

汎用炉物理解析コードシステムCBZのチュートリアル

千葉豪

1 はじめに

原子核に中性子が衝突すると、ある確率のもとで核分裂反応が起こり、原子核は通常2つに分裂します。その分裂した原子核（核分裂片）は莫大な運動エネルギー（約 167 MeV）を持っており、それを最終的に電気エネルギーに変換して利用するのが原子力発電です¹。

核分裂反応を起こす核燃料は原子炉容器に格納されます。原子炉容器内には種々の機器が配置されていますが、その中の核燃料が設置される周辺は**炉心**（Reactor core）と呼ばれ、文字通り原子力発電所の心臓部にあたります。

原子力発電では、核分裂反応を持続的に行わせるため、反応を生じさせる中性子の数を制御しています。すなわち、核分裂反応を適切に制御するためには、炉心内での中性子の振る舞いを知る必要があります。炉心内での中性子の振る舞いは Boltzman の輸送方程式と呼ばれる式に従いますので、この方程式の解を得なければなりません。この方程式を解いて中性子の振る舞いを知ることは**炉心解析**、**炉心の核特性解析**²と呼ばれ、原子力工学の重要な分野のひとつとなっています。

炉心解析を行うための種々のコードシステム（プログラム）が世界各国で開発されています。最も有名なものとしては STUDSVIK 社の CASMO/SIMULATE が挙げられるでしょう。このコードシステムは世界各国において実機軽水炉の炉心解析に広く用いられています。また、日本でも様々なものが開発されており、例えば原子燃料工業、原子力エンジニアリングが名古屋大学と共同で開発している AEGIS/SCOPE-2 や、三菱重工業が開発している GALAXY/COSMO-S などが挙げられます。

著者は JAEA 在籍時に、炉物理解析を学ぶために、独自にコードシステム CBG を開発し、それを用いていくつか論文を書きました。この CBG の開発経験を基に、北大に移ってから開発を始めたコードシステムが CBZ です（このチュートリアルでは「CBZ」と「CBG」が混在していますが、どちらも同じようなものなので、あまり気にしないで下さい）。CBZ では、臨界計算や動特性計算、放射線の遮蔽計算、核燃料の燃焼計算など、多様な炉物理解析を行うことが出来ます。本チュートリアルは CBZ の使い方を学んでもらうためのものです。

¹厳密には、核分裂片の運動エネルギーに加えて、核分裂反応とほぼ同時に発生する即発ガンマ線や、核分裂片が脱励起した後の核分裂生成物がベータ崩壊した際に発生するベータ線、遅発ガンマ線のエネルギーも利用されます。

²**核特性**というのは、中性子と原子核との反応に関連した原子炉の特性のことを示しており、例えば臨界量や出力分布、反応度係数などが挙げられます。

2 インストール

CBZ では、ユーザは CBZ の種々のモジュールを利用して自身の目的に沿った計算を行います。具体的には、ユーザは C++ を用いてプログラムを記述し、それをコンパイルすることで計算を行います。このコンパイルの際には、CBZ の種々のモジュールをコンパイルすることにより作成されたライブラリファイルである「libCBG.a」を**インクルード**する必要があります。この関係を Fig. 1 に示します。CBZ のソースプログラムが図中左側の「CBZ source」に対応し、ユーザーが作成するプログラムが右側の「Main programs for users」になります。ここでは、CBZ のソースプログラムのコンパイルを行います。ユーザが自身の目的に応じて記述する C++ プログラムのコンパイルについては、これ以降の章で行うことになります。

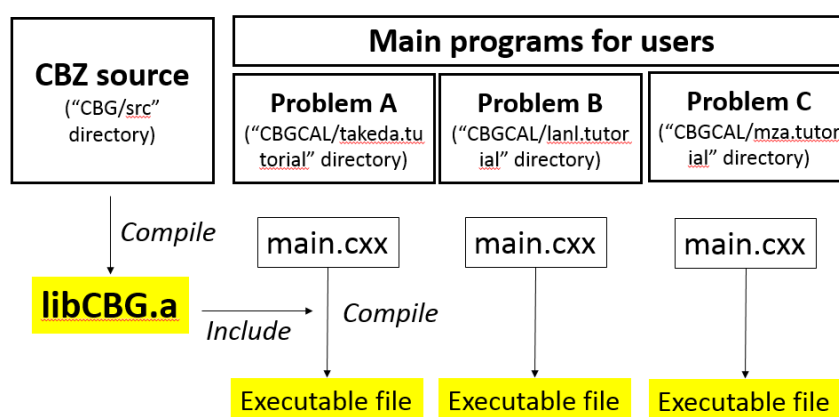


Fig. 1: CBZ を用いた計算の構造

パッケージ `cbz.xxxxxyyzz.tgz` を解凍して展開する（xxxxyyzz は日付を示す）³ と、ディレクトリ CBZ が生成され、その下に以下のディレクトリが作成されます⁴。なお、GitHub から DL する場合には、CBG の下の `src` ディレクトリ中のファイル入手することになりますので、以下の記述の必要な部分を参考にして下さい。

- CBG
- CBG/CAL
- CBGLIB
- CBGLIB_BURN

CBG ディレクトリの下には `src` ディレクトリがあるはずですが（`dist` ディレクトリもありますが、これについては後述します）。この `src` ディレクトリには、CBZ のソースプログラムが置かれています。一方、CBG/CAL ディレクトリは CBZ を用いていろいろな計算を行うためのデータを置くためのもので、配布パッケージには `Takeda.tutorial`、`lanl.tutorial`、`mza.tutorial` 等のディレクトリがあるかと思います。また、CBGLIB ディレクトリは CBZ のための断面積（核データ⁵）ライブラリ CBZLIB を置くためのもの、CBGLIB_BURN ディレクトリは燃焼計算を行う際に必要となるデータファイルを置くためのものです。

³ 「`tar xvfz cbz.xxxxxyyzz.tgz`」と打ち込む（鍵括弧は不要です）。

⁴ CBZ ディレクトリの下には階層構造でディレクトリが構成され、さまざまなファイルが置かれています。CBZ ディレクトリに含まれる全てのデータをどこかにもっていくためには、これらをひとつのファイルにまとめるのが便利です。ディレクトリ構造をひとつのファイルにまとめることを「tar する」といい、まとめられたファイルは **tar ファイル**と呼ばれます。例えば、CBZ ディレクトリを tar ファイルにまとめるためには、「`tar cvf cbz.tar CBZ`」と打ち込めば、CBZ ディレクトリが `cbz.tar` というひとつのファイルに変換されます。また、tar されたファイルは一般にサイズが大きくなるため、圧縮してあげます。そのコマンドが「`gzip cbz.tar`」であり、これを行うことにより、`cbz.tar.gz` ファイルが作成されます。この、tar して圧縮する一連の動作を一度に行うコマンドが「`tar cvfz cbz.tgz CBZ`」であり、その逆が「`tar xvfz cbz.tgz`」となります。

⁵ 核データとは、中性子と原子核との反応を記述するためのデータです。詳細は後ほど述べます。

それではCBZをコンパイルしましょう。まずはCBG/srcディレクトリに移動して下さい⁶。配布されたパッケージでは既にコンパイルされている可能性がありますので、まずは「make clean」と打ち込んで、既にコンパイルされている場合に作成されているファイルを消去します。その後、コンパイルを行うため、「make」と打ち込んでください。コンパイルが始まり、数分後くらいにlibCBG.aが生成されます。「ls *.a」と打ち込んで、このファイルが生成されていることを確認して下さい。

なお、gfortran がコンパイルされていない環境ではエラーで落ちますので、この場合は、gfortran を自らの計算機環境にインストールして下さい（インストールの方法が分からない場合には周りの学生や教員に聞いて下さい）。もしくは、「cp Makefile_wo_fortran Makefile」と打ち込んで、gfortran が不要な条件でMakeを行って下さい。

また、gfortran とは別な要因でコンパイルに失敗する場合があります。この場合には、src ディレクトリ内の Numeric.h の12行目（以下参照）のコメントアウトを外したうえで、再びコンパイルしてみてください。それでもコンパイルできない場合には、千葉に相談して下さい。

Listing 1: Numeric.h の冒頭部分の抜粋

```
1 #ifndef PIDEF
2 #define PIDEF
3
4 #include <string>
5 #include <vector>
6 #include <math.h>
7
8 typedef double real;
9 typedef float real4;
10 typedef double real8;
11
12 // #define _OLD_COMPILER_ENV_
13
14 const real PI = 3.1415926535897932384626433;
```

以上でインストールは終了です。

なお、CBZのソース（srcディレクトリ内のプログラムの一部）を新しいものに置換（上書き）した場合や、ある別なPCでインストールが完了したCBZディレクトリをそれとは異なるPCにコピーして利用する場合などには、改めてここで述べられた手順でコンパイルを行い、libCBG.aを作成する必要があります。特に、「新しいソースを使って欲しい」という連絡があった場合には、新しいソースのtgzファイルが提供されたり、GitHubから更新されたファイルを入手したりすることになります。前者の場合には、このファイルを解凍するとsrcディレクトリが生成されますので、この作業を行う前に、まずはそれまで使っていたsrcディレクトリの名前を変更し（例えばsrc.oldなど）、その後、CBGディレクトリで新しいソースのtgzファイルを解凍し、再コンパイルを行うとよいでしょう。この最後で述べた部分は、今の段階では全く関係ありませんが、今後、CBZを使っていく際には非常に重要な事項となりますので、必ず覚えておいて下さい。

⁶ 「cd CBZ/CBG/src」と打ち込む。

3 Takeda ベンチマークの計算

それでは CBZ を用いて、中性子輸送⁷ のベンチマーク問題として世界的に広く用いられている Takeda ベンチマークの計算をしてみましょう⁸。Takeda ベンチマーク問題には4つの体系が用意されていますが、ここでは最も簡単な体系1を計算します。体系1の概略図を Fig. 2 に示します。

