IAEA 60 Years

Atoms for Peace and Development

# Development status of endf-userpy

Georg Schnabel, Daniel Lopez Aldama
g.schnabel [at] iaea.org
dlopezaldama [at] gmail.com

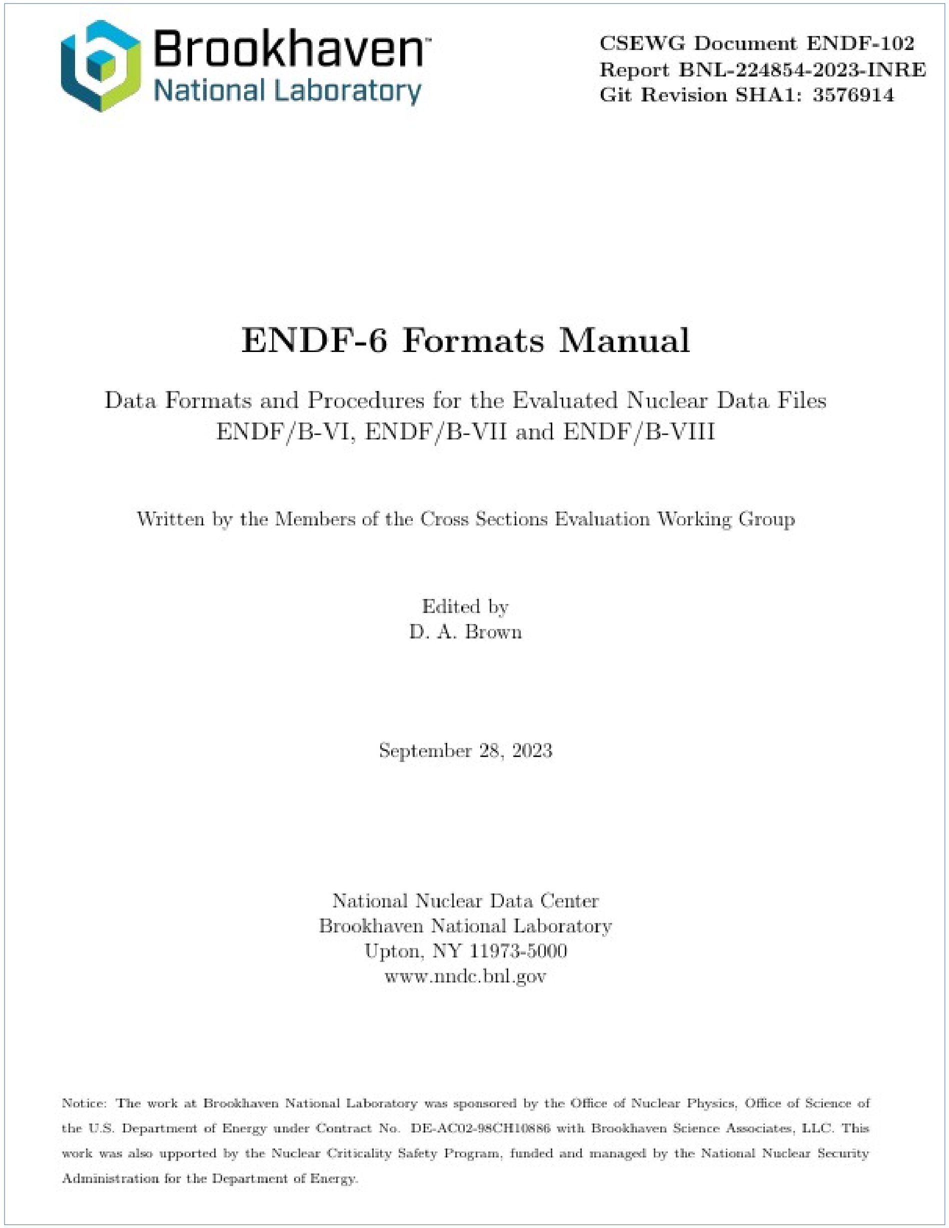
Nuclear Data Section
Division of Physical and Chemical Sciences NAPC
Department for Nuclear Sciences and Applications
IAEA, Vienna

Internal presentation

27 November 2024

# Outline

- Recap: ENDF-6 and endf-parserpy
- ENDF interpretation: endf-userpy
- Some examples
- What’s next?



CSEWG Document ENDF-102
Report BNL-224854-2023-INRE
Git Revision SHA1: 3576914

# ENDF-6 Formats Manual

Data Formats and Procedures for the Evaluated Nuclear Data Files
ENDF/B-VI, ENDF/B-VII and ENDF/B-VIII

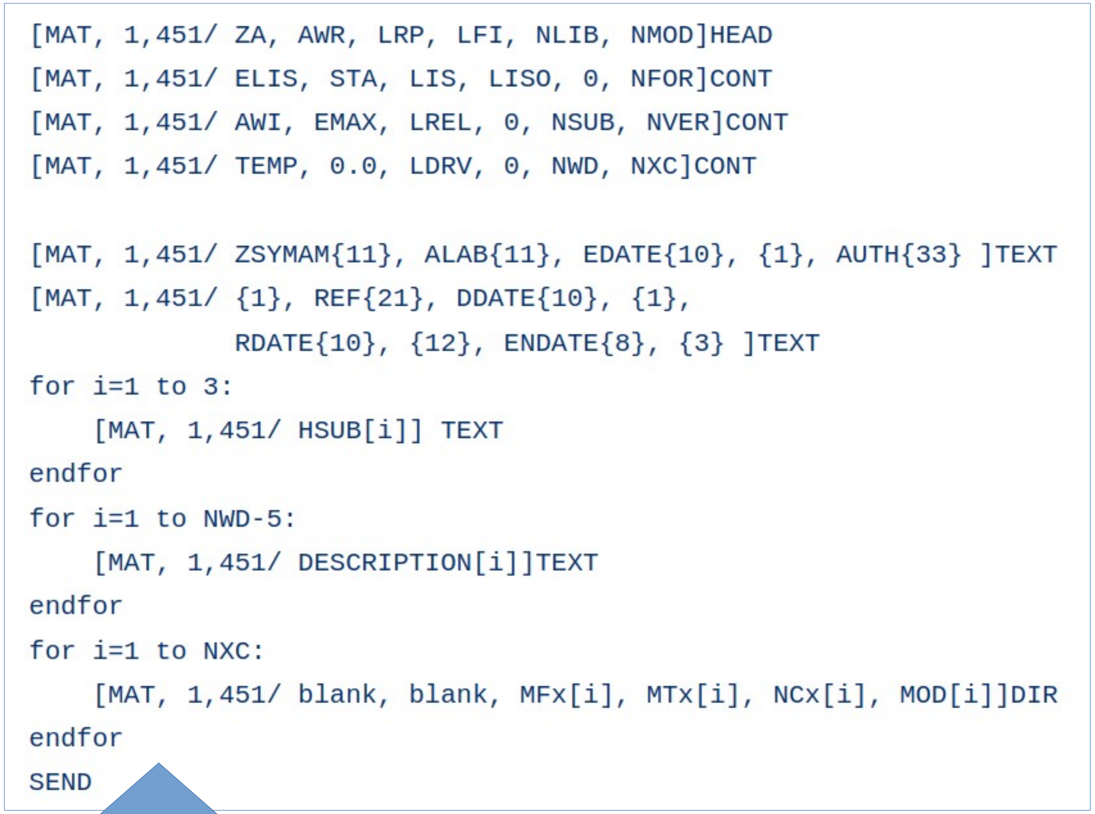
Written by the Members of the Cross Sections Evaluation Working Group

Edited by
D. A. Brown

September 28, 2023

National Nuclear Data Center
Brookhaven National Laboratory
Upton, NY 11973-5000
www.nndc.bnl.gov

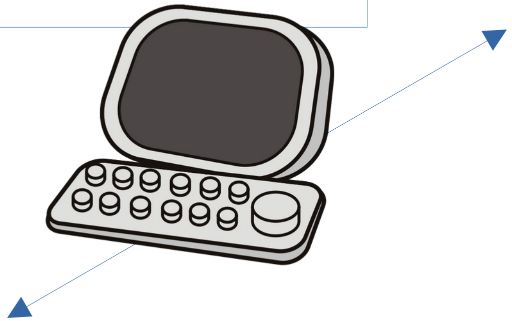
Notice: The work at Brookhaven National Laboratory was sponsored by the Office of Nuclear Physics, Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-98CH10886 with Brookhaven Science Associates, LLC. This work was also upported by the Nuclear Criticality Safety Program, funded and managed by the National Nuclear Security Administration for the Department of Energy.

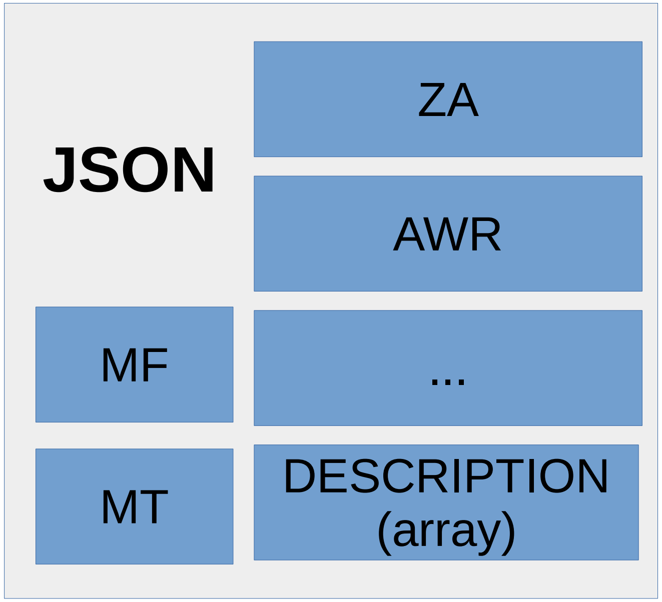


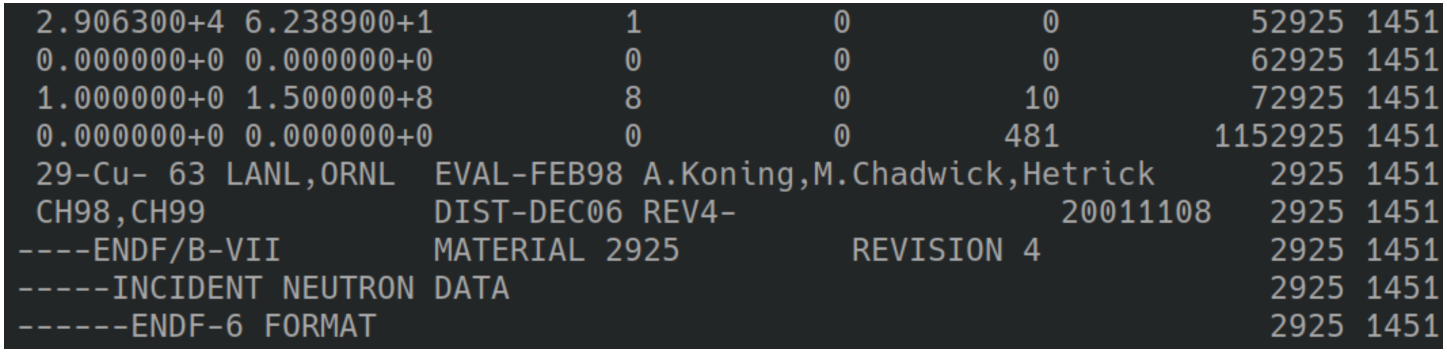
```
[MAT, 1,451/ ZA, AWR, LRP, LFI, NLIB, NMOD]HEAD
[MAT, 1,451/ ELIS, STA, LIS, LISO, 0, NFOR]CONT
[MAT, 1,451/ AWI, EMAX, LREL, 0, NSUB, NVER]CONT
[MAT, 1,451/ TEMP, 0.0, LDRV, 0, NWD, NXC]CONT

[MAT, 1,451/ ZSYMAM{11}, ALAB{11}, EDATE{10}, {1}, AUTH{33} ]TEXT
[MAT, 1,451/ {1}, REF{21}, DDATE{10}, {1},
             RDATE{10}, {12}, ENDATE{8}, {3} ]TEXT
for i=1 to 3:
    [MAT, 1,451/ HSUB[i]] TEXT
endfor
for i=1 to NWD-5:
    [MAT, 1,451/ DESCRIPTION[i]]TEXT
endfor
for i=1 to NXC:
    [MAT, 1,451/ blank, blank, MFx[i], MTx[i], NCx[i], MOD[i]]DIR
endfor
SEND
```

