Numer Algor (2014) 65:687–703
DOI 10.1007/s11075-014-9823-2

ORIGINAL PAPER

Two-derivative Runge–Kutta methods for PDEs using a novel discretization approach

Angela Y. J. Tsai · Robert P. K. Chan · Shixiao Wang

Received: 27 April 2013 / Accepted: 2 January 2014 / Published online: 17 January 2014

Abstract We develop a novel and general approach to the discretization of partial differential equations. This approach overcomes the rigid restriction of the traditional method of lines (MOL) and provides flexibility in the treatment of spatial discretization. This method is essential for developing efficient numerical schemes for PDEs based on two-derivative Runge–Kutta (TDRK) methods, where the first and second derivatives must be discretized in an efficient way. This is unlikely to be achieved by using MOL. We then apply the explicit TDRK methods to the advection equations and analyze the numerical stability in the linear advection equation case. We conduct numerical experiments on the Burgers' equation using the TDRK methods developed. We also apply a two-stage semi-implicit TDRK method of order-4 and stage-order-4 to the heat equation. A very significant improvement in the efficiency of this TDRK method is observed when compared to the popular Crank-Nicolson method. This paper is partially based on the work in Tsai's PhD thesis (2011) [10].

Keywords Two-derivative Runge–Kutta methods · PDE methods · Stability region · Advection equation · Heat equation

1 Introduction to two-derivative Runge–Kutta methods

We have discussed the derivation of *two-derivative Runge-Kutta (TDRK)* methods in [4], hence only the basic information on TDRK methods is shown in this section for convenience of reference.

A. Y. J. Tsai · R. P. K. Chan (✉) · S. Wang
Department of Mathematics, University of Auckland, 38 Princes Street, Auckland 1142, New Zealand
e-mail: symmetrizer@gmail.com

We consider an initial-value problem,

$$y' = f(y), \qquad y(x_0) = y_0,$$

with $f : \mathbb{R}^N \to \mathbb{R}^N$ and assume that the second derivative is also known,

$$y'' = g(y) := f'(y)f(y), \qquad g : \mathbb{R}^N \to \mathbb{R}^N.$$

An s-stage TDRK method for the step $y_n \mapsto y_{n+1}$ applied with stepsize h is defined by

$$\begin{aligned} Y_i &= y_n + h\sum_{j=1}^{s} a_{ij} f(Y_j) + h^2 \sum_{j=1}^{s} \widehat{a}_{ij} g(Y_j), \quad i = 1, \dots, s, \\ y_{n+1} &= y_n + h\sum_{i=1}^{s} b_i f(Y_i) + h^2 \sum_{i=1}^{s} \widehat{b}_i g(Y_i). \end{aligned}$$

It is convenient to rewrite the defining equations in block-matrix form,

$$\begin{aligned} Y &= e \otimes y_n + h(A \otimes I_N)F(Y) + h^2(\widehat{A} \otimes I_N)G(Y), \\ y_{n+1} &= y_n + h(b^T \otimes I_N)F(Y) + h^2(\widehat{b}^T \otimes I_N)G(Y), \end{aligned}$$

where e is the $s \times 1$ vector of units, and the block vectors in $\mathbb{R}^{sN}$ are defined by

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_s \end{bmatrix}, \quad F(Y) = \begin{bmatrix} f(Y_1) \\ \vdots \\ f(Y_s) \end{bmatrix}, \quad G(Y) = \begin{bmatrix} g(Y_1) \\ \vdots \\ g(Y_s) \end{bmatrix}.$$

The coefficients of the method can be displayed in an extended Butcher tableau

$$\begin{array}{c|c||c} c & A & \widehat{A} \\ \hline & b^T & \widehat{b}^T \end{array}$$

where A, $\widehat{A}$ are $s \times s$ matrices, and b, $\widehat{b}$, c are $s \times 1$ vectors.

The TDRK methods used in the later sections of this paper are TDRK4, TDRK4a and TDRK244sss; their coefficients are listed respectively from left to right as follows:

$$\begin{array}{c|cc||cc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & \frac{1}{8} & 0 \\ \hline & 1 & 0 & \frac{1}{6} & \frac{1}{3} \end{array}, \quad \begin{array}{c|ccc||ccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & \frac{3}{8} & \frac{1}{8} & 0 & 0 & 0 & 0 \\ \hline & \frac{1}{6} & \frac{1}{6} & \frac{2}{3} & 0 & 0 & 0 \end{array}, \quad \begin{array}{c|cc||cc} 0 & 0 & 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{12} & -\frac{1}{12} \\ \hline & \frac{1}{2} & \frac{1}{2} & \frac{1}{12} & -\frac{1}{12} \end{array}. \tag{1}$$

TDRK4 and TDRK4a are explicit order-4 methods while TDRK244sss is a 2-stage, order-4, stage-order-4, semi-implicit method which is also symmetric and stiffly accurate [4, 10].

2 A novel semi-discretization method for PDEs

In order to apply the TDRK methods to PDEs (partial differential equations), we develop a semi-discretization method in this section based on the method of Rothe

[5, 8]. The aim is two-fold. Firstly, it provides a method to numerically evaluate the second derivative with respect to time t, as required in a TDRK method. Secondly, it provides a general approach to discretize the PDEs with more flexibility than the traditional MOL [6]. Following the method of Rothe, we consider discretizing the temporal variable t in the first step by Runge-Kutta methods or TDRK method with error of order $O(\delta^{p+1})$, and leave the spatial variables to be discretized with error of order $O(h^q)$ in the next step. This guarantees a local error order $O(\delta^{p+1} + h^q)$. Meanwhile, the spatial discretization can be chosen in a more flexible way to meet stability and/or computational requirements. The approach is general and enables us to systematically apply ODE numerical methods to PDEs. In fact, many classical numerical schemes for PDEs can be considered as special cases of this general approach, for instance, the Lax-Wendroff scheme for the advection equation. This approach is valid for general PDEs. However, for simplicity, we consider only the PDEs with one spatial dimension. One may easily generalize the result to general PDEs. Letting $f(\eta)$ be a smooth function of η, we consider PDEs having the form

$$\frac{\partial u}{\partial t} = f(\mathcal{P}(u)), \tag{2}$$

where $\mathcal{P}(u)$ is a linear partial differential operator with constant coefficients such as

$$\mathcal{P}(u) = \frac{\partial}{\partial x}u \quad \text{and} \quad \mathcal{P}(u) = \frac{\partial^2}{\partial x^2}u.$$

We start by differentiating (2) with respect to t,

$$\begin{aligned}\frac{\partial^2 u}{\partial t^2} &= \frac{\partial}{\partial t} f(\mathcal{P}(u)) = f_\eta(\mathcal{P}(u))\frac{\partial}{\partial t}\mathcal{P}(u)\\ &= f_\eta(\mathcal{P}(u))\mathcal{P}\left(\frac{\partial u}{\partial t}\right) = f_\eta(\mathcal{P}(u))\mathcal{P}(f(\mathcal{P}(u))),\end{aligned}$$

or, written in a compact form,

$$\frac{\partial^2 u}{\partial t^2} = f_\eta \mathcal{P}(f). \tag{3}$$

Notice that $\frac{\partial}{\partial t}$ commutes with $\mathcal{P}$ $\left(\frac{\partial}{\partial t}\mathcal{P} = \mathcal{P}\frac{\partial}{\partial t}\right)$ and the chain rule plays a basic role in the derivation of (3). This can be compared with the similar formula for ODEs, $y'(t) = f(y)$,

$$\frac{d^2 y}{dt^2} = f_y f, \tag{4}$$

by using the chain rule. It is therefore clear that (4) differs from (3) only in the additional linear operator $\mathcal{P}$.

