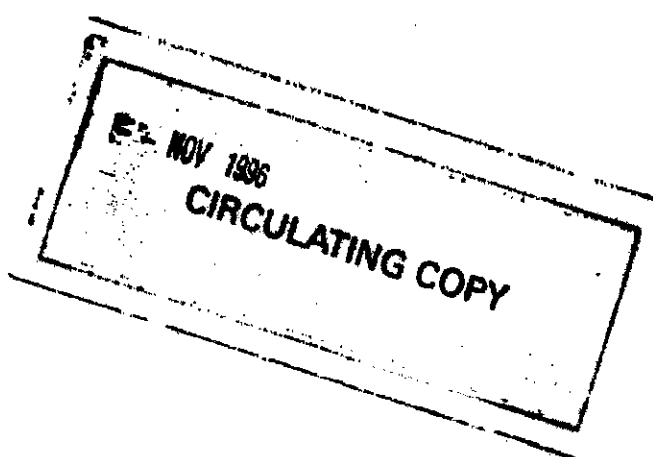
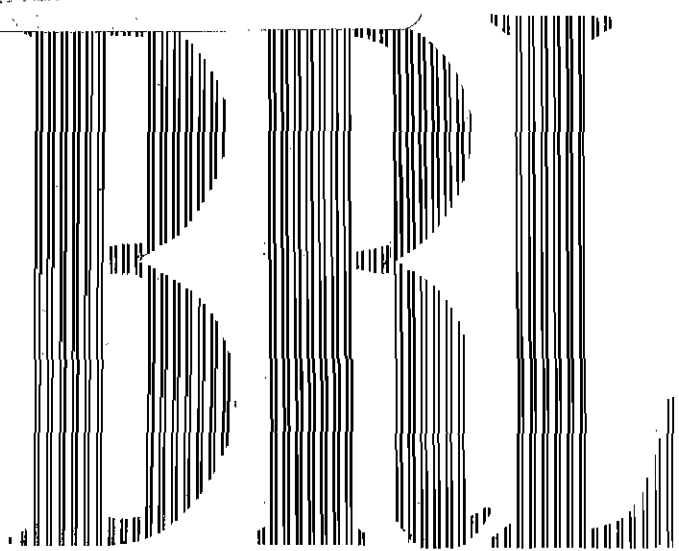


UNCLASSIFIED

AD NUMBER
AD030120
NEW LIMITATION CHANGE
TO Approved for public release, distribution unlimited
FROM Distribution authorized to U.S. Gov't. agencies and their contractors; Administrative/Operational Use; JAN 1954. Other requests shall be referred to Army Ballistic Research Laboratories, Aberdeen Proving Ground, MD 21005-5066.
AUTHORITY
ST-A per USAARDCOM ltr dtd 19 Sep 1983

THIS PAGE IS UNCLASSIFIED



REPORT No. 893

Automatic Error Control, The Initial Value Problem In Ordinary Differential Equations

S. GORN

R. MOORE

TECHNICAL LIBRARY
DRYBR-LB (Bldg. 305)
ABERDEEN PROVING GROUND, MD. 21005

DEPARTMENT OF THE ARMY PROJECT No. 503-06-002
ORDNANCE RESEARCH AND DEVELOPMENT PROJECT No. TB3-0007

BALLISTIC RESEARCH LABORATORIES



ABERDEEN PROVING GROUND, MARYLAND

U-10947

BALLISTIC RESEARCH LABORATORIES

REPORT NO. 893

JANUARY 1954

AUTOMATIC ERROR CONTROL,
THE INITIAL VALUE PROBLEM IN
ORDINARY DIFFERENTIAL EQUATIONS

S. Gorn

R. Moore

TECHNICAL LIBRARY
DRXBR-LB (Bldg. 305)
ABERDEEN PROVING GROUND, MD. 21005

Department of the Army Project No. 503-06-002
Ordnance Research and Development Project No. TB3-0007

ABERDEEN PROVING GROUND, MARYLAND

TABLE OF CONTENTS

	Page No.
Preface	5
1. Introduction	7
2. The Initial Value Problem on High-Speed, Digital Computing Machines	10
2.1 Formulation of the Computation Problem	10
2.2 Single-step Integrating Procedures	11
2.3 The Error in the Large	14
2.4 Variation of Step Size	20
3. Automatic Error Control	22
3.1 A Special Bounding Procedure	22
3.2 A General Error Estimating Procedure	28
3.3 Example & Conclusion	39

BALLISTIC RESEARCH LABORATORIES

REPORT NO. 893

JANUARY 1954

SGorn/RMoore/orw
Aberdeen Proving Ground, Md.
January 1954

AUTOMATIC ERROR CONTROL,
THE INITIAL VALUE PROBLEM IN
ORDINARY DIFFERENTIAL EQUATIONS

ABSTRACT

This report describes some general routines which will automatically analyze and control the accuracy of numerical procedures on high-speed digital computers. The routines described are applicable to any system of ordinary differential equations for which a solution is being sought to satisfy given initial conditions, provided that the numerical method being used is of the step-by-step type. Seven such numerical methods are listed, along with the results of applying the automatic control to the study of a spinning top.

INTENTIONALLY LEFT BLANK.

Preface

With high speed digital computers, the speed of running of a problem is naturally much less critical than it was with slow machines or with desk computers. In fact some of the simplest computing schemes, those which had to be discarded in the past as being unsuitable for hand computation on the basis of time consumed, now call for reconsideration. On the other hand, storage, which has been abundantly available for hand computers in the form of blank paper, published tables, and published books describing computational procedures, becomes a critical and controlling consideration for high speed machines, at least at present.

However, it is not true that running time of a problem is the only use of time that needs consideration. One cannot ignore the time needed in coding, code-checking, error analysis, and analysis generally. Now, since the high speed machines can make multiple choices as long as there is a fixed criterion determining such a choice at each encounter of an alternative, any sequence of arithmetic operations connected in such forms as "either or ... depending on whether ... or not" can be coded. The machines can therefore be made to exercise routine judgements for us. A large part of coding, code-checking, error analysis, and analysis generally can be so formalized, and is therefore capable of being done by the machines. And, indeed, serious attempts are currently being made to get machines to do some automatic coding. Furthermore a simultaneous semi-automatic coding and a saving of code-checking time can be achieved by accumulating a "library" of such basic subroutines as polynomial evaluation, root finding, integrating procedures, etc., any one of which can be called in automatically. As for analysis, one could right now code routines performing operations on power series; but the bulk of "mechanical" analysis will not be coded until machines are built with a large enough memory to permit us freely to imitate with an appropriate use of machine storage that extended use of variables in which they are called "indeterminates" (in German - "Unbestimmte").

We come now to the consideration of time spent in error analysis. No extensive computation can be regarded as satisfactory without a subsidiary computation which evaluates its accuracy. In planning a computational schedule an important use of such error-analyses is to determine how the computation should proceed (See, for example, F. J. Murray, [1]) if the accuracy is to be controlled. We are interested, for high speed machines, in having such analyses and controls coded to run along with the basic procedure so that not only do we compute our error estimates as we go along, but we also save the time that would be lost in stopping the basic computation to perform such analyses and to decide which way to continue. Most error estimates in the past have been deliberately crude in order to permit hand computation by some simple formula. This was necessary because a human computer was justifiably reluctant to keep track of the, perhaps, thousands of errors accumulating due to approximating formulae and methods and the accumulation of round-offs. With the use of high speed machines such reluctance should disappear. It is conceivable that each approximating formula can be coded to include

a parallel computation of an error estimate or bound for the error due to initial errors, and to the so-called "truncation" error of the formula. It is further conceivable that a general error estimating procedure could be coded which would do three things: 1 - it would assign a separate memory position to the error of each computed quantity, beginning with those designated as initial errors, 2 - the procedure would, for each arithmetic operation, compute its effect on the errors of the computed quantities, including an estimate of rounding-off at that operation, 3 - the procedure could distinguish among certain types of discriminating orders, and act accordingly. (Each discrimination order, according to the outcome of the comparison involved, orders either a repetition of a subroutine or the beginning of another). For one type of discrimination, error estimates are to be accumulated; for other types, a prior analysis must be made available for proper use of the discrimination. This analysis provides an estimate, with accumulation (see S. Gorn, [2]).

Such a procedure could be coded to do this error estimating, in each of its three functions mentioned above, either by finding absolute bounds or by finding statistical error estimates at each operation.

A general error estimating procedure of such a kind pays for its generality in the same way as do all the so-called "monitoring" subroutines, which examine orders before carrying them out, namely, by being much slower. In practice it would seem preferable to give up this complete generality in favor of separate error estimating procedures for large classes of problems. An inverse interpolation method whereby a machine can be made to find to a high accuracy the region of indeterminacy (due to round-off) of the root of an equation in one numerical variable is described in the last chapter of S. Gorn, [2]. The present report provides such an error estimating procedure for solving the initial value problem with systems of ordinary differential equations.

1. Introduction

We will now discuss a simple scheme for estimating and controlling errors in the solution of systems of ordinary differential equations. For the moment let us restrict ourselves to a single equation:

$$(1) \quad x' = f(t, x).$$

The error control schemes we have in mind assume that we have a single-step numerical integrating procedure so modified that it yields a reasonably good upper and lower bound for x in terms of the value at the preceding step. We will give several examples of such procedures below. Later we will see the effect of relaxing the bounding conditions to mere error estimating conditions. The most general subroutine discussed below applies at each step the well-known extrapolation to zero grid method (see L. F. Richardson, and J. A. Gaunt, [3]) to obtain such bounds. It in turn is applicable to all such single-step integrating procedures as those of Euler and Runge-Kutta.

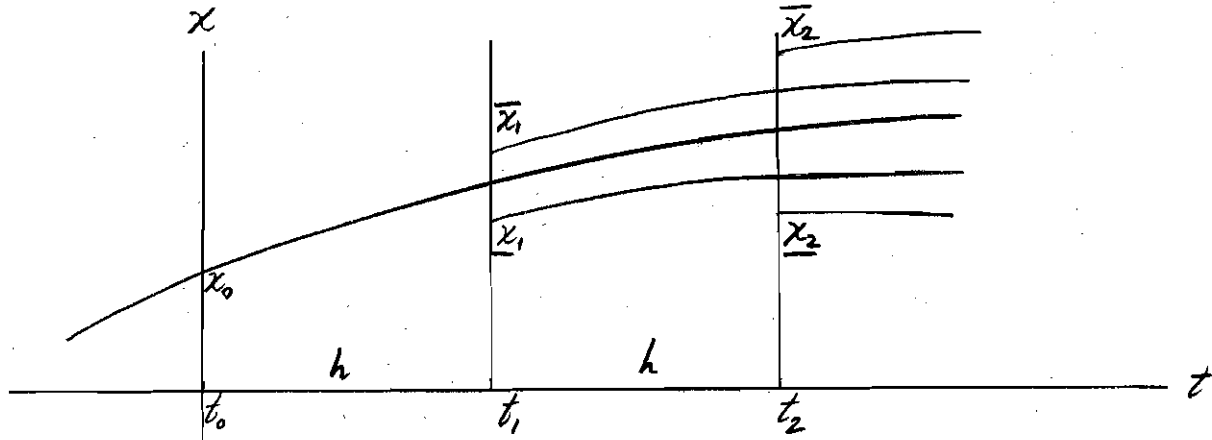


Figure 1

Assuming, then, that we have a number of such procedures for single step integration, standard modifying procedures can be made to apply to all of them, which modifications can estimate the errors sharply at each step and can begin the operation afresh at any desired point with modified mesh size.

In a region containing no singular points (see figure 1) the solution curves do not cross. Hence beginning with the initial condition x_0 at t_0 , one can compute the bounds $\underline{x}_1$ and $\bar{x}_1$ at t_1 ; then beginning with each of these as initial conditions, there will be two pair of bounds bracketing the corresponding solutions, so that the maximum, $\bar{x}_2$, and minimum, $\underline{x}_2$, will be bounds for the solution sought. This process can be continued, and bounds the accumulated error. If for preassigned measure of accuracy, ϵ , the region of indeterminacy of x , namely $\bar{x}_i - \underline{x}_i$,

ever becomes bigger than 2ϵ , the procedure can be set to divide the mesh size at the first step by two and begin the integration over again. This will be continued until either the integration is completed with y known to within ϵ or else the mesh size has gone below a given δ . (In this latter case the integration can be permitted to proceed without further interruption). The total time taken, with all these mesh-halvings, will never be more than twice the time required by the last mesh size (namely, that mesh deemed necessary according to the error estimating method used); for if T_j is the time needed for the last run through, the preceding run through, having less than half the number of mesh points, would need less than $\frac{1}{2} T_j$, and so on back.

Consequently the total time

$$< T_j + \frac{1}{2}T_j + \frac{1}{4}T_j + \dots + \frac{1}{2^j}T_j = (2 - \frac{1}{2^j}) T_j < 2T_j,$$

as stated.

As a result of this time consideration all that is needed in choosing an initial mesh size is to take it no larger than that required to be printed out, and preferably no larger than in the bracketing conditions discussed below, if these are convenient.

