

Instrukcja używania programu

Opracowanie przedstawia sposób uruchomienia oraz wytyczne do danych wejściowych dla programu stworzonego w ramach projektu „Proceduralne tworzenie efektów atmosferycznych 3D przy użyciu CUDA”. Instrukcja podzielona jest na dwie części. W pierwszej przedstawiono działania jakie należy podjąć, aby uruchomić program, a w drugiej opisano znaczenie parametrów wejściowych oraz to, jak je dobrać.

Uruchamianie programu

1. Wstęp

W pierwszej części instrukcji pokazano, jak można uruchomić program stworzony w ramach projektu „Proceduralne tworzenie efektów atmosferycznych 3D przy użyciu CUDA”.


2. Wymagania wstępne

Program został skompilowany dla komputerów z systemem Windows. Do poprawnego działania programu należy posiadać kartę graficzną wspierającą technologię CUDA. Dodatkowo na komputerze powinien być zainstalowany sterownik NVIDIA CUDA Runtime.

3. Uruchomienie programu

W celu uruchomienia programu należy wykonać następujące kroki:

1. Program znajduje się w repozytorium projektu w katalogu build. Jego nazwa to MPM_MC_CUDA.exe. Należy umieścić go w folderze, w którym ma być uruchamiany. Poniżej przykładowa lokalizacja.

Ten komputer > Dysk lokalny (C:) > projekt_badawczy > MPM_MC_CUDA > program			
Nazwa	Data modyfikacji	Typ	Rozmiar
 MPM_MC_CUDA.exe	09.03.2025 00:05	Aplikacja	1 101 KB

2. Program przeprowadza symulację na podstawie przekazanych mu plików wejściowych. Kilka przykładowych przypadków zostało załączone w repozytorium w katalogu examples. W instrukcji rozważony zostanie przykład umieszczony w katalogu examples/snowballs_collision. Przedstawia on symulację zderzenia niewspółosiowego dwóch kul śnieżnych.

Należy stworzyć nowy folder, w którym umieszczone będą pliki o rozszerzeniu .obj z siatkami z katalogu examples/snowballs_collision z repozytorium. Siatki te później będą wczytane do symulacji. Folder ten może znajdować się w dowolnym miejscu, tutaj przyjęto, że będzie to C:\projekt_badawczy\MPM_MC_CUDA\snowballs_collision. Dodatkowo w folderze tym tworzymy folder przeznaczony do przechowywania wyników symulacji, tutaj nazwiemy go „out”. Powinno to wyglądać następująco:

Ten komputer > Dysk lokalny (C:) > projekt_badawczy > MPM_MC_CUDA > snowballs_collision

Nazwa	Data modyfikacji	Typ	Rozmiar
out	08.03.2025 23:38	Folder plików	
kule_obok_kula1.obj	19.01.2025 01:32	3D Object	7 KB
kule_obok_kula2.obj	19.01.2025 01:32	3D Object	7 KB
kule_walls.obj	19.01.2025 01:32	3D Object	1 KB

- Poza plikami .obj w katalogu examples/snowballs_collision znajduje się plik config.json. Należy umieścić go w folderze, w którym znajduje się program.

Ten komputer > Dysk lokalny (C:) > projekt_badawczy > MPM_MC_CUDA > program

Nazwa	Data modyfikacji	Typ	Rozmiar
config.json	09.03.2025 00:07	Plik JSON	1 KB
MPM_MC_CUDA.exe	09.03.2025 00:05	Aplikacja	1 101 KB

- Plik config.json należy otworzyć edytorem tekstowym, aby dopasować ścieżkę do folderu, który został utworzony w kroku 2. W tym celu należy znaleźć linię z „base_path”

```
"animation": {
  "step": 10
},
"base_path": "C:\\sciezka\\do\\folderu\\",
"paths": {
  "result_name": "zderzenie_kul",
  "result_path": "out\\",
  "walls_name": "kule_walls.obj",
  "walls_path": "",
  "snow_objects_paths": [ "", "" ],
  "snow_objects_names": [ "kule_obok_kula1.obj", "kule_obok_kula2.obj" ],
  "snow_block_velocities": [
    [ 0, 10, 0 ],
    [ 0, -10, 0 ]
  ]
}
```

i zamienić występującą w niej ścieżkę na odpowiadającą stworzonemu w korku 2 folderowi.

```
},
"base_path": "C:\\projekt_badawczy\\MPM_MC_CUDA\\snowballs_collision\\",
"paths": {
  "result_name": "zderzenie_kul",
  "result_path": "out\\",
  "walls_name": "kule_walls.obj",
  "walls_path": "",
  "snow_objects_paths": [ "", "" ],
  "snow_objects_names": [ "kule_obok_kula1.obj", "kule_obok_kula2.obj" ],
  "snow_block_velocities": [
    [ 0, 10, 0 ],
    [ 0, -10, 0 ]
  ]
}
```

- Po zapisaniu zmian, można uruchomić program przez uruchomienie „MPM_MC_CUDA.exe”.












