JOURNAL OF COMPUTATIONAL PHYSICS **128,** 19–31 (1996)
ARTICLE NO. 0193

Additive Semi-Implicit Runge–Kutta Methods for Computing High-Speed Nonequilibrium Reactive Flows

XIAOLIN ZHONG*

Mechanical, Aerospace and Nuclear Engineering Department, University of California, Los Angeles, California 90095

Received December 12, 1994; revised November 1, 1995

This paper is concerned with time-stepping numerical methods for computing stiff semi-discrete systems of ordinary differential equations for transient hypersonic flows with thermo-chemical nonequilibrium. The stiffness of the equations is mainly caused by the viscous flux terms across the boundary layers and by the source terms modeling finite-rate thermo-chemical processes. Implicit methods are needed to treat the stiff terms while more efficient explicit methods can still be used for the nonstiff terms in the equations. This paper studies three different semi-implicit Runge–Kutta methods for additively split differential equations in the form of $u' = f(u) + g(u)$, where f is treated by explicit Runge–Kutta methods and g is simultaneously treated by three implicit Runge–Kutta methods: a diagonally implicit Runge–Kutta method and two linearized implicit Runge–Kutta methods. The coefficients of up to third-order accurate additive semi-implicit Runge–Kutta methods have been derived such that the methods are both high-order accurate and strongly A-stable for the implicit terms. The results of two numerical tests on the stability and accuracy properties of these methods are also presented in the paper. © 1996 Academic Press, Inc.

1. INTRODUCTION

This paper is concerned with numerical methods for computing stiff equations for transient hypersonic flows with thermo-chemical nonequilibrium. This work is motivated by our studies on the stability and transition of hypersonic boundary layers involving shock interactions and real gas effects [1, 2]. In addition to the effects of viscosity, heat-conduction, and diffusion, hypersonic flows often contain nonequilibrium processes of thermal excitations and chemical reactions because of high gas temperature and high speeds. One of the major difficulties in computing such flows is the stiffness of the governing equations in temporal integrations.

The stiffness is mainly caused by the viscous stress and heat flux terms in the boundary layers and by the source terms modeling finite-rate thermo-chemical processes. The viscous terms across the boundary layer are stiff because fine-grid spacing is used in the direction normal to the wall. Finite difference approximation to the viscous equations with these small-size grids lead to stiff systems of ordinary differential equations. The source terms are stiff because the chemical and thermal nonequilibrium processes have a wide range of time scales, some of which are much smaller than the transient flow ones. As a result, if explicit methods are used to integrate the stiff governing equations, the computations will become very inefficient because the time-step sizes dictated by the stability requirements are much smaller than those required by the accuracy considerations.

In order to remove the stability restriction on the explicit methods, implicit methods need to be used. For computing multidimensional reactive flow, global implicit methods are seldom used because it takes a prohibitively large amount of computer time and large memory to convert full implicit equations. Practical implicit methods for multidimensional reactive flow calculations include the fractional step method (or time-splitting method) and the additive semi-implicit method.

The fractional step methods [3, 4] solve the stiff terms and the nonstiff terms in two independent steps. The results from the partial calculations are combined together after the computations of the individual steps. The time-step restriction by stability conditions is removed by using different methods to compute the stiff and nonstiff terms. The drawback of these methods is that the temporal accuracy is limited to second-order accurate at most if a Strang [5] splitting method [6] is used.

The additive semi-implicit[1] methods, on the other hand, additively split the ordinary differential equations into stiff and nonstiff terms, where the stiff terms are treated implicitly while the nonstiff terms are treated explicitly. The semi-implicit methods are more efficient than the full implicit methods for reactive flow computations because the stiff

* E-mail: xiaolin@seas.ucla.edu.

[1] The term "semi-implicit," which is different from the term "semi-implicit Runge–Kutta methods" defined by Butcher [7–9], is used in this paper following the terminology often used in computational fluid dynamics literatures [10].

terms can be easily separated from the rest of the equations. The standard semi-implicit method for direct numerical simulation of incompressible turbulence is to use the implicit Crank–Nicolson method for the viscous terms normal to the wall and the explicit Adams–Bashford method for the rest of terms [10–14] (ABCN method). For compressible reactive flow, a semi-implicit MacCormack method [15–17, 6] has been used to compute the chemical source terms implicitly while the fluid terms are computed explicitly.

The temporal accuracy of these two methods, however, is usually only second-order accurate at most. To obtain simultaneous high-order accuracy and good stability properties, additive semi-implicit Runge–Kutta methods can be used. The derivation of an additive semi-implicit method with both high accuracy and good stability is not a straightforward task because of the coupling between the explicit and implicit terms. The first additive Runge–Kutta methods for stiff ordinary differential equations were studied by Cooper and Sayfy [18, 19]. They derived additive Runge–Kutta methods to solve a system of differential equations in a form of $x' = J(t)x + g(t, x)$, where the linear term on the right-hand side of the equation was stiff. Their additive methods solve the linear term using implicit A-stable Runge–Kutta methods and solve $g(t, x)$ simultaneously using explicit Runge–Kutta methods. Additive methods of up to fourth order were studied. Recently, Engquist and Sjogreen [20] derived additive semi-implicit Runge–Kutta methods for computing detonation waves. Their third-order schemes are $A(\alpha)$ stable for the stiff source term when the nonstiff term satisfies an explicit stability condition. Other methods for stiff ordinary differential equations were summerized in Hairer and Wanner [21].

Numerical methods for time-accurate computations of nonequilibrium hypersonic flow need to have simultaneous higher-order accuracy and good stiff stability properties. In this paper, three different sets of additive semi-implicit Runge–Kutta (ASIRK) methods are studied for additively split ordinary differential equations in the form of $u' = f(u) + g(u)$, where the nonstiff term f is treated by explicit Runge–Kutta methods, and the stiff term g is simultaneously treated by three implicit Runge–Kutta methods. The three implicit methods for g are a diagonally implicit Runge–Kutta method and two Rosenbrock linearized Runge–Kutta methods [22] with different ways of evaluating Jacobian matrixes. The new methods, which are different from those used in Ref. [20], are derived to be both high-order accurate and strongly A-stable ($A(\pi/2)$ stable) for the implicit term g. The strongly A-stable methods are needed for numerical results to reach correct asymptotic values for very stiff problems. Numerical test results on these methods for the stability and accuracy properties are also presented in this paper.

2. ADDITIVE SEMI-IMPLICIT RUNGE–KUTTA METHODS

2.1. *General Formulas of Three ASIRK Methods*

In the semi-discretization approach, the spatial derivatives in the governing partial differential equations are first approximated by spatial discretization methods. The spatial discretization leads to a system of first-order ordinary differential equations,

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}) + \mathbf{g}(\mathbf{u}), \tag{1}$$

where **u** is the vector of discretized flow field variables. The right-hand side of the differential equation above is additively split into two terms, **g** and **f**, where **g** is the vector resulting from the spatial discretization of the stiff terms and **f** is the vector resulting from the spatial discretization of the rest of the nonstiff flow equations. In general, the splitting of **f** and **g** terms is not unique.

The Runge–Kutta methods are one-step methods involving intermediate stages to achieve high-order accuracy [9, 7]. A general r-stage additive semi-implicit Runge–Kutta method integrates Eq. (1) by simultaneously treating **f** explicitly and **g** implicitly:

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \sum_{j=1}^{r} \omega_j \mathbf{k}_j \tag{2}$$

$$\mathbf{k}_i = h\left\{\mathbf{f}\left(\mathbf{u}^n + \sum_{j=1}^{i-1} b_{ij}\mathbf{k}_j\right) + \mathbf{g}\left(\mathbf{u}^n + \sum_{j=1}^{i-1} c_{ij}\mathbf{k}_j + a_i\mathbf{k}_i\right)\right\} \quad (i = 1, ..., r), \tag{3}$$

where h is the time-step size, and a_i, b_{ij}, c_{ij}, ω_j are parameters to be determined by accuracy and stability requirements. Because **g** is treated by a diagonally implicit Runge–Kutta method, Eq. (3) is a nonlinear equation at every stage of the implicit calculations if **g** is a nonlinear function of **u**. The computations of this method are relatively inefficient, since nonlinear solvers are required to solve such nonlinear equations.

