Computational Aircraft Prototype Syntheses

Enhanced CAPS: Award FA8650-20-2-2002

Bob Haimes, PI
haimes@mit.edu

John F. Dannenhoffer, III
jfdannen@syr.edu

Marshall Galbraith
galbramc@mit.edu

David Marcum
marcum.cavs.msstate@gmail.com

Outline

- Historical Context
- Enhanced CAPS
- Statement of Work
  - Core Capability Enhancement
    - Details of the Meshing tasks
  - Ease of Use & Robustness
  - Long-Term Availability and Maintainability
  - Demonstrations
    - Details of the Internal Demonstrations
- Closing Remarks
- Prolog

Historical Context

CAPRI – active research from about 1997 through 2011

- Focused on Analysis suites
- Vendor neutral API which allows access to native geometric data
  - Geometry kernels – Parasolid, OpenCASCADE
  - CAD system – Catia V4 & V5, I-DEAS, NX, PTC, SolidWorks
- BRep and associated discrete geometry
- Could drive CAD builds parametrically
- Primarily funded by NASA Aeronautics

CAPRI – difficulties

- CAD is designed for manufacturing
- Attribution support was spotty
- Licensing (and distribution of CAPRI)
- Sensitivities!

Historical Context

ESP (2011 - 2014) – Genesis 1/2

NASA Aeronautics Cooperative Agreement #NNX11AI66A:
"The objective of this effort was to provide the tools and techniques that will allow **Geometry** to take a pivotal position within the **OpenMDAO** architecture."

Realizations

- MDO frameworks cannot deal with *rich* (BReps & mesh) data
- CAD systems usually generate a single *body* for manufacturing
- CAD systems are not setup for attribution
  complete markup is easily accomplished during the build process
- Data transfers between disciplines requires special/custom coding
- Where are CAD parametric sensitivities?

ESP – Genesis 2/2

Initial development of these Open Source projects:

- `EGADS` – Geometry kernel interface (on top of OpenCASCADE)
  - BRep and associated discrete geometry (Tessellation Object)
  - Attribution through *Solid Boolean Operators* at the *Face* level
  - Builds can be *bottom up* and/or *top down*
- `OpenCSM` – Parametric Engine built on top of `EGADS`
  - Custom *features*
  - Script driven
  - Allows for full attribution during the build
  - Simultaneous generation of multi-fidelity/multidisciplinary geometry
- `WebViewer` – JavaScript API
  - WebGL
  - WebSockets
  - C API back-end

Historical Context

CAPS (2014 - 2017)

Goals:

- Provide the tools & techniques for generalizing analysis coupling
  - multidisciplinary coupling: aeroelastic, FSI
  - multi-fidelity coupling: conceptual and preliminary design
- Provide the tools & techniques for rigorously dealing with geometry in a design framework/process

Project Objective:

- Establish a computational geometry, meshing and analysis model generation tool that can be used across AFRL RQV.
- It is envisioned that `CAPS` will enable collaboration between conceptual design, multidisciplinary optimization and high fidelity simulation efforts.

Historical Context

CAPS Milestones

- CAPS infrastructure design and initial implementation
  - C API
  - Analysis Interface & Meshing (AIM) plugin architecture
- Continued development, maintenance & support for `ESP`
  - Analytic sensitivities throughout `OpenCSM`
  - Implementation of *blend* in `EGADS` with sensitivities
  - Added many UDP/UDFs (`OpenCSM` plugins) and UDCs (macros)
  - Continued improvement of the `EGADS` tessellator
- Conservative data transfers for multidisciplinary coupling (FSI)
  - Performed by `CAPS` and not some external pairwise custom code
  - Support for general discretizations (FV, FEM, High Order)
- The Hypergeometric Shell Model (HSM)
- AIMs generated: Cart3D, ASTROS, SU^2, HSM

Historical Context

CAPS II (2017 - 2019)

