The WIMSE library format

A. Hébert

2019/08/14

Outline

General presentation

The PyNjoy 2012 system

The WimsE framework

The (n,γ) energy

PyNjoy tutorial – wimsr

Scattering resonance information

The willie5 utility

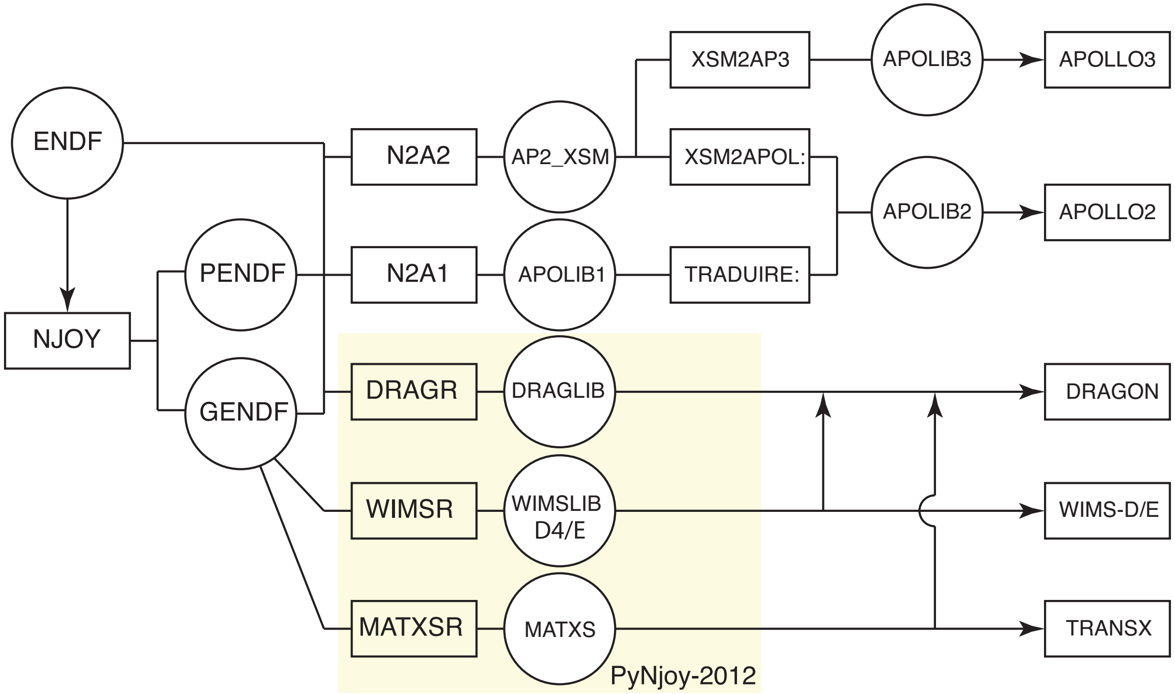
Rowlands benchmark results

Conclusions for WIMS-E support

Ressources

General presentation

Multigroup nuclear data libraries are required as input to lattice codes:



- The SCIENCE and ARCADIA systems for Pressurized Water Reactors (PWR) are based on APOLLO2 lattice code.
- The PyNjoy-2012 system developed at EPM is currently used to produce libraries in DRAGLIB and WIMSLIB-E formats.

The PyNjoy 2012 system

The NJOY saga

NJOY is the official Los-Alamos software for processing nuclear data evaluations in the ENDF format.

NJOY-99 The last distribution of the code written in Fortran-77.

NJOY-2012 A major rewrite of the code written in Fortran-90.

- Still based on UPD system for implementing updates.
- Los-Alamos license.
- Used by DRAGON, CASMO, WIMS, MONK, APOLLO, ERANOS and SCALE.

NJOY-2016 Equivalent to NJOY-2012, without UPD.

- Based on GIT for source update
- Open source

NJOY-2021 A major rewrite of the code written in C++.

