

Creating ORIGIN Reactor Libraries with TRITON

SCALE 6.3

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

Learning goals

- **Understand** difference between ORIGEN cross section library and ORIGEN reactor library
- **Understand** where to find SCALE's ORIGEN reactor library
- **Understand** what to use ORIGEN reactor libraries for
- **Learn** how to generate an ORIGEN reactor library
- **Learn** how to use a self-generated ORIGEN reactor library

ORIGEN cross section vs. reactor libraries



What are ORIGIN cross section vs. reactor libraries?

ORIGIN cross section library

One F33 file containing burnup-dependent 1-group cross sections for one specific assembly design.

Note common terminology:

ORIGIN cross section library
= *ORIGIN library*
= *ORIGIN 1-group library*
= *F33 file*

ORIGIN reactor library

A **set** of ORIGIN cross section libraries for a specific reactor type and fuel assembly configuration, parametrized to, for example, the following:

- Enrichment
- Moderator density
- Pu content, ^{239}Pu values (MOX fuel)

Example

g10_e40w03.f33

One file corresponding to a specific BWR GE 10×10 fuel assembly model: 4.0 wt.% ^{235}U enrichment and 0.3 g/cm³ coolant density

Example

g10_e05w01.f33	...
g10_e05w03.f33	g10_e80w07.f33
g10_e05w07.f33	g10_e80w09.f33
g10_e05w09.f33	g10_e85w01.f33
g10_e15w01.f33	g10_e85w05.f33
g10_e15w03.f33	g10_e85w07.f33
...	g10_e85w09.f33

A set of files for a BWR GE 10×10 fuel assembly for different combinations of enrichment and moderator density

What are these libraries used for?

ORIGEN cross section library

One F33 file containing burnup-dependent 1-group cross sections for one specific assembly design.

Note common terminology:

ORIGEN cross section library
= *ORIGEN library*
= *ORIGEN 1-group library*
= *F33 file*

ORIGEN reactor library

A **set** of ORIGEN cross section libraries for a specific reactor type and fuel assembly configuration, parametrized to, for example, the following:

- Enrichment
- Moderator density
- Pu content, ^{239}Pu values (MOX fuel)

ORIGEN

Used by ORIGEN to run a single depletion calculation of a specific problem to obtain the time-dependent inventory for this exact problem.

ORIGAMI

Used by ORIGAMI to rapidly run a depletion calculation and generate inventory for varying parameters.

User input: reactor library name together with desired enrichment/moderator density and depletion history

ORIGAMI calculation: interpolation of cross sections from pre-generated libraries followed by depletion to generate inventory

What data do ORIGEN cross section libraries contain?

An ORIGEN cross section library stores the following data:

Problem-**independent** data:

- Decay constants
- Molar masses
- Recoverable energy from decay (Q)
- Recoverable energy from photons
- Radiotoxicity factors
- Energy released per:
 - Fission event
 - Non-fission capture event

Problem-**dependent** data*:

- Transition coefficient matrix
- Removal/loss cross sections (1-group)
- Fission cross sections (1-group)
- Effective neutron yields (1-group)

*Problem-dependent data are stored at each burnup / time step on the library.

How are ORIGEN cross section libraries generated?

- In short: with TRITON and with Polaris
- TRITON and Polaris generate:
 - 1 ORIGEN cross section library for each depleted mixture
 - 1 ORIGEN cross section library for the average of all depletion mixtures in the problem
- 1 ORIGEN cross section library stores the data for 1 specific model—the model used in the TRITON/Polaris calculation (i.e., specific enrichment, moderator density, etc.)

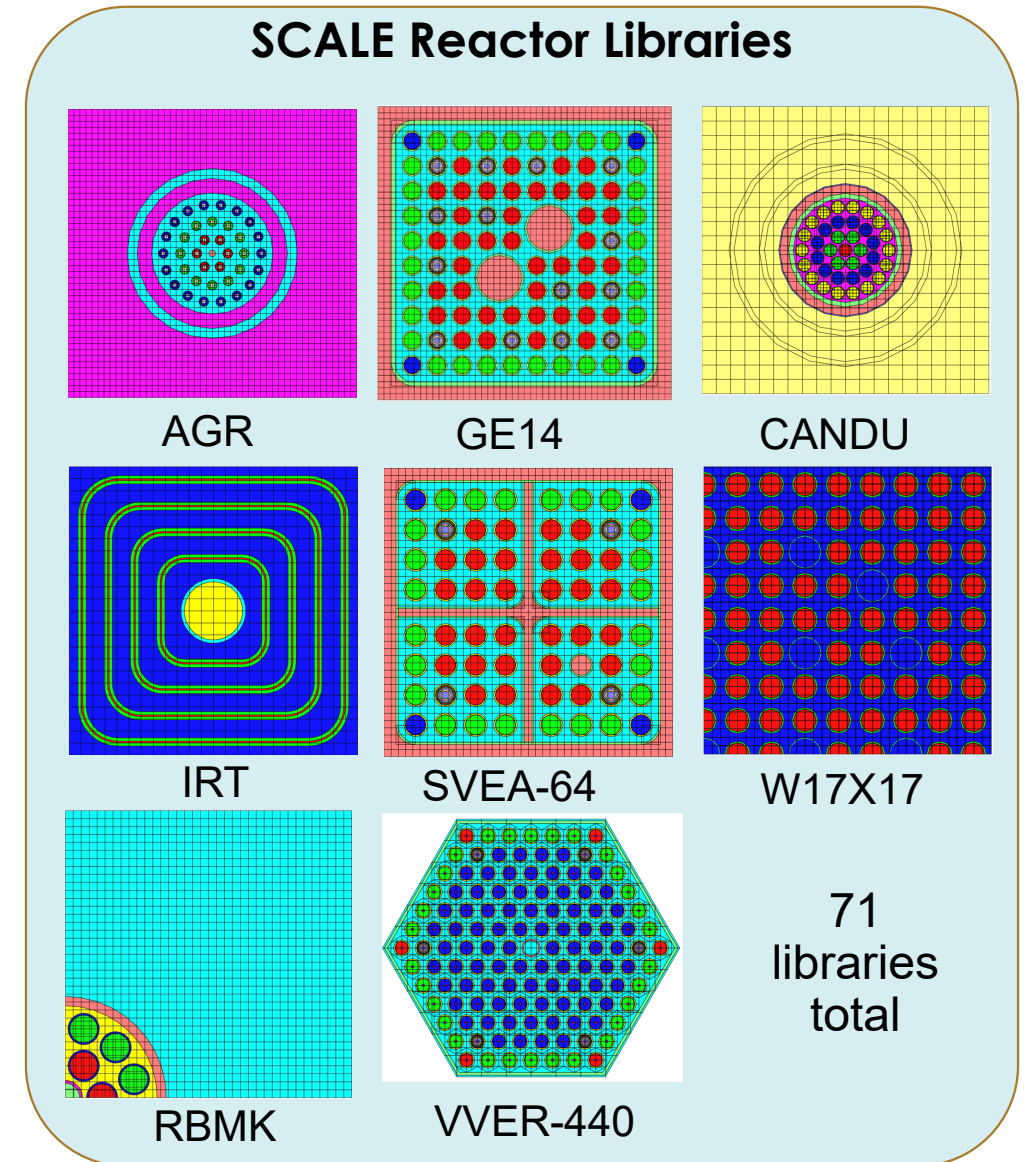
Multiple TRITON/Polaris calculations of a model with varying input parameters (enrichment, moderator density, etc.) are run to generate a **set of ORIGEN cross section libraries** that correspond to a **reactor library**.

