TM: Pelstorff

Get DRF

TECHNICAL NOTE NO. 22

Research Project A-588

DIFFERENCE EQUATIONS TO APPROXIMATE ORDINARY DIFFERENTIAL EQUATIONS

By Joseph L. Hammond, Jr.

Prepared for
George C. Marshall Space Flight Center
Huntsville, Alabama

Contract No. NAS8-2473

(Development of New Methods and
Applications of Analog Computation)

24 September 1970

1970

SEAL OF THE GEORGIA INSTITUTE OF TECHNOLOGY 1885 PROGRESS AND SERVICE

Engineering Experiment Station
GEORGIA INSTITUTE OF TECHNOLOGY
Atlanta, Georgia

GEORGIA INSTITUTE OF TECHNOLOGY
School of Electrical Engineering
Atlanta, Georgia 30332

(GIT/EES Report A-588/T22)

September 1970

DIFFERENCE EQUATIONS TO APPROXIMATE ORDINARY DIFFERENTIAL EQUATIONS

by

Joseph L. Hammond, Jr.

TECHNICAL NOTE NO. 22

on

Contract No. NAS8-2473

(Development of New Methods and
Applications of Analog Computation)

For

GEORGE C. MARSHALL SPACE FLIGHT CENTER
Huntsville, Alabama

FOREWORD

The work reported in this technical note has been supported partially by Contract No. NAS8-2473 and partially by the Georgia Tech School of Electrical Engineering. The results are related to and motivated by the subject contract and also by the graduate course sequence EE 641, 642, 643 - Computer Simulation, taught by Dr. Hammond.

The author wishes to thank Dr. C. O. Alford for several discussions of certain topics covered by the technical note, and for reading a portion of the text.

TABLE OF CONTENTS

ABSTRAC

This technical note reviews and consolidates material pertaining to one-step and multistep numerical methods with the objective of developing a tractable method for evaluating the error resulting from the use of difference equations to approximate ordinary differential equations.

A tractable approximate relation is developed between the error due to finite sampling rates and parameters describing numerical methods. The results are valid for small values of the product of the sampling period, h, and natural frequencies, λ, of the analog system.

The approach used approximates the analog system by isolated modes and the digital algorithm for each mode by its order, p, and certain constants C, $A_o, \ldots, A_p$. The results show that approximating an analog system by a digital algorithm results in a shift of each root and an error corresponding to each forcing function. The root shift is given by $-Ch^p\lambda^p$, while the error caused by each forcing function, $\gamma e^{\omega t}$, is given by $-Ch^p \sum_{i=0}^{p} A_i \lambda^i \omega^{p-i}$.

The results of the study are applied to a nontrivial numerical analysis problem, a digital filter problem and to an elucidation of the problem of "stiff" equations.

DIFFERENCE EQUATIONS TO APPROXIMATE ORDINARY DIFFERENTIAL EQUATIONS

I. INTRODUCTION

The interest in digital simulation using large scale computers and the development of digital filtering techniques has led in recent years to extensive use of digital systems described by difference equations as a means of either solving differential equations or replacing analog systems described by differential equations.

The underlying mathematical problem of approximating the solution of a differential equation by the solution of a difference equation has received considerable attention over a number of years in the area of numerical analysis and more recently, for linear equations, in the area of digital filter theory. Results in the former area are typified by the detailed text of Henrici [1] and the summary articles of Benyon [2] and Giloi [3]. The latter area is discussed, for example, in the book by Gold and Rader [4]. References [1], [2], and [4] have extensive bibliographies.

Applications in general simulation and digital filters are not such as to make possible a single unique answer to the question of what difference equation to use in approximating a given differential equation. Thus, the literature contains a number of methods for obtaining difference equations which in some sense approximate differential equations. The plurality of methods leads to the necessity of defining measures of the quality of approximation and also to an analysis problem in evaluating particular methods.

The purpose of the present paper is to present a new and tractable method for studying the accuracy of stable difference equation approximations to the solution of differential equations. Background for the new method is developed through a tutorial summary giving: (1) a unified discussion of the common methods used in numerical analysis for approximating differential equations with difference equations and (2) a discussion of the important parameters and central theorems necessary in evaluating the approximations.

The work is intended to be useful in the design and analysis of digital filters and digital simulations. For such applications errors arise from two sources, namely: (1) in approximating a differential equation by a difference equation and (2) in solving the resulting difference equation. This paper gives a detailed analysis of the former source of error but does not consider the latter. Thus, the results of this paper give a complete answer for accuracy if errors (primarily due to round off) in solving the required difference equation are negligible. In general, errors due to round off and related sources must be determined through additional analysis.

2. METHODS FOR OBTAINING DIFFERENCE EQUATIONS TO APPROXIMATE DIFFERENTIAL EQUATIONS

In this section some of the more common classes of methods for obtaining difference equations to approximate differential equations will be developed using direct procedures for approximating continuous variables with discrete variables. With this motivation, a much more general class of difference equations is then presented for analysis in later sections of the paper.

Attention will be restricted to the class of differential equations which can be expressed as

$$\dot{y}(t) = f(y,t) \tag{1}$$

where y and f are vectors and f is suitably restricted so that a unique solution exists. Commonly assumed restrictions on f are given, for example, by Henrici [1]. If (1) models a physical system any input is included implicitly in f. This work is concerned with obtaining difference equations whose solutions in some sense approximate the solution, y(t), of (1).

Equation (1) can be integrated to yield

$$y(T+t) - y(T) = \int_{T}^{T+t} f[y(\tau),\tau]\, d\tau \quad , \tag{2}$$

which can be used as a basis for obtaining many of the desired difference equations. Let T and t be integer multiples of a basic step size h so that T = nh and t = kh. Using the notation $y(mh) = y_m$,(2) can be written as

$$y_{n+k} - y_n = \int_{nh}^{(n+k)h} f[y(\tau),\tau]\, d\tau \quad . \tag{3}$$

A difference equation approximating (1) results when any one of several discrete approximations are used for the right-hand side of (3).

In the special case of a linear equation with constant coefficients, (1) becomes

$$\dot{y}(t) = Ay(t) + Bu(t) \tag{4}$$

where the inputs, u(t), are now shown explicitly. The equation corresponding to (3) for the linear constant coefficient case is

$$y_{n+k} - y_n = \int_{nh}^{(n+k)h} [Ay(\tau) + Bu(\tau)]\, d\tau \quad . \tag{5}$$

In this case, however, use can be made of the closed form solution of (4) to obtain the equation

$$y_{n+k} = \Phi_k \, y_n + \int_{nh}^{(n+k)h} \Phi[(n+k)h - \tau] \, Bu(\tau) \, d\tau \tag{6}$$

where $\Phi(t)$ is the transition matrix of (4). Note in (6) that for $u(t) = 0$ the equation

$$y[n+k)h] = \Phi \, (kh) \; y(nh) \tag{7}$$

gives y for $t = (n+k)h$ exactly for any value of k. This fact makes (6) much more accurate than (5) in most applications where (6) is valid. If $u(t)$ in (6) is not zero, however, it is necessary in obtaining a difference equation from (6) to use a discrete approximation for the integral in the same manner as in the general case.

Common methods for obtaining a discrete approximation for the right-hand side of (3) are classified as one step methods and multistep methods. Direct intuitive procedures resulting in subsets of these two classes will be presented below followed by a general formulation.

