

# FIER: Software for Analytical Modeling of Delayed Gamma-Ray Spectra

## Installation Guide and Reference Manual

M. Shinner, E.F. Matthews, B.L. Goldblum  
University of California, Berkeley

January 24, 2019

### 1 General Description and Physics

The FIER (Fission Induced Electromagnetic Response) code is a C++ program designed to predict delayed  $\gamma$ -ray spectra following fission. Given a fissioning isotope and irradiation scheme, FIER will determine the decay chains for each fission fragment and their daughters. Each product's time-dependent population is calculated using an analytical solution to the Bateman equations that guide decay transmutation [1]. These populations are then used to calculate the number of  $\gamma$ -ray emissions for a given counting scheme [2]. A Monte Carlo method is also available to determine the uncertainty of the fission product populations and delayed  $\gamma$ -ray intensities based on the uncertainties on the input nuclear data.

### 2 Files

The FIER program is contained in the *main.cpp* file with C++ classes separated into header and implementation files. The folder *input\_data* contains three required nuclear data libraries (distilled from ENDF File-8) [4]. The folder *yields* holds NNDC SIGMA independent fission yield data [5]. The input libraries and fission yield data are described further in Appendix A.

If running FIER using `make` (highly recommended), the makefile is located in the same (parent) directory as *main.cpp*. The default input deck is named *deck.txt*. Details on input deck construction are provided in Appendix B. The output of FIER will be placed in a folder labeled *output* if the default deck configuration is used (see Appendix C). **If the output folder is missing or the custom output location doesn't yet exist, FIER will not write the output to disk. Users must create any custom directories.** The folder *testing* contains files used in the included integrity test. Only modify these if you wish to create your own tests.

### 3 Prerequisites

The following are required to run FIER:

- C++11 or higher, and
- the g++ compiler.

In addition, these tools are highly recommended:

- `make`, to run the make file,
- `git`, to pull code from the repository,
- `doxygen` and `graphviz`, to compile documentation, and
- a Bourne-compatible shell, to run the multiprocess Monte Carlo analysis.

## 4 Setup and Execution

Once you have cloned the FIER repository, the makefile can be used to build, run, and test FIER on your system. To run FIER:

1. Edit the input deck (*deck.txt*) with your desired input parameters. See Appendix B for details.
2. Run `$ make` to build and execute FIER using the parameters outlined in *deck.txt*. If all goes well, you should see printed confirmation at each program step (i.e., import, build, calculate, and write).
3. The four output files (see Appendix C) will be found in */output*, unless the names and file paths were changed in the input deck.
4. If `make` fails for any reason, you can manually build and run FIER with the commands:
  - `$ g++ -o fier.exe main.cpp species_data.o helper_functions.o chains_data.o product_data.o monte_carlo.o`
  - `$ ./fier.exe deck.txt`

### 4.1 Documentation

FIER uses Doxygen to document the code. Users can compile this and read it with any html reader. To generate this, use `make` from within the main directory. This requires both Doxygen and the Graphviz package to be installed.

- Mac/Linux users can type `$ make docs`
- Windows users can type `$ make docsw`

After generation has finished, *index.html* will take you to the front page. The folder *html* contains all the documentation source pages.

## 5 Advanced Modes of Operation

### 5.1 Library Corrections

Accuracy of the FIER output is limited by the accuracy of the input data. There are a few known gaps in the ENDF data library. Although these are mostly negligible in impact, FIER comes with the capacity to address some of the more obvious omissions and discrepancies. This optional feature can be activated in the input deck by editing the second line to read `ON DECAY PREDICTION`. To turn it off, replace `ON` with `OFF`. This will perform several corrections to the included ENDF data, detailed below.