ORIGEN reactor libraries available with SCALE



SCALE's ORIGEN reactor libraries for LWR fuel

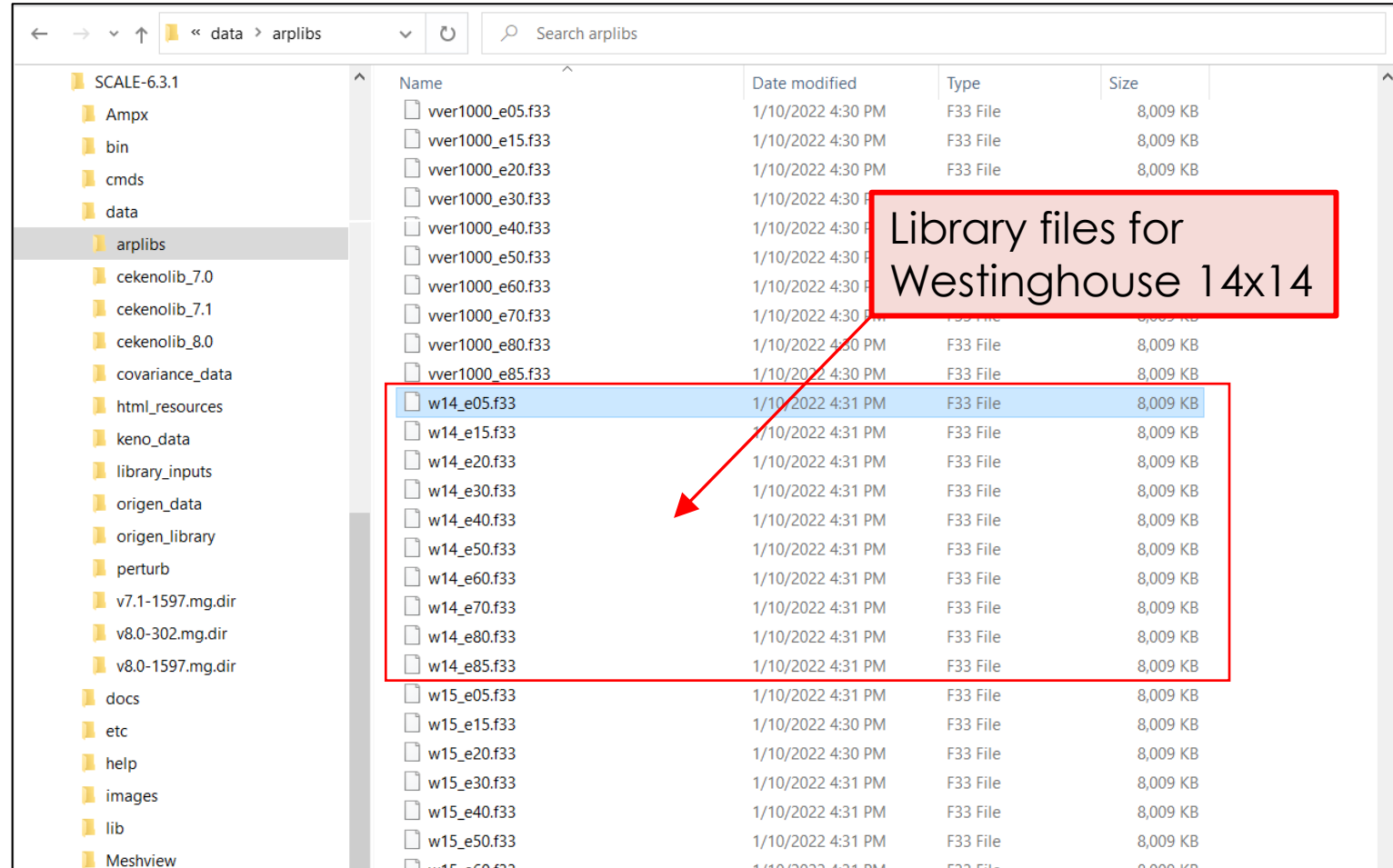
- The generation of ORIGEN reactor libraries can require significant computation time, depending on the complexity of the model and the needed parametrization
- Good news: We already used TRITON to generate a large number of ORIGEN reactor libraries for various LWR designs for you
- These reactor libraries cover a wide spectrum of LWR fuel designs and operating conditions for a range of discrete values for the following parameters:
 - Enrichment
 - Moderator density
 - Pu content, ^{239}Pu values (MOX fuel)
 - Burnup
- They are available in SCALE's DATA directory



SCALE's ORIGIN reactor libraries – Location

Fuel/reactor-specific ORIGIN
Reactor Libraries reside in
SCALE's DATA directory:

SCALE-6.3.3\data\arplib



Name	Date modified	Type	Size
vver1000_e05.f33	1/10/2022 4:30 PM	F33 File	8,009 KB
vver1000_e15.f33	1/10/2022 4:30 PM	F33 File	8,009 KB
vver1000_e20.f33	1/10/2022 4:30 PM	F33 File	8,009 KB
vver1000_e30.f33	1/10/2022 4:30 PM	F33 File	8,009 KB
vver1000_e40.f33	1/10/2022 4:30 PM	F33 File	8,009 KB
vver1000_e50.f33	1/10/2022 4:30 PM	F33 File	8,009 KB
vver1000_e60.f33	1/10/2022 4:30 PM	F33 File	8,009 KB
vver1000_e70.f33	1/10/2022 4:30 PM	F33 File	8,009 KB
vver1000_e80.f33	1/10/2022 4:30 PM	F33 File	8,009 KB
vver1000_e85.f33	1/10/2022 4:30 PM	F33 File	8,009 KB
w14_e05.f33	1/10/2022 4:31 PM	F33 File	8,009 KB
w14_e15.f33	1/10/2022 4:31 PM	F33 File	8,009 KB
w14_e20.f33	1/10/2022 4:31 PM	F33 File	8,009 KB
w14_e30.f33	1/10/2022 4:31 PM	F33 File	8,009 KB
w14_e40.f33	1/10/2022 4:31 PM	F33 File	8,009 KB
w14_e50.f33	1/10/2022 4:31 PM	F33 File	8,009 KB
w14_e60.f33	1/10/2022 4:31 PM	F33 File	8,009 KB
w14_e70.f33	1/10/2022 4:31 PM	F33 File	8,009 KB
w14_e80.f33	1/10/2022 4:31 PM	F33 File	8,009 KB
w14_e85.f33	1/10/2022 4:31 PM	F33 File	8,009 KB
w15_e05.f33	1/10/2022 4:31 PM	F33 File	8,009 KB
w15_e15.f33	1/10/2022 4:30 PM	F33 File	8,009 KB
w15_e20.f33	1/10/2022 4:30 PM	F33 File	8,009 KB
w15_e30.f33	1/10/2022 4:31 PM	F33 File	8,009 KB
w15_e40.f33	1/10/2022 4:31 PM	F33 File	8,009 KB
w15_e50.f33	1/10/2022 4:31 PM	F33 File	8,009 KB
w15_e60.f33	1/10/2022 4:31 PM	F33 File	8,009 KB

SCALE's ORIGIN reactor libraries – Control file

- Information about all ORIGIN reactor libraries is provided in file **arpdata.txt**
- File **arpdata.txt** is located in **SCALE-6.3.3\data**
- File **arpdata.txt** contains information on
 - Fuel type (**name** of the reactor library)
 - Number of values for each variable parameter
 - Parameter values
 - Burnup values for each library position
 - Filenames for parameter-dependent ORIGIN cross section libraries
- **This file tells ORIGAMI where to find the reactor libraries**

Information in *arpdata.txt* for a **PWR** reactor library

library name

enrichments, in wt% ²³⁵U

water density, in g/cm³

names of library files

burnup values, in MWd/MTU

```
!ce14x14
```

```
7      1      16
```

```
0.5 1.5 2.0 3.0 4.0 5.0 6.0
```

```
0.7332
```

```
'ce14_e05.f33' 'ce14_e15.f33' 'ce14_e20.f33'
```

```
'ce14_e30.f33' 'ce14_e40.f33' 'ce14_e50.f33'
```

```
'ce14_e60.f33'
```

```
0.0 1040.0 3000.0 5000.0 7500.0 10500.0
```

```
13500.0 16500.0 22500.0 28500.0 34500.0 40500.0
```

```
46500.0 58500.0 67500.0 70500.0
```

Information in *arpdata.txt* for a **BWR** reactor library

library name

enrichments in wt% ²³⁵U

water densities, in g/cm³

names of library files

Ordering of file names:

First enrichment, then moderator density:

1. Take first enrichment, list all combinations with moderator densities
2. Take second enrichment, list all combinations with moderator densities
3. ...

burnup values, in MWd/MTU

!ge10x10-8

7 5 23

0.5000 1.5000 2.0000 3.0000 4.0000 5.0000 6.0000

0.1000 0.3000 0.5000 0.7000 0.9000

'g10_e05w01.f33' 'g10_e05w03.f33' 'g10_e05w05.f33'

'g10_e05w07.f33' 'g10_e05w09.f33' 'g10_e15w01.f33'

'g10_e15w03.f33' 'g10_e15w05.f33' 'g10_e15w07.f33'

'g10_e15w09.f33' 'g10_e20w01.f33' 'g10_e20w03.f33'

'g10_e20w05.f33' 'g10_e20w07.f33' 'g10_e20w09.f33'

'g10_e30w01.f33' 'g10_e30w03.f33' 'g10_e30w05.f33'

'g10_e30w07.f33' 'g10_e30w09.f33' 'g10_e40w01.f33'

'g10_e40w03.f33' 'g10_e40w05.f33' 'g10_e40w07.f33'

'g10_e40w09.f33' 'g10_e50w01.f33' 'g10_e50w03.f33'

'g10_e50w05.f33' 'g10_e50w07.f33' 'g10_e50w09.f33'

'g10_e60w01.f33' 'g10_e60w03.f33' 'g10_e60w05.f33'

'g10_e60w07.f33' 'g10_e60w09.f33'

0.00 525.00 1500.00 2500.00 3750.00

5250.00 6750.00 8250.00 9750.00 11250.00

12750.00 15750.00 18750.00 20250.00 22500.00

31500.00 34500.00 40500.00 46500.00 52500.00

58500.00 67500.00 70500.00

Information in *arpdata.txt* for a **PWR MOX** reactor library

library name

Pu content (wt% Pu in heavy metal)

²³⁹Pu wt% ²³⁹Pu/Pu

water densities, in g/cm³

names of library files

Ordering of file names:

First water density, then ²³⁹Pu fraction, then Pu content:

1. Take first water density and first ²³⁹Pu fraction, list all combinations with Pu contents
2. Take first water density and second ²³⁹Pu fraction, list all combinations with Pu contents
3. ...

burnup values in MWd/MTU

```
!mox_w17x17
```

```
3      5      1      1      31
```

```
4.0     7.0     10.0
```

```
50.0 55.0 60.0 65.0 70.0
```

```
1.0
```

```
0.723
```

```
'mox_w17_e40v50.f33' 'mox_w17_e70v50.f33'
```

```
'mox_w17_e10v50.f33' 'mox_w17_e40v55.f33'
```

```
'mox_w17_e70v55.f33' 'mox_w17_e10v55.f33'
```

```
'mox_w17_e40v60.f33' 'mox_w17_e70v60.f33'
```

```
'mox_w17_e10v60.f33' 'mox_w17_e40v65.f33'
```

```
'mox_w17_e70v65.f33' 'mox_w17_e10v65.f33'
```

```
'mox_w17_e40v70.f33' 'mox_w17_e70v70.f33'
```

```
'mox_w17_e10v70.f33'
```

```
0.0 40.0 1040.0 3000.0 5000.0 7500.0
```

```
10500.0 13500.0 16500.0 19500.0 22500.0 25500.0
```

```
28500.0 31500.0 34500.0 37500.0 40500.0 43500.0
```

```
46500.0 49500.0 52500.0 55500.0 58500.0 61500.0
```

```
64500.0 67500.0 70500.0 73500.0 76500.0 79500.0
```

```
82500.0
```

Some notes on arpdata.txt

- Whereas *arpdata.txt* is located in **SCALE-6.3.3\data**, F33 files listed in *arpdata.txt* are located in **SCALE-6.3.3\data\arplibs**
- ORIGAMI must find *arpdata.txt* and the F33 files
- If libraries other than the libraries provided with SCALE are used, then *arpdata.txt* and F33 files must be copied to the temporary working directory (TMPDIR) so that SCALE finds the files (see later slide)
- The order of listed F33 matters; if unsure, check [Section 5.1.6.2](#) in the manual

Workshop: Use SSCALE's reactor libraries



Problem 1: Inventory of a PWR fuel assembly

- **Goal:** Use ORIGAMI to calculate inventory for **Westinghouse 15×15** fuel assembly

- **Steps:**

1. Create ORIGAMI input using ORIGAMI's UO2 express form:

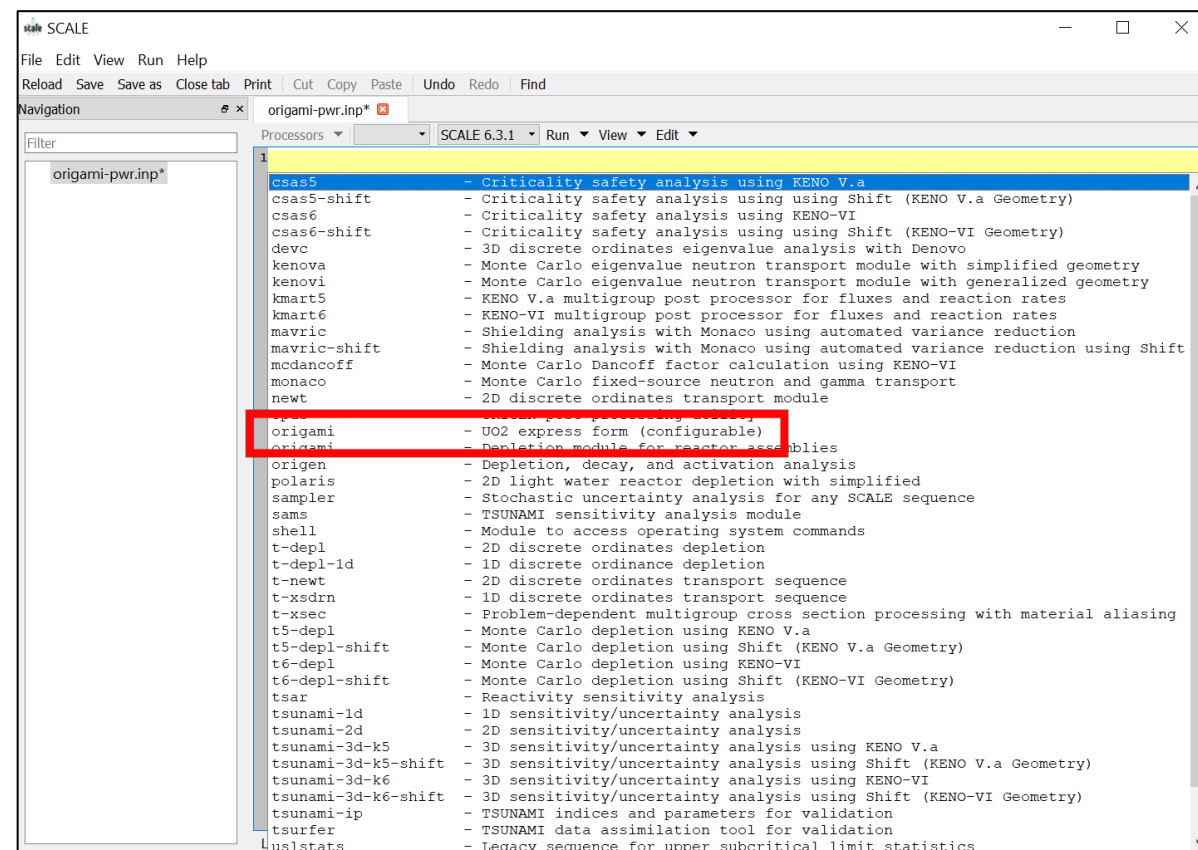
- | | |
|--|-------------------------------|
| • Title: "W15 from SCALE lib" | • Interpolations per cycle: 4 |
| • Fuel type: "w15x15" | • Cooling time: 1,000 days |
| • Fuel mass: 1 MTU | • Power history: 95% uptime |
| • Enrichment: 2.5 wt% ²³⁵ U | • Specific power: 40 MW/MTU |
| • Burnup: 35,000 MWd/MTU | |
| • Number of reactor cycles: 3 | |

