A hybrid CPU-GPU Fast Poisson Solver for Direct Numerical Simulations

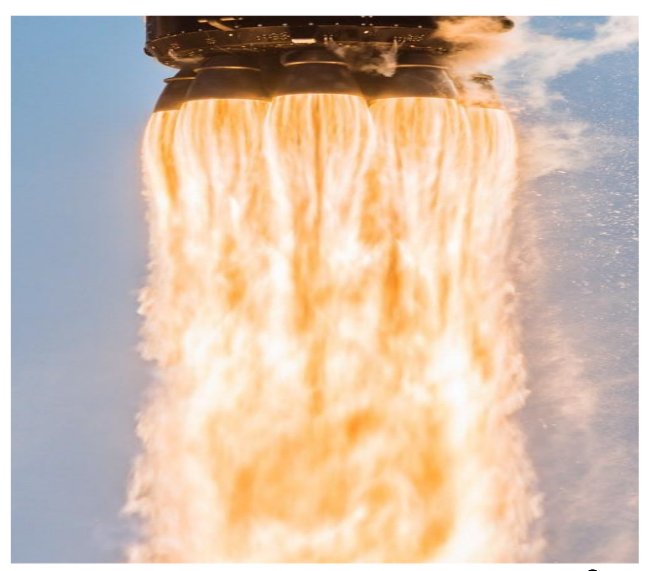
Neha Saripella, Emily Ku, Nithin Adidela

Computational Fluid Mechanics Lab (https://sites.uw.edu/cfmlab/)
William E. Boeing Department of Aeronautics & Astronautics

Motivation: Turbulent flows

Storm of turbulent gases in the Omega/Swan Nebula[1]
Scale: Light years; Re: ~10^{10}

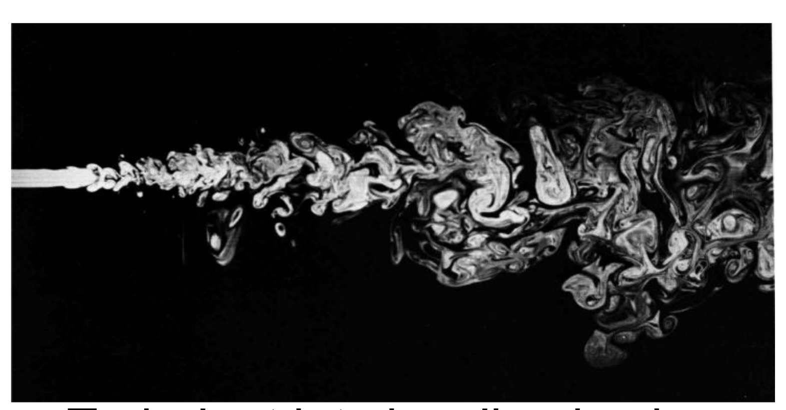
Eruption of Mount St. Helens on 18th May 1980[2]
Scale: ~1km, Re: ~10^6



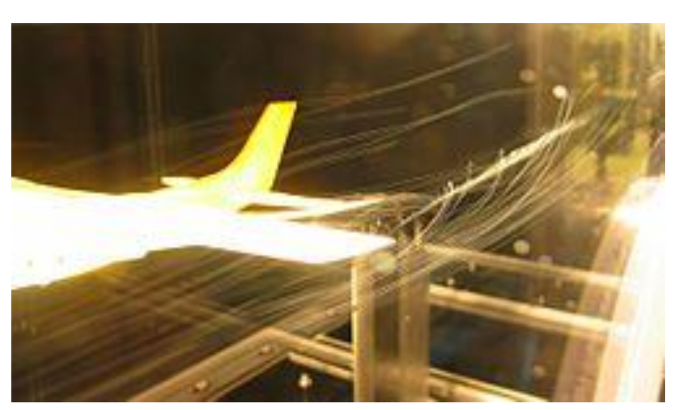
Falcon 9 Exhaust plume[3]
Scale: ~100m, Re: ~10^9