This procedure can be carried out for higher derivatives, for example,

$$\begin{aligned}\frac{\partial^3 u}{\partial t^3} &= \frac{\partial}{\partial t} f_\eta \mathcal{P}(f) = f_{\eta\eta}\mathcal{P}(f)\mathcal{P}(f) + f_\eta\mathcal{P}(f_\eta\mathcal{P}(f)),\\ \frac{\partial^4 u}{\partial t^4} &= \frac{\partial}{\partial t}(f_{\eta\eta}\mathcal{P}(f)\mathcal{P}(f) + f_\eta\mathcal{P}(f_\eta\mathcal{P}(f)))\\ &= f_{\eta\eta\eta}\mathcal{P}(f)\mathcal{P}(f)\mathcal{P}(f) + 3f_{\eta\eta}\mathcal{P}(f)\mathcal{P}(f_\eta\mathcal{P}(f))\\ &\quad + f_\eta\mathcal{P}(f_{\eta\eta}\mathcal{P}(f)\mathcal{P}(f)) + f_\eta\mathcal{P}(f_\eta\mathcal{P}(f_\eta\mathcal{P}(f))),\end{aligned}$$

and so on. The above formulae are similar to the counterparts in ODE theory. In fact, we can recover the elementary differential formulae for the scalar ODEs by simply removing the operator $\mathcal{P}$ (also the associated bracket) from above formulae. We then have the following theorem.

Theorem 1 *The n^{th} derivative of the exact solution is given by*

$$\frac{\partial^n u}{\partial t^n} = \sum_{|\tau|=n} \overline{\alpha}(\tau)\mathcal{F}(\tau)(u), \tag{5}$$

where

$$\mathcal{F}(\tau) = f^{(m)}\mathcal{P}(\mathcal{F}(\tau_1)) \cdots \mathcal{P}(\mathcal{F}(\tau_m)), \quad \textit{for } \tau = [\tau_1, \ldots, \tau_m],$$

and $\overline{\alpha}(\tau)$ is the number of ways of labelling τ with an ordered set.

The proof is exactly the same as in the ODE theory based on Butcher's tree theory [1–3]. Theorem 1 has a useful corollary:

Corollary 1 *For the linear PDEs of the form,*

$$\frac{\partial u}{\partial t} = \mathcal{P}(u),$$

Equation (5) reduces to

$$\frac{\partial^n u}{\partial t^n} = \mathcal{P}^n(u). \tag{6}$$

Using exactly the same approach, we can derive the elementary differential theorem and the order conditions using Butcher's tree theory. This observation leads to the conclusion that *the numerical methods for ODEs including RK methods and the TDRK methods are directly applicable to the discretization of the PDEs (2) with the time variable t.*

This approach differs from the classical MOL, where a PDE is first discretized in space and then an ODE numerical method is applied to solve the semi-discretized ODE system. To distinguish the current approach from MOL, we call the current approach MMOL (modified method of lines). MMOL can be considered as a direct generalization of MOL in the sense that all PDE schemes based on MOL can be obtained by MMOL as a special case but not vice versa. In fact, the numerical scheme derived by MOL corresponds only to a single numerical scheme by using one special spatial discretization in MMOL. Many additional numerical schemes can be obtained only by MMOL with using various different spatial discretization methods. The multiplicity of selection of the spatial discretization in MMOL would, no doubt, create opportunities to improve the numerical scheme obtained by MOL. It is therefore a necessary task to explore all possible numerical schemes based on MMOL in a systematical way, in order to achieve an optimal performance among them. This is particularly important in TDRK methods, where the numerical scheme for the second temporal derivative is required as an input; when the MOL approach is used the scheme for the second derivative is rigidly determined by the first step spatial discretization. This rigidity may directly result in poor performance schemes or unstable schemes in some PDEs. MMOL effectively removes this rigidity and provides flexibility in constructing a numerical PDE scheme and thus achieve optimal performance.

Example 1 To illustrate the advantage of the new approach, we consider the model **advection/wave equation**,

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0. \tag{7}$$

We interpret the well-known Lax-Wendroff scheme as a TDRK method with a suitable numerical scheme for the second time derivative used in the MMOL approach. Let $u_n(x)$ be the numerical approximation to the exact solution $u(x, t_n)$ at time t_n. We apply the explicit 2-stage TDRK method,

$$\begin{array}{c|cc||cc} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ \hline & 1 & 0 & \frac{1}{2} & 0 \end{array},$$

with timestep $\delta = t_{n+1} - t_n$ to the advection (7) and get

$$\begin{aligned} Y_1 &= u_n(x), \\ Y_2 &= u_n(x) - a\delta(u_n(x))_x, \\ u_{n+1}(x) &= u_n(x) - a\delta(u_n(x))_x + \frac{a^2\delta^2}{2}(u_n(x))_{xx}. \end{aligned} \tag{8}$$

A full discretization of the advection equation is obtained by applying a suitable spatial discretization for variable x. It can be shown that if the following spatial descretizations

$$\begin{aligned} (u_n(x_j))_x &\approx \frac{u_n(x_{j+1}) - u_n(x_{j-1})}{2\Delta x}, \\ (u_n(x_j))_{xx} &\approx \frac{u_n(x_{j+2}) - 2u_n(x_j) + u_n(x_{j-2})}{4\Delta x^2}, \end{aligned}$$

are used, the resulting numerical scheme is equivalent to the one derived using the RK2 method with MOL approach, which is a well-known unstable scheme. However, the flexibility of the current approach allows us to select a more compact scheme, using the neighbouring grid points, to approximate u_{xx} in (8),

$$(u_n(x_j))_{xx} \approx \frac{u_n(x_{j+1}) - 2u_n(x_j) + u_n(x_{j-1})}{\Delta x^2}.$$

This leads to the following numerical scheme

$$U_j^{n+1} = U_j^n - a\delta\frac{U_{j+1}^n - U_{j-1}^n}{2\Delta x} + a^2\delta^2\frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{2\Delta x^2}, \tag{9}$$

where $U_j^n \approx u(x_j, t_n)$, the numerical solution at the grid point (x_j, t_n). This is a stable numerical scheme with second-order temporal and spatial accuracy. In fact, this is the well-known Lax-Wendroff scheme. Our novel approach sheds new light on the relationship of the RK method and the Lax-Wendroff approach, and possibly opens more applications of RK methods to PDEs along this line in the future.

