

SPECTRA-PKA

December 8, 2016

1 Introduction

SPECTRA-PKA [1] is a command-line driven programme for calculating the expected primary knock-on atom (PKA) spectra for a given target nuclide under neutron or charged particle irradiation. NJOY-processed recoil matrices must be provided as the input nuclear data for each nuclide and reaction channel of interest. SPECTRA-PKA will read the nuclear data and collapse the data for each reaction channel in a file with a user-defined energy spectrum of incident particles. SPECTRA-PKA outputs the resulting PKA spectra for each reaction channel read-in, as well as summed PKA distributions for each different recoiling nuclide or element, including both the typically heavy residuals and the secondary-emitted light gas particles. Crucially, SPECTRA-PKA has the ability to process data, in a single run, for a complex material containing many different target species. The user must specify the atomic percentage for each target and the appropriate recoil cross section file, but then SPECTRA-PKA will automatically read-in and process each file and create global average (including as a function of isotope and element) PKA distributions. This feature, in particular, is a significant advancement over what was possible before (such as with SPECTER [1985] from ANL), where typically PKA distributions were only provided for single elements (and not separated by reaction channel) and based on only one nuclear data library (ENDF/B-V in the case of SPECTER). It allows, for example, the user to investigate the variation in PKA distributions as a function of time under irradiation, where a materials composition may change due to transmutation – see, for example, [3].

2 Command line options

SPECTRA-PKA is executed from the command-line. By default, the programme expects to find an input file called `specter.input` in the execution folder, but this behaviour can be over-ridden by specifying the name of an input file on the command-line. The input file provides the user with the option to override the default values of the various code words (see below) that SPECTRA-PKA relies upon to control its execution. The input file should be a text file but is largely free from other formatting requirements. Code word values are specified in any order, one per line, via syntax of type:

```
<code word name>=<code word value>
```

A description of each code word and its expected value(s) are given below. An alternative syntax allowing an easier description of a complex alloy (containing multiple target species) is also described.

3 Code words

Default values are given in brackets after each code word)

- `number_pka_files` [1] – the number of different target species (each with a separate file of recoil cross section input data) to be considered
- `pka_filename` [no default] — filename(s) of recoil cross section data for each target nuclide. In normal syntax the code word should be followed by `number_pka_files` space-separated (and quoted for complex directory specifications) filenames. The list of filenames may be extended across several consecutive lines. In `columns` format (see below), one filename will be specified per line.
- `flux_filename` [no default] — filename containing the incident particle flux spectrum. For compatibility reasons, the format of this file follows closely that used by the SPECTER code. The format is:

```
< text description >
<itype> <dummyi> <igroup> <dummyi> <acnm> <time>
<number_groups> <ksail>
< number_groups + 1 energy bin boundaries in MeV>
< number_groups flux values >
```

where `itype` identifies the type of the flux file. The only type currently supported by SPECTRA-PKA is that where `itype> 1` – so that the third line in the file has the `<number_groups><ksail>` format shown above. `dummyi` are obsolete entries where an integer value is still expected. `igroup` identifies the units of the subsequent flux values, and an appropriate conversion may be applied according to `flux_norm_type` (see below). In the present code only two choices are accepted: n s^{-1} (`igroup=2`), or $\text{n s}^{-1} \text{ MeV}^{-1}$ (any other value of `igroup`), both per unit area. `acnm` and `time` were, respectively, used for flux (power) normalization and to obtain the total fluence, but neither of these influence the output produced by SPECTRA-PKA. A `ksail` value greater than zero indicates that the spectrum file contains error values for the flux in each bin, which are propagated via covariance matrices to the error estimates in the PKA spectrum for each reaction channel (but not to the summed distributions as yet). The flux error vector (`number_groups` values) should immediately follow the `number_groups` flux values.

