Pergamon

Computers Math. Applic. Vol. 28, No. 10–12, pp. 353–384, 1994
Copyright©1994 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0898-1221/94 $7.00 + 0.00

0898-1221(94)00201-0

Approximate Solution of Ordinary Differential Equations and Their Systems Through Discrete and Continuous Embedded Runge-Kutta Formulae and Upgrading of Their Order

D. SARAFYAN
Department of Mathematics, University of New Orleans
New Orleans, LA 70148, U.S.A.

Abstract—The eight main contributions of the author to the field of approximate solutions of ordinary differential equations described herein are all application-oriented, with the purposes of simplification and the increase in efficiency and effectiveness of the Runge-Kutta processes generated. They range from the determination of an initial trial step-size to be adopted to expedite the approximation process through embedded Runge-Kutta algorithms to a more recent procedure for upgrading the order of Runge-Kutta processes. These contributions encompass all classes of differential equations of all orders, such as explicit, implicit, single or systems, and their treatment by Runge-Kutta processes of scalar or vector type (with the related equivalence conditions), of discrete or continuous kind, including the computer derivations of nonlinear algebraic equations associated with the Runge-Kutta processes. Specifically, the author developed the first fifth order Runge-Kutta formulae with fourth order embedded and the first C^1 approximate solution through interpolation and Runge-Kutta formulae, which he improved by developing C^1 embeddings with Runge-Kutta formulae without the use of interpolative techniques.

Keywords—Ordinary differential equations, Numerical analysis, Runge-Kutta algorithms, Embedded formulae, Interpolants, Continuous approximate solutions, Upgrading the order of the approximate solutions, Error estimation, Step-size control.

1. INTRODUCTION

Since the derivation of Runge-Kutta formulae a century ago by their first originators, Runge [1], Henn [2], and Kutta [3], many researchers contributed in different ways to this popular approximation process for the solution of initial value problems involving ordinary differential equations. The present author will describe in this article solely his own contributions extending from 1960 to the present. All these contributions are application-oriented, and usually they were first presented at a professional meeting, then in institutional technical reports issued and widely distributed, and finally published in journals.

We shall be concerned with the initial value problem

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

where at present we shall assume the differential equation to be of scalar type, that is, composed of a single equation. Later, we shall show that our treatment can be extended to the vector cases, that is, to systems of differential equations.

It shall further be assumed that the function f in (1) is continuous and satisfies a Lipschitz condition in a closed circular neighborhood of the initial point (x_0, y_0) to assure us of the existence and uniqueness of the solution $y(x)$ of the initial value problem (1).

A Runge-Kutta algorithm of order r and involving s stages or substitutions may be defined as follows:

$$\tilde{y}_{r,s}(x_0 + h) = y_0 + \sum_{i=0}^{s-1} w_i \, k_i, \tag{2a}$$

$$k_0 = hf(x_0, y_0),$$

$$k_i = hf\left(x_0 + a_i\, h,\ y_0 + \sum_{j=0}^{i-1} b_{i,j}\, k_j\right), \qquad i = 1, \dots, s-1, \tag{2b}$$

$$a_i = \sum_{j=0}^{i-1} b_{i,j}. \tag{2c}$$

Whenever confusion does not arise, we may write $\tilde{y}_{r,s}(x_0 + h)$ simply as $y_r(x_0 + h)$ or y_r, the subscript r specifying only the order of the process.

We shall refer to $f(x, y)$ in (1) as the "directional function," to the w's in (2a) as "weights" and to the k's in (2b) as "incremental coefficients," and finally to the a's and b's as "constants of the algorithm" or simply as "constants."

The author's contributions are described consecutively in the following eight sections. At times when appropriate we shall not enter into details, but refer the interested readers to the specific literature and/or other pertinent information.

2. RECTILINEAR SOLUTIONS

In 1960, this author discovered an interesting property of Runge-Kutta processes [4] which can be announced in the following theorem.

THEOREM. *If the solution of the initial value problem* (1) *is* $y = y_0 + (x - x_0)\, f(x_0, y_0)$, *a straight line, then any Runge-Kutta algorithm, regardless of its order, will yield the discrete exact solution* $y(x_0 + h)$.

However, in actual practice in certain problems, higher order processes, involving lengthy calculations, on account of the inevitable round-off errors introduced, will not yield the exact value but only approximations to it, so that after repeated applications of the algorithm, the process, so as to say, would be "off the track." Consequently, since all algorithms theoretically give the exact value, naturally it is preferable to use only the first order Runge-Kutta algorithm $\tilde{y}_1(x_0 + h) = y_0 + k_0$, which is actually none other than the well known Euler's formula

$$\tilde{y}(x) = y_0 + (x - x_0)\, f(x_0, y_0), \tag{3}$$

the round-off errors originating from the application of which formulae are relatively insignificant, if not nil.

Still better, why resort to an approximation process if one can obtain with ease the exact solution? Thus, before attacking the initial value problem (1) by an approximation process, it is advisable from the onset to verify whether the function $y_0 + (x - x_0)\, f(x_0, y_0)$ satisfies identically the differential equation (1), that is, whether

$$f(x_0, y_0) \equiv f\left(x,\ y_0 + (x - x_0)\, f(x_0, y_0)\right). \tag{4}$$

In this manner, one can readily determine if equation (3) constitutes the solution of the considered initial value problem (1) and thus avoid unnecessary labor, waste of time, and even more importantly, erroneous results.

It should be pointed out that the recommended verification of the identity (4) does not require additional evaluations of the directional function f, since the only necessary evaluation, namely $f(x_0, y_0)$, is the main component of the first stage, $k_0 = h\, f(x_0, y_0)$, of any Runge-Kutta algorithm.

This affinity of Runge-Kutta algorithms to rectilinear solutions is so strong that even in the case where the IVP (1) has more than one solution, one of which integral curves in a straight line, the Runge-Kutta algorithms will yield the exact solution. To this effect, consider, for example, the IVP

$$\frac{dy}{dx} = (y-1)^{2/3}, \qquad y(0) = 1. \tag{5}$$

Definitely, $y = 1$ and $y = \frac{x^3}{27} + 1$ are solutions. Actually, the former is a minor while the latter is a major solution, and there are infinitely many other solutions or integral curves between them. Yet the use of any Runge-Kutta algorithm, irrespective of its order, will yield only the rectilinear solution $y = 1$.

Later on, when interest in stiff differential equations emerged, this strong affinity of Runge-Kutta processes to integral curves which are straight lines, provided us insight into the way this type of equations should be treated. Indeed, sections of the integral curve, the solution of an IVP involving stiff differential equations, almost look like a straight line. Thus, one should not tackle these equations with high order Runge-Kutta algorithms, which would generate substantial round-off errors, but rather with low order Runge-Kutta algorithms, such as the fifth order processes or perhaps still more advantageously with other low order predictor-corrector type processes.

3. DERIVATION OF ALGEBRAIC EQUATIONS ASSOCIATED WITH THE RUNGE-KUTTA PROCESSES

In an r^{th} order process, the power series expansions about the point (x_0, y_0) of the exact solution $y(x_0 + h)$ and the approximate solution $\tilde{y}_{r,s}(x_0 + h)$, as defined in (2a,b,c), must coincide up through their r^{th} degree term. This matching procedure leads to a nonlinear system of algebraic equations in the unknowns: w's, a's, and b's involved in the Runge-Kutta formula. Then, this system is solved. Each of these two distinct processes becomes more and more tedious and time consuming with the increase of the order r of the Runge-Kutta formulae.

Various methods have been established and used [5–7] for the purpose of making the derivation of these equations less laborious. Nevertheless, these known methods are still quite involved, and there is always everywhere present the strong possibility of human error which may be difficult to detect.

Our method is totally different from the others and considerably simplifies the derivation of the nonlinear algebraic equations. Moreover, the method is adaptable to computer implementations. It is worth mentioning that the heart of the computer program lies in the values of prime numbers and uses their property for identification purposes. Thus, for instance, $w_3\, a_1^3\, b_{2,1}^2$ may be represented by 600 with the assignment of

$$w_3 = 3, \; a_1 = 2, \quad \text{and} \quad b_{2,1} = 5. \tag{6}$$

Two technical reports, [8,9], were issued, followed by a journal publication [10]. Subsequently, Fehlberg adopted this method in his treatment of second order differential equations [11]. The interested readers are referred to these works [8–10].

4. EMBEDDED RUNGE-KUTTA PROCESSES

Milne, in his well known treatise [12, p. 74] commented about the Runge-Kutta method that "the process does not contain in itself any simple means for estimating the errors" and mentioned that although Bieberbach had found an expression providing an upper bound for the errors, however, that estimate depended on quantities that do not appear directly in computation and therefore required some additional separate calculations.

Milne's comments were inappropriate because, as it will be shown below, Runge-Kutta algorithms are endowed with an internal error estimation property which does not require any additional evaluations of the directional functions f; that is, it is obtained "at no cost." However, this property was undetected, let alone exploited.

This author developed several families of fifth order six-stage algorithms which in appearance were like the standard, familiar formulae of that order. However, these formulae were endowed with an internal property which was not apparent: that with the first four incremental coefficients of the fifth order six-stage formula, namely, with the set $\{k_0, k_1, k_2, k_3\}$, a fourth order four-stage algorithm is obtained. And interestingly, in absolute value, the difference of the two approximations generated, that is

$$|\tilde{y}_{5,6}(x_0+h) - \tilde{y}_{4,4}(x_0+h)| = e(x_0+h), \tag{7}$$

constituted a very good estimate for

$$|y(x_0+h) - \tilde{y}_{4,4}(x_0+h)|, \tag{8}$$

the absolute truncation error in the fourth order approximation $\tilde{y}_{4,4}(x_0+h)$ to the exact value $y(x_0+h)$. The reason for this conclusion, which is supported by numerous test results, is as follows.

Let $e_4(x_0+h)$ and $e_5(x_0+h)$ designate the absolute truncation errors associated with the 4th and 5th order approximations considered above, respectively. We thus can write:

$$\begin{aligned} e_4(x_0+h) &= |y(x_0+h) - \tilde{y}_{4,4}(x_0+h)| = O(h^5), \\ e_5(x_0+h) &= |y(x_0+h) - \tilde{y}_{5,6}(x_0+h)| = O(h^6), \end{aligned} \tag{9}$$

where h is the chosen step-size.

For sufficiently small h, $\tilde{y}_{5,6}(x_0+h)$ constitutes a very close approximation to $y(x_0+h)$, the exact value. Consequently, substituting in (9) the exact value $y(x_0+h)$ by the close approximation $\tilde{y}_{5,6}(x_0+h)$, we may write:

$$e_4(x_0+h) = |y(x_0+h) - \tilde{y}_{4,4}(x_0+h)| \approx |\tilde{y}_{5,6}(x_0+h) - \tilde{y}_{4,4}(x_0+h)| = e(x_0+h). \tag{10}$$

And if it thus has been determined that $\tilde{y}_{4,4}(x_0+h)$ constitutes an n-decimal figure approximation to $y(x_0+h)$, the exact value, then $\tilde{y}_{5,6}(x_0+h)$, which is more accurate than $\tilde{y}_{4,4}(x_0+h)$, must be as well an n-decimal figure approximation, if not even better. Accordingly, we may modify (10) to

$$e_5(x_0+h) = |y(x_0+h) - \tilde{y}_{5,6}(x_0+h)| < |\tilde{y}_{5,6}(x_0+h) - \tilde{y}_{4,4}(x_0+h)| = e(x_0+h), \tag{11}$$

and consider $|\tilde{y}_{5,6}(x_0+h) - \tilde{y}_{4,4}(x_0+h)|$ as a bound or a valuable conservative estimate for $|y(x_0+h) - \tilde{y}_{5,6}(x_0+h)|$, that is, the error in $\tilde{y}_{5,6}(x_0+h)$. These formulae were originally called "pseudo-iterative," but subsequently they have come to be known as "embedded" Runge-Kutta formulae.

Among this new category of fifth order algorithms which are valid for scalar as well as vector IVPs, the following appears to be one of the best:

$$\tilde{y}_{5,6}(x_0+h) = y_0 + \frac{1}{336}(14k_0 + 35k_3 + 162k_4 + 125k_5), \tag{12a}$$

$$\tilde{y}_{4,4}(x_0+h) = y_0 + \frac{1}{6}(k_0 + 4k_2 + k_3), \tag{12b}$$

where

$$
\begin{aligned}
k_0 &= hf(x_0, y_0),\\
k_1 &= hf\left(x_0 + \frac{1}{2}h,\, y_0 + \frac{1}{2}k_0\right),\\
k_2 &= hf\left(x_0 + \frac{1}{2}h,\, y_0 + \frac{1}{4}(k_0 + k_1)\right),\\
k_3 &= hf(x_0 + h,\, y_0 - k_1 + 2k_2),\\
k_4 &= hf\left(x_0 + \frac{2}{3}h,\, y_0 + \frac{1}{27}(7k_0 + 10k_1 + k_3)\right),\\
k_5 &= hf\left(x_0 + \frac{2}{10}h,\, y_0 + \frac{1}{625}(28k_0 - 125k_1 + 546k_2 + 54k_3 - 378k_4)\right).
\end{aligned}
\tag{12c}
$$

These new fifth order processes and their error estimation property were first presented at an AMS meeting in 1965 [13]. Further information and/or details about these processes were given in [14–16], which contributed to their broad exposure [17–19]. Interestingly, the very same fifth order six-stage embedded process (12) with its error estimation property was rediscovered and published in 1969 [20]. On the other hand, Fehlberg issued several technical reports followed by journal publications concerning these embedded Runge-Kutta processes with error estimation property, to which he referred as Runge-Kutta formulae "with step-size control" [21–23]. However, in these works [20–23], the lower order embedded process plays the role of the main algorithm and is used for advancing step by step, while the higher one is used for error estimation. The approximations thus generated are further "off the track" than is necessary. A criticism of this detrimental procedure affecting the overall approximation process can be found in [19].

Thus, in these two other fifth order processes mentioned [18–20], the lower fourth order process is considered as the main algorithm and used for advancing step by step while the higher, fifth order approximation is used for error estimation. Thus, these two main approximations are further "off the track" than is necessary.

5. CONDITIONS FOR THE EQUIVALENCE OF THE SYSTEM OF ALGEBRAIC EQUATIONS ASSOCIATED WITH SCALAR AND VECTOR INITIAL VALUE PROBLEMS

The methods of derivation of the nonlinear system of algebraic equations associated with the initial value problem (1) vary, depending on whether the initial value problem is relative to a single differential equation, the scalar case, or a system of differential equations, the vector case. As shown in the following table [23], the number of algebraic equations grows with r, the order of the process, and much more rapidly for the vector case than for the scalar case.

