

Burnup Zr

Depletion Inventory

Moxe

FISPACT-II

Getting Started

Thomas Stainer, Mark Gilbert, Greg Bailey, Olga Vilkhivskaya United Kingdom Atomic Energy Authority

FISPACT-II workshop

October 23-25th, 2019, Manchester



This work was funded by the RCUK Energy Programme
[Grant number EP/P012450/1]



FISPACT-II Resources

There are a number of valuable resources for FISPACT-II

Main webpage: <https://fispact.ukaea.uk/>

Wiki - <https://fispact.ukaea.uk/wiki>

Manual - <http://www.ccfc.ac.uk/FISPACT-II/documentation/UKAEA-R18001.pdf>

Forum - <https://fispact.ukaea.uk/forum/index.php>

Nuclear Data - http://www.ccfc.ac.uk/FISPACT-II/Release-4.0/nuclear_data/

GitHub - <https://github.com/fispact>

DockerHub - <https://hub.docker.com/u/fispact/>

Command Line Tool

Single executable program

Takes **input** file and produces **output** files

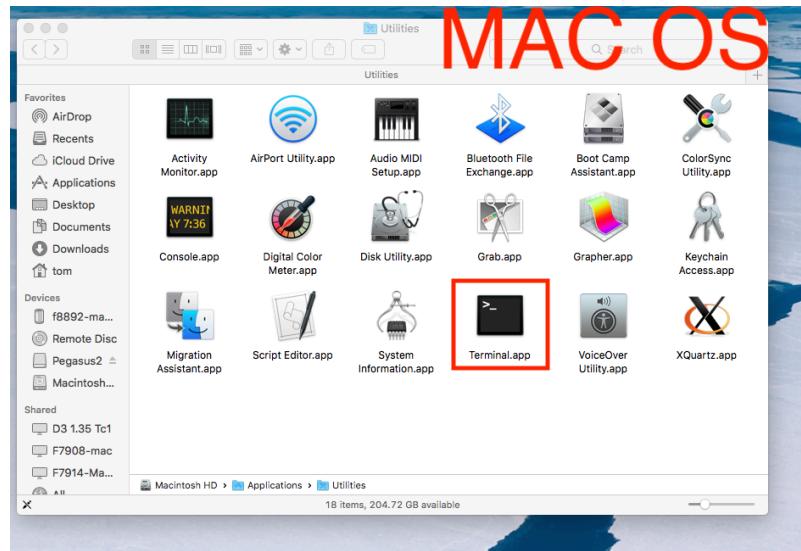


Running FISPACT-II

No GUI (yet!)
Command Line Program only!

Mac and Linux -> Terminal

Windows -> cmd



A note on the Terminal

We will use **Ubuntu** – Linux (hopefully)

If you are using **Windows** outside of this course, a few exceptions:

- Fispect has ‘.exe’ extension
- Use ‘\’ not ‘/’ for paths

You can use the same files file for Linux and Windows!

Input and output files formats are independent of operating system

Running FISPACT-II

Requires inputs as arguments
Error if no arguments added

```
[tom@mac:~/Dev/fispact]$ ./bin/fispact  
FATAL ERROR: ...
```

Correct way

```
[tom@mac:~/Dev/fispact]$ ./bin/fispact inputFile
```

Running FISPACT-II

First argument always required – input file

Second argument is optional – files file

First argument is the input file minus the extension (.i)

```
[tom@mac:~/Dev/fispact]$ ls  
run1.i          run2.i          run3.i          files
```

```
[tom@mac:~/Dev/fispact]$ ./bin/fispact run1
```

or

```
[tom@mac:~/Dev/fispact]$ ./bin/fispact run1 files
```

Creating an alias

Instead of every time calling the full binary path...

```
[tom@mac:~/Dev/fispact]$ /my/long/path/to/fispact run1
```

It can be easier to setup an environment variable or alias for fispact....

```
[tom@mac:~/Dev/fispact]$ export FISPACT=/my/long/path/to/fispact  
[tom@mac:~/Dev/fispact]$ $FISPACT run1
```

or

```
[tom@mac:~/Dev/fispact]$ alias FISPACT=''/my/long/path/to/fispact'  
[tom@mac:~/Dev/fispact]$ FISPACT run1
```

We will assume the former in examples

For long term use, it may be worth adding to **.bashrc**

If you are using the VM this is already done for you

Hands on

Let's run a simple example to see if it works...

Open the terminal and run

```
[fispactuser:~/]$ cd FISPACT-II  
[fispactuser:~/FISPACT-II]$ ls
```

```
bin      index.html    manual.pdf      nuclear_data  
system_tests   workshops      wrkdir
```

```
[fispactuser:~/FISPACT-II]$ cd wrkdir  
[fispactuser:~/FISPACT-II/wrkdir]$ $FISPACT run
```

You just ran FISPACT-II (hopefully)

Simple example

Did it run?
What are the outputs?

Not very exciting, but let's look at what we did

Let's start with the files file

What is the files file?

Declares the nuclear data libraries to be used in the run

FISPACT-II is useless without nuclear data!

And it shouldn't be neglected (see Olga's talk)

The files file is:

- Text file
- Typically named ‘files’, ‘Files’ or ‘FILES’
- But can be named anything e.g files.tendl2015.run1
- Order does not matter
- Use linux style ‘/’ for paths
- Don’t use tab to indent – causes parsing problems!

What is the files file?

```

# index of nuclides to be included
ind_nuc          /nuclear_data/TENDL2015data/tendl15_decay12_index

# Library cross section data
xs_endf         /nuclear_data/TENDL2015data/tal2015-n/gxs-709

# Library probability tables for self-shielding
prob_tab        /nuclear_data/TENDL2015data/tal2015-n/tp-709-294

# Library fission data
fy_endf         /nuclear_data/GEFY61data/gefy61_nfy

# Library spontaneous fission data
sf_endf         /nuclear_data/GEFY61data/gefy61_sf

# Library decay data
dk_endf         /nuclear_data/decay/decay_2012

# Library regulatory data
hazards          /nuclear_data/decay/hazards_2012
clear            /nuclear_data/decay/clear_2012
a2data           /nuclear_data/decay/a2_2012

# gamma attenuation data
absorp           /nuclear_data/decay/abs_2012

# fluxes file
fluxes           fluxes

# collapsed cross section data (in and out)
collapxi        COLLAPX
collapxo        COLLAPX

# condensed decay and fission data (in and out)
arrayx          ARRAYX

```

No tabs!

