UNIFORMLY ACCURATE NUMERICAL SOLUTIONS TO DIFFERENTIAL EQUATIONS USING EXTRAPOLATION AND INTERPOLATION

Thesis for the Degree of Ph. D.
MICHIGAN STATE UNIVERSITY
RICHARD ALLAN ROGERS
1974

This is to certify that the

thesis entitled

"Uniformly Accurate Numerical Solutions to Differential Equations Using Extrapolation and Interpolation"

presented by

Richard A. Rogers

has been accepted towards fulfillment of the requirements for

Ph.D. degree in Mathematics

Gerald D. Taylor/
Mary J. Winter
Major professor

Date July 25, 1974

Gerald D. Taylor
Mary J. Winter

O-7639

UNIFORMI
TO DIFFEREN

In this w
for solving ordi
those methods th
all powers of h
a fixed integer.
with such method
grid points belo
we develop the "
combines extrapo
coefficient func
produce a highly
finest mesh. Wh
yields uniform a
mesh.

In Chapt
is modified so a
problems. In ad
of V. Pereyra's
point boundary v

ABSTRACT

UNIFORMLY ACCURATE NUMERICAL SOLUTIONS
TO DIFFERENTIAL EQUATIONS USING EXTRAPOLATION
AND INTERPOLATION

By

Richard Allan Rogers

In this work we are concerned with numerical methods for solving ordinary differential equations. We consider those methods that have asymptotic error expansions involving all powers of h^q, where h is the steplength and q is a fixed integer. The process of extrapolation can be employed with such methods to obtain highly accurate solutions at grid points belonging to the coarsest mesh. In Chapter I we develop the "pullback interpolation method". This method combines extrapolation with Hermite interpolation of the coefficient functions for the asymptotic error expansion to produce a highly accurate solution at all grid points of the finest mesh. When q is 1 or 2 pullback interpolation yields uniform accuracy at all grid points of the finest mesh.

In Chapter II the pullback interpolation method is modified so as to be applicable to boundary value problems. In addition, an elementary proof of the stability of V. Pereyra's finite difference scheme for solving two point boundary value problems is given.

In Chap

equations with c

are shown to be

Because of the p

acy obtained thr

for these proble

A finite

second order del

in Chapter III.

have an asymptot

steplength.

In Chapter III we consider difference differential equations with constant retardation. The methods of Chapter I are shown to be applicable to first order delay equations. Because of the presence of the delay term, the uniform accuracy obtained through pullback interpolation is indispensible for these problems.

A finite difference scheme for directly solving second order delay equations is constructed and analyzed in Chapter III. The global discretization error is shown to have an asymptotic error expansion in even powers of the steplength.

UNIFORM
TO DIFFEREN

Mic
in partial

De

UNIFORMLY ACCURATE NUMERICAL SOLUTIONS TO DIFFERENTIAL EQUATIONS USING EXTRAPOLATION AND INTERPOLATION

By

Richard Allan Rogers

A THESIS

Submitted to
Michigan State University
in partial fulfullment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Mathematics

1974

TO MY WIFE
KAREL

I would

Dr. Gerald D. Ta

and encouragemen

to Dr. Lee M. So

advice and guidan

ACKNOWLEDGEMENTS

I would like to thank my major professors, Dr. Gerald D. Taylor and Dr. Mary J. Winter for their help and encouragement. Also, I want to extend my appreciation to Dr. Lee M. Sonneborn and Dr. Gerald D. Taylor for their advice and guidance over the course of my graduate career.

TABLE OF CONTENTS

LIST OF TABLES

In this

niques for obtai

initial value pr

solutions by mea

will also consid

computed solutio

Consider

$$x'(t) = f$$

(1)

$$x(a) = c$$

We will assume th

depends continuou

Conditions on $f($

known and are giv

The major

are based on disc

imate solution to

$[a,b]$. We will o

points are equall

point set can be

$$G = \{t_n = a$$

where the paramete

INTRODUCTION

In this section we will discuss some standard techniques for obtaining numerical solutions to first order initial value problems. The process of refining computed solutions by means of extrapolation will be explained. We will also consider the question of uniform accuracy of the computed solution.

Consider the first order initial value problem

$$\begin{aligned} &x'(t) = f(t,x(t)), \\ &x(a) = \alpha, \quad a \leq t \leq b. \end{aligned} \tag{1}$$

We will assume that (1) has a unique solution $\varphi(t)$ which depends continuously on the initial condition $x(a) = \alpha$. Conditions on $f(t,x(t))$ which will guarantee this are well known and are given in Chapter I.

The majority of procedures for solving (1) numerically are based on discretization. As such, they yield an approximate solution to (1) on a discrete point set contained in $[a,b]$. We will only consider the case where the discrete points are equally spaced in $[a,b]$. In this case the discrete point set can be conveniently represented as a grid

$$G = \{t_n = a+nh : n=0,1,\ldots,N\},$$

where the parameter $h = \frac{b-a}{N}$ and is called the steplength.

The theo

:(t) and the ap

by $X(t_n,h)$ for

A comput

which takes the

$X(t_{n+j},h)$ and

to be a linear

is referred to a

Three of

rule (sometimes

rule and Gragg's

Euler's

(2) $X(t_{n+1},h)$ =

Equation (2) exhi

is obtained given

imate solution at

at the preceeding

imate solution at

a one-step method

one piece of info

dition in (1). T

everything on the

are trying to com

The theoretical solution to (1) will be denoted by $\varphi(t)$ and the approximate numerical solution will be denoted by $X(t_n,h)$ for each grid point t_n.

A computational method for determining $X(t_{n+k},h)$ which takes the form of a linear relationship between $X(t_{n+j},h)$ and $f(t_{n+j},X(t_{n+j},h))$, $j=0,1,\ldots,k$ is said to be a linear k-step method. The class of all such methods is referred to as the class of linear multistep methods.

Three of the simpler multistep methods are Euler's rule (sometimes called the Euler-Cauchy method), the trapezoid rule and Gragg's modified midpoint rule.

Euler's rule is summarized by the equation

$$(2) \quad X(t_{n+1},h) = X(t_n,h)+hf(t_n,X(t_n,h)), \quad n=0,1,\ldots,N-1.$$

Equation (2) exhibits how the approximate solution at t_{n+1} is obtained given that one has already obtained an approximate solution at t_n. Since information is required only at the preceeding grid point in order to obtain an approximate solution at the next grid point, Euler's rule is a one-step method. To initialize or start the method requires one piece of information which is given by the initial condition in (1). That is, take $X(t_0,h) = x(a) = \alpha$. Since everything on the right hand side of (2) is known when we are trying to compute $X(t_{n+1},h)$, the relationship (2) is

expli

(3)

This

$X(t_0$

not c

ution

becau

Where

speci

$X(t_{n+}$

trape

becau

itly

itera

defin

(4) x

To co

$X(t_{n+1}$

values

be det

explicit and Euler's rule is an explicit multistep method.

The trapezoid rule is given by

$$(3)\quad X(t_{n+1},h) = X(t_n,h) + \frac{h}{2}[f(t_n,X(t_n,h)) + f(t_{n+1},X(t_{n+1},h))],$$

$$n=0,1,\ldots,N-1.$$

This is again a one-step method and is initialized by taking $X(t_0,h) = x(a) = \alpha$. However, the right hand side of (3) is not completely known. Indeed, when trying to compute the solution at t_{n+1} the term $f(t_{n+1},X(t_{n+1},h))$ is unknown because it involves the solution we are trying to compute. Whenever $f(t,x(t))$ is linear in $x(t)$, and in some other special cases, equation (3) can be solved explicitly for $X(t_{n+1},h)$. However, in general this is not possible and the trapezoid rule is said to be an implicit multistep method because of this behavior. When (3) cannot be solved explicitly for $X(t_{n+1},h)$ a root finding procedure or functional iteration can be used to find the solution.

The midpoint rule is an explicit two-step method defined by

$$(4)\quad X(t_{n+2},h) = X(t_n,h) + 2hf(t_{n+1},X(t_{n+1},h)),\quad n=0,1,\ldots,N-2.$$

To compute $X(t_{n+2},h)$ we need to know both $X(t_n,h)$ and $X(t_{n+1},h)$. Thus, the midpoint rule requires two starting values $X(t_0,h) = \alpha$ and $X(t_1,h)$. The second of these can be determined in a variety of ways.

is t

use E

Secor

to ma

proce

and i

Reyno

given

(5)

(3) a

asymp

for a

ezoid

solut

(6)

The f

fixed

Gragg's modification of the midpoint rule [11,12] is twofold. First, to obtain the additional starting value use Euler's rule. That is,

$$X(t_1,h) = \alpha + hf(a,\alpha).$$

Second, at the grid point $t_N = b$ use a smoothing procedure to make the computed solution more stable. The smoothing procedure consists of averaging three computed solutions and is similar to a device originally employed by Milne and Reynolds [20,21]. Gragg's modified midpoint rule is formally given by

$$X(t_1,h) = \alpha + hf(a,\alpha);$$

$$(5)\quad X(t_{n+2},h) = X(t_n,h) + 2hf(t_{n+1},X(t_{n+1},h)), \quad n=0,1,\ldots,N-1;$$

$$X(b,h) = \frac{1}{4}X(t_{N-1},h) + \frac{1}{2}X(t_N,h) + \frac{1}{4}X(t_{N+1},h).$$

The three numerical methods given by equations (2) (3) and (5) have an important similarity. Each has an asymptotic error expansion that involves all powers of h^q for a fixed integer q. That is, for Euler's rule, the trapezoid rule, and Gragg's modified midpoint rule, the computed solution satisfies

$$(6)\quad X(t_n,h) = \varphi(t_n) + \sum_{k=1}^{\infty} e_k(t_n)h^{qk}.$$

The functions $e_k(t)$ are independent of h and q is a fixed integer. For Euler's rule $q=1$ and for the trapezoid

rule

(6)

t_N =

grid

f g c

vati

(6')

The

beha

we s

C i

depe

that

(6)

Modi

Step

of s

trap

Nume

(6)

rule and Gragg's modified midpoint rule $q=2$. The expansion (6) for Gragg's modified midpoint rule is only valid at $t_N = b$ while for the other two methods (6) is valid at all grid points t_n. The expansion (6) is valid only when $f \in C^{\infty}[[a,b] \times (-\infty,\infty)]$.

If f has only a finite number of continuous derivatives, a truncated version of (6) is valid, namely

$$(6') \quad X(t_n,h) = \varphi(t_n) + \sum_{k=1}^{M} e_k(t_n)h^{qk} + \mathcal{O}(h^{q(M+1)}).$$

The notation $\mathcal{O}(h^j)$ means that the function being suppressed behaves like a constant multiplied by h^j as $h \to 0$. Formally, we say that $g(t) = \mathcal{O}(h^j)$ if $\frac{|g(t)|}{h^j} \leq C$ as $h \to 0$, where C is a constant. The length, M, of the expansion (6') depends on q and the number of continuous derivatives of f that exist.

The existence of asymptotic expansions of the form (6) or (6') for Euler's rule, the trapezoid rule and Gragg's modified midpoint rule was originally proved by Gragg [11,12]. Stetter [31] and Pereyra [24] have also studied the existence of such expansions. Gragg's results for Euler's rule and the trapezoid rule are presented in more detail in Chapter I.

The existence of such expansions is important because numerical methods which have error expansions of the form (6) or (6') are amenable to extrapolation. Basically, the

proc

comp

as t

back

Arch

firs

cent

extr

latt

exce

has

solu

as f

supp

a+H

k=0,

All

With

solu

expa

Y+1

process of extrapolation is a means of combining several computed solutions, each of low accuracy, in such a manner as to obtain a computed solution with high accuracy.

Extrapolation is not a recent development, dating back to at least 1654 when Huygens [14] used it to improve Archimedes polygonal approximation to π. Extrapolation was first systematically studied by Richardson [29] early in this century and has often been referred to as either Richardson extrapolation or the deferred approach to the limit, the latter being the title of Richardson's 1927 paper. An excellent survey article on extrapolation and its applications has been written by Joyce [16].

Extrapolation, when applied to obtaining an accurate solution to a differential equation, is most easily explained as follows: Let $H > 0$ be a fixed basic steplength and suppose an accurate solution to (1) is desired at the point $a + H$. Define a sequence of steplengths $h_k = H/2^k$ $k=0,1,\ldots,K$ and grids

$$G_k = \{t_i^k = a+ih_k: \quad i=0,1,\ldots,2^k\}.$$

All grids contain a and $a + H$ and the grids are nested with $G_k \subset G_{k+1}$ $\forall k$. On each grid G_k compute a numerical solution to (1) using a method which has an asymptotic error expansion of the form (6') with $M > K$. At $a + H$ we have $K + 1$ computed solutions $X(t_N,h_k)$ for $k=0,1,\ldots,K$.

Extra

of th

the f

const

(7)

an it

perfo

in (7

suita

at a

able

accu

q_k.

Howe

poin

extr

$a + \frac{h}{2}$

less

extr

Obta

Grid

Extrapolation is the process of forming a linear combination of these $K+1$ solutions in such a manner so as to eliminate the first K error terms of the expansion (6'). That is, constants c_k are determined so that

$$(7) \qquad \sum_{k=0}^{K} c_k X(a+H,h_k) = \varphi(b) + \mathcal{O}(H^{q(K+1)}).$$

Aitken [1] and Neville [22] independently devised an iterative scheme by means of which extrapolation can be performed without explicit computation of the constants c_k in (7). The convergence of this iterative scheme under suitable hypotheses was established by Gragg [11,12].

Note that in order to obtain $\mathcal{O}(H^{q(K+1)})$ accuracy at a grid point you must have $K+1$ computed solutions available to work with. Thus, extrapolation will yield $\mathcal{O}(H^{q(K+1)})$ accuracy only at the point $a+H$ which is common to all grids G_k. Extrapolation can be performed at other grid points. However, it will not yield $\mathcal{O}(H^{q(K+1)})$ accuracy at these points. For instance, using the fact that $G_k \subset G_{k+1}$ $\forall k$, extrapolation will yield $\mathcal{O}(H^{qK})$ accuracy at the midpoint $a+\frac{H}{2}$. At other grid points, extrapolation will yield even less accuracy.

Lindberg [19] has developed a method based on extrapolation and Lagrange interpolation which can be used to obtain $\mathcal{O}(H^{q(K+1)-1})$ accuracy all at grid points of the finest grid G_K.

pull
and
at a
four
and
Lind
inte
nume
pres

ordi
two

/8/

Pere
sche
has

pull
boun
Sect

In Chapter I we present what we have termed "the pullback interpolation method". It utilizes extrapolation and Hermite interpolation to obtain $\mathcal{O}(H^{q(K+1)})$ accuracy at all points of the finest grid when $q=1$ or 2. The first four sections of Chapter I are devoted to developing the method and establishing a theoretical basis for it. In Section 5 Lindberg's method is presented and compared with pullback interpolation both theoretically and numerically. Extensive numerical tests were performed and these results are also presented in Section 5 of Chapter I.

The focus in Chapter I is entirely on first order ordinary differential equations. In Chapter II we consider two point boundary value problems of the form

$$x''(t) = f(t,x(t),x'(t)),$$
$$x(a) = A, \quad x(b) = B. \tag{8}$$

Pereyra [24,25,26,27,28] has developed a finite difference scheme which yields an approximate solution to (8) that has an asymptotic error expansion of the form (6') with $q=2$.

Pereyra's results are summarized in Chapter II and pullback interpolation is modified so as to be applicable to boundary value problems of the form (8). In addition, in Section 2 of Chapter II, we present a new proof of the

stabi

to Pe

of di

ation

are a

shown

Numer

deve

direc

(9)

The

expa

pull

stability of Pereyra's difference scheme. In comparison to Pereyra's proof, ours is considerably more elementary.

In Chapter III we consider the numerical solution of difference differential equations with constant retardation. First order equations

$$\dot{x}(t) = f(t,x(t),x(t-r))$$

are analyzed in Section 1 and pullback interpolation is shown to be a viable solution technique for these problems. Numerical results are presented to support this contention.

The remainder of Chapter III is devoted to the development and analysis of a finite difference method for directly solving second order equations of the form

$$\ddot{x}(t) + f(t,x(t),x(t-r),\dot{x}(t),\dot{x}(t-r)) = 0. \tag{9}$$

The approximate solution to (9) is shown to have an asymptotic expansion of the form (6') with $q=2$. A modification of pullback interpolation is shown to be applicable.

Sect

valu

(1)

We s

of

resp

it i

solu

the

valu

Valu

of t

a sy

also

equa

equa

can

be c

CHAPTER I

THE PULLBACK INTERPOLATION METHOD FOR INITIAL VALUE PROBLEMS

Section 1. Statement of the Problem

In this chapter we consider the first order initial value problem.

$$\begin{aligned} y'(t) &= f(t,y(t)) \\ y(a) &= \alpha \qquad a \leq t \leq b. \end{aligned} \tag{1}$$

We shall assume that $f(t,y(t))$ is a continuous function of t and satisfies a uniform Lipschitz condition with respect to its second argument. Under these assumptions it is well known (see Keller [17]) that (1) has a unique solution, $\varphi(t)$, which depends Lipschitz continuously on the initial data $y(a) = \alpha$. $y(t)$ may be either a scalar-valued or a vector-valued function. If $y(t)$ is vector-valued then $f(t,y(t))$ will be a vector-valued function of the variable t and the vector $y(t)$ and (1) will be a system of first order differential equations. This case also arises when we reduce a m^{th} order differential equation to a system of m first order differential equations. The standard technique for accomplishing this can be found in Lambert [18]. The numerical methods to be considered for solving (1) will work for either the

scalar

extrap

this c

of the

the sa

withou

refine

on a p

Corlis

(1), I

k=0,1,

g_k =

2^k+1 p

$t^k_{2^k}$ =

for th

give a

tion,

sion

(2)

for ea

coeffi

b_k, fo

scalar or vector-valued case. In the vector-valued case extrapolation and the pullback method, to be explained in this chapter, can be done independently for each component of the solution vector to obtain a refined solution. Since the same work is done for each component, one at a time, without using any results involving other components, the refinement process appears well suited to implementation on a parallel processing computer such as ILLIAC IV (see Corliss [5]).

Turning our attention to the numerical solution of (1), let $h>0$ be a fixed basic steplength and for each $k=0,1,\ldots,K$ define steplengths $h_k = h/2^k$ and grids $G_k = \{t_i^k : t_i^k = a+ih_k,\ i=0,1,\ldots 2^k\}$. Each grid G_k contains 2^k+1 points; all grids contain the two points $t_0^k = a$ and $t_{2^k}^k = a+h$; and the grids are nested, that is, $G_k \subset G_{k+1}\ \forall k$.

In what follows we shall be concerned with methods for the numerical solution of (1) on the grids G_k which give an approximation, $Y(t_i^k,h_k)$, to the theoretical solution, $\varphi(t_i^k)$, such that the error has an asymptotic expansion

$$(2) \qquad Y(t_i^k,h_k) = \varphi(t_i^k) + \sum_{j=1}^{M} h_k^{jq} e_j(t_i^k) + O(h_k^{(M+1)q})$$

for each $i=0,1,\ldots,2^k$ and for each $k=0,1,\ldots,K$. The coefficient error functions, $e_m(t)$, are independent of h_k, for each k, and q is a fixed integer (usually 1 or 2)

peculi

of the

employ

exist.

f(t,y(

with

soluti

and S

expans

expans

of th

rule

above

funct

equat

(3)

The a

the t

e_1...

there

funct

The l

peculiar to the method being employed. The exact length of the expansion (2) will depend on both the method being employed and the number of derivatives of $f(\cdot,\cdot)$ which exist. In general, $(M+1)q$ continuous derivatives of $f(t,y(t))$ with respect to t are required for an expansion with M error terms. This is equivalent to the theoretical solution $\varphi(t)$ having $(M+1)q+1$ continuous derivatives.

As mentioned before, Gragg [11,12], Pereyra [24], and Stetter [31] have investigated the existence of such expansions. Examples of methods which yield such error expansions are Euler's method $(q=1)$, the usual generalization of the trapezoid rule $(q=2)$ and Gragg's modified midpoint rule $(q=2)$. An important result obtained in each of the above mentioned studies is that the coefficient error functions $e_m(t)$ satisfy an inhomogeneous linear variational equation on $a\leq t\leq b$, of the form

$$e_m'(t) - J(t)e_m(t) = b_m(\cdot)$$
$$e_m(a) = 0, \qquad m=1,\ldots,M. \tag{3}$$

The arguments of the inhomogeneous terms, $b_m(\cdot)$, involve the theoretical solution $\varphi(t)$, previous error functions $e_1,\ldots,e_{m-1}$, the function $f(t,y(t))$, and various derivatives thereof. The differentiability of the various error functions depends on the differentiability of $f(t,y(t))$. The left hand side of (3) is the Frechet derivative of (1),

considered as a differential operator, operating on $e_m(t)$. Alternately, $J(t)$ is the Jacobian of $f(t,y(t))$ evaluated at the theoretical solution, $\varphi(t)$, of (1). The left hand side of (3) may be obtained formally by assuming $y(t)$ depends on a parameter λ, differentiating (1) with respect to this parameter and setting $e_m' = \frac{dy'}{d\lambda}$ and $e_m = \frac{dy}{d\lambda}$.

If we compute the numerical solution of (1) on the grids G_k $k=0,1,\ldots,M$ using a method which has an expansion of the form (2), we may then employ extrapolation to obtain a solution $Y(a+h) = \varphi(a+h) + \mathcal{O}(h^{(M+1)q})$. However we are not able to obtain comparable accuracy at the intermediate points. For instance, using extrapolation, the solution at the midpoint satisfied $Y(a+\frac{h}{2}) = \varphi(a+\frac{h}{2}) + \mathcal{O}(h^{Mq})$. And, as the following example illustrates, we cannot interpolate several extrapolated values to obtain a solution with equivalent accuracy.

Example 1: $\quad y' = y^2; \quad y(0) = .2; \quad 0 \leq t \leq 3.$

The theoretical solution to this problem is $\varphi(t) = \frac{1}{5-t}$. Using the trapezoid rule and extrapolation with $h=1$, $M=3$ we compute the solution, $Y(t)$, at the points $t=1,2$, and 3 by resolving the same problem three times with initial conditions determined by the computed solution at $t=1$ and 2. The results are given in Table 1 below.

for

Two

Lag

(t,

Here

the

t

0

1

2

3

t

$\frac{1}{2}$ ·2

·2

·4

reve

that

A variety of interpolation schemes are possible for determining the solution at intermediate grid points. Two such are summarized in Table 2 below. $L(t)$ is the Lagrange cubic interpolation polynomial for the data $(t,Y(t))$ given in Table 1 and $H(t)$ is the quartic Hermite interpolation polynomial for the same data with the added condition $Y'(0) = (Y(0))^2 = (.2)^2 = .04$.

TABLE 1

t	$\varphi(t)$	$Y(t)$	$\varphi(t) - Y(t)$
0	.200 000 000	.200 000 000	0
1	.250 000 000	.250 000 000	0
2	.333 333 333	.333 333 330	3×10^{-9}
3	.5000 000 000	.499 999 762	2.38×10^{-7}

TABLE 2

t	$\varphi(t)$	$L(t)$	$\varphi(t)-L(t)$	$H(t)$	$\varphi(t)-H(t)$
$\frac{1}{2}$	.222222222	.223958320	-1.736×10^{-3}	.222395831	-1.736×10^{-4}
$\frac{3}{2}$	.285714285	.284375013	1.339×10^{-3}	.285312506	4.018×10^{-4}
$\frac{5}{2}$	.400000000	.403124923	-3.125×10^{-3}	.401562434	-1.562×10^{-3}

As an examination of Tables 1 and 2 quickly reveals, neither of the interpolation schemes gives accuracy that is comparable to that of the computed solution.

In the next section we will present and analyze the pullback interpolation method. It is based on a Hermite interpolation scheme for approximating the error functions $e_m(t)$ successively, beginning with the last term of the error expansion. The details are worked out for a general expansion of the form (2). However in practice q is usually either one or two and the reader may find it helpful to bear this fact in mind.

Section 2. The Pullback Interpolation Method

Still assuming that our numerical method and the problem at hand are such that the expansion (2) is valid, we now take $K = M$. At the point $t = a+h \in G_0$ we have the $M+1$ computed solutions and error expansions

$$(4) \quad Y(t,h_k) = \varphi(t) + \sum_{j=1}^{M} h_k^{jq} e_j(t) + \mathcal{O}(h_k^{(M+1)q}), \quad k=0,1,\ldots,M.$$

Equation (4) can be regarded as a system of $M+1$ polynomial equations in h. Since the h_k's are distinct these polynomials are linearly independent and (4) may be solved as a matrix system for the unknown vector $(\varphi(t), e_1(t), e_2(t), \ldots, e_M(t))^T$.

In matrix form (4) is given by

$$(4') \qquad AU = Y + E$$

where

$$A = \begin{bmatrix} 1, & h_0^q, & h_0^{2q}, & \ldots, & h_0^{(M-1)q}, & h_0^{Mq} \\ 1, & h_1^q, & h_1^{2q}, & \ldots, & h_1^{(M-1)q}, & h_1^{Mq} \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ 1, & h_M^q, & h_M^{2q}, & \ldots & h_M^{(M-1)q}, & h_M^{Mq} \end{bmatrix}$$

is $(M+1) \times (M+1)$; and U, Y and E are $(M+1) \times 1$ vectors given by

U =

resp

obta

star

tell

dens

of e

whe

$h^{(j)}$

the

ind

or

red

J =

sol

num

the

$$U = \begin{bmatrix} \varphi(t) \\ e_1(t) \\ \vdots \\ e_M(t) \end{bmatrix} ; \quad Y = \begin{bmatrix} Y(t,h_0) \\ Y(t,h_1) \\ \vdots \\ Y(t,h_M) \end{bmatrix} ; \text{ and } \quad E = \begin{bmatrix} \mathcal{O}(h_0^{(M+1)q}) \\ \mathcal{O}(h_1^{(M+1)q}) \\ \vdots \\ \mathcal{O}(h_M^{(M+1)q}) \end{bmatrix}$$

respectively.

Define (A,I) to be the $(M+1) \times 2(M+1)$ matrix obtained by adjoining the identity matrix to A. A standard theorem from linear algebra (see Cullen [6]) tells us that if (A,I) is row equivalent to (I,B) — denoted by $(A,I) \sim (I,B)$ — then $B = A^{-1}$.

Since A is nonsingular we can find a sequence of elementary row operations such that $(A,I) \sim (D,C)$ where D is the diagonal matrix whose (j,j) entry is $h^{(j-1)q}$, $j=1,\ldots,M+1$. Moreover, since $\forall\, k\ h_k = h/2^k$, the row operations which accomplish this reduction are independent of h, in that each involves multiplication or division by a constant only. (D,C) can be further reduced to (I,A^{-1}) by dividing the j^{th} row by $h^{(j-1)q}$.

