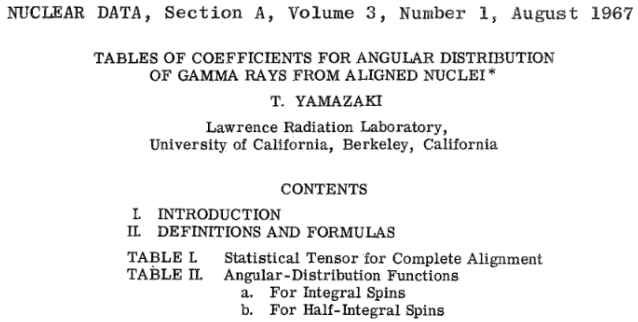
Analytic calculator for γ-ray angular distribution coefficients

Aaron M. Hurst
amhurst@berkeley.edu

US Nuclear Data Program Annual Meeting 2025
Brookhaven National Laboratory, Upton, NY, USA

October 28 – 31, 2025

Motivation



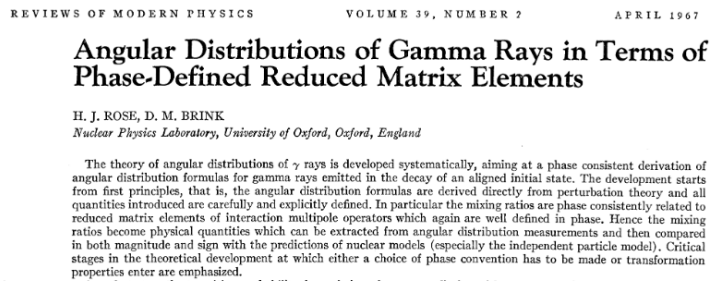
NUCLEAR DATA, Section A, Volume 3, Number 1, August 1967

TABLES OF COEFFICIENTS FOR ANGULAR DISTRIBUTION OF GAMMA RAYS FROM ALIGNED NUCLEI*

T. YAMAZAKI

Lawrence Radiation Laboratory, University of California, Berkeley, California

CONTENTS



REVIEWS OF MODERN PHYSICS VOLUME 39, NUMBER 2 APRIL 1967

Angular Distributions of Gamma Rays in Terms of Phase-Defined Reduced Matrix Elements

H. J. ROSE, D. M. BRINK
Nuclear Physics Laboratory, University of Oxford, Oxford, England

The theory of angular distributions of γ rays is developed systematically, aiming at a phase consistent derivation of angular distribution formulas for gamma rays emitted in the decay of an aligned initial state. The development starts from first principles, that is, the angular distribution formulas are derived directly from perturbation theory and all quantities introduced are carefully and explicitly defined. In particular the mixing ratios are phase consistently related to reduced matrix elements of interaction multipole operators which again are well defined in phase. Hence the mixing ratios become physical quantities which can be extracted from angular distribution measurements and then compared in both magnitude and sign with the predictions of nuclear models (especially the independent particle model). Critical stages in the theoretical development at which either a choice of phase convention has to be made or transformation properties enter are emphasized.

- Lots of important parameters used in nuclear physics data analysis only available in limited tabular form.
- Need to have libraries that support direct calculation of γ-ray angular-distribution quantities to enable automated analysis.
- `PyGammaRAD`: *Python project for Gamma-Ray Angular Distributions*
- Self-contained package providing capability to extend to arbitrary spin value, multipole order, or magnetic substate well beyond published tables.

Development of PyGammaRAD software package

- Small modular program implemented in Python.
- Separate classes based on multiple inheritance.
- Sum formulae for exact evaluation of vector coupling coefficients, angular distribution methods, machine-readable tabulated data.
- Open-source, well-documented README, docstrings, and reference article; permissive MIT license.
- Several `Jupyter Notebook` workflows.
- Logging framework for field reporting of application events.
- More than 150 unit tests to validate and benchmark performance.
- Package dependencies: `numpy`, `pandas`, `pytest`, and `scipy`.

```
from .tables import *
from .am_formulae import *
from .am_methods import *
from .angular_distributions import *
from .log_handlers import *

class AngularMomentum(Legendre):
    __doc__="""Class to handle the coupling and recoupling schemes of angular
    momenta needed in the calculation of gamma-ray angular distributions in
    aligned nuclei.  The angular momentum calculators also have purpose in
    wider quantum mechanical applications."""

    def __init__(self):
        Yamazaki.__init__(self)
        RoseAndBrink.__init__(self)
        AngularMomentumCalculations.__init__(self)
        AngularDistributions.__init__(self)
        Legendre.__init__(self)
```

