

APPLICATION OF BACKWARD DIFFERENTIATION FORMULA TO SPATIAL REACTOR KINETICS CALCULATION WITH ADAPTIVE TIME STEP CONTROL

CHEON BO SHIM¹, YEON SANG JUNG¹, JOO IL YOON², and HAN GYU JOO^{1,*}

¹Seoul National University

599 Gwanak-ro, Gwanak-gu, Seoul, 151-744, Korea

²KEPCO Nuclear Fuel Company

1047 Daedeok-daero, Daejeon, 305-353, Korea

*Corresponding author. E-mail : joohan@snu.ac.kr

Received January 26, 2011

Accepted for Publication June 06, 2011

The backward differentiation formula (BDF) method is applied to a three-dimensional reactor kinetics calculation for efficient yet accurate transient analysis with adaptive time step control. The coarse mesh finite difference (CMFD) formulation is used for an efficient implementation of the BDF method that does not require excessive memory to store old information from previous time steps. An iterative scheme to update the nodal coupling coefficients through higher order local nodal solutions is established in order to make it possible to store only node average fluxes of the previous five time points. An adaptive time step control method is derived using two order solutions, the fifth and the fourth order BDF solutions, which provide an estimate of the solution error at the current time point. The performance of the BDF- and CMFD-based spatial kinetics calculation and the adaptive time step control scheme is examined with the NEACRP control rod ejection and rod withdrawal benchmark problems. The accuracy is first assessed by comparing the BDF-based results with those of the Crank-Nicholson method with an exponential transform. The effectiveness of the adaptive time step control is then assessed in terms of the possible computing time reduction in producing sufficiently accurate solutions that meet the desired solution fidelity.

KEYWORDS : BDF, Adaptive Time Step Control, Spatial kinetics, CMFD, Nodal Method

1. INTRODUCTION

Development of an efficient nuclear reactor transient analysis module that provides high performance and accuracy is one of the most important issues in computational reactor physics. As far as the temporal solution scheme is concerned, a higher order method that allows larger time step sizes for the required solution fidelity should be the first element of an efficient transient analysis module. Another would be an adaptive time step control scheme that dynamically adjusts time step sizes according to the characteristics of the transient.

Among the numerous temporal schemes for solving the time-dependent three-dimensional (3-D) diffusion equation, the Crank-Nicholson method with an exponential transform¹ is considered as a quite accurate higher order method and is widely used. This method is particularly effective for the transients involving an exponentially rising period such as the one encountered during a control rod

ejection accident initiated at a hot-zero-power (HZP) condition. However, an adaptive step size control scheme is not derivable with this method, and thus a predefined set of time step sizes has to be specified by the user. With the ordinary Crank-Nicholson method, adaptive time step control is possible using the error estimates obtained from two different time step size trial runs². However it is only second-order accurate. On the other hand, multistep methods, such as the Runge-Kutta methods, render higher accuracy as well as a provision for adaptive step size control³. They are not, however, practical in realistic transient calculations involving thermal-hydraulic feedback because they require multiple evaluations of solution parameters within a time step. As an effort to accomplish the two goals of higher order accuracy and adaptive time step control, the backward differentiation formula (BDF) method⁴ is investigated here for applications to 3-D kinetics calculations with thermal feedback.

The BDF method is an implicit multipoint method

that uses known solution data determined at several previous time points as well as the unknown solution at the current point. The order of the error term can be easily adjusted by choosing a different number of previous data points. This method is known to provide good solution stability in solving stiff differential equations for which an implicit solution is essential⁵. Since the time-dependent multigroup neutron diffusion equation is a stiff differential equation⁶ because of the short neutron generation time and an implicit solution is essential due to the negative thermal effect, the BDF method would be well suited for the solution of spatial reactor kinetics problems. Another distinct advantage of this method is that the information about the order of the error can be used to estimate the error at the current time point. Thus, this method can provide a convenient means of error quantification that can be used in the determination of a proper step size.

There was an earlier study by Ginestar *et al.*⁷ about the application of the BDF method to the solution of the time-dependent neutron diffusion equation. In that study, the BDF formulation was applied to the ordinary differential equation for flux, which is obtained after spatial discretization of the time-dependent neutron diffusion equation. The precursor balance equation was excluded from the BDF solution process by employing the analytic precursor integration procedure. Their BDF solution was based on a fixed time step size. They introduced the so-called variable step scheme that changes selectively the order of BDF - out of 1, 2 and 4 - on the fly depending on the extent of change. Compared to a variable time step scheme, however, this variable step (or order) scheme would be much less flexible and efficient. Another application of the BDF method to spatial kinetics calculations was made using a public domain ordinary differential equation solver. Garcia *et al.*⁸ applied the FCVODE solver, the FORTRAN version of the CVODE solver⁹, which has a generic BDF solver capable of adaptive time step control. To meet the preset requirements of the generic solver, the finite difference method was used for spatial discretization. Since a nodal method has to be used for efficient reactor calculation and the nonlinear thermal-hydraulic feedback effect must be included in a realistic reactor core calculation, such a generic ODE solver that would require preset forms of the equation and coefficients is not well suited for an efficient and realistic reactor kinetics calculation. Rather, it is better that a reactor code specific BDF method is

established with the capability of adaptive time step control.

In this regard, the first objective of the work here is to devise an efficient implementation of the method BDF in spatial kinetics calculation and then to evaluate its performance. The second objective is to establish and assess a BDF-based adaptive time step control scheme. In the following section, a backward differentiation formula is derived for the variable time step case and a generic adaptive time step control method based on the BDF method is presented. The BDF method is applied in Section 3 to three-dimensional kinetics calculation employing the coarse mesh finite difference (CMFD) formulation with the source expansion nodal method (SENM) kernels¹⁰ and a practical adaptive time control scheme for the BDF-based spatial kinetics calculation is introduced. The performances of the BDF spatial kinetics calculation and the adaptive time step control scheme are assessed in Section 4 through the analyses of the NEACRP control rod ejection¹¹ and withdrawal¹² benchmark problems. Section 5 concludes the paper.

2. ADAPTIVE TIME STEP CONTROL USING BACKWARD DIFFERENTIATION FORMULA

The backward differentiation formula for the constant time step case can be readily derived and appears in numerous textbooks on numerical solution of ordinary differential equations. However, special attention is needed when time step size is variable. This is essential in adaptive time step control. In the following subsection, a BDF is derived with variable time step sizes and a BDF-based method of adaptive time step control is formulated.

2.1 BDF with Variable Time Step Sizes

The n^{th} order BDF is expressed as the derivative of a time-dependent function $f(t)$ at the current time point, at which the solution is newly obtained, in terms of the n previously known values and the unknown current solution. An n^{th} order polynomial interpolation can be used for determining a BDF. Suppose that the current Time Point (TP) index is i and the solution has been obtained up to TP $i-1$. Refer to Fig. 1 for the definition of time points, step sizes and the indices of n steps.

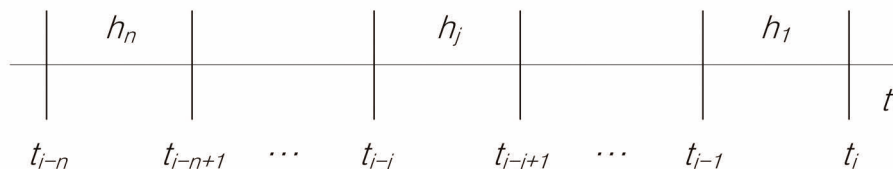


Fig. 1. Time Points and Steps Sizes

Let the n^{th} order Lagrange interpolation function of $f(t)$ be represented by $p_n(t)$ defined below:

$$p_n(t) = \sum_{k=0}^n L_k(t) f_{i-k} \quad (1)$$

where

$$L_k(t) = \prod_{\substack{j=0 \\ j \neq k}}^n \left(\frac{t - t_{i-j}}{t_{i-k} - t_{i-j}} \right). \quad (2)$$

The derivative of $p_n(t)$ at t_i is obtained as follows:

$$\begin{aligned} p'_n(t_i) &= \left(\frac{1}{h_1} + \frac{1}{h_1 + h_2} + \cdots + \frac{1}{h_1 + h_2 + \cdots + h_n} \right) f_i \\ &+ \left(-\frac{1}{h_1} \right) \left(\frac{h_1 + h_2}{h_2} \right) \left(\frac{h_1 + h_2 + h_3}{h_2 + h_3} \right) \cdots \left(\frac{h_1 + h_2 + \cdots + h_n}{h_2 + h_3 + \cdots + h_n} \right) f_{i-1} \\ &+ \left(-\frac{1}{h_1 + h_2} \right) \left(-\frac{h_1}{h_2} \right) \left(\frac{h_1 + h_2 + h_3}{h_3} \right) \\ &\quad \left(\frac{h_1 + h_2 + h_3 + h_4}{h_3 + h_4} \right) \cdots \left(\frac{h_1 + h_2 + \cdots + h_n}{h_3 + h_4 + \cdots + h_n} \right) f_{i-2} \\ &+ \cdots \\ &+ \left(-\frac{1}{h_1 + h_2 + \cdots + h_n} \right) \left(-\frac{h_1}{h_2 + h_3 + \cdots + h_n} \right) \\ &\quad \left(-\frac{h_1 + h_2}{h_3 + h_4 + \cdots + h_n} \right) \cdots \left(-\frac{h_1 + h_2 + \cdots + h_{n-1}}{h_n} \right) f_{i-n} \end{aligned} \quad (3)$$

Thus the BDF is given as

$$f'(t_i) \approx p'_n(t_i) = \sum_{k=0}^n a_k f_{i-k}, \quad (4)$$

where the BDF coefficients, a_k 's, are

$$a_k = \begin{cases} \sum_{l=1}^n \frac{1}{\sum_{j=1}^l h_j}, & k=0 \\ (-1)^k \frac{\prod_{l=1, l \neq k}^n \sum_{j=1}^l h_j}{\prod_{l=1}^k \sum_{j=l}^k h_j \times \prod_{l=k+1}^n \sum_{j=k+1}^l h_j}, & k=1 \sim n \end{cases}. \quad (5)$$

Note that it is natural that this formula can be applied to the problem of a constant step size as well.

