Journal of Applied Analysis and Computation
Volume 3, Number 3, August 2013
Website:http://jaac-online.com/
pp. 251–263

A SEMI-LAGRANGIAN RUNGE-KUTTA METHOD FOR TIME-DEPENDENT PARTIAL DIFFERENTIAL EQUATIONS

Daniel X. Guo

Abstract In this paper, a Semi-Lagrangian Runge-Kutta method is proposed to compute the numerical solution of time-dependent partial differential equations. The method is based on Lagrangian trajectory or the integration from the departure points to the arrival points (regular nodes). The departure points are traced back from the arrival points along the trajectory of the path. The high order interpolation is needed to compute the approximations of the solutions on the departure points, which most likely are not the regular nodes. On the trajectory of the path, the similar techniques of Runge-Kutta are applied to the equations to generate the high order Semi-Lagrangian Runge-Kutta method. The numerical examples show that this method works very efficient for the time-dependent partial differential equations.

Keywords Semi-Lagrangian, trajectory, Runge-Kutta, time-dependent partial differential equations.

MSC(2000) 76D05, 74H15, 68W25, 65Y20.

1. Introduction

Mathematical models (differential equations) in a vast range of fields, from science and technology to sociology and business almost every field where measurements can be taken, describe how quantities change as the time or location change. It depends on how many variables are involved, this leads naturally to the formulations of ordinary differential equations (ODEs), or partial differential equations (PDEs). Some of them, we can use the traditional techniques to provide an excellent explicit or implicit solutions, but they have limited practical value. Most realistic mathematical models cannot be solved in this way; instead, they must be dealt with by computational methods that provide approximate solutions (Burden and Faires [3]).

For this reason, we must rely on numerical methods that produce approximations to the desired solutions. Since the advent of personal computers in the mid of 20th century, a vast amount of effort has been expanded in designing, analyzing and applying computational techniques for differential equations (Burden and Faires [3] and Griffiths and Higham [8]). A great part of theoretical and practical developments have been made in this area, and new ideas continue to emerge.

For the well-posed initial-value problems of ODEs (Burden and Faires [3]),

$$\frac{dx(t)}{dt} = f(t, x(t)), \quad a \le t \le b, \qquad x(a) = \alpha,$$

Email address: guod@uncw.edu
Department of Mathematics and Statistics, University of North Carolina Wilmington, 601 South College Road, Wilmington, NC 28405, US

Runge-Kutta method obviously is a good choice to compute the approximations. Runge-Kutta methods are one-step methods composed of a number of stages (Griffiths and Higham [8]). They have high-order local truncation error while eliminating the need to compute and evaluate the derivatives of $f(t,x)$. A weighted average of the slopes (f) of the solution computed at nearby points is used to determine the solution at t_{n+1} from that at $t = t_n$.

However, the partial differential equations are much more complicated. In this paper, only time-dependent partial differential equations are investigated. The numerical solution of time-dependent partial differential equations is usually achieved by a proper combination of suitable spatial discretization and time integration processes. The method of lines (Houwen [11]) transforms initial-boundary value problems for time-dependent partial differential equations into initial-value problems for systems of ordinary differential equations. This is achieved by first discretization of the space variables by using finite difference, finite element or finite volume approximations. When a time-dependent partial differential equation involves terms of different types, it is a natural idea to employ different discretizations for them (Debrabant and Strehmel [5] and Driscol [6]). Implicit-explicit time-discretization schemes (Ascher etc. [1]) are an example of such a strategy. Fractional step Runge-Kutta methods (Portero etc. [12]) are a class of additive Runge-Kutta schemes that provide efficient time discretizations for evolutionary partial differential equations. Here the main idea is that the spatial discretization of PDEs leads to an initial-value problem of ODEs.Then, the traditional Runge-Kutta methods could be applied.

In this paper, the time-dependent differential equations as follows are considered,

$$u_t + uu_x = f(t,x,u,u_x,u_{xx}), \quad t > 0, \quad x \in \Omega$$

with initial-value and boundary conditions. Where Ω could be any dimensional. On the Lagrangian trajectory path, those equations could be considered as ODEs. In order to use Runge-Kutta methods, the arrival and departure points at t_{n+1} and $t = t_n$ are needed. One could be specified on regular nodes, but the other one needs to be computed. Also Runge-Kutta methods are applied to different types of partial differential equations (Bermejo and Carpio [2], Calvo etc. [4], Seaid [13], and Verwer [15]).

This paper is organized as follows: Section 2 shows the specified partial differential equations with given initial value and boundary conditions; The first order method is constructed and analyzed in Section 3; In Section 4, the high order Semi-Lagrangian Runge-Kutta methods are listed; The numerical examples are presented in Section 5; The final section is the consideration of future work.