# Bijective Mapping: ENDF-6 ↔ Dictionary

```
[MAT, 1,451/ ZA, AWR, LRP, LFI, NLIB, NMOD]HEAD
[MAT, 1,451/ ELIS, STA, LIS, LISO, 0, NFOR]CONT
[MAT, 1,451/ AWI, EMAX, LREL, 0, NSUB, NVER]CONT
[MAT, 1,451/ TEMP, 0.0, LDRV, 0, NWD, NXC]CONT
for i=1 to NWD:
    [MAT, 1,451/ DESCRIPTION[i]]TEXT
endfor
```

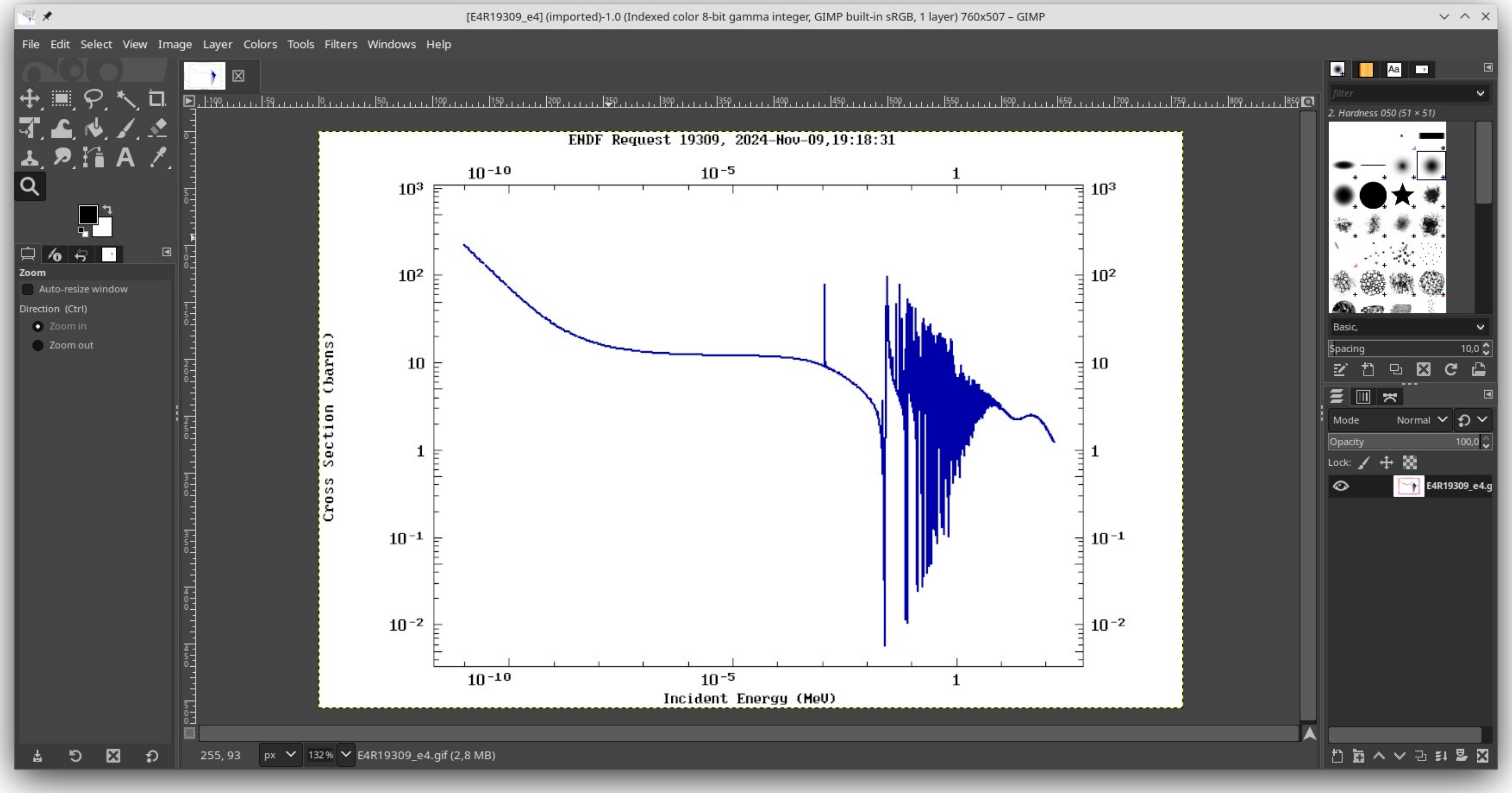






```
 2.906300+4 6.238900+1          1          0          0          52925 1451
 0.000000+0 0.000000+0          0          0          0          62925 1451
 1.000000+0 1.500000+8          8          0         10          72925 1451
 0.000000+0 0.000000+0          0          0        481       1152925 1451
 29-Cu- 63 LANL,ORNL  EVAL-FEB98 A.Koning,M.Chadwick,Hetrick      2925 1451
 CH98,CH99            DIST-DEC06 REV4-            20011108        2925 1451
----ENDF/B-VII        MATERIAL 2925         REVISION 4            2925 1451
-----INCIDENT NEUTRON DATA                                        2925 1451
------ENDF-6 FORMAT                                               2925 1451
```