A more computationally efficient additive semi-implicit Runge–Kutta method is a semi-implicit extension of the Rosenbrock Runge–Kutta method [22],

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \sum_{j=1}^{r} \omega_j \mathbf{k}_j \tag{4}$$

$$\left[\mathbf{I} - ha_i\mathbf{J}\left(\mathbf{u}^n + \sum_{j=1}^{i-1} d_{ij}\mathbf{k}_j\right)\right]\mathbf{k}_i = h\left\{\mathbf{f}\left(\mathbf{u}^n + \sum_{j=1}^{i-1} b_{ij}\mathbf{k}_j\right) + \mathbf{g}\left(\mathbf{u}^n + \sum_{j=1}^{i-1} c_{ij}\mathbf{k}_j\right)\right\} \quad (i = 1, ..., r), \tag{5}$$

where $\mathbf{J} = \partial \mathbf{g}/\partial \mathbf{u}$ is the Jacobian matrix of the stiff term $\mathbf{g}$ and d_{ij} is an additional set of parameters. Most of the Rosenbrock methods similar to Eqs. (4) and (5) use a single $a_i = a$ with $d_{ij} = 0$ in order to use a single LU decomposition in solving Eq. (5) for all intermediate stages [23]. However, LU decomposition is often not possible for multidimensional reactive flow problems because of the enormous requirement for computer memory and CPU times in the LU decomposition method. Therefore, in this paper, the a_i's are allowed to be different at different stages in order to have more flexibility in searching for the optimal parameters in both stability and accuracy. Two methods are used to compute the Jacobians by using either $d_{ij} = 0$ or $d_{ij} = c_{ij}$ in this paper.

The Rosenbrock additive semi-implicit Runge–Kutta method given by Eqs. (4) and (5) is similar to the implicit methods used in computational fluid dynamics and is much more efficient than the diagonally implicit version given by Eqs. (2) and (3). But, for some strongly nonlinear problems, the nonlinear diagonally semi-implicit method given by Eqs. (2) and (3) is necessary because it is more stable than the Rosenbrock additive semi-implicit Runge–Kutta method for nonlinear problems. Therefore, three versions of the additive semi-implicit Runge–Kutta methods are derived to be both high-order accurate and strongly A-stable for the implicit terms, i.e.,

Method A. "Fully implicit" additive semi-implicit Runge–Kutta method given by Eqs. (2) and (3).

Method B. Rosenbrock additive semi-implicit Runge–Kutta method given by Eqs. (4) and (5), and $d_{ij} = 0$.

Method C. Rosenbrock additive semi-implicit Runge–Kutta method given by Eqs. (4) and (5), and $d_{ij} = c_{ij}$.

The rth-stage additive semi-implicit Runge–Kutta methods are termed ASIRK-rA methods, ASIRK-rB methods, and ASIRK-rC methods for Methods A, B, C, respectively.

2.2. *Linear Stability Conditions*

The parameters of the additive semi-implicit Runge–Kutta methods are chosen based on both stability and accuracy requirements with the simultaneous coupling between the explicit and implicit terms. The use of an implicit method for the stiff term $\mathbf{g}$ permits a larger time step than that allowed by a fully explicit method. Unlike the explicit Runge–Kutta methods, whose stability conditions are the same for different choices of parameters as long as they have the same stages and accuracy, the stability properties of the additive semi-implicit Runge–Kutta methods of the same stages are different for different choices of parameters because of the coupling between the $\mathbf{f}$ and $\mathbf{g}$ terms.

For simplicity, only a linear stability analysis is conducted in this paper for a special kind of test function. The stability condition for an additive semi-implicit time-stepping scheme is analyzed by considering a simplified linear model equation,

$$\frac{du}{dt} = \lambda_f u + \lambda_g u, \tag{6}$$

where λ_f and λ_g represent the eigenvalues of $\partial \mathbf{f}/\partial \mathbf{u}$ and $\partial \mathbf{g}/\partial \mathbf{u}$ in Eq. (1). They are complex parameters satisfying $\mathrm{Re}\{\lambda_f\} \leq 0$ and $\mathrm{Re}\{\lambda_g\} \leq 0$, respectively. In general, $|\lambda_g|$ is much larger than $|\lambda_f|$ for stiff equations. Although Eq. (1) cannot be reduced to this model equation if the Jacobians of $\mathbf{f}$ and $\mathbf{g}$ do not commute, Eq. (6) is used as the first step in analyzing the linear stability properties of the additive semi-implicit Runge–Kutta methods. Further studies are needed to analyze the general stability properties of the additive Runge–Kutta methods using the nonlinear stability analysis by Hairer, Bader, and Lubich [24].

Substituting Eq. (6) into any of the three additive semi-implicit Runge–Kutta methods leads to the same equation for the characteristic root as

$$\gamma = \frac{u^{n+1}}{u^n} = 1 + \sum_{j=1}^{r} \omega_j k_j \tag{7}$$

$$k_i = \frac{h\lambda_f(1 + \sum_{j=1}^{i-1} b_{ij}k_j) + h\lambda_g(1 + \sum_{j=1}^{i-1} c_{ij}k_j)}{1 - a_i h\lambda_g} \qquad (i = 1, ..., r), \tag{8}$$

where $\gamma = \gamma\{h\lambda_f, h\lambda_g\}$ is a function of $h\lambda_f$ and $h\lambda_g$.

An $A(\alpha)$ stability region of a semi-implicit method in the complex plane of $h\lambda_f$ is defined as the region where

$$|\gamma\{h\lambda_f, h\lambda_g\}| \leq 1 \tag{9}$$

for $h\lambda_f$ within the region and for all $h\lambda_g$ within a wedge bounded by $[\pi - \alpha, \pi + \alpha]$ in the complex plane. When $\alpha = \pi/2$, the semi-implicit method is A-stable for $h\lambda_g$. The $A(\alpha)$ stability region of the semi-implicit Runge–Kutta method is computed numerically in this paper.

In order to obtain a correct asymptotic decay for stiff terms, it is desirable to have a strong A-stability (L-stability) condition for the semi-implicit schemes; i.e.,

$$\lim_{h|\lambda_g| \to \infty} |\gamma\{h\lambda_f, h\lambda_g\}| = 0. \tag{10}$$

The strong A-stability for the implicit term assures that the numerical solutions approach the correct solutions as step sizes increase. For the three additive semi-implicit

Runge–Kutta methods, the strongly A-stable condition can be obtained from Eqs. (7), (8), and (10) as

$$1 + \sum_{j=1}^{r} \omega_j \beta_j = 0, \tag{11}$$

where

$$\beta_i = -\left[1 + \sum_{j=1}^{i-1} c_{ij}\beta_j\right] / a_i \quad (i = 1, \ldots, r).$$

For practical reactive flow problems, it is important that the intermediate variables at each stage of the Runge–Kutta computations maintain their physical meanings; i.e., it is not acceptable to have negative temperatures in an intermediate stage even if the final results are positive. Consider the diagonally implicit Runge–Kutta method for solving $\mathbf{u}' = \mathbf{g}(\mathbf{u})$,

$$\begin{aligned} \mathbf{u}^{n+1} &= \mathbf{u}^n + \omega_1 \mathbf{k}_1 + \omega_2 \mathbf{k}_2 \\ \mathbf{k}_1 &= h\mathbf{g}(\mathbf{u}^n + a_1 \mathbf{k}_1) \\ \mathbf{k}_2 &= h\mathbf{g}(\mathbf{u}^n + c_{21}\mathbf{k}_1 + a_2 \mathbf{k}_2), \end{aligned} \tag{12}$$

where $a_1 = 0$, and $a_2 = c_{21} = \omega_1 = \omega_2 = \frac{1}{2}$. This method is second-order accurate and A-stable. However, the method in computing k_1 in the first stage is an explicit method because a_1 is zero. Because the explicit calculations at that stage can lead to nonphysical results if the equations are stiff, this scheme is not appropriate for stiff reactive flow calculations. Therefore, we impose the following additional condition on the additive semi-implicit Runge–Kutta methods:

$$a_i > 0. \tag{13}$$

2.3. *Accuracy Conditions*

Additive semi-implicit Runge–Kutta schemes are derived to be high-order accurate with the simultaneous coupling between the explicit and implicit terms. Taylor series expansions lead to the following accuracy conditions:

First order,

$$\omega_1 + \omega_2 + \omega_3 = 1; \tag{14}$$

Second order,

$$\omega_2 b_{21} + \omega_3(b_{31} + b_{32}) = \tfrac{1}{2}, \tag{15}$$

$$\omega_1 a_1 + \omega_2(a_2 + c_{21}) + \omega_3(a_3 + c_{31} + c_{32}) = \tfrac{1}{2}; \tag{16}$$

Third order,

$$\omega_2 b_{21}^2 + \omega_3(b_{31} + b_{32})^2 = \tfrac{1}{3}, \tag{17}$$

$$\omega_3 b_{21} b_{32} = \tfrac{1}{6}, \tag{18}$$

$$\omega_2(b_{21}a_2 + b_{21}a_1) + \omega_3 (a_1 b_{31} + a_2 b_{32} + c_{21}b_{32} + b_{21}c_{32} + a_3 b_{31} + a_3 b_{32}) = \tfrac{1}{3}, \tag{19}$$

$$\omega_1 a_1^2 + \omega_2(a_2^2 + a_2 c_{21} + a_1 c_{21}) + \omega_3(a_1 c_{31} + a_2 c_{32} + c_{21}c_{32} + a_3 c_{31} + a_3 c_{32} + a_3^2) = \tfrac{1}{6}. \tag{20}$$

Method A: $\omega_1 a_1^2 + \omega_2(c_{21} + a_2)^2 + \omega_3(c_{31} + c_{32} + a_3)^2 = \frac{1}{3}$

Method B: $\omega_2 c_{21}^2 + \omega_3(c_{31} + c_{32})^2 = \frac{1}{3}$ (21)

Method C: $\omega_2(c_{21}^2 + 2a_2 c_{21}) + \omega_3\{(c_{31} + c_{32})^2 + 2a_3(c_{31} + c_{32})\} = \frac{1}{3}$.

The accuracy equations for Methods A, B, or C are the same, except in the third-order equation (21). In addition, the accuracy conditions for explicit coefficients b_{ij} are decoupled from implicit coefficients, c_{ij} and a_i, except in the third-order Eq. (19). Therefore, for up to second-order accuracy, a direct combination of explicit and implicit Runge–Kutta methods will result in an additive semi-implicit Runge–Kutta method with the same order of accuracy, as long as the two schemes have the same set of ω_i. However, for accuracy equal to or higher than third order, the direct combination of explicit and implicit methods will likely be only second-order accurate because of the coupling between the explicit and implicit terms in Eq. (19).

We search for the optimal parameters in the additive semi-implicit Runge–Kutta schemes by simultaneously imposing the stability and accuracy conditions discussed above. For example, for the third-order schemes, the parameters are searched for those simultaneously satisfying the following conditions:

1. Eight accuracy equations given by Eqs. (14)–(21),
2. One strong A-stability condition given by Eq. (11),
3. Large stability region defined by (9) with $a_i > 0$.

Whenever possible, we try to choose the coefficients for the explicit term $\mathbf{f}$ to be the same as those of the conventional explicit Runge–Kutta methods, so that when $\mathbf{g} = 0$, the schemes reduce to conventional explicit Runge–Kutta schemes. It turns out that we can only do that for the first- and second-order additive semi-implicit schemes. For the third-order additive semi-implicit schemes, we cannot find satisfactory parameters using the same explicit coefficients as those from the third-order TVD Runge–Kutta schemes of Shu and Osher [25] or other classical third-order methods [9]. Hence, the third-order methods in this paper are not the same as the conventional Runge–Kutta methods in both the explicit and the implicit parts.

2.4. *First-Order Additive Semi-Implicit Runge–Kutta Methods*

The expressions for the first-order methods are

ASIRK-1A method,

$$\begin{aligned}\mathbf{k}_1 &= h\{\mathbf{f}(\mathbf{u}^n) + \mathbf{g}(\mathbf{u}^n + a_1\mathbf{k}_1)\}\\ \mathbf{u}^{n+1} &= \mathbf{u}^n + \omega_1\mathbf{k}_1;\end{aligned} \tag{22}$$

ASIRK-1B and ASIRK-1C methods,

$$\begin{aligned}[\mathbf{I} - ha_1\mathbf{J}(\mathbf{u}^n)]\mathbf{k}_1 &= h\{\mathbf{f}(\mathbf{u}^n) + \mathbf{g}(\mathbf{u}^n)\}\\ \mathbf{u}^{n+1} &= \mathbf{u}^n + \omega_1\mathbf{k}_1.\end{aligned} \tag{23}$$

The accuracy condition given by Eq. (14) leads to

$$\omega_1 = 1 \tag{24}$$

and the characteristic root is

$$\gamma = \frac{1 + h\lambda_f + (1 - a_1)h\lambda_g}{1 - a_1 h\lambda_g}. \tag{25}$$

It can be shown that when $\frac{1}{2} \leq a_1 < 1$, the method is A-stable for λ_g, but it is not strongly stable. The only choice for strongly A-stable semi-implicit method is $a_1 = 1$. Therefore, the parameters chosen for first-order strongly A-stable semi-implicit Runge–Kutta schemes are

$$\omega_1 = 1, \quad a_1 = 1. \tag{26}$$

The coefficients are the same for ASIRK-1A, ASIRK-1B, and ASIRK-1C methods. The stability condition with the parameters given by Eq. (26) is

$$|1 + h\lambda_f| \leq 1, \quad \mathrm{Re}\{\lambda_g\} \leq 0. \tag{27}$$

In other words, the stability condition for the first-order additive semi-implicit Runge–Kutta methods is the same as for first-order explicit Runge–Kutta methods for $h\lambda_f$ and is strongly A-stable for $h\lambda_g$.

2.5. *Second-Order Additive Semi-Implicit Runge–Kutta Methods*

The expressions for the second-order methods are

ASIRK-2A method,

$$\begin{aligned}\mathbf{k}_1 &= h\{\mathbf{f}(\mathbf{u}^n) + \mathbf{g}(\mathbf{u}^n + a_1\mathbf{k}_1)\}\\ \mathbf{k}_2 &= h\{\mathbf{f}(\mathbf{u}^n + b_{21}\mathbf{k}_1) + \mathbf{g}(\mathbf{u}^n + c_{21}\mathbf{k}_1 + a_2\mathbf{k}_2)\}\\ \mathbf{u}^{n+1} &= \mathbf{u}^n + \omega_1\mathbf{k}_1 + \omega_2\mathbf{k}_2;\end{aligned} \tag{28}$$

ASIRK-2B method,

$$\begin{aligned}[\mathbf{I} - ha_1\mathbf{J}(\mathbf{u}^n)]\mathbf{k}_1 &= h\{\mathbf{f}(\mathbf{u}^n) + \mathbf{g}(\mathbf{u}^n)\}\\ [\mathbf{I} - ha_2\mathbf{J}(\mathbf{u}^n)]\mathbf{k}_2 &= h\{\mathbf{f}(\mathbf{u}^n + b_{21}\mathbf{k}_1) + \mathbf{g}(\mathbf{u}^n + c_{21}\mathbf{k}_1)\ \}\\ \mathbf{u}^{n+1} &= \mathbf{u}^n + \omega_1\mathbf{k}_1 + \omega_2\mathbf{k}_2;\end{aligned} \tag{29}$$

ASIRK-2C method,

$$\begin{aligned}[\mathbf{I} - ha_1\mathbf{J}(\mathbf{u}^n)]\mathbf{k}_1 &= h\{\mathbf{f}(\mathbf{u}^n) + \mathbf{g}(\mathbf{u}^n)\}\\ [\mathbf{I} - ha_2\mathbf{J}(\mathbf{u}^n + c_{21}\mathbf{k}_1)]\mathbf{k}_2 &\\ = h\{\mathbf{f}(\mathbf{u}^n + b_{21}\mathbf{k}_1) &+ \mathbf{g}(\mathbf{u}^n + c_{21}\mathbf{k}_1)\}\\ \mathbf{u}^{n+1} &= \mathbf{u}^n + \omega_1\mathbf{k}_1 + \omega_2\mathbf{k}_2.\end{aligned} \tag{30}$$

The coefficients for a conventional explicit second-order Runge–Kutta method are used for the explicit parameters, i.e., $\omega_1 = \frac{1}{2}$, $\omega_2 = \frac{1}{2}$, and $b_{21} = 1$. The equations for second-order accuracy and strong A-stability (Eqs. (16) and (11)) lead to one free parameter:

$$a_1 = \frac{\frac{1}{2} - a_2}{1 - a_2}, \quad c_{21} = 1 - a_2 - \frac{\frac{1}{2} - a_2}{1 - a_2}. \tag{31}$$

The requirement of a_1 and a_2 being positive numbers leads to $0 \leq a_2 \leq 0.5$.

