JAEA-Data/Code
2008-011

EVIRA:FLWR研究開発支援用インターフェース

EVIRA:Assistance Interface of Research and Development for FLWR

深谷 裕司

Yuji FUKAYA

次世代原子力システム研究開発部門
革新的水冷却炉設計グループ

Innovative Water Reactor Design Group
Advanced Nuclear System Research and Development Directorate

May 2008

Japan Atomic Energy Agency　日本原子力研究開発機構

JAEA-Data/Code

本レポートは独立行政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。
本レポートの入手並びに著作権利用に関するお問い合わせは、下記あてにお問い合わせ下さい。
なお、本レポートの全文は日本原子力研究開発機構ホームページ（http://www.jaea.go.jp）
より発信されています。

独立行政法人日本原子力研究開発機構　研究技術情報部　研究技術情報課
〒319-1195　茨城県那珂郡東海村白方白根 2 番地 4
電話 029-282-6387, Fax 029-282-5920, E-mail:ird-support@jaea.go.jp

This report is issued irregularly by Japan Atomic Energy Agency
Inquiries about availability and/or copyright of this report should be addressed to
Intellectual Resources Section, Intellectual Resources Department,
Japan Atomic Energy Agency
2-4 Shirakata Shirane, Tokai-mura, Naka-gun, Ibaraki-ken 319-1195 Japan
Tel +81-29-282-6387, Fax +81-29-282-5920, E-mail:ird-support@jaea.go.jp

© Japan Atomic Energy Agency, 2008

JAEA-Data/Code 2008-011

EVIRA: FLWR 研究開発支援用インターフェース

日本原子力研究開発機構　次世代原子力システム研究開発部門　設計統括ユニット
深谷　裕司

(2008 年　2 月 29 日受理)

革新的水冷却炉(FLWR)の研究開発支援用インターフェースの開発を実施した。本インターフェースにおいては、核計算コード（現時点においては MOSRA）による解析結果をグラフィカルな表示により簡便にアクセスできる手法の確立を目指し、FLWR の設計研究等において計算結果を効率的に把握することが可能となる。

今回開発されたインターフェースの機能としては、解析結果である MOSRA の PDS ファイルに直接アクセスし、結果を視覚的に表現する機能である。PDS ファイルへのアクセスについては、マウス操作により簡単に実現できる。また、表示機能については炉心の径方向や軸方向における出力等の分布などである。特に径方向分布については３D 表示ができ一目で把握できるようになっている。また、これらの表示については画像ファイルとして出力することができ、報告書等にそのまま利用できる。

これらの機能を有する本インターフェースは FLWR の研究開発において多いに資するものと期待できる。

大洗研究開発センター（駐在）：〒311-1393 茨城県東茨城郡大洗町成田町 4002

JAEA-Data/Code 2008-011

EVIRA: Assistance Interface of Research and Development for FLWR

Yuji FUKAYA

FBR Cycle System Design Synthesis Unit
Advanced Nuclear System Research and Development Directorate
Japan Atomic Energy Agency
Oarai-machi, Higashiibaraki-gun, Ibaraki-ken

(Received February 29, 2008)

The assistance interface of calculation for innovative water reactor for flexible fuel cycle (FLWR) has been developed. The interface system should have an ability to display the output of core calculation code, graphically and simply.

This Interface has an ability to access the result which is calculated by MOSRA code system as PDS files, and an ability to show the result graphically. The procedure to access the PDS files can be performed by using only mouse. Moreover, the distribution of the results, *e.g,* thermal power, can be shown as not only the axial graph but also the radial graph. Especially, for the radial graph, it can be shown as 3 dimensional graphics. These graphs can be saved as an image file, and the image file has the quality to use for report directly.

Thus the outcome of developing the EVIRA code is beneficial for the research on FLWR.

Keywords: Enhanced Visual Interface for Reactor Analysis (EVIRA), Innovative Water Reactor for Flexible Fuel Cycle (FLWR), Visual Interface

目次

１．序論 ---------- 1
２．開発環境 ---------- 2
　２．１　GUI ライブラリと実行環境について ---------- 2
　２．２　プログラミング言語について ---------- 4
３．炉心解析コードとの接続 ---------- 6
　３．１　接続方法の考察 ---------- 6
　３．２　PDS ファイルの構造と読み込み ---------- 6
４．インターフェースの構成 ---------- 9
　４．１　チャンネルマップ ---------- 13
　４．２　出力密度・径方向ピーキング ---------- 13
　４．３　ボイド率 ---------- 15
　４．４　燃焼度 ---------- 15
　４．５　バッチ ---------- 17
　４．６　ダクト ---------- 19
　４．７　ツール ---------- 19
５．表示部と画像ファイルへの出力 ---------- 23
　５．１　２次元表示 ---------- 23
　５．２　３次元表示 ---------- 25
６．プログラムの構成 ---------- 27
７．EVIRA におけるデータ処理法について ---------- 29
８．チュートリアル：1/6 境界において分割される集合体の扱い ---------- 33
９．結論 ---------- 43
参考文献 ---------- 43
付録 インストール法 ---------- 44

Contents

１．Introduction ---------- 1
２．Development Environment ---------- 2
　２．１　GUI Library and Development Environment ---------- 2
　２．２　Programming Language ---------- 4
３．Connection to Reactor Core Analysis Code ---------- 6
　３．１　Consideration for Connection Method ---------- 6
　３．２　Structure and Access Method of PDS Files ---------- 6
４．Structure of Interface ---------- 9
　４．１　Channel Map ---------- 13
　４．２　Power Density and Radial Peaking ---------- 13
　４．３　Void Fraction ---------- 15
　４．４　Burn-up ---------- 15
　４．５　Batch ---------- 17
　４．６　Duct ---------- 19
　４．７　Tool ---------- 19
５．Displaying Part and Output to Image File ---------- 23
　５．１　2 Dimensional Display ---------- 23
　５．２　3 Dimensional Display ---------- 25
６．Structure of Programming ---------- 27
７．Data Processing Method in EVIRA ---------- 29
８．Tutorial：Treatment of Fuel Assemblies Divided by 1/6 Boundary ---------- 33
９．Conclusion ---------- 43
References ---------- 43
Appendix How to Install ---------- 44

図・目次

Fig. 4.1 インターフェースの構成 ---------- 11
Fig. 4.2 境界位置の移動 ---------- 11
Fig. 4.3 チャンネルマップの制御部 ---------- 13
Fig. 4.4 出力密度・径方向ピーキング制御部 ---------- 15
Fig. 4.5 ボイド率制御部 ---------- 15
Fig. 4.6 燃焼度制御部 ---------- 17
Fig. 4.7 バッチ制御部 ---------- 19
Fig. 4.8 ダクト制御部 ---------- 19
Fig. 4.9 ツール制御部 ---------- 21
Fig. 5.1 ２次元径方向表示 ---------- 23
Fig. 5.2 ２次元軸方向表示 ---------- 23
Fig. 5.3 ２次元軸方向分布の水平表示 ---------- 23
Fig. 5.4 ３次元径方向表示 ---------- 25
Fig. 6.1 EVIRA における主要なプログラムの構造 ---------- 28
Fig. 7.1 チャンネルマップ ---------- 31
Fig. 7.2 真空領域の削除後 ---------- 31
Fig. 7.3 六角集合体表示 ---------- 31
Fig. 8.1 EVIRA の起動 ---------- 33
Fig. 8.2 PDS ファイルパスの設定 ---------- 35
Fig. 8.3 設定が完了した状態 ---------- 35
Fig. 8.4 セーブメニュー ---------- 35
Fig. 8.5 シャッフリングパターン ---------- 37
Fig. 8.6 取出し燃料 ---------- 37
Fig. 8.7 取出し燃料判定の修正 ---------- 39
Fig. 8.8 取出し燃料判定のチェックを外した状態 ---------- 39
Fig. 8.9 シャッフリングパターンの修正 ---------- 39
Fig. 8.10 シャッフリングパターンの修正結果 ---------- 41
Fig. 8.11 取出し燃料判定の修正結果 ---------- 41

This is a blank page.

１．序論

革新的水冷却炉(FLWR)[1]の研究開発支援用インターフェースの開発を実施した。本インターフェースにおいては、核計算コード（現時点においてはMOSRA）による解析結果をグラフィカルな表示により簡便にアクセスできる手法の確立を目指し、FLWR の設計研究等において計算結果を効率的に把握することが可能となる。

一般的に、SRAC[2]をはじめ炉心解析コードの多くはテキストファイルによる入出力機能を有している。テキストによる入出力は自由度が高いため多様な対象に対する研究においては便宜的にも妥当であると考えられる。しかし、解析対象がある程度固まっているような状況においては、解析対象の幾何形状に関してはほぼ固定されており、燃料組成等の条件のみが変化する。このような場合、出力ファイルを手作業によって整理することは多くの労力を必要とするため、手法の簡略化は有用である。さらに、出力結果に対する簡便なアクセスのみではなく、それらの値を視覚的に把握することが大変有用である。このような目的により、炉心解析用可視化インターフェースを開発し、その名称を Enhanced Visual Interface for Reactor Analysis (EVIRA)とした。今後、FLWR の研究開発において多いに資するものと期待できる。

上記のような位置づけにおいて行われる内容の具体的な方針としては、以下のようなものである。はじめに、開発環境の選定を行う。現在、多くの炉心解析コードは FORTRAN により記述され UNIX のシェル機能を利用した制御を行うものが多い。しかし、EVIRA は２章で議論するようにマルチプラットホームによる使用が求められるのと同時に、Graphical User Interface (GUI)ベースの開発を目的としている。そのため、開発環境については適したものを選定する必要がある。次に、実際の開発に関しては、上記のような理由で開発環境の発展の度合いによっては他の開発環境への移植等も考える必要が生じるかもしれない。その意味においても、今後の開発に資する為にも、得られた経験そのものを記載し、まとめていくものとする。

２．開発環境

EVIRA の開発環境について述べる。EVIRA の主な目的は解析データの可視化にある。また、直感的な操作を実現する為に GUI ベースの開発を行う。実行環境に関してはマルチプラットホームにおける実行を目的とした。これは、SRAC をはじめとする炉心解析コードのほとんどが UNIX ベースの OS 環境において実行されるのに対し、解析結果のまとめや報告書作成作業は WindowsOS によって行われる場合が大半であるからである。このため、EVIRA に関しては UNIX,Windows 上において問題なく動くことを前提として開発環境の選定を行った。開発環境の詳細については以下に示す。

２．１　GUI ライブラリと実行環境について

GUI ベースのアプリケーションを作成する上での開発環境を選択する条件としては、実現できる機能の多様性が主要な条件となる。GUI ライブラリは高度に蓄積された有用な開発資源をまとめたものである。一部のオブジェクト指向言語によって記述されたライブラリの中には機能の一部をオーバーライドすることにより機能の拡張を行うことができるものもあるが、そのためにはライブラリの内部構造に関する理解が必要となるため、結局はライブラリの提供する基本的な機能の範囲内における開発に限られることが多い。
EVIRA に実装させる機能としては基本的な GUI 機能に加えて２次元及び３次元のグラフィック表示、そして UNIX (Linux) , Windows の OS 環境で実行できることが挙げられる。マルチプラットホームという点においては、バーチャルマシン上で動く JAVA などが先進的である。しかし、JAVA は一般に動作が重く不安定であると言われている。
　ネーティブな環境においてマルチプラットホームを実現するには UNIX における開発環境を使用することが考えられる。現在では多くの UNIX ソフトウェアーが Windows において動作している。その代表が GNU Image Manipulation Program (GIMP)である。GIMP は名称が示すとおり画像処理ソフトであるが、その開発には Application Programming Interface (API)の開発から行われている。この GUI ライブラリ群は The GIMP Toolkit (GTK+)と呼ばれる。GTK+によって開発されたソフトウェアーとしては GIMP のみではなく UNIX の X Window System 上で動作するデスクトップ環境である GNU Network Object Model Environment (GNOME)なども含まれる。よって、現在における UNIX 環境においての GUI プログラミングに関しては GTK+による開発が事実上の標準となりつつある。そこで、本研究においても GTK+ベースの開発を行うものとする。開発環境は以下のとおりである。