The error control just described can be coded as an independent procedure applicable to any integrating scheme of the type assumed. Its logical flow chart is shown in Figure 2, beginning at input i and ending at exit e.

If an initial error is to be assumed, we can even choose $\bar{x}_0 = x_0 + \eta$, $\underline{x}_0 = x_0 - \eta$ at $t = t_0$, where η is a bound or an absolute estimate of the initial error.

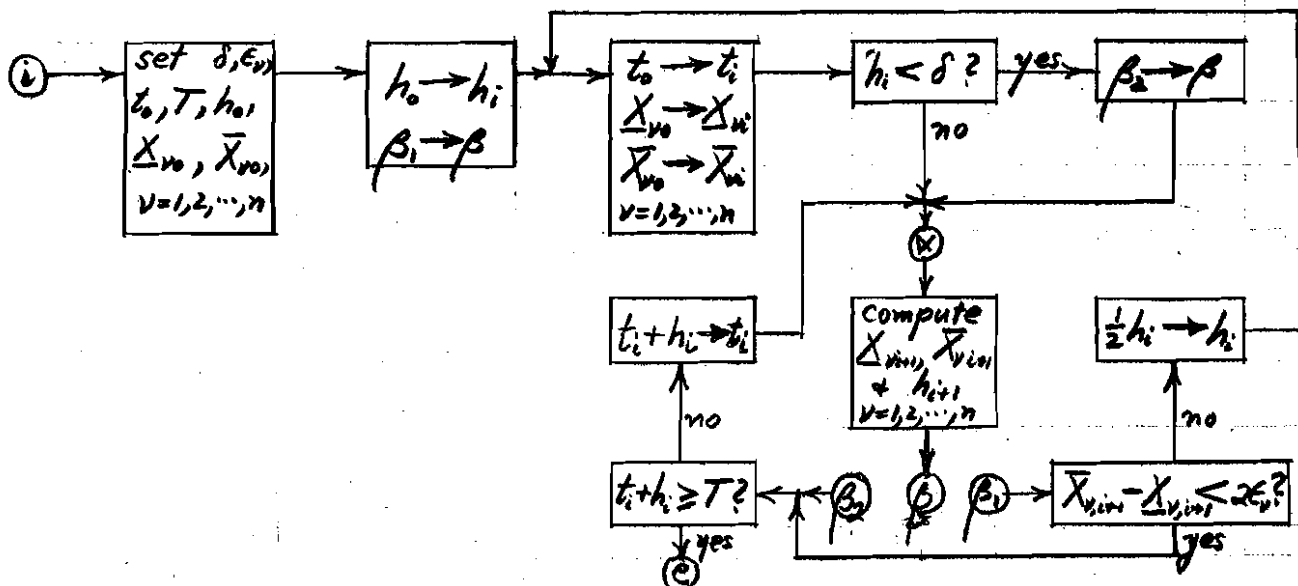


Figure 2

Error Control

At the place marked e in figure 2 one can print or store, as one wishes, either $\underline{X}_{vi}$, $\bar{X}_{vi}$, and $X_{vi} = 1/2 (\underline{X}_{vi} + \bar{X}_{vi})$ or X_{vi} and

$\Delta_{vi} = \frac{1}{2} |\bar{X}_{vi} - \underline{X}_{vi}|$ at the end of the sequence, or the complete sequences,

or any chosen subsequences. These printing options, as well as the two boxes directly above in figure 2 are already included in the general error estimating procedure of figure 11, which is one form that the section from α to β can take. Since other forms are possible, the orders from α to β should be last in the sequence from i to e.

2. The Initial Value Problem on High-Speed Digital Computing Machines

2.1 - Formulation of the Computational Problem

By a numerical solution of the initial value problem we will mean a set of numbers, $X_{v,i}$, considered to be approximations to $x_v(t_i)$, where the functions $x_v(t)$ are uniquely defined in some region by the n^{th} order system of ordinary differential equations:

$$(1) \quad \frac{dx_v(t)}{dt} = f_v(t, x_1, \dots, x_n), \quad v = 1, 2, \dots, n,$$

and the initial conditions:

$$(2) \quad x_v(t_0) = x_{v,0}, \quad v = 1, 2, \dots, n.$$

A numerical integrating procedure will begin at $t = t_0$ with $X_{v,0} = x_{v,0}$. To proceed from $t = t_i$ to $t = t_{i+1} > t_i$, increments,

$\Delta X_{v,i}$, will be computed to give

$$(3) \quad X_{v,i+1} = X_{v,i} + \Delta X_{v,i}$$

as approximations to $x_v(t)$ at $t = t_{i+1}$.

Suppose we have a numerical integrating procedure whose "local truncation error", $\tau_{v,i}$, is of order $k + 1$. In other words, if every numerical operation of the procedure were carried out with infinite precision, it would yield approximations, $X_{v,i}$ such that if $x_{v,i}(t)$ is the solution of equation (1) satisfying the initial condition $x_{v,i}(t_i) = X_{v,i}$, then at $t = t_{i+1}$ we should have:

$$(4) \quad \tau_{v,i} = x_{v,i}(t_{i+1}) - X_{v,i+1} = O(h^{k+1}),$$

where $h = t_{i+1} - t_i$. Such a numerical integrating procedure is said to be of k 'th order.

* $A = O(B)$ will have the standard meaning, namely, A/B is uniformly bounded for the whole range of variables in A and B . In particular, for our application $A = O(h^\alpha)$ means A/h^α remains bounded as $h \rightarrow 0$.

We will show in section 2.3 that equations (1) and (2) together imply

$$(5) \quad X_{v,i} = x_v(t_i) + O(h^k)$$

in a region where the derivatives, $\frac{\partial f_v}{\partial x_\mu}$, $\mu, v = 1, 2, \dots, n$

are uniformly bounded.

When a system (1), (2) is given, one is faced with the question of how a numerical solution in the above sense is to be chosen in an efficient way to attain a desired accuracy.

We begin, in the next section, by describing a number of such numerical procedures. In the section following we discuss the growth of errors over a number of steps. In chapter 3 we will describe methods of automatically estimating and controlling this growth of errors.

2.2 - Single-Step Integrating Procedures

Particularly convenient for machine application are the "single-step" methods. In these the computation of $\Delta X_{v,i}$ involves no values of the $X_{v,i}$ prior to the i 'th step.*

The convenience of such methods is due to three advantages they possess.

In the first place they make fewer demands on the memory of the machines than do multiple-step methods, which use values of $X_{v,j}$ for $j < i$ when $\Delta X_{v,i}$ is being computed. Thus larger systems of equations can be accommodated.

In the second place, and even more important, with a single-step method the interval of integration, $h_i = t_{i+1} - t_i$, may be changed from step to step without introducing any change in the formulae for computing $\Delta X_{v,i}$. With multiple-step methods, generally complicated interpolation procedures are required for a change in step-size. An appropriate variation in the step-size aids considerably in reducing the growth of error in the large, as will be remarked in Section 4 below.

* This definition may be found in Bennett, A. A., Milne, W. E., and Bateman, H. - Numerical Integration of Differential Equations - Bull. Nat. Res. Council, U. S. 92 (1933) - pp. 51-87.

The third advantage is that single-step methods can begin directly from the initial conditions, whereas multiple-step procedures require special formulae to compute the first few values of $X_{v,i}$.

We now list, both for reference and for illustrative purposes, some of the best known single-step methods. They are all applicable to systems of first order equations. For brevity we will use the vector notation:

$$A_i = \{A_{v,i} \ v = 1, 2, \dots, n\} = \{A_{1i}, A_{2i}, \dots, A_{ni}\}$$

for quantities with the subscript v .

M1. - "Euler's method" (see Levy and Baggott: [4])

$$k = 1 \quad \Delta X_i = h_i f(t_i, X_i).$$

M2. - (See Collatz [5])

$$k = 2 \quad \Delta_i = \frac{h_i}{2} f(t_i, X_i), \text{ and}$$

$$\Delta X_i = h_i f(t_i + \frac{h_i}{2}, X_i + \Delta_i).$$

M3. - "Modified method of Euler" (Levy and Baggott, loc. cit.)

$$k = 2 \quad \Delta X_i = \frac{1}{2} (\Delta_i + \Delta'_i), \text{ where}$$

$$\Delta_i = h_i f(t_i, X_i), \text{ and}$$

$$\Delta'_i = h_i f(t_i + h_i, X_i + \Delta_i).$$

M4. - "Kutta's third order approximation" (Levy and Baggott, loc. cit.)

$$k = 3 \quad \Delta X_i = \frac{1}{4} (\Delta_i + 3 \Delta''_i), \text{ where}$$

$$\Delta_i = h_i f(t_i, X_i),$$

$$\Delta'_i = h_i f(t_i + \frac{1}{3} h_i, X_i + \frac{1}{3} \Delta_i),$$

$$\Delta''_i = h_i f(t_i + \frac{2}{3} h_i, X_i + \frac{2}{3} \Delta'_i).$$

M5. - "Runge-Kutta" (Collatz, loc. cit.)

$$k = 4 \quad \Delta X_i = \frac{1}{6} (\Delta_i + 2 \Delta'_i + 2 \Delta''_i + \Delta'''_i), \text{ where}$$

$$\Delta_i = h_i f(t_i, X_i),$$

$$\Delta'_i = h_i f(t_i + \frac{1}{2} h_i, X_i + \frac{1}{2} \Delta_i),$$

$$\Delta''_i = h_i f(t_i + \frac{1}{2} h_i, X_i + \frac{1}{2} \Delta'_i),$$

$$\Delta'''_i = h_i f(t_i + h_i, X_i + \Delta''_i).$$

M6. - "Runge-Kutta-Gill" - This is a slight modification of M5 which results in not requiring so much machine storage. (S. Gill - [6]).

$$k = 4 \quad \Delta X_i = \frac{1}{6} (\Delta_i + [2 - \sqrt{2}] \Delta'_i + [2 + \sqrt{2}] \Delta''_i + \Delta'''_i), \text{ where}$$

$$\Delta_i = h_i f(t_i, X_i)$$

$$\Delta'_i = h_i f(t_i + \frac{1}{2} h_i, X_i + \frac{1}{2} \Delta_i)$$

$$\Delta''_i = h_i f(t_i + \frac{1}{2} h_i, X_i + \frac{1}{2} [-1 + \sqrt{2}] \Delta_i + \frac{1}{2} [2 - \sqrt{2}] \Delta'_i)$$

$$\Delta'''_i = h_i f(t_i + h_i, X_i - \frac{\sqrt{2}}{2} \Delta'_i + \frac{1}{2} [2 + \sqrt{2}] \Delta''_i).$$

M7. - "Kutta-Nyström" (E. J. Nystrom, [7]).

$$k = 5 \quad \Delta X_i = \frac{1}{192} (23 \Delta_i + 125 \Delta''_i - 81 \Delta_i^{iv} + 125 \Delta_i^v), \text{ where}$$

$$\Delta_i = h_i f(t_i, X_i),$$

$$\Delta'_i = h_i f(t_i + \frac{1}{3} h_i, X_i + \frac{1}{3} \Delta_i),$$

$$\Delta''_i = h_i f(t_i + \frac{2}{5} h_i, X_i + \frac{6 \Delta'_i + 4 \Delta_i}{25})$$

$$\begin{aligned}\Delta_i''' &= h_i f(t_i + h_i, X_i + \frac{15 \Delta_i'' - 12 \Delta_i' + \Delta_i}{4}), \\ \Delta_i^{iv} &= h_i f(t_i + \frac{2}{3} h_i, X_i + \frac{8 \Delta_i''' - 50 \Delta_i'' + 90 \Delta_i' + 6 \Delta_i}{81}), \\ \Delta_i^v &= h_i f(t_i + \frac{4}{5} h_i, X_i + \frac{8 \Delta_i'' + 10 \Delta_i' + 36 \Delta_i + 6 \Delta_i}{75}).\end{aligned}$$

2.3 - The Error in the Large

We will now take up the proof of 2.1 (5) in the case where all computations are assumed to be carried out to infinite precision, and will then see what modifications are necessary due to the fact that they are, in reality, carried out to a finite number of places.

