Applications of Nuclear Data: Introduction to FISPACT-II

ICTP-IAEA Workshop on the Evaluation of Nuclear Data for Applications

10 October 2017

Michael Fleming, Mark Gilbert, Jean-Christophe Sublet
michael.fleming@ukaea.uk
mark.gilbert@ukaea.uk
jean-christophe.sublet@ukaea.uk

UK Atomic Energy Authority

UK Atomic Energy Authority

The Abdus Salam International Centre for Theoretical Physics
www.ictp.it

IAEA International Atomic Energy Agency
Atoms for Peace

Agenda

Start-up

Introduction

- Background
- Basic code functionality

Code practicalities

- Overview
- Basic execution
- Input files
- Nuclear data handling
- Keywords

Login

We will be using the Linux Mint 18 operating system installed on these machines. If you are not familiar with non-Windows machines don't worry

Log into the machine now using your username/password on your badge

Check disks

We will be using the FISPACT-II code in this workshop, which is located on a shared space
`/home/nfs1/smr3151/FISPACT-II-3-20`
The nuclear data must be transferred to your local machine so please do the following:

- Check that your scratch space is not full:

```
machine:~ user$ df -h /* | grep scratch
```

- If the scratch space has less than 50 GB remaining *let me know* and we'll clear some space

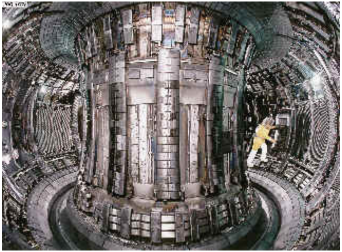
Installation

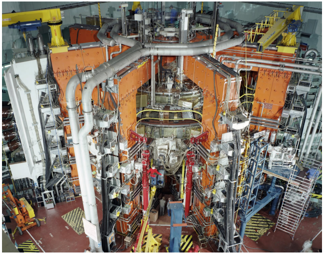
To move the essential FISPACT-II code components onto your local scratch space, run the custom installation script provided in:

```
source /home/nfs1/smr3151/scripts/install_fispact.sh
  Copying getting_started directory to:  /scratch/
  ...DONE
  Copying nuclear data to: /scratch/
  ...DONE
  Unpacking nuclear data
```

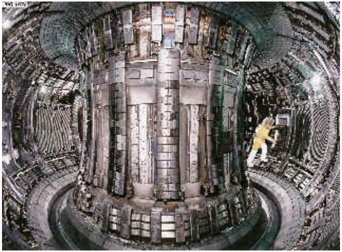
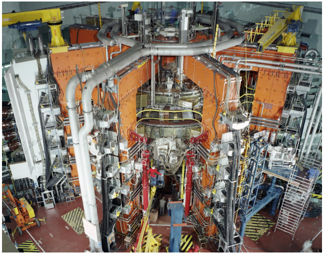
This will install the nuclear data onto your local `/scratch/smr3151/FISPACT-II-3-20/`
It will also add the `fispact` executable to your `PATH` and the `getting_started` directory to your scratch space

- The UK Atomic Energy Authority (UKAEA) includes the Culham Centre for Fusion Energy (CCFE), which hosts (amongst other things) the Joint European Torus (JET):





- The UK Atomic Energy Authority (UKAEA) includes the Culham Centre for Fusion Energy (CCFE), which hosts (amongst other things) the Joint European Torus (JET):

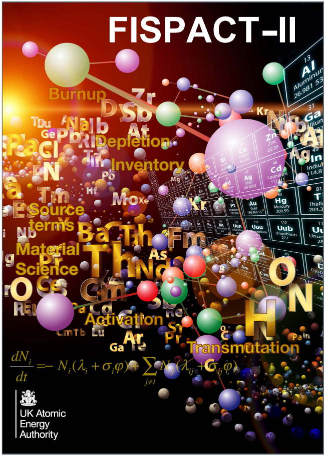
- Fusion utilises the D-T plasma which, as was described in a previous lecture, is an excellent source of 14 MeV neutrons

- The UKAEA developed FISPIN as an inventory code for fission fuel modelling

- The UKAEA developed FISPIN as an inventory code for fission fuel modelling
- In the 80s and 90s it was appreciated that more sophisticated nuclear simulation codes and nuclear data was required
  - Birth of FISPACT code + EAF data = EASY