- `flux_energy_rescale_value` [1.0] — an optional scaling factor to be applied to the boundaries of the input flux spectrum energy bins – e.g. to convert to the required MeV if not already.
- `flux_rescale_value` [1.0] — an optional flux rescale factor that is applied at the time of reading-in the values from the flux file.
- `flux_norm_type` [1.0] — flux normalization control. In the present version of the program, the emphasis is on outputs that are absolute PKAs per second, and so the normalization features are largely obsolete. However, if `flux_norm_type` is 2 then it is assumed that the flux values read-in are per cm^2 and the required conversion to barns (to match the recoil cross sections) is performed. If `flux_norm_type` is 3, then the fluxes used in the subsequent evaluations are exactly those read in from the flux file, with no unit conversion or rescaling. For any other value of `flux_norm_type` the barns conversion is still done, but additionally any unit conversion specified in the flux input file by `igroup` (see above) is also applied.
- `results_filename` [`results.out`] — the string to be used as the name of the file of output the PKA distributions

- `results_stub` [no default] — a filename stub (beginning) string to be used for both the file of output PKA distributions, and the index file listing the type of each distribution and their location (index) within the main file. If `results_stub` is non-empty it overrides any specification within `results_filename`. The resulting output files will be named

`<results_stub>.out` and `<results_stub>.indexes`,

for the PKA distributions and index file, respectively. If `results_stub` is empty the index file will have a default name of `index_summary.dat`

- `pka_ratios` [1.0 for all targets] — the weighting factors to be used when combining (summing) results from different targets (each with a separate PKA cross section file). Normally these would be atomic fractions, for example of the naturally occurring isotopes of W, but any non-negative numbers can be used. The user should specify a space-separated list of `number_pka_files` factors. The numbers given need not sum to one – SPECTRA-PKA will renormalise.
- `pka_filetype` [2] — an integer specifying the file format of the PKA recoil cross section matrices. The default of 2 is the new style output produced by the UKAEA-modified GROUPR routine of NJOY – and this is the format distributed to users with the SPECTRA-PKA code. Any other value of `pka_filetype` tells the program to expect an old-style format of cross sections – the format required by SPECTER.
- `do_mtd_sums` [.false.] — if true (and if `pka_filetype=2`) then produce summed total distributions, for each target species, of α -particle, proton, inelastic, and scattering (elastic+inelastic) recoils, as well as total distributions of heavy recoils where either α -particles or protons are produced. The sum distributions are output after the per-reaction-channel distributions of each target.
- `do_ngamma_estimate` [.false.] — if true then estimate the recoils, for each target species, that would results from (x, γ) capture reactions. Recoil cross section matrices for this channel are not output by NJOY and so an approximation is made, using the raw (x, γ) reaction cross section (that must be given in the cross section file of each target) – see [1] for more details. Note that the methodology requires SPECTRA-PKA to know the masses of both the parent (target) and (x, γ) daughter species via `ngamma_daughter_mass` and `ngamma_parent_mass` – see below. The (x, γ) recoil distribution is output at the end of the processing of the reaction channels for the current target (but before the sum distributions above).
- `ngamma_parent_mass` [55.934936326 – ^{56}Fe mass for all targets] — space-separated list of `number_pka_files` real numbers defining the mass (carbon-12 scale) of each target isotope. Required for calculations of approximate recoil distributions for neutron-capture (x, γ) reactions.
- `ngamma_daughter_mass` [56.935392841 – ^{57}Fe mass for all targets] =55.934936326_ — space-separated list of `number_pka_files` real numbers defining the mass (carbon-12 scale) of the daughter nuclide that would result from an (x, γ) reaction on each target. Required for calculations of approximate recoil distributions for neutron-capture (x, γ) reactions.
- `parent_ele` [Fe] — space-separated list of `number_pka_files` strings defining the element of each target isotope. Used to identify (based on the MT reaction numbers in NJOY) the name and mass number of the daughter for each reaction channel.
- `parent_num`[56] — space-separated list of `number_pka_files` integers defining the mass number of each target isotope. Used to identify (based on the MT reaction numbers in NJOY) the name and mass number of the daughter for each reaction channel.