As in section 2.1 we let $x_{v,i}(t)$ be the solution of the system 2.1 (1) satisfying the initial condition:

$$(1) \quad x_{v,i}(t_i) = X_{v,i}.$$

Let us define the "error functions":

$$(2) \quad E_{v,i}(t) = x_v(t) - x_{v,i}(t).$$

Then these error functions satisfy the system of differential equations:

$$\begin{aligned}(3) \quad E'_{v,i}(t) &= \frac{dE_{v,i}(t)}{dt} = \frac{dx_v(t)}{dt} - \frac{dx_{v,i}(t)}{dt} \\ &= f_v(t, x_1, \dots, x_n) - f_v(t, x_{1,i}, \dots, x_{n,i}).\end{aligned}$$

By the mean value theorem, if f_v are of class C' in a region R , then

$$(4) \quad E'_{v,i}(t) = \frac{\partial f_v}{\partial x_1} \bigg|_{P_{v,i}} E_{1,i}(t) + \dots + \frac{\partial f_v}{\partial x_n} \bigg|_{P_{v,i}} E_{n,i}(t),$$

where the partial derivatives $\frac{\partial f_v}{\partial x_\mu}$ ($\mu = 1, 2, \dots, n$) are

evaluated at $P_{v,i} = (t; \xi_{1,v,i}, \dots, \xi_{n,v,i})$ for some

$$\xi_{j,v,i} = x_j(t) + \theta_{v,i} [x_{j,i}(t) - x_j(t)] ,$$

$$0 \leq \theta_{v,i} \leq 1.$$

Also $E_{v,i}(t)$ satisfy the initial conditions:

$$(5) \quad E_{v,i}(t_i) = x_v(t_i) - X_{v,i}.$$

Now since f_v are of class C' in R , the derivatives are uniformly bounded in some closed subregion S of R , which subregion is assumed to contain the points $P_{v,i}$. Suppose, then,

$$(6) \quad \left| \frac{\partial f_v}{\partial x_j} \right| \leq L \text{ in } S.$$

It follows from equation (4) that

$$(7) \quad |E'_{v,i}(t)| \leq L \left\{ |E_{1,i}(t)| + \dots + |E_{n,i}(t)| \right\}.$$

Let us define $E_i(t)$ by:

$$(8) \quad E_i(t) = \max_v |E_{v,i}(t)|.$$

It follows from equation (7) that

$$(9) \quad |E'_{v,i}(t)| \leq nLE_i(t) \text{ for } v = 1, 2, \dots, n.$$

Now clearly, from equation (3), since f_v are continuous in R , $E_{v,i}(t)$ are of class C' for $t_0 \leq t \leq t_{i+1}$; hence $E_i(t)$ are continuous and the right-hand derivatives* $D_+ E_i(t)$ are piecewise continuous in the same interval. Furthermore, from the defining equations (8) we have, for each t and some v varying with t in $t_0 \leq t \leq t_{i+1}$ that $|D_+ E_i(t)| =$

$|E'_{v,i}(t)|$. Hence:

*

$$D_+ E_i(t) = \lim_{\bar{t} \rightarrow t + 0} \frac{E_i(\bar{t}) - E_i(t)}{\bar{t} - t}$$

$$(10) \quad |D_+ E_i(t)| \leq \max_v |E'_{v,i}(t)|, \quad t_0 \leq t \leq t_{i+1}.$$

It therefore follows from equation (9) that

$$(11) \quad |D_+ E_i(t)| \leq nL E_i(t), \quad t_0 \leq t \leq t_{i+1}.$$

It is now possible to apply the following:

Lemma: (E. Kamke - [8], Hilfssatz 2, p. 93) If $u(x)$ is continuous on the closed interval $[a, b]$ and has a right hand derivative for $a \leq x \leq b$, and if there exist constants, $M > 0$, $N \geq 0$ such that

$|D_+ u(x)| \leq M |u(x)| + N$, then for any two numbers x, ξ in $[a, b]$ we have

$$|u(x)| \leq u(\xi) e^{M|x-\xi|} + \frac{N}{M} (e^{M|x-\xi|} - 1).$$

Taking $E_i(t)$ for $u(x)$, nL for M , and 0 for N , $x = t_{i+1}$, $\xi = t_i$, and $h = t_{i+1} - t_i$ (in the remainder of this section h will be constant), we have:

$$(12) \quad E_i(t_{i+1}) \leq E_i(t_i) e^{nLh}.$$

Now from equation 2.1(4) we have that $x_v(t_{i+1}) - X_{v,i+1} = x_v(t_{i+1}) - x_{v,i}(t_{i+1}) + O(h^{k+1})$, i.e., by (2) and (1),

$$(13) \quad E_{v,i+1}(t_{i+1}) = E_{v,i}(t_{i+1}) + O(h^{k+1}).$$

Consequently, by (8), (12), and (13):

$$(14) \quad E_{i+1}(t_{i+1}) \leq E_i(t_i) e^{nLh} + O(h^{k+1}).$$

Applying (14) repeatedly therefore yields:

$$(15) \quad E_{i+1}(t_{i+1}) \leq E_0(t_0) e^{(i+1)nLh} + [1 + e^{nLh} + e^{2nLh} + \dots + e^{inLh}] O(h^{k+1}).$$

But since $X_{v,0} = x_v(t_0)$ initially, $E_0(t_0) = 0$, so that (15) becomes:

$$(16) \quad E_{i+1}(t_{i+1}) \leq \frac{e^{(i+1)nLh} - 1}{e^{nLh} - 1} O(h^{k+1}).$$

From the fact that $i+1 = \frac{1}{h} (t_{i+1} - t_0)$ and $e^{nLh} - 1 > nLh$ for $nLh > 0$

we get:

$$(17) \quad E_{i+1}(t_{i+1}) \leq \frac{e^{nL(t_{i+1}-t_0)} - 1}{nLh} O(h^{k+1}),$$

whence

$$(18) \quad E_i(t_i) = \max_v |x_v(t_i) - X_{v,i}| = O(h^k),$$

as stated in equation 2.1 (5). This completes the proof, promised in section 2.1, that if a k 'th order integrating procedure is carried out with infinite precision it yields an approximation in the large of order k in a region where the f_v have uniformly bounded first partial derivatives.

In practice, of course, computations will be performed with something less than infinite precision. Real numbers will be replaced by rational approximations to a fixed number of places. The functions f_v may involve infinite series; these will have to be truncated or otherwise approximated if they involve integrals, they may be approximated by finite sums; if they involve roots of algebraic equations, these may have to be approximated. In general, to solve digitally requires reduction of all computations to the rational operations, and the arithmetic operations, when performed on two s -place numbers do not, in general, yield exact s -place numbers; the result must be truncated or rounded to s -places. Scaling may be necessary to keep all number sizes within machine range; this may result in the loss of significant figures.

To see how this finite precision affects our accuracy, let us represent (for the remainder of this section only) by $\bar{X}_{v,i}$ the actual approximation corresponding to the "infinite precision" approximation $X_{v,i}$ yielded by the k 'th order method. Similarly, let $\bar{x}_{v,i}(t)$ be the solution of equations 2.1 (1) satisfying the initial condition $\bar{x}_{v,i}(t_i) = \bar{X}_{v,i}$. Then in place of 2.1 (4) we will have

$$(19) \quad \bar{x}_{v,i}(t_{i+1}) - \bar{X}_{v,i+1} = \tau_{v,i} + \epsilon_{v,i}.$$

That is, the local error will consist of the local truncation error $\tau_{v,i} = O(h^{k+1})$ plus an error, $\epsilon_{v,i}$, from sources mentioned in the preceding paragraph. For example, for method M1 given in 2.2, $\epsilon_{v,i}$ will have the form

$$(20) \quad \epsilon_{v,i} = h \textcircled{\times} \bar{f}_{v,i} - h f_{v,i},$$

where $f_{v,i} = f_v(t_i, \bar{X}_{1,i}, \bar{X}_{2,i}, \dots, \bar{X}_{n,i})$ for $v = 1, 2, \dots, n$,

$\bar{f}_{v,i}$ is the computed approximation to $f_{v,i}$, $\bar{h}$ is the computed approximation to h , and $\textcircled{\times}$ is the rounded multiplication. (Similar expressions

for $\epsilon_{v,i}$ can be obtained for methods M2 to M7 in 2.2). In this case we may (in the manner of von Neumann and Goldstine, [9]) decompose $\epsilon_{v,i}$ in the following way:

$$\begin{aligned}
 \epsilon_{v,i} &= \bar{h} \otimes \bar{f}_{v,i} - \bar{h} \bar{f}_{v,i} \\
 (21) \quad &+ \bar{h} \bar{f}_{v,i} - \bar{h} f_{v,i} \\
 &+ \bar{h} f_{v,i} - h f_{v,i} = \rho_{v,i} + \bar{h}(\bar{f}_{v,i} - f_{v,i}) + f_{v,i}(\bar{h} - h),
 \end{aligned}$$

where $\rho_{v,i}$ is the round-off error in the machine product of $\bar{h}$ by $\bar{f}_{v,i}$.

For machine multiplication with radix β and rounding to s places,

$|\rho_{v,i}| \leq \frac{1}{2} \beta^{-s}$. In any case, if $\epsilon_{v,i}$ is small compared to the local truncation error, $\tau_{v,i}$, for a given h , then estimates for the error $x_{v,i}(t_{i+1}) - \bar{x}_{v,i+1}$ will serve as well for estimates of $\bar{x}_{v,i}(t_{i+1}) - \bar{x}_{v,i+1}$. However, we will analyze the effect of the presence of $\epsilon_{v,i}$ in (19) on the error in the large.

We let $\bar{E}_{v,i}(t) = x_v(t) - \bar{x}_{v,i}(t)$. $\bar{E}_{v,i}(t)$ will satisfy the initial condition

$$(22) \quad \bar{E}_{v,i}(t_i) = x_v(t_i) - \bar{x}_{v,i}.$$

Corresponding to (3), we have

$$(23) \quad \bar{E}_{v,i}'(t) = f_v(t, x_1, \dots, x_n) - f_v(t, \bar{x}_{1,i}, \dots, \bar{x}_{n,i}).$$

We may follow through (3) to (12) exactly as before, but with $\bar{E}_{v,i}(t)$

and $\bar{x}_{v,i}(t)$ instead of $E_{v,i}(t)$ and $x_{v,i}(t)$. We have therefore,

$$(24) \quad \bar{E}_i(t_{i+1}) \leq \bar{E}_i(t_i) e^{nLh},$$

where $h = t_{i+1} - t_i$ and $\bar{E}_i(t) = \max_v |\bar{E}_{v,i}(t)|$. Now from (19) we have

$x_v(t_{i+1}) - \bar{x}_{v,i+1} = x_v(t_{i+1}) - \bar{x}_{v,i}(t_{i+1}) + \tau_{v,i} + \epsilon_{v,i}$, whence

$$(25) \quad \bar{E}_{v,i+1}(t_{i+1}) = \bar{E}_{v,i}(t_{i+1}) + \tau_{v,i} + \epsilon_{v,i}.$$

Let $\epsilon = \max_{v,i} |\epsilon_{v,i}|$, $\tau = \max_{v,i} |\tau_{v,i}|$. Then, using (24),

$$(26) \quad \bar{E}_{i+1}(t_{i+1}) \leq \bar{E}_i(t_i) e^{nLh} + \tau + \epsilon.$$

Now initially $\bar{X}_{v,0} + \eta_v = x_v(t_0)$, where η_v is the error in the initial data $\bar{X}_{v,0}$. The η_v 's may be merely the round-off errors committed in replacing the exact initial data by s -place machine numbers, or they may be due to other sources of error such as measurement in experiments in those cases where the initial value problem directly represents a physical phenomenon. In any event, we have

$$(27) \quad \bar{E}_0(t_0) = x_v(t_0) - \bar{X}_{v,0} = \eta_v.$$

Letting $\eta = \max |\eta_v|$, repeated application of (26) yields

$$(28) \quad \bar{E}_{i+1}(t_{i+1}) \leq (\tau + \epsilon) (1 + e^{nLh} + \dots + e^{inLh}) + e^{(i+1)nLh} \eta,$$

So that, for fixed $t_{i+1} = t_0 + (i+1)h$:

$$(29) \quad \bar{E}_{i+1}(t_{i+1}) \leq \frac{e^{nL(t_{i+1}-t_0)} - 1}{nLh} (\tau + \epsilon) + e^{nL(t_{i+1}-t_0)} \eta.$$

We recall that τ is the local truncation error, $\tau = O(h^{k+1})$ for a k 'th order integrating method, and where, for M1 for example,

$$(30) \quad \epsilon \leq \frac{1}{2} \beta^{-s} + h \max_{v,i} |\bar{f}_{v,i} - f_{v,i}| + \max_{v,i} |f_{v,i}| |h-h|$$

as in (21). If the initial data $\bar{X}_{v,0}$ are taken to be exact (i.e. $\eta = 0$), and if ϵ is small compared to τ , then for a fixed h a bound such as (17) for $E_i(t_i)$ might also be good for $\bar{E}_i(t_i)$. However, suppose we carry a fixed number, s , of places in a sequence of computations of $\bar{X}_{v,i}$, decreasing h in the sequence. Then $\lim_{h \rightarrow 0} \tau = 0$, whence by (17) $\lim_{h \rightarrow 0} \bar{E}_i(t_i) = 0$.