2.2 Time Step Size Control with BDF

The step size control method determines the current step size using the estimate of the error at the end of the current time step. When the estimated error is larger than

the required value, the step size is reduced and vice versa. Although there are some sophisticated step size control methods available for BDF^{9,13}, those methods may not necessarily be the best in spatial kinetics calculations in which the nonlinearity originating from thermal feedback and local nodal solutions can be readily resolved. For instance, in the CVODE solver of the SUNDIALS package⁹, the Newton iteration scheme is employed to resolve the nonlinearity originating from the implicit BDF formulation. The Newton iterative solutions are used as the predictor and the corrector solutions needed for determining local truncation errors. In addition, the order of the BDF is allowed to vary in order to maximize the allowable time step size. Furthermore, several heuristically determined control parameters, such as the safety factor of 1/6, the maximum step size reduction factor of 0.2 to be applied after two error test failures, and the minimum step size reduction factor 0.1 to be applied after three error test failures, are used. Since the Newton iteration is not necessary in the CMFD-based spatial kinetics calculation scheme used in this work, the local error estimation scheme using the Newton iterates and the associated variable order BDF scheme would not be beneficial here. Moreover, the heuristically determined control parameters of a generic adaptive step control scheme are not necessarily optimal in spatial kinetics calculations. In this regard, a simpler and potentially more efficient step size control method is derived below. The starting point of the derivation is to note that the local truncation error of the n^{th} order BDF solution at the current time point can be estimated after performing an additional BDF calculation with the order of $n-1$.

2.2.1 Basic Idea of Step Size Control

Given the desired error tolerance ε , the required time step size can be determined for an n^{th} order method as follows:

$$h_i = s_i h'_i = \left(\frac{\varepsilon}{e_i} \right)^{\frac{1}{n}} h'_i \quad (6)$$

where

h_i = time step size at the i^{th} step,

s_i = step size adjustment factor at the i^{th} point,

e_i = relative error of the solution vector at the i^{th} point obtained with h'_i , and

h'_i = trial value of h_i .

Note that the solution appears as a vector in general because a system of ordinary differential equations is solved in most cases including the spatial kinetics problem. Here the concern is now how to estimate the relative error of the solution vector. In the following, the derivation is, however, given as if the unknown function is a single scalar function, but it is meant that the notation for a function appearing below represents a vector of functions.

Consider the solution of the following equation at TP

i supposing that the correct solution is available at TP $i-1$:

$$y'(t) = f(t, y) \quad (y(t_{i-1}) = y_{i-1}). \quad (7)$$

The error at the i^{th} point obtained with an n^{th} order method is designated by

$$e_i^{(n)}(h) = y_i^{(n)} - y_i, \quad (8)$$

where y_i is the true solution and $y_i^{(n)}$ is the estimated solution obtained by the n^{th} order method. This local error is dependent on the time step size h to the $n+1^{\text{th}}$, namely,

$$e_i^{(n)}(h) = h^{n+1} g^{(n+1)}(\tau_i), \quad (9)$$

where $g^{(n+1)}(\tau_i)$ is the value of the $n+1^{\text{th}}$ order derivative of the solution, $y = g(t)$, at a point τ_i satisfying $t_{i-1} \leq \tau_i < t_i$. For the $n+1^{\text{th}}$ order method, the error is represented as

$$e_i^{(n-1)}(h) = y_i^{(n-1)} - y_i = h^n g^{(n)}(\tilde{\tau}_i) \quad (10)$$

with another $\tilde{\tau}_i$ located within the time step. Using Eq. (10), the error in Eq. (8) can be rewritten as

$$e_i^{(n)}(h) = y_i^{(n)} - y_i^{(n-1)} + h^n g^{(n)}(\tilde{\tau}_i) = h^{n+1} g^{(n+1)}(\tau_i). \quad (11)$$

From this equation, it follows that the difference between the two different order solutions is of order n as

$$y_i^{(n)} - y_i^{(n-1)} = -h^n g^{(n)}(\tilde{\tau}_i) + h^{n+1} g^{(n+1)}(\tau_i) = O(h^n). \quad (12)$$

From the relation of Eqs. (11) and (12), it seems reasonable to take the above difference as an approximation of the error of the n^{th} order solution. Namely,

$$e_i^{(n)}(h) \cong y_i^{(n)} - y_i^{(n-1)} = C(\tau_i) h^n \quad (13)$$

for some constant $C(\tau_i)$. This estimate of the true error would be conservative in that the length of the approximate solution vector would be larger than the true one because the difference consists of only a part of the error. With this larger error, the next time step size will be estimated to be smaller and conservative.

Noting that the order of the local error is reduced to n from $n+1$ by using this approximated error, the following relation is expected with a changed step by a factor α :

$$e_i^{(n)}(\alpha h) \equiv y^{(n)}(t_{i-1} + \alpha h) - y^{(n-1)}(t_{i-1} + \alpha h) = \alpha^n h^n C(\tilde{\tau}_i). \quad (14)$$

In order to have the relative error smaller than the desired tolerance with the adjusted step size, the following relation should hold:

$$\frac{\|e_i^{(n)}(\alpha h)\|}{\|y^{(n)}(t_{i-1} + \alpha h)\|} = \frac{\alpha^n h^n \|C(\tilde{\tau}_i)\|}{\|y^{(n)}(t_{i-1} + \alpha h)\|} \leq \varepsilon. \quad (15)$$

By ignoring the difference between $C(\tau_i)$ and $C(\tilde{\tau}_i)$ for the α values moderately deviating from unity and using Eq. (13), the following relation for the adjustment factor is derived:

$$\alpha^n \leq \frac{\|y^{(n)}(t_{i-1} + \alpha h)\|}{\|y_i^{(n)} - y_i^{(n-1)}\|} \varepsilon. \quad (16)$$

Since $y^{(n)}(t_{i-1} + \alpha h) \cong y_i^{(n)}$, Eq. (16) can be rewritten as

$$\alpha \leq \left(\frac{\varepsilon}{e_i} \right)^{\frac{1}{n}} \quad (17)$$

with the relative error defined by

$$e_i = \frac{\|y_i^{(n)} - y_i^{(n-1)}\|}{\|y_i^{(n)}\|}. \quad (18)$$

2.2.2 Considerations for Stable Step Size Control

There are many assumptions in the above derivation of the step size control method. It implies that the actual relative error can be larger than the given tolerance and a blind application of the above scheme can induce too frequent and alternating changes in the time step size. In this regard, some considerations for stabilization and assurance need to be made in the step size control scheme.

First of all, a safety factor is introduced to make the estimated relative error bigger. In general, this factor ranges from 0.8 to 0.9 and the relative error is divided by it to make it larger. After applying this safety factor, θ , the formula of the step size control method is expressed as

$$h_i = s_i h'_i = \left(\frac{\theta \varepsilon}{e_i} \right)^{\frac{1}{n}} h'_i. \quad (19)$$

This safety factor is set to 0.8 in this work.

In general, it is not good to increase the time step size whenever the step size adjustment factor s_i is greater than 1.0 because it may be less than in the next step because of the current increase. It is thus better to monitor the error trends for several time steps even though it appears to be adequate to increase the current time step. The number of error monitoring steps, a parameter M , is introduced here and it is adjusted depending on the history of time step change. The initial value of M is chosen to be 5 in order to be consistent with the 5th order BDF method, which will be used here. It can be reduced if a monotonous change lasts several time steps and is reset to 5 once the time step size decreases. The detailed algorithm of adjusting M will be discussed along with Fig. 2.

On the other hand, it is desirable to avoid a sudden large increase in the step size. This means that it is better to limit possible increases. The maximum increase factor B is thus introduced to prevent excessively large increases as follows:

$$h_i = \begin{cases} B h'_i & (s_i > B) \\ s_i h'_i & (B \geq s_i > 1.0) \end{cases}. \quad (20)$$

The value of B is set as 1.25 initially, but it can be

increased to 1.5 first and then further to 2.0 when the step size adjustment factor is larger than the conservative factor B during M time steps.