Aircraft wake turbulence[4]
Scale: ~100m, Re: ~10^7

Experimentally accessible flows



Turbulent jet visualized using Laser-induced fluorescence[5]



Wing-tip vortices visualized using helium-filled bubbles[6]

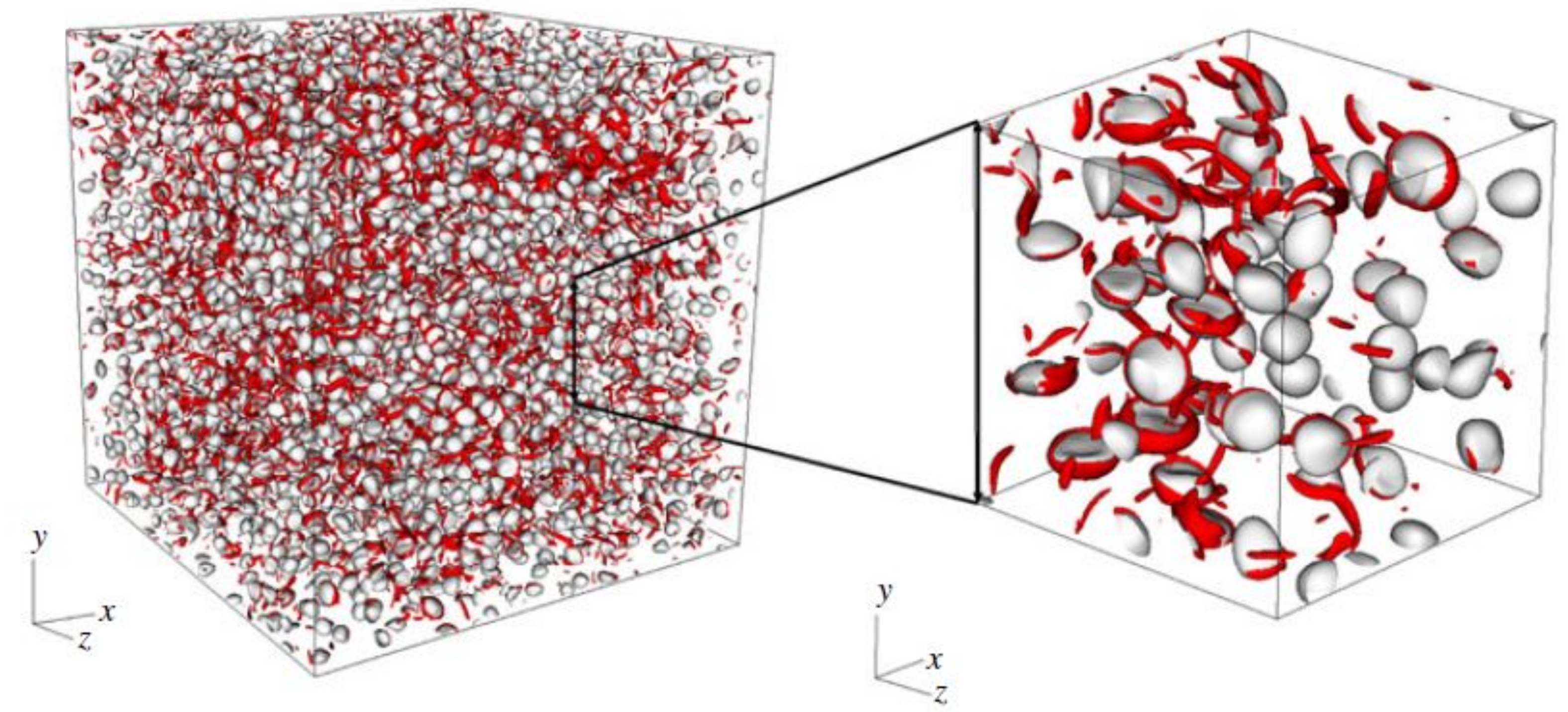
Experimentally inaccessible flows

- **Not feasible:** Stellar flows, volcanoes, cardiovascular flows, etc.
- **Expensive:** High Re turbulent flows in wind tunnels, flight tests, etc.
- **Unavailable**: Multiphase flows like particle-laden flows in clouds, droplet-laden flows in combustion chambers, etc.

