

The Runge–Kutta method in geometric multiplicative calculus

Mustafa Riza and Hatice Aktöre

ABSTRACT

This paper illuminates the derivation, applicability, and efficiency of the multiplicative Runge–Kutta method, derived in the framework of geometric multiplicative calculus. The removal of the restrictions of geometric multiplicative calculus on positive-valued functions of real variables and the fact that the multiplicative derivative does not exist at the roots of the function are presented explicitly to ensure that the proposed method is universally applicable. The error and stability analyses are also carried out explicitly in the framework of geometric multiplicative calculus. The method presented is applied to various problems and the results are compared to those obtained from the ordinary Runge–Kutta method. Moreover, for one example, a comparison of the computation time against relative error is worked out to illustrate the general advantage of the proposed method.

1. Introduction

The invention of multiplicative calculus can be dated back to 1972, when Michael Grossman and Robert Katz finished their book on non-Newtonian calculus [14], in which they proposed nine different non-Newtonian calculi. Later Michael Grossman elaborated the bigeometric multiplicative calculus in [13]. Bigeometric multiplicative calculus was also proposed independently by Córdova-Lepe [9] under the name proportional calculus. Although Volterra and Hostinsky proposed a kind of multiplicative calculus in [28], we cannot date the invention of multiplicative calculus back to 1938. After 25 years of silence in this field Bashirov *et al.* presented a mathematically precise description of the geometric multiplicative calculus in [5]. This work initiated numerous studies in the field of multiplicative calculus. Several multiplicative numerical approximation methods have been proposed and discussed; see, for example, [19–22, 24, 25]. Moreover, multiplicative calculus has found its way into biomedical image analysis [12] and modelling with differential equations [6]. Furthermore, the Runge–Kutta method was developed in the framework of bigeometric calculus for applications in dynamic systems by Aniszewska *et al.* in [1]. A more exact bigeometric Runge–Kutta method was proposed by Riza and Eminaga [24], based on the bigeometric Taylor theorem derived in [24].

One drawback of multiplicative calculus generally put forward is that it can only be applied to positive-valued functions of real variables. This restriction can be circumvented by using complex multiplicative calculus. The first attempt was presented by Uzer in [27]. A mathematically precise description of the complex geometric multiplicative calculus was given by Bashirov and Riza in [4, 7]. The fact that the derivative is a local property suggests the extension to the complex domain. So, a simple change to complex-valued functions of real variables removes the restriction to purely positive functions and allows us to treat the real and imaginary part independently as the multiplicative Cauchy–Riemann conditions become trivial in this case.

In §2 the second-order multiplicative Runge–Kutta method, or correspondingly the multiplicative Heun method, the third-order multiplicative Runge–Kutta method and the

Received 27 September 2014; revised 24 March 2015.

2010 Mathematics Subject Classification 65L06 (primary), 34K28 (secondary).

fourth-order multiplicative Runge–Kutta method for positive-valued functions of real variables will be elaborated, and the extension to complex-valued functions of real variables will be presented. Another well-known drawback of multiplicative calculus is the breakdown of the multiplicative derivative at the roots of the functions. Section 2.4 covers the solution to that problem. The error analysis for the geometric multiplicative Runge–Kutta method presented in §3, carried out in analogy to the ordinary Runge–Kutta method as for example in [26], shows that the error becomes considerably smaller for the same step size compared to the ordinary Runge–Kutta method. Furthermore, the stability analysis is presented explicitly in analogy to [19]. In §4 the geometric multiplicative Runge–Kutta method will be applied to a multiplicative initial-value problem, not involving the exponential or logarithmic functions, with a known closed-form solution. The results of the application of the multiplicative Runge–Kutta method are compared to the results of the ordinary Runge–Kutta method for a fixed step width h . Furthermore, the comparison of the computation time against the relative error with varying step width for this example is presented to show the superiority of the proposed method. Based on the Baranyi model for the growth of bacteria [2, 3] using differential equations, the multiplicative Runge–Kutta method was applied to the growth of bacteria in food modelled by Huang [15–17], and compared to the results from the ordinary Runge–Kutta method. As an example of a coupled system of multiplicative initial-value problems, a second-order differential equation, with well-known closed form solution also used in [25], is used to compare the multiplicative Runge–Kutta method with the multiplicative finite-difference method. All examples show the superiority of the multiplicative Runge–Kutta method, with respect to error as well as performance. Finally, all findings are summarized in §5.

In order to ease the reading of this paper, we will use the terms ‘multiplicative calculus’ and ‘geometric multiplicative calculus’ interchangeably.

2. Multiplicative Runge–Kutta method for real-valued functions of real variables

In this section the multiplicative Runge–Kutta method is explicitly derived for the second-order case. Only the starting equations and the results of the fourth-order multiplicative Runge–Kutta method will be presented.

The methods derived in the following will be used to find suitable approximations to the solution of multiplicative initial-value problems of the form

$$y^*(x) = f(x, y), \quad (2.1)$$

with the initial condition

$$y(x_0) = y_0. \quad (2.2)$$

2.1. Second-order multiplicative Runge–Kutta method

The simplest approach to find an approximation to the solution of the differential equation (2.1) with the initial value (2.2) is the second-order Runge–Kutta method, also known as Heun’s method. In analogy to the ordinary Heun method, we will derive in the following the second-order multiplicative Runge–Kutta method or the multiplicative Heun method by making the ansatz

$$y(x+h) = y(x) \cdot f_0^{ah} \cdot f_1^{bh}, \quad (2.3)$$

where

$$f_0 = f(x, y), \quad (2.4)$$

$$f_1 = f(x+ph, y \cdot f_0^{qh}). \quad (2.5)$$

The multiplicative Taylor expansion of $y(x + h)$ up to order 2 is given by

$$y(x + h) = y(x) \cdot y^*(x)^h \cdot y^{**}(x)^{h^2/2} \dots \tag{2.6}$$

Remembering that

$$y^*(x) = f(x, y) \quad \text{and} \quad y^{**}(x) = f_x^*(x, y) \cdot f_y^*(x, y)^{y \ln f(x, y)}, \tag{2.7}$$

the multiplicative Taylor expansion of $y(x + h)$ becomes

$$y(x + h) = y(x) \cdot f(x, y)^h \cdot f_x^*(x, y)^{h^2/2} \cdot f_y^*(x, y)^{y \ln f(x, y) h^2/2}, \tag{2.8}$$

where $f_x^*(x, y)$ denotes the multiplicative partial derivative with respect to x and $f_y^*(x, y)$ with respect to y , respectively.