Key – value strings

Absolute or relative paths

~40 potential keys

Only ~10 actually needed for most cases

Nuclear Data

FISPACT-II can process a lot of different nuclear data libraries

Can process **ENDF-6** format and **EAF** format libraries

Full list of libraries can be found here:

<http://fispact.ukaea.uk/nuclear-data/downloads>

A large number are shipped with version 4.0 already

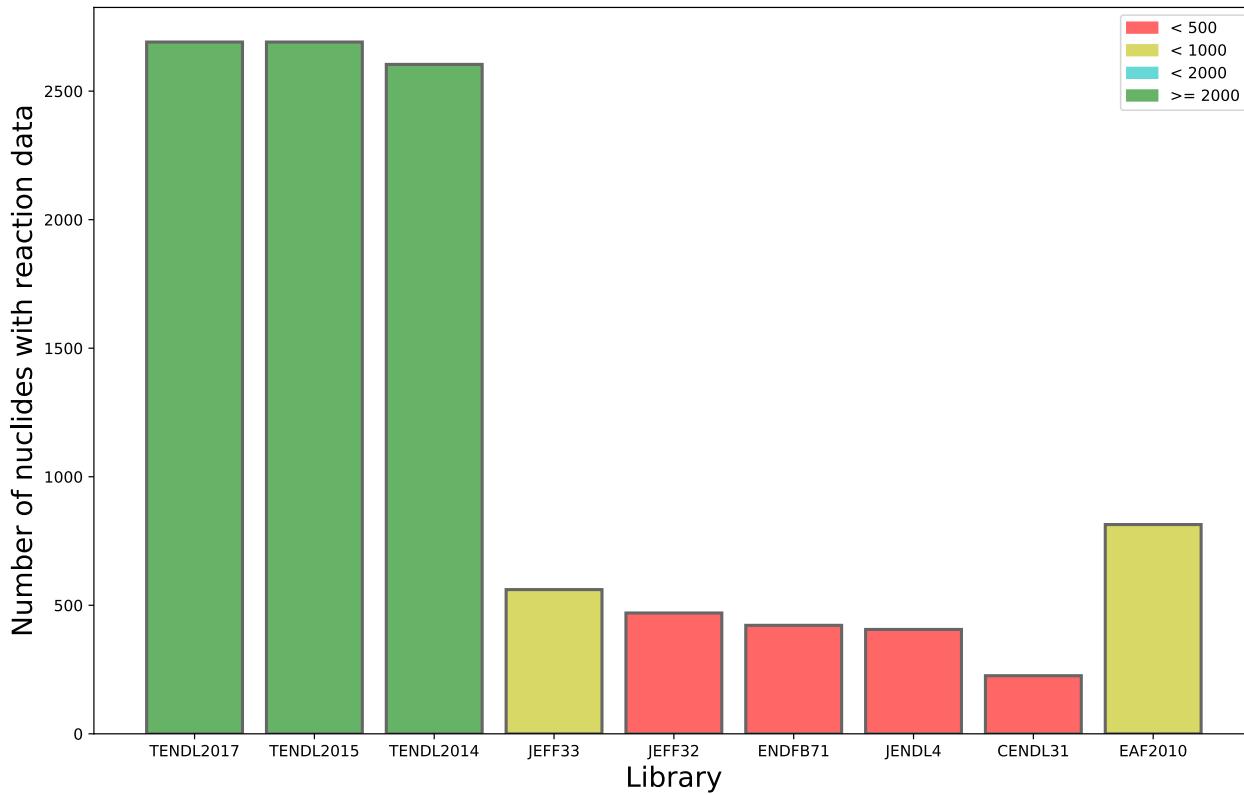
FISPACT-II uses **group wise** data, not point wise

Disclaimer: FISPACT-II does not hold responsibility for nuclear data errors/issues, it is simply a consumer of nuclear data

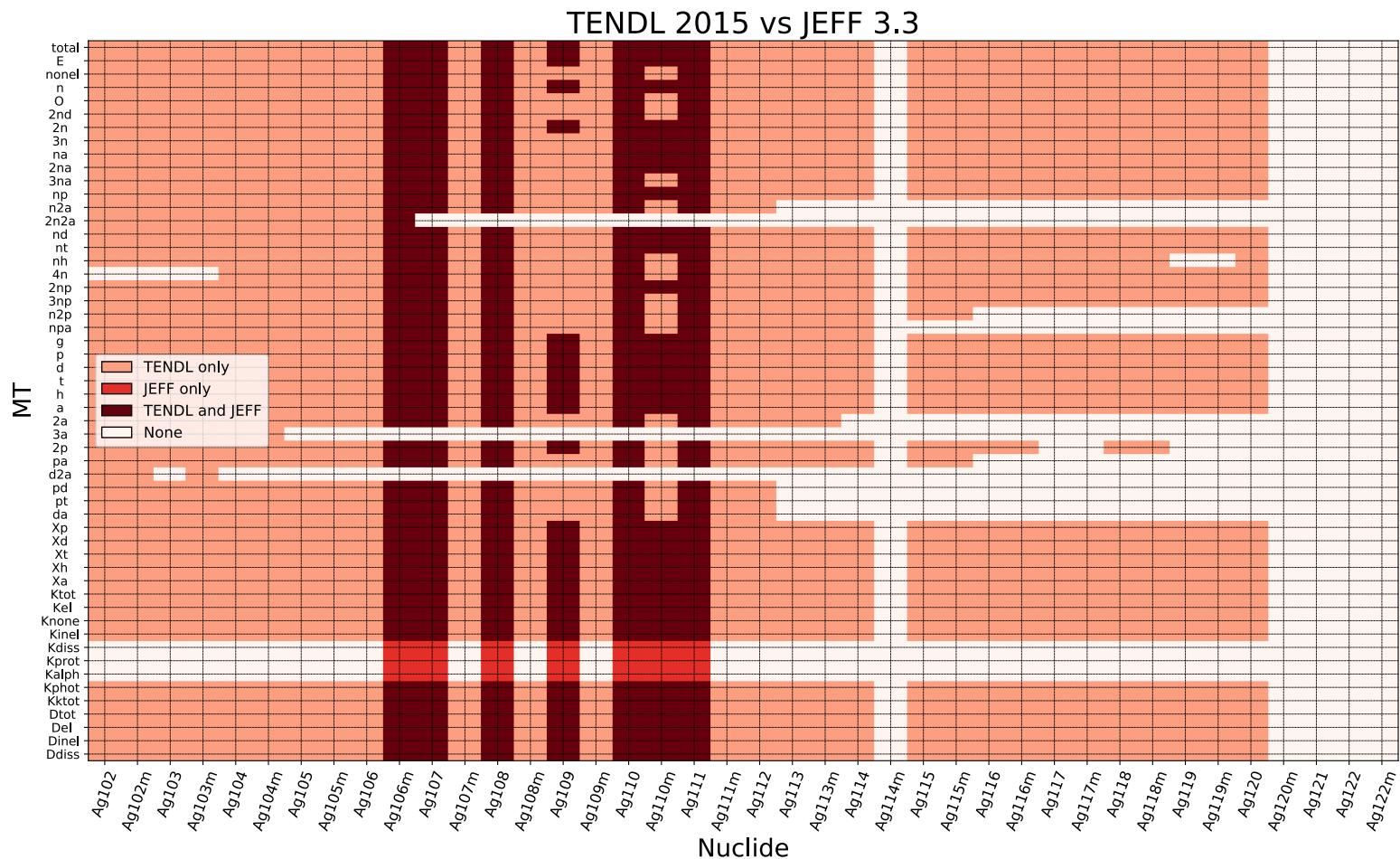
For more on nuclear data see Olga's talk, but a brief overview...

