JAEA-Data/Code
2015-016
DOI:10.11484/jaea-data-code-2015-016

JAEA-Data/Code

汎用炉心解析システムMARBLEにおけるORIGEN2コード整備

Implementation of ORIGEN2 Code
for the General-purpose Reactor Analysis Code System, MARBLE

菅原 隆徳　小玉 泰寛　西原 健司　平井 康志

Takanori SUGAWARA, Yasuhiro KODAMA, Kenji NISHIHARA and Yasushi HIRAI

原子力科学研究部門
原子力基礎工学研究センター
分離変換技術開発ディビジョン

Partitioning and Transmutation Technology Division
Nuclear Science and Engineering Center
Sector of Nuclear Science Research

October 2015

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

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。

国立研究開発法人日本原子力研究開発機構 研究連携成果展開部 研究成果管理課
〒319-1195 茨城県那珂郡東海村大字白方 2 番地4
電話 029-282-6387, Fax 029-282-5920, E-mail:ird-support@jaea.go.jp

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

汎用炉心解析システム MARBLE における ORIGEN2 コード整備

日本原子力研究開発機構 原子力科学研究部門
原子力基礎工学研究センター 分離変換技術開発ディビジョン
菅原 隆徳、小玉 泰寛 ＊1、西原 健司、平井 康志 ＊1

(2015 年 8 月 21 日受理)

汎用炉心解析システム MARBLE を用いた燃焼計算においては、核分裂生成物がランプ化して扱われるため、個々の核分裂生成物核種を取り扱うことができない。そのため、加速器駆動核変換システム（ADS）の核特性解析においては、燃料交換時に考慮されるべきレアアース等の残存を考慮することができないという問題があった。これを改善するため、燃焼計算コード ORIGEN2 コードを MARBLE で利用できるよう整備を行った。すなわち MARBLE 用に ORIGEN2 コードのカプセル化を行い、高速炉および ADS の燃焼計算に ORIGEN2 コードを使えるように整備した。これにより燃焼計算における核分裂生成物を核種毎に扱うことが可能となり、燃料交換時のレアアース等の残存を考慮することが可能となった。

原子力科学研究所：〒 319-1195 茨城県那珂郡東海村大字白方 2-4
＊ 1 原子燃料工業株式会社

Implementation of ORIGEN2 Code for the General-purpose Reactor Analysis Code System, MARBLE

Takanori SUGAWARA, Yasuhiro KODAMA* 1, Kenji NISHIHARA and Yasushi HIRAI* 1

Partitioning and Transmutation Technology Division
Nuclear Science and Engineering Center
Sector of Nuclear Science Research
Japan Atomic Energy Agency
Tokai-mura, Naka-gun, Ibaraki-ken

(Received August 21, 2015)

The general-purpose reactor analysis code system, MARBLE, has been used to calculate neutron transport and burn-up calculations for Accelerator-Driven System (ADS). In the burn-up calculation of MARBLE, fission product (FP) nuclides had been treated as lump FP in the past. It meant that MARBLE was unable to treat residual nuclides such as rare-earth ones which would be generated by the fuel exchange of the ADS. To treat residual nuclides, ORIGEN2, which was one of the most famous burn-up calculation codes was implemented to MARBLE. By the implementation of ORIGEN2 code, it was available to treat FP nuclides by each nuclide and to consider the residual nuclides in the ADS burn-up calculation.

Keywords: MARBLE, Reactor Analysis Code System, ORIGEN2, ADS3D

∗ 1 Nuclear Fuel Industries, Ltd.

目 次

Contents

List of Tables

List of Figures

1. 緒言

日本原子力研究開発機構（JAEA）では、マイナーアクチノイド（MA: Minor Actinide）の核変換を目的として加速器駆動核変換システム（ADS: Accelerator-Driven System）の研究開発を行っている[1–3]。ADS は大強度陽子加速器と MA 燃料を主体とした未臨界炉心から構成される。ADS は大量の MA を集中的に核変換することが可能であり、また未臨界状態で運転するため、臨界炉よりも臨界事故に至る危険性が低いという特徴を有する。一方で、燃焼期間中の ADS 炉心出力を一定に保つためには、実効増倍率の低下を補償するために陽子ビーム電流値を上げる必要があるため、加速器と未臨界炉心の境界を成すビーム窓に大きな負荷がかかる。このビーム窓の成立性が、ADS 研究開発の大きな課題の一つとなっている。

ビーム窓の設計条件を緩和し、その成立性を高めることを目指して、燃焼期間中の実効増倍率の低下を抑え、最大陽子ビーム電流値を下げるために、制御棒や可燃性毒物に代表される未臨界度調整機構を導入した概念の検討を進めている[4]。これまでに、制御棒や可燃性毒物を炉心内に非均質に配置した体系の核計算を効率的に行うため、JAEA で開発・整備を行っている汎用炉心解析システム MARBLE[5–7] の機能を利用して、ADS 用三次元炉心解析システム ADS3D を整備した[8]。ADS3D では、三次元の非均質な計算体系を対象として、陽子・中性子の輸送から、燃焼計算、燃料交換までを扱うことが可能である。ADS3D では、燃焼計算には MARBLE の BURNUP モジュールを採用した。BURNUP モジュールでは核分裂生成物はランプ化して扱われるが、ADS の核計算においては、燃料交換時にレアアース等の残存を考慮したい場合がある。この場合、核分裂生成物は核種毎に扱う必要があるため、従来の MARBLE を用いたシステムではこのような解析はできない状況にあった。この問題を解決するため、燃焼計算コード ORIGEN2 [9,10] を MARBLE で利用できるよう整備を行った。

第 2 章では、ORIGEN2 コードのカプセル化について述べる。第 3 章では、MARBLE への組み込み例として、SCHEME[5] および ADS3D システムへの組み込み例を紹介する。第 4 章では、検証計算の結果として、高速炉および ADS の燃焼計算結果を示す。

2. ORIGEN2コードのカプセル化

2.1 カプセル化の方針

2.1.1 基本方針

ORIGEN2 コードをカプセル化するにあたり、基本方針として、MARBLE における外部コードカプセル化の基本的な方法に則って行う。具体的には以下の方針に従う。

- 個々の入出力ファイルは ConventionalCodeFile クラスのサブクラスとしてそれぞれカプセル化する。
- 従来のコード実行用のシェルスクリプトに相当する処理を ConventionalCode および ConventionalCodeRunner クラスのサブクラスとしてカプセル化する。
- 入力ファイル用カプセル化クラスのインスタンス生成を省力化するための InputGenerator を用意する。これは ConventionalCodeInputGenerator クラスのサブクラスとして実装する。

2.1.2 入力ファイルのカプセル化

カプセル化対象となる入力ファイルは、機番 5 のファイルと 3 種類の外部ライブラリ（崩壊データ、断面積データおよび光子収率データ）である。これらのファイルをそれぞれカプセル化する。

機番 5 の入力ファイルについて、基本的な燃焼計算用のサンプルを以下に示す。

```
 -1
 -1
 -1
BAS   BURNUP OF TYPICAL FBR FUEL / LIBRARY: 600MWE-MOX  INNER CORE
LIP     0  0  0
LIB     0  1  2  3 821 822 823  9  0  0  1  62
PHO    101 102 103  10
TIT    BURNUP
INP    -1  1 -1 -1  1  1
MOV    -1  1  0  1.0
HED     1 * CHARGE
BUP
IRP      182.50    72.20      1   2   4   2
IRP      365.00    72.20      2   3   4   0
IRP      429.00     0.00      3   4   4   0
IRP      611.50    72.20      4   5   4   0
IRP      794.00    72.20      5   6   4   0
IRP      858.00     0.00      6   7   4   0
IRP     1040.50    72.20      7   8   4   0
IRP     1223.00    72.20      8   9   4   0
IRP     2684.00     0.00      9  10   4   0
```

```
 BUP
 OPTL    8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
 OPTA    8 8 8 8 5 8 5 8 5 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
 OPTF    8 8 8 8 5 8 5 8 5 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
 OUT    10  1 -1  0
 END
2 922350 2.430E+03 922380 8.076E+05 942380 5.700E+03 942390 1.007E+05
2 942400 4.750E+04 942410 2.280E+04 942420 1.330E+04 0      0.0
0
```