### 5.1.1 Decay Prediction

Due to experimental limitations, some isotopes far from the valley of stability have not had their  $\beta$ -delayed particle emission branching ratios measured. Although these species are expected to be 100%  $\beta$  emitters (with an unknown  $n/p$  emission branch), ENDF does not record a decay mode in the `decays.csv` file. This will result in FIER treating these isotopes as stable in default mode. With the Decay Prediction Mode, FIER can check for isotopes with non-zero half-lives (unstable) and fill in the missing decay modes as a  $\beta^-$ - or  $\beta^+$  reaction depending on whether the nuclei are neutron-rich or proton-rich, respectively.

### 5.1.2 Decay Mode Degeneracy

Rarely, an ENDF isotope will have duplicate decay modes listed. FIER detects this by checking to see if an isotope's listed decay modes share daughter isotopes. If they do, the Decay Prediction Mode removes all duplicate modes to prevent double counting of daughter populations. The total branching to the one daughter is conserved by adding the branching ratios of the degenerate decay modes together.

### 5.1.3 Normalized Branching Ratios

Although physically isotope decay branching ratios should sum to 100%, there are a few cases in ENDF where this is not the case. As a result, FIER will normalize the sum of all listed branching ratios for an isotope to 100%.

## 5.2 Monte Carlo Error Analysis

FIER includes a built-in Monte Carlo error analysis tool. With it, users can determine the uncertainty of decay product populations and delayed  $\gamma$ -ray emission uncertainties on a species-by-species basis. In FIER, this is achieved by resampling the input nuclear data parameters assuming a Gaussian distribution with a width determined by the uncertainties specified in the input libraries. Each distribution sample is then used to create a new set of nuclear data inputs and run through the FIER process. Each of these sample trials are aggregated and the standard deviation in the outputs are given as the uncertainties in the centroid calculation (see Appendix C).

### 5.2.1 Bourne-Compatible Shells

**This will only work on setups that can read `.sh` files.** If that isn't possible, use the method described in Sec. 5.2.2.

To decrease wall clock time, FIER was adapted to run in Monte Carlo mode on multiple processors simultaneously. To run, type:

```
$ make montecarlo
```

The program will then list the number of cores on your system. Input how many you wish to use and how many trials you wish to run, and then sit back and wait. A large number of trials will be very memory intensive, so if you see the program hang or crash, try lowering the number of trials. You do not need to edit the input deck for this feature.

While FIER runs in this mode, you should see an update every 100 trials. Each core will print data, which may appear intermixed and out of order; this is normal behavior. You will know that it finished successfully if FIER prints `joining thread:X` for each core you chose to use. The output with uncertainty will be stored in `threads/stdevs/` in an appropriately labeled `.csv` file. If you wish to

see the output of the calculations of any single processor, each of these are stored in *threads/gammas/* or *threads/populations/*.

### 5.2.2 Instructions for .sh Incompatible Systems

For users on other platforms, replace SINGLE with MONTECARLO:**X** in the first line of the input deck (Appendix B), where **X** is the number of trials you wish to run. This will produce the Monte Carlo derived uncertainties directly in the output file indicated by the input deck. This is much slower than the previous method, but will work on Windows systems.

## 6 Testing

The included unit test checks a sample deck modeling the decay of  $^{235}\text{U}$  irradiated with fission-spectrum neutrons against pre-made output. The test also checks that no negative populations are produced and that the population tracking between time steps is properly implemented. Provided the contents of the */testing* folder have not been modified, the test can be run with:

```
$ make test
```

This will build and run FIER on */testing/testdeck.txt*, */testing/testdeck2.txt*, */testing/testdeck3.txt*, and */testing/testdeck4.txt*. This produces output at */testing/output* and then checks for differences with reference output located at */testing/reference*. If there are any differences or an output file fails to be created, the testing program will print an error message to the terminal and the line. If no discrepancies are detected, FIER was built correctly.

