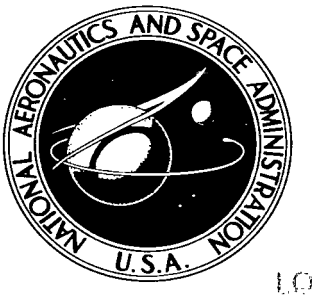
NASA TECHNICAL NOTE

NASA TN D-2946



NASA TN D-2946

ON THE EFFICIENT USE OF PREDICTOR-CORRECTOR METHODS IN THE NUMERICAL SOLUTION OF DIFFERENTIAL EQUATIONS

by David Rodabaugh and James R. Wesson

George C. Marshall Space Flight Center
Huntsville, Ala.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • AUGUST 1965

ON THE EFFICIENT USE OF PREDICTOR-CORRECTOR METHODS IN THE NUMERICAL SOLUTION OF DIFFERENTIAL EQUATIONS

By David Rodabaugh and James R. Wesson

George C. Marshall Space Flight Center
Huntsville, Ala.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – Price $2.00

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

ON THE EFFICIENT USE OF PREDICTOR-CORRECTOR METHODS IN THE NUMERICAL SOLUTION OF DIFFERENTIAL EQUATIONS

SUMMARY

Basic interpolation formulas by Hermite can be used to generate large classes of correctors to be used in predictor-corrector processes for the numerical solution of differential equations. Conditions for the numerical stability of predictor-corrector processes are known [ref. 2, p. 20; 4, p. 197; 5, p. 218]. Only stable processes should be used in any solution requiring more than a few steps.

Predictor-corrector methods are not self-starting; therefore, they are harder to program than starting processes. If programmed so that there are few wasted applications of the corrector, however, then the predictor-corrector processes will have a significant computing time advantage over starting methods.

The authors of this study are faculty members of the Mathematics Department of Vanderbilt University, Nashville, Tennessee, serving NASA under contract NAS8-2559.

SECTION I. INTRODUCTION

The primary purpose of this paper is to show how a predictor-corrector process can compete with a starting (self-starting) process.

Section II gives the derivation of Hermite interpolation formulas and their error terms. Also, the predictor-corrector process for solving differential equations is outlined. Numerical stability and propagated error are discussed. Some specific correctors are derived, and comments on the choice of a method are made.

In section III, the programming of predictor-corrector methods is discussed, analyzed, and illustrated. Comparisons with starting methods are discussed and charted. The usual programming of predictor-corrector methods is modified so as to save computing time, with no loss in numerical stability.

Roundoff errors are not discussed, but at least four "guarding" digits are maintained in all computations. That is, four digits beyond the accuracy of the method are carried along.

SECTION II. PREDICTOR-CORRECTOR METHODS

A. SOME BASIC FORMULAS

We present formulas for approximating a function f (x) when certain values of the function and its derivative are known. The function is assumed to be analytic in the interval under consideration, and the points at which the function is evaluated are regularly spaced along the x-axis. A general method of deriving such formulas is explained, and several specific formulas (with error terms included) are given [ref. 4, p. 128; 5, p. 209].

Let $y(x)$ be the function to be approximated, and let $x_i = x + ih$, $i = 0, 1, \ldots$. Denote the actual values of $y(x_i)$ and $y'(x_i)$ by z_i and z'_i. Let y_i, y'_i be approximations. We shall consider formulas of the type

$$y_{n+1} = \sum_{i=0}^{n} (A_i z_i + h B_i z'_i). \tag{1}$$

In the derivation of specific formulas of type (1), the following is important:

Theorem. If (1) is exact (that is, $y_{n+1} = z_{n+1}$) for $y(x) = x^m$ (m a positive integer), then (1) is exact for any polynomial $y(x)$ of degree not exceeding m.

Proof: First it is proved that if (1) is exact for $y(x) = x^m$, then it is exact for $y(x) = x^{m-1}$. Suppose,

$$[x + (n+1)h]^m = \sum_{i=0}^{n} \left[A_i (x + ih)^m + h B_i m (x + ih)^{m-1}\right],$$

is an identity in x and h. Equating coefficients of $x^{m-\alpha} h^{\alpha}$ gives $m + 1$ equations

$$
\left.
\begin{aligned}
&1 = \sum_{i=0}^{n} A_i \qquad \text{for} \quad \alpha = 0, \\
&n+1 = \sum_{i=0}^{n} \left(iA_i + B_i \right) \text{ for } \alpha = 1, \text{ and} \\
&\binom{m}{\alpha} (n+1)^{\alpha} = \sum_{i=1}^{n} \left[\binom{m}{\alpha} i^{\alpha} A_i + \binom{m-1}{\alpha-1} m i^{\alpha-1} B_i \right], \\
&\text{for } \alpha = 2, 3, \ldots, m.
\end{aligned}
\right\} \qquad (2)
$$

Here $\binom{m}{\alpha}$ is the usual binominal coefficient. Applying the rule $\binom{m}{\alpha} = \frac{m}{m-\alpha}\binom{m-1}{\alpha}$ three times to the last equation in (2) gives

$$
\binom{m-1}{\alpha} (n+1)^{\alpha} = \sum_{i=1}^{n} \left[\binom{m-1}{\alpha} i^{\alpha} A_i + \binom{m-2}{\alpha-1} (m-1) i^{\alpha-1} B_i \right],
$$

for $\alpha = 2, 3, \ldots, m-1$. This implies (compare with (2)) that (1) is exact for $y(x) = x^{m-1}$.

Next, observe that if formula (1) is exact for two functions f_1 and f_2, then it is exact for $c_1 f_1 + c_2 f_2$, where c_1, c_2 are arbitrary constants. Hence the exactness of formula (1) for x^m implies its exactness for any linear combination of x^m, x^{m-1}, ..., x^0, and the theorem is proved.

A formula (1) is said to be of <u>order m</u> if it is exact for a polynomial of degree m but not exact for a polynomial of degree m + 1.

In formula (1) the number of "back points" is n + 1. The requirement that the formula be exact for $z = x^{2n+1}$ is met by choosing A_i, B_i (i = 0, ..., n) to satisfy a system of 2n + 2 linear equations.

In order to interpret formula (1) geometrically, suppose that the values of a function z and its derivative z' are known for x_0, $x_0 + h$, ..., $x_0 + nh$. If formula (1) is of order $2n + 1$, then the application has the effect of fitting a polynomial curve of degree $2n + 1$ through the $n + 1$ given points so that at each point the polynomial and the function z have the same derivative. Then y_{n+1} is an approximation to z_{n+1}. This is the well known "Hermite" interpolation [ref. 4, p. 96].

Next are listed some Hermite interpolation formulas. They have the form of formula (1), with order $2n + 1$. In each case the number of back points is $n + 1$. The error terms, given in brackets, are discussed later in this section.

$$y_1 = z_0 + h\, z_0. \quad \left[\frac{h^2}{2} z'' \ (s) \right] \tag{3}$$

$$y_2 = 5z_0 - 4z_1 + h\ (2z_0' + 4z_1'). \quad \left[\frac{h^4}{6} z^{(4)} \ (s) \right] \tag{4}$$

$$y_3 = 10\, z_0 + 9\, z_1 - 18\, z_2 + h\ (3\, z_0' + 18\, z_1' + 9\, z_2'). \quad \left[\frac{h^6}{20} z^{(6)} \ (s) \right] \tag{5}$$

$$3y_4 = 47\, z_0 + 192\, z_1 - 108\, z_2 - 128\, z_3 + h\ (12\, z_0' + 144\, z_1' + 216\, z_2' + 48\, z_3').$$

$$\left[\frac{h^8}{70} z^{(8)} \ (s) \right] \tag{6}$$

$$6y_5 = 131\, z_0 + 1150\, z_1 + 600\, z_2 - 1400\, z_3 - 475\, z_4 + h\ (30\, z_0' + 600\, z_1'$$

$$+ 1800\, z_2' + 1200\, z_3' + 150\, z_4'). \quad \left[\frac{h^{10}}{252} z^{(10)} \ (s) \right] \tag{7}$$

The derivations of the preceding formulas are straightforward. Formula (3) is recognized as a Taylor expansion. For the derivations of the other four, solve equations (2) for A_i, B_i. For example, take $n = 1$ and $m = 3$. Then equation (2) becomes $A_0 + A_1 = 1$, $A_1 + B_0 + B_1 = 2$, $3\, A_1 + 6\, B_1 = 12$, $A_1 + 3\, B_1 = 8$. Hence $A_0 = 5$, $A_1 = -4$, $B_0 = 2$, $B_1 = 4$, and formula (4) is obtained. Formulas (5), (6), (7) are derived similarly.

The justification of the error terms is outlined in Hamming [ref. 4, p. 103]. The general error term is:

$$\frac{[(n+1)!]^2\, h^{2n+2}}{(2n+2)!}\, z^{(2n+2)}(s). \tag{8}$$

The error terms in formulas (3), ..., (7) are found by taking n = 0, ..., 4. These error terms represent the so-called truncation error and should not be confused with roundoff error.