2. Time-Dependent partial differential Equations

Many important partial differential equations (PDEs) have the format as

$$\frac{du(t,x)}{dt} = f(t,x,u,u_x,u_{xx}), \quad t > 0, \quad x \in \Omega,$$

with initial-value

$$u(0,x) = u_0(x), \quad x \in \Omega,$$

where Ω is a domain in one, two, or three dimensions. The boundary condition is provided for $u(t,x)$ on $\partial\Omega$ if it is needed.

The full derivative reads

$$\frac{d}{dt} = \frac{\partial}{\partial t} + u \cdot \frac{\partial}{\partial x},$$

where u is the velocity field and x is the spatial variable. In this paper, the dimensional one is considered for demonstration. It is more complicated for higher dimensions, but the idea is very similar.

For examples, the Burgers equation,

$$u_t + uu_x = \nu u_{xx},$$

with Dirichlet boundary conditions and initial conditions, and the Navier-Stokes equations,

$$\begin{cases} u_t + (u \cdot \nabla)u = f + \nu \Delta u - \nabla p, \\ \operatorname{div} u = 0, \end{cases}$$

with periodic boundary conditions and the initial conditions. Even the reformulated Shallow-Water equations (Guo and Drake [9, 10]) have the similar formate. This will be studied in the another paper with spectral method.

However, those equations have more partial differential terms in the equations. Those terms will cause some difficulties near boundaries. It depends on the boundary conditions. A suitable interpolation method on spatial direction is needed. In this paper, in order to focus on the proposed method, we will study the equations without those terms. Those terms will be studied in the future.

The following time-dependent differential equation will be studied in this paper

$$\frac{du}{dt} = f(t, x, u), \quad u(0, x) = u_0(x), \quad a \le x \le b \tag{2.1}$$

with periodic boundary conditions about x.

3. Semi-Lagrangian Euler's Method

In the reality, the exact solution of the problem (2.1) is hard to find. The approximations to u will be computed first at the given points, called grid points, in the domain $[0, T] \times [a, b]$. At other points in the domain, the approximate solutions are interpolated from the approximations on the grid points.

Now suppose that the grid points are equally distributed throughout the domain $[0, T] \times [a, b]$. Let N and M be two positive integer numbers. The grid points are formed by calculating

$$\begin{aligned} t_k &= k\tau, \text{ for each } k = 0, 1, 2, \ldots, N, \\ x_i &= a + ih, \text{ for each } i = 0, 1, 2, \ldots, M, \end{aligned}$$

with the step sizes $\tau = T/N$ and $h = (b - a)/M$.

Assume that all solutions u on (t_k, x_i^k) for $i = 0, 1, \ldots, M$ are known, and we need to find the solutions u on (t_{k+1}, x_i^{k+1}) for $i = 1, 2, \ldots, M-1, M$. Let x_A^{k+1} be one of x_i at $t = t_{k+1}$ for $i = 1, 2, \ldots, M-1$, noted as arrival point. From the view of Lagrangian trajectory, the particles on arrival points at $t = t_{k+1}$ shall come from some points, called departure points at $t = t_k$ along the trajectory of path.

In order to recover the departure point x_D^k at $t = t_k$ for the arrival point x_A^{k+1}, the following equation is considered,

$$\frac{dx}{dt} = u. \tag{3.1}$$

Note that the departure point x_D^k may not be any of x_i, $i = 0, 1, \ldots, M$. Then the departure point x_D^k is calculated by integrating the equation backwards along the trajectory from the arrival point x_A^{k+1} as

$$x_D^k = x_A^{k+1} - \int_{t_k}^{t_{k+1}} u\, dt.$$

It is obvious that Euler's method will provide the first order approximation as

$$x_D^k \approx x_A^{k+1} - \tau u(t_k, x_D^k).$$

This is an implicit equation about x_D^k and $u(t_k, x_D^k)$ is approximated by interpolation from $u(t_k, x_i)$ for $i = 0, 1, \ldots, M$. The results of numerical experiment showed that couple iterations are enough to obtain good approximation of x_D^k (Williamson and Olson [16] and Simmons [14]).

3.1. Algorithm

Along the trajectory of the path, we integrate the original equation from the time step t_k to the current step t_{k+1}, it gives

$$u(t_{k+1}, x_A^{k+1}) = u(t_k, x_D^k) + \int_{t_k}^{t_{k+1}} f(t, x, u)\, dt.$$

Let $w_A^{k+1} \approx u(t_{k+1}, x_A)$ and $w_D^k \approx u(t_k, x_D^k)$ for $k = 0, 1, \ldots, N-1$, then the Semi-Lagrangian Euler's method reads

$$\begin{cases} w_A^0 = u_0(x_A^0), \quad x_A^0 = x_A^k = x_i, i = 0, 1, \ldots, M, \\ x_D^k = x_A^{k+1} - \tau\, w_D^k, \quad w_D^k = \text{ Interpolation of } w_A^k, \\ w_A^{k+1} = w_D^k + \tau f(t_k, x_D^k, w_D^k), \quad \text{for each } k = 0, 1, \cdots, N-1. \end{cases} \tag{3.2}$$

Since the arrival points are the same as the regular grid points, for the time step $t = t_k$, we suppose that all of $w_A^k = w_i^k \approx u(t_k, x_i)$ for $i = 0, 1, \cdots, M$ are known. So, w_D^k can be approximated by an interpolation method from w_A^k. It depends on the overall accuracy of the proposed method, a suitable interpolation method is needed. However, for the Semi-Lagrangian Euler's method, a linear interpolation is good enough.

