RDECOM

Presented to:
12th Overset Grid Symposium
October 7, 2014

High-Performance Computing for Rotorcraft Modeling and Simulation

AMRDEC 50 YEARS
Staggering Accomplishments… Limitless Possibilities

TECHNOLOGY DRIVEN. WARFIGHTER FOCUSED.

Dr. Roger Strawn

US Army Aviation Development Directorate (AFDD)
Ames Research Center, Moffett Field, CA

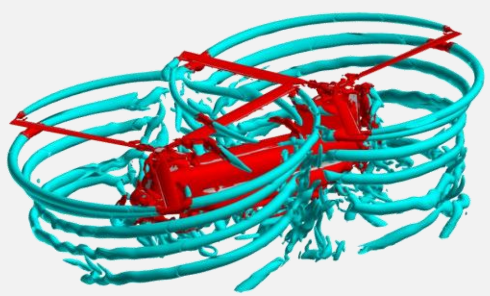
DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited

US Army Aviation Development Directorate

Located at Redstone Arsenal, Fort Eustis, NASA Ames, and NASA Langley Research Centers

Human Systems and Flight Control

Wind Tunnel Testing



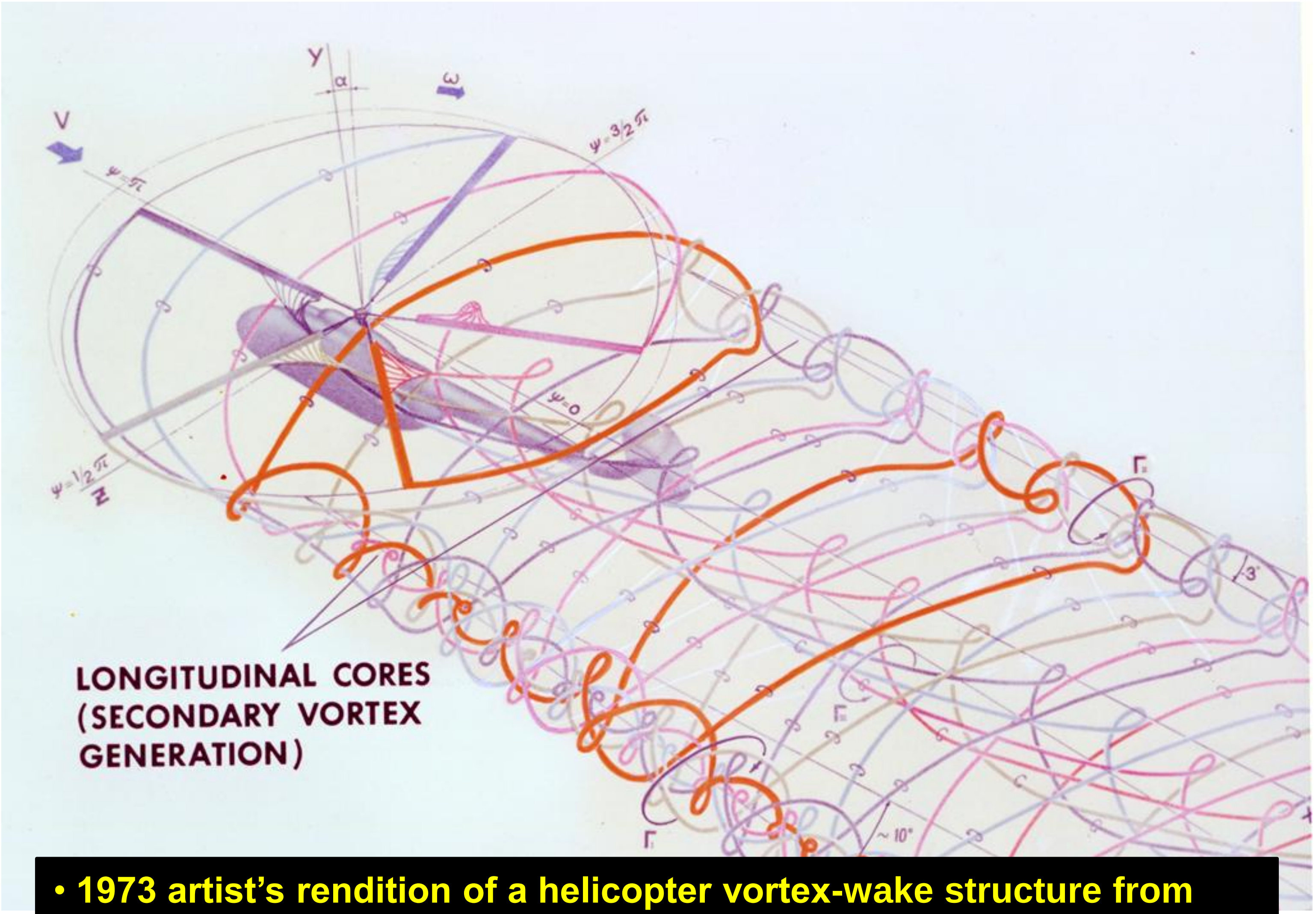
Computational Modeling

Preliminary Design

From Computations to Flight !

Flight Testing

Rotary-Wing Aeromechanics



• 1973 artist's rendition of a helicopter vortex-wake structure from Aviatsiya I Kosmonautika, a monthly Soviet-era aviation magazine

Recent DoD Rotorcraft Procurements

RAH-66 Comanche (1983-2004)

Canceled in 2004

V-22 Osprey (1983–2005)

Aeromechanics Problems in Development

- Shed vortices from hub interfered with tail control
  - Complete redesign of empennage in 2000
- Basic physics of fantail performance were poorly understood
- Main rotor regressive lag-mode instability

- Control problems during rapid descent
  - 19 Marines died in 2000 crash
- Poor hover performance
- Pitch up with 45 deg. crosswind
- Loss of lift with 90 deg. crosswind

We can do better !

CFD to Support DoD Aircraft Acquisition Programs

- **High-fidelity modeling and simulation to reduce risk, reduce cost, and enhance safety for new DoD acquisitions**

HPCMP CREATE™-AV Helios




Joint Multi-Role Rotorcraft

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

Helios Software Description

Computational Models and Solution Algorithms

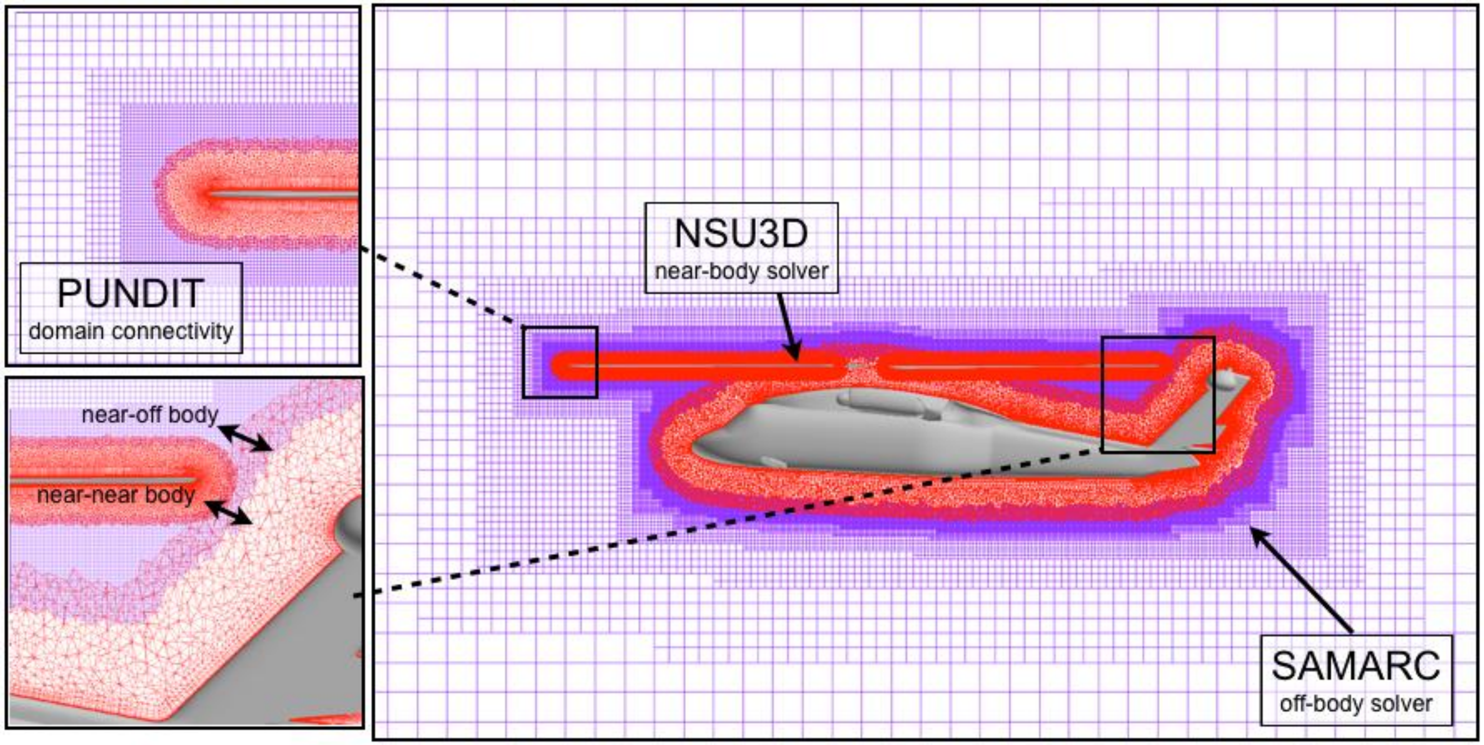
- Dual mesh approach (near-body / off-body)
- Automated overset grid domain connectivity
- Adaptive mesh refinement for wake capturing
- Aerodynamics and structural dynamics coupling

Software Architecture:

- Lightweight
- Object oriented
- Efficient execution on large parallel computers
- Modular components with generalized interfaces
- Extensible

How do you get 10 people to work efficiently together on CFD software development?

Helios Dual-Mesh CFD Paradigm



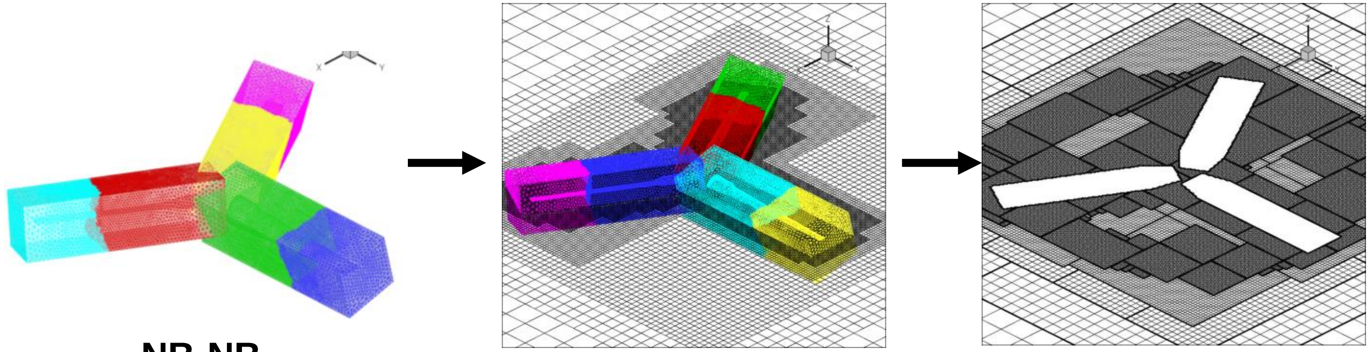
Unstructured "near-body"
- Resolve near-wall viscous flow
- Complex geometries

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

Fully automated domain connectivity for near-body and off-body meshes

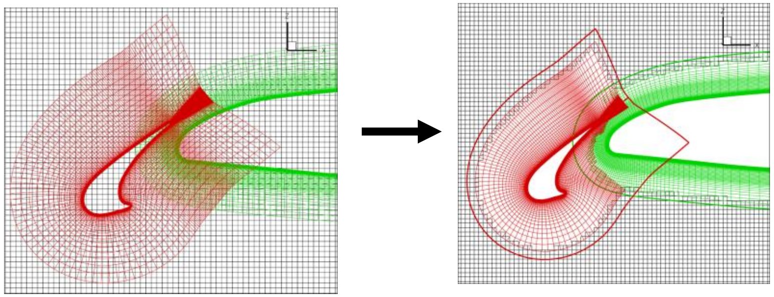
"PUNDIT" Overset Mesh Connectivity

- **Manages data interpolations between all overset meshes**
- **Trims back overlapping meshes and ensures optimal overlaps geometries between both near-body unstructured and off-body Cartesian meshes**



NB-NB connectivity → **OB grid generation** → **Hole Cutting & NB-OB connectivity**

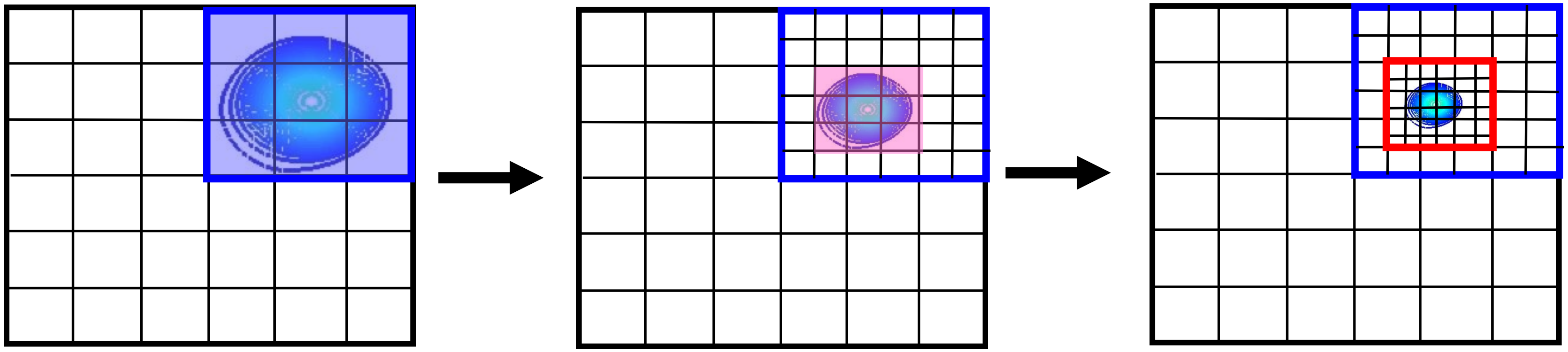
- **Implicit hole-cutting ……. automated with no required user input !!**
- **Constructs donor/receiver information between moving grids**



