DFVLR Deutsche Forschungs- und
Versuchsanstalt
für Luft- und Raumfahrt

Forschungsbericht

Scaled Runge-Kutta Algorithms for Handling Dense Output

Mary Kathleen Horn

DFVLR
Institut für Dynamik der Flugsysteme
Oberpfaffenhofen

43 Seiten
14 Tabellen
4 Literaturstellen

DFVLR-FB 81-13

DM 10,10

Manuskript eingereicht am 2. April 1981

Scaled Runge-Kutta Algorithms for Handling Dense Output

Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt
Forschungsbereich Flugmechanik/Flugführung
Institut für Dynamik der Flugsysteme
Abteilung Lenkung
Oberpfaffenhofen, D-8031 Weßling

Oberpfaffenhofen, 11. März 1981

Forschungsbereichsleiter:
Prof. Dr.-Ing. F. Thomas

Institutsleiter:
Dr.-Ing. J. Ackermann

Verfasser:
M.K. Horn, M.Sc., Ph.D.

Abteilungsleiter:
Dipl.-Ing. K.H. Well, Ph.D.

Runge-Kutta Methods, Numerical Analysis, Ordinary Differential Equations

Scaled Runge-Kutta Algorithms for Handling Dense Output

Summary

New, low order Runge-Kutta algorithms are developed which determine the solution of a system of ordinary differential equations at any point within a given integration step, as well as at the end of each step. The new, scaled Runge-Kutta methods are designed to be used with existing Runge-Kutta formulas, using the derivative evaluations of these defining algorithms as the core of the new system. Thus, for only a slight increase in computing time, the solution may be generated within the integration step, improving the efficiency of the Runge-Kutta algorithms, since the step length need no longer be severely reduced to coinside with the desired output point. Scaled Runge-Kutta algorithms are presented for orders 3 through 5, along with accuracy comparisons between the defining algorithms and their scaled versions for a test problem.

Runge-Kutta-Verfahren, Numerische Mathematik, Gewöhnliche Differentialgleichungen

Skalierte Runge-Kutta-Formeln für dichte Lösungsausgabe

Übersicht

Neue skalierte Runge-Kutta-Formeln dritter bis fünfter Ordnung, die die Lösung eines Systems von Differentialgleichungen an bestimmten Punkten innerhalb und am Ende eines jeden Integrationsschrittes bestimmen können, werden hergeleitet. Die neuen skalierten Algorithmen werden unter Verwendung der klassischen Runge-Kutta-Formeln entwickelt und benützen deren Schema von Funktionsauswertungen. Auf diese Weise kann die Lösung innerhalb des Schrittes mit wenig mehr Rechenzeit bestimmt werden, so daß die Leistungsfähigkeit der Runge-Kutta-Methoden in hohem Grade verbessert wird. Die Genauigkeit der Lösungen wird mit derjenigen der klassischen Formeln an Hand eines Beispieles verglichen.

Table of Contents

1. Introduction

The Runge-Kutta (RK) algorithm, designed to treat the initial value problem

$$(1.1) \qquad \frac{dy}{dt} = f(t,y) \quad , \quad y(t_0) = y_0$$

where y is an n-vector, is an explicit, single-step algorithm which uses a linear combination of derivative evaluations to approximate a truncated Taylor series. The high accuracy and the simplicity of the algorithm lends itself readily to the solution of many classes of ordinary differential equations (ODEs). To perform efficiently, however an RK method must operate with an optimal step size, consistent with the specified truncation error criteria. If frequent data output is requested, e.g., in iterative schemes to determine the perigee or apogee of an orbit, this efficiency may be reduced significantly. This difficulty arises because the Runge-Kutta algorithm generates the solution of the ODE only at the end point of the integration step, thus requiring step size reductions to output data at specified values of the independent variable. Shampine, Watts, and Davenport [4], however, have shown that such adjustments to the step size seriously impair the efficiency of the method. The Runge-Kutta algorithm must operate using near optimal step sizes if the method is to be competitive with other numerical methods for solving ODEs, and, thus, an alternate technique for treating the problem of frequent data output is essential.

Scaled, Runge-Kutta algorithms may be developed which determine the solution of the ODE at any point $t^* = t_0 + \sigma h$, once the solution has been evaluated at $t_1 = t_0 + h$, where the scaling factor, σ, generally ranges between 0 and 1. By using the derivative evaluations generated to obtain $y(t_1)$, this new scaled solution may be determined at a cost of, at most, only a few additional derivative evaluations. Thus, for a given RK algorithm, a family of scaled algorithms may be developed, which, when used in conjunction with this defining algorithm, gives the solution at any point within the interval of integration.

The derivation of the scaled algorithm is given for orders three through five, being applied to coefficients of defining algorithms develped by E. Fehlberg [1] and D.G. Bettis (private communication). Numerical results, comparing the accuracy of the scaled solution to the accuracy of the defining algorithm, are presented.

2. The Scaled Runge-Kutta Algorithm

The Runge-Kutta algorithms of order p, used to solve Eqn. (1.1), assumes the form

$$(2.1.a)\quad y = y_0 + h \sum_{j=0}^{r} C_j f_j$$

where

$$(2.1.b)\quad f_0 = f(t_0, y_0)$$

and

$$(2.1.c)\quad f_j = f(t_0 + \alpha_j h,\ y_0 + h \sum_{k=0}^{j-1} \beta_{j,k} f_k)$$

for $j = 1,2,\ldots,r$, where the α, β, and C coefficients have been selected so that the algorithmic solution, y, is equivalent to a Taylor sum of order p. Specifically, a term by term comparison of the Taylor series expansion of the algorithmic solution to the Taylor series expansion of the solution vector results in certain truncation error coefficients, $TEC_{i,j}$, i.e., expressions in α, β, and C for each order, i, with $j = 1,2,\ldots,\lambda_i$ where λ_i increases with increasing order. For an algorithm to be of order p all $TEC_{i,j}$ must vanish for $i = 1,2,\ldots,p$; $j = 1,2,\ldots,\lambda_i$. These vanishing truncation error coefficients are referred to as equations of condition. (See Table 1.)

The $\beta_{j,0}$ may be determined from the relations

$$(2.2) \qquad \sum_{k=0}^{j-1} \beta_{j,k} = \alpha_j$$

for $j = 1, 2, \ldots, r$.

The form of the scaled algorithm parallels that of the defining algorithm since the efficiency of the new methods requires the use of the derivative evaluations (2.1.b) and (2.1.c) as the core of the new system. One assumes that the solution has been advanced from t_0 to $t_1 = t_0 + h$, i.e., that f_1, $f_2, \ldots, f_r$ have been determined, and that the solution is required at some intermediate point, $t^* = t_0 + h^* = t_0 + \sigma h$, with σ a scaling factor, where generally $\sigma \in (0,1)$. More specifically,

$$(2.3) \qquad h^* = \sigma h \quad .$$

The scaled algorithm assumes the basic form of the defining algorithm

$$(2.4.a) \quad y(t_0 + h^*) = y_1^* = y_0 + h^* \sum_{j=0}^{r^*} C_j^* f_j^*$$

where

$$(2.4.b) \quad f_0^* = f(t_0, y_0)$$

and

$$(2.4.c) \quad f_j^* = f(t_0 + \alpha_j^* h^*, y_0 + h^* \sum_{k=0}^{j-1} \beta_{j,k}^* f_k^*)$$

for $j = 1, 2, \ldots, r^*$, where $r^* \geq r$. If α^* and β^* are chosen so that

$$(2.5) \qquad f_j^* = f_j$$

for $j = 1, 2, \ldots, r$, then the major expense of applying the algorithm has been avoided since f_1, $f_2, \ldots, f_r$ have already been determined during the evaluation of $y(t_1)$. Thus, only $f_{r+1}, \ldots, f_{r^*}$ need to be evaluated. Comparing eqn. (2.1.b) and (2.1.c) with (2.4.b) and (2.4.c), respectively, gives $f_j^* = f_j$ if

(2.6.a) $\alpha^*_j = \alpha_j h/h^* = \alpha_j/\sigma$

and

(2.6.b) $\beta^*_{j,k} = \beta_{j,k} h/h^* = \beta_{j,k}/\sigma$

for $j = 1,2,\ldots,r$; $k = 0,1,2,\ldots,j-1$. Thus, many of the coefficients of the scaled algorithm are predetermined, being related to their corresponding coefficients in the defing algorithm by a factor of $1/\sigma$.