Non-competitive *re-up* of CAPS with the same goals & objectives

- Geometry Tasks
  - Sculpting/B-Spline morphing
  - EGADS – evaluators, SBO improvements, parametric sensitivities
- Grid Generation – traditional bottleneck to full automation
  - Surface Quadrilaterals
  - Adaptation
  - Robust Integration of AFLR4
  - Pointwise Automation
- Parametric packaging
- User Support & Continuation Tasks
  - ESP & CAPS upgrades and maintenance
  - Trainings, documentation and software releases

Historical Context

CAPS II Milestones

- Surface Quadrilaterals – can generate full quads meshes on just about any geometry
- Adaptation – AVRO: nD ($n \leq 4$) metric driven anisotropic simplex adaptation
- AFLR4 can automatically generate quality isotropic triangular surface meshes on most `EGADS` geometry
- Pointwise can now generate unstructured meshes on `EGADS` geometry of arbitrary complexity automatically in `CAPS` through the Pointwise AIM
- AIMs distributed with `ESP`:
  AFLR2, AFLR3, AFLR4, ASTROS, AVL, AWAVE, Cart3D, `EGADS` tessellator, FRICTION, Fun3D, HSM, MASSTRAN, MYSTRAN, NASTRAN, Pointwise, Skeleton, SU^2, TetGen, TSFOIL, XFOIL

Historical Context

PAGODA – PArallel GeOmetry for Design and Analysis

- NASA Aeronautics Cooperative Agreement #NNX16AQ15A: "Develop a distributed multi-processor geometry system to fully support solver requirements (meshing, adaptation, and sensitivities) for analysis and design"
- Development of `EGADSlite`
- Back-ported to `EGADS` because of the performance gains
- Meshing Support for NASA's **Refine** and **OverGrid**
  - Robust STEP import
  - Improvement of `EGADS` tessellator
- Initiated an *Effective Topology* layer in `EGADS`
  - *Virtual Topology* with a rigorous scheme to effectively deal with *gaps* and *overlaps*
- Active through 30 September 2020

Historical Context

Trainings

Type	Date	Days	Participants
ESP	July 2015	3 1/2	41
ESP	August 2016	3	28
ESP	June 2018	2 1/2	46
CAPS	August 2018	3	23 (limited to invitees)
ESP/CAPS	June 2019	5	31
ESP	June 2020	5 – 2hr	∼145 (virtual)

Software Releases

Rev	Date
1.07	July 2015
1.08	October 2015
1.09	August 2016
1.10	September 2016
1.11	June 2017
1.12	December 2017

Rev	Date
1.13	May 2018
1.14	December 2018
1.15	May 2019
1.16	August 2019
1.17	January 2020
1.18	June 2020

Enhanced CAPS

Team

- MIT
  - Bob Haimes, PI
  - Marshall Galbraith
  - Cody Karcher – PhD Student
  - *Masters Student* – TBD
- Syracuse
  - John Dannenhoffer
  - *Masters Student* – TBD (software support)
- Mississippi State
  - David Marcum
  - Qiuhan Arnoldus
- Aurora Flight Sciences (A Boeing Company)
  - Alex Feldstein – Conceptual Aircraft Designer

Enhanced CAPS

Goal:

- Provide the tools & techniques for analysis-driven design
  - geometry centric
  - multidisciplinary & multi-fidelity coupled analyses
  - sensitivities & attribution throughout

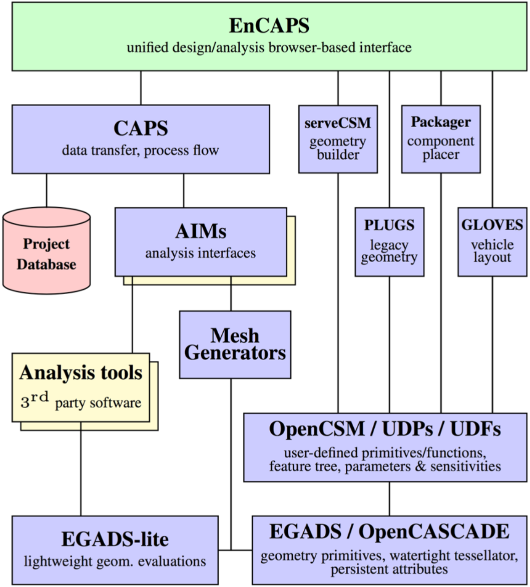
Project Objective:

- Establish an integrated computational geometry, meshing, analysis and design model generation tool that can be used across the Air Force.

