

UNIwersYTET WROCLAWSKI
WYDZIAŁ FIZYKI I ASTRONOMII
INSTYTUT FIZYKI TEORETYCZNEJ



Praca magisterska

MODELOWANIE MATERII GRANULOWANEJ Z TARCIEM

Rafał Korzyński

Opiekun:
dr Maciej Matyka

Wrocław, 2011

Streszczenie

Głównym celem pracy było zaimplementowanie symulacji dla modelu materii granulowanej z uwzględnieniem sił tarcia, zarówno statycznego, jak i dynamicznego. Celem szczegółowym było zbadanie zależności kąta stabilności krytycznej dla sterty usypanej z niejednorodnych ziaren o kształcie sferycznym.

Otrzymane wyniki konfrontuję z wynikami z publikacji J. Lee i H. J. Herrmann, *Angle of repose and angle of marginal stability: molecular dynamics of granular particles*, J. Phys. A: Math. Gen. 26(2), 1993.

Dodatkowo opisuję implementację metody całkowania Geara użytej w przeprowadzonej symulacji.

Abstract

Title: Modeling granular matter with friction.

The aim of this study is to implement a simulation of granular material model with static and dynamic friction. I focus on the calculation of an angle of repose for a two dimensional heap built of circular granules.

Then I have compared the results to the paper: J. Lee and H. J. Herrmann, *Angle of repose and angle of marginal stability: molecular dynamics of granular particles*, J. Phys. A: Math. Gen. 26(2), 1993.

In addition, I have described an implementation of the Gear integration method used in the simulation.

Spis treści

| | |
|---|-----------|
| Spis treści | i |
| 1 Wstęp | 3 |
| 1.1 Materia granulowana | 3 |
| 1.2 Tarcie | 4 |
| 1.3 Plan pracy | 5 |
| 2 Model materii granulowanej z tarciem | 7 |
| 2.1 Model Hertza | 7 |
| 2.2 Definicja modelu z tarciem | 8 |
| 3 Opis eksperymentu | 13 |
| 3.1 Omówienie symulacji | 13 |
| 3.2 Przeprowadzanie pomiarów | 13 |
| 3.2.1 Wypełnianie pojemnika | 13 |
| 3.2.2 Relaksacja | 14 |
| 3.2.3 Pomiar kąta | 15 |
| 4 Wyniki | 17 |
| 4.1 Omówienie wyników | 17 |
| 5 Podsumowanie | 25 |
| A Metoda całkująca Geara | 29 |
| Bibliografia | 37 |
| Spis rysunków | 39 |

Podziękowania

Chciałbym podziękować swojemu prowadzącemu za pomoc w przezwyciężaniu napotykanym trudności oraz nabyciu cennego doświadczenia.

Rozdział 1

Wstęp

1.1 Materia granulowana

Materia granulowana jest zbiorem dużej ilości makroskopowych drobin (ziaren), których oddziaływania zachodzą podczas kontaktu oraz są dysypatywne. W tego typu materii przeważają oddziaływania między drobinami i są one na tyle duże, iż wpływ oddziaływań związanych z ich energią termiczną można całkowicie pominąć.

Pojęcie materii granulowanej obejmuje bardzo szeroki zakres występujących zjawisk fizycznych. Natura tych zjawisk może wywodzić się z zachowań charakterystycznych dla ciała stałego, cieczy lub gazu. Występowanie tego typu materii jest bardzo powszechne od osuwisk skalnych i wydm piaskowych spotykanych na ziemi, aż po materię występującą w kosmosie (np. pierścienie Saturna). Najbardziej znana grupa zjawisk to formowanie się wzorców w materii [1]. Możliwe są fluktuacyjne zmiany gęstości przepływającej materii oraz powstawanie chwilowych zatorów, zaobserwowane przez G.W. Baxtera [2] w leju z przesypującym się piaskiem. Zjawiska te powstają dzięki oddziaływaniom dysypatywnym. Dobre odzwierciedlenie takich zachowań oddaje symulacja G.H. Ristowa i H.J. Herrmanna [3]. Kolejnym często spotykanym efektem jest *segregacja wielkościowa* powstająca w wyniku drgań wibrującego układu. Jako przykład można podać tak zwany *efekt brazylijskiego orzecha* [4]. W wyniku potrząsania pojemnika wypełnionego orzechami różnych rozmiarów, po pewnym czasie największe orzechy znajdują się na górze pojemnika a najmniejsze na spodzie. W układach gęsto upakowanych można spotkać się z dużymi naprężeniami deformujących się drobin w wyniku czego obserwuje się powstawanie tak zwanych *łańcuchów sił*. Można je zaobserwować dzięki foto-elastycznym materiałom [5]. Bogactwo występujących zjawisk jest argumentem zachęcającym, aby zająć się badaniem tego typu materii oraz symulacjami zjawisk tak często nas otaczających.

1.2 Tarcie

Tarcie jest złożonym zjawiskiem mechanicznym. Pomimo tego, iż jest ono łatwo obserwowalne i możliwe jest jego ilościowe zmierzenie, to natura tego zjawiska jest ściśle związana ze strukturą atomową powierzchni badanego materiału. Współczynnik tarcia zewnętrznego (zachodzącego w warstwie powierzchniowej) jest trudno mierzalny, gdyż powtarzanie pomiarów nie daje potwierdzenia stałej wartości współczynnika. W różnych pomiarach dla tych samych materiałów otrzymuje się różne jego wartości. Związane jest to z wieloma czynnikami nie tylko pochodzenia materiałowego, ale także z metodą wytworzenia przedmiotu, wilgotnością i zanieczyszczeniami elementów trących. W tej pracy zajmować się będą jedynie tarcie suche, tzn. oddziaływaniem przeciwdziałającym siłom powodującym ruch ciała stałego po innym ciele stałym. Bardzo ważne jest to, iż między nimi nie występują żadne inne ciała. Jak dotąd nie stworzono jeszcze jednej jednolitej teorii tarcia suchego, która opisywałaby jednoznacznie wartość współczynnika tarcia otrzymywanego w dowolnych warunkach badań [6].

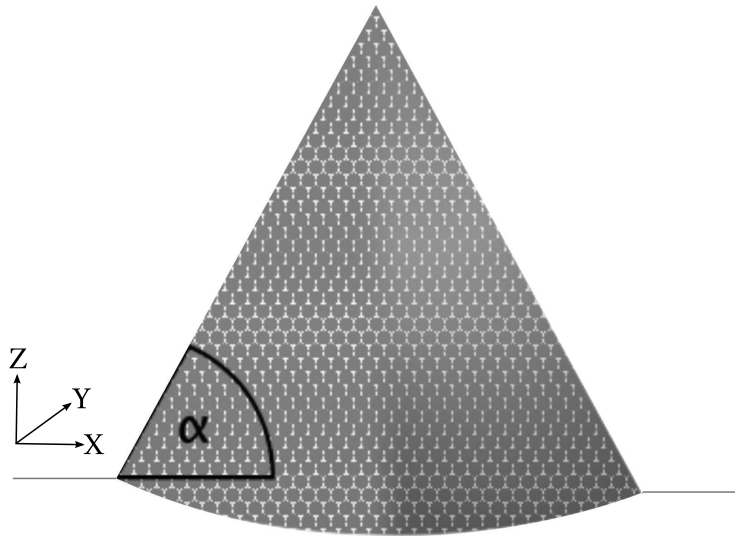
W fizyce można jeszcze spotkać się z pojęciem *tarcia wewnętrznego* materiału. Jest to siła oporu ruchu powstająca w wewnętrznej strukturze ciała stałego, podczas gdy ulega ono deformacji plastycznej. Siły powstające w wyniku odkształcania się ciała uwzględnione są w modelu Hertza, który zostanie omówiony w następnym rozdziale.

Z punktu widzenia kinetyki ruchu wyróżniamy *tarcie statyczne* oraz *tarcie kinetyczne*. Tarcie statyczne jest taką siłą, która nie pozwala na ruch ciała i dopiero przekroczenie pewnej wartości krytycznej tej siły umożliwi wprawienie go w ruch. Zależy ona od indywidualnych własności materiałowych stykających się ciał oraz od stanu ich powierzchni kontaktu. Coulomb podał warunek na wartość tarcia statycznego, mówiąc iż wartość siły tarcia statycznego F_{stat} nie może być większa niż współczynnik tegoż tarcia μ_{stat} razy siła normalna F_n działająca na to ciało:

$$F_{stat} \leq \mu_{stat} F_n. \quad (1.1)$$

Głównym celem tej pracy jest badanie współczynnika zwanego *krytycznym kątem stabilności* (ang. critical angle of repose). Jest to kąt nachylenia zbocza usypanej sterty materii granulowanej o kształcie stożka, widoczny na rysunku 1.1. Nazywa się go krytycznym, gdyż dosypywanie większej liczby materiału granulowanego powoduje powstawanie lawin i rozsypywanie się struktury kopca, ale wartość nachylenia zawsze fluktuuje wokół pewnej stałej wartości, zwanej *kątem stabilności krytycznej*. W późniejszych rozdziałach najczęściej nazywany jest on *kątem nachylenia zbocza*.

Z przeprowadzanych badań doświadczalnych wiemy, że zależność kąta nachylenia zbocza jest ściśle powiązana z wartością wewnętrznego współczynnika tarcia statycznego usypanej sterty. Jest to jedno ze zjawisk charakteryzujących materię



Rysunek 1.1: Schemat usypanego stożka oraz oznaczenie kąta stabilności krytycznej.

granulowaną, a zależność je opisująca jest postaci

$$\operatorname{tg}(\theta) = \mu_{stat}, \quad (1.2)$$

gdzie θ to kąt nachylenia zbocza, a μ_{stat} to współczynnik tarcia statycznego [7].

1.3 Plan pracy

W pierwszym rozdziale przedstawiłem definicję materiału granulowanego oraz omówiłem najważniejsze zjawiska fizyczne występujące w świecie materii granulowanej. W rozdziale drugim opiszę model teoretyczny materii granulowanej z uwzględnieniem tarcia statycznego. Natomiast w rozdziałach trzecim i czwartym przedstawię opis wykonanego eksperymentu. Rozdział trzeci zawierać będzie szczegóły dotyczące wykonania symulacji komputerowej, a w rozdziale czwartym przeprowadzę analizę otrzymanych wyników.

