A Parallel Fourth Order Rosenbrock Method: Construction, Analysis and Numerical Comparison

R. Ponalagusamy · K. Ponnammal

Published online: 1 October 2014
© Springer India Pvt. Ltd. 2014

Abstract The present paper sheds some light on constructing a new fourth order, four-stage parallel Rosenbrock method (NPROS4) and investigating its convergence and stability properties in view of solving stiff ordinary differential equations. The expression for local truncation error (LTE) has been derived and the parameters are chosen to minimize the coefficients in the LTE term. It is pertinent to point out here that this method possesses more free parameters to improve the stability region. The present method can also be efficiently implemented in a parallel computer having four processors with less error than the existing methods of the same order. Numerical examples are given to illustrate the efficiency of the present developed method.

Keywords Parallel Rosenbrock methods · Stiff ordinary differential equations · LTE · Stability

Introduction

Many physical processes such as chemical kinetics (Chan et al. [1]) and air pollution modeling (Verwer et al. [2] and Sandu et al. [3]) are represented by an initial value problem (IVP) associated with a set of stiff ordinary differential system of equations (ODE's) of first order given as [4]

$$y'(x) = f(x, y(x)), \quad x \in [x_0, x_n] \quad (1)$$

with a initial condition for the initial value prescribed as

$$y_0 = y(x_0) \quad (2)$$

R. Ponalagusamy (✉)

Department of Mathematics, National Institute of Technology, Tiruchirappalli 620 015, India
e-mail: rpalagu@nitt.edu

K. Ponnammal

Department of Mathematics, Periyar E.V.R. College, Tiruchirappalli 620 023, India
e-mail: ponnammal_k@yahoo.co.in

If f does not depend explicitly on x , the system is said to be autonomous; otherwise, it is considered non-autonomous. We have to use iterative implicit methods to solve these physical problems, which sometimes lead to problem of convergence of iterative methods due to nonlinear implicit equations. Rosenbrock [5], first presented a class of methods to generalize the linearly implicit approach by not using the iterative Newton method, instead to derive stable formulas by working the Jacobian matrix directly into the integration formula. This idea has found widespread use called Rosenbrock methods (ROW; Hairer and Wanner [6]).

Tambue et al. [7] have shown that the exponential Rosenbrock–Euler method and Rosenbrock type method lead to efficient tools to effectively solve geothermal systems. Schippmann and Burchard [8] have focused their attention on investigating the computational efficiency of Rosenbrock in comparison with Runge–Kutta and Patankar methods. The numerical solution of a discontinuous differential system by a Rosenbrock method has been obtained by Berardi [9]. It is further suggested that this technique is also applicable to solve discontinuous singularly perturbed systems. Rang [10] has derived new order conditions to reduce order reduction and developed a new third order diagonally implicit Runge–Kutta methods and Rosenbrock–Wanner method. The new schemes are applied to two stiff problems, namely, the Prothero–Robinson example and the incompressible Navier–Stokes equations.

The non-autonomous form (1) can be converted into autonomous form by adding $x' = 1$. For autonomous systems, and an exact Jacobian, an s -stage ROW methods are defined as

$$\begin{aligned} k_i &= hf \left(y_n + \sum_{j=1}^{i-1} \alpha_{ij} k_j \right) + hJ \sum_{j=1}^i \gamma_{ij} k_j, \quad i = 1, \dots, s \\ y_{n+1} &= y_n + \sum_{j=1}^s b_j k_j \end{aligned} \quad (3)$$

where α_{ij} , γ_{ij} and b_j are the determining coefficients and $J = f'(y_0)$, $\gamma_{ii} = \gamma$.

Many methods of this type are derived by Calahan [11], Cash [12], Norsett and Wolfbrandt [13], Kaps and Rentrop [14], Kaps and Wanner [15], Shampine [16] and Kaps et al. [17] in view of assuming $\gamma_{ii} = \gamma$. Parallel machines are computers with more than one processor and this facility might help to speed up the computations in ODE's. Therefore numerical scientists are interested in developing Rosenbrock methods suitable to parallel computers in solving stiff ODE's. Voss [18, 19] proposed a parallel Rosenbrock method (PRM) of second and fourth order by assuming $\gamma_{ii} = \gamma_j$, $j = 1, 2$. To raise the efficiency of sequential Rosenbrock methods, Chen and Liu [20] presented a class of PRMs, which also avoids nonlinear systems and is more efficient than the sequential ROW methods mentioned above. Cao et al. [21] proposed modified PRMs (MPROW) of third order two stage methods and fourth order third stage methods.

In this paper, a new fourth order, four-stage PRM (NPROS4(4)) to solve stiff ODEs is derived and studied. This organization of this article is as follows. Section “Order Conditions”, is followed by section “New PRMs of Order 4(4) (NPROS4(4)) for $\gamma_{ii} \neq \gamma$ ” which presents construction of a new class of methods by exploiting more number of free parameters, and further explains the parallel implementation of the sequential method. In the section “Derivation of the LTE”, LTE expression is derived using the fifth-order conditions. Analysis of convergence and stability is described in section “Convergence and Stability of NPROS4(4)”, while numerical experiments are given in the section “Numerical Experimentation”. Further, we intend here to compare the present four stage, fourth order PRM (NPROS4(4)) with the existing three stage, fourth order modified PRM (MPROW) described in Cao et al. [21].

Order Conditions

Consider the IVP given in Eq. (1) where the mapping $f(y)$ is assumed to satisfy a Lipschitz condition (Jain et al. [22]) and has all continuous derivatives used later. The exact solution of Eq. (1) is denoted by $y(t)$, $x_0 \leq t \leq x_n$. To solve this, a class of sequential fourth order four stage Rosenbrock method is defined as,

$$(I - h\gamma_{ii}J)k_i = hf\left(y_n + \sum_{j=1}^{i-1} \alpha_{ij}k_j\right) + hJ \sum_{j=1}^{i-1} \gamma_{ij}k_j, \quad i = 1, 2, 3, 4$$

$$y_{n+1} = y_n + \sum_{j=1}^4 b_j k_j \quad (4)$$

where $h > 0$ is the integration stepsize, $t_n = x_0 + nh$, α_{ij} , γ_{ij} and b_j are the real coefficients, I denotes the identity matrix, J denotes the Jacobian matrix, y_n is an approximation to $y(t_n)$, and each k_i denotes an approximation to some piece of information about the exact solution $y(t)$. To obtain the coefficients of NPROS4(4), the Taylor series expansion can be used or the order tree theory (Butcher [23]) is adopted. According to Butcher [24, 25], trees play a central role in the theory of Runge–Kutta methods and they also have applications to more general methods, involving multiple values and multiple stages. Keeping this in view, we have given the trees and order conditions upto order 5 in Table 1. The abbreviation $\beta_{ij} = \alpha_{ij} + \gamma_{ij}$ has been followed. The novelty of the present paper is that we do not assume the condition $\gamma_{ii} = \gamma$.

Using the set of order conditions corresponding to trees of order up to 4 (see Table 1), we derive the new Rosenbrock method NPROS4(4). With the help of the order conditions of trees of order 5 (see Table 1), we obtain the LTE of the NPROS4(4).

The conditions used to construct the four-stage fourth order Rosenbrock methods are written as:

$$b_1 + b_2 + b_3 + b_4 = 1 \quad (5)$$

$$b_2\beta'_2 + b_3\beta'_3 + b_4\beta'_4 = \frac{1}{2} - (b_1\gamma_{11} + b_2\gamma_{22} + b_3\gamma_{33} + b_4\gamma_{44}) \quad (6)$$

$$b_2\alpha_2^2 + b_3\alpha_3^2 + b_4\alpha_4^2 = \frac{1}{3} \quad (7)$$

$$b_2\beta'_2\gamma_{11} + b_2\beta'_2\gamma_{22} + b_3\{\beta'_2\beta_{32} + \beta_{32}\gamma_{22} + \beta_{32}\gamma_{33} + \beta_{31}(\gamma_{11} + \gamma_{33})\} \\ + b_4\{\beta'_2\beta_{42} + \beta_{31}\beta_{43} + \beta_{32}\beta_{43} + \beta_{41}\gamma_{11} + \beta_{42}\gamma_{22} + \beta_{43}\gamma_{33} + \beta_{41}\gamma_{44} + \beta_{43}\gamma_{44}\} \\ = \frac{1}{6} - (b_1\gamma_{11}^2 + b_2\gamma_{22}^2 + b_3\gamma_{33}^2 + b_4\gamma_{44}^2) \quad (8)$$

$$b_2\alpha_2^3 + b_3\alpha_3^3 + b_4\alpha_4^3 = \frac{1}{4} \quad (9)$$

$$b_2\alpha_2^2\gamma_{11} + b_3\alpha_3\{\alpha_{31}\gamma_{11} + \alpha_{32}\gamma_{22} + \alpha_{32}\beta'_2\} + b_4\alpha_4\{\alpha_{41}\gamma_{11} + \alpha_{42}\gamma_{22} \\ + \alpha_{43}\gamma_{33} + \alpha_{42}\beta'_2 + \alpha_{43}\beta'_3\} = \frac{1}{8} \quad (10)$$

$$\alpha_2^2\{b_2\gamma_{22} + b_3\beta_{32} + b_4\beta_{42}\} + \alpha_2^2\{b_3\gamma_{33} + b_4\beta_{43}\} + b_4\alpha_4^2\gamma_{44} = \frac{1}{12} \quad (11)$$

