OpenMDAO Development and Usage

What's New in OpenMDAO

Kenneth T. Moore

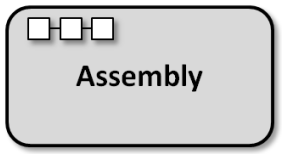
July 19th, 2012

Introduction to OpenMDAO

OpenMDAO Mission: To develop an open platform for engineering design that facilitates the use of advanced MDAO algorithms with multiple discipline analyses.

1. Exchanging information between analysis tools
2. Distributing analyses across multiple computational resources
3. Using surrogate modeling tools as part of an analysis
4. Implementing complex optimization processes around an analysis

Component – a computational engineering application that accepts input and returns output



Assembly – a container object which houses sub-components, defining data connections between them

Driver – an object that controls process iteration (solvers, optimizers, iterators, etc.)

Workflow – an object that specifies execution order of components for a driver

Plugin Interface

- Plugin: An external add-on to OpenMDAO
  - May have a different license
  - May "wrap" an external application
- New interface for downloading and installing plugins

```
$ plugin install --github pyopt_driver
```

- Plugins hosted at http://github.com/openmdao-plugins
- Three plugins contributed by Georgia Tech

Plugin Interface: Plugins

Components:

- adpac_wrapper
- excel_wrapper
- flops_wrapper
- nastranwrapper
- ommodelwrapper
- overflow_wrapper
- pdcyl_comp
- vsp_wrapper

Drivers:

- ipoptdriver
- pyopt_driver

Surrogate Models:

- neural_net

DOE Generators:

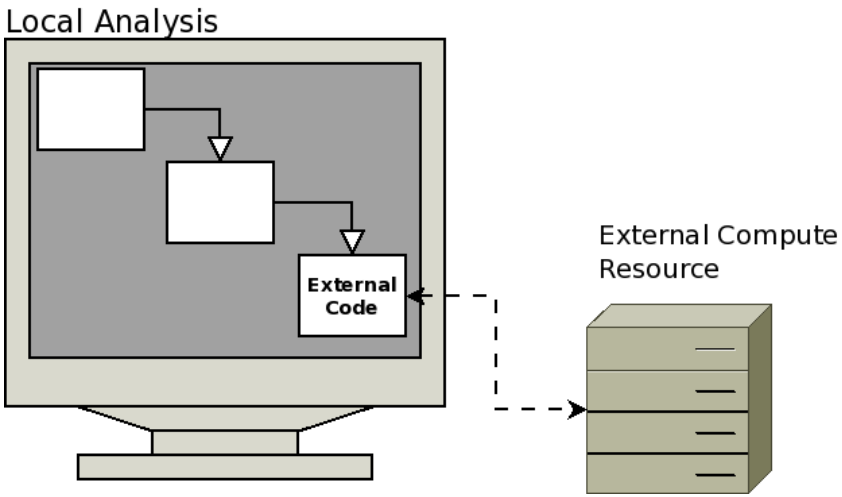
- montecarlo

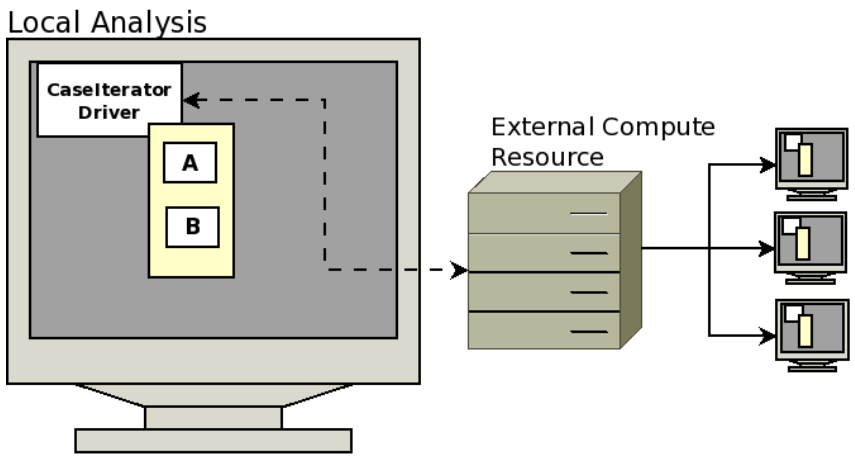
Resource Allocators:

- nas_access

Resource Allocation Manager (RAM)