This example shows that the numerical scheme for the second derivative is determined by that of the first derivative, resulting in a loose spread-out numerical scheme for the second derivative. This is a common flaw when the MOL approach is used in TDRK method. It is therefore necessary to take advantage of the flexibility of MMOL in applying TDRK methods to PDEs.

3 TDRK method for advection equations

3.1 A case study for the stability of TDRK methods applied to advection equations

We study the numerical stability of a TDRK method when applied to the model advection (7) on the interval $(0, 1)$ with spatially periodic boundary condition $u(0, t) = u(1, t)$ imposed. We use the explicit TDRK4a method (1) for demonstration in this study. To proceed, we find the second derivative using Corollary 1 and (7) as

$$\frac{\partial^2 u}{\partial t^2} = a^2\mathcal{P}^2(u) = a^2\mathcal{D}(u), \tag{10}$$

where $\mathcal{D}(u) = u_{xx}$ to indicate that $\mathcal{D}$ must be treated as a whole in the spatial discretization. Applying the TDRK4a scheme, we obtain

$$\begin{aligned} Y_1 &= u_n(x),\\ Y_2 &= u_n(x) - a\delta(Y_1)_x + \frac{a^2\delta^2}{2}\mathcal{D}(Y_1),\\ Y_3 &= u_n(x) - \frac{3a\delta}{8}(Y_1)_x - \frac{a\delta}{8}(Y_2)_x,\\ u_{n+1}(x) &= u_n(x) - \frac{a\delta}{6}((Y_1)_x + (Y_2)_x + 4(Y_3)_x), \end{aligned} \tag{11}$$

or, in terms of $\mathcal{D}$,

$$\begin{aligned} u_{n+1}(x) &= u_n(x) - a\delta(u_n(x))_x + \frac{a^2\delta^2}{2}(u_n(x))_{xx}\\ &\quad - \frac{a^3\delta^3}{12}((\mathcal{D}(u_n(x)))_x + (u_n(x))_{xxx}) + \frac{a^4\delta^4}{24}(\mathcal{D}(u_n(x))_{xx}. \end{aligned} \tag{12}$$

To obtain a full discretization of the advection equation we use the following grid points on the interval $[0, 1]$ for the spatial discretization,

$$x_j = jh, \quad j = 0, 1, \ldots, N, N+1, \quad h = \Delta x = 1/(N+1),$$

and the following fourth-order spatial discretizations for u_x and u_{xx}, respectively.

$$
\begin{aligned}
u_x(x_j,t_n) &= \frac{1}{h}\left(\frac{1}{12}u(x_{j-2},t_n) - \frac{2}{3}u(x_{j-1},t_n)\right.\\
&\qquad \left. +\frac{2}{3}u(x_{j+1},t_n) - \frac{1}{12}u(x_{j+2},t_n)\right) + O(h^4), \qquad (13)\\
u_{xx}(x_j,t_n) &= \frac{1}{h^2}\left(-\frac{1}{12}u(x_{j-2},t_n) + \frac{4}{3}u(x_{j-1},t_n)\right.\\
&\qquad \left. -\frac{5}{2}u(x_j,t_n) + \frac{4}{3}u(x_{j+1},t_n) - \frac{1}{12}u(x_{j+2},t_n)\right) + O(h^4). \qquad (14)
\end{aligned}
$$

Note that (14) is not derived from (13) using the MOL, which will have given us a much less compact scheme than (14). Let $U_j^n \approx u(x_j,t_n)$, the numerical approximation to the solution at the gridpoint (x_j,t_n), and let $\mathbf{U}^n = \left[U_1^n, U_2^n, \ldots, U_N^n\right]^T$. Inserting (13) and (14) into (11) (or (12)), after some algebraic manipulation, we have

$$\mathbf{U}^{n+1} = \left[I - a\delta A_h + \frac{a^2\delta^2}{2}A_h^2 - \frac{a^3\delta^3}{12}\left(A_h^3 + A_h D_h\right) + \frac{a^4\delta^4}{24}\left(A_h^2 D_h\right)\right]\mathbf{U}^n,$$

where A_h and D_h are the finite difference matrices with size $N \times N$ for the first and second derivatives respectively,

$$A_h = \frac{1}{h}\begin{bmatrix} 0 & \frac{2}{3} & -\frac{1}{12} & & \frac{1}{12} & -\frac{2}{3} \\ -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & & \frac{1}{12} \\ \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & \\ & \ddots & \ddots & \ddots & \ddots & \ddots \\ -\frac{1}{12} & & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} \\ \frac{2}{3} & -\frac{1}{12} & & \frac{1}{12} & -\frac{2}{3} & 0 \end{bmatrix}.$$

and

$$D_h = \frac{1}{h^2}\begin{bmatrix} -\frac{5}{2} & \frac{4}{3} & -\frac{1}{12} & & -\frac{1}{12} & \frac{4}{3} \\ \frac{4}{3} & -\frac{5}{2} & \frac{4}{3} & -\frac{1}{12} & & -\frac{1}{12} \\ -\frac{1}{12} & \frac{4}{3} & -\frac{5}{2} & \frac{4}{3} & -\frac{1}{12} & \\ & \ddots & \ddots & \ddots & \ddots & \ddots \\ -\frac{1}{12} & & -\frac{1}{12} & \frac{4}{3} & -\frac{5}{2} & \frac{4}{3} \\ \frac{4}{3} & -\frac{1}{12} & & -\frac{1}{12} & \frac{4}{3} & -\frac{5}{2} \end{bmatrix}.$$

A_h and D_h are circular matrices because $u(0,t) = u(1,t)$ [6]. The eigenvalues of A_h and D_h are found to be

$$\lambda_j(h) = \frac{i}{3h}\sin(2\pi jh)(4 - \cos(2\pi jh)), \quad \text{for } A_h, \qquad (15)$$

$$\overline{\lambda}_j(h) = -\frac{16}{3h^2}\sin^2(\pi jh)\left(1 - \frac{1}{4}\cos^2(\pi jh)\right), \quad \text{for } D_h. \qquad (16)$$

Stability requires

$$\left|1 - a\delta\lambda_j + \frac{a^2\delta^2}{2}\lambda_j^2 - \frac{a^3\delta^3}{12}\left(\lambda_j^3 + \lambda_j\overline{\lambda}_j\right) + \frac{a^4\delta^4}{24}\left(\lambda_j^2\overline{\lambda}_j\right)\right| \le 1, \qquad (17)$$

Inserting (15) and (16) into (17) and through a numerical estimate, we obtain a stability (CFL) range:

$$\frac{|a|\delta}{h} < 1.5.$$

This shows that the TDRK4a method exhibits a similar stability behaviour to RK4, the classical Runge–Kutta method of order 4, when applied to the advection equation. The same stability analysis can also be applied to TDRK4 in (1) and leads to a similar stability result.

