



JAEA-Testing

2024-001

DOI:10.11484/jaea-testing-2024-001

## 中性子共鳴イメージング解析コード RAIM マニュアル

Manuals for Resonance Analysis Code for Neutron IMaging "RAIM"

長谷美 宏幸 甲斐 哲也

Hiroyuki HASEMI and Tetsuya KAI

J-PARC センター

物質・生命科学ディビジョン

Materials and Life Science Division

J-PARC Center

August 2024

Japan Atomic Energy Agency

日本原子力研究開発機構

JAEA-Testing

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 研究開発推進部 科学技術情報課  
〒319-1112 茨城県那珂郡東海村大字村松4番地49  
E-mail: [ird-support@jaea.go.jp](mailto: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 Library, Institutional Repository and INIS Section, Research and Development Promotion Department, Japan Atomic Energy Agency.  
4-49 Muramatsu, Tokai-mura, Naka-gun, Ibaraki-ken 319-1112, Japan  
E-mail: [ird-support@jaea.go.jp](mailto:ird-support@jaea.go.jp)

## 中性子共鳴イメージング解析コード RAIM マニュアル

日本原子力研究開発機構 J-PARC センター  
物質・生命科学ディビジョン

長谷美 宏幸、甲斐 哲也

(2024年5月24日受理)

RAIM (Resonance Analysis code for neutron IMaging) は大強度陽子加速器施設 J-PARC の物質・生命科学実験施設 (MLF) などのパルス中性子源で測定した共鳴吸収スペクトルを解析し、試料の原子核の密度や温度などの情報を求める解析コードである。RAIM は中性子のパルス関数と断面積データをもとにした共鳴吸収関数の畳み込み計算によりパルス中性子源で測定される共鳴吸収スペクトルを再現し、測定で得られた共鳴吸収スペクトルデータに対してフィッティングを行うことで、試料に含まれる特定の元素の密度や温度を求めることができる。また、RAIM は計算設定のパラメータを極力少なくすることや、2次元検出器で測定した数多くのスペクトルを一括で処理するスクリプトを用意するなど、共鳴イメージングデータを容易に解析することを念頭に開発されている。

本書は RAIM の計算機へのインストール方法および共鳴吸収スペクトルのシミュレーションと測定データへのフィッティング方法について説明するマニュアルである。

## **Manuals for Resonance Analysis Code for Neutron IMaging “RAIM”**

Hiroyuki HASEMI and Tetsuya KAI

Materials and Life Science Division, J-PARC Center  
Japan Atomic Energy Agency  
Tokai-mura, Naka-gun, Ibaraki-ken

(Received May 24, 2024)

RAIM is an analysis code that analyzes resonance absorption spectra measured at pulsed neutron sources such as the Materials and Life Science Experimental Facility (MLF) at the Japan Proton Accelerator Research Complex (J-PARC) to obtain information on nuclear densities and temperatures. By calculating the convolution of the pulse functions of neutron beam and the resonance capture function that is based on the nuclear cross section data, RAIM reproduces the resonance absorption spectrum measured by a pulsed neutron source. Then, RAIM determines the density and temperature of specific nuclides in a sample by performing spectral fitting on the resonance absorption spectrum data. In addition, RAIM is developed to facilitate the analysis of resonance imaging data by minimizing the number of parameters for calculation setup and by providing scripts for processing many resonance absorption spectra measured by a two-dimensional detector at once. This manual explains how to install RAIM on a computer and how to simulate resonance absorption spectra and fit them to measured data.

Keywords: RAIM, Neutron Resonance Transmission Analysis, Neutron Resonance Capture Analysis, Neutron Imaging, MLF, J-PARC

目 次

1. はじめに	1
1.1. 本書について	1
1.2. RAIM の概要	1
1.3. RAIM の機能	1
1.4. 動作環境	4
2. インストール	5
2.1. 概要	5
2.2. 事前準備	5
2.3. FRENDY のコンパイル	6
2.4. RAIM 実行用スクリプト生成	6
3. 計算方法	7
3.1. 概要	7
3.2. 共通する計算手順	7
3.3. インプットファイル	8
3.4. シミュレーション計算	15
3.5. スペクトルフィッティング	17
3.6. 連続スペクトルフィッティング	19
3.7. 温度マップを用いたスペクトルフィッティング	20
3.8. 複数データの一括スペクトルフィッティング	22
謝辞	24
参考文献	25
付録 A 実行用シェルスクリプト	26
付録 B サンプルインプット	27

Contents

1. Introduction .....	1
1.1. About this document .....	1
1.2. Overview of RAIM .....	1
1.3. Features of RAIM .....	1
1.4. Operating environment .....	4
2. Installation .....	5
2.1. Outline .....	5
2.2. Preparations .....	5
2.3. Compiling FRENDY .....	6
2.4. Generating running scripts for RAIM .....	6
3. Calculation methods .....	7
3.1. Outline .....	7
3.2. Common calculation process .....	7
3.3. Input file .....	8
3.4. Simulation .....	15
3.5. Spectral fitting .....	17
3.6. Continuous spectral fitting .....	19
3.7. Spectral fitting using temperature map .....	20
3.8. Batch spectral fitting for multiple data .....	22
Acknowledgements .....	24
References .....	25
Appendix A Shell scripts to run calculation .....	26
Appendix B Sample input files .....	27

表リスト

Table 1	必要なソフトウェア	4
Table 2	ディレクトリ構成	5
Table 3	&control のパラメータ	9
Table 4	&xs のパラメータ	10
Table 5	&material のパラメータ	10
Table 6	&transmission のパラメータ	11
Table 7	&capture のパラメータ	12
Table 8	&fitting のパラメータ	13
Table 9	&pulse のパラメータ	15
Table A1	実行シェルスクリプトで指定される入力ファイル名および Python スクリプト名	26

図リスト

Figure 1	温度マップの例	4
Figure 2	計算された共鳴透過率スペクトル	16
Figure 3	計算された共鳴吸収量スペクトル	17
Figure 4	測定データとフィットした共鳴透過率スペクトル	18
Figure 5	測定データとフィットした共鳴吸収量スペクトル	19
Figure A1	透過率のシミュレーション計算のシェルスクリプト	26
Figure A2	透過率データの連続スペクトルフィッティングのシェルスクリプト	26
Figure B1	透過率のシミュレーション計算のインプットファイル	27
Figure B2	共鳴吸収量のシミュレーション計算のインプットファイル	28
Figure B3	透過率のスペクトルフィッティングのインプットファイル	29
Figure B4	共鳴吸収量のスペクトルフィッティングのインプットファイル	30
Figure B5	透過率データの連続フィッティングのインプットファイル	31
Figure B6	共鳴吸収量データの連続フィッティングのインプットファイル	32
Figure B7	温度マップを用いた透過率データのスペクトルフィッティングのインプットファイル	33
Figure B8	ポイントワイズデータの温度マップによる透過率のスペクトルフィッティングのインプットファイル	34
Figure B9	温度マップを用いた共鳴吸収量のスペクトルフィッティングのインプットファイル	35
Figure B10	複数の透過率データのスペクトルフィッティングのインプットファイル	36
Figure B11	温度マップを利用する複数の透過率データのスペクトルフィッティングのイン	

ットファイル .....	37
Figure B12 複数の共鳴吸収量データのスペクトルフィッティングのインットファイル	38
Figure B13 温度マップを利用する複数の共鳴吸収量データのスペクトルフィッティングの インットファイル .....	39



List of tables

Table 1	Required software	4
Table 2	Directory structure	5
Table 3	Parameters of &control	9
Table 4	Parameters of &xs	10
Table 5	Parameters of &material	10
Table 6	Parameters of &transmission	11
Table 7	Parameters of &capture	12
Table 8	Parameters of &fitting	13
Table 9	Parameters of &pulse	15
Table A1	Name of input files and Python scripts specified in the executing shell script	26

List of figures

Figure 1	Example of temperature map	4
Figure 2	Simulated resonance transmission spectrum	16
Figure 3	Simulated resonance capture spectrum	17
Figure 4	Fitted resonance transmission spectrum with measured data	18
Figure 5	Fitted resonance capture spectrum with measured data	19
Figure A1	Shell script of transmission simulation	26
Figure A2	Shell script of continuous fitting for transmission	26
Figure B1	Sample input file of transmission simulation	27
Figure B2	Sample input file of capture yield simulation	28
Figure B3	Sample input file of fitting for transmission data	29
Figure B4	Sample input file of fitting for capture yield data	30
Figure B5	Sample input file of continuous fitting for transmission data	31
Figure B6	Sample input file of continuous fitting for capture yield data	32
Figure B7	Sample input file of fitting for transmission data using temperature map	33
Figure B8	Sample input file of fitting for transmission data using temperature map of pointwise data	34
Figure B9	Sample input file of fitting for capture yield data using temperature map	35
Figure B10	Sample input file of fitting for multiple transmission data	36
Figure B11	Sample input file of fitting for multiple transmission data using temperature map	37
Figure B12	Sample input file of fitting for multiple capture yield data	38
Figure B13	Sample input file of fitting for multiple capture yield data using temperature map	39

This is a blank page.

## 1. はじめに

### 1.1. 本書について

本書は、大強度陽子加速器施設 J-PARC の物質・生命科学実験施設 (MLF) などのパルス中性子源で測定した共鳴吸収スペクトルの解析を行う解析コード RAIM (Resonance Analysis code for neutron IMaging) のマニュアルであり、計算機へのインストール方法の説明および RAIM を利用したシミュレーションやフィッティングを行うための利用方法の説明で構成される。

### 1.2. RAIM の概要

RAIM は J-PARC MLF などのパルス中性子源で測定した共鳴吸収スペクトルを解析し、試料に含まれる原子核の密度や温度などの情報を求める解析コードである。ここでいう共鳴吸収スペクトルは中性子共鳴吸収分光法 (N-RAS : Neutron Resonance Absorption Spectroscopy) で測定された中性子飛行時間 (TOF : Time-of-Flight) 依存の中性子透過率または共鳴吸収量 (即発ガンマ線収量) を指す<sup>1)</sup>。このマニュアルでは、前者を透過型 N-RAS、後者を即発ガンマ線型 N-RAS と呼ぶ。共鳴吸収スペクトルを解析することで試料に含まれる核種の同定、原子数密度 (単位面積当たりの原子数) や温度を求めることが可能となる。

RAIM の特徴はパルス関数 (減速材から放出される中性子の時間分布) と断面積データから得られる共鳴吸収関数の畳み込み計算を行うことで、パルス中性子源で測定される共鳴吸収スペクトルを正確に再現し、測定で得られた共鳴吸収スペクトルデータに対してフィッティングを行うことで、試料に含まれる特定の核種の密度や温度を求めることができる。また、RAIM はコード名にもあるように共鳴イメージングのために開発されたコードであり、計算設定のパラメータを極力少なくすることや、2次元検出器で測定した位置ごとの多数のスペクトルを一括で処理するスクリプトを用意するなど、共鳴イメージングデータを容易に解析できることを目指している。

### 1.3. RAIM の機能

RAIM は断面積計算とそれを利用した共鳴吸収スペクトルの計算、また測定データに対するスペクトルフィッティングによる解析を行うことができる。以下、それぞれの機能について説明する。

#### 1.3.1. 断面積計算

RAIM では核データ処理コード FREN DY<sup>2)</sup>を用いて断面積計算を行う。FREN DY は核データファイルから共鳴パラメータ等を読み取って断面積計算を行い、数値計算コードで使用する断面積ライブラリとして出力するコードである。RAIM では FREN DY の断面積計算機能を利用して、Single-Level Breit-Wigner、Multi-Level Breit-Wigner、Reich-Moore 等の近似による断面積計算を行うことができる。使用できる核データのフォーマットは ENDF-6 であり、評価済核データライブラリ JENDL-5<sup>3)</sup>などが対象となる。

