Overview of the Higher-Order Design Environment (HOrDE)

Erik D. Olson
Aeronautics Systems Analysis Branch
NASA-Langley Research Center

Sept. 17, 2019
Vision: Enabling higher-order analysis of conceptual-level, conventional and unconventional aircraft concepts through design refinement.

• Motivation
  – Low-order analysis limits ability to accurately model unconventional configurations
  – Meaningful high-order analysis is limited to fully-developed designs
  – Lacking a capability for creating “higher order geometry” to complement higher order analysis

• General Approach
  – Leverage existing MDAO frameworks (OpenMDAO, ModelCenter)
  – Exploit Vehicle Sketch Pad (OpenVSP) as a common geometry interface
  – Develop “higher-order geometry” through automation and intelligent streamlining in a multi-fidelity design process
  – Build new design tools that integrate existing analysis methods tailored to early design stages
**HOrDE Software Organization**

**Introduction**

HOrDELib Features

Wrapper Interfaces

Process Models

Documentation & Release

---

**HOrDELib Features**

- Data types (scalars, arrays, files, enumerations)
- Generic wrapper functionality (input variables, command execution, output file parsing)
- Automated data storage/retrieval and multi-fidelity mapping
- Units library

---

**Wrapper Interfaces**

- Phoenix wrapper interface
- OpenVSP, AVL, DATCOM, Xfoil, MSES, ASWing, etc.

---

**Process Models**

- Initial Transonic Wing Design
- High-Lift Systems Design
- Low-Order Aero-Structural Analysis

---

**HOrDELib Classes**

**ModelCenter**

- Initial Transonic Wing Design
- High-Lift Systems Design
- Low-Order Aero-Structural Analysis

**Python / OpenMDAO**

- Initial Transonic Wing Design
- High-Lift Systems Design
- Low-Order Aero-Structural Analysis

---

**Python bindings (py4j)**

---

**HOrDELib API**

- Data types (scalars, arrays, files, enumerations)
- Generic wrapper functionality (input variables, command execution, output file parsing)
- Automated data storage/retrieval and multi-fidelity mapping
- Units library
HOrDELib Features

Introduction
HOrDELib Features
Wrapper Interfaces
Process Models
Documentation & Release
Degenerate Geometry

Introduction

HOrDELib Features

Wrapper Interfaces

Process Models

Documentation & Release

Transformational Tools and Technologies Project

erik.d.olson@nasa.gov
Introduction

HOrDELib Features

Wrapper Interfaces

Process Models

Documentation & Release

Geometry Representation

- Vehicle
  - Surface
  - GeometryComponent
    - ControlGroup
    - PropulsionGroup
      - QuadMesh
      - TriMesh
      - DegenGeom
        - ControlSurface
        - Propulsor
          - DegenSurface
            - DegenPlate
            - DegenStick
            - DegenPoint
          - Blank
          - SubSurface
Multi-disciplinary, Multi-fidelity Mapping Process

Introduction

HOrDELib Features

Wrapper Interfaces

Process Models

Documentation & Release

WRITER: Erik D. Olson
PROJECT: Transformational Tools and Technologies Project
EMAIL: erik.d.olson@nasa.gov

HOrDELib Features

Wrapper Interfaces

Process Models

Documentation & Release
Other Features

Introduction

HOrDELlib Features

Wrapper Interfaces

Process Models

Documentation & Release

- **Data Catalog**
  - Hierarchical library of data keys (ex: Aero_2D.Drag_Polar.cd)
  - Compile-time data key lookup
  - Compile-time check on unit compatibility

- **Units Framework**
  - Implementation of Units of Measurement API v2.0
  - Units conversion
  - Convert units derived through combination of base units (e.g. Newton)
  - Wrappers accept inputs for length, mass, temperature, etc. units and maintain unit consistency of calculations throughout
  - Analysis data can be stored in one unit and retrieved in a different one.
Python Bindings

Introduction

HOrDELib Features

Wrapper Interfaces

Process Models

Documentation & Release

- **HOrDEGateway**
  - Dynamically access Java classes from Python using Py4j
  - Starts a Gateway Server prior to running Python scripts

