GERAPH: A Novel Approach to the General Solution of Pulsed History Activation Problems

P.P.H. Wilson, J.E. Sisolak, D.L. Henderson

June 1994

UWFDM-956

Prepared for the Eleventh Topical Meeting on the Technology of Fusion Energy, June 19–23, 1994, New Orleans LA.

FUSION TECHNOLOGY INSTITUTE

UNIVERSITY OF WISCONSIN

MADISON WISCONSIN

GERAPH: A Novel Approach to the General Solution of Pulsed History Activation Problems

P.P.H. Wilson, J.E. Sisolak, D.L. Henderson

Fusion Technology Institute
University of Wisconsin
1500 Engineering Drive
Madison, WI 53706

http://fti.neep.wisc.edu

June 1994

UWFDM-956

Prepared for the Eleventh Topical Meeting on the Technology of Fusion Energy, June 19–23, 1994, New Orleans LA.

GERAPH: A NOVEL APPROACH TO THE GENERAL SOLUTION OF PULSED HISTORY ACTIVATION PROBLEMS

P. P. H. Wilson, J. E. Sisolak, and D. L. Henderson
Nuclear Engineering and Engineering Physics
University of Wisconsin-Madison
1500 Johnson Drive
Madison, WI 53706-1687
(608) 265-4573

ABSTRACT

A new numerical method using generalized eigenvector decomposition for the solution of pulsed activation is described and tested. This method allows for the treatment of multiple occurences of isotopes in the decay scheme including the treatment of an arbitrary number of loops. Preliminary comparisons with existing activation codes show good agreement and confirm that the method produces accurate results.

I. BACKGROUND

The problem of determining the neutron activation in the first wall, blanket and shield of a fusion reactor vessel can be an intricate one. In even the simplest case, a large system of linear first-order ordinary differential equations will result, becoming more complicated with improved libraries and different neutron flux histories. Perhaps the most interesting complication is that of a pulsed or intermittent operation mode. GERAPH [Generalized Eigenvector Radioactivity Analysis for Pulsed Histories] is a new code that calculates the activation levels in such cases.

When a reactor has a pulsed flux history, decay of radioactive isotopes competes with neutron transmutation during the high flux periods while only decay occurs during the zero flux dwell periods. Two steady-state approximations that can be used to calculate induced radioactivity conserve fluence, but one conserves the flux level (generally used in magnetic fusion analysis) while the other conserves reactor lifetime (generally used in inertial fusion analysis). If we consider, for example, the first method, the approximation is valid for short lived radioactive isotopes; during each pulse, they reach secular equilibrium values and during each dwell time, they decay to nothing. For a similar reason, long-lived isotopes are not greatly affected by intermittent operation; during each pulse, they continue to build towards an equilibrium level, and during each dwell period there is little effect on their level. For this reason, the induced radioactivity due to these isotopes can be accurately calculated by using a steady-state approximation which conserves fluence and flux. However, for medium lived radioisotopes, those with a half-life of the same order as the dwell time, the effect is significant. They reach some appreciable level during the pulse and decay measurably during the dwell time. Sisolak, et al.,[1] showed that the difference between a pulsed solution and both steady-state approximations can be quite severe. The first method generally results in an overestimation of radiaoctivity while the second underestimates.

II. HISTORY

Many approaches can be used to perform this "accurate" pulsed solution, but traditionally, three have seen widespread use. The most straightforward is a method which breaks the decay and transmutation scheme into linear chains and solves the system using the Bateman equation.[2] Such solution schemes allow matrix type linear solutions for each pulse which can then be raised to a power to represent the many different pulses.[3] The result is a fast, analytically exact solution which is accurate and dependable for many systems, but has problems when the decay and transmutation scheme contains loops. Loops occur when reactions such as:

$$A \xrightarrow{(n,p)} B \xrightarrow{\beta^-} A \qquad \text{or} \qquad C \xrightarrow{(n,\gamma)} D \xrightarrow{(n,2n)} C$$

or their more complex equivalents appear in the reaction path. The analytical solution analogous to the Bateman equation for the general "loop" case is difficult to formulate and implement.

