High-Performance Computing Applications and Future Requirements for Army Rotorcraft

Dr. Roger Strawn
US Army Aviation Development Directorate (AMRDEC)
Ames Research Center, Moffett Field, CA

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited
DoD Future Vertical Lift Program (FVL)

- Goal is to develop future replacements for the entire US vertical lift fleet over the next 25-40 years
- FVL will provide the speed, range, payload and mission systems critical for success in future operational environments
- FVL aircraft will share common hardware such as sensors, avionics, engines, and countermeasures
- Army’s precursor science and technology program is called “Joint Multi-Role Technology Demonstrator” (JMR) that will provide flight demonstration in 2017
Joint Multi-Role Technology Demonstrator

- Four technology investment agreements to industry partners to demonstrate an operationally representative mix of capabilities to investigate realistic design trades and enabling technologies.

- Bell and Sikorsky/Boeing designs will move forward to technology demonstration in 2017.
High-fidelity modeling and simulation to reduce risk, reduce cost, and enhance safety for new DoD acquisitions

Initially targeted technologies to address:
- Automation
- Rotor wake resolution
- Complex geometry
- Aero-structural coupling

DoD High-Performance Computing Modernization Program (HPCMP)
CREATE™-AV Helios

V-22 Osprey
Helios Multiple Mesh Paradigm

- Off-body Cartesian Solver
- Near-Body Flow Solver
- Interpolated Overset Mesh Connectivity

- Near-body solver for complex geometry
- Off-body solver for speed and adaptive mesh refinement
- Automated and efficient interpolation schemes between grids
1. “Tag” cells containing high vorticity
2. Cluster tagged cells into blocks
3. Use blocks to create finer level

→ Repeat

- **Efficient computational performance**
  - 3% overhead on 512 processors
  - Tested by LLNL for >100,000 processors

- Minimal overhead
- Parallel mesh generation
- Load balance by distributing blocks

Based on LLNL SAMRAI software
Helios Adaptive Mesh Refinement

V-22 Osprey model rotor in hover

- Rotor vortices identified by a scaled normalized vorticity criteria
- Cartesian meshes automatically refine and de-refine in order to follow the vortices in the rotor wake
Rotary-Wing Fluid-Structure Interactions

Boeing MD-900 Rotor
Helios Python-Based Architecture

Object Oriented Python Integration Framework

PUNDIT
- Domain Connectivity

MELODI
- Fluid Structure Interface and Mesh Motion

Distributed Memory processors communicating via MPI

shared data

P0 P1 P2 PN

Near-Body CFD Solver-1
- NSU3D (U. Wyoming)

Near-Body CFD Solver-2
- OVERFLOW (NASA)

Near-Body CFD Solver-3
- FUN3D (NASA)

Off-Body Solver
- SAMCart (LLNL/CREATE-AV)

Computational Structural Dynamics
- RCAS (AFDD US Army)
- CAMRADII (Johnson Aeronautics)

Co-Visualization

Paraview
Fieldview

Light-Weight
Main execution script
Few hundred lines of code
Minimal Overheads

Object Oriented
Multiple codes
Multiple languages
Generalized Interfaces

Next-generation flow solver modules!
CH-47 Interactional Aerodynamics with 3-Solver Paradigm

- Three flow solvers
  - OVERFLOW
  - NSU3D
  - Cartesian
- Dual-rotor
- CFD-CSD coupling
- Mesh adaption to capture rotor wake details
- Free-flight trim
- Boeing has performed full-flight envelope modeling for CH-47 using Helios
Helios simulations provide high-fidelity modeling of the coaxial rotor system, the fuselage, and the propulsor

Alan Egolf, Ed Reed (Sikorsky)
Helios simulations provide unique capabilities for modeling interactional aerodynamics effects between coaxial rotor system and propulsor.
Helios Wind Farm Simulations

Currently supported by LLNL as a “Grand Challenge” application to demonstrate parallel scalability on their 1.1M processor IBM Blue Gene/Q computer.
Ideal Problem:
- Solve the Navier-Stokes equations in parallel on each partition
- Nearest-neighbor communication along partition boundaries
- Should lead to near-perfect scalability on large numbers of parallel processors for large enough problems
- Memory per processor and processor speed are the main requirements here

Non-Ideal Scaling Factors:
- Multi-disciplinary analyses don’t usually scale as well as single-discipline ones
  - Fluid/structures, combustion, chemical reactions, radiation, etc.
- Adaptive mesh refinement and derefinement

Processor memory is a must (1-2GB per processor)
How do we effectively utilize future exascale computing hardware to solve rotorcraft aeromechanics modeling design problems?

