

# New Numerical Integration Formulations for Ordinary Differential Equations

Serdar Beji

Faculty of Naval Architecture and Ocean Engineering, Istanbul Technical University, Istanbul, Türkiye

Email: sbeji@itu.edu.tr

**How to cite this paper:** Beji, S. (2024) New Numerical Integration Formulations for Ordinary Differential Equations. *Advances in Pure Mathematics*, 14, 650-666.  
<https://doi.org/10.4236/apm.2024.148036>

**Received:** June 11, 2024

**Accepted:** August 23, 2024

**Published:** August 26, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

An entirely new framework is established for developing various single- and multi-step formulations for the numerical integration of ordinary differential equations. Besides polynomials, unconventional base-functions with trigonometric and exponential terms satisfying different conditions are employed to generate a number of formulations. Performances of the new schemes are tested against well-known numerical integrators for selected test cases with quite satisfactory results. Convergence and stability issues of the new formulations are not addressed as the treatment of these aspects requires a separate work. The general approach introduced herein opens a wide vista for producing virtually unlimited number of formulations.

## Keywords

Single- and Multi-Step Numerical Integration, Unconventional Base-Functions, Ordinary Differential Equations

## 1. Introduction

The earliest methods for numerical solutions of ordinary differential equations may be traced back to Euler (1707-1783), arguably the most productive and inventive mathematician of all time. Butcher [1] presents an excellent treatise on numerical solutions of differential equations, covering the entire literature with original contributions. The origin of multi-step integration techniques dates back to 1883, the year Bashforth and Adams [2] published a theoretical and numerical study of capillary action on drops of liquids, in which probably the first multi-step formulation was introduced to numerically integrate the differential equation considered there. Runge [3], prior to Kutta [4], laid out the formulation of a very robust and accurate numerical integration scheme known today as the Runge-Kutta method. Based on the experiences of his computational works, Milne [5] gave a multi-step formulation. An ingenious approach to evaluating

derivatives for zero-step size via rational polynomials was developed by Bulirsch and Stoer [6] for the numerical integration of differential equations. All these fundamental works have given rise to a vast literature today, a good account of which may be seen in Butcher [1].

Among the contemporary works, Fatimah *et al.* [7] and Ogunrinde and Olubunmi [8] established algorithms to solve second-order linear differential equations while Signes-Pont *et al.* [9] gave an interesting treatment for ordinary differential equations.

The present work begins with the usual approach of integrating an initial value problem but differs substantially by introducing the so-called *base* or *extrapolating* functions, which are not necessarily polynomials. Furthermore, in determining the coefficients of the base functions not only definite points but also derivatives are employed. In this manner, the accuracy of *curve-fitting* process is improved to yield better computational results. Finally, removing the restriction to use only polynomials opens a variety of possibilities to develop virtually limitless new formulations suitable to the specific problems in hand.

## 2. Numerical Integration Approach

The general approach to establishing a numerical integration formulation may be outlined as follows. Let a first-order differential equation with a prescribed initial value  $y_0$  at  $x_0$  be described as

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0 \quad (1)$$

where  $x_0, y_0$ , and  $f(x, y)$  are all given. In order to perform an analytical integration with respect to the independent variable  $x$ ,  $f(x, y)$  must be expressed as an integrable function of  $x$  alone. Typically, this is achieved by expressing  $f(x, y)$  as a polynomial producing identical values with  $f(x, y)$  for definite  $(x, y)$  pairs within a relatively small interval of  $x$  already known or computed. Here, this approximating function is denoted by  $g(x)$  and is not restricted to polynomials only. Any function judged to represent  $f(x, y)$  suitably may be used as an *extrapolating* or *basis function*. Further, the determination of unknown constants of  $g(x)$  within an interval is not restricted to point-by-point satisfaction of selected  $f(x, y)$  values; depending on the formulation,  $f'(x, y)$ ,  $f''(x, y)$ , etc. derivative values may be used too.

Supposing that a function  $g(x)$  is established to represent  $f(x, y)$  within the relatively small interval of  $\Delta x$  then integrating Equation (1) from  $x_i$  to  $x_{i+1}$  gives

$$\int_{y_i}^{y_{i+1}} dy \approx \int_{x_i}^{x_{i+1}} g(x) dx \quad (2)$$

where  $x_{i+1} - x_i = \Delta x$  and  $y(x_i) = y_i$  is the presently known numerical solution of the differential equation and  $y(x_{i+1}) = y_{i+1}$  is the newly computed value by the use of base function  $g(x)$  within the interval  $\Delta x$ . A refinement to  $y_{i+1}$  may be introduced by redefining the coefficients of the interpolating function

$g(x)$  in an interval shifted by  $\Delta x$  to include the newly computed  $y_{i+1}$  as the *predicted* value.

Relevant literature, either for single- or multi-step formulations, is essentially based on the use of polynomial interpolating functions. For instance, the lowest-order formulation of Euler, which may be viewed as the basis of all elaborated methods of its kind, assumes a constant base function of the form  $g(x) = a_i$ , and upon satisfying the point  $x_i$  locally as  $g(0) = a_i = f_i$ , produces  $y_{i+1} = y_i + \int_0^{\Delta x} g(x) dx = y_i + f_i \Delta x$ , which is the simplest formula of all. Taylor series method on the other hand takes  $g(x) = a_0 + a_1 x + \dots + a_{n-1} x^{(n-1)}$  and satisfies  $g(0) = 0! a_0 = f_i$ ,  $g'(0) = 1! a_1 = f_i'$ ,  $\dots$ ,  $g^{(n-1)}(0) = (n-1)! a_{(n-1)} = f_i^{(n-1)}$  thus establishing the base function as  $g(x) = f_i + f_i' x + \frac{1}{2!} f_i'' x^2 + \dots + \frac{1}{(n-1)!} f_i^{(n-1)} x^{n-1}$ .

Integrating  $g(x)$  and using the standard notation,

$y_i' = f_i, y_i'' = f_i', \dots, y_i^{(n)} = f_i^{(n-1)}$ , produces

$y_{i+1} = y_i + y_i' \Delta x + \frac{1}{2!} y_i'' (\Delta x)^2 + \dots + \frac{1}{n!} y_i^{(n)} (\Delta x)^n$  in terms of  $y$  and its derivatives.