3.2. Local truncation error

For comparing the efficiency of various approximation methods, we introduce the local truncation error of the method. The local truncation error at a specified step measures the amount of error by which the exact solution to the differential equation fails to satisfy the difference equation used for the approximation.

Definition 3.1. The difference method

$$\begin{cases} w_A^0 = u_0(x_A^0), \quad x_A^0 = x_A^k = x_i, i = 0, 1, \ldots, M, \\ x_D^k = x_A^{k+1} - \tau\, w_D^k, \quad w_D^k = \text{ Interpolation of } w_A^k, \\ w_A^{k+1} = w_D^k + \tau\phi(t_k, x_D^k, w_D^k), \quad \text{for each } k = 0, 1, \cdots, N-1 \end{cases}$$

has local truncation error

$$\begin{aligned} \|\varepsilon_{k+1}(\tau)\|_{L_\infty} &= \left\| \frac{u_A^{k+1} - [u_D^k + \tau\phi(t_k, x_D^k, u_D^k)]}{\tau} \right\|_{L_\infty} \\ &= \left\| \frac{u_A^{k+1} - u_D^k}{\tau} - \tau\phi(t_k, x_D^k, u_D^k)) \right\|_{L_\infty}, \end{aligned}$$

where the norm is taken in the spatial variable x and $u_A^{k+1} = u(t_{k+1}, x_A^{k+1})$ and $u_D^k = u(t_k, x_D^k)$.

For the Semi-Lagrangian Euler's method, let $u(t, x)$ be the unique solution with up to order two continuous partial derivatives on $[0, T] \times [a, b]$, so that for each $k = 0, 1, \ldots, N-1$,

$$u(t_{k+1}, x_A^{k+1}) = u(t_k, x_D^k) + \tau \frac{\partial u(t_k, x_D^k)}{\partial t} + (x_A^{k+1} - x_D^k) \frac{\partial u(t_k, x_D^k)}{\partial x} + O(\tau^2 + \tau h + h^2).$$

Note that from the equation (3.1) and the approximation, we have

$$x_A^{k+1} - x_D^k = \tau\, u(t_k, x_D^k) + O(\tau^2).$$

Since $f(t, x, u) = \dfrac{\partial u}{\partial t} + u \cdot \dfrac{\partial u}{\partial x}$, then

$$u(t_{k+1}, x_A^{k+1}) = u(t_k, x_D^k) + \tau f(t_k, x_D^k, u(t_k, x_D^k)) + O(\tau^2 + \tau h + h^2). \tag{3.3}$$

The local truncation error at the kth step for the problem (2.1) is

$$\|\varepsilon_{k+1}(\tau)\|_{L_\infty} = O(\tau),$$

with the assumption $h/\tau = O(1)$.

3.3. Error bound

To derive an error bound for the Semi-Lagrangian Euler's method, we need two computational lemmas. Their proofs can be found on any numerical analysis book, for example [3].

Lemma 3.1. *For all $t \geq -1$ and any positive m, we have $0 \leq (1+t)^m \leq e^{mt}$.*

Lemma 3.2. *If s and t are positive real numbers, $\{a_k\}_{k=0}^n$ is a sequence satisfying $a_0 \geq -t/s$, and*

$$a_{k+1} \leq (1+s)a_k + t, \quad \textit{for each } k = 0, 1, 2, \cdots, n-1,$$

then

$$a_{k+1} \leq e^{(k+1)s}(a_0 + \frac{t}{s}) - \frac{t}{s}.$$

Theorem 3.1. *Suppose $x_i = a + ih$ for $i = 0, 1, \cdots, M$ and $h = (b-a)/M$ and $y \in C^2[a,b]$ with a constant M such that*

$$|y''| \le M, \text{ for all } x \in [a,b].$$

Then for each $x \in [a,b]$, if $p(x)$ is a piece-wise linear interpolation with $p(x_i) = w_i$, $i = 0, 1, \cdots, M$, then

$$|y(x) - p(x)| \le ||y(x_i) - w_i||_{L_\infty} + \frac{h^2}{2}M.$$

Proof. For any $x \in [a,b]$, if $x = x_i$ for any $i \in \{0, 1, \cdots, M\}$, then the result is clearly true, since $y(x_i) = w_i$. Suppose that $x \in (x_i, x_{i+1})$ for some i, then

$$p(x) = \frac{x - x_{i+1}}{x_i - x_{i+1}} w_i + \frac{x - x_i}{x_{i+1} - x_i} w_{i+1}$$

and there is $\xi \in [x_i, x_{i+1}]$ such that

$$y(x) = \frac{x - x_{i+1}}{x_i - x_{i+1}} y(x_i) + \frac{x - x_i}{x_{i+1} - x_i} y(x_{i+1}) + \frac{1}{2} y''(\xi)(x - x_i)(x - x_{i+1}).$$