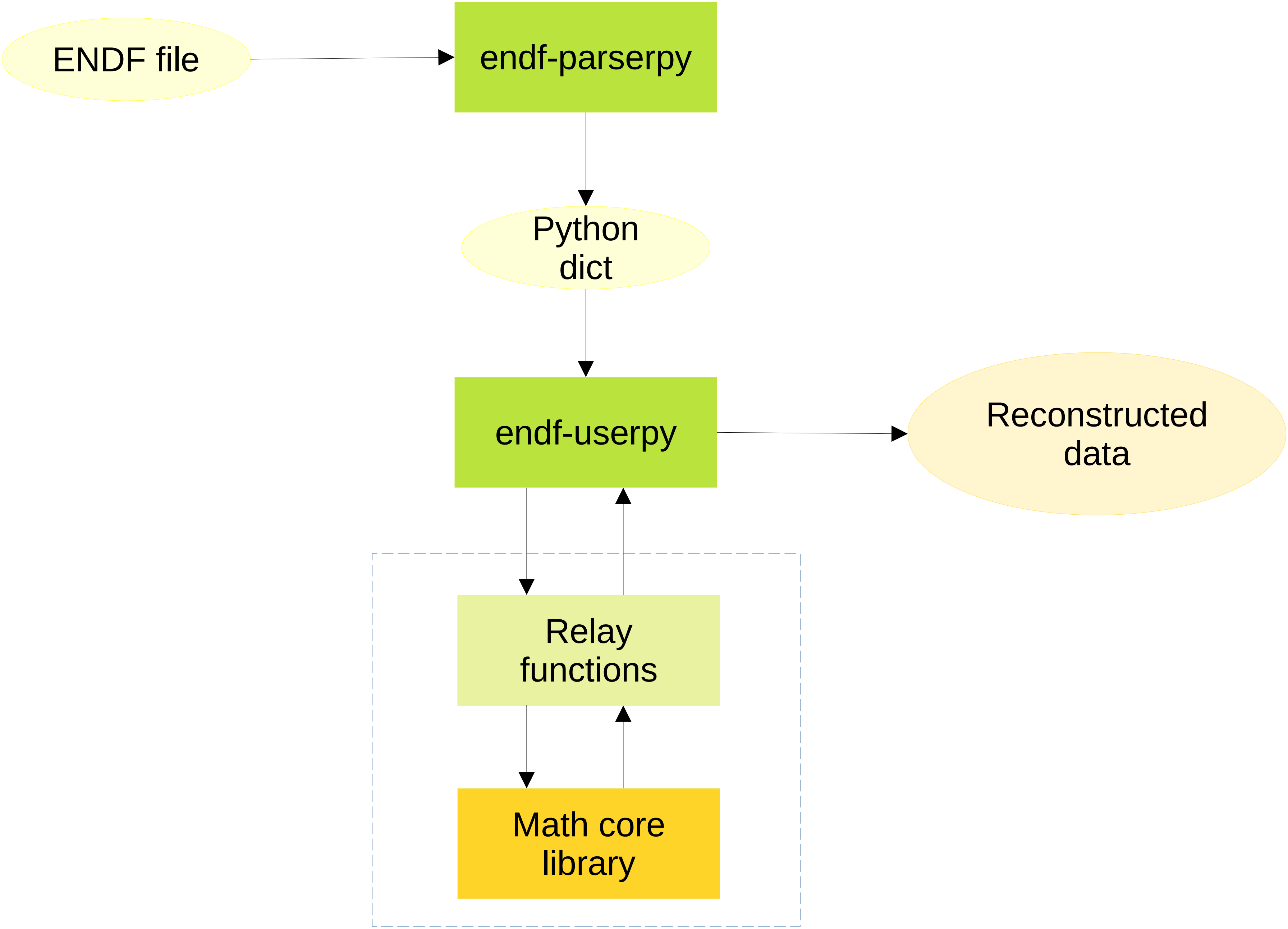
# Motivation for creating endf-userpy



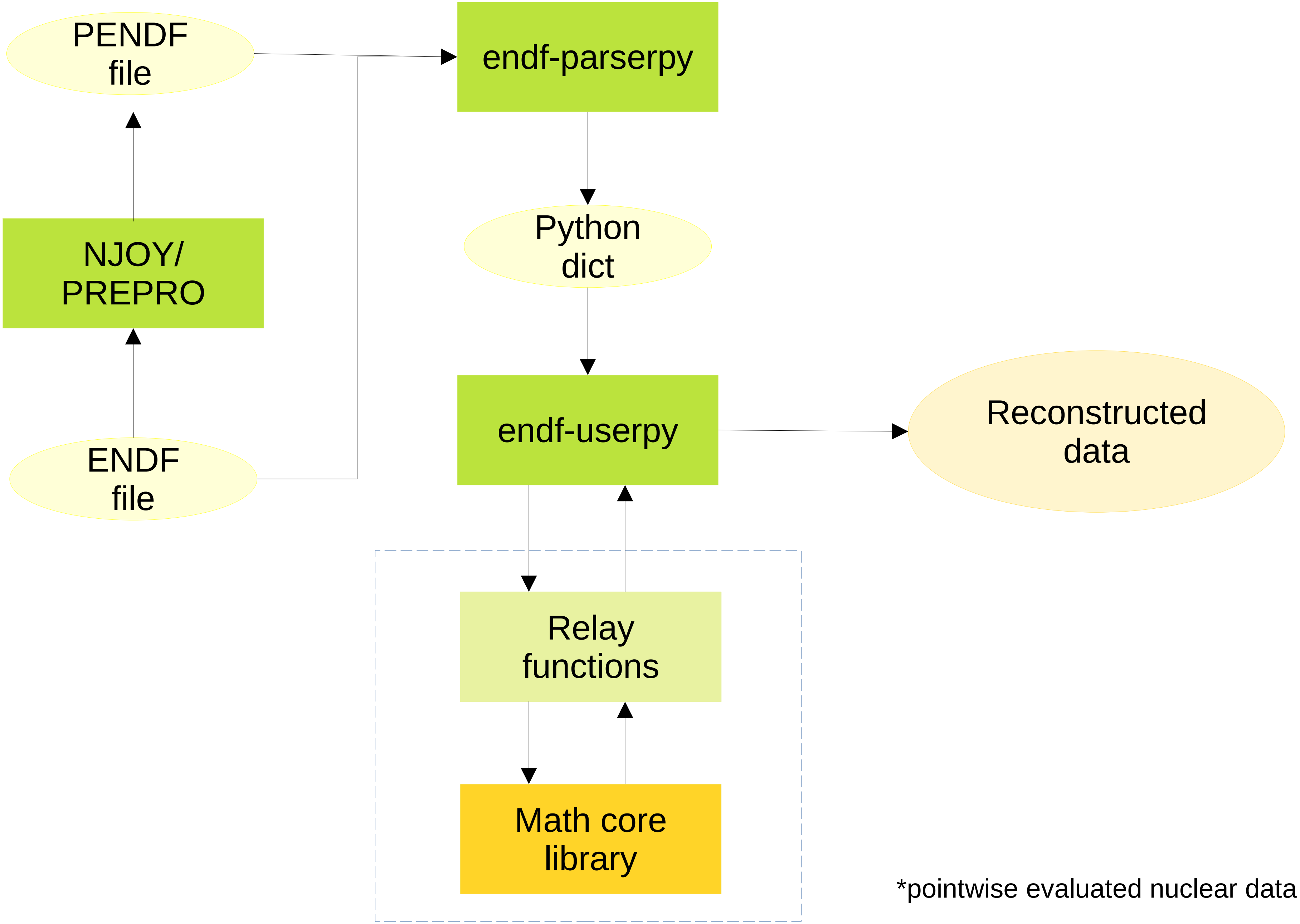
[E4R19309_e4] (imported)-1.0 (Indexed color 8-bit gamma integer, GIMP built-in sRGB, 1 layer) 760x507 – GIMP

E4R19309_e4.gif (2,8 MB)

# Package design



# Support of PENDF* files



*pointwise evaluated nuclear data file

# Math core library capabilities

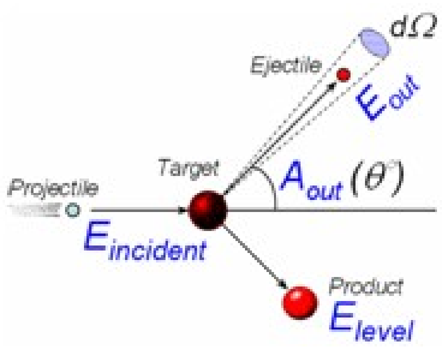
**Core functionality**

- Various interpolation schemes (lin-lin, lin-log, etc.)
- Evaluation of Legendre polynomials
- Center-of-Mass to Laboratory system conversion

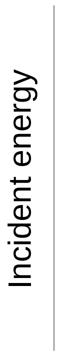
**Supports all angle-differential (LTT=1,2,3) and ddx representations (LAW=1,2,5,6,7):**

- Tabulated data (3-dimensional)
- Legendre representation
- Kalbach-Mann systematics representation
- Charged-particle elastic scattering
- N-Body Phase Space

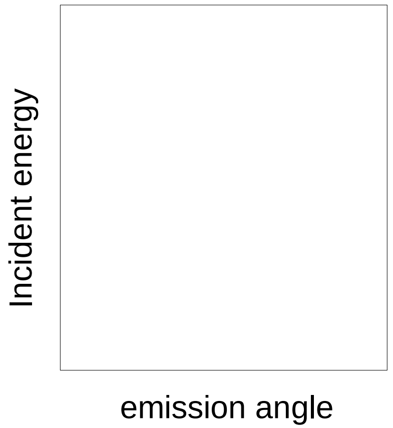
# Quantities of interest



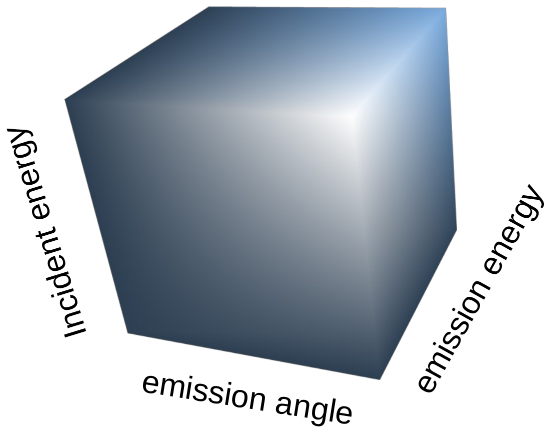
Cross Section



Angle-Differential
Cross Section



Double-Differential
Cross Section



# Three essential functions

## Cross section

compute_xs(endf_dict, mt, eincs)

## Angle-differential Cross Section

compute_dxs(end_dict, mt, zap, eincs, mus)

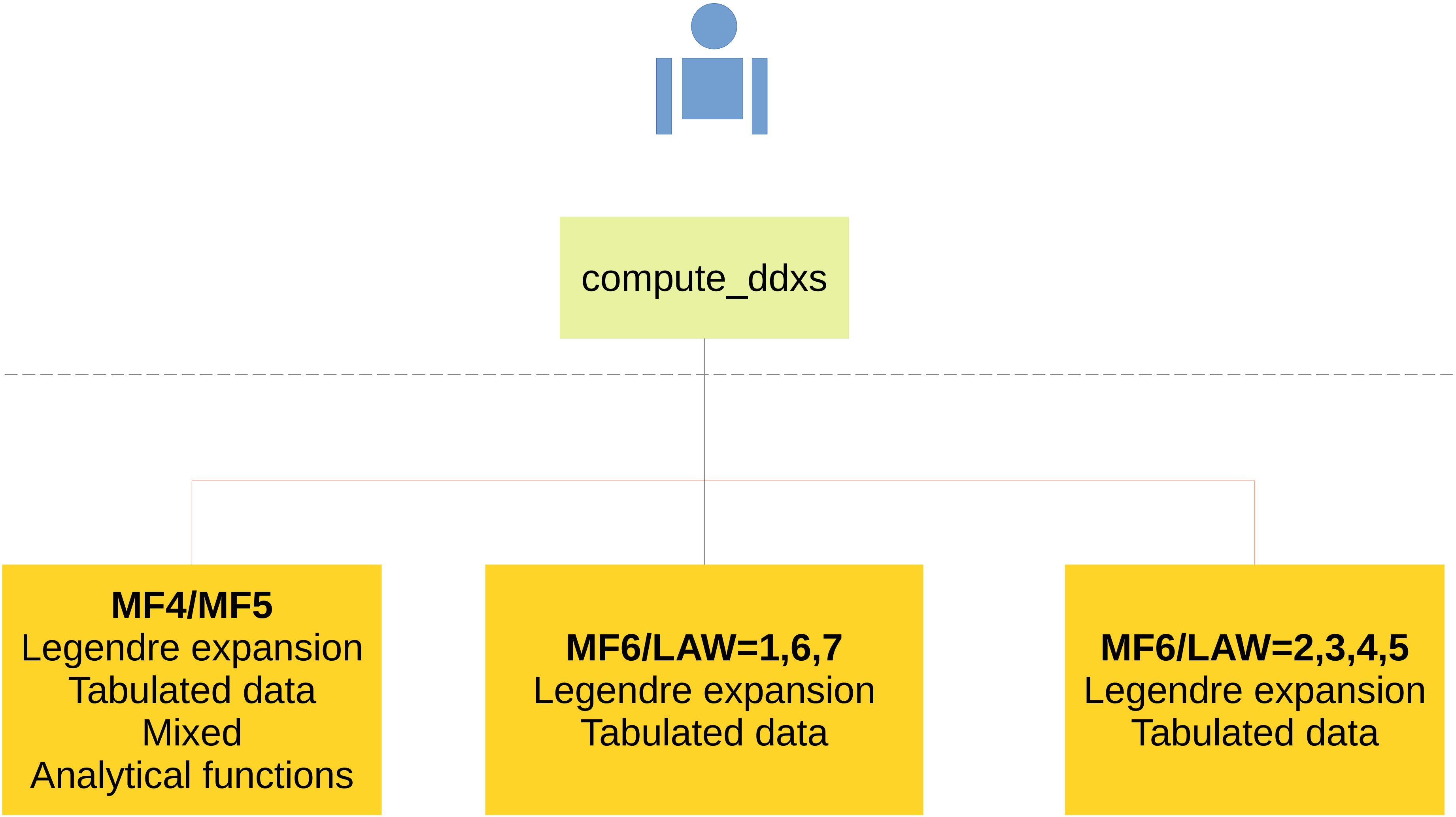
## Double-differential Cross Section

compute_ddxs(endf_dict, mt, zap, eincs, eouts, mus)

Input quantities and results  in lab system

**MT**: reaction number, e.g. 16 for (n,2n)
**Zap**: reaction product, e.g. zap=1 for neutron
**Eincs**: incident energy array
**mus**: angle cosine mesh
**Eouts**: outgoing energy array