The solution of the scaled algorithm depends upon the assumptions imposed during the derivation of the defining algorithm, i.e., these assumptions must generally hold in the scaled system in order for the new system to satisfy the same equations of condition. The most general assumptions imposed during the development of the low order RK methods concern the terms

$$(2.7) \qquad p_{j,i} = \sum_{k=1}^{j-1} \beta_{j,k}\alpha_k^i \quad ,$$

since setting

$$(2.8) \qquad p_{j,i} = \alpha_j^{i+1}/(i+1)$$

for $i = 1,2$, $j = 2,3,\ldots,r$, causes a number of equations of condition to become redundant. Clearly, after substituting the scaled parameters, eqn. (2.8) is also valid in the scaled system. An important consideration is the analysis of the higher order truncation error coefficients and the stability properties of the algorithm which are the major criteria used in the selection of the coefficients for the defining algorithm. In general, during the solution of the equations of condition, several α, β, or C parameters remain as free parameters which may be selected to reduce the TEC of the next higher order improving the accuracy of the method. Theses same parameters govern the stability properties of the method, i.e., they determine the error amplification or damping due to the algorithm. For the linearized equation, $y' = \lambda y$, the error amplification factor, $P(\lambda h)$, is determined by the polynomial

$$(2.9) \qquad Z(\lambda h) = \sum_{k=0}^{order} \frac{(\lambda h)^k}{k!} + \sum_{k=order+1}^{r+1} v_k (\lambda h)^k$$

where the v_k terms are expressions in α, β, and c, and where h is the step size and λ is an eigenvalue of the Jacobian matrix, $(\partial f/\partial y)$, evaluated at $t = t_0$. $|Z(\lambda h)| = 1$ defines the stability boundary for a given RK algorithm. Remembering that h will be scaled by σ, the TEC terms and the stability polynomial will also be affected significantly by this scaling parameter.

The expense of applying the scaled algorithm is determined by the number of additional derivative evaluations required. For the third order method studied, no additional derivative evaluations are needed, i.e., the solution may be obtained at any intermediate value of the independent variable at a cost of only a few arithmetic operations used in generating the C* coefficients. Using one additional derivative evaluation, a fourth order solution may be determined anywhere within the integration step. After this additional evaluation has been generated, the solution may be computed at further values of t at a cost of a few arithmetic operations. The complexity of the solution of the fifth order, scaled algorithm dictates the necessity of two forms of such algorithms. For two additional derivative evaluations, a fifth order solution may be determined at any point within the integration step (Horn, [2]). Each further solution, however, requires two additional derivative evaluations. Thus, for dense output within a given step, a second form of the scaled, fifth order solution is needed. Using five additional derivative evaluations, a fifth order solution may be generated within the integration step with the added advantage that further solutions may be determined with no additional derivative evaluation expense. Such a formulation is of particular importance in iterative schemes involving an RK integrator.

3. A Third Order, Scaled Runge-Kutta Algorithm

Using a third order, Runge-Kutta algorithm (due to D.G. Bettis, see Table 2.a), having an embedded second order solution for step size control, a scaled, third order algorithm may be developed having comparatively low truncation error coefficients throughout the interval as well as reasonable stability properties. Since the defining algorithm, RKT(2)3, has the advantage that the final derivative evaluation for one step equals the initial evaluation for the subsequent step, third order solutions may be evaluated at any value of the independent variable at a cost of three evaluations per integration step (plus the arithmetic operations required for the C* coefficients), unless the solution is to be advanced from a scaled solution evaluation, in which case one additional derivative evaluation is needed.

The equations of condition for the third order scaled algorithm are

$$C_0^*\alpha_0^j + C_1^*\alpha_1^j + C_2^*\alpha_2^j + C_3^*\alpha_3^j = \sigma^{j+1}/(j+1) \tag{3.1}$$

for $j = 0,1,2$, where $\alpha_0 = 0$ and

$$C_2^*\beta_{2,1}\alpha_1 + C_3^*(\beta_{3,1}\alpha_1 + \beta_{3,2}\alpha_2) = \sigma^3/6 \quad , \tag{3.2}$$

where the α_j and $\beta_{j,k}$ coefficients are from the defining algorithm (Table 2.a.), while the C* coefficients are the weighting factors in the scaled algorithm. These same equations determine the coefficients of the defining algorithm, except that $C_3 = 0$ and the $\beta_{3,k}$ coefficients are defined by $\beta_{3,k} = C_k$, where the C_k's may be written solely in terms of the α coefficients. The C* coefficients of the scaled algorithm may then be written:

$$C_1^* = \frac{-\sigma(2\sigma-3)(2-3\alpha_2)}{6\alpha_1(\alpha_1-\alpha_2)} = -\sigma(2\sigma-3)/3 \tag{3.3}$$

$$C_2^* = \frac{-\sigma(2\sigma-3)(2-3\alpha_1)}{6\alpha_2(\alpha_2-\alpha_1)} = -4\sigma(2\sigma-3)/9 \tag{3.4}$$

and

(3.5) $C^*_3 = \sigma^2 - \sigma$

with

(3.6) $C^*_0 = 1 - (C^*_1 + C^*_2 + C^*_3) = (9 + \sigma(-12 + 5\sigma))/9$.

Table 2.b presents values of G^*_4, the Euclidean norm of the fourth order truncation error coefficients,

(3.7) $C^*_i = \left(\sum_{k=1}^{\lambda_i} T^{*2}_{i,k} \right)^{1/2}$,

where the order of the TEC terms $i = 4$ and where the number of these terms for order $i, \lambda_i = 4$, with the $T_{i,k}$ coefficients (given in Table 1), as well as the stability limits on the real and imaginary axes, R^* and I^*, for various values of σ. Results comparing the accuracy of the scaled algorithm to that of the defining algorithm for the problem in Section 7 may be found in Table 2.c.

4. A Scaled, Fourth Order Runge-Kutta Algorithm

The fourth order, Runge-Kutta algorithm due to E. Fehlberg [1], having an embedded fifth order method for step size control, has particularly desirable properties for scaled algorithms. More specifically, once one additional derivative evaluation has been made, the solution may be determined at any number of points within the given step for a few arithmetic operations, while still maintaining the accuracy of the defining algorithm.

Since eqn.(2.8) is imposed for $i = 1$ and 2, the only remaining equations of condition reduce to:

(4.1) $\sum_{k=0}^{6} C^*_k \alpha_k^j = \sigma^j/(j+1)$

for $j = 0,1,2,3$, where $\alpha_0 = 0$, $C^*_1 = 0$, and

(4.2) $\sum_{k=2}^{6} C^*_k \alpha_k^j \beta_{k,1} = 0$, for $j = 0$. Sufficient parameters exist to solve Eqn. (4.2) for $j = 1$, a fifth order term, which results in overall lower TECs.

Since several solutions are sought within the step, the $\beta^*_{6,1}$ coefficient may not be used to solve Eqn. (4.2). Instead, only the C^* coefficients are available for generating the scaled solution with the $\beta_{6,k}$, for $\sigma = 1$, determined form Eqn. (2.8) and the remaining $\beta^*_{6,k}$ related to $\beta_{6,k}$ by a factor of $1/\sigma$ for $k = 0, 1, \ldots, 5$. (At first glance, a solution appears possible for $r = 5$. Eqn. (4.2), however, would require infinite C^* coefficients.)

Eqn. (2.8), for $j = 1$ and 2, determines the scaled $\beta^*_{6,4}$ and $\beta^*_{6,5}$ coefficients with the unscaled $\beta_{6,1}$, $\beta_{6,2}$, and $\beta_{6,3}$ as free parameters, giving

$$(4.3) \qquad \beta^*_{6,4} = \frac{\alpha_5^2(2\alpha_6 - 3\alpha_5) - 6\sum_{k=1}^{3} \beta_{6,k}\, \alpha_k(\alpha_k - \alpha_5)}{6\sigma\, \alpha_4(\alpha_4 - \alpha_5)}$$

with $\beta^*_{6,5}$ determined by permuting indices 4 and 5, where * denotes coefficients from the scaled algorithm, while the absence of * denotes coefficients from the defining algorithm or α^*_6 or $\beta^*_{6,k}$ for $\sigma = 1$. Eqn. (4.1), for $j = 1, 2, 3$, determines C^*_2, C^*_3, and C^*_4 in terms of C^*_5, C^*_6, and σ. Eqn. (4.2), for $j = 1$ and 2, then gives C^*_5 and C^*_6 solely in terms of σ.

$$(4.4.a) \quad C^*_2 = (AA_2 \pm DD_2 C^*_5 + EE_2 C^*_6)$$

where

$$(4.4.b) \quad AA_2 = \sum_{k=1}^{3} BB_{2,k}\sigma^k$$

$$= (\sigma^3/4 - \sigma^2(\alpha_3 + \alpha_4)/3 + \sigma\, \alpha_3\alpha_4)/(\alpha_2(\alpha_2 - \alpha_3)(\alpha_2 - \alpha_4))$$

$$(4.4.c) \quad DD_2 = -\alpha_5(\alpha_5^2 - \alpha_5(\alpha_3 + \alpha_4) + \alpha_3\alpha_4)/(\alpha_2(\alpha_2 - \alpha_3)(\alpha_2 - \alpha_3))$$

$$(4.4.d) \quad EE_2 = -\alpha_6(\alpha_6^2 - \alpha_6(\alpha_3 + \alpha_4) + \alpha_3\alpha_4)/(\alpha_2(\alpha_2 - \alpha_3)(\alpha_2 - \alpha_4))$$

with C^*_3 and C^*_4 determined by permuting indices, 2, 3, 4. The C^*_5 and C^*_6 terms become

$$(4.5.a) \quad C_5^* = \sum_{j=1}^{3} \frac{b_{22}d_{1j} - b_{12}d_{2j}}{b_{11}b_{22} - b_{21}b_{12}} \sigma^j$$

$$(4.5.b) \quad C_6^* = \sum_{j=1}^{3} \frac{b_{11}d_{2j} - b_{21}d_{1j}}{b_{11}b_{22} - b_{21}b_{12}} \sigma^j$$

where

$$(4.5.c) \quad b_{i,1} = \sum_{k=2}^{6} DD_k \, \alpha_k^{i-1} \, \beta_{k,1}$$

$$(4.5.d) \quad b_{i,2} = \sum_{k=2}^{6} EE_k \, \alpha_k^{i-1} \, \beta_{k,1}$$

and

$$(4.5.e) \quad d_{i,j} = \sum_{k=2}^{4} BB_{k,j} \, \alpha_k^{i-1} \, \beta_{k,1}$$

where $DD_5 = 1$, $DD_6 = 0$, $EE_5 = 0$, and $EE_6 = 1$. Then all C^* coefficients may be written as polynomials in σ.