Comparison of nuclear data

Not all nuclear data was created equally



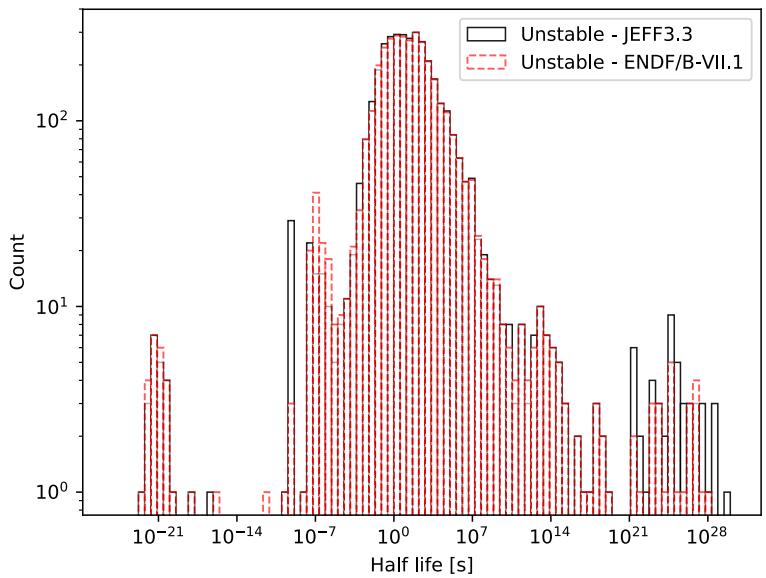
Comparison of nuclear data



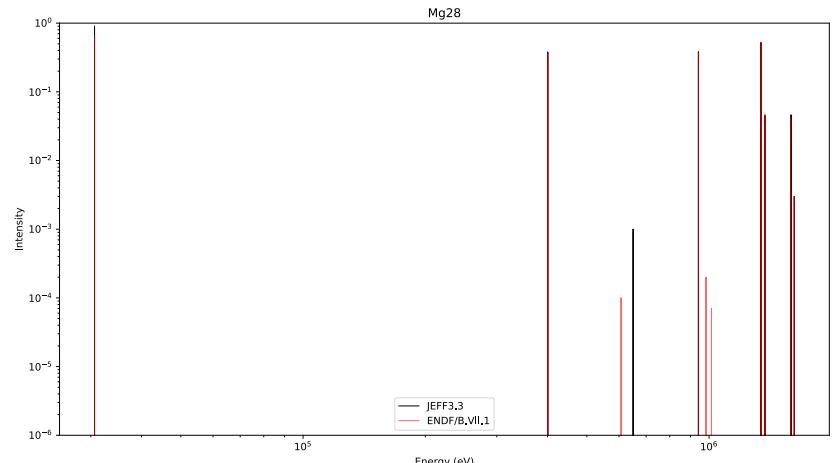
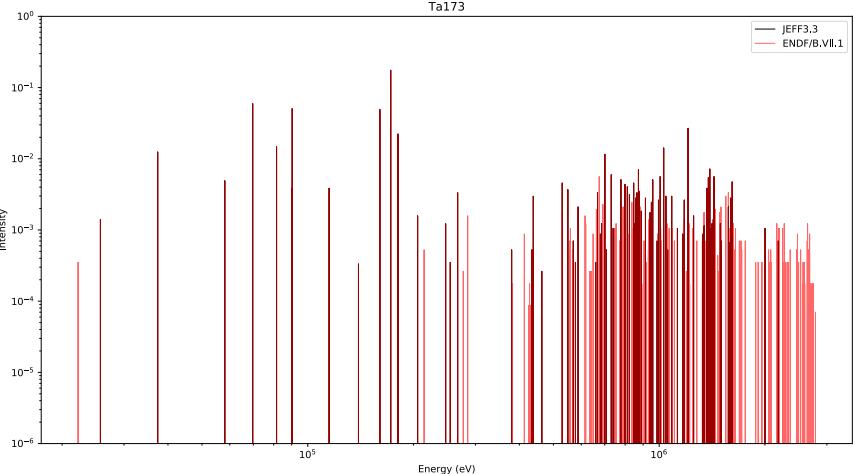
Not just cross sections

Decay data is different too!

Half lives – JEFF3.3 vs ENDF/B.VII.1



Gamma lines



Recommended nuclear data

TENDL recommended

FISPACT-II development has run in parallel with that of the
TALYS-based Evaluated Nuclear Data Library – TENDL*

*A. J. Koning, D. Rochman, et al.

https://tendl.web.psi.ch/tendl_2017/tendl2017.html

It is the largest and most complete nuclear data library!
2014, 2015 and 2017 included with FISPACT-II

Complete package tailored for all application needs:

- Nuclear fission and fusion, nuclear fuel cycle,
- Accelerator physics, isotope production,
- Material characterization, storage and life cycle,
- earth exploration, astrophysics, homeland security

Use TENDL17 + decay_2012 library if unsure about ND libraries!

Compress_xs_endf

Large nuclear data libraries can take a while to read...

- TENDL 2017 can take up to 2 minutes to read and process in it's entirety
- This can be improved by up to **60%** when using compress_xs_endf

Command line tool just like FISPACT-II

- Needs a lightweight files file
- No input file, just command line args
- Only works on ENDF-6 format, not EAF
- Only compresses reaction data, not decay data
- Binary files are platform and complier dependent (non-transferable)
- **Reduce the size from 9GB to > 2GB!**

Note: if we use binary files, we use GETXS -1 in input file and key xs_endfb must point to path of binary file

Try the exercise in 'basic' folder, to understand how to use it and create the libraries. We will then use this for the remainder of the exercises.

Check your files file

Can you tell what nuclear data libraries we used for the example we ran?

ind_nuc = Index file – tells what nuclides to include in the simulation

xs_endf = Cross section data directory (ENDF-6 format)

dk_endf = Decay data directory (ENDF-6 format)

a2data = file path of transportation values of radioactive material

clear = file path of clearance values of radioactive material

hazards = file path of ingestion and inhalation values for radionuclides

absorp = file path of air attenuation data for nuclides

= neutron fission yield data directory

= spontaneous fission yield data directory

= directory path to probability table data for self shielding

Note: some keys are different for EAF library version

i.e. **xs_endf** (ENDF) -> **crossec** (EAF)

Files file additional

The files file is also a bit more than nuclear data

It's in the name - it tells FISPACT-II where the **files are that it needs**

Non nuclear data keys that are important:

fluxes = path of fluxes file

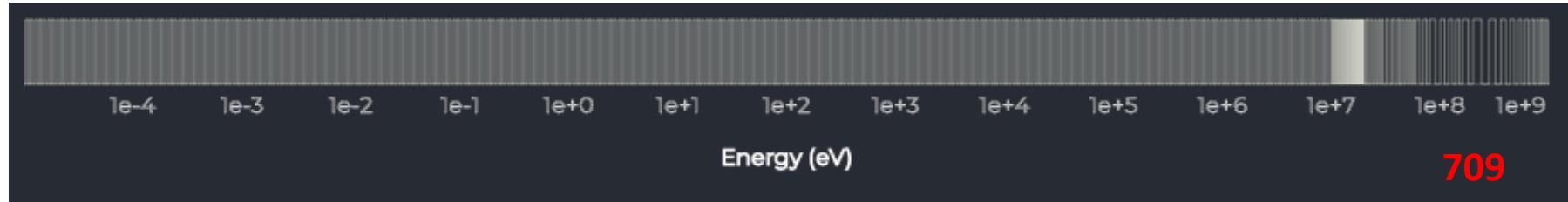
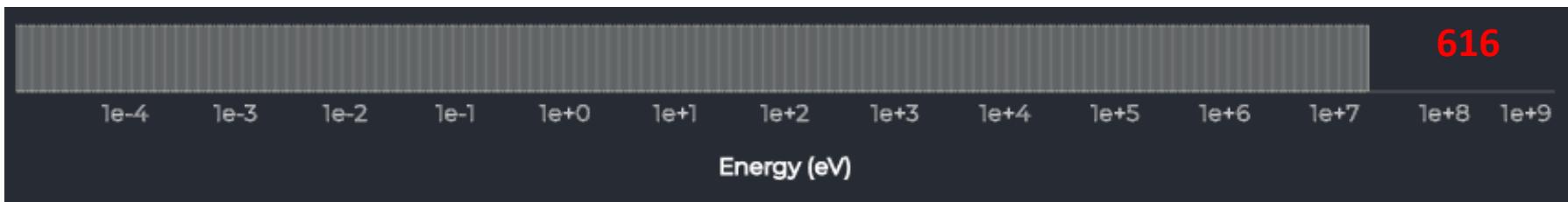
collapxi = if using a pre collapsed binary file, use this instead (more details later)

collapxo = pre collapsed binary file output path

arrayx = binary file for decay data output path

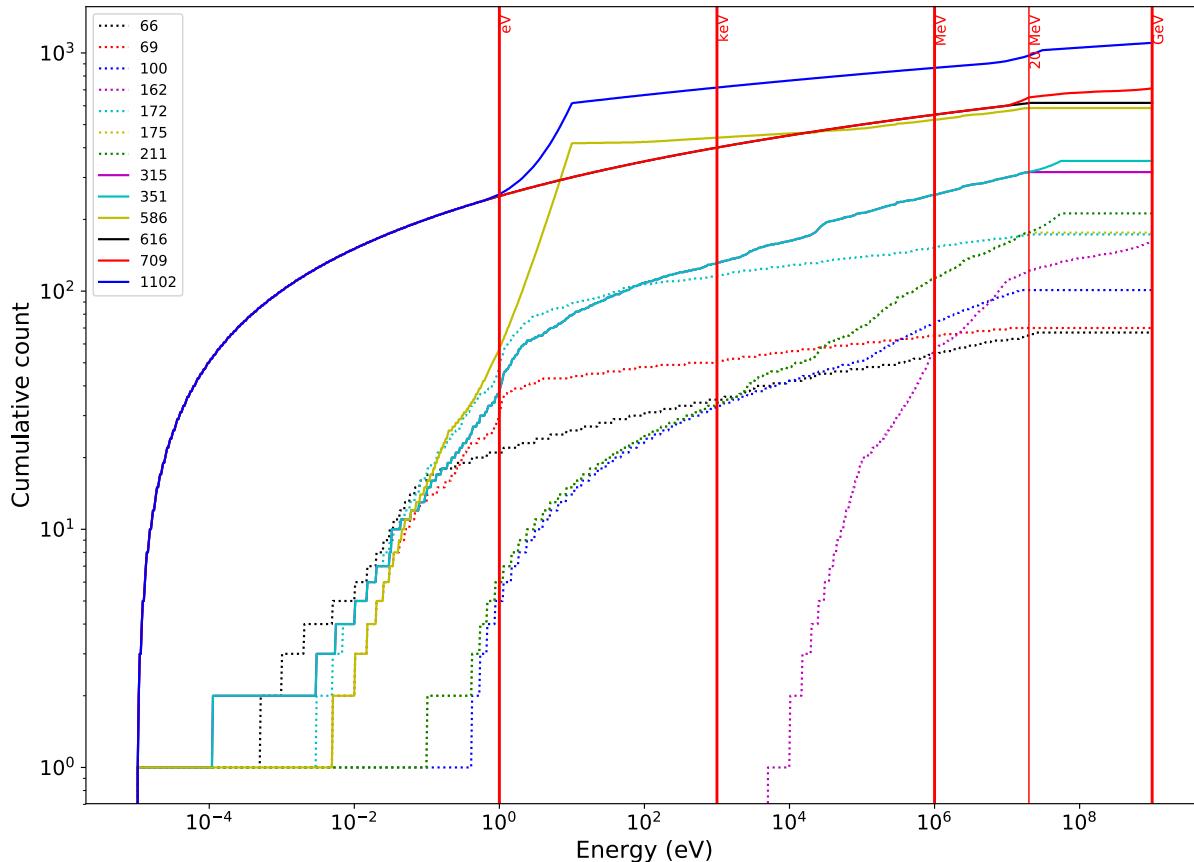
Fluxes file

- If you want to do irradiation then a fluxes file is needed
- This defines your incident particle energy spectrum
- We had to tell FISPACT-II in the ***files*** file about the ***fluxes*** file
- The file contains the group wise data as a list of values
- Note: Data must be specified in **descending** order (high to low energy)!
- Group structure must match that of nuclear data
 - E.g. If using TENDL2017 then you will need your spectra defined in 709 group.
 - Group convert (GRPCONVERT) can change group structure (more on this later)



Group structure

- Different group structures cover different energy regimes
- Some libraries are provided in a given energy range best suited for certain applications
- 709 group covers up to 1 GeV – most libraries shipped in this structure



The FISPACT-II input file

The input file governs everything for the run

- Output types – TAB, JSON, graph files
- Irradiation schedule
- Inventory
- Log options
- Verbosity options
- Solver settings
- Pathways
- Shelf Shielding
- ...

The FISPACT-II input file

- Text file
- Order does matter, unlike the files file
- No default name, unlike the files file
- Input file must have .i extension
- Use of keywords to perform different actions
- Allows comments with '<< comment here >>'

Note: There is a limit of 120 characters per line!

92 keywords in all!

```
CLOBBER
JSON
MONITOR 0
SPEK
GETDECAY 1
GETXS 1 709
FISPACT
* run name
DENSITY 1.0
FUEL 4
Y88 1e20
H3 1.3e21
Mg28 0.8e19
Ag109m 1.8e16
FLUX 0.0
ATOMS
TIME 2 DAYS ATOMS
...
```

The FISPACT-II input file

3 sections to the input file

- Control – 32 possible keywords
- Initialisation – 64 possible keywords
- Inventory – 41 possible keywords

$$32 + 64 + 41 \neq 92$$

Many keywords can be used in each input section

But some can only be used in one section!

