Berkeley Nuclear Database Projects: (n, γ)

Aaron M. Hurst
amhurst@berkeley.edu

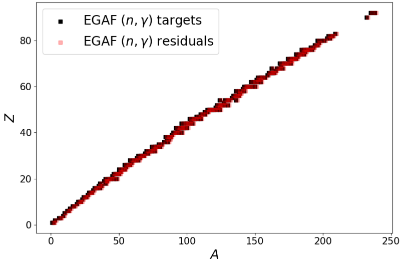
Technical Meeting on Nuclear Data Retrieval, Dissemination, and Data Portals
IAEA Headquarters, Vienna, Austria

November 11 – 15, 2024

Table of Contents

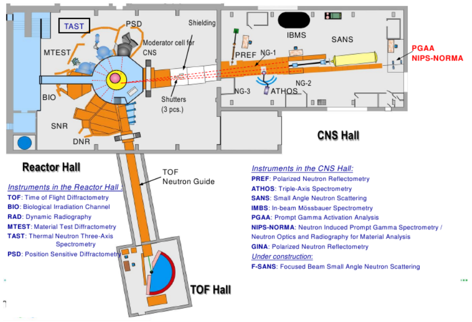
https://nucleardata.berkeley.edu/databases/

Evaluated Gamma-ray Activation File (EGAF)



- IAEA CRP initiative led by **Rick Firestone** (LBNL/UCB) and **Zsolt Révay** (BRR/FRM-II).
- Partial thermal neutron-capture γ-ray cross-section data measured at BRR.
- Natural targets $Z = 1 - 83, 90, 92$ except for Tc ($Z = 43$) and Pm $Z = 61$ (245 data sets).
- $\sim 38,000$ γ rays associated with $\sim 12,500$ levels.

PGAA @ Budapest Research Reactor



Instruments in the Reactor Hall :
TOF: Time of Flight Diffractometry
BIO: Biological Irradiation Channel
RAD: Dynamic Radiography
MTEST: Material Test Diffractometry
TAST: Thermal Neutron Three-Axis Spectrometry
PSD: Position Sensitive Diffractometry

Instruments in the CNS Hall:
PREF: Polarized Neutron Reflectometry
ATHOS: Triple-Axis Spectrometry
SANS: Small Angle Neutron Scattering
IMBS: In-beam Mössbauer Spectrometry
PGAA: Prompt Gamma Activation Analysis
NIPS-NORMA: Neutron Induced Prompt Gamma Spectrometry / Neutron Optics and Radiography for Material Analysis
GINA: Polarized Neutron Reflectometry
Under construction:
F-SANS: Focused Beam Small Angle Neutron Scattering

- Direct quantities:

  $E_\gamma,\ \sigma_\gamma$

- Derived quantities:

  $E_L,\ S_n,\ \sigma_0$

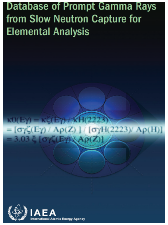
- Compiled quantities:

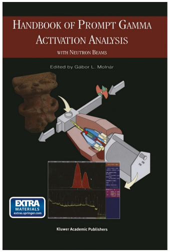
  $J^\pi,\ \alpha,\ \delta_\gamma,\ \lambda L$

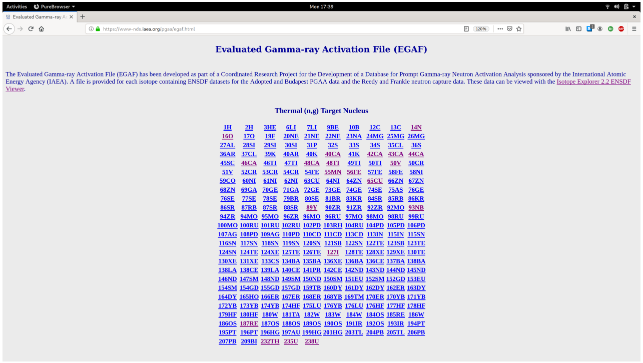
All (n, γ) measurements performed in a consistent manner using the same experimental configuration

EGAF disseminated in printed form and through the IAEA

https://www-nds.iaea.org/pgaa/egaf.html


Database of Prompt Gamma Rays from Slow Neutron Capture for Elemental Analysis
IAEA
International Atomic Energy Agency


Handbook of Prompt Gamma Activation Analysis
with Neutron Beams
Edited by Gábor L. Molnár
EXTRA MATERIALS extras.springer.com
Kluwer Academic Publishers



Evaluated Gamma-ray Activation File (EGAF)

The Evaluated Gamma-ray Activation File (EGAF) has been developed as part of a Coordinated Research Project for the Development of a Database for Prompt Gamma-ray Neutron Activation Analysis sponsored by the International Atomic Energy Agency (IAEA). A file is provided for each isotope containing ENSDF datasets for the Adopted and Budapest PGAA data and the Reedy and Frankle neutron capture data. These data can be viewed with the Isotope Explorer 2.2 ENSDF Viewer.

Thermal (n,g) Target Nucleus

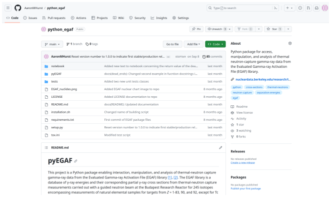
1H	2H	3HE	6LI	7LI	9BE	10B	12C	13C	14N
16O	17O	19F	20NE	21NE	22NE	23NA	24MG	25MG	26MG
27AL	28SI	29SI	30SI	31P	32S	33S	34S	35CL	36S
36AR	37CL	39K	40AR	40K	40CA	41K	42CA	43CA	44CA
45SC	46CA	46TI	47TI	48CA	48TI	49TI	50TI	50V	50CR
51V	52CR	53CR	54CR	54FE	55MN	56FE	57FE	58FE	58NI
59CO	60NI	61NI	62NI	63CU	64NI	64ZN	65CU	66ZN	67ZN
68ZN	69GA	70GE	71GA	72GE	73GE	74GE	74SE	75AS	76GE
76SE	77SE	78SE	79BR	80SE	81BR	83KR	84SR	85RB	86KR
86SR	87RB	87SR	88SR	89Y	90ZR	91ZR	92ZR	92MO	93NB
94ZR	94MO	95MO	96ZR	96MO	96RU	97MO	98MO	98RU	99RU
100MO	100RU	101RU	102RU	102PD	103RH	104RU	104PD	105PD	106PD
107AG	108PD	109AG	110PD	110CD	111CD	113CD	113IN	115IN	115SN
116SN	117SN	118SN	119SN	120SN	121SB	122SN	122TE	123SB	123TE
124SN	124TE	124XE	125TE	126TE	127I	128TE	128XE	129XE	130TE
130XE	131XE	133CS	134BA	135BA	136XE	136BA	136CE	137BA	138BA
138LA	138CE	139LA	140CE	141PR	142CE	142ND	143ND	144ND	145ND
146ND	147SM	148ND	149SM	150ND	150SM	151EU	152SM	152GD	153EU
154SM	154GD	155GD	157GD	159TB	160DY	161DY	162DY	162ER	163DY
164DY	165HO	166ER	167ER	168ER	168YB	169TM	170ER	170YB	171YB
172YB	173YB	174YB	174HF	175LU	176YB	176LU	176HF	177HF	178HF
179HF	180HF	180W	181TA	182W	183W	184W	184OS	185RE	186W
186OS	187RE	187OS	188OS	189OS	190OS	191IR	192OS	193IR	194PT
195PT	196PT	196HG	197AU	199HG	201HG	203TL	204PB	205TL	206PB
207PB	209BI	232TH	235U	238U					