Multiplying both sides of (4') by A^{-1} we have $U = A^{-1}AU = A^{-1}Y + A^{-1}E$. The vector U is the precise solution to the system (4) and the vector $A^{-1}Y$ is the numerical solution we can obtain. The vector $A^{-1}E$ is the vector of errors for the numerical solution.

E =

Since

of h^{-}

entrie

$S/h^{(M+}$

soluti

equal

with a

A

obtain

$h^{-1}Y$

U. Th

soluti

accura

We can

$S_w/a)$

Now $\forall k\ h_k \leq h$, so we can write

$$E = \begin{bmatrix} \mathcal{O}(h_0^{(M+1)q}) \\ \mathcal{O}(h_1^{(M+1)q}) \\ \vdots \\ \mathcal{O}(h_M^{(M+1)q}) \end{bmatrix} = \begin{bmatrix} \mathcal{O}(h^{(M+1)q}) \\ \mathcal{O}(h^{(M+1)q}) \\ \vdots \\ \mathcal{O}(h^{(M+1)q}) \end{bmatrix} .$$

Since each component of the j^{th} row of A^{-1} has a factor of $h^{-(j-1)q}$ and this is the only dependence of these entries on h, the j^{th} component of $A^{-1}E$ is $\mathcal{O}(h^{(M+1)q-(j-1)q}) = \mathcal{O}(h^{(M-j+2)q})$. Thus our computed solution $A^{-1}Y = (\bar{\varphi}(t), \bar{e}_1(t), \bar{e}_2(t), \ldots, \bar{e}_M(t))^T$ will be equal to the precise solution $U = (\varphi(t), e_1(t), e_2(t), \ldots, e_M(t))^T$ with an error of magnitude

$$A^{-1}E = (\mathcal{O}(h^{(M+1)q}), \mathcal{O}(h^{Mq}), \mathcal{O}(h^{(M-1)q}), \ldots, \mathcal{O}(h^q))^T.$$

It should be emphasized that what we actually obtain when computing a solution to (4) is the vector $A^{-1}Y$ which is only an approximation to the actual solution U. This can be conveniently summarized by saying the solution to (4), $(\varphi(t), e_1(t), \ldots, e_M(t))^T$, is known with accuracy $(\mathcal{O}(h^{(M+1)q}), \mathcal{O}(h^{Mq}), \ldots, \mathcal{O}(h^q))^T$.

Since $e_M(a) = 0$ and $\varphi(a)$ is known exactly; we can obtain $e_M'(a)$ exactly from (3). That is, $e_M(a) = 0$ implies that $e_M'(a) = b_M(a)$. To evaluate

$b_M(a)$ requi

$c(t)$, $J(t)$,

evaluated at

to us. Thes

differentiat

details will

this chapter

Thus

know $e_M(a+h$

interpolatin

pieces of da

Hermite inte

known to $\mathcal{C}$

Now $\forall k=1,..$

any positive

$\forall k=1,\ldots,M$

(5) h_k^{Mq}

where $\beta_M \equiv$

$\gamma_M = (M+1)q$

Proc

point in $G_$

and G_1 cor

such t, nam

this $t \in G_k$

solutions an

$b_M(a)$ requires knowledge of higher order derivatives of $\varphi(t)$, $J(t)$, $f(t,\varphi(t))$ and the errors $e_m(t)$, $m=1,\ldots,M-1$, evaluated at the point $t=a$, all of which are available to us. These derivatives may be obtained by successively differentiating (1) and (3) with respect to t. The details will be worked out for specific methods later in this chapter.

Thus we know $e_M(a)$ and $e_M'(a)$ exactly and we know $e_M(a+h)$ to $\mathcal{O}(h^q)$. Construct $P_M(t)$ the Hermite interpolating polynomial of degree 2 to the above three pieces of data. It is well known that the error in this Hermite interpolation is $\mathcal{O}(h^3)$. Thus, since $e_M(a+h)$ is known to $\mathcal{O}(h^q)$ we have $P_M(s) = e_M(s) + \mathcal{O}(h^3) + \mathcal{O}(h^q)$. Now $\forall k=1,\ldots,M$, $h_k<h$ implies $h_k^\gamma < h^\gamma$ where γ is any positive integer and therefore $\mathcal{O}(h_k^\gamma) < \mathcal{O}(h^\gamma)$. Thus $\forall k=1,\ldots,M$ and $\forall s \in [a,a+h]$ we have

$$(5) \qquad h_k^{Mq} P_M(s) = h_k^{Mq} e_M(s) + \mathcal{O}(h^{\beta_M})$$

where $\beta_M \equiv \min((M+1)q,\ Mq+3)$. Note that for $q \leq 3$ $\beta_M = (M+1)q$.

Proceeding to the grid G_1, let t be any point in $G_1 \backslash G_0$ (since G_0 contains $2^0 + 1 = 2$ points and G_1 contains $2^1 + 1 = 3$ points, there is only one such t, namely $t = a + h/2$). Since the grids are nested this $t \in G_k$ $\forall k=1,\ldots,M$ and we have the M computed solutions and error expansions

(6) $Y(t,h$

Substituting
of M linea
unknown vect

(7) $Y(t,h_k)$

because $\beta_M \leq$

Solvi
$(c(t),e_1(t)...$
$(\Im(h^{\beta}M)_.$

We now
For $t = a+h \epsilon$
and for $t = a$
$\Im(h^{(\beta}M^{-(M-1)}q_)$
mine $e'_{M-1}(a)$
the Hermite int
interpolating t
of interpolatio
$\gamma_s \epsilon [a,a+h]$.

$$(6)\qquad Y(t,h_k) = \varphi(t) + \sum_{j=1}^{M-1} h_k^{jq} e_j(t) + h_k^{Mq} e_M(t) + \mathcal{O}(h_k^{(M+1)q})$$

for $k=1,\ldots,M$.

Substituting (5) into (6), we obtain the following system of M linearly independent equations in h for the unknown vector $(\varphi(t), e_1(t), \ldots, e_{M-1}(t))^T$

$$(7)\quad Y(t,h_k) - h_k^{Mq} P_M(t) = \varphi(t) + \sum_{j=1}^{M-1} h_k^{jq} e_j(t) + \mathcal{O}(h_k^{(M+1)q}) + \mathcal{O}(h^{\beta_M})$$

$$= \varphi(t) + \sum_{j=1}^{M-1} h_k^{jq} e_j(t) + \mathcal{O}(h^{\beta_M}),$$

$$k=1,\ldots,M,$$

because $\beta_M \leq (M+1)q$ and $h_k < h$.

Solving (7) we obtain the solution vector $(\varphi(t), e_1(t), \ldots, e_{M-1}(t))^T$ with accuracy

$$(\mathcal{O}(h^{\beta_M}), \mathcal{O}(h^{\beta_M - q}), \mathcal{O}(h^{\beta_M - 2q}), \ldots, \mathcal{O}(h^{\beta_M - (M-1)q}))^T$$

We now have the following information about $e_{M-1}(s)$. For $t = a+h \in G_0$ we know $e_{M-1}(t)$ with accuracy $\mathcal{O}(h^{2q})$ and for $t = a+h/2 \in G_1 \setminus G_0$ we know $e_{M-1}(t)$ with accuracy $\mathcal{O}(h^{(\beta_M - (M-1)q)})$. As before $e_{M-1}(a) = 0$ and we can determine $e'_{M-1}(a)$ exactly. Thus we can construct $P_{M-1}(s)$ the Hermite interpolation polynomial of degree three interpolating these four data points. Since the error of interpolation will be $\mathcal{O}(h^4)$ and $h_k < h$ we have $\forall s \in [a, a+h]$,

(8) $h_k^{(M-1)q} p_{M-1}$

where $\beta_{M-1} \equiv$

fact that β_M

$\beta_{M-1} = (M+1)q.$

Proceed

$\ldots, P_{M-J}(s)$ f

$\forall j=0,1,\ldots,J,$

nomial of degr

(9) $h_k^{(M-j)q} p_{M-}$

The numbers β

(10) $\beta_{M-j} \equiv$

and β_M is de

Let t

G_{J+1} contains

points there a

$G_k \subset G_{k+1} \quad \forall k$

G_k for $k=J+2$

the following

in h:

$$(8)\quad h_k^{(M-1)q} P_{M-1}(s) = h_k^{(M-1)q} e_{M-1}(s) + \mathcal{O}(h^{(M-1)q+4}) + \mathcal{O}(h^{(M+1)q}) + \mathcal{O}(h^{\beta_M})$$

$$= h_k^{(M-1)q} e_{M-1}(s) + \mathcal{O}(h^{\beta_{M-1}}),$$

where $\beta_{M-1} \equiv \min(\beta_M, (M-1)p+4)$. Here we have used the fact that $\beta_M \leq (M+1)q$. We note that for $q \leq 2$, $\beta_{M-1} = (M+1)q$.

Proceeding by induction, suppose $P_M(s)$, $P_{M-1}(s)$, $\ldots, P_{M-J}(s)$ for $J < M-1$ have all been constructed, where $\forall\, j=0,1,\ldots,J$, $P_{M-j}(s)$ is a Hermite interpolation polynomial of degree 2^j+1 with

$$(9)\quad h_k^{(M-j)q} P_{M-j}(s) = h_k^{(M-j)q} e_{M-j}(s) + \mathcal{O}(h^{\beta_{M-j}}) \qquad \forall s \in [a, a+h].$$

The numbers β_{M-j} are defined by

$$(10)\quad \beta_{M-j} \equiv \min(\beta_{M-j+1},\ (M-j)q + 2^j + 2),\quad j=1,\ldots,J,$$

and β_M is defined by (5). Note that $\beta_{M-j} \leq \beta_{M-j+1}$ $\forall j$.

Let t be any point in $G_{J+1} \setminus G_J$. Since G_{J+1} contains $2^{J+1}+1$ points and G_J contains 2^J+1 points there are $2^{J+1} - 2^J = 2^J$ such points. Since $G_k \subset G_{k+1}$ $\forall k$ each $t \in G_{J+1} \setminus G_J$ is also an element of G_k for $k=J+2, J+3, \ldots, M$. Thus at each such t we have the following system of $M-J$ linearly independent equations in h:

(11) $Y(t,h_k)$

since $\beta_{M-J} \geq \beta$

the system (11)

we obtain the

$(\mathcal{O}(h^{\beta_{M-J}}), \mathcal{O}(h$

From s

obtain knowled

accuracy $\mathcal{O}(h^{\beta}$

know $e_{M-J-1}(a$

$\mathcal{O}(h^{(J+2)q})$. F

$e_{M-J-1}(a+h/2)$

inductive hypo

of $e_{M-J-1}(t)$

$\mathcal{O}(h^{\beta_{M-J}}(M-J-1$

$e_{M-J-1}(t)$ at

with varying a

$e_{M-J-1}(a) = 0$

$$(11)\quad Y(t,h_k) - \sum_{j=0}^{J} h_k^{(M-j)q} P_{M-j}(t)$$

$$= \varphi(t) + \sum_{j=1}^{M-J-1} h_k^{jq} e_j(t) + \mathcal{O}(h_k^{(M+1)q}) + \sum_{j=0}^{J} \mathcal{O}(h^{\beta_{M-j}})$$

$$= \varphi(t) + \sum_{j=1}^{M-J-1} h_k^{jq} e_j(t) + \mathcal{O}(h^{\beta_{M-J}}), \quad k=J+1, J+2, \ldots, M;$$

since $\beta_{M-J} \leq \beta_{M-J+1} \leq \cdots \leq \beta_{M-1} \leq \beta_M \leq (M+1)q$. Solving the system (11) for the unknown $(\varphi(t), e_1(t), \ldots, e_{M-J-1}(t))^T$, we obtain the solution with accuracy

$$(\mathcal{O}(h^{\beta_{M-J}}), \mathcal{O}(h^{\beta_{M-J}-q}), \mathcal{O}(h^{\beta_{M-J}-2q}), \ldots, \mathcal{O}(h^{\beta_{M-J}-(M-J-1)q}))^T.$$

From solving (11) at each point in $G_{J+1} \setminus G_J$ we obtain knowledge of $e_{M-J-1}(t)$ at 2^J points with accuracy $\mathcal{O}(h^{\beta_{M-J}-(M-J-1)q})$. From our work on G_0 we know $e_{M-J-1}(a+h)$ with accuracy $\mathcal{O}(h^{(M+1)q-(M-J-1)q}) = \mathcal{O}(h^{(J+2)q})$. From our work on $G_1 \setminus G_0$ we know $e_{M-J-1}(a+h/2)$ with accuracy $\mathcal{O}(h^{\beta_M-(M-J-1)q})$. By our inductive hypothesis $\forall j < J$, we have obtained knowledge of $e_{M-J-1}(t)$ at 2^j points in $G_{J+1} \setminus G_j$ with accuracy $\mathcal{O}(h^{\beta_{M-J}-(M-J-1)q})$. Thus we know

$$e_{M-J-1}(t) \text{ at } 1 + \sum_{j=0}^{J} 2^j = 1 + 2^{J+1} - 1 = 2^{J+1} \text{ points}$$

with varying accuracies. In addition we also have $e_{M-J-1}(a) = 0$ and we can determine $e'_{M-J-1}(a)$ exactly.

Thus we have k

and we know ϵ

interpolation

above data. T

$O(h^{2^{J+1}+2})$ an

(12) $h_k^{(M-J-1)}$

where β_{M-J-1}

the error in ir

$O(h^{(M-J-1)q+2^{J+}}$

$\min(h_k^{(M-J-1)q}$

=

=

=

since $\beta_{M-J} \leq$ (

Hence b

$j=0,1,\ldots,M-1$

(9) and (10) ar

On the

expression for

*see the commen-

Thus we have knowledge of $e_{M-J-1}(t)$ at $2^{J+1}+1$ points and we know $e'_{M-J-1}(a)$. Construct $P_{M-J-1}(s)$ the Hermite interpolation polynomial of degree $2^{J+1}+1$ based on the above data. The error of interpolation will be $\mathcal{O}(h^{2^{J+1}+2})$ and $\forall s \in [a,a+h]$,

$$(12)\quad h_k^{(M-J-1)q} P_{M-J-1}(s) = h_k^{(M-J-1)q} e_{M-J-1}(s) + \mathcal{O}(h^{\beta_{M-J-1}}),$$

where β_{M-J-1} is determined in the following manner: the error in interpolation is $h_k^{(M-J-1)q}\mathcal{O}(h^{2^{J+1}+2}) = \mathcal{O}(h^{(M-J-1)q+2^{J+1}+2})$ and the error in the given data is

$$\min(\{h_k^{(M-J-1)q}\mathcal{O}(h^{\beta_{M-j}-(M-J-1)q})\}_{j=0}^{J},\ h_k^{(M-J-1)q}\mathcal{O}(h^{(J+2)q}))$$

$$= \min(\{\mathcal{O}(h^{\beta_{M-j}})\}_{j=0}^{J},\ \mathcal{O}(h^{(M+1)q}))$$

$$= \min(\mathcal{O}(h^{\beta_{M-J}}),\ \mathcal{O}(h^{(M+1)q}))^{*}$$

$$= \mathcal{O}(h^{\beta_{M-J}})$$

since $\beta_{M-J} \le (M+1)q$. Thus $\beta_{M-J-1} = \min(\beta_{M-J}, (M-J-1)q+2^{J+1}+2)$.

Hence by induction we can determine $P_{M-j}(s)$ for $j=0,1,\ldots,M-1$ such that $P_{M-j}(s)$ has degree 2^j-1 and (9) and (10) are valid for $j=0,1,\ldots,M-1$.

On the last grid G_M we have the following expression for $\varphi(t)$ for each $t \in G_M \backslash G_{M-1}$,

*see the comment following (11)

(13) $Y(t,h_M$

since $\beta_1 = \beta_2$

To o

points in G

For $t = a+h$

the computed

$t \in G_M \setminus G_{M-1}$

$t \in G_{M-1}$, sin

there exists

$t \in G_{J+1} \setminus G_J$.

the first co

the system (

Sinc

solution $Y($

(14)

Befo

point out tha

of the iterat

extrapolation

$$(13)\quad Y(t,h_M) - \sum_{j=0}^{M-1} h_M^{(M-j)q} P_{M-j}(t)$$

$$= \varphi(t) + \mathcal{O}(h_M^{(M+1)q}) + \sum_{j=0}^{M-1} \mathcal{O}(h^{\beta_{M-j}})$$

$$= \varphi(t) + \mathcal{O}(h^{\beta_1})$$

since $\beta_1 \leq \beta_2 \leq \ldots \leq \beta_M \leq (M+1)q$.

To obtain the final solution, $Y(t)$, at all grid points in G_M, the finest grid, we proceed as follows. For $t = a+h$, $Y(t)$ is the first component, $\bar{\varphi}(t)$, of the computed solution to the system (4). For $t \in G_M \setminus G_{M-1}$, $Y(t)$ is the solution of (13). For any $t \in G_{M-1}$, since $\bigcup_{k=0}^{M} T_k = G_M$ and the G_k's are nested, there exists an index J, depending on t, such that $t \in G_{J+1} \setminus G_J$. The solution at this t, $Y(t)$, will be the first component, $\bar{\varphi}(t)$, of the computed solution to the system (11).

Since β_1 is the smallest of the β_j's, the solution $Y(t)$ constructed above satisfies

$$(14)\qquad Y(t) = \varphi(t) + \mathcal{O}(h^{\beta_1}).$$

Before we examine the β_j's in detail we should point out that the pullback method does not require use of the iterative scheme of Aitken and Neville to perform extrapolation. In the pullback method extrapolation is

actually ac

systems.

It

for the sys

matrix for

matrix of

the matrix

progressive

Spe

system (4)

(15)
$$\begin{bmatrix} 1, & & & & \\ & & & & \\ 1, & & & & \\ . & & & & \\ . & & & & \\ . & & & & \\ 1, & & & & \end{bmatrix}$$

The matrix

$$\begin{bmatrix} 1, & & & & \\ & & & & \\ 1, & & & & \\ . & & & & \\ . & & & & \\ . & & & & \\ 1, & & & & \end{bmatrix}$$

This matrix

first row a

actually accomplished when we solve the various matrix systems.

It should also be pointed out that the matrices for the systems (11) are "nested" in the sense that the matrix for the J^{th} system is an easily obtained sub-matrix of the (J-1)st system. Thus we need only define the matrix for (4) and the others can be obtained by progressively deleting rows and columns of this matrix.

Specifically, recall that the matrix for the system (4) - the case j=0 - is given by

$$(15)\qquad \begin{bmatrix} 1, & h_0^q, & h_0^{2q}, & \ldots\ldots, & h_0^{(M-1)q}, & h_0^{Mq} \\ 1, & h_1^q, & h_1^{2q}, & \ldots\ldots, & h_1^{(M-1)q}, & h_1^{Mq} \\ \vdots & \vdots & \vdots & & & \\ 1, & h_M^q, & h_M^{2q}, & \ldots\ldots, & h_M^{(M-1)q}, & h_M^{Mq} \end{bmatrix}$$

The matrix for the system (7) - the case j=1 - is given by

$$\begin{bmatrix} 1, & h_1^q, & h_1^{2q}, & \ldots\ldots, & h_1^{(M-1)q} \\ 1, & h_2^q, & h_2^{2q}, & \ldots\ldots, & h_2^{(M-1)q} \\ \vdots & \vdots & \vdots & & \\ 1, & h_M^q, & h_M^{2q}, & \ldots\ldots, & h_M^{(M-1)q} \end{bmatrix}$$

This matrix can be obtained from (15) by deleting the first row and the last column of (15).

For ge

is given by

$$\begin{bmatrix} 1, & \\ & \\ 1, & \\ \cdot & \\ \cdot & \\ \cdot & \\ \cdot & \\ 1, & \end{bmatrix}$$

which can eas

rows and the

Conse

it is only ne

For l

known to be i

have develope

Vandermonde s

system.

Turni

examine the c

section.

For general J the matrix of the system (11) is given by

$$\begin{bmatrix} 1, & h_{J+1}^{q}, & h_{J+1}^{2q}, & \ldots., & h_{J+1}^{(M-J-1)q} \\ 1, & h_{J+2}^{q}, & h_{J+2}^{2q}, & \ldots., & h_{J+2}^{(M-J-1)q} \\ \vdots & \vdots & & & \\ 1, & h_{M}^{q}, & h_{M}^{2q}, & \ldots., & h_{M}^{(M-J-1)q} \end{bmatrix},$$

which can easily be obtained by deleting the first $J+1$ rows and the last $J+1$ columns of the matrix (15).

Consequently, when using the pullback technique it is only necessary to define one matrix.

For large M the Vandermonde matrix (15) is known to be ill-conditioned. Björck and Pereyra [2] have developed an efficient algorithm for solving such Vandermonde systems, which can be utilized for solving our system.

Turning our attention to the β_j's, we shall examine the cases $q=1$ and $q=2$ in detail in the next section.

Section

Lemma 1

Proof.

Hence

with

x is

the v

this

(16)

and

(17)

Theo

Section 3. The Pullback Method for q=1 and 2

We will need the following rather obvious lemma.

Lemma 1. a) Let $f(x) = -2x + 2^{x+1}$ then $\forall\, x \in [1,\infty)$, $f(x) \geq 2$.

b) Let $g(x) = -x + 2^{x+1}$ then $\forall\, x \in [1,\infty)$, $g(x) \geq 2$.

Proof. a) $f'(x) = -2 + 2^{x+1} \ln 2 > -2 + 2^x \geq 0$ for $x \geq 1$. Hence on $[1,\infty)$ $f(x)$ is a monotone increasing function with $f(1) = 2$ and therefore $f(x) \geq 2$.

b) $g(x) = f(x) + x \geq f(x) \geq 2$ by part a) since x is positive. □

Our previous inductive argument has established the validity of equation (10) for $j=1,\ldots,M-1$. From this and (5) we can conclude for $q=2$ that

$$\begin{aligned} \beta_M &= \min((M+1)2, 2M+3) = (M+1)2 \\ \beta_{M-j} &= \min(\beta_{M-j+1}, (M-j)2 + 2^j + 2), \quad j=1,\ldots,M-1; \end{aligned} \tag{16}$$

and for $q=1$ that

$$\begin{aligned} \beta_M &= \min(M+1, M+3) = M + 1 \\ \beta_{M-1} &= \min(\beta_{M-j+1}, M-j + 2j + 2), \quad j=1,\ldots,M-1. \end{aligned} \tag{17}$$

Theorem 1. For $q=1$ and 2 we have

$$\beta_{M-j} = (M+1)q, \quad j=0,1,\ldots,M-1.$$

Proof.

and (17)

(16) and

from ou

for q=

q=1 or

which e

the pul

at ever

extrapo

will be

the acc

Proof. The proof is by induction on j. From (16) and (17) we have $\beta_M = (M+1)q$ and for $j=1$ we have

$$\beta_{M-1} = \min(\beta_M, (M-1)q + 2 + 2)$$

$$= \min((M+1)q, (M-1)q + 2 + 2)$$

$$= (M+1)q.$$

Assume $\beta_{M-j} = (M+1)q$ where $1<j<M-1$. Then by (16) and (17)

$$\beta_{M-(j+1)} = \beta_{M-j-1} = \min(\beta_{M-j}, (M-j-1)q + 2^{j+1} + 2)$$

$$= \min((M+1)q, Mq-qj + 2^{j+1} - q + 2)$$

from our inductive assumption. By Lemma 1 $-qj + 2^{j+1} \geq 2$ for $q=1$ or 2 and obviously $-q + 2 \geq 0$. Thus for $q=1$ or 2

$$(M+1)q \leq Mq + 2 \leq Mq - qj + 2^{j+1} - q + 2$$

which establishes that $\beta_{M-(j+1)} = (M+1)q$. □

Theorem 1 shows that for the cases $q=1$ and 2 the pullback interpolation method yields the same accuracy at every point of the finest grid as that obtained by extrapolation at the endpoint, $a+h$.

If M is large then the polynomials P_0, P_1, etc.; will be of high degree and "Runge's phenomenon" may destroy the accuracy of our computed solution. This can be

circumvented

nomials. Exa

limitation on

to which we k

need only int

accuracy comp

affect the ov

"Runge's phen

Befor

P_{M-j} we need

$e'_m(a)$ is giv

few more comm

pullback meth

In th

an ℓ-dimensi

and $e_j(t)$ a

the applicabl

solved ℓ ti

Also a comple

must be compu

point out the

computers in

There

not all the a

circumvented to some degree by using lower degree polynomials. Examining the proof of Theorem 1 we see that the limitation on the accuracy we can obtain is the accuracy to which we know e_{M-j}. Thus for large M and j we need only interpolate on enough points to guarantee accuracy comparable to that of our data. This will not affect the overall accuracy of our scheme and may avoid "Runge's phenomenon".

Before we can actually compute the polynomials P_{M-j} we need to know $e'_{M-j}(a)$. The method for determining $e'_m(a)$ is given in the next section but at this point a few more comments on the actual implementation of the pullback method are in order .

In the case where $y(t) = (y^1(t)\ldots,y^{\ell}(t))$ is an ℓ-dimensional vector, the functions $Y(t,h_k)$, $\varphi(t)$ and $e_j(t)$ are themselves ℓ-dimensional. Consequently the applicable system - (4), (7) or (11) - must be solved ℓ times, once for each component of the vectors. Also a complete set of interpolation polynomials P_{M-j} must be computed for each component. We again wish to point out the potential for using parallel processing computers in this case.

There are cases where y is ℓ-dimensional and not all the above work is required. Specifically, suppose

the L-dimens

order differe

order differe

one is intere

equation and

If th

need utilize

of the soluti

on $[0,b]$ wh

the problem (

equation on

as the initia

may be repeat

In th

solved ℓ ti

at h accura

polynomials

components of

the ℓ-dimensional problem resulted from reducing an ℓ^{th} order differential equation to a system of ℓ first order differential equations. Furthermore, suppose that one is interested only in the solution to the original equation and not in its various derivatives.

If the solution is desired on $[0,h]$ then one need utilize the pullback technique on the first component of the solution vector only. If the solution is desired on $[0,b]$ where $b>h$ the standard procedure is to solve the problem (1) on $[0,h]$, then solve the same differential equation on $[h,h+\bar{h}]$ using the computed solution at h as the initial condition for the new problem. This procedure may be repeated as many times as necessary to reach b.

In this latter case, the system (4) must be solved ℓ times since we need to know all the derivatives at h accurately. However the other systems and the polynomials P_{M-j} need to be computed only for the first components of the vectors.

