

OBIWAN, ORIGEN's Helpful Command Line Utility

SCALE 6.3.3

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



U.S. DEPARTMENT OF
ENERGY

OBIWAN: “This **is** the tool you are looking for”

- Origen Binary Interrogation
Without A scale input
- A command-line tool for viewing, manipulating, and converting ORIGEN binary files
- OBIWAN is typically invoked directly from the command line as **`${SCALE}/bin/obiwan`** or in a SCALE input inside a `=shell` block
 - But it's actually not that hard because OBIWAN provides help screens every step of the way

```
(base) f78@mbp115895 ~ % obiwan hello

sydNNNNMMNs+//++++sNNNNNNNNMMNmdyso/:-----::+ohmNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNmmmmddh
ossydMMMh+///+/+odNNNdyo+:-----:/ydNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNmmdd
ooooohNMMh+++//++ydy/.-----+ymNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNmmmm
s+:-/odNd+//++++:-
:::yd+//++++:-sNNNNNNNNNNmNmmmmNNNNNNNN
++//+:--:/:-omNNNNNNNNNNmmNNNNNNNN
./+/---:+++/oNNNNNNNNNNNNNNNNNNNN
./-://osssss+:mNNNNNNNNNNNNNNNNNNNN
--:/+oossyyyhho..dNNNNNNNNNNNNNNNNmm
++:/+osysysyhddds:-mNNNNNNMMNNNNNNNNNNmd
++:/+oyyshoyyydhyyo/.yNmmmmNNMMNNNNNNNNNNmdd
-::///+osoooooyyyysyo+:y/sdNNNNMMNNNNNNNNNNNNNN
-:::::///o/+ooooSSoo+-odNNNNNNNNNNNNNNNNNN
-:++:-:+/s/-:/oo++/'oNNNNNNNNNNNNNNNNNN
++-/+o+/.oso+:--:/++:/hmNNNNNNNNNNNNNNNNNN
-::///++:-+o+o:/+o++/++omNNNNNNNNNNNNNNNNNNNN
-:/+oo+...osooso+ooooo+//-/ooooohdmNNNNNNNNNN
-:/+o+--/osoosyyyysyo+//:-.....-:syhdmNNmd
-/+o:-/ssoosyhyso++//:-.....-:/oo/-
-:/+:-:/++oo+osoo+++//-
..-+/+:-:/+ossso+++++//
-:-:-:++//+ooSSoo+++/..-/+oo+o+//..-/+oo+o+//..
:/osso:-::///+++/ooo/-:-...-/osyso++/+oo+/:..
-/+yyo//+++++o+//:-.....-:/+oo+::/ossyysso:
-+hhs+++++/:-----:/-:-:-:/:/:/+osyy+
..shso+//:-.....-:/:/:/-...../oss:
./oo/-.....-:/o:..+syysso+/-..-:/:
:/:-/+++:..oo..osyyyhhs/-..
-:/-:-+s:..++osysooo+/:
-0o+//++/:/:---
..so-/+/-:-

Move along.

ERROR: found mode='hello' but mode must be (convert|diff|info|interp|tag|view|help)
(base) f78@mbp115895 ~ %
```

Can you see him???

OBIWAN background and history

- Introduced as a “developer / power user” tool in SCALE 6.2
 - Fully functional but not documented within the manual
- “Promoted” to a full member of the ORIGIN family (with documentation) as of SCALE 6.3
- Used to **view** and **manipulate**[†] many common ORIGIN file types:
 - ORIGIN **library** files (.f33)
 - ORIGIN **concentration** files (.f71)
 - ORIGIN **resource** files (.yld, .dec, etc.)

[†] .f71 and .f33 files only

OBIWAN modes

```
obiwan <mode> <options> FILE
```

Tip: As command-line tool, OBIWAN prints to STDOUT and STDERR. You can redirect STDERR to a log file by adding this to the end of the call: **2>1log**

info

- Print basic info (e.g., # of positions, format)

view

- View file content, including unit conversions (f71 files)

diff

- Compare two ORIGIN binary files

convert

- Convert .f33 & .f71 files between different SCALE formats (6.1 => 6.3+)

tag

- Add and manipulate tags on a binary file (used for identification and interpolation)

interp

- Interpolate a file along tags or burnup values

patch

- Rewrite / "patch" data on an existing file

The following slides provide some of the most common or useful options for the different modes. Please see the manual or the help screen for all details.

Run OBIWAN two ways

1. From the command line, like any other command line utility:

```
/path/to/obiwan ...
```

2. From within a SCALE input `shell` block

```
=shell
```

```
${SCALE} /bin/obiwan ...
```

```
end
```

Running from the command line can be very powerful when combined with other command line utilities like `grep` or `awk`, which are standard on Linux/Mac, but SCALE also installs a version on Windows.

Reminder: ORIGEN nuclide IDs and sublibraries

- Note the difference between ORIGEN nuclide IDs vs. SCALE IDs
 - ORIGEN ID: 21095242
 - SCALE ID: 1095242
- Also note the 3 sublibs
 - Some nuclides appear on multiple sublibs
 - 10002004
 - 20002004
 - 30002004
 - Use S=0 to get totals (.f71) or last data set (.f33)

ORIGEN nuclide ID: S1ZZZAAA

S	- 1 digit for sublib integer
I	- 1 digit for isomeric state
ZZZ	- 3 digits of atomic number
AAA	- 3 digits of mass number

ORIGEN sublibraries:

- 1 LT** light nuclides, typically the result of activation of common elements and structural materials, e.g., O-16, Li-6
- 2 AC** fissionable actinides and their immediate decay and activation products, e.g., U-235, Pu-239, Cm-244
- 3 FP** fission products and their decay and activation products, e.g., Te-135, Cs-135, Xe-135

When S=0 (in most cases), this implies the sum over all sublib "versions" of a nuclide.

Viewing file contents: OBIWAN info mode



OBIWAN info: File metadata information

```
obiwan info [-type=TYPE] FILE
```

- Displays metadata about an ORIGEN binary file
 - Library (.f33), StateSet / Concentrations (.f71), Yield Resource (yld)
- **Example:** Display info for all w15_ assemblies in **`\${DATA}/arplibs`**:

```
obiwan info `${DATA}/arplibs/w15*
```

fileName	dataType	numSets	fileFormat	appVersion
w15_e05.f33	Origen::Library	31	bof(binary)	6.3.1
w15_e15.f33	Origen::Library	31	bof(binary)	6.3.1
w15_e20.f33	Origen::Library	31	bof(binary)	6.3.1
w15_e30.f33	Origen::Library	31	bof(binary)	6.3.1
w15_e40.f33	Origen::Library	31	bof(binary)	6.3.1
w15_e50.f33	Origen::Library	31	bof(binary)	6.3.1
w15_e60.f33	Origen::Library	31	bof(binary)	6.3.1

Viewing file contents: OBIWAN view mode



OBIWAN view: View binary file contents

```
obiwan view <options> FILE
```

- Offers several options for viewing the data in the file, with various output formatting options
- **View** works for the following ORIGEN file types:
 - **Yield** and **Decay** resource files (yld / dec)
 - **Library** files (.f33) [ORIGEN cross section library]
 - **StateSet** files (.f71) [ORIGEN concentration file]
- In general, one can view nuclide-wise information, select specific positions on the library or print information about all positions, change the formatting of the nuclide IDs, ...