All PyGammaRAD methods have verbose documentation

```
Help on method calc_F in module PyGammaRAD.angular_distributions:

calc_F(k, Jf, L1, L2, Ji) method of PyGammaRAD.PyGammaRAD.AngularMomentum instance
    Calculate F distribution coefficient for given value of k, Jf, L1,
    L2, and Ji in accordance with Eq. (4) from Yamazaki's paper [1].

    Notes:
        [1]: T. Yamazaki Nucl. Data Sect. A, Vol. 3, Num. 1 (1967).

    Arguments:
        k: An integer object representing the order.
        Jf: A number object (int or float) representing the final spin.
        L1: An integer object representing the order of the first
            multipole.
        L2: An integer object representing the order of the second
            multipole.
        Ji: A number object (int or float) representing the initial spin.

    Returns:
        A float value corresponding to Fk(Jf L1 L2 Ji) which can be
        compared to the corresponding value from Table 2 in Yamazaki's
        paper [1].

    Raises:
        Fewer than 5 positional arguments ('k', 'Jf', 'L1', 'L2', 'Ji')
        raises a TypeError exception.

    Example:
        To calculate F(k=2,Jf=2,L1=2,L2=2,Ji=0):
        > calc_F(2,0,2,2,2)
```

```
In [1]: import PyGammaRAD as pg
In [2]: am=pg.AngularMomentum()
In [3]: help(am)
In [4]: help(am.calc_F)
```

- Short explanation of the method.
- Description of arguments needed to be passed.
- Return value of the method.
- Exceptions that may get raised.
- Examples invoking direct use of the method.

Angular-distribution functions and methods

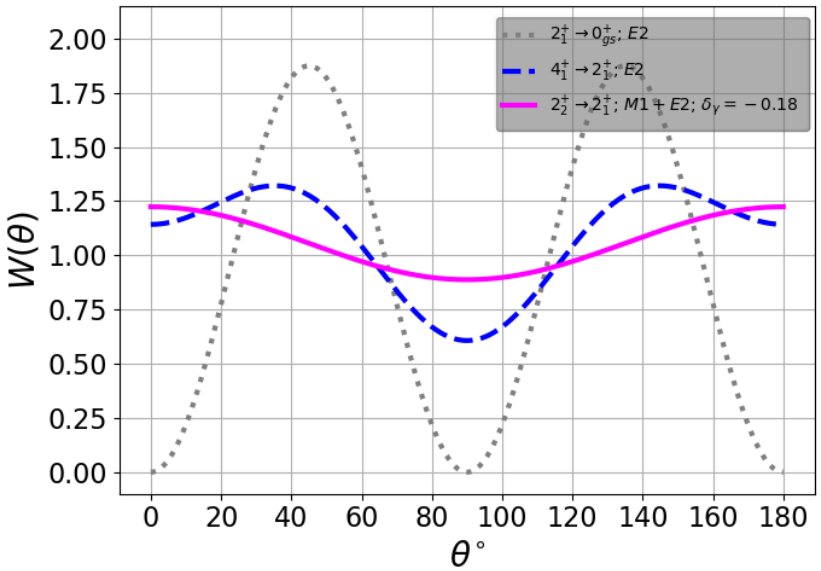
Returned quantity	Function [1],[2]	Method	Arguments
$W(\theta)$	Equation (2) [1]	`dist_W`	A_k, * [θ]
$F_k(J_f L_1 L_2 J_i)$	Equation (4) [1]	`calc_F`	k, J_f, L_1, L_2, J_i
$B_k(J)$	Equation (6) [1]	`calc_B`	k, J
$A_k^{max}(J_i L_1 L_2 J_f)$	Equation (7) [1]	`A_max`	k, J_i, L_1, L_2, J_f, δ_γ
$B_k(J_i)F_k(J_f L_1 L_2 J_i)$	Equation (8) [1]	`calc_BF`	k, J_f, L_1, L_2, J_i
$U_k(J_i L_1 L_2 J_f)$	Equation (13) [1]	`U_coeff`	k, J_i, L_1, L_2, J_f, δ_γ
$u_k(J_i L_1 J_f)$	Equation (14) [1]; Equation (3.45) [2]	`calc_u`	k, J_i, L_1, J_f
$R_k(L_1 L_2 J_i J_f)$	Equations (3.36) & (3.37) [2]	`calc_R`	k, L_1, L_2, J_i, J_f
$S_k(l_1 l_2 J s)$	Equation (3.59) [2]	`calc_S`	l_1, l_2, J, s, * k
$\rho_k(Jm)$	Equation (3.63) [2]	`calc_p`	k, J, * m

