi o relativnoj kutnoj brzini između tijela  $\omega_{rel}$  i o koeficijentima trenja  $\mu_{spin}$  i  $\mu_{roll}$ . Brzina vrtnje u smjeru normale kontakta jednaka je  $\omega_{rel,n} = \omega_{rel} N$  i ona označava vrtnju tijela dok je brzina vrtnje u smjeru tangente  $\omega_{rel,t} = \omega_{rel} - \omega_{rel,n} N$  i označava kotrljanje tijela. Kutna brzina  $\omega'_{rel}$  odredi se tako da se veličine normalne i tangencijalne komponente umanjuju za vrijednosti  $\mu_{spin} j_n$  i  $\mu_{roll} j_n$ . Pri tome valja limitirati vrijednosti na nulu da se ne bi obrnuo smjer vrtnje tijela.

Kutni impuls kojim se smanjuje relativna kutna brzina za vrijednost  $\omega'_{rel} - \omega_{rel}$  jednak je:

$$j_t = L_T^{-1} (\omega'_{rel} - \omega_{rel}) \quad (7.27)$$

gdje je:

$$L_T = I_1^{-1} + I_2^{-1}$$

Za nove kutne brzine vrijedi:

$$\omega' = \omega \pm I^{-1} j_t \quad (7.28)$$

## 8. Rješavanje kontakata

Kod rješavanja sudara izvršeno je nekoliko iteracija tako da su se tijela odbila jedna od drugih nekoliko puta i dobilo se uvjerljivo ponašanje tijela. Iako možda i nisu riješeni svi sudari pa se tijela i dalje sudaraju svejedno se ažuriraju brzine i krene se na rješavanje kontakata. Algoritam za rješavanje kontakata jednak je algoritmu za rješavanje sudara osim što se za koeficijent elastičnosti koristi vrijednost nula tj. sudari će se tretirati kao da su neelastični. Sudare je ispravno tretirati na ovaj način zbog toga što se tijela koja se mnogo puta sudaraju unutar kratkog vremenskog intervala vrlo brzo gube energiju i prestaju se gibati.

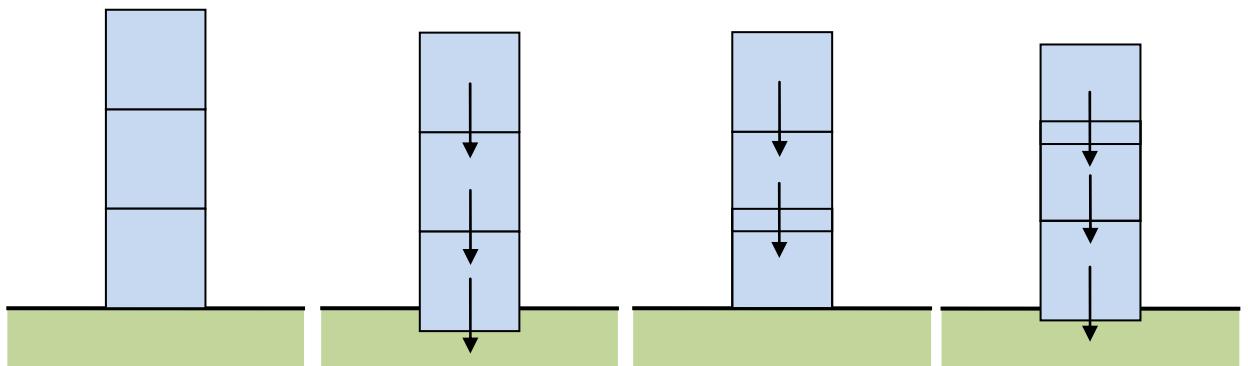
Isto kao i kod detektiranja sudara kontakti se detektiraju tako da se privremeno odrede novi položaji tijela koje bi poprimili u sljedećem vremenskom koraku i zatim se testiraju na sudare. Tako za tijela koja stoje na podlozi ništa se neće detektirati prilikom detekcije sudara ali nakon što im se promjeni brzina utjecajem gravitacije detektirat će se kontakti.

### 8.1. Koeficijent elastičnosti

Za svaki vrh koji se nalazi unutar drugog tijela označi se da je u kontaktu. Uzme se onaj vrh koji se nalazi najdublje unutar drugog tijela i giba se prema drugom tijelu i na njega se primjeni neelastični impuls. Za novu poziciju tijela ponovo se uzme vrh s najvećom dubinom sudara i na njega se primjeni neelastični impuls. Postupak se ponavlja sve dok se svi vrhovi barem jednom ne nađu izvan tijela ili se udaljavaju od tijela. Pri tome se za bolju preciznost umjesto primjenjivanja potpunog neelastičnog impulsa  $\varepsilon = 0$  za svaki vrh u kontaktu postepeno se u svakom koraku iteracije upotrebljavaju impulsi  $\varepsilon = -0.9$ ,  $\varepsilon = -0.8$ ,  $\varepsilon = -0.7$  sve do  $\varepsilon = 0$  u zadnjem koraku iteracije, čime se tijelo postepeno usporava umjesto da se potpuno zaustavi. Negativni koeficijent elastičnosti označava to da će se tijelo samo usporiti umjesto da promjeni smjer.

## 8.2. Propagirajući model

Veliki problem se javlja kada imamo tijela koja stoje jedno na drugom tako da će biti potrebno izvršiti više iteracija kako bi se svi kontakti detektirali. Takav način rješavanja kontakata naziva se propagirajući model za razliku od simultanog rješavanja kontakata.



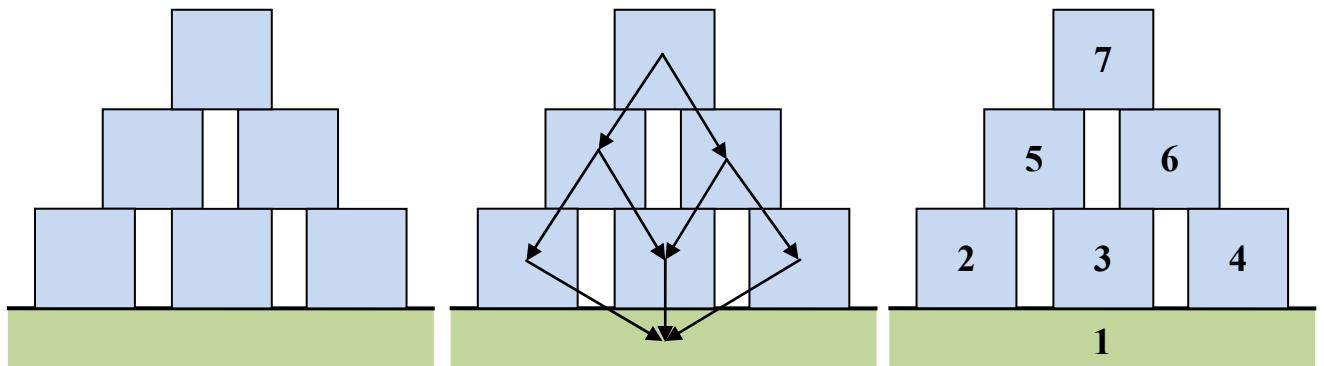
Slika 16. Propagirajući model

Neka za primjer imamo kocke koje stoje jedna na drugoj. Pod utjecajem gravitacije sve kocke će početi padati istom brzinom tako da će se samo za kocku na dnu detektirati kontakt sa podlogom. Rješavanjem kontakta kocka na dnu će doći u stanje mirovanja i u sljedećoj iteraciji će se detektirati kontakt između te kocke i kocke koja stoji na njoj. Sada će se riješiti kontakt kao neelastični sudar tako da se gornja kocka neće zaustaviti nego će se samo duplo sporije gibati. Da bi stvar bila još gora sada će se i donja kocka nastaviti gibati i biti će potrebno ponovo riješiti sudar između nje i podloge kako bi se zaustavila. Vidimo da tijela sporo konvergiraju ispravnom rješenju pa je potrebno mnogo iteracija kako bi se dobilo uvjerljivo ponašanje tijela.

## 8.3. Kontaktni graf

Kako bi se povećala efikasnost rješavanja kontakata potrebno je konstruirati kontaktni graf. Pomoću grafa će se odrediti redoslijed kojim će se rješavati kontakti između tijela. Željeni redoslijed rješavanja kontakata je u smjeru od podlove i drugih statičkih objekata prema vrhu. Nakon određivanja novih brzina tijela zbog utjecaja gravitacije za svako tijelo se odredi njegova nova pozicija u sljedećem vremenskog koraku, zatim se odrede svi kontakti između tog i ostalih tijela. Za svaki se kontakt u grafu doda usmjereni brid koji pokazuje od trenutnog tijela prema

tijelu s kojim je detektiran kontakt. Tijela je potrebno sortirati a to se postiže pretraživanjem grafa u dubinu. Tako se za naslagane kocke dobije sortirana lista kocaka od dna prema vrhu kao što pokazuje sljedeća slika.



Slika 17. Konstrukcija kontaktnog grafa

Za usmjereni aciklički graf  $G$  sortiranje je prilično jednostavno. Sljedećim algoritmom stvara se sortirana lista čvorova:

```

DFS (G)
{
    za svaki čvor u iz G
        u.visited = false;

    za svaki čvor u iz G
        ako (!u.visited) DFS-Visit (u);
}

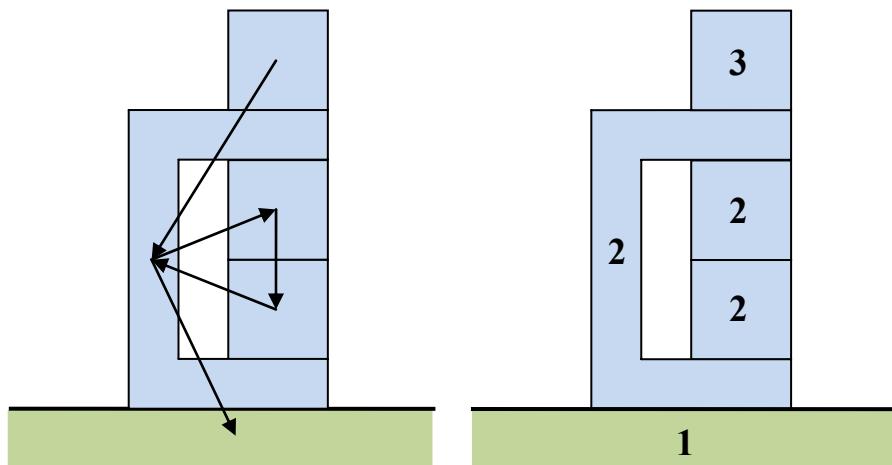
DFS-Visit (u)
{
    u.visited = true;

    za svaki čvor v iz u.čvoroviDjeca
        ako (!v.visited) DFS-Visit (v);

    dodaj čvor u u listu čvorova;
}

```

Za složenije slučajeve poput skupa domino pločica složenih u krug gdje svaka leži na onoj ispod sebe u jednu razinu se stavi podloga a sve pločice se stave u drugu razinu. Znači sva tijela koja međusobno imaju cikličku vezu postavljaju se u istu razinu kao što se vidi na slici 18.