B. THE NUMERICAL SOLUTION OF DIFFERENTIAL EQUATIONS BY PREDICTOR-CORRECTOR METHODS

The literature on the numerical solution of differential equations is vast [ref. 1; 4, p. 403; 5, p. 389]. Consider the differential equation $y' = f(x, y)$ with initial condition $y(x_0) = y_0$. There are various methods for producing points which lie on the solution curve [ref. 1, p. 79; 4, p. 212; 9]. The Runge-Kutta method is widely used. Recently E. B. Shanks has developed similar processes which are markedly more efficient from the standpoint of accuracy and computing time. Shanks' formulas are self-starting, and investigations of the present writers and others show that these formulas are ideally suited for problems in which computing time, accuracy, and error propagation are important.

Predictor-corrector methods have the disadvantage of not being self-starting. Before a predictor-corrector method requiring n back points can be used, a starting process must be applied n - 1 times. (All processes described in this paper use one of Shanks' formulas as a starting procedure.) It is obvious that predictor-corrector processes are harder to program than starting processes. The only motivation in investigating predictor-corrector processes is the possibility of gaining a time advantage in a particular problem or class of problems. The degree of this time advantage is discussed in sections which follow.

Let $(x_0, y_0), \ldots, (x_m, y_m)$ be m + 1 regularly spaced starting points for the equation $y' = f(x, y)$. A formula

$$y_{n+1} = \sum_{i=0}^{m} (A_i\, y_{n-i} + h\, B_i\, y'_{n-i}), \tag{9}$$

can be used repeatedly to produce new points. Of course, at each point the value of the derivative must be computed by using $y'_i = f(x_i, y_i)$. Usually it is wiser to use formula (9) each step to give a tentative or predicted value of y_{n+1}, y'_{n+1} and then to correct the value of y_{n+1} by repeated applications of a formula

$$y_{n+1} = \sum_{i=o}^{m} C_i y_{n-i} + h \sum_{i=-1}^{m} D_i y'_{n-i} . \qquad (10)$$

In this process, formula (9) is called the "predictor" and formula (10) the "corrector." In one step, formula (9) is used only once, but formula (10) may be used more than once. Note that for each application of the corrector, the function $f(x,y)$ must be evaluated. The number of corrector applications per step is a significant factor in any time study.

At first glance, one may be tempted to use only the predictor. This would certainly give an advantage in computing time, but the disadvantage is well-known. Errors in y_i are propagated unfavorably. This is extremely serious and usually prohibits the use of the predictor alone if the solution is to be extended for more than a few steps. For example, the size of the coefficients in the predictor (see formula (5)).

$$y_{n+1} = 10y_{n-2} + 9y_{n-1} - 18y_n + h\left(3y'_{n-2} + 18y'_{n-1} + 9y'_n\right)$$

indicates how unfavorably errors in y_i are propagated.

So we use a predictor-corrector combination. Furthermore, the corrector must be selected with care, or else the unfavorable propagation of error will persist.

C. NUMERICAL STABILITY IN PREDICTOR-CORRECTOR METHODS

Let $(x_0, y_0), \ldots, (x_m, y_m)$ be starting points for $y' = f(x, y)$, and apply the predictor-corrector pair

$$y_{n+1} = \sum_{i=0}^{m} \left(A_i y_{n-i} + h B_i y'_{n-i}\right) , \qquad (11)$$

$$y_{n+1} = \sum_{i=0}^{m} C_i y_{n-i} + h \sum_{i=-1}^{m} D_i y'_{n-i} . \qquad (12)$$

Suppose that in each step the corrector is applied repeatedly until two successive values of y_{n+1} are equal. (Later this will be weakened, and it will be required only that two successive values of y_{n+1} be "close" to each other.. A great deal of computing time will thus be saved.)

Take $z = z(x)$ as the exact solution of $z' = f(x, z)$, and let $\epsilon_i = z_i - y_i$ be the difference between the true solution and the solution given by equations (11), (12). Subtracting equation (12) from

$$z_{n+1} = \sum_{i=0}^{m} C_i z_{n-i} + h \sum_{i=-1}^{m} D_i z'_{n-i} + R,$$

gives

$$\epsilon_{n+1} = \sum_{i=0}^{m} C_i \epsilon_{n-i} + h \sum_{i=-1}^{m} D_i \epsilon'_{n-i} + R \ . \tag{13}$$

By the mean value theorem, there is a number t between y_i and z_i such that

$$\epsilon'_i = \frac{\partial f}{\partial y} (x_i, t) \epsilon_i \ , \tag{14}$$

since $\epsilon'_i = f(x_i, z_i) - f(x_i, y_i)$. We follow the usual procedure [ref. 4, p. 197] in assuming that $h\frac{\partial f}{\partial y} = K$ and R are constants. Then $h \epsilon'_i = K\epsilon_i$, and this with equation (13) gives the nonhomogeneous difference equation

$$\sum_{i=-1}^{m} (C_i + KD_i) \epsilon_{n-i} + R = 0 \tag{15}$$

in ϵ_{n-i}, with $C_{-1} = -1$.

If the roots $\lambda_0, \ldots, \lambda_m$ of the "characteristic" equation

$$\sum_{i=-1}^{m} (C_i + K D_i) \lambda^{m-i} = 0 \tag{16}$$

are distinct, then the general solution $(\epsilon_{n+1}, \ldots, \epsilon_{n-m})$ of equation (15) is given by

$$\epsilon_k = \sum_{i=0}^{m} \alpha_i \lambda_i^k + r_k \quad , \tag{17}$$

where $(r_{n+1}, \ldots, r_{n-m})$ is a particular solution of equation (15).

From equation (17) it is seen that it is desirable to have the characteristic roots $\lambda_0, \ldots, \lambda_m$ less than 1 in absolute value so that the propagated error will be favorable. (This last statement is not changed even if the characteristic roots are not all distinct. For a discussion see [ref. 5, p. 213].)

If $h = 0$, then $K = 0$ and one characteristic root is 1. For small $h \neq 0$, one characteristic root will be close to 1, but we require that the other characteristic roots be within the complex unit circle. It is impossible to predict in all situations what the range of values of K will be. Usually $|K|$ is kept smaller than 0.4 [ref. 3, p. 7; 4, p. 198].

We will use only correctors such that, for $h = 0$, the characteristic roots (except 1) are within the unit circle. There are many such formulas, but unless an investigation concerning the characteristic roots for $h \neq 0$ is made, Adams' formulas appear safer from the stability standpoint. In some instances, there are more efficient formulas.

For $h \approx 0$ (h approximately equal to 0), Fehlberg [ref. 3, p. 5] has given a method for comparing the error propagation of two correctors. Since the authors have found this result of great use, it is reproduced here. Set $h = 0$ in equation (13) to get

$$\epsilon_{n+1} = \sum_{i=0}^{m} C_i \epsilon_{n-i} + C\, h^{m+3}\, y^{(m+3)}(s) \quad , \tag{18}$$

and suppose there is a constant E defined by

$$\epsilon_k = E\, k\, h^{m+3}\, y^{(m+3)}(s) \quad . \tag{19}$$

This is the assumption that the errors ϵ_k are almost in arithmetic progression. Now equations (18), (19) give

$$E(n+1) = \sum_{i=0}^{m} C_i \, E(n-i) + C,$$

if $y^{(m+3)}$ is assumed constant. Since $\sum_{i=0}^{m} C_i = 1$, we have

$$E = \frac{C}{1 + \sum_{i=1}^{m} i \, C_i} \quad . \tag{20}$$

If two correctors are used on the same differential equation, the long-run propagated truncation errors will be proportional to the corresponding values of E, provided the correctors are stable. The result is of great use in choosing a corrector.

D. SOME SPECIFIC CORRECTORS

Correctors of orders 3, 5, 7, 9 are presented in this section. Each corrector is a linear combination of formulas of types (4) through (7).

First we obtain correctors of order 3. Formula (4) gives:

$$y_2 = 5y_0 - 4y_1 + h\,(2y_0' + 4y_1') + \frac{h^4}{6} y^{(4)}(s_1) \quad , \tag{21}$$

and reversing the order of the points (x_0, y_0), (x_1, y_1), (x_2, y_2) gives:

$$y_0 = 5y_2 - 4y_1 - h\,(2y_2' + 4y_1') + \frac{h^4}{6} y^{(4)}(s_2) \quad . \tag{22}$$

A general formula for y_2 is obtained by multiplying equations (21), (22) by $5b + 1$, b, respectively, and then adding. The result is:

$$y_2 = (-24b - 4)\, y_1 + (24b + 5)\, y_0 + h\,[-2b\, y_2' + (16b + 4)\, y_1' + (10b + 2)\, y_0']$$

$$+ C\, h^4\, y^{(4)}\ (s)\ , \qquad (23)$$

where $C \approx b + \frac{1}{6}$, and s is between x_0 and x_2. For h = 0, the characteristic roots of equation (23) are 1, -24b - 5. For numerical stability we require that $-\frac{1}{4} < b < -\frac{1}{6}$. Hence we have an infinite number of acceptable correctors. If we ignore the effect of roundoff error, the choice of a corrector (in this case, the choice of b) depends on two things: (1) the size of the step, or local, truncation error, $C h^4 y^{(4)}\ (s)$; (2) the propagated truncation error. It is to be expected that there is no best formula for all problems.