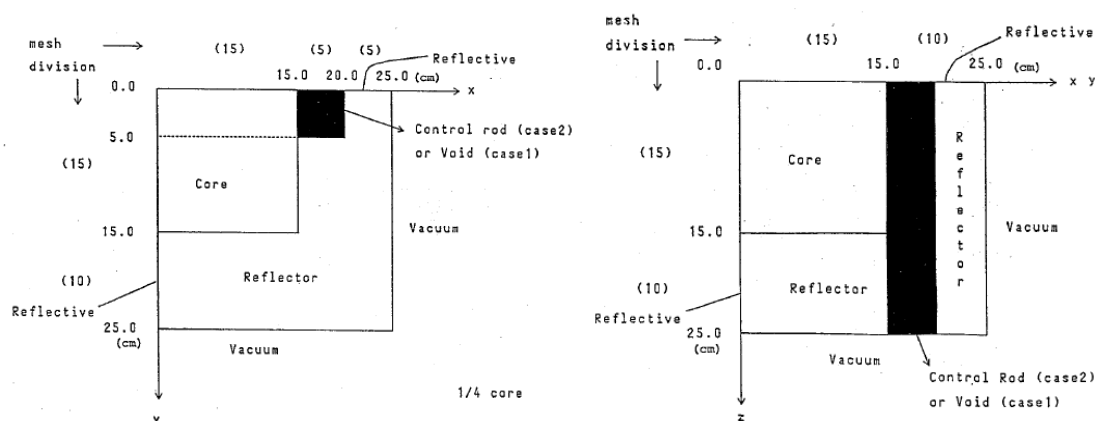


Fig. 2: Takeda ベンチマーク問題の体系1 (左が XY 平面、右が XZ (YZ) 平面)

Fig. 2 の左が上から見た図になります。全体の 1/4 が示されており、中央部に「炉心領域 (Core)」、外周部に「反射体領域 (Reflector)」が配置されていることが分かります。また、反射体領域の一部が黒塗りされていますが、この部分は制御棒が挿入される位置で、制御棒が挿入されていない状態では「制御棒フォロワ⁹」(実質的には空気)となります。

右は横から見た図になっており、制御棒(もしくはフォロワ)領域が縦に延びていることが分かります。また、軸方向についても炉心領域が反射体領域に囲まれていることが分かります。

それでは、CBGCAL/Takeda.tutorial ディレクトリの main.takeda1.dif.1.cxx をエディタで開いて下さい。

まず、Geometry data と書かれたコメント行が(20 行目くらいに)あると思います。ここから下の部分では、計算する体系の形状データを定義しています。今回計算するのは正方格子状の体系で、X、Y、Z 方向をそれぞれ5領域(ブロック)に区切っています。各々のブロックの各辺の長さは5cm となっており(配列 x1、y1、z1 で指定)、計算ではそれら各辺をさらに5分割しています(配列 xm、ym、zm で指定)。各辺が25に分割されることになるため、この体系の全計算メッシュ数は $25 \times 25 \times 25 = 15,625$ となります。また、 $5 \times 5 \times 5 = 125$ ブロックのそれぞれについて、どの物質(マテリアル)が割り当てられるかを配列 mat で定義します。ここでは、0 が燃料、1 が反射体、2 が制御棒、3 がフォロワとなります。なお Takeda ベンチマークの体系1では、制御棒挿入状態と制御棒引き抜き状態のふたつの状態について評価することになっています。この入力ファイルでは、制御棒挿入位置に物質3が割り振られているため、制御棒引き抜き状態に対応します。

体系の物質配置の定義の仕方がいまいちピンとこない場合には、ホームページの「CBZ」の項に「CBZ の XYZ 座標系の炉心体系情報に関するクラス」という文書があるので、その6.1 節「媒質マップと物質配置、境界条件の関係」を参考にして下さい。

次に、Material data と書かれたコメント行を(70 行目くらいに)見つけれられることと思います。ここから下の部分では、物質のデータ、具体的には中性子と原子核の反応断面積を定義しています。なお、この

⁷「輸送」というと、何か物を運ぶようなことが連想されますが、媒質内の中性子の移動を英語では **Neutron transport** と呼ぶため、これを和訳して**中性子輸送**となります。いずれ慣れます。

⁸「Takeda ベンチマーク」とは、元大阪大学教授(現福井大学教授)の竹田敏一先生が中性子輸送方程式を数値的に解くプログラムの妥当性を検証するために作成したベンチマーク問題です。1991 年に経済協力開発機構・原子力機関(OECD/NEA)が世界各国の研究者を召集し、このベンチマーク問題を用いて、参加者それぞれが開発、使用している計算コードの比較が行われました。

⁹制御棒の「通り道」ということで、このような名前となっています。

問題ではエネルギーを2つに分けています。ちなみに、分割されたエネルギー領域を**群** (group) と呼び、この例では「エネルギー群数は2」という言い方となります。1 群が高速中性子エネルギーに、2 群が熱中性子エネルギーに、それぞれ対応します。

最後に、それら定義した形状データ、物質データを用いて、中性子拡散方程式を解きます。なお、中性子の振る舞いは中性子輸送方程式によって記述されますが、輸送方程式に近似を導入することで拡散方程式が得られます。輸送方程式においては、中性子束は位置、エネルギー、運動方向に依存する（角度中性子束を扱う）一方、拡散方程式では運動方向の依存性を近似的に扱う（スカラー中性子束のみを扱う）ため、方程式に中性子束の運動方向依存性が現れず、計算負荷が小さくなります。ここでは、中性子拡散方程式を数値的に解くコード PLOS を呼び出して、処理を行わせています。

実際の計算は以下の通りに行います。まず「`cp main.takeda1.dif_1.cxx main.cxx`」と打ち込んで、`main.takeda1.dif_1.cxx` ファイルを `main.cxx` にコピーして下さい。その後、「`make`」と打ち込んでコンパイルして下さい¹⁰。その結果、`a.out` という名前の実行形式（ロードモジュール）ファイルが生成されますので、確認して下さい。このファイルを確認できたら「`./a.out`」と打ち込んで、計算を実行して下さい。以下のように処理結果がずらずらと出力されることと思います。

Listing 2: Takeda ベンチマークの計算例

```

1  *** Total Mesh : 15625
2  ****
3  ** System CBG
4  ** Power iteration
5  ** Solver : PLOS
6  ** Acceleration : none
7  ** Forward calculation
8  ** Convergence condition (k_eff) : 1e-05
9  **                        (flux) : 0.0001
10 **                        (source): 0.0001
11 ****
12 ** It:   K_eff   : Err in k : Err in   : Err in
13 ** er:      :      : flux      : source
14 ****
15 #  0:   0.906483  9.352e-02  2.233e+00  3.700e-01
16 #  1:   0.917541  1.220e-02  1.018e-01  7.723e-02
17 #  2:   0.925198  8.344e-03  6.630e-02  5.346e-02
18 #  3:   0.927640  2.640e-03  1.552e-02  1.376e-02
19 #  4:   0.927664  2.565e-05  4.699e-02  2.787e-02
20 #  5:   0.928331  7.198e-04  4.428e-03  2.952e-03
21 #  6:   0.927417  9.849e-04  1.424e-03  2.563e-02
22 #  7:   0.927214  2.192e-04  4.204e-03  1.606e-02
23 #  8:   0.927261  5.146e-05  2.991e-03  7.066e-03
24 #  9:   0.927328  7.128e-05  3.155e-03  2.690e-03
25 # 10:   0.927390  6.744e-05  6.710e-04  4.409e-04
26 # 11:   0.927305  9.221e-05  2.339e-04  2.580e-03
27 # 12:   0.927291  1.492e-05  4.324e-04  1.440e-03
28 # 13:   0.927291  6.520e-07  8.558e-04  7.044e-04
29 # 14:   0.927304  1.414e-05  1.460e-04  1.022e-04
30 # 15:   0.927295  9.924e-06  4.249e-04  3.309e-04
31 # 16:   0.927300  5.689e-06  5.646e-05  3.009e-05

```

PLOS では拡散方程式を反復法によって解きますので、反復毎に中性子実効増倍率 k_{eff} や中性子束（の空間、エネルギー）分布（の推定値）が更新されます。最終的に、実効増倍率や中性子束分布が一定値に収束したことを確認して、これらを計算結果とします。具体的には、今の反復回での結果と一つ前の反復回での結果の差異を評価して、それがある値よりも小さくなった場合に、収束と判断します。上の例に示されている情報はこの反復計算の過程であり、反復回数、その反復回で得られた実効増倍率、そして、実効増倍率、中性子束分布、核分裂中性子源分布についての前の反復回との相対差異（中性子束分布、中性子源分布についてはその最大値）が表示されています。最終的な実効増倍率の計算結果は最後の行から判断することができます。

数値計算には様々な誤差が含まれますが、代表的なものとして「空間メッシュの離散化誤差」が挙げられ

¹⁰ `make` により、同一のディレクトリにある `main.cxx` ファイルのコンパイルを行います（処理の詳細は `Makefile` ファイルで定義されています）。従って、計算したいファイルがある場合、本文中で述べられているように一旦それを `main.cxx` にコピーして `make` することになります。

ます。今回の計算では一辺 5cm のブロックを $5 \times 5 \times 5$ メッシュに分割していますが、この計算で得られるのはあくまで近似解であり、メッシュ数を無限大相当にしたときに、はじめて空間メッシュの離散化誤差がゼロの解を得ることが出来ます。

課題 3.1: ブロック各辺のメッシュ数を 2 倍、3 倍にした計算を行い、得られた実効増倍率と計算に要した時間を比較して下さい（計算時間の測定は、計算を実行するときに、「time ./a.out」というように time を挿入してやればよいです）。メッシュ数の変更は main.cxx ファイル中の配列 xm、ym、zm の値を変更することで行えます。

さて、このベンチマーク問題には、厳密数値解として**モンテカルロ法**と呼ばれる方法によって中性子輸送方程式を解いて得られた値が与えられており、制御棒が引き抜かれた体系では 0.9780 とされています。PLOS で得られた結果はどうだったのでしょうか？かなりずれていたのではないかと思います。既に述べましたが、PLOS はあくまで中性子輸送方程式の近似式である拡散方程式の解を計算します。従って、その解は近似解に過ぎないと言えます。

それでは次に中性子輸送方程式を解くプログラムを使ってみましょう。main.takeda1.sn.cxx を実行して下さい（ここでは、5cm のブロックを $10 \times 10 \times 10$ のメッシュに分割しています）。計算結果はどうなったのでしょうか？

ここで用いたプログラムは、SNT という、中性子輸送方程式を解くためのものです。SNT を使った場合、厳密数値解（すなわちモンテカルロ法による計算値 0.9780）にかなり近い値が得られた一方、計算時間が長くなってしまったのではないかと思います。数値計算の分野では、収束に至るまでの時間を可能な限り短縮するための手法の開発がひとつの大きなテーマとなっています。原子炉解析の分野でも同様であり、種々の方法が提案されています。ここでは例として、**拡散合成加速法**（Diffusion synthetic acceleration）を使ってみましょう。これは、輸送方程式の反復収束解の推定に、近似式である拡散方程式の解を用いる、という方法です。main.cxx（もともとは main.takeda1.sn.cxx）に sys.NoDSAAcceleration() なる行が（下から 5 行目くらいに）あるでしょうから、これを削除、もしくはコメントアウト（この行の頭に「//」を追加）して下さい。これをコンパイルして実行すると、だいぶ短い時間で収束解が得られることが分かるでしょう。