File types supported by OBIWAN view –type=**xyz**

- **f71** "Origen::StateSet"
set of time-dependent nuclide inventory, power/flux history, and other state information
- **f33** "Origen::Library"
nuclide-to-nuclide transition data, e.g., decay constants and 1-group reaction cross sections
- **arc** "Origen::Archive"
collections of multiple related libraries, e.g., ORIGEN library for specific assembly at two different moderator densities
- **yld** "Origen::YieldResource"
fundamental fission product yield data
- **dec** "Origen::DecayResource"
fundamental decay data
- **rct** "Origen::ReactionResource"
fundamental reaction data

The “-type” is optional. If not provided, OBIWAN uses trial and error to identify type.

OBIWAN view: ORIGIN concentrations file (.f71)

```
obiwan view <options> FILE.f71
```

-format=

- | | |
|---------|-------------------------------------|
| info | - information table (DEFAULT) |
| csv | - comma-separated values |
| ii.json | - inventory interface in JSON |
| tags | - tags stored on each state |
| stdcomp | - SCALE standard composition format |

Further discussed in the following slides

Every format supports different options.

OBIWAN view f71: -format=csv

- Print nuclide concentrations in CSV format
- Often used to check nuclide densities in the terminal
- Output can be piped into file for further use with Excel, Python, etc.

Options:

- | | |
|-----------------------------|--|
| • -cases=[array] | list of cases |
| • -steps=[array] | list of steps |
| • -pos=(integer) | specific file position |
| • -prec=(integer) | number of significant digits (default: 4) |
| • -symbols=(boolean) | whether to output numeric ids (1001) or symbols (H-1) (default: 0) |
| • -idform=(string) | how to print nuclide ids |
| • -units=(string) | unit (default: moles) |

OBIWAN view f71: -format=csv -idform=??? -symbols=???

Options to control the nuclide id format:

1. Leave default: numeric values including sublibraries
2. Set “-symbols=1” and get nuclide symbol including sublibraries
3. Specify your own format by setting “-idform=...”

-idform= String(“{.}{.}{.}...”)

{:S} – sublibrary integer

{:TY} – sublibrary name (type)

{:EE} – elemental symbol (also Ee, ee)

{:Z} – up to 3 digits of atomic number

(also ZZ and ZZZ that add padded zeros)

{:A} – mass number

(also AA and AAA that add padded zeros)

{:M} – metastable indicator (also m)

{:I} – isomeric state integer

Use any number and any order of the above ID elements.

Example: -idform='{ :Ee } { :AAA } { :m }' → “U-235”, “Am-242m”

-symbols=0 → -idform='{ :S } { :I } { :ZZZ } { :AAA }'

Example: “20092235”

-symbols=1 → -idform='{ :TY } { :ee } { :AAA } { :m }'

Example: “AC:u235”

OBIWAN view f71: -format=csv -units=???

Atom and mass-based units

gato	Gram-atoms (moles, DEFAULT)
gram	Grams
kilo	Kilograms
gper	Grams per cm ³
atom	Atoms per b·cm

Reaction rates

abso	Total absorption
fiss	Total fission
capt	Total capture

Radiotoxicity indices (dilution factors)

airm	Radiotoxicity index in air (m ³)
h2om	Radiotoxicity index in water (m ³)

Activity and decay heat

becq	Becquerels
curi	Curies
watt	Total thermal output (Watts)
mevs	Total thermal output (MeV)
gamw	Thermal output from gammas (Watts)
gamm	Thermal output from gammas (MeV)

Radiation emissions

part	Particles / second
inte	Particles / MeV-second
ener	MeV / second

OBIWAN view f71: `-format=csv` – Example

```
obiwan view -format=csv -cases='[511]'  
            -idform="{:ee}-{:A}{:m}" -units=atom -prec=7 my.f71
```

1. Print nuclide concentrations for all time steps of case 511 (TRITON mixture number 511 if the f71 is a TRITON result file)
2. Sum nuclide concentrations over all sub-libraries (skipping the sub-library index in the *idform* causes nuclide concentrations to be summed over all sub-libraries)
3. Print nuclide concentrations in units of atoms/b-cm
4. Increase precision to 7 decimals

case, step,	511, 0,	511, 1,	511, 2,	...	511, 7,	511, 8
time,	0.0000000E+00,	5.6376000E+06,	1.1275200E+07,	...	3.9463200E+07,	4.5100800E+07
power,	0.0000000E+00,	5.7348830E-01,	5.5213755E-01,	...	3.5975227E-01,	3.3326435E-01
flux,	0.0000000E+00,	4.8210065E+14,	4.9118618E+14,	...	4.9505085E+14,	4.9515131E+14
volume,	1.3680576E+02,	1.3680576E+02,	1.3680576E+02,	...	1.3680576E+02,	1.3680576E+02
ac-224,	0.000000E+00,	4.971853E-30,	1.733246E-29,	...	1.768557E-27,	2.592403E-27
ac-225,	1.000000E-20,	9.091576E-22,	2.984673E-21,	...	3.105950E-19,	4.690864E-19
ac-226,	1.000000E-20,	5.133514E-24,	1.691406E-23,	...	1.580155E-21,	2.377381E-21
ac-227,	1.000000E-20,	2.429843E-18,	1.195139E-17,	...	1.469918E-16,	1.788429E-16
ac-228,	0.000000E+00,	4.401466E-21,	2.240864E-20,	...	2.822283E-19,	3.433662E-19
...						

Exercise: View nuclide densities from a TRITON .f71 file

Using the provided example file “**obiwan.f71**” (produced from TRITON):

1. How many “positions” are on this file?
2. How many “cases” are on this file?
3. How can you check TRITON’s mass normalization?
4. Print nuclide concentrations:
 - In units of atoms/barn-cm
 - For **only** the last step
 - For **only** the fuel-average mixture (case=-2)
 - With 8 decimals
 - In a pretty nuclide ID format
 - Nuclide concentrations summed over the 3 sub-libraries

Solution: View the .f71 general file content with OBIWAN

```
obiwan view [-format=info] obiwan.f71
```

The “-format=info” is optional in this case.

pos (-)	time (s)	power (MW)	flux (n/cm ² -s)	fluence (n/cm ²)	energy (MWd)	initialhm (MTIHM)	libpos (-)	case (-)	step (-)	DCGNAB (-)
1	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	1.26855e-01	1	1	0	DC----
2	2.59200e+07	3.43861e+00	1.95889e+14	5.07743e+21	1.03158e+03	1.26855e-01	2	1	1	DC----
3	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	2.53700e-01	1	2	0	DC----
4	2.59200e+07	7.88776e+00	1.94630e+14	5.04482e+21	2.36633e+03	2.53700e-01	2	2	1	DC----
5	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	2.53700e-01	1	3	0	DC----
6	2.59200e+07	6.81147e+00	1.85120e+14	4.79831e+21	2.04344e+03	2.53700e-01	2	3	1	DC----
7	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	1.26850e-01	1	4	0	DC----
8	2.59200e+07	2.85220e+00	1.77108e+14	4.59064e+21	8.55659e+02	1.26850e-01	2	4	1	DC----
9	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	5.97238e-02	1	5	0	DC----
10	2.59200e+07	9.50811e-01	1.58367e+14	4.10486e+21	2.85243e+02	5.97238e-02	2	5	1	DC----
11	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	5.97237e-02	1	6	0	DC----
12	2.59200e+07	6.54083e-01	1.53567e+14	3.98045e+21	1.96225e+02	5.97237e-02	2	6	1	DC----
13	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	1.19447e-01	1	7	0	DC----
14	2.59200e+07	9.56058e-01	1.50565e+14	3.90265e+21	2.86817e+02	1.19447e-01	2	7	1	DC----
15	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	1.00000e+00	1	0	0	DC----
16	2.59200e+07	2.35510e+01	1.79879e+14	4.66246e+21	7.06530e+03	1.00000e+00	2	0	1	DC----
17	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	1.00000e+00	1	-2	0	DC----
18	2.59200e+07	2.35510e+01	1.79879e+14	4.66246e+21	7.06530e+03	1.00000e+00	2	-2	1	DC----

1. 18 positions
2. 7 “cases” (i.e., material regions)
3. **Mass normalization:** See **weighted sum of depleted materials** (case 0) and **weighted sum of fuel mixtures** (i.e., mixtures with HM content) (case -2)

Solution: View concentrations at specific .f71 file positions

```
obiwan view -format=csv -units=atom -pos=18 -prec=8  
-idform='{:ee}-{:AAA}{:m}' obiwan.f71
```

```
case, -2  
step, 1  
time, 2.5920000E+07  
power, 2.3550999E+01  
flux, 1.7987872E+14  
volume, 1.1290341E+05  
ac-224, 2.63523478E-28  
ac-225, 8.55815362E-21  
ac-226, 3.28139419E-23  
ac-227, 9.95996899E-18  
ac-228, 5.58260037E-21  
...
```

By using an **idform** that does not include the sub-library, OBIWAN sums over all sub-libraries.

