Neural, Parallel, and Scientific Computations 20 (2012) 437-458

# HIGH-ORDER EXPLICIT RUNGE-KUTTA METHODS USING *m*-SYMMETRY

T. FEAGIN
Department of Computer Science, University of Houston-Clear Lake
2700 Bay Area Blvd., Houston, Texas, 77058, USA
feagin@uhcl.edu

**Abstract:** *The Runge-Kutta equations of condition are reformulated. The concept of m-symmetry is defined. It is shown that any m-symmetric method is of order m. The equations of condition for a twelfth-order explicit Runge-Kutta method with twenty-five stages are solved using m-symmetry. The method contains an embedded tenth-order method that can be used to estimate the local truncation errors and thus to vary the stepsize. Numerical experiments demonstrate that the method compares favorably with other high-order methods, especially for those problems requiring highly accurate solutions.*

**AMS (MOS) Subject Classification:** 65L05. 65L06.
**Key words:** Runge-Kutta methods, equations of condition, high-order, *m*-symmetry, embedded Runge-Kutta methods, initial value problems

## 1. INTRODUCTION

The first-order system of ordinary differential equations:

$$\frac{dx}{dt} = f(t,x) \tag{1.1a}$$

subject to the initial condition:

$$x(t_0) = x_0 \tag{1.1b}$$

has been used to describe and model a wide variety of problems arising in scientific and engineering disciplines. Obtaining the analytical solution of this initial value problem is often difficult, especially when $f(t,x)$ is a nonlinear function of *x*, the dependent variable(s). When analytical solutions are unavailable, numerical methods are typically employed to obtain approximate solutions valid over some range $[t_0, t_f]$. The numerical solution of the initial value problem (1.1) is subject to errors due to truncation, rounding, and (potentially) numerical instability.

An *n*-stage, explicit Runge-Kutta method provides an approximation $x_n$ to $x(t_0+h)$, the solution of (1.1) at the point $t_0+h$, of the form:

$$x_n = x_0 + h\sum_{k=0}^{n-1} c_k f(t_k, x_k) \tag{1.2a}$$

where $t_k \equiv t_0 + \alpha_k h$ and the intermediate approximations $x_k$ are computed from

$$x_k = x_0 + h\sum_{j=0}^{k-1} \beta_{kj} f(t_j, x_j) \tag{1.2b}$$

Received March 13, 2011 

for $k = 1, \ldots, n-1$. The coefficients $\{c_k\}_{k=0}^{n-1}$, $\{\alpha_k\}_{k=0}^{n-1}$, and $\{\{\beta_{kj}\}_{j=0}^{k-1}\}_{k=1}^{n-1}$ are simply real constants. Note that $\alpha_0 = 0$ (as is normally assumed); that is, the first evaluation is $f(t_0, x_0)$. The method is said to be of order $m$ if the local truncation error, i.e. the difference between the approximation $x_n$ and the true solution $x(t_0 + h)$, is $O(h^{m+1})$. The notation used here follows that of Fehlberg, Bettis, and Horn [14, 1, 17].

The form (1.2) is often referred to as the Runge-Kutta ansatz. The value $x_k$ for $k \geq 1$ constitutes an approximation to $x(t_k)$ that is of order $m_k$, which is almost always less than $m$. Making the usual assumption that $x_k$ is at least of order one for all values of $k$ implies that

$$\alpha_k = \sum_{j=0}^{k-1} \beta_{kj} \quad \text{for} \quad k = 0, \ldots, n-1. \tag{1.3}$$

In fact, alternatively, one could simply view these equations as definitions or abbreviations. The $n(n+1)/2$ unknown coefficients for an $n$-stage method are thus simply $\{c_k\}_{k=0}^{n-1}$ and $\{\{\beta_{kj}\}_{j=0}^{k-1}\}_{k=1}^{n-1}$. In the following sections, the $\alpha_k$'s will not be counted as unknowns and the relations (1.3) will not be counted among the equations to be solved. Nevertheless, the $\alpha_k$'s are certainly convenient for both the development and the deployment of Runge-Kutta methods. It should also be noted that the $\alpha_k$'s are normally chosen or determined such that $0 \leq \alpha_k \leq 1$ for all $k$. Otherwise, difficulties could occur for certain problems (1.1) arising from trying to evaluate $f(t, x)$ where it is not even defined.

For an explicit Runge-Kutta method, the number of stages, $n$, is the number of times the derivative function $f(t, x)$ must be evaluated during each step of the integration. Obviously, in order to obtain an approximation to the solution $x(t)$ valid over the entire range $[t_0, t_f]$, a number of steps would normally have to be taken. At each step, the formula (1.2) would be used to provide an approximation to $x(t_0 + h)$; then the values of $t_0$ and $x_0$ would be updated accordingly. The stepsize $h$ might be adjusted at any point based upon any available estimate of the local truncation error. If necessary, for the very last step, $h$ would be reduced in order to stop the integration precisely at $t_f$.

For an $m^{\text{th}}$ order method, the global truncation error over the range $[t_0, t_f]$ is $O(h^m)$ [5, p.154]. The total error at any point $t \in [t_0, t_f]$ includes the accumulated truncation errors, rounding errors, and any errors due to numerical instability. Depending upon the specific initial value problem (1.1), the coefficients of the method (1.2), the value of the stepsize h, and the floating-point representation being used, any of these three components of the error could be the dominant source of error in the computation. Whenever possible, regions where errors due to rounding or numerical instability dominate are to be avoided.

The ultimate goal is to determine a high-order Runge-Kutta method with a reasonably small number of stages so that the overall efficiency of the method will be superior. Verner [28] has shown that a twelfth-order Runge-Kutta method with twenty-nine stages is *possible*.

The subsequent sections contain a description of the development, the properties, and the preliminary testing of an explicit Runge-Kutta method of order twelve that requires only twenty-five stages. Ono [21] has also discovered a twelfth-order method with twenty-five stages, however the method has some large coefficients which can cause a loss of significance and the method does not provide a convenient way of estimating the local truncation error. As will be shown, the method presented herein does not have these problems.

## 2. BACKGROUND

Numerous high-order explicit Runge-Kutta methods [9, 10, 11, 14, 18, 22, 26, 28] have been developed. Most notable is Hairer's tenth-order method with only seventeen stages [15]. A number of enhancements of the basic methods have also been studied including the abilities to vary the order [7, 8, 17], to estimate the local truncation error (often by providing a lower-order embedded formula) [1, 6, 12, 14, 22, 29, 30], and to provide for dense output (i.e., to make accurate, intermediate results available at any point within a given step) [17, 25, 27].

## 3. RUNGE-KUTTA EQUATIONS OF CONDITION

The equations of condition for a Runge-Kutta method are readily available [4, 16]. Shanks [25], for example, provides a recursive formula that can be readily incorporated into a computer program. The great difficulty for the high-order Runge-Kutta methods lies in solving these large systems of nonlinear algebraic equations for the unknown coefficients for a minimum (or near minimum) number of stages. For example, a Runge-Kutta method of order twelve, such as the one developed herein, requires the solution of 7,813 equations. For an n-stage formula there are $n(n+1)/2$ coefficients. The number of new equations for order $m$, $\ell_m$, is exhibited in Table 1 below. Also, the total number of equations, $\eta_m$, is shown. For the eighth-order and lower-order methods, the value shown for the number of stages has been proven to be the minimum possible. For all of these cases, methods with this minimum number of stages have been determined and most of these are in use today. For methods of order greater than nine, the values shown represent the minimum number of stages for known methods of the respective order. A dash in the table indicates an unknown value.

**TABLE 1**

| The order of the formula, $m$ | The number of new equations of order $m$ | The total number of equations for order $m$ | Number of stages $n$ | The number of unknowns |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 2 | 2 | 3 |
| 3 | 2 | 4 | 3 | 6 |
| 4 | 4 | 8 | 4 | 10 |
| 5 | 9 | 17 | 6 | 21 |
| 6 | 20 | 37 | 7 | 28 |
| 7 | 48 | 85 | 9 | 45 |
| 8 | 115 | 200 | 11 | 66 |
| 9 | 286 | 486 | 15 | 120 |
| 10 | 719 | 1205 | 17 | 153 |
| 11 | 1842 | 3047 | - | - |
| 12 | 4766 | 7813 | 25 | 325 |
| 13 | 12486 | 20299 | - | - |
| 14 | 32973 | 53272 | 35 | 630 |
| 15 | 87811 | 141083 | - | - |

It is readily apparent from the table that as the order increases, the number of equations increases exponentially and for $m \geq 6$, surpasses the number of unknowns. For $m = 10$, the ratio of equations to unknowns is almost eight; for $m = 12$, the ratio is more than twenty-four. Clearly, certain correlations or dependencies between these equations have been identified and/or simplifying assumptions have been made in order to reduce the number of equations and to obtain specific solutions.

The most commonly used and well-known assumptions are those due to Butcher [4]. The assumptions are of two types, the so-called column simplifying conditions:

$$\sum_{k=j+1}^{n-1} c_k \alpha_k^i \beta_{kj} + c_j (\alpha_j^{i+1} - 1)/(i+1) = 0 \tag{3.1}$$

which might be assumed for certain values of *i* and *j*, and the so-called row simplifying conditions:

$$\alpha_k^{\mu+1} - (\mu+1) \sum_{j=1}^{k-1} \beta_{kj} \alpha_j^{\mu} = 0 \tag{3.2}$$

which might be assumed for certain values of $\mu$ and *k*. Various combinations of these assumptions have been utilized by a number of investigators in order to reduce the number and complexity of the equations of condition for a given high-order method, thereby greatly facilitating their solution.