拡散合成法に加えて**粗メッシュ有限差分加速法**（Coarse-mesh finite difference acceleration）という方法を用いると、さらに計算を加速することが出来ます。この方法は、最終行近くにある CalIgen('none'); を CalIgen('cmfd'); に変更する（'none' を 'cmfd' に変更する）ことで用いることが出来ます。計算時間がさらに短縮することを確認して下さい。

課題 3.2: 今までには制御棒引き抜き状態について計算していましたが、次は制御棒挿入状態について計算してみましょう。これは物質配置のデータを変更することで容易に実施可能です。拡散計算、輸送計算でそれぞれ制御棒挿入前後の実効増倍率から制御棒反応度を求めて下さい。なお、制御棒挿入前後の実効増倍率を k_1 、 k_2 とした場合、反応度は $(1/k_1 - 1/k_2) = (k_2 - k_1)/(k_1 k_2)$ と計算されます（単位は $[\Delta k/k']$ ）。ちなみにモンテカルロ法では制御棒反応度は $-0.0166 \Delta k/k'$ と得られていますので、この参照値と比較して下さい。

ここまでは実効増倍率の観察を行ってきましたが、次は中性子束の空間分布を見てみましょう。

X 軸方向に沿った中性子束分布を出力するメソッド ShowNeutronFluxAlongXAxis がありますので、それを使います。main.takeda1.dif_1.cxx のファイルの最後の方にこのメソッドがコメントアウトしてあるはずなので、実行させるときはコメントアウトを外して下さい。なお、このメソッドの引数ですが、一つ目、二つ目がそれぞれ中性子束分布を出力させる Y、Z 方向の位置を示します。これがふたつともゼロだと、Y、Z 方向それぞれについて一つ目のメッシュ位置の X 軸方向の中性子束分布を出力することになります。また、三つ目の引数は出力させる中性子束分布のエネルギー群を示します（なお、この数値を -1 にす

ると全てのエネルギー群が出力されます)。ここで、メッシュ位置、エネルギー群ともに 0 から値が始まることに注意して下さい。

課題 3.3：制御棒引き抜き状態について、制御棒挿入位置を横断するように、X 軸に沿った中性子束分布を 1 群、2 群ともに出力させて、Gnuplot や Excel 等で図示して下さい。また、制御棒を挿入した状態でも同様の図を作成し、制御棒挿入前後で中性子束の空間分布がどのように変化するか、観察して下さい。

なお、中性子束の単位は $[\text{cm}^2/\text{s}]$ ですが、PLOS 等の計算では全核分裂中性子源が 1.0 となるように中性子束を規格化しているため、中性子束の絶対値自体は意味を持ちません。また、1 群の中性子束は「高速中性子」、2 群の中性子束は「熱中性子」にそれぞれ対応します。

計算結果は画面に出力されるため、これを編集するのは少々面倒臭いです。そこで、「./a.out > output」のようにしてやると、画面への出力がファイル output に書き込まれていくので、あとは output を編集すればよいことになります。このような方法をとるのがよいでしょう。

4 コマンド cbz の活用

ここで述べる内容は Mac には対応していないことが分かっていますので、Mac ユーザーは読み飛ばして下さい。

これまでに解説したように、CBZ で計算を行う際には、ユーザが準備したファイルを `main.cxx` にコピーし、`make` を行ったのち、`./a.out` によって実行するという流れとなります。このような方式は、慣れてくると面倒臭く感じるでしょう。また、入力ファイルを何度も修正、変更するような場合には、オリジナルの入力ファイル `mainXXXXXX.cxx` を変更しているのか、`main.cxx` を変更しているのか、作業中に混乱してしまうような場合もあり、作業の効率性を著しく損ねてしまう可能性があります。

このような問題を解決するため、2020 年 9 月に修士を卒業した Qin Tian 君が、当研究室に在籍中にコマンド `cbz` を開発しました。このコマンドは、例えば計算を行いたい入力ファイルとして `mainXXXXX.cxx` があった場合に、

```
cbz mainXXXXX.cxx
```

と入力することにより、上記の一連の動作を自動的に行ってくれるという優れたものです。

また、画面に表示される出力を `output` というファイルに出力したい場合は、

```
cbz mainXXXXX.cxx output
```

と入力すればよいです。

このコマンドを使うためにはいくつかの簡単な設定を行う必要がありますが、それについては CBZ/CBG ディレクトリの `ReadMe.txt` ファイルを参照して下さい。それほど難しくはありません。

このコマンドは活用してみると、かなり有効であることに気付くと思いますので、面倒臭がらずにこの設定を行うことを強く推奨します。

5 京都大学臨界集合体 KUCA の軽水減速架台の計算

核分裂エネルギーから電気を作る原子炉を**実機原子炉**、**出力炉**と呼んだりしますが、一方で研究用の**臨界集合体**というものがあります。これは原子炉内の中性子の振る舞いを実験的に検証するための装置であり、核分裂反応は持続しますが（臨界）、熱出力は極めて低く抑えられます（つまり中性子束レベルが極めて低いということ）。臨界集合体は日本にもいくつかあり、例えば、京都大学の臨界集合体 KUCA（Kyoto University Critical Assembly）が挙げられます。

また、原子炉は中性子のエネルギー分布から分類することも可能であり、大部分が高速中性子で構成される原子炉を**高速中性子炉**と呼び、中性子の減速材（例えば水やポリエチレン、炭素）を用いることにより熱中性子の寄与を支配的にする原子炉を**熱中性子炉**と呼びます。

ここでは、京都大学臨界集合体 KUCA の軽水減速架台の計算を行います。軽水減速架台とは、軽水で満たされたプール中に、アルミニウムで被覆された薄い金属ウラン板が複数配置されるような体系です。軽水がふんだんに使われますから、熱中性子炉となります。

今回計算する対象である KUCA の体系図を **Fig. 3** に示します。Z 軸方向の長さは固定されています。今回の計算では 57 cm として下さい。また、Y 軸方向の長さですが、今回の計算では 30 cm として下さい。燃料は X 軸方向に追加されていくので、X 軸方向の長さが可変となります。

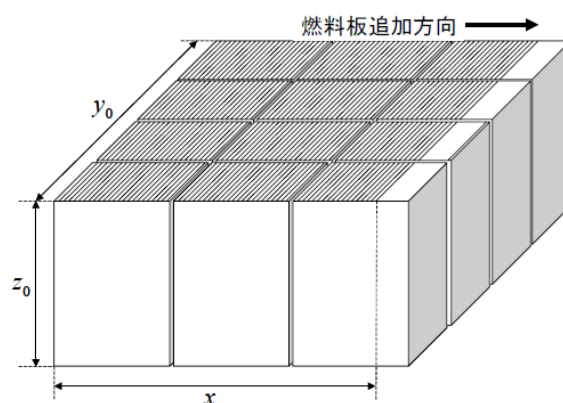


Fig. 3: KUCA の体系図

燃料領域では薄い燃料板が何枚も並べられますが、板と板との間には一定の間隙があり、そこには軽水が満たされることになります。ある燃料板の中心位置と隣の燃料板の中心位置の間の距離を**燃料配列ピッチ**と呼びます。燃料板の厚さは一定ですので、燃料配列ピッチの値が大きくなると、間隙に入り込む軽水の割合が増加することになります。KUCA では、このピッチの長さがおおよそ 3.0 mm の燃料を「C30」、3.5 mm のものを「C35」、4.5 mm のものを「C45」と、それぞれ呼称します。

燃料領域は、燃料板と間隙内の軽水、さらにはその他の構造物で構成されます。これらの構造を正確に模擬して数値計算するのは手間を要するので、ここでは「等価な均質の媒質」に置換したデータを用います。その値を **Table 1** に示します。表から明らかなように、エネルギー群は 2 群として扱うこととなります。また、この表には燃料領域を取り囲む軽水領域のデータも与えられています。

課題 5.1：C30、C35、C45 の 2 群断面積データを観察して、各断面積の燃料間の違いと、その違いが生じる理由について説明して下さい。

それでは実際に計算を始めましょう。CBGICAL/KUCA ディレクトリに移動して下さい。そこに `main.keff_1.cxx` なるファイルがあると思いますので、それをエディタで開いて下さい。

この入力には、C30、C35、C45、水領域の断面積データが既に与えられています¹¹。入力下部に体系を

¹¹なお、燃料領域の 2 群核分裂生成断面積 ($\nu\Sigma_{f,2}$) に、高速核分裂因子 ϵ を乗ずることによって、高速中性子による核分裂反応

Table 1: KUCA の 2 群断面積データ

		第1群(高速群)			第2群(熱群)			ε	δ (cm)
		D_1 (cm)	Σ_{1a} (cm ⁻¹)	$\Sigma_{1\rightarrow 2}$ (cm ⁻¹)	D_2 (cm)	Σ_{2a} (cm ⁻¹)	$\nu\Sigma_{2f}$ (cm ⁻¹)*		
炉 心 部	C30	1.58	0.00320	0.0178	0.271	0.0930	0.168	1.121	9.4
	C35	1.54	0.00286	0.0212	0.237	0.0850	0.149	1.092	8.2
	C45	1.50	0.00237	0.0254	0.203	0.0724	0.121	1.064	7.8
軽水 反射体		1.41	0.0	0.0476	0.117	0.0191	0.0	---	---

* $\nu=2.44$

構築するところがありますので、皆さんは適宜そこを変更して計算して下さい。なお、計算メッシュのサイズに留意する必要があるので、体系のサイズを変更する場合は、計算メッシュも適切に設定して下さい（1メッシュあたり 1cm 等、目安を決めておくことが必要です）。

課題 5.2：燃料領域を取り囲む水領域を無視した場合、燃料をどの程度装荷すれば（どの程度の X[cm] とすれば）臨界になるか、燃料として C30、C35、C45 を用いた場合それぞれについて計算して下さい。なお、計算メッシュは 1cm あたり 1 メッシュとすればよいでしょう。

以降の問題では水反射体を燃料の外側に設置しますので、main.keff_2.cxx を利用して下さい。

課題 5.3：燃料領域を厚さ 30cm の水領域が取り囲むとした場合、燃料をどの程度装荷すれば（どの程度の X[cm] とすれば）臨界になるか、燃料として C30、C35、C45 を用いた場合それぞれについて計算して下さい。

課題 5.4：中性子実効増倍率の観点から燃料領域を取り囲む水領域の厚さが無限相当と見做せる最小値を、燃料として C30、C35、C45 を用いた場合それぞれについて計算して下さい。