このサンプルでは、最初の 3 行はデータブロック、最後の 3 行は組成ブロックであり、残りはコマンドブロックである。ORIGEN2 コードの入力は、基本的にこの 3 つのブロックから構成されるが、入力ファイルのカプセル化にあたっては、この 3 つのブロックを表すクラスをそれぞれ実装する。さらにコマンドブロックについては、各コマンドを表すクラスをそれぞれ実装する。すなわち機番 5 の入力ファイルをカプセル化したオブジェクトは、3 つのブロックを表すそれぞれのオブジェクトを保持し、さらにコマンドブロックを管理するオブジェクトは各 ORIGEN2 コマンドのオブジェクトを保持する構成となる。

ORIGEN2 には使用可能なコマンドが計 30 個存在するが、今回のカプセル化作業においては、基本的な燃焼計算に必要となる最低限のコマンドとして、このうち 20 個の、LIP ／ LPU ／ LIB ／ PHO ／ RDA ／ TIT ／ BAS ／ HED ／ CUT ／ INP ／ MOV ／ BUP ／ IRP ／ IRF ／ DEC ／ OPTA ／ OPTF ／ OPTL ／ OUT ／ END コマンドを実装する [1]。今回実装しなかったコマンドについては、仮にユーザがそれを利用しようとした場合には例外を発生するよう実装する 。

外部ライブラリについては、崩壊データおよび断面積データについては既存のカプセル化クラスが存在し、これらをそのまま使用する。光子収率データについては、この 2 つカプセル化クラスの実装と同様の方法で、新規に実装する。

2.1.3 出力ファイルのカプセル化

カプセル化対象となるファイルは、機番 6 の出力ファイルである。これについては、組成データを読み込んで MARBLE の Material オブジェクトを生成するように実装する。なお読み込み対象となる組成データは g-atoms 単位で出力されたものとする。

2.1.4 実行処理のカプセル化

今回整備の対象とする ORIGEN2 コードは ORIGEN2.2UPJ[11)]（ORLIBJ32/J33）と ORLIBJ40[10)] の 2 種類であり [2]、またそれぞれ中性子スペクトルの違いにより fast と thermal の 2 種類に分か

[1] 各コマンドの簡単な説明を付録 A にまとめた。

[2] ORLIBJ32/J33 と ORLIBJ40 は、JENDL-3.2/3.3 および JENDL-4.0 に基づく ORIGEN2 用ライブラリであるが、それぞれ ORIGEN2 の実行モジュールを含み、別々に配布されている。それぞれのライブラリを用いたいため、今回は両方の配布物を対象として作業を行った。実行モジュール自体はほぼ同じものであり、ここでは ORLIBJ40 のモジュールをデフォルトとしている。

れているため、計 4 種類の実行モジュールが存在する。MARBLE のカプセル化では、各モジュールのカプセル化クラスにそのモジュールのデフォルトパスを与えているため、異なるモジュールについては異なるカプセル化クラスを用意する必要がある。そこでこれらのモジュールに対応するカプセル化クラスとして、ConventionalCode クラスのサブクラスをそれぞれ実装する 。またコードの実行処理を司るクラスとして ConventionalCodeRunner クラスのサブクラスを実装する。全てのモジュールでコードの実行方法は同一であるため、このクラスについては一つのみで十分である。

2.1.5 InputGenerator の作成

MARBLE におけるカプセル化手法に則って、機番 5 の入力ファイルのカプセル化オブジェクトの生成を補助するための InputGenerator を作成する。

ORIGEN2 コードの場合、各コマンドを自由に組み合わせることで多様かつ複雑な入力を作成することが可能であるが、これらのコマンドの組み合わせをユーザが逐一指定するのは面倒である。一方、高速炉や ADS の燃焼計算での利用を考えると、最低限の単純な燃焼および冷却計算用の入力を作成できれば十分と考えられる。そこで基本的な燃焼計算および冷却計算用の入力を作成する InputGenerator をそれぞれ実装する。ユーザはこれらの InputGenerator に、各燃焼ステップにおける燃焼期間や出力等の必要最低限度の追加情報を与えることで実行可能な ORIGEN2 の入力が生成されるようにする。

InputGenerator の構成としては、SLAROM-UF コードのカプセル化の構成と同様に、上で述べた基本的な燃焼／冷却計算の入力作成用の InputGenerator に加えて、各コマンドのインスタンスを生成するための InputGenerator を実装する。すなわち前者の InputGenerator が後者の各コマンド用 InputGenerator を利用する形をとる。

2.2 実装

機番 5 の入力ファイルのカプセル化のために実装したクラスの一覧を **Table 2.2.1** に示す。機番 5 の入力ファイル全体は OrigenFort5 クラスとしてカプセル化し、これがデータブロック／コマンドブロック／組成ブロックをカプセル化したそれぞれのクラスを管理する。各コマンドは OrigenFort5CommandBase クラスのサブクラスとして表され、今回サポートするそれぞれのコマンドについて実装した。データブロックおよび組成ブロックは、それぞれ OrigenFort5FractionalRecovery および OrigenFort5MaterialSet クラスとして実装した。外部ライブラリ用のカプセル化クラスの一覧を **Table 2.2.2** に示す。このうち新規に実装したのは OrigenPhotonLibrary クラスであり、他は既存のものを使用した。

出力ファイルのカプセル化クラスとして OrigenFort6 クラスを実装した。なお機番 6 のファイルに出力された組成データを読み込むにあたり、「SF250」という核種が出力されている。これは自発核分裂の結果生じる FP 核種を擬似的に扱っているものである。カプセル化した ORIGEN2

コードを利用する際にはこの出力値を取り込むため、MARBLE において核種の情報を定義している NuclideProperty クラスの `add_user_element` メソッドを用いてユーザー定義の核種を追加する必要がある。

コードの実行処理に関わるカプセル化クラスの一覧を **Table 2.2.3** に示す。ConventionalCode クラスのサブクラスとして OrigenBase クラスを実装し、このサブクラスとして各実行モジュールのカプセル化クラスを実装した。また ConventionalCodeRunner クラスのサブクラスとして、OrigenRunner クラスを実装した。

機番 5 の入力作成用の InputGenerator クラスの一覧を **Table 2.2.4** に示す。ConventionalCodeInputGenerator クラスのサブクラスとして OrigenFort5GeneratorBase クラスを実装した。さらにそのサブクラスとして基本的な燃焼計算用入力作成のための OrigenFort5BurnupCommandGenerator クラスを、また冷却計算用入力作成のための OrigenFort5DecayCommandGenerator クラスを実装した 。前者が生成するコマンドとその出力順は下記のとおりである。

- BAS
- LIP
- LIB
- PHO
- TIT
- INP
- HED
- BUP
- IRP もしくは IRF※（燃焼ステップ数分だけ生成される）
- BUP
- OPTL
- OPTA
- OPTF
- OUT

※ OrigenFort5DecayCommandGenerator の場合、DEC コマンド（ステップ数分だけ生成）に変更される。

OrigenFort5BurnupCommandGenerator および OrigenFort5DecayCommandGenerator クラスでは、基本的には出力するコマンドの種類と順序を定義しているだけであり、仮により複雑な計算を行うための Generator が必要となった場合には、その計算で必要となるコマンドを定義した新たな Generator クラスを作成すれば良い。

これらの Generator クラスとは別に、**Table 2.2.1** に挙げられている各コマンド用クラス（OrigenFort5CommandLip 等）のインスタンスを生成するための ConventionalCodeFile クラスを実装した。これらのクラスにおいて、各コマンドの入力パラメータに与えられるデフォルト値等が設定される。前述の OrigenFort5BurnupCommandGenerator 等は、これらを用いて各種のコマンド用オブジェクトを生成している。

