Interactive Simulation of Generalised Newtonian Fluids using GPUs

Somay Jain, Nitish Tripathi and P J Narayanan
IIIT Hyderabad, India

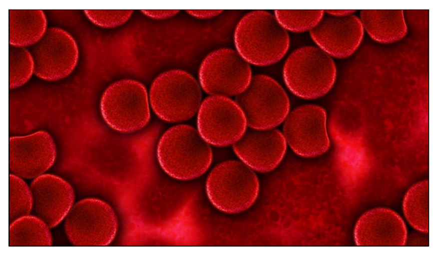
Goal

- To interactively simulate and visualise Generalised Newtonian Fluids (GNF) using GPUs.
- Simulate Newtonian and non-Newtonian fluids using a common framework in realtime for reasonable domain sizes.
- Demonstrate the potential to scale to larger domain sizes using MultiGPU implementation.

Many Applications

Movies

Video Games



Medical Research

Related Work

Lattice Boltzmann Method

(Ando et al. [SIGGRAPH'13], Thuerey et al. [SIGGRAPH'05], Thuerey et al. [Proceedings of Vision, Modeling and Visualization'06], Chen et al. [Annual Review of Fluid Mechanics'98])

- Newtonian fluids simulation
- Method on different grid types (tetrahedral and adaptive)

Non-Newtonian Fluids (Modelling and Simulation)

(Phillips et al. [IMA Journal of Applied Mathematics'11], Boyd et al. [Journal of Physics A: Mathematical and General], Desbrun et al. [EGCAS'96])

- Non-Newtonian fluid models
  - Cross, Carreau, Ellis Models etc.
- Viscoelastic fluid simulation using conventional methods.

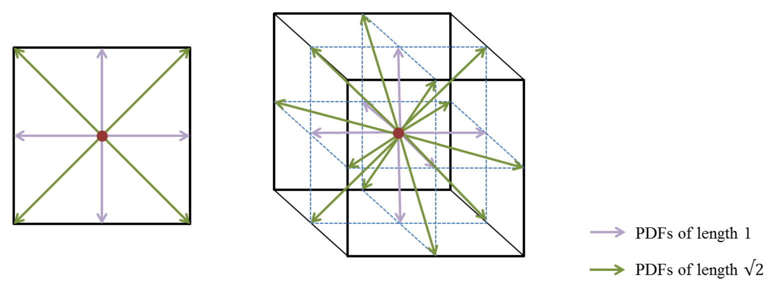
Lattice Boltzmann Method on GPUs

(Januszewski et al. [Computer Physics Communications'14], Schreiber et al. [Procedia Computer Science'11])

- Multi-component and Free Surface flows on single and multi-GPUs.

Our approach

- Lattice Boltzmann Method (LBM) using D3Q19 grid
  - A mesoscopic approach - particles (logical in nature) collide at grid centers and progress to neighbours in fixed directions
- Truncated Power Law to calculate viscosity for non-Newtonian fluids
- Marching Cubes for visualisation of the fluid
- Exploit the inherent parallelism of LBM with an efficient memory access pattern to create a fast GPU implementation



D2Q9 Lattice D3Q19 Lattice

Vector	Direction
$\mathbf{e_0}$	$(0,0,0)'$
$\mathbf{e_{1,2}}$	$(\pm1,0,0)'$
$\mathbf{e_{3,4}}$	$(0,\pm1,0)'$
$\mathbf{e_{5,6}}$	$(0,0,\pm1)'$
$\mathbf{e_{7\ldots10}}$	$(\pm1,\pm1,0)'$
$\mathbf{e_{11\ldots14}}$	$(0,\pm1,\pm1)'$
$\mathbf{e_{15\ldots18}}$	$(\pm1,0,\pm1)'$