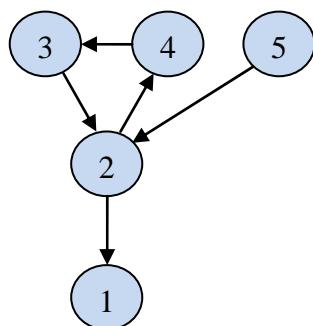


Slika 18. Rješavanje cikličke veze u kontaktnom grafu

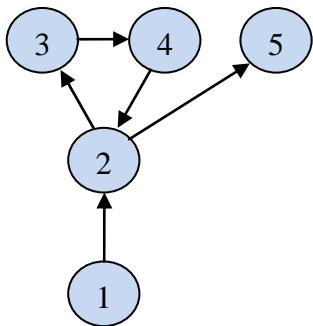
Sortiranje grafa koji ima cikličke veze obavlja se na sljedeći način:

1. izračunaj inverz grafa  $G^T$ .
2. pozovi  $\text{DFS}(G^T)$  kako bi se odredio redoslijed izlaza iz čvorova.
3. pozovi  $\text{DFS}(G)$  s time da se početni čvorovi u glavnoj petlji od DFS uzimaju obrnutim redoslijedom od redoslijeda izračunatog u koraku 2.
4. Za svaki novi čvor u glavnoj petlji od DFS broj razine povećaj za jedan, a sve čvorovi do kojih se dolazi rekurzivnim pozivom DFS-Visit označi trenutnim brojem razine.

Za primjer na slici 18. Dobije se sljedeći kontaktni graf  $G$ .



Inverz  $G^T$  grafa  $G$  dobije se promjenom smjera svih veza unutar grafa.



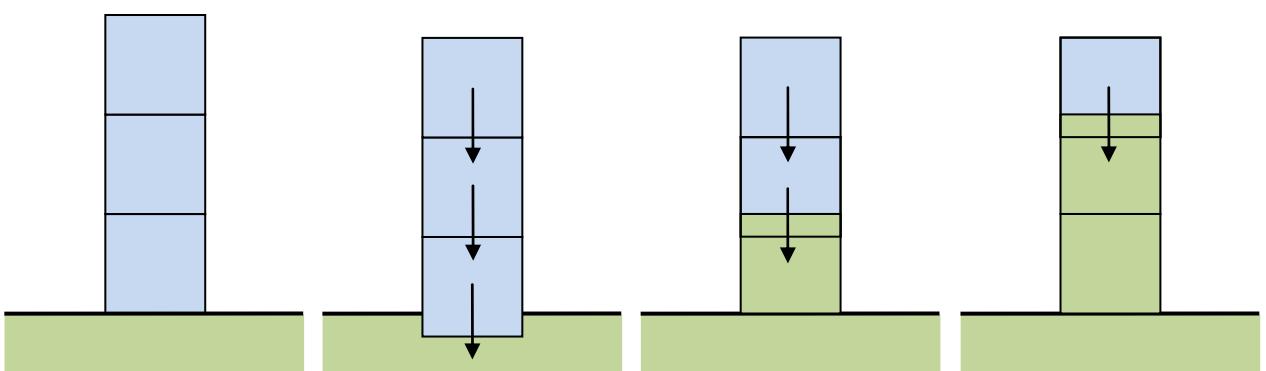
Pretraživanjem u dubinu inverznog grafa  $G^T$  u koraku 2 dobije se sortirana lista  $(4,3,5,2,1)$ . U koraku 3 graf  $G$  se pretražuje obrnutim redoslijedom od redoslijeda zapisanog u listi pa se tako prvo čvor 1 stavi u prvu razinu, zatim se u sljedeću razinu stave čvorovi 2,3,4 i u zadnjoj razini čvor 5 čime dobijemo sortirane čvorove po razinama kao na slici 17. Dokaz da je prethodni algoritam ispravan te njegov detaljniji opis mogu se pronaći u [10] i [11].

## 8.4. Propagiranje šoka

Bez obzira na korištenje kontaktnog grafa svejedno će biti potreban dosta veliki broj iteracija da bi se dobila uvjerljiva simulacija. No neovisno o tome koliki je broj iteracija korišten nakon nekog vremena će se primijetiti da tijela propadaju jedno u drugo pogotovo za simulaciju većeg broja tijela naslaganih jedno na drugo. Da bi se riješio taj problem koristit će se metoda propagiranja šoka [1]. To znači da će se nakon rješavanja kontakata izvršiti još jedna iteracija na poseban način. Nakon rješavanja kontakata u svakom nivou grafa za sva tijela u tom nivou masa se postavlja na beskonačnu vrijednost (matrica sudara se postavi na nulu). Sada se vidi prava svrha kontaktnog grafa. Ako se tijelo kojem je masa postavljena na beskonačnu vrijednost nađe u sudaru s nekim tijelom iz višeg nivoa njegova brzina se neće promijeniti pod utjecajem impulsa nego će tijelo iz višeg nivoa dobiti duplo veću brzinu tako da će se tijela ispravno razdvojiti. Nakon što završi propagiranje šoka, mase tijela je potrebno vratiti na prethodnu vrijednost. Valja napomenuti da za dva tijela koja su u kontaktu i nalaze se u istom nivou nijednom još nije postavljena beskonačna masa tako da se među njima rješavanje kontakata

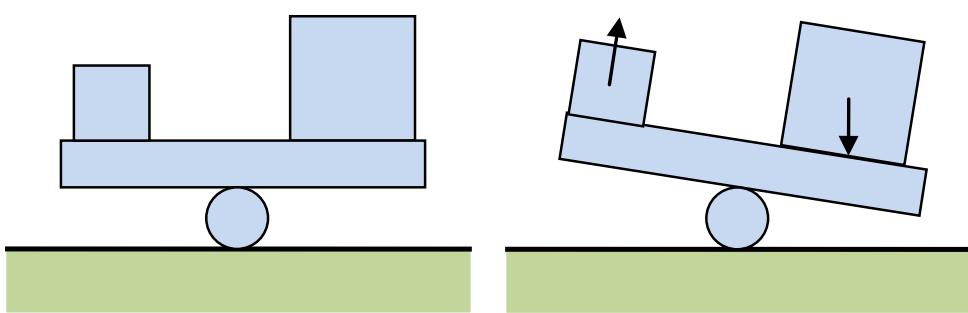
obavlja na uobičajen način. Međutim sada je spora konvergencija tijela lokalizirana na manji skup.

Primjer rada algoritma se može vidjeti na slici 19. Prolaskom kroz graf kontakata u prvom nivou zbog kontakta s podlogom kocka na dnu se zaustavlja te joj se masa postavlja na beskonačnu vrijednost. U sljedećem nivou kocka koja je iznad nje se zaustavlja i masa joj se postavlja na beskonačnu vrijednost i tako se nastavlja sve do zadnjeg nivoa. Vidimo kako nam kontaktni graf omogućuje da se u jednom prolazu ispravno zaustave sva tijela.



Slika 19. Princip rada metode propagiranja šoka.

Da bi vidjeli zašto je potrebno prvo izvršiti nekoliko iteracija rješavanja kontakata pomoću propagirajućeg modela prije primjene propagiranja šoka poslužit će primjer na slici 20. Kontaktni graf pokazuje od podloge prema vrhu. Ako bi se sada odmah primijenilo propagiranje šoka, daska bi poprimila beskonačnu masu tako da bi desna teža kocka vidjela dasku beskonačne mase i ne bi mogla gurnuti dasku prema dolje. Propagirajući model nam omogućava da daska dobije "osjećaj težine". Korištenjem određenog broja iteracija, daska i tijela u kontaktu s njom dobiju odgovarajuće brzine prije primjene konačnog propagiranja šoka, čime je omogućeno da desna kocka gurne dasku prema dolje i da daska gurne lijevu kocku prema gore.



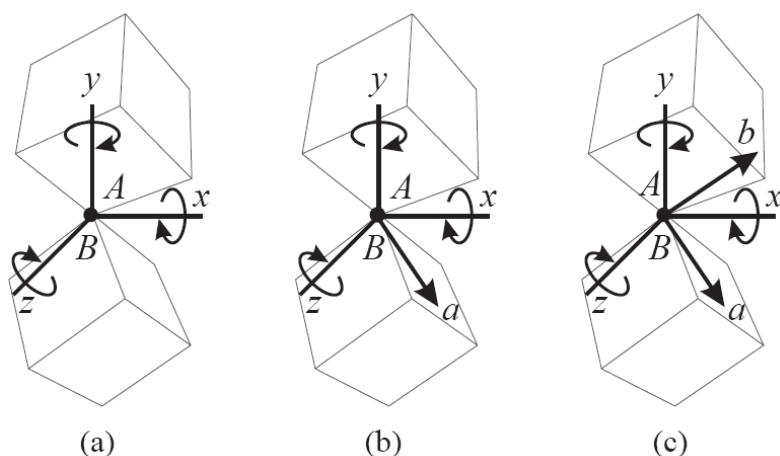
*Slika 20. Propagirajući model omogućava da teža kocka gurne dasku prema dolje i da zatim daska gurne lijevu kocku prema gore.*

## 9. Zglobovi

Kao što su se impulsi koristili za modeliranje sudara i kontakata na sličan način možemo odrediti impulse za modeliranje zglobova tj. za određivanje ograničenja pozicije i orientacije. Postoje dvije osnovne vrste zglobova: translacijski i rotacijski zglobovi.

### 9.1. Translacijski zglobovi

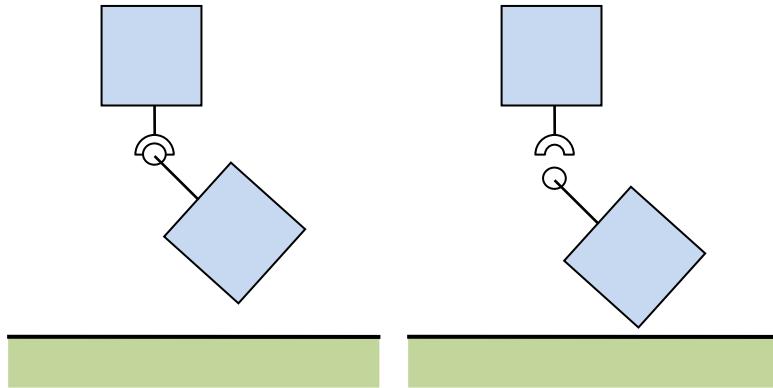
Translacijski zglobovi brišu translacijske stupnjeve slobode, to znači da se tijela neće moći translatirati u određenim smjerovima ali će se moći rotirati u svim smjerovima.



Slika 21. Stupnjevi slobode translacijskih zglobova

#### 9.1.1. Translacijski zglob s ograničena tri stupnja slobode

Najjednostavniji primjer zgloba je sferni zglob koji se vidi na slici 21a. Zglob spaja jednu točku tijela A s jednom točkom tijela B tako da se tijela jedino mogu rotirati oko zajedničke spojne točke. Ovaj zglob briše sva tri translacijska stupnja slobode. Naravno da se tijela mogu translatirati u globalnom sustavu, međutim u odnosu jedan na prema drugom tijela se ne mogu translatirati.



