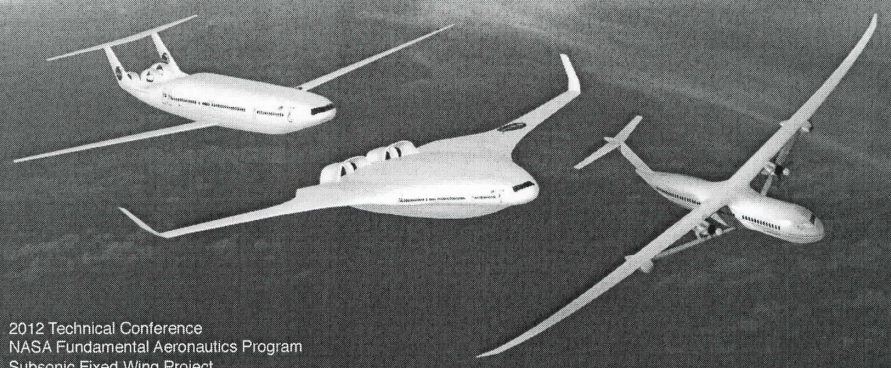
National Aeronautics and Space Administration

Modeling Efforts with the OpenMDAO Framework

Mr. Justin S. Gray

Aerospace Engineer
NASA Glenn Research Center



2012 Technical Conference
NASA Fundamental Aeronautics Program
Subsonic Fixed Wing Project
Cleveland, OH, March 13-15, 2012

www.nasa.gov

OpenMDAO Background

OpenMDAO.org

- Open source Multidisciplinary Design Analysis and Optimization (MDAO) Framework
- Built using the Python programming language
- Distributed under the Apache V2.0 open source license
- A research effort established with the goal of providing a common platform for MDAO that will help foster collaboration between industry, academia, and government

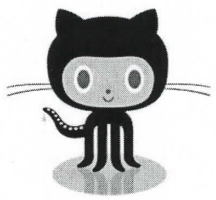
- Website: http://openmdao.org
- Source Code Repository: http://github.com/openmdao/openmdao-framework

Motivation

- Provide a common platform to expand the state of the art for MDAO through collaboration between industry, academia, and government
- Develop an MDAO framework that can support advanced MDAO algorithms to enable high-fidelity optimizations at all parts of the design process
- Encourage greater code re-use and software sharing in the MDAO field through the use of open source software development and distribution methods.

Open Source Software Community

- Released under the Apache V2.0 License
  - Extremely Permissive license which allows you to do almost anything
  - http://www.apache.org/licenses/LICENSE-2.0.html
  - Proprietary code can interface with OpenMDAO, without needing to be open source itself
  - Apache is compatible with the majority of other open source licenses out there
- Using Github as the portal for all community code contributions
  - Other software hosted on github.com: EGADS, GEM, GeoMACH, OpenMDAO official plugins
- Community forums: http://openmdao.org/forum
  - Active and growing user groups
  - Questions and answer style which encourages discussion



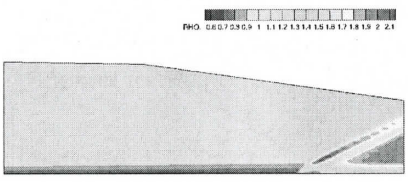
github
SOCIAL CODING

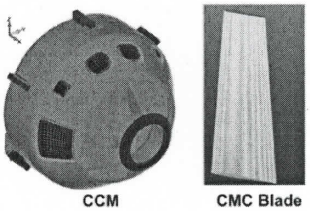
New Features in the Framework

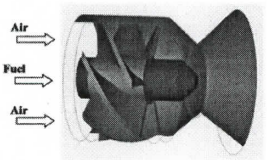
- OpenMDAO v 0.2.3 released, April 2012
- New Optimizers: SLSQP, COBYLA, PyOpt, IPopt
- Support for components with analytic derivatives
- Automatic implementations of MDAO Architectures:
  - IDF, MDF, CO, BLISS,BLISS-2000
- Greater support for High Performance Computing
- Easy installation of OpenMDAO plugins from the github plugins repository: http://github.com/openmdao-plugins
- Support for Python 2.7