# Double-differential distribution: Under the hat*



compute_ddxs

**MF4/MF5**
Legendre expansion
Tabulated data
Mixed
Analytical functions

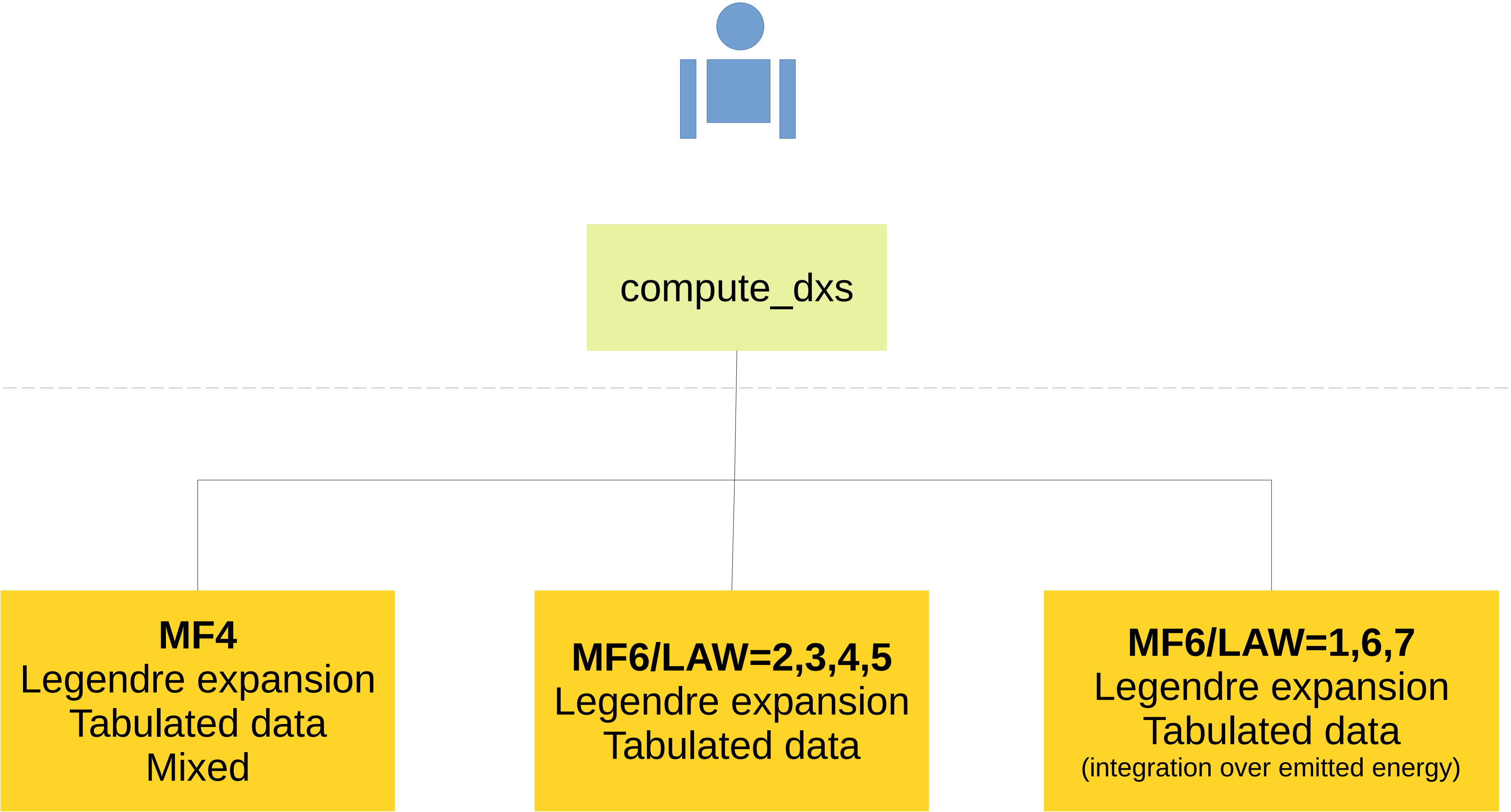
**MF6/LAW=1,6,7**
Legendre expansion
Tabulated data

**MF6/LAW=2,3,4,5**
Legendre expansion
Tabulated data

*Math core fully implemented (orange)
Relaying to MF4/MF5 and MF6/LAW=2,3,4,5 in progress

# Angular distributions: Under the hat*

(similar for energy spectra (MF5))



compute_dxs

**MF4**
Legendre expansion
Tabulated data
Mixed

**MF6/LAW=2,3,4,5**
Legendre expansion
Tabulated data

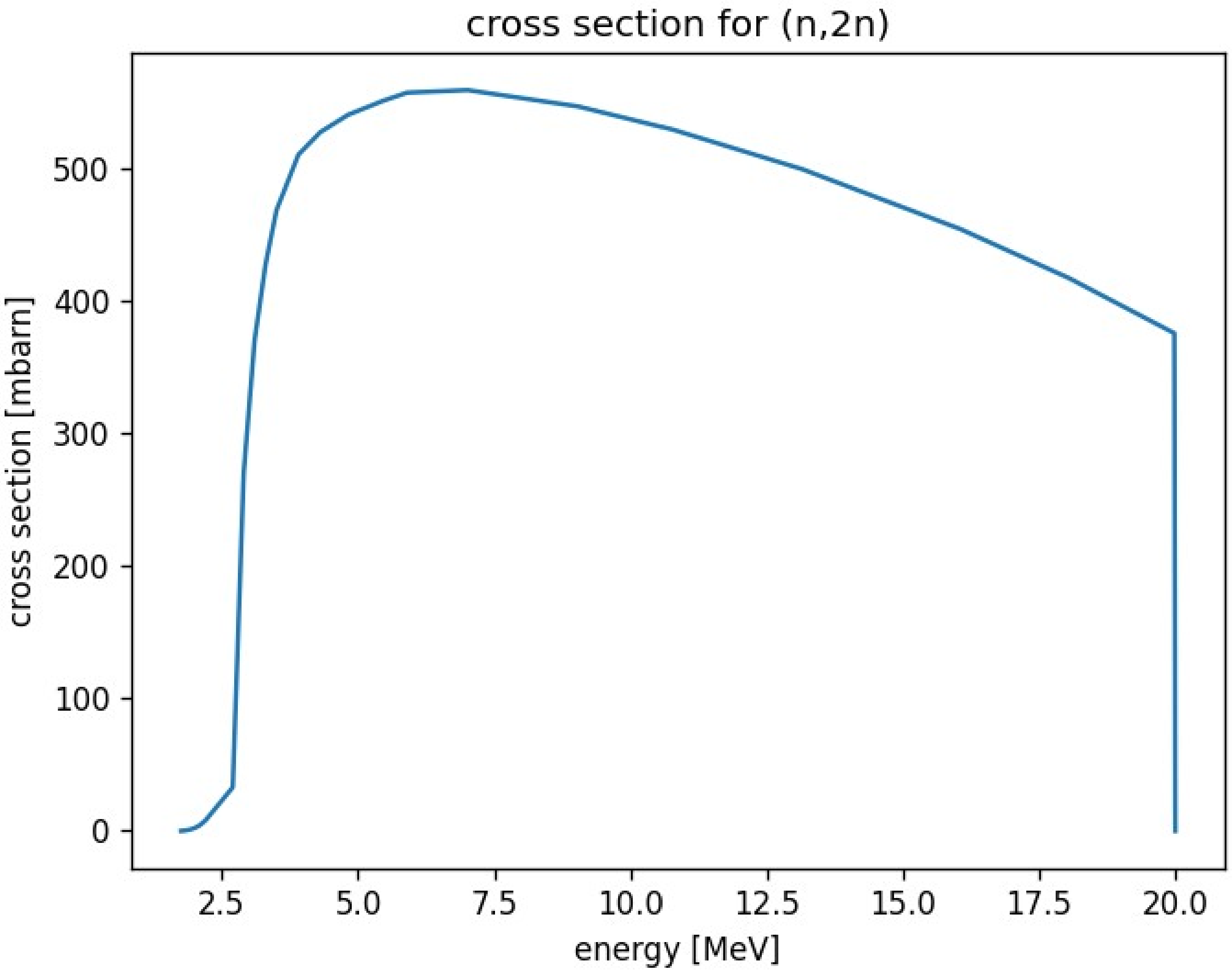
**MF6/LAW=1,6,7**
Legendre expansion
Tabulated data
(integration over emitted energy)