Section 4. D

In or

P_{M-j}, of the

determine e'_M

is given in t

carried out f

In or

results, we w

are for the c

plifies the a

valid for, and

ensional case,

Denote

at the theoret

$$J(t) =$$

and define sym

$a \leq t \leq b$, by

$$f^{(k)}(t, \varphi(t))$$

for $k=2,3,\ldots$

$$f^{(k)}(t$$

vative of $f(t$

In the case wh

Section 4. Determination of $e_m'(a)$

In order to construct the Hermite polynomials, P_{M-j}, of the previous section it is necessary to first determine $e_{M-j}'(a)$. The method for computing $e_m'(a)$ is given in this section. The actual computations are carried out for $m=1,\ldots,4$ and examples are given.

In order to make effecient use of Gragg's [11,12] results, we will follow his notation. The examples given are for the case when y is 1-dimensional. This simplifies the actual computations. However, the theory is valid for, and will be presented for, the general ℓ-dimensional case, $y=(y^1,\ldots,y^\ell)$.

Denote by J the Jacobian matrix of f, evaluated at the theoretical solution φ,

$$J(t) = \frac{\partial f(t,\varphi(t))}{\partial y} \qquad a \le t \le b$$

and define symmetric k-linear operators $f^{(k)}(t,\varphi(t))$, $a \le t \le b$, by

$$f^{(k)}(t,\varphi(t))y_1\cdots y_k = \sum_{\ell_1=1}^{\ell} \cdots \sum_{\ell_k=1}^{\ell} \frac{\partial^k f(t,\varphi(t))}{\partial y^{\ell_1}\cdots\partial y^{\ell_k}} y_1^{\ell_1}\cdots y_k^{\ell_k}$$

for $k=2,3,\ldots$.

$f^{(k)}(t,\varphi(t))y_1\cdots y_k$ is the k^{th} Fréchet derivative of $f(t,\varphi(t))$ operating on the vectors $y_1\cdots y_k$. In the case where y is 1-dimensional, $J(t)$ is the

first partial with respect to y of f and $f^{(k)}(t,\varphi(t))y_1\cdots y_k = \dfrac{\partial^{(k)} f(t,\varphi(t))}{\partial y^k} y_1\cdots y_k$; that is, $f^{(k)}(t,\varphi(t))y_1\cdots y_k$ is the k^{th} partial of f with respect to y multiplied by $y_1,\ldots,y_k$. The y_j's are functions of t on which $f^{(k)}(t,\varphi(t))$ operates.

Let

$$e_0(t) \equiv \varphi(t) \tag{18a}$$

and, for $m=1,2,\ldots$, let $e_m(t)$ satisfy

$$\begin{aligned} e_m'(t) &= J(t)e_m(t) + a_m(t) + b_m(t) \\ e_m(a) &= 0 \qquad a \le t \le b , \end{aligned} \tag{18b}$$

where

$$a_m(t) = -\sum_{k=1}^{m} \alpha_k e_{m-k}^{(qk+1)}(t) \tag{18c}$$

and

$$\sum_{m=1}^{\infty} b_m(t) z^m \equiv \sum_{k=2}^{\infty} \frac{1}{k!} f^{(k)}(t,\varphi(t)) \Big(\sum_{m=1}^{\infty} e_m(t) z^m \Big)^k . \tag{18d}$$

The integer q and the constants α_k are determined from a generating function

$$A(z) = \sum_{k=0}^{\infty} \alpha_k z^{qk} \tag{18e}$$

Gragg [11, 12] has shown that both Euler's rule and the trapezoid rule have asymptotic error expansions of the form (2) with the coefficient functions determined by

(18). For Euler's rule the generating function is

$$(19) \qquad A(z) = \sum_{k=0}^{\infty} \frac{1}{(k+1)!} z^k \equiv \frac{e^z-1}{z} .$$

For the trapezoid rule

$$(20) \qquad A(z) = \frac{2}{z} \tanh\left(\frac{z}{2}\right)$$

$$= 1 - \frac{1}{3}\left(\frac{z}{2}\right)^2 + \frac{2}{15}\left(\frac{z}{2}\right)^4 - \frac{17}{315}\left(\frac{z}{2}\right)^6 + \frac{62}{2835}\left(\frac{z}{2}\right)^8$$

$$- \ldots + \frac{(-1)^{n+1} 2^{2n} (2^{2n}-1)}{(2n)!} B_n \left(\frac{z}{2}\right)^{2n-1} \pm \ldots$$

where B_n is the n^{th} Bernoulli number.

From (18a) $e_0(t) \equiv \varphi(t)$ and from (1) $\varphi'(t) = f(t,\varphi(t))$, $\varphi(a) = \alpha$; therefore we can find higher order derivatives of $e_0(t)$ at $t=a$ by successively computing total derivatives of f and evaluating them at $t=a$:

$$(21) \qquad e_0^{(p)}(a) \equiv \varphi^{(p)}(a) = \frac{d^{p-1} f(t,\varphi(t))}{dt^{p-1}}\Big|_{t=a}, \quad p=1,2,\ldots .$$

From (18b) $e_1'(t) = J(t)e_1(t) + a_1(t) + b_1(t)$ and from (18d) $b_1(t) \equiv 0$. Using (18c), we find $a_1(t) = -\alpha_1 e_0^{(qk+1)}(t)$. Thus

$$(22) \qquad e_1^{(p)}(a) = \frac{d^{p-1}[J(t)e_1(t)]}{dt^{p-1}}\Big|_{t=a} - \alpha_1 e_0^{(qk+p)}(a).$$

The second term on the right of (22) is known from (21) and since $J(t)$ is known, the first term on the right

can be computed because it involves only lower order derivatives of $e_1(t)$.

Proceeding inductively we assume that all derivatives of $e_0, e_1, \cdots, e_{j-1}$ at the point $t=a$ can be evaluated, then from (18b)

$$e_j'(t) = J(t)e_j(t) + a_j(t) + b_j(t)$$

so that

$$e_j^{(p)}(a) = \frac{d^{p-1}[J(t)e_j(t)]}{dt^{p-1}}\Big|_{t=a} + a_j^{(p-1)}(t)\Big|_{t=a} + b_j^{(p-1)}(t)\Big|_{t=a} . \tag{23}$$

Let us now consider each term on the right of (23) in turn.

The derivatives of $e_j(t)$ appearing in $\frac{d^{p-1}[J(t)e_j(t)]}{dt^{p-1}}$ are all of order less than p and can be evaluated at $t=a$ successively from the lowest to the highest. Also $J(t)$ and all its derivatives can be computed. Thus, the first term on the right of (23) is known.

From (18c) we see that $a_j(t)$ depends on various derivatives of the error functions $e_0, e_1, \ldots, e_{j-1}$ all of which are known at the point $t=a$ by our inductive hypothesis. Consequently $a_j^{(p-1)}(t)\Big|_{t=a}$ is known.

Finally, $b_j(t)$ can be determined by collecting the proper coefficients of z^j on the right hand side of (18d). $b_j(t)$ will depend on various Fréchet derivatives of f operating on various error functions. The Fréchet derivatives can be computed. The error functions appearing as arguments of the Fréchet derivatives are from the collection $e_0, e_1, \ldots e_{j-1}$. This is easily seen by observing that the outer sum on the right hand side of (18d) begins with $k=2$, which precludes the possibility of $e_j(t)$ appearing as an argument of a Fréchet derivative in the expression for $b_j(t)$.

Once $b_j(t)$ is determined it is necessary to find its $(p-1)$st derivative. The Fréchet derivatives can be differentiated with respect to t and evaluated at $t=a$ and the derivatives of the error functions are known at $t=a$ by our inductive hypothesis. Thus, $b_j^{(p-1)}(t)\big|_{t=a}$ can be computed and equation (23) is valid.

Actually, equation (23) can be expressed entirely in terms of $e_0(t)$, $J(t)$, the Fréchet derivatives and various derivatives of these functions evaluated at $t=a$. This will be done later for the case $q=1$.

Theoretically, at least, we can obtain $e_m'(a)$ for any $m=0,1,\ldots$. Computationally, the larger m is, the more complicated the expression to be evaluated. As we

shall see, the computations of $e_1'(a)$, $e_2'(a)$ and $e_3'(a)$ are fairly easy to perform for $q=1$ or $q=2$. In addition, $e_4'(a)$ is reasonable.

We consider the cases $q=1$ and $q=2$ separately. In what follows we will use a superscript in parentheses to denote differentiation with respect to t. The sole exception to this notation will be that $f^{(k)}$ will continue to denote the k^{th} Fréchet derivative of f.

Assuming the validity of (18), with $q=1$, (18c) and (18e) become

(18c)' $$a_m(t) = -\sum_{k=1}^{m} \alpha_k e_{m-k}^{(k+1)}(t)$$

and

(18e)' $$A(z) = \sum_{k=0}^{\infty} \alpha_k z^k$$

respectively.

From (18b), with $t=a$, we have $e_m^{(1)}(a) = a_m(a) + b_m(a)$. However, since $e_m(a) = 0$, $\forall m$, we can conclude from (18d) that $b_m(a) = 0$, $\forall m$. Thus,

(24) $$e_m^{(1)}(a) = a_m(a)$$

For $m=1$ $a_1(t) = -\alpha_1 e_0^{(2)}(t)$, so that

(25) $$a_1^{(p)}(a) = -\alpha_1 e_0^{(p+2)}(a) \quad \text{for any } p.$$

In particular, we have

$$e_1^{(1)}(a) = -\alpha_1 e_0^{(2)}(a). \tag{26}$$

In order to determine $e_m^{(1)}(a)$ for $m=2,3,4$ we will need the second through fourth derivatives of $e_1(t)$ evaluated at $t=a$. To obtain these we proceed as follows: differentiating (18b) with $m=1$, we obtain

$$e_1^{(2)}(t) = \frac{d[J(t)e_1(t)]}{dt} + a_1^{(1)}(t) + b_1^{(1)}(t).$$

Now $b_1(t) \equiv 0$ and $a_1(t) = -\alpha_1 e_0^{(2)}(t)$. Therefore

$$e_1^{(2)}(t) = \frac{d[J(t)e_1(t)]}{dt} - \alpha_1 e_0^{(3)}(t) \tag{27a}$$

$$= J^{(1)}(t)e_1(t) + J(t)e_1^{(1)}(t) - A_1 e_0^{(3)}(t).$$

Since $e_1(a) = 0$ and $e_1^{(1)}(a) = -\alpha_1 e_0^{(2)}(a)$ we have

$$e_1^{(2)}(a) = -\alpha_1 J(a) e_0^{(2)}(a) - \alpha_1 e_0^{(3)}(a) \tag{27b}$$

$$= -\alpha_1 [e_0^{(3)}(a) + J(a) e_0^{(2)}(a)].$$

Differentiating (27a) we have

$$e_1^{(3)}(t) = \frac{d^2[J(t)e_1(t)]}{dt^2} - \alpha_1 e_0^{(4)}(t) \tag{28a}$$

$$= J^{(2)}(t)e_1(t) + 2J^{(1)}(t)e_1^{(1)}(t) + J(t)e_1^{(2)}(t)$$

$$- \alpha_1 e_0^{(4)}(t)$$

by Liebnitz's rule. Using (26), (27b) and $e_1(a) = 0$, we can evaluate (28a) for $t=a$ and obtain

$$(28b)\quad e_1^{(3)}(a) = -2J^{(1)}(a)\alpha_1 e_0^{(2)}(a) - J(a)\alpha_1[e_0^{(3)}(a)+J(a)e_0^{(2)}(a)]$$

$$-\alpha_1 e_0^{(4)}(a)$$

$$= -\alpha_1\{[2J^{(1)}(a)+(J(a))^2]e_0^{(2)}(a)+J(a)e_0^{(3)}(a)+e_0^{(4)}(a)\}.$$

Differentiating (28a) we have

$$(29a)\quad e_1(t) = \frac{d^3[J(t)e_1(t)]}{dt^3} - \alpha_1 e_0^{(5)}(t)$$

$$= J^{(3)}(t)e_1(t)+3J^{(2)}(t)e_1^{(1)}(t)+3J^{(1)}(t)e_1^{(2)}(t)$$

$$+J(t)e_1^{(3)}(t)-\alpha_1 e_0^{(5)}(t).$$

Using (26), (27b), (28b) and $e_1(a) = 0$ we have

$$(29b)\quad e_1^{(4)}(a) = -3\alpha_1 J^{(2)}(a)e_0^{(2)}(a) - 3\alpha_1 J^{(1)}(a)[e_0^{(3)}(a)+J(a)e_0^{(2)}(a)]$$

$$-\alpha_1 J(a)\{[2J^{(1)}(a)+(J(a))^2]e_0^{(2)}(a)+J(a)e_0^{(3)}(a)$$

$$+e_0^{(4)}(a)\}-\alpha_1 e_0^{(5)}(a)$$

$$= -\alpha_1\{[3J^{(2)}(a)+5J(a)J^{(1)}(a)+(J(a))^3]e_0^{(2)}(a) +$$

$$[3J^{(1)}(a)+(J(a))^2]e_0^{(3)}(a)+J(a)e_0^{(4)}(a)+e_0^{(5)}(a)\}.$$

Note that all expressions for the derivatives of e_1 evaluated at $t=a$ involve only the derivatives of e_0 and the derivatives of J with respect to t. The derivatives of J with respect to t are a straightforward calculation and the derivatives of e_0 can be obtained by successively differentiating and evaluating

the original equation (1).

For $m=2$; $a_2(t) = -\alpha_1 e_1^{(2)}(t) - \alpha_2 e_0^{(3)}(t)$ so that

(30) $a_2^{(p)}(a) = -\alpha_1 e_1^{(p+2)}(a) - \alpha_2 e_0^{(p+3)}(t)$ for any p.

In particular using (24) and (27b) we have

$$(31)\quad e_2^{(1)}(a) = a_2(a) = -\alpha_1 e_1^{(2)}(a) - \alpha_2 e_0^{(3)}(a)$$

$$= \alpha_1^2[e_0^{(3)}(a) + J(a) e_0^{(2)}(a)] - \alpha_2 e_0^{(3)}(a)$$

$$= (\alpha_1^2 - \alpha_2) e_0^{(3)}(a) + \alpha_1^2 J(a) e_0^{(2)}(a).$$

Now $e_m^{(1)}(a)$ for $m=3,4$ will require knowledge of the second and third derivatives of $e_2(t)$ evaluated at $t=a$. Differentiating (18b) with $m=2$, we obtain

$$(32)\quad e_2^{(2)}(t) = \frac{d[J(t)e_2(t)]}{dt} + a_2^{(1)}(t) + b_2^{(1)}(t).$$

From (18d) we have

$$(33)\quad b_2(t) = \frac{1}{2} f^{(2)}(t, \varphi(t)) e_1(t) e_1(t)$$

implying that

$$b_2^{(1)}(t) = \frac{1}{2} \frac{df^{(2)}(t, \varphi(t))}{dt} e_1(t) e_1(t) + f^{(2)}(t, \varphi(t)) e_1(t) e_1'(t).$$

Note that $f^{(2)}(t, \varphi(t)) \equiv f_{yy}(t, \varphi(t))$ for the one dimensional problem. Since $e_1(a) = 0$ we can conclude that $b_2^{(1)}(a) = 0$.

Using (30), (31) and $e_2(a) = 0$ we have

$$e_2^{(2)}(a) = J(a)e_2^{(1)}(a) + a_2^{(1)}(a)$$

$$= -\alpha_1 J(a)e_1^{(2)} - \alpha_2 J(a)e_0^{(3)}(a) - \alpha_1 e_1^{(3)}(a) - \alpha_2 e_0^{(4)}(a)$$

$$= -\alpha_1(e_1^{(3)}(a) + J(a)e_1^{(2)}(a)) - \alpha_2(e_0^{(4)}(a) + J(a)e_0^{(3)}(a))$$

Hence, $e_2^{(2)}(a)$ can be expressed in terms of derivatives of $e_0(a)$ by using (27b) and (28b),

$$e_2^{(2)}(a) = \alpha_1^2\{[2J^{(1)}(a) + (J(a))^2]e_0^{(2)}(a) + J(a)e_0^{(3)}(a) + e_0^{(4)}(a)$$

$$+ (J(a))^2 e_0^{(2)}(a) + J(a)e_0^{(3)}(a)\} - \alpha_2(e_0^{(4)}(a) + J(a)e_0^{(3)}(a)).$$

Collecting terms involving the various derivatives of $e_0(a)$, we have

$$(34)\quad e_2^{(2)}(a) = (\alpha_1^2 - \alpha_2)e_0^{(4)}(a) + (2\alpha_1^2 - \alpha_2)J(a)e_0^{(3)}(a)$$

$$+ 2\alpha_1^2[J^{(1)}(a) + (J(a))^2]e_0^{(2)}(a).$$

Differentiating (32) again, we have

$$e_2^{(3)}(t) = \frac{d^2[J(t)e_2(t)]}{dt^2} + a_2^{(2)}(t) + b_2^{(2)}(t).$$

By differentiating (33) twice and using $e_1(a) = 0$, we find that $b_2^{(2)}(a) = f^{(2)}(a, \varphi(a))e_1^{(1)}(a)e_1^{(1)}(a)$. From (30), $a_2^{(2)}(a) = -\alpha_1 e_1^{(4)}(a) - \alpha_2 e_0^{(5)}(a)$ and since $e_2(a) = 0$ we have

$$e_2^{(3)}(a) = 2J^{(1)}(a)e_2^{(1)}(a)+J(a)e_2^{(2)}(a)-\alpha_1 e_1^{(4)}(a)-\alpha_2 e_0^{(5)}(a)$$
$$+f^{(2)}(a,\varphi(a))e_1^{(1)}(a)e_1^{(1)}(a).$$

Substituting for $e_2^{(1)}(a)$, $e_2^{(2)}(a)$, $e_1^{(4)}(a)$ and $e_1^{(1)}(a)$ from equations (31), (34), (29b) and (26) respectively, we have

$$\begin{aligned} e_2^{(3)}(a) = {} & 2(\alpha_1^2-\alpha_2)J^{(1)}(a)e_0^{(3)}(a)+2\alpha_1^{(2)}J(a)J^{(1)}(a)e_0^{(2)}(a) \\ & +(\alpha_1^2-\alpha_2)J(a)e_0^{(4)}(a)+(2\alpha_1^{(2)}-\alpha_2)(J(a))^2e_0^{(3)}(a) \\ & +(2\alpha_1^2[J(a)J^{(1)}(a)+(J(a))^3]e_0^{(2)}(a) \\ & +\alpha_1^2[3J^{(2)}(a)+5J(a)J^{(1)}(a)+(J(a))^3]e_0^{(2)}(a) \\ & +\alpha_1^2[3J^{(1)}(a)+(J(a))^2]e_0^{(3)}(a)+\alpha_1^2J(a)e_0^{(4)}(a) \\ & +\alpha_1^2 e_0^{(5)}(a)-\alpha_2 e_0^{(5)}(a) \\ & +\alpha_1^2 f^{(2)}(a,\varphi(a))e_0^{(2)}(a)e_0^{(2)}(a). \end{aligned}$$

Collecting on the derivatives of $e_0(a)$, we have

$$\begin{aligned} (35)\quad e_2^{(3)}(a) = {} & (\alpha_1^2-\alpha_2)e_0^{(5)}(a)+(2\alpha_1^2-\alpha_2)J(a)e_0^{(4)}(a) \\ & +[(5\alpha_1^2-2\alpha_2)J^{(1)}(a)+(3\alpha_1^2-\alpha_2)(J(a))^2]e_0^{(3)}(a) \\ & +\alpha_1^2[3J^{(2)}(a)+9J(a)J^{(1)}(a)+3(J(a))^3]e_0^{(2)}(a) \\ & +\alpha_1^2 f^{(2)}(a,\varphi(a))e_0^{(2)}(a)e_0^{(2)}(a). \end{aligned}$$

With $m=3$ from (18c)' we have

$a_3(t) = -\alpha_1 e_2^{(2)}(t) - \alpha_2 e_1^{(3)}(t) - \alpha_3 e_0^{(4)}(t)$ so that

(36) $$a_3^{(p)}(a) = -\alpha_1 e_2^{(p+2)}(a) - \alpha_2 e_1^{(p+3)}(a) - \alpha_3 e_0^{(p+4)}(a)$$

for any p. From (24) we have

$$e_3^{(1)}(a) = a_3(a) = -\alpha_1 e_2^{(2)}(a) - \alpha_2 e_1^{(3)}(a) - \alpha_3 e_0^{(4)}(a).$$

Substituting (34) and (28b) for $e_2^{(2)}(a)$ and $e_1^{(3)}(a)$ respectively, we have

$$\begin{aligned} e_3^{(1)}(a) = {} & -\alpha_1(\alpha_1^2 - \alpha_2) e_0^{(4)}(a) - \alpha_1(2\alpha_1^2 - \alpha_2) J(a) e_0^{(3)}(a) \\ & - 2\alpha_1^3 [J^{(1)}(a) + (J(a))^2] e_0^{(2)}(a) \\ & + \alpha_1\alpha_2 [2J^{(1)}(a) + (J(a))^2] e_0^{(2)}(a) \\ & + \alpha_1\alpha_2 J(a) e_0^{(3)}(a) + \alpha_1\alpha_2 e_0^{(4)}(a) - \alpha_3 e_0^{(4)}(a). \end{aligned}$$

Combining like derivates of $e_0(a)$, this becomes

(37) $$\begin{aligned} e_3^{(1)}(a) = {} & (2\alpha_1\alpha_2 - \alpha_1^3 - \alpha_3) e_0^{(4)}(a) + 2(\alpha_1\alpha_2 - \alpha_1^3) J(a) e_0^{(3)}(a) \\ & + [(\alpha_1\alpha_2 - 2\alpha_1^3)(J(a))^2 + 2(\alpha_1\alpha_2 - \alpha_1^3) J^{(1)}(a)] e_0^{(2)}(a). \end{aligned}$$

Differentiating (18b) with m=3, we have

$$e_3^{(2)}(t) = \frac{d[J(t) e_3(t)]}{dt} + a_3^{(1)}(t) + b_3^{(1)}(t).$$

From (18d), $b_3(t) = \frac{1}{6} f^{(3)}(t, \varphi(t)) e_1(t) e_1(t) e_1(t) +$ $f^{(2)}(t, \varphi(t)) e_1(t) e_2(t)$ so that $b_3^{(1)}(a) = 0$ as

$e_1(a) = e_2(a) = 0$. Substituting for $a_3^{(1)}(t)$ and using $e_3(a) = 0$, we have

$$e_3^{(2)}(a) = J(a)e_3^{(1)}(a)-\alpha_1 e_2^{(3)}(a)-\alpha_2 e_1^{(4)}(a)-\alpha_3 e_0^{(5)}(a).$$

Substituting (37), (35) and (29b) into this equation yields

$$\begin{aligned} e_3^{(2)}(a) = {} & (2\alpha_1\alpha_2-\alpha_1^3-\alpha_3)J(a)e_0^{(4)}(a)+2(\alpha_1\alpha_2-\alpha_1^3)(J(a))^2 e_0^{(3)}(a) \\ & +[(\alpha_1\alpha_2-2\alpha_1^3)(J(a))^3+2(\alpha_1\alpha_2-\alpha_1^3)J(a)J^{(1)}(a)]e_0^{(2)}(a) \\ & +(\alpha_1\alpha_2-\alpha_1^3)e_0^{(5)}(a)+(\alpha_1\alpha_2-2\alpha_1^{(3)})J(a)e_0^{(4)}(a) \\ & +[(2\alpha_1\alpha_2-5\alpha_1^3)J^{(1)}(a)+(\alpha_1\alpha_2[3\alpha_1^3)(J(a))^2]e_0^{(3)}(a) \\ & -\alpha_1^3[3J^{(2)}(a)+9J(a)J^{(1)}(a)+3(J(a)]^3 e_0^{(2)}(a) \\ & -\alpha_1^3 f^{(2)}(a,\varphi(a))e_0^{(2)}(a)e_0^{(2)}(a)+\alpha_1\alpha_2 e_0^{(5)}(a) \\ & +\alpha_1\alpha_2 J(a)e_0^{(4)}(a)+\alpha_1\alpha_2[3J^{(1)}(a)+(J(a))^2]e_0^{(3)}(a) \\ & +\alpha_1\alpha_2[3J^{(2)}(a)+5J(a)J^{(1)}(a)+(J(a))^3]e_0^{(2)}(a)-\alpha_3 e_0^{(5)}(a). \end{aligned}$$

This can be rewritten as

$$\begin{aligned} (38)\quad e_3^{(2)}(a) = {} & 2(\alpha_1\alpha_2-\alpha_1^3-\alpha_3)e_0^{(5)}(a)+(4\alpha_1\alpha_2-3\alpha_1^3-\alpha_3)J(a)e_0^{(4)}(a) \\ & +[5(\alpha_1\alpha_2-\alpha_1^3)J^{(1)}(a)+(4\alpha_1\alpha_2-5\alpha_1^3)(J(a))^2]e_0^{(3)}(a) \\ & +[3(\alpha_1\alpha_2-\alpha_1^3)J^{(3)}(a)+(7\alpha_1\alpha_2-11\alpha_1^3)J(a)J^{(1)}(a) + \\ & \quad (2\alpha_1\alpha_2-5\alpha_1^3)(J(a))^3]e_0^{(2)}(a) \\ & -\alpha_1^3 f^{(2)}(a,\varphi(a))e_0^{(2)}(a)e_0^{(2)}(a). \end{aligned}$$

Finally for m=4 we have

$$e_4^{(1)}(a) = a_4(a) = -\alpha_1 e_3^{(2)}(a) - \alpha_2 e_2^{(3)}(a) - \alpha_3 e_1^{(4)}(a) - \alpha_4 e_0^{(5)}(a).$$

Using (38), (35) and (29b), this can be written as

$$\begin{aligned} e_4^{(1)}(a) = {} & 2(\alpha_1^4+\alpha_1\alpha_3-\alpha_1^2\alpha_2)e_0^{(5)}(a)+(3\alpha_1^4+\alpha_1\alpha_3-4\alpha_1^2\alpha_2)J(a)e_0^{(4)}(a) \\ & +[5(\alpha_1^4-\alpha_1^2\alpha_2)J^{(1)}(a)+(5\alpha_1^4-4\alpha_1^2\alpha_2)(J(a))^2]e_0^{(3)}(a) \\ & +[3(\alpha_1^4-\alpha_1^2\alpha_2)J^{(2)}(a)+(11\alpha_1^4-7\alpha_1^2\alpha_2)J(a)J^{(1)}(a)+ \\ & \qquad (5\alpha_1^4-2\alpha_1^2\alpha_2)(J(a))^3]e_0^{(2)}(a) \\ & +\alpha_1^4 f^{(2)}(a,\varphi(a))e_0^{(2)}(a)e_0^{(2)}(a) \\ & +(\alpha_2^2-\alpha_1^2\alpha_2)e_0^{(5)}(a)+(\alpha_2^2-2\alpha_1^2\alpha_2)J(a)e_0^{(4)}(a) \\ & +[(2\alpha_2^2-5\alpha_1^2\alpha_2)J^{(1)}(a)+(\alpha_2^2-3\alpha_1^2\alpha_2)(J(a))^2]e_0^{(3)}(a) \\ & -\alpha_1^2\alpha_2[3J^{(2)}(a)+9J(a)J^{(1)}(a)+3(J(a))^3]e_0^{(2)}(a) \\ & -\alpha_1^2\alpha_2 f^{(2)}(a,\varphi(a))e_0^{(2)}(a)e_0^{(2)}(a) \\ & +\alpha_1\alpha_3 e_0^{(5)}(a)+\alpha_1\alpha_3 J(a)e_0^{(4)}(a) \\ & +\alpha_1\alpha_3[3J^{(1)}(a)+(J(a))^2]+(J(a))e_0^{(3)}(a) \\ & +\alpha_1\alpha_3[3J^{(2)}(a)+5J(a)J^{(1)}(a)+(J(a))^3]e_0^{(2)}(a) \\ & -\alpha_4 e_0^{(5)}(a) \end{aligned}$$

Rearranging terms, we get

(39) $e_4^{(1)}$ (a

To su

can be used t

$e_4^{(1)}$(a) resp

recursive pro

Step 1 Comp

Step 2 Comp

Step 3 $e_1^{(1)}$

$e_1^{(2)}$

$e_1^{(3)}$

$e_1^{(4)}$

.

