

# A REFINED WAY OF SOLVING REACTOR POINT KINETICS EQUATIONS FOR IMPOSED REACTIVITY INSERTIONS

by

**Barry D. GANAPOL**

Received on November 11, 2009; accepted in revised form on November 24, 2009

We apply the concept of convergence acceleration, also known as extrapolation, to find the solution of the reactor kinetics equations (RKEs). The method features simplicity in that an approximate finite difference formulation is constructed and converged to high accuracy from knowledge of the error term. Through the Romberg extrapolation, we demonstrate its high accuracy for a variety of imposed reactivity insertions found in the literature. The unique feature of the proposed algorithm, called RKE/R(omberg), is that no special attention is given to the stiffness of the RKEs. Finally, because of its simplicity and accuracy, the RKE/R algorithm is arguably the most efficient numerical solution of the RKEs developed to date.

*Key words: nuclear reactor kinetics, Romberg extrapolation, reactivity insertion, thermal reactor, fast reactor*

## INTRODUCTION

An ever popular numerical pastime over the past 40 years has been the development of numerical solutions to the equations of nuclear reactor kinetics (RKEs). One need only survey the literature to find articles with titles such as – *A New Solution...*[1], *Resolution of the Stiffness...*[2], *Generalized Runge-Kutta Methods...*[3], *Efficient Numerical Solution of...*[4], *On Pade Approximations...*[5], *COntstant REactivity: A Numerical Algorithm...*[6], *Revisiting the Rosenbrock Numerical Solutions...*[7], *An Efficient Code System...*[8], and references to additional proposed algorithms (See [1-15]). Apparently, many of the proposed algorithms are constructed specifically to treat the stability and stiffness of the RKEs. All of us in the nuclear community have been particularly sensitized to the notion that these equations are stiff, which, of course, is a consequence of

the prompt neutron generation time being orders of magnitude less than the time for delayed neutrons to appear. While this is certainly of concern, in general, it has led to an unnecessary preoccupation with stiffness and stability. We see this in the creation of ad-hoc time step controls and change of dependent variables attempting to compensate for stiffness. Such considerations have made previous algorithms more complicated than necessary and, as a result, classroom unfriendly, except perhaps at the advanced level.

The theme of this presentation is simplicity and it is dedicated to answering the question: Can an elementary finite difference scheme give a highly accurate numerical solution to the RKEs? In this respect, we consider high accuracy to be at least five significant figures. As will be shown, our approach, which does not explicitly consider stiffness, is an efficient way to define highly accurate numerical solutions to ODEs. We substantiate this claim through exhaustive comparison to the benchmarks found in literature. In this way, we hope to achieve a goal of “sustainable accuracy” – that is, uniformly high accuracy regardless of application. To accomplish this, we first admit that discretization error is a natural element of the solution. We then show how to manage this error in a way that generates a reliable and robust algorithm.

Please note that, because of space limitation, we consider only imposed reactivity insertions and leave the issue of temperature feedback to a future effort.

Nuclear Technology & Radiation Protection  
Scientific paper  
UDC: 539.125.52:519.254  
DOI: 10.2298/NTRP0903157G

Author's address:  
Department of Aerospace and Mechanical Engineering,  
University of Arizona  
Tucson, AZ, 85721, USA

Author's e-mail:  
ganapol@cowboy.ame.arizona.edu

**THE EQUATIONS OF POINT KINETICS**

We begin with the following inhomogeneous, 6-delayed group point kinetics equations (in usual notation):

$$\frac{dy(t)}{dt} = A(t)y(t) + q(t) \tag{1a}$$

with vectors

$$y(t) = \begin{pmatrix} N(t) \\ C_1(t) \\ \dots \\ C_6(t) \end{pmatrix}, \quad q(t) = \begin{pmatrix} q(t) \\ 0 \\ \dots \\ 0 \end{pmatrix} \tag{1b}$$

and Jacobian matrix

$$A(t) = \begin{pmatrix} \frac{\beta}{\Lambda}[\rho_s(t) - 1] & \lambda_1 & \lambda_2 & \dots & \lambda_6 \\ \frac{\beta_1}{\Lambda} & \lambda_1 & 0 & & 0 \\ \dots & 0 & \lambda_2 & & \dots \\ & & & \dots & 0 \\ \frac{\beta_6}{\Lambda} & 0 & \dots & 0 & \lambda_6 \end{pmatrix} \tag{1c}$$

We solve eq. (1a), subject to initial conditions, normalized to a neutron density of one at time zero

$$y(0) = \begin{pmatrix} 1 \\ \frac{\beta_1}{\lambda_1 \Lambda} \\ \dots \\ \frac{\beta_6}{\lambda_6 \Lambda} \end{pmatrix} \tag{1d}$$

In our formulation, the reactivity,  $\rho_s(t)$ , measured in units of  $\beta$ 's, will be prescribed.

**DEVELOPMENT OF A SIMPLE FORWARD FINITE DIFFERENCE SCHEME**

Equation (1a) is discretized by integrating over the time interval  $[t_j, t_{j+1}]$ ,

$$y(t_{j+1}) - y(t_j) = \int_{t_j}^{t_{j+1}} dt A(t)y(t) + \int_{t_j}^{t_{j+1}} dt q(t) \tag{2}$$

We assume a uniform interval,  $h = t_{j+1} - t_j$  over the total time of the domain of interest. After the application of the trapezoidal rule approximation for the integrals in eq. (2), the following two-point discretization emerges:

$$I \frac{h}{2} A_{j+1} y_{j+1} - I \frac{h}{2} A_j y_j = \frac{h}{2} [q_{j+1} + q_j] \tag{3a}$$

where

$$\begin{matrix} A_j & A(t_j) \\ q_j & q(t_j) \end{matrix} \tag{3b}$$

and  $I$  is the unit matrix. In eq. (3a), we have replaced the exact solution  $y(t_j)$  with the approximate solution  $y_j$  where

$$y(t_j) = y_j + e_j \tag{4}$$

and  $e_j$  is the error vector of the approximate solution. The finite difference scheme of eq. (3a) serves as the heart of the development to follow. Note that a matrix inversion is required for the solution of eq. (3a) at each time step which we chose to do by the LU decomposition. An alternative analytical inversion is possible using the procedure developed in ref. [1]; however, the efficiency of the LU decomposition is sufficient for all benchmarks considered.