課題 5.5：課題 5.3 で計算した臨界体系において、炉中心位置（YZ 平面でのほぼ中心位置）における X 軸方向の 1 群、2 群の中性子束分布を計算し、図示して下さい（C30、C35、C45 全てについて）。中性子束分布は前節で説明した ShowNeutronFluxAlongXAxis メソッドで取り出して下さい。また、C30、C35、C45 間の結果の違いについて説明して下さい。

6 高速中性子炉の炉心燃料の中性子束エネルギースペクトルの計算

これまでの Takeda ベンチマークの計算や KUCA の計算では、中性子束のエネルギーは 1 群もしくは 2 群で取り扱っていました。これらの計算は一連の炉物理計算の手続きの中で「下流」にあたるわけですが、「上流」に近いところでは中性子束のエネルギー依存性をもう少し詳細に取り扱っています。その例として、ここでは高速中性子炉の炉心燃料に関する中性子束のエネルギースペクトル¹² 計算を行いたいと思います。

ここで対象とするのは、中型高速中性子炉の炉心燃料集合体について、燃料ペレットや被覆管、冷却材、ラップ管といった非均質構造を均質（ごちゃごちゃに混ぜあわせたもの）としてモデル化したものとなります。さらに、このような媒質が無限に広がると仮定するので、中性子束は空間的に一様となり、所謂「一点炉」となります。この場合、中性子束の空間的な振る舞いを考慮する必要がないため、中性子束のエネルギー的な振る舞いのみ、すなわち減速過程のみを考慮することになります。減速といっても高速中性子炉を対象としていますので、軽水のような減速材は媒質には含まれておらず、減速の程度は非常に小さいものとなります。

それではパッケージの monju ディレクトリ中の main.innercore.cxx を走らせてみて下さい。このプログラムは、対象媒質に対して中性子の減速計算を行い、中性子無限増倍率 k_{∞} と、エネルギー群毎の中性子束、中性子吸収断面積 Σ_a 、核分裂による中性子生成断面積 $\nu\Sigma_f$ 、(n,2n) 反応断面積 $\Sigma_{(n,2n)}$ を出力します。以下にその例を示します。

Listing 3: 高速中性子炉の炉心燃料の減速計算の出力例

```

1 # K-inf : 1.40722
2
3
4 # E_top      Flux      Sigma.a      Nu-Sigma.f      (n,2n)
5 1.00000e+07  1.28650e-01  1.47747e-02  3.37596e-02  7.06953e-03
6 7.78801e+06  3.28003e-01  1.14022e-02  2.66400e-02  1.79666e-03
7 6.06531e+06  7.21167e-01  7.69620e-03  1.81628e-02  1.02185e-05
8 4.72367e+06  1.18850e+00  7.19271e-03  1.75179e-02  0.00000e+00
9
10          (omitted)
11
12 8.76425e-01  5.19136e-10  3.71924e-01  1.74176e-01  0.00000e+00
13 6.82560e-01  5.15244e-10  2.85244e-01  3.08249e-01  0.00000e+00
14 5.31579e-01  2.40072e-10  5.01584e-01  7.51417e-01  0.00000e+00
15 4.13994e-01  4.69720e-11  1.78387e+00  2.97982e+00  0.00000e+00
16 3.22419e-01  4.00878e-10  1.52963e+00  2.68211e+00  0.00000e+00

```

以下の課題は、main.innercore.cxx を使って解いて下さい。

課題 6.1：高速中性子炉の炉心燃料について、中性子束エネルギースペクトルを図示して下さい。なお、横軸は対数軸とし、縦軸は線形軸、対数軸の 2 通りとすること。

中性子束に加えて、反応率 $\Sigma\phi$ を観察することで、どのようなエネルギー領域で反応が起こっているかを知ることが出来ます。

課題 6.2：高速中性子炉の炉心燃料について、吸収反応率 $\Sigma_a\phi$ 、核分裂生成反応率 $\nu\Sigma_f\phi$ 、(n,2n) 反応率 $\Sigma_{(n,2n)}\phi$ のエネルギースペクトルを図示して下さい。なお、横軸は対数軸とし、縦軸は線形軸、対数軸の 2 通りとすること。

エネルギー群依存で定義される反応率 $\Sigma_g\phi_g$ を全ての g について和をとることで、エネルギー積分された反応率を得ることが出来ます。エネルギー積分された吸収反応率を R_a 、生成反応率を R_f 、(n,2n) 反応率を $R_{(n,2n)}$ と記述した場合、中性子無限増倍率は $R_f/(R_a - R_{(n,2n)})$ と計算することが出来ます ((n,2n) 反応は中性子が 1 個増加する反応ですので、負の吸収反応と見做されます)。

¹²中性子束のエネルギースペクトルとは、中性子束のエネルギー分布のことです。略して、「中性子スペクトル」と言う場合もあります。中性子束のエネルギー分布が高い側に偏っている場合は「硬いスペクトル」、低い側に偏っている場合は「軟らかいスペクトル」と言います。

課題 6.3：エネルギー積分された反応率を用いて無限中性子増倍率を計算し、プログラムの出力に表示されているものと一致することを確認して下さい。

なお、課題 6.2、6.3 については、プログラムに自分で変更を加えて実施するのもよいでしょう。

7 超小型臨界集合体 Godiva、Flatop-25 の計算

ここでは、米国ロスアラモス国立研究所において、核兵器開発のために構築された超小型の臨界集合体 Godiva、及び Flatop-25 の計算を行います。Godiva は金属ウラン（U-235 の濃縮度が 90% 以上）で構成される裸の球、Flatop-25 は金属ウラン球の外側に天然ウランを巻いた球体系であり、いずれも高速中性子炉に分類されます。臨界集合体 Flatop の体系図を Fig. 4 に示します。中心に核分裂性物質の「core」が配置され、その周囲を、漏れ出た中性子を反射して戻すための天然ウランが取り囲んでいます。

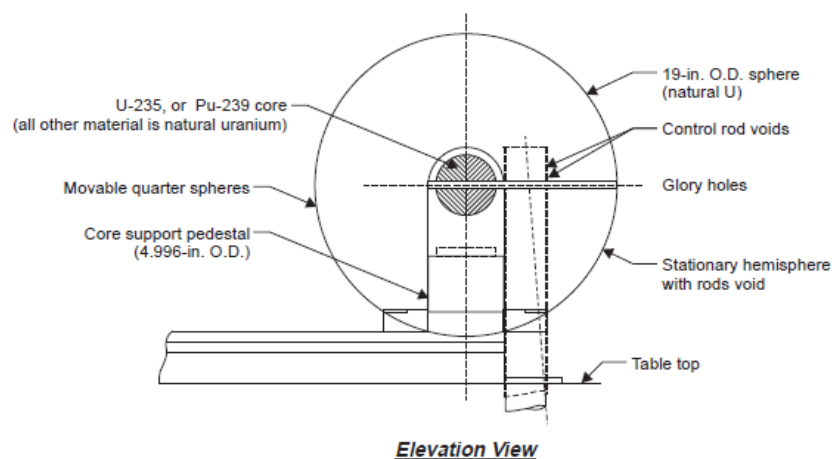


Fig. 4: Flatop の体系図

パッケージでは、CBGCAL/lanl.tutorial ディレクトリに、関連するデータが置かれています。

7.1 媒質の多群断面積の計算

単一もしくは複数の媒質から構成される原子炉の多群中性子輸送（もしくは拡散）方程式を解く際には、各媒質の多群断面積を予め計算しておく必要があります。前節の Takeda ベンチマークの問題では多群（2 群）断面積が予め与えられていましたが、ここでは CBZ により多群断面積を計算する方法について説明します。なお、高速中性子炉と熱中性子炉とでは計算の手続きが僅かに異なることに注意が必要です。

lanl.tutorial ディレクトリにおいて、多群断面積の計算を行うファイルは main.xs.cxx なので、適宜それを横目で眺めながら本稿を読み進めて下さい。

7.1.1 多群断面積ライブラリ CBZLIB の読み込み

媒質の多群断面積を計算するためには、個々の原子核の、中性子に対する反応断面積の情報（多群断面積セット）が必要となります。CBZ では CBZLIB と呼ばれる多群断面積セットが用意されており、それを事前に呼び出しておく必要があります。CBZLIB は、基になっている評価済み核データファイル、エネルギー群数に応じて様々存在します。高速中性子炉を計算する際の CBZLIB の読み込み例を以下に示します。

Listing 4: CBZLIB の読み込み（高速中性子炉計算時）

```

1 // xs library
2 string libdir("../CBGLIB/j4.70g.iwt7/");
3 XSLibrary xslib(libdir,"N-ENERGY");
4 string filename[]={
5     "U233","U234","U235","U238","Pu239",
6     "Pu240","Pu241","Pu242","Ga000",
7 };
8 xslib.ReadFile(9,libdir,filename);

```

2 行目の `string` 型の変数 `libdir` には、用いる CBZLIB のデータが置かれているディレクトリ名を指定します。CBZLIB は核種毎に異なったテキストファイルで同一のディレクトリにデータが置かれています。この例ではディレクトリ名として `j4.70g.iwt7` が指定されていますが、ここで使う CBZLIB は評価済み核データファイル JENDL-4.0 に基づく (`j4` に対応) 70 群の (`70g` に対応) ライブラリであることとなります (`iwt7` は多群断面積の作成に用いた重み関数を示しますが、ここでは分からなくてもよいです)。

3 行目では CBZLIB の実体である `XSLibrary` クラスのインスタンス `xslib` を作成しています。一つ目の引数ではライブラリデータが置かれているディレクトリの位置を指定しています。二つ目の引数ではエネルギー群のエネルギー境界の情報が格納されているテキストファイルの名前を指定しており、この部分は変更する必要はありません。

4 行目から 6 行目では、読み込む核種の指定を行っています。中性子との反応断面積が評価されている原子核の種類数は 400 を越えますが、いちいちこれら全ての原子核のデータを読み込むのは無駄なので、計算に必要な (計算対象の媒質に含まれる) 核種のデータのみをここで指定します。

最終行では、`XSLibrary` クラスに実装されている `ReadFile` メソッドで核種毎のテキストファイルを読み込みます。引数は、一つ目が読み込む核種数に、二つ目が CBZLIB のディレクトリに、三つ目が読み込むファイルの名前の配列に、それぞれ対応しています。

高速中性子炉を計算する場合には、一般的に 70 群、もしくは 175 群のライブラリを用いますが、熱中性子炉を計算する場合には 107 群のライブラリを用います。参考までに、熱中性子炉を計算する際の CBZLIB の読み込み例を以下に示します。