Don't forget to specify the unit. The default unit is moles!

The “-pos” keyword is supported since SCALE 6.3.3. Alternatively, use a combination of “-cases” and “-steps”:
-cases='[-2]' -steps='[1]'

OBIWAN view f71: -format=stdcomp

- Print nuclide concentrations in SCALE standard composition format
- Simplifies generation of CSAS, TRITON, or MAVRIC inputs

Mixture setting:

- **-mat_id=(integer)** id (default: case_id·100 + step_number)
- **-temper=(real)** temperature (default: 293.15 K)

Nuclide filters:

- **-cases=[array]** list of cases
- **-steps=[array]** list of steps
- **-pos=(int)** specific file position
- **-threshold=(real)** minimum nuclide concentration (default: 1e-20)
- **-addnux=[1,2,3,4]** pre-specified lists of nuclides as used in TRITON
- **-nuclides=[array]** list of nuclides (SIZZZAAA or IZZZAAA format)
- **-allow_s/i/z/a=[arr]** specific sub-libraries, isomeric state, Z, or A
- **-min_z, -max_z=(int)** minimum or maximum Z ← Less frequently used

Ensure that you select a specific location via cases and steps, or via the specific position. Otherwise, densities are summed over cases and/or steps.

Nuclide filters *addnux* and *threshold* are additive filters. Nuclides listed in the nuclides array are always included, even if their densities are below the *threshold*.

OBIWAN view f71: -format=stdcomp – Example

```
obiwan view -format=stdcomp -pos=1
            -mat_id=111 -temper=959.0
            -addnux=1 -threshold=1e-10
            -nuclides="[6012,6013,8016,8017]" my.f71
```

1. Print standard composition output for position 1
2. Set mixture number to 111 and mixture temperature to 959.0 K
3. Get all ADDNUX=1 nuclides that have densities higher than 1e-10 atoms/b-cm
4. Also add a specific list of nuclides that would otherwise not be included: C and O isotopes from the initial fresh UCO fuel composition

C-12	111	0.0	0.0116946	959	end
C-13	111	0.0	0.00012672	959	end
O-16	111	0.0	0.0353771	959	end
O-17	111	0.0	1.3488e-05	959	end
U-234	111	0.0	2.29257e-05	959	end
U-235	111	0.0	0.00225377	959	end
U-236	111	0.0	0.000444882	959	end
U-238	111	0.0	0.0179255	959	end
Np-237	111	0.0	3.06747e-05	959	end
Pu-238	111	0.0	9.29665e-06	959	end
Pu-239	111	0.0	0.000226096	959	end
Pu-240	111	0.0	8.04195e-05	959	end
Pu-241	111	0.0	7.12146e-05	959	end
Pu-242	111	0.0	3.05811e-05	959	end
Am-241	111	0.0	6.45801e-07	959	end
Am-242	111	0.0	5.62231e-09	959	end
Am-243	111	0.0	4.94861e-06	959	end
Cm-242	111	0.0	4.67344e-07	959	end
Cm-243	111	0.0	1.09649e-08	959	end

Exercise: Print nuclide densities from a TRITON .f71 file in standard composition input format

Using the provided example file “**obiwan.f71**” (produced from TRITON):

1. Print nuclide concentrations in SCALE's standard composition format:
 - For **only** the fuel-average mixture (case=-2)
 - For **only** the last step
 - With a temperature of 900 K
 - With a new mixture number 4321

Solution: Print standard composition format

```
obiwan view -format=stdcomp -pos=18 -temper=900 -mat_id=4321 -addnux=2 obiwan.f71
```

H-1	4321	0.0	2.90365e-09	900	end
B-10	4321	0.0	8.27973e-20	900	end
B-11	4321	0.0	1.10652e-14	900	end
N-14	4321	0.0	1.05977e-13	900	end
O-16	4321	0.0	0.0453952	900	end
Kr-83	4321	0.0	7.89671e-07	900	end
Zr-91	4321	0.0	6.47589e-06	900	end
Zr-93	4321	0.0	9.83035e-06	900	end
Zr-94	4321	0.0	1.01248e-05	900	end
Zr-95	4321	0.0	3.00707e-06	900	end
Zr-96	4321	0.0	1.01949e-05	900	end
Nb-93	4321	0.0	1.63609e-13	900	end
Nb-95	4321	0.0	1.57686e-06	900	end
Mo-95	4321	0.0	5.66147e-06	900	end
Mo-97	4321	0.0	9.7185e-06	900	end
Mo-98	4321	0.0	9.61215e-06	900	end
Mo-99	4321	0.0	1.33554e-07	900	end
Mo-100	4321	0.0	1.05247e-05	900	end
Tc-99	4321	0.0	9.79773e-06	900	end
Ru-101	4321	0.0	8.75665e-06	900	end
Ru-102	4321	0.0	7.72753e-06	900	end
Ru-103	4321	0.0	1.21934e-06	900	end
Ru-104	4321	0.0	4.22901e-06	900	end
Ru-106	4321	0.0	1.29692e-06	900	end
Rh-103	4321	0.0	4.5095e-06	900	end
Rh-105	4321	0.0	2.35039e-08	900	end
Pd-105	4321	0.0	2.59842e-06	900	end
Pd-107	4321	0.0	9.79473e-07	900	end
Pd-108	4321	0.0	5.57499e-07	900	end
Ag-109	4321	0.0	3.43645e-07	900	end
Cd-113	4321	0.0	4.25101e-09	900	end
In-115	4321	0.0	3.03454e-08	900	end
...					