Rozdział 2

Model materii granulowanej z tarciem

2.1 Model Hertza

Jest to model opisujący sytuację występującą podczas ruchu obiektu po powierzchni innego obiektu, gdzie mamy do czynienia z oporem wywołanym przez odkształcenia sprężyste. Opisuje on oddziaływania między elastycznymi kulami naciskającymi na siebie. Hertz zadał sobie pytanie: jak mocno trzeba ścisnąć dwie kule, aby utworzyć między nimi powierzchnię kontaktu o promieniu r [8]? Zaistniała sytuacja przedstawiona jest na rysunku 2.1. Chcemy wyznaczyć promień kontaktu. Na podstawie twierdzenia Pitagorasa otrzymujemy:

$$R_2^2 = r^2 + (R_2 - d_2)^2 = r^2 + R_2^2 - 2R_2d_2 + d_2^2 \approx r^2 + R_2^2 - 2R_2d_2. \quad (2.1)$$

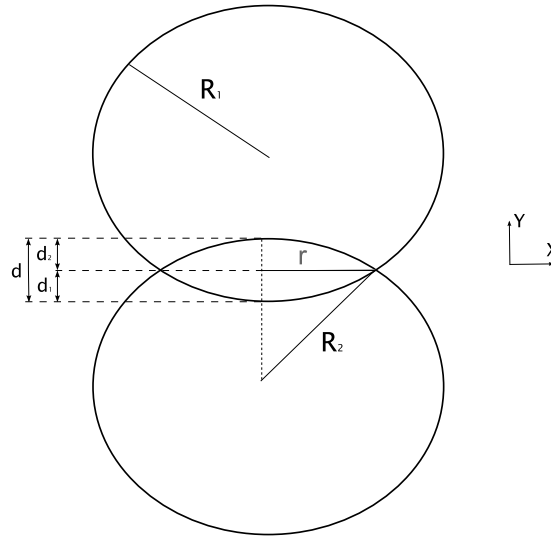
Dla $d_2 \ll R_2$ otrzymujemy dobre przybliżenie wyniku. Dzięki temu otrzymujemy wartość odkształcenia d_2 w kierunku promienia R_2 .

$$d_2 \approx \frac{r^2}{2R_2}, \quad d_1 \approx \frac{r^2}{2R_1} \quad (2.2)$$

W analogiczny sposób obliczamy d_1 , a więc suma odkształceń dla obu kul opisana jest wzorem

$$d(r) = d_1 + d_2 = \frac{r^2}{2R}, \quad (2.3)$$

gdzie promień zastępczy $R^{-1} = R_1^{-1} + R_2^{-1}$. Pomijam dalszy tok rozważania prowadzący do końcowego wyniku, ponieważ nie będzie ono nam potrzebne. Dokładniejsze omówienie tematu można znaleźć w artykule P.Borysa [8] oraz monografii Johansona [9]. Najważniejszym dla nas wynikiem z teorii Hertza jest wzór na siłę



Rysunek 2.1: Dwie elastyczne kule naciskające na siebie, tworzące powierzchnię kontaktu w płaszczyźnie XZ.

normalną działającą na ściskane kule

$$F_n = \frac{4Ea^3}{3R}, \quad (2.4)$$

gdzie E ($\frac{N}{m^2}$) to moduł Younga i $a \equiv r$. Dla pewnego *nacisku wywieranego na kule* otrzymujemy powierzchnię kontaktu o promieniu a , tak więc na podstawie wzoru (2.3) otrzymujemy $a = \sqrt{2Rd}$. Wstawiamy a do wzoru na siłę F_n i otrzymujemy

$$F_n = \frac{8\sqrt{2R}}{3} E d^{3/2}. \quad (2.5)$$

Rysunek 2.1 przedstawia kule, które nachodzą na siebie. W rzeczywistości ściśnięcie kul do takiej odległości spowodowałoby odkształcenie się powierzchni kuli w punkcie kontaktu. Natomiast takie przedstawienie dobrze obrazuje sytuację zachodzącą podczas symulacji komputerowej. Odległość d można także otrzymać z różnicy odległości między środkami kul.

$$d = R_1 + R_2 - |\vec{r}_1 - \vec{r}_2| \quad (2.6)$$

Wstawiając d do wzoru na siłę, otrzymujemy ostateczną postać siły normalnej działającej na kule podczas kontaktu, wywołaną przez odkształcenia sprężyste [10].

2.2 Definicja modelu z tarciem

Rzeczywiste oddziaływanie drobin materii granulowanej np. piasku, jest bardzo skomplikowane i trudno aby w jednym modelu uwzględnić wszystkie opisujące

je własności oraz parametry. W mojej pracy wybrałem model zaproponowany przez J.Lee i H.J.Herrmanna [11] uwzględniający siły tarcia statycznego. Stworzony przez nich model zawiera oddziaływania odpychające oraz rozpraszające. Nie uwzględnia on natomiast wpływu ruchu obrotowego. Drobiny modelowane są przez sferyczne cząstki, które mogą ze sobą oddziaływać tylko wtedy gdy są ze sobą w kontakcie. Dzięki temu, że pozwalamy na niewielkie nachodzenie cząstek na siebie, model ten może wykorzystać definicję siły normalnej zaproponowanej przez Hertza. Położenie środka cząstki i (j) opisane jest za pomocą \vec{r}_i (\vec{r}_j), a odległość między nimi to $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$. Wielkość cząstki i definiowana jest przez jej promień R_i . Wektor \vec{n} jest wektorem jednostkowym o kierunku zgodnym z \vec{r}_{ij} . Natomiast wektor \vec{s} jest wektorem prostopadłym do \vec{n} i odwróconym względem niego o 90 stopni zgodnie ze wskazówkami zegara. Można to zapisać w postaci $\vec{s} = (n_y, -n_x)$. Definiujemy także prędkość względną $\vec{v} = \vec{v}_i - \vec{v}_j$.

Siła działająca na cząstkę i , zapisana jest w postaci

$$\vec{F}_{j \rightarrow i} = F_n \vec{n} + F_s \vec{s}, \quad (2.7)$$

gdzie F_n jest siłą normalną złożoną z dwóch członów:

$$F_n = k_n (R_i + R_j - |\vec{r}_{ij}|)^{3/2} - \gamma_n m_e (\vec{v} \cdot \vec{n}). \quad (2.8)$$

Pierwszy człon odpowiada za oddziaływanie odpychające podczas elastycznej deformacji, gdzie k_n (kg/s²) jest współczynnikiem sprężystości materiału. Drugi człon jest zależny od prędkości i odpowiedzialny za rozpraszanie energii kinetycznej. Wielkość γ_n (s⁻¹) pełni funkcję współczynnika rozpraszania, a m_e (kg) oznacza masę efektywną zapisaną wzorem:

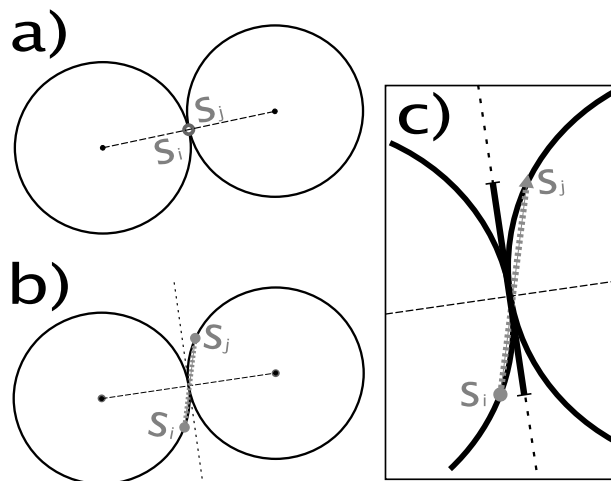
$$m_e = m_i m_j / (m_i + m_j). \quad (2.9)$$

Natomiast F_s to siła ścinająca, działająca w kierunku stycznym do powierzchni kontaktu:

$$F_s = -\gamma_s m_e (\vec{v} \cdot \vec{s}) - \text{sign}(\delta s) \min(k_s \delta s, \mu F_n). \quad (2.10)$$

Pierwszy jej człon zależy od prędkości i jest analogiczny do drugiego w równaniu (2.8). Człon drugi odpowiedzialny jest za symulację tarcia statycznego. Funkcja \min zapewnia spełnienie warunku $F_{stat} \leq \mu_{stat} F_n$ na tarcie statyczne, którego wartość nie może być większa niż pewna siła, w tym przypadku μF_n .

Cała idea uwzględnienia tarcia statycznego w oddziaływaniu, polega na umieszczeniu "wirtualnych" sprężyn pomiędzy cząstkami, które są w kontakcie (kolizji). Siła oddziaływania takiej sprężyny opisana jest wzorem $-k_s \delta s$, gdzie k_s jest współczynnikiem oddziaływania sprężystego dwóch kolidujących cząstek. Czynnikiem δs jest całkowitym przemieszczeniem ścinającym. Najważniejszą kwestią otrzymania wartości tej siły jest wyliczenie δs . Podczas wykrycia początkowej fazy kolizji należy zapamiętać położenie punktu kontaktu w układach odniesienia cząstek i oraz j . Początkowa faza kolizji oznacza pierwszy krok symulacji, w którym



Rysunek 2.2: Położenie wirtualnej sprężyny: a) Początkowa faza kolizji kul, pomiędzy nimi pojawia się punkt kontaktu. b) Położenie punktów kontaktu po kilkunastu krokach symulacji. c) Powiększenie widoku na sytuację b. Widzimy wektor \vec{s}_{ij} z okrągłym zakończeniem oraz rzut wektora \vec{s}_{ij} na płaszczyznę styczną do powierzchni kontaktu.

wykryliśmy kolizję między konkretną parą cząstek. Zdefiniujemy punkty kontaktu jako \vec{s}_i i \vec{s}_j . Tak zdefiniowane punkty są położeniami we współrzędnych globalnych. Wyliczane są one z zapamiętanych punktów kontaktu, będących we współrzędnych danej cząstki. Wektor $\vec{s}_{ij} = \vec{s}_i - \vec{s}_j$, wodzący między tymi punktami rzutowany na kierunek ścinania \vec{s} , da nam szukaną wartość δs . W każdym kolejnym kroku, w którym zachodzi ta sama kolizja, wyliczamy odległość s_{ij} . Gdy w kolejnym kroku symulacji kolizja dla danej pary już nie zachodzi, należy usunąć "wirtualne" sprężyny. Oznacza to tyle, iż zapamiętane przez nas punkty względne są już danymi nieaktualnymi i trzeba będzie wyliczyć je od nowa podczas następnej kolizji. Powyższą sytuację zobrazowałem na rysunku 2.2.

Dobrym pomysłem na techniczne rozwiązanie przechowywania i odświeżania punktów kontaktów jest lista list. Przy czym pierwsza indeksowana jest po wszystkich kulach. Natomiast druga zawiera informację (listing 2.1) o kolidującej z nią kuli, punkt kontaktu oraz flagę mówiącą o tym czy kolizja zachodziła w poprzednim kroku.