- Gtkmm(GTK+の C++ラッパー)
- Cairomm(ベクトル画、ラスター画を出力できるライブラリ)
- Gtkglextmm(OpenGL の Gtkmm ラッパー)

ここで、Gtkmm とは GTK+を C++言語から使用できるようにするためのラッパーである。GTK+自体は C 言語で記述されている。これをオブジェクト指向言語である C++から利用することにより効果的な開発やメンテナンスが可能となる。GTK＋に関してはバージョン１から２へのバージョンアップに際して大きな改良がなされている。このバージョンアップによってソースコードの互換性がなくなるという事態が発生した。つまりバージョン１環境において作成したソースコードをバージョン２で使用する際にはソースコードの修正が必要となったのである。このような自体は今後も起こりうる可能性がある。しかし、C++のラッパーを使用する限りにおいては、そのような低レベルのライブラリの変更はラッパーによって吸収されることが期待され、開発者が意識する必要が発生するような状況はないと期待される。

Cairomm については同様に C 言語ライブラリである Cairo の C++ラッパーであり、Gtkmm から利用が可能であるため、実質的に Gtkmm との区別をすること無く使用できる。Cairo においてはサーフェイスという共通のオブジェクトに対して描画処理を行い、その演算結果を多くのフォーマットへ出力できる。よって、開発者は出力形式を意識することなくコーディングを行うことができる。EVIRA においては、この Cairo の機能を使用することにより、GUI 上の表示だけではなく PING、PostScript、SVG、PDF など多彩な出力を可能としている。このうち、PING はラスター形式であるが、PostScript、SVG、PDF はベクトル形式であり、画質を落とすことなく拡大縮小ができるものである。

Gtkglextmm については Gtkglext の C++ラッパーであり Gtkmm からの利用が可能となっている。Gtkglext は GTK+における OpenGL の実装であり、実際のコーディングはほぼ OpenGL のプログラミングそのものである。OpenGL は SGI により開発された３D グラフィックライブラリであり Windows には標準的に組み込まれている。しかし、OpenGL 自体は商用ライブラリであるため、Unix には非公式の OpenGL である Mesa3D が組み込まれている。しかし、Mesa3D は OpenGL と完全に互換であるため実質的には、OpenGL プログラミングはプラットフォームを選ばない。Windows に限っていえば OpenGL に並んで DirectX というグラフィックライブラリがある。DirectX は Windows プラットフォームに特化した技術であることは言うまでも無く高級言語であることから、作成されたプログラムは開発環境において使用されたバージョン以上のランタイム環境が無ければ作動しない。一方で OprnGL は低級言語であり確実な動作が期待できる。しかし、Cairo のようなファイル出力用のインターフェースを持たないため、本研究においては描画結果を Bitmap フォーマットに出力させる機能を作成し実装した。

現時点における研究では、上記の GUI ライブラリを使用する。しかし、今後有用な開

発環境が普及するようなことがあれば積極的に乗り換えていく予定である。

２．２　プログラミング言語について

前節において GUI ライブラリについては Gtkmm を使用する旨を述べた。Gtkmm については前述のように C++言語からの使用を想定しているため、EVIRA 全体の開発も C++言語によって行うこととした。一般に、炉心解析コードは FORTRAN によって記述されることが多い。勿論、C++と FORTRAN の混合プログラミング（関数ごとにコンパイルを行いオブジェクトファイルのリンクを行う）も可能であるが開発及びメンテナンスに大変な労力を必要とすることから、C++のみの開発とする。炉心解析コードとの接続に関しては後に述べるように PDS ファイルへアクセスすることにより行う。それには FORTRAN のバイナリファイルの構造を知っていれば問題はない。

FORTRAN プログラムに慣れ親しんだプログラマーが Gtkmm およびに C++言語によってプログラムを組む際に戸惑いかねない点としては、Gtkmm に関してはイベントドリブンであること、C++に関してはオブジェクト指向である点が挙げられる。

イベントドリブンに関しては、GUI プログラミング特有の考え方である。FORTRAN を含む従来のプログラミングは、このイベントドリブン型と比較して手続き型に分類される。手続き型プログラミングにおいては記述した命令が逐次実行されていく。また、標準入力の入力待ちなどの他の入力を起因としての処理もあるが、基本的にプログラムは実行時にメモリ上に呼び出され、終了と同時にメモリ上から消去される。それに比べ、イベントドリブン型プログラミングにおいては、プログラムは処理の実行の有無にかかわらず、一度起動したらユーザーに終了を要求されるまでは実行され続ける。この間、プログラムの制御はある種の無限ループ内にある。ユーザーが発生させたイベント（例えば、ボタンを押す）に対応した関数が呼び出され実行される。実行が終了すると再び無限ループ内に制御が戻る。そのイベントに関連付けられた関数自体は手続き型プログラミングの記述と何ら変わりはない。ゆえに、プログラマーは GUI の構成、想定イベント、データ処理用に関するプログラミングを行う必要がある。この構成に関しては、どの GUI 開発環境においても類似しているため、経験自体は共有できる。ただ、イベント起因で処理が実行される際に、現在で行われている炉心解析など計算時間を要するものについて解析を行うと、制御が戻ってくるまでに時間がかかり、その間アプリケーションはハングした状態となる。一部の OS においては例外と判断し、強制終了するか否かを問い合わせてくる結果となる。これに対応するには、マルチスレッドという手法がある。これは同一プログラム内の幾つものプロセスを並列に実行・同期を図るシステムである。重い処理が必要な場合はマルチスレッドの導入も検討すべきである。

オブジェクト指向プログラミングとはプログラムをオブジェクトという部品を組み合わせて作り上げる技術であり、プログラムの再利用やメンテナンス性に定評がある。オブ

ジェクトは具体的にはクラスという単位によって構成される。クラスとは関連の強い変数とその変数への処理関数を一纏めにした単位であり、C++ではそれぞれ、メンバ変数、メンバ関数と呼んでいる。オブジェクト指向の特徴としては大きくポリモフィズム、カプセル化、継承の３つがあり、これらの特徴は、プログラムの再利用やメンテナンス性の向上などに直接つながるものである。オブジェクト指向については大きな部品をブラックボックスとして利用する技術と言っても過言ではないと思われる。ポリモフィズムもその一つである。これは、多態性のことを指す。これは、ある命令に関して受け取る側によって挙動が変わるという特徴のことである。これによりプログラミングに柔軟性が生まれる。次に、カプセル化については、内部のデータや挙動を隠蔽することである。これにより、ユーザーは必要なメンバ関数のみにアクセスすることにより、間違って内部の挙動に干渉することのない安全なプログラムとなる。最後に、クラスの継承であるが、これは、他のクラスを利用する際に必要となる。ある特定の機能を持った処理を行う際に、すでに似たような機能を持つクラスが存在する際には、同様の処理を別途記載することは非効率的な開発となり、メンテナンスする際にも同様な修正を多くの箇所に対して行なうことになりかねない。そのようなときに、継承を使うと既存のクラスの機能に新たな機能を追加して再定義することができる。この継承においては、既存のクラスの機能を修正するオーバーライドや、複数のクラスの機能をまとめて利用する多重継承などがある。また、既存のクラスを利用する際には、この継承の他にインスタンス生成による実装が考えられるが、
場合によって使い分けていく必要がある。

３．炉心解析コードとの接続

３．１　接続方法の考察

最終的にはインターフェースからの炉心解析コードに対する直接入力とプロセスの実行管理、結果の直接的な受け渡しが理想的である。しかし、そのためには炉心解析コード自体にも何らかの修正が必要となることが予想される。このため、制御する場合には、OS のシステムコールを通じての制御にとどめておくべきである。

EVIRA では炉心解析コードの PDS ファイルを直接読み取ることにより解析結果を取得する。PDS ファイルから得られない情報は補助的に炉心解析コードの入力ファイルを読み取ることとする。また、現段階で実質的に接続を想定している炉心解析コードは MOSRA[3)] である。

３．２　PDS ファイルの構造と読み込み

PDS ファイルを扱うには PDS ファイルの構造を理解する必要がある。PDS ファイルとは Partitioned Data Set の略語であり FORTRAN のバイナリ形式で記述される。ここでは、PDS ファイルの扱い方について説明する。SRAC 形式の PDS ファイルでは始めに配列の要素数が記述され、その後に要素の内容が羅列される。整数型、浮動小数点型、文字データ（４文字）ともに１ワード（4bite）で記述されるため、4bite ずつ読み取ればよい。（もし、8bite の倍精度実数型が混在するようだと読み取り長をそのつど変更する必要があるが、現在の PDS ファイルは整数型か浮動小数点型のみである。）

具体的な PDS ファイルの書き出しは以下のように行う。

********** pdswrite.f **********

```
      IMPLICIT REAL(A-H,O-Z)
       DIMENSION N(50),WORK(200),IWORK(200)
       EQUIVALENCE (WORK(1),IWORK(1))

       OPEN(UNIT=1,FILE='TESTPDS',FORM='UNFORMATTED',ACCESS='SEQUENTIAL')
       LENG=3
       N(1)=1
       N(2)=2
       N(3)=3
       DO I=1,LENG
       IWORK(I)=N(I)
       END DO
       WRITE(1) LENG, (WORK(I),I=1,LENG)
```

```
      CLOSE(UNIT=1)
      STOP
      END
```

****

このプログラムは要素数 3 の配列で各要素が 1,2,3 のものを書き出すプログラムである。OPEN 文で UNFORMATTED を指定しフォーマットなしで書き出すことによりバイナリ出力となる。書き出す際には WORK の中身を書き出している。WORK と IWORK は EQUIVALENCE で結合されているため参照番地は同一である。値を受け渡す際には整数の場合は IWORK,実数（浮動小数点型）の場合は WORK によりアクセスすれば型変換のような使い方ができる。同様に PDS ファイルの読み取り用プログラムについては以下のように記述できる。

********** pdsread.f **********

```
      IMPLICIT REAL(A-H,O-Z)
      DIMENSION N(50),WORK(200),IWORK(200)
      EQUIVALENCE (WORK(1),IWORK(1))
      OPEN(UNIT=1,FILE='TESTPDS',FORM='UNFORMATTED',ACCESS='SEQUENTIAL')
      READ(1) LENG, (WORK(I),I=1,LENG)
      WRITE(6,*) LENG, (IWORK(I),I=1,LENG)
      CLOSE(UNIT=1)
      STOP
      END
```

****

このプログラムを実行すると標準出力に以下のように出力される。

**********標準出力**********

3 1 2 3

****

要素数と要素の内容が羅列されている様子がわかる。

しかし、本プログラムは C 言語（C++）主体で開発を行うため FORTRAN のバイナリ形式を理解する必要がある。結果から言えば以下のプログラムで同じ PDS ファイルを読み取ることができる。

********** readpds.cpp **********

```
#include <iostream>
#include <stdio.h>
using namespace std;
int main(int argc, char *argv[])
{
```

```
    string Path="TESTPDS";
            const char *charFile=Path.data();
             FILE *fp;
                 fp = fopen(charFile, "rb");
     int iwork[6];
            for (int i= 0; i<6; i++){
            fread(&iwork[i], sizeof(int), 1,fp);
                cout<<iwork[i]<<"   ";
            }
    cout<<endl;
    return 0;
}
```

****

fopen 文によってファイルを開いているが属性を rb としているので読み取り専用かつバイナリモードで開いていることになる。fread はバイナリを指定バイト分開くことができる関数である。この場合は整数型のサイズを指定しているので 4 バイト分開いていることになる。この出力は以下のようになる。

**********標準出力**********

16 3 1 2 3 16

****

ここで、FORTRAN の出力と比較すると前後に 16 が余計に出力されている様子がわかる。これは FORTRAN のバイナリファイルの形式であり write 文で書き出される総バイト数である。4 バイトが 4 つ（3,1,2,3 の 4 つの整数）書き出されているので 16 バイトという意味である。C 言語で FORTRAN のバイナリを読む際には最初と最後の 4 バイトを無視すればよいことがわかる。