- Can process evaluations in Generalized Nuclear Data (GND) format in addition to traditional ENDF-102 format
- Open source

The PyNjoy 2012 system

General presentation
The PyNjoy 2012 system
The WimsE framework
The (n,γ) energy
PyNjoy tutorial – wimsr
Scattering resonance information
The willie5 utility
Rowlands benchmark results
Conclusions for WIMS-E support
Ressources

The PyNjoy 2012 system is made of the following components:

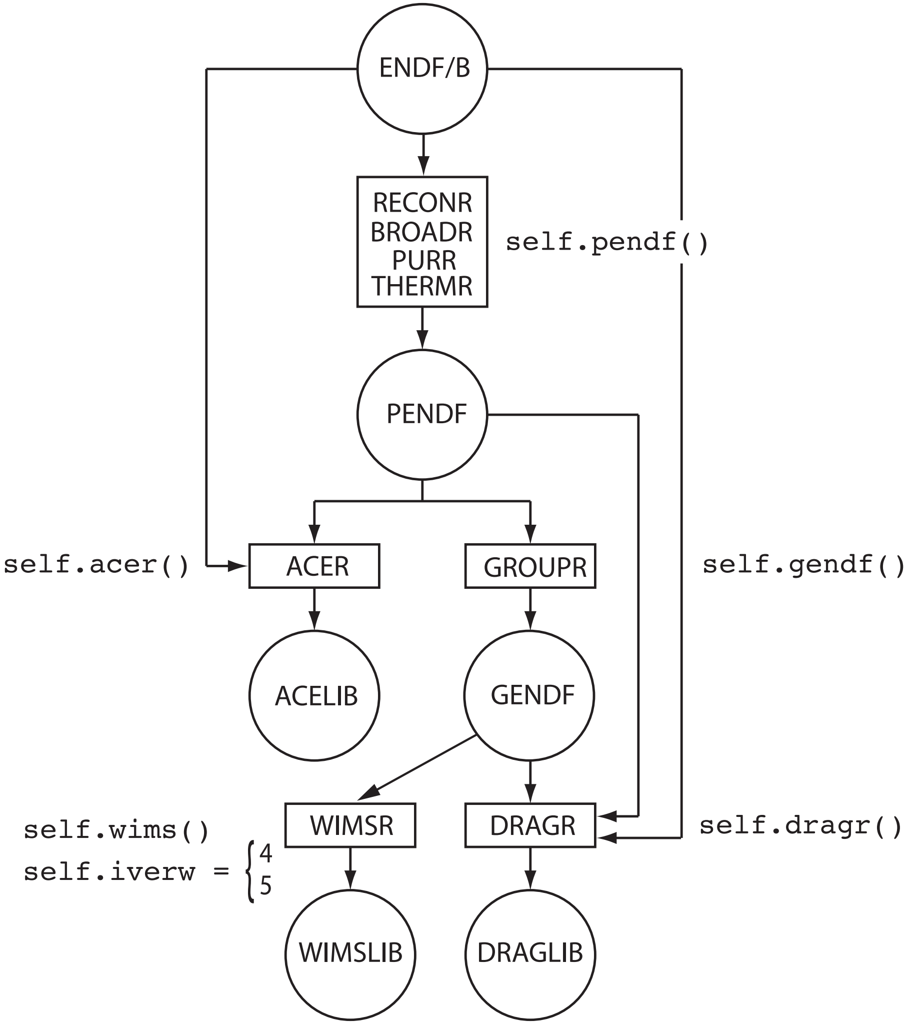
1. The NJOY-2012.0 distribution, as released by LANL.
2. A new Fortran-2003 module named `dragr` and included in the NJOY-2012 source for processing ENDF, PENDF and GENDF files and for producing the output DRAGLIB file.
3. New NJOY-2012 updates from EPM. These updates contain bug fixes and additional energy meshes for the `groupr` module:

   SHEM-281: Santamarina-Hfaiedh 281-group energy mesh used at the CEA, Areva, and EDF.
   SHEM-361: 361-group energy mesh defined as a collaboration between the CEA and EPM.
   SHEM-295: 295-group energy mesh defined at EPM.
   SHEM-315: 315-group energy mesh defined at EPM.
   Rahab-89: 89-group energy mesh defined at Atomic Energy of Canada Limited (AECL).

4. A generic Python script named PyNjoy.py for automating the preparation of datasets for NJOY. Processing of complete ENDF evaluations is greatly facilitated with the help of this script. The script is used for processing isotopes and for constructing depletion chains.
5. A collection of Python datasets, each of them corresponding to a specific evaluation (Jeff3.3, ENDFB-VIII, etc.) processed with a specific multigroup energy mesh (XMAS-172, SHEM-281, etc.).