At one end, $b = -\frac{1}{6}$, of the stability interval, equation (23) becomes Simpson's rule. At $b = -\frac{1}{4}$, the result is

$$y_2 = 2y_1 - y_0 + h\,(\tfrac{1}{2}\, y_2' - \tfrac{1}{2}\, y_0')\ .$$

Simpson's rule has an attractive error term, $-\frac{1}{90}\, h^5\, y^{(5)}\ (s)$, but is likely to be unstable [ref. 2, p. 37].

For $b = -\frac{5}{24}$, the midpoint of the stability interval, equation (23) becomes Adams' formula

$$y_2 = y_1 + \frac{h}{12}\ (5y_2' + 8y_1' - y_0')\ ,\quad (C = -\tfrac{1}{12})\ . \qquad (24)$$

For $b = -\frac{1}{5}$, the harmonic mean of $-\frac{1}{4}$ and $-\frac{1}{6}$, equation (23) becomes

$$y_2 = \frac{4}{5}\, y_1 + \frac{1}{5}\, y_0 + \frac{h}{5}\ (2y_2' + 4y_1')\ ,\quad (C = -\tfrac{1}{30})\ . \qquad (25)$$

Computation of the ratio of E-values of equations (24), (25) indicates that equation (25) will have the more favorable propagation of error. Numerical examples support this conclusion [ref. 3, p. 9].

We investigated one other formula from the infinite number of third-order formulas. Taking $b = -\frac{109}{600}$ in equation (23) gives

$$y_2 = \frac{9}{25}\, y_1 + \frac{16}{25}\, y_2 + \frac{h}{300}\ (109\, y_2' + 328\, y_1' + 55\, y_0')$$

$$(C \approx -0.015)\ . \tag{26}$$

The error propagated by formula (26) is about one-third of the error propagated by equation (25). This agrees with the ratio of the E-values.

Similar formulas of orders 5, 7, and 9 are now given. For the derivations, see Appendix I.

$$y_4 = \frac{1}{16}\, y_3 + \frac{1}{8}\, y_2 + \frac{1}{4}\, y_1 + \frac{9}{16}\, y_0$$

$$+ \frac{h}{11,520}\ (3703\, y_4' + 15,518\, y_3' + 6168\, y_2' + 10,898\, y_1' + 1873\, y_0')$$

$$+ C\, h^6\, y^{(6)}\ (s)\ , \tag{27}$$

where $C \approx -0.00725$. For $h = 0$, the characteristic roots are approximately 1, -0.800, $-0.069 \pm 0.836\, i$. The largest absolute value of a root is about 0.84. A glance at the y-coefficients shows that they are nearly in geometric progression.

A similar formula of order 7 is

$$y_6 = \frac{1}{64}\, y_5 + \frac{1}{32}\, y_4 + \frac{1}{16}\, y_3 + \frac{1}{8}\, y_2 + \frac{1}{4}\, y_1 + \frac{33}{64}\, y_0$$

$$+ \frac{h}{430,080}\ (128,627\, y_6' + 642,168\, y_5' + 130,167\, y_4'$$

$$+ 693,632\, y_3' + 142,137\, y_2' + 399,240\, y_1' + 61,469\, y_0')$$

$$+ C\, h^8\, y^{(8)}\ (s)\ , \tag{28}$$

where $C \approx -0.00497$. For $h = 0$, the characteristic roots have absolute values approximately 1, 0.85, 0.91, 0.91, 0.86, 0.86.

A formula of order 9 is:

$$2,560,016\, y_8 = 9784\, y_7 + 20,133\, y_6 + 41,040\, y_5$$

$$+ 79,775\, y_4 + 159,816\, y_3 + 319,691\, y_2 + 639,792\, y_1$$

$$+ 1,289,985\, y_0 + h\,(725,340\, y'_8 + 4,150,740\, y'_7$$

$$- 280,710\, y'_6 + 6,541,620\, y'_5 - 1,808,250\, y'_4$$

$$+ 5,630,940\, y'_3 + 244,290\, y'_2 + 2,458,620\, y'_1$$

$$+ 345,330\, y'_0) + C\, h^{10}\, y^{(10)}\,(s)\ , \qquad (29)$$

where $C \approx -0.00361$. The absolute values of the characteristic roots (for $h = 0$) are approximately 1, 0.88, 0.80, 0.80, 0.90, 0.90, 0.95, 0.95.

Formulas (26), (27), (28), (29) are examples of formulas with low truncation error terms and a reasonably stable behavior. These formulas, as well as Adams' formulas, are used to test and illustrate programming techniques in section III. Unless special circumstances arise, the authors prefer Adams' formulas because of their excellent stability behavior [ref. 8, p. 29].

E. CHOICE OF A METHOD

In case a great deal of computing time is at stake, a well-chosen predictor-corrector process is likely to be better than a starting process alone. Of course, the former process is harder to program, but it can save time. If computing time is not critical, then starting methods, especially those developed by Shanks, are recommended. Also, it should be remarked that Shanks' formulas seem to be applicable to a large class of equations at large step sizes, without loss of stability. This is an advantage not shared by all predictor-corrector processes.

If a predictor-corrector is used, the appropriate characteristic roots should be within the unit circle. If an estimate of K (see section II - C) is available, the characteristic roots should be examined further. If no such estimate is available, the appropriate Adams formula is recommended. However, it will be shown in sections which follow that in many cases other correctors are better.

The corrector, rather than the predictor, controls the error and stability behavior of these processes. A fifth-order predictor (formula (5), section II) is used in nearly all of the processes reported in this paper. A few other predictors were tried, but they proved inferior to the one just mentioned. However, our experiments with different predictors were few, and it may well be that a small gain in computing efficiency would follow a more careful matching of predictor with corrector.

This report does not include a comparison with methods designed specifically for second (or higher) order differential equations. The processes presented are used to solve second-order equations in the traditional way, that is, by solving two first-order equations. It is emphasized that the final program presented can be used to solve up to twenty first-order equations, or up to ten second-order equations, etc.

SECTION III. PROGRAMMING PREDICTOR-CORRECTOR METHODS

A. INTRODUCTORY REMARKS

Predictor-corrector methods usually require more than one back point. Consequently, a starter (starting process) is used to produce, from an initial point, the additional values needed for the predictor-corrector.

Special mention is now made concerning programming methods. All programs were done by an IBM 7072. Fortran, with certain modifications, was used. It has been observed that roundoff error can affect the last four digits on runs of fewer than 2,000 steps. Hence, single precision (8-digit accuracy) does not allow a genuine comparison of numerical methods, and double precision (16 digits) is used. Double precision statements at the Vanderbilt computer are in modified subroutine form. In a Fortran program, these subroutines are called by the statements

```
R = FMDF (A1, A2, B1, B2, C1, C2)
R = FADF (A1, A2, B1, B2, C1, C2)
R = FDDF (A1, A2, B1, B2, C1, C2)
```

These in turn produce the following:

```
C1 + C2 = (A1 + A2) (B1 + B2)
C1 + C2 = (A1 + A2) + (B1 + B2)
C1 + C2 = (A1 + A2) / (B1 + B2) .
```

In each case A1, B1, C1 are the first eight digits and A2, B2, C2 the second eight digits of A1 + A2, B1 + B2, C1 + C2, respectively. To illustrate, if A1 + A2 = 0.12234 55678 88999 9 $\times$ 10^{15}, then A1 = 0.12234 556 $\times$ 10^{15} and A2 = 0.78889 999 $\times$ 10^{7}. Double precision programming with these subroutines is quite cumbersome. Each multiplication, addition, or division must be programmed as a statement of the types represented above. This means the programming of a number of predictor-corrector methods would be greatly delayed if each predictor-corrector were matched with a different starter. However, in any single process, it is desirable to match the starter with the predictor-corrector so that they have the same order.