A search for the parameters shows that the optimal values for the parameters in stability are $a_1 = a_2 = 1 - \sqrt{2}/2$, and $c_{21} = \sqrt{2} - 1$. But they involve irrational numbers. An alternative set of parameters for the second-order additive semi-implicit Runge–Kutta method are

$$\begin{aligned}&\omega_1 = \tfrac{1}{2}, \quad \omega_2 = \tfrac{1}{2}, \quad b_{21} = 1\\ &a_1 = \tfrac{1}{4}, \quad a_2 = \tfrac{1}{3}, \quad c_{21} = \tfrac{5}{12}.\end{aligned}$$

The coefficients are the same for ASIRK-2A, ASIRK-2B, and ASIRK-2C methods.

The methods are second-order accurate and strongly A-stable for the implicit term $h\lambda_g$. The $A(\alpha)$ stability regions in the complex plane of $h\lambda_f$ are given by Fig. 1. The stability boundaries in the figure are not very smooth because only limited numbers of points are searched in the intensive computations of the stability boundaries for λ_f with all possible λ_g. Only the upper half of the stability region is plotted in the figure. The figure shows that the $A(\alpha)$ stability region for the explicit term $h\lambda_f$ is the same as that for the explicit RK-2 methods when α is 0, but the region becomes slightly smaller as α approaches $\pi/2$.

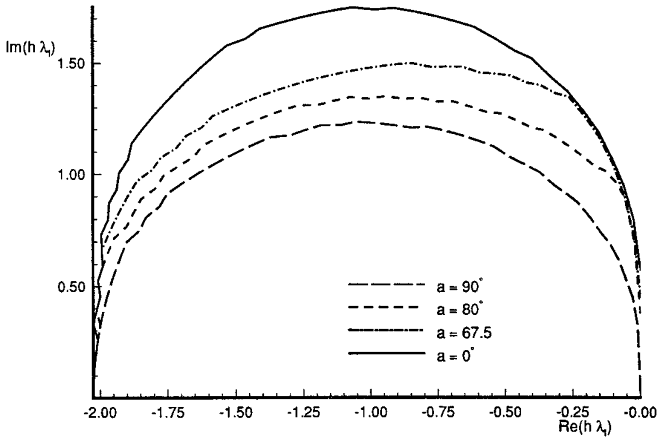

FIG. 1. The $A(a)$ stability region of the second-order semi-implicit Runge–Kutta methods for the explicit terms $h\lambda_f$.

2.6. *Third-Order Additive Semi-Implicit Runge–Kutta Methods*

The third-order ASIRK-3B method is

$$\begin{aligned}
&[\mathbf{I} - ha_1\mathbf{J}(\mathbf{u}^n)]\mathbf{k}_1 = h\{\mathbf{f}(\mathbf{u}^n) + \mathbf{g}(\mathbf{u}^n)\}\\
&[\mathbf{I} - ha_2\mathbf{J}(\mathbf{u}^n)]\mathbf{k}_2 = h\{\mathbf{f}(\mathbf{u}^n + b_{21}\mathbf{k}_1) + \mathbf{g}(\mathbf{u}^n + c_{21}\mathbf{k}_1)\}\\
&[\mathbf{I} - ha_3\mathbf{J}(\mathbf{u}^n)]\mathbf{k}_3 \qquad\qquad (32)\\
&\quad = h\{\mathbf{f}(\mathbf{u}^n + b_{31}\mathbf{k}_1 + b_{32}\mathbf{k}_2) + \mathbf{g}(\mathbf{u}^n + c_{31}\mathbf{k}_1 + c_{32}\mathbf{k}_2)\}\\
&\mathbf{u}^{n+1} = \mathbf{u}^n + \omega_1\mathbf{k}_1 + \omega_2\mathbf{k}_2 + \omega_3\mathbf{k}_3.
\end{aligned}$$

Similar expressions for ASIRK-3A and ASIRK-3C can be found in Section 2.1.

There are 12 undetermined parameters satisfying eight accuracy equations, Eqs. (14)–(21), and a strong A-stability condition, (Eq. 10). Therefore, there are three free parameters to be chosen to meet the stability requirements. During the search, we specify the values of ω_1 and ω_2, and then compute the values of all parameters numerically by solving Eqs. (14) to (21) using c_{31} and c_{32} as free parameters. For each set of c_{31} and c_{32}, the values of $Z_{\max}$ and $Z_{\max 1}$ are computed, where $Z_{\max}$ is the maximum magnitude of the characteristic root $|\gamma|$ for all possible $h\lambda_g$ in the left half-plane and all possible $h\lambda_f$ satisfying Eq. (27), and $Z_{\max 1}$ is the maximum characteristic roots $|\gamma|$ when $h|\{\lambda_g\}| \to \infty$. Optimal parameters with strong A-stability for implicit terms should lead to small $Z_{\max}$ and $Z_{\max 1} = 0$. Therefore, the contours of $Z_{\max}$ and $Z_{\max 1}$ are used to locate the approximate optimal values of parameters to meet the strong A-stability condition.

Figure 2 shows the contours of $Z_{\max}$ and $Z_{\max 1}$ respectively for the case of $\omega_1 = \omega_2 = \frac{1}{8}$ using c_{31} and c_{32} as independent variables for the ASIRK-3B method. The approximate values of the optimal parameters of c_{31} and c_{32} are those satisfying $Z_{\max} \leq 1$, $Z_{\max 1} = 0$, and $a_i > 0$ and can be located on the contours. From this figure, we choose the optimal value around $c_{31} = 0.5$ and $c_{32} \approx -0.69$. The exact value of c_{32} is then computed by solving the additional strong stability condition given by Eq. (11) by an iterative procedure. The approximate value of c_{32} is used as the initial guess in the iterative procedure. The parameters of all three methods obtained by the search are:

$$\begin{aligned}
\omega_1 &= \tfrac{1}{8}, & \omega_2 &= \tfrac{1}{8}\\
\omega_3 &= \tfrac{3}{4}, & b_{21} &= \tfrac{8}{7}\\
b_{31} &= \tfrac{71}{252}, & b_{32} &= \tfrac{7}{36}.
\end{aligned}$$

ASIRK-3A,

$$\begin{aligned}
a_1 &= .485561, & a_2 &= .951130\\
a_3 &= .189208, & c_{21} &= .306727\\
c_{31} &= .45, & c_{32} &= -.263111;
\end{aligned}$$

ASIRK-3B,

$$\begin{aligned}
a_1 &= 1.40316, & a_2 &= .322295\\
a_3 &= .315342, & c_{21} &= 1.56056\\
c_{31} &= \tfrac{1}{2}, & c_{32} &= -.696345;
\end{aligned}$$

ASIRK-3C,

$$\begin{aligned}
a_1 &= .797097, & a_2 &= .591381\\
a_3 &= .134705, & c_{21} &= 1.05893\\
c_{31} &= \tfrac{1}{2}, & c_{32} &= -.375939;
\end{aligned}$$

where a_1, a_2, a_3, c_{21}, and c_{32} are irrational numbers with six significant digits. The double-precision values of these parameters are listed in Table I.

The methods using the coefficients above are third-order accurate and strongly A-stable for the implicit term $h\lambda_g$. The $A(\alpha)$ stability region of the ASIRK-3C method is shown in Fig. 3. Similar to the stability region of the ASIRK-2C method, this figure shows that the $A(\alpha)$ stability region for the explicit term $h\lambda_f$ is the same as that for the explicit third-order RK methods when α is 0, but the region becomes slightly smaller as α approaches $\pi/2$.