Computational Fluid Dynamics (CFD)

CFD enables in-silico experiments with the following tools:

1) Direct Numerical Simulation (DNS) is a research tool that accurately solves the governing equations of the flow.
2) Reynolds Averaged Navier-Stokes (RANS) simulation is an industry standard tool to approximate turbulent flows.
3) Others (e.g., Large-Eddy Simulation (LES), Detached Eddy Simulation (DES), etc.)



DNS of 3130 droplets interacting with grid turbulence, simulating a spray combustion process to study atomization (Dodd & Ferrante *JFM* 2016)

Direct Numerical Simulation (DNS)

- DNS of high Reynolds number turbulent flows require high memory and petascale/exascale super computers.
- DNS solvers for incompressible flows like the pressure correction method often requires a solution to the Poisson equation at every time step.
  - E.g., Ansys Fluent, OpenFOAM, etc.
- The Poisson solver is the bottleneck of such codes.

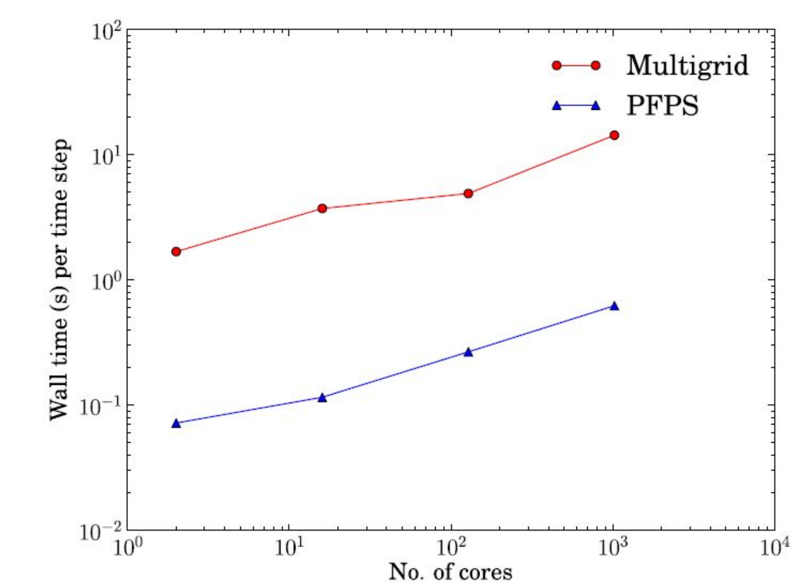
Poisson equation

$$\nabla^2 p^{n+1} = \nabla \cdot \left[\left(1 - \frac{\rho_0}{\rho^{n+1}} \right) \nabla p^* \right] + \frac{\rho_0}{\Delta t} \nabla \cdot \boldsymbol{u}^*$$