In the studies herein presented, the following starter by Shanks [ref. 9, p. 7] was used.

$$k_0 = h\ f_r,$$

$$k_1 = h\ f(x_r + \tfrac{1}{3}h,\ y_r + \tfrac{1}{3}\ k_0)\ ,$$

$$k_2 = h\ f[x_r + \tfrac{1}{2}h,\ y_r + \tfrac{1}{8}\ (k_0 + 3\ k_1)]\ ,$$

$$k_3 = h\ f[x_r + \tfrac{2}{3}h,\ y_r + \tfrac{1}{27}\ (4\ k_0 + 6\ k_1 + 8\ k_2)]\ ,$$

$$k_4 = h\ f[x_r + \tfrac{1}{3}h,\ y_r + \tfrac{1}{108}\ (17\ k_0 + 12\ k_1 + 16\ k_2 - 9\ k_3)]\ ,$$

$$k_5 = h\ f[x_r + \tfrac{1}{3}h,\ y_r + \tfrac{1}{108}\ (11\ k_0 + 12\ k_1 - 32\ k_2 + 9\ k_3 + 36\ k_4)]\ ,$$

$$k_6 = h\ f[x_r + h,\ y_r + \tfrac{1}{44}\ (-5\ k_0 - 12\ k_1 - 128\ k_2 + 81\ k_3 - 108\ k_4 + 216\ k_5)]\ ,$$

$$y_{r+1} = y_r + \frac{1}{120}\ (11\ k_0 - 64\ k_2 + 81\ k_3 + 81\ k_5 + 11\ k_6)\ . \qquad (30)$$

This starter is of order six (the error is of order $h^7\ y^{(7)}\ (s)$, where s is between x_r and x_{r+1}). Because many of the predictor-corrector methods were of orders higher than six, additional accuracy was obtained for the initial values by using a smaller step size for the starting process than for the continuing process. For example, if five initial values at h = 0.01 are needed for a seventh order predictor-corrector, these can be produced as follows. From the given first point, produce eight new points at h = 0.005. Denote the even numbered new points as (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), and denote the odd numbered new points as $(x_{\frac{1}{2}}, y_{\frac{1}{2}})$, $(x_{1\frac{1}{2}}, y_{1\frac{1}{2}})$, etc. The error from (x_i, y_i) to $(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}})$ is $C\ 2^{-7}\ h^7\ y^{(7)}\ (s)$. Assuming that the error builds up arithmetically, we have $C\ 2^{-6} h^7\ y^{(7)}\ (s)$ as the error from (x_i, y_i) to (x_{i+1}, y_{i+1}). For h = 0.01 this error is approximately $1.5625\ C\ y^{(7)}\ (s) \times 10^{-16}$. Thus we have points of sufficient accuracy for a seventh order continuing process. Unfortunately, the technique does take additional time.

Another part of the program that is peculiar to our needs is that all coefficients in both predictor and corrector are in the form of data. Special branching is needed to generalize the usual programming. Such branching takes time.

In the remainder of this section, we discuss the remaining sections and what we hope to accomplish. There are two basic ways of programming predictor-corrector methods. In one, the corrector is applied repeatedly at each step until two consecutive computed values are identical. In the second basic technique, the corrector is applied a specific number of times per step. In section B, the problem of these two techniques are presented and illustrated. An appropriate convergence technique and variations thereof are presented in section C. The technique in section C saves considerable time over the technique of "complete convergence" in section B. Considerably more time can be saved, however, under the assumption that if convergence is approximate at one step, then it is approximate at another step. A program based on this assumption gave our best results. This, our final program, is presented in section D.

For easy reference, we now list all computing processes and differential equations to be discussed in the remaining sections. The processes are called P1 through P6, and the equations are called E1 through E4. In each process the predictor (see formula (5))

$$y_{k+3} = -18y_{k+2} + 9y_{k+1} + 10y_k + h\,(9y'_{k+2} + 18y'_{k+1} + 3y'_k)$$

is used.

P1 is the process in which Shanks' starting formula (30) is used as a continuing process.

The other process are distinguished by their correctors, listed below. P3 and P5 are by Adams. The source of the others is given in Appendix I. (The symbols y_i, y'_i are omitted in the right members of the following. The indices are in descending order.)

P2. $$y_{k+4} = \frac{1}{16}(1 + 2 + 4 + 9) + \frac{h}{11,520}(3703 + 15,518 + 6168 + 10,898 + 1873)$$

$$\tilde{+}\,\frac{167}{23,040}\,h^6\,y^{(6)}\ (s).$$

P3. $$y_{k+6} = (1 + 0 + 0 + 0 + 0 + 0) + \frac{h}{60,480}(19,087 + 65,112 - 46,461 + 37,504$$

$$- 20,211 + 6312 - 863) + \frac{275}{24,192}\,h^8\,y^{(8)}\ (s)\ .$$

P4. $y_{k+6} = \frac{1}{64}(1 + 2 + 4 + 8 + 16 + 33) + \frac{h}{430,080}(128,627 + 642,168$

$+ 130,167 + 693,632 + 142,137 + 399,240 + 61,469) \tilde{+} \frac{285}{57,344} h^8 y^{(8)}$ (s) .

P5. $y_{k+8} = (1 + 0 + 0 + 0 + 0 + 0 + 0 + 0) + \frac{1}{3,628,800}(1,111,267 + 4,137,094$

$- 3,449,594 + 3,285,358 - 2,145,620 + 836,338 - 136,214 - 17,126$

$+ 7297) \tilde{+} (-.00936) h^9 y^{(9)}$ (s). (This is a modified Adams corrector.)

P6. $y_{k+8} = \frac{1}{2,560,016}(9784 + 20,133 + 41,040 + 79,775 + 159,816 + 319,691$

$+ 639,792 + 1,289,985) + \frac{h}{2,560,016}(725,340 + 4,150,740 - 280,710$

$+ 6,541,620 - 1,808,250 + 5,630,940 + 244,290 + 2,458,620 + 345,330)$

$\tilde{+} (-0.00361) h^{10} y^{(10)}$ (s).

The equations to be solved numerically are given next. Included are the initial values, the interval of the solution, and the correct solution.

E1. $y' = y$. Initial value $y(0) = 1$. Interval of solution $x = 0$ to $x = 18$. Correct solution $y = e^x$, $y(18) = 65659\ 969.13733\ 05111\ 38786$.

E2. $y'' = -\frac{xy' + y}{(xy)^2}$. Initial values $y(1) = y'(1) = 1$. Interval $x = 1$ to $x = 19$. Correct solution $y = \sqrt{1 + 2 \log x}$, $y' = \frac{1}{xy}$, $y(19) = 2.62466\ 72090\ 63443$, $y'(19) = 0.02005\ 26675\ 40335\ 10$.

E3. $y' = -y$. Initial value $y(0) = 1$. Interval $x = 0$ to $x = 18$. Correct solution $y = e^{-x}$, $y(18) = 0.15229\ 97974\ 47126\ 284 \times 10^{-7}$.

E4. $y' = -2 x y^2$. Initial value $y(0) = 1$. Interval $x = 0$ to $x = 18$. Correct solution $y = (1 + x^2)^{-1}$, $y(18) = 0.00307\ 69230\ 76923\ 07692\ 31$.

Future reference to the preceding is made according to the labels. For example, P2 - E1 means the process P2 used to solve equation E1.

The equation E2 is solved by solving the pair $z' = -\frac{xz + y}{(xy)^2}$, $y' = z$.

B. THE USUAL PREDICTOR-CORRECTOR TECHNIQUES

Both of the two basic techniques for programming predictor-corrector methods have definite problems. In solving differential equations, our goal is always to obtain a given accuracy with the smallest computing time. Because elegant starting processes exist, it becomes essential to find an efficient and dependable predictor-corrector program to compete effectively with these starting methods. Neither of the common methods is both efficient and dependable.

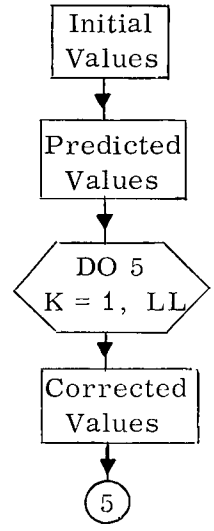


FIGURE 1. USING A SPECIFIC NUMBER OF CORRECTIONS

Figure 1 is an abbreviated flow chart for one step of a predictor-corrector method. (LL is the number of corrections desired.) Such a technique can be quite efficient as far as time alone is concerned. However, the method lacks predictability. If relative error builds up arithmetically (normally it appears to do so), it is not apparent what this error should be. At any given step, actual convergence may not occur until the tenth correction or later. Error approximations that arise from Taylor's series are based on actual convergence. As Table I shows, errors may change a great deal as the number (LL) of applications of the corrector per step is varied.

TABLE 1. ERRORS FOR DIFFERENT NUMBERS OF CORRECTIONS. P4 - E1. STARTERS PRODUCED AT ONE-HALF h

		h	
		0.08	0.20
Number	1	0.1498×10^{-3}	0.2976×10^{-1}
of	2	0.4263×10^{-6}	0.2252×10^{-3}
Corrections	3	0.4462×10^{-5}	0.2592×10^{-2}

The unpredictability of this technique makes it impractical in general.

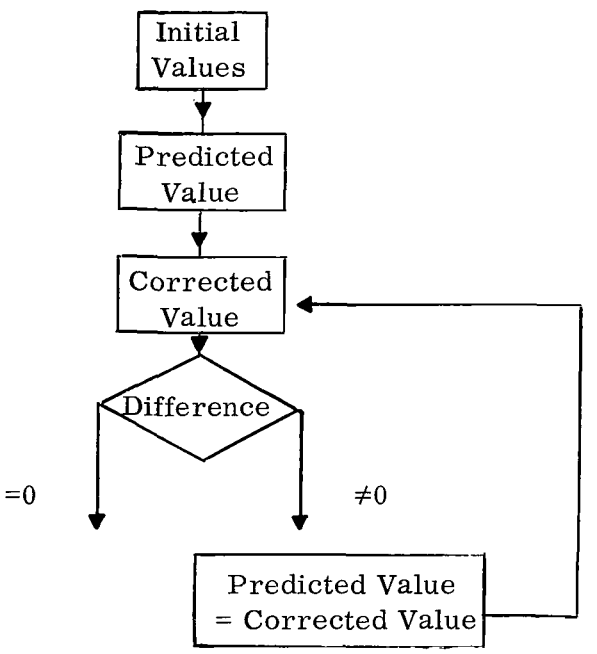


FIGURE 2. CONVERGENCE AT EACH STEP

The abbreviated flow chart of Figure 2 illustrates another technique. As we have mentioned, the error terms which are calculated from Taylor's series are valid. Hence, this technique is predictable as to error. For this reason, this technique is used very often. The drawback is its lack of efficiency. As shown in Table 2, considerable time is used to "converge to zero" in many cases. Hence, this method is impractical.

