Multi GPU, Interactive 3D Simulator for the Lattice Boltzmann Immersed Boundary Method

Bob Zigon

Beckman Coulter
Indianapolis, Indiana

April 7, 2016
Outline

- Lattice Boltzmann Method
- Immersed Boundary Method
- OpenMP & GPU program structure
- Visualization with Modern OpenGL
- Results
- Future Work
Lattice Boltzmann (LB) algorithm is a way to solve the Navier Stokes equations by considering the behavior of collections of fluid particles instead of individual particles.
Lattice Boltzmann (LB) algorithm is a way to solve the Navier Stokes equations by considering the behavior of collections of fluid particles instead of individual particles.

\[
f_i(x + e_i, t + 1) = f_i(x, t)
\]

(1) \textit{Streaming}

\(f_i(x, t)\) - defines the particle distribution in \(1\) of \(N\) discrete directions.

\(e_i\) - the direction vectors that enable streaming to adjacent cells.
Figure 1: 9 direction vectors $e_0$ through $e_8$ for 2D domain

Figure 2: 19 direction vectors for 3D domain
To complete the LB model, a term is added that describes the inevitable collision that occurs as particles move from cell to cell.

\[
f_i(x + e_i, t + 1) = f_i(x, t) - \frac{[f_i(x, t) - f_{i}^{eq}(x, t)]}{\tau} \]

(2)

\(f_{i}^{eq}(x, t)\) - describes the distribution at equilibrium.

\(\tau\) - describes the rate at which the collision term goes to zero.
To complete the LB model, a term is added that describes the inevitable collision that occurs as particles move from cell to cell.

$$f_i(x + e_i, t + 1) = f_i(x, t) - \frac{[f_i(x, t) - f_i^{eq}(x, t)]}{\tau}$$

(2)

- $f_i^{eq}(x, t)$ - describes the distribution at equilibrium.
- $\tau$ - describes the rate at which the collision term goes to zero.

Simple, elegant and embarrassingly parallel.
Immersed Boundary (IB) method is a way of simulating the interaction of a flexible structure and a fluid.

Examples

1. Fish swimming in the ocean
2. Red blood cells moving through arteries.
Immersed Boundary (IB) method is a way of simulating the interaction of a flexible structure and a fluid.

Examples

1. Fish swimming in the ocean
2. Red blood cells moving through arteries.

\[
f_i(x + e_i, t + 1) = f_i(x, t) - \left[ f_i(x, t) - f_i^\text{eq}(x, t) \right] \frac{\tau}{\tau} - w_i \cdot g(x, t) \tag{3}
\]

- \( g(x, t) \) - describes the force of the boundary on the fluid
- \( w_i \) - direction dependent weights
Immersed Boundary (IB) method is a way of simulating the interaction of a flexible structure and a fluid.

Examples

1. Fish swimming in the ocean
2. Red blood cells moving through arteries.

\[ f_i(x + e_i, t + 1) = f_i(x, t) - \frac{[f_i(x, t) - f_i^{eq}(x, t)]}{\tau} - w_i \cdot g(x, t) \]  \hspace{1cm} (3)

- \( g(x, t) \) - describes the force of the boundary on the fluid
- \( w_i \) - direction dependent weights

Simple, elegant and embarrassingly parallel.
Immersed boundaries are objects represented by collections of fibers. Fibers are composed of points and segments.

Figure 3: A fiber, its points $p_1, p_2, p_3, p_4$, and segments $s_1, s_2, s_3$
Segments can be stretched like a spring. We model the force using

$$S(\Delta x) = -K_s \Delta x.$$  \hspace{1cm} (4)
Segments can be stretched like a spring. We model the force using

$$S(\Delta x) = -K_s \Delta x.$$  \hspace{1cm} (4)

Segments can be bent. We model the force using $1 / \text{curvature}$ of the parameterized fiber path $p(x)$

$$B(x) = K_b \frac{|p''(x)|}{|1 + p'^2(x)|}$$ \hspace{1cm} (5)

and simplify to

$$B(x) = K_b |p''(x)|$$ \hspace{1cm} (6)

because we assume that $p'(x) \approx 0$. 
Now, the total boundary force, $g(\text{fiber}, t)$, is just

$$g(\text{fiber}, t) = \sum_{j=1}^{K} S(\text{fiber}_j) + B(\text{fiber}_j)$$

(7)

with

$$S(\text{fiber}_j) = \sum_{i=1}^{N} S(\text{segment}_{j,i})$$

(8)

and

$$B(\text{fiber}_j) = \sum_{i=1}^{N} B(\text{segment}_{j,i}).$$

(9)

Simple, elegant and embarrassingly parallel.
Figure 4: Cartoon example of the demo video
Video demo of the application running.
While not done
{
    Compute field at time $T_n$ on Tesla K40c$^1$
    
    Visualize field at time $T_n$ on Quadro K5000$^2$
    
    $n = n + 1$
}

---

$^1$CUDA 7.5, Driver 355.85, W7/64, Visual Studio 2013

$^2$OpenGL 4.2, Vertex and Fragment Shaders
While not done
{
    Compute field at time $T_n$
    1. Compute IB forces
    2. LB Stream and Collide
    3. Compute boundary conditions
    4. Update fluid density, vel
    5. Move the fibers
    6. Compute inlet & outlet boundary conditions

    Visualize field at time $T_n$

    $n = n + 1$
}

While not done
{
    Compute field at time $T_n$
    1. Compute IB forces 2%
    2. LB Stream and Collide 60%
    3. Compute boundary conditions 1%
    4. Update fluid density & velocity 15%
    5. Move the fibers 1%
    6. Compute inlet & outlet bc’s 10%

    Visualize field at time $T_n$ 5%

    $n = n + 1$ 6%
}

$^3$Domain size $256^3$
Visualization with Modern OpenGL

Two api’s – Legacy and Modern

Legacy
1. Easy to use
2. Lots of examples
3. Not well suited to current GPU’s
4. Our custom shaders 10x faster than Legacy
Visualization with Modern OpenGL

Two api’s – Legacy and Modern

Legacy
1. Easy to use
2. Lots of examples
3. Not well suited to current GPU’s

Modern
1. Difficult to use
2. Poor examples
3. Extremely well suited to current GPU’s
4. Our custom shaders 10x faster than Legacy
Figure 5: Execution times for 104 fibers and 103 points/fiber

---

4 Lenovo D30, 8 E5-2609@2.4GHz, 32GB RAM, W7/64
Future Work-Static complex internal boundary

MS Paint example... ...with 2D simulation.
Future Work - Red Blood Cells

Insert lots of Red Blood Cells with lots of fibers... 

...into this 3D simulation for interactive manipulation.
Acknowledgements

Professor Luoding Zhu, Indiana University-Purdue University of Indianapolis
robert.zigon@beckman.com