

Applying executable, graphical workflows in SAW to electrical simulation-based analysis

Matt Glickman

Center for Computing Research

Sandia National Laboratories

Outline

- Executable, graphical workflows in SAW
- Electrical/EM Applications
 - Wrapping an older, powerful EM code to lower usability barriers
 - Assembling a toolkit for compact model calibration
- Some observations
- Acknowledgements

SAW: Sandia Analysis Workbench

- Adaptation of open-source integrated software development environment to support analysis via computational simulation

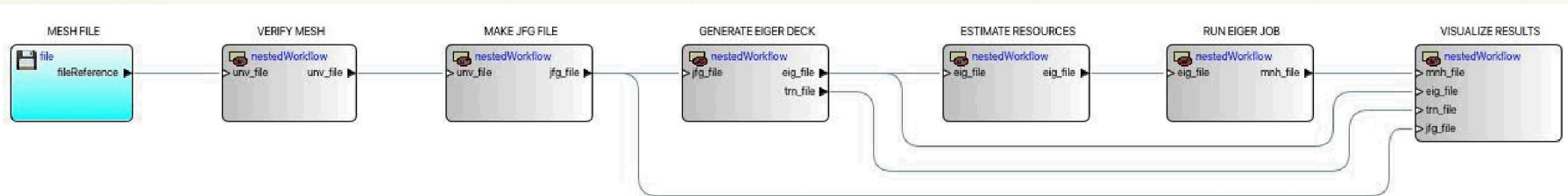


- Developed & refined at SNL over 10 years
- Supports
 - Integrated tools for model composition and configuration
 - HPC job submission and monitoring
 - Management of simulation data via a networked repository
- Value parallels that of software IDE's
 - Integrated, more uniform interface to wide array of capabilities
 - Automation of key functions where possible

Motivation

- Major steps in CompSim workflows are mediated via a combination of *analyst actions* and *small/informal scripts*
- Hard to scale
 - Analysts are limited resource
 - Scripts suffer from “poor software hygiene”
- Threat to credibility
 - Linkage between steps is obscured or lost, so workflows aren’t traceable, repeatable
- Proposed (SAW) solution
 - *Executable, graphical representation for workflows*

Why graphical workflows?



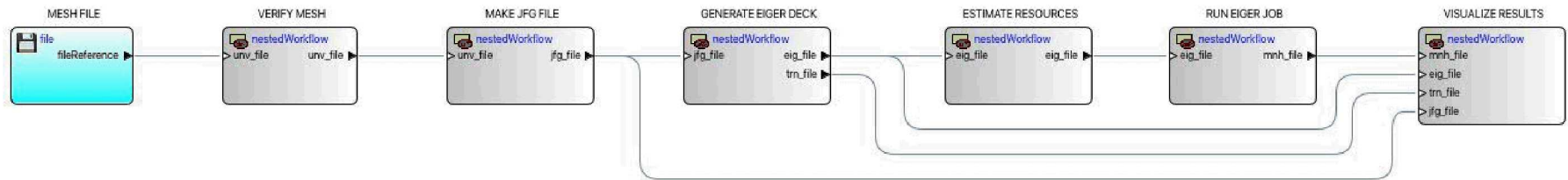
- Standardized, integrated solution
- Graphical workflows clearly represent linkage between artifacts
- Artifacts identified by *spatial context*, not *names*

File

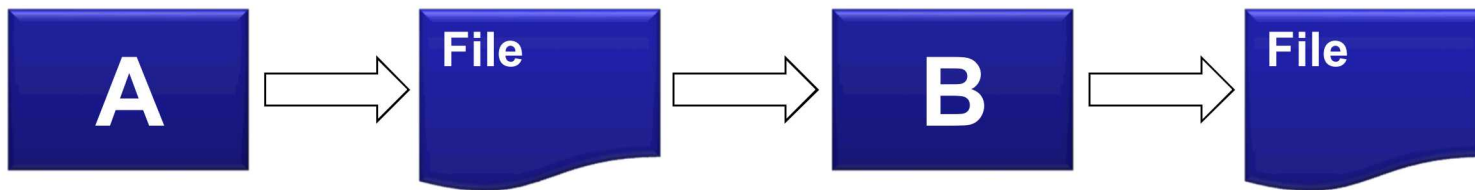
File

- Graphical representation “naturally” constrains complexity

Why graphical workflows?



