CREATE^{TM}-AV Kestrel Dual-Mesh Computations on the NASA Common Research Model

Tim Eymann and Robert Nichols

20 October 2016

Introduction

This work investigates the differences between Kestrel's fully unstructured and coupled unstructured-Cartesian simulations of a modern, transport aircraft

Outline:

1. Kestrel Description
2. The NASA Common Research Model
3. KCFD/SAMAir CRM Results
4. Effects of Subset Distance and AMR
5. Conclusions and Future Work

CREATE^{TM}-AV Kestrel

- DoD HPCMP initiated the Computational Research and Engineering Acquisition Tools and Environments (CREATE^{TM}) in FY2008
- The goal of all CREATE products is to give working-level engineers access to high-fidelity, physics-based codes
- CREATE Tools:
  - Enable virtual prototyping and early discovery of design defects during development
  - Allow for efficient evaluation of design changes during sustainment
- The CREATE program is developing dedicated software packages for ships, ground vehicles, antennas, meshing/geometry, and air vehicles
- Air vehicles is further split into rotorcraft (CREATE^{TM}-AV Helios) and fixed-wing applications (CREATE^{TM}-AV Kestrel)

Kestrel Architecture

- Kestrel's core functions are performed by components that are dedicated to a single task and unaware of other components present in the simulation
- The components are linked through a Python-based common scalable infrastructure (CSI) and share data via pointers
- Event based
  - Each component responds to events that are published by CSI
  - Components only subscribe to events that affect them
- This architecture enables an enormous amount of flexibility and facilitates:
  - Maintainable and extensible code, including from outside partners through a software development kit
  - The ability to efficiently run a variety of use-cases by excluding components that are not necessary

Dual-mesh Components

- KCFD
  - Kestrel's unstructured CFD solver
  - Supports mixed hex, prism, pyramid, and tet elements
- SAMAir
  - Kestrel v5 (2014) introduced an off-body Cartesian solver, SAMCart, based on technology pioneered by the Helios team
  - For Kestrel v7 (2016), the code was extensively re-worked and released as SAMAir
  - The name reflects the use of the Structured Adaptive Mesh Refinement Application Infrastructure (SAMRAI) from Lawrence Livermore National Laboratory and the code's NXAIR heritage
- PUNDIT
  - Parallel UNsteady Domain Information Transfer
  - Performs domain connectivity operations, including interpolation
  - Input-free, implicit hole-cutting

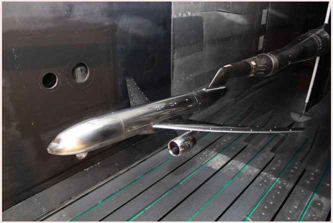
NASA Common Research Model

- Non-proprietary/non-sensitive geometry, representative of a modern transport aircraft
- The Wing/Body (WB) configuration has been studied in AIAA Drag Prediction Workshops (DPW) since DPW IV (2009)
- DPW VI (2016) included studies of configurations with engine nacelles; Wing/Body/Nacelle/Pylon (WBNP)
- Focus on comparison to other CFD codes rather than comparison to the wind tunnel results

CRM WB

NASA CRM website

CRM WBNP



DPW VI website

Problem Set-up

- Calculate the total drag coefficient at a fixed lift of $C_L = 0.5 \pm 0.0001$
- Each solution point started from initial, uniform flow solution
- Splart-Almaras turbulence model
- All other inputs left at default values
  - HLLE++ fluxes
  - Spatial accuracy 2^{nd}-order for KCFD, 3^{rd}-order for SAMAir
  - Global (time-accurate) time stepping at 2^{nd}-order temporal accuracy
- Solutions computed on the DPW VI meshes provided by NASA's GeoLab
  - "Merged" versions with prism boundary layers
  - WB cell count: 83,598,506 to 182,037,523
  - WBNP cell count: 120,990,279 to 266,916,327
- Focus on accuracy, not performance