- **HOrDEWrapper**
  - Lightweight Python-side interface to an individual Java wrapper
  - Get and set variable and array values as Python variables and lists
  - Run the Java wrapper

- **HOrDEModel**
  - Build a process model using one or more instances of HOrDEWrapper
  - Link inputs and outputs between wrappers
  - Scripted logic, iteration, intermediate calculations
  - Can be set up to work with OpenMDAO

- **HOrDEServer**
  - Initiated by the HOrDEModel at runtime
  - Provides interface between the HOrDEModel and the HOrDEGateway
Local and Remote Computing in Python

Introduction

HOrDELib Features

Wrapper Interfaces

Process Models

Documentation & Release

Local Machine

Model

Wrapper

Wrapper

Server

Remote Machine

Server

HOrDELib

Java Wrapper

Py4j Gateway

Pyro

HOrDELib

Java Wrapper

Py4j Gateway
ModelCenter Interface

- Lightweight client-side wrapper files provide access to the Java wrapper
- ScriptWrapper Interface
  - Wrapper object passed to interface
  - Input and output variable values replicated using native classes on the server side
- Support for dynamic output variables
Introduction

HOrDELib Features

Wrapper Interfaces

**Process Models**

Documentation & Release
Library of Wrapped Codes

Introduction
HOrDELib Features
Wrapper Interfaces
Process Models
Documentation & Release

• Geometry Definition
  – OpenVSP
• Aerodynamic Analysis and Design
  – FRICTION – Zero-lift drag estimation
  – AVL – Vortex-lattice aerodynamic analysis
  – XFOIL – Two-dimensional panel code with coupled boundary layer
  – MSES – Two-dimensional Euler CFD with coupled boundary layer
• Aerostructural Analysis
  – ASWing – lifting-line aerodynamics and equivalent-beam structures
### Utility Methods

**Introduction**

**HOrDELib Features**

**Wrapper Interfaces**

**Process Models**

**Documentation & Release**

- **VSP2Geom** – Create native geometry objects from OpenVSP export files
- **DATCOM Airfoil** – Empirical estimation of section lift and drag
- **Empirical High-Lift Performance** – Empirical estimation of high-lift system lift, drag, and moment
- **Target Spanwise Lift Distribution** – Generate target lift distributions for twist optimization
- **Sonic-Plateau Pressure Distribution** – Define target airfoil pressure-distribution for inverse design
- **BSpline Airfoil** – Fit fourth-order B-spline to smooth airfoil surface
- **Kulfan Airfoils** – Fit airfoils using Class-Shape Transformation
- **Tecplot Data File Generation** – Automatically generate Tecplot files from DegenGeom objects
Process Models Library

Introduction

HOrDELib Features

Wrapper Interfaces

Process Models

Documentation & Release

Initial Transonic Wing Design

Low-Order AeroStructural Analysis

High-Lift Geometry Definition

Low-Order High-Lift Analysis
Introduction

HOrDELib Features

Wrapper Interfaces

Process Models

Documentation & Release
Introduction

HOrDELib Features

Wrapper Interfaces

Process Models

Documentation & Release

- Documentation
  - Java API documented using Javadoc
  - Python bindings documented using Sphinx
  - HOrDELib User’s Guide
  - HOrDE Process Models Guide

- Release Status
  - Version 1.0 available from the NASA Software Catalog (https://software.nasa.gov/software/LAR-19572-1)
  - Approved for U.S. and foreign release
Related Publications

Introduction

HOreDELib Features

Wrapper Interfaces

Process Models

Documentation & Release

- Olson: Multi-Disciplinary, Multi-Fidelity Discrete Data Transfer Using Degenerate Geometry Forms, AIAA 2016-3208.
- Olson: Three-Dimensional Piecewise-Continuous Class-Shape Transformation of Wings, AIAA 2015-3238.
• This work was conducted as part of the NASA Transformational Tools and Technologies Project, led by Dr. Michael Rogers, within the Multi-Disciplinary Design, Analysis and Optimization element, led by Patricia Glaab.