Full list: https://fispact.ukaea.uk/wiki/FISPACT-II_keywords

Input file example

The input from the ‘simple’ exercise

```
<< CLOBBER tells fispact to overwrite existing files of same name >>
CLOBBER

<< GETXS 1 tells fispact to get the cross section data and do
collapse >>
GETXS 1 709

<< GETDECAY 1 tells fispact to get the decay data and do condense >>
GETDECAY 1

<< FISPACT signals end of control section >>
FISPACT
* example 1

<< END is required to signal end of input >>
END
* end of example
```

No initialization or inventory sections – no output

Collapse

The nuclear data collapse with GETXS takes the extensive reaction cross section data and probability tables for resonance self-shielding and folds (collapses) them with irradiation spectrum in fluxes

Produces a binary COLLAPX file of 1-group data with uncertainties

$$\bar{\sigma} = \frac{\sum_i^N \sigma_i \varphi_i}{\sum_i^N \varphi_i}$$

Printlib exercise

Printlib 4 prints all collapsed cross sections to disk
 The cross sections are given for each reaction

```

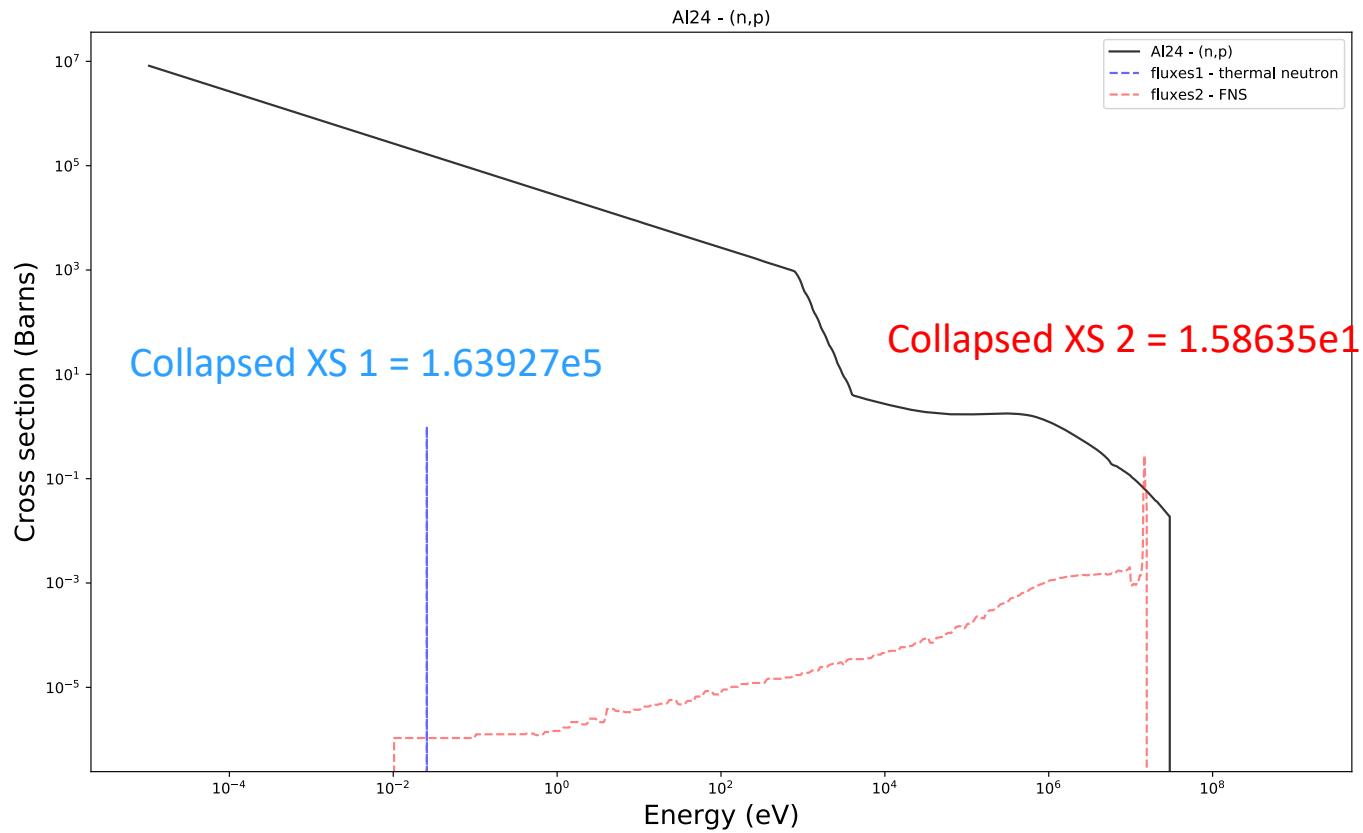
THIS RUN
timestamp:          09:52:10 5 June 2018
fileroot :          ex2
name of FILES file: files
FISPACT title:     * example 1

See the ex2.log file
and summary details at the end of this file for further information on files used by this run
=====
```

C O L L A P S E D C R O S S S E C T I O N S I N B A R N S							
NOTE: The diagnostic cross-sections for dpa (n,D--) and for kerma (n,K--) reactions are given in barns-ev.							
The cross section for the specified reaction is given in barns, followed by the error in percent.							
parent reaction	daughter	cross section	+error	parent reaction	daughter	cross section	+error
H 1 (n,Dtot)		1.73063E+02+-	0.00000E+00	H 1 (n,Ddiss)		1.73063E+02+-	0.00000E+00
H 1 (n,Ktot)		7.32927E+05+-	0.00000E+00	H 1 (n,Kel)		3.84884E-01+-	0.00000E+00
H 1 (n,Knone)		7.32926E+05+-	0.00000E+00	H 1 (n,Kphot)		7.32493E+05+-	0.00000E+00
H 1 (n,Kktot)		7.32927E+05+-	0.00000E+00	H 1 (n,Xd)		3.29462E-01+-	0.00000E+00
H 1 (n,total)		3.02774E+01+-	0.00000E+00	H 1 (n,E) H 1		2.99480E+01+-	2.96233E-01
H 1 (n,g) H 2		3.29417E-01+-	2.55297E+00	H 2 (n,Dtot)		5.18112E-01+-	0.00000E+00
H 2 (n,Ddiss)		5.18112E-01+-	0.00000E+00	H 2 (n,Ktot)		3.14013E+03+-	0.00000E+00
H 2 (n,Kel)		4.82672E-02+-	0.00000E+00	H 2 (n,Knone)		3.14009E+03+-	0.00000E+00
H 2 (n,Kktot)		3.53743E+00+-	0.00000E+00	H 2 (n,Xt)		5.01819E-04+-	0.00000E+00
H 2 (n,total)		4.22333E+00+-	1.50000E+00	H 2 (n,nonel)		5.01748E-04+-	0.00000E+00
H 2 (n,E) H 2		4.22283E+00+-	2.00000E+00	H 2 (n,g) H 3		5.01748E-04+-	3.00000E+00
H 3 (n,Ktot)		1.87166E-02+-	0.00000E+00	H 3 (n,Kel)		1.87166E-02+-	0.00000E+00
H 3 (n,Kktot)		1.87166E-02+-	0.00000E+00	H 3 (n,total)		1.93838E+00+-	0.00000E+00
H 3 (n,E) H 3		1.93838E+00+-	0.00000E+00	H 3 (n,Dtot)		1.05921E+07+-	0.00000E+00
He 3 (n,Ddiss)		1.05921E+07+-	0.00000E+00	He 3 (n,Ktot)		4.02870E+09+-	0.00000E+00
He 3 (n,Kel)		3.72152E-02+-	0.00000E+00	He 3 (n,Knone)		4.02870E+09+-	0.00000E+00
He 3 (n,Kktot)		4.02870E+09+-	0.00000E+00	He 3 (n,Xp)		5.274488E+03+-	0.00000E+00
He 3 (n,Xt)		5.274488E+03+-	0.00000E+00	He 3 (n,Xa)		5.45646E-05+-	0.00000E+00
He 3 (n,total)		5.27801E+03+-	0.00000E+00	He 3 (n,p) H 3		5.27415E+03+-	0.00000E+00
He 3 (n,E) He 3		3.85381E+00+-	0.00000E+00	He 3 (n,g) He 4		5.45572E-05+-	0.00000E+00
He 4 (n,Ktot)		7.12495E-03+-	0.00000E+00	He 4 (n,Kel)		7.12495E-03+-	0.00000E+00
He 4 (n,Kktot)		7.12495E-03+-	0.00000E+00	He 4 (n,total)		8.62086E-01+-	0.00000E+00
He 4 (n,E) He 4		8.62086E-01+-	1.10339E+00	Li 6 (n,Dtot)		5.61809E+06+-	0.00000E+00
Li 6 (n,Ddiss)		5.61809E+06+-	0.00000E+00	Li 6 (n,Ktot)		4.45511E+09+-	0.00000E+00
Li 6 (n,Kel)		4.91128E-03+-	0.00000E+00	Li 6 (n,Knone)		4.45511E+09+-	0.00000E+00
Li 6 (n,Kktot)		4.45483E+09+-	0.00000E+00	Li 6 (n,Xt)		9.31233E+02+-	0.00000E+00
Li 6 (n,Xa)		9.31233E+02+-	0.00000E+00	Li 6 (n,total)		9.31920E+02+-	0.00000E+00
Li 6 (n,nonel)		9.31143E+02+-	0.00000E+00	Li 6 (n,t) He 4		9.31105E+02+-	1.35019E-01
Li 6 (n,E) Li 6		7.76778E-01+-	2.00988E+00	Li 6 (n,g) Li 7		3.81990E-02+-	8.00000E+00

