

Status of REBUS Fuel Management Software Development for RERTR Applications

Arne P. Olson

Argonne National Laboratory
Argonne, Illinois 60439-4841 USA

Presented at the 2000 International Meeting
on Reduced Enrichment for Research and Test Reactors

October 1-6, 2000
Las Vegas, Nevada

Work supported by the U.S. Department of Energy
Office of Nonproliferation and National Security
under Contract No. W-31-109-ENG-38.

Status of REBUS Fuel Management Software Development for RERTR Applications

Arne P. Olson

Argonne National Laboratory
9700 S. Cass Avenue
Argonne, Illinois 60439-4841 USA

ABSTRACT

The REBUS-5 burnup code has evolved substantially in order to meet the needs of the ANL RERTR Program. This paper presents a summary of the past changes and improvements in the capabilities of this software, and also identifies future plans.

INTRODUCTION

The REBUS (Reactor burnup System) code has evolved substantially over more than three decades¹⁻⁵. The original concept of the code was to maximize its effectiveness for reactor fuel cycle and burnup calculations using the available large mainframe computers of the late 1960's. The best computers at that time were characterized by limited memory capacity, by different kinds of memory (main core, bulk core), and by short or long word length. Consequently the design of complex computer simulation and analysis codes was significantly constrained by machine hardware limitations, as well as by software limitations. This paper will discuss the advances in reactor fuel cycle simulation that have become available since the advent of powerful personal computers (PC's) and PC clusters that no longer are restricted by memory limitations. Advances in computer operating systems and compilers have also overcome, or even overwhelmed, many past limitations.

ADVANCES IN USING BURNUP-DEPENDENT NEUTRON CROSS SECTIONS

Monte Carlo Methods

REBUS is a very general code that uses multigroup finite-difference diffusion theory methods as embodied in the DIF3D code⁶⁻⁷. But diffusion theory cannot cope accurately with complex geometrical details or with strong neutron absorbers, as can Monte Carlo methods. MCNP⁸ is the most widely used Monte Carlo neutron/photon transport code because it is general-purpose, because it has generalized geometrical modeling, and because it can be run in parallel using the Parallel Virtual Machine (PVM) multiprocessing capability. About two years ago, the REBUS code was extended by including MCNP as an option to replace DIF3D⁹. MCNP can be used to generate burnup-dependent neutron fluxes and principal reaction cross sections. The scattering matrices are known to be not burnup-dependent. Consequently the reaction processes which must be modeled as burnup-dependent are: capture, fission, (n,alpha), (n,d),

(n,p), and (n,t). This coupled code, known as MC-REBUS, has operated successfully on a cluster of PC's under the Linux operating system. The coupled code operates with the aid of two interface routines which:

1. Prepare atom density data by burnup zone for use in MCNP;
2. Prepare the burnup-dependent data neutron flux and cross section data from MCNP in standard file formats for use in the Fuel Cycle Computational Module of REBUS.

The basic capabilities of REBUS-5 have evolved significantly over the past two years. These changes will be addressed below. We plan to upgrade MC-REBUS to include all of these changes later in this calendar year.

Diffusion Theory Methods

DIF3D is a 1-, 2-, or 3-dimensional finite-difference diffusion theory code that can be used in a wide variety of geometries. It is a multigroup code. Nodal methods for solution of the multigroup diffusion equations in a few geometries have been incorporated into DIF3D. The production version of VARIANT from 1999 (plus minor upgrades) is now used in REBUS-5.

Burnup Analysis Methods

The design and analysis of large breeder reactors of the LMFBR type was the original application for which REBUS was intended. However, due to its modular design and completely variable dimensioning, it soon was applied to all kinds of reactor systems in which neutron irradiation over time and fuel cycle analysis was an essential component of the problem. REBUS has been used since the start of the RERTR Program, because of its flexibility, and its quality of simulation. On the other hand, it has not been an easy code to use by the uninitiated due to the complexity of its general-purpose design.

At ANL, the cross sections are typically generated by the WIMS-RERTR code. For thermal reactor applications, burnup-independent cross sections that were effective for LMFBR applications are not enough. Consequently ANL some years ago developed a form of burnup-dependent cross sections which was better suited for research reactors. Only neutron capture and fission cross sections were explicitly burnup-dependent through use of least-squares-fitted polynomial representations. A module called POLYFI (POLYnomial FI) was created which enabled generation of burnup-dependent macroscopic cross sections at each burnup time step, for use by DIF3D. The implementation of burnup-dependent cross section data use in REBUS was implemented in the HMG4C module, which generated the macroscopic cross section file COMPXS, which in turn was passed to DIF3D in order to obtain the neutron flux distribution at each time point. A fully consistent treatment was obtained through changes to the Fuel Cycle Computational Module (FCC001) for computation of the transmutation matrix (the coefficient matrix of the set of differential equations which model atomic density changes with time). Reaction processes (n,alpha), (n,d), (n,p), and (n,t) were not yet processed as burnup-dependent. As a result, the implementation was not complete in terms of all of the possible reaction processes which could make up the isotopic transmutations. Hence the predicted atom densities across a burn step were better than before, because the neutron flux level in each burnup zone was affected by the burnup dependence.

