Analiza skalowalnośći kodu metodą Lattice Boltzman, wpływ kernela oraz kraty na efektywność obliczeń

Rybski Arkadiusz

2020

Spis treści

1 Wprowadzenie

1.1 Metoda Lattice Boltzmann

Metoda Lattice Boltzmann jest metodą numeryczną służącą do rozwiązywania równań z zakresu mechaniki płynów. Metoda kratowa Boltzmanna oparta jest na równaniu Boltmanna.

rownanie Boltzmnanna

Okazuje się, że w skali makroskopowej z równań Boltzmanna łatwo można przejść na równania Naviera-Stokesa. Niestety nie rozwiązuje to problemu analitycznego rozwiązania równania. Natomiast okazuję się, iż mimo skomplikowanej formy, równanie Boltzmana w prosty sposób można zaimplementować. W ten sposób możemy otrzymać równanie kratowe Boltzmanna.

Boltzmann Lattice equation

Rodzaje krat

Ze względu na to, że funkcja dystrybucji jest zależna nie tylko od czasu, położenia ale też i prędkości do implementacji potrzebujemy siatki zawierającej nie tylko położenie geometryczne, ale każdy węzeł musi zawierać prędkości. W literaturze przyjęto następujące oznaczenia krat.

$$D_n Q_m$$

gdzie n oznacza ilość wymiarów, natmoiast m oznacza ilość możliwych kierunków prędkości.
Przykładowe kraty:

Algorytm

Zasada działaania metody kratowej Boltzmanna oparta jest na dwóch fazach: propagacji i kolizji.

Faza kolizji Równanie

equation of colision

Faza propagacji

$$equation\ of\ stagnation$$

+odniesienia do literatury
Całość mechanizmu można podsumować w nastepujących krokach.

1. Wybór lokalizacji
2. Rejestracja informacji o nadchodzących cząsteczkach
3. Kolizja
4. Dystrybucja po kolizji
5. Wybór kolejnej lokalizacji

Rodzaje kernela

Przedstawiony w powyższym alogrytmie operator kolizji BGK(Bhatnagar-Gross-Krook) to jeden z wieluTmozliwych operatorów kolizji. Inne z nich to MRT(multiple relaxation time) czy SRT(single relaxation time).

1.2 Skalowalność kodu

Dlaczego ważna jest skalowalność? W celu zbadania efektywności obliczeń równoległych wprowadźmy wielkość zwana dalej przyspieszeniem *(z ang. speedup)*

$$speedup = \frac{t_1}{t_N}$$

, *gdzie* t_1 *oznacza czas wykonanaia procesu przy uyciu* 1 *procesora*,
t_N *oznacza czas wykonania procesu przy uyciu* N *procesorw*.
W idealnym przypadku wykres $speedup(N)$ byłby wykresem liniowym.

Silne skalowanie

Na czym polega, jakis prosty przyklad +prawo Amdahl'a

Słabe skalowanie

Analoigcznie jw.

2 Analiza

2.1 Cel analizy

Co dokładnie było badane, tzn skalowalnosc kodu W jakim celiu to było badane. jakie może to dać nam korzysci(informacja o tym jakie symulacje sa efektywe), ewentualnie nad czym pracowac by uefektywnić metode

2.2 Opis klastrów obliczeniowych

Specyfikacje na czym pracowaliśmy.

2.3 Badane modele

Po prostu, dać tu infrmacje jakie symulacje byly prowadzone. Rozmiary siatek, rodzaje kernela, rodzaj kraty uporzadkowane zeby latwo bylo odczytac.

kraty

siatki

3 Wyniki

3.1 Prezentacja wyników

tutaj wykresy najpierw, slabe skalowanie, potem silne

3.2 Analiza wynikow

Jak dziala skalowanie, a potem sprobowac cos wnioskowac, z zaleznosci tej powierzchni przeplywu informacji do rozmiarow siatki

4 Wnioski

Co dziala dobrze, co dziala zle.