Then we have

$$\begin{aligned} y(x) - p(x) = \quad & \frac{x - x_{i+1}}{x_i - x_{i+1}}[y(x_i) - w_i] + \frac{x - x_i}{x_{i+1} - x_i}[y(x_{i+1}) - w_{i+1}] \\ & + \frac{1}{2} y''(\xi)(x - x_i)(x - x_{i+1}) \end{aligned}$$

and

$$\begin{aligned} |y(x) - p(x)| \quad & \le \frac{x_{i+1} - x}{x_{i+1} - x_i} |y(x_i) - w_i| + \frac{x - x_i}{x_{i+1} - x_i} |y(x_{i+1}) - w_{i+1}| \\ & \quad + \frac{1}{2} |y''(\xi)| |(x - x_i)(x - x_{i+1})| \\ & \le \|y(x_i) - w_i\|_{L_\infty} + \frac{h^2}{2} M. \end{aligned}$$

□

Theorem 3.2. *Let $u(t,x)$ denote the unique solution to the problem (2.1) and w_A^0, w_A^1, $\cdots$, w_A^N be the approximations generated by the Semi-Lagrangian Euler's method for some positive integer N. τ and h are step sizes for t and x respectively. Suppose f is continuous and satisfies a Lipschitz condition in the third variable with constant L on*

$$D = \{(t,x,u) \,|\, 0 \le t \le T, a \le x \le b, -\infty < u < \infty\}$$

and that a constant M exists with

$$|\frac{d^2 u}{dt^2}| \le M, \text{ for all } (t,x) \in [0,T] \times [a,b].$$

Then, for each $k = 0, 1, 2, \cdots, N$,

$$\|u(t_k, x_A^k) - w_A^k\|_{L_\infty} \le \frac{(\tau^2 + h^2 + \tau h^2 L)M}{2\tau L} [e^{t_{k+1} L} - 1].$$

Proof. When $k = 0$ the result is clearly true, since $u(t_0, x_0) = w_A^0 = u_0(x_A^0)$. From the Taylor's theorem stated later, we have for $k = 0, 1, \cdots, N-1$

$$u(t_{k+1}, x_A^{k+1}) = u(t_k, x_D^k) + \tau f(t_k, x_D^k, u(t_k, x_D^k)) + \frac{\tau^2}{2}\frac{d^2u}{dt^2}(\xi_k, \eta),$$

and from the equation (3.2),

$$w_A^{k+1} = w_D^k + \tau f(t_k, x_D^k, w_D^k).$$

Let $u_A^{k+1} = u(t_{k+1}, x_A^{k+1})$ and $u_D^k = u(t_k, x_D^k)$, then we have

$$u_A^{k+1} - w_A^{k+1} = u_D^k - w_D^k + \tau[f(t_k, x_D^k, u(t_k, x_D^k)) - f(t_k, x_D^k, w_D^k)] + \frac{\tau^2}{2}\frac{d^2u}{dt^2}(\xi_k, \eta),$$

and

$$|u_A^{k+1} - w_A^{k+1}| \le |u_D^k - w_D^k| + \tau|f(t_k, x_D^k, u(t_k, x_D^k)) - f(t_k, x_D^k, w_D^k)| + \frac{\tau^2}{2}|\frac{d^2u}{dt^2}(\xi_k, \eta)|.$$

Since f satisfies a Lipschitz condition in the third variable with a constant L and $\|d^2u/dt^2\| \le M$, we have

$$|u_A^{k+1} - w_A^{k+1}| \le (1 + \tau L)|u_D^k - w_D^k| + \frac{\tau^2 M}{2}$$

then from Theorem 3.1,

$$\begin{aligned}\|u^{k+1} - w_A^{k+1}\|_{L_\infty} &\le (1 + \tau L)(\|u_A^k - w_A^k\|_{L_\infty} + \frac{h^2 M}{2}) + \frac{\tau^2 M}{2} \\ &\le (1 + \tau L)\|u_A^k - w_A^k\|_{L_\infty} + \frac{M}{2}(\tau^2 + h^2 + \tau h^2 L).\end{aligned}$$