Thus, an  $n^{\text{th}}$  order Taylor series expansion corresponds to satisfying a point and  $n$ -derivatives of an  $n^{\text{th}}$  order polynomial at the current point of integration. Finally, the multi-step methods use polynomial base functions to satisfy several discrete values of the right-hand side of Equation (1) to obtain  $g(x)$  and proceed with integration stated in (2). The most popular multi-step formulation is known as the Adams-Moulton method and is based on a third-order polynomial curve-fitting that satisfies four discrete values of  $f(x, y)$ .

This work in principle unites the approaches outlined above without being restricted to polynomial functions alone and considers any suitable and analytically integrable function as an eligible base function. On the other hand, among a variety of choices; trigonometric functions, exponential functions, and polynomials have two major advantages as extrapolating functions. First, a large class of linear or nonlinear ordinary differential equations has solutions expressed by sinusoidal, exponential, or polynomial forms. Second, these particular functions are analytically integrable, which is an essential requirement to establish a numerical integration formulation. Thus, the present work concentrates on sinusoidal, exponential, and polynomial forms and their combinations. Nevertheless, the choice of base functions should not be restricted by these types of functions alone; the functions best suitable to the class of problems treated may freely be selected as long as they are analytically integrable.

### 3. Trigonometric Base Functions

In this section, trigonometric functions in different arrangements are used to produce several single- and multi-step integration schemes.

#### 3.1. TBF-2C:1P1D $g(x) = a_i \cos x + b_i \sin x$

We begin with an extrapolating or base function of the form

$g(x) = a_i \cos x + b_i \sin x$ , which has two parameters  $a_i, b_i$  to be determined at

each step  $i$  of numerical integration. The coefficients are obtained by satisfying two conditions hence the method is abbreviated as TBF-2C:1P1D; that is, a trigonometric base function with coefficients determined from two conditions: one point and one derivative.

Employing a local coordinate system moving with the current integration point as shown in **Figure 1** so that  $\bar{x} = x - x_i$  hence at the point  $x = x_i$ ,  $\bar{x} = x_i - x_i = 0$ , the following equalities are written at each step

$$g(0) = a_i \cos 0 + b_i \sin 0 = f_i \Rightarrow a_i = f_i \tag{3a}$$

$$g'(0) = -a_i \sin 0 + b_i \cos 0 = f'_i \Rightarrow b_i = f'_i \tag{3b}$$

where the prime indicates differentiation with respect to the independent variable  $x$  while  $f_i = f(x_i, y_i)$  and  $f'_i = f'(x_i, y_i)$  stand respectively for the right-hand side of Equation (1) and its first derivative evaluated at the current point  $(x_i, y_i)$  of integration.

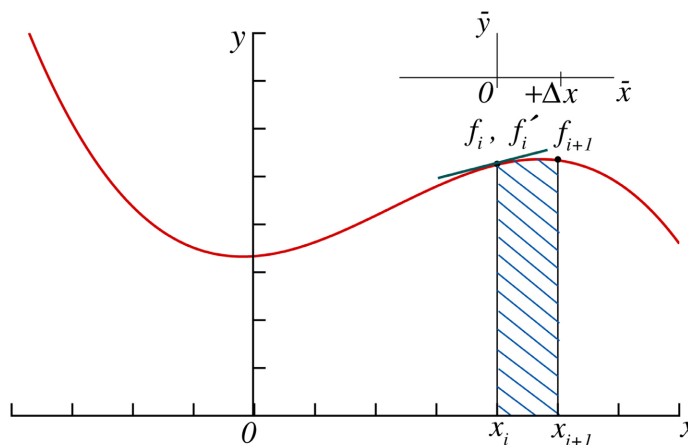
Having established the coefficients locally, the integration is carried out by substituting the proposed base function  $g(\bar{x}) = f_i \cos \bar{x} + f'_i \sin \bar{x}$  into Equation (2)

$$\int_{y_i}^{y_{i+1}} dy = \int_0^{\Delta x} g(\bar{x}) d\bar{x} = \int_0^{\Delta x} (f_i \cos \bar{x} + f'_i \sin \bar{x}) d\bar{x}$$

$$y_{i+1} - y_i = f_i \sin \bar{x} - f'_i \cos \bar{x} \Big|_0^{\Delta x} = f_i \sin \Delta x + f'_i(1 - \cos \Delta x) \tag{4}$$

which in turn yields the following single-step numerical integration formula for the integration step  $\Delta x$

$$y_{i+1} = y_i + f_i \sin \Delta x + f'_i(1 - \cos \Delta x) \tag{5}$$



**Figure 1.** Sketch of using a point  $f_i$  and a derivative  $f'_i$  for base function  $g(x) = a_i \cos x + b_i \sin x$ .

Note that since  $\Delta x$  is a fixed value,  $\sin \Delta x$  and  $\cos \Delta x$  need not be computed repeatedly at each step and the simple form of (5) results in very efficient and fast computations. An important detail concerns the use of a dimensional quantity,  $\Delta x$ , as the argument of sine and cosine functions. This problem may be overcome

by assuming a length scale, say  $\lambda$ , which can be used to non-dimensionalize the differential equation and then set to unit length  $\lambda = 1$ . In case of time increment  $\Delta t$ , the same process may be carried out with a time scale  $\tau$  which, eventually is to be set to unity. If the problem involves appreciable length or time scales non-dimensionalization may be carried out with an appropriate scaling quantity different from unity. On the other hand, trial computations with scaled equations revealed that the results are unaffected by the numerical value of scaling quantity and therefore use of Equation (5) in its present form poses no problems.

This simple integration formula, which is expected to perform remarkably well for differential equations involving trigonometric functions, is now tested for the following initial value problem,

$$\frac{dy}{dx} = \cos x, \quad y(0) = 0, \quad (6)$$

which has the exact solution  $y(x) = \sin x$ , satisfying  $y(0) = 0$ .