Slika 22. Sferni zglob

Ako bi se tijela simulirala bez uzimanja u obzir ograničenja, tijela će se razdvojiti kao na slici 22. Potrebno je pronaći impuls  $j$  na početku simulacijskog koraka kojim će se eliminirati udaljenost  $\Delta p$  koja nastaje na kraju simulacijskog koraka (stabilizacija pozicije). Udaljenost na kraju vremenskog koraka jednaka je:

$$\Delta p = x_1 + R_1 \cdot r_1 - x_2 - R_2 \cdot r_2 \quad (9.1)$$

gdje su:

$x_1, x_2$  pozicije tijela na kraju vremenskog koraka

$R_1, R_2$  orijentacije tijela na kraju vremenskog koraka

$r_1, r_2$  pozicije točke zgloba u lokalnim koordinatama tijela

Budući da točke na tijelu uglavnom imaju nelinearno gibanje odgovarajući impuls se može odrediti rješavanjem nelinearne jednadžbe iterativno kao u [12] no ovdje će se odrediti na pojednostavljen način kao u [13]. Ako je relativna brzina između dvije točke u simulacijskom koraku promijenjena za  $\frac{\Delta p}{\Delta t}$  kao da je relativno gibanje

između točaka linearno tada će izračunati impuls  $j$  kojim se eliminira brzina  $\frac{\Delta p}{\Delta t}$  smanjiti udaljenost  $\Delta p$  no neće se potpuno eliminirati. Stoga je potrebno iterativno nekoliko puta izračunati impuls  $j$  sve dok udaljenost ne nestane s određenom tolerancijom. U [13] su odredili da je za vremenski korak od 0.04s koji odgovara simulaciji od 25 sličica u sekundi dovoljno jedna do dvije iteracije da bi se eliminirala udaljenost s tolerancijom od  $10^{-6}$ m, osim u slučajevima gdje tijela imaju veliku brzinu. U primjerima napravljenim u ovom radu dovoljna je bila i jedna

iteracija za uvjerljivu simulaciju. Prednost ovakvog pojednostavljenja je da je jednadžba za određivanje odgovarajućeg impulsa linearna pa je impuls lako za izračunati.

Iz (7.12) slijedi da za impuls kojim se eliminira brzina  $\Delta u$  vrijedi:

$$\Delta u = K_T j \quad (9.2)$$

Iz čega se slijedi da za impuls kojim se eliminira udaljenost  $\Delta p$  na kraju vremenskog koraka vrijedi:

$$\frac{\Delta p}{\Delta t} = K_T j \quad (9.3)$$

gdje je:

$\Delta t$  veličina vremenskog koraka

Nakon završetka simulacijskog koraka potrebno je još zadovoljiti ograničenja brzine (stabilizacija brzine). Relativna brzina točaka tijela koje odgovaraju zglobu mora biti jednaka nuli tako da je potrebno eliminirati brzinu na kraju simulacijskog koraka. Brzina na kraju simulacijskog koraka jednaka je:

$$\Delta u = v_2 + \omega_2 \times r_2 - v_1 - \omega_1 \times r_1 \quad (9.4)$$

gdje su:

$v_1, v_2$  brzine tijela na kraju vremenskog koraka

$\omega_1, \omega_2$  kutne brzine tijela na kraju vremenskog koraka

$r_1, r_2$  pozicije točke zgloba u lokalnim koordinatama tijela

Impuls kojim se eliminira brzina  $\Delta u$  određen je jednadžbom (9.2).

### 9.1.2. Translacijski zglob s ograničena dva stupnja slobode

Druga vrsta translacijskog zgloba je zglob koji briše dva stupnja slobode i prikazan je na slici 21b. Zglob je određen točkama zgloba  $A$  i  $B$  u lokalnim koordinatama prvog i drugog tijela i vektora osi rotacije  $a$  u lokalnim koordinatama prvog tijela. Točka  $B$  može se slobodno gibati na liniji  $A + \lambda a$ . Prije računanja impulsa točka  $A$  trebala bi se pomaknuti po liniji određenoj s osi  $a$  na poziciju

gdje je najmanja udaljenost od točke  $B$ . Prilikom implementacije točka se neće direktno pomicati po liniji nego će se impuls izračunati na drugi način. Impuls kojim se eliminira promjena brzine  $\Delta u$  određen je jednadžbom (9.3). Impuls se izračuna projekcijom jednadžbe (9.3) u dvodimenzionalni prostor. Potrebno je odrediti linearne nezavisne vektore  $b$  i  $c$  okomite na vektor  $a$ . Vektor  $b$  izračuna se pomoću vektorskog produkta vektora  $a$  i vektora jedne od osi koordinatnog sustava, na primjer  $v = (1,0,0)$ . Problem je što vektorski produkt između dva paralelna vektora nije definiran pa ukoliko je kut između  $a$  i  $v$  približno jednak nuli vektor  $b$  se izračuna kao vektorski produkt vektora  $a$  i vektora neke druge osi npr.  $v = (0,1,0)$ . Za uvjet kuta se uzima vrijednost približno jednak nula a ne jednak nuli zbog bolje preciznosti prilikom računanja okomitog vektora. Vektor  $c$  jednak je vektorskem produktu vektora  $a$  i  $b$ . Izračunati vektori  $b$  i  $c$  odgovaraju stupnjevima slobode koje treba ograničiti. Slijedi da za jednadžbu pomoću koje se računa impuls vrijedi:

$$P \frac{\Delta p}{\Delta t} = PK_T P^T j' \quad (9.5)$$

gdje su:

$$P = \begin{pmatrix} b^T \\ c^T \end{pmatrix} \in R^{2 \times 3} \text{ matrica projekcije}$$

$j'$  projicirani impuls u dvije dimenzije

Impuls u tri dimenzije određen je sa:

$$j = P^T \cdot j' \quad (9.6)$$

Na isti način se odredi i impuls kojim se eliminira promjena brzine  $\Delta u$

$$P\Delta u = PK_T P^T j' \quad (9.7)$$

Također je potrebno impuls  $j'$  pomoću izraza (9.6) transformirati u 3D prostor.

### 9.1.3. Translacijski zglob s ograničenim jednim stupnjem slobode

Ovaj translacijski zglob briše samo jedan stupanj slobode i prikazan je na slici 21c. Zglob je određen točkama zgloba  $A$  i  $B$  u lokalnim koordinatama prvog i drugog tijela i vektora osi rotacije  $a$  i  $b$  u lokalnim koordinatama prvog tijela. Točka  $B$  može se slobodno gibati na ravnini  $A + \lambda a + \mu b$ . Potrebno je prije računanja impulsa pomaknuti točku  $A$  po ravnini na poziciju gdje je najmanje udaljena od točke  $B$ . Dakle, impuls će se izračunati projekcijom jednadžbi (9.2) i (9.3) u jednodimenzionalni prostor. Matrica projekcije jednaka je  $P = (c^T) \in R^{1x3}$  a impulsi kojima se eliminira promjena udaljenosti  $\Delta p$  i brzine  $\Delta u$  računaju se rješavanjem jednadžbi na isti način kao i u prethodnom poglavljju. I zglob sa sva tri ograničena stupnja slobode bi se također mogao riješiti na isti način s time da se za matricu projekcije uzme jedinična matrica.

## 9.2. Rotacijski zglobovi

Rotacijski zglobovi brišu jedan do tri rotacijska stupnja slobode. Za razliku od prijašnjih impulsa ovdje će biti potrebno izračunati kutni impuls  $j_T$ . Kutni impuls mijenja samo kutnu brzinu. Iz (7.18) se može vidjeti da promjena kutne brzine pod utjecajem kutnog impulsa iznosi:

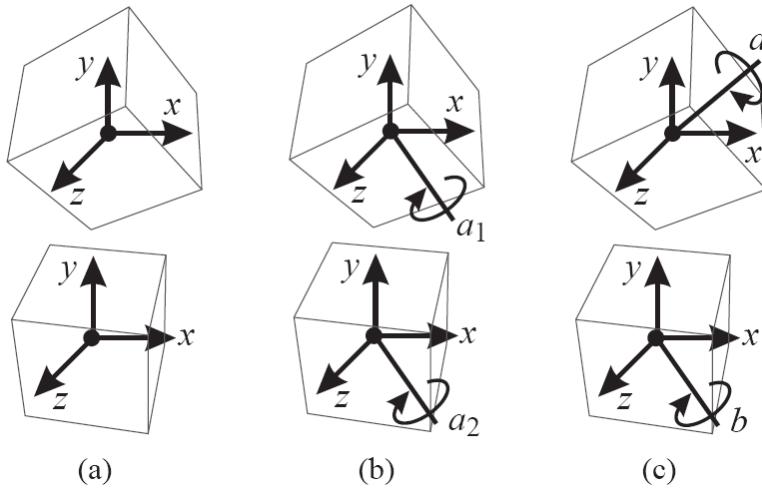
$$\omega' = \omega \pm I^{-1} j_T \quad (9.8)$$

Sada se može izračunati kutni impuls iz relativne promjene brzine:

$$\Delta\omega = L_T j_T \quad (9.9)$$

gdje je:

$$L_T = I_1^{-1} + I_2^{-1}$$



Slika 23. Stupnjevi slobode rotacijskih zglobova

### 9.2.1. Rotacijski zglob s ograničena tri stupnja slobode

Za zglob koji briše sva tri rotacijska stupnja slobode vrijedi da se mogu translatirati u svim smjerovima ali se ne mogu rotirati relativno u donosu jedan na prema drugom, prikazan je na slici 23a. Slično kao što je kod translacijskog zgloba bilo potrebno eliminirati udaljenost na kraju simulacijskog koraka tako je ovdje potrebno eliminirati rotaciju koja nastaje na kraju simulacijskog koraka. Promjena u relativnoj orientaciji između tijela na kraju simulacijskog koraka jednaka je:

$$\Delta q = (q_2(t_0)^{-1} \cdot q_2(t_0 + \Delta t))^{-1} \cdot (q_1(t_0)^{-1} \cdot q_1(t_0 + \Delta t)) \quad (9.10)$$

gdje su:

$q_1(t_0)$ ,  $q_2(t_0)$  orijentacije tijela na početku simulacijskog koraka

$q_1(t_0 + \Delta t)$ ,  $q_2(t_0 + \Delta t)$  orijentacije tijela na kraju simulacijskog koraka

Kvaternion  $\Delta q$  potrebno je pretvoriti u os rotacije  $a$  i kut rotacije  $\alpha$ . Rotacija koju je potrebno eliminirati jednaka je  $\Delta d = \alpha \cdot a$ . Iz (9.9) slijedi da za kutni impuls kojim se eliminira rotacija  $\Delta d$  vrijedi:

$$\frac{\Delta d}{\Delta t} = L_T \cdot j_T \quad (9.11)$$

Odgovarajući impuls  $j_T$  se računa iterativno sve dok rotacija  $\Delta d$  ne nestane s odgovarajućom tolerancijom. Na kraju simulacijskog koraka mora se eliminirati i

razlika u relativnoj kutnoj brzini između tijela  $\Delta\omega = \omega_2 - \omega_1$ . Za impuls kojim se eliminira kutna brzina  $\Delta\omega$  vrijedi:

$$\Delta\omega = L_T \cdot j_T \quad (9.12)$$

### 9.2.2. Rotacijski zglob s ograničena dva stupnja slobode

Rotacijski zglob koji briše dva rotacijska sloboda stupnja prikazan je na slici 23b. Tijela se mogu rotirati oko jedne zajedničke osi i translatirati u svim smjerovima. Zglob je određen s dvije osi  $a_1$  i  $a_2$  koje predstavljaju tu zajedničku os u lokalnim koordinatama tijela. Potrebno je odrediti linearne nezavisne vektore  $b$  i  $c$  okomite na vektor  $a_1$ . Vektori se odrede na isti način kako je opisano u poglavlju o translacijskim zglobovima. Greška koja nastaje prilikom simulacije tj. razlika u orientaciji na kraju simulacijskog koraka koju treba ispraviti jednaka je vektorskom produktu dviju rotacijskih osi u trenutku  $t_0 + \Delta t$ .

$$\Delta d = a_1(t_0 + \Delta t) \times a_2(t_0 + \Delta t) \quad (9.13)$$

Odgovarajući impuls izračunat će se projekcijom jednadžbe (9.11) u ravninu određenu sa vektorima  $b$  i  $c$ . Dakle, vrijedi:

$$P \frac{\Delta d}{\Delta t} = PL_T P^T \cdot j'_T \quad (9.14)$$

gdje su:

$$P = \begin{pmatrix} b^T \\ c^T \end{pmatrix} \in R^{2 \times 3} \text{ matrica projekcije}$$

$j'$  projicirani impuls u dvije dimenzije

Za kutni impuls kojim se eliminira razlika u kutnim brzinama  $\Delta\omega$  vrijedi:

$$P\Delta\omega = PL_T P^T \cdot j'_T \quad (9.15)$$

Kutni impuls u tri dimenzije jednak je  $j_T = P^T \cdot j'_T$ .