Source ENSDF-formatted datasets @ IAEA

Open-source Python library pyEGAF on GitHub

https://github.com/AaronMHurst/python_egaf

- Translated all 245 ENSDF-formatted EGAF datasets to a new `JSON` format.
- Generated RIPL-format EGAF for reaction calculations.
- Developed suite of Python modules enabling interaction, analysis, and visualization of the EGAF (n, γ) data.
- Docstrings provided for all methods.
- JSON schema keys documented extensively in `README`.
- 224 unit tests (multiple virtual `Python3` environments).
- Installation, testing scripts, and `Jupyter Notebooks` provided.
- `ENSDF`, `RIPL`, and `JSON` files bundled with software.
- Over 900 downloads.



AaronMHurst Reset version number to 1.0.0 to indicate first stable/production rel... 050f0d4 on Sep 8 85 commits

notebook	Added new text to notebook concerning the return value of the dea...	last month
pyEGAF	docs(dead_ends): Changed second example in fucntion docstrings i...	last month
tests	Added two new unit tests classes	last month
EGAF_nuclides.png	Added EGAF nuclear chart image to repo	8 months ago
LICENSE	Added LICENSE documentation to repo	8 months ago
README.md	docs(README): Updated documentation	last month
installation.sh	Changed name of building script	8 months ago
requirements.txt	First commit of EGAF package files	8 months ago
setup.py	Reset version number to 1.0.0 to indicate first stable/production rel...	last month
tox.ini	Modified test script	last month

README.md

pyEGAF

This project is a Python package enabling interaction, manipulation, and analysis of thermal-neutron capture gamma-ray data from the Evaluated Gamma-ray Activation File (EGAF) library [1], [2]. The EGAF library is a database of γ-ray energies and their corresponding partial γ-ray cross sections from thermal-neutron capture measurements carried out with a guided neutron beam at the Budapest Research Reactor for 245 isotopes encompassing measurements of natural elemental samples for targets from Z = 1-83, 90, and 92, except for Tc

About

Python package for access, manipulation, and analysis of thermal neutron-capture gamma-ray data from the Evaluated Gamma-ray Activation File (EGAF) library.

nucleardata.berkeley.edu/research/t...

python cross-sections thermal-neutrons neutron-capture separation-energies egaf

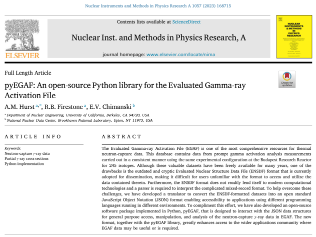
Readme
View license
Activity
1 star
3 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package



Nuclear Instruments and Methods in Physics Research A 1057 (2023) 168715

Contents lists available at ScienceDirect

Nuclear Inst. and Methods in Physics Research, A

journal homepage: www.elsevier.com/locate/nima

Full Length Article

pyEGAF: An open-source Python library for the Evaluated Gamma-ray Activation File

A.M. Hurst[a,*], R.B. Firestone[a], E.V. Chimanski[b]

[a] Department of Nuclear Engineering, University of California, Berkeley, CA 94720, USA
[b] National Nuclear Data Center, Brookhaven National Laboratory, Upton, NY 11973, USA

ARTICLE INFO

Keywords:
Neutron-capture γ-ray data
Partial γ-ray cross sections
Python implementation

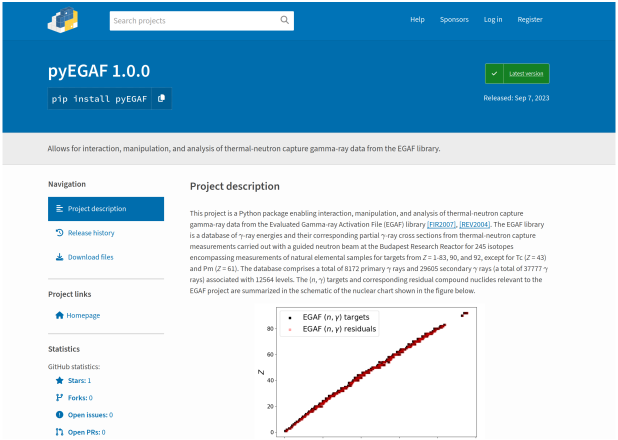
ABSTRACT

The Evaluated Gamma-ray Activation File (EGAF) is one of the most comprehensive resources for thermal neutron-capture data. This database contains data from prompt gamma activation analysis measurements carried out in a consistent manner using the same experimental configuration at the Budapest Research Reactor for 245 isotopes. Although these valuable datasets have been freely available for many years, one of the drawbacks is the outdated and cryptic Evaluated Nuclear Structure Data File (ENSDF) format that is currently adopted for dissemination, making it difficult for users unfamiliar with the format to access and utilize the data contained therein. Furthermore, the ENSDF format does not readily lend itself to modern computational technologies and a parser is required to interpret the complicated mixed-record format. To help overcome these challenges, we have developed a translator to convert the ENSDF-formatted datasets into an open standard JavaScript Object Notation (JSON) format enabling accessibility to applications using different programming languages running in different environments. To compliment this effort, we have also developed an open-source software package implemented in Python, pyEGAF, that is designed to interact with the JSON data structures for general purpose access, manipulation, and analysis of the neutron-capture γ-ray data in EGAF. The new format, together with the pyEGAF library, greatly enhances access to the wider applications community where EGAF data may be useful or is required.

```
git clone https://github.com/AaronMHurst/python_egaf.git
```

pyEGAF on the Python Package Index (PyPI) repository

https://pypi.org/project/pyEGAF/



pyEGAF 1.0.0

```
pip install pyEGAF
```

Released: Sep 7, 2023

Allows for interaction, manipulation, and analysis of thermal-neutron capture gamma-ray data from the EGAF library.

