AIAA Meeting Papers on Disc, June 1996
A9636449, F49620-94-1-0019, F49620-95-1-0405, AIAA Paper 96-1969

Semi-Implicit Runge-Kutta Schemes for non-autonomous differential equations in reactive flow computations

Jian W. Shen
California Univ., Los Angeles

Xiaolin Zhong
California Univ., Los Angeles

AIAA, Fluid Dynamics Conference, 27th, New Orleans, LA, June 17-20, 1996

This paper is concerned with time-stepping numerical methods for computing stiff semi-discrete systems of ordinary differential equations for transient hypersonic flows with thermochemical 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 thermochemical 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. For additively split autonomous differential equations in the form of u' = f(u) + g(u), three different semi-implicit Runge-Kutta methods have been derived and tested in previous papers, where f is treated by explicit Runge-Kutta methods and g is simultaneously treated by three implicit Runge-Kutta methods. The coefficients of up to third-order accuracy have been derived such that the methods are both high-order accurate and strongly A-stable for the implicit terms. However, these semi-implicit Runge-Kutta methods for the autonomous systems cannot be extended to non-autonomous systems of u' = f(t, u) + g(t, u) because of the coupling between the f and g terms in the split Runge-Kutta methods. In this paper, we derive and test three different semi-implicit Runge-Kutta schemes of up to third-order accuracy for the non-autonomous differential equations using the A-stability and accuracy conditions with four stages. The new schemes have been tested in computations of unsteady reactive flows with explicit time-dependent terms. (Author)

Semi-Implicit Runge-Kutta Schemes for Non-Autonomous Differential Equations in Reactive Flow Computations

Jian Wei Shen *and Xiaolin Zhong †
University of California, Los Angeles, California 90095

Abstract

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. For additively split autonomous differential equations in the form of $u' = f(u) + g(u)$, three different semi-implicit Runge-Kutta methods have been derived and tested in previous papers, where f is treated by explicit Runge-Kutta methods and g is simultaneously treated by three implicit Runge-Kutta methods. The coefficients of up to third-order accuracy have been derived such that the methods are both high-order accurate and strongly A-stable for the implicit terms. However, these semi-implicit Runge-Kutta methods for the autonomous systems cannot be extended to non-autonomous systems of $u' = f(t,u) + g(t,u)$ because of the coupling between the f and g terms in the split Runge-Kutta methods. In this paper, we derive and test three different semi-implicit Runge-Kutta schemes of up to third-order accuracy for the non-autonomous differential equations using the A-stability and accuracy conditions with four stages. The new schemes have been tested in computations of unsteady reactive flows with explicit time-dependent terms.

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 research in the direct numerical simulation of 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 multi-dimensional 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 multi-dimensional reactive flow calculations include the fractional step method (or time-splitting method) and the additive semi-implicit method.

This paper is concerned with the additive semi-implicit methods, which additively split the ordinary differential equations into stiff and nonstiff terms. 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

*Visiting Scholar, Member AIAA
†Assistant Professor, Mechanical and Aerospace Engineering Department, Member AIAA

for reactive flow computations because the stiff 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[3, 4, 5, 6, 7] (ABCN method). For compressible reactive flow, a semi-implicit MacCormack method[8, 9, 10, 11] 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 second-order accurate at most. To obtain simultaneously high-order accuracy and good stability properties, the simultaneous coupling between the explicit and implicit terms need to be considered. This can be accomplished by the additive semi-implicit Runge-Kutta methods. The derivation of an additive semi-implicit method with both high accuracy and good stability is difficult 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[12, 13]. 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.

In previous papers, Zhong[14, 15] derived and analyzed three different sets of additive semi-implicit Runge-Kutta methods 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[16] with different ways of evaluating Jacobian matrices. The semi-implicit Runge-Kutta methods are derived and analyzed based on the general form of implicit Runge-Kutta formulas. The fully implicit and Rosenbrock semi-implicit Runge-Kutta methods are both high-order and strongly A-stable for the implicit terms. The strongly A-stable methods are needed for numerical results to reach correct asymptotic values for very stiff problems. These new schemes have been tested in model equations and in reactive hypersonic flow computations[15].

These additive semi-implicit Runge-Kutta methods, however, cannot be applied to non-autonomous systems relating to unsteady reactive flows with explicit time terms in the equation of $u' = f(t,u) + g(t,u)$ due to the coupling between the f and g terms. An example of such flow is the unsteady reactive flow with time-dependent perturbations or boundary conditions. The objective of this paper is to extend the previous work of semi-implicit Runge-Kutta methods for the autonomous systems to the non-autonomous systems, where the nonstiff $f(t,u)$ term is treated by explicit Runge-Kutta methods, and the stiff term $g(t,u)$ is simultaneously treated by three implicit Runge-Kutta methods. The three implicit methods for $g(t,u)$ are a diagonally implicit Runge-Kutta method and two Rosenbrock linearized Runge-Kutta methods[16] with different ways of evaluating Jacobian matrices. The coefficients are derived such that the new methods are simultaneously high-order accurate and A-stable for the implicit term. We found that the semi-implicit Runge Kutta method for non-autonomous differential equations requires four stages to obtain third-order accuracy due to the strong coupling among the coefficients of 18 accuracy and stability conditions. The new schemes are tested in the computations of unsteady reactive flows with explicit time-dependent terms.

Semi-Implicit RK Methods

General Formulas of Semi-Implicit RK Methods for Non-Autonomous Systems

A general partial differential equation for reactive flows can be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_j}{\partial x_j} = \mathbf{W} \tag{1}$$

where $\mathbf{U}$ is the conservation variables, $\mathbf{F}_j$ is the flux terms, and $\mathbf{W}$ is the source term due to finite-rate reactions. 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. For systems of unsteady flow with time-dependent forcing terms or boundary conditions, the semi-implicit differential equations are non-autonomous, i.e.,

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(t,\mathbf{u}) + \mathbf{g}(t,\mathbf{u}) \tag{2}$$

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

The multiple time scales in the flow equations and the difficulty in computing fully implicit multi-dimensional flow equations for reactive flow calculations require that the stiff term g to be computed implicitly and the non-stiff term f to be computed explicitly. Because of the coupling between f and g, however, a straight-forward combination of an explicit and an implicit time-stepping method will not maintain the original accuracy of the individual methods. Therefore, the derivation of high-order semi-implicit methods needs to simultaneously consider the coupling between the implicit and explicit terms in both stability and accuracy analysis. The stability conditions in the semi-implicit method should be limited by the explicit terms only, i.e., the coupled method should be A-stable for the implicit terms when the explicit terms satisfies the CFL condition (or similar conditions). Therefore, we use semi-implicit Runge-Kutta methods to achieve simultaneous high-order accuracy and good stability.