2-1 One Step Methods

For one step methods, k is set equal to one and (3) becomes

$$y_{n+1} - y_n = \int_{nh}^{(n+1)h} f[y(\tau),\tau] \, d\tau \quad . \tag{7}$$

The integrand can be expanded in a Taylor series about $t = nh$ to obtain

$$f[y(\tau),\tau] = f_n + (\tau - nh) \, \dot{f}_n + \frac{(\tau - nh)^2}{2} \, \ddot{f}_n + \dots . \tag{8}$$

where

$$f_n = f[y(nh),nh] = \text{and} \quad f_n^{(i)} = \left. \frac{d^i f[y(t),t]}{dt^i} \right|_{t=nh} .$$

Introducing (8) into (7) then yields

$$y_{n+1} - y_n = hf_n + \frac{h^2}{2}\dot{f}_n + \frac{h^3}{6}\ddot{f}_n + \ldots + \frac{h^p}{p!}\overset{(p-1)}{f_n} + \ldots \tag{9}$$

Truncation of the infinite series of (9) after p terms yields

$$y_{n+1} - y_n = h[f_n + \frac{h}{2}\dot{f}_n + \ldots + \frac{h^{p-1}}{p!}\overset{(p-1)}{f_n}] , \tag{10}$$

a class of difference equations referred to as the Taylor expansion algorithms. The special case of $p = 1$ is called Euler's method. Note that p is the only parameter of this method.

In practical work, it is seldom desirable to generate derivatives of f because of the noise producing nature of the derivative operation. Thus, the algorithms of (10) with $p > 1$ have limited use.

The class of Runge-Kutta formulas make use of the fact that derivatives of f can be approximated in terms of f itself. The particular approximation identified with this name is given by

$$\int_{nh}^{(n+1)h} f[(y(\tau),\tau]\, d\tau \simeq h \sum_{i=0}^{N} \omega_i\, f[\mu_i,\eta_i] \tag{11}$$

where

$$\mu_o = nh \quad , \quad \mu_i = \mu_o + \alpha_i h, \quad \alpha_i \leq 1 \; ; \; i = 1, 2, \ldots N$$

$$\eta_o = y_n \quad , \quad \eta_i = \eta_o + h \sum_{k=0}^{i-1} \beta_{ik}\, f(\mu_k,\eta_k) \; ; \; i = 1, 2, \ldots N.$$

As can be noted in (11), the Runge-Kutta class of difference equations has the following parameters: N, N+1 values ω_i, N values of α_i and $\frac{N}{2}(N+1)$ values of β_{ik}. These parameters are evaluated by expanding

$$\int_{nh}^{(n+1)h} f[y(\tau),\tau]\, d\tau \quad \text{and} \quad h \sum_{i=0}^{N} \omega_i\, f[\mu_i,\eta_i]$$

both in Taylor series and choosing the parameters so that the coefficients of h^r are equal for $r = 1, 2, \ldots, M$. The tedious expansion operation is discussed, for example, by Henrici [1] and Ralston [5].

The results of a study of Runge-Kutta methods can be summarized as follows. A matching of coefficients can be carried only as far as $M = N+1$. Practical Runge-Kutta methods employ $N = 1, 2, 3$, and sometimes 4. For any value of N, the matching of coefficients of h^r will fix most of the parameters but at least one always remains free so that a class of algorithms exists for each value of N. For example, for $N = 1$ there is one independent degree of freedom and for $N = 3$ there are two independent degrees of freedom.

The Taylor series and Runge-Kutta methods are the more important one step methods, and attention is now given to multistep methods.

2-2 Multistep Methods

Many of the common multistep methods can be developed from (3) in the following manner. The function $f[y(t),t]$ in (3) is approximated by an interpolating polynomial and the necessary integration is then carried out to obtain a difference equation.

Given a function $z(t)$, an interpolating polynomial $P(t)$ of degree less than or equal to p can be found such that

$$P(t_i) = z(t_i) \quad ; \quad i = 0, 1, \ldots, p .$$

This polynomial can be expressed in terms of backward differences as[1]

[1]See, for example, Henrici [1].

$$P(t) = \sum_{m=0}^{p} (-1)^m \binom{\frac{t_n - t}{h}}{m} \nabla^m z_n \qquad (12)$$

where $\nabla^m z_n$ is the mth backward difference of $z(t)$ at t_n given by

$$\nabla^o z_n = z_n$$

$$\nabla^1 z_n = z_n - z_{n-1}$$

$$\vdots \qquad (13)$$

$$\nabla^j z_n = \sum_{i=0}^{j} (-1)^i \binom{j}{i} z_{n-i} \quad ,$$

and $\binom{u}{v}$ are the binomial coefficients given by

$$\binom{u}{o} = 1, \quad \binom{u}{v} = \frac{u(u-1) \; . \; . \; . \; . \; . \; (u-v+1)}{1.2 \; . \; . \; . \; . \; v} \quad . \qquad (14)$$

Note that, as in the case of (12), u in the binomial coefficient does not have to be a natural number. It is also useful to observe in (13) that $\nabla^j z_n$ depends exclusively on the set of discrete values of z given by

$\{z_n, z_{n-1}, \ldots, z_{n-j}\}$.

The polynomial P(t) can be introduced in (3) to approximate f[y(t),t] . Equation (3) then becomes

$$y_{n+k} - y_n = \int_{nh}^{(n+k)h} \sum_{m=0}^{p} (-1)^m \binom{\frac{t_{n+r} - \tau}{h}}{m} \nabla^m f_{n+r} \, d\tau \; . \qquad (15)$$

Note that since the left-hand side of (15) involves discrete values of y at the times nh and (n+k)h, backward differences of f can reasonably be taken for any time between nh and (n+k)h so that r can range between 0 and k. Since only the binomial coefficients depend on τ, (15) can be expressed as

$$y_{n+k} - y_n = h \sum_{m=0}^{p} \gamma_{m,k,r} \nabla^m f_{n+r} \tag{16}$$

where

$$\gamma_{m,k,r} = \frac{(-1)^m}{h} \int_{nh}^{(n+k)h} \binom{\frac{t_{n+r}-\tau}{h}}{m} d\tau \quad . \tag{17}$$

It can be shown that since $\nabla^m f_{n+r}$ depends exclusively on $f_{n+r}, \ldots, f_{n+r-m}$, (16) can also be expressed as

$$y_{n+k} - y_n = h \sum_{m=0}^{p} \beta_{m,k,r,p} f_{n+r-m} \quad . \tag{18}$$

Either (16) or (18) defines a general class of difference equations depending on the parameters p, k, and r so that particular values of p, k, and r result in fixed values of the $\gamma_{m,k,r}$ and $\beta_{m,k,r,p}$. The $\gamma_{m,k,r}$ are given explicitly by (17) and similar relations can be derived for the $\beta_{m,k,r,p}$.

The general class of difference equations given by (18) has two major subdivisions determined by whether or not r is chosen equal to k so that values of $f_{n+k} = f[y_{n+k}, t_{n+k}]$ are required in the algorithm. If $r = k$, (16), or (18) then has y_{n+k} on the left-hand side and also on the right-hand side in f_{n+k}. Such formulas are called implicit or closed formulas since in general they cannot be solved directly for y_{n+k}. If $r \neq k$, the resulting formulas can be solved directly and are called open or explicit formulas.

Closed formulas can be solved iteratively by the following procedure: (a) estimate $y_{n+k}^{(0)}$ for use in f_{n+k}, (b) solve (16) or (18) for $y_{n+k}^{(1)}$, (c) use $y_{n+k}^{(1)}$ in f_{n+k} to obtain $y_{n+k}^{(2)}$ from (16) or (18), and (d) repeating the process. Typically, the first estimation is accomplished with an open formula called a predictor. The closed formula is then referred to as a corrector and the complete calculation as a predictor-corrector algorithm.

Examination of (18) shows that to compute y_{n+k} requires y_n and values of f_i from $i = n+r$ to $i = n+r-p$. Since p is in general greater than r, values of f prior to t_n are typically required. This causes a problem in starting multistep methods since (18) cannot be applied unless all required past values of f_i are given. The startimg problem can be solved by computing the required values with some type of one step method since these methods are self-starting.

2-3 A General Difference Formula

A general difference formula which includes all of the formulas above as special cases is given by

$$\sum_{i=0}^{k} \alpha_i \, y_{n+i} = h \, F\{h, n, y_{n-p}, \ldots, y_{n+k}; f\} \qquad (19)$$

where k and p are fixed and $n = p, p+1, \ldots$. Equation (19) has parameters k, p, the α_i and the function F.