3.2 Numerical experiments on the Burgers' equation

We consider the inviscid Burgers' equation

$$\frac{\partial u}{\partial t} + \frac{\partial (u^2)}{\partial x} = 0. \tag{18}$$

The second derivative is found as

$$u_{tt} = 8uu_x^2 + 4u^2 u_{xx} = 8uu_x^2 + 4u^2\mathcal{D}(u),$$

where $\mathcal{D}(u) = u_{xx}$ as in (10). Applying the TDRK4a method to (18), the internal stage values are, for timestep δ,

$$\begin{aligned}
Y_1 &= u_n(x), \\
Y_2 &= u_n(x) - \delta\left(Y_1^2\right)_x + \frac{\delta^2}{2}\left(8Y_1((Y_1)_x)^2 + 4Y_1^2\mathcal{D}(Y_1)\right), \\
Y_3 &= u_n(x) - \frac{3\delta}{8}\left(Y_1^2\right)_x - \frac{\delta}{8}\left(Y_2^2\right)_x, \\
u_{n+1}(x) &= u_n(x) - \frac{\delta}{6}\left(\left(Y_1^2\right)_x + \left(Y_2^2\right)_x + 4\left(Y_3^2\right)_x\right).
\end{aligned} \tag{19}$$

Similarly, applying the TDRK4 method to (18), the internal stage values are, for timestep δ,

$$\begin{aligned}
Y_1 &= u_n(x), \\
Y_2 &= u_n(x) - \frac{\delta}{2}\left(Y_1^2\right)_x + \frac{\delta^2}{8}\left(8Y_1((Y_1)_x)^2 + 4Y_1^2\mathcal{D}(Y_1)\right), \\
u_{n+1}(x) &= u_n(x) - \delta\left(Y_1^2\right)_x + \frac{\delta^2}{6}\left(8Y_1((Y_1)_x)^2 + 4Y_1^2\mathcal{D}(Y_1)\right. \\
&\quad \left. +2\left(8Y_2((Y_2)_x)^2 + 4Y_2^2\mathcal{D}(Y_2)\right)\right).
\end{aligned} \tag{20}$$

A full discretization of the Burgers' equation can then be readily obtained by using sufficiently high order spatial discretizations for u_x and $\mathcal{D}(u)$. In our implementation of (19), the spatial discretization is performed using the sixth-order compact scheme described by Lele [7]. See Appendix A for details. The high order spatial resolution would allow us to capture the temporal error trend. This is important for an explicit method because the CFL condition imposes a severe restriction on the time stepsize δ, which is of the same order as the spatial stepsize h, and the spatial error would overwhelm the temporal error with a low order spatial numerical scheme.

We conduct numerical experiments on Burgers' (18) with the boundary and initial values,

$$u(0,t) = u(\pi,t), \qquad u(x,0) = \frac{\sin(2x)}{2}.$$

This Burgers' equation can be solved using the method of characteristics and the solution is given in parametric form:

$$u = \frac{\sin 2x_0}{2}, \qquad x = \sin(2x_0)t + x_0,$$

where $x_0 \in [0,\pi]$. The regular solution ceases to exist at time $t = 0.5$ and thereafter. A shock solution will occur after that time.

Figure 1 shows the solutions of Burgers' equation at time $t = 0, 0.21, 0.42$ and 0.48. The wave progressively steepens along time near the midpoint $x = \frac{\pi}{2}$.

Figure 2 shows the error behaviour of RK4 (MOL), TDRK4 (MMOL) and TDRK4a (MMOL) at position $x = 0.20\pi$ for various time points, $t = 0.21, 0.42$ and 0.48.

- At $t = 0.21$, the exact solution has a sine-like shape, as shown in Fig. 1. All three methods show similar behaviour, but RK4 performs slightly better than the other two.

