Helios Solver Developments Including Strand Meshes

Collaborators:

Jayanarayanan Sitaraman, U. Wyoming
Aaron Katz, Utah State
Robert Haimes, MIT

Presented by:

Andrew Wissink
U.S. Army Aeroflightdynamics Directorate
Aviation and Missile Research, Development and Engineering Center
Moffett Field, CA

Approved for public release; distribution unlimited.
Review completed by the AMRDEC Public Affairs Office (FN6065, 16 Oct 2012)
Presentation Outline

- Overview
- New capabilities
- Sample Results
- Strand Solver Developments
- Summary

Version 3 Rainier
Helios Software

Helios Helicopter Overset Simulations

Dual Mesh Paradigm
- Unstructured Near-body
- Cartesian off-body

Adaptive Mesh Refinement
- To resolve wake

Moving Body Overset
- Rotor-Fuselage and Multi-rotor moving mesh support

CFD/CSD Coupling
- RCAS and CAMRAD Structural Dynamics and Trim coupling

Advanced Software Infrastructure
- Python-based infrastructure readily supports addition of new software

High Performance Computing
- Runs on HPC hardware with focus on parallel scalability

Supports high-fidelity rotary-wing simulation by government and industry
Developed & maintained by a team at Army AFDD.
Dual Mesh CFD Paradigm

Unstructured “near-body”
- Resolve near-wall viscous flow
- Complex geometries

Cartesian “off-body”
- Computationally efficient
- High order accuracy
- Adaptive Mesh Refinement

Effective for time-dependent/moving-body applications
CFD Components

• Near-body NSU3D flow solver
  – Developed by Mavriplis at Univ. of Wyoming
  – General unstructured – tets, prism, hex
  – Reynolds-averaged Navier-Stokes
  – Spalart-Allmaras turbulence model
  – 2\textsuperscript{nd}-Order vertex-based spatial discretization
  – 2\textsuperscript{nd}-Order BDF time integration

• Off-body SAMARC flow solver
  – Couples LLNL SAMRAI with NASA Ames ARC3D
  – Block structured Cartesian
  – 5\textsuperscript{th}-Order spatial discretization
  – 3\textsuperscript{rd}-Order explicit Runge-Kutta time
  – Automated AMR

• Overset Communication
  – PUNDIT
  – Automated implicit hole cutting
Helios Released Capabilities

- **Version 1 Whitney**
  - Dual-mesh unstructured/Cartesian
  - Fixed off-body
  - Isolated rotor/fuselage

- **Version 2 Shasta**
  - Adaptive Mesh Refinement (AMR)
  - Rotor-Fuselage

- **Version 3 Rainier**
  - Automated AMR
  - Multi-rotor
  - DES Turbulence modeling
  - Coviz

**Release Timeline**

- **v1.0** 2010
- **v1.1** 2011
- **v1.2** 2011
- **v2.0** 2012
- **v2.1** 2012
- **v3.0** 2013
- **v3.1** 2013
- **v3.2** 2013
• Why not simply refine to vorticity or Q-criterion?
  – User dials in feature and quantity to adapt to \( (\omega_{\text{adapt}} \text{ or } Q_{\text{adapt}}) \)

<table>
<thead>
<tr>
<th>case</th>
<th>( C_T/\sigma )</th>
<th>R</th>
<th>chord</th>
<th>Refine Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>UH-60 high speed</td>
<td>0.084</td>
<td>322 in</td>
<td>22 in</td>
<td>( \omega_{\text{adapt}} = 0.0045 )</td>
</tr>
<tr>
<td>V-22 hover</td>
<td>0.14</td>
<td>228 in</td>
<td>22 in</td>
<td>( \omega_{\text{adapt}} = 0.005 )</td>
</tr>
<tr>
<td>HART BVI descent</td>
<td>0.06</td>
<td>79 in</td>
<td>4.8 in</td>
<td>( Q_{\text{adapt}} = 0.0005 )</td>
</tr>
</tbody>
</table>

Choosing appropriate refine criteria requires considerable user-expertise and tuning.