Enhanced direction

`CAPS` was originally designed to run concurrently with an MDO framework. This has turned out to be rarely the method of execution. In addition there were difficulties in restarting up from where the runs left off. And finally if MDO frameworks are not *in the mix*, then additional execution support is required within the `CAPS` environment.

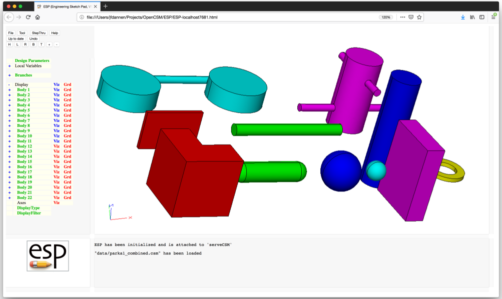
Block diagram of proposed EnCAPS system



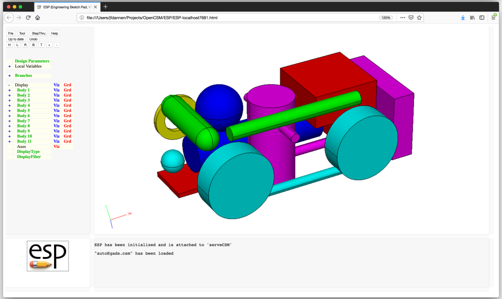
Enhanced CAPS

- Restarting runs the same script (or control program) *recycling* previous data.
- The AIMs ended up maintaining too much internal *state*, which made restarting almost impossible (requiring either rerunning or writing out the state). The AIMs will be recast to not hold so much data.
- A directory structure where the *Problem Database* contains all of the *Analysis I/O Files*.
- Full support for solver execution, which embraces asynchronous `CAPS` running when the Analysis is not run directly, but uses a batch subsystem or *mpirun*.
- More emphasis on tracking data and decisions during the session.
- Single UI (and integrated editor) for Geometry and Analysis.

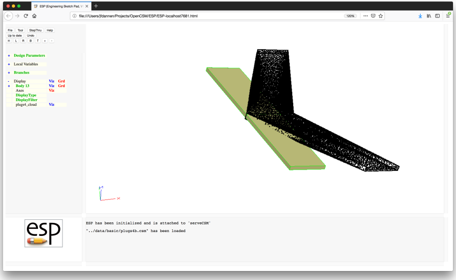
EnCAPS – Component Placement



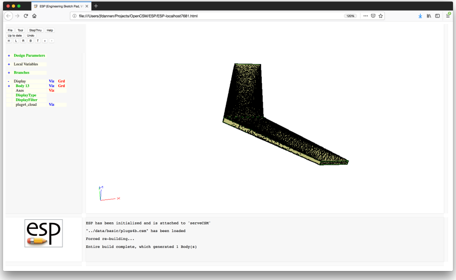
Input suite – minimal box



Voxel-based best packing



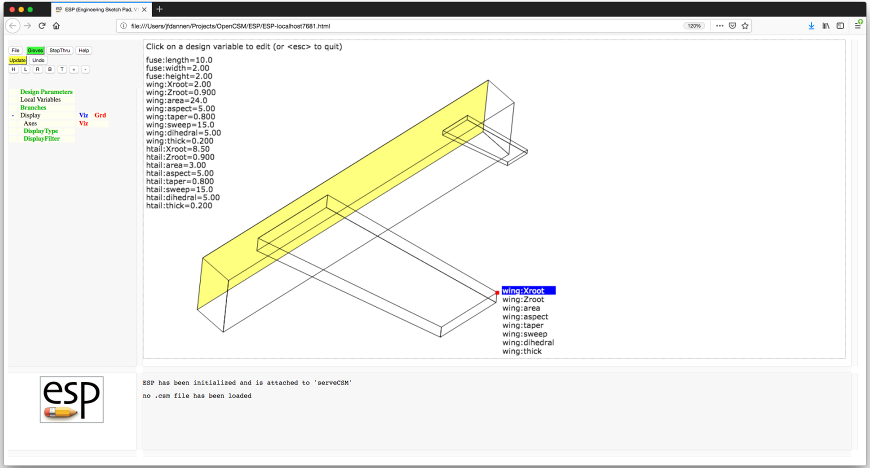
Cloud of points
& initial parametric guess



