A Parallel Lattice Boltzmann Algorithm for Multicomponent Fluid Flow in Complex Geometries

John Hagedorn
Delphine Goujon
Nicos Martys
Judith Devaney
National Institute of Standards and Technology

Purpose

To model the flow of multiple fluids through porous media such as rock, concrete, and soil.

Applications:

- oil recovery
- service life of materials
- spread of hazardous waste

Approach

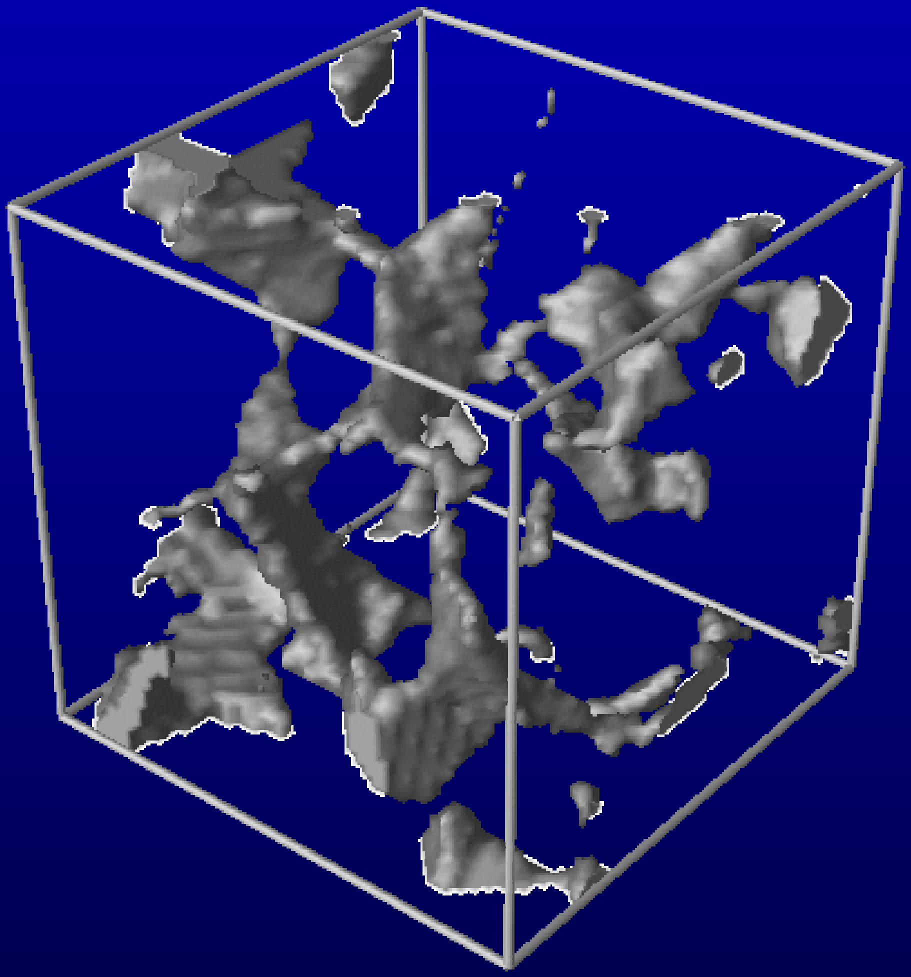
A lattice Boltzmann (LB) method is used.

At each cell the fluid mass and velocity information is represented.