During the course of adaptive time step control, the most significant problem occurs when the estimated relative error is larger than the specified relative error tolerance which means the current time step size is too large to meet the required error criterion. In this case, the solution must be obtained again with a smaller time step size. Since neither too large nor too small reduction is desirable, a fixed reduction of 20% is used in this work as follows:

$$h_i = 0.8h_i' \text{ if } e_i > \varepsilon. \quad (21)$$

With this algorithm as explained so far, the time step size can increase continuously if there is no perturbation which induces a large error. Sometimes, the increase is good because it saves the computing time greatly, but it can cause a serious problem when a sudden perturbation occurs after a long stable period. An example is reactor shutdown during a smooth power increasing period. To prevent this problem, the time step size is restricted not to exceed a preset maximum value h_{max} of which the default value is 1.0 sec. The step size control scheme described above is depicted in Fig. 2.

3. SPATIAL KINETICS SOLUTION USING BDF AND CMFD METHODS

The BDF method can be conveniently implemented in the CMFD formulation of the time-dependent neutron diffusion equation. In the following, time-dependent

CMFD formulation is derived with the temporal differencing performed by the BDF method. The actual adaptive time step control scheme implementing the algorithm of Fig. 2 in the spatial kinetics calculation involving thermal feedback is then presented.

3.1 Discretization of Time-Dependent Multigroup Neutron Diffusion Equation

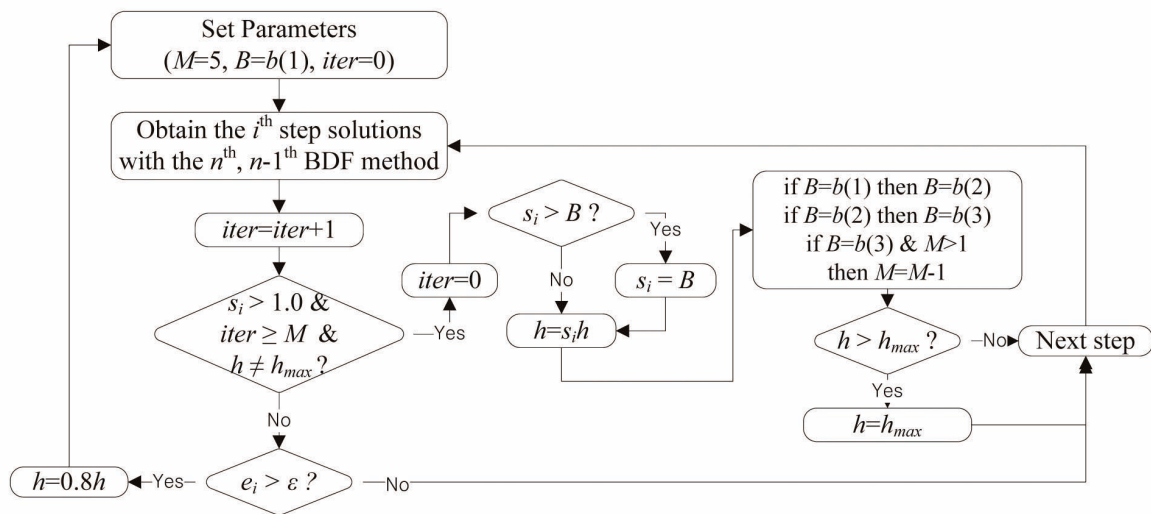
The time-dependent multigroup neutron diffusion equation can be written as follows in terms of the volume averaged flux for Node m and precursor density together with the precursor balance equation:

$$\frac{1}{v_g} \frac{d\phi_g^m(t)}{dt} = \chi_g (1 - \beta_m(t)) \psi_m(t) + \sum_{k=1}^6 \chi_{dgk} \lambda_k C_k^m(t) + \sum_{g'=1, \neq g}^G \Sigma_{g'g}^m(t) \phi_{g'}^m(t) - \left(\sum_{u=x,y,z} \xi_u^m (J_{gu}^{m+}(t) - J_{gu}^{m-}(t)) + \Sigma_{rg}^m(t) \phi_g^m(t) \right) \quad (22)$$

$$\frac{dC_k^m(t)}{dt} = \beta_{m,k}(t) \psi_m(t) - \lambda_k C_k^m(t) \quad (k=1, \dots, 6), \quad (23)$$

where the symbol ψ denotes the fission source and ξ_u^m is the inverse of the node width in u -direction. Using the CMFD relation for the interface current⁷, the current terms J in the loss term of Eq. (22) is expressed as

$$J_{gu}^{m+} - J_{gu}^{m-} = -(\tilde{D}_{gu}^{m-} - \hat{D}_{gu}^{m-}) \phi_g^{m-} - (\tilde{D}_{gu}^{m+} + \hat{D}_{gu}^{m+}) \phi_g^{m+} + (\tilde{D}_{gu}^{m+} + \tilde{D}_{gu}^{m-} - \hat{D}_{gu}^{m+} + \hat{D}_{gu}^{m-}) \phi_g^m. \quad (24)$$



* $b = [1.25 \ 1.5 \ 2.0]$.

Fig. 2. Stabilized Step Size Control Algorithm

$\tilde{D}_{gu}^{m\pm}$ are $\hat{D}_{gu}^{m\pm}$ the *base* nodal coupling and *corrective* nodal coupling coefficients (CNCC), respectively, defined at the positive(+) or negative(-) side surface in the u direction of node m .

After substituting Eq. (24) into Eq. (22), Eqs. (22) and (23) are expressed in terms of matrices and vectors as

$$\frac{1}{v_g} \frac{d\phi_g(t)}{dt} = \mathbf{s}_g(t) - \mathbf{L}_g(t) \phi_g(t) \quad (25)$$

$$\frac{dC_k(t)}{dt} = \beta_k \psi(t) - \lambda_k C_k(t) \quad (k=1, \dots, 6), \quad (26)$$

with M now being the number of total nodes,

$$\phi_g(t) = [\phi_g^1(t), \dots, \phi_g^M(t)]^T \in R^{M,1},$$

$$\mathbf{s}_g(t) = [s_g^1(t), \dots, s_g^M(t)]^T \in R^{M,1},$$

$$\mathbf{C}_k(t) = [C_k^1(t), \dots, C_k^M(t)]^T \in R^{M,1},$$

$$\beta_k \psi(t) = [\beta_k^1(t) \psi_1(t), \dots, \beta_k^M(t) \psi_M(t)]^T \in R^{M,1},$$

$$\mathbf{L}_g = \begin{bmatrix} \mathbf{I}_1 \\ \vdots \\ \mathbf{I}_m \end{bmatrix} \in R^{M,M},$$

$$\mathbf{I}_m = [\dots, -l_{gz}^m, \dots, -l_{gy}^m, \dots, -l_{gx}^m, d_g^m, -u_{gx}^m, \dots, -u_{gy}^m, \dots, -u_{gz}^m, \dots] \in R^{1,M},$$

$$s_g^m(t) = \chi_g (1 - \beta_m(t)) \psi_m(t) + \sum_{k=1}^6 \chi_{dgk} \lambda_k C_k^m(t) + \sum_{g'=1, \neq g}^G \Sigma_{g'g}^m(t) \phi_{g'}^m(t),$$

$$l_{gu}^m = \zeta_u^m (\tilde{D}_{gu}^{m-} - \hat{D}_{gu}^{m-}),$$

$$u_{gu}^m = \zeta_u^m (\tilde{D}_{gu}^{m+} + \hat{D}_{gu}^{m+}), \text{ and}$$

$$d_g^m = \zeta_u^m (\tilde{D}_{gu}^{m+} + \tilde{D}_{gu}^{m-} - \hat{D}_{gu}^{m+} + \hat{D}_{gu}^{m-}) + \Sigma_{rg}^m.$$

By applying the BDF to the LHS derivative of Eq. (25) at the i^{th} time point, the following discretized equation is obtained:

$$\frac{1}{v_g} \left(\sum_{k=0}^n a_k \phi_{g,i-k} \right) = \mathbf{s}_{g,i} - \mathbf{L}_{g,i} \phi_{g,i}. \quad (27)$$

Note that the BDF coefficients, a_i , has the dimension of inverse time because the time step sizes appear in the denominator as indicated in Eq. (5). Now, by collecting

known and unknown terms, the CMFD linear system to be solved appears as

$$\left(\mathbf{L}_{g,i} + \frac{a_0}{v_g} \mathbf{I} \right) \phi_{g,i} = \mathbf{s}_{g,i} - \frac{1}{v_g} \left(\sum_{k=1}^n a_k \phi_{g,i-k} \right) \quad (28)$$

The term $s_{g,i}$ in Eq. (28) contains 6 precursor density terms C_k ($k=1, \dots, 6$), which are also unknowns. These precursor densities can be treated as explicit unknowns by solving the precursor equation simultaneously with the flux equation after applying the BDF temporal differencing to the precursor equation as well. This simultaneous solution scheme would yield a large linear system in which the number of unknowns per node is eight instead of two for a two-group problem. The solution of the linear system thus can be inefficient. This inefficiency can be avoided by noting that the precursor density changes much more slowly than the flux due to the nature of delayed neutron emission, and thus the same order temporal discretization need not be applied to both the flux and the precursor equation. A simpler approach is to use the analytic solution of the precursor equation, which can be obtained once a proper form of the temporal variation of the fission source within the current time step is assumed. If a quadratic variation of the fission source is assumed for the current time step, the precursor density at the current time point can be expressed in terms of the unknown fission source at the current time point and the known fission sources of the two previous time points through the analytic integration procedure. The specific expression is given as follows¹:

$$C_{k,i}^m = \kappa_k C_{k,i-1}^m + \frac{\beta_k}{\lambda_k} \left(\Omega_{k,i-2} \psi_{m,i-2} + \Omega_{k,i-1} \psi_{m,i-1} + \Omega_{k,i} \psi_{m,i} \right). \quad (29)$$

The details of the coefficients can be found in the PARCS code theory manual¹. This approach of analytic precursor integration was also used in the work of Ginestar *et al.*, but they used only first order fission source variation. The adequacy of the second order precursor integration as opposed to the first order integration and to the direct BDF precursor solution approach will be examined in Section 4.1.