On the other hand, (30) shows that ϵ cannot be expected to approach 0, whence (29) shows that $\bar{E}_i(t_i)$ might conceivably grow indefinitely with decreasing h . In fact, then, one should set a lower limit on h by letting the bound in (29) be minimum, i.e. take no h smaller than that making

$$\frac{\tau + \epsilon}{h} \text{ minimum. Further decrease in } h \text{ would not only increase the}$$

labor of computation, but would produce worse results as well. Now for

a k 'th order integrating procedure $\frac{\tau + \epsilon}{h}$ behaves like $Ch^k + \epsilon_0/h$ for sufficiently small h , where C and ϵ_0 (= say, $\lim_{h \rightarrow 0} \epsilon$) are independent of h . This function has a minimum when $h = (\frac{\epsilon_0}{kC})^{1/(k+1)}$, which should

be the smallest value used. Note that, unlike the usual criterion (that it is useless to take h so small that the local truncation is less than the local round-off) this permits smaller values of h , namely down to

$$\tau = Ch^{k+1} = \frac{\epsilon_0}{k}.$$

2.4 Variation of Step Size from Step to Step.

For simplicity the analysis of the preceding section was carried through for step size, h , fixed throughout the range of integration. However, as was mentioned in 2.2, it may be more efficient computationally to vary the step size within the interval for a given problem. We have just discussed a lower limit for h . An upper limit is required to keep the local truncation error small enough. In a given problem we may gain markedly in efficiency by varying the step size between these.

As an example, one procedure for varying h which is sometimes useful is the following: (we restrict the examples to first order systems - the extensions to higher order being obvious) for

$$(1) \quad \frac{dx}{dt} = f(t, x), \quad x(t_0) = x_0,$$

choose

$$(2) \quad h_i = t_{i+1} - t_i = \frac{\delta}{\sqrt{1 + f^2(t_i, X_i)}},$$

where X_i is the numerical result for $x(t_i)$ and δ is fixed to make h lie between the limits discussed above. This method spaces (t_i, X_i) at approximately equal intervals of arc-length. Where the slope of the integral curve is large, i.e. when x is changing rapidly, small steps are taken, whereas for slowly varying x , where the curve is flat, larger steps are taken. When f varies widely considerable efficiency can be gained by not keeping the step size always small. Essentially the same results as with (2) are obtained by

$$(3) \quad h_i = \frac{\delta}{1 + |f(t_i, X_i)|},$$

which is computationally simpler.

Another procedure one might use is that of taking

$$(4) \quad h_i = \frac{\delta}{1 + |f(t_i, X_i)| + \left| \frac{\partial f(t_i, X_i)}{\partial x} \right|}.$$

This would take account not only of changes of slope of the integral

curves but also of changes in the rate of separation of neighboring integral curves. This method can be modified to avoid computing $\frac{\partial f}{\partial x}$ as follows:

$$(5) \quad h_i = \frac{\delta}{1 + \left| f(t_i, X_i) \right| + \left| \frac{f(t_i, X_i) - f(t_i, X_i^*)}{X_i - X_i^*} \right|} \quad)$$

where $X_i^* \neq X_i$, say $X_i^* = X_i + \delta$.

3.1 A Bounding Procedure for Euler's Method

We begin by describing how Euler's method may be modified to yield a bounding procedure. Let the subscript 0 applied to f refer to evaluation at (x_0, y_0) . From

$$(1) \quad y'' = f(x, y), \quad \text{we find}$$

$$(2) \quad \begin{aligned} y'' &= f' = f_x + y' f_y = f_x + f f_y \\ y''' &= f'' = f_{xx} + 2 f f_{xy} + f^2 f_{yy} + f_x f_y + f f_y^2. \end{aligned}$$

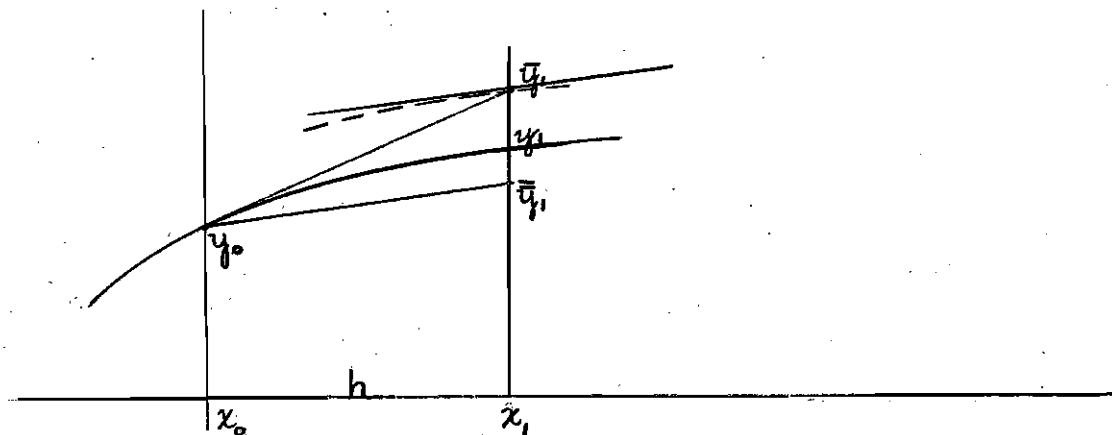


Figure 3

Taylor expansions of y and $f(x, y)$ are, therefore

$$(3) \quad \begin{cases} y = y_0 + h f_0 + \frac{h^2}{2} f'_0 + \frac{h^3}{6} f''_0 + \dots \\ f(x, y) = f_0 + h f_{x_0} + k f_{y_0} + \frac{1}{2} \{ h^2 f_{xx_0} + 2 h k f_{xy_0} + k^2 f_{yy_0} \} + \dots, \end{cases}$$

where

$$(4) \quad \begin{cases} h = x - x_0 \\ k = y - y_0. \end{cases}$$

The Euler approximation at $x_1 = x_0 + h_1$ is

$$(5) \quad \bar{y}_1 = y_0 + h_1 f_0.$$

A value $\bar{\bar{y}}_1$ such that y_1 is bracketed by it and $\bar{y}_1$ is, for small enough h_1 and an unexceptional (x_0, y_0) obtained (see figure 3) by drawing through (x_0, y_0) a line parallel to the direction at $(x_1, \bar{y}_1)$. The condition on (x_0, y_0) for the existence of such an h_1 , and the limitation on h_1 will be discussed below. Using (3) and (5): (note that $k_1 = h_1 f_0$)

$$(6) \quad \begin{cases} \bar{\bar{y}}_1 = y_0 + h_1 f(x_1, \bar{y}_1) \\ = y_0 + h_1 \left\{ f_0 + h_1 f_{x_0} + h_1 f_0 f_{y_0} + \frac{1}{2} [h_1^2 f_{xx_0} + 2h_1^2 f_{xy_0} + h_1^2 f_{yy_0}] + \dots \right\} \\ = y_0 + h_1 f_0 + \frac{h_1^2}{2} f_0' + \frac{1}{2} h_1^3 (f_0'' - f_{y_0} f_0') + \dots \end{cases}$$

From (5) and (6) we find

$$(7) \quad \frac{1}{2} (\bar{y}_1 + \bar{\bar{y}}_1) = y_0 + h_1 f_0 + \frac{h_1^2}{2} f_0' + \frac{h_1^3}{4} (f_0'' - f_{y_0} f_0') + \dots$$

Thus, on the one hand

$$(8) \quad \frac{1}{2} (\bar{\bar{y}} - \bar{y}) = \frac{h_1^2}{2} f_0' + \frac{h_1^3}{4} (f_0'' - f_{y_0} f_0') + \dots,$$

while, on the other

$$(9) \quad \frac{1}{2} (\bar{y}_1 + \bar{\bar{y}}_1) - y_1 = \frac{h_1^3}{12} (f_0'' - 3f_{y_0} f_0') + \dots$$

Consequently, if there is no point of inflection at (x_0, y_0) , i.e.

if

$$(10) \quad f_0' \neq 0,$$

$\frac{1}{2} (\bar{y}_1 + \bar{\bar{y}}_1)$ is a third order approximation to y_1 , while $\bar{\bar{y}}_1 - \bar{y}_1$ is only of second order. This is what we meant above when we said that the modification of the Euler procedure is relatively coarse. Notice, however, that $\frac{1}{2} (\bar{y}_1 + \bar{\bar{y}}_1)$ is a third order approximation whether $\bar{y}_1$ and $\bar{\bar{y}}_1$ bracket y_1 or not. This well-known approximation is the result of one iteration in the so-called modified Euler integrating method. It is also known as the Heun method.

Now let us find the condition on h_1 whereby this bracketing is assured at one step. Suppose that in a region R containing all the points (x, y) in which we are interested,

$$(11) \quad \left\{ \begin{array}{l} |f| \leq M, \\ M^{j-1} \left| f_{x^i y^j} \right| \end{array} \right\} \frac{1}{i+j} \leq L,$$

where i and j run through all integral values for which the corresponding partial derivatives are of interest. (This notation is similar to that in Bieberbach Differential-Gleichungen p. 55). In the present case

$$(12) \quad \left\{ \begin{array}{l} 0 \leq i \leq 2 \\ 0 \leq j \leq 2 \\ i + j \leq 2. \end{array} \right.$$

From (2) and (11) we therefore have in R

$$(13) \quad \left\{ \begin{array}{l} |f'| \leq 2LM \\ |f''| \leq 6L^2M. \end{array} \right.$$

Then, if at x_0 we take

$$(14) \quad h < \frac{|f'_0|}{10L^2M},$$

we will have y_1 bracketed by $\bar{y}_1$ and $\underline{\bar{y}}_1$. To prove this, observe that by using the inequality for f' and f'' from (13) in (3) and (6), we have, in addition to

$$\bar{y}_1 = y_0 + h_1 f_0,$$

$$\text{that } y_0 + h_1 f_0 + \frac{h_1^2}{2} f'_0 - h_1^3 L^2 M < y_1 < y_0 + h_1 f_0 + \frac{h_1^2}{2} f'_0 + h_1^3 L^2 M,$$

$$\text{and } y_0 + h_1 f_0 + h_1^2 f'_0 - 4h_1^3 L^2 M < \underline{\bar{y}}_1 < y_0 + h_1 f_0 + h_1^2 f'_0 + 4h_1^3 L^2 M;$$

if, then $f'_0 > 0$, it follows from several applications of (14) that

$$\begin{aligned} \bar{y}_1 &= y_0 + h_1 f_0 < y_0 + h_1 f_0 + \frac{h_1^2}{2} f'_0 - h_1^3 L^2 M \\ &< y_1 < y_0 + h_1 f_0 + \frac{h_1^2}{2} f'_0 + h_1^3 L^2 M \\ &< y_0 + h_1 f_0 + h_1^2 f'_0 - 4h_1^3 L^2 M < \underline{\bar{y}}_1 \end{aligned}$$

On the other hand, if $f'_0 < 0$, the same procedure yields

$$\begin{aligned} \bar{\bar{y}}_1 &< y_0 + h_1 f_0 + h_1^2 f'_0 + 4h_1^3 L^2 M < y_0 + h_1 f_0 + \frac{h_1^2}{2} f'_0 - h_1^3 L^2 M \\ &< y_1 < y_0 + h_1 f_0 + \frac{h_1^2}{2} f'_0 + h_1^3 L^2 M \\ &< y_0 + h_1 f_0 - 4h_1^3 L^2 M < \bar{y}_1. \end{aligned} \quad \text{q.e.d.}$$

If condition (14) is not fulfilled at a given point for a given mesh size h_1 , we have remarked as a result of equation (9) that

$\frac{1}{2}(\bar{y}_1 + \bar{\bar{y}}_1)$ is still a third order approximation to y_1 . By the same token equation (8) shows that $\frac{1}{2}(\bar{\bar{y}}_1 - \bar{y}_1)$ is still of the second order; the error estimate is still good even though bracketing may be lost.

Thus, if the initial point is near a point of inflection, the estimates are of the same order as above, but do not yield bracketing.

Suppose, now, that (14) has held for a number of steps beyond the initial point, so that at x_i , $\bar{y}_i$ and $\bar{\bar{y}}_i$ bracket y_i . Then even though (14) fail, so that bracketing of the solution curves with initial conditions y_i or $\bar{y}_i$ or $\bar{\bar{y}}_i$ would fail at x_{i+1} , we might still have y_{i+1} bracketed by $\bar{y}_{i+1}$ and $\bar{\bar{y}}_{i+1}$.