The two other methods rely on using a discrete time solution to the set of differential equations. Standard ODE solvers such as derivatives of the GEAR and Runge-Kutta methods are implemented by separating the complete time history into small increments of time and solving the system of equations on each time interval, using the results of the previous interval as the initial conditions of the next. These methods can handle loops exactly and can produce very accurate results, but when the history becomes long and intricate, these methods take a prohibitively long time to reach a solution.

The other discrete time-step method, known as the Matrix Exponential Method (MEM), takes note of the fact that the solution to a system of equations $\dot{\vec{N}} = \boldsymbol{A} \cdot \vec{N}$ is $\vec{N}(t) = e^{\boldsymbol{A} \cdot t} \cdot \vec{N}_o$. The solution is obtained by approximating the matrix exponential with its series expansion over a time step Δt. For the series expansion to be accurate, small time steps or many terms are required, both resulting in many matrix multiplications. This solution suffers from the same run-time problems as the ODE solvers and from accuracy problems related to the finite approximation to the infinite exponential series. The new method examined here hopes to eliminate all these problems by combining the speed of a matrix solution with the accuracy of a solution which allows for loops.

Another basic method to solve the matrix exponential is to use a matrix decomposition, an example of which is eigenvector decomposition. If $\boldsymbol{P}$ is the matrix whose columns are the eigenvectors of $\boldsymbol{A}$ and $\boldsymbol{\Lambda}$ is the diagonal matrix with the eigenvalues of $\boldsymbol{A}$ as its elements,

$$\begin{aligned}
\boldsymbol{A} &= \boldsymbol{P} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{P}^{-1} \\
\therefore e^{\boldsymbol{A} \cdot t} &= \boldsymbol{I} + \boldsymbol{A}t + \frac{\boldsymbol{A}^2 t^2}{2!} + \ldots \\
&= \boldsymbol{P}(\boldsymbol{I} + \boldsymbol{\Lambda}t + \boldsymbol{\Lambda}^2 \frac{t^2}{2!} + \ldots)\boldsymbol{P}^{-1} \\
&= \boldsymbol{P} \cdot e^{\boldsymbol{\Lambda} \cdot t} \cdot \boldsymbol{P}^{-1} .
\end{aligned}$$

This particular decomposition is plagued by the problem of loops in the reaction tree since loops result in a defective matrix, $\boldsymbol{A}$. There are methods, such as Schur decomposition,[4] that overcome this problem.

Braun,[5] Boyce and DiPrima,[6] and Redheffer[7] all provide matrix method solutions to this problem designed specifically for the solution of a set of linear ODE's. The latter results in the necessity to solve the system $\boldsymbol{D} \cdot \vec{\boldsymbol{M}} = \vec{\boldsymbol{C}}$ where not only is $\boldsymbol{D}$ defective, but $\vec{\boldsymbol{M}}$ and $\vec{\boldsymbol{C}}$ are vectors of matrices. The former two methods are similar in their outcome and resemble a Schur decomposition in their final formulation, but approach the problem from different angles. The method implemented here is a variation on that presented in Braun.[5]

III. CODE DESIGN

There are three significant areas in which the methodology of this approach is different from previous implementations: code design, problem understanding and completeness, and mathematical method. In addition, like other codes, this one supports multidimensional inhomogenous systems in a number of different geometries, relying on flux information generated by currently established neutron transport codes.

Unlike most engineering codes in the past, this one was implemented using C++, allowing interesting philosophies for both data structure and software design. The main impetus for this decision was the desire to take advantage of dynamic memory allocation with complex data structures, resulting in a system of code that is tailored to the problem and allows complex relationships between data to be handled with simplicity. It is this data structure philosophy that permits the straightforward use of the matrix decomposition methods without the restrictions of linear chains, but rather, with the power of n-ary tree structures. The matrix mathematics can also be optimized to take advantage of the specific characteristics of the solution methodology, such as sparse, lower-triangular matrices. Hopefully, this object-oriented methodology will result in a more easily maintainable and expandable code!