Implicit hole cutting

Cartesian Adaptive Mesh Refinement

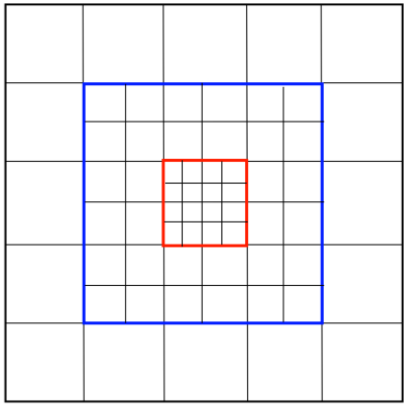
Based on LLNL SAMRAI software



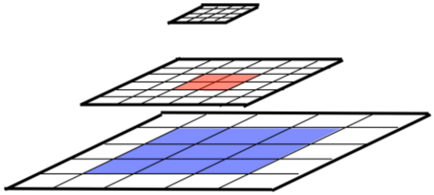
Coarse level **Intermediate** **Fine**

1. “Tag” cells containing high vorticity
2. Cluster tagged cells into blocks
3. Use blocks to create finer level

→ *Repeat*



Forms hierarchy of nested levels

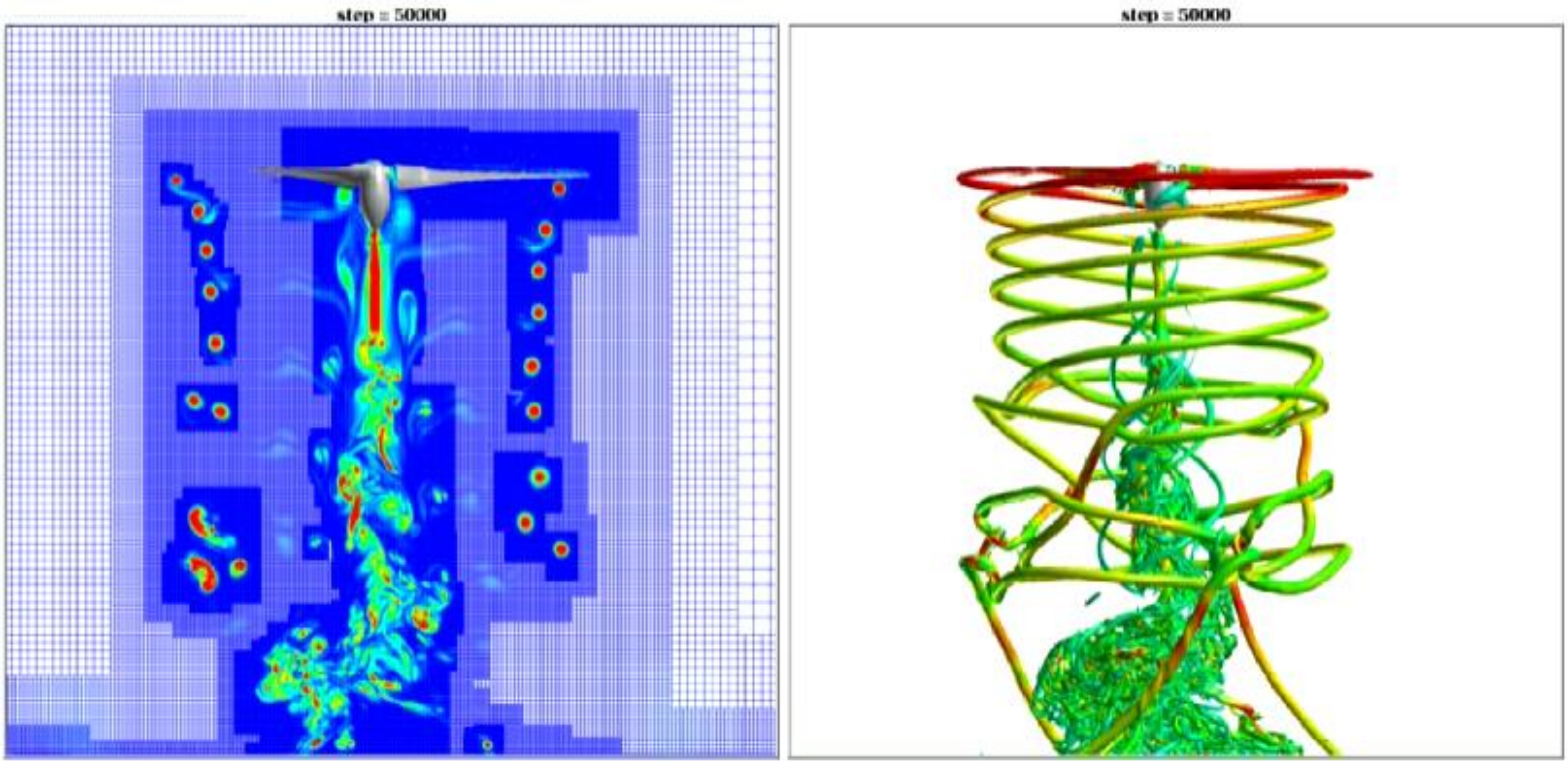


- **Efficient computational performance**
  - 3% overhead on 512 processors
  - Tested by LLNL for >40,000 processors

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

Helios Adaptive Mesh Refinement

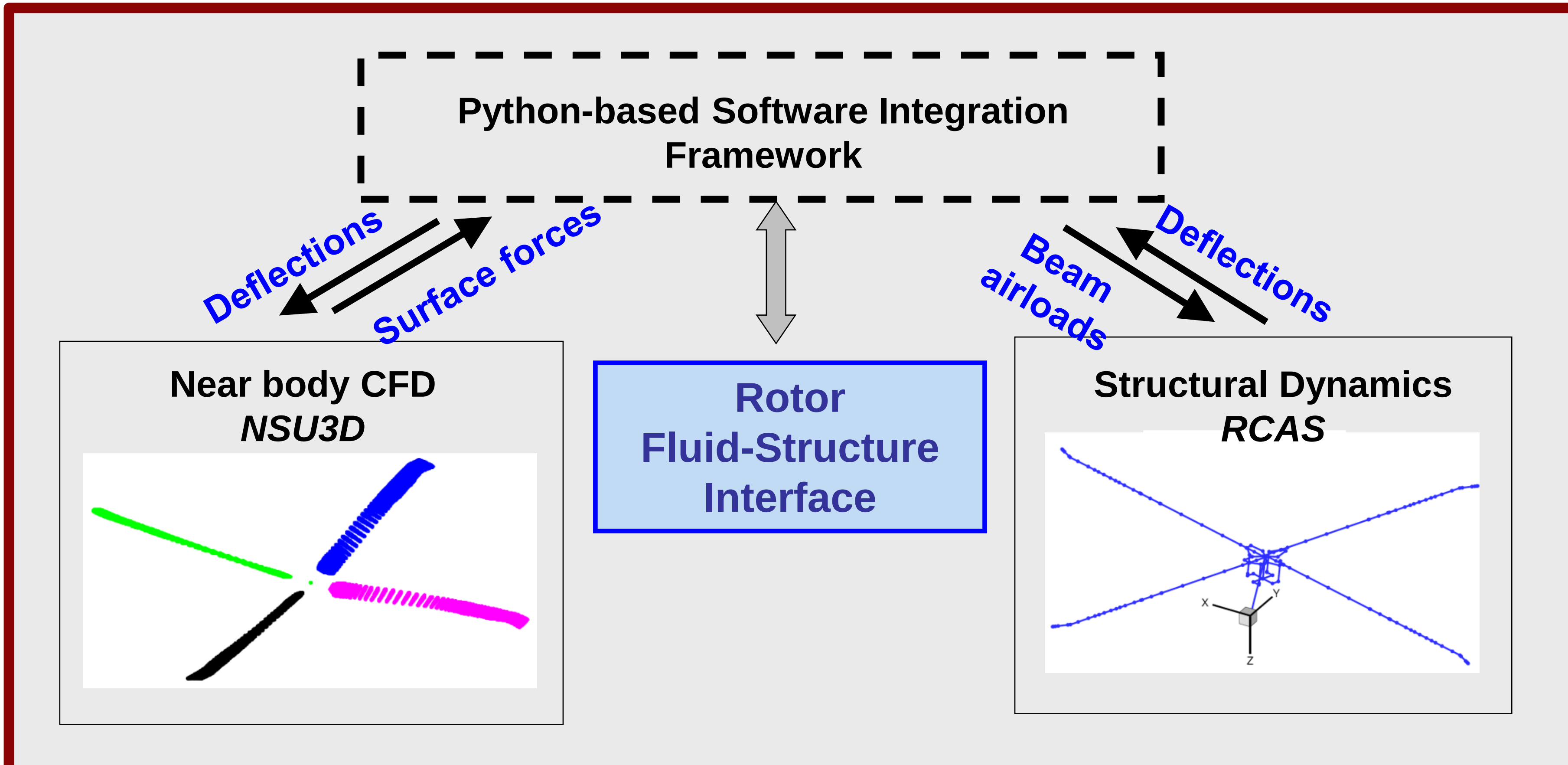
V-22 Osprey model rotor in hover



- 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

Helios Fluid-Structure Interface

- **Near-body CFD solver computes rotor surface forces**
- **Structural dynamics solver receives non-linear beam airloads, computes deflections, trim angles**
- **Near-body grid appropriately moved/deformed**



Helios Software Description

Computational Models and Solution Algorithms

- Dual mesh approach (near-body / off-body)
- Automated overset grid domain connectivity
- Adaptive mesh refinement for wake capturing
- Aerodynamics and structural dynamics coupling

Software Architecture:

- Lightweight
- Object oriented
- Efficient execution on large parallel computers
- Modular components with generalized interfaces
- Extensible

How do you get 10 people to work efficiently together on CFD software development?

Monolithic Multidisciplinary CFD Software Architecture

Surface and Volume Mesh Generation

I/O

Domain Connectivity Pre-processing

I/O

CFD Flow Solver

- Parallel partitioning
- Parallel communication
- Dynamic domain connectivity
- Mesh motion management
- Adaptive mesh refinement
- CFD solution algorithm
- Turbulence model
- Boundary conditions
- Software build environment (Makefile)

How do we leverage the work of additional software developers ??

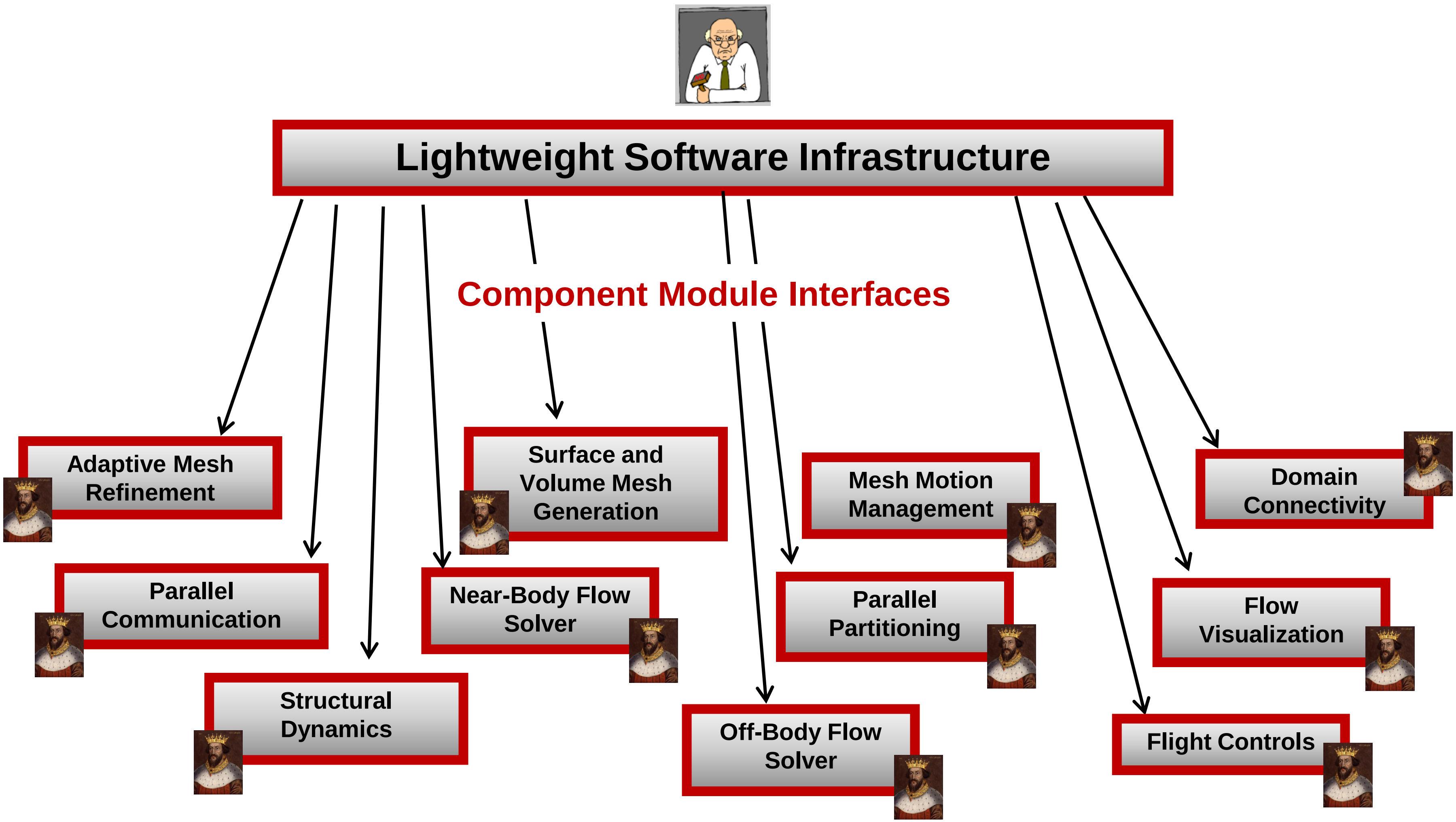
Structural Dynamics Solver

I/O

I/O

Flow Visualization

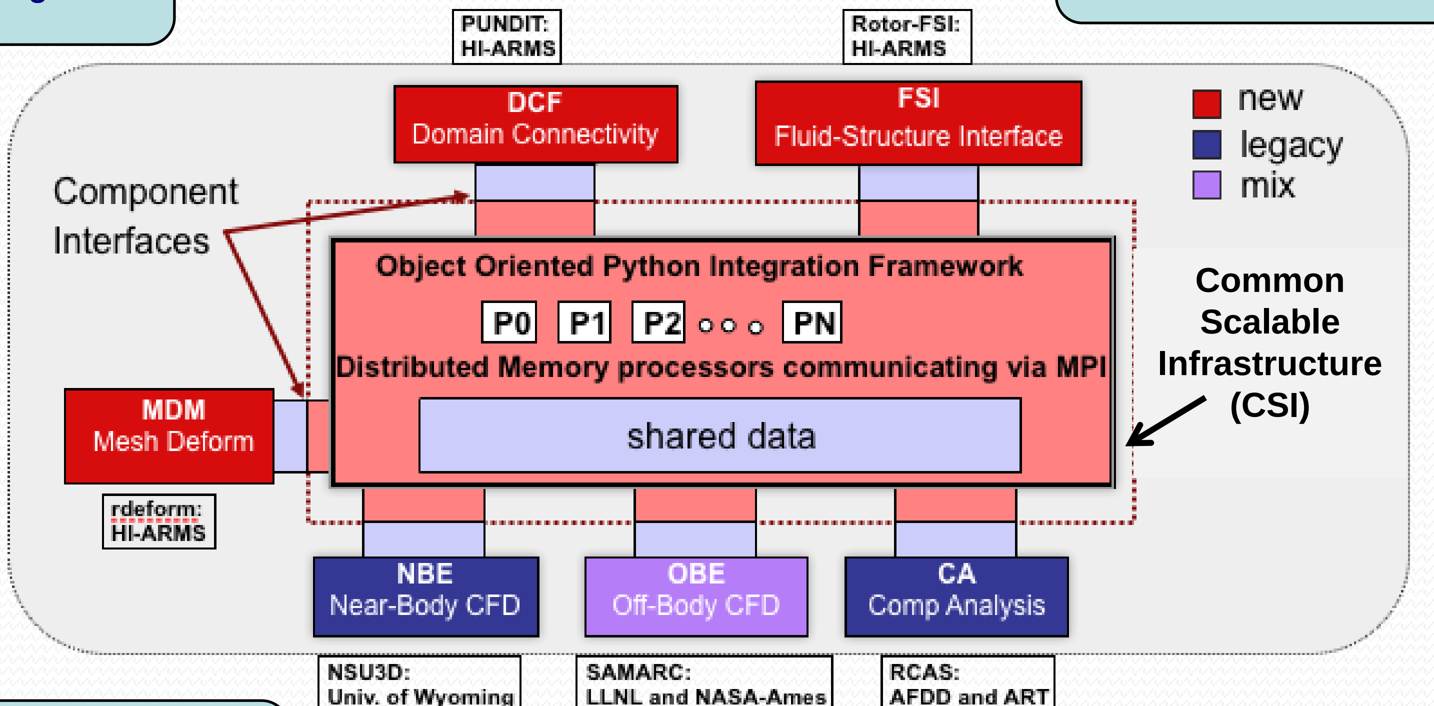
Component Module Multidisciplinary CFD Software Architecture