In order to compare (2.8) with (2.3) we need to expand f_1 using the multiplicative Taylor theorem up to order 1 as the power of the ansatz (2.3) also includes one h . Recalling that y is a function of x , the Taylor expansion for f_1 becomes, by the application of the multiplicative chain rule,

$$f_1 = f(x, y) \cdot f_x^*(x, y)^{ph} \cdot f_y^*(x, y)^{yqh \ln f_0}.$$

With $f_0 = f(x, y)$, the Taylor expansion of f_1 up to order 1 in h becomes

$$f_1 = f(x, y) \cdot f_x^*(x, y)^{ph} \cdot f_y^*(x, y)^{yqh \ln f(x, y)}. \tag{2.9}$$

Then, by substituting (2.9) and (2.4) in (2.3), we get the multiplicative Runge–Kutta expansion for the comparison with the multiplicative Taylor expansion of (2.8) as

$$y(x + h) = y(x) \cdot f(x, y)^{(a+b)h} \cdot f_x(x, y)^{bph^2} \cdot f_y(x, y)^{y \ln f(x, y) bqh^2}. \tag{2.10}$$

Comparison of the powers of $f(x, y)$ and its partial derivatives in (2.10) with (2.8) up to order 2 in h gives:

$$a + b = 1, \tag{2.11}$$

$$bp = \frac{1}{2}, \tag{2.12}$$

$$bq = \frac{1}{2}. \tag{2.13}$$

Obviously, we have infinitely many solutions of equations (2.11)–(2.13), as the number of unknowns is greater than the number of equations. Furthermore, we can see that $p = q$ and $a + b = 1$, which can be easily represented in analogy to the regular Butcher tableau [8] as the multiplicative Butcher tableau.

One possible choice of the parameters a, b, p , and q is as follows:

$$a = \frac{1}{2}, \quad b = \frac{1}{2}, \quad p = 1 \quad \text{and} \quad q = 1.$$

Here, we can see that we evaluate the function at the endpoints of the interval, and give equal weights to the contributions of f_0 and f_1 , resulting in the multiplicative Heun method formulae

$$\begin{aligned} y(x + h) &= y(x) \cdot f_0^{h/2} \cdot f_1^{h/2}, \\ f_0 &= f(x, y), \\ f_1 &= f(x + h, y \cdot f_0^h). \end{aligned}$$

Of course, depending on the problem, the parameters can be also chosen differently to optimize the solutions, satisfying the equations (2.11)–(2.13).

2.2. *Third-order multiplicative Runge–Kutta method*

For the derivation of the third-order multiplicative Runge–Kutta method we make the ansatz

$$y(x + h) = y(x) \cdot f_0^{ah} \cdot f_1^{bh} \cdot f_2^{ch}, \tag{2.14}$$

$$f_0 = f(x, y), \tag{2.15}$$

$$f_1 = f(x + ph, y \cdot f_0^{qh}), \tag{2.16}$$

$$f_2 = f(x + p_1h, y \cdot f_0^{q_1h} \cdot f_1^{q_2h}). \tag{2.17}$$

Analogously to the derivation of the multiplicative Heun method, presented above, the third-order multiplicative Runge–Kutta method is derived on the basis of equations (2.14)–(2.17), leading to

$$p = q, \tag{2.18}$$

$$p_1 = q_1 + q_2 \tag{2.19}$$

and

$$a + b + c = 1, \tag{2.20}$$

$$bp + cp_1 = \frac{1}{2}, \tag{2.21}$$

$$bp^2 + cp_1^2 = \frac{1}{3}. \tag{2.22}$$

As p and p_1 are determined by the choices of q , q_1 and q_2 , we have to solve the set of equations (2.20)–(2.22) with respect to a , b , c as functions of p and p_1 . We obtain

$$a = -\frac{-6pp_1 + 3p + 3p_1 - 2}{6pp_1}, \tag{2.23}$$

$$b = -\frac{3p_1 - 2}{6p(p - p_1)}, \tag{2.24}$$

$$c = -\frac{2 - 3p}{6p_1(p - p_1)}. \tag{2.25}$$

This results in the multiplicative Butcher tableau

0			
p	q		
p_1	q_1	q_2	
	a	b	c

As before, the number of solutions is infinite, as the number of unknowns is larger than the number of independent equations. One approach of choosing the constants is to evaluate the functions at the beginning of the interval (for f_0), in the middle of the interval (for f_1), and at the end of the interval (for f_2). Furthermore, we give equal weight to the function evaluated at the left and right endpoint of the interval of length h , and double the weight for the value in the middle of the interval. This results in the constants $a = \frac{1}{6}$, $b = \frac{2}{3}$, $c = \frac{1}{6}$, $p = \frac{1}{2}$, $p_1 = 1$, $q = \frac{1}{2}$, $q_1 = -1$ and $q_2 = 2$, and we get for the third-order multiplicative Runge–Kutta method:

$$y(x + h) = y(x) \cdot f_0^{h/6} \cdot f_1^{2h/3} \cdot f_2^{h/6},$$

$$f_0 = f(x, y),$$

$$f_1 = f\left(x + \frac{h}{2}, y \cdot f_0^{h/2}\right),$$

$$f_2 = f(x + h, y \cdot f_0^{-h} \cdot f_1^{2h}).$$

2.3. *Fourth-order multiplicative Runge–Kutta method*

In practice, the fourth-order Runge–Kutta method is mainly used. In analogy to the second-order multiplicative Runge–Kutta method described above, we will now employ the fourth-order multiplicative Runge–Kutta method. Consequently, we make the ansatz

$$y(x + h) = y(x) \cdot f_0^{ah} \cdot f_1^{bh} \cdot f_2^{ch} \cdot f_3^{dh}, \tag{2.26}$$

$$f_0 = f(x, y), \tag{2.27}$$

$$f_1 = f(x + ph, y \cdot f_0^{qh}), \tag{2.28}$$

$$f_2 = f(x + p_1h, y \cdot f_0^{q_1h} \cdot f_1^{q_2h}), \tag{2.29}$$

$$f_3 = f(x + p_2h, y \cdot f_0^{q_3h} \cdot f_1^{q_4h} \cdot f_2^{q_5h}). \tag{2.30}$$