**Table 1.** Exact and numerical solutions of  $dy/dx = \cos x$  for selected  $x$  values within  $0 \leq x \leq 50$ .

$x$	Exact	TBF-2C:1P1D	Runge-Kutta
0.0	-0.0000000	-0.0000000	-0.0000000
0.5	-0.4794255	-0.4794255	-0.4794360
1.0	-0.8414710	-0.8414710	-0.8414894
2.5	-0.5984721	-0.5984721	-0.5984852
5.0	-0.9589243	-0.9589243	-0.9589452
10.0	-0.5440211	-0.5440211	-0.5440330
20.0	-0.9129453	-0.9129452	-0.9129652
50.0	-0.2623748	-0.2623749	-0.2623807

**Table 1** lists the exact solution and numerical integration results at selected  $x$  values using Equation (5), which is indicated as TBF-2C:1P1D, and the fourth-order Runge-Kutta method for step size  $\Delta x = 0.5$ . Because of the perfect match between the base function and the solution of the differential equation considered, the performance of TBF-2C:1P1D is perfect even for the relatively large step size of  $\Delta x = 0.5$ . This observation reveals the importance of using a base function in accord with the characteristics of the problem in hand. Since all the existing numerical integration schemes use either Taylor series expansions or polynomials, no such arguments were brought up previously. In this respect, the present work opens a new perspective to develop a virtually unlimited number of schemes employing different base functions and satisfying different conditions such as points, first, second, and higher derivatives. Even further, the exact satisfaction of some or all the imposed conditions by base functions may be abandoned in favor of approximate ones such as the least-square method. In this

way, more conditions (points, derivatives) than the number of coefficients of the base function can be satisfied. Naturally, such varieties must first be investigated regarding their merits or demerits.

Relatively poor performance of Runge-Kutta method for solving Equation (6) is by no means an indication of a serious weakness of this particular method; the errors are essentially due to the large step size used. Likewise, the unusual success of TBF-2C:1P1D formulation, originating basically from the good matching of the base function with the solution characteristics, should not be seen as a precursor of excellent performance for all types of problems. To demonstrate this particular aspect, a differential equation of aperiodic characteristics is considered.

$$\frac{dy}{dx} = \frac{2}{\sqrt{\pi}} e^{-x^2}, \quad y(0) = 0, \quad (7)$$

Solution of (7) is the well-known error function  $y(\bar{x}) = \text{erf}(\bar{x}) = \frac{2}{\sqrt{\pi}} \int_0^{\bar{x}} e^{-x^2} dx$ .

Numerical integration is carried out for  $\Delta x = 0.05$  and the computational results are listed together with the tabulated values of Abramowitz and Stegun [10] in **Table 2**. Despite a ten times smaller step size, TBF-2C:1P1D formulation is now inferior to Runge-Kutta. More importantly, Runge-Kutta keeps virtually the same order of accuracy when the step size doubles to  $\Delta x = 0.1$  while TBF-2C:1P1D deteriorates even more. Values computed for  $\bar{x} = 50.00$  are regarded to correspond  $+\infty$ , as no change is observed in computations after  $\bar{x} = 3.75$ .

**Table 2.** Tabulated and numerical solutions of  $\frac{dy}{dx} = \frac{2}{\sqrt{\pi}} e^{-x^2}$ ,

$$y(\bar{x}) = \text{erf}(\bar{x}) = \frac{2}{\sqrt{\pi}} \int_0^{\bar{x}} e^{-x^2} dt.$$

$\bar{x}$	Abramowitz & Stegun	TBF-2C:1P1D	Runge-Kutta
0.00	0.0000000	0.0000000	0.0000000
0.25	0.2763264	0.2764338	0.2763264
0.50	0.5204999	0.5206550	0.5204999
0.75	0.7111556	0.7112712	0.7111557
1.00	0.8427008	0.8427080	0.8427009
1.25	0.9229001	0.9227743	0.9229002
1.50	0.9661051	0.9658622	0.9661052
1.75	0.9866717	0.9863465	0.9866717
2.00	0.9953223	0.9949495	0.9953224
$+\infty$	1.0000000	0.9995893	1.0000000

The preceding demonstrations, tabulated in **Table 1** and **Table 2**, contradicting each other in terms of the accuracy of the integration methods, reveal clearly

that the characteristics of the problem considered play an important role in the performance of the numerical integrator. Characteristics of the integration method on the other hand depend strictly on the underlying base functions used for the formulation. Consequently, it may be suggested to select a numerical integrator with an extrapolating function that accords well with the nature of the differential equation(s) to be solved.

### 3.2. TBF-3C:3P $g(x) = a_i \cos x + b_i \sin x + c_i$

A base function that differs only slightly by a constant from the one used in §3.1 is now considered. Satisfying three consecutive points makes the present approach a multi-step formulation, thus placing it in a distinctly different category from the single-step approach of TBF-2C:1P1D. Using the same terminology, the present formulation is abbreviated as TBF-3C:3P; namely, a trigonometric base function with coefficients determined from three conditions; more specifically, by satisfying three consecutive points of  $f(x, y)$ . Unknown coefficients  $a_i$ ,  $b_i$ , and  $c_i$  are obtained from the following linear equations

$$g(+\Delta x) = a_i \cos \Delta x + b_i \sin \Delta x + c_i = f_i \quad (8a)$$

$$g(0) = a_i \cos 0 + b_i \sin 0 + c_i = f_{i-1} \quad (8b)$$

$$g(-\Delta x) = a_i \cos(-\Delta x) + b_i \sin(-\Delta x) + c_i = f_{i-2} \quad (8c)$$

Note that the origin of the moving coordinate system  $(\bar{x}, \bar{y})$  shown in **Figure 1** is now placed at  $x_{i-1}$  so that  $x_i$  corresponds to  $+\Delta x$  and  $x_{i-2}$  to  $-\Delta x$ . Solving the above set of equations yields

$$a_i = \frac{-f_i + 2f_{i-1} - f_{i-2}}{2(1 - \cos \Delta x)}, \quad b_i = \frac{f_i - f_{i-2}}{2 \sin \Delta x}, \quad c_i = f_{i-1} - a_i \quad (9)$$

Carrying out the integration of the differential equation from  $\Delta x$  to  $2\Delta x$  results in

$$y_{i+1,p} = y_i + a_i (\sin 2\Delta x - \sin \Delta x) - b_i (\cos 2\Delta x - \cos \Delta x) + c_i \Delta x \quad (10)$$

which is called the predicted value  $y_{i+1,p}$  as we intend to improve the computations via a corrective step, which is achieved by shifting the entire process in (8) to the right by a step size  $\Delta x$  and making use of  $y_{i+1,p}$  for computing  $f_{i+1,p}$ . Thus, the corrected values  $y_{i+1,c}$  are computed from the following equations.