Helios Python Infrastructure

Object Oriented:
- Multiple codes
- Multiple languages

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



Interfaces
- Generalized interfaces
- Interoperable and extensible

Minimal Overheads
- Storage
- Efficiency

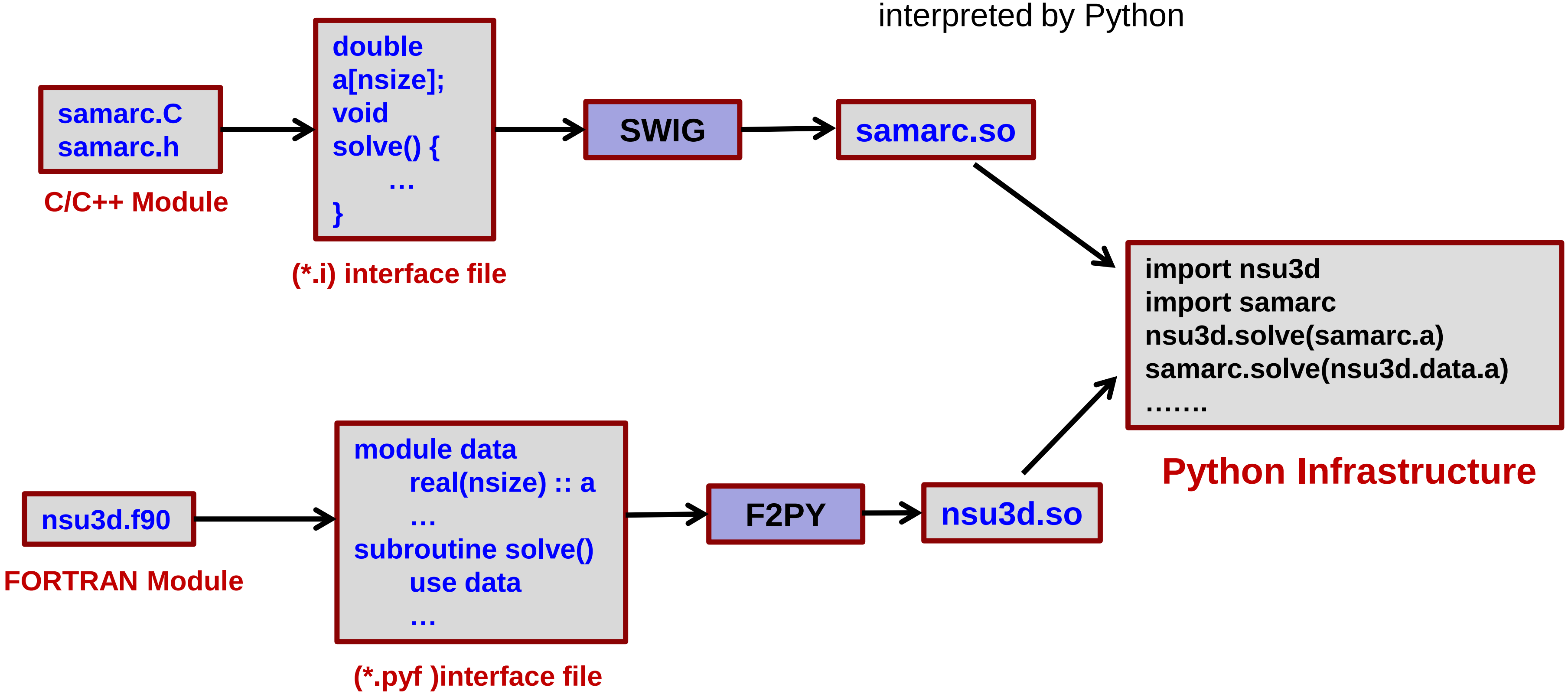
Python Wrapping for C, C^{++}, and FORTRAN

C/C^{++} Wrapping

- SWIG connects programs written in C and C++ with Python interpreter.
- Automatically parses C/C++ interfaces to generate the 'glue code' required for Python
- Resulting dynamic library (.so) can be interpreted by Python

FORTRAN Wrapping

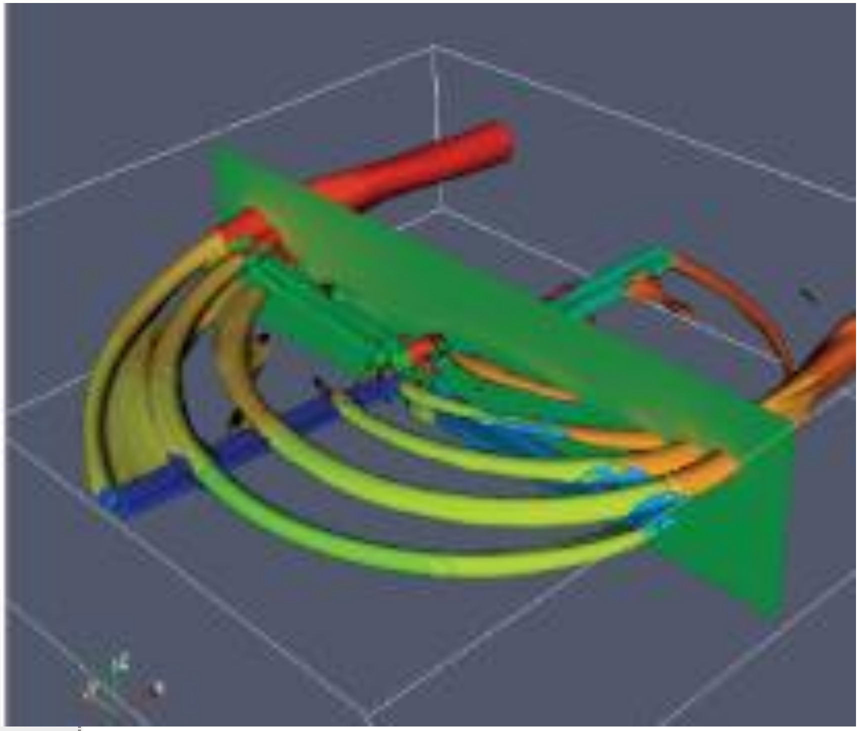
- f2py (.pyf) signature file contains all information about data pointers, functions and calling arguments
- f2py generates the C wrappers and compiles the FORTRAN source code
- Resulting dynamic library (.so) can be interpreted by Python

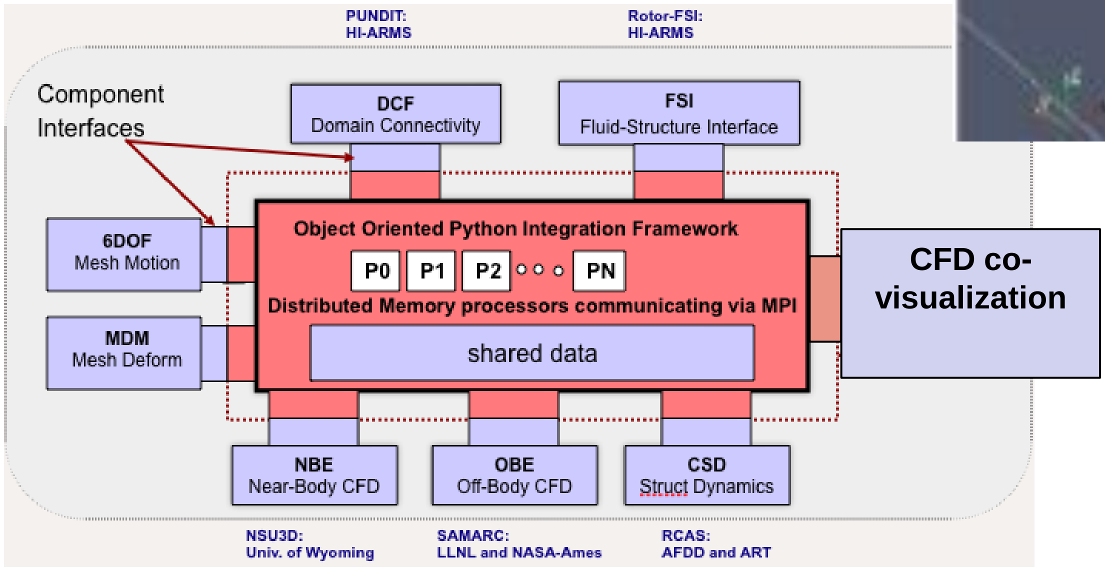


Adding New Component Modules

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

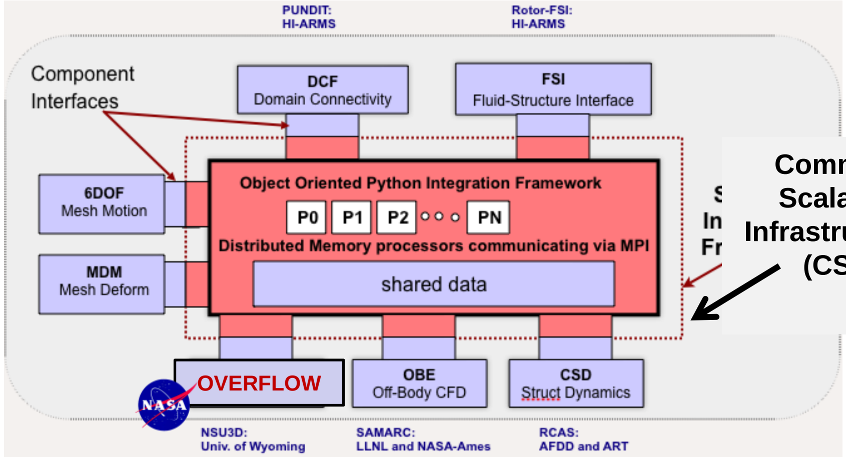




Substitution of Interchangeable Component Modules

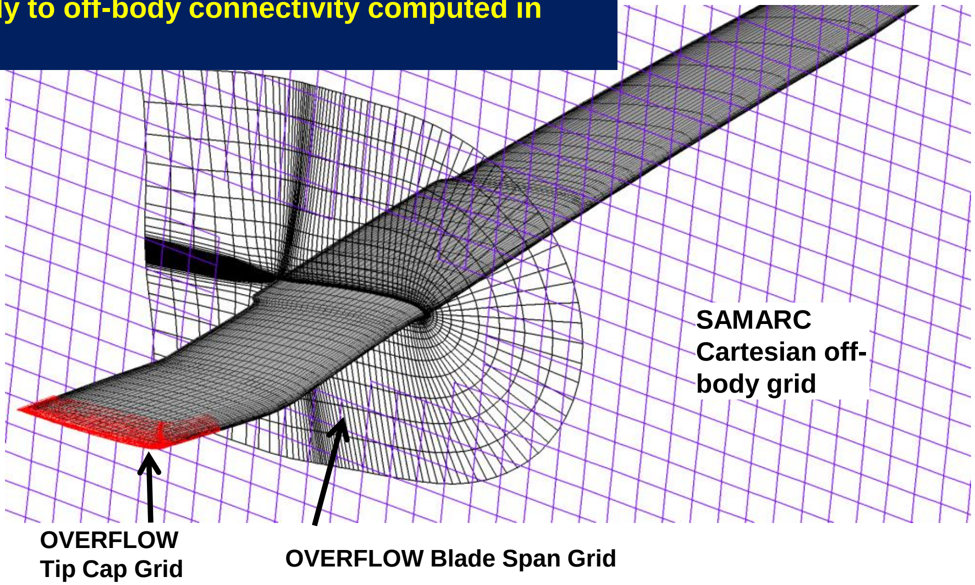
NASA OVERFLOW for Helios Near-Body solver

- Many existing OVERFLOW rotor grids
- Faster OVERFLOW execution time
- Higher OVERFLOW spatial accuracy
- OVERFLOW module interface is virtually the same as NSU3D module interface
- Three-solver paradigm with OVERFLOW for rotors, NSU3D for fuselage and Cartesian for rotor wakes



Common Scalable Infrastructure (CSI)

