

Research Article

A Strongly A-Stable Time Integration Method for Solving the Nonlinear Reaction-Diffusion Equation

Wenyuan Liao

Department of Mathematics and Statistics, University of Calgary, 2500 University Drive NW, Calgary, AB, Canada T2N 1N4

Correspondence should be addressed to Wenyuan Liao; wliao@ucalgary.ca

Received 28 July 2014; Accepted 17 October 2014

Academic Editor: Santanu Saha Ray

Copyright © 2015 Wenyuan Liao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The semidiscrete ordinary differential equation (ODE) system resulting from compact higher-order finite difference spatial discretization of a nonlinear parabolic partial differential equation, for instance, the reaction-diffusion equation, is highly stiff. Therefore numerical time integration methods with stiff stability such as implicit Runge-Kutta methods and implicit multistep methods are required to solve the large-scale stiff ODE system. However those methods are computationally expensive, especially for nonlinear cases. Rosenbrock method is efficient since it is iteration-free; however it suffers from order reduction when it is used for nonlinear parabolic partial differential equation. In this work we construct a new fourth-order Rosenbrock method to solve the nonlinear parabolic partial differential equation supplemented with Dirichlet or Neumann boundary condition. We successfully resolved the phenomena of order reduction, so the new method is fourth-order in time when it is used for nonlinear parabolic partial differential equations. Moreover, it has been shown that the Rosenbrock method is strongly A-stable hence suitable for the stiff ODE system obtained from compact finite difference discretization of the nonlinear parabolic partial differential equation. Several numerical experiments have been conducted to demonstrate the efficiency, stability, and accuracy of the new method.

1. Introduction

Let us consider the following parabolic partial differential equation:

$$u_t = Du_{xx} + f(u, x, t), \quad (x, t) \in (a, b) \times (0, T], \quad (1)$$

with the initial condition:

$$u(x, 0) = u_0(x), \quad x \in [a, b], \quad (2)$$

where D is a positive constant describing the diffusion property and $f(u, x, t)$ is a function representing the reaction term, which is nonlinear on u . The unknown function u represents, depending on the applications, variables such as mass concentration in chemical reaction process, temperature in heat conduction, neutron flux in nuclear reactors, and population density in population dynamics. On the boundary, either Dirichlet condition

$$u(a, t) = g_1(t), \quad u(b, t) = g_2(t), \quad 0 \leq t \leq T \quad (3)$$

or Neumann condition

$$u_x(a, t) = g_3(t), \quad u_x(b, t) = g_4(t), \quad 0 \leq t \leq T \quad (4)$$

is specified, where $g_1, g_2, g_3,$ and g_4 are sufficiently smooth functions. Here in this paper we restrict our attention on Dirichlet and Neumann boundary conditions, while the developed techniques can be easily extended to Robin boundary condition.

Efficient and accurate numerical methods for solving (1) had attracted great attentions from scientists and engineers, as for many application problems in science and engineering, it is preferable to use high-order compact numerical algorithms to compute accurate solutions. In the past several decades a great deal of work has been done in the development of efficient, accurate, and robust numerical algorithm for solving such problem. For more details, the reader is referred to [1–4].

Since both temporal and spatial derivatives are involved in the equation, we discuss the numerical treatments in

time and space separately. Here we first apply the high-order compact finite difference approximation to the spatial derivative, so a semidiscrete ODE system is obtained, which is then solved by a fourth-order Rosenbrock method that will be discussed later.

Recently there have been attempts to develop high-order compact scheme for the spatial derivative. In [5], a three-point combined compact difference scheme was proposed to approximate the first and second derivatives for problem with periodic boundary condition. The resulting scheme has up to sixth-order accuracy at all grid points including the boundary nodes for periodic boundaries; however it is only fourth-order accurate for nonperiodic boundary condition.

Because the semidiscrete ODE system obtained from spatial discretization, such as method of lines, of the nonlinear parabolic partial differential equation is highly stiff, the choices of time integration methods are limited to implicit methods only. Explicit algorithm is efficient in a single time step but suffers from strict step size restriction, which makes it less efficient. Implicit method, on the other side, is less efficient in a single step but the unconditional stability allows the use of larger time step; hence the overall computational efficiency can be significantly improved. One issue, however, is that the iteration is usually slow when large step size Δt is used. Also, due to the stiffness of the ODE system, only Newton-type iterative methods are applicable to solve the nonlinear algebraic system. Furthermore, strong A-stability or L-stability of the time integration method is necessary for error damping. A great deal of work has been done in the development of efficient time stepping methods for the stiff ODE system. In [6] explicit exponential Rosenbrock methods of order five have been constructed to solve the large-scale stiff ODE system. Through the derivation of stiff order condition, new pairs of embedded methods of higher-order can be obtained. Similarly, fifth-order explicit exponential Runge-Kutta methods were constructed to efficiently integrate the semilinear stiff problems in [7]. The authors have also shown that there does not exist an explicit exponential Runge-Kutta method of order 5 with less than or equal to 6 stages; therefore the resultant methods are 8-stage methods. In [8] a fourth-order time stepping method, which is a modification of the exponential time-differencing fourth-order Runge-Kutta method, has been developed for stiff ODEs. These methods are efficient and accurate. However, A-stability of the time stepping method is not sufficient for highly stiff problem. To overcome these difficulties, it is desirable to construct new algorithms with strong A-stability or L-stability that are free of solving nonlinear equations. It turns out that the Rosenbrock method, which was firstly reported by Rosenbrock [9] and then improved by Haines [10], responded to these issues with considerably satisfying and promising results.

The objective herein is to develop a strongly A-stable Rosenbrock method to solve the semidiscrete stiff ODEs resulting from compact high-order finite difference approximation of a semilinear parabolic partial differential equation. The rest of the paper is organized as the following. In Section 2, we discretize the spatial derivatives of the semilinear parabolic partial differential equation using a

fourth-order compact finite difference scheme, which is then combined with a newly proposed compact fourth-order boundary condition treatment to form the semidiscrete ODE system. In Section 3 we focus on the development of a fourth-order strongly A-stable Rosenbrock method and the stability analysis. Several numerical examples are used to demonstrate the accuracy and efficiency of the new algorithm in Section 4, which is followed by conclusions and possible future work.

2. Compact Fourth-Order Spatial Discretization

For the sake of simplicity, we assume that the 1D spatial domain $\Omega = [a, b]$ is divided into M subintervals with equal length $h = (b - a)/M$. Let $x_i = a + i \cdot h, i = 0, 1, \dots, M$, be the grid points. A variety of compact high-order discretizations can be utilized to approximate the second derivative u_{xx} in (1).

Here we introduce a compact finite difference scheme to approximate u_{xx} , such that the resulting semidiscrete ODE system is an accurate and compact approximation to the original semilinear parabolic partial differential equation. This operator-approximation based method has been widely used to solve various multidimensional problems. We first define the central finite difference operator δ_x^2 as

$$\delta_x^2 u_i = u_{i+1} - 2u_i + u_{i-1}; \quad (5)$$

then δ_x^2/h^2 gives second-order accurate approximation to u_{xx} . Using Taylor series to expand all terms on the right-hand side of (5), under the assumption that $u(x)$ is sufficiently smooth, we have

$$\frac{\delta_x^2}{h^2} u_i = u_{xx}(x_i) + \frac{h^2}{12} u_{xxxx}(x_i) + O(h^4), \quad 1 \leq i \leq M - 1. \quad (6)$$

To improve the above finite difference approximation to fourth-order accurate, one just needs to eliminate the second-order error term. Applying $\delta_x^2/12$ to both sides of (6), we have

$$\frac{\delta_x^2}{h^2} \frac{\delta_x^2}{12} u_i = u_{xxxx}(x_i) + O(h^4), \quad 1 \leq i \leq M - 1. \quad (7)$$

Combining (6) with (7), neglecting $O(h^4)$, we obtain the following fourth-order accurate approximation to u_{xx} at node x_i :

$$u_{xx}(x_i) \approx \frac{1}{h^2} \delta_x^2 \left(1 - \frac{\delta_x^2}{12} \right) u_i, \quad 1 \leq i \leq M - 1. \quad (8)$$

The drawback is that a five-point stencil is required; therefore the compactness is destroyed so the method becomes less efficient. Further investigation shows that the difference between $1 - \delta_x^2/12$ and $(1 + \delta_x^2/12)^{-1}$ is $O(h^4)$, so a natural way is to approximate $u_{xx}(x_i)$ as $\delta_x^2(1 + \delta_x^2/12)^{-1}u_i$, which is fourth-order accurate and compact.

