ADAPTIVE MESH REFINEMENT IN AERODYNAMICS

GOFUN 2018

Braunschweig

Thomas Schumacher

21.02.2018
Content

› Introduction to Engys & HELYX
› Adaptive Mesh Refinement (AMR)
  – Motivation & Concept
  – Application to Aerodynamics
› Update on GIB
Introduction Engys

- Founded 2009
- ~20 Developers & Engineers
  - OpenFOAM experience since 1999
- Worldwide Presence

- CFD Consultancy
- Code Development
- CFD Product Provider
  - HELYX®
  - ELEMENTS
Introduction HELYX

› General Purpose CFD Product
  – Open Source Core
  – Modern GUI
  – Comprehensive Documentation
  – Unlimited User Support
  – Code Maintenance
Adaptive Mesh Refinement (AMR)

› Work done by Daniel Deising (TU Darmstadt; Engys)
› Dynamically refine & unrefine mesh based on criterias
› Criterias:
  – Gradients
  – Interfaces
  – Iso-values
› Improves accuracy by putting mesh where it is needed
› Potentially saves run time by creating “unexpected” coarse mesh regions
Load Balancing

› Check for parallel processor imbalance w.r.t. cell count
› Redistribute dynamically (for example after every mesh refinement loop)
› Hierarchical decomposition
Example AMR

dynamicFvMesh dynamicRefineBalancedFvMesh;
refinementControls
{
  enableRefinementControl true;
  interface
  {
    alpha1 (2 5);
  }
  fields
  {
    alpha1 (0.01 1.1 3)
    C1 (0.001 0.05 2);
  }
  gradients
  {
    alpha1 (0.01 2 2);
  }
  curls
  {
    U (100 1e+05 3);
  }
  regions
  {
    cylinderToCell
    {
      p1 (0.015 0.015 0.015);
      p2 (0.015 0.033 0.015);
      radius 0.006;
    }
  }
}
Example Load Balancing
Automotive Aerodynamics

- Transient DDES simulation ~1-4 seconds
- Detailed car geometries
- Typical mesh sizes up to 200M cells
- Typical 100-1000 parallel processors
- Turnaround times ~12-36 hours
Example Mesh
Test Setup

› ERA electric car in Nürburgring record configuration
› Steady state RANS
› 8mm surface cells (coarser than production cases, which are typically ~1-2mm) for faster turn around and testing
› Symmetrical half model
› 48 processors
Refinement Criteria

› Start refinement at iteration 300
› Refine every 20 iterations
› Velocity gradient
  – 10, 50, 100 [1/s]
› Pressure gradient
  – 30, 150, 600 [m/s²]
› Values are chose “arbitrary” / engineering guess
Refinement Level 4 Gradients

gradU

gradp
Refinement Level 5 Gradients

gradU

gradp
Refinement Level 6 Gradients

\[ \text{gradU} \]

\[ \text{gradp} \]
Initial Mesh (Surface Refinement Only)

1.72M cells
Final AMR mesh

2.63M cells
Conventional Mesh

5.65M cells
Results

› Difference between final AMR mesh & conventional mesh

<table>
<thead>
<tr>
<th></th>
<th>Conventional Mesh</th>
<th>AMR</th>
<th>$\Delta c_{D_{convM}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drag</td>
<td>0.382</td>
<td>0.395</td>
<td>3.4%</td>
</tr>
<tr>
<td>Front lift</td>
<td>-0.070</td>
<td>-0.092</td>
<td>5.8%</td>
</tr>
<tr>
<td>Rear lift</td>
<td>-0.375</td>
<td>-0.335</td>
<td>10.5%</td>
</tr>
</tbody>
</table>

› Unclear yet, which mesh is more accurate

› Test with production cars where windtunnel data is available
Load Balancing

› Test Setup
› Run same AMR run with and without load balancing → compare run times
› Rebalance every 2 refinement loops
› ~50% speed up in this example

Iteration 300: Start of AMR
Next Steps

› Investigate which meshing strategy is more accurate
› Come up with DES AMR strategy (refinement criterias)
Thank You

› Contact us for more information
  info@engys.com

› Try HELYX-OS – freely available GUI for OpenFOAM
  http://engys.github.io/HELYX-OS/