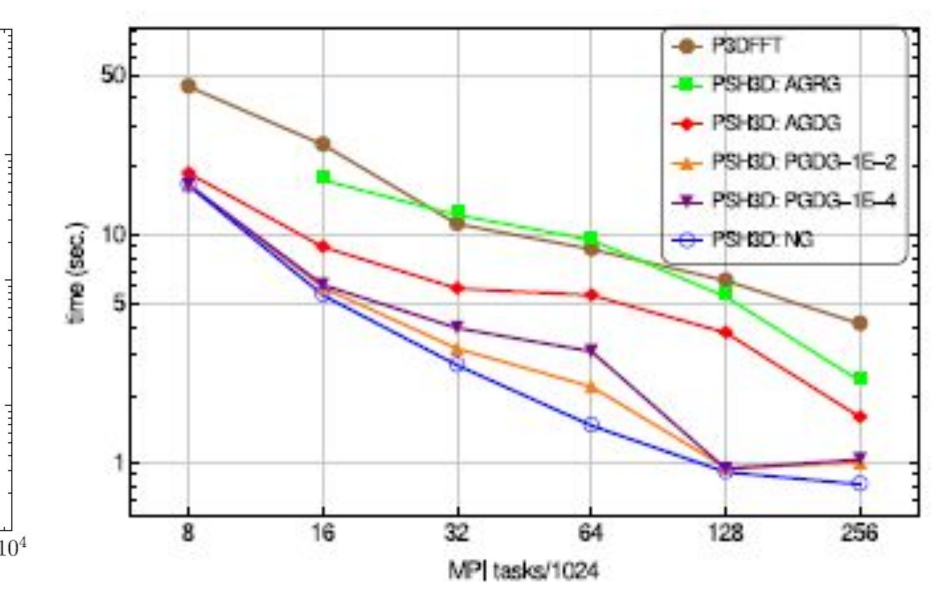
Poisson solvers

Poisson equation is solved using the following methods:

1) Direct methods involve constructing a linear system of eqns. and matrix inversion – computationally expensive.
2) Iterative methods are simple but converge slowly. Eg. Gauss-Jacobi, Multigrid method, etc.
3) Direct spectral methods with Fast-Fourier Transform for periodic domains – **Fast but difficult to 3D parallelize**

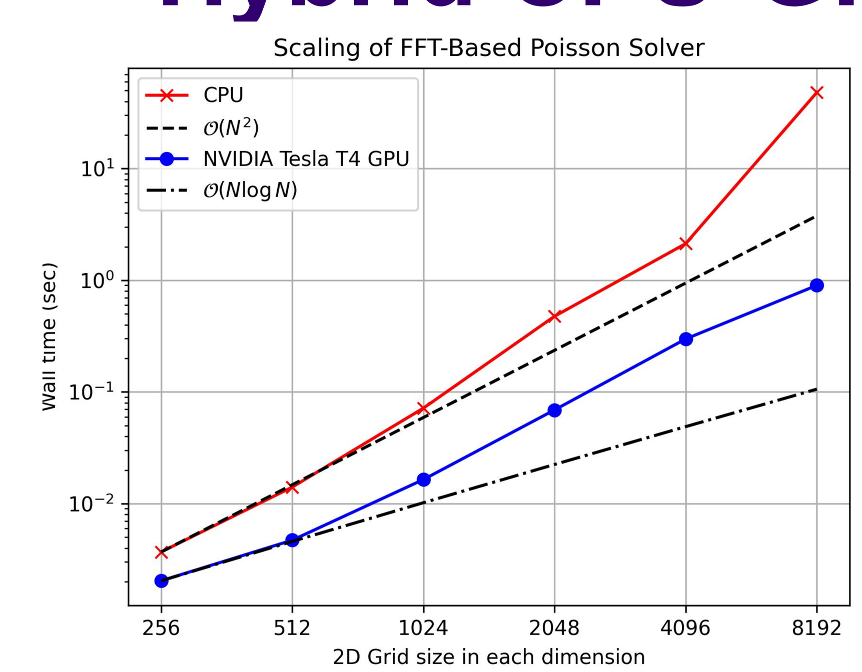


Weak scaling comparison of Multigrid method & Fast Poisson solver at $128^3/2$ grid points/core (Dodd & Ferrante *JCP* 2014)



Strong scaling performance comparison of various Fast Poisson solvers on an 8192^3 grid (Adams et al. *Parallel CFD* 2015)

Hybrid CPU-GPU Fast Poisson



- A 2D benchmark test case with periodic BCs is tested.
- FFT-based Fast Poisson on a CPU using NumPy and on an NVIDIA Telsa T4 GPU using CuPy
- Plan: Implement in 3D.
- Plan: Implement hybrid CPU-GPU Fast Poisson solver using MPI+CUDA.

Summary: Poisson solver is the bottleneck for DNS of turbulent flows. We explore a hybrid CPU-GPU Fast Poisson solver to improve performance.

References: [1]NASA, ESA, and J. Hester (ASU); [2]R. Krimmel, USGS Cascades Volcano Observatory; [3]Austin Sommars; [4]NASA Langley Research Center; [5] Dimotakis et al 1981; [6]Dale B. F. Rensselaer Polytechnic Institute