Using the RKF(4)5 coefficients listed in Table 3., and chosing $\alpha_6 = \alpha_5$, and $\beta_{6,1} = \beta_{6,2} = \beta_{6,3} = 0$, the scaled coefficients become

$$(4.6.a) \quad C_0^* = 1 - \sigma(301/120 + \sigma(-269/108 + \sigma(311/360)))$$

$$(4.6.b) \quad C_2^* = \sigma(7168/1425 + \sigma(-4096/513 + \sigma(14848/4275)))$$

$$(4.6.c) \quad C_3^* = \sigma(-28561/8360 + \sigma(199927/22572 + \sigma(-371293/75240)))$$

$$(4.6.d) \quad C_4^* = \sigma(57/50 + \sigma(-3 + \sigma(42/25)))$$

$$(4.6.e) \quad C_5^* = \sigma(-96/55 + \sigma(40/11 + \sigma(-102/55)))$$

$$(4.6.f) \quad C_6^* = \sigma(3/2 + \sigma(-4 + \sigma(5/2)))$$

with $C_1^* = 0$, and.

(4.7.a) $\beta_{6,0}^* = \beta_{6,4}^* = 1/(6\ \sigma)$

and

(4.7.b) $\beta_{6,5}^* = 2/(3\ \sigma)$.

As σ approaches unity, all fifth order TEC approach zero, giving a true fifth order solution at $\sigma = 1$. One should note that since the α_6^* and $\beta_{6,k}^*$ coefficients are multiplied by σ during the evaluation of f_6^*, f_6^* remains unchanged when differing values of σ are used. Therefore, f_6 needs to be evaluated only once.

Coefficients for the defining algorithm, RKF(4)5 are listed in Table 3, with the values of G_5^*, Eqn.(3.7), the Euclidean norm of the fifth order truncation error coefficients, as well as the stability limits along the real and imaginary axes, being presented in Table 4.a for various values of σ. Accuracy comparisons between the fourth order solution and the fourth order scaled solution are given in Table 4.b, with results for the fifth order solution having a scaled, fourth order solution being given in Table 4.c for the problem stated in Section 7.

5. A Five-Stage, Fifth Order, Scaled RK Algorithm

Due to the complexity of the equations of condition for the fifth order, Runge-Kutta method, the generation of a fifth order, scaled algorithm requires more additional derivative evaluations than are needed for the lower order, scaled solutions. To generate a scaled solution which is valid for all values of σ, once the set of additional evaluations has been made, requires five more function evaluations (five stages). Such an algorithm is ideally suited for the problem of dense output, e.g., for iterative schemes over a given step.

The equations of condition for the fifth order RK algorithm are given in Table 1, with additional assumptions, Eqn. (2.8), being used to force some of these equations to become redundant. Eqn. (2.8), for $j = 1$ and 2, reduces the equations of condition to

(5.1) $$\sum_{j=0}^{r^*} C_j^* \alpha_j^i = \sigma^i/(i+1)$$

for $i = 0, 1, 2, 3$, and 4,

(5.2)-(5.4) $$\sum_{j=1}^{r^*} C_j^* \alpha_j^i \beta_{j,k} = 0$$

with $i = 0$ or 1 for $k = 1$, and with $i = 0$ for $k = 2$,

(5.5) $$\sum_{j=2}^{r^*} C_j^* \sum_{k=1}^{j-1} \beta_{j,k} \alpha_k^3 = \sigma^4/20$$

and

(5.6) $$\sum_{j=4}^{r^*} C_j^* \sum_{k=3}^{j-1} \beta_{j,k} \beta_{k,1} = 0$$.

where $r^* = 10$ and where * indicates coefficients in the scaled algorithm, while the absence of * indicates either coefficients in the defining algorithm or α_j^*, $\beta_{j,k}^*$ for $\sigma = 1$.

The solution of (5.1) requires the inversion of an n×n Vandermonde matrix (i.e., a matrix with element $V_{i,j} = \alpha_{j+k_0}^{i-1}$ where k_0 is a constant), whose analytic solution involves the coefficients of the Lagrange polynomial, with the coefficients given by

(5.7) $$L_i(\chi) = \frac{\prod_{k=1}^{n}{}' (\chi - \mu_k)}{\prod_{k=1}^{n}{}' (\mu_i - \mu_k)} = \sum_{j=1}^{n} a_{i,j}\, \chi^{j-1}$$

on the defining set $M = \{\mu_1, \ldots, \mu_n\}$ where ' indicates $k \neq i$ and where $a_{i,j}$ is the (i,j) element of the inverse matrix.

A scaled, fifth order algorithm in which the solution may be determined at any number of points within a given integration step at a total cost of five derivative evaluations (plus the arithmetic operations used in evaluating the new C^*'s for each solution point), may be derived by setting the C_j, coefficients of the existing f_j

terms equal to zero. Then, by adding five new derivative evaluations, f_6^*, f_7^*, f_8^*, f_9^*, and f_{10}^*, with $\beta_{j,1}^* = \beta_{j,2}^* = 0$, for $j=6,\ldots,10$, and imposing Eqn. (2.8), for $i = 1,2$, and 3, the remaining equations of condition become: 1) Eqn. (5.1), for $i = 0,\ldots,4$, and 2) Eqn. (5.6). Eqn. (5.1) determines all C^* coefficients in terms of one C^*, say C_6^*, while Eqn. (2.8) determines $\beta_{j,3}^*$, $\beta_{j,4}^*$ and $\beta_{j,5}^*$, for $j = 6,\ldots,10$, with $\beta_{k,6}^*,\ldots,\beta_{k,k-1}^*$, $k = 7,\ldots,10$, as free parameters. As in the fourth order, scaled algorithm all $\beta_{j,k}^*$ coefficients are related to their corresponding $\beta_{j,k}$ for $\sigma = 1$, and therefore, $f_6^*,\ldots,f_{10}^*$ need to be evaluated only once for a given integration step. The C^* coefficients may be written

$$(5.8) \qquad C_{j+6}^* = \sum_{k=1}^{4} (a_{j,5-k}\sigma^{5-k}/((6-k)\alpha_{j+6})) - C_6^* (\sum_{k=1}^{4} \alpha_6^{5-k} \; a_{j,5-k}/\alpha_{j+6}) =$$

$$= \sum_{k=1}^{4} AA_{j+6,k}\sigma^k + BB_{j+6}\, C_6^*$$

with $j=1,2,3,4$, where the $a_{j,k}$ terms are defined on the set $M = \{\alpha_7,\alpha_8,\alpha_9,\alpha_{10}\}$, Eqn. (5.7), and the $\beta_{j,k}^*$ coefficients

$$(5.9) \qquad \beta_{j,k}^* = \sum_{m=1}^{3} (\alpha_j^{5-m}/(5-m) - KK_{j,4-m})\; a_{k-2,4-m}/(\sigma\, \alpha_j)$$

for $j=6,7,8,9,10$, $k=3,4,5$ where $a_{k-2,m}$ is defined on the set $M = \{\alpha_3,\alpha_4,\alpha_5\}$, Eqn. (5.7), and where

$$(5.10) \qquad KK_{j,k} = \sum_{n=6}^{j-1} \beta_{j,n}\, \alpha_n^k$$

for $j=7,8,9,10$ with $KK_6=0$. Now defining

$$(5.11) \qquad SS_j = \sum_{k=3}^{j-1} \beta_{j,k}\, \beta_{k,1} \quad ,$$

and solving Eqn. (5.6), C_6^* may be written as a polynomial in σ,

$$(5.12) \qquad C_6^* = \frac{-\sum_{j=7}^{10} (\sum_{k=1}^{4} AA_{j,k} SS_j\, \sigma^k)}{SS_6 + \sum_{J=7}^{10} BB_j SS_j}$$

from which all C_k^* terms may be written as fourth order polynomials in σ, (5.13) $C_j^* = \sigma(A_{j,1} + \sigma(A_{j,2} + \sigma(A_{j,3} + \sigma A_{j,4})))$

for j=6,7,8,9,10, where C_j^*=0, for j=1,2,3,4,5 and where $C_6^* = 1-(C_6^*+C_7^*+C_8^*+C_9^*+C_{10}^*)$.

Coefficients $A_{j,k}$ are listed in Table 5.a, for the defining set, RKF(4)5, (Table 3), while the norm of the sixth order TEC terms, G_6^*, Eqn.(3.7), and the stability limits along the real and imaginary axes are listed in Table 5.b. Test results comparing the accuracy of the scaled solution to that of the defining algorithm are given in Table 5.c for the problem stated in Section 7.