Printlib exercise

This can also be done by hand, and you should get the same result.
 Getting the group wise XS data can be done using another tool with
FISPACT-II =>
extract_xs_endf (more on this later).



Collapse and condense

Common practice is to separate out collapse and condense

collapse.i input file – GETXS 1 <group>

condese.i input file – GETDECAY 1

Computationally expensive to do these processes.

If flux is not changing, then no need to redo collapse.

We just did the collapse in the previous example.

We will do this the ‘inventory’ basic example

```
[tom@mac:~/workshop/exercises/basic/inventory]$ $FISPACT collapse
[tom@mac:~/workshop/exercises/basic/inventory]$ $FISPACT condense
[tom@mac:~/workshop/exercises/basic/inventory]$ $FISPACT input1
[tom@mac:~/workshop/exercises/basic/inventory]$ $FISPACT input2
```

Condense

The nuclear data condense via GETDECAY produces a binary ARRAYX file from the extensive decay and fission yield data

May also require an irradiation spectrum (fluxes) to calculate spectrum-dependent fission yields but is not normally the case

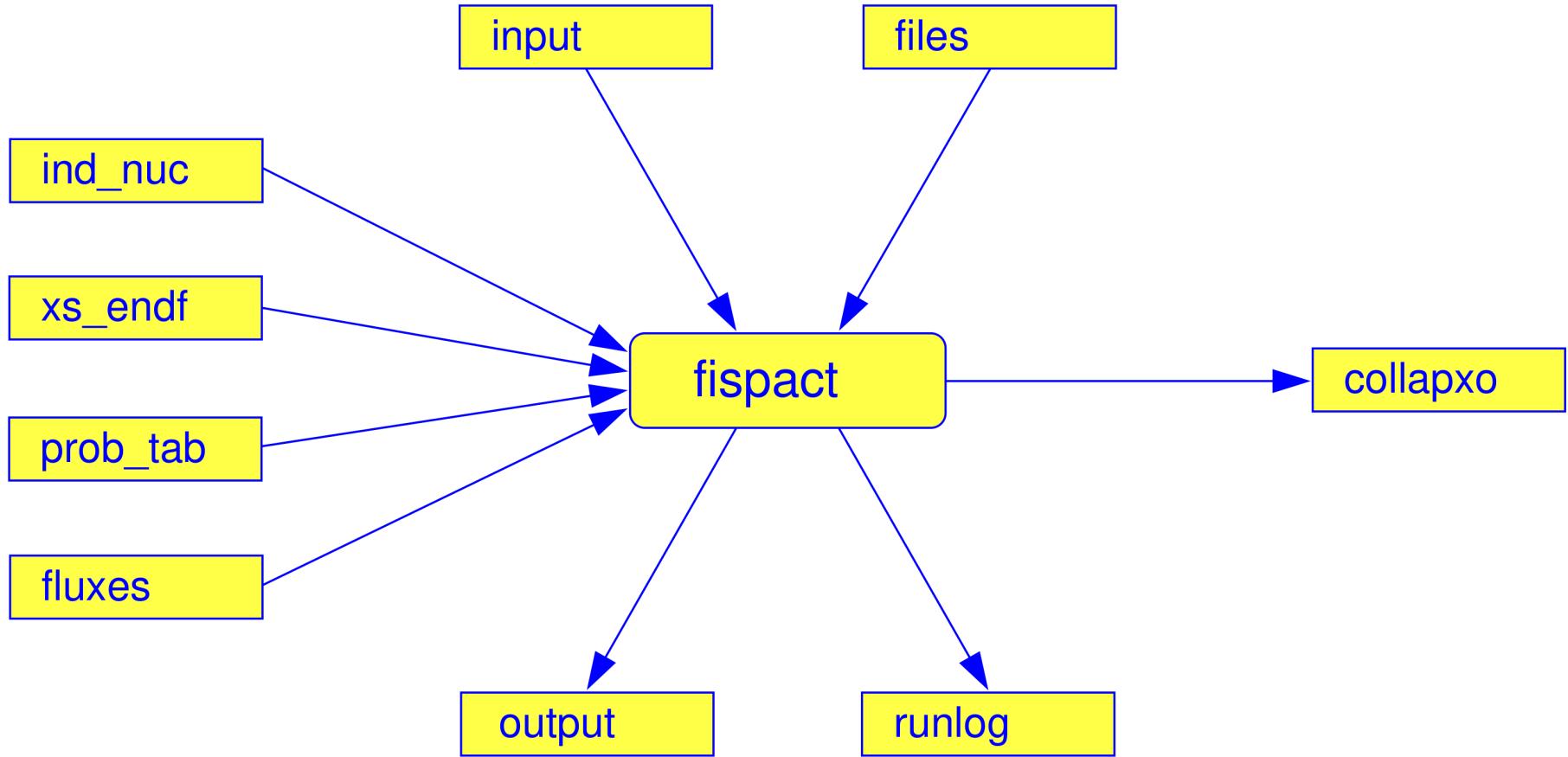
Intro Summary

In the basic exercises we will use the fluxes and nuclear data to calculate all of the effective cross sections and prepare the decay data for time-dependent simulations.