The Runge-Kutta methods are one-step methods involving intermediate stages to achieve high-order accuracy[17, 18]. A general r-stage additive semi-implicit Runge-Kutta method integrates Eq. (2) by simultaneously treating $\mathbf{f}$ explicitly and $\mathbf{g}$ implicitly:

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \sum_{j=1}^{r} \omega_j \mathbf{k}_j \tag{3}$$

$$\begin{aligned} \mathbf{k}_i &= h\{\mathbf{f}(t_n + r_i h, \mathbf{u}^n + \sum_{j=1}^{i-1} b_{ij}\mathbf{k}_j) \\ &+ \mathbf{g}(t_n + s_i h, \mathbf{u}^n + \sum_{j=1}^{i-1} c_{ij}\mathbf{k}_j + a_i \mathbf{k}_i)\} \\ &(i = 1, \cdots, r) \end{aligned} \tag{4}$$

where h is the time-step size, and a_i, b_{ij}, c_{ij}, r_i, s_i, and w_j are parameters to be determined by accuracy and stability requirements. Because $\mathbf{g}$ is treated by a diagonally implicit Runge-Kutta method, Eq. (4) is a nonlinear equation at every stage of the implicit calculations if $\mathbf{g}$ is a nonlinear function of $\mathbf{u}$. 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[16],

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \sum_{j=1}^{r} \omega_j \mathbf{k}_j \tag{5}$$

$$\begin{aligned} &\left[\mathbf{I} - h a_i \mathbf{J}(t_n + q_i h, \mathbf{u}^n + \sum_{j=1}^{i-1} d_{ij}\mathbf{k}_j)\right] \mathbf{k}_i \\ &= h\mathbf{f}(t_n + r_i h, \mathbf{u}^n + \sum_{j=1}^{i-1} b_{ij}\mathbf{k}_j) \\ &+ h\mathbf{g}(t_n + s_i h, \mathbf{u}^n + \sum_{j=1}^{i-1} c_{ij}\mathbf{k}_j) \\ &(i = 1, \cdots, r) \end{aligned} \tag{6}$$

Where $\mathbf{J} = \frac{\partial \mathbf{g}}{\partial \mathbf{u}}$ is the Jacobian matrix of the stiff term $\mathbf{g}$. The Rosenbrock additive semi-implicit Runge-Kutta method given by Eqs. (5) and (6) is similar to the implicit methods used in computational fluid dynamics and is much more efficient than the diagonally implicit version given by Eqs. (3) and (4). But, for some strongly nonlinear problems, the nonlinear diagonally semi-implicit method given by Eqs. (3) and (4) 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. (3) and (4).

Method B: Rosenbrock additive semi-implicit Runge-Kutta method given by Eqs. (5) and (6), and $d_{ij} = q_i = 0$.

Method C: Rosenbrock additive semi-implicit Runge-Kutta method given by Eqs. (5) and (6), and $q_i = s_i$, 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, and C respectively.

Linear Stability Conditions

The parameters of the additive semi-implicit Runge-Kutta methods are chosen based on both stability and accuracy requirements with 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 **f** and **g** terms.

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

$$\frac{du}{dt} = \lambda_f u + \lambda_g u \tag{7}$$

where λ_f and λ_g represent the eigenvalues of $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$ and $\frac{\partial \mathbf{g}}{\partial \mathbf{u}}$ in Eq. (2). They are complex parameters satisfying $Re\{\lambda_f\} \leq 0$ and $Re\{\lambda_g\} \leq 0$ respectively. In general, $|\lambda_g|$ is much larger than $|\lambda_f|$ for stiff equations. Though Eq. (2) cannot be reduced to this model equation if the Jacobians of **f** and **g** do not commute, Eq. (7) 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[19].

Substituting Eq. (7) into any of the three additive semi-implicit Runge-Kutta methods leads to the same equation for the characteristic root as follows:

$$\gamma = \frac{u^{n+1}}{u^n} = 1 + \sum_{j=1}^{r} \omega_j k_j \tag{8}$$

$$k_i = \frac{h\lambda_f(1 + \sum_{j=1}^{i-1} b_{ij} k_j) + h\lambda_g(1 + \sum_{j=1}^{i-1} c_{ij} k_j)}{1 - a_i h\lambda_g} \quad (i = 1, \cdots, r) \tag{9}$$

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

$$|\gamma\{h\lambda_f, h\lambda_g\}| \leq 1 \tag{10}$$

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 $h\lambda_g$.

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.,

$$\lim_{h|\lambda_g| \to \infty} |\gamma\{h\lambda_f, h\lambda_g\}| = 0 \tag{11}$$

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. (8), (9), and (11) as follows:

$$1 + \sum_{j=1}^{r} \omega_j \beta_j = 0 \tag{12}$$

where

$$\beta_i = -[1 + \sum_{j=1}^{i-1} c_{ij} \beta_j]/a_i \qquad (i = 1, \cdots, r) \tag{13}$$

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. Therefore, we impose the following additional condition on the additive semi-implicit Runge-Kutta methods:

$$a_i > 0 \tag{14}$$

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 if four steps are kept as a general case:

1st order:

$$\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1 \tag{15}$$

2nd order:

$$\omega_2 r_2 + \omega_3 r_3 + \omega_4 r_4 = \tfrac{1}{2} \tag{16}$$

$$\omega_1 s_1 + \omega_2 s_2 + \omega_3 s_3 + \omega_4 s_4 = \tfrac{1}{2} \quad (17)$$

$$\omega_2 b_{21} + \omega_3(b_{31} + b_{32}) + \omega_4(b_{41} + b_{42} + b_{43}) = \tfrac{1}{2} \quad (18)$$

$$\omega_1 a_1 + \omega_2(a_2 + c_{21}) + \omega_3(a_3 + c_{31} + c_{32}) + \omega_4(a_4 + c_{41} + c_{42} + c_{43}) = \tfrac{1}{2} \quad (19)$$

3rd order:

$$\omega_2 r_2^2 + \omega_3 r_3^2 + \omega_4 r_4^2 = \tfrac{1}{3} \quad (20)$$

$$\omega_1 s_1^2 + \omega_2 s_2^2 + \omega_3 s_3^2 + \omega_4 s_4^2 = \tfrac{1}{3} \quad (21)$$

$$\omega_1 a_1 s_1 + \omega_2(c_{21} s_1 + a_2 s_2) + \omega_3(c_{31} s_1 + c_{32} s_2 + a_3 s_3) + \omega_4(c_{41} s_1 + c_{42} s_2 + c_{43} s_3 + a_3 s_4) = \tfrac{1}{6} \quad (22)$$

$$\omega_1 a_1^2 + \omega_2[c_{21} a_1 + a_2(c_{21} + a_2)] + \omega_3[a_1 c_{31} + c_{32}(a_2 + c_{21}) + a_3(c_{31} + c_{32} + a_3)] + \omega_4[c_{41} a_1 + c_{42}(a_2 + c_{21}) + c_{43}(c_{31} + c_{32} + a_3) + a_4(c_{41} + c_{42} + c_{43} + a_4)] = \tfrac{1}{6} \quad (23)$$

