REVISITING THE ROSENBROCK NUMERICAL SOLUTIONS OF THE REACTOR POINT KINETICS EQUATION WITH NUMEROUS EXAMPLES

by

Xue YANG and Tatjana JEVREMOVIC

Received on March 10, 2009; accepted on April 3, 2009

The fourth order Rosenbrock method with an automatic step size control feature was described and applied to solve the reactor point kinetics equations. A FORTRAN 90 program was developed to test the computational speed and algorithm accuracy. From the results of various benchmark tests with different types of reactivity insertions, the Rosenbrock method shows high accuracy, high efficiency and stable character of the solution.

Key words: reactor point kinetics, Rosenbrock method, FORTRAN 90

INTRODUCTION

The point kinetics equations, the simplified nuclear kinetics model, are a system of stiff ordinary differential equations (ODE). Although the current research highlights focus on the space-time solutions, the point kinetics equations still play an important role in nuclear kinetics analysis. An accurate and efficient general numerical method is needed to solve the point kinetics equations with any reactivity driving functions.

Extensive literature can be found on how to solve the reactor point kinetics equations. For example, Chao and Attard [1] developed the well known stiffness confinement method that eliminates the stiffness of the delayed neutron precursor and confines the stiffness inside the neutron density. The traditional generalized Runge-Kutta method was used by Sánchez [2] to solve the stiffness problem. Recently, the power series solution [3], reactivity piecewise constant approximations [4], Padé approximations [5], CORE numerical algorithm [6], and better basis function [7] have been successively applied to numerical solutions of the reactor point kinetics equations. These methods are briefly discussed as follows.

Stiffness Confinement Method (SCM). By introducing an assistant function, the stiffness was confined inside the neutron density and eliminated from the delayed neutron precursor density. Within one time step, iteration and a linear extrapolation are required to find the value of the assistant function. The numerical tests demonstrate that SCM produces moderate accuracy under moderate stiff condition and low accuracy where the stiffness is strong. The implementation of this method is relatively complex.

Generalized Runge-Kutta method (GRK). The solution of the fourth order GRK method adopts the form of the Runge-Kutta solution. The coefficients are found by solving four systems of linear equations. The automatic step size control was realized based on the truncation error of each time step. There is no approximation in this method, and good agreement between the GRK solution and the exact value is obtained, but the relative error of GRK solution may exceed the predefined tolerance under the strong stiffness condition.

Power Series Solution (PWS). The power series was used to represent the solution of the point kinetics equations. The neutron density is obtained after computing the matrix elements and PWS coefficients in the recurrence relation. A high accuracy in the slow

Nuclear Technology & Radiation Protection
Scientific paper
UDC: 539.122:519.245
BIBLID: 1451-3994, *24* (2009), 1, pp. 3-12
DOI: 10.2298/NTRP0901003Y

Authors' address:
School of Nuclear Engineering,
Purdue University
West Lafayette, Ind., 47907, USA

E-mail address of corresponding author:
tatjanaj@ecn.purdue.edu (T. Jevremovic)

transient case can be obtained while the moderate accuracy under the condition of the strong stiffness is observed. The computation speed is relatively slow compared to the Rosenbrock method described in this paper.

Reactivity Piecewise Constant Approximations (PCA). PCA assumes that the reactivity slowly varies with respect to time. The piecewise constant function over a time partition is used to approximate the continuous reactivity function. Newton's method is utilized to find the roots of the inhour equation, which are the eigenvalues of the point kinetics matrix, and the eigenvector is computed by matrix multiplications. This methodology applies to the moderate fast or slow transient problems, since the error becomes large when the problem becomes very stiff. This method showed to be accurate for the step reactivity, and moderately accurate in computing the ramp reactivity. The implementation of PCA is simple and computation speed is expected to be fast.

Padé Approximations. The Padé rational approximation is used to approximate the exponential function in the point kinetics equation solution. The roots of the inhour formula are used as the eigenvalues of the point kinetics matrix. The numerical tests demonstrated that high accuracy could be achieved in strong step reactivity insertion. The low step size was required to compute the zigzag reactivity ramp to acquire the same accuracy.

CORE Numerical Algorithm. Laplace transform and Heaviside expansion theorem give an explicit solution to the point kinetics equations. The roots of the inhour equation serve as the parameters in the exponential term and the other coefficients can be found through another explicit formula. Then, determining the initial condition for the next time interval is required to advance to the next step. This methodology assures relatively moderate accuracy.

Better Basis Function (BBF). This recently published methodology is based on the third order exponential function used as a basis function to approximate the integration of the neutron density. The four coefficients in the basis function are obtained by solving four equations. The users have to provide the analytic equation of the integration of the multiplication of the reactivity function and an exponential function. Since the integration is not as friendly as the differentiation, the integration term may not be represented by the elementary functions for some reactivity functions. The numerical tests show that this method is relatively accurate compared to the large step size used in those benchmarks. The discrepancy becomes large when the neutron density changing is very fast.

