

FISPACT-II

Advanced Usage

Thomas Stainer
United Kingdom Atomic Energy Authority

FISPACT-II workshop
June 13-15, 2018, OECD/NEA, Paris



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



FISPACT-II Open Source Tools

There are a number of open source tools for FISPACT-II

The most useful for FISPACT-II analysis is **pypact**

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

GitHub



FISPACT-II Output Format

- Fairly easy to look inside and read a value
- Hard to script and parse values
- Output files can be quite large (often > 5MB)
- Grep will only take you so far

Two solutions → **pypact** and/or **JSON** output

FISPACT-II JSON

FISPACT-II 4.0 now has a [JSON](#) output

```
{  
    "name" : "json",  
    "description" : "I am JSON",  
    "id" : 4,  
    "value" : 4.3e+9  
}
```

Simply use **JSON** keyword in control section

We did this in FNS Inconel example

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

Why JSON?

- Human readable
- Light weight
- Easy to parse
- Very popular format
- Supported by many languages – Python, C++, Fortran, js, ...

FISPACT-II JSON

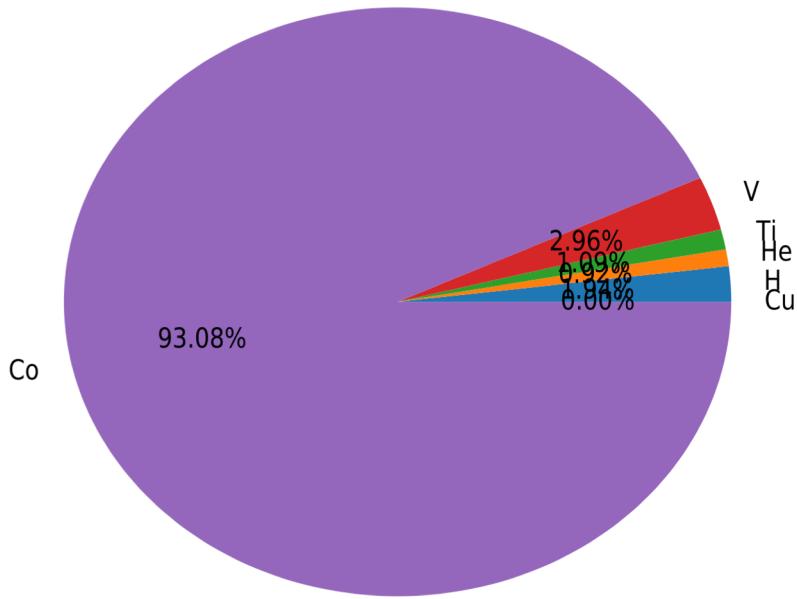
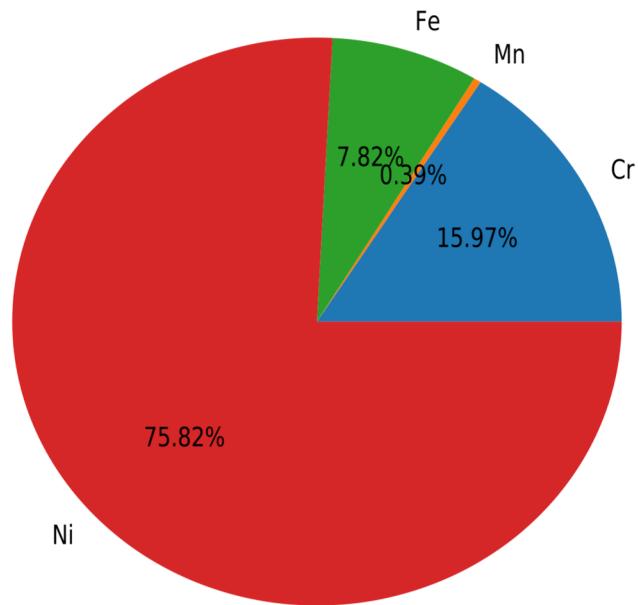
Supports most common output data

- Run data - timestamp, run name, flux name
- Inventory data - list of timesteps, each containing
 - Irradiation and cooling time
 - Flux amplitude
 - Heat - alpha, beta, gamma
 - Ingestion and inhalation
 - Dose
 - **Nuclides** → the inventory

Exercise 4

Use the JSON output to get the elemental composition

Exercise 4



Left: initial

Right: final (excluding initial)

Exercise 4

Could alternatively have used TAB1 keyword to get this data
Contains columns of nuclide with grams only!