The data structure that is implemented in the code uses an n-ary tree (see Figure 1) for which nomenclature is important. A tree is made up of nodes and branches. In Figure 1, the nodes are represented by the letters of the alphabet, A through J and L, while the branches are the arrows between them. In addition, one often refers to the children of a node, i, which are all those nodes below i in the tree but connected to it by branches. Similarly, a descendant of node i is any node whose production can be traced down branches from node i. The generations are enumerated by the rank of the tree. Finally, if a node has no children, it can also be refered to as a leaf. Thus, the tree in Figure 1 has 22 nodes, 21 branches, and 8

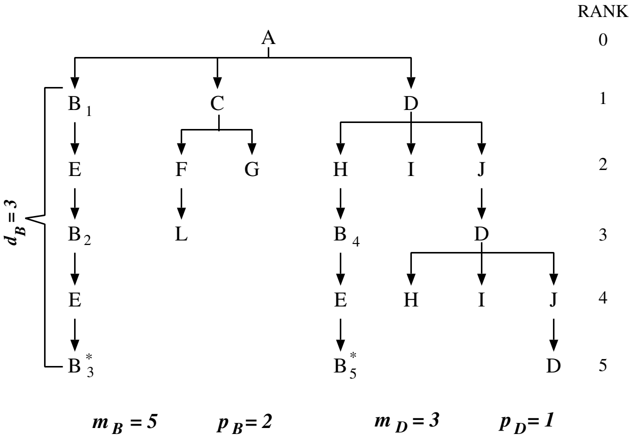

Fig. 1. Sample reaction tree.

leaves.

Each initial isotope will have its own tree, and each node in the tree representing an isotope; however, there may be many nodes in one tree that all represent the same isotope but each having different production paths (see Figure 1). Some of these multiple occurrences will be due to production of the isotope from different sources (B_1 vs. B_4 in Figure 1), while others will be due to "straightening" of loops in the reaction path (B_1, B_2 and B_3 in Figure 1). The result of having different nodes for the same isotope due to different reaction paths is that the matrix, $\boldsymbol{A}$, will have no more than two non-zero elements in each row: a single production rate (λ_{parent} or $\sum_{g=1}^{G} \sigma_{r,g,parent}\phi_g$ where G is the number of neutron energy groups) and a single destruction rate ($\lambda + \sum_{g=1}^{G} \sum_{r=1}^{R} \sigma_{r,g}\phi_g$ where R is the number of neutron interactions for that isotope), making matrix operations very simple and fast.

In implementing a solution to this problem, it was important to gain a complete understanding of the problem, the range of possible input parameters, and the limiting intricacies of the solution methodology, so that the code can be applied to the widest variety of cases without an appreciable loss of accuracy or increase in run-time. While gaining an understanding of the problem, a number of minor complications arose. The data libraries being used to generate the isotope tree have been updated to include up to 55 different neutron interactions, resulting in prohibitively large problems. A set of appropriate truncation rules had to be created to keep the problem within a reasonable size and complexity without losing accuracy of the solution. A point exists after which increasing the problem size has negligible effect on the solution accuracy. A simple maximum tree depth criterion is available to artificially limit the size of the problem; however, the standard method of truncation relies on doing an approximate calculation of the relative production level of the isotope in question. For each node, a conservative approximation of the production of that isotope is performed and compared to a user-defined tolerance. If this comparison indicates that this isotope should be a point of truncation, there will be no isotopes below that rank that are descendants of that node. There may, however, be nodes lower that this rank in other subtrees of the main tree. If the optional maximum tree depth is specified by the user, the relative production level rule is applied to all isotopes above the maximum depth. In Figure 1, utilizing the standard truncation method, isotopes G, L, and I were truncated immediately because their production rate is too low. On the other hand, isotope H was expanded once, but on its next occurrence, it was truncated. Finally, isotopes such as B and E were expanded many times, even though they represent loops.