また、ポイントワイズ形式のデータにも対応しており、核データファイルのポイントワイズデータおよびテーブル形式のデータをインポートすることも可能である。

### 1.3.2. 共鳴吸収スペクトル計算

実験で測定される共鳴吸収の TOF スペクトルは、断面積と密度から計算される中性子透過率や共鳴吸収量がパルス関数によって時間幅が広がった形となる。中性子透過率を  $Tr_{th}$ 、パルス関数を  $R$  とすると、測定される中性子透過率  $Tr_{obs}$  は以下の畳み込みの関係で表される。

$$Tr_{th}(E) = \exp(-\sigma(E)d)$$

$$Tr_{obs}(t) = \int_0^{\infty} R(t, E')T_{th}(E')dE'$$

ここで、 $E$  は中性子エネルギー、 $t$  は中性子の飛行時間、 $\sigma$  は全断面積、 $d$  は単位面積当たりの原子数密度を表す。同様にして、共鳴吸収量  $Y_{th}$  と測定される共鳴吸収量  $Y_{obs}$  の関係は以下の通りである。

$$Y_{th}(E) = (1 - \exp(-\sigma(E)d)) \frac{\sigma_{\gamma}(E)}{\sigma(E)}$$

$$Y_{obs}(t) = \int_0^{\infty} R(t, E')Y_{th}(E')dE'$$

ここで、 $\sigma_{\gamma}$  は捕獲断面積を表す。

RAIM は上記の式を用いてパルス関数を考慮した共鳴吸収スペクトルを計算することができる。また、J-PARC MLF の中性子源に入射する陽子ビームは時間的に 2 つのバンチを持つ、ダブルバンチ構造をしている。このダブルバンチ構造は特に中性子エネルギーの高い領域で共鳴ピークが 2 つに分離する等、減速材から放出される中性子の時間分布を表すパルス関数だけでは定義が難しい影響を与える。RAIM では陽子ビームのパルス構造を別に入力することにより、この影響を考慮できるようにした。

### 1.3.3. スペクトルフィッティング

測定された共鳴吸収スペクトルデータに対して最小二乗法によるスペクトルフィッティングを行うことができる。求めることができるパラメータは、飛行距離と TOF のずれ、解析対象の元素以外による中性子吸収に伴う中性子の減衰（透過型 N-RAS）やガンマ線計数（即発ガンマ線型 N-RAS）を表すベースライン、試料に含まれる特定の元素の密度や温度、パルス関数のパラメータである。

ベースラインのフィッティングでは共鳴ピークの前後のスペクトルを以下に示す最大 4 次の多項式で近似し、対象とする核種の共鳴ピークの抽出を可能とする。

$$B(t) = \sum_{i=0}^4 a_i t^i = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4$$

透過型 N-RAS のデータに対してはベースラインを乗算し、即発ガンマ線型 N-RAS のデータに対してはベースラインを加算する。フィッティングでは多項式の各項の係数を調整する。

RAIM では試料に含まれる元素（または複数の元素から成る物質）を定義する際に、複数の

元素を定義して元素ごとに密度を設定することができる。フィッティングする場合には、指定した元素の密度を求めることができる。例えば、元素 A、B、C で構成された試料を定義した場合、元素 A だけの密度を求めたり、元素 B と元素 C の密度を求めたりすることができる。このとき、各元素を構成する核種の比率は固定されるので、天然存在比や組成を崩さずに特定の元素の密度を求めることができる。

スペクトルフィッティングを行う際に、まずベースラインを求めてから共鳴ピークへのフィットを行い、元素の密度を求める場合がある。RAIM ではこの 2 段階のフィッティングプロセスを自動的に実行することができる。本書ではこれを連続スペクトルフィッティングと呼ぶ。

通常、フィッティングで温度を求める場合には、温度を変えた断面積計算を何度も繰り返す必要があるが、近似による断面積計算は複雑で計算に長い時間がかかる。温度を求める手法として、あらかじめ複数の温度における共鳴吸収スペクトルを計算し、それらの内挿から温度を求める機能を設けた。これを RAIM では温度マップと呼んでいる。例えば、Fig. 1 に示すような試料温度が 270, 280, 290, 300, 310 K の各点における透過率を計算した 3 次元のデータ（温度マップ）を作成しておくことで、この範囲内の任意の温度点における透過率を内挿により求めることができる。温度マップを利用したフィッティングの場合、内挿による簡易な計算で共鳴吸収スペクトルを得られるため、大幅に計算コストを削減することができる。

RAIM は J-PARC MLF BL10 や BL22 で使用することを念頭に開発されているため、それらの装置のパルス関数を再現できる関数として改良型の Cole-Windsor 関数<sup>4)</sup>を採用した。改良型の Cole-Windsor 関数を以下に示す。

$$F(t) = C[(1 - R)F_1(t) + RF_2(t)]$$

$$F_1(t) = F_2(t) = \exp\left[-\frac{1}{2}\left(\frac{\Delta t}{\sigma_1}\right)^2\right] \quad (t < t_0)$$

$$F_1(t) = \exp\left[-\frac{1}{2}\left(\frac{\Delta t}{\sigma_2}\right)^2\right] \quad (t_0 < t < t_0 + \gamma_1\sigma_2^2)$$

$$F_2(t) = \exp\left[-\frac{1}{2}\left(\frac{\Delta t}{\sigma_2}\right)^2\right] \quad (t_0 < t < t_0 + \gamma_2\sigma_2^2)$$

$$F_1(t) = \exp\left[\frac{(\gamma_1\sigma_2)^2}{2} - \gamma_1\Delta t\right] \quad (t_0 + \gamma_1\sigma_2^2 < t)$$

$$F_2(t) = \exp\left[\frac{(\gamma_2\sigma_2)^2}{2} - \gamma_2\Delta t\right] \quad (t_0 + \gamma_2\sigma_2^2 < t)$$

ここで  $\Delta t = t - t_0$  である。この関数のパラメータ ( $t_0$ ,  $\sigma_1$ ,  $\sigma_2$ ,  $\gamma_1$ ,  $\gamma_2$ ,  $C$ ,  $R$ ) はエネルギー依存性を有しているが、RAIM の機能を用いることでエネルギーごとのパラメータ値を求め、利用することができる。

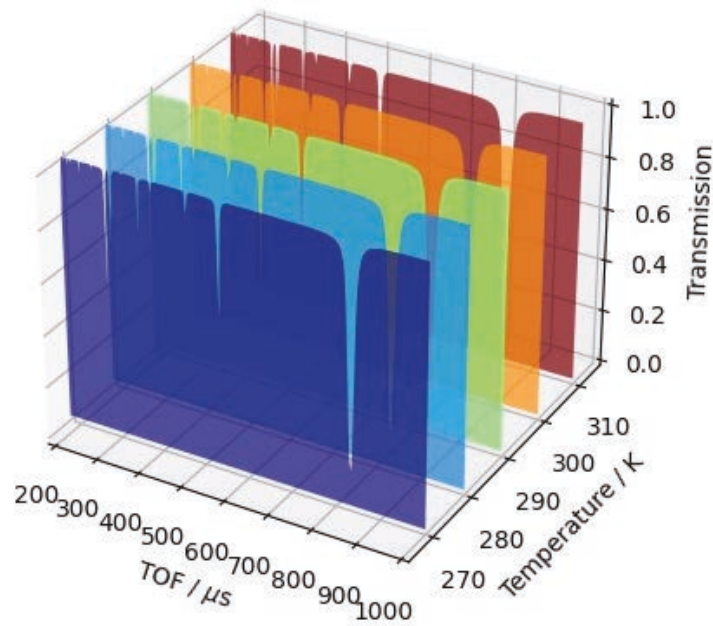


Figure 1 Example of temperature map

#### 1.3.4. イメージングデータ解析

イメージングデータを解析するために、位置ごとに測定した複数のスペクトルをまとめてフィッティングする機能がある。この機能を使うと、各データに対する密度や温度の値を CSV 形式のファイルとしてまとめて出力することができる。

#### 1.4. 動作環境

RAIM は Python3.6 以降が動作する Windows Subsystem on Linux (WSL) を含む Linux 計算機で動作する。開発は Ubuntu 20.04 で行っているが、RedHat 系 Linux など他の Linux ディストリビューションでも動作する。必要なソフトウェアを Table 1 に示す。

Table 1 Required software

項目	要件
プログラミング言語	Python3.6+, C/C++ (C++11 対応コンパイラ)
C/C++ライブラリ	Boost (libboost-all-dev)
Python ライブラリ	numpy, scipy, matplotlib, pybind11

## 2. インストール

### 2.1. 概要

RAIM のインストールは大きく分けると

- FRENDY のコンパイル
- RAIM のスクリプト生成

の2つの手順に分けられる。本章では Ubuntu 20.04 へのインストール方法について説明する。

### 2.2. 事前準備

#### 2.2.1. 必要なソフトウェアのインストール

断面積計算を行う FRENDY は C/C++ で書かれており、コンパイラおよびライブラリをインストールする必要がある。

```
$ sudo apt install g++ libboost-all-dev
```

RAIM 本体は Python で書かれており、Python3.6 以上が必要となる。環境に Python3.6 以上がインストールされていない場合はインストールを行っておく。ただし、`python3` コマンドおよび `python3-config` コマンドで使用する Python が呼び出されるように環境を構築しておくことが望ましい。Python3 の準備ができていれば必要なライブラリのインストールを行う。

```
$ pip install numpy scipy matplotlib pybind11
```

#### 2.2.2. インストールディレクトリ

RAIM は任意のディレクトリにインストール可能であり、以下のコマンドにより任意のディレクトリにソースコードを展開する。

```
$ tar zxvf raim.tar.gz
```

“raim”には Table 2 に示すフォルダが含まれている。ただし、一部のフォルダについては省略して示している。

Table 2 Directory structure

フォルダ名	説明
raim	RAIM ルートディレクトリ
raim/src/cpp/frendy	FRENDY ソースコード
raim/src/python/lib	RAIM ソースコード
raim/src/python/script	RAIM スクリプトファイル
raim/input	インプットファイル
raim/data/measurement	測定データ
raim/data/pulse	パルス関数設定ファイル
raim/shellscript	RAIM 実行用シェルスクリプト

### 2.3. FRENDY のコンパイル

RAIM では C++ で記述された FRENDY の機能を Python から呼び出すために `pybind11` という Python のモジュールを使用する。この `pybind11` によるラッパーのソースコードが含まれているディレクトリに移動し、コンパイルする。ただし、`python3` コマンドおよび `python3-config` コマンドが有効でない環境においては Makefile の **PYINCLUDE** と **PYCONFIG** を修正する。

```
$ cd raim/src/cpp/frendy/Wrapper
$ make
```

上記のコマンドにより共有ライブラリが生成する。環境によって異なるが、例えば Python3.10 が有効な環境の場合、`FrendyWrapper.cpython-310-x86_64-linux-gnu.so` というファイル名となる。

### 2.4. RAIM 実行用スクリプト生成

RAIM ルートディレクトリにおいて `install.sh` を実行する。このスクリプトは `raim/shellscript` フォルダに RAIM 実行用シェルスクリプトを生成する。

```
$ cd raim
$ ./install.sh
```



### 3. 計算方法

#### 3.1. 概要

RAIM は共鳴吸収スペクトルのシミュレーション機能と測定データに対するスペクトルフィッティング機能を有する。それぞれの計算は、目的の計算に応じた Python スクリプトを実行することで行うことができる。また、計算条件の設定はインプットファイルを用いて行う。

2.4.節の説明で生成したシェルスクリプトでは、Python スクリプトを実行するための環境設定やインプットファイルの指定を行っており、ユーザーはこのシェルスクリプトを実行することで計算を行う。

RAIM は核データファイルから断面積を計算するため、ユーザーは核データファイルを準備する必要がある。また、パルス関数を畳み込んだ共鳴吸収スペクトルを計算するため、パルス関数のパラメータおよび加速器からのビームの時間構造を定義したファイルも必要になる。

#### 3.2. 共通する計算手順

##### 3.2.1. 必要なファイルの準備

###### ● 核データファイル

核データファイルは日本原子力研究開発機構の核データ研究グループのウェブサイト ([https://www.ndc.jaea.go.jp/jendl/j5/j5\\_J.html](https://www.ndc.jaea.go.jp/jendl/j5/j5_J.html)) からのダウンロード等により ENDF-6 フォーマットのものを用意する必要がある。サンプルインプットファイルでは、JENDL-5 の核データファイルを `raim/data/jendl5` に保存することを想定している。拡張子は任意である。

###### ● パルス関数のパラメータ定義ファイル

RAIM では J-PARC MLF BL10 および BL22 でのパルス関数を計算できるように、1.3.3 項で示した改良型の Cole-Windsor 関数が実装されている。この関数のパラメータ ( $t_0$ ,  $\sigma_1$ ,  $\sigma_2$ ,  $\gamma_1$ ,  $\gamma_2$ ,  $C$ ,  $R$ ) のエネルギー依存性を計算するためのパラメータを定義するファイルが `raim/data/pulse/BL10.dat` および `BL22.dat` である。各行の数値は以下に示す各パラメータのエネルギー依存性を表す関数の係数  $a$ ,  $b$ ,  $c$  である。

$$t_0(E) = a + bE^c$$

$$\sigma_1(E) = a + bE^c$$

$$\sigma_2(E) = a + bE^c$$

$$\gamma_1(E) = a + bE^c$$

$$\gamma_2(E) = a + bE^c$$

$$R(E) = a + b \exp(cE)$$

$$C(E) = a \exp(bE^c)$$

他の装置における計算を行う場合には、この数値を変更したファイルを準備する。なお、#で始まる行はコメント行である。

- 陽子ビームの時間構造定義ファイル

MLF に入射する陽子ビームのダブルバンチ構造を再現するため、陽子ビームの時間構造定義ファイル `raim/data/pulse/double.dat` が準備されている。このファイルは 1 列目が時間（単位： $\mu\text{s}$ ）、2 列目が強度（任意単位）を表している。