Order r	1	2	3	4	5	6	7	8
Scalar	1	2	4	8	16	31	59	110
Vector	1	2	4	8	17	37	85	200

Order r	9	10	11	12	13	14
Scalar	201	361	639	1114	1917	3259
Vector	486	1205	3047	7813	20299	53272

Since the number of algebraic equations are so excessively large for formulae of higher order, it is naturally advantageous to solve the scalar case, which not only has fewer algebraic equations, but requires fewer stages to supply enough constants or parameters. Thus, the question arises as to whether a solution of the system of algebraic equations corresponding to the scalar case will also satisfy the algebraic equations of the vector case for the same order. When Butcher [24] analyzed Huta's two sixth-order eight-stage formulae, which were derived by solving the scalar equations, he found it "remarkable that, in fact, Huta's processes are applicable to a set of simultaneous (differential) equations."

The author has shown that the imposition of certain sets of conditions reduces the nonlinear systems of algebraic equations corresponding to scalar and vector processes to two equivalent systems. Consequently, the solution of the equivalent scalar system provides Runge-Kutta algorithms which are valid for scalar and vector IVPs.

Consider, for instance, the following sets denoted by (A) and (B) to which we may refer as squaring and cubing conditions, respectively:

$$\text{(A)} \qquad \sum_{j=1}^{i-1} a_j\, b_{i,j} = \frac{1}{2}\, a_i^2,$$

$$\text{(B)} \qquad \sum_{j=1}^{i-1} a_j^2\, b_{i,j} = \frac{1}{3}\, a_i^3, \qquad i = 2, 3, \dots, s-1,$$

where s indicates the number of stages of the Runge-Kutta process used.

The squaring conditions (A) with $s = 6$ constitute the equivalency criterion for the fifth order six-stage process [23,25]. On the other hand, the squaring and cubing conditions (A) and (B), jointly with $s = 8$, constitute an equivalency criterion for the sixth order eight-stage processes.

It should be mentioned that there are other criteria assuring the equivalency of scalar and vector Runge-Kutta processes. For instance, in the case of sixth order eight-stage processes,

$$\text{(A)} \quad \text{and} \quad a_7 = 1,$$

constitute another equivalency criterion [23,26].

Researchers confronted with the difficult task of obtaining solutions to lengthy and complex systems of algebraic equations corresponding to high order Runge-Kutta processes have introduced, at numerous times, simplifying assumptions which we have determined always included equivalence conditions. Thus, for instance, Huta [6,7], in dealing with the system of 31 algebraic equations corresponding to the sixth order eight-stage processes, introduces the squaring conditions (A) and $a_7 = 1$, which as just mentioned constitute an equivalence criterion and renders his formula valid also for the vector case. But he additionally introduces the cubing conditions (B) which, jointly with the conditions (A) already introduced, constitute another equivalence criterion, which again assures that Huta's sixth order formulae developed for the scalar case are valid for the vector case as well, a highly desirable property of which Huta himself was not aware.

In the existing literature of that time, we found only one case where the constants involved in a claimed sixth order formula for the scalar case did not satisfy the corresponding system of algebraic equations of the vector case. This was Shanks' formula (17) of [27]. But it was shown in [23] that the claim was erroneous and the formula in question was not of sixth order.

Sarafyan developed in 1967–68 [28–30] three sixth order eight-stage algorithms which may be considered improved versions of Huta's two formulae [6,7]. These sixth order processes of Huta and Sarafyan have an internal property in common, which as we shall see later, will enable us to transform these discrete Runge-Kutta processes into continuous (C^1) processes.

We produce below, for later use, one of the sixth order formulae of Sarafyan. We produce also a Sarafyan seventh order ten-stage algorithm [31] which we believe, on account of numerous tests, to be the most effective of all the seventh order algorithms. Finally, we give a family of eight order 13-stage algorithms of Sarafyan developed in 1970 under a NASA grant.

A SIXTH ORDER FORMULA OF SARAFYAN.

$$\tilde{y}_{6,8}(x_0+h) = y_0 + \frac{1}{840}\left[41\,(k_0+k_7) + 216\,(k_2+k_6) + 27\,(k_3+k_5) + 272\,k_4\right], \tag{13a}$$

$$\tilde{y}_{4,4}\left(x_0+\frac{1}{3}h\right) = y_0 + \frac{1}{18}\,(k_0+4k_2+k_3), \tag{13b}$$

where

$$\begin{aligned}
k_0 &= hf(x_0,y_0),\\
k_1 &= hf\left(x_0+\frac{1}{9}h,\ y_0+\frac{1}{9}k_0\right),\\
k_2 &= hf\left(x_0+\frac{1}{6}h,\ y_0+\frac{1}{24}(k_0+3k_1)\right),\\
k_3 &= hf\left(x_0+\frac{1}{3}h,\ y_0+\frac{1}{6}(k_0-3k_1+4k_2)\right),\\
k_4 &= hf\left(x_0+\frac{1}{2}h,\ y_0+\frac{1}{8}(k_0+3k_3)\right),\\
k_5 &= hf\left(x_0+\frac{2}{3}h,\ y_0+\frac{1}{9}(17\,k_0-63\,k_1+51\,k_2+k_4)\right),\\
k_6 &= hf\left(x_0+\frac{5}{6}h,\ y_0+\frac{1}{24}(-22\,k_0+33\,k_1+30\,k_2-58\,k_3+34\,k_4+3k_5)\right),\\
k_7 &= hf\left(x_0+h,\ y_0+\frac{1}{82}(281\,k_0-243\,k_1-522\,k_2+876\,k_3-346\,k_4-36\,k_5+72\,k_6)\right).
\end{aligned} \tag{13c}$$

THE SEVENTH ORDER FORMULAE OF SARAFYAN.

$$\tilde{y}_{7,10}(x_0+h) = y_0 + \frac{1}{17280}\left[751\,(k_0+k_9) + 3577\,(k_3+k_8) + 1323\,(k_4+k_7) + 2989\,(k_5+k_6)\right],$$

where

$$\begin{aligned}
k_0 &= hf(x_0,y_0),\\
k_1 &= hf\left(x_0+\frac{4}{63}h,\ y_0+\frac{4}{63}k_0\right),\\
k_2 &= hf\left(x_0+\frac{2}{21}h,\ y_0+\frac{1}{42}(k_0+3k_1)\right),\\
k_3 &= hf\left(x_0+\frac{1}{7}h,\ y_0+\frac{1}{28}(k_0+3k_2)\right),\\
k_4 &= hf\left(x_0+\frac{2}{7}h,\ y_0+\frac{1}{7}(k_0-3k_2+4k_3)\right),\\
k_5 &= hf\left(x_0+\frac{3}{7}h,\ y_0+\frac{1}{28}(3k_0+9k_4)\right),\\
k_6 &= hf\left(x_0+\frac{4}{7}h,\ y_0+\frac{1}{1827}(-202\,k_0+1302\,k_2-954\,k_4+898\,k_5)\right),\\
k_7 &= hf\left(x_0+\frac{5}{7}h,\ y_0+\frac{10}{131544}(-1881\,k_0-783\,k_2+10352\,k_3-3414\,k_4+5122\,k_6)\right),\\
k_8 &= hf\Big(x_0+\frac{6}{7}h,\ y_0+\frac{1}{2222850}(683663\,k_0+430650\,k_2\\
&\qquad -2032615\,k_3+2208930\,k_4+385270\,k_5-740735\,k_6+970137\,k_7)\Big),
\end{aligned}$$

$$k_9 = hf\left(x_0 + h,\ y_0 + \frac{1}{19601100}\,(-12175421\,k_0 - 11236050\,k_2 + 62891430\,k_3 - 43488585\,k_4 - 9947140\,k_5 + 51099720\,k_6 - 30879954\,k_7 + 13337100\,k_8\right).$$

FAMILIES OF EIGHT ORDER FORMULAE OF SARAFYAN.

$$\tilde{y}_{8,13}(x_0 + h) = y_0 + \frac{1}{28350}\left[989\,(k_0 + k_{12}) + 5888\,(k_5 + k_{11}) - 928\,(k_6 + k_{10}) + 10496\,(k_7 + k_9) - 4540\,k_8\right],$$

where ($t \neq 0$ represents a free parameter):

$$k_0 = hf(x_0, y_0),$$
$$k_1 = hf(x_0 + th,\ y_0 + tk_0),$$
$$k_2 = hf\left(x_0 + \frac{1}{8}\,h,\ y_0 + \left(\frac{1}{8} - \frac{1}{128t}\right)k_0 + \frac{1}{128t}\,k_1\right),$$
$$k_3 = hf\left(x_0 + \frac{3}{16}\,h,\ y_0 + \frac{1}{64}\,(3k_0 + 9k_2)\right),$$
$$k_4 = hf\left(x_0 + \frac{3}{40}\,h,\ y_0 + \frac{3}{2000}\,(29\,k_0 + 33\,k_2 - 12\,k_3)\right),$$
$$k_5 = hf\left(x_0 + \frac{1}{8}\,h,\ y_0 + \frac{1}{1296}\,(33\,k_0 + 4k_3 + 125\,k_4)\right),$$
$$k_6 = hf\left(x_0 + \frac{1}{4}\,h,\ y_0 + \frac{1}{648}\,(6k_0 + 112\,k_3 + 125\,k_4 - 81\,k_5)\right),$$
$$k_7 = hf\left(x_0 + \frac{3}{8}\,h,\ y_0 + \frac{1}{35424}\,(-4305\,k_0 - 3608\,k_3 + 25625\,k_4 - 18819\,k_5 + 14391\,k_6)\right),$$
$$k_8 = hf\left(x_0 + \frac{1}{2}\,h,\ y_0 + \frac{1}{648}\,(30\,k_0 + 32\,k_3 - 125\,k_4 + 324\,k_5 - 162\,k_6 + 225\,k_7)\right),$$
$$k_9 = hf\left(x_0 + \frac{5}{8}\,h,\ y_0 + \frac{1}{1700352}\,(1313148\,k_0 + 2714528\,k_3 - 5806625\,k_4 + 5442012\,k_5 - 4098114\,k_6 + 1167885\,k_7 + 329886\,k_8)\right),$$
$$k_{10} = hf\left(x_0 + \frac{3}{4}\,h,\ y_0 + \frac{1}{50112}\,(413718\,k_0 + 1955296\,k_3 - 923875\,k_4 - 763398\,k_5 + 224046\,k_6 - 1778805\,k_7 + 1167156\,k_8 - 256554\,k_9)\right),$$
$$k_{11} = hf\left(x_0 + \frac{7}{8}\,h,\ y_0 + \frac{1}{953856}\,(-166224\,k_0 + 4734688\,k_3 + 4689125\,k_4 - 10023912\,k_5 + 4103298\,k_6 - 4238505\,k_7 + 1104678\,k_8 + 669060\,k_9 - 37584\,k_{10})\right),$$
$$k_{12} = hf\left(x_0 + h,\ y_0 + \frac{1}{80109}\,(236121\,k_0 - 675392\,k_3 - 1544500\,k_4 + 2189295\,k_5 - 160218\,k_6 - 473850\,k_7 + 849582\,k_8 - 400545\,k_9 + 59616\,k_{11}\right).$$

Recommended values for t are: $\frac{1}{128}$, $\frac{1}{16}$, $\frac{1}{8}$, and 1.

At this point, the following comment is appropriate. Fehlberg stated [32, (see Introduction, p. 3)] that he believed the optimum order for Runge-Kutta formulae was eight order.

However, his conclusion was based on comparative tests using only the one tenth order and two ninth order formulae known at that time (in fact, the correctness of two of the formulae cannot be verified, since they were not published). Definitely more research is needed in this direction.

6. A PRACTICAL RULE FOR THE DETERMINATION OF A TRIAL OR INITIAL "SUFFICIENTLY SMALL" STEP-SIZE

As we have seen, the error estimation property of embedded type Runge-Kutta processes has been established with the assumption that the chosen step-size h is "sufficiently small." In the next two sections, when discussing the generation of approximate continuous Runge-Kutta solutions to the initial value problem (1), again it will be assumed that the selected step-size h is sufficiently small. Then, naturally, the question arises as to how to select, as economically as possible, preferably "at no cost," such a step-size.

A tentative initial step-size h_t, which may be considered acceptably close to a "sufficiently small h" to provide a useful estimate for the errors is given by the following practical rule established by Sarafyan [33]. Its effectiveness has been confirmed by numerous tests.

RULE. *Given the initial value problem*

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

then h_t given by

$$h_t = \frac{1}{2}\left|\frac{y_0}{f(x_0,y_0)}\right|, \qquad y_0 \neq 0, \quad f(x_0,y_0) \neq 0,$$

is a trial, sufficiently small step-size.

This simple and easily applicable rule, which eliminates the arbitrary and hazardous choice of a sufficiently small step-size h, thus preventing waste of time and labor, is obtained "at no cost" since it makes use of y_0 which is given and of $f(x_0,y_0)$ which is a component of $k_0 = hf(x_0,y_0)$, the first incremental coefficient to be used in any Runge-Kutta algorithm, regardless of its order. Naturally, whenever there does not arise any confusion, h_t may simply be written h.

Assume now that the hypotheses of the existence and uniqueness theorems for the solution $y(x)$ of the initial value problem (1) are satisfied. Then, the exact solution $y(x_0 + h)$ and a Runge-Kutta approximation to it obtained through the use of the trial step-size h, both fall in the so-called "butterfly region." In particular, the singular points, if any, are avoided.

To put into evidence the effectiveness of this *rule* for obtaining a *safe* and *useful initial step-size* from the very start of any Runge-Kutta process, or, as a matter of fact, with *any other approximation process*, we consider, for example, two innocent looking but actually vicious initial value problems, specifically:

$$\text{(i)} \quad \frac{dy}{dx} = 10y^2, \qquad y(0) = 1, \qquad \left(\text{Solution: } y = \frac{1}{1-10x}\right),$$

$$\text{(ii)} \quad \frac{dy}{dx} = -y - \frac{2x}{y}, \qquad y(0) = 1, \qquad \left(\text{Solution: } y = \sqrt{1-2x}\right).$$

We observe that, for the first problem, the solution is not defined at $x = h = 0.1$, and whenever $x = h > 0.1$ a Runge-Kutta algorithm will yield a positive valued approximation, $\tilde{y}_{r,s}(x_0+h) > 0$, while the exact solution $y(x_0 + h) < 0$. The rule yields $h = 0.05$, a safe trial step-size to initiate the approximation process.

In the case of the second problem, the exact solution of which is $y = \sqrt{1-2x}$, it is not defined if $x > 1/2$. The rule provides $x = h = 1/2$, a step-size for which the solution is defined, to initiate the approximation process.

This rule for the determination of a trial step-size, together with a third order embedded Runge-Kutta process, was adopted in the SR 60 Texas Instruments calculators in 1976 [33, p. 79]. The reason for the choice of a third order Runge-Kutta process although higher order algorithms were then available was the memory limitations of these calculators.

In the case of *vector initial value problems*, such as when dealing with a system of two differential equations, the modified rule for trial step-size determination may be announced as follows.