Listing 5: CBZLIB の読み込み (熱中性子炉計算時)

```

1 // +++ xs library
2 string libdir("/home/chiba/CBGLIB/j4.107g.iwt7/");
3 XSLibrary xslib(libdir,"N-ENERGY");
4 string filename[]={
5     "U234","U235","U236","U238","F019",
6     "H001","O016","Al027","Si028","Si029",
7     "Si030","Cu063","Cu065","Zn000","Mn055"
8 };
9 xslib.ReadFile(15,libdir,filename);
10
11 // +++ thermal scattering data read
12 string libth("/home/chiba/CBGLIB/j4.107g.iwt7/Thermal/");
13 xslib.GetLibData(10010).GetThScat().ReadFile(libth,"H.H2O");
14 xslib.GetLibData(80160).GetThScat().ReadFile(libth,"O016");

```

ここで注意すべきなのは、高速中性子炉用の CBZLIB を読み込む場合と異なり、熱中性子散乱データ (Thermal scattering data) も読み込む必要があるという点です。それがこの例での 12 行目から 14 行目に対応しており、ここでは水分子中の水素原子と、(自由気体と仮定した) 酸素原子の熱中性子散乱データを外部ファイルから読み込んでいます。このデータを読み込んで CBZLIB に含めることにより、熱中性子エネルギー領域における散乱での分子束縛効果を考慮することが可能となります。媒質に水を含む熱中性子炉を解析する際には、このデータを含めることは必須となります。

7.1.2 媒質データの作成

ここでは媒質データの作成方法について述べます。媒質データとしては、基本的には、含まれる原子核の種類とその数密度、媒質の温度が挙げられます。CBZ では `Medium` クラスが媒質に対応し、これらのデータを保持します。Medium クラスのインスタンスの作成例を以下に示します。

Listing 6: 媒質データの作成

```

1 // *** Medium data
2 Medium god_med; // godiva
3 god_med.PutImax(group);
4 god_med.PutPL(pl,1);
5 int god_mat[]={922340,922350,922380};
6 real god_den[]={4.9184e-4, 4.4994e-2, 2.4984e-3};
7 god_med.PutNuclide(3,god_mat,god_den);
8 god_med.PutTemperatureForAllNuclide(300.);

```

2 行目では `god_med` という名前の `Medium` クラスのインスタンスを作成しています。

3 行目では `Medium` クラスの `PutImax` メソッドにより、媒質データのエネルギー群数を指定しています。この例では `int` 型の変数 `group` が引数として与えられています（この例では事前に `group` に値が与えられています）。なお、ここで与えるエネルギー群数は、用いる CBZLIB のエネルギー群数と一致させる必要があります。

4 行目では `PutPL` メソッドにより、その媒質データに対して散乱の非等方性をどの程度考慮するかを指定します。一つ目の引数が非等方散乱断面積で考慮する最大の（ルジャンドル多項式に展開した際の）次数に対応します。この例では `int` 型変数 `pl` にて指定しています。二つ目の引数については説明を省略しますが、通常は 1 を指定して下さい。

7 行目では `PutNuclide` メソッドにより、媒質に含まれる原子核の種類と数密度の情報を与えています。引数の一つ目が原子核の種類数、二つ目が原子核の ID¹³、三つ目が数密度に対応しています。この例では、含まれる原子核の ID は 5 行目で、数密度は 6 行目でそれぞれ与えられています。

最下行では `PutTemperatureForAllNuclide` メソッドにより、媒質に含まれる全ての原子核の温度を指定しています。温度（単位は K）は引数で与えます。

解析する体系によっては、空気の媒質データを必要とする場合があります。そのようなときのために、空気の媒質データの作成例を以下に示します。MAT 番号の 70140、80160 はそれぞれ N-14、O-16 に対応しています。

Listing 7: 空気の媒質データの作成

```
1 // (air)
2 Medium air;
3 air.PutImax(group);
4 air.PutPL(1);
5 int mat0[] = {70140, 80160};
6 real den0[] = {3.2269e-5, 8.6569e-6};
7 air.PutNuclide(2, mat0, den0);
8 air.PutTemperatureForAllNuclide(293.);
```

7.1.3 媒質の実効微視的断面積と巨視的多群断面積の計算、およびファイル出力

媒質に含まれる核種の断面積が中性子の入射エネルギーに対して強い依存性を持っている場合（共鳴構造を有している場合）、多群（すなわちエネルギー平均）の微視的断面積は、その核種が媒質にどの程度含まれているか（換言すると、その核種が他の核種にどの程度希釈されているか）によって、大きく値が変わります¹⁴。そして、媒質に応じて計算される多群の微視的断面積を多群**実効**微視的断面積と呼ぶこともあります。

また、媒質に含まれる全ての核種の実効微視的断面積 σ が得られれば、数密度 N を用いて媒質の巨視的断面積 Σ ($= \sum_i N_i \sigma_i$, i は核種を示す index) を計算することができます。この巨視的断面積が中性子輸送（拡散）方程式に現れるパラメータとなります。実効微視的断面積と巨視的断面積の計算例を以下に示します。

Listing 8: 実効微視的断面積、巨視的多群断面積の計算とファイル出力

```
1 // *** self-shielding treatment
2 OnePointCalculator opc;
3 opc.CalSelfShieldingInfiniteSystem(god_med, xslib);
4 opc.CallIncidentEnergyDependentChi(god_med, xslib);
```

¹³CBZ における原子核に対する ID の与え方ですが、原子核の原子番号を Z 、質量数を A 、励起レベルを L としたとき、その原子核の ID を $Z \times 10000 + A \times 10 + L$ と定義します。基本的に大部分の原子核は基底状態にあるため $L = 0$ となりますが、一部の原子核で励起状態で準安定となるものがある（例えば Am-242 など）ため、その場合は $L > 0$ となります。また、天然同位体組成の原子核については、質量数を 0 として定義します。例えば天然同位体組成の鉄については、ID は「260000」として与えられます。

¹⁴一般に、断面積が大きい値を示すエネルギーでは中性子束は歪みます（小さくなります）。多群断面積は中性子束荷重で平均化しますので、断面積が大きい値を示すエネルギー点での荷重が小さくなるほど、エネルギー平均の断面積は小さくなります。この効果を**共鳴自己遮蔽効果**と呼びます。ただし、断面積が共鳴構造を有する核種が他の核種に強く希釈されている場合には、共鳴による中性子束の歪みは小さくなるため、エネルギー平均の断面積は大きくなります。

```

5
6  god_med.WriteFile(xmdir,"god",micxs_write);

```

微視的多群断面積、巨視的断面積の計算には CBZ の `OnePointCalculator` クラスを用いるので、そのインスタンス `opc` を 2 行目で作成しています。

3 行目では `OnePointCalculator` クラスのメソッド `CalSelfShieldingInfiniteSystem` により、`Medium` クラスのインスタンスに含まれる核種の微視的多群断面積を計算し、さらに媒質の巨視的断面積の計算を行っています。このメソッドでは、一つ目の引数に計算する `Medium` クラスのインスタンスを、二つ目の引数に計算に用いる `XSLibrary` クラスのインスタンスを、それぞれ指定します。

4 行目では `OnePointCalculator` クラスのメソッド `CalIncidentEnergyDependentChi` により、核分裂スペクトルの計算を行っています¹⁵。核分裂性物質を含む媒質に対してはこのメソッドを行っておいたほうが確実ですが、中性子エネルギーが比較的高い超小型原子炉を除けば、このメソッドは使わなくてもよいです。

ここまでで中性子輸送計算に必要な巨視的多群断面積が作成され、`Medium` クラスのインスタンス `god_med` に情報が格納されました。引き続いて中性子輸送計算を行う場合には、このインスタンスをそのまま用いればよいわけですが、一般的に、引き続いて行う中性子輸送計算は異なるファイルで行うため、ここで作成した `god_med` のデータを一度外部ファイルに保存させる必要があります。ファイルへの保存は最下行で示されているように、`Medium` クラスの `WriteFile` メソッドで行います。このメソッドの引数は、一つ目がデータファイルを書き出すディレクトリ、二つ目がファイル名、三つ目が微視的断面積もファイルに出力するかを指定する `boolean` 変数となっています。

なお、熱中性子炉の媒質を計算する場合には、熱中性子散乱を考慮する必要があります。その場合の巨視的断面積の計算例を以下に示します。

Listing 9: 実効微視的断面積、巨視的多群断面積の計算とファイル出力（熱中性子炉の場合）

```

1  // self-shielding treatment
2  OnePointCalculator opc;
3  opc.CalSelfShieldingInfiniteSystem(sol1,xslib);
4  opc.CalThermalScatteringMatrix(sol1,xslib,3.93);
5
6  sol1.WriteFile("./XS/CBGXS/" ,"SOL1");

```

この例では `Medium` クラスのインスタンス `sol1` の巨視的断面積を計算しています。熱中性子散乱を考慮させるため、`OnePointCalculator` クラスの `CalThermalScatteringMatrix` メソッドを用いています。このメソッドの三つ目の引数は熱中性子散乱を考慮する上限のエネルギーに対応しており、この例のように 3.93 eV を指定してやれば十分です。なお、この例では `CalIncidentEnergyDependentChi` メソッドを省略しています。

それでは `main.xs.cxx` を実行させて、Godiva、Flatop-25 を構成する媒質の断面積データを作成してみてください。XS/CBGXS ディレクトリに `god`、`fu`、`fb` という三つのファイルが生成されたことと思います。このうち `god` は Godiva の燃料領域、`fu`、`fb` はそれぞれ Flatop-25 の燃料、ブランケット領域の断面積データに対応しています。

7.2 原子炉の計算

原子炉の各媒質の多群断面積が計算されたので、次は原子炉の中性子の振る舞いを支配する多群中性子輸送（拡散）方程式を数値的に解き、原子炉内の中性子の挙動を評価します。ここでは、超小型臨界集合体 Flatop-25 を、離散座標法に基づく中性子輸送ソルバー SNR で計算する場合を例として示します。

¹⁵CBZLIB には入射エネルギー依存の核分裂スペクトルが与えられていますが、中性子輸送計算ソルバーでは入射エネルギーに依存しない核分裂スペクトルを用います。そこで、このメソッドでは、近似的に媒質の中性子エネルギースペクトルを計算し、それを用いて入射エネルギーに依存しない核分裂スペクトルの計算を行っています。

パッケージでは `main.flatu.cxx` がこの部分に該当していますので、以降はそのファイルを横目で眺めながら読み進んでいって下さい（Godiva の場合は `main.godiva.cxx` が対応します）。

7.2.1 体系情報の作成

球形の原子炉の体系情報は `CartMeshInfo` クラスにより定義します。Flattop-25 の体系情報の作成例を以下に示します。

Listing 10: 球形の原子炉の体系情報の作成

```

1 // CartMeshInfo
2 CartMeshInfo cmi;
3 int xm[]={30,16,30}; // for flattop-u
4 real xl[]={6.1156,3.,15.0086}; // for flattop-u
5 int mat[]={0,1,1};
6 cmi.PutMeshInfo(3,xm,xl,mat);
7 cmi.PutBoundaryCondition("Reflective","Vacuum");