2. Run ORIGAMI input

3. Analyze results using F71 concentration file: Plot the ²³⁵U and ²³⁹Pu masses (kg) as a function of time (days) during irradiation

- **Questions:**

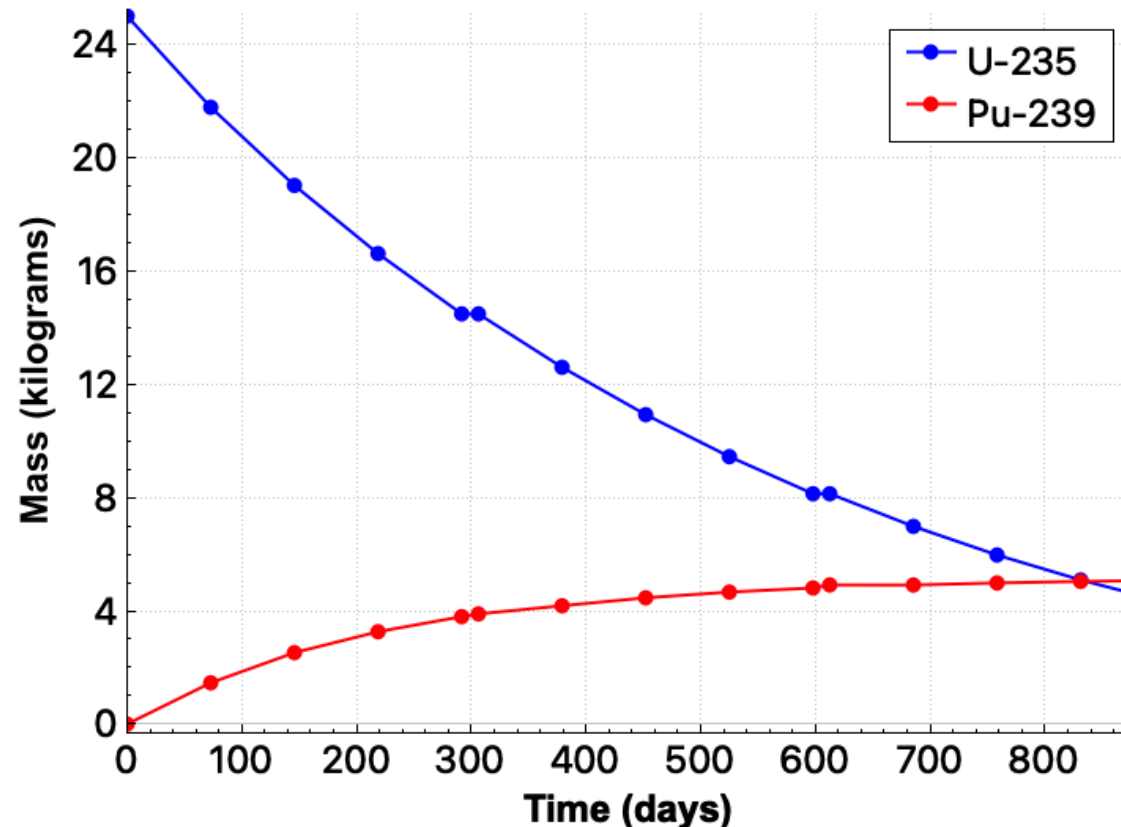
- How does ORIGAMI find the right library files?
- Which file is the key, and where is it located?
- How can you confirm which file was used?



**Auto-completion in Fulcrum offers
ORIGAMI UO2 express form**

Solution 1: Inventory of a PWR fuel assembly

Mass per 1 MTU in a PWR fuel assembly



origami - UO2 express form (configurable)

Title	W15 from SCALE lib
Fuel Type	w15x15
Uranium (MTU)	1.0
Enrichment (Wt%U235)	2.5
Burnup (MWd/MTU)	35000
Cycles	3
Number of Burnup Interpolations per Cycle	4
Cooling Time (days)	1000
Power History - Percent Up	95
Power History - Average Power (MW/MTU)	40
Moderator Density (g/cc)	0.7332

```
=origami
title="W15 from SCALE lib"
options{ mtu=1.0 ft71=all}
libs=[ "w15x15" ]
fuelcomp{
  uox(fuel){ enrich=2.5 }
  mix(1){ comps=[ fuel=100 ] }
}
modz=[ 0.7332 ]
pz=[ 1.0 ]
hist[
  cycle{ power=40 burn=291.67 nlib=4 down=14.58 }
  cycle{ power=40 burn=291.67 nlib=4 down=14.58 }
  cycle{ power=40 burn=291.67 nlib=4 down=0 }
  cycle{ down=1000 }
]
end
```

Name specified in "libs" keyword links to reactor library name in *arpdata.txt*!