For the common multistep algorithms F is given by

$$F = \sum_{i=0}^{k} \beta_i \, f_{n+i} \qquad (20)$$

so that (19) becomes

$$\sum_{i=0}^{k} \alpha_i \, y_{n+i} = h \sum_{i=0}^{k} \beta_i \, f_{n+i} \quad . \tag{21}$$

Note that (21) has parameters k, the α_i, and the β_i. Table I gives the parameter values for a number of common open multistep methods. For such methods β_k is always zero. Table II gives the parameters for common closed multistep methods. Both tables give truncation error for the algorithms, a parameter which will be defined and discussed below.

For one step methods, $\alpha_o = -1$, $\alpha_1 = 1$ and all other α_k in (19) are zero. For the Runge-Kutta algorithms, which are the most practical one step methods, F is the function defined in (11). Thus, (19) reduces to

$$y_{n+1} - y_n = h \sum_{i=0}^{N} \omega_i f[\mu_i, \eta_i] \tag{22}$$

where

$$\mu_o = nh \quad ; \quad \mu_i = \mu_o + \theta_i h \quad , \quad i = 1, 2, \ldots, N$$
$$\theta_i \leq 1$$

$$\eta_o = y_n \quad ; \quad \eta_i = \eta_o + h \sum_{k=0}^{i-1} \beta_{ik} f(\mu_k, \eta_k) \; .$$

Typical values for the parameters N, ω_i, θ_i, and β_{ik} of the Runge-Kutta algorithm are given in Table III.

The formulas preceeding (19) were derived so as to insure that their solutions approximate that of (1) in some sense. Equation (19), on the other hand, is much more general and in order for it to have any utility in approximating the solutions of (1), restrictions on its parameters must be developed. The next section discusses such restrictions through defining measures of quality for the required approximations.

TABLE I: OPEN MULTISTEP METHODS

	k	α_o	α_1	α_2	α_3	α_4	β_o	β_1	β_2	β_3	Truncation Error
Euler	1	-1	1				1				$\frac{1}{2}h^2 y^{(2)}$
Nystrom	2	-1		1		weakly stable		2			$\frac{1}{3}h^3 y^{(3)}$
A-B 2 Step	2		-1	1			$\frac{1}{-2}$	$\frac{3}{2}$			$\frac{5}{12}h^3 y^{(3)}$
A-B 3 Step	3			-1	1		$\frac{5}{12}$	$\frac{16}{12}$	$\frac{23}{12}$		$\frac{3}{8}h^4 y^{(4)}$
A-B 4 Step	4				-1	1	$-\frac{9}{24}$	$\frac{37}{24}$	$-\frac{59}{24}$	$\frac{55}{24}$	$\frac{251}{720}h^5 y^{(5)}$
Milne-Hamming	4	-1				1		$\frac{8}{3}$	$-\frac{4}{3}$	$\frac{8}{3}$	$\frac{14}{45}h^5 y^{(5)}$
Hamming	5	1	-1			-1		$-\frac{8}{3}$	4	-4	$-\frac{11}{72}h^5 y^{(5)}$

TABLE II: CLOSED MULTISTEP METHODS

	k	α_o	α_1	α_2	α_3	α_4	α_5	β_o	β_1	β_2	β_3	β_4	β_5	Truncation Error
Modified Euler	1	-1	1					$\frac{1}{2}$	$\frac{1}{2}$					$-\frac{1}{12}h^3 y^{(3)}$
A-M 2 Step	2		-1	1				$-\frac{1}{12}$	$\frac{8}{12}$	$\frac{5}{12}$				$-\frac{1}{24}h^4 y^{(4)}$
A-M 3 Step	3			-1	1			$\frac{1}{24}$	$-\frac{5}{24}$	$\frac{19}{24}$	$\frac{9}{24}$			$-\frac{19}{720}h^5 y^{(5)}$
Milne	3		-1		1				$\frac{1}{3}$	$\frac{4}{3}$	$\frac{1}{3}$			$-\frac{1}{90}h^5 y^{(5)}$
Hamming	3	$\frac{1}{8}$		$-\frac{9}{8}$	1				$-\frac{3}{8}$	$\frac{6}{8}$	$\frac{3}{8}$			$-\frac{1}{40}h^5 y^{(5)}$
A-M 4 Step	4				-1	1		$-\frac{19}{720}$	$\frac{106}{720}$	$-\frac{264}{720}$	$\frac{646}{720}$	$\frac{251}{720}$		$-\frac{3}{160}h^5 y^{(5)}$
Hamming	4	$\frac{9}{121}$	$\frac{14}{121}$		$-\frac{126}{121}$	1			$\frac{24}{121}$	$\frac{-54}{121}$	$\frac{108}{121}$	$\frac{42}{121}$		$\frac{-21}{1210}h^6 y^{(6)}$

TABLE III: RUNGE-KUTTA ONE STEP METHODS

N	ω_0	ω_1	ω_2	ω_3	θ_1	θ_2	θ_3	β_{10}	β_{20}	β_{21}	β_{30}	β_{31}	β_{32}
3	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	1
	Principle Error Function N = 3:	$\varphi(t,y) = \frac{1}{2880} f^{iv} - \frac{1}{576} f_y \dddot{f} + \frac{1}{288} (f_y^2 - f_{ty} - f_{yy} f) \ddot{f} + \frac{1}{192} (2f_{ty} f_y + 3f_{yy} f_y f - 2f_y^3 + f_{yy} f_t) \dot{f}$											
3	.174	-.551	1.205	.171	.4	.455	1	.4	.296	.158	.218	-3.050	3.832
					8 place numbers truncated to 3 places								
2	$\frac{2}{9}$	$\frac{1}{3}$	$\frac{4}{9}$		$\frac{1}{2}$	$\frac{3}{4}$		$\frac{1}{2}$	0	$\frac{3}{4}$			
2	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$		$\frac{1}{2}$	1		$\frac{1}{2}$	-1	2			
1 $\alpha=1$	0	1			$\frac{1}{2}$			$\frac{1}{2}$					
1 $\alpha=\frac{3}{4}$	$\frac{1}{4}$	$\frac{3}{4}$			$\frac{2}{3}$			$\frac{2}{3}$					
1 $\alpha=\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$			1			1					
	Principle Error Function N = 1:	$\varphi(t,y) = (\frac{1}{8\alpha} - \frac{1}{6}) \ddot{f} - \frac{1}{8\alpha} f_y \dot{f}$											

3. MEASURES OF QUALITY OF APPROXIMATION

Attention is given in this section to the problem of restricting the parameters of (19) in such a way as to insure that its solution, y_n, for n = p, p+1, ..., approximates in some sense the solution, y(t), of (1) at t = nh for the same values of n. To avoid confusion, in the remainder of the paper the solution of (1) will be denoted as Y(t).

Two a priori restrictions are placed on F, namely: (1) F is restricted so that small changes in its parameters result in small changes in F, a condition similar to that required of f, and (2) F is restricted so that $F\{h,n, y_{n-p},\ldots, y_{n+k}; 0\} = 0$.

Equation (19) is an algorithm for determining y_{n+1} given y_n and other required data. Thus, much of the evaluation of (19) can be based on a consideration of the typical step in the iterative calculation of y_n. To consider the effect of a single step in the calculation, y_n and other required data are assumed to be exact at the beginning of the typical step and (19) is used to calculate y_{n+1}. The quantity y_{n+1}. Assuming the errors occurring in only one step, is denoted[2] $\hat{y}_{n+1}$.

A quantity of central importance in studying (19) is local truncation error, T_{n+1}, defined by

$$T_{n+1} = \hat{y}_{n+1} - Y[(n+1)h] \tag{23}$$

[2]Note that in computing y_{n+1}, true values are assumed for all y_n, i=0,1,...., k-1 and also for any y_{n+k} required as an argument of F.