**Error term**

In the scheme to follow, it is important to determine the form of the error vector  $e_j$ . To accomplish this, we integrate eq. (1a) over the interval  $[0, t_j]$  to give

$$y(t_j) - y(0) = \int_0^{t_j} dt A(t)y(t) + \int_0^{t_j} dt q(t) \tag{5}$$

We then sum eq. (3a) over  $j$  to  $j-1$

$$y_j - y_0 = \frac{h}{2} \sum_{j=0}^{j-1} [A_{j+1} y_{j+1} - A_j y_j] + \frac{h}{2} \sum_{j=0}^{j-1} [q_{j+1} + q_j] \tag{6}$$

If

$$I(f;0,t_j) = \int_0^{t_j} dt f(t) \tag{7a}$$

and the trapezoidal rule is

$$T(f;0,t_j) = \frac{h}{2} \sum_{j=0}^{j-1} [f_{j+1} + f_j] \tag{7b}$$

then, from the Euler-Maclaurin sum formula [16]

$$I(f;0,t_j) - T(f;0,t_j) = \sum_{k=1}^{\infty} \frac{B_{2k}}{(2k)!} [f^{(2k-1)}(t_j) - f^{(2k-1)}(0)] h^{2k} \tag{8}$$

In this equation,  $B_{2k}$  is the Bernoulli number of order  $2k$ . Subtracting eqs. (5) and (6), and noting that  $y(0) = y_0$ , we find

$$y(t_j) - y_j = I(Ay;0,t_j) - T(Ay;0,t_j) + I(q;0,t_j) - T(q;0,t_j)$$

and with eq. (8)

$$e_j = \sum_{k=1}^{\infty} \frac{B_{2k}}{(2k)!} \{[(A\mathbf{y})^{(2k-1)}(t_j) - (A\mathbf{y})^{(2k-1)}(0)] [q^{(2k-1)}(t_j) - q^{(2k-1)}(0)]\} h^{2k} \quad (9)$$

Hence, the error term is formally of the form

$$e_j = \sum_{k=1}^{\infty} a_{jk} h^{2k}$$

and the exact solution is

$$\mathbf{y}(t_j) = \mathbf{y}_j + \sum_{k=1}^{\infty} a_{jk} h^{2k} \quad (10)$$

The particular form of the error term indicates that the finite difference scheme is of second order. In addition, it suggests the application of the Romberg convergence acceleration [17]. In this algorithm, error orders,  $h^{2k}$ ,  $k = 1, 2, \dots$ , are sequentially eliminated by halving the interval  $h$  with each grid refinement representing a discretized solution of eq. (1a) through eq. (3a). Thus, rather than choose a single discretization to be sufficiently accurate by some *ad-hoc* measure, we consider a series of discretizations in a systematic way to give a solution that, in principle, extrapolates to zero discretization. The solution, therefore, is now a sequence of solutions tending toward their limit of zero discretization. Therefore, in applying the Romberg acceleration, no longer will just one discretization be the desired solution, since the solution now becomes an extrapolation of a sequence of solutions on a prescribed path in the discretized time domain. This procedure gives a new definition to the numerical solution of the RKEs.

### ROMBERG ACCELERATION

As indicated above, the Romberg acceleration [17] uses the known form of the error series in eq. (10) to sequentially eliminate higher order error terms. For example, we can write eq. (10) as

$$\mathbf{y}(t_j) = \mathbf{y}_{j,0}(h) + \sum_{k=1}^{\infty} a_{jk,0} h^{2k} \quad (11)$$

where  $\mathbf{y}_{j,0}(h)$  is the original finite difference approximation  $\mathbf{y}_j(h)$  from eq. (3a), and  $j$  now refers to the time  $t_j$  on the initial grid configuration initiated by eq. (1d). Then, eliminating the first term of the error series by considering  $\mathbf{y}_{j,0}(h)$  and the same edit on a grid of half the original interval,  $\mathbf{y}_{j,0}(h/2)$ , simultaneously, one finds

$$\mathbf{y}_{j,1}(h) = \frac{2^2 \mathbf{y}_{j,0}(h/2) - \mathbf{y}_{j,0}(h)}{2^2 - 1} \quad (12)$$

as the next highest order approximation. The flux representation for this new approximation becomes

$$\mathbf{y}(t_j) = \mathbf{y}_{j,1}(h) + \sum_{k=2}^{\infty} a_{jk,1} h^{2k} \quad (13)$$

Continuing to eliminate higher orders in this fashion, sequentially, gives the following recurrence relation for increasingly higher order approximations:

$$\mathbf{y}_{j,m}(h) = \frac{2^{2m} \mathbf{y}_{j,m-1}(h/2) - \mathbf{y}_{j,m-1}(h)}{2^{2m} - 1}, m = 1, 2, \dots, \quad (14)$$

and the solution at the original edit, which is now  $t_{2^m j}$  in the refined grid, becomes

$$\mathbf{y}(t_j) = \mathbf{y}_{j,m}(h) + \sum_{k=m+1}^{\infty} a_{jk,m} h^{2k} \quad (15)$$

It should be apparent that the Romberg convergence acceleration applies only to the original time edit which each grid refinement inherits. One way to ensure this is to perform the grid refinement between the requested time edits.