Listing 2.1: Przykład struktury danych dla wirtualnej sprężyny w języku C++.

```
struct vSpring {
    uint   particle2_id;
    bool   previousState;
    Vec2   anchorPoint;
};
```

Modelowanie takich struktur jak ścianki odbywa się za pomocą kul statycznych. Podczas rozwiązywania równań ruchu nie są wykonywane dla nich żadne obliczenia. Podczas kolizji zwykłej kuli z kulą statyczną oddziaływanie jest identyczne, jak oddziaływanie zwykłych kul. Jedyną różnicą jaką zastosowałem w symulacji dla kul statycznych, było ustawianie wartości dwa razy większej dla stałej sprężystości k_n we wzorze (2.8) na siłę F_n . Powodem takiego działania było zapobiegnięcie przelatywaniu cząstek przez ściany. Oprócz sił działających na cząstki podczas kolizji, istnieje jeszcze siła wynikająca z oddziaływania pola grawitacyjnego równa $m_i \vec{g}$.

Symulacja została wykonana techniką dynamiki molekularnej. Rozwiązywanie równań ruchu wykonane było przy użyciu metody *predictor-corrector* z dokładnością do piątego rzędu. Optymalizacja sposobu wykrywania kolizji została wykonana metodą tzw. *haszowania przestrzeni*. Polega ona na tym, iż przestrzeń dzielona jest na mniejsze podprzestrzenie, natomiast kolizji cząstki szukamy tylko w tej podprzestrzeni, w której się znajdujemy oraz w sąsiednich podprzestrzeniach. Pozwala to na zredukowanie liczby sprawdzeń możliwych kolizji, a więc i przyspieszenie obliczeń.

Rozdział 3

Opis eksperymentu

3.1 Omówienie symulacji

Aby sprawdzić poprawność modelu opisanego w rozdziale 2.1, postanowiłem badać kąt nachylenia sterty usypanej z granulek. Każda z przeprowadzonych symulacji składała się z trzech części:

1. wypełnienia pojemnika granulkami,
2. otwarcia bocznej ścianki,
3. relaksacji układu.

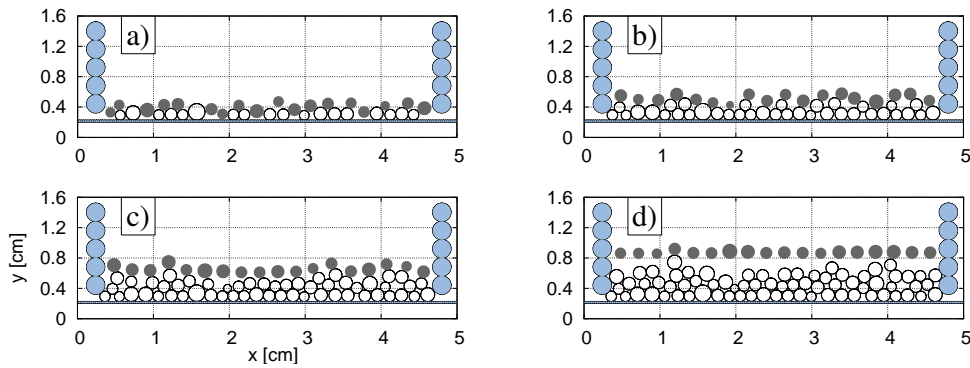
Po tych trzech krokach dokonuję pomiaru kąta nachylenia zbocza usypanej sterty kul i porównuję do równania 1.2

3.2 Przeprowadzanie pomiarów

3.2.1 Wypełnianie pojemnika

Pierwszym i zarazem najważniejszym punktem całej symulacji jest wypełnienie pojemnika kulami. Jego powodzenie ma znaczący wpływ na przebieg i stabilność kolejnych dwóch punktów. Warto także wspomnieć, iż większość czasu symulacji zajmuje właśnie ten punkt.

Pojemnik jest przestrzenią ograniczoną przez dwie ściany oraz podstawę. Ściany zbudowane są z ułożonych obok siebie kul o promieniu $R_{wall} = 0.1\text{cm}$ i masie 1 grama. Podstawa składa się z mniejszych kul. Ma to na celu uniknięcie zbyt dużego wpływu tarcia podłoża na usypany układ. W większości wykonywanych pomiarów promień kul podłoża miał ustaloną wartość $R_p = 0.025\text{cm}$, natomiast



Rysunek 3.1: Stopniowe wypełnianie pojemnika poprzez usypywanie kolejnych warstw.

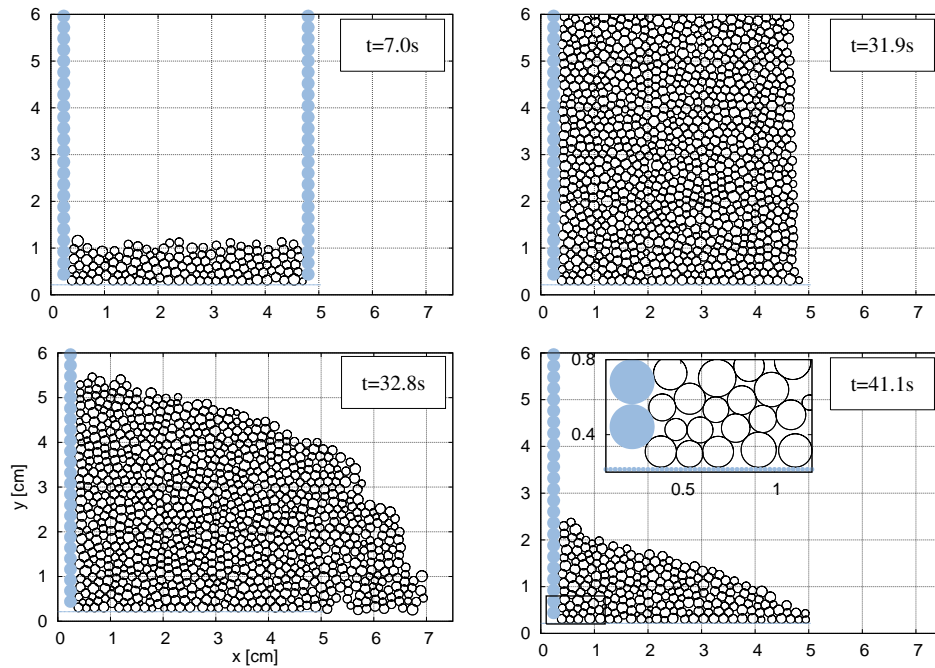
masa obliczona została na podstawie objętości kuli. Gęstość została dobrana tak, aby dla promienia 0.1cm otrzymać masę 1 grama.

Po pierwszych próbach okazało się, że bezpośrednie wsypywanie wszystkich kul do pojemnika powoduje powstawanie sporych naprężeń między kulkami. Naciskanie kul na siebie generuje duże wartości sił, co powoduje wydostawanie się kul z pojemnika poprzez ściany. Taką metodą niemożliwe jest uzyskanie stabilnego układu, gdyż przy szybkim nasypywaniu kul na siebie, kule położone wyżej utrudniają poruszanie się tym położonym w niższych warstwach. W wyniku czego otrzymujemy rejony o wysokich naprężeniach i układ przestaje być stabilny numerycznie.

Na podstawie kilku wykonanych przeze mnie prób wykonania tej fazy doszedłem do wniosku, że najlepsza jest metoda usypywania warstwami. Rysunek 3.1 przedstawia schemat usypywania czterech warstw. Nowa warstwa kul pojawia się na wysokości najwyższej ostatnio położonej kuli (plus odległość wynikająca ze scałkowania siły ciężenia) i spada swobodnie. Po usypaniu jednej warstwy należy odczekać, aż wszystkie kule się ustabilizują. Moim pomysłem było czekanie, aż średnia prędkość przekroczy ustaloną wartość v_{min} prędkości. Wartość ta została dobrana empirycznie tak, aby otrzymać stabilne i szybkie wypełnianie pojemnika.

3.2.2 Relaksacja

Następna faza polega na usunięciu prawej ściany pojemnika i czekaniu aż kule się wysypią. Kule opuszczające obszar pojemnika są usuwane po osiągnięciu wysokości mniejszej niż jego podstawa. Nie usuwałem kul po przekroczeniu linii prawej ściany, gdyż przypuszczam, że znikanie kul mogłoby powodować dodatkowe przyspieszenie działające na każdą z nich. Jak można zaobserwować w rzeczywistym rozsypywaniu piasku - drobinki piasku stanowią same dla siebie pewien opór. Uważam, że nagłe usuwanie kul ze strumienia mogłoby powodować efekt wysysania drobin z układu. Ilustrację rozsypywania się kul można zobaczyć na rysunku 3.2.



Rysunek 3.2: Stan układu w kilku chwilach symulacji. Wypełnianie pojemnika granulami (lewy-górny). Usuwanie prawej bariery (prawy-górny). Wysypywanie się granulek z układu (lewy-dolny). Czekać na relaksację układu (prawy-dolny). Dodatkowo na ostatnim rysunku możemy zobaczyć na powiększeniu kule podłoża.

Ostatnia faza to oczekiwanie, aż część materii wydobędzie się z układu. Natomiast pozostała materia powinna się ustabilizować. W tym wypadku również czekałem, aż średnia prędkość drobin obecnych w pojemniku przekroczy minimalną wartość v_{min} .

3.2.3 Pomiar kąta

Kiedy już otrzymamy usypaną stertę (rysunek 3.2), należy wykonać pomiar kąta nachylenia zbocza. Najprostszą metodą wykonania takiego pomiaru jest podzielenie wysokości sterty przez długość podstawy, ale nie jest to zbyt dokładna metoda. Szczególnie, gdy struktura powierzchni zbocza nie jest płaszczyzną lub np. jedna z kul będzie leżeć poza usypaną stertą. Dokładniejsza i nieskomplikowana metoda, jaką wymyśliłem, to podzielenie przestrzeni na zbiór prostokątnych komórek o szerokości równej średnicy kulki. Dla każdego takiego paska znajdujemy dwie (trzy - zależnie od algorytmu) najwyższe położone kulki, resztę usuwamy. Do otrzymanego zbioru punktów (położeń kul) dopasowujemy prostą, a współczynnik nachylenia tej prostej jest równy tangensowi kąta nachylenia. Istnieje jeszcze bardziej precyzyjna metoda pomiaru, która daje najniższe kąty nachylenia spośród tych trzech, które wymieniłem. Opis można znaleźć w publikacji [11].

