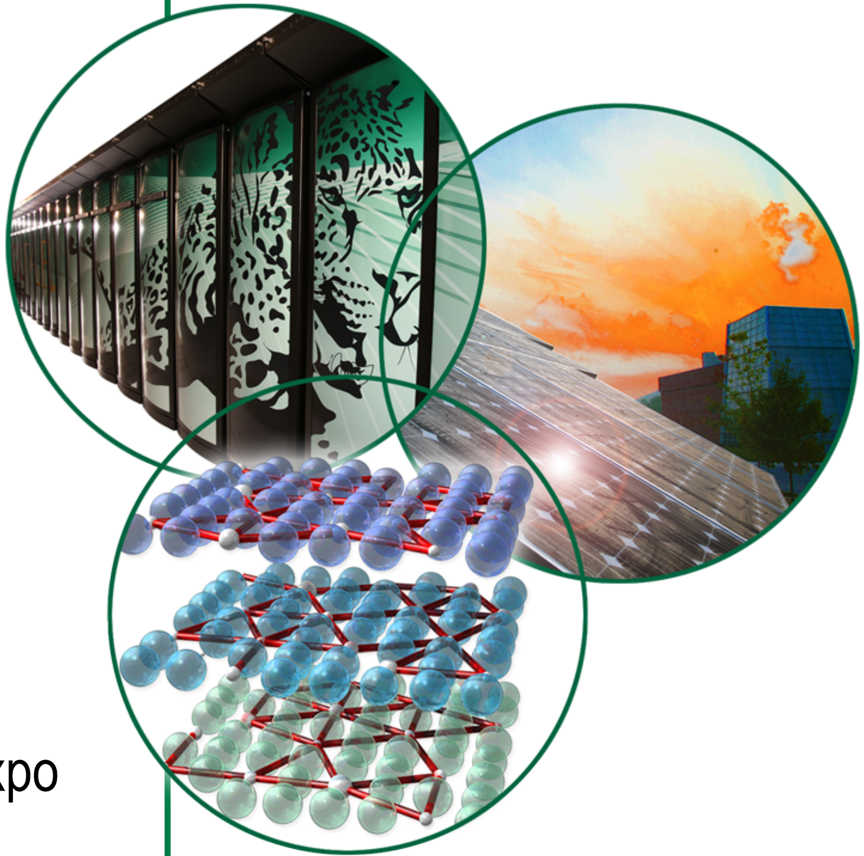
PRATHAM

PaRAllel Thermal Hydraulics Simulations using Advanced Mesoscopic Methods

Abhijit S. Joshi Jaime A. Mudrich

Emilian L. Popov Prashant K. Jain



Presented by

Abhijit Joshi

joshias@ornl.gov

ANS Winter Meeting & Nuclear Technology Expo

San Diego, CA

November 2012

U.S. DEPARTMENT OF ENERGY

OAK RIDGE NATIONAL LABORATORY

MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

MOTIVATION

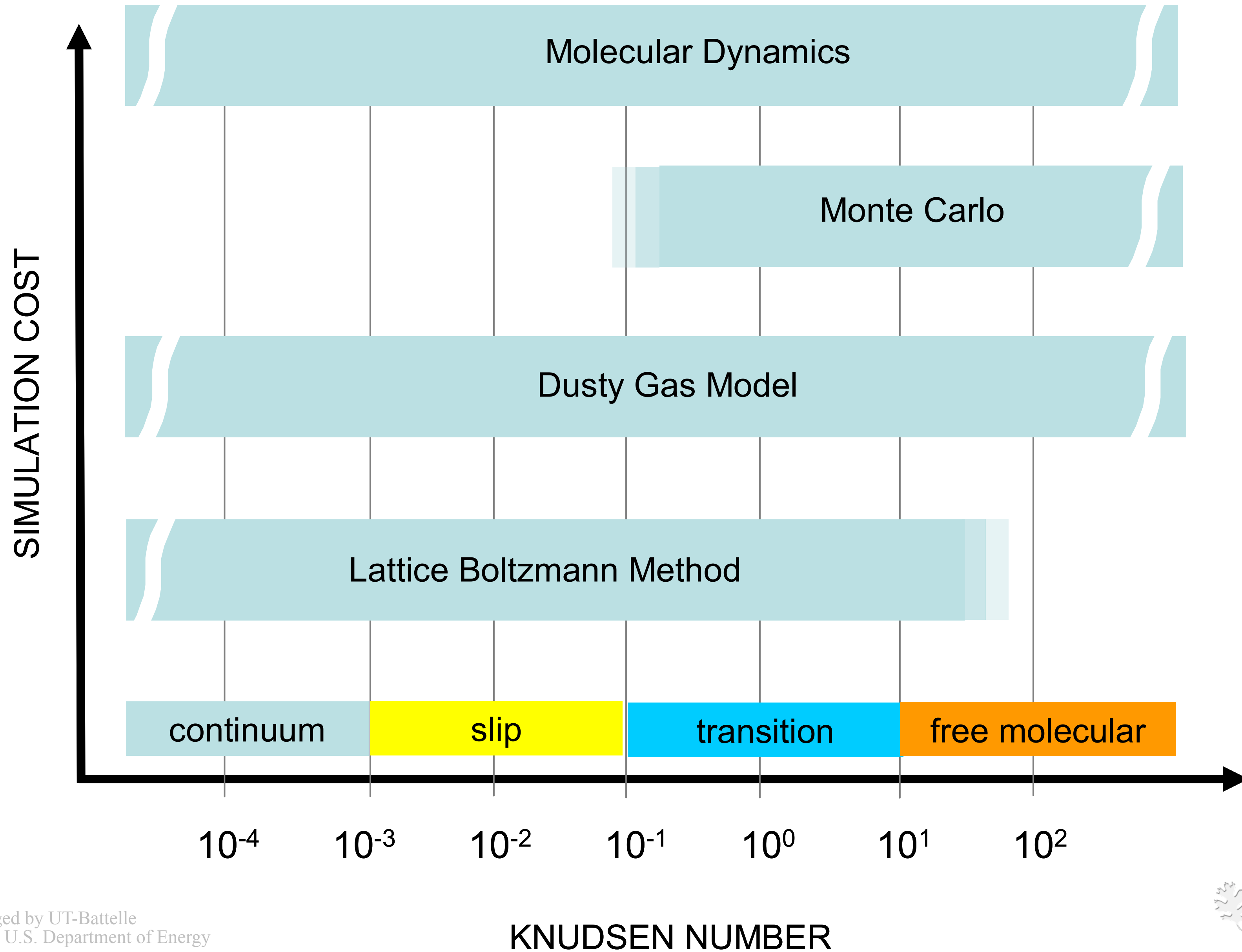
- Most commercial CFD codes use RANS-based turbulence models that are not suitable for accurately modeling flow transients and instabilities
- There is a need to develop highly accurate CFD models that can run on thousands of processors with good parallel efficiency
- The lattice Boltzmann method (LBM) has emerged as a promising tool for accurate CFD and is well-suited for flow in complex geometries and for ease in incorporating complex physics

MOTIVATION (continued)

- **LBM has advanced rapidly in the last 20 years and is now seen as a viable alternative to the more traditional CFD approach of solving the Navier-Stokes equations**

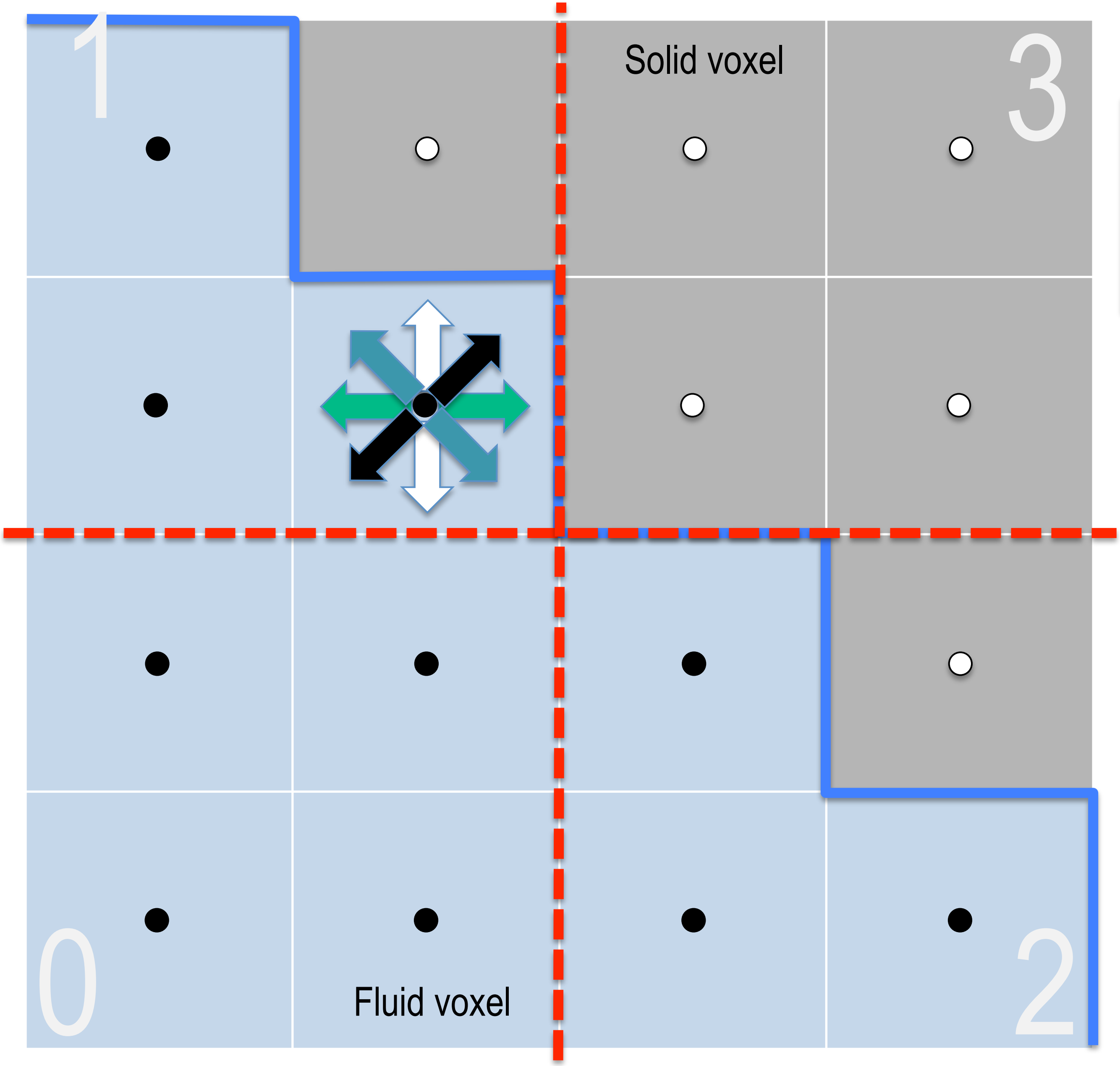
- **PRATHAM is a 3D, parallel LBM code being developed at ORNL to demonstrate the accuracy and scalability of LBM for turbulent flow applications**

