SPECTRA-PKA

August 26, 2020

1 Introduction

SPECTRA-PKA [1, 2, 3, 4] 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 [5] 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, [2]. More recent additions to the code include per-channel evaluations of damage rates (measured as the so-called NRT-dpa metric – see [3] for details), the ability to import a user-defined energy grid for the output of distributions (e.g. to overload the logarithmically-distributed scale commonly used for nuclear physics applications with a more modelling-friendly linear scale), and the application of PKA distributions to "real" atomistic systems to define when and where PKAs will occur in a given lattice of atoms, which can be subsequently applied to modelling activities. The latter has been extended further with a prototyped interface to a code (SDTrimSP [6]) that applies the binary collision approximation (BCA) to simulate the cascade of recoil events associated with a given PKA.

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) is given below. An alternative syntax allowing an easier description of a complex alloy (containing multiple target species) is also described. Unless otherwise stated, all numbers given in the main input file (or other files for the irradiation spectrum, etc.) should use a dot (.) to represent the decimal place (including in numbers represented in scientific notation) – use of a comma (,) will result in incorrect reading of files and potentially misleading results.

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 ascending 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. In the present code only two choices are accepted: n s^{-1} (`igroup=2`), or n s^{-1} MeV^{-1} (any other value of `igroup`), both per unit area (typically cm^2). Further (and potentially overriding) normalization can be applied via `flux_norm_type` (see below) to convert the read flux values into the necessary per barns (10^{-24} cm^2) units. An appropriate flux conversion may be applied according to `flux_rescale_value` (see below), for example to scale a normalized spectrum to the actual total flux. `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. This, for example, could be needed to scale a normalized spectrum to the correct total flux of the irradiation.

- flux_norm_type [2] — 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 in units of n s^{-1} 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 – typically this would be apply if the provided flux values are already in n s^{-1} $barns^{-1}$. 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. The index file of channel numbers will be called index_summary.dat and the log file will be <results_filename>.log. It is recommended that the user always use results_stub (below) to define an explicit naming convention for all files and avoid issues of accidental file overwrites.

- 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, <results_stub>.indexes and <results_stub>.log ,

  for the output file PKA distributions, the reaction channel index file and run log-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 an 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 an 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).

- do_tdam [.false.] — if true then, for each reaction channel, calculate the equivalent damage-energy bin boundaries associated with each PKA energy bin following the Lindhard-Scharff-Schiøtt [7] equations described in [3, 8]. Subsequently, these are used to calculate displacement-energy [9] and displacements-per-atom (dpa) rates for each bin and the total rates for the reaction channel (see [3]

for more details). SPECTRA-PKA will also propagate the displacement-energy and dpa rates onto the global sums if do_globals is also true.

- assumed_ed [40.0] — threshold displacement energy (in eV) for the material used for displacements-per-atom (dpa) evaluation via the Norgett-Robinson-Torrens [10] formula. A single value to be used for the entire execution. Even in a complex alloy, one atomic species will typically dominate, and so the conversion of damage energy to dpa should use a single value, which could be an average over the major elemental constituents. For example, the typical value for a material dominated by elements such as Fe and its neighbours in the periodic table is 40 eV, and is the default in SPECTRA-PKA (and in many other codes).

- 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 (for the result to be correct) to identify the daughter product from each reaction, and hence to perform the correct summing by isotope/element/reaction (if do_globals is true).