Filled ORIGAMI UO2 express form

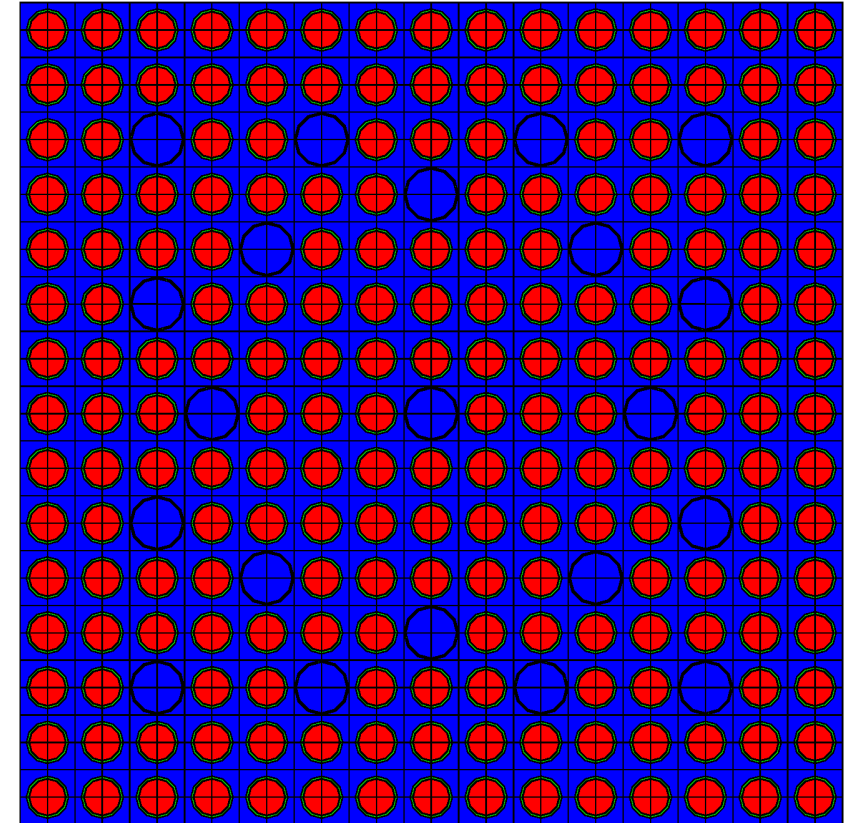
SCALE 6.3| Last Updated: October 2025

ORIGEN reactor library generation for LWRs and non-LWRs



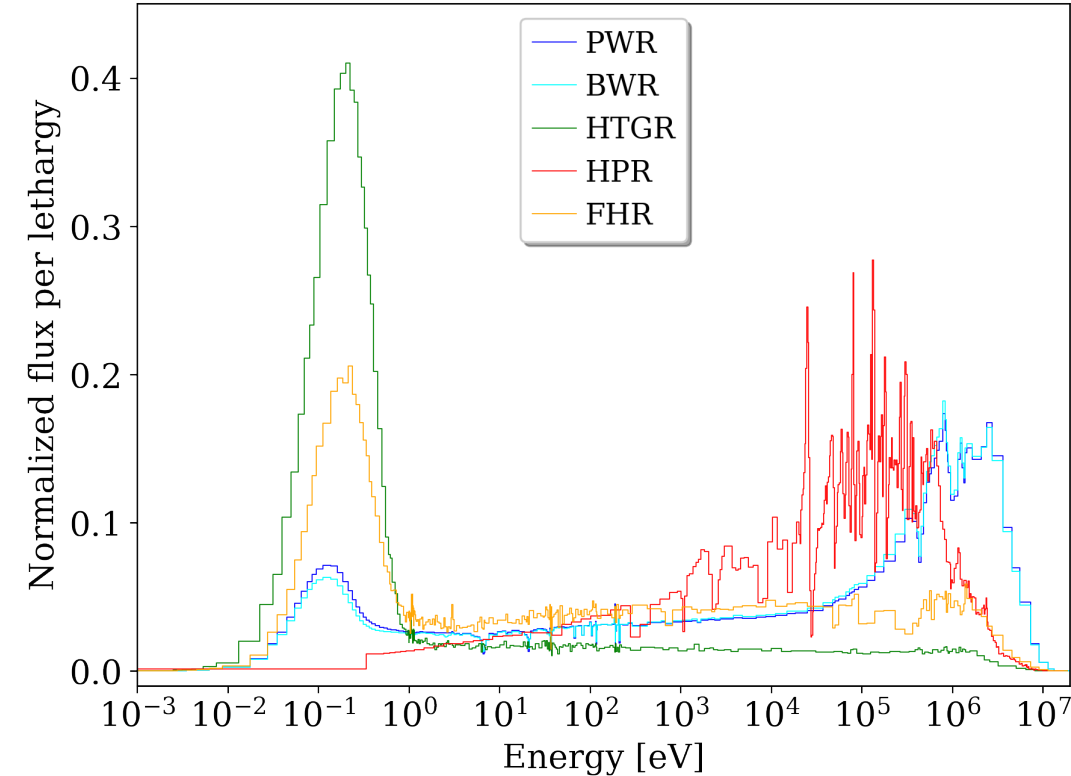
ORIGEN reactor library generation for **LWRs**

- The ORIGEN reactor libraries provided with SCALE 6.3 were generated with TRITON for LWRs using 2D LWR assembly models
- This is considered adequate given that the neutron flux within a reflected assembly model is representative of the neutron flux in the core
- Parametrization:
 - ^{235}U enrichments: 0.5-8.5 wt.%
 - SCALE 6.2.4: 0.5-6.0 wt.%
 - Moderator density / void fraction (BWRs)
 - Pu content, ^{239}Pu values (MOX fuel)
- Burnup: 0 to 82.5 GWd/MTU for most BWR and PWR assemblies
 - SCALE 6.2.4: 0 to 70.5 GWd/MTU



ORIGEN reactor library generation for **non-LWRs**

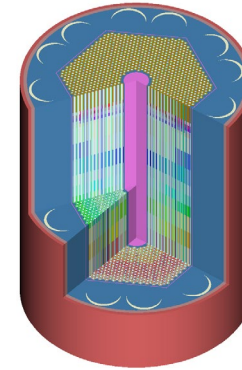
- Model for non-LWR ORIGEN reactor library generation:
 - The assumption that an assembly has a similar neutron flux spectrum as the core no longer holds in all non-LWRs
 - Systems with graphite as moderator (e.g., HTGRs, FHRs) have a much longer mean free path
 - Fast spectrum systems (e.g., SFRs, HPRs) are relatively transparent to neutrons so that neutron coupling is found across the reactor
 - Solution: Find other surrogate models for library generation



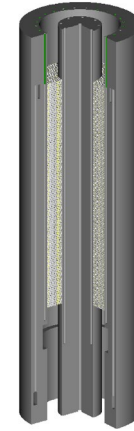
Comparison of neutron flux between different reactor types

ORIGEN reactor library generation for **non-LWRs**

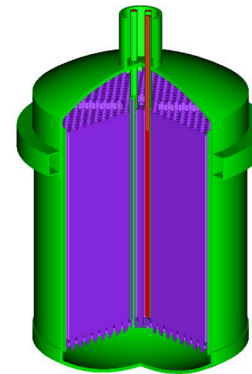
- Parametrization for non-LWRs:
 - Not all of the standard parameters are relevant for non-LWRs
 - Fast spectrum reactors do not have moderators
 - Graphite-moderated reactors don't need moderator density parametrization
 - Other parameters are relevant
 - Fast spectrum reactors are sensitive to radial and axial expansion
 - Graphite-moderated reactors show large spectral differences over their radial zones
 - Graphite-moderated reactors are sensitive to fuel and reflector temperatures
 - Solution:
 - Determine dominant parameters that influence 1-group cross sections
 - Generate ORIGEN reactor libraries for these parameters using appropriate surrogate model



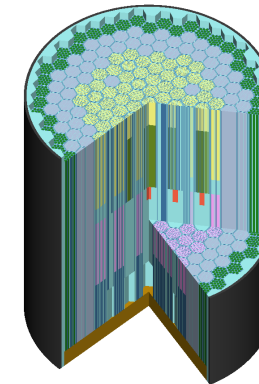
Heat pipe reactor



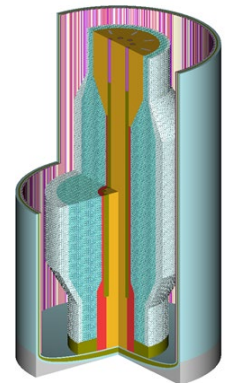
High temperature gas-cooled reactor



Molten salt reactor



Sodium-cooled fast reactor



Fluoride salt-cooled high temperature reactor

ORIGEN reactor library generation for **non-LWRs**

- Notes on the use of self-generated non-LWR reactor libraries:
 - ORIGAMI up to SCALE 6.3 is focused on LWR analysis:
 - The format in *arpdata.txt* only permits limited parameters: enrichment, moderator density, Pu content (for MOX fuel), ^{239}Pu values (for MOX fuel)
 - ORIGAMI uses these fixed parameters
 - ORIGAMI does not care how F33 files were generated: for example, if the non-LWR libraries are only parametrized according to enrichment, then ORIGAMI in SCALE 6.3 can be used
 - ORIGAMI in SCALE 7.0 will support additional non-LWR capabilities
 - New library archive with user-defined parametrization
 - New pebble-bed reactor capability

Generation of your own ORIGEN reactor library



ORIGEN cross section library generation with TRITON

- ORIGEN cross section libraries are automatically generated in all TRITON sequences and are copied to the output directory
- TRITON generates:
 - One ORIGEN cross section library for each depleted mixture
 - One ORIGEN cross section library for the average of all depletion mixtures in the problem
- The libraries are created with the following nomenclature:
 - For each depletion mixture, file `${BASENAME}.mixXXXX` is created, where `XXXX` is the mixture number, e.g., `${BASENAME}.mix0007.f33` for mixture 7
 - For the system-averaged cross sections, file `${BASENAME}.system.f33` is created
 - This system-average file is the file we are interested in!