Equations (19) and (23) can be used to express T_{n+k} as

$$T_{n+k} = hF\{h,n,Y[(n-p)h], \ldots, Y[(n+k)h;\ f]\} - \sum_{i=0}^{k} \alpha_i\ Y[(n+i)h] \tag{24}$$

where the subscript n+k is used because the general algorithm evaluates y_{n+k} using past data.

For a given algorithm T_{n+k} is conveniently computed by expanding $\sum_{i=0}^{k} \alpha_i\ Y(t)$ and $hF\{h,n,Y[(n-p)h], \ldots, Y[(n+k)h];f\}$ in Taylor series about a convenient point such as $t = nh$ and subtracting the series term by term to obtain

$$T_{n+k} = \phi_o + h\phi_1 + \frac{h^2}{2}\phi_2 + \frac{h^3}{6}\ \phi_3 + \ldots . \tag{25}$$

where the ϕ_i depend on both F and Y(t). In any particular case the first p+1 functions ϕ_i, i = 0, 1, 2, ..., p will be zero and the method is termed to be of <u>order</u> <u>p</u>.

It will be useful below to note that for many methods T_{n+k} can be expressed as

$$T_{n+k} = h^{p+1}\phi\ (\mu, Y) + O(h^{p+2}) \tag{26}$$

where μ is a point "near" (n+k-1)h and $O(h^{p+2})$ indicates a group of terms, which approach zero as h approaches zero at least as rapidly as h^{p+2}.

The function $\phi(u,Y)$ is called the princple error function by Henrici [1] in discussions of one step methods. For many such methods $\phi\ (u,Y)$ for a scalar problem equation is given by

$$\phi(u,Y) = a_{p+1}(u,Y) \overset{(p+1)}{Y}(u) + a_p(u,Y) \overset{p}{Y}(u) + \dots + a_2(u,Y) \overset{''}{Y}(u) \qquad (27)$$

where p is the order of the method and the a_i are in general functions of u and Y which depended on the given problem equation. The principle error functions for several Runge-Kutta Algorithms applied to scalar problem equations are given in Table III.

For most multistep methods

$$\phi(u,Y) = C \overset{(p+1)}{Y} (u) \qquad (28)$$

Thus, for such methods $\phi(u,Y)$ can be obtained by dividing the truncation error (as given for several methods in Tables I and II) by h^{p+1}.

If a method is of at least first order, it is said to be <u>consistent</u>. Note that for a consistent method the local truncation error is at least $O(h^2)$.

The cumulative effect of truncation error in all of the steps from n = p to n = j is a total <u>pointwise error</u>, e_j, at t = jh. This quantity, given by

$$e_j = y_j - Y(jh) , \qquad (29)$$

differs from T_j in that e_j includes the total effect of errors in many steps, rather than in a single step.

Using pointwise error it is possible to define convergence of y_n to Y(nh) for a given method as follows.

If for any f(y,t)

$$\max_n |e_n| \to 0 \text{ as } h \to 0 ,$$

then the method is convergent.

Using (19) to express y_n and (23) and (24) to express $Y(nh)$, a difference equation can be obtained for e_n from (29) as

$$\sum_{j=0}^{k} \alpha e_{n+j} = h[F\{h,n,y_{n-p},\ldots,y_{n+k};f\} - F\{h,n,Y[(n-p)h,\ldots, Y[(n+k)h];f\} + T_{n+k} . \qquad (30)$$

Several types of stability can be discussed for algorithms in the class (19). A type which is tractable for the general case can be termed <u>limiting stability</u> in the sense that it is applied for either h or f assumed to be identically zero. With this assumption, (19) becomes

$$\sum_{j=0}^{k} \alpha_j y_{n-j} = 0 . \qquad (31)$$

The solution to this difference equations has the form $\sum_{i=1}^{k} \alpha_i \beta_i^n$ where the β_i are the roots of the polynomial

$$P(\beta) = \sum_{j=0}^{k} \alpha_j \beta^j = 0 . \qquad (32)$$

Thus, (31) is stable if and only if the β_i satisfy the conditions

$$|\beta_i| \leq 1 \qquad (33)$$

with the additional requirement that if $|\beta_j| = 1$ then β_j must be a simple root. This condition is referred to as the <u>root condition</u>. The root condition, which guarantees limiting stability is necessary but not sufficient for more general types of stability.

The following theorem which relates the root condition to convergence of y_n to $Y(nh)$ can be proved:[3] If (19) is consistent, then it is convergent if and only if the root condition is satisfied. Thus, if the root condition is satisfied and (19) is consistent, some small enough h can always be found so that y_n is a good approximation to $Y(nh)$. The stability and pointwise error of (19) for fixed nonzero values of h will be investigated below but much more restrictive conditions will have to be assumed.

4. POINTWISE ERROR

The purpose of this section is to derive a tractable differential equation for pointwise error starting from the difference equation (30) assuming that the method is stable and that T_{n+k} can be approximated by $h^{p+1}\phi(u,Y)$, (see (26)). Consider the function $F\{h,n,Y[(n-p)h], \ldots. Y[(n+k)h]; f\}$. Using (29), the true value $Y(nh)$ can be expressed as

$$Y(nh) = y_n - e_n, \tag{34}$$

and F can then be expanded in a Taylor series about the points y_n. Assuming that e_n is small it is reasonable to truncate the Taylor series to two terms and the result is

$$F\{h,n,Y[(n-p)h], \ldots, Y[(n+k)h]; f\} = \tag{35}$$

$$F\{h,n,y_{n-p}, \ldots, y_{n+k};f\} - \left[\frac{\partial F}{\partial y_{n-p}} e_{n-p} + \ldots + \frac{\partial F}{\partial y_{n+k}} e_{n+k}\right] .$$

Using (35) in (30) then gives the difference equation

$$\sum_{j=0}^{k} \alpha_j e_{n+j} = h \left[\frac{\partial F}{\partial y_{n-p}} e_{n-p} + \frac{\partial F}{\partial y_{n+k}} e_{n+k}\right] + T_{n+k}, \tag{36}$$

[3] See for example Isaacson and Keller [6]

which applies for a general F.

The approach in approximating the solution e_n of (36) by the solution of a differential equation is to find, to a given order of accuracy, a differential equation which results in the difference equation (36) through application of the algorithm of the method being investigated. The procedure will be carried out for the two special cases of one-step and multistep methods.

For one-step methods (19) becomes

$$y_{n+1} - y_n = hF\{h,n,y_n;f\} \tag{37}$$

and the equation analogous to (36) for the specific F of (37) becomes

$$e_{n+1} - e_n = h\left[\frac{\partial F}{\partial y_n} e_n + h^p \phi(\mu,Y)\right] \tag{38}$$

with $nh \leq \mu \leq (n+1)h$.

Now by defining

$$\bar{e}_n = e_n h^{-p} \quad , \tag{39}$$

(39) becomes

$$\bar{e}_{n+1} - \bar{e}_n = h\left[\frac{\partial F}{\partial y_n} e_n + \phi(\mu,Y)\right], \tag{40}$$

But Euler's method applied to $\dot{z} = g(z;t)$ yields the difference equation $z_{n+1} - z_n = h\, g(z_n,nh)$. Thus, it can be asserted by choosing

$$g(z_n,nh) = \frac{\partial F}{\partial y_n} \bar{e}_n + \phi(\mu,Y) \tag{41}$$

that the differential equation

$$\dot{\bar{e}}(t) = \frac{\partial f}{\partial Y} \bar{e} + \phi (\mu, Y) \tag{42}$$

has a solution $\bar{e}(t)$ which approximates $\bar{e}_n$ to within an error $O(h)$. In obtaining (42) use is made of the fact that F approximates f within errors which are $O(h)$. Equation (42) written in terms of $e(t)$ rather $\bar{e}(t)$ becomes

$$\dot{e}(t) = \frac{\partial f}{\partial Y} e(t) + h^p \phi(\mu, Y) \tag{43}$$

and the errors are now $O(h^{P+1})$.