$$\omega_3 b_{32} r_2 + \omega_4(b_{42} r_2 + b_{43} r_3) = \tfrac{1}{6} \quad (24)$$

$$\omega_2 b_{21}^2 + \omega_3(b_{31} + b_{32})^2 + \omega_4(b_{41} + b_{42} + b_{43})^2 = \tfrac{1}{3} \quad (25)$$

$$\omega_2 r_2 b_{21} + \omega_3 r_3(b_{31} + b_{32}) + \omega_4 r_4(b_{41} + b_{42} + b_{43}) = \tfrac{1}{3} \quad (26)$$

$$\omega_3 b_{32} b_{21} + \omega_4[b_{42} b_{21} + b_{43}(b_{31} + b_{32})] = \tfrac{1}{6} \quad (27)$$

$$\omega_2(b_{21} a_2 + b_{21} a_1) + \omega_3(a_1 b_{31} + a_2 b_{32} + c_{21} b_{32} + b_{21} c_{32} + a_3 b_{31} + a_3 b_{32}) + \omega_4[b_{41} a_1 + b_{42}(a_2 + c_{21}) + c_{42} b_{21} + b_{43}(c_{31} + c_{32} + a_3) + c_{41}(b_{31} + b_{32}) + a_4(b_{41} + b_{42} + b_{43})] = \tfrac{1}{3} \quad (28)$$

$$\omega_2 a_2 r_2 + \omega_3(c_{32} r_2 + a_3 r_3) + \omega_4(c_{42} r_2 + c_{43} r_3 + a_4 r_4) = \tfrac{1}{6} \quad (29)$$

$$\omega_2 b_{21} s_1 + \omega_3(b_{31} s_1 + b_{32} s_2) + \omega_4(b_{41} s_1 + b_{42} s_2 + b_{43} s_3) = \tfrac{1}{6} \quad (30)$$

Method A :

$$\omega_1 a_1^2 + \omega_2(c_{21} + a_2)^2 + \omega_3(c_{31} + c_{32} + a_3)^2 + \omega_4(c_{41} + c_{42} + c_{43} + a_4)^2 = \tfrac{1}{3} \quad (31)$$

$$\omega_1 a_1 s_1 + \omega_2 s_2(c_{21} + a_2) + \omega_3 s_3(c_{31} + c_{32} + a_3) + \omega_4 s_4(c_{41} + c_{42} + c_{43} + a_4) = \tfrac{1}{3} \quad (32)$$

Method B :

$$\omega_2 c_{21}^2 + \omega_3(c_{31} + c_{32})^2 + \omega_4(c_{41} + c_{42} + c_{43})^2 = \tfrac{1}{3} \quad (33)$$

$$\omega_2 s_2 c_{21} + \omega_3 s_3(c_{31} + c_{32}) + \omega_4 s_4(c_{41} + c_{42} + c_{43}) = \tfrac{1}{3} \quad (34)$$

Method C :

$$\omega_2(c_{21}^2 + 2a_2 c_{21}) + \omega_3[(c_{31} + c_{32})^2 + 2a_3(c_{31} + c_{32})] + \omega_4[(c_{41} + c_{42} + c_{43})^2 + 2a_4(c_{41} + c_{42} + c_{43})] = \tfrac{1}{3} \quad (35)$$

$$\omega_1 a_1 s_1 + \omega_2 s_2(c_{21} + a_2) + \omega_3 s_3(c_{31} + c_{32} + a_3) + \omega_4 s_4(c_{41} + c_{42} + c_{43} + a_4) = \tfrac{1}{3} \quad (36)$$

For up to second-order accuracy, direct combination of an explicit and an 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 an explicit and an implicit methods will likely be only second-order accurate because of the coupling between the explicit and implicit terms.

We search for the optimal parameters in the additive semi-implicit Runge-Kutta schemes by simultaneously imposing the stability and accuracy conditions discussed above. To be consistent with the explicit Runge-Kutta methods, the coefficients for the t terms are set to be

$$r_i = \sum_{j=1}^{i-1} b_{ij} \quad (37)$$

for all the three methods. But for the implicit t terms, method A is different from method B and C. For method A,

$$s_i = a_i + \sum_{j=1}^{i-1} c_{ij} \quad (38)$$

For method B and method C,

$$s_i = r_i \quad (39)$$

The reason for this is obviously that if equation (38) were available for method C, then combine Equations (21) and (35) we have

$$\omega_1 a_1^2 + \omega_2 a_2^2 + \omega_3 a_3^2 + \omega_4 a_4^2 = 0 \quad (40)$$

That means there are no solutions that satisfy the conditions of $\omega_i > 0$ and $a_i > 0$. For method B, s_1=0, but a_1 is required to be $a_1 > 0$ for the stiff terms.

The difference between the set of accuracy equations derived for conventional explicit Runge-Kutta scheme and for the explicit part of the semi-implicit scheme differs only in the additional equations generated by

the third-order cross terms as a result of the coupling between the explicit and implicit terms. And similarly the accuracy equations for implicit part of the semi-implicit method is same as the conventional implicit method for first- and second-order schemes only, excluding third-order schemes.

Since it is difficult to obtain close form solution of the accuracy and stability equations for a coupled semi-implicit scheme, a numerical searching method is used to find the optimal parameters. A computer program were developed to conduct extensive search for the optimal parameters numerically. The computational search makes it possible to locate the optimal parameters to satisfy both the stability and accuracy conditions for the methods.

First-Order Additive Semi-Implicit Runge-Kutta Methods

The expressions for the first-order methods are
ASIRK-1A Method:

$$\begin{cases} \mathbf{k}_1 = h\{\mathbf{f}(t_n, \mathbf{u}^n) + \mathbf{g}(t_n + s_1 h, \mathbf{u}^n + a_1 \mathbf{k}_1)\} \\ \mathbf{u}^{n+1} = \mathbf{u}^n + \omega_1 \mathbf{k}_1 \end{cases} \tag{41}$$

ASIRK-1B and ASIRK-1C Methods:

$$\begin{cases} [\mathbf{I} - h a_1 \mathbf{J}(t_n, \mathbf{u}^n)]\,\mathbf{k}_1 = h\{\mathbf{f}(t_n, \mathbf{u}^n) + \mathbf{g}(t_n, \mathbf{u}^n)\} \\ \mathbf{u}^{n+1} = \mathbf{u}^n + \omega_1 \mathbf{k}_1 \end{cases} \tag{42}$$

The coefficients are the same as the autonomous systems derived in Ref. [15]:

$$\omega_1 = a_1 = s_1 = 1 \tag{43}$$

The coefficients are the same for Methods A, B, C for the first-order case. The stability condition for the first-order additive semi-implicit Runge-Kutta methods is the same as first-order explicit Runge-Kutta methods for $h\lambda_f$ and is strongly A-stable for $h\lambda_g$.