The recent work which connected MCNP to REBUS provided an improvement to this problem: the coding used by the interface code which communicated between MCNP and REBUS, as well as the revised module FCC004, which was changed to utilize the burnup-dependent data returned by MCNP, now should be put to work by completing the burnup-dependent simulation model when using DIF3D for the neutron flux calculations. The cross sections returned by MCNP were written to a binary file in standard format (ISOTXS). The necessary software to implement this change when using DIF3D is a small extension of the HMG4C module to create a new binary file containing the latest interpolated burnup-dependent principal cross sections at that point in time. Prior to computing predicted atom densities for the next calculation, the presence of this file triggers its use. This upgrade will be taken in the next few months. At that point, we will have a logically simple, complete, and consistent treatment of burnup dependence for either MCNP or DIF3D.

Meanwhile, many years of use of burnup-dependent cross sections made it apparent that there were other inadequacies in the model. Some research reactors have important reaction processes other than capture and fission. For example, burnable poisons involving lithium must deal with the (n, α) reaction. Due to the design of REBUS, it was clear that modeling of the remaining reaction processes (n, α), (n,d), (n,p), and (n,t) could be readily accommodated at the same time as a new option for cubic spline fits for the burnup-dependence was being added to REBUS-5. These processes have now been implemented. Cubic splines offer an improved ability to track oddly varying neutron cross sections without the errors introduced by polynomial fits. Historical use had shown that while low-order fits may smoothly follow many cross sections, higher-order fits would pass near or through the data points but would introduce significant error in between. Highly unphysical negative cross sections were often the result (such results were detected by REBUS; appropriate error messages were added to the code output).

Natural cubic splines are used in many numerical fitting applications where a continuous smooth curve is desired (with continuous first derivative) and the function is assumed to be linear over the first and last intervals. The WIMS-ANL code can generate a set of neutron cross sections for a given isotope at many different burn points. The new spline interpolation process required very little change to the REBUS input requirements other than to provide the option and to extend the reaction processes available.

VALIDATION OF THE CUBIC SPLINE FIT OF BURNUP-DEPENDENT NEUTRON CROSS SECTIONS

The reactor analyst needs information in useful form for understanding the physical variables and their effects on his reactor model. Many years ago, a graphics package called DISPLA was used by the Argonne reactor codes to plot key information. Line printer plots were also used in some of the codes. But now that essentially all computations are performed on PC's, the RERTR project has initiated an effort to standardize on a few graphics packages for off-line analysis. The *Mathematica* software was used to program (outside of REBUS) the cubic spline algorithms and the polynomial fit algorithms for the purposes of comparison, and to assist in debugging of new FORTRAN-77 coding in REBUS-5. Experience is being gained as to the relative advantages and disadvantages of polynomial fits versus spline fits. There is essentially

no difference in running time, since the computational time involved in processing burnup-dependence is only a few tenths of a percent of the total job time.

Table I provides a comparison of eigenvalues obtained by polynomial fits versus a preliminary implementation of spline fits, for the BMRR. For this reactor, the effects of burnup-dependent cross sections are not large. Either interpolation method yields practically the same results. The reactivity differences shown are a measure of the reactivity uncertainty introduced by the cross section fitting technique.

Table I: BMRR Eigenvalue for Each Neutronic Calculation, Using Burnup-Dependent Cross Sections

Time, days	k_{eff} (polynomial)	k_{eff} (spline)	Reactivity difference, 10^{-3}
0	1.053209	1.053262	-0.048
0.33	1.039278	1.039426	-0.137
1.	1.026924	1.026926	-0.002
1.33	1.035560	1.035622	-0.006
2.	1.023934	1.023939	-0.005
2.33	1.034912	1.034982	-0.007
3.	1.023365	1.023376	-0.001
3.33	1.034537	1.034613	-0.007
4.	1.022979	1.022995	-0.002
4.33	1.034165	1.034248	-0.078
7.	1.049590	1.049670	-0.073
7.33	1.037271	1.037350	-0.073

OTHER SOFTWARE CHANGES

The RERTR Project has converted its analysis codes to operate under the Linux operating system on PC's, rather than on SUN workstations. The reasons supporting this change were: the Linux operating system is extremely stable; Linux supports multiprocessing in PVM (needed by MCNP); Lahey's FORTRAN compilers are available in almost identical form for Linux and for WINDOWS; newer PC's are much more cost-effective platforms for reactor analysis than are the SUN workstations; modern PC's no longer are limited by the hardware or software limitations that in the past have restricted the size of problems. The codes are all compiled under the Lahey FORTRAN 95 Linux compiler. Source code is generally at the FORTRAN 77 level, but some successful experimentation has been carried out by automated source conversion to FORTRAN 90. Limitations of previous versions of the Lahey FORTRAN 90 compiler, such as limits on binary record lengths, have been troublesome for some codes. But the FORTRAN 95 compiler ,

in both the Linux and the Windows versions, has been extremely successful and effective. We have had a “memory leak” problem with FORTRAN 95 under Windows, which was overcome by revising the logic used for dynamic allocation and deallocation of core containers. There has been a performance gain in speed of 20-30% from using the FORTRAN 95 compiler versus the FORTRAN 90 compiler.