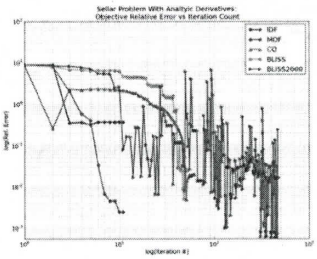
Modeling Efforts with OpenMDAO

- Working with researchers from Subsonic Fixed Wing and Supersonics to enable new research with MDAO methods



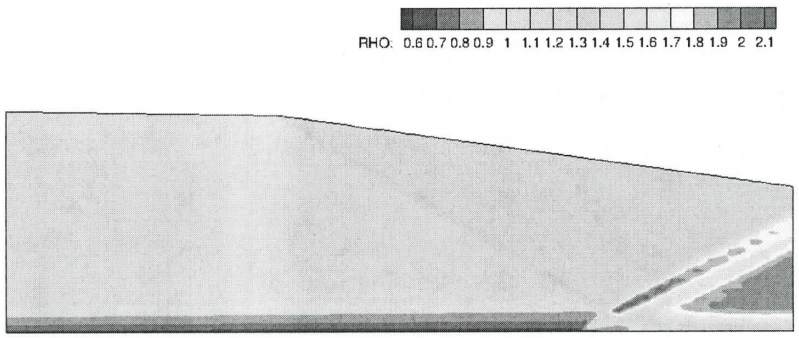






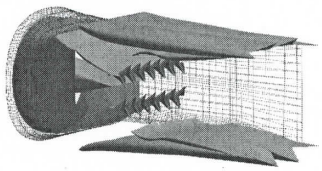
Modeling Efforts: Supersonic Inlet

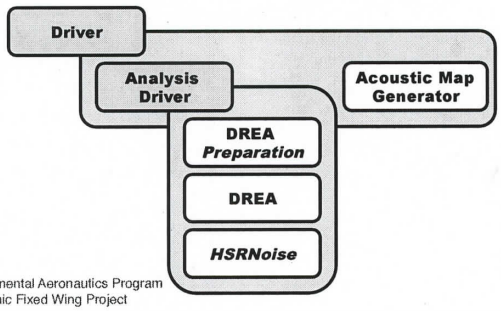
- Supersonic Inlet Geometry Optimization with Overflow
  - Wrapper for Overflow V2.2 available on http://github.com/openmdao-plugins
- Model built to investigate the optimal location and size for bleed holes in the inlet flow path
- Execution run remotely via High Performance Compute Cluster



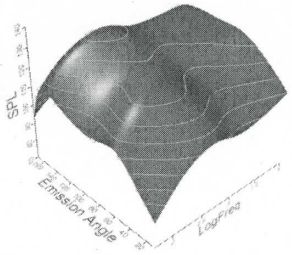
Modeling Efforts: Mixer-Ejector Nozzle

- Goal: develop a multidisciplinary analysis providing performance and acoustic maps for system level analysis
- Low fidelity analysis tools
  - Mixer-ejector performance: DREA
  - Mixer-ejector acoustics: HSRNoise
- Multidisciplinary analysis capability has been demonstrated for a notional design