４．インターフェースの構成

本章ではインターフェースの構成について説明する。尚、インターフェース自体開発中であり、開発が完了した後も使い勝手によっては継続的に改良を加えるため、機能や外観に関しては変更する可能性がある。

インターフェースは Fig. 4.1 に示されるように、ひとつのウインドウにより構成されている。上部に結果の画像を、下部に制御用のパネルを配置している。これらの２つの領域の境界は Fig. 4.2 に示されるように上下の移動が可能であり、ユーザーの好みにより自由に設定できる。また、それぞれのパネルがスクロールウィンドウに格納されているため表示位置を自由に決定できる。

上部の画像出力部では径方向断面の２次元表示、３次元表示、軸方向の２次元表示の３タイプをタブにより選択ができる。これらの表示部分は各表示項目（燃焼度、ボイド率など）について共通であり、下部の制御部分において描画の変更が行える。表示された結果についてはメニュー部分にある保存項目から画像ファイルへの出力が可能である。

また、下部に設けられた制御部分については PDS ファイルパスの設定、Rendering Option、Render ボタンの順に並んでおり、ユーザーは PDS ファイルのパスを設定後に Rendering Option を指定して Render ボタンを押すと目的の画像が上部の画像出力部に出力される。制御部分は表示項目ごとに独立しており、必要最低限の PDS ファイルパスを設定すれば表示の実行ができるよう工夫がなされている。勿論、複数の表示項目に対し必要な PDS ファイルが重複している。このため、PDS ファイルパスが該当項目以外についても自動的に連動して設定される。これらの機能によりユーザーは特定の項目について必要とされる複数の PDS ファイルの種類について記憶しておく必要がない。制御部の PDS ファイルパスを機械的に埋めていけば最低限の操作により目的の出力を得ることができる。

現状においては対応させる核計算コードとしては MOSRA を想定しているため、EVIRA ユーザーは MOSRA ユーザーであることが前提となっている。そこで、これらの操作に関しては MOSRA の PDS ファイルに対する操作となる。しかし、上記のような手法を確立することにより、複雑な PDS ファイルの選択に関して特別に深い専門知識を要することなく機械的に行うことができる。

This is a blank page.

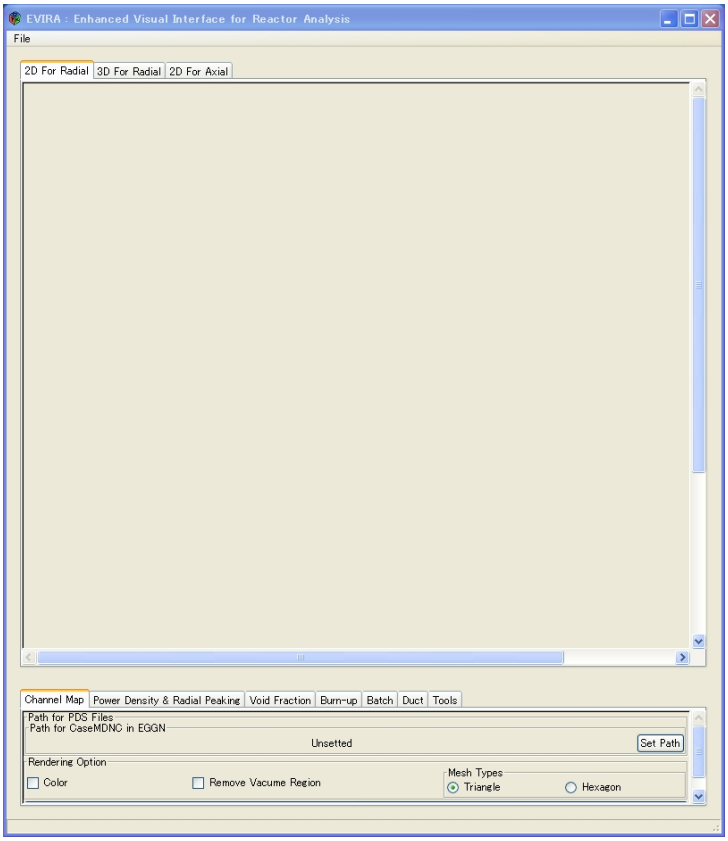
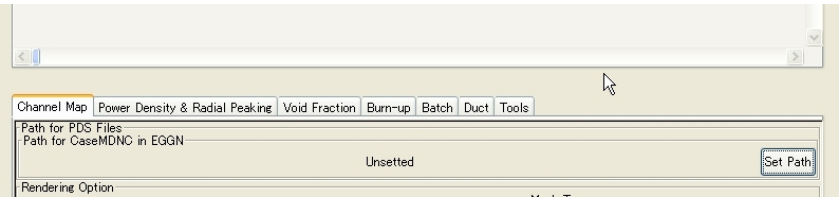
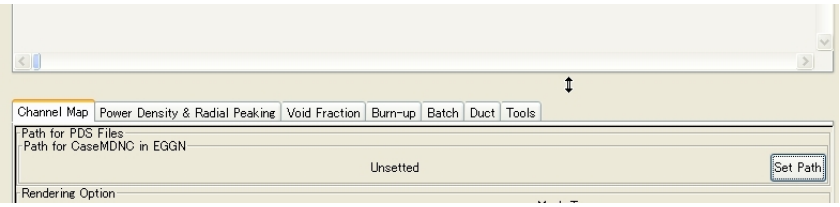


Fig. 4.1 インターフェースの構成



境界付近にマウスカーソルを置いた状態



境界に達して矢印に変わった状態（この状態でドラッグすると移動できる。）

Fig. 4.2 境界位置の移動

This is a blank page.

４．１　チャンネルマップ

現在本コードは、MOSRA-CITATION における Tri-Z 体系のみをサポートしている。この体系における MOSRA-CITATION のチャンネルマップの情報を収納している PDS ファイルは EGGN フォルダ内に格納されている CaseMDNC であり、ここではそのパスのみを指定すればよい。設定オプションとしては「Color」はチャンネル番号に対してレインボーカラーを割り当てるか否かを指定するもの、「Remove Vacume Region」は真空領域の表示の有無、Triangle,Hexagon の選択は、三角格子表示をするか、六角集合体を意識できるような表示にするのかを選択するものである。制御部を Fig. 4.3 に示す。

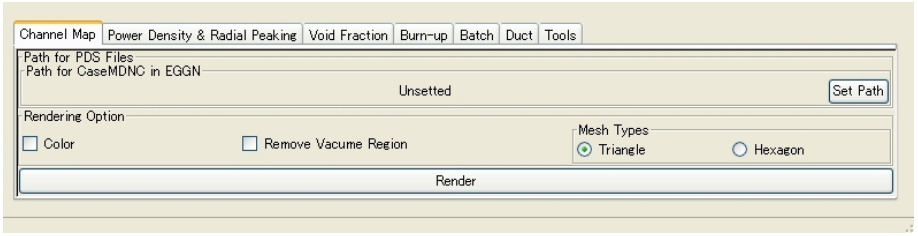


Fig. 4.3　チャンネルマップ制御部

４．２　出力密度・径方向ピーキング

出力分布・径方向ピーキングに関しては Fig. 4.4 に示されるようにチャンネルマップを収納している PDS ファイル CaseMDNC と出力分布を収納している PDS ファイル CasebbNP （EGGN フォルダに格納されている)の指定が必要である。Power Density, Radial Peaking の選択により出力分布と径方向ピーキングの表示が選択できる。出力分布のみに関しては Set Layer No.におけるレイヤー（軸方向の分割領域）の指定によって、任意の高さにおける出力分布を表示できる。

また、2D Axial の項目においては軸方向分布表示についての設定ができる。Average を選択するか否かによって、径方向平均を示すか各チャンネルの値を示すかを選択することができる。平均化は六角集合体のみを対象とし反射体領域は除外される。また、Horizontal Graph を選択することにより２次元軸方向表示のグラフを横向きに変更することもできる。

This is a blank page.

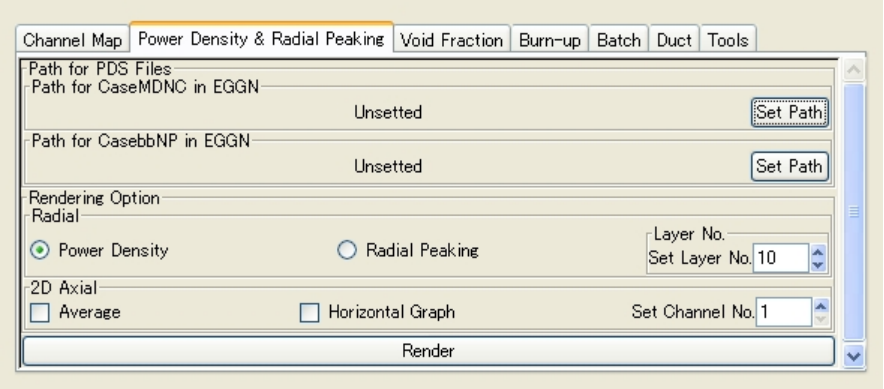


Fig. 4.4　出力密度・径方向ピーキング制御部

４．３　ボイド率

ボイド率に関しては Fig. 4.5 に示されるようにチャンネルマップを収納している PDS ファイル CaseMDNC とボイド率分布を収納している PDS ファイル CasebbCL(EGGC フォルダに格納)の指定が必要である。径方向表示に関しては Each Layer, Layer Averege の選択を行うことによって各レイヤーのボイド率分布か、複数レイヤーを平均化したボイド分布を表示するかを選択できる。複数レイヤーを平均化したボイド率は炉心平均ボイド率を求める際や、それをもとにボイド反応度係数を算出する際に便利である。軸方向表示の選択に関しては出力密度に関するものと同様である。

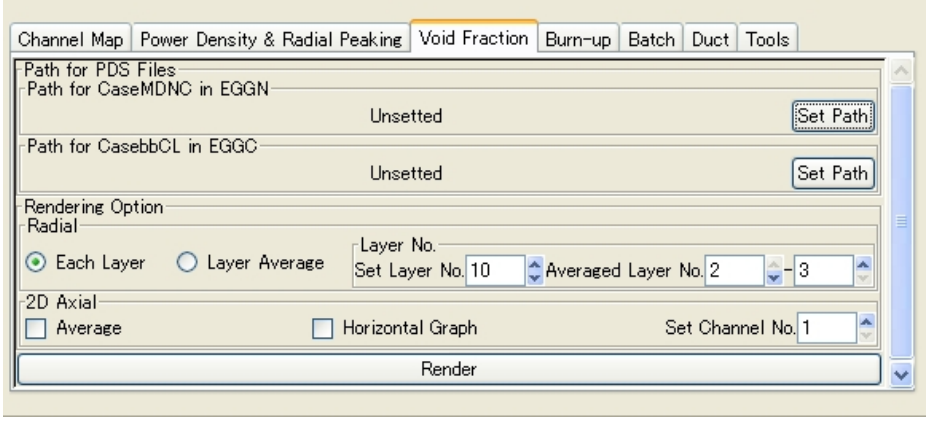


Fig. 4.5　ボイド率制御部

４．４　燃焼度

燃焼度に関しては Fig. 4.6 に示されるようにチャンネルマップを収納している PDS ファイル CaseMDNC と燃焼度分布を収納している PDS ファイル CasebbBP (EGGN フォル

This is a blank page.