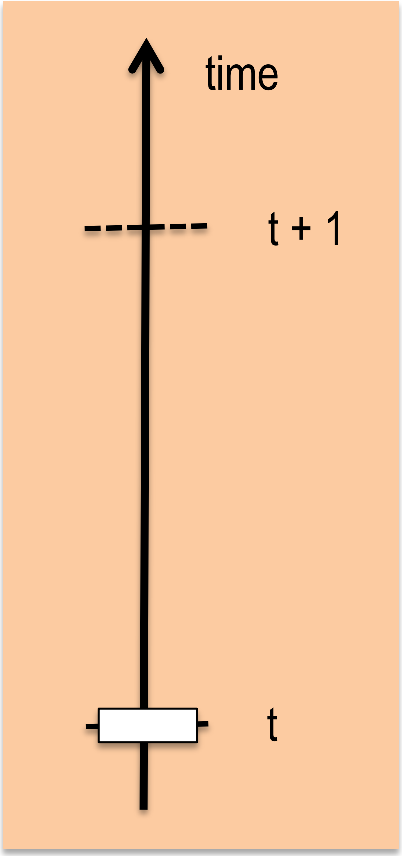
MESOSCOPIC METHODS – some examples



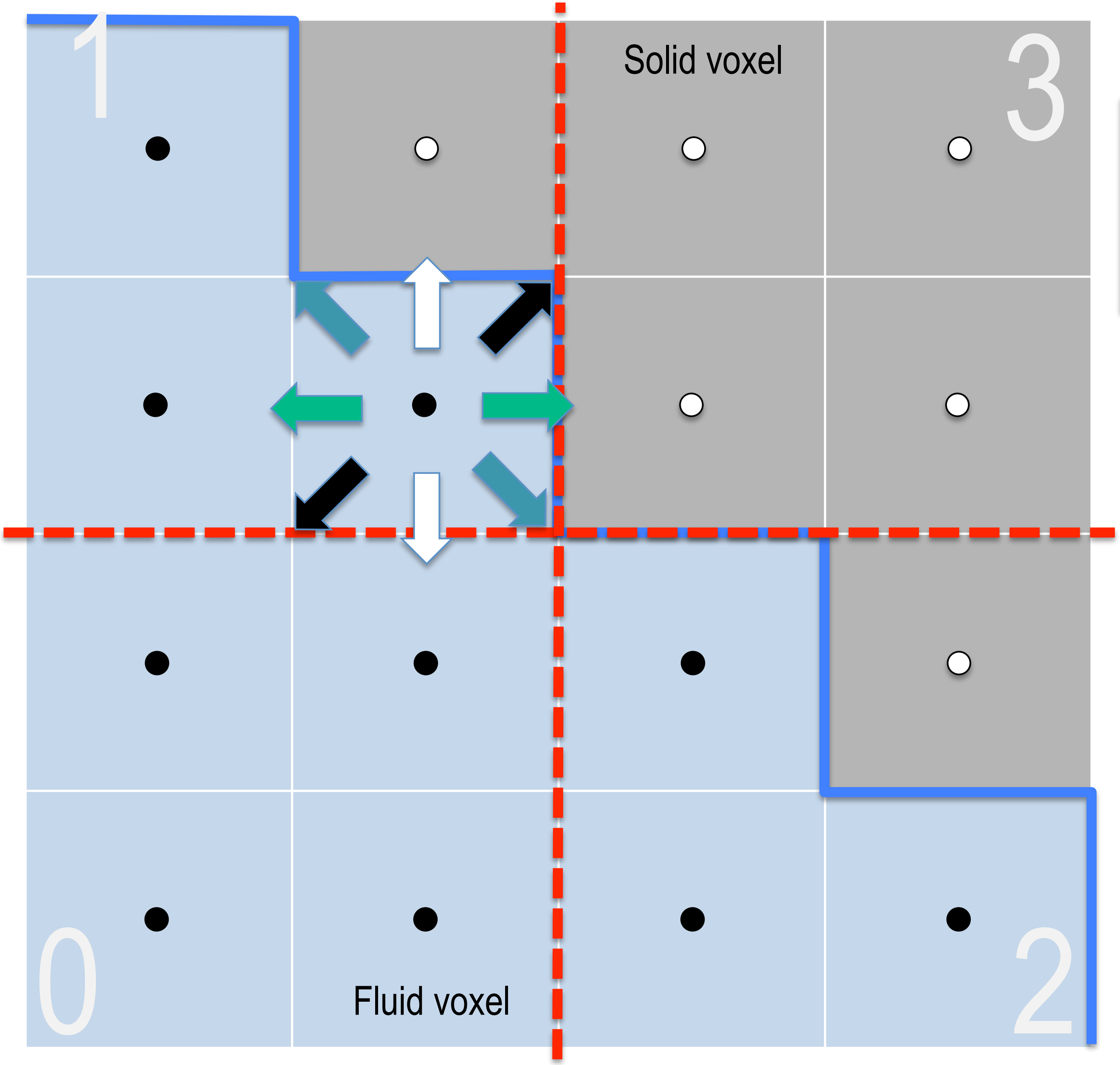
LBM steps

stream

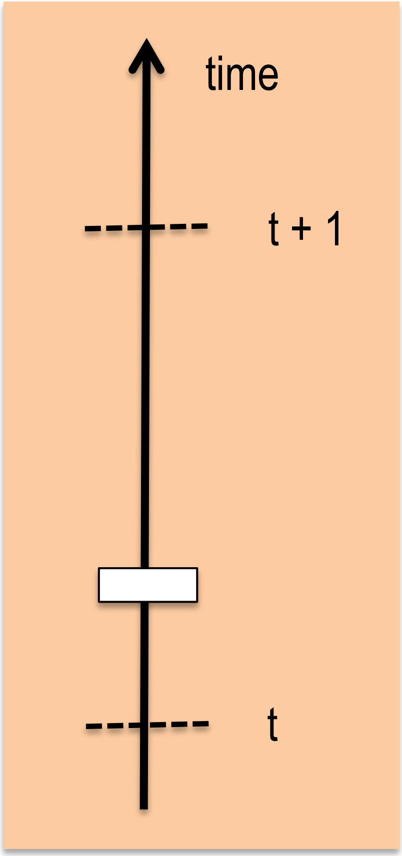




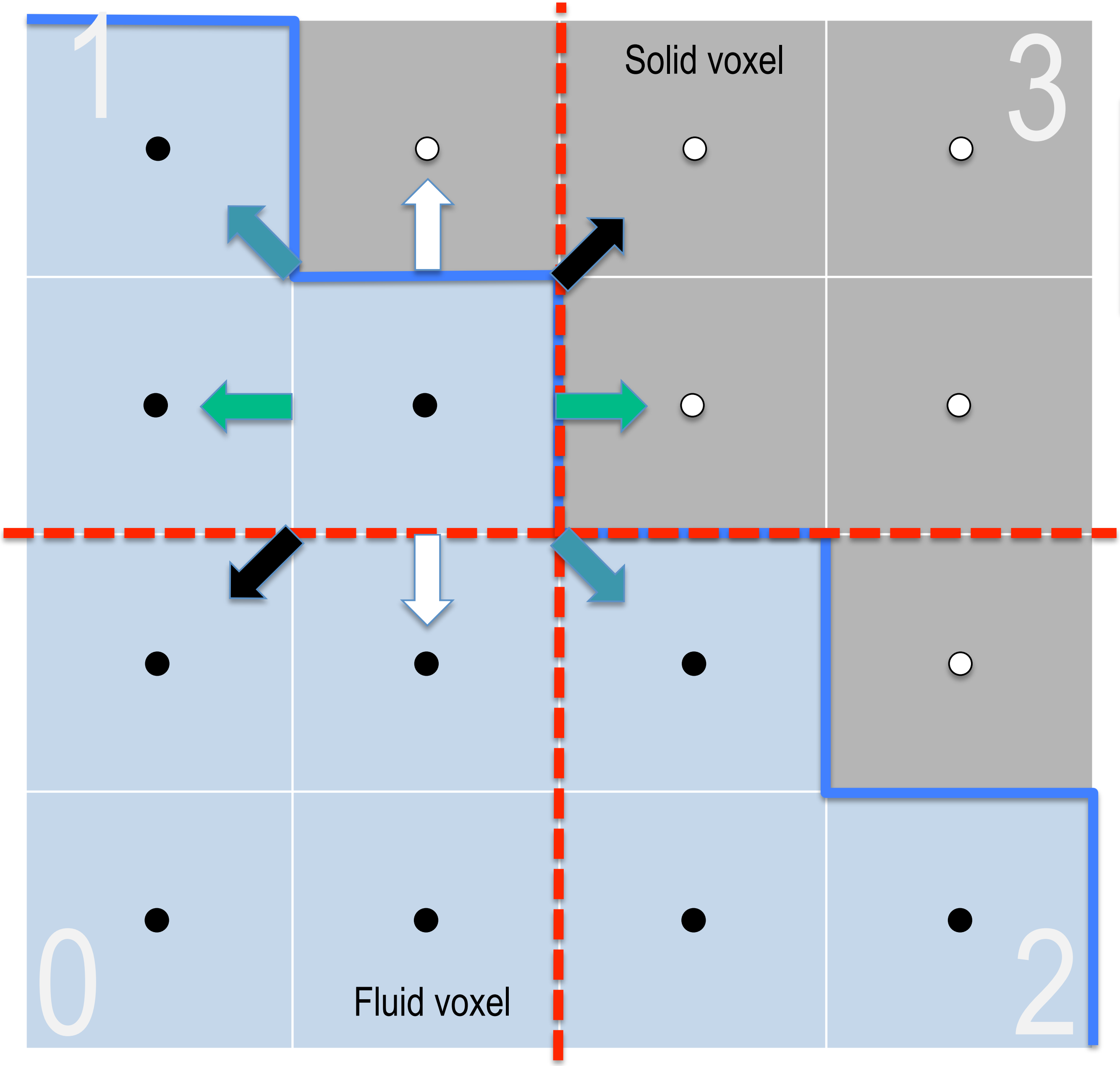
LBM steps

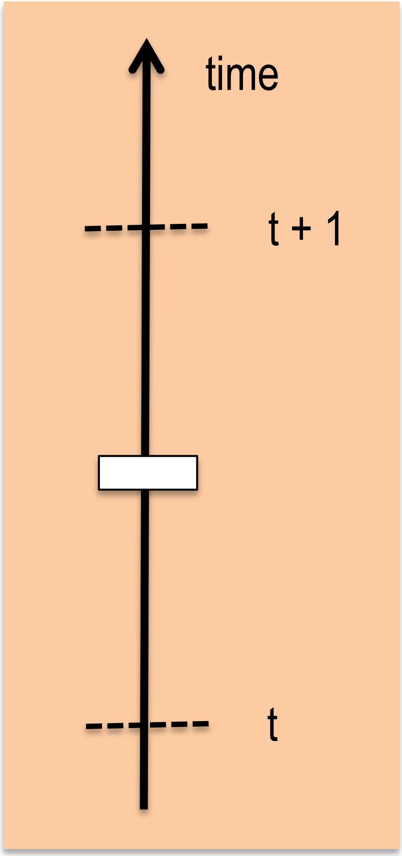


stream

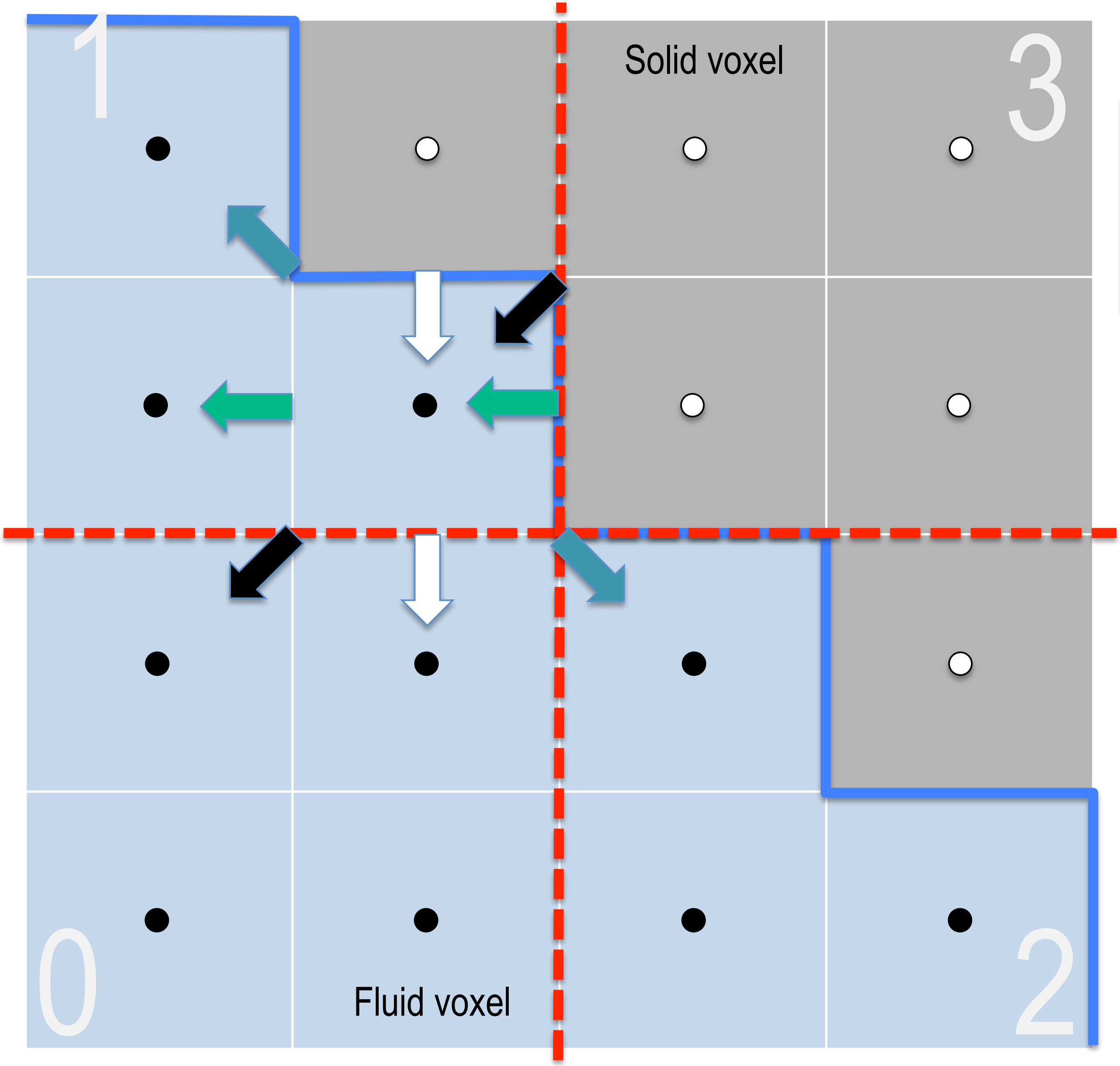


LBM steps

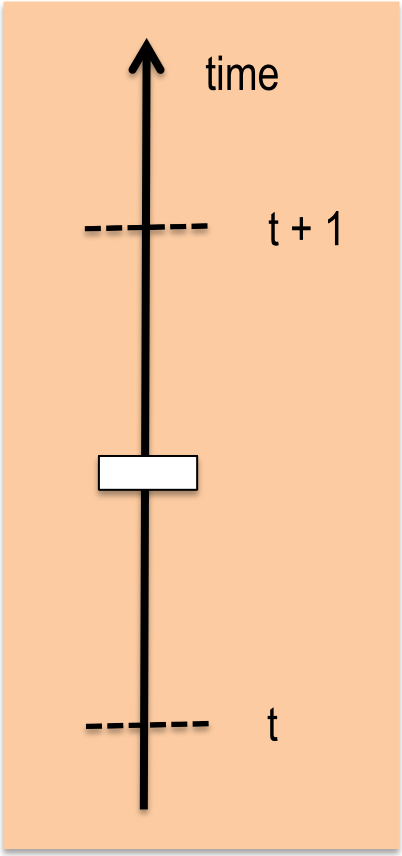




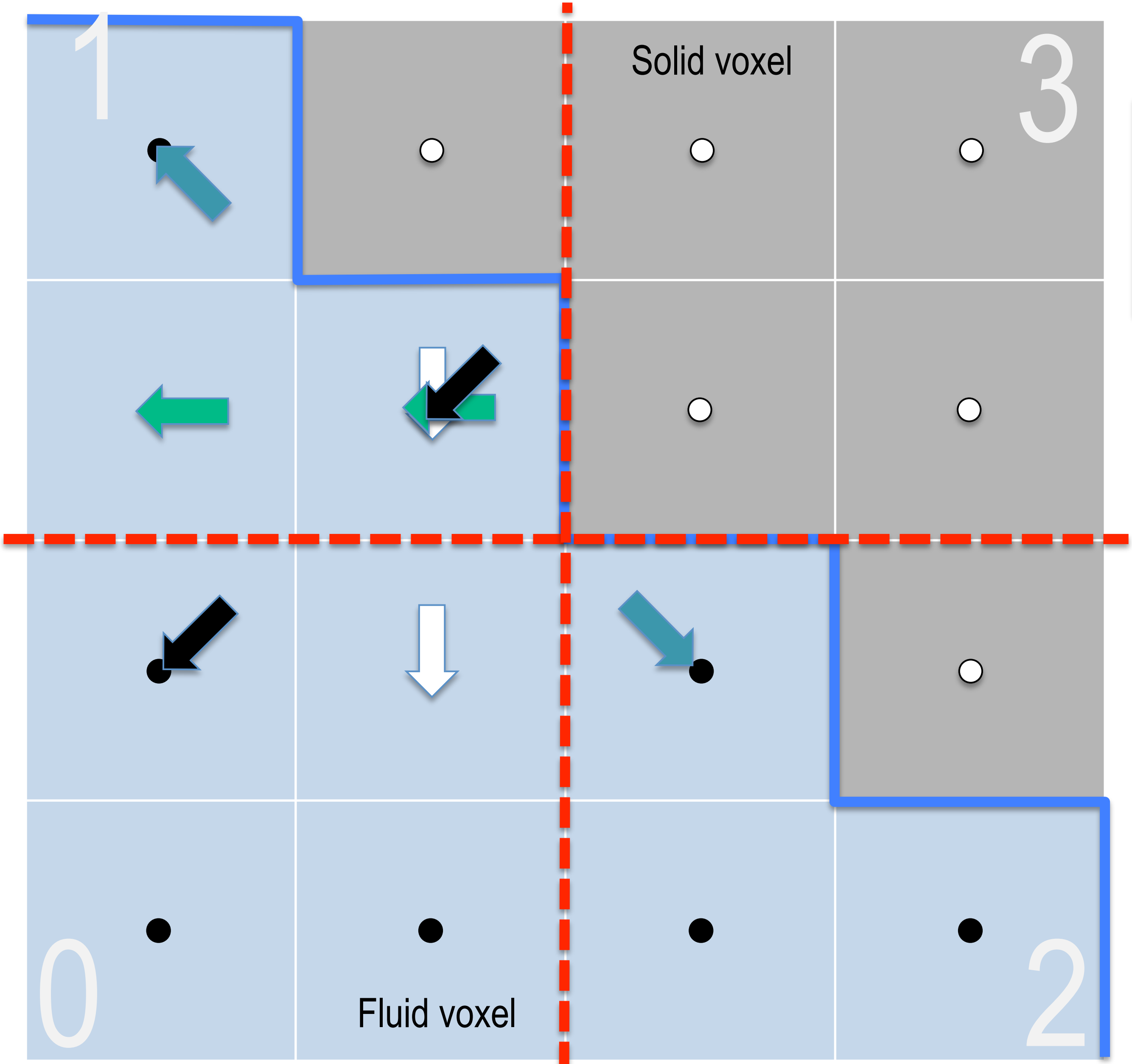
LBM steps



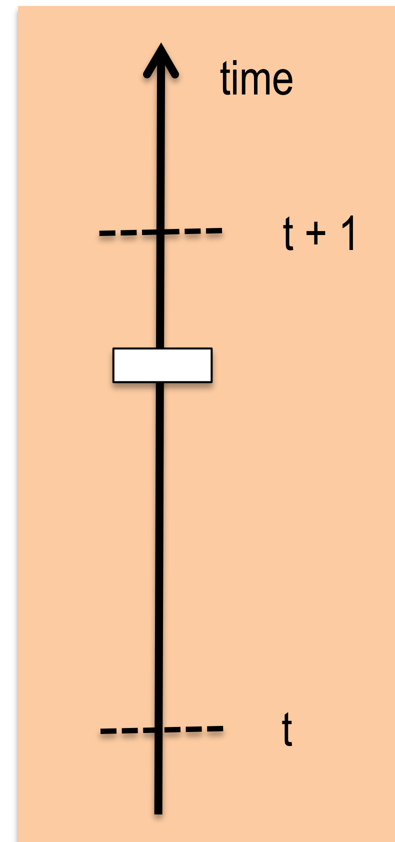
stream



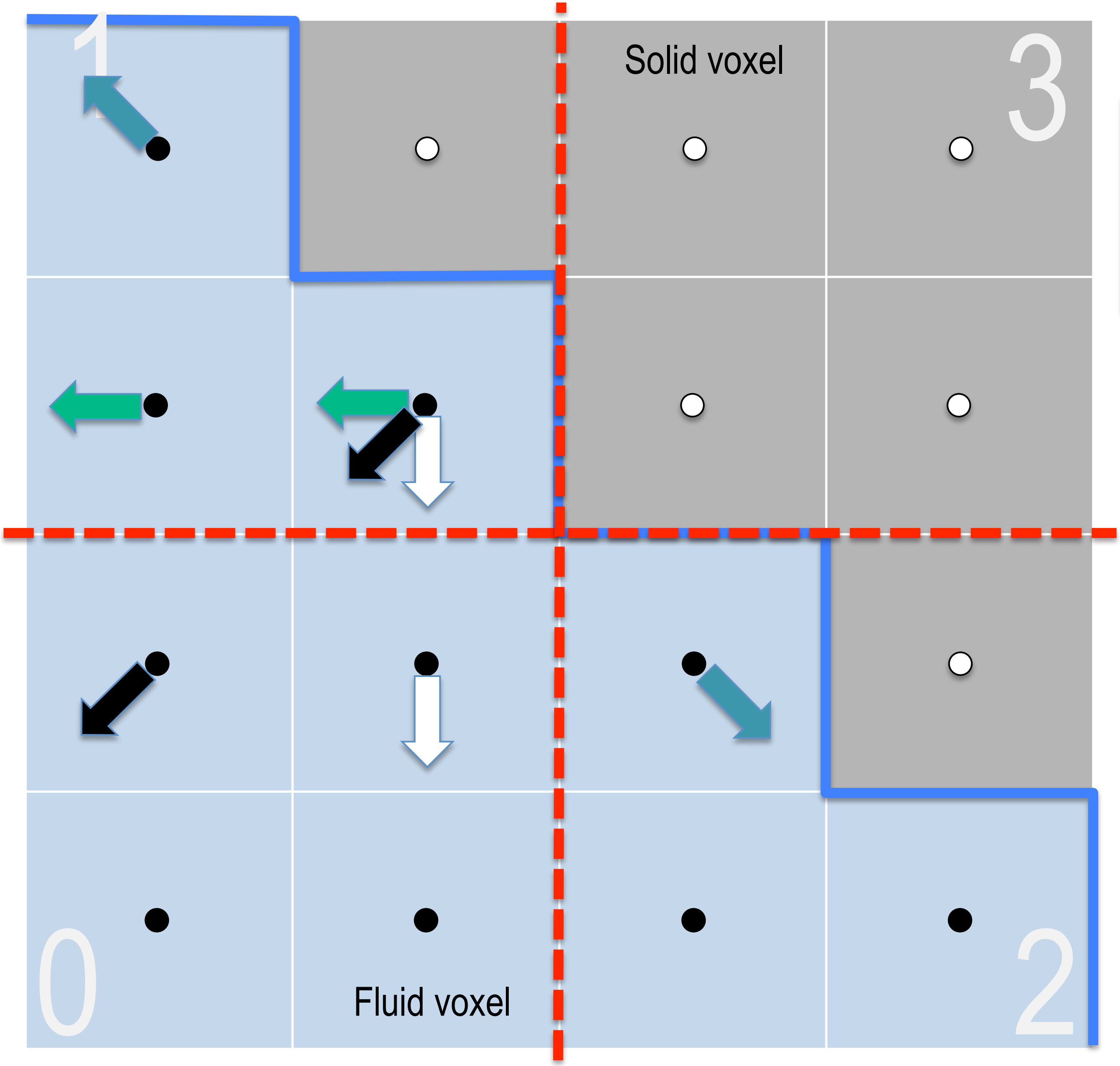
LBM steps



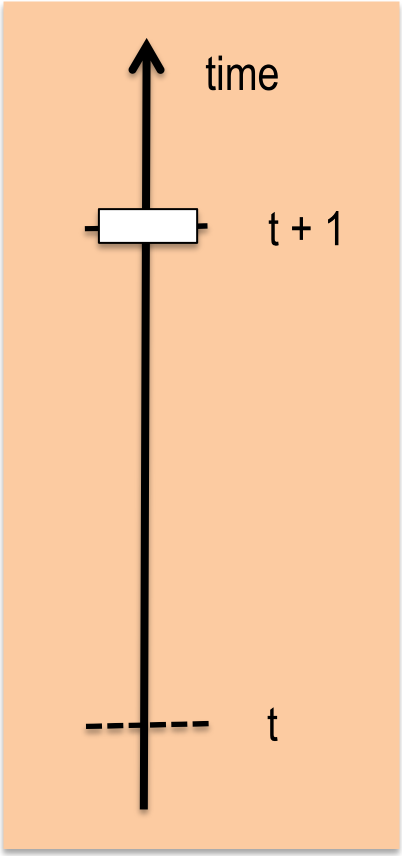
stream



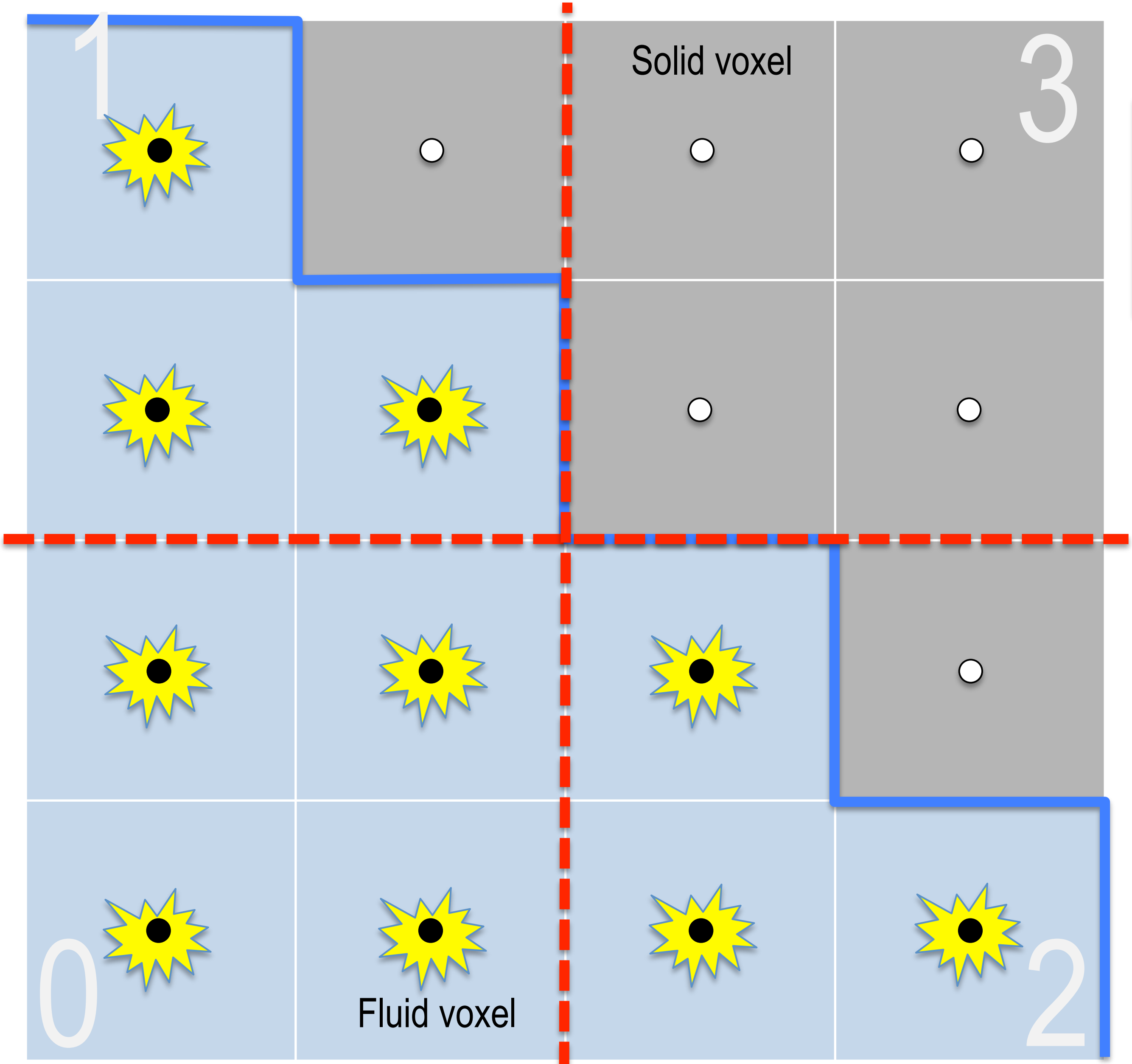
LBM steps



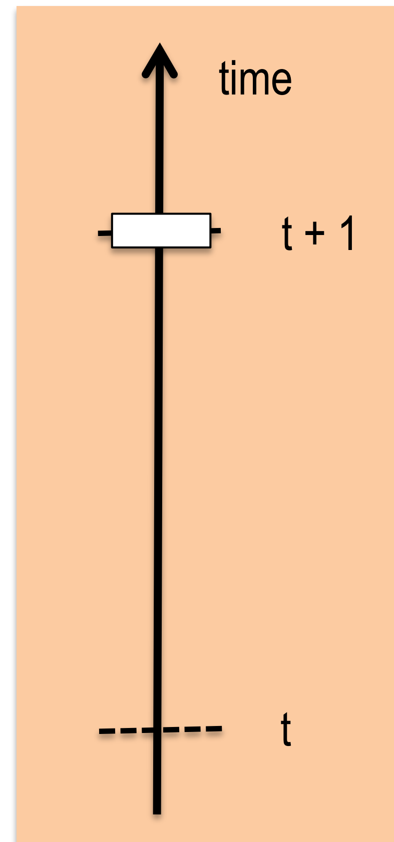
stream



LBM steps



collide



PRATHAM : COLLISION KERNELS

single relaxation time (SRT)

- Easier to implement
- Less expensive computationally
- May not be stable for high Re for certain problems

multiple relaxation time (MRT)

- Comparatively difficult to implement
- More expensive computationally
- Improvements in stability and able to reach higher Re

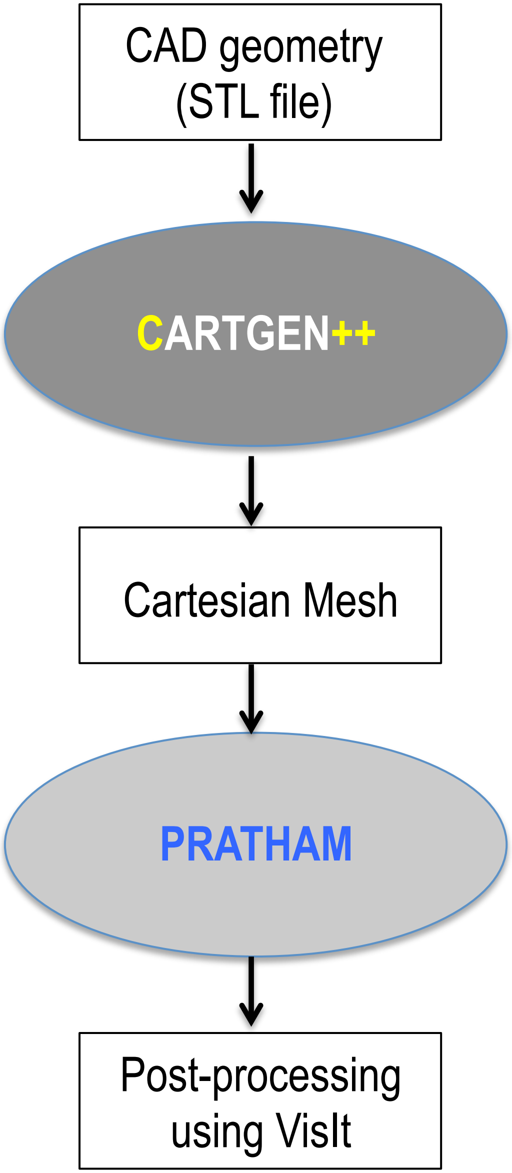
PRATHAM: code organization

in/	problem parameters, geometry input & seed point locations
src/	source code & Makefile
out/	SILO files for post-processing
rest/	restart files in binary format
trace/	time-trace files for seed points
README	instructions for installing and running PRATHAM

PRATHAM: compile time options

-DUSE_MIDWAY_BOUNCE_BACK	implements off-site bounce-back for all solid nodes (fixed as well as moving) recommended for stationary as well as moving walls and fixed obstacles inside the domain.
-DUSE_LES	turns on the sub-grid Smagorinsky model where fluid viscosity is modified based on the rate-of-strain. Recommended for large Reynolds numbers.
-DUSE_LBGK	uses the single-relaxation-time scheme (LBGK) instead of the multiple-relaxation-time scheme.
-DCHECK_DFDT	allocates additional memory (f_old) for checking MAX|dfdt| and also writes f_old to restart files. This increases the memory requirement during run-time and may be expensive for large 3D runs. Not recommended for runs using > 500 cores.
-DWRITE_UNSTRUC_MESH	writes unstructured SILO data for fluid nodes with a specified connectivity (solid nodes are ignored).
-DUSE_PROBES	writes time-traces for probe points (seeds) defined before the start of the run.