Parametric fit

Parametric Legacy Unstructured Geometry System

EnCAPS – GLOVES



Graphical Layout Of Aircraft SystemS

EnCAPS – Statement of Work

Core Capability Enhancement 1/2

- Geometric modeling
  - *Flends* – A combination of fillets and blends
  - B-spline editing
  - Mating & Mechanisms
- Parallel distributed computing
  - Geometry build parallelization (building upon PAGODA)
  - `EGADSlite` – GPU port (building upon PAGODA)
  - Parallel AFLR3 meshing
- Vehicle subsystem packaging and layout
  - Collision detection
  - Optimization
  - IML/OML parametric variations

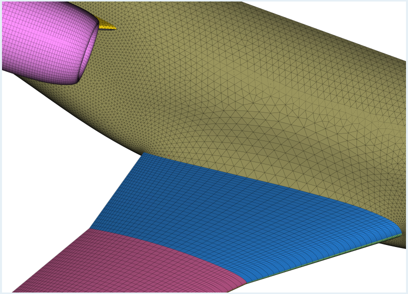
EnCAPS – Statement of Work

Core Capability Enhancement 2/2

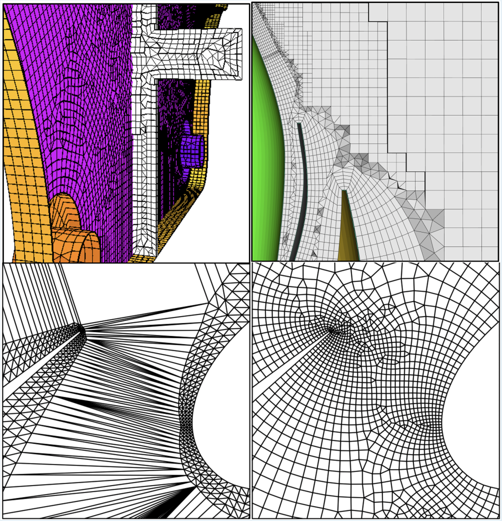
- Improve inter-analysis data transfer for multidisciplinary workflows
- Combine geometric sensitivities with analysis sensitivities to guide gradient-based optimizations
  - Cart3D Design Framework
  - Fun3D Design Framework
- Support and maintenance
- Meshing
  - Anisotropic surface meshing
  - High-order surface meshing
  - Alternative mesh element types
  - Meshing for structural analysis – surface quadding

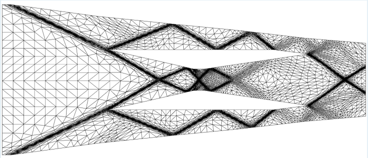
Anisotropic Meshing & Mixed Element Types 1/2

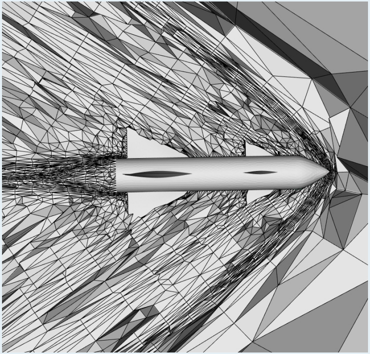
- AFLR 2D planar and 3D volume mesh generators have the capability to use metric fields to create anisotropic meshes through the uses of a novel metric-aligned point-placement strategy that produces mesh elements that align with the metric fields.
- They also have the capability to generate right-angle elements by using an alternative point-placement strategy that allows the formation of alternative element type meshes. The 3D capability is, at present, limited to anisotropic boundary layer and isotropic pseudo-structured regions.
- These capabilities will be implemented within AFLR surface meshing using a curvature-driven metric for anisotropic mixed quad/tria-face surface meshes.