Alternatively, we can use `pypact`

Pypact

Python package to parse fispact output files

Advisable to use JSON - but pypact works on standard output file (*.out) also

Will work on FISPACT-II files pre version 4.0

Makes scripting easier and analyse FISPACT-II outputs

Easy to install via pip (python package management system)

```
pip3 install pypact
```

First let's do a very simple example with pypact

```
# import the library
import pypact as pp

# read the output file
with pp.Reader('inventory.out') as output:
    print(output.run_data.timestamp)
    print(output.run_data.run_name)
```

Another more interesting example

```
import pypact as pp
import matplotlib.pyplot as plt

time = []
data = []

with pp.Reader('inventory.out') as output:
    for t in output.inventory_data:
        if not t.isirradiation():
            time.append(t.currenttime)
            data.append(t.gamma_heat)

plt.plot(time, data)
plt.show()
```

Pypact

Can also use the JSON file

```
import pypact as pp
import matplotlib.pyplot as plt

time = []
data = []

output = pp.Output()
with open('inventory.json') as f:
    output.json_deserialize(f.read())
    for t in output.inventory_data:
        if not t.isirradiation():
            time.append(t.currenttime)
            data.append(t.gamma_heat)

plt.plot(time, data)
plt.show()
```

Exercise 5

Use pypact to get all the output times from the output file

Pypact

It can be done in 3 lines

```
import pypact as pp

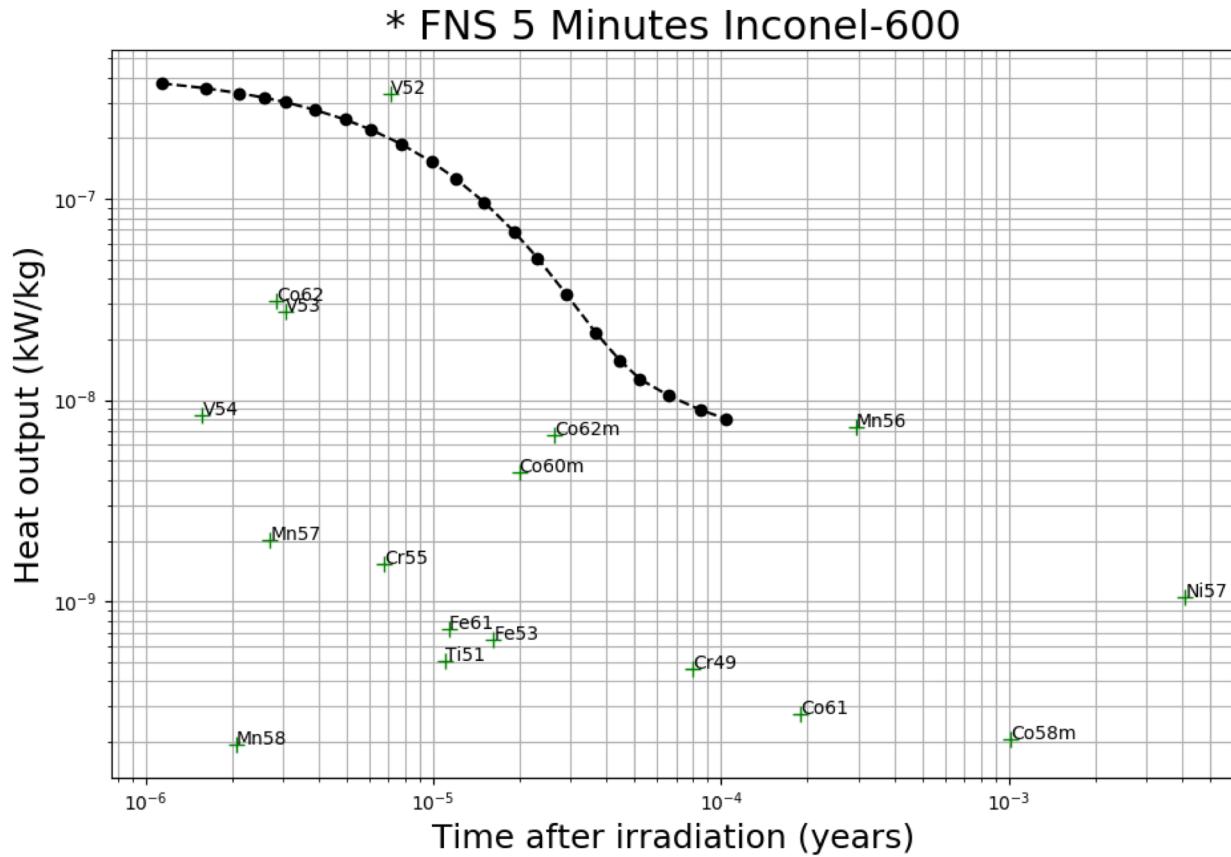
with pp.Reader('inventory.out') as o:
    [print("Time {} (secs)".format(t.currenttime)) for t in o.inventory_data]
```