Table 2.2.1　機番 **5** の入力ファイルのカプセル化クラス一覧

クラス名	内容
OrigenFort5	機番 5 の入力ファイルのカプセル化クラス
OrigenFort5FractionalRecovery	データブロックのカプセル化クラス
OrigenFort5CommandBase	コマンドブロックの各コマンド用の基底クラス
OrigenFort5CommandLip	「LIP」コマンド用クラス
OrigenFort5CommandLpu	「LPU」コマンド用クラス
OrigenFort5CommandLib	「LIB」コマンド用クラス
OrigenFort5CommandPho	「PHO」コマンド用クラス
OrigenFort5CommandRda	「RDA」コマンド用クラス
OrigenFort5CommandTit	「TIT」コマンド用クラス
OrigenFort5CommandHed	「HED」コマンド用クラス
OrigenFort5CommandCut	「CUT」コマンド用クラス
OrigenFort5CommandInp	「INP」コマンド用クラス
OrigenFort5CommandMov	「MOV」コマンド用クラス
OrigenFort5CommandBup	「BUP」コマンド用クラス
OrigenFort5CommandIrp	「IRP」コマンド用クラス
OrigenFort5CommandIrf	「IRF」コマンド用クラス
OrigenFort5CommandDec	「DEC」コマンド用クラス
OrigenFort5CommandOpt	「OPTA」「OPTF」「OPTL」コマンド用クラス
OrigenFort5CommandOut	「OUT」コマンド用クラス
OrigenFort5MaterialSet	組成ブロックのカプセル化クラス

Table 2.2.2　外部ライブラリのカプセル化クラス一覧

クラス名	内容
OrigenLibrary	外部ライブラリのカプセル化基底クラス
OrigenDecayLibrary	崩壊データのカプセル化クラス
OrigenCrossSectionLibrary	断面積データのカプセル化クラス
OrigenPhotonLibrary	光子収率データのカプセル化クラス

Table 2.2.3　外部モジュールのカプセル化クラス一覧

クラス名	内容
OrigenBase	各モジュール用 ConventionalCode クラスの基底クラス
OrigenOrigen22upjFast	ORIGEN2.2UPJ の高速群モジュール用クラス
OrigenOrigen22upjThermal	ORIGEN2.2UPJ の熱群モジュール用クラス
OrigenOrlibj40Fast	ORLIBJ40 の高速群モジュール用クラス
OrigenOrlibj40Thermal	ORLIBJ40 の熱群モジュール用クラス
OrigenRunner	ORIGEN コード用の ConventionalCodeRunner クラス

Table 2.2.4　InputGenerator クラス一覧

クラス名	内容
OrigenFort5GeneratorBase	OrigenFort5 用 InputGenerator の基底クラス
OrigenFort5BurnupCommandGenerator	基本的な燃焼計算用の InputGenerator クラス
OrigenFort5DecayCommandGenerator	基本的な冷却計算用の InputGenerator クラス
OrigenFort5CommandLipInputGenerator	「LIP」コマンド用 InputGenerator クラス
OrigenFort5CommandLpuInputGenerator	「LPU」コマンド用 InputGenerator クラス
OrigenFort5CommandLibInputGenerator	「LIB」コマンド用 InputGenerator クラス
OrigenFort5CommandPhoInputGenerator	「PHO」コマンド用 InputGenerator クラス
OrigenFort5CommandRdaInputGenerator	「RDA」コマンド用 InputGenerator クラス
OrigenFort5CommandTitInputGenerator	「TIT」コマンド用 InputGenerator クラス
OrigenFort5CommandBasInputGenerator	「BAS」コマンド用 InputGenerator クラス
OrigenFort5CommandHedInputGenerator	「HED」コマンド用 InputGenerator クラス
OrigenFort5CommandCutInputGenerator	「CUT」コマンド用 InputGenerator クラス
OrigenFort5CommandInpInputGenerator	「INP」コマンド用 InputGenerator クラス
OrigenFort5CommandMovInputGenerator	「MOV」コマンド用 InputGenerator クラス
OrigenFort5CommandBupInputGenerator	「BUP」コマンド用 InputGenerator クラス
OrigenFort5CommandIrpInputGenerator	「IRP」コマンド用 InputGenerator クラス
OrigenFort5CommandIrfInputGenerator	「IRF」コマンド用 InputGenerator クラス
OrigenFort5CommandIrfInputGenerator	「IRF」コマンド用 InputGenerator クラス
OrigenFort5CommandDecInputGenerator	「DEC」コマンド用 InputGenerator クラス
OrigenFort5CommandOptaInputGenerator	「OPTL」コマンド用 InputGenerator クラス
OrigenFort5CommandOptlInputGenerator	「OPTA」コマンド用 InputGenerator クラス
OrigenFort5CommandOptfInputGenerator	「OPTF」コマンド用 InputGenerator クラス
OrigenFort5CommandOutInputGenerator	「OUT」コマンド用 InputGenerator クラス

2.3 使用例

2.3.1 燃焼計算用 OrigenFort5 オブジェクトの生成（その１）

基本的な燃焼計算用の OrigenFort5 オブジェクトを生成する処理の例を示す。

```
# コマンドブロック
generator = OrigenFort5BurnupCommandGenerator(num_bups=2, use_irf=False)
generator.set_options(bas1_title="BURNUP FBR CORE")
generator.set_options(lib1_nlib2=1)
generator.set_options(lib1_nlib3=2)
generator.set_options(lib1_nlib4=3)
generator.set_options(lib1_nlib5=821)
generator.set_options(lib1_nlib6=822)
generator.set_options(lib1_nlib7=823)
generator.set_options(pho1_npho1=101)
generator.set_options(pho1_npho2=102)
generator.set_options(pho1_npho3=103)
generator.set_options(hed1_nhed=1)
generator.set_options(hed1_heading="* CHARGE")
generator.set_options(irp1_rirp1=182.50)
generator.set_options(irp1_rirp2=72.20)
generator.set_options(irp1_nirp1=1)
generator.set_options(irp1_nirp2=2)
generator.set_options(irp1_nirp3=4)
generator.set_options(irp1_nirp4=2)
generator.set_options(irp2_rirp1=365.0)
generator.set_options(irp2_rirp2=72.20)
generator.set_options(irp2_nirp1=2)
generator.set_options(irp2_nirp2=3)
generator.set_options(irp2_nirp3=4)
generator.set_options(irp2_nirp4=0)
generator.set_options(out1_nout1=3)
fort5 = generator.generate()

# データブロック
fr1 = OrigenFort5FractionalRecovery(num_column=3)
fr1.add([92, 1, 1.0])
fr1.add([94, 1, 1.0])

fr2 = OrigenFort5FractionalRecovery(num_column=3)
fr3 = OrigenFort5FractionalRecovery(num_column=2)

fort5.set_individual_element_fractional_recovery(fr1)
fort5.set_element_group_fractional_recovery(fr2)
fort5.set_element_group_membership(fr3)

# 組成ブロック
fort5_matset = OrigenFort5MaterialSet()
fort5_matset.add(mat)

fort5.set_materialset(fort5_matset)
```

またこの処理により生成される ORIGEN2 用の入力ファイルを以下に示す。

```
 92  1 1.000
 94  1 1.000
 -1
 -1
 -1
BAS     BURNUP SAMPLE
LIP        0   0   0
LIB        0   1   2   3 821 822 823   9   0   0   1   0
PHO      101 102 103  10
TIT     AUTOMATICALLY GENERATED ORIGEN INPUT
INP        1   2  -1  -1   1   1
HED        1  * CHARGE
BUP
IRP         182.50     72.20   1   2   4   2
IRP         365.00     72.20   2   3   4   0
BUP
OPTL    8 8 5 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
OPTA    8 8 5 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
OPTF    8 8 5 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
OUT        3   1   1  -1
END
 2 922350 2.43000E+03 922380 8.07600E+05 942380 5.70000E+03 942390 1.00700E+05
 2 942400 4.75000E+04 942410 2.28000E+04 942420 1.33000E+04      0         0.0
 0
```

OrigenFort5 を生成するにあたり、まず OrigenFort5BurnupCommandGenerator オブジェクトを生成する。この際、コンストラクタの引数として「`num_bups=2`」および「`use_irf=False`」を与えている。前者は燃焼計算のステップ数であり、後者の指定とあわせて、ここで与えたステップ数だけ IRP コマンドが発行される（デフォルトでは IRF コマンドが発行される）。次に生成した InputGenerator に対して、`set_options` メソッドで各コマンドのパラメータの値を設定する。例えば引数で「`irp1_rirp1=182.50`」と与えた場合、これは「1 番目の IRP コマンドの RIRP1 パラメータに 182.50 を与える」ことを示している。2 番目の IRP コマンドを対象とする場合は、「`irp2_rirp1`」などとすれば良い。