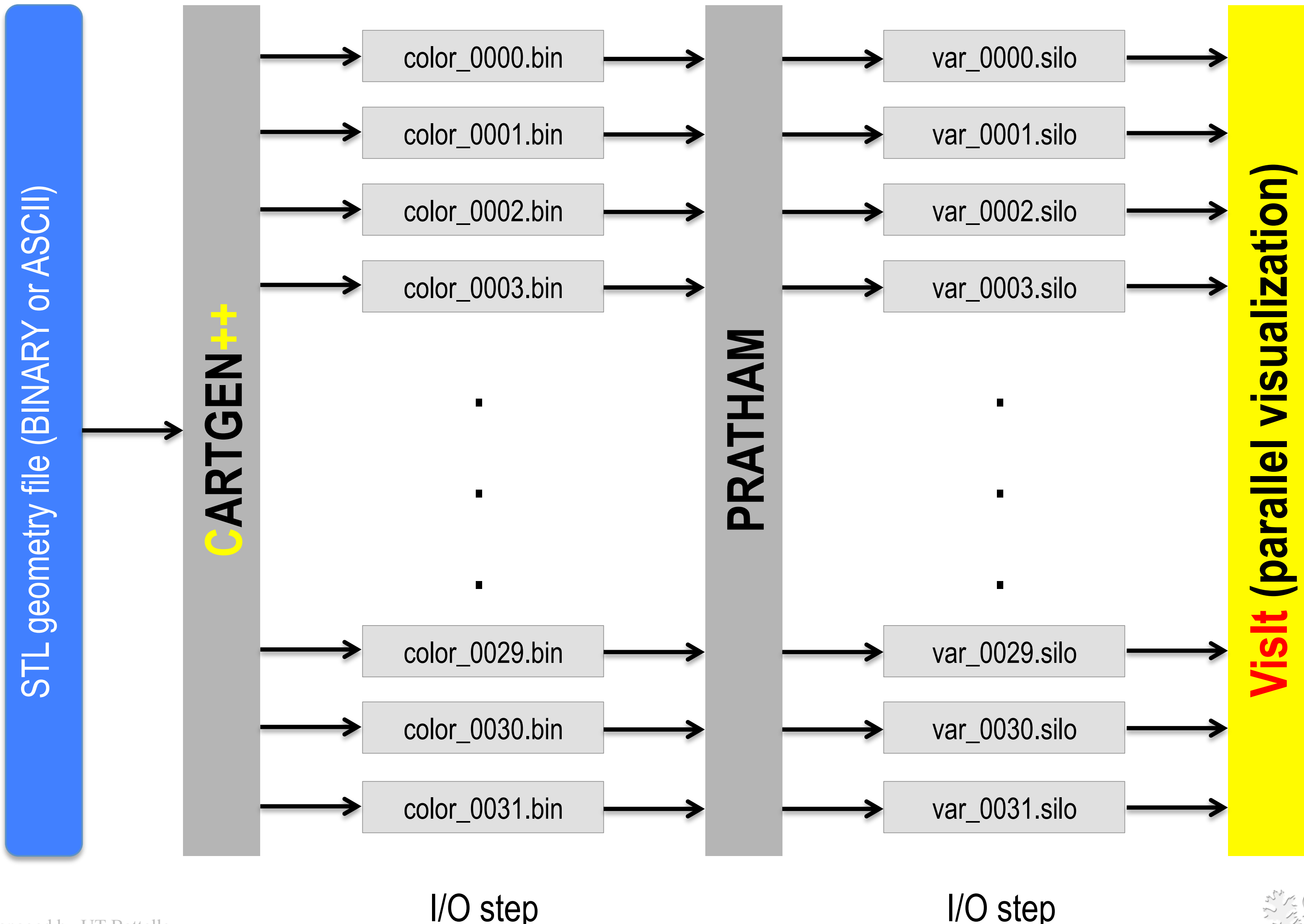
PRATHAM: work flow overview



Fully parallel mesh generator
C++ with MPI

Fully parallel LBM based code for CFD analysis
FORTRAN-90 with MPI

PARALLEL WORKFLOW FROM MESHING TO VISUALIZATION



PRATHAM: lattice units and physical units

The LBM calculations are carried out in the “LBM world” where quantities are in “lattice units”

DIMENSIONLESS NUMBERS in the lattice and physical world are identical

A simple EXCEL spreadsheet (see below) is used to move from physical to lattice units

Physical problem:	value	units
Length scale	**1.29E-01**	m
Kinematic viscosity	**1.00E-06**	m2/s
Velocity scale	0.245	m/s
REYNOLDS NUMBER	**31,577.00**	
Time scale	0.527	s
DIMENSIONLESS TIME	1.11111	
Simulation time	0.586	s

LBM problem	value	units
Length scale	**90**	lu
Kinematic viscosity	2.85E-04	lu
Velocity scale	**0.1**	lu
REYNOLDS NUMBER	31,577.00	
Time-scale	900	lu
DIMENSIONLESS TIME	1.111111	
Simulation time	**1000**	lu

Relaxation time(tau)	0.5008551
Mach number	0.173

COMPUTATIONAL RESOURCES USED

- For large 3D problems, both **CARTGEN++** and **PRATHAM** have been run on up to 6912 cores on JAGUAR, one of the nation's most powerful supercomputers, showing very good scaling.

- Several small clusters at ORNL were used to run medium sized 2D and 3D problems. These include MEGY (32 processors) and OIC (upto 480 cores were used).

- For many small problems (typically 2D), the codes can also run on an average laptop after installing the necessary open-source libraries (OpenMPI, hdf5, Silo).

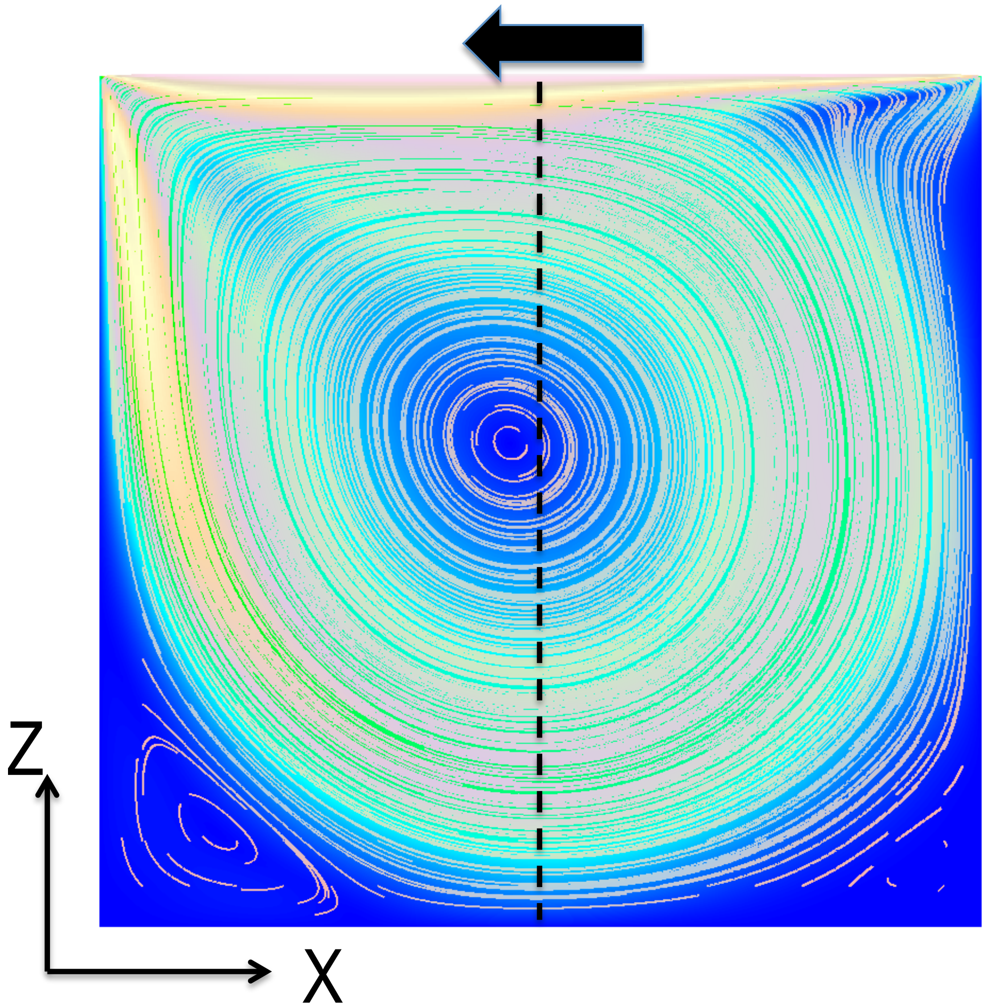
SUMMARY OF CASE STUDIES

PROBLEM	GRID SIZE	MACHINE	CORES
2D lid driven cavity (Re = 1,000)	200 x 2 x 200	MEGY	32
3D lid driven cavity (Re = 10,000)	400 x 400 x 400	OIC	480
3D lid driven cavity (Re = 20,000)	400 x 400 x 400	OIC	480
2D flow around a cylinder	1000 x 2 x 200	OIC	216
3D flow around a cylinder	1000 x 250 x 250	OIC	240
3D flow through a long pipe	8000 x 200 x 200	JAGUAR	6912

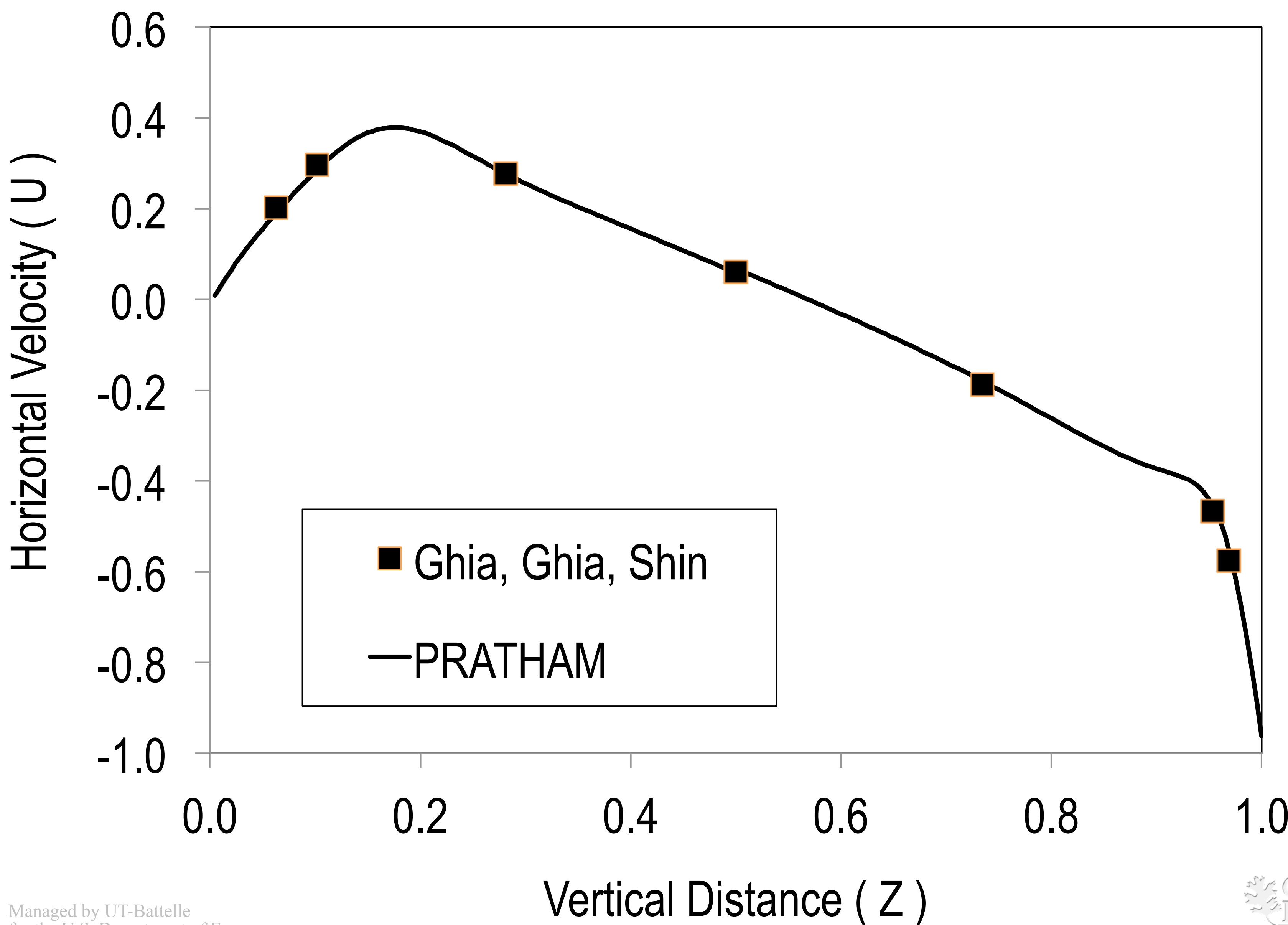
CASE STUDY 1

Flow inside a 2D driven cavity at Re = 1,000

2D DRIVEN CAVITY - schematic



2D DRIVEN CAVITY (Re = 1,000)



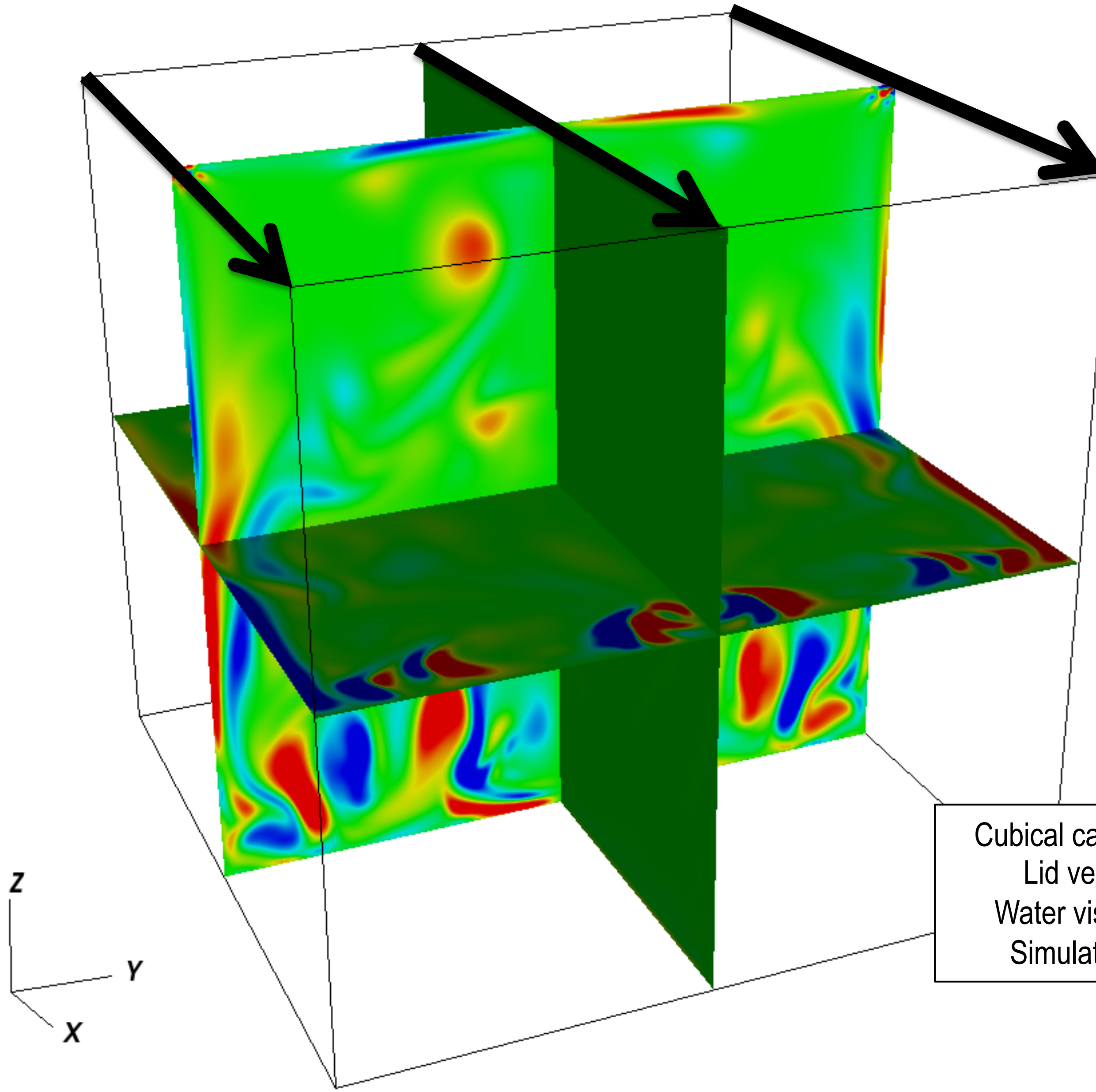
CASE STUDY 2

Flow inside a 3D driven cavity at Re = 10,000

3D DRIVEN CAVITY (Re = 10,000)

Plot of

$$\left(\nabla \times \mathbf{u}\right)_x$$



Simulation using a **400 × 400 × 400** lattice.

LES model on.
Midway BB on.

Run on **480 cores** (6 × 8 × 10 decomposition).

Machine: OIC

This image is obtained after **70,000 time steps**, corresponding to a run time of **13.48 hours**.