ダブルバンチ以外の構造（例えばシングルバンチ構造）を計算したい場合、このファイルを参考にして時間構造を記述したファイルを準備する。なお、`#`で始まる行はコメント行である。

- 測定データファイル

スペクトルフィッティングを行う場合には RAIM で読み込める形式にした測定データファイルが必要である。RAIM が対応している測定データの形式は、1 列目：TOF、2 列目：透過率または共鳴吸収量、3 列目：誤差が空白区切りで記述されたテキストファイルである。

### 3.2.2. インプットファイルの作成

計算の設定を記述したインプットファイルを作成し、`raim/input` に保存する。インプットファイルの内容は次節において説明する。

### 3.2.3. シェルスクリプトの実行

`raim/shellscrip` において RAIM 実行用シェルスクリプトを実行する。このシェルスクリプト内で定義されている変数 `input_file` の値を変更してインプットファイルの指定を行う。

### 3.2.4. 計算結果ファイル

計算結果等のファイルはインプットファイルで指定したフォルダに保存される。計算中は計算内容等の情報が標準出力に表示されるが、その情報は `summary.txt` という名前のファイルに出力される。また、インプットファイルで指定したデータファイル（断面積や共鳴吸収スペクトルなど）や FRENDY の計算ログファイルも出力される。

## 3.3. インプットファイル

### 3.3.1. 概要

インプットファイルは複数のグループで構成される。各グループは **&グループ名** で始まり、`/` で終わる。各グループにおいて、**変数名 = 値** の形式で計算条件を設定していく。

値が文字列の場合、その値はシングルクォテーション（`'`）またはダブルクォーテーション（`"`）で囲む。また、`#` または `!` 以降の文字列はコメントとして扱われる。

以下、各グループの説明を行う。

### 3.3.2. &control

計算全般に関する設定を行う。Default が設定されている変数は省略可能である。このグループは省略することはできない。Table 3 に変数の説明を示す。

Table 3 Parameters of &control

変数名	型：値	Default	説明
project	文字列	project1	プロジェクト名を設定する。出力ファイルの保存先フォルダ名となる。
proj_ow	整数： 0/1	1	既に同じプロジェクトが存在する場合、(1)出力ファイルを上書きするか、(0)しないかを選択する。(0)の場合、 <b>project</b> に日時を付与したフォルダに出力ファイルを保存する。
sim	整数： 1/2/3	1	シミュレーション計算において、(1)断面積計算、(2)パルス関数を畳み込まない共鳴吸収スペクトル計算、(3)パルス関数を畳み込んだ透過率計算の選択を行う。
fit	整数： 0/1		フィッティング計算において、(1)フィッティングを行うか、(0)行わないかを選択する。(0)は初期値の確認等に使用する。
pulse_fit	整数： 0/1	0	フィッティング計算において、(0)パルス関数を最適化しないモードか、(1)最適化するモードかを選択する。
fn_xs	文字列	xs	計算した断面積を出力する際のファイル名のプレフィックス。後に設定する <b>material</b> ごとに断面積が出力される。
fn_tr0	文字列	tr0	パルス関数を畳み込まない共鳴吸収スペクトルを出力する際のファイル名。
fn_trr	文字列	trR	パルス関数を畳み込んだ共鳴吸収スペクトルを出力する際のファイル名。
fn_fiti	文字列	initial	フィッティング計算において、初期値で計算した共鳴吸収スペクトルを出力する際のファイル名。
fn_fitf	文字列	final	フィッティング計算において、フィッティング結果で計算した共鳴吸収スペクトルを出力する際のファイル名。
format	整数： 1/2/3	1	出力ファイルのフォーマットを指定する。(1)CSV、(2)テキスト（スペース区切り）、(3)(1)と(2)の両方
outdir	文字列	output	プロジェクトフォルダの保存先フォルダ名。 <b>raim</b> からの相対パスで指定する。
matplotlib	整数： 0/1	0	計算終了時に <b>matplotlib</b> によるグラフの表示を(1)行うか、(0)行わないかを選択する。
show_baseline	整数： 0/1	0	<b>matplotlib</b> によるグラフにベースラインを(1)表示するか、(0)表示しないかを選択する。

### 3.3.3. &xs

断面積計算を行う際の中性子エネルギーの範囲の設定を行う。このグループは省略することはできない。Table 4 に変数の説明を示す。

Table 4 Parameters of &xs

数名	型	Default	説明
e_l	実数		断面積計算を行う際のエネルギーの下限値[eV]
e_h	実数		断面積計算を行う際のエネルギーの上限値[eV]
enum	整数		断面積計算を行う際のエネルギーの群数を設定する。正の場合は対数、負の場合は線形で等分点が与えられる。

### 3.3.4. &material

試料に含まれる元素(物質)の定義を行う。このグループは複数定義することが可能であり、**&material1**、**&material2** の様に名前が重複しないようにグループ名の末尾に番号等を付与する。1つ以上の**&material** を定義する必要がある。Table 5 に変数の説明を示す。

Table 5 Parameters of &material

変数名	型	Default	説明
elm*	文字列, 実数(, 文字列)		各核種の核データファイルパスとこの元素での原子数比を「,」区切りで指定する。 <i>elm1</i> 、 <i>elm2</i> の様にして複数の元素を設定可能。ポイントワイズデータの場合、存在比の後に「,」区切りで <i>pw</i> と記述する。
dens	実数		元素の密度を体積密度[g/cm <sup>3</sup> ]で指定する。 <i>area_dens</i> で密度を定義する場合、省略可能。
thickness	実数		元素の厚さを[cm]で指定する。 <i>area_dens</i> で密度を定義する場合、省略可能。
area_dens	実数		元素の密度を面密度[atoms/barn]で指定する。 <i>dens</i> と <i>thickness</i> で密度を定義する場合、 <i>area_dens</i> を定義するとエラーとなる。
teff	実数		元素の温度(実効温度 <sup>1</sup> )を[K]で指定する。
pw*	文字列, 実数		温度マップを利用した計算時にポイントワイズデータのファイルパスとその温度を「,」区切りで指定する。 <i>pw1</i> 、 <i>pw2</i> の様にして複数の温度点を設定する。温度マップを利用しない場合、省略可能。

<sup>1</sup> 共鳴断面積の Doppler 拡がりの計算において理想ガスモデルを使用する際、試料が固体結晶である場合には原子同士の結合の効果を考慮する必要があるため、実際の試料の温度の代わりに実効温度を用いて計算を行う<sup>5)</sup>。

### 3.3.5. &transmission

透過率計算を行う際の設定を行う。Default が設定されている変数は省略可能である。インプットファイルに&fitting グループが設定されている場合、このグループは無視される。また、&transmission と&capture を同時に設定している場合、エラーメッセージが出力されて計算ができない。また、&control グループにおいて sim=1 を設定している（断面積計算のみを行う）場合はこのグループを省略できる。Table 6 に変数の説明を示す。

Table 6 Parameters of &transmission

変数名	型	Default	説明
t_l	実数		透過率計算を行う際の TOF の下限値[μs]。&xs グループで設定したエネルギー範囲を外れた値を設定した場合、その範囲の透過率は 0 になる。0 を設定した場合、エネルギーが無限大になってしまうため、最初の TOF ビンを切り詰めて計算を行う。
t_h	実数		透過率計算を行う際の TOF の上限値[μs]。&xs グループで設定したエネルギー範囲を外れた値を設定した場合、その範囲の透過率は 0 になる。
tnum	整数		TOF のビン数を設定する。正の場合は対数、負の場合は線形で等分点が与えられる。
e_l	実数		透過率計算を行う際のエネルギーの下限値[eV]。&xs グループで設定したエネルギー範囲を外れた値を設定した場合、その範囲の透過率は 0 になる。
e_h	実数		透過率計算を行う際のエネルギーの上限値[eV]。&xs グループで設定したエネルギー範囲を外れた値を設定した場合、その範囲の透過率は 0 になる。
enum	整数		エネルギーの群数を設定する。正の場合は対数、負の場合は線形で等分点が与えられる。
L	実数	18.0	飛行距離[m]を設定する。
td	実数	0.0	TOF のずれ[μs]を設定する。+方向が遅れ、-方向が進みである。
a0	実数	1.0	ベースラインを表す多項式の 0 次の係数
a1	実数	0.0	ベースラインを表す多項式の 1 次の係数
a2	実数	0.0	ベースラインを表す多項式の 2 次の係数
a3	実数	0.0	ベースラインを表す多項式の 3 次の係数
a4	実数	0.0	ベースラインを表す多項式の 4 次の係数

t\_l、t\_h、tnum と e\_l、e\_h、enum はどちらか一方のみ設定すればよい。また、計算範囲の設定は TOF がエネルギーに優先される。つまり、t\_l、t\_h、tnum が設定されている場合、e\_l、e\_h、enum は無視される。

### 3.3.6. &capture

共鳴吸収量計算を行う際の設定を行う。Default が設定されている変数は省略可能である。インプットファイルに&fitting グループが設定されている場合、このグループは無視される。また、&transmission と&capture を同時に設定している場合、エラーメッセージが出力されて計算ができない。また、&control グループにおいて sim=1 を設定している場合はこのグループを省略できる。Table 7 に変数の説明を示す。

Table 7 Parameters of &capture

変数名	型	Default	説明
t_l	実数		共鳴吸収量計算を行う際の TOF の下限値[μs]。&xs グループで設定したエネルギー範囲を外れた値を設定した場合、その範囲の共鳴吸収量は 0 になる。0 を設定した場合、エネルギーが無限大になってしまうため、最初の TOF ビンを切り詰めて計算を行う。
t_h	実数		共鳴吸収量計算を行う際の TOF の上限値[μs]。&xs グループで設定したエネルギー範囲を外れた値を設定した場合、その範囲の共鳴吸収量は 0 になる。
tnum	整数		TOF のビン数を設定する。正の場合は対数、負の場合は線形で等分点が与えられる。
e_l	実数		共鳴吸収量計算を行う際のエネルギーの下限値[eV]。&xs グループで設定したエネルギー範囲を外れた値を設定した場合、その範囲の共鳴吸収量は 0 になる。
e_h	実数		共鳴吸収量計算を行う際のエネルギーの上限値[eV]。&xs グループで設定したエネルギー範囲を外れた値を設定した場合、その範囲の共鳴吸収量は 0 になる。
enum	整数		エネルギーの群数を設定する。正の場合は対数、負の場合は線形で等分点が与えられる。
L	実数	18.0	飛行距離[m]を設定する。
td	実数	0.0	TOF のずれ[μs]を設定する。+方向が遅れ、-方向が進みである。
a0	実数	1.0	ベースラインを表す多項式の 0 次の係数
a1	実数	0.0	ベースラインを表す多項式の 1 次の係数
a2	実数	0.0	ベースラインを表す多項式の 2 次の係数
a3	実数	0.0	ベースラインを表す多項式の 3 次の係数
a4	実数	0.0	ベースラインを表す多項式の 4 次の係数
c	実数	1.0	共鳴吸収量の強度を規格化する係数

&transmission と同様に、t\_l、t\_h、tnum と e\_l、e\_h、enum はどちらか一方のみ設定すればよい。また、計算範囲の設定は TOF がエネルギーに優先される。つまり、t\_l、t\_h、tnum が設定されている場合、e\_l、e\_h、enum は無視される。

### 3.3.7. &fitting

スペクトルフィッティングを行う際の設定を行う。Default が設定されている変数は省略可能である。&control グループで fit が定義されていない場合、このグループは無視される。Table 8 に変数の説明を示す。

Table 8 Parameters of &fitting (1/3)