- `do_global_sums [.false.]` — if true then calculate sum PKA distributions (including as a function of recoiling element and isotope) across all targets using `pka_ratios` weighting factors. A final, total, PKA distribution is also produced. The distributions are appended to the PKA output file after the reaction channels of all targets.
- `do_exclude_light_from_total [.true.]` — if false then include α and proton recoils in final, total recoil spectra. Not that this should be used with care because such recoils are usually high energy and so including them in the total distribution could significantly alter its profile. The light α and proton recoils would typically produce a very different kind of damage to a heavy recoil (close to the target mass) of the same energy, and so they should be treated separately.
- `do_exclude_unknown_from_total [.true.]` — if false then include the PKAs associated with unknown recoil daughters in the total distribution. These “unknowns” typically come from (n, x) -like reactions, and it is up to the user to decide if they are important.
- `max_global_recoils [200]` — when defining global sums, it is not known in advance how many different recoiling elements or isotopes there will be. In the current version of the code, the array that holds this information is dynamically allocated and so the user must guess an expected upper limit on the number of different recoiling species. 200 is sufficient for most applications.
- `energies_once_perfile [.false.]` — if true then for `pka_filetype=2` the energy bin structure will only be included with the first recoil cross section matrix given in each target file. This overcomes some of the redundancy associated with the default file format, where the input and output bin structure was given with each channel, which came from the legacy format associated with SPECTER. For libraries distributed with SPECTRA-PKA, the new style will be the default, and so `energies_once_perfile=.true.` should be specified (the default may change in future releases).
- `incident_particle [n]` — specification of incident particle type. So far, can only be `n` for neutrons or `p` for protons. Doesn’t alter the results for an individual channel, but is required (to be correct) to identify the daughter product from each reaction, and hence to perform the correct summing by isotope/element/reaction.
- `num_columns [1]` — the number of different columns (i.e. code words) to be specified in the `columns` format – see below
- `columns [no default - specified as required]` — allows specification of input values for the different target species in a column format. Immediately after the code word (after the `=`) should be a string of space-separated code words, with a number of entries equal to `num_columns`. The next `number_pka_files` rows of the input file should specify the values of each code word for each target species to be considered in the execution. Allowed code words in the `columns` format are `pka_filename`,
`pka_ratios`,
`parent_ele`,
`parent_num`,
`ngamma_parent_mass`,
`ngamma_daughter_mass`
A typical use of the `columns` format, for the 5 naturally occurring isotopes of Zr, and showing all allowable code words in the format, is given below:

```
num_columns=6
columns= pka_filename pka_ratios parent_ele parent_num ngamma_parent_mass ngamma_daughter_mass
```

```
"Zr090s.asc" 0.51450000 Zr 90 89.904697659 90.905639587
"Zr091s.asc" 0.11220000 Zr 91 90.905639587 91.905034675
"Zr092s.asc" 0.17150000 Zr 92 91.905034675 92.906469947
"Zr094s.asc" 0.17380000 Zr 94 93.906310828 94.908038530
"Zr096s.asc" 0.02800000 Zr 96 95.908271433 96.910951206
```

4 Example calculation

In the folder containing the distributed executables of SPECTRA-PKA there is a “test” folder to get the user started. The folder contains the necessary input files to evaluate the PKA distributions of pure zirconium (Zr) in a typical PWR fission spectrum. To run the test navigate to the test folder and then, on the command line, type:

```
<location of SPECTRA-PKA>SPECTRA_PKA ZR.in
```

Here <location of SPECTRA-PKA> is the folder containing the SPECTRA-PKA executable. The ZR.in is:

```
flux_filename="fluxes_specter.dat"
results_stub="ZR"
num_columns=6
columns= pka_filename pka_ratios parent_ele parent_num ngamma_parent_mass ngamma_daughter_mass
"<PKA data folder>Zr090s.asc" 0.51450000 Zr 90 89.904697659 90.905639587
"<PKA data folder>Zr091s.asc" 0.11220000 Zr 91 90.905639587 91.905034675
"<PKA data folder>Zr092s.asc" 0.17150000 Zr 92 91.905034675 92.906469947
"<PKA data folder>Zr094s.asc" 0.17380000 Zr 94 93.906310828 94.908038530
"<PKA data folder>Zr096s.asc" 0.02800000 Zr 96 95.908271433 96.910951206
flux_norm_type=2
pka_filetype=2
do_mtd_sums=.true.
do_ngamma_estimate=.t.
do_global_sums=.t.
do_exclude_light_from_total=.t.
number_pka_files=5
flux_rescale_value=3.25e14
max_global_recoils=400
energies_once_perfile=.t.
```

Note that in the test folder <PKA data folder> in the above has been replaced by the appropriate directory path for the default hierarchy of the FISPACT-II system. The user should tailor this file as necessary to their own system configuration before running the test.