### 9.2.3. Rotacijski zglob s ograničenim jednim stupnjem slobode

Rotacijski zglob s ograničenim samo jednim rotacijskim stupnjem slobode prikazan je na slici 23c. Tijela se mogu translatirati u svim smjerovima i rotirati oko dva linearno nezavisna vektora  $a$  i  $b$ . Zglob je određen s dva vektora  $a$  i  $b$  pri čemu se  $a$  nalazi u lokalnom koordinatnom sustavu tijela prvog tijela dok se  $b$  nalazi u lokalnom koordinatnom sustavu drugog tijela. Oba dva vektora su normalizirana. Zglob onemogućava rotiranje oko osi  $c = a \times b$ , to znači da kut između vektora  $a$  i  $b$   $\varphi = \arccos(a \cdot b)$  mora biti jednak tokom cijele simulacije. Greška koja nastaje na kraju simulacijskog koraka tj. razlika u rotaciji koju treba eliminirati jednaka je:

$$\Delta d = (\varphi(t_0 + \Delta t) - \varphi(t_0)) \cdot c \quad (9.16)$$

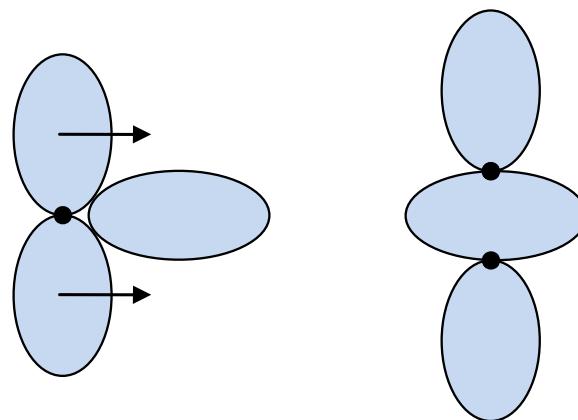
Kutni impuls kojim se eliminira nastala greška računa se projekcijom jednadžbe (9.11) u jednodimenzionalni prostor. Matrica projekcije jednaka je  $P = (c^T) \in R^{1x3}$ . Razlika u kutnim brzinama  $\Delta\omega$  eliminira se projekcijom jednadžbe (9.12) s istom matricom projekcije. Svi rotacijski zglobovi se mogu simulirati pomoću jednadžbi (9.11) i (9.12) s time da se za rotacijski zglob s ograničena tri stupnja slobode koristi jedinična matrica za matricu projekcije.

## 10. Hibridni model

Prednost prethodne metode modeliranja zglobova je što se može jednostavno integrirati u simulaciju sa sudarima i kontaktima. Korak integracije je sada:

- detekcija i rješavanje sudara i stabilizacija brzine
- određivanje novih brzina i stabilizacija brzine
- detekcija i rješavanje kontakata i stabilizacija pozicije
- određivanje novih pozicija i stabilizacija brzine

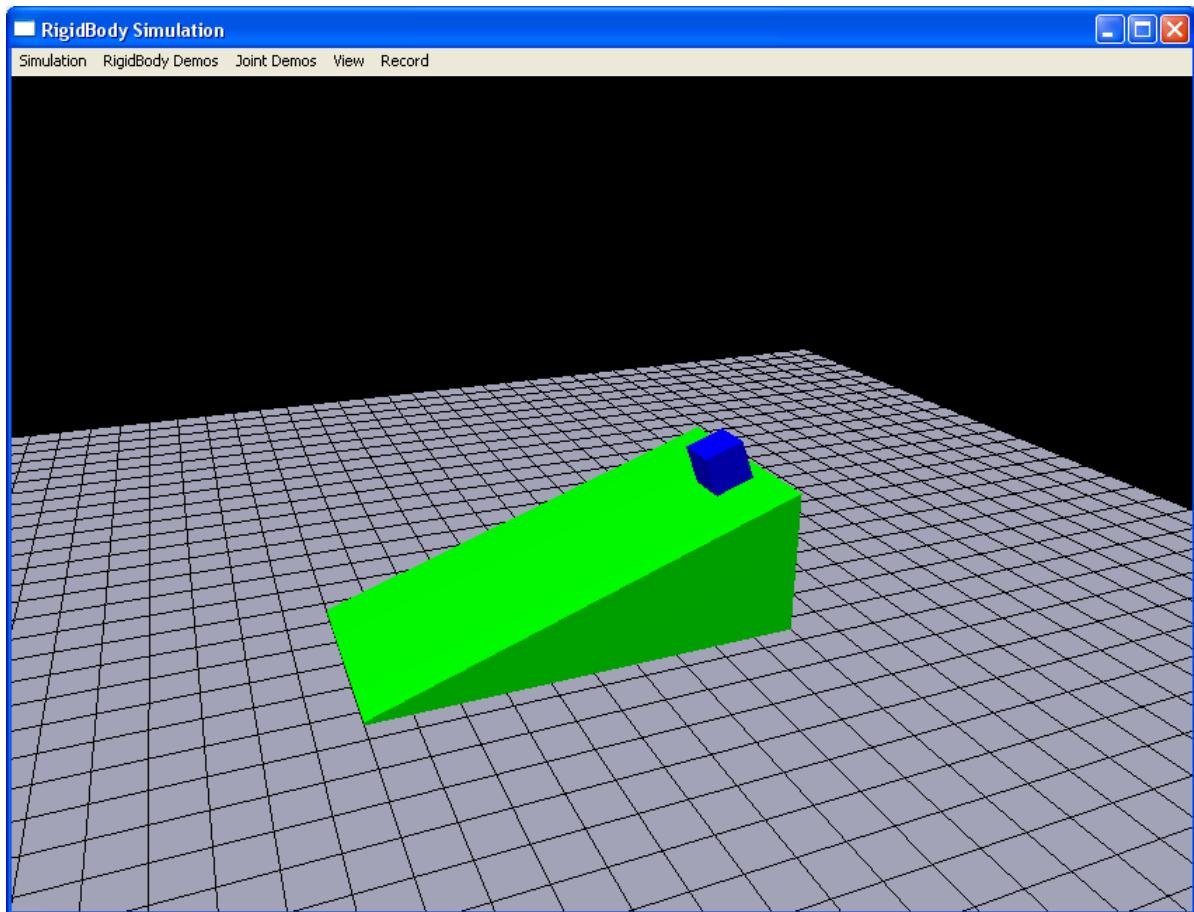
Budući da se prilikom rješavanja sudara i određivanja novih brzina pod utjecajem gravitacije mijenjaju brzine potrebno je izvršiti stabilizaciju brzine nakon tih koraka. Stabilizacija pozicije je povezana s rješavanjem kontakata. Nakon konstruiranja kontaktnog grafa rješavaju se kontakti jedan po jedan od najniže prema najvišoj razini. Nakon rješavanja svih kontakata između tijela na određenoj razini s tijelima na nižoj razini rješavaju se ograničenja (zglobovi) između tijela na toj razini i tijela na nižoj razini. Također se izvodi i propagiranje šoka na isti način. Na kraju se odrede nove pozicije i orijentacije i ponovo se izvede stabilizacija brzine. Razlog zbog kojeg se stabilizacija pozicije vrši unaprijed je zbog toga jer ako bi se stabilizacije izvela nakon određivanja novih pozicija došlo bi do preklapanja objekata. Na sljedećoj slici se vidi slučaj kada se ne koristi stabilizacija unaprijed. Tijela na slici lijevo bi došla u konfiguraciju prikazanu desno i tada bi se pokušalo spojiti točke zgloba, što je nemoguće bez preklapanja tijela. Korištenjem stabilizacije unaprijed onemogućuje se uopće dolazak u desnu konfiguraciju [12].



Slika 24. Stabilizacija unaprijed

## 11. Implementacija

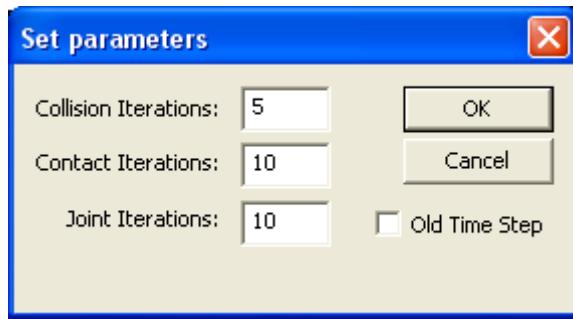
Program za simulaciju je napravljen u programskom jeziku C++ korištenjem OpenGL grafičkog sučelja te koda za stvaranje .avi datoteka AVIGenerator.cpp.



Slika 25. Program za simulaciju

Pritiskom lijeve tipke miša i pomicanjem miša moguće je rotirati kameru dok je kursorskim tipkama moguće pomicati kameru. Ostalo upravljanje programom obavlja se pomoću menija u gornjem dijelu prozora.

U grupi Simulation moguće je pokretati i zaustavljati simulaciju te odrediti parametre simulacije a to su broj iteracija rješavanja sudara, kontakata zglobova te korištenje novog ili starog koraka simulacije.



Slika 26. Postavljanje parametra simulacije

U grupama RigidBody Demos i JointDemos pokreću se razne simulacije.