Referring to Lemma 3.2 and letting $s = \tau L$, $t = (\tau^2 + h^2 + \tau h^2 L)M/2$, and $a_k = \|u_A^k - w_A^k\|_{L_\infty}$, for each $k = 0, 1, \cdots, N$, we see that

$$\begin{aligned}\|u_A^{k+1} - w_A^{k+1}\|_{L_\infty} \le \quad & e^{(k+1)\tau L}[\|u_A^0 - w_A^0\|_{L_\infty} + \frac{(\tau^2 + h^2 + \tau h^2 L)M}{2\tau L}] \\ & - \frac{(\tau^2 + h^2 + \tau h^2 L)M}{2\tau L}.\end{aligned}$$

Since $|u_A^0 - w_A^0| = 0$ and $(k+1)\tau = t_{k+1} - t_0 = t_{k+1}$, we have

$$\|u_A^{k+1} - w_A^{k+1}\|_{L_\infty} \le \frac{(\tau^2 + h^2 + h^2\tau L)M}{2\tau L})(e^{t_{k+1}L} - 1),$$

for each $k = 0, 1, \cdots, N-1$.

□

The importance of above error-bound formula is that the bound depends linearly on the step size τ with the assumption $h/\tau = O(1)$. However, the round-off error is not considered. As τ becomes smaller, more calculations are necessary and more round-off error is expected.

4. Semi-Lagrangian Runge-Kutta Methods

It is well known that Runge-Kutta methods have the high-order local truncation error while eliminating the need to compute and evaluate the derivatives of $f(t,x,u)$. Before presenting the ideas behind their derivation, we need to state Taylor's theorem in three variables. The proof of this result can be found in any standard book on advanced calculus.

A multi-index is an n-tuple of nonnegative integers. Let $\alpha=(\alpha_1,\alpha_2,\ldots,\alpha_n)$, for example, be a n-index and $x=(x_1,x_2,\cdots,x_n)$ a vector with x_i as a real number. We define

$$|\alpha|=\alpha_1+\alpha_2+\cdots+\alpha_n,\quad \alpha!=\alpha_1!\alpha_2!\cdots\alpha_n!,$$

and

$$x^\alpha=x_1^{\alpha_1}x_2^{\alpha_2}\cdots x_n^{\alpha_n}$$
$$\partial^\alpha f=\partial_1^{\alpha_1}\partial_2^{\alpha_2}\cdots\partial_n^{\alpha_n}=\frac{\partial^{|\alpha|}f}{\partial x_1^{\alpha_1}\partial x_2^{\alpha_2}\cdots\partial x_n^{\alpha_n}},$$

where f is a scale function. The number $|\alpha|=\alpha_1+\alpha_2+\cdots+\alpha_n$ is called the order or degree of α. For example, with $n=3$ and $\mathbf{x}=(x,y,z)$, we could have

$$\partial^{(3,0,0)}f=\frac{\partial^3 f}{\partial x^3},\quad \partial^{(2,1,0)}f=\frac{\partial^3 f}{\partial x^2\partial y}.$$

Theorem 4.1. *(Folland [7]) Suppose that $f(t,x,u)$ and all its partial derivatives of order less than or equal to 4 are continuous on $D=\{(t,x,u)|0\le t\le T, a\le x\le b, c\le u\le d\}$, and let $(t_0,x_0,u_0)\in D$. For every $(t,x,u)\in D$, there exist ξ between t and t_0, λ between x and x_0, and μ between u and u_0 with*

$$f(t,x,u)=P_3(t,x,u)+R_3(t,x,u),$$

where the function $P_3(t,x,u)$ is called the third order Taylor polynomial in three variables for the function f about (t_0,x_0,u_0), and $R_3(t,x,u)$ is the remainder term associated with $P_3(t,x,u)$. They are defined as follows

$$P_3(t,x,u)=\sum_{|\alpha|\le 3}\frac{\partial^\alpha f(t_0,x_0,u_0)}{\alpha!}(t-t_0)^{\alpha_1}(x-x_0)^{\alpha_2}(u-u_0)^{\alpha_3}$$

and the reminder is given in Lagrange's form by

$$R_3(t,x,u)=\sum_{|\alpha|=4}\frac{\partial^\alpha f(\xi,\lambda,\mu)}{\alpha!}(t-t_0)^{\alpha_1}(x-x_0)^{\alpha_2}(u-u_0)^{\alpha_3}.$$

It is similar to deriving a Runge-Kutta method for the ordinary differential equations, we can use the above Taylor theorem for multiple variables to have higher order methods.

Modified Semi-Lagrangian Euler's method:

$$\begin{cases} w_A^0=u_0(x_A^0),\quad x_A^0=x_A^k=x_i, i=0,1,\ldots,M,\\ x_D^k=x_A^{k+1}-\tau\, w_D^k,\quad w_D^k=\text{ Interpolation of } w_A^k,\\ k_1=f(t_k,x_D,w_D^k),\\ k_2=f(t_{k+1},x_A^{k+1},w_D^k+\tau f(t_k,x_D^k,w_D^k)),\\ w_A^{k+1}=w_D^k+\frac{1}{2}\tau(k_1+k_2), \end{cases} \tag{4.1}$$

for all of arrival points A and $k = 0, 1, 2, \cdots, N-1$. It has a local truncation error of $O(\tau^2)$, provided that the interpolation is at least the first order.