- The UKAEA developed FISPIN as an inventory code for fission fuel modelling
- In the 80s and 90s it was appreciated that more sophisticated nuclear simulation codes and nuclear data was required
  - Birth of FISPACT code + EAF data = EASY
- Unsupportable growth of ‘physicist’ code and need to accommodate next-generation TENDL data led to re-development as FISPACT-II in 2009

- The UKAEA developed FISPIN as an inventory code for fission fuel modelling
- In the 80s and 90s it was appreciated that more sophisticated nuclear simulation codes and nuclear data was required
  - Birth of FISPACT code + EAF data = EASY
- Unsupportable growth of 'physicist' code and need to accommodate next-generation TENDL data led to re-development as FISPACT-II in 2009
- FISPACT-II now released through OECD-NEA Data Bank, ORNL RSICC and UKAEA - deprecated EASY not supported nor recommended



- multiphysics platform for predicting the inventory changes in materials under both neutron and charged-particle interactions
  - calculates activation, transmutation, burn-up, dpa, PKAs, gas production, etc.
- employs the most up-to-date international nuclear data libraries containing:
  - nuclear reaction data (reaction cross sections)
  - radioactive decay data (half-lives and decay schemes)
  - fission yield data (ratios)

Introduction
Bateman equations

$$\frac{dN_i}{dt} = \underbrace{-N_i(\lambda_i + \sigma_i\phi)}_{\text{loss}} + \sum_{j\neq i} \underbrace{N_j(\lambda_{ji} + \sigma_{ji}\phi)}_{\text{creation}}$$

- coupled differential equations
  - one equation for each nuclide i at concentration N_i
  - solved numerically by FISPACT-II (using Livermore ODE solver, LSODE) and used to update material composition

Introduction

Bateman equations

$$\frac{dN_i}{dt} = \underbrace{-N_i(\lambda_i + \sigma_i\phi)}_{\text{loss}} + \sum_{j\neq i} \underbrace{N_j(\lambda_{ji} + \sigma_{ji}\phi)}_{\text{creation}}$$

- coupled differential equations
  - one equation for each nuclide i at concentration N_i
  - solved numerically by FISPACT-II (using Livermore ODE solver, LSODE) and used to update material composition
- σ_{ji}: energy-dependent reaction cross sections for $j \rightarrow i$ reactions (e.g. (n,γ), (n,α), (n,2n), etc.) from nuclear libraries collapsed with (normalised) neutron energy spectra from neutron transport; σ_i is sum over all $i \rightarrow j$ reactions

$$\frac{dN_i}{dt} = \underbrace{-N_i(\lambda_i + \sigma_i\phi)}_{\text{loss}} + \sum_{j\neq i} \underbrace{N_j(\lambda_{ji} + \sigma_{ji}\phi)}_{\text{creation}}$$

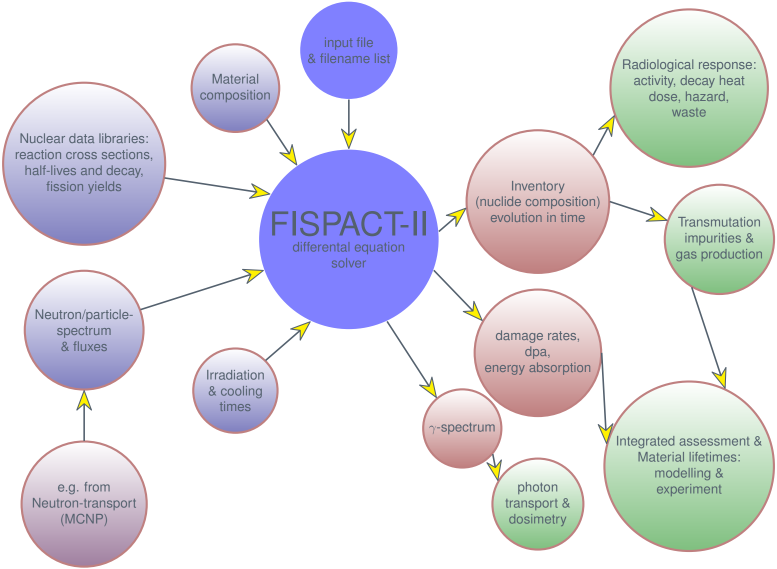
- coupled differential equations
  - one equation for each nuclide i at concentration N_i
  - solved numerically by FISPACT-II (using Livermore ODE solver, LSODE) and used to update material composition
- σ_{ji}: energy-dependent reaction cross sections for $j \rightarrow i$ reactions (e.g. (n,γ), (n,α), (n,2n), etc.) from nuclear libraries collapsed with (normalised) neutron energy spectra from neutron transport; σ_i is sum over all $i \rightarrow j$ reactions
- decay constants λ_i, λ_{ji} (from decay library of measurements)