A similar line of reasoning is used to give the result for multistep methods. For such methods

$$F = \sum_{j=0}^{k} \beta_j f_{n+j} \tag{44}$$

and

$$\frac{\partial F}{\partial y_{n+j}} = \beta_j \frac{\partial f_{n+j}}{\partial y_{n+j}} .$$

Thus, the equation corresponding to (36) becomes

$$\sum_{j=0}^{h} \alpha_j e_{n+j} = h \left[\sum_{j=0}^{k} \beta_j \frac{\partial f_{n+j}}{\partial y_{n+j}} e_{n+j} + h^p \phi(\mu, Y) \right] . \tag{45}$$

A multistep algorithm using (44) can be expressed as

$$z_{n+1} - z_n = h \sum_{j=0}^{k} \beta_j \dot{z}_{n+j} . \tag{46}$$

where $\dot{z} = f(z,t)$ has been assumed. Equation (46) shows that within an error which is $O(h)$ the following equation holds

$$\frac{z_{n+1} - z_n}{h} \simeq \dot{z}_n = \sum_{j=0}^{k} \beta_j \dot{z}_{n+j} \,. \tag{47}$$

If $\phi(\mu,Y)$ is identified with $\dot{z}(t)$ in (47), it follows that within an error which is $O(h)$, $\phi(\mu,Y)$ can be expressed as

$$\phi(\mu,Y) = \sum_{j=0}^{k} \beta_j \phi(\mu + jh, Y). \tag{48}$$

Hence (49) can be written as

$$\sum_{j=0}^{h} \alpha_j \bar{e}_{n+j} = h \sum_{j=0}^{k} \beta_j \left[\frac{\partial f_{n+j}}{\partial y_{n+j}} \bar{e}_{n+j} + \phi(\mu + jh, Y) \right] \tag{49}$$

where $\bar{e}_n$ is defined as in (39). But the differential equation

$$\dot{\bar{e}}(t) = \frac{\partial f}{\partial Y} \bar{e}(t) + \phi(\mu,Y) \tag{50}$$

results in the difference equation (49) using the multistep methods. Thus, when (50) is transformed to use $e(t)$ instead of $\bar{e}(t)$, an equation identical to (43) results.

If a method with pointwise error e_n has an order p, then $e(nh)$ given by (43) satisfies

$$e_n = e(nh) + O(h^{p+1}) \,, \tag{51}$$

and the error in approximating e_n by $e(nh)$ is of an order in h one higher than the order of the method.

Since (51) approximates pointwise error at all time, t, it is reasonable to consider a variable $y^*(t)$, defined so that $y^*(nh) = y_n$, which represents the solution of the approximating difference equation at all times. In analogy to (32), $Y(t)$, $y^*(t)$, and $e(t)$ are related by

$$Y(t) = y^*(t) - e(t) \; . \tag{52}$$

It is useful to note that the function $\phi(\mu,Y)$ of (43) can be expanded in a Taylor series about a point (t,y^*) to obtain

$$\phi(\mu,Y) = \phi(t,y^*) - e \frac{\partial\phi}{\partial Y} + (\mu - t) \frac{\partial\phi}{\partial\mu} + \ldots \; . \tag{53}$$

If $\frac{\partial\phi}{\partial Y}$ and $\frac{\partial\phi}{\partial\mu}$ are bounded, as is usually the case, then for e at least $O(h)$ and $\mu - t \leq h$

$$\phi(\mu,Y) = \phi(t,y^*) + O(h) \; . \tag{54}$$

Thus, to the accuracy being used, (43) can be expressed as

$$\dot{e}(t) = \frac{\partial f}{\partial Y} e(t) + h^p \phi(t,y^*) \; . \tag{55}$$

Using (55), (52) and the equation resulting from differentiating both sides of (52), it is a simple matter to obtain the equation

$$\dot{y}^*(t) = \frac{\partial f}{\partial Y} y^* + \dot{Y}(t) - \frac{\partial f}{\partial Y} Y(t) + h^p \phi(t,y^*) \tag{56}$$

the solution of which closely approximates the solution of a difference equation obtained by the one step or multistep methods. For the test problem to be discussed in the next section, (56) gives a tractable expression for evaluating accuracy.

5. EVALUATION OF APPROXIMATIONS WITH TEST PROBLEMS

The discussion up to this point pertains to the general class of differential equations given by (1). To proceed further with the development of a tractable procedure for evaluating difference equation approximations to differential equations, it is necessary to make simplifying assumptions. The approach which has been chosen is that of using a tractable linear constant coefficient test problem to replace the given differential equation in the evaluation of difference equation approximations. Such an approach, which is not uncommon in engineering analysis, can be justified in several ways. For example, the linear constant coefficient equation can be chosen to represent the incremental behavior of a general system described by (1) about some average tracjectory.

Lomax [7] gives an extended discussion justifying the linear test problem approach. In addition, he shows that errors associated with each natural frequency of a coupled system of linear constant coefficient equations and with each forcing function can be treated separately so that it is sufficient for the test problem to contain only a single (possibly complex) eigenvalue and a single forcing function. Thus, an adequate test problem is given by

$$\dot{y}(t) = \lambda y(t) + \gamma e^{\omega t} \qquad (57)$$

where $y(t)$ is a scalar and λ and ω can be either real or complex.

When using (57) to study a difference equation approximation for a specified problem, λ and ω are chosen to correspond to extreme or "worst case" natural frequencies or forcing function components in a manner which will be illustrated below.

Two considerations, namely stability and accuracy are necessary in evaluating an approximation and these will now be treated separately using the test problem.

5-1 Stability

Limiting stability as defined above can be studied without recourse to a test problem and it can be noted that all one step methods are stable in this sense. In practical work, however, the question of stability for non zero values of h must be faced and one approach is to study such a situation using the test problem of (57). When studying stability a forcing function is not required and thus γ can be set equal to zero.

Generally speaking, stability can be studied by formulating the difference equation resulting from a particular algorithm applied to the test problem of (57) with $\gamma = 0$. Due to the nature of the test problem, the difference equation thus obtained is linear with constant coefficients. Since the test problem has a single root, one step methods produce a difference equation with a single root. Multistep methods on the other hand produce extraneous roots. In any case, the solution of the difference equation is the sum of terms of the form $(r_i)^n$, where the r_i are the roots of the difference equation. It follows that the condition $|r_i| \leq 1$, (with all roots for which $r_i = 1$ simple), insures stability. Note that the r_i depend on λ and h so that boundaries on $h\lambda$, which insure that $|r_i| \leq 1$, can be obtained. One way to present these boundaries is to plot the real versus the imaginary part of $h\lambda$ for the condition that $|r_{max}| = 1$. Curves of stability boundaries for a number of methods are given by Benyon.[4]

5-2 Accuracy

The test problem given by (57) can also be used to investigate the accuracy of a given algorithm. After the algorithm has been shown to be stable, the differential equation (56) applies and gives a good approximation to the solution that is obtained by solving the difference equations corresponding to the algorithm being investigated. Thus, to study accuracy, (56) is formulated for the test problem by evaluating $\frac{\partial f}{\partial Y}$ and $\phi(t,y^*)$ for this equation and algorithm being investigated. The quantity $\frac{\partial f}{\partial Y}$ is obtained easily as

$$\frac{\partial f}{\partial Y} = \lambda. \tag{58}$$

The principle error function $\phi(t,y^*)$ is obtained as follows. For one step methods applied to the test problem, the $a_i(\mu,Y)$ in (27) are found to be independent of μ and Y and $\phi(t,Y)$ can be expressed as

$$\phi(t,Y) = -[C_1 \lambda^{p+1} Y(t) + C_2 \gamma e^{\omega t} \sum_{i=0}^{p} a_i \lambda^i \omega^{p-i}] \tag{59}$$

where for one step methods $C_1 = C_2 = C$

For example in considering the Runge-Kutta methods for N = 1 and 3 as given in Table III, the only nonzero partial derivative of $f = \lambda Y + \gamma e^{\omega t}$ is $\frac{\partial f}{\partial Y}$ which is equal to λ. Thus, for N = 3 the a_i of (27) become $a_5 = \frac{1}{2880}$, $a_4 = \frac{-\lambda}{576}$, $a_3 = \frac{\lambda^2}{288}$ and $a_2 = \frac{-2\lambda^3}{192}$. For N = 1