Note: By "complete convergence" or "convergence to zero" we mean application of the corrector each step until two successive values of y_i are identical.

TABLE 2. P4-E1. STARTERS PRODUCED AT ONE-HALF h

	4 Corrections Per Step		Complete Convergence	
h	Error	Time (min)	Error	Time (min)
0.15	2.015	0.2920	2.019	0.4072
0.18	7.196	0.2477	7.215	0.3583
0.20	14.99	0.2261	15.04	0.3345
0.24	53.11	0.1931	53.36	0.2964
0.30	246.9	0.1603	248.8	0.2561

C. APPROXIMATE ERROR RATIO CONVERGENCE

Consider the seventh order process P4. In this, the error term E_{i+1} is less than or equal to $(285/57,344)h^8 y^{(8)}(s)$, where s lies between x_i and x_{i+1}. Normally, $y^{(8)}(s)$ is not known because $y^{(8)}(x)$ is seldom known and s is not predictible. However, it may be possible to approximate E_{i+1} with some value E'_{i+1} calculated from what is known. Let $(y_{i+1})_j$ and $(y_{i+1})_{j+1}$ be two successive approximations of y_{i+1} with a given corrector. We make the following definition of variations on convergence.

Definition 1. $(y_{i+1})_n$ converges to $(y_{i+1})_j$ within a ratio r of the error if $\left| (y_{i+1})_j - (y_{i+1})_{j+1} \right| \leq \left| r\, E_{i+1} \right|$.

Definition 2. $(y_{i+1})_n$ converges to $(y_{i+1})_j$ within a ratio r of the approximate error if $\left| (y_{i+1})_j - (y_{i+1})_{j+1} \right| \leq \left| r\, E'_{i+1} \right|$.

Let a given predictor-corrector technique require n back points. Define, for $i = n$, $\bar{y}_{i+1} = (y_{i+1})_j$, where $(y_{i+1})_n \rightarrow (y_{i+1})_j$ within a ratio r of the error E_{i+1}. By using this approximation of y_{i+1} in place of y_{i+1}, we can calculate $\bar{y}_{i+2}$. Continuing the process inductively, we can approximate the solution given by the technique requiring complete convergence at each step. This new process will be called "error ratio convergence." If, instead of Definition 1, we use Definition 2 in approximating y_{i+1}, we will call the process of solving the equation "approximate error ratio convergence." This process has the following flow chart for one step.

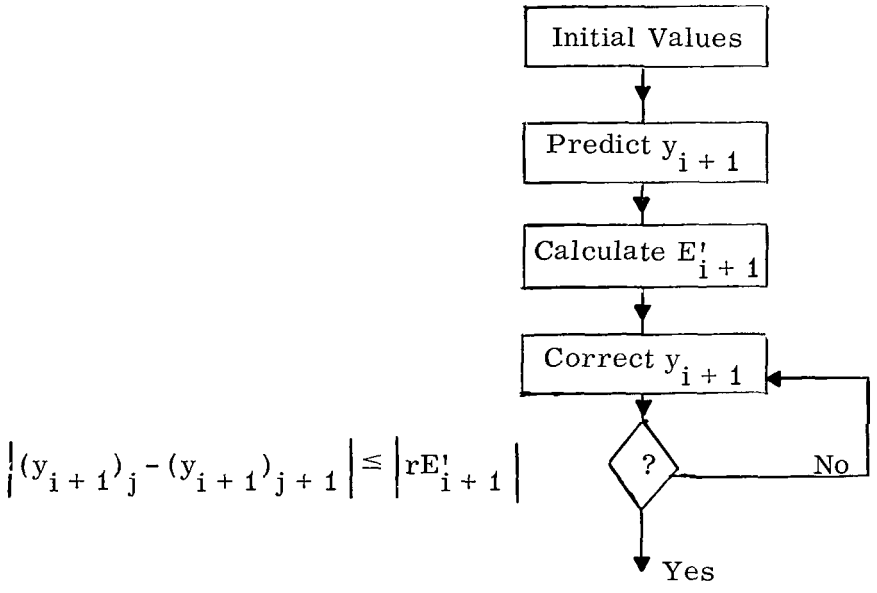


FIGURE 3. APPROXIMATE ERROR RATIO CONVERGENCE

We now give a useful way of obtaining E'. For the present, assume that the y_i, y'_i are exact. For $y = y(x)$, define backward differences $\nabla y_i = y_i - y_{i-1}$, and $\nabla^{j+1} y_i = \nabla^j y_i - \nabla^j y_{i-1}$, in the usual manner. Use $h^{-n}\nabla^n y_i$ as an approximation for $y^{(n)}(x_i)$ [ref. 6, p. 128]. Normally, an n-th order process requires n-1 back points and has an error E_i of the form

$$E_i = C\, h^{n+1}\, y^{(n+1)}(s_i) \quad \text{for } s_i \text{ in } (x_{i-n+1}, x_i).$$

We can then define E_i' by:

$$E_i' = C\,h\,\nabla^n\,\bar{y}_i' \qquad \text{for } i \geq n,$$

where the exact values y_i' are replaced by the approximate values $\bar{y}_i'$, which are calculated by the process being used. This definition of E' is quite useful.

For an n-th order process, the calculation of E' requires n back points, one more than would be needed without this calculation. In all of the following sections E_i' is calculated by using $\bar{y}_j'$ for $j < i$, and the predicted value of y_i' in place of $\bar{y}_i'$.

We will use E_i' in three ways. First E_{i+1}' can be calculated at each step with $\bar{y}_{i+1}$ defined as $(y_{i+1})_j$, where $(y_{i+1})_n \rightarrow (y_{i+1})_j$ within a ratio r of E'. This calculation of E_i' at each step does consume needless time, as we will demonstrate. In Table 3 we see the result of using one process with E_i' calculated at each step. (r is taken to be 0.04.)

TABLE 3. P4 - E1. STARTERS PRODUCED AT ONE-HALF h

	Approximate Error Ratio Convergence r = 0.04		Complete Convergence Each Step	
h	Error	Time (min)	Error	Time (min)
0.12	0.4232	0.4989	0.4232	0.4770
0.15	2.018	0.4039	2.019	0.4072
0.18	7.214	0.3397	7.215	0.3583
0.20	15.03	0.3083	15.04	0.3345
0.24	53.34	0.2600	53.36	0.2964
0.30	248.6	0.2125	248.8	0.2561

It is clear that the calculation of E_i' does require some time. Also, an additional correction is used needlessly to show that $(y_{i+1})_n \rightarrow (y_{i+1})_j$ within a ratio r of E_{i+1}'. For example, if $(y_{i+1})_n \rightarrow (y_{i+1})_5$ within a ratio r of E_{i+1}', we must find $(y_{i+1})_6$ to

show that Definition (2) is satisfied. Normally, if $(y_{i+1})_n \rightarrow (y_{i+1})_5$ within a ratio r of E'_{i+1}, then $(y_{i+1})_n \rightarrow (y_{i+1})_6$ within a ratio r of E'_{i+1}.

It is possible to reduce this needless waste of time by further changes in the program. One change is given in this section, another in the next section.

Define F_i, F'_i by

$$F_i = (y_i E_1)/y_1 ,$$

$$F'_i = (y_i E'_1)/y_1 .$$

If we use F'_i instead of E'_i in the technique just described, additional time is saved. Essentially, the use of F_i means that we are taking the relative error at the first step as an approximation for the relative error at the i-th step. The use of F'_i is similarly interpreted.

The following Table 4 shows the time gained by using F'_i. It is emphasized that the values of y_i produced by the two programs are identical.

TABLE 4. P4 - E1. STARTERS PRODUCED AT ONE-HALF h. r = 0.04

h	Computing Time in Min	
	Using E'_i	Using F'_i
0.12	0.4989	0.4058
0.15	0.4039	0.3303
0.18	0.3397	0.2794
0.20	0.3083	0.2541
0.24	0.2600	0.2161
0.30	0.2125	0.1784

As before, it is still necessary to correct each step an additional time in order to make the decision that enough corrections have been made. This will be remedied in the next section where the technique is modified again. In the new program, errors differ only slightly from those obtained with the preceding technique.

D. THE FINAL PROGRAM--AN ASSUMED APPROXIMATE ERROR RATIO CONVERGENCE

The final program involves the additional, though trustworthy, assumption that if $(y_j)_n \rightarrow (y_j)_m$ within a ratio r of the approximate error E'_j, then, for all i, $(y_i)_n \rightarrow (y_i)_m$ within a ratio r of the approximate error E'_i or else the deviation is so slight as to be insignificant.