ダに格納）の指定が必要である。また、取出燃焼度を求めるためにシャッフリングパターンを記したテキストファイルの指定が必要となる。このテキストファイルは MOSRA の FRAME における入力のシャッフリングについての項目のみに関して記してあればよいが、FRAME の入力をそのまま指定してやってもよい。径方向表示に関しては Each Layer, Layer Averege の選択を行うことによって各レイヤーの燃焼度分布か、複数レイヤーを平均化した燃焼度分布を表示するかを選択できる。複数レイヤーを平均化した燃焼度は炉心平均燃焼度を求めたり、取出燃焼度を求める際に便利である。軸方向表示の選択に関しては径方向平均化に対する項目について、Each Channel, Channel Average, Discharged Average の選択ができ、それぞれ、平均化せずにそれぞれのチャンネルについての表示、径方向全体の平均化（反射体は除く）、取出し燃料のみについて平均化に対応している。

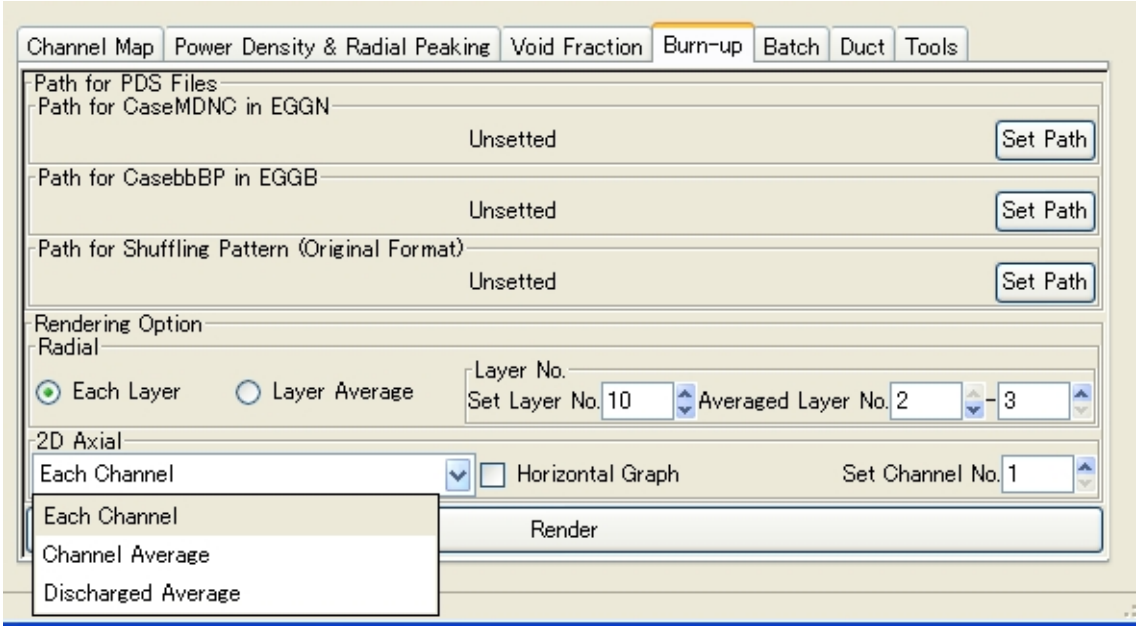


Fig. 4.6　燃焼度制御部

４．５　バッチ

バッチに関しては Fig. 4.7 に示されるようにチャンネルマップを収納している PDS ファイル CaseMDNC と、シャッフリングパターンを記したテキストファイル（MOSRA -FRAME の入力ファイル）の指定が必要となる。選択項目としては Batch No.、Shuffling Pattern の選択ができ、それぞれバッチ分布の表示とシャッフリングパターンを燃料の移動を矢印で表示させたものの選択ができる。シャッフリングパターンに関しては Set Batch No.により移動元のバッチ番号を指定することができる。ここで、最終バッチ数を指定すると取り出し燃料が表示される。また、1/6 境界条件によって分割される半分の集合体に関して修正が必要な場合には Rendering Option の Shuffling Pattern を選択して Render を実

This is a blank page.

行し、2D Radial の表示部において修正作業を行うことになる。この詳細に関しては８章に記述した。

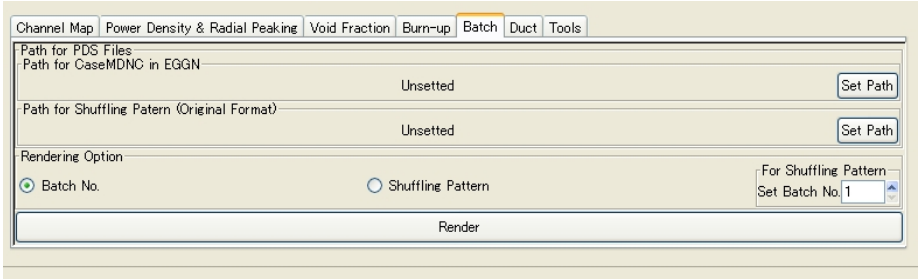


Fig. 4.7　バッチ制御部

４．６　ダクト

この項目については出力の表示ではなく MOSRA-Hydro の入力チェック用である。チャンネルマップを収録している PDS ファイル CaseMDNC と Hydro の入力を指定する必要がある。Fig. 4.8 に示すように、それぞれ、ICHTP,ANUM,AWI についてのチェックができる。詳細は各値の詳細は MOSRA の報告書を参照。

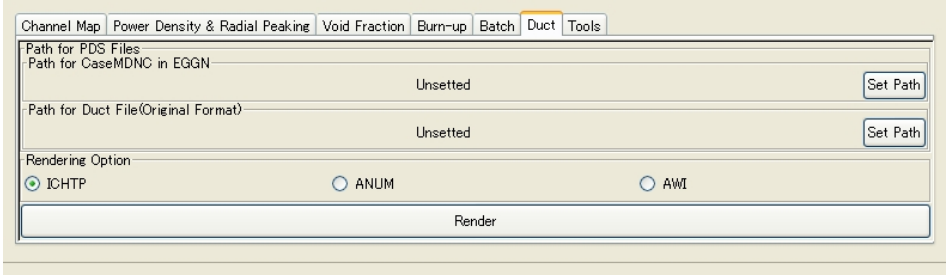


Fig. 4.8　ダクト制御部

４．７　ツール

MOSRA では、FRAME の入力において燃焼度、履歴減速材密度、冷却材温度、減速材密度を設定することにより任意の状態からの解析を行うことができる。しかし、各項目に関して体系にもよるが、およそ 5000 個程度の数値の入力が必要なことや、FORTRAN の制約により１行 72 文字におさえなければならいことから、手作業による入力は非効率的であるといえる。このツールにおいては、各項目についてのテキスト出力が可能となっている。必要な PDS ファイルとしては Fig. 4.9 に示すように、共通に必要なものとしてチャンネルマップを収納している PDS ファイル CaseMDNC、それから、各項目について燃焼度、履歴減速材密度、冷却材温度、減速材密度、それぞれに CasebbBP, CasebbBP,CasebbCL,CasebbCL である。

This is a blank page.

このツール部においては、数値の表示などは目的とせず、あくまでも MOSRA 入力ファイル作成に限った支援ツールである。この項目に関しては、結果の表示の目的とは異なるが、上記の作業の自動化が多くの労力の節約につながると判断したため追加したものである。

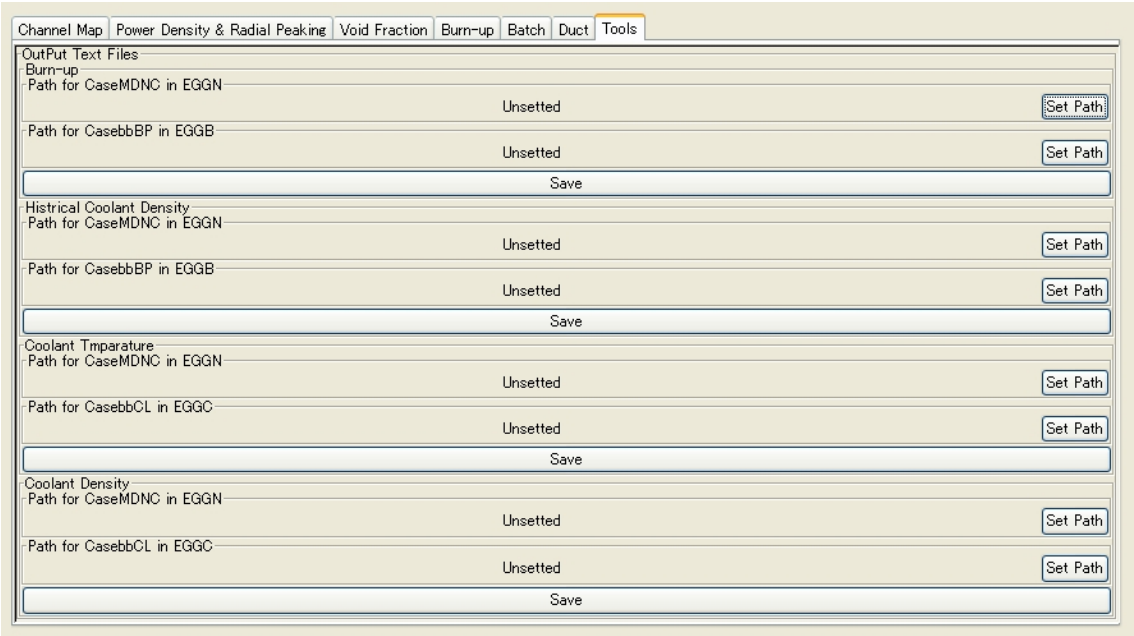


Fig. 4.9　ツール制御部

This is a blank page.

５．表示部と画像ファイルへの出力

５．１　２次元表示

２次元表示は Fig. 5.1, Fig. 5.2 , Fig. 5.3 に示されるように径方向断面と軸方向断面の２つの表示に用いられている。これらの表示に関しては２．１節で述べたようにグラフィックライブラリである Cairomm の特性を利用して柔軟な出力を可能としている。この表示部に示された画像については最上部にあるツールバーを操作することにより多くのファイル形式による出力が可能である。現状では PING、PostScript、SVG、PDF に関して出力をサポートしている。

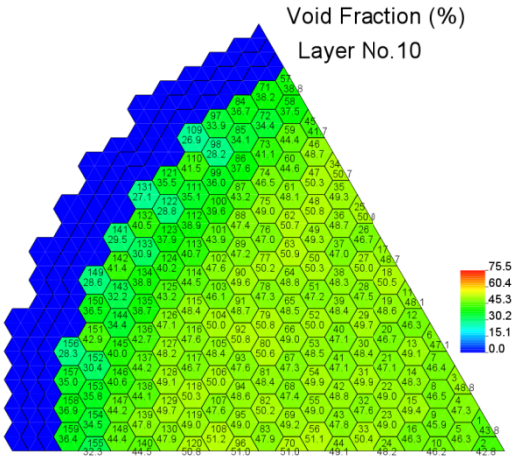


Fig. 5.1　２次元径方向表示

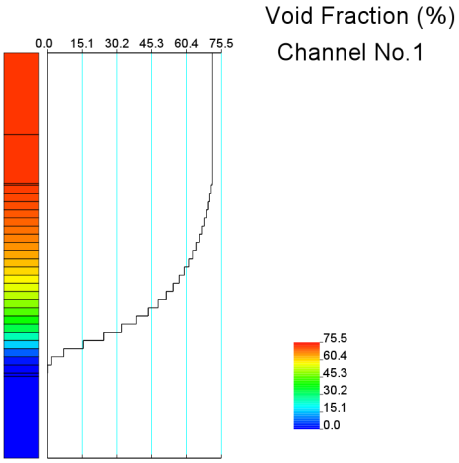


Fig. 5.2　２次元軸方向表示

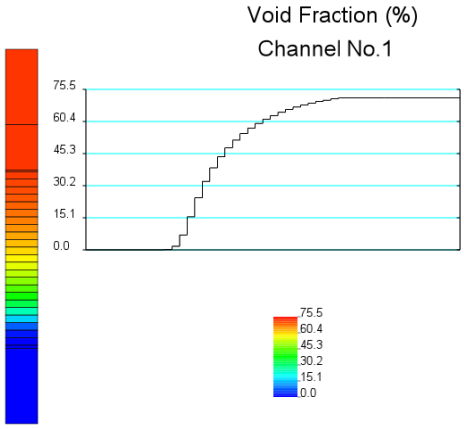


Fig. 5.3　２次元軸方向分布の水平表示

This is a blank page.