- Abstracts the task of communicating with compute resources
- Produced RAM's for the Glenn HX cluster, Ames NAS cluster, local multi-core cpu
- Provides support for distributed and concurrent computing
  - External Code: Distributed computing
  - CaseIteratorDriver: Concurrent computing

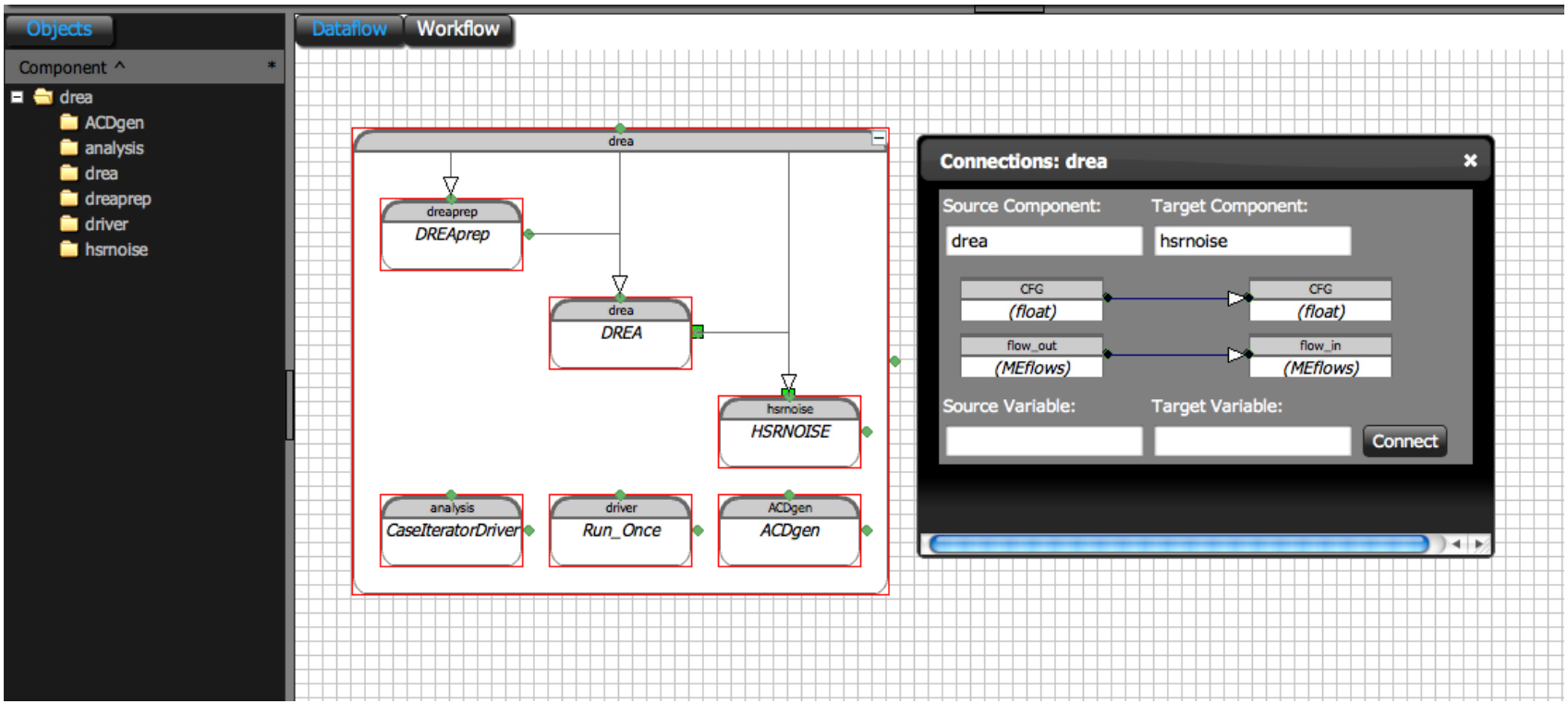




Browser Based GUI

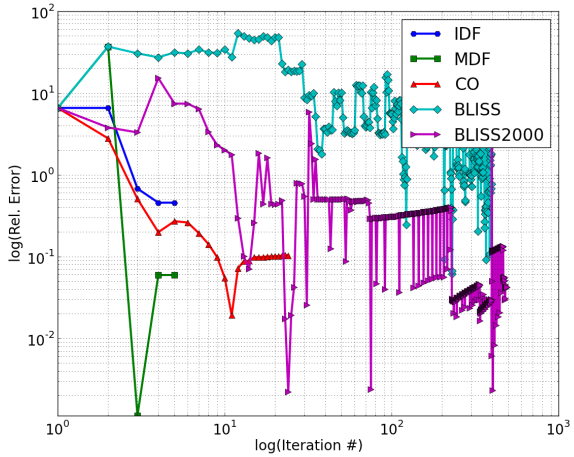
- Graphical view of both dataflow and workflow
- Web-based code editor
- Current version is part of the code-base, but not officially advertised or supported
- Official release scheduled for September 30, 2012
- Technologies: Javascript, ZeroMQ, WebGL, HTML5, WebSockets