Fixed: \( \omega_{\text{adapt}} \text{ low} \) (too many points)

Good: \( \omega_{\text{adapt}} \text{ high} \) (too few points)

UH60 high speed forward flight

Manually determined through trial and error…
• Define $f$ as Q-criteria normalized by shear strain

• Refine where $f > 1$, otherwise don’t refine

$$Q = \frac{1}{2}(||\Omega||^2 - ||S||^2), \quad f = Q$$

$$f = \frac{1}{2} \left( \frac{||\Omega||^2}{||S||^2} - 1 \right)$$

Automated Off-Body AMR
TRAM Rotor

<table>
<thead>
<tr>
<th></th>
<th>Figure of Merit</th>
<th>Difference</th>
<th>Mesh Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experiment</td>
<td>0.78</td>
<td>-</td>
<td>--</td>
</tr>
<tr>
<td>Computation</td>
<td>0.773</td>
<td>-0.9% (+/- 0.2%)</td>
<td>86 M</td>
</tr>
</tbody>
</table>

- Finest mesh resolution applied to all regions of swirling flow
- Richardson-error refinement cutoff also available

feature detection with $f=1$ used for off-body refinement

$M_{tip}=0.625$
$\theta=14^\circ$ coll
• Current and future rotorcraft utilize multiple rotors

• Version 3 fully supports multi-rotor vehicle configurations
  – Multiple motion files (specified control angles)
  – CSD coupling with comprehensive analysis codes

Hover

Forward flight

Hypothetical coaxial TRAM

Hypothetical coaxial configuration

Sikorsky X2

CH47 (tandem)
What Can We Do?
An Assessment of Current Capabilities

- Helios v3 applied to HART II case
  - 40% scaled Bo105 model rotor & “fuselage” tested in DNW windtunnel
  - Wake-based PIV measurements of vortex locations and strength

Helios simulation
Near-body: 3M nodes
Off-body: 200M-300M nodes
Requires ~4 days on 256 procs

0.025c spacing
Highly-resolved Off-body Grid

Compare PIV-measured vorticity to computation

| Pos 1  | 4.7° |
| Pos 9  | 14.3° |
| Pos 17 | 25.3° |
• Vorticity dissipates most quickly in near-body mesh
• Strength preserved in off-body mesh
Presentation Outline

- Overview
- New capabilities
- Sample Results
- Strand Solver Developments
- Summary
Current Bottlenecks

- **Near-body unstructured grid generation**
  - Not straightforward for the typical design engineer
  - Fixed near-body subset distance
  - Not adaptive

- **Lower order near-body solver**
  - Unstructured solver limited to 2\textsuperscript{nd}-O (off-body Cartesian is 5\textsuperscript{th}-O)

**Computational Cost**

- **Near-body: 9.4M**
  - Off-body: 34.1M

- **Gridpoints**

<table>
<thead>
<tr>
<th></th>
<th>Near-body</th>
<th>Off-body</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAM Hover</td>
<td>9.4M</td>
<td>34.1M</td>
</tr>
<tr>
<td>Tip vortex</td>
<td>9.4M</td>
<td>34.1M</td>
</tr>
<tr>
<td>Blade Tip</td>
<td>15.4M</td>
<td>146M</td>
</tr>
<tr>
<td>Total</td>
<td>66%</td>
<td>65%</td>
</tr>
</tbody>
</table>

- **TRAM Hover - 128 procs**
  - 137 hrs (8 revs/6 days)

- **UH60 Fwd Flight - 512 procs**
  - 28.5 hrs (1 rev/1.2 days)
**Strand/Cartesian Approach**

- **“Strand” near-body grid**
  - Straight line segments grown directly from surface tessellation
  - Transitions from viscous spacing at surface to Cartesian off-body
  - **Automatic** viscous mesh generation

- **Concept introduced and studied in earlier works**

- **Recent work focuses on combining these developments into a usable 3D code**

Works with any surface tessellation
Mesh Generation
Directly from CAD

**Step 1:**

**Generate Solid Geometry**
- Available in modern CAD packages e.g. STEP files
- CREATE products that produce STEP include DaVinci, Capstone

**Step 2:** *(automatic)*

**Tessellate Surface**
- Supply desired resolution: $\Delta x$
- Parameters control resolution in high curvature regions, small protrusions