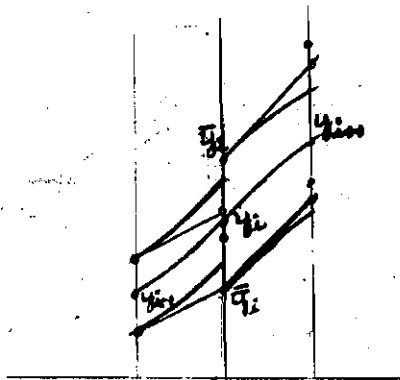


Figure 4

In other words (see figure 4), the number of steps might have been sufficient to yield a separation from the actual y_i by the bounds, which separation is large enough to counteract the local loss of bracketing. In still other words, the procedure might have had a good enough running start to get by the danger point without loss of bracketing. In line

with this idea we may proceed farther. We can give a condition on the separation, k , between y_0 and $\bar{y}_0$ such that at least one of the bounds at the next step, beginning at $\bar{y}_0$, still bounds y_1

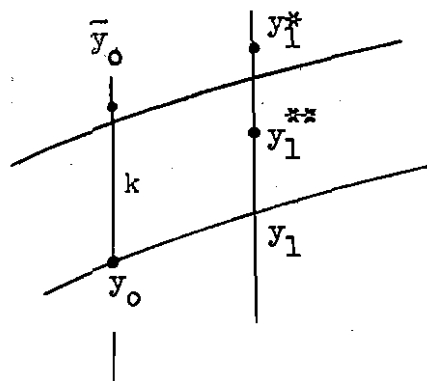


Figure 5

on the same side. The condition is

$$(15) \quad k > \frac{3h^2 ML}{1-hL-h^2L^2}, \quad (0 < hL < \left| \frac{\sqrt{5}-1}{2} \right|).$$

(The routine might be modified, if desired, to incorporate (15) as a test. This would ensure bracketing throughout). The proof is similar to that of (14):

$$\text{Since } \bar{y}_0 = y_0 + k,$$

$$\begin{aligned} y_1^* &= \bar{y}_0 + hf(x_0, \bar{y}_0) = y_0 + k + h \{f_0 + kf_y\} \\ &= y_0 + hf_0 + k \{1 + hf_y\}, \end{aligned}$$

where f_y is evaluated at x_0 and some $y = y_0 + k\theta_2$, $0 \leq \theta_2 \leq 1$. Thus,

since $x_1 = x_0 + h$,

$$\begin{aligned} y_1^{**} &= \bar{y}_0 + hf(x_1, y_1^*) \\ &= y_0 + k + h \{f_0 + hf_x^* + [hf_0 + k(1 + hf_y)] f_y^*\} \\ &= y_0 + hf_0 + h^2 (f_x^* + f_0 f_y^*) + k \{1 + [1 + hf_y] hf_y^*\}, \end{aligned}$$

$0 \leq \theta_2' \leq 1$. But $y_1 = y_0 + hf_0 + \frac{h^2}{2} f'$, where f' is evaluated at $x_0 + h\theta_1''$, y_0 . Consequently, if $y_1 < \max\{y_1^*, y_1^{**}\}$, we must have

$$k > \max \left\{ \frac{\frac{1}{2} h^2 f'}{1 + h f_y}, \frac{\frac{1}{2} h^2 [f' - 2 (f_x^* + f_y^*)]}{1 + h f_y^* [1 + h f_y]} \right\}$$

(as long as $hL < \frac{\sqrt{5}-1}{2}$). By using (11) and (13) on this we get (15).
q.e.d.

Clearly the same analysis would apply to $\bar{y}_0$ using $-k$ instead of k .

Note that if we begin at a point of inflection, we could, to preserve bracketing, use k in (15) to make our start from $y_0 = k$ and $y_0 \neq k$.

The logical flow chart for the above modification of Euler's method is the following:

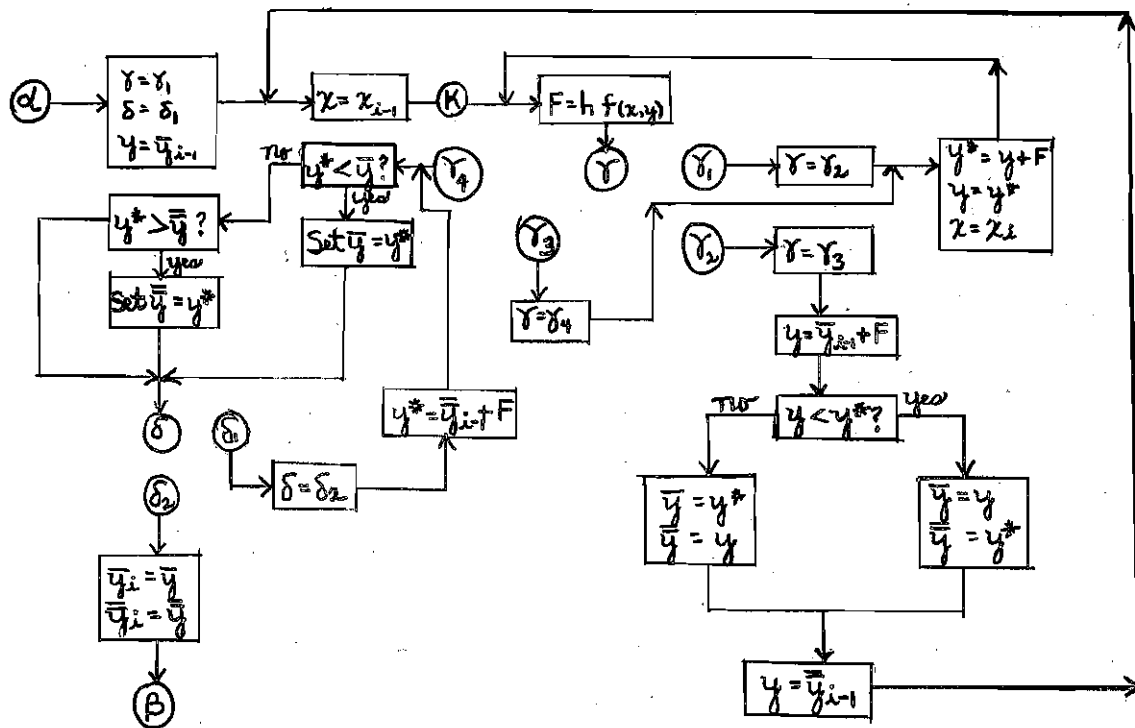


Figure 6

Logical Block Flow Chart for a Bounding Procedure on Euler's Method.

Here, because the F is variable in application the coding of (K) to (Y) should be last in sequence.

3.2 A General Error Estimating Procedure

We now approach a general method of modifying any single-step integrating procedure to produce simultaneously a higher order procedure and an error-estimating procedure. It will be approached in three stages. First there will be the already mentioned method of extrapolation to zero grid. Then this method will be reorganized to be applied locally at each step. Finally this local extrapolation to zero grid will be modified to yield bounds at each step if the grid size is small enough; we have seen in the introduction how this would yield error bounds at each step. In practice it will not matter whether the grid size is small enough to yield strict bounds, since the effect will be to produce a higher order procedure in any event, and to change the error bounds to error estimates.

Let us, then, begin by recalling the method of extrapolation to zero grid, with a quick proof which will be made more precise in our application.

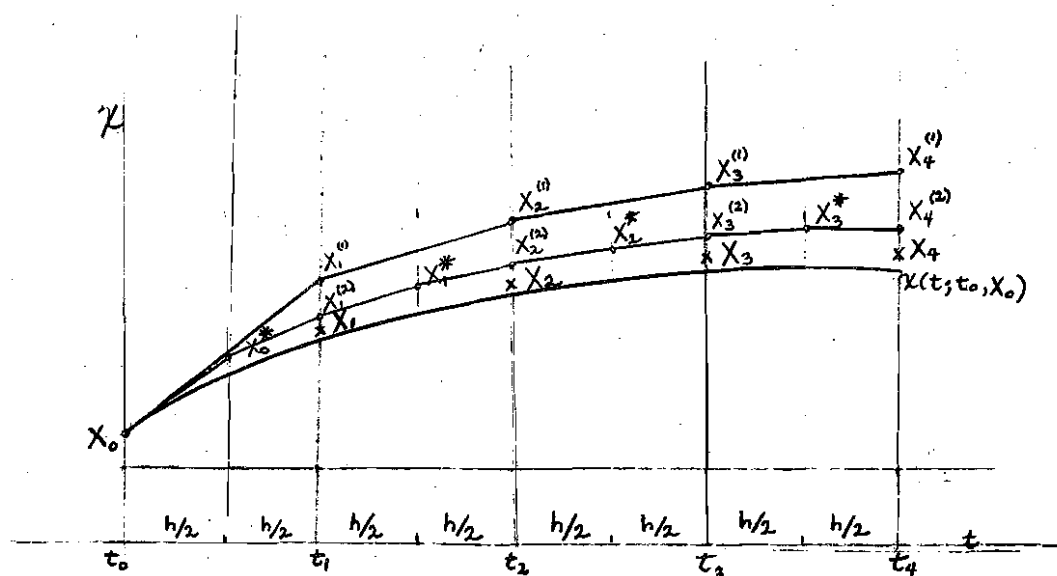


Figure 7

Extrapolation to Zero Grid in the Large

Given a single step integrating procedure which approximates the exact solution (we will omit, for brevity, the subscripts for the different dependent variables until we set up the final procedure) with initial condition $x = X_0$ at $t = t_0$, $x(t; t_0, X_0)$ by the values

(1) $X_1^{(1)}$ at $t_1 = t_0 + h$, $X_2^{(1)}$ at $t_2 = t_1 + h$, etc., the solution process was run through again with grid size, say, $h/2$ to yield the values X_0^* at $t_0 + h/2$, $X_1^{(2)}$ at t_1 , X_1^* at $t_1 + h/2$, $X_2^{(2)}$ at t_2 , etc. (See fig. 7). A better approximation to $x(t_i; t_0, X_0)$ is then given by the formula

$$\frac{2^k X_i^{(2)} - X_i^{(1)}}{2^k - 1}. \text{ We have seen in section 2.3 (equation (18)) that a}$$

"k'th order integrating procedure", whose error in the small is of order $k + 1$, actually has an error in the large of order k . Consequently, with an extra differentiability condition, which will be given precisely below, we may write:

$$(1) \quad \begin{aligned} X_i^{(1)} &= x(t_i; t_0, X_0) + O(h^k) \\ &= x(t_i; t_0, X_0) + C(t_i, X_0) h^k + O(h^{k+1}). \end{aligned}$$

Applying this, using grid size $h/2$, therefore yields:

$$(2) \quad X_i^{(2)} = x(t_i; t_0, X_0) + C(t_i, X_0) \frac{h^k}{2^k} + O(h^{k+1}).$$

It follows from (1) and (2) that an error estimate for $X_i^{(2)}$ is given by:

$$(3) \quad \Delta_i = \frac{X_i^{(2)} - X_i^{(1)}}{2^k - 1} = -C(t_i, X_0) \frac{h^k}{2^k} + O(h^{k+1}),$$

while a better estimate for $x(t_i; t_0, X_0)$ is given by

$$(4) \quad X_i = X_i^{(2)} + \Delta_i = \frac{2^k X_i^{(2)} - X_i^{(1)}}{2^k - 1} = x(t_i; t_0, X_0) + O(h^{k+1}).$$

This, by now, classical method therefore yields either an error estimate for $X_i^{(2)}$ or a method of one higher order for $x(t_i; t_0, X_0)$.

In payment for this one has two separate runs, the second of which has twice as many computations; thus in addition to the computations (3) and (4), the running time is multiplied by three. Also the values of the first run, $X_1^{(1)}$, $X_2^{(1)}$, ... must be stored until such time, during or after the second run, as they are used in computations (3) and (4). This last storage requirement was not an important factor in hand computations where the work sheets provided the storage. In high speed machine computation it is a nuisance. Let us, then, as a next step to what we are

after (an error bounding or estimating method to apply to the resulting values) see what the effect would be if the extrapolation method were reorganized to be applied at each step.

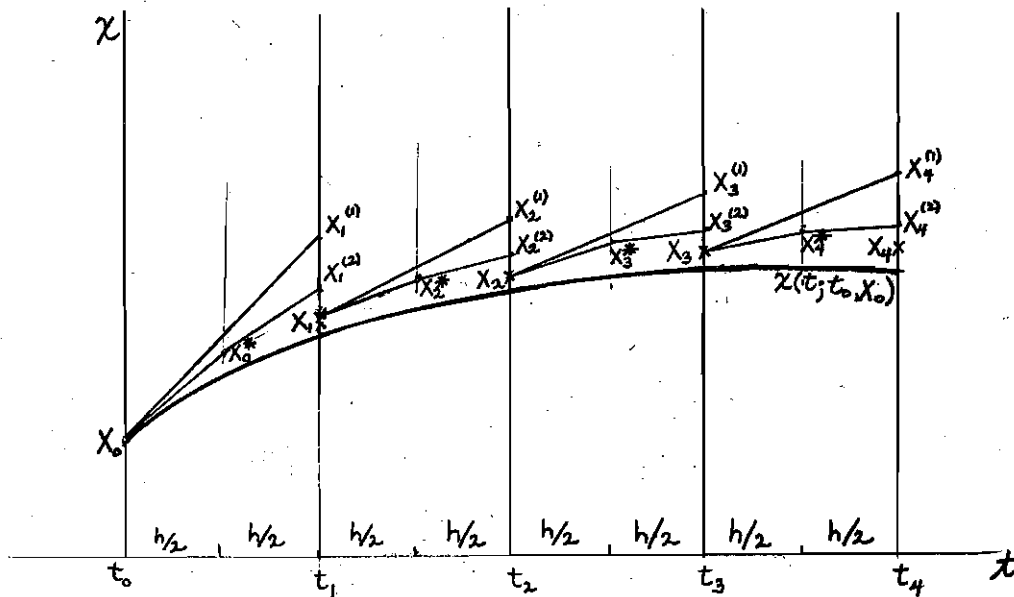


Figure 8

Extrapolation to Zero Grid applied Locally at each Step

In this procedure, given X_{i-1} , one computes $X_i^{(1)}$ and $X_i^{(2)}$ from the integrating method, then one computes Δ_i from the first equation in (3), after which one computes X_i from the first equation in (4), and one is ready to go on to the next step (see Fig. 8). Thus the whole process from X_{i-1} to X_i can be considered a new single step integrating procedure. If, therefore, we show that locally this new procedure is of one order higher than the one it is applied to, the general result in section 2.3 will then apply to yield the rest of equation (4). We must therefore repeat the above argument, applying it locally, and we take this opportunity to make it more precise.

