Annals of Nuclear Energy 116 (2018) 280–289

Contents lists available at ScienceDirect

Annals of Nuclear Energy

journal homepage: www.elsevier.com/locate/anucene

An adaptive time step control scheme for the transient diffusion equation

J. Boffie *, J.M. Pounders [1]

Nuclear Engineering Program, Department of Chemical Engineering, University of Massachusetts Lowell, Lowell, MA 01854, USA

ARTICLE INFO

Article history:
Received 20 October 2017
Received in revised form 22 February 2018
Accepted 23 February 2018

Keywords:
Diffusion equation
Time step adaptivitiy
Transient analysis
Stability analysis
Error control

ABSTRACT

The stability and accuracy of an adaptive time step control scheme are analyzed for the transient diffusion equation. This scheme is based on the commonly-implemented backward difference discretization of the diffusion equation and recommends optimal time steps based on constraints applied to estimates of the local truncation error. Methods are derived for both error estimation and error control, each of which potentially impacts the stability of the scheme and the global accuracy of the solution. Asymptotic stability and convergence of the recommended time steps are investigated theoretically and demonstrated numerically to identify optimal realizations of the method. This adaptive time stepping scheme requires no solution evaluations or operator inversions beyond those already performed in the adaption-free solution and requires no modifications to the numerical solution algorithm. As such, this adaptivity scheme can be easily implemented in virtually any reactor physics simulation code based on a backward difference discretization of transient neutronics.

1. Introduction

Adaptive time integration schemes allow codes to optimally select time steps to satisfy some prescribed error tolerance. These schemes provide improved efficiency because time is not wasted solving the system to a level of accuracy beyond relevance. Moreover, adaptive schemes remove the necessity for the user to select time steps before the simulation has been run. For these reasons, adaptive time stepping has received much attention, although typically formal methods have been restricted to multi-step and multi-stage integration methods (e.g., Rabiti et. al., 2005; Shim et. al., 2011) while informal methods are usually based on rather *ad hoc* error criteria. Although there are common approaches for doing this, there are no widespread methods that are easy to implement, and routinely applied to reactor physics calculations.

In transient reactor physics the need for adaptive time stepping is accentuated by the fact that there are multiple time scales involved. From the neutronic perspective, the presence of both prompt and delayed neutrons lead to time scales spanning orders of magnitude, leading to a so-called "stiff" problem in time. In addition, the reactor power is affected by transient thermal and hydraulic conditions that are governed by separate physical processes with independent time scales. These external effects influence the reactor power to a degree that is problem-dependent and thus *a priori* unpredictable. Ideally, a neutronic solver should be able to pick an optimal time step size adaptively based on the current state of a transient. This promotes computational efficiency and is also a key enabling technology for general asynchronous simulations, wherein each constituent physical model is solved only when solution accuracy dictates that it should be solved.

Recently a new method for adaptively selecting time steps for backward Euler (implicit) finite difference solutions of the neutron diffusion equation was developed (Hackemack and Pounders, 2014). The backward difference method was selected because of its abundant use in current reactor physics codes. Moreover, the adaptive method that was developed requires no fundamental solver modification or enhancement and thus can be implemented in a straightforward manner into any existing code that is based on a backward difference time solution.

This new method is conceptually simple, but numerical issues arising from oscillating time step recommendations were observed in some cases, caused by subtle mathematical differences in the formulation of the method (Pounders and Boffie, 2015). In this work, we present combinations of two different error predictors and three different error constraints. Additionally, the asymptotic convergence and stability of each combination is investigated theoretically and numerically to predict under what conditions these oscillations occur.

* Corresponding author.
E-mail address: Joseph_Boffie@student.uml.edu (J. Boffie).
[1] Currently at General Assembly, Atlanta, GA, USA.

https://doi.org/10.1016/j.anucene.2018.02.044
0306-4549/

The scope of the present investigation is restricted to one-dimensional, one-group diffusion theory with time-independent cross sections. These restrictions allow meaningful mathematical conclusions to be drawn in a simple setting. Experience has shown, however, that these results extrapolate to the much more complex configurations of realistic reactor models (i.e., multi-dimensional, multi-group problems with feedbacks).

2. Methods

In this work we consider a simple model of one-group diffusion theory with one group of delayed neutrons and no external feedback,

$$\frac{1}{\upsilon}\frac{\partial\phi}{\partial t} = \frac{\partial}{\partial x}D\frac{\partial\phi}{\partial x} - \sigma_r\phi + (1-\beta)\nu\sigma_f\phi + \lambda C, \tag{1}$$

$$\frac{\partial C}{\partial t} = \beta\nu\sigma_f\phi - \lambda C \tag{2}$$

where all notation is standard (e.g., Bell and Glasstone, 1979). The solution of these equations can be approximated using a backward difference approximation to the time derivative. The finite difference approximation is associated with a sequence of finite time steps which we call $\mathfrak{H} = \{h_j\}_{j=1,2,3,\ldots}$. The time steps are related to the discrete solution points by $h_j = t_j - t_{j-1}$. In methods with adaptive time step selection this sequence is not *a priori* known, but it is determined as the solution evolves so that as few time steps are taken as possible given some constraint on the approximation error.