$$\frac{dN_i}{dt} = \underbrace{-N_i(\lambda_i + \sigma_i\phi)}_{\text{loss}} + \sum_{j\neq i} \underbrace{N_j(\lambda_{ji} + \sigma_{ji}\phi)}_{\text{creation}}$$

- coupled differential equations
  - one equation for each nuclide i at concentration N_i
  - solved numerically by FISPACT-II (using Livermore ODE solver, LSODE) and used to update material composition
- σ_{ji}: energy-dependent reaction cross sections for $j \rightarrow i$ reactions (e.g. (n,γ), (n,α), (n,2n), etc.) from nuclear libraries collapsed with (normalised) neutron energy spectra from neutron transport; σ_i is sum over all $i \rightarrow j$ reactions
- decay constants λ_i, λ_{ji} (from decay library of measurements)
- total fluxes ϕ from radiation transport simulations, experiments, operational scenarios, etc.

Introcution

Flow of data

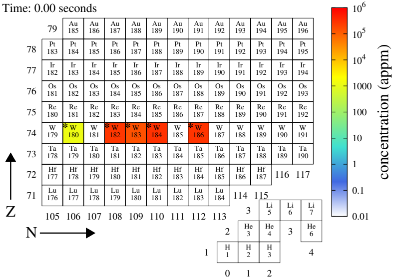


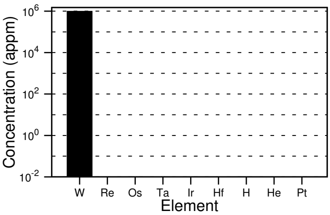
Introduction

Radiological outputs

- Activity measured in becquerels (Bq) – number of disintegrations (decays) per second – the primary measure
  - can be separated by decay type – α, β, γ – in FISPACT-II output
- decay heat, measured in kilowatts (kw)
  - can be separated by decay type - α, β, γ
  - how much heat will be generated in a material even when not exposed to irradiation
  - critical to determine whether cooling is needed to prevent melting
- (contact) γ dose rate, measured in sieverts (Sv) per hour
  - J kg^{-1} deposition rate of radiation energy in biological tissue
  - there are also ingestion and inhalation hazard versions
- clearance index
  - IAEA based measure
  - a nuclide can be disposed of as if it were non radioactive when the index is less than 1