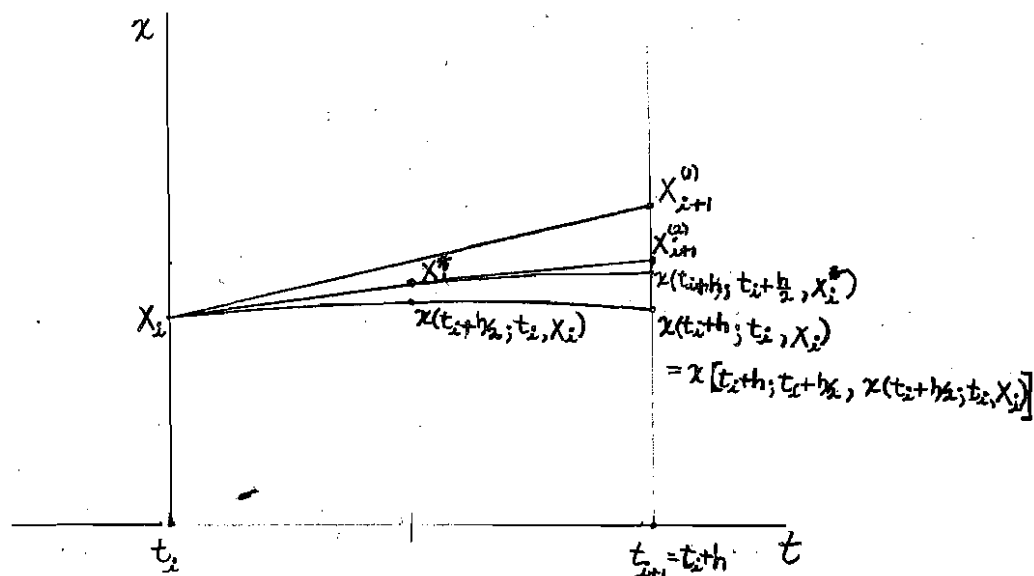


Figure 9

We assume that all computed values lie within some simply-connected region containing no singular points of the equations 2.1 (1), and further that the functions f_v in these equations belong to class C^{k+1} , i.e. have bounded, continuous partial derivatives of order $k+1$ throughout S . The first k derivatives are needed to obtain the error estimates for the k 'th order integrating procedure so that we may write equation 2.1 (4); the extra derivative permits us to analyze the error term further so that we have

$$(5) \quad X_{v,i+1}^{(1)} = x_v(t_i + h; t_i, X_{1i}, \dots, X_{ni}) + C_v(t_i, X_{1i}, \dots, X_{ni}) h^{k+1} + O(h^{k+2}),$$

where the functions C_v are of class C^1 . (It is now essential that we exhibit explicitly the subscript v for the dependent variables; see Fig. 9 for notation, otherwise): Applying this twice, with step size $h/2$, we also have:

$$(6) \quad X_{v,i}^* = x_v(t_i + \frac{h}{2}; t_i, X_{1i}, \dots, X_{ni}) + C_v(t_i, X_{1i}, \dots, X_{ni}) \frac{h^{k+1}}{2^{k+1}} + O(h^{k+2}),$$

$$(7) \quad X_{v,i+1}^{(2)} = x_v(t_i + h; t_i + \frac{h}{2}, X_{1i}^*, \dots, X_{ni}^*) + C_v(t_i + \frac{h}{2}, X_{1i}^*, \dots, X_{ni}^*) \frac{h^{k+1}}{2^{k+1}} + O(h^{k+2}).$$

Now, by definition,

$$(8) \quad x_v(t_0; t_0, y_1, \dots, y_n) = y_v,$$

hence, for some value between t_i and $t_i + \frac{h}{2}$ we have

$$(9) \quad x_v(t_i + \frac{h}{2}; t_i, x_{1i}, \dots, x_{ni}) = x_{vi} + \frac{h}{2} \frac{\overline{dx}_v}{dt} = x_{vi} + \frac{h}{2} \bar{f}_v = x_{vi} + O(h).$$

Consequently, applying (9) to (6):

$$(10) \quad x_{v,i}^* = x_{v,i} + O(h),$$

whence

$$(11) \quad C_v(t_i + \frac{h}{2}, x_{1i}^*, \dots, x_{ni}^*) = C_v(t_i, x_{1i}, \dots, x_{ni}) + O(h)$$

(This is where differentiability assumptions on C_v , and hence on the f_v are made).

Applying (11) to (7) now yields the first simplification in $X_{v,i+1}^{(2)}$:

$$(12) \quad X_{v,i+1}^{(2)} = x_v(t_i + h; t_i + \frac{h}{2}, x_{1i}^*, \dots, x_{ni}^*) + C_v(t_i, x_{1i}, \dots, x_{ni}) \frac{h^{k+1}}{2^{k+1}} + O(h^{k+2}).$$

In order to be able to use this with (5) in the same way that (2) was used with (1), we must analyze the first term.

Now, in general, because of (8), for some mean $\bar{t}$,

$$(13) \quad x_v(t; t_0, y_1, \dots, y_n) = y_v + (t-t_0) \left. \frac{\partial x_v}{\partial t} \right|_{t=\bar{t}}$$

Hence,

$$(14) \quad \frac{\partial x_v}{\partial y_\mu} = \delta_{v\mu} + (t-t_0) \frac{\partial}{\partial y_\mu} \left[\left. \frac{\partial x_v}{\partial t} \right|_{t=\bar{t}} \right],$$

where the Kronecker $\delta_{v\mu}$ means 1 if $v = \mu$ and 0 otherwise. It follows, using (14), that

$$(15) \quad \begin{aligned} x_v(t; t_0, \bar{y}_1, \dots, \bar{y}_n) &= x_v(t; t_0, y_1, \dots, y_n) + \sum_{\mu=1}^n (\bar{y}_\mu - y_\mu) \frac{\partial x_v}{\partial y_\mu} \\ &= x_v(t; t_0, y_1, \dots, y_n) + (\bar{y}_v - y_v) + (t-t_0) \sum_{\mu=1}^n (\bar{y}_\mu - y_\mu) \frac{\partial}{\partial y_\mu} \left[\left. \frac{\partial x_v}{\partial t} \right|_{t=\bar{t}} \right]. \end{aligned}$$

Now let $t = t_i + h$, $t_0 = t_i + \frac{h}{2}$, $\bar{y}_\mu = x_{\mu i}^*$,

and $y_\mu = x_\mu(t_i + \frac{h}{2}; t_i, X_{1i}, \dots, X_{ni})$,

remembering that $x_v[t_i + h; t_i + \frac{h}{2}, x_1(t_i + \frac{h}{2}; t_i, X_{1i}, \dots, X_{ni}), \dots,$

$x_n(t_i + \frac{h}{2}; t_i, X_{1i}, \dots, X_{ni})] = x_v(t_i + h; t_i, X_{1i}, \dots, X_{ni})$ by definition,

and that $\bar{y}_\mu - y_\mu = C_\mu(t_i, X_{1i}, \dots, X_{ni}) \frac{h^{k+1}}{2^{k+1}} + O(h^{k+2}) = O(h^{k+1})$,

and $t - t_0 = O(h)$. (15) then becomes:

$$(16) \quad x_v(t_i + h; t_i + \frac{h}{2}, X_{1i}^*, \dots, X_{ni}^*) = x_v(t_i + h; t_i, X_{1i}, \dots, X_{ni}) \\ + C_v(t_i, X_{1i}, \dots, X_{ni}) \frac{h^{k+1}}{2^{k+1}} + O(h^{k+2}).$$

Substituting this into (12) now yields

$$(17) \quad X_{v,i+1}^{(2)} = x_v(t_i + h; t_i, X_{1i}, \dots, X_{ni}) + C_v(t_i, X_{1i}, \dots, X_{ni}) \frac{h^{k+1}}{2^k} \\ + O(h^{k+2}).$$

The proof now proceeds as for the case in the large: from (17) and (5) an error estimate for $X_{v,i+1}^{(2)}$ is given by

$$(18) \quad \Delta_{v,i+1} = \frac{X_{v,i+1}^{(2)} - X_{v,i+1}^{(1)}}{2^k - 1} = -C_v(t_i, X_i) \frac{h^{k+1}}{2^k} + O(h^{k+2}),$$

while an improved approximation for $x_v(t_i + h; t_i, X_{1i}, \dots, X_{ni})$ is

$$(19) \quad X_{v,i+1} = X_{v,i+1}^{(2)} + \Delta_{v,i+1} = \frac{2^k X_{v,i+1}^{(2)} - X_{v,i+1}^{(1)}}{2^k - 1} \\ = x_v(t_i + h; t_i, X_{1i}, \dots, X_{ni}) + O(h^{k+2})$$

Equation (4) now follows, in the large, as described above. In this procedure we have increased the order of the given integrating procedure, but are now without any sharp error estimates. We make the final modification by noting that $\bar{X}_{i+1}$ locally approximates the solution to one order higher than Δ_{i+1} approximates the last correction; thus, for a normal starting point X_0 we should have $\bar{X}_1 = X_1 + |\Delta_1| = X_1^{(2)} + \Delta_1 + |\Delta_1|$ and

$\underline{X}_1 = X_1 - |\Delta_1| = X_1^{(2)} + \Delta_1 - |\Delta_1|$ bounding the solution above and below for a sufficiently small h ; thenceforth one uses $\bar{X}_i$ and $\underline{X}_i$ as initial values for the next step, computes corresponding $\bar{\Delta}_{i+1}$ and $\underline{\Delta}_{i+1}$ and $\bar{X}_{i+1} = \bar{X}_{i+1}^{(2)} + \bar{\Delta}_{i+1} + |\bar{\Delta}_{i+1}|$, $\underline{X}_{i+1} = \underline{X}_{i+1}^{(2)} + \underline{\Delta}_{i+1} - |\underline{\Delta}_{i+1}|$

(see figure 10). In other words,

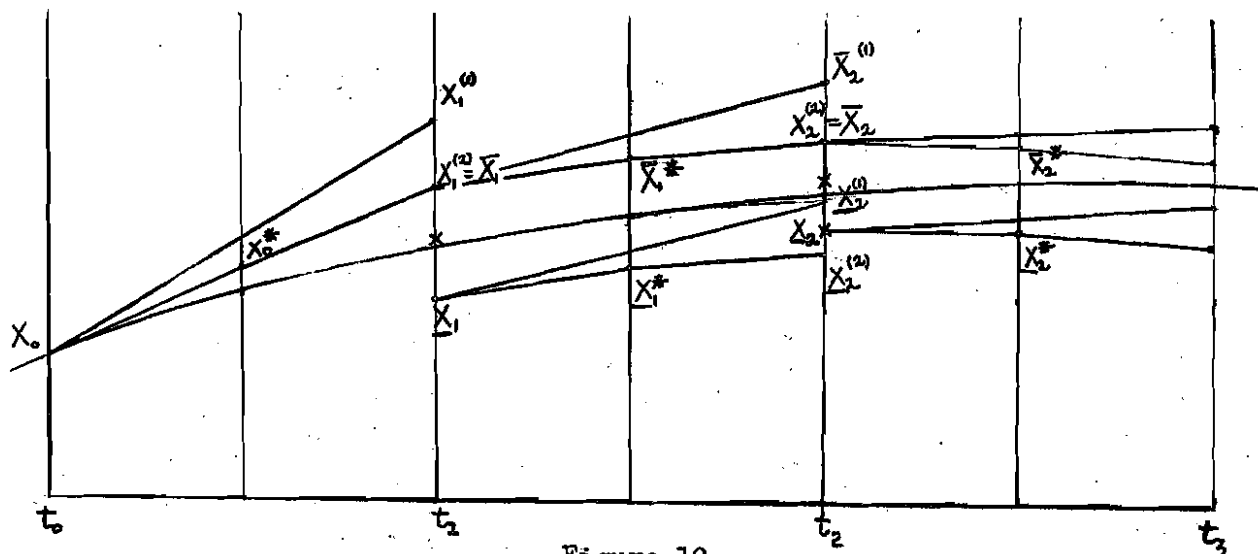


Figure 10

$$(20) \quad \underline{X}_{v,0} = \bar{X}_{v,0} = x_{v,0} \quad ,$$

$$(21) \quad \underline{\Delta}_{v,i} = \frac{\underline{X}_{v,i}^{(2)} - \underline{X}_{v,i}^{(1)}}{2^k - 1} \quad ,$$

$$(22) \quad \bar{\Delta}_{v,i} = \frac{\bar{X}_{v,i}^{(2)} - \bar{X}_{v,i}^{(1)}}{2^{k-1}} \quad ,$$

$$(23) \quad \underline{X}_{v,i} = \underline{X}_{v,i}^{(2)} + \underline{\Delta}_{v,i} - |\underline{\Delta}_{v,i}|,$$

$$(24) \quad \overline{X}_{v,i} = \overline{X}_{v,i}^{(2)} + \overline{\Delta}_{v,i} + |\overline{\Delta}_{v,i}|,$$

are the formulae for the new procedure, where local extrapolation to zero grid on $\underline{X}_{v,i-1}$ yield $\underline{X}_{v,i}^{(1)}$, $\underline{X}_{v,i}^{(2)}$, and $\underline{\Delta}_{v,i}$, etc. We must show that, assuming the basic integrating procedure and the above computations carried out with infinite precision, we will have:

$$(25) \quad \frac{\overline{X}_{v,i} + \underline{X}_{v,i}}{2} = x_v(t_i; t_0, X_0) + O(h^{k+1}),$$

$$(26) \quad \frac{\overline{X}_{v,i} - \underline{X}_{v,i}}{2} = O(h^k),$$

and there exists an h^* such that

$$(27) \quad \underline{X}_{v,i} \leq x_v(t_i; t_0, X_0) \leq \overline{X}_{v,i} \quad \text{for } h \leq h^*.$$