Also, to ensure that the results would include a wide range of information useful to many, the tracking of light ion production (^{1}H, ^{2}H, ^{3}He, ...) as by-products of neutron interactions and decay was incorporated. This information is important since tritium can make a major contribution to the induced radioactivity in a reactor vessel and the light ion production has significant effects on its material properties. Automatic truncation occurs for all light ion reaction by-products except ^{3}H and ^{3}He, under the philosophy that allowing the ions to undergo reactions would cause an increase in problem size not countered by the corresponding gain in accuracy or information.

The code was designed to encourage use by employing a descriptive input format and allowing a broad range of input problems. The input file is fully commented, which permits the user to have full understanding of every inputted piece of information. The support of point, slab, cylindrical, spherical and toroidal geometries in 0 through 3 dimensions and the existence of interface programs to allow communication via existing transport code output make it a widely useful tool.

IV. SOLUTION METHODOLOGY

This section focuses on the mathematical engine which forms the basis of the solution. In so doing, it will be necessary to discuss the data structures or tree creation methods, but they will not be explained in detail.

Assuming that a basis, $\boldsymbol{V} = \{\vec{v}_0, \vec{v}_1, \ldots, \vec{v}_n\}$, for our problem space has been determined, the solution would go as:

$$\begin{aligned}
\dot{\vec{N}} &= \boldsymbol{A}\vec{N} \\
\vec{N}_o &= n_0\vec{v}_0 + n_1\vec{v}_1 + \ldots + n_k\vec{v}_k = \boldsymbol{V}\vec{n} \\
\therefore \vec{n} &= \boldsymbol{V}^{-1}\vec{N}_o \\
\vec{N}(t) &= e^{\boldsymbol{A}t}\vec{N}_o = e^{\boldsymbol{A}t}\boldsymbol{V}\vec{n} \\
&= \sum_{i=0}^{k} e^{\lambda_i t} e^{(\boldsymbol{A}-\lambda_i\boldsymbol{I})t}\vec{v}_i\, n_i\,.
\end{aligned} \tag{1}$$

If we let

$$\boldsymbol{q}_i = e^{(\boldsymbol{A}-\lambda_i\boldsymbol{I})t} \tag{2}$$

and $\boldsymbol{Q} = \{\boldsymbol{q}_0 \cdot \vec{v}_0, \boldsymbol{q}_1 \cdot \vec{v}_1, \ldots, \boldsymbol{q}_k \cdot \vec{v}_k\}$ then Equation (1) becomes:

$$\begin{aligned}
\vec{N}(t) &= \sum_{i=0}^{k} e^{\lambda_i t}\boldsymbol{q}_i\vec{v}_i\, n_i \\
&= \boldsymbol{Q} \cdot e^{\boldsymbol{\Lambda}t} \cdot \vec{n} = \boldsymbol{Q} \cdot e^{\boldsymbol{\Lambda}t} \cdot \boldsymbol{V}^{-1} \cdot \vec{N}_o
\end{aligned} \tag{3}$$

where $\boldsymbol{\Lambda}$ is a diagonal array with the elements being the eigenvalues of $\boldsymbol{A}$.

With careful analysis of this matrix exponential expansion, comparison to the Schur decomposition method becomes interesting. If we multiply a vector, $\vec{v}_i$, by the characteristic equation, $\boldsymbol{A} - \lambda_i\boldsymbol{I}$, we will get a linear combination of the other vectors in the set $\boldsymbol{V}$. In fact, it will be a linear combination of the vectors which correspond to all the occurrences of isotope i below the occurrence which corresponds to $\vec{v}_i$. Thus, it is possible to perform this successive multiplication by the characteristic equation and get a set of vectors $\boldsymbol{W}$, all of whose members are the *generalized eigenvectors* for λ_i. When we do this, the final solution takes the form $e^{\boldsymbol{A}t} = \boldsymbol{W}\boldsymbol{E}\boldsymbol{W}^{-1}$ where $\boldsymbol{E}$ is a lower triangular matrix which contains the powers of t and scalar exponentials of the eigenvalues in the locations to form the correct sum. If $\boldsymbol{E}$ is interpreted as the exponential $e^{\boldsymbol{T}t}$ of some lower triangular matrix, $\boldsymbol{T}$, this would be identical to a Schur decomposition. With the implementation in use by GERAPH, however, the computation of a lower triangular matrix exponential is avoided, eliminating the complications of such an approach.[8] This can be done for any system of linear first-order ODE's which can be represented in an n-ary tree structure.