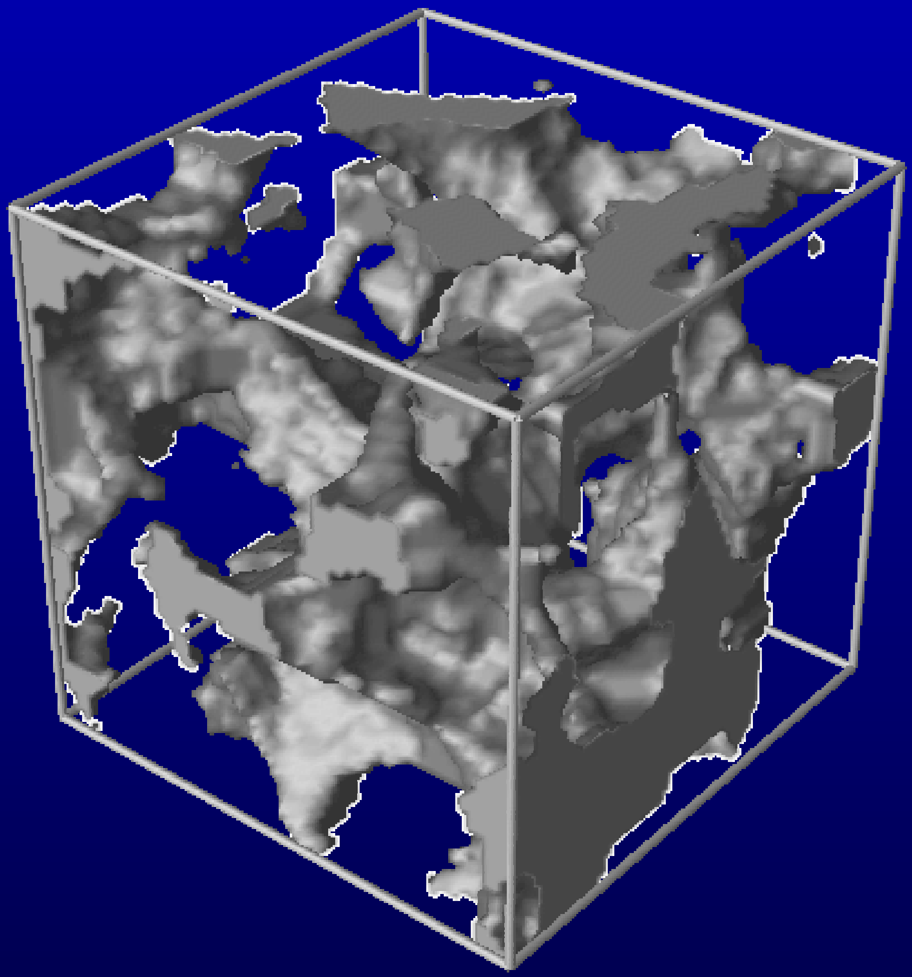
At each iteration, mass is redistributed to neighboring cells and velocities change based on:

- interactions between fluids
- interactions between fluids and solid walls
- an external force (gravity)

Example media: sandstone



7.5% porosity



22% porosity

Boundary Conditions

X & Y : mirror boundary condition

Z : periodic boundary condition

Problem

The LB method is extremely memory and compute intensive.

Solution

Restructure memory representation to reduce memory usage

Parallelize

Memory Usage

The initial version of the algorithm stores 19 floats for each fluid at each site. Furthermore, a second data volume was used to hold intermediate results.

So the total memory required is:

$$N^3 \cdot 2 \cdot (\#\ \text{fluids}) \cdot 19 \cdot \text{sizeof(float)}$$

For a 256^3 system and two fluid components, we require approximately 4.8 gigabytes.