Statistics

GitHub statistics:

- Stars: 1
- Forks: 0
- Open issues: 0
- Open PRs: 0

Project description

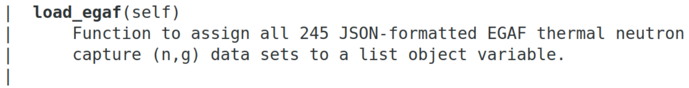
This project is a Python package enabling interaction, manipulation, and analysis of thermal-neutron capture gamma-ray data from the Evaluated Gamma-ray Activation File (EGAF) library [FIR2007], [REV2004]. The EGAF library is a database of γ-ray energies and their corresponding partial γ-ray cross sections from thermal-neutron capture measurements carried out with a guided neutron beam at the Budapest Research Reactor for 245 isotopes encompassing measurements of natural elemental samples for targets from $Z = 1$-83, 90, and 92, except for Tc ($Z = 43$) and Pm ($Z = 61$). The database comprises a total of 8172 primary γ rays and 29605 secondary γ rays (a total of 37777 γ rays) associated with 12564 levels. The (n, γ) targets and corresponding residual compound nuclides relevant to the EGAF project are summarized in the schematic of the nuclear chart shown in the figure below.

- pip install pyegaf
- pyEGAF 1.0.0: Production/stable development release.
- FreeBSD License.

Accessing the thermal (n, γ) data using pyEGAF

https://github.com/AaronMHurst/python_egaf

```
$ ipython
In [1]: import pyEGAF as egaf
In [2]: e = egaf.EGAF()
In [3]: edata = e.load_egaf() # EGAF data
```

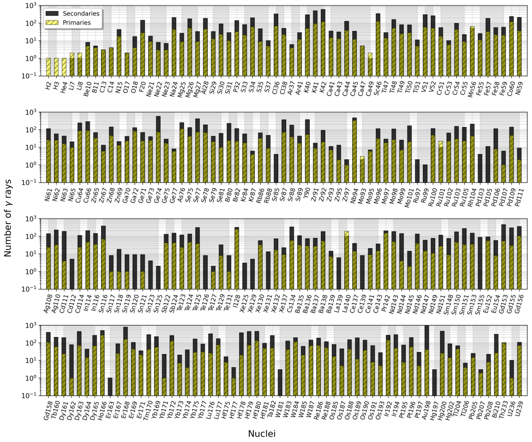


```
| load_egaf(self)
|     Function to assign all 245 JSON-formatted EGAF thermal neutron
|     capture (n,g) data sets to a list object variable.
|
```

- Run through Jupyter Notebooks from GitHub:
  - Manipulation of EGAF data: `grin_pyegaf.ipynb`
  - Statistical-modeling methods: `eval_28Si_thermal_capture.ipynb`

What data is in EGAF?

And what else do we need?



- Courtesy: *E.V. Chimanski, BNL*
- 8,172 primary γ rays.
- 29,605 secondary γ rays.
- 12,564 levels.
- Cross sections: σ_γ, σ_0
- Associated nuclear structure properties, e.g., J^π, I_γ . . .
- α, δ_γ, λL for improved RIPL and ENDF libraries.

ENSDF-formatted EGAF data sets

$^{28}\mathrm{Si}(n,\gamma)^{29}\mathrm{Si}$

```
29SI    28SI(N,G) E=THERMAL: {~EGAF}
29SI c  Evaluated Gamma-ray Activation File (EGAF).
29SI2c  Evaluated by R.B. Firestone (LBNL), December 2003.
29SI c  BR$|s{-0}=0.177 5 (1981MuZQ)
29SI cG RI$Elemental |s(|g) assuming %Abundance=92.2297  7
29SI  N  1.084249  8
29SI PN                                                                     C
29SI2PN Thermal cross section in barns.
29SI cN NR$Isotopic |s(|g)=NR*RI.
29SI2cN Divide by |s{-0} for intensity per neutron capture.
29SI  L       0.0   1/2+              STABLE
29SI  L  1273.379 17 3/2+             290 FS    10
29SI  G  1273.349 17  0.0289 6
```

- All γ-ray information needed is provided in EGAF:
  - `NR`, `RI`, and abundance.
  - Adopted σ_0.

- Lots of applications source thermal-capture data.
- Different applications/users may want different subsets of data.
- γ-ray energies and *intensities* certainly...
- I_γ: Partial elemental or isotopic cross sections; populations per neutron capture; relative intensities.
- Associated decay-scheme properties, e.g., initial and final levels (floats and/or indices); identification of *primary* and *secondary* γ rays...
- Difficult to capture all this information in *Evaluated Nuclear Structure Data File* (ENSDF).
- ENSDF is inconvenient to work with for the uninitiated and requires a parser.

Conversion of EGAF to JSON: $^{28}\mathrm{Si}(n,\gamma)^{29}\mathrm{Si}$

https://github.com/AaronMHurst/python_egaf

EGAF (JSON)

```
"neutronCaptureNormalization": [
    {
        "normalizationRecord": [
            {
                "multiplierIsotopicCorrection": 1.084249,
                "dMultiplierIsotopicCorrection": 8e-06,
                "naturalIsotopicAbundance": 92.22973689622955,
                "dNaturalIsotopicAbundance": 0.0006805059494358181,
                "adoptedTotalThermalCaptureCrossSection": 0.177,
                "dAdoptedTotalThermalCaptureCrossSection": 0.005,
                "unitAdoptedCrossSection": "b",
                "keyNumber": "1981MuZQ"
            }
        ]
    }
],
"levelScheme": [
    {
        "levelIndex": 0,
        "levelEnergy": 0.0,
        "dLevelEnergy": 0.0,
        "levelIsIsomer": false,
        "isomerDecay": [],
        "numberOfSpins": 1,
        "spins": [
            {
                "spinIndex": 0,
                "spinReal": 0.5,
                "spinIsTentative": false,
                "spinIsLimit": false,
                "spinLimits": null,
                "parity": 1,
                "paritySign": "positive",
                "parityIsTentative": false
            }
        ],
        "numberOfGammas": 0,
        "gammaDecay": []
    },
    {
        "levelIndex": 1,
        "levelEnergy": 1273.379,
        "dLevelEnergy": 0.017,
        "levelIsIsomer": true,
        "isomerDecay": [
            {
                "halfLifeBest": 290.0,
                "dHalfLifeBest": 10.0,
                "unitHalfLifeBest": "fs",
```

EGAF (ENSDF)

```
29SI    28SI(N,G) E=THERMAL: {~EGAF}
29SI c  Evaluated Gamma-ray Activation File (EGAF).
29SI2c  Evaluated by R.B. Firestone (LBNL), December 2003.
29SI c  BR$|s{-0}=0.177 5 (1981MuZQ)
29SI cG RI$Elemental |s(|g) assuming %Abundance=92.2297  7
29SI  N  1.084249  8
29SI PN                                                                     C
29SI2PN Thermal cross section in barns.
29SI cN NR$Isotopic |s(|g)=NR*RI.
29SI2cN Divide by |s{-0} for intensity per neutron capture.
29SI  L      0.0   1/2+                  STABLE
29SI  L  1273.379 17 3/2+                290 FS    10
29SI  G  1273.349 17  0.0289 6
```

- Partial schema illustrated for $^{28}\mathrm{Si}(n,\gamma)^{29}\mathrm{Si}$.
- Representative JSON translation for ENSDF-formatted EGAF datasets.
- All 245 EGAF datasets have been converted.