The Romberg acceleration requires that one generate the following sequence of finite difference approximations from eq. (3a):

$$\mathbf{y}_{j,0}(h/2^m), m = 0, 1, 2, \dots, \quad (16)$$

so as to give the Romberg sequence for the recurrence of eq. (14). The Romberg acceleration, therefore, rearranges the original sequence into a more efficiently convergent one. To test for convergence, we therefore have the choice of the original sequence

$$e_0 = \max \left| \frac{\mathbf{y}_{j,0,l}(h/2^m) - \mathbf{y}_{j,0,l}(h/2^{m-1})}{\mathbf{y}_{j,0,l}(h/2^m)} \right| \varepsilon \quad (17a)$$

or the Romberg sequence

$$e_R = \max \left| \frac{\mathbf{y}_{j,m,l}(h) - \mathbf{y}_{j,m-1,l}(h)}{\mathbf{y}_{j,m,l}(h)} \right| \varepsilon \quad (17b)$$

Note that we base convergence on the worst relative error of the seven components of  $\mathbf{y}$  at a desired edit.

### A DEMONSTRATION

The evaluation procedure, say for the edit  $t_1$ , begins with subdividing the interval  $[0, t_1]$  into four equal sub-intervals. Then the recurrence given by eq. (3a) is initially applied to this partition. The grid between the subdivisions is refined by two, until either eq. (17a) or (17b) is satisfied at the edit  $t_1$  only. Note that none of the times between  $[0, t_1]$  are accelerated. For the next

edit, say  $t_2$ , the converged solution at edit  $t_1$  serves as the initial condition for the next interval,  $[t_1, t_2]$ , for which the scheme of eq. (3a) again applies. The evaluation in this interval is complete when either of eqs. (17) are satisfied for edit  $t_2$ . In this way, we cover all edits sequentially. The application of eq. (3a) accelerated by eq. (14) until either of eqs. (17) are satisfied is called the RKE/R(omberg) algorithm, which we now demonstrate through consideration of several benchmark series.

In the demonstrations to follow, error tolerance in eqs. (17) is case dependent and varies between  $10^{-4}$  and  $10^{-7}$ , unless otherwise indicated\*. The coding of the RKE/R algorithm is a mixed FORTRAN/77-95. All computational times are relative to a Gateway, 1.2 GHz laptop.

### Benchmark series 1: step reactivity insertion

For this benchmark series, only step reactivity insertions are considered. The RKE/R method results are compared to well established benchmarks.

#### B1.1: reactivity insertion of \$0.5 into fast reactor I, refs. [1] and [7]

The first benchmark considered is for a step reactivity insertion of \$0.5 into a fast reactor. The neutron generation time is  $\Lambda = 10^{-7}$ s and the kinetics parameters given in tab. 1.

**Table 1. Kinetics parameters for fast reactor I**

$\beta = 0.0044$		
$i$	$\beta_i$	$\lambda_i [s^{-1}]$
1	0.0001672	0.0129
2	0.001232	0.0311
3	0.0009504	0.134
4	0.001443	0.331
5	0.0004534	1.26
6	0.000154	3.21

Table B1.1.1 gives the neutron density from the difference equation at the last grid refinement and the converged neutron density for which 3 edits are requested. In addition, the relative error between the difference and converged densities, the number of discretizations per interval (No of dis.) between edits and the final number of Romberg iterations (Level) are given. The converged densities agree to all of the digits of the reference solution [1] and with the (revisited) Rosenbrock solution of ref. [7], which is one of the most accurate of recent solutions. It is curious that the origin of the accepted reference solution first quoted in

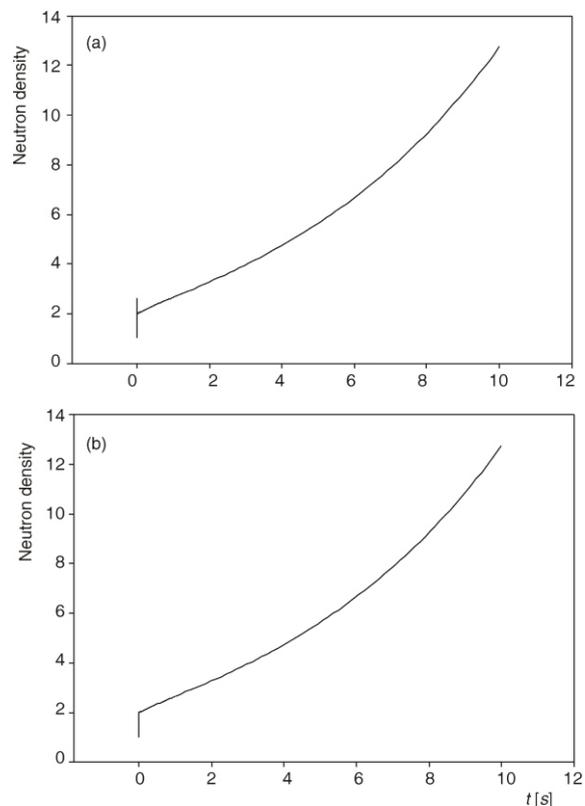
\* In this way, the RKE/R method is comparable to the Rosenbrock algorithm of ref. 7, with regard to the average time step as required by the reviewers

**Table B1.1.1. Step insertion of \$0.5 into fast reactor I**

$t$ [s]	Difference	Converged	Relative error	No of dis.	Level
1.00E-01	2.075317E+00	2.075317E+00	0.00E+00	256	6
1.00E+00	2.655861E+00	2.655853E+00	2.96E-06	32	3
1.00E+01	1.274724E+01	1.274654E+01	5.52E-05	64	4

[1] is unknown and, reassuring that the RKE/R and Rosenbrock algorithms both confirm. We also point out that, for the last two edits, the converged solution is, at least, two digits more accurate (when rounded) than the original difference solution (Difference). This is quite remarkable when one considers that the converged solution is simply a linear combination of finite difference solutions on increasingly refined grids.