Cubical cavity of length **10 [cm]**
Lid velocity = **10 [cm/s]**
Water viscosity = $\mathbf{10^{-6}}$ **[m²/s]**
Simulation time = **17.5 [s]**

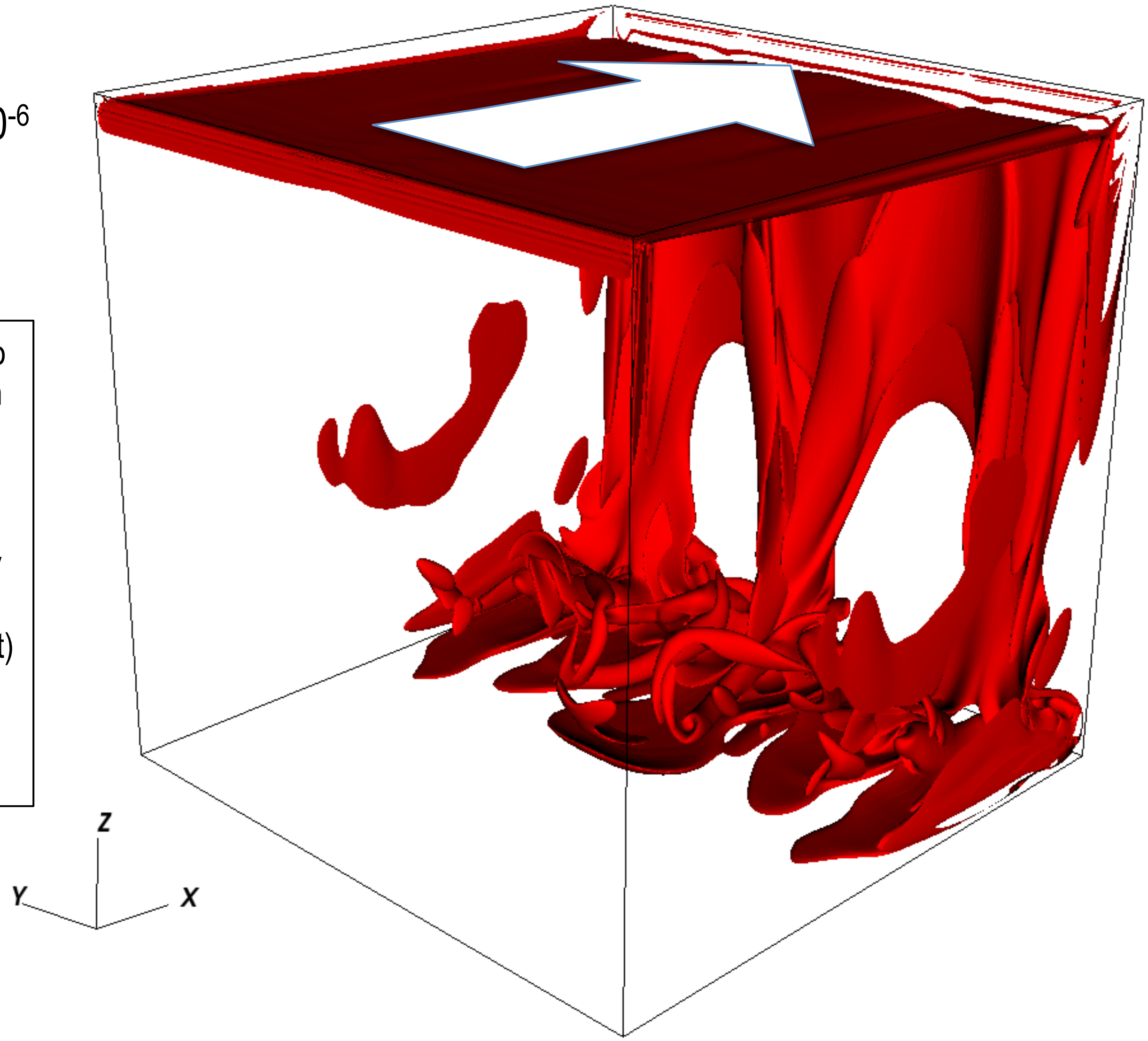
3D DRIVEN CAVITY (Re = 10,000)

ENSTROPHY = 5 x 10^{-6}

$$(\nabla \times \mathbf{u}) \cdot (\nabla \times \mathbf{u})$$

"**Enstrophy**" can be useful to get an idea of the dissipation of K.E. in turbulent flow.

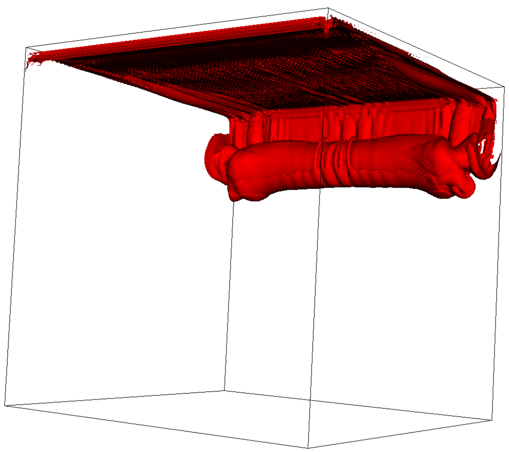
In this case, the energy is being constantly provided by the motion of the lid and eventually dissipated (as heat) at the viscous scale.



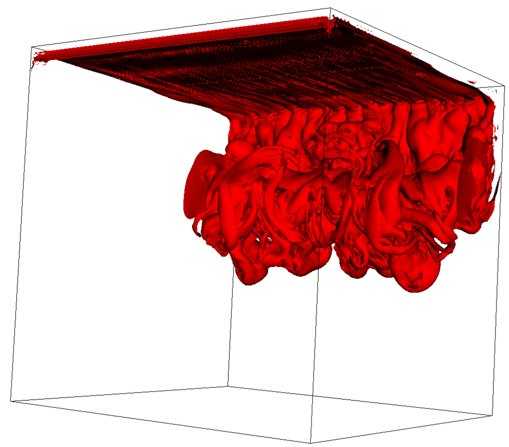
CASE STUDY 3

Flow inside a 3D driven cavity at Re = 20,000

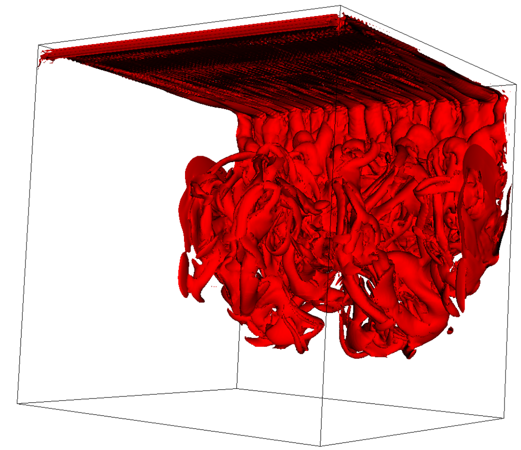
3D DRIVEN CAVITY (Re = 20,000)