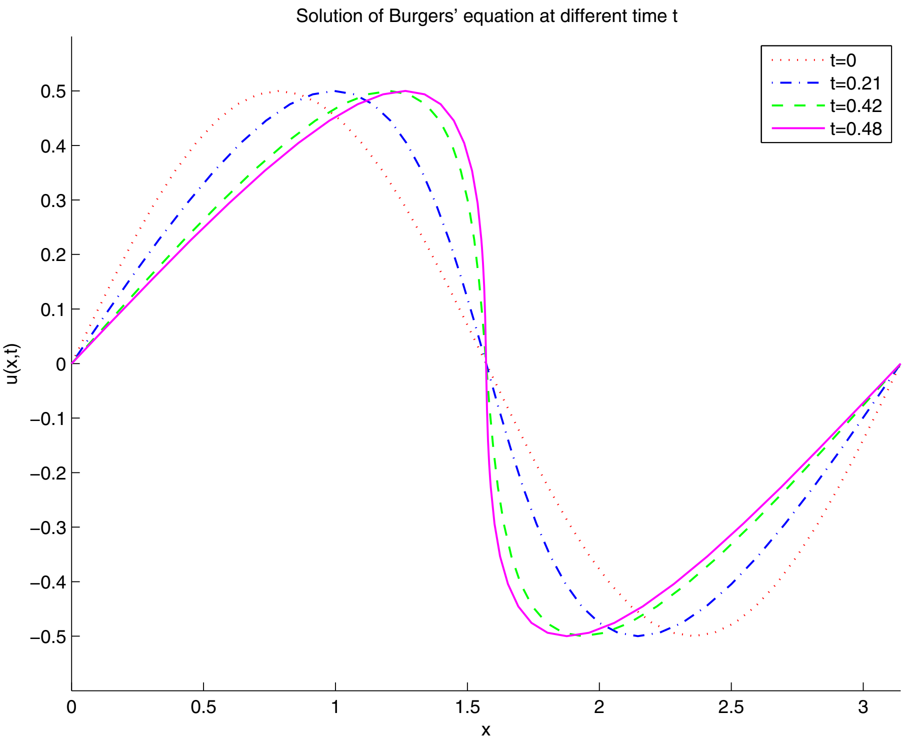

Fig. 1 Solution of Burgers' equation

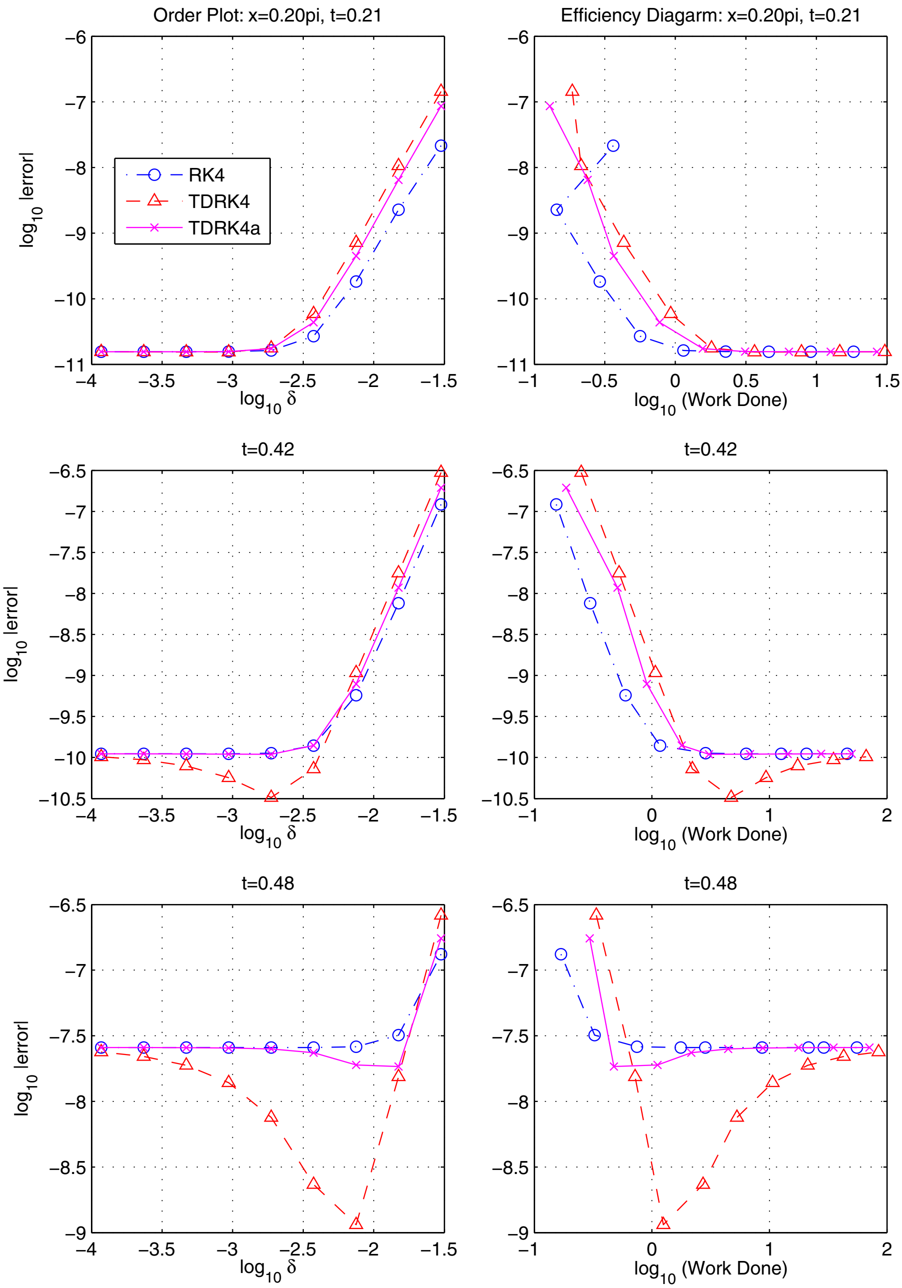

Fig. 2 Order plots and efficiency diagrams for Burgers' equation

- At $t = 0.42$, the solution has a steeper wave shape. TDRK4a has a similar error behaviour as RK4, but TDRK4 performs significantly better than these two methods.
- The wave is at the edge of breaking at $t = 0.48$, both TDRK4 and TDRK4a perform better than RK4. Here, TDRK4 is clearly a superior method over RK4 and TDRK4a. The reasons are in its treatment of nonlinear wave evolution and the singularity development. The scheme that TDRK4 uses to calculate the second derivatives, $\mathcal{D}$ in (20), is more compact than the scheme derived from the MOL. Hence it is more efficient in capturing the sharp wave shape. (In all figures work done is measured in CPU time). We notice that TDRK4 uses a more compact scheme for $\mathcal{D}$ than TDRK4a, and this explains why TDRK4 performs better than TDRK4a.

In Fig. 3, we observe a similar trend in the error behaviour of the methods at position $x = 0.45\pi$ for $t = 0.42$ and 0.48 as in Fig. 2 at position $x = 0.20\pi$.

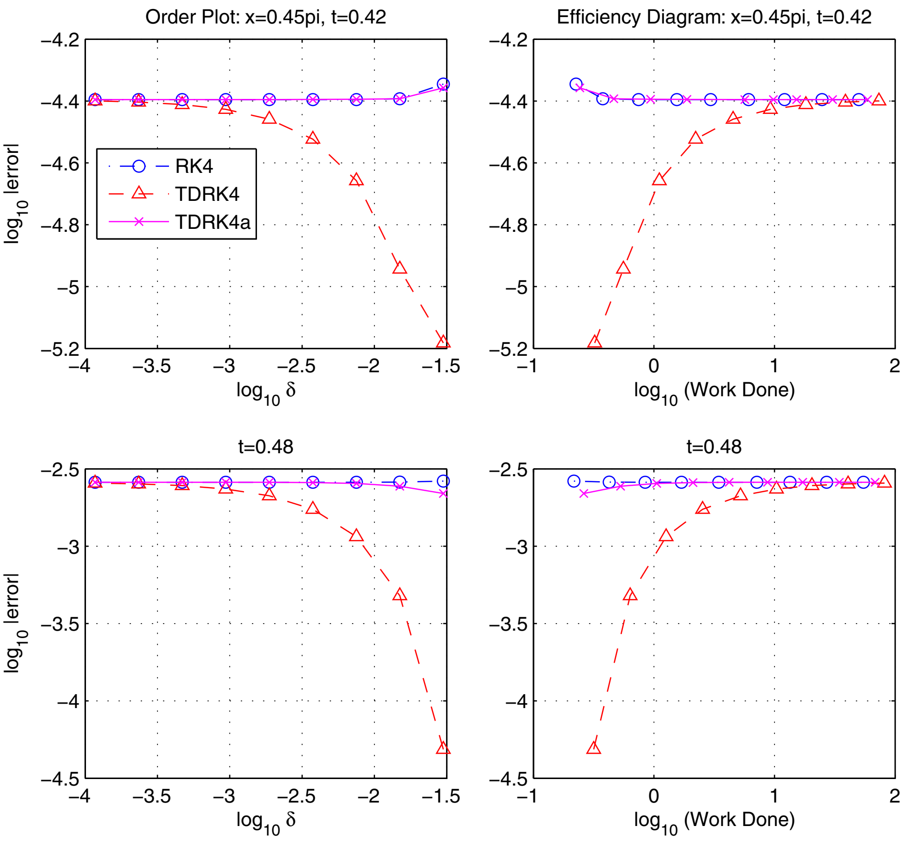

Fig. 3 Order plots and efficiency diagrams for Burgers' equation

4 The TDRK method for the heat equation

We conduct numerical experiments on the heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \tag{21}$$