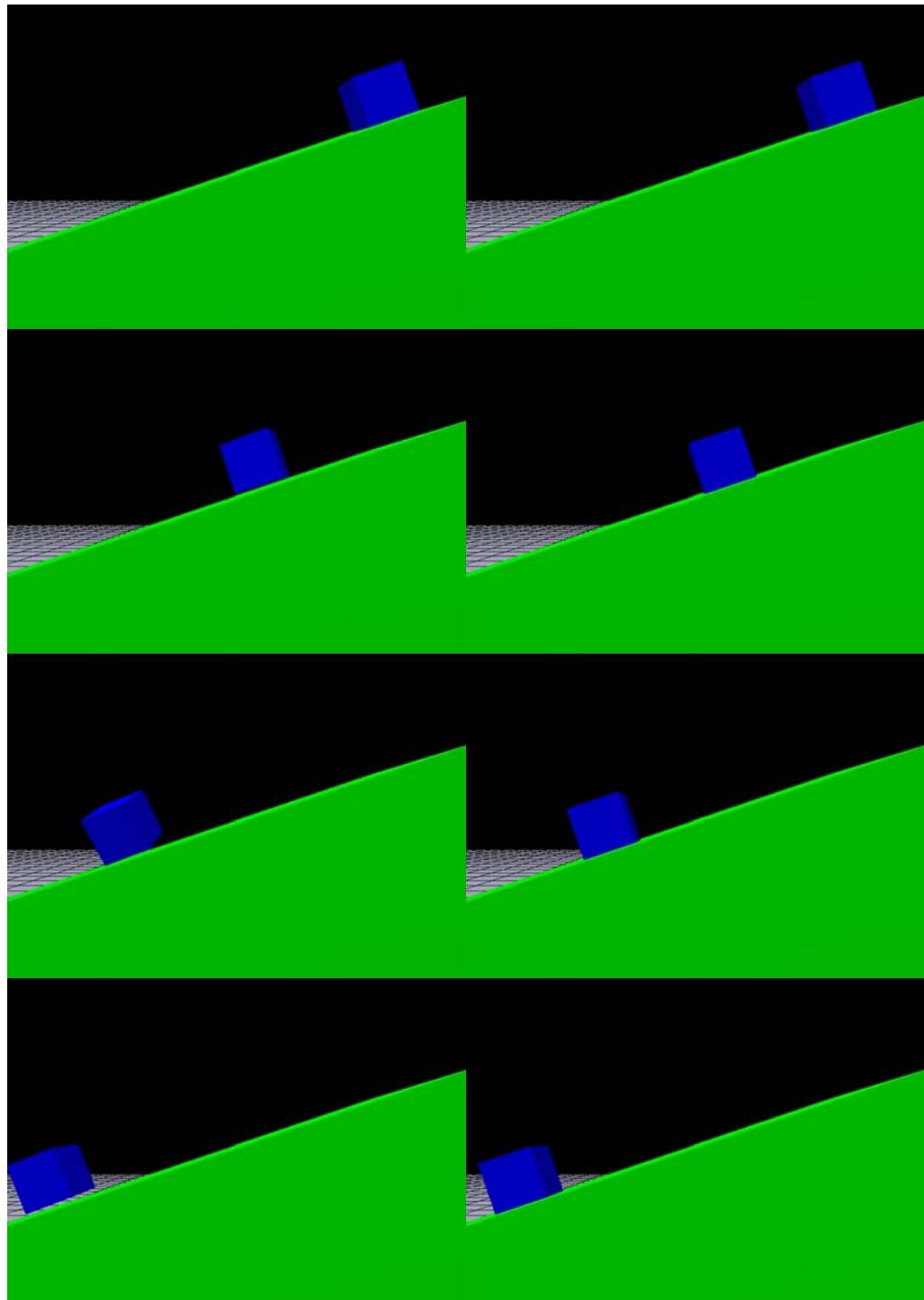
U grupi view moguće je odrediti način prikaza tijela.

- Wireframe prikazuje žičani oblik tijela.
- Sample Points prikazuje točke koje se koriste prilikom detekcije sudara.
- Depth Field prikazuje polje udaljenosti gdje su zelenom bojom prikazane točke unutar tijela a crvenom točke izvan tijela.
- World AABB za svako tijelo prikazuje omeđujući kvadar poravnat s koordinatnim osima.
- Planar Shadows za svako tijelo prikazuje njegovu sjenu.

U grupi Record moguće je pokrenuti i zaustaviti snimanje simulacije.

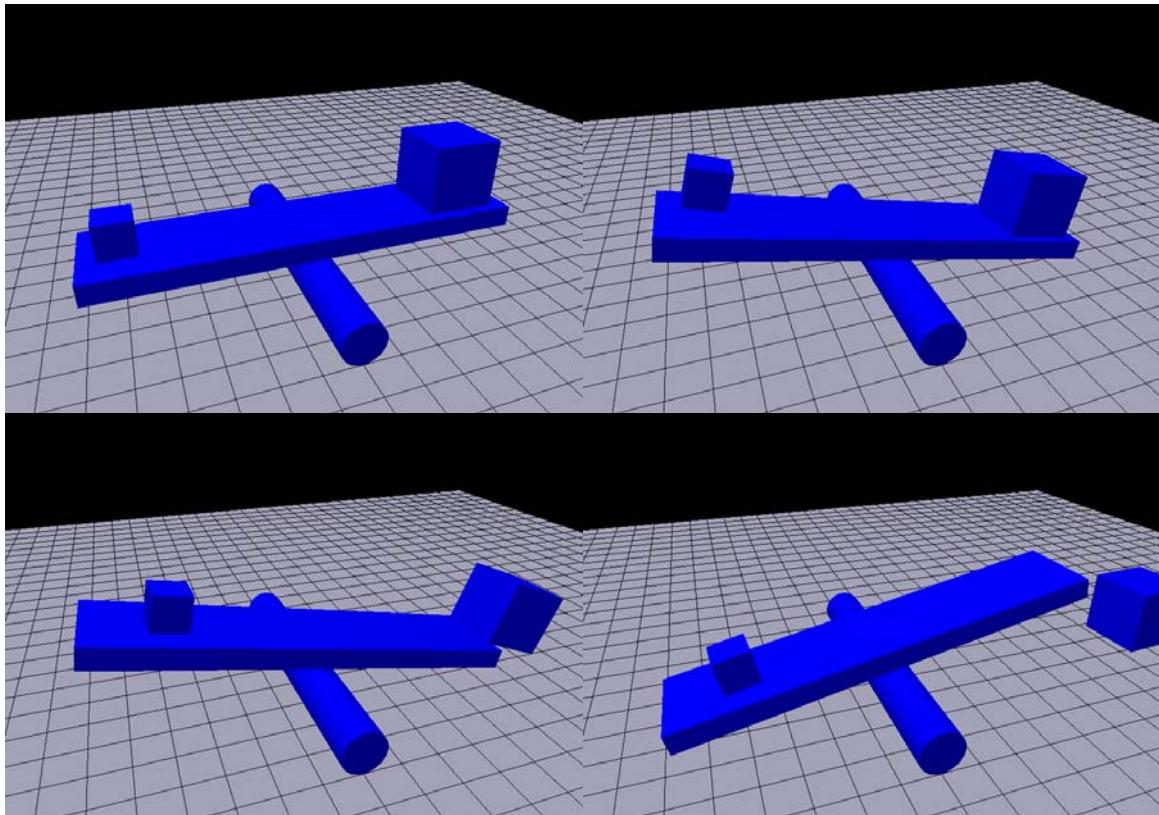
## 12. Rezultati

Dobiveni rezultati prikazani su u sljedećim primjerima. Ukoliko nije drugčije navedeno korišteni parametri su 5 iteracija rješavanja sudara, 10 iteracija rješavanja kontakata i 10 iteracija rješavanja zglobova.



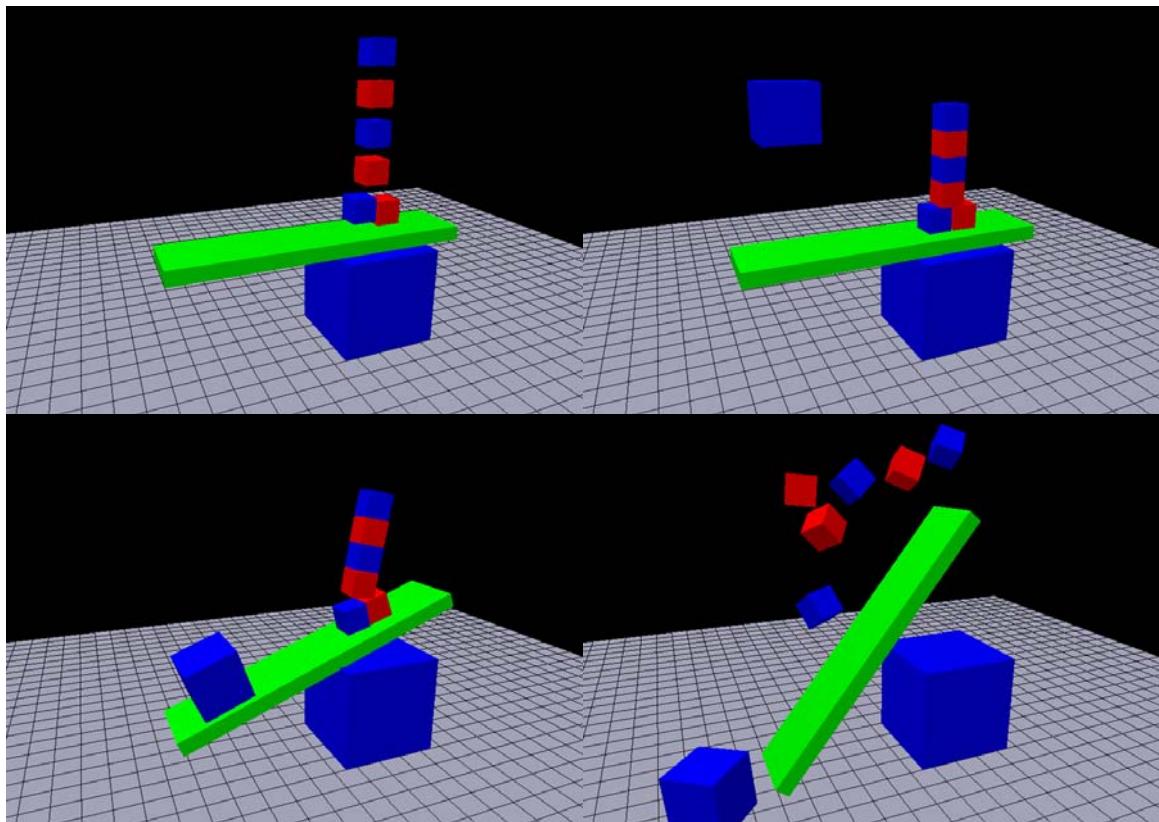
Slika 27. Savršeno elastično tijelo na kosini

Na slici 27 prikazan je primjer savršeno elastičnog tijela na kosini gdje je u lijevom stupcu korišten standardni korak integracije, dok je u desnom stupcu korišten novi postupak integracije opisan u petom poglavlju. Vidi se kako u prvom primjeru kocka nekontrolirano skakuće dok u drugom primjeru ispravno klizi niz kosinu.