New JSON and RIPL format for EGAF

https://github.com/AaronMHurst/python_egaf

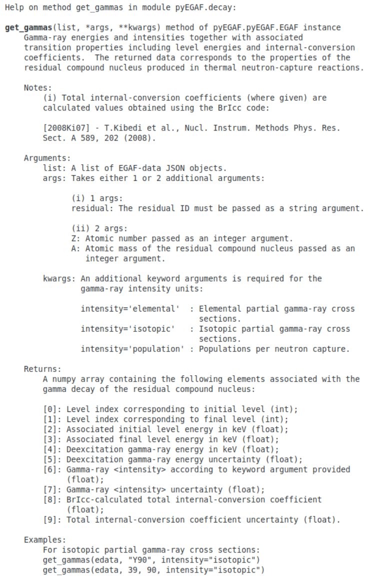
- New JSON format overcomes space-limited *cryptic* ENSDF format.
- Include more complete information in an intuitive manner.
- For example, separation energies in the ^{29}Si residual:

```
"energyNeutronSeparationAME2020": 8473.6025,
"energyNeutronSeparationEGAF": 8473.537,
"energyProtonSeparationAME2020": 12333.3331,
```

- Developed software package pyEGAF for interacting with and manipulating the EGAF JSON data sets.
- Reaction and statistical-model codes (e.g., DICEBOX, CoH, TALYS) often require (n,γ) data in *Reference Input Parameter Library* (RIPL) format.
- JSON and RIPL data sets generated for corresponding EGAF files (245 total).

```
29Si    29    14    14    46    13     8     8.473603    12.333333
  1    0.000000    0.5   1   -1.00E+00   0 0                        1/2+   0
  2    1.273379    1.5   1    2.90E-13   1 0                        3/2+   0
                                                 1      1.273  1.000E+00  1.000E+00  0.000E+00
  3    2.028060    2.5   1    3.06E-13   2 0                        5/2+   0
                                                 2      0.755  6.367E-02  6.367E-02  0.000E+00
                                                 1      2.028  9.363E-01  9.363E-01  0.000E+00
```

Docstrings methods: `help(e.get_gammas)` method



```
Help on method get_gammas in module pyEGAF.decay:

get_gammas(list, *args, **kwargs) method of pyEGAF.pyEGAF.EGAF instance
    Gamma-ray energies and intensities together with associated
    transition properties including level energies and internal-conversion
    coefficients.  The returned data corresponds to the properties of the
    residual compound nucleus produced in thermal neutron-capture reactions.

    Notes:
        (i) Total internal-conversion coefficients (where given) are
        calculated values obtained using the BrIcc code:

        [2008Ki07] - T.Kibedi et al., Nucl. Instrum. Methods Phys. Res.
        Sect. A 589, 202 (2008).

    Arguments:
        list: A list of EGAF-data JSON objects.
        args: Takes either 1 or 2 additional arguments:

              (i) 1 args:
              residual: The residual ID must be passed as a string argument.

              (ii) 2 args:
              Z: Atomic number passed as an integer argument.
              A: Atomic mass of the residual compound nucleus passed as an
                 integer argument.

        kwargs: An additional keyword arguments is required for the
                gamma-ray intensity units:

                intensity='elemental'  : Elemental partial gamma-ray cross
                                         sections.
                intensity='isotopic'   : Isotopic partial gamma-ray cross
                                         sections.
                intensity='population' : Populations per neutron capture.

    Returns:
        A numpy array containing the following elements associated with the
        gamma decay of the residual compound nucleus:

        [0]: Level index corresponding to initial level (int);
        [1]: Level index corresponding to final level (int);
        [2]: Associated initial level energy in keV (float);
        [3]: Associated final level energy in keV (float);
        [4]: Deexcitation gamma-ray energy in keV (float);
        [5]: Deexcitation gamma-ray energy uncertainty (float);
        [6]: Gamma-ray <intensity> according to keyword argument provided
             (float);
        [7]: Gamma-ray <intensity> uncertainty (float);
        [8]: BrIcc-calculated total internal-conversion coefficient
             (float);
        [9]: Total internal-conversion coefficient uncertainty (float).

    Examples:
        For isotopic partial gamma-ray cross sections:
        get_gammas(edata, "Y90", intensity="isotopic")
        get_gammas(edata, 39, 90, intensity="isotopic")
```

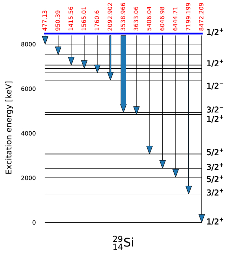
```
$ ipython
In [1]: import pyEGAF as egaf
In [2]: e = egaf.EGAF()
In [3]: data = e.load_egaf()
In [4]: help(e) # Method resolution order
In [5]: help(e.get_gammas) # Method
```

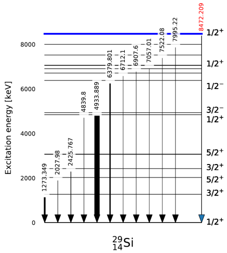
Docstrings are provided for all `pyEGAF` methods with the following general structure:

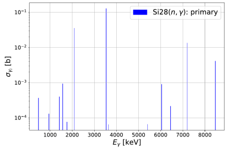
- Function description.
- Any special notes and/or references.
- Arguments passed to the function.
- What the function returns.
- Examples of function use.

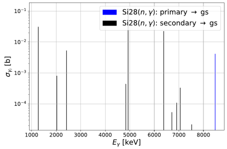
pyEGAF applications: Manipulating the γ data

Primary and secondary γ rays easily distinguishable using `pyEGAF`





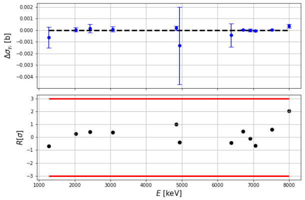




Determination of σ_0 using $\sum \sigma_{\gamma_i}$ from:

- Primaries
- Secondaries direct to GS (+ single primary to GS)

γ-ray intensity balance: `intensity_balance`



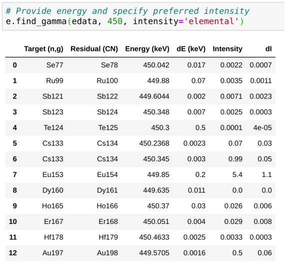
`gammas = e.get_gammas(edata, 'Si29', intensity=<kwarg>):`

- σ_{γ_i}: `intensity='isotopic'`
- σ_{γ_e}: `intensity='elemental'`
- P_γ: `intensity='population'`
- I_γ: `intensity='relative'`

Forensics applications using pyEGAF

There's a γ ray at $\sim$ 450 keV in my spectrum; where does it come from?