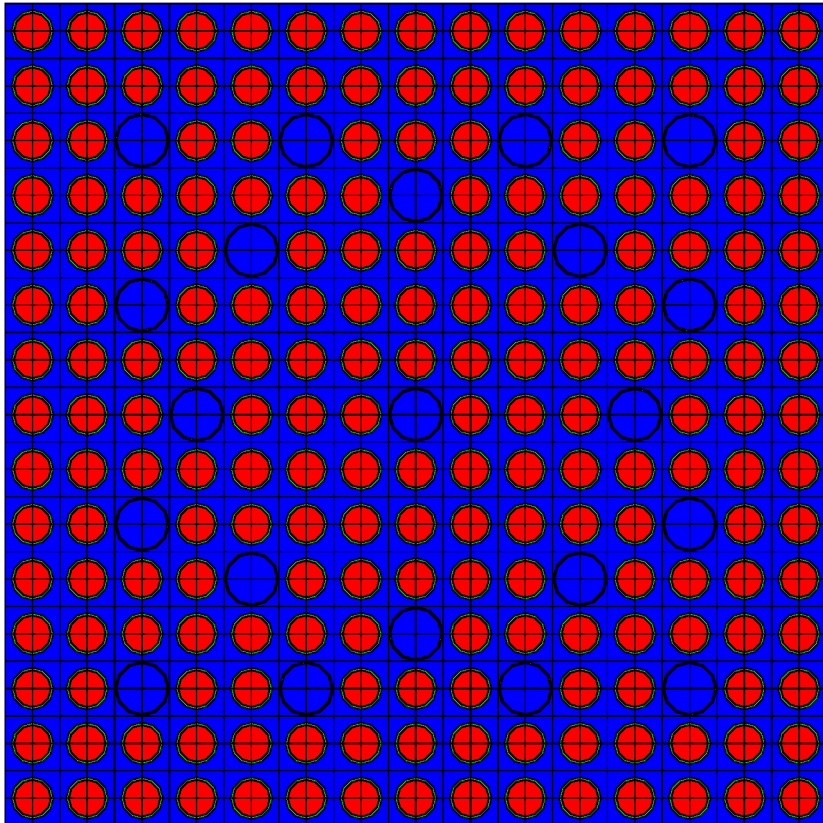
Note that in SCALE 6.2 and prior releases, f33 files were not automatically copied back to the output directory. Files with name “ft33f001.cmbined” were created in TMPDIR and had to be manually copied back.

Steps for LWR ORIGEN reactor library generation

1. Set up a **template TRITON input file** for the desired reactor design with generic flags for parameters to be changed
2. Generate **specific TRITON input files** for each parameter combination
 - For example, if 3 enrichments and 3 moderator densities are desired, then 9 TRITON inputs in total are needed
3. Run SCALE with each TRITON input file and collect output F33 files
4. Create your own *arpdata.txt* file with new reactor library information

Step-by-step example

- Goal: Generate ORIGEN libraries for a W15 × 15 assembly



Enrichments: 1.5, 2.0, 3.0, 4.0, 5.0 wt% ^{235}U

Moderator density: 0.7135 g/cm³

Burnup: 0 to 72 GWd/MTU

Number of burnup steps: 8

Step-by-step example

- Step 1
 - Create a TRITON template input file: **w15_template.i**
 - Use generic parameter flags for fuel composition data:

```
read comp
  uo2 1 0.95 923 92235 conc_u235
                        92238 conc_u238 end
  ...
end comp
```

← parameter for ^{235}U concentration

← parameter for ^{238}U concentration

Step-by-step example

- Step 2

- Generate five specific TRITON input files:
 - One input file corresponding to one specific enrichment
 - Input files based on the template input file **w15_template.i**

w15_e15.inp → 1.5 wt% ^{235}U enrichment

w15_e20.inp → 2.0 wt% ^{235}U enrichment

w15_e30.inp → 3.0 wt% ^{235}U enrichment

w15_e40.inp → 4.0 wt% ^{235}U enrichment

w15_e50.inp → 5.0 wt% ^{235}U enrichment

Step-by-step example

- Step 3

- Run SCALE with each of these five input files
- Five system-average library files are created in the output directory

w15_e15.system.f33

w15_e20.system.f33

w15_e30.system.f33

w15_e40.system.f33

w15_e50.system.f33

- Each F33 file contains data at burnup points according to the specification in TRITON's BURNDATA input

Step-by-step example

- Step 3 (continued)

- TRITON displays the “Library Burnup” in a table at the beginning of the output file, consistent with the settings in the BURNDATA block

TRITON input with 8 depletion steps

```
read burndata
  power=40 burn=1800 down=0 nlib=8 end
end burndata
```

TRITON output showing 9 library burnups

Sub-Interval No.	Depletion Interval	Sub-interval in interval	Specific Power (MW/MTIHM)	Burn Length (d)	Decay Length (d)	Library Burnup (MWd/MTIHM)
0	****Initial Bootstrap Calculation****					0.00000E+00
1	1	1	40.000	225.000	0.000	4.50000e+03
2	1	2	40.000	225.000	0.000	1.35000e+04
3	1	3	40.000	225.000	0.000	2.25000e+04
4	1	4	40.000	225.000	0.000	3.15000e+04
5	1	5	40.000	225.000	0.000	4.05000e+04
6	1	6	40.000	225.000	0.000	4.95000e+04
7	1	7	40.000	225.000	0.000	5.85000e+04
8	1	8	40.000	225.000	0.000	6.75000e+04

NOTE: Library Burnup is the cumulative burnup computed at the midpoint of the depletion sub-interval.
Specific Power and Library Burnup depend on basis material(s) selected in DEPLETION block.

Note that zero burnup is included: 8 depletion steps result in 9 burnup points.

Step-by-step example

Attention Windows users: Your explorer might hide file extensions. A common pitfall is creating a file with name "*arpdata.txt.txt*".

• Step 4

- Rename F33 libraries as desired (e.g., remove "system")
- Create new file *arpdata.txt* using the burnup information from the TRITON output

<code>!my_w15</code>	library name
<code>5 1 9</code>	5 enrichments, 1 moderator density, 9 burnups
<code>1.5 2.0 3.0 4.0 5.0</code>	enrichments
<code>0.7135</code>	moderator density
<code>'w15_e15.f33' 'w15_e20.f33' 'w15_e30.f33'</code>	names of library files
<code>'w15_e40.f33' 'w15_e50.f33'</code>	
<code>0 4500 13500 22500 31500</code>	burnups (in MWd/MTU)
<code>40500 49500 58500 67500</code>	

- Optional: Copy created libraries (`w15*.f33`) to new folder (e.g., "my_arplibs")
- **Warning:** Do not use a library name that is already used in SCALE's *arpdata.txt* file!

How to use your own *arpdata.txt* file

- In theory, you can modify file `SCALE-6.3.3\data\arpdata.txt` and add libraries to folder `SCALE-6.3.3\data\arplibs\`; however, this might not always be possible, or it might not be desired because of unintentional changes and overwrites
- Better option: Create your own *arpdata.txt* file and make SCALE use the correct data by copying the files to the temporary working directory (TMPDIR)
- SCALE looks for *arpdata.txt* in TMPDIR; it moves on to looking in the DATA directory only if it doesn't find it there

```
=shell
  cp ${INPDIR}/my_arpdata.txt  ${TMPDIR}/arpdata.txt
  mkdir  ${TMPDIR}/arplibs
  cp ${INPDIR}/my_folder/*f33  ${TMPDIR}/arplibs
end
=origami
...
end
```

← Make SCALE use
your *arpdata.txt*

← Copy your libraries to
expected location

ORIGEN reactor library generator (SLIG)

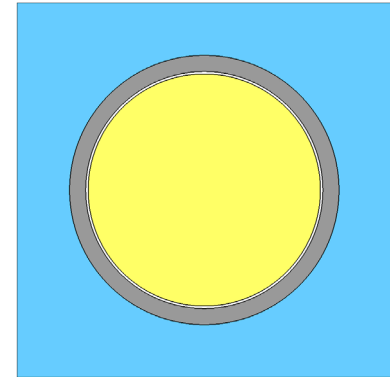
- You can generate libraries using the automated tool SLIG
 - Manual for SLIG can be found in [Section 5.3](#) of the SCALE manual
 - Package is located on `C:\SCALE-6.3.3\etc\slig`
 - **TRITON template files** are available for all libraries released with SCALE 6.3.3 on `C:\SCALE-6.3.3\etc\slig\testing\templates`
 - Generation of **actual TRITON input files** via execution of Python script as instructed in the manual

