Decay heat pre-processing for MELCOR 1.8.6

Petr Vokáč, vok@ujv.cz
ÚJV Řež, a.s.

5th EMUG, 2.-3. 5. 2013, ÚJV Řež, a.s.

Content

- DCH pre-processing
  - Motivation (why a new tool is needed)
  - Solution and implementation
  - Distribution of radionuclides into MELCOR elements
  - Verification
  - Current applications
  - Where to get it
  - How to use it
- Application for activity transport post-processing ?

Motivation

- decay heat calculation according to the ANS standard is applicable only for EOC reactor scenarios
- direct preparation of decay curves input for MELCOR from ORIGEN outputs (as done in past) is cumbersome due to :
  - necessity to produce decay curves for each assembly type by ORIGEN calculation
  - necessity to process large amount of ORIGEN output to set up input data for MELCOR
- number of various assembly types is increasing, e.g.:

  4, 5, 6 year fuel cycle of VVER-440 with different core powers,

  VVANTAGE6 or TVEL fuel for VVER-1000

⇒ **need for a pre-processing tool** to simplify and to standardize input preparation for DCH:

- ORIGEN input limited just to FP activity and mass at the end of irradiation of each assembly
- output formated as MELCOR input for DCH

Solution and implementation

- The `endf.py` code was written in Python 2.7 :
  - Decay data were taken from ENDF/B-VII.1 Radioactive Decay Data File (e.g.: www.nndc.bnl.gov)

    Python interface for the database was written — selected data needed for the algorithm were extracted from ENDF and written to auxiliary file using Python pickle format
  - Decay chains calculation using matrix exponential from SciPy

    Function scipy.linalg.expm uses “Padé” approximation
    (see e.g.: linalg tutorial)

Distribution of radionuclides into MELCOR elements

It is a questionable step of the algorithm — some short lived child isotopes were moved to their parent elements, current selection:

Parent	Short lived child
As	SE77M, SE79M
Br	KR83M
Cd	IN115M, IN117, IN117M, IN118
Ce	PR144, PR144M
Cs	BA136M, BA137M
Dy	HO166
I	XE134M
Kr	RB88, RB90

Mo	TC103
Ni	CU66
Pd	AG109M, AG111M, AG112, RH100
Ru	RH103M, RH105M, RH106, RH106
Sr	Y89M, Y90, Y91M
Ta	W183M
Te	I132
W	RE188
Zr	NB95, NB95M, NB97, NB97M

- it might be subject of change depending of simulated accident
- sensitivity of results to this input was not investigated

Verification

Comparison of total decay heat of single assembly during 10 years decay with ORIGEN-S — results are slightly underestimated with maximum difference about 1% at 10 years

- considered acceptable taking into account other uncertainties
- difference is probably caused by neglecting decay chains after spontaneous fission — results can be improved (but it is not worth the effort)

Various tests with different requested time — very good stability of the algorithm

Current applications

- VVER-440 shutdown scenarios
- VVER-440 spent fuel pool scenarios
- VVER-1000 spent fuel pool scenarios

Where to get it?

Available for download from :
http://www.ujv.cz/web/ujv-200/meltools-download-page, file `endf2012open.tar.gz`

License: "as-is"

How to use it?

- Necessary inputs:
  - For each assembly (from ORIGEN-S output):
    - activity (at the end of irradiation) of radionuclides considered [Bq]
    - mass of elements considered [g]
  - Composition of the core or the spent fuel pool inventory — fraction of each assembly in total inventory
    (including U mass of fresh assemblies with zero activity)
- prepare configuration file — use included sample config file as an example or consult `index.html`
- run the calculation using python script which may look like:

```
import endf
endf.config("config.conf")
```

- Output can be included to MELCOR input using r*i*f

Limitations

- list of elements to calculate specific power is fixed in the `endf.py` code:

  Kr, Xe, Rb, Cs, Sr, Ba, Br, I, Se, Te, Ru, Rh, Pd, Nb, Mo, Tc, Zr, Ce, Np, Pu, Y, La, Pr, Nd, Pm, Sm, Eu, U, Ag, Sn, As, Sb, Cd, Am, Cm

  Power of elements which are not in the list is added to the U specific power

- list of compounds is fixed in the `endf.py` code — currently only CsI

  Data structures for compounds is foreseen in the code. However compounds should be defined directly in the code.

End of DCH pre-processing

Use for activity transport?

Similar algorithm can be used to post-process activity transport (taking into account decay chains) under assumption that the transport follows the first order kinetics :

Decay in single volume

$$\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t} = -\mathbf{\Lambda a}$$

$$\mathbf{a}(0) = \mathbf{a}_0$$

$$\mathbf{a}(t) = e^{-\mathbf{\Lambda} t}\mathbf{a}_0$$

Decay with first order kinetics transport

$$\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t} = -\left(\mathbf{R} + \mathbf{\Lambda}\right)\mathbf{a}$$

$$\mathbf{a}(0) = \mathbf{a}_0$$

$$\mathbf{a}(t) = e^{-(\mathbf{R}+\mathbf{\Lambda})t}\mathbf{a}_0$$

MELCOR calculates:

- release from fuel for groups of FP
- transport of non-condensible gas, vapors and aerosols with bulk media (CVH/FL)
- deposition of aerosols (hygroscopic and non-hygroscopic)

Problem : how to calculate $\mathbf{R}$ from available MELCOR outputs?

Possibilities to evaluate transport constant :

- flow (deposition) rate: $\frac{\mathrm{d}m}{\mathrm{d}t} = -\frac{w}{V}m \Rightarrow r = \frac{w}{V}$

  However flow rate in the output should be an average not instantaneous value.

- integral of flow (deposition) rate : $\frac{\mathrm{d}m}{\mathrm{d}t} = -\frac{\Delta V}{V}m \Rightarrow r = \frac{\Delta V}{V}$

  However the transport direction should not change during the plot file time step.

- mass of transported media:

  Then the matrix of factors should fulfill equation:

$$\mathbf{m}(t + \Delta t) = e^{-\mathbf{R}\Delta t}\mathbf{m}(t)$$

  R can be calculated using "matrix logarithm" (available e.g.: from SciPy scipy.linalg.logm), however it might be numerically unstable.

Is it feasible to add "r" variables for equation $\mathbf{a}(t) = e^{-(\mathbf{R}+\mathbf{\Lambda})t}\mathbf{a}_0$ to MELCOR output?

Thank you for your attention

Any questions? (anwers?)