

評価済み核データファイル作成支援コード DeCE

ロスアラモス国立研究所

河野 俊彦

kawano@lanl.gov

1. はじめに

ENDF や JENDL のデータが格納されている評価済み核データファイルは、過去のパンチカードイメージを引き継ぎ、一行に極力多くのデータを詰め込もうとする非常に厄介なフォーマットとなっている。一般ユーザが実際に元データファイルを直接利用する機会は少ないとは言え、このフォーマットに慣れていない人は、どのデータがどこに書かれているのか、マニュアルと首っ引きで読んでも埒が明かない。非常な厄介さは非情さも兼ね備え、評価済みデータファイルを作成するとなると数多くの難解仔細な規則に立ち往生する羽目になる。評価済みデータファイルの作成と利用に関わる多くの人は、なんらかのツールを利用してファイルの読み書きをしている。核反応理論計算コード EMPIRE [1] や TALYS [2, 3] は ENDF ファイル作成ツールをパッケージに含んでいるが、これらコードの計算結果をファイル化するのに特化されたものである。CCONE [4, 5] も同様だと想像する。なおここで ENDF ファイルと言っているのは、合衆国の評価済み核データファイルではなく、ENDF-6 フォーマット [6] で作成された一般的な核データファイルを指している。

そんな ENDF ファイルを作成するのに優れたツールが、JAEA の中川庸雄氏が開発された CRECTJ コード [7] である。ENDF ファイルの読み書きに留まらず、内容の修正や反応断面積の和の計算など、評価済み核データファイルの編集に必要な操作をプログラム感覚で行える。自分が JENDL 開発に携わっていたとき CRECTJ は必要不可欠なツールであったが、残念ながら国外への持ち出しは(表立って)できないのことで、困った状況に直面した。他方、自分でも簡単な ENDF 読み書きツールを Perl で作成してはいたが、融通がきかない ENDF フォーマットに辟易しつつ CRECTJ 並の機能を持ったコードの必要性を強く感じ、独自のコード DeCE の開発が始まった。DeCE に関する論文を今年出したので [8]、ここではその論文に書かなかった(書けなかった)話題を中心にコードを紹介する。

2. DeCE コード

2.1 開発の経緯

統計模型計算コード CoH₃ の出力を ENDF ファイル化するための短い C++コード E-Type が最初に書かれ、その後 ENDF フォーマットを読み書きする C++ライブラリ endflib と endfio に書

き直された。それと平行してインタラクティブな ENDF ファイル操作コード DeCE (Descriptive Correction of ENDF-6 format) を作成した。「記述的修正」とはやや変な表現であるが、実のところ dece という名前を先に思いつき、後から無理にアクリニムを付けた経緯がある。dece という単語に意味はなく、強いて言うなら decent の短縮と認識される程度である。一般名詞にしなかったのは、検索時に発見されやすいようにする、すなわち Googlability を上げるためである。実際“dece kawano”で検索すれば GitHub に置かれたソースコードがすぐに見つかるはずである。

「ディース」と発音するが、イタリア系の友人は「ディーチェ」、英国系は「ダイス」と発音してるのがおもしろい。ダイスに触発され、DeCE のロゴは以下のごとくサイコロ形になっている。



図 1 DeCE logo

ENDF ファイルは単なるテキストファイルなので、ファイル作成はテキストエディタがあれば可能である。但し、些細な変更を施しただけで、処理コードから蹴られてしまいがちなのが ENDF フォーマットの厳しさでもある。ファイルの始めの方に、データの履歴や評価の手法が通常の英文で書かれたコメント部分がある。ここに一行書き加えただけで、処理コードはもう動いてくれない。なぜならファイル先頭付近に「コメント部分は全部で〇〇行ありますよ」という数字があり、ここをひとつ増やしておかないと、以後全部ずれてしまうからである。そんなファイル見たら分かるじゃないかと文句を言うのは、パンチカードに触れたことがない世代である。勿論自分も触れたことはない。

