Mini-projet du cours de programmation GPU

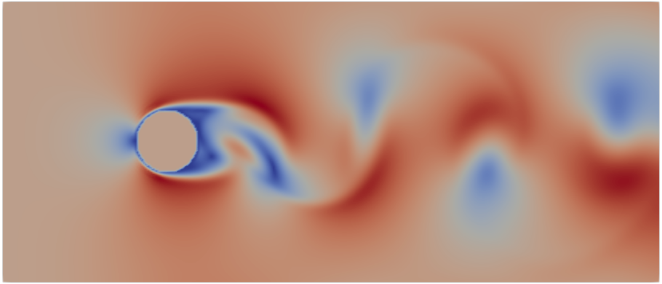
Pierre KESTENER

janvier 2024

CEA
Master spécialisé HPC-IA, Mines Paritech, Sophia Antipolis

Lattice Boltzmann Method

- LBM (Lattice Boltzmann Method) : fluid flow simulation
- alternative to other numerical fluid simulation methode (finite difference, finite volume, spectral, etc..)
- instead of solving macroscopic Navier-Stokes equations, solves **Boltzmann equation** : i.e. look at the evolution of microscopic particules using a **statistical description**, and deduce fluid macroscopic characteristics from that.



Kinetic theory of gases

- Kinetic theory of gases
- distribution function $f(\overrightarrow{\mathbf{r}}, \overrightarrow{\mathbf{c}}, t)$, avec $\overrightarrow{\mathbf{r}} \in \mathbb{R}^3$ and $\overrightarrow{\mathbf{c}} \in \mathbb{R}^3$
- interpretation of $f(\overrightarrow{\mathbf{r}}, \overrightarrow{\mathbf{c}}, t)\,\mathrm{d}\overrightarrow{\mathbf{r}}\,\mathrm{d}\overrightarrow{\mathbf{c}}$: number of particules elementary volume $\mathrm{d}\overrightarrow{\mathbf{r}}$ for which velocity is in range $\left[\overrightarrow{\mathbf{c}}, \overrightarrow{\mathbf{c}} + \mathrm{d}\overrightarrow{\mathbf{c}}\right[$
- **usual macroscopic hydrodynamics variables** (density, velocity, energy, ...) are defined as statistical moments of the distribution functions :
  - **density** : $\rho(\overrightarrow{\mathbf{r}}, t) = \int f(\overrightarrow{\mathbf{r}}, \overrightarrow{\mathbf{c}}, t)\,\mathrm{d}\overrightarrow{\mathbf{c}}$
  - **momentum** : $\overrightarrow{\mathbf{p}}(\overrightarrow{\mathbf{r}}, t) = \int f(\overrightarrow{\mathbf{r}}, \overrightarrow{\mathbf{c}}, t)\overrightarrow{\mathbf{c}}\,\mathrm{d}\overrightarrow{\mathbf{c}} = \rho\overrightarrow{\mathbf{v}}$
  - **kinetic energy** $e_{kin}(\overrightarrow{\mathbf{r}}, t) = \int f(\overrightarrow{\mathbf{r}}, \overrightarrow{\mathbf{c}}, t)||\overrightarrow{\mathbf{c}}||^2\,\mathrm{d}\overrightarrow{\mathbf{c}} = \rho\overrightarrow{\mathbf{v}}$
- if one knows how to evolve f in time, when fluid dynamics can be obtained $\Rightarrow$ Boltzmann equation

Boltzmann Equation

Boltzmann equation is transport equation which describes how particule distribution function evolves in time

$$\frac{\partial f}{\partial t} + \overrightarrow{\mathbf{c}} \cdot \nabla_{\overrightarrow{\mathbf{r}}} f + \overrightarrow{\mathbf{F}} \cdot \frac{\partial f}{\partial \overrightarrow{\mathbf{c}}} = \left(\frac{\partial f}{\partial t} \right)_{\mathrm{coll}}$$

Lattice Boltzmann Method

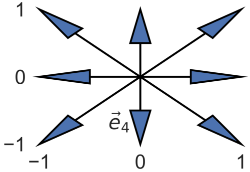
- **Problem :** to perform numerical simulation of Boltzmann equation, one needs to discretize $f(\overrightarrow{\mathbf{r}}, \overrightarrow{\mathbf{c}}, t)$ knowing that $\overrightarrow{\mathbf{r}} \in \mathbb{R}^3$ and $\overrightarrow{\mathbf{c}} \in \mathbb{R}^3 \Rightarrow$ the total number of variables to hold in memory is gigantic ($\sim N^6$)
  example : $N = 100 \Rightarrow N^6 = 10^{12}$ i.e. 8 TBytes of RAM just for storing f at one given time, this is **unrealisable** in practice
- **Solution :**
  - discretize space (variable $\overrightarrow{\mathbf{r}}$ using a N^3 grid
  - velocity $\overrightarrow{\mathbf{c}}$ using only a small number of discrete values (a few units or a few ten's)
  - $\Rightarrow$ use a discrete set of distribution functions $f_n(i, j, k, t)$ where (i, j, k) are the space index, and n is the velocity index.
- This approach is called **Lattice Boltzmann method**, which can provides in the end numerical simulations in good agreement with experimental data.
  main advantages of LBM : (very) easy to implement in software and easy to parallelize.

Lattice Boltzmann Method

In a simplified manner, one can say the time evolution of distribution functions (or more precisely the evolution of the q componets of f) is computed by iterating the following two operators :

- collision : $f_i(\vec{x}, t + \delta_t) = f_i(\vec{x}, t) + \frac{f_i^{eq}(\vec{x},t) - f_i(\vec{x},t)}{\tau_f}$, avec $0 \leq i < q$
  the collision term is modelling a relaxation to equilibrium distribution
- streaming (or transport) : $f_i(\vec{x} + \vec{e}_i, t + 1) = f_i(\vec{x}, t)$
  particules modelled by f_i are moving in direction e_i

Illustrating D2Q9 LBM :
9 possible velocities



Mini-project

Goal : Either using the serial python version, either C++ serial version, implement a parallel version of LBM using a parallel programing model of your choice.

- Chose parallel programing model :
  - numba (or Cupy) for parallelizing the serial python version
  - CUDA/C++ or OpenACC or Kokkos
- Provide a video of 12 to 15 minutes reporting your work (code presentation, parallel strategy, kernels) a performance study : one can for example measure a speedup curve (ration of the serial version time over parallel version time).