Dual-mesh Workflow

1. Subset an existing, unstructured mesh
   - Kestrel provides this functionality with two tools: the Kestrel user interface (KUI) and Carpenter
   - User specifies a subset distance, but everything else is **automatic**, including tagging an "overset" boundary
2. Use KUI to define a Cartesian mesh
   - User supplies boundary extents with either min/max corners or as a number of reference lengths away from the body
   - Kestrel **automatically** calculates the number of refinement levels required to generate Cartesian cells that are similar in size to the unstructured cells at the overset boundary
3. Execute the Kestrel simulation
   - Domain connection is **automatic**; PUNDIT uses information from the unstructured and Cartesian meshes to assign hole, fringe, and orphan points
   - SAMAir logs the number of orphans and averages the solution when the total number is below a user-defined threshold
   - Single input adds geom. refinement to resolve excess orphans

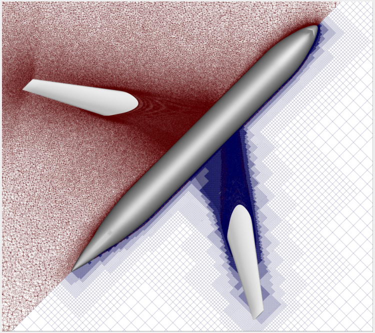
Mesh Adaptation in SAMAir

- Cells are tagged for refinement based on geometric or solution-based parameters
- For geometric refinement:
  - PUNDIT assigns a representative length scale to the unstructured cells close to the near-body overset surface
  - If any Cartesian cell width is significantly larger than this length scale, it is marked for refinement
  - This process continues until the Cartesian mesh size is comparable to the unstructured cells, or the maximum number of refinement levels is reached
- Solution refinement:
  - Tags cells based on local measures of density, entropy, vorticity magnitude, scaled Q-criterion, or shock sensor
- SAMRAI's routines then assemble regular blocks around the tagged regions

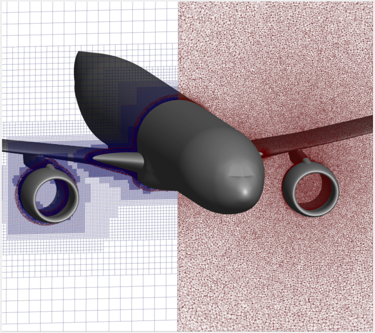
CRM Dual-mesh System

- SAMAir meshes subset at 5% MAC ($\approx$ 14")
  - WB DOF count: 56,530,970 to 158,613,041
  - WBNP DOF count: 67,286,091 to 130,395,586
  - 20%-50% reduction in DOF compared to single meshes
- No solution refinement; only geometric

WB "Coarse" Mesh



WBNP "Coarse" Mesh



Drag Coefficient Comparisons

Finite-Volume Codes:

- KCFD; 2^{nd}-order, cell-centered, unstructured
- KCFD/SAMAir; KCFD + 3^{rd}-order, node-centered, Cartesian
- OVERFLOW; 3^{rd}-order, node-centered, structured
  - *James Coder, University of Tennessee - Knoxville*
- FUN3D; 2^{nd}-order, node-centered, unstructured
  - *Eric Nielsen and William Jones, FUN3D Development Team*

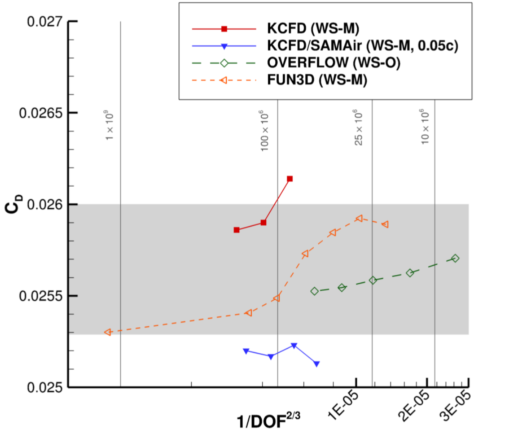
Meshes:

- WS-M; DPW VI mixed-element meshes (NASA GeoLab)
- WS-O; DPW VI overset mesh system (Boeing Long Beach)

WB Drag Coefficient

- DPW VI "Medium": $C_D = 0.02570 \pm 0.0026$
- Shaded region shows the range of solutions, excluding outliers
- SAMAir with 160×10^6 DOF is within one count of the 1×10^9 DOF FUN3D solution
- SAMAir C_D not sensitive to near-body refinement
- Default settings produce good solutions