Rozdział 4

Wyniki

4.1 Omówienie wyników

Pierwszy zbiór pomiarów został wykonany dla najbardziej uproszczonych założeń modelu. Po pierwsze, postać siły odpychającej działającej na dwie kolidujące kule została uproszczona do postaci liniowej zależności siły od odległości między kulami:

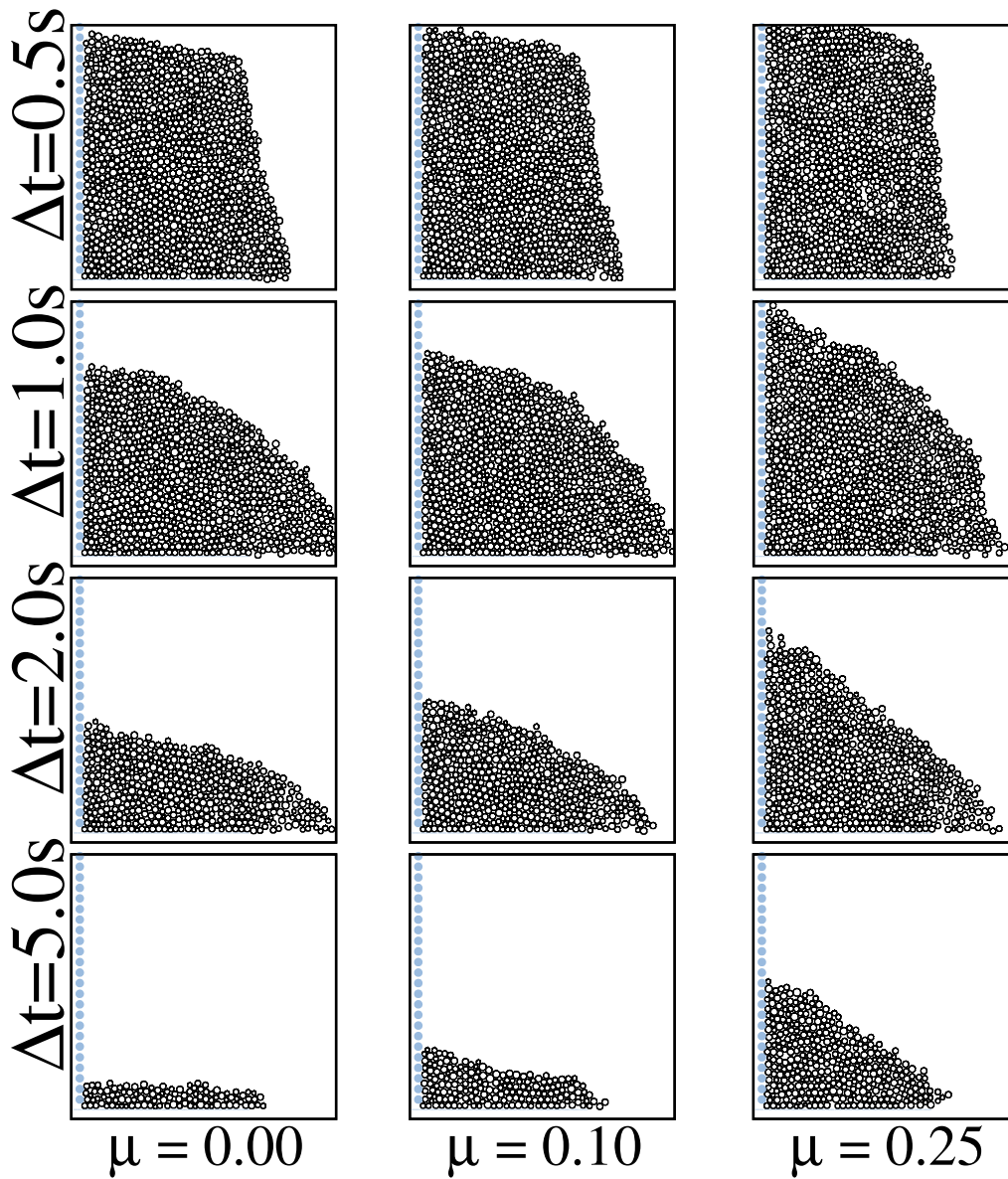
$$F_n = -k_n(R_i + R_j - |\vec{r}_{ij}|) - \gamma_s m_e (\vec{v} \cdot \vec{n}), \quad (4.1)$$

zaś człon tłumienia pozostał taki sam jak w oryginalnym modelu. Natomiast postać zastosowana w drugim modelu uwzględnia wykładnik 3/2:

$$F_n = -k_n(R_i + R_j - |\vec{r}_{ij}|)^{3/2} - \gamma_s m_e (\vec{v} \cdot \vec{n}). \quad (4.2)$$

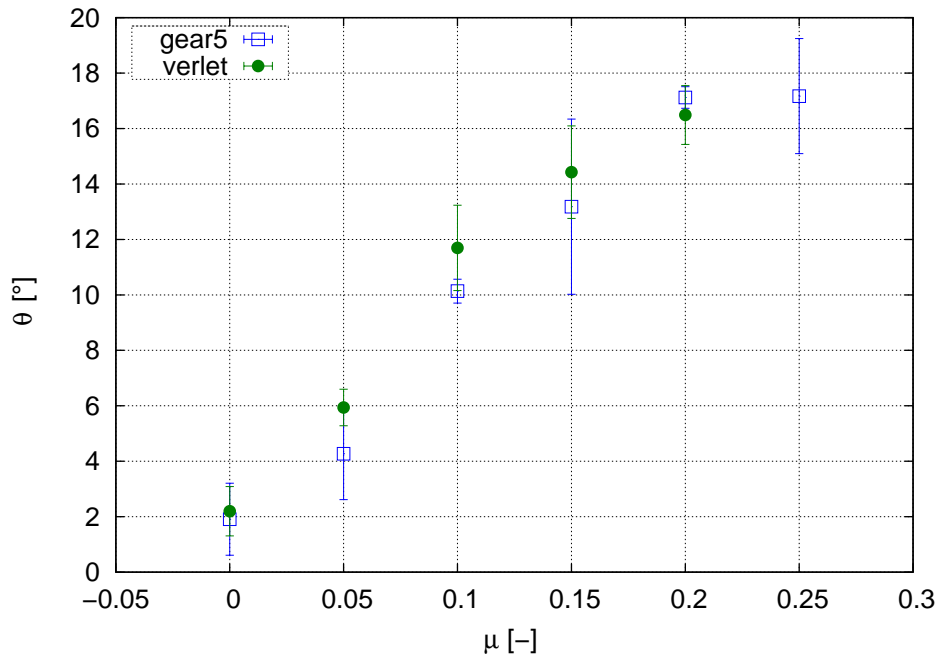
Metoda Verleta 2-go rzędu nie była stabilna dla zależności potęgowej i po pewnym czasie symulacja przestawała działać poprawnie. Pomimo tego, przetestowałem zachowanie modelu używając tej metody, nawet kosztem powyższego uproszczenia. Kolejnym uproszczeniem, jakie zastosowano podczas symulacji, jest ustawienie jednakowej masy dla wszystkich kul, równej 1 gram - niezależnie od wielkości promienia. Nie jest to aż tak znaczące uproszczenie, ponieważ rozmiary promieni zostały wylosowane przy użyciu rozkładu Gaussa ze średnim promieniem równym 0.1cm i odchyleniem 0.02cm. Więc większość kul będzie miała zbliżoną masę. W rozdziale tym, dla uproszczenia zapisu współczynnik μ_{stat} będzie równoznaczny współczynnikowi μ .

Na samym początku chciałbym pokazać, jak zmienia się wartość kąta nachylenia zbocza wraz z upływem czasu dla trzech różnych współczynników tarcia statycznego. Zestawienie zobrazowane jest na rysunku 4.1. Można na nim zobaczyć proces formowania się kopca. Różnice pomiędzy współczynnikami μ widać w nachyleniu materii dla każdej uchwyczonej chwili.



Rysunek 4.1: Formowanie się kąta nachylenia zbocza w czasie, gdzie Δt jest czasem od momentu otwarcia bariery. Porównanie dla trzech współczynników tarcia statycznego μ .

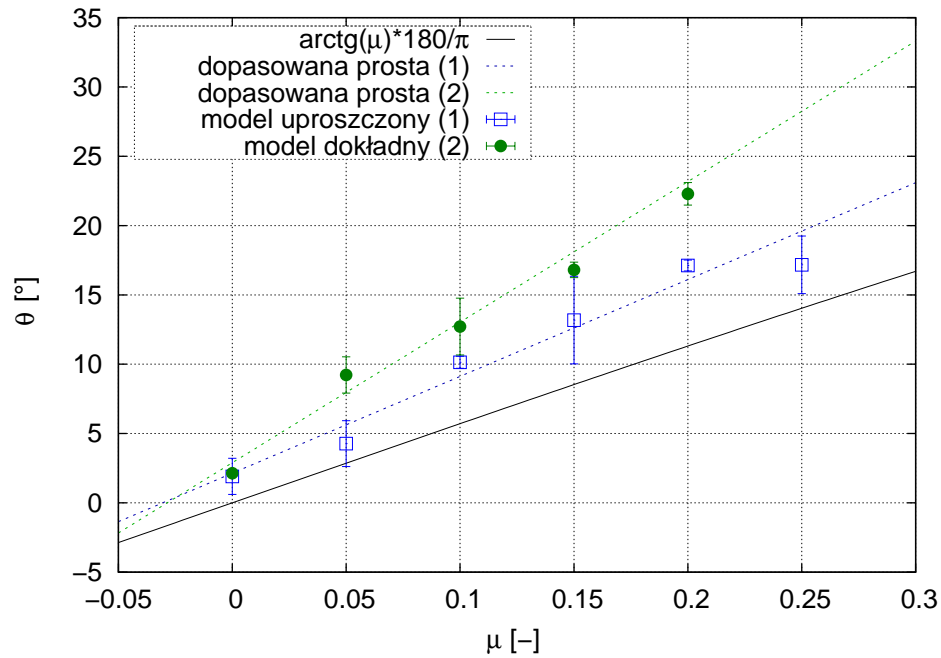
Badanie modelu zacząłem od sprawdzenia, jak się zachowują wyniki w zależności od użytej metody do rozwiązywania równań różniczkowych pierwszego rzędu (ODE). Do porównań użyłem dwóch metod: metody Verleta (wersji leap frog) z dokładnością do drugiego rzędu względem rozwinięcia w szereg Taylora oraz metody Gear5 typu *predictor-corrector* z dokładnością do piątego rzędu. Na wykresie 4.2 przedstawione zostały pomiary otrzymane przy użyciu obu metod.