with the fourth-order semi-implicit TDRK244sss method (1). Using Corollary 1, we have

$$\frac{\partial^2 u}{\partial t^2} = \mathcal{P}^2(u) = u_{xxxx}.$$

Let $u_n(x) \approx u(x, t_n)$ and the semi-discretization of the problem for timestep δ reads as:

$$\begin{aligned} u_{n+1}(x) &= u_n(x) + \frac{\delta}{2}((u_n(x))_{xx} + (u_{n+1}(x))_{xx}) \\ &\quad + \frac{\delta^2}{12}((u_n(x))_{xxxx} - (u_{n+1}(x))_{xxxx}), \end{aligned} \tag{22}$$

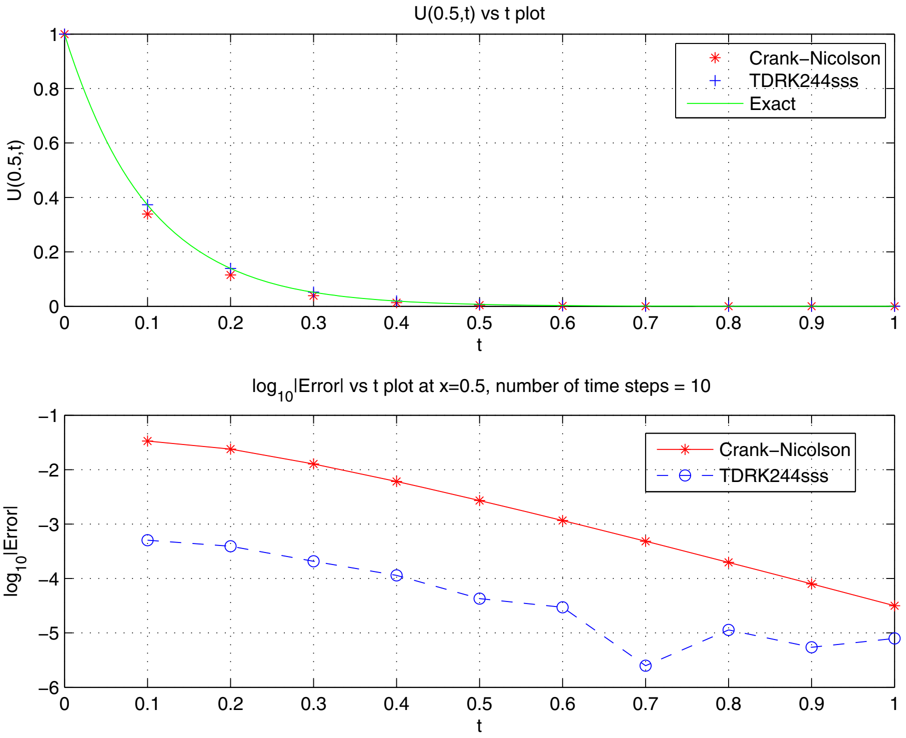

Fig. 4 The numerical solution vs the exact solution plot and the error behaviour plot at $x = 0.5$ for the heat equation

or

$$u_{n+1}(x) - \frac{\delta}{2}(u_{n+1}(x))_{xx} + \frac{\delta^2}{12}(u_{n+1}(x))_{xxxx} = \frac{\delta}{2}(u_n(x))_{xx} + \frac{\delta^2}{12}(u_n(x))_{xxxx} + u_n(x). \quad (23)$$

Let us consider the initial and boundary conditions

$$u(x,0) = s(x), \quad u(0,t) = p(t) \quad \text{and} \quad u(1,t) = q(t).$$

Note that (23) is a boundary value problem for a fourth order ODE, and, hence, needs four boundary conditions. The other two conditions, derived from the heat equation, are

$$u_{xx}(0,t) = p'(t) \quad \text{and} \quad u_{xx}(1,t) = q'(t).$$

A full-discretization of (21) can be obtained by using the central differences

$$u_{xx}(x_j,t_n) = \frac{u(x_{j-1},t_n) - 2u(x_j,t_n) + u(x_{j+1},t_n)}{h^2} + O(h^2), \quad (24)$$

$$u_{xxxx}(x_j,t_n) = \frac{u(x_{j-2},t_n) - 4u(x_{j-1},t_n) + 6u(x_j,t_n) - 4u(x_{j+1},t_n) + u(x_{j+2},t_n)}{h^4} + O(h^2). \quad (25)$$