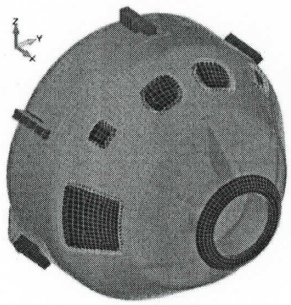




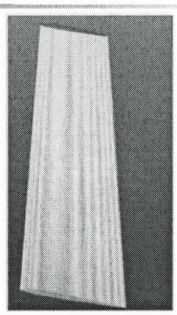
SPL Values for the Flyover Observer



Modeling Efforts: Structural Optimization



CCM

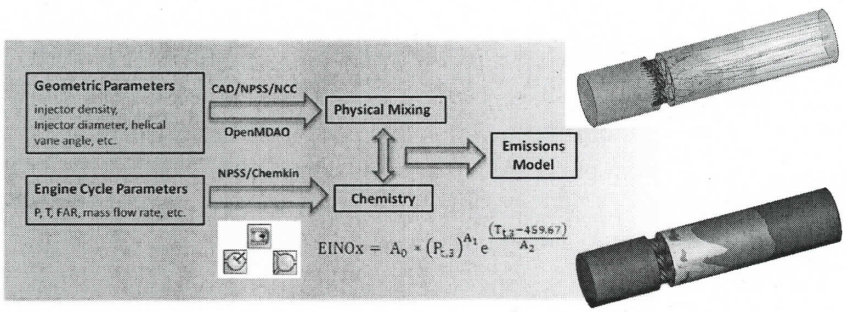


CMC Blade

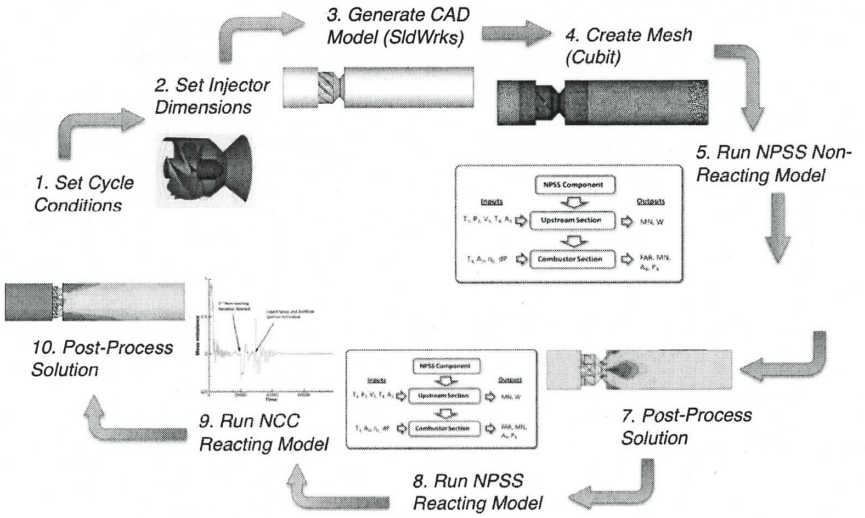
- Wide Range of Different Finite Element Models: Composite Crew Module, 25 Bar Truss, Turbine Blades
- Deterministic and stochastic structural optimizations
- Experimenting with optimization strategies using multiple optimizers
- Working with multi-objective optimizations

Modeling Efforts: Lean Direct Injection

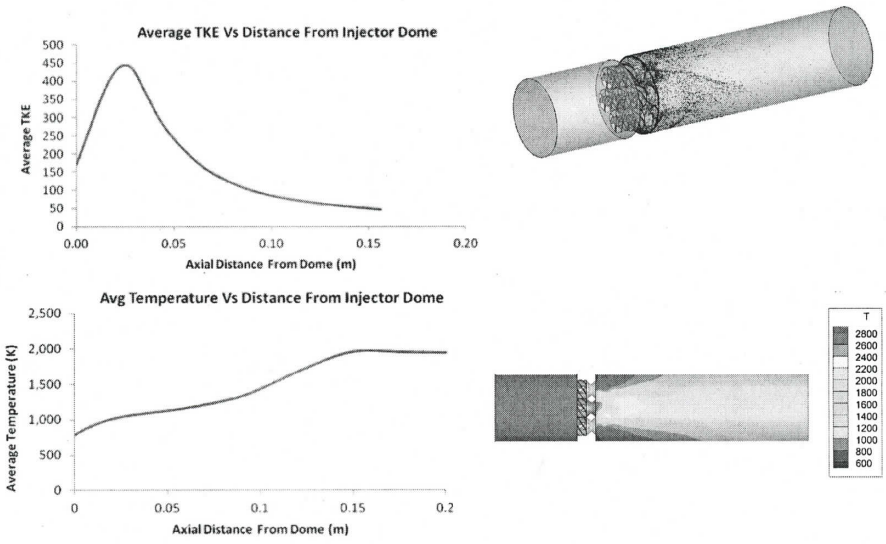
- Multi-fidelity modeling effort, using NPSS and NCC
- Tight integration with Solidworks parametric geometry
- Design space includes major topological changes to geometry