$$y_{i+1,c} = y_i + a_i (\sin 2\Delta x - \sin \Delta x) - b_i (\cos 2\Delta x - \cos \Delta x) + c_i \Delta x \quad (11)$$

where

$$\begin{aligned} a_i &= \frac{-f_{i+1,p} + f_{i-1} + 2(f_i - f_{i-1}) \cos \Delta x}{2(1 - \cos \Delta x)} \\ b_i &= \frac{f_i - f_{i-1} + a_i (1 - \cos \Delta x)}{\sin \Delta x} \\ c_i &= f_{i-1} - a_i \end{aligned} \quad (12)$$

$f_{i+1,p}$  is computed from  $f(x_{i+1}, y_{i+1,p})$  by making use of the predicted value  $y_{i+1,p}$  given in (10). Note that the form of (11) remains the same as (10) since

the integration range is unchanged; only the coefficients are redefined because in the corrective stage  $f_{i-1}, f_i, f_{i+1,p}$  are used for determining  $a_i, b_i, c_i$  instead of  $f_{i-2}, f_{i-1}, f_i$ . Unlike the single-step formulation TBF-2C:1P1D, this scheme is not self-starting; it requires the values of two previous steps,  $y_{i-2}, y_{i-1}$  besides the known current step  $y_i$ , in order to proceed. For this and similar schemes the Taylor series method, typically to the sixth to eighth-order of accuracy, is employed here to generate the required starting values.

The performance of TBF-3C:3P is very similar to that of TBF-2C:1P1D; therefore, to avoid repetition, comparisons for test cases are not included. It is also worthwhile to point out that the corrective step, Equation (11), does not introduce appreciable corrections and may be omitted completely. In view of these performance characteristics, the simplicity and efficiency of TBF-2C:1P1D definitely make it a better choice for problems of periodic nature.

### 3.3. TBF-4C:2P2D $g(x) = a_i \cos x + b_i \sin x + c_i x + d_i$

TBF-4C:2P2D, a trigonometric base function satisfying four conditions: two points and two derivatives, is the last sample scheme given in this category here. The formulation is only outlined since the details can be constructed with the help of previous derivations. To give a glimpse of further possibilities a few more schemes employing trigonometric functions, already developed and tested, are described at the end of this section. TBF-4C:2P2D with the base function  $g(x) = a_i \cos x + b_i \sin x + c_i x + d_i$  satisfies the following conditions

$$g(0) = f_i, \quad g'(0) = f'_i, \quad g(-\Delta x) = f_{i-1}, \quad g'(-\Delta x) = f'_{i-1} \quad (13)$$

The coefficients are determined as

$$\begin{aligned} a_i &= \frac{-f'_i + f'_{i-1} + b_i(1 - \cos \Delta x)}{\sin \Delta x} \\ b_i &= \frac{(f'_i - f'_{i-1})(1 - \cos \Delta x) + (f_i - f_{i-1} - f'_i \Delta x) \sin \Delta x}{2(1 - \cos \Delta x) - \Delta x \sin \Delta x} \\ c_i &= f'_i - b_i \\ d_i &= f_i - a_i \end{aligned} \quad (14)$$

which minimizes the number of computations but  $b_i$  must be computed first. Note also that denominators do not become zero unless  $\Delta x$  is zero and that  $\cos \Delta x$  and  $\sin \Delta x$  are to be computed once as  $\Delta x$  is constant. Integrating  $g(x)$  from 0 to  $+\Delta x$  yields the required expression for  $y_{i+1}$ :

$$y_{i+1} = y_i + a_i \sin \Delta x + b_i (1 - \cos \Delta x) + \frac{1}{2} c_i (\Delta x)^2 + d_i \Delta x \quad (15)$$

A predictor-corrector type scheme is not pursued for this and the remaining formulations since the effect is virtually unobservable. Formulation (15) performs quite satisfactorily as demonstrated for the following set of *stiff* differential equations (Chapra and Canale [11], p. 758), which is selected especially as a challenging case.

$$\frac{dy_1}{dx} = -5y_1 + 3y_2 \quad (16a)$$

$$\frac{dy_2}{dx} = 100y_1 - 301y_2 \quad (16b)$$

For the initial conditions  $y_1(0) = 52.29$  and  $y_2(0) = 83.82$ , the exact solutions are

$$y_1 = 52.96e^{-3.9899x} - 0.67e^{-302.0101x}, \quad y_2 = 17.83e^{-3.9899x} + 65.99e^{-302.0101x} \quad (17)$$

**Table 3** shows the exact values computed from (17) against the present formulation (15) and Runge-Kutta fourth-order classic scheme. Computations were done for a very small step size  $\Delta x = 1/1000$  because TBF-4C:2P2D failed to work for  $\Delta x > 1/800$  while Runge-Kutta did somewhat better and failed if  $\Delta x > 1/500$ . Only some selected results are given in **Table 3**; however, at a glance it is clear that TBF-4C:2P2D performs virtually the same as Runge-Kutta does. This is quite encouraging because the problem considered, besides being a *stiff* set of differential equations, is aperiodic and has a tendency to become unstable.

**Table 3.** Exact and numerical solutions of a set of stiff equations.

	Exact	TBF-4C:2P2D	Runge-Kutta	Exact	TBF-4C:2P2D	Runge-Kutta
$x$	$y_1(x)$	$y_1(x)$	$y_1(x)$	$y_2(x)$	$y_2(x)$	$y_2(x)$
0.00	52.289999	52.289999	52.289999	83.819998	83.819998	83.819998
0.10	35.536021	35.533585	35.533585	11.963883	11.963764	11.963764
0.20	23.844578	23.842864	23.842864	8.027735	8.027628	8.027628
0.50	7.203642	7.203053	7.203053	2.425244	2.425188	2.425188
1.00	0.979843	0.979746	0.979746	0.329883	0.329870	0.329870
1.50	0.133279	0.133263	0.133263	0.044871	0.044868	0.044868
2.00	0.018129	0.018126	0.018126	0.006103	0.006103	0.006103
3.00	0.000335	0.000335	0.000335	0.000113	0.000113	0.000113
4.00	0.000006	0.000006	0.000006	0.000002	0.000002	0.000002
5.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Relatively good performances of formulations TBF-2C:1P1D and TBF-4C:2P2D essentially originate from the use of derivatives in determining the coefficients of the base functions. Satisfying definite points alone does not suffice for a good enough curve-fitting, as this process may be termed. In the following sections this point is further demonstrated by making use of even higher derivatives.