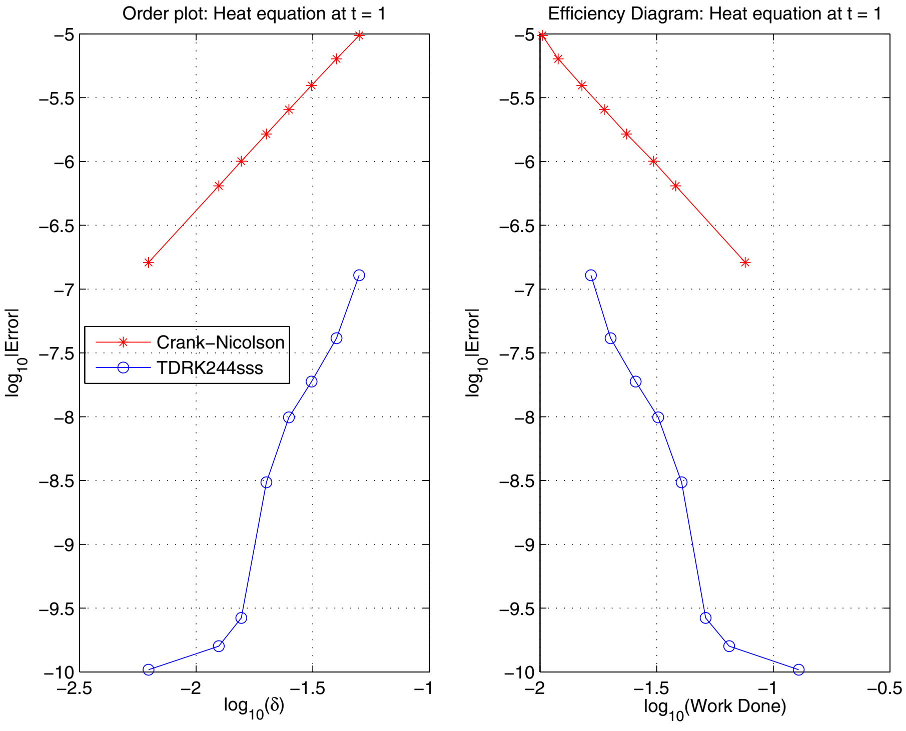

Fig. 5 Order plot and efficiency diagram for the heat equation at $x = 0.5$

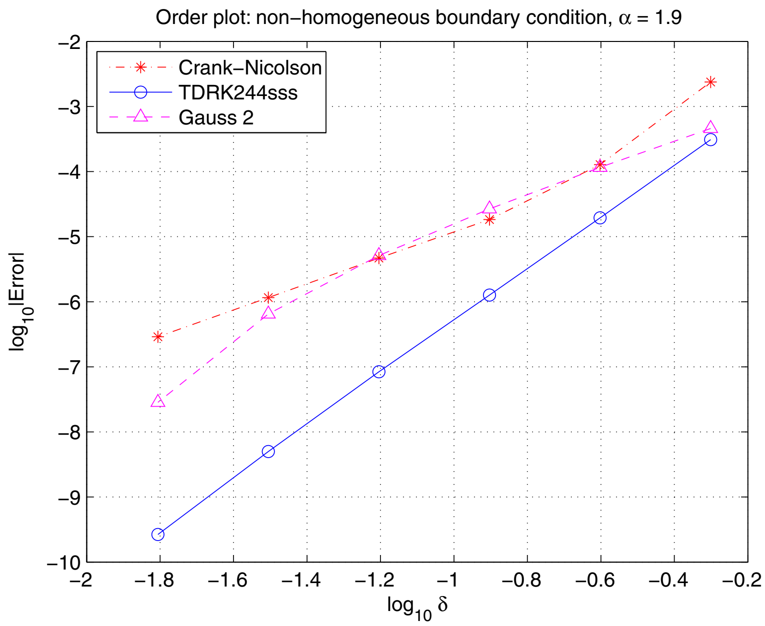

Fig. 6 Order plot for the heat equation at $x = 0.5$ and $t = 1$

The first (also the last) internal grid point uses a different scheme which accounts for the contribution of the boundary:

$$u_{xx}(x_1, t_n) = \frac{p(t_n) - 2u(x_1, t_n) + u(x_2, t_n)}{h^2} + O(h^2),$$
$$u_{xxxx}(x_1, t_n) = \frac{-2p(t_n) + 5u(x_1, t_n) - 4u(x_2, t_n) + u(x_3, t_n)}{h^4} + \frac{p'(t_n)}{h^2} + O(h^2).$$

The details of the resulting numerical scheme are described in Appendix B.

The numerical computation is carried out with the homogeneous boundary conditions $p(t) = q(t) = 0$, and initial value $s(x) = \sin(\pi x)$. An analytic solution $u = \sin(\pi x)\exp(-\pi^2 t)$ is available for this problem.

Figure 4 shows a plot of the exact solution against the numerical solutions using TDRK244sss and the Crank-Nicolson method, respectively, at $x = 0.5$ and the errors for time $t = 0.1(0.1)1.0$. Figure 5 compares the numerical result of TDRK244sss with that of the classical Crank-Nicolson method, which is based on the implicit trapezoidal rule and can be written as

$$u_{n+1}(x) = u_n(x) + \frac{\delta}{2}((u_n(x))_{xx} + (u_{n+1}(x))_{xx}),$$

where $u_n(x) \approx u(x, t_n)$. It is found that the TDRK244sss is far more efficient than the popular Crank-Nicolson method. Note that TDRK244sss holds its fourth order until the spatial error becomes dominant.

Notice that the implicit method, when it is applied to PDEs with relatively large temporal stepsize compared to the spatial stepsize, could exhibit order reduction.

However, order reduction behaviour does not occur for the current problem where the boundary condition is homogeneous. In [9], the authors show that order reduction does not occur for the advection equation with homogenous boundary conditions.

We conduct numerical experiments for non-homogeneous boundary condition, $p(t) = t^{\alpha}$ for $\alpha > 1$ and $f(x) = 0$. Figure 6 shows the error behaviour of Crank-Nicolson, 2-stage Gauss and TDRK244sss for $\alpha = 1.9$. The order reduction of the 2-stage Gauss method is clearly seen in this case. It behaves like a second order method similar to the Crank-Nicolson method. On the other hand, TDRK244sss preserves its order-4 behaviour, a demonstration of the benefit of higher stage order.

5 Conclusions

We have developed a novel approach, MMOL, for the treatment of discretization of PDEs. The approach is motivated by applying TDRK methods to PDEs, where the second derivative u_{tt} must be discretized by using an appropriate finite difference scheme. It turns out that the traditional MOL often leads to a rather 'loose' scheme. The MMOL overcomes this problem and allows choices of more compact finite difference schemes for u_{tt}. The advantage of using the compact scheme for u_{tt} was clearly observed in the numerical experiment on the Burgers' equation when applying the explicit TDRK methods to the equation.

The MMOL will provide a systematic way to apply the numerical ODE method to PDEs. It offers a more flexible treatment of the spatial discretization to meet stability and/or computational requirements. The Lax-Wendroff scheme (9) for the advection equation discussed in Section 2 is an example to demonstrate such applications. It is clear that many classical PDE schemes can be interpreted in the same way in terms of MMOL. Future work along this line, combining with TDRK methods, may lead to the discovery of more efficient numerical schemes for PDEs. Note that the stability region of individual numerical schemes obtained by using MMOL is generally different from those of MOL, depending on the specific spatial finite difference scheme used and shall be analyzed on a case by case basis. We have presented a typical stability analysis in Section 3 and this type of technique is applicable to other numerical schemes.

The advantage of TDRK methods is clearly demonstrated in the application of TDRK244sss to the heat equation. Firstly, TDRK244sss is a fourth-order, stage-order-four, semi-implicit method with two stages, which cannot be achieved by a RK method. The numerical implementation of the TDRK244sss to the heat equation has a low computation cost, incurring only twice the cost of the second-order Crank-Nicolson method.

Moreover, TDRK244sss, a stage-order-4 method, exhibits an excellent feature to overcome the phenomenon of order reduction as shown in Fig. 6. We will report more detailed results on the order reduction analysis of TDRK methods for PDEs in another paper.

Acknowledgments The authors wish to thank the referees for their valuable comments.