The backward difference operator may be defined as an approximation of the time derivative at point t_j in terms of the sequence of time steps $\mathfrak{H}$:

$$D_{\mathfrak{H}}u(t_j) = \frac{u(t_j) - u(t_j - h_j)}{h_j} \approx \frac{du}{dt}|_{t_j} \tag{3}$$

where u is the state vector to Eqs. (1) and (2). i.e.

$$u = \begin{bmatrix}\phi\\ C\end{bmatrix} \tag{4}$$

The approximation may be quantified at any given time point by the local truncation error (LTE), which is defined as

$$\tau_j^u = \left.\frac{du}{dt}\right|_{t_j} - D_{\mathfrak{H}}u(t_j) \tag{5}$$

For smooth solutions, it is well-known that the leading term of the truncation error is first-order in the time step and proportional to the second time derivative of the function (e.g. LeVeque, 2007):

$$\tau_j^u = \frac{h_j}{2}\left.\frac{d^2u}{dt^2}\right|_{t_j} + O(h_j^2) \tag{6}$$

In reference (Hackemack and Pounders, 2014), a method was introduced to adaptively select each time step h_j so that the local error, τ_j^u, accrued at each step satisfies some constraint. The local truncation error at each step is approximated by the leading term of Eq. (6),

$$\tilde{\tau}_j^u = \frac{h_j}{2}\left.\frac{d^2u}{dt^2}\right|_{t_j} \approx \tau_j^u \tag{7}$$

There are two fundamental issues that arise in the implementation of this approach, each carrying implications for solution accuracy and stability:

1. The second derivative of the solution must be estimated in an accurate and efficient manner;
2. A suitable constraint on the truncation error estimate must be selected.

Each of these issues is considered in the following subsections.

2.1. Estimating the second derivative

Accurately estimating the second derivative of the solution is critical for a good estimate of the truncation error, which in turn determines the time step recommendation. Two finite difference estimates are presently considered. In both cases it is assumed that the only available solution values are those previously computed by the backward-difference solution, and no additional operator inversions should be required to obtain the second derivative estimate. These restrictions are imposed to make the method immediately applicable for any code currently based on a backward difference solver with minimal or no modification of the solver itself.

The first estimate is based on Taylor expansions, or equivalently a second-order polynomial interpolating the solution at t_{j-1}, t_{j-2} and t_{j-3}. The result is

$$\left.\frac{d^2u}{dt^2}\right|_{t_j} = \frac{2u(t_{j-1})}{h_{j-1}(h_{j-1}+h_{j-2})} - \frac{2u(t_{j-2})}{h_{j-1}h_{j-2}} + \frac{2u(t_{j-3})}{h_{j-2}(h_{j-1}+h_{j-2})} + O(\mathfrak{H}) \tag{8}$$

It is assumed that $O(h_j)$ is independent of j, so the notation $O(\mathfrak{H})$ is used to generically represent the limiting behavior of neglected terms with respect to time step sizes. The second derivative approximation is obtained by neglecting first-order and higher terms in the above expression. This approach will be called the *interpolating* (INT) approximation.

It will later be shown that the interpolating estimate for the second derivative can sometimes lead to oscillations in the time step size. For this reason, an alternative formula has been developed (Pounders and Boffie, 2015) which will be shown to be asymptotically stable in *all* cases presently considered. This second estimate is obtained by dual application of the backward difference operator:

$$\left.\frac{d^2u}{dt^2}\right|_{t_j} \approx D_{\mathfrak{H}}D_{\mathfrak{H}}u(t_{j-1}) \tag{9}$$

This will subsequently be called the *nested difference* (ND) approach. The result in terms of discrete solution values can be worked out in a straight forward manner, and with the assistance of some Taylor expansions we may write

$$\left.\frac{d^2u}{dt^2}\right|_{t_j} = \frac{u(t_{j-1})}{h_{j-1}h_{j-1}} - \frac{(h_{j-1}+h_{j-2})u(t_{j-2})}{h_{j-1}h_{j-1}h_{j-2}} + \frac{u(t_{j-3})}{h_{j-1}h_{j-2}} + O(\mathfrak{H}) \tag{10}$$

In addition to a neglected first-order error, there is also a *grid bias* term, $\varepsilon_{\mathfrak{H}}(t_j)$, that is equal to

$$\varepsilon_{\mathfrak{H}}(t_j) = \frac{(2h_j - h_{j-1} - h_{j-2})}{2h_{j-1}}\left.\frac{d^2u}{dt^2}\right|_{t_j} \tag{11}$$

As the three time steps in this estimate tend uniformly to zero, $\varepsilon_{\mathfrak{H}}(t_j)$ tends to a constant, so the approximation is technically $O(1)$. The practical size of this bias, however, is roughly a function of how rapidly the time steps change over three consecutive steps. In the case of uniform time steps, the bias vanishes completely. If $h_j \approx h_{j-1} \approx h_{j-2}$ then the bias should be small. We presently assume that the latter is the case, and that sufficient programming logic can be added to a code to ensure that this condition is satisfied. Although the nested difference formulation lacks the asymptotic consistency of the interpolating method in the case of a non-