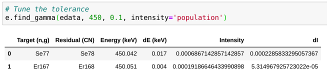
- Find all 450 keV γ rays in EGAF.
- Default tolerance ± 0.5 keV.



```python
# Provide energy and specify preferred intensity
e.find_gamma(edata, 450, intensity='elemental')
```

	Target (n,g)	Residual (CN)	Energy (keV)	dE (keV)	Intensity	dI
0	Se77	Se78	450.042	0.017	0.0022	0.0007
1	Ru99	Ru100	449.88	0.07	0.0035	0.0011
2	Sb121	Sb122	449.6044	0.002	0.0071	0.0023
3	Sb123	Sb124	450.348	0.007	0.0025	0.0003
4	Te124	Te125	450.3	0.5	0.0001	4e-05
5	Cs133	Cs134	450.2368	0.0023	0.07	0.03
6	Cs133	Cs134	450.345	0.003	0.99	0.05
7	Eu153	Eu154	449.85	0.2	5.4	1.1
8	Dy160	Dy161	449.635	0.011	0.0	0.0
9	Ho165	Ho166	450.37	0.03	0.026	0.006
10	Er167	Er168	450.051	0.004	0.029	0.008
11	Hf178	Hf179	450.4633	0.0025	0.0033	0.0003
12	Au197	Au198	449.5705	0.0016	0.5	0.06

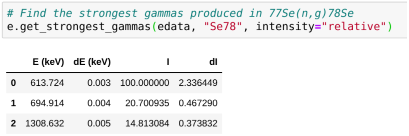
- Really accurate measurement!
- Tune tolerance window to ± 0.1 keV.



```python
# Tune the tolerance
e.find_gamma(edata, 450, 0.1, intensity='population')
```

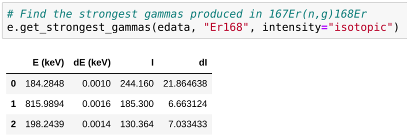
	Target (n,g)	Residual (CN)	Energy (keV)	dE (keV)	Intensity	dI
0	Se77	Se78	450.042	0.017	0.0006867142857142857	0.0002285833295057367
1	Er167	Er168	450.051	0.004	0.00019186646433990898	5.314967925723022e-05

- Find 3 strongest γ rays from ^{77}Se(n, γ).
- Find 3 strongest γ rays from ^{167}Er(n, γ).



```python
# Find the strongest gammas produced in 77Se(n,g)78Se
e.get_strongest_gammas(edata, "Se78", intensity="relative")
```

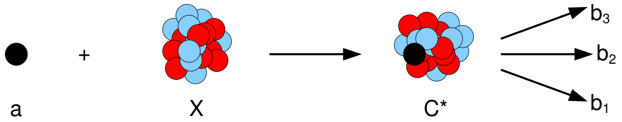
	E (keV)	dE (keV)	I	dI
0	613.724	0.003	100.000000	2.336449
1	694.914	0.004	20.700935	0.467290
2	1308.632	0.005	14.813084	0.373832



```python
# Find the strongest gammas produced in 167Er(n,g)168Er
e.get_strongest_gammas(edata, "Er168", intensity="isotopic")
```

	E (keV)	dE (keV)	I	dI
0	184.2848	0.0010	244.160	21.864638
1	815.9894	0.0016	185.300	6.663124
2	198.2439	0.0014	130.364	7.033433

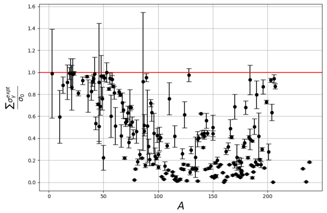
Assess completeness of capture-γ data using pyEGAF



- The cross section for fusion in entrance channel (a) is given by the sum of cross sections for decay to all final channels (b_i):

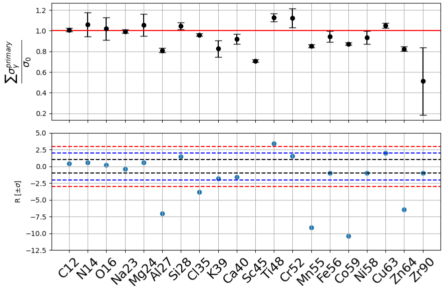
$$\sigma_F(a) = \sum_{b_i}^{N} \sigma^{C}_{a\rightarrow b_i} \quad \therefore \quad \sigma_0 = \sum_{i=1}^{N} \sigma^{\text{primary}}_{\gamma_i}.$$

- Compare $\sum_{i=1}^{N} \sigma^{\text{primary}}_{\gamma_i}$ from pyEGAF to Atlas-adopted total neutron-capture σ_0 values to assess *completeness*.



Materials of interest for planetary spectroscopy

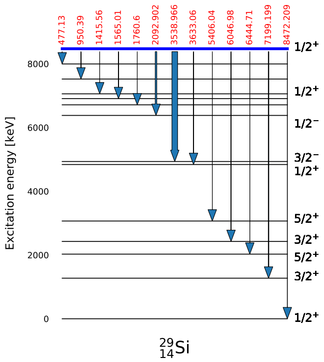
Isotope	Z	Abundance [%]
^{12}C	6	98.9
^{14}N	7	100
^{16}O	8	99.8
^{23}Na	11	100
^{24}Mg	12	79
^{27}Al	13	100
^{28}Si	14	92.2
^{35}Cl	17	75.8
^{39}K	19	93.3
^{40}Ca	20	96.9
^{45}Sc	21	100
^{48}Ti	22	73.7
^{52}Cr	24	83.8
^{55}Mn	25	100
^{56}Fe	26	91.7
^{59}Co	27	100
^{58}Ni	28	68.1
^{63}Cu	29	69.2
^{64}Zn	30	48.6
^{90}Zr	40	51.4

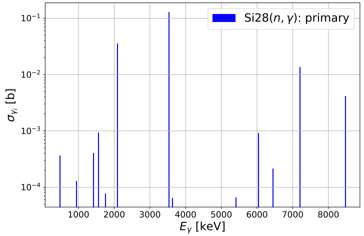


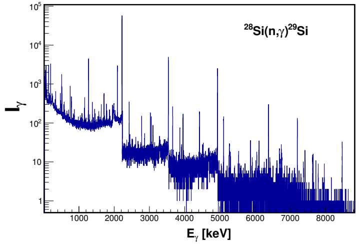
- 1σ agreement (complete!)
- 2σ agreement
- Differ by $\geq 3\sigma$

EGAF contains *complete* primary-γ datasets for many isotopes of interest.