Rule. *Given the initial value problem*

$$\begin{aligned}\frac{dy}{dx} &= f(x,y,z),\\ \frac{dz}{dx} &= g(x,y,z), \qquad y(x_0)=y_0, \quad z(x_0)=z_0.\end{aligned}$$

Letting

$$f(x_0,y_0,z_0)=f_0, \quad g(x_0,y_0,z_0)=g_0,$$

an initial trial step-size h_t is given by

$$h_t=\frac{1}{2}\min\left\{\left|\frac{y_0}{f_0}\right|,\left|\frac{z_0}{g_0}\right|\right\}, \qquad y_0\neq 0, \quad z_0\neq 0, \quad f_0\neq 0, \quad g_0\neq 0.$$

However, before beginning the determination of an approximate solution for the IVP and, consequently, before the application of the rule for obtaining a trial step-size, as recommended in the Section 2, it is advantageous to verify whether the solution of the considered IVP is a straight line. In other words, when dealing with the scalar case, we must verify whether the identity $f_0 \equiv f\,(x,\, y_0+(x-x_0)\,f_0)$ holds, while in the vector case (restricting to a system composed of two equations for simplification), we must verify whether the identities $f_0 \equiv f\,(x,\, y_0+(x-x_0)\,f_0,\, z_0+(x-x_0)\,g_0)$ and $g_0 \equiv g\,(x,\, y_0+(x-x_0)\,f_0,\, z_0+(x-x_0)\,g_0)$ hold.

7. THE GENERATION OF APPROXIMATE CONTINUOUS (C^1) SOLUTIONS TO ORDINARY DIFFERENTIAL EQUATIONS AND THEIR SYSTEMS

Originally (1965), this author had embedded in three families of fifth order six-stage Runge-Kutta processes, $\tilde{y}_{5,6}(x_0+h)$, a fourth order four-stage process, $\tilde{y}_{4,4}(x_0+h)$. Soon thereafter (1966), it was realized that in a main, fifth order six-stage algorithm $\tilde{y}_{5,6}(x_0+h)$, a fourth order four-stage algorithm $\tilde{y}_{4,4}(x_0+ch)$, $0<c<1$, could be embedded. In other words, the approximations generated by the embedded fourth order algorithm was not at the end of the step for $x=x_0+h$ as before, but at any interior point of $[x_0, x_0+h]$, the interval of application of the main, fifth order algorithm and, in particular, at $x=x_0+(1/2)\,h$, the midpoint of the interval.

Thus, a family of fifth order six-stage processes $\tilde{y}_{5,6}(x_0+h)$ in which were embedded fourth order four-stage algorithms $\tilde{y}_{4,4}(x_0+\frac{1}{2}\,h)$ had been developed [34]. The following pair of formulae appeared to be one of the simplest in this family:

$$\tilde{y}_{5,6}(x_0+h)=y_0+\frac{1}{90}\left[7\,(k_0+k_5)+32\,(k_2+k_4)+12\,k_3\right], \tag{14a}$$

$$\tilde{y}_{4,4}\left(x_0+\frac{1}{2}\,h\right)=y_0+\frac{1}{12}\,(k_0+4k_2+k_3), \tag{14b}$$

with

$$\begin{aligned}
k_0 &= hf(x_0, y_0),\\
k_1 &= hf\left(x_0 + \frac{1}{4}\,h,\ y_0 + \frac{1}{4}\,k_0\right),\\
k_2 &= hf\left(x_0 + \frac{1}{4}\,h,\ y_0 + \frac{1}{8}\,(k_0 + k_1)\right),\\
k_3 &= hf\left(x_0 + \frac{1}{2}\,h,\ y_0 - \frac{1}{2}\,k_1 + k_2\right),\\
k_4 &= hf\left(x_0 + \frac{3}{4}\,h,\ y_0 + \frac{3}{16}\,(k_0 + 3k_3)\right),\\
k_5 &= hf\left(x_0 + h,\ y_0 + \frac{1}{7}\,(-3k_0 + 2k_1 + 12\,k_2 - 12\,k_3 + 8k_4)\right).
\end{aligned} \tag{14c}$$

The substitution of h by $2h$ in this process transforms these two formulae into the following form:

$$\tilde{y}_{5,6}(x_0 + 2h) = y_0 + \frac{1}{45}\,[7\,(k_0 + k_5) + 32\,(k_2 + k_4) + 12\,k_3], \tag{15a}$$

$$\tilde{y}_{4,4}(x_0 + h) = y_0 + \frac{1}{6}\,(k_0 + 4k_2 + k_3), \tag{15b}$$

with

$$\begin{aligned}
k_0 &= hf(x_0, y_0),\\
k_1 &= hf\left(x_0 + \frac{1}{2}\,h,\ y_0 + \frac{1}{2}\,k_0\right),\\
k_2 &= hf\left(x_0 + \frac{1}{2}\,h,\ y_0 + \frac{1}{4}\,(k_0 + k_1)\right),\\
k_3 &= hf(x_0 + h,\ y_0 - k_1 + 2k_2),\\
k_4 &= hf\left(x_0 + \frac{3}{2}\,h,\ y_0 + \frac{3}{8}\,(k_0 + 3k_3)\right),\\
k_5 &= hf\left(x_0 + 2h,\ y_0 + \frac{2}{7}\,(-3k_0 + 2k_1 + 12\,k_2 - 12\,k_3 + 8k_4)\right).
\end{aligned} \tag{15c}$$

These new formulae were referred to as having composite or two-step form. They have been highly recommended for the users. For instance, Lapidus and Seinfeld comment in [19, p. 73]:

> The importance of this feature cannot be overstressed since it means that starting from x_0 it is possible to calculate (an approximation to $y(x_0 + h)$ via a fourth-order formula and (an approximation to $y(x_0 + 2h)$) via a fifth-order formula, requiring a total of six evaluations (of the function $f(x, y)$). On a per-step basis only three stages are required as compared to four stages with any other fourth-order formula. Thus this procedure is the most economical that we have encountered which yields at least fourth-order results... Obviously this entire formulation is highly recommended for the user.

Subsequently, these authors use this composite fifth order formula to solve two parabolic partial differential equations (through the Method of Lines), one of linear and the other of nonlinear type, and they find the composite formula more stable than the other two methods [19, pp. 230–233].

Furthermore, quite recently, Abraham refers to the above composite process as "one of the most efficient general purpose methods of Runge-Kutta 4/5 type" and adopts it in his modular software package [31]. Yet this composite formula will be further improved in the following section.

It must be mentioned that besides the fifth order formula (14) or (15), we have discovered that our three sixth order eight-stage formulae, one of which namely formula (13), already exhibited, also provides a fourth order approximation; however, not at $x = x_0 + (1/2)\,h$, as in the preceding case but now at the interior point $x = x_0 + (1/3)\,h$. Specifically, the main sixth order eight-stage algorithm and the embedded fourth order interior algorithms are:

$$\tilde{y}_{6,8}(x_0 + h) = y_0 + \frac{1}{840}\left[41\,(k_0 + k_7) + 216\,(k_2 + k_6) + 27\,(k_3 + k_5) + 272\,k_4\right],$$
$$\tilde{y}_{4,4}\left(x_0 + \frac{1}{3}\,h\right) = y_0 + \frac{1}{18}\,(k_0 + 4k_2 + k_3),$$

which have been exhibited previously as formulae (13a) and (13b), respectively, while the corresponding set of incremental coefficients $\{k_0, \dots, k_7\}$ was defined in (13c).

We have exploited the existence of *interior points* for various purposes, specifically:

(a) for the determination of a rough estimate for truncation errors and an acceptable step-size, which methods are outmoded by now and thus will not be discussed further; for curiosity one may see, for instance, [19, pp. 71, 72, (2.8.9)–(2.8.11)];
(b) to show the efficiency and effectiveness of Runge-Kutta processes of higher order (≥ 5); see, for instance, [19, p. 73];
(c) and, most importantly, for the generation of approximate continuous (C^1) solutions of the IVP (1), which will be subject of the present discussion.

We acknowledge from the onset of our discussion that the *interior points* as specified above have been, so as to say, the stepping stone for initiating the study of approximate continuous solution for IVPs through interpolative processes, whose solutions will be denoted by $H(x)$.

Since the generation of $H(x)$ is the same for either algorithms (13) or (14), irrespective of their order, we shall choose to work with the higher order process, specifically the sixth order eight-stage algorithm (13a,b,c), which yields more accurate approximations at the nodal or terminal point $x = x_0 + h$.

In view of the fact that the known interior point is of fourth order, $(x_0 + (1/3)\,h,\ \tilde{y}_{4,4}(x_0 + (1/3)\,h)$, that incited us to consider a fourth order interpolant, that is, a fourth degree polynomial

$$H(x) = a_0 + \sum_{i=1}^{4} (x - x_0)^i\, a_i, \tag{16}$$

where the five coefficients a_i's are unknown.

We need five conditions to determine these five unknowns. Through the use of the initial, interior and terminal points, (x_0, y_0), $(x_0 + (1/3)\,h,\ \tilde{y}_{4,4}\,(x_0 + (1/3)h))$, and $(x_0 + h,\ \tilde{y}_{6,8}(x_0 + h))$, we readily obtain three conditions, namely:

$$H(x_0) = y_0, \quad H\left(x_0 + \frac{1}{3}\,h\right) = \tilde{y}_{4,4}\left(x_0 + \frac{1}{3}\,h\right), \quad H(x_0 + h) = \tilde{y}_{6,8}(x_0 + h). \tag{17a}$$

Our goal is the determination of these unknowns with the use of data already available, that is, without the necessity of additional evaluations of the function f, briefly said at "no cost." We thus observe that the first stage of our algorithm is $k_0 = hf(x_0, y_0)$ and $f(x_0, y_0) = y'(x_0)$. We thus impose the condition corresponding to a very desirable property for $H(x)$, that is

$$H'(x_0) = f(x_0, y_0) = f_0. \tag{17b}$$

We further observe that in the next application of our sixth order algorithm, its first stage would be $(k_0) = hf\,(x_0 + h,\ \tilde{y}_{6,8}(x_0 + h))$. Therefore, $f\,(x_0 + h, \tilde{y}_{6,8}(x_0 + h))$ would be available in the next application of the formula; however, we make early use of it so as to say by "borrowing" it and imposing the condition corresponding to another desirable property for $H(x)$, that is

$$H'(x_0 + h) = f\,(x_0 + h,\ \tilde{y}_{6,8}(x_0 + h)) = f_1. \tag{17c}$$

We now have the necessary five conditions for the determination of the five unknown a_i's. On noting that

$$H'(x) = \sum_{i=1}^{4} i\,(x - x_0)^{i-1}\, a_i, \tag{18}$$

with a few simple operations we determine:

$$\begin{aligned}
a_0 &= y_0, \qquad a_1 = f_0, \\
a_2 &= \frac{1}{4h^2}\left[9\,(y_4 - y_6) + 72\,(y_4 - y_0) - 20\,hf_0 + 2hf_1\right], \\
a_3 &= \frac{1}{2h^3}\left[17\,(y_4 - y_6) + 64\,(y_4 - y_0) - 14\,hf_0 + 4hf_1\right], \\
a_4 &= \frac{3}{4h^4}\left[7\,(y_4 - y_6) + 20\,(y_4 - y_0) - 4hf_0 + 2hf_1\right],
\end{aligned} \tag{19}$$

where, for convenience, $\tilde{y}_{4,4}\,(x_0 + (1/3)\,h)$ and $\tilde{y}_{6,8}(x_0 + h)$ are written simply y_4 and y_6.

The above procedure for the determination of $H(x)$ may be referred to as semi-Hermitian interpolation. Had we needed additionally the condition for the derivative: $H'\,(x_0 + (1/3)\,h) = f\,(x_0 + (1/3)\,h,\, \tilde{y}_{4,4}(x_0 + (1/3)\,h))$ the procedure used would correspond to a Hermitian interpolation.

Letting $x = x_0 + c\,h$, where h is fixed and *preferably* $x \in [0,1]$, formulae (16) and (18) may be written

$$H(x_0 + ch) = y_0 + (ch)\,f_0 + (ch)^2\,a_2 + (ch)^3\,a_3 + (ch)^4\,a_4, \tag{20}$$
$$H'(x_0 + ch) = f_0 + 2\,(ch)\,a_2 + 3\,(ch)^2\,a_3 + 4\,(ch)^3\,a_4, \tag{21}$$

where a_2, a_3, and a_4 are defined as above. In general, formula (20) provides excellent approximations to $y(x_0 + ch)$, while formula (21) provides satisfactory approximations to $y'(x_0 + ch)$.

The distinctive features of our approximation process are the "interior point" and the "borrowing" procedure which consisted in making early use of $f\,(x_0 + h, \tilde{y}_{6,8}(x_0 + h)) = f_1$ available in the next application of the algorithm.

They contributed each to the increase of the degree of the interpolating polynomial $H(x)$ by one unit, that is in total of two units, thus increasing considerably the accuracy of the generated approximate continuous solution. Additionally, the use of the borrowed first stage of the process in its next application rendered the derivative of the interpolating polynomial continuous at the mesh points also (such as the point $x = x_0 + h$) in the repeated applications of the approximation process, in other words, providing approximate C^1 solutions to the IVP (1).

The generation of approximate continuous solutions by Runge-Kutta processes and interpolative methods can be traced to composite or multistep formulae discovered by this author in 1966 (see [34]). Further information for their development is given in [35,37] and more importantly in [38], a journal article about the sixth order case, which process has just been described above.

This article was reviewed in 1974 [39]. However, in this review one cannot find any mention of the generation of approximate continuous solutions to IVP (1), by interpolating polynomials or interpolants $H(x)$, and original undertaking which was the main purpose of the article. Yet, the reviewer needlessly interjects and praises Fehlberg's formulae, regardless of the fact that their "interior points" were then unknown, and thus, they were defective to be used in our interpolating procedure. Fortunately, certain researchers who were not acquainted with this destructive review read the author's article and have implemented in their research [40,41] the formulae described herein.

In 1983, at last, Horn determined [42] an "interior point" for Fehlberg's fifth order six-stage algorithm. Then the reviewer with three coauthors adopted the procedure described by us in [38] and generated a semi-Hermitian interpolant $H(x)$, which they described by $u_0(x)$ and referred to as "quartic interpolant." Their work was first issued as a technical report [43] (1985) and published in [44] (1986) without any mention of this author's original article published fourteen years earlier. They also upgraded $u_0(x)$ to a "quintic interpolant" $u_1(x)$, but only at the cost of two additional functional evaluations, and with no improvement of the approximation at the end of the interval, $x = x_0 + h$.