Figure 1(a) shows a plot of the unaccelerated neutron density given by the finite difference scheme on the last grid refinement. As one might anticipate, because of the step reactivity jump, the finite difference solution initially oscillates as the derivative attempts to accommodate the abrupt change – observed as the overshoot near the initial time. At the same time, however, since the first desired edit is many neutron generations away from transient initiation, the converged solution is not affected, indicating a stable algorithm. Poor accommodation is also indicated by the relatively large number of discretizations (256) required for convergence in the first interval.



**Figure 1. Benchmark B1.1: step insertion \$0.5 (a) 3 edits; (b) 7 edits**

A particularly appealing feature of the RKE/R algorithm is that it can naturally eliminate oscillations at early times, just through input. This is accomplished by including edits on the order of the neutron generation time. The densities for four additional edits ( $t = 10^{-r}$  s,  $r = 8, 6, 4, 2$ ), included before the first desired edit (0.1 s), are given in tab. B1.1.2. Note that the number of discretizations in the first interval is now only 8. The reduction affects the average (computational) time step to achieve the converged numerical solution, which is the overall time interval of interest divided by the total number of discretizations on the final grid of the converged solution. The average time step is a measure of computational effort used for comparison with other algorithms. For the results of tab. B1.1.1, the average time step is 0.028 s, while by simply adding edits at  $10^{-7}$  s and  $10^{-3}$  s, the average time step increases to 0.096 s. For the achieved accuracy, this is comparable to most algorithms, including at least one with a time step control [7]. However, in this author's opinion, the additional effort required and the complication engendered in an approximate time step control, as enforced in ref. [7], is not worth the savings. In this work, we do not consider optimizing the time step, but rather accept it as a natural part of the solution which we manipulate to gain accuracy and at the same time maintain algorithmic simplicity.

Finally, observing fig. 1(b), we note that the inclusion of additional edits has eliminated the overshoot.

**Table B1.1.2. Step insertion of \$0.5 into fast reactor I**

$t$ [s]	Difference	Converged	Relative error	No of dis.	Level
1.00E-08	1.000220E+00	1.000220E+00	0.00E+00	8	1
1.00E-06	1.021760E+00	1.021760E+00	0.00E+00	8	1
1.00E-04	1.889314E+00	1.889220E+00	4.93E-05	32	3
1.00E-02	2.007681E+00	2.007681E+00	1.00E+00	64	4
1.00E-01	2.075317E+00	2.075317E+00	1.27E-07	16	2
1.00E+00	2.655861E+00	2.655853E+00	2.96E-06	32	3
1.00E+01	1.274724E+01	1.274654E+01	5.52E-05	64	4

The computational time in both cases is less than 0.016 s. Hence, with additional (converged) edits, we obtain increased accuracy with little additional computational effort.

*B1.2: reactivity insertion of \$0.5 into thermal reactor I, ref. [5]*

Reference [5] contains a benchmark for the same insertion into a thermal reactor with nuclear properties of tab. 2 and the resulting densities given in tab. B1.2. Here, the neutron generation time is  $\Lambda = 10^{-5}$  s.

All values agree to all digits of the reference solution. Even though the average time step of 0.1 s is less than that of ref. [5] (0.25 s), no eigenvalues need be found, making the RKE/R solution, overall, that much more convenient.

**Table 2. Kinetics parameters for thermal reactor I**

$\beta = 0.0075$		
$i$	$\beta_i$	$\lambda_i$ [s <sup>-1</sup> ]
1	0.000285	0.0127
2	0.0015975	0.0317
3	0.00141	0.115
4	0.0030525	0.311
5	0.00096	1.40
6	0.000195	3.87

**Table B1.2. Step insertion of \$0.5 into thermal reactor I**

$t$ [s]	Difference	Converged	No of dis.
1.00E+00	2.511520E+00	2.511494E+00	32
5.00E+00	5.753564E+00	5.753393E+00	32
1.00E+01	1.421578E+01	1.421503E+01	32

*B1.3: reactivity insertion of -\$0.5 into thermal reactor I, refs. [1],[5], and [7]*

The next benchmark is for a step reactivity insertion of -\$0.5 into a thermal reactor whose kinetics parameters are given in tab. 2 and the neutron generation time is  $\Lambda = 5 \cdot 10^{-4}$  s (tab. B 1.3).

All digits agree (when rounded) to what seems to be the most accurate reference solution given in refs. [1], [5], and [7], except for the one density highlighted which is off by one unit in the last place. The average time step is 0.125 s, which is larger than the "optimized" time step of 0.093 s of the Rosenbrock algorithm [7]. The computational time for this case is about 0.016 s.

**Table B1.3. Step insertion of -\$0.5 into thermal reactor I**

$t$ [s]	Difference	Converged	No of dis.
1.00E-01	6.988932E-01	6.989252E-01	32
1.00E+00	6.070532E-01	6.070536E-01	64
5.00E+00	4.825523E-01	4.825530E-01	64
1.00E+01	3.960759E-01	3.960777E-01	31

*B1.4: reactivity insertion of \$1 into thermal reactor I, refs. [1] and [7]*

Another benchmark in this series comes from refs. [1] and [7], and represents a stronger insertion than the benchmarks so far. Again, we consider thermal reactor I (tab. 2) for a prompt critical insertion. The results, shown in tab. B1.4, confirm the last digit of the reference solution at edit 1 s to be correct relative to the Rosenbrock algorithm [7] (which is within its desired error). Also, the RKE/R algorithm outperforms the algorithm of ref. [1] with an average time

**Table B1.4. Step insertion of \$1 for thermal reactor I**

$t$ [s]	Difference	Converged	No of dis.
1.00E-01	2.515795E+00	2.515766E+00	16
5.00E-01	1.036488E+01	1.036253E+01	16
1.00E+00	3.218645E+01	3.218354E+01	32

step of 0.02 s. The average time step is the same as for the Rosenbrock algorithm.