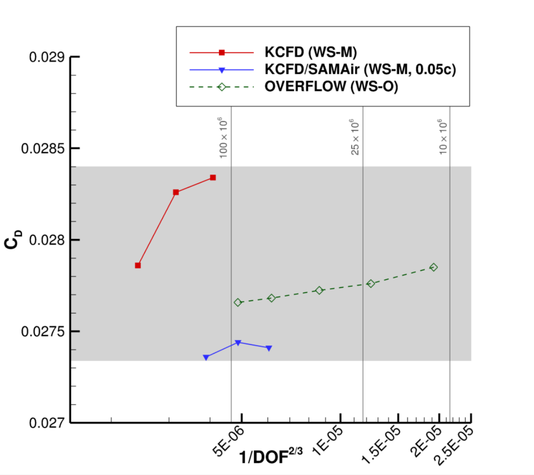
WB Drag Comparison



WBNP Drag Coefficient

- DPW VI "Medium": $C_D = 0.02803 \pm 0.0043$
- Larger spread in workshop results compared to the WB
- SAMAir within 2 counts of finest OVERFLOW solution
- SAMAir C_D not sensitive to near-body refinement
- Default settings produce good solutions

WBNP Drag Comparison



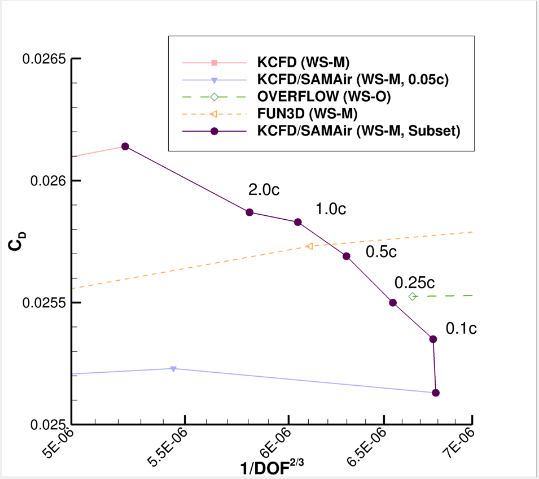
Drag Coefficient Summary

- SAMAir solutions compare well to the DPW VI workshop results
- Drag is slightly low compared to other workshop participants
- Default settings give good drag coefficients with “tiny” meshes
- Subsetting and wrapping the unstructured mesh with SAMAir has a large impact on drag, in the correct direction
- How does moving away from the baseline affect the solution?
    - Increase the subset distance
    - Activate solution refinement

Drag with Increasing Subset Distance

- Smooth transition between SAMAir and single-mesh KCFD solutions
- More investigation needed to identify "correct" distance
- Plots of pressure and skin-friction components of drag may be instructive

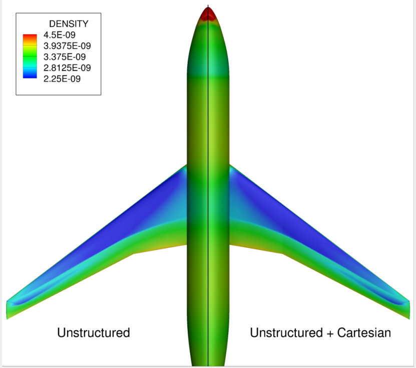
WB C_D with Subset



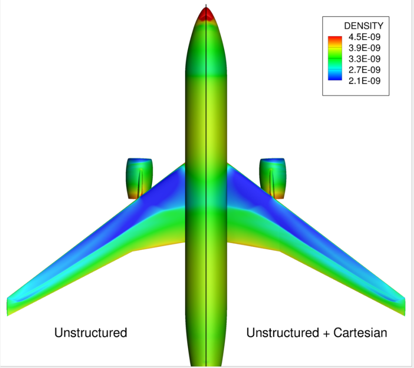
Single vs. Dual-mesh Density

- Subset plot shows the Cart. mesh makes a difference; where?
- Difficult to visually distinguish the two surface solutions

WB "Coarse"

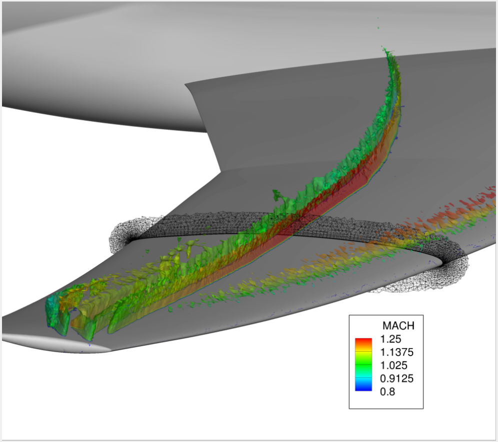


WBNP "Coarse"



Single vs. Dual-mesh Pressure Coeff.