The equations of condition sufficient to assure that an explicit *n*-stage Runge-Kutta method (1.2) is of order *m* can be written as [25]:

$$1 - (i+1) \sum_{k=0}^{n-1} c_k q_{k,i,\tau} = 0 \quad \text{for} \quad i = 0, \ldots, m-1 \text{ and } \tau = 1, \ldots, \ell_{i+1} \tag{3.3a}$$

where $\ell_i$ is the number of new equations introduced at the $i^{\text{th}}$-order. Again, the values for $\ell_i$ and $\eta_i$ (the total number of equations of condition up to and including the $i^{\text{th}}$-order) are shown in Table 1 for $i \leq 15$. For $k = 1, \ldots, n-1$, the terms $q_{k,i,\tau}$ are recursively defined from the following relations,

$$q_{k,0,1} \equiv 1 \tag{3.3b}$$

$$q_{k,1,1} \equiv \alpha_k \tag{3.3c}$$

and for $i \geq 2$,

$$q_{k,i,\tau} \equiv \alpha_k q_{k,i-1,\tau} \quad \text{for} \quad \tau = 1, \ldots, \ell_i \tag{3.3d}$$

$$q_{k,i,\tau} \equiv i \sum_{j=1}^{k-1} \beta_{kj} q_{j,i-1,\tau} \quad \text{for} \quad \tau = \ell_i + 1, \ldots, 2\ell_i \tag{3.3e}$$

with the additional relations for $i \geq 4$,

$$q_{k,i,\tau} \equiv q_{k,w,\nu} q_{k,i-w,y} \quad \text{for} \quad \tau = 2\ell_i + 1, \ldots, \ell_{i+1} \tag{3.3f}$$

where $\nu = \ell_w + 1, \ldots \ell_{w+1}$, $y = \ell_{i-w} + 1, \ldots \ell_{i-w+1}$, for $2 \leq w \leq i/2$, and $q_{k,i,\tau} \neq q_{k,i,z}$ for $z < \tau$.

The final restriction on the relation (3.3f) serves to eliminate unnecessary duplication of terms (and consequently, of equations). The restriction is only operative for $m > 7$. In the following

sections, it will sometimes be useful to order the equations of condition given by (3.3) according to the value of $\eta_i + \tau$, where $\eta_0 \equiv 0$.

## 4. REFORMULATION OF THE EQUATIONS OF CONDITION: PHASE ONE

The equations of condition can be rewritten in a number of ways. The reformulation described here simplifies the set of equations (3.3) in a way that provides insight and facilitates their solution. The reformulation occurs in two phases. Define the variables:

(4.1a) $p_{k,i,1} \equiv \alpha_k^i$ for $k = 0,\ldots,n-1,\ i = 1,\ldots,m-1,$

(4.1b) $p_{k,i,\tau} \equiv \alpha_k^i - q_{k,i,\tau}$ for $k = 1,\ldots,n-1,\ i = 2,\ldots,m-1,\ \tau = 2,\ldots,2\ell_i,$ and for $i = 4,\ldots,m-1$

(4.1d) $p_{k,i,\tau} \equiv p_{k,w,\nu}\, p_{k,i-w,y}$ for $\tau = 2\ell_i + 1,\ldots,\ell_{i+1}$ where $\nu = \ell_w + 1,\ldots\ell_{w+1},\ y = \ell_{i-w} + 1,\ldots\ell_{i-w+1},$ for $2 \le w \le i/2,$ and $p_{k,i,\tau} \ne p_{k,i,z}$ for $z < \tau,$

again avoiding unnecessary duplication of terms. The term $p_{k,i,\tau}$ for $\tau = \ell_i + 1,\ldots,2\ell_i$ represents the error in satisfying the $(\tau - \ell_i)^{th}$ of the new $i^{th}$-order equations of condition for the intermediate result $x_k$. In terms of the new variables, $p_{k,i,\tau}$, the equations of condition can be rewritten as:

(4.2a) $$1 - \sum_{k=0}^{n-1} c_k = 0$$

(4.2b) $$1 - (i+1)\sum_{k=1}^{n-1} c_k\, p_{k,i,1} = 0 \quad \text{for} \quad i = 1,\ldots,m-1$$

(4.2c) $$\sum_{k=1}^{n-1} c_k\, p_{k,i,\tau} = 0 \quad \text{for} \quad i = 2,\ldots,m-1 \ \text{and} \ \tau = 2,\ldots,2\ell_i.$$

(4.2d) $$\sum_{k=1}^{n-1} c_k\, p_{k,i,\tau} = 0 \quad \text{for} \quad i = 4,\ldots,m-1 \ \text{and} \ \tau = 2\ell_i + 1,\ldots,\ell_{i+1}$$

The equations (4.2a) and (4.2b) are sometimes referred to as the quadrature equations, because if they are satisfied and $f = f(t)$ only, then the method will be of order *m* (even if the other equations are not satisfied). The equations (4.2d) contain the "cross terms" and are equivalent to the equations arising from (3.3a) and (3.3f) because

$$\sum_{k=1}^{n-1} c_k\, p_{k,w,\nu}\, p_{k,i-w,y} = \sum_{k=1}^{n-1} c_k (\alpha_k^w - q_{k,w,\nu})(\alpha_k^{i-w} - q_{k,i-w,y})$$

$$= \sum_{k=1}^{n-1} c_k (\alpha_k^i - \alpha_k^w q_{k,i-w,y} - \alpha_k^{i-w} q_{k,w,\nu} + q_{k,w,\nu} q_{k,i-w,y})$$

$$= \sum_{k=1}^{n-1} c_k \alpha_k^i - \sum_{k=1}^{n-1} c_k \alpha_k^w q_{k,i-w,y} - \sum_{k=1}^{n-1} c_k \alpha_k^{i-w} q_{k,w,\nu} + \sum_{k=1}^{n-1} c_k q_{k,i,\tau}$$

$$= \frac{1}{i+1} - \frac{1}{i+1} - \frac{1}{i+1} + \sum_{k=1}^{n-1} c_k q_{k,i,\tau} = \sum_{k=1}^{n-1} c_k q_{k,i,\tau} - \frac{1}{i+1}$$

Hence, the satisfaction of (4.2d) implies the satisfaction of (3.3a) when $q_{k,i,\tau}$ derives from (3.3f) and vice versa (provided that all previously occurring equations are satisfied). It should be noted in the above that the two equations:

$$0 = 1 - (i+1)\sum_{k=1}^{n-1} c_k \alpha_k^w q_{k,i-w,y}$$

$$0 = 1 - (i+1)\sum_{k=1}^{n-1} c_k \alpha_k^{i-w} q_{k,w,\nu}$$

are equivalent to the equations

$$0 = \sum_{k=1}^{n-1} c_k p_{k,i,z}$$

$$0 = \sum_{k=1}^{n-1} c_k p_{k,i,\mu}$$

respectively, where $z$ and $\mu$ are less than $\tau$ and therefore have already appeared in the ordered list of equations. The equations (4.2) reveal much about the fundamental nature of the equations of condition and provide subtle clues about how they might be possibly solved.

## 5. REFORMULATION OF THE EQUATIONS OF CONDITION: PHASE TWO

Sometimes the reworked equations (4.2) are useful in their own right. However, recasting the equations into yet another form will prove to be even more useful for helping to solve the equations of condition for high-order explicit Runge-Kutta methods.

Define the variables:

$$(5.1) \qquad r_{ji} \equiv \sum_{k=j+1}^{n-1} c_k \alpha_k^i \beta_{kj} + c_j (\alpha_j^{i+1} - 1)/(i+1) \text{ for } j = 0, \ldots, n-1 \text{ and } i = 0, \ldots, m-3$$

Note that when $r_{ji} = 0$, the corresponding column simplifying assumption (3.1) is true. However, $r_{ji}$ will also prove to be of interest when it is *non*-zero.

For $m > 4$, most of the equations (3.3), when expanded, turn out to be of the form:

$$(5.2) \qquad 1 - (w+i+2)\sum_{k=1}^{n-1} c_k \alpha_k^w (i+1)\sum_{j=1}^{k-1} \beta_{kj} q_{ji\tau} = 0 \text{ with } 0 \le w \le m-3,\ 1 \le i \le m-w-2$$

Any equation of this form can be replaced by the equation:

$$(5.3) \qquad \sum_{j=1}^{n-1} r_{jw} q_{ji\tau} = 0.$$

The replacement is valid and the resulting set of equations is equivalent to the original set (3.3) because

$$\sum_{j=1}^{n-1} r_{jw} q_{ji\tau} = \sum_{j=1}^{n-1} \left( \sum_{k=j+1}^{n-1} c_k \alpha_k^w \beta_{kj} + c_j (\alpha_j^{w+1} - 1)/(w+1) \right) q_{ji\tau}$$

$$= \sum_{k=1}^{n-1} c_k \alpha_k^w \sum_{j=1}^{k-1} \beta_{kj} q_{ji\tau} + \frac{1}{(w+1)} \sum_{j=1}^{n-1} c_j \alpha_j^{w+1} q_{ji\tau} - \frac{1}{(w+1)} \sum_{j=1}^{n-1} c_j q_{ji\tau}$$

$$= \sum_{k=1}^{n-1} c_k \alpha_k^w \sum_{j=1}^{k-1} \beta_{kj} q_{ji\tau} + \frac{1}{(w+1)(w+i+2)} - \frac{1}{(w+1)(i+1)}$$

$$= \sum_{k=1}^{n-1} c_k \alpha_k^w \sum_{j=1}^{k-1} \beta_{kj} q_{ji\tau} - \frac{1}{(w+i+2)(i+1)},$$

which will be zero if and only if the original equation (5.2) is satisfied, provided that the two equations:

$$1-(w+i+2)\sum_{j=1}^{n-1} c_j \alpha_j^{w+1} q_{ji\tau} = 0$$

$$1-(i+1)\sum_{j=1}^{n-1} c_j q_{ji\tau} = 0,$$

which appear before (5.2) in the ordered list of equations, are satisfied. Moreover, any equation replaced by (5.3) either in the original set of equations (3.3) or in the set (4.2), could also be replaced by

$$\sum_{j=1}^{n-1} r_{jw} p_{ji\tau} = 0 \tag{5.4}$$

because

$$\sum_{j=1}^{n-1} r_{jw} p_{ji\tau} = \sum_{j=1}^{n-1} r_{jw} \alpha_j^i - \sum_{j=1}^{n-1} r_{jw} q_{ji\tau} = \sum_{j=1}^{n-1} r_{jw} q_{ji1} - \sum_{j=1}^{n-1} r_{jw} q_{ji\tau} = 0 - 0 = 0,$$

where, obviously from (3.3c) and (3.3d), $q_{ji1} = \alpha_j^i$. Here, the replacement (5.4) is *only* made for $0 \le w \le \lceil (m-4)/2 \rceil$ and $1 \le i \le m-w-2$, so in the resulting equations there are no appearances of $r_{jw}$ such that $w > \lceil (m-4)/2 \rceil$. Consequently, the only equations in (4.2) that are replaced are those generated via (4.2c) when $\ell_{i-w}+1 \le \tau \le 2\ell_{i-w}$ for $0 \le w \le \min(\lceil (m-4)/2 \rceil, i-2)$.