Further, these authors, by applying the same procedures, have also constructed interpolants with the use of fourth and sixth order Runge-Kutta algorithms. But these were not Fehlberg's formulae, because contrary to the reviewer's praise and promotion of the "very efficient class of Runge-Kutta formulae up to 8th order" of Fehlberg which "are easily implemented" they could not be used for the simple reason that their "interior points" were unknown. In fact, with the exception of the fifth order formula RK4(5), the "interior points" of all other Fehlberg formulae were unknown then, and they still remain unknown as of now, and thus are unfit in efficient interpolating procedures.

In 1983, at the time of Horn's determination of the "interior point" of Fehlberg's fifth order formula, which rekindled interest in the generation of interpolants through the use of these points as originally described by us in the 1972 article, we came to realize the generated interpolation processes were none other than continuous Runge-Kutta processes (involving variable weights $w_i(c)$, $c \in [0,1]$). We saw that it was far more advantageous to derive these continuous Runge-Kutta algorithms directly from the corresponding algebraic equations rather than through *restrictive* interpolative processes; this procedure will be undertaken in the next section.

8. THE GENERATION OF CONTINUOUS RUNGE-KUTTA PROCESSES

Consider again the initial value problem (1). For an approximate (discrete) solution of this problem, we shall use a Runge-Kutta algorithm. For the sake of convenience and simplicity assume this algorithm to be of fifth order involving six stages (the minimum number required), which may be written:

$$\tilde{y}_{5,6}(x_0 + h) = y_0 + \sum_{i=0}^{5} w_i\, k_i, \tag{22a}$$

with

$$\begin{aligned} k_0 &= hf(x_0, y_0), \\ k_i &= hf\left(x_0 + a_i\, h,\ y_0 + \sum_{j=0}^{i-1} b_{i,j}\, k_j\right), \qquad i = 1, \dots, 5, \\ a_i &= \sum_{j=0}^{i-1} b_{i,j}, \end{aligned} \tag{22b}$$

where h is the chosen step-size.

It will be assumed that the directional function f is sufficiently differentiable, more precisely that $f \in C^5$ in a neighborhood of the initial point (x_0, y_0). Then, the solution $y(x) \in C^6$.

To determine the appropriate values of the constants, w's, a's, and b's, the power series of $y(x_0 + h)$ and $\tilde{y}_{5,6}(x_0 + h)$ are matched up through their fifth degree terms, which process yields a system of 16 or 17 algebraic equations for the scalar and vector cases, respectively, involving 21 unknowns (the constants or parameters), and then the system is solved.

It will be advantageous to introduce the four squaring and four cubing equivalency conditions as defined in the Section 5; that is

$$\text{(A)} \qquad A_{i-1} = \sum_{j=1}^{i-1} a_j\, b_{i,j} = \frac{1}{2}\, a_i^2, \qquad i = 2, \dots, 5, \tag{23a}$$

$$\text{(B)} \qquad B_{i-1} = \sum_{j=1}^{i-1} a_j^2\, b_{i,j} = \frac{1}{3}\, a_i^3, \qquad i = 2, \dots, 5. \tag{23b}$$

Then the algebraic equations associated with the considered fifth order processes for scalar and vector cases, reduce to an equivalent system of nine algebraic equations. Furthermore, they render $w_1 = 0$. It follows that in total, including the eight equivalency conditions, we have $9 + 8 = 17$ equations involving $(21 - 1 =)\, 20$ unknown constants or parameters. We thus have three free parameters at our disposal.

Using these freedoms, we have thus discovered families of such fifth order algorithms. In particular, we obtained three different fifth order six-stage Runge-Kutta algorithms specified as Methods 1, 2, and 3. These algorithms, which will be soon exhibited, appeared from the standpoint of efficiency and effectiveness to be, if not superior, at least competitive with the best of known fifth order six-stage algorithms, $\tilde{y}_{5,6}(x_0 + h)$.

Motivated by the continuous processes developed through interpolation, we are interested in embedding in a (discrete) fifth order six-stage algorithm $\tilde{y}_{5,6}(x_0 + h)$, a continuous Runge-Kutta process which provides fourth order approximations at the interior points $x = x_0 + ch$, $c \in (0, 1)$, to $y(x_0 + ch)$, in such a way that the approximation is upgraded to the fifth order approximation for $c = 1$, that is at the end of the step, $x = x_0 + h$. To this effect, we have considered fourth order six-stage algorithms defined as follows:

$$\tilde{y}_{4,6}(x_0 + ch) = y_0 + \sum_{i=0}^{5} w_i(c)\, k_i, \qquad w_1(c) = 0, \tag{24}$$

where the weights are not constants any more, but functions of the variable c, and the k_i's are the incremental coefficients associated with the chosen fifth order six-stage algorithm $\tilde{y}_{5,6}(x_0 + h)$.

Then $y(x_0 + ch)$ and $\tilde{y}_{4,6}(x_0 + ch)$ were expanded in powers of h up to their fourth degree term and, as usual, the matching procedure of terms of like powers yielded the system of five algebraic equations relating to the considered fourth order continuous case, which system is displayed in [45, p. 146]. The solution of this continuous system for each of the preferred three fifth order six-stage algorithms (Methods 1, 2, and 3) provided us the respective three embedded continuous fourth order Runge-Kutta processes [46, pp. 78–81].

Actually, this system of algebraic equations permitted us to generate other continuous processes of orders $n = 1,\ 2,\ 3$, which could be used for a variety of purposes. All these processes, as part of their respective Methods 1, 2, and 3, are also exhibited in the precited article [46]. We shall treat herein only the improved versions of these three methods, which will be achieved at "no cost" thanks to the by now familiar procedure of borrowing the first stage of the next application of the algorithm $\tilde{y}_{5,6}(x_0 + h)$.

Indeed, it should be mentioned that since the said system consisted of five linear equations in five unknowns, specifically the variable weights: $w_0(c)$, $w_i(c)$, $i = 2, \dots, 5$, its solution was unique, thus resulting in the generation of a single algorithm $\tilde{y}_{4,6}(x_0 + ch)$ as defined in (24). The "borrowing" procedure will enable us to add a seventh coefficient $k_6 = hf\,(x_0 + h,\ \tilde{y}_{5,6}(x_0 + h))$ to the original set of six incremental coefficient $\{k_0, \dots, k_5\}$. With the addition of this seventh stage or incremental coefficient k_6, we have now six unknowns, namely, the variable weights: $w_0(c)$, $w_i(c)$, $i = 2, \dots, 6$, in a system composed of five linear algebraic equations so that one of w's, in particular $w_6(c)$, could be chosen as a free parameter (provided that, as we shall see, certain conditions were met).

Consequently, with this expanded set of seven coefficients $\{k_0, \dots, k_5, k_6\}$, we were able to develop a family of fourth order seven-stage algorithms:

$$\tilde{y}_{4,7}(x_0 + ch) = y_0 + \sum_{i=0}^{6} w_i(c)k_i, \qquad w_1(c) = 0, \tag{25}$$

instead of the single fourth order six-stage algorithm $\tilde{y}_{4,6}(x_0 + ch)$, with the objective of the determination of the best of such processes.

It must be observed that, since at $x = x_0 + h$, that is for $c = 1$, we must have

$$\tilde{y}_{4,7}(x_0 + h) = \tilde{y}_{5,6}(x_0 + h) \quad \text{and} \quad (\tilde{y}_{4,7}(x_0 + ch))'_{c=1} = f(x_0 + h,\ \tilde{y}_{5,6}(x_0 + h)) = \frac{1}{h} k_6; \tag{26}$$

it follows that w_6, as a free parameter, must be so chosen as to have $w_6(1) = 0$ and $\frac{d}{dc} w_6(1) = 1$. These conditions assure us that the obtained approximate continuous solution $\tilde{y}_{4,7}(x_0 + ch)$, like $H(x)$ of the interpolative case, is C^1.

With each of the three Methods 1, 2, and 3, we shall present two fourth order seven-stage continuous algorithms $\tilde{y}_{4,7}(x_0 + ch)$, which are upgraded for $c = 1$ to their respective fifth order parent algorithm $\tilde{y}_{5,6}(x_0 + h)$ to which we may refer as the main discrete algorithm. Additionally, we shall present another embedded algorithm for the purpose of estimation of errors, and also other low order algorithms as substitute for the continuous algorithm $\tilde{y}_{4,7}(x_0 + ch)$ if the use of such low order algorithms is found advantageous. As later it will be seen, these low order algorithms, jointly with $\tilde{y}_{4,7}(x_0 + ch)$ may be also used for an assessment of the trend of the approximation process.

These fourth order seven-stage algorithms can be expressed in terms of the variable weights $w_i(c)$, $i = 0, 2, \dots, 6$, or equivalently and more conveniently, in polynomial form, in powers of c. We shall use both forms only in the presentations of the main algorithms relative to Method 1; for Methods 2 and 3, the presentations of the continuous algorithms shall be made only in polynomial form. It should also be mentioned that all these formulae, of maximum order five are valid for scalar as well as vector initial value problems (1).

Method 1

Formulae of Method 1 presented below range, for the discrete case, from the first order to the (main) fifth order six-stage formula $\tilde{y}_{5,6}(x_0 + h)$, and for the continuous case, from a first order to two different (main) fourth order seven-stage formulae: $\tilde{y}_{4,7}(x_0 + ch)$. They are all defined with the set or subsets of the following seven incremental coefficients $\{k_0, \dots, k_5, k_6\}$, the last coefficient of which is a "borrowed" coefficient or stage from the next application of the main algorithm used.

$$\begin{aligned}
k_0 &= hf(x_0, y_0), \\
k_1 &= hf\left(x_0 + \frac{1}{6}h,\ y_0 + \frac{1}{6}k_0\right), \\
k_2 &= hf\left(x_0 + \frac{1}{4}h,\ y_0 + \frac{1}{16}(k_0 + 3k_1)\right), \\
k_3 &= hf\left(x_0 + \frac{1}{2}h,\ y_0 + \frac{1}{4}(k_0 - 3k_1 + 4k_2)\right), \\
k_4 &= hf\left(x_0 + \frac{3}{4}h,\ y_0 + \frac{1}{16}(3k_0 + 9k_3)\right), \\
k_5 &= hf\left(x_0 + h,\ y_0 + \frac{1}{7}(-4k_0 + 3k_1 + 12k_2 - 12k_3 + 8k_4)\right), \\
k_6 &= hf(x_0 + h,\ \tilde{y}_{5,6}(x_0 + h)), \\
\tilde{y}_{5,6}(x_0 + h) &= y_0 + \frac{1}{90}(7k_0 + 32k_2 + 12k_3 + 32k_4 + 7k_5).
\end{aligned} \tag{27}$$

THE FIRST MAIN CONTINUOUS FORMULA. The main fifth order (discrete) formula, $\tilde{y}_{5,6}(x_0+h)$, has just been defined in (27). Our first main fourth order seven-stage continuous formula in the variable weight form may be written, with $x = x_0 + ch$ or $c = (x - x_0)/h$, as follows:

$$\tilde{y}_{4,7}(x_0+ch) = y_0 + \sum_{i=0}^{6} [w_i(c)]\, k_i, \tag{28a}$$

where

$$\begin{aligned}
w_0(c) &= \frac{c}{90}(90 - 375\,c + 700\,c^2 - 600\,c^3 + 192\,c^4),\\
w_1(c) &= 0,\\
w_2(c) &= \frac{8}{45}\,c^2\,(45 - 130\,c + 135\,c^2 - 48\,c^3),\\
w_3(c) &= \frac{2}{15}\,c^2\,(-45 + 190\,c - 240\,c^2 + 96\,c^3),\\
w_4(c) &= \frac{8}{45}\,c^2\,(15 - 70\,c + 105\,c^2 - 48\,c^3),\\
w_5(c) &= \frac{7}{90}\,c^2\,(-180 + 700\,c - 855\,c^2 + 336\,c^3),\\
w_6(c) &= \frac{c^2}{2}\,(27 - 104\,c + 125\,c^2 - 48\,c^3).
\end{aligned} \tag{28b}$$

The algorithm (28a,b) in equivalent polynomial form in powers of $c = (x - x_0)/h$ may be written:

$$\tilde{y}_{4,7}(x_0+ch) = \tilde{y}_\Omega(x_0+ch) = y_0 + \sum_{i=1}^{5} \Omega_i\, c^i, \tag{29a}$$

where

$$\begin{aligned}
\Omega_1 &= k_0,\\
\Omega_2 &= \frac{1}{6}(-25\,k_0 + 48\,k_2 - 36\,k_3 + 16\,k_4 - 84\,k_5 + 81\,k_6),\\
\Omega_3 &= \frac{1}{9}(70\,k_0 - 208\,k_2 + 228\,k_3 - 112\,k_4 + 490\,k_5 - 468\,k_6),\\
\Omega_4 &= \frac{1}{6}(-40\,k_0 + 144\,k_2 - 192\,k_3 + 112\,k_4 - 399\,k_5 + 375\,k_6),\\
\Omega_5 &= \frac{8}{15}(4k_0 - 16\,k_2 + 24\,k_3 - 16\,k_4 + 49\,k_5 - 45\,k_6).
\end{aligned} \tag{29b}$$

The derivative with respect to x is obtained by:

$$(\tilde{y}_\Omega(x_0+ch))' = \frac{1}{h}\sum_{i=1}^{5} i\,\Omega_i\, c^{i-1}. \tag{29c}$$

As expected, we have

$$(\tilde{y}_\Omega(x_0+ch))_{c=1} = \tilde{y}_{5,6}(x_0+h) = y_0 + \frac{1}{90}\,[7(k_0+k_5) + 32\,(k_2+k_4) + 12\,k_3]\,, \tag{29d}$$

$$(\tilde{y}_\Omega(x_0+ch))'_{c=1} = \frac{1}{h}\,k_6 = f\,(x_0+h,\, \tilde{y}_{5,6}(x_0+h))\,. \tag{29e}$$

THE SECOND MAIN CONTINUOUS FORMULA. Our second main fourth order seven-stage continuous formula in the variable weight form may be written:

$$\tilde{y}_{4,7}(x_0+ch) = y_0 + \sum_{i=0}^{6} [w_i(c)]\, k_i, \tag{30a}$$

where

$$\begin{aligned}
w_0(c) &= \frac{c}{90}\,(90 - 267\,c + 292\,c^2 - 108\,c^3),\\
w_1(c) &= 0,\\
w_2(c) &= \frac{16}{45}\,c^2\,(9 - 14\,c + 6c^2),\\
w_3(c) &= \frac{2}{15}\,c^2\,(9 - 14\,c + 6c^2) = \frac{3}{8}\,w_2,\\
w_4(c) &= \frac{16}{45}\,c^2\,(-6 + 16\,c - 9c^2),\\
w_5(c) &= \frac{7}{90}\,c^2\,(9 - 14\,c + 6c^2) = \frac{7}{32}\,w_2,\\
w_6(c) &= c^3\,(-1 + c).
\end{aligned} \tag{30b}$$

In equivalent polynomial form, we have:

$$\tilde{y}_{4,7}(x_0 + ch) \equiv \tilde{y}_\omega(x_0 + ch) = y_0 + \sum_{i=1}^{4} \omega_i\, c^i, \tag{31a}$$

where

$$\begin{aligned}
\omega_1 &= k_0,\\
\omega_2 &= \frac{1}{30}\,(-89\,k_0 + 96\,k_2 + 36\,k_3 - 64\,k_4 + 21\,k_5),\\
\omega_3 &= \frac{1}{45}\,(146\,k_0 - 224\,k_2 - 84\,k_3 + 256\,k_4 - 49\,k_5 - 45\,k_6),\\
\omega_4 &= \frac{1}{15}\,(-18\,k_0 + 32\,k_2 + 12\,k_3 - 48\,k_4 + 7k_5 + 15\,k_6).
\end{aligned} \tag{31b}$$

The derivative with respect to x, is given by:

$$(\tilde{y}_\omega(x_0 + ch))' = \frac{1}{h} \sum_{i=1}^{4} i\,\omega_i\, c^{i-1}. \tag{31c}$$

As expected, as in the first main formula, we have

$$(\tilde{y}_\omega(x_0 + ch))_{c=1} = \tilde{y}_{5,6}(x_0 + h) = y_0 + \frac{1}{90}\,[7\,(k_0 + k_5) + 32\,(k_2 + k_4) + 12\,k_3]\,, \tag{31d}$$

$$(\tilde{y}_\omega(x_0 + ch))'_{c=1} = \frac{1}{h}\,k_6 = f\,(x_0 + h,\, \tilde{y}_{5,6}(x_0 + h))\,. \tag{31e}$$

It follows that the considered two main algorithms, $\tilde{y}_\Omega$ and $\tilde{y}_\omega$, generate approximate C^1 solutions. However, it should be mentioned that the first main algorithm, (28a,b) or (29a,b), generates the most accurate approximations to the solution $y(x)$, as well as to its derivative $y'(x)$ at interior points $x \in (x_0,\, x_0 + h)$. The second main algorithm is recommended rather for its simplicity, particularly in its variable weight form, (30b).