After substituting Eq. (29) into the precursor terms of $s_{g,i}$ in Eq. (28), the total delayed neutron source and total source at node m , group g , and time point i are expressed as follows:

$$\sum_{k=1}^6 \chi_{dgk} \lambda_k C_{k,i}^m = s_{dg,i-1}^m + \omega_{g,i} \psi_{m,i}. \quad (30)$$

The total source at the current time step is then represented as

$$s_{g,i}^m = \chi_g (1 - \beta_{m,i}) \psi_{m,i} + s_{dg,i-1}^m + \omega_{g,i} \psi_{m,i} + \sum_{g'=1, \neq g}^G \Sigma_{g'g}^m \phi_{g',i}^m, \quad (31)$$

$$= \tilde{s}_{g,i}^m + s_{dg,i-1}^m$$

where

$$s_{dg,i-1}^m = \sum_{k=1}^6 \chi_{dgk} \left(\kappa_k \lambda_k C_{k,i-1}^m + \beta_k \left(\Omega_{k,i-2} \psi_{m,i-2} + \Omega_{k,i-1} \psi_{m,i-1} \right) \right),$$

$$\omega_{g,i} = \sum_{k=1}^6 \chi_{dgk} \beta_k \Omega_{k,i}, \text{ and}$$

$$\tilde{s}_{g,i}^m = \left(\chi_g (1 - \beta_{m,i}) + \omega_{g,i} \right) \psi_{m,i} + \sum_{g'=1, \neq g}^G \Sigma_{g'g,i}^m \phi_{g',i}^m$$

Finally, Eq. (28) is simplified as

$$\left(\mathbf{L}_{g,i} + \frac{a_0}{v_g} \mathbf{I} \right) \boldsymbol{\Phi}_{g,i} - \tilde{\mathbf{s}}_{g,i} = \mathbf{s}_{dg,i-1} - \frac{1}{v_g} \left(\sum_{k=1}^n a_k \boldsymbol{\Phi}_{g,i-k} \right). \quad (32)$$

Due to the use of the BDF, RHS of Eq. (32) contains the node average flux terms of the previous n points. Note that it does not involve the term related to the nodal imbalance term of the previous time point, namely, the difference between the production and the loss rates of the previous time point. This tracking of the imbalance term is essential in the Crank-Nicholson method, and it makes the method a little complicated to implement. Because of the absence of the imbalance term, the BDF method is in fact very easy to implement. Only the additional storage required for the previous time point node average flux values and the construction of the RHS terms of Eq. (32) with the BDF coefficients are necessary.

On the other hand, the CNCC, $\hat{D}_{gu}^{m\pm}$, should be calculated by a local higher order nodal calculation. In the construction of the local higher order problem involving the transverse integrated one-dimensional neutron diffusion equation, the transient specific source term that includes the $\frac{a_k}{v_g} \phi_{g,i-k}^m$ term appearing on the RHS of Eq. (32), is placed on the RHS as a function of the intra-nodal position. If this intra-nodal dependence is to be treated accurately, all coefficients of the higher order nodal solution, which determines the intra-nodal flux shape, should be stored and used. This requirement would pose a significant problem in the case of BDF because n previous fluxes are used on the RHS, and thus that much more storage and operations are required. In order to avoid this problem, a practical iterative solution approach is introduced here. This scheme moves the $\frac{a_0}{v_g} \phi_{g,i}^m$ term of the LHS to the RHS and uses the previous iterative value obtained at the current time point so that the transient specific source term becomes

$$s_{g,i}^{m,TS} = -\frac{1}{v_g} \left(a_0 \phi_{g,i}^{m,j} + \sum_{k=1}^n a_k \phi_{g,i-k}^m \right), \quad (33)$$

where j is the index of the nodal calculation to update the CNCC. Note that this term is in fact the BDF form of $\frac{1}{v_g} \frac{d\phi_g}{dt} \Big|_{t=t_i}$ which is small due to the large v_g values. A similar

idea of this difference formulation can be found with more details in the transient CMFD formulation for the Crank-Nicholson method with exponential transform given elsewhere¹. The transient specific source terms of Eq. (33) given by the difference is small in size and thus the impact of the less accurate intra-nodal shape is not great. With this rationale, the transient fixed term is now added to the transverse leakage source term and is represented as a simple quadratic polynomial as in the standard nodal methods. The higher solution for the one-dimensional equation is obtained by the source expansion nodal method (SENM)⁷, which represents the source term, including the fission source term, as a quartic order polynomial that leads to the exponential functions in the homogeneous part of the solution.

The CMFD- and BDF-based transient calculation method was implemented in the RENSUS multigroup nodal code⁷, which has a simplified closed channel thermal-hydraulics calculation module that determines the intrapellet fuel temperature distribution for assembly averaged fuel pins. This code performs iterative estimation of the fuel temperature distribution at the current time point to incorporate prompt reactivity feedback due to fuel temperature change. The multiple calculations of the fuel temperature profiles are performed to be properly coordinated with the nodal calculations to update the CNCCs in the CMFD calculation with thermal feedback. The following algorithm is used at each transient calculation step of the RENSUS code for a given time step size:

- 1) Determine the new fuel temperature distribution at the current time point using the power distribution of the previous time point.
- 2) Update the macroscopic cross sections of each node incorporating external perturbations, such as control rod movement and thermal feedback.
- 3) Determine the source terms on the RHS of the CMFD linear system, Eq. (32), including the BDF terms, fission and delayed neutron source terms originating from the previous steps.
- 4) Set up the LHS matrix of the CMFD linear system using the temperature dependent cross sections and the CNCC so far known at each nodal interface.
- 5) Obtain a partially converged solution of the CMFD problem.
- 6) Perform the SENM nodal calculation to update the CNCCs only if there is a need for nodal update due to sufficiently large cross section changes occurring at this step.
- 7) Perform the fuel temperature calculation with the updated power distribution.
- 8) Update the nodal cross sections.
- 9) Return to Step 4 if the flux convergence is not met. Otherwise proceed to the next step.
- 10) Determine the core power level and relative power distribution.

- 11) Perform a full T/H calculation to update the coolant temperature profile.
- 12) Update the precursor number density distribution.
- 13) Proceed to the next time step.

Note that Step 6 above realizes the so-called conditional nodal update scheme, which performs the nodal calculation to update the CNCC only when there are sufficient changes in the cross sections since the last update. This scheme allows fast transient calculations by effectively skipping expensive nodal calculations. The accuracy and efficiency of this scheme has been well verified elsewhere¹.

3.2 Step Size Control Algorithm

For step size control by the scheme presented in Section 2.2, the solution at the current time needs to be obtained two times: first with the n^{th} order BDF coefficients and then with the $n-1^{\text{th}}$ order. Since the time advancement to the next time step is allowed only when the estimated solution error is less than the specified value, Steps 10 through 13 in the above algorithm should be reserved during the step size determination process. Once the calculation for the current time step is initiated with an estimated step size, the first calculation with the n^{th} order can thus proceed to Step 10 and the second calculation needs to be started from Step 3 because the BDF coefficients change while the time step size remains the same. In the second calculation with the $n-1^{\text{th}}$ order, any variable updated during Steps 3 through 10 should be reset to the beginning of the time step value so that the second calculation can be performed at the same condition as the first one. Once the two solutions are obtained at Step 10, the error of the n^{th} order solution needs to be estimated as explained in the following.

In a three-dimensional multi-group kinetics problem, the primary solution is the flux vector consisting of nodewise multi-group fluxes. However, a more meaningful solution is the nodewise fission source, which includes the fission cross sections because it reflects the power generation rate. In addition, the dimension of the fission source vector is much smaller than the flux vector because there is no energy group dependency. For these two reasons, the fission source vector is taken as the solution and the L_2 norm of the difference of the two fission source vectors obtained with the n^{th} and $n-1^{\text{th}}$ BDF schemes is used as the global error estimator in this work.