Most of these methods cannot achieve the performance of the Rosenbrock method. In other words, the Rosenbrock method can achieve high accuracy with fewer integration steps. Besides, without the function of the automatic step size control, users will have difficulty in determining the step size based on the desired accuracy and problem stiffness. Compared to the traditional GRK method, three major improvements were made: the formulas were revised to remove the time variable from the list of the dependent variables to reduce the computation time; the original set of parameters was replaced by another set proposed by Shampine [8] for better performance; LU (Lower triangular Upper triangular matrix) decomposition substituted the matrix inverting to reduce the tedious mathematic derivation and coding work, while the general mathematic procedure made the program have the function of solving the point kinetics equation with any delayed neutron groups.

In this paper, the Rosenbrock method solution of the reactor point kinetics equations is presented. A FORTRAN 90 program was developed to test the accuracy and efficiency of this methodology (see Appendix). Numerous benchmark tests are presented leading to a conclusion that the presented methodology shows high accuracy and high efficiency. We also used this development to incorporate into experiential interactive learning tools on reactor physics at the junior level in nuclear engineering programs at the universities in United States. There is an evident nuclear renaissance started with this century, inspiring educational system to revitalize and revisit teaching methods.

ROSENBROCK METHOD IN POINT KINETICS EQUATIONS

The Rosenbrock method [9], also known as the Kaps-Rentrop method, evolves from the GRK method. It seeks the solution of stiff ODE in the following form

$$y \quad f(y) \tag{1}$$

where y is an n-dimensional column vector. The solution of eq. (1) at each time step is given with

$$y(t_0 \quad h) \quad y(t_0) \quad \sum_{i\ 1}^{s} b_i g_i \tag{2}$$

where t_0 is the initial time, h – the step size, b_i – the constants, g_i – the column vectors, and s – the order of the Rosenbrock method.

In order to perform the automatic step size control, two estimates of eq. (2) are computed: one y with the order of s and coefficients b_i and the other $\hat{y}$, with a lower order of $\hat{s}$ and different coefficients $\hat{b}_i$. The difference between y and $\hat{y}$ gives the truncation error at this time step that can be used for step size adjustment. Kaps and Rentrop [10] suggested the smallest value of s and $\hat{s}$ to be 4 and 3, respectively; this is called the 4th order method [9].

Adding t to the list of dependent variables is not necessary, even if the right-hand side (RHS) of eq. (1), $f(y, t)$ explicitly depends on t. The g_i needed in eq. (2) is obtained as follows

$$\begin{aligned}
&\frac{1}{\gamma h} I - \frac{\partial f}{\partial y}\, g_1 = f(y_0, t_0) + hc_1 \frac{\partial f}{\partial t} \\
&\frac{1}{\gamma h} I - \frac{\partial f}{\partial y}\, g_2 = f(y_0 + a_{21} g_1, t_0 + a_2 h) \\
&\quad + hc_2 \frac{\partial f}{\partial t} + \frac{c_{21} g_1}{h} \\
&\frac{1}{\gamma h} I - \frac{\partial f}{\partial y}\, g_3 = f(y_0 + a_{31} g_1 + a_{32} g_2, t_0 + a_3 h) \\
&\quad + hc_3 \frac{\partial f}{\partial t} + \frac{c_{31} g_1 + c_{32} g_2}{h} \\
&\frac{1}{\gamma h} I - \frac{\partial f}{\partial y}\, g_4 = f(y_0 + a_{31} g_1 + a_{32} g_2, t_0 + a_3 h) \\
&\quad + hc_4 \frac{\partial f}{\partial t} + \frac{c_{41} g_1 + c_{42} g_2 + c_{43} g_3}{h}
\end{aligned} \tag{3}$$

where h is the step size, a, c, and γ are constants, and I is the identity matrix.

Equations (3) consists of four linear systems of equations. Because the left-hand side (LHS) matrices are the same, only one time of the LU decomposition is needed to factorize the LHS matrix. Thus, starting from the first equation of eqs. (3), g_i are found by four back-substitution with four different RHS of these equations. The estimate of y is computed by eq. (2), and the truncation error is obtained as follows

$$err = \sum_{i=1}^{4} e_i g_i \tag{4}$$

Since the error criterion is defined by the relative error, a vector y_{scale}, is required to scale the truncation error at each time step (tiny value is added to avoid y_{scale} to become zero)

$$y_{scale} = |y(t_0)| + |hf(y_0, t_0)| + 10^{-30} \tag{5}$$

The maximum error of the vector y is obtained with

$$err_{max} = \max \left| \frac{err}{y_{scale}} \right| \tag{6}$$

If the err_{max} is larger than the predefined error control criterion ε, the already computed y estimate will be abandoned, and a smaller step size h_{retry} will be found according to

$$h_{retry} = \max[0.9h(err_{max})^{1/3}, 0.5h] \tag{7}$$

to recalculate y. Otherwise, this y estimate will be accepted and the next step size will be determined based on the truncation error of this time step as follows

$$h_{next} = \begin{cases} 0.9h(err_{max})^{0.25} & \text{if } err_{max} > 0.1296 \\ 1.5h & \text{else} \end{cases} \tag{8}$$

Instead of using the coefficients of Kaps and Rentrop [10], the parameters suggested by Shampine [8], which are assumed to have better stability and accuracy, are used in developing this methodology. Those parameters are

$\gamma = 0.5$;
$a_{21} = 2$, $a_{31} = 1.92$, $a_{32} = 0.24$;
$c_{21} = -8$, $c_{31} = 14.88$, $c_{32} = 2.4$, $c_{41} = -0.896$,
$c_{42} = -0.432$, $c_{43} = -0.4$;
$b_1 = 19/9$, $b_2 = 0.5$, $b_3 = 25/108$, $b_4 = 125/108$;
$e_1 = 17/54$, $e_2 = 7/36$, $e_3 = 0$, $e_4 = 125/108$;
$c_1 = 0.5$, $c_2 = -1.5$, $c_3 = 2.42$, $c_4 = 0.116$;
$a_2 = 1$, $a_3 = 0.6$.