6. A Two Stage, Fifth Order, Scaled RK Algorithm

The RKF(4)5 algorithm has a fifth order, scaled solution available for only two additional derivative evaluations, (Horn, [2]). (Each additional solution, however, requires two more function evaluations.) Efficient use of such an algorithm may require *a priori* knowledge of the σ values, since dynamically generating the scaled coefficients could use more computing time than the actual derivative evaluations. Thus, one may need to use σ values for which the scaled coefficients have been stored. By adjusting the step length externally so that the desired output point coincides with a predetermined value of σ, the Runge-Kutta algorithm may be operated at near optimal step length, while still determining the solution at the desired output point. For dense output within a given step, however, the five stage algorithm, (Section 5), is more efficient.

While a single stage, scaled solution appears possible from Eqns. (5.1)-(5.6), such a derivation leads to infinite C* values (for σ≠1), so that one must impose two additional derivative evaluations. Then Eqn. (5.1) may be solved for j=1,2,...,6, giving

(6.1) $$C_{j+1}^* = \sum_{k=1}^{6} a_{j,7-k}\sigma^{7-k}/((8-k)\alpha_{j+1})$$

defined upon the set, $M=\{\alpha_2,\alpha_3,\ldots,\alpha_7\}$ for j=1,2,...,6, where C_1^*=0, and where C_0^* is determined from

(6.2) $$C_0^* = 1-(C_1^*+C_2^*+C_3^*+C_4^*+C_5^*+C_6^*+C_7^*)$$

Eqns. (5.2)-(5.4) determine $\beta_{6,1}^*$, $\beta_{7,1}^*$, and $\beta_{7,2}^*$ with $\beta_{6,2}^*$ as a free parameter, giving

(6.3.a) $$\beta_{6,1}^* = (K_1-K_0\alpha_7)/(C_6^*\sigma(\alpha_6-\alpha_7))$$

and

(6.3.b) $$\beta_{7,1}^* = (K_1-K_0\alpha_6)/(C_7^*\sigma(\alpha_7-\alpha_6))$$

where

(6.3.c) $$K_j = -\sum_{k=2}^{5} C_k^*\alpha_k^j\beta_{k,1}$$

for j=0,1, and

(6.4) $$\beta_{7,2}^* = -(\sum_{j=2}^{6} C_j^*\beta_{j,2})/(\sigma\, C_7^*) \quad .$$

With $\beta_{6,5}^*$ as an additional free parameter, Eqns. (2.8), for i=1 and 2, determine $\beta_{6,3}^*$ and $\beta_{6,4}^*$,

(6.5) $$\beta_{6,3}^* = (K_2-K_1\alpha_4)/(\sigma\, \alpha_3(\alpha_3-\alpha_4))$$

and

(6.6.a) $$\beta_{6,4}^* = (K_2-K_1\alpha_3)/(\sigma\, \alpha_4(\alpha_4-\alpha_3))$$

where

(6.6.b) $$K_i = \alpha_6^{i+1}/(i+1)-\beta_{6,1}\alpha_1^i-\beta_{6,2}\alpha_2^i-\beta_{6,5}\alpha_5^i$$

for i=1,2. Eqns. (5.5) and (5.6), along with Eqns. 2.8 for i=1,2, j=7, determine the remaining unknowns, $\beta_{7,3}^*$, $\beta_{7,4}^*$, $\beta_{7,5}^*$ and $\beta_{7,6}^*$. Eqn. (5.5) gives

(6.7) $$P_{7,3} = (\sigma^4/20 - \sum_{j=2}^{6} C_j^* P_{j,3})/C_7^* \quad ,$$

and

(6.8) $$S_j = C_j^* \sum_{k=3}^{j-1} \beta_{j,k}\, \beta_{k,1}/C_7^* \quad ,$$

gives

(6.9.a) $$b^*_{7,6} = \frac{-\sum_{j=4}^{6} S_j - \sum_{j=3}^{5} A_j\, \beta_{j,1}}{\sigma((\sum_{j=3}^{6} B_j\, \beta_{j,1})+\beta_{7,1})}$$

where

(6.9.b) $$A_3 = (K_3-K_2(\alpha_4+\alpha_5)+K_1\alpha_4\alpha_5)/(\alpha_3(\alpha_3-\alpha_4)(\alpha_3-\alpha_5))$$

(6.9.c) $$B_3 = -\alpha_6(\alpha_6^2-\alpha_6(\alpha_4+\alpha_5)+\alpha_4\alpha_5)/(\alpha_3(\alpha_3-\alpha_4)(\alpha_3-\alpha_5))$$

and

(6.9.d) $$K_i = P_{7,i}-\beta_{7,1}\alpha_1^i-\beta_{7,2}\alpha_2^i$$

with A_4, B_4, A_5, and B_5 determined by permuting indices 3,4, and 5 on the α terms. The remaining $\beta^*_{7,k}$'s may be determined from

(6.10) $$\beta^*_{7,k} = A_k + B_k\, \beta^*_{7,k}$$

for $k = 3,4,5$.

The free parameters, α_6^*, α_7^*, $\beta^*_{6,2}$ and $\beta^*_{6,5}$, may be used to reduce the sixth order TECs for a given value of σ. The RKF(4)5 coefficients, Table 3, have sixth order TECs which are quite sensitive to the choice of α_6^* and α_7^*, but, in general, only mildly sensitive to the choice of $\beta^*_{6,2}$ and $\beta^*_{6,3}$. In fact, one may set the free $\beta^*_{j,k}$ terms to zero without greatly changing the resulting norm of the TECs, G_6^*, Eqn.(3.7). Suggested values of α_6^* and α_7^*, and the resulting G_6^* are presented in Table 6.a for a wide range of σ, while several sets of scaled coefficients are presented in Table 6.b. Test results comparing the accuracy of the scaled, fifth order solution to that of the defining algorithm are given in Table 6.c for the test problem stated in the following section.

7. Test Problem

The two-body problem with an elliptical orbit of eccentricity 0.6 is described by the following system of equations:

$$(7.1) \qquad y_1' = y_3, \quad y_2' = y_4, \quad y_3' = -y_1/r^3, \quad y_4' = -y/r_2^3$$

where $r = (y_1^2+y_2^2)^{1/2}$, and where $y_1(0) = 0.4$, $y_2(0) = 0.0$, $y_3(0) = 0.0$, $y_4(0) = 2.0$. The solution at t may be generated by solving Kepler's equation, u - 0.6*sir(u) = t, for u giving $y_1(t)$ = cos(u)-0.6, $y_2(t)$ = 0.8*sin(u), $y_3(t)$ = -sin(u)/(1-0.6*cos(u)), and $y_4(t)$ = 0.8*cos(u)/(1-0.6*cos(u)).

The solution of Eqns. (7.1), has been generated for several specified accuracies, using the RKF45 software package due to Watts and Shampine [3], whose core is the RKF(4)5 fifth order solution. This package has also been adapted into an RKT23 routine, which uses the RKT(2)3 coefficients for generating a third order solution. Both software packages were operated in a step-by-step mode with the resulting solution for each step being scaled for various values of σ. All computations were performed in double precision (16 digits) on the AMDAHL 470 V/6 and IBM 3020 computing system, operating system OS/MVS, (using a WATFIV compiler) at the DFVLR, Oberpfaffenhofen.

The average of the absolute value of the errors of each component during one revolution, $(0 < t < 2\pi)$, are presented in Tables 2.c, for the third order, scaled solution, 4.b and 4.c for the fourth order, scaled solution, 5.c for the five stage, fifth order, scaled solution, and 6.c for the two stage, fifth order, scaled solution, where the defining solution has remained unaffected by the scaled solution. The solutions for the third order integrator are listed for requested accuracies of 1.DO-02 and 1.OD-04, whereas those for the fourth and fifth order solutions are listed for 1.OD-04 and

1.0D-06, since, in general, one would operate the two methods using different specified tolerances. The tables show that the scaled solutions maintain the accuracy of the defining algorithm. Of particular importance are Tables 4.b and 4.c, which compare the fourth order, scaled solution with the defining algorithm being operated as a fourth and a fifth order method, respectively. The RKF(4)5 unscaled solution in Table 4.c (of fifth order) is used to generate a fourth order scaled solution. Thus, the fourth order errors in the scaled solution result from the local trancation errors and not from the conditions at the beginning of the integration step, which are of fifth order accuracy. Since the error estimate requires that the difference between the fourth and fifth order solutions be less than a prescribed tolerance, one would expect excellent agreement between the fourth order scaled solution and the fifth order defining solution. Thus, for determining data throughout a step, the fourth order solution may suffice. If, however, the solution is to be advanced repeatedly from a scaled output point, the user may wish to generate a fifth order scaled solution, since frequent use of a fourth order solution may tend to degrade the accuracy of a fifth order, defining algorithm.

8. Conclusions

Scaled Runge-Kutta algorithms which generate the solution of an ODE within a given integration step have been developed for orders three through five. Each scaled algorithm is used in conjunction with a defining Runge-Kutta algorithm with the derivative evaluations of the defining formula forming the core of the new scaled methods. For the third order, scaled solution, no further derivative evaluations are required, while for the fourth order, scaled solution, any number of scaled solutions may be generated within a given step at a total cost of only one added derivative evaluation (plus the arithmetic operations involved in generating the scaled coefficients). A scaled, fifth order RK solution is available for an additional two derivative evaluations, but each further solution requires two more such evaluations. This two stage, scaled algorithm may also require storage of the RK scaled coefficients for specified values of the scaling parameter or *a priori* knowledge of the desired output point if the computing time for the scaled coefficients equals that needed for derivative evaluations. A more general fifth order, scaled algorithm permits any number of solutions to be generated during a given step for a total of five additional derivative evaluations, making the method more suitable for dense output within a given step. Test results show that the accuracy of the scaled algorithmic solution is comparable to the accuracy achieved using the defining algorithm.