The relative error to be used in Eq. (18) is defined as

$$e_i = \frac{\sqrt{\sum_{m=1}^M (\psi_i^{m,(n)} - \psi_i^{m,(n-1)})^2}}{\sqrt{\sum_{m=1}^M (\psi_i^{m,(n)} - \psi_0^m)^2}}, \quad (34)$$

where ψ_0 is taken as the fission source vector of the initial steady state if the initial core power is larger than 1%. Otherwise it is a null vector. This measure is needed to

define the error relative to the increment from the steady-state values when the initial core power is substantial while the increase is marginal. The control rod ejection from hot-full-power (HFP) conditions is a good example of these cases.

If the step size adjustment factor determined by Eqs. (19) and (34) is less than unity, the step size should be reduced and the calculation must be restarted from Step 1 of the previous algorithm with a reduced time step size. The detailed calculation algorithm with the CMFD calculation with step size control is given in Fig. 3.

In this figure, the shaded and hashed blocks are relevant to the BDF method and step size control. If the step size control is not turned on, the hashed blocks are skipped. In principle, the nodal and T/H calculations are also needed to be performed for the $n-1^{\text{th}}$ order calculation. Note, however, that the CNCC and the fuel temperature profiles for the $n-1^{\text{th}}$ order case are not very different from those of the n^{th} order result, so these calculations are skipped to save the computing time as identified by the conditional “ $n=n_{bdf}$?” appearing in the middle of the first column of blocks of Fig. 3. Here n_{bdf} is the order of BDF, which is normally 5. At the second calculation of each time step having a lower BDF order, namely, $n_{bdf}-1$, the conditional is not satisfied and the right hand side branch which does not involve the nodal and T/H calculation is executed. On the other hand, during the initial 5 time steps, the value of n_{bdf} increases linearly from 1 to 5 because there are not enough previous points during those initial steps.

4. PERFORMANCE EXAMINATION

As stated earlier, the Crank-Nicholson with exponential transform (CNET) method renders quite accurate solutions for most transients particularly for the cases involving an exponential rising period. Since the newly introduced BDF method should be as effective as the state-of-the-art method, its accuracy is first assessed against the CNET method using the RENU code, which has the CNET option as well as the regular Crank-Nicholson (CN) option. In this comparison, which will be shown in the following subsection, the same set of predefined time step sizes are used for both methods. The two important transients of control rod ejection and uncontrolled withdrawal are examined with the NEACRP benchmark problem specification^{7,8}. Note that these benchmark problems are used to demonstrate the transient capability of industry standard codes such as ANCK¹⁴. A more complete description of these benchmark problems and typical comparison results can be found elsewhere¹¹. The effectiveness of the adaptive time step control solving these problems is then assessed in the second and third subsections, first for the ejection problems and then for the withdrawal problems.

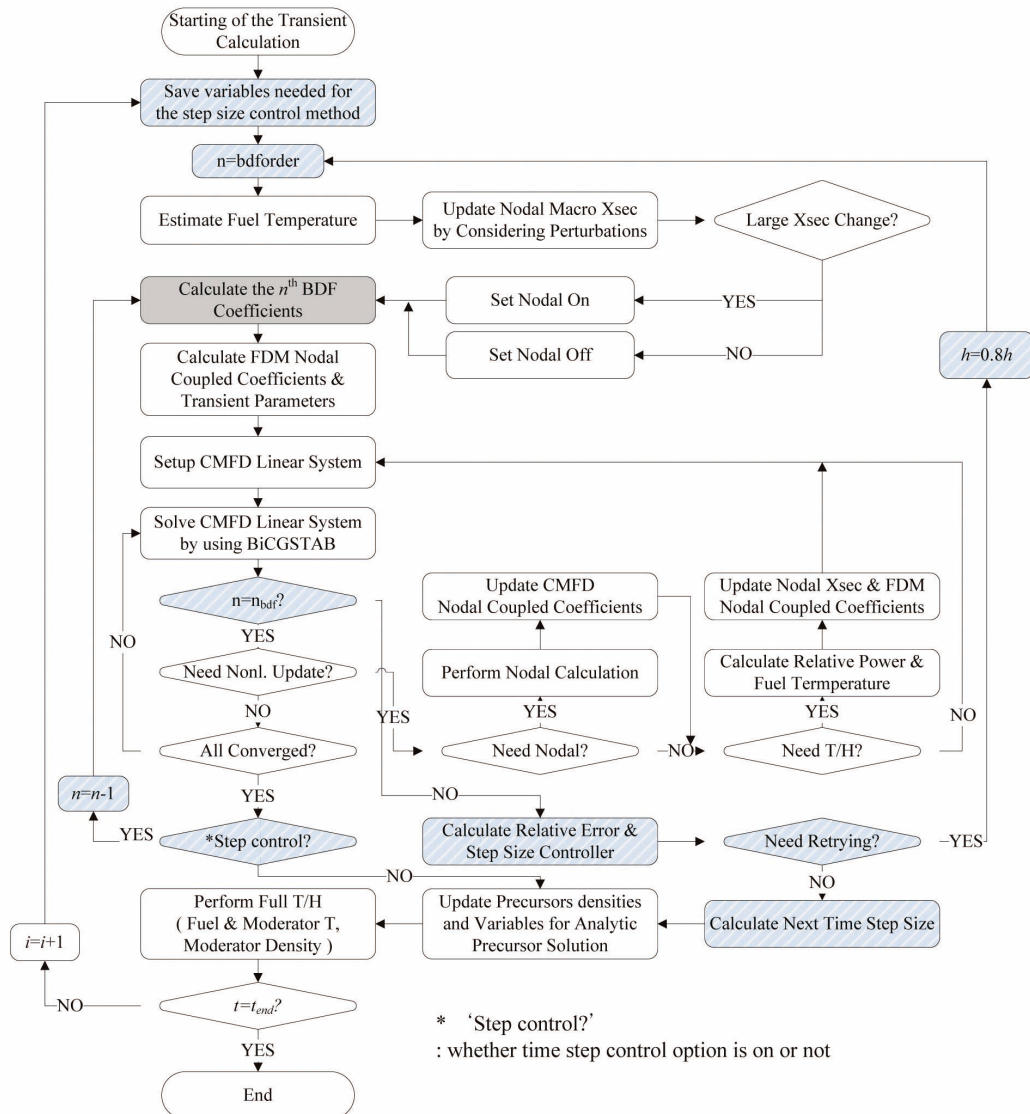


Fig. 3. CMFD Calculation Flow at Each Time Step with Step Size Control

4.1 Accuracy of the 5th Order BDF Solution

The accuracy of the RENU solutions is demonstrated for the A1 and C1 HZP control rod ejection problems. The core solutions shown in Fig. 4 were obtained with the CN option and a time step size of 1 ms. The excellent agreement with the reference PANTHER solution¹⁵ indicates the soundness of the CMFD based solution method of RENU. For the same problems, the solutions of the BDF method of order 5 are compared with the CNET solution in Fig. 5. The choice of BDF order 5 was dictated first by its popular use, but it was also confirmed in the RENU solution for the C1 problem with a time step size of 10 ms that the 6th

order BDF solution produces an oscillatory solution beyond the peak and the 7th and 8th order solutions are totally wrong. This is due to the instability of excessively high order BDF.

Fig. 5 reveals that the accuracy of BDF is comparable to CNET for the A1 case regardless of the time step size. In the C1 case in Fig. 6, however, the 20 ms BDF calculation predicts the peak earlier and reveals an oscillatory behavior after the peak whereas the corresponding CNET case still shows a stable behavior despite some over-prediction of the peak power. This is because the ejected rod worth is higher in the C1 case (1.25\$ for C1 vs. 1.08\$ for A1) so