Anisotropic Meshing & Mixed Element Types 2/2







Parallelization

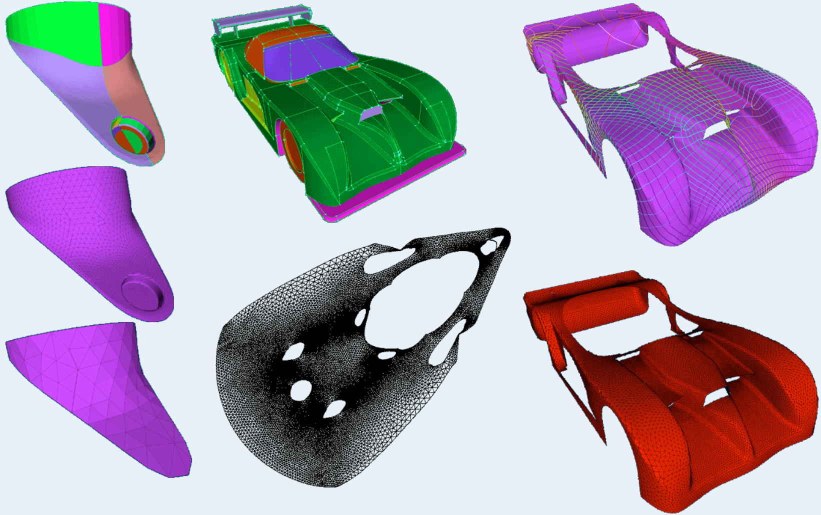
- AFLR surface meshing can be parallelized/threaded in part at the surface patch level relatively easily. Additional areas of parallelization within the initial setup, background mesh generation, and proximity testing will be investigated.
- Parallelization within AFLR volume meshing has been in process and the resulting capability will be available for `CAPS`. For the present work, a moderate level of integration work is proposed to make the AFLR3 parallel process automated and easy to utilize for `CAPS` users.

Other Possible AFLR Enhancements

- AFLR3 *Attributes*: Full integration of AFLR volume meshing parameters within `CAPS` via Body and Face Attributes.
- *Virtual Faces*: AFLR surface meshing has a capability built-in for *Virtual Faces*. Each *Virtual Face* is composed of multiple traditional Faces. The *Virtual Face* is a subset of *Effective Topology* which is essentially a global interpolation map of the underlying Faces. A global $[u, v]$ generator within AFLR provides a topological means for evaluations to be made on the Virtual Face without any modification of the underlying Faces.
  - Eliminates any surface mesh artifacts due to Face layout in the `EGADS` Body
  - Provides a means to eliminate small (sliver) geometric entities
  - Already used by `EGADS` *Effective Topology*