Applying the fourth-order Padé approximation to u_{xx} in (1), we obtain the following ODE system

$$u'_i(t) = \frac{D}{h^2} \frac{\delta_x^2}{1 + \delta_x^2/12} u_i(t) + f(u_i(t), x_i, t), \quad (9)$$

$$1 \leq i \leq M - 1,$$

which is a fourth-order accurate approximation (in space) to the original semilinear parabolic partial differential equation defined in (1).

However, the above algorithm is difficult to implement, so we multiply $1 + \delta_x^2/12$ to both sides to obtain the following implicit ODE system:

$$\left(1 + \frac{\delta_x^2}{12}\right) u'_i(t) = \frac{D}{h^2} \delta_x^2 u_i(t) + \left(1 + \frac{\delta_x^2}{12}\right) f(u_i(t), x_i, t),$$

$$1 \leq i \leq M - 1, \quad 0 < t \leq T, \quad (10)$$

which can be written in vector form as

$$AU'(t) = F(U, X, t), \quad 0 < t \leq T, \quad (11)$$

where $U(t) = (u_1(t), u_2(t), \dots, u_{M-1}(t))$ is the discrete solution of (1) at time t , with $u_i(t) = u(x_i, t)$, A is an $(M - 1) \times (M - 1)$ tridiagonal matrix, and F is a vector-valued function defined through (10). To complete the ODE system, we need the boundary conditions at $x = x_0$ and $x = x_M$, which can be derived from the original boundary conditions defined in (3) or (4).

First, if the Dirichlet boundary condition (3) is specified, one can add the following two ODEs to (9):

$$u'_0(t) = g'_1(t), \quad u'_M(t) = g'_2(t). \quad (12)$$

Consequently the matrix is modified as

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \frac{1}{12} & \frac{5}{6} & \frac{1}{12} & 0 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots & \frac{1}{12} & \frac{5}{6} & \frac{1}{12} \\ 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & 1 \end{pmatrix}, \quad (13)$$

while the vector-valued function $F(U, X, t)$, after modifications, is defined as

$$F_0 = g'_1(t),$$

$$F_i = \frac{D}{h^2} \delta_x^2 u_i(t) + \left(1 + \frac{\delta_x^2}{12}\right) f(u_i(t), x_i, t),$$

$$1 \leq i \leq M - 1,$$

$$F_M = g'_2(t). \quad (14)$$

Alternatively, we can incorporate the boundary condition by replacing $u_0(t)$ and $u_M(t)$ in (10) with $g_1(t)$ and $g_2(t)$, respectively, so the ODE system equation (11) has only $M - 2$ equations.

As one can imagine, the situation is more complicated when the Neumann boundary condition (4) is specified. To complete the ODE system and maintain the higher-order overall accuracy, a compact fourth-order approximation of the Neumann boundary condition is needed. Let us use the boundary condition at $x = a$ as the example to demonstrate the idea of the new algorithm.

Unlike the Dirichlet boundary condition, which specifies the solution u on the boundary point explicitly, the Neumann boundary condition defines u_x at the boundary points; thus $u_0(t)$ and $u_M(t)$ need to be calculated along with solution at the interior grid points. Consequently, the range for i in (10) should be changed to $0 \leq i \leq M$, so A is an $(M + 1) \times (M + 1)$ matrix. To approximate the derivative at x_0 , we introduce a ghost point $x_{-1} = a - h$ and assume that (1) holds and the solution u is sufficiently smooth on the extended domain $[a - h, b]$. Let $u_{-1}(t)$ denote the solution at $x_{-1} = a - h$ and then apply the second-order central finite difference approximation to $u_x(a, t)$,

$$\frac{u_1(t) - u_{-1}(t)}{2h} = u_x(a, t) + \frac{h^2}{6} u_{xxx}(a, t) + O(h^4). \quad (15)$$

Taking partial derivative with respect to x on both sides of (1), we have

$$u_{xxx} = \frac{1}{D} (u_{xt} - f_x - f_u \cdot u_x). \quad (16)$$

Letting $x \rightarrow a$ in (16) and then applying the Neumann boundary condition (4), we obtain

$$u_{xxx}(a) = \frac{1}{D} (g'_3(t) - f_x(u_0(t), a, t) - f_u(u_0(t), a, t) \cdot g_3(t)). \quad (17)$$

Combining (15) with (17), we obtain the following fourth-order compact approximation for $u_{-1}(t)$:

$$u_{-1}(t) = u_1(t) - 2hg_3(t) - \frac{h^3}{3D} (g'_3(t) - f_x(u_0(t), a, t) - f_u(u_0(t), a, t) \cdot g_3(t)), \quad (18)$$

which involves $u_0(t)$ and $u_1(t)$ only, so the compact structure is preserved.

Similarly, the fourth-order compact approximation for $u_{M+1}(t)$ can be derived as

$$\begin{aligned} u_{M+1}(t) &= u_{M-1}(t) + 2hg_4(t) \\ &+ \frac{h^3}{3D} (g_4'(t) - f_x(u_M(t), b, t) \\ &- f_u(u_M(t), b, t) \cdot g_4(t)). \end{aligned} \quad (19)$$

Now the matrix A involves t and U , and the first and last rows are modified as

$$\begin{aligned} A_{0,0\dots M} &= \left(\frac{10}{12} + \frac{h^3}{36D} (f_{xu}(u_0, a, t) \right. \\ &\quad \left. + f_{uu}(u_0, a, t) \cdot g_3(t)), \frac{1}{6}, \dots, 0 \right), \\ A_{M,0\dots M} &= \left(0, \dots, \frac{1}{6}, \frac{10}{12} - \frac{h^3}{36D} (f_{xu}(u_M, b, t) \right. \\ &\quad \left. + f_{uu}(u_M, b, t) \cdot g_4(t)) \right). \end{aligned} \quad (20)$$

Consequently the first and last components of F are

$$\begin{aligned} F_0 &= \frac{D}{h^2} (2u_1 - 2u_0 - 2hg_3) \\ &+ \frac{1}{12} (10f(u_0, a, t) + f(u_1, a + h, t)) \\ &- \frac{h}{3} (g_3' - f_x(u_0, a, t) - f_u(u_0, a, t) \cdot g_3) + \frac{h}{6} g_3' \\ &+ \frac{h^3}{36D} (g_3'' - f_{xt}(u_0, a, t) - f_u(u_0, a, t) \cdot g_3' \\ &\quad - f_{ut}(u_0, a, t) \cdot g_3) \\ &+ \frac{1}{12} f \left(u_1 - 2hg_3 - \frac{h^3}{3D} \right. \\ &\quad \left. \times (g_3' - f_x(u_0, a, t) - f_u(u_0, a, t) \cdot g_3), \right. \\ &\quad \left. a - h, t \right), \end{aligned}$$

$$\begin{aligned} F_M &= \frac{2D}{h^2} (u_{M-1} - u_M + hg_4) \\ &+ \frac{1}{12} (10f(u_M, b, t) + f(u_{M-1}, b - h, t)) \end{aligned}$$

$$\begin{aligned} &- \frac{h}{3} (g_4' - f_x(u_M, b, t) - f_u(u_M, b, t) \cdot g_4) - \frac{h}{6} g_4' \\ &- \frac{h^3}{36D} (g_4'' - f_{xt}(u_M, b, t) - f_u(u_M, b, t) \cdot g_4' \\ &\quad - f_{ut}(u_M, b, t) \cdot g_4) \\ &+ \frac{1}{12} f \left(u_{M-1} + 2hg_4 + \frac{h^3}{3D} \right. \\ &\quad \left. \times (g_4' - f_x(u_M, b, t) - f_u(u_M, b, t) \cdot g_4), \right. \\ &\quad \left. b + h, t \right). \end{aligned} \quad (21)$$