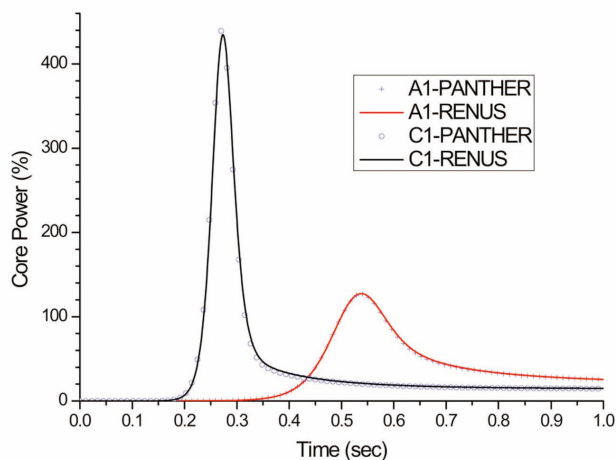


Fig. 4. Accuracy of RENSUS Solutions for A1 and C1 Ejection Problems

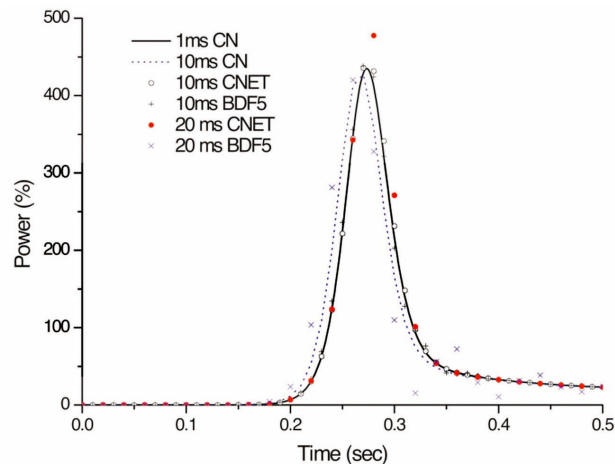


Fig. 6. Comparison of BDF and CNET Solutions for C1

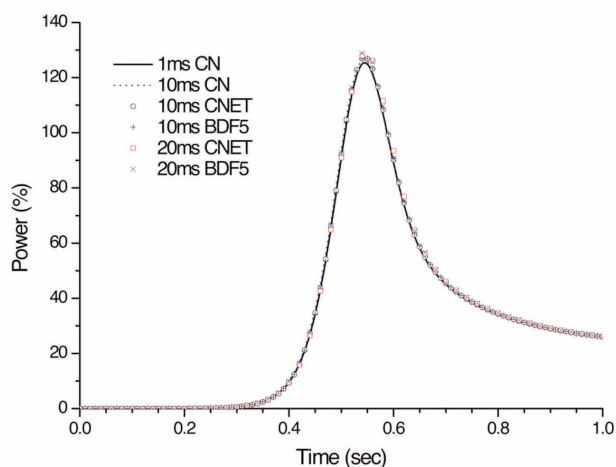


Fig. 5. Comparison of BDF and CNET Solutions for A1

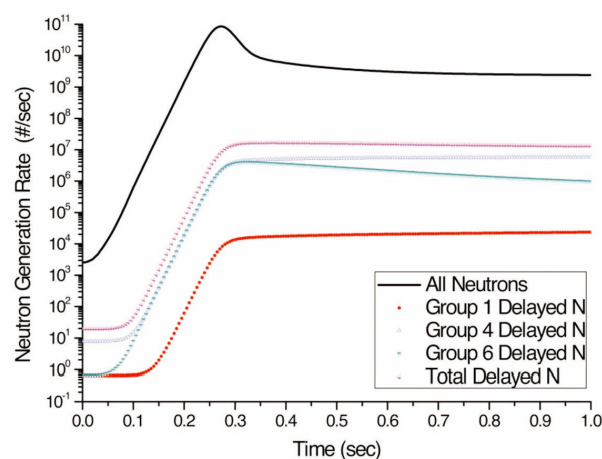


Fig. 7. Behavior of All Neutron and Delayed Neutron Sources at Ejected Rod Position

that the transient progress is much faster. Note that the ordinary Crank-Nicholson case without exponential transform reveals an earlier peak with 10 ms while the corresponding cases with BDF5 and CNET are still good. This degradation in the accuracy of the BDF method with 20 ms is not considered significant because a time step size of 20 ms is in fact quite large for this severe super-prompt critical transient. Later it will be shown that there will be more gain attainable with the adaptive time step control scheme readily available with the BDF method.

Before investigating the effectiveness of the BDF-based adaptive step size control scheme, it would be worthwhile to assess the accuracy of the second order analytic

precursor integration in which the fission source variation is assumed to be quadratic within the current time step. Fig. 7 shows the behaviors of the axially-integrated total neutron source (generation rate) and the delayed neutron sources at the location where the control rod is ejected in the NEACRP C1 problem. This figure clearly indicates that the variation of the delayed neutron sources and consequently that of the precursor densities is much smoother than that of the total fission source. This is because there is a sufficiently long delay between the generation and the destruction of precursors, whereas the response of prompt neutrons to the external change is immediate. This smoother variation provides the rationale that the same 5th order description as the flux equation

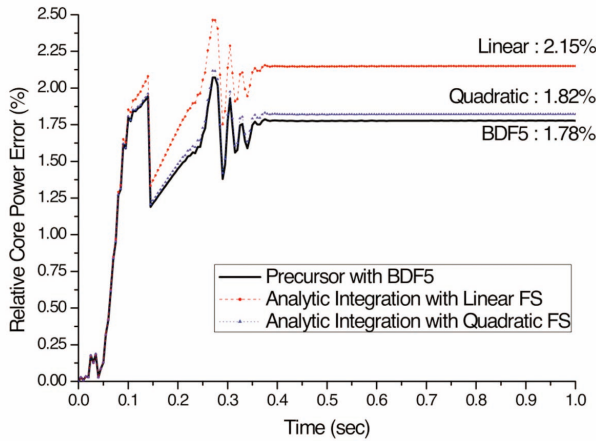


Fig. 8. Core Power Error for Various Precursor Treatments in Point Kinetics Calculation

would not be needed for the precursor equation.

In order to compare the error of the second order precursor integration against the 5th order BDF solution of the precursor equation and also against the first order precursor integration, the point kinetic equation was solved for a problem equivalent to the NEACRP C1 benchmark. Note that the following representations are used for the 5th order BDF and for the first and second order precursor integration:

$$\text{BDF5: } \sum_{j=0}^5 a_{i-j} C_{k,i-j} = \beta_k p_i - \lambda_k C_{k,i} \quad (35)$$

$$\text{Linear: } C_{k,i} = \kappa_k C_{k,i-1} + \frac{\beta_k}{\lambda_k} (\Omega_{k,i-1} p_{i-1} + \Omega_{k,i} p_i) \quad (36)$$

$$\text{Quadratic: } C_{k,i} = \kappa_k C_{k,i-1} + \frac{\beta_k}{\lambda_k} (\Omega_{k,i-2} p_{i-2} + \Omega_{k,i-1} p_{i-1} + \Omega_{k,i} p_i) \quad (37)$$

The point kinetic equation (PKE) is used here for the ease of implementing these schemes, yet it is sufficient to show the effectiveness of each scheme. The relative power errors of the three precursor treatment schemes are shown in Fig. 8. The reference solution for the error estimation was obtained by using a very fine time step size of 0.1 ms while the actual calculations were performed with a time step size of 5 ms. The figure indicates that the accuracy of the second order precursor integration is essentially the same as the 5th order BDF solution of the precursor equation while it is substantially better than the first order integration scheme. The second order precursor integration has about 0.04% point larger error than the 5th order BDF solution of the precursor equation. There is no doubt that

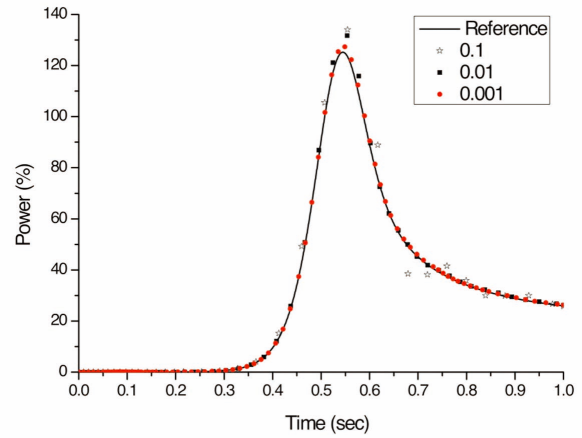


Fig. 9. Core Power Behavior with Various Error Tolerance Values for A1

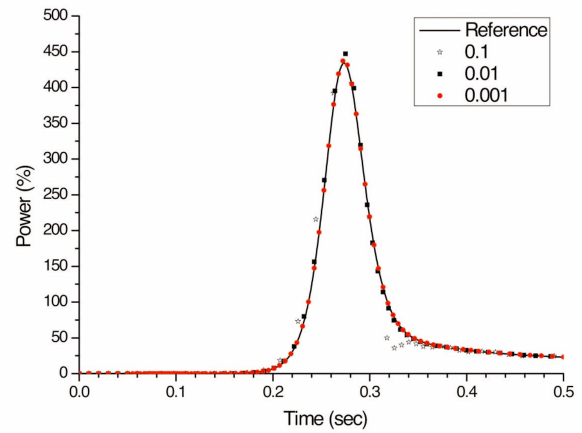


Fig. 10. Core Power Behavior with Various Error Tolerance Values for C1

the gain of the simplification of the final linear system attainable by avoiding the simultaneous solution of the precursor equation as the result of the second order precursor integration is much greater than the negligible loss in solution accuracy.

4.2 Adaptive Time Step Control for Control Rod Ejection Problems

The step size control algorithm presented in Fig. 2 involves a user input value for the desired error tolerance, ϵ . The performance of the adaptive RENSUS calculation was first assessed for the control rod ejection transients A1, C1, and C2 with different values of tolerance: 0.1, 0.01, and 0.001. As shown in Figs. 9 through 11, the

adaptive solution becomes closer to the reference value in all the three cases as the tolerance becomes tighter. Considering the computing time shown in Table 1 as well as the solution accuracy, an error tolerance of 1% was chosen as the optimum value to be used as the default in RENUS.