Pojawi się konsola i wyświetlone zostaną wczytane z pliku config.json parametry.

```
WybierzC:\projekt_badawczy\MPM_MC_CUDA\program\MPM_MC_CUDA.exe
Theta_c: 0.019
Theta_s: 0.0075
xi: 10
rho_0: 100
Modul Younga (E_0): 150000
nu: 0.2
Wsp^czynnik Lamego (mu_0): 62500
Wsp^czynnik Lamego (lambda_0): 41666.7
Krok czasowy (dt): 0.00025
t: 0
t_0: 0
t_end: 50
sticky: 0
alpha: 0.95
particle_diameter: 0.0144
h: 0.03125
MARCHING_CUBES_EPSILON: 1e-06
isolevel: 0.001
step: 10
base_path: C:\projekt_badawczy\MPM_MC_CUDA\snowballs_collision\
result_path: C:\projekt_badawczy\MPM_MC_CUDA\snowballs_collision\out\
result_name: zderzenie_kul
walls_path: C:\projekt_badawczy\MPM_MC_CUDA\snowballs_collision\
walls_name: kule_walls.obj
n: 88713
n: 88713
C:\projekt_badawczy\MPM_MC_CUDA\snowballs_collision\out\
=====
Timestep number: 0
Simulation time: 0.00025
```

W dalszej kolejności prowadzone będą obliczenia i wyświetlane będą informacje dotyczące numeru kroku czasowego, czasu w symulacji, całkowitej masy cząstek w symulacji i w przypadku, gdy w danym kroku czasowym następuje zapis wyników, wyświetlona zostanie ścieżka do pliku, w którym będą one zapisane.

```
WybierzC:\projekt_badawczy\MPM_MC_CUDA\program\MPM_MC_CUDA.exe
Total mass in simulation: 52.9463
=====
Timestep number: 15
Simulation time: 0.004
Total mass in simulation: 52.9463
=====
Timestep number: 16
Simulation time: 0.00425
Total mass in simulation: 52.9463
=====
Timestep number: 17
Simulation time: 0.0045
Total mass in simulation: 52.9463
=====
Timestep number: 18
Simulation time: 0.00475
Total mass in simulation: 52.9463
=====
Timestep number: 19
Simulation time: 0.005
Total mass in simulation: 52.9463
C:\projekt_badawczy\MPM_MC_CUDA\snowballs_collision\out\
=====
Timestep number: 20
Simulation time: 0.00525
Total mass in simulation: 52.9463
=====
Timestep number: 21
Simulation time: 0.0055
Total mass in simulation: 52.9463
```

6. Uzyskane wyniki przechowywane są w wcześniej przygotowanym folderze. Można je poddać dalszej obróbce, na przykład wczytując do programu Blender.

Ten komputer > Dysk lokalny (C:) > projekt_badawczy > MPM_MC_CUDA > snowballs_collision > out				
Nazwa	Data modyfikacji	Typ	Rozmiar	
 zderzenie_kul0.obj	09.03.2025 02:52	3D Object	1 423 KB	
 zderzenie_kul1.obj	09.03.2025 02:52	3D Object	1 424 KB	
 zderzenie_kul2.obj	09.03.2025 02:52	3D Object	1 436 KB	
 zderzenie_kul3.obj	09.03.2025 02:52	3D Object	1 436 KB	
 zderzenie_kul4.obj	09.03.2025 02:52	3D Object	1 436 KB	
 zderzenie_kul5.obj	09.03.2025 02:52	3D Object	1 429 KB	
 zderzenie_kul6.obj	09.03.2025 02:52	3D Object	1 430 KB	
 zderzenie_kul7.obj	09.03.2025 02:53	3D Object	1 435 KB	
 zderzenie_kul8.obj	09.03.2025 02:53	3D Object	1 442 KB	
 zderzenie_kul9.obj	09.03.2025 02:53	3D Object	1 444 KB	
 zderzenie_kul10.obj	09.03.2025 02:53	3D Object	1 434 KB	

7. W obecnej implementacji trójkąty zapisywane są osobno. Z tego względu przed dalszą obróbką należy usunąć powtarzające się krawędzie i punkty. W programie Blender można to wykonać używając funkcji „merge by distance”.

Wytyczne do danych wejściowych

1. Wstęp

Druga część opisuje wymagania dotyczące danych wejściowych dla programu do przeprowadzania symulacji śniegu metodą MPM stworzonego w ramach projektu badawczego „Proceduralne tworzenie efektów atmosferycznych 3D przy użyciu CUDA”.