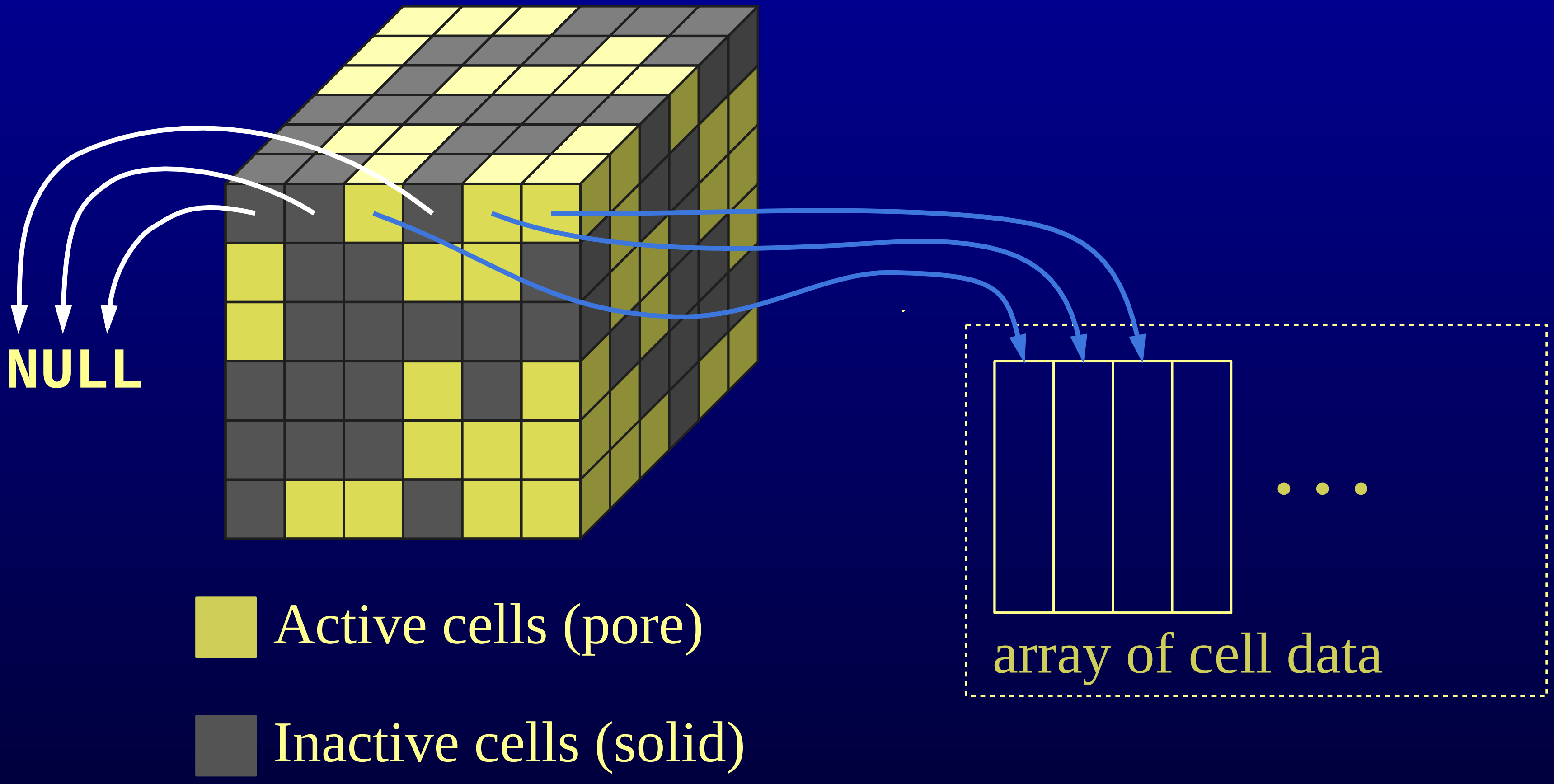
Memory Optimization

- Store data only at active sites: yields large savings based on the porosity of the medium.
- Use a 3 plane circular buffer for intermediate results: eliminates need for an entire second data volume.

Memory Optimization (continued)

Each active cell points to a structure in an array.

Each inactive cell points to NULL.



🟨 Active cells (pore)

⬛ Inactive cells (solid)

Memory Optimization (continued)

Now the memory requirement is:

$$(N^3+3N^2) \cdot (\text{sizeof(pointer)} + \rho\cdot(19\cdot\#\text{fluids})\cdot\text{sizeof(float)})$$

where ρ is the porosity of the medium.

For a 256^3 system with two fluid components and 10% porosity, this requires approximately 311 megabytes.

This is a savings of about 93%.

Memory Optimization (continued)

Additional large memory savings can be achieved by making simplifying assumptions on the LB equations.