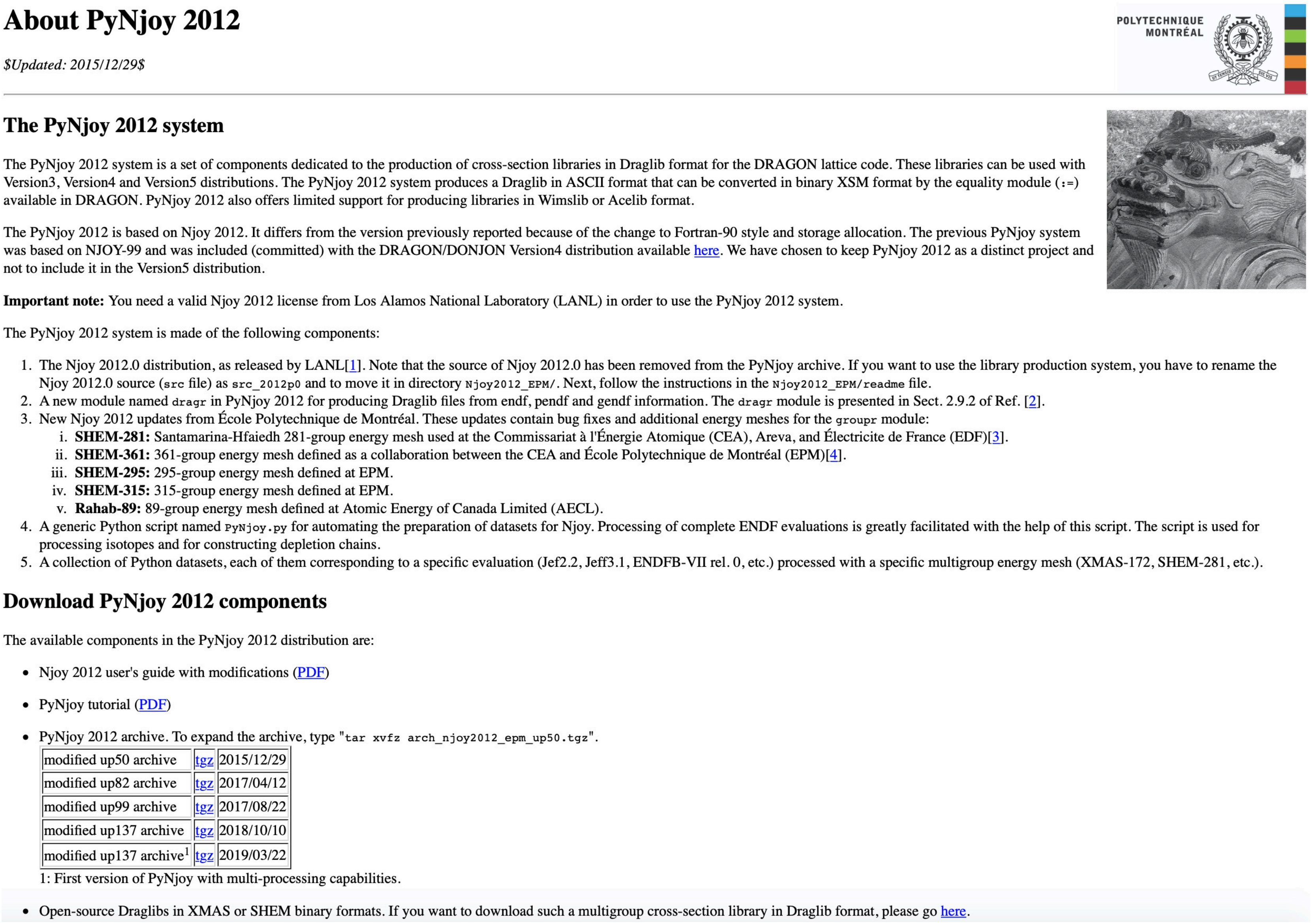
The PyNjoy 2012 system

The flowchart of NJOY-2012 processing, including `dragr`, is presented here:



The PyNjoy 2012 system

PyNjoy 2012 is available on page `http://merlin.polymtl.ca/pynjoy2012.htm`



About PyNjoy 2012

$Updated: 2015/12/29$

The PyNjoy 2012 system

The PyNjoy 2012 system is a set of components dedicated to the production of cross-section libraries in Draglib format for the DRAGON lattice code. These libraries can be used with Version3, Version4 and Version5 distributions. The PyNjoy 2012 system produces a Draglib in ASCII format that can be converted in binary XSM format by the equality module (`:=`) available in DRAGON. PyNjoy 2012 also offers limited support for producing libraries in Wimslib or Acelib format.