With these replacements, equations (4.2) become:

(5.5a) $1-\sum_{k=0}^{n-1} c_k = 0$

(5.5b) $1-(i+1)\sum_{k=1}^{n-1} c_k p_{k,i,1} = 0 \quad \text{for} \quad i = 1,\ldots,m-1$

(5.5c) For $i = 2,\ldots,m-1$ and $2 \le \tau \le 2\ell_i$, either

(5.5c1) $\sum_{k=1}^{n-1} r_{kw} p_{k,i-w-1,\tau-\ell_{i-w}} = 0$ when $\ell_{i-w}+1 \le \tau \le 2\ell_{i-w}$ where $0 \le w \le \min(\lceil (m-4)/2 \rceil, i-2)$, or

(5.5c2) $\sum_{k=1}^{n-1} c_k p_{k,i,\tau} = 0$ for any other values of $\tau$ in the range

(5.5d) $\sum_{k=1}^{n-1} c_k p_{k,i,\tau} = 0 \quad \text{for} \quad i = 4,\ldots,m-1 \text{ and } \tau = 2\ell_i+1,\ldots,\ell_{i+1}.$

Note that although the *form* of any individual equation in the set of equations (3.3) or (4.2) is not dependent upon *m*, this is not true for the set of equations (5.5) because of (5.5c1). Nevertheless, the respective sets of equations are all equivalent for a given value of $m$. Finally,

it should be noted that the equations generated via (5.5c1) could also be generated, albeit in a different order, by the simpler expression:

$$\text{(5.6)}\quad \sum_{k=1}^{n-1} r_{kw}\, p_{k,i,\tau} = 0 \quad \text{for } w = 0,\ldots,\lceil (m-4)/2 \rceil,\ i = 1,\ldots,m-2-w, \text{ and } \tau = 1,\ldots,\ell_{i+1}$$

## 6. DEFINITION OF *m*-SYMMETRY

The definition given here will simplify the solution process described in the following sections.

**Definition 6.1:** Any Runge-Kutta method (1.2) with coefficients, $\{c_k\}_{k=0}^{n-1}$, $\{\alpha_k\}_{k=0}^{n-1}$, and $\{\{\beta_{kj}\}_{j=0}^{k-1}\}_{k=1}^{n-1}$ will be defined to be ***m*-symmetric** if all of the following are true:

A. The set of integer subscripts $\{k\}_{k=0}^{n-1}$ can be partitioned into three subsets:

1. The non-empty set of integers $Q \equiv \{k_1, k_2, \cdots, k_u\}$ such that the coefficients $\{c_k, \alpha_k\}_{k=k_1}^{k_u}$ are a set of coefficients satisfying $1 - \sum_{k=k_1}^{k_u} c_k = 0$ and $1 - (i+1)\sum_{k=k_1}^{k_u} c_k \alpha_k^i = 0$ for $i = 1,\ldots,m-1$. The $u$ points $\{\alpha_k\}_{k=k_1}^{k_u}$ will be referred to as the *quadrature* points in the following.
2. The $v$ integers $M \equiv \{k_{u+1}, k_{u+2}, \cdots, k_{u+v-1}, k_{u+v}\}$ where $v$ is an even number, $\alpha_{k_{u+1}} = \alpha_{k_{u+2}}, \cdots, \alpha_{k_{u+v-1}} = \alpha_{k_{u+v}}$, $m_{k_{u+1}} = m_{k_{u+2}}, \cdots, m_{k_{u+v-1}} = m_{k_{u+v}}$, and $c_{k_{u+1}} = -c_{k_{u+2}}, \cdots, c_{k_{u+v-1}} = -c_{k_{u+v}}$. The coefficients $\{c_k\}_{k=k_{u+1}}^{k_{u+v}}$ are all non-zero. The points $\{\alpha_k\}_{k=k_{u+1}}^{k_{u+v}}$ will be referred to as the *matching* points.
3. The $n-u-v$ integers $N \equiv \{k_{u+v+1}, \cdots, k_n\}$ such that $c_{k_{u+v+1}} = 0, \cdots, c_{k_n} = 0$. The $n-u-v$ points $\{\alpha_k\}_{k=k_{u+v+1}}^{k_n}$ will be referred to as the *non-matching* points.

B. At any quadrature point $\alpha_k$, $p_{k,i,\tau} = 0$ for $i = 2,\cdots,\lfloor m/2 \rfloor$, $\tau = \ell_i + 1,\ldots,2\ell_i$ and $r_{k,w} = 0$ for $w = 0,\cdots,\lceil (m-4)/2 \rceil$.

C. For any pair of matching points, $\{\alpha_{k_z}, \alpha_{k_{z+1}}\}$,

$r_{k_z,w} = -r_{k_{z+1},w}$ for $w = 0,\cdots,\lceil (m-4)/2 \rceil$ and

$p_{k_z,i,\tau} = p_{k_{z+1},i,\tau}$ for $2 \le i \le m-2-\gamma_{k_z}$, and $2 \le \tau \le \ell_{i+1}$, where

$\gamma_{k_z} \equiv \min(m_{k_z}, \hat{w}_{k_z})$ and $\hat{w}_{k_z}$ is the smallest value of $w$ such that $r_{k_z,w} \neq 0$.

D. At any non-matching point $\alpha_k$, $r_{kw}\, p_{k,i,\tau} = 0$ (i.e., $p_{k,i,\tau} = 0$ or $r_{kw} = 0$) for $w = 0,\cdots,\lceil (m-4)/2 \rceil$, $i \le m-w-2$, and $1 \le \tau \le \ell_{i+1}$.

One consequence of definition (6.1) is that for any pair of matching points $\{\alpha_{k_z}, \alpha_{k_{z+1}}\}$,

(6.2) $p_{k_z,i,\tau}p_{k_z,i',\tau'} = p_{k_{z+1},i,\tau}p_{k_{z+1},i',\tau'}$ where $i+i' \le m-1$,

$1 \le i \le m-2-m_{k_z}$, $1 \le i' \le m-2-m_{k_z}$, $1 \le \tau \le \ell_{i+1}$, and $1 \le \tau' \le \ell_{i'+1}$. If $\tau = 1$ and $\tau' = 1$, then (6.2) becomes $\alpha_{k_z}^{i+i'} = \alpha_{k_{z+1}}^{i+i'}$ which is obviously true. Otherwise and if $i \le m_{k_z}$ or $i' \le m_{k_z}$, (6.2) becomes $0 = 0$. If, on the other hand, $i > m_{k_z}$ and $i' > m_{k_z}$, then $i \le m-1-i' \le m-2-m_{k_z}$ and $i' \le m-1-i \le m-2-m_{k_z}$, so $p_{k_z,i,\tau} = p_{k_{z+1},i,\tau}$ and $p_{k_z,i',\tau'} = p_{k_{z+1},i',\tau'}$ from the definition, and thus (6.2) follows.

## 7. THEOREM

**Theorem 7.1**: Any *m*-symmetric Runge-Kutta method (1.2) is of order *m*.

**Proof**: Given any *m*-symmetric method, the equation (5.5a) is satisfied because according to part A of definition (6.1), $\sum_{k\in Q} c_k = 1$, $\sum_{k\in M} c_k = 0$, and $\sum_{k\in N} c_k = 0$, and thus $\sum_{k=0}^{n-1} c_k = 1$. Similarly, for such a method, (5.5b) is satisfied for $i = 1,\ldots,m-1$ because $\sum_{k\in Q} c_k \alpha_k^i = \frac{1}{i+1}$, $\sum_{k\in M} c_k \alpha_k^i = 0$, and $\sum_{k\in N} c_k \alpha_k^i = 0$, and thus $\sum_{k=1}^{n-1} c_k \alpha_k^i = \frac{1}{i+1}$.

The satisfaction of equation (5.5c1) or the simpler form (5.6) derives from the following, which are true for $1 \le i \le m-2-w$ with $w = 0,\cdots,\lceil (m-4)/2 \rceil$ and $\tau = 1,\cdots,\ell_{i+1}$.

1) $\sum_{k\in Q} r_{kw} p_{ki\tau} = 0$ because $r_{kw} = 0$ for any $k \in Q$ from part B of the definition (6.1).

2) $\sum_{k\in M} r_{kw} p_{ki\tau} = 0$ because $r_{k_z w} p_{k_z i\tau} + r_{k_{z+1} w} p_{k_{z+1} i\tau} = 0$ for a given pair of matching points, $\{\alpha_{k_z}, \alpha_{k_{z+1}}\}$. Clearly, if $r_{k_z w} = 0$, then it's true. If $r_{k_z w} \ne 0$, then $p_{k_z i\tau} = p_{k_{z+1} i\tau}$ from the last part of part C of the definition (6.1) and so $r_{k_z w} p_{k_z i\tau} + r_{k_{z+1} w} p_{k_{z+1} i\tau} = r_{k_z w} p_{k_z i\tau} - r_{k_z w} p_{k_{z+1} i\tau} = 0$.

3) $\sum_{k\in N} r_{kw} p_{ki\tau} = 0$ due to part D of the definition.

The satisfaction of (5.5c2) derives from the following.

1) For $k \in Q$ and $i \le m-1$, if it can be written that $p_{ki\tau} = \alpha_k^\mu p_{ksv} p_{k,i-\mu-s,y}$ (where any appearance of $p_{ki1}$ has been replaced by $\alpha_k^i$) for $\mu \ge 1$, then either $s < m/2$ or $i-\mu-s < m/2$, so either $p_{ksv} = 0$ or $p_{k,i-\mu-s,y} = 0$ which implies $p_{ki\tau} = 0$ and therefore that $\sum_{k\in Q} c_k p_{ki\tau} = 0$.

On the other hand, if $p_{ki\tau} = \alpha_k^\mu p_{k,i-\mu,y}$ when fully expanded, then it must be true that $\mu > (m-4)/2$. Otherwise, this equation would have been incorporated into (5.5c1). Therefore $i-\mu \le m/2$ and consequently $p_{k,i-\mu,y} = 0$. Thus, $p_{ki\tau} = 0$ and $\sum_{k\in Q} c_k p_{ki\tau} = 0$.