Velocity vectors for D3Q19

Basic LBM

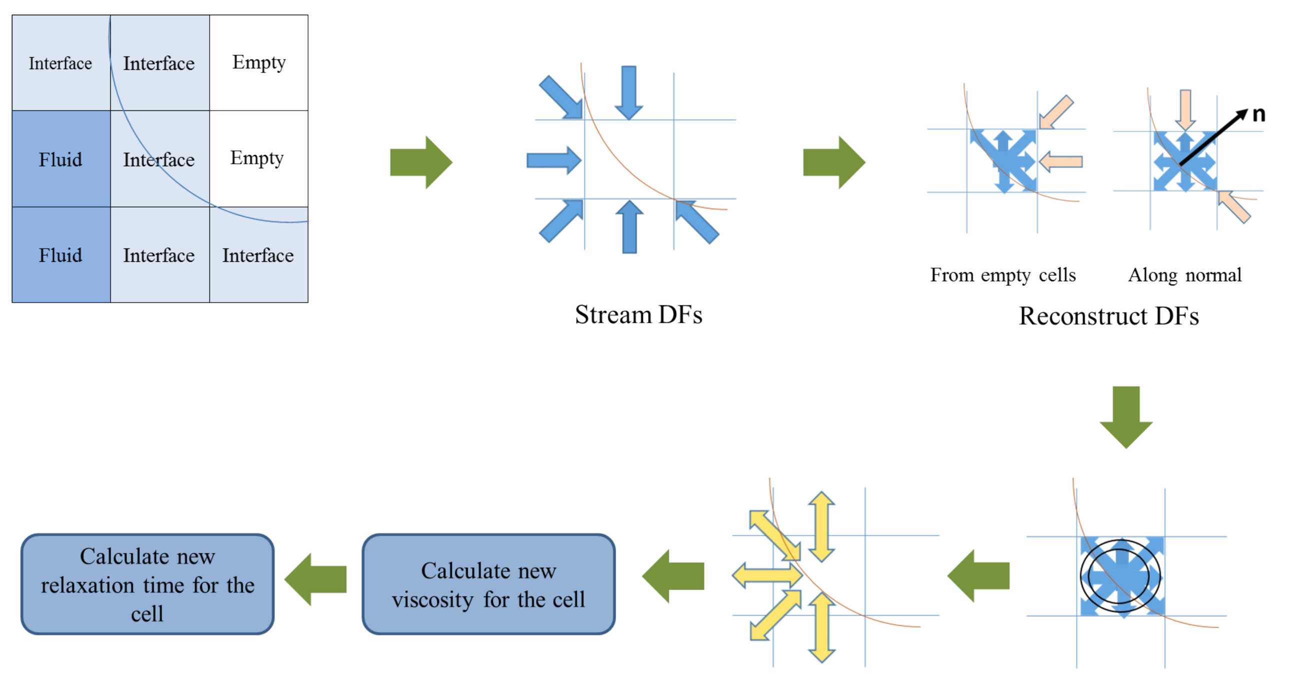
- Streaming - Read and update neighbours' distribution function (DF) for corresponding directions
- Collision - Calculate density and velocity for each cell, collide them and update the DFs -

$$\rho = \sum df_i \qquad \mathbf{u} = \sum df_i \cdot \mathbf{e_i}$$

$$df_i^{eq}(\rho, \mathbf{u}) = w_i \left(\rho - \frac{3}{2}\mathbf{u}^2 + 3\mathbf{e_i} \cdot \mathbf{u} + \frac{9}{2}(\mathbf{e_i} \cdot \mathbf{u})^2 \right)$$

$$df_i = (1 - \omega) df_i + \omega df_i^{eq}$$

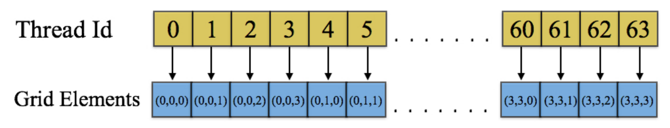
Free Surface LBM



Parallel Implementation using CUDA

Data	Size	Use
Previous DFs	19 floats	Previous iteration distribution function
Current DFs	19 floats	Current iteration distribution function
Previous State	1 int	Type of cell in previous iteration
Current State	1 int	Type of cell in current iteration
Epsilon	1 float	Intermediate, visualisation purposes
Velocity	3 floats	Intermediate, visualisation purposes