Third order Semi-Lagrangian Runge-Kutta method:

$$\begin{cases} w_A^0 = u(x_A), \\ x_D = x_A - \tau\, w_D^k, \quad w_D^k = \text{ Interpolation of } w_A^k, \\ \bar{w}_A^{k+1} = w_D^k + \tau f(t_k, x_D, w_D^k), \\ x_D = x_A - \frac{1}{2}\tau\,(w_D^k + \bar{w}_A^{k+1}), \quad w_D^k = \text{ Interpolation of } w_A^k, \\ k_1 = f(t_k, x_D, w_D^k), \\ k_2 = f(t_k + \frac{1}{2}\tau, x_D + \frac{1}{2}\tau w_D^k, w_D^k + \frac{1}{2}\tau k_1), \\ k_3 = f(t_{k+1}, x_A, w_D^k - \tau k_1 + 2\tau k_2), \\ w_A^{k+1} = w_D^k + \frac{1}{6}\tau(k_1 + 4k_2 + k_3), \end{cases} \tag{4.2}$$

for all of arrival points A and $k = 0, 1, 2, \cdots, N-1$. It has a local truncation error of $O(\tau^3)$, provided that the interpolation is at least the second order.

The Semi-Lagrangian Runge-Kutta method of order four:

$$\begin{cases} w_A^0 = u(x_A), \\ x_D = x_A - \tau\, w_D^k, \quad w_D^k = \text{ Interpolation of } w_A^k, \\ \bar{w}_A^{k+1} = w_D^k + \tau f(t_k, x_D, w_D^k), \\ x_D = x_A - \frac{1}{2}\tau\,(w_D^k + \bar{w}_A^{k+1}), \quad w_D^k = \text{ Interpolation of } w_A^k, \\ k_1 = f(t_k, x_D, w_D^k), \\ k_2 = f(t_k + \frac{1}{2}\tau, x_D + \frac{1}{2}\tau w_D^k, w_D^k + \frac{1}{2}\tau k_1), \\ k_3 = f(t_k + \frac{1}{2}\tau, x_D + \frac{1}{2}\tau w_D^k, w_D^k + \frac{1}{2}\tau k_2), \\ k_4 = f(t_{k+1}, x_A, w_D^k + \tau k_3), \\ w_A^{k+1} = w_D^k + \frac{1}{6}\tau(k_1 + 2k_2 + 2k_3 + k_4), \end{cases} \tag{4.3}$$

for all of arrival points A and $k = 0, 1, 2, \cdots, N-1$. It has a local truncation error of $O(\tau^4)$, provided that the interpolation is at least the second order.

5. Numerical Experiments and Results

The main computational effort in applying the Semi-Lagrangian Runge-Kutta methods is the evaluation of f. In the second-order method, the local truncation error is $O(\tau^2)$, and the cost is two functional evaluations per step. The Semi-Lagrangian Runge-Kutta method of order three requires four evaluations per step, and the local truncation error is $O(\tau^3)$.

Example 5.1. Consider the problem

$$u_t + u\ u_x = t + \sin(2\pi x) + 2\pi u\, t\cos(2\pi x), \quad 0 < t \le 1, 0 < x < 1,$$

$$u(0, x) = 0, \qquad 0 < x < 1$$

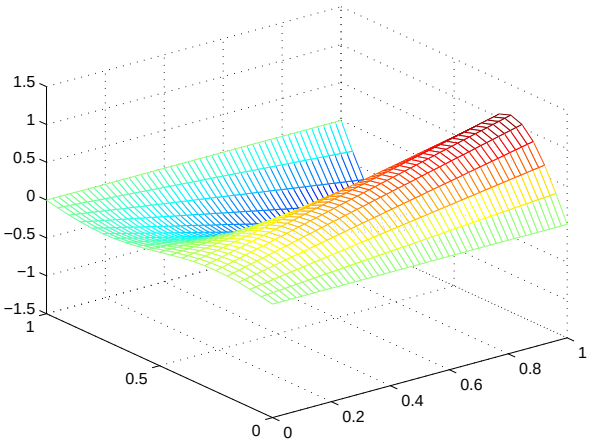

Figure 1. Solution of the initial-value problem.

with periodic boundary condition for x.

The exact solution for this problem is $u(t,x) = t^2/2 + t\sin(2\pi x)$. The figure 1 shows the graph of the exact solution. In order to focus on the truncation error from the proposed Semi-Lagrangian method, the periodic boundary condition is supplied. For other boundary conditions, we can easily use the interpolation to locate the departure points.