Workshop: Generate and use your own ORIGIN reactor library



Problem 2, Part 1: Generate your own LWR reactor library

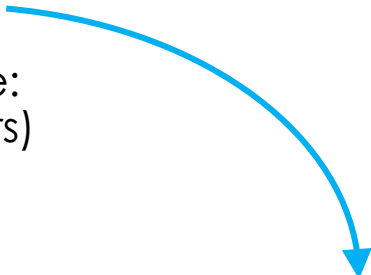
- **Goal:** Generate an ORIGEN reactor library to support rapid inventory calculations of >5 wt% ^{235}U enriched PWR models up to high burnup levels
- **Basis TRITON input:** “pwr_tdepl.inp”
 - Input is a complete TRITON-NEWT input for depletion of a PWR pin cell
 - Based on actual analysis work reported in ORNL/TM-2020/1833*
- **Steps:**
 1. Generate individual TRITON input files for the enrichments: 4, 6, 8, 10 wt% ^{235}U
 2. Run all input files and confirm presence of system-average F33 files
(Note that you can start multiple calculations at the same time; no need to wait for one calculation to be complete before starting the next one)
- **Analysis:**
 - Inspect the TRITON output to find the burnup steps of the F33 files
 - Use OBIWAN to extract ^{235}U , ^{239}Pu , and ^{240}Pu 1-group loss cross section from F33 files, and plot them as a function of burnup
 - *New in SCALE 6.3.3: Use Fulcrum to plot 1-group cross sections from F33 files*
 - Discuss whether the burnup steps are sufficiently fine

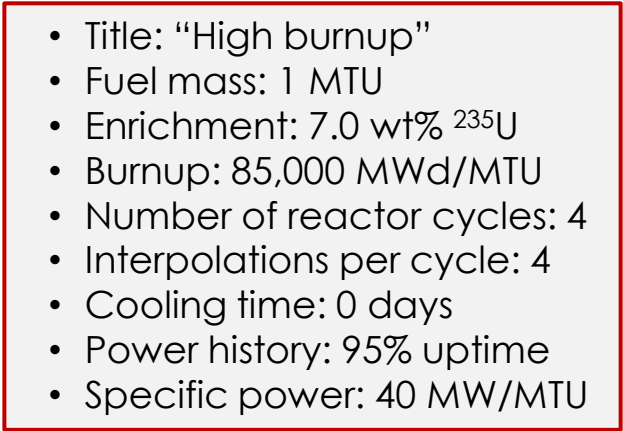


PWR fuel pin:
High enrichment
High burnup

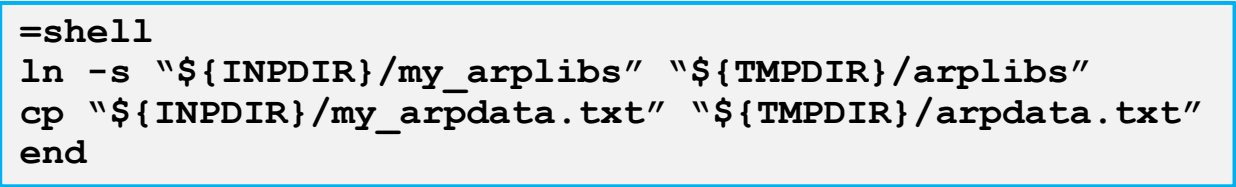
*Input adapted from: B. Hall, et al. (2021). "Isotopic and Fuel Lattice Parameter Trends in Extended Enrichment and Higher Burnup LWR Fuel Vol I: PWR fuel," ORNL/TM-2020/1833.
<https://code.ornl.gov/scale/analysis/haleu-hbu-latt-phys/>.

Problem 2, Part 2: Use your own LWR reactor library

- **Goal:** Use ORIGEN reactor library to run a rapid inventory calculation of 7 wt% ^{235}U enriched PWR models up to 85 GWd/MTU burnup
- **Steps:**
 1. Rename F33 files so that they have short names ("pwr_e04.f33", etc.) and move them into a directory "my_arplibs"
 2. Create a file "my_arpdata.txt" and fill in the information for your new PWR library
 - Initiate the file by copying the block for the "w17x17" library from the *arpdata.txt* file in SCALE's DATA directory
 - Give the library name "my_pwr"
 3. Create **ORIGAMI input** using ORIGAMI's UO2 express form 
 - What do you have to choose as fuel type?
 4. Add a **shell block** to copy your "my_arpdata.txt" file and your F33 files into the temporary working directory 
 5. Run ORIGAMI and analyze results using F71 concentration file:
Plot the ^{235}U and ^{239}Pu masses (kg) as a function of time (years)
- **Questions:**
 - How can you confirm that your file was used?
 - What is the maximum burnup that you can request in your ORIGAMI calculation?



- Title: "High burnup"
- Fuel mass: 1 MTU
- Enrichment: 7.0 wt% ^{235}U
- Burnup: 85,000 MWd/MTU
- Number of reactor cycles: 4
- Interpolations per cycle: 4
- Cooling time: 0 days
- Power history: 95% uptime
- Specific power: 40 MW/MTU



```
=shell
ln -s "${INPDIR}/my_arplibs" "${TMPDIR}/arplibs"
cp "${INPDIR}/my_arpdata.txt" "${TMPDIR}/arpdata.txt"
end
```

Solution 2, Part 1: Generate your own LWR reactor library

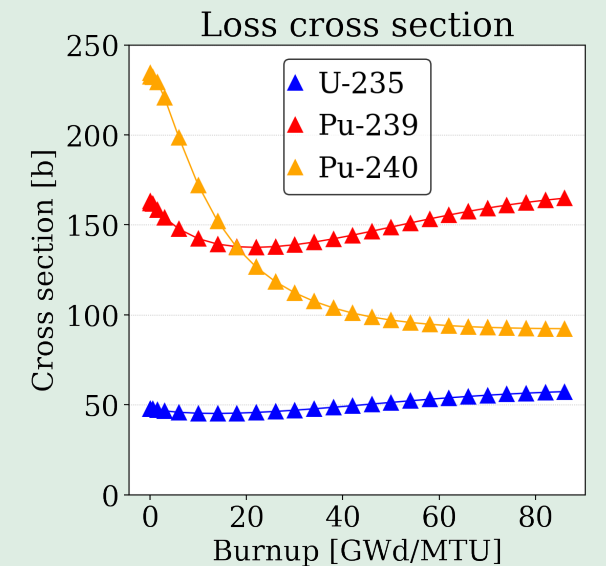
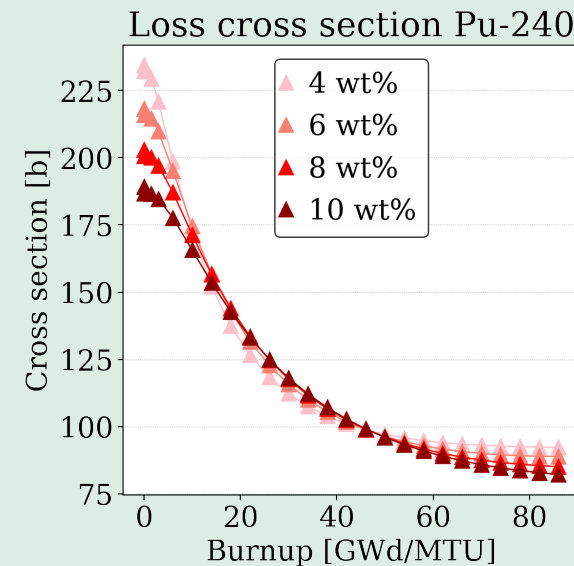
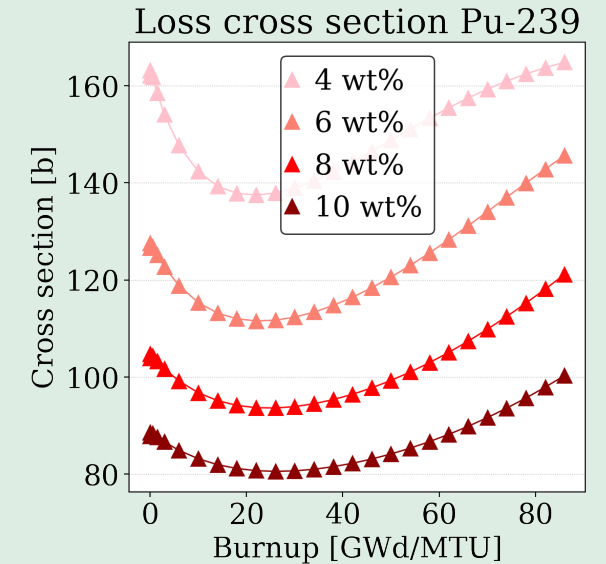
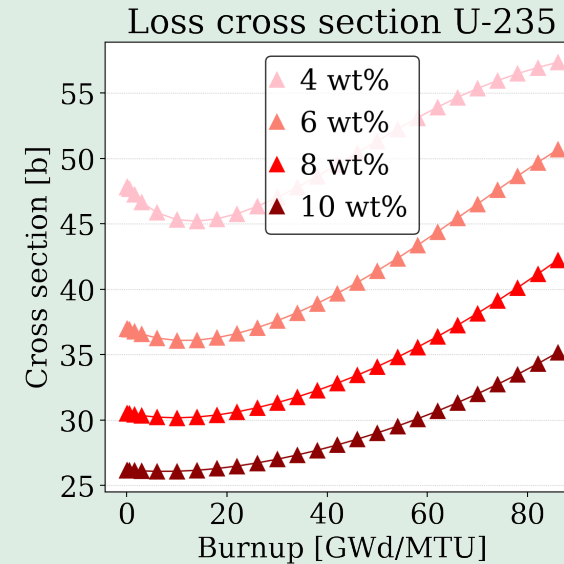
Overview of depletion calculation in TRITON output