$$b_2\gamma_{11}^2\beta'_2 + b_2\gamma_{11}\beta'_2\gamma_{22} + b_2\beta'_2\gamma_{22}^2 + b_3\gamma_{11}^2\beta_{31} + b_3\gamma_{11}\beta_{21}\beta_{32} \\ + b_3\beta_{21}\gamma_{22}\beta_{32} + b_3\gamma_{22}^2\beta_{32} + b_3\gamma_{11}\beta_{31}\gamma_{33} + b_3\beta_{21}\beta_{32}\gamma_{33} + b_3\gamma_{22}\beta_{32}\gamma_{33} \\ + b_3\beta_{31}\gamma_{33}^2 + b_3\beta_{32}\gamma_{33}^2 + b_4\gamma_{11}^2\beta_{41} + b_4\gamma_{11}\beta_{21}\beta_{42} + b_4\beta_{21}\gamma_{22}\beta_{42} + b_4\gamma_{22}^2\beta_{42}$$

$$\begin{aligned}
& + b_4 \gamma_{11} \beta_{31} \beta_{43} + b_4 \beta'_2 \beta_{32} \beta_{43} + b_4 \gamma_{22} \beta_{32} \beta_{43} + b_4 \beta_{31} \gamma_{33} \beta_{43} + b_4 \beta_{32} \gamma_{33} \beta_{43} \\
& + b_4 \gamma_{33}^2 \beta_{43} + b_4 \gamma_{11} \beta_{41} \gamma_{44} + b_4 \beta'_2 \beta_{42} \gamma_{44} + b_4 \gamma_{22} \beta_{42} \gamma_{44} + b_4 \beta_{31} \beta_{43} \gamma_{44} \\
& + b_4 \beta_{32} \beta_{43} \gamma_{44} + b_4 \gamma_{33} \beta_{43} \gamma_{44} + b_4 \beta_{41} \gamma_{44}^2 + b_4 \beta_{42} \gamma_{44}^2 + b_4 \beta_{43} \gamma_{44}^2 \\
& = \frac{1}{24} - b_1 \gamma_{11}^3 - b_2 \gamma_{22}^3 - b_3 \gamma_{33}^3 - b_4 \gamma_{44}^3
\end{aligned} \tag{12}$$

where

$$\alpha_i = \sum_{j=1}^{i-1} \alpha_{ij}, \quad \beta'_i = \sum_{j=1}^{i-1} \beta_{ij} \tag{13}$$

Internal Stability of the Method

Rosenbrock methods use s function evaluations at the points

$$u_i = y_n + \sum_{j=1}^{i-1} \alpha_{ij} k_j \tag{14}$$

to avoid function evaluations away from the solution, stability for these points is needed. For Eq. (14), we obtain $u_i = R_i(z)y_n$, where

$$\begin{aligned}
u_1 &= 1, \quad R_2 = 1 + \alpha_2 w, \quad R_3 = 1 + \alpha_3 w + \alpha_{32} \beta_2 w^2 \\
R_4 &= 1 + \alpha_4 w + (\alpha_{42} \beta_2 + \alpha_{43} \beta_3) w^2 + \alpha_{43} \beta_{32} \beta_2 w^3
\end{aligned}$$

and

$$w = \frac{z}{1 - \gamma_{ii} z}, \quad \alpha_i = \sum_{j=1}^{i-1} \alpha_{ij}, \quad \beta'_i = \sum_{j=1}^{i-1} \beta_{ij}$$

A method is said to be internally stable, if $R_i(z)$ are A-stable for $1 \leq i \leq s$ (Verwer [26]).

New PRMs of Order 4(4) (NPROS4(4)) for $\gamma_{ii} \neq \gamma$

The computation of the coefficients α_{ij} , γ_{ij} , β_{ij} and b_j satisfying (5)–(13) have been carried out as follows:

Step 1 we choose $\gamma_{ii} > 0$ with the following stability properties (Kaps et al. [17]):

- (i) Small values for the non-vanishing error terms of LTE described in section 4.
- (ii) Positive coefficients b_j .
- (iii) α_{ij} , β_{ij} are of small value in magnitude.
- (iv) Internal stability of the method (described in section 2.1).

Step 2 choose α_2 , α_3 , α_4 and b_1 , b_2 , b_3 , b_4 in such a way that the three conditions (5), (7) and (9) are satisfied.

Step 3 take β_{43} and β_{31} as free parameters and solve conditions (6), (8), (11) and (12) to find β'_2 , β'_3 , β'_4 and β_{42} .

Step 4 once the β'_i are known, one can easily find β_{32} , β_{41} by using (13).

Step 5 find α_{41} , α_{42} , α_{43} by using (10) and (13). We assume here α_{43} as a free parameter. The remaining free parameters can be chosen.

In particular, we follow steps 1–5 to select the parameters,

$$\begin{aligned}\alpha_{21} &= \frac{1}{2}, \quad \alpha_{31} = \frac{1}{2}, \quad \alpha_{32} = \frac{1}{2}, \quad \alpha_{41} = -0.178278, \quad \gamma_{11} = \frac{1}{6} \\ \gamma_{22} &= \frac{2}{6}, \quad \gamma_{33} = \frac{2}{6}, \quad \gamma_{44} = \frac{1}{6} \\ \alpha_{42} &= 0.219945, \quad \alpha_{43} = \frac{1}{3}, \quad \gamma_{21} = -0.12642566, \\ \gamma_{31} &= -0.5, \quad \gamma_{32} = -1.22102225482 \\ b_1 &= \frac{43}{758}, \quad b_2 = \frac{83}{192}, \quad b_3 = \frac{137}{768}, \quad b_4 = \frac{1}{3}\end{aligned}$$

Then the new fourth order four stage Rosenbrock method is given as follows:

$$\begin{aligned}\left(I - \frac{h}{6}J\right)k_1 &= hf(y_n) \\ \left(I - \frac{2h}{6}J\right)k_2 &= hf(y_n + (0.5)k_1) - hJ(0.12642566)k_1 \\ \left(I - \frac{2h}{6}J\right)k_3 &= hf(y_n + (0.5)k_1 + (0.5)k_2) + hJ(-0.5k_1 - 1.22102225482k_2) \\ \left(I - \frac{h}{6}J\right)k_4 &= hf\left(y_n - 0.178278k_1 + 0.219945k_2 - \frac{k_3}{3}\right) \\ &\quad + hJ(0.178278k_1 + 0.088797k_2 - 0.37395k_3) \\ y_{n+1} &= y_n + \frac{43k_1}{768} + \frac{83k_2}{192} + \frac{137k_3}{768} + \frac{k_4}{3}\end{aligned}\quad (15)$$

Parallel Implementation

The method described in Eq. (15) is rewritten as

$$\begin{aligned}(I - h\gamma_{11}J)k_{1,n} &= hf(y_n) \\ (I - h\gamma_{22}J)k_{2,n} &= hf(y_n + \alpha_{21}k_{1,n-1}) - hJ\gamma_{21}k_{1,n-1} \\ (I - \gamma_{33}J)k_{3,n} &= hf(y_n + \alpha_{31}k_{1,n-1} + \alpha_{32}k_{2,n-1}) + hJ(\gamma_{31}k_{1,n-1} + \gamma_{32}k_{2,n-1}) \\ (I - \gamma_{44}J)k_{4,n} &= hf(y_n + \alpha_{41}k_{1,n-1} + \alpha_{42}k_{2,n-1} + \alpha_{43}k_{3,n-1}) \\ &\quad + hJ(\gamma_{41}k_{1,n-1} + \gamma_{42}k_{2,n-1} + \gamma_{43}k_{3,n-1}) \\ y_{n+1} &= y_n + b_1k_{1,n} + b_2k_{2,n} + b_3k_{3,n} + b_4k_{4,n}\end{aligned}\quad (16)$$

where

$$\begin{aligned}\alpha_{21} &= \frac{1}{2}, \quad \alpha_{31} = \frac{1}{2}, \quad \alpha_{32} = \frac{1}{2}, \quad \alpha_{41} = -0.178278, \quad \gamma_{11} = \frac{1}{6} \\ \gamma_{22} &= \frac{2}{6}, \quad \gamma_{33} = \frac{2}{6}, \quad \gamma_{44} = \frac{1}{6} \\ \alpha_{42} &= 0.219945, \quad \alpha_{43} = \frac{1}{3}, \quad \gamma_{21} = -0.12642566, \quad \gamma_{31} = -0.5, \quad \gamma_{32} = -1.22102225482 \\ b_1 &= \frac{43}{758}, \quad b_2 = \frac{83}{192}, \quad b_3 = \frac{137}{768}, \quad b_4 = \frac{1}{3}\end{aligned}$$

and each $k_{i,n}$ denotes an approximation to certain information $k_i(t_n, h)$ about the exact solution $y(t)$, for each calculating step the unknowns $k_{1,n}$, $k_{2,n}$, $k_{3,n}$, $k_{4,n}$ can be calculated

in parallel on four processors with each processor only solving one linear system provided that the back values $y_n, k_{1,n-1}, k_{2,n-1}, k_{3,n-1}, k_{4,n-1}$ are known. Initially we assume some values to set $k_{1,n-1}, k_{2,n-1}, k_{3,n-1}, k_{4,n-1}$ as zero.

