Improvements to the Pegasus5 Overset CFD Software

Stuart E. Rogers

Applied Modeling and Simulation Branch/Code TNA
NASA Advanced Supercomputing Division
NASA Ames Research Center, Moffett Field, CA

12th Symposium on Overset Composite Grids and Solution Technology
October 8th, 2014
Outline

- Introduction: motivation and background
- New Features in Pegasus version 5.2
- Automatic decomposition into multiple hole-cutters
- Improved parallel performance
- Assessment of computational cost
- Conclusion
Introduction
Motivation for Improvements to Pegasus5

- Complex geometries and larger grids drive need for improved automation and efficiency
  - Reduce user input
  - Improved hole-cutting
  - Reduce orphans
  - Parallel execution improvements

- Requests from users for additional features

- Assess potential for use in unsteady moving-body problems
Background: Pegasus 5.0 and 5.1 Development

Version 5 History

Version 5.0: 5th-generation overset software
- Developed 1998 - 2000
- Initially funded by NASA Advanced Subsonics Technology (AST) Program
- Primary Authors: Norman Suhs and William Deitz, Microcraft
- Completely new version of the software written in Fortran90
- Significant improvements in oversetting process
- Massive reduction in required user input
- Working version delivered to NASA Ames

Version 5.1 developed and supported by NASA:
- NASA Space Shuttle Program
- Constellation/MPCV Programs
Enabled AST Program level-1
milestone: High-Lift Aircraft
CFD in 50 days (2000)
Enabled AST Program level-1 milestone: High-Lift Aircraft CFD in 50 days (2000)

Space Shuttle Program Return-To-Flight (2003-2006)
- Enabled AST Program level-1 milestone: High-Lift Aircraft CFD in 50 days (2000)
- Space Shuttle Program Return-To-Flight (2003-2006)
- Boeing high-lift and cruise CFD analysis
Background: Pegasus5 Usage

Version 5 History: 1998 to present

- Enabled AST Program level-1 milestone: High-Lift Aircraft CFD in 50 days (2000)
- Space Shuttle Program Return-To-Flight (2003-2006)
- Boeing high-lift and cruise CFD analysis
- Orion Launch Abort Vehicle (2010-2013)
Enabled AST Program level-1 milestone: High-Lift Aircraft CFD in 50 days (2000)
Space Shuttle Program Return-To-Flight (2003-2006)
Boeing high-lift and cruise CFD analysis
Orion Launch Abort Vehicle (2010-2013)
Distributed to over 400 outside organizations and users
Background: Pegasus5 Features and Capabilities

- **Automatic hole-cutting**
  - Multi-step hybrid method using indirect and direct hole cutting
  - Cartesian hole maps provide indirect representation of hole shape
  - Line-of-sight test using surface-grid elements: direct refined hole cutting

- Hole optimization through use of “level 2” interpolation

- Internal projections between overlapping surface grids

- Finds best interpolation stencil through exhaustive search

- Parallel execution using MPI

- Automatic restart capability

- Maintains manual hole-cutting capability from Pegasus4
New Features in Pegasus 5.2

- Released April 2014, NASA 1750.2A compliant
New Features in Pegasus 5.2

- Released April 2014, NASA 1750.2A compliant
- Cell-centered grid capability
New Features in Pegasus 5.2

- Released April 2014, NASA 1750.2A compliant
- Cell-centered grid capability
- Improved parallel performance
New Features in Pegasus 5.2

- Released April 2014, NASA 1750.2A compliant
- Cell-centered grid capability
- Improved parallel performance
- New domain decomposition option for hole cutting
New Features in Pegasus 5.2

- Released April 2014, NASA 1750.2A compliant
- Cell-centered grid capability
- Improved parallel performance
- New domain decomposition option for hole cutting
- Triple-fringe layer option
New Features in Pegasus 5.2