Tungsten irradiation in fusion reactor environment

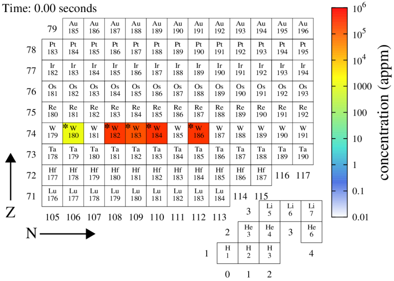


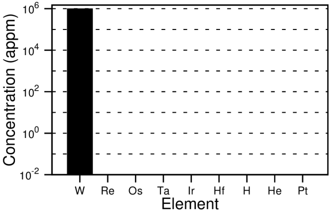


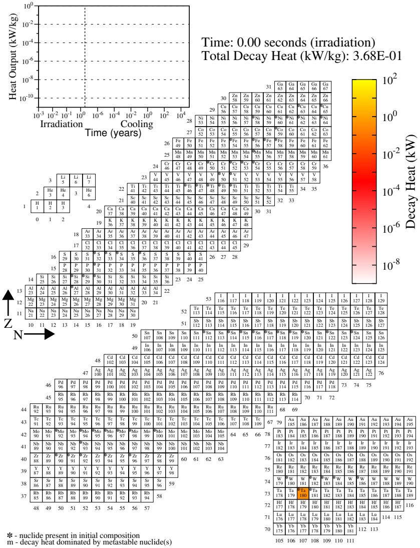
Introduction

Example results

Tungsten irradiation in fusion reactor environment



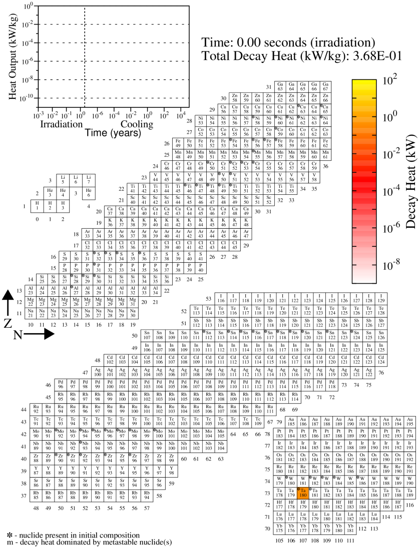




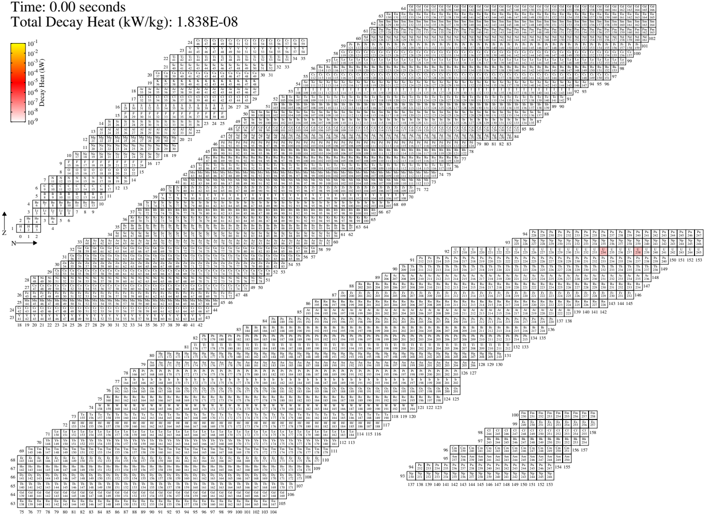
Heat Output (kW/kg)
Irradiation
Cooling
Time (years)
Time: 0.00 seconds (irradiation)
Total Decay Heat (kW/kg): 3.68E-01
Decay Heat (kW)
Z
N
* - nuclide present in initial composition
m - decay heat dominated by metastable nuclide(s)

Introduction

Example results



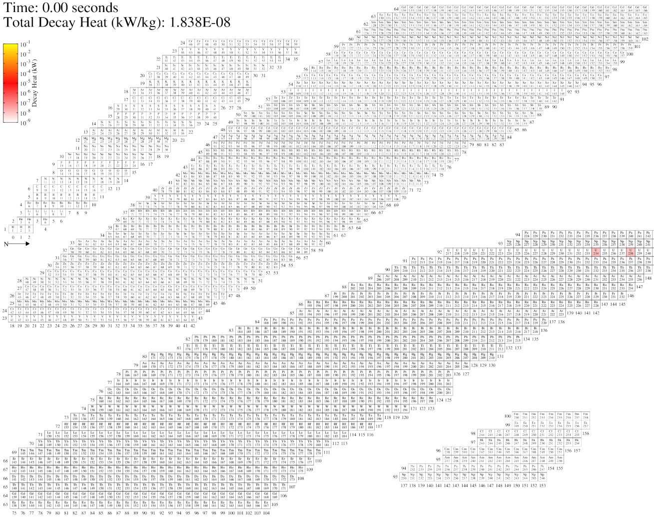
U235 thermal fission pulse decay heat



Applications of Nuclear Data: FISPACT-II
Michael Fleming

Start-up
Introduction
Background
Basic code functionality
16
Code practicalities
Overview
Basic execution
Input files
Nuclear data handling
Keywords
EXERCISES

U235 thermal fission pulse decay heat



Code practicalities

I/O overview

- FISPACT-II is run through terminal commands, using input files and generating output files with simulation results
  - Recommended use of Linux or OS X systems
  - Windows users have options (Cygwin, VirtualBox, etc.) and we have developed a new container solution for next release

Code practicalities
I/O overview

- FISPACT-II is run through terminal commands, using input files and generating output files with simulation results
  - Recommended use of Linux or OS X systems
  - Windows users have options (Cygwin, VirtualBox, etc.) and we have developed a new container solution for next release
- The input files are precisely curated sets of **keywords** with keyword options