t = 5,000



t = 10,000



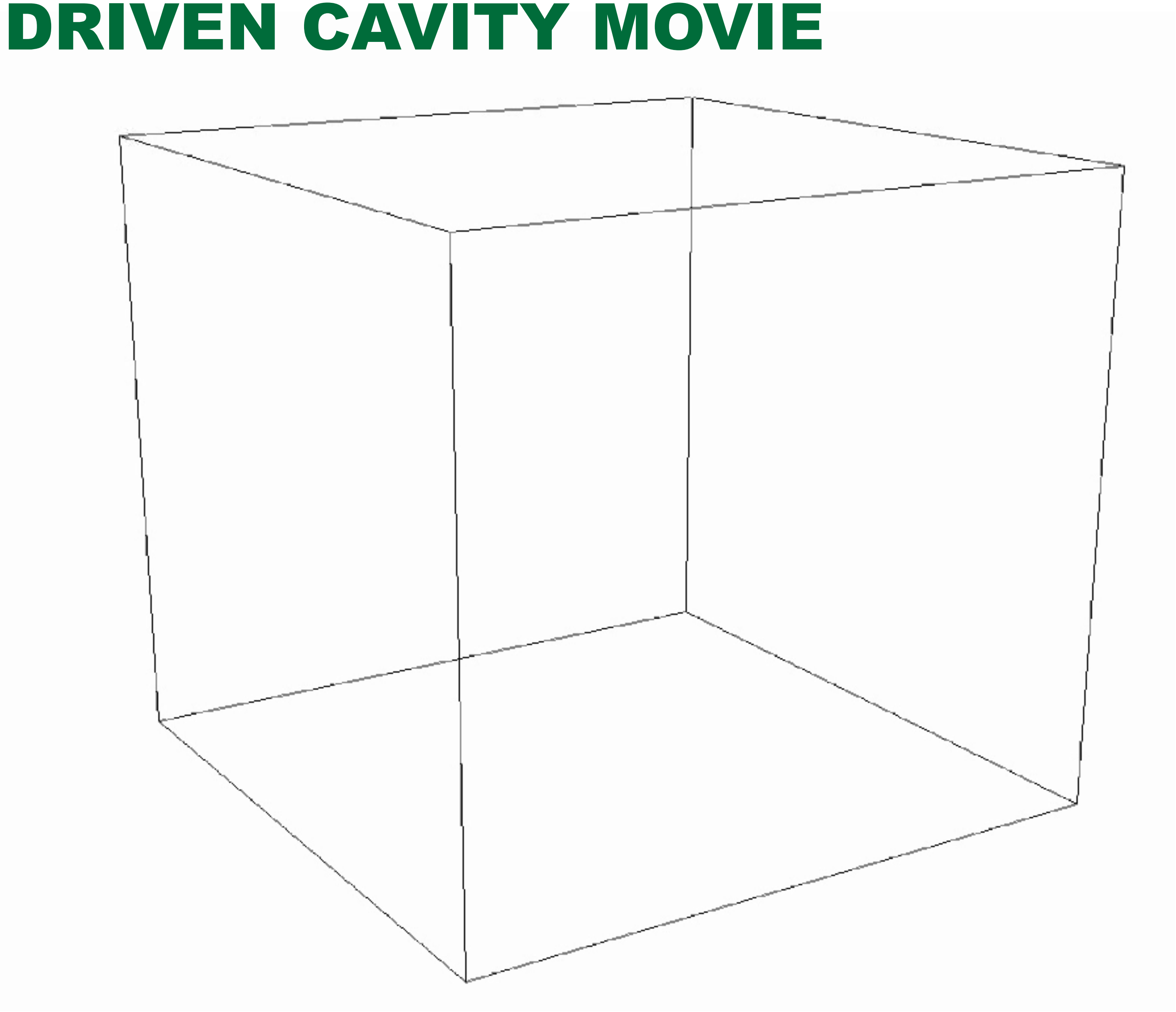
t = 15,000

ENSTROPHY = 1×10^{-5}

$$(\nabla \times \mathbf{u}) \cdot (\nabla \times \mathbf{u})$$

200 x 200 x 200 grid - 216 cores on OIC

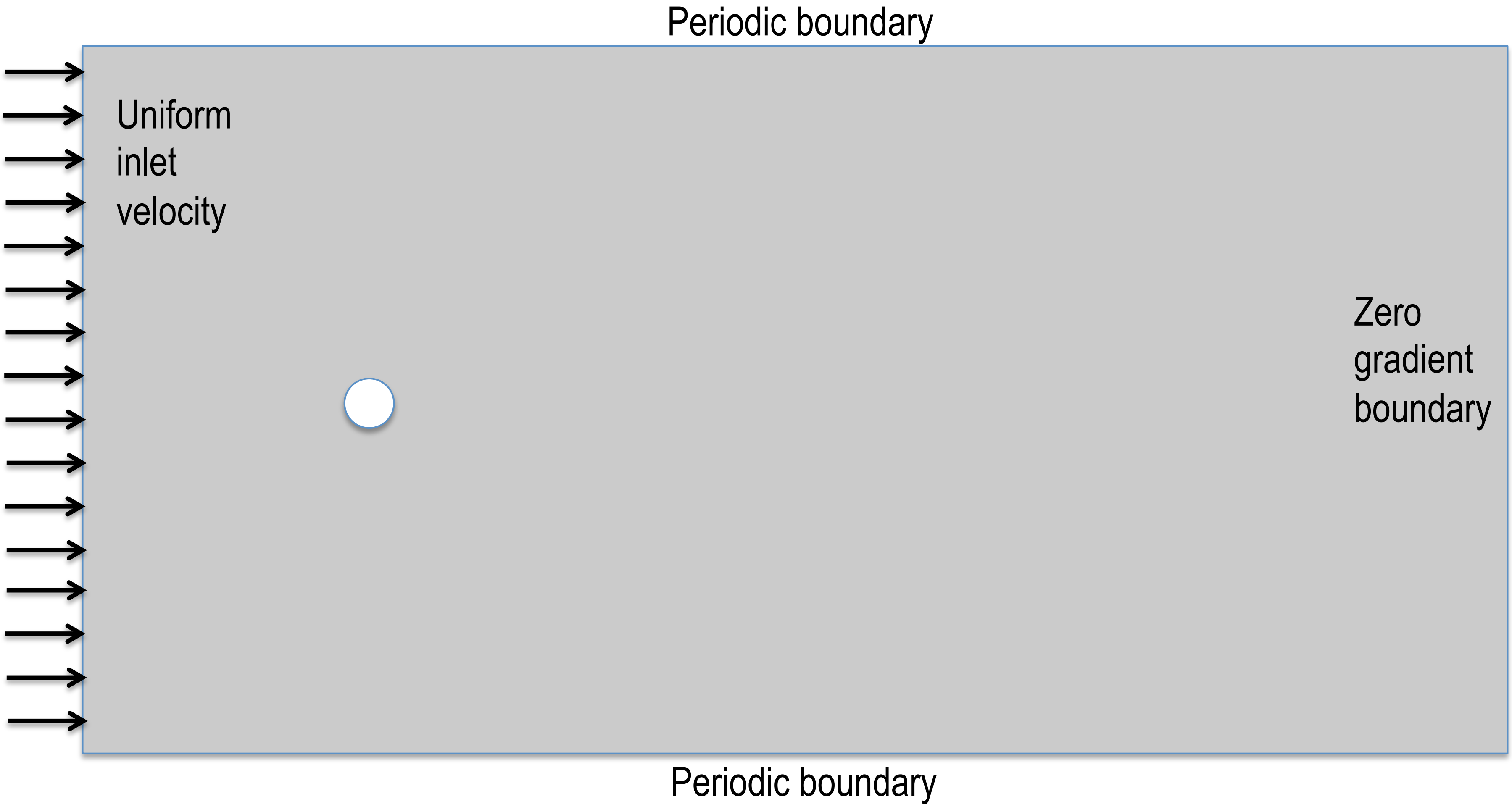
3D DRIVEN CAVITY MOVIE



CASE STUDY 4

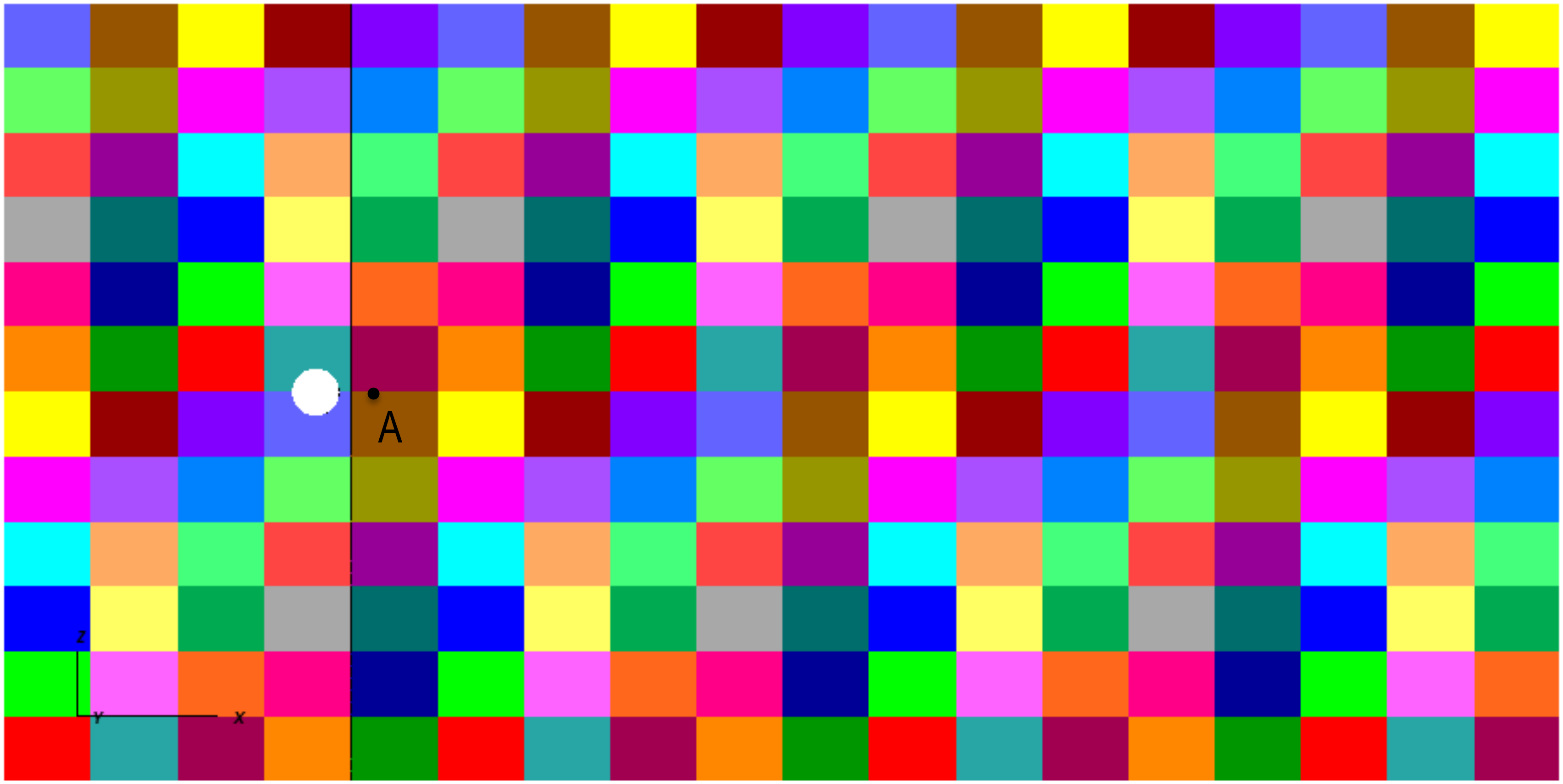
Flow around a 2D cylinder

FLOW AROUND A 2D CYLINDER



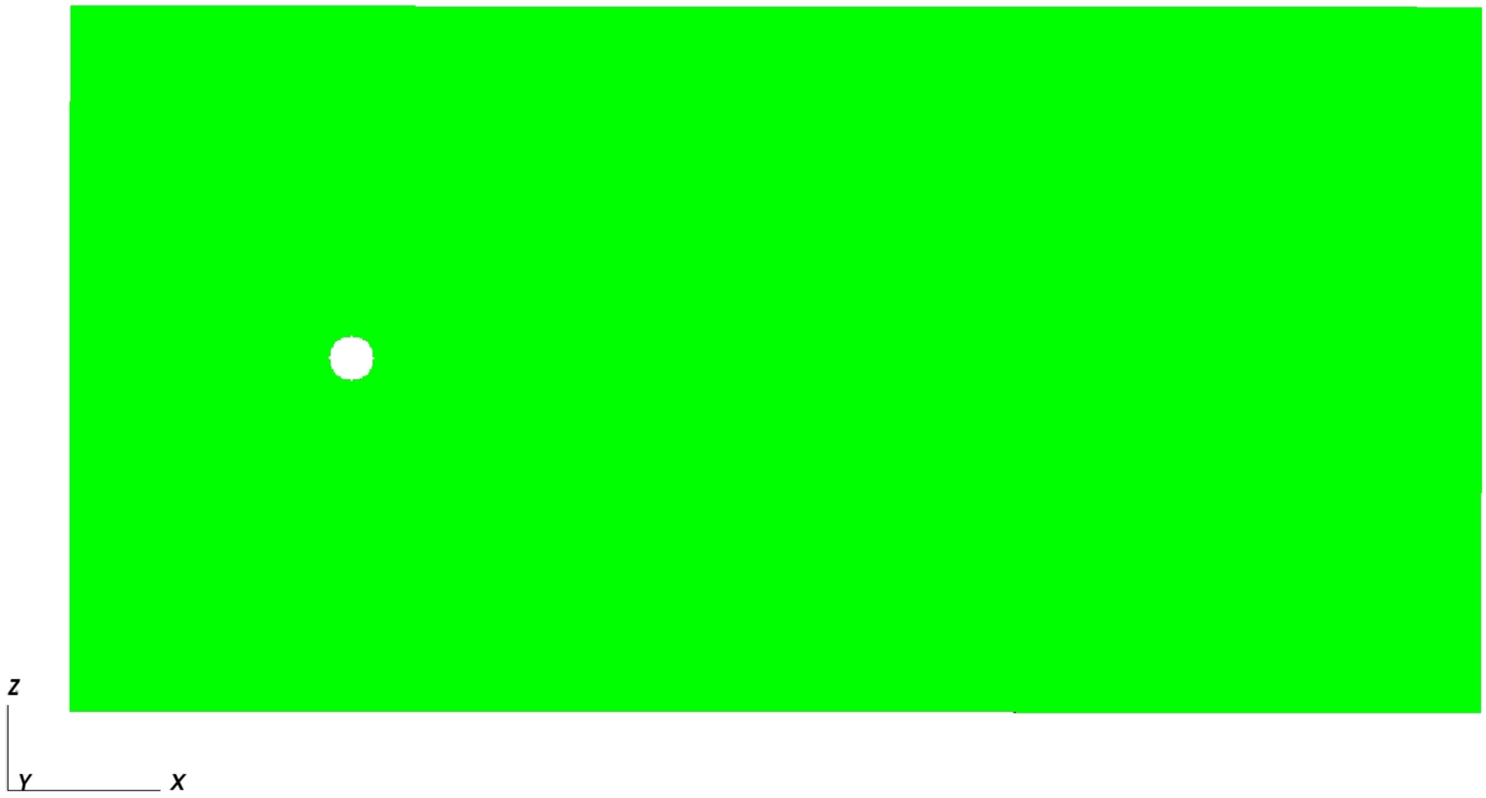
FLOW AROUND A 2D CYLINDER

DOMAIN DECOMPOSITION – 18 x 1 x 12



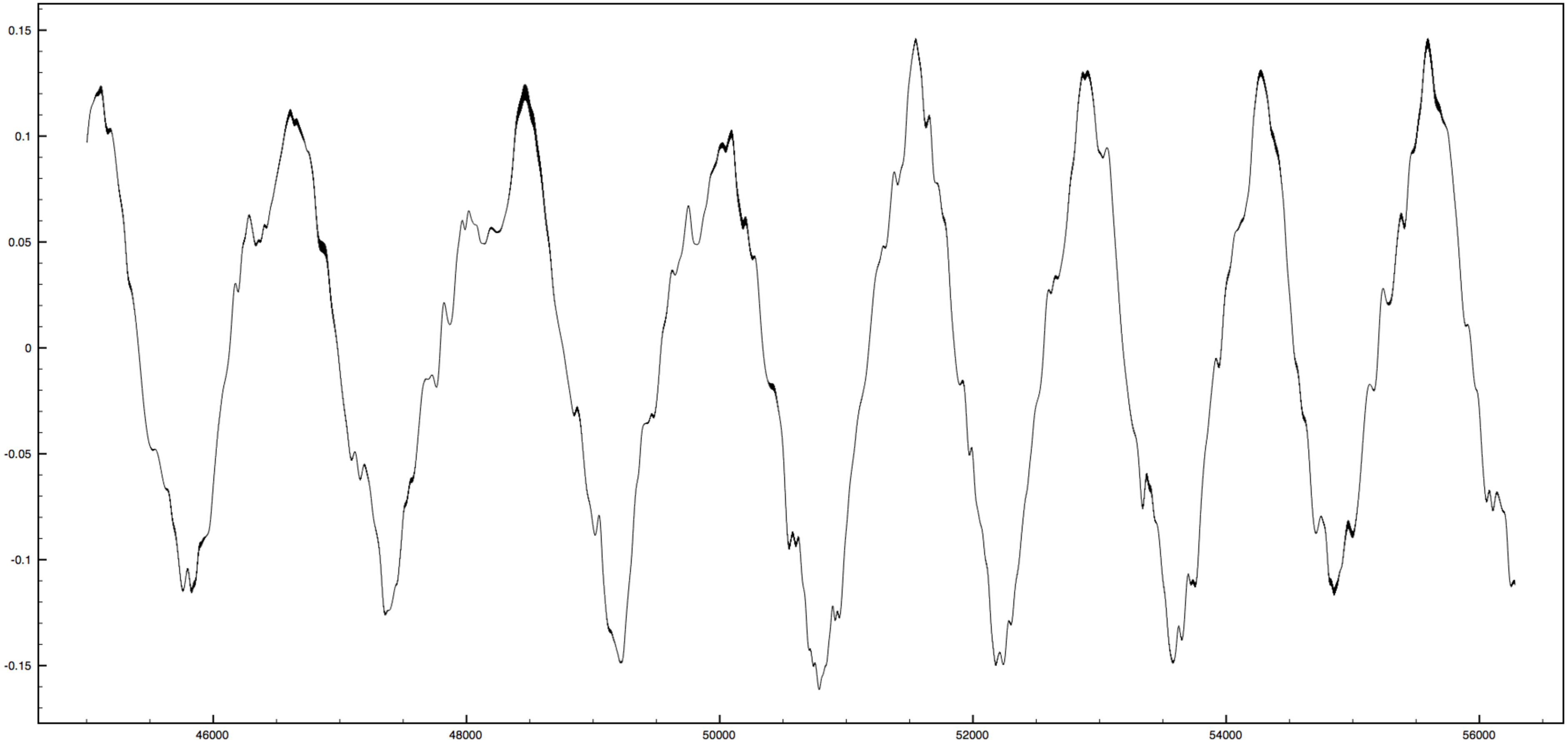
Simulation run on OIC using 216 cores.
At point A, the Z-component of velocity is written to a data file “tracepoint.data” at EVERY time-step.

VORTEX SHEDDING MOVIE



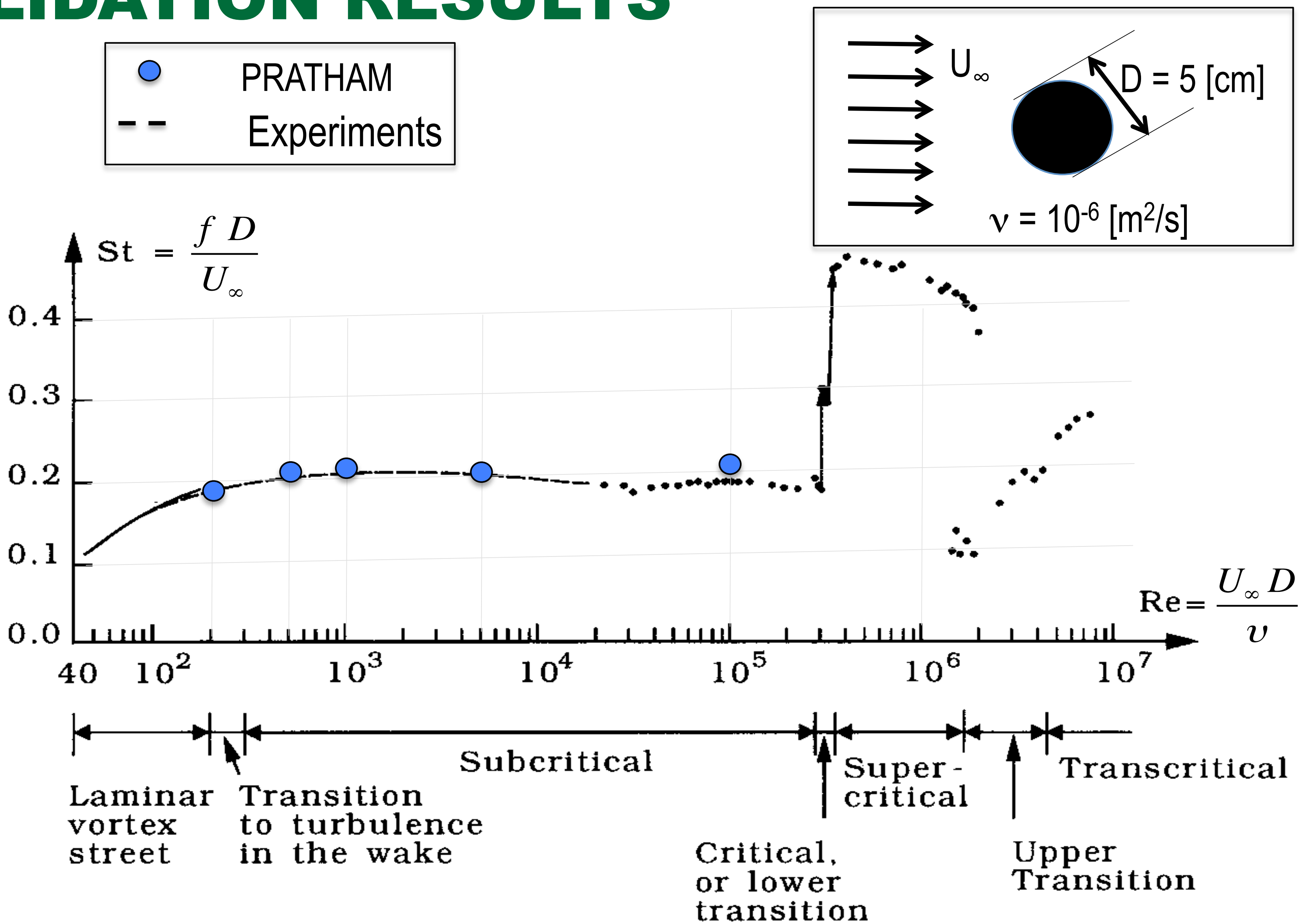
VORTEX SHEDDING FREQUENCY

Z-component of velocity



Lattice time-step

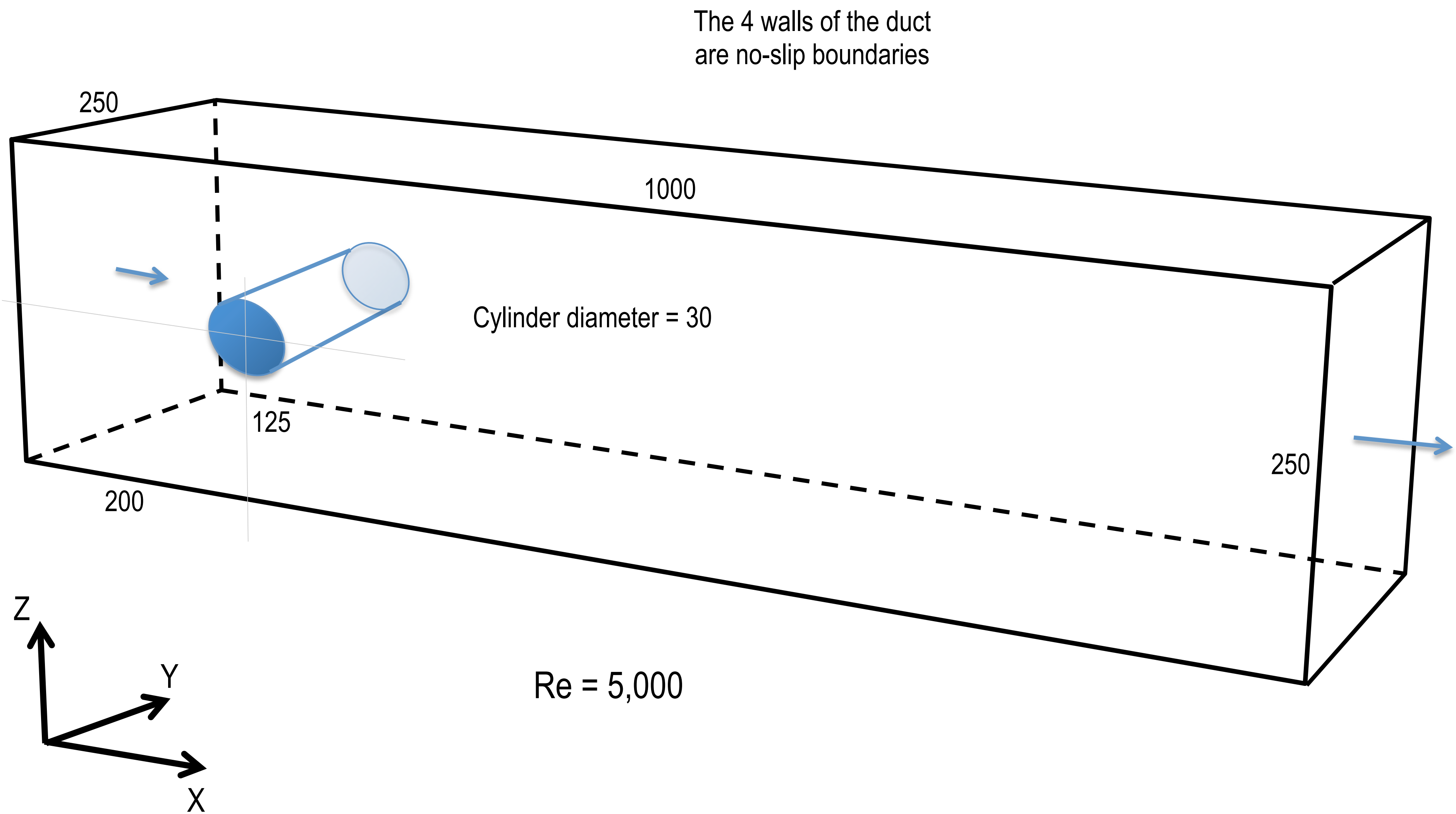
VALIDATION RESULTS



CASE STUDY 5

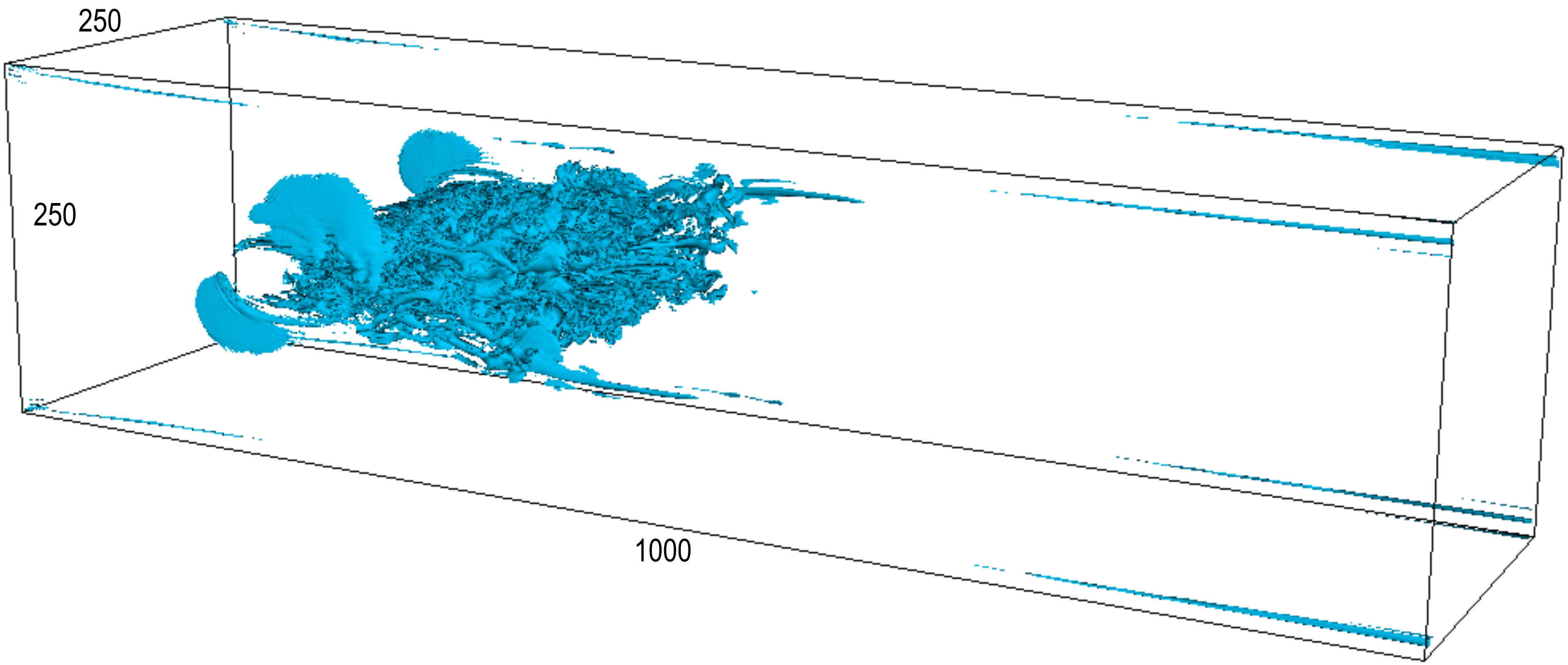
Flow around a 3D cylinder

FLOW AROUND A 3D CYLINDER



FLOW AROUND A 3D CYLINDER

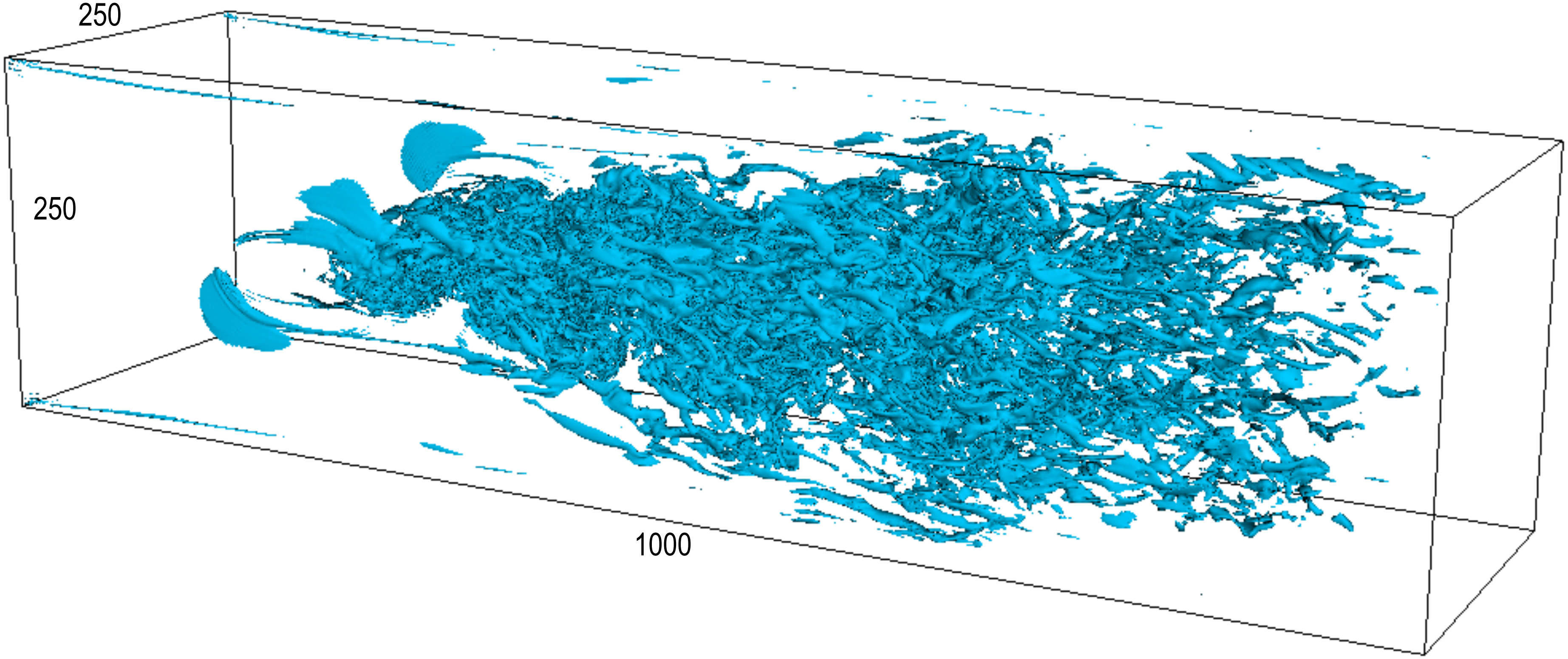
Iso-surface of Helicity = 5×10^{-4} is used to visualize turbulent flow structures in the wake



t = 5,000

FLOW AROUND A 3D CYLINDER

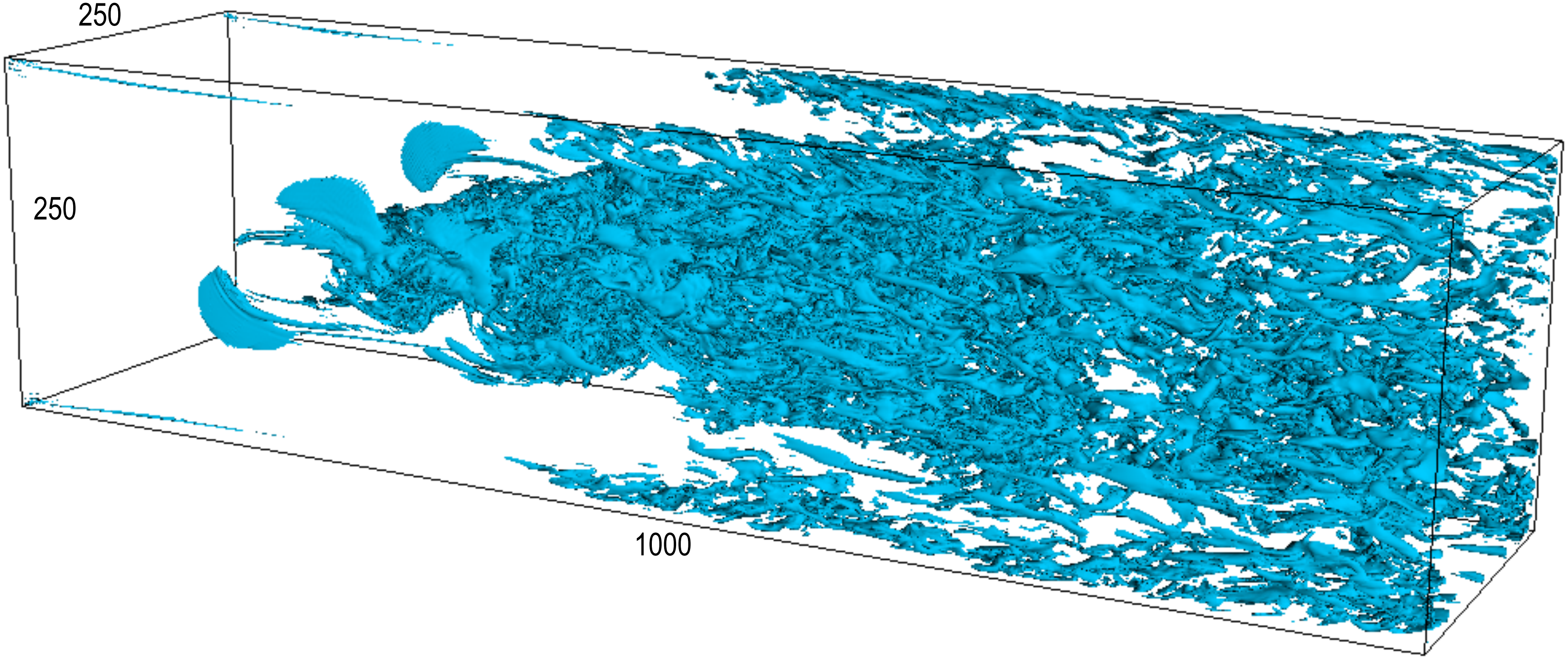
Iso-surface of Helicity = 5×10^{-4} is used to visualize turbulent flow structures in the wake