The test folder also has an “example_results” folder containing the .out and .indexes files that should be produced from a successful execution of the test. The “plot_example” folder provides example .plt files for GNU PLOT that will produce plots of the summed elemental and isotopic PKA distributions. Users familiar with GNU PLOT will note the use of the index option in each plot command. The index numbers currently used in the .plt files are appropriate for the example results (see example_results/ZR.indexes).

Important: Using a different version of the PKA source libraries could result in a different ordering of the results, and so the user should compare the .plt files with the .indexes file from their execution of the test to check that the correct distributions are plotted. The images that should result from this test are given in figure 1.

References

- [1] M. R. Gilbert, J. Marian, and J. -Ch. Sublet, “Energy spectra of primary knock-on atoms under neutron irradiation,” *J. Nucl. Mater.*, vol. 467, pp. 121–134, 2015. <http://dx.doi.org/10.1016/j.jnucmat.2015.09.023>.

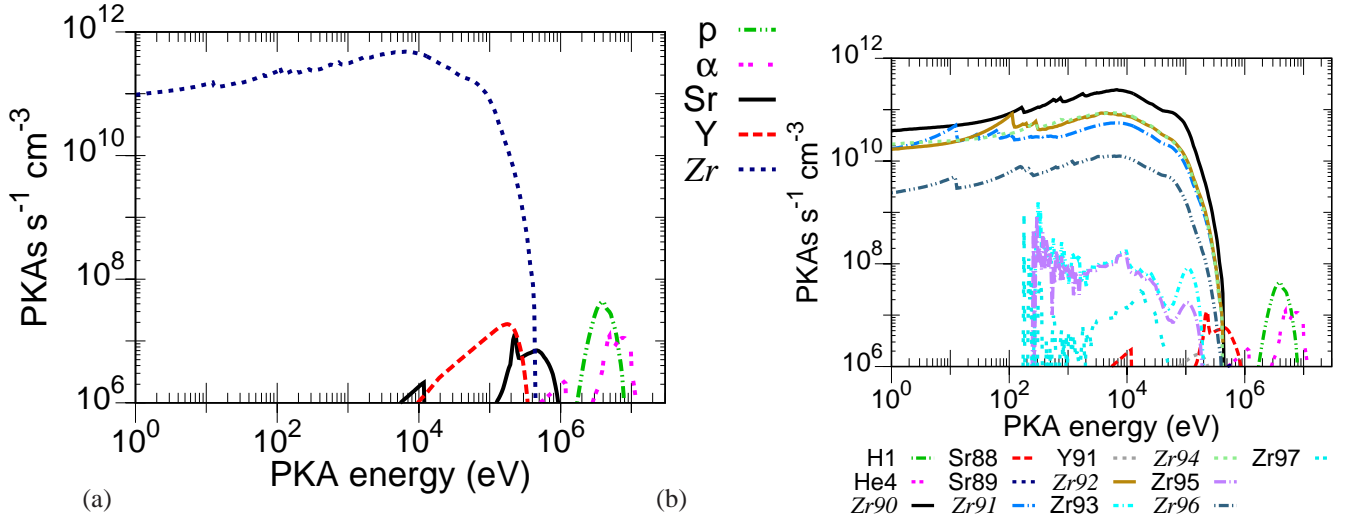


Figure 1: (a) elemental and (b) isotopic PKA distributions for pure Zr under PWR conditions.

- [2] L. R. Greenwood, and R. K. Smither, “Specter: Neutron damage calculations for materials irradiations”, January 1985. Document Number: ANL/FPP/TM-197. SPECTER and SPECOMP are available from the NEA databank. <http://www.oecd-nea.org/tools/abstract/detail/psr-0263>.
- [3] M. R. Gilbert and J.-Ch. Sublet, “PKA distributions: Contributions from transmutation products and from radioactive decay,” *Nucl. Mater. Energy*, 2016. in press. <http://dx.doi.org/10.1016/j.nme.2016.02.006>.