The time step size variation shown in Fig. 12, which was obtained with an error tolerance of 1% demonstrates the effectiveness of the adaptive time step control in that the time step size monotonously increases to the peak power points, and then it starts to decrease and then to make a turnover to continuously increase in the asymptotic period. The saving in the computing time identified in Table 1 with a tolerance of 1% compared with the fixed time step size case is however marginal for the first 1 second.

However, noting that the control rod ejection calculation should be performed until an asymptotic state is reached for the estimation of the accumulated enthalpy in the

pellet, the calculations were performed up to 20 seconds with a 1% relative error tolerance. As a measure of the overall solution error for the 20 second simulation, the relative root mean square error of the core power defined by the following was estimated.

$$e_G = \frac{\left(\frac{1}{t_{end}} \int_0^{t_{end}} (p(t) - p_{ref}(t))^2 dt \right)^{1/2}}{\frac{1}{t_{end}} \int_0^{t_{end}} (p(t) - p(0)) dt} \quad (38)$$

The integral is evaluated simply by the trapezoidal rule. The overall solution error, the number of time steps and computing time are compared with the results of the fixed time case in which the 10 ms time step is used throughout the transient. In one sense, it may not be fair to compare the fixed time step solution for which a step size of 10

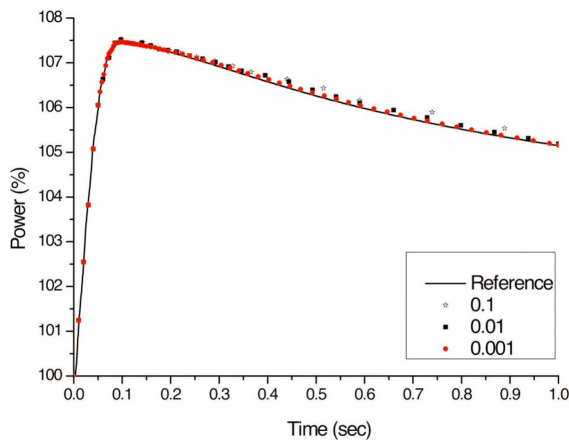


Fig. 11. Core Power Behavior with Various Error Tolerance Values for C2

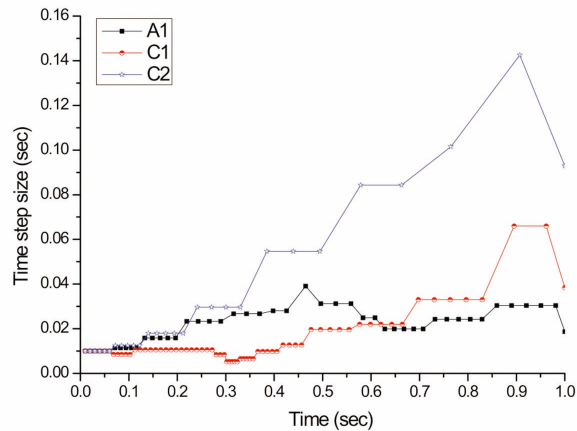


Fig. 12. Step Size Changes for the Rod Ejection Problems

Table 1. Adaptive Time Control Results for Ejection Problems for Initial 1 Second Simulation

Case	Constant Step of 10 ms (100 Time Steps)		Error Tolerance					
	CNET Tcpu ¹⁾	BDF Tcpu	0.1		0.01		0.001	
			N _{TS} ²⁾	Tcpu	N _{TS}	Tcpu	N _{TS}	Tcpu
A1	23.9	19.3	34	11.8	47	15.8	87	24.5
C1	22.6	19.9	49	16.3	67	19.7	115	28.0
C2	16.5	16.0	26	7.1	28	7.6	58	13.0

¹⁾ CPU time in seconds, ²⁾ Number of time steps

ms is too small in the later phase of the transient, but it should be noted that the time step size structure adequate to the problem can be known only by performing a series of pilot calculations if no adaptive scheme is available.

Table 2 shows that about 250 seconds are spent regardless of the type of problem if a constant step is used. With the adaptive time step control scheme, only about 24 seconds is needed for solving Series 1 HZP problems and 10 seconds for Series 2 HFP problems. Thus it is possible to save the computing time more than 10 times.

4.3 Adaptive Time Step Control for Control Rod Withdrawal Problems

The HZP control rod withdrawal events are much longer lasting transients than the ejection events because the reactivity is inserted gradually as the control rod is being withdrawn. During the initial phase of this transient the core power variation is rather slow because of the insufficient reactivity insertion. The reactor period continuously decreases as times goes on so that the core power eventually reaches tens of percent of the rated power which will induce Doppler reactivity feedback to bring the core power down to lead a pulse type power change. During the course of the transient, the reactor scram signal can be activated due to the overpower trip set. It is generally not known when the scram would occur because the core power behavior is not known until the problem is solved completely. In this regard, this control rod withdrawal problem is quite dynamic and an adaptive time step size control is strongly desired in the analysis.

The RENS calculation results for the NEACRP control rod withdrawal cases A and B are shown in Figs. 13-14. The initial core power of these two transients is $10^{-11}\%$ and there is a factor of 3.5×10^{12} increase in the

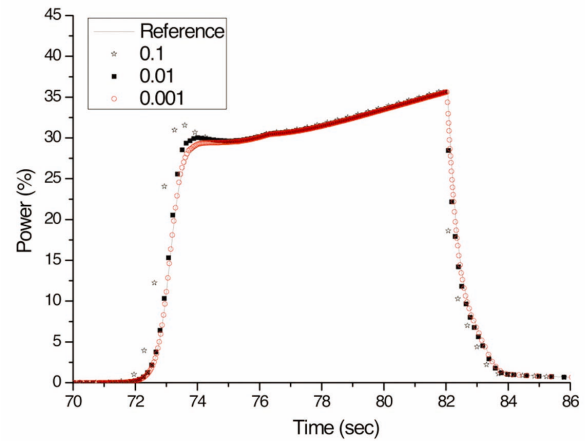


Fig. 13. Core Power Behavior with Various Error Tolerance Values for RWA

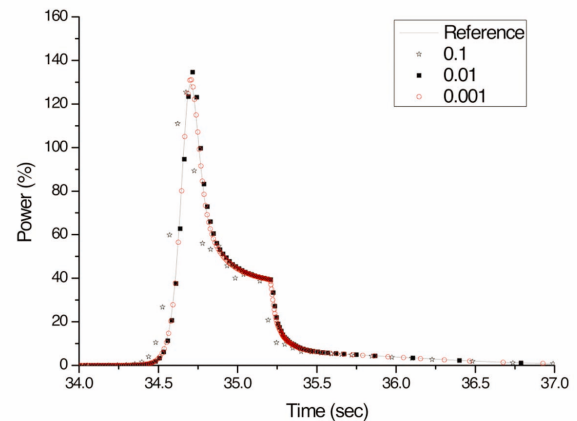


Fig. 14. Core Power Behavior with Various Error Tolerance Values for RWB

Table 2. Number of Time Steps and Computing Time Data for 20 sec Simulations

Case	Fixed ¹⁾ Tcpu ³⁾	Overall Error, %	Adaptive ²⁾		Ratio	
			N _{TS} ⁴⁾	Tcpu	N _{TS}	Tcpu
A1	254.7	1.97	77	24.1	26.0	10.6
B1	253.7	1.91	78	24.2	25.6	10.5
C1	257.0	3.64	96	27.5	20.8	9.3
A2	249.8	1.14	49	10.9	40.8	22.9
B2	249.5	0.76	48	10.0	41.7	25.0
C2	251.1	1.26	49	11.0	40.8	22.8

¹⁾ Fixed Time Step Case, ²⁾ Adaptive Time Step Case, ³⁾ CPU time in seconds, ⁴⁾ Number of time steps