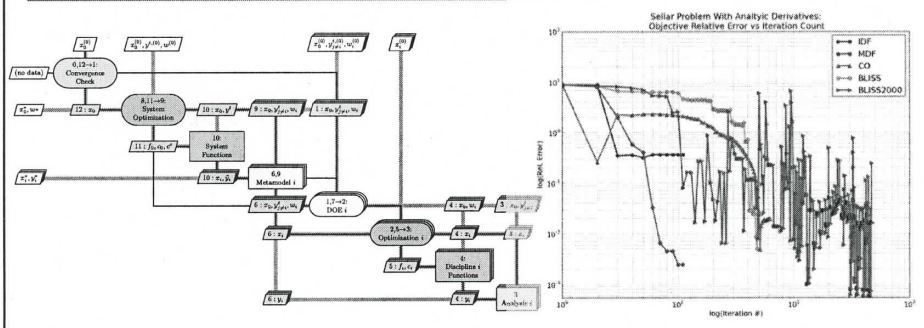
Modeling Efforts: LDI cont.



Modeling Efforts: LDI cont.



Modeling Efforts: MDAO Benchmarking



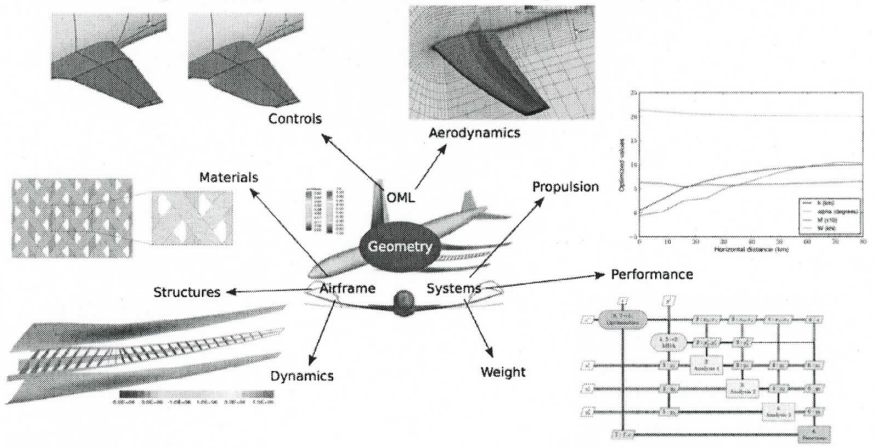
- Using OpenMDAO for standardized MDAO architecture testing
- Implemented IDF, MDF, CO, BLISS, BLISS-2000
- Automatically apply MDAO architectures to any problem
- Building a suite of test problems to test architectures against
- Working with AIAA MDO Technical Committee to expand test suite.

Integrating Geometry Capabilities

- Developing a tight integration with multiple geometry tools via a common interface
- NRA Effort: University of Michigan, MIT:
  - GeoMACH: open source conceptual geometry engine suitable for optimization with high fidelity analysis tools
- NRA Effort: MIT, Syracuse University:
  - EGADS: Utility for interfacing with OpenCASCADE kernel
  - OpenCSM: open source CAD based approach to geometry for MDAO
  - Geometry Engine for MDAO (GEM): universal interface for working with geometry tools

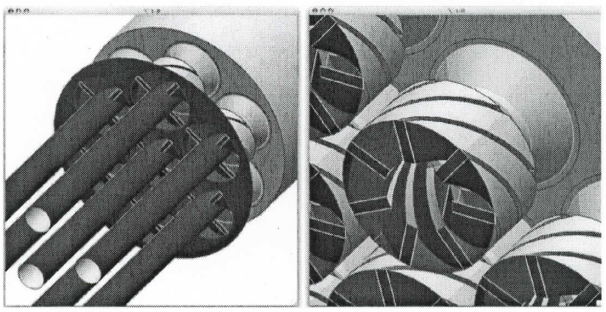
GeoMACH

- PI: Dr. Martins, University of Michigan
- Fast b-spline based geometry generation with analytic derivatives
- Produces geometry suitable for use with high fidelity optimizations



OpenCSM and EGADS

- PI: Robert Haimes, MIT; CO-PI: Dr. Dannenhoffer, Syracuse University
- CAD based approach to geometry generation
- Support for Analytic Derivatives
- EGADS provides services to access the OpenCASCADE Kernel
  - OpenCSM uses EGADS to talk to OpenCASCADE geometry Kernel
  - OpenVSP project is using EGADS for STEP file output capability
  - GeoMach links with EGADS to provide an efficient link to the GEM API



Geometry Environment for MDAO (GEM)