Again we need to find the Taylor expansions of f_0, f_1, f_2 and f_3 in order to substitute them into the fourth-order multiplicative Runge–Kutta formula and compare it with the Taylor expansion of $y(x + h)$ up to order 4. After a lengthy calculation, we get by comparison the following set of equations:

$$p = q, \tag{2.31}$$

$$p_1 = q_1 + q_2, \tag{2.32}$$

$$p_2 = q_3 + q_4 + q_5, \tag{2.33}$$

and

$$a + b + c + d = 1, \tag{2.34}$$

$$bp + cp_1 + dp_2 = \frac{1}{2}, \tag{2.35}$$

$$bp^2 + cp_1^2 + dp_2^2 = \frac{1}{3}. \tag{2.36}$$

This results in the multiplicative Butcher tableau

0				
p	q			
p_1	q_1	q_2		
p_2	q_3	q_4	q_5	
	a	b	c	d

We can easily see that if the function $f(x, y)$ is independent of y , the result is independent of the selection of q_1, \dots, q_5 , and therefore any selection will give the same result.

2.4. *Extension to complex-valued functions of real variable*

One of the drawbacks of multiplicative calculus generally put forward is its restriction to positive-valued functions of real variables. In order to overcome this restriction the theory of multiplicative calculus was extended to the complex domain. It is well known from complex analysis that the differentiation rules are a slightly more complicated for complex-valued functions of complex variables as the Cauchy–Riemann conditions have to be satisfied. But here we are only interested in complex-valued functions of real variables, which simplifies the issue drastically, as the multiplicative counterparts of the Cauchy–Riemann conditions need not be taken into account and the differentiation can be carried out independently for the real and the imaginary part. As illustrated in [7], the multiplicative derivative can be calculated everywhere except at the point $0 + 0i$ in the complex plain. So the fourth-order multiplicative Runge–Kutta method can be extended to negative-valued functions as the phase factor is responsible

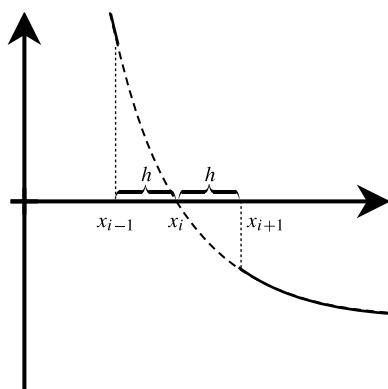


FIGURE 1. Bypass the roots where the multiplicative derivative becomes undefined. The dashed line denotes the region where the ordinary Runge–Kutta method is applied to prevent the multiplicative derivative becoming infinite. The multiplicative Runge–Kutta method is applied in the region of the solid line.

for the change of sign. The only problem that could not be solved by extending multiplicative calculus to the complex domain is that the multiplicative derivative is not defined at the roots of the function. So, a switch to Newtonian calculus becomes inevitable at these points. In every step of the multiplicative Runge–Kutta method, we get the value of the function at this point and its multiplicative derivative at this point and use the ordinary Runge–Kutta method for a couple of steps until the multiplicative derivative again becomes reasonably large and these values are then used as input to the multiplicative Runge–Kutta method. The results are reasonably good, and often even better than using the ordinary Runge–Kutta method alone.

If we assume that $f(x_{i-1}) > 0$ and $f(x_{i+1}) < 0$ and that the function is decreasing, then accordingly there must be a point $\xi \in [x_{i-1}, x_{i+1}]$ where $f(\xi) = 0$ (see Figure 1). In this case the multiplicative derivative of $f(x)$ is not defined at ξ . Therefore, the multiplicative Runge–Kutta method will be applied on the intervals $[x_0, x_{i-1}]$, and $[x_{i+1}, x_n]$. On $[x_{i-1}, x_{i+1}]$ we apply the ordinary Runge–Kutta method, using the values $f(x_{i-1})$ and $f^*(x_{i-1})$ calculated by the multiplicative Runge–Kutta method as input for the ordinary Runge–Kutta method, and vice versa for the point x_{i+1} .

The handover has been tested on several examples, working properly.

3. Error and stability analysis

3.1. Convergence of one-step methods

In this section we examine the convergence behaviour of our one-step method as $h \rightarrow 0$ of an approximate solution $\eta(x; h)$. We assume that f is once *-differentiable on the interval (a, b) , and $y(x)$ denotes the exact solution of the initial-value problem

$$y^* = f(x, y), \quad y(x_0) = y_0.$$

Let $\Phi(x, y; h)$ define a one-step method,

$$\eta_0 := y_0,$$

for $i = 0, 1, \dots$, written

$$\begin{aligned} \eta_{i+1} &:= \eta_i \Phi(x_i, \eta_i; h)^h, \\ x_{i+1} &:= x_i + h, \end{aligned}$$

which for $x \in R_h := \{x_0 + ih \mid i = 0, 1, 2, \dots\}$ produces the approximate solution $\eta(x; h)$:

$$\eta(x; h) := \eta_i \quad \text{if } x = x_0 + ih.$$

Let x and y be arbitrary, but fixed, and let $z(t)$ be the exact solution of the initial-value problem

$$z^*(t) = f(t, z(t)), \quad z(x) = y, \tag{3.1}$$

with initial values x, y . Then the function

$$\Delta(x, y; h) := \begin{cases} \left(\frac{z(x+h)}{y}\right)^h & \text{if } h \neq 0, \\ f(x, y) & \text{if } h = 0, \end{cases}$$

represents the multiplicative ratio function of the exact solution $z(t)$ of (3.1) for step size h , while $\Phi(x, y; h)$ is the multiplicative ratio function for step size h of the approximate solution of (3.1) produced by Φ . The multiplicative ratio function is the multiplicative counterpart to the difference quotient in Newtonian calculus.

The magnitude of the ratio

$$\tau(x, y; h) := \frac{\Delta(x, y; h)}{\Phi(x, y; h)}$$

indicates how well the value $z(x+h)$ at $x+h$ obeys the equation of the one-step method.