The well-known reactor point kinetics equations are

$$\begin{aligned}
\frac{dn(t)}{dt} &= \frac{\rho(t) - \beta}{\Lambda} n(t) + \sum_{i=1}^{6} \lambda_i C_i(t) \\
\frac{dC_i(t)}{dt} &= \frac{\beta_i}{\Lambda} n(t) - \lambda_i C_i(t) \quad i = 1, 2, \ldots, 6
\end{aligned} \tag{9}$$

where $n(t)$ is the neutron density, $C_i(t)$ – the delayed neutron precursor density of i-th group, $\rho(t)$ – the reactivity driving function, β – the total delayed neutron fraction, β_i – the delayed neutron fraction of the i-th group, λ_i – the decay constant of the i-th group, and Λ – the average neutron generation time.

Therefore, for a given reactivity driving function $\rho(t)$, the matrices in eq. (3) are defined as

$$f(y,t) = \begin{bmatrix} \frac{\rho(t) - \beta}{\Lambda} & \lambda_1 & \cdots & \lambda_6 \\ \frac{\beta_1}{\Lambda} & \lambda_1 & & \\ \vdots & & \ddots & \\ \frac{\beta_6}{\Lambda} & & & \lambda_6 \end{bmatrix} y \tag{10}$$

where

$$y = \begin{bmatrix} n(t) \\ C_1(t) \\ \vdots \\ C_6(t) \end{bmatrix}$$

$$\frac{\partial f}{\partial y} = \begin{bmatrix} \frac{\rho(t) - \beta}{\Lambda} & \lambda_1 & \cdots & \lambda_6 \\ \frac{\beta_1}{\Lambda} & \lambda_1 & & \\ \vdots & & \ddots & \\ \frac{\beta_6}{\Lambda} & & & \lambda_6 \end{bmatrix} \tag{11}$$

$$\frac{\partial f}{\partial t} \quad \begin{matrix} \frac{n(t)}{\Lambda}\frac{\mathrm{d}\rho(t)}{\mathrm{d}t} \\ 0 \\ \vdots \\ 0 \end{matrix} \qquad (12)$$

Because the matrix dimension is reduced by one by removing the time variable from the list of the dependent variables, the computation time can be saved for 30% (for six delayed neutron groups) to 60% (for one delayed neutron group) compared to the conventional GRK method [2].

BENCHMARK TESTS

We present seven different benchmark examples [7, 11, 12] including the step reactivity, ramp reactivity, zigzag ramp reactivity and sine reactivity to verify the methodology and numerical solutions, the accuracy and computational time. In all presented tests the automatic step size control with an initial tentative step size of 0.01s and the error toleration of 10^{-6} is specified. All of the presented kinetics examples start from the equilibrium state, therefore, the $C_i(0)$ are computed using

$$C_i(0) \quad \frac{\beta_i}{\Lambda\lambda_i} n(0) \qquad (13)$$

with $n(0) = 1.0$. The benchmarks 1, 2, 3, 5, and 6 share two sets of typical reactor parameters that are listed in tab. 1. The exact solutions of those benchmark tests are adopted from da Nobrega [11], but the method that produced the exact solution was not mentioned in this reference.

The computation time of the numerical solutions presented for all benchmark tests is obtained by running the FORTRAN90 code on the AMD Opteron 2354 Processor. The average step size is obtained as the total integration time divided by the total number of calculation steps.

Table 1. Kinetics parameters for two typical reactors

Group	Thermal reactor		Fast reactor	
	$\lambda_i[s^{-1}]$	β_i	$\lambda_i[s^{-1}]$	β_i
1	0.0127	0.000285	0.0129	0.0001672
2	0.0317	0.0015975	0.0311	0.001232
3	0.115	0.00141	0.134	0.0009504
4	0.311	0.0030525	0.331	0.001443
5	1.40	0.00096	1.26	0.0004534
6	3.87	0.000195	3.21	0.000154
β	0.0075		0.0044	
$\Lambda(s)$	0.0005		10^{-7}	

Benchmark 1: fast reactor, reactivity step of $0.5*

A constant reactivity of $\rho(t) = 0.0022$ is inserted into a fast reactor. The neutron density at three time points is presented in tab. 2. The average step size is 0.078 s. The neutron density response produced by Rosenbrock method exactly matches the reference value. Figure 1 shows the neutron density and delayed neutron precursor density *vs*. time. Each point represents one calculated time point. At the beginning of the positive reactivity insertion, the neutron density rapidly increases in a very short period of time; therefore, the step sizes at the beginning are very small to overcome the strong stiffness. After the initial jump, the increasing speed of neutron density becomes slow, resulting in large step sizes. The delayed neutron precursor density follows the relatively same increasing trend but without the initial jump. Therefore, the main stiffness occurs in the neutron density.

