

自己出力型放射線検出器の出力電流値計算コードの作成

－アルゴリズムの構築と計算結果の比較－

Calculation Code of Output Current for Self-powered Radiation Detector

-Algorithm Construction and Comparison of Calculation Results-

柴田 裕司 武内 伴照 関 美沙紀 柴田 晃
中村 仁一 井手 広史

Hiroshi SHIBATA, Tomoaki TAKEUCHI, Misaki SEKI, Akira SHIBATA
Jinichi NAKAMURA and Hiroshi IDE

高速炉・新型炉研究開発部門

大洗研究所

環境技術開発センター

材料試験炉部

Department of JMTR

Waste Management and Decommissioning Technology Development Center

Oarai Research and Development Institute

Sector of Fast Reactor and Advanced Reactor Research and Development

March 2022

Japan Atomic Energy Agency

日本原子力研究開発機構

JAEA-Data/Code

本レポートは国立研究開発法人日本原子力研究開発機構が不定期に発行する成果報告書です。本レポートはクリエイティブ・コモンズ表示 4.0 国際 ライセンスの下に提供されています。本レポートの成果（データを含む）に著作権が発生しない場合でも、同ライセンスと同様の条件で利用してください。（<https://creativecommons.org/licenses/by/4.0/deed.ja>）
なお、本レポートの全文は日本原子力研究開発機構ウェブサイト（<https://www.jaea.go.jp>）より発信されています。本レポートに関しては下記までお問合せください。

国立研究開発法人日本原子力研究開発機構 JAEA イノベーションハブ 研究成果利活用課
〒 319-1195 茨城県那珂郡東海村大字白方 2 番地 4
電話 029-282-6387, Fax 029-282-5920, E-mail:ird-support@jaea.go.jp

This report is issued irregularly by Japan Atomic Energy Agency.
This work is licensed under a Creative Commons Attribution 4.0 International License (<https://creativecommons.org/licenses/by/4.0/deed.en>).
Even if the results of this report (including data) are not copyrighted, they must be used under the same terms and conditions as CC-BY.
For inquiries regarding this report, please contact Institutional Repository and Utilization Section, JAEA Innovation Hub, Japan Atomic Energy Agency.
2-4 Shirakata, Tokai-mura, Naka-gun, Ibaraki-ken 319-1195 Japan
Tel +81-29-282-6387, Fax +81-29-282-5920, E-mail:ird-support@jaea.go.jp

自己出力型放射線検出器の出力電流値計算コードの作成 —アルゴリズムの構築と計算結果の比較—

日本原子力研究開発機構 高速炉・新型炉研究開発部門 大洗研究所
環境技術開発センター 材料試験炉部

柴田 裕司、武内 伴照、関 美沙紀
柴田 晃⁺¹、中村 仁一⁺²、井手 広史

(2021年12月1日受理)

日本原子力研究開発機構大洗研究所に設置されている材料試験炉(Japan Materials Testing Reactor: JMTR)では過去30年以上、多様な原子炉材料や照射技術及び計測装置の開発が行われてきた。その中では自己出力型中性子検出器(Self-Powered Neutron Detectors: SPNDs)やガンマ線検出器(Self-Powered Gamma Detectors: SPGDs)の開発も行われており、複数の研究成果が報告されている。

しかし、それら成果のほとんどは検出器の開発に関する創意工夫と炉内照射試験、コバルト60によるガンマ線照射試験の結果を整理又は考察したものであり、検出器出力の理論的な解析及び評価はあまり行われてこなかった。

そこで、これら自己出力型放射線検出器の理論的な評価を行うための準備として1974年にH.D.WarrenとN.H.Shahが著した論文『Neutron and Gamma-Ray Effects on Self-Powered In-Core Radiation Detectors』を基に数値計算コードの作成を行った。本稿は作成した数値計算コードの内容について報告を行うものである。

大洗研究所：〒311-1393 茨城県東茨城郡大洗町成田町4002番地

+1 原子力科学研究部門 J-PARC センター 物質・生命科学ディビジョン

+2 安全研究・防災支援部門 安全研究センター 原子炉安全研究ディビジョン

**Calculation Code of Output Current for Self-powered Radiation Detector
- Algorithm Construction and Comparison of Calculation Results -**

Hiroshi SHIBATA, Tomoaki TAKEUCHI, Misaki SEKI
Akira SHIBATA⁺¹, Jinichi NAKAMURA⁺² and Hiroshi IDE

Department of JMTR,
Waste Management and Decommissioning Technology Development Center,
Oarai Research and Development Institute,
Sector of Fast Reactor and Advanced Reactor Research and Development,
Japan Atomic Energy Agency
Oarai-machi, Higashiibaraki-gun, Ibaraki-ken

(Received December 1, 2021)

Japan Materials Testing Reactor (JMTR) in Oarai Research and Development Institute of the Japan Atomic Energy Agency has been developing various reactor materials, irradiation techniques and instruments for more than 30 years. Among them, the development of self-powered neutron detectors (SPNDs) and gamma detectors (SPGDs) has been carried out, and several research results have been reported.

However, most of the results are based on the design study of the detector development and the results of in-core irradiation tests and gamma irradiation tests using Cobalt-60.

In this report, a numerical code is developed based on the paper "Neutron and Gamma-Ray Effects on Self-Powered In-Core Radiation Detectors" written by H.D. Warren and N.H. Shah in 1974, in order to theoretically evaluate the self-powered radiation detectors.

Keywords: Self-powered Radiation Detector, SPND, SPGD, Numerical Calculation

+ 1 Materials and Life Science Division, J-PARC Center,
Sector of Nuclear Science Research

+ 2 Reactor Safety Research Division, Nuclear Safety Research Center,
Sector of Nuclear Safety Research and Emergency Preparedness

目 次

1. 序論	1
2. 参照論文の解釈と具体的な計算式の整理	1
2.1 放射線検出器の概要と数値計算コードの必要性	1
2.2 一段階及び二段階での相互作用の共通事項	3
2.3 一段階の相互作用による発生電流の計算式	6
2.3.1 コンプトン効果による発生電流の計算式	6
2.3.2 光電効果による発生電流の計算式	6
2.3.3 内部転換による発生電流の計算式	7
2.3.4 ベータ崩壊による発生電流の計算式	7
2.4 二段階の相互作用による発生電流の計算式	8
2.4.1 中性子捕獲反応－コンプトン効果による発生電流の計算式	8
2.4.2 中性子捕獲反応－光電効果による発生電流の計算式	8
3. 数値計算コードの設計	9
3.1 全体動作	9
3.2 グローバル変数と関数の定義	10
3.3 読み込む離散データの形式と参照方法	13
3.4 中性子捕獲率" R_n "の計算方法	14
3.5 一段階の相互作用による発生電流の計算方法	15
3.5.1 コンプトン効果による発生電流の計算方法	15
3.5.2 光電効果による発生電流の計算方法	18
3.5.3 内部転換及びベータ崩壊による発生電流の計算方法	20
3.6 二段階の相互作用による発生電流の計算方法	20
4. 数値計算コードの実行結果	21
4.1 読込データの準備	21
4.2 計算結果の比較と考察	23
5. まとめ	26
謝辞	27
参考文献	27
付録 ソースコード	28

Contents

1. Introduction	1
2. Interpretation of the Reference Article and Rearranging of the Calculating Formula	1
2.1 Overview of the radiation detector and Necessity of the Calculation Code	1
2.2 Common Items of Single/Two Interactions	3
2.3 Calculation Formula of Output Current for Single-interaction	6
2.3.1 Calculation Formula for Compton Effect	6
2.3.2 Calculation Formula for Photoelectric Effect	6
2.3.3 Calculation Formula for Internal Conversion	7
2.3.4 Calculation Formula for Beta Decay	7
2.4 Calculation Formula of Output Current for Two-interaction	8
2.4.1 Calculation Formula for Compton Effect with Neutron Capture	8
2.4.2 Calculation Formula for Photoelectric Effect with Neutron Capture	8
3. Design of the Calculation Code	9
3.1 Overall Movement	9
3.2 Global Variable and Definition of the Functions	10
3.3 Definition of the Input Data Form and Reference Method	13
3.4 Calculation Method of the Neutron Capture Rate "R _n "	14
3.5 Calculation Method of Output Current with Single-interaction	15
3.5.1 Calculation Method for Compton Effect	15
3.5.2 Calculation Method for Photoelectric Effect	18
3.5.3 Calculation Method for Internal Conversion and Beta Decay	20
3.6 Calculation Method of Output Current with Two-interaction	20
4. Result of Calculation Code	21
4.1 Preparations for Input Data	21
4.2 Comparison of the Calculation Results	23
5. Concluding Remarks	26
Acknowledgements	27
References	27
Appendix Programming Source Code	28

1. 序論

原子炉施設や放射線利用施設における放射線計測の手法には様々なものがある。しかし、稼働中の原子炉炉心近傍のような過酷な環境において使用可能な放射線計測器は、機械的に強固で材料力学的に安定しているとともに、小型かつ単純な構造のものである必要がある。

この要求を満たし得る計測器として、自己出力型の中性子検出器(Self-Powered Neutron Detectors: SPNDs)及びガンマ線検出器(Self-Powered Gamma Detectors: SPGDs)が挙げられる。日本原子力研究開発機構(原子力機構)大洗研究所に設置されている材料試験炉(Japan Materials Testing Reactor: JMTR)では過去30年以上にわたって同検出器の開発及び照射試験が行われてきた。しかし、試験結果に関して、理論的な解析及び評価はほとんど行われていない。

このため、本稿は同検出器の照射試験データに対して、同検出器の出力電流値に関する理論的な計算を扱った論文¹⁾に基づいて、汎用性を有する数値計算コードを作成する事を目的としている。

2. 参照論文の解釈と具体的な計算式の整理

本稿で作成する数値計算コードの基盤となる論文は、1974年にH.D.WarrenとN.H.Shahが著した『Neutron and Gamma-Ray Effects on Self-Powered In-Core Radiation Detectors』¹⁾である。本章では始めに自己出力型放射線検出器の概要を説明し、解析上の前提条件や共通事項を記した後、各項で相互作用毎の具体的な発生電流の計算式を整理する。

2.1 放射線検出器の概要と数値計算コードの必要性

まず、本稿で想定している放射線検出器の概要について Fig.2.1 を用いて説明する。この検出器の放射線感受部は細い円柱状の形をしており、中心部には放射線との相互作用で電子を放出するエミッタと呼ばれる金属材料が配置され、その周囲を絶縁材とシース/コレクタに覆われた構造となっている。放射線がエミッタに当たり相互作用を起こすと電子がエミッタからコレクタに向けて移動し、コレクタから代わりの電子が微小電流計を通りエミッタに移動するという現象が発生する。測定原理としては、この電流計測値から放射線の強さを推算する形となっている。

電流は放射線とエミッタの複数の相互作用に起因して発生するため、本稿では前述の論文に記された計算式を用いて数値計算を行い、試験結果の妥当性及び放射線とエミッタ間で生じる各相互作用の寄与の割合を見積もるための準備を行う。しかし、放射線場や反応断面積等の情報が計算式の各所に現れるため、数値データの読込/参照機能を作成する必要があり、また解析的に解く事ができない部分については近似を用いる等、工夫が必要となる。

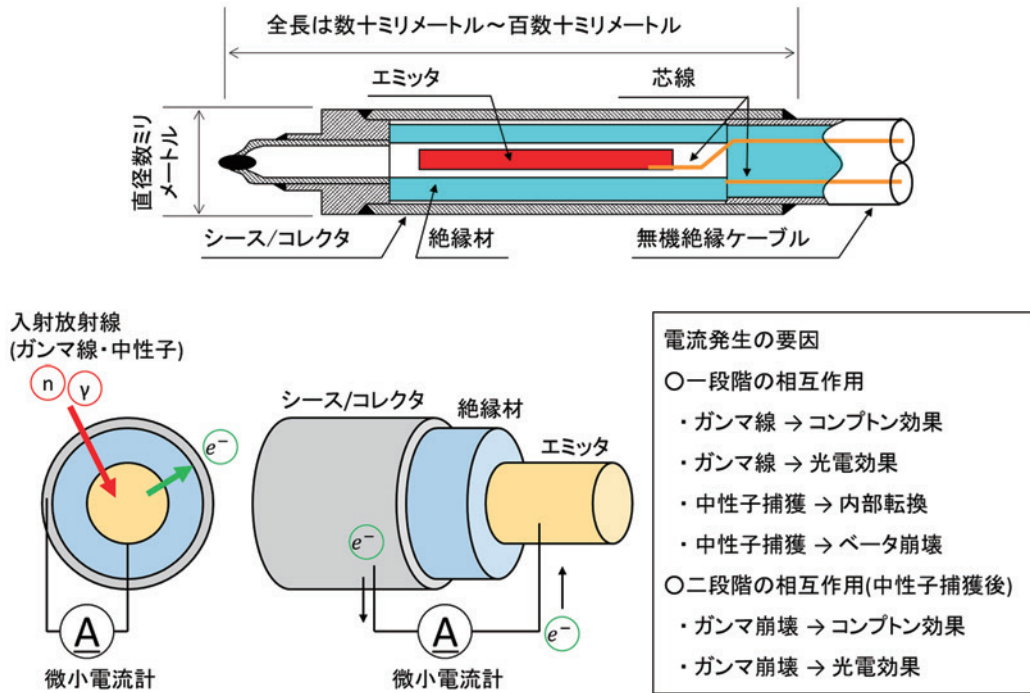


Fig.2.1 自己出力型放射線検出器の構造と測定原理

検出器で生じる相互作用として論文¹⁾で取り扱われているものは以下の 6 種類であり、いずれの相互作用についても具体的な計算式が示されている。

○一段階の相互作用(Single-interactions)

- ・ コンプトン効果(γ, e_{ce}) ・ 光電効果(γ, e_{pe})
- ・ 内部転換(n, e_{ic}) ・ ベータ崩壊(n, β^-)

○二段階の相互作用(Two-interactions)

- ・ 中性子捕獲反応－ガンマ崩壊－コンプトン効果(n, γ, e_{ce})
- ・ 中性子捕獲反応－ガンマ崩壊－光電効果(n, γ, e_{pe})

なお、論文に記されている計算式には以下に示す前提条件が定められている。

- (1) 放射線検出器の形状は同軸円柱状で、十分細いものである事。
即ち対称性があり、エミッタの中心部と外側で電子の運動エネルギーの損失に大きな差がない事。
- (2) 放射線は等方的で時間変化が無い事。
即ちエミッタで生じる相互作用の分布に偏りが無い事。
- (3) 中性子照射による相互作用で、核変換に伴う相互作用の総数の変化は考慮しない。
- (4) 単位系は CGS 単位系とする。

2.2 一段階及び二段階での相互作用の共通事項

始めに、Fig.2.2に示すような一段階の相互作用(Single-interactions)において基礎となる方程式とその定義及び内容について記す。この式は論文¹⁾中において次式のように示されており、同式中の関数及び文字は以下のように定義される。

$$dI = eV \times S(E', E)dE \times P(E_i, E')dE' \times \phi(E_i)dE_i$$

e : 電気素量(1.6×10^{-19} [C])

V : 放射線と相互作用をする部分(エミッタ)の体積[cm^3]

$S(E', E)$: エミッタ内部で発生した初期エネルギー E' の電子が、エミッタの表面に到達した時に、どの程度のエネルギーを持っているのかを表す関数で、エネルギーが E となる確率を $S(E', E)$ と表す。

エミッタの材質や形状に依存する関数ではあるが、前節の条件(1),(2)より、論文の中では最終的に下式のように近似して表されている。

$$S(E, E') \cong \frac{S K}{4V \rho}$$

ここで、式中の文字の定義は以下の通りである。

S : エミッタの表面積[cm^2]

ρ : エミッタの密度[g cm^{-3}]

K : エミッタの密度と線阻止能の逆数を掛けたもの。エネルギーに対して変化する値で、厳密には定数ではないが、エネルギーが 0.2~6.0[MeV]の時の加重平均値として 6.5×10^{-4} [$\text{g cm}^{-2} \text{keV}^{-1}$]が与えられている。

$P(E_i, E')$: 入射してきた放射線(ガンマ線・中性子)によって、エミッタ内部でどの程度の初期エネルギーを持つ電子が生成されるのかを表す関数で、エネルギーが E' となるエミッタの単位長さ当たりの確率を $P(E_i, E')$ と表す。この関数は4つの相互作用毎に定義されている。

コンプトン効果 : $P(E_i, E') \equiv NZ\sigma(E_\gamma, E_\gamma - E')$

光電効果 : $P(E_i, E') \equiv \mu(E_\gamma)\delta(E_\gamma - E_K - E')$

内部転換(K殻) : $P(E_i, E') \equiv \Sigma_{it}(E_n)f(E_n)\frac{\eta_K}{1+\eta_T}\delta(E_{it} - E_K - E')$

ベータ崩壊 : $P(E_i, E') \equiv \frac{105}{16}E_\beta^{-7/2}(E_\beta - E')^2\sqrt{E'}\Sigma_\beta(E_n)f(E_n)$