These assumptions enable us to reduce the data at each cell to 3+(#fluids) floats.

For a two fluid model, this is a savings of over 86% of the memory devoted to the active cells.

Parallelization

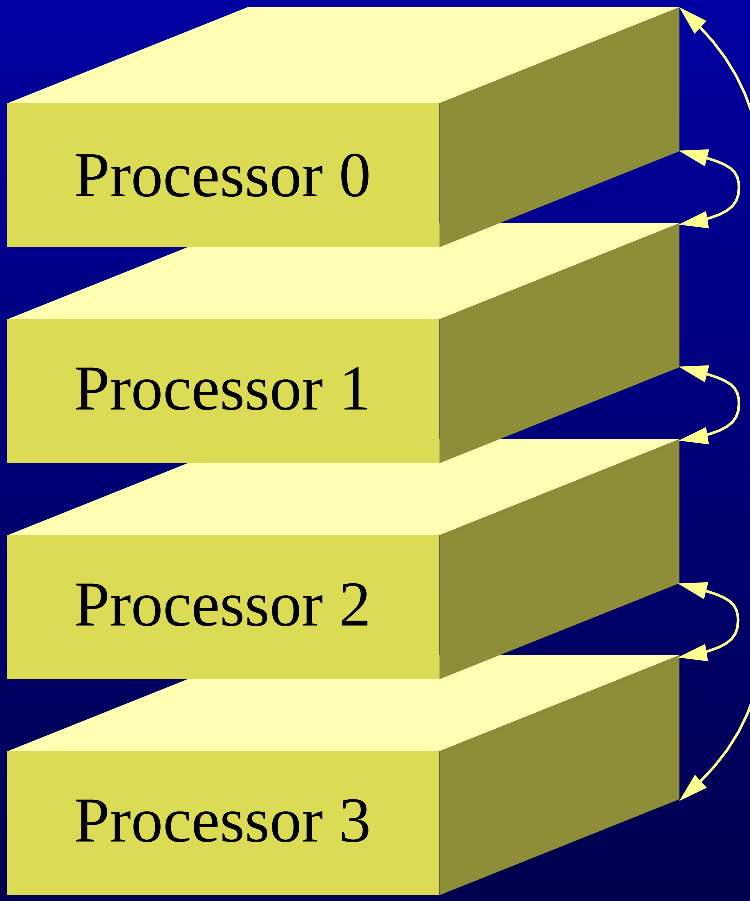
Needed to improve performance

The nearest−neighbor dependency suggested that the algorithm would be well suited to parallelization.

We used:

- MPI
- SPMD programming model

Parallelization (continued)