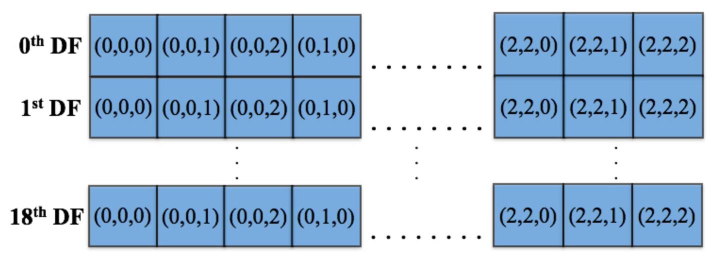
Data Requirement for each cell



Thread Mapping with Grid Elements

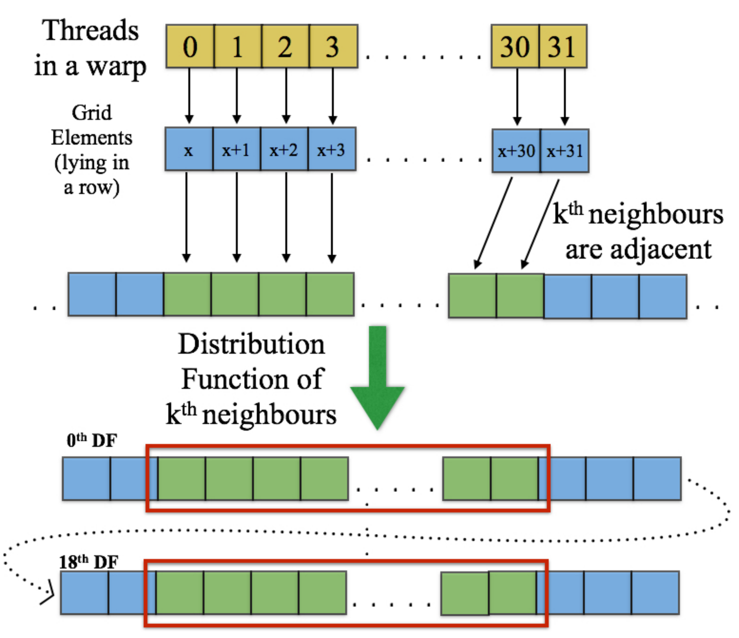
- Data stored in global memory
- One thread per cell

- Data for 3D grid stored linearly as Structure of Arrays in a 1D array in row major format