ここで、式中の文字の定義は以下の通りである。

N : 単位体積当たりの原子数[個 cm^{-3}]

Z : 原子番号

$\Sigma_{it}(E_n), \Sigma_\beta(E_n)$: エネルギーが E_n の中性子に対する、エミッタの核異性体転移とベータ崩壊に関する中性子捕獲断面積。

$f(E_n)$: エネルギーが E_n の中性子に対するエミッタの自己遮蔽係数。

η_K, η_T : K殻での内部転換係数。全殻での内部転換係数。

E_K, E_β, E_{it} : K 殻電子の結合エネルギー。ベータ線の最大エネルギー。
 内部転換に関わる核異性体転移のエネルギー。

$\sigma(E_\gamma, E_\gamma - E')$: コンプトン散乱の断面積を表す関数で次式のように定義される。

$$\sigma(E_\gamma, E_1) = \pi r_0^2 \frac{mc^2}{E_\gamma^2} \left[\frac{E_\gamma}{E_1} + \frac{E_1}{E_\gamma} + \left(\frac{mc^2}{E_1} - \frac{mc^2}{E_\gamma} \right)^2 - 2 \left(\frac{mc^2}{E_1} - \frac{mc^2}{E_\gamma} \right) \right]$$

ここで、式中の文字の定義は以下の通りである。

r_0^2 : トムソン散乱の断面積から得られる値($7.94 \times 10^{-26} [\text{cm}^2]$)

mc^2 : 電子の静止エネルギー($511 [\text{keV}]$)

E_1 : 散乱ガンマ線のエネルギー(= $E_\gamma - E'$)

$\mu(E_\gamma)$: 光電子吸収係数を表す関数で次式のように定義される。

$$\mu(E_\gamma) = \frac{5}{4} \times \frac{32}{3} \sqrt{2} \pi r_0^2 N Z^5 \left(\frac{1}{137} \right)^4 \left(\frac{mc^2}{E_\gamma} \right)^{7/2}$$

ここで、右辺の5/4は、K 殻で光電子の約 80[%]が吸収される事から、その補正として乗じている係数である。

$\phi(E_i)$: 単位時間[sec]、単位面積[cm^2]を通過する放射線(ガンマ線・中性子)の密度を表す関数で、以下の制約条件が与えられている。

$$\int_0^{E_{i \max}} \phi(E_i) dE_i = 1 \quad [\text{個 sec}^{-1} \text{ cm}^{-2}]$$

ここで、 $E_{i \max}$ は入射する放射線の最大エネルギーを表す。この関数は言い換えれば「規格化された入射粒子のエネルギー毎の粒子フルエンス率」とも表現できる。以降、簡単のためこれをスペクトルと呼ぶ。

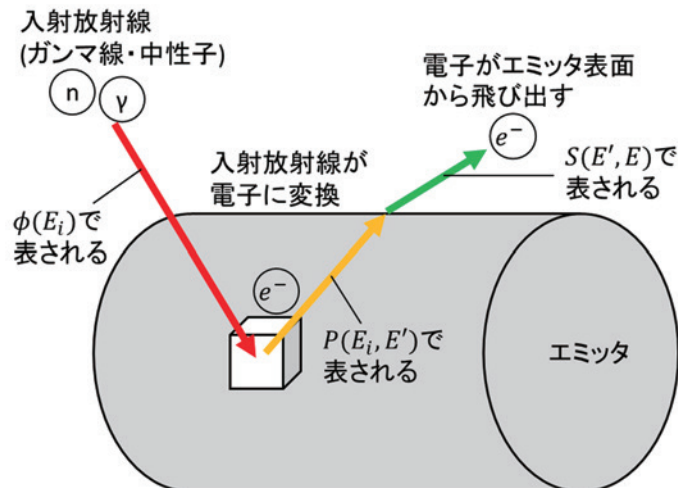


Fig.2.2 一段階の相互作用の模式図

次に、Fig.2.3に示すような二段階の相互作用(Two-interactions)において基礎となる方程式とその定義及び内容について記す。この式は論文¹⁾中において次式のように示されており、同式中の関数及び文字は以下のように定義される。なお、既出のものは省略する。

$$I_i = eV \times S(E', E)dE \times Q(E_{\gamma i}, E')dE' \times M(E_{\gamma i}) \times R_n$$

$Q(E_{\gamma i}, E')$: 中性子捕獲によってエミッタで発生したガンマ線が電子を生成する相互作用を表す関数で、生成した電子のエネルギーが E' となる確率を $Q(E_{\gamma i}, E')$ と表す。コンプトン効果と光電効果の場合では式が異なり、以下のように定義される。ただし、これは前節の条件(2),(3)に示されるようにエミッタ内での相互作用の発生確率に偏りが無いという前提での式である。

コンプトン効果 : $Q(E_{\gamma i}, E') \equiv NZ\bar{l}\sigma(E_{\gamma i}, E_{\gamma i} - E')$

光電効果 : $Q(E_{\gamma i}, E') \equiv \mu(E_{\gamma i})\bar{l}\delta(E_{\gamma i} - E_K - E')$

ここで、式中の文字の定義は以下の通りである。

\bar{l} : ガンマ線のエミッタ表面までの平均飛程。 d はエミッタの直径。

$$\bar{l} = \begin{cases} 2d/3 & (\text{半径が十分小さい場合}) \\ d/2 & (\text{中性子捕獲断面積が大きい場合 例.Gd}) \end{cases}$$

$M(E_{\gamma i})$: 中性子捕獲反応(n, γ)によって、生成されるエネルギー $E_{\gamma i}$ のガンマ線の本数。

Orphanらの研究成果²⁾が用いられ、数式の形では表現されない。

R_n : 中性子捕獲率を表す項で、以下のように定義される。

$$R_n = \int_0^{E_m} \Sigma(E_n)f(E_n)\phi(E_n)dE_n$$

ここで、式中の文字の定義は以下の通りである。

$\Sigma(E_n)$: エネルギー E_n の中性子に対するエミッタの中性子捕獲断面積。

$f(E_n)$: エネルギー E_n の中性子に対するエミッタの自己遮蔽係数。

$\phi(E_n)$: 中性子場のスペクトル。

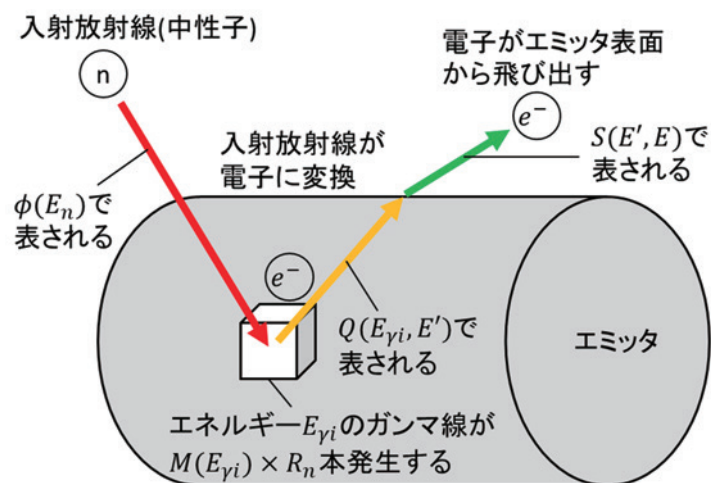


Fig.2.3 二段階の相互作用の模式図

2.3 一段階の相互作用による発生電流の計算式

2.3.1 コンプトン効果による発生電流の計算式

コンプトン効果による発生電流の計算式を定義通りに記述すると以下ようになる。

$$dI = eV \times \left[\frac{S}{4V} \frac{K}{\rho} dE \right] \times [NZ\sigma(E_\gamma, E_\gamma - E')dE'] \times \phi(E_\gamma)dE_\gamma$$

上式からコンプトン効果により発生する電流を求めるには三重積分を行う必要があり、積分範囲も書き加えて最終的な計算式を表すと次式のようにになる。なお、積分の順序は論文¹⁾内で記述されており、先に E' の積分を行ってから順に E, E_γ と計算する。

$$I_{yce} = eV \frac{SK}{4V\rho} NZ \int_{E_l(EMIN)}^{E_{max}} dE_\gamma \phi(E_\gamma) \int_{EMIN}^{E_M(E_\gamma)} dE \int_E^{E_M(E_\gamma)} dE' \sigma(E_\gamma, E_\gamma - E')$$

積分範囲にある文字及び関数の定義は以下の通りである。

E_{max} : 入射するガンマ線の最大エネルギー

$EMIN$: 電流発生に寄与するガンマ線の平均最小エネルギー

$E_l(EMIN)$: エネルギーが $EMIN$ となるコンプトン電子を生じさせるのに必要なガンマ線の最小エネルギーで次式のように定義される。

$$E_l(EMIN) = \frac{EMIN}{2} + \left\{ \left(\frac{EMIN}{2} \right)^2 + mc^2 \frac{EMIN}{2} \right\}^{1/2}$$

$E_M(E_\gamma)$: コンプトン電子の最大エネルギーであり、次式のように定義される。

$$E_M(E_\gamma) = \frac{2E_\gamma^2}{mc^2 + 2E_\gamma}$$

2.3.2 光電効果による発生電流の計算式

光電効果による発生電流の計算式を定義通りに記述すると以下ようになる。

$$dI = eV \times \left[\frac{S}{4V} \frac{K}{\rho} dE \right] \times [\mu(E_\gamma)\delta(E_\gamma - E_K - E')dE'] \times \phi(E_\gamma)dE_\gamma$$

上式から光電効果により発生する電流を求めるには三重積分を行う必要があり、積分範囲も書き加えて最終的な計算式を表すと次式のようにになる。なお、積分の順序は論文¹⁾内で記述されており、先に E' の積分を行ってから順に E, E_γ と計算する。

$$I_{ype} = eV \frac{SK}{4V\rho} \int_{EMIN+E_K}^{E_{max}} dE_\gamma \phi(E_\gamma) \mu(E_\gamma) \int_{EMIN}^{E_\gamma - E_K} dE \int_0^\infty dE' \delta(E_\gamma - E_K - E')$$

ここで、変数 E' には「放射線の入射によって生じた電子のエネルギー」という意味合いがあるため、可能性の範囲として $0 \sim \infty$ が考えられるので、積分範囲も $0 \sim \infty$ とした。これにより再度、式を整理すると次式のようにになる。

$$I_{ype} = eV \frac{SK}{4V\rho} \int_{EMIN+E_K}^{E_{max}} dE_\gamma \phi(E_\gamma) \mu(E_\gamma) (E_\gamma - E_K - EMIN)$$

2.3.3 内部転換による発生電流の計算式

内部転換による発生電流の計算式を定義通りに記述すると以下ようになる。

$$dI = eV \times \left[\frac{S}{4V\rho} \frac{K}{\rho} dE \right] \times \left[\Sigma_{it}(E_n) f(E_n) \frac{\eta_K}{1 + \eta_T} \delta(E_{it} - E_K - E') dE' \right] \times \phi(E_n) dE_n$$

上式から内部転換により発生する電流を求めるには三重積分を行う必要があるが、この式には $\Sigma_{it}(E_n) f(E_n)$ と $\phi(E_n) dE_n$ が含まれており、合わせて積分すると前節2.2節より $R_{n(it)}$ に置き換えられる。積分範囲も書き加えて最終的な計算式を表すと次式ようになる。なお、積分の順序は論文¹⁾内で記述されており、先に E' の積分を行ってから E について積分する。

$$I_{nic} = eV \frac{SK}{4V\rho} \frac{\eta_K}{1 + \eta_T} R_{n(it)} \int_{EMIN}^{E_{it}-E_K} dE \int_0^{\infty} dE' \delta(E_{it} - E_K - E')$$

$$R_{n(it)} = \int_0^{E_m} \Sigma_{it}(E_n) f(E_n) \phi(E_n) dE_n$$

ここで、変数 E' には「放射線の入射によって生じた電子のエネルギー」という意味合いがあるため、可能性の範囲として $0 \sim \infty$ が考えられるので、積分範囲も $0 \sim \infty$ とした。これにより再度、式を整理すると次式ようになる。

$$I_{nic} = eV \frac{SK}{4V\rho} \frac{\eta_K}{1 + \eta_T} R_{n(it)} (E_{it} - E_K - EMIN)$$

2.3.4 ベータ崩壊による発生電流の計算式

ベータ崩壊による発生電流の計算式を定義通りに記述すると以下ようになる。

$$dI = eV \times \left[\frac{S}{4V\rho} \frac{K}{\rho} dE \right] \times \left[\frac{105}{16} E_{\beta}^{-7/2} (E_{\beta} - E')^2 \sqrt{E'} \Sigma_{\beta}(E_n) f(E_n) dE' \right] \times \phi(E_n) dE_n$$

上式からベータ崩壊により発生する電流を求めるには三重積分を行う必要があるが、この式には $\Sigma_{\beta}(E_n) f(E_n)$ と $\phi(E_n) dE_n$ が含まれており合わせて積分すると前節2.2節より $R_{n(\beta)}$ に置き換えられる。積分範囲も書き加えて最終的な計算式を表すと次式ようになる。なお、積分の順序は論文¹⁾内で記述されており、先に E' の積分を行ってから E について積分する。

$$I_{n\beta^-} = eV \frac{SK}{4V\rho} \frac{105}{16} E_{\beta}^{-7/2} R_{n(\beta)} \int_{EMIN}^{E_{\beta}} dE \int_E^{E_{\beta}} dE' (E_{\beta} - E')^2 \sqrt{E'}$$

$$R_{n(\beta)} = \int_0^{E_m} \Sigma_{\beta}(E_n) f(E_n) \phi(E_n) dE_n$$

2.4 二段階の相互作用による発生電流の計算式

2.4.1 中性子捕獲反応－コンプトン効果による発生電流の計算式

中性子捕獲反応－コンプトン効果による発生電流の計算式を定義通りに記述すると以下のようになる。

$$I_i = eV \times \left[\frac{S}{4V} \frac{K}{\rho} dE \right] \times [NZ\bar{l}\sigma(E_{\gamma i}, E_{\gamma i} - E') dE'] \times M(E_{\gamma i}) \times R_{n(\gamma)}$$

$$R_{n(\gamma)} = \int_0^{E_m} \Sigma_{\gamma}(E_n) f(E_n) \phi(E_n) dE_n$$

上式から中性子捕獲反応－コンプトン効果により発生する電流を求めるには二重積分とその合算を行う必要があり、積分範囲も書き加えて最終的な計算式を表すと次式のようにになる。なお、積分の順序は論文¹⁾内で記述されており、先に E', E の順に積分を行ってから添字 i について総和を計算する。

$$I_{nyce} = eV \frac{SK}{4V\rho} NZ\bar{l}R_{n(\gamma)} \sum_{i=1}^n M(E_{\gamma i}) \int_{EMIN}^{E_t(E_{\gamma i})} dE \int_E^{E_t(E_{\gamma i})} dE' \sigma(E_{\gamma i}, E_{\gamma i} - E')$$

積分範囲にある関数の定義は以下の通りである。

$E_t(E_{\gamma i})$: 中性子捕獲に伴うガンマ線によって生じるコンプトン電子の最大エネルギー。

$$E_t(E_{\gamma i}) = \frac{2E_{\gamma i}^2}{mc^2 + 2E_{\gamma i}}$$

2.4.2 中性子捕獲反応－光電効果による発生電流の計算式

中性子捕獲反応－光電効果による発生電流の計算式を定義通りに記述すると以下のようになる。なお、 $R_{n(\gamma)}$ の定義は前項 2.4.1 項に記したものと同じである。

$$I_i = eV \times \left[\frac{S}{4V} \frac{K}{\rho} dE \right] \times [\mu(E_{\gamma i})\bar{l}\delta(E_{\gamma i} - E_K - E') dE'] \times M(E_{\gamma i}) \times R_{n(\gamma)}$$

上式から中性子捕獲反応－光電効果により発生する電流を求めるには二重積分とその合算を行う必要があり、積分範囲も書き加えて最終的な計算式を表すと次式のようにになる。なお、積分の順序は論文¹⁾内で記述されており、先に E', E の順に積分を行ってから添字 i について総和を計算する。

$$I_{nype} = eV \frac{SK}{4V\rho} \bar{l}R_{n(\gamma)} \sum_{i=1}^n M(E_{\gamma i}) \mu(E_{\gamma i}) \int_{EMIN}^{E_{\gamma i} - E_K} dE \int_0^{\infty} dE' \delta(E_{\gamma i} - E_K - E')$$

ここで、変数 E' には「放射線の入射によって生じた電子のエネルギー」という意味合いがあるため、可能性の範囲として $0 \sim \infty$ が考えられるので、積分範囲も $0 \sim \infty$ とした。これにより再度、式を整理すると次式のようにになる。

$$I_{nype} = eV \frac{SK}{4V\rho} \bar{l}R_{n(\gamma)} \sum_{i=1}^n M(E_{\gamma i}) \mu(E_{\gamma i}) (E_{\gamma i} - E_K - EMIN)$$