3. STABILITY OF OTHER SEMI-IMPLICIT SCHEMES

Although the model equation (6) does not represent all stiff equations, it can be used as a test of stability of semi-

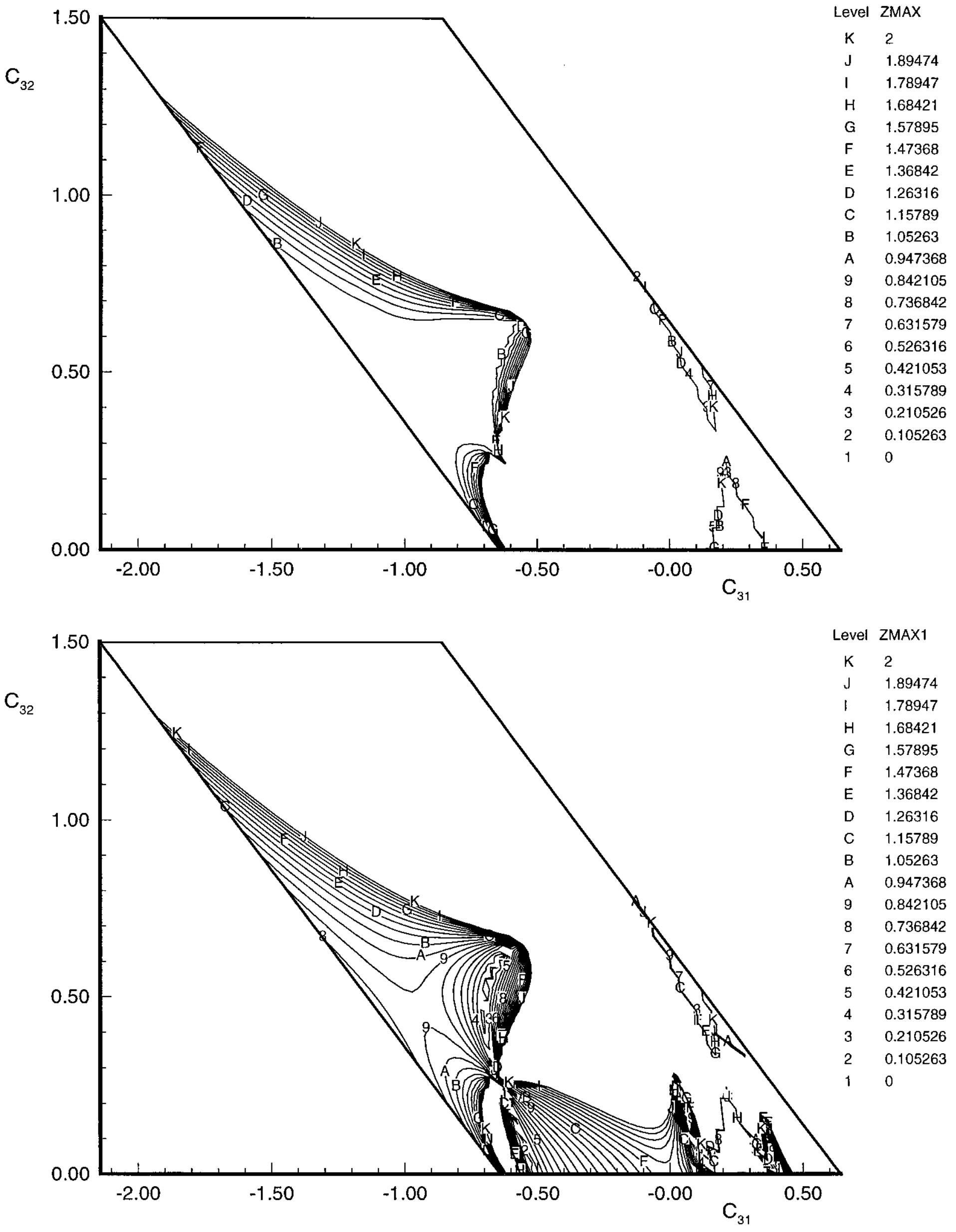

FIG. 2. The contours of the maximum magnitude of the characteristic root $Z_{\max}$ and the maximum characteristic root $Z_{\max 1}$ when $h|\lambda_g| \rightarrow \infty$ $(\omega_1 = \omega_2 = \frac{1}{8})$.

implicit methods. The stability properties of other commonly used semi-implicit methods are presented below by applying the methods to the model equation.

The semi-implicit MacCormack predictor–corrector method used for reactive flow [15–17, 6] is

$$\begin{aligned}[\mathbf{I} - \tfrac{1}{2}h\mathbf{J}(\mathbf{u}^n)]\mathbf{k}_1 &= h\{\mathbf{f}(\mathbf{u}^n) + \mathbf{g}(\mathbf{u}^n)\}\\ [\mathbf{I} - \tfrac{1}{2}h\mathbf{J}(\mathbf{u}^n)]\mathbf{k}_2 &= j\{\mathbf{f}(\mathbf{u}^n + \mathbf{k}_1) + \mathbf{g}(\mathbf{u}^n + c_{21}\mathbf{k}_1)\} \qquad (33)\\ \mathbf{u}^{n+1} &= \mathbf{u}^n + \tfrac{1}{2}(\mathbf{k}_1 + \mathbf{k}_2).\end{aligned}$$

This method is a two-stage semi-implicit Runge–Kutta method similar to the ASIRK-2 methods. LeVeque and Yee [6] showed that the traditional choice of $c_{21} = 1$ is only first-order accurate, and the coefficient for a second-order semi-implicit Runge–Kutta method is $c_{21} = 0$. This method should have a similar stability region as the ASIRK-2 method, but it is not strongly A-stable because

$$|\gamma| \rightarrow 1 \quad \text{as } h|\lambda_g| \rightarrow \infty. \qquad (34)$$

TABLE I

The Double-Precision Values of the Parameters for the SIRK-3 Methods

SIRK-3A	$a_1 = .4855612330925677$	$a_2 = .9511295466999914$
	$a_3 = .1892078709825326$	$c_{21} = .3067269871935408$
	$c_{31} = .45$	$c_{32} = -.2631108321468882$
SIRK-3B	$a_1 = 1.403160446775581$	$a_2 = .3222947153259484$
	$a_3 = .3153416455775987$	$c_{21} = 1.560563684998894$
	$c_{31} = \frac{1}{2}$	$c_{32} = -.6963447867610024$
SIRK-3C	$a_1 = .7970967740096232$	$a_2 = .5913813968007854$
	$a_3 = .1347052663841181$	$c_{21} = 1.058925354610082$
	$c_{31} = \frac{1}{2}$	$c_{32} = -.3759391872875334$

The ABCN method, which uses a combined Crank–Nicolson method and Adams–Bashford method, can be written as

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \frac{h}{2}[3\mathbf{f}_n - \mathbf{f}_{n-1}] + \frac{h}{2}[\mathbf{g}_n + \mathbf{g}_{n+1}]. \qquad (35)$$

The method is second-order accurate, and the $A(a)$ stability region is shown in Fig. 4. This figure shows that the stability region for the ABCN method becomes much smaller when α increases. Like the previous method, the ABCN method is not strongly A-stable for the implicit term. Figure 5 compares the *A(π/2)* stability regions of the semi-implicit Runge–Kutta methods with the ABCN method. The semi-implicit Runge–Kutta methods have much larger stability regions than the semi-implicit ABCN method because of the strong coupling between the explicit and implicit terms in the ASIRK methods.

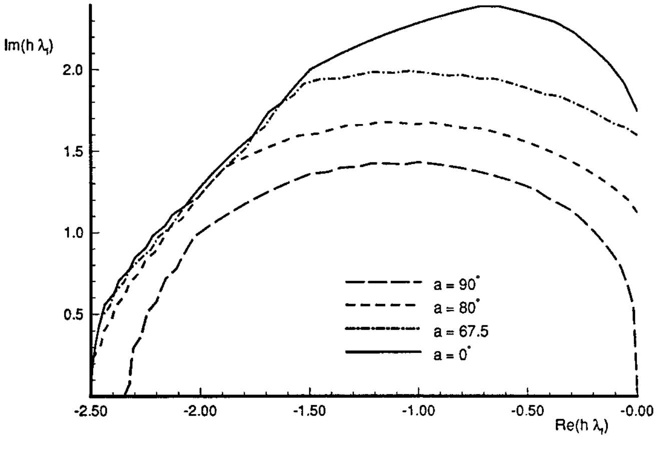

FIG. 3. The $A(a)$ stability region of the ASIRK-3C method for the explicit terms $h\lambda_f$.