Sub-Interval No.	Depletion Interval	Sub-interval in interval	...	Library Burnup (MWd/MTUHM)
0	****Initial Bootstrap Calculation****	1		0.00000E+00
1	1	1		5.00000e+01
2	1	2		5.50000e+02
3	1	3		1.50000e+03
4	1	4		3.00000e+03
5	1	5		6.00000e+03
6	1	6		1.00000e+04
7	1	7		1.40000e+04
8	1	8		1.80000e+04
9	1	9		2.20000e+04
10	1	10		2.60000e+04
11	1	11		3.00000e+04
12	1	12		3.40000e+04
13	1	13		3.80000e+04
14	1	14		4.20000e+04
15	1	15		4.60000e+04
16	1	16		5.00000e+04
17	1	17		5.40000e+04
18	1	18		5.80000e+04
19	1	19		6.20000e+04
20	1	20		6.60000e+04
21	1	21		7.00000e+04
22	1	22		7.40000e+04
23	1	23		7.80000e+04
24	1	24		8.20000e+04
25	1	25		8.60000e+04

Example OBIWAN command to extract loss cross section

```
obiwan view -format=loxs -symbols=1 pwr_e40.f33 | grep pu239
```

```
AC:pu239 1.0385e+02 1.0469e+02 ... 1.2111e+02
```



Solution 2, Part 2: Use your own LWR reactor library

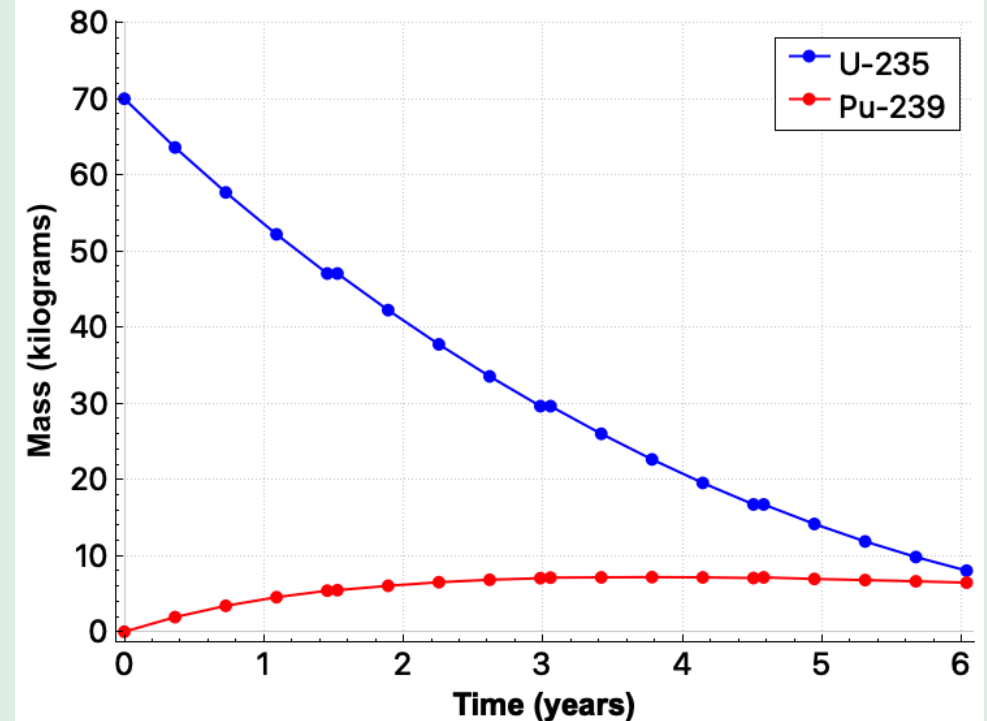
ORIGAMI input

```
=shell
ln -s "${INPDIR}/my_arplibs" "${TMPDIR}/arplibs"
cp "${INPDIR}/my_arpdata.txt" "${TMPDIR}/arpdata.txt"
end

=origami
title="High burnup"
options{ mtu=1.0 ft71=all}
libs=[ "my_pwr" ]
fuelcomp{
  uox(fuel){ enrich=7.0 }
  mix(1){ comps=[ fuel=100 ] }
}
modz=[ 0.7048 ]
pz=[ 1.0 ]
hist[
  cycle{ power=40 burn=531.25 nlib=4 down=26.56 }
  cycle{ power=40 burn=531.25 nlib=4 down=26.56 }
  cycle{ power=40 burn=531.25 nlib=4 down=26.56 }
  cycle{ power=40 burn=531.25 nlib=4 down=0 }
]
end
```

Your reactor library name
as you call it in arpdata.txt

Mass per 1 MTU in a PWR fuel assembly



ORIGAMI output

```
data library: my_pwr
directory: /private/var/folders/kr/swt5qtfx6bd0thw2fxrf9fbrb3718p/T/scale.f78.36245/arplibs
definition: /private/var/folders/kr/swt5qtfx6bd0thw2fxrf9fbrb3718p/T/scale.f78.36245/arpdata.txt
...
cycle time(d) power(MW) burnup(MWd/MTU) burnup(MWd/MTU) burnup(MWd/MTU) nlib
1 531.250 40.0000 21250.0 21250.0 10625.0 4
2 531.250 40.0000 21250.0 42500.0 31875.0 4
3 531.250 40.0000 21250.0 63750.0 53125.0 4
4 531.250 40.0000 21250.0 85000.0 74375.0 4
```

Location of arpdata.txt
correctly points to TMPDIR

Confirm requested
final burnup

Summary



ORIGEN reactor library generation – Summary

- **What are reactor libraries?**
 - They are a set of ORIGEN cross section libraries (F33 files) for a specific reactor type and fuel assembly configuration, parametrized to relevant design parameters
- **Which SCALE module uses reactor libraries?**
 - They are used by ORIGAMI for **rapid** depletion calculations (in seconds)
 - See ORIGAMI lecture
 - They are used by ARP for library interpolation
- **How are reactor libraries generated?**
 - F33 files are generated in depletion calculations of detailed models by TRITON and Polaris
 - File *arpdata.txt* lists all F33 files belonging to one reactor library, and it specifies a name so that ORIGAMI can find the F33 files
- **Why would I generate my own reactor libraries instead of just using the ones that are provided with SCALE?**
 - Reactor libraries need to be generated if the reactor/fuel type or the parameter space is not covered by SCALE's libraries
 - Examples: high burnup, high enrichment, test reactors, non-LWRs
- **What kind of model should I use when I generate my own reactor libraries?**
 - For LWRs, a 2D fuel assembly is usually sufficient
 - For non-LWRs or small reactors, find an appropriate surrogate model that represents the flux in the full core
 - Also, determine relevant parameters