変数名	型：値	Default	説明
meas_file	文字列		測定データのファイルパスを設定する。複数ファイルを一括でフィッティングする場合には、フォルダ内の任意の 1 つのファイルを指定する。
adjust	文字列		どのパラメータをフィットするか選択する。 path：飛行距離と TOF のずれ base：ベースライン dens：密度 base_dens：ベースラインから密度の連続フィット temp：温度 temp_map：温度（温度マップでフィットする場合） temp_pw：温度（ポイントワイズデータを利用した温度マップでフィットする場合） c：共鳴吸収量の強度の規格化係数 pulse：パルス関数のパラメータ（t0, sig1, sig2, g1, g2, ratio）
L	実数		飛行距離[m]を設定する。adjust=“path”の場合、初期値となる。
td	実数		TOF のずれ[μs]を設定する。adjust=“path”の場合、初期値となる。
teff	実数		実効温度[K]を設定する。 adjust=“temp”、“temp_map”、“temp_pw”の場合、初期値となる。それ以外の場合には&material の温度が計算に使用される。
teff_list	実数		温度マップを作成する際の温度点。「,」区切りで 4 点以上設定する必要がある。adjust=“temp_map”の場合に有効。

Table 8 Parameters of fitting (2/3)

変数名	型:値	Default	説明
a0	実数	1.0	ベースラインを表す多項式の 0 次の係数。adjust=“base”または“base_dens”の場合、初期値となる。
a1	実数	0.0	ベースラインを表す多項式の 1 次の係数。adjust=“base”または“base_dens”の場合、初期値となる。
a2	実数	0.0	ベースラインを表す多項式の 2 次の係数。adjust=“base”または“base_dens”の場合、初期値となる。
a3	実数	0.0	ベースラインを表す多項式の 3 次の係数。adjust=“base”または“base_dens”の場合、初期値となる。
a4	実数	0.0	ベースラインを表す多項式の 4 次の係数。adjust=“base”または“base_dens”の場合、初期値となる。
c	実数	1.0	共鳴吸収量の強度を規格化する係数。adjust=“c”の場合、初期値となる。
t0	実数		パルス関数のパラメータ $t_0$ を設定する。pulse_fit=1 の場合に必要となる。adjust=“pulse”の場合、初期値となる。
sig1	実数		パルス関数のパラメータ $\sigma_1$ を設定する。pulse_fit=1 の場合に必要となる。adjust=“pulse”の場合、初期値となる。
sig2	実数		パルス関数のパラメータ $\sigma_2$ を設定する。pulse_fit=1 の場合に必要となる。adjust=“pulse”の場合、初期値となる。
g1	実数		パルス関数のパラメータ $\gamma_1$ を設定する。pulse_fit=1 の場合に必要となる。adjust=“pulse”の場合、初期値となる。
g2	実数		パルス関数のパラメータ $\gamma_2$ を設定する。pulse_fit=1 の場合に必要となる。adjust=“pulse”の場合、初期値となる。
ratio	実数		パルス関数のパラメータ $R$ を設定する。pulse_fit=1 の場合に必要となる。adjust=“pulse”の場合、初期値となる。
fit_material	文字列		adjust=“dens”または“base_dens”の場合に、密度を求める対象となる元素をグループ名で指定する。複数の元素を指定する場合は「,」区切りで指定する。
fit_range*	実数, 実数		フィッティング範囲を TOF [ $\mu$ s]で指定する。下限と上限を「,」区切りで指定する。また、fit_range1、fit_range2 の様に複数の範囲を指定することができる。
base*	実数, 実数		ベースラインをフィットする際のフィッティング範囲を TOF [ $\mu$ s]で指定する。下限と上限を「,」区切りで指定する。また、base1、base2 の様に複数の範囲を指定することができる。



Table 8 Parameters of &fitting (3/3)

変数名	型：値	Default	説明
level	実数	0.0	この値以下の透過率もしくは共鳴吸収量の TOF 範囲をフィッティングから除外する。
show	整数： 0/1	0	フィッティング中にイタレーション計算の途中経過を標準出力に(1)表示するか、(0)表示しないかを選択する。

adjust="base"または"base\_dens"においてベースラインのフィッティングを行う場合、a0 から a4 までのうち定義されている中で最大の次数のパラメータと同じ次数の多項式で計算が行われる。例えば、a0、a1、a2 が定義されている場合、2 次の多項式でベースライン計算を行う。すべてのパラメータが定義されていない場合、デフォルト値を係数とした 4 次の多項式で計算を行う。

### 3.3.8. &pulse

パルス関数に関する設定を行う。Default が設定されている変数は省略可能である。このグループは省略することができる。Table 9 に変数の説明を示す。

Table 9 Parameters of &pulse

変数名	型	Default	説明
pulse_file	文字列	(使用しない)	パルス関数のパラメータ設定ファイルを相対パスで指定する。指定しない場合、ソースコード内に記述されている J-PARC MLF BL10 の値で計算する。
proton_file	文字列	(使用しない)	陽子ビームの時間構造定義ファイルを相対パスで指定する。指定しない場合、ソースコード内に記述されているダブルバンチ構造の値で計算する。

## 3.4. シミュレーション計算

シミュレーション計算および測定データに対するフィッティングについて、ソースコードに同梱されるサンプルインプットファイルを用いて説明する。計算を実行するためのシェルスクリプトの内容は付録 A に、インプットファイルの内容は付録 B に示す。

### 3.4.1. 透過率のシミュレーション計算

透過率のシミュレーション計算に用いる実行用シェルスクリプトおよびサンプルインプットファイルは以下の通りである。

- 実行用シェルスクリプト (Fig. A1)

- run\_simulation\_tr.sh
- サンプル入力ファイル (Fig. B1)
  - sample\_sim\_tr.inp

他の計算で使用するシェルスクリプトは、入力ファイル名の指定(input\_file=)およびpythonスクリプトのファイル名の指定(script\_file=)が異なる以外は、前項の計算で使用したシェルスクリプトと同様である(ベースラインフィットと元素の密度のフィッティングを連続して行うrun\_continuous\_fitting\_tr.shとrun\_continuous\_fitting\_ca.shを除く)。入力ファイル名とpythonスクリプトのファイル名の一覧をTable A1に示す。

サンプル入力ファイルは2つの元素で構成された試料の透過率をシミュレーションする場合の計算条件が記述されている。

計算結果はraim/output/sample\_sim\_trに出力される。また、計算終了時に計算した透過率がmatplotlibのグラフで表示される(Fig. 2)。

比較的高いエネルギーの共鳴を見るため、&xsにおいて高いエネルギー領域の断面積を計算するように設定している(e\_l = 50.0、e\_h = 5000.0)。また、&transmissionでは速い領域のTOFを計算するように設定している(t\_l = 20.0、t\_h = 150.0)。

元素の組成や密度、温度は2つの&materialで設定しており、material 1は銅、material 2はタンタルである。

&pulseでは、パルス関数のパラメータとしてBL10のもの、陽子ビームの時間構造をダブルバンチ構造とし、それぞれのデータが格納されたファイル名を指定している。

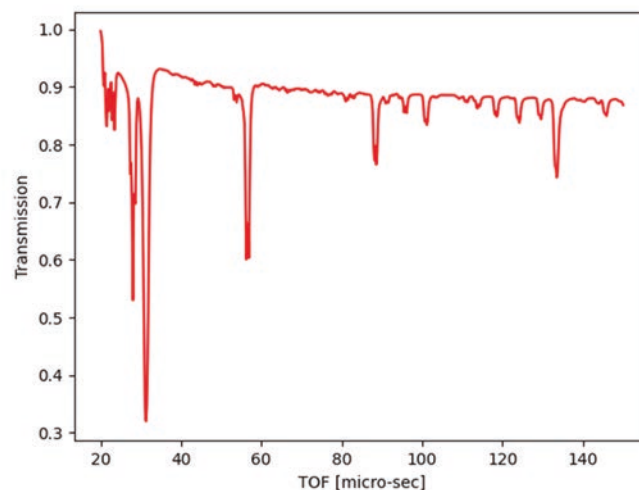


Figure 2 Simulated resonance transmission spectrum

### 3.4.2. 共鳴吸収量のシミュレーション計算

共鳴吸収量のシミュレーション計算に用いる実行用シェルスクリプトおよびサンプル入力ファイルは以下の通りである。

- 実行用シェルスクリプト
  - `run_simulation_ca.sh`
- サンプル入力ファイル (Fig. B2)
  - `sample_sim_ca.inp`

サンプル入力ファイルは2つの元素で構成された試料の共鳴吸収量をシミュレーションする場合の計算条件が記述されている。

計算結果は `raim/output/sample_sim_ca` に出力される。また、計算終了時に計算した共鳴吸収量が `matplotlib` のグラフで表示される (Fig. 3)。

タンタルの第1共鳴 (4.28 eV) を見るため、`&xs` において低いエネルギー領域の断面積を計算するように設定している (`e_l = 1.0`、`e_h = 2000.0`)。また、`&capture` では比較的遅い領域の TOF を計算するように設定している (`t_l = 20.0`、`t_h = 800.0`)。

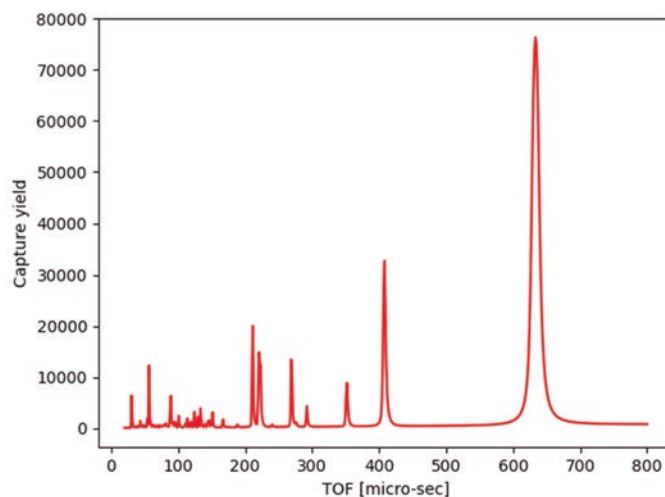


Figure 3 Simulated resonance capture spectrum

### 3.5. スペクトルフィッティング

#### 3.5.1. 透過率データのスペクトルフィッティング

透過率データのスペクトルフィッティングに用いる実行用シェルスクリプトおよびサンプル入力ファイルは以下の通りである。

- 実行用シェルスクリプト
  - `run_fitting_tr.sh`
- サンプル入力ファイル (Fig. B3)
  - `sample_fit_tr.inp`

サンプル入力ファイルはタンタルを含む試料の透過率をフィッティングする場合の計算条件が記述されている。フィッティング対象となる透過率データは、`meas_file = "data/measurement/Tr-Ta.dat"` のようにデータを格納したファイル名で指定する。

計算結果は `raim/output/sample_fit_tr` に出力される。また、計算終了時に測定データとフィットしたスペクトルが `matplotlib` のグラフで表示される (Fig. 4)。

フィッティングで求める物理量は元素の密度で、`&fitting` の `adjust = "dens"` で指定している。対象となる元素は `&material1` で定義している。`&material1` で設定している元素の密度がフィッティング計算における初期値となる。

フィッティング範囲は `fit_range0 = 800.0, 860.0` により、1つの領域 (800~860  $\mu\text{s}$ ) だけを設定している。

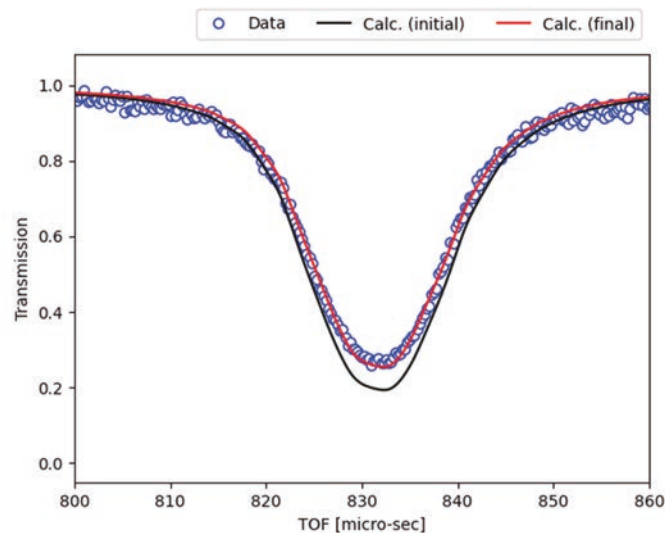


Figure 4 Fitted resonance transmission spectrum with measured data

### 3.5.2. 共鳴吸収量データのスペクトルフィッティング

共鳴吸収量データのスペクトルフィッティングに用いる実行用シェルスクリプトおよびサンプル入力ファイルは以下の通りである。

- 実行用シェルスクリプト
  - `run_fitting_ca.sh`
- サンプル入力ファイル (Fig. B4)
  - `sample_fit_ca.inp`

サンプル入力ファイルはタンタルを含む試料の共鳴吸収量をフィッティングする場合の計算条件が記述されている。

計算結果は `raim/output/sample_fit_ca` に出力される。また、計算終了時に測定データとフィットしたスペクトルが `matplotlib` のグラフで表示される (Fig. 5)。