We write

$$u_{i,n} = \sum_{j=1}^i \gamma_{ij} k_{j,n}, \quad i = 1, 2, 3, 4 \quad \text{for } n = 1, 2, \dots \quad (17)$$

If $\gamma_{ii} \neq 0$ for all i , then the matrix $\Gamma = (\gamma_{ij})$ is invertible and then we write k_i as

$$k_{i,n} = \frac{u_{i,n}}{\gamma_{ii}} - \sum_{j=1}^{i-1} c_{ij} u_{j,n-1} \quad \text{where}$$

$$C = \text{diag}(\gamma_{11}^{-1}, \gamma_{22}^{-1}, \gamma_{33}^{-1}, \gamma_{44}^{-1}) - \Gamma^{-1}, \quad i = 1, 2, 3, 4$$

Inserting in Eq. (16) and dividing by h , we get

$$\left(\frac{I}{h\gamma_{ii}} - J \right) u_{i,n} = f \left(y_n + \sum_{j=1}^{i-1} a_{ij} u_{j,n-1} \right) + \sum_{j=1}^{i-1} \frac{c_{ij} u_{j,n-1}}{h}, \quad i = 1, 2, 3, 4 \quad (18)$$

$$y_{n+1} = y_n + \sum_{j=1}^4 m_j u_j$$

$$(a_{ij}) = (\alpha_{ij}) \Gamma^{-1}, \quad (m_1, m_2, m_3, m_4) = (b_1, b_2, b_3, b_4) \Gamma^{-1}, \quad i = 1, 2, 3, 4$$

Each u_i is a system to be solved using a call to LU decomposition. Therefore at each step (i) the Jacobian is evaluated and (ii) four linear systems have been solved. The following are the steps taken to implement the algorithm on a parallel machine having four processors P1, P2, P3 and P4.

Step 1

Compute $h = \frac{x_n - x_0}{n}$, the step-size of the method on P1 and initially allocate $k_{1,n-1} = 0$ on P1, $k_{2,n-1} = 0$ on P2, $k_{3,n-1} = 0$ on P3 and $k_{4,n-1} = 0$ on P4.

Step 2

Evaluate

$$\left(\frac{I}{h\gamma_{11}} - J \right) u_{1,n} = f(y_n) \text{ on P1}$$

$$\left(\frac{I}{h\gamma_{22}} - J \right) u_{2,n} = f(y_n + a_{21}u_{1,n-1}) + \frac{c_{21}u_{1,n-1}}{h} \text{ on P2}$$

$$\left(\frac{I}{h\gamma_{33}} - J \right) u_{3,n} = f(y_n + a_{31}u_{1,n-1} + a_{32}u_{2,n-1}) + \frac{c_{31}u_{1,n-1} + c_{32}u_{2,n-1}}{h} \text{ on P3 and}$$

$$\left(\frac{I}{h\gamma_{44}} - J \right) u_{4,n} = f(y_n + a_{41}u_{1,n-1} + a_{42}u_{2,n-1} + a_{43}u_{3,n-1})$$

$$+ \frac{c_{41}u_{1,n-1} + c_{42}u_{2,n-1} + c_{43}u_{3,n-1}}{h}$$

on P4, respectively where $u_{j,n}$ are defined as in Eq. (18).

Step 3

Finally, the results are combined to yield an approximation to the solution at the next time step. This step requires the computation

$$y_{n+1} = y_n + \sum_{j=1}^4 m_j u_j$$

on any of the processors P1, P2, P3 and P4.

Derivation of the LTE

By making use of the order conditions for fifth order trees in Table 1, the LTE can be obtained. Since the Rosenbrock schemes are the particular case of implicit Runge–Kutta methods or semi-implicit Runge–Kutta methods, the LTE is obtained using the following Theorem 1.

Theorem 1 *If the Runge–Kutta method is of order p and if f is $(p + 1)$ -times continuously differentiable, then the LTE can be obtained from*

$$y^J(x_0 + h) - y_1^J = \frac{h^{p+1}}{(p+1)!} \sum_{t \in T_{p+1}} \alpha(t) e(t) F^J(t)(y_0) + O(h^{p+2}) \quad (19)$$

where

$$e(t) = 1 - \gamma(t) \sum_{j=1}^s b_j \Phi_j(t) \quad (20)$$

and $\alpha(t)$ be the number of possible different monotonic labellings of t , $\gamma(t)$ be the density of the tree $t \in \text{LT}_q$ and $\Phi_j(t)$ be the elementary weight $\Phi_j(t)$ for stage j . The expressions $e(t)$ are called the error coefficients (Hairer et al. [27]).

The error coefficients and the derivative terms in the LTE are computed using the elementary differentials corresponding to fifth order trees given in Table 1. We assume that $f^1 = 1$, $f^2 = f$, $J=1, 2$. The following are the error coefficients and their corresponding derivatives:

$$e_1 = \left\{ 1 - 5 \left(b_2 \alpha_{2,1}^4 + b_3 \alpha_{3,1}^4 + 4b_3 \alpha_{3,1}^3 \alpha_{3,2} + 6b_3 \alpha_{3,1}^2 \alpha_{3,2}^2 + 4b_3 \alpha_{3,1} \alpha_{3,2}^3 + b_3 \alpha_{3,2}^4 \right) \right\} \quad (21)$$

$$e_2 = \left\{ 1 - 10(b_3 \alpha_{3,1}^2 \alpha_{3,2} \beta_{2,1} + 2b_3 \alpha_{3,1} \alpha_{3,2}^2 \beta_{2,1} + b_3 \alpha_{3,2}^3 \beta_{2,1} + b_2 \alpha_{2,1}^3 \gamma_{1,1} + b_3 \alpha_{3,1}^3 \gamma_{1,1} + 2b_3 \alpha_{3,1}^2 \alpha_{3,2} \gamma_{1,1} + b_3 \alpha_{3,1} \alpha_{3,2}^2 \gamma_{1,1} + b_3 \alpha_{3,1}^2 \alpha_{3,2} \gamma_{2,2} + 2b_3 \alpha_{3,1} \alpha_{3,2}^2 \gamma_{2,2} + b_3 \alpha_{3,2}^3 \gamma_{2,2}) \right\} \quad (22)$$

$$e_3 = \left\{ 1 - 15 \left(b_3 \alpha_{2,1}^2 \alpha_{3,1} \alpha_{3,2} + b_3 \alpha_{2,1}^2 \alpha_{3,2}^2 \right) \right\} \quad (23)$$

$$e_4 = \left\{ 1 - 30(b_3 \alpha_{3,1} \alpha_{3,2} \beta_{2,1} \gamma_{1,1} + b_3 \alpha_{3,2}^2 \beta_{2,1} \gamma_{1,1} + b_2 \alpha_{2,1}^2 \gamma_{1,1}^2 + b_3 \alpha_{3,1}^2 \gamma_{1,1}^2 + b_3 \alpha_{3,1} \alpha_{3,2} \gamma_{1,1}^2 + b_3 \alpha_{3,1} \alpha_{3,2} \beta_{2,1} \gamma_{2,2} + b_3 \alpha_{3,2}^2 \beta_{2,1} \gamma_{2,2} + b_3 \alpha_{3,1} \alpha_{3,2} \gamma_{2,2}^2 + b_3 \alpha_{3,2}^2 \gamma_{2,2}^2) \right\} \quad (24)$$

$$e_5 = \left\{ 1 - 20(b_3 \alpha_{3,1}^3 \beta_{3,2} + b_4 \alpha_{2,1}^3 \beta_{4,2} + b_4 \alpha_{3,1}^3 \beta_{4,3} + 3b_4 \alpha_{3,1}^2 \alpha_{3,2} \beta_{4,3} + 3b_4 \alpha_{3,1} \alpha_{3,2}^2 \beta_{4,3} + b_4 \alpha_{3,2}^3 \beta_{4,3} + b_2 \alpha_{2,1}^3 \gamma_{2,2} + b_3 \alpha_{3,1}^3 \gamma_{3,3} + 3b_3 \alpha_{3,1}^2 \alpha_{3,2} \gamma_{3,3} + 3b_3 \alpha_{3,1} \alpha_{3,2}^2 \gamma_{3,3} + b_3 \alpha_{3,2}^3 \gamma_{3,3}) \right\} \quad (25)$$

$$e_6 = \left\{ 1 - 20(b_3\alpha_{2,1}^3\beta_{3,2} + b_4\alpha_{2,1}^3\beta_{4,2} + b_4\alpha_{3,1}^3\beta_{4,3} + 3b_4\alpha_{3,1}^2\alpha_{3,2}\beta_{4,3} + 3b_4\alpha_{3,1}\alpha_{3,2}^2\beta_{4,3} \right. \\ \left. + b_4\alpha_{3,2}^3\beta_{4,2} + b_2\alpha_{3,2}^3\gamma_{2,2} + b_3\alpha_{3,1}^3\gamma_{3,3} + 3b_3\alpha_{3,1}^2\alpha_{3,2}\gamma_{3,3} + 3b_3\alpha_{3,1}\alpha_{3,2}^2\gamma_{3,3} \right. \\ \left. + b_3\alpha_{3,2}^3\gamma_{3,3} \right\} \quad (26)$$