さらに各部分には行番号が振られており、一行追加すれば、その後ろの行番号を全部ずらす必要がある。そんなの見たら分かるじゃないかと言うのは、パンチカードの入ったボックスを落としてカードを床にぶちまけてしまった経験のない世代である。ちなみに自分はこの経験はある。パンチカードに触れたことがないくせにカードをぶちまけた経験は矛盾すると言うのなら、カードは専用の引き出しに入っていた。カードに触れずとも、床にぶちまけることは技術的に可能である。

コメント部の一番下には、ディクショナリと呼ばれる特別なセクションがあり、以下に続

く個々のデータセクションがそれぞれ何行あるかが書かれている。これも以下の部分を見れば不要と思われるが、入れることが決まっているので仕方ない。プロトコルの変更は得てして保守的になりがちである。こういった作業を自動化してしまおうというのも、DeCE 開発の目的の一つであった。図 2(a)に、既存のファイルへコメントを数行追加した例を示している。またこのファイルにはディクショナリがない。DeCE を通すと (b) のようになり、テキスト行数を正しく計算し、ファイルの内容からディクショナリを自動生成している。また行番号も自動的に振られている。ちなみに現行 ENDF フォーマットではようやく行番号がオプションとなったので、最新版 DeCE はデフォルトでは行番号をつけないようになっている。

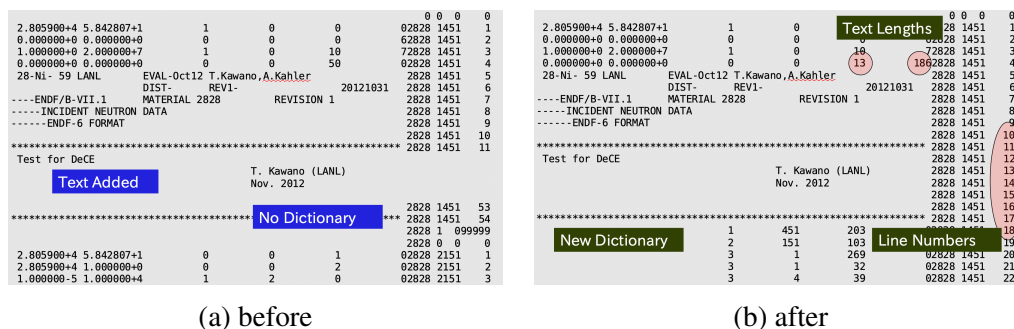


図 2 DeCE による行数調整，ディクショナリ作成，ならびに行番号付け。

2.2 基本的なコードの機能

基本的な DeCE の動作は、まず ENDF ファイルを読み込み、与えられたコマンドを一つずつ処理し、最終的に新たな ENDF ファイルを作成する流れとなる。詳細なコマンドの一覧は DeCE 配布パッケージに付属するマニュアルに譲るが、他のファイルにあるデータを読み込んで ENDF フォーマット化したり、2つの断面積の和や差を取るなど、CRECTJ の機能を踏襲している。もちろん CRECTJ のクローンではなく、DeCE ユーザが一番よく利用しているのは、ENDF ファイルの内容の表示である。コマンドラインオプションで MF, MT 番号を指定すれば、その核データが読みやすい形式になって表示される。例えば JENDL-4 の $^{56}\text{Fe}(n, 2n)$ 断面積は以下のようにになっている。

```
% dece -f 3 -t 16 Fe056.dat
# [ 2631 : 3 : 16 ] 26 - 56
# Cross section
# QM -1.120270e+07 mass difference Q-value
# QI -1.120270e+07 reaction Q-value
# NR 1 number of interpolation range
# NP 11 lin-lin interpolation
1.140470e+07 0.000000e+00
1.170000e+07 1.622410e-02
1.200000e+07 4.800450e-02
...
```

読み込む ENDF ファイルは完全な評価データである必要はなく、例えば核分裂断面積 MF3, MT18 だけの断片でも構わない。数値データを読み込んで ENDF 化する部分は、やや CoH_3