## 7 Notes on Floating Point Precision

Due to limitations on the way computers represent floating point numbers, there is a limit on the number of decimal places that FIER can work with. For the most part, this limit is never reached. However during the calculation of populations using the continuous production equation, the numerator terms containing  $1 - e^{-\lambda t}$  may be calculated with significant numerical error when the argument,  $-\lambda t$ , becomes small enough that  $e^{-\lambda t}$  approaches 1. As a result, these terms in the summation have large numerical error and the result of the summation becomes inaccurate. This can lead to negative populations being calculated.

As a remedy to this, FIER checks when the value of  $e^{-\lambda t}$  in the continuous production calculations approaches the value of 1 within *single* precision error (while the calculations FIER performs are done in *double* precision). If it does, the nuclear species in question is treated as stable and all subsequent species in the decay chain are given 0 population. Please note that this approach has limitations. With very short irradiation periods, the populations of short-lived species are calculated with large numerical error.

## References

- [1] H. Bateman, Solution of a system of differential equations occurring in the theory of radio-active transformations, Proc. Cambridge Philos. Soc. 15 (1910) 423-427.
- [2] E.F. Matthews, B.L. Goldblum, et al., FIER: Software for analytical modeling of delayed gamma-ray spectra, Nucl. Instrum. Meth. A 891 (2018) 111-117.
- [3] Doxygen, <http://www.stack.nl/~dimitri/doxygen/>
- [4] Evaluated Nuclear Data File (ENDF) <https://www.nndc.bnl.gov/exfor/endl00.jsp>
- [5] NNDC SIGMA Library <https://www.nndc.bnl.gov/sigma/index.jsp>
- [6] ENDF User Manual (See File 8 for Relevant info) <https://www.nndc.bnl.gov/csewg/docs/endl-manual.pdf>

## A Libraries

In addition to the input deck (see Appendix B), FIER runs using three required ENDF nuclear data libraries located in */input\_data*. A yields file is also required that provides information on the independent fission yields of the target isotope.

### A.1 Input Nuclear Data

The nuclear data files (*decays.csv*, *isotopes.csv*, and *gammas.csv*) were parsed from ENDF, File 8 [4]. Each isotope in the ENDF record was parsed to obtain information on the decay modes and branching ratios,  $\gamma$ -ray energies and intensities, and half-lives and decay constants. For users that wish to compile these data themselves, the parsing format can be found on p. 166 of the ENDF Manual [6]. If for some reason these files are deleted or missing, they must be re-downloaded from the repository or FIER will fail. If these files are renamed or moved, their new position must be recorded in the input deck or FIER will fail.

**For the ENDF libraries, energy units are in keV and time units are in seconds.**

The file *decays.csv* records the decay modes and daughters of all possible isotopes. This is recorded in the comma separated format:

**Zp, Ap, Ip, T, dT, BR, dBR, Zd, Ad, Id**

**Zp, Ap, Ip** is the parent isotope. **Zd, Ad, Id** is the daughter isotope (I is the isomer number, 0 for ground, 1+ for higher). **T, dT** is the half-life and uncertainty. **BR, dBR** is the branching ratio and uncertainty.

The file *isotopes.csv* records the half-lives and energies of each isotope, in the format:

**Zp, Ap, Ip, Ep, dEp, T, dT**

**Ep, dEp** is the parent excitation energy and uncertainty. **T, dT** is the half-life and uncertainty.

The file *gammas.csv* records the gamma radiation energies for each decay, in the format:

**Zp, Ap, Ip, Eg, dEg, Ig, dIg**

**Eg, dEg** are the gamma energies and uncertainties. **Ig, dIg** are the gamma intensities and uncertainties.

### A.2 Fission Yields

There are two modes for sourcing independent fission yield data in FIER.

#### A.2.1 Fission Yields from Library Data

The default mode uses a yields file obtained from the NNDC SIGMA library [5] and contained in the */yields/* directory. Toggled by writing YIELDS:ER in the input deck, this mode searches the following line of the input deck for the elemental symbol, atomic number, and energy group, and then finds the file in the library matching that request. For example, if a user wanted to use the independent fission yields of  $^{235}\text{U}$  produced from fission spectrum irradiation, they would write “**U,235,Fission**” in the relevant line of the input deck (no spaces!). This is detailed further in Appendix B.