フィッティングで求める物理量は共鳴吸収量の規格化係数（検出器の検出効率や立体角の効果を考慮するためのファクター）で、`&fitting` の `adjust = "c"` で指定している。

ベースラインは0次の多項式（一定値）として `a0=1` で指定している。ベースラインとして考慮する範囲はパラメータ `base` で指定しており、この計算では `base0 = 675, 800`、`base1 = 450, 600`、`base2 = 305, 340`、`base3 = 65, 80` である。

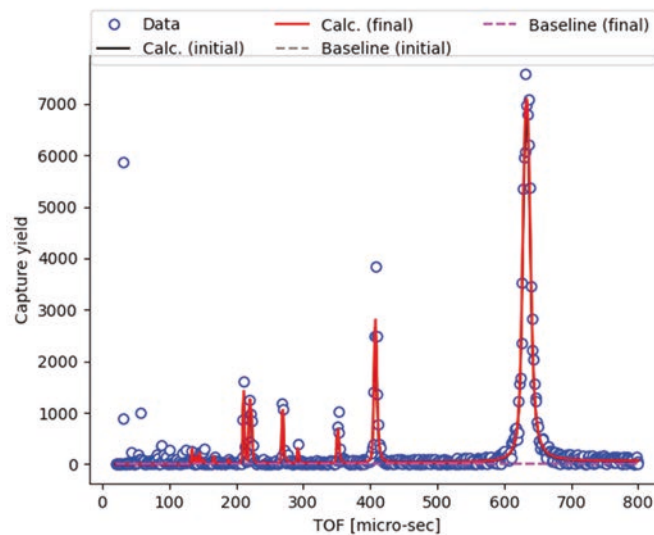


Figure 5 Fitted resonance capture spectrum with measured data

### 3.6. 連続スペクトルフィッティング

1.3.3 項で説明した連続スペクトルフィッティングを実行するスクリプトを用意している。

#### 3.6.1. 透過率データの連続スペクトルフィッティング

透過率データの連続スペクトルフィッティングに用いる実行用シェルスクリプトおよびサンプルインプットファイルは以下の通りである。

- 実行用シェルスクリプト (Fig. A2)
  - `run_continuous_fitting_tr.sh`
- サンプルインプットファイル (Fig. B5)
  - `sample_cont_fit_tr.inp`

実行用シェルスクリプトでは、`base_dens` パラメータを指定し、Python スクリプト実行時の引数として読み込んでいる。

`base_dens=1` の場合、2 段階目のフィッティングでは密度とベースラインパラメータを同時に求める。`base_dens=0` の場合、2 段階目のフィッティングでは密度のみを求める。

サンプルインプットファイルはタンタルを含む試料の透過率を連続フィッティングする場合の計算条件が記述されている。まず、ベースラインをフィットし、次に密度を求める。

計算結果は `raim/output/sample_cont_fit_tr` に出力される。また、2 段階目の計算終了時に測定データとフィットしたスペクトルが `matplotlib` のグラフで表示される。

ベースラインのフィッティング範囲は 5 つの範囲を `base0`~`base4` の 5 つの変数で設定している。密度を求める際のフィッティング範囲は `fit_range0=800.0, 860.0` により、1 つの領域 (800~860  $\mu\text{s}$ ) だけを設定している。

### 3.6.2. 共鳴吸収量データの連続スペクトルフィッティング

共鳴吸収量データの連続スペクトルフィッティングに用いる実行用シェルスクリプトおよびサンプルインプットファイルは以下の通りである。

- 実行用シェルスクリプト
  - `run_continuous_fitting_ca.sh`
- サンプルインプットファイル (Fig. B6)
  - `sample_cont_fit_ca.inp`

実行用シェルスクリプトでは、前項と同様に `base_dens` を指定する。`base_dens=1` の場合、2 段階目のフィッティングでは密度とベースラインパラメータを同時に求める。一方で、`base_dens=0` の場合、2 段階目のフィッティングでは密度のみを求める

サンプルインプットファイルはタンタルを含む試料の共鳴吸収量を連続フィッティングする場合の計算条件が記述されている。まず、ベースラインをフィットし、次に密度を求める。

計算結果は `raim/output/sample_cont_fit_ca` に出力される。また、2 段階目の計算終了時に測定データとフィットしたスペクトルが `matplotlib` のグラフで表示される。このとき、ベースラインも重ねて表示される。

ベースラインのフィッティング範囲は 5 つの範囲を設定している。密度を求める際のフィッティング範囲は 1 つの領域 (800~860  $\mu\text{s}$ ) だけを設定している。

## 3.7. 温度マップを用いたスペクトルフィッティング

### 3.7.1. 透過率データのスペクトルフィッティング

温度マップを用いた透過率データのスペクトルフィッティングに用いる実行用シェルスクリプトおよびサンプルインプットファイルは以下の通りである。

- 実行用シェルスクリプト
  - `run_fitting_tr_tmap.sh`
- サンプルインプットファイル (Fig. B7)
  - `sample_fit_tr_tmap.inp`

サンプルインプットファイルはタンタルを含む試料の透過率をフィッティングする場合の計算条件が記述されている。

計算結果は `raim/output/sample_fit_tr_tmap` に出力される。また、計算終了時に測定データとフィットしたスペクトルが `matplotlib` のグラフで表示される。

温度マップを利用したフィッティングで温度を求める。温度マップは 270、280、290、300、310 K における断面積を計算して作成している。作成する温度は、`teff_list` で指定する。温度の初期値は 300 K である。

フィッティング範囲は 1 つの領域 (800~860  $\mu\text{s}$ ) だけを設定している。

### 3.7.2. 透過率データのスペクトルフィッティング (ポイントワイズデータ使用)

温度マップはポイントワイズデータを利用して作成することもできる。ポイントワイズデータを利用した温度マップを用いた透過率データのスペクトルフィッティングに用いる実行用シェルスクリプトおよびサンプルインプットファイルは以下の通りである。

- 実行用シェルスクリプト
  - `run_fitting_tr_tmap.sh`
- サンプルインプットファイル (Fig. B8)
  - `sample_fit_tr_tmap_pw.inp`

サンプルインプットファイルはタンタルを含む試料の透過率をフィッティングする場合の計算条件が記述されている。

計算結果は `raim/output/sample_fit_tr_tmap_pw` に出力される。また、計算終了時に測定データとフィットしたスペクトルが `matplotlib` のグラフで表示される。

温度マップを利用したフィッティングで温度を求める。断面積はポイントワイズデータを用いる。`&material1` において、`pw1,2,3,4` でポイントワイズデータの相対パスと温度を設定している。`elm1` はポイントワイズデータではない核データファイルを指定しているが、これは密度等の計算に必要な数値を取得するために指定している。温度の初期値は 300 K である。注意点として、指定方法が煩雑になることから現バージョンでは 1 つの元素に対して 1 つの核種のみに対応している。ただし、複数元素を設定することは可能である。

フィッティング範囲は 1 つの領域 (800~860  $\mu\text{s}$ ) だけを設定している。

### 3.7.3. 共鳴吸収量データのスペクトルフィッティング

温度マップを用いた共鳴吸収量データのスペクトルフィッティングに用いる実行用シェルスクリプトおよびサンプルインプットファイルは以下の通りである。

- 実行用シェルスクリプト
  - `run_fitting_ca_tmap.sh`
- サンプルインプットファイル (Fig. B9)
  - `sample_fit_ca_tmap.inp`

サンプルインプットファイルはタンタルと銅を含む試料の共鳴吸収量をフィッティングする場合の計算条件が記述されている。

計算結果は `raim/output/sample_fit_ca` に出力される。また、計算終了時に測定データとフィットしたスペクトルが `matplotlib` のグラフで表示される。

温度マップを利用したフィッティングで温度を求める。温度マップは 270、280、290、300、310 K における断面積を計算して作成している。温度の初期値は 300 K である。

フィッティング範囲は 1 つの領域 (600~675  $\mu\text{s}$ ) だけを設定している。

### 3.8. 複数データの一括スペクトルフィッティング

#### 3.8.1. 透過率データのスペクトルフィッティング

複数の透過率データのスペクトルフィッティングに用いる実行用シェルスクリプトおよびサンプルインプットファイルは以下の通りである。

- 実行用シェルスクリプト
  - `run_multi_fitting_tr.sh`
- サンプルインプットファイル (Fig. B10)
  - `sample_multi_fit_tr.inp`

サンプルインプットファイルはタンタルを含む試料の透過率をフィッティングする場合の計算条件が記述されている。フィッティング対象となる測定データは複数あり、これらをすべてフィッティングして結果を CSV ファイルにまとめて出力する。

計算結果は `raim/output/sample_multi_fit_tr` に出力される。`&control` の `matplotlib` に関わらず、計算終了時に結果のグラフは表示されない。

フィッティングで求める物理量は密度である。対象となる元素は `&material1` で定義している。`&material1` で設定している元素の密度がフィッティング計算における初期値となる。

測定データファイルの指定は `&fitting` の `meas_file` で行っている。複数のデータファイルを指定する場合、それらの内 1 つのファイルを指定することで、そのファイルが保存されているフォルダに保存されている測定データファイルすべてを解析対象とする。ただし、測定データの拡張子は `.dat` でなくてはならない。つまり、`data/measurement/imaging/` に保存されているファイルの内、拡張子が `.dat` のものをすべてフィッティングする。

フィッティング範囲は 1 つの領域 (800~860  $\mu\text{s}$ ) だけを設定している。

#### 3.8.2. 透過率データのスペクトルフィッティング (温度マップ使用)

複数データに対するスペクトルフィッティングについても温度マップを使用することができる。温度マップを利用した複数の透過率データのスペクトルフィッティングに用いる実行用シェルスクリプトおよびサンプルインプットファイルは以下の通りである。

- 実行用シェルスクリプト



- `run_multi_fitting_tr_tmap.sh`
- サンプルインプットファイル (Fig. B11)
  - `sample_multi_fit_tr_tmap.inp`

サンプルインプットファイルはタンタルを含む試料の透過率をフィッティングする場合の計算条件が記述されている。フィッティング対象となる測定データは複数あり、これらをすべてフィッティングして結果を CSV ファイルにまとめて出力する。

計算結果は `raim/output/sample_multi_fit_tr_tmap` に出力される。&control の `matplotlib` に関わらず、計算終了時に結果のグラフは表示されない。

温度マップを利用したフィッティングで温度を求める。温度マップは 270、280、290、300、310 K における断面積を計算して作成している。温度の初期値は 300 K である。

`data/measurement/imaging/` に保存されているファイルの内、拡張子が `.dat` のものをすべてフィッティングする。

フィッティング範囲は 1 つの領域 (800~860  $\mu\text{s}$ ) だけを設定している。

### 3.8.3. 共鳴吸収量データのスペクトルフィッティング

複数の共鳴吸収量データのスペクトルフィッティングに用いる実行用シェルスクリプトおよびサンプルインプットファイルは以下の通りである。

- 実行用シェルスクリプト
  - `run_multi_fitting_ca.sh`
- サンプルインプットファイル (Fig. B12)
  - `sample_multi_fit_ca.inp`

サンプルインプットファイルは銅とタンタルを含む試料の共鳴吸収量をフィッティングする場合の計算条件が記述されている。フィッティング対象となる測定データは複数あり、これらをすべてフィッティングして結果を CSV ファイルにまとめて出力する。

計算結果は `raim/output/sample_multi_fit_ca` に出力される。&control の `matplotlib` に関わらず、計算終了時に結果のグラフは表示されない。

フィッティングで求める物理量は密度である。対象となる元素は `&material2` で定義している。&material2 で設定している元素の密度がフィッティング計算における初期値となる。

`data/measurement/capture/imaging/` に保存されているファイルの内、拡張子が `.dat` のものをすべてフィッティングする。

フィッティング範囲は 1 つの領域 (600~675  $\mu\text{s}$ ) だけを設定している。

### 3.8.4. 共鳴吸収量データのスペクトルフィッティング (温度マップ使用)

温度マップを利用した複数の共鳴吸収量データのスペクトルフィッティングに用いる実行用シェルスクリプトおよびサンプルインプットファイルは以下の通りである。

- 実行用シェルスクリプト
  - `run_multi_fitting_ca_tmap.sh`
- サンプルインプットファイル (Fig. B13)
  - `sample_multi_fit_ca_tmap.inp`

サンプルインプットファイルはタンタルを含む試料の共鳴吸収量をフィッティングする場合の計算条件が記述されている。フィッティング対象となる測定データは複数あり、これらを一括でフィッティングして結果を CSV ファイルにまとめて出力する。