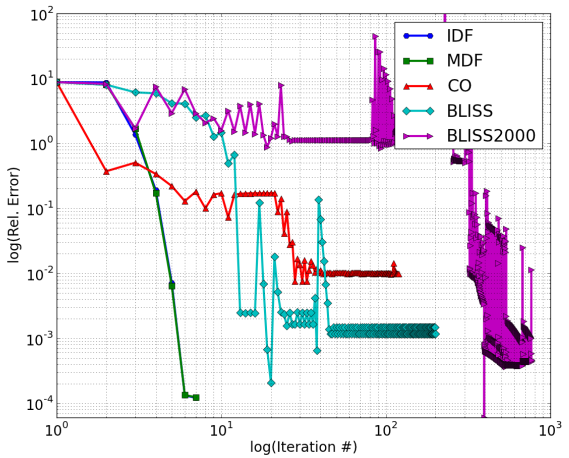
Browser Based GUI



Architecture Testing Platform

- Automatic implementations for BLISS2000, MDF, IDF, and CO
- Integrated test suite, with ability to run all architectures on all problems
  - Scalable test problem
  - Sellar problem
  - .....
- Test problems can include analytic derivatives





Support for Analytic Derivatives

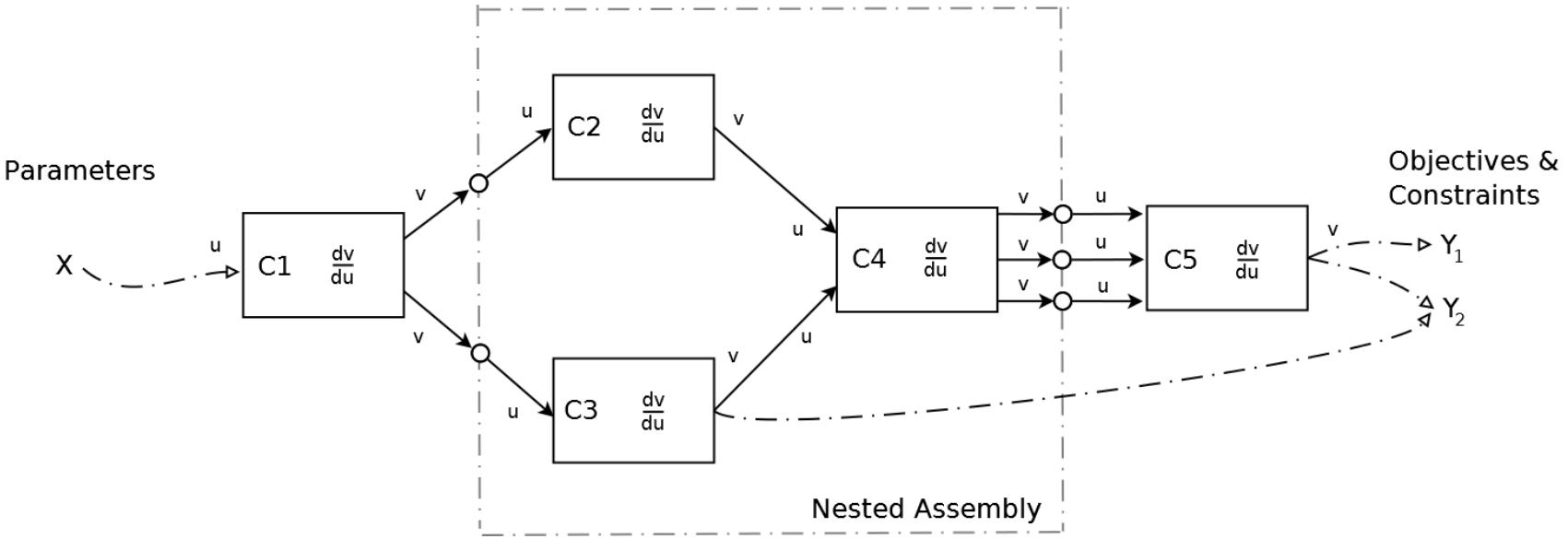
Three differentiators in OpenMDAO:

- Finite Difference
  - Analytic derivatives speed up the execution
  - Direct / ~~Adjoint~~ — Functional / Residual — Coupled
- Chain Rule 

  - Sub-blocks without derivatives must be finite differenced
  - Direct / Adjoint — Functional / Residual — ~~Coupled?~~
- Analytic 

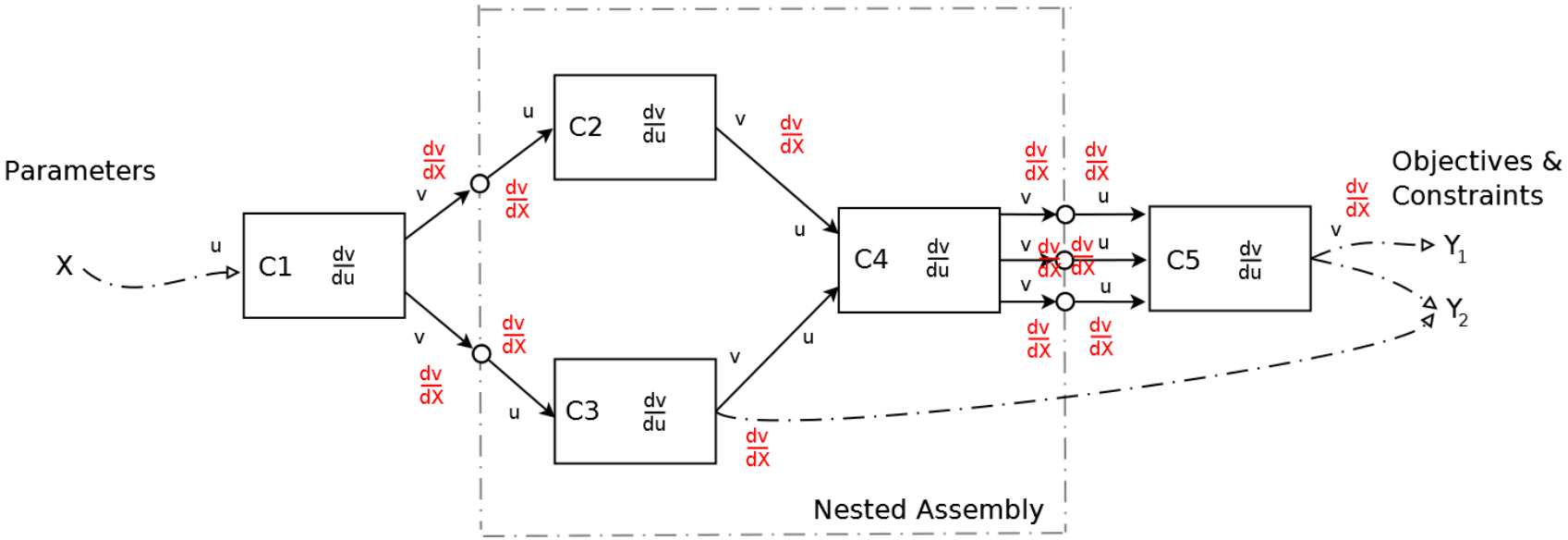
  - Sub-blocks without derivatives must be finite differenced
  - Direct / Adjoint — Functional / Residual — Coupled