Appendix: A

In the following, the value of the numerical solution at the node (x_i, t_n) is denoted by $U_i = U_i^n \approx u(x_i, t_n)$. Note that n is omitted as we are only concerned with the spatial discretization here. The first derivative, u_x, is approximated by,

$$\widetilde{\alpha}U'_{i-1} + U'_i + \widetilde{\alpha}U'_{i+1} = \widetilde{b}\frac{U_{i+2} - U_{i-2}}{4h} + \widetilde{a}\frac{U_{i+1} - U_{i-1}}{2h}, \tag{26}$$

where $U'_i \approx u_x(x_i, t_n)$, $\widetilde{\alpha} = \frac{1}{3}$, $\widetilde{b} = \frac{1}{9}$ and $\widetilde{a} = \frac{14}{9}$.

The second derivative, $U''_j \approx u_{xx}(x_j, t_n)$ is approximated by

$$\widetilde{\alpha}U''_{j-1} + U''_j + \widetilde{\alpha}U''_{j+1} = \widetilde{b}\frac{U_{j+2} - 2U_j + U_{j-2}}{4h^2} + \widetilde{a}\frac{U_{j+1} - 2U_j + U_{j-1}}{h^2}, \tag{27}$$

with $\widetilde{\alpha} = \frac{2}{11}$, $\widetilde{b} = \frac{3}{11}$ and $\widetilde{a} = \frac{12}{11}$. This is an order-six scheme from Lele [7]. U'_i and U''_i for $i = 1, 2, \ldots, N$ are found by solving (26) and (27), respectively, for given U'_i, $i = 1, 2, \ldots, N$.

It shall be pointed out that the second derivative discretization is not derived from the first derivative using the MOL scheme. The latter generates a less compact scheme.

Appendix: B

We use the following grid points on the interval [0, 1] for the spatial discretization,

$$x_j = jh, \quad j = 0, 1, \ldots, N, N+1, \quad h = \Delta x = 1/(N+1).$$

Note that j starts from 0, and the two boundary points are x_0 and x_{N+1}. With the boundary values given at x_0 and x_{N+1}, we seek the numerical solutions for the internal grid points x_j where $j = 1, 2, \ldots, N$. Let U_j^n denote the approximation to $u(x_j, t_n)$. Inserting (24) and (25) into (22), we obtain

$$A_h \mathbf{U^{n+1}} = \mathbf{Z}, \tag{28}$$

where

$$A_h = \begin{bmatrix} \widetilde{a}_1 & \widetilde{b} & \widetilde{c} & & & & \\ \widetilde{b} & \widetilde{a} & \widetilde{b} & \widetilde{c} & & & \\ \widetilde{c} & \widetilde{b} & \widetilde{a} & \widetilde{b} & \widetilde{c} & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & \widetilde{c} & \widetilde{b} & \widetilde{a} & \widetilde{b} & \widetilde{c} \\ & & & \widetilde{c} & \widetilde{b} & \widetilde{a} & \widetilde{b} \\ & & & & \widetilde{c} & \widetilde{b} & \widetilde{a}_1 \end{bmatrix},$$

$$\widetilde{a} = 6r_2 + 2r_1 + 1, \quad \widetilde{b} = -4r_2 - r_1, \quad \widetilde{c} = r_2, \quad \widetilde{a}_1 = 1 + 2r_1 + 5r_2,$$

with $r_1 = \dfrac{\delta}{2h^2}$ and $r_2 = \dfrac{\delta^2}{12h^4}$. For the RHS of (28), $Z = [z_1, z_2, \ldots, z_N]^T$, where $z_j = r_2U^n_{j-2} - (4r_2 - r_1)U^n_{j-1} + (6r_2 - 2r_1 + 1)U^n_j - (4r_2 - r_1)U^n_{j+1} + r_2U^n_{j+2}$, for $j = 3, \ldots, N-2$; for $j = 1$ and 2, we have

$$
\begin{aligned}
z_1 &= (1 - 2r_1 + 5r_2)U^n_1 + (r_1 - 4r_2)U^n_2 + r_2U^n_3 + (r_1 - 2r_2)p(t_n) \\
&\quad + (r_1 + 2r_2)p(t_{n+1}) + \frac{r_1\delta}{6}(p'(t_n) - p'(t_{n+1})), \\
z_2 &= -(4r_2 - r_1)U^n_1 + (6r_2 - 2r_1 + 1)U^n_2 - (4r_2 - r_1)U^n_3 + r_2U^n_4 \\
&\quad + r_2(p(t_n) - p(t_{n+1})),
\end{aligned}
$$

and similar terms for z_{N-1} and z_N

$$
\begin{aligned}
z_{N-1} &= -(4r_2 - r_1)U^n_N + (6r_2 - 2r_1 + 1)U^n_{N-1} - (4r_2 - r_1)U^n_{N-2} + r_2U^n_{N-3} \\
&\quad + r_2(q(t_n) - q(t_{n+1})), \\
z_N &= (1 - 2r_1 + 5r_2)U^n_N + (r_1 - 4r_2)U^n_{N-1} + r_2U^n_{N-2} + (r_1 - 2r_2)q(t_n) \\
&\quad + (r_1 + 2r_2)q(t_{n+1}) + \frac{r_1\delta}{6}(q'(t_n) - q'(t_{n+1})).
\end{aligned}
$$

This pentadiagonal system can be solved with twice the computation cost of a tridiagonal system.

References

1. Butcher, J.C.: An algebraic theory of integration methods. Math. Comput. **26**, 79–106 (1972)
2. Butcher, J.C.: The Numerical Analysis of Ordinary Differential Equations: Runge–Kutta and General Linear Methods. Wiley, Chichester (1987)
3. Butcher, J.C.: Numerical Methods for Ordinary Differential Equations, 2nd edn. Wiley, Chichester (2008)
4. Chan, R.P.K., Tsai, A.Y.J.: On explicit two-derivative Runge–Kutta methods. Numer. Algoritm. **53**, 171–194 (2010)
5. Gerisch, A., Lang, J., Podhaisky, H., Weiner, R.: High-order linearly implicit two-step peer - finite element methods for time-dependent PDEs. Appl. Numer. Math. **59**, 624–638 (2009)
6. Hundsdorfer, W.H., Verwer, J.G.: Numerical solution of time-dependent advection-diffusion-reaction equations. Springer Series in Computational Mathematics, vol. 33. Springer, Berlin (2003)
7. Lele, S.K.: Compact finite difference schemes with spectral-like resolution. J. Comput. Phys. **103**(1), 16–42 (1992)
8. Rektorys, K.: The Method of Discretization in Time and Partial Differential Equations. D. Reidel Publishing Company, Dordrecht (1982)
9. Sanz-Serna, J.M., Verwer, J.G., Hundsdorfer, W.H.: Convergence and order reduction of Runge–Kutta schemes applied to evolutionary problems in partial differential equations. Numer. Math. **50**(4), 405–418 (1987)
10. Tsai, A.Y.J.: Two-Derivative Runge–Kutta Methods for Differential Equations. PhD thesis, University of Auckland, New Zealand (2011)