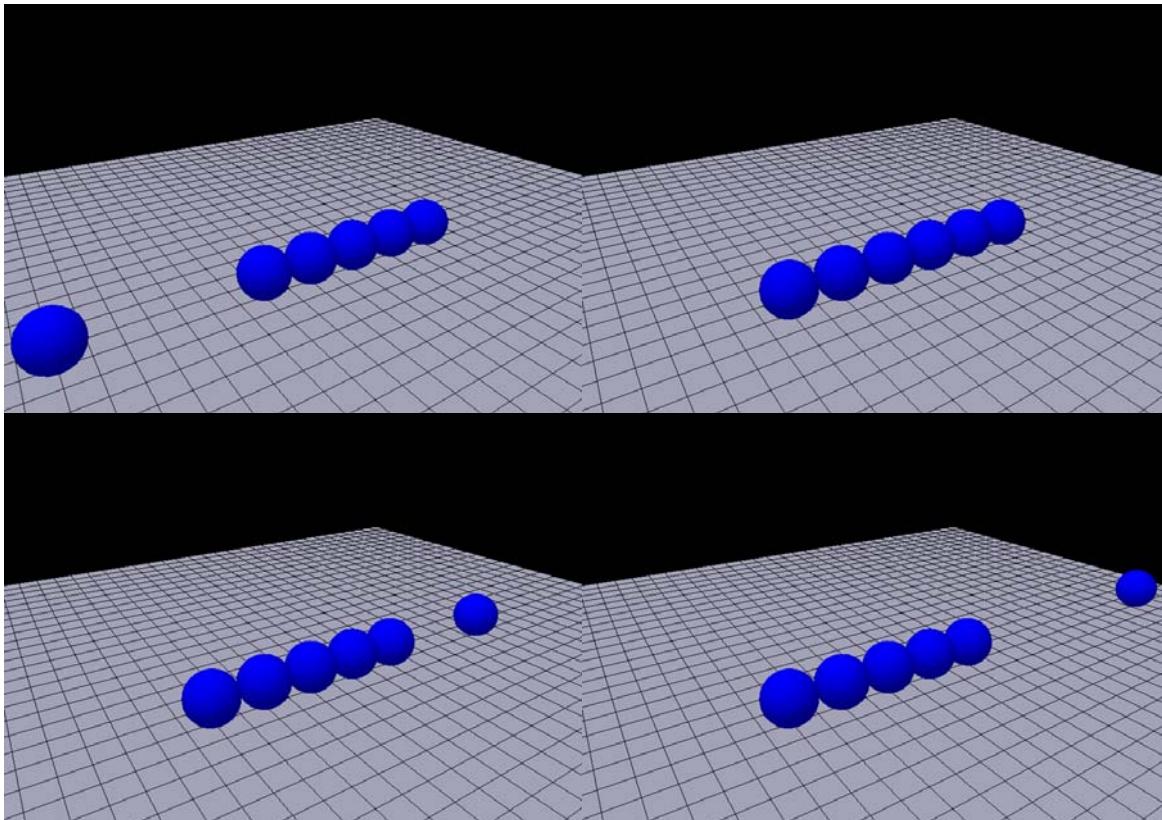
Slika 28. Klackalica

Na prethodnoj slici prikazan je primjer klackalice. Teža kocka ispravno gurne lakšu kocku i nakon što teža kocka otkliže s klackalice lakša kocka gurne nazad klackalicu prema dolje. Ukoliko se koristi samo propagiranje šoka bez da se izvrši nekoliko koraka rješavanja kontakata klackalica će ostati u početnom položaju i neće se uopće pomaknuti.



Slika 29. Prevrtanje kocki

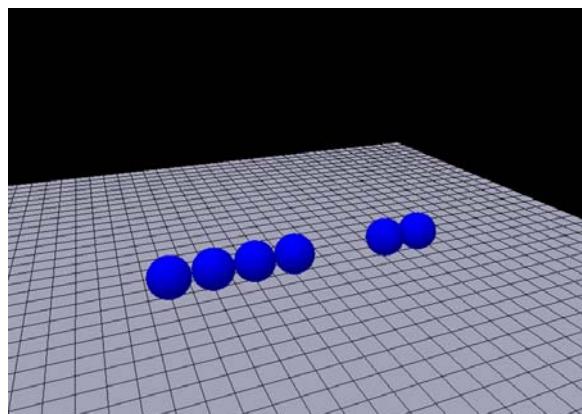
Na slici 29 je primjer u kojem je uvjерljivo prikazano prevrtanje kocki. Prvo se daska lagano počinje prevrtati, zatim se kocke naslažu na dasku i onemogućuju dasku da se prevrne te se dolaskom veće kocke prevrće daska zajedno s naslaganim kockama.



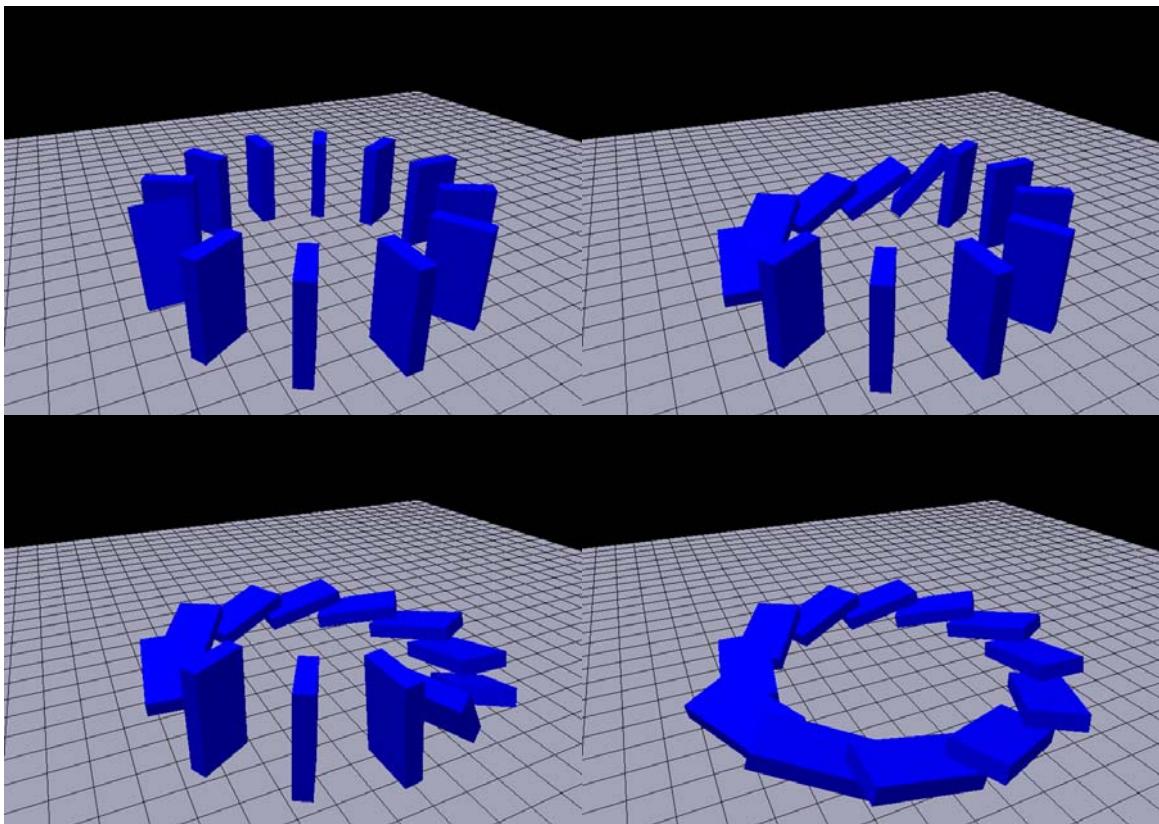
Slika 30. Biljarske kugle

Na slici 30 prikazan je rad propagirajućeg modela. Biljarska kugla udari jedan kraj kugli poslaganih u red i impuls propagira na kraj gdje se zadnja kugla počne kotrljati i utjecajem trenja kotrljanja kugla se zaustavi. Ukoliko se koristi samo statičko trenje kugla se neće nikad zaustaviti.

Također ukoliko nije korišten dovoljan broj koraka rješavanja sudara kugle se neće ispravno odbiti. Na sljedećoj slici je prikazan rezultat simulacije sa 4 koraka rješavanja sudara.



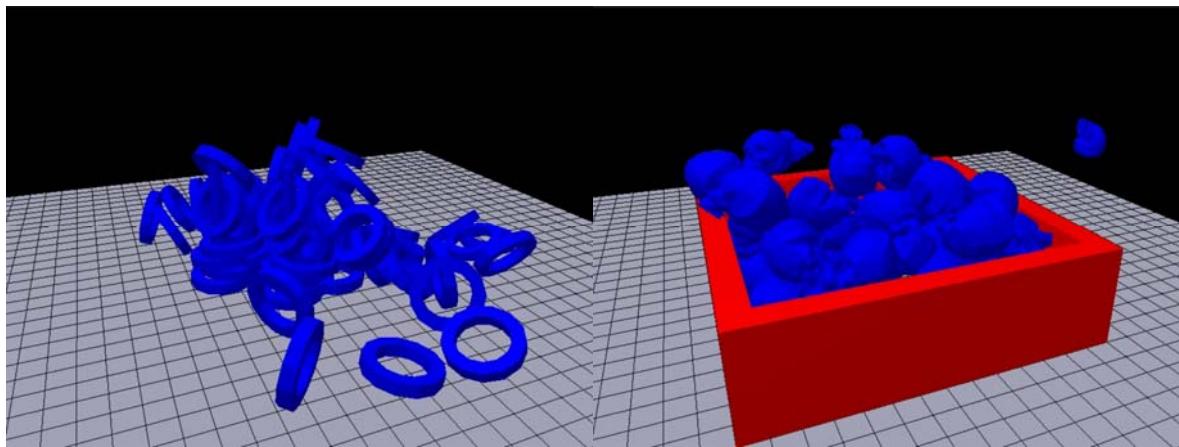
Slika 31. Nedovoljan broj koraka simulacije



*Slika 32. Domino*

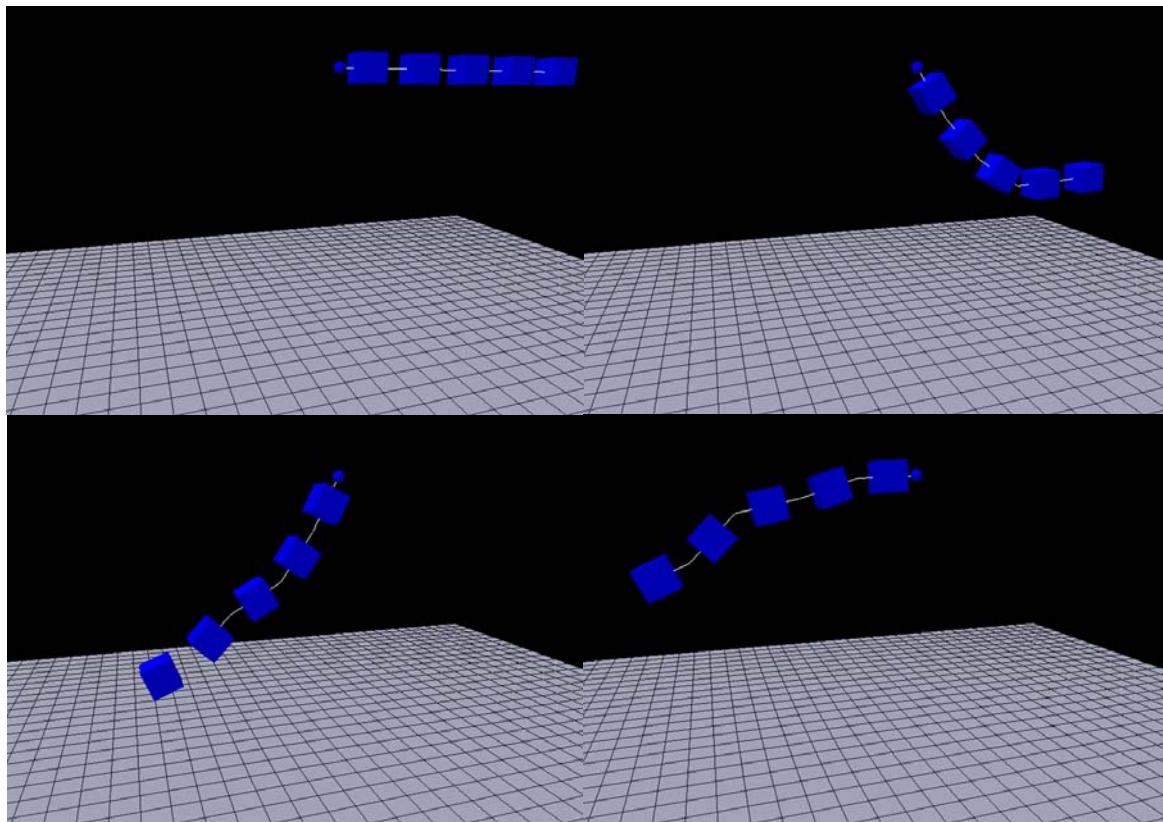
Slika 32 prikazuje ispravno rušenje domino pločica. Ukoliko bi se kontaktni graf konstruirao jednostavnim pretraživanjem u dubinu bez rješavanja cikličke veze kod konstruiranja kontaktnog grafa nailaskom na cikličku vezu program bi zablokirao jer bi se beskonačno vrtio u cikličkoj vezi.