- 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
```

- do_user_output_energy_grid [.false.] — if true then this variable causes the program to read a user-defined PKA energy grid (in MeV units) for a specified (via user_energybin_file) file. This grid will be used instead of those read from the input data streams to output the various summed PKA spectra (see do_mtd_sums and do_global_sums). This option can be particularly useful to recast the spectra onto a linear energy grid (instead of the logarithmic grid typically used in nuclear physics to represent nuclear cross sections and/or flux spectra) for use in integrated modelling schemes.

- user_energybin_file [ebins.dat] — the name of the file from where the user-defined output energy grid will be read. The first line of this file is expected to give the number of energy bounds, which are subsequently listed in the file (in increasing MeV units). It is assumed the minimum energy (below the first bound) is zero, but the code does not check if the maximum energy

is high enough to capture all of the evaluated PKA energies (PKA fluxes above the maximum will be omitted from the output). If this file cannot be found, or if an error occurs during reading, then the program will default to the original grids from the input data.

- `user_grid_option` [2] — the use of a user-defined output energy grid can result in very large output files, particularly if using a fine resolution linear grid over several orders of magnitude (in energy). In such cases it may be desirable to restrict the application of the user grid. If `user_grid_option` is 1 (or any other integer not discussed here) then the user grid will be used to output all summed spectra (sums across nuclides, elements, reaction sets, etc.). If 2 then only the global sums at the end of the evaluation will use the grid – sums performed at the end of the processing of each target nuclide will use the default energy grid obtained from the input data steams, which are typically based around log-scaling. If `user_grid_option` is 3 then only the very final "total" PKA spectrum will use the specified grid.

- `do_outputs` [`.true.`] — switch off to reduce program execution output (particularly to the log-file).

- `do_timed_configs` [`.false.`] — if true then define PKAs at the atomic level and as a function of time according to the evaluated distributions. Currently, `SPECTRA-PKA` defines an ideal lattice of atoms according to various input parameters (below), but future developments will allow pre-defined systems (e.g. containing defects and/or precipitates) to be read-in. The code applies stochastic sampling to define which atoms will become a PKA and when. A pseudo-random-number generator is used to give the PKA an energy and initial direction vector. Each PKA event is recorded in the file `config_events.pka` (one event per line with position, velocity vector, PKA species [which may be different from the original target], mass, and energy). A further output file `config_analysis.pka` records various global analyses of the population of PKA events.

- `config_max_pka_vectors` [200] — in a very complex system (many different target species) the number of possible PKA reaction channels will be large. Applying the stochastic sampling to a very large number of minor channels (i.e. ones that are unlikely to generate a PKA on the simulation times considered) could be time-consuming an unnecessary. This parameter allows the user to restrict the analysis (for atomic PKA events only) to the `config_max_pka_vectors` most dominant PKA channels – the filtering is performed according to the total PKA-rate above `assumed_ed` in each channel.

- `config_do_exclude_light_pkas` [`.true.`] — if true (the default) do not allow PKA channels generating gas-particle PKAs to be considered in the atomistic analysis. This is typically the correct approach (i.e. for transition metal elements) since gas (helium/hydrogen) PKAs will behave very differently for PKAs of elements similar in mass to the host lattice. However, if `SPECTRA-PKA` is being interfaced with a BCA code, it might be appropriate to consider these PKAs since they can be considered within the binary-collision approximation.

- `timestep` [`1e-12` (1 ps)] — Δt time-step (in seconds) to use when defining atomic PKAs. `SPECTRA-PKA` will define the correct number of PKAs for each time-step according to the stochastic sampling. Typically, `timestep` should be chosen to be large enough that at least one PKA event happens per Δt, but not so large that independence between individual events in a time-step cannot be guaranteed.

- `nsteps` [100] — number of Δt time-steps to define PKAs for. `nsteps` $\times$ `timestep` will define the total simulation time considered and would normally be constrained by the rate of secondary evolution factors such as transmutation-induced composition changes or defect-evolution timescales.

- box_type [1] — for the generation of perfect crystalline lattices the user can select body-centered-cubic (bcc) [1] , face-centered-cubic (fcc) [2] , or hexagonal close-packed (hcp) [3] .
- box_units [10] — number of fundamental unit cells in along each axis of the lattice. For bcc/fcc this will result in a perfectly cubic box, while for hcpb the box will be elongated along c.
- latt [2.8664] — lattice parameter (in Angstroms) to use when defining atomic positions (the default is typical of iron/steel).
- do_output_configs [.false.] — do not use – at prototype stage
- do_bca [.false.] — if true then interface with a BCA code (currently only SDTrimSP is allowed) and generate a recoil-displacement cascade for each defined PKA (above assumed_ed). SPECTRA-PKA writes the necessary (SDTrimSP) input files, calls the code, and then reads and processes the result. Each cascade is placed back onto the defined lattice system according to the random direction vector (and shifted through PBCs if requested and necessary). SPECTRA-PKA stores all cascades in one output file (config_bca.dat) that can be subsequently applied to other modelling (or simply plotted).
- bca_type [1] — type of BCA simulation to perform. Currently only SDTrimSP is available, so this parameter has no effect.
- sdtrim_path [SDTrimSP] — path to SDTrimSP executable.
- do_store_bca_output [.true.] — if true then store each set of output files from the bca execution. Files are number sequentially. If false, then only the processed set of recoils within each cascade is stored (in config_bca.dat).
- bca_cell_size [2.8664] — defines grid resolution (in Angstroms) when deciding if cascades overlap – at prototype stage, use with caution
- overlap_stop [.false.] — if true then the generation of PKAs and BCA simulations will stop as soon as the (prototype) overlap criteria is met – at prototype stage, use with caution
- do_bca_pbc [.true.] — if true then make sure all cascades fall within the defined atomic box by assuming periodic boundary conditions (PBCs) and shifting recoil events by one (or more) box lengths in appropriate directions as required.

4 Input PKA data download

SPECTRA-PKA uses NJOY-generated recoil cross section matrices. The output files required from NJOY are non-standard and are produced by a slightly modified version of the GROUPR module within NJOY. An evolving selection of databases produced using various international reaction cross section libraries (for various incident particle types) have been pre-calculated and are available to download as compressed tar archives from https://www.ccfe.ac.uk/FISPACT-II/nuclear_data/PKA/. These can be used immediately with SPECTRA-PKA, or can be used as a template to guide users who may want to create their own input data files (for example, for libraries not currently included on the download page).

5 Example calculation

In the subfolder containing the pdf manual distributed with SPECTRA-PKA there are some examples calculations to get the user started. The first, in manual/test, 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 in the git repository 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.
do_tdam=.t.
assumed_ed=40.0
```