One calls $\tau(x, y; h)$ the *multiplicative local discretization error* at the point (x, y) . For a reasonable one-step method one will require that

$$\lim_{h \rightarrow 0} \tau(x, y; h) = 1.$$

We are interested in the behaviour of the *multiplicative global discretization error*

$$e(x; h) := \frac{\eta(x; h)}{y(x)}$$

for fixed x and $h \rightarrow 0$, $h \in H_x := \{(x - x_0)/n \mid n = 1, 2, \dots\}$. Since $e(x; h)$, like $\eta(x; h)$, is only defined for $h \in H_x$, we have to study the convergence of

$$e(x; h_n), \quad h_n := \frac{x - x_0}{n} \quad \text{as } n \rightarrow \infty.$$

We say that the one-step method is *convergent* if

$$\lim_{n \rightarrow \infty} e(x; h_n) = 1$$

for all $x \in [a, b]$ and all functions f that are once $*$ -differentiable on the interval (a, b) .

If f is p -times $*$ -differentiable on (a, b) , methods of order $p > 0$ are convergent and satisfy

$$e(x; h_n) = O(e^{h_n^p}).$$

The order of the multiplicative global discretization error is thus equal to the order of the multiplicative local discretization error.

LEMMA 1. *If the numbers ξ_i satisfy estimates of the form*

$$|\xi_{i+1}| \leq |\xi_i|^{(1+\delta)} B, \quad \delta > 0, B \geq 0, i = 0, 1, 2, \dots,$$

then

$$|\xi_n| \leq |\xi_0|^{e^{n\delta}} B^{(e^{n\delta}-1)/\delta}.$$

Proof. From the assumptions we immediately obtain

$$\begin{aligned} |\xi_1| &\leq |\xi_0|^{(1+\delta)} B, \\ |\xi_2| &\leq |\xi_0|^{(1+\delta)^2} B^{1+(1+\delta)}, \\ &\vdots \\ |\xi_n| &\leq |\xi_0|^{(1+\delta)^n} B^{[1+(1+\delta)+(1+\delta)^2+\dots+(1+\delta)^{n-1}]} \\ &= |\xi_0|^{(1+\delta)^n} B^{((1+\delta)^n-1)/\delta} \\ &\leq |\xi_0|^{e^{n\delta}} B^{(e^{n\delta}-1)/\delta}, \end{aligned}$$

since $0 < 1 + \delta \leq e^\delta$ for $\delta > -1$. □

THEOREM 1. *Consider, for $x_0 \in [a, b], y_0 \in \mathbb{R}$, the initial-value problem*

$$y^* = f(x, y), \quad y(x_0) = y_0,$$

having the exact solution $y(x)$. Let the function Φ be continuous on

$$G := \left\{ (x, y, h) \mid a \leq x \leq b, \left| \frac{y}{y(x)} \right| \leq \gamma, 0 \leq |h| \leq h_0 \right\}, \quad h_0 > 0, \gamma > 1,$$

and let there exist positive constants M and N such that

$$\left| \frac{\Phi(x, y_1; h)}{\Phi(x, y_2; h)} \right| \leq \left| \frac{y_1}{y_2} \right|^M$$

for all $(x, y_i, h) \in G, i = 1, 2$, and

$$|\tau(x, y(x); h)| = \left| \frac{\Delta(x, y(x); h)}{\Phi(x, y(x); h)} \right| \leq e^{N|h|^p}, \quad p > 0,$$

for all $x \in [a, b], |h| \leq h_0$. Then there exists an $\bar{h}, 0 < \bar{h} \leq h_0$, such that for the multiplicative global discretization error $e(x; h) = \eta(x; h)/y(x)$,

$$|e(x; h_n)| \leq e^{|h_n|^p N (e^{M|x-x_0|-1})/M}$$

for all $x \in [a, b]$ and all $h_n = (x - x_0)/n, n = 1, 2, \dots$, with $|h_n| \leq \bar{h}$. If $\gamma = \infty$, then $\bar{h} = h_0$.

Proof. The function

$$\tilde{\Phi}(x, y; h) = \begin{cases} \Phi(x, y; h) & \text{if } (x, y; h) \in G, \\ \Phi(x, y(x)\gamma; h) & \text{if } x \in [a, b], |h| \leq h_0, y \geq y(x)\gamma, \\ \Phi(x, \frac{y(x)}{\gamma}; h) & \text{if } x \in [a, b], |h| \leq h_0, y \leq \frac{y(x)}{\gamma}, \end{cases}$$

is evidently continuous on $\tilde{G} := \{(x, y, h) \mid x \in [a, b], y \in \mathbb{R}, |h| \geq h_0\}$ and satisfies the condition

$$\left| \frac{\tilde{\Phi}(x, y_1; h)}{\tilde{\Phi}(x, y_2; h)} \right| \leq \left| \frac{y_1}{y_2} \right|^M \tag{3.2}$$

for all $(x, y_i, h) \in \tilde{G}$, $i = 1, 2$, and because of $\tilde{\Phi}(x, y(x); h) = \Phi(x, y(x); h)$, the condition

$$\left| \frac{\Delta(x, y(x); h)}{\tilde{\Phi}(x, y(x); h)} \right| \leq e^{N|h|^p} \quad \text{for } x \in [a, b], |h| \leq h_0. \tag{3.3}$$

is also satisfied. Let the one-step method generated by $\tilde{\Phi}$ furnish the approximate values $\tilde{\eta}_i := \tilde{\eta}(x_i; h)$ for $y_i := y(x_i)$, $x_i := x_0 + ih$:

$$\tilde{\eta}_{i+1} = \tilde{\eta}_i \cdot \tilde{\Phi}(x_i, \tilde{\eta}_i; h)^h.$$

In view of

$$y_{i+1} = y_i \cdot \Delta(x_i, y_i; h)^h,$$

one obtains, for the error $\tilde{e}_i := \tilde{\eta}_i/y_i$, the recurrence formula

$$\tilde{e}_{i+1} = \tilde{e}_i \cdot \left[\frac{\tilde{\Phi}(x_i, \tilde{\eta}_i; h)}{\tilde{\Phi}(x_i, y_i; h)} \right]^h \cdot \left[\frac{\tilde{\Phi}(x_i, y_i; h)}{\Delta(x_i, y_i; h)} \right]^h. \tag{3.4}$$