Finally, the ODE system is written in the form of $A(t, U)U' = F(U, t)$. Apparently, the matrix A preserves tridiagonal structure, but it depends on t and U ; hence the development of Rosenbrock method becomes difficult. Fortunately, we notice that t and U are involved in two entries: $A_{0,0}$ and $A_{M,M}$ only. Further investigation shows that the extra terms in (20) are

$$\frac{h^3}{36D} (f_{xu}(u_0, a, t) + f_{uu}(u_0, a, t) \cdot g_1(t)), \quad (22)$$

$$- \frac{h^3}{36D} (f_{xu}(u_M, b, t) + f_{uu}(u_M, b, t) \cdot g_4(t)), \quad (23)$$

respectively. We can eliminate these two extra terms by incorporating them into vector F ; therefore F_0 and F_M are modified as

$$\begin{aligned} \bar{F}_0 &= F_0 \\ &- \frac{h^3}{36D} [(f_{xu}(u_0, a, t) + f_{uu}(u_0, a, t) \cdot g_3(t))] \cdot u_0'(t), \end{aligned} \quad (24)$$

$$\begin{aligned} \bar{F}_M &= F_M \\ &+ \frac{h^3}{36D} [(f_{xu}(u_M, b, t) + f_{uu}(u_M, b, t) \cdot g_4(t))] \\ &\cdot u_M'(t). \end{aligned} \quad (25)$$

Using (1), we have

$$\begin{aligned} u_0'(t) &= \frac{2D}{h^2} (u_1(t) - u_0(t) - hg_3(t)) \\ &+ f(u_0(t), a, t) + O(h), \end{aligned} \quad (26)$$

$$\begin{aligned} u_M'(t) &= \frac{2D}{h^2} (u_{M-1}(t) - u_M(t) + hg_4(t)) \\ &+ f(u_M(t), b, t) + O(h). \end{aligned} \quad (27)$$

Inserting (26) into (24) and then ignoring the fourth-order error term $O(h^4)$, we obtain

$$\begin{aligned} \bar{F}_0 = F_0 - \frac{h^3}{36D} & [(f_{xu}(u_0, a, t) + f_{uu}(u_0, a, t) \cdot g_3(t))] \\ & \times \left(\frac{2D}{h^2} (u_1 - u_0 - hg_3(t)) + f(u_0, a, t) \right). \end{aligned} \tag{28}$$

Similarly, inserting (27) into (25) and then ignoring the fourth-order error term, we obtain

$$\begin{aligned} \bar{F}_M = F_M - \frac{h^3}{36D} & [(f_{xu}(u_M, b, t) + f_{uu}(u_M, b, t) \cdot g_4(t))] \\ & \times \left(\frac{2D}{h^2} (u_{M-1}(t) - u_M(t) + hg_4(t)) \right. \\ & \left. + f(u_M(t), b, t) \right). \end{aligned} \tag{29}$$

We then obtain the closed ODE system $AU'(t) = F(U, t)$, where the vector-valued function is given as $F = (\bar{F}_0, F_1, \dots, F_{M-1}, \bar{F}_M)$, and A is a nonsingular $(M+1) \times (M+1)$ constant matrix given as

$$A = \begin{pmatrix} \frac{5}{6} & \frac{1}{12} & 0 & 0 & \dots & 0 & 0 & 0 \\ \frac{1}{12} & \frac{5}{6} & \frac{1}{12} & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & \frac{1}{12} & \frac{5}{6} & \frac{1}{12} \\ 0 & 0 & 0 & 0 & \dots & \dots & \frac{1}{12} & \frac{5}{6} \end{pmatrix}. \tag{30}$$

If the Robin boundary condition is specified, a similar numerical technique can be used to derive the semidiscrete ODE system.

Here we mention, without theoretical proof, that the resulting semidiscrete ODE system (9) is a fourth-order accurate approximation to the original semilinear parabolic partial differential equation given in (1), supplemented with either Dirichlet or Neumann boundary conditions. Interested readers can find a similar theorem and proof in [11].

3. Fourth-Order Strongly A-Stable Rosenbrock Method

Various numerical methods can be used to solve the ODE system in (11). However, due to the stiffness of the problem, only stiffly stable methods are applicable; thus the choices are limited to the subclass of implicit methods such as implicit linear multistep methods and implicit Runge-Kutta methods. It is known that A-stability is necessary for stiff problem, and in general strong A-stable or even L-stable methods are preferred. The A-stability was firstly introduced and defined by Dahlquist [12] as the following.

Definition 1. A numerical method is called A-stable if there is no restriction on the step size, when it is applied to solve the test equation $y' = \lambda y$, where $\text{Re}(\lambda) < 0$.

For a single-step method such as Runge-Kutta method that can be written as $y_{n+1} = R(y_n)$, the A-stability is equivalent to the condition $|R(z)| \leq 1$ for any $z \in \mathbb{C}^{-1}$, where $R(z)$ is called the stability function of the method. Although successfully used in various applications, an A-stable linear multistep method has the highest order of 2. In fact, the second-order A-stable linear multistep method with optimal error constant is the Trapezium rule [12]. Further, Gourlay [13] pointed out that an A-stable method is necessary but not sufficient for very stiff system as it has the incorrect damping rate. For example, the widely used Trapezium rule has stability function $R(z) = (1+z)/(1-z)$ satisfying $|R(z)| < 1$ for any $z \in \mathbb{C}^{-1}$, but its damping rate converges to -1 when $z \rightarrow -\infty$. To overcome this difficulty, strong A-stability was introduced.

Definition 2. A numerical method is called strongly A-stable if it is A-stable and $|R(-\infty)| < 1$.

It has been shown that a numerical method with strong A-stability is effective in damping numerical oscillations for highly stiff system. For more details about the description and comparison of A-stability and strong A-stability, the readers are referred to [14].

Implicit Runge-Kutta method is usually unconditionally stable but suffers from the issue of high computational complexity, especially for nonlinear ODE system. For example, during each time step, an algebraic system with $s \times M$ unknown variables needs to be solved, if an s -stage implicit Runge-Kutta method is used to solve an ODE system with M equations. Therefore, fully implicit Runge-Kutta methods are too computationally expensive to be useful for large-scale problems. In the past several decades, efforts have been made to reduce the computational cost, which results in various modified implicit Runge-Kutta methods, such as diagonally implicit Runge-Kutta method, singly diagonally implicit Runge-Kutta method, explicit-implicit Runge-Kutta method, to name a few. For more details of these methods, the reader is referred to [15–19].

To completely avoid solving a nonlinear algebraic system, Rosenbrock method, which is a special class of Runge-Kutta method, had been proposed. Since the spatial discretization is fourth-order, our aim herein is to develop a strongly A-stable fourth-order Rosenbrock method for solving the ODE system (11), so the new algorithm is fourth-order accurate in both temporal and spatial dimensions.

3.1. Rosenbrock Method for Scalar Equation. We first derive the Rosenbrock method based on an autonomous scalar equation $y' = f(y)$, for which the initial condition is given as $y(t_0) = y_0$. Nonautonomous equations can be converted to autonomous form by adding an extra equation to the system. Some previous research [20] suggested that it is unlikely to find a 3-stage fourth-order Rosenbrock method with strong A-stability or L-stability, so herein we focus on

the development of a 4-stage Rosenbrock method. Suppose the numerical solution at time t_n is known as y_n , the 4-stage Rosenbrock method calculates the numerical solution at t_{n+1} as

$$y_{n+1} = y_n + b_1 k_1 + b_2 k_2 + b_3 k_3 + b_4 k_4, \quad (31)$$

$$k_i = \Delta t f \left(y_n + \sum_{j=1}^{i-1} \alpha_{ij} k_j \right) + \Delta t J(y_n) \sum_{j=1}^i \gamma_{ij} k_j, \quad (32)$$

$$i = 1, 2, 3, 4,$$

where b_i , α_{ij} , and γ_{ij} are coefficients to be determined and $J(y_n) = f_y(y_n)$.