---

[4]Op. cit.[2] p. 227.

the values are $a_3 = \frac{1}{8\alpha} - \frac{1}{6}$, $a_2 = \frac{-\lambda}{8\alpha}$. In each case

$$C_2 = \sum_{i=2}^{p+1} a_i$$

$$A_i = \frac{1}{C_2} \sum_{p+1-i}^{p+1} a_i \; ; \; i = 0, 1, \ldots p-1 \qquad (60)$$

$$A_p = 1$$

For the multistep methods, $\varphi(t,Y)$ as given by (28), becomes equal to the expression of (59) with $C_1 = C_2 = C$ and the $A_i = 1$, $i = 0, 1, \ldots p$. There are methods with digital filter equations being the prime example, for which C_1 and C_2 are not equal. In fact, for the digital filter case $C_1 = 0$. For all the algorithms investigated, however, $\varphi(t,Y)$ can be evaluated and expressed in the form of (59).

When (58) and (59), with y^* replacing Y, are introduced in (56) the equation becomes

$$\dot{y}^*(t) = [\lambda - C_1 h^p \lambda^{p+1}]\, y^*(t) + \gamma e^{\omega t} - C_2 h^p \gamma e^{\omega t} \sum_{i=0}^{p} A_i \lambda^i \; ^{p-i} \qquad (61)$$

Note that the solution of (61) approximates the solution of a difference equation for the test problem obtained using an algorithm described by an order p, constants C_1 and C_2 and $A_i = 0, 1, \ldots, p$.

If (61) is compared to (57), it can be seen that using a difference equation to approximate a differential equation has produced two types of error, namely:

(1) The root λ has been shifted to the new position $\lambda - C_1 h^p \lambda^{p+1}$ and

(2) The forcing function has been perturbed from $\gamma e^{\omega t}$ to

$$\gamma e^{\omega t} [1 - C_2 h^p \sum_{i=0}^{p} A_i \lambda^i \ \ p^{-i}].$$

These effects will be considered separately by obtaining the "root shift" with no forcing function and the "forced error" assuming no root shift.

5-2.1 Root Shift: If fractional root shift, E_r, is defined in an obvious fashion there results

$$E_r = \frac{(\lambda - C_1 h^p \lambda^{p+1}) - \lambda}{\lambda} = -C_1 h^p \lambda^p \tag{62}$$

where it should be recalled that both E_r and λ can be complex. This equation can be expressed in the following equivalent and useful ways:

$$h \lambda = \left(\frac{|E_r|}{|C_1|} \right)^{1/p} \tag{63}$$

$$\frac{1}{h|\lambda|} = \left(\frac{|C_1|}{|E_r|} \right)^{1/p} \tag{64}$$

$$\ln|E_r| = p \ln h|\lambda| + \ln|C_1|. \tag{65}$$

Note that the units of $\frac{1}{h|\lambda|}$ are steps/radian.

Equation (68) shows that $|E_r|$ increases with h. Therefore, if $|E_r|$ is regarded as maximum permissible root shift, $h|\lambda|$ must satisfy

$$h|\lambda| \leq \left(\left| \frac{E_r}{C_1} \right| \right)^{1/p} \tag{66}$$

and similar expressions can be obtained to correspond to (64) and (65).

Equation (65) plots on log-log paper as a straight line with slope p intercepting $h|\lambda| = 1$ at a point equal to $|C_1|$ The normalized quantity $|E_r|/C_1$ is plotted versus $h|\lambda|$ and $2\pi / h|\lambda|$ for various values of p in Figure 1.

<u>5-2.2 Forced Error</u>: A fractional forced error, E_f, is defined as

$$E_f = \frac{y^*(t) - Y(t)}{Y(t)} \quad . \tag{67}$$

Since (57), and (61), whose solutions are $Y(t)$ and $y^*(t)$, are linear, it follows that

$$y^*(t) = [1 - C_2 h^p \sum_{i=0}^{p} A_i \lambda^i \omega^{p-i}] \; Y(t). \tag{68}$$

(Recall that root shift is being ignored in considering forced error so that the homogeneous parts of (57) and (61) become identical.) Substituting (68) into (67) yields

$$E_f = -C_2 h^p \sum_{i=0}^{p} A_i \, \lambda^i \omega^{p-i}. \tag{69}$$

Note that E_f is independent of time since $y^*(t)$ is a constant times $Y(t)$.

It is reasonable to consider three cases which result in the following tractable approximations:

<u>Case 1: $|\lambda| << |\omega|$</u>

$$E_f \simeq - A_o \; C_2 h^p \omega^p \tag{70}$$

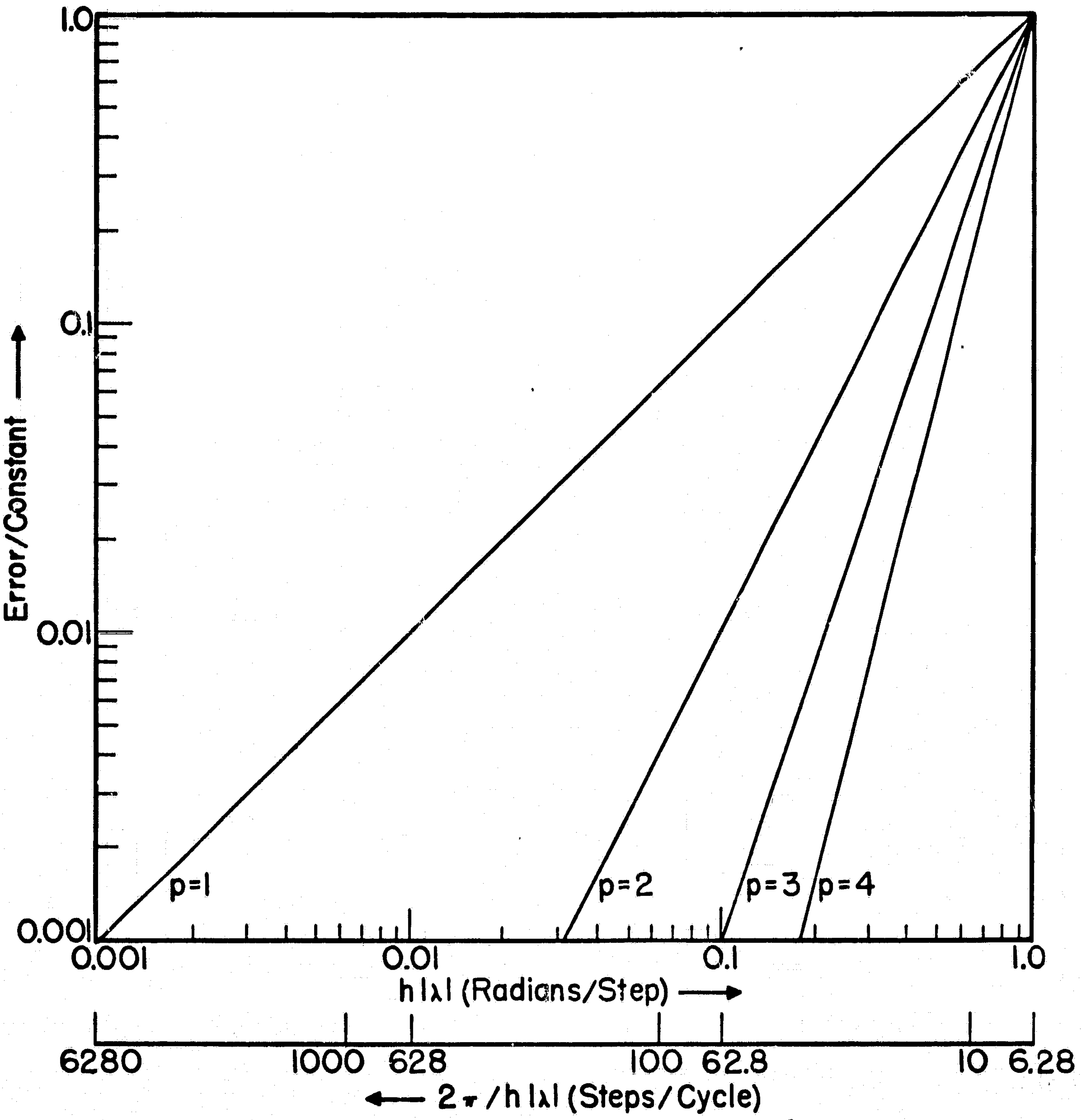

Figure 1. Normalized error versus $h|\lambda|$ and $2\pi/h|\lambda|$ for various values of p.