2) For a given pair of matching points, $\{\alpha_{k_z}, \alpha_{k_{z+1}}\}$, it suffices to demonstrate that $p_{k_z,i,\tau} = p_{k_{z+1},i,\tau}$ and consequently that $\sum_{k\in M} c_k p_{ki\tau} = 0$. First, if $p_{k_z,i,\tau} = \alpha_{k_z}^{\mu} p_{k_z,s,v} p_{k_z,i-\mu-s,y}$, with $\mu \geq 1$, then $\alpha_{k_z}^{\mu} p_{k_z,s,v} p_{k_z,i-\mu-s,y} = \alpha_{k_{z+1}}^{\mu} p_{k_{z+1},s,v} p_{k_{z+1},i-\mu-s,y}$ from (6.2) and thus $p_{k_z,i,\tau} = p_{k_{z+1},i,\tau}$. On the other hand, if $p_{k_z,i,\tau} = \alpha_{k_z}^{\mu} p_{k_z,i-\mu,y}$, with $\mu \geq 1$, then $\alpha_{k_z}^{\mu} p_{k_z,i-\mu,y} = \alpha_{k_{z+1}}^{\mu} p_{k_{z+1},i-\mu,y}$ using (6.2) with $\tau' = 1$ and $i' = \mu$ and so $p_{k_z,i,\tau} = p_{k_{z+1},i,\tau}$.

3) For $k \in N$, $c_k = 0$, and therefore $\sum_{k\in N} c_k p_{ki\tau} = 0$.

Consequently, equations belonging to either form (5.5c1) or form (5.5c2) are satisfied.

The satisfaction of (5.5d) derives from the following.

1) For $k \in Q$, $i \leq m-1$, and $p_{ki\tau} = p_{ksv} p_{k,i-s,y}$, then either $s < m/2$ or $i - s < m/2$, so either $p_{ksv} = 0$ or $p_{k,i-s,y} = 0$ which implies $p_{ki\tau} = 0$ and therefore that $\sum_{k\in Q} c_k p_{ki\tau} = 0$.

2) For a given pair of matching points, $\{\alpha_{k_z}, \alpha_{k_{z+1}}\}$ with $p_{k_z,i,\tau} = p_{k_z,s,v} p_{k_z,i-s,y}$, it is true that $c_{k_z} p_{k_z,i,\tau} = -c_{k_{z+1}} p_{k_{z+1},i,\tau}$ using (6.2), and therefore $\sum_{k\in M} c_k p_{ki\tau} = 0$.

3) For $k \in N$, $c_k = 0$, and therefore $\sum_{k\in N} c_k p_{ki\tau} = 0$.

Consequently, $\sum_{k=1}^{n-1} c_k p_{ki\tau} = 0$ and all of the equations of condition (5.5) are now satisfied by the given method. □

## 8. SOLVING THE EQUATIONS OF CONDITION USING *m*-SYMMETRY

A solution of the equations of condition for the coefficients of an $m^{\text{th}}$ order explicit Runge-Kutta method can now be accomplished by finding an *m*-symmetric method. Concomitantly, it will be desirable to try to keep the number of stages, $n$, to a minimum.

One approach to determining an *m*-symmetric method is to perform the steps enumerated below. It should be emphasized that other approaches *are* feasible.

1. Determine a quadrature formula of order $m$ or higher with $u$ weights and $u$ nodes $\{c_k, \alpha_k\}_{k=k_1}^{k_u}$. The Gauss-Lobatto formulae are a possible and usually convenient choice, especially for even values of $m$. The nodes $\{\alpha_k\}_{k=k_1}^{k_u}$ comprise the quadrature points. It is highly recommended to select $k_1 = 0$ and, for even values of $m$, to select $k_u = n-1$ (i.e., $\alpha_0$ and $\alpha_{n-1}$ are the first and last, respectively, of the quadrature points). It should be noted that the particular permutation chosen for the values $\{k_2, k_3, \ldots, k_{u-1}\}$ will often have a significant effect upon the performance of the resulting method. The twelfth-order method given in the next section is based on the seven-point Gauss-Lobatto formula with the simple ordering, $0 = \alpha_{k_1} < \alpha_{k_2} < \alpha_{k_3} < \alpha_{k_4} < \alpha_{k_5} < \alpha_{k_6} < \alpha_{k_7} = 1$, with $k_1 = 0$, $k_2 = 12$, $k_3 = 13$, $k_4 = 14$, $k_5 = 15$, $k_6 = 16$, and $k_7 = 24$.

2. Determine (or establish equations governing the values of) the points $\{\alpha_k\}_{k=1}^{k_2-1}$ in such a way that the intermediate order, $m_k$, for $k_2 \le k \le k_u$ will be at least $\lfloor m/2 \rfloor$. When determining these points, it is generally advantageous to try to keep $k_2$ to a minimum which aids in trying to keep $n$ to a minimum, consistent with retaining real values for the coefficients and maintaining $0 \le \alpha_k \le 1$ for all values of *k*. One approach to building up the order at the intermediate points fairly quickly for $m > 5$ is to make the following successive assumptions:

$$\text{(8.1)} \qquad \alpha_2 = \frac{2}{3}\alpha_3, \qquad \alpha_4 = \frac{4\alpha_3 - 3\alpha_5}{6\alpha_3 - 4\alpha_5}\alpha_5,$$

$$\alpha_7 = \frac{20\alpha_5\alpha_6 - 15(\alpha_5+\alpha_6)\alpha_8 + 12\alpha_8^2}{30\alpha_5\alpha_6 - 20(\alpha_5+\alpha_6)\alpha_8 + 15\alpha_8^2}\alpha_8,$$

$$\alpha_{11} = \frac{20\alpha_8\alpha_9\alpha_{10} - 15(\alpha_8\alpha_9+\alpha_8\alpha_{10}+\alpha_9\alpha_{10})\alpha_{12} + 12(\alpha_8+\alpha_9+\alpha_{10})\alpha_{12}^2 - 10\alpha_{12}^3}{30\alpha_8\alpha_9\alpha_{10} - 20(\alpha_8\alpha_9+\alpha_8\alpha_{10}+\alpha_9\alpha_{10})\alpha_{12} + 15(\alpha_8+\alpha_9+\alpha_{10})\alpha_{12}^2 - 12\alpha_{12}^3}\alpha_{12}, \text{ etc.}$$

For $m = 6$ and 7, only the first assumption is made; for $m = 8$ and 9, only the first two assumptions are used; for $m = 10$ and 11, only the first three assumptions are made, etc. These assumptions arise from the use of (3.2) and requiring that $m_1 = 1$, $m_2 = 2$, $m_3 = 3$, $m_4 = 3$, $m_5 = 4$, $m_6 = 4$, $m_7 = 4$, $m_8 = 5$, $m_9 = 5$, $m_{10} = 5$, $m_{11} = 5$, $m_{12} = 6$, etc. while maintaining $\beta_{k,1} = 0$ for $k = 3,\ldots k_{u-1}$; $\beta_{k,2} = 0$ for $k = 5,\ldots k_{u-1}$; $\beta_{k,3} = 0$ and $\beta_{k,4} = 0$ for $k = 8,\ldots k_{u-1}$; $\beta_{k,5} = 0$, $\beta_{k,6} = 0$, and $\beta_{k,7} = 0$ for $k = 12,\ldots k_{u-1}$, etc. For example, in order to have $m_{12} = 6$, given the above sequence of values, the system of equations:

$$\begin{bmatrix} \alpha_8 & \alpha_9 & \alpha_{10} & \alpha_{11} & -\alpha_{12}^2/2 \\ \alpha_8^2 & \alpha_9^2 & \alpha_{10}^2 & \alpha_{11}^2 & -\alpha_{12}^3/3 \\ \alpha_8^3 & \alpha_9^3 & \alpha_{10}^3 & \alpha_{11}^3 & -\alpha_{12}^4/2 \\ \alpha_8^4 & \alpha_9^4 & \alpha_{10}^4 & \alpha_{11}^4 & -\alpha_{12}^5/2 \\ \alpha_8^5 & \alpha_9^5 & \alpha_{10}^5 & \alpha_{11}^5 & -\alpha_{12}^6/2 \end{bmatrix} \begin{bmatrix} \beta_{12,8} \\ \beta_{12,9} \\ \beta_{12,10} \\ \beta_{12,11} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

must be satisfied, which is true iff the determinant of the coefficient matrix vanishes. Given that the $\alpha_k$'s are all distinct for $k = 0, 8, 9, 10,$ and 11, the determinant will vanish iff $\alpha_{11} = \dfrac{20\alpha_8\alpha_9\alpha_{10} - 15(\alpha_8\alpha_9+\alpha_8\alpha_{10}+\alpha_9\alpha_{10})\alpha_{12} + 12(\alpha_8+\alpha_9+\alpha_{10})\alpha_{12}^2 - 10\alpha_{12}^3}{30\alpha_8\alpha_9\alpha_{10} - 20(\alpha_8\alpha_9+\alpha_8\alpha_{10}+\alpha_9\alpha_{10})\alpha_{12} + 15(\alpha_8+\alpha_9+\alpha_{10})\alpha_{12}^2 - 12\alpha_{12}^3}\alpha_{12}$. Similar arguments hold for the other assumptions (8.1) made above.

In addition, the following successive assumptions are often useful for $m > 6$:

$$\text{(8.2)} \qquad (1-\alpha_{n-3}) = \frac{2}{3}(1-\alpha_{n-4}), \qquad (1-\alpha_{n-5}) = \frac{4(1-\alpha_{n-4}) - 3(1-\alpha_{n-6})}{6(1-\alpha_{n-4}) - 4(1-\alpha_{n-6})}(1-\alpha_{n-6}),$$

$$(1-\alpha_{n-8}) = \frac{20(1-\alpha_{n-6})(1-\alpha_{n-7}) - 15((1-\alpha_{n-6})+(1-\alpha_{n-7}))(1-\alpha_{n-9}) + 12(1-\alpha_{n-9})^2}{30(1-\alpha_{n-6})(1-\alpha_{n-7}) - 20((1-\alpha_{n-6})+(1-\alpha_{n-7}))(1-\alpha_{n-9}) + 15(1-\alpha_{n-9})^2}(1-\alpha_{n-9}),$$

etc. These relations arise from the use of (3.1). For $m = 7$ and 8, only the first assumption is made; for $m = 9$ and 10, only the first two assumptions are used; for $m = 11$ and 12, only the

first three assumptions are made; etc. These assumptions allow for $r_{kw} = 0$ as much as is reasonably possible for $k_{u-1} < k < n-1$ where $0 \le w \le \lceil (m-4)/2 \rceil$, thereby simplifying step 6 below.