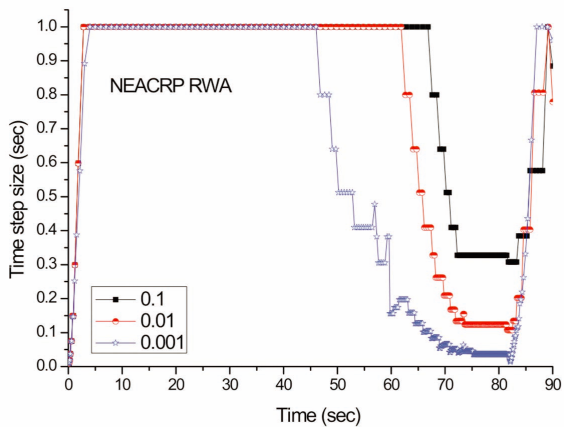


Fig. 15. Step Size Changes with Various Error Tolerance Values for RWA

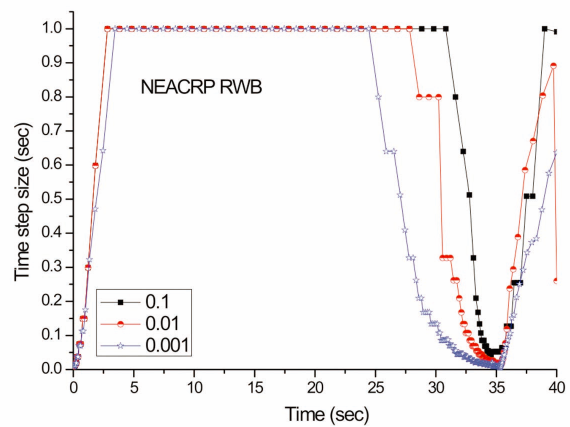


Fig. 16. Step Size Changes with Various Error Tolerance Values for RWB

Table 3. Adaptive Time Control Results for Withdrawal Problems

Case	Constant Time Steps		Error Tolerance					
	CNET Tcpu ¹⁾	BDF Tcpu	0.1		0.01		0.001	
			N _{TS} ²⁾	Tcpu	N _{TS}	Tcpu	N _{TS}	Tcpu
RWA ³⁾	208.8	188.4	152	20.2	226	32.5	538	63.5
RWB ⁴⁾	111.6	104.2	114	17.4	181	28.0	339	47.7

¹⁾ CPU time in seconds²⁾ Number of time steps³⁾ 90 sec simulation with 2700 time steps ($h_1=0.1$ sec from 0 to 70 sec and $h_2=0.01$ sec from 70 to 90 sec)⁴⁾ 40 sec simulation with 1300 time steps ($h_1=0.1$ sec from 0 to 30 sec and $h_2=0.01$ sec from 30 to 40 sec)

core power before the scram. It takes about 70 and 35 seconds to reach this level of multiplication for cases A and B, respectively, as shown in the figures. The reference solutions in these plots were obtained by CNET with 10 ms throughout the transient while the 10 ms time step is used only at the first few time steps in the case of adaptive time step control with BDF.

Similarly to the control rod ejection cases, an error tolerance of 1% gives quite accurate solutions for the withdrawal problems as identified in Figs. 13 and 14. The step size variations shown in Figs. 15 and 16 indicate that the step size starts to decrease much earlier before the peak in the case of a 0.1% error tolerance than the other two cases to get prepared for the rapid rise of the

core flux near the peak. Note that in these plots the step size reduces before the peak, but large time steps are eventually recovered after the peak. The step size could be greater than 1 sec for the initial slowly varying phase, but it is forced to be 1.0 because it is the allowable maximum size which was set so as to prevent excessive increase in spatial kinetics calculations.

With the adaptive time step control, the number of time steps and the computing time can be significantly reduced as demonstrated in Table 3. For a fair comparison with the constant step size case, a larger step size was used for most of the period in the constant step case as identified by the footnotes in Table 3. For the default case of a 1% error tolerance, it turned out that a factor of 5.8 is possible

for the rod withdrawal case A and 3.7 for case B. This significant reduction in the computing time demonstrates the effectiveness of the BDF-based step size control, which would ensure the desired solution fidelity as well.

5. CONCLUSIONS

The Backward Differentiation Formula (BDF) method was successfully applied to spatial kinetics calculation through the use of the coarse mesh finite difference (CMFD) formulation. In this implementation, only the node average flux values of the previous five time points had to be stored additionally and to be considered in the construction of the right hand side (RHS) source term in the temporally discretized nodal balance equation. This simplified BDF implementation was possible by moving the unknown flux term of the current time point to the RHS of the temporally discretized, transverse-integrated one-dimensional neutron diffusion equation and by introducing an iterative scheme to update the current time point flux terms. As a result, the BDF method could be readily implemented into a CMFD-based nodal kinetics code without the complications of calculating and storing the nodal imbalance term, which is inevitable in the Crank-Nicholson method. The fifth order BDF-based spatial kinetics calculations of rapid transients turned out to be as nearly accurate as the state-of-the-art method of Crank-Nicholson with exponential transform as demonstrated by the results of the NEACRP hot-zero-power control rod ejection analyses.

The BDF-based adaptive time step control capability was accomplished by obtaining an additional solution with the fourth order formulation. By taking the difference between the fifth and fourth order solutions, an estimate of the solution error could be obtained, which was used to determine the proper time step size to satisfy the given error tolerance. A practical and stable step size control algorithm for the spatial kinetic calculation with thermal feedback was established based on this error estimate. The computational overhead introduced by using the one lower order solution could be minimized by reusing corrective nodal coupling coefficients determined by the regular order solution.

The performance of the adaptive time step control was examined for the NEACRP control rod ejection and uncontrolled withdrawal events. It was verified that an error tolerance of 1% gave essentially the same solutions as the reference resulting in flexibly adapting step sizes. For a long time simulation up to the asymptotic state of the control rod ejection events, substantial reductions in the computing time ranging from 10 to 20 times were possible. For the long lasting withdrawal transients, a factor of 4 to 6 savings in the computing time was possible with the adaptive time step control. The results obtained

in this work prove that the BDF- and CMFD-based spatial kinetics calculation together with the two order solution-based adaptive time step control enables very efficient yet accurate spatial kinetics calculations.

ACKNOWLEDGEMENTS

The corresponding author thanks Dr. Vyacheslav G. Zimin of the Moscow Engineering Physics Institute who proposed a collaborative research of applying the BDF into spatial kinetics. This work was supported by the Ministry of Education, Science, and Technology of Korea through National Research Foundation of Korea Grant 2009-0083414.

REFERENCES

- [1] H. G. Joo, D. Barber, G. Jiang, and T. Downar, "PARCS: A Multi-Dimensional Two-Group Reactor Kinetics Code Based on the Nonlinear Analytic Nodal Method," PUNE-98-26, Purdue University (1998).
- [2] N. Crouzet and P. Turinsky, "A Second-Derivative-Based Adaptive Time-Step Method for Spatial Kinetics Calculations," *Nucl. Sci. Eng.*, **123**, 206 (1996).
- [3] A. E. Aboanber, Y. M. Hamada, "Generalized Runge-Kutta method for two- and three-dimensional space-time diffusion equation," *Ann. Nucl. Energy*, **35**, 1024-1040 (2008).
- [4] C.W. Gear, "The Simultaneous Numerical Solution of Differential-Algebraic Equations," *IEEE Trans. Circuit Theory*, TC 18 (1) 89-95 (1971).
- [5] L. F. Shampine, "Numerical solution of ordinary differential equations," p. 181, Chapman & Hall (1994).
- [6] Y. A. Chao and A. Attard, "A Resolution of the Stiffness Problem of Reactor Kinetics," *Nucl. Sci. Eng.*, **90**, 40-46 (1985).
- [7] D. Ginestar, G. Verdu, V. Vidal, R. Bru, J. Marin, and J. L. Munoz-Cobo, "High order backward discretization of the neutron diffusion equation", *Ann. Nucl. Energy*, Vol 25, pp. 47-64 (1998).
- [8] Victor M. Garcia, V. Vidal, G. Verdu, J. Garayoa, and R. Miro, "Parallel Resolution of the Two-Group Time Dependent Neutron Diffusion Equation with Public Domain ODE Codes", *Lecture Note in Computer Science*, Vol. 3402, pp. 457-497 (2005).
- [9] Alan C. Hindmarsh, Peter N. Brown, Keith E. Grant, Steven L. Lee, Radu Serban, Dan E. Shumaker, and Carol S. Woodward, "SUNDUALS : Suite of Nonlinear and Differential/Algebraic Equation Solvers", *ACM Transactions on Mathematical Software*, Vol. 31, No. 3, PP 363-396 (2005).
- [10] J. I. Yoon and H. G. Joo, "Two-Level Coarse Mesh Finite Difference Formulation with Multigroup Source Expansion Nodal Kernels," *J. Nucl. Sci. Technol.*, **45**, 668-682 (2008).
- [11] H. Finnemann and A. Galati, "NEACRP 3D LWR Core Transient Benchmark-Final Specification", NEACRP-L-335 (Rev. 1), Nuclear Energy Agency Committee on Reactor Physics (1992).
- [12] R. Fraikin, "PWR Benchmark on Uncontrolled Rods Withdrawal at Zero Power," Final Report, NEA/NSC/DOC(96)20 (1997).
- [13] M. Zhuang and W. Mathis, "Research on Stepsize Control

- in the BDF Method for Solving Differential-Algebraic Equations,” IEEE Proceedings (ISCAS94) 229-232 (1994).
- [14] S. Aoki, T. Suemura, J. Ogawa and T. Takeda, “The Verification of 3 Dimensional Nodal Kinetics Code ANCK Using Transient Benchmark Problems,” *J. Nucl. Sci. Technol.*, **44**, 862-868 (2007).
- [15] M. P. Knight and P. Bryce, “Derivation of a refined PANTHER solution to the NEACRP PWR rod ejection transients,” *Proc. Joint Int. Conf. Mathematical Methods and Supercomputing for Nuclear Applications*, Saratoga Springs, N.Y., 302-313 (1997)