The PyNjoy 2012 is based on Njoy 2012. It differs from the version previously reported because of the change to Fortran-90 style and storage allocation. The previous PyNjoy system was based on NJOY-99 and was included (committed) with the DRAGON/DONJON Version4 distribution available here. We have chosen to keep PyNjoy 2012 as a distinct project and not to include it in the Version5 distribution.

Important note: You need a valid Njoy 2012 license from Los Alamos National Laboratory (LANL) in order to use the PyNjoy 2012 system.

The PyNjoy 2012 system is made of the following components:

1. The Njoy 2012.0 distribution, as released by LANL[1]. Note that the source of Njoy 2012.0 has been removed from the PyNjoy archive. If you want to use the library production system, you have to rename the Njoy 2012.0 source (`src` file) as `src_2012p0` and to move it in directory `Njoy2012_EPM/`. Next, follow the instructions in the `Njoy2012_EPM/readme` file.
2. A new module named `dragr` in PyNjoy 2012 for producing Draglib files from endf, pendf and gendf information. The `dragr` module is presented in Sect. 2.9.2 of Ref. [2].
3. New Njoy 2012 updates from École Polytechnique de Montréal. These updates contain bug fixes and additional energy meshes for the `groupr` module:
   i. **SHEM-281:** Santamarina-Hfaiedh 281-group energy mesh used at the Commissariat à l'Énergie Atomique (CEA), Areva, and Électricite de France (EDF)[3].
   ii. **SHEM-361:** 361-group energy mesh defined as a collaboration between the CEA and École Polytechnique de Montréal (EPM)[4].
   iii. **SHEM-295:** 295-group energy mesh defined at EPM.
   iv. **SHEM-315:** 315-group energy mesh defined at EPM.
   v. **Rahab-89:** 89-group energy mesh defined at Atomic Energy of Canada Limited (AECL).
4. A generic Python script named `PyNjoy.py` for automating the preparation of datasets for Njoy. Processing of complete ENDF evaluations is greatly facilitated with the help of this script. The script is used for processing isotopes and for constructing depletion chains.
5. A collection of Python datasets, each of them corresponding to a specific evaluation (Jef2.2, Jeff3.1, ENDFB-VII rel. 0, etc.) processed with a specific multigroup energy mesh (XMAS-172, SHEM-281, etc.).

Download PyNjoy 2012 components

The available components in the PyNjoy 2012 distribution are:

- Njoy 2012 user's guide with modifications (PDF)
- PyNjoy tutorial (PDF)
- PyNjoy 2012 archive. To expand the archive, type "`tar xvfz arch_njoy2012_epm_up50.tgz`".

modified up50 archive	tgz	2015/12/29
modified up82 archive	tgz	2017/04/12
modified up99 archive	tgz	2017/08/22
modified up137 archive	tgz	2018/10/10
modified up137 archive[1]	tgz	2019/03/22

1: First version of PyNjoy with multi-processing capabilities.

- Open-source Draglibs in XMAS or SHEM binary formats. If you want to download such a multigroup cross-section library in Draglib format, please go here.

The WimsE framework

A. Hébert, “A Proposal Beyond the Wims Library Update Project,” *M&C*, Jeju, Korea, April 16–20, 2017.

The WIMS-E library production framework is made of the following components (module `dragr` is not used):

1. The NJOY-2012.0 distribution, as released by LANL.
2. New NJOY-2012 updates from École Polytechnique de Montréal (EPM). These updates contain bug fixes and
   - corrections for the `iverw=5` option of `wims` module.
   - additional energy meshes for the `groupr` module.
3. A generic Python script named PyNjoy.py for automating the preparation of datasets for NJOY. The script is used for
   - processing isotopes and
   - for constructing depletion chains.
4. A collection of Python datasets.
   - E. g.: `endfb7r0_shem281_wimsE.py`
5. A utility named `willie5` to produce the final Wimslib and to perform ascii< – >binary conversion. Similar to the `willie` utility of the WLUP project.

