ARMI™ Workshop

Joint ICTP-IAEA Workshop on Open-Source Nuclear Codes for Reactor Analysis | (smr 3865)

2023-08-10

TerraPower, LLC

armi-devs@terrapower.com

Agenda

- Welcome
- Brief ARMI™ introduction presentation

- Installation of ARMI™
- **Check-in**: checking installation

- Additional high-level introduction while people finish get installed
- Open up the gallery and go through a few examples.
- **Check-in**: Gallery help from chat questions

- Presentation about the ARMI Data Model
- Walkthrough data tutorial but with FFTF inputs
- **Check-in**: Answer Q&A questions from chat

- If time: Given H5 database, make a VTK or XDMF file
- View output in Paraview

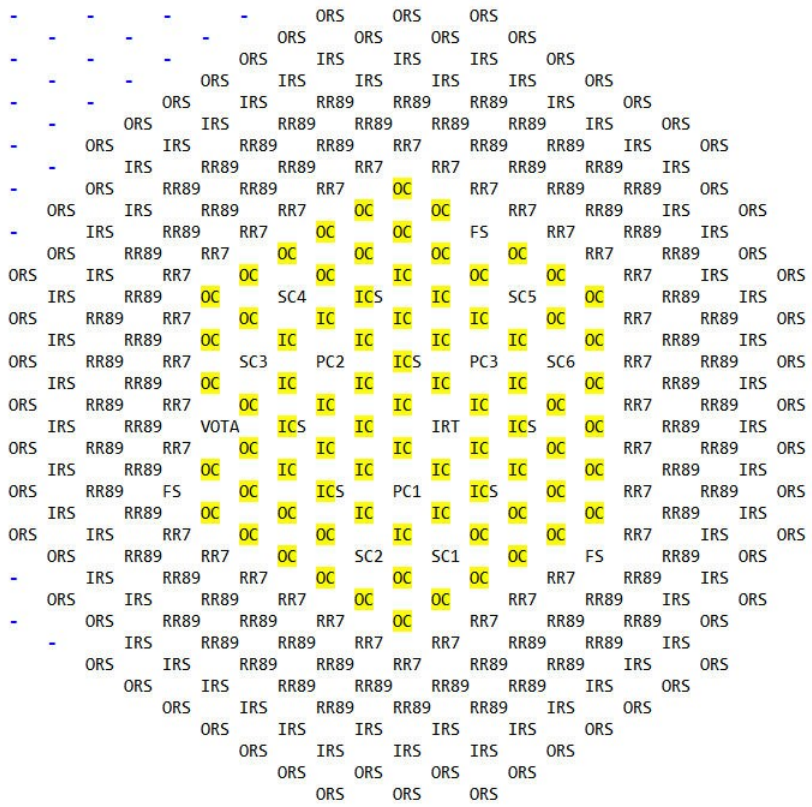
- Q&A and Discussion

TerraPower

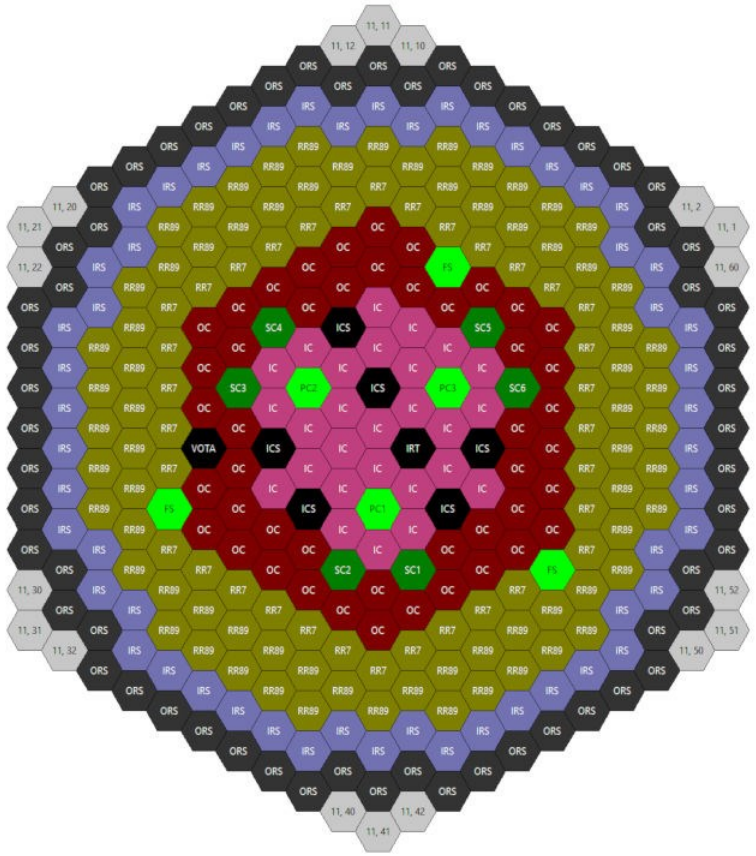
What to expect

- Command line
- Python programming
- Non-coders: there will be pretty pictures!
- ARMI is a *framework* with a built-in nuclear reactor specific data model
- ARMI is good at integration and automation
- Your team needs *developers* working with ARMI to make gainful use of ARMI at this point
- Later: We hope an "off-the-shelf" set of plugins exists sufficient to do meaningful work without much extra development
- The TerraPower ARMI plugins are largely proprietary at this point
- ARMI has had a SFR-dominant focus

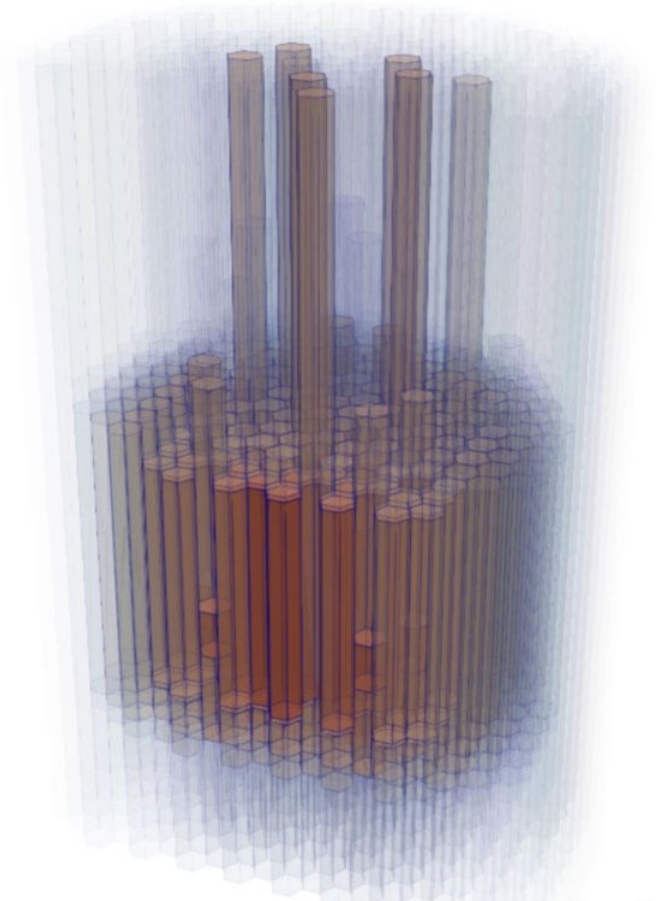
Teaser of what we'll do today



Input File



Grid Editor GUI



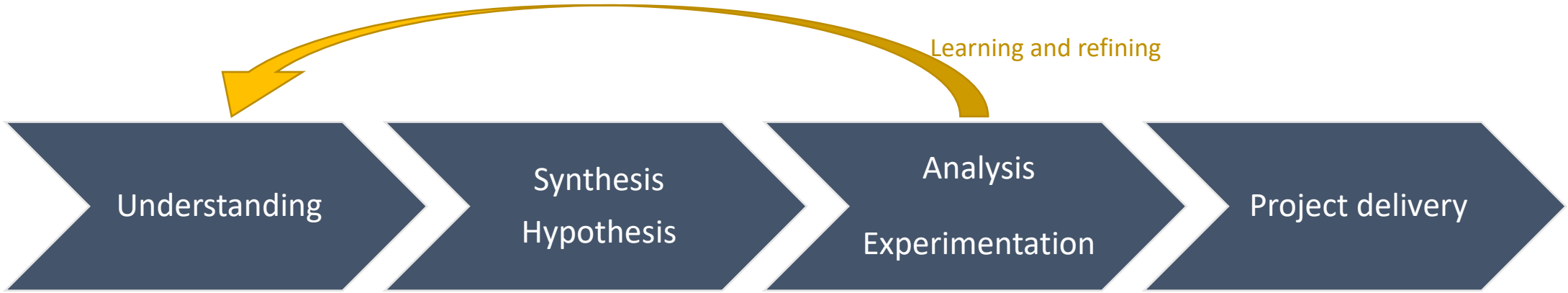
Output (VTK file)

TerraPower

What is ARMI™?

TerraPower

Improving reactor performance



- Established theory
- History
- Geopolitics
- Cost drivers

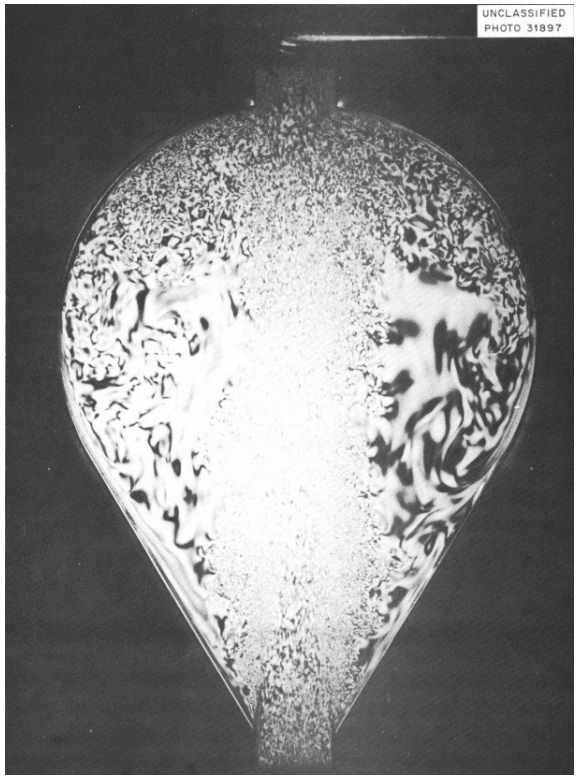
- Creativity
- Vision

- **Classical analysis**
- **Numerical analysis**
- Physical laboratory
- **Numerical laboratory**
- **Trade-offs**
- **Decision making**

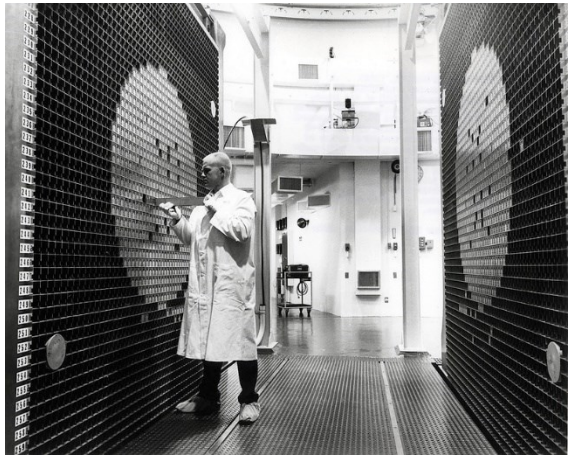
- Licensing
- Construction
- Operation
- Maintenance
- Decommissioning