To extend the above algorithm to the nonautonomous problems $y' = f(y, t)$, we first convert it to autonomous form by adding a new equation $t' = 1$ and then apply the algorithm (31) to the augmented system. Note that the components corresponding to the last variable can be computed explicitly; thus we can derive the modified algorithm as the following:

$$y_{n+1} = y_n + b_1 k_1 + b_2 k_2 + b_3 k_3 + b_4 k_4,$$

$$k_i = \Delta t f \left(y_n + \sum_{j=1}^{i-1} \alpha_{ij} k_j, t_n + \alpha_i \Delta t \right) + \gamma_i \Delta t^2 f_t(y_n, t_n) + \Delta t f_y(y_n, t_n) \sum_{j=1}^i \gamma_{ij} k_j,$$

$$i = 1, 2, 3, 4,$$

where

$$\alpha_i = \sum_{j=1}^{i-1} \alpha_{ij}, \quad \gamma_i = \sum_{j=1}^i \gamma_{ij}. \quad (34)$$

A Rosenbrock method of order p is obtained through choosing coefficients in (31)-(32) so that the local error satisfies $y(t_n + \Delta t) - y_{n+1} = O(\Delta t^{p+1})$. This can be done either by solving the so-called *Butcher Series* [14] or by straightforward differentiation. Here we derive the order conditions for the fourth-order Rosenbrock method in a different way using Taylor series. First, both $y(t_n + \Delta t)$ and k_i are expanded as Taylor series so that the difference $y(t_n + \Delta t) - y_{n+1}$ can be expressed as a Taylor series and its coefficients of the terms up to $O(\Delta t^4)$ are set to 0, which results in a set of equations involving these coefficients.

Given $y(t_n) = y_n$, the Taylor series of $y(t_n + \Delta t)$ at t_n is expanded as

$$y(t_n + \Delta t) = y_n + \Delta t f_n + \frac{\Delta t^2}{2} J_n f_n + \frac{\Delta t^3}{6} (J_n^2 f_n + J_n' f_n^2) + \frac{\Delta t^4}{24} (J_n'' f_n^3 + 2J_n' J_n f_n^2 + 2J_n J_n' f_n^2 + J_n^3 f_n) + O(\Delta t^5), \quad (35)$$

where $f_n = f(y_n)$, $J_n = (\partial f / \partial y)(y_n)$, $J_n' = (\partial^2 f / \partial y^2)(y_n)$, and $J_n'' = (\partial^3 f / \partial y^3)(y_n)$. For the sake of simplicity, let $\beta_{ij} = \alpha_{ij} + \gamma_{ij}$ and $\alpha_{ii} = 0$.

Letting $i = 1$ in (32) we have

$$k_1 = \Delta t (1 - \Delta t \gamma_{11} J_n)^{-1} f_n = \Delta t f_n + \Delta t^2 \gamma_{11} J_n f_n + \Delta t^3 \gamma_{11}^2 J_n^2 f_n + \Delta t^4 \gamma_{11}^3 J_n^3 f_n + O(\Delta t^5).$$

Similarly, letting $i = 2$ in (32) we have

$$k_2 = \Delta t (1 - \Delta t \gamma_{22} J_n)^{-1} (f(y_n + \alpha_{21} k_1) + \gamma_{21} J_n k_1). \quad (37)$$

Combining it with the Taylor series of k_1 in (36), we obtain

$$k_2 = \Delta t f_n + \Delta t^2 (\alpha_{21} + \gamma_{21} + \gamma_{22}) J_n f_n + \Delta t^3 [(\gamma_{11} + \gamma_{22})(\alpha_{21} + \gamma_{21}) + \gamma_{22}^2] J_n^2 f_n + \frac{\Delta t^3}{2} \alpha_{21}^2 J_n' f_n^2 + \Delta t^4 [(\gamma_{11} \gamma_{22} + \gamma_{11}^2 + \gamma_{22}^2)(\alpha_{21} + \gamma_{21}) + \gamma_{22}^3] J_n^3 f_n + \Delta t^4 \gamma_{11} \alpha_{21}^2 J_n' J_n f_n^2 + \frac{\Delta t^4}{2} \gamma_{22} \alpha_{21}^2 J_n J_n' f_n^2 + \frac{\Delta t^4}{6} \alpha_{21}^3 J_n'' f_n^3 + O(\Delta t^5). \quad (38)$$

Following the same way, we have

$$k_3 = \Delta t f_n + \Delta t^2 (\beta_{31} + \beta_{32} + \beta_{33}) J_n f_n + \Delta t^3 \left[\sum_{j=1}^3 \left((\alpha_{3j} + \gamma_{3j}) \cdot \sum_{l=1}^j (\alpha_{jl} + \gamma_{jl}) \right) \right] J_n^2 f_n + \frac{\Delta t^3}{2} \alpha_{31}^2 J_n' f_n^2 + \Delta t^4 \left[\sum_{i=1}^3 \left((\alpha_{3i} + \gamma_{3i}) \cdot \sum_{j=1}^i \left((\alpha_{ij} + \gamma_{ij}) \cdot \sum_{l=1}^j (\alpha_{jl} + \gamma_{jl}) \right) \right) \right] J_n^3 f_n + \frac{\Delta t^4}{2} [\alpha_2^2 \beta_{32} + \gamma_{33} \alpha_3^2] J_n J_n' f_n^2 + \Delta t^4 [\alpha_3 ((\beta_{21} + \gamma_{22}) \alpha_{32} + \gamma_{11} \alpha_{31})] J_n' J_n f_n^2 + \frac{\Delta t^4}{6} (\alpha_{31} + \alpha_{32})^3 J_n'' f_n^3 + O(\Delta t^5),$$

$$\begin{aligned}
 k_4 &= \Delta t f_n + \Delta t^2 (\beta_{41} + \beta_{42} + \beta_{42} + \gamma_{44}) J_n f_n \\
 &+ \Delta t^3 \left[\sum_{j=1}^4 \left((\alpha_{4j} + \gamma_{4j}) \cdot \sum_{l=1}^j (\alpha_{jl} + \gamma_{jl}) \right) \right] J_n^2 f_n \\
 &+ \frac{\Delta t^3}{2} \alpha_4^2 J_n' f_n^2 \\
 &+ \Delta t^4 \left[\sum_{i=1}^4 \left((\alpha_{4i} + \gamma_{4i}) \cdot \sum_{j=1}^i \left((\alpha_{ij} + \gamma_{ij}) \right. \right. \right. \\
 &\qquad \qquad \qquad \left. \left. \left. \cdot \sum_{l=1}^j (\alpha_{jl} + \gamma_{jl}) \right) \right) \right] J_n^3 f_n \\
 &+ \Delta t^4 [\alpha_4 ((\gamma_{33} + \beta_{32} + \beta_{31}) \alpha_{43} \\
 &\qquad \qquad \qquad + (\gamma_{22} + \beta_{21}) \alpha_{42} + \gamma_{11} \alpha_{41})] J_n' J_n f_n^2 \\
 &+ \frac{\Delta t^4}{2} [\gamma_{44} \alpha_4^2 + \beta_{43} \alpha_3^2 + \beta_{42} \alpha_2^2] J_n J_n' f_n^2 \\
 &+ \frac{\Delta t^4}{6} (\alpha_{41} + \alpha_{42} + \alpha_{43})^3 J_n'' f_n^3 + O(\Delta t^5).
 \end{aligned} \tag{39}$$