Case 2: $|\lambda| >> |\omega|$

$$E_f \simeq - A_p C_2 h^p \lambda^p \tag{71}$$

Case 3: $|\lambda| \simeq |\omega|$

$$E_f \simeq - C_2 h^p \lambda^p \sum_{i=0}^{p} A_i \tag{72}$$

In all three cases, E_f has the same form as E_r given by (62). In cases 2 and 3 E_f is proportional to $h^p \lambda^p$ just as is E_r but with different constants in the two cases.

In case 1, E_f is proportional to $h^p \omega^p$ which has the same form as E_r with ω replacing λ. Thus the expressions and curves for E_r can also be used for E_f with appropriate changes in the constants and with λ replaced by ω for case 1.

Applications of the accuracy equations in studying difference equation approximations will be given in the next section.

6. EXAMPLES

6-1 General Example:

To illustrate the techniques presented above, consider the problem of choosing a numerical algorithm for digital simulation of an aerospace vehicle using the simplified model discussed by Ryan et al [8]. This model can be represented as a forth order linear differential equation with a time varying coefficient which is proportional to vehicle attitude. The differential equation has a random forcing function (caused by wind perturbation) whose power spectral band width is also proportional to attitude.

In order to study various algorithms, the problem can be regarded as quasi-stationary with roots whose position depends on the time varying altitude.

The ranges of these roots and of the power spectral band width of the forcing function are tabulated in Table IV.

Table IV. Frequencies associated with the Space Vehicle Problem.

	Range of Root Location or Maximum Bandwidth
Real Roots	$\lambda_1 = 0$; $-0.05 \leq \lambda_2 \leq 0$
Complex Roots	$\lambda_{3,4} = \omega_c \angle \pm 127^o$; $0.5 \leq \omega_c \leq 2.0$
Forcing Function Power Spectral Band width	$0 \leq \omega \leq B$; $0 \leq B \leq 6.0$

In considering various algorithms, it is clear that stability is required for successful approximation. For purposes of illustration let it also be required that fractional root shift, E_r, and fractional forced error E_f be limited in magnitude to 1%.

Use is now made of the normalized curves of Figure 1 and Benyon's [2] results on stability to obtain limits on the step size for stable and accurate operation using representative algorithms. Limits on $h|\lambda|$ for stability are tabulated for a test problem with a real root and a root location along the 127 degree line, in the complex plane for five representative methods in Table V. The table also gives the limit on $h|\lambda|$ for $|E_r| \leq 1\%$ and the value of $h\,\Omega$ for $|E_f| = 1\%$ under the three assumptions $|\lambda| << \Omega$, $|\lambda| \simeq \Omega$ and $|\lambda| >> \Omega$ where Ω is the single forcing frequency and λ is the single natural frequency of a test problem.

It can be noted from Table V that for the one-step methods, (Euler, RK-2, RK-4), the limits on $h|\lambda|$ (or $h\Omega$) for stability tend to be significantly

Table V. Limits on $h|\lambda|$ or $h\,\Omega$ for various conditions.

		Stability			$\|E_f\| \le 1\%$		
Algorithm		Real Roots	$\lambda = \lambda_c \angle 127^\circ$	$\|E_r\| \le 1\%$	$\|\lambda\| \ll \Omega$	$\|\lambda\| \simeq \Omega$	$\|\lambda\| \gg \Omega$
Euler		2.0	1	.02	.023	.01	.023
AB-4		.3	.3	.4	.41	.3	.41
AM-4		1.8	.9	.9	.9	.6	.9
RK-2	$\alpha = 1/2$	2.0	2.1	.25	.3	.15	.2
	$\alpha = 1$				.4	.15	
	$\alpha = 3/4$				very large	.15	
RK-4		2.7	2.6	1.05	2.3	.9	1.05

larger than the limits imposed by E_r or E_f. On the other hand for the multistep methods, (AB-4, AM-4), the limits imposed by stability tend to be approximately the same as those imposed by E_r and E_f.

For the space vehicle problem, the complex roots will be the limiting factor due to the relative sizes of the four roots.

The forcing function for the problem covers a range of frequencies such that the results for the single test frequence, Ω, approximately equal to the magnitude of the complex root would seem to be the best guide to accuracy of the forced problem. Thus setting $\Omega = B$ and $|\lambda| = \omega_c$ yields the numbers given in Table VIwhich lists bounds on h caused by ω_c and by the forcing function. The table also gives final bounds on h assuming the two conditions $\omega_c = 2$, $B = 6$, and $\omega_c = B = 2$. The table shows that the RK-4 algorithm has the largest limiting step size for both of the conditions cited.

6-2 Digital Filter

A recursive digital filter is an algorithm, usually implemented in real time, which can be expressed as a difference equation. One important digital filter design technique involves tailoring the difference equation defining the digital filter so that its response closely approximates the solution of a differential equation defining the response of an analog filter, (see for example Rader and Gold [9]). Using this point of view, the general results of this paper apply and will give, for example, the accuracy with which the digital filter response approximates the analog filter response in the region of excitation for which the product of step size and frequency is small. The latter point should be noted since digital filter responses are

Table VI. Limits on h for Various Conditions.

Algorithm		Limit resulting from ω_c		Limit resulting from forcing function	Final Limit	
		limit on h	Cause		$\omega_c = 2$, $B = 6$	$\omega_c = 2$, $B = 2$
Euler		$\frac{.02}{\omega_c}$	E_r	$\frac{.01}{B}$	.002	.005
AB-4		$\frac{.3}{\omega_c}$	Stability	$\frac{.3}{B}$	.05	.15
AM-4		$\frac{.9}{\omega_c}$	Stability or E_r	$\frac{.6}{B}$	.1	.3
RK-2	$\alpha = 1/2$	$\frac{.2}{\omega_c}$	E_r	$\frac{.15}{B}$	.02	.07
	$\alpha = 1$	$\frac{.2}{\omega_c}$	E_r	$\frac{.15}{B}$	.02	.07
	$\alpha = 3/4$	$\frac{.2}{\omega_c}$	E_r	$\frac{.15}{B}$	.02	.07
RK-4		$\frac{1.05}{\omega_c}$	E_r	$\frac{.9}{B}$	.15	.45

sometimes examined in the frequency region where this restriction does not apply.