*Math core fully implemented (orange)
Relaying to MF6 in progress

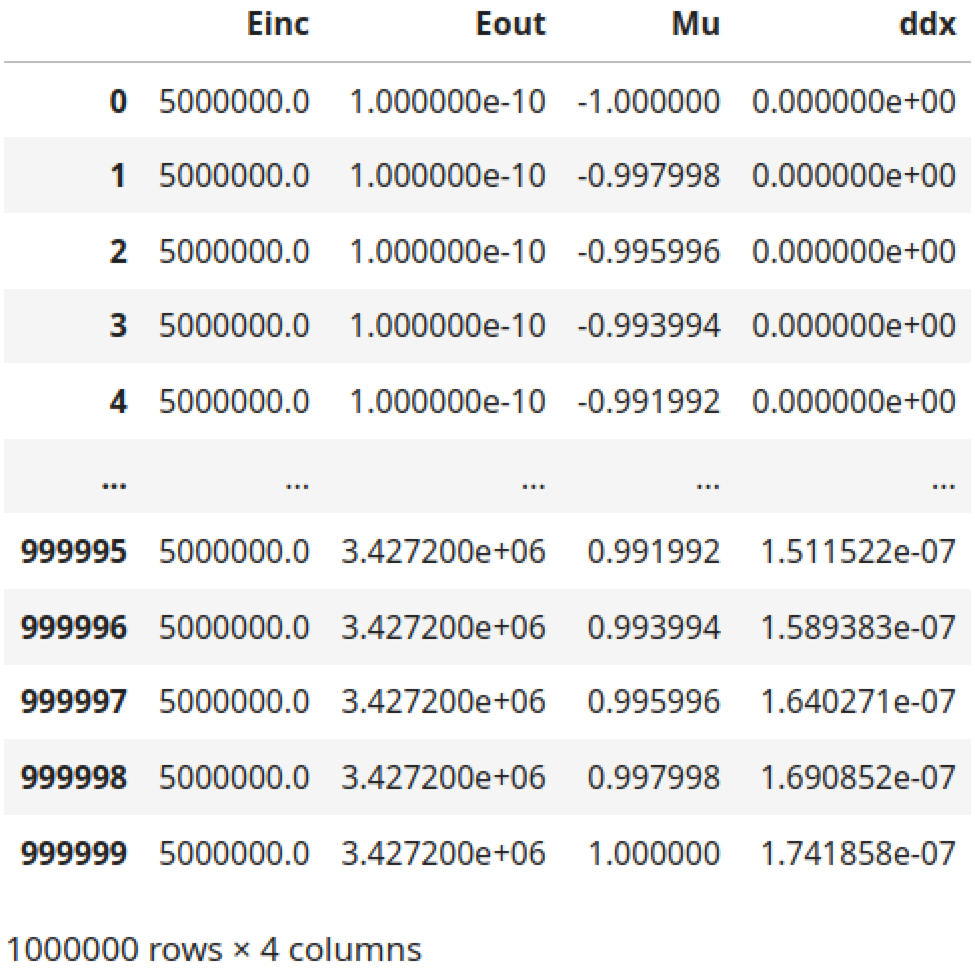
# FENDL 3.2b: Be-9(n,2n) DDX

LAW=7

# FENDL-3.2b: Be-9

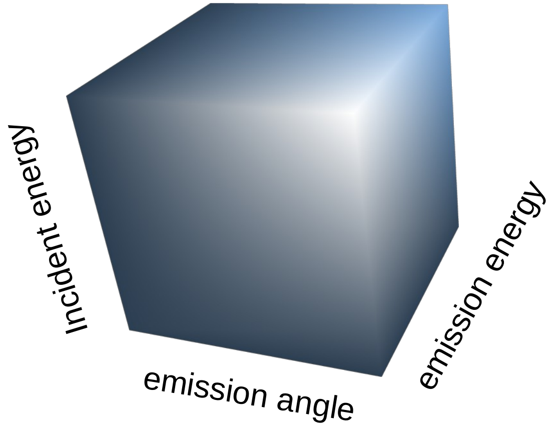


# FENDL-3.2b: Be-9 DDX

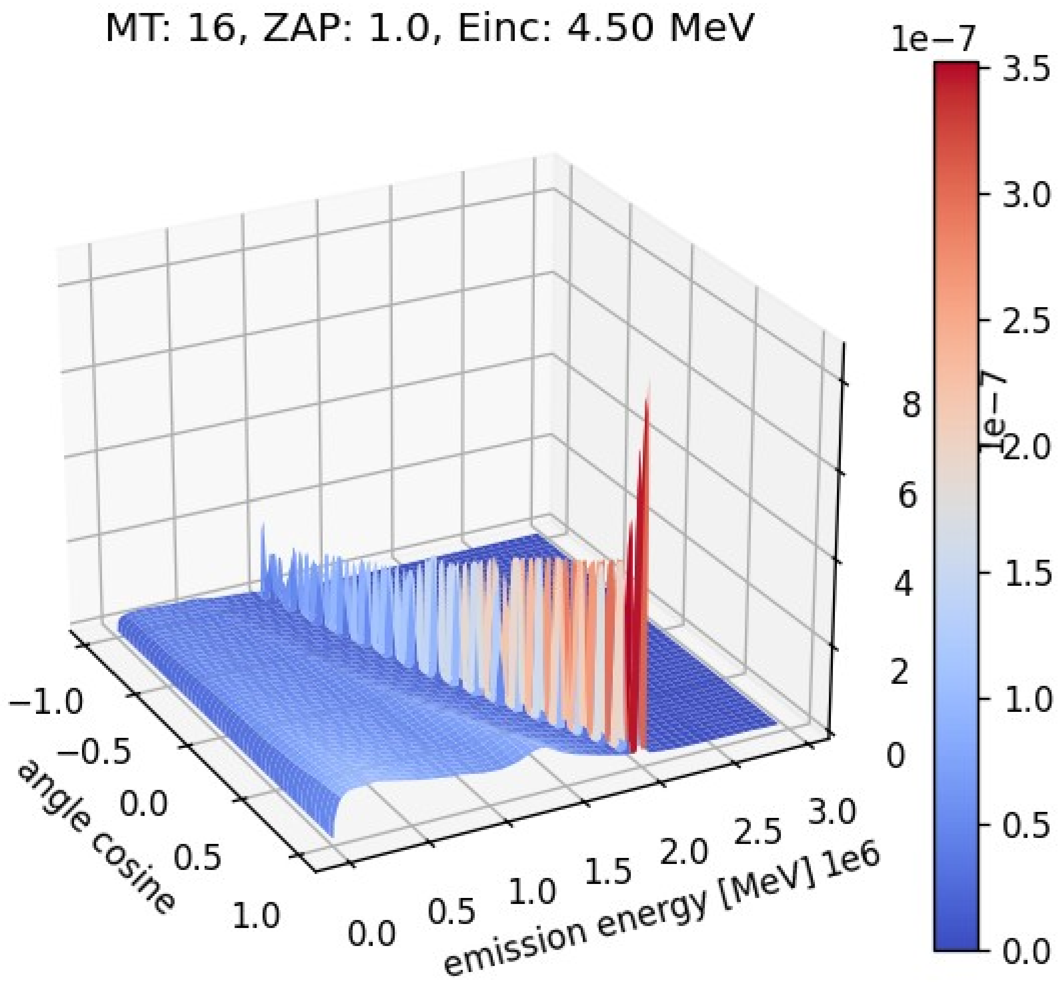


|  | Einc | Eout | Mu | ddx |
|---|---|---|---|---|
| **0** | 5000000.0 | 1.000000e-10 | -1.000000 | 0.000000e+00 |
| **1** | 5000000.0 | 1.000000e-10 | -0.997998 | 0.000000e+00 |
| **2** | 5000000.0 | 1.000000e-10 | -0.995996 | 0.000000e+00 |
| **3** | 5000000.0 | 1.000000e-10 | -0.993994 | 0.000000e+00 |
| **4** | 5000000.0 | 1.000000e-10 | -0.991992 | 0.000000e+00 |
| **...** | ... | ... | ... | ... |
| **999995** | 5000000.0 | 3.427200e+06 | 0.991992 | 1.511522e-07 |
| **999996** | 5000000.0 | 3.427200e+06 | 0.993994 | 1.589383e-07 |
| **999997** | 5000000.0 | 3.427200e+06 | 0.995996 | 1.640271e-07 |
| **999998** | 5000000.0 | 3.427200e+06 | 0.997998 | 1.690852e-07 |
| **999999** | 5000000.0 | 3.427200e+06 | 1.000000 | 1.741858e-07 |

