

OpenACC EuroHack15

Team: FluidFlow
U.S. Naval Academy
CDR Stu Blair (PI), Prof. Carl Albing,
Beau Bayless, Mark Schnabel
Mentors: Alex Grund (U. of Dresden),
Andreas Jocksch(CSCS)

* no pointers were used/harmed in the making of this presentation

Lattice Boltzmann Method

Fluid Flow simulation

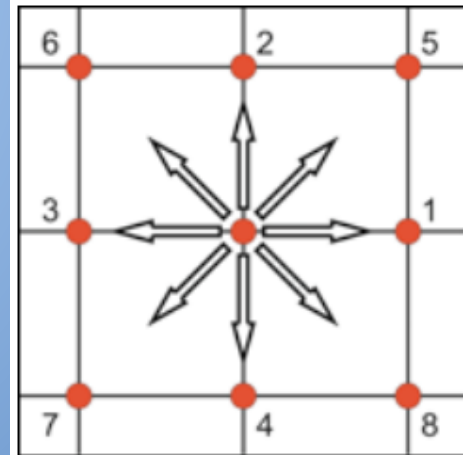
Avoids the expense of PDEs

Cellular Automata-based techniques

Cell states

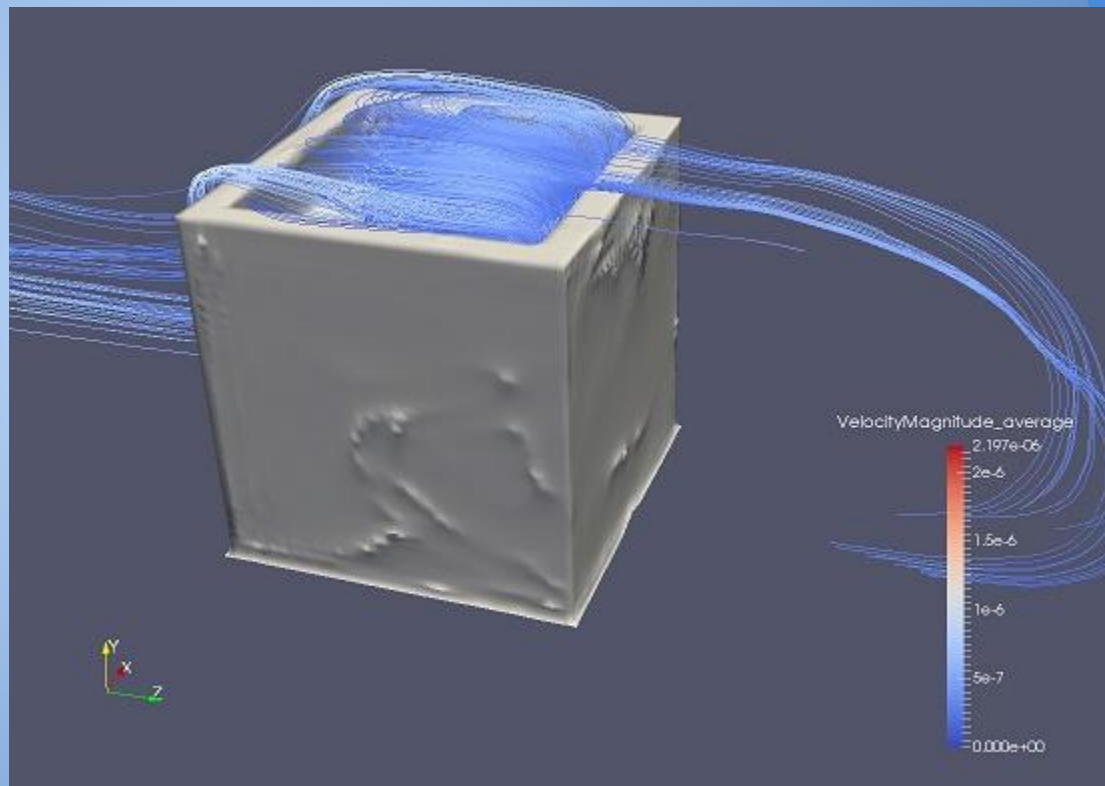
not a single Boolean

a set of floating point numbers

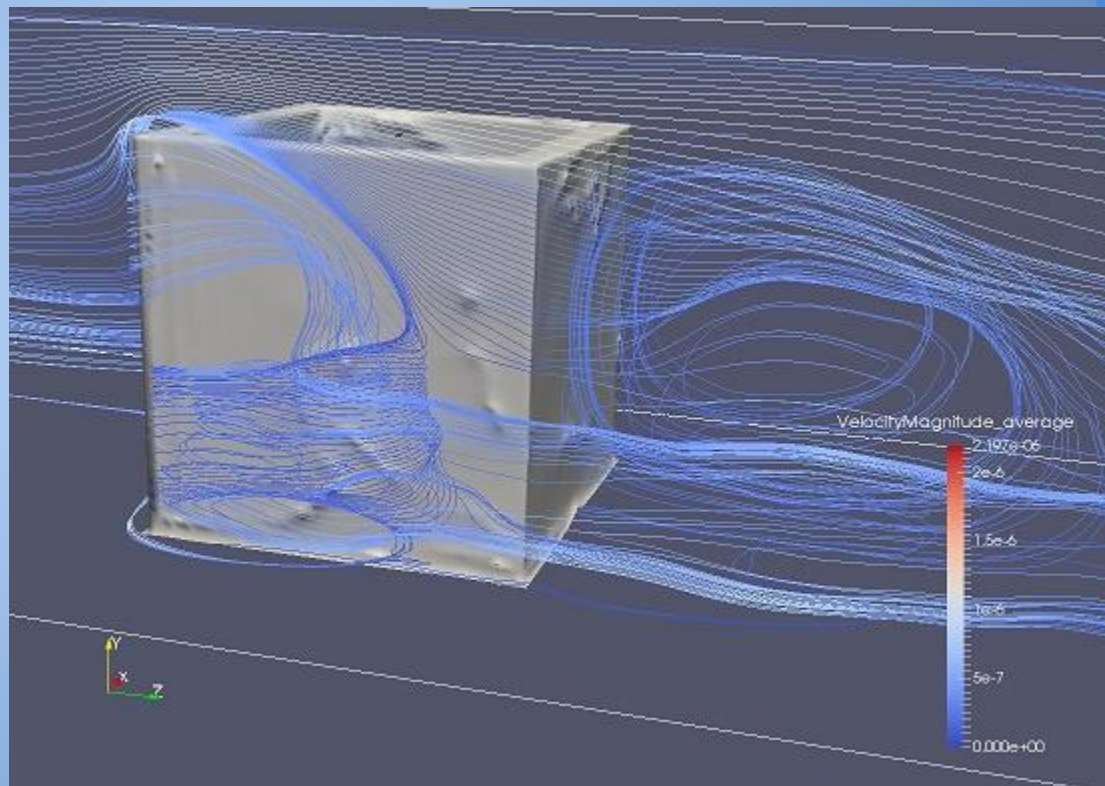


Examples

stream tracer - cut view



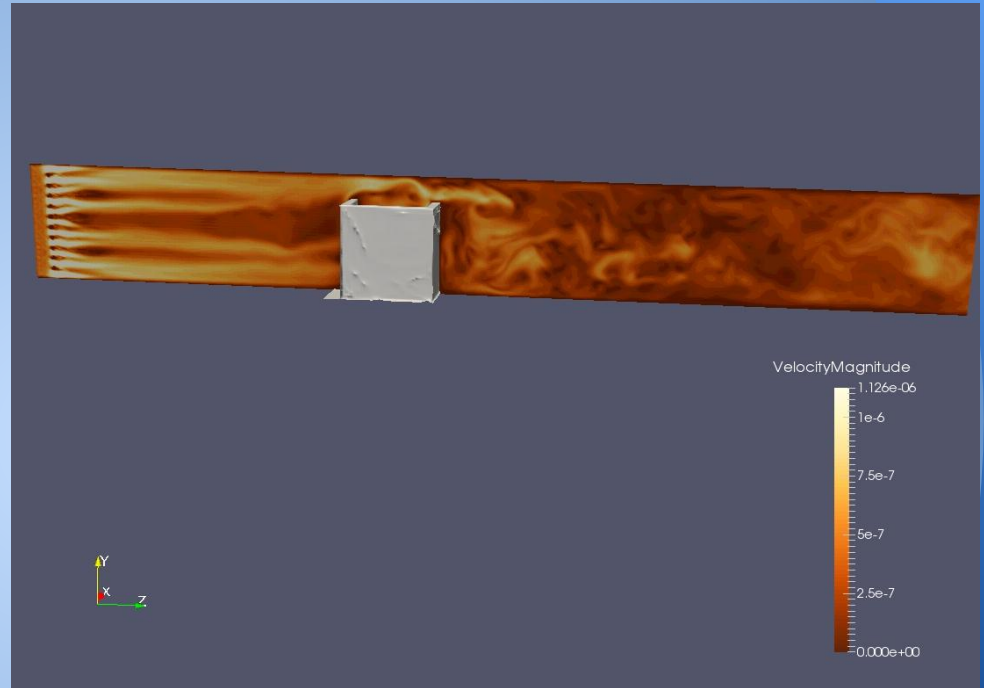
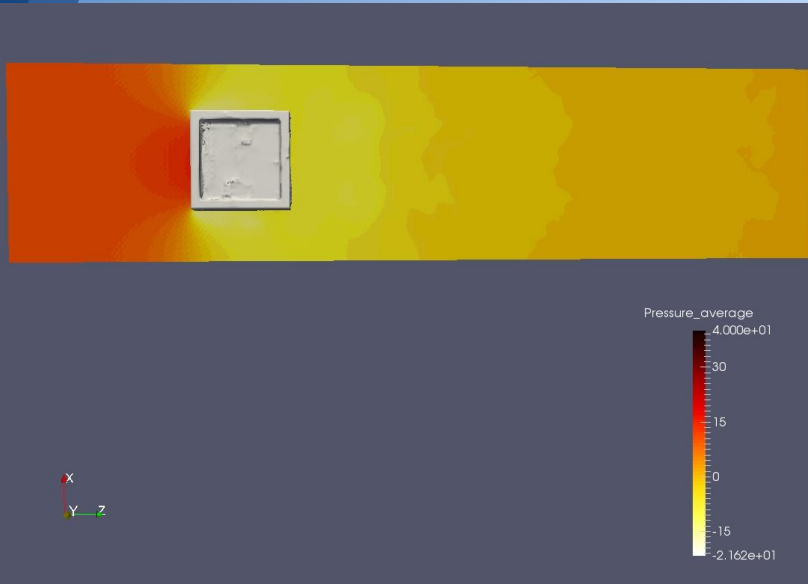
Examples