*B1.5: reactivity insertion of 0.003, 0.0055, 0.007, and 0.008 into thermal reactor II, refs. [2],[3], and [4]*

We consider a fifth set of benchmarks, for sub-prompt critical step reactivity insertions of 0.003 (\$0.4286), 0.055 (\$0.7857), a prompt critical insertion of \$1 and a super-prompt critical insertion of 0.008 (\$1.1429) into the thermal reactor II whose kinetics parameters are given in tab. 3. The neutron generation time is  $\Lambda = 2 \cdot 10^{-5}$  s. The last insertion represents a true test of the RKE/R scheme, since the neutron density is predicted to increase by 23 orders of magnitude in 1 s.

**Table 3. Kinetics parameters for thermal reactor II**

$\beta = 0.007$		
$i$	$\beta_i$	$\lambda_i [s^{-1}]$
1	0.000266	0.0127
2	0.001491	0.0317
3	0.001316	0.115
4	0.002849	0.311
5	0.000896	1.40
6	0.000182	3.87

Tables B1.5(a)-(d) give the neutron densities for all four cases. Compared to the published reference solution found in ref. [2], we observe that the RKE/R densities agree to all (four) digits when rounded, except for the three highlighted which disagree to one unit in the fourth place. The question which is the correct reference solution (RKE/R or ref. [2]) needs to be addressed. Arguably, all investigations indicate that the RKE/R values are correct to one digit in the last place and that, therefore, one should consider the RKE/R algorithm to be the reference solution. In any case, the RKE/R algorithm easily outperforms the algorithms of refs. [2] and [3] in accuracy and is on a par (for 4-digits only) with that of ref. [4]. Note the relatively large number of discretizations required for the super-prompt case. This reflects the relatively short average time step. The RKE/R method, as indicated above, allows one to, if desired, increase the average time step by including more converged edits within the total interval. In particular, we double the average time step with little increase in computational time by adding 10 edits where the density variation is the greatest.

For completeness, we indicate the average time steps and computational times in tab. 4.

**Table 4. Average time steps for benchmark B1.5**

Case	Average time step [s]	$T_{\text{compt}}$ [s]
a	0.057	0.02
b	0.016	0.02
c	0.0026	0.02
d	0.00024	0.11

**Table B1.5a. Step insertion of 0.003 into thermal reactor II**

$t$ [s]	Difference	Converged	No of dis.
1.00E+01	2.209841E+00	2.209840E+00	256
1.00E+01	8.019434E+00	8.019200E+00	64
2.00E+00	2.830177E+01	2.829740E+01	32

**Table B1.5b. Step insertion of 0.0055 into thermal reactor II**

$t$ [s]	Difference	Converged	No of dis.
1.00E-01	5.210043E+00	5.210028E+00	64
2.00E+00	4.303139E+01	4.302514E+01	64
1.00E+01	1.388834E+05	1.388602E+05	512

**Table B1.5c. Step insertion of 0.007 into thermal reactor II**

$t$ [s]	Difference	Converged	No of dis.
1.0000E-02	4.508927E+00	4.508858E+00	8
5.0000E-01	5.347151E+03	5.345888E+03	256
2.0000E+00	2.062648E+11	2.059156E+11	512

**Table B1.5d. Step insertion of 0.008 into thermal reactor II**

$t$ [s]	Difference	Converged	No of dis.
1.0000E-02	6.203439E+00	6.202854E+00	16
1.0000E-01	1.411195E+03	1.410422E+03	128
1.0000E+00	6.166620E+23	6.163334E+23	4096

*B1.6: reactivity insertion of 0.003, -0.007, and 0.007 into thermal reactor II, ref. [9]*

In this series, we include a benchmark from ref. [9] for thermal reactor II of tab. 3 and a generation time of  $\Lambda = 2 \cdot 10^{-4}$  s. While we have considered some of these insertions above, here they will be for different

**Table B1.6a. Step insertion of 0.003 into thermal reactor II**

$t$ [s]	Referent sol.	Converged	No of dis.
2.00E-01	1.851268E+00	1.851268E+00	128
4.00E-01	1.947593E+00	1.947593E+00	16
6.00E-01	2.037922E+00	2.037922E+00	16
8.00E-01	2.124832E+00	2.124835E+00	16
1.00E+00	2.209841E+00	2.209840E+00	16

**Table B1.6b. Step insertion of -0.007 into thermal reactor II**

$t$ [s]	Referent sol.	Converged	No of dis.
2.00E-01	4.809743E-01	4.809732E-01	128
4.00E-01	4.652903E-01	4.652893E-01	16
6.00E-01	4.519650E-01	4.519640E-01	16
8.00E-01	4.402732E-01	4.402723E-01	16
1.00E+00	4.297830E-01	4.297820E-01	16

**Table B1.6c. Step insertion of 0.007 into thermal reactor II**

$t$ [s]	Referent sol.	Converged	No of dis.
2.00E-01	1.597257E+02	1.597258E+02	64
4.00E-01	1.667286E+03	1.667288E+03	64
6.00E-01	1.713190E+04	1.713193E+04	64
8.00E-01	1.758905E+05	1.758910E+05	64
1.00E+00	1.805726E+06	1.805732E+06	64

time edits. Tables B1.6(a)-(c) give the RKE/R and the reference solution quoted in ref. [9]. Discrepancies in the reference solution (Hermite polynomial algorithm) and RKE/R are quite apparent.