1000000 rows × 4 columns

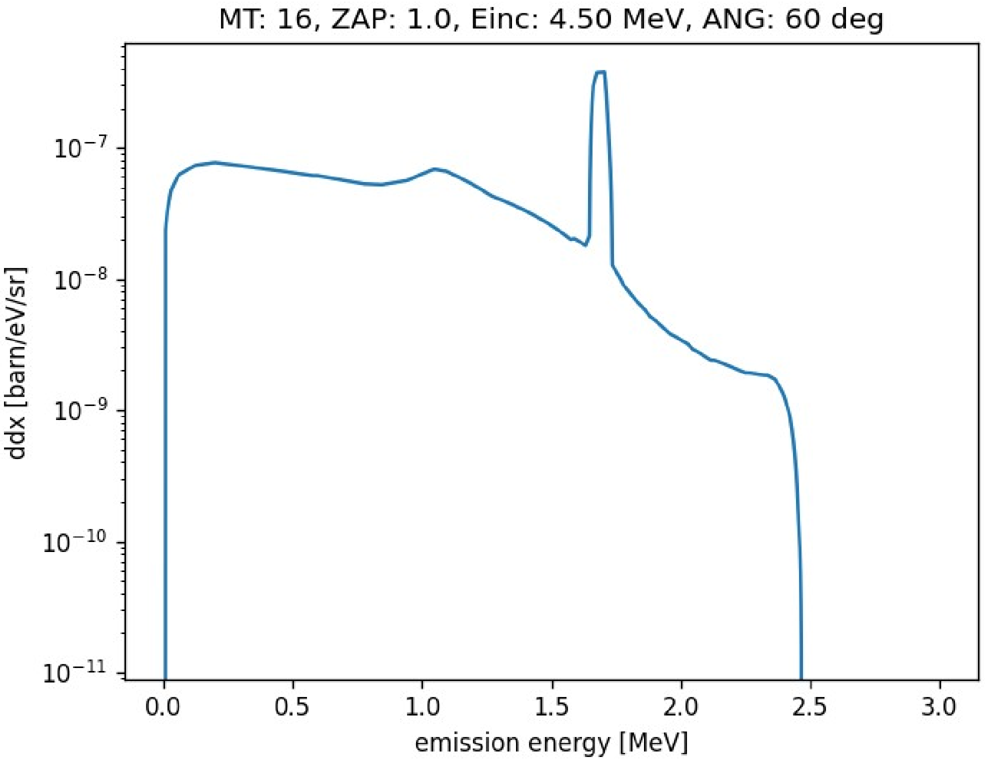


# FENDL-3.2b: Be-9 DDX

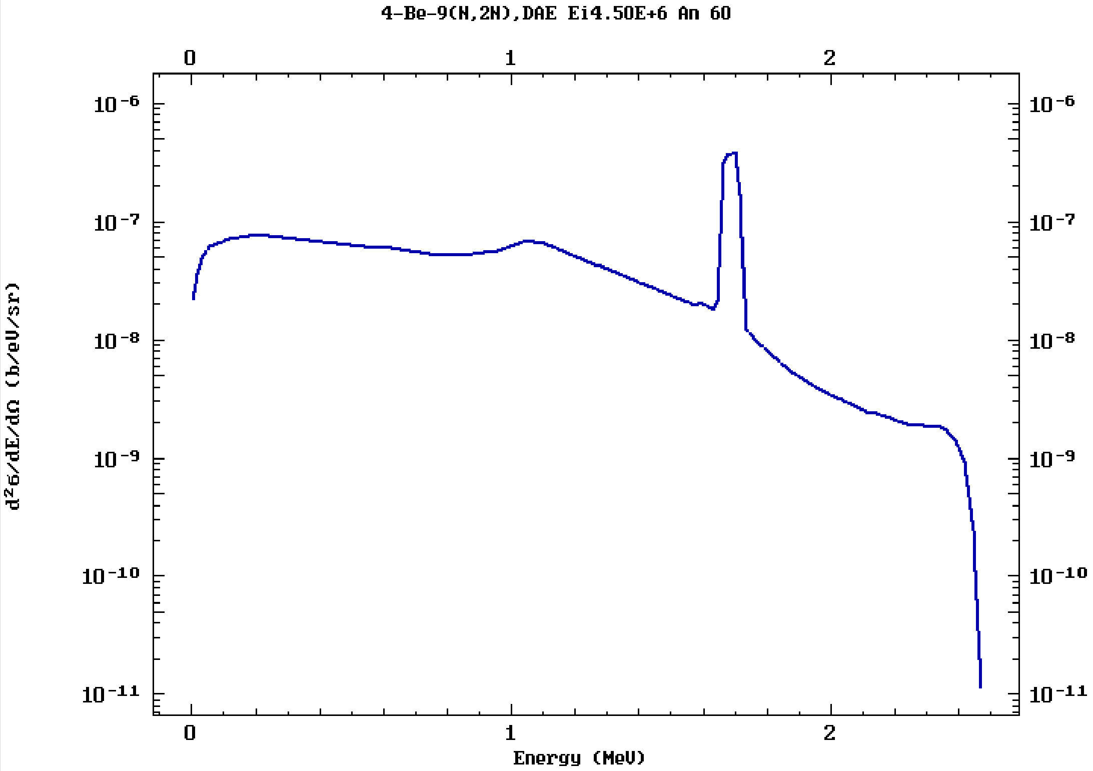


LAW=7

# Slice of 3-dim DDX (LAW=7)



endf-userpy

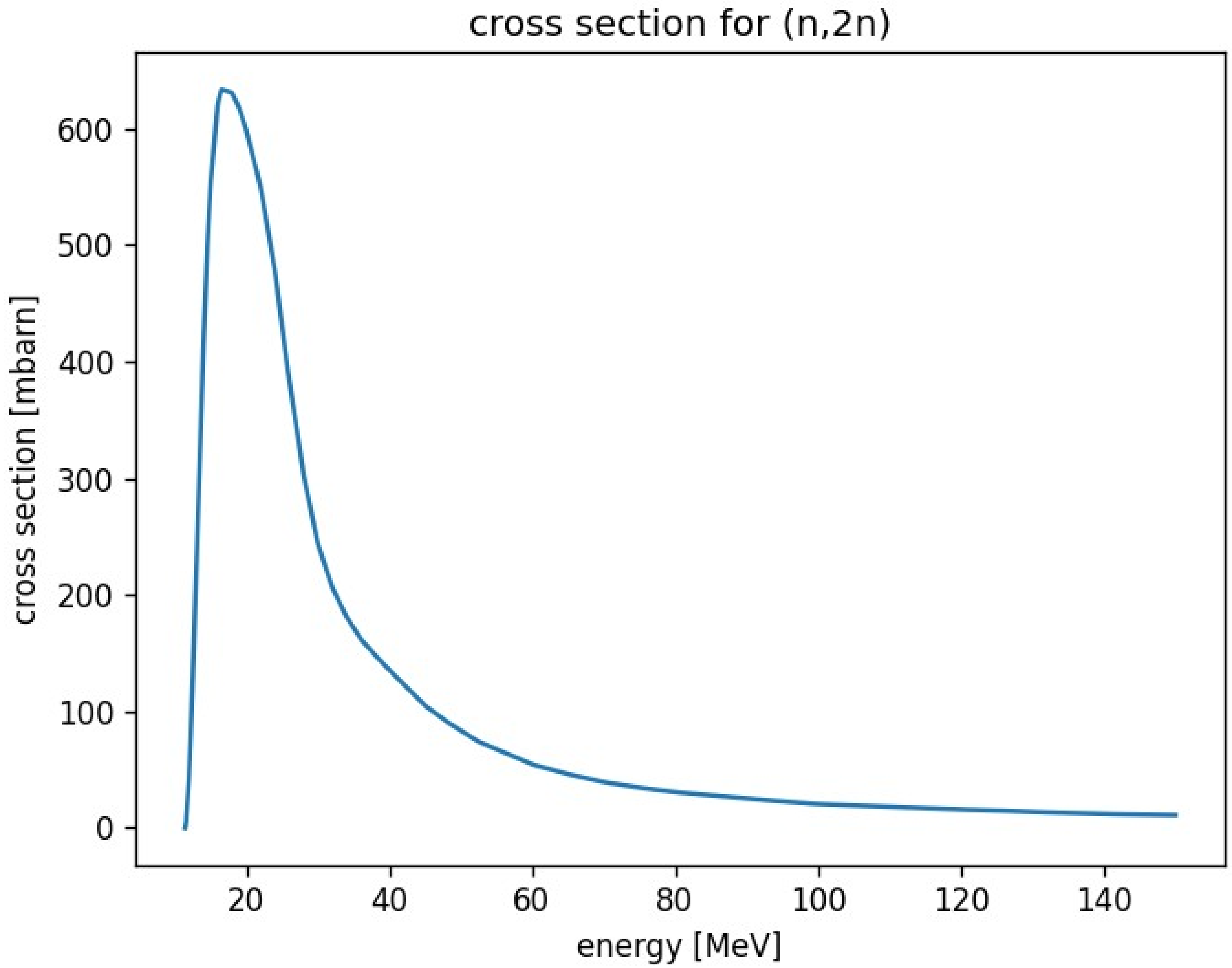


NDS ENDF web interface

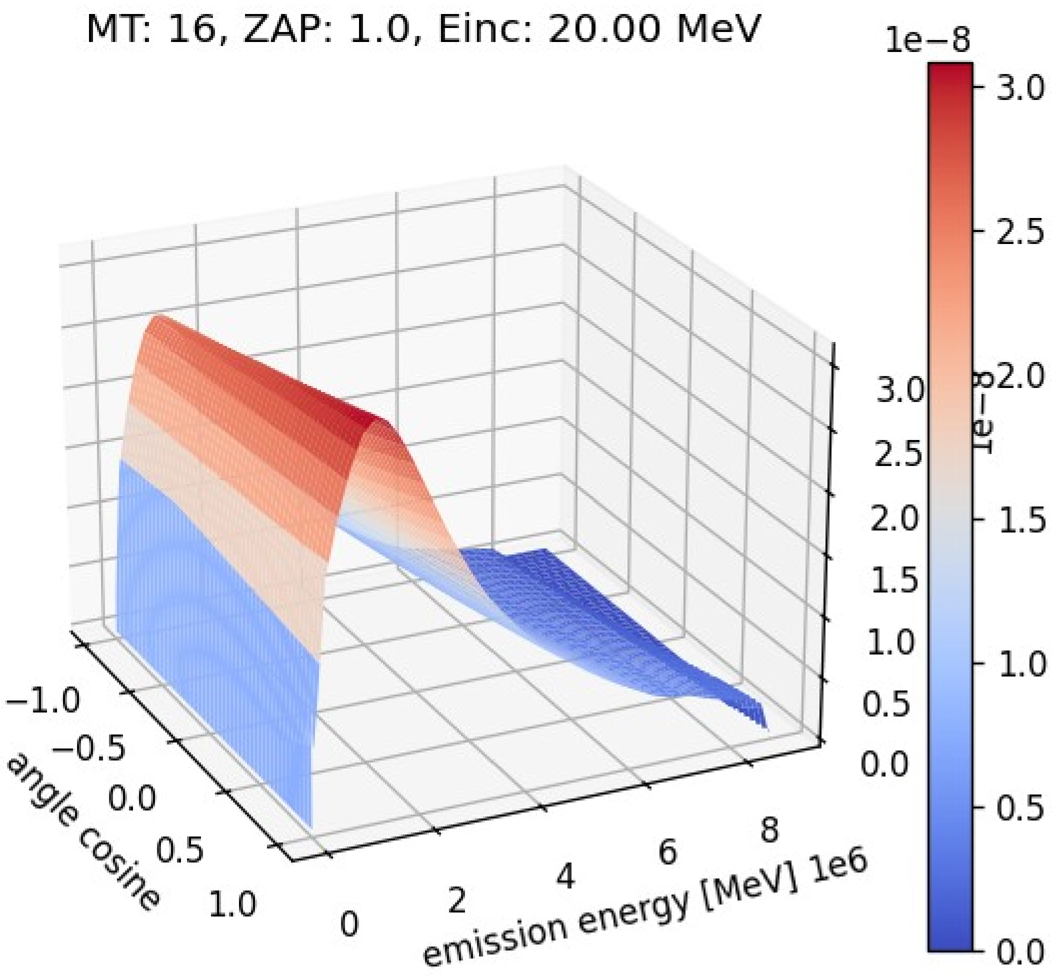
# FENDL 3.2b: Fe-56(n,2n) DDX

(LAW=1, LANG=1)

# FENDL-3.2b: Fe-56



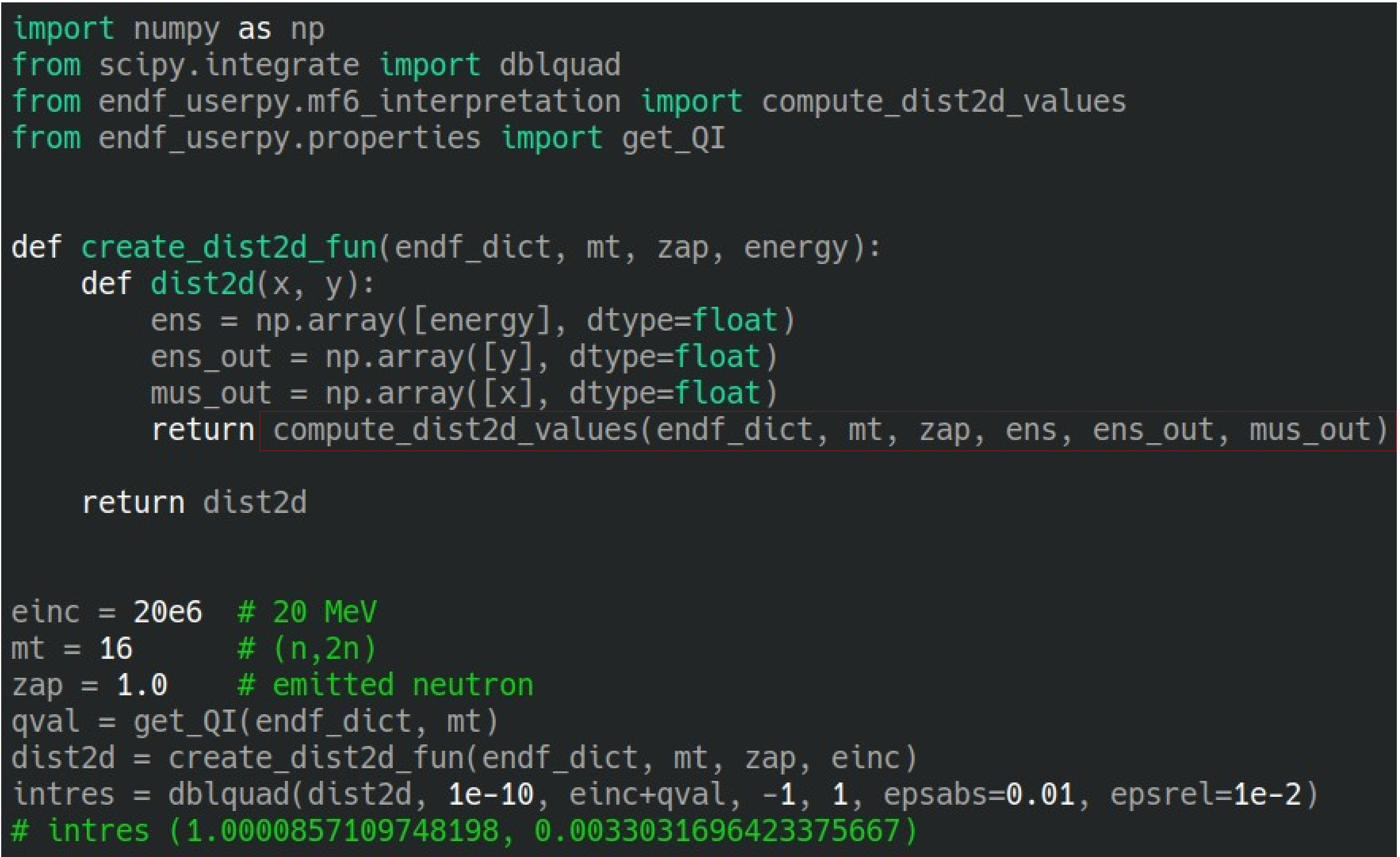
# FENDL-3.2b: Fe-56 DDX



$$\sigma_i(\mu, E, E') = \sigma(E)\ y_i(E)\ f_i(\mu, E, E')/2\pi$$

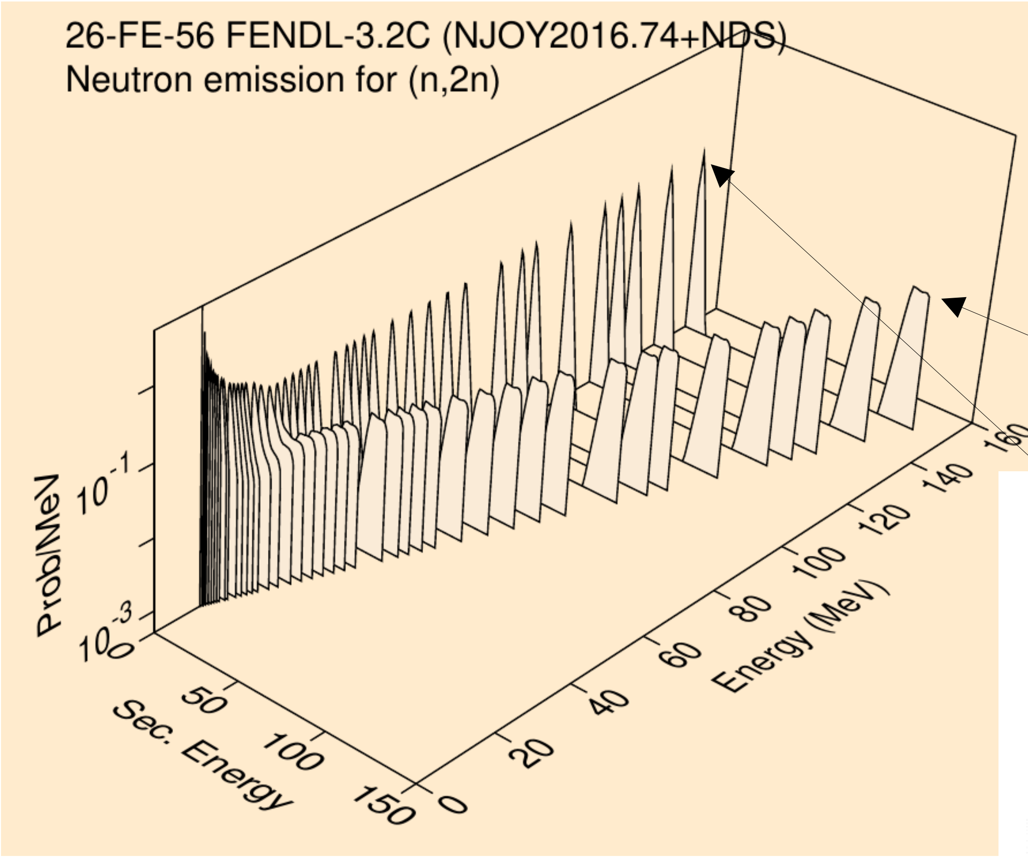
# FENDL-3.2b: Fe-56 (Normalization check)