計算結果は `raim/output/sample_multi_fit_ca_tmap` に出力される。&control の `matplotlib` に関わらず、計算終了時に結果のグラフは表示されない。

温度マップを利用したフィッティングで温度を求める。温度マップは 270、280、290、300、310 K における断面積を計算して作成している。温度の初期値は 300 K である。

`data/measurement/capture/imaging/` に保存されているファイルの内、拡張子が `.dat` のものをすべてフィッティングする。

フィッティング範囲は 1 つの領域 (600~675  $\mu\text{s}$ ) だけを設定している。

## 謝辞

RAIM が断面積データの計算に使用する核データ処理コード FRENDY は、日本原子力研究開発機構 原子力科学研究所 原子力基礎工学研究センター 炉物理・熱流動研究グループによって開発され、公開されている。FRENDY を使用することにより、RAIM の開発を効率的に行うことが可能となった。

参考文献

- 1) Hasemi, H., Harada, M., Kai, T., Shinohara, T., Ooi, M., Sato, H., Kino, K., Segawa, M., Kamiyama, T., Kiyonagi, Y., Evaluation of nuclide density by neutron resonance transmission at the NOBORU instrument in J-PARC/MLF, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 773, 2015, pp.137-149.
- 2) Tada, K., Yamamoto, A., Kunieda, S., Konno, C., Kondo, R., Endo, T., Chiba, G., Ono, M., Tojo, M., Development of nuclear data processing code FRENDY version 2, Journal of Nuclear Science and Technology, Vol.61, 2024, pp.830-839.
- 3) Iwamoto, O., Iwamoto, N., Kunieda, S., Minato, F., Nakayama, S., Abe, Y., Tsubakihara, K., Okumura, S., Ishizuka, C., Yoshida, T., Chiba, S., Otuka, N., Sublet, J. C., Iwamoto, H., Yamamoto, K., Nagaya, Y., Tada, K., Konno, C., Matsuda, N., Yokoyama, K., Taninaka, H., Oizumi, A., Fukushima, M., Okita, S., Chiba, G., Sato, S., Ohta, M., Kwon, S., Japanese evaluated nuclear data library version 5: JENDL-5, Journal of Nuclear Science and Technology, Vol. 60, No. 1, 2023, pp.1-60.
- 4) Izumi, F., Asano, H., Murata, H., Watanabe, N., Rietveld analysis of powder patterns obtained by TOF neutron diffraction using cold neutron sources, Journal of Applied Crystallography, Vol. 20, 1987, pp.411-418.
- 5) Jackson, H. E, Lynn, J. E., Resonance Absorption of Neutrons by Crystals, Phys. Rev., Vol. 127, 1962, pp.461-468.

付録 A 実行用シェルスクリプト

```
#!/bin/bash
input_file=sample_sim_tr.inp
MAINPATH=/home/raim
export LD_LIBRARY_PATH=$MAINPATH/src/cpp/frendy/Wrapper:$LD_LIBRARY_PATH
export PYTHONPATH=$MAINPATH/src/python/lib:$PYTHONPATH
export PYTHONPATH=$MAINPATH/src/cpp:$PYTHONPATH
script_file=$MAINPATH/src/python/script/simulation_transmission_script.py
cd $MAINPATH
python3 $script_file ./input/$input_file
```

Figure A1 Shell script of transmission simulation (run\_simulation\_tr.sh)

```
#!/bin/bash
input_file=sample_cont_fit_tr.inp
MAINPATH=/home/raim
export LD_LIBRARY_PATH=$MAINPATH/src/cpp/frendy/Wrapper:$LD_LIBRARY_PATH
export PYTHONPATH=$MAINPATH/src/python/lib:$PYTHONPATH
export PYTHONPATH=$MAINPATH/src/cpp:$PYTHONPATH
script_file=$MAINPATH/src/python/script/continuous_fitting_transmission_script.py
base_dens=1
cd $MAINPATH
python3 $script_file ./input/$input_file $base_dens
```

Figure A2 Shell script of continuous fitting for transmission (run\_continuous\_fitting\_tr.sh)

Table A1 Name of input files and Python scripts specified in the executing shell script

章	シェルスクリプトファイル名	入力ファイル名	Python スクリプト名
3.4.1	run_simulation_tr.sh	sample_sim_tr.inp	simulation_transmission_script.py
3.4.2	run_simulation_ca.sh	sample_sim_ca.inp	simulation_capture_script.py
3.5.1	run_fitting_tr.sh	sample_fit_tr.inp	fitting_transmission_script.py
3.5.2	run_fitting_ca.sh	sample_fit_ca.inp	fitting_capture_script.py
3.6.1	run_continuous_fitting_tr.sh*	sample_cont_fit_tr.inp	continuous_fitting_transmission_script.py
3.6.2	run_continuous_fitting_ca.sh*	sample_cont_fit_ca.inp	continuous_fitting_capture_script.py
3.7.1	run_fitting_tr_tmap.sh	sample_fit_tr_tmap.inp	fitting_transmission_tmap_script.py
3.7.2	run_fitting_tr_tmap.sh	sample_fit_tr_tmap_pw.inp	fitting_transmission_tmap_script.py
3.7.3	run_fitting_ca_tmap.sh	sample_fit_ca_tmap.inp	fitting_capture_tmap_script.py
3.8.1	run_multi_fitting_tr.sh	sample_multi_fit_tr.inp	multi_fitting_transmission_script.py
3.8.2	run_multi_fitting_tr_tmap.sh	sample_multi_fit_tr_tmap.inp	multi_fitting_transmission_tmap_script.py
3.8.3	run_multi_fitting_ca.sh	sample_multi_fit_ca.inp	multi_fitting_capture_script.py
3.8.4	run_multi_fitting_ca_tmap.sh	sample_multi_fit_ca_tmap.inp	multi_fitting_capture_tmap_script.py

\* 追加パラメータあり

付録 B サンプルインプット

```

&control
project = "sample_sim_tr" # project name (default "project1")
proj_ow = 1 # overwrite project directory, No:0, Yes:1 (default 1)
sim = 3 # 1:xs, 2:tr, 3:tr(pulse) (default 1)
fn_xs = "xs" # file name of cross section (default "xs")
fn_tr0 = "tr0" # file name of transmission (default "tr0")
fn_trr = "trR" # file name of transmission (pulse) (default "trR")
format = 3 # 1:csv, 2:text, 3:csv and text (default 1)
outdir = "./output" # base directory (default "./output")
matplotlib = 1 # 0:no, 1:yes (default 0)
/

&xs
e_l = 50.0 # lower limit of neutron energy [eV]
e_h = 5000.0 # upper limit of neutron energy [eV]
enum = 2000 # number of group (+:logscal, -:linscale)
/

&material1
elm1 = "data/jendl5/Cu63.res", 0.6915 # ndata, abundance
elm2 = "data/jendl5/Cu65.res", 0.3085 # ndata, abundance
dens = 8.9 # density of sample [g/cm^3]
thickness = 0.2 # thickness of sample [cm]
teff = 318.0 # effective temperature [K] (default 300.0)
tsamp = 300.0 # sample temperature [K]
debye = 343 # debye temperature [K]
/

&material2
elm1 = "data/jendl5/Ta181.res", 1.0 # ndata, abundance
area_dens = 1.68e-4 # area density of sample [at/b]
teff = 300.0 # effective temperature [K] (default 300.0)
tsamp = 300.0 # sample temperature [K]
debye = 240 # debye temperature [K]
/

&transmission
t_l = 20.0 # lower limit of TOF [us]
t_h = 150.0 # upper limit of TOF [us]
tnum = -600 # number of TOF bin (+:logscal, -:linscale)
L = 18.0 # flight path [m] (default 18.0)
td = 1.0 # delay of TOF [us] (default 0.0)
/

&pulse
pulse_file = "data/pulse/BL10.dat" # parameters of pulse function (default none)
proton_file = "data/pulse/double.dat" # data of proton pulse shape (default none)
/

```

Figure B1 Sample input file of transmission simulation (sample\_sim\_tr.inp)

```

&control
project = "sample_sim_ca"      # project name (default "project1")
proj_ow = 1                    # overwrite project directory, No:0, Yes:1 (default 1)
sim = 3                        # 1:xs, 2:ca, 3:ca(pulse) (default 1)
fn_xs = "xs"                  # file name of cross section (default "xs")
fn_tr0 = "ca0"                # file name of capture yield (default "tr0")
fn_trr = "caR"                # file name of capture yield (pulse) (default "trR")
format = 3                     # 1:csv, 2:text, 3:csv and text (default 1)
outdir = "./output"           # base directory (default "./output")
matplotlib = 1                # 0:no, 1:yes (default 0)
/

&xs
e_l = 1.0                      # lower limit of neutron energy [eV]
e_h = 2000.0                  # upper limit of neutron energy [eV]
enum = 2000                   # number of group (+:logscal, -:linscale)
/

&material1
elm1 = "data/jendl5/Cu63.res", 0.6915 # ndata, abundance
elm2 = "data/jendl5/Cu65.res", 0.3085 # ndata, abundance
dens = 8.9                    # density of sample [g/cm^3]
thickness = 0.2               # thickness of sample [cm]
teff = 318.0                  # effective temperature [K] (default 300.0)
/

&material2
elm1 = "data/jendl5/Ta181.res", 1.0 # ndata, abundance
area_dens = 1.68e-4           # area density of sample [at/b]
teff = 300.0                  # effective temperature [K] (default 300.0)
/

&capture
t_l = 20.0                    # lower limit of TOF [us]
t_h = 800.0                   # upper limit of TOF [us]
tnum = -600                   # number of TOF bin (+:logscal, -:linscale)
L = 18.0                      # flight path [m] (default 18.0)
td = 1.0                      # delay of TOF [us] (default 0.0)
a0 = 100                      # baseline parameter a0
c = 100000                    # scale factor (default 1.0)
/

&pulse
pulse_file = "data/pulse/BL10.dat" # parameters of pulse function (default none)
proton_file = "data/pulse/double.dat" # data of proton pulse shape (default none)
/

```

Figure B2 Sample input file of capture yield simulation (sample\_sim\_ca.inp)

```

&control
project = "sample_fit_tr" # project name (default "project1")
proj_ow = 1 # overwrite project directory, No:0, Yes:1 (default 1)
fit = 1 # 0:simulation, 1:fitting
pulse_fit = 0 # 0:normal, 1:adjust/fix parameters of pulse function (default 0)
fn_xs = "xs_ta" # file name of cross section (default "xs")
fn_fiti = "initial_ta" # file name of initial fitting line (default "initial")
fn_fitf = "final_ta" # file name of final fitting line (default "final")
format = 3 # 1:csv, 2:text, 3:csv and text (default 1)
outdir = "./output" # base directory (default "./output")
matplotlib = 1 # 0:no, 1:yes (default 0)
/

&xs
e_l = 1.0 # lower limit of neutron energy
e_h = 100.0 # upper limit of neutron energy
enum = 500 # number of group (+:logscal, -:linscale)
/

&material1
elml = "data/jendl5/Ta181.res", 1.0 # path of nuclear data file
dens = 16.65 # density of sample [g/cm^3]
thickness = 0.003 # thickness of sample [cm]
teff = 300.0 # effective temperature [K] (default 300.0)
/

&fitting
meas_file = "data/measurement/Tr_Ta.dat"
# initial parameters
L = 23.7 # flight path [m]
td = 0 # delay of TOF [us]
# dens = 16.65 # density of sample [g/cm^3]
# thickness = 0.003 # thickness of sample [cm]
teff = 300.0 # effective temperature [K]
# material
fit_material = &material1
# fitting range
fit_range0 = 800.0, 860.0 # lower and upper limit of fitting range (TOF [us])
# mask
level = 0.0 # masking level of transmission (Default 0.0)
# adjusted parameter(s)
adjust = "dens" # Type of fitting parameter(s)
# path:L, delay, dens:density, temp:temperature
# pulse:pulse function, base:baseline
show = 1 # show progress of fitting; 0:no, 1:yes (Default 0)
/

&pulse
pulse_file = "data/pulse/BL10.dat" # parameters of pulse function (default none)
proton_file = "data/pulse/double.dat" # data of proton pulse shape (default none)
/