2. Parametry symulacji

- Czas/krok czasowy

Krok czasowy jest bardzo istotny. Należy go dostosować do liczby klatek, wraz z liczbą kroków czasowych pomijanych przy zapisywaniu siatek wynikowych. Prawie zawsze na pojedynczą klatkę animacji konieczne będzie przeliczenie wielu kroków czasowych.

Obliczenia są prowadzone schematem jawnym, oznacza to, że symulacja jest bardzo wrażliwa na wartość kroku czasowy. W przypadku gdy występuje rozbieżność wyników do nieskończoności, w pierwszej kolejności problemu można szukać w za dużej wartości kroku czasowego.

W tej metodzie numerycznej występuje pewna wersja warunku CFL - należy pamiętać, że zwiększenie rozdzielczości symulacji (więcej cząstek, gęstsza siatka) dla zachowania stabilności będzie wymagało zmniejszenia kroku czasowego.

t_0 – czas początkowy symulacji, domyślnie 0.0

dt – krok czasowy

t_{end} – czas zakończenia symulacji, po osiągnięciu tego czasu program zostanie zamknięty

`step` – liczba kroków czasowych, co którą następuje zapis siatki, jeżeli `step` wynosi 10, to co 10 policzonych kroków czasowych wykonany zostanie zapis, pozwala wraz z krokiem czasowym na dobranie liczby klatek na sekundę.

- Parametry materiałowe

Porównanie parametrów materiałowych wraz z opisem przedstawiono na animacjach znajdujących się w repozytorium w katalogu `examples/material_comparison`.

Odształcenia w materiałach dzielą się na sprężyste i plastyczne. Sprężyste zanikają gdy siła je wymuszająca znika, natomiast plastyczne pozostają. Materiał do pewnego momentu odkształca się sprężysto, po przekroczeniu pewnej granicy odkształca się plastycznie (trwale). Parametry materiałowe są związane z mechaniką ośrodków ciągłych i teorią sprężystości-plastyczności.

`theta_c` – opisuje jak mocno można ścisnąć sprężysto, przed przejściem do zakresu plastycznego

`theta_s` – opisuje jak mocno można rozciągnąć sprężysto przed przejściem do zakresu plastycznego

`xi` – utwardzenie pod wpływem zmniejszenia objętości

`E_0` – moduł Younga, opisuje naprężenia (w praktyce przyrost siły) w materiale pod wpływem rozciągania lub ściskania w zakresie sprężystym

`nu` – liczba Poissona; w mocnym uproszczeniu: jak będziemy rozciągać, to jak mocno będzie się zmniejszał przekrój, albo jak się ma sprężystość poprzeczna do podłużnej), nie powinna wymagać zmiany

`sticky` – spowolnienie prędkości stycznej przy kolizji cząstki z ścianą, mniejsza wartość oznacza, że ruch będzie szybciej zanikał po kontakcie ze ścianą, domyślnie 0.9.

- Parametry symulacji

`alpha` – parametr odpowiedzialny za stosunek w jakim połączone są schematy numeryczne FLIP i PIC, domyślna wartość 0,95 nie powinna wymagać zmian

- Średnica cząstki (particle diameter), rozdzielczość siatki (hh)

W testach dobrze sprawdzało się dobranie rozdzielczości siatki tak, aby w komórkach siatki, w których znajdują się cząstki w komórce były około 4 cząstki. W testach dobrze wypadł stosunek średnicy cząstki do rozdzielczości siatki wynoszący 0,5.

- Isolevel metody Marching Cubes

Od tego parametru zależy, jak duży obszar wokół cząstek będzie tworzył siatkę trójkątów. W przypadku, gdy będzie brakować trójkątów w siatce należy go zmniejszyć. Jeśli śnieg ma być bardziej „puszysty” należy ten parametr zmniejszyć, jeśli ma być bardziej „chudy”, skupiony bliżej cząstek, należy ten parametr zwiększyć.

- Wczytywane elementy

`base_path` – ścieżka katalogu, z którego wczytywane są siatki

`result_name` – nazwa plików wynikowych

`result_path` – ścieżka katalogu, do którego zapisywane będą wyniki, podawana względem `base_path`

`walls_path` – ścieżka katalogu, z którego wczytane są siatki ścian, podawana względem `base_path`

`walls_name` – nazwa pliku zawierającego ściany symulacji

`snow_objects_paths` – ścieżka katalogów, z których wczytywane są siatki brył mających być wypełnionymi śniegiem i umieszczonych na scenie

`snow_objects_names` – nazwy plików, z których wczytywane są siatki brył mających być wypełnionymi śniegiem i umieszczonych na scenie