**Strategies**
- Overlap flow solver execution and communication to reduce the adverse effects of unbalanced communications and communications latency
- Parallelize in both space and time
- Domain specific programming languages to most efficiently utilize a variety of new heterogeneous super computer architectures
- Co-Visualization to minimize I/O requirements
Super-scalable computations with multiple flow solvers requires overlapping the execution of all of the iterative flow solvers with asynchronous communication exchanges at the flow solver boundaries.
Parallelization in Time and Space for CFD Solvers

• Time-spectral formulation allows for problem formulation as a single combined space-time solution
  – Inherent periodicity for rotorcraft problems simplifies this process
  – Solution procedure already demonstrated for dual-mesh overset-grid problems with complex geometries

• Partition the problem for large parallel machines in both space and time
  – Planned scalability testing this summer for 100,000+ processors

1,000 processors in space x 1,000 processors in time = 1 million processors!
Heterogeneous Hybrid Hardware

- Cray Titan (Oak Ridge National Laboratory)
  - 18,688 nodes, 16 CPU cores/node, 2 NUMA domains
  - 2,496 CUDA cores/node with separate memory

- Tianhe-2 (China’s National University of Defense Technology)
  - 17K nodes, 3M cores
  - 2 CPU’s _ 3 Xenon Phis per node
  - 1.34 PB total RAM

- Qualcomm Snapdragon S4
  - 4 CPU cores, GPU, video, …
  - 2 memories, 3-level cache hierarchy
  - N-Levels of Cloud
**Domain Specific Language (DSL) for Hybrid Architectures**

**Purpose**
Domain-specific programming languages will automatically map parallel CFD flow solvers toexascale computing architectures, including in-chip computations

**Practical example**
1. Liszt is a high level language for solving PDEs.
2. Liszt uses different strategies to parallelize programs for clusters, SMPs, GPUs, and FPGAs.
3. Liszt performance comparable to best hand-written code.

**Results / Future Work**
The SU2 code (2nd-order accurate, finite volume, unstructured cell-based, implicit RANS) solver developed within the ADL has been ported to Liszt.

---

**Pat Hanrahan and Alex Aiken, Stanford University**

```
// Space integration (inviscid contribution)
for (f = zone_interior) {
  calcRhsInteriorInviscid(f); calcJacInteriorInviscid(f);
}

// Space integration (viscous contribution)
if (IC.nu_ref > 0.0) {
  for (f = zone_interior) {
    // Rhs and Jacobian for the mean flow
    calcRhsInteriorViscous(f); calcJacInteriorViscous(f);
    // Rhs and Jacobian for the turbulence model
    calcRhsInteriorViscousTurb(f); calcJacInteriorViscousTurb(f);
  }
}

// Space integration (source contribution)
for (c = cell(mesh))
  calcRhsSource(c); calcJacSource(c);
```

---

**Performance of the language using different examples**

<table>
<thead>
<tr>
<th>Applications</th>
<th>Euler</th>
<th>NS</th>
<th>FEM</th>
<th>SW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speedup over Scalar</td>
<td>40x</td>
<td>30x</td>
<td>20x</td>
<td>10x</td>
</tr>
</tbody>
</table>
Helios CFD Co-Visualization Module

- Transient 3-D datasets are too large for file transfer and display on local workstations.
- Helios users specify cutting planes and isosurfaces during problem setup.
- ParaView plug-in module writes 2-D extracts to disk at runtime.
- Users then transfer 2-D extracts back to local workstations for display.
- Developed by Kitware as part of an Army SBIR project.

Co-Visualization for Post-Processing
- High-fidelity rotor wake capturing
- Full-vehicle geometric complexity
- Reduce turnaround time for high-fidelity computational aeromechanics simulations by 1000x or more

- What’s next?
Exascale Computing for Future Vertical Lift Aircraft Development

Requirements:
• Exascale computing required for multi-disciplinary design and optimization
  – Need to increase current scalability by 1000x for acceptable turnaround time
• Automation for faster problem setup and guaranteed accuracy
  – Need to take computational fluid dynamics out of the hands of specialists and into the hands of aircraft designers

Products:
• Accurate, efficient, easy-to-use, and validated aeromechanics modeling and simulation tools based on computational fluid and structural dynamics on high-performance parallel computers

Payoff:
• Greatly reduced time, cost, and risk for advanced aircraft development, and fielded system upgrades
• High-performance, safe and reliable aircraft for the DoD
● Rotorcraft design using fully-automated high-fidelity modeling and simulation including multi-point adjoint-based optimization