Second-Order Additive Semi-Implicit Runge-Kutta Methods

The expressions for the second-order methods are:

ASIRK-2A Method:

$$\begin{cases} \mathbf{k}_1 = h\{\mathbf{f}(t_n, \mathbf{u}^n) + \mathbf{g}(t_n + s_1 h, \mathbf{u}^n + a_1 \mathbf{k}_1)\} \\ \mathbf{k}_2 = h\{\mathbf{f}(t_n + r_2 h, \mathbf{u}^n + b_{21} \mathbf{k}_1) \\ \qquad\qquad +\mathbf{g}(t_n + s_2 h, \mathbf{u}^n + c_{21} \mathbf{k}_1 + a_2 \mathbf{k}_2)\} \\ \mathbf{u}^{n+1} = \mathbf{u}^n + \omega_1 \mathbf{k}_1 + \omega_2 \mathbf{k}_2 \end{cases} \tag{44}$$

ASIRK-2B Method:

$$\begin{cases} [\mathbf{I} - h a_1 \mathbf{J}(t_n, \mathbf{u}^n)]\,\mathbf{k}_1 = h\{\mathbf{f}(t_n, \mathbf{u}^n) + \mathbf{g}(t_n, \mathbf{u}^n)\} \\ [\mathbf{I} - h a_2 \mathbf{J}(t_n, \mathbf{u}^n)]\,\mathbf{k}_2 = h\{\mathbf{f}(t_n + r_2 h, \mathbf{u}^n + b_{21} \mathbf{k}_1) \\ \qquad\qquad +\mathbf{g}(t_n + s_2 h, \mathbf{u}^n + c_{21} \mathbf{k}_1)\} \\ \mathbf{u}^{n+1} = \mathbf{u}^n + \omega_1 \mathbf{k}_1 + \omega_2 \mathbf{k}_2 \end{cases} \tag{45}$$

ASIRK-2C Method:

$$\begin{cases} [\mathbf{I} - h a_1 \mathbf{J}(t_n, \mathbf{u}^n)]\,\mathbf{k}_1 = h\{\mathbf{f}(t_n, \mathbf{u}^n) + \mathbf{g}(t_n, \mathbf{u}^n)\} \\ [\mathbf{I} - h a_2 \mathbf{J}(t_n + s_2 h, \mathbf{u}^n + c_{21} \mathbf{k}_1)]\,\mathbf{k}_2 = h\{\mathbf{f}(t_n + \\ \quad r_2 h, \mathbf{u}^n + b_{21} \mathbf{k}_1) + \mathbf{g}(t_n + s_2 h, \mathbf{u}^n + c_{21} \mathbf{k}_1)\} \\ \mathbf{u}^{n+1} = \mathbf{u}^n + \omega_1 \mathbf{k}_1 + \omega_2 \mathbf{k}_2 \end{cases} \tag{46}$$

Similar to the first-order case, the coefficients for the second-order methods are the same as the corresponding coefficients for a autonomous systems, and the coefficients are the same for all three methods. Ref.[15] shows two sets of coefficients:

Case I:

$$\begin{array}{lll} \omega_1 = \frac{1}{2} & \omega_2 = \frac{1}{2} & b_{21} = 1 \\ a_1 = \frac{1}{4} & a_2 = \frac{1}{3} & c_{21} = \frac{5}{12} \end{array}$$

Case II:

$$\begin{array}{lll} \omega_1 = \frac{1}{2} & \omega_2 = \frac{1}{2} & b_{21} = 1 \\ a_1 = 1 - \frac{\sqrt{2}}{2} & a_2 = 1 - \frac{\sqrt{2}}{2} & c_{21} = \sqrt{2} - 1 \end{array}$$

where r_i and s_i are given by Eqs. (37) and (38) or (39)

The methods are second-order accurate and strongly A-stable for the implicit term $h\lambda_g$.

Third-Order Additive Semi-Implicit Runge-Kutta Methods

ASIRK-3A Method

$$\begin{cases} \mathbf{k}_1 = h\{\mathbf{f}(t_n, \mathbf{u}^n) + \mathbf{g}(t_n + s_1 h, \mathbf{u}^n + a_1 \mathbf{k}_1)\} \\ \mathbf{k}_2 = h\{\mathbf{f}(t_n + r_2 h, \mathbf{u}^n + b_{21} \mathbf{k}_1) \\ \qquad\qquad +\mathbf{g}(t_n + s_2 h, \mathbf{u}^n + c_{21} \mathbf{k}_1 + a_2 \mathbf{k}_2)\} \\ \mathbf{k}_3 = h\{\mathbf{f}(t_n + r_3 h, \mathbf{u}^n + b_{31} \mathbf{k}_1 + b_{32} \mathbf{k}_2) \\ \qquad +\mathbf{g}(t_n + s_3 h, \mathbf{u}^n + c_{31} \mathbf{k}_1 + c_{32} \mathbf{k}_2 + a_3 \mathbf{k}_3)\} \\ \mathbf{k}_4 = h\{\mathbf{f}(t_n + r_4 h, \mathbf{u}^n + b_{41} \mathbf{k}_1 + b_{42} \mathbf{k}_2 + b_{43} \mathbf{k}_3) \\ +\mathbf{g}(t_n + s_4 h, \mathbf{u}^n + c_{41} \mathbf{k}_1 + c_{42} \mathbf{k}_2 + c_{43} \mathbf{k}_3 + a_4 \mathbf{k}_4)\} \\ \mathbf{u}^{n+1} = \mathbf{u}^n + \omega_1 \mathbf{k}_1 + \omega_2 \mathbf{k}_2 + \omega_3 \mathbf{k}_3 + \omega_4 \mathbf{k}_4 \end{cases} \tag{47}$$

ASIRK-3B Method

$$
\begin{cases}
[\mathbf{I} - ha_1\mathbf{J}(t_n,\mathbf{u}^n)]\,\mathbf{k}_1 = h\{\mathbf{f}(t_n,\mathbf{u}^n) + \mathbf{g}(t_n,\mathbf{u}^n)\} \\
[\mathbf{I} - ha_2\mathbf{J}(t_n,\mathbf{u}^n)]\,\mathbf{k}_2 = h\{\mathbf{f}(t_n + r_2h, \mathbf{u}^n + b_{21}\mathbf{k}_1) \\
\qquad\qquad +\mathbf{g}(t_n + s_2h, \mathbf{u}^n + c_{21}\mathbf{k}_1)\} \\
[\mathbf{I} - ha_3\mathbf{J}(t_n,\mathbf{u}^n)]\,\mathbf{k}_3 = h\{\mathbf{f}(t_n + r_3h, \mathbf{u}^n + b_{31}\mathbf{k}_1 \\
\quad +b_{32}\mathbf{k}_2) + \mathbf{g}(t_n + s_3h, \mathbf{u}^n + c_{31}\mathbf{k}_1 + c_{32}\mathbf{k}_2)\} \\
[\mathbf{I} - ha_4\mathbf{J}(t_n,\mathbf{u}^n)]\,\mathbf{k}_4 = h\{\mathbf{f}(t_n + r_4h, \mathbf{u}^n + b_{41}\mathbf{k}_1 \\
\quad +b_{42}\mathbf{k}_2 + b_{43}\mathbf{k}_3) + \mathbf{g}(t_n + s_4h, \mathbf{u}^n + c_{41}\mathbf{k}_1 \\
\qquad\qquad +c_{42}\mathbf{k}_2 + c_{43}\mathbf{k}_3)\} \\
\mathbf{u}^{n+1} = \mathbf{u}^n + \omega_1\mathbf{k}_1 + \omega_2\mathbf{k}_2 + \omega_3\mathbf{k}_3 + \omega_4\mathbf{k}_4
\end{cases}
\tag{48}
$$

