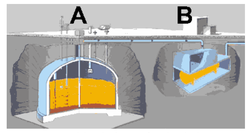
# Implementation of Parallel Computing for Multiphase Flows using the Lattice Boltzmann Method

*Jaime Mudrich (DOE Fellow), Rinaldo G. Galdamez (DOE Fellow), and Seckin Gokaltun, Ph.D.*
*Applied Research Center, Florida International University, Miami, FL*

## Introduction

- 53 million gallons of radioactive waste at Hanford site
- Stored in leaking single shell tanks (SST)
- Double shell tanks (DST) introduced in 1968
- Unlike the SSTs, DSTs show no leaking
- Waste is being transported from SSTs to DSTs
- Transport of heterogeneous waste clogs piping
- Pulsed-air mixing used to "stir" heterogeneous material
- LBM simulates bubbles rising to predict mixing

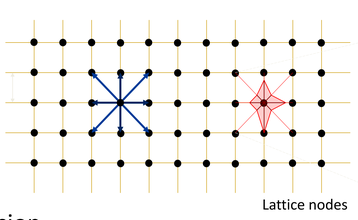


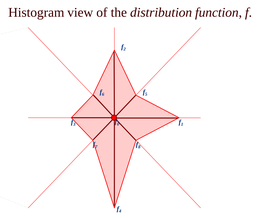
SST (A), and DST (B)


Rising bubbles mix slurry

## Methodology

- Lattice Boltzmann method is based on the Boltzmann transport equation
- Domain is discretized with lattice nodes instead of rigorous meshing
- Independence from mesh allows for complex domains like porous media
- Masses at nodes collide and then stream information to neighbors


Lattice nodes




### Collision

$$f_a^t(\mathbf{x},t) = f_a(\mathbf{x},t) - \frac{\left[f_a(\mathbf{x},t) - f_a^{eq}(\mathbf{x},t)\right]}{\tau}$$

...where...

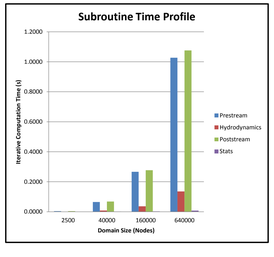
$$f_a^{eq}(\mathbf{x}) = w_a \rho(\mathbf{x})\left[1 + 3\frac{\mathbf{e_a}\bullet\mathbf{u}}{c^2} + \frac{9}{2}\frac{(\mathbf{e_a}\bullet\mathbf{u})^2}{c^4} - \frac{3}{2}\frac{\mathbf{u}^2}{c^2}\right]$$

### Stream

$$f_a(\mathbf{x}+\mathbf{e_a}\Delta t, t+\Delta t) = f_a^t(\mathbf{x},t)$$

| | |
|---|---|
| $\mathbf{x}$ | = position of particles |
| $\mathbf{u}$ | = macroscopic velocity at the node |
| $w_a$ | = constant, direction-specific weight |
| $c$ | = model speed of sound |
| $\tau$ | = dimensionless relaxation time |
| $\mathbf{e_a}$ | = basis vector at node (9 total) |

## Time Profile for Serial LBM Code




- For the multiphase simulations that are being studied, the iterative algorithm is comprised of three steps
- Diagnostics performed to identify sluggish areas
- "Hydrodynamics", "Prestream", and "Poststream" will benefit the most from parallelization
- These subroutines will be split amongst various processors to share the load, speeding up the solution

## Parallel Processing Background

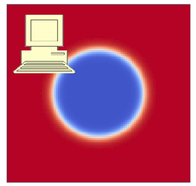
- **Master** processor divides the problem domain amongst multiple **slaves**
- Message passing interface (MPI) allows CPUs to bridge information across sub domains
- Reduction of processing time is ultimately limited by communications between processors and the components of the program that must run sequentially
- Effectiveness of parallelization is measured by **speedup**, *S(N)*, for *N* processors
- When increasing the number of CPUs shows minimal performance increase, optimal quantity has been reached

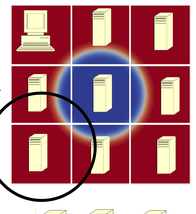
*Amdahl's Law :* $S(N) = \frac{T(1)}{T(N)}$

T(1) = Single processor computation time

T(N) = Multiple processor computation time

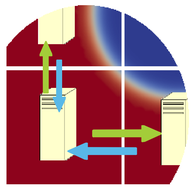
In the serial configuration, only one processor is used to solve the entire domain of the problem