$$e_7 = \left\{ 1 - 40(b_4\alpha_{3,1}\alpha_{3,2}\beta_{2,1}\beta_{4,3} + b_4\alpha_{3,2}^2\beta_{2,1}\beta_{4,3} + b_3\alpha_{2,1}^2\beta_{3,2}\gamma_{1,1} + b_4\alpha_{2,1}^2\beta_{4,2}\gamma_{1,1} \right. \\ \left. + b_4\alpha_{3,1}^2\beta_{4,3}\gamma_{1,1} + b_4\alpha_{3,1}\alpha_{3,2}\beta_{4,3}\gamma_{1,1} + b_4\alpha_{3,1}\alpha_{3,2}\beta_{4,3}\gamma_{2,2} + b_4\alpha_{3,2}^2\beta_{4,3}\gamma_{2,2} \right. \\ \left. + b_2\alpha_{2,1}^2\gamma_{1,1}\gamma_{2,2} + b_3\alpha_{3,1}\alpha_{3,2}\beta_{2,1}\gamma_{2,2} + b_3\alpha_{3,2}^2\beta_{2,1}\gamma_{3,3} + b_3\alpha_{3,1}^2\gamma_{1,1}\gamma_{3,3} \right. \\ \left. + b_3\alpha_{3,1}\alpha_{3,2}\gamma_{2,2}\gamma_{3,3} + b_4\alpha_{3,2}^2\gamma_{2,2}\gamma_{3,3} \right\} \quad (27)$$

$$e_8 = \left\{ 1 - 60(b_4\alpha_{2,1}^2\beta_{3,2}\beta_{4,3} + b_3\alpha_{2,1}^2\beta_{3,2}\gamma_{2,2} + b_4\alpha_{2,1}^2\beta_{4,2}\gamma_{2,2} + b_2\alpha_{2,1}^2\gamma_{2,2}^2 \right. \\ \left. + b_3\alpha_{2,1}^2\beta_{3,2}\gamma_{3,3} + b_4\alpha_{3,1}^2\beta_{4,3}\gamma_{3,3} + 2b_4\alpha_{3,1}\alpha_{3,2}\beta_{4,3}\gamma_{3,3} + b_4\alpha_{3,2}^2\beta_{4,3}\gamma_{3,3} \right. \\ \left. + b_3\alpha_{3,1}^2\gamma_{3,3}^2 + 2b_3\alpha_{3,1}\alpha_{3,2}\gamma_{3,3}^2 + b_3\alpha_{3,2}^2\gamma_{3,3}^2 + b_4\alpha_{2,1}^2\beta_{4,2}\gamma_{4,4} + b_4\alpha_{3,1}^2\beta_{4,3}\gamma_{4,4} \right\} \\ \left. + 2b_4\alpha_{3,1}\alpha_{3,2}\beta_{4,3}\gamma_{4,4} + b_4\alpha_{3,2}^2\beta_{4,3}\gamma_{4,4} \right) \quad (28)$$

$$e_9 = \left\{ 1 - 120(b_4\beta_{2,1}\beta_{3,2}\beta_{4,3}\gamma_{1,1} + b_3\beta_{2,1}\beta_{3,2}\gamma_{1,1}^2 + b_4\beta_{2,1}\beta_{4,2}\gamma_{1,1}^2 + b_4\beta_{3,1}\beta_{4,3}\gamma_{1,1}^2 \right. \\ \left. + b_2\beta_{2,1}\gamma_{1,1}^3 + b_3\beta_{3,1}\gamma_{1,1}^3 + b_4\beta_{4,1}\gamma_{1,1}^3 + b_1\gamma_{1,1}^4 + b_4\beta_{2,1}\beta_{3,2}\beta_{4,3}\gamma_{2,2} \right. \\ \left. + b_3\beta_{2,1}\beta_{3,2}\gamma_{1,1}\gamma_{2,2} + b_4\beta_{2,1}\beta_{4,2}\gamma_{1,1}\gamma_{2,2} + b_2\beta_{2,1}\gamma_{1,1}^2\gamma_{2,2} + b_3\beta_{2,1}\beta_{3,2}\gamma_{2,2}^2 \right. \\ \left. + b_4\beta_{2,1}\beta_{4,2}\gamma_{2,2}^2 + b_4\beta_{3,2}\beta_{4,3}\gamma_{2,2}^2 + b_2\beta_{2,1}\gamma_{1,1}\gamma_{2,2}^2 + b_2\beta_{2,1}\gamma_{2,2}^3 + b_3\beta_{3,2}\gamma_{2,2}^3 \right. \\ \left. + b_4\beta_{4,2}\gamma_{2,2}^3 + b_2\gamma_{2,2}^4 + b_4\beta_{2,1}\beta_{3,2}\beta_{4,3}\gamma_{3,3} + b_3\beta_{2,1}\beta_{3,2}\gamma_{1,1}\gamma_{3,3} + b_4\beta_{3,1}\beta_{4,3}\gamma_{1,1}\gamma_{3,3} \right. \\ \left. + b_3\beta_{3,1}\gamma_{1,1}^2\gamma_{3,3} + b_3\beta_{2,1}\beta_{3,2}\gamma_{2,2}\gamma_{3,3} + b_4\beta_{3,2}\beta_{4,3}\gamma_{2,2}\gamma_{3,3} + b_3\beta_{3,2}\gamma_{2,2}^2\gamma_{3,3} \right. \\ \left. + b_3\beta_{2,1}\beta_{3,2}\gamma_{3,3}^2 + b_4\beta_{3,1}\beta_{4,3}\gamma_{3,3}^2 + b_4\beta_{3,2}\gamma_{4,3}\gamma_{3,3}^2 + b_3\beta_{3,1}\gamma_{1,1}\gamma_{3,3}^2 + b_3\beta_{3,2}\gamma_{2,2}\gamma_{3,3}^2 \right. \\ \left. + b_3\beta_{3,1}\gamma_{3,3}^3 + b_3\beta_{3,2}\gamma_{3,3}^3 + b_4\beta_{4,3}\gamma_{3,3}^3 + b_3\gamma_{3,3}^4 + b_4\beta_{2,1}\beta_{3,2}\beta_{4,3}\gamma_{4,4} \right. \\ \left. + b_4\beta_{2,1}\beta_{4,2}\gamma_{1,1}\gamma_{4,4} + b_4\beta_{3,1}\beta_{4,3}\gamma_{1,1}\gamma_{4,4} + b_4\beta_{3,1}\beta_{4,3}\gamma_{1,1}\gamma_{4,4} + b_4\beta_{4,1}\gamma_{1,1}^2\gamma_{4,4} \right. \\ \left. + b_4\beta_{2,1}\beta_{4,2}\gamma_{2,2}\gamma_{4,4} + b_4\beta_{3,2}\beta_{4,3}\gamma_{3,3}\gamma_{4,4} + b_4\beta_{4,3}\gamma_{3,3}^2\gamma_{4,4} + b_4\beta_{2,1}\beta_{4,2}\gamma_{4,4}^2 \right. \\ \left. + b_4\beta_{3,3}\beta_{4,3}\gamma_{4,4}^2 + b_4\beta_{3,2}\beta_{4,3}\gamma_{4,4}^2 + b_4\beta_{4,1}\gamma_{1,1}\gamma_{4,4}^2 + b_4\beta_{4,2}\gamma_{2,2}\gamma_{4,4}^2 \right. \\ \left. + b_4\beta_{4,2}\gamma_{2,2}\gamma_{4,4}^2 + b_4\beta_{4,3}\gamma_{3,3}\gamma_{4,4}^2 + b_4\beta_{4,1}\gamma_{4,4}^3 + b_4\beta_{4,2}\gamma_{4,4}^3 + b_4\beta_{4,3}\gamma_{4,4}^3 + b_4\gamma_{4,4}^4 \right) \} \quad (29)$$

The derivative terms are computed and given as follows:

$$f_{xxxx} + 4f_{xxxy}f + 6f_{xxyy}f^2 + 4f_{xyyy}f^3 + f_{yyyy}f^4 \quad \text{for } t \in t_{61}$$

$$f_{xxy}f_x + 2f_{xyy}f_xf + f_{xxy}f_yf + 2f_{xyy}f_yf^2 + f_{yyy}f_xf^2 + f_{yyy}f_yf^3 \quad \text{for } t \in t_{62}$$

$$f_{xy}f_{xx} + 2f_{xy}^2f + f_{yy}f_{xx}f^2 + 2f_{xy}f_{yy}f^2 + f_{yy}f_{xy}f^2 + f_{yy}^2f^3 \quad \text{for } t \in t_{63}$$

$$f_{xy}f_yf_x + f_{xy}f_y^2f + f_{yy}f_yf_xf + f_{yy}f_y^2f^2 \quad \text{for } t \in t_{64}$$

$$f_{yy}f_x^2 + 2f_{yy}ff_yf_x + f_{yy}f_y^2f^2 \quad \text{for } t \in t_{65}$$

$$f_yf_{xxx} + 3f_yf_{xxy} + 3f_{xxy}f_yf^2 + f_yf_{xxx}f^3 \quad \text{for } t \in t_{66}$$

$$f_{xy}f_yf_x + f_{xy}f_y^2f + f_{yy}f_yf_xf + f_{yy}f_y^2f^2 \quad \text{for } t \in t_{67}$$

$$f_y^2f_{xx} + 2f_{xy}f_y^2f + f_{yy}f_y^2f^2 \quad \text{for } t \in t_{68}$$

$$f_y^3f_x + f_y^4f^2 \quad \text{for } t \in t_{69}.$$

Using Eqs. (21)–(29) and the derivative terms (given in Table 1), the LTE for the NPROS4(4) is computed. In this manner, the expression of LTE for a parallel fourth order four stage Rosenbrock algorithm is obtained as follows.