An estimation of errors associated with the approximations to the solution $y(x)$ as well as its derivative $y'(x)$, generated by the two main algorithms, $\tilde{y}_\Omega$ and $\tilde{y}_\omega$, may be obtained through the use of the algorithm:

$$\tilde{y}_{4,7}(x_0 + ch) \equiv \tilde{y}_T(x_0 + ch) = y_0 + \sum_{i=1}^{4} T_i\, c^i, \tag{32a}$$

where

$$\begin{aligned} T_1 &= k_0, \\ T_2 &= \frac{1}{54}(-161\,k_0 + 176\,k_2 + 60\,k_3 - 112\,k_4 + 28\,k_5 + 9k_6), \\ T_3 &= \frac{1}{225}(718\,k_0 - 1072\,k_2 - 492\,k_3 + 1328\,k_4 - 392\,k_5 - 90\,k_6), \\ T_4 &= \frac{1}{60}(-68\,k_0 + 112\,k_2 + 72\,k_3 - 208\,k_4 + 77\,k_5 + 15\,k_6). \end{aligned} \tag{32b}$$

The derivative of this approximate solution with respect to x is:

$$(\tilde{y}_T(x_0 + ch))' = \frac{1}{h}\sum_{i=1}^{4} i\,T_i\,c^{i-1}. \tag{32c}$$

Then

$$\tilde{y}_T(x_0 + h) = y_0 + \frac{1}{2700}(206\,k_0 + 976\,k_2 + 336\,k_3 + 976\,k_4 + 161\,k_5 + 45\,k_6), \tag{32d}$$

$$(\tilde{y}_T(x_0 + ch))'_{c=1} = \frac{1}{h}\left[\frac{1}{675}(52\,k_0 - 208\,k_2 + 312\,k_3 - 208\,k_4 + 637\,k_5 + 90\,k_6)\right]. \tag{32e}$$

With the subsets of the set of incremental coefficient defined in (27), that is for Method 1, which permitted us to generate our two main continuous algorithms $\tilde{y}_\Omega(x_0 + ch)$ and $\tilde{y}_\omega(x_0 + ch)$, we have additionally generated also the following six embedded processes of orders one through four (inclusive) which, for $c = 1$, that is at end of the step $x = x_0 + h$, become (discrete) processes of orders one through five. We have:

$$\tilde{y}_{4,6}(x_0 + ch) \equiv \tilde{y}_S(x_0 + ch) = y_0 + \sum_{i=1}^{4} S_i\,c^i, \tag{33a}$$

$$\tilde{y}_{3,5}(x_0 + ch) \equiv \tilde{y}_R(x_0 + ch) = y_0 + \sum_{i=1}^{4} R_i\,c^i, \tag{34a}$$

$$\tilde{y}_{3,4}(x_0 + ch) \equiv \tilde{y}_Q(x_0 + ch) = y_0 + \sum_{i=1}^{3} Q_i\,c^i, \tag{35a}$$

$$\tilde{y}_{2,3}(x_0 + ch) \equiv \tilde{y}_p(x_0 + ch) = y_0 + \sum_{i=1}^{3} P_i\,c^i, \tag{36a}$$

$$\tilde{y}_{2,2}(x_0 + ch) = \tilde{y}_N(x_0 + ch) = y_0 + \sum_{i=1}^{2} N_i\,c^i, \tag{37a}$$

$$\tilde{y}_1(x_0 + ch) = y_0 + ck_0. \tag{38a}$$

The pertinent terms, S's, R's, Q's, P's, and N's are defined as follows:

$$\begin{aligned} S_1 &= k_0, \\ S_2 &= \frac{1}{30}[-89\,k_0 + 96\,k_2 + 36\,k_3 - 64\,k_4 + 21\,k_5], \\ S_3 &= \frac{2}{45}[71\,k_0 - 104\,k_2 - 54\,k_3 + 136\,k_4 - 49\,k_5], \\ S_4 &= \frac{2}{9}[-5k_0 + 8k_2 + 6k_3 - 16\,k_4 + 7k_5]; \end{aligned} \tag{33b}$$

$$\begin{aligned} R_1 &= k_0, \\ R_2 &= \frac{1}{3}\left[-11\,k_0 + 18\,k_2 - 9k_3 + 2k_4\right], \\ R_3 &= \frac{8}{3}\left[2k_0 - 5k_2 + 4k_3 - k_4\right], \\ R_4 &= \frac{8}{3}\left[-k_0 + 3k_2 - 3k_3 + k_4\right]; \end{aligned} \tag{34b}$$

$$\begin{aligned} Q_1 &= k_0; \\ Q_2 &= -3k_0 + 4k_2 - k_3, \\ Q_3 &= \frac{8}{3}\left[k_0 - 2k_2 + k_3\right]; \end{aligned} \tag{35b}$$

$$\begin{aligned} P_1 &= k_0, \\ P_2 &= -5k_0 + 9k_1 - 4k_2, \\ P_3 &= 8\left[k_0 - 3k_1 + 2k_2\right]; \end{aligned} \tag{36b}$$

$$\begin{aligned} N_1 &= k_0, \\ N_2 &= 3\left[-k_0 + k_1\right]. \end{aligned} \tag{37b}$$

Note that for $c = 1$, the fourth order algorithm is upgraded to fifth order, while the orders of the other algorithms remain unchanged. Specifically, we have (for $c = 1$):

$$\tilde{y}_{5,6}(x_0 + h) \equiv \tilde{y}_S(x_0 + h) = y_0 + \frac{1}{90}\left[7(k_0 + k_5) + 32\,(k_2 + k_4) + 12\,k_3\right], \tag{33c}$$

$$\tilde{y}_{3,5}(x_0 + h) \equiv \tilde{y}_R(x_0 + h) = y_0 + \frac{1}{3}\left[2k_2 - k_3 + 2k_4\right], \tag{34c}$$

$$\tilde{y}_{3,4}(x_0 + h) \equiv \tilde{y}_Q(x_0 + h) = y_0 + \frac{1}{3}\left[2k_0 - 4k_2 + 5k_3\right], \tag{35c}$$

$$\tilde{y}_{2,3}(x_0 + h) \equiv \tilde{y}_P(x_0 + h) = y_0 + 4k_0 - 15\,k_1 + 12\,k_2, \tag{36c}$$

$$\tilde{y}_{2,2}(x_0 + h) \equiv \tilde{y}_N(x_0 + h) = y_0 - 2k_0 + 3k_1, \tag{37c}$$

$$\tilde{y}_1(x_0 + h) = y_0 + k_0. \tag{38c}$$

At this time, it must be recalled that an error estimation algorithm will provide reliable error estimates only if the chosen step-size h is sufficiently small, and that our rule described in Section 6 provides such an initial step-size. Then for $c = 1$, that is at the end of the step, $x = x_0 + h$, the use of the error estimation algorithm (32d), $\tilde{y}_T(x_0 + h)$, would provide useful, reliable estimates for the absolute errors in the main approximations $\tilde{y}_\Omega(x_0 + h)$ and $\tilde{y}_\omega(x_0 + h)$, specifically by the formula

$$\begin{gathered} e(x_0 + h) = \left|\tilde{y}_{5,6}(x_0 + h) - \tilde{y}_T(x_0 + h)\right|, \qquad \text{since} \\ \tilde{y}_\Omega(x_0 + h) = \tilde{y}_\omega(x_0 + h) = \tilde{y}_{5,6}(x_0 + h). \end{gathered} \tag{39}$$

If the tolerated maximum absolute error E is specified, and it is found for instance, that $e(x_0 + h) > E$, then the step-size h can be adjusted to nh where $n \approx \sqrt[5]{E/e}$ for the main approximate solutions, to satisfy the specified requirement.

At the interior points, $x = x_0 + ch$, $c \in (0, 1)$, the estimation of errors in the main approximate solutions $\tilde{y}_\Omega$ and y_ω is somewhat complex, since $\tilde{y}_\Omega$, $\tilde{y}_\omega$, and $\tilde{y}_T$ are all fourth order seven-stage processes, and sometimes the respective curves may intersect. Nevertheless, their comparison with each other, together with the approximate solutions obtained through the other embedded processes of increasing accuracy, $\tilde{y}_1$, $\tilde{y}_{2,2}$, $\tilde{y}_{2,3}$, $\tilde{y}_{3,4}$, $\tilde{y}_{3,5}$, lead us, in general, to satisfactory estimates. The same procedure may also be applied for estimates of errors in the derivatives of the main approximate solutions.

Method 2

As in the case of Method 1, with a different set of seven incremental coefficient $\{k_0, \dots, k_5, k_6\}$, where k_6 is a "borrowed" stage, we have derived two main continuous algorithms $\tilde{y}_\Omega(x_0 + ch)$ and $\tilde{y}_\omega(x_0 + ch)$, of fourth order, involving seven stages, of which $\tilde{y}_\Omega$ is the more accurate. Both of these two main algorithms are upgraded to the same fifth order six-stage process at the end of the step $x = x_0 + h$, denoted as usual by $\tilde{y}_{5,6}(x_0 + h)$. We have generated another third order five-stage algorithm $\tilde{y}_T(x_0 + ch)$ for estimation of errors. These algorithms are presented below. Additionally, we are presenting algorithms of different orders mostly for use in the error estimation process at interior points, $x = x_0 + ch$, $c \in (0, 1)$, for the approximate continuous solutions $\tilde{y}_\Omega$ and $\tilde{y}_\omega$ and their first order derivatives with respect to x, and also (if found desirable) to be used as low order continuous Runge-Kutta processes.

The set of incremental coefficients is

$$\begin{aligned}
k_0 &= hf(x_0, y_0), \\
k_1 &= hf\left(x_0 + \frac{1}{6}h,\ y_0 + \frac{1}{6}k_0\right), \\
k_2 &= hf\left(x_0 + \frac{1}{4}h,\ y_0 + \frac{1}{16}(k_0 + 3k_1)\right), \\
k_3 &= hf\left(x_0 + \frac{2}{5}h,\ y_0 + \frac{2}{125}(7k_0 - 6k_1 + 24k_2)\right), \\
k_4 &= hf\left(x_0 + \frac{4}{5}h,\ y_0 + \frac{2}{375}(87k_0 - 36k_1 - 176k_2 + 275k_3)\right), \\
k_5 &= hf\left(x_0 + h,\ y_0 + \frac{1}{440}(-1111k_0 + 528k_1 + 3648k_2 - 3300k_3 + 675k_4)\right), \\
k_6 &= hf(x_0 + h,\ \tilde{y}_{5,6}(x_0 + h)), \\
\tilde{y}_{5,6}(x_0 + h) &= y_0 + \frac{3}{32}k_0 + \frac{64}{297}k_2 + \frac{125}{432}k_3 + \frac{125}{352}k_4 + \frac{5}{108}k_5 \\
&= y_0 + \frac{1}{9504}(891k_0 + 2048k_2 + 2750k_3 + 3375k_4 + 440k_5).
\end{aligned} \tag{40}$$

THE FIRST MAIN CONTINUOUS FORMULA.

$$\tilde{y}_{4,7}(x_0 + ch) \equiv \tilde{y}_\Omega(x_0 + ch) = y_0 + \sum_{i=1}^{5} \Omega_i c^i, \tag{41a}$$

where

$$\begin{aligned}
\Omega_1 &= k_0, \\
\Omega_2 &= -\frac{35}{8}k_0 + \frac{1024}{99}k_2 - \frac{125}{18}k_3 + \frac{125}{88}k_4 - \frac{35}{18}k_5 + \frac{3}{2}k_6, \\
\Omega_3 &= \frac{69}{8}k_0 - \frac{9728}{297}k_2 + \frac{3125}{108}k_3 - \frac{625}{88}k_4 + \frac{170}{27}k_5 - 4k_6, \\
\Omega_4 &= 5\left(-\frac{49}{32}k_0 + \frac{64}{9}k_2 - \frac{1025}{144}k_3 + \frac{75}{32}k_4 - \frac{47}{36}k_5 + \frac{1}{2}k_6\right), \\
\Omega_5 &= 5\left(\frac{1}{2}k_0 - \frac{256}{99}k_2 + \frac{25}{9}k_3 - \frac{25}{22}k_4 + \frac{4}{9}k_5\right).
\end{aligned} \tag{41b}$$

The derivative with respect to x, is given by:

$$(\tilde{y}_\Omega(x_0 + ch))' = \frac{1}{h}\sum_{i=1}^{5} i\,\Omega_i c^{i-1}. \tag{41c}$$

As expected, we have

$$(\tilde{y}_\Omega(x_0+ch))_{c=1} = \tilde{y}_{5,6}(x_0+h) = y_0 + \frac{3}{32}k_0 + \frac{64}{297}k_2 + \frac{125}{432}k_3 + \frac{125}{352}k_4 + \frac{5}{108}k_5, \tag{41d}$$

$$(\tilde{y}_\Omega(x_0+ch))'_{c=1} = \frac{1}{h}k_6 = f\left(x_0+h,\, \tilde{y}_{5,6}(x_0+h)\right). \tag{41e}$$