InputGenerator に必要となるパラメータを設定した後、generate メソッドを発行すると、返り値として OrigenFort5 オブジェクトが得られる。なお値が未設定のパラメータについては、各コマンドの InputGenerator クラスにて定義されたデフォルト値が設定される。デフォルト値が存在せず、値を設定することが必須のパラメータが存在する場合は、generate メソッド発行時に例外が発生する。

次に、上記で InputGenerator から得られた OrigenFort5 オブジェクトに対して、データブロックおよび組成ブロックを設定する。データブロックについては、“A. 元素別回収率の変更指定”、“B. 元素グループ別回収率の変更指定”、“C. 元素グループ内に含める元素の追加指定”それぞれについて、OrigenFort5FractionalRecovery オブジェクトを生成し、それらを OrigenFort5 オ

ブジェクトに設定する。なおオブジェクトを生成する際に、A および B については、コンストラクタの引数に「num_column=3」を与え、C については「num_column=2」を与える必要がある。例えば、A. 元素別回収率の変更指定として「92 1 1.0」を与える場合、生成したオブジェクトに対して add メソッドを用いてデータを設定する。その後、OrigenFort5 オブジェクトに対して set_individual_element_fractional_recovery メソッド等を用いてこのオブジェクトを設定する。

最後に、OrigenFort5 オブジェクトに対して組成ブロックを設定する。まず組成ブロック用に OrigenFort5MaterialSet オブジェクトを生成し、add メソッドを用いて MARBLE の Material オブジェクトを設定する。その後、OrigenFort5 オブジェクトに対して set_materialset メソッドを用いて上記のオブジェクトを設定する。

2.3.2 燃焼計算用 OrigenFort5 オブジェクトの生成（その２）

既存の ORIGEN2 の入力ファイルを読み込んで、OrigenFort5 オブジェクトを生成する例を以下に示す。なお、サポートしていない ORIGEN2 のコマンドが使用された入力ファイルを読み込んだ場合、例外が発生する。

```
fin = open(origen_file_name)    # origen_file_name: ORIGEN 入力ファイルのパス

fort5 = OrigenFort5()
fort5.read_file(fin)
```

2.3.3 ORIGEN2 コードの実行例

カプセル化クラスを用いた ORIGEN2 コードの実行例を以下に示す。

```
# 外部ライブラリ
decaylib = OrigenDecayLibrary(
    os.path.join(CODE_DIR、 “ORLIBJ40”, “lib”, “DECAYJ40.LIB”))

xslib = OrigenCrossSectionLibrary(
    os.path.join(CODE_DIR、 “ORLIBJ40”, “lib”, “600MMXICJ40.LIB”))

photonlib = OrigenPhotonLibrary(
    os.path.join(CODE_DIR、 “ORLIBJ40”, “lib”, “JNNOBREM.LIB”))


# 計算実行
origen = OrigenOrlibj40Fast()
origen.set_fort5(fort5)
origen.set_decay_library(decaylib)
origen.set_cross_section_library(xslib)
origen.set_photon_library(photonlib)
origen.run()
```

```
# 計算結果取得
fort6 = origen.outlist()
mat = fort6.material()
```

使用するモジュールに対応するカプセル化クラス（この例では OrigenOrlibj40Fast）のインスタンスを生成し、OrigenFort5 オブジェクトおよび各外部ライブラリのオブジェクトを設定する。その後、run メソッドを実行すると、計算実行環境に ORIGEN2 の入力ファイル等が生成され、コードが実行される。

コードの実行が正常終了した後、outlist メソッドを呼び出すと計算結果をカプセル化した OrigenFort6 オブジェクトが得られる。このオブジェクトの material メソッドにより、計算結果が MARBLE の Material オブジェクトとして得られる。

3. MARBLEへの組み込み

3.1 SCHEMEへの組み込み

3.1.1 実装の方針

MARBLEでは、高速炉核特性解析システムとしてSCHEMEが整備されている。これはSLAROM-UF[12]により格子計算を行い、CITAITON[13]やPARTISN[14,15]などにより中性子輸送計算を行い、核特性を解析するシステムである。燃焼計算にはBURNUPというモジュールが用いられていたが、この燃焼計算にORIGEN2を用いるため、SCHEMEにORIGEN実行用の関数を実装する。ここでは他のSCHEME関数とは完全に独立のものとし、marble.scheme.origenモジュールに新規の関数群を追加する。

まず燃焼計算および冷却計算実行用の関数をそれぞれ用意する。これらの関数の内部では、前章で述べたカプセル化クラスを用いて、基本的な燃焼計算および冷却計算用のORIGEN2の入力を作成し、計算を実行する。関数の戻り値として燃焼もしくは冷却後の組成を返す。

その他に、マテリアル領域毎の出力を算出する関数、MARBLEのDecayConstantSetの値でORIGENの崩壊データライブラリの値を更新するための関数、およびMARBLEのMicroscopicCrossSectionと、中性子束からORIGEN2の断面積データライブラリの値を更新する関数を用意する。

3.1.2 実装

実装したSCHEME関数の一覧を**Table 3.1.1**に示す。また、燃焼計算用の`origen_burnup`関数のインターフェイスを以下に示す。

【引数】
mat：Materialオブジェクト
decaylib：OrigenDecayLibraryオブジェクト
xslib：OrigenCrossSectionLibraryオブジェクト
photonlib：OrigenPhotonLibraryオブジェクト
version：モジュールのバージョン名（origen22upjもしくはorlibj40．デフォルトはorlibj40）
neutron_spectrum：スペクトルの区分（fastもしくはthermal．デフォルトはfast）
individual_element_fractional_recoveries：元素別の回収率変更指定（リスト。省略可）
element_group_fractional_recoveries：元素グループ別の回収率変更指定（リスト。省略可）
element_group_membership：元素グループに追加する元素の指定（リスト。省略可）
periods：燃焼期間（*1）
aveflux：中性子束（numpy.array）
thermalpowers：出力（リスト）
thermalpower_unit：出力の単位（WもしくはMW．デフォルトはW）
load_module：実行モジュールのパス。既定とは異なるものを使用する場合にのみ指定
outlist：出力ファイルのパス。既定とは異なるパスに出力する場合にのみ指定
debug：デバッグオプション（デフォルトはFalse）

**options：OrigenFort5BurnupCommandGenerator に set_options メソッドで与える情報

【戻り値】
Material オブジェクト（燃焼後）

*1: 燃焼期間は、燃焼期間とその単位のタプルを燃焼ステップ数分だけリストにしたものを与える。燃焼期間の単位は、s、m、h、d、y のいずれか。
*2: aveflux および thermalpowers はいずれか一方を指定する。

なお、origen_decay 関数については、このインターフェイスから中性子束および出力に関する指定を除いたものとなる。

中性子束を規格化し、マテリアル領域毎の出力を算出する関数 origen_normalize_power のインターフェイスを以下に示す。

【引数】
meshflux：MeshFlux オブジェクト
macset：MacroscopicCrossSectionSet オブジェクト
thermalpower：出力
unit：出力の単位（デフォルトは W）

【戻り値】
MeshFlux オブジェクト（規格化後）
ディクショナリ（マテリアル毎の出力）

崩壊データライブラリ更新用の origen_decay_library 関数のインターフェイスを以下に示す。この関数では引数で与えられた DecayConstantSet オブジェクトから得られる各核種の半減期値で、崩壊データライブラリの THALF の値を置換する。

【引数】
decaylib：OrigenDecayLibrary オブジェクト
dconst：DecayConstantSet オブジェクト

【戻り値】
OrigenDecayLibrary オブジェクト（更新後）

断面積データライブラリ更新用の origen_cross_section_library 関数のインターフェイスを以下に示す。この関数では引数で与えられた MicroscopicCrossSection オブジェクトおよび中性子束から 1 群断面積を作成し、その値で断面積データの値を更新する。なお、MicroscopicCrossSection オブジェクトの capture と n2n を断面積データライブラリの SNG、SNGX および SN2N、SN2NX へ割り当てるが、この際、断面積データライブラリ側にて断面積の分岐比が変わらないようにしている。

【引数】
xslib：OrigenCrossSectionLibrary オブジェクト
micro：MicroscopicCrossSection オブジェクト
flux：中性子束（numpy.array）