.

.

.

$$(39)\quad e_4^{(1)}(a) = (2\alpha_1^4-3\alpha_1^2\alpha_2+3\alpha_1\alpha_3+\alpha_2^2-\alpha_4)e_0^{(5)}(a)$$

$$+(3\alpha_1^4-6\alpha_1^2\alpha^2+2\alpha_1\alpha_3+\alpha_2^2)e_0^{(4)}(a)$$

$$+[(5\alpha_1^4-10\alpha_1^2\alpha_2+3\alpha_1\alpha_3+2\alpha_2^2)J^{(1)}(a) +$$

$$(5\alpha_1^4-7\alpha_1^2\alpha_2+\alpha_1\alpha_3+\alpha_2^2)(J(a))^2]e_0^{(3)}(a)$$

$$+[3(\alpha_1^4-2\alpha_1^2\alpha_2+\alpha_1\alpha_3)J^{(2)}(a)+(11\alpha_1^4-16\alpha_1^2\alpha_2 +$$

$$5\alpha_1\alpha_3)J(a)J^{(1)}(a)+(5\alpha_1^4-5\alpha_1^2\alpha_3+\alpha_1\alpha_3)(J(a))^3]e_0^{(2)}(a)$$

$$+(\alpha_1^4-\alpha_1^2\alpha_2)f^{(2)}(a,\varphi(a))e_0^{(2)}(a)e_0^{(2)}(a).$$

To summarize, equations (26), (31), (37) and (39) can be used to determine $e_1^{(1)}(a)$, $e_2^{(1)}(a)$, $e_3^{(1)}(a)$ and $e_4^{(1)}(a)$ respectively. A computationally more effective recursive procedure for $q=1$ is given in Table 3 below.

TABLE 3

Step 1 Compute $e_0^{(1)}(a)$, $e_0^{(2)}(a)$, $e_0^{(3)}(a)$, $e_0^{(4)}(a)$, $e_0^{(5)}(a)$....

Step 2 Compute $J(a)$, $J^{(1)}(a)$, $J^{(2)}(a)$, ...

Step 3 $e_1^{(1)}(a) = -\alpha_1 e_0^{(2)}(a)$

$e_1^{(2)}(a) = J(a)e_1^{(1)}(a)-\alpha_1 e_0^{(3)}(a)$

$e_1^{(3)}(a) = 2J^{(1)}(a)e_1^{(1)}(a)+J(a)e_1^{(2)}(a)-\alpha_1 e_0^{(4)}(a)$

$e_1^{(4)}(a) = 3J^{(2)}(a)e_1^{(1)}(a)+3J^{(1)}(a)e_1^{(2)}(a)+J(a)e_1^{(3)}(a)$

$\vdots \qquad\qquad -\alpha_1 e_0^{(5)}(a)$

$\vdots$

Step 4 $e_2^{(}$

$e_2^{(}$

$e_2^{(}$

Step 5 $e_3^{(}$

$e_3^{(}$

Step 6 $e_4^{(}$

Tab

computation

satisfies (

continued.

at t=a is

at each ste

to the tabl

simply the

1 and 2. an

as they bec

TABLE 3 continued

Step 4
$$e_2^{(1)}(a) = -\alpha_1 e_1^{(2)}(a) - \alpha_2 e_0^{(3)}(a)$$
$$e_2^{(2)}(a) = J(a)e_2^{(1)}(a) - \alpha_1 e_1^{(3)}(a) - \alpha_2 e_0^{(4)}(a)$$
$$e_2^{(3)}(a) = 2J^{(1)}(a)e_2^{(1)}(a) + J(a)e_2^{(2)}(a) - \alpha_1 e_1^{(4)}(a)$$
$$\vdots \qquad -\alpha_2 e_0^{(5)}(a) + f^{(2)}(a,\varphi(a))e_1^{(1)}(a)e_1^{(1)}(a)$$

Step 5
$$e_3^{(1)}(a) = -\alpha_1 e_2^{(2)}(a) - \alpha_2 e_1^{(3)}(a) - \alpha_3 e_0^{(4)}(a)$$
$$e_3^{(2)}(a) = J(a)e_3^{(1)}(a) - \alpha_1 e_2^{(3)}(a) - \alpha_2 e_1^{(4)}(a) - \alpha_3 e_0^{(5)}(a)$$
$$\vdots$$

Step 6
$$e_4^{(1)}(a) = -\alpha_1 e_3^{(2)}(a) - \alpha_2 e_2^{(3)}(a) - \alpha_3 e_1^{(4)}(a) - \alpha_4 e_0^{(5)}(a)$$
$$\vdots$$

Table 3 lists the equations necessary for the computation of $e_1^{(1)}(a),\ldots,e_4^{(1)}(a)$ for a method which satisfies (18) with $q=1$. Of course, Table 3 can be continued. Each new error function, whose first derivative at $t=a$ is to be computed, requires one additional computation at each step of the table and the addition of one more step to the table. The only computations in Table 3 which are not simply the result of plugging in formulae are those in steps 1 and 2. and the computation of various Fréchet derivatives as they become necessary.

To i

$e_2^{(1)}(a)$, $e_3^{(1}$

in example 1

Example 2:

the solution

Y

From equatio

rule. In th

Step 1: $e_0^{(1}$

$e_0^{(2}$

$e_0^{(3}$

$e_0^{(4}$

$e_0^{(5}$

Step 2:

$J($

$J($

To illustrate the use of Table 3 we compute $e_1^{(1)}(a)$, $e_2^{(1)}(a)$, $e_3^{(1)}(a)$ and $e_4^{(1)}(a)$ for the differential equation in example 1 using a numerical method for which $q=1$.

Example 2: Suppose we are using Euler's rule to compute the solution to

$$y'=y^2; \quad y(0)=.2; \quad 0\leq t\leq 1.$$

From equation (19) we have that $\alpha_k = \frac{1}{(k+1)!}$ for Euler's rule. In this case Table 3 becomes

Step 1: $e_0^{(1)}(0) = (y(0))^2 = .040000$

$$e_0^{(2)}(2) = 2y(0)y^{(1)}(0) = .016000$$

$$e_0^{(3)}(0) = 2[y(0)y^{(2)}(0)+(y^{(1)}(0))^2] = .009600$$

$$e_0^{(4)}(0) = 2[y(0)y^{(3)}(0)+3y^{(1)}(0)y^{(2)}(0)] = .007680$$

$$e_0^{(5)}(0) = 2[y(0)y^{(4)}(0)+4y^{(1)}(0)y^{(3)}(0)+3(Y^{(2)}(0))^2]$$

$$= .007680$$

Step 2: $J(0) = 2y(0) = .400000$

$$J^{(1)}(0) = 2y^{(1)}(0) = .080000$$

$$J^{(2)}(1) = 2y^{(2)}(0) = .032000$$

Step 3: $e_1^{(1}$

$e_1^{(2}$

$e_1^{(3}$

$e_1^{(4}$

Step 4: $e_2^{(}$

$e_2^{(}$

$e_2^{(}$

Step 5: $e_3^{(}$

$e_3^{(}$

Step 6: $e_4^{(}$

It

Steps 1 and

Also note t

functions a

of the erro

Tur

(18) is val

Step 3: $e_1^{(1)}(0) = -.008000$

$e_1^{(2)}(0) = -.008000$

$e_1^{(3)}(0) = -.008520$

$e_1^{(4)}(0) = -.009936$

Step 4: $e_2^{(1)}(0) = .002400$

$e_2^{(2)}(0) = .001810$

$e_2^{(3)}(0) = .006076$ since $f^{(2)}(a,\varphi(a)) = 2.$

Step 5: $e_3^{(1)}(0) = -.007950$

$e_3^{(2)}(0) = -.004882$

Step 6: $e_4^{(1)}(0) = .0017780$

It should be emphasized that the computations in Steps 1 and 2 are performed by the user not by the machine. Also note that the signs on the derivatives of the error functions alternate in precisely the same manner as the signs of the errors for Euler's rule.

Turning our attention to the case $q=2$, we assume (18) is valid with $q=2$ so that (18c) and (18e) become

(18c)"

and

(18e)"

In

analogous

about the

$b_1(t$

$b_2(t$

(40) $b_3(t$

$b_4(t$

Si

$e_1^{(1)}(a)\ldots$

necessary

have

since $e_4(a$

$e_4^{(1)}(a$

(18c)" $$a_m(t) = -\sum_{k=1}^{m} \alpha_k \, e_{m-k}^{(2k+1)}(t)$$

and

(18e)" $$A(z) = \sum_{k=0}^{\infty} \alpha_k \, z^{2k} \ .$$

In order to construct a table for the case $q=2$ analogous to Table 3 we will need to collect more information about the functions $b_m(t)$ for $m=1,\ldots,4$. From (18d) we have

$$b_1(t) \equiv 0$$

$$b_2(t) = \tfrac{1}{2} f^{(2)}(t,\varphi(t)) e_1(t) e_1(t)$$

(40) $$b_3(t) = \tfrac{1}{6} f^{(3)}(t,\varphi(t)) e_1(t) e_1(t) e_1(t) + f^{(2)}(t,\varphi(t)) e_1(t) e_2(t)$$

$$b_4(t) = \tfrac{1}{24} f^{(4)}(t,\varphi(t)) e_1(t) e_1(t) e_1(t) e_1(t) + \tfrac{1}{2} f^{(3)}(t,\varphi(t))$$
$$e_1(t) e_1(t) e_2(t) + f^{(2)}(t,\varphi(t)) e_1(t) e_3(t)$$
$$+ \tfrac{1}{2} f^{(2)}(t,\varphi(t)) e_2(t) e_2(t) .$$

Since our goal is to produce a table for evaluating $e_1^{(1)}(a),\ldots,e_4^{(1)}(a)$ we examine what information will be necessary to determine $e_4^{(1)}(a)$. From (18b) with $m=4$ we have

$$e_4^{(1)}(a) = a_4(a)$$

since $e_4(a) = b_4(a) = 0$. From (18c)" we can conclude

$$e_4^{(1)}(a) = -\alpha_1 e_3^{(3)}(a) - \alpha_2 e_2^{(5)}(a) - \alpha_3 e_1^{(7)}(a) - \alpha_4 e_0^{(9)}(a) .$$

In genera

$$e_m^p(a)$$

Thus in

to deter

$e_1^{(7)}(a)$

given be

function

zero. F

therefor

$$b_2^{(1)}$$

$$b_2^{(2)}$$

$$b_2^{(3)}$$

$$b_2^{(4)}$$

Using t

In general, by (18b) we see that

$$e_m^p(a) = \frac{d^{p-1}[J(t)e_m(t)]}{dt^{p-1}} + a_m^{(p-1)}(t) + b_m^{(p-1)}(t); \quad m=1,\ldots,4.$$

Thus in order to determine $e_3^{(3)}(a)$ we need to know $b_3^{(2)}(a)$; to determine $e_2^{(5)}(a)$ we need $b_2^{(4)}(a)$; and to determine $e_1^{(7)}(a)$ we need $b_1^{(6)}(a)$. The necessary computations are given below. To simplify the notation, the arguments of the functions are suppressed.

Since $b_1(t) \equiv 0$ all derivatives of $b_1(t)$ are zero. From (40), $b_2(t) = \frac{1}{2}f^{(2)}(t,\varphi(t))e_1(t)e_1(t)$ and therefore;

$$b_2^{(1)}(t) = \frac{1}{2}\frac{d(f^{(2)})}{dt}e_1e_1 + f^{(2)}e_1e_1^{(1)}$$

$$b_2^{(2)}(t) = \frac{1}{2}\frac{d^2f^{(2)}}{dt^2}e_1e_1 + 2(f^{(2)})^{(1)}e_1e_1^{(1)} + f^{(2)}e_1e_1^{(2)} + f^{(2)}e_1^{(1)}e_1^{(1)}$$

$$b_2^{(3)}(t) = \frac{1}{2}\frac{d^3f^{(2)}}{dt^3}e_1e_1 + 3(f^{(2)})^{(2)}e_1e_1^{(1)} + 3\frac{d(f^{(2)})}{dt}e_1e_1^{(2)} + 3\frac{d(f^{(2)})}{dt}e_1^{(1)}e_1^{(1)} + 3f^{(2)}e_1^{(1)}e_1^{(2)} + f^{(2)}e_1e_1^{(3)}$$

$$b_2^{(4)}(t) = \frac{1}{2}\frac{d^4f^{(2)}}{dt^4}e_1e_1 + 4\frac{d^3f^{(2)}}{dt^3}e_1e_1^{(1)} + 6\frac{d^2f^{(2)}}{dt^2}e_1^{(1)}e_1^{(1)} + 6\frac{d^2f^{(2)}}{dt^2}e_1e_1^{(2)} + 4\frac{d(f^{(2)})}{dt}e_1e_1^{(3)} + 12\frac{d(f^{(2)})}{dt}e_1^{(1)}e_1^{(2)} + 3f^{(2)}e_1^{(2)}e_1^{(2)} + 4f^{(2)}e_1^{(1)}e_1^{(3)} + f^{(2)}e_1e_1^{(4)}.$$

Using the fact that $e_1(a) = 0$ we can obtain

$b_2^{(1)}$

$b_2^{(2)}$

(41) $b_2^{(3)}$

$b_2^{(4)}$

Us

have

$b_3^{(1)}(t) =$

$b_3^{(2)}(t) =$

Since $e_1($

$$b_2^{(1)}(a) = 0$$

$$b_2^{(2)}(a) = f^{(2)}(a,\varphi(a))e_1^{(1)}(a)e_1^{(1)}(a)$$

$$(41)\quad b_2^{(3)}(a) = 3[f^{(2)}(a,\varphi(a))e_1^{(1)}(a)e_1^{(2)}(a) + \frac{df^{(2)}(t,\varphi(t))}{dt}\Big|_{t=a} e_1^{(1)}(a)e_1^{(1)}(a)]$$

$$b_2^{(4)}(a) = 6\frac{d^2f^{(2)}(t,\varphi(t))}{dt^2}\Big|_{t=a} e_1^{(1)}(a)e_1^{(1)}(a) + 12\frac{df^{(2)}(t,\varphi(t))}{dt}\Big|_{t=a} e_1^{(1)}(a)e_1^{(2)}(a) + 3f^{(2)}(a,\varphi(a))e_1^{(2)}(a)e_1^{(2)}(a) + 4f^{(2)}(a,\varphi(a))e_1^{(1)}(a)e_1^{(a)}(a).$$

Using the definition of $b_3(t)$ given in (40), we have

$$b_3^{(1)}(t) = \frac{1}{6}\frac{df^{(3)}}{dt} e_1e_1e_1 + \frac{1}{2}f^{(3)}e_1e_1e_1^{(1)} + (f^{(2)})^{(1)}e_1e_2 + f^{(2)}e_1^{(1)}e_2 + f^{(2)}e_1e_2^{(1)}$$

$$b_3^{(2)}(t) = \frac{1}{6}\frac{d^2f^{(3)}}{dt^2} e_1e_1e_1 + \frac{df^{(3)}}{dt} e_1e_1e_1^{(1)} + f^{(3)}e_1e_1^{(1)}e_1^{(1)} + \frac{1}{2}f^{(3)}e_1e_1e_1^{(2)} + \frac{d^2f^{(2)}}{dt^2} e_1e_2 + 2\frac{df^{(2)}}{dt} e_1^{(1)}e_2 + 2\frac{df^{(2)}}{dt} e_1e_2^{(1)} + f^{(2)}e_1^{(2)}e_2 + 2f^{(2)}e_1^{(1)}e_2^{(1)} + f^{(2)}e_1e_2^{(2)}.$$

Since $e_1(a) = e_2(a) = 0$, at $t=a$ we have

$$b_3^{(}$$

(42)

$$b_3^{(}$$

W

evaluatir

Step 1:

Step 2:

Step 3:

$$
\begin{aligned}
& b_3^{(1)}(a) = 0 \\
(42) \qquad & \\
& b_3^{(2)}(a) = 2f^{(2)}(a,\varphi(a))e_1^{(1)}(a)e_2^{(1)}(a).
\end{aligned}
$$

We can now summarize the computations involved for evaluating $e_1^{(1)}(a),\ldots,e_4^{(1)}(a)$ when $q=2$.

TABLE 4

Step 1: Compute $e_0^{(1)}(a)$, $e_0^{(2)}(a),\ldots,e_0^{(8)}(a)$, $e_0^{(9)}(a),\ldots$

Step 2: Compute $J(a)$, $J^{(1)}(a),\ldots,J^{(5)}(a),\ldots$

Step 3:

$$e_1^{(1)}(a) = -\alpha_1 e_0^{(3)}(a)$$

$$e_1^{(2)}(a) = J(a)e_1^{(1)}(a)-\alpha_1 e_0^{(4)}(a)$$

$$e_1^{(3)}(a) = 2J^{(1)}(a)e_1^{(1)}(a)+J(a)e_1^{(2)}(a)-\alpha_1 e_0^{(5)}(a)$$

$$e_1^{(4)}(a) = 3J^{(2)}(a)e_1^{(1)}(a)+3J^{(1)}(a)e_1^{(2)}(a)+J(a)e_1^{(3)}(a) - \alpha_1 e_0^{(6)}(a)$$

$$e_1^{(5)}(a) = \sum_{k=0}^{3}\binom{4}{k}J^{(k)}(a)e_1^{(4-k)}(a)-\alpha_1 e_0^{(7)}(a)$$

$$e_1^{(6)}(a) = \sum_{k=0}^{4}\binom{5}{k}J^{(k)}(a)e_1^{(5-k)}(a)-\alpha_1 e_0^{(8)}(a)$$

$$e_1^{(7)}(a) = \sum_{k=0}^{5}\binom{6}{k}J^{(k)}(a)e_1^{(6-k)}(a)-\alpha_1 e_0^{(9)}(a)$$

$$\vdots$$

Step 4: $b_2^{(}$

$b_2^{(2}$

$b_2^{(3}$

$b_2^{(4}$

Step 5: $e_2^{(}$

$e_2^{(}$

e_2

e

e

TABLE 4 continued

Step 4:
$$b_2^{(1)}(a) = 0$$

$$b_2^{(2)}(a) = f^{(2)}(a,\varphi(a))e_1^{(1)}(a)e_1^{(1)}(a)$$

$$b_2^{(3)}(a) = 3[f^{(2)}(a,\varphi(a))e_1^{(1)}(a)e_1^{(2)}(a) + \frac{df^{(2)}(t,\varphi(t))}{dt}\Big|_{t=a} e_1^{(1)}(a)e_1^{(1)}(a)]$$

$$b_2^{(4)}(a) = 6\frac{d^2f^{(2)}(t,\varphi(t))}{dt^2}\Big|_{t=a} e_1^{(1)}(a)e_1^{(1)}(a) + 12\frac{df^{(2)}(t,\varphi(t))}{dt}\Big|_{t=a} e_1^{(1)}(a)e_1^{(2)}(a) + 3f^{(2)}(a,\varphi(a))e_1^{(2)}(a)e_1^{(2)}(a) + 4f^{(2)}(a,\varphi(a))e_1^{(1)}(a)e_1^{(3)}(a)$$

$$\vdots$$

Step 5:
$$e_2^{(1)}(a) = -\alpha_1 e_1^{(3)}(a) - \alpha_2 e_0^{(5)}(a)$$

$$e_2^{(2)}(a) = J(a)e_2^{(1)}(a) - \alpha_1 e_1^{(4)}(a) - \alpha_2 e_0^{(6)}(a)$$

$$e_2^{(3)}(a) = 2J^{(1)}(a)e_2^{(1)}(a) + J(a)e_2^{(2)}(a) - \alpha_1 e_1^{(5)}(a) - \alpha_2 e_0^{(7)}(a) + b_2^{(2)}(a)$$

$$e_2^{(4)}(a) = \sum_{k=0}^{2}\binom{3}{k}J^{(k)}(a)e_2^{(3-k)}(a) - \alpha_1 e_1^{(6)}(a) - \alpha_2 e_0^{(8)}(a) + b_2^{(3)}(a)$$

$$e_2^{(5)}(a) = \sum_{k=0}^{3}\binom{4}{k}J^{(k)}(a)e_2^{(4-k)}(a) - \alpha_1 e_1^{(7)}(a) - \alpha_2 e_0^{(9)}(a) + b_2^{(4)}(a)$$

$$\vdots$$

TABLE 4 continued

Step 6: $b_3^{(1)}(a) = 0$

$$b_3^{(2)}(a) = 2f^{(2)}(a,\varphi(a))e_1^{(1)}(a)e_2^{(1)}(a)$$

$$\vdots$$

Step 7: $e_3^{(1)}(a) = -\alpha_1 e_2^{(3)}(a) - \alpha_2 e_1^{(5)}(a) - \alpha_3 e_0^{(7)}(a)$

$$e_3^{(2)}(a) = J(a)e_3^{(1)}(a) - \alpha_1 e_2^{(4)}(a) - \alpha_2 e_1^{(6)}(a) - \alpha_3 e_0^{(8)}(a)$$

$$e_3^{(3)}(a) = 2J^{(1)}(a)e_3^{(1)}(a) + J(a)e_3^{(2)}(a)$$

$$- \alpha_1 e_2^{(5)}(a) - \alpha_2 e_1^{(7)}(a) - \alpha_3 e_0^{(9)}(a) + b_3^{(2)}(a)$$

$$\vdots$$

Step 8: $e_4^{(1)}(a) = -\alpha_1 e_3^{(3)}(a) - \alpha_2 e_\beta^{(5)}(a) - \alpha_3 e_1^{(7)}(a) - \alpha_4 e_0^{(9)}(a)$

$$\vdots$$

Table 4 can be continued. Each new $e_m^{(1)}(a)$ to be computed will require two additional computations at each existing step of the table and the addition of two new steps to the table. One new step will contain the computations of $b_{m-1}^{p}(a)$ for $p=1$ and 2 and will preceed the step where $e_{m-1}^{(1)}(a)$ is computed. The other new step will be the computation of $e_m^{(1)}(a)$.

Example 3: Compute $e_1^{(1)}(a),\ldots,e_4^{(1)}(a)$ for the equation $y' = y^2$; $y(0) = .2$; $0 \leq t \leq 1$ when the trapezoid rule is

used as the numerical method for solving the problem. The generating function for the coefficients in (18) for the trapezoid rule is given by (20). Specifically, $\alpha_1 = -\frac{1}{12}$, $\alpha_2 = \frac{1}{120}$, $\alpha_3 = \frac{-17}{315} \cdot \frac{1}{2^6}$ and $\alpha_4 = \frac{62}{2835} \cdot \frac{1}{2^8}$. If the computations in Table 4 are performed for this example the results are: $e_1^{(1)}(a) = .0008$; $e_2^{(1)}(a) = .000032$; $e_3^{(1)}(a) = 1.3867 \times 10^{-6}$; and $e_4^{(1)}(a) = 6.229 \times 10^{-8}$.

As is apparent, the computation of $e_m^{(1)}(a)$ for $m \geq 4$ becomes quite involved, particularly when $q=2$. However, there are applications (see Section 1 of Chapter 3) of the pullback method with $M \leq 4$ which are more accurate than other methods currently available.

Section 5. Numerical Results for Initial Value Problems

In this section the pullback method and the method due to Lindberg [19] are compared theoretically and numerically. Since Lindberg's approach is conceptually and notationally quite different from ours, a discussion of his method is included. We will confine the discussion initially to the case $M=4$ and $q=2$ as this will suffice to point out the differences between the methods. Unfortunately, Lindberg's notation and that used here are in some instances in complete opposition to one another. The differences will be pointed out in footnotes.

Let $h>0$ be the basic steplength and for $k=0,1,\ldots,4$ define steplengths $h_k = h/2^k$ and grids $G_k = \{t_i^k = a+ih_k : i=0,\ldots,2^k\}$. In addition, let $\hat{h} = h/2^4 = h/16$*.

Assume that the numerical method being employed to solve the problem (1) is such that the expansion

$$(43)\quad Y(t_i^k,h_k) = \varphi(t_i^k)+e_1(t_i^k)h_k^2+e_2(t_i^k)h_k^4+e_3(t_i^k)h_k^6+e_4(t_i^k)h_k^8+\mathcal{O}(h_k^{10})$$

is valid for each $i=0,1,\ldots,2^k$ and for each $k=0,1,\ldots,4$.

Lindberg's method is as follows. Number all grid points according to their order of occurrence in the finest

*In Lindberg's paper $\hat{h}$ is taken to be the basic steplength and $h_k=2^k\hat{h}$ $k=0,1,\ldots,4$. Thus the grids G_k and steplengths h_k are numbered in the opposite order.

grid, obtaining $t_0=a, t_1, \ldots, t_{16}=a+h$. At each t_i perform as many extrapolations as possible and denote the computed solution after n extrapolations as $Y_{n+1}(t_i,\hat{h})$. Thus at t_{16} we can perform 4 extrapolations obtaining $Y_1(t_{16},\hat{h})$, $Y_2(t_{16},\hat{h})$, $Y_3(t_{16},\hat{h}),\ldots,$ $Y_5(t_{16},\hat{h})$ where $Y_1(t_{16},\hat{h})$ is the solution computed with the numerical method; i.e., $Y_1(t_i,\hat{h}) = Y(t_i^4,h_4)$. At t_8 we can perform 3 extrapolations; at t_4 and t_{12}, 2 extrapolations and at t_2, t_6, t_{10} and t_{14}, 1 extrapolation.

