**Motivation & Contributions**

- **What are the limitations of today’s CFD simulations?**
  - Production CFD codes operate at 3–5% of the architectures peak performance
  - Deterioration in convergence rate of the numerical solver with
    - increasing problem size
    - large parallel partitioning
- **Why do we need an efficient CFD solver?**
  - To reduce the simulation times by taking advantage of the full potential of today’s computing powers
  - To enable new simulations currently impossible, by utilization of High Performance Computing
- **Contributions of this work:**
  - Targets an entire application, as opposed to a single stencil kernel
  - Presents a highly optimized and scalable parallel CFD solver
  - Maps well onto different platforms, with different memory hierarchy and CPU architectures

**Navier-Stokes Flow**

<table>
<thead>
<tr>
<th>Governing equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 )</td>
</tr>
<tr>
<td>( \frac{\partial \mathbf{v}}{\partial t} + \nabla \cdot (\mathbf{v} \otimes \mathbf{v}) = -\nabla p + \nabla \cdot \mathbf{S} + \mathbf{F} )</td>
</tr>
</tbody>
</table>

\( \mathbf{F} = \) Endo Flux + Inviscid Flux + Artificial Dissipation

**Implementation**

Our algorithm was designed to overcome the limitations of today’s simulations, with the goal of scalability on different multicore systems in mind.

**Verification**

We use the following simulations in order to verify the obtained results:

- **Laminar Step Flow**
  - Mach number 0.2
  - Grid Size 200x100
  - Steady (Reynolds number 50)

- **Unsteady (Reynolds number 250)**

**Performance Results**

**GFlops Improvement**

The following figures illustrate the improvements of Gflop per second with each optimization for the step flow on a grid with size 1000x500 cells.

The initial baseline code achieved 1.28 GFlops for double precision and 2.18 GFlops for single precision.

**Simulation Speedup**

The following figures illustrate the speedup achieved for the execution time of a full simulation, with a residual drop of 3 orders of magnitude.

**Future Work**

As future work, we plan to continue with this trend in improving these simulations, and add more features to our solver. The following are some of our future goals:

- **Parallel Implementation of Multigrid**
- **Supporting Turbulence Models**
  - This is essential to capture the physics in numerous real-world applications
- **Heterogeneous Algorithms**
  - Computational resources are wasted when either CPU or GPU is left idle.
  - We plan to partition the grid to utilize the entire system to its full potential
- **Distributed Implementation**
- **Simulations on Supercomputers**
  - Our ultimate goal is to enable new simulations using supercomputers