The final step, and perhaps the most important, is to choose a set of basis vectors, $\boldsymbol{V}$, that allow for an exact solution of the exponential in Equation (2). From Braun, we see that for a degenerate eigenvalue, λ_i, with multiplicity, m_i, although there may be fewer than m_i unique eigenvectors there are always m_i linearly independent vectors that satisfy $(\boldsymbol{A} = \lambda_i\boldsymbol{I})^{d_i}\vec{x} = \vec{0}$ for some power, d_i. Of these vectors, p_i (the dimension of the eigenspace) will be eigenvectors and the others will be linear combinations of the generalized eigenvectors. Thus, if one can determine d_i, $\boldsymbol{q}_i \cdot \vec{v}_i$ reduces to:

$$\begin{aligned}
e^{(\boldsymbol{A}-\lambda_i\boldsymbol{I})t}\vec{v}_i &= \sum_{j=0}^{\infty} \frac{(\boldsymbol{A}-\lambda_i\boldsymbol{I})^j t^j}{j!} \cdot \vec{v}_i \\
&= \sum_{j=0}^{d_i-1} \frac{(\boldsymbol{A}-\lambda_i\boldsymbol{I})^j t^j}{j!} \cdot \vec{v}_i\ .
\end{aligned} \tag{4}$$

At this stage, it is useful to consider the data structure and formulation of the problem. The combination of this exact expansion of the matrix exponential and the sparse matrices resulting from the data structure creation provides improvements in speed and accuracy over other methods that use the series expansion of the matrix exponential. The reason that degenerate eigenvalues exist is that certain isotopes will exist more than once in the isotope tree. The number of distinct eigenvectors, p_i, can be determined by counting the number of occurrences of isotope i that have no occurrences of i below them in the tree. These occurrences of isotope B are marked with an '*' in Figure 1.

The next interesting result is that d_i corresponds exactly to the number of nodes that represent isotope i, in the linear chain that has the most occurrences of isotope i. Therefore, by doing a recursive depth first analysis of the tree, we can obtain all the important information, such as multiplicity m_i, maximum depth d_i, and the dimension of the eigenspace p_i, giving us all the necessary components to generate a mathematically exact solution using Equations (3) and (4) (see Figure 1).

Finally, there is a complication involved when evaluating $e^{\boldsymbol{\Lambda}t}$ in instances when $\lambda_i t$ is a large number. If $\boldsymbol{E} = e^{\boldsymbol{D}t}$ for some diagonal matrix, $\boldsymbol{D}$, then $\boldsymbol{E}$ is a diagonal matrix with elements $e_{ii} = e^{d_{ii}t}$. If $\lambda_i t$ is too large (very short half-lives), the exponential term on that diagonal will be zero, essentially eliminating the information about that isotope. To overcome this problem, the solution matrix $\boldsymbol{Q}e^{\Lambda t}\boldsymbol{V}^{-1}$ is evaluated with some time step, Δt, and this solution is raised the power $\frac{t}{\Delta t}$. Methods which rely on the identity $e^{s\boldsymbol{A}} = (e^{s\boldsymbol{A}/m})^m$ can experience serious problems with roundoff errors in some cases,[8] and as such, new methods are currently being investigated to circumvent this problem.

V. RESULTS

Results from GERAPH were compared with results from the analytically verified demonstration code PULSAR[3] using linear chain construction from DKR-ICF. Five different pulsing histories were used with two different input elements, manganese and iron, and in both cases, a number of radioactive isotopes were chosen for comparison. The cross-section and decay data for both codes were extracted from the USACT93[9] data library.