TerraPower

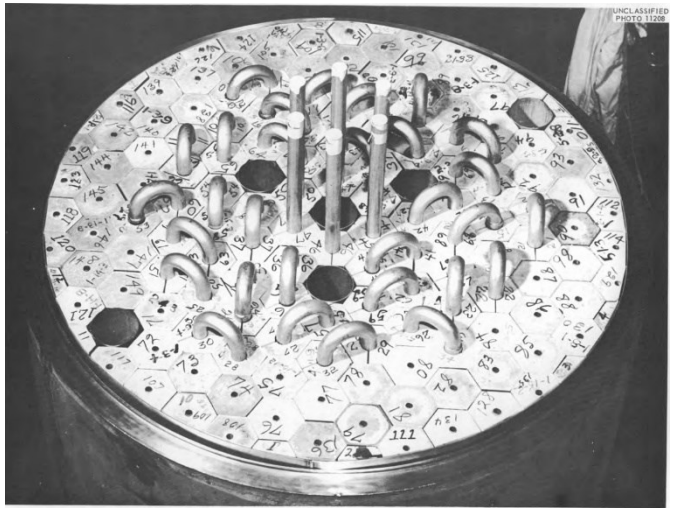
In the old days, we built exceedingly exotic reactors, but we had to use *physical* models/analogs



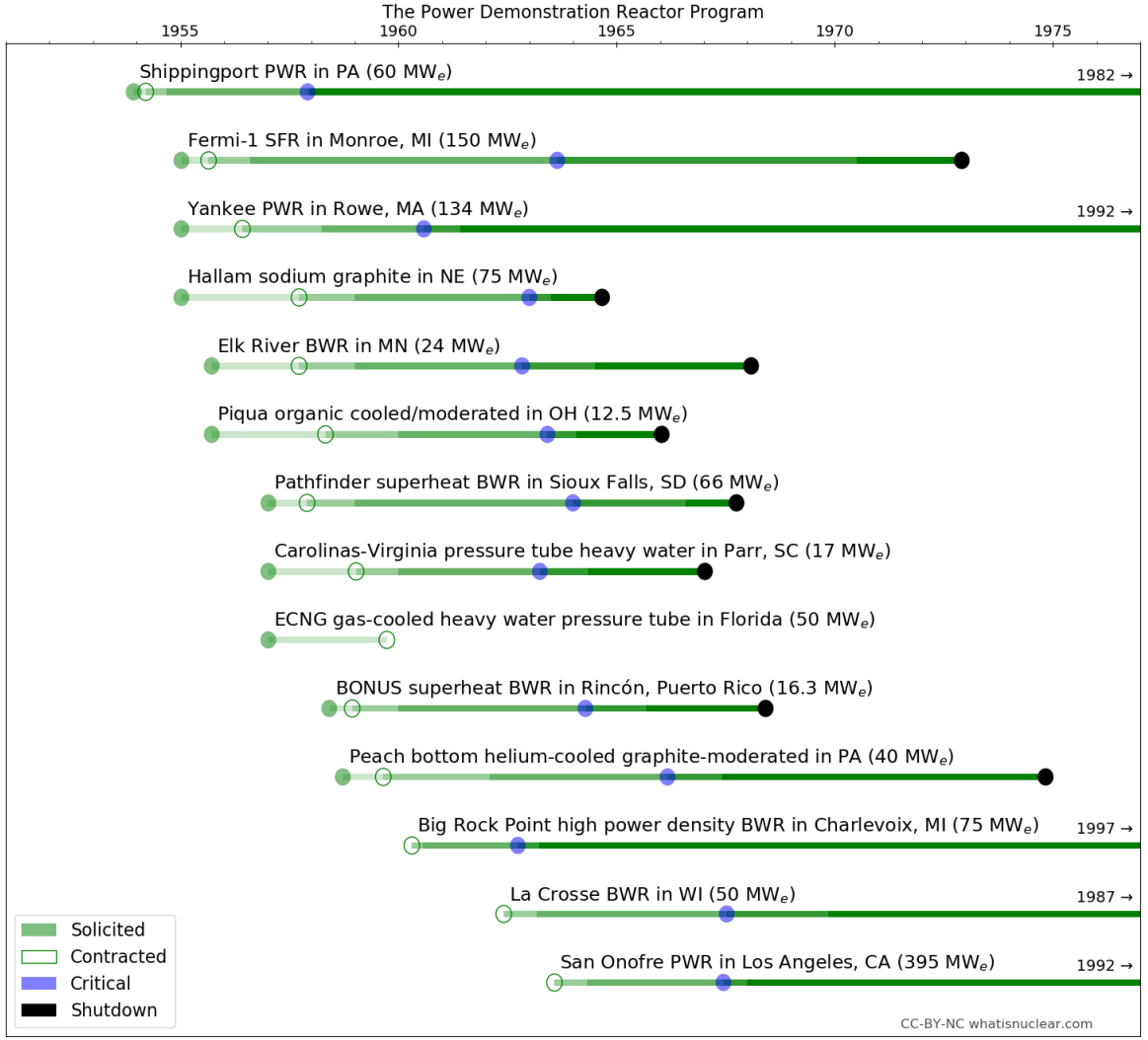
A Milling-Yellow solution exhibits an optical property called *flow double refraction*



Split table critical core mockup facility



The Aircraft Reactor Experiment (1954)



The Power Demonstration Reactor Program

TerraPower

Today, we run 1000s of models on one piece of hardware

- Better understanding of physical data
  - Nuclear cross sections
  - Previous experiments
- Higher-power digital computers
  - 3D fine mesh
  - Monte Carlo
  - CFD

New tech still requires experimental validation, but this can be very targeted

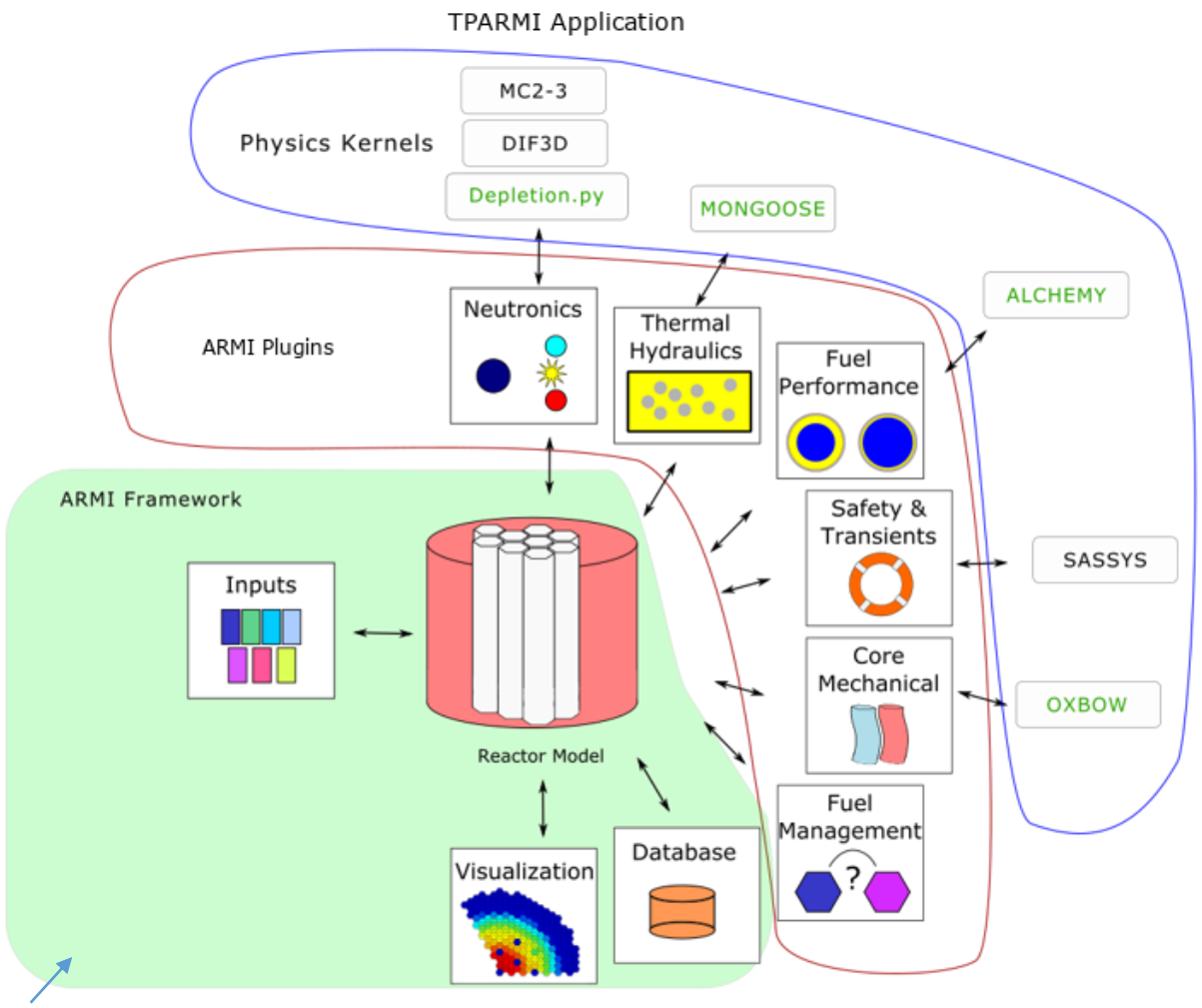
TerraPower, with its small team, sought to leverage software to a new degree

ARMI provides *Reactors at your fingertips*

Advanced Reactor Modeling Interface (ARMI ™)

A system for reactor design & analysis

- Enables efficient *communication* and *coupling* between codes and the analysts that use them.
- Automates detailed and complex modeling workflows
- Accommodates development of new tools