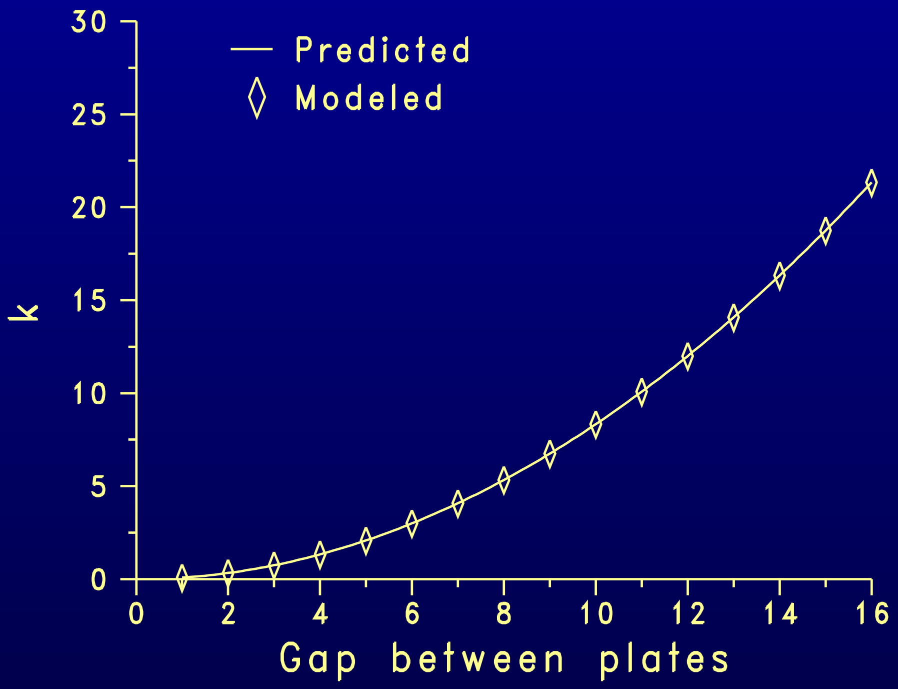


The data volume is divided into blocks along the Z axis. Each block is assigned to a processor.

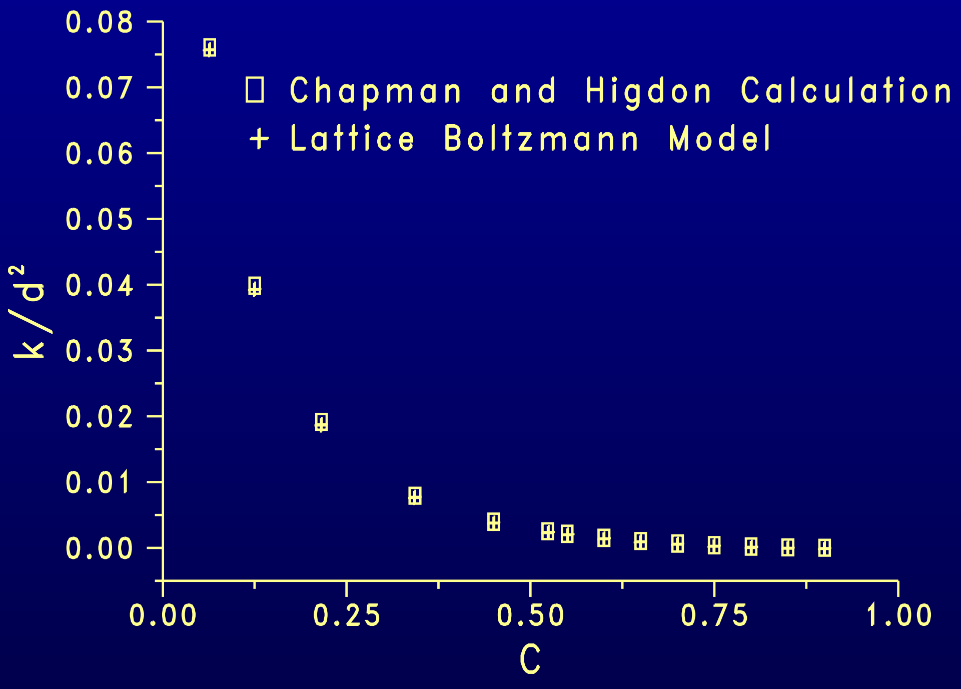
At the end of each iteration, partial results from the bordering planes are exchanged between adjacent blocks.

Numerical Tests

The algorithm was run on several types of "ideal" media such as parallel plates, and overlapping spheres:

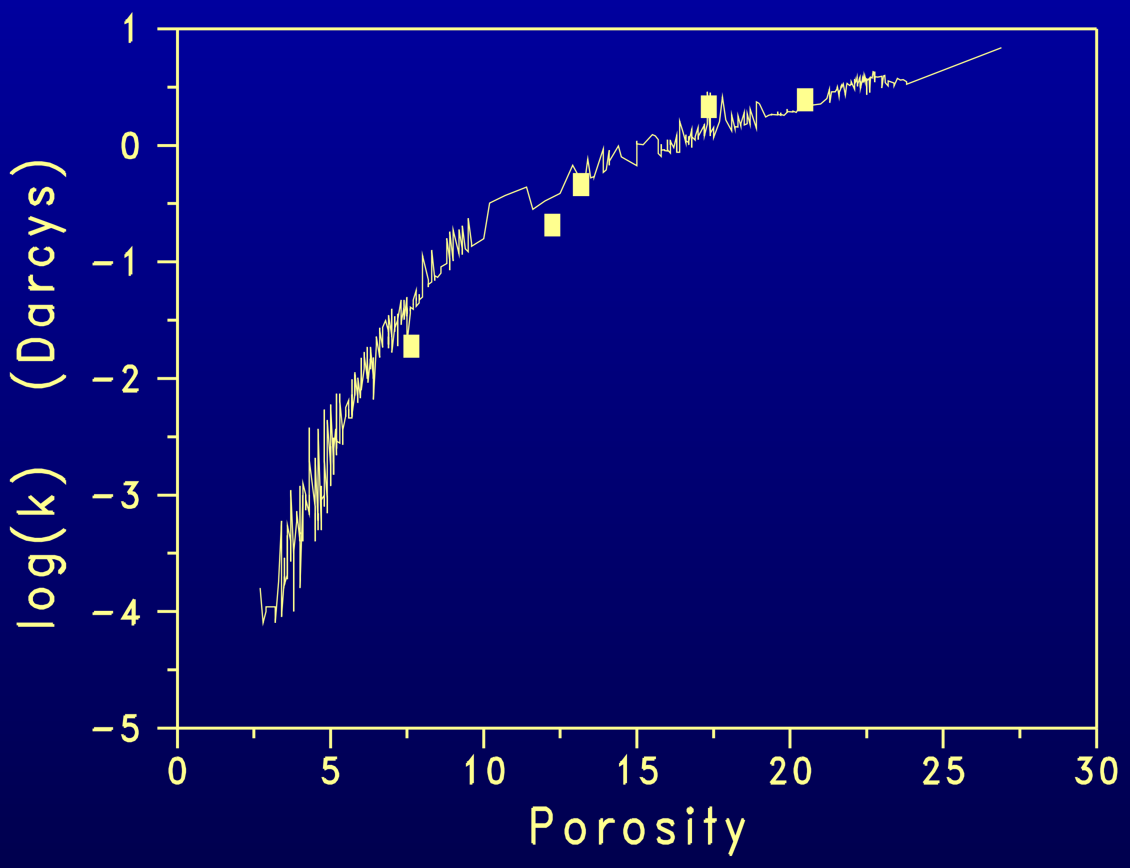


Flow through parallel plates.



Flow through spheres centered on cubic lattice

Comparison with Experimental Data



Measured and modeled permeabilites of Fontainebleau sandstone media.

Performance Results

We have run performance tests on an IBM SP2 and an SGI Onyx. The performance data closely agree with a very simple model:

$$T = P / N + S$$

Where:

T is the total time for one iteration
P is the time for the parallelizable computation
N is the number of processors
S is the time for the non−parallelizable computation

Performance Results (continued)