3. 数値計算コードの設計

本章では、前章で整理した計算式を計算機で実行可能な形にするためのアルゴリズムの設計について記述する。なお、本稿で数値計算コードの作成に用いるプログラミング言語はC言語とし、コンパイルと実行がすぐに行えるよう1つのソースコードファイルの中で必要な機能が完結するようにした。

3.1 全体動作

本節では数値計算コードの全体動作について記す。動作フローは Fig.3.1 に示す通りで、始めにエミッタの材質や大きさ等のパラメータを設定した後、計算に使用する変数群の宣言及び初期化を行う。その後、ガンマ線及び中性子のスペクトルや中性子捕獲断面積等、計算に必要な CSV ファイルの読み込みを行う。この CSV ファイルの中身については後述する。必要な情報が揃った後は各相互作用による発生電流の計算を行い、計算結果を出力する。

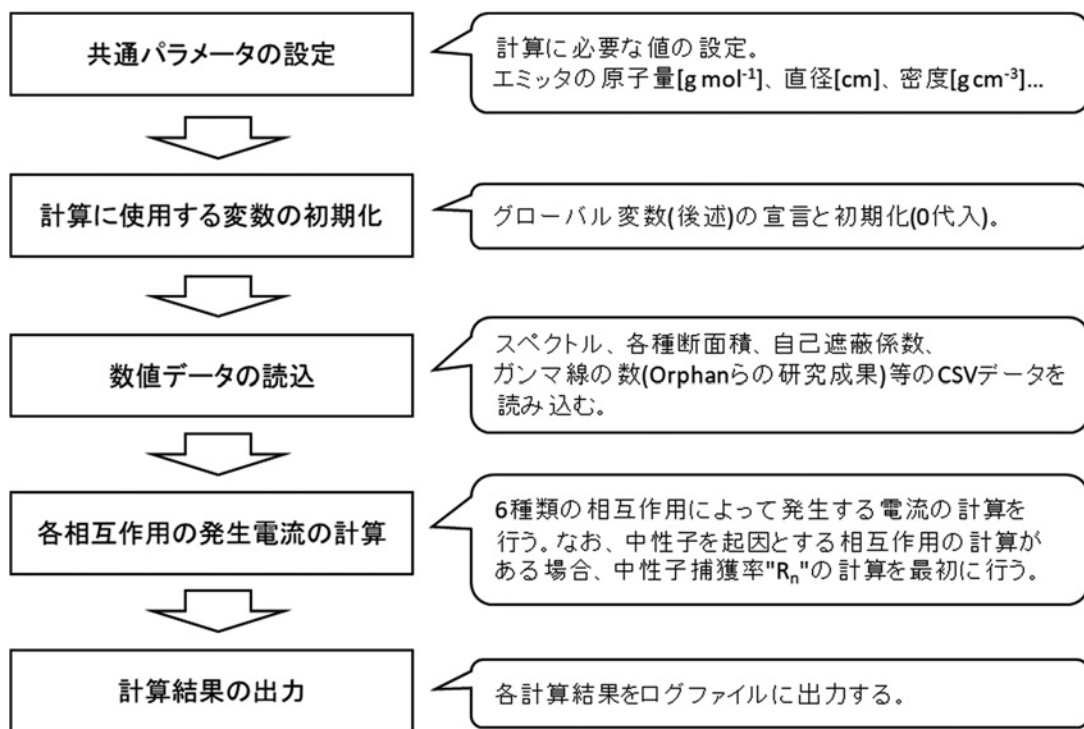


Fig.3.1 数値計算コードの全体動作フロー

3.2 グローバル変数と関数の定義

数値計算コードの始めに宣言するグローバル変数の一覧を Table 3.1 に示す。左列番号 1,2 の変数は数学的又は物理的な定数、番号 3~5 の変数はエミッタに関するパラメータで"main"関数の中で値を設定するものである。番号 6,7 の配列は入射するガンマ線及び中性子のエネルギーとスペクトルが格納され、中性子を起因とする相互作用の計算を行う場合、適宜、番号 8~12 の配列にデータを格納する必要がある。

Table 3.1 グローバル変数一覧(計算に直接関わるもの)

番号	変数名	備考
1	M_PI, N0, ec	円周率(3.14159265358979323846) アボガドロ数($6.022140857 \times 10^{23}$) 電気素量($1.6021766208 \times 10^{-19}$ [C])
2	mc2, r02, pr2	電子の静止エネルギー(511[keV]) トムソン散乱の断面積から得られる値(7.94×10^{-26} [cm ²]) 使用頻度の高い定数 pr2(=M_PI×r02)
3	gA, gP, gZ, gN	エミッタの材料に関するパラメータ 原子量[g mol ⁻¹]、密度[g cm ⁻³]、原子番号 単位体積当たりの原子数(=N0×gP/gA)[個 cm ⁻³]
4	gD, gL, gS	エミッタの大きさに関するパラメータ 直径[cm]、長さ[cm]、表面積(=π×gD×gL)[cm ²]
5	gEk, gAL, gK	K 殻電子の結合エネルギー[keV]。平均飛程[cm] エミッタの密度と線阻止能の逆数を掛けたもの。 (6.5×10^{-4} [g cm ⁻² keV ⁻¹])
6	pgx[], pgy[]	入射ガンマ線のスペクトル "pgx[]"はエネルギー(横軸)、"pgy[]"はスペクトル(縦軸)
7	pnx[], pny[]	入射中性子のスペクトル "pnx[]"はエネルギー(横軸)、"pny[]"はスペクトル(縦軸)
8	six[], siy[]	核異性体転移(内部転換)に関わる中性子捕獲断面積 "six[]"はエネルギー(横軸)、"siy[]"は中性子捕獲断面積(縦軸)
9	sbx[], sby[]	ベータ崩壊に関わる中性子捕獲断面積 "sbx[]"はエネルギー(横軸)、"sby[]"は中性子捕獲断面積(縦軸)
10	sgx[], sgy[]	ガンマ崩壊に関わる中性子捕獲断面積 "sgx[]"はエネルギー(横軸)、"sgy[]"は中性子捕獲断面積(縦軸)
11	ffx[], ffy[]	自己遮蔽係数 "ffx[]"はエネルギー(横軸)、"ffy[]"は自己遮蔽係数(縦軸)
12	mmx[], mmy[]	ガンマ線の数 "mmx[]"はエネルギー、"mmy[]"はガンマ線の数

Table 3.2 グローバル変数一覧(計算上又はプログラム上必要なもの)

番号	変数名	備考
1	npg, npn	スペクトルが格納された配列の最後尾配列番号 "npg"はガンマ線、"npn"は中性子
2	nsi, nsb, nsg	反応毎の中性子捕獲断面積が格納された配列の最後尾配列番号 "nsi"は内部転換、"nsb"はベータ崩壊、"nsg"はガンマ崩壊
3	nff, nmm	自己遮蔽係数とガンマ線数が格納された配列の最後尾配列番号 "nff"は自己遮蔽係数、"nmm"はガンマ線の数
4	sfp_i[], sfp_b[] sfp_g[]	中性子捕獲率 R_n の計算に必要なパラメータ毎のエネルギーを ソート及び統合したもの。 "sfp_i[]"は内部転換、"sfp_b[]"はベータ崩壊 "sfp_g[]"はガンマ崩壊
5	nni, nnb, nng	ソートされたエネルギーを格納した配列の最後尾配列番号 "nni"は内部転換、"nnb"はベータ崩壊、"nng"はガンマ崩壊
6	Rni, Rnb, Rng	中性子捕獲率 R_n の計算結果 "Rni"は内部転換、"Rnb"はベータ崩壊、"Rng"はガンマ崩壊
7	gCP	光電効果の計算に関わる定数
8	NN, MM	配列(フルエンス率、中性子捕獲断面積等)の上限を定める変数 "NN"は読み込んだデータを格納する配列の上限数 "MM"はソートされたエネルギーを格納する配列の上限数
9	*fp	ログファイルのファイルポインタ

Table 3.2 のグローバル変数は、計算を行う上で間接的に必要となるものである。例として "npg"は積分の上端として必要なガンマ線の最大エネルギーを参照する際、"pgx[npg]"のような使い方をする。同様に"npn","nff"等も中性子捕獲率 R_n の計算において、計算の終端を知るために必要となる。

次頁の Table 3.3 には数値計算コードを構成する関数名の一覧を示す。"main"関数は、全体の動作を管理するもので主要なパラメータの設定や実行する関数の選択を行い、基本関数群は CSV ファイルの入出力やグローバル変数の初期化、共通で利用する機能等を指す関数群である。電流計算関数群は前章で整理した各相互作用による発生電流を計算する関数群であり、補助数式関数群は電流計算関数群が繰り返し利用する複雑な数式を関数化したものである。

Table 3.3 関数一覧

番号	関数名	役割
1	main	プログラム全体の流れを制御する。
基本関数		
2	introduction	本プログラムの紹介文をログファイルに記述する。
3	init_setting	グローバル変数の初期化を行う。
4	setting_view	設定されたパラメータをログファイルに記述する。
5	file_reader	CSV ファイルの読み込みを行う。
6	file_writer	確認のため、CSV ファイルの書き出しを行う。
7	DV_funcs	与えられた変数に対応する離散データ (CSV ファイルの値) を求める。
8	sort_func	値が昇順に格納されている 2 本の配列を 1 本の配列に昇順に統合する。重複した値は 1 つの値として扱う。
電流計算関数		
9	Rn_calc	3 種類の中性子捕獲率" R_n "の計算を行う。
10	SI_GC	コンプトン効果(γ, e_{ce})による発生電流の計算を行う。
11	SI_GP	光電効果(γ, e_{pe})による発生電流の計算を行う。
12	SI_NI	内部転換(n, e_{ic})による発生電流の計算を行う。
13	SI_NB	ベータ崩壊(n, β^-)による発生電流の計算を行う。
14	TI_NC	中性子捕獲反応－コンプトン効果(n, γ, e_{ce})による発生電流の計算を行う。
15	TI_NP	中性子捕獲反応－光電効果(n, γ, e_{pe})による発生電流の計算を行う。
補助数式関数		
16	SIMPSON	シンプソン公式を用いて" $H(E_\gamma, E_{MIN})$ "の数値積分を行う。
17	H_func	後述のコンプトン効果による発生電流の計算で用いる関数。
18	I_func	後述の光電効果による発生電流の計算で用いる関数。

3.3 読み込む離散データの形式と参照方法

前章で記した計算を行うには、ガンマ線、中性子のスペクトル及び中性子捕獲断面積等のデータを読み込む必要がある。本節ではこれらのデータの読込形式と参照方法について記す。

まず、読込データは全てエネルギーを変数(横軸)としたグラフになるので、定義域はゼロ以上となり、スペクトルや中性子捕獲断面積のようにエネルギーに対応付けられる値(縦軸)も基本的にゼロ以上となる。つまり Fig.3.2 に示すようなデータとグラフを想定して読込と参照の機能を考える必要がある。

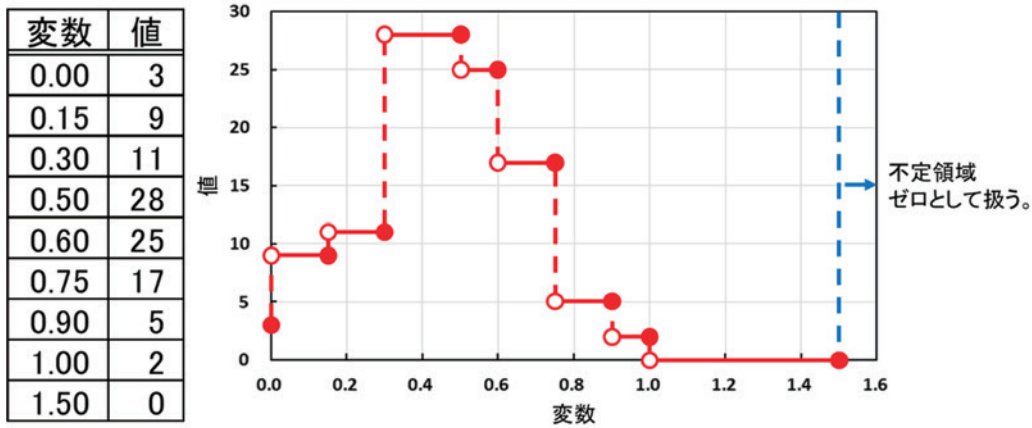


Fig.3.2 データとグラフの対応関係例

ここで、図中左の表には変数とそれに対応する値が記されており、未定義の変数に対する値は、その変数が含まれる区間の上端の変数に対応する値に等しいものとする。具体例として上図の場合、変数が"0.01"ならば値は"9"、変数が"0.49"ならば値は"28"、変数が"2.0"ならば値は"0"となる。また、変数が負の場合は不定領域と同様に値を"0"として扱う。

読み込むデータの形式は CSV で、文字コードは ASCII や SHIFT_JIS 等の 1 バイト文字コードとし、Windows 標準のテキストエディタで開くと Fig.3.3 のようになる。本計算コードでは、必要な情報をこのような形の CSV ファイルから読み込み、各計算に利用する。

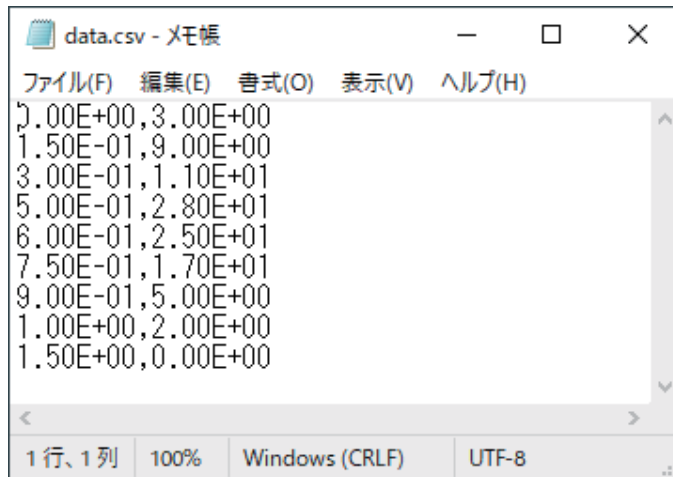


Fig.3.3 読込データファイル

読込データの参照には離散値関数(Discrete Value functions) "DV_funcs"を用いる。この関数は与えられた変数(エネルギー)に対応する値とその値が格納された配列の配列番号を返却する。動作の概略は Fig.3.4 のフローチャートに示す通りで非効率的な探索アルゴリズムではあるが、計算効率よりも実装の単純化を優先しバグの混入が起きにくくなるようにした。

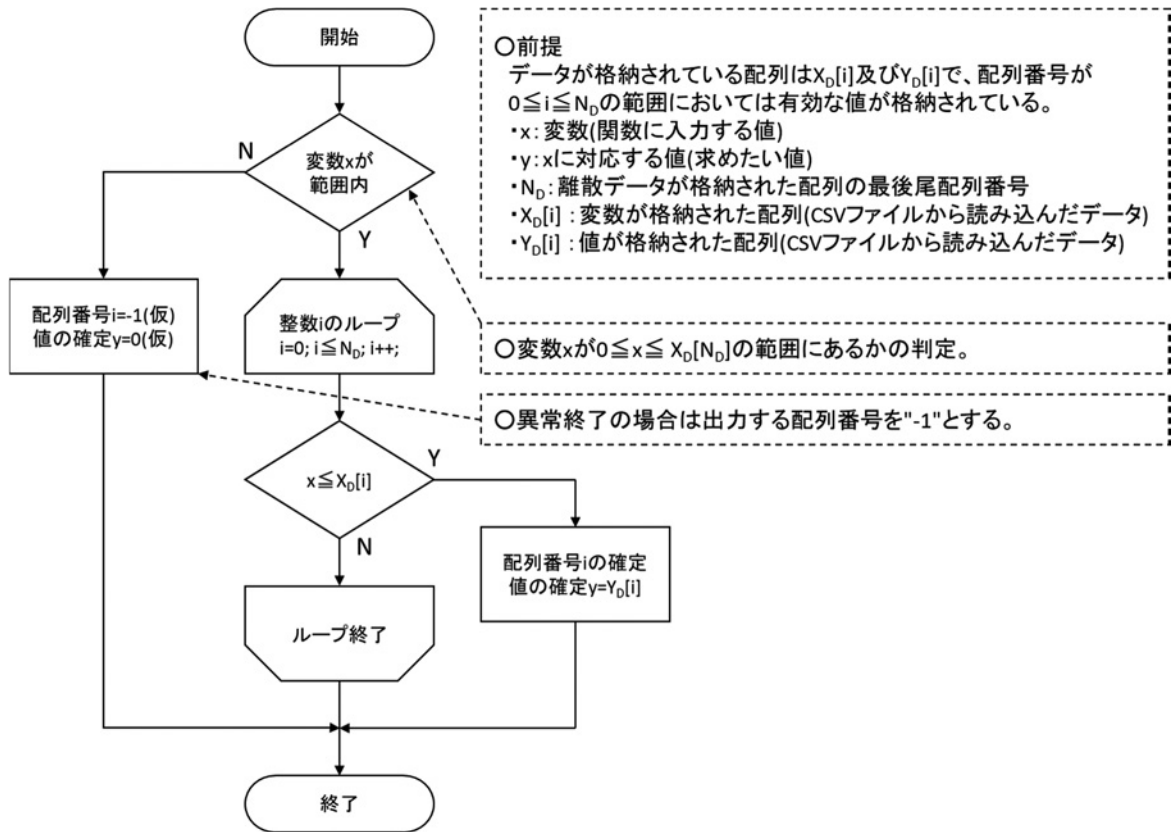


Fig.3.4 離散値関数のフローチャート

3.4 中性子捕獲率" R_n "の計算方法

中性子捕獲率" R_n "の計算式は前章の 2.2 節の終わりで定義されているが、式中の 3 つの項の各値は CSV データの読込によって作成された配列の参照によって求められる。従って、この積分は、あるエネルギーの点列 $\{E_j\}$ が与えられた場合、次式のように書き換える事ができる。

$$R_n = \int_0^{E_m} \Sigma(E_n) f(E_n) \phi(E_n) dE_n = \sum_{j=1} \Sigma(E_j) f(E_j) \phi(E_j) \cdot (E_j - E_{j-1})$$