...					
Sm-151	4321	0.0	2.42302e-07	900	end
Sm-152	4321	0.0	9.46275e-07	900	end
Sm-153	4321	0.0	7.02137e-09	900	end
Eu-151	4321	0.0	3.86019e-10	900	end
Eu-153	4321	0.0	4.41993e-07	900	end
Eu-154	4321	0.0	3.96448e-08	900	end
Eu-155	4321	0.0	3.32478e-08	900	end
Eu-156	4321	0.0	1.37531e-08	900	end
Gd-152	4321	0.0	1.41299e-10	900	end
Gd-154	4321	0.0	7.82109e-06	900	end
Gd-155	4321	0.0	2.5419e-05	900	end
Gd-156	4321	0.0	0.000106087	900	end
Gd-157	4321	0.0	1.24164e-05	900	end
Gd-158	4321	0.0	0.000140241	900	end
Gd-160	4321	0.0	8.28998e-05	900	end
U-234	4321	0.0	6.17764e-06	900	end
U-235	4321	0.0	0.000580966	900	end
U-236	4321	0.0	3.41694e-05	900	end
U-238	4321	0.0	0.0215576	900	end
Np-237	4321	0.0	1.08908e-06	900	end
Pu-238	4321	0.0	7.38996e-08	900	end
Pu-239	4321	0.0	5.95709e-05	900	end
Pu-240	4321	0.0	7.2892e-06	900	end
Pu-241	4321	0.0	2.38151e-06	900	end
Pu-242	4321	0.0	1.55757e-07	900	end
Am-241	4321	0.0	2.33199e-08	900	end
Am-242	4321	0.0	4.70285e-11	900	end
Am-243	4321	0.0	4.66344e-09	900	end
Cm-242	4321	0.0	2.15387e-09	900	end
Cm-243	4321	0.0	9.4897e-12	900	end
Cm-244	4321	0.0	2.41893e-10	900	end
Am-242m	4321	0.0	1.95957e-10	900	end

OBIWAN view f71: -format=ii.json

- Save nuclide concentrations in inventory interface (II) JSON format
- Simplifies data exchange between SCALE and non-SCALE programs
- An II.JSON file contains the following data:

- Relevant nuclear data for each nuclide:
 - Nuclide ID and element name
 - Molar mass
 - Atomic and mass number
 - Decay constant
 - Decay energy
- Each **case** in a file contains the following data:
 - Nuclide-wise inventory (default: in moles) for this case, at one or more time points
 - Volume of this case (default: in cm³)
- Each case can have a different time vector and use different units, but consistency is recommended
- A case corresponds to the case on an ORIGEN F71 concentration file

Options:

- | | | | |
|-------------------------|------------------------|--------------------------------|-------------------------|
| • -cases=[array] | list of cases | • -allow_s =[array] | specific sub-libraries |
| • -steps=[array] | list of steps | • -allow_i/z/a =[array] | isomeric state, Z, or A |
| • -pos=(int) | specific file position | • -min_z, -max_z=(int) | minimum or maximum Z |

OBIWAN view f71: `-format=ii.json` – Example

```
obiwan view -format=ii.json -cases='[541]'  
           -allow_s="[2,3]" my.f71    >  my.ii.json
```

1. Generate II.JSON file for case number 541 (TRITON mixture number 541)
2. Only include sub-libraries 2 and 3 — exclude sub-library 1 (light nuclides)
3. Pipe printout into file with ending JSON

By including relevant nuclear data, the II.JSON is a standalone file that can be used to calculate decay heat, mass, activity, and other derived quantities from the amount of moles of each nuclide.

We developed several Jupyter Notebooks that read II.JSON files and analyze inventory, decay heat, and other quantities over time, identify top contributors to these quantities, and compare data from multiple files. Ask if you are interested.

OBIWAN view: Library file (.f33) output formats (-format=)

Transition matrix

coeff	output tabular transition coefficients
coefft	same as coeff but with transition IDs
matrix	output full transition matrix

Neutron production

nprodr	neutron production ratio
genneu	neutron generation

One-group library cross sections

loxs	1-group effective loss cross section
fisxs	1-group fission cross section

Library metadata

tags	output file tags
nucl	nuclide list
burnups	output burnup list
reaction	allowed reactions

Output formats for post-processing

json	output in JSON format
graphviz	output dot file transitions to STDOUT

Example: View ^{240}Pu loss XS as a function of burnup

(1) Obtain the loss XS: `obiwan view -format=loxs -symbols=1 ${DATA}/arplibs/w17_e50.f33`

```
...
AC:pu238 1.9918E+01 1.9426E+01 1.9117E+01 1.8876E+01 1.8700E+01 1.8550E+01 1.8445E+01 1.8403E+01
AC:pu239 1.5127E+02 1.4876E+02 1.4515E+02 1.4010E+02 1.3598E+02 1.3195E+02 1.2831E+02 1.2563E+02
AC:pu240 2.2669E+02 2.2674E+02 2.2551E+02 2.1707E+02 2.0451E+02 1.8794E+02 1.6994E+02 1.5479E+02
AC:pu241 1.2945E+02 1.2672E+02 1.2398E+02 1.2075E+02 1.1815E+02 1.1565E+02 1.1346E+02 1.1192E+02
...

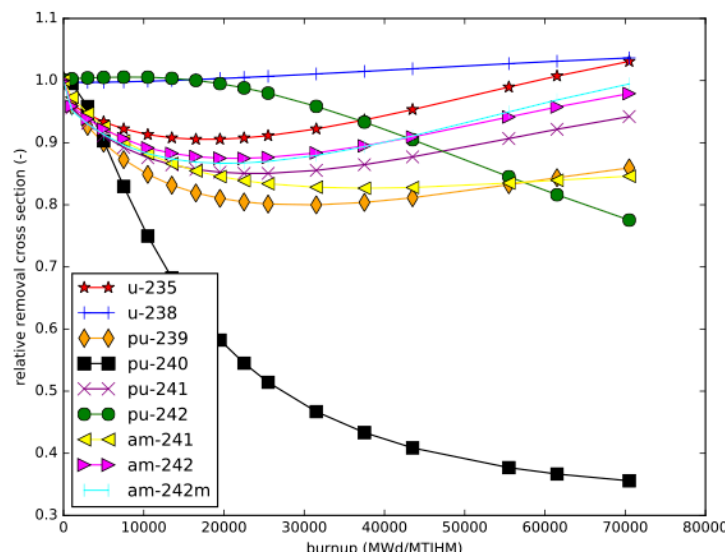
```

(2) Obtain the burnup: `obiwan view -format=burnup ${DATA}/arplibs/w17_e50.f33`

pos MWd/MTIHM

1	0
2	40
3	1040
4	3000
5	5000
6	7500
7	10500
8	13500
...	...

(3) Plot with external tool:



Same method can be applied across different assembly dimensions (initial enrichment, moderator density, etc.)

Fulcrum of SCALE 6.3.3 can visualize fission XS and loss XS from f33 file.

Exercise: View a TRITON-generated ORIGEN library file (.f33)

Using the provided example file “**obiwan.f33**” (produced from TRITON):

1. Print the list of burnups contained on the .f33 library
2. Find the loss cross section of U-235, Pu-239 as a function of burnup
3. Print all the transition coefficients (yield times cross section) to daughter Te-135 (with `-format=coef`)

Solution: View burnup points in the .f33 file

```
obiwan view -format=burnups obiwan.f33
```

pos	MWd/MTIHM				
		11	22500.0000	22	55500.0000
1	0.0000	12	25500.0000	23	58500.0000
2	40.0000	13	28500.0000	24	61500.0000
3	1040.0000	14	31500.0000	25	64500.0000
4	3000.0000	15	34500.0000	26	67500.0000
5	5000.0000	16	37500.0000	27	70500.0000
6	7500.0000	17	40500.0000	28	73500.0000
7	10500.0000	18	43500.0000	29	76500.0000
8	13500.0000	19	46500.0000	30	79500.0000
9	16500.0000	20	49500.0000	31	82500.0000
10	19500.0000	21	52500.0000		