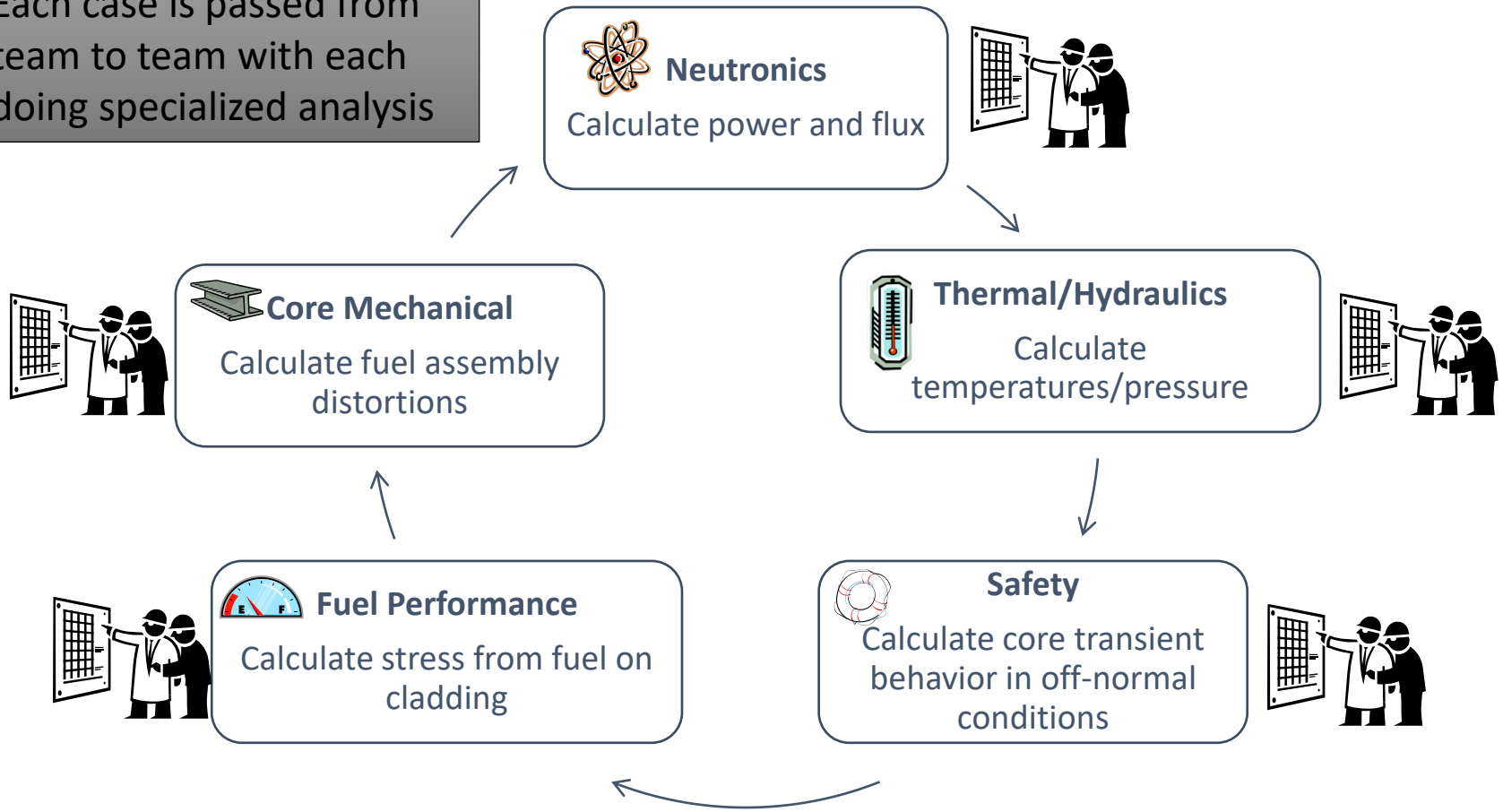


https://github.com/terrapower/armi

TerraPower

Traditional workflow is slow, laborious, and error-prone

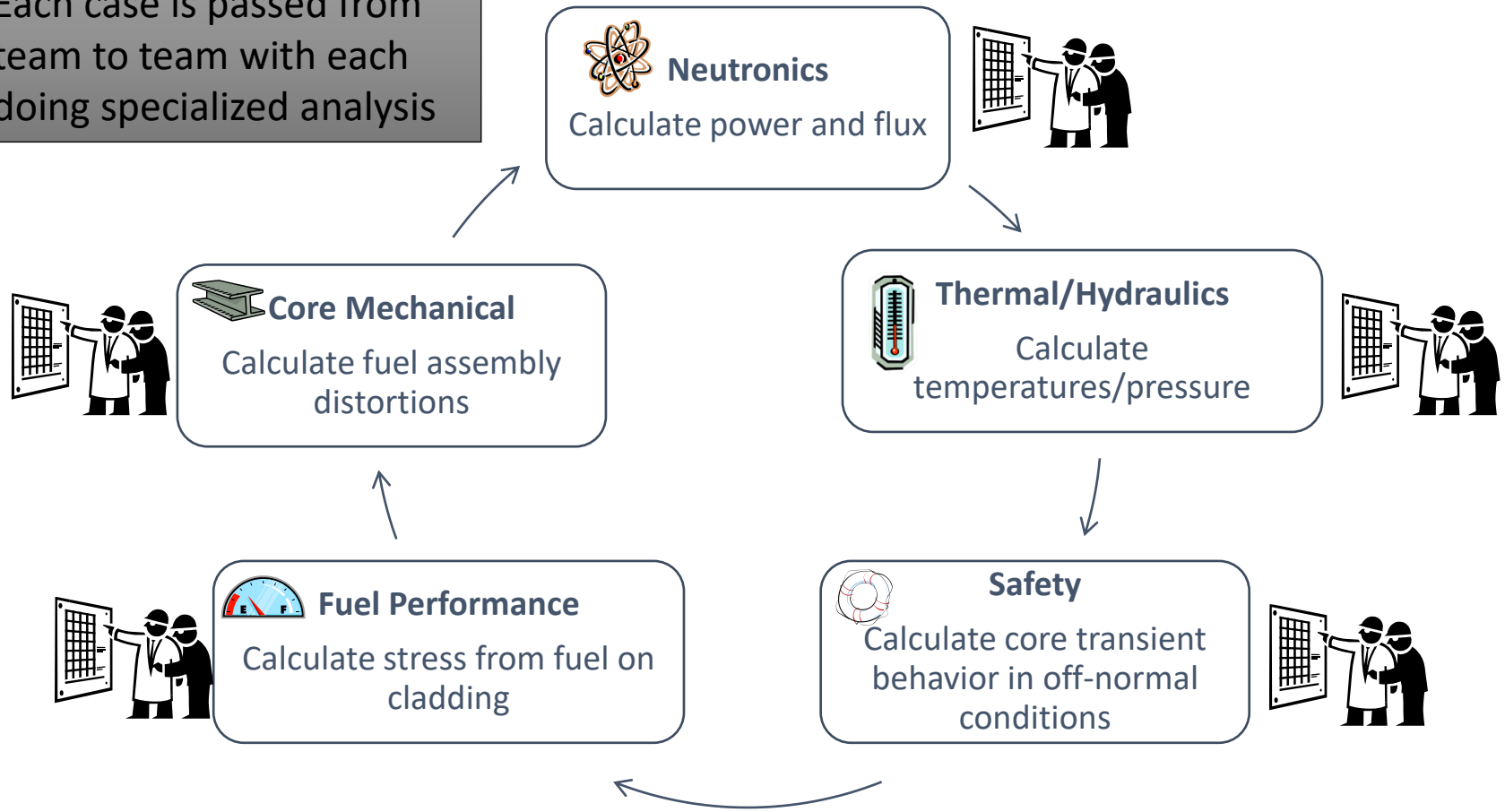
Each case is passed from team to team with each doing specialized analysis



TerraPower

Traditional communication and coupling

Each case is passed from team to team with each doing specialized analysis

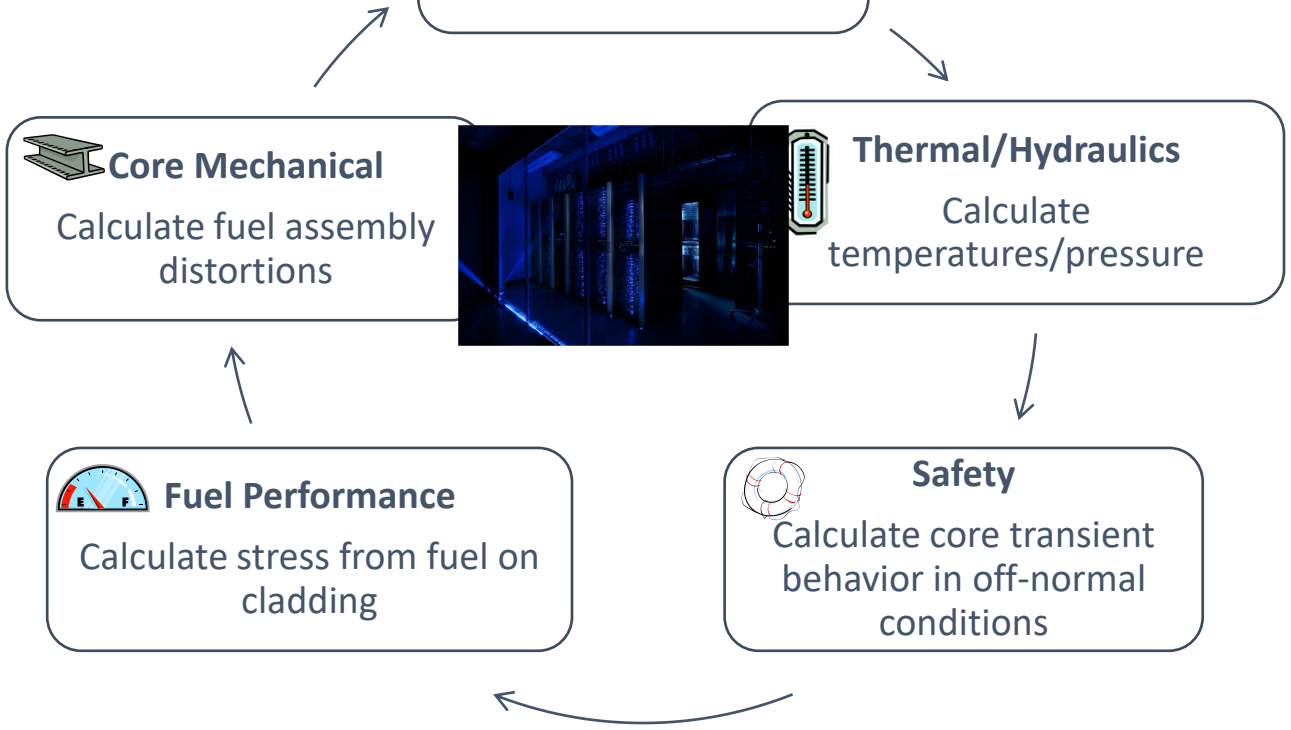


TerraPower

ARMI workflow is streamlined

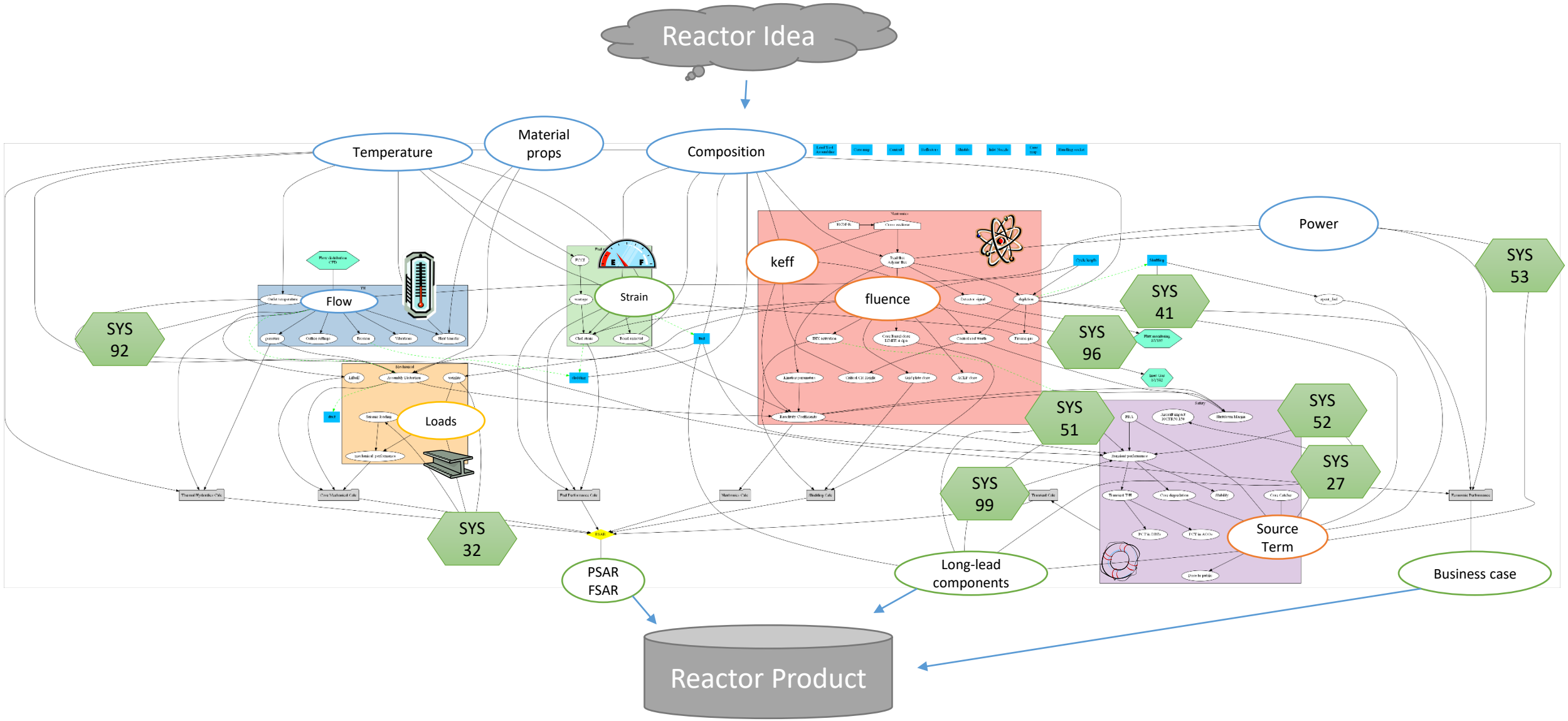
ARMI allows each single analyst to consider all aspects of core design