This means that $(y_i)_{m+1}$ needs to be computed only for the first step in which the predictor-corrector is used. At the first step, the number of corrections is computed. From that point on, the number of corrections per step is constant. This gives our final program, an "assumed approximate error ratio convergence" technique. Its flow chart is shown in figure 4.

In tables 5 and 6 we see not only the time saved, but also the change in error, when the final program is used in place of the one which requires complete convergence at each step. Comparison of table 6 with table 4 shows the time advantage of the final program over the two earlier methods. In tables 5 and 6, the "percent change in error" entries should not be misinterpreted. For example, the reading 2.17 in table 5 comes from comparing two relative errors, each of which is approximately 10^{-6}.

In one instance, the time for the complete convergence process exceeded fourteen minutes, while the time for the final program was 0.5672 minutes. This however, was unusual. Tables 5 and 6 represent typical time advantages.

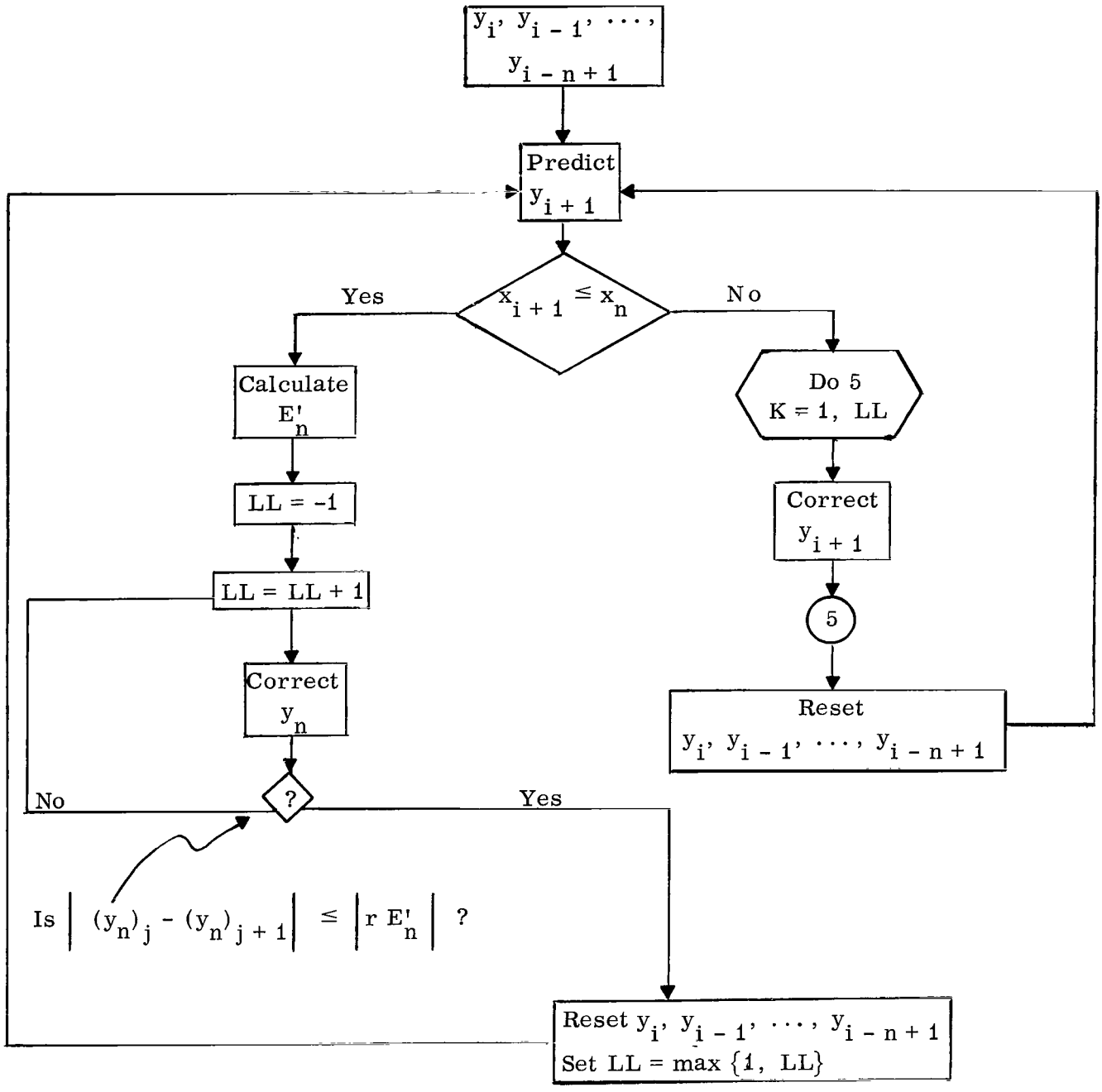


FIGURE 4. THE FINAL PROGRAM

TABLE 5. P2 - E1. r = 0.08.

h	Complete Convergence		Final Program		Comparison	
	Error	Time (min)	Error	Time (min)	Percent Change in Error	Percent Time Saved
0.12	67.65	0.4331	66.18	0.2464	2.17	43.11
0.15	208.2	0.3703	201.3	0.2003	3.31	45.91
0.18	521.8	0.3222	497.6	0.1692	4.64	47.49
0.20	887.1	0.3033	837.3	0.1539	5.61	49.26
0.24	2220	0.2678	2048	0.1305	7.75	51.27
0.30	6805	0.2314	6726	0.1180	1.16	49.01

TABLE 6. P4 - E1. r = 0.04.

h	Complete Convergence		Final Program		Comparison	
	Error	Time (min)	Error	Time (min)	Percent Change in Error	Percent Time Saved
0.12	0.4232	0.4770	0.4227	0.3570	0.11	25.16
0.15	2.019	0.4072	2.015	0.2920	0.19	28.29
0.18	7.215	0.3583	7.196	0.2477	0.26	30.87
0.20	15.04	0.3345	14.99	0.2261	0.33	32.41
0.24	53.36	0.2964	53.11	0.1931	0.46	34.85
0.30	248.8	0.2561	246.9	0.1603	0.76	37.41

E. A NOTE ON MATCHING STARTER WITH PREDICTOR-CORRECTOR

One of Shanks' sixth order starters, equation (30), was used with every process reported in this paper. Whenever the corrector was of higher order than the starter, the step size for the starter was taken as a fraction of that of the continuing process. For correctors of order seven, the starters were produced at half the regular step size (see section III - A). Similarly, starters for a ninth-order corrector were produced at a fifth of the regular step size. This was an obvious waste of time for large step sizes.

The error of the starter is $C_1 h^7 y^{(7)} (s_1)$, while the error of an n-th order corrector is $C_2 h^{n+1} y^{(n+1)} (s_2)$. For $h = 0.01$ and $n = 7$ it is sufficient to compute the starting points at half the regular step size. The results for a ninth-order corrector at step sizes 0.01 and 0.02 are confused by roundoff error, even though double precision is used. For $h = 0.03$ and $n = 9$, corrector error is $5.9049\ (10^{-16})\ C_2\ y^{(10)}\ (s_2)$. For starters produced at h/5, the error is approximately $5(2.79936)\ (10^{-16})\ C_1\ y^{(7)}\ (s_1)$ or $1.39968\ (10^{-15})\ C_1\ y^{(7)}\ (s_1)$. The use of an eighth-order starter would reduce calculation time considerably.

For larger step sizes the starters may be produced more quickly. If $h = 0.3$ and $n = 9$, the corrector error is $5.9049\ (10^{-6})\ C_2\ y^{(10)}\ (s_2)$. If the starters are produced at h/2, then the starter error is approximately $2(1.7)\ (10^{-6})\ C_1\ y^{(7)}\ (s_1)$.

It may be that using the starting process for so many steps is unnecessary in some cases, as the following table indicates. This particular aspect deserves attention in future studies. The question, "How do inaccuracies in initial values affect the final results?" would be pertinent to such a study.

The approximate values in Table 8 are useful in estimating the time wasted in producing initial values at a fraction of the regular step size.

TABLE 7. P5 - E3

h	Starters Produced at 1/5 h		Starters Produced at h	
	Error	Time (min)	Error	Time (min)
0.03	0.1606×10^{-19}	1.2011	0.1325×10^{-19}	1.1164
0.06	0.5402×10^{-18}	0.6514	0.3014×10^{-18}	0.5667
0.09	0.1478×10^{-16}	0.4683	0.1058×10^{-16}	0.3834
0.12	0.1791×10^{-15}	0.4037	0.1467×10^{-15}	0.3189
0.18	0.5315×10^{-14}	0.3028	0.4729×10^{-14}	0.2181
0.24	0.6150×10^{-13}	0.2520	0.5685×10^{-13}	0.1675

TABLE 8. TIME USED FOR PRODUCING STARTERS AT A FRACTION OF THE REGULAR STEP SIZE

Equation	Time (min) for One Starter Application	Time Wasted in Program	
		Seventh-order. Starters Produced at 1/2 h	Ninth-order. Starters Produced at h/5
E1	0.0026668	0.0160	0.0853
E2	0.006342	0.0381	0.2029
E3	0.0026743	0.0160	0.0856
E4	0.0035717	0.0214	0.1143

SECTION IV. CONCLUSIONS

Our results indicate that in a large number of cases predictor-corrector processes may compete successfully with starting methods. However, unless care is taken to reduce the number of wasted applications of the corrector, then the starting process alone is more efficient.