５．２　３次元表示

３次元表示は２次元表示と違い視点と対象物の位置関係により描画される画像が変化する。出力結果を Fig. 5.4 に示す。EVIRA ではマウスのドラッグにより視点や対象物の移動・回転を行うことにより直感的に目的の画像に調整できる。左ドラッグによりグラフの回転を行い、右ドラッグによりグラフの平行移動を行う。また、左右同時のドラッグで上部方向にドラッグを行うとズームイン、下方向でズームアウトとなる。また、左のダブルクリックによって数値情報の表示、右のダブルクリックによって高さ方向の軸の表示を選択できる。

出力フォーマットについては Bitmap による出力をサポートしている。

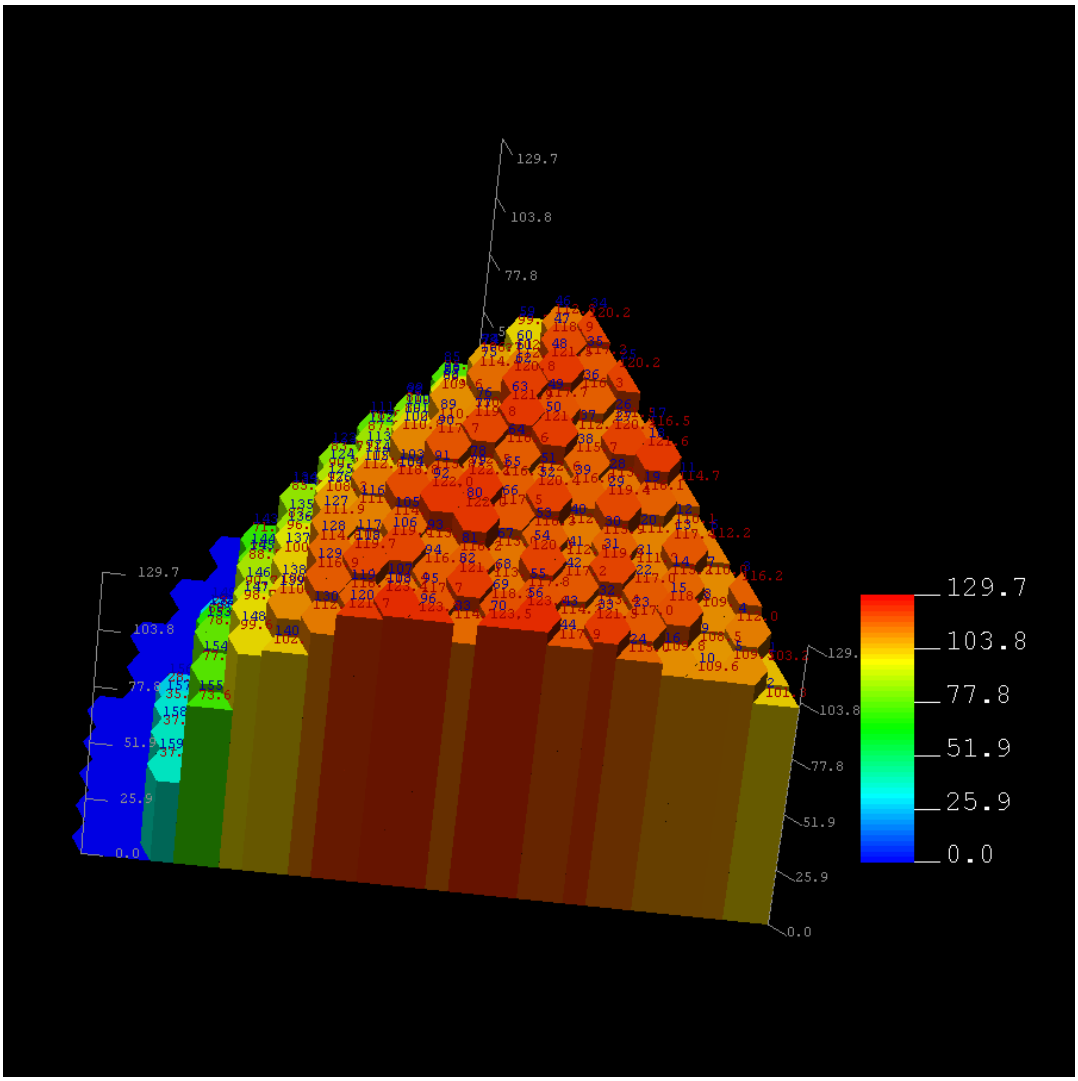


Fig. 5.4　３次元径方向表示

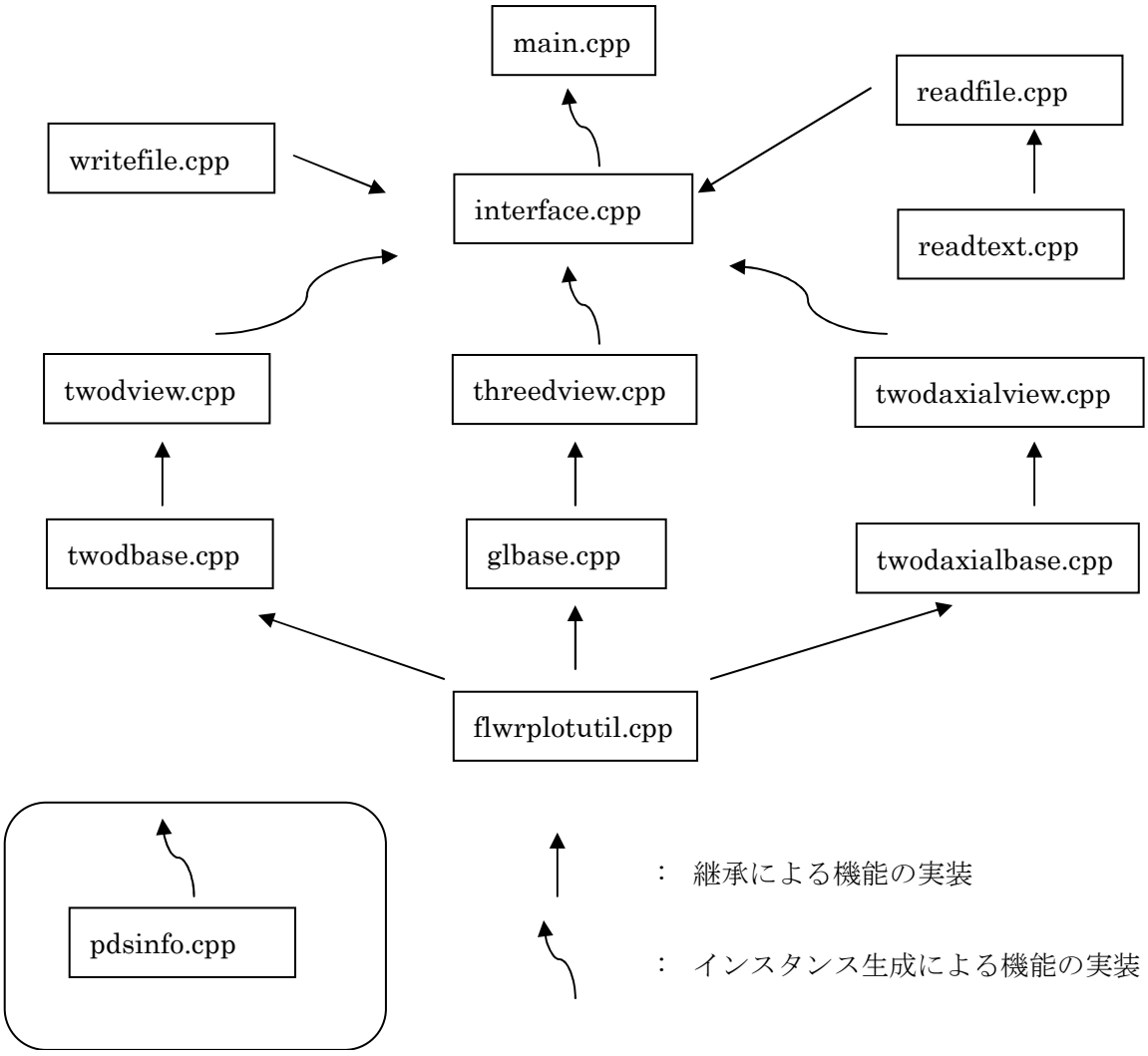
This is a blank page.

６．プログラムの構成

EVIRA については２．２節において述べられているようにオブジェクト指向言語である C++による記述を行っている。また、GUI のプログラムであることから、イベントドリブンという方式を用いており、多くの炉心解析算コードに用いられている手続き型のプログラムとは大きく異なる。

イベントドリブンでは GUI の操作により発生したイベントを起因として目的とする計算が行われる。通常時においてはプログラムの制御は無限ループ内にあり、ボタンなどが押された時に関連付けられた関数が呼び出される。また、オブジェクト指向のプログラミングを行っているため、機能の実装には主に継承とインスタンスの生成によって行われる。EVIRA の主な構造を Fig.6.1 に示す。また、それぞれのプログラムの機能については以下にしめす。

main.cpp：gtkmm, gtkglextmm のメインループの生成。
interface.cpp:インターフェースの配置。
readfile.cpp:ファイルの読み込み。
readtext.cpp:テキストファイルの読み込み。
writefile.cpp:テキストファイルの書き出し。
twodview.cpp:径方向２次元出力の描画。
twodbase.cpp:twodview.cpp の基本機能。
threedview.cpp:径方向３次元出力の描画。
glbase.cpp:threedview.cpp の基本機能。
twodaxialview.cpp:軸方向２次元出力の描画。
twodaxialbase.cpp:twodaxialview.cpp の基本機能。
flwrplotutil.cpp:幾何形状に関する処理やレインボーカラーの作成などの共通機能。



※ pdsinfo に関しては主要なクラスすべてに実装

Fig. 6.1 EVIRA における主要なプログラムの構造

７．EVIRA におけるデータ処理法について

EVIRA の出力結果は MOSRA における出力を演算し瞬時にグラフとして表示される。これにより、ユーザーは実績のある体系を対象にして、PDS ファイルのパスを正しく設定してさえいれば、正確な結果を得ることができる。しかし、一般的にインターフェースの制作時のコンセプトとしては、特定の状況を想定し結果の出力を行うものであり、自由度を限定することにより例外の排除を行うものである。現状の EVIRA においても特定の状況を仮定した演算を行っているため、ユーザーの使い方によっては例外が発生するかもしれない。特に、ユーザーの過失により間違った体系による解析が行われる場合には誤作動を起こしかねない。そこで、ユーザーは例外が発生した時点で、自分の解析手法と EVIRA の設計の両方について確認する必要がある。その意味において、EVIRA が対象としている MOSRA の解析法とその結果の処理法について知っておく必要がある。

EVIRA が想定している MOSRA 解析に関しては FLWR1/6 炉心を解析する為の MOSRA-CITATION Tri-Z ３次元体系に限られる。三角格子の一辺に六角集合体の一辺を割り当て、全集合体９００体＋水反射体領域を含めて描画させている。このため、もしこれ以外の体系において EVIRA を使用すれば、画像表示部に収まらない、もしくは、大幅な余白が生じることとなる。また、六角集合体を想定しているため、間違った入力を行っている際には間違った結果が表示されることがある。