Teams focus on developing models



TerraPower

ARMI automation enables consistency in complex systems



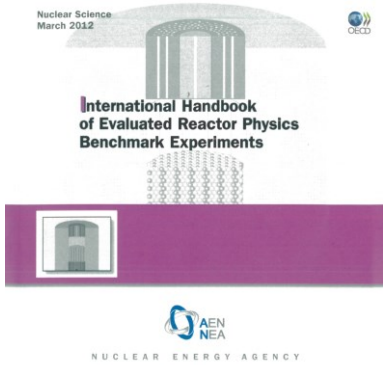
TerraPower

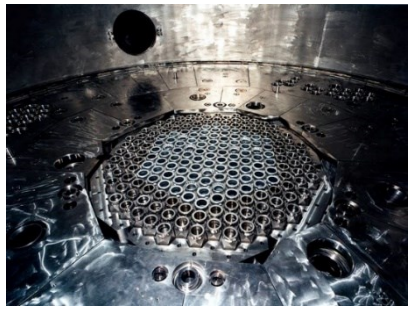
ARMI's differentiators

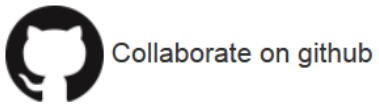
- Unapologetically nuclear specific
  - Reactor data model
- Made by reactor analysts for reactor analysts
- 14 years of advanced reactor industrial use
- High-level programming language: low barrier for 'regular' engineers
- Open source

Other benefits of a universal reactor model

- Collaborative benchmark models for V&V
  - Capture widest knowledge to best preserve degrading experimental data
- Reduce data management burden of nuclear development
  - Plug your new code into a ecosystem, leverage entire community's efforts
- New opportunities in collaborative reactor ***design*** projects
  - Less institutional friction to collaborate
  - Projects could be more robust against hardship
- Collaboratively develop and approve automated analysis *methodologies*
  - Identify mistakes early ("fail fast")
  - Regulator could become familiar, have more faith







TerraPower

Additional use cases for research institutions

- ARMI can help facilitate open science
  - Fully-defined reactor models (not just tables of highlights)
  - Fully-defined methodologies (equations + *implementation!*)

- ARMI can improve research efficiency
  - High-level, nuclear-specific data management
  - Architecture as "training wheels" for newbs
  - Temporal continuity

Workshop Interactivity Review (and help)

- Make sure your computer has Python 3.9-3.11

```
> python -V
Python 3.11.4
```

- Make a virtual environment (also: what is a virtual environment)

```
> python -m venv C:\users\yourusername\venvs\armi-venv
```

- Activate Virtual Environment

```
> C:\users\yourusername\venvs\armi-
venv\scripts\activate
```

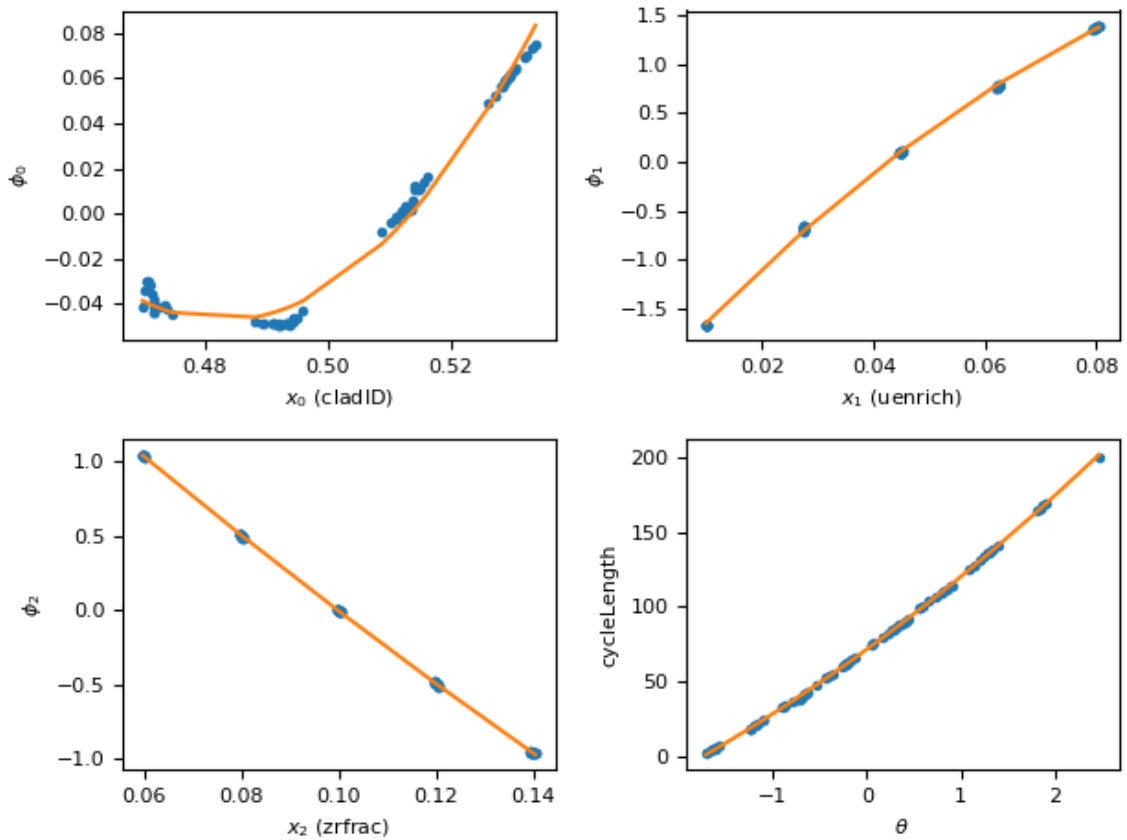
Shortcut to instructions: **2ran.net/armi**

More intro

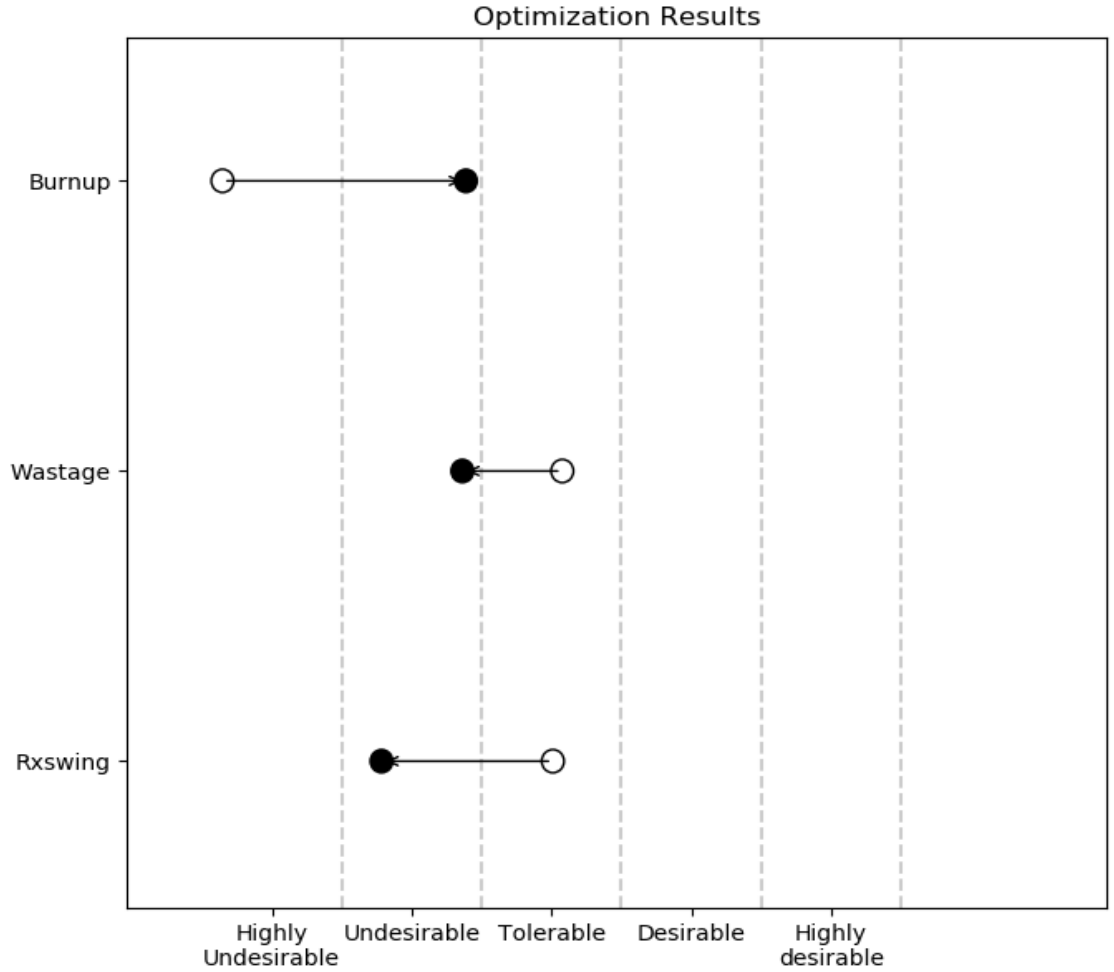
TerraPower

Statistical analysis and optimization

We use sophisticated, non-linear regressions to build high-speed surrogates of the design space



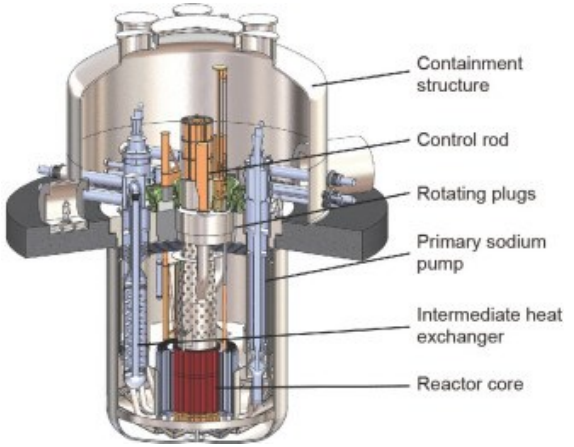
Surrogates can passed into multi-objective optimization algorithms

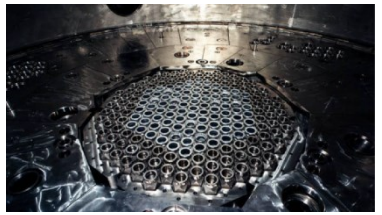


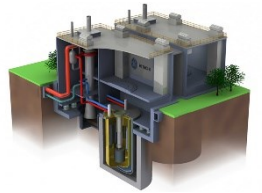
TerraPower

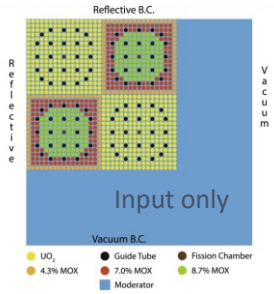
Successes

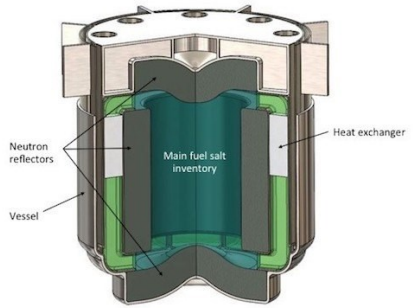
- TWR fuel management
- Driven core design and analysis of the project for past decade
- Investments in *methodologies* live on
  - Resilient to design changes
  - Resilient to staffing changes
- MCFR uses ARMI
- Medical isotopes team
- VTR model built in 2 days; could perform preliminary design-level analysis
- Executed contract with INL to couple one of their flagship physics tools (BISON) to ARMI
- Other successes?