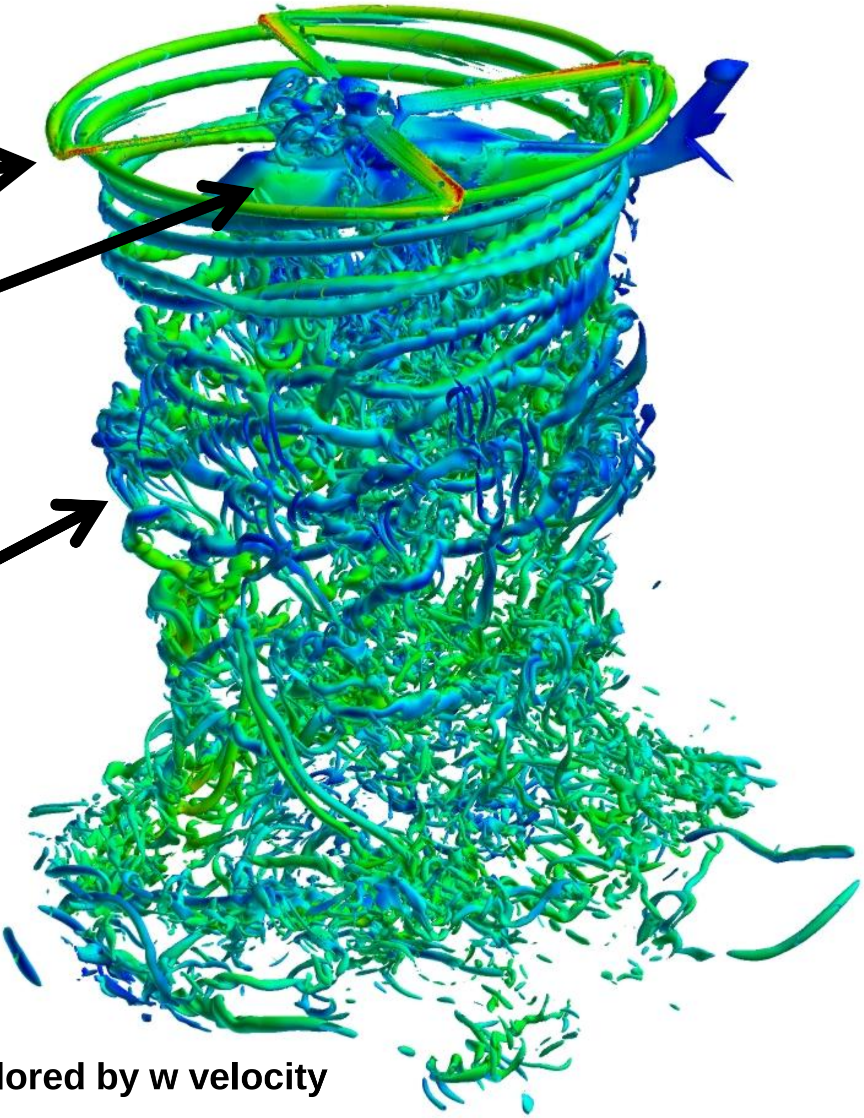
Near-body to off-body connectivity computed in Helios



OVERFLOW computes near-body to near-body surface and volume mesh connectivity

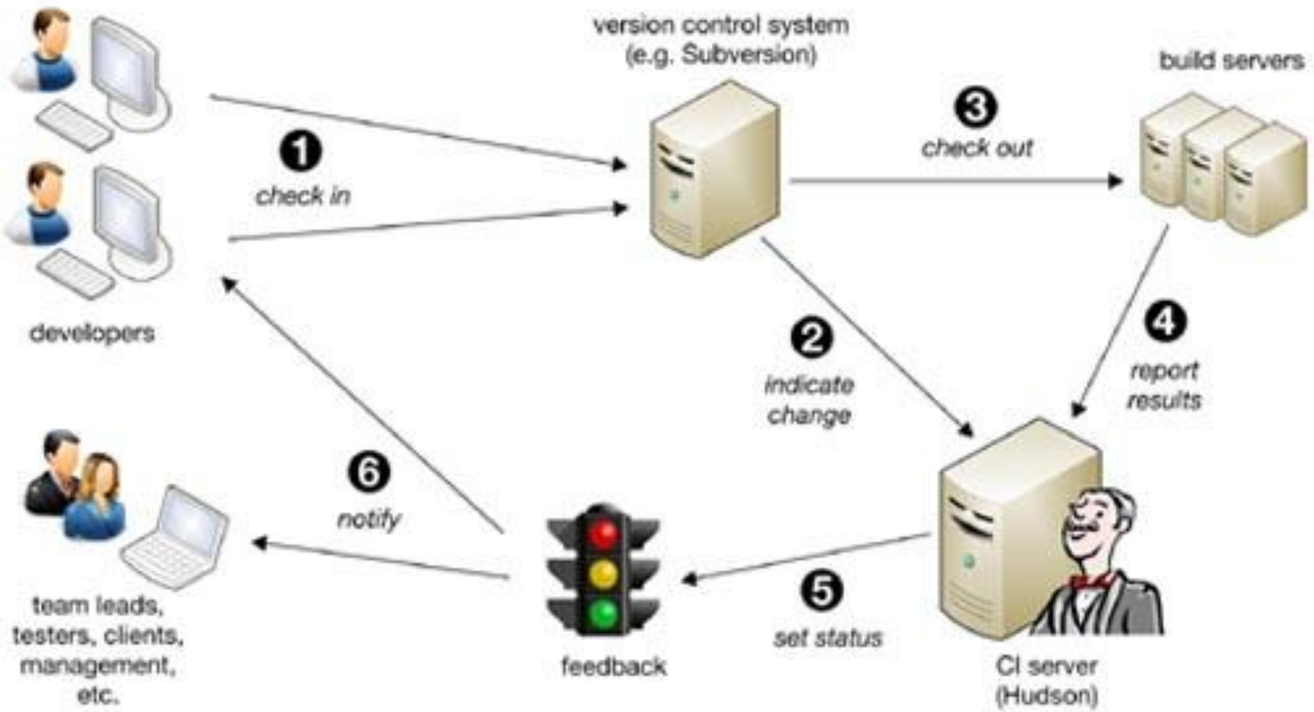
Three-Solver CFD Paradigm

- **OVERFLOW for blade grids**
  - 4^{th} order accuracy
  - Simple geometry
- **NSU3D for fuselage**
  - 2^{nd} order accuracy
  - Complex geometry
- **Cartesian solver for off-body**
  - 4^{th} order accuracy
  - AMR for rotor wake capturing



Q isosurface colored by w velocity

Helios Software Management

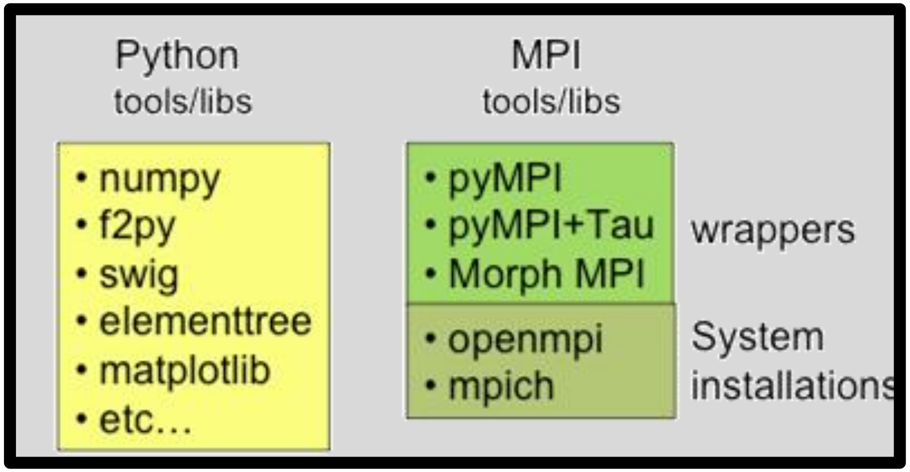


- **Individual SVN repositories for component modules and for integrated Helios build**
- **Use Jenkins for continuous integration**
- **Beta testing by US helicopter industry and US government labs**
- **GUI for user input and resources for user help**
  - JIRA , Wiki, Web based support
  - User manual, test suites, tutorials

Helios Build/Runtime Environment

- **Helios requires a variety of freely available build & runtime system libraries**

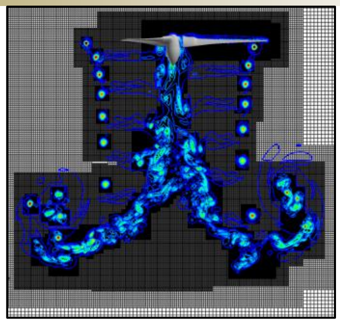
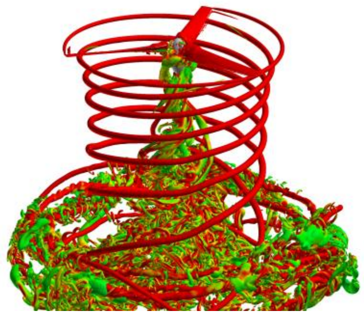
- **Difficult to install this environment consistently across systems**
  - Requires expertise in compilers, linkers, runtime systems, and system administration
  - Very time consuming, even for computer scientists



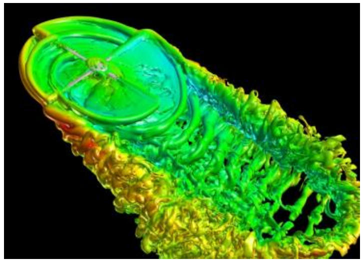
Don't want to force Helios users to shoulder this burden

- **Collaborate on the development of a generalized HPC-based Python build & runtime environment**
  - Automated installation of build environment prior to new Helios installations
  - pTools build environment developed by Sameer Shende from ParaTools Inc

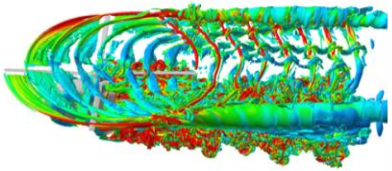
Helios Computational Simulations

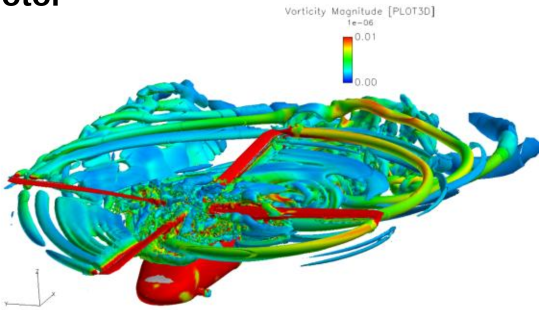
V-22 model rotor

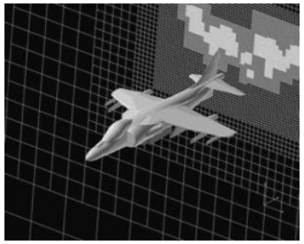
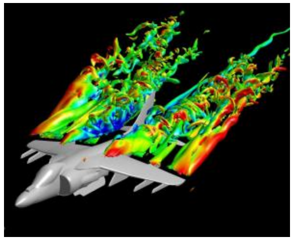
Sikorsky X-2

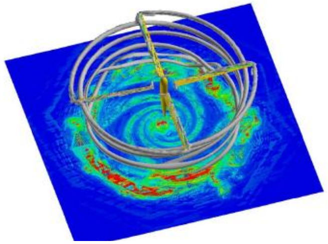
HART-II rotor and test stand



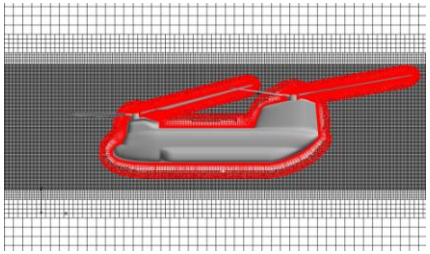
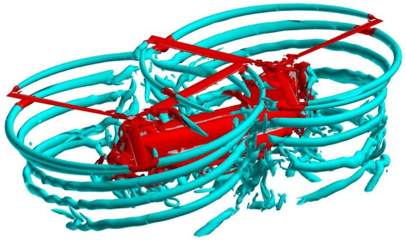
UH-60 rotor and fuselage

AV-8 Harrier

Apache composite main rotor

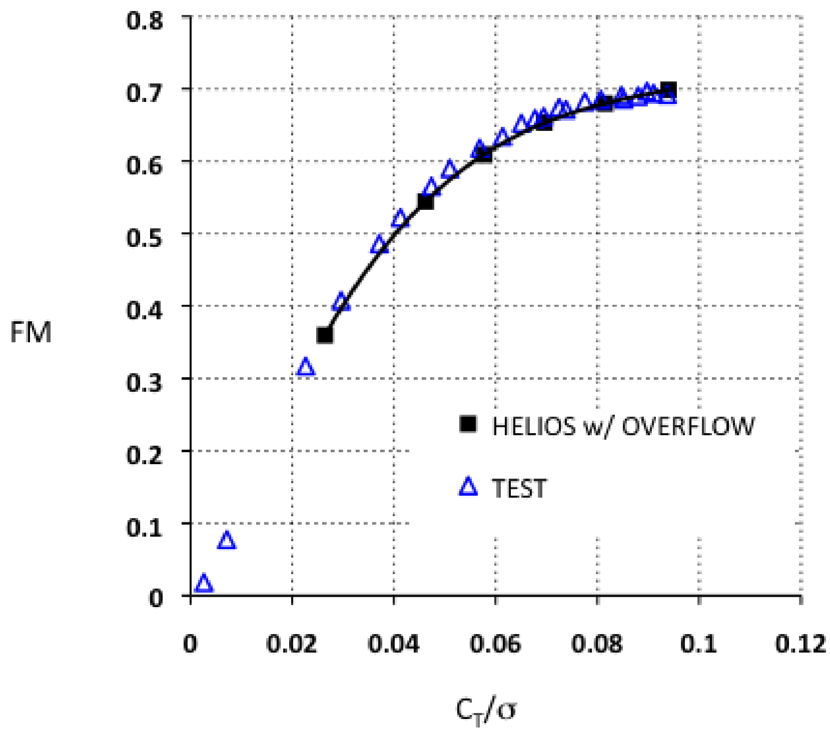



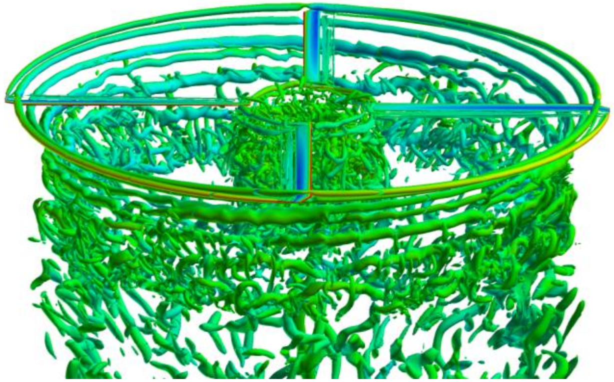
CH-47 Chinook rotor and fuselage

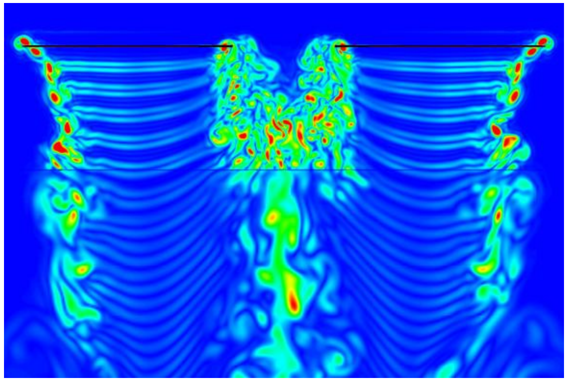
High-fidelity computational aeromechanics for general multi-rotor and rotor/fuselage combinations

Helios Validation for Hover Performance Prediction

- January 2013 AIAA Hover Prediction Workshop focused on S-76 hover performance
- Helios showed excellent results for S-76 hover performance prediction compared with other CFD solvers from national and international participants