Solution: Loss XS for U-235, Pu-239 as a function of burnup

```
obiwan view -format=loxs -sizzzaaa=92235 -symbols=1 -transpose=1 obiwan.f33
```

```
obiwan view -format=loxs -sizzzaaa=94239 -symbols=1 -transpose=1 obiwan.f33
```

burnupAC:u235				burnupAC:pu239			
	3.7500e+04	3.0899e+01			3.7500e+04	9.0775e+01	
0.0000e+00	3.2040e+01	4.0500e+04	3.1126e+01	0.0000e+00	1.0965e+02	4.0500e+04	9.0957e+01
4.0000e+01	3.1489e+01	4.3500e+04	3.1375e+01	4.0000e+01	1.0820e+02	4.3500e+04	9.1244e+01
1.0400e+03	3.1104e+01	4.6500e+04	3.1643e+01	1.0400e+03	1.0624e+02	4.6500e+04	9.1626e+01
3.0000e+03	3.0816e+01	4.9500e+04	3.1929e+01	3.0000e+03	1.0360e+02	4.9500e+04	9.2095e+01
5.0000e+03	3.0608e+01	5.2500e+04	3.2233e+01	5.0000e+03	1.0140e+02	5.2500e+04	9.2645e+01
7.5000e+03	3.0407e+01	5.5500e+04	3.2553e+01	7.5000e+03	9.9101e+01	5.5500e+04	9.3269e+01
1.0500e+04	3.0237e+01	5.8500e+04	3.2889e+01	1.0500e+04	9.6874e+01	5.8500e+04	9.3960e+01
1.3500e+04	3.0135e+01	6.1500e+04	3.3238e+01	1.3500e+04	9.5119e+01	6.1500e+04	9.4715e+01
1.6500e+04	3.0092e+01	6.4500e+04	3.3600e+01	1.6500e+04	9.3750e+01	6.4500e+04	9.5525e+01
1.9500e+04	3.0100e+01	6.7500e+04	3.3973e+01	1.9500e+04	9.2700e+01	6.7500e+04	9.6387e+01
2.2500e+04	3.0151e+01	7.0500e+04	3.4356e+01	2.2500e+04	9.1914e+01	7.0500e+04	9.7295e+01
2.5500e+04	3.0240e+01	7.3500e+04	3.4749e+01	2.5500e+04	9.1350e+01	7.3500e+04	9.8245e+01
2.8500e+04	3.0363e+01	7.6500e+04	3.5147e+01	2.8500e+04	9.0977e+01	7.6500e+04	9.9227e+01
3.1500e+04	3.0516e+01	7.9500e+04	3.5551e+01	3.1500e+04	9.0770e+01	7.9500e+04	1.0024e+02
3.4500e+04	3.0695e+01	8.2500e+04	3.5957e+01	3.4500e+04	9.0708e+01	8.2500e+04	1.0127e+02

Solution: Transition coefficients for Te-135

```
obiwan view -format=coefft -symbols=1 obiwan.f33 | grep te135
```

```
FP:te135 <--[decay,beta-|-2]-- FP:sb135 3.2201e-01 3.2201e-01 3.2201e-01 ... 3.2201e-01 3.2201e-01 3.2201e-01
FP:te135 <--[decay,beta-,neutron|-26]-- FP:sb136 3.5814e-01 3.5814e-01 3.5814e-01 ... 3.5814e-01 3.5814e-01 3.5814e-01
FP:te135 <--[fission|18]-- AC:th227 3.1997e-01 3.1626e-01 3.1352e-01 ... 3.2551e-01 3.2788e-01 3.3027e-01
FP:te135 <--[fission|18]-- AC:th229 2.9720e-01 2.9668e-01 2.9619e-01 ... 2.8938e-01 2.8957e-01 2.8977e-01
FP:te135 <--[fission|18]-- AC:th232 9.5584e-04 9.5732e-04 9.5907e-04 ... 9.6253e-04 9.6160e-04 9.6069e-04
FP:te135 <--[fission|18]-- AC:pa231 6.0233e-03 6.0323e-03 6.0412e-03 ... 6.0172e-03 6.0108e-03 6.0044e-03
FP:te135 <--[fission|18]-- AC:u232 8.0034e-02 7.9714e-02 7.9541e-02 ... 8.4851e-02 8.5161e-02 8.5468e-02
FP:te135 <--[fission|18]-- AC:u233 6.2912e-01 6.2154e-01 6.1601e-01 ... 6.5611e-01 6.6118e-01 6.6629e-01
FP:te135 <--[fission|18]-- AC:u234 1.0625e-02 1.0640e-02 1.0655e-02 ... 1.0598e-02 1.0587e-02 1.0575e-02
FP:te135 <--[fission|18]-- AC:u235 7.9705e-01 7.8172e-01 7.7099e-01 ... 8.7605e-01 8.8701e-01 8.9806e-01
FP:te135 <--[fission|18]-- AC:u236 1.2414e-02 1.2428e-02 1.2356e-02 ... 1.0736e-02 1.0717e-02 1.0699e-02
FP:te135 <--[fission|18]-- AC:u237 7.1777e-02 7.1809e-02 7.1841e-02 ... 7.3530e-02 7.3547e-02 7.3570e-02
FP:te135 <--[fission|18]-- AC:u238 4.7928e-03 4.8002e-03 4.8085e-03 ... 4.8074e-03 4.8023e-03 4.7972e-03
...
FP:te135 <--[n,p|103]-- FP:i135 0.0000e+00 0.0000e+00 0.0000e+00 ... 0.0000e+00 0.0000e+00 0.0000e+00
FP:te135 <--[n,2p|111]-- FP:xe136 5.2420e-28 5.5115e-28 9.5624e-28 ... 3.9169e-25 4.1051e-25 4.2891e-25
FP:i135 <--[decay,beta-|-2]-- FP:te135 3.6482e-02 3.6482e-02 3.6482e-02 ... 3.6482e-02 3.6482e-02 3.6482e-02
```

OBIWAN view: Decay and yield resource files

YieldResource

DecayResource

Parent ID	EALF (eV)	Daughter ID	Yield
...			
92235	2.53000e-02	33088	1.30179e-03
92235	2.53000e-02	34088	2.74899e-03
92235	2.53000e-02	35088	1.40008e-02
92235	2.53000e-02	36088	1.76224e-02
...			

Line #	Z AID	Sym.	λ (1/s)	Q (MeV)	Q_Y (fraction)	Decay mode(s)
540	1041102	nb102m	5.3319e-01	4.8393e+00	0.5000	beta-[m0] 1.0000e+00
541	41103	nb103	4.6210e-01	3.5429e+00	0.4690	beta-[m0] 1.0000e+00 beta-,neutron[m0] 1.1770e-08
542	41104	nb104	1.4146e-01	5.5154e+00	0.5544	beta-[m0] 9.9940e-01 beta-,neutron[m0] 6.0000e-04
543	1041104	nb104m	7.3739e-01	5.5162e+00	0.5545	beta-[m0] 9.9950e-01 beta-,neutron[m0] 5.0000e-04
544	41105	nb105	2.3497e-01	4.4695e+00	0.4777	beta-[m0] 9.8300e-01 beta-,neutron[m0] 1.7000e-02

Exercise: Viewing an ORIGEN decay resource

Using the ORIGEN decay resource at
`${DATA}/origen_data/origen.rev03.decay.data`

1. Find the decay constant of Xe-135
2. Find a nuclide that decays into a metastable state with branching ratio >40%

Solution: Viewing an ORIGIN decay resource

- Xe-135 decay constant:

```
obiwan view ${DATA}/origen_data/origen.rev03.decay.data | grep xe135
```

977	54135	xe135	2.1066e-05	5.6798e-01	0.4370	beta-[m0]	1.0000e+00	
978	1054135	xe135m	7.5553e-04	5.2673e-01	0.8138	beta-[m0]	3.0000e-03	i.t.[m0] 9.9700e-01

- Nuclide(s) with metastable branching ratio >40%

```
obiwan view ${DATA}/origen_data/origen.rev03.decay.data | grep "m1"
```

(... scroll through data and find any case where the branching fraction to [m1] > 0.40 ...)