コード専用に傾斜した面もあるが、実際はテキストファイルであればどのようなデータでも読み込むことができるので、他のコードでも利用できるはずである。

DeCE は共鳴パラメータから断面積を計算する NJOY [9] の reconr 相当の機能も持つ。但し NJOY や FRENDY [10] のような処理コードではないので、Doppler 計算や群定数計算は行わず、また計算精度を保証するものではない。一番簡単な使用方法は、`-r` オプションを与えることで、OK での熱中性子断面積を印刷する。

```
% dece -r U235.dat
# Energy[eV]   Total[b]      Elastic[b]   Capture[b]   Fission[b]
  2.530000e-02  6.974081e+02  1.507620e+01  9.847858e+01  5.838533e+02
```

NJOY の計算結果と比べて 0.2%ほどの差があるが、そこまで厳密な精度保証はしていないので、実用範囲内であろう。そもそも自分は処理コード専門家でもないので、時間が許せば NJOY と全く同じ数値になるよう修正するかもしれないという体たらくである。

共鳴計算に関しては、NJOY との比較で NJOY 側に軽微な修正を施すことになったこともある。断面積が負になったというエラーメッセージを NJOY が出す核種があったのだが、DeCE で計算した断面積は常にプラスであった。数値計算での丸め誤差かと NJOY を調べてみると、断面積が極めて小さい場合、それがプラスであっても NJOY は「負になった」と判断してエラーメッセージを出していたのである。

2.3 E の怨念

以下、ある JENDL-4 のデータの一部である。

```
9.223500+4 2.330250+2      0      0      0      09228 3 1 1
0.000000+0 0.000000+0      0      0      2      4259228 3 1 2
      4      2      425      5      0      09228 3 1 3
1.000000-5 0.000000+0 2.530000-2 0.000000+0 7.733040+1 0.000000+09228 3 1 4
5.000000+2 4.855524-9 5.000000+2 3.316550+1 5.029950+2 3.446720+19228 3 1 5
```

パソコンで見慣れた数値表現と違い、E が落とされている。 10^{-5} は ISO6093 表記では 1.00000E-5 と書かれるが、たった一文字“E”を節約して有効桁数を稼いでいるのが ENDF フォーマットである。これは FORTRAN の入出力の慣例に従っているのだが、他の言語ではこれは読めない。endfrib には ENDFPadExp(), ENDFDelExp() という 2 つの関数があり、E 文字の挿入・削除を行っている。問題は FORTRAN の表記方法にバラエティがあり、どこまでそれに対応できるかという点である。1.0 -5 も 10^{-5} として読み込めと言われても、簡単に納得できるものではない。

2.4 固定されたバージョン

DeCE のソースコードは GitHub で公開されている。

<https://github.com/toshihikokawano/DeCE>

macOS や Linux であれば、git の clone コマンド

```
% git clone https://github.com/toshihikokawano/DeCE.git
```

でローカルに DeCE というディレクトリができる。git を使わない場合は、上記 URL のページに表示される Clone or Download ボタンから ZIP ファイルをダウンロードすればよい。パッケージは GNU Autotools で管理されており、コンパイルが簡略化されている。

2016 年に DeCE が BSD ライセンスでオープンソース化されて以来、DeCE のバージョンは v.1.2 である。おそらくこの番号が変わることは永遠に無い。その公開手続きの際、うっかりバージョン番号を丁寧に書き込んでしまったものだから、DeCE 1.2 がオープンソースとなってしまった。つまり 1.3 にすると面倒な書類手続きを繰り返さなければならない。一旦オープンソースになっているので二度目はより簡単であろうと思うが、そういうのは避けたいのが人情である。

永久 1.2 の代わりに、DeCE には鉱物にちなんだコードネームが付けられている。 β 版時代のコードネームは滑石、石膏、方解石、蛍石、ver.1 になったのが燐灰石からで、これはモース硬度にちなんでいる。柔らかい石から始まり、最終目的は全くバグの無い安定版硬度 10 のダイヤモンドになるはずである。現在は黄鉄鉱、硬度が 6~6.5 で、まだ石英にも届かないが、実用上問題ないはずである。おそらくダイヤモンドに到達する前に、ENDF フォーマットそのものが時代遅れになっていることだろう。ENDF の XML 化が進んでいるところである [11, 12]。それより前に自分自身のリタイアがやってきそうでもある。