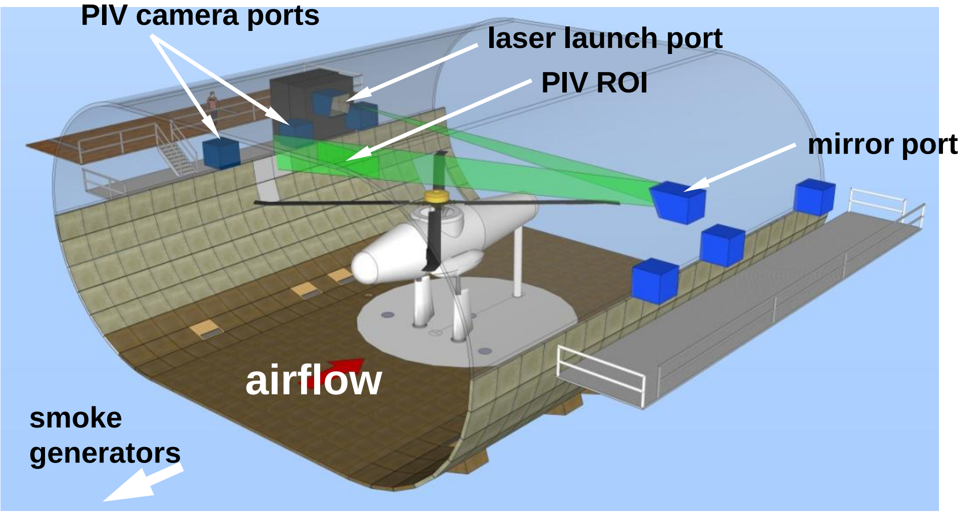
Test and Helios predictions agree to within 1 count in FM. (Reported experimental uncertainty in FM ~0.6 counts)

Mark Potsdam, Rohit Jain (Army ADD)

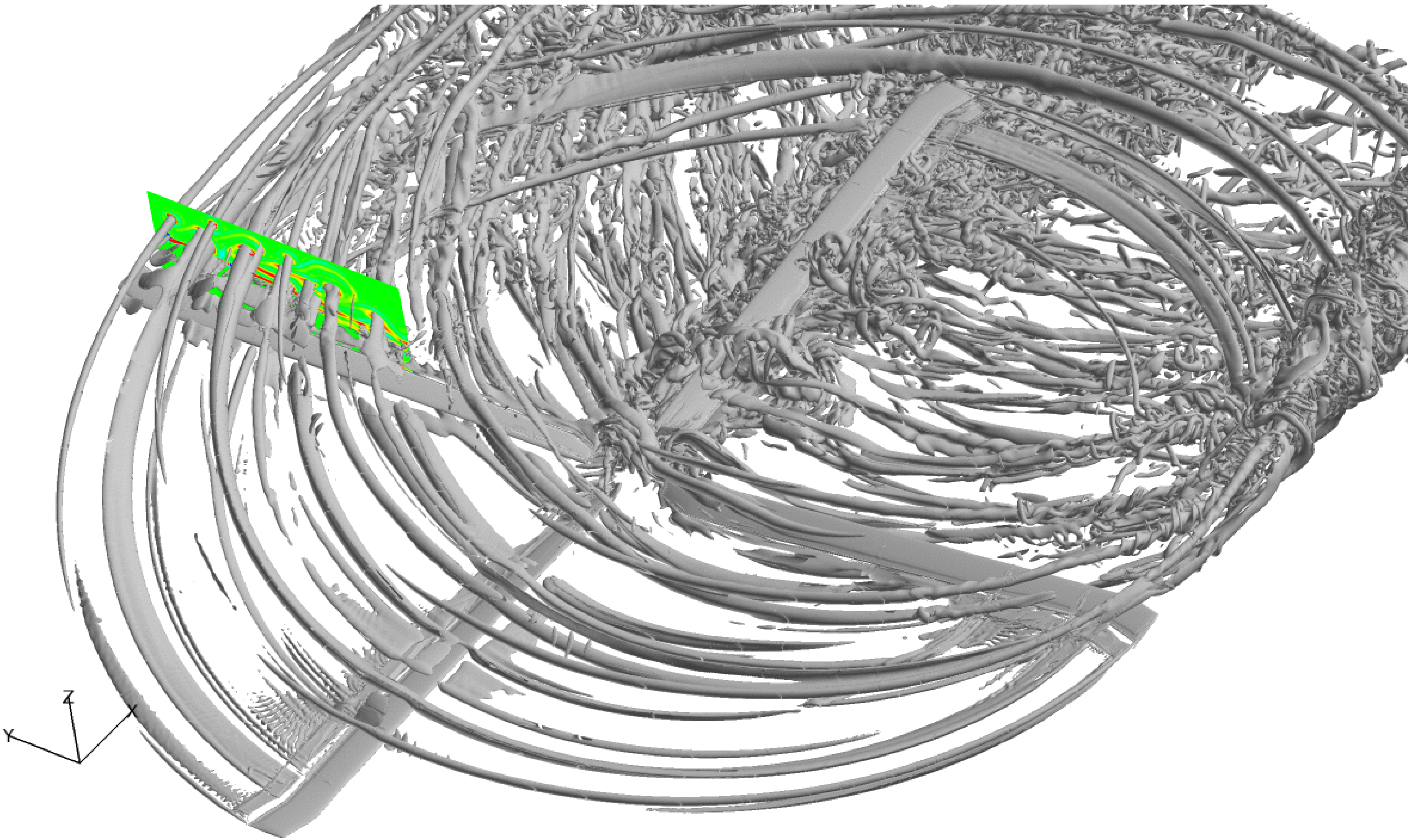
UH-60A WIND TUNNEL TEST

- Full-scale UH-60A rotor in Air Force 40- by 80-foot Wind Tunnel under NASA/Army Airloads Wind Tunnel Test Program (2010)
  - Extensive database on highly instrumented rotor for validating analytical tools
    - Performance, hub loads, air loads, structural loads, *wake PIV, blade deformation, RBOS*
- PIV phase
  - Wake measurements at 90 deg azimuth over 50% of outer blade
  - 7 PIV test conditions (μ = 0.15 – 0.304, varying C_T/σ)
  - Vortex characteristics extracted: circulation, size, position


NASA/Army UH-60A NFAC test

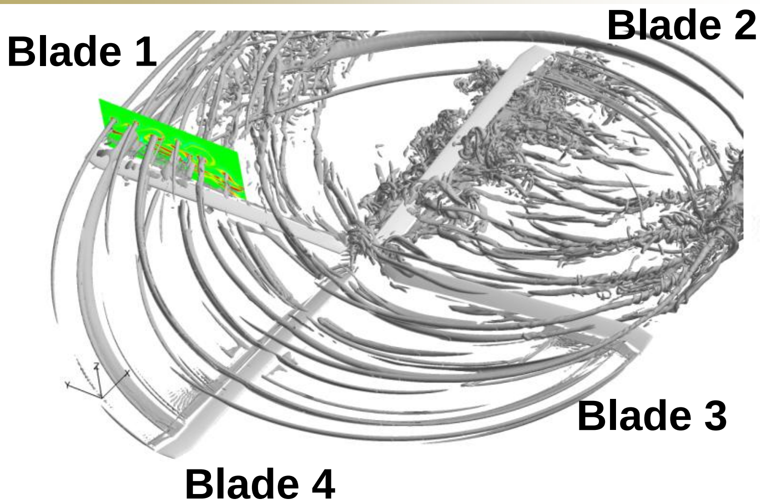


UH-60A Low Speed Wake

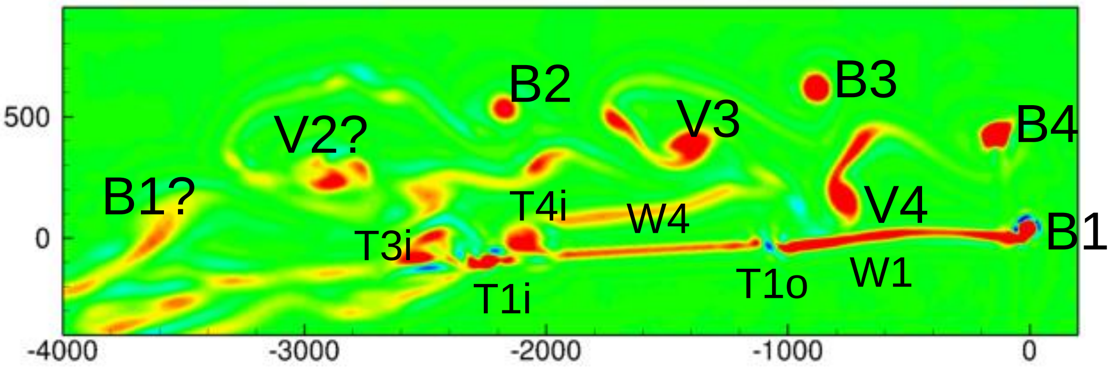


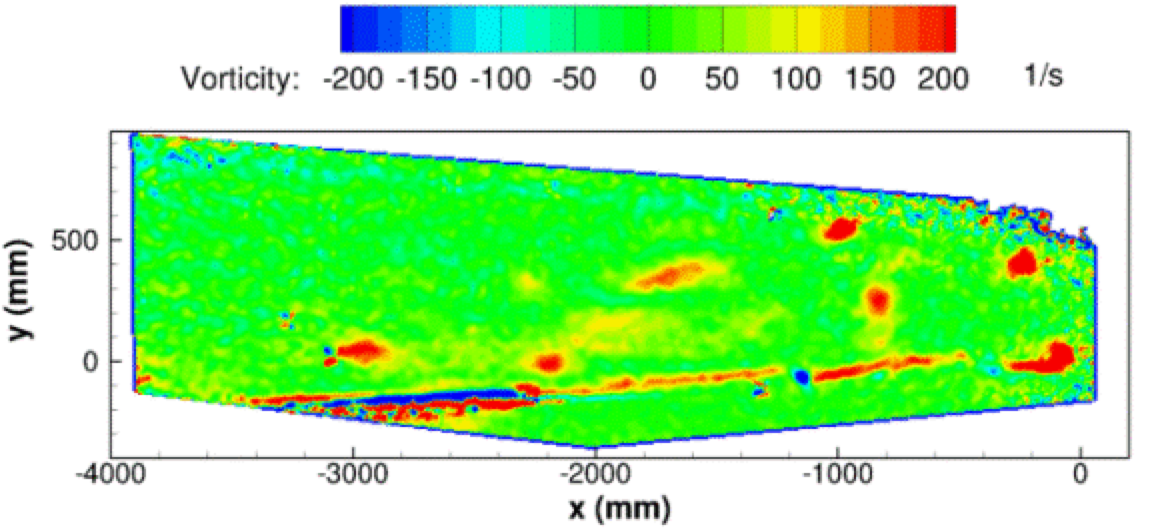
Mark Potsdam, Buvana Jayaraman (Army ADD)

UH-60A Vortex Flow Structures



CFD



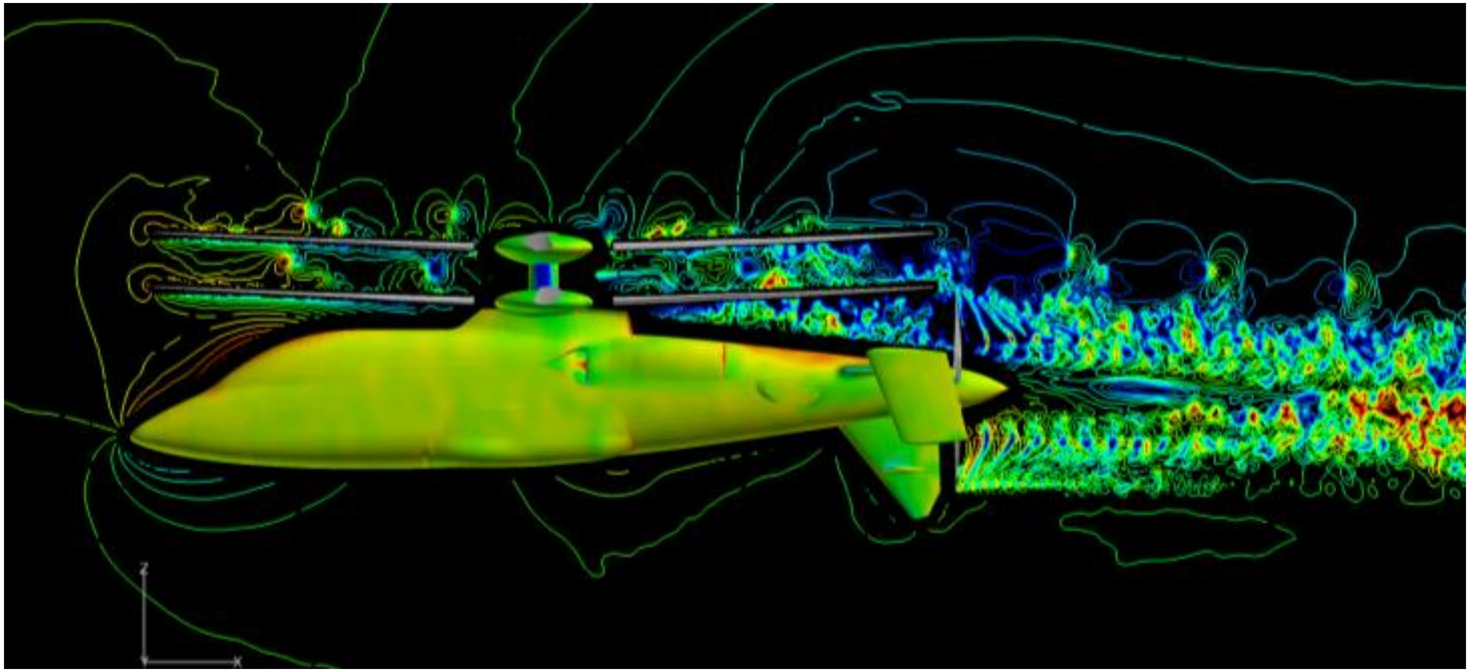


PIV

- Blade tip vortex (B)
- Twist/planform vortex (V)
- Trim tab vortex (T)
  - Inboard (i)
  - Outboard (o) of opposite sense
- Wake sheet (W)
- Wake goes above rotor plane
- V persists independently of B