3. Finally, the matching and non-matching points must be identified. For example (if $\alpha_{n-1} = 1$, as is often the case especially when *m* is even), one might assume:

(8.3) $\alpha_1 = \alpha_{n-2}$, $\alpha_2 = \alpha_{n-3}$, $\alpha_4 = \alpha_{n-4}$, $\alpha_6 = \alpha_{n-5}$, $\alpha_7 = \alpha_{n-6}$, $\alpha_9 = \alpha_{n-7}$, $\alpha_{10} = \alpha_{n-8}$, ...

for the matching points and $\alpha_3$, $\alpha_5$, $\alpha_8$, $\alpha_{11}$, ... for the non-matching points. The precise selection of these points (as to which point should match which other point) often has an effect on how easy it is to solve the resulting combined system of nonlinear equations (8.1), (8.2), and (8.3). Note that, for any non-matching point, $\alpha_k$, it is also usually desirable to require that $k < k_2$ in order to try to minimize the total number of stages.

4. Obtain values for any of the $\alpha_k$'s yet to be determined. If the above approach is taken, obtain these values by solving the nonlinear equations resulting from the assumptions (8.1), (8.2), and (8.3). For the twelfth-order method given in the next section, these assumptions comprise fourteen equations in eighteen unknowns (the seven pairs of matching points and the four non-matching points). The fourteen equations are composed of the first four assumptions from (8.1), the first three from (8.2), and the first seven from (8.3). When solving these equations, it is important to maintain $0 \le \alpha_k \le 1$ for all *k*. One way to solve the equations is as follows:

a. Pick a value for $\alpha_1$ and obtain $\alpha_{23}$ from the first equation in (8.3).

b. Pick a value for $\alpha_5$. Solve the first two equations from (8.1), the first equation from (8.2), and the second and third equations from (8.3), thereby obtaining values for $\alpha_2, \alpha_3, \alpha_4, \alpha_{21}$, and $\alpha_{22}$.

c. Pick a value for $\alpha_8$. Solve the third equation from (8.1), the second from (8.2), and the fourth and fifth from (8.3), thereby obtaining values for $\alpha_6$, $\alpha_7$, $\alpha_{19}$, and $\alpha_{20}$.

d. Pick a value for $\alpha_{11}$. Using the values for $\alpha_{12}$ and $\alpha_{16}$ established in step 1, solve the four remaining equations for $\alpha_9$, $\alpha_{10}$, $\alpha_{17}$, and $\alpha_{18}$.

5. Select non-zero values for the free parameters $\{c_k\}_{k=k_{u+1}}^{k_{u+v}}$ such that $c_{k_{u+1}} = -c_{k_{u+2}}$, ..., $c_{k_{u+v-1}} = -c_{k_{u+v}}$. It should be noted that the particular values selected can influence the properties of the method with respect to truncation errors, rounding errors, and numerical stability. At this point, all of the values $\{c_k, \alpha_k\}_{k=0}^{n-1}$ and some of the vanishing $\beta_{kj}$'s are known.

6. Solve the remaining equations required to make the method *m*-symmetric. For example:

a. Solve $p_{222} = 0$, making $m_2 = 2$ (i.e., make the result at $t_0 + \alpha_2 h$ second-order), if $m \ge 4$. In other words, solve $\alpha_2^2 - 2\beta_{21}\alpha_1 = 0$ for $\beta_{21}$.

b. Solve $p_{k22}=0$, $p_{k33}=0$, and $p_{k34}=0$, making $m_k=3$ for $k=3,4$ (i.e., make the results at $t_0+\alpha_3 h$ and $t_0+\alpha_4 h$ third-order), if $m\ge 6$. In other words, obtain $\beta_{32}$, $\beta_{42}$, and $\beta_{43}$.

c. Continue in this way, making $m_5=4$, $m_6=4$, $m_7=4$, $m_8=5$, $m_9=5$, $m_{10}=5$, $m_{11}=5$, $m_{12}=6$, etc. up to $m_{k_2}=\lfloor m/2\rfloor$.

d. Continue with the remaining quadrature points, making $m_{k_3}=\lfloor m/2\rfloor$, ..., $m_{k_u}=\lfloor m/2\rfloor$.

e. Continue with the remaining points, making $m_{k_{u+1}}=m_{k_{u+2}}$, ..., $m_{k_{u+v-1}}=m_{k_{u+v}}$.

f. For any pair of matching points, $\{\alpha_{k_z},\alpha_{k_{z+1}}\}$, solve any remaining conditions of the form $p_{k_z,i,\tau}=p_{k_{z+1},i,\tau}$ as required by (6.1C) for *m*-symmetry.

g. Impose the conditions on $r_{k,w}$ according to (6.1B), (6.1C), and (6.1D).

7. With the exception of step 4, almost all of the above conditions constitute linear equations or can be made to do so by appropriately choosing special values for some of the $\beta_{kj}$'s. At this point, all of the conditions for *m*-symmetry have been satisfied. If there are any remaining free parameters, they can be used along with any free parameters occurring in step 2 or step 5 above to attempt to improve the method in various ways. Also, consideration of alternate permutations of the ordering of the quadrature points in step 1 can provide additional improvement.

For the twelfth-order method described in the next section, there are at least eighteen free parameters and the method has been improved somewhat by judiciously adjusting the values of a few of these parameters. The free parameters include $\alpha_1$, $\alpha_3$, $\alpha_5$, $\alpha_8$, $\alpha_{11}$, $c_1$, $c_2$, $c_4$, $c_6$, $c_7$, $c_9$, $c_{10}$, $\beta_{11,9}$, $\beta_{11,10}$, $\beta_{17,9}$, $\beta_{19,13}$, $\beta_{24,14}$, and $\beta_{24,17}$. An exhaustive search for the globally optimal values of all of these parameters and of all of the 120 permutations of the five interior quadrature points would have required an enormous computational effort. In fact, simply evaluating the leading terms of the truncation error (i.e., the thirteenth-order terms) for a given set of values of these parameters requires measuring the dissatisfaction of 982 of the 12,486 thirteenth-order equations of condition. Consequently, conducting such a search in a space of dimension eighteen is a daunting task best postponed for a future effort.

The steps given in this section have been tested and successfully employed in obtaining *m*-symmetric methods of orders one, two, three, four, five, six, eight, ten, and (of course) twelve. The number of stages in each case is the minimum known value.

## 9. THE COEFFICIENTS OF AN EXPLICIT RK METHOD OF ORDER TWELVE

Using the procedure outlined in the previous section and encoded in *Mathematica*, an *m*-symmetric Runge-Kutta method of order twelve has been found with twenty-five stages and will be referred to as RK12. The method is based on the seven-point Gauss-Lobatto quadrature formula. There are seven pairs of matching points and four non-matching points.

The Figures 1-4 below are provided in order to offer some insight regarding the method RK12 as it relates to *m*-symmetry. In the chart of Figure 1, the values of the $\alpha_k$'s are depicted. The quadrature points are shown in black. The brackets below the chart indicate the

respective pairings of the matching points, which are depicted in various shades of gray. The non-matching points are shown in white. The $c_k$'s are similarly depicted in Figure 2. Note the symmetry of the values of the $\alpha_k$'s and the anti-symmetry of the $c_k$'s for the matching points in these figures.

**FIGURE 1** ***m*-SYMMETRY IN RK12 – $\alpha_k$ vs $k$**

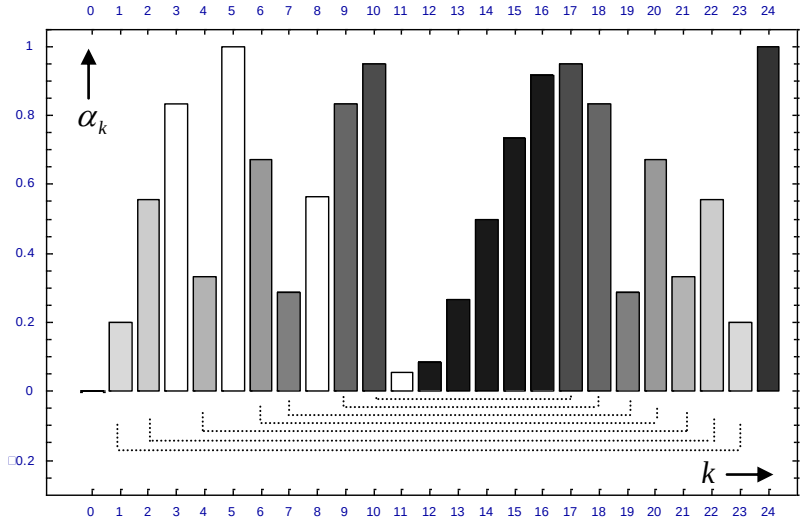



**FIGURE 2** ***m*-SYMMETRY IN RK12 – $c_k$ vs $k$**

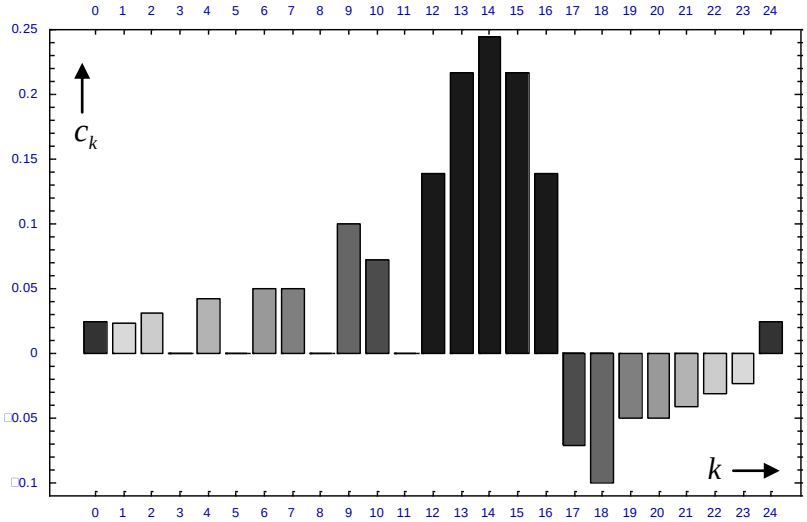



Figure 3 portrays an example of the behavior of the $r_{kw}$'s by showing what occurs when $w = 4$. Figure 4 illustrates the behavior of the $p_{ki\tau}$'s for the example case of $i = 6$ and $\tau = 21$. Note the anti-symmetric characteristics of the $r_{kw}$'s and the symmetric characteristics of the $p_{ki\tau}$'s for the matching points in these figures. Also observe that these variables vanish at the quadrature points. These characteristics are the hallmark of *m*-symmetric methods.