ここで、点列 $\{E_j\}$ は 3 つの項の積 $\Sigma(E_n) f(E_n) \phi(E_n)$ の値が変化する時のエネルギーの値の集合で、模式図で表現すると Fig.3.5 中の $\{E^{(\Sigma f \phi)}_i\}$ にあたる。本計算コードでは、共通で利用する $f(E_j)$ と $\phi(E_j)$ のエネルギーの点列 $\{E^{(f)}_i\}$ 及び $\{E^{(\phi)}_i\}$ をマージソートで統合し、その後、核異性体転移(内部転換) $\Sigma_{it}(E_j)$ 、ベータ崩壊 $\Sigma_{\beta}(E_j)$ 、ガンマ崩壊 $\Sigma_{\gamma}(E_j)$ 毎に、先のソート結果と統合した点列を用意する。用意した点列はそれぞれグローバル配列"sfp_i[]", "sfp_b[]", "sfp_g[]" (Table 3.2 参照)に格納される。この操作により上式の計算が実行可能となる。

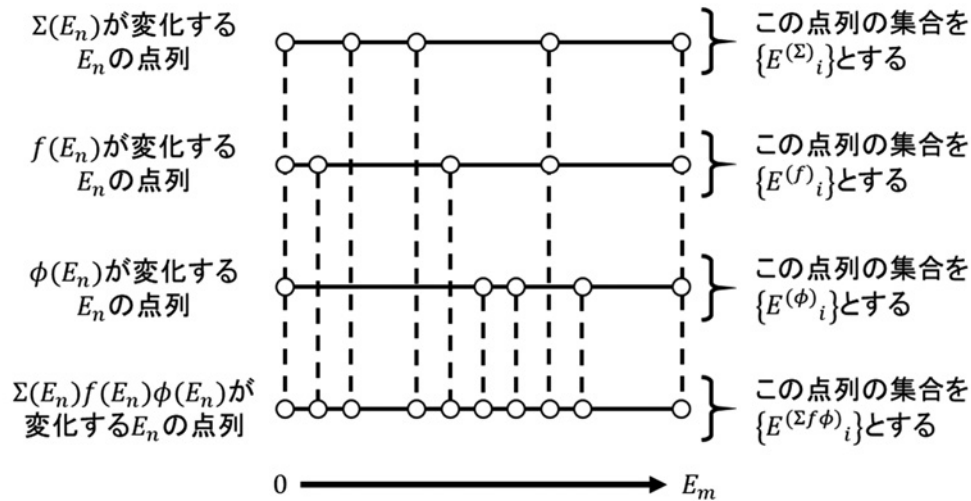


Fig.3.5 中性子捕獲率"R_n"の計算に関わるエネルギーの点列

3.5 一段階の相互作用による発生電流の計算方法

3.5.1 コンプトン効果による発生電流の計算方法

コンプトン効果による発生電流の計算式は前章 2.3.1 項で整理した内容から次式のように表される。

$$I_{yce} = eV \frac{SK}{4V\rho} NZ \int_{E_l(EMIN)}^{E_{max}} dE_\gamma \phi(E_\gamma) \int_{EMIN}^{E_M(E_\gamma)} dE \int_E^{E_M(E_\gamma)} dE' \sigma(E_\gamma, E_\gamma - E')$$

式中の $\sigma(E_\gamma, E_\gamma - E')$ は E', E に関しては初等関数の範囲で積分可能であるが、 E_γ の積分については $\phi(E_\gamma)$ を除いたとしても初等関数で表現する事は不可能である。従って、本計算コードでは始めの二重積分に関しては解析的に解き、その後シンプソン公式に従って数値積分を行う。

始めの E' に関する定積分を求めるため、 $\sigma(E_\gamma, E_\gamma - E')$ の不定積分(積分定数省略)を求めると以下のようなになる。今後、簡単のためこの式を $F(E_\gamma, E')$ とおく。

$$\begin{aligned} F(E_\gamma, E') &= \int dE' \sigma(E_\gamma, E_\gamma - E') \\ &= \pi r_0^2 \frac{mc^2}{E_\gamma^2} \left\{ -E_\gamma \ln(E_\gamma - E') + E' - \frac{E'^2}{2E_\gamma} \right\} + \pi r_0^2 \frac{(mc^2)^3}{E_\gamma^2} \left\{ \frac{1}{E_\gamma - E'} + \frac{2}{E_\gamma} \ln(E_\gamma - E') + \frac{E'}{E_\gamma^2} \right\} \\ &\quad - 2\pi r_0^2 \frac{(mc^2)^2}{E_\gamma^2} \left\{ -\ln(E_\gamma - E') - \frac{E'}{E_\gamma} \right\} \end{aligned}$$

上式を基に、 E', E に関する二重定積分を整理すると以下のように表す事ができる。

$$\begin{aligned} \int_{EMIN}^{E_M(E_\gamma)} dE \int_E^{E_M(E_\gamma)} dE' \sigma(E_\gamma, E_\gamma - E') &= \int_{EMIN}^{E_M(E_\gamma)} dE \left\{ F(E_\gamma, E_M(E_\gamma)) - F(E_\gamma, E) \right\} \\ &= F(E_\gamma, E_M(E_\gamma)) \{E_M(E_\gamma) - EMIN\} - \int_{EMIN}^{E_M(E_\gamma)} dE F(E_\gamma, E) \end{aligned}$$

ここで、 $F(E_\gamma, E)$ の定積分が出てくるので、さらに不定積分(積分定数省略)を行うと以下のようになる。先程と同様に簡単のためこの式を $G(E_\gamma, E)$ とおく。

$$\begin{aligned}
 G(E_\gamma, E) &= \int dE F(E_\gamma, E) \\
 &= \pi r_0^2 \frac{mc^2}{E_\gamma^2} \left\{ E_\gamma(E_\gamma - E) \ln(E_\gamma - E) + E_\gamma E + \frac{E^2}{2} - \frac{E^3}{6E_\gamma} \right\} \\
 &\quad + \pi r_0^2 \frac{(mc^2)^3}{E_\gamma^2} \left\{ -\ln(E_\gamma - E) - \frac{2(E_\gamma - E)}{E_\gamma} \ln(E_\gamma - E) - \frac{2E}{E_\gamma} + \frac{E^2}{2E_\gamma^2} \right\} \\
 &\quad - 2\pi r_0^2 \frac{(mc^2)^2}{E_\gamma^2} \left\{ (E_\gamma - E) \ln(E_\gamma - E) + E - \frac{E^2}{2E_\gamma} \right\}
 \end{aligned}$$

以上から E', E に関する二重定積分は次のように表す事ができ、簡単のためこの式を $H(E_\gamma, EMIN)$ とおく。

$$H(E_\gamma, EMIN) = F(E_\gamma, E_M(E_\gamma)) \{E_M(E_\gamma) - EMIN\} - \{G(E_\gamma, E_M(E_\gamma)) - G(E_\gamma, EMIN)\}$$

上式を用いて、 I_{yce} の式を整理すると次のようになる。

$$\begin{aligned}
 I_{yce} &= eV \frac{SK}{4V\rho} NZ \int_{E_l(EMIN)}^{E_{max}} dE_\gamma \phi(E_\gamma) H(E_\gamma, EMIN) \\
 &= eV \frac{SK}{4V\rho} NZ \left\{ \phi(E_\gamma[m]) \int_{E_l(EMIN)}^{E_\gamma[m]} dE_\gamma H(E_\gamma, EMIN) + \sum_{j=m+1}^{N_{max}} \phi(E_\gamma[j]) \int_{E_\gamma[j-1]}^{E_\gamma[j]} dE_\gamma H(E_\gamma, EMIN) \right\}
 \end{aligned}$$

ここで、式中の新たな文字の定義は以下に示す通りで、これらの概念を図示すると Fig.3.6 のようになる。また、積分領域を分割した理由は、 $\phi(E_\gamma)$ の参照回数を減らし効率的に計算を行うためである。

$E_\gamma[m]$: ガンマ線スペクトルのエネルギーの点列を格納した配列。

m : $E_l(EMIN) \leq E_\gamma[m]$ を満たす最小の整数(配列番号)。

N_{max} : E_γ を格納した配列の最後尾配列番号。Table 3.2 に記した"npg"に対応する変数。

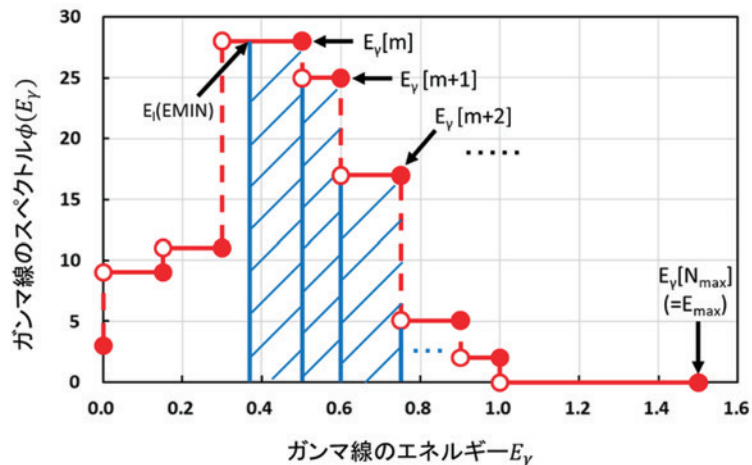


Fig.3.6 積分範囲分割のイメージ

この計算のフローチャートを Fig.3.7 に示す。なお、図中の(5),(8)では関数呼出を行っているが、これはシンプソン公式を用いた定積分を算出する関数"SIMPSON" (Table 3.3 参照) を呼び出して、上端・下端及び E_γ の刻み幅 ΔE_γ を指定して計算結果を受け取る動作を行っている。また、(3)の配列番号の特定には"DV_funcs" (本章 3.3 節参照) を用いている。

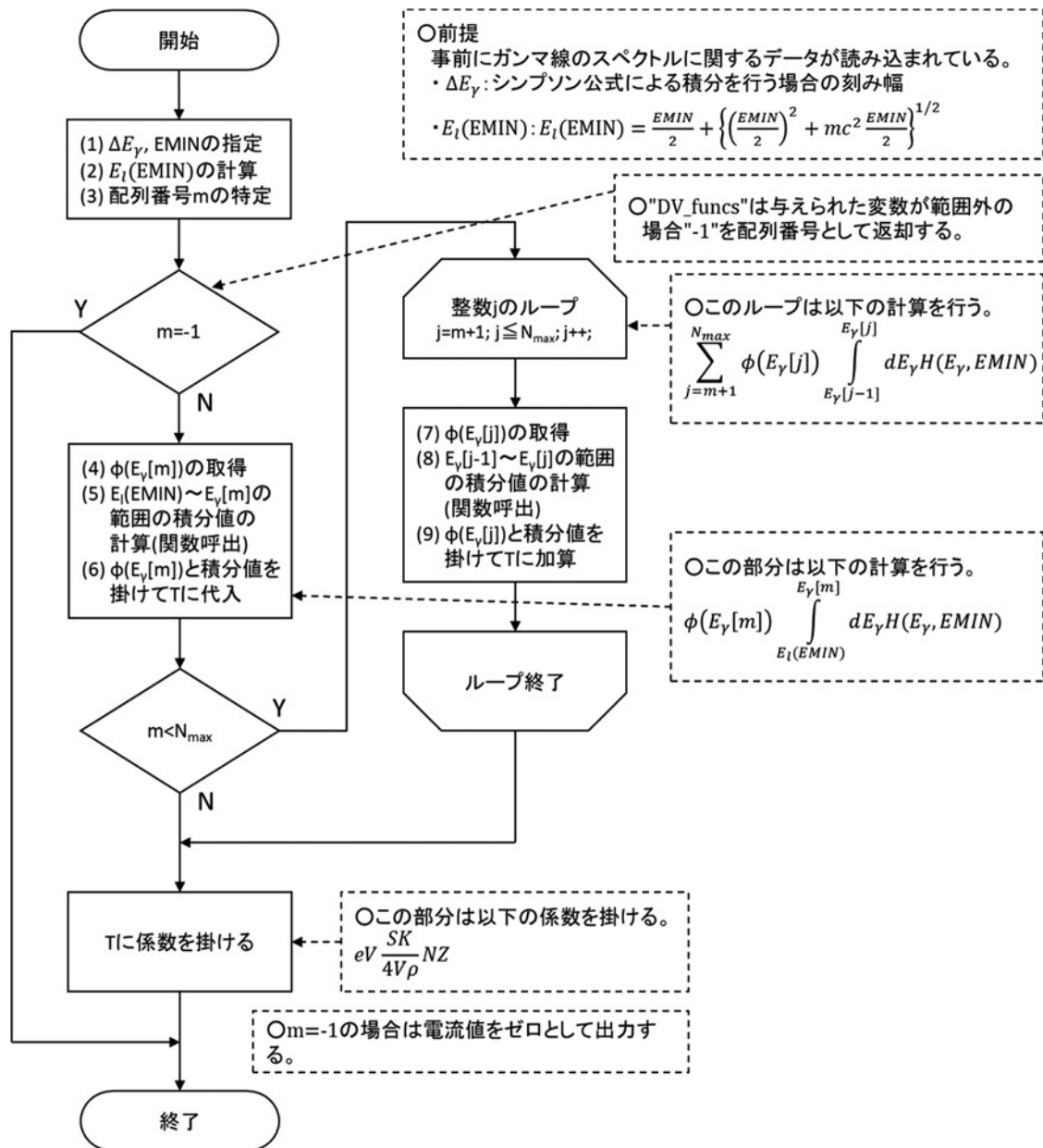


Fig.3.7 コンプトン効果による発生電流の計算フローチャート

3.5.2 光電効果による発生電流の計算方法

光電効果による発生電流の計算式は前章 2.3.2 項で整理した内容から次式のように表される。

$$I_{\gamma pe} = eV \frac{SK}{4V\rho} \int_{EMIN+E_K}^{E_{max}} dE_{\gamma} \phi(E_{\gamma}) \mu(E_{\gamma}) (E_{\gamma} - E_K - EMIN)$$

ここで、式中の $\phi(E_{\gamma})$ は離散的な値であり、 E_{γ} に対して区間毎に一定値を取る関数である。従って、上式は次のように書き換える事が可能となる。

$$\begin{aligned} I_{\gamma pe} &= eV \frac{SK}{4V\rho} \left\{ \frac{40\sqrt{2}\pi r_0^2 N Z^5 (mc^2)^{7/2}}{3 \cdot 137^4} \right\} \int_{EMIN+E_K}^{E_{max}} dE_{\gamma} \phi(E_{\gamma}) E_{\gamma}^{-7/2} (E_{\gamma} - E_K - EMIN) \\ &= eV \frac{SK}{4V\rho} \left\{ \frac{40\sqrt{2}\pi r_0^2 N Z^5 (mc^2)^{7/2}}{3 \cdot 137^4} \right\} \sum \phi(E_{\gamma}) \int dE_{\gamma} E_{\gamma}^{-7/2} (E_{\gamma} - E_K - EMIN) \end{aligned}$$

上式の E_{γ} に関する定積分は解析的に求める事ができるので、これを $I(A, B, EMIN)$ とにおいて定積分を求めると以下のようなになる。

$$I(A, B, EMIN) = \int_A^B dE_{\gamma} E_{\gamma}^{-7/2} (E_{\gamma} - E_K - EMIN) = \left[-\frac{2}{3} E_{\gamma}^{-3/2} + \frac{2}{5} (E_K + EMIN) E_{\gamma}^{-5/2} \right]_{E_{\gamma}=A}^{E_{\gamma}=B}$$