- FISPACT-II is run through terminal commands, using input files and generating output files with simulation results
  - Recommended use of Linux or OS X systems
  - Windows users have options (Cygwin, VirtualBox, etc.) and we have developed a new container solution for next release
- The input files are precisely curated sets of **keywords** with keyword options
- Use of different keyword and option combinations may result in various types of simulations

- FISPACT-II may be run in many ways, but the most common execution process takes four stages:

- FISPACT-II may be run in many ways, but the most common execution process takes four stages:
- Handling the nuclear data libraries and incident spectra
  - Condense fission yields and decay data
  - Collapse (fold) cross-sections with incident spectra
  - Print summary of library data (optional, but often useful)

Code practicalities

Code execution overview

- FISPACT-II may be run in many ways, but the most common execution process takes four stages:
- Handling the nuclear data libraries and incident spectra
  - Condense fission yields and decay data
  - Collapse (fold) cross-sections with incident spectra
  - Print summary of library data (optional, but often useful)
- Setting initial conditions (material composition, etc.) & select options (output format, uncertainty quantification)

- FISPACT-II may be run in many ways, but the most common execution process takes four stages:
- Handling the nuclear data libraries and incident spectra
  - Condense fission yields and decay data
  - Collapse (fold) cross-sections with incident spectra
  - Print summary of library data (optional, but often useful)
- Setting initial conditions (material composition, etc.) & select options (output format, uncertainty quantification)
- Simulate irradiation phases, subsidiary calculations and output data

Code practicalities
Code execution overview

- FISPACT-II may be run in many ways, but the most common execution process takes four stages:
- Handling the nuclear data libraries and incident spectra
  - Condense fission yields and decay data
  - Collapse (fold) cross-sections with incident spectra
  - Print summary of library data (optional, but often useful)
- Setting initial conditions (material composition, etc.) & select options (output format, uncertainty quantification)
- Simulate irradiation phases, subsidiary calculations and output data
- Simulate cooling phases and output summary data

- FISPACT-II is run through terminal commands:

```
machine:~ user$ fispact input files
```

Code practicalities

Basic execution

- FISPACT-II is run through terminal commands:

```
machine:~ user$ fispact input files
```

- If you are not familiar with using a terminal, **don't worry**, but please let me know so you can get the most out of this workshop

Code practicalities
Basic execution

- FISPACT-II is run through terminal commands:

```
machine:~ user$ fispact input files
```

- If you are not familiar with using a terminal, **don't worry**, but please let me know so you can get the most out of this workshop
- The code requires two files:
  - a simulation specification file `input.i`,

Code practicalities

Basic execution

- FISPACT-II is run through terminal commands:

```
machine:~ user$ fispact input files
```

- If you are not familiar with using a terminal, **don't worry**, but please let me know so you can get the most out of this workshop
- The code requires two files:
  - a simulation specification file `input.i`,
  - an amusing named `files` file that lists all input/output files

- The files file contains the mapping of all library files that FISPACT-II will use in the calculation. Every required data must be listed – otherwise a fatal error will be issued.

Code practicalities

Anatomy of the `files` file

- The files file contains the mapping of all library files that FISPACT-II will use in the calculation. Every required data must be listed – otherwise a fatal error will be issued.
- These are listed in Tables 1-3 of the User Manual Section 3.1 pages 23-24

- The files file contains the mapping of all library files that FISPACT-II will use in the calculation. Every required data must be listed – otherwise a fatal error will be issued.
- These are listed in Tables 1-3 of the User Manual Section 3.1 pages 23-24
- Comments are given by #, an example:

```
# Cross section data from TENDL-2015
xs_endf /path/to/tendl-2015/neutron/709-data
# My input spectra
fluxes /my/working/directory/my_spectra
# Decay data from ENDF/B-VII.1
dk_endf /path/to/endfb7.1/decay
```

Code practicalities

Anatomy of the `files` file

- The files file contains the mapping of all library files that FISPACT-II will use in the calculation. Every required data must be listed – otherwise a fatal error will be issued.
- These are listed in Tables 1-3 of the User Manual Section 3.1 pages 23-24
- Comments are given by #, an example:

```
# Cross section data from TENDL-2015
xs_endf /path/to/tendl-2015/neutron/709-data
# My input spectra
fluxes /my/working/directory/my_spectra
# Decay data from ENDF/B-VII.1
dk_endf /path/to/endfb7.1/decay
```