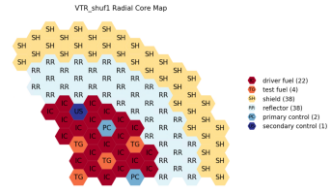






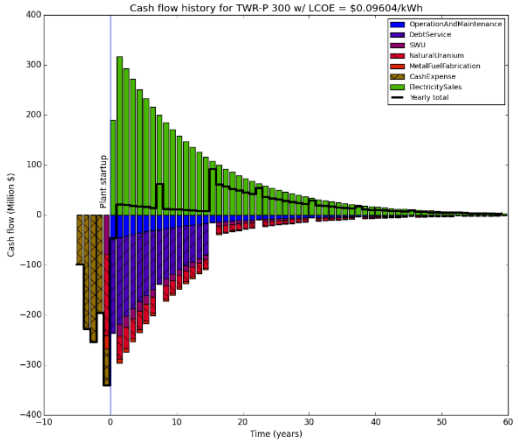


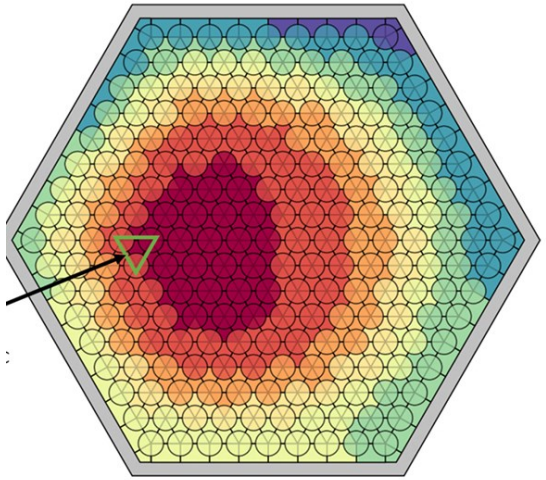


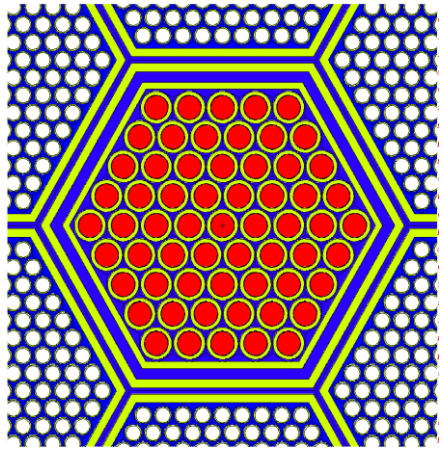


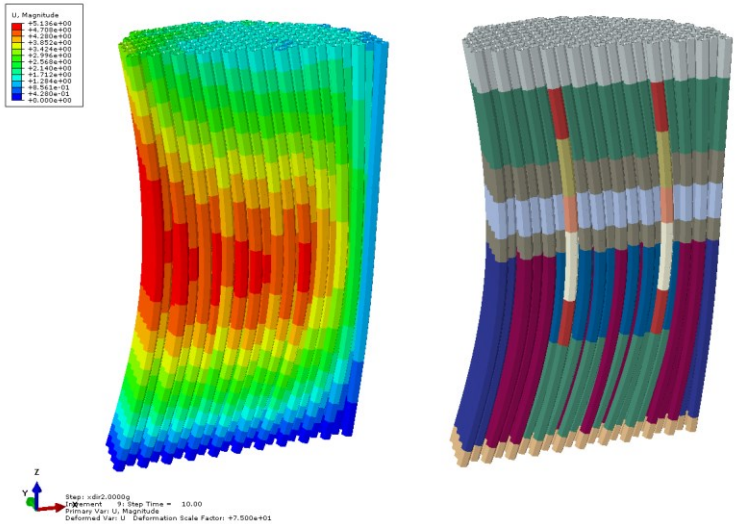
TerraPower

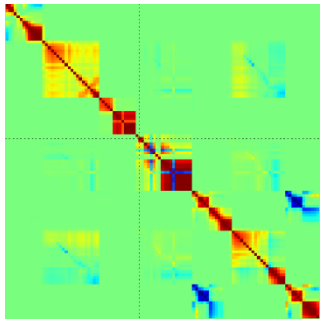
Some SFR ARMI highlights with TerraPower plugins

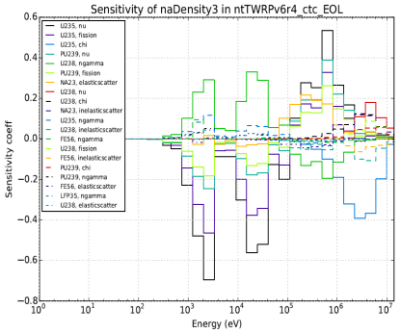


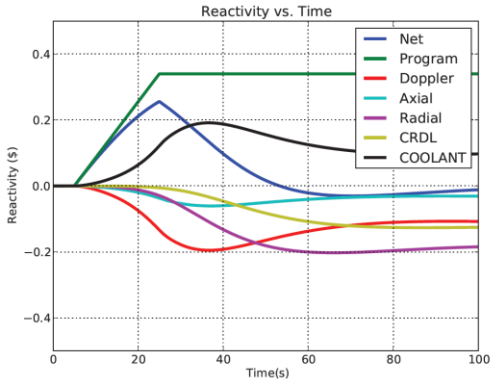


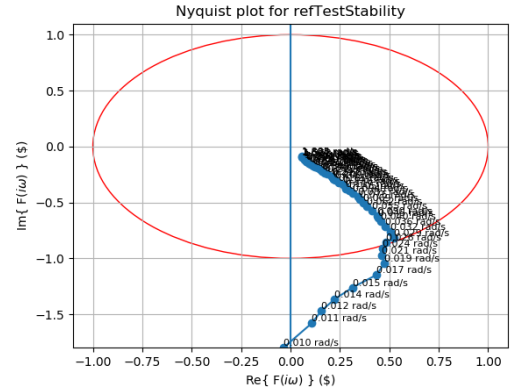






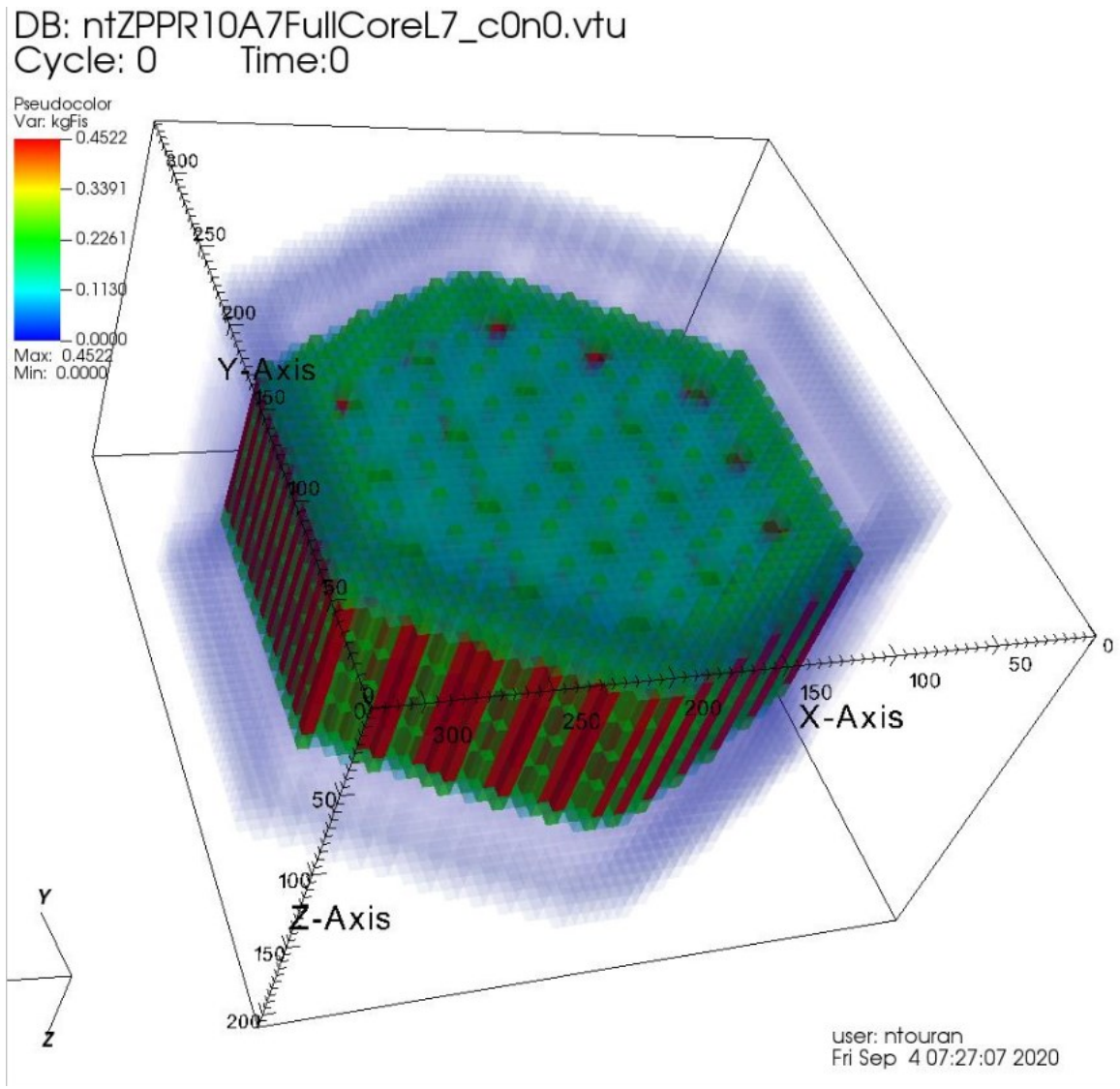


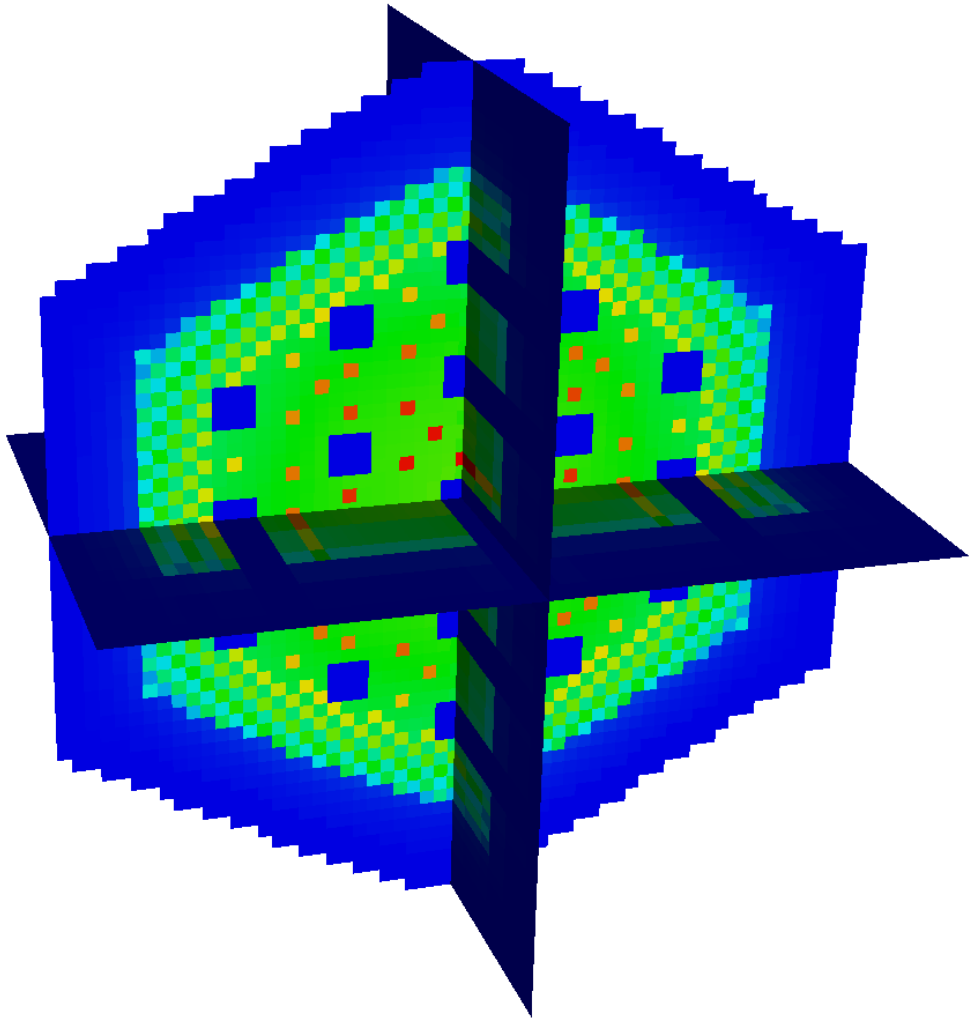




TerraPower

ZPPR Models





TerraPower

List of existing plugins (mostly not open source)

