PEGASUS 5: An Automated Pre-Processor for Overset-Grid CFD

Stuart Rogers
NASA Ames Research Center
http://people.nasa.gov/~rogers/home.html
rogers@nas.nasa.gov

Pegasus Authors:
Norman Suhs, William Dietz, Stuart Rogers
Steve Nash, William Chan, Robert Tramel, Jeff Onufer

Overset Short Course
8th Overset Symposium
October 2nd, 2006

Outline

- Pegasus5 features and automation
  - Auto hole cutting
  - Interpolation and overlap optimization
  - Projection
  - Restarting
  - Parallelization
- Overview of Usage
  - Required inputs
  - Basic Usage
  - Overcoming problems
PEGASUS 5

- Fifth-generation overset software
- Primary goal: complete automation of overset process
  - Complexity of CFD problems grown
  - Hundreds of overset zones
  - Grid points in tens of millions
  - Manual control of process became intractable
- Required all-new approach to
  - Hole-cutting
  - Overset optimization
- Required significant improvements in ease of use
  - Parallelization
  - Automatic restarts
  - Projection
- Maintained backward compatibility allowing manual control where needed
- Pegasus5 is mostly automated, but still requires user expertise

Auto Hole-Cutting Using Cartesian Map

Air-tight solid-wall surfaces

Find fringe elements

Find outside elements: painting algorithm

Inside elements
Auto Hole Cutting
Cutting of Candidate Points

- All volume grid points considered as candidate hole points
- Points in an outside element are not hole points
- Points in an inside element are hole points
- Points in a fringe element use a line-of-sight test

Interpolation Boundary Identification

- Outer boundary fringe points
  - All points on the boundary of a zone that do not receive a flow-solver boundary condition is assumed to be an interpolated outer boundary
  - Single or double fringes can be specified
- Hole boundary fringe points
  - Points adjacent to a hole point are interpolated hole boundary points
  - Single or double fringes can be specified
Boundary Point Interpolation

- Pegasus searches for all possible interpolation stencil donors from all zones for **every single grid point**
  - Uses alternating-digital tree to search for a near-by cell, then a stencil-jumping approach to find exact donor cell and stencil
- Best interpolation stencil is selected for each boundary interpolation point
  - Uses a measure of the interpolation quality and the relative cell size to interpolate from a similar-sized cell

Overlap Optimization

- Begins with outer boundary and hole boundary interpolations
- Has effect of expanding the automatically-cut holes and shrinking the outer edges of overlapping zones
- Finest Mesh Points Retained
- Coarser Mesh Points are Interpolated
- Methodology is robust, requires no user inputs, and maximizes communication between overlapping zones
Non-Optimized Overlap

Optimized Overlap

• Corrects interpolation problems that may occur on curved viscous surfaces
  – Caused by linear discretization of curved surfaces
• Pegasus 5 projection step alters interpolation coefficients, not grids
Restarting

- Pegasus5 process consists of many individual sub-processes.
- Each sub-process has a defined set of dependencies (inputs).
- Each sub-process results in one or more output files saved to disk.
- Automatically determines which sub-processes are out of date based on time-stamps of inputs and outputs.
- Upon execution, Pegasus5 automatically determines which sub-processes need to be run.
- Can successfully restart for:
  - Modifications in user inputs or meshes.
  - Addition of new meshes.
  - Incomplete previous run or computer crash.
- Allows incremental buildup of your CFD problem.

Parallelization

- Most sub-processes are independent of each other and can be run in parallel.
- Uses Message-Passing-Interface (MPI):
  - One master process to distribute and monitor the work.
  - Many worker processes, one per CPU.
  - Shared or distributed memory.
- Reliably reproduces results of serial execution.
- The larger the problem, the better the scaling.
- Requires that all CPUs have access to the same working directory.
Boeing 777 Parallel Speedup
SGI O3K 400Mhz: Total CPU Time = 283 min

PEGASUS 5 Parallelization
15 Processors on an SGI O2K
Harrier grid system: 52 zones, 2.5 million grid points
PEGASUS 5 Parallelization

Parallel execution: barrier between ADT and INTERPOLATION

---

**Input requirements**
- Standard input file
- Volume grids in individual files
  - `X_DIR/meshname1.x, X_DIR/meshname2.x, etc`
- Methods to assist in generating these inputs
  - `peg_setup` script
    - Requires Overflow input file and multi-zone plot3d grid file containing all the volume grids
  - Chimera Grid Tools scripts: `BuildPeg5i`
PEGASUS 5
Input Requirements (cont.)

$GLOBAL
FRINGE = 2,
OFFSET = 1,
$END

$SMESH NAME = 'body', JINCLUDE= 2, -2, LINCLUDE= 2, -1
OFFSET=2, $END

$SMESH NAME = 'bodynose', JINCLUDE= 2, -1, LINCLUDE= 2, -1, $END

$SMESH NAME = 'wing', $END

$SMESH NAME = 'wingcap', $END

$SMESH NAME = 'wingcol', $END

$BCINP ISPARTOF = 'body',
IETYP = 5, 17, 17, 15,
IBDIR = 3, 2, -2, -1,
JBCS = 1, 1, 1, -1,
JRCE = -1, -1, -1, -1,
KBCS = 1, 1, -1, 1,
KCCE = -1, 1, -1, -1,
LBCS = 1, 1, 1, 1,
LBCE = 1, -1, -1, -1,
YSYM = 1,
$END
PEGASUS 5
Execution

- Once the `peg.i` file is available, and the volume grids (*.x) are in the `X_DIR` directory, PEGASUS 5 can be launched:
  - Serial version on a single CPU:
    
    ```
    pegasus5 < peg.i > &! peg.out
    ```

  - MPI Parallel on `$NCPUs`:
    ```
    mpirun -NP $NCPUs pegasus5mpi < peg.i > peg.out
    ```

Execution (cont.)