Treatment of trigonometric base functions is concluded here although more formulations and their schemes, such as TBF-3C:1P2D and TBF-3C:2P1D, both using the same base function  $g(x) = a_i \cos x + b_i \sin x + c_i$ , are developed. Furthermore, trigonometric base functions need not be restricted to basic forms; *analytically integrable functions* such as  $\cos^2 x$ ,  $x^2 \sin x^3$ ,  $\sin 5x$ , etc. may all be used, provided that they are found appropriate to the problem in hand. Ar-

guments with unknown constants like  $\alpha_i$  in  $\cos \alpha_i x$  should be avoided as determination of  $\alpha_i$  would be both difficult and problematic due to potential zero denominator values. Despite such restrictions, virtually countless number of different *integrable* base or extrapolating functions may be proposed. Next section considers exponential base functions as another candidate in establishing numerical integration formulations.

## 4. Exponential Base Functions

In this section, a single- and a multi-step integration scheme employing exponential functions are presented.

### 4.1. EBF-2C:1P1D $g(x) = a_i e^x + b_i$

First, the simplest possible case for the exponential base functions,  $g(x) = a_i e^x + b_i$  is considered. Satisfying a single point and a derivative at the current point of calculation  $x = x_i$ , which is taken to correspond to  $\bar{x} = 0$  for the moving coordinate system depicted in **Figure 1**, gives

$$g(0) = a_i + b_i = f_i, \quad g'(0) = a_i = f_i' \quad (18)$$

which in turn results in the following single-step formulation

$$y_{i+1} = y_i + f_i'(e^{\Delta x} - 1) + (f_i - f_i')\Delta x \quad (19)$$

for the numerical integration of  $dy/dx = f(x, y)$ . Note that  $e^{\Delta x} - 1$  need be computed only once and the total number of operations in each step of (19) is only five; namely, subtracting  $f_i'$  from  $f_i$ , multiplying  $f_i'$  by  $e^{\Delta x} - 1$  and  $(f_i - f_i')$  by  $\Delta x$ , and finally making two additions to  $y_i$ . **Table 4** shows computed values of the exact solution, EBF-2C:1P1D, and Runge-Kutta for the differential equation  $dy/dx = \cos x$  with  $y(0) = 0$  for some selected  $x$  values. Although EBF-2C:1P1D stably works for as large as  $\Delta x = 1/2$ , **Table 4** is constructed from computations with  $\Delta x = 1/100$  for accurate enough results.

**Table 4.** Exact and numerical solutions of  $dy/dx = \cos x$ ,  $y(0) = 0$ .

$x$	Exact	EBF-2C:1P1D	Runge-Kutta
0.0	-0.0000000	-0.0000000	-0.0000000
0.5	-0.4794255	-0.4794314	-0.4794255
1.0	-0.8414710	-0.8414769	-0.8414709
2.0	-0.9092974	-0.9092875	-0.9092976
3.0	-0.1411200	-0.1410871	-0.1411203
4.0	-0.7568025	-0.7568443	-0.7568021
5.0	-0.9589243	-0.9589530	-0.9589240
10.0	-0.5440211	-0.5440625	-0.5440205
15.0	-0.6502879	-0.6502672	-0.6502884
20.0	-0.9129453	-0.9129496	-0.9129447

Runge-Kutta is clearly better with five-decimal-place accuracy maintained for the entire range of  $x$  values. On the other hand, four-decimal-place accuracy of the EPF-2C:1P1D, although inferior to that of Runge-Kutta, is very impressive in view of the extreme simplicity of the EPF-2C:1P1D scheme, Equation (19).

#### 4.2. EBF-4C:2P2D $g(x) = a_i e^x + b_i x^2 + c_i x + d_i$

The second case considered for the exponential base functions is  $g(x) = a_i e^x + b_i x^2 + c_i x + d_i$  which satisfies two points and two derivatives at consecutive points  $x_{i-1}$  and  $x_i$ , thus producing a multi-step scheme. Without entering into details the resulting formulation can be stated as follows.

$$y_{i+1} = y_i + a_i (e^{\Delta x} - 1) + \frac{1}{3} b_i (\Delta x)^3 + \frac{1}{2} c_i (\Delta x)^2 + d_i \Delta x \quad (20)$$

where

$$\begin{aligned} a_i &= \frac{2(f_i - f_{i-1}) - (f'_i + f'_{i-1})\Delta x}{(2 - \Delta x) - (2 + \Delta x)e^{-\Delta x}} \\ b_i &= \frac{a_i (e^{-\Delta x} - 1) + (f'_i - f'_{i-1})\Delta x}{2\Delta x} \\ c_i &= f'_i - a_i \\ d_i &= f_i - a_i \end{aligned} \quad (21)$$

**Table 5.** Exact and numerical solutions of  $dy/dx = \cos x$ ,  $y(0) = 0$ .

$x$	Exact	EBF-4C:2P2D	Runge-Kutta
0.0	-0.0000000	-0.0000000	-0.0000000
0.5	-0.4794255	-0.4794256	-0.4794255
1.0	-0.8414710	-0.8414709	-0.8414709
2.0	-0.9092974	-0.9092976	-0.9092976
3.0	-0.1411200	-0.1411204	-0.1411203
4.0	-0.7568025	-0.7568020	-0.7568021
5.0	-0.9589243	-0.9589241	-0.9589240
10.0	-0.5440211	-0.5440207	-0.5440205
15.0	-0.6502879	-0.6502880	-0.6502884
20.0	-0.9129453	-0.9129448	-0.9129447

Note that the denominators of  $a_i$  and  $b_i$  become zero only when  $\Delta x = 0$ . We now reconsider the test case given in §4.1 so that a comparison between EBF4C:2P2D and EBF2C:1P1D is also possible. Accordingly, **Table 5** lists computed values of the exact solution, EBF-4C:2P2D, and Runge-Kutta for the differential equation  $dy/dx = \cos x$  with  $y(0) = 0$  for some select  $x$  values. In order to make the comparisons consistent,  $\Delta x = 1/100$  although both schemes work for much greater step sizes without any problem.