```

2 行目では `CartMeshInfo` クラスのインスタンス `cmi` を作成しています。

6 行目では `cmi` に対して、領域の分割、物質配置、メッシュの分割情報を `PutMeshInfo` メソッドにより与えています。引数はそれぞれ、一つ目が領域分割数を、二つ目が各領域のメッシュ分割数を、三つ目が各領域の幅（単位は cm）を、四つ目が各領域の物質番号を示しています。この例では、領域のメッシュ分割数、幅、物質番号が 3 行目から 5 行目に与えられています。球体系を扱っているため、一つ目の領域の幅は球中心位置からの幅となります。すなわち、この例では、半径 24.1cm の球の内側に、半径 6.1156cm の球があると考えればよいわけです。なお、各領域の幅を領域の外側位置（球の中心を 0 としたときの各領域の右端位置）で指定することも可能で、その場合は `PutMeshInfo` メソッドの五つ目の引数として文字列 `“cumulative”` を追加すればよいです。

最終行では境界条件を指定しており、球形状の原子炉の場合はこの例の通り、左側に `Reflective`（完全反射）、右端に `Vacuum`（真空）を指定します。

7.2.2 中性子輸送計算のための角度求積セットの指定

離散座標法に基づく中性子輸送ソルバーでは、いくつかの特定の方向を向いた中性子束についてのみ輸送計算を行い、（スカラー）中性子束などの角度積分値は求積法で求めます。そのため、この求積法に用いる角度離散点とその重みからなる求積セットを計算の前に指定する必要があります。角度求積セットの指定の例を以下に示します。

Listing 11: 角度求積セットの指定

```

1 // Quadrature
2 SNRQuadrature quad1(pl);
3 quad1.PutDoubleGaussian(24);

```

2 行目では、`SNRQuadrature` クラスのインスタンス `quad1` を作成しています。引数としては非等方散乱断面積に対して考慮する最大次数を与えます。

3 行目では `PutDoubleGaussian` メソッドにより、二重ガウス求積セット（Double Gaussian angular quadrature set）を指定しています。引数として与えられているのはその離散点数となります（離散点数が n の場合、次数は $n/2 - 1$ となります）。

非等方散乱断面積に対する次数、求積セットに与える離散点数はそれぞれ十分大きな値とする必要があります。すなわち、これらをパラメータとした計算を実施し、これらの値を増加しても計算結果に有意に影響を与えない程度（例えば実効増倍率では 0.0001 以内程度）まで大きな値を設定する必要があります。

7.2.3 媒質情報の読み込み

ここでは計算対象としている原子炉に含まれる媒質の情報を外部ファイルから読み込みます。すでに体系情報の作成の箇所でも説明したように、この例で対象としている原子炉 Flattop-25 は二種類の媒質（濃縮ウランの燃料と、天然ウランのブランケット）から構成されています。媒質情報のファイルからの読み込みの例を以下に示します。

Listing 12: 媒質情報の読み込み

```

1 // Medium
2 Medium med[mednum];
3 string mdir("./XS/CBGXS/");
4 med[0].ReadFile(mdir,"fu",5); // o
5 med[1].ReadFile(mdir,"fb",5); // o

```

まず、Medium クラスのインスタンスの配列 med を作成します。この例では int 型の変数 mednum が用いられていますが、この値は 2 となっています。Medium クラスにはファイルからデータを読み込むメソッドとして ReadFile が実装されており、引数としてファイルの位置、ファイル名を指定します。

7.2.4 計算オプションの設定

中性子輸送計算には反復計算の収束条件、最大反復回数などの計算オプションが必要となります。その情報を持つのが GeneralOption クラスであり、通常の輸送方程式（前進式 forward equation）を解くか、随伴式 adjoint equation を解くかもこのクラスで定義します。GeneralOption クラスの使用例を以下に示します。

Listing 13: 計算オプションの設定

```

1 // GeneralOption
2 GeneralOption opt;

```

この例では GeneralOption クラスのインスタンス opt を作成しています。インスタンス作成時には種々の計算条件はデフォルト値に設定されます。解くべき方程式については、デフォルトでは前進式に設定されます。随伴式を解く場合には、PutAdjointCal メソッドを用います。

7.2.5 中性子輸送方程式の数値計算

以上により中性子輸送計算に必要な情報は全て準備されました。ここでは中性子輸送計算について説明します。

中性子輸送方程式を解くためのプログラムは CBZ にいくつか備えられていますが、球体系を離散座標法を用いて計算する場合は SNR を用います。SNR を用いた輸送方程式の計算例を以下に示します。

Listing 14: SNR による計算

```

1 // SNR for adjoint
2 SNRSystem snr(dim,group,mednum); //group & medium
3 snr.AddMedium(med[0]);
4 snr.AddMedium(med[1]);
5 snr.PutCartMeshInfo(cmi,"Sphere");
6 snr.PutPL(pl);
7 snr.SetQuadrature(&quad1);
8 snr.PutGeneralOption(opt);
9 snr.PutWriteFlux();
10 real k1=snr.CalIgen();

```

2 行目では SNR の実体となるクラス SNRSystem のインスタンス snr を生成しています。引数は一つ目が次元数（球体系の場合は 1）、二つ目がエネルギー群数、三つ目が原子炉に含まれる媒質の数に対応しています。

3、4 行目では媒質を示す `Medium` クラスのインスタンスの情報を追加していきます。追加された順番で SNR は `Medium` クラスのインスタンスに物質 ID を付与していくため、この例では SNR は `med[0]` の物質 ID を 0、`med[1]` の物質 ID を 1 と認識しています。

5 行目では体系情報を示す `CartMeshInfo` クラスのインスタンスの情報を SNR のインスタンスに与えています。引数の二つ目は体系の形状を定義する文字列であり、球体系を扱う場合はこの例の通りにして下さい。

6 行目では非等方散乱断面積で考慮する最大次数を定義します。

7 行目では作成した角度求積セットを定義する `SNRQuadrature` クラスのインスタンスを、8 行目では `GeneralOption` クラスのインスタンスを `snr` に与えます。

9 行目は角度中性子束の情報を SNR に持たせておくという指定であり、通常は不要です（感度計算、摂動計算を行う際には角度中性子束の情報が必要となるため、このメソッドを行う必要があります）。

最後の 10 行目で `snr` に中性子輸送計算を行わせ、得られた実効増倍率を `real` 型の変数 `k1` に代入しています。

課題 7.1：それでは実際に `main.godiva.cxx`、`main.flatu.cxx` を実行させてみて下さい。実効増倍率としてどの程度の値が得られましたか？この体系は臨界ですので、実効増倍率の実験値はほぼ 1.0 となります。計算値がどの程度実験値と一致しているか確認してみてください。また、計算値と実験値のずれは、燃料領域の厚さにしてどの程度か評価してみてください。

7.2.6 炉中心反応率比の計算

以上では、超小型臨界集合体 Godiva、Flatop-25 の中性子実効増倍率を計算する方法について説明してきましたが、炉物理パラメータは中性子増倍率以外にもいろいろあります。ここでは臨界集合体を用いた臨界実験において測定される「反応率比（スペクトルインデックス）」の計算方法について説明します。

反応率比の計算には、反応断面積に加えて着目位置での中性子束のエネルギースペクトルの情報が必要となります。従って、中性子輸送計算を行ったあとに初めて計算可能となります。

体系の任意のメッシュにおける微視的反応率比を計算するメソッドとして、`GeneralSystem` クラスには `CalPointReactionRateRatio` が実装されています。その使用例を以下に示します（サンプルの `main.godiva.cxx` の下側にコメントアウトされています）。

Listing 15: `CalPointReactionRateRatio` メソッドの使用例

```
1 // (Reaction rate ratio calculation)
2 int rrr_pos=0;
3 real tmp1=snr.CalPointReactionRateRatio(rrr_pos,922380,sigf,922350,sigf);
4 cout<<"#\n#_Reaction_rate_ratio_of_U-238(n,f)_to_U-235(n,f)_:_"<<tmp1<<"\n";
```

引数ですが、一つ目が反応率比を計算するメッシュ位置となります。この例では 0 が指定されているので、最も中心寄りのメッシュで計算されることになります。二つ目と三つ目の引数は反応率比の分子の反応率を定義するためのもので、それぞれ核種 ID と反応名（核分裂反応だと `sigf`、捕獲反応だと `sigc` を指定）となっています。この例では U-238 の核分裂反応となっています。四つ目と五つ目の引数は分母の反応率を定義するためのものです。

注意しなければならないのが、微視的反応率比を計算する場合は、各媒質データ（`Medium` クラスのインスタンス）が核種毎の微視的断面積データを持っている必要がある点です。多群断面積を作成するファイル `main.xs.cxx` では、微視的断面積の情報はファイルに書き出さなくなっていますので、これを指定する `boolean` 変数 `micxs_write` を `true` にして、多群断面積を再作成する必要があります。

8 臨界実験装置 ZEBRA に構築された炉心 MZA の計算

ここではナトリウム冷却高速中性子炉の特性を把握するために英国の臨界実験装置 ZEBRA で構築された炉心 MZA の解析を行います。パッケージでは、CBGCAL/mza.tutorial ディレクトリに、関連するデータが置かれています。

ここで、繰り返しになるかもしれませんが、簡単に原子炉、臨界集合体の核特性解析方法について説明します。

実機原子炉では、複数本の燃料ピンが規則的に配置された燃料集合体が炉心内に規則的に配置されます。一方、臨界集合体では、さまざまな物質板が規則正しく並べられた入れ物が、炉心内に規則的に配置されます。臨界実験装置 ZEBRA では、鉛直方向に延びた入れ物に薄い物質プレートが積み上げられる形で配置されます。これは実機原子炉の燃料集合体に該当するもので、**燃料要素**と呼ばれます。燃料要素の概念図を Fig. 5 に示します。物質板の一辺の長さは 2 インチ (5.08 cm) 程度となっています。