Table 2. Neutron density of benchmark 1 (fast reactor, step of $0.5); CPU time: 0.002 s

t (*s*)	Reference [11]	Rosenbrock	Relative error
0.1	2.075317	2.075317	0.0
1.0	2.655853	2.655853	0.0
10.0	12.74654	12.74654	0.0

Benchmark 2: thermal reactor, reactivity step of –$0.5

In this example the reactivity driving function is $\rho(t) = -0.00375$, which equates –$0.5 step insertion. Table 3 shows the neutron density change, and the average step size is 0.093 s. The zero relative error indicates that the relative error of the neutron density is much lower than the predefined error tolerance. The neutron density and delayed neutron precursor density are displayed in fig. 2. The negative reactivity insertion introduces a neutron density jump at the beginning; therefore, the Rosenbrock method automatically selects small step sizes at the start. On the other hand, the curves of the delayed neutron precursor density are relatively flat. Thus the main stiffness also occurs within the neutron density.

Table 3. Neutron density of benchmark 2 (thermal reactor, step of –$0.5); CPU time: 0.002 s

t (*s*)	Reference [11]	Rosenbrock	Relative error
0.1	0.698925	0.698925	0.0
1.0	0.607054	0.607054	0.0
10.0	0.396078	0.396078	0.0

* $1 = β

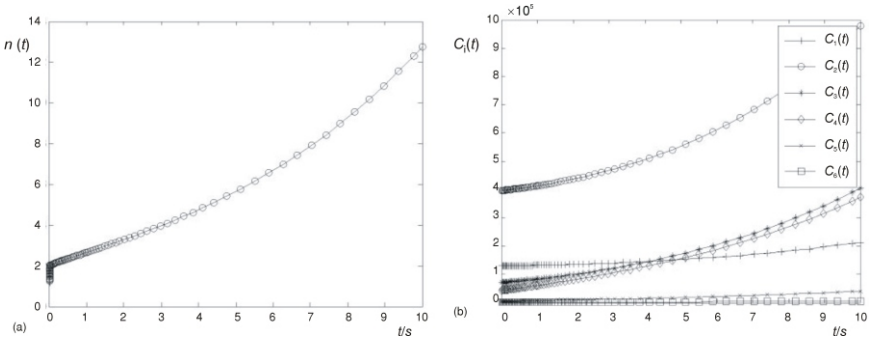


Figure 1. Benchmark 1: (a) Neutron density; (b) Delayed neutron precursor density

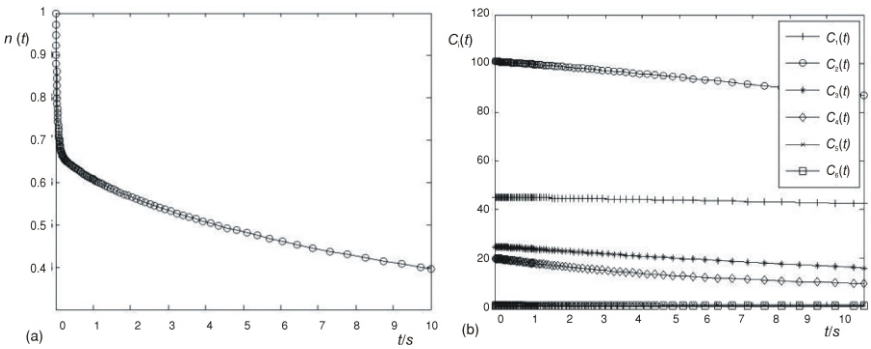


Figure 2. Benchmark 2: (a) Neutron density; (b) Delayed neutron precursor density

Benchmark 3: thermal reactor, reactivity step of \$1

The neutron density listed in tab. 4 corresponds to \$1 step reactivity insertion into a thermal reactor, where $\rho(t) = 0.0075$. The average step size is 0.020 s. The Rosenbrock neutron density is also accurate in this strong positive reactivity step insertion test case. Figure 3 displays the neutron density and delayed neutron precursor. A larger increasing slope is expected due to the stronger step reactivity insertion. Because, as shown with eqs. (8), for stability reason, the maximum h growing factor is 1.5, the initial 0.01 s step size gradually increases till the end of the integration time.