従って、 $I_{\gamma pe}$ の式は整理すると以下のようなになる。

$$\begin{aligned} I_{\gamma pe} &= eV \frac{SK}{4V\rho} \left\{ \frac{40\sqrt{2}\pi r_0^2 N Z^5 (mc^2)^{7/2}}{3 \cdot 137^4} \right\} \sum \phi(E_{\gamma}) \int dE_{\gamma} E_{\gamma}^{-7/2} (E_{\gamma} - E_K - EMIN) \\ &= eV \frac{SK}{4V\rho} \left\{ \frac{40\sqrt{2}\pi r_0^2 N Z^5 (mc^2)^{7/2}}{3 \cdot 137^4} \right\} \phi(E_{\gamma}[m]) I(EMIN + E_K, E_{\gamma}[m], EMIN) \\ &\quad + eV \frac{SK}{4V\rho} \left\{ \frac{40\sqrt{2}\pi r_0^2 N Z^5 (mc^2)^{7/2}}{3 \cdot 137^4} \right\} \sum_{j=m+1}^{N_{max}} \phi(E_{\gamma}[j]) I(E_{\gamma}[j-1], E_{\gamma}[j], EMIN) \end{aligned}$$

ここで、式中の新たな文字の定義は以下の通りである。

$E_{\gamma}[m]$: ガンマ線スペクトルのエネルギーの点列を格納した配列。

m : $EMIN + E_K \leq E_{\gamma}[m]$ を満たす最小の整数(配列番号)。

N_{max} : E_{γ} を格納した配列の最後尾配列番号。Table 3.2 に記した"npg"に対応する変数。

この計算のフローチャートを Fig.3.8 に示す。なお、図中の(4),(7)では関数呼出を行っているが、これは定積分を算出する関数"I_func" (Table 3.3 参照) を呼び出して、上端及び下端を指定して計算結果を受け取る動作を行っている。また、(2)の配列番号mの特定には"DV_funcs" (本章 3.3 節参照) を用いている。

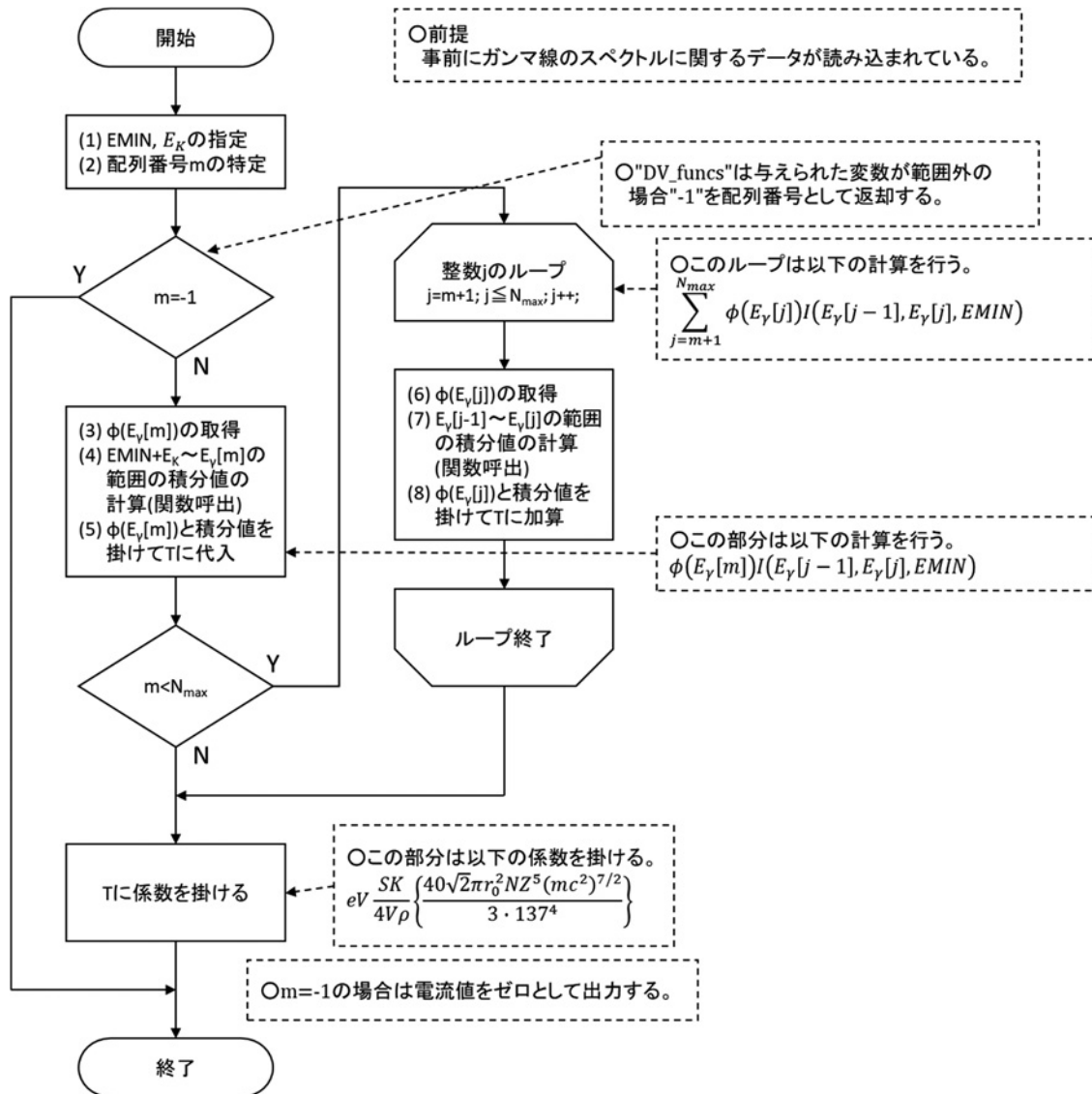


Fig.3.8 光電効果による発生電流の計算フローチャート

3.5.3 内部転換及びベータ崩壊による発生電流の計算方法

内部転換による発生電流の計算式は前章 2.3.3 項で整理した内容から次式のように表される。

$$I_{nic} = eV \frac{SK}{4V\rho} \frac{\eta_K}{1 + \eta_T} R_{n(it)} (E_{it} - E_K - EMIN)$$

内部転換により生じる電流の計算では $R_{n(it)}$ の計算以外には代入操作だけで計算結果が求められるが、 $(E_{it} - E_K - EMIN) < 0$ の場合に計算結果をゼロとする処理を作成する必要がある。

また、前章 2.3.4 項で整理したベータ崩壊による発生電流の計算式には二重定積分が含まれているが、この部分は解析的に求める事が可能であり、最終的には以下のように整理する事ができる。この場合も $R_{n(\beta)}$ の計算に関しては内部転換の場合と同様で、後は代入操作だけで計算結果が求められる。なお、ベータ崩壊の場合は、 $(E_\beta - EMIN) < 0$ の時に計算結果をゼロとする処理を作成する必要がある。

$$\begin{aligned} I_{n\beta^-} &= eV \frac{SK}{4V\rho} \frac{105}{16} E_\beta^{-7/2} R_{n(\beta)} \int_{EMIN}^{E_\beta} dE \int_E^{E_\beta} dE' (E_\beta - E')^2 \sqrt{E'} \\ &= eV \frac{SK}{4V\rho} R_{n(\beta)} \left[\frac{1}{3} E_\beta - (EMIN) + \frac{(EMIN)^{5/2}}{12E_\beta^{3/2}} \left\{ 21 - 18 \frac{(EMIN)}{E_\beta} + 5 \frac{(EMIN)^2}{E_\beta^2} \right\} \right] \end{aligned}$$

3.6 二段階の相互作用による発生電流の計算方法

中性子捕獲反応－コンプトン効果による発生電流の計算式は前章 2.4.1 項で整理した内容から次式のように表される。

$$I_{nyce} = eV \frac{SK}{4V\rho} NZ \bar{I} R_{n(\gamma)} \sum_{i=1}^n M(E_{\gamma i}) \int_{EMIN}^{E_t(E_{\gamma i})} dE \int_E^{E_t(E_{\gamma i})} dE' \sigma(E_{\gamma i}, E_{\gamma i} - E')$$

上式の二重定積分の上端 $E_t(E_{\gamma i})$ は $E_M(E_\gamma)$ と関数形が同じなので、前項 3.5.1 項で示した関数 $H(E_\gamma, EMIN)$ を用いて次式のように書き換える事ができる。

$$I_{nyce} = eV \frac{SK}{4V\rho} NZ \bar{I} R_{n(\gamma)} \sum_{i=1}^n M(E_{\gamma i}) H(E_{\gamma i}, EMIN)$$

また、中性子捕獲反応－光電効果による発生電流の計算式は次式のように表されるので、最終的に両者とも単純に足し上げていく事で値が求められる。

$$\begin{aligned} I_{nype} &= eV \frac{SK}{4V\rho} \bar{I} R_{n(\gamma)} \sum_{i=1}^n M(E_{\gamma i}) \mu(E_{\gamma i}) (E_{\gamma i} - E_K - EMIN) \\ &= eV \frac{SK}{4V\rho} \left\{ \frac{40\sqrt{2}\pi r_0^2 NZ^5 (mc^2)^{7/2}}{3 \cdot 137^4} \right\} \bar{I} R_{n(\gamma)} \sum_{i=1}^n M(E_{\gamma i}) E_{\gamma i}^{-7/2} (E_{\gamma i} - E_K - EMIN) \end{aligned}$$

4. 数値計算コードの実行結果

本章では、前章で記したアルゴリズムを用いて作成した計算コードの実行結果について記す。計算コードの信頼性は、論文¹⁾に記されている表やグラフと計算結果を比較して評価する。

4.1 読込データの準備

本節では計算に際し必要となるガンマ線のスペクトルデータ $\phi(E_\gamma)$ の作成方法について記す。具体的な作業としては論文に記されているプール型原子炉、PWR 炉心、コバルト 60 線源のスペクトルを表すグラフを拡大して 1 つ 1 つ値を読み取り、次式の制約条件を満たすよう値の調整を行った後、CSV ファイルに書き出す。

$$\int_0^{E_{\gamma max}} \phi(E_\gamma) dE_\gamma = 1 \text{ [個 sec}^{-1} \text{ cm}^{-2}]$$

従って、作成したデータから再現されるスペクトルのグラフ形状は、論文のグラフと完全には一致しない。特に、プール型原子炉と PWR 炉心のスペクトルについてはグラフ形状が複雑であり、読み取り誤差が計算結果に強く影響すると考えられる。

一方、コバルト 60 線源については 1173.2[keV]と 1332.5[keV]の二種類のガンマ線を放射するので、それぞれを中心に 50[keV]ずつ幅を持たせたスペクトルデータを用意した。これは論文のグラフから下式が成り立つように値を求めれば、 $\Delta E_\gamma = 0.05[\text{MeV}] = 50[\text{keV}]$ となるので、これによりコバルト 60 線源のスペクトルデータは完全に一致していると思われる。

$$2 \times \Delta E_\gamma [\text{MeV}] \times 10 [\text{cm}^{-2} \text{ sec}^{-2} \text{ MeV}^{-1}] = 1 [\text{cm}^{-2} \text{ sec}^{-2}]$$

Fig.4.1 にスペクトルデータを格納した CSV ファイルの中身を、Fig.4.2～Fig.4.4 にこれらをグラフ化したものを示す。

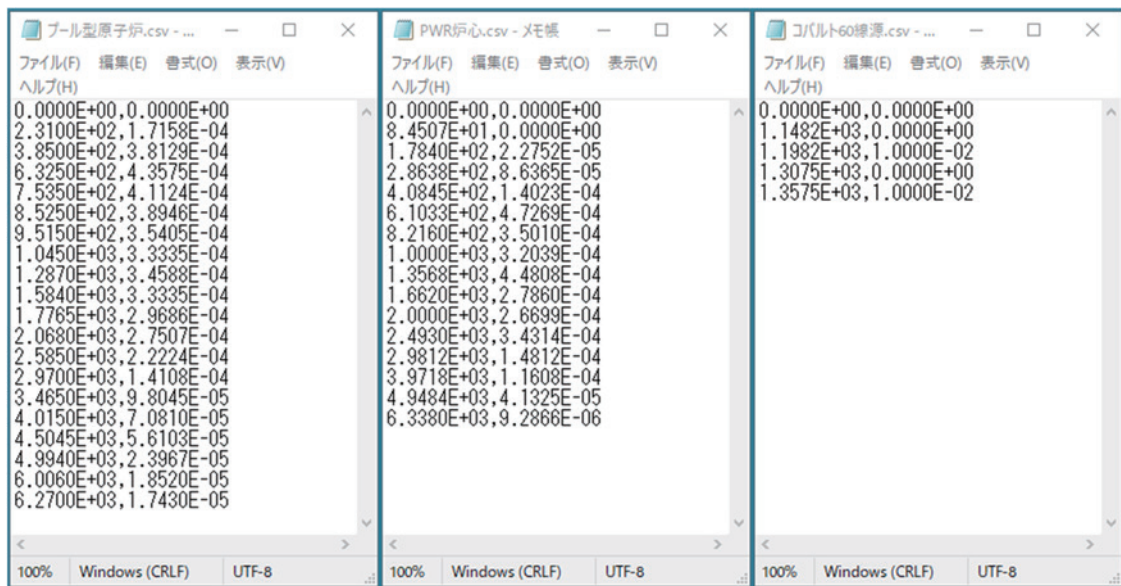


Fig.4.1 ガンマ線スペクトルの CSV ファイル
(左：プール型原子炉、中：PWR 炉心、右：コバルト 60 線源)

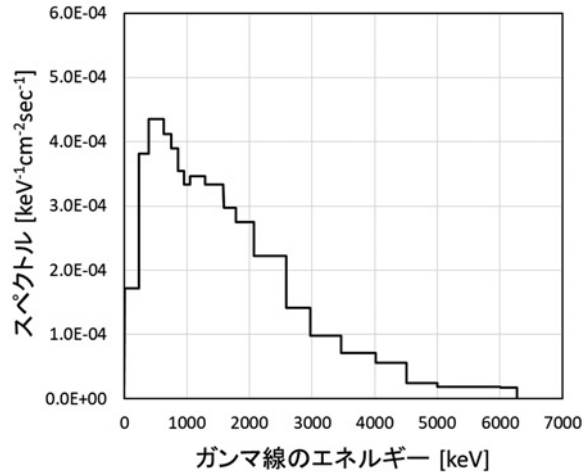


Fig.4.2 プール型原子炉のガンマ線スペクトル

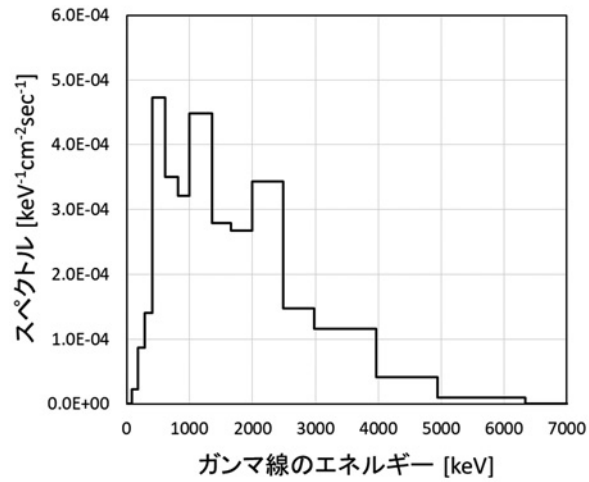


Fig.4.3 PWR 炉心のガンマ線スペクトル

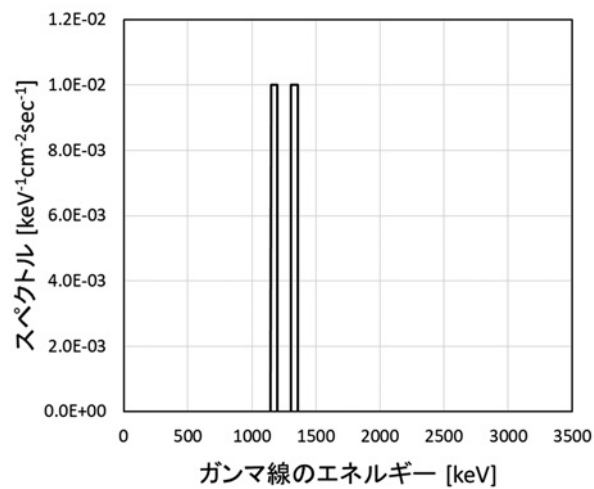


Fig.4.4 コバルト 60 線源のガンマ線スペクトル

4.2 計算結果の比較と考察

本節では作成した数値計算コードを用いて、論文¹⁾に記されている表及びグラフと同様な結果が得られるか評価した内容について記す。まず、論文ではエミッタの材質や形状に依存しない関数が定義され、この関数を用いて計算結果の整理が行われている事から、本稿でもこれに倣って結果の整理を行う。コンプトン効果による発生電流に対しては $T(E_\gamma, EMIN)$ 、光電効果による発生電流に対しては $D(E_\gamma, EMIN)$ という関数が定められており、以下のように表される。

$$T(E_\gamma, EMIN) = \left(\frac{A}{D \cdot L \cdot K \cdot Z} \right) I_{yce} \quad D(E_\gamma, EMIN) = \left(\frac{A}{D \cdot L \cdot K \cdot Z^5} \right) I_{ype}$$