299	33073	as73	9.9908e-08	9.7528e-03	0.4788	beta+[m1]	1.0000e+00
-----	-------	------	------------	------------	--------	-----------	------------

Exercise: YieldResource data for Te-135 fission yields

Using the ORIGEN fission yield resource at
`${DATA}/origen_data/origen.rev05.yields.data`

- a) Find the fission yield of **U-235** to **Te-135** for:
 - a) Thermal fission (2.53000e-02 eV)
 - b) Fast fission (2.00000e+06 eV)
- b) Find the fission yield of **Pu-239** to **Te-135** for:
 - a) Thermal fission (2.53000e-02 eV)
 - b) Fast fission (2.00000e+06 eV)

Hint: ZAID for
Te-135 is 52135

Solution: YieldResource data for Te-135 fission yields

```
obiwan view ${DATA}/origen_data/origen.rev05.yields.data | grep 52135
```

• U-235:

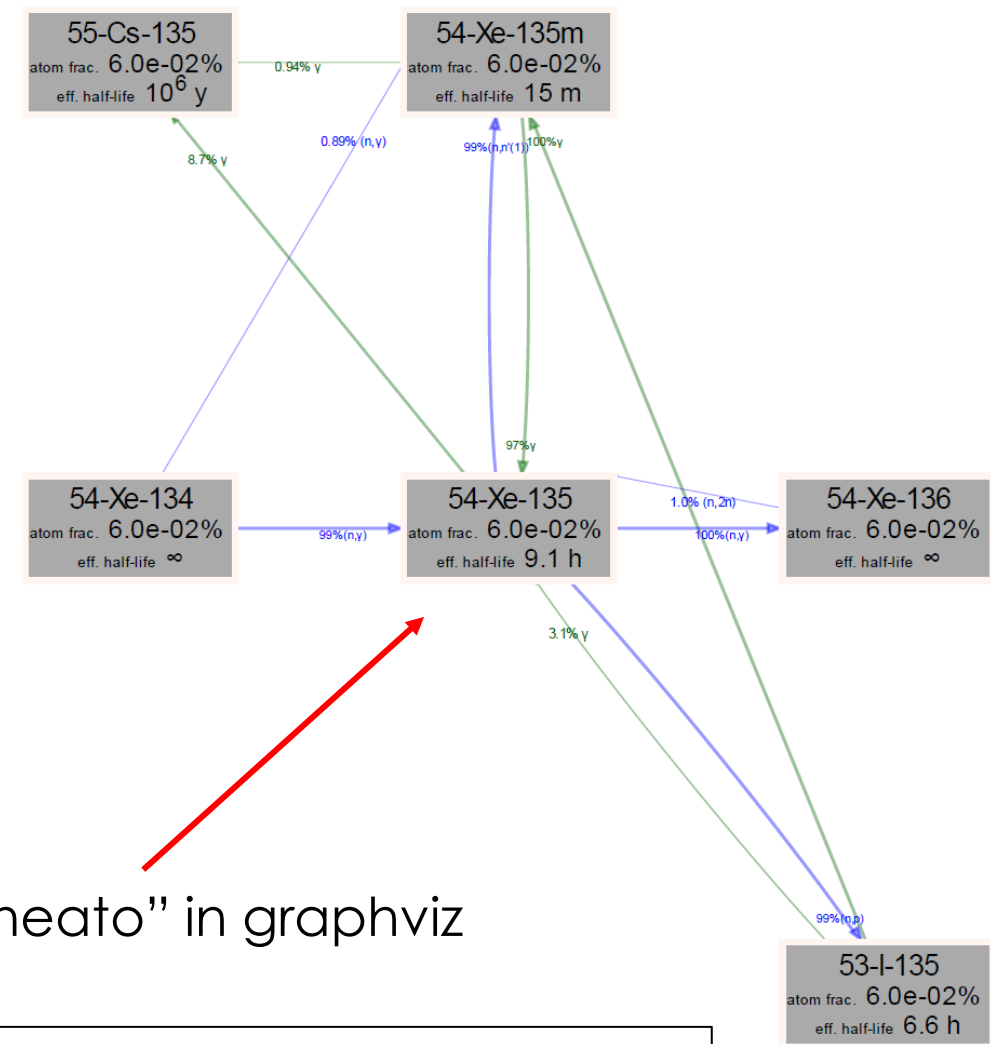
Parent ID	EALF (eV)	Daughter ID	Yield
92235	2.530000e-02	52135	3.22825e-02
92235	2.000000e+06	52135	2.47357e-02

• PU-239:

Parent ID	EALF (eV)	Daughter ID	Yield
94239	2.530000e-02	52135	2.19297e-02
94239	2.000000e+06	52135	2.10715e-02

OBIWAN view: Visualizing transitions using graphviz

- OBIWAN view can also produce a transition matrix output from an f33 file with an enhanced and visually appealing format (graphviz)
- Graphviz layout programs take descriptions of graphs in a text language (i.e., .dot) and make diagrams in several useful formats such as images



```
$SCALE/bin/obiwan view -format=graphviz origen.f33 > origen.dot
```

Visualizing transitions from graphviz .dot files

- Users can obtain graphviz:
 - **Windows:** via an installer from the graphviz website:
<https://graphviz.org/>
 - **MacOS/Linux:** via a package manager
 - e.g. `brew install graphviz` or `sudo apt install graphviz`
- Generating a .dot file from an f33 library:

```
$SCALE/bin/obiwan view -format=graphviz origen.f33 > transitions.dot
```

- Rendering a transition graph using graphviz:

```
neato -n1 transitions.dot -T pdf >transitions.pdf
```

Understanding sources for Xe-135

- Xe-135 sources

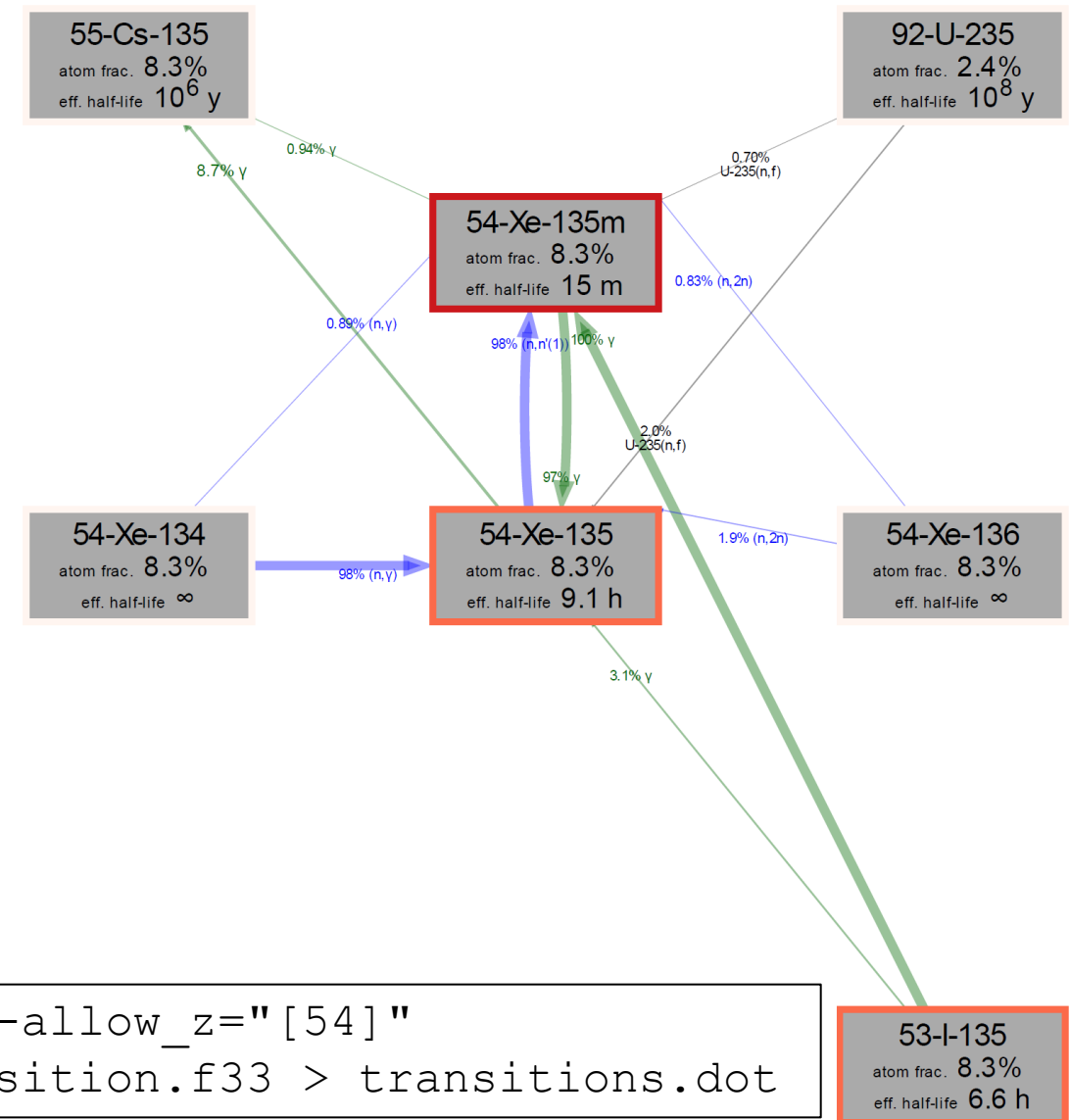
- Fission Product: 2.0% from U-235 (n,f)
- Reaction: 1.9% from Xe-136 (n, 2n)
- Reaction: 98% from Xe-134 (n,γ)
- Decay: 97% I.T. from Xe-135m
- Decay: 3.1% beta decay from I-135