Now from (3.2) and (3.3) it follows that

$$\left| \frac{\tilde{\Phi}(x_i, \tilde{\eta}_i; h)}{\tilde{\Phi}(x_i, y_i; h)} \right| \leq \left| \frac{\tilde{\eta}_i}{y_i} \right|^M = |\tilde{e}_i|^M,$$

$$\left| \frac{\tilde{\Phi}(x_i, y_i; h)}{\Delta(x_i, y_i; h)} \right| \leq e^{N|h|^p},$$

and hence from (3.4) we get the recursive estimate

$$|\tilde{e}_{i+1}| \leq |\tilde{e}_i|^{(1+|h|M)} e^{N|h|^{p+1}}.$$

As we are dealing with an initial-value problem, the initial values have to be considered exact, and therefore $\tilde{e}_0 = \tilde{\eta}_0/y_0 = 1$, resulting in

$$|\tilde{e}_k| \leq e^{N|h|^p(e^{k|h|M} - 1)/M}. \tag{3.5}$$

Now let $x \in [a, b]$, $x \neq x_0$, be fixed and $h := h_n = (x - x_0)/n$, $n > 0$ an integer. Then $x_n = x_0 + nh = x$ and from (3.5) with $k = n$, since $\tilde{e}(x; h_n) = \tilde{e}_n$, it follows at once that

$$|\tilde{e}(x; h_n)| \leq e^{N|h_n|^p(e^{M|x-x_0|-1})/M}$$

for all $x \in [a, b]$ and h_n with $|h_n| \leq h_0$. Since $|x - x_0| \leq |b - a|$ and $\gamma > 0$, there exists an \bar{h} , $0 < \bar{h} \leq h_0$, such that $|\tilde{e}(x; h_n)| \leq \gamma$ for all $x \in [a, b]$, $|h_n| \leq \bar{h}$, that is, for the one-step

method generated by Φ ,

$$\begin{aligned} \eta_0 &= y_0, \\ \eta_{i+1} &= \eta_i \Phi(x_i, \eta_i; h), \end{aligned}$$

we have for $|h| \leq \bar{h}$, according to the definition of $\tilde{\Phi}$,

$$\tilde{\eta}_i = \eta_i, \quad \tilde{e}_i = e_i \quad \text{and} \quad \tilde{\Phi}(x_i, \tilde{\eta}_i; h) = \Phi(x_i, \eta_i; h).$$

The assertion of the theorem,

$$|\tilde{e}(x; h_n)| \leq e^{N|h_n|^p(e^{M|x-x_0|-1})/M},$$

thus follows for all $x \in [a, b]$ and all $h_n = (x - x_0)/n$, $n = 1, 2, \dots$, with $|h_n| \leq \bar{h}$. □

3.2. Stability analysis

In this section we study the stability analysis of the multiplicative Runge–Kutta methods. Because of the presence of the Butcher tableau, the stability analysis could be carried out for the n th-order multiplicative Runge–Kutta method, but we do so exemplarily for the fourth-order multiplicative Runge–Kutta method to be able to show its behaviour explicitly. In Newtonian calculus, the stability properties of the Runge–Kutta methods are analysed by the basic test equation

$$y'(x) = \lambda y(x), \quad y(x_0) = y_0, \tag{3.6}$$

where $\lambda \in \mathbb{C}$. The behaviour of (3.6) was studied extensively by [10, 11, 18, 23]. The stability analysis of the multiplicative Runge–Kutta methods can also be done based on this test equation. For this purpose, we rewrite the test equation in terms of the multiplicative calculus. We will consider the fourth-order multiplicative Runge–Kutta method as denoted in (2.26)–(2.30).

By (2.26) we obtain

$$y_{n+1} = y_n [f_0^a \cdot f_1^b \cdot f_2^c \cdot f_3^d]^h, \tag{3.7}$$

where

$$a + b + c + d = 1. \tag{3.8}$$

In analogy to [19] the multiplicative form of the basic test equation is given by

$$y^*(x) = e^\lambda, \quad y(x_0) = y_0, \tag{3.9}$$

which has the analytic solution

$$y(x) = e^{\lambda(x-x_0)} y_0. \tag{3.10}$$

As $x \rightarrow \infty$ and $\text{Re}(\lambda) < 0$, the solution of the system approaches zero. If the method also has the same behaviour, then we can say that the method is A-stable [10].

Since $y^*(x)$ is a constant function, equations (2.27)–(2.30) simplify to $f_0 = f_1 = f_2 = f_3 = e^\lambda$. Then, by (3.9) and (3.7), we obtain

$$\frac{y_{n+1}}{y_n} = e^z = R(z), \tag{3.11}$$

where $z = \lambda h$. $R(z)$ is the stability function of the proposed method. Then the domain of stability is

$$S^* = \{z \in \mathbb{C} : |R(z)| < 1\}. \tag{3.12}$$

Consequently, by (3.12) we obtain

$$0 < e^{-|\lambda|h} < 1, \tag{3.13}$$

which leads to

$$0 < h < \infty. \tag{3.14}$$

Thus, the result shows that the proposed method is unconditionally stable. By (3.12), it can be seen that $\text{Re}(z) < 0$, where $|e^z| = e^{\text{Re}(z)}$. When $\text{Re}(z) < 0$ the left half plane will be the region of absolute stability, thus the method is A-stable. In Newtonian calculus, the explicit multistep methods cannot be A-stable and the implicit multistep methods can be A-stable if the order is at most 2. In contrast, in multiplicative calculus both explicit and implicit methods are A-stable. One can say that a method is L-stable if it is A-stable and $R(z) \rightarrow 0$ when $|z| \rightarrow \infty$ [11]. Since we have shown that the multiplicative Runge–Kutta methods are A-stable and $e^z \rightarrow 0$ when $|z| \rightarrow \infty$, we can say that the proposed methods are L-stable by [11].