9. Acknowledgements

The author wishes to thank Mr. Victor R. Bond, who encouraged the development of the two stage, fifth order scaled algorithm, and the National Research Council, NASA Post Doctoral Fellowship, which supported the work. She extends her thanks to Dr. D.G. Bettis for his support during the early stages in the development of the scaled algorithms, and particularly thanks Dr. K.H. Well for his encouragement and support during the development of the scaled algorithms for treating dense output.

10. References

[1] Fehlberg, E. — Low-Order Classical Runge-Kutta Formulas with Step Size Control and Their Application to Some Heat Transfer Problems. NASA TR R-315 (July 1969).

[2] Horn, M.K. — Scaled Runge-Kutta Algorithms for Treating the Problem of Dense Output. TMX- (number to be assigned).

[3] Shampine, L.F., Watts, H.A. — Practical Solution of Ordinary Differential Equations by Runge-Kutta Methods. SAND 76-0585 (1976). Sandia Laboratories, Albuquerque, New Mexico (Dec. 1976).

[4] Shampine, L.F., Watts, H.A., Davenport, S.M. — Solving Nonstiff Ordinary Differential Equations - the State of the Art. SIAM Review 18 (1976) No. 3, pp. 376-410.

11. Tables

Table 1. Truncation Error Coefficients, $T_{i,j}$, for Runge-Kutta Algorithms of order i, i=1,2,3,4,5.

$$T_{1,1} = \sum_{i=0}^{r} c_i - 1$$

$$T_{2,1} = \sum_{i=1}^{r} c_i \alpha_i - \frac{1}{2}$$

$$T_{3,1} = \frac{1}{2} \left(\sum_{i=1}^{r} c_i \alpha_i^2 - \frac{1}{3} \right)$$

$$T_{3,2} = \sum_{i=2}^{r} c_i \sum_{j=1}^{i-1} \beta_{i,j} \alpha_j - \frac{1}{6}$$

$$T_{4,1} = \frac{1}{6} \left(\sum_{i=1}^{r} c_i \alpha_i^3 - \frac{1}{4} \right)$$

$$T_{4,2} = \sum_{i=2}^{r} c_i \alpha_i \sum_{j=1}^{i-1} \beta_{i,j} \alpha_j - \frac{1}{8}$$

$$T_{4,3} = \frac{1}{2} \left(\sum_{i=2}^{r} c_i \sum_{j=1}^{i-1} \beta_{i,j} \alpha_j^2 - \frac{1}{12} \right)$$

$$T_{4,4} = \sum_{i=3}^{r} c_i \sum_{j=2}^{i-1} \beta_{i,j} \sum_{k=1}^{j-1} \beta_{j,k} \alpha_k - \frac{1}{24}$$

$$T_{5,1} = \frac{1}{24} \left(\sum_{i=1}^{r} c_i \alpha_i^4 - \frac{1}{5} \right)$$

$$T_{5,2} = \frac{1}{2} \left(\sum_{i=2}^{r} c_i \alpha_i^2 \sum_{j=1}^{i-1} \beta_{i,j} \alpha_j - \frac{1}{10} \right)$$

$$T_{5,3} = \frac{1}{2} \left(\sum_{i=2}^{r} c_i \alpha_i \sum_{j=1}^{i-1} \beta_{i,j} \alpha_j^2 - \frac{1}{15} \right)$$

$$T_{5,4} = \sum_{i=3}^{r} C_i \, \alpha_i \sum_{j=2}^{i-1} \beta_{i,j} \sum_{k=1}^{j-1} \beta_{j,k} \, \alpha_k - \frac{1}{30}$$

$$T_{5,5} = \frac{1}{6} \left(\sum_{i=2}^{r} C_i \sum_{j=1}^{i-1} \beta_{i,j} \, \alpha_j^3 - \frac{1}{20} \right)$$

$$T_{5,6} = \sum_{i=3}^{r} C_i \sum_{j=2}^{i-1} \beta_{i,j} \, \alpha_j \sum_{k=1}^{j-1} \beta_{j,k} \, \alpha_k - \frac{1}{40}$$

$$T_{5,7} = \frac{1}{2} \left(\sum_{i=3}^{r} C_i \sum_{j=2}^{i-1} \beta_{i,j} \sum_{k=1}^{j-1} \beta_{j,k} \, \alpha_k^2 - \frac{1}{60} \right)$$

$$T_{5,8} = \sum_{i=4}^{r} C_i \sum_{j=3}^{i-1} \beta_{i,j} \sum_{k=2}^{j-1} \beta_{j,k} \sum_{l=1}^{k-1} \beta_{k,l} \, \alpha_l - \frac{1}{120}$$

$$T_{5,9} = \frac{1}{2} \left(\sum_{i=2}^{r} C_i \left(\sum_{j=1}^{i-1} \beta_{i,j} \, \alpha_j \right)^2 - \frac{1}{20} \right)$$

Table ~~2.a~~ Coefficients for a Runge-Kutta (Second) Third Order Algorithm, RKT(2)3

j	α_j	$\beta_{j,k}$			c_j	$\bar{c}_j$
		k= 0	1	2		
0	0				1/2	2/9
1	1/2	1/2			0	1/3
2	3/4	0	3/4		0	4/9
3	1	2/9	1/3	4/9	1/2	0

Note: 3rd order solutions are formed with $\bar{c}_j$ coefficients; 2nd order solutions are formed with c_j coefficients.

$$\text{Error estimate} = |y - \bar{y}| = \left| \sum_{j=0}^{3} ((c_j - \bar{c}_j) * f_j) \right|$$

Table 2.b Characteristic Properties for the Third Order, Scaled RKT(2)3 Algorithm.

Sigma	$\lvert TEC_4 \rvert$	R^*	I^*
0.00	0.4181109D-01	(-2.512, 0)	(0, 1.732)
0.05	0.5393866D-03	(-8.056, 0)	(0, 8.247)
0.10	0.1394878D-02	(-5.681, 0)	(0, 7.777)
0.15	0.3358796D-02	(-4.648, 0)	(0, 4.672)
0.20	0.6325054D-02	(-4.046, 0)	(0, 4.008)
0.25	0.9433857D-02	(-3.644, 0)	(0, 3.551)
0.30	0.1256625D-01	(-3.354, 0)	(0, 3.211)
0.35	0.1579347D-01	(-3.136, 0)	(0, 2.945)
0.40	0.1902608D-01	(-2.966, 0)	(0, 2.729)
0.45	0.2218770D-01	(-2.831, 0)	(0, 2.550)
0.50	0.2521847D-01	-2.722, 0)	(0, 2.399)
0.59	0.2807299D-01	(-2.634, 0)	(0, 2.270)
0.60	0.3071779D-01	(-2.564, 0)	(0, 2.158)
0.65	0.3312337D-01	(-2.508, 0)	(0, 2.062)
0.70	0.3528587D-01	(-2.466, 0)	(0, 1.978)
0.75	0.3717357D-01	(-2.436, 0)	(0, 1.907)
0.80	0.3377359D-01	(-2.418, 0)	(0, 1.847)
0.85	0.4006441D-01	(-2.414, 0)	(0, 1.799)
0.90	0.4101979D-01	(-2.424, 0)	(0, 1.763)
0.95	0.4161061D-01	(-2.454, 0)	(0, 1.740)

$\lvert TEC_4 \rvert$, Euclidean norm of fourth order truncation error coefficients

Stability Limits: Real axis, $R \in (R^*, 0)$

Imaginary axis, $I \in (0, I^*)$

Table 2.c.—Accuracy Comparisons for the Third Order RKT(2)3 Algorithm with Third Order Scaled Solution (No Additional Derivative Evaluations).

Average of the absolute value of the error in each component over one revolution vs. sigma.