Inserting the four Taylor series into (35) and matching the coefficients of Δt^p for $p = 0, 1, 2, 3, 4$ on both sides of (31), we obtain the following order conditions:

$$1 = \sum_{i=1}^4 b_i, \tag{40}$$

$$\frac{1}{2} = \sum_{i=1}^4 b_i \cdot \left(\gamma_{ii} + \sum_{j=1}^{i-1} \beta_{ij} \right), \tag{41}$$

$$\frac{1}{6} = \sum_{i=1}^4 b_i \cdot \left(\sum_{j=1}^i \beta_{ij} \cdot \sum_{l=1}^j \beta_{jl} \right), \tag{42}$$

$$\frac{1}{3} = \sum_{i=1}^4 b_i \cdot \left(\sum_{j=1}^{i-1} \alpha_{ij} \right)^2, \tag{43}$$

$$\frac{1}{4} = \sum_{i=1}^4 b_i \cdot \left(\sum_{j=1}^{i-1} \alpha_{ij} \right)^3, \tag{44}$$

$$\frac{1}{24} = \sum_{i=1}^4 b_i \cdot \left(\alpha_i \sum_{j=1}^{i-1} \alpha_{ij} \cdot \left(\gamma_{jj} + \sum_{l=1}^{j-1} \beta_{jl} \right) \right), \tag{45}$$

$$\frac{1}{12} = \sum_{i=1}^4 b_i \cdot \left(\sum_{j=1}^i \beta_{ij} \cdot \left(\sum_{l=1}^{j-1} \alpha_{jl} \right)^2 \right), \tag{46}$$

$$\frac{1}{24} = \sum_{i=1}^4 b_i \cdot \left(\sum_{j=1}^i \left(\beta_{ij} \cdot \sum_{l=1}^j \left(\beta_{jl} \cdot \sum_{k=1}^l \beta_{lk} \right) \right) \right). \tag{47}$$

Note that the set of order conditions obtained by using Butcher series [14, page 108] is a special case here when $\gamma_{ii} = \gamma$.

Also it is worthy to point out that if $f(y)$ is a scalar function of a single variable, $J_n J_n' = J_n' J_n$, thus conditions (45) and (46) can be combined to one single condition. However here since the method is developed for ODE system, both conditions should be satisfied individually.

Apparently there are 12 degrees of freedom to determine the method defined in (32) since there are 20 parameters while only 8 constraints are given by (40)–(47). To simplify the procedure of determining these parameters and reduce the number of matrix inversion, we assume that $\beta_{ii} = \gamma$, for $i = 1, 2, 3, 4$, so that only one matrix inversion is required to solve all k_i during each time step. After this simplification, there are 17 parameters left, and the order conditions are simplified as well.

It is known that the Rosenbrock method suffers from order reduction when applied to nonlinear parabolic partial differential equations; see [20, 21]. In order to avoid such reduction of accuracy, the following two extra order conditions, simplified after taking into account the previous order conditions, should be satisfied [22]:

$$b_4 \beta_{43} \beta_{32} \alpha_2^2 = -2\gamma^4 + 4\gamma^3 - \frac{5}{3}\gamma^2 + \frac{\gamma}{6}, \tag{48}$$

$$b_4 \beta_{43} \beta_{32} \alpha_2^2 = -\frac{8\gamma^4}{3} + 5\gamma^3 - 2\gamma^2 + \frac{7\gamma}{36}, \tag{49}$$

which imply

$$\gamma^3 - \frac{3}{2}\gamma^2 + \frac{\gamma}{2} - \frac{1}{24} = 0. \tag{50}$$

Solving (50) results in three distinct real roots, while stability analysis shows that only one ($\gamma = 1.06857902130162885$) can ensure A-stability for the Rosenbrock method. More details regarding stability will be provided later.

To perform step size control and error estimation, we need a third-order embedded formula [23] defined as $y_{n+1} = y_n + \sum_{j=1}^s \tilde{b}_j k_j$ which uses the same values of k_j 's as the Rosenbrock method in (31) but coefficients of k_j 's are different. To ensure the existence of such embedded formula, we need $\text{Det}(\widehat{A}) = 0$, where \widehat{A} is a 4×4 square matrix defined as

$$\widehat{A} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ \beta_1 & \beta_2 & \beta_3 & \beta_4 \\ \beta_1^2 & \sum_{i=1}^2 \beta_{2i} \cdot \beta_i & \sum_{i=1}^3 \beta_{3i} \cdot \beta_i & \sum_{i=1}^4 \beta_{4i} \cdot \beta_i \\ 0 & \alpha_2^2 & \alpha_3^2 & \alpha_4^2 \end{pmatrix} \tag{51}$$

and $\beta_i = \sum_{j=1}^i \beta_{ij}$.

Taking into account the order conditions, we obtain the simplified condition as $\bar{\beta}_3^2 \alpha_2^2 \beta_{43} = 0$, which unfortunately contradicts the order condition given in (47). Thus one can either sacrifices efficiency by relaxing the condition $\gamma_{ii} = \gamma$ for the existence of a third-order embedded formula or simply skip the condition and then ignore step size control and error estimation. In this work, we are interested in the

TABLE 1: Coefficients of the 4th-order Rosenbrock method.

$b_1 = 0.4074074074074$	$b_2 = -0.2568608534470$	$b_3 = 0.2$
$b_4 = 0.6494534460396$	$\alpha_{21} = 0.75$	$\alpha_{31} = 0.75$
$\alpha_{32} = 0$	$\alpha_{41} = 2.9193596398302$	$\alpha_{42} = 0.4$
$\alpha_{43} = -2.5693596398302$	$\gamma_{21} = -0.75$	$\gamma_{31} = -1.3152686912402$
$\gamma_{32} = 0.75$	$\gamma_{41} = -2.8738466294648$	$\gamma_{42} = -3.3778743470341$
$\gamma_{43} = 4.5693596398302$	$\gamma = 1.068579021301629$	

efficiency and order reduction of the Rosenbrock method, so the condition for step size control is bypassed.

Upon the determination of γ , there are 17 free parameters remaining while the order conditions are simplified as the following:

$$b_1 + b_2 + b_3 + b_4 = 1, \tag{52}$$

$$b_2\bar{\beta}_2 + b_3\bar{\beta}_3 + b_4\bar{\beta}_4 = \frac{1}{2} - \gamma, \tag{53}$$

$$b_3\beta_{32}\bar{\beta}_2 + b_4(\beta_{42}\bar{\beta}_2 + \beta_{43}\bar{\beta}_3) = \frac{1}{6} - \gamma + \gamma^2, \tag{54}$$

$$b_2\alpha_{21}^2 + b_3\alpha_3^2 + b_4\alpha_4^2 = \frac{1}{3}, \tag{55}$$

$$b_2\alpha_{21}^3 + b_3\alpha_3^3 + b_4\alpha_4^3 = \frac{1}{4}, \tag{56}$$

$$b_3\alpha_3\alpha_{32}\bar{\beta}_2 + b_4\alpha_4(\alpha_{42}\bar{\beta}_2 + \alpha_{43}\bar{\beta}_3) = \frac{1}{8} - \frac{\gamma}{3}, \tag{57}$$

$$b_3\beta_{32}\alpha_2^2 + b_4(\beta_{42}\alpha_2^2 + \beta_{43}\alpha_3^2) = \frac{1}{12} - \frac{\gamma}{3}, \tag{58}$$

$$b_4\beta_{43}\beta_{32}\bar{\beta}_2 = \frac{1}{24} - \frac{\gamma}{2} + \frac{3}{2}\gamma^2 - \gamma^3, \tag{59}$$

which are equivalent to those given in [14, page 108], where $\bar{\beta}_i = \sum_{j=1}^{i-1} \beta_{ij}$.

Since γ is a root of (50), $b_4\beta_{43}\beta_{32}\bar{\beta}_2 = 0$, but $b_4\beta_{43}\beta_{32}\alpha_2^2 \neq 0$, thus $\beta_{21} = 0$. Then the order conditions in (52)–(58) can be further simplified.