- Standardized, integrated solution
- Graphical workflows clearly represent linkage between artifacts
- Artifacts identified by *spatial context*, not *names*

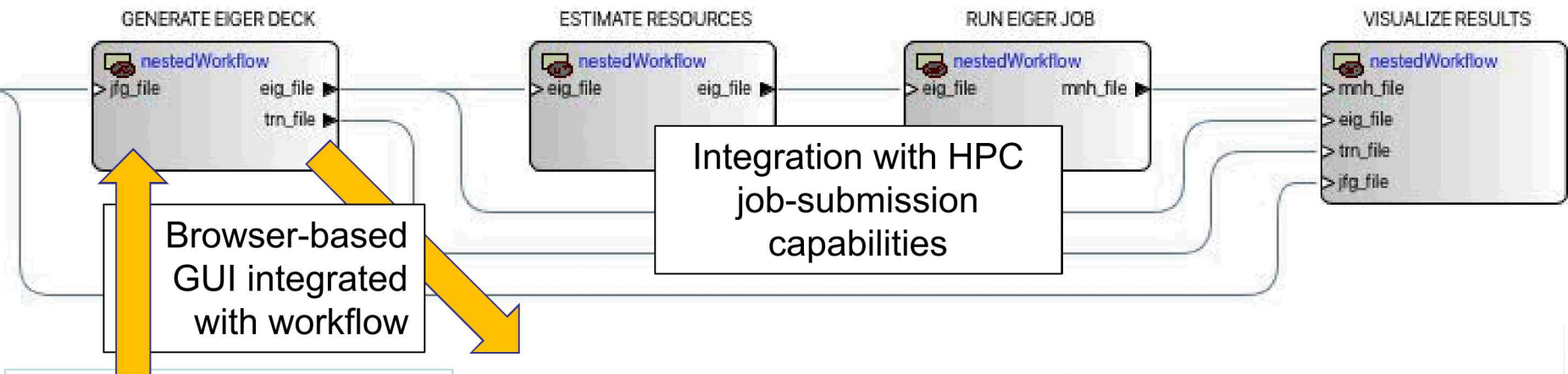


- Graphical representation “naturally” constrains complexity

Graphical workflow for EIGER

- EIGER: powerful, frequency-domain EM modeling code with unique capabilities
- Designed for “developers = users” use-paradigm
 - Heavy emphasis on computation (not usability)
 - Complicated, unforgiving toolchain, cryptic error messages
- Desired benefits from graphical workflow:
 - ability to generate flexible workflow schema allowing *non-developer analysts* to set up, run, and post-process EIGER jobs, including integration with Dakota
- Workflow highlights:
 - input deck preparation
 - HPC job submission
 - overall integration