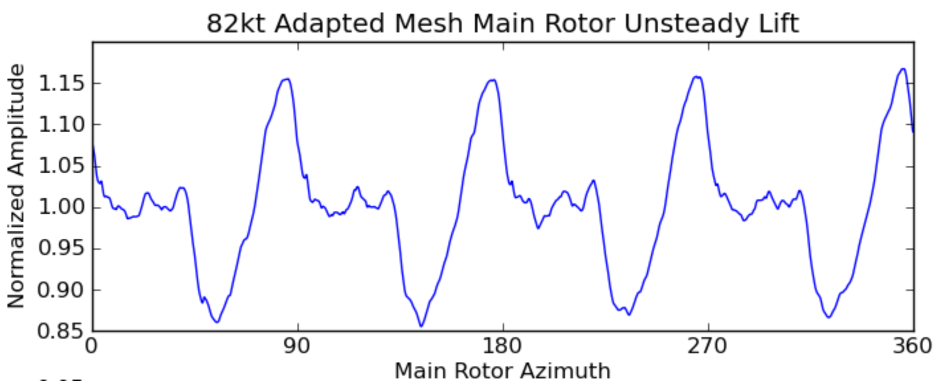
Sikorsky X-2 Helios Simulations

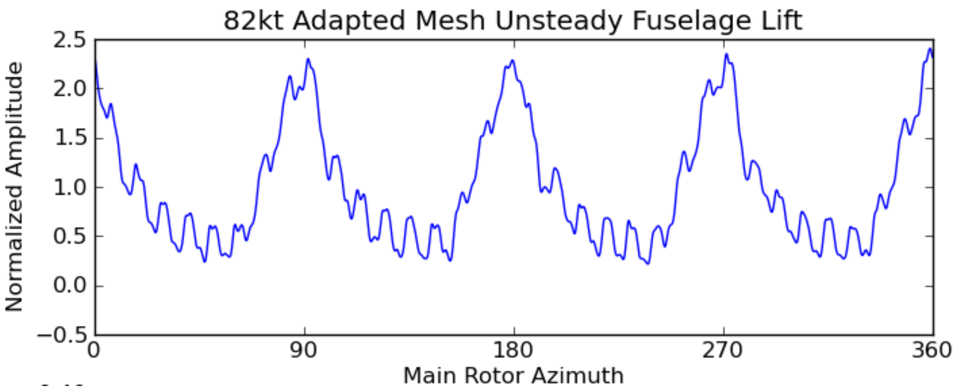
Alan Egolf, Ed Reed (Sikorsky)

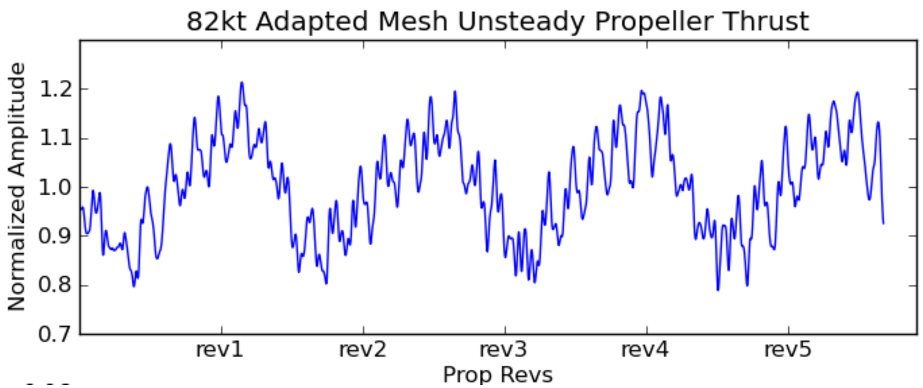


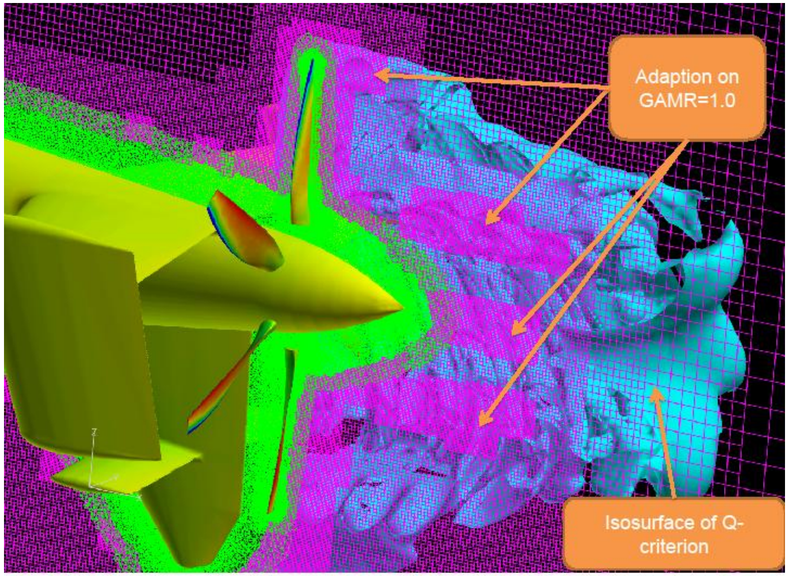
- Helios simulations provide high-fidelity modeling of the coaxial rotor system, the fuselage, and the propulsor

Sikorsky X-2 Helios Simulations



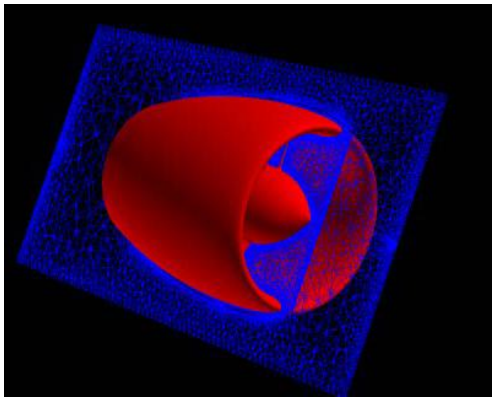


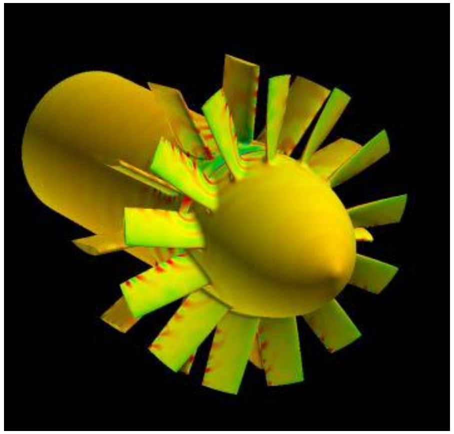


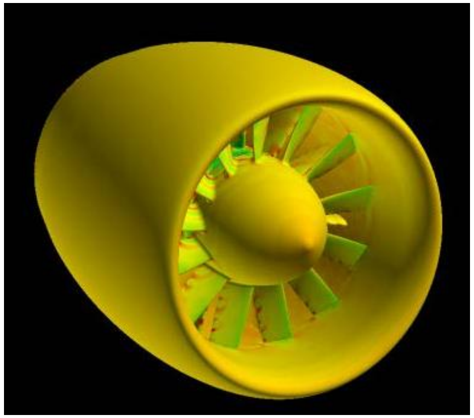


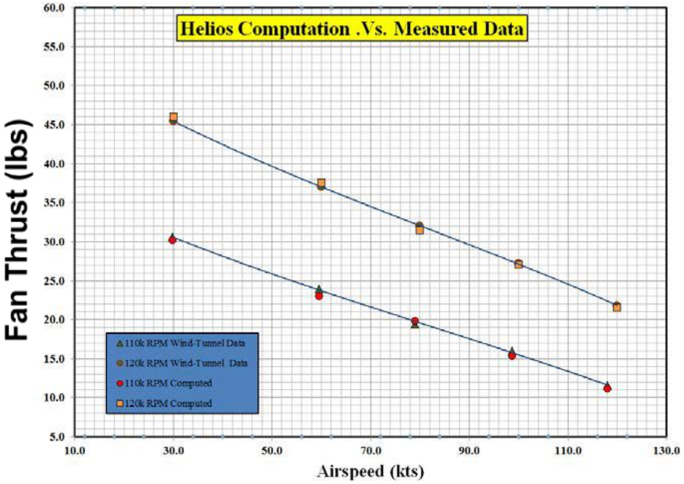
Helios simulations provide unique capabilities for modeling interactional aerodynamics effects between coaxial rotor system and propulsor

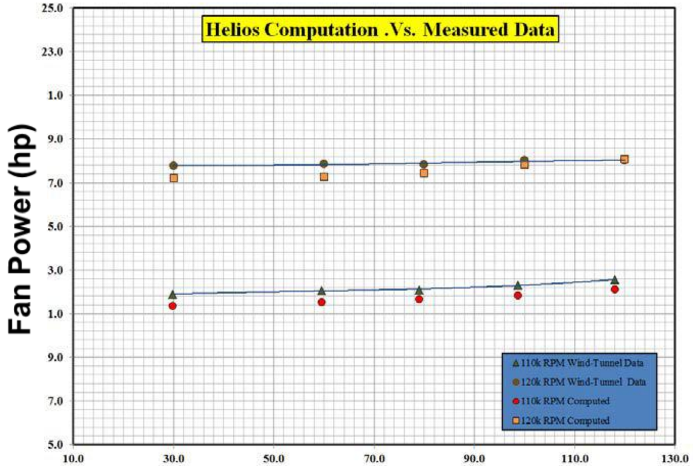
Boeing Ducted Fan Simulations with Helios











Airspeed (kts)

Hormoz Tadghighi (Boeing)

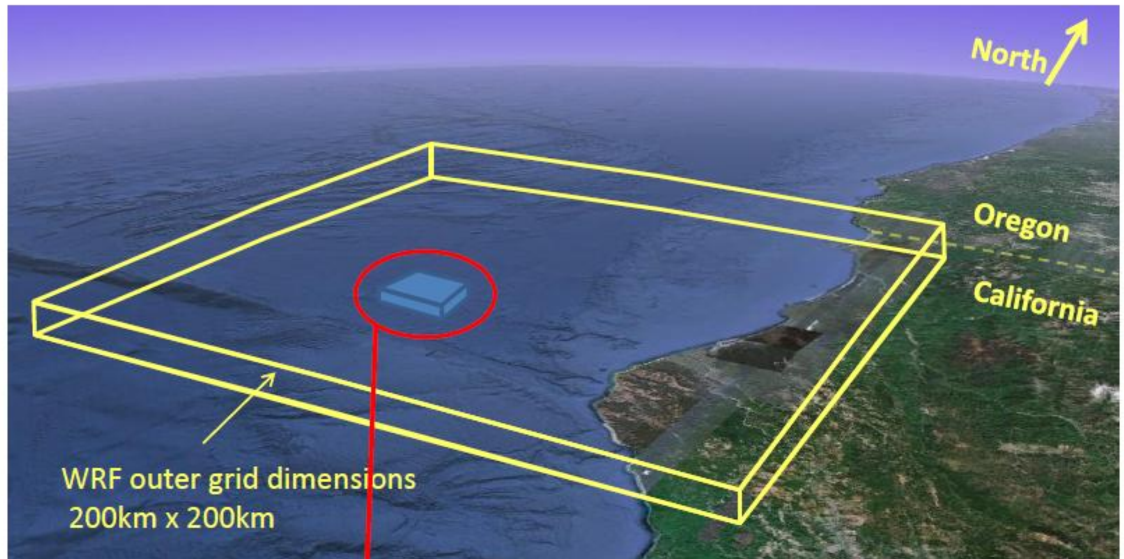
Boeing X-Plane Simulations with Helios

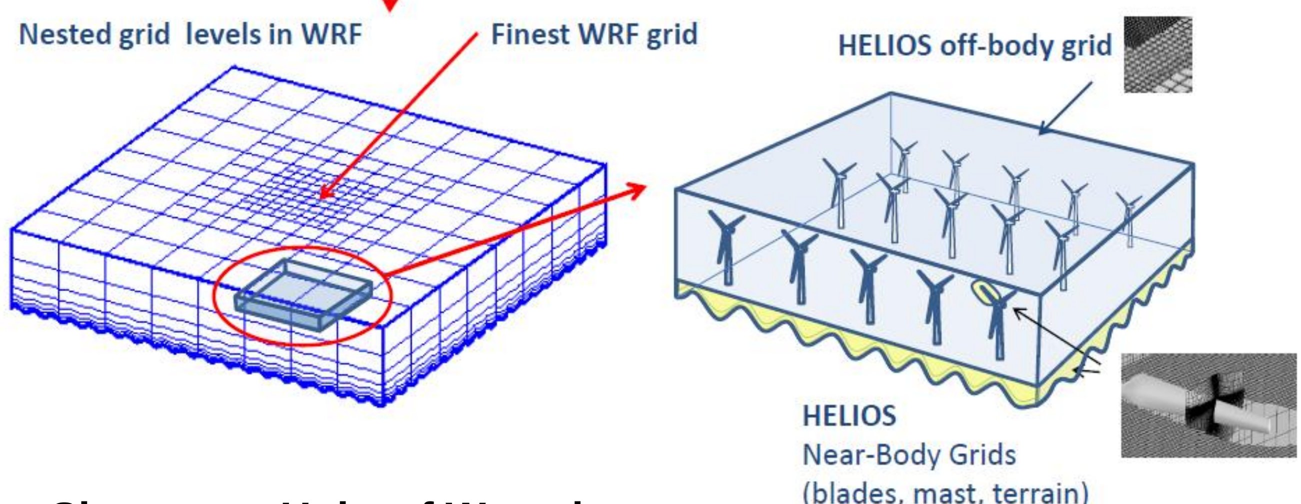
- **DARPA's X-Plane competition is looking for a VTOL aircraft that can fly fast, hover efficiently and carry lots of cargo**
- **Helios is being used extensively to help Boeing develop its "Phantom Swift" X-plane design**
  - Helios modeling for both the ducted fan analysis and for the interactional aerodynamics between the rotors and the fuselage

Boeing's Phantom Swift prototype

Helios Coupled to WRF for DOE Wind Farm Modeling

- Collaborative effort involving Lawrence Livermore National Laboratory (LLNL), Univ. of Wyoming and Army AFDD
- Meso-scale large-eddy simulations with NCAR's Weather and Forecasting Model (WRF) to model atmospheric gusts
- Systematic reduction in length scales to match Helios off-body Cartesian meshes
- Helios high-fidelity modeling for rotor loads, and interactional aerodynamics