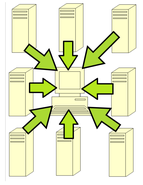


In the parallel configuration, multiple processors split the domain of the problem, reducing overall computation time

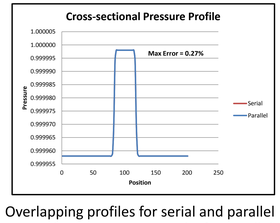
Processors communicate with their neighbors through MPI to "patch" the sub domains



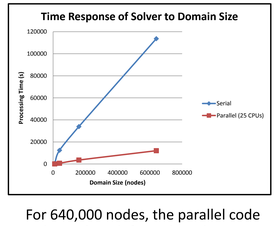
Finally, the master collects the results from the various slave nodes



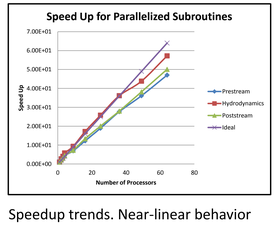
## Validation of the Parallel LBM Code



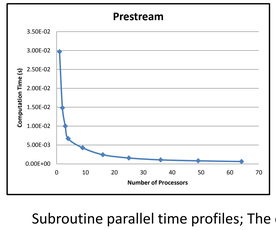
Overlapping profiles for serial and parallel case indicates accurate results for parallel code

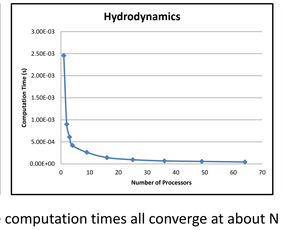


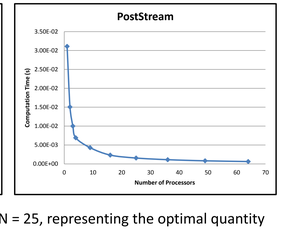
For 640,000 nodes, the parallel code reduces the job from thirty hours to only three hours



Speedup trends. Near-linear behavior confirms correct parallelization



Subroutine parallel time profiles; The computation times all converge at about N = 25, representing the optimal quantity

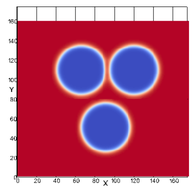
## Results

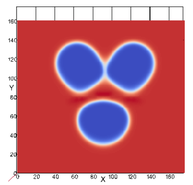
- Study performed on static bubble with a density ratio of 1,000 and a uniform initial pressure distribution
- The solution was checked against Laplace's law for surface tension described below

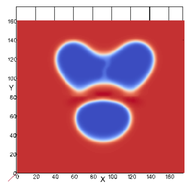
*Laplace's Law :* $\Delta P = \frac{\sigma}{R}$

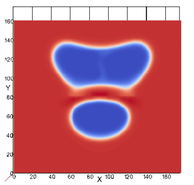
| | |
|---|---|
| $\sigma$ | = Surface tension |
| $\Delta P$ | = Pressure difference across fluid interface |
| $R$ | = Radius of bubble |

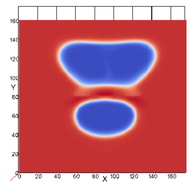
- Parallelization allows for more simulations to performed in a much shorter time
- Using the parallel code and the experimentally determined optimal quantity of processors (N = 25) the following simulation was performed

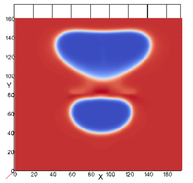


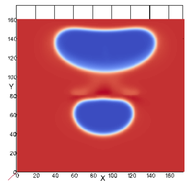



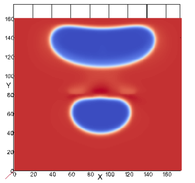




This series illustrates a case of three equal radius bubbles with minimal separation. LBM captures the coalescence of the top bubbles. Density ratio = 100, Interface width = 5 lattice units, vertical acceleration = -2.0 x $10^{-7}$ lattice units per lattice time squared, Interfacial tension = 0.1, and relaxation time for both fluids = 2.71 x $10^{-2}$

## Conclusions and Future Work

- Parallelization with the optimal number of processors results in significant savings in computer time (10 times for N=25 and 640,000 lattice nodes)
- Parallelization allows for simulation of larger domains or longer times
- Future work will include extension of the code from 2D to 3D
- In addition, fluid-solid interactions will be also implemented

## Acknowledgements

This research was supported by the U.S. Department of Energy through the DOE-FIU Science and Technology Workforce Development Program, under grant No. DE-EM0000598.

Special thanks to Leonel Lagos, Ph.D., PMP ®, Director of the DOE-FIU Science and Technology Workforce Development Program