**FIGURE 3** ***m*-SYMMETRY IN RK12 – $r_{k4}$ vs $k$**

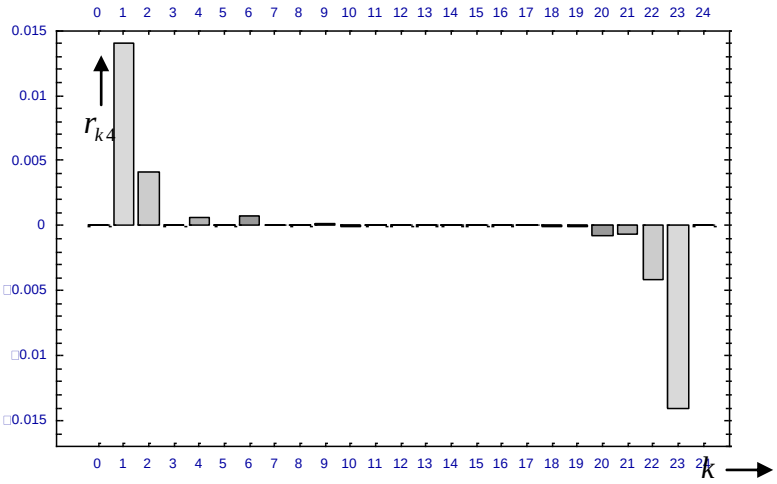
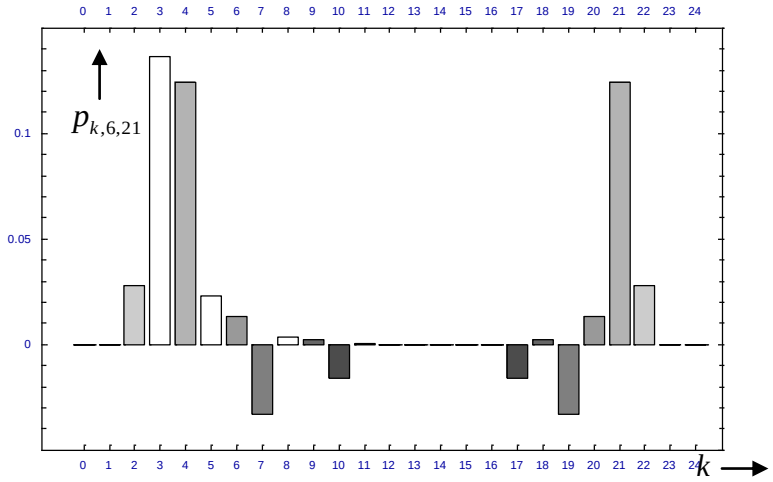



**FIGURE 4** ***m*-SYMMETRY IN RK12 – $p_{k,6,21}$ vs $k$**




The coefficients of RK12, as well as those of some other *m*-symmetric methods, are given to 60 significant digits at the website: http://sce.uhcl.edu/rungekutta/. About 40% of the coefficients are zero. The largest absolute value of any coefficient is less than 12.4. Consequently, the method is well-behaved with respect to rounding errors. The coefficients of the method satisfy not only all of the 7,813 equations for the twelfth-order, but also 11,504 of the 12,486 additional equations for thirteenth-order (more than 92%) and 27,819 of the 32,973 additional equations for fourteenth-order (more than 84%). Consequently, the method promises to be a favorable twelfth-order method with regard to reduced truncation errors.

## 10. NUMERICAL EXPERIMENTS

The twelfth-order method RK12, given in the previous section, has been applied to a number of initial value problems of the form (1.1). Three applications are described in this section. The performance of the method with respect to its ability to obtain an accurate numerical solution over the range $[t_0, t_f]$ for each of these problems has been compared with that of several other methods using efficiency diagrams. For a given method displayed in these diagrams, the

negative logarithm (base 10) of the *absolute value of the largest component of the error* (denoted by *E*) is plotted as a function of the logarithm (base 10) of the *number of evaluations of* $f(t,x)$ *required* (denoted by *NF*). For those problems in which $f(t,x)$ is computationally expensive, the time of computation is essentially proportional to *NF*. On such a diagram, the curve connecting the individual experimental results is usually a nearly straight line with slope equal to the order of the method for those regions where the dominant source of error is truncation error.

Figure 5 exhibits an efficiency diagram for the initial value problem (1.1) where the differential equations represent the equations of motion for the two-body problem:

(10.1a) $$\frac{dx_1}{dt} = x_2\,,\ \frac{dx_2}{dt} = -x_1(x_1^2 + x_3^2)^{-\frac{3}{2}}\,,\ \frac{dx_3}{dt} = x_4\,,\ \frac{dx_4}{dt} = -x_3(x_1^2 + x_3^2)^{-\frac{3}{2}}$$

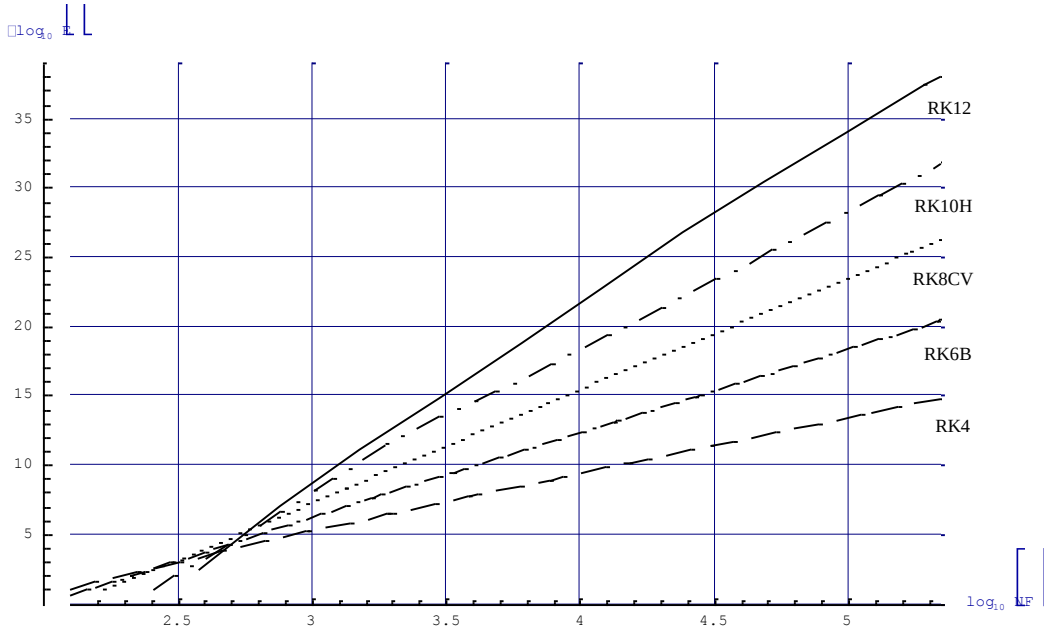
subject to the initial conditions:

(10.1b) $$x_1(0) = 1\,,\ \ x_2(0) = e\,,\ \ x_3(0) = 0\,,\ \ x_4(0) = \sqrt{1-e^2}$$

where the eccentricity $e = 1/5$.

The error is measured at the end of two complete revolutions, i.e. at $t = 4\pi$. In the diagram, the results for RK12 are connected by the solid line, the results for Hairer's tenth-order method RK10H [15] by the dash-dotted line, Cooper and Verner's eighth-order method RK8CV [9] by the dotted line, a sixth-order method due to Butcher RK6B [5, p.177] by the dash-dash-dotted line, and the classical fourth-order Runge-Kutta method RK4 by the dashed line.

**FIGURE 5 EFFICIENCY DIAGRAM – TWO-BODY PROBLEM**




It can be seen from the diagram that as *NF* increases (i.e., as the stepsize decreases), the RK12 method provides the most accurate results for this problem. For low accuracy, other methods provide more efficient approximations.

Figure 6 exhibits an efficiency diagram for the simple harmonic oscillator:

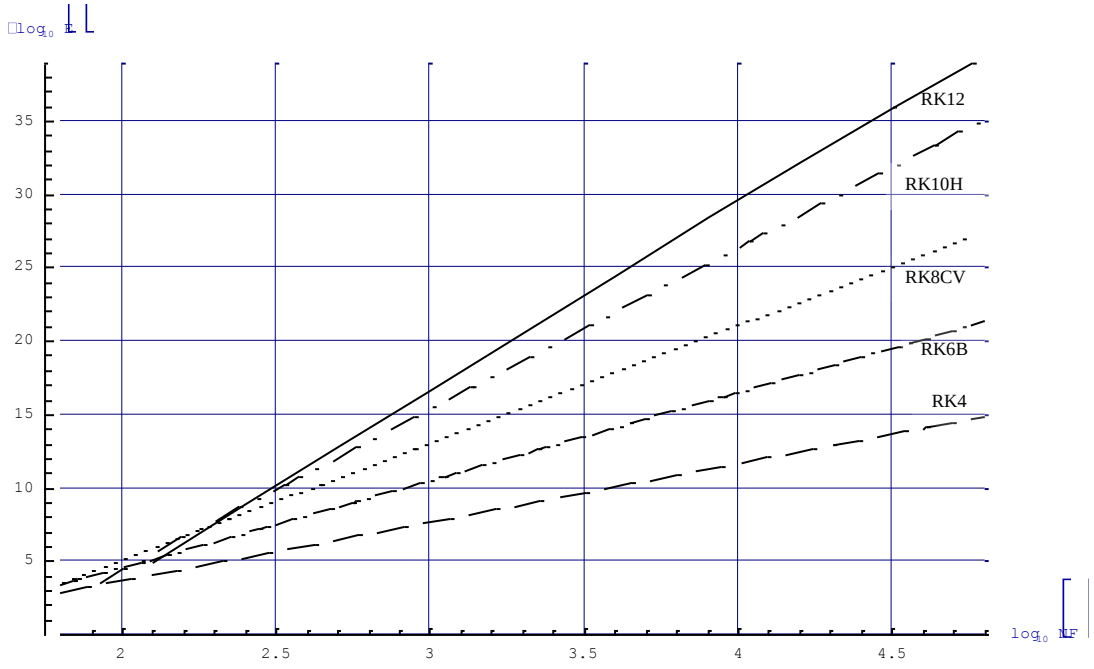
(10.2a) $$\frac{dx_1}{dt} = x_2, \qquad \frac{dx_2}{dt} = -x_1,$$

with the initial conditions:

(10.2b) $$x_1(0) = 0, \qquad x_2(0) = 1.$$

The error is measured at $t = 2\pi$.