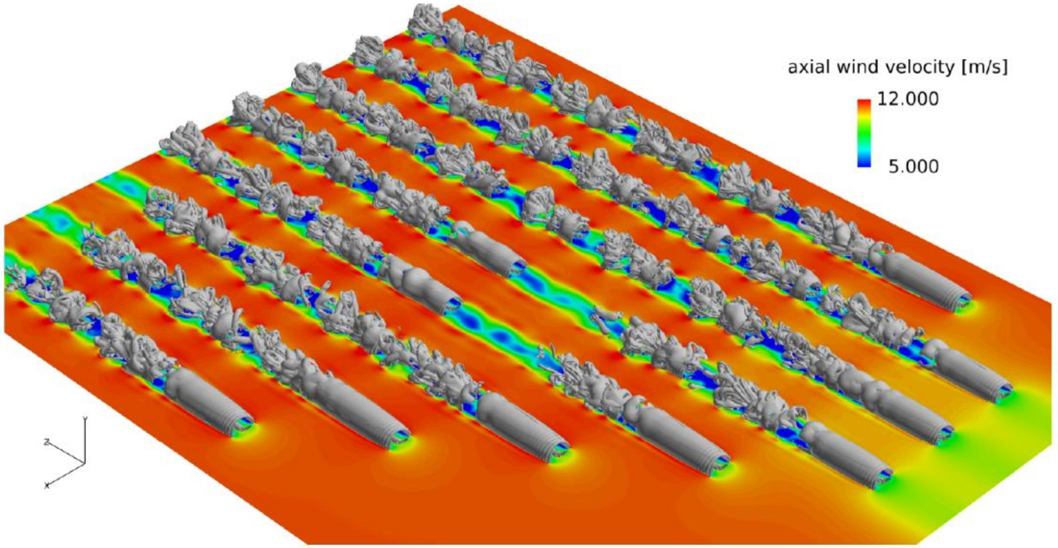


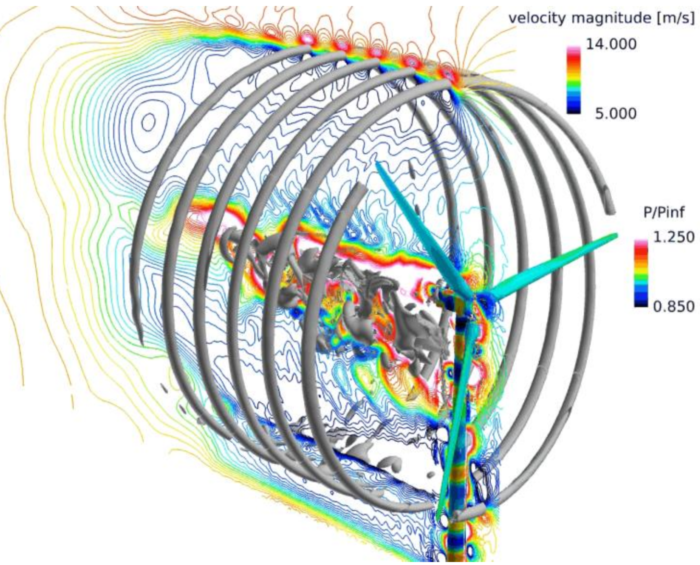


Jay Sitaraman, Univ. of Wyoming

Helios Wind Farm Simulations

Helios provides cutting-edge technology for wind turbine aeromechanics modeling

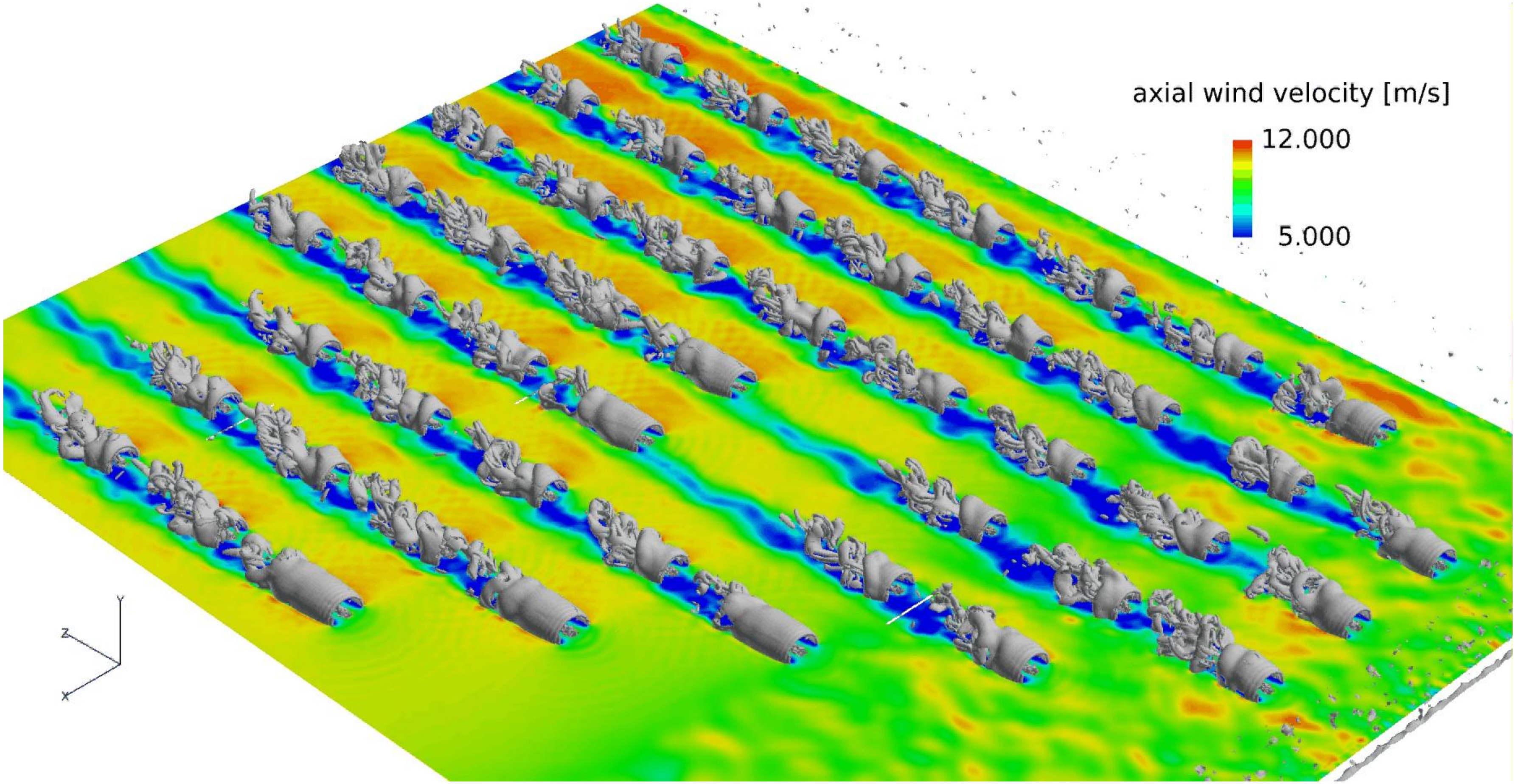




Helios wind farm simulations with high-fidelity aerodynamic modeling of 48 individual wind turbines. Future simulations will target the Chisholm View Wind Farm in Oklahoma with 140 turbines.

Currently supported by LLNL as a “Grand Challenge” application to demonstrate parallel scalability on their 1.1M processor IBM Blue Gene/Q computer

Wind Farm Simulations with Turbulent Atmospheric Inflow



Jay Sitaraman, Univ of Wyoming

Helios Development Summary

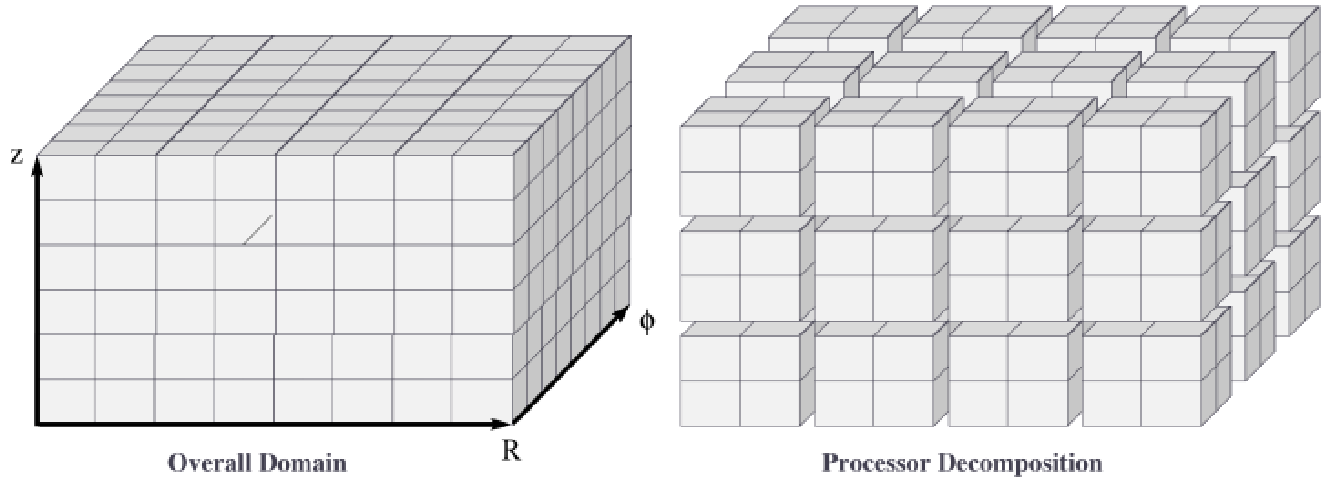
- **Helios multi-flow solver and multi-mesh paradigm facilitates analysis of complex geometry and rotor wake resolution**
- **Helios Python-based software architecture facilitates:**
  - **Interchangeable use of legacy software**
  - **Rapid introduction of new technology**
  - **Effective efforts by multiple software developers**
- **Helios software is currently being used for for high-fidelity rotary-wing aeromechanics modeling**
  - **Helios analyses targeted for Army JMR-TD and Future Vertical Lift**
  - **Interactional rotor-airframe aerodynamics**
  - **Strongly-coupled rotor dynamics**

Future Challenges for Computational Rotorcraft Aeromechanics

How do we reduce 50 hour job times to 5 minute job times?

- Need to effectively use 1000x more processors
- Fast run times are required for use of high-fidelity modeling and simulation in a multidisciplinary design and optimization (MDAO) environment

JMR JOINT MULTI-ROLE TECH DEMO

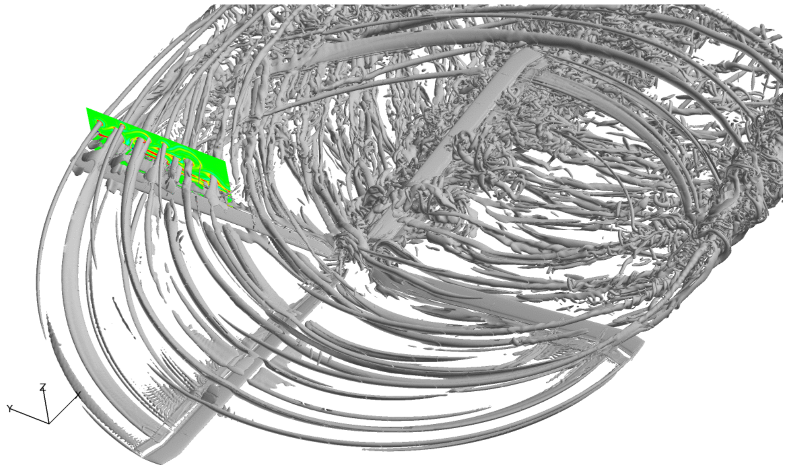


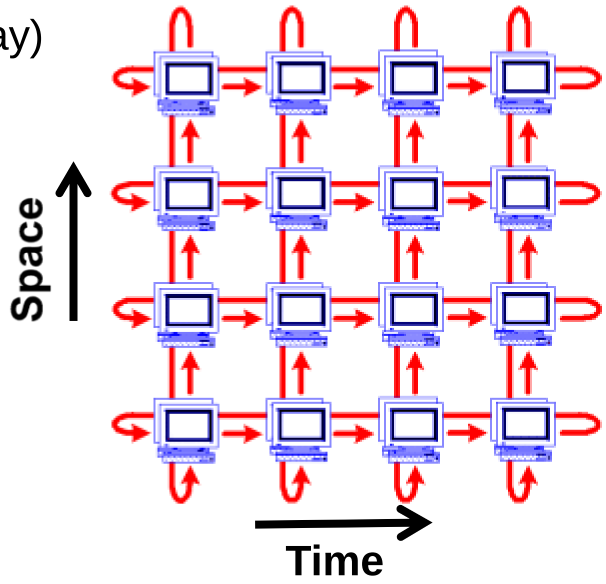
Ideal parallel scalability model

Parallelization in Time

- **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**
  - 1000 processors in space x 1000 processors in time = 1 million processors
- **How do we formulate time-parallel to work in an overset-grid environment with relative motion between grids?**

  (see Joshua Leffell's presentation on Thursday)

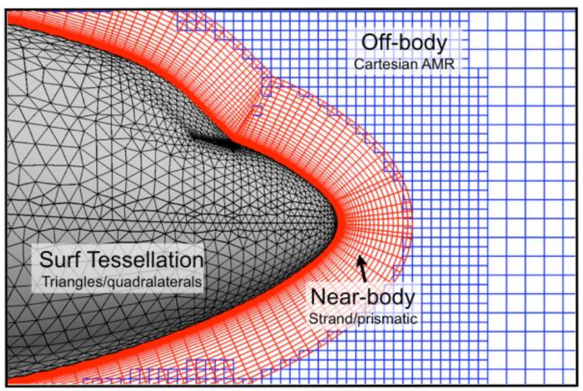




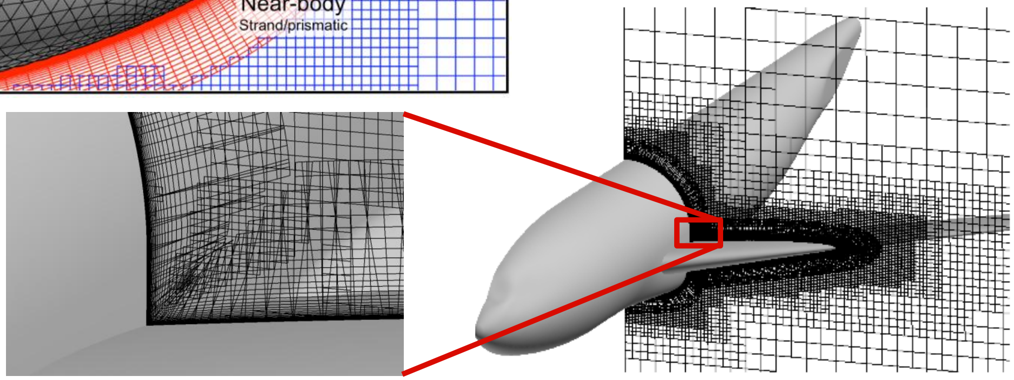
How do we automate problem setup and execution?