THE SECOND MAIN CONTINUOUS FORMULA.

$$\tilde{y}_{4,7}(x_0+ch) = y_0 + \sum_{i=1}^{4} \omega_i c^i, \tag{42a}$$

where

$$\begin{aligned}
\omega_1 &= k_0,\\
\omega_2 &= -\frac{55}{24}k_0 - \frac{128}{297}k_2 + \frac{125}{27}k_3 - \frac{875}{264}k_4 - \frac{5}{54}k_5 + \frac{3}{2}k_6,\\
\omega_3 &= \frac{47}{24}k_0 + \frac{512}{297}k_2 - \frac{875}{108}k_3 + \frac{2125}{264}k_4 + \frac{10}{27}k_6 - 4k_6,\\
\omega_4 &= 5\left(-\frac{11}{96}k_0 - \frac{64}{297}k_2 + \frac{325}{432}k_3 - \frac{925}{1056}k_4 - \frac{5}{108}k_5 + \frac{1}{2}k_6\right).
\end{aligned} \tag{42b}$$

Then

$$(\tilde{y}_\omega(x_0+ch))' = \frac{1}{h}\sum_{i=1}^{4} i\,\omega_i c^{i-1}, \tag{42c}$$

and

$$(\tilde{y}_\omega(x_0+ch))_{c=1} = \tilde{y}_{5,6}(x_0+h), \tag{42d}$$

$$(\tilde{y}_\omega(x_0+ch))'_{c=1} = \frac{1}{h}k_6 = f\left(x_0+h,\, \tilde{y}_{5,6}(x_0+h)\right). \tag{42e}$$

An estimation of errors associated with the approximations to the solution $y(x)$ as well as its derivative $y'(x)$, generated by $\tilde{y}_\Omega(x_0+ch)$ and $\tilde{y}_\omega(x_0+ch)$ may be obtained through the use of algorithm:

$$\tilde{y}_{3,5}(x_0+ch) \equiv \tilde{y}_T(x_0+ch) = y_0 + \sum_{i=1}^{4} T_i c^i, \tag{43a}$$

where

$$\begin{aligned}
T_1 &= k_0,\\
T_2 &= -\frac{31}{8}k_0 + \frac{256}{33}k_2 - \frac{25}{6}k_3 + \frac{25}{88}k_4,\\
T_3 &= 5\left(\frac{29}{24}k_0 - \frac{128}{33}k_2 + \frac{35}{12}k_3 - \frac{65}{264}k_4\right),\\
T_4 &= 25\left(-\frac{1}{8}k_0 + \frac{16}{33}k_2 - \frac{5}{12}k_3 + \frac{5}{88}k_4\right).
\end{aligned} \tag{43b}$$

Then

$$(\tilde{y}_T(x_0+ch))' = \frac{1}{h}\sum_{i=1}^{4} i\,T_i c^{i-1}, \tag{43c}$$

and

$$(\tilde{y}_T(x_0+ch))_{c=1} = \tilde{y}_{4,5}(x_0+h) = y_0 + \frac{1}{264}(11k_0 + 128k_2 + 125k_4), \tag{43d}$$

$$(\tilde{y}_T(x_0+ch))'_{c=1} = \frac{1}{h}\left[\frac{1}{88}(-99k_0 + 512k_2 - 550k_3 + 225k_4)\right]. \tag{43e}$$

We have additionally generated six embedded continuous processes of orders one through four (inclusive) which, for $c = 1$, that is at the end of the step $x = x_0 + h$, become (discrete) processes of orders one through five. These continuous processes (for the sake of completeness including $\tilde{y}_T$) are as follows:

$$\tilde{y}_{4,7}(x_0 + ch) \equiv \tilde{y}_R(x_0 + ch) = y_0 + \sum_{i=1}^{4} R_i\, c^i, \tag{44a}$$

$$\tilde{y}_{3,6}(x_0 + ch) \equiv \tilde{y}_S(x_0 + ch) = y_0 + \sum_{i=1}^{5} S_i\, c^i, \tag{45a}$$

$$\tilde{y}_{3,5}(x_0 + ch) \equiv \tilde{y}_T(x_0 + ch) = y_0 + \sum_{i=1}^{4} T_i\, c^i, \tag{43a}$$

$$\tilde{y}_{3,4}(x_0 + ch) \equiv \tilde{y}_Q(x_0 + ch) = y_0 + \sum_{i=1}^{3} Q_i\, c^i, \tag{46a}$$

$$\tilde{y}_{2,3}(x_0 + ch) \equiv \tilde{y}_P(x_0 + ch) = y_0 + \sum_{i=1}^{3} P_i\, c^i, \tag{47a}$$

$$\tilde{y}_{2,2}(x_0 + ch) \equiv \tilde{y}_N(x_0 + ch) = y_0 + N_1\, c + N_2\, c^2, \tag{48a}$$

$$\tilde{y}_1(x_0 + ch) = y_0 + ck_0. \tag{49a}$$

The pertinent terms, R's, S's, T's, Q's, P's, and N's are defined as follows:

$$\begin{aligned} R_1 &= k_0, \\ R_2 &= \frac{1}{24}\,(-57\,k_0 + 100\,k_3 - 75\,k_4 - 4k_5 + 36\,k_6), \\ R_3 &= \frac{1}{24}\,(55\,k_0 - 150\,k_3 + 175\,k_4 + 16\,k_5 - 96\,k_6), \\ R_4 &= \frac{5}{96}\,(-15\,k_0 + 50\,k_3 - 75\,k_4 - 8k_5 + 48\,k_6); \end{aligned} \tag{44b}$$

$$\begin{aligned} S_1 &= k_0, \\ S_2 &= \frac{1}{792}\,(-3465\,k_0 + 8192\,k_2 - 5500\,k_3 + 1125\,k_4 - 352\,k_5), \\ S_3 &= \frac{1}{2376}\,(20493\,k_0 - 77824\,k_2 + 68750\,k_3 - 16875\,k_4 + 5456\,k_5), \\ S_4 &= \frac{5}{288}\,(-441\,k_0 + 2048\,k_2 - 2050\,k_3 + 675\,k_4 - 232\,k_5), \\ S_5 &= \frac{5}{198}\,(99\,k_0 - 512\,k_2 + 550\,k_3 - 225\,k_4 + 88\,k_5); \end{aligned} \tag{45b}$$

$$T_i, \qquad i = 1,2,3,4, \qquad \text{defined above}; \tag{43b}$$

$$\begin{aligned} Q_1 &= k_0, \\ Q_2 &= \frac{1}{12}\,(-39\,k_0 + 64\,k_2 - 25\,k_3), \\ Q_3 &= \frac{10}{9}\,(3k_0 - 8k_2 + 5k_3); \end{aligned} \tag{46b}$$

$$\begin{aligned} P_1 &= k_0, \\ P_2 &= -5k_0 + 9k_1 - 4k_2, \\ P_3 &= 8\,(k_0 - 3k_1 - 2k_2); \end{aligned} \tag{47b}$$

$$\begin{aligned} N_1 &= k_0, \\ N_2 &= 3\,(-k_0 + k_1). \end{aligned} \tag{48b}$$

For $c = 1$, the third order algorithms $\tilde{y}_{3,5} \equiv \tilde{y}_T$ and $\tilde{y}_{3,6} \equiv \tilde{y}_S$ are upgraded to fourth and fifth order, respectively, while the orders of other algorithms remain unchanged. Specifically we have:

$$\tilde{y}_{4,6}(x_0 + h) \equiv \tilde{y}_R(x_0 + h) = y_0 + \frac{1}{96}\,(13\,k_0 + 50\,k_3 + 25\,k_4 + 8k_5), \tag{44c}$$

$$\begin{aligned} \tilde{y}_{5,6}(x_0 + h) &\equiv \tilde{y}_S(x_0 + h) \equiv \tilde{y}_\Omega(x_0 + h) \equiv \tilde{y}_\omega(x_0 + h) \\ &= y_0 + \frac{1}{9504}\,(891\,k_0 + 2048\,k_2 + 2750\,k_3 + 3375\,k_4 + 440\,k_5), \end{aligned} \tag{45c}$$

$$\tilde{y}_{4,5}(x_0 + h) \equiv \tilde{y}_T(x_0 + h) = y_0 + \frac{1}{264}\,(11\,k_0 + 128\,k_2 + 125\,k_4), \tag{43d}$$

$$\tilde{y}_{3,4}(x_0 + h) \equiv \tilde{y}_Q(x_0 + h) = y_0 + \frac{1}{36}\,(39\,k_0 - 128\,k_2 + 125\,k_3), \tag{46c}$$

$$\tilde{y}_{2,3}(x_0 + h) \equiv \tilde{y}_P(x_0 + h) = y_0 + 4k_0 - 15\,k_1 + 12\,k_2, \tag{47c}$$

$$\tilde{y}_{2,2}(x_0 + h) \equiv \tilde{y}_N(x_0 + h) = y_0 - 2k_0 + 3k_1, \tag{48c}$$

$$\tilde{y}_1(x_0 + h) = y_0 + k_0. \tag{49b}$$

Concerning the estimation of errors relative to approximate solutions generated by the two main algorithms $\tilde{y}_\Omega$ and $\tilde{y}_\omega$, the comments made about Method 1 are valid also for Method 2. In particular, the formula

$$e(x_0 + h) = |\tilde{y}_{5,6}(x_0 + h) - \tilde{y}_T(x_0 + h)|, \tag{50}$$
$$\tilde{y}_{5,6}(x_0 + h) \equiv \tilde{y}_\Omega(x_0 + h) \equiv \tilde{y}_\omega(x_0 + h),$$

provides reliable estimates for the absolute errors associated with the main approximate solutions at $x = x_0 + h$.

For the estimation of absolute errors associated with the approximate solutions generated with either of the main algorithms $\tilde{y}_\Omega$ and $\tilde{y}_\omega$ at the interior points $x \in (x_0, x_0 + h)$, the formula (50) is written:

$$e(x_0 + ch) = |\tilde{y}_{4,7}(x_0 + ch) - \tilde{y}_{3,5}(x_0 + ch)|, \tag{51}$$

where

$$\begin{aligned} \tilde{y}_{4,7}(x_0 + ch) &= \tilde{y}_\Omega(x_0 + ch), && \text{as in (41a), or} \\ \tilde{y}_{4,7}(x_0 + ch) &\equiv \tilde{y}_\omega(x_0 + ch), && \text{as in (42a),} \end{aligned}$$

depending whether the main formula used is $\tilde{y}_\Omega$ or $\tilde{y}_\omega$, and

$$\tilde{y}_{3,5}(x_0 + ch) \equiv \tilde{y}_T(x_0 + ch).$$

Similarly, the formula

$$e'(x_0 + ch) = \left|(\tilde{y}_{4,7}(x_0 + ch))' - (\tilde{y}_{3,5}(x_0 + ch))'\right|, \tag{52}$$

relates to the absolute errors involved in the derivatives of the main approximate solutions with respect to x. In this connection, let us recall that at the nodal point $x = x_0 + h$ (or for $c = 1$) the derivatives of $\tilde{y}_{4,7}(x_0 + ch)$ and $\tilde{y}_{3,5}(x_0 + ch)$ are given in (41e), (42e), and (43e), respectively.

Finally, it must be pointed out that we now have at our disposal (including $\tilde{y}_\Omega$ and $\tilde{y}_\omega$) an array of nine distinct algorithms, providing us at all points $x = x_0 + ch$, $c \in (0, 1]$, with a set of nine approximate solutions of increasing order (and/or increasing stages), ranging from the first to fifth order. These nine consecutively obtained approximate solutions show *the trend of* *the overall approximation process* and help us to detect arithmetical or other kinds of errors and mistakes. However, still better, they allow us at "no cost" to project a value for the next higher order approximate solution, particularly at the nodal point $x = x_0 + h$, a sixth order approximate solution. And interestingly, whenever our consecutive approximate solutions, $\tilde{y}_i$ of orders $i = 1, \dots, 5$, are of oscillatory type—an indication that the power series expansion of the exact solution $y(x)$ is an alternating series—for instance, if it is obtained consecutively, $\tilde{y}_1 < \tilde{y}_2$, $\tilde{y}_1 < \tilde{y}_3 < \tilde{y}_2$, $\tilde{y}_1 < \tilde{y}_3 < \tilde{y}_4 < \tilde{y}_2$, $\tilde{y}_1 < \tilde{y}_3 < \tilde{y}_5 < \tilde{y}_4 < \tilde{y}_2$, then the last two consecutive approximate solutions, namely $\tilde{y}_4$ and $\tilde{y}_5$, may be reliably considered in most cases as constituting bounds for the exact solution y; that is, $\tilde{y}_4 < y < \tilde{y}_5$, or $\tilde{y}_5 < y < \tilde{y}_4$. Further research in this direction is in progress, and in particular, the case of sixth order processes applied to oscillatory problems. All these algorithms apply to initial value problems regardless of whether they are of scalar or vector type.

Method 3

The procedures are similar to those used in Method 1 and 2, except that the set of incremental coefficients $\{k_0, \dots, k_5, k_6\}$ is different and thus the two main algorithms generated, $\tilde{y}_\Omega$ and $\tilde{y}_\omega$, are different; the former as before provides more accurate approximate solutions than the latter.

The set of incremental coefficients in Method 3, is [46, p. 81]:

$$\begin{aligned}
k_0 &= hf(x_0, y_0), \\
k_1 &= hf\left(x_0 + \frac{1}{5}h,\ y_0 + \frac{1}{5}k_0\right), \\
k_2 &= hf\left(x_0 + \frac{3}{10}h,\ y_0 + \frac{3}{40}(k_0 + 3k_1)\right), \\
k_3 &= hf\left(x_0 + \frac{3}{5}h,\ y_0 + \frac{3}{10}(k_0 - 3k_1 + 4k_2)\right), \\
k_4 &= hf\left(x_0 + \frac{5}{6}h,\ y_0 + \frac{5}{5832}(227\,k_0 - 135\,k_1 + 320\,k_2 + 560\,k_3)\right), \\
k_5 &= hf\left(x_0 + h,\ y_0 + \frac{1}{540}(-614\,k_0 + 1350\,k_1 + 175\,k_2 - 1100\,k_3 + 729\,k_4)\right), \\
k_6 &= hf\left(x_0 + h,\ \tilde{y}_{5,6}(x_0 + h)\right), \\
\tilde{y}_{5,6}(x_0 + ch) &= y_0 + \frac{13}{135}k_0 + \frac{625}{1512}k_2 + \frac{125}{756}k_3 + \frac{81}{280}k_4 + \frac{1}{28}k_5 \\
&= y_0 + \frac{1}{7560}(728\,k_0 + 3125\,k_2 + 1250\,k_3 + 2187\,k_4 + 270\,k_5).
\end{aligned} \tag{53}$$

With this set of incremental coefficients (53) the following two main algorithms $\tilde{y}_\Omega$ and $\tilde{y}_\omega$, which generate C^1 solutions, have been established.