We will do another analysis using pypact to produce a similar plot
as in FNS Inconel for gnuplot

Exercise 6

Use pypact to produce a gnuplot style plot using .out and .json files

Exercise 6



Pypact Summary

Status

- Handle legacy FISPACT-II output files and JSON files
- Process nuclides, doses, ingestion, burnup, ..
- Create and read files files
- Contains all fispact keywords and group structures built in - 66, 69,, 709, 1102
- Simple plotting with analysis extension

Future

- Pathways and uncertainties
- Gamma spectrum
- Optimizations and API improvements

Feedback welcome!

Engine for running containers...

A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it

Can be thought of as a lightweight Virtual Machine.



FISPACT-II with Docker

We utilise Docker to bundle all dependencies in a single image

Including nuclear data (in binary format) - faster!

To read binary data we use GETXS -1

Docker not installed on your machines, but you can run it with

```
docker run -it -v /opt/fispact/bin/ubuntu/16.04/ifort18/fispact:/fispact
fispact/ubuntu:16.04_gfortran_5_data bash
```

FISPACT-II with Docker

Then we have ubuntu with nuclear data!

```
root@580a028b793c:/# ./fispact
FATAL ERROR: unable to find FILES file
              also unable to find input file
```

RUN TERMINATED

Either provide the FILES name as a second argument, or
provide a file named 'files', 'Files' or 'FILES'
in the present working directory, or
provide input file .i
in the present working directory.

FISPACT-II tools

There are some other tools shipped with FISPACT-II 4.0

- `compress_xs_endf` - makes binary format versions of nuclear data
- `extract_xs_endf` - extract data from nuclear libraries
- `listreactions`
- `makenuclideindex`

Not covered in this course but in the manual.

FISPACT-II Advanced Summary

- Open source tools – in active development
- Many ways to analyse FISPACT-II output
 - GNU plotting built in
 - TAB files can make life easier but
 - JSON output has many parsers ready - C++, Fortran, Python, js, ...
- Pypact
 - Supports both JSON and standard output file formats
 - Very easy to script plots and analysis
 - Can read and plot flux file
 - Contains tool to create files file
 - Contains all keywords for FISPACT-II
- Docker
 - Many images ready made with nuclear data inside
 - Self contained environment to run fispact easily
 - Nuclear data in binary format – performance improvement