Once we have shown these, the errors $\epsilon_{v,i}$ described in paragraph 2.3 can be taken into account. Equation (27) will be discussed in greater detail below. It can be derived immediately from (25) and (26) as follows:

$$\text{Since } \frac{\overline{X}_{v,i} + \underline{X}_{v,i}}{2} - x_v(t_i; t_0, X_0) \text{ is of higher order than } \frac{\overline{X}_{v,i} - \underline{X}_{v,i}}{2},$$

for any $C > 0$ there exists an h_c such that for $h = t_{j+1} - t_j \leq h_c$

($j = 0, 1, \dots, i-1$) we have

$$\left| \frac{\overline{X}_{v,i} + \underline{X}_{v,i}}{2} - x_v(t_i; t_0, X_0) \right| < C \left| \frac{\overline{X}_{v,i} - \underline{X}_{v,i}}{2} \right|.$$

Consequently for $h \leq h_c$:

$$\begin{aligned} \frac{\overline{X}_{v,i} + \underline{X}_{v,i}}{2} - C \frac{\overline{X}_{v,i} - \underline{X}_{v,i}}{2} &\leq x_v(t_i; t_0, X_0) \leq \frac{\overline{X}_{v,i} + \underline{X}_{v,i}}{2} + \\ &+ C \frac{\overline{X}_{v,i} - \underline{X}_{v,i}}{2}. \end{aligned}$$

Taking $C = 1$ yields (27).

To prove (25) and (26), let us apply equation (18) to the definitions, (21) and (22):

$$(28) \quad \begin{aligned} \underline{\Delta}_{v,i+1} &= -C_v(t_i, \underline{X}_i) \frac{h^{k+1}}{2^k} + O(h^{k+2}) \\ \overline{\Delta}_{v,i+1} &= -C_v(t_i, \overline{X}_i) \frac{h^{k+1}}{2^k} + O(h^{k+2}). \end{aligned}$$

By applying this and (19) to the definitions (23) and (24) we find:

$$(29) \quad \begin{aligned} \underline{X}_{v,i+1} &= x(t_{i+1}; t_i, \underline{X}_{1i}, \dots, \underline{X}_{ni}) - |C_v(t_i, \underline{X}_i)| \frac{h^{k+1}}{2^k} + O(h^{k+2}) \\ \overline{X}_{v,i+1} &= x(t_{i+1}; t_i, \overline{X}_{1i}, \dots, \overline{X}_{ni}) + |C_v(t_i, \overline{X}_i)| \frac{h^{k+1}}{2^k} + O(h^{k+2}). \end{aligned}$$

Now, just from the cruder information in (29) that the errors are $O(h^{k+1})$, it follows by use of the result in section 2.3, just as equation (5) of 2.1 followed from (4), and from definition (20) above that

$$(30) \quad \underline{X}_{v,i+1} = x(t_{i+1}; t_0, x_{10}, \dots, x_{n0}) + O(h^k) = \overline{X}_{v,i+1}.$$

Equation (26) is an immediate consequence of this. Since we now know that $\underline{X}_{v,i+1}$ and $\overline{X}_{v,i+1}$ differ by a quantity of order k , the second equation of (29), using the fact that C_v are of class C^1 , becomes

$$\overline{X}_{v,i+1} = x(t_{i+1}; t_i, \overline{X}_{1i}, \dots, \overline{X}_{ni}) + |C_v(t_i, \underline{X}_i)| \frac{h^{k+1}}{2^k} + O(h^{k+2}).$$

Together with the first equation of (29) this yields:

$$(31) \quad \frac{\overline{X}_{v,i+1} + \underline{X}_{v,i+1}}{2} = \frac{x_v(t_{i+1}; t_i, \overline{X}_i) + x_v(t_{i+1}; t_i, \underline{X}_i)}{2} + O(h^{k+2}).$$

The proof at the beginning of paragraph 2.3, when applied to $\frac{x_v(t; t_i, \overline{X}_i) + x_v(t; t_i, \underline{X}_i)}{2}$ instead of the $x_{v,i}(t)$ in (2) of 2.3 and

using $\frac{1}{2} \left\{ \frac{\partial f_v}{\partial x_\mu} \Big|_{\overline{p}} + \frac{\partial f_v}{\partial x_\mu} \Big|_{\underline{p}} \right\}$ in (4) of 2.3 then yields:

$$\begin{aligned}\frac{\bar{X}_{v,i} + \underline{X}_{v,i}}{2} &= \frac{x_v(t_i; t_0, \bar{X}_0) + x_v(t_i; t_0, \underline{X}_0)}{2} + O(h^{k+1}) \\ &= x_v(t_i; t_0, x_{v0}) + O(h^{k+1}),\end{aligned}$$

which is (25) as desired.

The logical flow chart for this final procedure is the following:

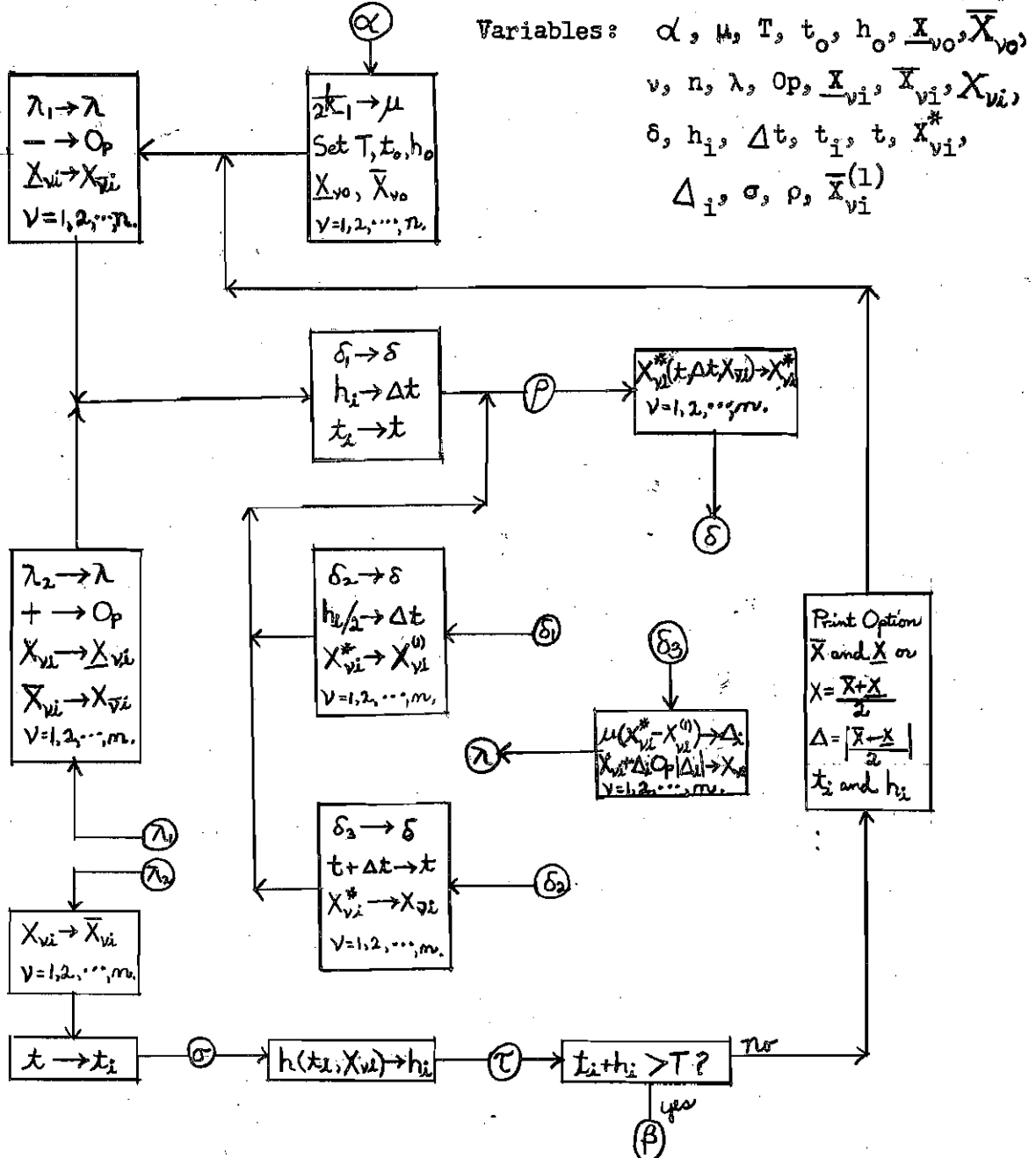


Figure 11

Logical Block Flow Chart for General Error Estimating Procedure

Note that the section from ρ to δ includes the integrating procedure. This section should therefore be placed last in a coded sequence, for we saw in section 2.2 how much it can vary in complexity. By the same token, in the coding of such general integrating procedures, the evaluation of the f_v should be placed last.

If the procedure is to be adapted to take into consideration the errors, $\epsilon_{v,i}$, described in section 2.3, one should complete the analysis of the $\epsilon_{v,i}$ in the particular integrating procedure and system of equations by finding a bound: $|\epsilon_{v,i}| \leq B_{v,i}$ and, in place of (23) and (24), use

$$(33) \quad \underline{x}_{vi} = \underline{x}_{vi}^{(2)} + \underline{\Delta}_{vi} - |\underline{\Delta}_{vi}| = B_{v,i-1},$$

$$(34) \quad \overline{x}_{vi} = \overline{x}_{vi}^{(2)} + \overline{\Delta}_{vi} + |\overline{\Delta}_{vi}| = B_{v,i-1}.$$

Also, errors in the initial conditions may be taken into account, if they are η_v and are bounded by H_v , by using

$$(35) \quad \underline{x}_{v0} = x_{v0} - H_v,$$

$$(36) \quad \overline{x}_{v0} = x_{v0} + H_v.$$

Due to the fact that these last local errors often vary in sign, in many practical problems the procedure (20) to (24) more than suffice to cover them.

At this point, for each integrating procedure an analysis like that leading to (14) and (15) in section 3.1 is possible as a supplement to equation (27). Such analyses increase in complexity with the order of the single step integrating procedure. For instance, for Euler's method the complexity of the derivation of these "sufficient conditions on h for bracketing" is not much greater than the derivation in section 3.1. We will not give it here, but will reiterate that most of the time sufficiently accurate error estimates rather than strict error bounds will be satisfactory.

3.3 Example and Conclusion

The error-control procedure of figure 2 and the error estimating procedure of figure 11, applied to the Runge-Kutta-Gill procedure, M6 of section 2.2, using constant grid size were applied to the following problem, a spinning top study being made by H. Reed of the Computing Laboratory:

$$\begin{aligned}
 (37) \quad \dot{u} &= \frac{1}{8} z, & u_0 &= \sqrt{15}/4, \\
 \dot{w} &= -\frac{1}{8} x, & w_0 &= 0, \\
 \dot{x} &= \frac{1}{4} z - wy, & x_0 &= \sqrt{15}/4, \\
 \dot{y} &= wx - uz, & y_0 &= 1/4, \\
 \dot{z} &= uy - \frac{1}{4} x, & z_0 &= 0, \\
 & & t_0 &= 0.
 \end{aligned}$$

As a check on the results the following identities, corresponding respectively to the conservation of energy, the constancy of the dynamical arm, and the conservation of spin can be used:

$$\begin{aligned}
 (38) \quad E &= u^2 + w^2 + y/4 \equiv 1, \\
 A &= x^2 + y^2 + z^2 \equiv 1, \\
 S &= ux + \frac{1}{4} y + wz \equiv 1,
 \end{aligned}$$

The Problem was run on Edvac. The maximum error permitted by the control was set at $\epsilon = 5 \times 10^{-6}$, and the minimum step size (which was never needed) was set at $\delta = 1/32$; the initial step size was taken at $\Delta t = 2$, since the first set of print-outs was required for this interval. The control reset Δt to 1 after one step, reset Δt to 1/2 after one step, and computed with this Δt up to $t = 2.5$ when the error estimate in z went out of bounds; it finally computed with $\Delta t = 1/4$ and was permitted to continue to beyond $t = 34$. Finally, the control was removed, and the integration plus the automatic error estimation was run with step size $\Delta t = \frac{1}{8}$ up to $T = 200$ with a running time of about three hours and print-outs of $t, u, w, x, y, z, e(u), e(w), e(x), e(y), e(z)$ for $t = 0$ (2) 10; 10 (10) 200. The error estimates fluctuated, never exceeding (in the print-outs) 36×10^{-7} . The computing time was found to be (not counting print-out time) 9 mins. 30 secs. per interval of $t = 10$, an average of 7 secs. per step of Runge-Kutta-Gill and automatic increase of order and error estimation for this fifth order system.