- Xe-135m sources

- Fission Product : 0.70% from U-235 (n,f)
- Reaction: 0.83% from Xe-136 (n, 2n)
- Reaction: 0.89% from Xe-134 (n,γ)
- Reaction: 98% from Xe-135 (n,n')
- Decay: 100% beta decay from I-135

```
${SCALE}/bin/obiwan view -format=graphviz -allow_z="[54]"  
-allow_a="[135,235]" -allow_fission=1 transition.f33 > transitions.dot
```

U-235 irradiation problem



File comparison: OBIWAN diff mode



OBIWAN diff: Comparing ORIGEN files

```
obiwan diff -format=y [opts] file1 file2
```

Option	Description	Default
-reltol	Relative tolerance threshold (%)	-reltol=0.01
-pos	File position at which to perform a comparison	-pos=1
-format=y	Smart format (side-by-side comparison)	Difference indicated via return (status) code
-format=genneu	Neutron generation (by nuclide)	
-format=eq	Exact match (no output)	
-format=approx_eq	Fast comparison (no output)	

OBIWAN diff: Comparing binary ORIGIN files

```
obiwan diff -format=String(y) [opts] file1 file2
```

- Works for any binary ORIGIN file format
 - YieldResource, DecayResource
 - StateSet (.f71)
 - Library (.f33)
- Tolerance can be adjusted to look only for differences above a threshold

Exercise

Tip: GE10×10-8 variants in `DATA/arp1ibs/` are named as follows: `g10-eXX-wYY.f33`
XX is the initial U-235 enrichment (wt %), and **YY** is the moderator density (Y.Y g/cc)

For the GE10×10-8 BWR library:

1. Between 1.5% and 3.0% enrichment (0.5 g/cc moderator density), how many transition coefficients change by:
 - a) >5%?
 - b) >10%?
 - c) >50%?
2. Between 0.3 and 0.5 g/cc moderator density (at 3.0% enrichment), how many transition coefficients change by:
 - a) >5%?
 - b) >10%?
 - c) >50%?

Hints:

- Use “grep” to look for transitions with a difference over the threshold
- You can also use “wc -l” to get the number of lines in a file

Solution: Transition coefficient differences for GE10×10-8

```
obiwan diff -reltol=5.0 ${DATA}/arplibs/g10_e15w05.f33  
${DATA}/arplibs/g10_e30w05.f33 | grep DIFF | wc -l
```

```
10001001 <-- 10001002    1.38080e-03    1.45989e-03    5.7e+00% DIFF(|)  
10001001 <-- 10002003    4.78756e+02    3.65623e+02    2.4e+01% DIFF(|)  
10001001 <-- 10003006    1.54189e-03    1.62720e-03    5.5e+00% DIFF(|)  
10001001 <-- 10004007    3.34785e+03    2.55457e+03    2.4e+01% DIFF(|)  
10001001 <-- 10004009    1.04794e-11    1.82783e-11    7.4e+01% DIFF(|)  
10001001 <-- 10005010    3.12971e-03    3.28689e-03    5.0e+00% DIFF(|)  
10001001 <-- 10005011    1.79766e-07    1.90195e-07    5.8e+00% DIFF(|)  
...
```

1. **GE10×10-8:** 1.5% vs. 3.0% enrichment
(mod. dens = 0.5 g/cc)

- a) 15,175
- b) 2,308
- c) 2,096

2. **GE10×10-8:** 0.3 vs. 0.5 g/cc moderator
density (enrich = 3.0%)

- a) 14,984
- b) 1,458
- c) 328

File comparison: OBIWAN convert mode



OBIWAN convert: Migrating between file formats

- Supported conversion **formats** (`-format=`)
 - **s61**: 6.1 “legacy” format
 - **bof**: 6.2+ binary object format
 - **hdf5**: New HDF5 format for 6.3+ (*libraries only*)
- Supported conversion **versions** (`-version=`)
 - **6.1**: Equivalent to “`-format=s61`”
 - **6.2**: Equivalent to “`-format=bof`”

```
obiwan convert -format=bof myOldLibrary.f33
```

OBIWAN convert: Filtering on file type

- Supported conversion **type** filters (`-type=`)
 - **f33**: Origen::Library
 - **f71**: Origen::StateSet
 - **yld**: Origen::YieldResource
 - **dec**: Origen::DecayResource

OBIWAN convert: Setting library burnup values

- The selection of data for specific burnup points with OBIWAN requires burnup information on the library. If this is not available (e.g., if the libraries were generated with SCALE 6.2), we need to append the burnup information to the library using the `-setbu` argument:

```
obiwan convert -setbu=[1500.0, 3025.0, 10000.0] -i my_library.f33
```

```
obiwan convert -setbu=[1500.0, 3025.0, 10000.0] -dir my_dir my_library.f33
```

-setbu=[array]: Specify an **array** of burnup values (typically MWd/MTIHM) corresponding to **each library position** (one entry per position)

-i Overwrite existing file in-place (optional)

-dir Specify *existing* directory into which new file is saved

OBIWAN convert: Library “thinning” and interpolation

- Libraries can be “thinned,” reducing the library to an interpolated set of burnup points (similar to ARP)
- To **select** and potentially **interpolate** time/burnup steps, specify **thin=1** and a set of **time values** (burnups) to thin to:

```
obiwan convert -interp=cubic  
-tvals=[2000.0, 5500.0, 7500.0] -thin=1 -i my_library.f33
```

-tvals= Array of time (burnup) values to interpolate to, corresponding to values in the library

-interp= Interpolation method: “near” = nearest-neighbor, “lin” = linear interpolation and “cubic” = cubic spline

Exercise: Burnup interpolation of a reactor library (.f33)

For a Westinghouse 17×17 PWR assembly at 4% initial enrichment (`${DATA}/arplibs/w17_e40.f33`):