- DRAGON
- Windows HPC
- Crucible
- Depletion
- DIF3D
- Economics
- Equilibrium fuel cycle
- Fluxrecon
- Gamma transport
- Intrinsic source
- MC2-3
- MCNP
- NJOY
- Nuclear Data UQ
- PARTISN
- Reactivity control
- REBUS
- Reactivity Coefficients
- SASSYS
- Stability
- SFR T/H
- VirDenT

For details/descriptions, see: https://github.com/terrapower/armi-plugin-directory

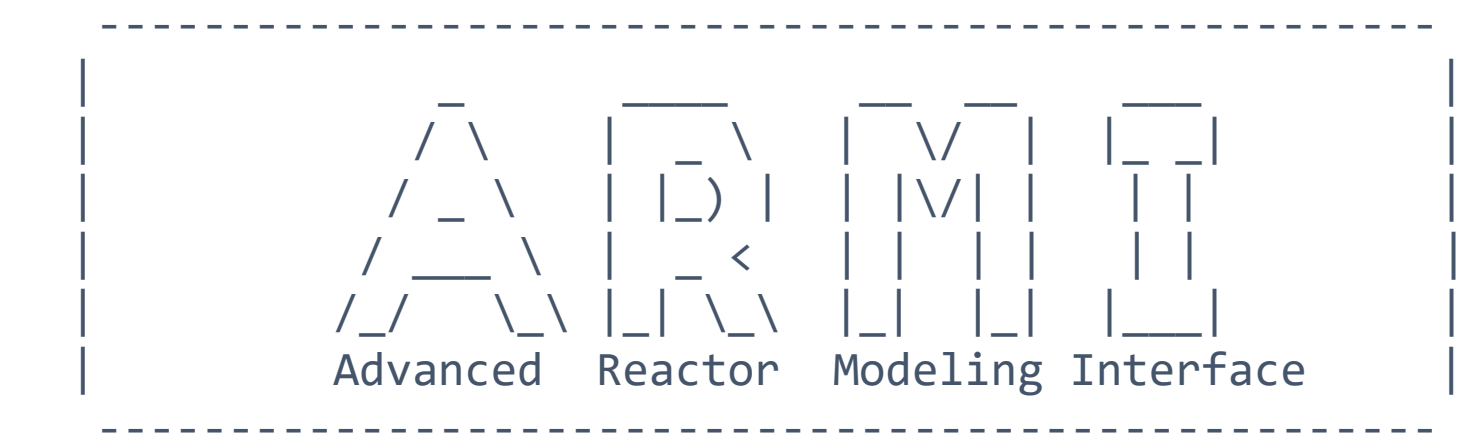
Installing ARMI

TerraPower

Installing ARMI

```
(armi-venv) $ pip install https://github.com/terrapower/armi/archive/main.zip
```

```
(armi-venv) $ armi
```



TerraPower

Exploring the Gallery!

https://terrapower.github.io/armi/gallery

First, run this:

```
> pip install ipython
```


TerraPower

The FFTF input files

https://github.com/terrapower/fftf-isothermal-model

TerraPower

Try out the grid editor

Windows/Linux: `pip install wxpython`

Mac: `brew install wxpython`

Linux if compiling: `apt install python3-wxgtk4.0`

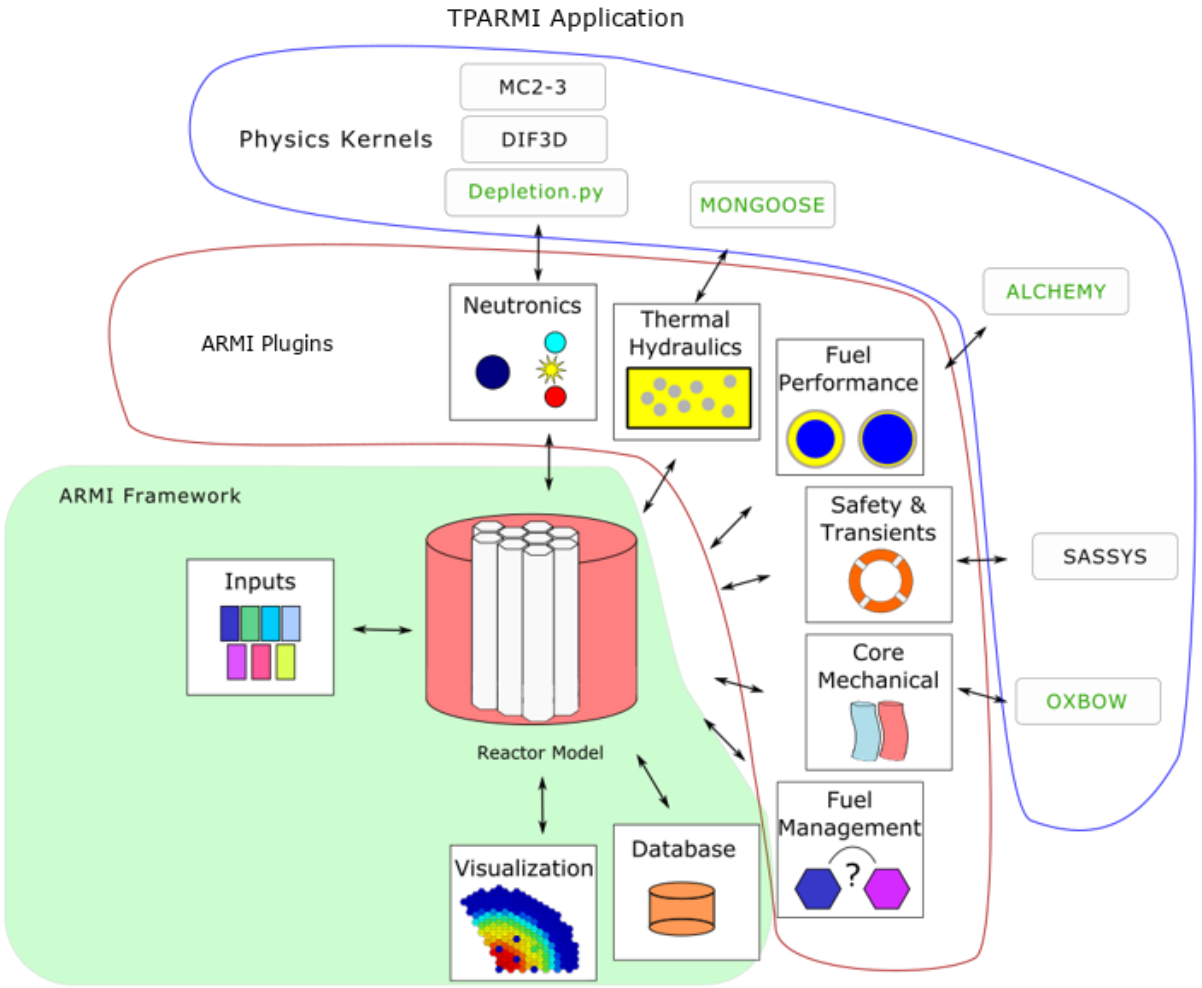
Then run:

```
armi grids FFTF-blueprints.yaml
```


TerraPower

ARMI ARCHITECTURE

Framework, Plugin, Application



TerraPower

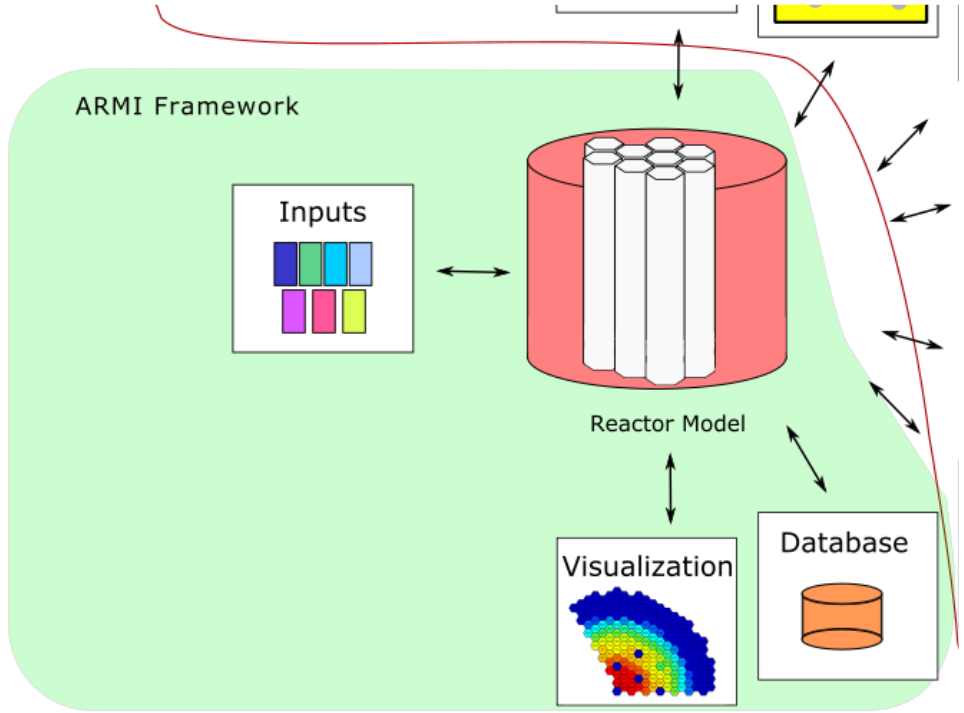
ARMI Framework Concepts

The ARMI Framework is comprised of the following concepts:

- Composite Reactor Model
- Parameters
- Database I/O
- Interfaces
- Operator
- Blueprints
- Settings