According to Lindberg each $Y_j(t,\hat{h})$ satisfies the relationship

$$(44) \qquad Y_j(t,\hat{h}) = \varphi(t) + \sum_{\nu=j}^{\infty} \kappa_{j\nu} e_\nu(t) \hat{h}^{2\nu}$$

where for each j

$$(45) \qquad \kappa_{j\nu} = \prod_{p=1}^{j-1} \frac{2^{2p}-2^{2\nu}}{2^{2p}-1} .$$

The goal here is to define Y_5 at all points of the finest grid. Initially Y_5 is known only at the two points $t_0=a$ and $t_{16}=a+h=a+16\hat{h}$. At each of these points form $Y_5(t,\hat{h})-Y_4(t,\hat{h}) = -\kappa_{44}e_4(t)\hat{h}^8+\mathcal{O}(\hat{h}^{10})$ by (44). Via linear interpolation obtain an $\mathcal{O}(\hat{h}^{10})$* approximation to $-\kappa_{44}e_4(t_8)\hat{h}^8$. Add this approximation to $Y_4(t_8,\hat{h})$ to obtain $Y_5(t_8,\hat{h})+\mathcal{O}(\hat{h}^{10})$. At the next stage form the

*This will be discussed in some detail later.

differences $Y_5(t,\hat{h})-Y_3(t,\hat{h})$ at the points $t=t_0, t_8$ and t_{16}. By (44), $Y_5(t,\hat{h})-Y_3(t,\hat{h})=-\aleph_{33}e_3(t)\hat{h}^6-\aleph_{34}e_4(t)\hat{h}^8 +\mathcal{O}(h^{10})$. Using quadratic Lagrange interpolation obtain an $\mathcal{O}(\hat{h}^3)$ approximation to $-\aleph_{33}e_3(t)\hat{h}^6-\aleph_{34}e_4(t)\hat{h}^8$ at $t=t_4$ and $t=t_{12}$ which is added to $Y_3(t,h)$ at these points to obtain $Y_5(t,\hat{h})$ for $t=t_4$ and $t=t_{12}$. However $Y_5(t,\hat{h})$, $t=t_4, t_{12}$, is only an $\mathcal{O}(h^9)$ approximation to $\varphi(t)$. This process is continued until Y_5 has been defined at all t_i, $i=0,\ldots,16$. The result, according to Lindberg is $Y_5(t_i,\hat{h})=\varphi(t_i)+\mathcal{O}(\hat{h}^9)$ for $i=1,\ldots,7$ and $9,\ldots,15$ while $Y_5(t_i,\hat{h})=\varphi(t_i)+\mathcal{O}(\hat{h}^{10})$ for $i=8$ and 16.

Now the pullback interpolation method will yield $\mathcal{O}(h^{10})$ accuracy at all t_i's and Lindberg's method yields $\mathcal{O}(\hat{h}^9)$ at most t_i's. Since $\hat{h}=h/16$ it looks as if Lindberg's method is of higher accuracy. This is not so for the following reasons: First, Lindberg presents the error analysis for extrapolation in terms of the smallest stepsize available, $\hat{h}$. The '$\mathcal{O}$' notation is designed to be information-supressing and the remainder terms in extrapolation which are said to be $\mathcal{O}(h^\beta)$ are actually of the form $Ch^\beta g(t)$ where C and β are constants and $g(t)$ is a continuous function of t. On a closed t interval $g(t)$ will be bounded and we can write $Ch^\beta g(t) \leq C_0 h^\beta = \mathcal{O}(h^\beta)$. We could also write $Ch^\beta g(t) = \hat{C}\hat{h}^\beta g(t) \leq \hat{C}_0\hat{h}^\beta = \mathcal{O}(\hat{h}^\beta)$ by defining $\hat{C} = 16^\beta C$. The problem here is the fact that $\hat{C}_0$ will be

quite large compared to C_0. When the error in extrapolation is expressed in terms of the largest or basic steplength, h, the constant C will be smaller than 1. If this same error is expressed in terms of the smallest stepsize, $\hat{h}$, the constant $\hat{C}$ will be of the form (45) and is larger than 1. In fact, when expressing the error in terms of the smallest stepsize we are unable to prove the convergence of the Aitken-Neville extrapolation scheme; indeed, an analysis of (45) shows that the constants $\mathcal{K}_{kk} \to \infty$ as the stepsizes $h_k \to 0$.

Secondly, the magnitude of the constant $\hat{C}$ is further increased in Lindberg's analysis when interpolation is performed. Recall that Lindberg expresses the error in interpolation in terms of the smallest stepsize also. This means that at the first step when Lindberg is performing linear interpolation to $-\mathcal{K}_{44} e_4(t) \hat{h}^8$ at the points t_0 and t_{16} he obtains the error $-\frac{(t-t_0)(t-t_{16})}{2!} \mathcal{K}_{44} e_4^{(2)}(\xi) \hat{h}^8$ where $t_0 < \xi < t_{16}$. Lindberg calls this error $\mathcal{O}(\hat{h}^{10})$; to do this one must interpret the maximum of $(t-t_0)(t-t_{16})$ as being $\mathcal{O}(\hat{h}^2)$. The maximum occurs at the midpoint t_8 and is easily seen to be $\left(\frac{h}{2}\right)^2 = \frac{h^2}{4} = \frac{16^2 \hat{h}^2}{4}$. To call this $\mathcal{O}(h^2)$ further increases the size of the constant being suppressed. Note that in the error analysis for the pullback method, $\left(\frac{h}{2}\right)^2$ is interpreted as $\frac{1}{4}h^2 = \mathcal{O}(h^2)$ which causes the constant to decrease.

When Lindberg claims an error of the form $\mathcal{O}(\hat{h}^{\beta})$ and the pullback method claims $\mathcal{O}(h^{\beta})$, the inequality, $\mathcal{O}(h^{\beta}) \leq \mathcal{O}(\hat{h}^{\beta})$, holds when the constants c_o and $\hat{c}_o$ are taken into account. Thus $\mathcal{O}(h^{10})$ in actuality is smaller than $\mathcal{O}(\hat{h}^{9})$ and the pullback interpolation method yields a more accurate solution.

This increase in accuracy is due solely to the fact that we are using Hermite interpolation with the additional data $e_m^{(1)}(a)$. In fact, if the pullback interpolation method is changed so that Lagrange interpolation is used the results are identical to Lindberg's. In this case $\mathcal{O}(h^{\beta}) \equiv \mathcal{O}(\hat{h}^{\beta})$.

Let's examine the differences between using Hermite and Lagrange interpolation when $q=2$ for arbitrary M.

For $j=0$, we know $e_M(t)$ at two points with accuracy $\mathcal{O}(h^2)$. Let $P_M(t)$ be the Hermite polynomial constructed as outlined and let $L_M(t)$ be the Lagrange interpolation polynomial for these two data points. We have $P_M(t) = e_M(t)+\mathcal{O}(h^3)+\mathcal{O}(h^2)$ and $L_M(t) = e_M(t)+\mathcal{O}(h^2)+\mathcal{O}(h^2)$ where the first '$\mathcal{O}$' term is the error in interpolation and the second is the error in the given data. Thus both $P_M(t)$ and $L_M(t)$ are $\mathcal{O}(h^2)$ approximations to $e_M(t)$. When $L_M(t)$ is used numerically, as it is in Lindberg's method, the solution at the midpoint of the interval is,

in most instances, less accurate than the solution which can be obtained by just performing extrapolation (see the examples concluding this section). This phenomenon, which Lindberg calls attention to in his paper, does not occur when $P_M(t)$ is used.

For $j=1$, we know $e_{M-1}(t)$ at three points with accuracy $\mathcal{O}(h^4)$. In this case, $P_{M-1}(t) = e_{M-1}(t) + \mathcal{O}(h^4)$ $+\mathcal{O}(h^4) = e_{M-1}(t) + \mathcal{O}(h^4)$ and $L_{M-1}(t) = e_{M-1}(t) + \mathcal{O}(h^3) + \mathcal{O}(h^4)$ $= e_{M-1}(t) + \mathcal{O}(h^3)$. Here it is the error in interpolation which determines the final accuracy so that there is a real gain when Hermite interpolation is used.

The situation is the same when $j=2$, that is, $P_{M-2}(t) = e_{M-t}(t) + \mathcal{O}(h^6) + \mathcal{O}(h^6)$ while $L_{M-2}(t) = e_{M-2}(t)$ $+ \mathcal{O}(h^5) + \mathcal{O}(h^6)$. Again the error in interpolation dominates and Hermite interpolation yields higher accuracy.

For any $j \geq 3$, it is the error to which we know $e_{M-j}(t)$ that dominates, since for $j \geq 3$ we have $2^j+1 > 2j+2$. The last inequality can be established by a proof similar to that of Lemma 1. Thus the advantages of using Hermite interpolation occur when $j=0,1$ and 2.

We now examine some examples of results obtained using the pullback interpolation method when numerically solving first order initial value problems. All numerical computations were done on the CDC 6500 installation at

Michigan State University. Calculations were done in single precision floating point arithmetic yielding 14 accurate decimal places.

The first example in this section is a continuation of Example 2, Section 4.

Example 4: Solve $y'=y^2$, $y(0)=.2$, $0\leq t\leq 1$ using Euler's method and pullback interpolation with $M=4$.

The basic stepsize is taken to be $h=1$ and, since $M=4$, the finest grid, G_4, consists of 17 equally spaced points in $[0,1]$. That is,

$$G_4 = \{t_i : t_i=0+i\ h/16,\ i=0,1,\ldots,16\}.$$

The theoretical solution is $y(t) = 1/(5-t)$ and the numerical results are reported in Table 5 below. Table 5 is constructed to exhibit the error in the solution computed by Euler's rule and the error in the solution computed by pullback interpolation at each grid point of G_4. In each case the error is the computed solution minus the theoretical solution.

TABLE 5

i	Error using Euler	Error using pullback
0	0	0
1	-3.15×10^{-5}	$.01 \times 10^{-7}$
2	-6.53×10^{-5}	2.97×10^{-7}
3	-10.11×10^{-5}	8.47×10^{-7}
4	-13.93×10^{-5}	13.16×10^{-7}
5	-17.99×10^{-5}	14.87×10^{-7}
6	-22.31×10^{-5}	13.26×10^{-7}
7	-26.92×10^{-5}	9.17×10^{-7}
8	-31.82×10^{-5}	3.62×10^{-7}
9	-37.05×10^{-5}	-2.01×10^{-7}
10	-42.63×10^{-5}	-6.58×10^{-7}
11	-48.58×10^{-5}	-8.92×10^{-7}
12	-54.93×10^{-5}	-8.26×10^{-7}
13	-71.71×10^{-5}	-4.97×10^{-7}
14	-68.95×10^{-5}	-1.26×10^{-7}
15	-76.68×10^{-5}	$- .54 \times 10^{-7}$
16	-84.96×10^{-5}	-3.30×10^{-7}

The results given in Table 5 can be quickly summarized by noting that Euler's method can guarantee only two accurate digits at all grid points of G_4 while the pullback method can guarantee 5 accurate digits.

Pullback interpolation based on Euler's rule with $M=4$ was also used to compute the solution to $y'=y^2$, $y(0) = .2$ with basic stepsizes of $h=1/2$ and $h=1/4$. In the first case the solution is computed on $[0,1/2]$ and in the second case the solution is computed on $[0,1/4]$. The results are summarized in Table 6 below. The first two columns are the results for $h=1/2$ and the last two are the results for $h=1/4$. Table 6 is arranged to exhibit the correspondence between grid points for the two different mesh sizes.

TABLE 6

h = 1/2		h = 1/4	
i	Error using pullback	i	Error using pullback
0	0	0	0
1	1.96×10^{-10}	1	$.15 \times 10^{-10}$
2	189.25×10^{-10}	2	11.93×10^{-10}
3	532.79×10^{-10}	3	33.93×10^{-10}
4	817.32×10^{-10}	4	50.92×10^{-10}
5	905.78×10^{-10}	5	55.90×10^{-10}
6	777.95×10^{-10}	6	47.10×10^{-10}
7	486.78×10^{-10}	7	27.94×10^{-10}
8	110.79×10^{-10}	8	3.45×10^{-10}
9	269.05×10^{-10}	9	21.15×10^{-10}
10	571.77×10^{-10}	10	40.66×10^{-10}
11	718.91×10^{-10}	11	50.02×10^{-10}
12	657.64×10^{-10}	12	45.84×10^{-10}
13	408.36×10^{-10}	13	29.33×10^{-10}
14	108.33×10^{-10}	14	9.12×10^{-10}
15	26.89×10^{-10}	15	1.17×10^{-10}
16	32.13×10^{-10}	16	$.40 \times 10^{-10}$

As can be seen by examing Table 5 and Table 6 the smaller the basic steplength is the more accurate are the computed solutions. This is in agreement with results for exprapolation (see Lambert [18] and Gragg [11,12]).

Example 5: Solve $y'=y$, $y(0)=1$, $0\leq t\leq 1$ using the trapezoid rule, and pullback interpolation with $M=4$. Using a basic steplength of $h=1$ we again obtain the solution at 17 equally spaced points in $[0,1]$. The theoretical solution of this equation is $y(t)=e^t$ and the error reported is the computed solution minus the theoretical solution. The results of Example 5 and the next example will be presented simultaneously in Table 7.

Example 6: Solve $y'=-\sin t$, $y(0)=1$, $0\leq t\leq 1$ using the trapezoid rule and pullback interpolation with $M=4$. The basic steplength is taken to be $h=1$ and the grids are the same as in Example 5. The theoretical solution is given by $y(t)=\cos(t)$.

TABLE 7

i	Error in pullback for y'=y with h=1	Error in pullback for y'=-sin t with h=1
0	0	0
1	$.12 \times 10^{-10}$	$-.3 \times 10^{-13}$
2	$.48 \times 10^{-10}$	$-.7 \times 10^{-13}$
3	1.05×10^{-10}	$-.9 \times 10^{-13}$
4	1.70×10^{-10}	$-.7 \times 10^{-13}$
5	2.30×10^{-10}	$-.3 \times 10^{-13}$
6	2.74×10^{-10}	$-.1 \times 10^{-13}$
7	2.94×10^{-10}	0.0×10^{-13}
8	2.90×10^{-10}	$-.2 \times 10^{-13}$
9	2.65×10^{-10}	$-.3 \times 10^{-13}$
10	2.34×10^{-10}	$-.3 \times 10^{-13}$
11	2.11×10^{-10}	$.2 \times 10^{-13}$
12	2.16×10^{-10}	1.2×10^{-13}
13	2.63×10^{-10}	2.5×10^{-13}
14	3.57×10^{-10}	3.6×10^{-13}
15	4.86×10^{-10}	2.9×10^{-13}
16	6.13×10^{-10}	-2.1×10^{-13}

The equations in examples 5 and 6 were solved in the same manner with a basic stepsize of h=1/2 to obtain solutions at 17 equally spaced points in [0,1/2]. The results of these computations are presented in Table 8.

TABLE 8

i	Error in pullback for y'=y with h=1/2	Error in pullback for y'=-sin t with h=1/2
0	0	0
1	0.0×10^{-13}	-0×10^{-14}
2	$.2 \times 10^{-13}$	-1×10^{-14}
3	$.4 \times 10^{-13}$	-1×10^{-14}
4	$.6 \times 10^{-13}$	-1×10^{-14}
5	$.9 \times 10^{-13}$	-0×10^{-14}
6	1.0×10^{-13}	-1×10^{-14}
7	$.7 \times 10^{-13}$	-0×10^{-14}
8	$.3 \times 10^{-13}$	-1×10^{-14}
9	$-.4 \times 10^{-13}$	-1×10^{-14}
10	$-.9 \times 10^{-13}$	-1×10^{-14}
11	-1.4×10^{-13}	-1×10^{-14}
12	-1.7×10^{-13}	-2×10^{-14}
13	-1.8×10^{-13}	-2×10^{-14}
14	-1.4×10^{-13}	-2×10^{-14}
15	$-.9 \times 10^{-13}$	-3×10^{-14}
16	$-.6 \times 10^{-13}$	-3×10^{-14}

As the results presented in Table 7 and Table 8 so vividly indicate, pullback interpolation will yield uniform accuaracy at all grid points of the finest grid.

Lindberg in [19] presents numerical results obtained when solving $y'=y$, $y(0)=1$, on $[0,1]$ using the trapezoid rule and his interpolation method. The largest error Lindberg obtains is at t_5 and has magnitude 16×10^{-10}. Examining Table 7 we see that the largest error produced for this equation when using the pullback method is the error in extrapolation at the endpoint, $t_{16}=1$. The magnitude of this error is 6.13×10^{-10}.

In the next series of examples extrapolation, Lindberg's method and pullback interpolation are compared numerically.

Example 7: Solve $y'=y^2$, $y(0)=1$, $0 \leq t \leq 1$ using the trapezoid rule. Choosing a basic steplength of $h=1$ and $M=3$, the solution is computed first by using extrapolation as often as is possible at each grid point of the finest grid, $G_3 = \{t_i : t_i=0+i\ h/8,\ i=0,1,\ldots,8\}$. Secondly, the solution is computed on G_3 by using Lindberg's method as described earlier in this section. Lastly, the solution is computed on G_3 using pullback interpolation. All errors are given as the computed solution minus the theoretical solution. The results are presented in Table 9.

TABLE 9

i	Error in the best extrapolated value	Error using Lindberg's method	Error using pullback
0	0	0	0
1	1.7×10^{-6}	$.8 \times 10^{-10}$	$-.4 \times 10^{-10}$
2	-1.0×10^{-8}	-26.7×10^{-10}	-3.0×10^{-10}
3	5.9×10^{-6}	-30.3×10^{-10}*	-4.7×10^{-10}
4	3.7×10^{-10}	-5.9×10^{-10}	-1.9×10^{-10}
5	1.2×10^{-5}	22.0×10^{-10}	3.7×10^{-10}
6	-5.3×10^{-8}	23.9×10^{-10}	6.2×10^{-10}*
7	1.9×10^{-5}*	-4.0×10^{-10}	$.8 \times 10^{-10}$
8	-1.5×10^{-10}	-1.5×10^{-10}	-1.5×10^{-10}

*greatest absolute error for the method

Example 8: Solve $y'=-\sin t$, $y(0)=1$, $0 \leq t \leq 1$ using the trapezoid rule. We again take $h=1$ and $M=3$ and compute solutions using extrapolation, Lindberg's method, and pullback interpolation. The results are given in Table 10.

TABLE 10

i	Error in the best extrapolated value	Error using Lindberg's method	Error using pullback
0	0	0	0
1	1.02×10^{-5}	178.36×10^{-10}	9.46×10^{-10}
2	-4.22×10^{-8}	37.54×10^{-10}	6.33×10^{-10}
3	9.05×10^{-5}	-47.24×10^{-10}	-2.47×10^{-10}
4	9.96×10^{-10}	-9.23×10^{-10}	$.37 \times 10^{-10}$
5	24.62×10^{-5}	31.70×10^{-10}	4.92×10^{-10}
6	-36.48×10^{-8}	-63.15×10^{-10}	-17.54×10^{-10}
7	46.76×10^{-5}*	-239.94×10^{-10}*	-62.65×10^{-10}*
8	$.96 \times 10^{-10}$	$-.96 \times 10^{-10}$	$-.96 \times 10^{-10}$

*greatest absolute error for the method

Comparing the results in Example 7 and Example 8, we see that the solutions obtained with both Lindberg's method and the pullback interpolation method, at the intermediate grid points t_i, i=1,...,7, are more accurate than the best extrapolated values. The pullback interpolation method is the most accurate of all, being almost a full decimal place better than Lindberg's method at all intermediate grid points.

As mentioned before, one drawback to Lindberg's method is that his results at the midpoint of the interval are no better, and in fact often worse, than the results

obtained by just performing extrapolation. This phenomenon does not occur when the pullback interpolation method is used, as a comparison of the errors at t_4 in Table 9 and Table 10 clearly demonstrates.

Examining Table 10 we note that the pullback method is dramatically less accurate at t_7 than it is at all other grid points. In an attempt to explain this behavior we computed the following example.

Example 9: Solve $y'=-\sin(t)$, $y(0)=1$ on $0\leq t\leq 1/2$. The only difference between this example and Example 8 is that we now take $h=1/2$ as the basic stepsize. The results are presented as

TABLE 11

i	Error in the best extrapolated value	Error in Lindberg's method	Error in pullback
0	0	0	0
1	6.36×10^{-7}	71.71×10^{-12}	3.89×10^{-12}
2	-6.62×10^{-10}	15.02×10^{-12}	2.55×10^{-12}
3	5.71×10^{-6}	-19.21×10^{-12}	-1.15×10^{-12}
4	-3.92×10^{-12}	-3.87×10^{-12}	$.03 \times 10^{-12}$
5	1.58×10^{-5}	12.74×10^{-12}	1.99×10^{-12}
6	-5.89×10^{-9}	-25.85×10^{-12}	-7.54×10^{-12}
7	$3.07 \times 10^{-5*}$	$-98.14 \times 10^{-12*}$	$-26.90 \times 10^{-12*}$
8	-1.1×10^{-13}	$-.11 \times 10^{-12}$	$-.11 \times 10^{-12}$

*greatest absolute error for the method

Once again t_7 is the least accurate solution for all three methods. The only reasonable explanation which suggests itself is that the solution $\cos t$ is changing very rapidly (relative to its behavior on the rest of the interval) between t_7 and t_8. This behavior is compensated for by extrapolation at t_8, but since no extrapolation is done at t_7 no correction is possible there.

From these examples it appears that pullback interpolation is quite sensitive to the accuracy to which we know the solution at the intermediate grid points and the type of equation we are solving. Thus it appears that it is the error in the original data and not the error in interpolation that is determining the overall accuracy of the method.

We next examine what happens when we solve the same equation on intervals $[a,b]$, $[b,c]$ and $[c,d]$ using the last computed solution as initial data for the next interval.

Example 10: Solve $y'=y^2$, $y(0)=1$, $0\leq t\leq 3/2$ by computing the solution on intervals of length $1/2$ and using the computed solution at the endpoint of the previous interval as the initial data for the equation on the next interval. The method of solution on each interval will be the trapezoid rule and pullback interpolation with $M=3$. The results are presented as Table 12.

TABLE 12

[0,1/2]		[1/2,1]		[1,3/2]	
i	Error	i	Error	i	Error
0	0	0	0	0	0
1	$-.01 \times 10^{-12}$	1	$.56 \times 10^{-12}$	1	1.13×10^{-12}
2	$-.55 \times 10^{-12}$	2	-1.20×10^{-12}	2	-4.39×10^{-12}
3	$-.74 \times 10^{-12}$	3	-1.95×10^{-12}	3	-6.93×10^{-12}
4	$.13 \times 10^{-12}$	4	$.27 \times 10^{-12}$	4	$-.64 \times 10^{-12}$
5	1.59×10^{-12}	5	4.19×10^{-12}	5	11.24×10^{-12}
6	2.21×10^{-12}*	6	5.85×10^{-12}*	6	16.07×10^{-12}*
7	$.94 \times 10^{-12}$	7	2.39×10^{-12}	7	5.04×10^{-12}
8	$.56 \times 10^{-12}$	8	1.13×10^{-12}	8	$.84 \times 10^{-12}$

*greatest absolute error

Some deterioration in the accuracy of the computed solution for larger t can be observed in Table 12. This can be partly attributed to an accumulation of roundoff errors and partly to the sensitivity of the pullback method to the accuracy of the data it receives.

If we compare the first two columns of errors in Table 12 to the results given in Table 9, we see that in terms of greatest absolute error it is better to take a smaller basic steplength and solve the problem several times in succession. However, we should point out that this can be quite expensive computationally.

Also, note that the largest absolute error always occurs at the same relative position of the three grids, namely at t_6. This is further support for our contention that the nature of the equation and the accuracy of the data points used when interpolating are the factors which control the overall accuracy of the pullback method.

In each example computed using the trapezoid rule the largest error in the pullback method has occurred to the right of the midpoint. This is no coincidence. The trapezoid rule yields more accurate solutions at grid points nearer the initial point and consequently extrapolation will be more accurate in the first half of the interval. Also, when we perform Hermite interpolation we have an extra data point at the initial grid point. All of these factors combined make it reasonable to expect that the largest errors will be produced to the right of the midpoint. Thus, a solution computed using pullback interpolation should be most reliable in the first half of the interval on which the solution is computed.

Note that the largest error in Lindberg's method can occur in the first half of the interval, as an examination of Table 9 reveals. This is due to the fact that it is the error in interpolation which determines the overall accuracy of Lindberg's method.

In summary, pullback interpolation coupled with the trapezoid rule will yield highly accurate solutions at all

grid points. In fact, if the solutions obtained with the trapezoid rule and extrapolation are sufficiently accurate, pullback interpolation will yield uniform accuracy at all grid points.

Pullback interpolation coupled with Euler's rule is not recommended as a viable solution technique. Euler's rule is simply not accurate enough to enable pullback interpolation to operate effectively.

CHAPTER II

TWO POINT BOUNDARY VALUE PROBLEMS

Section 1. The Problem and Its Discretization

In this chapter we will consider two point boundary value problems of the form

$$\begin{aligned} &x''(t)-f(t,x(t),x'(t)) = 0 \\ &x(a) = A, \quad x(b) = B. \end{aligned} \tag{1}$$

We assume that $f(t,x,x') \in C^1[[a,b] \times (-\infty,\infty) \times (-\infty,\infty)]$, $f(t,y,z)$ is uniformly Lipschitz continuous in y and z, $0 < \epsilon < \frac{\partial f}{\partial y}$ and $|\frac{\partial f}{\partial z}| \leq K$ where K is a constant. Under these assumptions (see Keller [17]) problem (1) has a unique solution which we will denote by $\varphi(t)$.

The continuous problem (1) will be denoted by

$$F(x) = 0. \tag{1'}$$

The operator $F(x) \equiv x''(t) - f(t,x(t),x'(t))$ maps the Banach space of twice continuously differentiable functions defined on $[a,b]$ into $C[a,b]$.