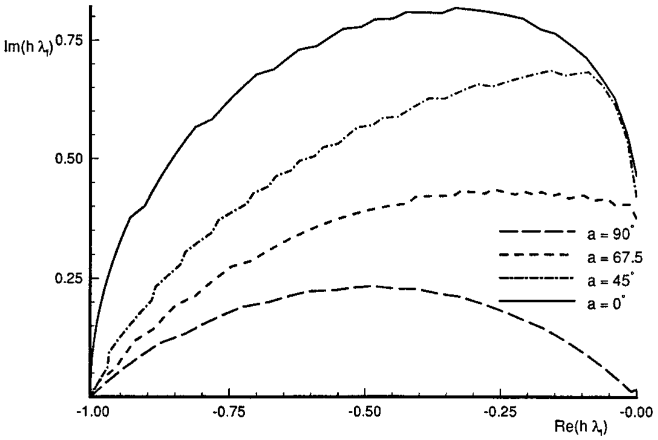

FIG. 4. The $A(a)$ stability region of the ABCN method for the explicit terms $h\lambda_f$.

4. TEST CASES

4.1. *Systems of Ordinary Differential Equations*

Lambert [26] pointed out that for any nonlinear system for which the solution suddenly increases in an isolated peak due to the eigenvalue straying into the right half-plane, there is a danger that a strongly A-stable method will lose solution information due to its excessive stability. This is due to the stability region of a strongly A-stable method encroaches into the positive regime. Lambert [26] considered the test case

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix}' = \begin{bmatrix} 42.2 & 50.1 & -42.1 \\ -66.1 & -58 & 58.1 \\ 26.1 & 42.1 & -34 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \qquad (36)$$

with initial conditions

$$\begin{aligned} u(\pi/8) &= \exp(-50\pi/8) \\ v(\pi/8) &= -\exp(.1\pi/8) - \exp(-50\pi/8) \\ w(\pi/8) &= -\exp(.1\pi/8) + \exp(-50\pi/8). \end{aligned} \qquad (37)$$

The eigenvalues are 0.1 ± 8i and −50. The exact solutions are

$$\begin{aligned} u(x) &= e^{0.1x}\sin 8x + e^{-50x} \\ v(x) &= e^{0.1x}\cos 8x + e^{-50x} \\ w(x) &= e^{0.1x}(\cos 8x + \sin 8x) + e^{-50x}. \end{aligned} \qquad (38)$$

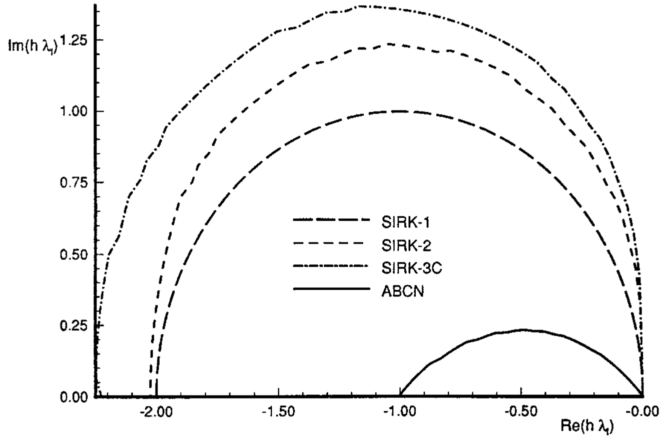

FIG. 5. Comparison of the $A(\pi/2)$ stability regions of semi-implicit methods.

Although this problem does not satisfy $\mathrm{Re}(\lambda) \leq 0$, it can be taken as a model for nonlinear systems for which the solutions suddenly increase for a restricted period of time.

Lambert solved Eq. (36) with $h = \pi/64$ by the A-stable trapezoidal method and by the strongly A-stable backward Euler method. The results showed that the backward Euler method damps out the solution quickly when it is supposed to increase. The cause of the damping in the backward Euler method is that its stability region is the whole complex plane, except a circle with radius 1 about the center 1. The complex number $h(0.1 \pm 8i)$ in this test case is not in this circle. Therefore, the results of the backward Euler method show damping for this particular step size. In order to obtain an acceptable solution using the backward Euler method, the step size has to be less than 0.00312 so that $h(0.1 \pm 8i)$ is within the region of instability. This indicates that the excessive damping is not a result of the strong A-stability of the scheme; it is related to the particular time step used in the computations.

We tested the semi-implicit Runge–Kutta schemes by using the ASIRK-1, ASIRK-2, and ASIRK-3C methods to solve Eq. (36) by dropping the explicit terms in the methods. The numerical solutions are shown in Fig. 6, where ASIRK-1 reduces to the backward Euler method. The results show that the backward Euler method damps the solution quickly as expected, but the strongly A-stable second-order implicit Runge–Kutta scheme performs slightly better than the trapezoidal method. The third-order implicit Runge–Kutta method (ASIRK-3C) performs better than the second-order methods. Therefore, the unsatisfactory performance of the backward Euler method is caused by the particular time step size used, instead of its strong A-stability.

4.2. *Model Convection-Diffusion Equation*

The new semi-implicit Runge–Kutta methods are tested by computing the linear decay of a two-dimensional model convection-diffusion equation bounded by two parallel walls,

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = \frac{1}{R}\frac{\partial^2 u}{\partial y^2}, \tag{39}$$

where R is the "Reynolds number." The boundary conditions are $u(x, 0) = u(x, 1) = 0$. This model problem is not a practical flow problem, but it is used to test the accuracy of the additive semi-implicit Runge–Kutta methods because it is simple enough to be use to evaluate the orders of the methods.

Similar to the stability analysis of the Navier–Stokes equations, we look for the temporal development of the solution in the form

$$u(x, y, t) = Y(y)e^{ikx}e^{-i\omega t}, \tag{40}$$

where k is a real number. The complex parameter ω and $Y(y)$ are an eigenvalue and an eigenfunction of the characteristic equation. Substituting Eq. (40) into Eq. (39) leads to the solution of ω and $Y(y)$,

$$Y_n(\mathrm{y}) = Ce^{Ry/2}\sin n\pi y, \quad \omega_n = -\alpha_n i + k, \tag{41}$$

where $\alpha_n = R/4[1 + (2n\pi/R)^2]$ and $n = 1, 2, ...$ The corresponding solution is

$$u_n(x, y, t) = Ce^{Ry/2}\sin n\pi y e^{ik(x-t)}e^{-\alpha_n t}. \tag{42}$$

The solution represents an exponential decay of the oscillation energy. Therefore, if we use $u_n(x, y, 0)$ given by Eq. (42) as an initial condition, the exact solution of the model equation is also given by the same equation.

When R is large, there is a thin viscous boundary layer on the upper wall with large gradients in y direction. Nu-

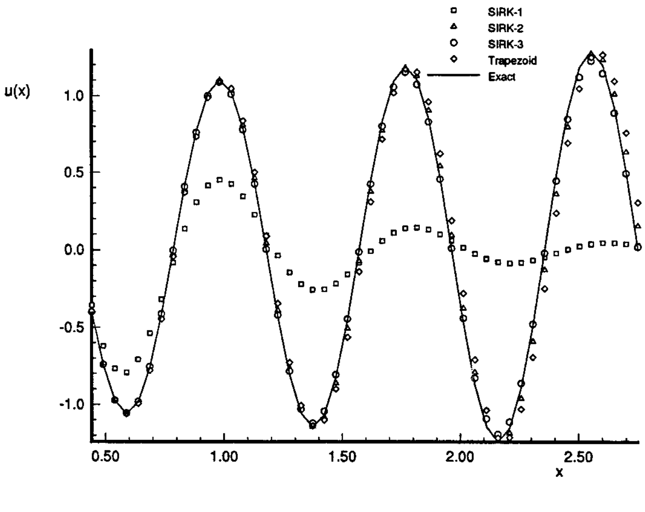

FIG. 6. Solution of the system of equations used by Lambert (1980).

merical solutions of this equation will be stiff for large R because fine grids are needed in y direction to resolve the boundary layer. In order to overcome the stiffness in the equation, we use the semi-implicit Runge–Kutta methods developed in this paper to treat u_x in Eq. (39) explicitly and to treat u_y and u_{yy} implicitly. Although stretched grids are often used in practice, simple uniform grids are used here in order to evaluate the accuracies of the schemes by grid refinement studies.