The WimsE framework

The WIMS-D4 format shortcomings

The improvements obtained with a transition from WIMS-D4 to WIMS-E are:

- With the WIMS-D4 format, temperature-independent P_1 scattering cross sections are available only for four isotopes: H in H_2O, D in D_2O, ^{16}O and graphite. All other isotopes have transport-corrected P_0 scattering cross sections. With the WIMS-E format, all isotopes have both temperature-dependent P_0 and P_1 scattering cross sections.
- With the WIMS-D4 format, all fissile isotopes are using the same fission spectrum. With the WIMS-E format, each fissile isotope has a specific fission spectrum.
- (n,2n) cross sections are only available in a WIMS-E formatted library.
- Resonance tables for the scattering reaction are only available in a WIMS-E formatted library. With the WIMS-D4 format, cross sections other than absorption and fission are self-shielded at a fixed value given by the `sgref` parameter (selected during the NJOY run).

The WimsE framework

Resonances tables

Two-dimensional *resonance integral* tables are available in a WIMS-E library for three reactions: absorption, fission and scattering as a function of dilution σ_e and absolute temperature T. They are defined in energy group g as

$$\begin{aligned} I_{\mathrm{a},g}(\sigma_e,T) &= \bar{\sigma}_{\mathrm{a},g}(\sigma_e,T)\,\bar{\phi}_g(\sigma_e,T) \\ I_{\mathrm{f},g}(\sigma_e,T) &= \overline{\nu\sigma}_{\mathrm{f},g}(\sigma_e,T)\,\bar{\phi}_g(\sigma_e,T) \\ I_{\mathrm{s},g}(\sigma_e,T) &= \bar{\sigma}_{\mathrm{s},g}(\sigma_e,T)\,\bar{\phi}_g(\sigma_e,T) \end{aligned} \tag{1}$$

respectively, where $\bar{\sigma}_{\rho,g}(\sigma_e,T)$ is the self-shielded cross section for reaction ρ, as computed by NJOY. The self-shielded flux in group g is defined as

$$\bar{\phi}_g(\sigma_e,T) = \frac{\sigma_e}{\sigma_e + \bar{\sigma}_{\mathrm{a},g}(\sigma_e,T)} = \frac{\sigma_e - I_{\mathrm{a},g}(\sigma_e,T)}{\sigma_e}. \tag{2}$$

Resonances tables for scattering $I_{\mathrm{s},g}(\sigma_e,T)$ are only available in a WIMS-E formatted library.

The WimsE framework

The self-shielded P_0 transfer cross sections are the product of the P_0 scattering law times the self-shielded scattering cross sections. They are assumed to be equal to

$$\bar{\sigma}_{\mathrm{s},h\leftarrow g}(\sigma_e,T)=\bar{\sigma}_{\mathrm{s},h\leftarrow g}(\infty,T)\,\frac{\bar{\sigma}_{\mathrm{s},g}(\sigma_e,T)}{\bar{\sigma}_{\mathrm{s},g}(\infty,T)} \tag{3}$$

where

$$\bar{\sigma}_{\mathrm{s},g}(\sigma_e,T)=\sum_h \bar{\sigma}_{\mathrm{s},h\leftarrow g}(\sigma_e,T). \tag{4}$$

- Equation (3) is an approximation related to the choice of the WIMS-E library format.
- Approximation (3) is avoided in `dragr` and in DRAGLIB libraries.
- Approximation (3) is avoided in `matxsr` and in MATXS libraries (Los Alamos format).
- Approximation (3) is used in APOLLO2 cross-section libraries.

The WimsE framework

wimsr module update

The following update is proposed in NJOY version 2012.82 to support `iverw` = 5 in `wimsr` module:

```
*/ wimsr -- 2october2016
*/ correct bugs related to iverw=5 implementation
*d wimsr.34
   integer::ifiss,ifissr,nfiss
*d wimsr.438
   real(kr)::xid,siglam,sigb,siga,sigs,sig
*i wimsr.450
   ifissr=0
*i wimsr.522
   if ((mth.gt.50).and.(mth.le.91)) go to 162
*d wimsr.541
*d wimsr.582,wimsr.583
   !--elastic or inelastic scattering
   else if ((mth.eq.2).or.((mth.gt.50).and.(mth.le.91))) then
*i wimsr.588
   else
      go to 300
*i wimsr.664
            sigs=elas(loc+iz-1)
*i wimsr.666
            if (iverw.ne.4) elas(loc+iz-1)=sigb*sigs/(sigb+siga)
*i wimsr.671
   ifissr=jfiss
*i wimsr.932
   if (iverw.eq.5) isg=1
*i wimsr.2043
   if((ifis.eq.3).and.(ifissr.ne.3)) ifis=2
```