【戻り値】
OrigenCrossSectionLibrary オブジェクト（更新後）

Table 3.1.1　ORIGEN 用 SCHEME 関数一覧

関数名	内容
origen_burnup	燃焼計算を行う。
origen_decay	冷却計算を行う。
origen_fort5_for_burnup	燃焼計算用の OrigenFort5 オブジェクトを生成する。
origen_fort5_for_decay	冷却計算用の OrigenFort5 オブジェクトを生成する。
origen_normalize_power	中性子束を規格化し、マテリアル毎の出力を算出する。
origen_decay_library	DecayConstantSet で崩壊データライブラリを更新する。
origen_cross_section_library	MicroscopicCrossSection で断面積ライブラリを更新する。

3.1.3　使用例

origen_burnup 関数を使用した燃焼計算の例を以下に示す。

```
# 外部ライブラリ
decaylib = OrigenDecayLibrary(
    os.path.join(CODE_DIR、 "ORLIBJ40", "lib", "DECAYJ40.LIB"))

xslib = OrigenCrossSectionLibrary(
    os.path.join(CODE_DIR、 "ORLIBJ40", "lib", "600MMXICJ40.LIB"))

photonlib = OrigenPhotonLibrary(
    os.path.join(CODE_DIR、 "ORLIBJ40", "lib", "JNNOBREM.LIB"))


# SCHEME 関数
fractional_recoveries=[[92, 1, 1.0]、 [94, 1, 1.0]]

periods = [(182.50, "d"), (182.50, "d")]

thermalpowers = [72.20, 72.20]


burned_material = origen_burnup(
    mat、
```

```
    xslib、 decaylib、 photonlib、
    individual_element_fractional_recoveries=fractional_recoveries,
    periods=periods、
    thermalpowers=thermalpowers、 thermalpower_unit="MW"
    version="orlibj40" neutron_spectrum="fast",
    bas1_title="BURNUP SAMPLE")
```

基本的には origen_burnup 関数のインターフェイスに従って、引数を与えて関数を呼び出せば良い。このとき、各コマンドのパラメータの値を明示的に与える場合には、引数に「bas1_title=…」と与えれば良い。このときの与え方は OrigenBurnupCommandGenerator の set_options メソッドの場合と同一である。なお origen_burnup および origen_decay 関数では、各コマンドのパラメータのうち、bas1_title（1 番めの BAS カードの title パラメータ）の指定は必須であり、これが未指定の場合には例外が発生する。

3.2 ADS3D への組み込み

3.2.1 実装の方針

ADS3D に ORIGEN2 の燃焼計算機能を組み込むにあたり、まず ADS3D の入力である計算制御ファイルの仕様を拡張した。以下にその抜粋を示す。

```
solver:
  origen:
    version: "orlibj40"
    neutron_spectrum: "fast"
    num_sub_steps: 1
    decaylib: $MARBLE_CODE_PATH/ORLIBJ40/lib/DECAYJ40.LIB
    photonlib: $MARBLE_CODE_PATH/ORLIBJ40/lib/JNNOBREM.LIB
    xslib:
      PB0011: $MARBLE_CODE_PATH/ORLIBJ40/lib/600MMXICJ40.LIB
      PB0021: $MARBLE_CODE_PATH/ORLIBJ40/lib/600MMXICJ40.LIB
      PB003*: $MARBLE_CODE_PATH/ORLIBJ40/lib/600MMXICJ40.LIB
       …
```

上記の通り、solver ブロックに新たに ORIGEN2 コードによる計算条件を指定する origen キーワードを追加する。従来の burnup キーワードと origen キーワードは互いに排他関係にあり、同時に指定することはできない。

origen キーワード以下では、使用するモジュールを特定する version、neutron_spectrum キーワードと、使用する崩壊データ／断面積データ／光子収率データのファイルパスを指定する。断面積データについてはゾーン毎に個別に指定する。断面積データの指定はワイルドカードのアスタリスク「*」を使用することで、任意の文字列とマッチする複数のゾーンを指定することができる。また ADS3D の 1 ステップの計算につき、ORIGEN による計算を行うステップ数（IRF コマンドの数）を num_sub_steps で指定する。

ADS3D では従来の燃焼計算ソルバー BURNUP を用いた計算実行を管理するクラスとして SolverBurnup クラスが用意され、これを全体の計算処理を管理する Scenario クラスがコントロールする構成となっていた。今回、新たに ORIGEN2 コードによる燃焼計算に対応するにあたり、ORIGEN2 による燃焼計算を管理する SolverOrigen クラスを追加する。さらに、従来 Scenario クラスにて管理していた計算処理フローについて、燃焼計算部分のロジックを切り出して Senario クラスのサブクラスとし、このサブクラスを BURNUP ソルバー用と ORIGEN 用の 2 種類用意することとする。前者を ScenarioWithMarbleBurnup クラス、後者を ScenarioWithOrigenBurnup クラスとする。

3.2.2 実装

新規に追加もしくは修正を行った ADS3D のファイルの一覧を **Table 3.2.1** に示す。

Table 3.2.1　ADS3D の実装・修正ファイル一覧

ファイル	内容
ads3d/input/ 下	
CalcConditionFile.py	入力仕様変更にともなう修正
ads3d/main/ 下	
Controller.py	Scenario クラスの制御のための修正
Scenario.py	Scenario クラスを抽象クラス化し、以下を新規追加
	ScenarioWithMarbleBurnup クラス
	ScenarioWithOrigenBurnup クラス
SolverOrigen.py	SolverOrigen クラス（新規追加）
SolverSlaromuf.py	計算の対象とする核種の拡張にともなう修正

4. 検証計算

4.1 高速炉の燃焼計算

4.1.1 計算条件

MARBLE のサンプル計算として整備されている「もんじゅ」体系[3] について、SCHEME を用いて ORIGEN2 コードによる燃焼計算を行い、従来の BURNUP ソルバーによる計算結果と比較した。

この計算では、もんじゅ炉心（熱出力 716MW）を対象として 148 日の燃焼計算を行う。SLAROM-UF で用いる核データライブラリについては、どちらのケースでも JENDL-3.3 [4] を用いた。燃焼計算においては、BUNRUP の計算では、standard2006 の燃焼チェーンを使用した。ORIGEN2 の計算では JENDL-3.3 ベースのライブラリを使用した。

4.1.2 計算結果

燃焼末期のある位置における原子数密度の比較を **Table 4.1.1** に示す。ここである核種 i の原子数密度の差異 ΔN_i は、BURNUP により計算された原子数密度を N^i_{burnup}、ORIGEN2 により計算された原子数密度を $N^i_{origen2}$ としたとき、

$$\Delta N_i = \frac{N^i_{origen2} - N^i_{burnup}}{N^i_{burnup}} \tag{4.1}$$

で計算している。また standard2006 については、半減期の短い U-237、Np-238、Am-242 などをショートカットし、Np-237、Pu-238、Cm-242 に加算しているため、ORIGEN2 の結果のうち、これら 3 核種については同様の処理を行った。

この表から、従来の BURNUP ソルバーと今回整備した ORIGEN2 による燃焼計算の間で、1%以下の差が生じていることがわかる。これらの差異は、BURNUP ソルバーが使用した燃焼チェーン standard2006 と、ORIGEN2 コードの詳細な燃焼チェーンの差であると考えられるが、U-235 や Pu-239 などの主要な核種についてはほとんど差が無く、大きな問題はないものと考えられる。

燃焼期間中の実効増倍率変化を **Fig. 4.1.1** に示す。燃焼期間中の挙動はほぼ同じで、燃焼末期において 0.05%の差が生じていた。これらの結果から、SCHEME で ORIGEN2 を用いた場合でも十分な計算精度を持っていることが確認できた。

[3] marble/example/monju/scheme_burnup のサンプル計算
[4] オリジナルのサンプルインプットでは JENDL-3.2 が用いられている。