Wykres 4.2: Zależność kąta nachylenia zbrocza od współczynnika tarcia dla uproszczonego modelu. Porównanie wyników dla metod Verlet i Gear5.

Natomiast błąd wyznaczony jest z trzech pomiarów dla trzech różnych konfiguracji. Każdy punkt reprezentuje średnią, a słupki błędów odchylenie standardowe. Niezależnie od wyboru metody, można uznać że wyniki są takie same, gdyż znajdują się one we wzajemnych granicach błędu statystycznego.

Na wykresie 4.3 przedstawiona jest zależność kąta od współczynnika tarcia statycznego dla uproszczonej (4.1) oraz dokładnej (4.2) zależności siły. Widzimy, iż otrzymane zależności można dobrze przybliżyć funkcją liniową. W badanym zakresie można było takich wyników oczekiwać, ale nie należy zapominać, że spodziewana zależność jest zależnością typu $\arctg(\mu)$. W przedziale $[0, 0.2]$ funkcja ta pokrywa się dość dobrze z funkcją liniową, gdyż jest to pierwszy wyraz rozwinięcia jej w szereg Maclaurina. Poprzez dopasowanie prostej do punktów pomiarowych, możemy otrzymać współczynnik skalujący k , a dopasowana funkcja jest postaci $\theta = k \arctg(\mu)$. Współczynnik ten będzie nam mówił, jak bardzo wartości są odchyłone od przewidywań doświadczalnych. Stosując takie dopasowanie dla prostej (1) otrzymuję $k = 1.22 \pm 0.13$, a dla dopasowania prostej (2) mam $k = 1.77 \pm 0.12$. Trzecia krzywa ciągła przedstawia funkcję odwrotną zależności opisanej wzorem (1.2). Między obiema metodami widać wyraźną różnicę. Z otrzymanego porównania można wysnuć następujące wnioski. Dla prostych obliczeń zachowanie punktów (1) jest podobne do (2), lecz daje inne kąty nachylenia. Uproszczony model daje wyniki bliższe teoretycznej zależności. Ostatecznie

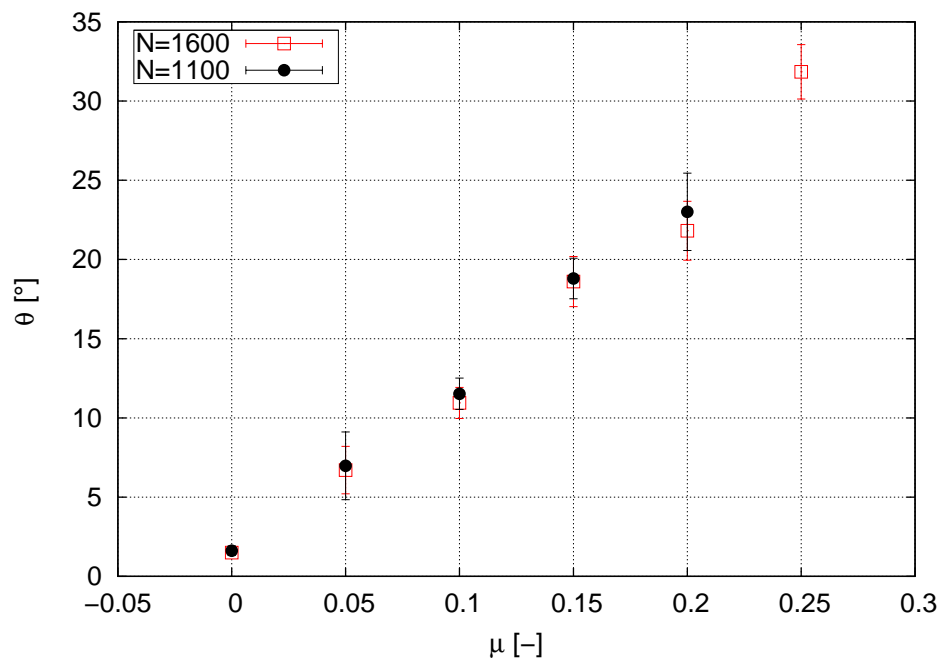


Wykres 4.3: Zależność kąta nachylenia zbrocza od współczynnika tarcia. Przycięcie wyników pomiarów do oczekiwanego rozwiązania.

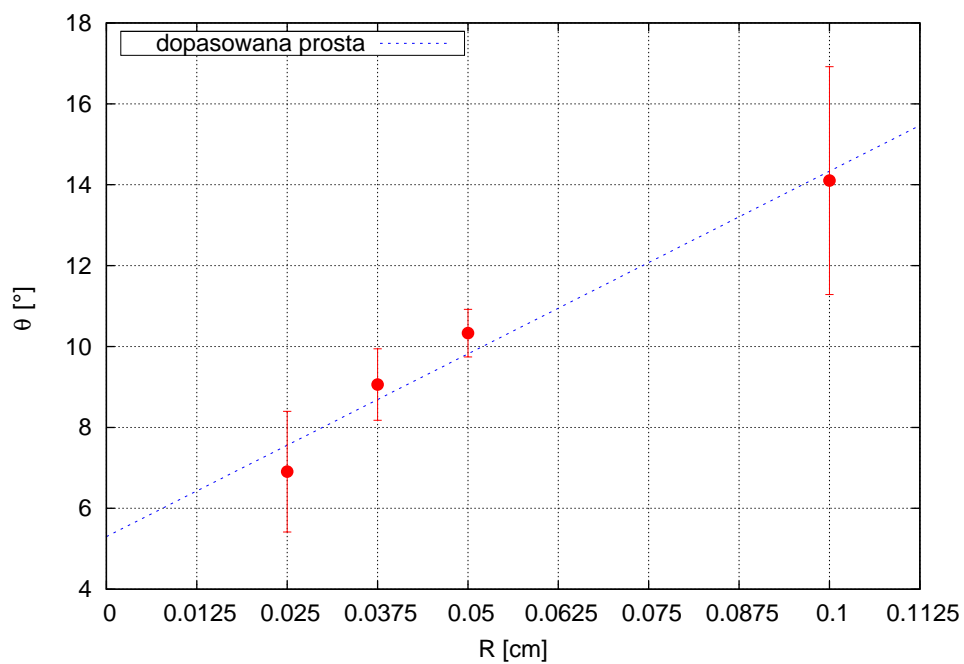
współczynnik nachylenia k prostej dla modelu uproszczonego jest mniejszy niż dla modelu dokładniejszego. Natomiast widać tu potęgę metody Gear5 - tylko ona dała sobie radę z rozwiązywaniem równań ruchu - inne metody miały zbyt małą dokładność. Myślę, że zastosowanie powyższego uproszczenia nadaje się do zastosowań w animacjach komputerowych, ze względu na mniejszą złożoność obliczeniową.

Kolejne pomiary zostały wykonane z zastosowaniem bardziej fizycznych warunków oraz skupiły się na badaniu zmienności kąta nachylenia zbrocza. Wprowadziłem masę zależną od objętości kuli, a zależności siły uwzględniłem wykładnik $3/2$. Większość pomiarów wykonano dla następujących parametrów: $k_n = 10^6 \text{ kg/s}^2$, $\gamma_s = 5 \cdot 10^2 \text{ 1/s}$, $k_s = 5 \text{ kg/s}^2$, $R_p = 0.025 \text{ cm}$, $g = 9.81 \text{ m/s}^2$, $N = 1100$ kul, $dt = 1.0 \cdot 10^{-4} \text{ s}$, gdzie: R_p - promień kul podłoża, g - przyspieszenie ziemskie, N - liczba kul wypełniająca pojemnik, dt - krok czasowy symulacji.

Pierwszą możliwością jaką sprawdziłem, była zależność kąta nachylenia od ilości kulek w pojemniku. Wykres 4.4 przedstawia porównanie dla $N = 1100$ i $N = 1600$. Nie widać, aby w tym wypadku ilość kul miała jakiegokolwiek znaczenie, gdyż obie zależności się pokrywają.



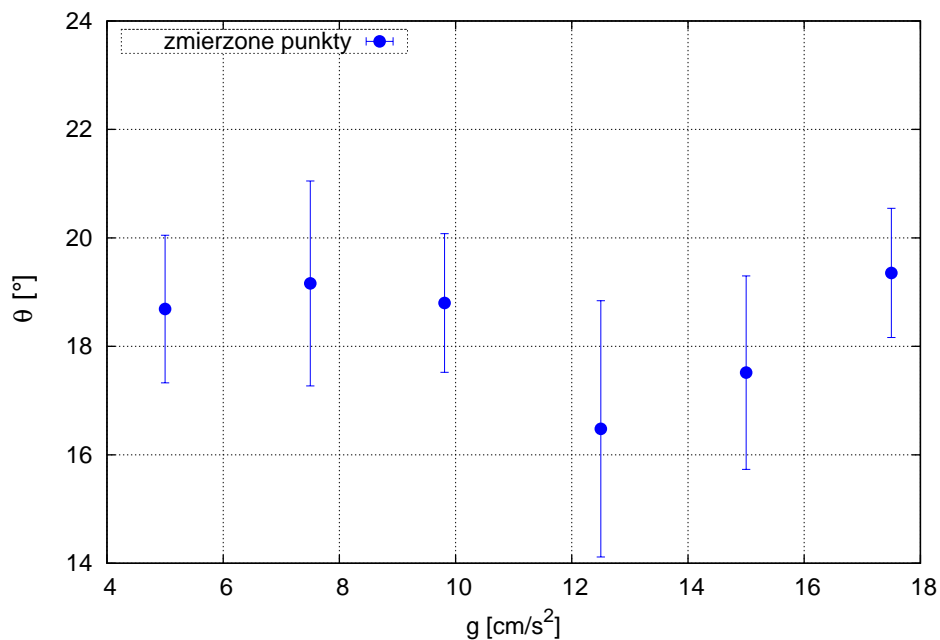
Wykres 4.4: Zależność kąta nachylenia zbroza od współczynnika tarcia dla $N=1100$ i $N=1600$ kul.



Wykres 4.5: Zależność kąta nachylenia zbroza od wielkości promienia kul podłoża.