Performances of both schemes are virtually the same and minimum five-decimal-place accuracy is attained. As indicated previously, the good performances of the schemes developed in this work may basically be attributed to satisfying not only  $f(x, y)$  but also its derivative  $f'(x, y)$  at selected points. Such an approach makes the curve-fitting more accurate and results in better formulations. Nevertheless, this should not be taken as the sole criterion; for instance, as an alternative to the present formulation, another EBF-4C:2P2D based on  $g(x) = a_i e^x + b_i e^{-x} + c_i x + d_i$  was formulated but the results were not as good as the present one. Furthermore, an attempt using  $g(x) = a_i e^{\alpha_i x} + b_i x + c_i$  likewise did not work well basically because the determination of  $\alpha_i$  caused difficulties despite some simplifying assumptions for its determination. This problem, which is mentioned at the end of §3.3, once more emphasizes the importance of avoiding unknown coefficients in the arguments of interpolating functions.

Finally, as for the trigonometric base functions, a variety of analytically integrable exponential functions such as  $x e^{-x^2}$ ,  $x^2 e^{x^3}$ , and even more elaborate ones,  $(\cos x) e^{\sin x}$ , may all be considered as long as they yield solvable robust systems for the unknown coefficients and work well for the problems in hand.

## 5. Polynomial Base Functions

The Adams-Moulton multi-step formulation, which is based on a cubic polynomial base function, is probably the most commonly used numerical integrator of its category. At each step the cubic polynomial satisfies four consecutive points  $(x_i, f_i)$ ,  $(x_{i-1}, f_{i-1})$ ,  $(x_{i-2}, f_{i-2})$ ,  $(x_{i-3}, f_{i-3})$  to interpolate and therefore requires integrated values of three previous steps besides the current step. The first three steps must be computed by a single-step formulation or Taylor series method in order to initiate the computations. The predictor-corrector version of the Adams-Moulton, which makes a supposedly improved estimate by use of the advanced interval  $(x_{i+1}, f_{i+1})$ ,  $(x_i, f_i)$ ,  $(x_{i-1}, f_{i-1})$ ,  $(x_{i-2}, f_{i-2})$ , is known as the Adams-Bashforth-Moulton (ABM) method. Considering that the Adams-Moulton method is the preferred multi-step approach we begin with a third-order polynomial as our base function but satisfy two points and two derivatives instead of four points. The use of only two consecutive steps and employing derivatives in the curve-fitting process is expected to improve the accuracy of this approach.

### 5.1. PBF-4C:2P2D $g(x) = a_i x^3 + b_i x^2 + c_i x + d_i$

In line with the cubic polynomial formulation of Adams-Moulton the base function is taken as  $g(x) = a_i x^3 + b_i x^2 + c_i x + d_i$  and required to satisfy two points  $g(0) = f_i$ ,  $g(-\Delta x) = f_{i-1}$ , and two derivatives  $g'(0) = f'_i$ ,  $g'(-\Delta x) = f'_{i-1}$ . As always, the local coordinate system depicted in **Figure 1** is used. Carrying on the now well-established procedure yields

**Table 6.** Exact and numerical solutions of  $dy/dx = -2xy$ ,  $y(0) = 1$ 

$x$	Exact	PBF-4C:2P2D	ABM	Runge-Kutta
0.0	0.0000000	1.0000000	1.0000000	1.0000000
0.5	0.7788008	0.7788008	0.7788003	0.7788008
1.0	0.3678795	0.3678779	0.3678341	0.3678811
1.5	0.1053992	0.1054003	0.1054051	0.1054056
2.0	0.0183156	0.0183168	0.0183307	0.0183225
2.5	0.0019305	0.0019303	0.0019293	0.0019334
3.0	0.0001234	0.0001232	0.0001209	0.0001240
3.5	0.0000048	0.0000048	0.0000042	0.0000049
4.0	0.0000001	0.0000001	0.0000001	0.0000001

$$y_{i+1} = y_i + \frac{1}{2}\Delta x \left[ -f_i + 3f_{i-1} + \frac{1}{6}\Delta x(17f'_i + 7f'_{i-1}) \right] \quad (22)$$

As in the ABM scheme it is possible to include a corrective step, which takes the form of  $y_{i+1,c} = y_i + \frac{1}{2}\Delta x \left[ f_{i+1,p} + f_i + \frac{1}{6}\Delta x(-f'_{i+1,p} + f'_i) \right]$ ; however, this extension was found unnecessary as remarked in §2. The performance of PBF-4C:2P2D is tested for the problem

$$\frac{dy}{dx} = -2xy, \quad y(0) = 1, \quad (23)$$

which has the exact solution  $y(x) = e^{-x^2}$ . **Table 6** compares PBF-4C:2P2D, ABM, and the fourth-order Runge-Kutta methods with the exact solution using the numerical computations made for  $\Delta x = 1/10$ . For this problem, PBF-4C:2P2D compares quite well with the exact solution and outperforms both the ABM and the classic fourth-order Runge-Kutta methods. In particular, the good performance of PBF-4C:2P2D against ABM is important because both formulations are based on third-order polynomials; they differ only in terms of conditions they satisfy in determining the polynomial coefficients.