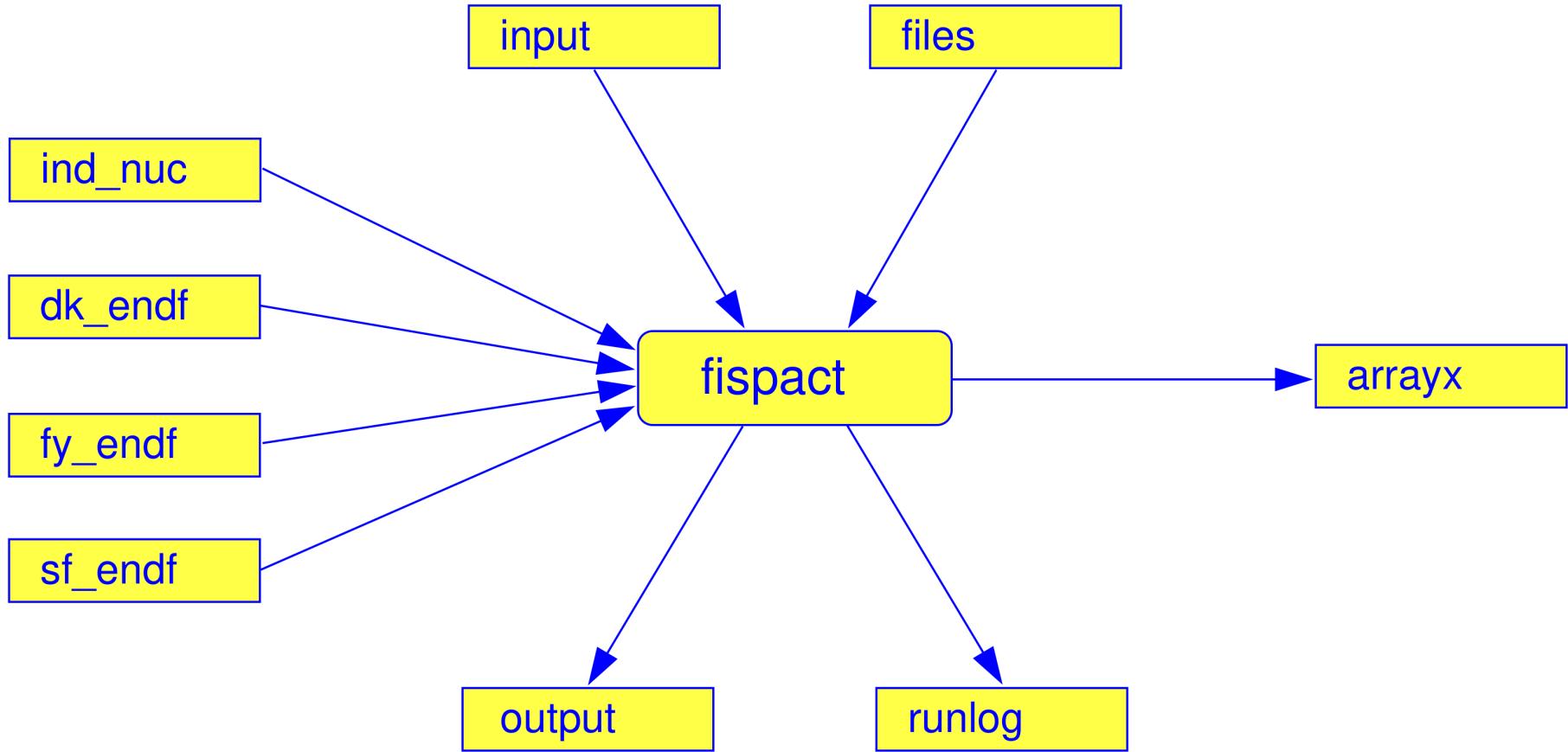
Next we will provide the initial conditions and then simulate some irradiation and cooling phases.

We will then use FISPACT-II to return various quantities such as activity, (spectroscopic) heat, dose rates, emitted spectra and various other quantities.

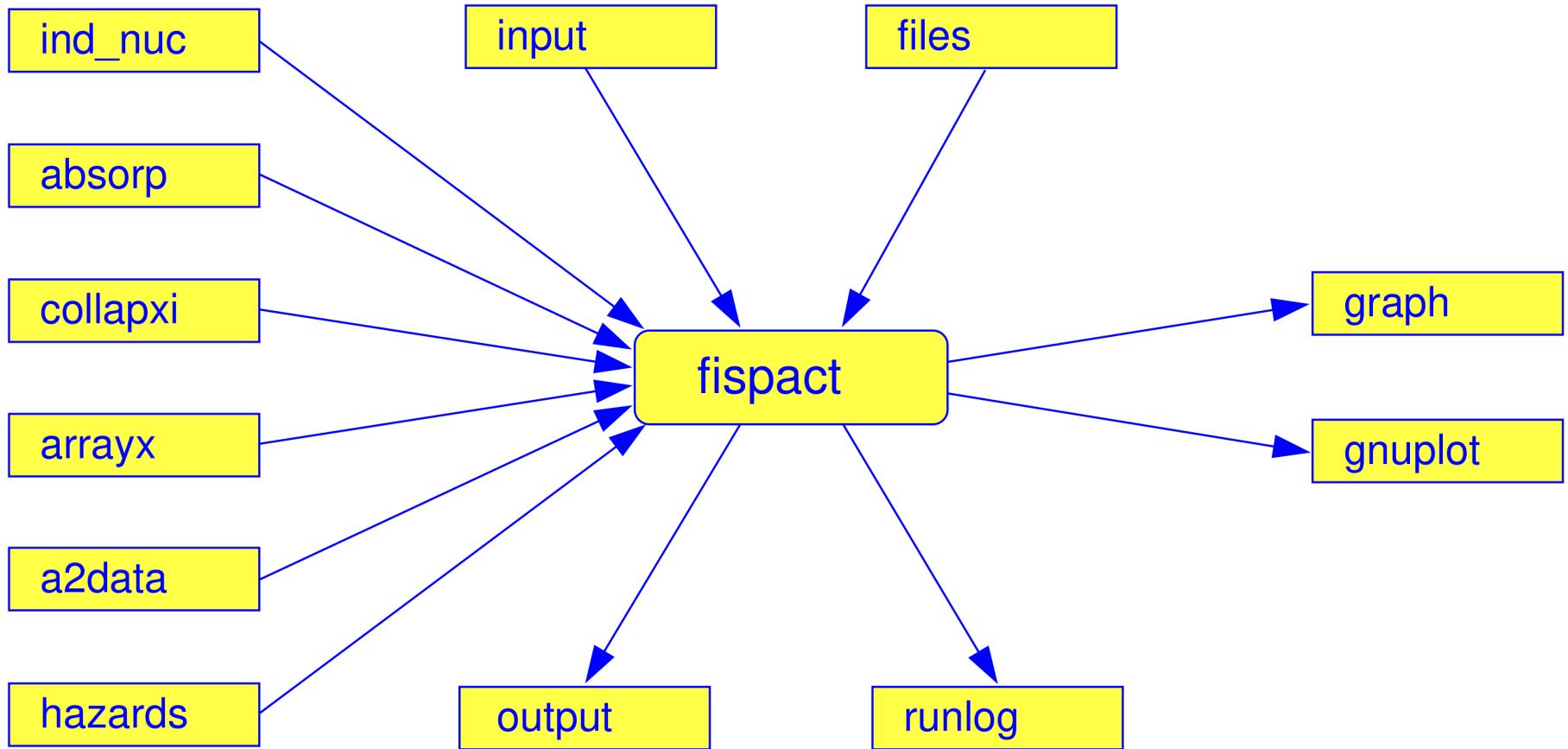
Collapse Summary



Condense Summary



Inventory Summary



FNS Inconel example

One of the exercises and included in the getting started.