ASIRK-3C Method

$$
\begin{cases}
[\mathbf{I} - ha_1\mathbf{J}(t_n,\mathbf{u}^n)]\,\mathbf{k}_1 = h\{\mathbf{f}(t_n,\mathbf{u}^n) + \mathbf{g}(t_n,\mathbf{u}^n)\} \\
[\mathbf{I} - ha_2\mathbf{J}(t_n + s_2h, \mathbf{u}^n + c_{21}\mathbf{k}_1)]\,\mathbf{k}_2 = h\{\mathbf{f}(t_n + r_2h, \\
\quad \mathbf{u}^n + b_{21}\mathbf{k}_1) + \mathbf{g}(t_n + s_2h, \mathbf{u}^n + c_{21}\mathbf{k}_1)\} \\
[\mathbf{I} - ha_3\mathbf{J}(t_n + s_3h, \mathbf{u}^n + c_{31}\mathbf{k}_1 + c_{32}\mathbf{k}_2)]\,\mathbf{k}_3 = \\
\qquad h\{\mathbf{f}(t_n + r_3h, \mathbf{u}^n + b_{31}\mathbf{k}_1 + b_{32}\mathbf{k}_2) \\
\qquad +\mathbf{g}(t_n + s_3h, \mathbf{u}^n + c_{31}\mathbf{k}_1 + c_{32}\mathbf{k}_2)\} \\
[\mathbf{I} - ha_4\mathbf{J}(t_n + s_4h, \mathbf{u}^n + c_{41}\mathbf{k}_1 + c_{42}\mathbf{k}_2 + c_{42}\mathbf{k}_3)]\,\mathbf{k}_4 \\
\quad = h\{\mathbf{f}(t_n + r_4h, \mathbf{u}^n + b_{41}\mathbf{k}_1 + b_{42}\mathbf{k}_2 + b_{43}\mathbf{k}_3) \\
\quad + \mathbf{g}(t_n + s_4h, \mathbf{u}^n + c_{41}\mathbf{k}_1 + c_{42}\mathbf{k}_2 + c_{43}\mathbf{k}_3)\} \\
\mathbf{u}^{n+1} = \mathbf{u}^n + \omega_1\mathbf{k}_1 + \omega_2\mathbf{k}_2 + \omega_3\mathbf{k}_3 + \omega_4\mathbf{k}_4
\end{cases}
\tag{49}
$$

We use four stages in ASIRK-3 to get third-order accuracy. In fact, if three-stage ASIRK schemes were used, there would be 18 undetermined parameters for the third-order method. After satisfying the accuracy and strong A-stability conditions, there were only two free parameters for the ASIRK-3 methods. If ω_1 and ω_2 were chosen as free parameters, then the third-order accuracy equations and one stability equations would be solved exactly. We searched ω_1 and ω_2 from 0 to 1 and found that no solution satisfies all $a_i > 0$. For a given set of ω_1 and ω_2, the accuracy conditions lead to either a_1 or a_3 to be zero. This is not an acceptable solution because the resulting ASIRK-3 schemes do not satisfy stability condition (12). That means if a three-stage additive semi-implicit Rungge-Kutta scheme was used to obtain third-order accuracy for non-autonomous differential equations, one stage whould be a full explicit scheme. In order to get the solutions that satisfy the third-order accuracy and stability conditions, meanwhile all a_i are positive, four-stage semi-implicit RK schemes are needed to add more parameters to get the third-order accuracy scheme.

The searching procedure for the numerical solutions is as follows: First, ω_1, ω_2 and ω_4 are chosen as free parameters to obtain ω_3 from Equation (15). Then, b_{32}, b_{42} and b_{43} are chosen as free parameters to determine r_2, r_3 and r_4 from Equations (16),(20) and (24). For method B and method C, s_1-s_3 are determined by $s_i = r_i$; for method A, another parameter s_4 is chosen to solve Equations (17),(21) and (30) for s_1, s_2 and s_3. Next, Equations (22),(23),(28) and (29) are solved for a_1, a_2, a_3 and a_4 with c_{32}, c_{42} and c_{43} as free parameters. Finally, the solutions are checked by stability Equation (10) and (14). If the solutions do not satisfy those conditions, the parameters are changed to repeat the procedure until all $a_i > 0$. The searching range for free parameters of ω_i and s_4 is from 0 to 1. The range for b_{ij} and c_{ij} is from -2 to 2.

The parameters for the explicit and implicit terms are different for the three versions of ASIRK3 methods because the explicit/implicit coupling.

ASIRK-3A Method:

$\omega_1 = 0.13$	$\omega_2 = \frac{1}{4}$
$\omega_3 = 0.52$	$\omega_4 = \frac{1}{10}$
$b_{21} = 0.338170$	$b_{31} = -0.019084$
$b_{32} = 0.779584$	$b_{41} = -0.300000$
$b_{42} = 0.200000$	$b_{43} = 0.300000$
$a_1 = 1.174810$	$a_2 = 0.526766$
$a_3 = 0.158717$	$a_4 = 0.100000$
$a_{21} = -0.293999$	$c_{31} = 0.149135$
$c_{32} = 0.200000$	$c_{41} = -1.130818$
$c_{42} = 1.780818$	$c_{43} = -0.500000$

ASIRK-3B Method:

$\omega_1 = \frac{1}{8}$	$\omega_2 = \frac{1}{4}$
$\omega_3 = 0.525$	$\omega_4 = \frac{1}{10}$
$b_{21} = 0.309921$	$b_{31} = 0.169758$
$b_{32} = 0.591232$	$b_{41} = -0.37000$
$b_{42} = -0.55000$	$b_{43} = 1.14999$
$a_1 = 0.130476$	$a_2 = 0.052913$
$a_3 = 0.067873$	$a_4 = 0.424531$
$c_{21} = 0.160000$	$c_{31} = 0.361513$
$c_{32} = 0.400000$	$c_{41} = -0.974181$
$c_{42} = -0.500000$	$c_{43} = 1.000000$