ちなみに DeCE には ENDF ファイルを XML 化する機能もあった。しかし ENDF の XML 化でのスキーマが頻繁に変わり、現時点でそれに追従するのは無駄として、その部分は全て削除された。将来復活することもあるかと、一旦バックアップファイルに移しておいたのだが、それを git で管理することなく削除してしまったのは、あまり人に言えない秘密である。

3. おわりに

手前味噌ではあるが、DeCE はほぼ日常的に使う重要なツールとなっている。元々 DeCE はあくまで個人用に開発したもので、利用できる編集コマンドも個人の需要に基づいている。コードそのものの需要もさほどないであろうと高を括っていたが、グループ内で共有していると次第に同僚らの利用も増え、また彼・彼女らのリクエストから機能も増えていった。核データ界限で十分需要と実用性があると考え、オープンソース化に踏み切った。現時点で最も厄介なリクエストは Windows 版が欲しいというものであるが、Windows マシンが周りに一台も無い環境なので、誰かバイナリを作成してくれないかと密かに願っている。カスタマーサポートは厄介な面もあるが、他研究所からのバグレポートもあり、品質向上に役立っている。

参考文献

- [1] M. Herman, R. Capote, B. V. Carlson, P. Obložinský, M. Sin, A. Trkov, H. Wienke, and V. Zerkin, “EMPIRE: Nuclear reaction model code system for data evaluation,” *Nuclear Data Sheets*, **108**, 2655 (2007).
- [2] A. J. Koning, S. Hilaire, and M. C. Duijvestijn, “TALYS-1.0,” *EPJ Web of Conferences*, Proc. Int. Conf. on Nuclear Data for Science and Technology, 22 – 27 Apr., 2007, Nice, France, pp. 211 – 214 (2008).

- [3] A. J. Koning and D. Rochman, “Modern nuclear data evaluation with the TALYS code system,” *Nuclear Data Sheets*, **113**, 2841 (2012).
- [4] O. Iwamoto, “Development of a comprehensive code for nuclear data evaluation, CCONE, and validation using neutron-induced cross sections for uranium isotopes,” *Journal of Nuclear Science and Technology*, **44**, 687 (2007).
- [5] O. Iwamoto, N. Iwamoto, S. Kunieda, F. Minato, and K. Shibata, “The CCONE code system and its application to nuclear data evaluation for fission and other reactions,” *Nuclear Data Sheets*, **131**, 259 (2016).
- [6] A. Trkov, M. Herman, and D. A. Brown, “ENDF-6 formats manual, data formats and procedures for the evaluated nuclear data files, ENDF/B-VI and ENDF/B-VII,” ENDF-102, BNL-90365-2009 Rev.2, Brookhaven National Laboratory (2012).
- [7] T. Nakagawa, “CRECTJ: A computer program for compilation of evaluated nuclear data,” JAERI-Data/Code 99-041, Japan Atomic Energy Research Institute (1999).
- [8] T. Kawano, “DeCE: the ENDF-6 data interface and nuclear data evaluation assist code,” *Journal of Nuclear Science and Technology*, **56**, 1029 (2019).
- [9] R. E. MacFarlane and A. C. Kahler, “Methods for processing ENDF/B-VII with NJOY,” *Nuclear Data Sheets*, **111**, 2739 (2010).
- [10] K. Tada, Y. Nagaya, S. Kunieda, K. Suyama, and T. Fukahori, “Development and verification of a new nuclear data processing system FRENDY,” *Journal of Nuclear Science and Technology*, **54**, 806 (2017).
- [11] B. R. Beck and C. M. Mattoon, “FUDGE: A toolkit for nuclear data management and processing,” LLNL-PROC-648476, Lawrence Livermore National Laboratory (2014).
- [12] C. M. Mattoon, B. R. Beck, N. R. Patel, N. C. Summers, G. W. Hedstrom, and D. A. Brown, “Generalized nuclear data: A new structure (with supporting infrastructure) for handling nuclear data,” *Nuclear Data Sheets*, **113**, 3145 (2012).