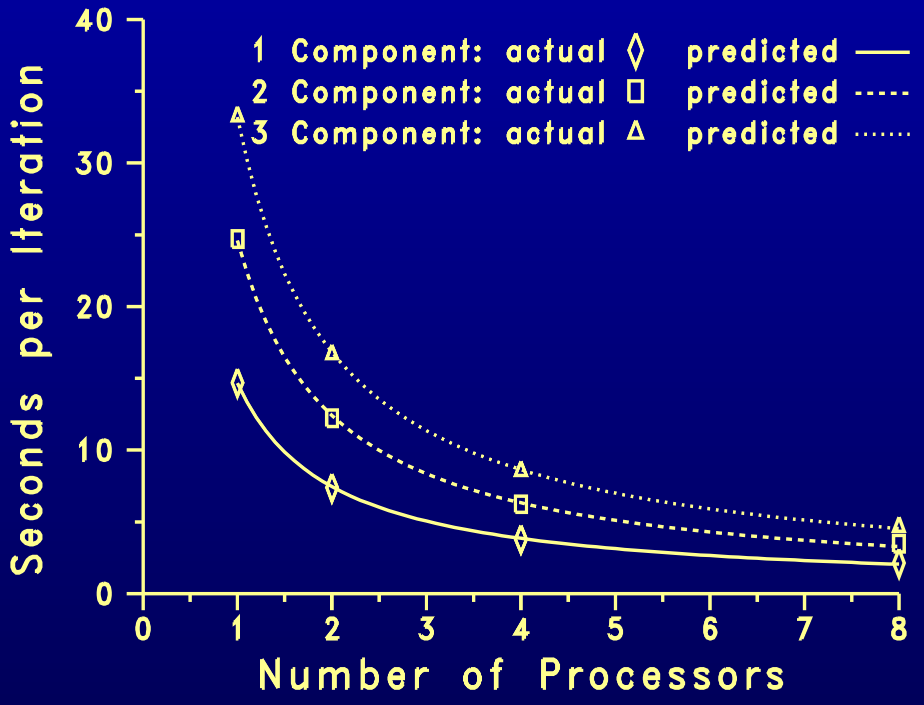
For a two−component, 128x128x256 system, the measured performance closely matched:

$$T = 24.36 / N + .237$$ seconds on the SGI Onyx

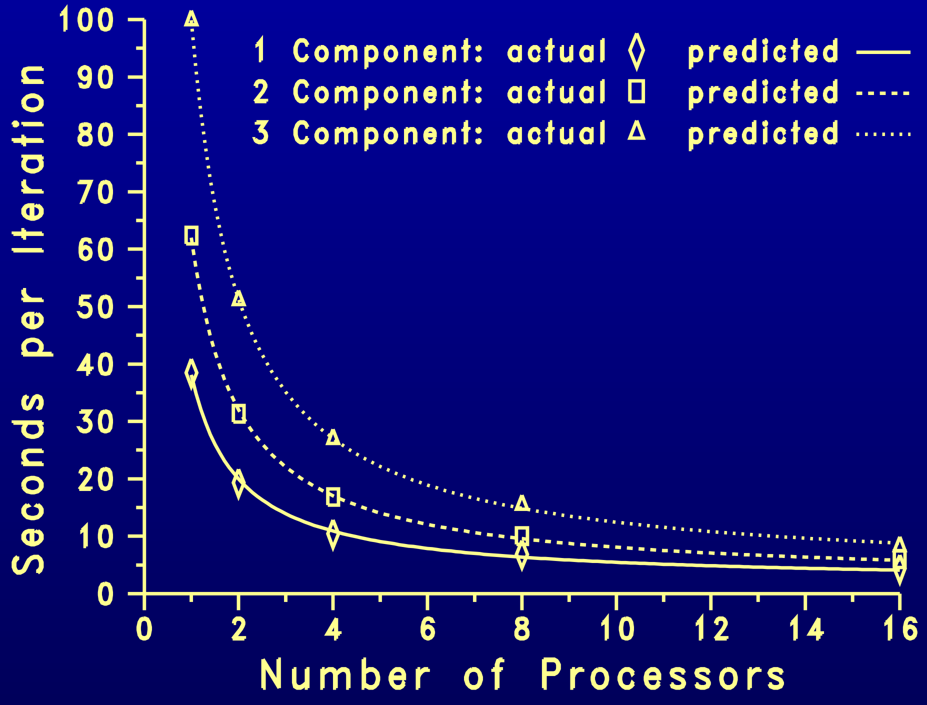
and

$$T = 59.92 / N + 2.09$$ seconds on the IBM SP2

Performance Results (continued)



SGI Onyx



IBM SP2

Future Work

- More tests on real−world media
- Additional performance studies
- Tests on very large systems
- Investigation of impact of data layout on performance
- Refinement of the model

Conclusion

- This LB model has proven accurate and effective.
- Through some simple restructuring of data, we have obtained large memory reductions.
- The model parallelized very well, enabling us to run system sizes that were previously not feasible.

We have an effective tool for investigating multi−fluid flow through complex porous media.