$$\begin{aligned} \text{LTE} = \frac{h^5}{5!} \{ & (e_1) (f_{xxxx} + 4f_{xxxy}f + 6f_{xxyy}f^2 + 4f_{xyyy}f^3 + f_{yyyy}f^4) \\ & + (6e_2) (f_{xxy}f_x + 2f_{xyy}f_xf + f_{xxy}f_yf + 2f_{xyy}f_yf^2 + f_{yyy}f_xf^2 + f_{yyy}f_yf^3) \\ & + (4e_4) (f_{xy}f_yf_x + f_{xy}f_y^2f + f_{yy}f_yf_xf + f_{yy}f_y^2f^2) + (3e_5) (f_{yy}f_x^2 \\ & + 2f_{yy}ff_yf_x + f_{yy}f_y^2f^2) + (e_6) (f_yf_{xxx} + 3f_yff_{xxy} + 3f_{xyy}f_yf^2 \\ & + f_yf_{xxx}f^3) + (3e_7) (f_{xy}f_yf_x + f_{xy}f_y^2f + f_{yy}f_yf_xf + f_{yy}f_y^2f^2) \\ & + (e_8) (f_y^2f_{xx} + 2f_{xy}f_y^2f + f_{yy}f_y^2f^2) + (e_9) (f_y^3f_x + f_y^4f^2) \} \end{aligned}$$

The values of $e_1, e_2, \dots, e_9$ are calculated by using the values of the parameters of NPROS4(4) $\alpha_{21}, \alpha_{31}, \alpha_{32}, \alpha_{41}, \alpha_{42}, \alpha_{43}, \gamma_{11}, \gamma_{21}, \gamma_{22}, \gamma_{31}, \gamma_{32}, \gamma_{33}, \gamma_{41}, \gamma_{42}, \gamma_{43}$ and b_1, b_2, b_3, b_4 .

By using the values given below,

$$\begin{aligned} \alpha_{21} &= \frac{1}{2}, \quad \alpha_{31} = \frac{1}{2}, \quad \alpha_{32} = \frac{1}{2}, \quad \alpha_{41} = -0.178278, \quad \gamma_{11} = \frac{1}{6}, \quad \gamma_{22} = \frac{2}{6}, \quad \gamma_{33} = \frac{2}{6} \\ \gamma_{44} &= \frac{1}{6}, \quad \alpha_{42} = 0.219945, \quad \alpha_{43} = \frac{1}{3}, \quad \gamma_{21} = -0.12642566, \quad \gamma_{31} = -0.5 \\ \gamma_{32} &= -1.2210222548, \quad b_1 = \frac{43}{758}, \quad b_2 = \frac{83}{192}, \quad b_3 = \frac{137}{768}, \quad b_4 = \frac{1}{3} \end{aligned}$$

the LTE of NPROS4(4) is obtained as,

$$\begin{aligned} & \frac{1}{120}h^5 \left(0.259259259 (f_x^3 + f_y^4) - 0.13564066 (f_{xx}f_y^2 + 2ff_{xy}f_y^2 + f^2f_y^2f_{yy}) \right. \\ & - 1.8948329844 (f_xf_{xy}f_y + ff_{xy}f_y^2 + ff_xf_yf_{yy} + f^2f_y^2f_{yy}) \\ & + \frac{1,363}{512} (f_{xx}f_{xy} + 2ff_{xy}^2 + f^2f_{xx}f_{yy} + 3f^2f_{xy}f_{yy} + f^3f_{yy}^2) \\ & - 0.5685019 (f_xf_{xxy} + 2ff_xf_{xyy} + ff_{xxy}f_y + 2f^2f_{xyy}f_y + f^2f_xf_{yyy} + f^3f_yf_{yyy}) \\ & - \frac{83 (f_{xxxx} + 4ff_{xxx} + 6f^2f_{xxy} + 4f^3f_{xyy} + f^4f_{yyy})}{3,072} \\ & - 0.214438 (f_{xxx}f_y + f^3f_{xxx}f_y + 3f^2f_{xyy}f_y + 3f_yff_{xxy}) \\ & \left. - 0.643315606 (f_x^2f_{yy} + f^2f_y^2f_{yy} + 2f_xf_{yy}ff_y) \right) \end{aligned} \quad (30)$$

LTE Bound

If we assume the following bounds for f and its partial derivatives for $x \in [x_0, x_n]$ and $y \in [-\infty, \infty]$ according to Lotkin [28] and Ponalagusamy et al. [4], then we write,

$$\begin{aligned} |f(x, y)| &< Q \\ \left| \frac{\partial^{i+j} f(x, y)}{\partial x^i \partial y^j} \right| &< \frac{P^{i+j}}{Q^{j-1}}, \quad i + j \leq p \end{aligned} \quad (31)$$

where P and Q are positive constants, and p is the order of the method. Here $p = 4$. Using Eq. (31), the derivatives in Eq. (30) are obtained as

$$\begin{aligned} |f_x f_y^3| &< \frac{P}{Q^{-1}} \times \left(\frac{P^3}{Q^0}\right) = P^4 Q \\ |f f_y^4| &< Q \times \left(\frac{P}{Q^{1-1}}\right)^4 = P^4 Q \\ |f_{xx} f_y^2| &< \frac{P^2}{Q^{0-1}} \times \frac{P^2}{(Q^0)^2} = P^4 Q \end{aligned}$$

and so on.

Therefore, we obtain from Eq. (30),

$$\text{LTE}_{\text{NPROS4}(4)} \leq \frac{4.4244256673333h^5 P^4 Q}{120} \quad (32)$$

This LTE bound is useful in selecting suitable h values according to the error at each step. If we select an error tolerance $\text{TOL} = 0.0001$, then the step-size h can be made as

$$\frac{4.4244256673333h^5 P^4 Q}{120} < \text{TOL} \quad \text{or} \quad h < \left(\frac{27.1222 \cdot \text{TOL}}{P^4 Q}\right)^{\frac{1}{5}} \quad (33)$$

Convergence and Stability of NPROS4(4)

In regard to analyze the convergence properties, we consider the method given in Eq. (16) as a general multivalued method of the form (Li [29]) as,

$$y^{(n)} = Ay^{(n-1)} + h\varphi(y^{(n-1)}, h) \quad (34)$$

where

$$\begin{aligned} y^{(n)} &= \left(k_{1,n}^T, k_{2,n}^T, k_{3,n}^T, k_{4,n}^T, y_{n+1}^T\right)^T \\ A &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \varphi(y^{(n-1)}, h) = \begin{pmatrix} \varphi_1(y^{(n-1)}, h) \\ \varphi_2(y^{(n-1)}, h) \\ \varphi_3(y^{(n-1)}, h) \\ \varphi_4(y^{(n-1)}, h) \\ \sum_{i=1}^4 b_i \varphi_i(y^{(n-1)}, h) \end{pmatrix} \end{aligned} \quad (35)$$

Here,

$$\begin{aligned} \varphi_i(y^{(n-1)}, h) &= (I - h\gamma_{ii}J)^{-1} \left(f\left(y_n + \sum_{j=1}^{i-1} \alpha_{ij}k_{j,n-1}\right) + J \sum_{j=1}^{i-1} \beta_{ij}k_{j,n-1} \right), \\ i &= 1, 2, 3, 4 \end{aligned}$$

It is seen that the minimum polynomial $(\lambda - 1)$ of the matrix A satisfies the root condition, and therefore this method is convergent and its convergence order is equal to the consistency order.

Fig. 1 Stability region of NPROS4(4)

To investigate the numerical stability properties, we use the test equation $y' = \lambda y$, $h > 0$. The stability function of NPROS4(4) may be obtained as

$$\frac{(-12.4473 + z)(-2.31466 + z)(11.2456 + z(5.76191 + z))}{(18 + (-9 + z)z)^2}$$

The corresponding stability region (the white portion) is illustrated in Fig. 1. It is found that NPROS4(4) is strongly A-stable. The stability region of the existing three stage fourth order method MPROW is also presented in Fig. 2. We have observed that the stability region of NPROS4(4) includes the imaginary axis in comparison with that of MPROW. It is also possible to find other set of parameters for the new method NPROS4(4) described through the steps given in section 3 and one can determine the intervals of stability of various values of γ_{ii} , which also leads to have comparisons of various sets of parameters.

Numerical Experimentation

We present the results obtained for the integration of three problems of practical interest. The following adaptive procedure is adopted for the examples considered in order to have error control.

Adaptive Procedure

- (1) Initially, h is selected using Eq.(33) and a local error tolerance $\epsilon = 10^{-4}$ has been imposed.
- (2) The algorithm defined by Eq. (16) is implemented. The LTE is estimated using (30).
- (3) The stepsize of integration be h and that to be chosen for the next step is h_1 . The relationship between h and h_1 is as follows:
 - (a) If $\text{LTE} > \epsilon$, $h_1 = h/2$ and re-do the next step.