Fig. 7.1 は CITATION のチャンネルマップをそのまま書き出させたものである。このチャンネルマップ自体はユーザーが手作業により MOSRA-CITATION の入力として、およそ 30 行 60 列に及ぶ数値データを入力していく。一般のテキストエディタ上では確認作業は困難を極めるためミスが発生する可能性が高い。この時点で間違いがあれば正しい結果を得ることができない。次に、Fig. 7.2 に真空領域を削除した結果を示す。真空領域のチャンネル番号は MOSRA-CITATION によって一番大きなチャンネル番号として与えられているため、EIVRA においてもこの方法によって真空領域と判断する。次に、Fig. 7.3 に六角集合体表示を示す。この際には EVIRA では２つの方法によって判別描画を行う。一つ目は六角集合体であると判定してからの描画である。これは、チャンネル番号の数を番号ごとに集計して６個であれば一つの集合体であり３個であれば半分の集合体である可能性のある候補として扱う。さらにそれぞれの、チャンネル番号の位置関係によって集合体であると判別された際には描画を行う。その他の反射体領域等に関しては、隣接するセルに同じチャンネル番号が与えられている場合には境界線を設けず、異なるチャンネル番号のセルが隣接する際のみに境界線を描画する。図で示される反射体の３領域はシュラウド及びその内側と外側の領域を示している。これらの図を表示させた時点で不自然な結果が得られた場合には、解析手法に関して見直す必要がある。

上記のような集合体数は炉心平均の値を得る時、半分の集合体の重みづけなどに使用される。また、平均値を求める際には領域の軸方向の長さにも留意する必要がある。これ

に関してはPDSファイルからノード体積の情報を得て補正している。

他に平均値算出等で留意すべき点は、取出し燃料であるか否かの判断である。これは、取出燃焼度算出の際に使用される。この判別においてはMOSRA－FRAMEの入力ファイルを読み取ることにより判別する。現状のEVIRAでは、はじめに行われたシャッフリングのみに注目して行われる。そのため、途中からシャッフリングパターンが変わるような場合には、シャッフリングパターンファイルとしてFRAMEの入力に代えて、下記の手順に対応したテキストファイルを作成することにより柔軟に対処できる。EVIRAによる処理では、はじめに%SHUFFLEのキーワードを検索する。発見し次第データの読み込みを始める。次に、NSHを読み込む。この値はシャッフリングする際の集合体数（チャンネル数）であるが、この値が０のときはシャッフリングが行われないため０ではない値を取得するまでデータを読み込む。NSHを読み込み終えるとISHPOSとISHCHAをひと組ずつ読み込んでデータの取得を終了とする。ここで、ISHPOSは移動後のチャンネルマッピング、ISHCHAは移動前のものである。それぞれNSH個だけの数値テーブルとして与えられる。ここで、ISHPOSとISHCHAを比較することによりバッチ番号や取出し燃料であるか否かの判定をおこなう。はじめに、バッチ番号の決定法について説明する。１バッチ目は初装荷燃料として与えられるため、ISHCHAの負の値のものをチェックする。その負の値のものに対応するISHPOSが１バッチ目の集合体と決定される。それ以降は、同様に１バッチ目から２バッチ目への追跡を行うという操作を繰り返していくとすべての集合体に対しバッチ数を決定することができる。この際にISHPOS, ISHCHAの同じ位置に占めるチャンネルに関しては移動しない集合体と考え０バッチ目とする。ここで、取出し燃料の判定法について説明する。整数バッチ炉心においては、最後のバッチを取出し燃料とみなせば良い。しかし、端数がつくバッチ数（例えば4.5バッチ）炉心においては、最後のバッチに加えてひとつ前のバッチにおいても取出し燃料が存在する。その判定にはISHPOSにある集合体でISHCHAにない集合体を探せば良い。これが取出し燃料とみなしうるものである。

ここまでの手続き自体は問題がないが、ある特殊なケースにおいては取出し燃料に対するさらなる対処が必要となる。それは、1/6境界において分割される半分の集合体の扱い方についてである。これに関しては、結論からいえば半分の集合体２つをある完全な一つの集合体位置へシャッフリングするような場合がそれにあたる。この際にはMOSRAの入力上は一つの半分の集合体に関しては移動先となる完全な一つの集合体部分に対してISHPOS, ISHCHA間における対応関係を与えることができる。しかし、もう一つの半分の集合体に関しては無視されるため、EVIRAでは取出し燃料と誤認してしまう。このため、この半分の集合体に関しては取出し燃料の指定を手動で外す必要がある。これは、MOSRAの入力においては適切な情報が不足しているため、ユーザーによって補足する必要があるためである。

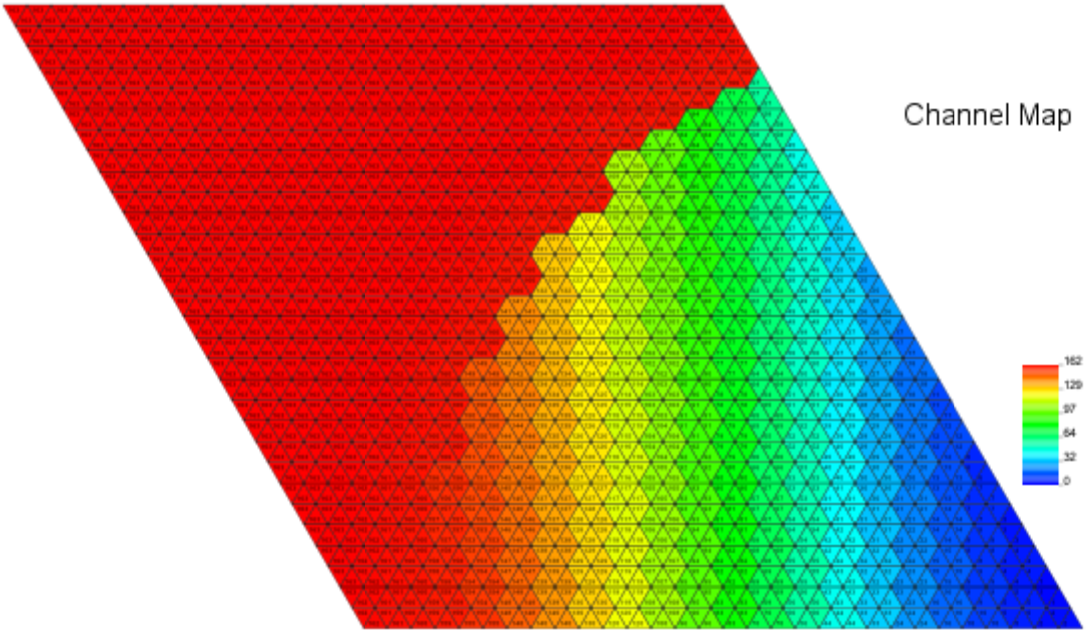


Fig. 7.1 チャンネルマップ

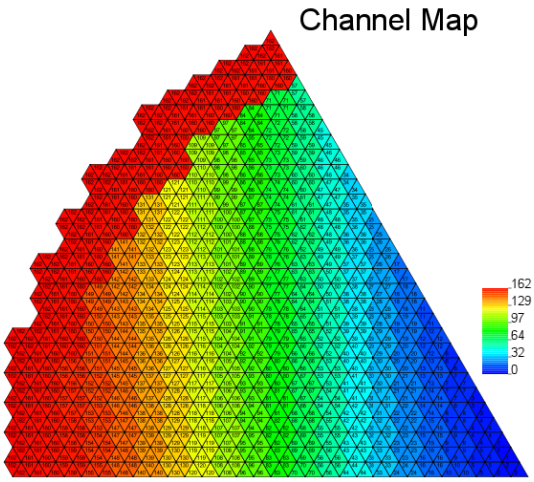


Fig. 7.2 真空領域の削除後

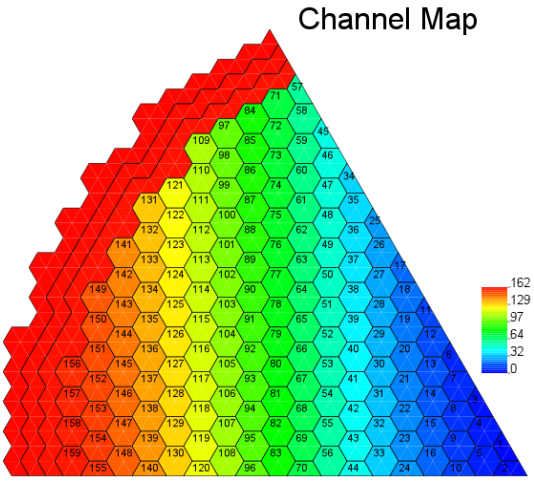


Fig. 7.3 六角集合体表示

This is a blank page.

８．チュートリアル：1/6 境界において分割される集合体の扱い

本章においては、EVIRA の使用法について実例を以って示す。また、操作法についてはどの項目についても類似しているため、前章で問題とされている 1/6 境界において分割された集合体の扱いについて取り扱っていく。ここでは、ある 3.45 バッチ炉心を想定する。この炉心においては、３バッチ目から４バッチ目において３バッチ目の一つの集合体(85 チャンネル)が４バッチ目の２つの半分の集合体（57,155 チャンネル）へ分割されるケースと、３バッチ目の２つの半分の集合体（3,34 チャンネル）が４バッチ目の一つの集合体（109 チャンネル）へまとめられるケースの２ケース含んでいる。前章で述べたようにはじめのケースについては問題とされていないため、それを確認する上で示している。また、２番目のケースが修正の必要なケースなので操作方法を詳細に記述する。

操作１．EVIRA を起動し Batch タブを選択する。

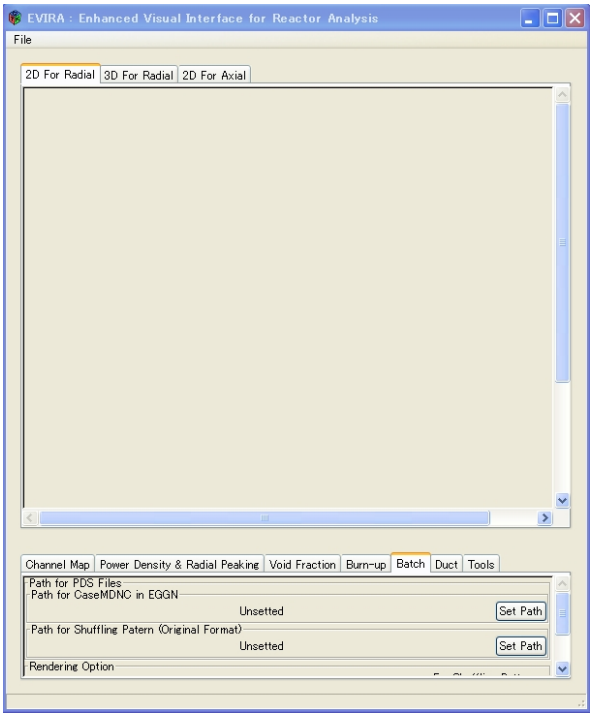


Fig. 8.1 EVIRA の起動

EVIRA の起動を確認したら Fig.8.1 のように Batch タブを選択する。その後は必要なファイルパスを設定し、レンダリングオプションを選びレンダリングを実行する。

This is a blank page.

操作２．設定を行う。

PDS ファイルのパスを指定する為に Set Path ボタンをクリックする。すると Fig. 8.2 のようなファイルチューザーが起動する。この際にどのファイルを開くべきか忘れてしまった時はファイルチューザーのタイトルを見ると「Select CaseMDNC File」などの表示があり迷うことがないようになっている。

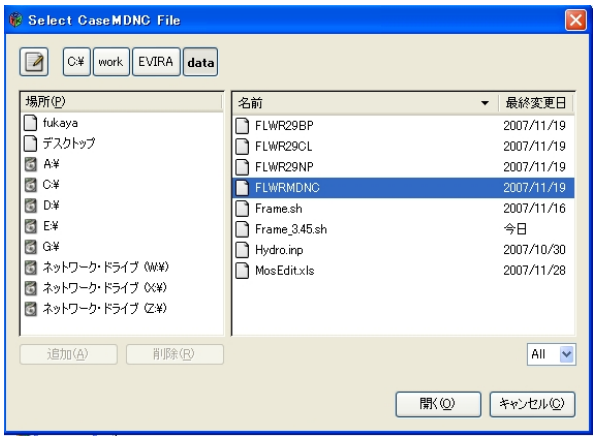


Fig. 8.2　PDS ファイルパスの設定

すべての設定を終えると Fig. 8.3 の状態になる。

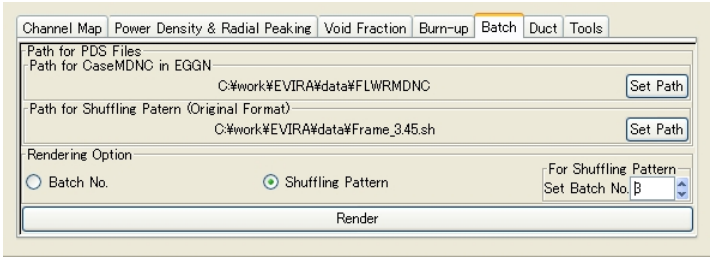


Fig. 8.3　設定が完了した状態

この状態で Render ボタンを押すと描画される。

操作３．結果の保存

Fig.8.4 のように最上部にあるメニューバーより File>Save Image as>Radial 2D を選択し実行するとファイル保存のためのダイアログが開き保存ができる。

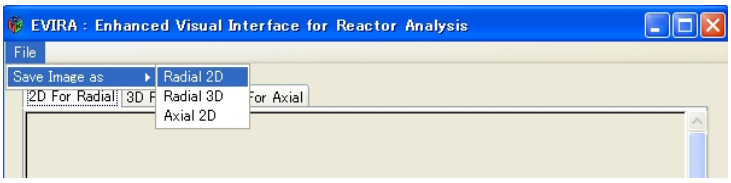


Fig. 8.4 セーブメニュー

This is a blank page.

