Nuclear Engineering and Design 286 (2015) 130–138

Contents lists available at ScienceDirect

Nuclear Engineering and Design

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

A point implicit time integration technique for slow transient flow problems

Samet Y. Kadioglu[a], Ray A. Berry[b,*], Richard C. Martineau[b]

[a] Department of Mathematical Engineering, Yildiz Technical University, 34210 Davutpasa-Esenler, Istanbul, Turkey
[b] Idaho National Laboratory, P.O. Box 1625, MS 3840, Idaho Falls, ID 83415, USA

HIGHLIGHTS

- This new method does not require implicit iteration; instead it time advances the solutions in a similar spirit to explicit methods.
- It is unconditionally stable, as a fully implicit method would be.
- It exhibits the simplicity of implementation of an explicit method.
- It is specifically designed for slow transient flow problems of long duration such as can occur inside nuclear reactor coolant systems.
- Our findings indicate the new method can integrate slow transient problems very efficiently; and its implementation is very robust.

ARTICLE INFO

Article history:
Received 16 August 2012
Received in revised form 16 January 2015
Accepted 16 February 2015

ABSTRACT

We introduce a point implicit time integration technique for slow transient flow problems. The method treats the solution variables of interest (that can be located at cell centers, cell edges, or cell nodes) implicitly and the rest of the information related to same or other variables are handled explicitly. The method does not require implicit iteration; instead it time advances the solutions in a similar spirit to explicit methods, except it involves a few additional function(s) evaluation steps. Moreover, the method is unconditionally stable, as a fully implicit method would be. This new approach exhibits the simplicity of implementation of explicit methods and the stability of implicit methods. It is specifically designed for slow transient flow problems of long duration wherein one would like to perform time integrations with very large time steps. Because the method can be time inaccurate for fast transient problems, particularly with larger time steps, an appropriate solution strategy for a problem that evolves from a fast to a slow transient would be to integrate the fast transient with an explicit or semi-implicit technique and then switch to this point implicit method as soon as the time variation slows sufficiently. We have solved several test problems that result from scalar or systems of flow equations. Our findings indicate the new method can integrate slow transient problems very efficiently; and its implementation is very robust.

1. Introduction

In this paper, we introduce a new method for time integration of slow transient flow problems. Slow transient flow phenomena can occur in many engineering applications such as the dynamics of fluid flow inside nuclear reactor coolant systems, especially in multiphysics applications where the fluid flow is coupled to other phenomena exhibiting a much slower time scale. As a bounding example, the time scale of the fuel cycle in a nuclear reactor is of order years. Thus the flow undergoes a slow transient, with durations of the order of the other system to which it is coupled, yet the flow equations must be integrated over this long duration in an efficient manner (Williamson et al., 2012). Even the interaction of phases in the relaxation models (Berry et al., 2010) for two-phase flow can be on a much slower scale than the corresponding single-phase dynamics. We remark that fast transient flow dynamics can also occur in a nuclear reactor system. For instance, waterhammer events due to sudden valve closure or steam bubble collapse can produce fast transient wave phenomena, and a sudden power ramp can quickly increase the temperature of the fluid resulting in rapid phase change with subsequent coupling back to the neutronics. In general, the governing equations of flow dynamics are a set of time dependent partial differential equations typically requiring a numerical solution procedure due to lack of sufficient analytical solution ability, e.g. Berry et al. (2014). We would like to

* Corresponding author. Tel.: +1 208 526 1254; fax: +1 208 526 2930.
E-mail addresses: kadioglu@yildiz.edu.tr (S.Y. Kadioglu), ray.berry@inl.gov (R.A. Berry).

http://dx.doi.org/10.1016/j.nucengdes.2015.02.002
0029-5493/

numerically solve these kinds of equations for slowly time-varying problems. Commonly known numerical time integration methods are the *explicit*, *semi-implicit*, *implicit*, or hybrid *implicit–explicit (IMEX) methods* (Thomas, 1998, 1999; Strikwerda, 1989; Leveque, 1998; Wesseling, 2000; Kadioglu and Knoll, 2011). The explicit methods impose stringent stability criteria on time-step sizes that can be impractical for slow transient problems. The implicit methods can take larger time steps. However, other issues such as time inaccuracies with very large time steps, large number of functions evaluations or matrix operations, and robustness issues can be associated with these approaches. Semi-implicit and hybrid IMEX methods can step over certain fine time scales (e.g., ones associated with the acoustic waves), but they still have to follow material Courant time stepping criteria for stability purposes (Kadioglu et al., 2005, 2009, 2010, 2010; Kadioglu and Knoll, 2010, 2011, 2013; Ascher et al., 1995; Ruuth, 1995; Wesseling, 2000). In the past, others have attempted to stabilize explicit time integration methods to permit larger time-steps, e.g. Kujawski (1988), Gnoffo (1990), Thareja and Stewart (1989). Our point implicit method is devised to overcome most of the difficulties listed above. The method treats the solution variables of interest (that can be located at cell centers, cell edges, or cell nodes) implicitly, and the rest of the information related to same or other variables are handled explicitly. The point-wise implicit terms are expanded in Taylor series with respect to the explicit version of the same terms, at the same locations, resulting in a time marching method that is similar to the explicit methods and, unlike the fully implicit methods, does not require implicit iterations. This new method shares the characteristics of the robust implementation of explicit methods and the stability properties of the unconditionally stable implicit methods. This method is specifically designed for slow transient flow problems wherein, for efficiency, one would like to perform time integrations with very large time steps. We have found that the method can be time inaccurate for fast transient problems, particularly with larger time steps. Therefore, an appropriate solution strategy for a problem that evolves from a fast to a slow transient would be to integrate the fast transient with an explicit or semi-implicit technique and then switch to this point implicit method as soon as the time variation slows sufficiently. A major benefit of this strategy for nuclear reactor applications will reveal itself when fast response coolant flow is coupled to slow response heat conduction structures for a long duration, slow transient. In this scenario, as a result of the stable nature of numerical solution techniques for heat conduction one can time integrate the heat part with very large (implicit) time steps. However, such large time-steps cannot normally be efficiently tolerated by flow dynamics solution algorithms. Moreover, one may have to perform the time integration for significantly longer times for these kinds of couplings. Our point implicit method can stably and effectively time integrate the slowly changing, nearly steady-state, flow model with whatever time-step sizes the other physics requires. In addition, the numerical implementation of our method is very robust since one can always call this method from within any solver technology as part of the function evaluation routines.