DF Layout for a 3^3 Grid, stored in row major format

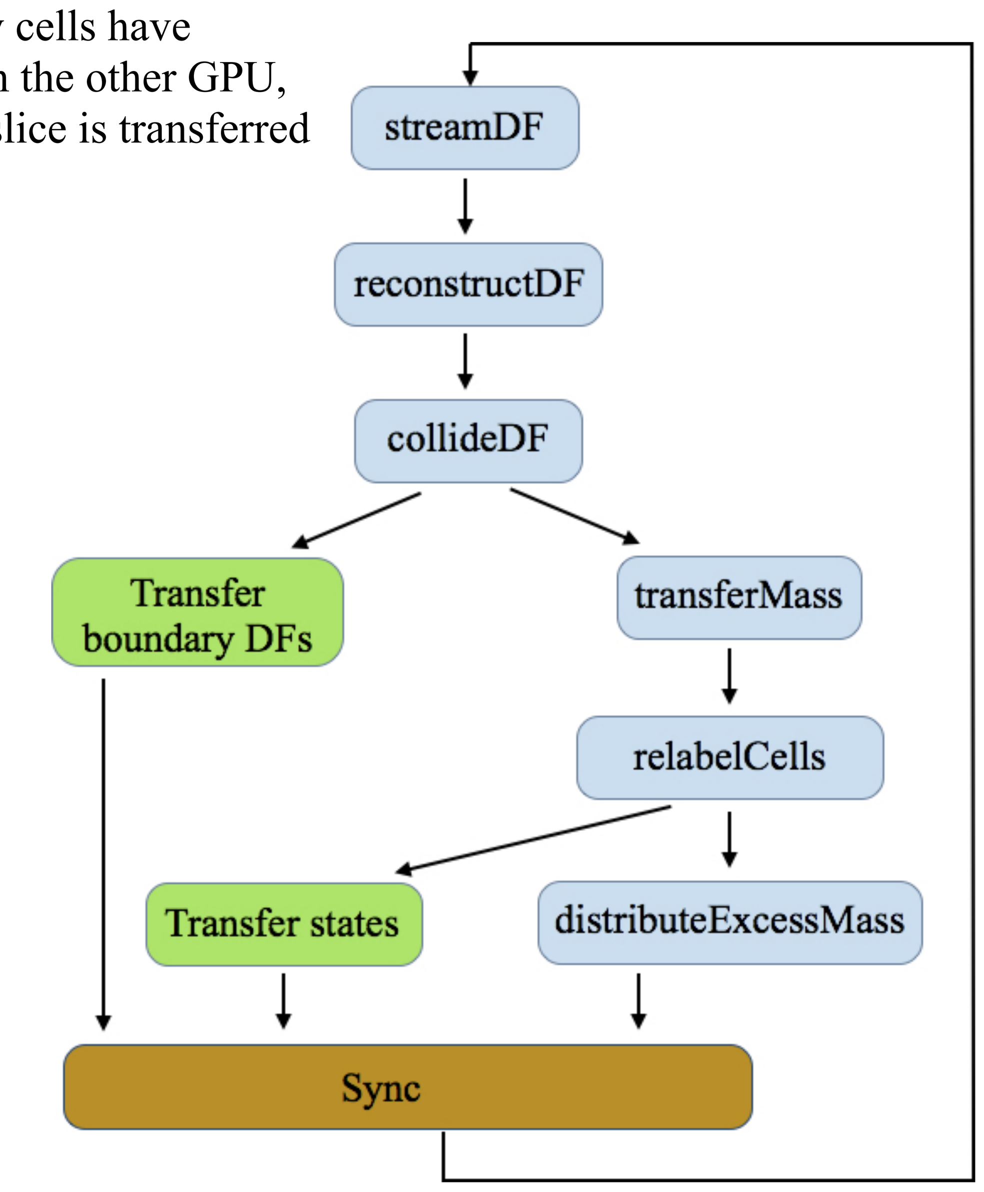
- Threads in a warp read/update the DF for a particular direction simultaneously.
- These fully coalesced because adjacent threads map to horizontally adjacent cells
- 75-100% kernel occupancy for such accesses



DFs for kth neighbours of adjacent cells

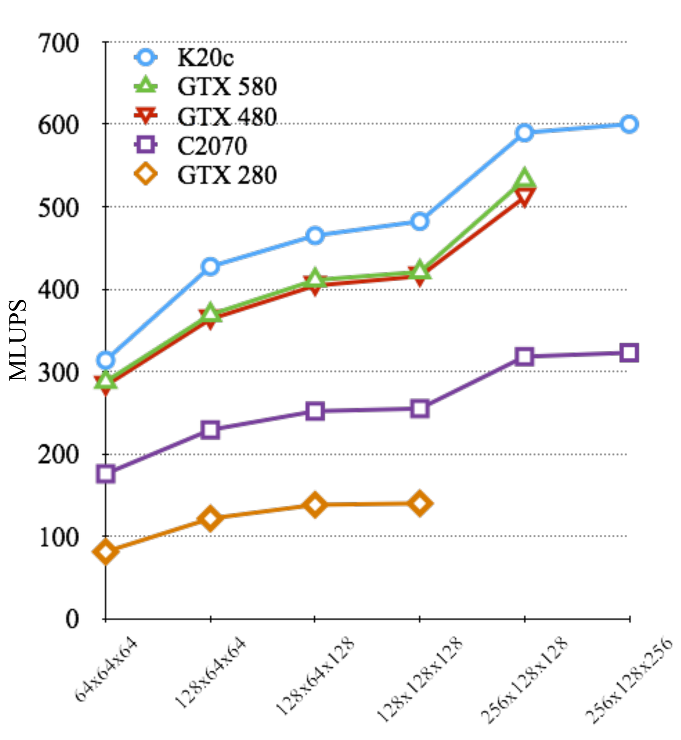
Multiple GPUs

- Data divided into 2 parts by slicing the grid along the z-axis
- The boundary cells have neighbours on the other GPU, so boundary slice is transferred