For many years, some of the power-related edits such as burnup have been slightly inconsistent with the power edits from DIF3D. It was determined that the origin of this problem was that DIF3D and the fuel cycle computational module did not use the same method or data in order to deduce power. DIF3D was correct, while the affected REBUS edits were not fully consistent because updated zone-averaged power conversion factors were not being supplied. This problem was corrected by adding the necessary information to the COMPXS file.

Output edits of REBUS were originally designed for large mainframe computers with 132 character line printers. All output was sent to text files FT06 and to FT10, and to a series of “summary” text files created by the SUMMARY module. This hardware and software environment greatly influenced the design of the code. But now, in a PC environment, the programmer has much greater flexibility. We have taken advantage of this new flexibility by writing specialized output to a multitude of formatted files for review using standard text editors. The length of output files for complex reactor models has become so large that coding changes were necessary to delete redundant data from them as these files are generated. A backspacing control routine was developed to over-write outdated information on many output files during iterations at the same time point, leaving only the final data. The organization of specialized output has also been significantly improved to ease its use by the reactor analyst.

Conversion from one computer platform to another has required extensive validation of prior benchmark calculations. Due to binary file format differences on different hardware platforms, it was necessary to convert neutron cross section libraries from binary to BCD, and back again. Our standard conversion process codes only retained 6 significant figures—not full single precision. Precise benchmarking cannot be achieved under those conditions. All of the cross section conversion codes, both free-standing and in REBUS, were converted to retain full single precision by retaining 8 significant figures in BCD files. The BCD to binary file conversion process within REBUS was revised to detect and process either format without the need for any change to the input file.

A new methodology created by M. Bretscher¹⁰ to represent strongly-absorbing regions containing Gd₂O₃-Al in REBUS/DIF3D burnup calculations was implemented and tested. This approach permitted burnup-dependent boundary conditions, which hitherto had been fixed for all time by the REBUS input processor at job initiation. The problem geometry was conventionally processed (once) into the standard interface file GEODST. Coding logic was changed to permit updating, at each time step, the boundary condition constants stored in GEODST in order to correspond to burnup-dependent changes in the boundary conditions. Quite reasonable results were obtained with this new method when it was compared to MC-REBUS results.

CONCLUSIONS

The needs of the research reactor analyst continue to evolve as reactor designs become more complex. Software for reactor analysis must evolve to meet those needs, as well as to take advantage of enormous advances in operating environments and compilers, and computer power afforded by modern PC's and clusters of PC's. REBUS-5 continues to evolve to meet those challenges. Some aspects of the fuel cycle calculations within REBUS, and of the neutronics solution within DIF3D, could be broken into parallel calculations and sent to a cluster of computers. We look forward to the future possibility that some use of the PVM multiprocessing capability as used in MCNP will find its way into other parts of REBUS.

REFERENCES

1. J. Hoover, G. K. Leaf, D. A. Meneley, and P. M. Walker, "The Fuel Cycle Analysis System, REBUS," Nucl. Sci. Eng. **45**, p. 53, 1971.
2. A. P. Olson, J. P. Regis, D. A. Meneley, and L. J. Hoover, "A User's Manual for the Reactor Burnup System, REBUS," FRA-TM-41, Argonne National Laboratory, September 28, 1972.
3. A. P. Olson, "A User's Manual for the Reactor Burnup System, REBUS-2," FRA-TM-62, Argonne National Laboratory, March 1, 1974.
4. R. P. Hosteny, "The ARC System Fuel Cycle Analysis Capability, REBUS-2," ANL-7721, Argonne National Laboratory, October 1978.
5. B. J. Toppel, "A User's Guide for the REBUS-3 Fuel Cycle Analysis Capability," ANL-83-2, Argonne National Laboratory, 1983.
6. K. L. Derstine, "DIF3D, A Code to Solve One-, Two-, and Three-Dimensional Finite-Difference Diffusion Theory Problems," ANL-83-64, Argonne National Laboratory, 1983.
7. G. Palmiotti, E. E. Lewis, and C. B. Carrico, "VARIANT: VARIational Anistropic Nodal Transport for Multidimensional Cartesian and Hexagonal Geometry Calculation," ANL-95/40, Argonne National Laboratory, October 1995.
8. J. F. Breismeister, ed., "MCNP - A General Monte Carlo N-Particle Transport Code, Version 4B," LA-12625-M. Los Alamos National Laboratory, 1997.
9. N. A. Hanan, A. P. Olson, R. B. Pond, and J. E. Matos, "A Monte Carlo Burnup Code Linking MCNP and REBUS," presented at the 1998 International Meeting on Reduced Enrichment for Research and Test Reactors, October 18-23, Sao Paulo, Brazil.
10. M. M. Bretscher, Private Communication, Argonne National Laboratory, 1999.