- As the code is running, PEG5 creates:
  - A directory named `/WORK` which contains all the time-stamped work files needed by different processes in the code. Delete this directory to start from a job from "scratch".
  - A log file named `log.mmdd.hhmm` which contains all the standard output from the run. Check this file for input echo, orphan count, and run times, etc.

  *NOTE:* When running with MPI, `log.mmdd.hhmm.{0000,0001,...,NCPU-1}` files are created, and concatenated into one file on completion.

- Parallel version requires that all CPUs have access to working directory and files in `/WORK`
XINTOUT contains interpolation stencils, etc. used by flow solver.

Use peg_plot to create grid.in file for flow solver:
- Option 3 shows minimum holes (all fringe levels).
- Option 1 (single-fringe) and option 2 (double-fringe) blank out higher level fringes, and show location where flow information will be passed within grid system.

Higher level fringes are interpolated in the flow solver, but not used because code is only 2nd-order accurate.

However, they are still important because they can be used as donors by other 1st or 2nd fringe points from other grids.
Option 3 shows exactly what the flow solver "sees":
- Use to see all fringes

Option 2 shows exact region where flow information is exchanged between grids
- Use to see "virtual" holes

Orphans are hole or outer boundary points that do not find valid interpolation stencils during the PEGASUS process.

2nd-level Orphans reset to field points in pegasus5

Output from peg_plot:

<table>
<thead>
<tr>
<th>Mesh No.</th>
<th>Name</th>
<th>ORPHANS: 1st-level</th>
<th>2nd-level</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>body</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>boynose</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>wing</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>wingtip</td>
<td>0</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>wingcap</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>boxbody</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>boxwing</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>boxouter</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Total

0 21 0
Typically, PEG5 will be iterated several times on a new configuration to eliminate all the orphans that would cause the flow solver to fail, or give erroneous results.

- Usually, a few hundred orphans remain in a grid system of >20M points.
- OVERFLOW deals with orphans by "averaging" the surrounding nodes, so orphans in areas of small gradients, etc., are okay.

Orphans in OVERFLOW
- 2nd-level fringe orphans are reset to field points
- 1st-level fringe orphans are "averaged" by using valid surrounding data to compute solution vector.

Guidelines:
- Always "fix" large groups of orphans anywhere in the grid domain. The bigger the clump, the greater the chance of bad flow interpolation/extrapolation in OVERFLOW.
- Orphans on surfaces usually indicate serious problems with surface resolution or projection, and must be fixed.

Plot Orphans using Plot3d (function 3), or Overgrid
PEGASUS 5
Orphans (cont.)

• Usual causes:
  – Insufficient overlap
  – Poorly resolved geometry (overlapping surfaces that are not projected properly)
  – Inappropriate or missing BCs
  – Failure in holecutting algorithm

To fix:
1. Increase surface overlap, or
2. Use "splaying" in HYPGEN or LEGRID, or
3. Add field box grid to resolve open space
Custom Hole Cutting

- $\texttt{HCU}$T namelists are used to define separate hole-cutters. The default is to include ALL solid wall surfaces in one hole-cutter.
- Adding an $\texttt{HCU}$T entry eliminates the default hole-cutter, and you must provide all of the hole-cutting inputs.
- Adding multiple $\texttt{HCU}$T entries increases parallel efficiency.

```plaintext
$\texttt{HCU}$T NAME = 'hcutter1',
MEMBER = 'body1', 'body2',
INCLUDE = 'bodynose', 'wing', 'wingcol',
CNX = 512, CNY = 512, CNZ = 512,
CARTX = -100.0, 100.0,
CARTY = -50.0, 50.0,
CARTZ = 0.0, 100.0,
$\texttt{END}$
```

Continuous $\texttt{HCU}$T

Custom Hole Cutting

- Multiple $\texttt{HCU}$T definitions can be used to better resolve and focus the Cartesian hole-cutter:

Single $\texttt{HCU}$T

Three $\texttt{HCU}$T namelists:
- Created by specifying CARTX, CARTY, CARTZ
- Focuses Cartesian cells in volume of interest
Hole-Cutting Issues

- Holes too small near thin bodies (i.e. trailing edge of a thin wing)
  - Use OFFSET to enlarge holes
  - Increase CNX, CNY, CNZ to increase resolution
- Hole points not cut out properly near collar grids
  - Use OFFSET to enlarge holes
- Holes cut at zone boundaries on surface when no holes should be made
  - Occurs on curved surfaces with poor resolution
  - Can be undone by "unblanking" with $REGION and $VOLUME names lists

PEGASUS 5
Some Useful Utilities

- peg_setup is a menu-driven script which initializes the PEG5 problem, given a single, multi-zone PLOT3D file that contains all the volumes grids, and an OVERFLOW input file.
- peg_plot creates the iblanked PLOT3D grid file used in OVERFLOW.
- peg_hole_surf creates a multi-zone PLOT3D file of all the solid wall subsets in each defined holecutter. Used to visually examine the hole-cutting surfaces.
- peg_diag creates a diagnostic file for plotting quality and cell difference values.
- peg_orph is a script to list orphans by zone, etc.
Summary

- PEGASUS 5 successfully automates most of the overset process
  - Dramatic reduction in user input over previous generations of overset software
  - Order of magnitude reduction in both turn-around time and in user-expertise requirements
- Not a "black-box" procedure: care must be taken to examine the resulting grid system
- Additional documentation and examples available online:
  http://people.nas.nasa.gov/~rogers/pegasus/status.html