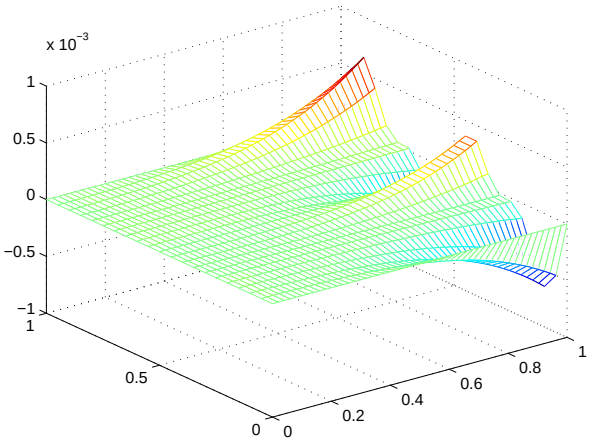

Figure 2. Difference between the solution of the initial-value problem and its approximation.

Semi-Lagrangian Euler's method, the modified Semi-Lagrangian Euler's method, and the Semi-Lagrangian Runge-Kutta method of order three are compared with $\tau = 0.02$ and $h = 0.04$. In the figure 2, it shows the difference between the exact

solution and the approximation from the Semi-Lagrangian Runge-Kutta method of order three at the common mesh points. As expected, the truncation errors are increasing as t goes from 0 to 1, but there is little changes when x goes from 0 to 1. The maximum error reached at the $t = 1$ for the certain points of x.

In the figure 3, the maximum truncation error presented at the each step of t for Semi-Lagrangian Euler's method, the modified Semi-Lagrangian Euler's method, and the Semi-Lagrangian Runge-Kutta method of order three. From the graph, the Semi-Lagrangian Runge-Kutta method of order three is much better than the rest of two methods. The next example will show the similar results but more precisely by date.

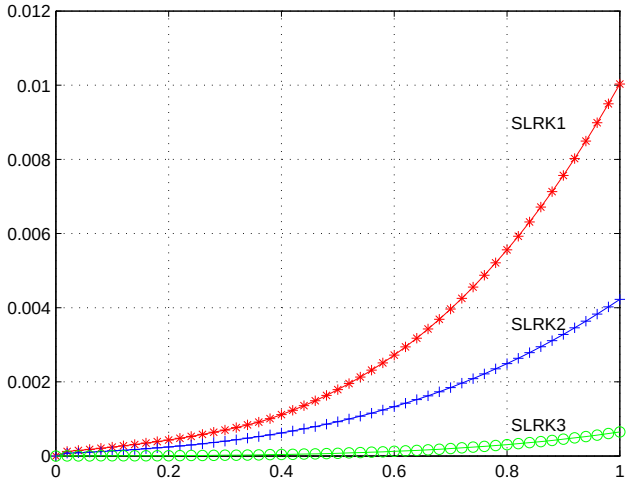

Figure 3. Truncation errors: $-*-$ first-order; $-+-$ second-order; $-o-$ third-order.

One measure of comparing the low-order Semi-Lagrangian Runge-Kutta methods is described as follows: The Semi-Lagrangian Runge-Kutta method of order four requires five evaluations per step, so it should give more accurate answers than Semi-Lagrangian Euler's method with one-fifth the step size if it is to be superior. Similarly, if the Semi-Lagrangian Runge-Kutta method of order four is to be superior to the second-order methods, it should give more accuracy with step size τ than a second-order method with step size $\frac{2}{5}\tau$, because the fourth-order method requires 2.5 times as many evaluations per step.

An illustration of the superiority of the Semi-Lagrangian Runge-Kutta method of order four by this measure is shown in the following example.

Example 5.2. Consider the problem

$$u_t + u\,u_x = -\frac{3}{10}e^{-0.3t} + \cos(2\pi x) - 2\pi u\,t\,\sin(2\pi x), \quad t > 0,\ 0 < x < 1,$$
$$u(0,x) = 1, \qquad 0 < x < 1$$

with period boundary condition for x.

Semi-Lagrangian Euler's method with $\tau = 0.01$, the modified Semi-Lagrangian Euler's method with $\tau = 0.02$, and the Semi-Lagrangian Runge-Kutta method of

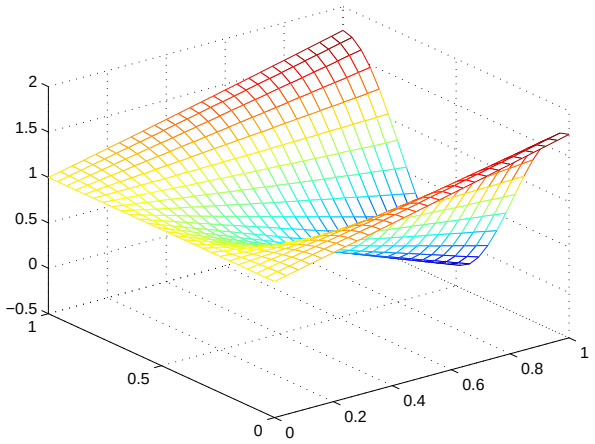

Figure 4. Solution of the initial-value problem.