ASIRK-3C Method:

$\omega_1 = \frac{1}{8}$	$\omega_2 = \frac{1}{4}$
$\omega_3 = 0.525$	$\omega_4 = \frac{1}{10}$

$b_{21} = 0.324692$ $b_{31} = -0.000745$
$b_{32} = 0.767118$ $b_{41} = 0.300000$
$b_{42} = -1.000000$ $b_{43} = 0.890000$
$a_1 = 0.170366$ $a_2 = 0.107914$
$a_3 = 0.041351$ $a_4 = 0.029692$
$c_{21} = 0.150000$ $c_{31} = 0.033636$
$c_{32} = 0.706262$ $c_{41} = 0.314661$
$c_{42} = -1.000000$ $c_{43} = 0.696340$

The parameters with longer significant digits can be obtained from the authors. The methods using the coefficients above are third-order accurate and strongly A-stable for the implicit term $h\lambda_g$.

Test Cases

Systems of Ordinary Differential Equations

In order to check the temporal accuracy of semi-implicit Runge-Kutta method derived in this paper, two cases of systems of ordinary differential equations are tested.

Case I Non-stiff Systems of ODE

First, we consider a system of ordinary differential equations as follows

$$\frac{d\mathbf{u}}{dt} = \mathbf{A}\mathbf{u} + \mathbf{f}(t) \tag{50}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -5 & -4 \end{bmatrix} \quad \mathbf{f} = \begin{bmatrix} 0 \\ 0 \\ -4\sin(t) - 2\cos(t) \end{bmatrix} \tag{51}$$

the initial condition is $\mathbf{u}(_{t=0}) = \{1, 0, -1\}$.

Because this simple case has an exact solution of $u_1 = \cos(t)$, we can evaluate the temporal order of accuracy of semi-implicit scheme by the refinement of time steps. The errors of numerical solution u_h is

$$e_h = u_{ex} - u_h \tag{52}$$

where e_h depends on the time step h. For a p-th order numerical method,

$$\frac{e_h}{e_{\frac{h}{2}}} = R_p = 2^p \tag{53}$$

Table 1: Temporary Accuracy of the ASIRK-2A and ASIRK-3A Methods($t = 2.5$, and $u_{ex} = -0.80114362$)

	RK-3		ASIRK-3A		ASIRK-2A	
Δt	e_3	R_3	e_3	R_3	e_2	R_2
h=0.25	1.26D-3	8.2	1.40D-3	7.1	1.11D-3	4.2
h/2	1.53D-4	8.1	1.96D-4	7.6	2.65D-4	4.1
h/4	1.88D-5	8.1	2.58D-5	7.8	6.50D-5	4.0
h/8	2.33D-6	8.0	3.29D-6	7.9	1.61D-6	4.0
h/16	2.90D-7	8.0	4.15D-7	8.0	4.01D-7	4.0
h/32	3.62D-8	-	5.20D-8	-	1.00D-7	-

Table 1 shows the results of the grid refinement study. The results of 2nd-order and 3rd-order additive semi-implicit Runge-Kutta scheme (ASIRK-2A and ASIRK-3A) using the same time steps are given in this table. As a comparison, the results of conventional third-order explicit Runge-Kutta method(RK-3) are also listed in the table. Because this case is a non-stiff system of ordinary differential equations, the right hand part of the equations are treated as the part $\mathbf{g}$ in equation (2). In another word, r_i and b_{ij} in Eq.(47) are set to zero, the equations are calculated with implicit method. Therefore there are some difference between the 3rd-order explicit RK method and semi-implicit RK method. Just as predicted by equation (53), the value of R_p gives out the temporal accuracy of the schemes. Table 1 shows that the ASIRK-2A is second-order accuracy and the ASIRK-3A using four steps is third-order temporal accuracy.

Case II Systems of ODE with Stiff Terms

Then we tested the scheme of ASIRK-3C for systems of ordinary differential equations with stiff terms given by

$$\frac{d\mathbf{u}}{dt} = \mathbf{A}(t)\mathbf{u} + \mathbf{f_1}(t) + \frac{1}{\epsilon}\left[\mathbf{B}(t)\mathbf{u} + \mathbf{f_2}(t)\right] \tag{54}$$

where

$$\mathbf{A} = \begin{bmatrix} 12.2 & 5\sin(t) & sin(4t) \\ 4 & 15 & 2\cos(2t) \\ 3 & 4\cos(3t) & 7 \end{bmatrix} \quad \mathbf{f_1} = \begin{bmatrix} \cos(t) \\ 2\sin(t) \\ 1 \end{bmatrix} \tag{55}$$

$$\mathbf{B} = \begin{bmatrix} 42.2 & 50.1 & -42.1\sin(2t) \\ -66.1 & -58 & 58.1\cos(2t) \\ 26.1 & 42.1 & -34t^{0.1} \end{bmatrix} \quad \mathbf{f_2} = \begin{bmatrix} \sin(t) \\ \cos(t) \\ \sin(2t) \end{bmatrix} \tag{56}$$

where ϵ is a small parameter chosen to be 0.01. The results of ASIRK-2C and ASIRK-3C are listed in Table 2. The exact solution u_{ex} used in equation (52) in

this case is obtained from the Richardson extrapolated solution at the smallest time step. The results show that the ASIRK-3C method is third-order accuracy.

Table 2: Temporary Accuracy of ASIRK-2C and ASIRK-3C Method for Stiff Equations(t = 0.3967, and u_{ex} = 0.310395394964)

	ASIRK-3C		ASIRK-2C	
Δt	e_3	R_3	e_2	R_2
h=0.001	0.12	5.6	0.47	10.4
h/2	2.14D-2	7.5	4.51D-2	3.9
h/4	2.87D-3	10.6	1.14D-2	4.0
h/8	2.71D-4	10.7	2.86D-3	4.0
h/16	2.53D-5	10.0	7.15D-4	4.0
h/32	2.54D-6	9.2	1.78D-4	4.0
h/64	2.77D-7	-	4.47D-5	-

Two-Dimensional Viscous Reactive Flows

After validating the accuracy of the present ASIRK-3 method, we apply these schemes to a two-dimensional viscous flows over a circular cylinder with thermo-chemical nonequilibrium. There are two features of viscous reactive flows that are stiff for time integration, one is the stretched grid in the viscous boundary layer near the wall, and another is the source terms due to the finite rate thermal and chemical processes. The ASIRK-3 methods derived in this paper are applied to these stiff equations. The free stream conditions are

$$C_{N_2} = 0.927 \qquad C_N = 0.073$$
$$u_\infty = 5590m/s \qquad T_\infty = 1833K$$
$$p_\infty = 2910pa \qquad T_w = 1000K$$