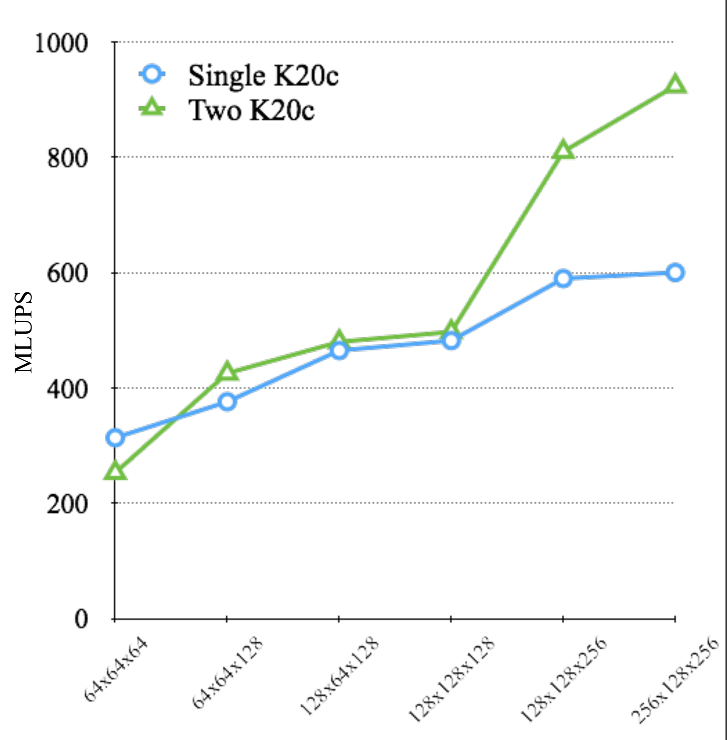


Overlap of data transfers with computation

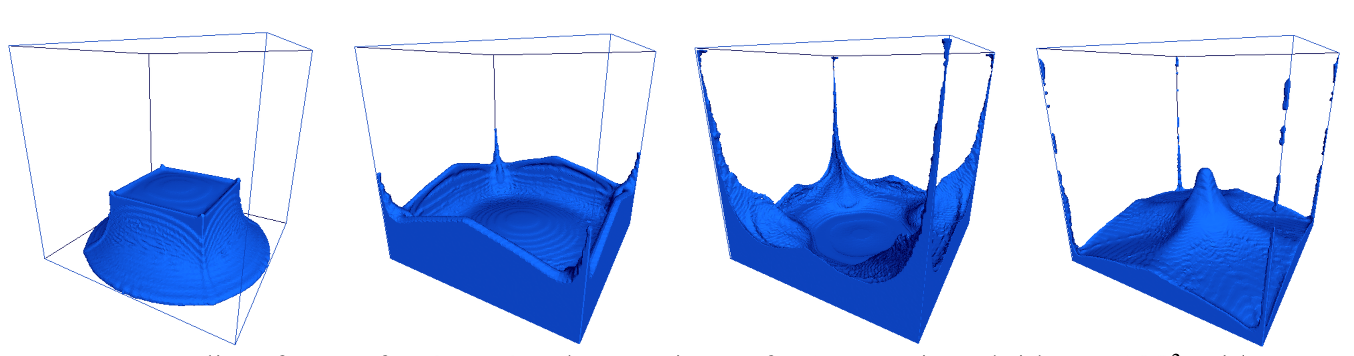
Results



Performance of the Dam Break Experiment on various GPUs

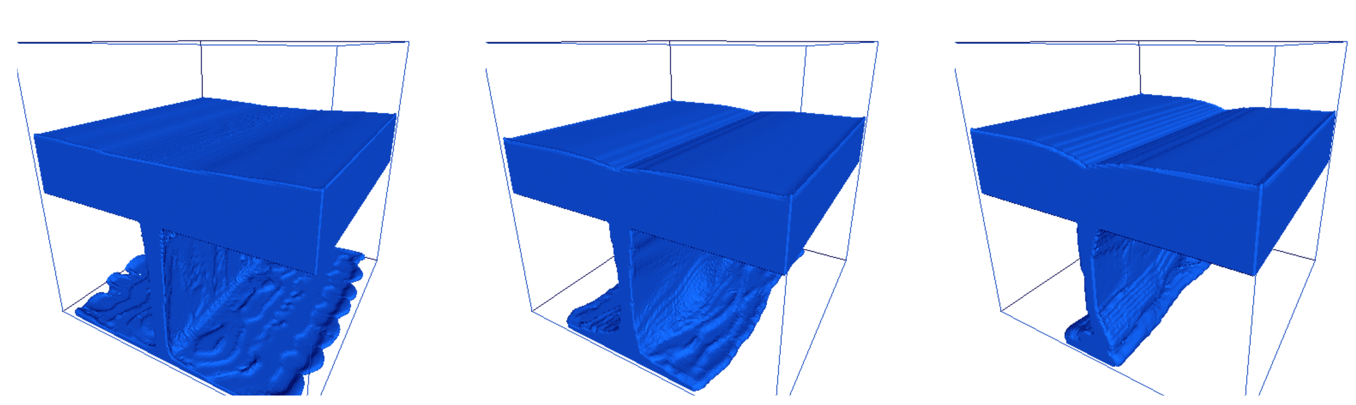


Performance of the Dam Break Experiment on single and multi-GPUs



Intermediate frames for Dam Break Experiment for Newtonian Fluid on a 128^3 grid, running at an average of 5 fps with 50 LBM iterations per frame

Non-Newtonian Characteristics

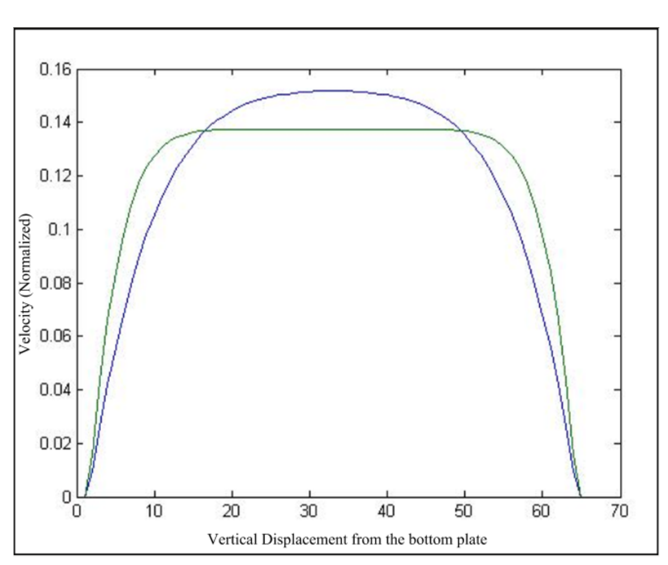


Shear Thinning
More fluidity
(decrease in viscosity)
upon impact

Newtonian
No change in viscosity
upon impact

Shear Thickening
Folding on itself
(increase in viscosity)
upon impact

- Newtonian fluid curve follows a parabolic path
- Shear thinning fluid curve flattens on approaching the center of the channel.



Normalised velocity profiles for Newtonian & shear thinning fluids

Summary

- 500 MLUPS ~ 5fps for 128^3 grid on a single GPU
- 350 MLUPS ~ 25fps for 64^3 grid on a single GPU
- Upto 900 MLUPS using two K20c GPUs
- Similar performance for Newtonian and non-Newtonian fluids

Future Work

- We have dealt with laminar flows in this work. A study on turbulent fluids using LBM is an interesting area for future work
- Visual quality of the simulations can be enhanced using ray-tracing
- Simulation of out of core grids (512^3 and above) is another interesting area for future work

Center for Visual Information Technology
International Institute of Information Technology, Hyderabad, India