操作４．結果の検証

画像として保存されたものを Fig. 8.5, Fig. 8.6 に示す。上記の Fig.8.3 の操作部にある Set Batch No.のスピンボタンで値を設定すると、そのバッチ番号から始まるシャッフリングのみを表示する。最大バッチを指定するとシャッフリングパターンではなく取出し燃料のマーキングが表示される。

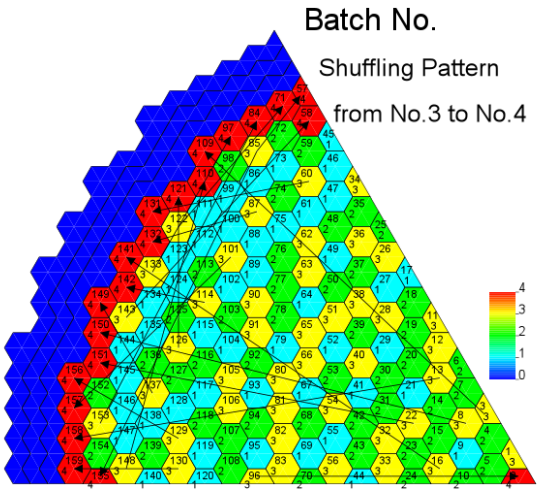


Fig. 8.5 シャッフリングパターン

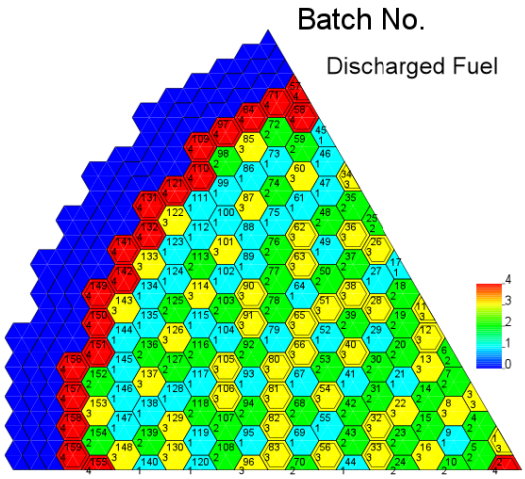


Fig. 8.6 取出し燃料

結果を検証すると、２つの半分の集合体が一つにまとめられる場合に関しては問題なく表示されている。具体的には 85 チャンネルから 57,155 に２つの矢印が伸びていることが確認できる。一方で、問題があるとされる一つの完全な集合体が２つの半分の集合体に分割される場合に関しては表示におかしな点がある。3,34 チャンネルから 109 チャンネルにまとめられるべきであるが、Fig.8.5 のシャッフリングの図では 34 から 109 へ向かう矢印がなく、Fig.8.6 の取出し燃料の図では 34 チャンネルが取出し燃料とされている点である。この理由に関しては前節で述べたとおりであり、手動で訂正しなければならない。特に取出し燃料については取出燃焼度の計算に反映する為適切な処置が必要となる。

This is a blank page.

操作５．結果の修正

はじめに、取出し燃料判定の修正から行う。2D for Radial の表示部が Discharged Fuel 表示になっていることを確認して問題が発生している 34 番チャンネルを左ダブルクリックする。すると、Fig. 8.7 のように取出し燃料判定の修正用のダイアログが表示される。

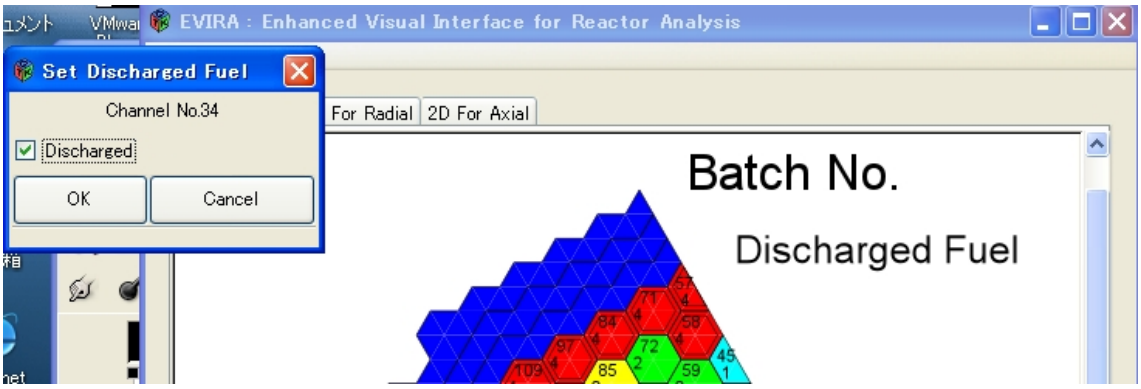


Fig. 8.7 取出し燃料判定の修正

ここでは、Discharged のチェックが入っているので Fig. 8.8 のようにチェックを外し OK ボタンを押す。

Fig. 8.8　取出し燃料判定のチェックを外した状態

これで、取出し燃料判定の修正を終了した。同様にシャッフリングに対する修正を行う。これに関してはシャッフリングパターンの図を使わない場合には特に必要ではない。問題が発生している 34 チャンネルが属している 3 バッチ目の表示になっていることを確認する。問題の 34 チャンネルを左ダブルクリックすると修正用ダイアログが表示される。ここで、移動先となる 109 チャンネルを指定し OK を押す。ラジオボタンに関しては Add で追加を行い Remove で間違って追加した矢印の削除を行う。これで修正が完了した。

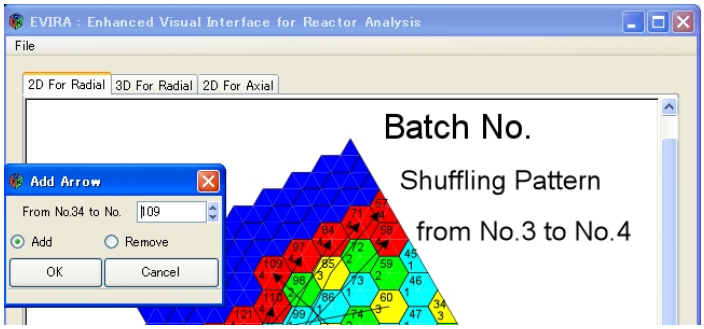
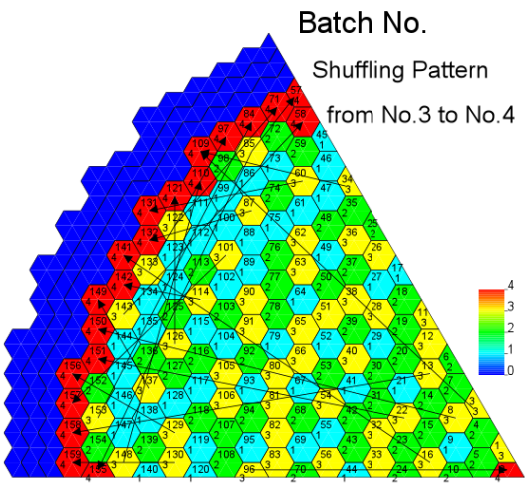


Fig. 8.9 シャッフリングパターンの修正

This is a blank page.

操作６．結果の確認

修正作業の結果を保存したものを以下の Fig. 8.10, Fig. 8.11 に示す。



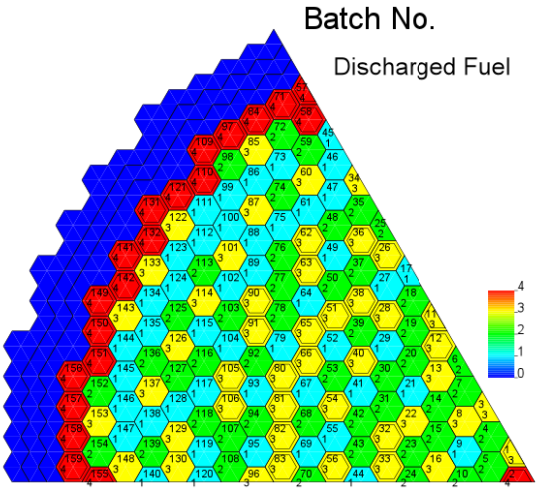


Fig. 8.10 シャッフリングパターンの修正結果　Fig. 8.11　取出し燃料判定の修正結果

結果を見ると 34 チャンネルからシャッフリングパターンの矢印が 109 チャンネルへ伸び、34 チャンネルの取出し燃料判定が消えていることがわかる。これらの作業により意図した結果が得られている様子がわかる。

This is a blank page.

９．結論

FLWR 研究開発支援用インターフェース EVIRA の開発を行った。現在、EVIRA は実質的に MOSRA のインターフェースとしての位置づけである。今回インターフェースの作成を行った経験から汎用的なインターフェースというものの作成は難しく、結局、解析対象に特化したコーディングを余儀なくされることなどがわかった。また、開発環境に関しても、今回は、要求される性能や各開発環境の発展状況を考えて、Gtkmm と C++によるプログラミングを選択したが、今後の発展次第では他の開発環境への乗り換えも積極的に検討すべきであると感じるようになった。しかし、これらの環境については将来的に明るく発展していくものと思われる。

今後、必要があれば機能の拡張・変更などを行っていきたい。

参考文献

1) Uchikawa, S.,Okubo, T., Kugo ,T., et al.：“Conceptual Design of Innovative Water Reactor for Flexible Fuel Cycle (FLWR) and its Recycle Characteristics” ,J. Nucl. Sci. Technol., Vol.44, No.3, p.227, (2007).

2) 奥村啓介、金子邦男、土橋敬一郎 ：“SRAC95: 汎用核計算コードシステム”, JAERI-Data/Code 96-015 (1996).

3) 将来型炉研究グループ・炉物理研究グループ・熱流体研究グループ：“低減速スペクトル炉心の研究 -平成 10～11 年度報告書- （共同研究）”, JAERI-Research 2000-035 (2000).