Vremena simulacije prethodnih primjera bila su zanemariva. Na slici 33 su prikazani primjeri simulacije složenih konkavnih tijela. Kod simulacije 60 prstena od po 216 trokuta koji padaju na 3 stupca od 84 trokuta za simulaciju od 12 sekundi bilo je potrebno 9 minuta, tj. 0.5 sekunde po simulacijskom koraku u trenutku kada su tijela naslagana jedno na drugo. Kod simulacije 50 naslaganih konkavnih lubanja gdje svaka lubanja ima 2970 trokuta za simulaciju od 7 sekunde bilo je potrebno 11 minuta, tj. 1 sekunda po simulacijskom koraku u trenutku kada su tijela naslagana jedno na drugo. Testno računalo je bilo Athlon64 2.4GHz, 1GB RAM. U [1] su simulirali 500 prstena te 500 kostiju nešto veće složenosti (lubanja je imala 3520 trokuta) i vremena jednog simulacijskog koraka su bila 3 te 5 minuta.

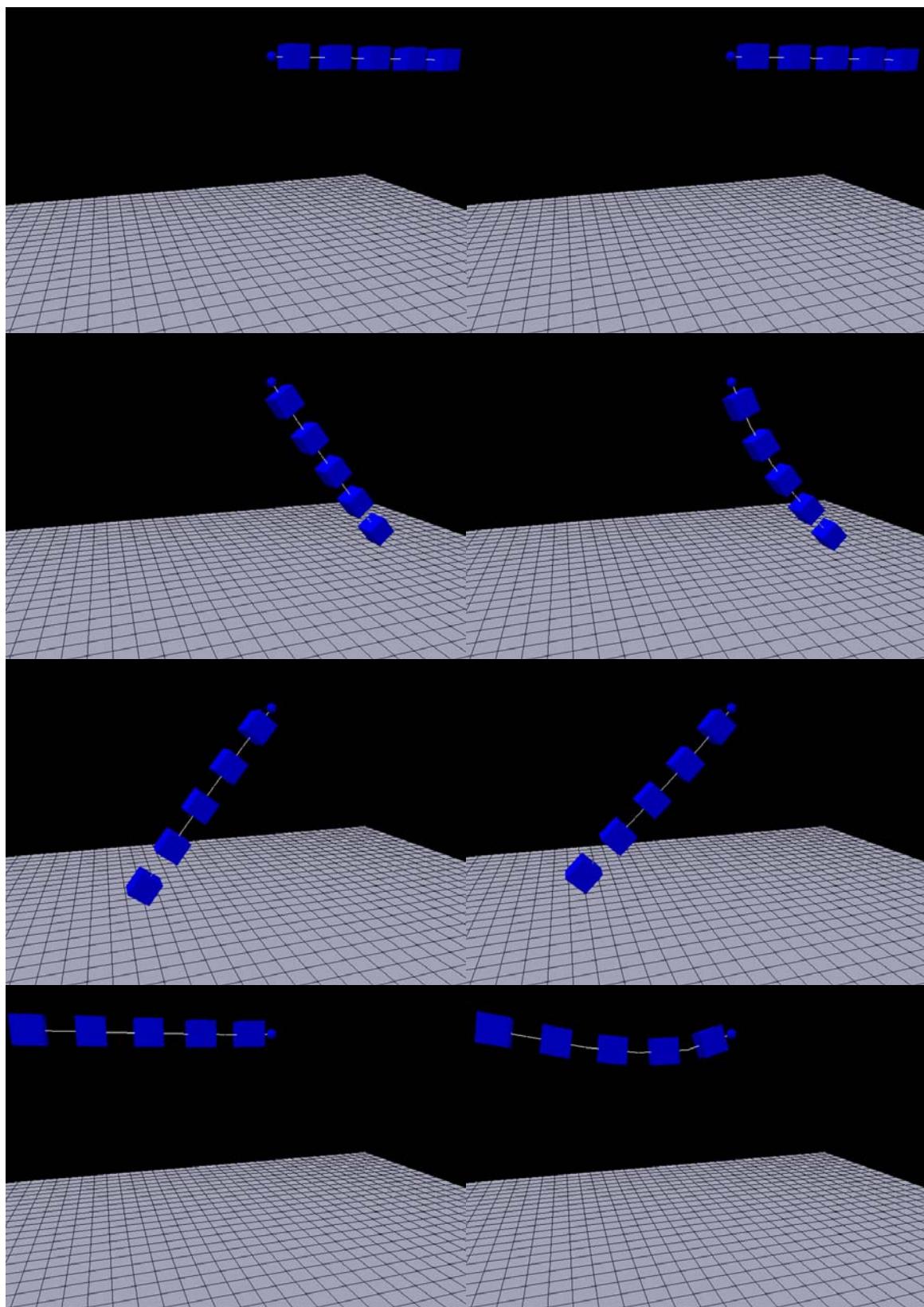


Slika 33. Slaganje prstena na stupce te slaganje konkavnih lubanja

Sljedeći primjeri prikazuju simulaciju zglobova. Na slici 34 je njihalo kod kojeg su sve kocke spojene pomoću translacijskih ograničenja sa ograničena sva 3 translacijska stupnja slobode.



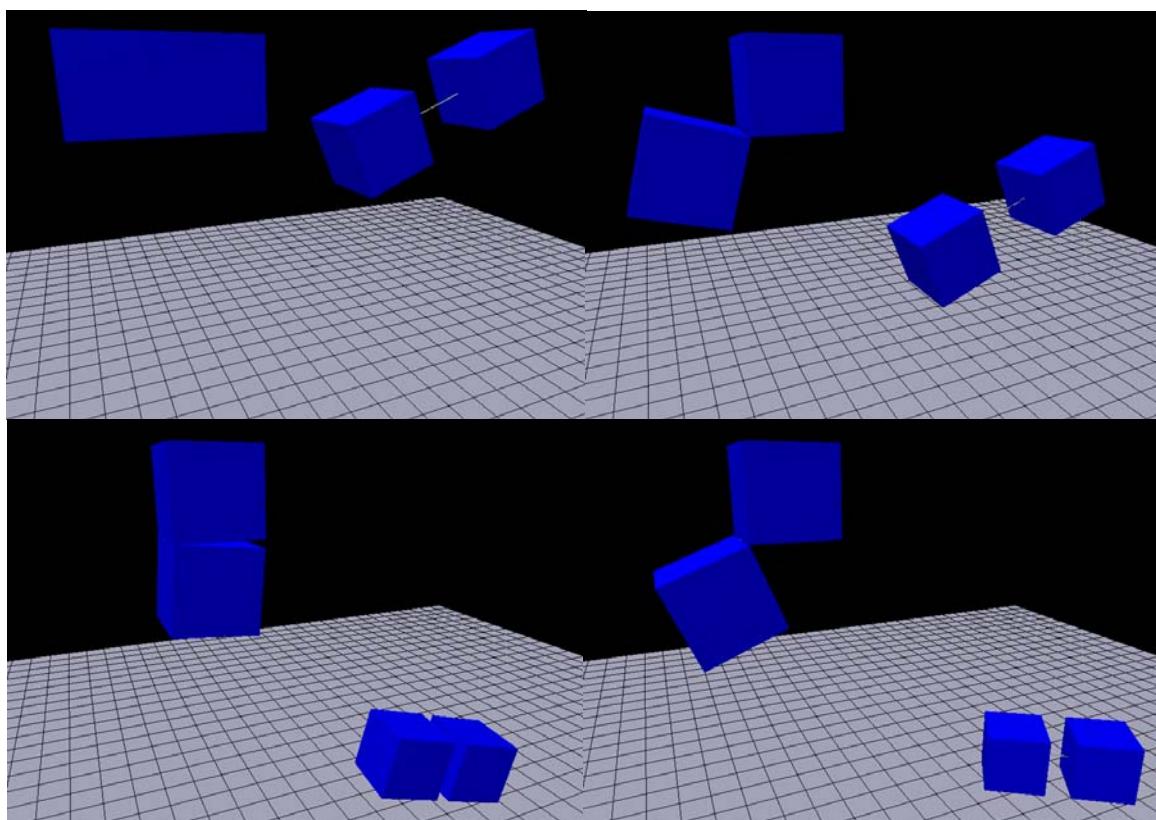
Slika 34. Višestruko njihalo sastavljenog pomoću translacijskih ograničenja



Slika 35. Njihalo sastavljeno pomoću translacijskih i rotacijskih ograničenja

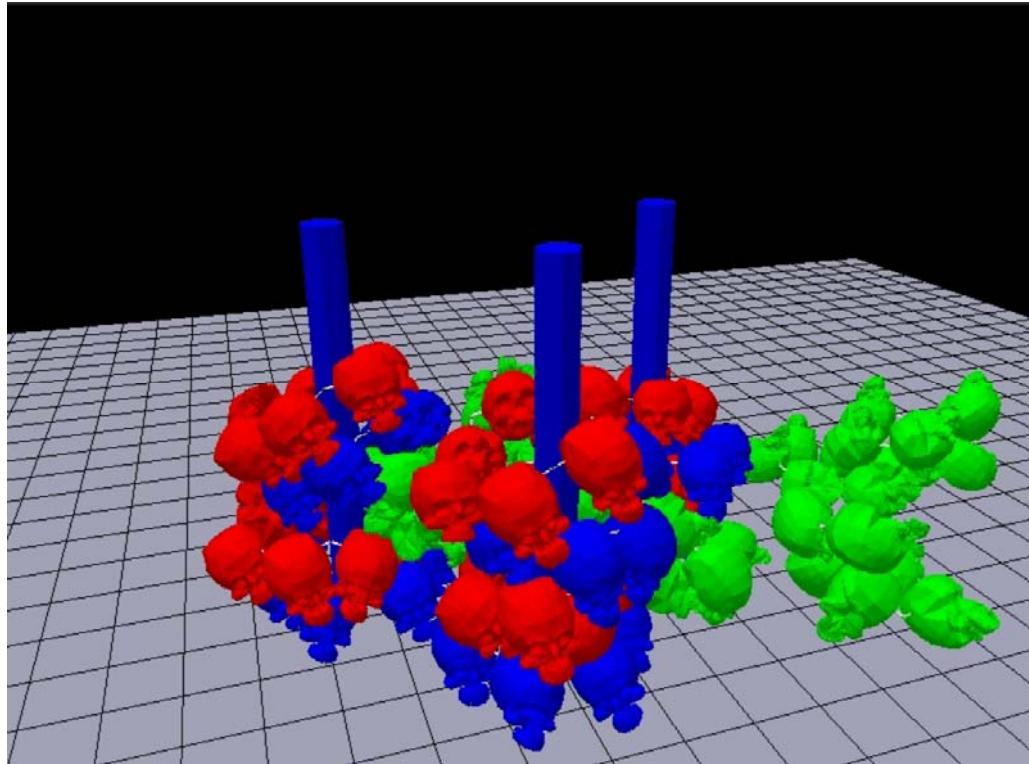
Na slici 35 je prikazano njihalo kod kojeg su sve kocke spojene pomoću translacijskih i rotacijskih ograničenja tako da su ograničeni svi stupnjevi slobode. Na lijevom primjeru je prikazan ispravan rad njihala dok je u desnom primjeru prikazano njihalo ukoliko se koristi samo jedna iteracija rješavanja zglobova. Dolazi do pogreške jer se rješavanjem jednog zgloba javlja greška u drugom zglobu.