Table 4. Neutron density of benchmark 3 (thermal reactor, step of \$1); CPU time: 0.002 s

$t(s)$	Reference [11]	Rosenbrock	Relative error
0.1	2.515766	2.515766	0.0
0.5	10.36253	10.36253	0.0
1.0	32.18354	32.18355	$3.1\ 10^{-7}$

Benchmark 4: thermal reactor, reactivity ramp of \$0.1/s

The thermal reactor decay constants λ_i given in tab. 1 are used for this test, but the other parameters are replaced with: $\beta_1 = 0.000266$, $\beta_2 = 0.001491$, $\beta_3 = = 0.001316$, $\beta_4 = 0.002849$, $\beta_5 = 0.000896$, $\beta_6 = = 0.000182$, $\beta = 0.007$, and $\Lambda = 2\ 10^{-5}$ s [2]. The reactivity driving function is $\rho(t) = 0.0007\ t$. The neutron density at five time points between 0 and 10 s is computed and shown in tab. 5. The reference value was provided by

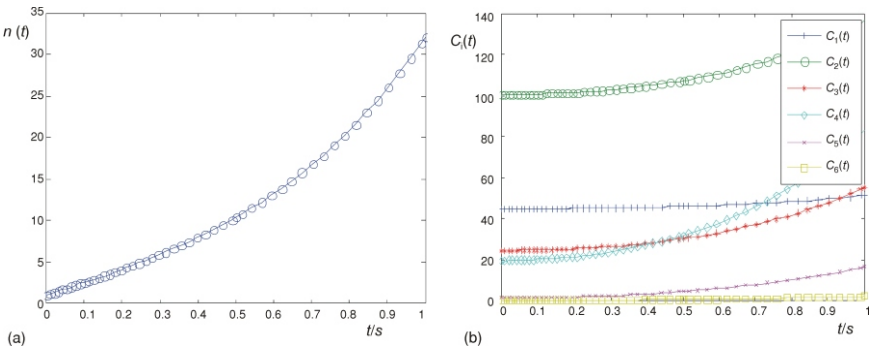


Figure 3. Benchmark 3: (a) Neutron density; (b) Delayed neutron precursor density

Table 5. Neutron density of benchmark 4 (thermal reactor, ramp of \$0.1/s); CPU time: 0.029 s

$t(s)$	Reference [7]	Rosenbrock	Relative error
2	1.338200	1.338200	0.0
4	2.228442	2.228442	0.0
6	5.582052	5.582051	$-1.8\ 10^{-7}$
8	42.78630	42.78629	$-2.3\ 10^{-7}$
10	451163.6	451163.6	0.0

Hermite polynomial method with the fixed step size of 0.0001 s [7]. An average step size of 0.008 s was obtained with the Rosenbrock method. By examining the relative error given in tab. 5, the Rosenbrock method gives accurate values consistently. All relative errors are much lower than the predefined error tolerance. The neutron density and delayed neutron precursor density are presented in fig. 4. After 5 s, when the reactivity exceeds \$0.5, the neutron density and delayed neutron precursor density exponentially increase. The Rosenbrock method could accurately solve this stiffness. If this benchmark is calculated with the error tolerance of $2\ 10^{-5}$, the maximum Rosenbrock relative error is $-9.7\ 10^{-6}$, which is lower than the maximum GRK relative error of $-3.0\ 10^{-5}$ computed by 10^{-5} error tolerance. In addition, the Rosenbrock average step size of 0.023 s is larger than the GRK average step size of 0.020 s [2]. By comparing the Rosenbrock CPU time with PWS CPU time of 3.02 s [3], the Rosenbrock method is about a hundred times faster than the PWS method.

Benchmark 5: fast reactor, reactivity ramp of \$1/second

A fast ramp of \$1/s produces the neutron density given in tab. 6. The stiffness is relatively strong. The reactivity driving function is defined as $\rho(t) = 0.0044\,t$. Figure 5 illustrates the fast increasing neutron density and delayed neutron precursor density. After 0.5 s, this problem appears to be very stiff. The neutron density and delayed neutron precursor density exponentially increase after 0.5 s. The relatively small average step size of 0.0006 s of this case also shows that the step size can be automatically determined based on the stiffness of the problem. The relative error that is lower than the predefined tolerance in tab. 6 also proves that the Rosenbrock method gives the exact results as the mathematical expectation.

Table 6. Neutron density of benchmark 5 (fast reactor, ramp of \$1/s); CPU time: 0.03 s

$t(s)$	Reference [11]	Rosenbrock	Relative error
0.5	2.136407	2.136406	$-4.7\ 10^{-7}$
1.0	1207.813	1207.814	$8.3\ 10^{-7}$

Benchmark 6: thermal reactor, zigzag reactivity ramp of \$1/second

A zigzag ramp reactivity is inserted into a thermal reactor. The reactivity increases at the speed of \$1/s till 0.5 s, then it decreases with a slope of –\$1/s

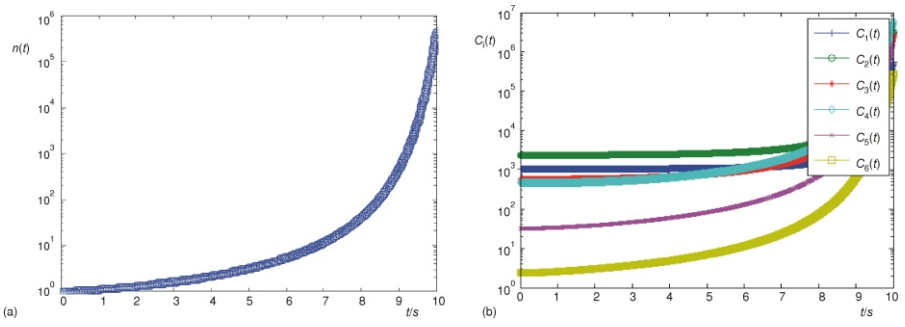


Figure 4. Benchmark 4: (a) Neutron density; (b) Delayed neutron precursor density