The (n,γ) energy

- In the official specification, a unique energy release value is given for fission in the depletion chain.
- Consider the ^{241}Pu isotope. The PyNjoy data for method `wims()` reads

```
endfb.yields = """
   77   2.06470E-11 ! (ntis) (efiss)
94242   1.00000E+00 ! (id_ng) (br_ng)
95241   1.53100E-09 ! (id_lambda) (lambda)
36083   2.01460E-03 ! (id_fp1) (yield_fp1)
42095   3.92560E-02 ! (id_fp2) (yield_fp2)
...
64160   2.04990E-04 ! (id_fp75) (yield_fp75)
"""
```

where

ntis: number of lines in the `yield` specification for this isotope. A non-depleting isotope has `ntis`=2.

efiss: energy per fission (Joule), including radiative capture energy

id_ng: id of the isotope produced by (n,γ) reaction

br_ng: corresponding branching ratio (usually equal to 1.0)

id_lambda: id of the isotope produced by radioactive decay reaction

lambda: corresponding radioactive decay constant (s^{-1}).

id_fp1: id of the first fission product

yield_fp1: yield of the first fission product

The (n,γ) energy

- An extension to the WIMS-D4 and WIMS-E formats is available to provide both fission and (n,γ) energies as separate values.
- No modifications are required in module `wimsr`.
- The specification extension consists to provide specific values of energy release for different nuclear reactions. The PyNjoy data for method `wims()` reads

```
endfb.yields = """
   78  1.81021E-11 ! (ntis) (efiss)
94242  1.00000E+00
95241  1.53100E-09
   -1  6.08809E-13 ! (nener) (energy)
36083  2.01460E-03
42095  3.92560E-02
...
64160  2.04990E-04
"""
```

where

`ntis:` number of lines in the `yield` specification for this isotope. A non-depleting isotope has `ntis`=3.

`efiss:` energy per fission (Joule), not including radiative capture energy

`nener:` type of reaction other than fission for producing energy (-1: (n,γ); -2: radioactive decay)

`energy:` energy produced by reaction `nener`

PyNjoy tutorial – wimsr

General presentation
The PyNjoy 2012 system
The WimsE framework
The (n,γ) energy
PyNjoy tutorial – wimsr
Scattering resonance information
The willie5 utility
Rowlands benchmark results
Conclusions for WIMS-E support
Ressources

En example of PyNjoy dataset for processing ^{238}U from ENDF/B-VIIr0 with a SHEM281energy mesh follows (`wimsr`–specific data is in blue):

```
#!/usr/local/bin/python
from PyNjoy import *
from os import uname
endfb = PyNjoy()
endfb.evaluationName = "/tmp/shem281_endfb7r0_u238wimsE"
endfb.execDir = "../" + uname()[0]
endfb.nstr = 24
endfb.iwt = -4
endfb.autolib = (22.53556, 1345.061, 0.001)
endfb.jp1 = 0
endfb.iverw = 5
endfb.yields = None
endfb.purr = None
endfb.scatteringLaw = None
endfb.temperatures = ( 293., 550., 900., 1200., 2000. )
endfb.eFiss = None
endfb.legendre = 0

endfb.hmat = "U238"
endfb.mat = 9237
endfb.evaluationFile = "$HOME/evaluations/ENDFB7r0/n-ENDF-VII0.endf/n-092_U_238.endf"
endfb.fission = 2 # fission with delayed neutrons
endfb.potential = 11.8237
endfb.dilutions = ( 1.e10, 2.e4, 3.6e3, 1.0e3, 2.6e2, 140., 64., 52., 28., 10.)
endfb.pendf()
endfb.gendf()
endfb.wmat = 92238
endfb.goldstein = 0.2
endfb.wims()
```

PyNjoy tutorial – wimsr

General presentation
The PyNjoy 2012 system
The WimsE framework
The (n,γ) energy
PyNjoy tutorial – wimsr
Scattering resonance information
The willie5 utility
Rowlands benchmark results
Conclusions for WIMS-E support
Ressources