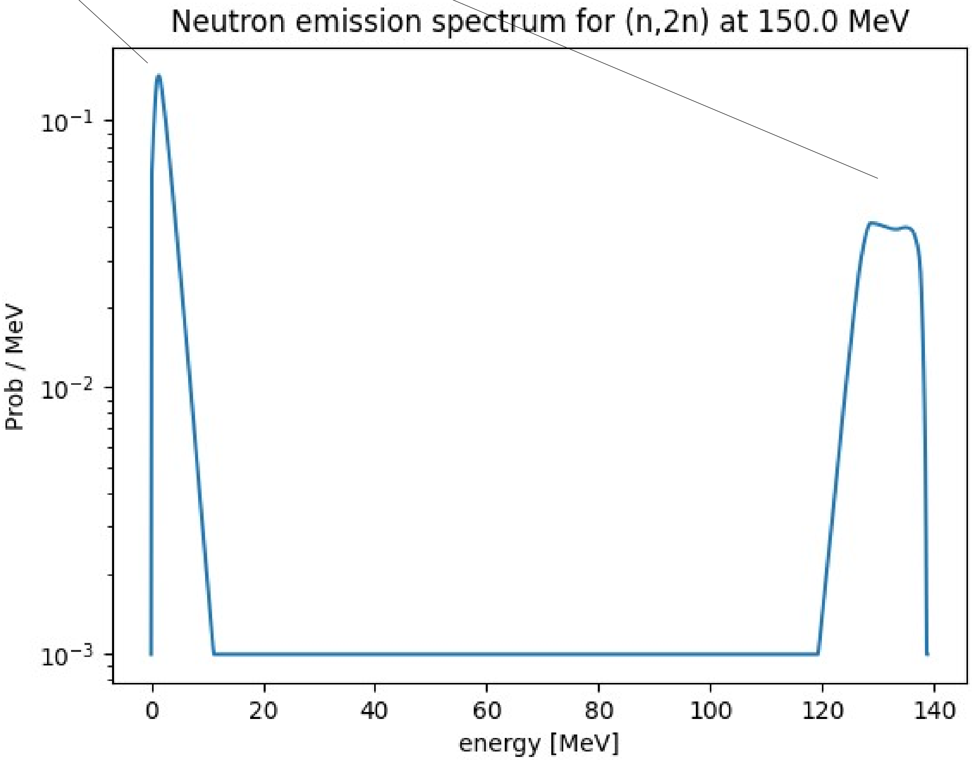
$$\int dE' \int d\mu \, \boxed{f_i(\mu, E, E')} = 1$$



```
import numpy as np
from scipy.integrate import dblquad
from endf_userpy.mf6_interpretation import compute_dist2d_values
from endf_userpy.properties import get_QI


def create_dist2d_fun(endf_dict, mt, zap, energy):
    def dist2d(x, y):
        ens = np.array([energy], dtype=float)
        ens_out = np.array([y], dtype=float)
        mus_out = np.array([x], dtype=float)
        return compute_dist2d_values(endf_dict, mt, zap, ens, ens_out, mus_out)

    return dist2d


einc = 20e6  # 20 MeV
mt = 16      # (n,2n)
zap = 1.0    # emitted neutron
qval = get_QI(endf_dict, mt)
dist2d = create_dist2d_fun(endf_dict, mt, zap, einc)
intres = dblquad(dist2d, 1e-10, einc+qval, -1, 1, epsabs=0.01, epsrel=1e-2)
# intres (1.0000857109748198, 0.0033031696423375667)
```