EnCAPS – Meshing

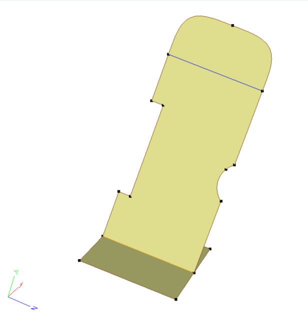
Multiple Faces as a single entity



EnCAPS – Effective Topology

Virtual Topology & BRep closure

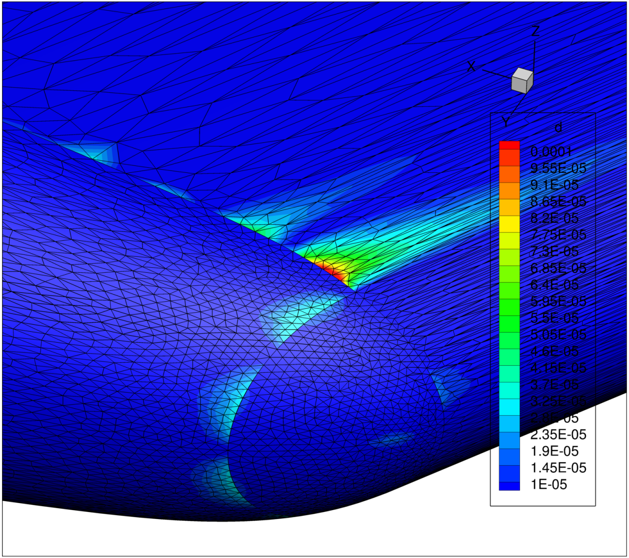
- BRep Topology inhibits *quality* meshes
  - Spurious Nodes
  - Small Edges that could be coalesced
  - Sliver Faces
- EGADS' *Effective Topology*
  - Automatically removes spurious Nodes
  - Automatically coalesces Edges
  - Collects Faces explicitly or by attribute uses *EGADS* tesellation & global $[u, v]$
  - Adjusts *Effective Face & Edge* evaluations based on closure



- EGADS has been updated to generate fewer spurious Nodes
- Probably used in AIMs driven by Face attributes

EnCAPS – Effective Topology

Surface displacement distributed by a Laplacian (Mike Park)



Ease of Use and Robustness 1/2

- Enhanced CAPS software design review
- Develop an AFRL software access point
  - AFRL internal distribution environment
  - CAPS testing process (*Jenkins* clone)
- Documentation
  - Written
  - Video (tutorials and trainings)
- Reduce the time required to evaluate vehicle configurations **move towards an integrated analysis driven design system**
  - PLUGS
  - Integrated GUI using ESP
  - Surface related outputs to GUI
  - Line plots of tables (capsValue Objects)

EnCAPS – Statement of Work

Ease of Use and Robustness 2/2

- Improve the speed and robustness of geometry operations
  - Geometry construction speed
  - Robustness (`EGADS` internal improvements)
  - Provide a set of *best practices* to avoid known problems (i.e., coincidence)
  - Debugging infrastructure for geometry creation
- Treatment of errors
  - Better AIM error reporting
  - Improved `CAPS` error reporting and handling
  - Provide the pyCAPS user with better diagnostics

EnCAPS – Statement of Work

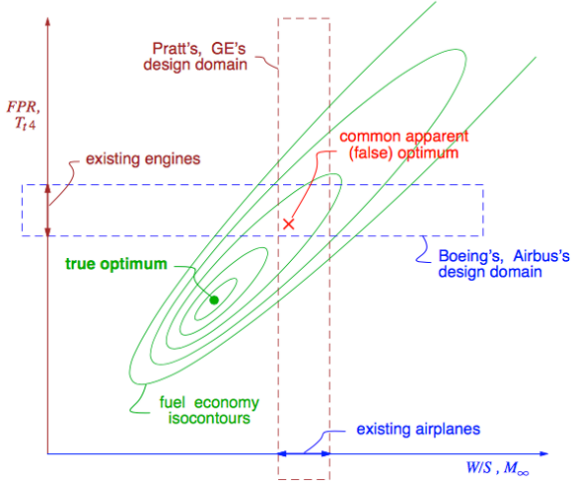
Long-Term Availability and Maintainability

- Bug & Feature tracking system
- Develop a plan for long-term availability and maintainability (beyond EnCAPS)
  - The migration plan
  - Pilot project for the plan