The final program given in section III - D demonstrates clearly that predictor-corrector processes can save significant time when compared with very efficient starting processes.

There is no single corrector (of a given order) which is the best one for all differential equations. Two correctors may compete differently when applied to distinct equations or when used at different step sizes. The process with the smaller error term may be efficient and accurate in one situation, but unstable in another.

If the number K of section II - C can be approximated in advance, it is possible to design a corrector specifically for a given problem.

APPENDIX I. DERIVATION OF FORMULAS

The derivations of formulas (27), (28), and (29) are outlined:

Equation (27). From section I, equation (5),

$$y_3 = 10\,y_0 + 9\,y_1 - 18\,y_2 + h\,(3\,y_0' + 18\,y_1' + 9\,y_2') + \frac{h^6}{10}\,y^{(6)}\ (s_1),$$

$$y_0 = 10\,y_3 + 9\,y_2 - 18\,y_1 - h\,(3\,y_3' + 18\,y_2' + 9\,y_1') + \frac{h^6}{10}\,y^{(6)}\ (s_2),$$

$$y_4 = 10\,y_1 + 9\,y_2 - 18\,y_3 + h\,(3\,y_1' + 18\,y_2' + 9\,y_3') + \frac{h^6}{10}\,y^{(6)}\ (s_3),$$

$$y_1 = 10\,y_4 + 9\,y_3 - 18\,y_2 - h\,(3\,y_4' + 18\,y_3' + 9\,y_2') + \frac{h^6}{10}\,y^{(6)}\ (s_4).$$

Multiply the preceding equations in turn by 1873, -710, -2470, -3703, and then add. This leads to (27).

Equation (28). From section I, equation (6),

$$3\,y_4 = 47\,y_0 + 192\,y_1 - 108\,y_2 - 128\,y_3 + h\,(12\,y_0' + 144\,y_1' + 216\,y_2' + 48\,y_3'),$$

$$3\,y_0 = 47\,y_4 + 192\,y_3 - \ldots \qquad - h\,(12\,y_4' + \ldots\ ),$$

$$3\,y_5 = 47\,y_1 + 192\,y_2 - \ldots \qquad + h\,(12\,y_1' + \ldots\ ),$$

$$3\,y_1 = 47\,y_5 + 192\,y_4 - \ldots \qquad - h\,(12\,y_5' + \ldots\ ),$$

$$3\,y_6 = 47\,y_2 + 192\,y_3 - \ldots \qquad + h\,(12\,y_2' + \ldots\ ),$$

$$3\,y_2 = 47\,y_6 + 192\,y_5 - \ldots \qquad - h\,(12\,y_6' + \ldots\ ).$$

Multiplying the preceding equations respectively by 184,407; 227,923; -103,472; -833,968; -884,509; -385,881 and then adding gives equation (28).

Equation (29). From section I, equation (7), we have:

$$6\,y_5 = 131\,y_0 + 1150\,y_1 + 600\,y_2 - 1400\,y_3 - 475\,y_4$$

$$+ h\,(30\,y_0' + 600\,y_1' + 1800\,y_2' + 1200\,y_3' + 150\,y_4'),$$

and similar equations for $6\,y_0$, $6\,y_6$, $6\,y_1$, $6\,y_7$, $6\,y_2$, $6\,y_8$, $6\,y_3$. As in the derivation of (27), (28), use multipliers 11,511; 36,326; 33,364; -47,718; -135,347; -160,883; -101,217; -24,178.

It is noted that formulas (27), (28), (29) are linear combinations of sets of formulas obtained from section II - A. Of course, an infinite number of correctors can be so generated. The ones just presented have their y-coefficients in approximate geometric progression. This makes the error term attractive, and for h = 0 the characteristic roots are within the unit circle. At large step sizes, Adams' formulas are sometimes preferred because they are less likely to become unstable.

Adams' formulas may be derived also by taking linear combinations of formulas obtained from section II - A. Also, they are found in the literature [refs. 1, pp. 10, 536; 5, pp. 194, 204]. They are given by

$$y_{r+1} = y_r + h \sum_{i=0}^{p} B_i \nabla^i y'_{r+1} + h^{p+2} B_{p+1} y^{(p+2)}(s).$$

The first few coefficients are $B_0 = 1$, $B_1 = -\frac{1}{2}$, $B_2 = -\frac{1}{12}$, $B_3 = -\frac{1}{24}$, $B_4 = -\frac{19}{720}$, $B_5 = -\frac{3}{160}$, $B_6 = -\frac{863}{60,480}$, $B_7 = -\frac{275}{24,192}$, $B_8 = -\frac{33,953}{3,628,800}$, $B_9 = -\frac{73,647}{9,331,200}$.

Equivalent formulas (without remainders listed) are:

$$y_1 = y_0 + \frac{h}{2}(y_0' + y_1'),$$

$$y_2 = y_1 + \frac{h}{12}(-y_0' + 8\,y_1' + 5\,y_2'),$$

$$y_3 = y_2 + \frac{h}{24}(y_0' - 5\,y_1' + 19\,y_2' + 9\,y_3'),$$

$$y_4 = y_3 + \frac{h}{720}\left(- 19\, y_0' + 106\, y_1' - 264\, y_2' + 646\, y_3' + 251\, y_4'\right) \quad ,$$

$$y_5 = y_4 + \frac{h}{1440}\left(27\, y_0' - 173\, y_1' + 482\, y_2' - 798\, y_3' + 1427\, y_4' - 1475\, y_5'\right) \quad ,$$

$$y_6 = y_5 + \frac{h}{60,480}\left(- 863\, y_0' + 6312\, y_1' - 20,211\, y_2' + 37,504\, y_3' - 46,461\, y_4' + 65,112\, y_5' + 19,087\, y_6'\right) .$$

A similar formula of order eight is listed as P5 in section III-A.

APPENDIX II. TABULATED RESULTS.

In these tables we include the estimated loss of time mentioned in section III - E. Notations identifying processes and differential equations follow that of the text. For P2 through P6, the final program was used (see section III). The constant r is the one defined in section III.

TABLE A-1. RESULTS OF E1.

Process	h	Error	Time (min)	Starters Produced At	Mins. Time Loss
P1	.03	0.5636×10^{-3}	1.6075		
	.06	0.3443×10^{-1}	0.8069		
	.10	0.7100×10^{0}	0.4881		
	.12	0.2080×10^{1}	0.4073		
	.18	0.2236×10^{2}	0.2741		
	.24	0.1186×10^{3}	0.2075		
P2 (r = .08)	.03	0.5482×10^{-1}	0.8305	h	0
	.06	0.2057×10^{1}	0.4789	"	0
	.10	0.2660×10^{2}	0.2936	"	0
	.12	0.6618×10^{2}	0.2466	"	0
	.18	0.4976×10^{3}	0.1692	"	0
	.24	0.2048×10^{4}	0.1305	"	0
P3 (r = .04)	.03	0.1948×10^{-3}	1.0475	h/2	0.0160
	.06	0.3231×10^{-1}	0.5953	"	"
	.09	0.5128×10^{0}	0.4070	"	"

TABLE A1. RESULTS OF E1. (Concluded)

Process	h	Error	Time (min)	Starters Produced At	Mins. Time Loss
P3 (cont.)	.12	0.3565×10^{1}	0.3128	h/2	0.0160
	.18	0.5245×10^{2}	0.2186	"	"
	.24	0.3385×10^{3}	0.1720	"	"
P4 (r = .04)	.03	0.1219×10^{-4}	1.2289	h/2	0.0160
	.06	0.3269×10^{-2}	0.6850	"	"
	.10	0.1178×10^{0}	0.4234	"	"
	.12	0.4227×10^{0}	0.3570	"	"
	.18	0.7196×10^{1}	0.2477	"	"
	.24	0.5311×10^{2}	0.1931	"	"
P5 (r = .04)	.03	0.7016×10^{-4}	1.1992	h/5	0.0853
	.06	0.1566×10^{-2}	0.6503	"	"
	.09	0.3710×10^{-1}	0.4672	"	"
	.12	0.3900×10^{0}	0.4028	"	"
	.18	0.8225×10^{1}	0.3019	"	"
	.24	0.6763×10^{2}	0.2513	"	"
P6 (r = .04)	.03	0.1540×10^{-4}	1.5636	h/5	0.0853
	.06	0.3689×10^{-4}	0.7744	"	"
	.10	0.5471×10^{-3}	0.5377	"	"
	.12	0.2833×10^{-2}	0.4633	"	"
	.18	0.1059×10^{0}	0.3414	"	"
	.24	0.1345×10^{1}	0.2803	"	"