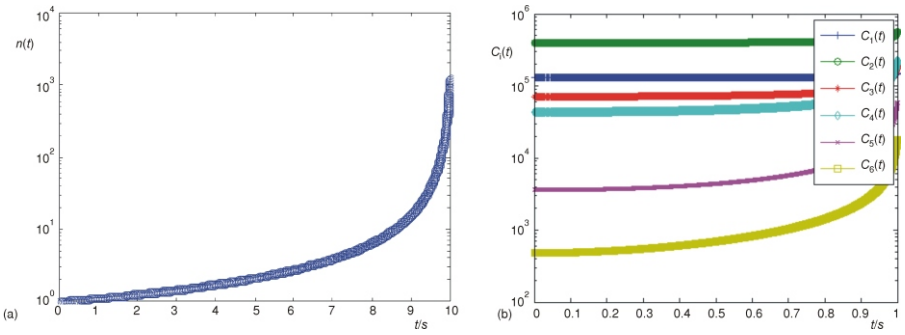


Figure 5. Benchmark 5: (a) Neutron density; (b) Delayed neutron precursor density

from 0.5 s to 1.0 s, thereafter is followed by a ramp of \$1/s up to 1.5 s again. From there, it maintains the constant at \$0.5. The reactivity driving function is

$$\rho(t) \quad \begin{array}{c} 0.0075t \quad 0 \quad t \quad 0.5 \\ 0.0075(t \quad 0.5) \quad 0.00375 \quad 0.5 \quad t \quad 1 \\ 0.0075(t \quad 1) \quad 1 \quad t \quad 1.5 \\ 0.00375 \quad 1.5 \quad t \quad 10 \end{array}$$

The neutron density at five sample time points is shown in tab. 7. The average step size is 0.05 s. According to all zero relative errors in tab. 7, the Rosenbrock method provides accurate results. The neutron density, reactivity driving function, and delayed neutron precursor density profiles are shown in fig. 6. The zigzag reactivity ramp brings fluctuation to the neutron density profile whose curve shape follows the reactivity driving function, but the relative magnitude of the response of the delayed neutron precursor density to the zigzag reactivity ramp is relatively small.

Compared to Padé and SCM maximum relative error of 6 10^{-4} calculated with the fixed 0.01s step size [1], the Rosenbrock method has much better performance, because it could achieve much higher accuracy with a quite large average step size. If this benchmark is calculated by the Rosenbrock method with the fixed step size of 0.03 s, the maximum relative error is 8.3 10^{-6} which is still lower than the maximum relative error of 9.5 10^{-6} computed by the GRK method with the fixed step size 0.01 s [2]

Table 7. Neutron density of benchmark 6 (thermal reactor, zigzag ramp of \$1/s); CPU time: 0.004 s

$t(s)$	Reference [11]	Rosenbrock	Relative error
0.5	1.721422	1.721422	0.0
1.0	1.211127	1.211127	0.0
1.5	1.892226	1.892226	0.0
2.0	2.521601	2.521601	0.0
10.0	12.04711	12.04711	0.0

Benchmark 7: sinusoidal reactivity

This example demonstrates the neutron density response to a sinusoidal reactivity oscillation. The one group calculation parameters are $\beta = 0.0079, \lambda =$ $= 0.077\ s^{-1}$, and $\Lambda = 10^{-8}$ s. The reactivity driving function is

$$\rho(t) \quad 0.005333\sin\frac{\pi}{50}t$$

The total CPU time is 0.025 s. The calculation was performed for one period (0-100 s). The $n(t), \rho(t)$, and $C_i(t)$ are shown in fig. 7. This neutron density curve exactly matches the prompt jump approximation (PJA) solutions provided by [12]. The one group delayed neutron precursor density curve has the similar shape, but the peak is delayed approximately 10 s. The PJA maximum response value is 61.4963 [2], while the Rosenbrock maximum neutron density is 61.4964 calculated with the average step size of 0.083 s. The peak neutron density of the Rosenbrock method is much closer to the PJA analytic solution than the GRK result, 61.509 [2].

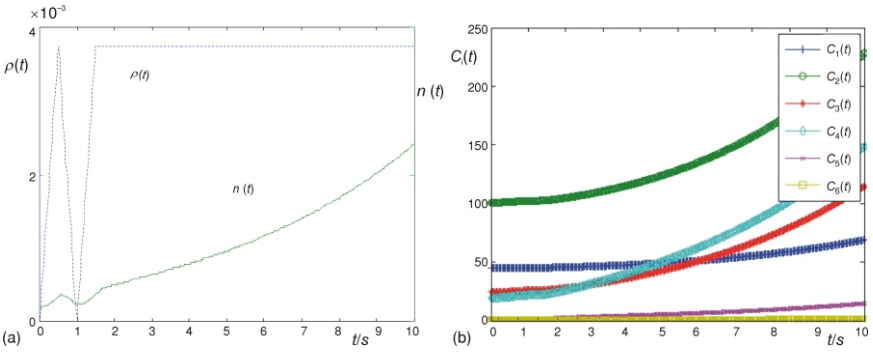


Figure 6. Benchmark 6: (a) Neutron density and reactivity; (b) Delayed neutron precursor density