- k : Order of the coefficient or polynomial degree.
- J_i : Initial nuclear level of the associated γ-ray transition.
- J_f : Final nuclear level of the associated γ-ray transition.
- L_1 : First multipole order.
- L_2 : Second (interfering) multipole order.
- δ_γ : γ-ray multipole mixing ratio.
- A_k : Anisotropy coefficient of order k.
- $l1$: Orbital angular momentum of first partial wave.
- $l2$: Orbital angular momentum of second (interfering) partial wave.
- s: Reaction channel spin.
- m: Magnetic substate projection quantum number.

[1] T. Yamazaki, *Tables of Coefficients for Angular Distribution of Gamma Rays from Aligned Nuclei*, Nucl. Data, Sect. A, Vol. **3**, Num. 1 (1967).

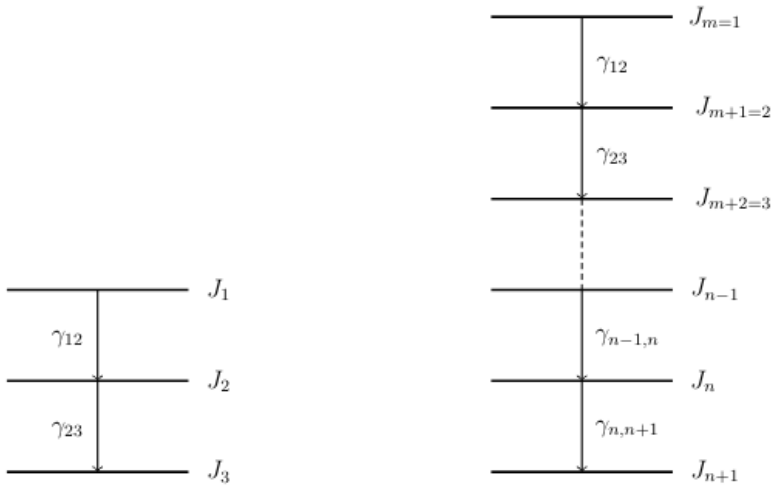
[2] H.J. Rose, D.M. Brink, *Angular Distributions of Gamma Rays in Terms of Phase-Defined Reduced Matrix Elements*, Rev. Mod. Phys., Vol. **39**, Num. 2, p. 306 (1967).

Calculated distributions



$$W(\theta) = 1 + A_2 P_2(\cos\theta) + A_4 P_4(\cos\theta)$$

- Individual γ rays.
- Definite J and L.
- δ_γ if available.

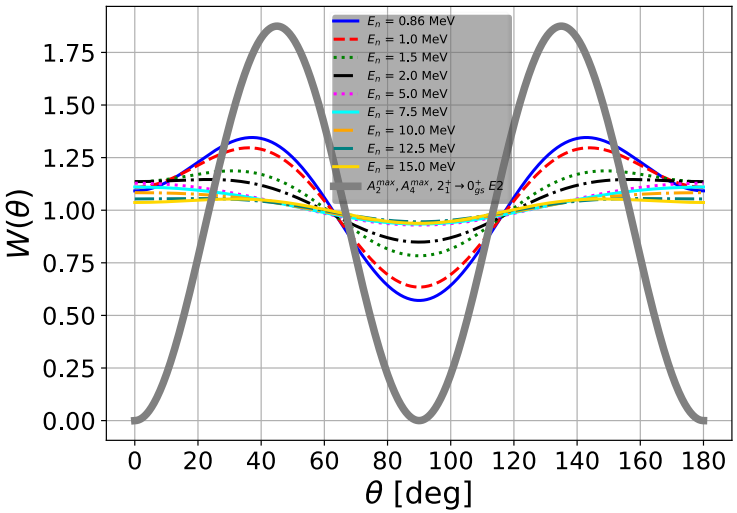


- For 2-step and multi-step cascades:

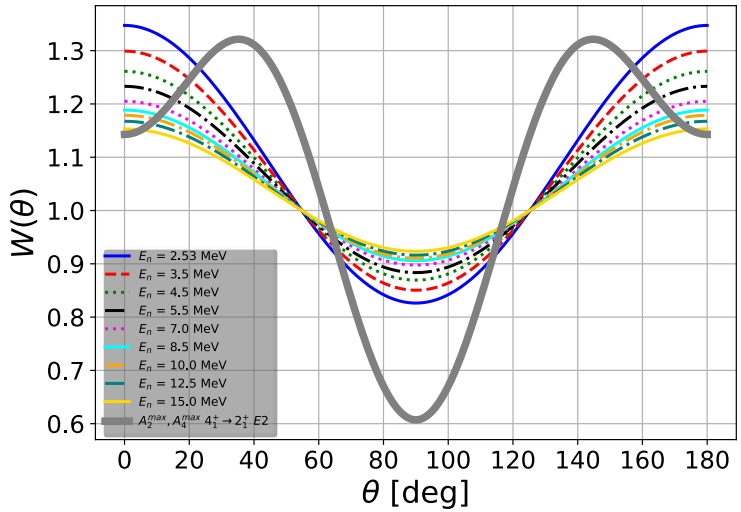
$$W(\theta) = \sum_{k=0,2,4} \left[A_k(J_n, J_{n+1}) P_k(\cos\theta) \times \prod_{J_m, J_{m+1}}^{J_{n-1}, J_n} U_k(J_m, J_{m+1}) \right]$$

- All preceding *"non-observed"* cascade-γ rays modified by U_k coefficient.

Comparison with experimental distributions



^{56}Fe: $2_1^+ \rightarrow 0_{gs}^+$



^{56}Fe: $4_1^+ \rightarrow 2_1^+$

- Compare calculated $W(\theta)$ to E_n-dependent experimental distributions for stretched quadrupole $E2$ transitions observed in ^{56}Fe$(n, n'\gamma)$: $W(\theta) = 1 + a_2 P_2(\cos\theta) + a_4 P_4(\cos\theta)$
- Experimental attenuated anisotropy coefficients (a_i) are spline-interpolated from fit to evaluated data [M.V. Savin *et al.*, J. Nucl. Sci. Tech. **37**, 748 (2000)].
- The E_n-dependent distributions differ significantly from the pure theoretical scenario assuming complete nuclear alignment of the magnetic substates where $P_{m_i}(J) = P_{-m_i}(J)$.

Manipulate and print tabulated data

- Effectively an API for interacting with the published data tables.
- User-defined data retrieval and manipulation.
- Regenerate original tables by Yamazaki and Rose and Brink in machine-readable CSV and JSON formats.


```
    "U2": 0.0,
    "U4": 0.0
},
{
    "Ji": 3,
    "Jf": 2,
    "L1": 2,
    "L2": 2,
    "F2": -0.12372,
    "B2F2": 0.14286,
    "F4": 0.67006,
    "B4F4": 0.85714,
    "U2": 0.20702,
    "U4": -0.62678
},
{
    "Ji": 3,
    "Jf": 3,
```


Yamazaki: *"Table 2(a)"*


```
    "S6": 0.39563,
    "S8": 0.0
},
{
    "l1": 4,
    "l2": 4,
    "J": 4,
    "s": 4,
    "S0": 1.0,
    "S2": 0.1702,
    "S4": -0.28152,
    "S6": -0.48561,
    "S8": 0.18123
},
{
    "l1": 4,
    "l2": 4,
    "J": 5,
```


Rose & Brink: *"Integral S"*

Angular momentum methods

Returned quantity	Coefficient/Symbol	Method	Arguments
Clebsch-Gordan	$< j_1 m_1 j_2 m_2 \mid jm >$	`cg`	j_1, m_1, j_2, m_2, j, m
Wigner 3-j	$\begin{matrix} j_1 & j_2 & j \\ m_1 & m_2 & m \end{matrix}$	`symb3j`	j_1, j_2, j, m_1, m_2, m
Racah	$W(j_1 j_2 j_3 j_4; j_5 j_6)$	`racah`	$j_1, j_2, j_3, j_4, j_5, j_6$
Wigner 6-j	$\begin{matrix} j_1 & j_2 & j_3 \\ j_4 & j_5 & j_6 \end{matrix}$	`symb6j`	$j_1, j_2, j_3, j_4, j_5, j_6$
Wigner 9-j	$\begin{matrix} j_1 & j_2 & j_3 \\ j_4 & j_5 & j_6 \\ j_7 & j_8 & j_9 \end{matrix}$	`symb9j`	$j_1, j_2, j_3, j_4, j_5, j_6, j_7, j_8, j_9$

- j: Angular momentum vector.
- m: Magnetic substate quantum number (z-axis projection).
- The usual coupling rules apply — see README or reference article.