Sample results are tabulated below

	$t=30, \Delta t = 1/4$	$t=30, \Delta t = 1/8$	$t=200, \Delta t = 1/8$
u	-.81157642	-.81157644	+.98238136
e(u)	.00000222	.00000008	.00000040
w	+.56530414	+.56530407	+.13117951
e(w)	.00000262	.00000009	.00000099
x	-.74349391	-.74349431	+.96746748
e(x)	.00000076	.00000002	.00000048
y	+.08709987	+.08710019	+.07087767
e(y)	.00000140	.00000005	.00000257
z	+.66304646	+.66304581	+.24286412
e(z)	.00000352	.00000011	.00000030
E	1.00000002	1.00000005	1.00000610
A	1.00000019	.99999998	.99999995
S	.99999995	1.00000000	1.00000023

This example would indicate that very little practical purpose is served in deriving formulae corresponding to (14) of section 3.1 which would provide sufficient conditions for h to be small enough to make the error estimates actually error bounds. However, it is certainly conceivable that problems will arise for which they are necessary.

The extrapolation to zero grid method has been successfully applied to the initial value problem for hyperbolic partial differential equations. See, for example, R. F. Clippinger & N. Gerber [10], W. C. Carter & G. L. Spencer II, [11]. It seems reasonable to expect, then, that the automatic error estimation and control procedure last discussed can be extended to this type of problem.

Acknowledgements are due to Professor A. A. Bennett for detailed editorial and critical advice, and to M. Romanelli for the coding of the error control, error estimating, and Runge-Kutta-Gill routines for Edvac and their application to the spinning top study.

S. Gorn

S. Gorn

R. Moore

Bibliography

- [1] - F. J. Murray - Planning and Error Considerations for Numerical Solution of a System of Differential Equations, Etc. - Mathematical Tables and Other Aids to Computation - IV - 1950 - pp. 133-144.
- [2] - S. Gorn - On the Study of Computational Errors - BRL Report No. 816.
- [3] - L. F. Richardson and J. A. Gaunt - The Deferred Approach to the Limit - Part I - Single Lattice - Phil. Trans. Vol. 226 (1927) p. 299.
- [4] - Levy & Baggott - Numerical Solution of Differential Equations - Dover - 1950.
- [5] - Collatz - Numerische Behandlung von Differentialgleichungen - Springer - 1951.
- [6] - S. Gill - A Process for Step-by-Step Integration of Differential Equations on High Speed Computing Machines - Proc. Camb. Phil. Soc. 47 1950 - pp. 96-108.
- [7] - E. J. Nystrom - "Über die Numerische Integration von Differentialgleichungen" - Acta Soc. Scient. Fennicae - 50 - No. 13 (1925).
- [8] - E. Kamke - Differentialgleichungen Reeller Funktionen Leipzig - 1930.
- [9] - von Neumann & Goldstine - "Numerical Inverting of Matrices of High Order" - Bull. Am. Math. Soc. Vol. 53 No. 11.
- [10] - H. F. Clippinger & N. Gerber - Supersonic Flow over Bodies of Revolution - BRL Report No. 719.
- [11] - W. C. Carter & G. L. Spencer II - On the Numerical Solution of Hyperbolic Systems of Partial Differential Equations with Two Characteristic Directions - BRL Report No. 813.

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
4	Chief of Ordnance Department of the Army Washington 25, D. C. Attn: ORDTB - Bal Sec	2	Commanding Officer Naval Air Missile Test Center Point Mugu, California Attn: D. E. Dufford R. L. Causey
10	British - ORDTB for distribution	1	Commanding Officer Naval Aviation Ordnance Test Station Chincoteague, Virginia
4	Canadian Joint Staff - ORDTB for distribution	1	Commander Naval Air Development Center Johnsville, Pennsylvania
4	Chief, Bureau of Ordnance Department of the Navy Washington 25, D. C. Attn: Re3	1	Commanding General Air Force Missile Test Center Patrick Air Force Base Cocoa, Florida Attn: Technical Library (MTTL)
5	Commander Naval Proving Ground Dahlgren, Virginia Attn: C. G. Bramble	1	Chief of Staff U. S. Air Force The Pentagon Washington 25, D. C. Attn: AFAPA-3B J. V. Natrella
2	Commander Naval Ordnance Laboratory White Oak Silver Spring 19, Maryland	1	Commanding General Air Research & Development Command P. O. Box 1395 Baltimore 3, Maryland
2	Commander Naval Ordnance Test Station Inyokern P. O. China Lake, California Attn: Technical Library	4	Commanding General Wright Air Development Center Wright-Patterson Air Force Base, Ohio Attn: CRRM - 2 cys Aircraft Radiation Laboratory Attn: Mr. R. Rawhouser - 1 cy Aircraft Research Laboratory Attn: Dr. M. Scherberg Dr. F. Bubb
1	Superintendent Naval Postgraduate School Monterey, California		
1	Commanding Officer & Director Naval Research Laboratory Washington 25, D. C.		
1	Chief, Bureau of Aeronautics Department of the Navy Washington 25, D. C.	1	Director Air University Library Maxwell Air Force Base, Alabama

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
2	Director, Project RAND Department of the Air Force 1700 Main Street Santa Monica, California Attn: Herman Kahn Mario L. Juncosa	3	Commanding General Picatinny Arsenal Dover, New Jersey
5	Director Armed Services Technical Information Agency Documents Service Center Knott Building Dayton 2, Ohio Attn: DSC - SA	2	Commanding Officer Frankford Arsenal Philadelphia 37, Pennsylvania Attn: Ernest C. Casole, Head Math, Section Pitman Dunn Laboratory
3	Director National Advisory Committee for Aeronautics 1724 F Street, N. W. Washington 25, D. C.	1	Commanding Officer Watertown Arsenal Watertown 72, Massachusetts
1	Director National Advisory Committee for Aeronautics Langley Memorial Aeronautical Laboratory Langley Field, Virginia	1	Commanding General White Sands Proving Ground Las Cruces, New Mexico Attn: Flight Determination Laboratory
7	National Bureau of Standards National Applied Mathematics Laboratory Computation Laboratory Washington 25, D. C. Attn: Franz L. Alt Dr. A. J. Hoffman Dr. P. Davis I. Rhodes Dr. H. A. Antosiewicz Dr. G. Blanch John Todd	1	Los Alamos Scientific Laboratory Box 1663 Los Alamos, New Mexico
		3	Office of Ordnance Research Department of the Army Box CM, Duke Station Durham, North Carolina
		2	Atomic Energy Commission Washington 25, D. C.
		2	Atomic Energy Commission Oak Ridge, Tennessee Attn: A. S. Householder
		2	Operations Research Officer 6410 Connecticut Avenue Chevy Chase, Maryland Attn: Joseph O. Harrison, Jr.
1	Institute for Numerical Analysis National Bureau of Standards 405 Hilgard Avenue Los Angeles 24, California	1	Director U. S. Bureau of Mines 4800 Forbes Street Pittsburgh 13, Pennsylvania

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
3	Commanding Officer & Director David Taylor Model Basin Washington 7, D. C. Attn: J. V. Holberton A. Polchek	2	Watson Scientific Computing Laboratory 612 W. 116th Street New York 27, New York Attn: L. H. Thomas F. J. Murray
1	Applied Physics Laboratory 8621 Georgia Avenue Silver Spring, Maryland	1	Computation Laboratory Harvard University Cambridge 38, Massachusetts Attn: H. Aiken
2	Electronic Digital Computer Project University of Illinois Urbana, Illinois Attn: S. Gill D. J. Wheeler	2	Moore School of Electrical Engineering University of Pennsylvania Philadelphia, Pennsylvania
1	Department of Mathematics Montana State College Bozeman, Montana Attn: R. E. Lowney	2	Department of Mathematics Institute for Advanced Study Princeton, New Jersey Attn: J. von Neumann H. H. Goldstine
2	Willow Run Research Center University of Michigan Willow Run Airport Ypsilanti, Michigan Attn: Dr. John Carr, III, Reports Section	1	Air Weapons Research Center University of Chicago Chicago 37, Illinois
2	Digital Computer Laboratory Massachusetts Institute of Technology 211 Massachusetts Avenue Cambridge 39, Massachusetts	1	Institute for Cooperative Research Johns Hopkins University 2229 Charles Street Baltimore 2, Maryland Attn: E. R. C. Miles
1	Department of Mathematics University of Michigan Ann Arbor, Michigan Attn: T. H. Hildebrandt, Chairman	3	University of Delaware Newark, Delaware Attn: G. C. Webber R. Remage R. F. Jackson
1	Sandia Corporation Sandia Base P. O. Box 5800 Albuquerque, New Mexico	2	Institute of Applied Mathematics & Mechanics New York University New York, New York Attn: Dr. Fritz John Dr. E. Bromberg

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
1	Haller, Raymond & Brown, Inc. State College, Pennsylvania Attn: R. Raymond	1	Department of Mathematics North Carolina State College Raleigh, North Carolina Attn: John W. Cell
1	Department of Mathematics University of Illinois Urbana, Illinois Attn: A. H. Taub	1	Department of Mathematics Institute of Technology University of Minnesota Minneapolis, Minnesota
1	Department of Mathematics Harvard University Cambridge, Massachusetts Attn: Garrett Birkhoff	1	Purdue University Computation Laboratory Lafayette, Indiana Attn: A. J. Perlis
2	Department of Mathematics University of Maryland College Park, Maryland Attn: Monroe H. Martin David M. Young	1	University of Florida Statistical Laboratory, Building OE Gainesville, Florida Attn: Herbert A. Meyer
1	Department of Mathematics Oregon State College Corvallis, Oregon Attn: W. E. Milne	1	Professor A. A. Bennett Department of Mathematics Brown University Providence 12, Rhode Island
1	Department of Mathematics Georgetown University Washington 7, D. C. Attn: Malcolm Oliphant	2	Raytheon Manufacturing Company 100 River Road Waltham 54, Massachusetts Attn: Dr. W. C. Carter
1	Department of Mathematics Stanford University Stanford, California Attn: Gabor Szego	1	Engineering Research Associates 1902 W. Minnehaha Avenue St. Paul, Minnesota
1	Department of Mechanical Engineering Case Institute of Technology University Circle Cleveland 6, Ohio Attn: G. Kuerti	3	General Electric Company 1 River Road Schenectady 5, New York Attn: H. R. Koenig W. B. Jordon H. Poritsky

DISTRIBUTION LIST

<u>No. of Copies</u>	<u>Organization</u>	<u>No. of Copies</u>	<u>Organization</u>
2	Remington-Rand Corporation Eckert-Mauchly Computer Division 23 W. Alleghany Avenue Philadelphia 29, Pennsylvania Attn: H. F. Mitchell, Jr. Dr. John Mauchly	1	Librascope, Incorporation 1607 Flower Street Glendale 1, California Attn: H. J. Schultheis, Design Eng.
3	Remington Rand Corporation Philadelphia, Pennsylvania Attn: Dr. Grace Hopper	1	McLennan Laboratory University of Toronto Toronto, Ontario, Canada Attn: H. S. Gellman
1	Cornell Aeronautical Laboratory, Incorporated 4455 Genesee Street Buffalo 21, New York	1	Linda Hall Library Kansas City, Missouri Attn: Joseph Shipman
1	Armour Research Foundation 35 West 33rd Street Chicago 16, Illinois Attn: Librarian of Documents	1	Logistics Research Company 141 S. Pacific Avenue Redondo Beach, California Attn: R. Pearson, Librarian
1	University of California 942 Hilldale Avenue Berkeley, California Attn: D. H. Lehmer		
1	General Electric Company Electronics Park Syracuse, New York Attn: L. T. DeVore		
1	International Business Machines Corporation Engineering Laboratory Poughkeepsie, New York Attn: G. R. Trimble, Jr.		
1	Consolidated Engineering Corporation 717 N. Lake Avenue Pasadena, California Attn: Paul Brock		
1	IBM Corporation 139-143 Washington Avenue Endicott, New York		