Table 4.1.1　もんじゅ体系の燃焼末期の原子数密度

核種	BURNUP	ORIGEN2	ΔN_i
U-234	8.6176E-10	8.5648E-10	-0.61%
U-235	9.2899E-06	9.2894E-06	-0.01%
U-236	4.1639E-07	4.1662E-07	0.05%
U-238	5.2949E-03	5.2949E-03	0.00%
Np-237	4.2483E-07	4.2598E-07*1	0.27%
Pu-238	7.5237E-08	7.4833E-08*2	-0.54%
Pu-239	8.6189E-04	8.6187E-04	0.00%
Pu-240	3.8050E-04	3.8051E-04	0.00%
Pu-241	1.8527E-04	1.8527E-04	0.00%
Pu-242	6.5006E-05	6.5008E-05	0.00%
Am-241	3.6329E-06	3.6202E-06	-0.35%
Am-242m	3.3938E-08	3.3849E-08	-0.26%
Am-243	1.8382E-06	1.8351E-06	-0.17%
Cm-242	1.3722E-07	1.3831E-07 *3	0.79%
Cm-243	1.5676E-09	1.5617E-09	-0.38%
Cm-244	9.8856E-08	9.8654E-08	-0.20%
Cm-245	1.5626E-09	1.5627E-09	0.00%

*1: U-237 の量を加算。*2: Np-238 の量を加算。*3: Am-242 の量を加算。

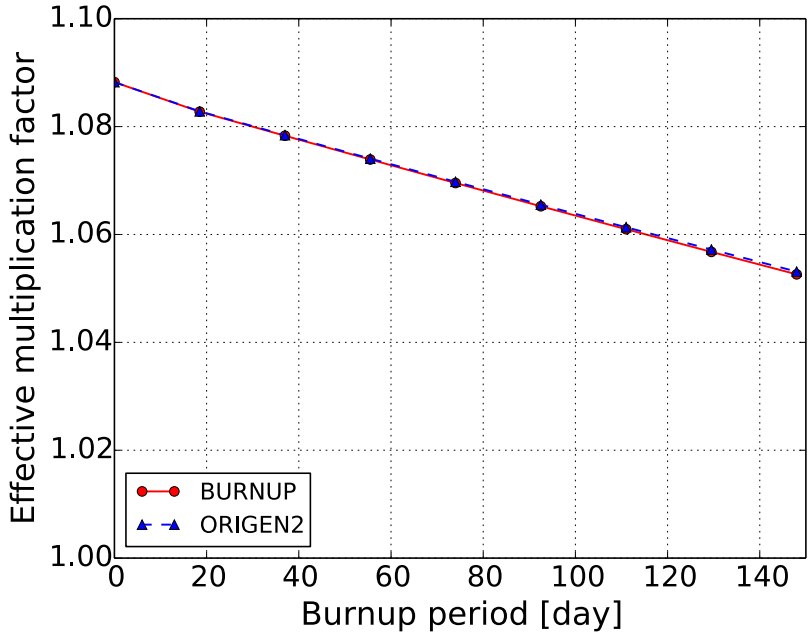

Fig. 4.1.1　もんじゅ体系の実効増倍率の変化

4.2 ADS の燃焼計算

4.2.1 計算条件

JAEA で検討している鉛ビスマス冷却型 ADS を対象として、ADS3D コードシステムに ORIGEN2 コードを用いて燃焼計算を行い、従来の BURNUP ソルバーによる計算の結果と比較した。解析の対象とした体系を **Fig. 4.2.1** に示す。この図では、燃料領域が 4 層に分かれているが、ここでは 1 領域炉心として全ての領域で同じ燃料を用いている。

この ADS 概念は熱出力 800MW、燃焼期間 600 日を想定しており、陽子ビームエネルギー 1.5GeV、最大ビーム出力 30MW (20mA) の陽子ビームで運転を行う。炉心に関する主要なパラメータを **Table 4.2.1** に示す。MA 窒化物燃料を用いた炉心概念となっている。

SLAROM-UF で用いる核データライブラリについては、どちらのケースでも JENDL-4.0 を用いた。燃焼計算においては、BUNRUP の計算では、ChainJ40[16] の燃焼チェーンを使用した。ORIGEN2 の計算では ORLIBJ40 を使用した。なお、どちらの計算ケースについても、200 日燃焼を 3 回続けるとし、ORIGEN2 の計算においては 1 回の燃焼計算におけるサブステップ数（`num_sub_steps`）を 10 とした [5]。

Table 4.2.1 対象 **ADS** の主要なパラメータ

Fuel assembly (FA)	
Number of FA	84
Pitch	233.9mm
Width	232.9mm
Number of fuel pins per FA	391
Number of tie rods per FA	6
Fuel	
Composition	(MA+Pu)N+ZrN
Pu enrichment	25.86%
ZrN ratio	56.20%
Pin outer diameter	7.65mm
Thickness of fuel pin	0.5mm
Pin pitch	11.48mm
Active height	1000mm

4.2.2 計算結果

燃焼末期の原子数密度の比較を **Table 4.2.2** に、燃焼を通じた実効増倍率変化の解析結果を **Fig. 4.2.2** に示す。実効増倍率の変化、燃焼末期の原子数密度ともに、BURNUP の計算結果と

[5] サブステップ数を減らして、一回の燃焼計算の時間幅（ここでは、200/10=20 日）が長くなると、結果が異なる可能性があるため、燃焼計算の時間幅をできるだけ小さくすることを推奨する（サブステップ数は〜12 まで設定可）。

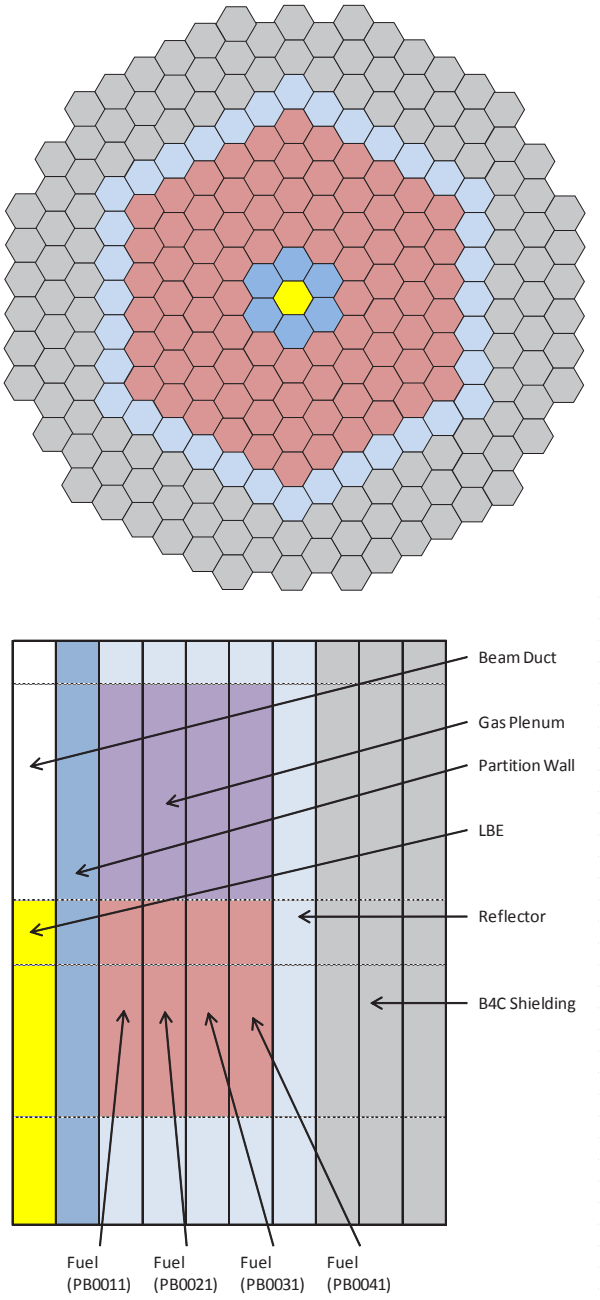

Fig. 4.2.1　ADS 体系の解析モデル

良く一致する結果が得られた。実効増倍率については、燃焼末期で 0.04%以内の差であり、原子数密度についても主要な核種については 0.1%以下の差で BURNUP の結果と一致しており、実用上問題がないことが確認された。

また ORIGEN2 を用いることで、FP を核種毎に扱うことが可能となった。例えば、代表的な長寿命核分裂生成物（LLFP）である Tc-99 と I-129 などを扱うことが可能となり、LLFP 核変換用の ADS 検討を行うことが可能となった。また、燃料交換時に生じるレアアース核種の残存も考慮することが可能となった。