The considerations involved in obtaining accuracy for digital filters can be adequately illustrated by considering the realization of a single complex pole pair. An impulse invariant digital filter corresponding to an analog filter satisfying the differential equation

$$\ddot{y} + 2a\,\dot{y} + (a^2 + b^2)\,y = a\,\chi(t) + \dot{\chi}(t) \tag{73}$$

satisfies the difference equation[5] [9]

$$y_{n+2} = 2e^{-ah}\cos bhy_{n+1} - e^{-2ah}\,y_n + h[\chi_{n+2} - e^{-at}\cos bh\,\chi_{n+1}]. \tag{74}$$

Equation (73) corresponds to the transfer function

$$T(s) = \frac{s+a}{(s+a)^2+b^2}$$

which has a pair of complex poles

$$\lambda_{1,2} = -a \pm jb.$$

It is convenient in using the results of section 5 to consider a first order equation with y(t) complex, instead of (73) in which y(t) is real. It is straight forward to show that the first order equation

[5]Digital filter designs are typically normalized to h = 1. This does not seem approximate here and an h multiplying the braketed term in (74) is added to Rader and Gold's equation.

$$y^*(t) = \lambda\, y(t) + \chi(t), \qquad (75)$$

with $y(t)$ and $\chi(t)$ complex and $\lambda = -a+jb$

has the property that $R_e\, y(t)$ in (75) satisfies (73) if the excitation is $R_e\, \chi(t)$. The difference equation corresponding to (74) can similarly be shown to be

$$y_n = K\, y_{n-1} + h\, \chi_n \qquad (76)$$

where y_n is complex and $K = e^{\lambda h}$.

The techniques of section 5 will now be used to determine the accuracy of the solution of (76) as an approximation to the solution of (75). Equation (75) can be identified as being identical to the test problem of (57) if the excitation is chosen to be $\gamma e^{\omega t}$. Hence, the equations for E_r and E_f given in (62) - (72) can be used for accuracy by identifying C_1, C_2, and p for the algorithm of (76).

The principle error function $\varphi(t,Y)$, which is given by (59) for differential equations in the form of (75) or (57), gives implicitly the required constants C_1, C_2 and p.

The principle error function for (76) is evaluated as follows. The chain of approximations

$$\phi(t,Y) \simeq \varphi(\mu,Y) \simeq -\, h^{-(p+1)}\, T_{n+1}\,, \qquad (77)$$

where $(n+1)h \geq \mu \geq n\,h$, has been established and discussed above. Thus, what is required is T_{n+1} given by

$$T_{n+1} = \hat{y}_{n+1} - Y[(n+1)h] \tag{78}$$

where $\hat{y}_{n+1}$ and $Y[(n+1)h]$ are defined with (23) above.

From (78), $\hat{y}_{n+1}$ is expressed as

$$\hat{y}_{n+1} = e^{-(a-jb)h}\, Y(nh) + h\chi[(n+1)h] \; . \tag{79}$$

The true solution of (79) at (n+1)h, Y[(n+1)h], can be expressed in terms of the soultion at nh using the well known analytic solution for linear constant coefficient equations. The result is

$$Y[(n+1)h] = e^{-(a-jb)h}\, Y(nh) + \int_{nh}^{(n+1)h} e^{-(a+jb)[(n+1)h-v]}\, \chi(v)\, dv \; . \tag{80}$$

Using (79) and (80), (78) becomes

$$T_{n+1} = h\, \chi[(n+1)h] - \int_{nh}^{(n+1)h} e^{-(a-jb)[(n+1)h-v]}\, \chi(v)\, dv \; . \tag{81}$$

The integrand of the integral appearing in (81) can be expanded in a Taylor series and the integration carried out term by term to obtain

$$T_{n+1} = \frac{-1}{2} h^2 \left\{ (a-jb)\, \chi[n+1)h] + \dot{\chi}[(n+1)h] \right\} + \text{higher order terms} \tag{82}$$

The fact that T_{n+1} is $O(h^2)$, shows that the algorithm of (76) is first order so that p = 1. Thus using (77),

$$\phi(t,Y) \simeq + \tfrac{1}{2} \left[(a-jb)\, \chi(t) + \dot{\chi}(t) \right] \; . \tag{83}$$

Note that with $\chi(t) = \gamma e^{\omega t}$, $\phi(t,\mathbf{Y})$ becomes

$$\phi(t,Y) \simeq + \frac{1}{2}[\omega-\lambda]\gamma e^{\omega t} \tag{84}$$

which is in the form of (59) with $C_1 = 0$ and $C_2 = \frac{1}{2}$.

It is interesting to note that since $C_1 = 0$, $\phi(t,Y)$ is independent of Y, there is no root shift and $E_r = 0$. This of course results from making use of the analytic solution of the differential equation in obtaining the homogeneous part of the difference equation in the digital filter design.

The forced error E_f is not zero and, in fact, in this example is that of a first order method. Equation (69) gives E_f for this example as

$$E_f = \frac{1}{2}h(\lambda-\omega) \ . \tag{85}$$

and the approximations of (70) through (72) become

$$|E_f| \simeq \frac{1}{2}h|\omega| \ , \ |\lambda| << |\omega| \tag{86}$$

$$|E_f| \simeq h|\lambda| \ , \ |\lambda| \simeq |\omega| \tag{87}$$

$$|E_f| \simeq \frac{1}{2}h|\lambda| \ , \ |\lambda| >> |\omega| \ . \tag{88}$$

Among other conclusions that can be drawn from these expressions is the fact that for a fractional forced error of less than 1%, $h|\omega| < .01$, or 628 samples must be made per cycle of the highest forcing frequency.

6-3 Stiff Equations

In numerical analysis, equations with widely separated eigenvalues are referred to as "stiff" equations. Difference algorithms corresponding to stiff differential equations usually require a considerable computing time for solution. The reason for this fact can be illustrated by considering algorithms for solving an equation whose solution is given by

$$y(t) = A_1 e^{-\lambda_1 t} + A_2 e^{-\lambda_2 t} \tag{89}$$

where λ_1 and λ_2 will be assumed to be real with $\lambda_2 = \alpha\lambda_1$ and α large.

In solving most problems, a solution over at least one time constant of the lowest frequency is typically required. Thus the solution time T must satisfy an equation of the form

$$T = 1/\lambda_1 \tag{90}$$

At the same time the solution must be both stable and accurate in the sense defined above and this imposes an upper limit on $h\lambda$ for all eigenvalues. The latter requirement can be expressed for this example by the equation

$$h\lambda_2 = \alpha h \lambda_1 \leq \beta \tag{91}$$

where β is determined by a particular algorithm.

A relation for the number of steps, N, required to obtain a solution under the condition stated results from combining (90) and (91) to obtain,

$$N \geq \frac{\alpha}{\beta} . \tag{92}$$

Values for β can be obtained from Table V which shows limits on real roots for stability and reasonable root shift for typical algorithms. The values range from 1.05 for the RK-4 to .02 for the Euler method. If α is assumed to be 10^3 the required N becomes 1050 for the RK-4 and 43,500 for the Euler method.

REFERENCES

[1] P. Henrici, Discrete Variable Methods in Ordinary Differential Equations, New York: John Wiley and Sons, 1962.

[2] P. Benyon, "A Review of Numerical Methods for Digital Simulation," Simulation, Vol. 11, No. 5, pp. 219-238, Nov. 1968.

[3] W. Giloi and H. Grebe, "Construction of Multistep Integration Formulas for Simulation Purposes," IEEE Trans on Computers, Vol. C-17, No. 7, pp. 1121-1131, Dec. 1968.

[4] B. Gold and C. M. Rader, Digital Processing of Signals, New York: McGraw-Hill, 1969.

[5] A. Ralston, A First Course in Numerical Analysis, New York: McGraw-Hill, 1965.

[6] E. Isaacson and H. B. Keller, Analysis of Numerical Methods, New York: John Wiley and Sons, 1966.

[7] H. Lomax, "An Operational Unification of Finite Difference Methods for the Numerican Integration of Ordinary Differential Equations," NASA Technical Report NASA TR-R-262, May 1967.

[8] R. S. Ryan, et al., "Use of Wind Shears in the Design of Aerospace Vehicles," J. Spacecraft, Vol. 4, pp. 1526-1532, November 1967.

[9] C. M. Rader and B. Gold, "Digital Filter Design Techniques in the Frequency Domain," Proc IEEE, Vol. 55, pp. 149-171, February 1967.