To resolve the viscous boundary layer on the wall, the grids are stretched in the direction normal to the wall. The total number of grid points used in numerical calculations is 42×142. Fig.1 shows the structured grids used in the calculation. The Navier-Stoke equations are solved by finite volume method. A second-order TVD scheme is used for the descretization of spatial fluxes and the 4-stage semi-implicit Runge-Kutta (ASIRK-3C) method is used to integrate the equations in time. The source terms due to thermo-chemical nonequilibrium, as well as the fluxes in the direction normal to the wall, are treated implicitly. Fig. 2 shows the pressure contours of the mixture gas. Fig.3 and Fig.4 are the distributions of pressure along the stagnation line and along the wall, respectively. Fig.3 shows that the pressure in the shock layer increases about two order-of-magnitude as compared with the freestream conditionis. The change of pressure indicates that the thickness of the shock front is about two grids spatial distance. Fig.4 shows that the highest pressure region on the wall is near the stagnation line. Figs.5-8 are the species concentrations of N_2 and N. Fig.5 is the distributions of mass fractions along the stagnation line and Fig.6 is along the wall. The figure shows that N_2 is dissociated behind the bow shock due to high temperature, but it recombines in the boundary layer because of the low wall temperature. Fig. 7 and Fig. 8 are contours of N_2 and N mass fractions. The calculations show that the new method is robust and accurate.

Conclusions

The additive semi-implicit Runge-Kutta methods of up to third-order accuracy for non-autonomous differential equations have been derived and tested in this paper. Unlike the methods for autonomous differential equations, four steps are required to get third-order temporal accuracy and every step is implicit for the stiff terms. The temporal order of accuracy of the four-stage additive semi-implicit Runge-Kutta methods have been validated by the refinement study of time steps. The application of the third-order ASIRK method to two dimensional thermo-chemical reactive flow shows that the semi-implicit methods are efficient and robust for the stiff differential equations.

Acknowledgments

This research was supported by the Air Force Office of Scientific Research under grant numbers F49620-94-1-0019 and F49620-95-1-0405 monitored by Dr. Len Sakell. More information on this work and related ongoing research can be found on the World Wide Web at http://cfdlab5.seas.ucla.edu.

References

[1] C.-H. Chiu and X. Zhong. Numerical Simulation of Transient Hypersonic Flow Using the Essentialy Nonoscillatory Schemes. *AIAA Journal*, Vol. 34, No. 4, pp. 655–661, 1995. (Also AIAA paper 95-0469, 1995).

[2] X. Zhong, X. Joubert, and T. K. Lee. The interaction of freestream disturbances and the bow shock in hypersonic flow past a cylinder. 1995. Presented in 20th International Symp. on Shock Waves, July 23-28, 1995, Pasadena, California.

[3] P. Moin and J. Kim. On the numerical solution of the time-dependent viscous incompressible fluid flows involving solid boundaries. *J. of Computational Physics*, 35:381–392, 1980.

[4] Thomas A. Zang and M. Y. Hussaini. Numerical experiments on subcritical transition mechanisms. *AIAA Paper 81-1227*, 1981.

[5] R. S. Rogallo and P. Moin. Numerical simulation of turbulent flows. *Ann. Rev. Fluid Mech.*, 16:99–137, 1984.

[6] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods in Fluid Dynamics.* Springer-Verlag, 1988.

[7] L. Kleiser and T. A. Zang. Numerical Simulation of Transition in Wall-Bounded Shear Flows. *Ann. Rev. Fluid Mech.*, Vol. 23, pp. 495–535, 1991.

[8] T. R. A. Bussing and E. M. Murman. A finite volume method for the calculation of compressible chemically reacting flows. *AIAA Paper 85-0296*, 1985.

[9] J. P. Drummond, R. C. Rogers, and M. Y. Hussaini. Spectral methods for modeling supersonic chemically reacting flow fields. *AIAA Paper 85-0302*, 1985.

[10] H. C. Yee and J. L. Shinn. Semi-implicit and fully implicit shock capturing methods for hypersonic conservation laws with stiff source terms. *AIAA Paper 87-1116*, 1987.

[11] R. J. Leveque and H. C. Yee. A study of numerical methods for hyperbolic conservation laws with stiff source terms. *J. of Computational Physics*, 86:187–210, 1990.

[12] G. J. Cooper and A. Sayfy. Additive methods for numerical solutions of ordinary differential equations. *Mathematics of Computation*, 35(152):1159–1172, October 1980.

[13] G. J. Cooper and A. Sayfy. Additive rungekutta methods for stiff ordinary differential equations. *Mathematics of Computation*, 40(161):207–218, January 1983.

[14] X. Zhong. Semi-Implicit Runge-Kutta Schemes for Direct Numerical Simulation of Transient High-Speed Reactive Flows. *To appear in the Journal of Computational Physics*, 1996.

[15] X. Zhong. New High-Order Semi-implicit Runge-Kutta Schemes for Computing Transient Nonequilibrium Hypersonic Flow. *AIAA Paper 95-2007*, 1995.

[16] H. H. Rosenbrock. Some general implicit processes for the numerical solution of differential equations. *Comput. J.*, 5:329, 1963.

[17] J. D. Lambert. *Computational Methods in Ordinary Differential Equations.* John Wiley & Sons, 1973.

[18] J. C. Butcher. Implicit Runge-Kutta processes. *Math. Comp.*, 18:50–64, 1964.

[19] E. Hairer, G. Bader, and CH. Lubich. On the stability of semi-implicit methods for ordinary differential equations. *BIT*, 22:211–232, 1982.

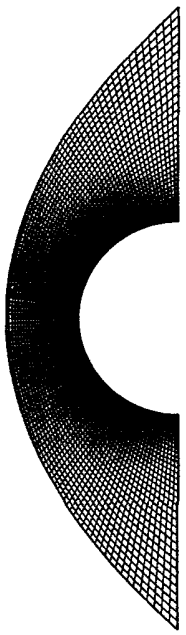

Figure 1: Computational Grids.

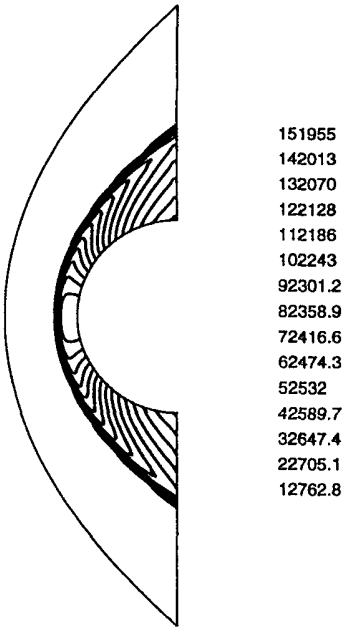

Figure 2: Pressure contours of the mixtured nitrogen.