Table I shows the irradiation history used for the verification cases. These histories were chosen as typical intermittent operation histories for an experimental magnetic fusion energy reactor, such as ITER. In particular, cases 2 through 4 represent a 5 day week with no operation on the weekends. The last case was chosen because it demonstrates the effects of different pulsed operation histories more severely.

Table I. Pulsed History Used for Verification

		Level 1		Level 2		Level 3	
	Width t_p	Dwell Δt_1	# n_1	Dwell Δt_2	# n_2	Dwell Δt_3	# n_3
1	2300 s	200 s	50960	–	–	–	–
2	2300 s	786 s	140	48 h	364	–	–
3	2300 s	786 s	13	13.1 h	5	61.1 h	784
4	2300 s	1 h	13	3.7 h	5	51.7 h	784
5	600 s	360 s	50960	–	–	–	–

The truncation routine used a reference operation time which was equal to the sum of the irradiation times for the entire operation history (i.e.: for case 1, $t_{ref} = 2300\text{s} \times 50960 = 1.17208 \times 10^8\text{s}$), and a truncation tolerance of 4×10^{-7}. This resulted in the inclusion of all the children of ^{55}Mn and the expansion to the third rank of one of those children, resulting in 32 children with 29 leaves. In the iron case, the 4 naturally occurring isotopes were the roots for 4 trees with 48, 31, 31, and 48 nodes and 45, 28, 29, and 45 leaves, respectively.

The results for the manganese and iron case, are shown in Table II. Four isotopes were chosen in each case as major contributors to the final radioactivity.

The difference between these values is most probably due to the fact that the input for PULSAR is derived from the output of DKR, which has been rounded off for output. This round-off error of the input results in the difference of less than 1% for all cases. In addition, the possible existence of loops could cause a difference, since the DKR/PULSAR combination uses the Bateman solution and is unable to handle loops correctly. Given this, the agreement between GERAPH and PULSAR is sufficient to verify the generalized eigenvector decomposition method.

Table II. Activity at Shutdown [Bq]

Parent: Mn			
^{55}Cr, $t_{1/2} = 3.55m$			
Case	GERAPH	PULSAR	% Difference
1	9.681e6	9.617e6	0.67
2	9.679e6	9.615e6	0.67
3	9.679e6	9.615e6	0.67
4	9.679e6	9.615e6	0.67
5	8.715e6	8.656e6	0.68
^{54}Mn, $t_{1/2} = 312.5d$			
Case	GERAPH	PULSAR	% Difference
1	7.155e7	7.106e7	0.69
2	4.303e7	4.274e7	0.68
3	2.002e7	1.988e7	0.70
4	2.002e7	1.989e7	0.65
5	3.615e7	3.590e7	0.70
^{56}Mn, $t_{1/2} = 2.58h$			
Case	GERAPH	PULSAR	% Difference
1	2.165e10	2.149e10	0.75
2	1.791e10	1.778e10	0.73
3	1.701e10	1.692e10	0.5
4	1.033e10	1.025e10	0.8
5	1.480e10	1.469e10	0.75
^{55}Fe, $t_{1/2} = 2.7y$			
Case	GERAPH	PULSAR	% Difference
1	4.893e2	4.858e2	0.72
2	4.025e2	3.996e2	0.72
3	2.612e2	2.594e2	0.69
4	2.613e2	2.595e2	0.69
5	4.012e1	3.984e1	0.70
Parent: Fe			
^{51}Cr, $t_{1/2} = 27.7d$			
Case	GERAPH	PULSAR	% Difference
1	7.691e5	7.700e5	0.1
2	4.567e5	4.571e5	0.09
3	2.130e5	2.133e5	0.1
4	2.121e5	2.123e5	0.09
5	5.225e5	5.231e5	0.1
^{56}Mn, $t_{1/2} = 2.58h$			
Case	GERAPH	PULSAR	% Difference
1	1.510e7	1.512e7	0.1
2	1.249e7	1.251e7	0.2
3	1.189e7	1.191e7	0.2
4	7.204e6	7.217e6	0.18
5	1.032e7	1.033e7	0.1
^{55}Fe, $t_{1/2} = 2.7y$			
Case	GERAPH	PULSAR	% Difference
1	1.611e8	1.611e8	0.00
2	1.208e8	1.209e8	0.08
3	6.603e7	6.605e7	0.03
4	6.605e7	6.608e7	0.05
5	5.554e7	5.553e7	0.02
^{59}Fe, $t_{1/2} = 44.6d$			
Case	GERAPH	PULSAR	% Difference
1	6.093e6	6.081e6	0.2
2	3.585e6	3.578e6	0.2
3	1.668e6	1.665e6	0.2
4	1.664e6	1.661e6	0.2
5	4.139e6	4.130e6	0.2