Determination of σ_0 from primary γ rays





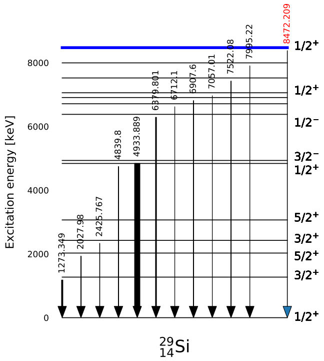


Data from EGAF via pyEGAF

$$\sigma_0 = \sum_{p=1}^{N} \sigma_{\gamma_p}(1 + \alpha_p).$$

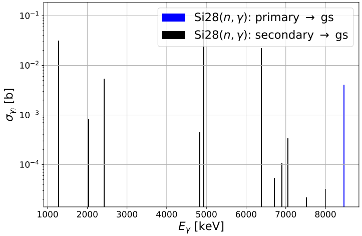
$\sigma_0 = 0.185(2)$ b cf. 0.177(5) b (Atlas); 0.186 b (ENDF)

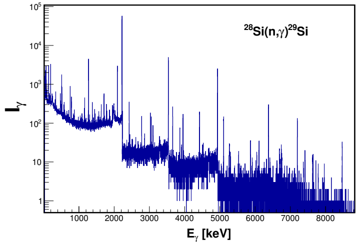
Determination of σ_0 from primary and secondary γ rays



Data from EGAF via pyEGAF

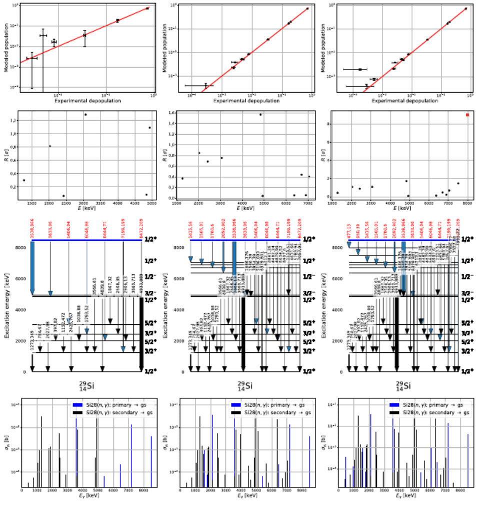
$$\sigma_0 = \sum_{i=1}^{N} \sigma_{\gamma_{i\to gs}} (1 + \alpha_{i\to gs}).$$



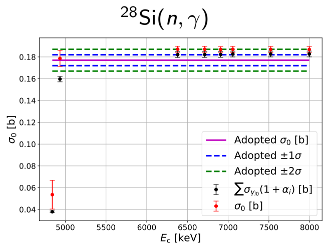


σ_0 = 0.187(3) b cf. 0.177(5) b (Atlas); 0.186 b (ENDF)

Modeling the (n,γ) reaction using pyEGAF methods

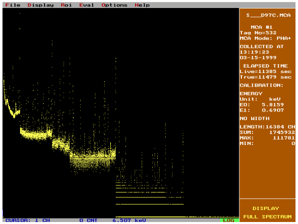


(a) $E_c = 4934.339$ keV (b) $E_c = 7057.94$ keV (c) $E_c = 7996.4$ keV



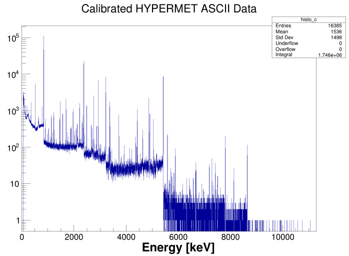
- Statistical-model analysis of (n,γ) reaction, e.g., ^{28}Si(n,γ).
- Compare modeled population of levels to experimental data.
- Establish critical energy E_c.

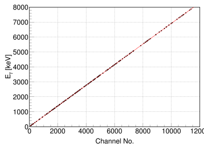
HypermetPC-to-ROOT: Conversion of entire Budapest PGAA catalogue



```
S___D97C.MCA

   MCA #1
Tag No=532
MCA Mode: PHA+
COLLECTED AT
13:19:23
03-15-1999
 ELAPSED TIME
Live=11385 sec
True=11479 sec
CALIBRATION:
ENERGY
Unit:   keV
E0:   5.8159
E1:   0.6907
NO WIDTH
LENGTH:16384 CH
SUM:    1745932
MAX:     111781
MIN:          0

CURSOR: 1 CH      0 CNT      6.507 keV
```

$^{32}\mathrm{S}(n,\gamma)$



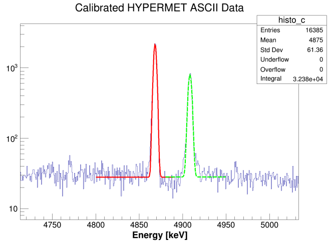


```
#ifndef CALIB_COEFFS_HPP
#define CALIB_COEFFS_HPP

//Linear coefficients
const double c0 = 3.87457e+00;
const double c1 = 6.90714e-01;
//Non-linear coefficients
const double c2 = 3.31126e-07;
const double c3 = -7.14708e-11;
const double c4 = -1.08361e-14;
const double c5 = 5.24697e-18;
const double c6 = -6.86670e-22;
const double c7 = 3.93610e-26;
const double c8 = -8.51060e-31;

#endif
```

- Permits wide dissemination in accordance with FAIR open-data principlces.
- Generate `ASCII` dump of 1D histos from `HypermetPC`.
- Run `ReadHypermetASCII.C` script in `C++` interpreter to generate correspond ing `ROOT` file.
- Reads in appropriate `<calib_coeffs.H>` file.
- Contains calibrated and uncalibrated `ROOT` histos.