ここで、各右辺の括弧中の文字の定義は以下の通りである。

A : 原子量 Z : 原子番号 D : エミッタの直径[cm] L : エミッタの長さ[cm]
 K : エミッタの密度と線阻止能の逆数を掛けたもの($6.5 \times 10^{-4} [\text{g cm}^{-2} \text{keV}^{-1}]$)

以上の関数を用いると材質や形状の違いを考慮する必要はなくなるが、本稿で作成した計算コードの都合上、適当にエミッタの条件を以下のように与えて計算を行った。なお、コンプトン効果による発生電流の計算において、シンプソン公式の微小区間幅は 1.0[keV]としている。

材質 : イッテルビウム 原子番号(Z) : 70 原子量(A) : 171
 直径(D) : 1.27[cm] 長さ(L) : 1.00[cm] 密度 : 6.90[g cm^{-3}]

Fig.4.5 に $T(E_\gamma, EMIN)$ のグラフ、Fig.4.6 に $D(E_\gamma, EMIN)$ のグラフを示し、Table 4.1 及び Table 4.2 にそれらの値を示す。

3種類のスペクトルに対して、 $T(E_\gamma, EMIN)$ については、論文に記された値と良く一致した結果が得られており、 $D(E_\gamma, EMIN)$ についても $EMIN$ が 300[keV]以上の範囲においては良く一致した結果が得られた。

ここで、計算結果が完全に一致しない理由として主に以下の要因が考えられる。

- (1) 用意したスペクトルデータの誤差。
- (2) 計算機環境の違い。(例：単精度・倍精度、打切り誤差等)
- (3) アルゴリズムの違い。(例：数式処理、数値積分の方法)

最も影響が大きいのは(1)で、プール型原子炉と PWR 炉心のスペクトルデータはグラフの読み取りにより作成したものなので、完全に誤差を取り除くことができない。次に(2)の影響が考えられ、浮動小数点演算の有効桁数の違い等が最終的な計算結果に差を生じさせたと考えられる。最後に(3)であるが、論文にはアルゴリズムの詳細については記載されていないので、本稿で作成した計算コードと詳細な比較をする事ができない。

また、 $T(E_\gamma, EMIN)$ の結果の比較では論文との差はほぼ一定であるが、 $D(E_\gamma, EMIN)$ については $EMIN$ が 250[keV]以下の場合で差が大きくなっている。

この要因としては、 $T(E_\gamma, EMIN)$ と $D(E_\gamma, EMIN)$ では $EMIN$ が同じ値の場合でも互いに E_γ についての積分範囲が異なる事、また、スペクトルデータ作成時の E_γ の読み取り誤差の影響度合いが異なる事が考えられる。

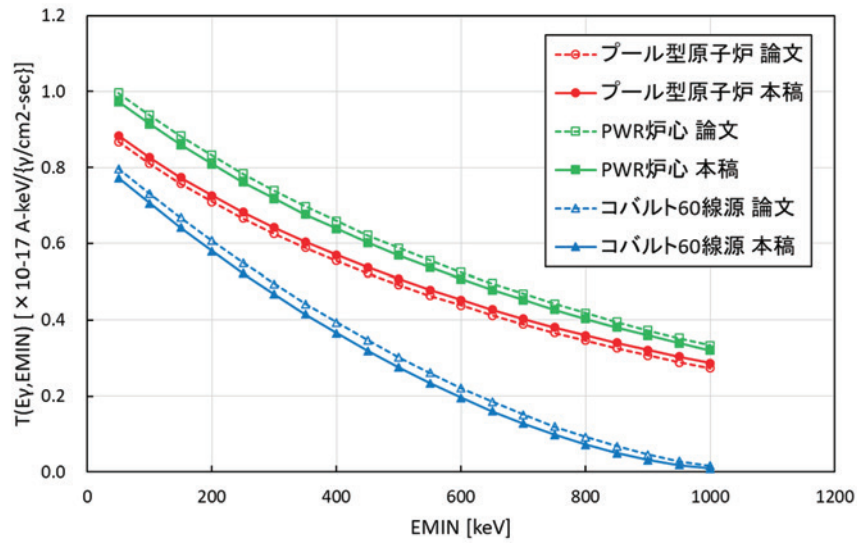


Fig.4.5 $T(E_\gamma, E_{MIN})$ の比較

Table 4.1 $T(E_\gamma, E_{MIN})[\times 10^{-17} \text{ A}\cdot\text{keV}\{\gamma/\text{cm}^2\cdot\text{sec}\}]$ の結果

EMIN [keV]	プール型原子炉		PWR 炉心		コバルト 60 線源	
	論文	本稿	論文	本稿	論文	本稿
50	0.868	0.884	0.996	0.973	0.797	0.772
100	0.810	0.826	0.937	0.914	0.731	0.705
150	0.757	0.773	0.883	0.859	0.668	0.641
200	0.710	0.726	0.832	0.809	0.608	0.581
250	0.666	0.682	0.784	0.761	0.550	0.522
300	0.626	0.642	0.739	0.717	0.495	0.467
350	0.589	0.605	0.698	0.677	0.443	0.415
400	0.555	0.570	0.659	0.639	0.394	0.366
450	0.522	0.538	0.622	0.603	0.347	0.319
500	0.492	0.507	0.588	0.569	0.302	0.275
550	0.464	0.479	0.556	0.537	0.261	0.234
600	0.438	0.452	0.525	0.507	0.221	0.196
650	0.412	0.427	0.496	0.479	0.185	0.160
700	0.389	0.403	0.469	0.452	0.151	0.128
750	0.367	0.381	0.443	0.427	0.120	0.098
800	0.346	0.360	0.418	0.403	0.092	0.072
850	0.326	0.340	0.395	0.380	0.068	0.050
900	0.307	0.321	0.373	0.359	0.046	0.031
950	0.289	0.303	0.352	0.339	0.029	0.018
1000	0.273	0.287	0.333	0.320	0.016	0.010

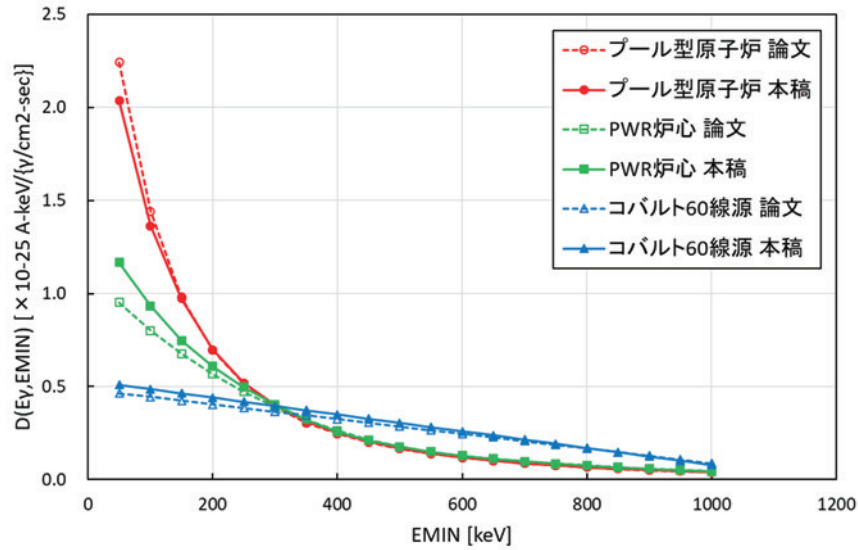


Fig.4.6 $D(E_\gamma, E_{MIN})$ の比較

Table 4.2 $D(E_\gamma, E_{MIN})[\times 10^{-25} \text{ A}\cdot\text{keV}/\{\gamma/\text{cm}^2\cdot\text{sec}\}]$ の結果

EMIN [keV]	プール型原子炉		PWR 炉心		コバルト 60 線源	
	論文	本稿	論文	本稿	論文	本稿
50	2.241	2.035	0.951	1.167	0.464	0.509
100	1.442	1.363	0.801	0.933	0.444	0.486
150	0.980	0.972	0.676	0.747	0.424	0.464
200	0.695	0.697	0.566	0.609	0.404	0.441
250	0.516	0.520	0.472	0.497	0.385	0.418
300	0.394	0.400	0.392	0.403	0.365	0.396
350	0.305	0.314	0.323	0.322	0.345	0.373
400	0.246	0.251	0.262	0.258	0.325	0.350
450	0.200	0.204	0.213	0.210	0.306	0.328
500	0.166	0.169	0.177	0.175	0.286	0.305
550	0.139	0.141	0.150	0.149	0.266	0.282
600	0.118	0.119	0.129	0.128	0.246	0.260
650	0.101	0.102	0.112	0.111	0.227	0.237
700	0.087	0.088	0.098	0.097	0.207	0.215
750	0.076	0.076	0.086	0.085	0.187	0.192
800	0.066	0.066	0.076	0.075	0.167	0.169
850	0.058	0.058	0.067	0.066	0.148	0.147
900	0.051	0.051	0.059	0.058	0.128	0.124
950	0.045	0.045	0.052	0.051	0.108	0.101
1000	0.040	0.040	0.046	0.045	0.088	0.079

5. まとめ

本稿では、自己出力型放射線検出器の出力電流値に対する理論的な評価を行うための準備として、C言語を用いた数値計算コードを作成した。この数値計算コードの作成に際しては、自己出力型放射線検出器の理論的な考察がまとめられた論文¹⁾を基に、計算式の整理や必要な機能及びアルゴリズムの検討/設計を行った。

計算コードの妥当性の確認については、論文¹⁾で示されている計算条件とほぼ同じ条件で数値計算を行い、得られた値を比較した。ここで、比較したのは一段階の相互作用の内、コンプトン効果及び光電効果による発生電流の計算結果のみで、他の相互作用については中性子スペクトル等の情報が不足していたため比較する事はできなかった。比較した二種類の相互作用については、論文の値とほぼ一致しており、これにより作成した計算コードは十分使用に堪え得るものであると考えられる。

この数値計算コードによって、過去に照射試験が行われた SPND や SPGD の照射試験結果を評価する事や多様な放射線場(スペクトル)における自己出力型放射線検出器の出力電流値を推定する事が可能となった。

謝辞

本稿では、これまで照射試験が行われてきた自己出力型放射線検出器の出力電流値の妥当性を評価するため、或いはこれから製作する検出器の放射線場毎の出力特性を予測するために数値計算コードを作成した。

本稿及び計算コードの作成に際して貴重なご意見を頂きました石塚悦男高温工学試験研究炉部 HTTR 技術課長、土谷邦彦材料試験炉部長、また、原子力機構を退職されました荒克之先生には深く感謝するとともに関わられた皆様に謹んで謝意を表します。

参考文献

- 1) H.D.Warren, N.H.Shah, Neutron and Gamma-Ray Effects on Self-Powered In-Core Radiation Detectors, Nuclear Science and Engineering, vol.54, no.4, (1974), pp.395-415.
- 2) V.J.Orphan, N.C.Rasmussen, T.L.Harper, Line and Continuum Gamma-Ray Yields from Thermal-Neutron Capture in 75 Elements, Gulf Energy & Environmental Systems, DASA2570, (1970), 680p.
- 3) 柴田徳思 編, 放射線概論 第8版, 通商産業研究社, ISBN978-4-86045-108-0, (2012), 859p.
- 4) 皆本晃弥, C 言語による数値計算入門, サイエンス社, ISBN4-7819-1114-5, (2005), 232p.
- 5) 久保田光一, 工学基礎 数値解析とその応用, 数理工学社, ISBN978-4-901683-57-9, (2010), 218p.
- 6) 柴田望洋, 新 明解 C 言語入門編, ソフトバンククリエイティブ, ISBN4-7973-2792-8, (2004), 340p.

付録 ソースコード

グローバル変数・配列・関数の宣言

```

#include<stdio.h>
#include<math.h>

#define NN (int)200
#define MM (int)3*NN
#define M_PI (double)3.14159265358979323846e0
#define ec (double)1.6021766208e-19 //electronic charge(C)
#define mc2 (double)5.109989461e2 //electron rest energy(keV)
#define NO (double)6.022140857e23 //Avogadro constant
#define r02 (double)7.94e-26 //r0^2(cm^2)
#define pr2 (double)M_PI*r02

FILE *fp;
int npg, npn, nff, nmm, nsi, nsb, nsg, nni, nnb, nng;
double gA, gD, gL, gS, gP, gZ, gN, gK, gEk, gAL, gCP, Rni, Rnb, Rng; //basic parameters and Rn
double pgx[NN], pgy[NN], pnx[NN], pny[NN]; //density of gamma and neutron
double ffx[NN], ffy[NN], mmx[NN], mmy[NN]; //self-shielding factor, number of gamma-rays
double six[NN], siy[NN], sbx[NN], sby[NN], sgx[NN], sgy[NN]; //cross section
double sfp_i[MM], sfp_b[MM], sfp_g[MM]; //Internal Conversion, beta, Gamma

void introduction(void);
void init_setting(void);
void setting_view(void);
int file_reader(char bfn[100], double bx[], double by[]);
void file_writer(void);
int DV_funcs(int bn, double bx[], double by[], double bxr, double *byr);
int sort_func(int bna, int bnb, int bnc, double ba[], double bb[], double bc[]);
void Rn_calc(void);
void SI_GC(double bDEg, double bEMIN);
void SI_GP(double bEMIN);
void SI_NI(double bETk, double bETt, double bEit, double bEMIN);
void SI_NB(double bEb, double bEMIN);
void TI_NC(double bEMIN);
void TI_NP(double bEMIN);
double SIMPSON(double bDEg, double bLV, double bUV, double bEMIN);
double H_func(double bEg, double bEMIN);
double I_func(double bLV, double bUV, double bEMIN);

```

main 関数

```

int main(void)
{
    int i;
    double bEMIN, bDEg, bETk, bETt, bEit, bEb;
    double buf0, buf1;
    FILE *bf;

    //****+ user setting area start ****+

    //basic parameters
    gA=1.71e2;           //gram atomic weight
    gD=1.27e-1;         //cylindrical diameter
    gL=1.0e0;           //cylindrical length
    gS=M_PI*gD*gL;      //surface area of material
    gP=6.90e0;          //density of absorber
    gZ=7.0e1;           //atomic number
    gN=NO*gP/gA;        //number of atoms per unit volume
    gK=6.5e-4;          //density times inverse of specific energy loss
    gEk=6.1332e1;       //K-shell electron binding energy
    gAL=2.0e0*gD/3.0e0; //average track length

    bEMIN=5.0e1;        //average minimum kinetic energy (EMIN)
    bDEg=1.0e0;         //step size for integral, simpson's rule
    bETk=1.0e0;         //IT coefficient for K-shell (eta_K)
    bETt=1.0e0;         //IT coefficient for all shell (eta_T)
    bEit=1.0e0;         //energy of isomeric transition (E_it)
    bEb=5.0e2;          //maximum beta-particle energy

    //**+ user setting area completion **+

    gCP=4.0e1*sqrt(2.0e0)*pow(1.37e2, -4.0e0)/3.0e0;
    gCP*=pr2*gN*pow(gZ, 5.0e0)*pow(mc2, 3.5e0);

    fp=fopen("log-data.txt", "w");
    printf("=== program start (main) ===%n");
    fprintf(fp, "=== program start (main) ===%n");
    introduction();
    init_setting();
    setting_view();

    //input data file setting
    npg=file_reader("gamma-density.csv", pgx, pgy);
    npn=file_reader("neutron-density.csv", pnx, pny);
    nff=file_reader("self-shielding-factor.csv", ffx, ffy);
    nmm=file_reader("number-of-gamma.csv", mnx, mmy);
    nsi=file_reader("neutron-cross-section(I).csv", six, siy);
    nsb=file_reader("neutron-cross-section(B).csv", sbx, sby);
    nsg=file_reader("neutron-cross-section(G).csv", sgx, sgy);

    Rn_calc();
    SI_GC(bDEg, bEMIN);
    SI_GP(bEMIN);
    SI_NI(bETk, bETt, bEit, bEMIN);
    SI_NB(bEb, bEMIN);
    TI_NC(bEMIN);
    TI_NP(bEMIN);
    file_writer();

    printf("%n=== program completion (main) ===%n");
    fprintf(fp, "%n=== program completion (main) ===%n");
    fclose(fp);
    return 0;
}