t = 10,000

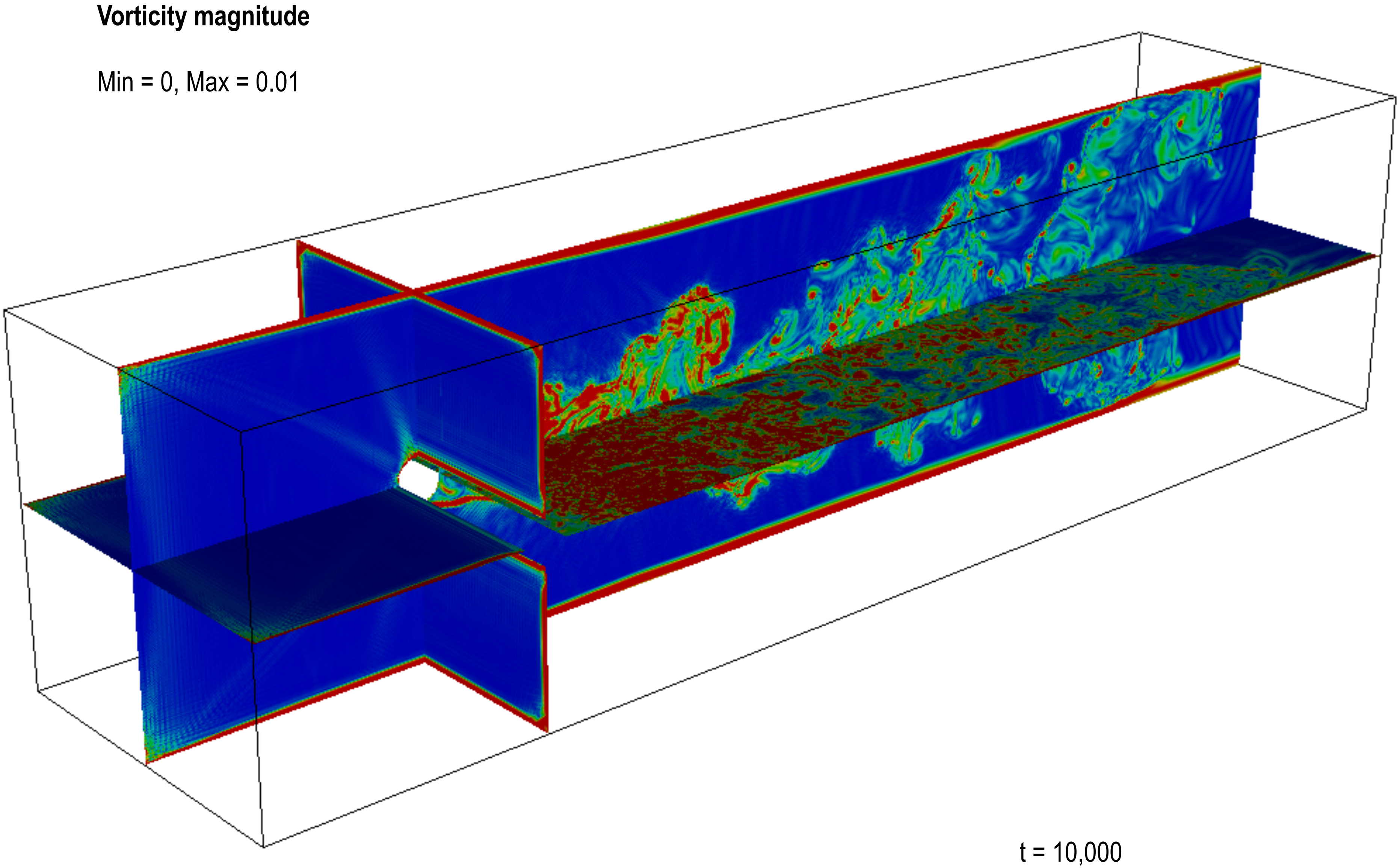
FLOW AROUND A 3D CYLINDER

Iso-surface of Helicity = 5×10^{-4} is used to visualize <u>turbulent flow structures</u> in the wake

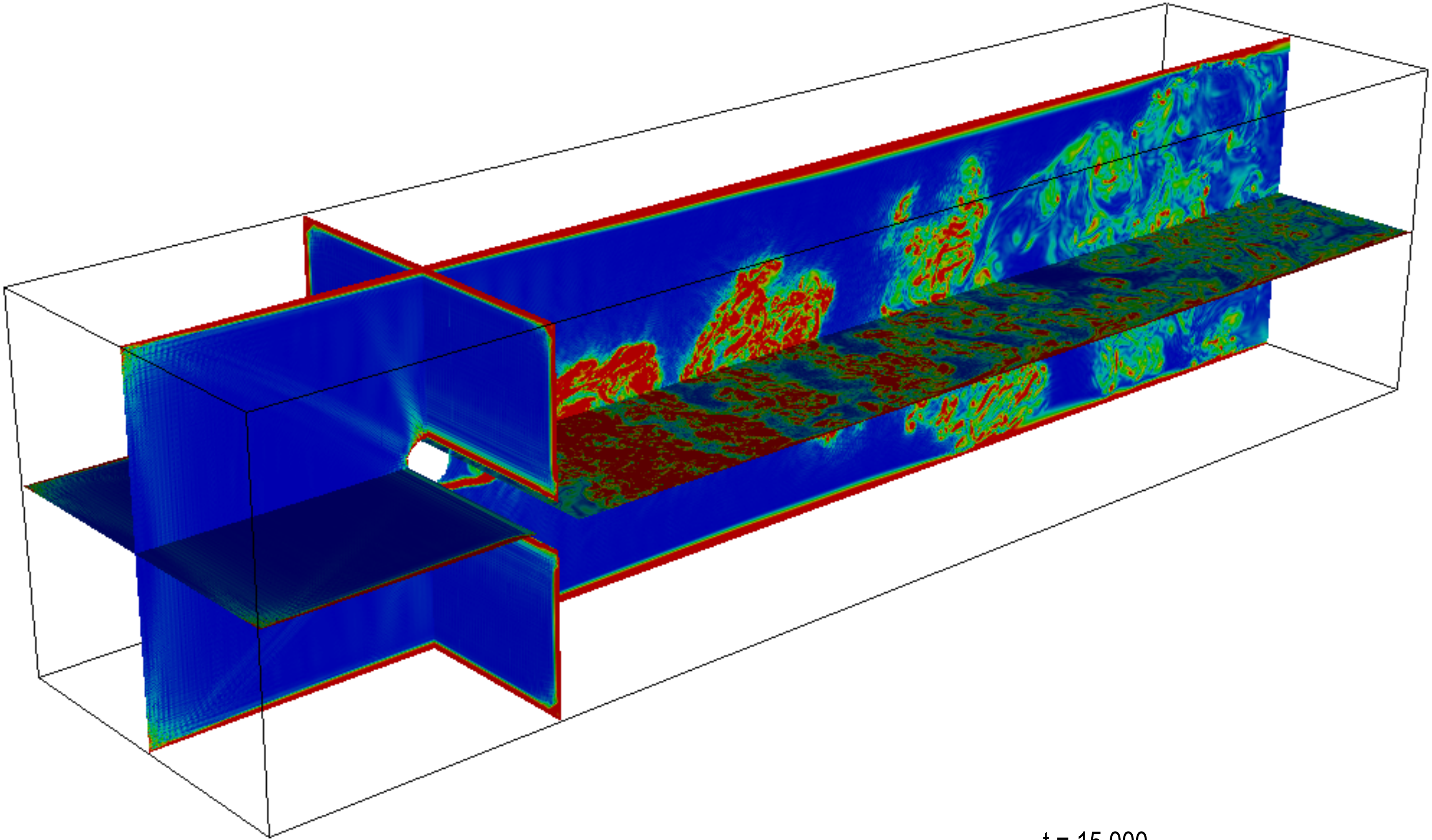


t = 15,000

FLOW AROUND A 3D CYLINDER

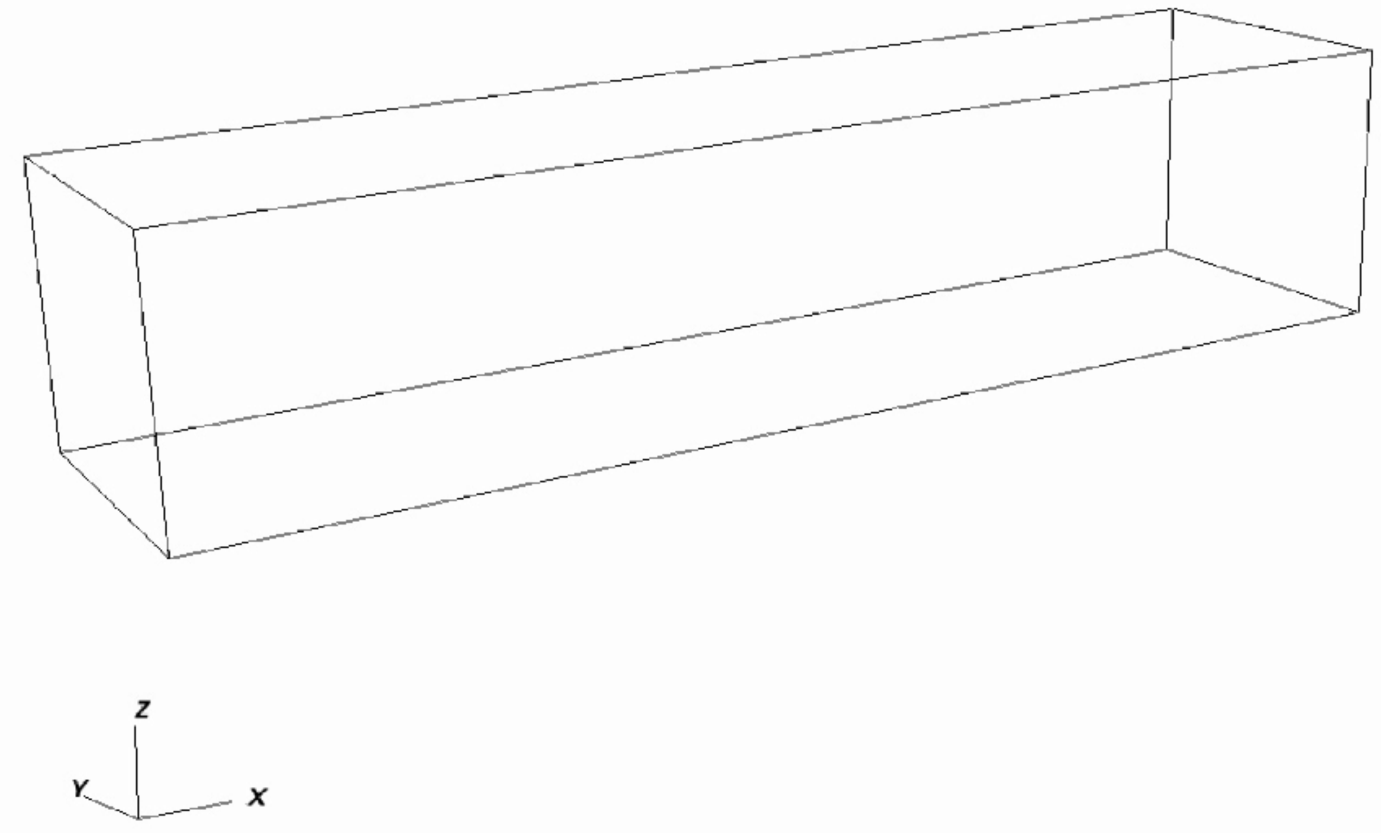


FLOW AROUND A 3D CYLINDER



t = 15,000

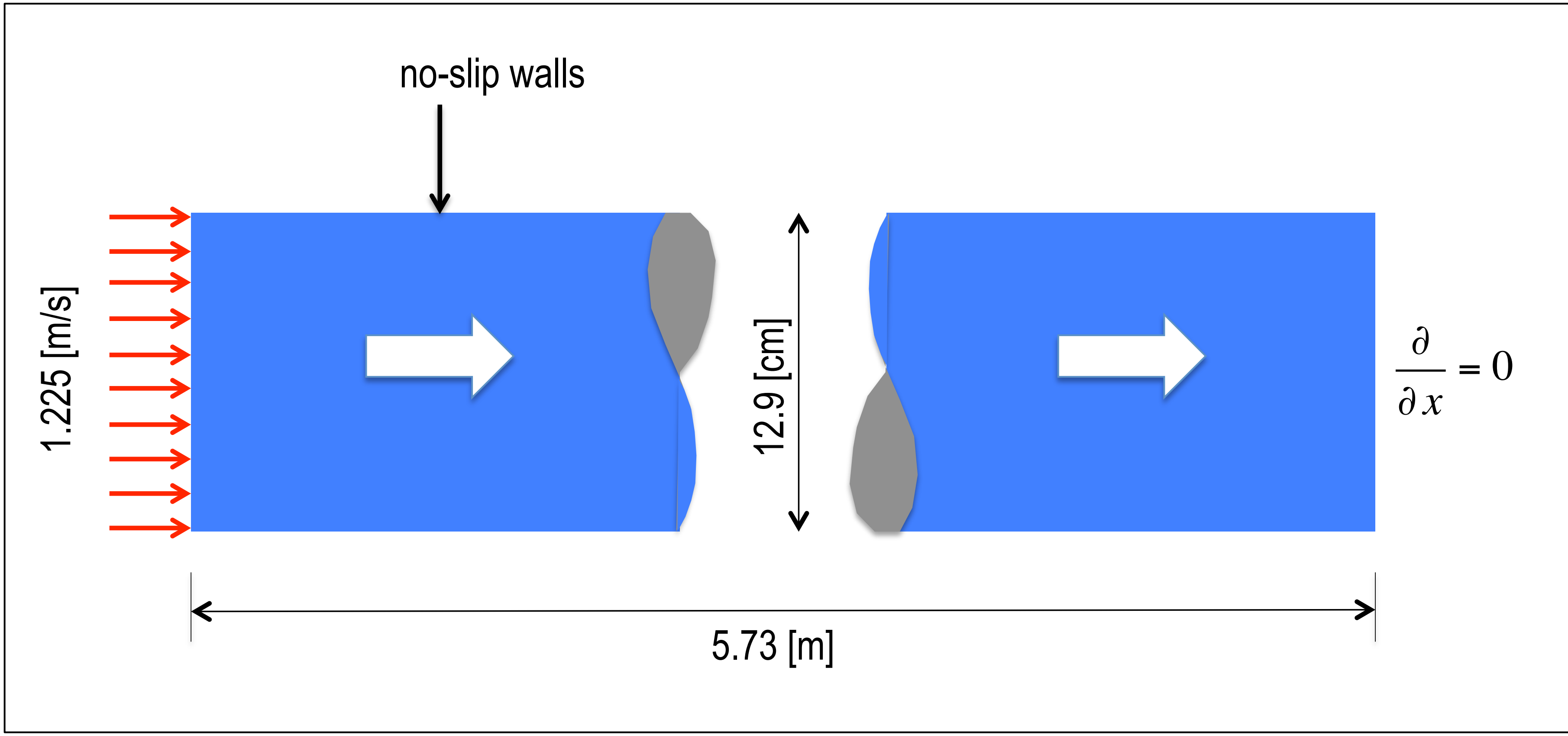
FLOW AROUND A 3D CYLINDER - MOVIE



CASE STUDY 6

Turbulent Flow inside a Long Cylindrical Pipe

FLOW INSIDE A LONG, CYLINDRICAL PIPE



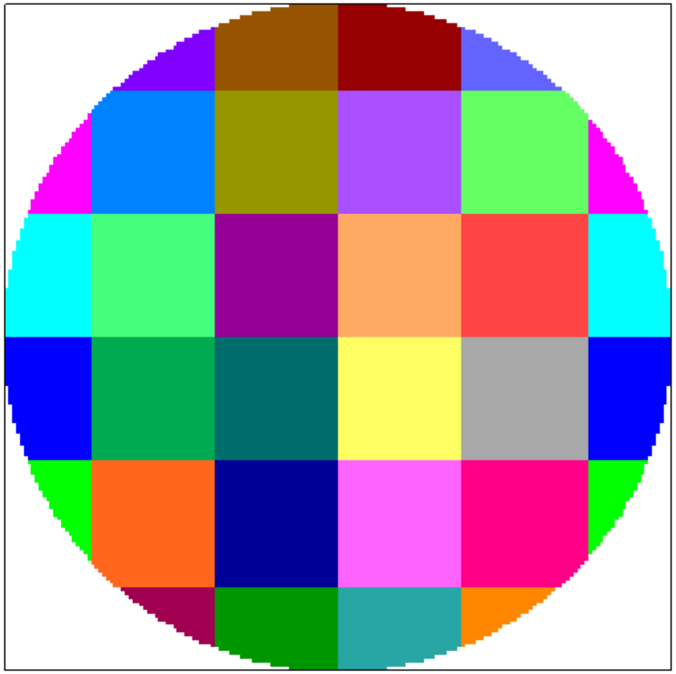
Working fluid = AIR
Kinematic viscosity = 1.58 E-05
Reynolds number = 31,577

FLOW INSIDE A LONG, CYLINDRICAL PIPE

JAGUAR run using 6912 cores

8000 x 200 x 200 lattice

(320,000,000 voxels)