It contains 4 inputs:

- Collapse
- Condense
- Printlib
- Inventory

The first 3 are similar to the previous two exercises

The inventory input is then used to simulate our irradiation and cooling phases

FNS Inconel example

Read the collapsed and condensed data

```
<< -----set initial switches and get nuclear data----- >>  
CLOBBER  
JSON  
GETXS 0  
GETDECAY 0  
FISPACT  
* FNS 5 Minutes Inconel-600
```

FNS Inconel example

Initial conditions

```
DENSITY 8.42
```

```
MASS 1.0E-3 4
```

```
NI 75.82
```

```
MN 0.39
```

```
FE 7.82
```

```
CR 15.97
```

1 gram of **Inconel 600** with density of 8.42 g/cc
Inconel 600 – 75.82% Ni, 15.97% Cr, 7.82% Fe and 0.39% Mn

FNS Inconel example

Initial conditions part 2

```
MIND 1E3
GRAPH 1 2 1 3
UNCERTAINTY 2
HALF
HAZARDS
```

MIND – ignore nuclides with inventory < 1000 atoms

GRAPH – create gnuplot data and file

UNCERTAINTY – pathways analysis

HALF – output half lives

HAZARDS – output ingestion and inhalation

FNS Inconel example

Irradiation phase

```
<< -----irradiation phase----- >>
FLUX 1.116E+10
ATOMS
TIME 5.0 MINS
ATOMS
```

FLUX – specifies the energy integrated flux ($\text{cm}^{-2}\text{s}^{-1}$)

ATOMS – solves rate equation matrix + outputs initial inventory

TIME – specify **time interval**, not elapsed time

ATOMS – solves rate equation matrix + output inventory after 5 mins

Note: Without either ATOMS, STEP, or SPECTRUM, nothing will be calculated!

For less verbose output, use STEP, instead of ATOMS.

FNS Inconel example

Cooling phase

```
<< -----cooling phase----- >>
FLUX 0.
ZERO
TIME    36 ATOMS
TIME    15 ATOMS
TIME    16 ATOMS
...
END
* END
```

FLUX – sets the flux to zero (cooling)

ZERO – reset time to zero, start of cooling

TIME – output the inventory at time intervals

END – finish simulation

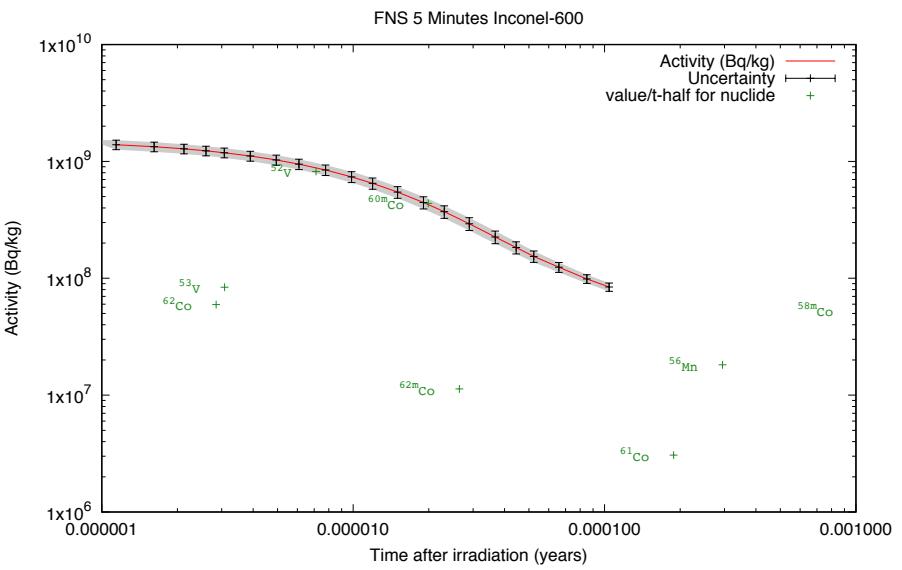
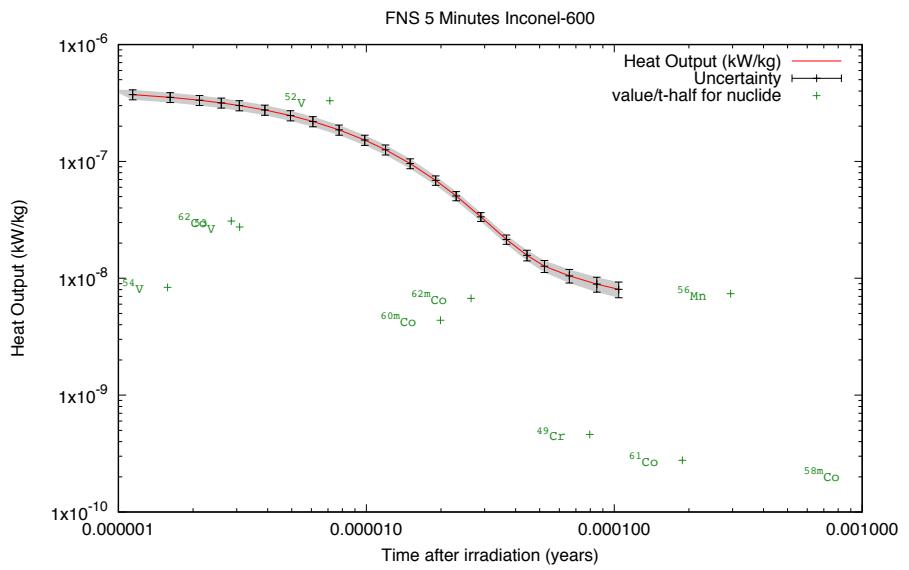
Inventory exercise

FNS Inconel 600:

- Can you run all 4 inputs in the correct order and get it to produce the total heat as a function of time?
- Can you do this using just one input file?
- Now try running the same example but now producing the total activity as a function of time?
 - Total activity=1 for GRAPH option

Inventory exercise

Graph outputs



file name = inventory.gra run timestamp = 14:59:04 14 June 2019

file name = inventory.gra run timestamp = 15:08:52 14 June 2019

Inventory exercise

How do we do custom analysis?

- Generated graphs and plots are good, but limited
- We can look and copy values from the output file, but difficult for big simulations
- Output file can be tricky to parse
- Other output files – **TAB** and **JSON** to help file reading issues
- Python tool – **pypact**, can be used to easily parse FISPACT-II output files

All this and more in the upcoming talks....

Summary

We should now know

- How to construct a files file, fluxes file and input file
- How to run fispact
- How to get some basic outputs

Advanced usage, later

- JSON + pypact
- Other tools (compress, extract, group convert,...)
- Docker, API, and much more...