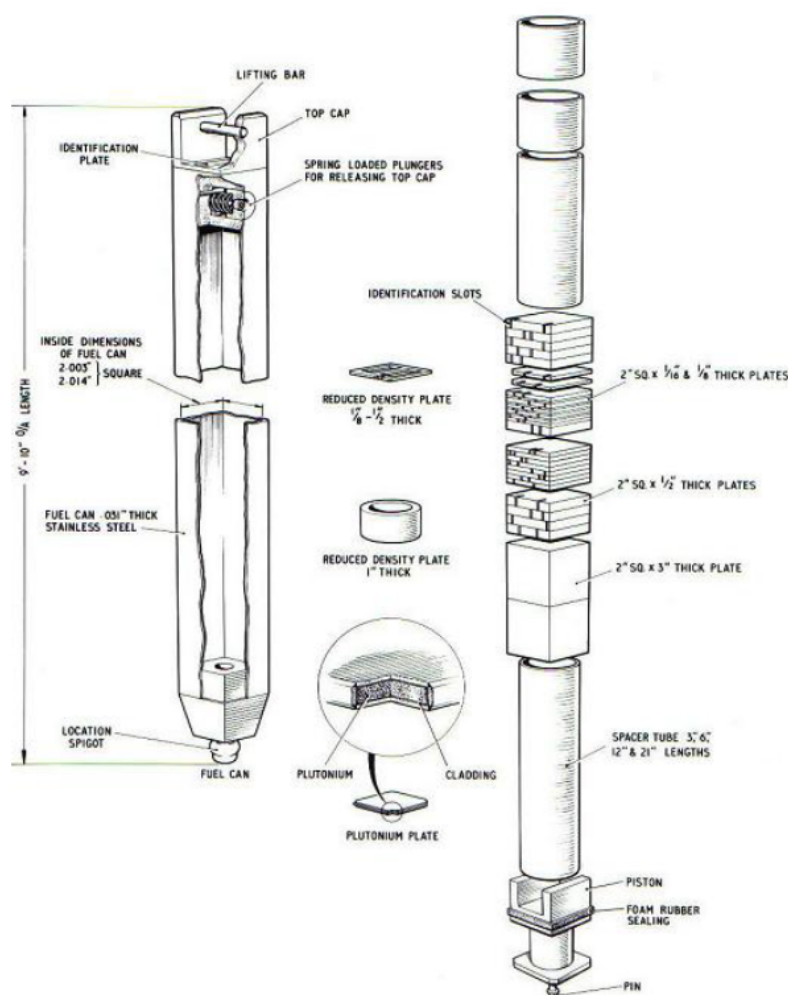


Fig. 5: 臨界実験装置 ZEBRA における燃料要素

また、原子炉の中央部に配置される燃料要素内には、燃料領域を構成する部分、燃料領域を上下に挟むブランケット領域を構成する部分、さらにブランケット領域を挟む中性子反射体（もしくはガスプレナム）領域を構成する部分があり、その各々は、規則正しく配置された種々の物質プレートで構成されます。この燃料要素内の燃料領域部分、ブランケット部分などを**単位格子**と呼びます。Fig. 6 に MZA 炉心で燃料領域を構成する単位格子の物質配置例を示します。Pu プレート（図中「Pu METAL」）やナトリウムプレート（同「Na」）で構成されており、実機高速炉の燃料集合体の組成を模擬しています。

Na
UO2
Na
Pu METAL
C
UO2
Na
Na
UO2
SS
Pu METAL
Na
UO2
Na

Fig. 6: MZA 炉心の単位燃料格子の物質配置例

原子炉や臨界集合体の全炉心体系を計算する際に、燃料集合体、燃料要素内の詳細な構造を考慮することは非常に大変です。そこで、一般的には、単位格子の計算と全炉心の計算を分離する取り扱いをします。前者は**格子計算**、後者は**炉心計算**と呼ばれます。

格子計算では、単位格子のみを取り扱い、それに周期境界条件を課すことにより、単位格子が無限に配列されるような系を考え、その系での中性子輸送方程式を解きます。その後、単位格子と等価となるような均質の媒質を作成し、それを炉心計算で用いるようにします。すなわち、炉心計算においては単位格子は均質な媒質として扱われ、内部の詳細な構造は全く考慮する必要がなくなります。この単位格子に対する操作は**均質化**と呼ばれ、炉心解析での重要な手続きとなっています。

なお、このチュートリアルで用意されている入力データ `main.xs.cxx` では、単位格子に含まれる種々の原子核の数密度はすでに体積重みで均質化（平均化）されたものとして与えられていますので、この入力を用いた場合、単位格子は単一の物質で構成されていると見做されます。

MZA の炉心体系図を **Fig. 7** に示します。左は上から見た図で、アルファベット A から D で示されている箇所に燃料を含んだ燃料要素が装荷されています。その外側で何も記載されていないところにはブランケットを含んだ燃料要素が装荷されます。また、右は横から見た図（全炉心の 1/4）であり、軸方向中心に燃料単位格子（C11-1A 等、「Core」と記載されている領域）が、その上下に軸方向ブランケット（B11-1 等）が配置されていることが分かります。

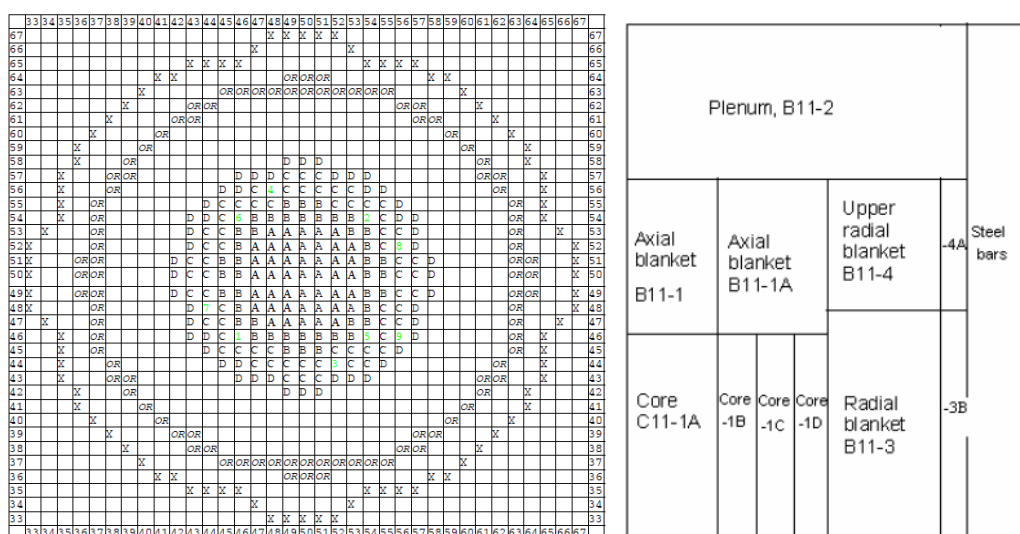


Fig. 7: MZA の炉心体系図（左：径方向、右：軸方向）

8.1 格子計算（多群断面積の計算）

MZA は 10 種を超える単位格子（単位媒質）から構成されますので、その各々について多群断面積を計算する必要があります。入力ファイル `main.xs.cxx` を用いて多群断面積の計算を行ってみましょう。XS/CBGXS ディレクトリに多群断面積データが生成されることと思います¹⁶。

ここで、`main.xs.cxx` を眺めてみて下さい。この入力ファイルでは 15 個の媒質の多群断面積を作成しますが、「C11」で始まるものは炉心燃料に、「B11」で始まるものはブランケットに、それぞれ対応します。また、「C11-1AV」「B11-1V」のように末尾に「V」がついているものは、「V」がついていない媒質からナトリウムが除去されたことを意味します。つまり、「C11-1AV」は「C11-1A」から冷却材模擬物質であるナトリウムが除去されたものとなります。これらは後述する Na ボイド反応度を実験によって測定する際に使われます。なお、実験は室温で行われているため、ナトリウムは固体状のプレートとなっています。実際の実験では、ナトリウムがボイド化したことを模擬するためステンレスで覆われた中空のプレートをナトリウムプレートの代わりに設置しています。

8.2 炉心計算

8.2.1 中性子実効増倍率の計算

はじめに、MZA の XYZ 体系の固有値（実効増倍率）を計算してみましょう。`main.xyz.cxx` を走らせてみて下さい。固有値計算が始まって、ちょっとしたら終わるはずです。

`main.xyz.cxx` の中身を眺めてみて下さい。はじめに燃料要素の情報を定義し（`Assembly`、`AssemblySet` のあたり）、次に、それが XY 平面上でどのように配置されるかを指定しています（`CartCore` のあたり）。その後、断面積データを読み込んで、固有値計算を行います。

次に、いろいろ計算オプションを変えて計算してみましょう。`main.cxx` を開いて下さい。

ファイルの頭部分で `string solver='PLOS'` という箇所があります。ここでは、使用するソルバー名を指定しますが、さきほどの計算は拡散ソルバー PLOSで行っています。では、輸送ソルバー SNT で計算してみましょう。PLOS を SNT と書き換えて実行してみてください。始めに、PLOSE というソルバーが起動しますが、これは初期値を求めるための端メッシュ差分拡散ソルバーです（PLOS は中心メッシュ差分）。なお、PLOSE は別に、SNT 内部の拡散合成加速のソルバーとしても使われています。PLOSE による計算が終了して初期値が推定されたのち、SNT による計算が始まります。

PLOS 及び SNT で得た実効増倍率はどの程度になったのでしょうか？この体系の実効増倍率の実験値としては 1.0100 が与えられています。大きく異なる結果が得られたと思いますが、これは格子計算（多群断面積の計算）において、単位格子の非均質な構造を無視し、体積重みで均質化された核種数密度データを用いていることに起因しています。実は、パッケージには燃料単位格子 C11-1A、C11-1B、C11-1C、C11-1D について、その非均質構造を一次元平板にモデル化した格子計算のための入力データ `main.xs_het.cxx` が含まれています。それではこの入力を走らせてみましょう。この入力データでは、Fig. 6 のような体系に対して、中性子輸送方程式を衝突確率法により解き、中性子束分布を求め、それを用いて均質断面積を計算させています。均質断面積を定義するファイルは XS/CBGXS ディレクトリに同一の名前で上書きされることになります。このようにして格子計算で非均質構造を考慮して得た均質断面積を用いて、再度、`main.xyz.cxx` により体系の実効増倍率を求めてみましょう。だいたい実験値（1.0100）に近い値が得られるものと思います。

¹⁶ さきほどは Flattop-25 の計算を行っていましたが、ここでは MZA が計算対象となっていますので、新たに多群断面積の計算が必要になります。

課題 8.1：非均質構造を考慮した格子計算により均質断面積を求めることにより実効増倍率が増加する結果が得られます。その要因は複数ありますが、そのひとつとして格子平均の中性子吸収断面積が低下することが挙げられます。格子平均の中性子吸収断面積 $\tilde{\Sigma}_a$ は、非均質体系における領域 i の中性子束分布を ϕ_i 、体積を V_i としたとき、以下の式で求められます。

$$\tilde{\Sigma}_a = \frac{\sum_{i=1}^I \Sigma_i \phi_i V_i}{\sum_{i=1}^I \phi_i V_i} \quad (1)$$