End-to-end EIGER workflow



Prepare Eiger input deck

Jungfrau geometry file: *geometry-file-name-goes-here*

Eiger deck basename

Description

Units

Properties

Type

Material

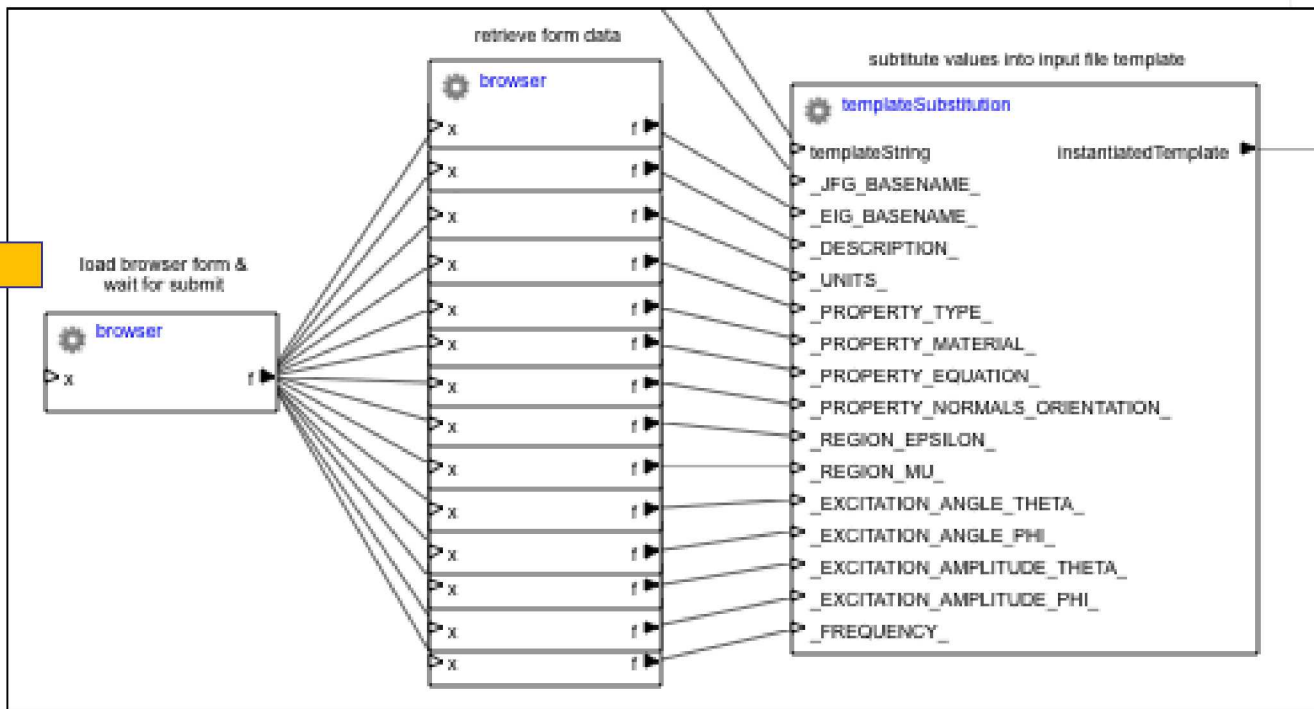
Equation to be applied

Orientation of surface normals

Region Properties

Relative ϵ of homogenous region

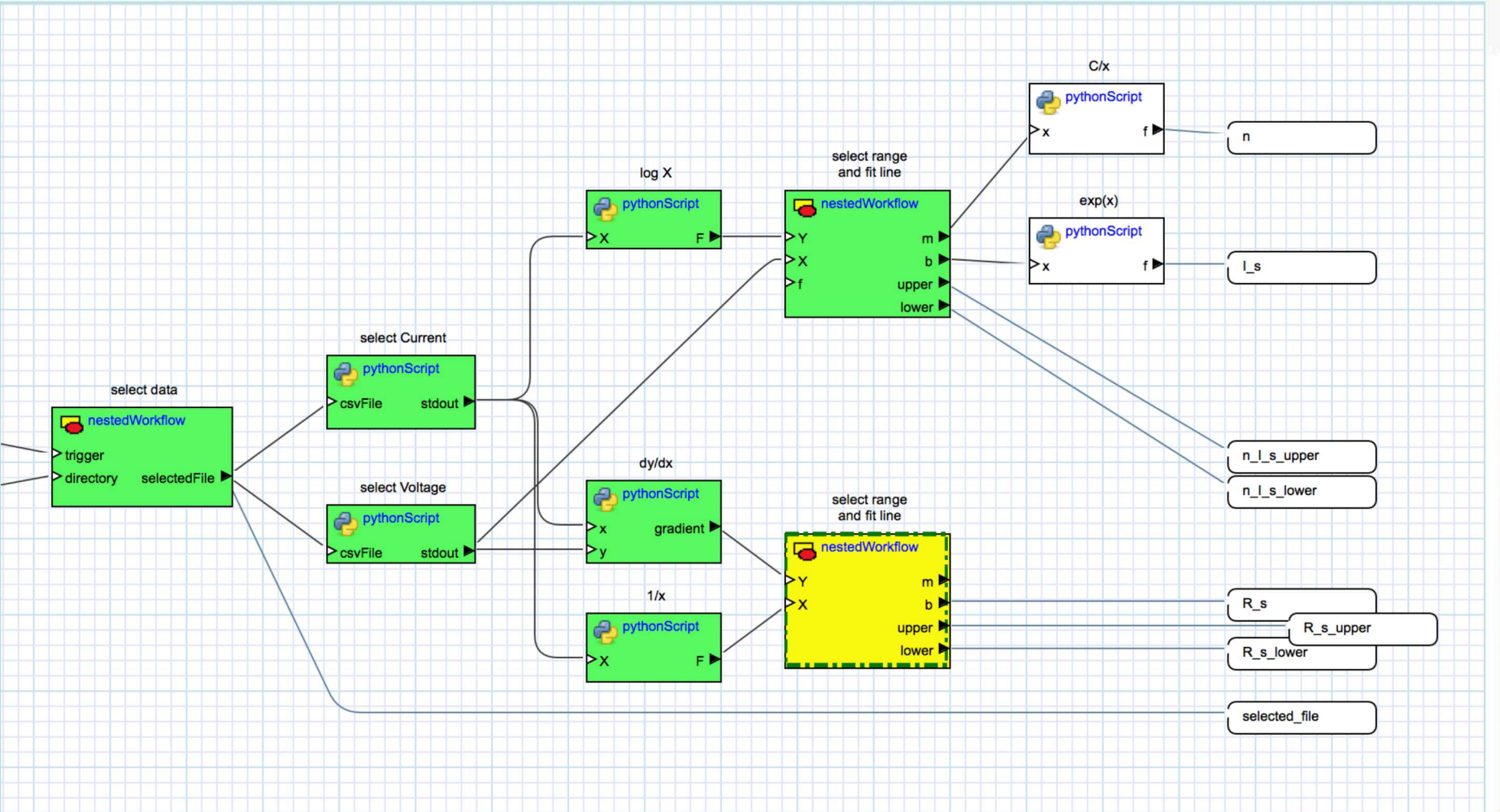
Relative μ of homogenous region



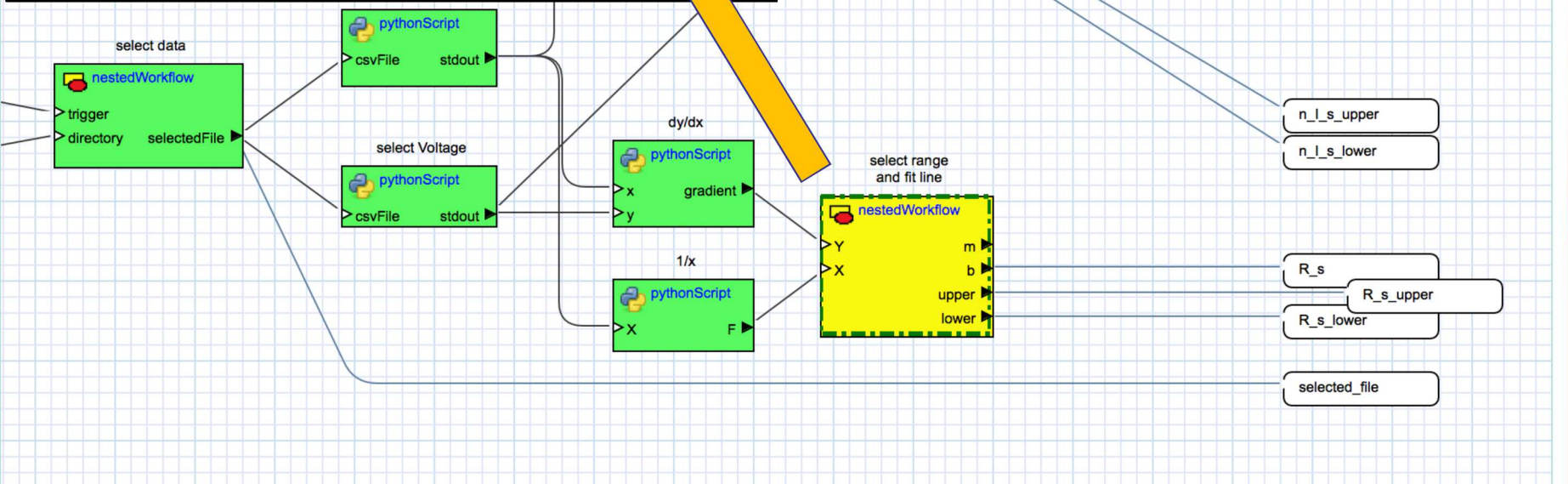
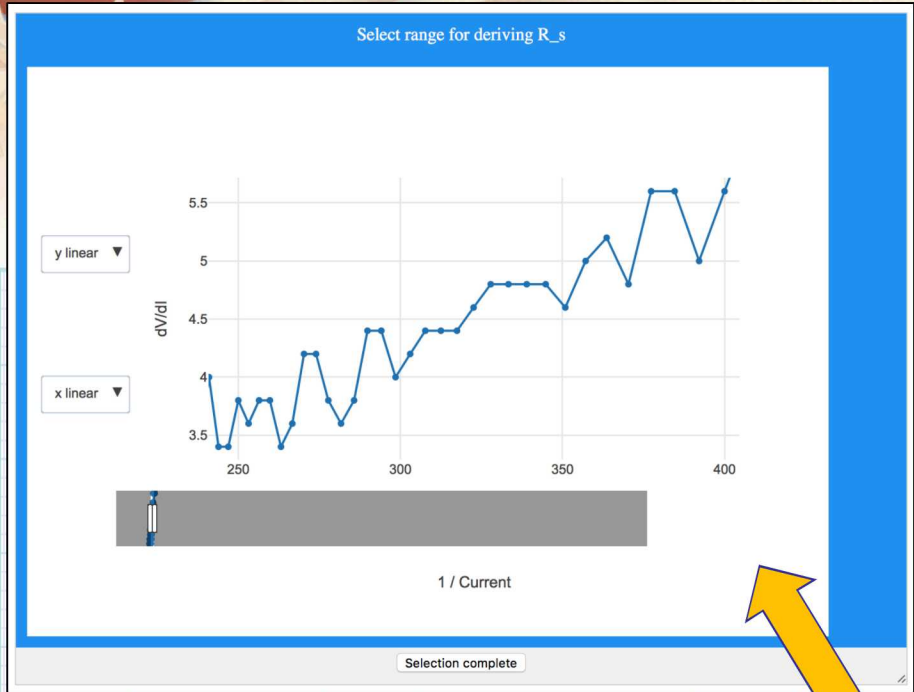
Compact model calibration framework

- Currently carried out via commercial tools
 - Workflow hard to capture, difficult to integrate
- Goal: a “toolkit” from which calibration workflows may be *constructed, saved and shared*
 - "Jailbreak for model calibration"
- Benefits
 - Customization/flexibility
 - Share, reuse, traceability
 - Ability to integrate with new capabilities as developed

Zener Diode Forward Parameter Extraction



Zener Diode Forward Parameter Extraction



Observations

- Crafting the highest-utility subcomponents is likely a long, iterative process.
- Rich GUI elements seem highly valuable, but so far require “classical” programming.
- Building end-to-end workflows is alluring, but risks over-constraining analysts
 - *Better: rapid iterations with smaller prototypes*
- Intrinsic benefits, appeal of graphical representation continue to appear very real.

Acknowledgements

- **SAW**

- Ernest Friedman-Hill (*principal architect of SAW workflow capability*)
- Ed Hoffman
- Marcus Gibson
- Kevin Olson
- George Orient
- Robert Clay

- **EIGER**

- Joe Kotulski

- **Compact model calibration**

- Shahed Reza
- Tom Buchheit
- Ian Wilcox
- Jason Verley