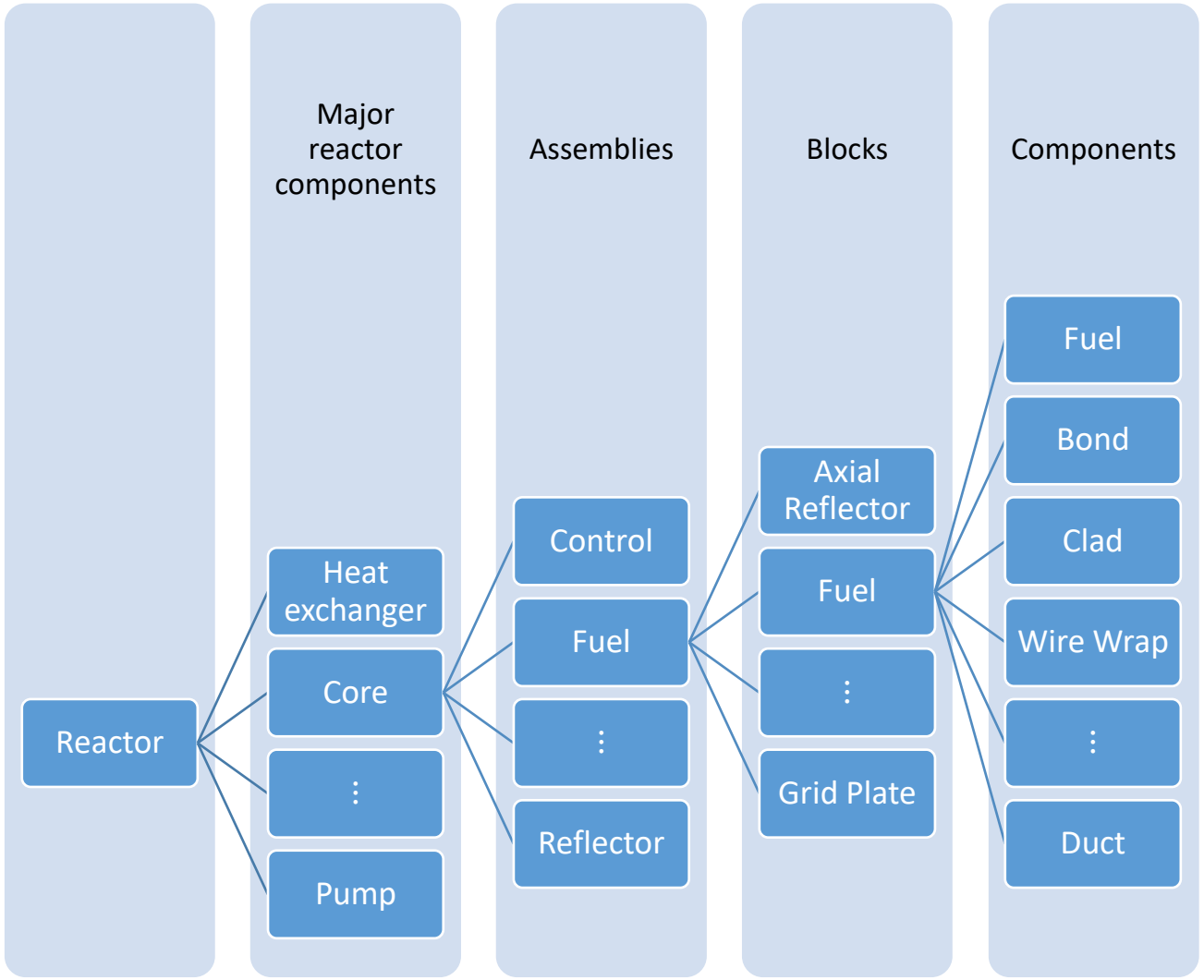
Flags

USER INPUT



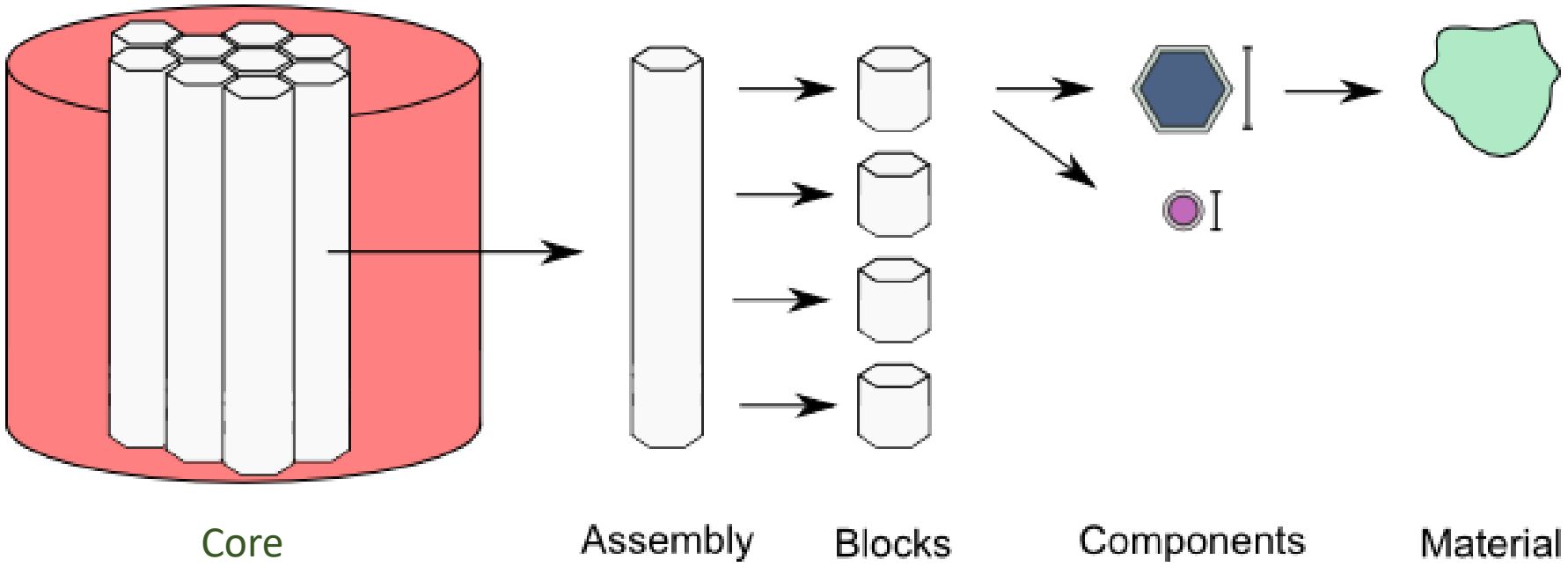
TerraPower

Composite Reactor Model



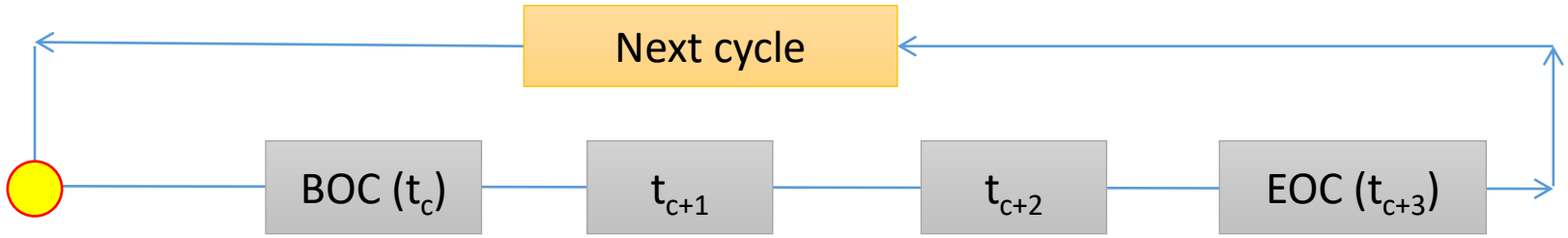
* Object types do not necessarily need to be the same at each level of the hierarchy, but in practice they are, at least for Core models.

Composite Reactor Model



Interactive Exploration of the Reactor Model

The Operator Interface Stack



Shuffling	Move fuel			
Cross-sections	Update			
Flux/power	Compute	Compute	Compute	Compute
Temp./pressure	Compute	Compute	Compute	Compute
Fuel performance				Update
Database	Save state	Save state	Save state	Save state

Example Interface Stack

Index	Type	Name	Function	Enabled	EOL order	BOL forced
01	Main	main		Yes	Reversed	No
02	FissionProductModel	fissionProducts		Yes	Normal	No
03	CrossSectionGroupManager	xsGroups		Yes	Normal	No
04	EquilibriumShuffler	fuelHandler		Yes	Normal	No
05	Depletion	depletion		Yes	Normal	Yes
06	Mc2v3	mc2v3	latticePhysics	Yes	Normal	No
07	Dif3d	dif3d	globalFlux	Yes	Normal	Yes
08	HistoryTracker	history		Yes	Normal	No
09	Report	report		Yes	Normal	No
10	Database	database		No	Normal	No
11	MemoryProfiler	memoryProfiler		Yes	Normal	No
12	Snapshot	snapshot		Yes	Normal	No

Interactive state modifications

Simple fuel management input leads to rich reload optimization capabilities

```
from armi.physics.fuelCycle import fuelHandlers
fh = fuelHandlers.fuelHandlerFactory(o)
```

```
moved = []
for n in range(10):
    high = fh.findAssembly(param="power", compareTo=1.0e6, blockLevelMax=True, exclusions=moved)
    low = fh.findAssembly(param="power", compareTo=0.0, blockLevelMax=True, exclusions=moved)
    fh.swapAssemblies(high, low)
    moved.extend([high, low])
```

Interactive fuel management

Database I/O

- The database maps reactor state (parameters) to persistent storage
- HDF5 format
- User can restore state from any time node from a database for follow-on analysis
- Data can be visualized using Paraview (VTK or XDMF) or VisIT (VTK only)