Requested Accuracy = 0.1000D-01
Number of Derivative Evaluations = 94 (unscaled solution)

Sigma	Avg |ERR y_1|	Avg |ERR y_2|	Avg |ERR y_3|	Avg |ERR y_4|
0.00	0.108092D-01	0.799637D-02	0.198615D-01	0.133819D-01
0.10	0.105145D-01	0.670180D-02	0.161793D-01	0.121847D-01
0.20	0.105536D-01	0.676512D-02	0.163445D-01	0.122379D-01
0.30	0.105853D-01	0.683129D-02	0.165148D-01	0.122899D-01
0.40	0.106106D-01	0.689935D-02	0.166887D-01	0.123357D-01
0.50	0.106301D-01	0.696860D-02	0.168637D-01	0.123745D-01
0.60	0.106448D-01	0.703833D-02	0.170380D-01	0.124058D-01
0.70	0.106555D-01	0.710791D-02	0.172093D-01	0.124293D-01
0.80	0.106630D-01	0.717653D-02	0.173761D-01	0.124443D-01
0.90	0.106681D-01	0.724365D-02	0.176165D-01	0.124598D-01

Requested Accuracy = 0.1000D-03
Number of Derivative Evaluations = 289 (unscaled solution)

Sigma	Avg |ERR y_1|	Avg |ERR y_2|	Avg |ERR y_3|	Avg |ERR y_4|
0.00	0.743613D-04	0.141640D-03	0.248142D-03	0.235040D-03
0.10	0.731829D-04	0.133985D-03	0.223898D-03	0.231014D-03
0.20	0.732310D-04	0.134249D-03	0.225077D-03	0.231244D-03
0.30	0.732928D-04	0.134616D-03	0.226193D-03	0.231481D-03
0.40	0.733597D-04	0.135025D-03	0.227392D-03	0.231732D-03
0.50	0.734330D-04	0.135451D-03	0.228597D-03	0.231989D-03
0.60	0.735116D-04	0.135883D-03	0.229814D-03	0.232248D-03
0.70	0.735940D-04	0.136324D-03	0.231041D-03	0.232507D-03
0.80	0.736652D-04	0.136761D-03	0.232285D-03	0.232733D-03
0.90	0.737535D-04	0.137185D-03	0.233497D-03	0.232958D-03

Table 3 Coefficients for the Runge-Kutta-Fehlberg (Fourth) Fifth Order Algorithm, RKF(4)5.

j	α_j	C_j	$\bar{C}_j$
0	0	25/216	16/135
1	1/4	0	0
2	3/8	1408/2565	1408/12825
3	12/13	2197/4104	6656/56430
4	1	-1/5	-9/50
5	1/2	0	2/55

j	$\beta_{j,k}$				
	k = 0	1	2	3	4
1	1/4				
2	3/32	9/32			
3	1932/2197	-7200/2197	7296/2197		
4	439/216	-8	3680/513	-845/4104	
5	-8/27	2	-3544/2565	1859/4104	-11/40

$$\text{Error estimate} = |y - \bar{y}| = \left| \sum_{j=0}^{5} ((C_j - \bar{C}_j) * f_j) \right|$$

Note: 5th order solutions are formed with $\bar{C}_j$ coefficients; 4th order solutions with C_j.

Table 4.a—Characteristic Properties for the Fourth Order, Single Stage, Scaled RKF(4)5 Algorithm

Sigma	$\lvert TEC_5 \rvert$	R^*	I^*
0.00	0.183124D-02	(-3.020, 0)	(3.274, 3.202)
0.05	0.127218D-03	(-4.434, 0)	(0, 0)
0.10	0.427646D-03	(-3.622, 0)	(0, 0)
0.15	0.300828D-03	(-3.241, 0)	(0, 0)
0.20	0.117244D-02	(-3.016, 0)	(0, 0)
0.25	0.149131D-02	(-2.872, 0)	(0, 0)
0.30	0.172644D-02	(-2.778, 0)	(0, 0)
0.35	0.186395D-02	(-2.722, 0)	(0, 0)
0.40	0.190401D-02	(-2.697, 0)	(0, 0)
0.45	0.135760D-02	(-2.703, 0)	(0, 0)
0.50	0.174305D-02	(-2.745, 0)	(0, 0)
0.55	0.158227D-02	(-2.837, 0)	(0, 0)
0.60	0.139658D-02	(-3.032, 0)	(0, 0)
0.65	0.120249D-02	(-3.786, 0)	(0, 0)
0.70	0.100351D-02	(-3.418, 0)	(0, 0)
0.75	0.314987D-03	(-2.979, 0)	(0, 0)
0.80	0.618160D-03	(-2.801, 0)	(0, 0)
0.85	0.417629D-03	(-2.738, 0)	(0, 0.6093)
0.90	0.224498D-03	(-2.774, 0)	(0, 0.6282)
0.95	0.679723D-04	(-3.002D 0)	(0, 0)

$\lvert TEC_5 \rvert$—Euclidean norm of fifth order truncation error coefficient

Stability Limits: Real axis, $R \in (R^*, 0)$

Imaginary axis, $I \in (I_1^*, I_2^*)$

Table 4.b Accuracy Comparisons for the Fourth Order RKF4(5) Algorithm with Fourth Order Scaled Solution (One Stage)

Average of the absolute value of the error in each component over one revolution vs. sigma.

Requested Accuracy = 0.1000D-03
Number of Derivative Evaluations = 117 (unscaled solution)

Sigma	Avg $\lvert \mathrm{ERR}\ y_1 \rvert$	Avg $\lvert \mathrm{ERR}\ y_2 \rvert$	Avg $\lvert \mathrm{ERR}\ y_3 \rvert$	Avg $\lvert \mathrm{ERR}\ y_4 \rvert$
0.00	0.2201628D-02	0.2363481D-02	0.5283069D-02	0.3862028D-02
0.10	0.2199240D-02	0.1662720D-02	0.3098084D-02	0.3869112D-02
0.20	0.2205637D-02	0.1694677D-02	0.3185227D-02	0.3889998D-02
0.30	0.2212536D-02	0.1727151D-02	0.3271972D-02	0.3908134D-02
0.40	0.2219272D-02	0.1760104D-02	0.3388099D-02	0.3923980D-02
0.50	0.2225409D-02	0.1793471D-02	0.3511778D-02	0.3939755D-02
0.60	0.2230342D-02	0.1827056D-02	0.3634780D-02	0.3952688D-02
0.70	0.2233963D-02	0.1860790D-02	0.3756650D-02	0.3962620D-02
0.80	0.2235830D-02	0.1896488D-02	0.3877272D-02	0.3969343D-02
0.90	0.2236027D-02	0.1933620D-02	0.3996676D-02	0.3972724D-02

Requested Accuracy = 0.1000D-05
Number of Derivative Evaluations = 278 (unscaled solution)

Sigma	Avg $\lvert \mathrm{ERR}\ y_1 \rvert$	Avg $\lvert \mathrm{ERR}\ y_2 \rvert$	Avg $\lvert \mathrm{ERR}\ y_3 \rvert$	Avg $\lvert \mathrm{ERR}\ y_4 \rvert$
0.00	0.8853653D-04	0.8668324D-04	0.1754191D-03	0.1536550D-03
0.10	0.8848032D-04	0.7603060D-04	0.1417430D-03	0.1532748D-03
0.20	0.8852633D-04	0.7658618D-04	0.1435634D-03	0.1532763D-03
0.30	0.8855515D-04	0.7715804D-04	0.1454413D-03	0.1532383D-03
0.40	0.8860156D-04	0.7770629D-04	0.1473113D-03	0.1531758D-03
0.50	0.8864331D-04	0.7819969D-04	0.1491015D-03	0.1531055D-03
0.60	0.8867674D-04	0.7871130D-04	0.1508857D-03	0.1530250D-03
0.70	0.8870430D-04	0.7924462D-04	0.1526622D-03	0.1529476D-03
0.80	0.8872082D-04	0.7976409D-04	0.1544141D-03	0.1528909D-03
0.90	0.8869219D-04	0.8022937D-04	0.1561135D-03	0.1528012D-03

Table 4.c Accuracy Comparisons for the Fifth Order RKF(4)5 Algorithm with Fourth Order Scaled Solution (One Stage).

Average of the absolute value of the error in each component over one revolution vs. sigma.

Requested Accuracy = 0.1000D-03
Number of Derivative Evaluations = 117 (unscaled solution)

Sigma	Avg |ERR y_1|	Avg |ERR y_2|	Avg |ERR y_3|	Avg |ERR y_4|
0.10	0.277124D-03	0.768345D-03	0.606257D-03	0.812701D-03
0.20	0.280695D-03	0.775102D-03	0.632935D-03	0.824149D-03
0.30	0.284825D-03	0.781798D-03	0.661357D-03	0.836419D-03
0.40	0.289274D-03	0.788786D-03	0.689995D-03	0.848384D-03
0.50	0.293376D-03	0.796045D-03	0.718751D-03	0.858550D-03
0.60	0.296850D-03	0.803683D-03	0.744956D-03	0.866417D-03
0.70	0.299410D-03	0.811642D-03	0.767672D-03	0.871352D-03
0.80	0.300788D-03	0.819816D-03	0.786527D-03	0.873093D-03
0.90	0.300857D-03	0.828078D-03	0.801508D-03	0.871657D-03

Requested Accuracy = 0.1000D-05
Number of Derivative Evaluations = 278 (unscaled solution)

Sigma	Avg |ERR y_1|	Avg |ERR y_2|	Avg |ERR y_3|	Avg |ERR y_4|
0.00	0.373769D-04	0.353999D-04	0.773149D-04	0.641644D-04
0.10	0.373616D-04	0.305350D-04	0.623438D-04	0.639782D-04
0.20	0.373902D-04	0.307689D-04	0.631054D-04	0.639738D-04
0.30	0.374033D-04	0.310067D-04	0.639115D-04	0.639911D-04
0.40	0.374370D-04	0.312388D-04	0.647056D-04	0.640047D-04
0.50	0.374639D-04	0.314214D-04	0.654074D-04	0.640249D-04
0.60	0.374938D-04	0.316295D-04	0.661055D-04	0.640266D-04
0.70	0.375148D-04	0.318693D-04	0.668776D-04	0.639988D-04
0.80	0.375246D-04	0.321151D-04	0.676578D-04	0.639714D-04
0.90	0.374842D-04	0.322994D-04	0.684391D-04	0.639167D-04

Table 5.a Coefficients for the Five Stage, Fifth Order, Scaled RKF(4)5 Algorithm.