THE FIRST MAIN CONTINUOUS FORMULA.

$$\tilde{y}_{4,7}(x_0 + ch) \equiv \tilde{y}_\Omega(x_0 + ch) = y_0 + \sum_{i=1}^{5} \Omega_i\, c^i, \tag{54a}$$

where

$$\begin{aligned}
\Omega_1 &= k_0, \\
\Omega_2 &= -\frac{18}{5}k_0 + \frac{625}{84}k_2 - \frac{625}{84}k_3 + \frac{729}{140}k_4 - \frac{87}{28}k_5 + \frac{3}{2}k_6, \\
\Omega_3 &= \frac{799}{135}k_0 - \frac{3625}{189}k_2 + \frac{1037}{378}k_3 - \frac{729}{35}k_4 + \frac{149}{14}k_5 - 4k_6, \\
\Omega_4 &= -\frac{41}{9}k_0 + \frac{9125}{504}k_2 - \frac{2000}{63}k_3 + \frac{1539}{56}k_4 - \frac{165}{14}k_5 + \frac{5}{2}k_6, \\
\Omega_5 &= \frac{4}{3}k_0 - \frac{125}{21}k_2 + \frac{250}{21}k_3 - \frac{81}{7}k_4 + \frac{30}{7}k_5.
\end{aligned} \tag{54b}$$

The derivative with respect to x $(c = (x - x_0)/h)$, is obtained by:

$$(\tilde{y}_\Omega(x_0 + ch))' = \frac{1}{h} \sum_{i=1}^{5} i\,\Omega_i\, c^{i-1}, \tag{54c}$$

and

$$\begin{aligned}(\tilde{y}_\Omega(x_0 + ch))_{c=1} &= \tilde{y}_{5,6}(x_0 + h)\\ &= y_0 + \frac{1}{7560}\,(728\,k_0 + 3125\,k_2 + 1250\,k_3 + 2187\,k_4 + 270\,k_5),\end{aligned} \tag{54d}$$

$$(\tilde{y}_\Omega(x_0 + ch))'_{c=1} = \frac{k_6}{h} = f\,(x_0 + h,\, \tilde{y}_{5,6}(x_0 + h)). \tag{54e}$$

THE SECOND MAIN CONTINUOUS FORMULA.

$$\tilde{y}_{4,7}(x_0 + ch) \equiv \tilde{y}_\omega(x_0 + ch) = y_0 + \sum_{i=1}^{4} \omega_i\, c^i, \tag{55a}$$

where

$$\begin{aligned}\omega_1 &= k_0,\\ \omega_2 &= \frac{1}{8}\left(-\frac{307}{15}\,k_0 + \frac{625}{28}\,k_2 + \frac{625}{42}\,k_3 - \frac{4293}{140}\,k_4 + \frac{27}{14}\,k_5 + 12\,k_6\right),\\ \omega_3 &= \frac{1}{28}\left(\frac{9457}{135}\,k_0 - \frac{11875}{108}\,k_2 - \frac{4625}{54}\,k_3 + \frac{4941}{20}\,k_4 - \frac{19}{2}\,k_5 - 112\,k_6\right),\\ \omega_4 &= \frac{1}{28}\left(-\frac{427}{18}\,k_0 + \frac{3125}{72}\,k_2 + \frac{1375}{36}\,k_3 - \frac{1053}{8}\,k_4 + \frac{15}{4}\,k_5 + 70\,k_6\right).\end{aligned} \tag{55b}$$

The derivative with respect to x is given by:

$$(\tilde{y}_\omega(x_0 + ch))' = \frac{1}{h} \sum_{i=1}^{4} i\,\omega_i\, c^{i-1}, \tag{55c}$$

and

$$(\tilde{y}_\omega(x_0 + ch))_{c=1} = \tilde{y}_{5,6}(x_0 + h), \tag{55d}$$

$$(\tilde{y}_\omega(x_0 + ch))'_{c=1} = \frac{1}{h}\,k_6 = f\,(x_0 + h,\, \tilde{y}_{5,6}(x_0 + h)). \tag{55e}$$

An estimation of errors associated with the approximations to the solution $y(x)$ as well as its derivative $y'(x)$, generated by $\tilde{y}_\Omega(x_0 + ch)$ and $\tilde{y}_\omega(x_0 + ch)$ may be obtained through the use of the algorithm:

$$\tilde{y}_{3,5}(x_0 + ch) \equiv \tilde{y}_T(x_0 + ch) = y_0 + \sum_{i=1}^{4} T_i\, c^i, \tag{56a}$$

where

$$\begin{aligned}T_1 &= k_0,\\ T_2 &= \frac{1}{840}\,(-2604\,k_0 + 4375\,k_2 - 2500\,k_3 + 729\,k_4),\\ T_3 &= \frac{1}{756}\,(2912\,k_0 - 7525\,k_2 + 6800\,k_3 - 2187\,k_4),\\ T_4 &= \frac{5}{168}\,(-56\,k_0 + 175\,k_2 - 200\,k_3 + 81\,k_4).\end{aligned} \tag{56b}$$

The derivative with respect to x is given by:

$$(\tilde{y}_T(x_0+ch))' = \frac{1}{h}\sum_{i=1}^{4} i\,T_i\,c^{i-1}, \tag{56c}$$

and

$$(\tilde{y}_T(x_0+ch))_{c=1} = \tilde{y}_{4,5}(x_0+h) = y_0 + \frac{1}{1890}\,(161\,k_0 + 875\,k_2 + 125\,k_3 + 729\,k_4), \tag{56d}$$

$$(y_T(x_0+ch))'_{c=1} = \frac{1}{h}\left[\frac{1}{90}(-28\,k_0 + 125\,k_2 - 250\,k_3 + 243\,k_4)\right]. \tag{56e}$$

We have again additionally generated six embedded continuous processes of orders one through five. These continuous processes (including $\tilde{y}_T$) are as follows:

$$\tilde{y}_{3,5}(x_0+ch) \equiv \tilde{y}_T(x_0+ch) = y_0 + \sum_{i=1}^{4} T_i\,c^i, \tag{56a}$$

$$\tilde{y}_{4,7}(x_0+ch) \equiv \tilde{y}_R(x_0+ch) = y_0 + \sum_{i=1}^{4} R_i\,c^i, \tag{57a}$$

$$\tilde{y}_{3,6}(x_0+ch) \equiv \tilde{y}_S(x_0+ch) = y_0 + \sum_{i=1}^{5} S_i\,c^i, \tag{58a}$$

$$\tilde{y}_{3,4}(x_0+ch) \equiv \tilde{y}_Q(x_0+ch) = y_0 + \sum_{i=1}^{3} Q_i\,c^i, \tag{59a}$$

$$\tilde{y}_{2,3}(x_0+ch) \equiv \tilde{y}_P(x_0+ch) = y_0 + \sum_{i=1}^{3} P_i\,c^i, \tag{60a}$$

$$\tilde{y}_{2,2}(x_0+ch) \equiv \tilde{y}_N(x_0+ch) = y_0 + N_1\,c + N_2\,c^2, \tag{61a}$$

$$\tilde{y}_1(x_0+ch) = y_0 + ck_0. \tag{62a}$$

The pertinent terms, R's, S's, T's, Q's, P's, and N's are defined as follows:

$$\begin{aligned}
R_1 &= k_0,\\
R_2 &= \frac{1}{420}(-812\,k_0 + 3125\,k_3 - 3888\,k_4 + 945\,k_5 + 630\,k_6),\\
R_3 &= \frac{1}{630}(1022\,k_0 - 6875\,k_3 + 10368\,k_4 - 1995\,k_5 - 2520\,k_6),\\
R_4 &= \frac{1}{28}(-14\,k_0 + 125\,k_3 - 216\,k_4 + 35\,k_5 + 70\,k_6);
\end{aligned} \tag{57b}$$

$$\begin{aligned}
S_1 &= k_0,\\
S_2 &= \frac{1}{420}(-1512\,k_0 + 3125\,k_2 - 3125\,k_3 + 2187\,k_4 - 675\,k_5),\\
S_3 &= \frac{1}{1890}(11186\,k_0 - 36250\,k_2 + 51875\,k_3 - 39366\,k_4 + 12555\,k_5),\\
S_4 &= \frac{1}{540}(-2296\,k_0 + 9125\,k_2 - 16000\,k_3 + 13851\,k_4 - 4680\,k_5),\\
S_5 &= \frac{1}{21}(28\,k_0 - 125\,k_2 + 250\,k_3 - 243\,k_4 + 90\,k_5);
\end{aligned} \tag{58b}$$

$$T_i, \qquad i = 1,2,3,4, \qquad \text{defined above}; \tag{56b}$$

$$\begin{aligned} Q_1 &= k_0, \\ Q_2 &= \frac{5}{6}\left(-3k_0 + 4k_2 - k_3\right), \\ Q_3 &= \frac{50}{27}\left(k_0 - 2k_2 + k_3\right); \end{aligned} \tag{59b}$$

$$\begin{aligned} P_1 &= k_0, \\ P_2 &= \frac{5}{6}\left(-5k_0 + 9k_1 - 4k_2\right), \\ P_3 &= \frac{50}{9}\left(k_0 - 3k_1 + 2k_2\right); \end{aligned} \tag{60b}$$

$$\begin{aligned} N_1 &= k_0, \\ N_2 &= \frac{5}{2}\left(-k_0 + k_1\right). \end{aligned} \tag{61b}$$

For $c = 1$, the third order algorithms $\tilde{y}_{3,5} \equiv \tilde{y}_T$ and $\tilde{y}_{3,6} \equiv \tilde{y}_S$ are upgraded to fourth and fifth orders, respectively, while the orders of the other algorithms remain unchanged. Specifically, we obtain:

$$\tilde{y}_{4,6}(x_0 + h) \equiv \tilde{y}_R(x_0 + h) = y_0 + \frac{1}{630}\left(119\,k_0 + 625\,k_3 - 324\,k_4 + 210\,k_5\right), \tag{57c}$$

$$\begin{aligned} \tilde{y}_{5,6}(x_0 + h) &\equiv \tilde{y}_S(x_0 + h) \equiv \tilde{y}_\Omega(x_0 + h) \equiv \tilde{y}_\omega(x_0 + h) \\ &= y_0 + \frac{1}{7560}\left(728\,k_0 + 3125\,k_2 + 1250\,k_3 + 2187\,k_4 + 270\,k_5\right), \end{aligned} \tag{58c}$$

$$\tilde{y}_{4,5}(x_0 + h) \equiv \tilde{y}_T(x_0 + h) = y_0 + \frac{1}{1890}\left(161\,k_0 + 875\,k_2 + 125\,k_3 + 729\,k_4\right), \tag{56d}$$

$$\tilde{y}_{3,4}(x_0 + h) \equiv \tilde{y}_Q(x_0 + h) = y_0 + \frac{1}{54}\left(19\,k_0 - 20\,k_2 + 55\,k_3\right), \tag{59c}$$

$$\tilde{y}_{2,3}(x_0 + h) \equiv \tilde{y}_P(x_0 + h) = y_0 + \frac{1}{18}\left(43\,k_0 - 165\,k_1 + 140\,k_2\right), \tag{60c}$$

$$\tilde{y}_{2,2}(x_0 + h) \equiv \tilde{y}_N(x_0 + h) = y_0 - \frac{3}{2}\,k_0 + \frac{5}{2}\,k_1, \tag{61c}$$

$$\tilde{y}_1(x_0 + h) = y_0 + k_0. \tag{62b}$$

As far as the estimation of absolute errors in the approximations $\tilde{y}_\Omega$ and $\tilde{y}_\omega$ and their derivatives with respect to x is concerned, the comments made relative to Methods 1 and 2 are valid also for Method 3. Thus, in particular, the error estimation formulae (50), (51), and (52) relative to Method 2, unaltered are enumerated as (63), (64), and (65), respectively, with the understanding that now the algorithms involved relate to Method 3, the counterpart of the algorithms in Method 2. We thus exhibit below these error estimation formulae relative to Method 3 without further details:

$$e(x_0 + h) = \begin{bmatrix} \left|\tilde{y}_\Omega(x_0 + h) - \tilde{y}_T(x_0 + h)\right| \\ \left|\tilde{y}_\omega(x_0 + h) - \tilde{y}_T(x_0 + h)\right| \end{bmatrix} = \left|\tilde{y}_{5,6}(x_0 + h) - \tilde{y}_{4,5}(x_0 + h)\right|, \tag{63}$$

$$e(x_0 + ch) = \begin{cases} \left|\tilde{y}_\Omega(x_0 + ch) - \tilde{y}_T(x_0 + ch)\right|, \\ \left|\tilde{y}_\omega(x_0 + ch) - \tilde{y}_T(x_0 + ch)\right|, \end{cases} \quad c \in (0, 1), \tag{64}$$

$$e'(x_0 + ch) = \begin{cases} \left|(\tilde{y}_\Omega(x_0 + ch))' - (\tilde{y}_T(x_0 + ch))'\right|, \\ \left|(\tilde{y}_\omega(x_0 + ch))' - (\tilde{y}_T(x_0 + ch))'\right|. \end{cases} \tag{65}$$

The comments made previously relative to the array of nine approximate solutions $\left(\tilde{y}_{J'} J = \Omega, \omega, T, S, R, Q, P, N, 1\right)$, which show the trend of the overall approximation process at any point $x = x_0 + ch$, and those comments made relative to alternating series are appropriate for Method 3 as well.

To conclude this section, it should be mentioned that the origins of this research into multiorder embedding of Runge-Kutta continuous processes and the related error estimation rules can be traced to a 1967 presentation [28], of research done by the author in 1966 relative to discrete Runge-Kutta processes.

9. CONTINUOUS APPROXIMATE SOLUTIONS OF EXPLICIT SECOND AND HIGHER ORDER AND IMPLICIT EQUATIONS AND UPGRADING OF APPROXIMATE SOLUTIONS

Consider, to begin with, the initial-value problem involving an explicit second order differential equation [46, p. 84]:

$$\begin{aligned} &y'' = g(x, y, y'), \\ &y(x_0) = y_0, \quad y'(x_0) = y_0'. \end{aligned} \tag{66}$$

We may, as usual, letting $z(x) = y'(x)$ transforms this initial value problem into a system of differential equations of first order:

$$\begin{aligned} &\frac{dy}{dx} = z \equiv f^{(1)}(x, y, z), \\ &\frac{dz}{dx} = g(x, y, z) \equiv f^{(2)}(x, y, z), \\ &y(x_0) = y_0, \quad z(x_0) = z_0 = y_0'. \end{aligned} \tag{67}$$

The exact solution of (67) is a pair of functions: $\{y(x_0 + ch), z(x_0 + ch)\}$.