The output of Njoy-2012 execution is

```
Alains-MacBook-Pro:python alainhebert$ python endfb7r0_shem281_u238wimsE.py
 --- make pendf for U238 ---
 njoy 2012.82  20jan17                                       04/13/17 10:11:12
 *****************************************************************************
 moder...                                                                 0.0s

 reconr...                                                                0.1s

 broadr...                                                              183.1s
      293.0 deg                                                         183.9s
      550.0 deg                                                         203.1s
      900.0 deg                                                         225.2s
     1200.0 deg                                                         250.2s
     2000.0 deg                                                         277.4s

 unresr...                                                              309.1s

 thermr...                                                              310.3s
 wrote thermal data for temp = 2.9300E+02                               313.8s
 wrote thermal data for temp = 5.5000E+02                               317.2s
 wrote thermal data for temp = 9.0000E+02                               320.5s
 wrote thermal data for temp = 1.2000E+03                               323.7s
 wrote thermal data for temp = 2.0000E+03                               326.9s

 moder...                                                               326.9s
                                                                        397.5s
 *****************************************************************************
```

PyNjoy tutorial – wimsr

```
--- make gendf for U238 ---
njoy 2012.82  20jan17                                       04/13/17 10:17:55
*****************************************************************************
moder...                                                                 0.0s

moder...                                                                 0.0s

groupr...                                                                6.8s

moder...                                                               148.4s
                                                                       149.7s
*****************************************************************************
--- make wimslib for U238 ---
njoy 2012.82  20jan17                                       04/13/17 10:20:27
*****************************************************************************
moder...                                                                 0.0s

wimsr...                                                                 0.2s
---message from wminit---mat 9237 mf  3 has both mt18 and mt19
                         mt18 will be used
---message from wminit---mat 9237 has no mf3, mt252
                         isotropic c.m.scattering will be assumed
                                                                         0.3s
*****************************************************************************
```

Scattering resonance information

- Scattering resonance information in the Wimslib is generally computed at infinite dilution.
- It is possible to select a dilution smaller than infinity by setting the `sgref` variable in `wimsr` module to a value chosen in the set of dilutions used by the `groupr`, `unresr` or `purr` modules.
- The WLUP libraries in WIMS-D4 format use `sgref` = 28 barn for U238 and `sgref` = 800 barn for U235. The PyNjoy data for U238 used by the WLUP project is

```
endfb.iverw = 4
endfb.sgref = 28.0
```

- A study made with DRAGON5 leads to the following dilution values for U238:

Case	U238 average dilution (barn)
Rowlands UOX pincell	65.0
Rowlands MOX pincell	77.0
UOX 17×17 PWR assembly	135.0
MOX 17×17 PWR assembly	135.0

The willie5 utility

- The NJOY output is a set of ASCII files, one per isotope, located in directory `/tmp/shem281_endfb7r0_u238wimsE/`.
- The following operations remains to be done:
  - Concatenate all the isotopic directories
  - Construct the depletion data for the library
  - Convert the library to binary WIMS-E format

The `willie5` utility is a simple conversion tool for a WIMSLIB in WIMS-E format. It has four commands:

`MAKE`: Catenate the individual isotopic WIMSLIB outputs from the `wimsr` module of NJOY (executed with `iverw` = 5) into a single ASCII library in WIMS-E format

`FOBI`: Convert a WIMS library from ASCII to sequential binary.

- WIMS-D4 ASCII to WIMS-D4 sequential binary
- WIMS-E ASCII to WIMS-E sequential binary
- WIMS-D4 ASCII to WIMS-E sequential binary
- SPIC format ASCII to WIMS-E sequential binary (only available in SPIC version)

`BIFO`: Convert a WIMS-E library from sequential binary to ASCII.

- WIMS-E sequential binary to WIMS-E ASCII

`END`: End the processing

The willie5 utility

The Willie5 utility executes in a console (user's commands are in blue):

```
Alains-MacBook-Pro:willie5 alainhebert$ ./willie5
$Enter wimslib version (4 or 5)
5
$Enter the selected option
MAKE
$Process option MAKE
$Enter NJOY working directory:
/tmp/shem281_endfb7r0_u238wimsE/
$Enter input Wimslib-E ascii name [wimslib_out.txt]:
libshem281_endfb7r0_u238wimsE.txt
$number of nuclides in NJOY working directory=    1
$Process isotopic wimslib:  /tmp/shem281_endfb7r0_u238wimsE/wimslibU238
$Number of fission products=    0
$Number of fissile isotopes=    0
$End of NJOY directory processing: /tmp/shem281_endfb7r0_u238wimsE/
$Enter the selected option
FOBI
$Process option FOBI
$Enter input Wimslib-E ascii name [wimslib_out.txt]:
libshem281_endfb7r0_u238wimsE.txt
$Enter output Wimslib-E binary name [wimslib_out.bin]:
libshem281_endfb7r0_u238wimsE.bin
$Process isotope  92238
$Enter the selected option
END
$Successful end of processing
```

Rowlands benchmark results

We have based our validation study on a subset made up of eight Rowlands pin-cell benchmark cases.

