A Projection Method Based Fast Transient Solver for Incompressible Turbulent Flows

Chris Sideroff

08 June 2015
Applied CCM

- Specialize in the application, development and support of OpenFOAM®-based software
- Creators and maintainers of Caelus
- Locations: Canada, Australia, USA
Motivation

Why develop another transient solver?

- DES and LES attractive because RANS tends to be problem specific
- Low cost hardware + open-source software ⇒ DES and LES feasible
- Traditional transient, incompressible algorithms (PISO and SIMPLE) do not scale well for large HPC, GPU and Many Integrated Core (MIC) environments
- Let’s review PISO algorithm
PISO Overview

Pressure Implicit with Splitting of Operators (PISO)\(^1\) method:

1. Solve momentum equation (*predictor step*)
2. Calculate intermediate velocity, \(u^*\) (pressure dissipation added)
3. Calculate momentum fluxes
4. Solve pressure equation:
   \[ \nabla \cdot \left( \frac{1}{A_p} \nabla p \right) = \nabla \cdot u^* \]
5. Correct momentum fluxes
6. Correct velocity (*corrector step*)

Repeat steps 2–6 for PISO (1–6 for transient SIMPLE)

Fractional Step Error

- Step 2 main issue with PISO
- Predicted velocity used only to update matrix coefficients:
  \[ u^* = \frac{1}{a_p} \left( \sum a_{nb} u_{nb} - (\nabla p - \nabla p) \right) \]
- Pseudo-velocity, \( u^* \), is used on the RHS of pressure equation
- Therefore requires at least two corrections to make velocity and pressure consistent
Pressure Matrix

- Non-constant coefficients \( \frac{1}{a_p} \) in pressure matrix affects multi-grid solver performance
- Multi-grid agglomeration levels cached first time pressure matrix assembled
- Coefficients \( \frac{1}{a_p} \) only valid for the first time step
- Turning off caching of agglomeration too expensive
SLIM Overview

Semi Linear Implicit Method (SLIM), based on projection method\(^1\): decompose velocity into vortical and irrotational components.

1. Solve momentum equation (vortical velocity)
2. Calculate momentum fluxes (pressure dissipation added)
3. Solve pressure equation (irrotational velocity):
   \[ \Delta t \nabla^2 (p) = \nabla \cdot u \]
4. Correct momentum flux
5. Correct velocity (solenoidal)

Use incremental pressure approach to recover correct boundary pressure

Fractional Step Error

- Velocity split into vortical and potential components - much smaller fractional step error
- Pressure and velocity maintain stronger coupling
- Continuity satisfied within one pressure solve because predicted velocity used directly in pressure equation
Pressure Matrix

- Pressure matrix coefficients purely geometric
- Multi-grid agglomeration levels assembled during first step now consistent for all time steps
- Significantly improves parallel scalability for multi-grid solver
2D Periodic Hills

- Two dimensional, stream-wise, staggered hills of polynomial shape
- $Re_h = 10,595$
- Stream-wise and span-wise boundaries periodic. Hills and top boundaries no slip.
- Grid: $\sim 4.5$ million hex cells; LES model: Smagorinsky
Validation

- Experimental data of Rapp (2009)
- Mean and second moment components at 10 vertical rakes
x/h = 2

- Both compare favorably
- SLIM slightly closer than PISO
\( x/h = 4 \)

- SLIM consistently closer than PISO at all locations
- Likely due to lower fractional step error
Simulation Time

- SLIM on average about 30% faster on modest HPC system
- Fewer total iterations of pressure equation (SLIM: 10; PISO: 14)

<table>
<thead>
<tr>
<th># cores</th>
<th>PISO</th>
<th>SLIM</th>
<th>% diff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2095</td>
<td>1550</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>988</td>
<td>711</td>
<td>28</td>
</tr>
<tr>
<td>10</td>
<td>419</td>
<td>302</td>
<td>28</td>
</tr>
<tr>
<td>20</td>
<td>330</td>
<td>231</td>
<td>30</td>
</tr>
<tr>
<td>40</td>
<td>219</td>
<td>147</td>
<td>33</td>
</tr>
<tr>
<td>60</td>
<td>216</td>
<td>138</td>
<td>36</td>
</tr>
</tbody>
</table>
Precursor Simulation

- Establish turbulent conditions to use as initial condition for wind park simulation
- Start from quiescent condition. Run until fully turbulent.
- Steam-wise and span-wise periodic
- Grid size: 50 million hex cells
- Results courtesy of Greg Oxley at Vestas using Firestorm super computer
Mean Wind Profile

- SLIM slightly more accurate than PISO
- Fully turbulent condition reached sooner than PISO
Scaling

- Consistent multi-grid agglomeration levels give SLIM significant advantage
MPI Profiling

- Profiled MPI calls on 125 million cell mesh up to 4096 cores

![Bar Chart]

Total number of MPI calls
- SUM: 1.6E+11
- PISO: 2.7E+11 (4096 cores), 1.3E+11 (2048 cores)
Future Work

- For static grids, pressure matrix construction may be pulled entirely from time loop to save assembly of pressure matrix every time step.
- Advantageous for GPU and MIC computing. Compute pressure matrix once. Only need to transfer RHS vector.
- For peta-scale core counts, solve momentum equations explicitly (Runge-Kutta). Combined with above, could perform close to fully explicit codes.
- Solvers have been developed and are undergoing testing.
Summary

- SLIM significantly faster than PISO. Problem dependent but 30-100% is typical improvement and even more for very large HPC calculations.
- Exact velocity splitting improves both convergence and accuracy.
- Geometric pressure matrix coefficients advantageous for parallel efficiency, particularly for multi-grid solvers.
- Additional modifications enable scaling to very large number of cores (HPC, GPU, MIC).
Strategic Perspective

Select research and development projects that are unique and help transfer knowledge to industrial applications.

- Solvers: transient, compressible, multi-phase, combustion, acoustics
- Turbulence: RANS, DES and LES, VLES, wall models
- Sensitivity, design optimisation, and uncertainty propagation: adjoint, tangent
- Numerical acceleration and stabilisation
- Platforms and architectures: HPC, GPU, MIC