4. Examples of the multiplicative Runge–Kutta method

4.1. Solution of first-order multiplicative differential equations

4.1.1. *Square-root example.* As our first example we want to discuss the following multiplicative initial-value problem, where no exponential function or logarithm is involved:

$$y^*(x) = e^{1/2y^2}, \quad y(0) = 1. \tag{4.1}$$

Here the corresponding Newtonian initial-value problem becomes

$$y'(x) = \frac{1}{2y}, \quad y(0) = 1. \tag{4.2}$$

The general solution of these two initial-value problems (4.1) and (4.2) is

$$y(x) = \sqrt{x + 1}. \tag{4.3}$$

Application of the fourth-order multiplicative Runge–Kutta method and the fourth-order ordinary Runge–Kutta method gives the results summarized in Table 1.

Table 1 shows that the relative error is 4 orders of magnitude greater for the fourth-order ordinary Runge–Kutta method than for the fourth-order multiplicative Runge–Kutta method. This is in good agreement with the error analysis presented in §3. On the other hand, we know that the basic operations used in multiplicative calculus are multiplication, division,

TABLE 1. Comparison of the multiplicative Runge–Kutta method and ordinary Runge–Kutta method.

x	y_{exact}	y_{MRK}	Relative errMRK in %	y_{Newt}	Relative errNewt in %
0	1	1	0	1	0
0.6	1.2649111	1.2649153	3.38×10^{-6}	1.2382302	0.021093074
1.2	1.4832397	1.483244	2.88×10^{-6}	1.4409643	0.028502049
1.8	1.6733201	1.673324	2.36×10^{-6}	1.6205072	0.031561693
2.4	1.8439089	1.8439125	1.97×10^{-6}	1.783364	0.032835088
3	2	2.0000034	1.69×10^{-6}	1.9334697	0.033265139

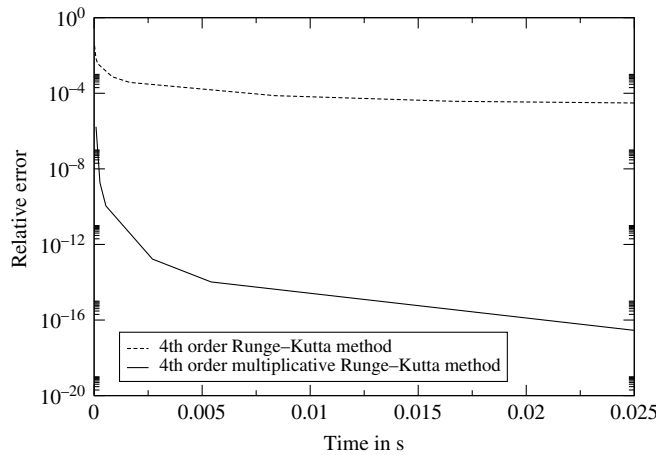


FIGURE 2. Comparison of the computation time and the relative error for the multiplicative initial-value problem (4.1) and the initial-value problem (4.2) for the same initial values $x_0 = 0$ and $y_0 = 1$ and fixed final values $x_n = 3$, $y_n = 2$ by varying h .

calculation of the exponential function and calculation of the logarithm function, whereas in the Newtonian case we only have to consider multiplication, summation and subtraction.

Let all numbers be of size n bits. The computational complexity for addition and subtraction is $O(n)$, for multiplication and division is $O(n^2)$ and for taking exponentials and logarithms is $O(n^{5/2})$. Evidently, then, the number of operations must be significantly smaller for the multiplicative Runge–Kutta method than for the ordinary Runge–Kutta method. In order to consider the multiplicative method as a serious alternative to the ordinary method, the performance of the former has to be at least comparable. Performance means higher accuracy, that is, smaller errors, for the same computation time. Therefore, the relative error as a function of the computation time has been measured by keeping the starting and end point fixed and varying the step size h . The results for both methods are compared in Figure 2.

The comparison of the relative errors as function of the computation time shows that the multiplicative Runge–Kutta method works more efficiently than the ordinary method, with a significant difference between the results. This comparison has also been carried out for other sample problems with known closed-form solutions. The results show again that the fourth-order multiplicative Runge–Kutta method is more efficient than the fourth-order ordinary Runge–Kutta method.

4.1.2. *Logarithmic example.* We next consider a function which has logarithmic solution and compare the results of the fourth-order multiplicative and ordinary Runge–Kutta methods. The multiplicative initial-value problem is

$$y^*(x) = e^{(x-1)/xy}, \quad y(1) = 1. \tag{4.4}$$

The corresponding ordinary initial-value problem is given by

$$y'(x) = 1 - \frac{1}{x}, \quad y(1) = 1. \tag{4.5}$$

The analytic solution of this initial-value problem can be easily obtained as

$$y(x) = x - \ln x. \tag{4.6}$$

The comparison of the results of the multiplicative and ordinary methods is given in Table 2. In this example the advantage of the multiplicative method over the ordinary method can be seen quite easily.

TABLE 2. Comparison of the multiplicative Runge–Kutta method and the ordinary Runge–Kutta method.

x	y_{exact}	y_{MRK}	Relative err _{MRK} in %	y_{Newt}	Relative err _{Newt} in %
1	1	1	0	1	0
1.5	1.0945	1.0945	3.128×10^{-3}	1.2123	10.76
2	1.3069	1.3068	2.955×10^{-3}	1.4892	13.95
2.5	1.5837	1.5837	2.681×10^{-3}	1.8068	14.09
3	1.9014	1.9013	2.339×10^{-3}	2.1527	13.22

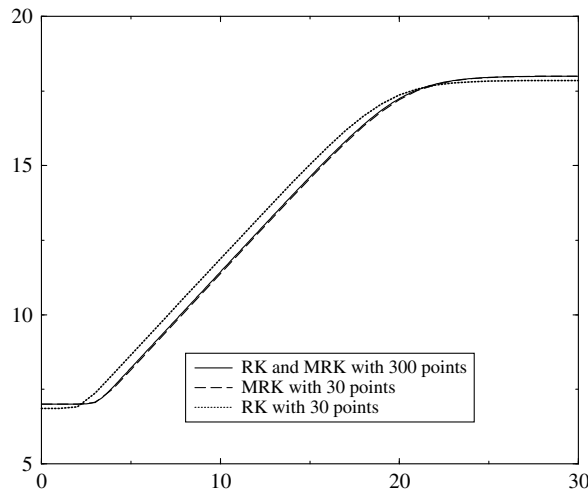


FIGURE 3. Solution for bacteria growth model, $\lambda = 3.21$, $\mu_{\text{max}} = 0.644$, $\alpha = 4$, $y_{\text{max}} = 18$.