ROOT: Bespoke analysis framework

```
#include <TMath.h>
//'x' and 'par' are effectively passed as arrays
Double_t fitGaus(Double_t *x, Double_t *par)
{
  Double_t arg = 0;
  if(par[2]!=0) arg = (x[0] - par[1])/par[2];

  Double_t fitval = par[0]*TMath::Exp(-0.5*pow(arg,2.0));

  return fitval;
}


Double_t backGround(Double_t *x, Double_t *par)
{
  //cf. y=a + b(x) + c(x^2)
  Double_t lineshape=par[0] + par[1]*x[0] + par[2]*pow(x[0],2.0) + par[3]*pow(x[0],3.0);

  return lineshape;
}

Double_t twoFuncs(Double_t *x, Double_t *par)
{

  return fitGaus(x,par)+backGround(x,&par[4]);

}
```



```
root [8]
root [8] .L gfit.C
root [9] .X fit.C
 FCN=391.217 FROM MIGRAD    STATUS=CONVERGED     348 CALLS         349 TOTAL
                     EDM=1.35505e-07    STRATEGY= 1  ERROR MATRIX UNCERTAINTY   2.4 per cent
  EXT PARAMETER                                   STEP         FIRST
  NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
   1  Constant     2.13573e+03   2.45826e+01  -9.98436e-02   7.90768e-06
   2  Mean         4.86821e+03   1.60429e-02  -1.04752e-05  -1.69886e-02
   3  Sigma        1.66131e+00   1.28960e-02  -4.67142e-05   3.76846e-02
   4  Slope        1.00000e+00   1.41421e+00  -0.00000e+00   0.00000e+00
   5  Intercept    2.80084e+01   4.93886e-01   1.98140e-03   9.85918e-04
 FCN=227.945 FROM MIGRAD    STATUS=CONVERGED     298 CALLS         299 TOTAL
                     EDM=8.22012e-09    STRATEGY= 1  ERROR MATRIX UNCERTAINTY   2.0 per cent
  EXT PARAMETER                                   STEP         FIRST
  NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
   1  Constant     7.62814e+02   1.41456e+01  -2.24281e-02  -1.49738e-06
   2  Mean         4.90839e+03   2.97096e-02  -1.35672e-04  -3.16059e-03
   3  Sigma        1.98723e+00   2.73521e-02   7.64100e-05  -6.18406e-03
   4  Slope        1.00000e+00   1.41421e+00  -0.00000e+00   0.00000e+00
   5  Intercept    2.82214e+01   5.79437e-01   1.16074e-03  -4.53992e-05
root [10]
```

- Simple Gaussian on a nonlinear background (`twoFuncs`).
- More suitable or more complicated functions can be programmed by the user.
- Results can be compared to original `Hypermet-PC` results contained in `PeakList`.

CapGam cf. pyEGAF::capgam

CapGam

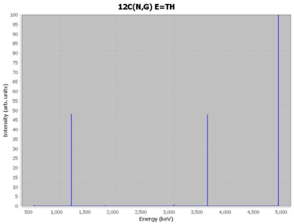
12C(N,G) E=TH
Target: ^{12}C
Product: ^{13}C

Last modified: 06/01/2023

PDF

Strongest Transition:
E(γ) = 4945.301 ± 0.003
I(γ) = 67.47 ± 0.92

Thermal Neutron Capture Cross Section (2006MuZX):
0.00353 b ± 0.00007 b

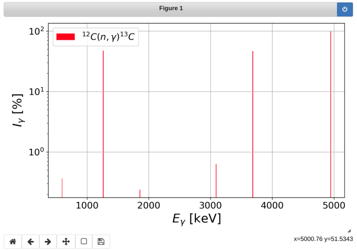


Download as .csv

Type	E(γ) (keV)	ΔE(γ) (keV)	I(γ)/I(γ)max × 100	Δ(I(γ)/I(γ)max)
Secondary	595.013	*0.011*	0.3557	*0.0156*
Primary	1261.764	*0.012*	47.9621	*0.9225*
Primary	1856.716	*0.012*	0.2371	*0.0152*
Secondary	3089.049	*0.020*	0.6373	*0.0309*
Secondary	3683.921	*0.023*	47.6360	*1.1482*
Primary	4945.301	*0.003*	100.0000	*1.9233*

pyEGAF

```
In [4]: # Define compound nucleus as a string, e.g., 12C(n,g)13C:
        compound_nucleus = 'C13'

In [5]: # Extract capture-gamma data for defined compound nucleus and write results to CSV file in pwd
        df=e.capgam(edata,"{0}".format(compound_nucleus))
        df.to_csv("capgam_style_{0}.csv".format(compound_nucleus), index=False)
        print(df.to_string(index=False))

        Target nucleus: C12
        Residual (compound nucleus): C13
        C12(n,g)C13
        Total radiative thermal neutron-capture cross section = 0.00387 b ± 3e-05
        Reference: 2018MuZY
        Maximum I = 0.0026389710000000004 b at E = 4945.301 keV; RI = 100.
             Type        E     dE         RI       dRI
         secondary   595.015  0.009   0.363985  0.015326
           primary  1261.765  0.009  47.509579  1.149512
           primary  1856.717  0.009   0.237548  0.015326
         secondary  3089.057  0.009   0.632184  0.030652
         secondary  3683.920  0.009  46.743295  1.149509
           primary  4945.301  0.003 100.000000  1.915939

In [6]: # Plot gamma-ray intensities from DataFrame
        plot_df_data(df['E'], df['RI'], "{0}".format(compound_nucleus))
```



CapGam cf. pyEGAF::capgam

CapGam

12C(N,G) E=TH
Target: ^{12}C
Product: ^{13}C

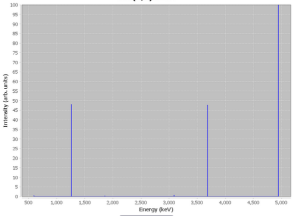
Last modified: 06/01/2023

PDF

Strongest Transition:
E(γ) = 4945.301 ± 0.003
I(γ) = 67.47 ± 0.92

Thermal Neutron Capture Cross Section (2006MuZX):
0.00353 b ± 0.00007 b

12C(N,G) E=TH



Download as .csv

Type	E(γ) (keV)	ΔE(γ) (keV)	I(γ)/I(γ)$_{max}$ × 100	Δ(I(γ)/I(γ)$_{max}$)
Secondary	595.013	*0.011*	0.3557	*0.0156*
Primary	1261.764	*0.012*	47.9621	*0.9225*
Primary	1856.716	*0.012*	0.2371	*0.0152*
Secondary	3089.049	*0.020*	0.6373	*0.0309*
Secondary	3683.921	*0.023*	47.6360	*1.1482*
Primary	4945.301	*0.003*	100.0000	*1.9233*

pyEGAF: ‘more’

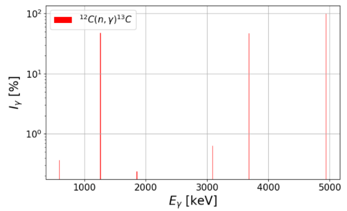
In [4]:
```python
# Define compound nucleus as a string
compound_nucleus = 'C13'
```

In [5]:
```python
# Extract capture-gamma data for defined compound nucleus and write results to CSV file in pwd
df=e.capgam(edata,"{0}".format(compound_nucleus),"more")
df.to_csv("capgam_style_{0}.csv".format(compound_nucleus), index=False)
print(df.to_string(index=False))
```

```
Target nucleus: C12
Residual (compound nucleus): C13
C12(n,g)C13
Total radiative thermal neutron-capture cross section = 0.00387 b ± 3e-05
Reference: 2018MuZY
Maximum I = 0.0026389710000000004 b at E = 4945.301 keV; RI = 100.
      Type  i  f      E(i)      E(f)         E     dE          RI       dRI
 secondary  2  1  3684.480  3089.451   595.015  0.009    0.363985  0.015326
   primary  3  2  4946.311  3684.480  1261.765  0.009   47.509579  1.149512
   primary  3  1  4946.311  3089.451  1856.717  0.009    0.237548  0.015326
 secondary  1  0  3089.451     0.000  3089.057  0.009    0.632184  0.030652
 secondary  2  0  3684.480     0.000  3683.920  0.009   46.743295  1.149509
   primary  3  0  4946.311     0.000  4945.301  0.003  100.000000  1.915939
```

In [6]:
```python
# Plot gamma-ray intensities from DataFrame
plot_df_data(df['E'], df['RI'], "{0}".format(compound_nucleus))
```

Figure 1



pyEGAF also displays associated level information.