TABLE A2. RESULTS OF E2

Process	h	Error	Time (min)	Starters Produced At	Time Loss (min)
P1	.02	0.1845×10^{-9}	5.7336		
	.04	0.1172×10^{-7}	2.8706		
	.08	0.7258×10^{-6}	1.4403		
	.12	0.7780×10^{-5}	0.9638		
	.18	0.7795×10^{-4}	0.6467		
	.24	0.3744×10^{-3}	0.4881		
P2 (r = .08)	.03	0.1008×10^{-5}	2.2150	h	0
	.06	0.1000×10^{-4}	1.1203	"	0
	.10	0.2685×10^{-4}	0.6836	"	0
	.12	0.2538×10^{-4}	0.5733	"	0
	.18	0.9650×10^{-4}	0.3911	"	0
	.24	0.5559×10^{-3}	0.3000	"	0
P3 (r = .04)	.03	0.6210×10^{-6}	2.2022	h/2	0.0381
	.06	0.1890×10^{-4}	1.1361	"	"
	.09	0.9929×10^{-4}	0.7808	"	"
	.12	0.2725×10^{-3}	0.6033	"	"
	.18	0.8876×10^{-3}	0.4258	"	"
	.24	0.1729×10^{-2}	0.3370	"	"

TABLE A2. RESULTS OF E2 (Concluded)

Process	h	Error	Time (min)	Starters Produced At	Time Loss (min)
P4 (r = .04)	.03	0.3518×10^{-7}	2.5664	h/2	0.0381
	.06	0.6126×10^{-6}	1.3164	"	"
	.10	0.7187×10^{-6}	0.8175	"	"
	.12	0.1971×10^{-5}	0.6917	"	"
	.18	0.3483×10^{-4}	0.4836	"	"
	.24	0.1166×10^{-3}	0.3794	"	"
P5 (r = .04)	.03	0.1472×10^{-6}	2.8772	h/5	0.2029
	.06	0.5140×10^{-5}	1.3867	"	"
	.09	0.2725×10^{-4}	1.0042	"	"
	.12	0.7352×10^{-4}	0.8127	"	"
	.18	0.2444×10^{-3}	0.5683	"	"
	.24	0.4357×10^{-3}	0.4872	"	"
P6 (r = .04)	.04	0.9493×10^{-8}	2.5948	h/5	0.2029
	.06	0.3740×10^{-7}	1.6367	"	"
	.10	0.6723×10^{-6}	1.0761	"	"
	.12	0.2375×10^{-5}	0.9344	"	"
	.18	0.1899×10^{-4}	0.6472	"	"
	.24	0.5845×10^{-4}	0.5834	"	"

TABLE A3. RESULTS OF E3

Process	h	Error	Time (min)	Starters Computed At	Time Loss (min)
P1	.03	0.1382×10^{-18}	1.6106		
	.06	0.8964×10^{-17}	0.8083		
	.09	0.1051×10^{-15}	0.5414		
	.12	0.6077×10^{-15}	0.4078		
	.18	0.7334×10^{-14}	0.2744		
	.24	0.4366×10^{-13}	0.2075		
P2 (r = .08)	.03	0.1662×10^{-16}	0.8322	h	0
	.06	0.4399×10^{-15}	0.4800	"	0
	.09	0.3222×10^{-14}	0.3247	"	0
	.12	0.1303×10^{-13}	0.2472	"	0
	.18	0.2818×10^{-13}	0.1698	"	0
	.24	0.2435×10^{-9}	0.1306	"	0
P3 (r = .04)	.03	0.5743×10^{-19}	1.0488	h/2	0.0160
	.06	0.9824×10^{-17}	0.5403	"	"
	.09	0.1784×10^{-15}	0.3709	"	"
	.12	0.1420×10^{-14}	0.2861	"	"
	.18	0.2750×10^{-13}	0.2014	"	"
	.24	0.2350×10^{-12}	0.1589	"	"

TABLE A3. RESULTS OF E3 (Concluded)

Process	h	Error	Time (min)	Starters Computed At	Time Loss (min)
P4 (r = .04)	.03	0.3890×10^{-20}	1.2317	h/2	0.0160
	.06	0.7173×10^{-18}	0.6309	"	"
	.09	0.1194×10^{-16}	0.4303	"	"
	.12	0.1430×10^{-15}	0.3305	"	"
	.18	0.1296×10^{-10}	0.2302	"	"
	.24	0.4631×10^{-8}	0.1803	"	"
P5 (r = .04)	.03	0.1606×10^{-19}	1.2011	h/5	0.0856
	.06	0.5402×10^{-18}	0.6514	"	"
	.09	0.1478×10^{-16}	0.4683	"	"
	.12	0.1791×10^{-15}	0.4037	"	"
	.18	0.5315×10^{-14}	0.3028	"	"
	.24	0.6150×10^{-13}	0.2520	"	"
P6 (r = .04)	.03	0.2320×10^{-20}	1.4541	h/5	0.0856
	.06	0.1153×10^{-19}	0.7761	"	"
	.09	0.1578×10^{-13}	0.5867	"	"
	.12	0.9120×10^{-10}	0.4644	"	"
	.18	0.5630×10^{-6}	0.3419	"	"
	.24	0.3578×10^{-4}	0.2808	"	"

TABLE A4. RESULTS OF E4

Process	h	Error	Time (min)	Starters Computed At	Time Loss (min)
P1	.03	0.5170×10^{-15}	2.1484		
	.06	0.2778×10^{-13}	1.0769		
	.09	0.3268×10^{-12}	0.7203		
	.12	0.1899×10^{-11}	0.5419		
	.18	0.2309×10^{-10}	0.3639		
	.24	0.1381×10^{-9}	0.2747		
P2 (r = .08)	.03	0.3892×10^{-13}	0.9873	h	0
	.06	0.2265×10^{-11}	0.5025	"	0
	.09	0.1135×10^{-10}	0.4653	"	0
	.12	0.2672×10^{-10}	0.3067	"	0
	.18	0.1134×10^{-9}	0.2100	"	0
	.24	0.2320×10^{41}	0.1394	"	0
P3 (r = .04)	.03	0.1015×10^{-13}	1.1733	h/2	0.0214
	.06	0.1302×10^{-11}	0.6072	"	"
	.09	0.6201×10^{-11}	0.4186	"	"
	.12	0.7905×10^{-10}	0.3244	"	"
	.18	0.2019×10^{-8}	0.2303	"	"
	.24	0.3918×10^{-8}	0.1831	"	"

TABLE A4. RESULTS OF E4 (Concluded)

Process	h	Error	Time (min)	Starters Computed At	Time Loss (min)
	.03	0.9590×10^{-15}	1.3561	h/2	0.0214
	.06	0.8544×10^{-13}	0.7914	"	"
P4	.09	0.1494×10^{-12}	0.5400	"	"
(r = .04)	.12	0.9297×10^{-11}	0.3686	"	"
	.18	0.3043×10^{-8}	0.2889	"	"
	.24	0.2025×10^{-6}	0.2042	"	"
	.03	0.8970×10^{-15}	1.5397	h/5	0.1143
	.06	0.6492×10^{-12}	0.8364	"	"
	.09	0.9604×10^{-11}	0.6020	"	"
P5	.12	0.3883×10^{-10}	0.4397	"	"
(r = .04)	.18	0.1866×10^{-8}	0.3386	"	"
	.24	0.4390×10^{-8}	0.3092	"	"
	.03	0.1460×10^{-15}	1.7933	h/5	0.1143
	.06	0.2042×10^{-14}	0.9614	"	"
P6	.09	0.2125×10^{-11}	0.6841	"	"
(r = .04)	.12	0.7706×10^{-8}	0.5458	"	"
	.18	0.1833×10^{-4}	0.3783	"	"
	.24	0.5610×10^{-3}	0.3167	"	"

REFERENCES

1. Collatz, L.: The Numerical Treatment of Differential Equations, 3rd ed., Springer, Berlin, 1960.

2. Dahlquist, Germund.: Stability and Error Bounds in the Numerical Integration of Ordinary Differential Equations. Almquist and Wiksells Boktrycheri A B, Uppsala, 1958.

3. Fehlberg, Erwin.: Numerically Stable Interpolation Formulas with Favorable Error Propagation for First and Second Order Differential Equations. NASA TN D-599, 1961.

4. Hamming, R. W.: Numerical Methods for Scientists and Engineers. McGraw-Hill, New York, 1962.

5. Henrici, Peter.: Discrete Variable Methods in Ordinary Differential Equations. Wiley, New York, 1962.

6. Kunz, K. S.: Numerical Analysis. McGraw-Hill, New York, 1957.

7. Milne, W. E.: Numerical Solution of Differential Equations. Wiley, New York, 1953.

8. Nordsieck, Arnold.: On Numerical Integration of Ordinary Differential Equations. Mathematics of Computation, vol. 16, no. 77, January 1962.

9. Shanks, E. B.: Formulas for Obtaining Solutions of Differential Equations by the Evaluation of Functions. Vanderbilt University, Nashville, Tennessee, 1963.

NASA-Langley, 1965 M509

2/22/65

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Technical information generated in connection with a NASA contract or grant and released under NASA auspices.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

TECHNICAL REPRINTS: Information derived from NASA activities and initially published in the form of journal articles.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities but not necessarily reporting the results of individual NASA-programmed scientific efforts. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546