4.1.3. *Biological example.* In order to show that the proposed method can also be used to obtain better results for mathematical models in biology, we discuss the growth of bacteria in food modelled by Huang [15–17].

The Baranyi model [2, 3] for the growth of bacteria in food is described by the differential equation

$$y'(t) = \mu_{\text{max}} \frac{1 - e^{y - y_{\text{max}}}}{1 + e^{-\alpha(t - \lambda)}}. \tag{4.7}$$

The multiplicative counterpart of equation (4.7) is

$$y^*(t) = \exp \left\{ \frac{\mu_{\text{max}}}{y} \frac{1 - e^{y - y_{\text{max}}}}{1 + e^{-\alpha(t - \lambda)}} \right\}, \tag{4.8}$$

with the initial value $y_0 = y(0) = 7$. As there is no closed-form solution available for these initial-value problems, we solved both initial-value problems using the fourth-order multiplicative and ordinary Runge–Kutta methods for small h , where both solutions coincide. Then we increased the step size h and checked which method deviates first from the solutions for small h . As depicted in Figure 3, the ordinary method deviates first from the accurate solution. We then compared the greatest h , where the multiplicative still coincides with the solutions for small h , and the ordinary method for this h . Also in this case, the multiplicative method gives better performance results than the ordinary methods.

The numerical solutions of the differential equations (4.7) and (4.8) using the corresponding Runge–Kutta methods are not distinguishable for $h = 0.1$. But, as depicted in Figure 3, the

fourth-order multiplicative Runge–Kutta method for $h = 1$ still coincides with the solution for $h = 0.1$, whereas the fourth-order Runge–Kutta method is significantly different (dotted line).

4.2. *Solution of a second-order multiplicative differential equation*

As an example of higher-order multiplicative initial-value problems we consider the following second-order initial-value problem:

$$y^{**}(x) = f(x, y, y^*), \quad y(x_0) = y_0 \quad \text{and} \quad y^*(x_0) = y_1. \tag{4.9}$$

This initial-value problem can be solved by solving the coupled system of first-order multiplicative differential equations

$$y_0^*(x) = y_1(x), \tag{4.10}$$

$$y_1^*(x) = f(x, y_0, y_1). \tag{4.11}$$

We want to solve the initial-value problem for the second-order multiplicative differential equation

$$y^{**}(x) = e. \tag{4.12}$$

The corresponding second-order ordinary differential equation is

$$y''(x) = \frac{y'(x)^2}{y(x)} + y(x). \tag{4.13}$$

The general solution of the differential equations (4.12) and (4.13) is

$$y(x) = \alpha \exp\left\{\frac{x^2}{2} + \beta x\right\}. \tag{4.14}$$

This initial-value problem was solved also as an example of a multiplicative boundary-value problem in [25]. In order to be able to compare the results with the multiplicative finite-difference methods solution, discussed in [25], we select $\alpha = 1$, $\beta = 1$, $x_0 = 1$, and $h = 0.25$, resulting in the initial conditions

$$y_0 = e^{3/2} \quad \text{and} \quad y_1 = e^2, \tag{4.15}$$

and compare the results in Table 3.

Table 3 shows the numerical approximation using the fourth-order multiplicative Runge–Kutta method for (4.12) with the initial conditions (4.15) and the corresponding results for the multiplicative finite-difference method from [25]. In this case we can see that the multiplicative Runge–Kutta method is slightly better than the multiplicative finite-difference method by an order of magnitude in the relative error.

TABLE 3. *Comparison of the multiplicative Runge–Kutta method and multiplicative finite-difference method.*

x	y_{exact}	y_{MRK}	Relative err _{MRK} in %	y_{MFD}	Relative err _{MFD} in %
1	4.48168907	4.481689070	0	4.48168907	0
1.25	7.62360992	7.62360992	9.3×10^{-15}	7.62360991	3.5×10^{-13}
1.5	13.80457419	13.80457419	1.3×10^{-14}	13.80457418	5.3×10^{-13}
1.75	26.60901319	26.60901319	1.7×10^{-14}	26.60913187	1.8×10^{-13}

On the other hand, if we solve the corresponding ordinary differential equation (4.13) with the corresponding initial values

$$y_0 = e^{3/2} \quad \text{and} \quad y_1 = 2e^{3/2}, \tag{4.16}$$

we get the results as shown in Table 4. Obviously, the ordinary Runge–Kutta method fails drastically in this case, as the relative error differs by 13 orders in magnitude compared to its multiplicative counterpart. Both the multiplicative Runge–Kutta method and the multiplicative finite-difference method succeed in producing proper results. Also in this case the performance of the multiplicative Runge–Kutta method method is significantly better than the ordinary Runge–Kutta method.

TABLE 4. Comparison of the multiplicative and ordinary Runge–Kutta methods.

x	y_{exact}	y_{MRK}	Relative err_{MRK} in %	y_{Newt}	Relative err_{Newt} in %
1	4.48168907	4.481689070	0	4.48168907	0
1.25	7.62360992	7.62360992	9.3×10^{-15}	7.61823131	7.1×10^{-2}
1.5	13.80457419	13.80457419	1.3×10^{-14}	13.77941017	1.8×10^{-1}
1.75	26.60901319	26.60901319	1.7×10^{-14}	26.51619718	3.5×10^{-1}

5. Conclusion

Having given a short motivation for the problem in the introduction, we described the multiplicative Runge–Kutta method for the solution of multiplicative initial-value problems of the form

$$y^*(x) = f(x, y) \quad \text{with} \quad y(x_0) = y_0,$$

where x_0 is the starting point and y_0 the initial value. The derivation of the second-order multiplicative Runge–Kutta method was carried out explicitly in detail. For the higher-order methods the ansätze, solutions, and corresponding Butcher tableaux are presented. The methods most successful at overcoming the restrictions of multiplicative calculus are presented in § 2.4. These methods ensure that the multiplicative Runge–Kutta method is a universally applicable tool. The error analysis and the convergence of multiplicative one-step methods were discussed in detail in § 3. The stability analysis closed the theoretical part of this study. Finally, the multiplicative Runge–Kutta method was applied to several problems, and the results were compared with the results from the ordinary Runge–Kutta method and the multiplicative finite-difference method. We saw that in these examples the multiplicative Runge–Kutta method produces significantly better results for the same step width than the ordinary Runge–Kutta method. Furthermore, the performance of both methods was compared explicitly for one example. We observed that the multiplicative Runge–Kutta method produced smaller errors for the same computation time than the ordinary Runge–Kutta method, demonstrating the universal applicability of the proposed method. The multiplicative Runge–Kutta method was also applied to the solution of a bacterial growth model proposed by Baranyi and compared to the ordinary Runge–Kutta method, with similar results.