```

introduction 関数と init_setting 関数

```

void introduction(void)
{
    printf("+-----+");
    printf("                                     |%n");
    printf("                Numerical Calculation Program      |%n");
    printf("                for Self-Powered Radiation Detector  |%n");
    printf("                Version 1.0 (2021/12/01)            |%n");
    printf("+-----+");
    fprintf(fp, "+-----+");
    fprintf(fp, "                Numerical Calculation Program      |%n");
    fprintf(fp, "                for Self-Powered Radiation Detector  |%n");
    fprintf(fp, "                Version 1.0 (2021/12/01)            |%n");
    fprintf(fp, "+-----+");

    return;
}

void init_setting(void)
{
    int i;
    printf("%n=== initialization start      (init_setting) ===%n");
    fprintf(fp, "%n=== initialization start      (init_setting) ===%n");

    npg=-1; npn=-1; nff=-1; nmm=-1;
    nsi=-1; nsb=-1; nsg=-1; nni=-1; nnb=-1; nng=-1;
    Rni=-1.0e0; Rnb=-1.0e0; Rng=-1.0e0;
    for (i=0; i<MM; i++) {sfp_i[i]=0.0e0; sfp_b[i]=0.0e0; sfp_g[i]=0.0e0;}
    for (i=0; i<NN; i++) {
        pgx[i]=0.0e0; pgy[i]=0.0e0; pnx[i]=0.0e0; pny[i]=0.0e0;
        ffx[i]=0.0e0; ffy[i]=0.0e0; mmx[i]=0.0e0; mmy[i]=0.0e0;
        six[i]=0.0e0; siy[i]=0.0e0; sbx[i]=0.0e0; sby[i]=0.0e0;
        sgx[i]=0.0e0; sgy[i]=0.0e0;
    }

    printf("=== initialization completion (init_setting) ===%n");
    fprintf(fp, "=== initialization completion (init_setting) ===%n");
    return;
}

```

setting_view 関数と file_reader 関数

```

void setting_view(void)
{
    printf("¥n=== setting view start      (setting_view) ===¥n");
    fprintf(fp,"¥n=== setting view start      (setting_view) ===¥n");

    printf("gram atomic weight (A)                :%12.4e (g mol-1)¥n",gA);
    printf("cylindrical diameter (d)                :%12.4e (cm)¥n",gD);
    printf("cylindrical length (L)                    :%12.4e (cm)¥n",gL);
    printf("surface area of material (S)                :%12.4e (cm2)¥n",gS);
    printf("density of absorber (rho)                   :%12.4e (g cm-3)¥n",gP);
    printf("atomic number, number of electrons per atom (Z) :%12.4e¥n",gZ);
    printf("number of atoms per unit volume (N)          :%12.4e (cm-3)¥n",gN);
    printf("density times inverse of specific energy loss (K) :%12.4e (g cm-2 keV-1)¥n",gK);
    fprintf(fp,"gram atomic weight (A)                :%12.4e (g mol-1)¥n",gA);
    fprintf(fp,"cylindrical diameter (d)                :%12.4e (cm)¥n",gD);
    fprintf(fp,"cylindrical length (L)                    :%12.4e (cm)¥n",gL);
    fprintf(fp,"surface area of material (S)                :%12.4e (cm2)¥n",gS);
    fprintf(fp,"density of absorber (rho)                   :%12.4e (g cm-3)¥n",gP);
    fprintf(fp,"atomic number, number of electrons per atom (Z) :%12.4e¥n",gZ);
    fprintf(fp,"number of atoms per unit volume (N)          :%12.4e (cm-3)¥n",gN);
    fprintf(fp,"density times inverse of specific energy loss (K) :%12.4e (g cm-2 keV-1)¥n",gK);

    printf("=== setting view completion (setting_view) ===¥n");
    fprintf(fp,"=== setting view completion (setting_view) ===¥n");
    return;
}

int file_reader(char bfn[100],double bx[],double by[])
{
    int i;
    double buf0,buf1;
    FILE *bf;
    printf("¥n=== file reading start      (file_reader) ===¥n");
    fprintf(fp,"¥n=== file reading start      (file_reader) ===¥n");

    i=-1;
    bf=fopen(bfn,"r");
    if(bf==NULL){
        printf("file opening failure [%s]¥n",bfn);
        fprintf(fp,"file opening failure [%s]¥n",bfn);
    }else{
        printf("file opening succeed [%s]¥n",bfn);
        fprintf(fp,"file opening succeed [%s]¥n",bfn);
        i=0;
        while(fscanf(bf,"%lf,%lf",&buf0,&buf1)!=EOF){
            bx[i]=buf0; by[i]=buf1; i++;
            if(i>(NN-1)){
                printf("array over¥n");
                fprintf(fp,"array over¥n");
                i=0;
                break;
            }
        }
        fclose(bf);
        i--;
    }

    printf("=== file reading completion (file_reader) ===¥n");
    fprintf(fp,"=== file reading completion (file_reader) ===¥n");
    return i;
}

```

file_writer 関数

```

void file_writer(void)
{
    int i;
    double buf0, buf1, buf2;
    FILE *bf;
    printf("¥n=== file writing start      (file_writer) ===¥n");
    fprintf(fp, "¥n=== file writing start      (file_writer) ===¥n");

    bf=fopen("confirmation_parameter.csv", "w");
    fprintf(bf, "No, pgx, pgy, pnx, pny, ffx, ffy, mmx, mmy¥n");
    for (i=0; i<NN; i++) {
        fprintf(bf, "%d, %12. 4e, %12. 4e, %12. 4e, %12. 4e, %12. 4e, %12. 4e, %12. 4e, %12. 4e¥n"
            , i, pgx[i], pgy[i], pnx[i], pny[i], ffx[i], ffy[i], mmx[i], mmy[i]);
    }
    fclose(bf);

    bf=fopen("confirmation_cross-section.csv", "w");
    fprintf(bf, "No, six, siy, sbx, sby, sgx, sgy¥n");
    for (i=0; i<NN; i++) {
        fprintf(bf, "%d, %12. 4e, %12. 4e, %12. 4e, %12. 4e, %12. 4e, %12. 4e¥n"
            , i, six[i], siy[i], sbx[i], sby[i], sgx[i], sgy[i]);
    }
    fclose(bf);

    bf=fopen("confirmation_IT.csv", "w");
    fprintf(bf, "No, sfp_i, sigma, f, phi¥n");
    for (i=0; i<MM; i++) {
        DV_funcs(nsi, six, siy, sfp_i[i], &buf0);
        DV_funcs(nff, ffx, ffy, sfp_i[i], &buf1);
        DV_funcs(npn, pnx, pny, sfp_i[i], &buf2);
        fprintf(bf, "%d, %12. 4e, %12. 4e, %12. 4e, %12. 4e¥n", i, sfp_i[i], buf0, buf1, buf2);
    }
    fclose(bf);

    bf=fopen("confirmation_beta.csv", "w");
    fprintf(bf, "No, sfp_b, sigma, f, phi¥n");
    for (i=0; i<MM; i++) {
        DV_funcs(nsb, sbx, sby, sfp_b[i], &buf0);
        DV_funcs(nff, ffx, ffy, sfp_b[i], &buf1);
        DV_funcs(npn, pnx, pny, sfp_b[i], &buf2);
        fprintf(bf, "%d, %12. 4e, %12. 4e, %12. 4e, %12. 4e¥n", i, sfp_b[i], buf0, buf1, buf2);
    }
    fclose(bf);

    bf=fopen("confirmation_gamma.csv", "w");
    fprintf(bf, "No, sfp_g, sigma, f, phi¥n");
    for (i=0; i<MM; i++) {
        DV_funcs(nsg, sgx, sgy, sfp_g[i], &buf0);
        DV_funcs(nff, ffx, ffy, sfp_g[i], &buf1);
        DV_funcs(npn, pnx, pny, sfp_g[i], &buf2);
        fprintf(bf, "%d, %12. 4e, %12. 4e, %12. 4e, %12. 4e¥n", i, sfp_g[i], buf0, buf1, buf2);
    }
    fclose(bf);

    printf("=== file writing completion (file_writer) ===¥n");
    fprintf(fp, "=== file writing completion (file_writer) ===¥n");
    return;
}

```

DV_funcs 関数と sort_func 関数

```

int DV_funcs(int bn, double bx[], double by[], double bxr, double *byr)
{
    int i;
    if((bn<0) || (bxr<0.0e0)) {*byr=0.0e0; return -1;}
    if(bxr>bx[bn]) {*byr=0.0e0; return -1;}
    for(i=0; i<=bn; i++) {if(bxr<=bx[i]) {*byr=by[i]; return i;}}
    *byr=0.0e0;
    return -1;
}

int sort_func(int bna, int bnb, int bnc, double ba[], double bb[], double bc[])
{
    int i, j, k, l, bEF;
    double bufa, bufb;
    if ((bna<0) || (bnb<0) || (bnc<0)) {return -1;}
    i=0; j=0; k=0; bEF=0;

    while (bEF==0) {
        bufa=ba[i]; bufb=bb[j];
        if (bufa==bufb) {
            if (k<=bnc) {bc[k]=bufa;}
            i++; j++; k++;
        } else if (bufa<bufb) {
            if (k<=bnc) {bc[k]=bufa;}
            i++; k++;
        } else //(bufa>bufb)
            if (k<=bnc) {bc[k]=bufb;}
            j++; k++;
        }
        if ((i>bna) && (j<=bnb)) {bEF=1;}
        if ((i<=bna) && (j>bnb)) {bEF=2;}
        if ((i>bna) && (j>bnb)) {bEF=3;}
        if (k>bnc) {bEF=4;}
    }

    switch (bEF)
    {
        case 1://exist surplus of bb[]
            for (l=j; l<=bnb; l++) {
                if (k<=bnc) {
                    bc[k]=bb[l]; k++;
                } else {
                    return -1;//lack of bc[]
                }
            }
            break;
        case 2://exist surplus of ba[]
            for (l=i; l<=bna; l++) {
                if (k<=bnc) {
                    bc[k]=ba[l]; k++;
                } else {
                    return -1;//lack of bc[]
                }
            }
            break;
        case 4://lack of bc[]
            return -1;
            break;
        default:
            break;
    }

    k--;
    return k;
}

```

Rn_calc 関数

```

void Rn_calc(void)
{
    int i, bn0, bn1, bn2;
    double buf0, buf1, buf2, buf3;
    double bFP[2*NN];
    printf("%n=== calculation of neutron capture rate start          (Rn_calc) ===%n");
    fprintf(fp, "%n=== calculation of neutron capture rate start          (Rn_calc) ===%n");

    if(npn<0) {
        printf("data of neutron-density is nothing%n");
        printf("=== calculation of neutron capture rate discontinuation (Rn_calc) ===%n");
        fprintf(fp, "data of neutron-density is nothing%n");
        fprintf(fp, "=== calculation of neutron capture rate discontinuation (Rn_calc) ===%n");
        return;
    }
    if(nff<0) {
        printf("data of self-shielding-factor is nothing%n");
        printf("=== calculation of neutron capture rate discontinuation (Rn_calc) ===%n");
        fprintf(fp, "data of self-shielding-factor is nothing%n");
        fprintf(fp, "=== calculation of neutron capture rate discontinuation (Rn_calc) ===%n");
        return;
    }
    bn0=2*NN-1; bn1=MM-1;
    for (i=0; i<=bn0; i++) {bFP[i]=0.0e0;}
    bn2=sort_func(npn, nff, bn0, pnx, ffx, bFP);
    //Rn of IT
    if(nsi>=0) {
        nni=sort_func(bn2, nsi, bn1, bFP, six, sfp_i);
        Rni=0.0e0;
        for (i=1; i<=nni; i++) {
            buf0=sfp_i[i]-sfp_i[i-1]; //dE=E(i)-E(i-1)
            DV_funcs(nsi, six, siy, sfp_i[i], &buf1);    DV_funcs(nff, ffx, ffy, sfp_i[i], &buf2);
            DV_funcs(npn, pnx, pny, sfp_i[i], &buf3);
            Rni+=buf0*buf1*buf2*buf3;
        }
    }
    //Rn of beta decay
    if(nsb>=0) {
        nnb=sort_func(bn2, nsb, bn1, bFP, sbx, sfp_b);
        Rnb=0.0e0;
        for (i=1; i<=nnb; i++) {
            buf0=sfp_b[i]-sfp_b[i-1]; //dE=E(i)-E(i-1)
            DV_funcs(nsb, sbx, sby, sfp_b[i], &buf1);    DV_funcs(nff, ffx, ffy, sfp_b[i], &buf2);
            DV_funcs(npn, pnx, pny, sfp_b[i], &buf3);
            Rnb+=buf0*buf1*buf2*buf3;
        }
    }
    //Rn of gamma
    if(nsg>=0) {
        nng=sort_func(bn2, nsg, bn1, bFP, sgx, sfp_g);
        Rng=0.0e0;
        for (i=1; i<=nng; i++) {
            buf0=sfp_g[i]-sfp_g[i-1]; //dE=E(i)-E(i-1)
            DV_funcs(nsg, sgx, sgy, sfp_g[i], &buf1);    DV_funcs(nff, ffx, ffy, sfp_g[i], &buf2);
            DV_funcs(npn, pnx, pny, sfp_g[i], &buf3);
            Rng+=buf0*buf1*buf2*buf3;
        }
    }
    printf("Rn(IT)=%12.4e, Rn(beta)=%12.4e, Rn(gamma)=%12.4e%n", Rni, Rnb, Rng);
    fprintf(fp, "Rn(IT)=%12.4e, Rn(beta)=%12.4e, Rn(gamma)=%12.4e%n", Rni, Rnb, Rng);
    printf("=== calculation of neutron capture rate completion          (Rn_calc) ===%n");
    fprintf(fp, "=== calculation of neutron capture rate completion          (Rn_calc) ===%n");
    return;
}

```


SI_GC 関数

```

void SI_GC(double bDEg, double bEMIN)
{
    int i, bm;
    double Igc, bEmax, bEl, buf0, buf1;
    printf("¥n=== single interactions (compton electron) start      (SI_GC) ===¥n");
    fprintf(fp, "¥n=== single interactions (compton electron) start      (SI_GC) ===¥n");

    if((npg<0) || (bDEg<=0.0e0)) {
        printf("npg error or bDEg error¥n");
        printf("=== calculation of gamma-compton discontinuation      (SI_GC) ===¥n");
        fprintf(fp, "npg error or bDEg error¥n");
        fprintf(fp, "=== calculation of gamma-compton discontinuation      (SI_GC) ===¥n");
        return;
    }

    bEmax=0.0e0;
    for (i=npg; i>=0; i--) {
        if (pgy[i]>0.0e0) { bEmax=pgx[i]; break; } //maximum energy of gamma-ray
    }

    buf0=5.0e-1*bEMIN; //EMIN/2
    bEl=buf0+sqrt(buf0*buf0+mc2*buf0); //calculation of E_l (EMIN)
    bm=DV_funcs(npg, pgx, pgy, bEl, &buf0); //gamma density
    if (bm!=-1) {
        Igc=buf0*SIMPSON(bDEg, bEl, pgx[bm], bEMIN);
        if (bm<npg) {
            for (i=bm+1; i<=npg; i++) {
                DV_funcs(npg, pgx, pgy, pgx[i], &buf0);
                Igc+=buf0*SIMPSON(bDEg, pgx[i-1], pgx[i], bEMIN);
            }
        }
        Igc*=(2.5e-1*ec*gS*gK*gN*gZ/gP);
    } else {
        Igc=0.0e0;
    }

    buf1=Igc*gA/(gL*gD*gK*gZ);
    printf("maximum energy of gamma-ray (E_max)      :%12.4e (keV)¥n", bEmax);
    printf("minimum gamma energy (E_l (EMIN))          :%12.4e (keV)¥n", bEl);
    printf("average minimum kinetic energy (EMIN)        :%12.4e (keV)¥n", bEMIN);
    printf("current (I_gamma, compton)                    :%12.4e (A)¥n", Igc);
    printf("current (I_gamma, compton) (1cm)              :%12.4e (A)¥n", Igc/gL);
    printf("T(E_gamma, EMIN) (1cm)                       :%12.4e¥n", buf1);
    printf("dE_gamma                                       :%12.4e (keV)¥n", bDEg);
    fprintf(fp, "maximum energy of gamma-ray (E_max)      :%12.4e (keV)¥n", bEmax);
    fprintf(fp, "minimum gamma energy (E_l (EMIN))          :%12.4e (keV)¥n", bEl);
    fprintf(fp, "average minimum kinetic energy (EMIN)        :%12.4e (keV)¥n", bEMIN);
    fprintf(fp, "current (I_gamma, compton)                    :%12.4e (A)¥n", Igc);
    fprintf(fp, "current (I_gamma, compton) (1cm)              :%12.4e (A)¥n", Igc/gL);
    fprintf(fp, "T(E_gamma, EMIN) (1cm)                       :%12.4e¥n", buf1);
    fprintf(fp, "dE_gamma                                       :%12.4e (keV)¥n", bDEg);

    printf("=== single interactions (compton electron) completion (SI_GC) ===¥n");
    fprintf(fp, "=== single interactions (compton electron) completion (SI_GC) ===¥n");
    return;
}