- There are up to 20 unique types of data, use the example inputs as guidance and/or the User Manual

$$\frac{dN_i}{dt} = \underbrace{-N_i(\lambda_i + \sigma_i\phi)}_{\text{loss}} + \sum_{j\neq i} \underbrace{N_j(\lambda_{ji} + \sigma_{ji}\phi)}_{\text{creation}}$$

$$\frac{dN_i}{dt} = \underbrace{-\,N_i(\lambda_i \qquad\qquad)}_{\text{loss}} + \sum_{j\neq i} \underbrace{N_j(\lambda_{ji} \qquad\qquad)}_{\text{creation}}$$

- decay constants λ_i, λ_{ji} (s^{-1})
- GETDECAY to read-in from pre-prepared ARRAYX file
  - or to create ARRAYX

$$\frac{dN_i}{dt} = \underbrace{-N_i(\qquad \sigma_i\phi)}_{\text{loss}} + \underbrace{\sum_{j\neq i} N_j(\qquad \sigma_{ji}\phi)}_{\text{creation}}$$

- (neutron) fluxes ϕ and energy dependent spectra in neutrons cm^{-2}s^{-1}
- `GETXS` to collapse (fold) `FLUXES` file with reaction data to produce `COLLAPX` file of σ_i, σ_{ji} values (or read from it)
- `FLUX` to specify total flux ϕ

- Users provide an incident particle spectrum that is used to generate '1-group' or effective cross sections

$$\overline{\sigma}_r = \sum_i \phi_i \sigma_{i,r}$$

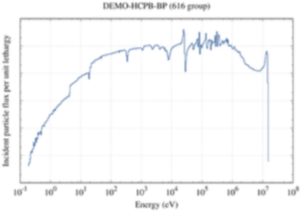
- Users provide an incident particle spectrum that is used to generate ‘1-group’ or effective cross sections

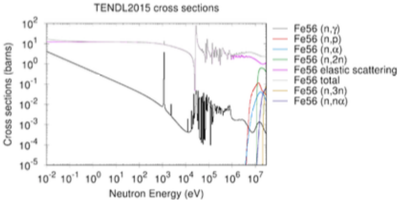
$$\overline{\sigma}_r = \sum_i \phi_i \sigma_{i,r}$$

- Requires `fluxes` input and data files: `ind_nuc` (nuclide indices), `xs_endf` (ENDF reaction data) and `prob_tab` probability tables (for self-shielding)

Code practicalities

Collapse schematic





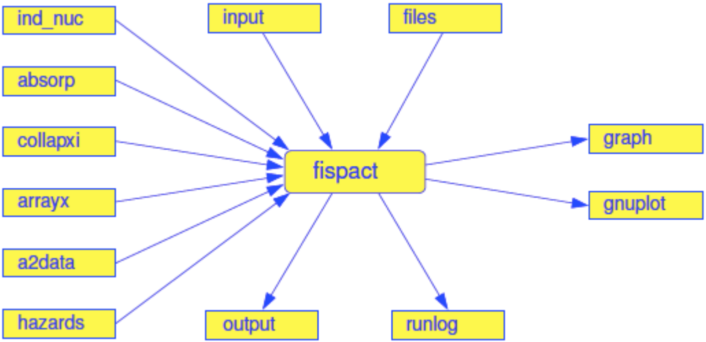
Fe 56	(n,a)	Cr 53	9.30007E-03+-	1.87183E+02	Fe 56	(n,h)	Cr 54	1.43520E-11+-	5.52180E+02
Fe 56	(n,2p)	Cr 55	1.89736E-12+-	0.00000E+00	Fe 56	(n,t)	Mn 54	2.71346E-08+-	4.06438E+02
Fe 56	(n,np)	Mn 55	1.09412E-02+-	1.71659E+01	Fe 56	(n,d)	Mn 55	9.78951E-04+-	2.52961E+02
Fe 56	(n,p)	Mn 56	3.09369E-02+-	6.14934E+00	Fe 56	(n,2n)	Fe 55	1.10279E-01+-	1.43515E+01
Fe 56	(n,E)	Fe 56	3.39432E+00+-	8.08801E-01	Fe 56	(n,n)	Fe 56	3.59211E-01+-	1.13654E+01
Fe 56	(n,g)	Fe 57	1.11989E-02+-	3.18853E+00					