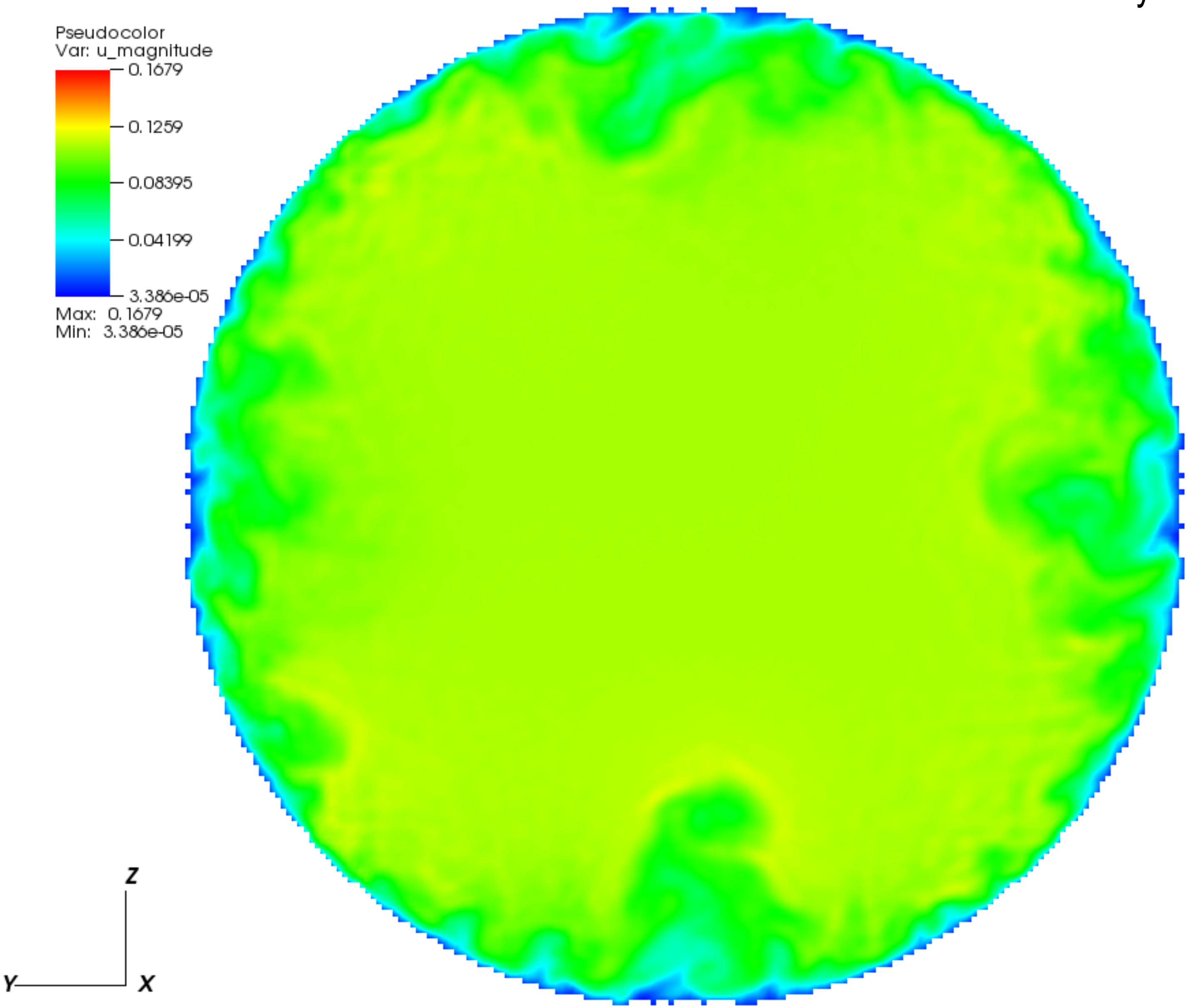
Domain decomposition 192 x 6 x 6

Simulation time = 10,000 time-steps

Clock time = 2.8 hours

PIPE FLOW RESULTS

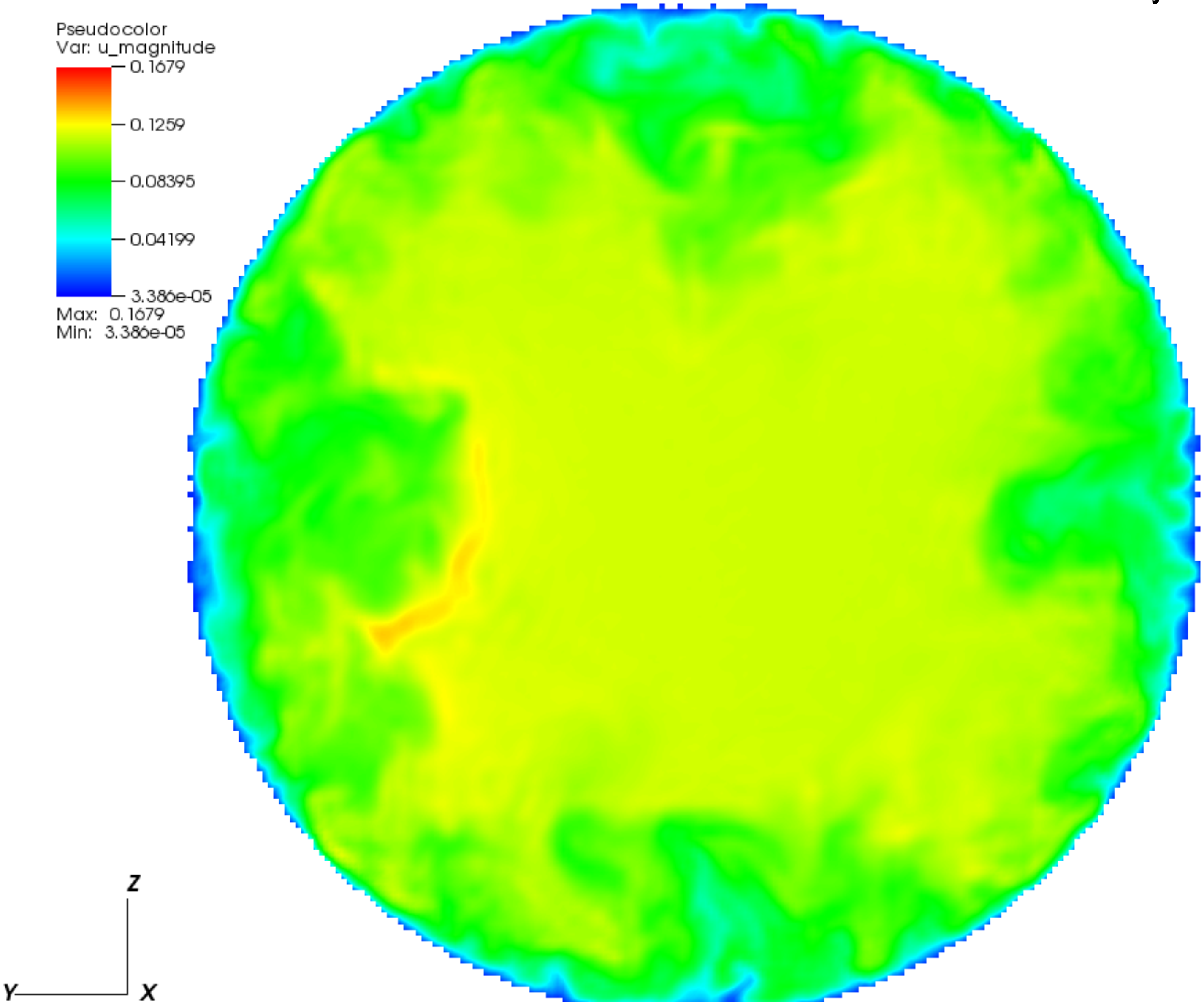
Velocity magnitude



10% of pipe length from inlet

PIPE FLOW RESULTS

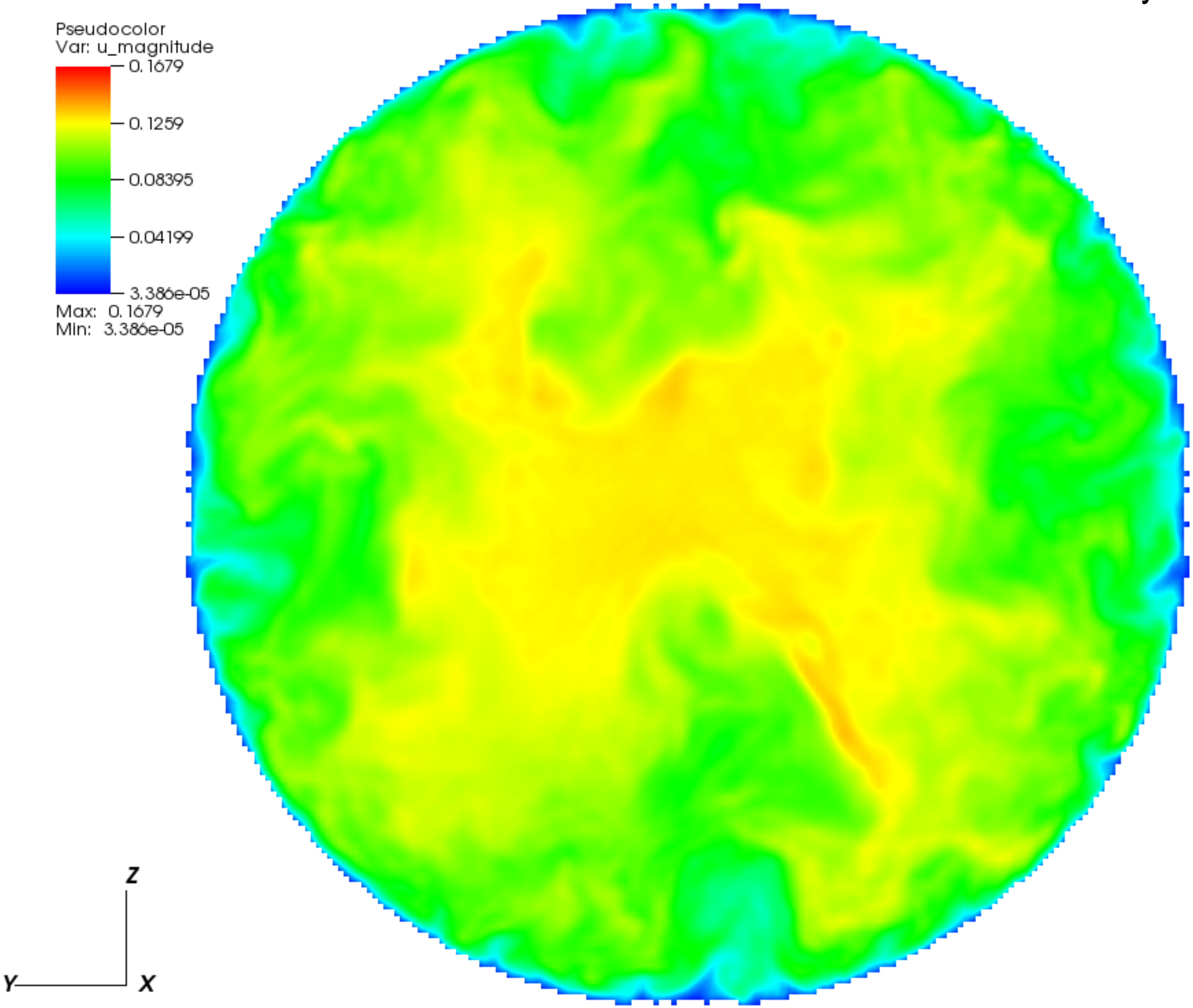
Velocity magnitude



20% of pipe length from inlet

PIPE FLOW RESULTS

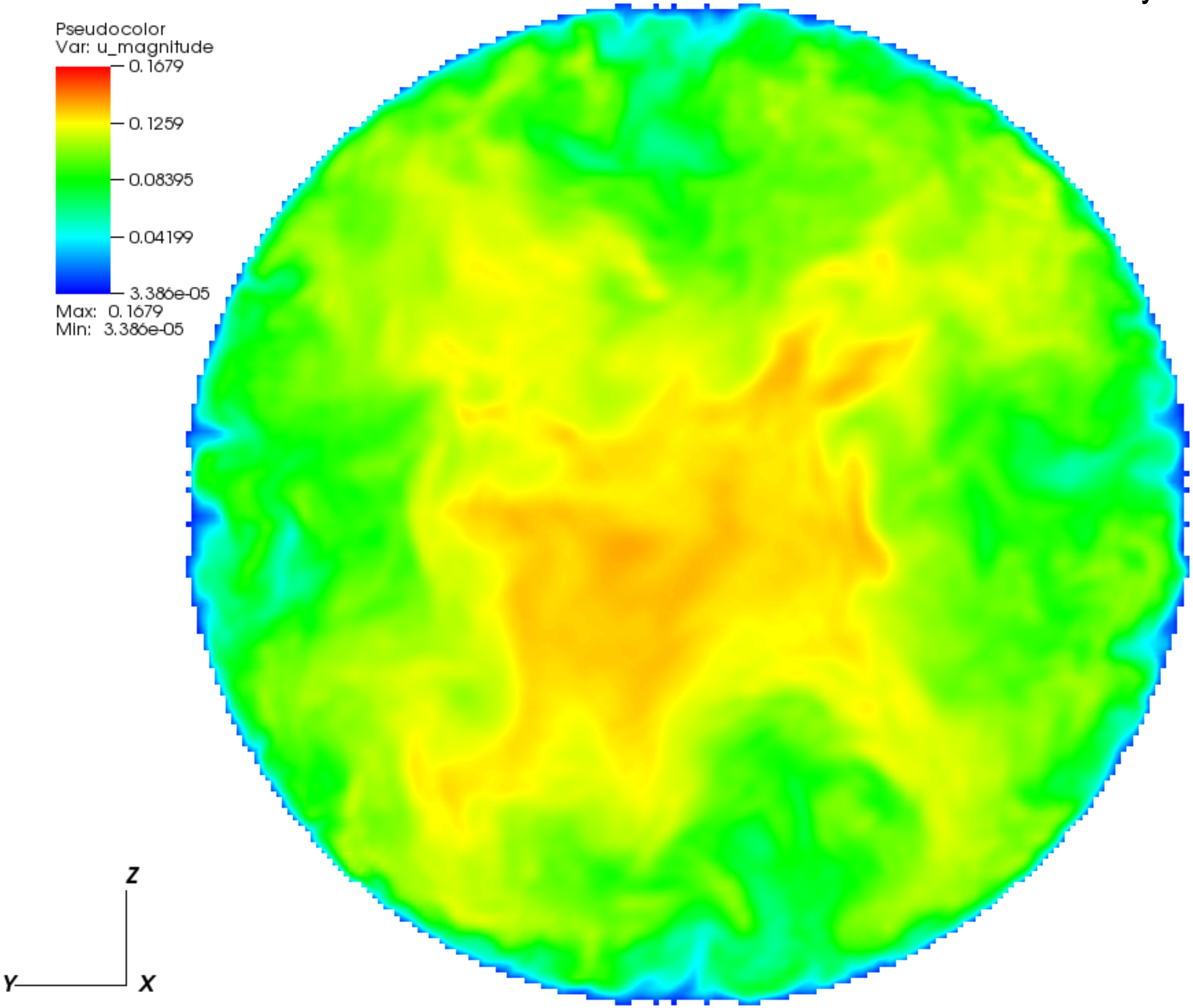
Velocity magnitude



30% of pipe length from inlet

PIPE FLOW RESULTS

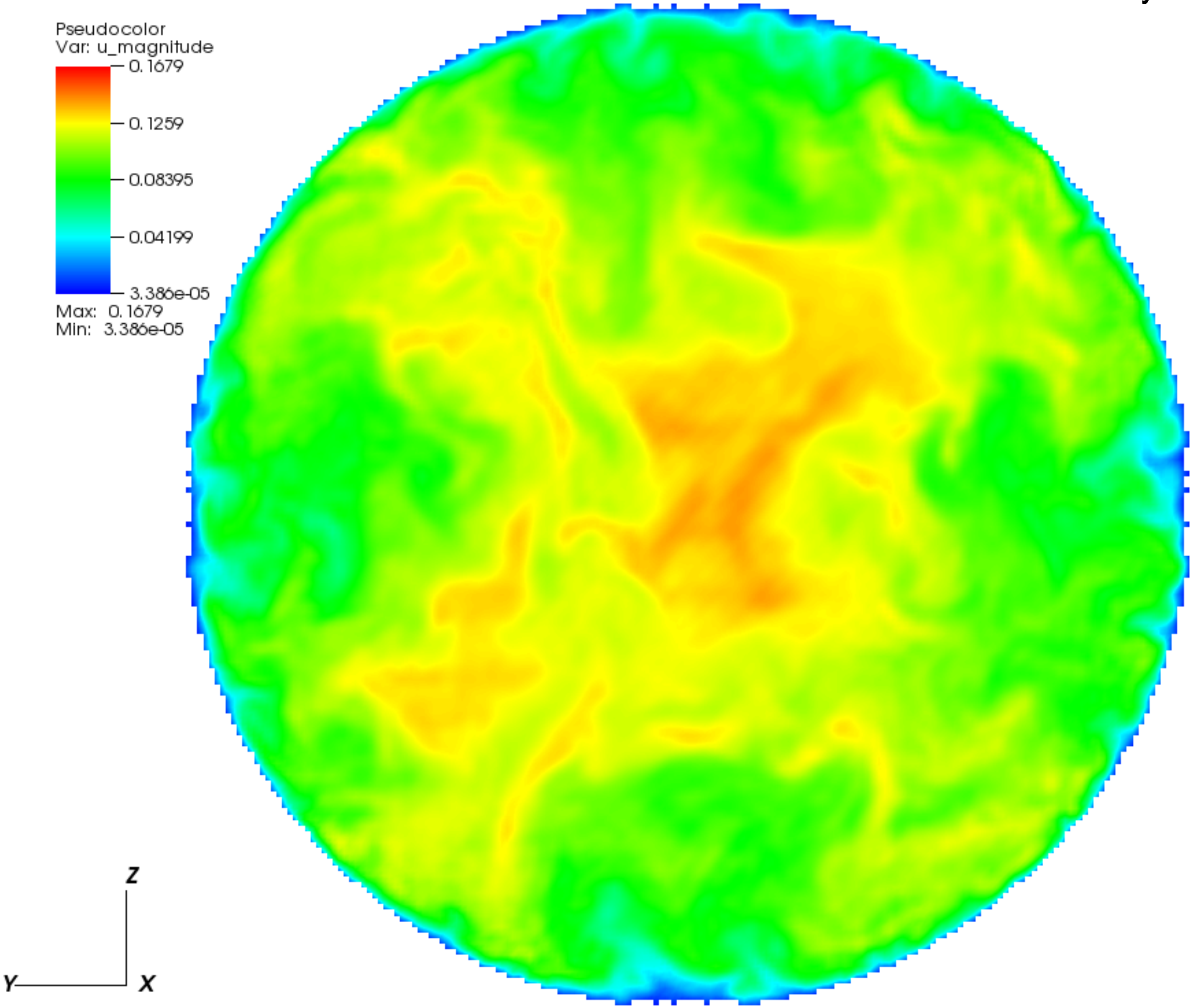
Velocity magnitude



40% of pipe length from inlet

PIPE FLOW RESULTS

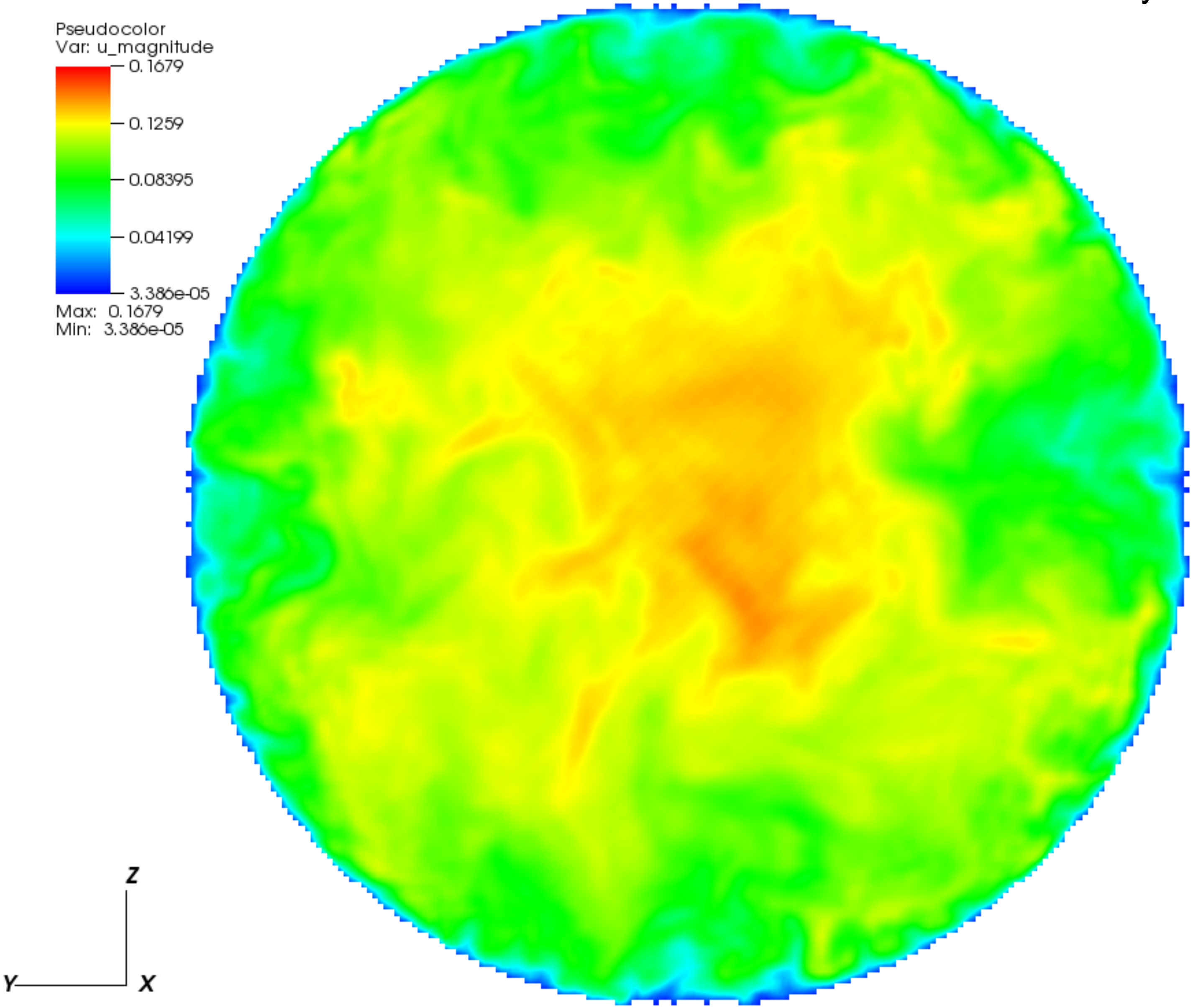
Velocity magnitude



50% of pipe length from inlet

PIPE FLOW RESULTS

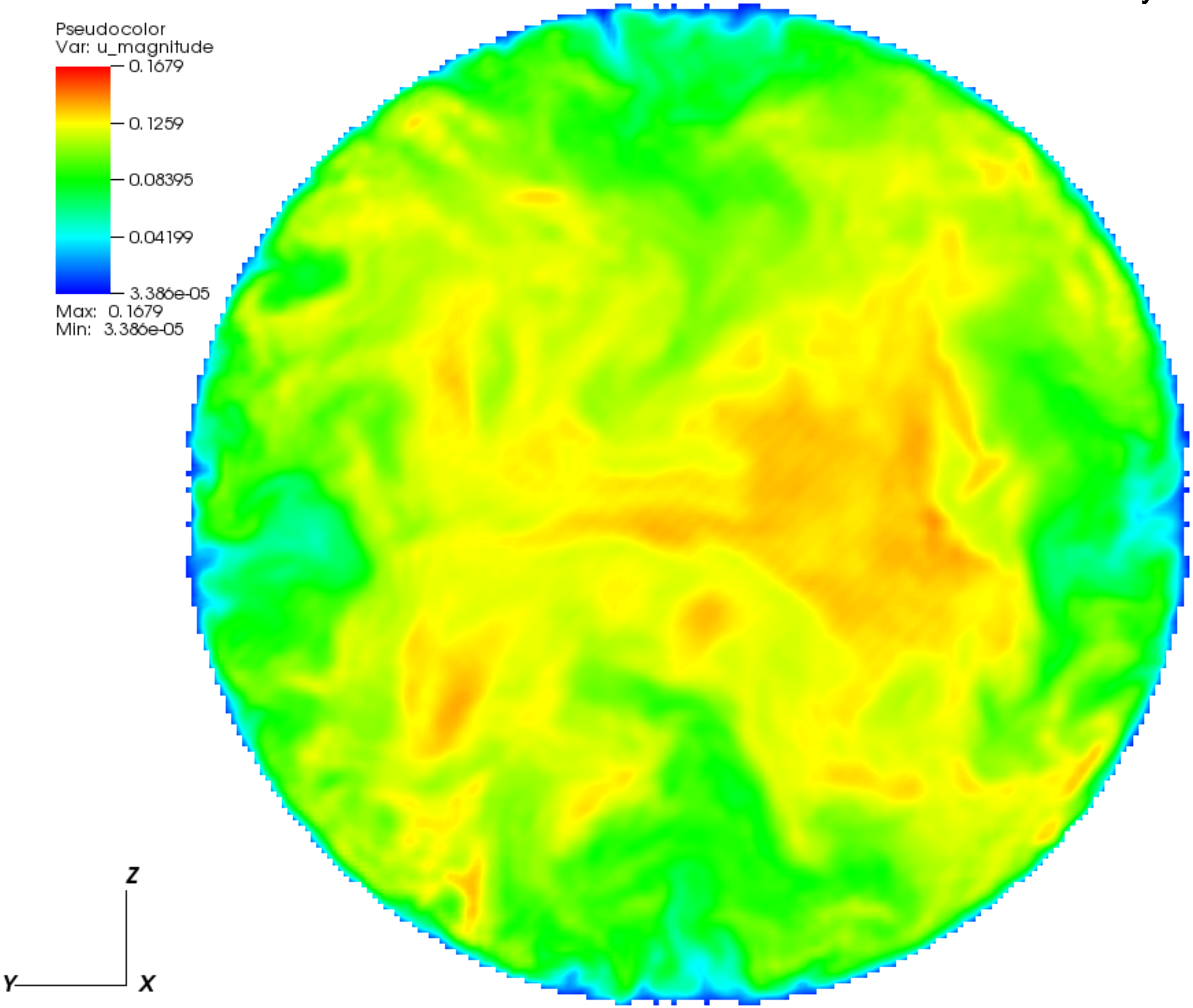
Velocity magnitude



60% of pipe length from inlet

PIPE FLOW RESULTS

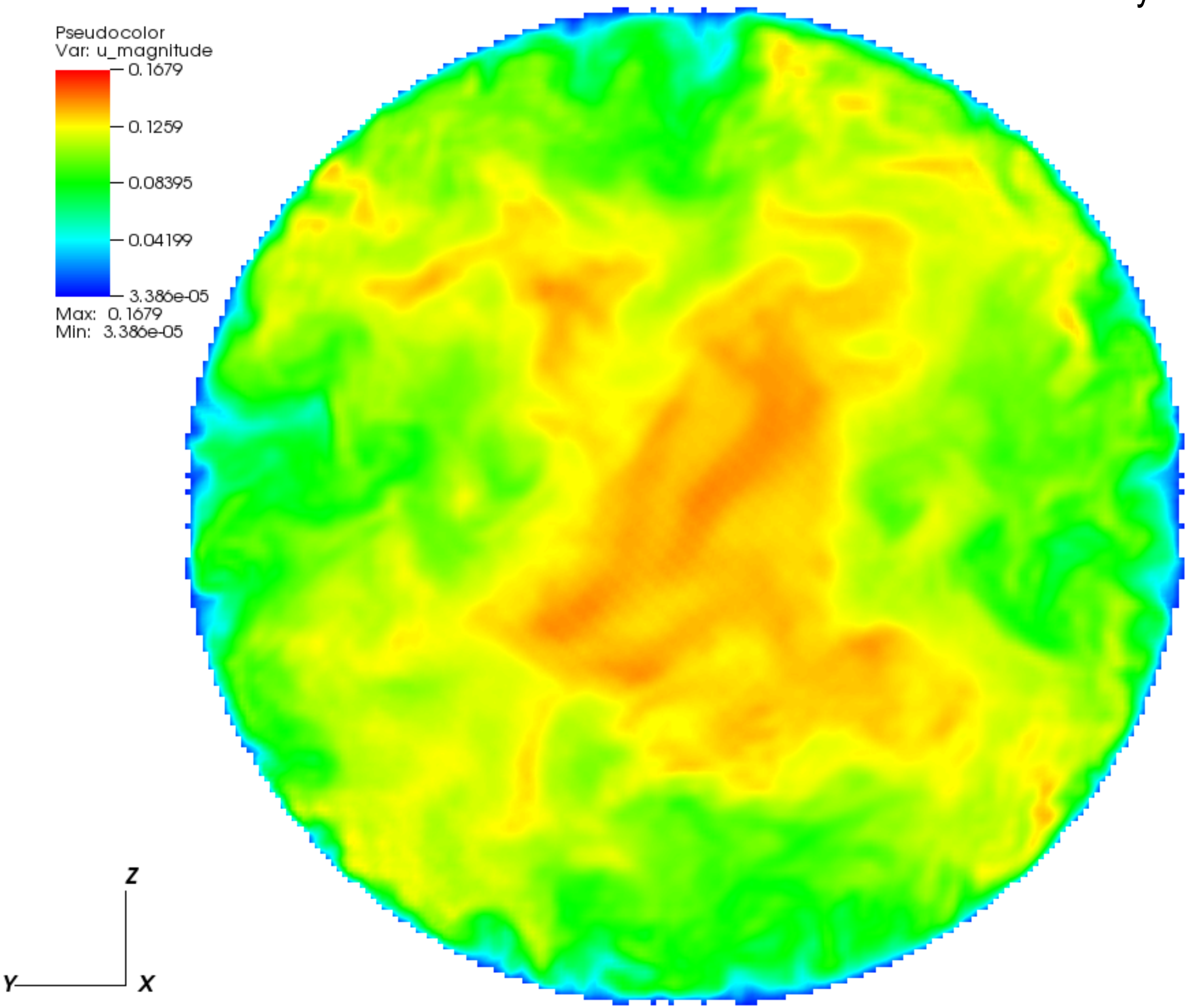
Velocity magnitude



70% of pipe length from inlet

PIPE FLOW RESULTS

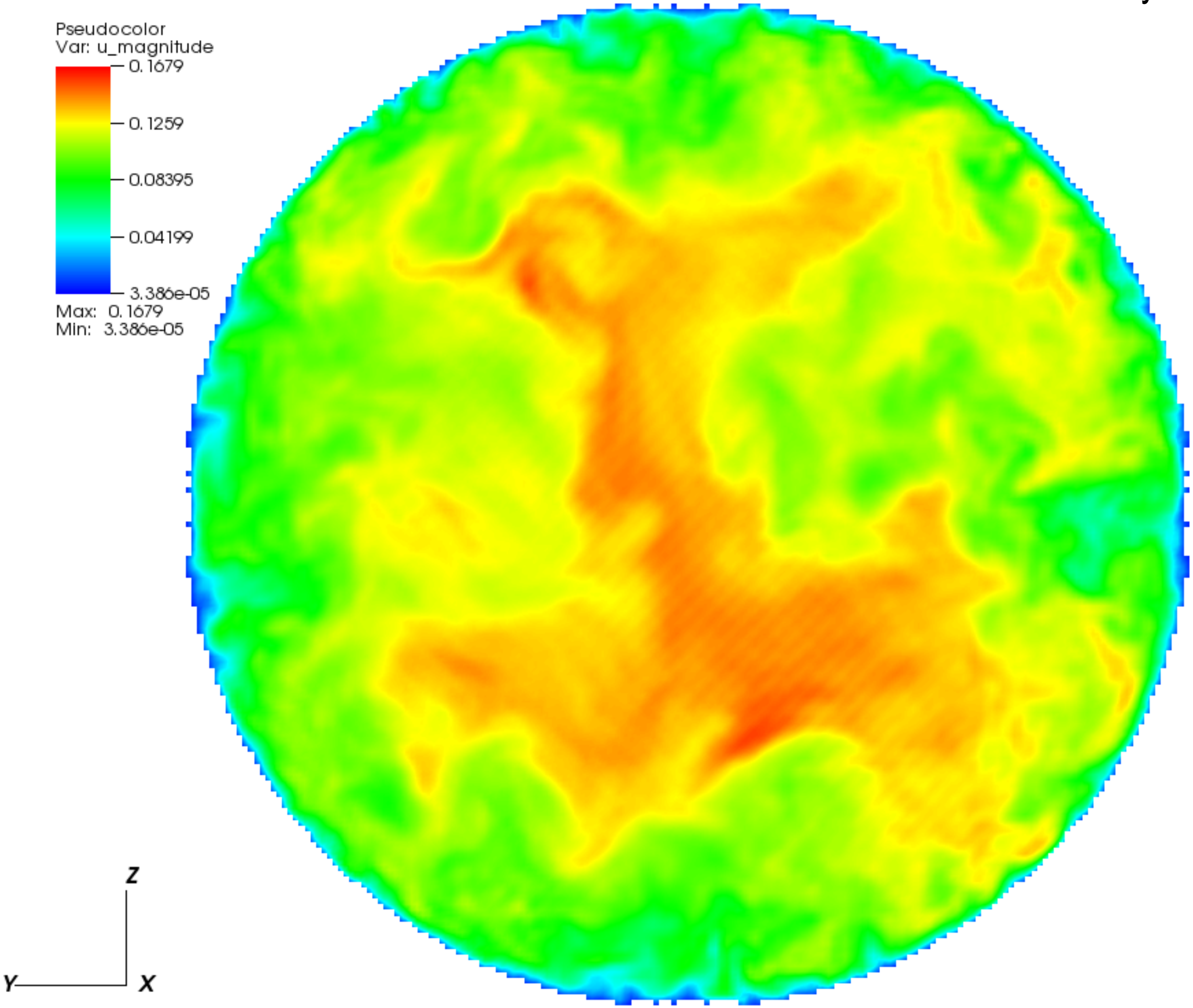
Velocity magnitude



80% of pipe length from inlet

PIPE FLOW RESULTS

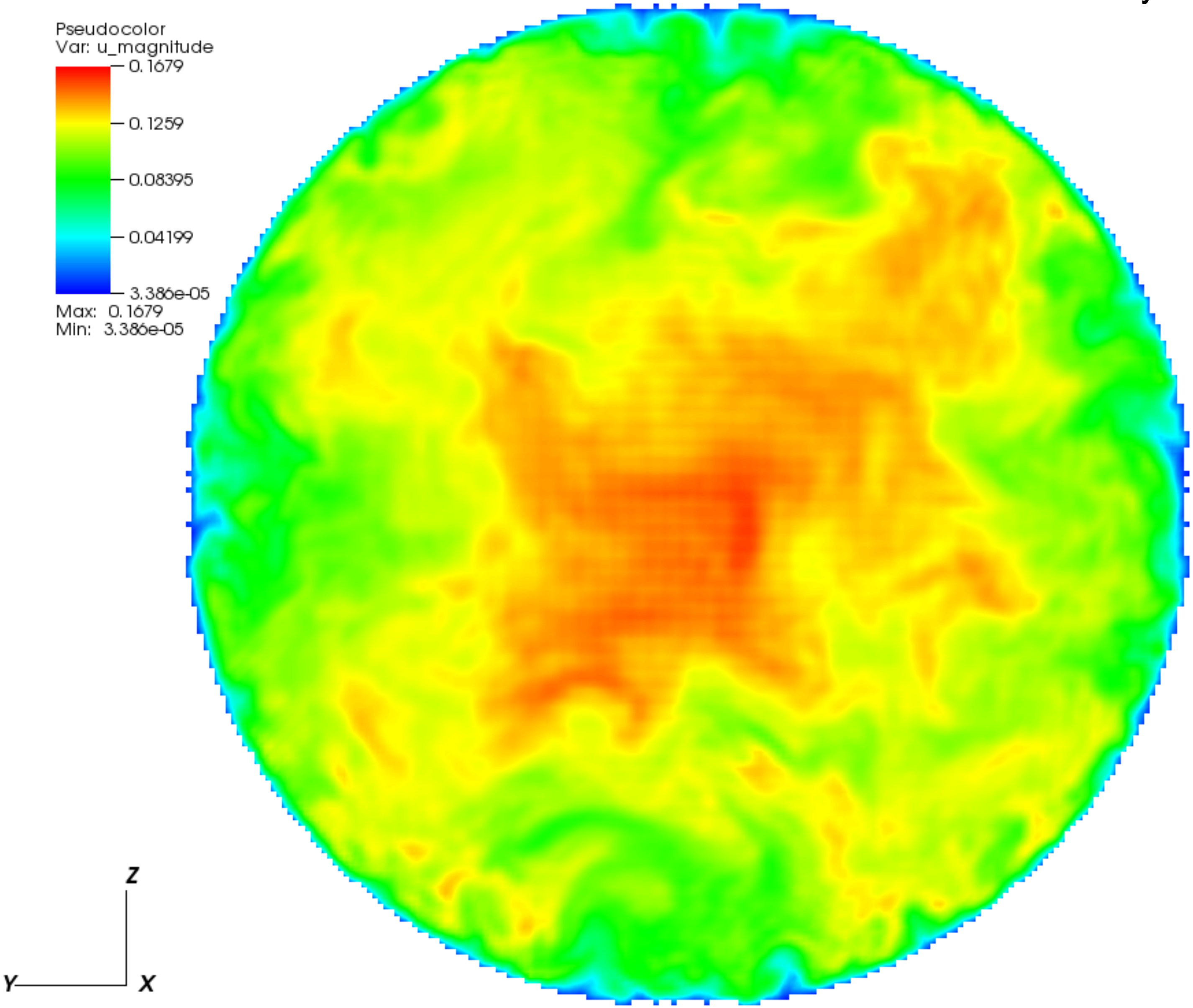
Velocity magnitude



90% of pipe length from inlet