order four with $\tau = 0.05$ are compared at the come mesh points of these methods $0.1, 0.2, 0.3, 0.4$, and 0.5. Each of these techniques requires 50 functional evaluations to determine the values listed in Table to approximate $u(0.5, x)$. In this example, the step size of x is the same as $h = 0.05$. In the table, the maximum errors between the the approximation and the exact solution for all nodes in x direction at the same time level were listed for the different methods. The Semi-Lagrangian Runge-Kutta method of order four is clearly superior.

t_k	Semi-L Euler $\tau = 0.01$	Modified Semi-L Euler $\tau = 0.02$	Semi-L Runge-Kutta order-four $\tau = 0.05$
0.0	0.0000000	0.0000000	0.0000000
0.1	0.0006533	0.0001689	0.0000642
0.2	0.0007900	0.0003966	0.0001198
0.3	0.0010513	0.0007493	0.0003829
0.4	0.0014093	0.0012410	0.0009169
0.5	0.0019023	0.0018842	0.0016314

6. Conclusions

The theorem in the section 3 and the examples in the section 5 are shown that the Semi-Lagrangian Runge-Kutta methods work for time-dependent partial differential equations. They are efficient. They are also very convenient for implementation.

In the examples, only one spatial variable was used. However, more spatial variables should work for them, but more complicated.

As stated in the introduction, only special time-dependent partial differential equations were used for examples. With more partial differential terms about the spatial variables, they will been treated more carefully. For different treatments, more results will be reported.

References

[1] Uri M. Ascher, Steven J. Ruuth and Raymond J. Spiteri, *Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations*, Applied Numerical Mathematics, 25 (1997), 151-167.

[2] R. Bermejo and J. Carpio, *An adaptive finite element semi-Lagrangian implicit-explicit Runge-Kutta-Chebyshev method for convection dominated reaction-diffusion problems*, Applied Numerical Mathematics, 58 (2009), 16-39.

[3] R. L. Burden and J. D. Faires, *Numerical Analysis*, Thomason Brooks/Cole, 2005.

[4] M. P. Calvo, J. de Frutos and J. Novo, *Linearly implicit Runge-Kutta methods for advection-reaction-diffusion equations*, Applied Numerical Mathematics, 535 (2001).

[5] K. Debrabant and K. Strehmel, *Convergence of Runge-Kutta methods applied to linear partial differential-algebraic equations*, Applied Numerical Mathematics, 53 (2005), 213-229.

[6] Tobin A. Driscol, *A Composite Runge-Kutta Method for the Spectral Solution of Semilinear PDEs*, Journal of Computational Physics, 182 (2002), 357-367.

[7] G. B. Folland, *Higher-Order Derivatives and Taylor's Formula in Several Variables*, www.math.washington.edu/∼folland/math425/taylor2.pdf, 2005.

[8] D. F. Griffiths and D. J. Higham, *Numerical Methods for Ordinary Differential Equations: Initial Value Problems*, Springer Undergrauate Mathematics Series, Springer-Verlag London Limited, 2010.

[9] Daniel X. Guo and John B. Drake, *A global semi-Lagrangian spectral model of the shallow water equations with variable resolution*, J. Comput. Phys., 206 (2005), 559-577.

[10] Daniel X. Guo and John B. Drake, *A global semi-Lagrangian spectral model for the reformulated shallow water equations*, Dynamical systems and differential equations (Wilmington, NC, 2002). Discrete Contin. Dyn. Syst. suppl., 2003, 375-385.

[11] P. J. van Houwen, *The development of Runge-Kutta methods for partial differential equations*, Applied Numerical Mathematics, 20 (1996), 261-272.

[12] L. Portero, A. Arraras and J. C. Jorge, *Variable step-size fractional step Runge-Kutta methods for time-dependent partial differential equations*, Applied Numerical Mathematics, 62 (2012), 1463-1476.

[13] M. Seaid, *An Eulerian-Lagrangian method for parabolic-hyperbolic equations*, Applied Numerical Mathematics, 59 (2009), 745-768.

[14] A. AJ. Simmons, *Development of a high resolution, semi-Lagrangian version of the ECMWF forecast model*, Proc. of the Seminar on Numerical Methods in Atmospheric Models, Reading, United Kingdom, ECMWF, 1991, 281-324.

[15] J. G. Verwer, *Explicit Runge-Kutta methods for parabolic partial differential equations*, Applied Numerical Mathematics, 22 (1996), 359-379.

[16] D. L. Williamson and J. G. Olson, *Climate Simulations with a Semi-Lagrangian Version of the NCAR Community Climate Model*, Monthly Weather Review, 122 (1994), 1594-1610.