To obtain a discrete version of (1), let $n \geq 2$ be any natural number, define $h=(b-a)/n$ and form the uniform mesh $\{t_k=a+ih\}_{i=0}^{n}$ in $[a,b]$. The discrete problem is then given by

$$(2)\qquad \frac{X_{i+1}-2X_i+X_{i-1}}{h^2} - f\left(t_i, X_i, \frac{X_{i+1}-X_{i-1}}{2h}\right) = 0, \quad i=1,\ldots,n-1,$$

$$X_0 = A, \quad X_n = B,$$

where we have introduced the notation $X_i = X(t_i)$. Problem (2) may be thought of as a nonlinear system of equations in E^{n-1} with the unknown being the vector $(X_1,\ldots,X_{n-1})^T$. The solution to (2) will be denoted by $X(h)$ and we introduce the operator notation

$$(2')\qquad F_h(X) = 0$$

for problem (2).

In order to set up the correspondence between the continuous and discrete problems we will need to define a space discretization operator ω_h. Thus, let $z(t)$ be an arbitrary function defined on $[a,b]$ and define ω_h acting on z by

$$\omega_h z \equiv \omega_h(z(t)) = (z(t_1),\ldots,z(t_{n-1}))^T$$

$$\equiv (z_1,\ldots,z_{n-1})^T.$$

Note that $\omega_n z$ is the vector in E^{n-1} whose components are obtained by evaluating z at the grid points.

Problem (1) and the discretization (2) have been studied by Pereyra [24,27]. Stetter [31] and Pereyra [26,28] have also studied the special case of (1) when x' is not present in the equation.

To formalize what is meant by convergence of the discrete solution $X(h)$ to the continuous solution $\varphi(t)$ we have the following definition (see Lambert [18] and Pereyra [28]).

Definition 1: We shall say that the discrete solution $X(h)$ converges discretely to the theoretical solution $\varphi(t)$ if and only if

$$\lim_{h\to 0} \|X(h) - \omega_h \varphi\|_{(h)} = 0, \tag{3}$$

where $\|\cdot\|_{(h)}$ is the maximum norm on E^{n-1}.

The subscript (h) will be omitted from the norms throughout the remainder of this chapter. We should point out that Pereyra in [27] utilizes a different norm than the one we have used here, while in his earlier work [24] he uses the maximum norm.

Typically, discrete convergence depends on the two properties of consistency and stability of the discrete operator F_h in (2'). In formulating the definitions of these concepts we have followed the approach of Pereyra [25,28].

Definition 2: The operator F_h is said to be consistent of order $p > 0$ if, for each solution $\varphi(t)$ of (1) and for all $h \leq h_0$ we have

$$\|F_h(\omega_h \varphi)\| = \mathcal{O}(h^p). \tag{4}$$

Definition 3: The operator F_h is stable if for any pair of discrete functions U and W and for all $h \leq h_0$ there exists a constant $C > 0$, independent of h, such that

$$\|U-W\| \leq C \|F_h(U)-F_h(W)\|. \tag{5}$$

The standard result for discrete operators, that consistency and stability imply discrete convergence, is valid here. This theorem, in the context of linear multistep methods for ordinary differential equations, is originally due to Dahlquist [8,9]. For completeness we present Pereyra's statement of this result and briefly summarize his proof.

Lemma 1 (Pereyra [ ]): If F_h is stable then it is locally invertible around $\omega_h\varphi$ and the inverse mapping is locally Lipschitz continuous for all $h \leq h_0$.

Proof: Let $B_h = B(\omega_h\varphi,\rho)$, be the open ball of radius ρ centered at $\omega_h\varphi$, where $\rho > 0$ is independent of h. If $U,W \in B_h$ then stability implies that F_h is one-to-one on B_h, for otherwise we can violate (5). Therefore $F_h:B_h \to F_h(B_h)$ is a bijection, implying that F_h^{-1} exists on $F_h(B_h)$. If $X,Y \in F_h(B_h)$ then we can write (5) as

$$\|F_h^{-1}(X)-F_h^{-1}(Y)\| \leq C\|X-Y\|. \quad \square$$

Theorem 1. (Pereyra [25 28]): Assume (1) has a unique solution φ and F_h is stable on $B_h \equiv B(\omega_h\varphi,\rho)$ for all $h \leq h_0$.

Let F_h be consistent of order p with F. Then there exists an $\bar{h}_0 > 0$ such that:

(a) $\forall h \leq \bar{h}_0$ there exists a unique solution $X(h)$ for the discrete problem $F_h(X) = 0$, and

(b) the solution satisfies

(6) $\|X(h) - \omega_h \varphi\| = \mathcal{O}(h^p)$.

Equation (6) can be summarized by saying the solutions are (discretely) convergent of order p.

Proof: By Lemma 1, F_h is a homeomorphism between its domain B_h and its range $R_h \equiv F_h(B_h)$ $\forall h \leq h_0$. Brouwer's Invariance of Domain Theorem implies F_h maps the interior of B_h onto the interior of R_h and the boundary onto the boundary.

If $V \in \partial B_h$, stability implies that $\frac{\rho}{C} \leq \|F_h(V) - F_h(\omega_h \varphi)\|$ and therefore by letting V vary over ∂B_h we see that $B(F_h(\omega_h \varphi), \frac{\rho}{C}) \subset R_h$.

Consistency implies $\|F_h(\omega_h \varphi)\| = \mathcal{O}(h^p)$ and therefore $\|F_h(\omega_h \varphi)\| \to 0$ as $h \to 0$. Thus, there exists $\bar{h}_0 \leq h_0$ such that $\|F_h(\omega_h \varphi)\| < \frac{\rho}{C}$ $\forall h \leq \bar{h}_0$ and $0 \in B(F_h(\omega_h \varphi), \frac{\rho}{C})$. But F_h is one-to-one and onto and therefore there exists a unique $X(h) \in B_h$ such that $F_h(X(h)) = 0$.

Stability implies that

$$\|X(h) - \omega_h \varphi\| \leq C \|F_h(X(h)) - F_h(\omega_h \varphi)\| = C \|F_h(\omega_h \varphi)\| = \mathcal{O}(h^p). \quad \square$$

Thus, in order to establish the discrete convergence of the numerical solution to the theoretical solution it suffices

to show the numerical method is both consistent and stable.

The usual technique for proving that F_h is consistent is to examine the local truncation error at each grid point t_i. The local truncation error $\tau_h(t_i)$ is obtained by applying F_h to the discretization of the theoretical solution and evaluating the result at t_i. Formally

$$\tau_h(t_i) \equiv F_h(w_h\varphi)(t_i) = \frac{\varphi(t_{i+1})-2\varphi(t_i)+\varphi(t_{i-1})}{h^2} - f(t_i,\varphi(t_i),\frac{\varphi(t_{i+1})-\varphi(t_{i-1})}{2h}) \tag{7}$$

for $i=1,\ldots,n-1$.

If we assume that $f(t,y,z)$ has M total derivatives with respect to t on $[a,b]$, then by expanding $F_h(w_h\varphi)(t_i)$ in a Taylor series about $\varphi(t_i)$ it can be shown that the local truncation error $\tau_h(t_i)$ satisfies

$$\tau_h(t_i) = F(\varphi)|_{t_i} + \sum_{k=1}^{M} h^{2k}\{\frac{2}{(2k+2)!}\varphi^{(2k+2)}(t_i)-g_{2k}(\cdot)\} + \mathcal{O}(h^{2M+2}), \tag{8}$$

where t_i is any grid point in $[a,b]$. The functions g_{2k} are obtained from

$$\sum_{k=1}^{M} h^{2k}g_{2k}(\cdot) = \sum_{k=1}^{M}\{\frac{1}{k!}\frac{\partial^k f}{\partial z^k}\Big|_{(t_i,\varphi(t_i),\varphi'(t_i))}\}\cdot \{\sum_{j=1}^{M}\frac{h^{2j}}{(2j+1)!}\varphi^{(2j+1)}(t_i)\}^k, \tag{9}$$

by rearranging the right hand side in powers of h.

For instance,

$$g_2(\cdot) = \frac{1}{3!}\frac{\partial f}{\partial z}\Big|_{(t_i,\varphi(t_i),\varphi'(t_i))}\varphi'''(t_i)$$

and

$$g_4(\cdot) = \frac{1}{5!}\frac{\partial f}{\partial z}\Big|_{(t_i,\varphi(t_i),\varphi'(t_i))}\varphi^{(5)}(t_i) + \frac{1}{2!}\frac{\partial^2 f}{\partial z^2}\Big|_{(t_i,\varphi(t_i),\varphi'(t_i))}\left(\frac{1}{3!}\varphi'''(t_i)\right)^2.$$

Because equations (8) and (9) are well-known we have not given the details for constructing them. The reader interested in more detail is referred to [27] or the work in Section 2.1 of Chapter 3.

By observing that $f(\varphi) \equiv 0$, equation (8) implies that $f_h(X)$ is consistent of order 2.

In the next section we establish the stability of the finite difference operator F_h.

Section 2. Stability

In this section we will investigate the stability of the discrete scheme (2). The fact that (2) is stable in the uniform norm has been proven by Pereyra [24]. The proof given by Pereyra uses the theory of monotone operators and is quite technical in nature. We present here a simple and direct proof of the fact that F_h is stable.

In what follows we will be considering vectors V which properly belong to E^{n+1}. However we will restrict these vectors to the $(n-1)$-dimensional subspace consisting of all vectors in E^{n+1} whose first and last components are given by the boundary conditions in (2). That is, all vectors will have the form $V = (A, V_1, \ldots, V_{n-1}, B)^T$, and we will regard F_h as an operator on $V = (V_1, \ldots, V_{n-1}) \in E^{n-1}$.

Let U and W be any two vectors in E^{n-1}. For $2 \leq i \leq n-2$ the i^{th} component of $F_h(U) - F_h(W)$ is given by

(10)
$$
\begin{aligned}
&[F_h(U) - F_h(W)]_i \\
&\quad = h^{-2}(U_{i+1} - 2U_i + U_{i-1}) - f\left(t_i, U_i, \frac{U_{i+1} - U_{i-1}}{2h}\right) \\
&\quad - h^{-2}(W_{i+1} - 2W_i + W_{i-1}) + f\left(t_i, W_i, \frac{W_{i+1} - W_{i-1}}{2h}\right) \\
&\quad = h^{-2}[(U_{i+1} - W_{i+1}) - 2(U_i - W_i) + (U_{i-1} - W_{i-1}) \\
&\quad - \left[f\left(t_i, U_i, \frac{U_{i+1} - U_{i-1}}{2h}\right) - f\left(t_i, W_i, \frac{W_{i+1} - W_{i-1}}{2h}\right)\right]
\end{aligned}
$$

Using the Mean Value Theorem for continuous functions of two variables (see Widder [33]) and regarding $f(t_i, V_i, \frac{V_{i+1}-V_{i-1}}{2h})$ as a function of V_i and $\frac{V_{i+1}-V_{i-1}}{2h}$, we can obtain

$$(11) \quad f(t_i, U_i, \frac{U_{i+1}-U_{i-1}}{2h}) - f(t_i, W_i, \frac{W_{i+1}-W_{i-1}}{2h})$$

$$= f_y(t_i, \alpha_i, \beta_i)(U_i - W_i) + f_z(t_i, \alpha_i, \beta_i) \cdot$$

$$(\frac{U_{i+1}-U_{i-1}}{2h} - \frac{W_{i+1}-W_{i-1}}{2h})$$

where

$$(12a) \quad \alpha_i = W_i + \theta_i(U_i - W_i)$$

and

$$(12b) \quad \beta_i = \frac{W_{i+1}-W_{i-1}}{2h} + \theta_i (\frac{U_{i+1}-U_{i-1}}{2h} - \frac{W_{i+1}-W_{i-1}}{2h})$$

with $0 < \theta_i < 1$.

Substituting (11) into (10) we have

$$(13) \quad [F_h(U) - F_h(W)]_i$$

$$= h^{-2}(U_{i-1} - W_{i-1}) - 2h^{-2}(U_i - W_i) + h^{-2}(U_{i+1} - W_{i+1})$$

$$- f_y(t_i, \alpha_i, \beta_i)(U_i - W_i) + \frac{1}{2h} f_z(t_i, \alpha_i, \beta_i)(U_{i-1} - W_{i-1})$$

$$- \frac{1}{2h} f_z(t_i, \alpha_i, \beta_i)(U_{i+1} - W_{i+1})$$

$$= h^{-2}[(1 + \frac{h}{2} f_z(t_i, \alpha_i, \beta_i))(U_{i-1} - W_{i-1})$$

$$+(-2-h^2 f_y(t_i,\alpha_i,\beta_i))(U_i-W_i)$$

$$+(1-\frac{h}{2} f_z(t_i,\alpha_i,\beta_i))(U_{i+1}-W_{i+1})].$$

For $i=1$ and $i=n-1$ we get similar expressions except the terms multiplying $(U_0 - W_0)$ and $(U_n - W_n)$ are identically zero in the respective equations because $U_0 \equiv W_0$ and $U_n \equiv W_n$ by our earlier convention.

If we write (13) for $i=1,\ldots,n-1$ as a matrix system we obtain

$$F_h(U)-F_h(W) = M_h(U,W)(U-W), \tag{14}$$

where $M_h(U,W)$ is an $(n-1) \times (n-1)$ matrix whose rows are given by

$$[M_h(U,W)]_1 = h^{-2}(-2-h^2 f_y(t_1,\alpha_1,\beta_1), 1-\frac{h}{2} f_z(t_1,\alpha_1,\beta_1), 0,\ldots,0),$$

$$[M_h(U,W)]_i = h^{-2}(0,\ldots,0,1+\frac{h}{2} f_z(t_i,\alpha_i,\beta_i), -2-h^2 f_y(t_i,\alpha_i,\beta_i),$$

$$1-\frac{h}{2} f_z(t_i,\alpha_i,\beta_i), 0,\ldots,0)$$

for $i=2,\ldots,n-2$, and

$$[M_h(U,W)]_{n-1} = h^{-2}(0,\ldots,0,1+\frac{h}{2} f_z(t_{n-1},\alpha_{n-1},\beta_{n-1}),$$

$$-2-h^2 f_y(t_{n-1},\alpha_{n-1},\beta_{n-1})),$$

with the nonzero entries for the i^{th} row, $2 \leq i \leq n-2$, occuring in the $(i-1)$st, i^{th} and $(i+1)$st positions. The arguments α_i and β_i are given by (12).

Taking norms in (14) we have

$$\|F_h(U)-F_h(W)\| = \|M_h(U,W)(U-W)\| .$$

Multiplying both sides of this equation by $\frac{1}{\|U-W\|}$, using the linearity of $M_h(U,W)$, and noting that $\frac{U-W}{\|U-W\|}$ has norm one, we have for all $U \neq W$

$$\frac{\|F_h(U)-F_h(W)\|}{\|U-W\|} = \frac{\|M_h(U,W)(U-W)\|}{\|U-W\|}$$

$$= \|M_h(U,W)\frac{U-W}{\|U-W\|}\|$$

$$\geq \inf_{\|Z\|=1} \|M_h(U,W)\,Z\|.$$

Thus, if $\|M_h(U,W)\,Z\|$ is bounded below by a constant C, independent of h, for all vectors Z belonging to the unit ball in E^{n-1} we can write

$$\|F_h(U)-F_h(W)\| \geq C\|U-W\|,$$

which proves that F_h is stable.

To establish that $\|M_h(U,W)Z\|$ is bounded below we proceed as follows:

let $h_0 = 2/K$ and define

$$a_i = \frac{h}{2} f_z(t_i,\alpha_i,\beta_i)$$

and

$$b_i = f_y(t_i,\alpha_i,\beta_i)$$

for each $i=1,\ldots,n-1$.

For any $h < h_0$, because of our hypotheses on f_y and f_z, we see that $|a_i| < 1$ and $\beta_i > \epsilon > 0$ for all i, $1 \leq i \leq n-1$.

In this notation the matrix $M_h(U,W)$ is given by

$$[M_h(U,W)]_{1\cdot} = h^{-2}(-2-h^2b_1, 1-a_1, 0, \ldots, 0);$$

$$[M_h(U,W)]_{i\cdot} = h^{-2}(0, \ldots, 0, 1+a_i, -2-h^2b_i, 1-a_i, 0, \ldots, 0)$$

for $2 \leq i \leq n-2$; and

$$[M_h(U,W)]_{n-1\cdot} = h^{-2}(0, \ldots, 0, 1+a_{n-1}, -2-h^2b_{n-1}).$$

Now let $Z = (Z_1, \ldots, Z_{n-1})^T$ be any unit vector in E^{n-1} with respect to the maximum norm on E^{n-1}. This means that at least one component of Z has absolute value one and no component of Z has absolute value larger than one.

Suppose $Z_1 = 1$, then the first component of $M_h(U,W)Z$ has absolute value

$$\begin{aligned} h^{-2}|-2-h^2b_1+(1-a_1)Z_2| &\geq h^{-2}(|2+h^2b_1| - |1-a_1|\ |Z_2|) \\ &> h^{-2}(2+h^2b_1 - 2|Z_2|) \\ &\geq h^{-2}(2+h^2b_1-2) \\ &= b_1 \\ &> \epsilon. \end{aligned}$$

We have used the facts that $b_1 > 0$, $|a_1| < 1$ and $|z_2| \leq 1$ in making the above estimates.

Similarly, if $Z_{n-1} = 1$ then the last component of $M_h(U,W)Z$ has absolute value larger than ϵ.

Suppose $Z_i = 1$ where $2 \leq i \leq n-2$, then the i^{th} component of $M_h(U,W)Z$ has absolute value

$$h^{-2}|(1+a_i)Z_{i-1}-2-h^2b_i+(1-a_i)Z_{i+1}|$$

$$\leq h^{-2}(2+h^2b_i-|(1+a_i)Z_{i-1}+(1-a_i)Z_{i+1}|)$$

$$\leq h^{-2}(2+h^2b_i-(|Z_{i-1}+Z_{i+1}|+|a_i|\ |Z_{i-1}-Z_{i+1}|))$$

$$\leq h^{-2}(2+h^2b_i-(|Z_{i-1}+Z_{i+1}|+|Z_{i-1}-Z_{i+1}|))$$

$$\leq h^{-2}(2+h^2b_i-2)$$

$$> \epsilon.$$

The next to last inequality is valid for the following reason. Consider the expression

$$|\alpha+\beta| + |\alpha-\beta| \quad \text{with} \quad |\alpha| \leq 1, \ |\beta| \leq 1.$$

Squaring we obtain

$$(15) \quad (\alpha+\beta)^2+2|\alpha^2-\beta^2|+(\alpha-\beta)^2 = 2\alpha^2+2\beta^2+2|\alpha^2-\beta^2|.$$

If $\alpha^2 = \beta^2$ then this is trivially ≤ 4 so without loss of generality we can assume $\alpha^2 > \beta^2$. In this case equation (15) becomes

$$(|\alpha+\beta|+|\alpha-\beta|)^2 = 2\alpha^2+2\beta^2+2\alpha^2-2\beta^2 = 4\alpha^2 \leq 4.$$

Thus for any two real numbers α and β with $|\alpha| \leq 1$ and $|\beta| \leq 1$ we have

$$|\alpha+\beta| + |\alpha-\beta| \leq 2.$$

Therefore, for any unit vector Z, some component of $M_h(U,W)Z$ has absolute value larger than ϵ. This implies that $\|M_h(U,W)Z\| > \epsilon$ and consequently

$$\inf_{\|Z\|=1} \|M_h(U,W)Z\| \geq \epsilon.$$

With this our proof of stability is complete. In order to obtain the desired result we quite explicitly assumed that $0 < \epsilon < \frac{\partial f}{\partial y}$. The formal hypothesis given by Pereyra in [24] is that $0 \leq \frac{\partial f}{\partial y}$. However, he implicitly assumes that $\frac{\partial f}{\partial y}$ is bounded away from zero in his proof of stability. Thus our proof is as general as Pereyra's, and considerably more elementary.

Since the operator F_h is both consistent and stable we can apply Theorem 1 to conclude that the discrete scheme (2) has a unique solution, $X(h)$, which converges discretely to the theoretical solution, $\varphi(t)$, of problem (1).

Section 3. The Numerical Method

In this section we will summarize the results of Pereyra [24,27] and Stetter [31] concerning the global discretization error for the method (2). Using the results obtained by these authors we will develop a pullback interpolation scheme for solving boundary value problems of the form (1).

The global discretization error for the numerical method (2) is defined to be

$$X(h) - \omega_h \varphi.$$

Both Pereyra [24,27] and Stetter [31] have presented a general theorem which, when applied to our problem, becomes

Theorem 2. (Pereyra, Stetter):

Let F and F_h have $M+1$ continuous Fréchet derivatives. Then for sufficiently small h the global discretization error satisfies

$$(16) \quad X(h)-\omega_h\varphi = \sum_{k=1}^{M} h^{2k}\omega_h(e_k(t)) + \mathcal{O}(h^{2M+2}).$$

The functions $e_k(t)$ are independent of h and satisfy the linear two point boundary value problem

$$e_k''(t)-f_z(t,\varphi(t),\varphi'(t))e_k'(t)-f_y(t,\varphi(t),\varphi'(t))e_k(t) = b_k(\cdot),$$

$$(17) \quad e_k(a) = 0, \quad e_k(b) = 0.$$

The functions b_k depend on previous error functions

$e_1,\ldots,e_{k-1}$, various derivatives of φ, and various Fréchet derivatives of F.

We will prove an analogous theorem in Chapter 3 and the reader interested in details can refer to either Section 2.3 of the next chapter or to the aforementioned papers.

To set up the numerical method, define stepsizes $h_k = \frac{b-a}{2^{k+1}}$ for $k=0,1,\ldots,M$ and construct the uniform grids

$$G_k = \{t_i^k = a+ih_k : i=0,1,\ldots,2^{k+1}\} \subset [a,b].$$

Using the numerical method (2), compute a solution vector $X(h_k)$ on each grid G_k.

The computation of $X(h_k)$ for each k requires us to solve a system of $2^{k+1}-1$ equations. These equations will be nonlinear whenever f is nonlinear in either x or x'. In the nonlinear case we must use a root finding procedure to solve the system (e.g. Newton's method). In the linear case, the matrix is tridiagonal and can be easily solved using an LU decomposition and back substitution (see Isaacson and Keller [15]).

Once the solution vectors $X(h_k)$ for $k=0,1,\ldots,M$ have been obtained, extrapolation can be performed to obtain an $\mathcal{O}(h_0^{2M+2})$ approximation to $\varphi(t)$ at $t=\frac{b-a}{2}$. This has been studied by Pereyra. As was the case for initial value problems, extrapolation at other grid points does not yield comparable accuracy.

The task of implementing a pullback interpolation scheme for obtaining $\mathcal{O}(h_0^{2M+2})$ accuracy at all grid points of G_M is simplified in this case. This is due to the fact that from (17) we now have two pieces of information available to us concerning each of the error functions e_k, namely, $e_k(a) = e_k(b) = 0$. Recall that for initial value problems only one piece of information was readily available concerning e_k. Most of the effort involved in implementing pullback interpolation for initial value problems was expended in obtaining an approximation to e_k' at the initial point. However, by using the boundary conditions $e_k(a) = 0$ and $e_k(b) = 0$, we will have enough data points to enable Lagrange interpolation to yield accuracy comparable to that of our data (see the discussion in Section 5 of Chapter 1).

Thus a pullback interpolation scheme based on Lagrange interpolation can be devised for solving boundary value problems of the form (2).

At $t = \frac{b-a}{2} \equiv a+h_0$ we solve an $(M+1) \times (M+1)$ system to obtain the solution vector $(\varphi(t), e_1(t), \ldots, e_M(t))$ with accuracy $(\mathcal{O}(h_0^{2M+2}), \mathcal{O}(h_0^{2M}), \ldots, \mathcal{O}(h_0^2))$. Thus we know $e_M(a) = e_M(b) = 0$ exactly and $e_M(a+h_0)$ with accuracy $\mathcal{O}(h_0^2)$. Construct $L_M(s)$, the second degree Lagrange interpolating polynomial to the above data. Then $L_M(s) = e_M(s) + \mathcal{O}(h_0^2) + \mathcal{O}(h_0^3)$. The first '$\mathcal{O}$' term is the error in the initial data and the second is the error in interpolation.

Notice that we are using a sharper order estimate for the error of interpolation than was utilized in Chapter 1. It is well known that when performing interpolation on equally spaced points the error is of order a power of the stepsize (see Isaacson and Keller [15]). However, in Chapter 1, when interpolating on the three points a, $a+\frac{h}{2}$, and $a+h$ we referred to the error as being $\mathcal{O}(h^3)$. Actually, this error is $\mathcal{O}((\frac{h}{2})^3) < \mathcal{O}(h^3)$, so we were quite generous in our error estimates.

In order to show that pullback interpolation for solving boundary value problems yields $\mathcal{O}(h_0^{2M+2})$ accuracy at all grid points of the finest grid, we must use the sharper estimate for the error in interpolation when interpolating to e_M. When interpolating to the other error functions we can again be 'generous' and interpret the error as being in terms of h_0 which will be larger than all the other stepsizes.

The details for constructing the pullback interpolation scheme for boundary value problems are nearly identical to those given in Chapter 1. The exceptions are knowledge about e_k' is replaced by $e_k(b) = 0$ and we have one more data point each time we interpolate. Therefore, pullback interpolation will yield $\mathcal{O}(h^{2M+2})$ accuracy at all grid points of the finest grid when used in conjunction with the numerical method (2) for solving two point boundary value problems of the form (1).

CHAPTER III

THE NUMERICAL SOLUTION OF DIFFERENCE DIFFERENTIAL EQUATIONS WITH CONSTANT RETARDATION

Section 1. First Order Equations

In this section we will be concerned with difference differential equations of the form

$$\dot{x}(t) = f(t,x(t),x(t-r)), \quad r>0, \quad t>0. \tag{1}$$

Since $r>0$, (1) is an equation of retarded type. For brevity, we will sometimes refer to (1) as a delay differential equation.

This equation and variants of it occur frequently in mathematics. For instance, Lord Cherwell is credited by Wright [34] for using the equation

$$\dot{x}(t) = -\alpha x(t-1)(1+x(t))$$

in the study of the application of probability methods to the theory of asymptotic prime number densities. Cunningham [7] uses a variant of this equation as a growth model to describe a fluctuating population of organisms under certain conditions. Cooke and Yorke [4] use a variant of (1) as an economic model for the growth of capital stock.

Because of the presence of the delay term $x(t-r)$ in (1), it is necessary to know the solution at time $t-r$ to obtain a solution at time t. Therefore to solve (1) on the interval $(0,r]$, it is necessary to specify the solution

on $[-r,0]$ by means of an initial function, $\varphi(t)$. A solution of (1) with initial value φ is a continuous function which agrees with $\varphi(t)$ for $t \in [-r,0]$ and satisfies (1) for $t > 0$. The solution for $t > 0$ will also be denoted by $\varphi(t)$.