- Released April 2014, NASA 1750.2A compliant
- Cell-centered grid capability
- Improved parallel performance
- New domain decomposition option for hole cutting
- Triple-fringe layer option
- Support for Overflow data-surface zones
New Features in Pegasus 5.2

- Released April 2014, NASA 1750.2A compliant
- Cell-centered grid capability
- Improved parallel performance
- New domain decomposition option for hole cutting
- Triple-fringe layer option
- Support for Overflow data-surface zones
- Manual hole-cut efficiency improvements
New Feature: Automatic HCUT Creation
Automatic Decomposition To Fit The Geometry

- Enhance auto hole cutting using domain decomposition

One Hole-Cutter
64 Hole-Cutters
Auto HCUT Creation: Approach

- Recursively split the domain
Auto HCUT Creation: Approach

- Recursively split the domain
  - Split the box in the longest dimension
Auto HCUT Creation: Approach

- Recursively split the domain
  - Split the box in the longest dimension
  - Split the box with the most surface-grid points
Auto HCUF Creation: Approach

- Recursively split the domain
  - Split the box in the longest dimension
  - Split the box with the most surface-grid points
Auto HCUT Creation: Approach

- Recursively split the domain
  - Split the box in the longest dimension
  - Split the box with the most surface-grid points

8 Hole Cutters
Auto HCUT Creation: Approach

- Recursively split the domain
  - Split the box in the longest dimension
  - Split the box with the most surface-grid points

16 Hole Cutters
Auto HCUT Creation: Approach

- Recursively split the domain
  - Split the box in the longest dimension
  - Split the box with the most surface-grid points
  - Never create a box completely inside

32 Hole Cutters
Auto HCUT Creation: Approach

- Recursively split the domain
  - Split the box in the longest dimension
  - Split the box with the most surface-grid points
  - Never create a box completely inside
Automatic Decomposition Features

- Auto detection of which solid walls are contained in each hole-cutter
- Auto detection of which meshes can be cut by each hole-cutter
- Improved parallel efficiency
- Improved hole-cutting resolution
- Each hole-cutter can use fewer Cartesian elements
Wing-Body Test Case: Cartesian Fringe Elements
Ratio of Total Cartesian Volume = 10.1

One Hole-Cutter: 512x512x512

64 Hole-Cutters: 128x128x128
Wing-Body Test Case: Cartesian Fringe Elements

Ratio of Total Cartesian Volume = 10.1

One Hole-Cutter: 512x512x512

64 Hole-Cutters: 128x128x128
Improved Parallel Performance
Example: 55 zones, 79 million points

- **Original approach:** coarse-grained parallelization
  - Force synchronization between major process groups
  - Complete all projection processes before starting interpolation
  - Cannot scale to large numbers of CPUs

![Diagram showing parallel performance over CPUs and time](image)
Improved Parallel Performance

Example: 55 zones, 79 million points

- **Original approach:** coarse-grained parallelization
  - Force synchronization between major process groups
  - Complete all projection processes before starting interpolation
  - Cannot scale to large numbers of CPUs

- **New approach:** finer-grained parallelization
  - Build process dependency map for each individual process
  - Improves ability to scale to more processors
Scaling of New Parallel Approach

MPCV Launch Abort Vehicle: 55 zones, 79 million points

12 MPI Processes

Old Algorithm

New Algorithm
Scaling of New Parallel Approach

MPCV Launch Abort Vehicle: 55 zones, 79 million points

24 MPI Processes

Old Algorithm

New Algorithm
Scaling of New Parallel Approach
MPCV Launch Abort Vehicle: 55 zones, 79 million points

48 MPI Processes

Old Algorithm

New Algorithm
Scaling of New Parallel Approach

MPCV Launch Abort Vehicle: 55 zones, 79 million points

72 MPI Processes

Old Algorithm

New Algorithm
Scaling of New Parallel Approach

MPCV Launch Abort Vehicle: 55 zones, 79 million points

96 MPI Processes

Old Algorithm

New Algorithm

---

Rogers

13 / 17
Scaling of New Parallel Approach

MPCV Launch Abort Vehicle: 55 zones, 79 million points

Asymptotic performance: 0.5 \( \mu \text{sec per grid-pt} \)