Fig. 2 Stability region of MPROW of Cao et al. [16]

- (b) If $\epsilon > \text{LTE} > \epsilon/20$, $h_1 = h$.
- (c) If $\text{LTE} < \epsilon/20$, $h_1 = 2h$.

Example 1 As a first example to verify the sharpness of our error bounds, we have considered the Robertson's chemical reaction problem. The problem consists of a stiff system of three non-linear ODEs and describes the kinetics of an autocatalytic reaction. The present problem is very popular in numerical studies and it is often used as a test problem in the stiff integrators comparisons (Edsberg [30]).

Mathematical Description of the Problem

The problem is of the form

$$\begin{cases} y' = f(y) \\ y(0) = y_0, \quad 0 \leq t \leq T \end{cases}$$

where

$$f(y) = \begin{pmatrix} -0.04y_1 + 10^4 y_2 y_3 \\ 0.04y_1 - 10^4 y_2 y_3 - 3 \times 10^7 y_2^2 \\ 3 \times 10^7 y_2^2 \end{pmatrix}, \quad y_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

The large difference among the reaction rate constants is the reason for stiffness. At $t = 0$, the three eigen values of the Jacobian are $\lambda_1 = \lambda_2 = 0$, $\lambda_3 = -0.04$ and for large t these eigen values approach the limits $\lambda_1 = \lambda_2 = 0$, $\lambda_3 = -10^4$. The stiffness ratio is 10^4 .

The reference solution computed by RADAU on an Alphaserwer DS20E, with a 667 MHz EV67 processor, using double precision (Mazzia and Iavernaro [31]), is assumed to be exact. Using adaptive step-size h , the computed result is listed in Table 1 for some solution steps integrated in the interval $[0, 400]$. The errors in the solutions of the problem mentioned above by using NPROS4(4) and MPROW are illustrated in Figs. 3, 4 and 5.

Table 1 Trees and order conditions up to order 5

q	t	Graph	$\gamma(t)$	$F^J(t)(y)$	$\sum_j b_j \Phi_j(t) = 1/\gamma(t)$
1	τ		1	f^J	$\sum_j b_j = 1$
2	t_{21}		2	$\sum_K f_K^L f^K$	$\sum_k \beta_{jk} = 1/2$
3	t_{31}		3	$\sum_{K,L} f_{KL}^J f_L^K f^L$	$\sum_{k,l} \alpha_{jk} \alpha_{jl} = 1/3$
3	t_{32}		6	$\sum_{K,L} f_{KL}^J f_L^K f^L$	$\sum_{k,l} \beta_{jk} \beta_{kl} = 1/6$
4	t_{41}		4	$\sum_{K,L,M} f_{KLM}^J f^K f^L f^M$	$\sum_{k,l,m} \alpha_{jk} \alpha_{jl} \alpha_{jm} = 1/4$
4	t_{42}		8	$\sum_{K,L,M} f_{KM}^J f_L^K f^L f^M$	$\sum_{k,l,m} \alpha_{jk} \beta_{kl} \alpha_{jm} = 1/8$
4	t_{43}		12	$\sum_{K,L,M} f_K^J f_{LM}^K f^L f^M$	$\sum_{k,l,m} \beta_{jk} \alpha_{kl} \alpha_{km} = 1/12$
4	t_{44}		24	$\sum_{K,L,M} f_K^J f_L^K f_M^L f^M$	$\sum_{k,l,m} \beta_{jk} \beta_{kl} \beta_{lm} = 1/24$
5	t_{51}		5	$\sum_{K,L,M,P} f_{KLM P}^J f^K f^L f^M f^P$	$\sum_{k,l,m,p} \alpha_{jk} \alpha_{jl} \alpha_{jm} \alpha_{jp} = 1/5$
5	t_{52}		10	$\sum_{K,L,M,P} f_{KM P}^J f_L^K f^L f^M f^P$	$\sum_{k,l,m,p} \alpha_{jk} \beta_{kl} \alpha_{jm} \alpha_{jp} = 1/10$
5	t_{53}		15	$\sum_{K,L,M,P} f_{K P}^J f_{M L}^K f^L f^M f^P$	$\sum_{k,l,m,p} \alpha_{jk} \alpha_{kl} \alpha_{km} \alpha_{jp} = 1/15$
5	t_{54}		30	$\sum_{K,L,M,P} f_{K P}^J f_L^K f_M^L f^M f^P$	$\sum_{k,l,m,p} \alpha_{jk} \beta_{kl} \beta_{lm} \alpha_{jp} = 1/30$
5	t_{55}		20	$\sum_{K,L,M,P} f_{K M}^J f_L^K f^L f^M f^P$	$\sum_{k,l,m,p} \alpha_{jk} \beta_{kl} \alpha_{jm} \beta_{mp} = 1/20$
5	t_{56}		20	$\sum_{K,L,M,P} f_{K}^J f_{L M P}^K f^L f^M f^P$	$\sum_{k,l,m,p} \beta_{jk} \alpha_{kl} \alpha_{km} \alpha_{kp} = 1/20$
5	t_{57}		40	$\sum_{K,L,M,P} f_K^J f_{L P}^K f_M^L f^M f^P$	$\sum_{k,l,m,p} \beta_{jk} \alpha_{kl} \beta_{lm} \alpha_{kp} = 1/40$

Table 1 continued

q	t	Graph	$\gamma(t)$	$F^J(t)(y)$	$\sum_j b_j \Phi_j(t) = 1/\gamma(t)$
5	t_{58}		60	$\sum_{K,L,M,P} f_K^J f_L^K f_{MP}^L f^M f^P$	$\sum_{k,l,m,p} \beta_{jk} \beta_{kl} \alpha_{lm} \alpha_{lp} = 1/60$
5	t_{59}		120	$\sum_{K,L,M,P} f_K^J f_L^K f_M^L f_P^M f^P$	$\sum_{k,l,m,p} \beta_{jk} \beta_{kl} \beta_{lm} \beta_{mp} = 1/120$

Fig. 3 Error using NPROS4(4) and MPROW for the first 10 steps of Example 1 **Fig. 4** Error using NPROS4(4) for the Example 1

Fig. 5 Error using MPROW for the Example 1

Example 2 We consider weakly damped oscillatory stiff differential equations given by

$$\begin{cases} y' = Ay \\ y(0) = y_0, \quad 0 \leq t \leq 10 \end{cases}$$

where

$$A = \begin{pmatrix} -0.01 & -1 & -1 \\ 2 & -100.005 & 99.995 \\ 2 & 99.995 & -100.005 \end{pmatrix}, \quad y_0 = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$$

The exact solution of the system is

$$\begin{aligned} y_1(t) &= e^{-0.01t} (\cos 2t - \sin 2t) \\ y_2(t) &= e^{-0.01t} (\cos 2t + \sin 2t) + e^{-200t} \\ y_3(t) &= e^{-0.01t} (\cos 2t + \sin 2t) - e^{-200t}. \end{aligned}$$

The solutions using NPROS4(4) and MPROW are presented in Table 2. Figures 6, 7 and 8 represent the errors in solutions computed by the two methods.

Example 3 The problem consists of a stiff system of three non-linear ODEs. This refers to the Oregonator popular model which is described by ODE. Field et al. [32] proposed this model for the Belousov–Zhabotinskii (BZ) reaction and formulated the model for the most important parts of the kinetic mechanism that gives rise to oscillation in the BZ reaction. This mechanism can be summarized as three concurrent processes by Gray [33].

Mathematical Description of the Problem

The problem is of the form

$$\begin{cases} y' = f(y) \\ y(0) = y_0, \quad 0 \leq t \leq 360 \end{cases}$$

Table 2 NPROS4(4) and MPROW solutions for Example 1

t	NPROS4(4)		ns	MPROW		ns
	H	$[y_1, y_2, y_3]^T$		h	$[y_1, y_2, y_3]^T$	
0.24603	6.45412×10^{-2}	$9.899655172 \times 10^{-1}$ $3.470586395 \times 10^{-5}$ $1.000000012 \times 10^{-2}$	83	6.15402×10^{-3}	$9.899656272 \times 10^{-1}$ $3.470582596 \times 10^{-5}$ $1.000000012 \times 10^{-2}$	96
0.40000	6.45412×10^{-2}	$9.851721250 \times 10^{-1}$ $3.386395399 \times 10^{-5}$ $1.479572101 \times 10^{-2}$	85	5.66501×10^{-4}	$9.851721125 \times 10^{-1}$ $3.386395179 \times 10^{-5}$ $1.479572002 \times 10^{-2}$	112
4.0000	4.13257×10^{-1}	$9.055086791 \times 10^{-1}$ $2.240375693 \times 10^{-5}$ $9.446891665 \times 10^{-2}$	101	7.83233×10^{-1}	$9.055086865 \times 10^{-1}$ $2.240375213 \times 10^{-5}$ $9.446890676 \times 10^{-2}$	154
40	1.02354	$7.158180688 \times 10^{-1}$ $9.185534768 \times 10^{-6}$ $2.841627459 \times 10^{-1}$	134	0.12975	$7.158180889 \times 10^{-1}$ $9.185533812 \times 10^{-6}$ $2.841627459 \times 10^{-1}$	176
400	3.35357×10^1	$4.504876685 \times 10^{-1}$ $3.222901442 \times 10^{-6}$ $5.495071086 \times 10^{-1}$	185	4.26468×10^{-1}	$4.504875723 \times 10^{-1}$ $3.222900133 \times 10^{-6}$ $5.495071152 \times 10^{-1}$	239

Fig. 6 Error using NPROS4(4) and MPROW of Example 2

where

$$f(y) = \begin{pmatrix} 77.27 (y_2 - y_1 y_2 + y_1 - 8.375 \times 10^{-6} y_1^2) \\ \frac{1}{77.27} (-y_2 - y_1 y_2 + y_3) \\ 0.161 (y_1 - y_3) \end{pmatrix}, \quad y_0 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

Fig. 7 Error for the Example 2 using NPROS4(4)

Fig. 8 Error for the Example 2 using MPROW

Using double precision, the reference solution is computed by RADAU on an Alphaser server DS20E, with a 667 MHz EV67 processor and it is considered as exact solution (Mazzia and Magherini [34]).