### 5.2. PBF-6C:2P4D $g(x) = a_i x^5 + b_i x^4 + c_i x^3 + d_i x^2 + e_i x + h_i$

A fifth-order polynomial  $g(x) = a_i x^5 + b_i x^4 + c_i x^3 + d_i x^2 + e_i x + h_i$  is now required to satisfy two points  $g(0) = f_i$ ,  $g(-\Delta x) = f_{i-1}$ , two first derivatives  $g'(0) = f'_i$ ,  $g'(-\Delta x) = f'_{i-1}$ , and two second derivatives  $g''(0) = f''_i$ ,  $g''(-\Delta x) = f''_{i-1}$ . Carrying on the procedure results in

$$y_{i+1} = y_i + \frac{1}{10}\Delta x \left[ 75f_i - 65f_{i-1} - \Delta x(31f'_i + 29f'_{i-1}) + \frac{1}{12}(\Delta x)^2(111f''_i - 49f''_{i-1}) \right] \quad (24)$$

which may be considered as the predictor stage. Proceeding to the corrector stage gives  $y_{i+1,c} = y_i + \frac{1}{10}\Delta x \left[ 5f_{i+1,p} + 5f_i - \Delta x(f'_{i+1,p} - f'_i) + \frac{1}{12}(\Delta x)^2(f''_{i+1,p} + f''_i) \right]$ . For

the test case here, both the predictor and corrector stages are used. As indicated before, use of the corrector stage does not make appreciable improvements in the results; nevertheless, for this particular formulation it is found to make the scheme more stable. The performance of PBF-6C:2P4D, using both the predictor and corrector stages, is tested for the following set of equations.

$$\frac{dy_1}{dx} = y_2 - y_1 \quad (25a)$$

$$\frac{dy_2}{dx} = -2y_2 + \cos x \quad (25b)$$

For the initial conditions  $y_1(0) = 0$  and  $y_2(0) = 2$ , the exact solutions are

$$y_1 = xe^{-x} + \sin x, \quad y_2 = e^{-x} + \sin x + \cos x \quad (26)$$

**Table 7.** Exact and numerical solutions of Equations (25).

$x$	Exact $y_1(x)$	TBF-6C:2P4D $y_1(x)$	Runge-Kutta $y_1(x)$	Exact $y_2(x)$	TBF-6C:2P4D $y_2(x)$	Runge-Kutta $y_2(x)$
0.00	0.000000	0.000000	0.000000	2.000000	2.000000	2.000000
1.00	1.209350	1.209350	1.209351	1.749653	1.749653	1.749653
2.00	1.179968	1.179968	1.179968	0.628486	0.628486	0.628486
3.00	0.290481	0.290481	0.290481	-0.799085	-0.799085	-0.799085
4.00	-0.683540	-0.683540	-0.683540	-1.392130	-1.392130	-1.392130
5.00	-0.925235	-0.925234	-0.925235	-0.668524	-0.668524	-0.668524
6.00	-0.264543	-0.264543	-0.264543	0.683234	0.683234	0.683234
7.00	0.663370	0.663370	0.663370	1.411801	1.411801	1.411801
8.00	0.992042	0.992042	0.992042	0.844194	0.844194	0.844193
9.00	0.413229	0.413229	0.413229	-0.498888	-0.498889	-0.498888
10.00	-0.543567	-0.543567	-0.543567	-1.383047	-1.383047	-1.383047
15.00	0.650292	0.650292	0.650292	-0.109400	-0.109400	-0.109400
20.00	0.912945	0.912946	0.912945	1.321027	1.321027	1.321027

**Table 7** compares PBF-6C:2P4D and the fourth-order Runge-Kutta method with the exact solution using the numerical computations carried out for  $\Delta x = 1/20$ . Both TBF-6C:2P4D and Runge-Kutta produce virtually identical results with the exact values.

### 5.3. PBF-6C:3P3D $g(x) = a_i x^5 + b_i x^4 + c_i x^3 + d_i x^2 + e_i x + h_i$

Again a fifth-order polynomial  $g(x) = a_i x^5 + b_i x^4 + c_i x^3 + d_i x^2 + e_i x + h_i$  is used but now the polynomial satisfies three points  $g(0) = f_i$ ,  $g(-\Delta x) = f_{i-1}$ ,  $g(-2\Delta x) = f_{i-2}$  and three first derivatives  $g'(0) = f'_i$ ,  $g'(-\Delta x) = f'_{i-1}$ ,  $g'(-2\Delta x) = f'_{i-2}$ . Applying the usual procedure for the prediction and correction stages gives

$$y_{i+1} = y_i + \frac{1}{240} \Delta x \left[ -949f_i + 608f_{i-1} + 581f_{i-2} + \Delta x(637f'_i + 1080f'_{i-1} + 173f'_{i-2}) \right] \quad (27a)$$

$$y_{i+1,c} = y_i + \frac{1}{240} \Delta x \left[ 101f_{i+1,p} + 128f_i + 11f_{i-1} + \Delta x(-13f'_{i+1,p} + 40f'_i + 3f'_{i-1}) \right] \quad (27b)$$

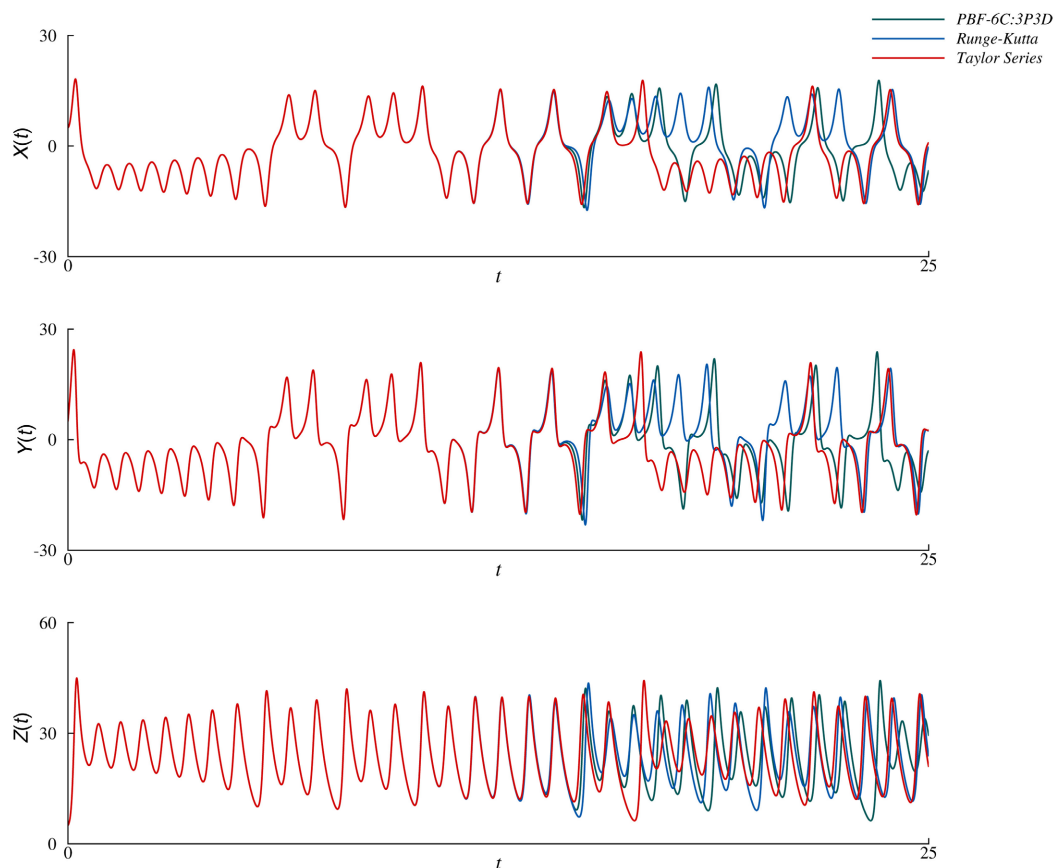
Both the predictor and corrector stage Equation (27a) and Equation (27b) are used for solving the Lorenz equations (Lorenz [12]):