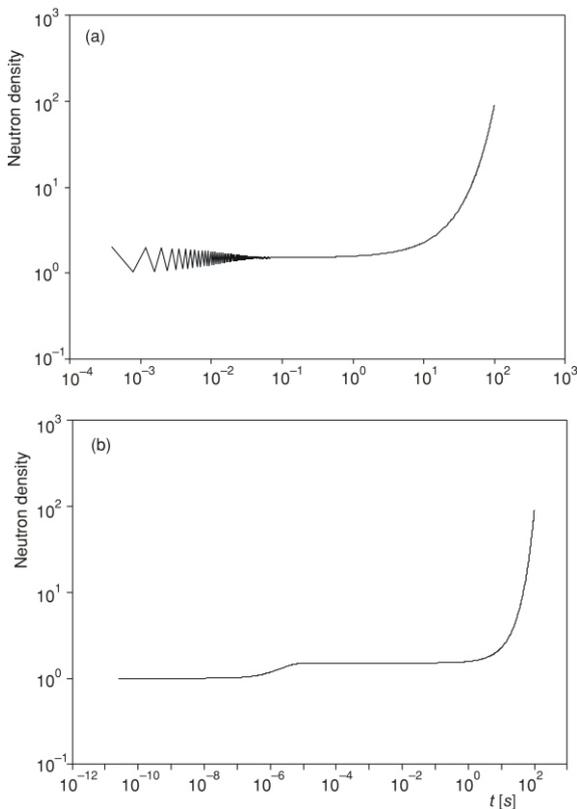
The discrepant digits are highlighted. Again, we conjecture that the RKE/R densities are the more accurate of the two.

In contrast, the (analytical) reference solution for the one delayed group benchmark of ref. [9] matches exactly all 8 figures, as given in tab. B1.6d.

**Table B1.6d. Step insertion of 0.0022 into a one group ( $\beta = 0.0065$ ,  $\lambda = 0.08 \text{ s}^{-1}$ ) fast reactor II**

$t$ [s]	Difference	Converged	No of dis.
1.0000E+00	1.5747825E+00	1.5747825E+00	2560
1.0000E+01	2.2761600E+00	2.2761585E+00	80
1.0000E+02	9.0592534E+01	9.0577793E+01	160

Since the neutrons regenerate so quickly ( $\Lambda = 10^{-8} \text{ s}$ ), this case provides an excellent example of how the RKE/R algorithm overcomes the initial oscillations, as shown in fig. 2(a). If we introduce additional edit points in decades from  $10^{-9}$  to  $10^{-2} \text{ s}$ , we eliminate the oscillations, as shown in fig. 2(b). The increase in computation time is negligible.



**Figure 2. Benchmark B1.6d:** (a) 3 edits, (b) 10 edits

*B1.7: various reactivity insertions into thermal reactor III, refs. [8] and [10]*

We conclude the series of step reactivity insertions with relative small insertions into the thermal reactor III, with kinetics parameters given in tab. 5 and  $\Lambda = 10^{-4} \text{ s}$ .

**Table 5. Kinetics parameters for thermal reactor III**

$\beta = 0.0064$		
$i$	$\beta_i$	$\lambda_i [\text{s}^{-1}]$
1	0.000211	0.0124
2	0.001402	0.0305
3	0.001254	0.115
4	0.002528	0.301
5	0.000736	1.138
6	0.000269	3.01

The reference solution comes from ref. [10]. As observed from tab. B1.7, the RKE/R solution agrees with the analytical solution and gives one more digit for reference purposes.

**Table B1.7. Various step insertions into thermal reactor III**

$t$ [s]	$\rho$	Converged
0.01	$6.4 \cdot 10^{-6}$	1.0000473E+00
0.1	$1.12 \cdot 10^{-5}$	1.0010263E+00
0.175	$2.24 \cdot 10^{-5}$	1.0018483E+00
0.35	$4.48 \cdot 10^{-5}$	1.0039083E+00

**Benchmark series 2: ramp reactivity insertion**

In this benchmark series, reactivity is inserted linearly,  $\rho(t) = \rho_0 + at$ .

*B2.1: ramp reactivity insertion rate \$0.1/s into thermal reactor I, refs. [2], [3], [4], [6], [7], [9], [10], and [12]*

We begin this series with one of the more popular benchmarks found in literature. This is a \$ 0.1/s ramp ( $\rho_0 = 0$ ) into the thermal reactor I with neutron generation time  $\Lambda = 5 \cdot 10^{-5} \text{ s}$ .

The results, shown in tab. B2.1, are nearly in complete agreement with what seems to be the ac-

**Table B2.1. Ramp insertion of \$0.1 into thermal reactor I**

$t$ [s]	Difference	Converged	No of dis.
2.0000E+00	1.338200E+00	1.338200E+00	128
4.0000E+00	2.228485E+00	2.228442E+00	32
6.0000E+00	5.582592E+00	5.582053E+00	32
8.0000E+00	4.279758E+01	4.278630E+01	64
1.0000E+01	4.514887E+05	4.511636E+05	512
1.1000E+01	1.792953E+16	1.792214E+16	2048

cepted reference solution [9]. Only one value in the sixth place is discrepant (highlighted). The RKE/R density is, however, confirmed in ref. [12]. We also confirm the reference solution of ref. [7] at  $t = 8$  s. The average time step for this calculation is 0.004 s and the time of computation is 0.063 s. If the last time edit is not included, then the average time step is 0.012 s, as compared to 0.008 s of ref. 7.

*B2.2: ramp reactivity insertion rate \$1/s in fast reactor I, refs. [1] and [7]*

The last ramp benchmark is for \$1/s into the fast reactor I with a neutron generation time of  $\Lambda = 10^{-7}$ s. As seen from tab. B2.2, we confirm the reference solution to only five places. The average time step is 0.0009 s, again larger than in ref. 7, while the time of computation is about 0.016 s.

**Table B2.2. Ramp insertion of \$1 in fast reactor I**

$t$ [s]	Referent sol.	Converged	No of dis.
5.0000E-01	2.136407E+00	2.136409E+00	512
1.0000E+00	1.207813E+03	1.207814E+03	2048

**Benchmark series 3: specifically induced transients**

The final series concerns expressly crafted reactivity insertions with non-uniform insertion rates.