This problem is taken as tool to test any stiff solver by several researchers who have been performing active research in the field of stiff differential equations. We can also find the useful discussions of this problem in Cash [12].

The problem is solved by using NPROS4(4) and MPROW and the results are tabulated in Table 3. The errors computed by using NPROS4(4) and MPROW are presented in Figs. 9,

Table 3 NPROS4(4) and MPROW solutions for Example 2

t	NPROS4(4)		MPROW		Exact solution
	H	$[y_1, y_2, y_3]^T$	h	$[y_1, y_2, y_3]^T$	
1.0618	0.1142	-1.361596640 0.322537409 0.322537409	0.0984	-1.361596640 0.322537409 0.322537409	-1.361596640 0.322537409 0.322537409
2.1248	0.2585	0.438939034 -1.313057928 -1.313057928	0.0619	0.438939034 -1.313057928 -1.313057928	0.438939034 -1.313057928 -1.313057928
3.0762	0.1585	1.087886938 0.834962733 0.834962733	0.1845	1.087886938 0.834962733 0.834962733	1.087886938 0.834962733 0.834962733
4.2053	0.1454	-1.320643295 0.307520549 0.307520549	0.0958	-1.320643295 0.307520549 0.307520549	-1.320643295 0.307520549 0.307520549
5.2839	0.1541	0.263586741 -1.316370783 -1.316370783	0.0730	0.263586741 -1.316370783 -1.316370783	0.263586741 -1.316370783 -1.316370783
6.8043	0.1451	-0.3356039517 1.2778517640 1.2778517640	0.0626	-0.3356039517 1.2778517640 1.2778517640	-0.3356039517 1.2778517640 1.2778517640
7.8059	0.1538	-1.009441993 -0.831831319 -0.831831319	0.1981	-1.009441993 -0.831831319 -0.831831319	-1.009441993 -0.831831319 -0.831831319
8.8854	0.1569	1.238695095 -0.374176635 -0.374176635	0.1089	1.238695095 -0.374176635 -0.374176635	1.238695095 -0.374176635 -0.374176635
9.9419	0.1360	-0.315338409 1.240937963 1.240937963	0.1401	-0.315338409 1.240937963 1.240937963	-0.315338409 1.240937963 1.240937963
10.0000	0.1691	-0.456819104 1.195314942 1.195314942	0.1519	-0.456819104 1.195314942 1.195314942	-0.456819104 1.195314942 1.195314942

10 and 11. Figure 9 presents the error found in both the methods. Figures 10 and 11 show the error found using NPROS4(4) and error found using MPROW, respectively.

Discussion

In the case of Example 1, the total number of steps to achieve the solution for $t = 400$ by NPROS4(4) is 185 which is much lesser than the steps taken by MPROW which comes as 239. This is where the problem is run in sequential and at each step the function evaluations required to evaluate the solution is 4×185 , that is, 740 function evaluations are required

Fig. 9 Error using NPROS4(4) and MPROW of Example 3

Fig. 10 Error using NPROS4(4) of Example 3

to complete the integration in sequential. It is of interest to pin-point out that in the case of a parallel machine, the function evaluations can be performed concurrently, and hence the integration can be completed quickly with 185 function evaluations. Further, Table 1 and Figs. 3, 4 and 5 show that NPROS4(4) could be a better stiff solver for the Robertson's chemical reaction problem (Example 1) than MPROW in view of faster convergence towards solution and error control. One can see that an improvement in reducing the number of functions evaluations is gained by the present method of NPROS4(4) in comparison with the method of MPROW.

Fig. 11 Error using MPROW of Example 3

It is observed from Example 2 that both NPROS4(4) and MPROW perform well with solutions correct to 12 decimal places. But the number of steps taken to integrate the differential equations is considerably less in NPROS4(4) method as compared to that of MPROW method. Further, NPROS4(4) takes 41 steps using adaptive step-size whereas the number of steps taken by MPROW in the interval $[0, 10]$ is 97. Hence, the number of function evaluations performed by NPROS4(4) in parallel machine remains 41 unlike the case of MPROW which takes 97 function evaluations in a parallel machine. Numerical solutions by NPROS4(4) and MPROW and exact solutions for weekly damped oscillatory stiff differential equations are tabulated in Table 2 and the corresponding errors are shown in Figs. 6, 7 and 8.

Another important observation from Example 3 is that NPROS4(4) requires 248 steps in the integration interval $[0, 360]$ to complete the iterations but MPROW requires 791 steps to complete the iteration in the same interval. The solutions are given in Table 3 and the errors are plotted in Figs. 9, 10 and 11. It is pertinent to note that the presently developed method of NPROS4(4) outperforms well and is more suitable and efficient one for solving Oregonator model in comparison with MPROW method (Table 4).

Further, in a p -processor environment where $p=s=4$, NPROS4(4) requires the computations of s function evaluations, one Jacobian evaluation and one LU decomposition at each function evaluation. Since one LU decomposition requires $O(s^3)$ for the input size s , the speed-up is calculated as follows:

$$S = \frac{t_s}{t_{p=4}} = \frac{s \times O(s^3)}{O(s^3)} = s$$

where t_s is the time period elapsed between the beginning and end of the algorithm execution, parallel time, t_p , is the time period elapsed between the beginning of the first processor and the end of the last processor during the execution of the algorithm and p is the number of processors used.

Table 4 NPROS4(4) and MPROW solutions for Example 3

t	NPROS4(4)		MPROW		Exact solution
	Adaptive h	$[y_1, y_2, y_3]^T$	Adaptive h	$[y_1, y_2, y_3]^T$	
30	2.1386×10^0	$1.00066146729049 \times 10^0$ $1.51277893734824 \times 10^3$ $1.03585431276722 \times 10^4$	3.5557×10^{-1}	$1.00066147629049 \times 10^0$ $1.51277893734825 \times 10^3$ $1.03585431276726 \times 10^4$	$1.00066146718049 \times 10^0$ $1.51277893734824 \times 10^3$ $1.03585431276722 \times 10^4$
60	2.7906×10^0	$1.00087462530962 \times 10^0$ $1.14433697238506 \times 10^3$ $8.37214996662763 \times 10^1$	4.0332×10^{-1}	$1.00087462551062 \times 10^0$ $1.14433697239124 \times 10^3$ $8.372149966686763 \times 10^1$	$1.00087462519962 \times 10^0$ $1.14433697238449 \times 10^3$ $8.37214996662463 \times 10^1$
90	3.3306×10^0	$1.00189036844315 \times 10^0$ $5.29992623230355 \times 10^2$ $1.66227957956242 \times 10^0$	5.6841×10^{-1}	$1.00189036851315 \times 10^0$ $5.29992623232355 \times 10^2$ $1.66227958891242 \times 10^0$	$1.00189036843875 \times 10^0$ $5.29992623229555 \times 10^2$ $1.66227957904242 \times 10^0$
120	8.4240×10^0	$1.00411802261534 \times 10^0$ $2.43832608046034 \times 10^2$ $1.00882222405864 \times 10^0$	2.1942×10^0	$1.00411802262291 \times 10^0$ $2.43832608696534 \times 10^2$ $1.00882222414864 \times 10^0$	$1.00411802261264 \times 10^0$ $2.43832607991034 \times 10^2$ $1.00882222404864 \times 10^0$
150	1.5808×10^1	$1.00899541677406 \times 10^0$ $1.12166438866319 \times 10^2$ $1.00778322917531 \times 10^0$	2.4923×10^0	$1.00899541677406 \times 10^0$ $1.12166438866919 \times 10^2$ $1.00778323005531 \times 10^0$	$1.00899541663406 \times 10^0$ $1.12166438866253 \times 10^2$ $1.00778322906531 \times 10^0$
180	1.7041×10^1	$1.01976347573729 \times 10^0$ $5.15976132302183 \times 10^1$ $1.01698577896417 \times 10^0$	2.4923×10^0	$1.01976348015729 \times 10^0$ $5.15976132362183 \times 10^1$ $1.01698577936417 \times 10^0$	$1.01976347253729 \times 10^0$ $5.15976132294753 \times 10^1$ $1.01698577895637 \times 10^0$
210	1.5269×10^1	$1.04398508855047 \times 10^0$	2.0956×10^2	$1.04398508915047 \times 10^0$	$1.04398508852747 \times 10^0$