Następny pomiar miał na celu zbadanie wpływu wielkości promienia R_p kul podłoża na wartość kąta nachylenia zbocza. Pomiar został wykonany dla ustalonego współczynnika tarcia $\mu = 0.05$ między ziarnami. Otrzymane wyniki zaprezentowane są na wykresie 4.5. Wyniki te wykazują także liniową zależność współczynnika tarcia od promienia. Współczynnik nachylenia świadczy o tym, iż zachowanie funkcji kąta od wartości promienia jest takie samo jak dla zależności kąta od współczynnika tarcia. Widzimy, iż zwiększanie wielkości promienia kul podłoża powoduje wzrost siły tarcia, co prowadzi do powstawania stert z większym kątem nachylenia. Widać, że tarcie podłoża ma tu duże znaczenie i ustawienie zbyt dużego rozmiaru kul podłoża mogłoby powodować odbieganie od zależności (1.2). Pojawienie się takiej własności pozwala nam na symulację rzeczywistych materiałów. Moglibyśmy wpływać na wartość współczynnika tarcia zmieniając stan powierzchni trącej (np. powierzchnia modelowanego podłoża). W pozostałych przeprowadzonych symulacjach wartość promienia kul podłoża została ustalona i wynosiła 0.025 cm.

W bieżącej serii pomiarów postanowiłem sprawdzić zachowanie zależności kąta nachylenia zbocza dla kilku różnych wartości przyspieszenia ziemskiego. Wykres 4.6 przedstawia otrzymane wyniki pomiarów. W tym przypadku trudno jest powiedzieć o jakiegokolwiek dobrze znanej zależności. Możliwe, iż siły oddziaływań między drobinowymi są na tyle duże, że nawet zwiększanie wartości grawitacji nie

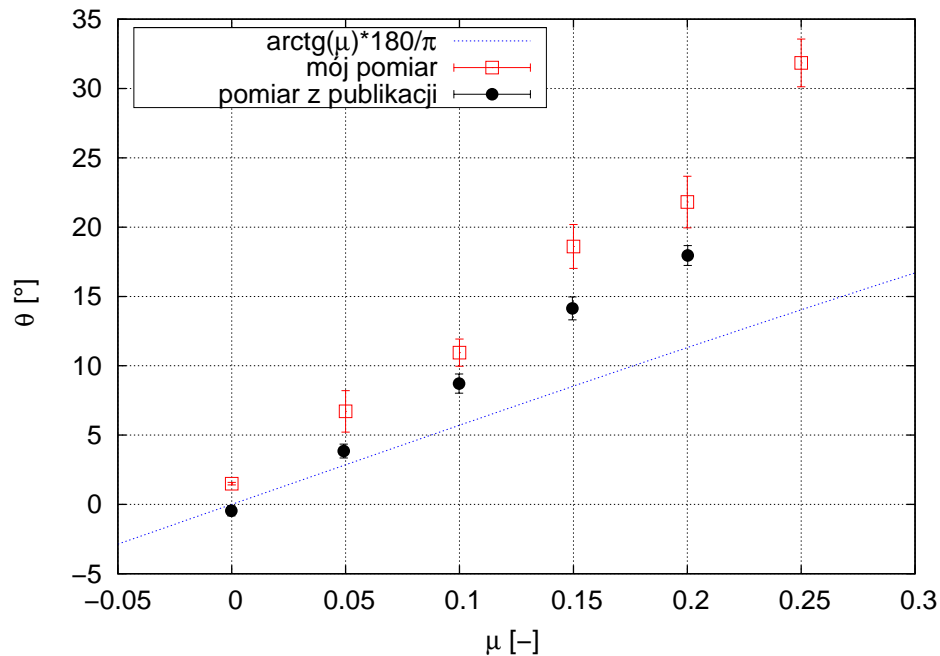


Wykres 4.6: Zależność kąta nachylenia zbocza od wartości przyspieszenia ziemskiego.

ma dużego wpływu na wartość nachylenia zbocza. Intuicyjnie wydawałoby się, że w wyniku większego napierania wyższych warstw kul na warstwy położone niżej powinna zmniejszyć się wartość współczynnika nachylenia zbocza, ale tak nie jest. Kluczową sprawą w tym wypadku staje się działanie sił tarcia statycznego, gdyż pod większym naciskiem drobin wartości sił tarcia są na tyle duże, iż mogą przeciwdziałać wpływowi siły ciężenia. Jeśli się dobrze przyjrzymy równaniom: (2.10) i (1.1) to zrozumiemy, że wzrost wartości siły nacisku powoduje wzrost wartości siły granicznej potrzebnej, aby wprawić ciało w ruch (pokonać siły tarcia statycznego).

Podsumowując wszystkie pomiary, nasuwa mi się takie skojarzenie. Myślę, że otrzymywanie tak dużych odchyłeń wartości kąta nachylenia (w niektórych wypadkach nawet o kilka stopni) można powiązać z zachowaniem obserwowanym w rzeczywistości podczas pomiaru współczynnika tarcia. Fluktuacje kąta nachylenia dobrze obrazują niepowtarzalność pomiaru współczynnika tarcia, gdyż za każdym razem mamy do czynienia z inną (zmienioną) strukturą ziaren w stercie.

Na koniec postanowiłem skonfrontować otrzymane wyniki z wynikami z publikacji [11] J.Lee'go i H.J. Herrmanna. Obie zależności znajdują się na wykresie 4.7. Dodąłem także zależność $\arctg(\mu)$ aby móc porównać oba rozwiązania w stosunku do zależności mierzalnej doświadczalnie. Okazuje się, iż pomiary wykonane przeze mnie odbiegają bardziej od empirycznego wzoru, tzn. nachylenie zależno-



Wykres 4.7: Konfrontacja kąta nachylenia zbocza z wynikiem otrzymanym przez J.Lee i H.J. Hermann (rycina 3).

ści jest większe niż w przypadku pomiarów zaczerpniętych z wyżej wymienionej publikacji. Może to być spowodowane różnicą w stosunku do szerokości pojemnika, która w moim wypadku wynosiła $W = 4.56$. Natomiast w pomiarze autorów wynosiła ona $W = 4$. Wnioskowanie te opieram na wynikach, które oni uzyskali. Wynika z nich, iż kąt nachylenia zależy też od szerokości pojemnika, z którego rozsypujemy stertę [11]. Znaczenie może też mieć wpływ wielkości promienia kul podłoża na otrzymywane kąty (np. nachylenie około 7° dla $R_p = 0.025$ cm, wykres 4.5). Być może zmniejszenie R_p pomogłoby zniwelować tę różnicę. Co więcej dość zaskakująca jest wartość $\theta \neq 0$ otrzymana dla $\mu = 0$, takie zachowanie może wynikać z metody pomiaru. Wyniki autorów zostały wygenerowane za pomocą innej metody, dającej najniższe kąty. Natomiast ja użyłem metody opisanej w rozdziale 3.2.3, co prawdopodobnie wpłynęło na otrzymywanie większych wartości. Jak widać na wykresie, także wyniki autorów nie pokrywają się z oczekiwaną zależnością. Wydaje mi się, że można by się pokusić o takie dobranie pozostałych parametrów układu, aby jak najbardziej zbliżyć otrzymywane pomiary z poprawną zależnością. Myślę, że wartym przetestowania parametrem, byłby współczynnik tłumienia k_s ze wzoru (2.10) na siłę sprężystości, pełniący funkcję tarcia statycznego.

Rozdział 5

Podsumowanie

Opisany przeze mnie model materii granulowanej okazuje się być dobrym przybliżeniem rzeczywistego zachowania usypywanej sterty drobin. Jak wykazują pomiary, kąt stabilności krytycznej jest wartością stałą dla danego materiału, w naszym wypadku charakteryzowanego przez wartość współczynnika tarcia. Otrzymana zależność kąta stabilności nie pokrywa się z oczekiwaną funkcją, pomimo tego jej zachowanie jest jej dobrym przybliżeniem. Według mnie jest jeszcze możliwość odpowiedniego dobrania parametrów modelu tak aby otrzymana krzywa pokryła się z dokładnym rozwiązaniem. Myślę, iż kluczową kwestią jest dobranie wartości współczynnika k_s we wzorze na siłę wywieraną przez *wirtualną* sprężynę odgrywającą rolę oddziaływania tarcia statycznego. Jest to zagadnienie, któremu warto byłoby się przyjrzeć w dalszej pracy nad tym modelem. Model ten może być dobrą bazą do tworzenia bardziej złożonych symulacji, gdyż bazuje on na modelu odkształceń sprężystych Hertza. Daje to duże możliwości użycia go w symulacjach uwzględniających naprężenia materiału. Dobrym pomysłem byłoby użycie go do symulacji pęknięcia materiału. Jak wiadomo w wyniku naprężeń między drobinami powstają łańcuchy sił, które mogłyby wyznaczać linie pęknięć.

Podczas pracy nad symulacją modelu materii granulowanej miałem styczność z kilkoma innymi złożonymi modelami [10, 12, 3], spośród nich wybrałem ten, który był najdokładniej opisany. Jest to argument na korzyść przedstawionego w tej pracy modelu. Trudności, jakie możemy napotkać podczas implementacji modelu, związane są z implementacją tarcia statycznego. Wymaga ona odrębnych struktur danych do przechowywania informacji potrzebnych do tworzenia lub usuwania wirtualnej sprężyny. Kolejną trudność, jaką możemy napotkać to odpowiednie dobranie parametrów modelu. Najważniejszymi według mnie to wartość kroku symulacji dt , ponieważ zbyt duża wartość wpływa negatywnie na stabilność oraz dokładność otrzymywanego rozwiązania równań ruchu. Następnym ważnym parametrem jest masa drobin. Okazało się, iż najlepsze rezultaty dla tego modelu

otrzymuje się dla masy około 10^{-3} kg (rzeczywista wartość np. dla drobin piasku to około 10^{-5} kg). Na odpowiednie dobranie tych parametrów poświęciłem najwięcej czasu.

Okazało się, iż kluczową kwestią działania tego typu symulacji jest metoda numeryczna rozwiązująca równania ruchu. Musi ona zapewnić odpowiednią stabilność oraz dokładność rozwiązania. Początkowo próbowałem metod 2-go rzędu, jednak okazały się one niewystarczające. Dopiero metoda *Gear5* zapewniała odpowiednią poprawność rozwiązań. Jest to zgodne z obserwacjami innych badaczy [12, 10]. Opis oraz implementacja tej metody dostępna jest w załączniku.

Symulacje materii granulowanej nie należą do wydajnych, gdyż układy w takich modelach są gęsto upakowane i liczba oddziaływań w jednej chwili czasu jest duża. To przekłada się na dużą liczbę obliczeń w krótkiej chwili czasu. Nawet przy użyciu odpowiednich technik podziału przestrzeni, symulacja zaledwie 1100 kul dawała 12 klatek na sekundę na procesorze o taktowaniu 1.8GHz. Widać, że symulowanie układów z kilkudziesięciu tysiącami kul jest niemożliwe dla współczesnych procesorów. Jest to problem dla twórców symulacji czasu rzeczywistego. Pomimo tak dużej złożoności obliczeniowej, model ten ma tę zaletę (oddziaływania krótkozasięgowe), że daje on możliwości wprowadzenia wielowątkowych obliczeń (CPU, GPU). To być może pozwoli na przeprowadzenie tego typu symulacji w czasie rzeczywistym w przyszłości (np. w grach komputerowych) [13].