We now choose the coefficients α_{ij} with the purpose of reducing the number of functional evaluations. Particularly, letting $\alpha_2 = \alpha_3 = \alpha_4$, (55)–(56) imply that $\alpha_{21} = \alpha_2 = 3/4$, which further implies $b_2 + b_3 + b_4 = 16/27$, so $b_1 = 11/27$. Letting $\alpha_{31} = \alpha_{21}$ and $\alpha_{32} = 0$, we can reduce the number of functional evaluation by 1.

We choose β_{43} as a free parameter, so

$$b_4\bar{\beta}_3 = \frac{1 - 6\gamma + 6\gamma^2}{6\beta_{43}}, \tag{60}$$

$$\alpha_{43} = \frac{(1 - 8\gamma/3)\beta_{43}}{1 - 6\gamma + 6\gamma^2}.$$

By choosing α_{42} as a free parameter, we have

$$\alpha_{41} = \frac{3}{4} - \alpha_{42} - \frac{(1 - 8\gamma/3)\beta_{43}}{1 - 6\gamma + 6\gamma^2}. \tag{61}$$

Letting β_{32} be a free parameter, we obtain

$$b_4 = \frac{16(-2\gamma^4 + 4\gamma^3 - 5\gamma^2/3 + \gamma/6)}{9\beta_{43}\beta_{32}}, \tag{62}$$

$$\beta_{31} = \frac{1 - 6\gamma + 6\gamma^2}{6\beta_{43}b_4} - \beta_{32}.$$

Finally, choose b_3 as free parameter, so we have $b_2 = 16/27 - b_3 - b_4$ and

$$\beta_{42} = \frac{(4 - 16\gamma - 27b_3\beta_{32})}{(27b_4)} - \beta_{43}, \tag{63}$$

$$\beta_{41} = \frac{(1/2 - \gamma - b_3(\beta_{31} + \beta_{32}))}{b_4} - \beta_{42} - \beta_{43}.$$

In summary, the coefficients of the Rosenbrock method are listed in Table 1.

Note that one can also use these free parameters to eliminate some fifth-order truncation error terms, so the truncation error constant can be further optimized.

3.2. Extend the Rosenbrock Method to ODE System.

We now extend the Rosenbrock algorithm to the ODE system $AU'(t) = F(U, t)$. Suppose that $U^n = (u_0^n, u_1^n, \dots, u_M^n)$ is the numerical solution of (10) at time t_n ; then U^{n+1} is calculated as

$$U^{n+1} = U^n + b_1K_1 + b_2K_2 + b_3K_3 + b_4K_4, \tag{64}$$

$$(A - \gamma\Delta tJ_n)K_i = F\left(U_n + \sum_{j=1}^{i-1} \alpha_{ij}K_j, t_n + \sigma_i\Delta t\right) + \mu_i\Delta t^2 \frac{\partial F}{\partial t}(U_n, t_n) + \Delta t\mathcal{F}_n\left(\sum_{j=1}^{i-1} \gamma_{ij}K_j\right), \tag{65}$$

$i = 1, 2, 3, 4,$

where $J_n = (\partial F/\partial U)(U_n, t_n)$ is the Jacobi matrix σ_i and μ_i are defined through (34).

Obviously, each stage of this method consists of solving a linear system of $M + 1$ equations. Since $\gamma_{11} = \dots = \gamma_{44} = \gamma$, only one LU -decomposition is required during each time step.

3.3. Stability of the Rosenbrock Method. We now study the stability of the Rosenbrock method defined by (31)-(32) with coefficients given in Table 1. Applying the Rosenbrock method to the linear test equation $y' = \lambda y, \lambda < 0$, and taking into account the order condition equation (49), we obtain the stability function of the Rosenbrock method as

$$R(z) = \left(1 + (1 - 4\gamma)z + \left(\frac{1}{2} - 4\gamma + 6\gamma^2 \right)z^2 - \left(\frac{5\gamma^2}{4} - \frac{5\gamma}{8} + \frac{1}{16} \right)z^4 \right) \times \left((1 - \gamma z^4)^{-1} \right), \tag{66}$$

where $z = \lambda \Delta t$.

In order for the Rosenbrock method to be strongly A-stable, we need

$$|R(z)| \leq 1, \quad \forall z \in \mathbb{C}^-, \tag{67}$$

$$\lim_{z \rightarrow -\infty} |R(z)| < 1. \tag{68}$$

Apparently,

$$\lim_{z \rightarrow -\infty} |R(z)| = \left| \frac{5\gamma^2/4 - 5\gamma/8 + 1/16}{\gamma^4} \right| = 0.6304149382 < 1, \tag{69}$$

so (68) is satisfied.

To prove (67), we can substitute z with $x + iy$ and then show that $|R(x + iy)| \leq 1$ for any $x < 0$. However such proof is technically tedious although possible. Here we use the Boundary-Locus method to draw the region of absolute stability of the method, which is shown in Figure 1. Obviously, the region of absolute stability contains the whole half-plane \mathbb{C}^- , so the Rosenbrock method is A-stable. Combined with (69), we can prove that the Rosenbrock method is strongly A-stable.

4. Numerical Examples and Discussions

We solve three numerical examples to demonstrate the accuracy and efficiency of the new algorithm. The first and second examples are reaction-diffusion equations supplemented with Dirichlet and Neumann boundary conditions, respectively. The two examples are solved by the new method to demonstrate the fourth-order accuracy in time and space. The third example is used to demonstrate that the new algorithm is free of order reduction and more efficient than several other existing fourth-order Rosenbrock methods. For comparison, the rate of convergence is compared with several other fourth-order methods which suffer from order reduction. In what follows, we use *HOC-ROSB4* to represent

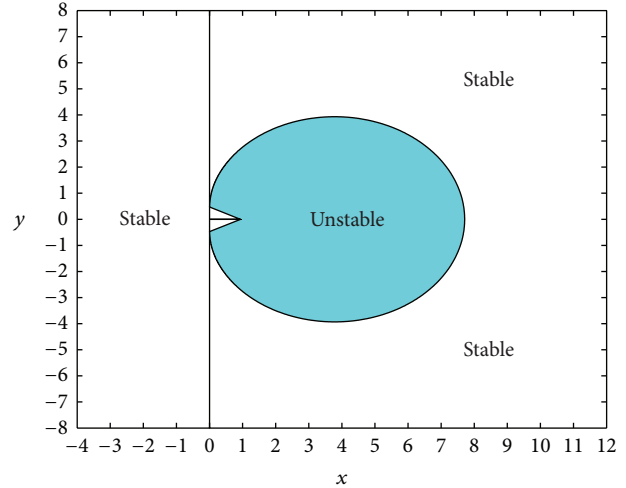


FIGURE 1: Absolute stable region of the Rosenbrock method.

the new algorithm developed in this paper and *HOC-GRK4A* to represent the fourth-order method that combines high-order compact finite difference scheme and fourth-order *GRK4A* [23]. *HOC-SHAMP* represents the algorithm that combines high-order compact finite difference scheme with the fourth-order Rosenbrock method *Shampine* [24]. *HOC-LSTAB* represents the algorithm that combines the high-order compact finite difference scheme in space and the L-stable fourth-order Rosenbrock method [14] in time. Finally, we use *HOC-VELD* to represent the algorithm that combines the high-order compact finite difference scheme in space with fourth-order Rosenbrock method proposed in [25].

4.1. Example 1. Consider the following reaction-diffusion equation:

$$\begin{aligned} u_t &= u_{xx} + \cos(u) - \cos(e^{-t} \cos(x)), \\ (x, t) &\in (0, 2) \times (0, 1], \\ u(x, 0) &= \cos(x), \quad x \in [0, 2], \\ u(0, t) &= e^{-t}, \quad u(2, t) = \cos(2)e^{-t}, \quad t \in [0, 1], \end{aligned} \tag{70}$$

for which the analytical solution is $u(x, t) = e^{-t} \cos(x)$.