Old Algorithm

New Algorithm
Performance of New Parallel Approach
Space Launch System: 892 zones, 375 million points

- 100 CPUs
- Significant start-up time: building process dependency link-lists
- Significant final output time: serial output
Performance of New Parallel Approach
Space Launch System: 892 zones, 375 million points

- Wallclock-time to create overset, sec:
- 20 CPUs: 1100
Performance of New Parallel Approach
Space Launch System: 892 zones, 375 million points

- Wallclock-time to create overset, sec:
  - 20 CPUs: 1100
  - 40 CPUs: 550
Performance of New Parallel Approach
Space Launch System: 892 zones, 375 million points

- Wallclock-time to create overset, sec:
  - 20 CPUs: 1100
  - 40 CPUs: 550
  - 80 CPUs: 280
Performance of New Parallel Approach
Space Launch System: 892 zones, 375 million points

- Wallclock-time to create overset, sec:
  - 20 CPUs: 1100
  - 40 CPUs: 550
  - 80 CPUs: 280
  - 160 CPUs: 240
Performance of New Parallel Approach
Space Launch System: 892 zones, 375 million points

- Wallclock-time to create overset, sec:
  - 20 CPUs: 1100
  - 40 CPUs: 550
  - 80 CPUs: 280
  - 160 CPUs: 240
  - 200 CPUs: 240

Asymptotic performance: 0.6 $\mu$sec per grid-pt
Relative cost of Overflow and Pegasus 5.2
Intel Ivy-Bridge Nodes

- Approximate wall-clock time per time step, in seconds
- Dual time-stepping time-advance algorithm
- Overflow: $\approx 5$ micro-seconds per sub-iter per grid pt

<table>
<thead>
<tr>
<th>NITNWT</th>
<th>Points (millions)</th>
<th>100 CPUs</th>
<th>200 CPUs</th>
<th>400 CPUs</th>
<th>800 CPUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overflow</td>
<td>10</td>
<td>100</td>
<td>50</td>
<td>25</td>
<td>–</td>
</tr>
<tr>
<td>Overflow</td>
<td>50</td>
<td>100</td>
<td>250</td>
<td>125</td>
<td>–</td>
</tr>
<tr>
<td>Pegasus5</td>
<td>100</td>
<td>60</td>
<td>60</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Overflow</td>
<td>10</td>
<td>200</td>
<td>100</td>
<td>50</td>
<td>25</td>
</tr>
<tr>
<td>Overflow</td>
<td>50</td>
<td>200</td>
<td>500</td>
<td>250</td>
<td>125</td>
</tr>
<tr>
<td>Pegasus5</td>
<td>200</td>
<td>120</td>
<td>120</td>
<td>120</td>
<td>–</td>
</tr>
<tr>
<td>Overflow</td>
<td>10</td>
<td>400</td>
<td>–</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>Overflow</td>
<td>50</td>
<td>400</td>
<td>–</td>
<td>500</td>
<td>250</td>
</tr>
<tr>
<td>Pegasus5</td>
<td>400</td>
<td>–</td>
<td>240</td>
<td>240</td>
<td>240</td>
</tr>
</tbody>
</table>
Conclusion

- Released version 5.2 of Pegasus
- Many improvements and some new features
- Automatic domain decomposition into automatic hole cutters
- Improved parallel efficiency through fine-grained parallelization
  - \( \approx 0.6 \, \mu \text{seconds per grid point} \)
  - Further process optimization required for additional scaling improvements
- Potential applications to time-dependent moving body problems
  - Pegasus 5.2 can re-process the entire grid in 0.1 to 2.0 times one time-step in Overflow