*B3.1: zigzag reactivity insertion, refs. [1],[2],[3],[5], and [7]*

The zigzag reactivity insertion is also popular for proposed benchmarking methods. The zigzag reactivity is the following function of time,

$$\rho(t) = \begin{matrix} 0.0075t & 0 & t & 0.5 \\ 0.0075(t - 0.5) & 0.00375 & 0.5 & t & 1 \\ 0.0075(t - 1) & 1 & t & 1.5 \\ 0.00375 & 1.5 & t & \end{matrix}$$

The reactivity insertion is for thermal reactor I with a neutron generation time of  $\Lambda = 5 \cdot 10^{-5}$  s. The results are given in tab. B3.1. Again, RKE/R gives the reference solution for an average time step of 0.052 s and a computation time of 0.031 s.

**Table B3.1. Zigzag insertion for thermal reactor I**

$t$ [s]	Difference	Converged	No of dis.
5.0000E-01	1.721483E+00	1.721422E+00	32
1.0000E+00	1.211149E+00	1.211127E+00	32
1.5000E+00	1.892280E+00	1.892226E+00	32
2.0000E+01	2.521628E+00	2.521601E+00	32
1.0000E+01	1.204777E+01	1.204711E+01	64

*B3.2: sinusoidal reactivity insertion, refs. [2],[3],[4],[5],[7],[8], and [9]*

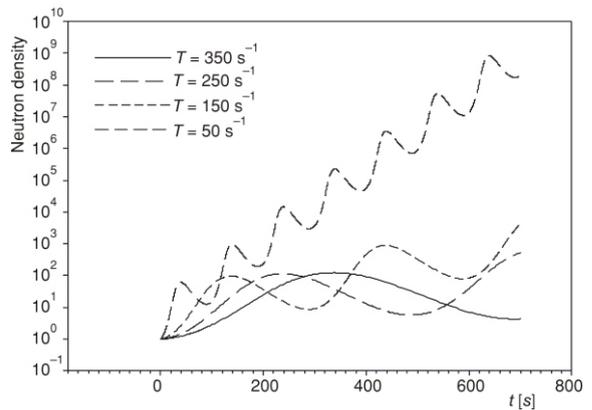
In the last benchmark, we consider the following sinusoidal reactivity variation [13]

$$\rho(t) = \frac{8\beta}{8 \lambda T} \sin(\pi t/T)$$

for a fast reactor with one delayed group. In this case  $\beta = 0.0079$ ,  $\lambda = 0.077 \text{ s}^{-1}$ , and  $\Lambda = 10 \text{ s}^{-8}$ .

This is the simplest example of an imposed self-limiting reactivity where the power oscillates in time, experiencing a steady average increase, as shown in fig. 3 for four half periods  $T$ .

For this benchmark, we only verify the analytical results quoted in ref. [14] for the time of occurrence specified for the first density peak and its value. The results shown in tab. B3.2 are nearly in complete agreement with the reference solution, except in the fourth place of the highlighted entry. The average time step in this case is 0.3 s.



**Figure 3. Benchmark B3.2: sinusoidal insertion for various periods**

**Table B3.2. Sinusoidal insertion for a one group fast reactor**

Half period $T$ [s <sup>-1</sup> ]	$t$ [s]	Converged
50	3.9108E+01	6.15339E+01
150	1.3732E+02	9.58190E+01
250	2.3712E+02	1.13458E+02
350	3.3707E+02	1.23820E+02

**DISCUSSION**

We have demonstrated the solution of RKEs via a second order finite difference algorithm enhanced through the Romberg acceleration. The RKE/R algorithm has been benchmarked against nearly every major benchmark for imposed reactivity insertion of which this author is aware. We have reproduced most reference solutions and there is indication that, when

they differ, the RKE/R solution is likely to be the most accurate one. The applications covered step and ramp insertions, as well as zigzag and sinusoidal insertions. All of the results obtained required minimal computational effort, with computing times generally under 0.2 s. Notably, the RKE/R algorithm performance is comparable to the Rosenbrock algorithm [7] in computational effort. However, when the two algorithms are compared with respect to simplicity and ease of use, the RKE/R wins without question which, to this author, is the most important measure of pedagogical intent. The (revisited) Rosenbrock algorithm of ref. [7] requires 26 parameters to be specified, including a specialized time stepping procedure which, one might suspect, might not be as efficient in all anticipated cases. Only the finite difference forms of eq. (3) and the Romberg acceleration of eq. (14) are required for the RKE/R, with no need to determine any parameters. One could only imagine that teaching and learning the RKE/R algorithm is generally easier than is the case with other algorithms, but all algorithms should be taught, if possible.

The RKE/R algorithm's most outstanding feature is its simplicity. No special consideration of stiffness is required – no series expansions in either polynomials or exponentials, no approximations of the exponential, no determination of eigenvalues. While the author would like to take credit for the RKE/R algorithm, he cannot do so, as it was first proposed by Izumi and Noda [18] in 1970. However, from then on, it seems not to have undergone any further development. Had this been the case, the RKE/R algorithm would almost certainly have emerged as the method of our choice today.

To conclude and further emphasize the superiority of the RKE/R algorithm, we will consider one final case.

If the converged solution is indeed accurate, it can be used to reliably estimate the order of the finite difference approximation. To verify this, we use the simple one-group benchmark devised by Kinard and Allen [4]. In this case,  $\beta = 1$ ,  $\lambda = 1$  s, and  $\Lambda = 1$  s<sup>-1</sup>, with reactivity insertion  $t$ . Neutron density is given in tab. 6, where  $p$  is the estimate of the order of the difference scheme of eq. (3a). As observed, we have indeed captured the order to five places, exactly; however, the density has converged well before to ten-places. However, quadruple precision arithmetic is required to obtain the indicated order.