The main continuous approximate solution is composed of a pair of continuous functions (polynomials in powers of $c = (x - x_0)/h$) : $\{\tilde{y}_\Omega(x_0 + ch),\ z_\Omega(x_0 + ch)\}$, $\tilde{y}_\Omega(x_0 + ch) \equiv \tilde{y}_{4,7}(x_0 + ch)$, $z_\Omega(x_0 + ch) \equiv \tilde{z}_{4,7}(x_0 + ch)$. The Runge-Kutta algorithms corresponding to the present vector case consist of a set of seven pairs of incremental coefficients: $\{k_i^{(1)},\ k_i^{(2)}\}$, $i = 0, \dots, 5, 6$, where $k^{(1)}$ and $k^{(2)}$ relate to the directional functions $f^{(1)}$ and $f^{(2)}$, respectively. We have:

$$\tilde{z}_\Omega(x_0 + ch) \equiv \tilde{z}_{4,7}(x_0 + ch) = z_0 + \sum_{i=1}^{5} {\Omega_i}^{(2)}\, c^i. \tag{68}$$

In consideration of $y'(x) \equiv z(x)$ with $y(x_0) = y_0$, $y'(x_0) = y_0' = z_0$, an upgrading procedure or rule based upon the integration of the polynomial function $\tilde{z}_{r,s}(x_0 + ch)$ is established, which in the present case yields:

$$\tilde{Y}_{5,7}(x_0 + ch) = y_0 + h \left[y_0' c + \sum_{i=1}^{5} {\Omega_i}^{(2)} \frac{c^{i+1}}{i+1} \right], \tag{69}$$

where $\tilde{Y}_{5,7}$ is used instead of $\tilde{y}_{5,7}$ to indicate that this *fifth* order seven stage approximation to $y(x_0 + ch)$ is obtained through the upgrading rule and not through the direct use of a Runge-Kutta formula. In similar ways the other approximate solutions obtained through the use of the other embedded algorithms are upgraded.

Consider now an initial-value problem involving an explicit third order differential equation:

$$\begin{aligned} &y''' = g(x, y, y', y''), \\ &y(x_0) = y_0, \quad y'(x_0) = y_0', \quad y''(x_0) = y_0''. \end{aligned} \tag{70}$$

Letting now $z(x) = y'(x)$ and $u(x) = z'(x) = y''(x)$, Equation (70) is transformed into an initial-value problem involving a system of three differential equations of first order as follows:

$$\begin{aligned}
&\frac{dy}{dx} = z \equiv f^{(1)}(x,y,z,u),\\
&\frac{dz}{dx} = u \equiv f^{(2)}(x,y,z,u),\\
&\frac{du}{dx} = g(x,y,z,u) \equiv f^{(3)}(x,y,z,u),\\
&y(x_0) = y_0, \quad z(x_0) = z_0, \quad u(x_0) = u_0 = y_0''.
\end{aligned} \tag{71}$$

The exact solution of (71) is a triplet of functions: $\{y(x_0+ch),\ z(x_0+ch),\ u(x_0+ch)\}$.

Briefly, we consider

$$\tilde{u}_\Omega(x_0+ch) \equiv \tilde{u}_{4,7}(x_0+ch) = u_0 + \sum_{i=1}^{5} \Omega_i{}^{(3)}\, c^i. \tag{72}$$

Since in the present case $u(x) = z'(x)$, $z(x) = y'(x)$, with the initial conditions $y(x_0) = y_0$, $y'(x_0) = y_0' = z_0$, $y''(x_0) = y_0'' = z_0' = u_0$, after two consecutive integrations of $\tilde{u}_{4,7}(x_0+ch)$, thus with two consecutive upgradings, we obtain:

$$\tilde{Y}_{6,7}(x_0+ch) = y_0 + hy_0'\,c + h^2\left[y_0''\,\frac{c^2}{2} + \sum_{i=1}^{5}\Omega_i{}^{(3)}\,\frac{c^{i+2}}{(i+1)\,(i+2)}\right],$$

a sixth order seven-stage approximation to the exact solution $y(x_0+ch)$.

Had the considered explicit differential equation been, say, of seventh order, we would have obtained $\tilde{Y}_{10,7}(x_0+ch)$, a tenth order seven-stage approximation to $y(x_0+ch)$.

Assume now the differential equation to be in implicit form

$$F(x,\ y,\ y', \dots, y^{(n)}) = 0,$$

and that it cannot be solved for the highest derivative involved, $y^{(n)}$, in closed form. Then using a well known procedure, differentiation of the implicit equation with respect to the independent variable x, one obtains an $(n+1)^{\text{st}}$ order differential equation, which can be put in explicit form, to which the above described continuous approximation processes including the upgrading technique can be applied.

In general, in the case of an explicit differential equation of order n and implicit equations of orders $n-1$, the upgrading rule applied to the main continuous fourth order seven-stage algorithm (of Methods 1, 2 or 3) would yield a continuous approximate solution of orders $n+3$ for all $x = x_0 + ch$, $c \in (0,1]$, that is, for the entire interval of application, including the end of the step, the nodal point $x = x_0 + h$.

For the proof of this upgrading rule and other pertinent information the interested readers are referred to the original article [46, pp. 82–84; Part 2, pp. 84–89].

The continuous approximate solutions generated through this upgrading procedure applied to explicit second and higher order differential equations appear to be competitive from the standpoint of efficiency and effectiveness with those generated through the use of "standard" continuous approximate Runge-Kutta processes. Then the question arises naturally whether it is advantageous to transform through differentiation a first order explicit differential equation, $y' = f(x,y)$, into a second or even higher order explicit differential equation $y^{(n)} = f\left(x,\ y',\ y'', \dots, y^{(n-1)}\right)$ and then apply the upgrading rule to obtain higher order approximate continuous or discrete solutions. It appears again that more research in this direction would be rewarding.

For further support of this conclusion, let us recall that for the generation of scalar and vector Runge-Kutta algorithms of 14th order the derivation of 3259 and 53272 nonlinear algebraic equations are necessary, respectively. Even if this enormous undertaking would be possible thanks to present day supercomputers, still huge systems of equations must be manipulated and solved to determine the values of the unknown constants involved. Quite possibly such a Runge-Kutta process would consist of about 50 stages. Again, even if these systems would be solved and a 14th order Runge-Kutta algorithm developed, which is highly doubtful, such a Runge-Kutta formula would require a volume to be described, and its practicality would be nil. In view of these and several other obstacles, the development of equivalent high order processes based upon the "upgrading" procedure appears to be one of the best, if not the best, alternative.

REFERENCES

1. C. Runge, Über die numerische Auflösung von Differentialgleichungen, *Math. Ann.* **46**, 167–178 (1895).
2. K. Heun, Neue Methoden zur approximativen Integration der Differentialgleichungen einer unabhängigen Veränderlichen, *Z. Math. Phys.* **45**, 23–38 (1900).
3. W. Kutta, Beitrag zur näherungsweisen Integration totaler Differentialgleichungen, *Z. Math. Phys.* **46**, 435–453 (1901).
4. D. Sarafyan, Doctoral Thesis, Dept. of Math., University of Rome, (1960).
5. J.C. Butcher, Coefficients for the study of Runge-Kutta integration processes, *J. Austral. Math. Soc.* **3**, 185–201 (1963).
6. A. Huta, Une amélioration de la methode de Runge-Kutta-Nyström poar la résolution numérique des équations differentielles du premier ordre, *Acta Math. Univ. Comenianae* **1**, 201–224 (1956).
7. A. Huta, Contribution à la formule de sixième ordre daas la methode de Runge-Kutta-Nyström, *Acta Math. Univ. Comenianae* **2**, 21–24 (1957).
8. D. Sarafyan, Improvements in the derivation of Runge-Kutta formulas and computer implementations, Tech. Rept. No. 4, Dept. of Math., Louisiana State University in New Orleans, (1965).
9. D. Sarafyan and R. Brown, Computer derivation of algebraic equations associated with Runge-Kutta formulas, Tech. Rept. No. 11, Dept. of Math., Louisiana State University in New Orleans, (1965); Reissued with an added Appendix in 1966.
10. D. Sarafyan and R. Brown, Computer derivation of algebraic equations associated with Runge-Kutta formulas, *BIT* **7**, 156–162 (1967).
11. E. Fehlberg, Klassische Runge-Kutta-Nyström-Formeln mit Schrittweiten Kontrolle für Differentialgleichungen $\ddot{x} = f(t, x, \dot{x})$, *Computing* **14**, 371–387 (1975); Also issued as Tech. Rept. 432, NASA, (1974).
12. W.E. Milne, *Numerical Solution of Differential Equations*, John Wiley and Sons, (1953).
13. D. Sarafyan, Error estimation for Runge-Kutta methods, *Notices of AMS* **12** (5), 572 (August 1965).
14. D. Sarafyan, Error estimation for Runge-Kutta methods through pseudo-iterative formulas, *Notices of AMS* **13** (2), 224 (Feb. 1966); Issued as Tech. Rept. No. 14, (55 pages), Dept. of Math., Louisiana State University in New Orleans, (May 1966).
15. D. Sarafyan, Numerical solution of ordinary differential equations with pseudo-iterative formulas and estimation of errors, *Intern. Congress of Mathematicians,* (August 1966), Moscow, *Abstracts of Brief Scientific Communications Section* **14**, 17.
16. N. Gylden and B. Einarsson, Classical calculations of inverse bremsstrahlung cross sections in screened potentials, *J. Quant. Spectrose. Radiat. Transfer* **9**, 1117–1131 (1969).
17. D. Sarafyan, Error estimation for Runge-Kutta methods through pseudo-interative formulas, *Rivista di Math.,* Univ. Parma **9**, 1–42 (1968).
18. R. England, Error estimates for Runge-Kutta type solutions to systems of ordinary differential equations, *Comput. J.* **12**, 166–170 (1969).
19. E. Fehlberg, Klassische Runge-Kutta-Formeln fünfer und siebenter Ordnung mit schrittweiten-Kontrolle, *Computing* **4**, 93–106 (1969); Also issued as NASA Tech. Rept. 287, (1968).
20. E. Fehlberg, Klassische Runge-Kutta-Formeln vierter und niedrigerer Ordnung mit schrittweiten-Kontrolle und ihre Anwendung auf wämeleitungsprobleme, *Computing* **6**, 61–71 (1970); Also issued as NASA Tech. Report No. 315 (1969).
21. L. Lapidus and J.H. Seinfeld, *Numerical Solutions of Ordinary Differential Equations*, Academic Press, New York, (1971).
22. R.E. Crosbie and W. Heyes, Variable-step integration methods for simulation application, *Applied Mathematical Modeling* **1**, 137–140 (1976).
23. D. Sarafyan, C. Outlaw and L. Derr, An investigation of Runge-Kutta processes, and equivalence of scalar and vector cases, *J. Math. Analysis and Applic.* **104**, 568–588 (1984).
24. J.C. Butcher, On integration processes of A. Huta, *J. Austral. Math. Soc.* **3**, 202–206 (1963).
25. D. Sarafyan, An investigation concerning fifth order scalar and vector processes, *Rivista di Math.,* Univ. Parma, 41–45 (1971).

26. D. Sarafyan, An investigation concerning the equivalence of scalar and vector sixth order Runge-Kutta processes, *Notices of AMS* **20** (1), A-210 (Jan. 1973).
27. E.B. Shanks, Solution of differential equations by evaluation of functions, *Math. Comp.*, 21–38 (1966).
28. D. Sarafyan, Numerical solution of differential equations through multiorder Runge-Kutta formulas, *Notices of AMS* **14** (3), 373 (April 1967).
29. D. Sarafyan, Sixth order Runge-Kutta formulas possessing error estimation property, *Notices of AMS* **15** (5), 759 (Aug. 1968).
30. D. Sarafyan, Improved sixth order Runge-Kutta formulas and approximate continuous solution of ordinary differential equations, Tech. Rept. No. 40, Dept. of Math., Louisiana State Univ. in New Orleans, (Feb. 1970) (13 pages), and published in, *J. Math. Appl.* **40**, 436–445 (1972).
31. D. Sarafyan, Effective and efficient numerical solution of ordinary differential equations, Tech. Rept. No. 54, Dept. of Math., Louisiana State Univ. in New Orleans, Oct. 1971, and published in *Proceedings of SHARE XXXVIII* **2**, 1-15 (1972).
32. E. Fehlberg, Classical fifth-, sixth, seventh-, and eight-order Runge-Kutta formulas with stepsize control, NASA Tech. Rept. **287** (1968).
33. *Texas Instruments, SR 60 Calculators*, Math. Library, pp. 79–84, (1976).
34. D. Sarafyan, Composite and multi-step Runge-Kutta formulas, Tech. Rept. No. 18, Dept. of Math., Louisiana State Univ. in New Orleans, (Nov. 1966).
35. D. Sarafyan, Multistep Runge-Kutta formulas, *Notices of AMS* **13** (7), 847 (Nov. 1966).
36. R. Abraham, *Modular Software Package, User Manual*, Aerial Press, Dynamic Software, Santa Cruz, CA, (1990).
37. D. Sarafyan, Runge-Kutta approximate continuous solutions for ordinary differential equations, *Notices of AMS* **16** (1), 207 (Jan. 1969).
38. D. Sarafyan, Improved sixth order Runge-Kutta formulas and approximate continuous solution of ordinary differential equations, Tech. Rept. No. 40, 13p., Dept. of Math., Louisiana State Univ. in New Orleans, (Febuary 1970); *J. Math. Anal. Appl.* **40**, 436–445 (1972).
39. W.H. Enright and D. Sarafyan, 7915, *Math. Reviews* **47** (5) (May 1974).
40. R.E. Crosbie and J.L. Hay, Digital techniques for the similation of discontinuities, *Simulation (Summer Conference)*, 87–91 (1978).
41. I. Watanabe, A. Oda, H. Hojo and Nishikawa, Numerical study of RF-plugging of plasma particles in a line-cusp magnetic field, *J. of the Physical Soc. of Japan* **49**, 1988–1994 (1980).
42. M.K. Horn, Fourth-and fifth-order, scaled Runge-Kutta algorithms for treating dense output, *SIAM J. Numer. Anal.* **20**, 558–568 (1983).
43. W.H. Enright, K.R. Jackson, S.P. Norsett and P.G. Thomsen, Interpolants for Runge-Kutta formulas, Rep. 180/85, Dept. of Computer Science, Univ. of Toronto, (1985).
44. W.H. Enright, K.R. Jackson, S.P. Norsett and P.G. Thomsen, Interpolants for Runge-Kutta formulas, *ACM Transactions on Mathematical Software* **12**, 193–218 (Sept. 1986).
45. D. Sarafyan, Continuous approximate solutions of ordinary differential equations and their systems, *Computers Math. Applic.* **10** (2), 139–159 (1984).
46. D. Sarafyan, New algorithms for the continuous approximate solution of ordinary differential equations and the upgrading of the order of the processes, *Computers Math. Applic.* **20** (1), 77–100 (1990).