Installing and running the program

https://test.pypi.org/project/PyGammaRAD/

- PyGammaRAD is freely available on PyPI.
- A public GitHub repository will be available soon.
- In the meantime, the PyPI test version of the program can be installed via pip from the command line:
  `$ pip install PyGammaRAD`
- Run through Jupyter Notebooks for example workflows.

```
In [1]: import PyGammaRAD as pg
In [2]: am = pg.AngularMomentum()

# Calculate Clebsch-Gordan coefficient <j1=5/2 m1=3/2
   j2=5/2 m2=-1/2 | j=1 m=1>
In [3]: am.cg(2.5, 1.5, 2.5, -0.5, 1, 1)
Out[3]: -0.4780914437337574 # -2*sqrt(2/35)

# Calculate coefficient F(k=2,Jf=0,L1=2,L2=2,Ji=2)
In [4]: am.calc_F(2, 0, 2, 2, 2)
Out[4]: -0.5976143046671966

# Compare to result in "Table 2(a)" [Yamazaki 1967]
In [5]: am.get_row_table2(2,0,2,2,2,coeff='F')
Out[5]: -0.59761

# Print "Table 2(a)" from Ref. [Yamazaki 1967] to file
   in JSON format
In [6]: am.table2file('T2A', 'JSON')
INFO:PyGammaRAD.log_handlers:YamazakiTable2a.json
```

Verification of the program and checking the data

- All data from Yamazaki's table reproduced exactly by PyGammaRAD.
- Serves as underlying check for the evaluation of Clebsch-Gordan and Racah coefficients needed to calculate γ-ray angular distribution coefficients and tensors.
- We found 3 errors/typos in Rose and Brink data tables; all other results are also reproduced precisely by PyGammaRAD.
- Vector-coupling coefficients: Clebsch-Gordan, Wigner 3-j, 6-j, and 9-j calculation methods were also verified extensively against published tabulated data.
- Users can run through suite of unit-tests and Jupyter Notebooks to demonstrate conformity.
- Some large-integer arguments have also been confirmed based on Newton's iterative method, e.g.:

$$\text{Wigner 3-}j\text{:}\quad \begin{pmatrix} 70 & 75 & 80 \\ 20 & -40 & 20 \end{pmatrix} = -\frac{6347289054877299473594}{105} \times \sqrt{\frac{3116246}{15918408720394785824242360109175680500179549419669 9}}$$

$$\text{Wigner 6-}j\text{:}\quad \begin{Bmatrix} 80 & 70 & 60 \\ 60 & 70 & 80 \end{Bmatrix} = \frac{22937084180342998024692848841405644906343471096259}{380794291048817629474203186897840050293419659023 6245} \times \sqrt{\frac{133733857259383840245}{65742467686458947573719}}$$

$$\text{Wigner 9-}j\text{:}\quad \begin{Bmatrix} 100 & 80 & 50 \\ 50 & 100 & 70 \\ 60 & 50 & 100 \end{Bmatrix} = 1.05597798 \times 10^{7}$$

Summary and outlook

- Set of calculators has been developed in Python to return precise γ-ray angular distribution coefficients and tensors.
- Project available on PyPI; easy to install and run; well documented.
- Software package loaded with stand-alone general purpose vector-coupling calculator enabling a broader range of quantum-mechanical applications based on the theory of angular momentum.
- All calculation methods verified and benchmarked against a large plethora of published data.
- Coefficients can be calculated for arbitrary values of J, L, m, k, . . . well beyond the limited range of published tabulated data.
- Generate machine-readable feature-rich formats with contextual information allowing for mapping of raw data into feature vectors providing useful labeled γ-ray distribution datasets.
- Small modular nature of `PyGammaRAD` enables straightforward integration into larger codebases.

A.M. Hurst[1,2], D.A. Matters[2], T. Kawano[3], M. Cromaz[2]

[1] University of California, Berkeley

[2] Lawrence Berkeley National Laboratory

[3] Los Alamos National Laboratory

This material is based upon work supported by the Department of Energy National Nuclear Security Administration through the Nuclear Science and Security Consortium under Award Number DE-NA0003996. Support was also received through the Lawrence Berkeley National Laboratory Contract No. DE-AC02-05CH11231 for the USNDP and by the Los Alamos National Laboratory under Contract No. 89233218CNA000001.