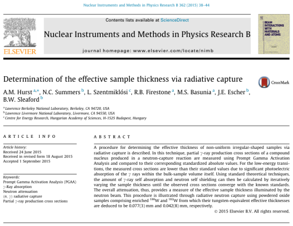
Attenuation in Prompt Gamma Activation Analysis

https://github.com/AaronMHurst/attenuation_integration

- C++ implementation for calculating attenuation integrated over sample thickness (t):

$$\frac{I_\gamma}{I_0}\int_{x=0}^{x=t} dx = \int_{x=0}^{x=t} \exp\left(\frac{-\rho\left(\frac{\mu_\gamma}{\rho}\right)_{E_\gamma} x}{\cos\theta}\right) dx.$$

- Build project out-of-source with CMake.
- Finds unique solution for effective t corresponding to observed attenuation.
- Project bundled with mass-attenuation coefficients for 100 elements (H ($Z = 1$) to Fm ($Z = 100$)) taken from XMuDat.
- Interpolated energies from 1 keV to 20 MeV.
- Program can also be used to calculate simple attenuation for elemental (e.g., Re, La. . .) or stoichiometric compound (e.g., $ReCl_3$, $LaCl_3 \cdot 7H_2O$. . .) samples.


Nuclear Instruments and Methods in Physics Research B 362 (2015) 38–44

Contents lists available at ScienceDirect

Nuclear Instruments and Methods in Physics Research B

journal homepage: www.elsevier.com/locate/nimb

Determination of the effective sample thickness via radiative capture

A.M. Hurst [a,*], N.C. Summers [b], L. Szentmiklósi [c], R.B. Firestone [a], M.S. Basunia [a], J.E. Escher [b], B.W. Sleaford [b]

[a] Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
[b] Lawrence Livermore National Laboratory, Livermore, CA 94550, USA
[c] Centre for Energy Research, Hungarian Academy of Sciences, H-1525 Budapest, Hungary

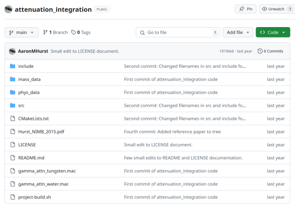
ARTICLE INFO

Article history:
Received 24 June 2015
Received in revised form 18 August 2015
Accepted 1 September 2015

Keywords:
Prompt Gamma Activation Analysis (PGAA)
γ-Ray absorption
Neutron attenuation
(n, γ) radiative capture
Partial γ-ray production cross sections

ABSTRACT

A procedure for determining the effective thickness of non-uniform irregular-shaped samples via radiative capture is described. In this technique, partial γ-ray production cross sections of a compound nucleus produced in a neutron-capture reaction are measured using Prompt Gamma Activation Analysis and compared to their corresponding standardized absolute values. For the low-energy transitions, the measured cross sections are lower than their standard values due to significant photoelectric absorption of the γ rays within the bulk-sample volume itself. Using standard theoretical techniques, the amount of γ-ray self absorption and neutron self shielding can then be calculated by iteratively varying the sample thickness until the observed cross sections converge with the known standards. The overall attenuation, thus, provides a measure of the effective sample thickness illuminated by the neutron beam. This procedure is illustrated through radiative neutron capture using powdered oxide samples comprising enriched ^{186}W and ^{182}W from which their tungsten-equivalent effective thicknesses are deduced to be 0.077(3) mm and 0.042(8) mm, respectively.


attenuation_integration Public

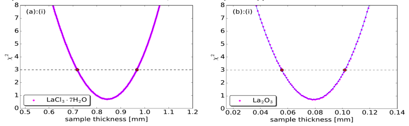
AaronMHurst Small edit to LICENSE document. 191feb6 · last year 6 Commits

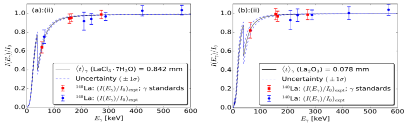
include	Second commit: Changed filenames in src and include fo...	last year
mass_data	First commit of attenuation_integration code	last year
phys_data	First commit of attenuation_integration code	last year
src	Second commit: Changed filenames in src and include fo...	last year
CMakeLists.txt	Second commit: Changed filenames in src and include fo...	last year
Hurst_NIMB_2015.pdf	Fourth commit: Added reference paper to tree	last year
LICENSE	Small edit to LICENSE document.	last year
README.md	Few small edits to README and LICENSE documentation.	last year
gamma_attn_tungsten.mac	First commit of attenuation_integration code	last year
gamma_attn_water.mac	First commit of attenuation_integration code	last year
project-build.sh	First commit of attenuation_integration code	last year

Effective thickness determination in La and Re samples

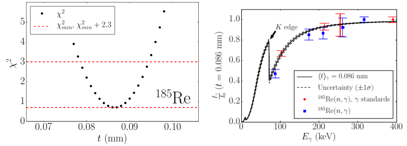
Prompt	Response
Calculate attenuation assuming coefficients for: 1 - gamma-ray attenuation only 2 - gamma-ray and neutron attenuation combined	1
Sample in [mm] or [cm] ? 1 - [mm] 2 - [cm]	2
Give sample thickness [cm]:	2.5
Temperature of neutron beam [K] ?	293
Natural Element (1) or Compound Sample (2) ?	2
Density of compound [g/cm^{3}]?	1.0
Number of Elements in compound?	2
Chemical symbol for Absorber No. 1 ?	H
Number of atoms belonging to H (i.e. stoichiometry) ?	2
Use adopted elemental absorption cross section from Mughabghab's Atlas of Neutron Resonances (Ed. 2006)? 1 - Yes 2 - No	1
Chemical symbol for Absorber No. 2 ?	O
Number of atoms belonging to O (i.e. stoichiometry) ?	1
Use adopted elemental absorption cross section from Mughabghab's Atlas of Neutron Resonances (Ed. 2006)? 1 - Yes 2 - No	1

[A.M. Hurst *el al.* PRC **99**, 024310 (2019)]





[D.A. Matters *el al.* PRC **93**, 054319 (2016)]



Iterate calculation over sample thickness t values to find unique solution.