PIPE FLOW RESULTS

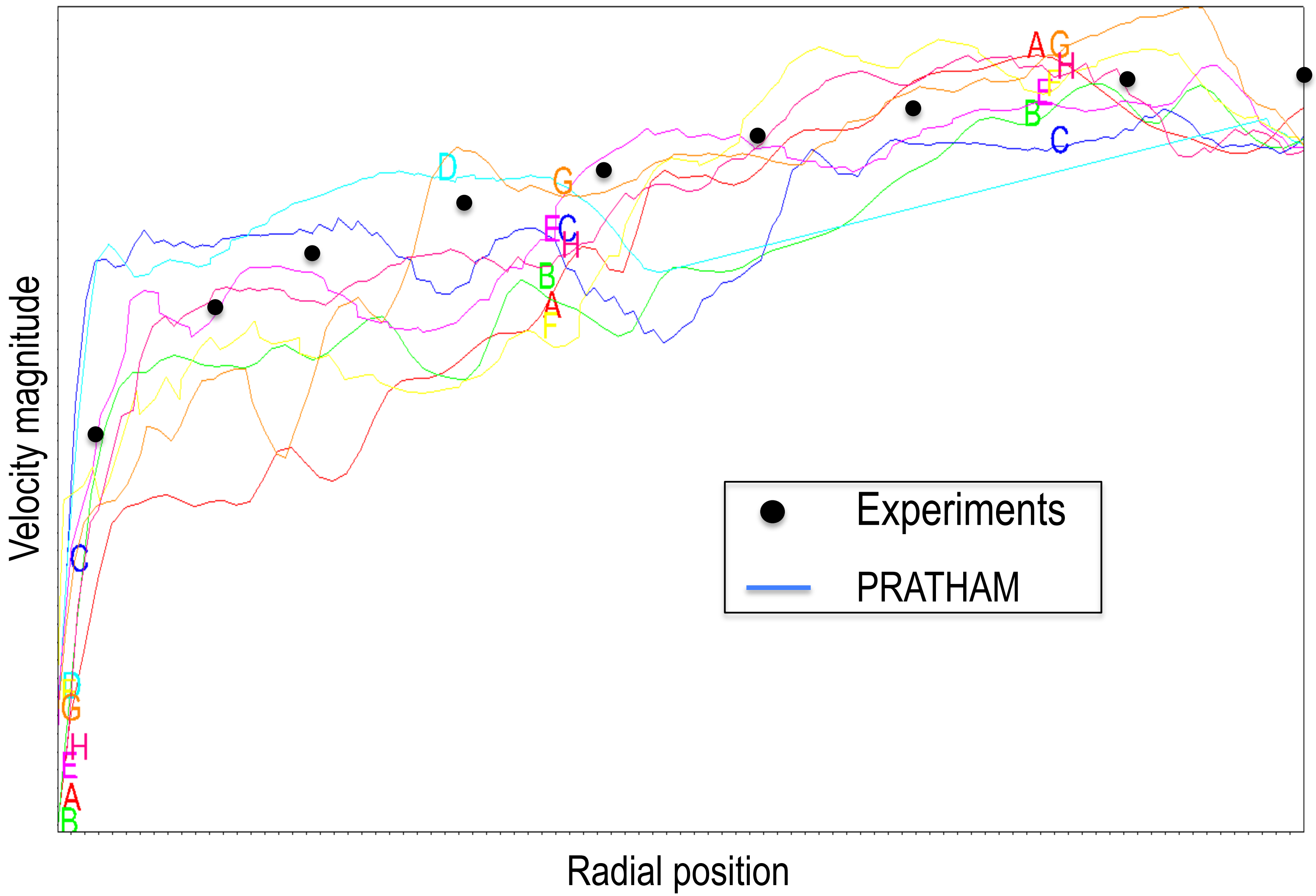
Velocity magnitude



99% of pipe length from inlet

VALIDATION WITH EXPERIMENTS

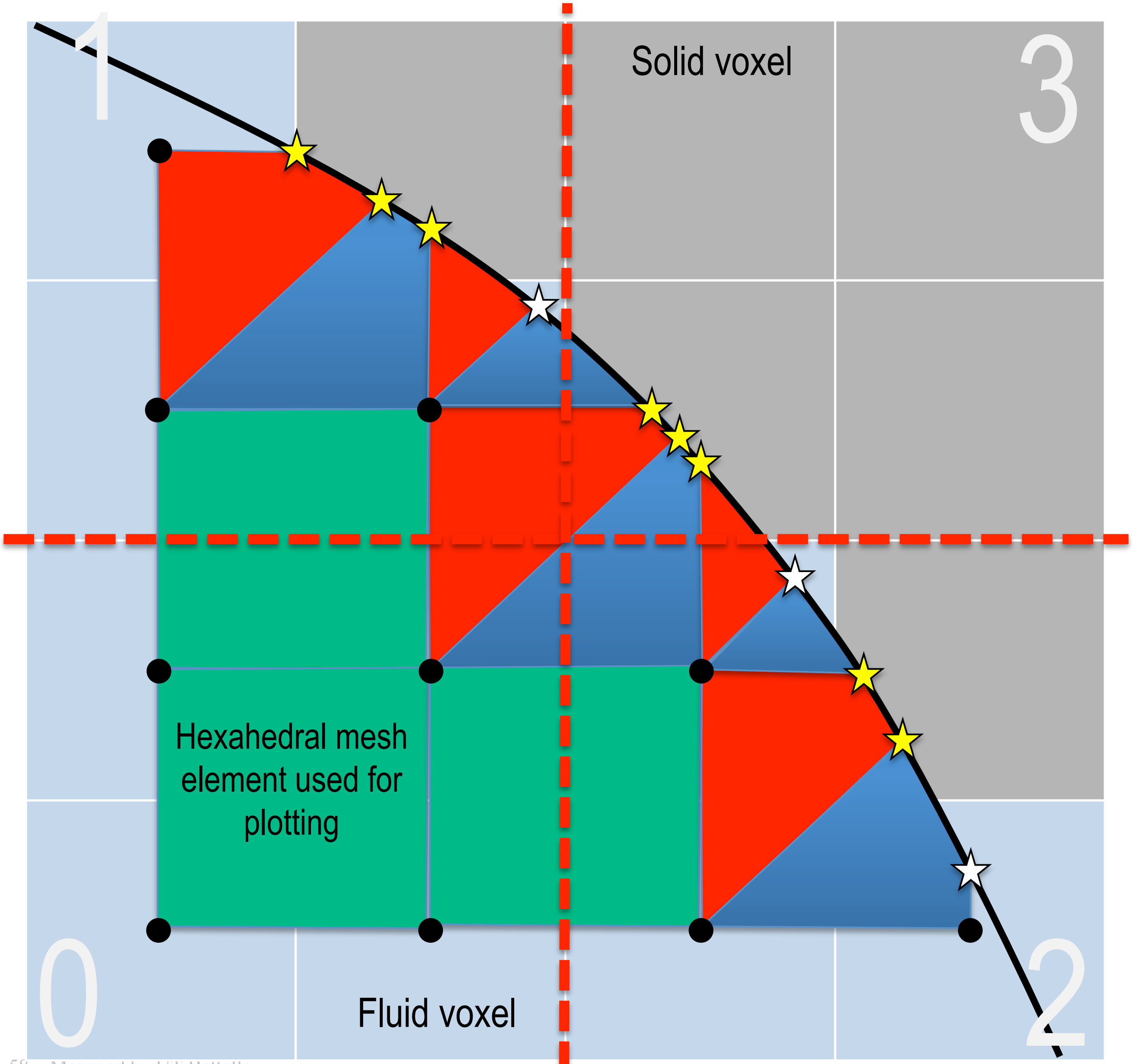
PIPE FLOW RESULTS



FUTURE DIRECTIONS

- Add in a capability to model curved surfaces without the staircase approximation
- Run the pipe flow case and collect statistics for turbulence analysis
- Run the spacer grid on JAGUAR / TITAN and compare results with experimental observations
- Change the PRATHAM I/O to deal with XDMF-HDF5 format
- Investigate the pros and cons of using PHDF5 (parallel HDF5) for I/O
- Add a feature to use non-uniform mesh for both CARTGEN++ and PRATHAM
- Add a GUI to both codes (CARTGEN++ and PRATHAM)
- Possibly convert PRATHAM from FORTRAN-90 to C++ and combine with CARTGEN++ to avoid the intermediate I/O step

WORK IN PROGRESS…



Modification of the BOUNCE-BACK rule to account for the exact location of the boundary

- ● Velocity and density values are available here for plotting purposes
- ★ ☆ Add new mesh elements near the boundary so that smoother plots are produced

Thank you for your attention.

Questions ?