`snow_block_velocities` – wektory prędkości początkowych dla wczytanych brył

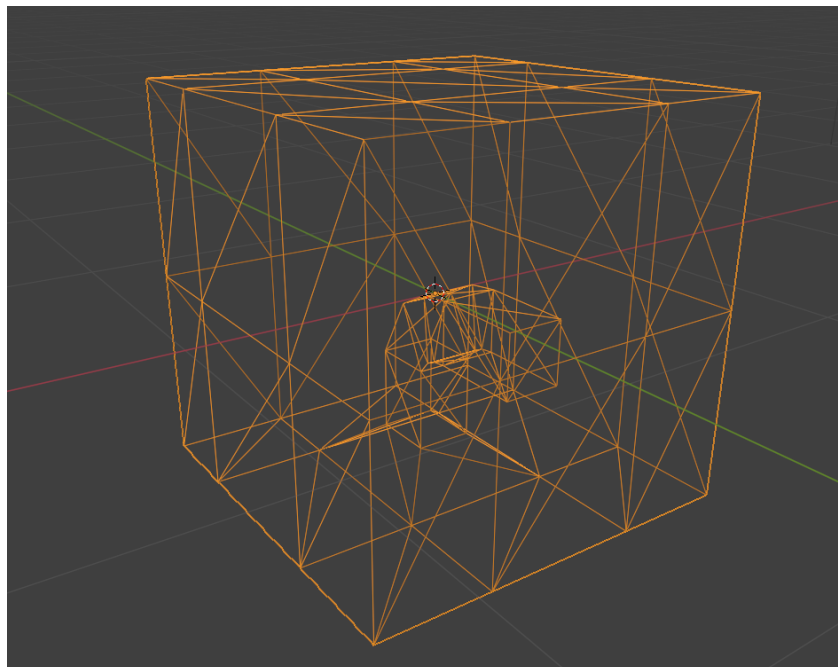
`snow_block_rotation` – wektory początkowych prędkości kątowych dla wczytanych brył

`snow_block_rotation` – wartości gęstości cząstek dla wczytanych brył

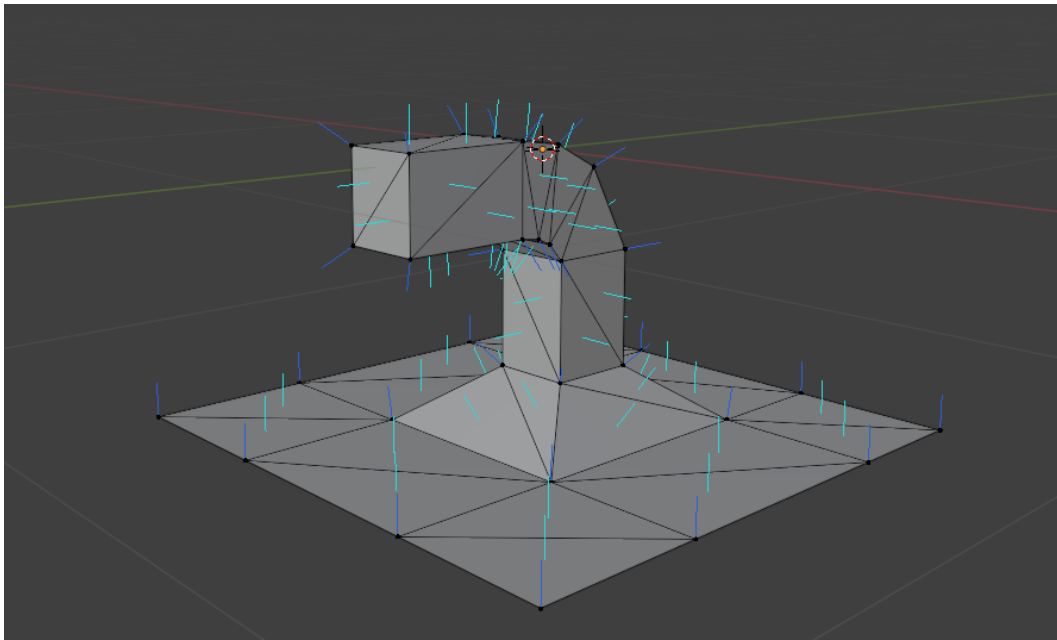
3. Wczytywane siatki

Pliki zawierające wczytywane siatki mają format `.obj`. Wczytywane siatki dzielą się na siatki ścian oraz siatki wyznaczające, w których miejscach na scenie ma się znajdować śnieg.

Siatka ścian stanowi jednocześnie bounding box symulacji. Przykładowa siatka wygląda następująco:



Jest to sześcian, z którego podstawy wystaje pewna geometria.



Ważne jest, aby wektory normalne powierzchni ścian skierowane były do wnętrza obszaru obliczeniowego.

Podobnie dla siatek wewnątrz których ma zostać zamieszczony śnieg. Ich wektory normalne muszą być skierowane w stronę po której ma się pojawiać śnieg (np. dla kuli domyślnie do wewnątrz).