付録　インストール法

○Windows について

Windows については、開発環境の整備が難しいため、バイナリとランタイム環境の配布のみにとどめる。現時点においては gtk-2.10.7,gtkmm-2.10.11 のランタイムパッケージと gtkglextmm-1.2 の dll ファイル（今回作成）のインストールが必要となるが特記すべき問題はない。

○Linux について

Linux についてはソースからのコンパイルが必要である。これに関しては Fedora7, Fedora8 において実績がある。必要な環境としては gtkmm,gtkglextmm の開発環境であるが古い環境においては gtkmm と cairomm の連携が取れないため新しい環境が必要である。(gtkmm バージョン 2.10.11 未満では動作確認を行っていない)依存性が正しく解消される環境であれば gtkglextmm をインストールするだけで他に必要なパッケージもインストールされる。また、パッケージにはランタイム環境と開発環境のものがあるがインストールすべきものとしては開発環境のほうである。Fedora では開発環境のパッケージには devel がつくことが多い。devel が 付いたものとそうでないものが存在する場合には必ず devel の方を選ぶべきである。環境の整備が完了したらソースファイルを展開して make を実行すれば実行ファイル EVIRA.exe が作成される。Make ファイルは pkg-config を使用してインクルード先やリンクするライブラリのパスを与えている。もし、make がうまくいかない時にはこれらのコマンドを単体で動かしてみるとよい。コマンドを以下に示す。

```
pkg-config gtkmm-2.4 –cflags
pkg-config gtkglextmm-1.2 --cflags
```

これらの結果としてエラーメッセージが出力された場合には、そのパッケージに不備があると考えてよい。これらの環境が整わなければコンパイルはできない。

国際単位系（SI）

表１．SI 基本単位

基本量	SI 基本単位	
	名称	記号
長さ	メートル	m
質量	キログラム	kg
時間	秒	s
電流	アンペア	A
熱力学温度	ケルビン	K
物質量	モル	mol
光度	カンデラ	cd

表２．基本単位を用いて表されるSI組立単位の例

組立量	SI 基本単位	
	名称	記号
面積	平方メートル	m^2
体積	立法メートル	m^3
速さ，速度	メートル毎秒	m/s
加速度	メートル毎秒毎秒	m/s^2
波数	毎メートル	m-1
密度（質量密度）	キログラム毎立法メートル	kg/m^3
質量体積（比体積）	立法メートル毎キログラム	m^3/kg
電流密度	アンペア毎平方メートル	A/m^2
磁界の強さ	アンペア毎メートル	A/m
（物質量の）濃度	モル毎立方メートル	mol/m^3
輝度	カンデラ毎平方メートル	cd/m^2
屈折率	（数の）1	1

表５．SI 接頭語

乗数	接頭語	記号	乗数	接頭語	記号
10^{24}	ヨタ	Y	10^{-1}	デシ	d
10^{21}	ゼタ	Z	10^{-2}	センチ	c
10^{18}	エクサ	E	10^{-3}	ミリ	m
10^{15}	ペタ	P	10^{-6}	マイクロ	µ
10^{12}	テラ	T	10^{-9}	ナノ	n
10^{9}	ギガ	G	10^{-12}	ピコ	p
10^{6}	メガ	M	10^{-15}	フェムト	f
10^{3}	キロ	k	10^{-18}	アト	a
10^{2}	ヘクト	h	10^{-21}	ゼプト	z
10^{1}	デカ	da	10^{-24}	ヨクト	y

表３．固有の名称とその独自の記号で表されるSI組立単位

組立量	SI 組立単位			
	名称	記号	他のSI単位による表し方	SI基本単位による表し方
平面角	ラジアン(a)	rad		$m \cdot m^{-1}=1$ (b)
立体角	ステラジアン(a)	sr(c)		$m^2 \cdot m^{-2}=1$ (b)
周波数	ヘルツ	Hz		s^{-1}
力	ニュートン	N		$m \cdot kg \cdot s^{-2}$
圧力，応力	パスカル	Pa	N/m^2	$m^{-1} \cdot kg \cdot s^{-2}$
エネルギー，仕事，熱量	ジュール	J	N・m	$m^2 \cdot kg \cdot s^{-2}$
工率，放射束	ワット	W	J/s	$m^2 \cdot kg \cdot s^{-3}$
電荷，電気量	クーロン	C		s・A
電位差（電圧），起電力	ボルト	V	W/A	$m^2 \cdot kg \cdot s^{-3} \cdot A^{-1}$
静電容量	ファラド	F	C/V	$m^{-2} \cdot kg^{-1} \cdot s^4 \cdot A^2$
電気抵抗	オーム	Ω	V/A	$m^2 \cdot kg \cdot s^{-3} \cdot A^{-2}$
コンダクタンス	ジーメンス	S	A/V	$m^{-2} \cdot kg^{-1} \cdot s^3 \cdot A^2$
磁束	ウエーバ	Wb	V・s	$m^2 \cdot kg \cdot s^{-2} \cdot A^{-1}$
磁束密度	テスラ	T	Wb/m^2	$kg \cdot s^{-2} \cdot A^{-1}$
インダクタンス	ヘンリー	H	Wb/A	$m^2 \cdot kg \cdot s^{-2} \cdot A^{-2}$
セルシウス温度	セルシウス度(d)	℃		K
光束	ルーメン	lm	cd・sr(c)	$m^2 \cdot m^{-2} \cdot cd=cd$
照度	ルクス	lx	lm/m^2	$m^2 \cdot m^{-4} \cdot cd=m^{-2} \cdot cd$
（放射性核種の）放射能	ベクレル	Bq		s^{-1}
吸収線量，質量エネルギー分与，カーマ	グレイ	Gy	J/kg	$m^2 \cdot s^{-2}$
線量当量，周辺線量当量，方向性線量当量，個人線量当量，組織線量当	シーベルト	Sv	J/kg	$m^2 \cdot s^{-2}$

(a)ラジアン及びステラジアンの使用は、同じ次元であっても異なった性質をもった量を区別するときの組立単位の表し方として利点がある。組立単位を形作るときのいくつかの用例は表４に示されている。
(b)実際には、使用する時には記号rad及びsrが用いられるが、習慣として組立単位としての記号“１”は明示されない。
(c)測光学では、ステラジアンの名称と記号srを単位の表し方の中にそのまま維持している。
(d)この単位は、例としてミリセルシウス度m℃のようにSI接頭語を伴って用いても良い。

表６．国際単位系と併用されるが国際単位系に属さない単位

名称	記号	SI 単位による値
分	min	1 min=60s
時	h	1h =60 min=3600 s
日	d	1 d=24 h=86400 s
度	°	1° =(π/180) rad
分	’	1’ =(1/60)° =(π/10800) rad
秒	”	1” =(1/60)’ =(π/648000) rad
リットル	l、L	1l=1 $dm^3=10^{-3}m^3$
トン	t	1t=10^3 kg
ネーパ	Np	1Np=1
ベル	B	1B=(1/2)ln10(Np)

表７．国際単位系と併用されこれに属さない単位で SI単位で表される数値が実験的に得られるもの

名称	記号	SI 単位であらわされる数値
電子ボルト	eV	1eV=1.60217733(49)×10^{-19}J
統一原子質量単位	u	1u=1.6605402(10)×10^{-27}kg
天文単位	ua	1ua=1.49597870691(30)×10^{11}m

表８．国際単位系に属さないが国際単位系と 併用されるその他の単位

名称	記号	SI 単位であらわされる数値
海里		1海里=1852m
ノット		1ノット=1海里毎時=(1852/3600)m/s
アール	a	1 a=1 $dam^2=10^2m^2$
ヘクタール	ha	1 ha=1 $hm^2=10^4m^2$
バール	bar	1 bar=0.1MPa=100kPa=1000hPa=10^5Pa
オングストローム	Å	1 Å=0.1nm=10^{-10}m
バーン	b	1 b=100$fm^2=10^{-28}m^2$

表４．単位の中に固有の名称とその独自の記号を含むSI組立単位の例

組立量	SI 組立単位		
	名称	記号	SI 基本単位による表し方
粘度	パスカル秒	Pa・s	$m^{-1} \cdot kg \cdot s^{-1}$
力のモーメント	ニュートンメートル	N・m	$m^2 \cdot kg \cdot s^{-2}$
表面張力	ニュートン毎メートル	N/m	$kg \cdot s^{-2}$
角速度	ラジアン毎秒	rad/s	$m \cdot m^{-1} \cdot s^{-1}=s^{-1}$
角加速度	ラジアン毎平方秒	rad/s^2	$m \cdot m^{-1} \cdot s^{-2}=s^{-2}$
熱流密度，放射照度	ワット毎平方メートル	W/m^2	$kg \cdot s^{-3}$
熱容量，エントロピー	ジュール毎ケルビン	J/K	$m^2 \cdot kg \cdot s^{-2} \cdot K^{-1}$
質量熱容量（比熱容量），質量エントロピー	ジュール毎キログラム毎ケルビン	J/(kg・K)	$m^2 \cdot s^{-2} \cdot K^{-1}$
質量エネルギー（比エネルギー）	ジュール毎キログラム	J/kg	$m^2 \cdot s^{-2} \cdot K^{-1}$
熱伝導率	ワット毎メートル毎ケルビン	W/(m・K)	$m \cdot kg \cdot s^{-3} \cdot K^{-1}$
体積エネルギー	ジュール毎立方メートル	J/m^3	$m^{-1} \cdot kg \cdot s^{-2}$
電界の強さ	ボルト毎メートル	V/m	$m \cdot kg \cdot s^{-3} \cdot A^{-1}$
体積電荷	クーロン毎立方メートル	C/m^3	$m^{-3} \cdot s \cdot A$
電気変位	クーロン毎平方メートル	C/m^2	$m^{-2} \cdot s \cdot A$
誘電率	ファラド毎メートル	F/m	$m^{-3} \cdot kg^{-1} \cdot s^4 \cdot A^2$
透磁率	ヘンリー毎メートル	H/m	$m \cdot kg \cdot s^{-2} \cdot A^{-2}$
モルエネルギー	ジュール毎モル	J/mol	$m^2 \cdot kg \cdot s^{-2} \cdot mol^{-1}$
モルエントロピー，モル熱容量	ジュール毎モル毎ケルビン	J/(mol・K)	$m^2 \cdot kg \cdot s^{-2} \cdot K^{-1} \cdot mol^{-1}$
照射線量（X線及びγ線）	クーロン毎キログラム	C/kg	$kg^{-1} \cdot s \cdot A$
吸収線量率	グレイ毎秒	Gy/s	$m^2 \cdot s^{-3}$
放射強度	ワット毎ステラジアン	W/sr	$m^4 \cdot m^{-2} \cdot kg \cdot s^{-3}=m^2 \cdot kg \cdot s^{-3}$
放射輝度	ワット毎平方メートル毎ステラジアン	$W/(m^2 \cdot sr)$	$m^2 \cdot m^{-2} \cdot kg \cdot s^{-3}=kg \cdot s^{-3}$

表９．固有の名称を含むCGS組立単位

名称	記号	SI 単位であらわされる数値
エルグ	erg	1 erg=10^{-7} J
ダイン	dyn	1 dyn=10^{-5}N
ポアズ	P	1 P=1 dyn・s/cm^2=0.1Pa・s
ストークス	St	1 St =1$cm^2/s=10^{-4}m^2/s$
ガウス	G	1 G ≙10^{-4}T
エルステッド	Oe	1 Oe ≙(1000/4π)A/m
マクスウェル	Mx	1 Mx ≙10^{-8}Wb
スチルブ	sb	1 sb =1cd/$cm^2=10^4cd/m^2$
ホト	ph	1 ph=10^4lx
ガル	Gal	1 Gal =1cm/$s^2=10^{-2}m/s^2$

表10．国際単位に属さないその他の単位の例

名称	記号	SI 単位であらわされる数値
キュリー	Ci	1 Ci=3.7×10^{10}Bq
レントゲン	R	1 R = 2.58×10^{-4}C/kg
ラド	rad	1 rad=1cGy=10^{-2}Gy
レム	rem	1 rem=1 cSv=10^{-2}Sv
X線単位		1X unit=1.002×10^{-4}nm
ガンマ	γ	1γ=1 nT=10^{-9}T
ジャンスキー	Jy	1 Jy=10^{-26}W・m^{-2}・Hz^{-1}
フェルミ		1 fermi=1 fm=10^{-15}m
メートル系カラット		1 metric carat = 200 mg = 2×10^{-4}kg
トル	Torr	1 Torr = (101 325/760) Pa
標準大気圧	atm	1 atm = 101 325 Pa
カロリー	cal	
ミクロン	µ	1 µ =1um=10^{-6}m

（第７版，1998年改訂）

この印刷物は再生紙を使用しています