References

1. D. ANISZEWSKA, ‘Multiplicative Runge–Kutta methods’, *Nonlinear Dynam.* 50 (2007) no. 1, 265–272.
2. J. BARANYI and T. A. ROBERTS, ‘A dynamic approach to predicting bacterial growth in food’, *Int. J. Food Microbiol.* 23 (1994) no. 3, 277–294.

3. J. BARANYI and T. A. ROBERTS, 'Mathematics of predictive food microbiology', *Int. J. Food Microbiol.* 26 (1995) no. 2, 199–218.
4. A. BASHIROV and M. RIZA, 'Complex multiplicative calculus', Preprint, 2011, [arXiv:1103.1462v1](https://arxiv.org/abs/1103.1462v1).
5. A. E. BASHIROV, E. M. KURPINAR and A. ÖZYAPICI, 'Multiplicative calculus and its applications', *J. Math. Anal. Appl.* 337 (2008) no. 1, 36–48.
6. A. E. BASHIROV, E. MISIRLI, Y. TANDOĞDU and A. ÖZYAPICI, 'On modeling with multiplicative differential equations', *Appl. Math. J. Chinese Univ.* 26 (2011) no. 4, 425–438.
7. A. E. BASHIROV and M. RIZA, 'On complex multiplicative differentiation', *TWMS J. Appl. Eng. Math.* 1 (2011) no. 1, 51–61.
8. J. C. BUTCHER, 'A stability property of implicit Runge–Kutta methods', *BIT* 15 (1975) no. 4, 358–361.
9. F. CÓRDOVA-LEPE, 'The multiplicative derivative as a measure of elasticity in economics', *TMAT Revista Latinoamericana de Ciencias e Ingenieria* 2 (2006) no. 3.
10. G. G. DAHLQUIST, 'A special stability problem for linear multistep methods', *BIT* 3 (1963) no. 1, 27–43.
11. B. L. EHLE, 'On Pade approximations to the exponential function and A-stable methods for the numerical solution of initial value problems', PhD Thesis, University of Waterloo, 1969.
12. L. FLORACK, 'Regularization of positive definite matrix fields based on multiplicative calculus', *Scale space and variational methods in computer vision*, Lecture Notes in Computer Science 6667 (eds A. M. Bruckstein, B. M. Haar Romeny, A. M. Bronstein and M. M. Bronstein; Springer, Berlin, 2012) 786–796.
13. M. GROSSMAN, *Bigeometric calculus, A system with a scale-free derivative* (Archimedes Foundation, Rockport, MA, 1983).
14. M. GROSSMAN and R. KATZ, *Non-Newtonian calculus* (Lee Press, Pigeon Cove, MA, 1972).
15. L. HUANG, 'Growth kinetics of *Listeria monocytogenes* in broth and beef frankfurters—determination of lag phase duration and exponential growth rate under isothermal conditions', *J. Food Sci.* 73 (2008) no. 5, E235–E242.
16. L. HUANG, 'Growth kinetics of *Escherichia coli* O157:H7 in mechanically-tenderized beef', *Int. J. Food Microbiol.* 140 (2010) no. 1, 40–48.
17. L. HUANG, 'Optimization of a new mathematical model for bacterial growth', *Food Control* 32 (2012) no. 1, 283–288.
18. A. Q. M. KHALIQ and E. H. TWIZELL, 'Stability regions for one-step multiderivative methods in PECE mode with application to stiff systems', *Int. J. Comput. Math.* 17 (1985) no. 3–4, 323–338.
19. E. MISIRLI and Y. GUREFE, 'Multiplicative Adams Bashforth–Moulton methods', *Numer. Algorithms* 57 (2011) no. 4, 425–439.
20. A. ÖZYAPICI and E. M. KURPINAR, 'Exponential approximation on multiplicative calculus', *6th ISAAC Congress* (2007) 471.
21. A. ÖZYAPICI and E. M. KURPINAR, 'Notes on multiplicative calculus', *20th International Congress of the Jangjeon Mathematical Society*, vol. 67 (2008), available from <http://icjms20.home.uludag.edu.tr/abstracts.pdf>.
22. A. ÖZYAPICI, M. RIZA, B. BILGEHAN and A. E. BASHIROV, 'On multiplicative and Volterra minimization methods', *Numer. Algorithms* (2013) 1–14.
23. A. PROTHERO and A. ROBINSON, 'On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations', *Math. Comp.* 28 (1974) no. 125, 145–162.
24. M. RIZA and B. EMINAĞA, 'Bigeometric calculus—a modelling tool', Preprint, 2014, [arXiv:1402.2877](https://arxiv.org/abs/1402.2877).
25. M. RIZA, A. ÖZYAPICI and E. MISIRLI, 'Multiplicative finite difference methods', *Quart. Appl. Math.* 67 (2009) no. 4, 745.
26. J. STOER and R. BULIRSCH, *Introduction to numerical analysis*, 2nd edn (Springer, New York, 1993).
27. A. ÜZER, 'Multiplicative type complex calculus as an alternative to the classical calculus', *Comput. Math. Appl.* 60 (2010) no. 10, 2725–2737.
28. V. VOLTERRA and B. HOSTINSKY, *Opérations infinitésimales linéaires* (Gauthier-Villars, Paris, 1938).

Mustafa Riza
 Department of Physics
 Eastern Mediterranean University
 Gazimağusa, North Cyprus
 via Mersin 10
 Turkey

mustafa.riza@emu.edu.tr

Hatice Aktöre
 Department of Mathematics
 Eastern Mediterranean University
 Gazimağusa, North Cyprus
 via Mersin 10
 Turkey

hatice.aktore@emu.edu.tr