1. Print out the U-235 and Pu-239 fission cross sections as a function of burnup
2. Create a **new** library interpolated to burnup points of 650, 6,250, 21,500, and 36,750 MWd/MTHM
3. Compare the original and interpolated fission cross sections for U-235 and Pu-239 as a function of burnup

Solution: Burnup interpolation of a reactor library (.f33)

View fission XS:

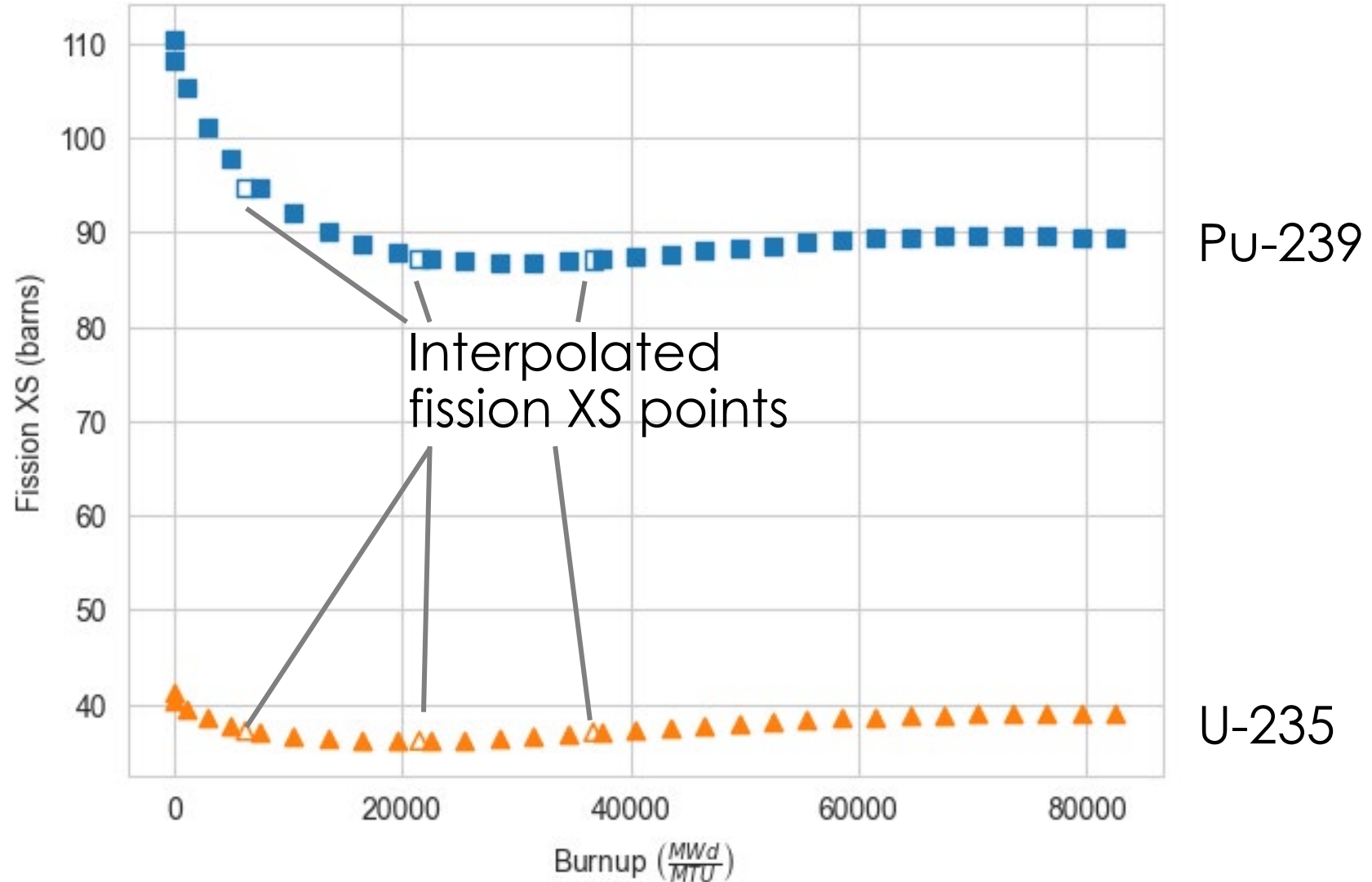
```
obiwan view -format=fisxs -symbols=1  
${DATA}/arplibs/w17_e40.f33 | grep -E 'u235|pu239'
```

```
AC:u235  4.1298e+01 4.0263e+01 3.9423e+01 ... 3.8998e+01 3.8978e+01 3.8937e+01  
AC:pu239 1.1034e+02 1.0830e+02 1.0533e+02 ... 8.9598e+01 8.9504e+01 8.9370e+01
```

Interpolate library:

```
obiwan convert -thin=1 -tvals='[6250,21500,36750]'  
${DATA}/arplibs/w17_e40.f33
```

Solution: Comparison of interpolated fission XS



Library metadata and interpolation: OBIWAN tag and interp modes



OBIWAN tag: Adding and viewing library metadata

Two types of tags: “ID” (categorical) and “interp” (continuous)

ID tags

- String-valued
- Correspond to some common characteristic between libraries
- Values can be used to represent **discrete** interpolation characteristics / states
 - Gd-bearing rods
 - Assembly / fuel pebble location

Interp tags

- Real-valued
- Correspond to **continuously valued** parameters **used for interpolation**
- Examples
 - Initial ^{235}U enrichment
 - Average moderator density
 - Average fuel temperature
 - Average moderator boron concentration

OBIWAN tag: Example of adding library metadata

Sets ID tags "name" = "bob" and
"type" = "LWR"

obiwan tag -**idtags**="name=bob, type=LWR"
-**interptags**="pitch=0.5, temp=293.0"

Sets interpolation tags
"pitch" = 0.5 and "temp" = 293.0

Interpolation tags are arbitrary and can therefore have any name with values of any unit. It is recommended to use standard units such as cm and K.

OBIWAN interp: Interpolating libraries across tags

```
obiwan interp <options> FILE1 FILE2
```

- Interpolates between ORIGEN libraries based on a specified series of tag attributes to generate a new library
- **idtags** – List of ID tags to down-select for interpolation
- **interptags** – List of interpolable values to interpolate the new library to

Example:

```
obiwan interp  
  -idtags='Assembly Type=w17x17,Fuel Type=MOX'  
  -interptags='Enrichment=4.5,Moderator Density=0.55'  
  file1.f33 file2.f33 file3.f33
```

Upcoming features



Coming attractions for OBIWAN in SCALE 7

- Blend concentrations from multiple .f71 files
`obiwan blend -wts=<weights> -in=<files> -out=<file>`
- More options when printing concentrations from .f71 files
 - Permit union of nuclide filters
 - Support all Z/A/I/S nuclide filters with all formats
- Conversion of multiple .f33 files into a consolidated HDF5 “library archive”
`obiwan convert format=arc FILES`
- More library checking/verification functionality

Supplemental



OBIWAN patch

```
obiwan patch -from=SOURCE -data="path/to/data" <options> FILE
```

- Patches a library with data from another file
- Selection of data for patch via hierarchical data description

Example patching of a 3-position library:

- Patch library with data from 1,1,2
`obiwan patch -from=my.f33 -data=coeff[1|1|2]/n_production_xs patched.f33`
- Patch library with new data (my.f33 must have 3 positions)
`obiwan patch -from=my.f33 -data=coeff[*]/n_production_xs patched.f33`
- Patch library with new data (my.f33 must have 3 positions)
`obiwan patch -from=my.f33 -data=coeff/n_production_xs patched.f33`