We first show that the compact finite difference scheme for spatial discretization and the compact boundary condition treatment are fourth-order accurate. In order to do so, we fixed $\Delta t = 0.0001$ so the truncation error from time discretization is negligible. The results included in Table 2 clearly demonstrate the fourth-order convergence in space, as the maximal error reduced by a factor of 16 (roughly) when the grid size h is halved.

We then show the fourth-order accuracy in time by fixing $h = 0.001$. The data in Table 3 confirmed that the Rosenbrock method is fourth-order accurate in time, so the new algorithm is free of order reduction.

Since the method is fourth-order accurate in both temporal and spatial dimensions, we can write the leading term of

TABLE 2: Numerical results of *example 1* by HOC-ROSB4 with $\Delta t = 0.0001$.

h	1/10	1/20	1/40	1/80	1/160
$E(h)$	$7.38e - 08$	$4.62e - 09$	$2.89e - 10$	$1.80e - 11$	$1.06e - 12$
$E(h)/E(h/2)$	—	15.9562	16.0049	16.0607	17.0153
Convergence rate	—	3.9960	4.0004	4.0055	4.0888

TABLE 3: Numerical results of *example 1* by HOC-ROSB4 with $h = 0.001$.

Δt	1/10	1/20	1/40	1/80	1/160
$E(\Delta t)$	$9.03e - 06$	$6.16e - 07$	$3.96e - 08$	$2.45e - 09$	$1.49e - 10$
$E(\Delta t)/E(\Delta t/2)$	—	14.6679	15.5402	16.1607	16.4631
Convergence rate	—	3.8746	3.9579	4.0144	4.0412

TABLE 4: Numerical results of *example 1* by HOC-ROSB4 with $h/\Delta t = 3.2$.

h	1/10	1/20	1/40	1/80	1/160
Δt	1/32	1/64	1/128	1/256	1/512
$E(\Delta t, h)$	$5.94e - 08$	$4.09e - 09$	$2.73e - 10$	$1.78e - 11$	$1.15e - 12$
$E(\Delta t, h)/E(\Delta t/2, h/2)$	—	14.4970	14.9793	15.3189	15.4959
Convergence rate	—	3.8577	3.9049	3.9372	3.9538

the truncation error as $E(h, \Delta t) = C_1 h^4 + C_2 \Delta t^4$, where C_1 and C_2 are two constants. To obtain optimal performance, we can adjust the ratio $\Delta t/h$ to balance the two error terms. To do so, we adjust Δt and h so that the two error terms are balanced, which is utilized by letting $|C_2 \Delta t^4 / C_1 h^4| \approx 1$, from which we can estimate the optimal ratio as $(C_1/C_2)^{1/4}$. To estimate C_1 , we solve the example using small Δt and then $C_1 \approx E(h)/h^4$. Similarly, we can estimate C_2 using the same method.

Simple calculation based on numerical results from Tables 2 and 3 suggests that the optimal ratio for this example is $h/\Delta t \approx 3.2$. We then solve the reaction-diffusion equation using the optimal ratio to show that the new algorithm is fourth-order accurate in both temporal and spatial dimensions. The numerical results in Table 4 indicate that the new algorithm is fourth-order accurate in both temporal and spatial dimensions, as the maximal error is reduced by a factor of 16 (roughly) when Δt and h are halved simultaneously. We also notice that the algorithm is very efficient as the maximal error drops to the level of 10^{-12} when h and Δt are still reasonably large.

4.2. Example 2. We solve the following semilinear parabolic partial differential equation with the Neumann boundary conditions:

$$\begin{aligned} u_t &= 2u_{xx} + u + u^2 - e^{-2t} \cos^2(x), & (x, t) \in (0, 2) \times (0, 1], \\ u(x, 0) &= \cos(x), & x \in [0, 2], \\ u_x(0, t) &= 0, \quad u_x(2, t) = -\sin(2)e^{-t}, & t \in [0, 1], \end{aligned} \quad (71)$$

for which the analytical solution is $u(x, t) = e^{-t} \cos(x)$.

Notice that the boundary conditions are approximated by the compact fourth-order boundary scheme given in (18)-(19), so the compactness of the resulting linear system is

preserved; consequently, in each time step, only a tridiagonal system is solved for each k_j , so the solution procedure is very efficient. However we point out that some tedious work is needed to form the semidiscrete ODE system, for instance, the first and last rows of the matrix A in (20) and the first and last components of F in (28)-(29).

To obtain the optimal performance, we choose the optimal ratio as Δt as $h/\Delta t = 2.5$ to balance the two error terms. The data in Table 5 shows that the method is fourth-order accurate in both time and space, as the maximum error is reduced by a factor of 16 (roughly), when h and Δt are halved.

It is worthy to mention that one can also use any other one-sided formula to approximate the Neumann boundary condition, which apparently reduces the effort to derive the semidiscrete ODE system, but will destroy the tridiagonal structure of the matrix A and consequently affect the efficiency of the algorithm. Another side-effect of using one-sided approximation on the boundary is the stability issue that may arise because the modifications to matrix A may result in positive eigenvalues of A .

4.3. Example 3. In this example we compare the new algorithm with several other fourth-order Rosenbrock methods in terms of the rate of convergence in time and efficiency. The nonlinear parabolic partial differential equation to be solved is defined as

$$\begin{aligned} u_t &= u_{xx} + u^3 - e^{-3t} \cos^3(x), & (x, t) \in (0, 1) \times (0, 1] \\ u(x, 0) &= \cos(x), & x \in [0, 1], \\ u(0, t) &= e^{-t}, \quad u(1, t) = \cos(1)e^{-t}, & t \in [0, 1], \end{aligned} \quad (72)$$

for which the exact solution is $u(x, t) = e^{-t} \cos(x)$.

The data in Table 6 shows that HOC-ROSB4 is fourth-order accurate and thus is free of order reduction, while the

TABLE 5: Numerical results of *example 2* by *HOC-ROSB4* with $h/\Delta t = 2.5$.

h	1/10	1/20	1/40	1/80	1/160
Δt	1/25	1/50	1/100	1/200	1/400
$E(h, \Delta t)$	$6.27e - 06$	$4.40e - 07$	$2.95e - 08$	$1.96e - 09$	$1.31e - 10$
$E(h, \Delta t)/E(h/2, \Delta t/2)$	—	14.2495	14.9178	15.0510	15.0135
Convergence rate	—	3.8328	3.8990	3.9118	3.9082

TABLE 6: Comparison of the rate of convergence among the five fourth-order Rosenbrock methods for *example 3* with $h = 0.001$. $E(\Delta t)$ represents the maximum error of the numerical solution obtained by using Δt .

	Δt	1/10	1/20	1/40	1/80
<i>HOC-ROSB4</i>	$E(\Delta t)$	$9.59e - 06$	$6.94e - 0$	$4.58e - 08$	$2.88e - 09$
	Rate of Conv.	—	3.78757	3.9221	3.9896
<i>HOC-GRK4A</i>	$E(\Delta t)$	$5.88e - 06$	$6.16e - 07$	$6.83e - 08$	$7.56e - 09$
	Rate of Conv.	—	3.2552	3.1728	3.1757
<i>HOC-LSTAB</i>	$E(\Delta t)$	$3.96e - 06$	$4.56e - 07$	$5.23e - 08$	$6.19e - 09$
	Rate of Conv.	—	3.1205	3.1223	3.0795
<i>HOC-VELD</i>	$E(\Delta t)$	$8.41e - 06$	$8.77e - 07$	$9.77e - 08$	$1.14e - 08$
	Rate of Conv.	—	3.2613	3.1653	3.0996
<i>HOC-SHAMP</i>	$E(\Delta t)$	$6.35e - 06$	$6.34e - 07$	$6.92e - 08$	$8.01e - 09$
	Rate of Conv.	—	3.3230	3.1965	3.1099

TABLE 7: Comparison of the efficiency among the five fourth-order Rosenbrock methods for *example 3*.