非均質構造を無視した格子計算 (`main.xs.cxx` に対応) は、中性子束分布 ϕ_i を平坦と仮定していることとほぼ同一と考えてよいです。 $\tilde{\Sigma}_a$ を計算する際、 ϕ_i の分布を考慮することによって、考慮しない場合と比較して $\tilde{\Sigma}_a$ が小さくなる理由を述べて下さい。

8.2.2 摂動理論によるナトリウムボイド反応度の計算

それでは次に、高速炉の炉心設計で非常に重要なパラメータであるナトリウムボイド反応度の計算を試みましょう。ナトリウムボイド反応度とは、冷却材であるナトリウムが沸騰してボイド化した場合に炉心に添加される反応度です。

反応度 ρ はふたつの状態の炉心の実効増倍率 k_1 、 k_2 から、 $\rho = 1/k_2 - 1/k_1$ として計算されますが、 k_1 と k_2 との差が小さい場合には反応度の有効数字は小さくなってしまいます。そこで、通常、反応度計算は**摂動理論**を用いて行います。摂動理論については、炉物理の教科書や、千葉のホームページの「炉物理プログラム演習：随伴方程式と摂動計算」を読んで勉強して下さい。

それではナトリウムボイド反応度を計算してみましょう。なお、ここでは格子計算を均質モデルで揃える必要があるため、`main.xs.cxx` を再度流して、全ての格子の均質断面積を再計算して下さい。

`main.xyz.vid.cxx` を眺めて下さい。燃料要素として 13 種類が定義されていますが、最後の 5 つでは、燃料領域、ブランケット領域（の一部）で冷却材がボイド化しています（「c11-1av」「b11-1v」という名前で区別されています）。摂動計算では、基準体系の随伴中性子束と、摂動体系（すなわち冷却材がボイド化した体系）の中性子束が必要になりますので、基準体系と摂動体系について中性子輸送（拡散）方程式を解きます。CBZ では収束条件等の計算条件は `GeneralOption` クラスで定義され、そのインスタンスがソルバーに渡されます。随伴系の計算をするためには、`GeneralOption` クラスで `PutAdjointCal` メソッドを使います。`main.xyz.vid.cxx` の中を確認してみてください。

また、摂動体系の体系情報は、基準体系の集合体配置を定義する `CartCore` クラスのインスタンスに対して `ChangeAssembly` メソッドを実行させることにより体系を変更させて得ています。`ChangeAssembly(0,0,8)` とした場合は「X=0、Y=0 の集合体を No.8 集合体に置換せよ」というメソッドになります。この例では炉中心位置にある 9 つの燃料要素が、ナトリウムが一部ボイド化した燃料要素に置換されます。

ふたつの体系の固有値を計算したあと、`CalReactivity` メソッドにより反応度が計算されます。なお、随伴系では CMFD 加速は有効ではないことに注意して下さい。

それでは `main.xyz.vid.cxx` を実行してみましょう。固有値計算が二回行われ、最後に反応度が出力されたかと思えます¹⁷。

摂動計算によるナトリウムボイド反応度の計算例を以下に示します。

Listing 16: ナトリウムボイド反応度の計算例

```
1 # +++ Results +++
2 #
```

¹⁷CBZ では計算結果は全て画面上に出力されるので、ファイルに保存したい場合には、(`cbz` コマンドを利用していないならば) `./a.out > output` のようにして下さい。

```

3 # Yield      : -1.46472e-05
4 # Absorption : 1.60003e-04
5 # Scattering : 3.44868e-04
6 # N2N       : -1.68593e-10
7 # (Non-Leak) : 4.90224e-04
8 # Leakage-x  : -1.91749e-05
9 # Leakage-y  : -1.91785e-05
10 # Leakage-z  : -4.31141e-05
11 # (Leakage)  : -8.14674e-05
12 #
13 # ** Perturbation Cal. : 0.00040876
14 # ** Direct Cal.      : 0.00039815

```

摂動計算では、反応度は次のように要因別に、さらにはエネルギー群別に分類され出力されています。

- Yield 成分：摂動によって核分裂断面積が変化し、核分裂による中性子生成率が変化したことにより印加される反応度成分
- Absorption 成分：摂動によって吸収断面積が変化し、中性子消滅率が変化したことにより印加される反応度成分
- Scattering 成分：摂動によって散乱断面積が変化し、中性子スペクトルが変化したことにより印加される反応度成分
- (n,2n) 成分：摂動によって (n,2n) 断面積が変化し、(n,2n) 反応による中性子生成率が変化したことにより印加される反応度成分
- Leakage 成分：摂動によって全断面積が変化し、中性子漏洩率が変化したことにより印加される反応度成分

例えば、ナトリウムがボイド化した場合、媒質の吸収断面積はナトリウムの分だけ小さくなるはずです。それによる反応度効果は上記の Absorption 成分に対応します。

これら各成分の総和が Perturbation Cal. として出力されます。なお、Leakage 成分を除いた成分の和は「Non-leakage 成分」としてまとめられます。また、Leakage 成分は方向毎にも与えられます。

Direct Cal. はふたつの実効増倍率からの反応度計算値を示します。両者は収束精度内でおおよそ一致していることを確認して下さい（なお、固有値計算の実効増倍率に対する収束条件は 10^{-5} となっています）。

課題 8.2 : main.xyz.vid.cxx を流して得られるナトリウムボイド反応度の計算結果について、要因毎の反応度を横軸をエネルギーとして図示して下さい。

ナトリウムボイド反応度の Scattering 成分は高エネルギー領域では負の値をとる。このことは、 η 値 ($\nu\Sigma_f/\Sigma_a$) のエネルギー依存性で説明することが出来る。

課題 8.3 : エネルギー依存の η 値をプロットとし、その情報を用いて、課題 8.2 で得られたナトリウムボイド反応度の Scattering 成分が高エネルギー領域で負の値をとることを説明して下さい。なお、核分裂生成断面積 $\nu\Sigma_f$ 、吸収断面積 Σ_a は main.xyz.cxx の下部にコメントアウトしてある部分を流すことによって得ることが出来ます。

課題 8.4 : main.xyz.vid.cxx には、ナトリウムの一部がボイド化している燃料要素が 5 種類あります (Assembly クラスのインスタンスの asmi [8] から asmi [12])。デフォルトでは炉中心位置の燃料要素のいくつかを asmi [8] に置換した場合の反応度を計算させていますが、これを asmi [9] から asmi [12] の各々に置換した場合の反応度を要因毎に計算し、結果を表にまとめて下さい。なお、それぞれの燃料要素ではどの領域のナトリウムがボイド化されているのか、しっかり把握して下さい (asmi がどのように定義されているかを見れば分かります)。

8.2.3 実効遅発中性子割合の計算

原子炉の動特性計算や、反応度の dk/k' 単位からドル単位への換算に用いられる実効遅発中性子割合 β_{eff} も重要な炉物理パラメータです。入力ファイル `main.xyz.beff.cxx` はその計算を行うためのもので、詳細は省きますが、とりあえず流してみましょう。以下のような結果が得られる筈です。

Listing 17: 実効遅発中性子割合の計算例

```

1 #####
2 # Delayed neutron parameter calculation
3 #
4 # Beta_eff = 3.485725e-03
5 #
6 # [Nuclide-wise (relative contribution)]
7 #
8 # 922340 : 1.128802e-07 (3.238357e-05)
9 # 922350 : 1.705120e-04 (4.891724e-02)
10 # 922380 : 1.540003e-03 (4.418028e-01)
11 # 942390 : 1.455391e-03 (4.175289e-01)
12 # 942400 : 1.255714e-04 (3.602447e-02)
13 # 942410 : 1.899315e-04 (5.448837e-02)
14 # 942420 : 4.203142e-06 (1.205816e-03)
15 #
16 # [Nuclide&family-wise]
17 #
18 #           1           2           3           4           5           6
19 # 922340 6.118e-09 2.205e-08 2.026e-08 4.401e-08 1.498e-08 5.463e-09
20 # 922350 5.847e-06 3.681e-05 3.583e-05 6.624e-05 2.151e-05 4.276e-06
21 # 922380 1.983e-05 2.108e-04 2.471e-04 6.001e-04 3.462e-04 1.159e-04
22 # 942390 4.764e-05 4.163e-04 3.167e-04 4.753e-04 1.250e-04 7.448e-05
23 # 942400 3.987e-06 3.188e-05 1.881e-05 4.149e-05 2.251e-05 6.891e-06
24 # 942410 3.418e-06 4.281e-05 2.687e-05 6.638e-05 3.747e-05 1.299e-05
25 # 942420 8.229e-08 9.764e-07 5.237e-07 1.372e-06 9.469e-07 3.019e-07
26 # (sum) 8.081e-05 7.396e-04 6.458e-04 1.251e-03 5.537e-04 2.149e-04
27 #
28 # [Averaged decay constant]
29 # (beta-weight) (beta-weight-inverse)
30 # 1 1.292e-02 1.291e-02
31 # 2 3.072e-02 3.070e-02
32 # 3 1.282e-01 1.276e-01
33 # 4 3.384e-01 3.370e-01
34 # 5 1.277e+00 1.242e+00
35 # 6 3.447e+00 3.325e+00
36 #####

```

まずはこの体系の β_{eff} として 0.00349 が得られています。これはすなわち全核分裂中性子のうち遅発中性子が占める割合がおおよそ 0.35%であることを示しています。

β_{eff} は核分裂核種毎に定義される $\beta_{\text{eff},i}$ の総和として計算されるため、核分裂性核種毎の寄与も得ることができます。その結果が上記の例における Nuclide-wise (relative contribution) に対応しており、 $\beta_{\text{eff},i}$ が相対寄与（括弧内）とともに示されています。この体系では U-238 と Pu-239 からの遅発中性子が主であることが分かるでしょう。

さらに、 β_{eff} は遅発中性子先行核家系（もしくは先行核群）毎にも定義されることから、核分裂性核種毎でかつ先行核家系毎の数値もそのあとに示されています。

最後の Average decay constant は、先行核家系毎の崩壊定数の核種平均値 λ_i の結果を示しています（先行核の崩壊定数は核分裂性核種毎に異なる値が与えられるため、炉心全体としての値を得るためには平均化の操作が必要となります）。平均のとり方にいくつか異なる方法があり、ここでは2通りの方法に基づく結果が示されています。

これらの β_i や λ_i といったパラメータが、原子炉の動特性解析に用いられることになります。

9 おわりに

ということで、おおよその流れは掴めたものと思います。お疲れさまでした。それでは、それぞれ、目的に応じて CBZ を使ってみて下さい。