To show the importance of the truncation tolerance, three additional manganese cases were run to compare to case 1 in Table II with truncation tolerances of 4×10^{-5}, 4×10^{-9}, and 4×10^{-11}. For the case when the tolerance was 4×10^{-5}, the reaction tree was truncated before the inclusion of ^{55}Fe, but in all cases the activities were unaffected by the change in truncation tolerance because the additional contributions to the activities by isotopes occurring later in the tree are negligibly small. However, all cases with lower tolerance had significant increases in problem size, up to 171 nodes and 154 leaves, resulting in a longer run-time. This validates the earlier claim that *appropriate* truncation of the reaction tree does not jeopardize the results.

VI. SUMMARY

GERAPH has been verified against a previously verified code, PULSAR, which uses a different numerical method. The results for the test cases which were conducted show agreement within 0.75%. In addition the demonstration of the importance of the truncation tolerance showed that if set too high, information will be lost, and if set too low, no new information is gained.

Two goals were specified for this new code: speed and accuracy. Unfortunately, no comparison exists for both of these criteria. DKR-ICF is presently under modification so that it will be able to be used as a comparison for multi-level pulsed histories, but currently does not handle loops exactly or account for the feedback of loops. Currently, GERAPH's runtime is on the order of DKR-ICF (within 10%), but that is prior to generating the pulsed solution with PULSAR, and thus the goal of increased speed has been realized for these sample problems.

Future work will consist of verification of the loop results against analytical solutions and the conversion of GERAPH from a demonstration code to a releasable product. Some of this work will include modification of the mathematical engine to optimize the speed and remove the use of the scaling and exponentiation of the matrices.

VII. REFERENCES

1. J.E. Sisolak, S.E. Spangler, and D.L. Henderson, "Pulsed/Intermittent Activation in Fusion Energy Reactor Systems," *Fusion Tech.* **21**, 2145 (1992).

2. D.L. Henderson and O. Yasar, "DKRICF: A Radioactivity and Dose Rate Calculation Code Package," UWFDM-714, University of Wisconsin Fusion Technology Institute, Vol. 1, Madison, Wisconsin (1986).

3. S.E. Spangler, J.E. Sisolak, and D.L. Henderson, "Calculational Models for the Treatment of Pulsed/Intermittent Activation Within Fusion Energy Devices" *Fusion Engineering & Design,* **22**, 349 (July 1993).

4. B. Noble and J.W. Daniel, *Applied Linear Algebra,* pp. 326-327, Prentice-Hall, Englewood Cliffs, New Jersey (1988).

5. M. Braun. *Differential Equations and Their Applications,* pp. 463ff, Springer-Verlag, New York, New York (1975).

6. W.E. Boyce and R.C. DiPrima, *Elementary Differential Equations and Boundary Value Problems, 3^{rd} ed.,* pp. 317ff, John Wiley & Sons, New York, New York (1977).

7. R. Redheffer, *Differential Equations: Theory and Applications,* pp. 388ff, Jones and Bartlett, Boston (1991).

8. C. Moler and C. Van Loan, "Nineteen Dubious Ways to Compute the Exponential of a Matrix," *SIAM Review,* **20**, 801 (Oct. 1978).

9. F. Mann, HEDL, personal communications.