Demonstrations

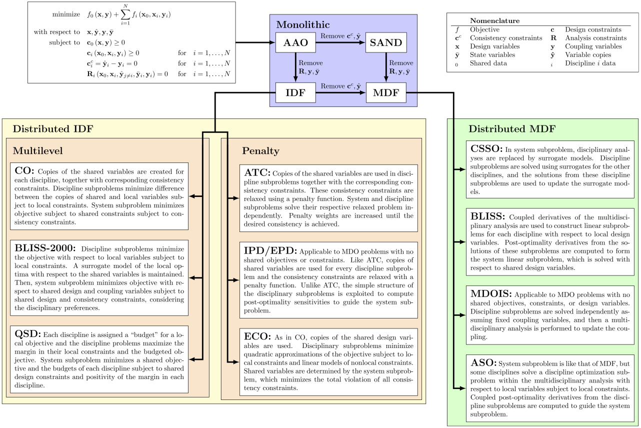
- Internal design cases
- Aurora tool chain
- Execute a design of interest to Aurora
- Use of GLOVES
- Design process tracking

Lack of communication between analysis modules
leads to poor designs ...



Drela, 2010

... and classical MDAO frameworks and architectures have done little to solve the problem in the past 30 years

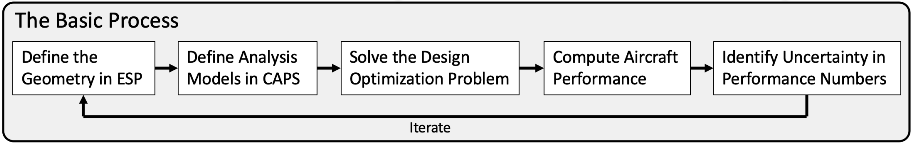


Martins, 2013

EnCAPS – Internal Demonstrations

EnCAPS Design Optimization

- Conclusion: Optimization must be tightly integrated into the design process
- Proposed Goals
  - Define aircraft design spaces using `ESP` UDCs (or similar)
  - Create optimization friendly analysis modules in `CAPS`
  - Incorporate gradient based and heuristic optimization schemes
  - Advance log-convex optimization to non-explicit formulations
  - Enable multi-fidelity analysis in the optimization process
  - Develop methods of uncertainty propagation



EnCAPS – Internal Demonstrations

Subsonic Transport

- USAF operates nearly 1000 subsonic transport and tanker aircraft, therefore performance gains in this area correspond to significant cost savings
- Case study facilitates the development of a wide range of discipline models that span multiple fidelities
- Future concepts rely on tight coupling between disciplines, which provides a challenging design problem
- Extensive literature for verification and validation

EnCAPS – Internal Demonstrations

Electric Vertical Takeoff and Landing

- Electric aircraft are widely utilized in the field (UAS) and are the future of many realms of aviation
- Case study facilitates the development of the models for electrical powertrains and VTOL aerodynamics
- Poses a very different design problem due to high uncertainty and low technological maturity – will require a different perspective than the subsonic transport problem
- Growing body of literature for verification and validation

Concluding Remarks

Questions/Directions

- What does a *Design through Analysis* system look like?
- Controlling execution
  - Local vs. off workstation
  - What execution paradigms should `CAPS` support? *mpirun*, what queuing (batch) systems?
  - Option for complete data driven execution?
- Integrated Development Environment (IDE) like? Handling of errors with integrated source editor
- What are the best methods to coexist with Python?
- *Digital Twin/Digital Thread* integration? Under whose charter at AFRL?

Concluding Remarks

A Cooperative Agreement

- Both CAPS & CAPS II were *Contracts*, but the interaction with AFRL was as if we had a *Cooperative Agreement*
- We have received a list of *desirements* from various groups at AFRL – most will be handled over the course of EnCAPS
- Our intention is to continue the close collaboration
  - Frequent AFRL visits & COVID-19?
    - Need to find an alternative mode for communicating with the ARFL `CAPS` user base
  - Allows us to generate targeted and more useful software
  - Great satisfaction seeing the fruits of our labors being used daily

Thank you!

Prolog

The Next CAPS Training

Following the format used during the June 2020 ESP training:

- Blackboard, which only requires a web browser (no plugins) and can support hundreds of attendees
- 5 sessions of 2 hours each (10:00am – noon Eastern):
  9, 11, 14, 16 & 18 September 2020
- Exercises done as homework
- Uses ESP Rev 1.18 (June 2020)
- If you are interested send an e-mail to haimes@mit.edu