```

SI_GP 関数

```

void SI_GP(double bEMIN)
{
    int i, bm;
    double Igp, bEmax, buf0, buf1;
    printf("¥n=== single interactions (photoelectron) start      (SI_GP) ===¥n");
    fprintf(fp, "¥n=== single interactions (photoelectron) start      (SI_GP) ===¥n");

    if(npg<0) {
        printf("npg error¥n");
        printf("=== calculation of gamma-photoelectron discontinuation (SI_GP) ===¥n");
        fprintf(fp, "npg error¥n");
        fprintf(fp, "=== calculation of gamma-photoelectron discontinuation (SI_GP) ===¥n");
        return;
    }

    bEmax=0. 0e0;
    for (i=npg; i>=0; i--) {
        if(pgy[i]>0. 0e0) {bEmax=pgx[i]; break;} //maximum energy of gamma-ray
    }

    buf0=gEk+bEMIN;
    bm=DV_funcs(npg, pgx, pgy, buf0, &buf1);
    if (bm!=-1) {
        Igp=buf1*I_func(buf0, pgx[bm], bEMIN);
        if (bm<npg) {
            for (i=bm+1; i<=npg; i++) {
                DV_funcs(npg, pgx, pgy, pgx[i], &buf1);
                Igp+=buf1*I_func(pgx[i-1], pgx[i], bEMIN);
            }
        }
        Igp*=(2. 5e-1*ec*gS*gK/gP)*gCP;
    } else {
        Igp=0. 0e0;
    }

    buf1=Igp*gA/(gL*gD*gK*pow(gZ, 5. 0e0));
    printf("maximum energy of gamma-ray (E_max)      :%12. 4e (keV)¥n", bEmax);
    printf("K-shell electron binding energy (E_K)      :%12. 4e (keV)¥n", gEk);
    printf("average minimum kinetic energy (EMIN)        :%12. 4e (keV)¥n", bEMIN);
    printf("current (I_gamma, photoelectron)              :%12. 4e (A)¥n", Igp);
    printf("current (I_gamma, photoelectron) (1cm)        :%12. 4e (A)¥n", Igp/gL);
    printf("D(E_gamma, EMIN) (1cm)                       :%12. 4e¥n", buf1);
    fprintf(fp, "maximum energy of gamma-ray (E_max)      :%12. 4e (keV)¥n", bEmax);
    fprintf(fp, "K-shell electron binding energy (E_K)      :%12. 4e (keV)¥n", gEk);
    fprintf(fp, "average minimum kinetic energy (EMIN)        :%12. 4e (keV)¥n", bEMIN);
    fprintf(fp, "current (I_gamma, photoelectron)              :%12. 4e (A)¥n", Igp);
    fprintf(fp, "current (I_gamma, photoelectron) (1cm)        :%12. 4e (A)¥n", Igp/gL);
    fprintf(fp, "D(E_gamma, EMIN) (1cm)                       :%12. 4e¥n", buf1);

    printf("=== single interactions (photoelectron) completion      (SI_GP) ===¥n");
    fprintf(fp, "=== single interactions (photoelectron) completion      (SI_GP) ===¥n");
    return;
}

```

SI_NI 関数

```

void SI_NI(double bETk,double bETt,double bEit,double bEMIN)
{
    double Ini,buf0,buf1;
    printf("¥n=== single interactions (internal conversion) start      (SI_NI) ===¥n");
    fprintf(fp,"¥n=== single interactions (internal conversion) start      (SI_NI) ===¥n");

    if(Rni<0.0e0) {
        printf("Rni error¥n");
        printf("=== calculation of internal conversion discontinuation  (SI_NI) ===¥n");
        fprintf(fp,"Rni error¥n");
        fprintf(fp,"=== calculation of internal conversion discontinuation  (SI_NI) ===¥n");
        return;
    }

    buf0=(2.5e-1*ec*gS*gK/gP)*(bETk/(1.0e0+bETt))*Rni;
    buf1=(bEit-gEk-bEMIN);
    if(buf1>=0.0e0) {
        Ini=buf0*buf1;
    }else{
        Ini=0.0e0;
    }

    printf("neutron capture rate (Rn_IT)           :%12.4e (cm-3 sec-1)¥n",Rni);
    printf("IT coefficient for K-shell (eta_K)         :%12.4e¥n",bETk);
    printf("IT coefficient for all shell (eta_T)         :%12.4e¥n",bETt);
    printf("energy of isomeric transition (E_it)        :%12.4e (keV)¥n",bEit);
    printf("K-shell electron binding energy (E_K)       :%12.4e (keV)¥n",gEk);
    printf("average minimum kinetic energy (EMIN)      :%12.4e (keV)¥n",bEMIN);
    printf("current (I_IT)                               :%12.4e (A)¥n",Ini);
    printf("current (I_IT) (1cm)                        :%12.4e (A)¥n",Ini/gL);
    fprintf(fp,"neutron capture rate (Rn_IT)           :%12.4e (cm-3 sec-1)e¥n",Rni);
    fprintf(fp,"IT coefficient for K-shell (eta_K)         :%12.4e¥n",bETk);
    fprintf(fp,"IT coefficient for all shell (eta_T)         :%12.4e¥n",bETt);
    fprintf(fp,"energy of isomeric transition (E_it)        :%12.4e (keV)¥n",bEit);
    fprintf(fp,"K-shell electron binding energy (E_K)       :%12.4e (keV)¥n",gEk);
    fprintf(fp,"average minimum kinetic energy (EMIN)      :%12.4e (keV)¥n",bEMIN);
    fprintf(fp,"current (I_IT)                               :%12.4e (A)¥n",Ini);
    fprintf(fp,"current (I_IT) (1cm)                        :%12.4e (A)¥n",Ini/gL);

    printf("=== single interactions (internal conversion) completion (SI_NI) ===¥n");
    fprintf(fp,"=== single interactions (internal conversion) completion (SI_NI) ===¥n");
    return;
}

```

SI_NB 関数

```

void SI_NB(double bEb, double bEMIN)
{
    double Inb, buf0, buf1;
    printf("\n=== single interactions (beta decay) start      (SI_NB) ===\n");
    fprintf(fp, "\n=== single interactions (beta decay) start      (SI_NB) ===\n");

    if((bEb <= 0.0e0) || (Rnb < 0.0e0)) {
        printf("beta decay energy error or Rnb error\n");
        printf("=== calculation of beta decay discontinuation  (SI_NB) ===\n");
        fprintf(fp, "beta decay energy error or Rnb error\n");
        fprintf(fp, "=== calculation of beta decay discontinuation  (SI_NB) ===\n");
        return;
    }

    buf0 = pow(bEMIN, 2.5e0) / (1.2e1 * pow(bEb, 1.5e0));
    buf1 = 2.1e1 - 1.8e1 * bEMIN / bEb + 5.0e0 * bEMIN * bEMIN / (bEb * bEb);
    buf0 *= buf1;
    buf1 = bEb / 3.0e0 - bEMIN;
    buf0 += buf1;
    buf1 = (2.5e-1 * ec * gS * gK / gP) * Rnb;
    Inb = buf0 * buf1;

    printf("neutron capture rate (Rn_beta)           :%12.4e (cm-3 sec-1)\n", Rnb);
    printf("maximum beta-particle energy (E_beta) :%12.4e (keV)\n", bEb);
    printf("average minimum kinetic energy (EMIN) :%12.4e (keV)\n", bEMIN);
    printf("current (I_n, beta)                       :%12.4e (A)\n", Inb);
    printf("current (I_n, beta) (1cm)                 :%12.4e (A)\n", Inb/gL);
    fprintf(fp, "neutron capture rate (Rn_beta)           :%12.4e (cm-3 sec-1)\n", Rnb);
    fprintf(fp, "maximum beta-particle energy (E_beta) :%12.4e (keV)\n", bEb);
    fprintf(fp, "average minimum kinetic energy (EMIN) :%12.4e (keV)\n", bEMIN);
    fprintf(fp, "current (I_n, beta)                       :%12.4e (A)\n", Inb);
    fprintf(fp, "current (I_n, beta) (1cm)                 :%12.4e (A)\n", Inb/gL);

    printf("=== single interactions (beta decay) completion (SI_NB) ===\n");
    fprintf(fp, "=== single interactions (beta decay) completion (SI_NB) ===\n");
    return;
}

```

TI_NC 関数

```

void TI_NC(double bEMIN)
{
    int i;
    double Inc,buf0;
    printf("%n=== two interactions (compton electron) start          (TI_NC) ===%n");
    fprintf(fp,"%n=== two interactions (compton electron) start          (TI_NC) ===%n");

    if((nmm<0)|| (Rng<0.0e0)) {
        printf("nmm error or Rng error%n");
        printf("=== calculation of neutron-compton electron discontinuation (TI_NC) ===%n");
        fprintf(fp,"nmm error or Rng error%n");
        fprintf(fp,"=== calculation of neutron-compton electron discontinuation (TI_NC) ===%n");
        return;
    }

    Inc=0.0e0;
    for(i=0;i<=nmm;i++) {
        buf0=mmy[i]*H_func(mmx[i], bEMIN);
        Inc+=buf0;
    }
    Inc*=(2.5e-1*ec*gS*gK/gP)*gN*gZ*gAL*Rng;

    printf("neutron capture rate (Rn_gamma)          :%12.4e (cm-3 sec-1)%n",Rng);
    printf("average track length (overline l)          :%12.4e (cm)%n",gAL);
    printf("average minimum kinetic energy (EMIN) :%12.4e (keV)%n",bEMIN);
    printf("current (I_n, gamma, compton)          :%12.4e (A)%n",Inc);
    printf("current (I_n, gamma, compton) (1cm)      :%12.4e (A)%n",Inc/gL);
    fprintf(fp, "neutron capture rate (Rn_gamma)          :%12.4e (cm-3 sec-1)%n",Rng);
    fprintf(fp, "average track length (overline l)          :%12.4e (cm)%n",gAL);
    fprintf(fp, "average minimum kinetic energy (EMIN) :%12.4e (keV)%n",bEMIN);
    fprintf(fp, "current (I_n, gamma, compton)          :%12.4e (A)%n",Inc);
    fprintf(fp, "current (I_n, gamma, compton) (1cm)      :%12.4e (A)%n",Inc/gL);

    printf("=== two interactions (compton electron) completion          (TI_NC) ===%n");
    fprintf(fp,"=== two interactions (compton electron) completion          (TI_NC) ===%n");
    return;
}

```

TI_NP 関数

```

void TI_NP(double bEMIN)
{
    int i;
    double Inp, buf0, buf1;
    printf("¥n=== two interactions (photoelectron) start          (TI_NP) ===¥n");
    fprintf(fp, "¥n=== two interactions (photoelectron) start          (TI_NP) ===¥n");

    if((nmm<0) || (Rng<0.0e0)) {
        printf("nmm error or Rng error¥n");
        printf("=== calculation of neutron-photoelectron discontinuation (TI_NP) ===¥n");
        fprintf(fp, "nmm error or Rng error¥n");
        fprintf(fp, "=== calculation of neutron-photoelectron discontinuation (TI_NP) ===¥n");
        return;
    }

    Inp=0.0e0;
    for (i=0; i<=nmm; i++) {
        if (mmx[i]<=0.0e0) {continue;}
        buf0=mmx[i]-gEk-bEMIN;
        if (buf0<=0.0e0) {continue;}
        buf1=pow(mmx[i], -3.5e0);
        Inp+=mmy[i]*buf0*buf1;
    }
    Inp*=(2.5e-1*ec*gS*gK/gP)*gCP*gAL*Rng;

    printf("neutron capture rate (Rn_gamma)           :%12.4e (cm-3 sec-1)¥n", Rng);
    printf("average track length (overline l)           :%12.4e (cm)¥n", gAL);
    printf("K-shell electron binding energy (E_K)         :%12.4e (keV)¥n", gEk);
    printf("average minimum kinetic energy (EMIN)         :%12.4e (keV)¥n", bEMIN);
    printf("current (I_n, gamma, photoelectron)           :%12.4e (A)¥n", Inp);
    printf("current (I_n, gamma, photoelectron) (1cm)     :%12.4e (A)¥n", Inp/gL);
    fprintf(fp, "neutron capture rate (Rn_gamma)           :%12.4e (cm-3 sec-1)e¥n", Rng);
    fprintf(fp, "average track length (overline l)           :%12.4e (cm)¥n", gAL);
    fprintf(fp, "K-shell electron binding energy (E_K)         :%12.4e (keV)¥n", gEk);
    fprintf(fp, "average minimum kinetic energy (EMIN)         :%12.4e (keV)¥n", bEMIN);
    fprintf(fp, "current (I_n, gamma, photoelectron)           :%12.4e (A)¥n", Inp);
    fprintf(fp, "current (I_n, gamma, photoelectron) (1cm)     :%12.4e (A)¥n", Inp/gL);

    printf("=== two interactions (photoelectron) completion (TI_NP) ===¥n");
    fprintf(fp, "=== two interactions (photoelectron) completion (TI_NP) ===¥n");
    return;
}

```

SIMPSON 関数と I_func 関数

```

double SIMPSON(double bDEg, double bLV, double bUV, double bEMIN)
{
    int i, bn;
    double ANS, buf0, buf1, buf2, buf3;

    if ((bDEg<=0.0e0) || (bUV<=bLV)) {return 0.0e0;}
    bn=(int)((bUV-bLV)/bDEg);

    if (bn==0) {
        buf0=5.0e-1*(bLV+bUV);
        ANS=H_func(bLV, bEMIN)+4.0e0*H_func(buf0, bEMIN)+H_func(bUV, bEMIN);
        ANS*=(bUV-bLV)/6.0e0;
    } else {
        ANS=0.0e0;
        buf0=5.0e-1*bDEg;
        for (i=1; i<=bn; i++) {
            buf1=bLV+(double)(i-1)*bDEg;
            buf2=buf1+buf0;
            buf3=buf1+bDEg;
            ANS+=H_func(buf1, bEMIN)+4.0e0*H_func(buf2, bEMIN)+H_func(buf3, bEMIN);
        }
        ANS*=bDEg/6.0e0;
        buf1=buf3;
        buf2=5.0e-1*(buf1+bUV);
        buf3=bUV;
        buf0=H_func(buf1, bEMIN)+4.0e0*H_func(buf2, bEMIN)+H_func(buf3, bEMIN);
        buf0*=(bUV-buf1)/6.0e0;
        ANS+=buf0;
    }

    return ANS;
}

double I_func(double bLV, double bUV, double bEMIN)
{
    double bC0, bC1, ANS;

    if ((bLV<=0.0e0) || (bUV<=bLV)) {return 0.0e0;}
    bC0=-2.0e0/3.0e0;
    bC1=4.0e-1*(gEk+bEMIN);

    ANS=(bC0/bUV+bC1/(bUV*bUV))/sqrt(bUV);
    ANS=(bC0/bLV+bC1/(bLV*bLV))/sqrt(bLV);

    return ANS;
}

```

H_func 関数

```

double H_func(double bEg, double bEMIN)
{
    double bC0, bC1, bC2, bEMEG;
    double bEd, bDE, bDL, bDEDL, bVF, bVGu, bVGI, ANS;

    if((bEg<=0.0e0) || (bEg<=bEMIN)) {return 0.0e0;}
    bEMEG=2.0e0*bEg*bEg/(mc2+2.0e0*bEg); //E_M(E_gamma) or E_t(E_gamma)
    if(bEMEG<=bEMIN) {return 0.0e0;}

    bC0=pr2*mc2/(bEg*bEg);
    bC1=bC0*mc2*mc2;
    bC2=-2.0e0*bC0*mc2;

    //F function(E_gamma, E_M(E_gamma))
    bEd=bEMEG;
    bDE=bEg-bEd;
    bDL=log(bDE);
    bVF=bC0*(-bEg*bDL+bEd-(5.0e-1*bEd*bEd/bEg));
    bVF+=bC1*((1.0e0/bDE)+(2.0e0*bDL/bEg)+(bEd/(bEg*bEg)));
    bVF+=bC2*(-bDL-(bEd/bEg));

    //G function(E_gamma, E_M(E_gamma))
    bEd=bEMEG;
    bDE=bEg-bEd;
    bDL=log(bDE);
    bDEDL=bDE*bDL;
    bVGu=bC0*(bEg*bDEDL+bEg*bEd+(5.0e-1*bEd*bEd)-(bEd*bEd*bEd/(6.0e0*bEg)));
    bVGu+=bC1*(-bDL-(2.0e0*bDEDL/bEg)-(2.0e0*bEd/bEg)+(5.0e-1*bEd*bEd/(bEg*bEg)));
    bVGu+=bC2*(bDEDL+bEd-(5.0e-1*bEd*bEd/(bEg)));

    //G function(E_gamma, EMIN)
    bEd=bEMIN;
    bDE=bEg-bEd;
    bDL=log(bDE);
    bDEDL=bDE*bDL;
    bVGI=bC0*(bEg*bDEDL+bEg*bEd+(5.0e-1*bEd*bEd)-(bEd*bEd*bEd/(6.0e0*bEg)));
    bVGI+=bC1*(-bDL-(2.0e0*bDEDL/bEg)-(2.0e0*bEd/bEg)+(5.0e-1*bEd*bEd/(bEg*bEg)));
    bVGI+=bC2*(bDEDL+bEd-(5.0e-1*bEd*bEd/(bEg)));

    //H function(E_gamma, EMIN)
    ANS=bVF*(bEMEG-bEMIN)-(bVGu-bVGI);

    return ANS;
}

```