Isosurface of $\Delta C_p = 0.02$

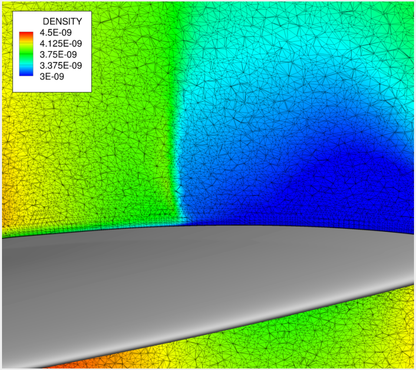


- KCFD and SAMAir computations use the same near-body mesh
- Differencing C_p between the two can give clues about where the solutions differ
- Largest difference is near the shock
- Use the shock sensor as the automatic mesh refinement (AMR) variable

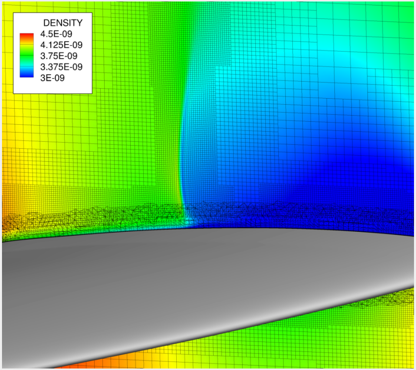
Shock Sensor AMR

- Sensor value: $\phi = \frac{\ell(\mathbf{u}\cdot\nabla p)}{ap}$

WB "Tiny" Baseline



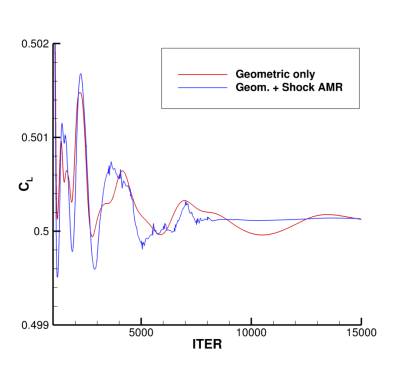
WB "Tiny" Adapted



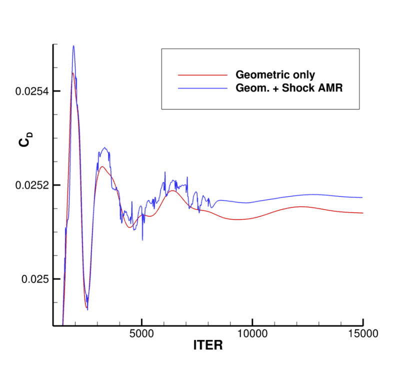
AMR Loads Convergence

- AMR every 500 iterations from iteration 500 to 8000
- Accelerates lift convergence
- Little effect on drag: $\Delta C_D \approx 5 \times 10^{-5}$

SAMAir Lift



SAMAir Drag



Kestrel

Closing Thoughts

Summary:

- KCFD/SAMAir dual-mesh simulations provide users good drag results with minimal effort
- Proper subset distance is an open question
- Shock-based AMR improves wall-clock time to an answer

Future work:

- Extract pressure coefficient profiles along the wing span
- Coarsen the workshop family of meshes
- Break out pressure and skin-friction contributions to total drag
- Investigate fifth-order SAMAir solutions
- Compare to high-order Streamwise Upwind/Petrov-Galerkin solutions from Kestrel's Conservative Field Finite-Element (COFFE) solver

Acknowledgements

Material presented in this brief is a product of the CREATE™-AV element of the Computational Research and Engineering for Acquisition Tools and Environments (CREATE) Program sponsored by the U.S. Department of Defense High Performance Computing Modernization Program Office.

Thank you Jim Coder (UT) and Eric Nielsen (NASA Langley) for providing comparison data!