Dodatki

Dodatek A

Metoda całkująca Geara

Załączam ten dodatek, ponieważ miałem problem ze znalezieniem dokładnego opisu i implementacji tej metody, mam nadzieję, że będzie on przydatny dla innych osób chcących skorzystać z tej metody. Opracowany on został na podstawie [14].

Zalety metody:

- dokładność piątego rzędu.
- jednokrotne wywołanie funkcji liczącej siły.
- stosunkowo mała złożoność obliczeniowa jak na metodę tego rzędu.

Wady metody:

- w większości wypadków dla utrzymania dużej dokładności wyników wymaga inicjalizacji pochodnych położenia \vec{r} ,
- do poprawnego działania wymaga małych kroków czasowych.

Najważniejsze kroki algorytmu:

1. krok przewidywania
2. wyliczenie sił
3. korekcja przewidzianych rozwiązań

Krok przewidywania jest odpowiedzialny za przewidzenie wartości położeń, prędkości, przyspieszeń i kolejnych pochodnych położenia dla następnego kroku symulacji $t + \Delta t$. Wartość funkcji wyliczana jest z rozwinięcia jej w szereg Taylora z

dokładnością do n-tego rzędu. Poniżej przedstawiam formalny zapis kroku przewidywania:

$$\begin{aligned}\vec{r}_p(t + \Delta t) &= \vec{r}(t) + \vec{r}^I(t)\Delta t + \vec{r}^{II}(t)\frac{(\Delta t)^2}{2!} + \vec{r}^{III}(t)\frac{(\Delta t)^3}{3!} + \vec{r}^{IV}(t)\frac{(\Delta t)^4}{4!} + \vec{r}^V(t)\frac{(\Delta t)^5}{5!} \\ \vec{r}_p^I(t + \Delta t) &= \vec{r}^I(t) + \vec{r}^{II}(t)\Delta t + \vec{r}^{III}(t)\frac{(\Delta t)^2}{2!} + \vec{r}^{IV}(t)\frac{(\Delta t)^3}{3!} + \vec{r}^V(t)\frac{(\Delta t)^4}{4!} \\ \vec{r}_p^{II}(t + \Delta t) &= \vec{r}^{II}(t) + \vec{r}^{III}(t)\Delta t + \vec{r}^{IV}(t)\frac{(\Delta t)^2}{2!} + \vec{r}^V(t)\frac{(\Delta t)^3}{3!} \\ \vec{r}_p^{III}(t + \Delta t) &= \vec{r}^{III}(t) + \vec{r}^{IV}(t)\Delta t + \vec{r}^V(t)\frac{(\Delta t)^2}{2!} \\ \vec{r}_p^{IV}(t + \Delta t) &= \vec{r}^{IV}(t) + \vec{r}^V(t)\Delta t \\ \vec{r}_p^V(t + \Delta t) &= \vec{r}^V(t)\end{aligned}$$

W kolejnym kroku sprawdzamy kolizję i liczymy siły działające na każde ciało, używając do ich policzenia przewidzianych pozycji oraz prędkości. A z policzonych wartości sił wyliczamy przyspieszenia.

W kroku korekcji na samym początku dla każdego ciała wyliczymy różnicę między wyliczonym i przewidywanym przyspieszeniem:

$$\Delta\vec{r}^{II} = \vec{r}^{II}(t + \Delta t) - \vec{r}_p^{II}(t + \Delta t).$$

W algorytmie Geara ta różnica przyspieszeń posłuży do korekcji wszystkich pochodnych położenia w następujący sposób:

$$\begin{aligned}\vec{r} &= \vec{r}_p + \alpha_0\Delta\mathbf{R2} \\ \vec{r}^I\Delta t &= \vec{r}_p^I\Delta t + \alpha_1\Delta\mathbf{R2} \\ \frac{\vec{r}^{II}(\Delta t)^2}{2!} &= \frac{\vec{r}_p^{II}(\Delta t)^2}{2!} + \alpha_2\Delta\mathbf{R2} \\ \frac{\vec{r}^{III}(\Delta t)^3}{3!} &= \frac{\vec{r}_p^{III}(\Delta t)^3}{3!} + \alpha_3\Delta\mathbf{R2} \\ \frac{\vec{r}^{IV}(\Delta t)^4}{4!} &= \frac{\vec{r}_p^{IV}(\Delta t)^4}{4!} + \alpha_4\Delta\mathbf{R2} \\ \frac{\vec{r}^V(\Delta t)^5}{5!} &= \frac{\vec{r}_p^V(\Delta t)^5}{5!} + \alpha_5\Delta\mathbf{R2}\end{aligned}$$

gdzie

$$\Delta\mathbf{R2} \equiv \frac{\Delta\vec{r}^{II}(\Delta t)^2}{2!}.$$

Współczynniki korekcji α_i są unikalnymi liczbami wyprowadzonymi dla metody Geara z dokładnością rzędu piątego i jednocześnie rozwiązującej pochodne drugiego stopnia. Wartości współczynników umieszczone są w tabeli A.1, są tam także współczynniki dla metod z niższym rzędem dokładności.

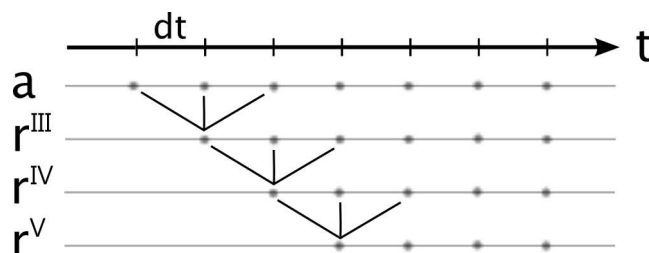
| D | α_0 | α_1 | α_2 | α_3 | α_4 | α_5 |
|---|------------------|-------------------|------------|-----------------|----------------|----------------|
| 3 | $\frac{1}{6}$ | $\frac{5}{6}$ | 1 | $\frac{1}{3}$ | - | - |
| 4 | $\frac{19}{120}$ | $\frac{3}{4}$ | 1 | $\frac{1}{2}$ | $\frac{1}{12}$ | - |
| 5 | $\frac{3}{16}$ | $\frac{251}{360}$ | 1 | $\frac{11}{18}$ | $\frac{1}{6}$ | $\frac{1}{60}$ |

Tabela A.1: Wartości współczynników korygujących w metodzie Geara dla pochodnych drugiego stopnia używając przewidywania z dokładnością do rzędu \mathbf{D} .

Ważną kwestią dla poprawnego działania algorytmu jest inicjalizacja parametrów początkowych układu. W tym przypadku trzeba nadać wartości początkowe dla pochodnych stopnia wyższego niż drugi, co w ogólności nie jest prostą sprawą. W większości przypadków budując symulację, znamy co najwyżej siły działające w układzie lub potencjały generujące te siły, ale nie znamy wyższych pochodnych. Inicjalizację tych parametrów można wykonać na dwa sposoby:

1. przeliczyć kilka kroków symulacji inną metodą i na ich podstawie wyliczyć wartość wyższych pochodnych,
2. ustawić ich wartości na zero.

Pierwszy sposób jest trudniejszy, ale ogólniejszy i gwarantuje dużą dokładność metodzie całkującej. Ja wykonałem to w następujący sposób. Dla kroku czasowego dużo mniejszego niż krok czasowy symulacji. Wykonuje sześć kroków symulacji metodą Runge Kutty 4-tego rzędu, podczas każdego kroku zapamiętuję wartości przyspieszenia dla każdego ciała. Dodatkowo w trzecim kroku zapamiętuję wartość prędkości i położenia, zasadność czynności wyjaśni rysunek A.1. Następnie na podstawie zapamiętanych przyspieszeń liczę wartości wyższych pochodnych dla każdego ciała. Na koniec ustawiam wartości początkowe zapamiętane w kroku trzecim. Wyliczam jeszcze różnicę czasu między krokiem symulacji a czasem, który upłynął w przeliczonych trzech krokach. Po to, aby każdy krok symulacji miał taką samą długość.



Rysunek A.1: Przedstawia wyliczone punkty pochodnych, w kolejnych krokach wyższa pochodna r . Linie pokazują punkty użyte do wyliczenia pochodnej w konkretnym czasie.

Drugi sposób jest bardzo prosty i można go stosować, wtedy gdy przez kilka pierwszych kroków symulacji ciała nie oddziałują, ponieważ dostrojenie się wartości wyższych pochodnych będzie wymagało przejścia kilkunastu kroków symulacji. Przez takie działanie podczas tych pierwszych kroków nie uwzględniamy wpływu wyższych pochodnych na rozwiązanie. Ja radziłbym porównać działanie symulacji dla dwóch sposobów i jeśli w obu przypadkach działa ona poprawnie wybrać prostsze rozwiązanie.

Poniżej zamieszczony kod zawiera metodę Geara, a także metodę inicjalizującą wartości wyższych pochodnych. W metodzie tej linie 4-12 odpowiedzialne są za tworzenie wektorów przechowujących przyśpieszenia dla każdego ciała `Body` w sześciu krokach. Natomiast pętla wykonuje sześć kroków symulacji (linie 15-51), zapamiętując wartości przyśpieszeń działających na ciała. Linie 45-50 odpowiedzialne są za zapamiętanie położenia i prędkości w trzecim kroku, które będą użyte jako wartości inicjalizacyjne. Trzeci krok jest pierwszym krokiem, w którym znamy wartości wszystkich wyższych pochodnych, dlatego z niego startujemy symulację. Następnie z zapamiętanych przyśpieszeń wyliczamy wartości wyższych pochodnych (linie 55-69). Funkcje `derive` i `derive2` są funkcjami liczącymi wartość pierwszej pochodnej i drugiej pochodnej z trzech punktów funkcji. Na koniec metoda inicjalizująca zwraca długość kroku czasowego pozostałego do przeliczenia. Funkcja `int_gear5` zawiera implementację metody Geara. Występujące w niej funkcje `add` oraz `Vec2Mul` są odpowiedzialne za dodawanie oraz mnożenie wektora przez liczbę. Natomiast funkcja `collisionDetection` wstawia sprężynę `Spring` do tablicy sprężyn `springs`, odpowiadających za reakcję na kolizję pary kul. Funkcja `intGear5` jest to właściwa metoda licząca (linie 169-178), która w pierwszym kroku wykona inicjalizację wszystkich potrzebnych parametrów.