**FIGURE 6 EFFICIENCY DIAGRAM - SIMPLE HARMONIC OSCILLATOR**




Again, it can be seen from the diagram that as *NF* increases, the RK12 method (represented by the solid line) provides the most accurate results.

Figure 7 exhibits an efficiency diagram for the Lotka-Volterra equations:

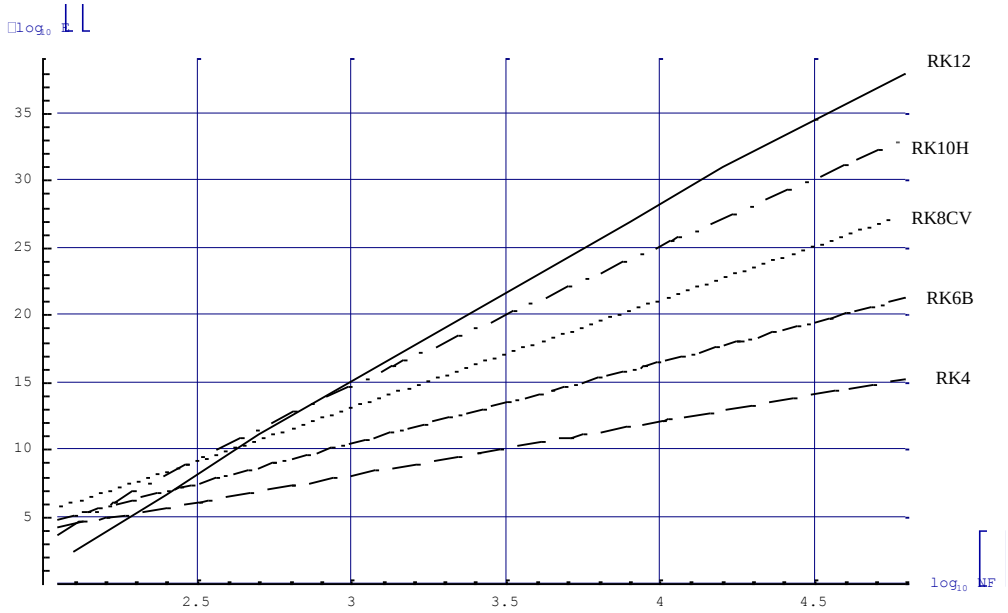
(10.3a) $$\frac{dx_1}{dt} = x_1(2 - x_2), \qquad \frac{dx_2}{dt} = x_2(x_1 - 1),$$

with the initial conditions:

(10.3b) $$x_1(0) = 2, \qquad x_2(0) = 2.$$

The error is measured at $t = 4$.

**FIGURE 7 EFFICIENCY DIAGRAM - LOTKA-VOLTERRA EQUATIONS**




The Lotka-Volterra equations are also known as the predator-prey equations. For this problem, as *NF* increases, the RK12 method again provides the most accurate results.

## 11. THE NUMERICAL STABILITY OF RK12

The twelfth-order Runge-Kutta method RK12, given in section 9, has a reasonably large absolute stability region, very similar to that of the tenth-order method developed by Hairer [15]. The absolute stability region of RK12 in the complex plane is shown in Figure 8 for the scalar ordinary differential equation $x' = \lambda x$. The treatment is sufficient for describing the stability of coupled systems of ordinary differential equations by letting $\lambda$ assume the values of the eigenvalues of the coefficient matrix of the differential equation [20]. The relative stability region is depicted in Figure 9.

## 12. ESTIMATION OF LOCAL TRUNCATION ERRORS

The twelfth-order method, RK12, was developed without consideration for estimating the local truncation errors. However, the method does admit certain embedded, lower-order results without additional evaluations of $f(t,x)$. Let $\hat{x}_n$ represent an approximation to $x(t_0+h)$ of order $\hat{m} < m$ such that

$$\hat{x}_n = x_0 + h\sum_{k=0}^{n-1} \hat{c}_k f(t_k, x_k). \tag{12.1}$$

The coefficients $\{\hat{c}_k\}_{k=0}^{n-1}$ represent new weights for an embedded lower-order method. For RK12, the largest value of $\hat{m}$ that is possible is ten. In other words, any method of order eleven that uses the same values for the coefficients $\{\alpha_k\}_{k=0}^{n-1}$ and $\{\{\beta_{kj}\}_{j=0}^{k-1}\}_{k=1}^{n-1}$ must also have $\hat{c}_k = c_k$ for $k = 0,\ldots,n-1$ and consequently would not be distinct from RK12.

**FIGURE 8 THE ABSOLUTE STABILITY REGION - RK12**

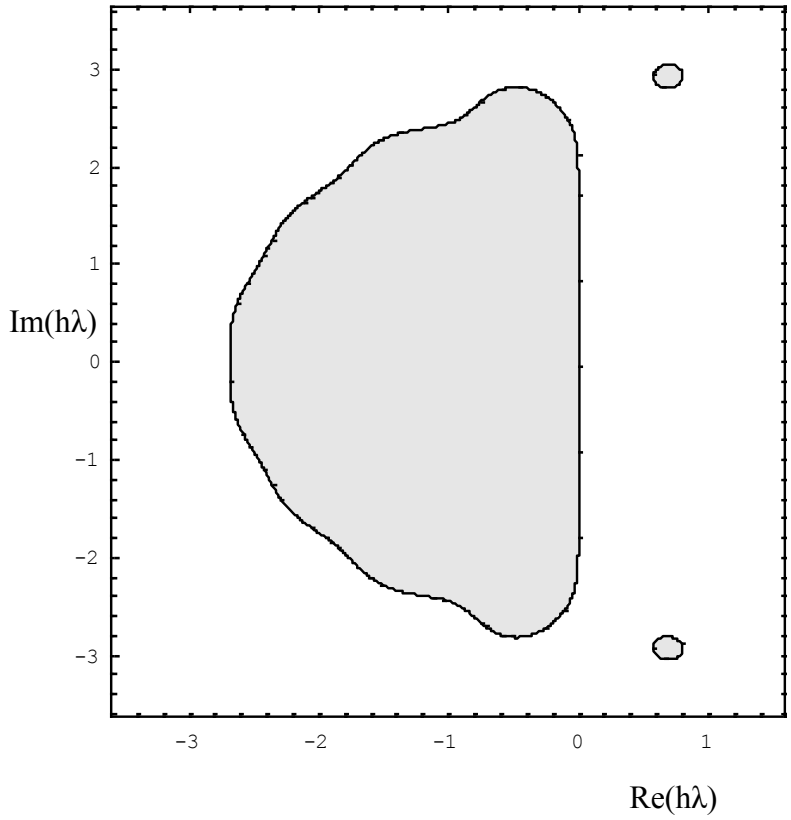
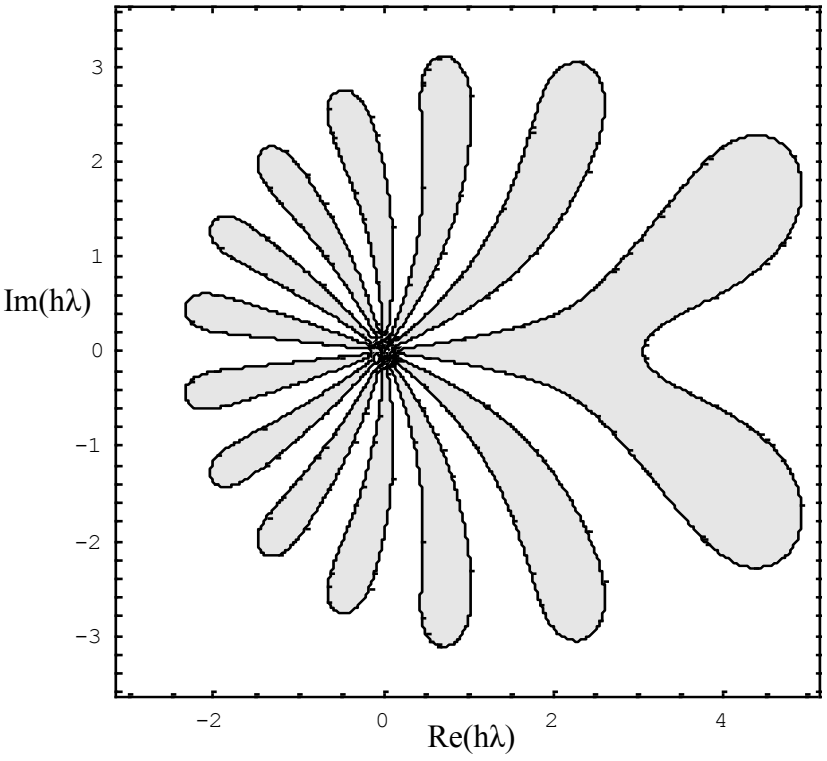