Slika 36 prikazuje zglobove s dva ograničena stupnja slobode. Lijeve dvije kocke su spojene pomoću jednog translacijskog zgloba s 3 ograničena stupnja i rotacijskog zgloba s 2 ograničena stupnja. Kako je desna kocka nepomična lijeva kocka pod utjecajem gravitacije rotira samo oko jedne osi. Desne dvije kocke su spojene s rotacijskim zglobom s 3 ograničena stupnja slobode i translacijskim zglobom s 2 ograničena stupnja slobode tako da se kocke mogu samo translatirati po jednoj osi u ovisnosti jedna prema drugoj.



Slika 36. Zglobovi s ograničena dva stupnja slobode

Na slici 37 je prikazano kombiniranje zglobova sa simulacijom sudara i kontakata. Simulirano je 108 lubanja (od kojih se svaka sastoji od 2970 trokuta) kako padaju na stupce s time da je po 6 lubanja povezano zglobovima u krug te se vidi kako zglobovi ostaju nerazdvojeni tijekom simulacije. Simulacija od 6 sekundi je trajala 26 minuta. Testno računalo je Athlon64 2.4GHz, 1GB RAM.



Slika 37. Slaganje konkavnih lubanja povezanih zglobovima.

## 13. Zaključak

U ovom radu predstavljena je jedna od mnogih metoda simulacije krutih tijela. Veliki dio rada se može iskoristiti i za simulaciju tijela pomoću neke druge metode, pogotovo dio o gibanju tijela te računanju svojstva tijela. Prednost ove metode je u dvostrukom prikazu geometrije tijela tako da je jednostavno za odrediti sudsar vrha s tijelom te normalu sudsara. Upotrebo novog redoslijeda integracije nije potrebno koristiti otprilike određenu granicu brzine za razlikovanje sudsara od kontakata te metoda propagiranja šoka omogućuje bržu a ujedno i uvjerljivu simulaciju naslaganih tijela.

Nedostatak je u tome što se preklapanje između objekata još uvijek može dogoditi zbog toga što su neki objekti stavljeni u istu razinu prilikom konstrukcije grafa. Osim toga metoda je i prilično spora i to najviše zbog toga što se vrši preveliki broj detekcija sudsara. Ukoliko se želi napraviti što brža simulacija ne bi se smjelo koristiti više od jedne detekcije sudsara po koraku simulacije. Neke od bržih metoda su [8] i [14]. Metoda [14] je slična opisanoj metodi s time da se ne vrši ponovna detekcija sudsara prilikom rješavanja svake točke sudsara. Dok je kod metode [8] određivanje reakcije na sudsar veoma pojednostavljeno (koristi se sila opruge) tako da je moguće paralelno rješavanje sudsara a time ujedno i implementacija na grafičkoj kartici te tada i simulacija velikog broja konkavnih tijela radi u stvarnom vremenu.

## 14. Popis slika

<i>Slika 1. 2D polje udaljenosti za simbol R [2]</i> .....	2
<i>Slika 2. Interpolacija udaljenosti</i> .....	3
<i>Slika 3. Skalarno polje i njemu odgovarajuće vektorsko polje</i> .....	4
<i>Slika 4. Interpolacija normale</i> .....	4
<i>Slika 5. 2D prikaz određivanja ćelije u kojoj se nalazi točka te njezine koordinate u koordinatnom sustavu ćelije</i> .....	5
<i>Slika 6. Sudar između dva brida</i> .....	6
<i>Slika 7. Skalarni umnožak <math>r = p - x</math> s jednom normalom daje pozitivan rezultat a s drugom negativan</i> .....	7
<i>Slika 8. Sedam različitih Voronoi-evih regija ravnine u kojoj se nalazi trokut</i> .....	8
<i>Slika 9. Određivanje udaljenosti točke od trokuta</i> .....	9
<i>Slika 10. Određivanje položaja točke u odnosu na simetrale kutova</i> .....	10
<i>Slika 11. Pozicija i orijentacija tijela</i> .....	12
<i>Slika 12. Linearna i kutna brzina tijela</i> .....	13
<i>Slika 13. Brzina točke na tijelu</i> .....	15
<i>Slika 14. Gibanje kocke na kosini pri standardnom koraku integracije</i> .....	20
<i>Slika 15. Gibanje kocke na kosini pri novom koraku integracije</i> .....	21
<i>Slika 16. Propagirajući model</i> .....	30
<i>Slika 17. Konstrukcija kontaktnog grafa</i> .....	31
<i>Slika 18. Rješavanje cikličke veze u kontaktnom grafu</i> .....	32
<i>Slika 19. Princip rada metode propagiranja šoka</i> .....	34
<i>Slika 20. Propagirajući model omogućava da teža kocka gurne dasku prema dolje i da zatim daska gurne lijevu kocku prema gore</i> .....	35
<i>Slika 21. Stupnjevi slobode translacijskih zglobova</i> .....	36
<i>Slika 22. Sferni zglob</i> .....	37

---

<i>Slika 23. Stupnjevi slobode rotacijskih zglobova .....</i>	41
<i>Slika 24. Stabilizacija unaprijed .....</i>	44
<i>Slika 25. Program za simulaciju .....</i>	45
<i>Slika 26. Postavljanje parametra simulacije .....</i>	46
<i>Slika 27. Savršeno elastično tijelo na kosini.....</i>	47
<i>Slika 28. Klackalica .....</i>	48
<i>Slika 29. Prevrtanje kocki.....</i>	49
<i>Slika 30. Biljarske kugle .....</i>	50
<i>Slika 31. Nedovoljan broj koraka simulacije .....</i>	50
<i>Slika 32. Domino .....</i>	51
<i>Slika 33. Slaganje prstena na stupce te slaganje konkavnih lubanja .....</i>	52
<i>Slika 34. Višestruko njihalo sastavljenog pomoću translacijskih ograničenja .....</i>	52
<i>Slika 35. Njihalo sastavljenog pomoću translacijskih i rotacijskih ograničenja ...</i>	53
<i>Slika 36. Zglobovi s ograničena dva stupnja slobode.....</i>	54
<i>Slika 37. Slaganje konkavnih lubanja povezanih zglobovima .....</i>	55

## 15. Literatura

1. Eran Guendelman, Robert Bridson and Ronald Fedkiw. Nonconvex rigid bodies with stacking. ACM Transactions on Graphics, (SIGGRAPH 03), pp. 871-878, Lipanj 2003.
2. Sarah F. Frisken, Ronald N. Perry, Alyn P. Rockwood, and Thouis R. Jones. Adaptively Sampled Distance Fields: a general representation of shape for computer graphics. In Proc. SIGGRAPH 2000, pp. 249-254, 2000.
3. J. Andreas Bærentzen and Henrik Aanæs. Generating Signed Distance Fields From Triangle Meshes. IMM-TECHICAL REPORT, 2002.
4. Mark W. Jones. 3D Distance from a Point to a Triangle. Technical Report CSR-5-95, Department of Computer Science, University of Wales Swansea, Veljača 1995.
5. David Baraff. Physically Based Modeling: Rigid Body Simulation. Online SIGGRAPH 2001 Course Notes, 2001.
6. Rick Parent. Computer Animation - Algorithms and Techniques, 2002.
7. Brian Mitrich. Fast and accurate computation of polyhedral mass properties. J. Graph. Tools 1, 2, pp. 31-50, 1996.
8. Masayuki Tanaka, Mikio Sakai and Seiichi Koshizuka. Particle-based Rigid Body Simulation and Coupling with Fluid Simulation. Transactions of JSCES, 2007.
9. Brian Mitrich. Impulse-based Dynamic Simulation of Rigid Body Systems, doktorska disertacija, University of California, Berkly, Jesen 1996.
10. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest. Introduction to algorithms. The MIT Press, 1990.
11. Robert Sedgewick. Algorithms in Java, Third Edition, Part 5: Graph Algorithms. Addison Wesley. 2003.
12. Rachel Weinstein, Joseph Teran, and Ron Fedkiw. Dynamic Simulation of Articulated Rigid Bodies with Contact and Collision. IEEE Transactions on visualization and computer graphics. 2005.
13. Jan Bender and Alfred A. Schmitt. Fast Dynamic Simulation of Multy-Body Systems Using Impulses. Institut für Betriebs- und Dialogsysteme, Universität Karlsruhe, Germany. 2006.
14. Jan Bender and Alfred A. Schmitt. Constraint-based collision and contact handling using impulses. Universität Karlsruhe, Germany. 2006.

## Sažetak

### (Fizikalna simulacija sudara konkavnih krutih tijela)

U ovom radu je opisana fizikalna simulacija krutih tijela. Objasnjenje su osnove gibanja krutih tijela te računanja fizikalnih svojstva tijela. Sudari između konkavnih tijela se detektiraju korištenjem polja udaljenosti dok se za reakciju na sudar koriste impulsi. Prilikom reakcije na sudar uzeto je u obzir statičko i dinamičko trenje te trenje prilikom kotrljanja i vrtanje. Korišten je novi redoslijed integracije kojim se eliminira potreba za korištenje praga brzine za razlikovanje između sudara i kontakta. Velika važnost je dana uvjerljivom simuliranju velikog broja naslaganih tijela a to je postignuto korištenjem metode propagiranja šoka. Također je opisana simulacija zglobova pomoću impulsa i njihovo integriranje sa simulacijom krutih tijela.

**Ključne riječi:** konkavna kruta tijela, sudar, kontakt, polja udaljenosti, trenje, naslagana tijela, zglobovi

# Abstract

## (Physical simulation of collisions between concave rigid bodies)

This paper describes physical simulation of rigid bodies. It explains basics of rigid body motion and computation of rigid body physical properties. Collisions between concave bodies are detected using signed distance fields while collision reaction is performed using impulses. Collision reaction is using static, dynamic, rolling and spinning friction. New time stepping scheme is used that eliminates the need for velocity threshold to differentiate between collisions and contacts. Great importance is given to efficient simulation of large number of stacked bodies and that is accomplished using shock propagation algorithm. It also describes simulation of joints using impulses and their integration with rigid body simulation.

**Keywords:** concave rigid bodies, collision, contact, signed distance fields, friction, stacking, joints