Coefficients are listed for $\sigma = 1$, with $\beta_{j,1} = \beta_{j,2} = 0$, for $j = 6,7,8,9,10$, with α_j, $\beta_{j,k}$ given in Table 3 for $j < 6$.

$\alpha_{6} = 6 / 25$

$\alpha_{7} = 3 / 4$

$\alpha_{8} = 29 / 30$

$\alpha_{9} = 39 / 40$

$\alpha_{10} = 29 / 30$

$\beta_{6,0} = 113121 / 781250$

$\beta_{6,3} = -2379351 / 8593750$

$\beta_{6,4} = 76824 / 390625$

$\beta_{6,5} = 753696 / 4296875$

$\beta_{7,0} = 711437 / 32000000$

$\beta_{7,3} = 174357033 / 352000000$

$\beta_{7,4} = -2604771 / 8000000$

$\beta_{7,5} = 1245429 / 22000000$

$\beta_{7,6} = 1 / 2$

$\beta_{8,0} = 11647913 / 300000000$

$\beta_{8,3} = -533165763 / 1100000000$

$\beta_{8,4} = 16151543 / 75000000$

$\beta_{8,5} = -39081137 / 206250000$

$\beta_{8,6} = 1 / 2$

$\beta_{8,7} = 1 / 2$

$\beta_{9,0} = 22830393 / 512000000$

$\beta_{9,3} = -1803163583 / 5632000000$

$\beta_{9,4} = 28308821 / 128000000$

$\beta_{9,5} = -165661179 / 352000000$

$\beta_{9,6} = 1 / 2$

$\beta_{9,7} = 1 / 2$

$\beta_{9,8} = 1 / 2$

$\beta_{10,0} = 121551427 / 2400000000$

$\beta_{10,3} = -8315477879 / 8800000000$

$\beta_{10,4} = 65813119 / 600000000$

$\beta_{10,5} = -934650581 / 1650000000$

$\beta_{10,6} = 1 / 2$

$\beta_{10,7} = 1 / 2$

$\beta_{10,8} = 1 / 2$

$\beta_{10,9} = 1 / 2$

$A_{0,1} = 18661 / 4524$

$A_{0,2} = -76165 / 10179$

$A_{0,3} = 63625 / 10179$

$A_{0,4} = -20000 / 10179$

$A_{6,1} = 1173125 / 169932$

$A_{6,2} = -7215625 / 382347$

$A_{6,3} = 7203125 / 382347$

$A_{6,4} = -2500000 / 382347$

$A_{7,1} = 12064 / 2601$

$A_{7,2} = -166880 / 7803$

$A_{7,3} = 718000 / 23409$

$A_{7,4} = -320000 / 23409$

$A_{8,1} = -3066859635697684 / 1337382789874677$

$A_{8,2} = 2747862949557238 0 / 4012148369624031$

$A_{8,3} = -5859228892470550 0 / 12036445108872093$

$A_{8,4} = 5444997483920000 / 12036445108872093$

$A_{9,1} = -371200 / 452907$

$A_{9,2} = 5363200 / 1358721$

$A_{9,3} = -25120000 / 4076163$

$A_{9,4} = 12800000 / 4076163$

$A_{10,1} = -5795511385214816 / 1337382789874677$

$A_{10,2} = 8818667434 1465120 / 4012148369624031$

$A_{10,3} = -38793486635973200 0 / 12036445108872093$

$A_{10,4} = 17634722858608000 0 / 12036445108872093$

$\alpha_j^* = \alpha_j/\sigma, \quad \beta_{j,k}^* = \beta_{j,k}/\sigma$

$C_j^* = \quad (A_{j,1} + \sigma(A_{j,2} + \sigma(A_{j,3} + \sigma\ A_{j,4})))$

for $j = 6,7,8,9,10$, with $C_k^* = 0$, for $k = 1,2,3,4,5$ and

$C_0^* = 1 - \quad (A_{0,1} + \sigma(A_{0,2} + \sigma(A_{0,3} + \sigma\ A_{0,4})))$

Table 5.b Characteristic Properties for the Fifth Order, Five Stage, Scaled RKF(4)5 Algorithm.

Sigma	$\lvert TEC_6 \rvert$	R^*	I^*
0.00	0.3339775D-02	(-3.678, 0)	(2.046, 3.607)
0.05	0.4115561D-03	(-3.202, 0)	(0, 0)
0.10	0.1231088D-02	(-3.128, 0)	(0, 0)
0.15	0.2210340D-02	(-3.113, 0)	(0, 0)
0.20	0.2970554D-02	(-3.131, 0)	(0, 0)
0.25	0.3332843D-02	(-3.189, 0)	(0, 0)
0.30	0.3430460D-02	(-3.340, 0)	(0, 0)
0.35	0.3233464D-02	(-3.146, 0)	(0, 0)
0.40	0.2931142D-02	(-3.049, 0)	(1.501, 2.981)
0.45	0.2734775D-02	(-2.899, 0)	(0, 2.683)
0.50	0.2717859D-02	(-2.805, 0)	(0, 2.410)
0.55	0.2736137D-02	(-2.730, 0)	(0, 2.219)
0.60	0.2739563D-02	(-2.663, 0)	(0, 2.080)
0.65	0.2602328D-02	(-2.602, 0)	(0, 1.965)
0.70	0.2231226D-02	(-2.544, 0)	(0, 1.846)
0.75	0.2146668D-02	(-2.495, 0)	(0, 1.638)
0.80	0.2537549D-02	(-2.463, 0)	(0, 0)
0.85	0.3179205D-02	(-2.466, 0)	(0, 0)
0.90	0.3413383D-02	(-2.551, 0)	(0, 0)
0.95	0.2520916D-02	(-3.012, 0)	(0, 0)

$\lvert TEC_6 \rvert$, Euclidean norm of sixth order truncation error coefficients

Stability Limits: Real axis, $R^* \in (R^*, 0)$

Imaginary axis, $I^* \in (I^*_1, I^*_2)$

Table 5.e Accuracy Comparisons for the Fifth Order RKF(4)5 Algorithm with Fifth Order Scaled Solution (Five Stages).

Average of the absolute value of the error in each component over one revolution vs. sigma.

Requested Accuracy = 0.1000D-03
Number of Derivative Evaluations = 117 (unscaled solution)

Sigma	Avg $\lvert ERR\ y_1 \rvert$	Avg $\lvert ERR\ y_2 \rvert$	Avg $\lvert ERR\ y_3 \rvert$	Avg $\lvert ERR\ y_4 \rvert$
0.00	0.280100D-03	0.903693D-03	0.987765D-03	0.817048D-03
0.10	0.282579D-03	0.780164D-03	0.634261D-03	0.823408D-03
0.20	0.290666D-03	0.797997D-03	0.687977D-03	0.843550D-03
0.30	0.289251D-03	0.808492D-03	0.729663D-03	0.852017D-03
0.40	0.291007D-03	0.813082D-03	0.756138D-03	0.847604D-03
0.50	0.290452D-03	0.817466D-03	0.775154D-03	0.835958D-03
0.60	0.292738D-03	0.828221D-03	0.798258D-03	0.825291D-03
0.70	0.280713D-03	0.849522D-03	0.833153D-03	0.822062D-03
0.80	0.260057D-03	0.879855D-03	0.911652D-03	0.831687D-03
0.90	0.280883D-03	0.909604D-03	0.984397D-03	0.840744D-03

Requested Accuracy = 0.1000D-05
Number of Derivative Evaluations = 278 (unscaled solution)

Sigma	Avg $\lvert ERR\ y_1 \rvert$	Avg $\lvert ERR\ y_2 \rvert$	Avg $\lvert ERR\ y_3 \rvert$	Avg $\lvert ERR\ y_4 \rvert$
0.00	0.371140D-04	0.354093D-04	0.774081D-04	0.642269D-04
0.10	0.374693D-04	0.307795D-04	0.632404D-04	0.642209D-04
0.20	0.375383D-04	0.312677D-04	0.647821D-04	0.644220D-04
0.30	0.375572D-04	0.317971D-04	0.662699D-04	0.644859D-04
0.40	0.374685D-04	0.323365D-04	0.677887D-04	0.644072D-04
0.50	0.373560D-04	0.328930D-04	0.692797D-04	0.642754D-04
0.60	0.373014D-04	0.334150D-04	0.708427D-04	0.641876D-04
0.70	0.373411D-04	0.338935D-04	0.725455D-04	0.641679D-04
0.80	0.374655D-04	0.343385D-04	0.743786D-04	0.642263D-04
0.90	0.373359D-04	0.348043D-04	0.761000D-04	0.642368D-04

Table 6.a Free Parameter Selection and Characteristic Properties for the Fifth Order, Two Stage, Scaled RKF4(5) Algorithm.