# FENDL-3.2b: Fe-56(n,2n) neutron emission spectrum



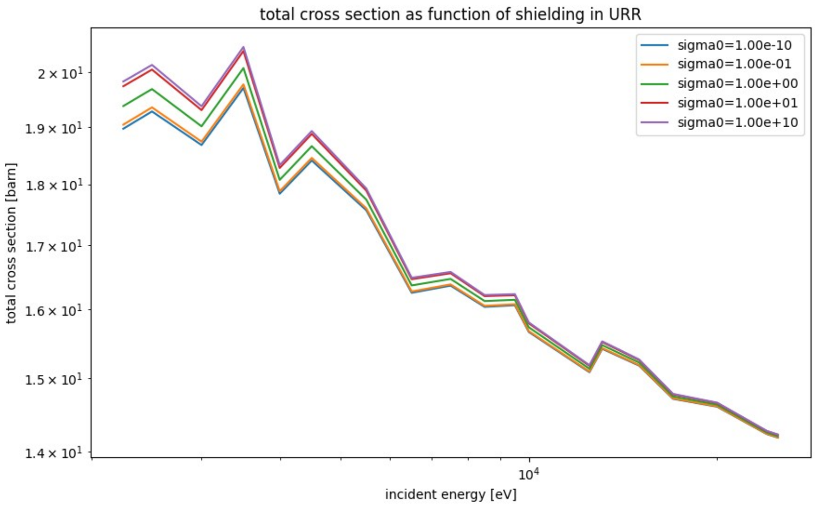


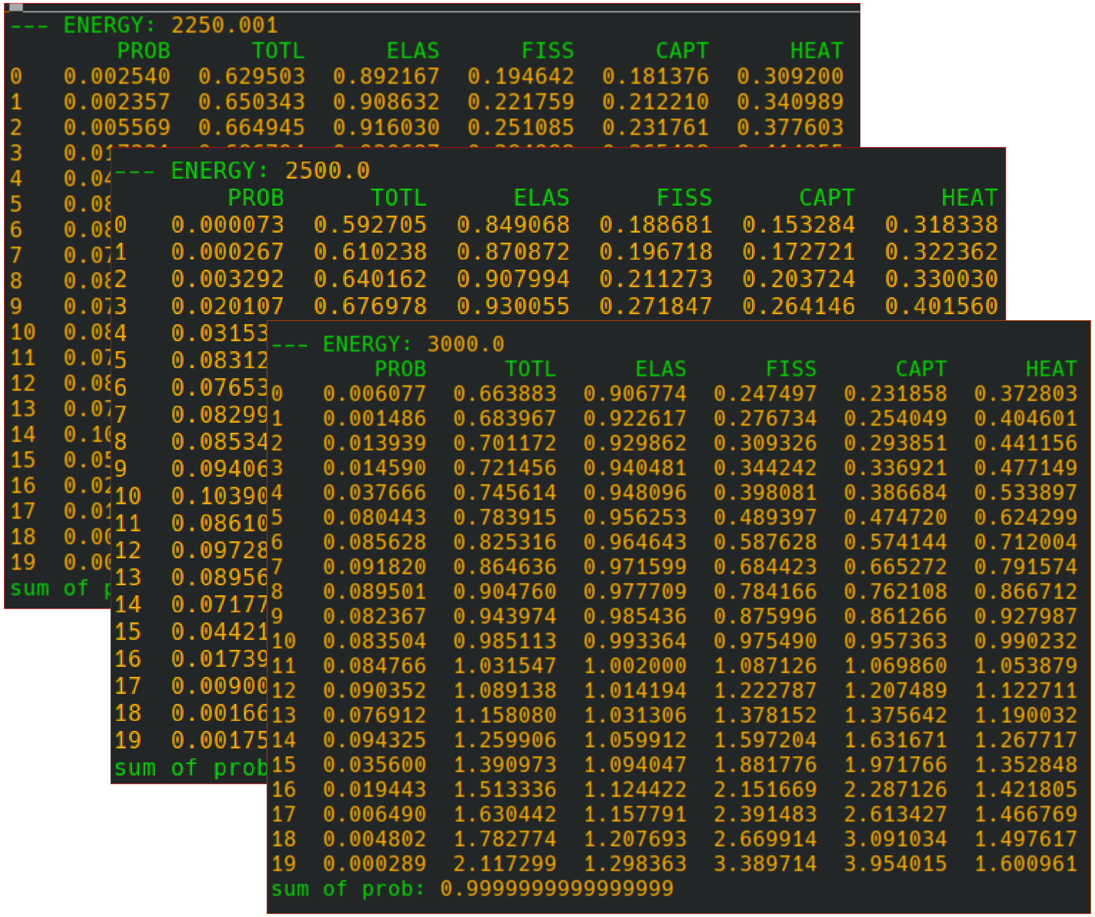
Obtained by integrating out angle

$$f_i(\mu, E, E')$$

# Self-shielded Cross Sections

# Self-shielded cross sections





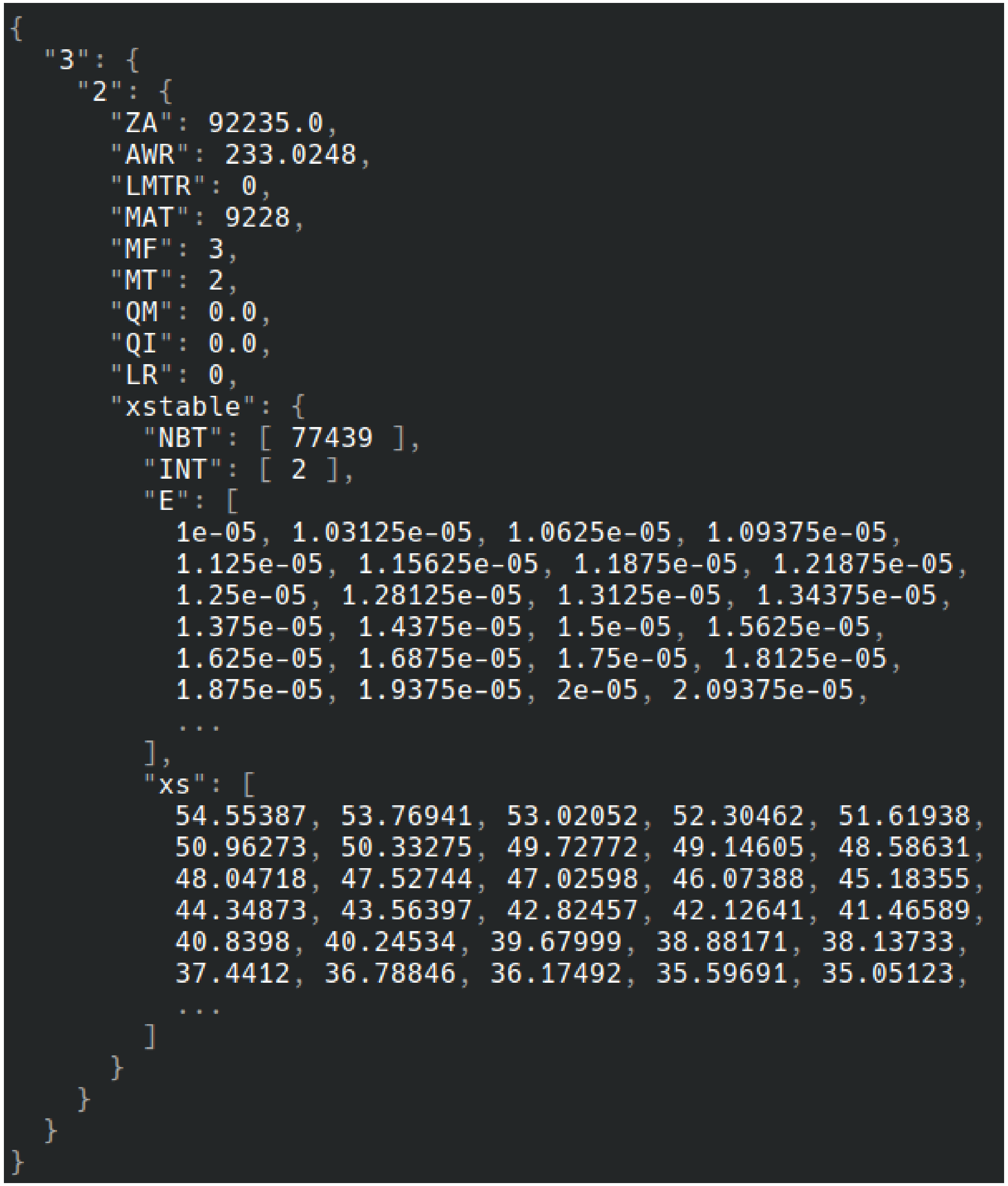
```
--- ENERGY: 2250.001
         PROB      TOTL      ELAS      FISS      CAPT      HEAT
0    0.002540  0.629503  0.892167  0.194642  0.181376  0.309200
1    0.002357  0.650343  0.908632  0.221759  0.212210  0.340989
2    0.005569  0.664945  0.916030  0.251085  0.231761  0.377603
3    0.01
4    0.04
5    0.08
6    0.08
7    0.07
8    0.08
9    0.07
10   0.08
11   0.07
12   0.08
13   0.07
14   0.10
15   0.05
16   0.02
17   0.01
18   0.00
19   0.00
sum of p
```

```
--- ENERGY: 2500.0
         PROB      TOTL      ELAS      FISS      CAPT      HEAT
0    0.000073  0.592705  0.849068  0.188681  0.153284  0.318338
1    0.000267  0.610238  0.870872  0.196718  0.172721  0.322362
2    0.003292  0.640162  0.907994  0.211273  0.203724  0.330030
3    0.020107  0.676978  0.930055  0.271847  0.264146  0.401560
4    0.03153
5    0.08312
6    0.07653
7    0.08299
8    0.08534
9    0.09406
10   0.10390
11   0.08610
12   0.09728
13   0.08956
14   0.07177
15   0.04421
16   0.01739
17   0.00900
18   0.00166
19   0.00175
sum of prob
```

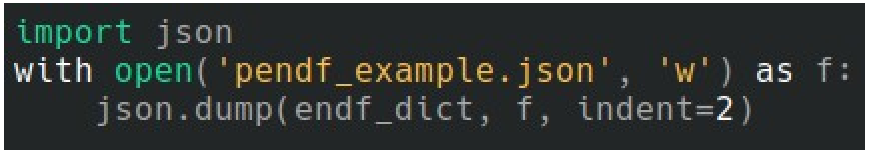
```
--- ENERGY: 3000.0
         PROB      TOTL      ELAS      FISS      CAPT      HEAT
0    0.006077  0.663883  0.906774  0.247497  0.231858  0.372803
1    0.001486  0.683967  0.922617  0.276734  0.254049  0.404601
2    0.013939  0.701172  0.929862  0.309326  0.293851  0.441156
3    0.014590  0.721456  0.940481  0.344242  0.336921  0.477149
4    0.037666  0.745614  0.948096  0.398081  0.386684  0.533897
5    0.080443  0.783915  0.956253  0.489397  0.474720  0.624299
6    0.085628  0.825316  0.964643  0.587628  0.574144  0.712004
7    0.091820  0.864636  0.971599  0.684423  0.665272  0.791574
8    0.089501  0.904760  0.977709  0.784166  0.762108  0.866712
9    0.082367  0.943974  0.985436  0.875996  0.861266  0.927987
10   0.083504  0.985113  0.993364  0.975490  0.957363  0.990232
11   0.084766  1.031547  1.002000  1.087126  1.069860  1.053879
12   0.090352  1.089138  1.014194  1.222787  1.207489  1.122711
13   0.076912  1.158080  1.031306  1.378152  1.375642  1.190032
14   0.094325  1.259906  1.059912  1.597204  1.631671  1.267717
15   0.035600  1.390973  1.094047  1.881776  1.971766  1.352848
16   0.019443  1.513336  1.124422  2.151669  2.287126  1.421805
17   0.006490  1.630442  1.157791  2.391483  2.613427  1.466769
18   0.004802  1.782774  1.207693  2.669914  3.091034  1.497617
19   0.000289  2.117299  1.298363  3.389714  3.954015  1.600961
sum of prob: 0.9999999999999999
```

# ENDF Format Conversions

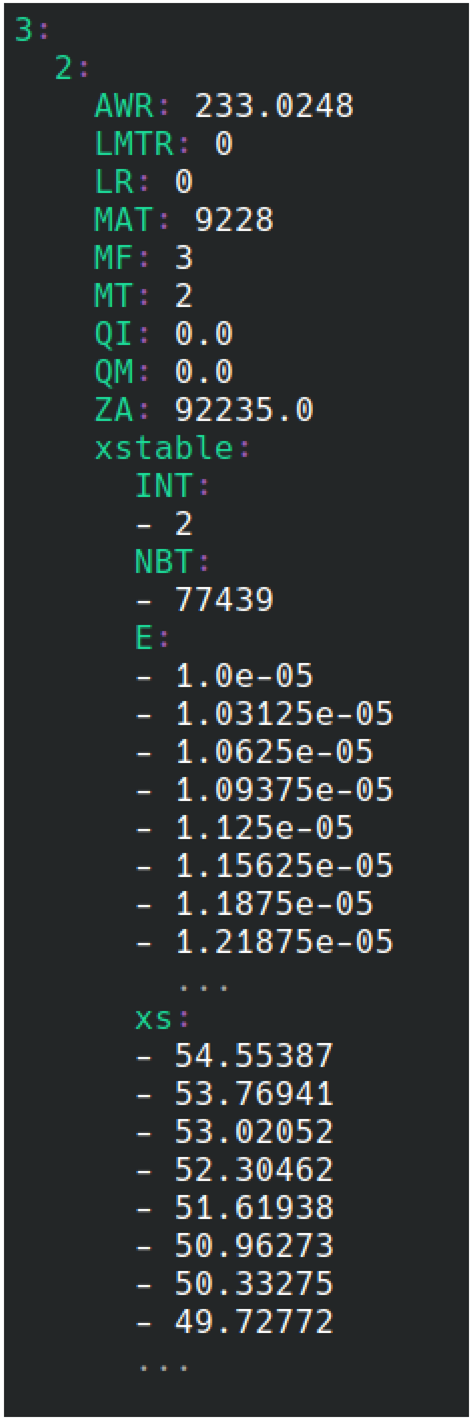
# ENDF to JSON or YAML file



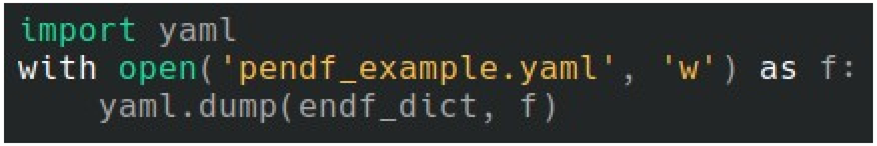
```
{
  "3": {
    "2": {
      "ZA": 92235.0,
      "AWR": 233.0248,
      "LMTR": 0,
      "MAT": 9228,
      "MF": 3,
      "MT": 2,
      "QM": 0.0,
      "QI": 0.0,
      "LR": 0,
      "xstable": {
        "NBT": [ 77439 ],
        "INT": [ 2 ],
        "E": [
          1e-05, 1.03125e-05, 1.0625e-05, 1.09375e-05,
          1.125e-05, 1.15625e-05, 1.1875e-05, 1.21875e-05,
          1.25e-05, 1.28125e-05, 1.3125e-05, 1.34375e-05,
          1.375e-05, 1.4375e-05, 1.5e-05, 1.5625e-05,
          1.625e-05, 1.6875e-05, 1.75e-05, 1.8125e-05,
          1.875e-05, 1.9375e-05, 2e-05, 2.09375e-05,
          ...
        ],
        "xs": [
          54.55387, 53.76941, 53.02052, 52.30462, 51.61938,
          50.96273, 50.33275, 49.72772, 49.14605, 48.58631,
          48.04718, 47.52744, 47.02598, 46.07388, 45.18355,
          44.34873, 43.56397, 42.82457, 42.12641, 41.46589,
          40.8398, 40.24534, 39.67999, 38.88171, 38.13733,
          37.4412, 36.78846, 36.17492, 35.59691, 35.05123,
          ...
        ]
      }
    }
  }
}
```



```python
import json
with open('pendf_example.json', 'w') as f:
    json.dump(endf_dict, f, indent=2)
```



```yaml
3:
  2:
    AWR: 233.0248
    LMTR: 0
    LR: 0
    MAT: 9228
    MF: 3
    MT: 2
    QI: 0.0
    QM: 0.0
    ZA: 92235.0
    xstable:
      INT:
      - 2
      NBT:
      - 77439
      E:
      - 1.0e-05
      - 1.03125e-05
      - 1.0625e-05
      - 1.09375e-05
      - 1.125e-05
      - 1.15625e-05
      - 1.1875e-05
      - 1.21875e-05
        ...
      xs:
      - 54.55387
      - 53.76941
      - 53.02052
      - 52.30462
      - 51.61938
      - 50.96273
      - 50.33275
      - 49.72772
      ...
```



```python
import yaml
with open('pendf_example.yaml', 'w') as f:
    yaml.dump(endf_dict, f)
```

# Future work: What’s next?

**Fortran level:**

- Linearization and integration routines
- Relativistic conversion from LAB to CM and vice-versa
- Photon distributions (MF12, MF13, MF14, MF15)
- Covariance data interpretation (MF31, MF32, MF33, MF34, MF35)
- Resonance reconstruction from resonance parameters (MF2)
- Thermal scattering cross section reconstruction
- Photo-atomic and atomic data (MF26, MF27, MF28)
- Making things faster

**Python level:**

- Extend capabilities of command line interface
- High-level routines for standard use cases
- Validation exercise: Compare with EXFOR
- Documentation!