Note that in the test folder <PKA data folder> in the above has been replaced by the appropriate directory path for the default path to PKA data files in the hierarchy of FISPACT-II. 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 GNUPLOT that will produce plots of the summed elemental and isotopic PKA distributions. Users familiar with GNUPLOT 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.

A further example, which doesn't include a plot example (as described above), can be found in the folder manual/usergrid_test. This "test" calculates the PKA distributions for a single Fe isotope (^{56}Fe) irradiated in a typical fission irradiation spectrum. It exemplifies the use of a user-defined energy output grid for the PKA distributions and additionally provides examples of the format of the required user-grid input file.

A final example demonstrates the possibilities of the atomistic PKA capabilities newly developed in the code, including the ability to interface with BCA simulations. In the folder manual/bca_test is another Zr example. In this case the code is asked to stochastically define the PKA events that would take place in a hcp box of 500x500x500 unit cells during a 100 s irradiation under the PWR fission spectrum. Additionally, the BCA code SDTrimSP is run for each PKA above the 40 eV threshold for

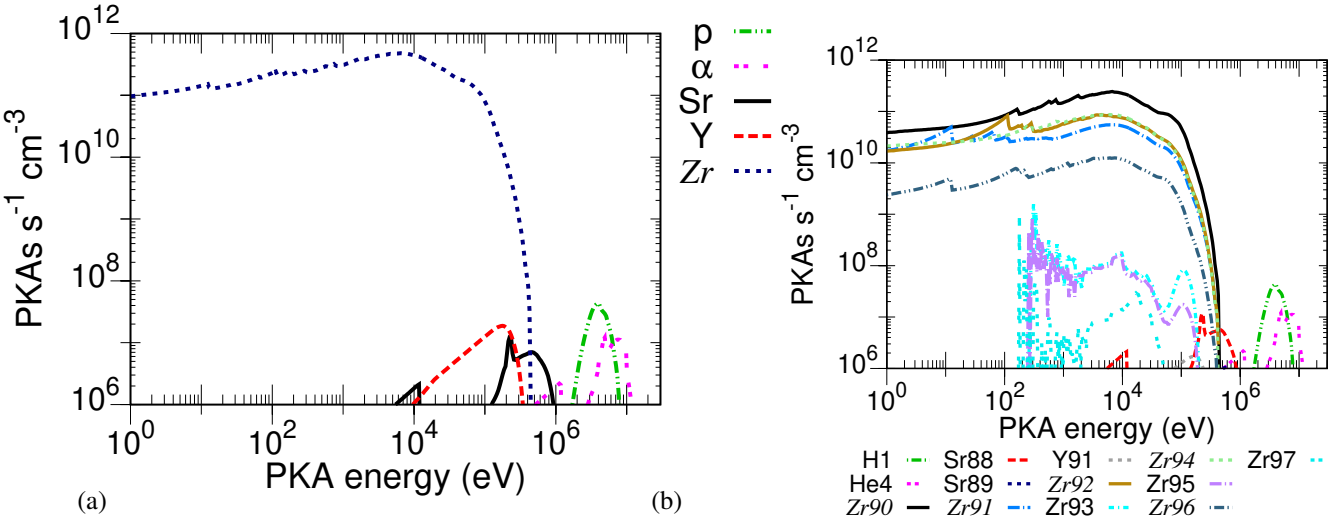

Figure 1: (a) elemental and (b) isotopic PKA distributions for pure Zr under PWR conditions.

atomic displacement in Zr. Two example plot scripts are provided to plot the resulting final accumulation of PKA events and the corresponding cascade distributions. Figure 2 demonstrates the output produced from this simulation.

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.

[2] M. R. Gilbert and J.-Ch. Sublet, "PKA distributions: Contributions from transmutation products and from radioactive decay," *Nucl. Mater. Energy*, vol. 9, pp. 576–580, 2016. http://dx.doi.org/10.1016/j.nme.2016.02.006.

[3] M.R. Gilbert and J.-Ch. Sublet, "Differential dpa calculations with SPECTRA-PKA," *J. Nucl. Mater.*, vol. 504, pp. 101–108, 2018. https://doi.org/10.1016/j.jnucmat.2018.03.032.