Δx = 0.06  Δx = 0.015
(coarse)  (fine)
**Mesh Generation**
Directly from CAD (cont)

**Step 3:** (automatic)

**Near-body Strand Generation**
- Surface $\Delta x$ determines strand length
- Construct “lifted surface” with isotropic spacing
- Lifted surface defines multiple strands at convex edges/nodes
- Apply smoothing at concave corners

**Step 4:** (automatic)

**Off-body Cartesian Generation**
- Refine to lifted surface
- Telescope to resolve clipped strands
Domain Connectivity

- **OSCAR: Automated Implicit hole cutting**
  - Donors identified for every point of every mesh
  - All donor searches completed locally (since entire strand mesh description available to every processor)

- **Inverse maps accelerate donor search**
  - Holds indices of strand “super cell”
  - Logarithmic inverse map for on-strand index search

- **Surface-based partitioning enhances scalability**
• **Parallel Infrastructure for Cartesian And Strand Solvers**
  
  − Provides strand grid generation and adaptation with interfaces to surface meshing and geometry packages
  
  − Facilitates easy integration of new flow solvers, extensible.
Flow Solver

- **Strand solver – A. Katz**
  - Cell-centered
  - Reynolds-averaged Navier-Stokes
  - No turbulence model (yet)
  - Accommodates both prisms & hexes
  - 2nd-Order spatial discretization
  - Implicit pseudo-time marching
  - *Details in AIAA-2012-2779*

- **Cartesian Solver**
  - SAMARC solver used in Helios
  - *Details in AIAA-2010-4554*

\[
\hat{F} = \frac{1}{2} (F(Q_L) + F(Q_R)) - D(Q_L, Q_R),
\]

**CUSP:**
\[
D(Q_L, Q_R) = \frac{1}{2} \alpha^* (Q_R - Q_L) + \frac{1}{2} \beta (F(Q_R) - F(Q_L)).
\]

**Nodal projection gradients:**
\[
\frac{\partial Q}{\partial x} = \sum_i a_{xi} Q_i
\]
Inputs:
   - STEP file
   - Wall spacing
   - Flow conditions

Solution after 25000 steps
Flat tip
Aspect ratio = 6.6
M=0.1235
$\alpha = 12^\circ$
Re = 5000

PICASSO
32 procs

Experimental Results
McAlister, Takahashi

Mesh

Vorticity contours

Vorticity iso-surface
Flat tip
Aspect ratio = 6.6
M=0.1235
$\alpha=12^\circ$
Re = 5000

Helios Results
unstruct mesh, NSU3D

Trailing edge flow should be improved by multi-strand capability
Targeted for Jan 2013

PICASSO Results
strand mesh, Strand3d

June 2012
AIAA-2012-2916

Oct 2012
Summary

- **Helios** is a high-fidelity rotorcraft overset CFD/CSD analysis tool intended for DoD acquisition engineers

- **Current CFD bottlenecks**
  - Unstructured mesh generation non-trivial
  - Much of the wake dissipation occurs in near-body mesh
  - Near-body solver dominates computation time

- **Strand technology a promising new approach**
  - Automatic viscous mesh generation directly from CAD solid geometry
  - Parallel flow solver that exploits structure in normal direction
  - Scalable domain connectivity

- **First dedicated strand solver implemented in PICASSO**
  - Demonstrated automatic CAD-to-mesh for simple geometries
  - Actively pursuing more complex cases
Future

- Application of Helios v3 to candidate Joint Multi-Role acquisition designs
- **Strand improvements**
  - Capability to handle multiple strands
  - Turbulence models
  - More complex geometries
- **Higher order algorithms**
- **Near-body Mesh Adaptation**
- **Higher fidelity structural dynamics**
  - Current CSD uses 1D beam elements
  - 3D dynamics finite element solver needed for composite blades/hubs
- **Scalability to 1000’s processors**
Acknowledgements

Helios development team

Supported by the CREATE-AV (air vehicles) element of the Computational Research and Engineering for Acquisition Tools and Environments (CREATE) Program sponsored by the U.S. Department of Defense HPC Modernization Program Office.