Listing A.1: Implementacja metody w języku c++

```

1 double ODE::initGear5(double dt)
2 {
3     double sdt = dt / 130.0; /// small dt
4     std::vector<Vec2> array_of_v0;
5     std::vector<Vec2> array_of_x0;
6     std::vector<Vec2> array_of_a0;
7     /// pamiętamy w niej pierwsze 7 przyśpieszeń
8     std::vector<ArrayOfVec2> array_of_a;
9     array_of_v0.reserve(bodies.size());
10    array_of_x0.reserve(bodies.size());
11    array_of_a0.reserve(bodies.size());
12    array_of_a.reserve(bodies.size());
13
14    for (uint i=0; i<7; ++i)

```

```

15     {
16         /// czyści tmp_r[] dla każdego ciała
17         clearBodyData ();
18
19         /// sumujemy przyśpieszenia działające na ciało
20         for (uint j=0; j<springs.size(); ++j)
21         {
22             Spring &sp = springs[j];
23             Vec2Pair as = equationCall(sp);
24
25             bodies[ sp.body1_id ].tmp_r[2].add(as.v1);
26             bodies[ sp.body2_id ].tmp_r[2].add(as.v2);
27         }
28
29         /// zapamiętanie przyśpieszeń w array_of_a
30         for (uint bi=0; bi<bodies.size(); ++bi)
31         {
32             array_of_a.push_back( ArrayOfVec2() );
33             array_of_a[bi].push_back( bodies[bi].tmp_r[2] );
34             /// lokalna wartość przyśpieszenia
35
36             /// pamiętanie wartości startowych
37             if (i == 4)
38                 array_of_a0.push_back( bodies[bi].tmp_r[2] );
39         }
40
41         if (i < 6)
42             intrk4(sdt); /// przeliczenie całego układu o krok dt
43
44         /// pamiętanie wartości startowych
45         if (i == 3)
46             for (uint bi=0; bi<bodies.size(); ++bi)
47             {
48                 array_of_x0.push_back( bodies[bi].r );
49                 array_of_v0.push_back( bodies[bi].v );
50             }
51     }
52
53     /// liczenie wyższych pochodnych r
54     for (uint bi=0; bi<bodies.size(); ++bi)
55     {

```

```

56     ArrayOfVec2 b = DeriveMethod::derive(array_of_a[bi], sdt);
57     ArrayOfVec2 c = DeriveMethod::derive2(array_of_a[bi], sdt);
58     ArrayOfVec2 d = DeriveMethod::derive2(b, sdt);
59
60     /// ustawienie wartości startowych
61     Body &body = bodies[bi];
62     body.tmp_r[0] = array_of_x0[bi];
63     body.tmp_r[1] = array_of_v0[bi];
64     body.tmp_r[2] = array_of_a0[bi];
65     /// wyższe pochodne położenia
66     body.tmp_r[3] = b[2];
67     body.tmp_r[4] = c[2];
68     body.tmp_r[5] = d[1];
69 }
70
71     for (uint i=0; i<bodies.size(); ++i)
72         bodies[i].tmp_r[7] = Vec2Zero;
73
74     /// bo zatrzymaliśmy się na trzecim kroku i
75     /// nieznamy wcześniejszych poch.
76     return (dt - 3.0*sdt);
77 }
78
79 void ODE::int_gear5(double dt)
80 {
81     /// przewidywanie
82     static double dt2 = dt*dt, dt3 = pow(dt, 3),
83                 dt4 = pow(dt, 4), dt5 = pow(dt, 5);
84     static double s2 = 1.0/2.0, s3 = 1.0/6.0,
85                 s4 = 1.0/24.0, s5 = 1.0/120.0;
86     static double dt2_s2 = dt2*s2, dt3_s3 = dt3*s3,
87                 dt4_s4 = dt4*s4, dt5_s5 = dt5*s5;
88
89     for (uint bi=0; bi<bodies.size(); ++bi)
90     {
91         Body &b = bodies[bi];
92         if (b.isStatic) continue;
93
94         Vec2 r[6];
95         Vec2 *r = b.tmp_r;
96

```

```

97         rp[0] = r[0];
98         rp[0].add( Vec2Mul(r[1], dt) );
99         rp[0].add( Vec2Mul(r[2], dt2_s2) );
100        rp[0].add( Vec2Mul(r[3], dt3_s3) );
101        rp[0].add( Vec2Mul(r[4], dt4_s4) );
102        rp[0].add( Vec2Mul(r[5], dt5_s5) );
103
104        rp[1] = r[1];
105        rp[1].add( Vec2Mul(r[2], dt) );
106        rp[1].add( Vec2Mul(r[3], dt2_s2) );
107        rp[1].add( Vec2Mul(r[4], dt3_s3) );
108        rp[1].add( Vec2Mul(r[5], dt4_s4) );
109
110        rp[2] = r[2];
111        rp[2].add( Vec2Mul(r[3], dt) );
112        rp[2].add( Vec2Mul(r[4], dt2_s2) );
113        rp[2].add( Vec2Mul(r[5], dt3_s3) );
114
115        rp[3] = r[3];
116        rp[3].add( Vec2Mul(r[4], dt) );
117        rp[3].add( Vec2Mul(r[5], dt2_s2) );
118
119        rp[4] = r[4];
120        rp[4].add( Vec2Mul(r[5], dt) );
121
122        for (int i=0; i<5; ++i)
123            b.tmp_r[i] = rp[i];
124
125        b.r = rp[0];
126        b.v = rp[1];
127    }
128
129    /// liczenie sił
130    // jeśli ciała kolidują to dodaj między nimi sprężynę
131    collisionDetection();
132
133    for (uint si=0; si<springs.size(); ++si)
134    {
135        Spring &sp = springs[si];
136        Vec2Pair as = equationCall(sp);
137

```

```

138         bodies [ sp.body1_id ].tmp_r[7].add( as.v1 );
139         bodies [ sp.body2_id ].tmp_r[7].add( as.v2 );
140     }
141
142     /// korekcja
143     static double c[6] = {3.0/16.0, 251.0/360.0, 1.0,
144                          11.0/18.0, 1.0/6.0, 1.0/60.0};
145     static double tt[6] = {1.0, 1.0/dt, 2.0/dt2, 6.0/dt3,
146                          24.0/dt4, 120.0/dt5};
147     static double ctt[6] = {c[0]*tt[0], c[1]*tt[1],
148 c[2]*tt[2], c[3]*tt[3], c[4]*tt[4], c[5]*tt[5]};
149
150     for (uint bi=0; bi<bodies.size(); ++bi)
151     {
152         Body &b = bodies[bi];
153         if (b.isStatic) continue;
154
155         Vec2 DA = Vec2Sub(b.tmp_r[7], b.tmp_r[2]);
156         Vec2 DR2 = Vec2Mul(DA, dt2_s2);
157
158         for (int i=0; i<6; ++i)
159             b.tmp_r[i].add( Vec2Mul(DR2, ctt[i]) );
160
161         b.r = b.tmp_r[0];
162         b.v = b.tmp_r[1];
163
164         // czyści tymczasowe przyśpieszenie
165         b.tmp_r[7] = Vec2Zero;
166     }
167 }
168
169 void ODE::intGear5(double dt)
170 {
171     if (!gear5_initialized)
172     {
173         dt = initGear5(dt);
174         gear5_initialized = true;
175     }
176
177     int_gear5(dt);
178 }

```

Bibliografia

- [1] Granular material. [online]. Dostępny w Internecie pod adresem: http://en.wikipedia.org/wiki/Granular_material.
- [2] G. William Baxter, R. P. Behringer, Timothy Fagert, and G. Allan Johnson. Pattern formation in flowing sand. *Phys. Rev. Lett.*, 62:2825–2828, Jun 1989.
- [3] Gerald H. Ristow and Hans J. Herrmann. Density patterns in two-dimensional hoppers. *Phys. Rev. E*, 50:R5–R8, Jul 1994.
- [4] Anthony Rosato, Katherine J. Strandburg, Friedrich Prinz, and Robert H. Swendsen. Why the brazil nuts are on top: Size segregation of particulate matter by shaking. *Phys. Rev. Lett.*, 58:1038–1040, Mar 1987.
- [5] Jeffrey Olafsen. *Experimental and Computational Techniques in Soft Condensed Matter Physics*. Cambridge University Press, 2010. rozdział 9.
- [6] M. Hebda and A. Wachal. *Trybologia*. Wydawnictwo Naukowo-Techniczne, 1980.
- [7] Angle of repose. [online]. Dostępny w Internecie pod adresem: http://en.wikipedia.org/wiki/Angle_of_repose.
- [8] Przemysław Borys. *Skąd się bierze tarcie?* FOTON, 2009.
- [9] K.L. Johnson. *Contact Mechanics*. Cambridge University Press, 1985.
- [10] N. Bell, Y. Yu, and P. J. Mucha. *Particle-Based Simulation of Granular Materials*. ACM SIGGRAPH, 2005.
- [11] J. Lee and H.J. Herrmann. Angle of repose and angle of marginal stability: molecular dynamics of granular particles. *Journal of Physics A: Mathematical and General*, 26(2):373, 1993.
- [12] Thorsten Pöschel and Volkhard Buchholtz. Static friction phenomena in granular materials: Coulomb law versus particle geometry. *Phys. Rev. Lett.*, 71:3963–3966, Dec 1993.
- [13] Rachan Tananuchittikul. *Particle-Based Simulation of Granular Materials on GPUs*. California State University, Fullerton, 2011.
- [14] J.M. Haile. *Molecular Dynamics Simulation - Elementary Methods*. Clemson University, 1992.

Spis rysunków

| | | |
|-----|--|----|
| 1.1 | Oznaczenie kąta stabilności | 5 |
| 2.1 | Oddziaływanie plastycznych kul | 8 |
| 2.2 | Położenie wirtualnej sprężyny | 10 |
| 3.1 | Stopniowe wypełnianie pojemnika | 14 |
| 3.2 | Przykład usypywania sterty | 15 |
| 4.1 | Formowanie się kąta nachylenia zbocza | 18 |
| 4.2 | Porównanie wyników dla metod Verlet i Gear5 | 19 |
| 4.3 | Przyrównanie pomiarów do oczekiwanego rozwiązania | 20 |
| 4.4 | Zachowanie kąta nachylenia dla różnej ilości kul | 21 |
| 4.5 | Zależność kąta nachylenia od wielkości kul podłoża | 21 |
| 4.6 | Zależność kąta nachylenia od przyspieszenia ziemskiego | 22 |
| 4.7 | Konfrontacja wyników pomiaru | 23 |
| A.1 | Wyliczenie punktów wyższych pochodnych | 31 |