Table 4 continued

t	NPROS4(4)		MPROW		Exact solution $[y_1, y_2, y_3]^T$
	Adaptive h	$[y_1, y_2, y_3]^T$	Adaptive h	$[y_1, y_2, y_3]^T$	
240	1.1792×10^1	$2.37344202758962 \times 10^0$	2.4196×10^2	$2.37344202798962 \times 10^0$	$2.37344202753152 \times 10^0$
		$1.03769184359052 \times 10^0$		$1.03769184369052 \times 10^0$	$1.03769184354452 \times 10^0$
		$1.10084907316792 \times 10^0$		$1.10084913316792 \times 10^0$	$1.10084907166792 \times 10^0$
		$1.09153380547232 \times 10^1$		$1.09153380551232 \times 10^1$	$1.09153380546902 \times 10^1$
		$1.08583197445086 \times 10^0$		$1.08583199027486 \times 10^0$	$1.08583196981086 \times 10^0$
270	7.0698×10^1	$1.24910213002145 \times 10^0$	2.7186×10^3	$1.24910213004387 \times 10^0$	$1.24910213002057 \times 10^0$
		$5.01394517886144 \times 10^0$		$5.01394517912644 \times 10^0$	$5.01394517860544 \times 10^0$
		$1.20832662924787 \times 10^0$		$1.20832663924787 \times 10^0$	$1.20832662623787 \times 10^0$
		$1.77972475593701 \times 10^0$		$1.77972515193701 \times 10^0$	$1.77972475193701 \times 10^0$
		$2.28185238867240 \times 10^0$		$2.28185239205540 \times 10^0$	$2.28185238554240 \times 10^0$
330	2.2559×10^1	$1.61375402423872 \times 10^0$	3.3004×10^2	$1.61375402437842 \times 10^0$	$1.61375402367172 \times 10^0$
		$1.00088932690780 \times 10^0$		$1.00088932691454 \times 10^0$	$1.00088932690350 \times 10^0$
		$1.12543858574889 \times 10^3$		$1.12543858577861 \times 10^3$	$1.12543858574659 \times 10^3$
		$1.64104948377717 \times 10^4$		$1.64104948377716 \times 10^4$	$1.64104948377716 \times 10^4$
		$1.00081487088552 \times 10^0$		$1.00081487123952 \times 10^0$	$1.00081487031852 \times 10^0$
360	2.3705×10^1	$1.22817852154989 \times 10^3$	1.6044×10^{-2}	$1.22817852154989 \times 10^3$	$1.22817852154988 \times 10^3$
		$1.32055494284708 \times 10^2$		$1.32055494284728 \times 10^2$	$1.32055494284651 \times 10^2$

$$\text{Efficiency of the new algorithm} = \frac{t_s}{4t_p} = \frac{s \times O(s^3)}{s \times O(s^3)} = 1.$$

Hence perfect speed-up is achievable for this method where $p = s = 4$.

Conclusion

We have presented a new approach for constructing an A-stable parallel fourth-order four stage Rosenbrock method and examined its convergence and stability aspects. The proposed approach allows us to derive the order conditions for high-order methods in a systematic and simple way. LTE is derived. Numerical experimentation has been performed which shows the efficiency of the method. It is concluded from the numerical experimentation considered in this paper that NPROS4(4) outperforms well for computing the solution of strongly stiff system of non-linear differential equations with less error as compared to the method of MPROW. The present new method can further be generalized to a parallel machine of 2^n processors, for any n .

References

1. Chan, Y.N.I., Birnbaum, I., Lapidus, L.: Solution of stiff differential equations and the use of Imbedding techniques. *Ind. Eng. Chem. Fundam.* **3**, 133–148 (1978)
2. Verwer, J.G., Spee, E.J., Blom, J.G., Hundsdorfer, W.: A second order Rosenbrock method applied to photo-chemical dispersion problems. *SIAM. J. Sci. Comput.* **20**, 1456–1480 (1999)
3. Sandu, A., Verwer, J.G., van Loon, M., Carmichael, G.R., Potra, F.A., Dabdub, D., Seinfeld, J.H.: Benchmarking stiff ODE solvers for atmospheric chemistry problems I: implicit versus explicit. *Atmos. Environ.* **31**, 3151–3166 (1997)
4. Ponalagusamy, R., Alphonse, P.J.A., Chandru, M.: Development of new fifth-order fifth-stage Runge Kutta method based on heronian mean. *Int. J. Eng. Sci. Adv. Comput. Biotechnol.* **2**, 162–197 (2011)
5. Rosenbrock, H.H.: Some general processes for the numerical solution of differential equations. *Comput. J.* **5**, 329–330 (1963)
6. Hairer, E., Wanner, G.: *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, 2nd revised edn. Springer Series in Computational Mathematics, Berlin (1996)
7. Tambue, A., Berre, I., Nordboten, J.M.: Efficient simulation of geothermal processes in heterogeneous porous media based on the exponential Rosenbrock–Euler and Rosenbrock type methods. *Adv. Water Resour.* **53**, 250–262 (2013)
8. Schippmann, B., Burchard, H.: Rosenbrock methods in biogeochemical modelling—a comparison to Runge–Kutta methods and modified Patankar schemes. *Ocean Model.* **37**, 112–121 (2011)
9. Berardi, M.: Rosenbrock-type methods applied to discontinuous differential systems. *Math. Comput. Simulat.* **95**, 229–243 (2013)
10. Rang, J.: An analysis of the Prothero–Robinson example for constructing new DIRK and ROW methods. *J. Comput. Appl. Math.* **262**, 105–114 (2013)
11. Calahan, D.A.: A stable, accurate method of numerical integration for nonlinear systems. *Proc. IEEE* **55**, 2016–2017 (1968)
12. Cash, J.R.: A semi-implicit Runge–Kutta formula for the integration of stiff systems of ordinary differential equations. *Chem. Eng. J.* **20**, 219–224 (1980)
13. Norsett, S.P., Wolfbrandt, A.: Order conditions for Rosenbrock-type methods. *Numer. Math.* **32**, 1–15 (1979)
14. Kaps, P., Rentrop, P.: Generalized Runge–Kutta methods of order four with stepsize control for stiff ordinary differential equations. *Numer. Math.* **33**, 55–68 (1979)
15. Kaps, P., Wanner, G.: A study of Rosenbrock-type methods of high order. *Numer. Math.* **38**, 279–298 (1981)
16. Shampine, L.F.: Implementation of Rosenbrock methods. *ACM Trans. Math. Softw.* **8**, 93–113 (1982)
17. Kaps, P., Poon, S.W.H., Bui, T.D.: Rosenbrock methods for stiff ODEs: a comparison of Richardson extrapolation and embedding technique. *Computing* **34**, 17–40 (1985)

18. Voss, D.A., Khaliq, A.Q.M.: Parallel Rosenbrock methods for chemical systems. *Comput. Chem.* **25**, 101–107 (2001)
19. Voss, D.A.: Fourth-Order Parallel Rosenbrock Formulae for Stiff Systems. *Mathematical and Computer Modelling* **40**, 1193–1198 (2004)
20. Chen, L.R., Liu, D.G.: A class of parallel Rosenbrock formulas. In: *Proceedings of 3rd BICSC'95, Beijing*, pp. 89–92, 1995
21. Cao, X.N., Li, S.F., Liu, D.G.: Modified parallel Rosenbrock methods for stiff differential equations. *J. Comput. Math.* **20**, 23–34 (2002)
22. Jain, M.K., Iyengar, S.R.K., Jain, R.K.: *Numerical Methods for Scientific and Engineering Computation*. Wiley, New Delhi (1985)
23. Butcher, J.C.: *The Numerical Analysis of Ordinary Differential Equations*, 2nd revised edn. Wiley, Chichester (1987)
24. Butcher, J.C.: Trees, B-series and exponential integrators. *IMA J. Numer. Anal.* **30**, 131–140 (2009)
25. Butcher, J.C.: Trees and numerical methods for ordinary differential equations. *Numer. Algorithms* **53**, 153–170 (2010)
26. Verwer, J.G.: Internal S-Stability for Generalized Runge–Kutta Methods. Report NW21/75. Mathematisch Centrum, Amsterdam (1975)
27. Hairer, E., Norsett, S.P., Wanner, G.: *Solving ordinary Differential Equations I, Nonstiff problems*, 2nd revised edn. Springer Series in Computational Mathematics, Switzerland (1993)
28. Lotkin, M.: On the accuracy of RK methods. *MYAC* **5**, 128–132 (1951)
29. Li, S.F.: *Theory of Computational Methods for Stiff Differential Equations*. Human Science and Technology Press (1997)
30. Edsberg, L.: *Integration Package for Chemical Kinetics*. Plenum Press, New York (1974)
31. Mazzia, F., Iavernaro, F.: Test Set for Initial Value Problem Solvers. Department of Mathematics, University of Bari (2003). http://www.dm.uniba.it/_testset
32. Field, R.J., Koros, E., Noyes, R.M.: Oscillation in chemical systems, Part 2. Thorough analysis of temporal oscillations in the bromated-cerium–malonic acid system. *J. Am. Soc.* **94**, 8649–8664 (1972)
33. Gray, C.: An analysis of the Belousov–Zhabotinskii reaction. *Rose-Hulman Undergrad. Math. J.* **3**(1), 36–47 (2002)
34. Mazzia, F., Magherini, C.: Test Set for Initial Value Problem Solvers. Department of Mathematics, University of Bari and INdAM, Research Unit of Bari (2008). http://www.dm.uniba.it/_testset