Chain Rule Differentiator



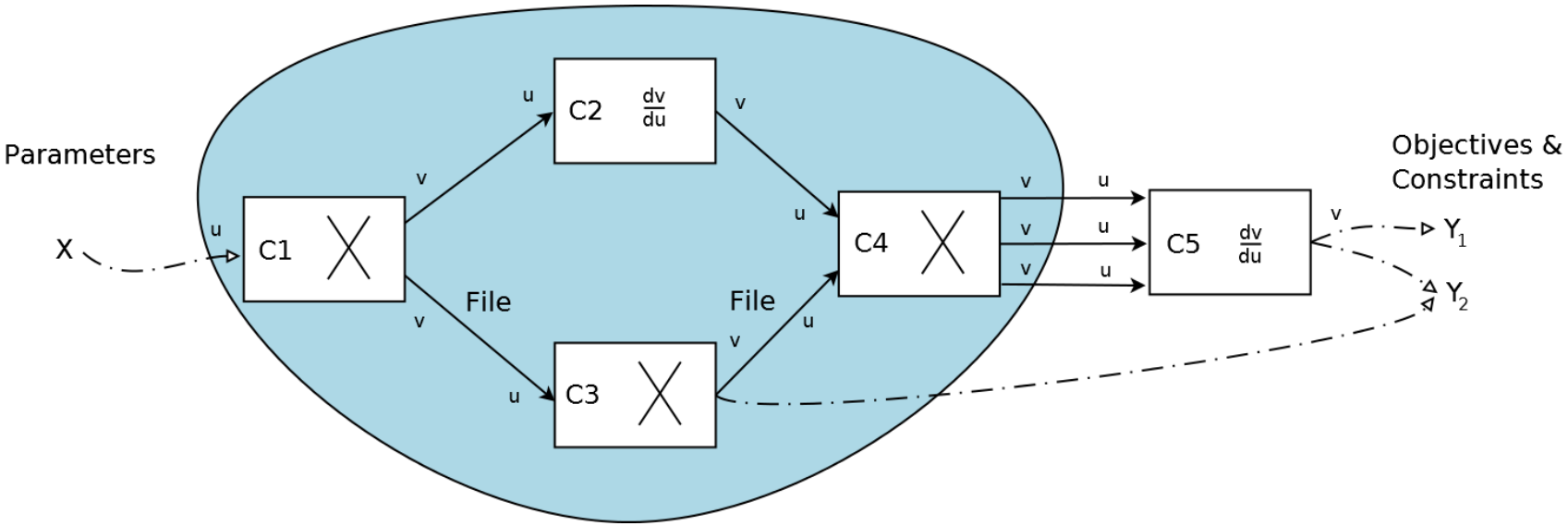
- Traverse network graph, cascading derivatives as you go
- Note: connections have expressions and unit conversion factors to differentiate

Analytic Differentiator



- Assemble system of equations from network graph
- Solution by numpy linalg.solve (more methods in the future, including sparse)

Identifying Blocks to Finite Difference



1. Comps with non-differentiable connections must be grouped
2. Differentiable islands are not permitted
3. Otherwise group for convenience

Note: Work in progress

Miscellaneous

Framework:

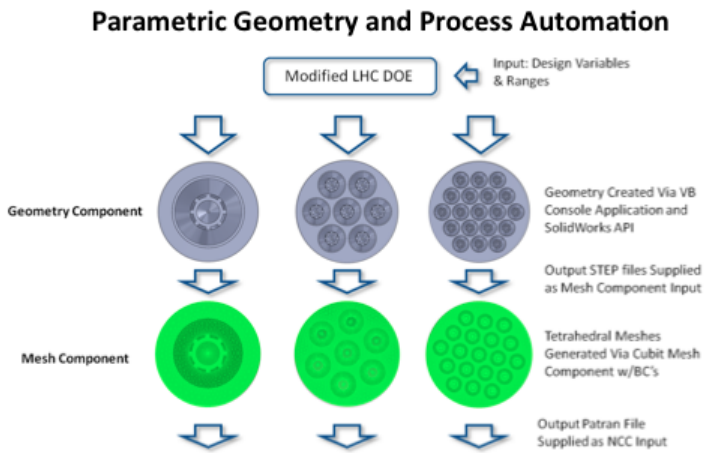
- New Optimizers: COBYLA, SLSQP
- New CaseRecorder/Iterator: CSV
- Uniform case recorder interface across all drivers
- Iteration hierarchy coordinate to help debug your model
- Variable connections can now have expressions

Infrastructure:

- Automated testing on 8 platforms using Amazon EC2
- Q&A forum on website (similar to Stack Overflow)

Lean Direct Injector (LDI) Design Space Exploration

- Analysis includes CAD, meshing tools, NPSS, NCC, TecPlot
- Calculations performed on HPC clusters automatically
- 30 Different designs being considered
- Each design requires approximately 24 hours on multiple CPUs



Future Work: Framework

Framework Development:

- Continued GUI development
- Integration with geometry tools
- Code performance and benchmarking

Infrastructure:

- New Website
- OpenMDAO Code Cookbook

Future Work: Test and Analysis

Test Problems:

- Developing an aircraft sizing optimization test case, using surrogate models, based on the NASA Adavanced Single Aisle Transport (ASAT) model

Analysis Problems:

- Turbo-electric distributed Propulsion
- Propulsion airframe integration for Supersonic aircraft
- All electric aircraft design problem

Questions?