There are 5 irradiation energy groups included in the library. The energy group keywords for use in the input deck are provided in Table 1. The available fissionable isotopes and irradiation energy groups are provided in Table 2. Note: Independent fission yields for both the Am-242 ground state and Am-242m isomer are included in the library. To reference the isomer, use (**Am,242m,Thermal**).

Keyword	Description
SF	Spontaneous fission (no input particle)
Thermal	Fission induced by a neutron spectrum with a room temperature Maxwell-Boltzmann distribution
Fission	Fission induced by neutrons from other fission events
DD	Fission induced by 2.45 MeV DD neutrons
DT	Fission induced by 14.1 MeV DT neutrons

Table 1: Independent fission yield energy groups and their input deck keywords.

Element	Number	Energy Group
Th	227	Thermal
Th	229	Thermal
Th	232	Fission
Pa	231	Fission
U	232	Thermal, DT
U	233	Fission, Thermal
U	234	DT, Fission
U	235	DT, Fission, Thermal
U	236	DT, Fission
U	238	DT, Fission, SF
Np	237	DT, Fission, Thermal
Np	238	Fission
Pu	238	Fission
Pu	239	DD, DT, Fission, Thermal
Pu	240	DT, Fission, Thermal
Pu	241	Fission, Thermal
Pu	242	DT, Fission, Thermal
Am	242	DT, Fission, Thermal
Am	242m	Thermal
Am	243	Thermal
Cm	242	Fission
Cm	243	Fission, Thermal
Cm	244	Fission, Thermal, SF
Cm	245	Thermal
Cm	246	Fission, SF
Cm	248	Fission, SF
Cf	249	Thermal
Cf	250	SF
Cf	251	Thermal
Cf	252	SF
Es	253	SF
Es	254	Thermal
Fm	254	SF
Fm	255	Thermal
Fm	256	SF

Table 2: Isotopes with independent fission yield data [5] available in FIER and the associated energy groups.

### A.2.2 Custom Independent Fission Yield Data

Users may also create and use their own yield files. To run FIER in this mode, replace `YIELDS:ER` with `YIELDS:FILE` and write the file path in the following line of the input deck in place of the ordered triple. This is detailed further in Appendix B.

Any reliable source for decay yields can be used for this file, but the format must be: **Z, A, I, Y, dY** where **Y, dY** are the yield percentages and uncertainty for each isotope (**Z, A, I**).

For example, to obtain fission yields from the NNDC SIGMA library [5]:

1. Navigate to <https://www.nndc.bnl.gov/sigma/index.jsp>.
2. Select Neutron-induced fission yields from the sub-library drop-down menu.
3. Select the desired element on the periodic table.
4. Select the desired isotope on the sidebar to the right.
5. In the green box to the right of the isotope, click on the link *interpreted* to the right of *fission yields*.
6. Find the table corresponding to the desired irradiation energy and then parse the information into a comma separated format.
7. Save this as a .csv file and place in the `/input_data` folder. In your input deck, use the flag `YIELDS:FILE` on Line 6 with the path to the new yields file on Line 7 of your input deck as `/input_data/yourfile.csv`.
8. You are all set to run FIER with the new isotope!

## B The Input Deck

Table 3 details each line of the input deck, one line per row. Bold single letters (e.g., **X**) represent input variables whose possible values are detailed under the definition column. Write any terms in all caps exactly as given (e.g., for INITIALIZE, write INITIALIZE). White space to the right of any line is ignored. Lists of values are comma but not space separated. The keywords INITIALIZE, IRRADIATION, POPULATIONS, COUNTS, and END are required in the input deck.