Method	$(\Delta t, h)$	Max. error	CPU time (seconds)
<i>HOC-ROSB4</i>	(1/180, 1/40)	$7.72e - 011$	0.107
<i>HOC-GRK4A</i>	(1/256, 1/40)	$7.28e - 011$	0.134
<i>HOC-LSTAB</i>	(1/360, 1/40)	$7.83e - 011$	0.174
<i>HOC-VELD</i>	(1/400, 1/40)	$7.77e - 011$	0.188
<i>HOC-SHAMP</i>	(1/360, 1/40)	$7.60e - 011$	0.170

other four fourth-order Rosenbrock methods, which obviously suffer from order reduction, show rates of convergence ranging from 3.0 to 3.25. Note that $h = 0.001$ is fixed and the same spatial discretization method is used for all five methods, so the demonstrated rate of convergence is in time. One can see that the *HOC-LSTAB* method has the most optimal error constant, while the *HOC-ROSB4* method has the largest error constant. However, due to the freedom in determining those coefficients in Rosenbrock method, better error constant can be accomplished by eliminating several fifth-order truncation error terms.

Finally, we show that *HOC-ROSB4* is the most efficient method. For consistency, we adjust Δt and h for each method to reach the same error level and record the average CPU time of 5 simulation runs. Since all of the five methods use the same spatial discretization, the same h is used for all methods; hence we adjust Δt only. The comparison result included in Table 7 clearly indicates that the new method is the most efficient one. The higher efficiency apparently is obtained from the fact that the new method is free of order reduction; hence large time step can be used to reach the same error level.

5. Conclusion

An efficient fourth-order numerical algorithm that combines the Padé approximation in space and fourth-order accurate Rosenbrock method in time is proposed in this paper. Our investigation shows that many widely used fourth-order Rosenbrock methods (A-stable or L-stable) suffer from order reduction when they are used to solve nonlinear parabolic partial differential equation. To avoid order reduction, extra order conditions are required, which are implemented in this paper to develop the new algorithm. Also, it has been shown [26] that the extra condition to resolve order reduction contradicts the L-stability of the Rosenbrock method, so there is no L-stable Rosenbrock method that is also free of order reduction. The new method can be used to solve nonlinear parabolic partial differential equation with all types of boundary conditions; however, for Neumann or Robin boundary condition, extra efforts are needed to form the semidiscrete ODE system. Two numerical examples are solved to demonstrate that the new method is fourth-order accurate in both time and space, while the third example shows that the new method is free of order reduction and

is very efficient. In the future, we plan to extend the new method to multidimensional problems supplemented with various types of boundary conditions.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work of was supported by the Natural Sciences & Engineering Research Council of Canada (NSERC) through the individual Discovery Grant program. The author gratefully acknowledges the financial support from NSERC.

References

- [1] Y. Adam, "Highly accurate compact implicit methods and boundary conditions," *Journal of Computational Physics*, vol. 24, no. 1, pp. 10–22, 1977.
- [2] A. R. Mitchell and D. F. Griffiths, *The Finite Difference Method in Partial Differential Equations*, John Wiley & Sons, New York, NY, USA, 1980.
- [3] J. I. Ramos, "Linearization methods for reaction-diffusion equations: multidimensional problems," *Applied Mathematics and Computation*, vol. 88, no. 2–3, pp. 225–254, 1997.
- [4] J. I. Ramos, "Implicit, compact, linearized θ -methods with factorization for multidimensional reaction-diffusion equations," *Applied Mathematics and Computation*, vol. 94, no. 1, pp. 17–43, 1998.
- [5] P. C. Chu and C. Fan, "A three-point combined compact difference scheme," *Journal of Computational Physics*, vol. 140, no. 2, pp. 370–399, 1998.
- [6] V. Luan and A. Ostermann, "Exponential Rosenbrock methods of order five—construction, analysis and numerical comparisons," *Journal of Computational and Applied Mathematics*, vol. 255, no. 1, pp. 417–431, 2014.
- [7] V. T. Luan and A. Ostermann, "Explicit exponential Runge-Kutta methods of high order for parabolic problems," *Journal of Computational and Applied Mathematics*, vol. 256, pp. 168–179, 2014.
- [8] A.-K. Kassam and L. N. Trefethen, "Fourth-order time-stepping for stiff PDEs," *SIAM Journal on Scientific Computing*, vol. 26, no. 4, pp. 1214–1233, 2005.
- [9] H. Rosenbrock, "Some general implicit processes for the numerical solution of differential equations," *The Computer Journal*, vol. 5, pp. 329–330, 1963.
- [10] C. F. Haines, "Implicit integration processes with error estimate for the numerical solution of differential equations," *The Computer Journal*, vol. 12, no. 2, pp. 183–188, 1969.
- [11] W. Liao and Y. Yan, "Singly diagonally implicit Runge-Kutta method for time-dependent reaction-diffusion equation," *Numerical Methods for Partial Differential Equations*, vol. 27, no. 6, pp. 1423–1441, 2011.
- [12] G. G. Dahlquist, "A special stability problem for linear multistep methods," *BIT Numerical Mathematics*, vol. 3, pp. 27–43, 1963.
- [13] A. R. Gourlay, "A note on trapezoidal methods for the solution of initial value problems," *Mathematics of Computation*, vol. 24, pp. 629–633, 1970.
- [14] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations, II, Stiff and Algebraic Problems*, Springer, Berlin, Germany, 2nd edition, 1996.
- [15] R. Alexander, "Diagonally implicit Runge-Kutta methods for stiff O.D.E.s," *SIAM Journal on Numerical Analysis*, vol. 14, no. 6, pp. 1006–1021, 1977.
- [16] K. Burrage, J. C. Butcher, and F. H. Chipman, "An implementation of singly-implicit Runge-Kutta methods," *BIT Numerical Mathematics*, vol. 20, no. 3, pp. 326–340, 1980.
- [17] H. Claus, "Singly-implicit Runge-Kutta methods for retarded and ordinary differential equations," *Computing*, vol. 43, no. 3, pp. 209–222, 1990.
- [18] W. Liniger and R. A. Willoughby, "Efficient integration methods for stiff systems of ordinary differential equations," *SIAM Journal on Numerical Analysis*, vol. 7, pp. 47–66, 1970.
- [19] J. G. Verwer, E. J. Spee, J. G. Blom, and W. Hundsdorfer, "A second-order Rosenbrock method applied to photochemical dispersion problems," *SIAM Journal on Scientific Computing*, vol. 20, no. 4, pp. 1456–1480, 1999.
- [20] J. Lang and J. Verwer, "ROS3P—an accurate third-order Rosenbrock solver designed for parabolic problems," *BIT Numerical Mathematics*, vol. 41, no. 4, pp. 731–738, 2001.
- [21] W. H. Hundsdorfer, "Stability and B -convergence of linearly implicit Runge-Kutta methods," *Numerische Mathematik*, vol. 50, no. 1, pp. 83–95, 1986.
- [22] C. Lubich and A. Ostermann, "Linearly implicit time discretization of non-linear parabolic equations," *IMA Journal of Numerical Analysis*, vol. 15, no. 4, pp. 555–583, 1995.
- [23] P. Kaps and P. Rentrop, "Generalized Runge-Kutta methods of order four with stepsize control for stiff ordinary differential equations," *Numerische Mathematik*, vol. 33, no. 1, pp. 55–68, 1979.
- [24] L. F. Shampine, "Implementation of Rosenbrock methods," *ACM Transactions on Mathematical Software*, vol. 8, no. 2, pp. 93–113, 1982.
- [25] M. van Veldhuizen, " D -stability and Kaps-Rentrop-methods," *Computing. Archives for Scientific Computing*, vol. 32, no. 3, pp. 229–237, 1984.
- [26] T. D. Bui, "On an L -stable method for stiff differential equations," *Information Processing Letters*, vol. 6, no. 5, pp. 158–161, 1977.