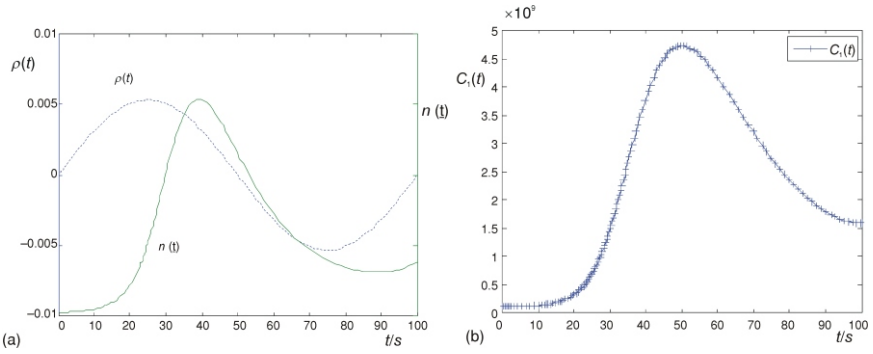


Figure 7. Benchmark 7: (a) Neutron density and reactivity; (b) Delayed neutron precursor density

DISCUSSION

The relative error that is lower than the predefined criteria in all first six test cases proves that the Rosenbrock method could solve the point kinetics equations with sufficient accuracy. From the comparisons between the Rosenbrock and GRK for benchmarks 4, 6, and 7, it is shown that the Rosenbrock method gives more accurate results with fewer calculation steps, which means that the Rosenbrock method has better accuracy and stability than the GRK method. The CPU time for all these benchmarks is quite short; the maximum time was only 0.03 s; on average, 2 10^{-5} s is needed to advance to the next time step. Therefore, the computation burden of the Rosenbrock method in solving point kinetics equations is negligible. Moreover, the implementation of this method is relatively simple. In order to develop a general code that could handle arbitrary delayed neutron group structure and reduce the coding work, inverting the LHS matrix in eq. (2) is not recommended, although it may save some calculation time. The numerical solution presented in this paper shows to be directly applicable in solving the point kinetics with any $\rho(t)$, as long as the derivatives of $\rho(t)$ are analytic. Otherwise, numerical differencing subroutine is needed.

CONCLUSION

This paper presents the fourth order Rosenbrock method applied to solve the reactor point kinetics equations. From the presented various benchmark tests, it can be concluded that the Rosenbrock algorithm exhibits high efficiency, high accuracy and stable features. By adopting the automatic step size control, the users do not have to determine the step size. The developed numerical code can automatically output the results with the desired accuracy.

The developed program has become a part of the interactive web-based course module for reactor physics teaching at the undergraduate level [13]. It will assist the students to understand and solve the neutron density response of different point kinetics models.

ACKNOWLEDGEMENT

This work is supported by the NRC Educational Grant.

REFERENCE

[1] Chao, Y., Attard, A., A Resolution of the Stiffness Problem of Reactor Kinetics, *Nucl. Sci. Eng., 90* (1985), 1, pp. 40-46

[2] Sánchez, J., On the Numerical Solution of the Point Reactor Kinetics Equations by Generalized Runge-Kutta Methods, *Nucl. Sci. Eng., 103* (1989), 1, pp. 94–99

[3] Aboanber, A. E., Hamada, Y. M., PWS: An Efficient Code System for Solving Space-Independent Nuclear Reactor Dynamics, *Ann. Nucl. Energy*, *29* (2002), 18, pp. 2159–2172

[4] Kinard, M., Allen, E. J., Efficient Numerical Solution of the Point Kinetics Equations in Nuclear Reactor Dynamics, *Ann. Nucl. Energy, 31* (2004), 9, pp. 1039–1051

[5] Aboanber, A. E., Nahla, A. A., On Padé Approximations to the Exponential Function and Application to the Point Kinetics Equations, *Prog. Nucl. Energy, 44* (2004), 4, pp. 347–368

[6] Quintero-Leyva, B., CORE: A Numerical Algorithm to Solve the Point Kinetics Equations, *Ann. of Nucl. Energy*, *35* (2008), 11, pp. 2136–2138

[7] Li, H., Chen, W., Luo, L., Zhu, Q., A New Integral Method for Solving the Point Reactor Neutron Kinetics Equations, *Ann. Nucl. Energy*, 36 (2009), 4, pp. 427-432

[8] Shampine, L. F., ACM Transactions on Mathematical Software, 1982, 8, pp. 93-113

[9] Press, W. H., Teukoisky, S. A., Vetterling, W. T., Flannery, B. P., Numerical Recipes in FORTRAN, the Art of Scientific Computing, 2nd ed., Cambridge University Press, Cambridge, UK, 1992, pp. 731–735

[10] Kaps, P., Rentrop, P., Generalized Runge Kutta Methods of Order Four with Step Size Control for Stiff ODE's, *Numer. Math., 33* (1979), pp. 55-68