$$\frac{dX}{dt} = -\sigma X + \sigma Y \quad (28a)$$

$$\frac{dY}{dt} = -XZ + rX - Y \quad (28b)$$

$$\frac{dZ}{dt} = XY - bZ \quad (28c)$$

where  $X$ ,  $Y$ , and  $Z$  are functions of time  $t$  only and the parameters are assigned the values  $\sigma = 10$ ,  $b = 8/3$ , and most importantly  $r = 28$ , a supercritical value that ensures a chaotic behavior for Equations (28a)-(28c). The initial values are set as  $x(0) = 5.0$ ,  $y(0) = 5.0$ ,  $z(0) = 5.0$  at the initial time  $t = 0$  and the time step is taken  $\Delta t = 1/1000$  for all the methods used in the simulation. **Figure 2** shows the variations of  $X$ ,  $Y$ , and  $Z$ , from top to bottom, within the time interval  $0 \leq t \leq 25$  as computed by using the PBF-6C:3P3D formulation (blue),



**Figure 2.** Numerical solutions of Lorenz equations,  $X$ ,  $Y$ ,  $Z$  from top to bottom, for  $r = 28$  using PBF-6C:3P3 (blue), Runge-Kutta (green), Taylor series (red) methods.

the Runge-Kutta method (green), and the Taylor series method developed to the sixth-order (red). The most remarkable feature of these solutions is that, despite starting with exactly the same initial values, the solutions diverge from one another after a definite time. This is typically observed for slightly different initial values; here, the chaotic behavior is triggered not by slightly differing initial values but by different numerical integration formulations. This aspect of the so-called chaotic equations should be explored in depth as it clearly implies that no solutions to such equations can be claimed as true. Concerning the PBF-6C:3P3D formulation, it is clear that this particular formulation performs as well as the other two well-known methods because initially all the solutions agree perfectly well and when they start to diverge they do so at nearly the same time.

## 6. Concluding Remarks

New single- and multi-step formulations are developed for the numerical integration of ordinary differential equations. The general framework described here makes use of the base functions or extrapolating functions satisfying different conditions for better curve-fitting purposes. In this manner, eight different formulations using trigonometric, exponential, and polynomial functions are developed. Numerical tests reveal quite satisfactory and even in some cases excellent results. Flexible selection of function types and imposed conditions to determine coefficients opens virtually unlimited possibilities for formulating new schemes particularly suitable for the problems considered.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] Butcher, J.C. (2008) Numerical Methods for Ordinary Differential Equations. Wiley. <https://doi.org/10.1002/9780470753767>
- [2] Bashforth, F. and Adams, J.C. (1883) An Attempt to Test the Theories of Capillary Action by Comparing the Theoretical and Measured Forms of Drops of Fluid. Cambridge University Press. <https://archive.org/details/attempttotestthe00bashuoft>
- [3] Runge, C. (1895) Ueber die numerische Auflösung von Differentialgleichungen. *Mathematische Annalen*, **46**, 167-178. <https://doi.org/10.1007/bf01446807>
- [4] Kutta, W. (1901) Beitrag zur näherungsweise Integration totaler Differentialgleichungen. *Zeitschrift für angewandte Mathematik und Physik*, **46**, 435-453.
- [5] Milne, W.E. (1926) Numerical Integration of Ordinary Differential Equations. *The American Mathematical Monthly*, **33**, 455-460. <https://doi.org/10.1080/00029890.1926.11986619>
- [6] Bulirsch, R. and Stoer, J. (1966) Numerical Treatment of Ordinary Differential Equations by Extrapolation Methods. *Numerische Mathematik*, **8**, 1-13. <https://doi.org/10.1007/bf02165234>
- [7] Fatimah, B.O., Senapon, W.A. and Adebowale, A.M. (2015) Solving Ordinary Dif-

- ferential Equations with Evolutionary Algorithms. *Open Journal of Optimization*, **4**, 69-73. <https://doi.org/10.4236/ojop.2015.43009>
- [8] Ogunrinde, R.B. and Olubunmi, J.O. (2019) Solution of Some Second Order Ordinary Differential Equations Using a Derived Algorithm. *Applied Mathematics*, **10**, 312-325. <https://doi.org/10.4236/am.2019.105022>
- [9] Signes-Pont, M.T., Boters-Pitarch, J., Cortés-Plana, J.J. and Mora-Mora, H. (2022) Approximating Ordinary Differential Equations by Means of the Chess Game Moves. *Journal of Applied Mathematics and Physics*, **10**, 3240-3263. <https://doi.org/10.4236/jamp.2022.1010215>
- [10] Abramowitz, M. and Stegun, I.A. (1972) Handbook of Mathematical Functions. Dover Publications.
- [11] Chapra, S.C. and Canale, R.P. (2005) Numerical Methods for Engineers. McGraw-Hill Education.
- [12] Lorenz, E.N. (1963) Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences*, **20**, 130-141. [https://doi.org/10.1175/1520-0469\(1963\)020<0130:dnf>2.0.co;2](https://doi.org/10.1175/1520-0469(1963)020<0130:dnf>2.0.co;2)