Table 4.2.2　ADS 体系の燃焼後 3 年冷却後の原子数密度

核種	BURNUP	ORIGEN2	ΔN_i
Np-237	8.2310E-04	8.2275E-04	-0.04%
Pu-238	2.3736E-04	2.3767E-04	0.13%
Pu-239	3.5998E-04	3.5986E-04	-0.04%
Pu-240	2.2729E-04	2.2734E-04	0.02%
Pu-241	6.6019E-05	6.6005E-05	-0.02%
Pu-242	7.7824E-05	7.7862E-05	0.05%
Am-241	5.2524E-04	5.2499E-04	-0.05%
Am-242m	1.5070E-05	1.5084E-05	0.09%
Am-243	2.2515E-04	2.2507E-04	-0.04%
Cm-242	3.1811E-07	3.1778E-07	-0.10%
Cm-243	2.2549E-06	2.2640E-06	0.40%
Cm-244	1.0089E-04	1.0094E-04	0.05%
Cm-245	1.1489E-05	1.1502E-05	0.11%

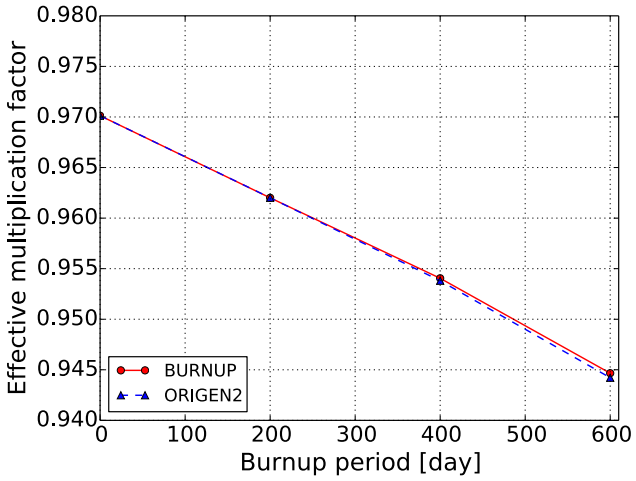

Fig. 4.2.2　ADS 体系の燃焼期間中の実効増倍率の変化

5. 結言

汎用炉心解析システム MARBLE を用いて高速炉や ADS の燃焼計算を行う場合に、核分裂生成物を核種毎に扱うため、燃焼計算コード ORIGEN2 コードを MARBLE で利用できるよう整備を行った。すなわち MARBLE 用に ORIGEN2 コードのカプセル化を行い、SCHEME および ADS3D コードシステムで扱えるよう実装した。

高速炉及び ADS を対象として検証計算を行った結果、燃焼チェーンの違いによる差が認められたものの、従来用いられてきた BURNUP モジュールによる解析結果とほぼ同じ結果が得られることを確認した。

これにより燃焼計算における核分裂生成物を核種毎に扱うことが可能となり、特に ADS 解析における燃料交換時のレアアース等の残存を考慮することが可能となった。また長寿命核分裂生成物の核変換を目的とした核変換システムの解析も可能となった。

謝辞

本コードの整備を行うにあたり、原子力基礎工学研究センター 核工学・炉工学ディビジョン 炉物理標準コード研究グループの横山賢治氏および株式会社 NESI の神智之氏には多くの有意義なコメントを頂きました。また、原子力基礎工学研究センター 分離変換技術開発ディビジョン 核変換システム開発グループの各位にも有意義なコメントを頂きました。ここに感謝の意を表します。

参考文献

1) K. Tsujimoto, T. Sasa, K. Nishihara et al., “Neutronics Design for Lead-Bismuth Cooled Accelerator-Driven System for Transmutation of Minor Actinide”, J. Nucl. Sci. and Technol., 41, 1, pp. 21-36 (2004).

2) K. Tsujimoto, H. Oigawa, N. Ouchi et al., “Research and Development Program on Accelerator-Driven System in JAEA”, J. Nucl. Sci. and Technol., 44, 3, pp. 483-490 (2007).

3) 辻本 和文、西原 健司、武井 早憲 他, “鉛ビスマス冷却加速器駆動システムを用いた核変換技術の成立性検討”, JAEA-Research 2010-012, (2010), 59p.

4) T. Sugawara, K. Nishihara, H. Iwamoto, et al., “Current Activities for Research and Development on Accelerator-Driven System in JAEA”, Proceedings of Global 2015, Paris, France, 20-24 Sept. (2015).

5) 横山 賢治、巽 雅洋、平井 康志 他, “次世代炉心解析システム MARBLE の開発”, JAEA-Data/Code 2010-030, (2010), 148p.

6) K. Yokoyama, T. Hazama, K. Numata, T. Jin, “Development of comprehensive and versatile framework for reactor analysis, MARBLE”, Annals of Nuclear Energy, 66, pp. 51-60, (2014).

7) 横山 賢治 他, “汎用炉心解析システム MARBLE2 の開発”, JAEA-Data/Code 2015-009, (2015), 120p.

8) 菅原 隆徳、平井 康志、西原 健司 他, “加速器駆動核変換システム用三次元炉心解析コード ADS3D の整備”, JAEA-Data/Code 2014-024, (2015), 86p.

9) A. G. Croff, “A Users Manual for the ORIGEN2 Computer Code”, ORNL/TM-7175, (1980).

10) 奥村 啓介、杉野 和輝、小嶋 健介 他, “JENDL-4.0 に基づく ORIGEN2 用断面積ライブラリセット：ORLIBJ40”, JAEA-Data/Code 2012-032, (2013), 148p.

11) “NEA-1642 ZZ-ORIGEN2.2-UPJ”, https://www.oecd-nea.org/tools/abstract/detail/nea-1642 , (2006).

12) T. Hazama, G. Chiba and K. Sugino, “evelopment of a fine and ultra-fine group cell calculation code SLAROM-UF for fast reactor analyses”, J. Nucl. Sci. Technol., 43, 8, pp. 908-918, (2006).

13) T. B. Flower et al., “Nuclear reactor core analysis code : Citation”, ORNL-TM-2496, Rev. 2, (1969).

14) R. E. Alcouffe, R. S. Baker, F. W. Brinkley et al., “DANTSYS: A Diffusion Accelerated Neutral Particle Code System,” LA-12969-M (1995).

15) R. E. Alcouffe, R. S. Baker, J. A. Dahl, S.A. Turner, and Robert Ward, “PARTISN: A Time-Dependent, Parallel Neutral Particle Transport Code System,” LA-UR-05-3925 (2005).

16) K. Okumura, S. Asai, Y. Hanzawa et al., “Analyses of Assay Data of LWR Spent Nuclear Fuels with a Continuous-Energy Monte Carlo Code MVP and JENDL-4.0 for Inventory Estimation of ^{79}Se, ^{99}Tc, ^{126}Sn and ^{135}Cs”, Prog. in Nucl. Sci. and Tecnol. 2, pp. 369-374 (2011).

付録A　ORIGEN2コマンドの説明

今回実装したORIGEN2コマンドの簡単な説明を以下に示す。詳細はORIGEN2コードのマニュアル[9]を参照のこと。

- LIP: インプットデータライブラリの表示を制御
- LPU: 核種定義の読み込み
- LIB: 崩壊および断面積ライブラリの指定
- PHO: 光子ライブラリの指定
- RDA: コメントの記述
- TIT: 出力におけるタイトルを与える
- BAS: 出力におけるコメント（basis）を与える
- HED: ベクトル見出しの指定
- CUT: 出力テーブルにおけるカットオフ割合の指定
- INP: 核種組成、燃料供給率・除去率の指定
- MOV: ベクトルからベクトルへの核種組成の移動
- BUP: 燃焼計算の指定
- IRP: 比出力による照射の指定
- IRF: 中性子束による照射の指定
- DEC: 崩壊計算の指定
- OPTA: アクチノイド核種についての出力テーブルの指定
- OPTF: 核分裂生成物核種についての出力テーブルの指定
- OPTL: 放射化物についての出力テーブルの指定
- OUT: 計算結果の出力
- END: 計算終了

This is a blank page.