- The ‘condense’ of the decay data is the reading of all ENDF-6 data files into a set of data for inventory simulations (λ for each nuclide with branching ratios and spectral information)

- The 'condense' of the decay data is the reading of all ENDF-6 data files into a set of data for inventory simulations (λ for each nuclide with branching ratios and spectral information)
- The fission yield reading also weights the energy-dependent yields:

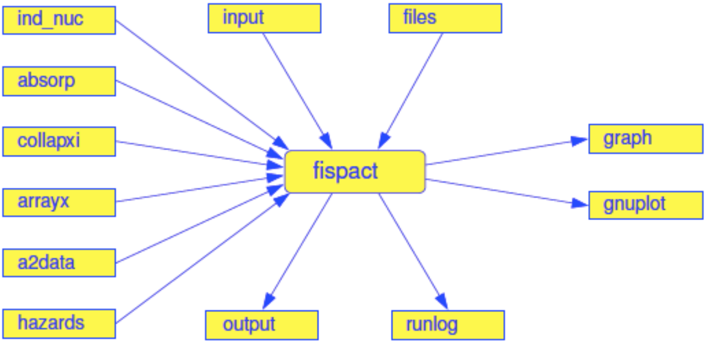
$$Y(Z,A,I) = \frac{\sum_i Y_i(Z,A,I)\,\phi_i\,\sigma_{mt=18,i}}{\sum_i \phi_i\,\sigma_{mt=18,i}}$$

Code practicalities
Inventory schematic



- The inventory simulation requires the transcribed cross section, decay and yield data along with user-supplied irradiation/cooling/response information

Code practicalities
Inventory schematic



- The inventory simulation requires the transcribed cross section, decay and yield data along with user-supplied irradiation/cooling/response information
- Variety of outputs, graph data, tabulated, etc.

Code practicalities

Anatomy of input files

- The FISPACT-II inputs use keywords in specific order - use the examples for direction (<...> are comments):

```
< -- Control phase -- >
GETXS option_1 option_2
GETDECAY option_3
FISPACT
* Description of calculation
< -- Initial phase -- >
< material definition, simulation options, etc. >
...
< -- Inventory phase -- >
< irradiation definitions, cooling, outputs >
...
```

Code practicalities

Keyword options

- The most recent distribution (Dec 2016 v 3-20) that is available through the NEA Data Bank, RSICC, etc., has some 100 keywords - see the manual or wiki:

  `https://fispact.ukaea.uk/wiki/FISPACT-II_keywords`

- The most recent distribution (Dec 2016 v 3-20) that is available through the NEA Data Bank, RSICC, etc., has some 100 keywords - see the manual or wiki:

  `https://fispact.ukaea.uk/wiki/FISPACT-II_keywords`

- Options are also well documented, for example

  `GETXS option_1 option_2:`

- The most recent distribution (Dec 2016 v 3-20) that is available through the NEA Data Bank, RSICC, etc., has some 100 keywords - see the manual or wiki:

  `https://fispact.ukaea.uk/wiki/FISPACT-II_keywords`

- Options are also well documented, for example

  `GETXS option_1 option_2`:

  - `option_1 = -1` read binary group data
  - `option_1 = 0` read binary collapsed data
  - `option_1 = 1` read text group data with `option_2` group

FISPACT-II keywords

A run of FISPACT-II is controlled by a sequence of commands given in a user-supplied input file, the anatomy of which is described in the code execution page. This requires a series of keywords to be specified for control, initial and inventory stages. Below are all of the keywords available in the most recent release of FISPACT-II, sorted by their phase and with a short description. Pages for each of the keywords are linked from this table which provide details of how to use each. Please note that as the code develops new keywords are added to allow access to new features while retaining backwards compatibility. For clarity, keywords added since the 3-00-00 distribution are identified with the scheme below:

Previous release 3-00-00	Current supported release 3-20-00

FISPACT-II Keywords

Keyword	Version	Control	Initial	Invent.	Description
ALLDISPEN	3-20-00	✓			Sets the displacement energies (in eV) for all nuclides
ATDISPEN	3-20-00	✓			Sets the displacement energies (in eV) for specified elements
ATOMS	3-00-00		✓	✓	Sets initial conditions and initiates output with inventories and observables. After the ZERO keyword will also output uncertainties
ATWO	3-00-00		✓		Causes output of legal limits for activity in transport of radioactive material
BREMSSTRAHLUNG	3-00-00		✓		Causes output of brehmsstrahlung contributions for specified nuclides
CLEAR	3-00-00		✓		Causes clearance data of radionuclides to be output
CLOBBER	3-00-00	✓			Allows FISPACT-II to overwrite existing output with same name
CNVTYPE	3-20-00	✓			Allows the user to specify formalism for conversion of incident particle spectra
COVARIANCE	3-00-00	✓			Causes cross-channel covariances to be calculated (if present in nuclear data)
CULTAB	3-00-00		✓		Produces additional lines in tab files for specific post-processing tools
CUMFYLD	3-00-00	✓			Allows the cumulative fission yields to be read, rather than the default independent yields

Code practicalities
Wiki - GETXS

Keyword:GETXS

GETXS *libxs <ebins>*

This keyword has two integer parameters. If the first parameter *libxs* is set to 0, then the second parameter should be omitted, and cross section data are read from the existing collapsed library (`collapx` file) specified in the `files` file. If *libxs* is 1, then the second parameter *ebins* gives the number of energy bins to be used in collapsing the cross section data from the ENDF library files and fluxes or `arb_flux` files specified in the `files` file. If *libxs* is -1, then the ENDF data are read from the compressed binary version of the ENDF data stored in the file specified by `xs_endfb` in the `files` file. Note that the value *libxs* = -1 is not valid for legacy EAF libraries. For information on the preparation of the compressed binary ENDF data files see this page.

The **GETXS** keyword may also be used in the initial conditions and inventory calculation phases to modify the cross sections, for example to handle time-dependent projectile spectra and temperature changes in cross-sections.

The number of energy groups *ebins* must be consistent with the number of groups in the supplied library file. The permitted numbers of groups for cross-section data are, for the most recent public distribution, the CCFE-709 for neutrons and CCFE-162 for charged particles. Legacy EAF libraries and their various group structures may also be used.

Example usage, where the cross sections are re-collapsed every 20 days during some irradiation simulation:

```
< -- Control phase -- >
 ...
 GETXS 1 709
 FISPACT
```

Code practicalities

Most common keywords

Some can be used multiple times, for example re-collapse data in burn- up with spectral shift or change the flux/power normalisation – See the wiki for all details or the User Manual

```
FISPACT              Required to start simulation
GETXS/GETDECAY       Required to process nuclear data
END                  Required to end input
TIME                 Used to specify time intervals
ATOMS/STEP           Used to instruct the code to solve
HAZARD/DOSE          Used to request standard outputs
MASS/FUEL            Used to specify input material
USEFISSION           Used to turn on fission
FLUX/POWER           Used to set flux normalisation
PRINTLIB             Used to print nuclear data
ZERO                 Used to zero the clock for cooling
SSFCHOOSE/           Used to for self-shielding
SSFMASS/SSFFUEL
```

Code practicalities
Getting started

- FISPACT-II comes with a `getting_started` tutorial suite that we will use in this session
- Steps:

- FISPACT-II comes with a `getting_started` tutorial suite that we will use in this session
- Steps:
  - Open a terminal in Linux Mint and navigate to `/getting_started/FNS_Inconel`

- FISPACT-II comes with a `getting_started` tutorial suite that we will use in this session
- Steps:
  - Open a terminal in Linux Mint and navigate to `/getting_started/FNS_Inconel`
  - Open the user manual (online at `http://fispact.ukaea.uk`) to section 3

- FISPACT-II comes with a `getting_started` tutorial suite that we will use in this session
- Steps:
  - Open a terminal in Linux Mint and navigate to `/getting_started/FNS_Inconel`
  - Open the user manual (online at `http://fispact.ukaea.uk`) to section 3
  - Complete the first exercises and ensure that you perform the pulsed decay heat example

- FISPACT-II comes with a `getting_started` tutorial suite that we will use in this session
- Steps:
  - Open a terminal in Linux Mint and navigate to `/getting_started/FNS_Inconel`
  - Open the user manual (online at `http://fispact.ukaea.uk`) to section 3
  - Complete the first exercises and ensure that you perform the pulsed decay heat example
- If you have already attended the NEA training course **let me know** and I'll provide more challenging examples

Lecture break for exercises

UK Atomic Energy Authority

The Abdus Salam International Centre for Theoretical Physics
www.ictp.it

IAEA
International Atomic Energy Agency
Atoms for Peace