The finite-difference discretization of the spatial derivatives leads to a system of semi-discrete ordinary different equations

$$\frac{\partial u_{ij}}{\partial t} = f(u_{ij}) + g(u_{ij}), \tag{43}$$

where

$$f(u_{ij}) = \left\{ -\frac{\partial u}{\partial x} \right\}_{ij} \tag{44}$$

$$g(u_{ij}) = \left\{ -\frac{\partial u}{\partial y} + \frac{1}{R}\frac{\partial^2 u}{\partial y^2} \right\}_{ij}, \tag{45}$$

where explicit third-order upwind approximation is used for u_x and fourth-order central difference approximation is used for u_y and u_{yy} terms, i.e.,

$$f(u_{ij}) = -\frac{11u_{ij} - 18u_{i-1j} + 9u_{i-2j} - 2u_{i-3j}}{6\Delta x} \tag{46}$$

$$g(u_{ij}) = -\frac{-u_{ij+2} + 8u_{ij+1} - 8u_{ij-1} + u_{ij-2}}{12\Delta y} + \frac{1}{R}\frac{-u_{ij+2} + 16u_{ij+1} - 30u_{ij} + 16u_{ij-1} - u_{ij-2}}{\Delta y^2}. \tag{47}$$

A periodic boundary condition is used in the x direction. Either a three-point extrapolation or an antisymmetric boundary condition is used at the walls ($j = 1$ and $j = JL$) to calculate u located at one grid-point outside of the walls. The boundary conditions at the lower wall are

$$u_{i1} = 0 \tag{48}$$

and

$$u_{i0} = \begin{cases} -3u_{i2} + u_{i3}, & \text{3-point extrapolation,} \\ -u_{i2}, & \text{antisymmetric.} \end{cases} \tag{49}$$

Similar conditions are applied to the upper wall. Both methods of setting the boundary conditions at the walls are used in the computations to compare the results. The numerical results show that the antisymmetric boundary conditions for u at the wall reduce the accuracy of the third-order results because it is only second-order accurate.

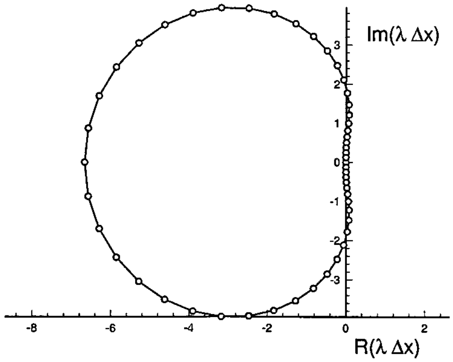

FIG. 7. The eigenvalues of the f term in Eq. (46).

The eigenvalues for the f terms in Eq. (46) with periodic boundary conditions are located on the curve shown in Fig. 7. The range of the real parts of the eigenvalues is

$$-\frac{6.67}{\Delta x} \leq \mathrm{Re}(\lambda_f) \leq \frac{0.082}{\Delta x}. \tag{50}$$

The very small positive values should not cause stability problems because of the strong effects of the g terms on the combined systems. On the other hand, the eigenvalues of the g terms, neglecting the first-order derivative and the

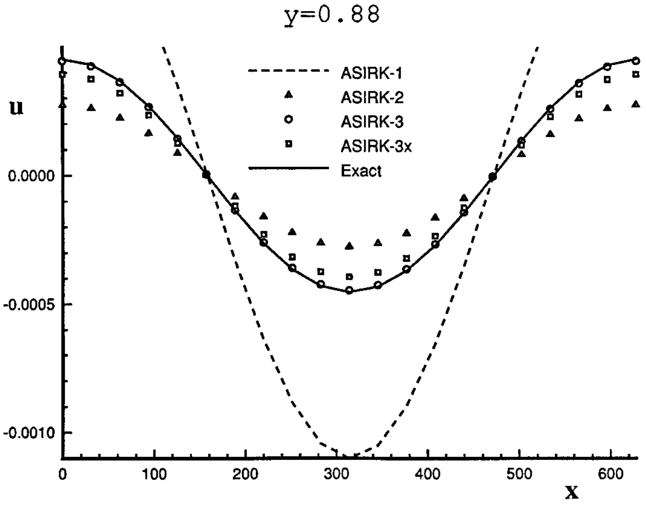

FIG. 8. The distribution of transient solution in the x direction at $y = 0.88$, $t = 1.054237$.

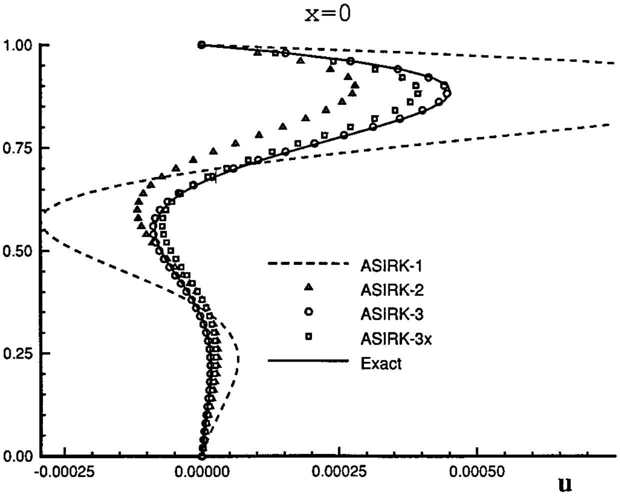

FIG. 9. The distribution of transient solution in y direction at $x = 0$, $t = 1.054237$.

effects of boundary conditions, are located along the real axis in the range of

$$-\frac{64}{R\Delta y^2} \leq \operatorname{Re}(\lambda_g) \leq 0. \tag{51}$$

For the current case of $R = 10$, $\Delta y = \frac{1}{20}$, and $\Delta x = 4\pi$, it can be shown that

$$\frac{\max|\lambda_g|}{\max|\lambda_f|} \approx 2400. \tag{52}$$

Therefore, the g terms are much stiffer than the f terms for this case. The additive semi-implicit Runge–Kutta methods can be used to treat the stiff g term implicitly and the nonstiff f term explicitly.

The semi-discretized Eq. (43) is solved using a first-order ASIRK-1 method, a second-order ASIRK-2 method, and two third-order ASIRK-3 and ASIRK-3x methods. Both ASIRK-3 and ASIRK-3x use the same third-order semi-implicit Runge–Kutta method (ASIRK-3C) in time but with different boundary conditions at the walls: ASIRK-3 uses 3-point extrapolation method while ASIRK-3x uses an antisymmetric condition. The conditions for calculations are: $R = 10$, $k = 0.01$, $C = 1$, and $n = 3$. The initial condition is given by Eq. (42). The computations domain, bounded by $(0, 2\pi/k) \times (0, 1)$, is discretized by a set of 51×21 grids.

Figure 8 shows the solution distribution along the x direction at $y = 0.88$ and at $t = 1.054237$ using $h = t/60$. Figure 9 shows the distribution along the y direction at $x = 377$ at the same moment. Figure 10 shows the contours of the solutions. These figures show that the first-order ASIRK-1 method produces large errors in the solution. While both the ASIRK-2 and ASIRK-3 methods are quite accurate, the ASIRK-3 method is more accurate than the ASIRK-2 method. Furthermore, as the order of accuracy increases, the numerical results of the semi-implicit Runge–Kutta methods become more accurate. The figures also show that the use of antisymmetric condition at the wall in the ASIRK-3x method leads to less accurate results.

In order to test the temporal accuracy of the semi-implicit Runge–Kutta methods derived in this paper, a grid refinement study is conducted by successively repeating the computations with doubled time step numbers. The same spatial grids are used for all test cases in order to exclude the spatial discretization errors from the temporal errors. The temporal order of accuracy is numerically determined by computing the parameter R_p defined by

$$e_h = u_{ex} - u_h \tag{53}$$

$$R_p = \frac{e_h}{e_{h/2}}, \tag{54}$$

where e_h is the numerical errors due to the temporal discretization only, u_h is the numerical solution computed using time step h, u_{ex} is the numerical exact solution computed using the Richardson extrapolated solution at the smallest time step, and p is the computed order of the method. For a pth order method, the expected value of R_p is

$$R_p = 2^p. \tag{55}$$

Table II shows the results of the grid refinement study. From the table, we can see that the ASIRK-3C method is third-order accurate, while ASIRK-2 method is second-order accurate.