**Table 6. Kinard and Allen benchmark**

$h$	$n$	$p$
2.500E-01	4.5119376409E+00	2.14800E+00
1.250E-01	4.5280399694E+00	2.03467E+00
6.250E-02	4.5279110538E+00	2.00853E+00
3.125E-02	4.5279113100E+00	2.00213E+00
1.562E-02	4.5279113099E+00	2.00053E+00
7.812E-03	4.5279113099E+00	2.00013E+00
3.906E-03	4.5279113099E+00	2.00003E+00
1.953E-03	4.5279113099E+00	2.00001E+00
9.766E-04	4.5279113099E+00	2.00000E+00

There is no question that the RKE/R algorithm is the high order method we have been searching for over the past 40 years.

## REFERENCES

- [1] Nobrega, J. A. W. da., A New Solution of the Point Kinetics Equations, *Nucl. Sci. & Eng.*, 46 (1971), 3, pp. 366-375
- [2] Chao, Y., Attard, A., A Resolution of the Stiffness Problem of Reactor Kinetics, *Nucl. Sci. & Eng.*, 90 (1985), 1, pp. 40-46
- [3] Sanchez, J., On the Numerical Solution of Point Reactor Kinetics Equations by the Generalized Runge-Kutta Methods, *Nucl. Sci. & Eng.*, 103 (1989), pp. 94-99
- [4] Kinard, M., Allen, E., Efficient Numerical Solution of Point Kinetics Equations in Nuclear Dynamics, *Ann. Nucl. Ener.*, 31 (2004), 9, pp. 1039-1051
- [5] Aboanber, A., Nahla, A. E., On Pade' Approximations to the Exponential Function and Application to Point Kinetics Equations, *Prog. Nucl. Ener.*, 44 (2004), 4, pp. 347-368
- [6] Quintero-Leyva, B., CORE: A Numerical Algorithm to Solve the Point Kinetics Equations, *Ann. Nucl. Ener.*, 35 (2008), 11, pp. 2136-2138
- [7] Yang, X., Jevremovic, T., Revisiting the Rosenbrock Numerical Solutions of the Reactor Point Kinetics Equation with Numerous Examples, *Nuclear Technology & Radiation Protection*, 24 (2009), 1, pp. 3-12
- [8] Aboanber, A., Hamada, Y., PWS: An Efficient Code System for Solving Space-Independent Nuclear Reactor Dynamics, *Ann. Nucl. Ener.*, 29 (2002), 18, pp. 2159-2172
- [9] Li, H., Chen, W., Luo, L., Zhu, Q., A New Integral Method for Solving the Point Reactor Neutron Kinetics Equations, *Ann. Nucl. Ener.*, 36 (2009), 4, pp. 427-432
- [10] Lewins, J. D., Reactivity Oscillations and Stability with Delayed Neutrons, *Ann. Nucl. Ener.*, 22 (1995), 6, pp. 411-414
- [11] Kaganove, J. J., Numerical Solution of the One Group Space-Independent Reactor Kinetics Equations for Neutron Density Given Excess Reactivity, ANL-6132, Argonne National Laboratory, Lemont (Ill), USA, 1960
- [12] Van den Eynde, G., Comments on "A resolution of the Stiffness Problem of Reactor Kinetics", *Nucl. Sci. & Eng.*, 153 (2006), 2, pp. 200-202
- [13] Szeligowski, J., Hetrick, D. L., A Comparison of Some Numerical Methods for Solving the Equations of Reactor Dynamics, *Proceedings, Conf. Utilization of Research Reactors and Reactor Mathematics and Computation*, Mexico City, Mexico, 1967, CUM-R-2 (Vol. 2), pp. 994-1014
- [14] Vigil, J., Solution of the Reactor Kinetics Equations by Analytic Continuation, *Nucl. Sci. & Eng.*, 29 (1967), pp. 392-401
- [15] Hetrick, D. L., Dynamics of Nuclear Reactors, The University of Chicago Press, Chicago, Ill, USA, 1971
- [16] Abramowitz, M., Stegun, I., eds., Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, Dover Publications, New York, USA, 10<sup>th</sup> ed., ISBN 978-0-486-61272-0, 1972, p. 886
- [17] Atkinson, K., Han, W., Elementary Numerical Analysis, John Wiley & Sons, Inc, Mass., USA, 2004
- [18] Izumi, M., Noda, T., An Implicit Method for Solving the Lumped Parameter Reactor-Kinetics Equations by Repeated Extrapolation, *Nucl. Sci. & Eng.*, 41 (1971), pp. 299-303

**Бари Д. ГАНАПОЛ**

**УСАВРШЕН НАЧИН РЕШАВАЊА ЈЕДНАЧИНА ТАЧКАСТЕ КИНЕТИКЕ  
РЕАКТОРА ЗА ЗАДАТУ УНЕТУ РЕАКТИВНОСТ**

Да се реше једначине реакторске кинетике примењен је концепт убрзања конвергенције, такође познат као екстраполација. Једноставност методе исказана је формулацијом решења заснованог на апроксимацији коначних разлика, која конвергира са високом тачношћу користећи познату вредност грешке. Помоћу Ромбергове екстраполације показана је висока тачност поступка за различите задате промене реактивности познате у литератури. Јединствена особина предложеног алгоритма, названог РКЕ/Р(омберг), у томе је да се не обраћа посебна пажња чврстини кинетичких једначина реактора. Најзад, по својој једноставности и тачности, може се РКЕ/Р алгоритам основано сматрати најделотворнијим нумеричким решењем реакторских кинетичких једначина развијеним до сада.

*Кључне речи: кинетика нуклеарних реактора, Ромбергова екстраполација, унета реактивности, термички реактор, брзи реактор*

---