The organization of this paper is as follows. In Section 2, the governing equations are defined. In Section 3, the numerical solution procedure is described. In Section 4, the computational results are presented. Section 5 contains our concluding remarks.

2. Governing equations

In the scalar test cases, we consider the one-dimensional Burgers equation,

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left[\frac{1}{2}u^2\right] = 0, \tag{1}$$

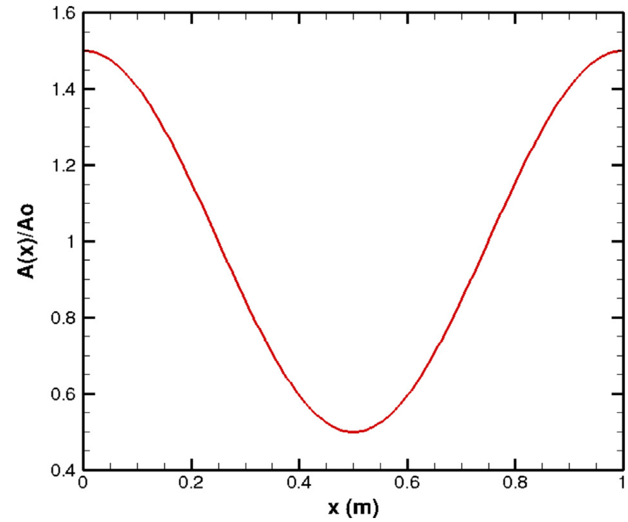

Fig. 1. Normalized nozzle cross-sectional area.

where u corresponds to a nonlinear advection velocity. In the compressible fluid flow system cases, we consider the one-dimensional, variable area, Euler equations,

$$\frac{\partial(\rho A)}{\partial t} + \frac{\partial}{\partial x}[\rho u A] = 0, \tag{2}$$

$$\frac{\partial}{\partial t}(\rho u A) + \frac{\partial}{\partial x}[(\rho u^2 + p)A] = p\frac{\partial A}{\partial x}, \tag{3}$$

$$\frac{\partial(EA)}{\partial t} + \frac{\partial}{\partial x}[u(E+p)A] = 0, \tag{4}$$

where ρ, u, p, E, are the mass density, flow velocity, fluid pressure, and total energy of the fluid, $A = A(x)$ represents the cross-sectional area of the one-dimensional domain. For the demonstration examples used later we use area distribution $A(x)/A_0 = 1 + 0.5\cos(2\pi x)$ on the interval [0, 1], where A_0 is arbitrary (see Fig. 1). E can be related to the other variables by $E = \rho e + 1/2\rho u^2$. We will be primarily employing the stiffened gas equation of state (SGEOS), given by $p = (\gamma - 1)\rho(e - q) - \gamma p_\infty$ where γ, q, and p_∞ are material parameters that can be set to model either a compressible water flow or a compressible gas (vapor) flow (Harlow and Amsden, 1971; LeMetayer et al., 2004). Later, we make use of SGEOS utilizing temperature, $p = (\gamma - 1)\rho c_v T - p_\infty$, where c_v is an additional material parameter.

3. Numerical algorithm

3.1. Point implicit method for the scalar case

Suppose, we rewrite the Burgers equations as

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}[f(u)] = 0, \tag{5}$$

where the flux function $f(u) = \frac{1}{2}u^2$, then we consider the following discretization:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = g(u_{i-1}^n, u_i^{n+1}, u_{i+1}^n), \tag{6}$$

where u_i^n denotes the numerical solution at the ith cell and the nth time level, g corresponds to the spatial discretization of $-\partial f/\partial x$. Notice that g is the function of u at only three stencil points because, for simplicity, we assume that $\partial f/\partial x$ is discretized based on a first-order up-winding scheme. For a second-order scheme, g would be a function of u at five stencil points (e.g.,