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 組立単位	
	名称	記号
面積	平方メートル	m^2
体積	立方メートル	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	キログラム毎立方メートル	kg/m^3
面積密度	キログラム毎平方メートル	kg/m^2
比体積	立方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
量濃度(a)，濃度	モル毎立方メートル	mol/m^3
質量濃度	キログラム毎立方メートル	kg/m^3
輝度	カンデラ毎平方メートル	cd/m^2
屈折率(b)	（数字の） 1	1
比透磁率(b)	（数字の） 1	1

(a) 量濃度（amount concentration）は臨床化学の分野では物質濃度（substance concentration）ともよばれる。
(b) これらは無次元量あるいは次元１をもつ量であるが、そのことを表す単位記号である数字の１は通常は表記しない。

表３．固有の名称と記号で表される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(b)	rad	1 (b)	m/m
立体角	ステラジアン(b)	sr(c)	1 (b)	m^2/m^2
周波数	ヘルツ(d)	Hz		s^{-1}
力	ニュートン	N		m kg s^{-2}
圧力，応力	パスカル	Pa	N/m^2	m^{-1} kg s^{-2}
エネルギー，仕事，熱量	ジュール	J	N m	m^2 kg s^{-2}
仕事率，工率，放射束	ワット	W	J/s	m^2 kg 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 kg s^{-3} A^{-1}
静電容量	ファラド	F	C/V	m^{-2} kg^{-1} s^4 A^2
電気抵抗	オーム	Ω	V/A	m^2 kg s^{-3} A^{-2}
コンダクタンス	ジーメンス	S	A/V	m^{-2} kg^{-1} s^3 A^2
磁束	ウエーバ	Wb	Vs	m^2 kg s^{-2} A^{-1}
磁束密度	テスラ	T	Wb/m^2	kg s^{-2} A^{-1}
インダクタンス	ヘンリー	H	Wb/A	m^2 kg s^{-2} A^{-2}
セルシウス温度	セルシウス度(e)	℃		K
光束	ルーメン	lm	cd sr(c)	cd
照度	ルクス	lx	lm/m^2	m^{-2} cd
放射性核種の放射能(f)	ベクレル(d)	Bq		s^{-1}
吸収線量，比エネルギー分与，カーマ	グレイ	Gy	J/kg	m^2 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	シーベルト(g)	Sv	J/kg	m^2 s^{-2}
酸素活性	カタール	kat		s^{-1} mol

(a)SI接頭語は固有の名称と記号を持つ組立単位と組み合わせても使用できる。しかし接頭語を付した単位はもはやコヒーレントではない。
(b)ラジアンとステラジアンは数字の１に対する単位の特別な名称で、量についての情報をつたえるために使われる。実際には、使用する時には記号radおよびsrが用いられるが、習慣として組立単位としての記号である数字の１は明示されない。
(c)測光学ではステラジアンという名称と記号srを単位の表し方の中に、そのまま維持している。
(d)ヘルツは周期現象についてのみ、ベクレルは放射性核種の統計的過程についてのみ使用される。
(e)セルシウス度はケルビンの特別な名称で、セルシウス温度を表すために使用される。セルシウス度とケルビンの単位の大きさは同一である。したがって、温度差や温度間隔を表す数値はどちらの単位で表しても同じである。
(f)放射性核種の放射能（activity referred to a radionuclide）は、しばしば誤った用語で"radioactivity"と記される。
(g)単位シーベルト（PV,2002,70,205）についてはCIPM勧告2（CI-2002）を参照。

表４．単位の中に固有の名称と記号を含む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} kg s^{-1}
力のモーメント	ニュートンメートル	N m	m^2 kg s^{-2}
表面張力	ニュートン毎メートル	N/m	kg s^{-2}
角速度	ラジアン毎秒	rad/s	m m^{-1} s^{-1}=s^{-1}
角加速度	ラジアン毎秒毎秒	rad/s^2	m m^{-1} 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 s^{-3}
熱容量，エントロピー	ジュール毎ケルビン	J/K	m^2 kg s^{-2} K^{-1}
比熱容量，比エントロピー	ジュール毎キログラム毎ケルビン	J/(kg K)	m^2 s^{-2} K^{-1}
比エネルギー	ジュール毎キログラム	J/kg	m^2 s^{-2}
熱伝導率	ワット毎メートル毎ケルビン	W/(m K)	m kg s^{-3} 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} kg s^{-2}
電界の強さ	ボルト毎メートル	V/m	m kg s^{-3} A^{-1}
電荷密度	クーロン毎立方メートル	C/m^3	m^{-3} s A
表面電荷	クーロン毎平方メートル	C/m^2	m^{-2} s A
電束密度，電気変位	クーロン毎平方メートル	C/m^2	m^{-2} s A
誘電率	ファラド毎メートル	F/m	m^{-3} kg^{-1} s^4 A^2
透磁率	ヘンリー毎メートル	H/m	m kg s^{-2} A^{-2}
モルエネルギー	ジュール毎モル	J/mol	m^2 kg s^{-2} 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 kg s^{-2} K^{-1} 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} s A
吸収線量率	グレイ毎秒	Gy/s	m^2 s^{-3}
放射強度	ワット毎ステラジアン	W/sr	m^4 m^{-2} kg s^{-3}=m^2 kg s^{-3}
放射輝度	ワット毎平方メートル毎ステラジアン	$W/(m^2$ sr)	m^2 m^{-2} kg s^{-3}=kg s^{-3}
酵素活性濃度	カタール毎立方メートル	kat/m^3	m^{-3} s^{-1} mol

表５．SI 接頭語

乗数	名称	記号	乗数	名称	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６．SIに属さないが、SIと併用される単位

名称	記号	SI 単位による値
分	min	1 min=60 s
時	h	1 h =60 min=3600 s
日	d	1 d=24 h=86 400 s
度	°	1°=(π/180) rad
分	’	1’=(1/60)°=(π/10 800) rad
秒	”	1”=(1/60)’=(π/648 000) rad
ヘクタール	ha	1 ha=1 hm^2=10^4m^2
リットル	L, l	1 L=1 l=1 dm^3=10^3cm^3=$10^{-3}m^3$
トン	t	1 t=10^3 kg

表７．SIに属さないが、SIと併用される単位で、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 単位で表される数値
電子ボルト	eV	1 eV=1.602 176 53(14)×10^{-19}J
ダルトン	Da	1 Da=1.660 538 86(28)×10^{-27}kg
統一原子質量単位	u	1 u=1 Da
天文単位	ua	1 ua=1.495 978 706 91(6)×10^{11}m

表８．SIに属さないが、SIと併用されるその他の単位

名称	記号	SI 単位で表される数値
バール	bar	1 bar=0.1MPa=100 kPa=10^5Pa
水銀柱ミリメートル	mmHg	1 mmHg≈133.322Pa
オングストローム	Å	1 Å=0.1nm=100pm=10^{-10}m
海里	M	1 M=1852m
バーン	b	1 b=100fm^2=(10^{-12}cm)^2=$10^{-28}m^2$
ノット	kn	1 kn=(1852/3600)m/s
ネーパ	Np	SI単位との数値的な関係は、対数量の定義に依存。
ベル	B	
デシベル	dB	

表９．固有の名称をもつCGS組立単位

名称	記号	SI 単位で表される数値
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 =1cm^2 s^{-1}=$10^{-4}m^2$ s^{-1}
スチルブ	sb	1 sb =1cd cm^{-2}=10^4cd m^{-2}
フォト	ph	1 ph=1cd sr cm^{-2} =10^4lx
ガル	Gal	1 Gal =1cm s^{-2}=$10^{-2}ms^{-2}$
マクスウエル	Mx	1 Mx = 1G cm^2=10^{-8}Wb
ガウス	G	1 G =1Mx cm^{-2} =10^{-4}T
エルステッド(a)	Oe	1 Oe ≙ (10^3/4π)A m^{-1}

(a) ３元系のCGS単位系とSIでは直接比較できないため、等号「 ≙ 」は対応関係を示すものである。

表10．SIに属さないその他の単位の例

名称	記号	SI 単位で表される数値
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
ガンマ	γ	1 γ=1 nT=10^{-9}T
フェルミ		1 フェルミ=1 fm=10^{-15}m
メートル系カラット		1 メートル系カラット= 0.2 g = 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	1 cal=4.1858J（「15℃」カロリー），4.1868J（「IT」カロリー），4.184J（「熱化学」カロリー）
ミクロン	μ	1 μ =1μm=10^{-6}m

（第8版，2006年）