```

Figure B3 Sample input file of fitting for transmission data (sample\_fit\_tr.inp)

```

&control
project = "sample_fit_ca" # project name (default "project1")
proj_ow = 1 # overwrite project directory, No:0, Yes:1 (default 1)
fit = 1 # 0:simulation, 1:fitting
pulse_fit = 0 # 0:normal, 1:adjust/fix parameters of pulse function (default 0)
fn_xs = "xs" # file name of cross section (default "xs")
fn_fiti = "initial" # file name of initial fitting line (default "initial")
fn_fitf = "final" # file name of final fitting line (default "final")
format = 3 # 1:csv, 2:text, 3:csv and text (default 1)
outdir = "./output" # base directory (default "./output")
matplotlib = 1 # 0:no, 1:yes (default 0)
show_baseline = 1
/

&xs
e_l = 1.0 # lower limit of neutron energy
e_h = 100.0 # upper limit of neutron energy
enum = 500 # number of group (+:logscal, -:linscale)
/

&material1
elm1 = "data/jendl5/Cu63.res", 0.6915 # ndata, abundance
elm2 = "data/jendl5/Cu65.res", 0.3085 # ndata, abundance
dens = 8.9 # density of sample [g/cm^3]
thickness = 0.2 # thickness of sample [cm]
teff = 318.0 # effective temperature [K] (default 300.0)
/

&material2
elm1 = "data/jendl5/Ta181.res", 1.0 # path of nuclear data file
area_dens = 1.68e-4
teff = 300.0 # effective temperature [K] (default 300.0)
/

&fitting
meas_file = "data/measurement/capture/test_ta.dat"
# initial parameters
L = 18.0 # flight path [m]
td = 1.0 # delay of TOF [us]
# dens = 16.65 # density of sample [g/cm^3]
# thickness = 0.003 # thickness of sample [cm]
teff = 300.0 # effective temperature [K]
a0 = 1 # baseline parameter a0
c = 9000 # scale factor (default 1.0)
# material
fit_material = &material2
# fitting range
fit_range0 = 600.0,675.0 # lower and upper limit of fitting range (TOF [us])
# fit_range1 = 400.0,420.0 # lower and upper limit of fitting range (TOF [us])
# fit_range2 = 345.0,360.0 # lower and upper limit of fitting range (TOF [us])
base0 = 675,800
base1 = 450,600
base2 = 305,340
base3 = 65,80
# mask
level = 0.0 # masking level of transmission (Default 0.0)
# adjusted parameter(s)
adjust = "c" # Type of fitting parameter(s)
# path:L, delay, dens:density, temp:temperature
# pulse:pulse function, base:baseline
show = 1 # show progress of fitting; 0:no, 1:yes (Default 0)
/

&pulse
pulse_file = "data/pulse/BL10.dat" # parameters of pulse function (default none)
proton_file = "data/pulse/double.dat" # data of proton pulse shape (default none)
/

```

Figure B4 Sample input file of fitting for capture yield data (sample\_fit\_ca.inp)



```

&control
project = "sample_cont_fit_tr" # project name (default "project1")
proj_ow = 1 # overwrite project directory, No:0, Yes:1 (default 1)
fit = 1 # 0:simulation, 1:fitting
pulse_fit = 0 # 0:normal, 1:adjust/fix parameters of pulse function (default 0)
fn_xs = "xs_ta" # file name of cross section (default "xs")
fn_fiti = "initial_ta" # file name of initial fitting line (default "initial")
fn_fitf = "final_ta" # file name of final fitting line (default "final")
format = 3 # 1:csv, 2:text, 3:csv and text (default 1)
outdir = "./output" # base directory (default "./output")
matplotlib = 1 # 0:no, 1:yes (default 0)
/

&xs
e_l = 1.0 # lower limit of neutron energy
e_h = 1000.0 # upper limit of neutron energy
enum = 1000 # number of group (+:logscal, -:linscale)
/

&material1
elml = "data/jendl5/Ta181.res", 1.0 # path of nuclear data file
dens = 16.65 # density of sample [g/cm^3]
thickness = 0.003 # thickness of sample [cm]
teff = 300.0 # effective temperature [K] (default 300.0)
/

&fitting
meas_file = "data/measurement/Tr_Ta.dat"
# initial parameters
L = 23.7 # flight path [m]
td = 0 # delay of TOF [us]
# dens = 16.65 # density of sample [g/cm^3]
# thickness = 0.003 # thickness of sample [cm]
teff = 300.0 # effective temperature [K]
# material
fit_material = &material1
# fitting range
fit_range0 = 800.0,860.0 # lower and upper limit of fitting range (TOF [us])
# baseline fitting range
base0 = 60,250
base1 = 300,340
base2 = 475,515
base3 = 600,750
base4 = 880,1000
# mask
level = 0.0 # masking level of transmission (Default 0.0)
# adjusted parameter(s)
adjust = "base_dens" # Type of fitting parameter(s)
# path:L, delay, dens:density, temp:temperature
# pulse:pulse function, base:baseline
show = 1 # show progress of fitting; 0:no, 1:yes (Default 0)
/

&pulse
pulse_file = "data/pulse/BL10.dat" # parameters of pulse function (default none)
proton_file = "data/pulse/double.dat" # data of proton pulse shape (default none)
/

```

Figure B5 Sample input file of continuous fitting for transmission data  
(sample\_cont\_fit\_tr.inp)

```

&control
project = "sample_cont_fit_ca" # project name (default "project1")
proj_ow = 1 # overwrite project directory, No:0, Yes:1 (default 1)
fit = 1 # 0:simulation, 1:fitting
pulse_fit = 0 # 0:normal, 1:adjust/fix parameters of pulse function (default 0)
fn_xs = "xs_ta" # file name of cross section (default "xs")
fn_fiti = "initial_ta" # file name of initial fitting line (default "initial")
fn_fitf = "final_ta" # file name of final fitting line (default "final")
format = 3 # 1:csv, 2:text, 3:csv and text (default 1)
outdir = "./output" # base directory (default "./output")
matplotlib = 1 # 0:no, 1:yes (default 0)
show_baseline = 1
/

&xs
e_l = 1.0 # lower limit of neutron energy
e_h = 1000.0 # upper limit of neutron energy
enum = 1000 # number of group (:logscal, -:linscale)
/

&material1
elm1 = "data/jendl5/Cu63.res", 0.6915 # ndata, abundance
elm2 = "data/jendl5/Cu65.res", 0.3085 # ndata, abundance
dens = 8.9 # density of sample [g/cm^3]
thickness = 0.2 # thickness of sample [cm]
teff = 318.0 # effective temperature [K] (default 300.0)
/

&material2
elm1 = "data/jendl5/Ta181.res", 1.0 # path of nuclear data file
# dens = 16.65 # density of sample [g/cm^3]
# thickness = 0.003 # thickness of sample [cm]
area_dens = 1.68e-4
teff = 300.0 # effective temperature [K] (default 300.0)
/

&fitting
meas_file = "data/measurement/capture/test_ta.dat"
# initial parameters
L = 18.0 # flight path [m]
td = 1.0 # delay of TOF [us]
# dens = 16.65 # density of sample [g/cm^3]
# thickness = 0.003 # thickness of sample [cm]
teff = 300.0 # effective temperature [K]
a0 = 1 # baseline parameter a0
c = 9000 # scale factor (default 1.0)
# material
fit_material = &material2
# fitting range
fit_range0 = 600.0, 675.0 # lower and upper limit of fitting range (TOF [us])
base0 = 675, 800
base1 = 450, 600
base2 = 305, 340
base2 = 65, 80
# mask
level = 0.0 # masking level of transmission (Default 0.0)
# adjusted parameter(s)
adjust = "base_dens" # Type of fitting parameter(s)
# path:L, delay, dens:density, temp:temperature
# pulse:pulse function, base:baseline
show = 1 # show progress of fitting; 0:no, 1:yes (Default 0)
/

&pulse
pulse_file = "data/pulse/BL10.dat" # parameters of pulse function (default none)
proton_file = "data/pulse/double.dat" # data of proton pulse shape (default none)
/

```

Figure B6 Sample input file of continuous fitting for capture yield data  
(sample\_cont\_fit\_ca.inp)

```

&control
project = "sample_fit_tr_tmap" # project name (default "project1")
proj_ow = 1 # overwrite project directory, No:0, Yes:1 (default 1)
fit = 1 # 0:simulation, 1:fitting
pulse_fit = 0 # 0:normal, 1:adjust/fix parameters of pulse function (default 0)
fn_xs = "xs_ta" # file name of cross section (default "xs")
fn_fiti = "initial_ta" # file name of initial fitting line (default "initial")
fn_fitf = "final_ta" # file name of final fitting line (default "final")
format = 3 # 1:csv, 2:text, 3:csv and text (default 1)
outdir = "./output" # base directory (default "./output")
matplotlib = 1 # 0:no, 1:yes (default 0)
/

&xs
e_l = 1.0 # lower limit of neutron energy
e_h = 100.0 # upper limit of neutron energy
enum = 500 # number of group (+:logscal, -:linscale)
/

&material1
elml = "data/jendl5/Ta181.res", 1.0 # path of nuclear data file
dens = 16.65 # density of sample [g/cm3]
thickness = 0.003 # thickness of sample [cm]
teff = 300.0 # effective temperature [K] (default 300.0)
/

&fitting
meas_file = "data/measurement/Tr-Ta.dat"
# initial parameters
L = 23.7 # flight path [m]
td = 0 # delay of TOF [us]
teff = 300.0 # effective temperature [K]
# temperature list
teff_list = 270, 280, 290, 300, 310
# material
fit_material = &material1
# fitting range
fit_range0 = 800.0, 860.0 # lower and upper limit of fitting range (TOF [us])
# mask
level = 0.0 # masking level of transmission (Default 0.0)
# adjusted parameter(s)
adjust = "temp_map" # Type of fitting parameter(s)
# path:L, delay, dens:density, temp:temperature
# temp_map:temperature by map
# pulse:pulse function, base:baseline
show = 1 # show progress of fitting; 0:no, 1:yes (Default 0)
/

&pulse
pulse_file = "data/pulse/BL10.dat" # parameters of pulse function (default none)
proton_file = "data/pulse/double.dat" # data of proton pulse shape (default none)
/

```

Figure B7 Sample input file of fitting for transmission data using temperature map  
(sample\_fit\_tr\_tmap.inp)

```

&control
project = "sample_fit_tr_tmap_pw" # project name (default "project1")
proj_ow = 1 # overwrite project directory, No:0, Yes:1 (default 1)
fit = 1 # 0:simulation, 1:fitting
pulse_fit = 0 # 0:normal, 1:adjust/fix parameters of pulse function (default 0)
fn_xs = "xs_ta" # file name of cross section (default "xs")
fn_fitf = "initial_ta" # file name of initial fitting line (default "initial")
fn_fitf = "final_ta" # file name of final fitting line (default "final")
format = 3 # 1:csv, 2:text, 3:csv and text (default 1)
outdir = "./output" # base directory (default "./output")
matplotlib = 1 # 0:no, 1:yes (default 0)
/

&xs
e_l = 1.0 # lower limit of neutron energy
e_h = 100.0 # upper limit of neutron energy
enum = 500 # number of group (+:logscal, -:linscale)
/

&material1
elml = "data/jendl5/Ta181.res", 1.0 # path of nuclear data file
dens = 16.65 # density of sample [g/cm3]
thickness = 0.003 # thickness of sample [cm]
teff = 300.0 # effective temperature [K] (default 300.0)
pw1 = "data/jendl5/pointwise/Ta181_0k.res", 0 # path of pointwise file
pw2 = "data/jendl5/pointwise/Ta181_300k.res", 300 # path of pointwise file
pw3 = "data/jendl5/pointwise/Ta181_300k.res", 301 # path of pointwise file (dummy)
pw4 = "data/jendl5/pointwise/Ta181_300k.res", 302 # path of pointwise file (dummy)
/

&fitting
meas_file = "data/measurement/Tr-Ta.dat"
# initial parameters
L = 23.7 # flight path [m]
td = 0 # delay of TOF [us]
teff = 300.0 # effective temperature [K]
# material
fit_material = &material1
# fitting range
fit_range0 = 800.0,860.0 # lower and upper limit of fitting range (TOF [us])
# mask
level = 0.0 # masking level of transmission (Default 0.0)
# adjusted parameter(s)
adjust = "temp_pw" # Type of fitting parameter(s)
# path:L, delay, dens:density, temp:temperature
# temp_map:temperature by map
# temp_pw:temperature by pointwise
# pulse:pulse function, base:baseline
show = 1 # show progress of fitting; 0:no, 1:yes (Default 0)
/

&pulse
pulse_file = "data/pulse/BL10.dat" # parameters of pulse function (default none)
proton_file = "data/pulse/double.dat" # data of proton pulse shape (default none)
/