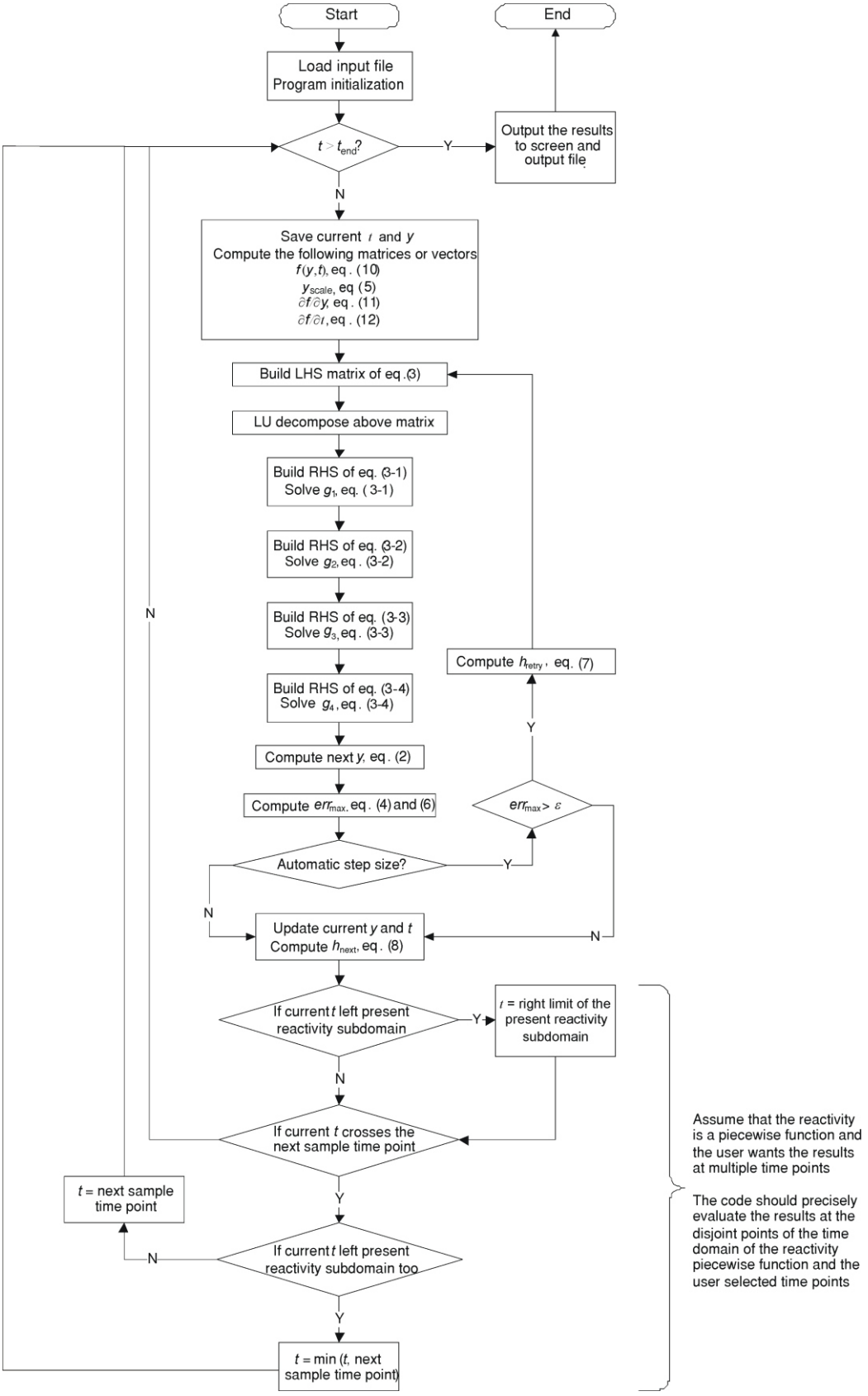
[11] da Nobrega, J. A. W., A New Solution of the Point Kinetics Equations, *Nucl. Sci. Eng., 46* (1971), 3, pp. 366-375

[12] Hetrick, D. L., Dynamics of Nuclear Reactors, The University of Chicago Press, Chicago, Ill., USA, 1971

[13] Satvat, S., Perry, J., Yang, X, Xiao, S., Jevremovic, T., Development of Educational Undergraduate Course Modules for Interactive Reactor Physics, ANS Annual Meeting, Atlanta, Geo., USA, June, 2009

APPENDIX

Flowchart of FORTRAN90 Rosenbrock Methodology for Reactor Point Kinetics



Шуе ЈАНГ, Татјана ЈЕВРЕМОВИЋ

РОЗЕНБРОКОВО НУМЕРИЧКО РЕШЕЊЕ КИНЕТИЧКЕ ЈЕДНАЧИНЕ РЕАКТОРА СА ПРИМЕРИМА

У овом раду описана је Розенброкова метода четвртог реда са аутоматском контролом величине интервала и њена примена у решавању кинетике нуклеарних реактора. На основу ове методе развијен је нумерички програм заснован на рачунарском језику ФОРТРАН 90 и тестирани су његова брзина и тачност. Приказано је више примера који укључују различит тип промене реактивности реактора. У свим тестовима, описана процедура решења кинетике реактора на основу Розенброкове методе показала је високу тачност, а нумерички програм изузетну брзину решења.

Кључне речи: кинетика реактора, Розенброкова метода, ФОРТРАН 90