The usual method of obtaining the theoretical solution to (1) is called "the method of steps". A continuous initial function $\varphi(t)$ is specified on $[-r,0]$ and the initial value problem

$$\begin{aligned} \dot{x}(t) &= f(t,x(t),\varphi(t-r)) \\ x(0) &= \varphi(0), \qquad 0 \leq t \leq r \end{aligned} \tag{2}$$

is solved. Denoting the solution to (2) by $\varphi(t)$, the process is repeated on intervals of length r to obtain a solution, $\varphi(t)$, defined on $[-r,\infty)$.

In general, the solution $\varphi(t)$ will have jump discontinuities in various derivatives at the multiples of r, zero included. In terms of smoothness of the solution, the worst possible situation is that a jump discontinuity will occur in the $(k+1)$st derivative of $\varphi(t)$ at the point kr for $k=0,1,\ldots$. By adding conditions on $\varphi(t)$ for $t \in [-r,0]$ these jump discontinuities can be avoided to some extent. For instance, if we require that the left hand derivative of $\varphi(t)$ at $t=0$, $\varphi'(0-)$, exists and satisfies

$$\varphi'(0-) = f(0,x(0),\varphi(-r));$$

then the jump discontinuity at kr is in the $(k+2)$nd derivative for $k=0,1,\ldots$. By imposing similar conditions on the initial function we can force the jump discontinuities to occur at whatever derivative of the solution we desire.

For f to have n continuous total derivatives with respect to t on $(0,r]$ it is necessary that $\varphi(t)$ be n times continuously differentiable on $[-r,0]$. In this case the solution will be $n+1$ times continuously differentiable on $(0,r]$. However, without the added conditions discussed above on the left hand derivatives of $\varphi(t)$ at $t=0$ there will still be a discontinuity in the first and all higher order derivatives of the solution at $t=0$.

Note that as we proceed to the right using the method of steps to solve (1) we gain differentiability of the solution provided we do not lose any differentiability of f. That is, if f is n times continuously differentiable with respect to each of its arguments on $(-\infty,\infty)$ and the initial function is n times continuously differentiable on $[-r,0]$, then the solution to (1) will be $n+k$ times continuously differentiable for $t \in ((k-1)r,kr]$, $k=1,2,\ldots$.

In order to insure the existence of a unique solution, $\varphi(t)$, to (1) which is continuous on $[-r,\infty)$ we will make the following assumptions (see Hale [13]):

(i) the initial function $\varphi(t)$ is continuous on $[-r, 0]$;

(ii) $f(t,u,v)$ is a continuous function of each of its arguments;

(iii) $f(t,u,v)$ satisfies uniform Lipschitz conditions with respect to u and v.

Under these hypotheses we can also show that the solution to (1) depends continuously on the initial function $\varphi(t)$ (see Hale [13]).

In order to obtain a numerical solution of (1) we shall require more stringent conditions on f and the initial function. These conditions will be stated as they become necessary.

To solve (1) numerically, given an initial function $\varphi(t)$ on $[-r,0]$, define $F(t,x(t)) \equiv f(t,x(t),\varphi(t-r))$ and numerically solve the initial value problem

$$\dot{x}(t) = F(t,x(t))$$
$$x(0) = \varphi(0), \qquad 0 \leq t \leq r. \tag{3}$$

To solve (3) numerically we select a basic stepsize h in such a manner that r is an integer multiple of h; i.e., $r = Nh$ for some integer $N > 0$. The reason for this restriction on h will become apparent later.

As in Chapter 1 we define stepsizes $h_k = h/2^k$ and grids $G_k = \{t_i^k = ih_k : i=0,1,\ldots,2^k\}$ for $k=0,1,\ldots,M$. We assume that the numerical method employed to solve (3) is such that an asymptotic expansion of the form

$$(4) \qquad X(t_i^k,h_k) = \varphi(t_i^k) + \sum_{j=1}^{M} e_j(t_i^k)h_k^{jq} + \mathcal{O}(h_k^{(M+1)q})$$

is valid at each t_i^k for every h_k. Here, $X(t_i^k,h_k)$ is the numerical solution to (3) at the grid point t_i^k obtained with stepsize h_k and $\varphi(t_i^k)$ is the solution to (1) at t_i^k.

The existence of an expansion (4) will, in general, require the existence of $(M+1)q$ continuous derivatives of the solution $\varphi(t)$. This wil be assured if we assume that $f(t,u,v)$ has $(M+1)q-1$ continuous derivatives with respect to each of its arguments and that the initial function $\varphi(t)$ is $(M+1)q-1$ times continuously differentiable on $[-r,0]$.

If the numerical method used to solve (3) is such that $q=1$ or 2, then we can proceed as in Chapter 1 to obtain a numerical solution $X(t)$ which satisfies

$$X(t) = \varphi(t) + \mathcal{O}(h^{(M+1)q}).$$

This relationship will be valid for all t belonging to the finest grid, G_M.

The presence of jump discontinuities in derivatives of the solution at multiples of the delay r does not affect

the applicability of the pullback interpolation method. The reason for this is that the pullback interpolation method employs derivatives that are computed using the differential equation at the initial point. Since the differential equation is valid only to the right of the initial point all derivatives considered in Chapter 1 are right hand derivatives at the initial point. If $f(t,x(t),x(t-r))$ has M total derivatives with respect to t and $\varphi(t)$ has M right hand derivatives at $t=-r$, the solution of our delay differential equation will have M right hand derivatives at zero and therefore at all other multiples of r also. These derivatives can be computed by differentiating the differential equation in the same manner as was done in Chapter 1.

Repeating the solution procedure for the problem,

$$\dot{x}(t) = F(t,x(t))$$

$$x(h) = X(h), \quad h \leq t \leq 2h,$$

we can obtain an $\mathcal{O}(h^{(M+1)q})$ solution to (1) at the points $\{h+ih_M : i=0,1,\ldots,2^M\}$.

If we repeat the above procedure N times we will have a computed solution, $X(t)$, satisfying

$$X(t) = \varphi(t) + \mathcal{O}(h^{(M+1)q}) \tag{5}$$

for all t such that

$$t \in \{jh+ih_M : i=0,1,\ldots,2^M,\ j=0,1,\ldots,N-1\} = \{ih_M : i=0,\ ,\ldots,N2^M\}$$

$$\subset [0,r].$$

Let's examine what happens when we now try to obtain a solution to (1) on $[r,2r]$. Suppose t is any grid point in $[r,2r]$; because $G_k \subset G_M$ $\forall k$, t can be represented as $t=r+ih_M$ where $0 \leq i \leq N2^M$. Since $t \in [r,2r]$, $t-r \in [0,r]$ and in order to solve (1) we must be able to evaluate $\varphi(t-r)$. Using the above representation of t, we can write $t-r = r+ih_M-r = ih_M$ with $0 \leq i \leq N2^M$. This is a grid point in $[0,r]$ and by (5) we have

$$\varphi(t-r) = X(ih_M) + \mathcal{O}(h^{(M+1)q}). \tag{6}$$

It is easy to see that there is a 1-1 correspondence between grid points in $[r,2r]$ and grid points in $[0,r]$. This fact and equation (6) emphasize the importance of the pullback method for obtaining numerical solutions to difference differential equations. No numerical method is any more accurate than the data it receives and the solution of (1) on $[r,2r]$ requires the solution on $[0,r]$ as data. The uniform accuracy guaranteed by (6) means that the entire process used to obtain the solution on $[0,r]$ can be repeated to obtain a solution on $[r,2r]$.

Because normal extrapolation without pullback interpolation produces a solution whose accuracy varies from grid point to grid point, it is not an effective procedure for this problem.

Extrapolation has been used to obtain solutions to problems of the form (1). Feldstein [10] has developed two algorithms based on Euler's rule that have (partial) asymptotic expansions of the form

$$(7) \qquad X(t) = \varphi(t) + e(t)h + \mathcal{O}(h^2).$$

An expansion of the form (7) justifies the use of one extrapolation.

Cooke and List [3] have developed an algorithm which they call the "modified Euler-Feldstein" algorithm which can be used to solve delay equations with non-constant retardation. They indicate that a sketch of a proof that their method has an expansion of the form (7) has been supplied by John Hutchison. However, neither the proof nor its sketch is contained in [3].

When using the "Euler-Feldstein" or the "modified Euler-Feldstein" algorithm the basic stepsize h is taken to be quite small (usually $r/16$ or smaller) and the problem is solved on several grids with stepsizes $h/2^k$. Extrapolation is performed at all multiples of the basic stepsize. The retarded term $x(t-r)$ is obtained by using the solution in the previous interval that was computed using the same stepsize with which one is now working.

This procedure appears to work well in practice but is computationally quite expensive since the basic

stepsize is so small. Also, the existence of an expansion of the form (7) is not sufficient justification for using more than one extrapolation.

Of course, systems of delay equations can be solved numerically by generalizing the above procedure in the obvious manner. A more interesting problem is the equation

$$\dot{x}(t) = f(t,x(t),x(t-r_1),x(t-r_2),\ldots,x(t-r_n))$$

where $r_i > 0$ is a rational number for $i=1,\ldots,n$. The same numerical procedure will work for this problem provided the basic stepsize is chosen in a manner which insures that the information required by the delayed terms is available. Without loss of generality, assume the delays r_i are ordered: $0 \leq r_1 \leq r_2 \leq \ldots \leq r_n$. Let d be the least common multiple of the denominators of the r_i for $i=1,\ldots,n$, then with $h=1/d$ this equation may be solved in the same manner as (1). This choice of the basic stepsize insures that each r_i will be an integer multiple of the smallest stepsize employed and therefore $t-r_i$ will be a grid point at which the solution is known $\forall i$ whenever t itself is a grid point.

We close this section with two numerical examples.

Example 1: Solve $\dot{x}(t) = -x(t-1)$, $0 \leq t \leq 3$, numerically for the initial function $\varphi(t) = t^2$ on $[-1,0]$.

Here, the delay is $r=1$ and the theoretical solution is given by

$$\varphi(t) = \begin{cases} \dfrac{-(t-1)^3-1}{3} & 0 \le t \le 1 \\ \dfrac{(t-2)^4+4t-9}{12} & 1 \le t \le 2 \\ \dfrac{-(t-3)^5-10t^2+65t-96}{60} & 2 \le t \le 3 \end{cases} .$$

Note that $\varphi(t)$ is given by a polynomial on each subinterval.

To compute a numerical solution on $[0,3]$, we first solve the equation numerically on $[0,1]$ using the trapezoid rule, extrapolation and pullback interpolation with $M=3$. The finest grid G_3 contains nine equally spaced points in $[0,1]$. Denote the computed solution at these nine points by $X_1(t)$. Using $X_1(t)$ as the (discrete) initial function we next solve the same equation on $[1,2]$ in the same manner to obtain $X_2(t)$. This procedure is repeated a third time to obtain $X_3(t)$ on $[2,3]$.

The computations of $e_1'(t)$, $e_2'(t)$ and $e_3'(t)$ at the initial points $0,1$ and 2 are greatly simplified for this problem. One factor contributing to this simplification is that the Jacobian $J(t)$, is identically zero. The second factor is that all derivatives of the solution $\varphi(t)$ at the points $0,1$ and 2 can be expressed in terms of the derivatives of the initial function $\varphi(t)=t^2$ at

$t=-1$ and the computed solution at smaller multiples of $r=1$. In fact, by noting that for any $k=0,1,2,\ldots$ and for any $j > k$ we have $x^{(j)}(kr) = x^{(j-1)}((k-1)r)$ it is easily seen that $x^{(j)}(kr) = x^{(j-k-1)}(-r)$. While, if $j \leq k$, then $x^{(j)}(kr) = x((k-j)r)$.

The numerical results are presented as Table 13. The error given is the computed solution minus the theoretical solution. Each column of Table 13 presents the errors at the nine equally spaced grid points $t_0, t_1, \ldots, t_8$ contained in the interval indicated by the heading of the column.

TABLE 13

i	Error on [0,1]	Error on [1,2]	Error on [2,3]
0	0.0	0.0×10^{-13}	-1.5×10^{-13}
1	0.0×10^{-13}	$-.3 \times 10^{-13}$	0.0×10^{-13}
2	0.0×10^{-13}	$-.4 \times 10^{-13}$	$.2 \times 10^{-13}$
3	0.0×10^{-13}	$-.5 \times 10^{-13}$	$.3 \times 10^{-13}$
4	0.0×10^{-13}	$-.6 \times 10^{-13}$	$.4 \times 10^{-13}$
5	0.0×10^{-13}	$-.8 \times 10^{-13}$	$.5 \times 10^{-13}$
6	0.0×10^{-13}	-1.0×10^{-13}	$.6 \times 10^{-13}$
7	0.0×10^{-13}	-1.3×10^{-13}	$.7 \times 10^{-13}$
8	0.0×10^{-13}	-1.5×10^{-13}	$.8 \times 10^{-13}$

The errors reported in Table 13 are almost entirely due to the limitations of machine accuracy and the particular

routine used to compute the Hermite interpolating polynomials. Only the first twelve digits can be absolutely guaranteed to be accurate and to twelve decimal places all errors are zero! This is not particularly surprising, since the trapezoid rule and extrapolation yield extremely good results when the theoretical solution is a polynomial, as it is in this case.

Example 2: Solve $\dot{x}(t) = -x(t-1)$, $0 \leq t \leq 3$ numerically for the initial function $\varphi(t) = e^t$ on $[-1,0]$. The differential equation is the same as that in Example 1. The theoretical solution for this initial function is given by

$$\varphi(t) = \begin{cases} -e^{(t-1)} + 1 + e^{-1} & 0 \leq t \leq 1 \\ e^{(t-2)} - (1+e^{-1})(t-1) & 1 \leq t \leq 2 \\ -e^{(t-3)} + (1+e^{-1})\frac{t^2}{2} - 2(1+e^{-1})(t-1) & 2 \leq t \leq 3 \end{cases}.$$

In this case the solutions are not polynomials.

The same numerical method was used to compute the solution as was used in Example 1. The results are presented as Table 14.

TABLE 14

i	Error on [0,1]	Error on [1,2]	Error on [2,3]
0	0.0	-1.2×10^{-10}	-7.2×10^{-10}
1	-5.3×10^{-10}	4.5×10^{-10}	-12.8×10^{-10}
2	-3.2×10^{-10}	3.0×10^{-10}	-11.2×10^{-10}
3	2.3×10^{-10}	-2.6×10^{-10}	-5.6×10^{-10}
4	$.7 \times 10^{-10}$	-1.2×10^{-10}	-7.0×10^{-10}
5	-2.4×10^{-10}	2.1×10^{-10}	-10.2×10^{-10}
6	12.0×10^{-10}	-12.7×10^{-10}	4.7×10^{-10}
7	42.4×10^{-10}	-46.7×10^{-10}	38.9×10^{-10}
8	-1.2×10^{-10}	-7.2×10^{-10}	$.3 \times 10^{-10}$

Since the theoretical solution is not a polynomial, the computed solution is not as accurate as the computed solution in Example 1. However, we do obtain eight reliable decimal places at all grid points, which is comparable to the accuracy that extrapolation yields at the endpoints 1, 2 and 3.

Also, it should be noted how stable the numerical results are. There is very little deterioration in the accuracy of the computed solution as we move to the right. Thus, extrapolation and pullback interpolation appears to be a viable solution technique for first order delay equations.

Section 2. Second Order Equations

For the remainder of this chapter we will be working with the second order difference differential equation of retarded type

$$(8)\quad \ddot{x}(t) + f(t,x(t),x(t-r),\dot{x}(t),\dot{x}(t-r)) = 0, \quad r > 0, \quad t > 0.$$

By writing (8) as a system of two first order equations and imposing conditions (i), (ii) and (iii) of Section 1 or the more general conditions of Sansone [30] on this system we can insure the existence of a unique solution to (8) which depends continuously on the initial data (see Hale [13] and Norkin [23]).

In either case, f is assumed to be a continuous function of its arguments. For our purposes it is enough to assume the existence of a unique solution to (8) for $t > 0$ whenever we are given a continuously differentiable function $\varphi(t)$ on $[-r,0]$.

To solve (8) theoretically, the method of steps can again be used. The analysis of the behavior of the solution to (8) is completely analogous to that given for first order equations in Section 1 of this chapter. Accordingly, we will not go into this in detail. When specific aspects of the theoretical solution become important they will be mentioned.

If (8) is written as a system we can of course use the method discussed in Section 1 to solve this system.

The rest of this chapter will be devoted to an investigation of a direct method for solving (8) which does not involve reducing it to a system of first order equations.

We introduce the notation

$$F(x) = 0 \tag{8'}$$

for the continuous operator defined by (8) and we will refer to (8) as the continuous problem.

For ease in referring to partial derivatives we will refer to equation (8) as

$$\ddot{x}(t) + f(t,u,v,y,z) = 0.$$

In setting up a discrete analog of (8) we actually get two slightly different problems; one when we are trying to obtain the solution on $[0,r]$ and the second when we seek a solution on an interval of the form $[(L-1)r,Lr]$ with $L>1$. The version of the problem for the interval $[0,r]$ is a special case of the version for the general interval $[(L-1)r,Lr]$ and will be explained in the subsection on implementing the algorithm (see section 2.4). Thus until further notice we will concern ourselves with the discrete analog of (8) on the interval $[(L-1)r,Lr]$ with $L>1$.

Let $R>1$ be any given natural number and define $h = r/R$. Form the uniform mesh $\{t_i=(L-1)r+ih\}_{i=-(R+1)}^{R}$.

We assume the solution has already been obtained at the grid points t_i for $i=-(R+1),-R,\ldots,0$ and we denote this solution by $\varphi(t_i)$.

Setting $X_i = X(t_i)$, the discrete problem corresponding to (8) is given by

$$\frac{X_{i+1}-2X_i+X_{i-1}}{h^2} + f\left(t_i, X_i, X_{i-R}, \frac{X_{i+1}-X_{i-1}}{2h}, \frac{X_{i+1-R}-X_{i-1-R}}{2h}\right) = 0,$$

(9)

$$i=0,1,\ldots,R-1;$$

$$X_{-j} = \varphi(t_{-j}), \quad j=0,1,\ldots,R+1.$$

The discrete problem (9) will be denoted by

(9') $$F_h(X) = 0.$$

Since X_{-j} is known for $j=0,\ldots,R+1$, this may be thought of as a (in general) nonlinear system of equations in E^R with the unknown being the vector $(X_1,\ldots,X_R)^T$. The solution to (9) will be denoted by $X(h)$.

Since each equation in (9) taken in the order $i=0,1,\ldots,R-1$ introduces only one unknown, we can solve the system uniquely by forward substitution. In general, the nonlinear nature of f will necessitate the use of a root finding procedure (e.g. Newton's method) to solve each equation.

We will adopt the same space discretization operator w_h that was utilized in Chapter 2. That is, for any continuous function $z(t)$ on $[-r,\infty)$

$$\omega_h z(t) = (z(t_1),\ldots,z(t_R))^T$$

$$\equiv (z_1,\ldots,z_R)^T.$$

We will continue to use the definitions of discrete convergence, consistency and stability (definitions 1, 2 and 3 respectively) that were adopted in Section 1 of Chapter 2.

The proofs of Lemma 1 and Theorem 1 in Section 1 of Chapter 2 are valid in this setting. In fact, Pereyra mentions in [28] that these results are not truly tied up to the two-point boundary value problem.

In order to establish the existence and discrete convergence of a unique solution $X(h)$ to (9) we will again show that (9) is consistent and stable.

In order to prove that F_h is stable we have to make some restrictive assumptions about the equation (8). These assumptions are necessary for our proof of stability to be valid. However, we believe that F_h is stable for a much larger class of functions $f(t,u,v,y,z)$ than what we are able to establish.

Section 2.1. Consistency

Let $\varphi(t)$ denote the theoretical solution to the continuous problem, (8). If we apply the operator F_h defined by (9) to the discretization of $\varphi(t)$ we obtain what is known as the local truncation error, τ_h. Formally

$$\begin{aligned}\tau_h(t_i) &\equiv F_h(\omega_h\varphi)(t_i)\\ &\equiv \frac{\varphi(t_{i+1})-2\varphi(t_i)+\varphi(t_{i-1})}{h^2} + f\Big(t_i,\varphi(t_i),\varphi(t_{i-R}),\\ &\qquad \frac{\varphi(t_{i+1})-\varphi(t_{i-1})}{2h},\ \frac{\varphi(t_{i+1-R})-\varphi(t_{i-1-R})}{2h}\Big)\end{aligned} \tag{10}$$

where t_i is any grid point $i=1,\ldots,R$.

If $\tau_h(t_i) = \mathcal{O}(h^p)$ $\forall i$ then we obviously have that our method is consistent of order p. Instead of investigating (10) directly we let $x(t)$ be any sufficiently differentiable function and investigate $F_h(\omega_h x)(t_i)$. If we make the "localizing" assumption that no previous truncation errors have been made, we may use Taylor's Theorem repeatedly to obtain an asymptotic expansion for $F_h(\omega_h x)(t_i)$.

Let t_i be any grid point, $i=1,\ldots,R$. If $x(t)$ has M_1+2 derivatives at t_i we can write

$$\begin{aligned}x(t_{i+1}) &= x(t_i)+hx'(t_i)+\frac{h^2}{2!}x''(t_i)\\ &\quad + \sum_{j=3}^{M_1+1}\frac{h^j}{j!}x^{(j)}(t_i)+\mathcal{O}(h^{M_1+2})\end{aligned} \tag{11}$$

and

$$(12)\quad x(t_{i-1}) = x(t_i) - hx'(t_i) + \frac{h^2}{2!}x''(t_i) + \sum_{j=3}^{M_1+1} \frac{(-1)^j h^j}{j!} x^{(j)}(t_i) + \mathcal{O}(h^{M_1+2}).$$

Using (11) and (12) we have

$$(13)\quad \frac{x(t_{i+1})-2x(t_i)+x(t_{i-1})}{h^2}$$

$$= h^{-2}\left(h^2 x''(t_i) + \sum_{k=2}^{[\frac{M_1}{2}]} \frac{h^{2k}}{2k!} x^{(2k)}(t_i) + \mathcal{O}(h^{2[\frac{M_1}{2}]+2})\right)$$

$$= x''(t_i) + 2\sum_{k=2}^{K} \frac{h^{2k-2}}{2k!} x^{(2k)}(t_i) + \mathcal{O}(h^{2K})$$

where $[\cdot]$ is the greatest integer function and $K = [\frac{M_1}{2}]$.

Expanding $f(\cdot,\cdot,\cdot,y,\cdot)$ in a Taylor series about $x'(t_i)$ where $y = \frac{x(t_{i+1})-x(t_{i-1})}{2h}$ we obtain

$$(14)\quad f(\cdot,\cdot,\cdot,y,\cdot) = f(\cdot,\cdot,\cdot,x'(t_i),\cdot) + f_y(\cdot,\cdot,\cdot,x'(t_i),\cdot)(y-x'(t_i))$$
$$+ f_{yy}(\cdot,\cdot,\cdot,x'(t_i),\cdot)\frac{(y-x'(t_i))^2}{2!}$$
$$+ \sum_{j=3}^{M_2} \frac{1}{j!} \frac{\partial^j f}{\partial y^j}\Big|(\cdot,\cdot,\cdot,x'(t_i),\cdot)\,(y-x'(t_i))^j$$
$$+ \mathcal{O}[(y-x'(t_i))^{M_2+1}],$$

where M_2+1 is the number of partial derivatives of $f(t,u,v,y,z)$ with respect to y which exist.

Using (11), (12) and the definition of y, we obtain

$$(15)\quad y-x'(t_i) \equiv \frac{1}{2h}\{x(t_{i+1})-x(t_{i-1})\}-x'(t_i)$$

$$= \frac{1}{2h}\{2hx'(t_i)+2\sum_{j=1}^{K}\frac{h^{2j+1}}{(2j+1)!}\cdot(t_i)+\mathcal{O}(h^{2K+1})\}-x'(t_i)$$

$$= \sum_{j=1}^{K}\frac{h^{2j}}{(2j+1)!}x^{(2j+1)}(t_i)+\mathcal{O}(h^{2K}).$$

Substituting (15) in (14), we have

$$f(\cdot,\cdot,\cdot,y,\cdot) = f(\cdot,\cdot,\cdot,x'(t_i),\cdot)+f_y(\cdot,\cdot,\cdot,x'(t_i),\cdot)\cdot$$

$$\{\sum_{j=1}^{K}\frac{h^{2j}}{(2j+1)!}x^{(2j+1)}(t_i)+\mathcal{O}(h^{2K})\}$$

$$+\frac{1}{2}f_{yy}(\cdot,\cdot,\cdot,x'(t_i),\cdot)\{\sum_{j=1}^{K}\frac{h^{2j}}{(2j+1)!}x^{(2j+1)}(t_i)+\mathcal{O}(h^{2K})\}^2$$

$$+\sum_{k=3}^{M_2}\{\frac{1}{k!}\frac{\partial^k f}{\partial y^k}\Big|(\cdot,\cdot,\cdot,x'(t_i),\cdot)\}\cdot$$

$$\{\sum_{j=1}^{K}\frac{h^{2j}}{(2j+1)!}x^{(2j+1)}(t_i)+\mathcal{O}(h^{2k})\}^k$$

$$+\mathcal{O}(\{\sum_{j=1}^{K}\frac{h^{2j}}{(2j+1)!}x^{(2j+1)}(t_i)+\mathcal{O}(h^{2K})\}^{M_2+1}).$$

But $\mathcal{O}(\{\sum_{j=1}^{K}\frac{h^{2j}}{(2j+1)!}x^{(2j+1)}(t_i)+\mathcal{O}(h^{2K})\}^{M_2+1}) =$

$\mathcal{O}(\{\mathcal{O}(h^2)+\mathcal{O}(h^{2K})\}^{M_2+1}) = \mathcal{O}(h^{2M_2+2})$, and for any $\ell=1,\ldots,M_2$, we have, from the binomial theorem,

$$\{\sum_{j=1}^{K}\frac{h^{2j}}{(2j+1)!}x^{(2j+1)}(t_i)+\mathcal{O}(h^{2K})\}^{\ell}=(\sum_{j=1}^{K}\frac{h^{2j}}{(2j+1)!}x^{(2j+1)}(t_i))^{\ell}$$

$$+\mathcal{O}(h^{2K}).$$

Consequently we can write

$$(16)\quad f(\cdot,\cdot,\cdot,y,\cdot)=f(\cdot,\cdot,\cdot,x'(t_i),\cdot)+f_y(\cdot,\cdot,\cdot,x'(t_i),\cdot)\cdot$$

$$\sum_{j=1}^{K}\frac{h^{2j}}{(2j+1)!}x^{(2j+1)}(t_i)$$

$$+\sum_{k=2}^{M_2}\{\frac{1}{k!}\frac{\partial^k f}{\partial y^k}\Big|(\cdot,\cdot,\cdot,x'(t_i),\cdot)\}\cdot$$

$$\{\sum_{j=1}^{K}\frac{h^{2j}}{(2j+1)!}x^{(2j+1)}(t_i)\}^k$$

$$+\mathcal{O}(h^{2M_2+2})+\mathcal{O}(h^{2K}).$$

Next, expand $f(.,.,.,x'(t_i),z)$ and its partial derivatives with respect to y as functions of z in Taylor series about $x'(t_i-r)$ and then set $z=\frac{x(t_{i+1-R})-x(t_{i-1-R})}{2h}$ to obtain

$$(17)\quad f(\cdot,\cdot,\cdot,x'(t_i),z)=f(\cdot,\cdot,\cdot,x'(t_i),x'(t_i-r))$$

$$+f_z(\cdot,\cdot,\cdot,x'(t_i),x'(t_i-r))(z-x'(t_i-r))$$

$$+\sum_{\ell=2}^{M_3}\{\frac{1}{\ell!}\frac{\partial^{\ell}f}{\partial z^{\ell}}\Big|(\cdot,\cdot,\cdot,x'(t_i),x'(t_i-r))\}\cdot$$

$$(z-x'(t_i-r))^{\ell}$$

$$+\mathcal{O}(\{z-x'(t_i-r)\}^{M_3+1}).$$

In the same manner, for $m=1,\ldots,M_2$, we can obtain

$$(18)\quad \frac{\partial^m f}{\partial y^m}\Big|(\cdot,\cdot,\cdot,x'(t_i),z) = \frac{\partial^m f}{\partial y^m}\Big|(\cdot,\cdot,\cdot,x'(t_i),x'(t_i-r))$$

$$+ \frac{\partial^{m+1} f}{\partial z\,\partial y^m}\Big|(\cdot,\cdot,\cdot,x'(t_i),x'(t_i-r))\,(z-x'(t_i-r))$$

$$+ \sum_{\ell=2}^{M_3} \frac{1}{\ell!}\,\frac{\partial^{m+\ell} f}{\partial z^{\ell}\,\partial y^m}\Big|(\cdot,\cdot,\cdot,x'(t_i),x'(t_i-r))\,(z-x'(t_i-r))^{\ell}$$

$$+\mathcal{O}(\{z-x'(t_i-r)\}^{M_3+1});$$

where M_3+1 is the number of partial derivatives of $f(t,u,v,y,z)$ with respect to z which exist.