[4] M. R. Gilbert, "SPECTRA-PKA", (2018) SPECTRA-PKA is available to download from github at https://github.com/fispact/SPECTRA-PKA.

[5] 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.

[6] A. Mutzke, R. Schneider, W. Eckstein and R. Dohmen, "SDTrimSP", see http://www2.ipp.mpg.de/~stel/SDTrimSP.html.

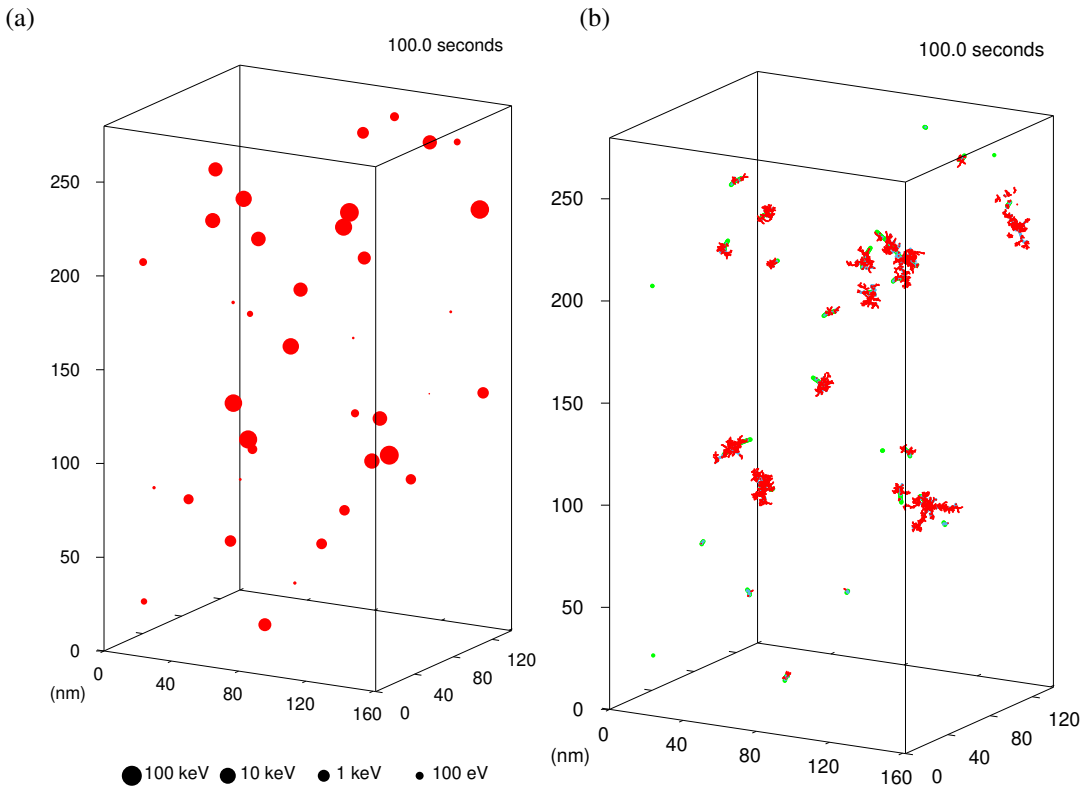

Figure 2: (a) PKA events and (b) BCA-simulated cascades during a 100 s irradiation of a 500x500x500 unit-cell hcp lattice of pure Zr under PWR conditions.

[7] J. Lindhard, M. Scharff, and H. E. Schiøtt, "Range concepts and heavy ion ranges (notes on atomic collisions, II)," *Mat. Fys. Medd. Dan. Vid. Selsk*, vol. 33 (14), pp. 1–42, 1963. https://www.osti.gov/scitech/biblio/4153115

[8] M. T. Robinson, "Basic physics of radiation damage production," *J. Nucl. Mater.*, vol. 216, pp. 1–28, 1994. https://doi.org/10.1016/0022-3115(94)90003-5

[9] A. Adrych-Brunning, M. R. Gilbert, J. -Ch. Sublet, A. Harte, and C. P. Race, "Modelling the interaction of primary irradiation damage and precipitates: Implications for experimental irradiation of zirconium alloys," *J. Nucl. Mater*, vol. 498, pp. 282–289, 2018. https://doi.org/10.1016/j.jnucmat.2017.10.022.

[10] M. J. Norgett, M. T. Robinson, and I. M. Torrens, "A proposed method of calculating displacement dose rates," *Nucl. Eng. Des.*, vol. 33, pp. 50–54, 1975. https://doi.org/10.1016/0029-5493(75)90035-7.