- The comparisons were made for light-water reactor pin-cells without leakage.
- Two types of pin-cell were studied,
  - UO_2 fuelled (UOX),
  - $UPuO_2$ fuelled, in two versions with different isotopic compositions (MOX–1 and MOX–2).

UOX case 1:	Isothermal 293 K
UOX case 2:	Reduced H2O Density
UOX case 3:	Fuel at 900 K
UOX case 4:	Isothermal 574 K
MOX case 1:	MOX Fuel 1,Isothermal 300 K
MOX case 2:	MOX Fuel 1, Fuel at 560 K
MOX case 3:	MOX Fuel 2,Isothermal 300 K
MOX case 4:	MOX Fuel 2, Fuel at 560 K

Rowlands benchmark results

K_{eff} results

	UOX case 1	UOX case 2	UOX case 3	UOX case 4
WIMS-D4				
`sgref` = 28 b	1.385201	1.330805	1.295787	1.309841
`sgref` = 64 b	1.385240	1.330844	1.295833	1.309883
`sgref` = 140 b	1.385305	1.330916	1.295913	1.309958
`sgref` = 20000 b	1.386631	1.332373	1.297506	1.311475
`sgref` = 1.0×10^{10} b	1.386865	1.332636	1.297788	1.311745
WIMS-E	1.385100	1.330629	1.295625	1.309671
DRAGLIB	1.385281	1.330901	1.295877	1.309990

	MOX case 1	MOX case 2	MOX case 3	MOX case 4
WIMS-D4				
`sgref` = 28 b	1.223384	1.207996	1.268746	1.254351
`sgref` = 64 b	1.223392	1.208006	1.268759	1.254367
`sgref` = 140 b	1.223406	1.208025	1.268781	1.254400
`sgref` = 20000 b	1.223767	1.208515	1.269336	1.255093
`sgref` = 1.0×10^{10} b	1.223824	1.208600	1.269433	1.255219
WIMS-E	1.223208	1.207827	1.268565	1.254176
DRAGLIB	1.223446	1.208057	1.268818	1.254421

Rowlands benchmark results

- In the case of pin-cell UOX Rowlands benchmark, the average dilution of ^{238}U isotope can be estimated around 100 barn. The expected result should therefore correspond to the WIMS-D4 calculation with `sgref` $= 140$ b.
- The corresponding K_{eff} is consistent with the value obtained with a DRAGLIB, but is 20 to 30 pcm above the WIMS-E result.
- This discrepancy with the WIMS-E library is due to the approximation in Eq. (3) which consists to neglect the self-shielding of the scattering law (but *not* to neglect the self-shielding of the scattering cross section).
- Similar conclusions can be made about the MOX results.

Conclusions for WIMS-E support

- We are proposing a framework to upgrade from WIMS-D4 to WIMS-E library format.
- Such a replacement would improve the overall quality of the cross section data used in lattice codes actually based on WIMS-D4 libraries.
- The similarity between the two formats would facilitate the software modifications inside the lattice codes (few weeks work to make the modifications).
- Capability to read WIMS-E formatted libraries is available in DRAGON version 5.0.3 and up.
- The complete system (with the exception of the initial NJOY-2012.0 source) is and will remain openly available and developed under the Lesser General Public License and is available on site `http://merlin.polymtl.ca/`.
- The current implementation is based on update 137 of NJOY-2012.
  - Update 137 was provided by the NEA data bank.
  - Update 137 is required for processing ENDF/B-VIII and Jeff3.3 evaluations.

Ressources

PyNjoy2012 distribution:

`https://merlin.polymtl.ca/pynjoy2012.htm`

Willie5:

System for processing WIMSE libraries (`tgz`)

WIMSE libraries:

Use `WIMSE` keyword in module `LIB:` of DRAGON5.

- SHEM-281, ENDF/B-VII r0, little endian (gz)
- SHEM-281, ENDF/B-VII r0, big endian (gz)