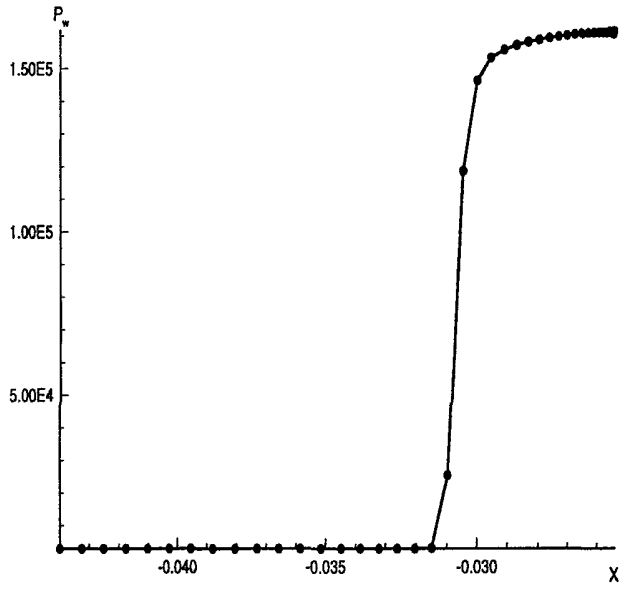

Figure 3: Mixtured pressure on the stagnation line.

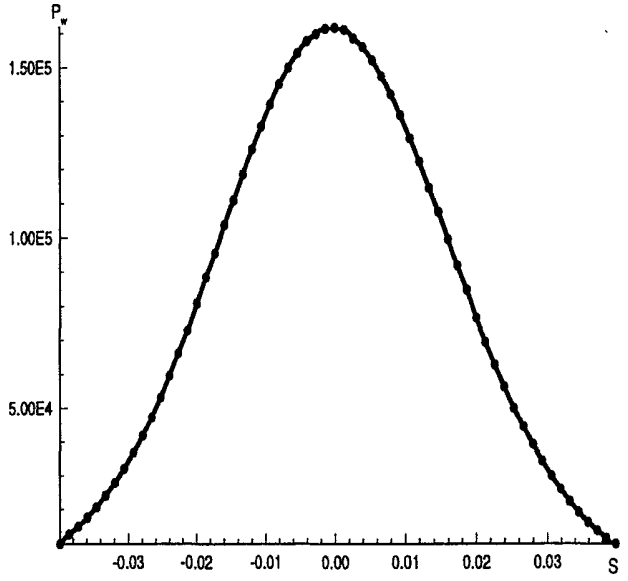

Figure 4: Pressure of the mixtured nitrogen on the wall.

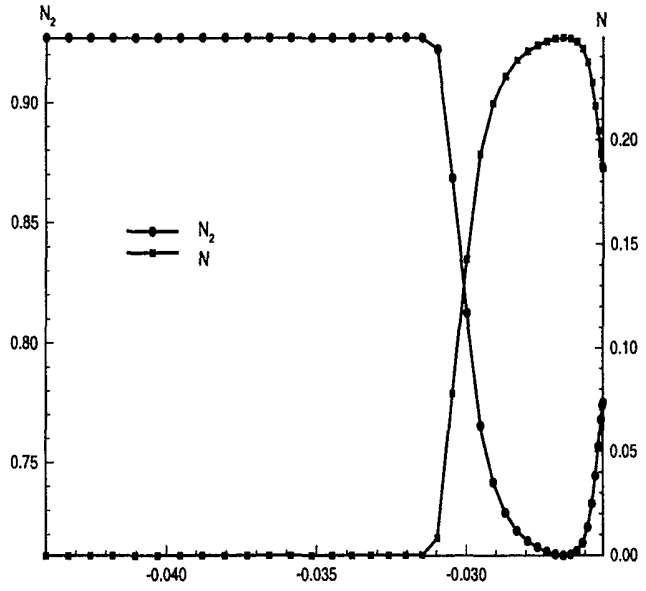

Figure 5: Species concetrations on the stagnation line.

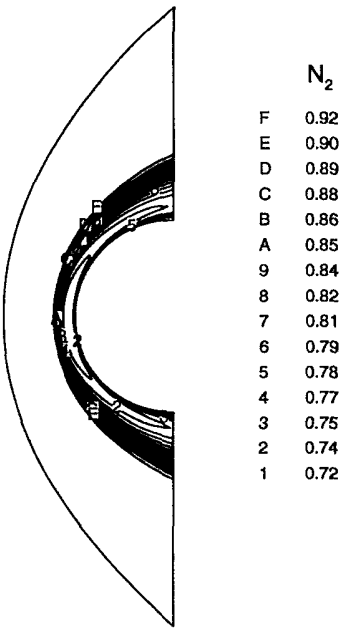

Figure 7: Species contours of N_2.

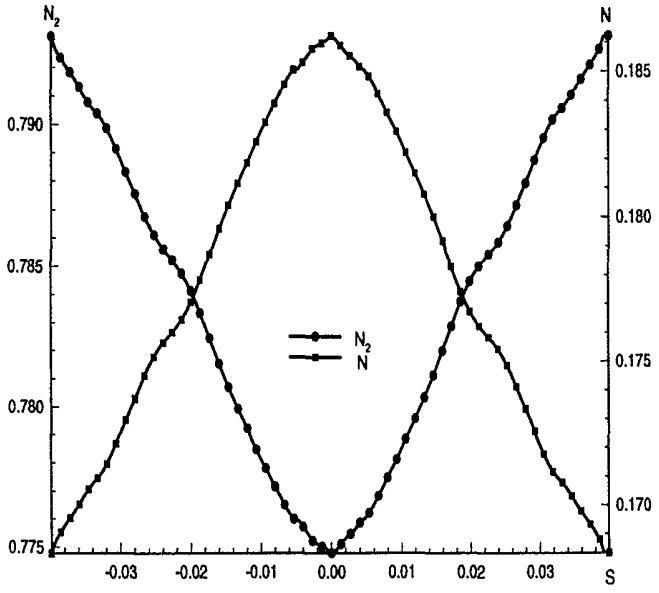

Figure 6: Species concentrations on the wall.

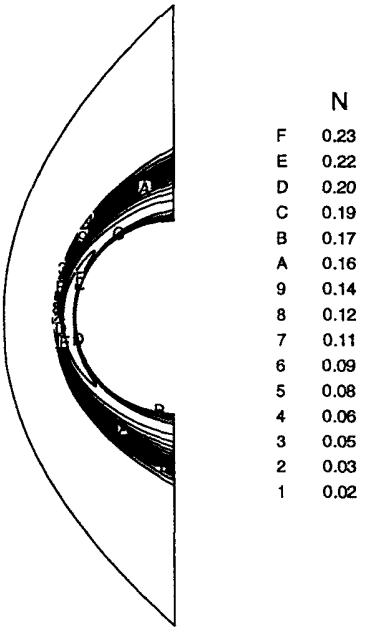

Figure 8: Species contours of N.