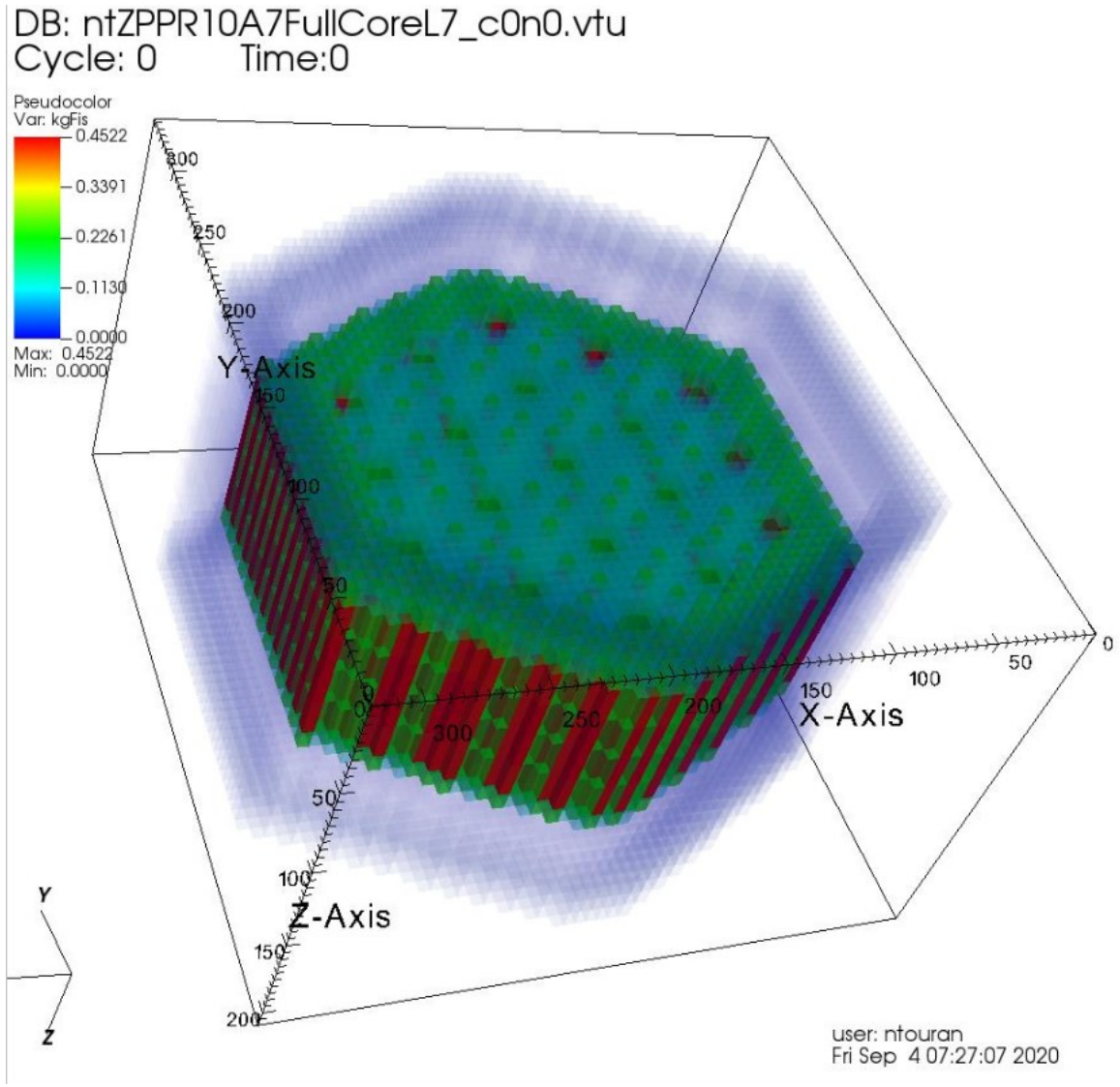
Interactive DB writing

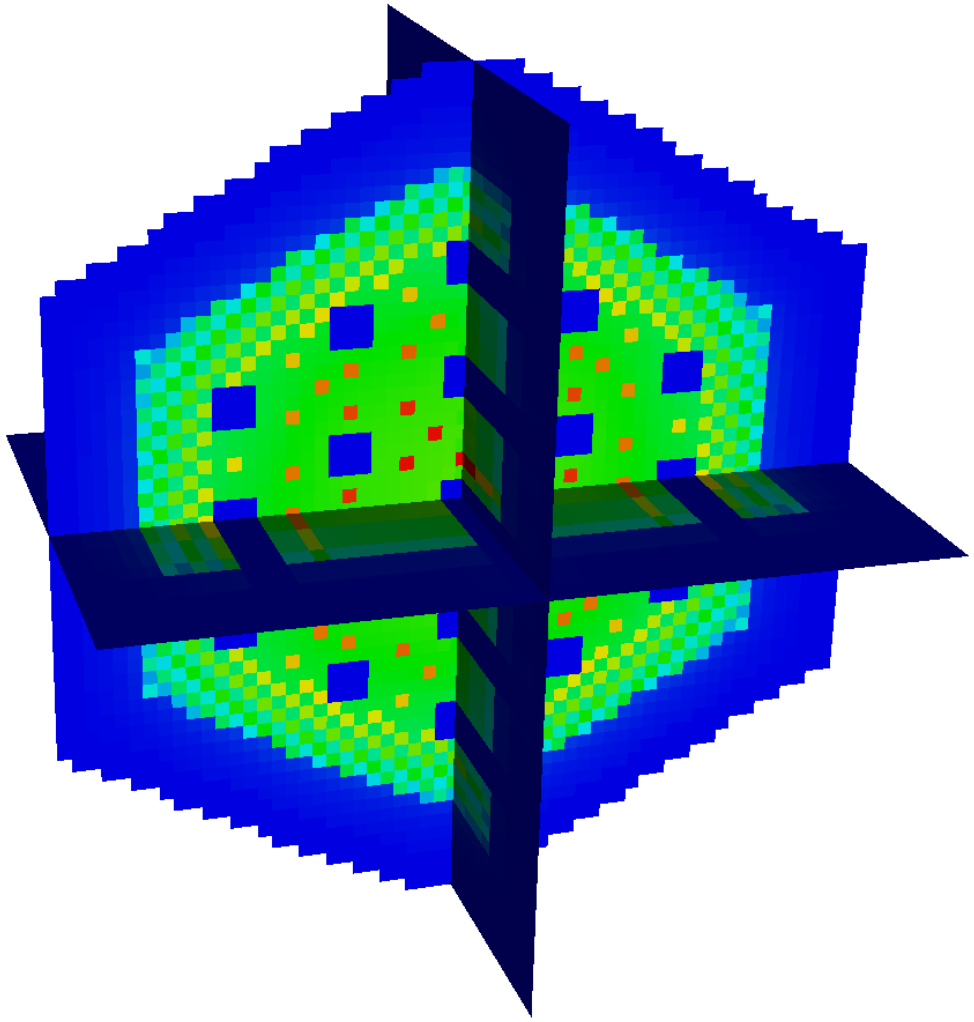
TerraPower

Loading output in Paraview

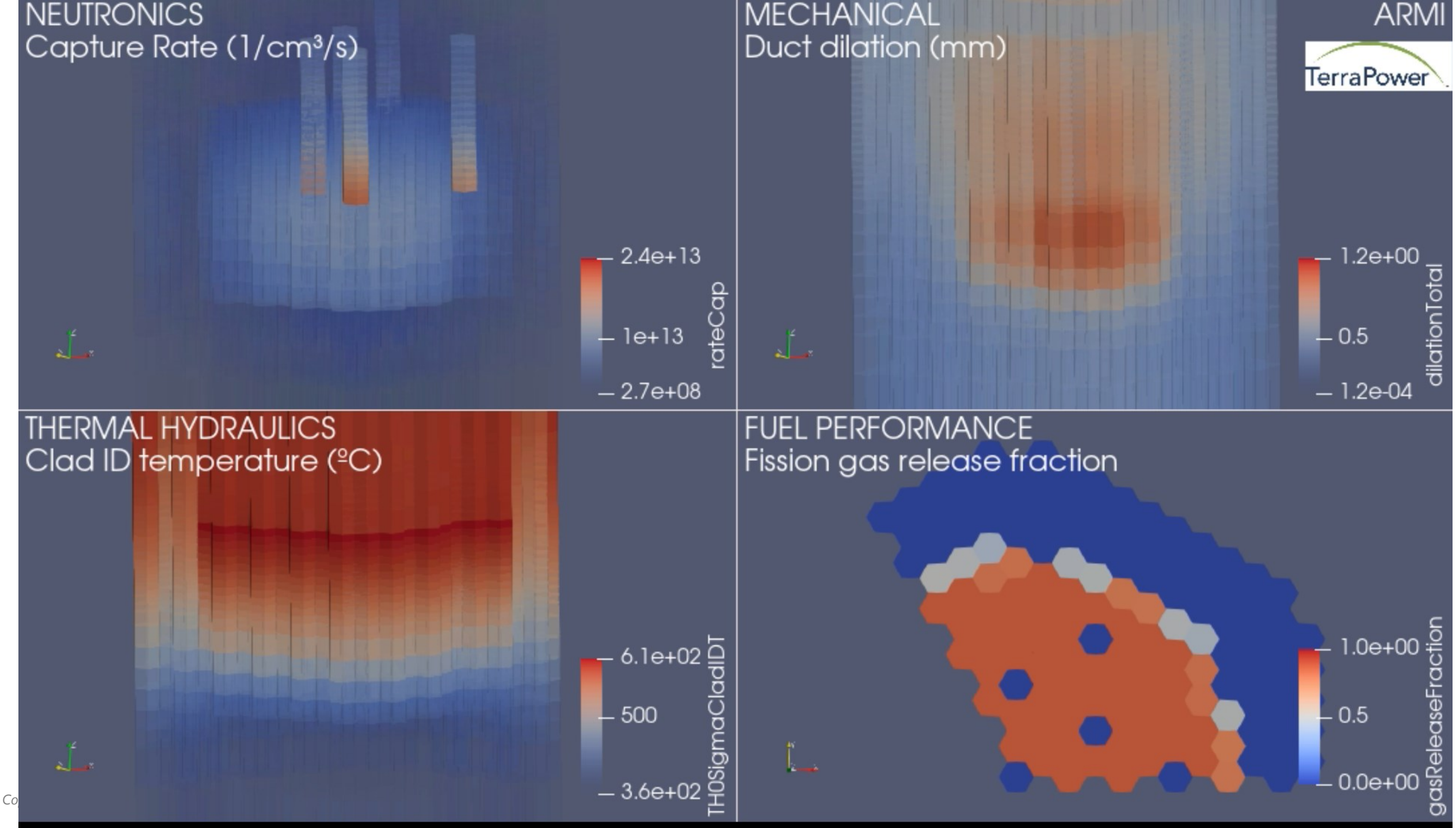
TerraPower

VTK and XDMF output support



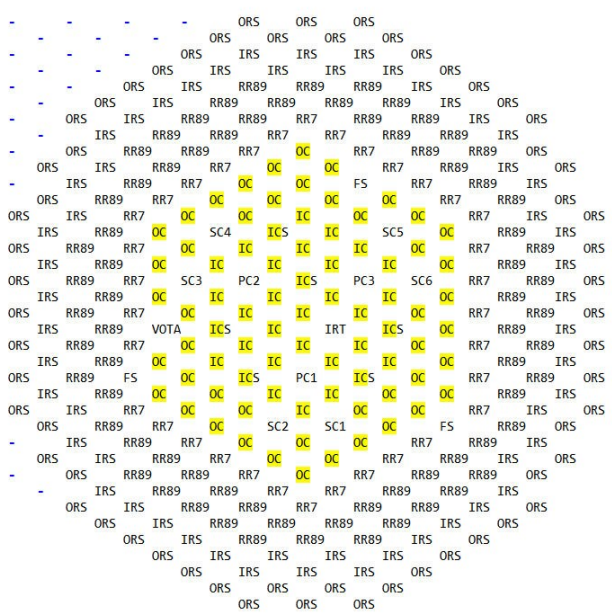


TerraPower

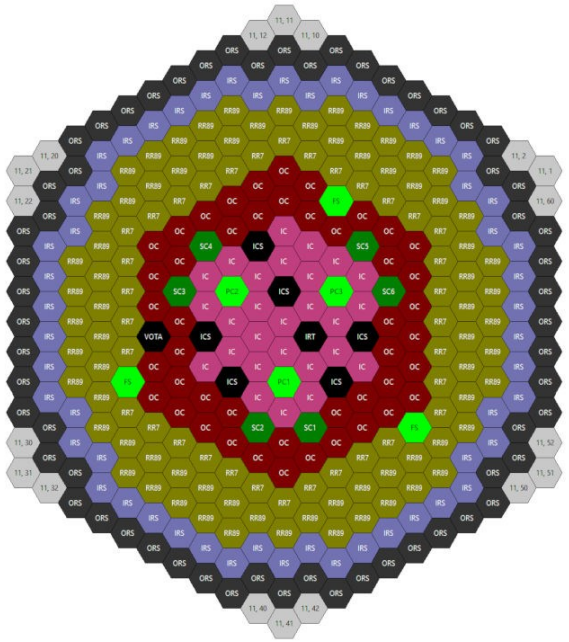

NEUTRONICS
Capture Rate (1/cm³/s)
2.4e+13
1e+13
2.7e+08
rateCap
MECHANICAL
Duct dilation (mm)
ARMI
TerraPower
1.2e+00
0.5
1.2e-04
dilationTotal
THERMAL HYDRAULICS
Clad ID temperature (ºC)
6.1e+02
500
3.6e+02
TH0SigmaCladIDT
FUEL PERFORMANCE
Fission gas release fraction
1.0e+00
0.5
0.0e+00
gasReleaseFraction

Once again with the FFTF isothermal benchmark model

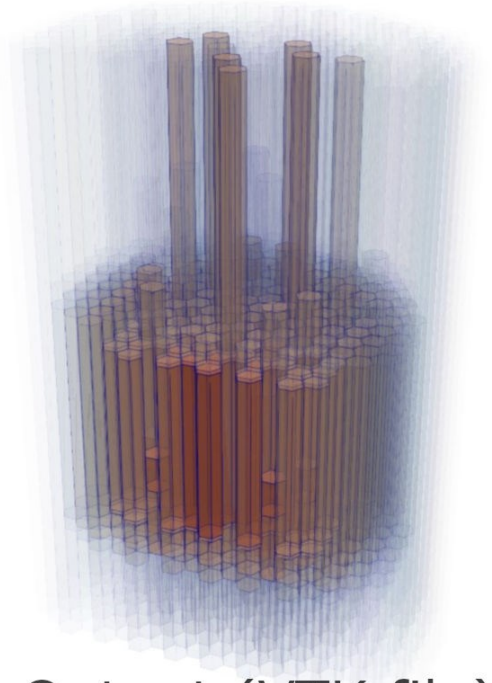
https://github.com/terrapower/fftf-isothermal-model



Input File



Grid Editor GUI



Output (VTK file)

TerraPower

Our plans, hopes, and dreams

- Release FFTF isothermal model (done)
- Release global flux plugin to interface with DONJON (in progress)
- Build open source DIF3D plugin (done)
- Build open source MC2 plugin (planned)

- What we hope others might do with ARMI:
  - Make plugin for other codes
  - Make ARMI model for BEAVRS LWR (maybe better for us to do)
  - Do awesome research
- Shout-out: Lots of other open-source nuclear work going on

  List here: https://github.com/paulromano/awesome-nuclear

Summary and Next Steps

ARMI's differentiators:

- Unapologetically nuclear specific
  - Unique reactor data model
- Made by reactor analysts for reactor analysts
- 14 years of professional use (mostly targeting SFR)
- High-level language
- Open source

Getting started:

- Try out the other online ARMI tutorials
- Obtain or make a ARMI reactor model you care about
- Make an ARMI plugin for your favorite physics code
- Make your own ARMI app for your specific needs
- Make a PR!!! ☺

Getting help:

- Github issue
- Email us (armi-devs@terrapower.com)
- (We'll make a forum soon)

https://terrapower.github.io/armi

Thank you for your time!

Discussion/Q&A

TerraPower