Line #	SYNTAX	DEFINITION
1	MODE: <b>X</b>	If <b>X</b> = MONTECARLO <b>N</b> run Monte Carlo uncertainty analysis with <b>N</b> trials. If <b>X</b> = SINGLE omit Monte Carlo uncertainty analysis.
2	<b>X</b> DECAY PREDICTION	If <b>X</b> = ON, decay prediction is on. If <b>X</b> = OFF, decay prediction is off. See Section 5.1 for more information.
3	/path/to/isotopes.csv	Half-lives file input with .csv format (see isotopes.csv for example)
4	/path/to/decays.csv	Decay mode file input with .csv format (see decays.csv for example)
5	/path/to/gammas.csv	Gamma intensity file input with .csv format (see gammas.csv for example)
6	YIELDS: <b>X</b>	If <b>X</b> = ER, FIER will use an independent fission yield file in the included library, specified in the Line 7 input. If <b>X</b> = FILE, FIER will instead use a custom yields file indicated by the file path in Line 7. See Appendix A.2 for additional details.
7	<b>E,A,G</b>	If YIELDS: <b>ER</b> , FIER will use library data for the fissionable isotope. This is specified as a <i>spaceless</i> ordered triple of the element symbol ( <b>E</b> ), the atomic number ( <b>A</b> ), and the fission energy group ( <b>G</b> ). If YIELDS: <b>FILE</b> , this line should instead be a fission yields file with .csv format (write as <i>/path/to/yields.csv</i> ). See Appendix A.2 for additional details.
8	/output/chains.csv	Decay chains output file. Replace with NONE to suppress creation.
9	/output/stems.csv	Decay stems output file.
10	/output/populations.csv	Product populations output file.
11	/output/gamma_output.csv	Gamma energy spectrum output file.
12	/output/errlog.txt	Error log
13	INITIALIZE	(Required Line) Signifies beginning of list of initial populations. Should be included even if no initial populations are specified. The initial population of any species not listed under this keyword will be assumed to be zero.

Continued on next page

---

	Continued from previous page	
14	<b>Z1,A1,I1,POP1</b>	(Optional) Populations of initial species. <b>Z</b> is the atomic mass, <b>A</b> is the atomic number, <b>I</b> is the isomer number. <b>POP</b> = initial population of listed isotope.
	...	
	<b>Zn,An,In,POPn</b>	
15	IRRADIATION	(Required Line) Signifies the start of the irradiation profile. If no irradiation pattern is specified, the default is batch decay.
16	<b>ATI1,FR1</b>	(Optional) External irradiation pattern. <b>ATI</b> = absolute time interval (seconds), <b>FR</b> = fissions/second.
	...	Add one per irradiation step desired
	<b>ATIn,FRn</b>	
17	POPULATIONS	(Required Line) Signifies the start of population time interval lines.
18	<b>TIME</b>	Add absolute time to record populations at desired times. <b>TIME</b> = time (seconds w.r.t. t = 0)
19	COUNTS	(Required Line) Signifies the start of time-bin lines.
20	<b>T1,T2</b>	(Optional) <b>T1,T2</b> start and end times of time bins in which to put integrated gamma-ray emissions.
	...	
	<b>Tn1,Tn2</b>	
21	END	(Required Line) Tells FIER to stop reading time-bin lines and to end file input.

---

Table 3: Input Deck Structure

The following is an example of an input deck:

```
1 MODE:SINGLE
2 ON DECAY PREDICTION
3 input_data/isotopes.csv      ISOTOPES  FILE
4 input_data/decays.csv       DECAYS    FILE
5 input_data/gammas.csv       GAMMAS    FILE
6 YIELDS:ER
7 U,235,fission               YIELDS    FILE
8 output/decay_chains.csv     CHAINS    OUTPUT
9 output/decay_stems.csv      STEMS     OUTPUT
10 output/populations.csv     POPS      OUTPUT
11 output/gamma_output.csv     GAMMAS    OUTPUT
12 output/err_log.txt         ERROR     LOG
13 INITIALIZE
14 IRRADIATION
15 1e-6,1e30
16 POPULATIONS
17 5e-6
18 10e-6
19 COUNTS
20 1e-6,2000
21 END
```

This input deck was designed to run FIER without a Monte Carlo analysis (Line 1) (All lines here refer to the line at the example deck printed above). Decay prediction, which will intelligently guess any missing half-lives or decay daughters, is on (Line 2).