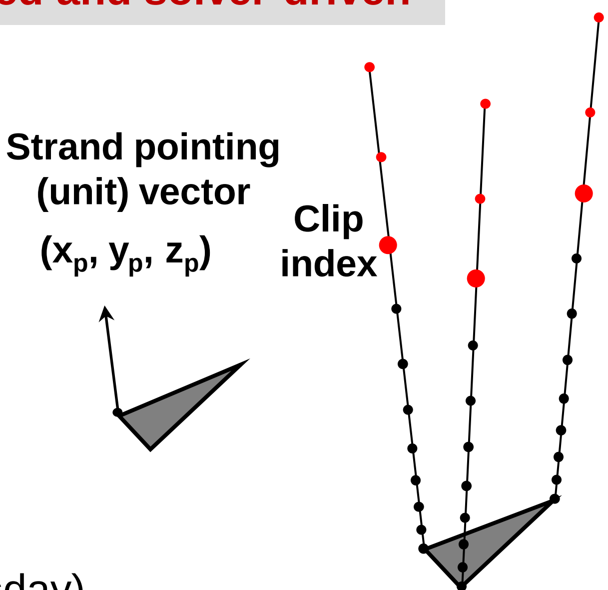
- Need to go directly from CAD to problem solution with full automation
- Need to guarantee solution accuracy
- Need to accommodate aircraft design engineers as well as CFD experts

New strand mesh solver for more automation, higher accuracy, and faster execution



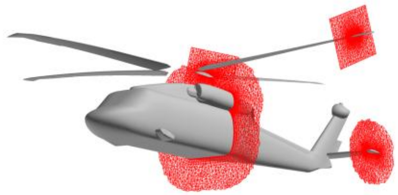
Entire mesh generation and mesh connectivity is automated and solver-driven

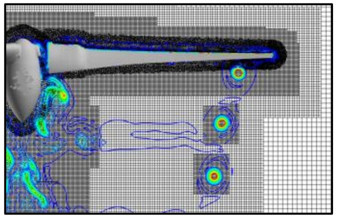


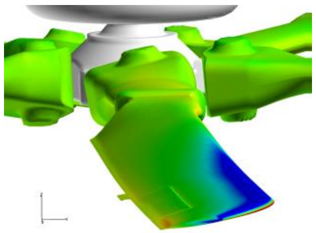


(see Andrew Wissink's presentation on Thursday)

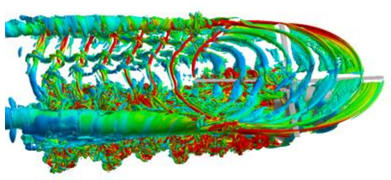
Automation for Rotorcraft Computational Modeling







- Legacy CAD
- Native CAD
- Surface mesh
- Legacy Surface Mesh
- Near-body volume mesh
- Off-body Cartesian mesh
- Domain connectivity
- Flow Solution with co-visualization
- CFD/CSD with 1-D beam or 3-D FEM
- Adaptive Mesh Refinement
- Post-Processing



– Currently under development by others (yellow)

– Currently under development by Helios team (grey)

– Currently working in Helios (green)

Future Challenges for Computational Rotorcraft Aeromechanics

How do make sure that our current and future software is maintainable and extensible to future computer hardware?

- Continue with Python framework with plug-in component modules
- Need to target future heterogeneous computer architectures

- **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 (Chine'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 Programming Languages (DSL)

Pat Hanrahan and Alex Aiken, Stanford University

Purpose

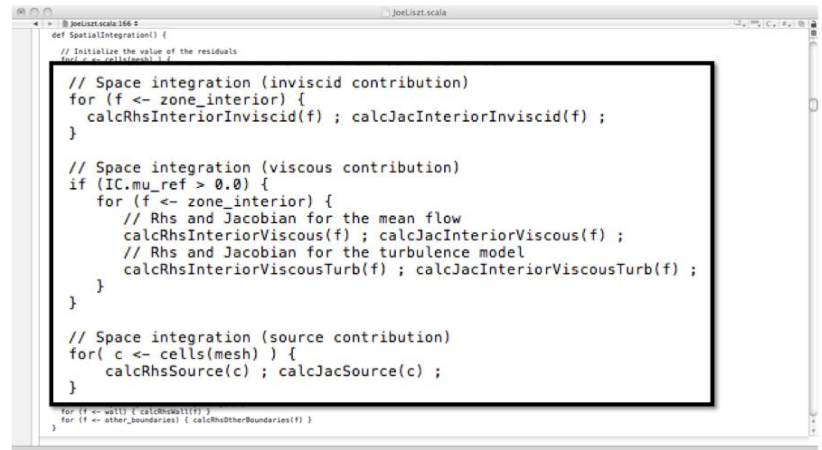
Domain-specific programming languages will automatically map parallel flow solvers to exascale computing architectures, including in-chip computations

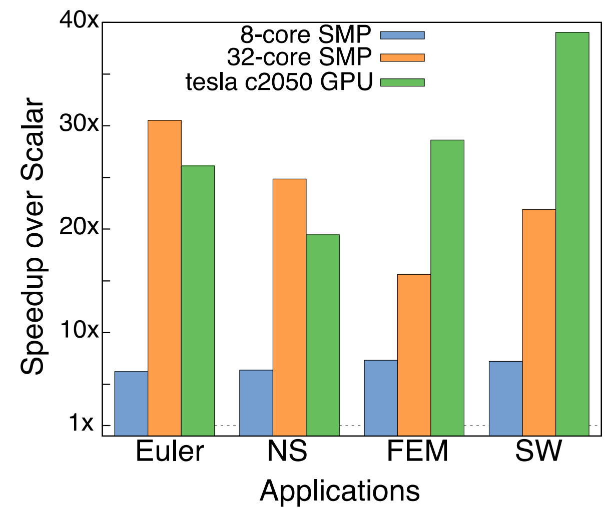
Practical example

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.

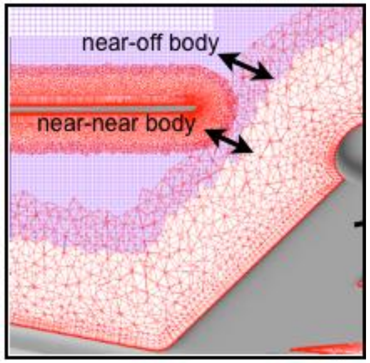




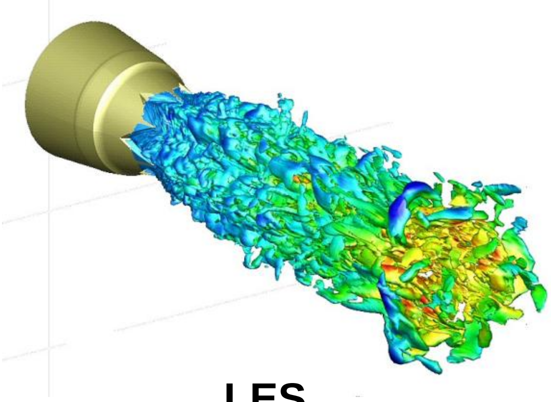
Performance of the language using different examples

How do we improve our flow solvers for higher accuracy and faster execution?

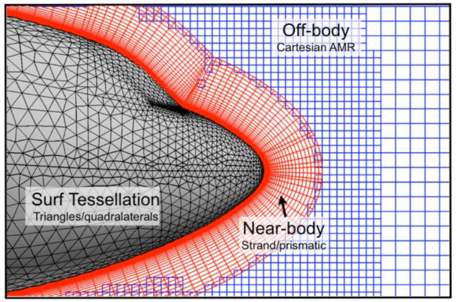
- Robust and fast high-order accurate flow solvers needed for near-body unstructured grids
- Need LES turbulence modeling for accurate modeling of difficult flow regimes such as flow separation
- Adjoint formulations are required for design optimization



Near-body unstructured



LES

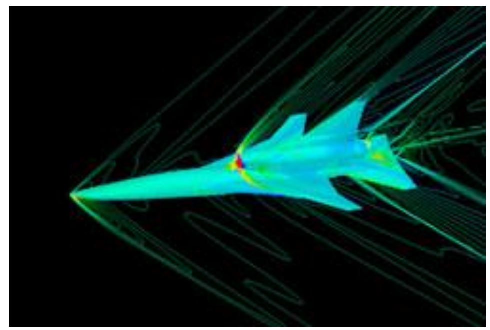


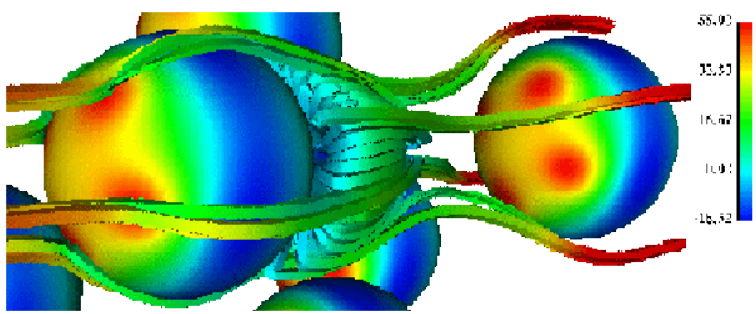
Strand mesh

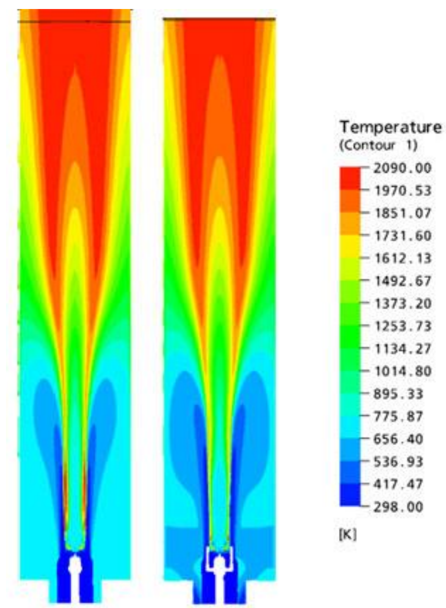
Future Challenges for Computational Rotorcraft Aeromechanics

How do we incorporate accurate and scalable multi-disciplinary physics into CFD?

- Acoustics
- Structural dynamics
- Two-phase flow
- Chemistry
- Combustion
- Particles
- ??



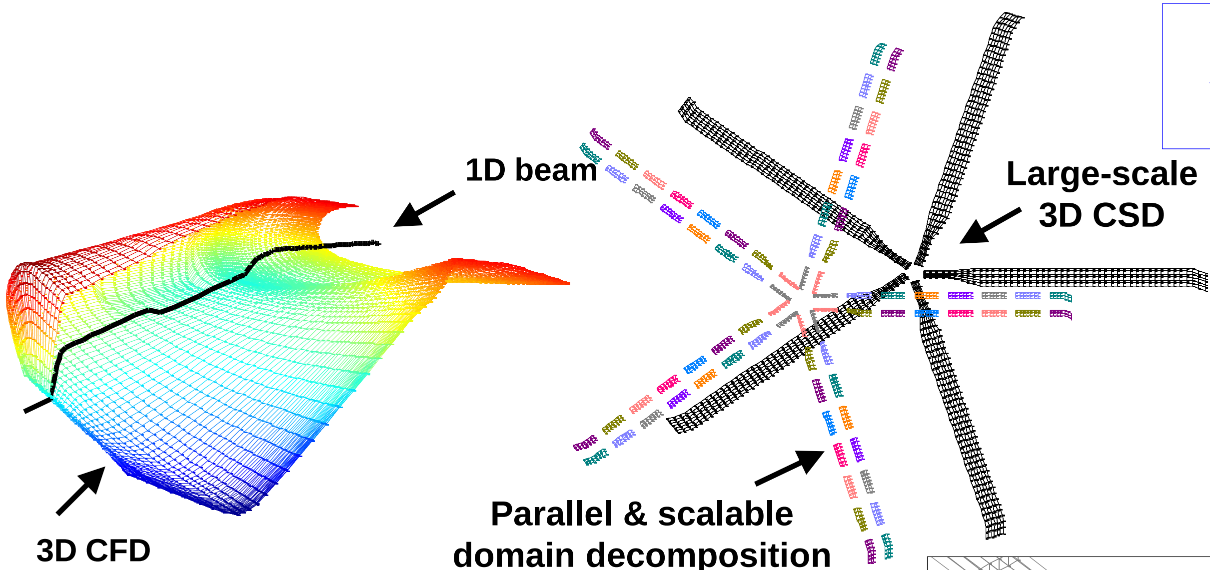


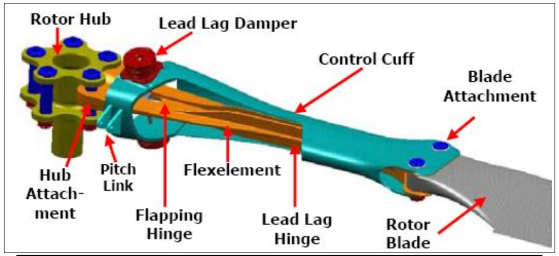


3-D Structural Dynamics for Rotor Blades

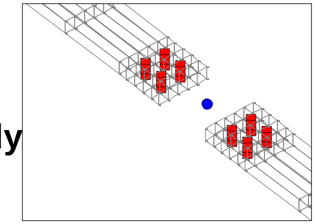
- Scalable 3-D structural dynamics solvers for accurate internal rotor stresses
- 3-D structural dynamics mesh generation directly from CAD
- Automated and scalable coupled CFD and CSD simulations

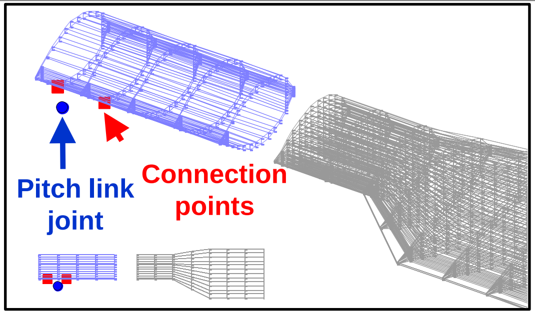
TOMORROW
3D CSD – 1,000/10,000 proc (1hr)
3D CFD – 1,000/10,000 proc (1hr)





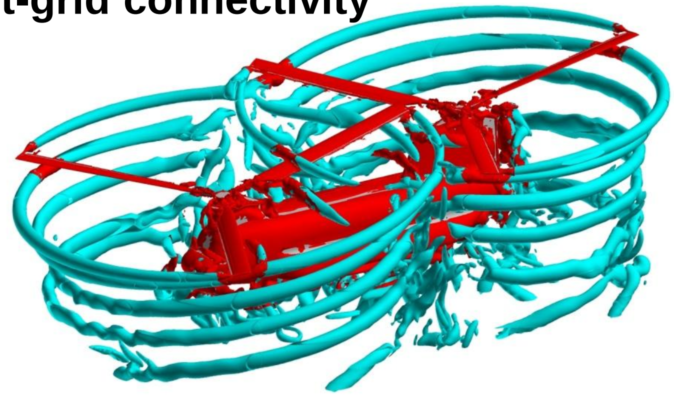
TODAY
1D Beam CSD – 1 proc (1 hr)
3D CFD – 500 proc (2 days)





Overall Summary

- **Practical CFD problems with complex geometries, relative motion, and high-fidelity modeling require overset grid solvers**
- **Current and future multidisciplinary CFD software development needs to be a collaborative process**
  - **Multiple software developers on the same project**
  - **Leveraged efforts from outside developers**
  - **Interchangeable component software modules**
- **High payoff challenge areas for future work**
  - **Fast turnaround times with parallel scalability to millions of cores**
  - **Automation for problem setup, execution, and post-processing**
  - **High-order accuracy flow solvers and overset-grid connectivity formulations**
- **High-performance computing is changing fast and our CFD flow solvers and work paradigms need to evolve in order to keep pace with new computer architectures**



Questions?

HPCMP CREATE^{TM}-AV Helios (2014)

Aviatsiya I Kosmonautika (1973)