5. CONCLUSIONS

Three additive semi-implicit Runge–Kutta methods up to third-order accurate have been derived in this paper for direct numerical simulation of nonequilibrium hypersonic flows. These high-order accurate additive semi-implicit schemes are strongly A-stable for the implicit terms when the explicit terms are in stability regions similar to those of pure explicit Runge–Kutta methods. Linear stability analysis shows that these methods have good stability properties for the explicit terms. The new methods have been tested by computing a test case used by Lambert and by a two-dimensional model boundary layer stability problem. The test results show that these schemes are stable and accurate for the calculations. The temporal orders of accuracy of the additive semi-implicit Runge–Kutta methods

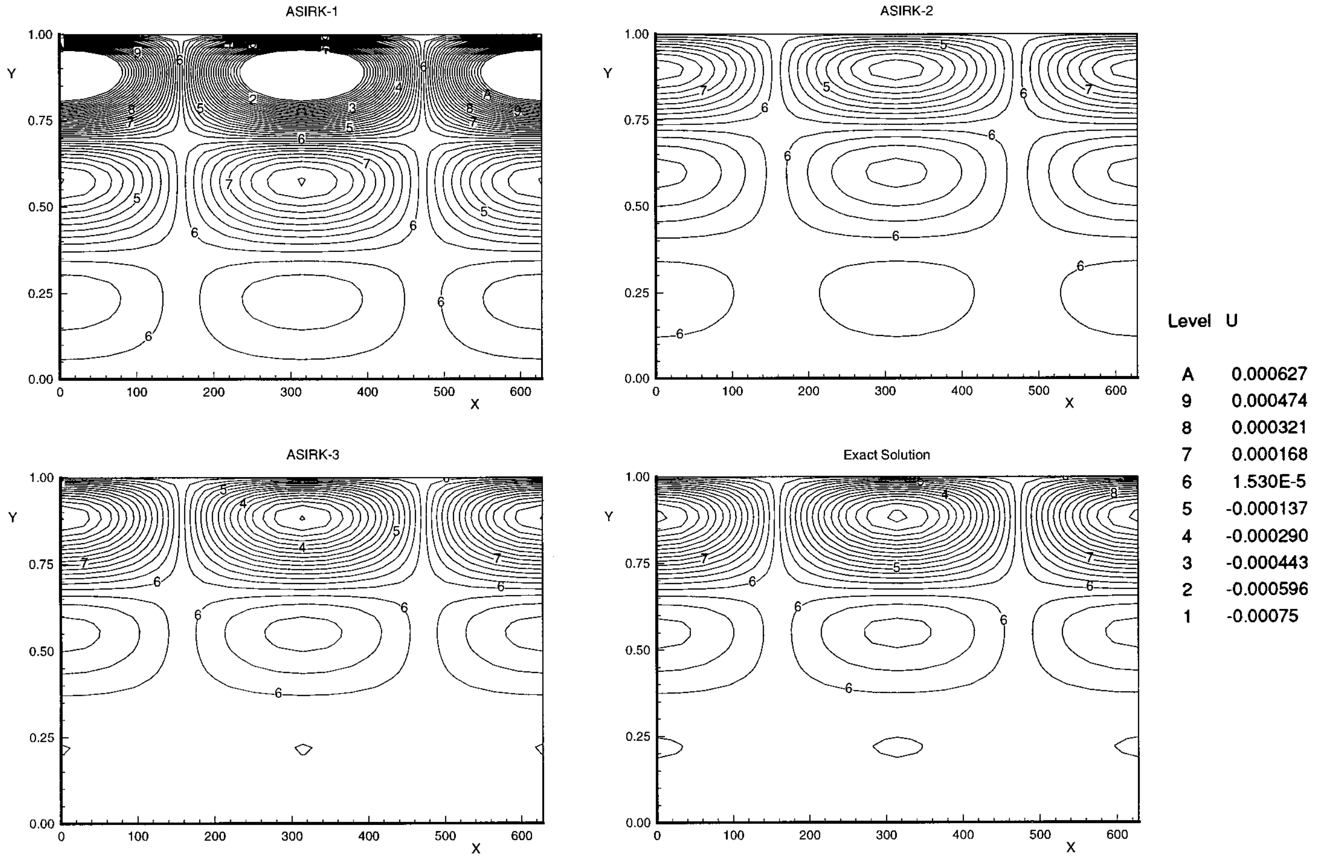

FIG. 10. The contours of instantaneous solution at $t = 1.054237$.

TABLE II

Temporary Accuracy of the ASIRK-2 and ASIRK-3C Methods

Δt	e_3	R_3	e_2	R_2
$h = 0.0439265254816$	5.47D-2	6.7	1.26D-1	3.9
$h/2$	8.22D-3	7.2	3.20D-2	4.0
$h/4$	1.14D-3	7.6	7.98D-3	4.0
$h/8$	1.50D-4	7.8	1.99D-3	4.0
$h/16$	1.93D-5	7.9	4.96D-4	4.0
$h/32$	2.44D-6	8.0	1.24D-4	4.0
$h/64$	3.05D-7	—	3.09D-5	—

Note. $x = 0$, $y = 0.84$, $t = 1.054237$, and $u_{\text{ex}} = -.403729064318\text{D-}03$.

have been numerically validated by a grid refinement study in a test case of this paper.

ACKNOWLEDGMENTS

This research was supported by AFOSR Grant F49620-94-1-0019, with Dr. L. Sakell as the grant monitor. The author gratefully acknowledges valuable discussions with Dr. E. S. Oran of the Naval Research Lab.

REFERENCES

1. C.-H. Chiu and X. Zhong, *AIAA Paper 95-0469,* 1995, *AIAA J,* Vol. 34, No. 4, 1996, pp. 655–661.
2. X. Zhong, X. Joubert, and T. K. Lee, in *20th International Symp. on Shock Waves, July 23–28, 1995, Pasadena, CA* (unpublished).
3. N. N. Yanenko, *The Method of Fractional Steps* (Springer-Verlag, New York/Berlin, 1971).
4. E. S. Oran and J. P. Boris, *Numerical Simulation of Reactive Flow* (Elsevier Science, New York, 1987).
5. G. Strang, *SIAM J. Numer. Anal.* **5**, 506 (1968).
6. R. J. Leveque and H. C. Yee, *J. Comput. Phys.* **86,** 187 (1990).
7. J. C. Butcher, *Math. Comput.* **18,** 50 (1964).
8. R. May and J. Noye, "The Numerical Solution of Ordinary Differential Equations: Initial Value Problems," in *Computational Techniques for Differential Equations,* edited by J. Noye (North-Holland, Amsterdam, 1983), p. 1.
9. J. D. Lambert, *Computational Methods in Ordinary Differential Equations* (Wiley, New York, 1973).
10. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics* (Springer-Verlag, New York/Berlin, 1988).
11. P. Moin and J. Kim, *J. Comput. Phys.* **35,** 381 (1980).
12. T. A. Zang and M. Y. Hussaini, *AIAA Paper 81-1227,* 1981 (unpublished).

13. R. S. Rogallo and P. Moin, *Annu. Rev. Fluid Mech.* **16,** (1984).
14. L. Kleiser and T. A. Zang, *Annu. Rev. Fluid Mech.* **23,** 495 (1991).
15. T. R. A. Bussing and E. M. Murman, *AIAA Paper 85-0296* 1985 (unpublished).
16. J. P. Drummond, R. C. Rogers, and M. Y. Hussaini, *AIAA Paper 85-0302* 1985 (unpublished).
17. H. C. Yee and J. L. Shinn, *AIAA Paper 87-1116*, 1987 (unpublished).
18. G. J. Cooper and A. Sayfy, *Math. Comput.* **35**(152), 1159 (1980).
19. G. J. Cooper and A. Sayfy, *Math. Comput.* **40**(161), (1983).
20. B. Enquist and B. Sjogreen, Report 91-03, CAM, March 1991, Department of Mathematics, University of California, Los Angeles (unpublished).
21. E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II* (Springer-Verlag, New York/Berlin, 1987).
22. H. H. Rosenbrock, *Comput. J.* **5,** 329 (1963).
23. P. Kaps and P. Rentrop, *Numer. Math.* **33,** 55 (1979).
24. E. Hairer, G. Bader, and CH. Lubich, *BIT* **22,** 211 (1982).
25. C. Shu and S. Osher, Contractor Report 181656, ICASE Report No. 88-24, NASA, April 1988 (unpublished).
26. J. D. Lambert, Stiffness, in *Computational Techniques for Ordinary Differential Equations,* edited by I. Gladwell and D. K. Sayers (Academic Press, San Diego, 1980), p. 19.