Sigma	α_6	α_7	TEC_6	R^*	I^*
0.00	-	-	0.3356D-02	(3.678,0)	(2.045,3.607)
0.50	0.3500	0.850	0.2546D-06	(-11.123,0)	(0,0)
0.10	0.2500	0.800	0.4513D-05	(-7.033,0)	(0,3.604)
0.15	0.4500	0.900	0.1162D-03	(-4.339,0)	(0,0)
0.20	0.4166*	1.000	0.3767D-04	(-5.313,0)	(0,0)
0.25	0.2500	1.000	0.1642D-02	(-2.501,0)	(0,2.520)
0.30	0.3750	1.000	0.2347D-03	(-4.040,0)	(3.213,4.104)
0.35	0.3500	1.000	0.4977D-03	(-3.514,0)	(2.814,3.492)
0.40	0.3125	1.000	0.8575D-03	(-3.436,0)	(2.596,3.581)
0.45	0.2727*	1.000	0.1500D-02	(-3.203,0)	(2.360,3.407)
0.50	0.5200	0.470	0.3295D-03	(-3.402,0)	(0,3.181)
0.55	0.2600	0.830	0.3221D-02	(-2.765,0)	(1.988,2.991)
0.60	0.3300	0.720	0.3812D-02	(-2.700,0)	(1.870,2.979)
0.65	0.4500	0.600	0.1775D-02	(-3.210,0)	(2.108,3.636)
0.70	0.4600	0.590	0.1430D-02	(-3.997,0)	(0,0)
0.75	0.4500	0.590	0.3651D-02	(-5.202,0)	(0,0)
0.80	0.4300	0.580	0.3770D-02	(-2.701,0)	(1.605,3.120)
0.85	0.4200	0.550	0.3611D-02	(-2.789,0)	(1.605,3.232)
0.90	0.4100	0.500	0.3609D-02	(-2.270,0)	(0,1.142)
0.95	0.3750	0.450	0.1910D-01	(-1.945,0)	(1.260,2.208)

* repeating decimals

Free-Parameters: α_6, α_7

Characteristic properties:

$|TEC_6|$, Euclidean norm of sixth order truncation error coefficients

Stability Limits: Real axis, $R^* \in (R^*, 0)$

Imaginary axis, $I^* \in (I^*_1, I^*_2)$

Table 6.b Coefficients for the Two Stage, Fifth-Order, Scaled-RKF(4)5 Algorithm for Values of the Scaling Parameter, σ.

$A(j) = \alpha_j / \sigma$

$B(j,k) = \beta_{j,k} / \sigma$ $j = 1,2,3,4,5;\ k = 0,1,\ldots,j-1$ (from Table 3)

σ = 0.2000000000000000D+00

A(6)	=	0.4166666666666667D+00
A(7)	=	0.1000000000000000D+01
C0	=	0.1249895238095236D+00
C(1)	=	0.0000000000000000D+00
C(2)	=	-0.1741280111844241D-01
C(3)	=	-0.1726641291560887D-03
C(4)	=	0.8289177489177062D-04
C(5)	=	0.4130724386724331D-02
C(6)	=	0.6268760256948226D+00
C(7)	=	0.2615062995816362D+00
B0(6)	=	0.3254591429778357D+00
B(6,1)	=	0.1025525465501115D+00
B(6,2)	=	0.0000000000000000D+00
B(6,3)	=	-0.3088396535322991D-01
B(6,4)	=	0.2853894249194929D-01
B(6,5)	=	0.0000000000000000D+00
B0(7)	=	-0.2425211742757231D+00
B(7,1)	=	-0.3082978320350067D+00
B(7,2)	=	0.1087182249969140D+00
B(7,3)	=	0.5720610466680882D-02
B(7,4)	=	-0.9043353051310987D-02
B(7,5)	=	0.4706619454105300D-01
B(7,6)	=	0.1198051329357392D+01

σ	=	0.4000000000000000D+00
A(6)	=	0.3125000000000000D+00
A(7)	=	0.1000000000000000D+01
C0	=	0.9098835970834918D-01
C(1)	=	0.0000000000000000D+00
C(2)	=	0.1511689766638826D+01
C(3)	=	-0.1233204211194970D-02
C(4)	=	0.5317641723355576D-03
C(5)	=	0.9218955266955576D-01
C(6)	=	0.4898813273626358D+00
C(7)	=	-0.1184067566420507D+01
B0(6)	=	0.1997487995087076D+00
B(6,1)	=	0.1349930681999505D+00
B(6,2)	=	0.0000000000000000D+00
B(6,3)	=	-0.1043230585627165D+00
B(6,4)	=	0.8208119085405835D-01
B(6,5)	=	0.0000000000000000D+00
B0(7)	=	0.2400248761202557D+00
B(7,1)	=	0.1142029463421648D+01
B(7,2)	=	-0.2692274450122054D+00
B(7,3)	=	0.1212229776127877D+00
B(7,4)	=	-0.7216879014020882D-01
B(7,5)	=	-0.7744341804366997D-01
B(7,6)	=	-0.2344376639586075D+00

σ	=	0.5000000000000000D+00
A(6)	=	0.4300000000000000D+00
A(7)	=	0.5300000000000000D+00
C0	=	0.9177701438099328D-01
C(1)	=	0.0000000000000000D+00
C(2)	=	-0.8399911158151296D+01
C(3)	=	0.1190999330473669D+00
C(4)	=	-0.4779650070206371D-01
C(5)	=	-0.7674883851958462D+00
C(6)	=	0.6791125691350189D+01
C(7)	=	0.3713193405270657D+01
B0(6)	=	0.1321587392059496D+00
B(6,1)	=	0.2064715493884858D+00
B(6,2)	=	0.0000000000000000D+00
B(6,3)	=	0.1085878378491868D-01
B(6,4)	=	-0.1846907237935407D-01
B(6,5)	=	0.0000000000000000D+00
B0(7)	=	-0.2799871066289993D-02
B(7,1)	=	0.8198193282633387D+00
B(7,2)	=	-0.3747013763692046D+00
B(7,3)	=	0.7113607481166296D-01
B(7,4)	=	-0.3002685456165180D-01
B(7,5)	=	0.8076674026480739D-02
B(7,6)	=	0.8349602499566385D-01

σ	=	0.6000000000000000D+00
A(6)	=	0.3300000000000000D+00
A(7)	=	0.7200000000000000D+00
C0	=	0.9356034471712229D-01
C(1)	=	0.0000000000000000D+00
C(2)	=	-0.6489274121253740D+00
C(3)	=	0.1482179832556966D-01
C(4)	=	-0.7113612427683513D-02
C(5)	=	-0.2477401180863559D-02
C(6)	=	0.5625659895314688D+00
C(7)	=	0.9375702931597602D+00
B0(6)	=	0.1382626113915468D+00
B(6,1)	=	0.2346911060268858D+00
B(6,2)	=	0.0000000000000000D+00
B(6,3)	=	-0.2203622318522454D+00
B(6,4)	=	0.1774085144338128D+00
B(6,5)	=	0.0000000000000000D+00
B0(7)	=	0.1691491924687175D+00
B(7,1)	=	0.1686177657728288D+00
B(7,2)	=	-0.2725776674275082D-02
B(7,3)	=	-0.5296332249359525D-01
B(7,4)	=	0.2281937886920632D-01
B(7,5)	=	0.1129426157937126D+00
B(7,6)	=	0.2221621462664050D+00

Table 6.c Accuracy Comparisons for the Fifth Order RKF(4)5 Algorithms With Fifth Order Scaled Solution (Two Stages).

Average of the absolute value of the error in each component over one revolution vs. sigma.

Requested Accuracy = 0.1000D-03
Number of Derivative Evaluations = 117 (unscaled solution)

Sigma	Avg $\|ERR\ y_1\|$	Avg $\|ERR\ y_2\|$	Avg $\|ERR\ y_3\|$	Avg $\|ERR\ y_4\|$
0.00	0.280002D-03	0.903827D-03	0.987520D-03	0.816767D-03
0.10	0.276894D-03	0.776032D-03	0.621298D-03	0.815867D-03
0.20	0.277124D-03	0.790713D-03	0.658005D-03	0.824895D-03
0.30	0.278426D-03	0.805448D-03	0.695657D-03	0.831273D-03
0.40	0.279326D-03	0.821274D-03	0.740068D-03	0.837894D-03
0.50	0.274523D-03	0.832982D-03	0.757606D-03	0.830954D-03
0.60	0.284338D-03	0.854715D-03	0.836542D-03	0.857009D-03
0.70	0.271525D-03	0.860934D-03	0.808151D-03	0.820222D-03
0.80	0.277812D-03	0.877107D-03	0.855832D-03	0.828825D-03
0.90	0.276415D-03	0.884657D-03	0.896221D-03	0.779959D-03

Requested Accuracy = 0.1000D-05
Number of Derivative Evaluations = 278 (unscaled solution)

Sigma	Avg $\|ERR\ y_1\|$	Avg $\|ERR\ y_2\|$	Avg $\|ERR\ y_3\|$	Avg $\|ERR\ y_4\|$
0.00	0.373769D-04	0.353999D-04	0.773149D-04	0.641644D-04
0.10	0.373663D-04	0.307878D-04	0.630549D-04	0.640826D-04
0.20	0.373715D-04	0.313039D-04	0.646014D-04	0.642039D-04
0.30	0.373354D-04	0.317496D-04	0.659455D-04	0.642843D-04
0.40	0.374171D-04	0.322966D-04	0.676708D-04	0.643662D-04
0.50	0.373372D-04	0.327278D-04	0.691296D-04	0.641708D-04
0.60	0.374366D-04	0.333353D-04	0.710936D-04	0.643279D-04
0.70	0.373690D-04	0.337979D-04	0.723090D-04	0.640660D-04
0.80	0.374781D-04	0.342842D-04	0.737902D-04	0.639477D-04
0.90	0.371453D-04	0.349161D-04	0.754714D-04	0.636525D-04