```

Figure B8 Sample input file of fitting for transmission data using temperature map of pointwise data (sample\_fit\_tr\_tmap\_pw.inp)

```

&control
project = "sample_fit_ca_tmap" # project name (default "project1")
proj_ow = 1 # overwrite project directory, No:0, Yes:1 (default 1)
fit = 1 # 0:simulation, 1:fitting
pulse_fit = 0 # 0:normal, 1:adjust/fix parameters of pulse function (default 0)
fn_xs = "xs_ta" # file name of cross section (default "xs")
fn_fiti = "initial_ta" # file name of initial fitting line (default "initial")
fn_fitf = "final_ta" # file name of final fitting line (default "final")
format = 3 # 1:csv, 2:text, 3:csv and text (default 1)
outdir = "./output" # base directory (default "./output")
matplotlib = 1 # 0:no, 1:yes (default 0)
/

&xs
e_l = 1.0 # lower limit of neutron energy
e_h = 100.0 # upper limit of neutron energy
enum = 500 # number of group (+:logscal, -:linscale)
/

&material1
elm1 = "data/jendl5/Cu63.res", 0.6915 # ndata, abundance
elm2 = "data/jendl5/Cu65.res", 0.3085 # ndata, abundance
dens = 8.9 # density of sample [g/cm^3]
thickness = 0.2 # thickness of sample [cm]
teff = 318.0 # effective temperature [K] (default 300.0)
/

&material2
elm1 = "data/jendl5/Ta181.res", 1.0 # path of nuclear data file
# dens = 16.65 # density of sample [g/cm^3]
# thickness = 0.003 # thickness of sample [cm]
area_dens = 1.68e-4
teff = 300.0 # effective temperature [K] (default 300.0)
/

&fitting
meas_file = "data/measurement/capture/test_ta.dat"
# initial parameters
L = 18.0 # flight path [m]
td = 1.0 # delay of TOF [us]
teff = 300.0 # effective temperature [K]
a0 = 1 # baseline parameter a0
c = 10000 # scale factor (default 1.0)
# temperature list
teff_list = 270, 280, 290, 300, 310
# material
fit_material = &material2
# fitting range
fit_range0 = 600.0, 675.0 # lower and upper limit of fitting range (TOF [us])
# mask
level = 0.0 # masking level of transmission (Default 0.0)
# adjusted parameter(s)
adjust = "temp_map" # Type of fitting parameter(s)
# path:L, delay, dens:density, temp:temperature
# temp_map:temperature by map
# pulse:pulse function, base:baseline
show = 1 # show progress of fitting; 0:no, 1:yes (Default 0)
/

&pulse
pulse_file = "data/pulse/BL10.dat" # parameters of pulse function (default none)
proton_file = "data/pulse/double.dat" # data of proton pulse shape (default none)
/

```

Figure B9 Sample input file of fitting for capture yield data using temperature map  
(sample\_fit\_ca\_tmap.inp)

```

&control
project = "sample_multi_fit_tr" # project name (default "project1")
proj_ow = 1 # overwrite project directory, No:0, Yes:1 (default 1)
fit = 1 # 0:simulation, 1:fitting
pulse_fit = 0 # 0:normal, 1:adjust/fix parameters of pulse function (default 0)
fn_xs = "xs_ta" # file name of cross section (default "xs")
fn_fiti = "initial_ta" # file name of initial fitting line (default "initial")
fn_fitf = "final_ta" # file name of final fitting line (default "final")
format = 3 # 1:csv, 2:text, 3:csv and text (default 1)
outdir = "./output" # base directory (default "./output")
matplotlib = 1 # 0:no, 1:yes (default 0)
/

&xs
e_l = 1.0 # lower limit of neutron energy
e_h = 100.0 # upper limit of neutron energy
enum = 500 # number of group (+:logscal, -:linscale)
/

&material1
elml = "data/jendl5/Ta181.res", 1.0 # path of nuclear data file
dens = 16.65 # density of sample [g/cm^3]
thickness = 0.003 # thickness of sample [cm]
teff = 300.0 # effective temperature [K] (default 300.0)
/

&fitting
meas_file = "data/measurement/imaging/Tr-Ta.dat"
# initial parameters
L = 23.7 # flight path [m]
td = 0 # delay of TOF [us]
# dens = 16.65 # density of sample [g/cm^3]
# thickness = 0.003 # thickness of sample [cm]
teff = 300.0 # effective temperature [K]
# material
fit_material = &material1
# fitting range
fit_range0 = 800.0,860.0 # lower and upper limit of fitting range (TOF [us])
# mask
level = 0.0 # masking level of transmission (Default 0.0)
# adjusted parameter(s)
adjust = "dens" # Type of fitting parameter(s)
# path:L, delay, dens:density, temp:temperature
# pulse:pulse function, base:baseline
show = 1 # show progress of fitting; 0:no, 1:yes (Default 0)
/

&pulse
pulse_file = "data/pulse/BL10.dat" # parameters of pulse function (default none)
proton_file = "data/pulse/double.dat" # data of proton pulse shape (default none)
/

```

Figure B10 Sample input file of fitting for multiple transmission data  
(sample\_multi\_fit\_tr.inp)

```

&control
project = "sample_multi_fit_tr_tmap" # project name (default "project1")
proj_ow = 1 # overwrite project directory, No:0, Yes:1 (default 1)
fit = 1 # 0:simulation, 1:fitting
pulse_fit = 0 # 0:normal, 1:adjust/fix parameters of pulse function (default 0)
fn_xs = "xs_ta" # file name of cross section (default "xs")
fn_fiti = "initial_ta" # file name of initial fitting line (default "initial")
fn_fitf = "final_ta" # file name of final fitting line (default "final")
format = 3 # 1:csv, 2:text, 3:csv and text (default 1)
outdir = "./output" # base directory (default "./output")
matplotlib = 1 # 0:no, 1:yes (default 0)
/

&xs
e_l = 1.0 # lower limit of neutron energy
e_h = 100.0 # upper limit of neutron energy
enum = 500 # number of group (+:logscal, -:linscale)
/

&material1
elml = "data/jendl5/Ta181.res", 1.0 # path of nuclear data file
dens = 16.65 # density of sample [g/cm3]
thickness = 0.003 # thickness of sample [cm]
teff = 300.0 # effective temperature [K] (default 300.0)
/

&fitting
meas_file = "data/measurement/imaging/Tr-Ta001.dat"
# initial parameters
L = 23.7 # flight path [m]
td = 0 # delay of TOF [us]
teff = 300.0 # effective temperature [K]
# temperature list
teff_list = 270, 280, 290, 300, 310
# material
fit_material = &material1
# fitting range
fit_range0 = 800.0, 860.0 # lower and upper limit of fitting range (TOF [us])
# mask
level = 0.0 # masking level of transmission (Default 0.0)
# adjusted parameter(s)
adjust = "temp_map" # Type of fitting parameter(s)
# path:L, delay, dens:density, temp:temperature
# temp_map:temperature by map
# pulse:pulse function, base:baseline
show = 1 # show progress of fitting; 0:no, 1:yes (Default 0)
/

&pulse
pulse_file = "data/pulse/BL10.dat" # parameters of pulse function (default none)
proton_file = "data/pulse/double.dat" # data of proton pulse shape (default none)
/

```

Figure B11 Sample input file of fitting for multiple transmission data using temperature map (sample\_multi\_fit\_tr\_tmap.inp)

```

&control
project = "sample_multi_fit_ca" # project name (default "project1")
proj_ow = 1 # overwrite project directory, No:0, Yes:1 (default 1)
fit = 1 # 0:simulation, 1:fitting
pulse_fit = 0 # 0:normal, 1:adjust/fix parameters of pulse function (default 0)
fn_xs = "xs_ta" # file name of cross section (default "xs")
fn_fiti = "initial_ta" # file name of initial fitting line (default "initial")
fn_fitf = "final_ta" # file name of final fitting line (default "final")
format = 3 # 1:csv, 2:text, 3:csv and text (default 1)
outdir = "./output" # base directory (default "./output")
matplotlib = 1 # 0:no, 1:yes (default 0)
/

&xs
e_l = 1.0 # lower limit of neutron energy
e_h = 100.0 # upper limit of neutron energy
enum = 500 # number of group (+:logscal, -:linscale)
/

&material1
elm1 = "data/jendl5/Cu63.res", 0.6915 # ndata, abundance
elm2 = "data/jendl5/Cu65.res", 0.3085 # ndata, abundance
dens = 8.9 # density of sample [g/cm^3]
thickness = 0.2 # thickness of sample [cm]
teff = 318.0 # effective temperature [K] (default 300.0)
/

&material2
elm1 = "data/jendl5/Ta181.res", 1.0 # path of nuclear data file
# dens = 16.65 # density of sample [g/cm^3]
# thickness = 0.003 # thickness of sample [cm]
area_dens = 1.68e-4
teff = 300.0 # effective temperature [K] (default 300.0)
/

&fitting
meas_file = "data/measurement/capture/imaging/Ca-Ta001.dat"
# initial parameters
L = 18.0 # flight path [m]
td = 1.0 # delay of TOF [us]
# dens = 16.65 # density of sample [g/cm^3]
# thickness = 0.003 # thickness of sample [cm]
teff = 300.0 # effective temperature [K]
a0 = 1 # baseline parameter a0
c = 10000 # scale factor (default 1.0)
# material
fit_material = &material2
# fitting range
fit_range0 = 600.0,675.0 # lower and upper limit of fitting range (TOF [us])
# mask
level = 0.0 # masking level of transmission (Default 0.0)
# adjusted parameter(s)
adjust = "dens" # Type of fitting parameter(s)
# path:L, delay, dens:density, temp:temperature
# pulse:pulse function, base:baseline
show = 1 # show progress of fitting; 0:no, 1:yes (Default 0)
/

&pulse
pulse_file = "data/pulse/BL10.dat" # parameters of pulse function (default none)
proton_file = "data/pulse/double.dat" # data of proton pulse shape (default none)
/

```

Figure B12 Sample input file of fitting for multiple capture yield data  
(sample\_multi\_fit\_ca.inp)



```

&control
project = "sample_multi_fit_ca_tmap" # project name (default "project1")
proj_ow = 1 # overwrite project directory, No:0, Yes:1 (default 1)
fit = 1 # 0:simulation, 1:fitting
pulse_fit = 0 # 0:normal, 1:adjust/fix parameters of pulse function (default 0)
fn_xs = "xs_ta" # file name of cross section (default "xs")
fn_fiti = "initial_ta" # file name of initial fitting line (default "initial")
fn_fitf = "final_ta" # file name of final fitting line (default "final")
format = 3 # 1:csv, 2:text, 3:csv and text (default 1)
outdir = "./output" # base directory (default "./output")
matplotlib = 1 # 0:no, 1:yes (default 0)
/

&xs
e_l = 1.0 # lower limit of neutron energy
e_h = 100.0 # upper limit of neutron energy
enum = 500 # number of group (+:logscal, -:linscale)
/

&material1
elm1 = "data/jendl5/Cu63.res", 0.6915 # ndata, abundance
elm2 = "data/jendl5/Cu65.res", 0.3085 # ndata, abundance
dens = 8.9 # density of sample [g/cm^3]
thickness = 0.2 # thickness of sample [cm]
teff = 318.0 # effective temperature [K] (default 300.0)
/

&material2
elm1 = "data/jendl5/Ta181.res", 1.0 # path of nuclear data file
# dens = 16.65 # density of sample [g/cm^3]
# thickness = 0.003 # thickness of sample [cm]
area_dens = 1.68e-4
teff = 300.0 # effective temperature [K] (default 300.0)
/

&fitting
meas_file = "data/measurement/capture/imaging/Ca-Ta001.dat"
# initial parameters
L = 18.0 # flight path [m]
td = 1.0 # delay of TOF [us]
teff = 300.0 # effective temperature [K]
a0 = 1 # baseline parameter a0
c = 10000 # scale factor (default 1.0)
# temperature list
teff_list = 270, 280, 290, 300, 310
# material
fit_material = &material2
# fitting range
fit_range0 = 600.0,675.0 # lower and upper limit of fitting range (TOF [us])
# mask
level = 0.0 # masking level of transmission (Default 0.0)
# adjusted parameter(s)
adjust = "temp_map" # Type of fitting parameter(s)
# path:L, delay, dens:density, temp:temperature
# temp_map:temperature by map
# pulse:pulse function, base:baseline
show = 1 # show progress of fitting; 0:no, 1:yes (Default 0)
/

&pulse
pulse_file = "data/pulse/BL10.dat" # parameters of pulse function (default none)
proton_file = "data/pulse/double.dat" # data of proton pulse shape (default none)
/

```

Figure B13 Sample input file of fitting for multiple capture yield data using temperature map (sample\_multi\_fit\_ca\_tmap.inp)

This is a blank page.