stream tracer - side view

Examples

pressure - avg. - top view



velocity - instantaneous - side view

Performance

Starting with

4x MPI + 8x OpenMP per node

2M points, 2000 iterations

output every 500 iterations

174 M Lattice Point Updates / second

Changes

First OpenAcc version:

- 1 MPI rank/node (no OpenMP threads)

- 2M points, 2000 iterations

- output every 500 iterations

- NOT 174 M LPU / second

- Only 18 M “

Incremental Changes

First OpenACC attempt	17 M
CopyIn/Out	51 M
Move Bounds out	60 M
Fix copy bug; Stream In/Out	40 M

Types of Changes

```
#pragma omp parallel for collapse(3)
```

```
#pragma acc data \
```

```
    copyin(inl[0:nnodes], onl[0:nnodes], snl[0:nnodes]) \
```

```
    copyin(Mspeeds[0:numMspeeds], Pspeeds[0:numPspeeds]) \
```

```
    copyin(fEven[0:nnodes*numSpd], fOdd[0:nnodes*numSpd])
```

```
#pragma acc parallel loop collapse(3) \
```

```
    present(streamSpeeds[0:numStreamSpeeds]) \
```

```
    present(fIn_b[0:Nx*Ny*numSpd*HALO]) \
```

```
    copyin(buff_in[0:Nx*Ny*numStreamSpeeds*HALO])
```

Big Breakthrough

Min. data movement (halo only) except on write (every 500 timesteps)

662 M LPU/sec! (c.f. 172 M , 18 M)

Cached directive

757 M LPU/sec!

More Speed

Transposed our access to our data:

More Speed

Transposed our access to our data:

1308 M LPU/s (1,3 G LPU/s) !!

More Speed

Transposed our access to our data:

1308 M LPU/s (1,3 G LPU/s) !!

Rebuilt/reran baseline w/improved algor.:
227 M LPU/s (up from 174 M) (4ranks 8thrds)

Next Steps

Continue to modify
profile, re-arrange code (1500 M LPU/s)

Besides bricks

Other domain decompositions

Tools we wish we had

Visual memory mapper

Live comparator

- two running versions

- to see where they diverge

More magic

Tools we liked

NVIDIA Visual Profiler !!

PGI compiler

esp. output - right level of info

Cray compiler

PizDaint - Thank you, CSCS !

Eurohack'15

Great experience
concentrated efforts
mentors, experts

Great location
hotel, town, hosts
time zone advantage, too

During summer break

Tools we liked

NVIDIA Visual Profiler !!

PGI compiler

esp. output - right level of info

Cray compiler

PizDaint - Thank you, CSCS !



* final slide as per Michael Wolfe