**FIGURE 9 THE RELATIVE STABILITY REGION - RK12**



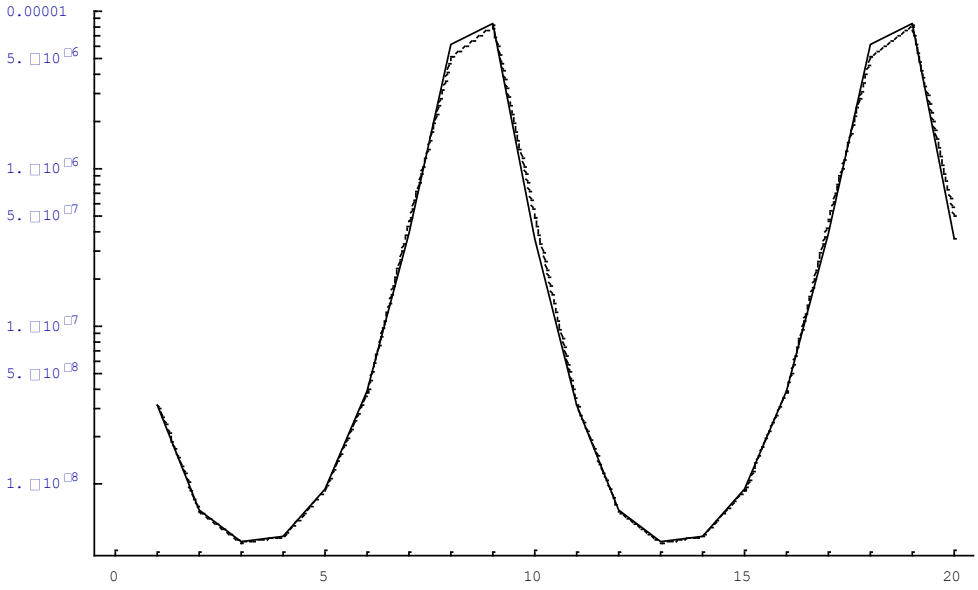
A tenth-order result for $\hat{x}_n$ can be obtained by taking $\hat{c}_k = c_k$ for all values of $k$ except for $k = 1$ and $k = 23$. Choosing, for example, the values $\hat{c}_1 = 1/10$ and $\hat{c}_{23} = -1/10$ (vis à vis $c_1 = 3/128$ and $c_{23} = -3/128$ for RK12) causes $\hat{x}_n$ to be a tenth-order result. The difference

$$\hat{x}_n - x_n = \frac{49}{640} h\big(f(t_1, x_1) - f(t_{23}, x_{23})\big) \tag{12.2}$$

can then be used as an estimate of the local truncation error of the tenth-order result, which in turn could be used to adjust the stepsize of the numerical integration and hopefully provide for a more efficient algorithm overall. It should be noted, however, that the difference $\hat{x}_n - x_n$ vanishes for problems in which $f = f(t)$ only (i.e., for quadrature problems). Consequently, the error estimate must be used with caution.

As it turns out, the tenth-order result described above satisfies all but forty-eight of the 1,842 additional eleventh-order equations. The estimate of the local truncation error (12.2) is a very good estimate as is illustrated by the following experiment. The *actual* and *estimated* local truncation errors for the embedded tenth-order method have been computed at each step of the numerical solution of the initial value problem (10.1). In Figure 10, the maximum absolute value of the four components of these errors is plotted (on a log scale) as a function of the number of steps taken for a given stepsize ($h = \pi/5 \cong 0.628319$). The solid line depicts the actual error and the dashed line depicts the estimate of the error. As one might expect, the estimate tracks the

**FIGURE 10 ACTUAL AND ESTIMATED LOCAL TRUNCATION ERRORS AT EACH STEP FOR EMBEDDED TENTH ORDER METHOD**




true error very well, even though the range of the error extends over several orders of magnitude. Experiments at other stepsizes for this problem have shown that the estimate

(12.2) is always within a few percent of the actual error. Of course, the estimate asymptotically conforms with the actual error as $h \to 0$.

Consequently, when applying the RK12 method, the associated estimate (12.2) of the local truncation error of the embedded result could be used to monitor the accuracy of the numerical integration and to control the stepsize, thus providing for a more reliable and efficient algorithm overall. Of course, the twelfth-order result would be used to propagate the solution to the next step.

## 13. CONCLUSIONS

The concept of *m*-symmetry and its use in solving the equations of condition for high-order explicit Runge-Kutta methods has been demonstrated. The equations of condition have been solved and the resulting coefficients have been exhibited for a particular twelfth-order method. Preliminary numerical experiments indicate that the new twelfth-order method, RK12, seems to be a reasonably stable, effective, and efficient method, particularly for those problems where highly accurate solutions are desired. In addition, an embedded tenth-order result has been identified that could be used to provide an estimate of the local truncation error, thereby permitting the accuracy to be monitored and the stepsize of the numerical integration to be varied in a meaningful way.

## ACKNOWLEDGEMENTS

The author gratefully acknowledges the guidance and encouragement received over the years from Dr. Dale G. Bettis and Dr. Erwin Fehlberg. The author is especially indebted to Dr. Mike Mezzino for his moral support and encouragement regarding the development of the twelfth-order method described herein.

## REFERENCES

[1] D.G. BETTIS, *Efficient embedded Runge-Kutta methods,* in *Numerical Treatment of Differential Equations*: *Proceedings Oberwolfach, 1976*, *Lecture Notes in Mathematics, No. 631,* R. Bulirsch, Ed., Springer, Berlin (1978), pp. 9-18.

[2] J.C. BUTCHER, *Coefficients for the study of Runge-Kutta integration processes*, J. Aust. Math. Soc., 3 (1963), pp. 185-201.

[3] J.C. BUTCHER, *On Runge-Kutta processes of high order*, J. Aust. Math. Soc., 4 (1964), pp. 179-194

[4] J.C. BUTCHER, *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods*, Wiley and Sons, New York, (1987).

[5] J.C. BUTCHER, *Numerical Methods for Ordinary Differential Equations*, Wiley and Sons, New York, (2003), pp.123-196.

[6] M. CALVO, J.I. MONTIJANO, L. RÁNDEZ, *A new embedded pair of Runge-Kutta formulas or orders 5 and 6*, Computers Math. Applic., 20 (1990), pp. 15-24.

[7] J.R. CASH, *Block Runge-Kutta Methods for the Numerical Integration of Initial Value Problems in Ordinary Differential Equations Part I. The Nonstiff Case*, Math. Comp. 40 (1983), pp. 175-191.

[8] J.R. CASH AND ALAN H. KARP, *A variable order Runge-Kutta method for initial value problems with rapidly varying right-hand sides*, ACM Trans. Math. Software, 16, Issue 3 (1990), pp. 201-222.

[9] G.J. COOPER AND J.H. VERNER, *Some explicit Runge-Kutta methods of high order*, SIAM J. Numer. Anal., 9 (1972), pp. 389-405.

[10] A.R. CURTIS, *An eighth order Runge-Kutta process with eleven function evaluations per step*, Numer. Math., 16 (1970), pp. 268-277.

[11] A.R. CURTIS, *High-order explicit Runge-Kutta formulae, their uses, and limitations*, J. Inst. Maths. Applic., 16 (1975), pp. 35-55.

[12] R. ENGLAND, *Error estimates for Runge-Kutta type solutions to systems of ordinary differential equations*, Comp. Jour., 12 (1969), pp. 166-170.

[13] L. EULER, *De integratione aequationum differentialium per approximationem*, in Opera Omnia, 1st series, Vol. 11, Institutiones Calculi Integralis, Teubner, Leipzig and Berlin, (1913), pp.424-234.

[14] E. FEHLBERG, *Classical Fifth-, Sixth-, Seventh- , and Eighth-Order Runge-Kutta Formulas with Stepsize Control*, NASA TR R-287, (1968).

[15] E. HAIRER, *A Runge-Kutta method of order 10*, J. Inst. Math. Applics. 21 (1978), pp. 47-59.

[16] E. HAIRER, S.P. NØRSETT, AND G. WANNER, *Solving Ordinary Differential Equations I: Nonstiff Problems*, 2nd ed., Springer-Verlag, Berlin (1993).

[17] M.K. HORN, *Fourth- and fifth-order, scaled Runge-Kutta algorithms for treating dense output*, SIAM J. Numer. Anal. 20 (1983), pp. 558-568.

[18] A. HUŤA, *Contribution à la formule de sixième ordre dans la méthode de Runge-Kutta-Nyström*, Acta Fac. Nat. Univ. Comenian. Math., 2 (1957), pp. 21-24.

[19] W. KUTTA, *Beitrag zur näherungsweisen Integration total Differentialgleichungen*, ZAMP 46, (1901) pp. 435-453.

[20] LEON LAPIDUS AND JOHN H. SEINFELD, *Numerical Solution of Ordinary Differential Equations*, Academic Press, New York (1971), p. 129.

[21] HIROSHI ONO, *On the 25 stage 12th order explicit Runge--Kutta method*, JSIAM Journal, Vol. 16, No. 3, 2006, p. 177-186.

[22] P.J. PRINCE AND J.R. DORMAND, *High-order embedded Runge-Kutta formulae*, J Comput. Appl. Math., 7 (1981), pp. 67-76.

[23] C. RUNGE, *Über die numerische Auflösung von Differentialgleichungen*, Math. Ann., 46 (1895), pp. 167-178.

[24] C. RUNGE, *Über die numerische Auflösung von Differentialgleichungen*, Nachr. Gesel. Wiss., (1905), pp. 252-257.

[25] LAWRENCE F. SHAMPINE, *Interpolation for Runge-Kutta Methods*, SIAM J. Numer. Anal. 22 (1985), pp. 1014-1027.

[26] E. BAYLIS SHANKS, *Solutions of Differential Equations by Evaluations of Functions*, Math. Comp. 20, No. 93 (1966), pp. 21-38.

[27] P.W. SHARP AND J.H. VERNER, *Generation of high-order interpolants for explicit Runge-Kutta pairs*, ACM Trans. Math. Software, 24, Issue 1 (1998), pp. 13-29.

[28] J.H. VERNER, *The derivation of high order Runge Kutta methods,* Univ. of Auckland, New Zealand, Report No. 93, (1976), 27 pages.

[29] J.H. VERNER, *Explicit Runge-Kutta methods with estimates of the local truncation error*, SIAM J. Numer. Anal., (1978), pp. 772-790.

[30] J.H. VERNER AND P.W. SHARP, *Completely imbedded Runge-Kutta formula pairs,* Dept. Mathematics and Statistics, Queen's University, Kingston, Preprint 1991-01, 24 pages.