The default input libraries are being used (Lines 3-5), and yields for Uranium 235 Fission are being pulled from the included library (Line 6-7).

FIER will calculate decay chains, decay stems, populations, and delayed  $\gamma$ -ray emissions for all fission products and deposit them in the default *output/* folder (Line 8-12).

No initial populations are specified, so all populations are from fission (no line between Line 13 and Line 14). The target is being irradiated with a single microsecond pulse of 1e30 fission/second (Line 14-15).

Populations of the products are being recorded at 5 and 10 microseconds (Lines 16-18). The counting scheme yields all gamma-emissions between 1  $\mu$ s and 2000 s (Lines 19-20). The keyword END ends all line reading (Line 21).

## C Output Files

FIER writes four output files in .csv format. FIER will automatically create the files, but not any new directory, **so make sure the output folder that you specify exists.**

The first output is a file of decay chains. This is broken into blocks separated by rows of dashed lines, where the top row of each block is the parent isotope and each column (left to right) is the **Z, A, I, Excitation Energy, Decay Constant, Branching Ratio.**

The second output is a file of decay stems. This is broken into blocks separated by rows of dashed lines, where the first row is the parent isotope. Below are sub-blocks separated by rows of dashed lines where each daughter isotope in the stem is listed in order in the form **Z, A, I, Excitation Energy, Branching Ratio.**

The third output is a file of the populations of all products. The first three rows hold the **Z, A, I** values for each isotope, respectively. Thus, each column is a different isotope/isomer. The fourth row is the half-life for each isotope in seconds. The fifth row is the excitation energy of each isotope in keV. The sixth row is the initial population of each isotope. Each row after that provides the population at the time specified in the first column. If run with the Monte Carlo uncertainty estimator on, each population row is followed by an **UNC** row that contains the uncertainties on the values in the row above. Otherwise, these **UNC** rows will not be present. A representative population output file is illustrated in Table 4.

Z	23	23	24	...
A	66	66	67	...
I	0	0	0	...
t <sub>1/2</sub> (s)	3.60E-07	1.00E-09	5.00E-02	...
t <sub>1</sub> (s)/E (keV)	0	0	0	...
0	0	0	0	...
UNC	0	0	0	...
1E-6	1.99E+06	1.06E+03	4.03E+11	...
UNC	9.15E+05	6.53E+02	2.03E+11	...
⋮	⋮	⋮	⋮	⋮

Table 4: Example of population output.

The fourth output is a file of the number of  $\gamma$ -ray emission in each specified time bin. Like the population output, the first three rows are the isotope (column by column), and the fourth row is the half-life. The fifth row is the excitation energy in keV. The sixth row is the energy of the gamma emission in keV. Each subsequent row is the number of emissions in the time intervals listed in the input deck. If run with the Monte Carlo uncertainty estimator on, each gamma emission row is followed by an **UNC** row that contains the uncertainties on the values in the row above. Otherwise, these **UNC** rows will not be present. A representative  $\gamma$ -ray emission output file is illustrated in Table 5.

	Z	26	26	26	...
	A	65	65	65	...
	I	0	0	0	...
	t_1/2 (s)	0.81	0.81	0.81	...
	E_level (keV)	0	0	0	...
t0 (s)	t1 (s)/E_gamma (keV)	127.6	212.5	340.07	...
1.00E-06	2000	2.10E+11	6.86E+11	2.90E+12	...
	UNC	1.53E+11	3.57E+11	1.63E+12	...
⋮	⋮	⋮	⋮	⋮	⋮

Table 5: Example of gamma-ray emission output.