Since $t_i-r \equiv t_{i-R}$, an expansion similar to (15) is valid for $z-x'(t_i-r)$. Introducing the notation $\frac{\partial^\circ f}{\partial y^\circ}$ for the function f we can, in direct analog to (16), rewrite (17) and (18) as

$$(19)\quad \frac{\partial^m f}{\partial y^m}\Big|(\cdot,\cdot,\cdot,x'(t_i),z) = \frac{\partial^m f}{\partial y^m}\Big|(\cdot,\cdot,\cdot,x'(t_i),x'(t_i-r))$$

$$+ \sum_{\ell=1}^{M_3}\Big\{\frac{1}{\ell!}\,\frac{\partial^{m+\ell} f}{\partial z^{\ell}\,\partial y^m}\Big|(\cdot,\cdot,\cdot,x'(t_i),x'(t_i-r))\Big\}\cdot$$

$$\Big\{\sum_{j=1}^{K}\frac{h^{2j}}{(2j+1)!}\,x^{(2j+1)}(t_i-r)\Big\}^{\ell}$$

$$+\mathcal{O}(h^{2M_3+2}) + \mathcal{O}(h^{2K}),$$

for $m=0,1,\ldots,M_2$.

Combining equations (16) and (19), we have

$$
\begin{aligned}
(20)\quad f(\cdot,\cdot,\cdot,y,z) &= f(\cdot,\cdot,\cdot,x'(t_i),x'(t_i-r)) \\
&+ \sum_{\ell=1}^{M_3} \frac{1}{\ell!} \frac{\partial^\ell f}{\partial z^\ell}\Big|_{(\cdot,\cdot,\cdot,x'(t_i),x'(t_i-r))} \cdot \{ \sum_{j=1}^{K} \frac{h^{2j}}{(2j+1)!} x^{(2j+1)}(t_i-r) \}^\ell \\
&+ \mathcal{O}(h^{2M_3+2}) + \mathcal{O}(h^{2K}) \\
&+ \frac{\partial f}{\partial y}\Big|(\cdot,\cdot,\cdot,x'(t_i),x'(t_i-r)) \sum_{j=1}^{K} \frac{h^{2j}}{(2j+1)!} x^{(2j+1)}(t_i) \\
&+ \Big(\sum_{\ell=1}^{M_3} \{ \frac{1}{\ell!} \frac{\partial^{1+\ell} f}{\partial z^\ell \partial y}\Big|_{(\cdot,\cdot,\cdot,x'(t_i),x'(t_i-r))} \cdot \\
&\quad \{ \sum_{j=1}^{K} \frac{h^{2j}}{(2j+1)!} x^{(2j+1)}(t_i-r) \}^\ell \Big) \cdot \sum_{j=1}^{K} \frac{h^{2j}}{(2j+1)!} x^{(2j+1)}(t_i) \\
&+ \mathcal{O}(h^{2M_3+4}) + \mathcal{O}(h^{2K+2}) \\
&+ \sum_{k=2}^{M_2} \frac{1}{k!} \{ \Big(\sum_{\ell=0}^{M_3} [\frac{1}{\ell!} \frac{\partial^{k+\ell} f}{\partial z^\ell \partial y^k}\Big|_{(\cdot,\cdot,\cdot,x'(t_i),x'(t_i-r))} \Big] \cdot \\
&\quad [\sum_{j=1}^{K} \frac{h^{2j}}{(2j+1)!} x^{(2j+1)}(t_i-r)]^\ell \Big) \cdot \Big(\sum_{j=1}^{K} \frac{h^{2j}}{(2k+1)!} x^{(2j+1)}(t_i) \Big)^k \} \\
&+ \mathcal{O}(h^{2M_2+2}) + \mathcal{O}(h^{2M_2+2}) + \mathcal{O}(h^{2K}).
\end{aligned}
$$

Combining equations (13) and (20), and evaluating all derivatives at $(t_i, x(t_i), x(t_i-r), x'(t_i), x'(t_i-r))$ we can write $F_h(\omega_h x)(t_i)$ as

$$F_h(\omega_h x)(t_i) \equiv \frac{x(t_{i+1})-2\ (t_i)+x(t_{i-1})}{h^2} + f(t_i, x(t_i), x(t_{i-R}),$$

$$\frac{x(t_{i+1})-x(t_{i-1})}{2h}, \frac{x(t_{i+1-R})-x(t_{i-1-R})}{2h})$$

$$= x''(t_i) + 2\sum_{k=2}^{K} \frac{h^{2k-2}}{(2k)!} x^{(2k)}(t_i) + \mathcal{O}(h^{2K})$$

$$+ f(t_i, x(t_i), x(t_i-r), x'(t_i), x'(t_i-r))$$

$$+ \sum_{\ell=1}^{M_3} \frac{1}{\ell!} \frac{\partial^\ell f}{\partial z^\ell} \left(\sum_{j=1}^{K} \frac{h^{2j}}{(2j+1)!} x^{(2j+1)}(t_i-r)\right)^\ell$$

$$+ \sum_{k=1}^{M_2} \frac{1}{k!} \left\{\left[\sum_{\ell=0}^{M_3} \frac{\partial^{k+\ell} f}{\partial z^\ell \partial y^k} \left(\sum_{j=1}^{K} \frac{h^{2j}}{(2j+1)!} x^{(2j+1)}(t_i-r)\right)^\ell\right]\right.$$

$$\left.\left[\sum_{j=1}^{K} \frac{h^{2j}}{(2j+1)!} x^{(2j+1)}(t_i)\right]^k\right\}$$

$$+ \mathcal{O}(h^{2M_2+2}) + \mathcal{O}(h^{2M_3+2}) + \mathcal{O}(h^{2K}).$$

Define $M = \min(K-1, M_2, M_3)$; then, reindexing, we have

$$(21)\quad F_h(\omega_h x)(t_i) = F(x)(t_i) + 2\sum_{k=2}^{M} \frac{h^{2k-2}}{(2k)!} x^{(2k)}(t_i)$$

$$+ \sum_{\ell=1}^{M} \frac{1}{\ell!} \frac{\partial^\ell f}{\partial z^\ell} \left(\sum_{j=1}^{M} \frac{h^{2j}}{(2j+1)!} x^{(2j+1)}(t_i-r)\right)^\ell$$

$$+ \sum_{k=1}^{M} \frac{1}{k!} \left\{\left[\sum_{\ell=0}^{M} \frac{\partial^{k+\ell} f}{\partial z^\ell \partial y^k} \left(\sum_{j=1}^{M} \frac{h^{2j}}{(2j+1)!} x^{(2j+1)}(t_i-r)\right)^\ell\right]\right.$$

$$\left.\left[\sum_{j=1}^{M} \frac{h^{2j}}{(2j+1)!} x^{(2j+1)}(t_i)\right]^k\right\}$$

$$+ \mathcal{O}(h^{2M+2}).$$

If we regroup (21) in powers of h we obtain

$$(21') \quad F_h(\omega_h x)(t_i) = F(x)(t_i) + \sum_{j=1}^{M} h^{2j} g_{2j}(\cdot)\Big|_{x,t_i} + \mathcal{O}(h^{2M+2}),$$

where the arguments of the functions $g_{2j}(\cdot)$ involve various derivatives of x evaluated at t_i and $t_i - r$, and various partial derivatives of $f(t,u,v,y,z)$ with respect to y and/or z evaluated at $(t_i, x(t_i-r), x'(t_i), x'(t_i-r))$. For instance

$$g_2(\cdot) = \frac{1}{12} x^{(4)}(t_i) + \frac{1}{6}\left(\frac{\partial f}{\partial z} x^{(3)}(t_i - r) + \frac{\partial f}{\partial y} x^{(3)}(t_i)\right)$$

and

$$\begin{aligned} g_4(\cdot) = {} & \frac{2}{6!} x^{(6)}(t_i) + \frac{1}{2!} \frac{1}{(3!)^2} \frac{\partial^2 f}{\partial z^2} \left(x^{(3)}(t_i - r)\right)^2 \\ & + \frac{1}{5!}\left(\frac{\partial f}{\partial z} x^{(5)}(t_i - r) + \frac{\partial f}{\partial y} x^{(5)}(t_k)\right) \\ & + \frac{1}{(3!)^2} \frac{\partial^2 f}{\partial z\, \partial y} x^{(3)}(t_i - r) x^{(3)}(t_i) + \frac{\partial^2 f}{\partial y^2}\left(x^{(3)}(t_i)\right)^2). \end{aligned}$$

To obtain the local truncation error for the discrete problem (9), by (10) we need only take $x(t)$ to be the theoretical solution $\varphi(t)$ of problem (8) and use either equation (21) or (21'). Thus,

$$(22) \quad \tau_h(t_i) = F(\varphi)(t_i) + \sum_{j=1}^{M} h^{2j} g_{2j}(\cdot)\Big|_{\varphi,t_i} + \mathcal{O}(h^{2M+2})$$

from equation (21').

To establish the consistency of our discrete operator we need only note that $F(\varphi) \equiv 0$. Consequently

$\tau_h(t_i) = \mathcal{O}(h^2)$ for every grid point t_i, i=1,...,R. Thus our discrete operator is consistent of order 2.

Section 2.2. Stability

In this section we investigate the stability of the discrete scheme (9). The first part of the analysis is done for the general method and is analogous to the work done in Section 2 of Chapter 2.

In investigating the stability of the discrete problem (9), it is convenient to view F_h as a function from E^R to E^R. The domain of F_h will consist of vectors $V = (V_1,\ldots,V_R)^T \in E^R$. Of course, F_h continues to have the initial conditions

$$V_{-j} = \varphi(t_{-j}), \quad j=0,1,\ldots,R+1.$$

When considering the j^{th} component of $F_h(V)$ we will suppress the (constant) first, third, and fifth arguments of $f(t,u,v,y,z)$ and write

$$f(t_{j-1}, V_{j-1}, V_{j-1-R}, \frac{V_j - V_{j-2}}{2h}, \frac{V_{j-R} - V_{j-2-R}}{2h})$$

$$= f(\cdot, V_{j-1}, \cdot, \frac{V_j - V_{j-2}}{2h}, \cdot) .$$

Note that the j^{th} component of $F_h(V)$ is obtained from the $(j-1)st$ equation in (9) for $j=1,\ldots,R$.

Let U,W be any two vectors in E^R with $U \neq W$. The cases $j=1$ and $j=2$ are considered individually later. For $3 \leq j \leq R$ the j^{th} component of $F_h(U)-F_h(W)$ is

given by

$$(23)\quad [F_h(Y)-F_h(W)]_j$$

$$= h^{-2}(U_{j-2}-2U_{j-1}+U_j)+f(\cdot,U_{j-1},\cdot,\frac{U_j-U_{j-2}}{2h},\cdot)$$

$$-h^{-2}(W_{j-2}-2W_{j-1}+W_j)+f(\cdot,W_{j-1},\cdot,\frac{W_j-W_{j-2}}{2h},\cdot)$$

$$= h^{-2}[(U_{j-2}-W_{j-2})-2(U_{j-1}-W_{j-1})+(U_j-W_j)]$$

$$+f(\cdot,U_{j-1},\cdot,\frac{U_j-U_{j-2}}{2h},\cdot)-f(\cdot,W_{j-1},\cdot,\frac{W_j-W_{j-2}}{2h},\cdot).$$

Proceeding as in Chapter 2, we use the Mean Value Theorem for continuous functions of two variables (Widder [33]) to obtain

$$(24)\quad f(\cdot,U_{j-1},\cdot\frac{U_j-U_{j-2}}{2h},\cdot)-f(\cdot,W_{j-1},\cdot,\frac{W_j-W_{j-2}}{2h},\cdot)$$

$$= f_u(\cdot,\alpha_j,\cdot,\beta_j,\cdot)(U_{j-1}-W_{j-1})$$

$$+f_y(\cdot,\alpha_j,\cdot,\beta_j,\cdot)(\frac{U_j-U_{j-2}}{2h}-\frac{W_j-W_{j-2}}{2h});$$

where

$$(25a)\qquad \alpha_j = W_{j-1} + \theta_j(U_{j-1}-W_{j-1})$$

and

$$(25b)\quad \beta_j = \frac{W_j-W_{j-2}}{2h} + \theta_j(\frac{U_j-U_{j-2}}{2h}-\frac{W_j-W_{j-2}}{2h}),$$

with $0 < \theta_j < 1$.

Substituting (24) into (23) we have for

$3 \leq j \leq R$

$$(26)\quad [F_h(U)-F_h(W)]_j$$

$$= h^{-2}(U_{j-2}-W_{j-2})-2h^{-2}(U_{j-1}-W_{j-1})+h^{-2}(U_j-W_j)$$

$$+f_u(\cdot,\alpha_j,\cdot,\beta_j,\cdot)(U_{j-1}-W_{j-1})$$

$$+\frac{1}{2h}f_y(\ ,\alpha_j,\ ,\beta_j,\ )(U_j-W_j)-\frac{1}{2h}f_y(\cdot,\alpha_j,\cdot,\beta_j,\cdot)\cdot$$

$$\left(\frac{U_{j-2}-W_{j-2}}{2h}\right)$$

$$= h^{-2}[(1-\frac{h}{2}f_y(\cdot,\alpha_j,\cdot,\beta_j,\cdot))(U_{j-2}-W_{j-2})$$

$$+(-2+h^2f_u(\cdot,\alpha_j,\cdot\beta_j,\cdot))(U_{j-1}-W_{j-1})$$

$$+(1+\frac{h}{2}f_y(\cdot,\alpha_j,\cdot\beta_j,\cdot))(U_j-W_j)].$$

For $j=1$ and 2 we can reason in the same manner to obtain

$$(27)\quad [F_h(U)-F_h(W)]_1 = h^{-2}(1+\frac{h}{2}f_y(\cdot,\alpha_1,\cdot,\beta_1,\cdot))(U_1-W_1)$$

and

$$(28)\quad [F_h(U)-F_h(W)]_2 = h^{-2}[(-2+h^2f_u(\cdot,\alpha_2,\cdot,\beta_2,\cdot)(U_1-W_1)$$

$$+(1+\frac{h}{2}f_y(\cdot,\alpha_2,\cdot,\beta_2,\cdot)(U_2-W_2)].$$

The functions $\alpha_1,\beta_1,\alpha_2,$ and β_2 are given by

$$(29a)\qquad \alpha_1 = \varphi(t_o) \equiv W_o,$$

$$(29b)\qquad \beta_1 = \frac{W_1-W_{-1}}{2h} + \theta_1\left(\frac{U_1-W_1}{2h}\right),$$

$$(30a)\qquad \alpha_2 = W_1 + \theta_2(U_1-W_1),$$

and

$$\beta_2 = \frac{W_2 - W_0}{2h} + \theta_2 \left(\frac{U_2 - W_2}{2h}\right) \tag{30b}$$

with $0 < \theta_1 < 1$ and $0 < \theta_2 < 1$.

Writing (26), (27) and (28) as a matrix system we obtain

$$F_h(U) - F_h(W) = M_h(U,W)(U-W) \tag{31}$$

where $M_h(U,W)$ is an $R \times R$ matrix whose rows are given by:

$$[M_h(U,W)]_{1\cdot} = h^{-2}\left(1 + \frac{h}{2}f_y(\cdot,\alpha_1,\cdot\beta_1,\cdot), 0, 0, \ldots, 0\right);$$

$$[M_h(U,W)]_{2\cdot} = h^{-2}\left(-2+h^2 f_u(\cdot,\alpha_2,\cdot\beta_2,\cdot), 1 + \frac{h}{2}f_y(\cdot,\alpha_2,\cdot\beta_2,\cdot), 0,\ldots,0\right); \tag{32}$$

$$[M_h(U,W)]_{j\cdot} = h^{-2}\left(0,\ldots,0, 1 - \frac{h}{2}f_y(\cdot,\alpha_j,\cdot\beta_j,\cdot), -2+h^2 f_u(\cdot,\alpha_j,\cdot,\beta_j,\cdot), 1 + \frac{h}{2}f_y(\cdot,\alpha_j,\cdot,\beta_j,\cdot), 0,\ldots,0\right)$$

for $3 \leq j \leq R$;

with the non-zero entries for $j \geq 3$ occurring in the (j-2)nd, (j-1)st and jth positions. The arguments α_j and β_j for j=1,...,R are given by the appropriate equation (25), (29) or (30). Note that since $R = r/h$ the dimensionality of $M_h(U,W)$ depends on h.

Since $M_h(U,W)$ is not diagonally dominant, as it was for boundary value problems, we are not able to prove directly that $M_h(U,W)$ is bounded below. However, we do have the following equivalent formulation of the concept of stability.

Suppose $M_h(U,W)$ has an inverse $M_h^{-1}(U,W)$ that is bounded above in norm, independent of h. Then we can write (31) as

$$U-W = M_h^{-1}(U,W)\,(F_h(U)-F_h(W)).$$

Taking norms and using the fact that the norm of a product is less than or equal to the product of the norms, we have

$$\|U-W\| \leq \|M_h^{-1}(U,W)\|\ \|F_h(U)-F_h(W)\| \leq C\|F_h(U)-F_h(W)\|$$

where C is the bound on $\|M_h^{-1}(U,W)\|$. This inequality is precisely the definition of stability given in Section 1 of Chapter 2.

The norm inequality we have used is valid as long as we use the matrix norm induced by our vector norm. Since we are working with the maximum vector norm on E^R, the induced matrix norm is the maximum absolute row sum (see Isaacson and Keller [15]).

We are also unable to prove directly that $\|M_h^{-1}(U,W)\|$ is uniformly bounded above. However, we are able to establish this fact with some additional hypotheses on f.

For notational convenience we will assume that $r=1$. This is a rescaling of the delay interval and has no effect on any of our results. We will also write the dimensionality of the matrix system (34) as $n \equiv R = 1/h$.

The special case for which we will prove stability is given by the following assumptions:

$$\frac{\partial f}{\partial y} \equiv 0,$$

(33)

$$-2 < -K < \frac{\partial f}{\partial u} < -\sigma < 0,$$

where $\sigma < K$ are positive constants.

Write $b_i = f_u(\cdot,\alpha_i,\cdot,\beta_i,\cdot)$. Then the assumptions (33) imply that b_i is negative for all i; in fact,

(34) $$-K < b_i < -\sigma.$$

Rewriting the matrix $M_h \equiv M_h(U,W)$ given in (32) in terms of b_i and n and imposing the assumptions (33), we obtain

$$M_h = n^2 \cdot \begin{bmatrix} 1, & 0, & 0, & 0, & \cdots & 0 \\ -2+\frac{b_2}{n^2}, & 1, & 0, & 0, & & \vdots \\ 1, & -2+\frac{b_3}{n^2}, & 1, & 0, & & \vdots \\ 0, & 1, & -2+\frac{b_4}{n^2}, & 1, & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0, & \cdots & 0, & 1, & -2+\frac{b_n}{n^2}, & 1 \end{bmatrix}.$$

Define

$$S_h = \frac{1}{n^2} \begin{bmatrix} 0, & -\frac{1}{4}, & 0, & 0, & \cdots & 0 \\ 0, & 0, & -\frac{1}{4}, & 0, & & \vdots \\ 0, & 0, & 0, & -\frac{1}{4}, & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & 0 \\ 0, & & & & 0, & -\frac{1}{4} \\ \frac{n}{4n^2}, & \frac{n-1}{4n^2}, & \frac{n-2}{4n^2}, & \frac{n-3}{4n^2}, & \cdots \frac{2}{4n^2}, & \frac{1}{4n^2} \end{bmatrix},$$

then

$$S_h M_h = \begin{bmatrix} \frac{1}{2} - \frac{b_2}{4n^2}, & -\frac{1}{4}, & 0 & \cdots & 0 \\ -\frac{1}{4} & \frac{1}{2} - \frac{b_3}{4n^2}, & -\frac{1}{4} & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & & -\frac{1}{4}, & \frac{1}{2} - \frac{b_n}{4n^2}, & -\frac{1}{4} \\ \frac{(n-1)b_2}{4n^4}, & \frac{(n-2)b_3}{4n^4}, & \cdots & \frac{b_n}{4n^4}, & \frac{1}{4n^2} \end{bmatrix}.$$

Write

$$S_h M_h = I_h - T_h \tag{35}$$

with

$$T_h = \begin{bmatrix} \frac{1}{2} + \frac{b_2}{4n^2}, & \frac{1}{4}, & 0 & \cdots & 0 \\ \frac{1}{4}, & \frac{1}{2} + \frac{b_3}{4n^2}, & \frac{1}{4} & & \vdots \\ 0, & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & & & \frac{1}{4}, & \frac{1}{2} + \frac{b_n}{4n^2}, & \frac{1}{4} \\ -\frac{(n-1)b_2}{4n^4}, & -\frac{(n-2)b_3}{4n^4}, & \cdots & -\frac{b_n}{4n^4}, & 1 - \frac{1}{4n^2} \end{bmatrix}$$

where I_h is the $n \times n$ identity matrix.

Because of (34), for n sufficiently large (h sufficiently small) we have

$$0 < \frac{1}{2} + \frac{b_i}{4n^2} < \frac{1}{2} - \frac{\sigma}{4n^2}$$

for $i=2,\ldots,n$. Therefore the first through $(n-1)$st rows of T_h all have absolute row sum less than $1 - \frac{\sigma}{4n^2}$.

All entries of the last row of T_h are positive, so the absolute row sum of the n^{th} row is

$$\begin{aligned} 1 - \frac{1}{4n^2} - \frac{1}{4n^4} \sum_{i=1}^{n-1} i b_{n+1-i} &= 1 - \frac{1}{4n^2} + \frac{1}{4n^4} \sum_{i=1}^{n-1} i |b_{n+1-i}| \\ &\leq 1 - \frac{1}{4n^2} + \frac{1}{4n^4} K \sum_{i=1}^{n-1} i \\ &= 1 - \frac{1}{4n^2} + \frac{Kn(n-1)}{8n^4} . \end{aligned}$$

Let $\delta = 2 - K > 0$, then

$$\begin{aligned} \frac{1}{4n^2} - \frac{Kn(n-1)}{8n^4} &= \frac{1}{8n^2} \left(2 - K\left(\frac{n-1}{n}\right)\right) \\ &> \frac{1}{8n^2} (2-K) \\ &= \frac{\delta}{8n^2} . \end{aligned}$$

Therefore,

$$\begin{aligned} 1 - \frac{1}{4n^2} - \frac{1}{4n^4} \sum_{i=1}^{n-1} i b_{n+1-i} &\leq 1 - \left(\frac{1}{4n^2} - \frac{Kn(n-1)}{8n^2}\right) \\ &< 1 - \frac{\delta}{8n^2} . \end{aligned}$$

Thus,

$$\|T_h\| \leq \max\ (1 - \frac{\sigma}{4n^2},\ 1 - \frac{\delta}{8n^2})$$

$$= 1 - \frac{1}{4n^2} \min\ (\sigma, \frac{\delta}{2}).$$

Consequently, for all n $\|T_h\| < 1$. This implies by a well known theorem in linear algebra (see Varga [32] or Isaacson and Keller [15]) that $(I_h - T_h)^{-1}$ exists and moreover

$$\|(I_h - T_h)^{-1}\| \leq \frac{1}{1 - \|T_h\|}. \tag{36}$$

S_h is easily seen to be invertible by observing that it is row equivalent to I_h. Therefore M_h must be nonsingular and from (35) we have

$$M_h^{-1} = (I_h - T_h)^{-1} S_h. \tag{37}$$

Taking norms in (37) and using the norm inequality (36), we can write

$$\|M_h^{-1}\| \leq \|(I_h - T_h)^{-1}\| \cdot \|S_h\| \tag{38}$$

$$\leq \frac{\|S_h\|}{1 - \|T_h\|}.$$

Since $\|T_h\| \leq 1 - \frac{1}{4n^2} \min\ (\sigma, \frac{\delta}{2})$, we have

$$1 - \|T_h\| \geq 1 - (1 - \frac{1}{4n^2},\ \min\ (\sigma, \frac{\delta}{2}))$$

$$= \frac{1}{4n^2} \min\ (\sigma, \frac{\delta}{2}).$$

But

$$\|S_h\| = \max\left(\frac{1}{4n^2}, \frac{1}{4n^4}\sum_{i=1}^{n} i\right)$$

$$= \max\left(\frac{1}{4n^2}, \frac{1}{4n^4}\frac{n(n+1)}{2}\right)$$

$$= \frac{1}{4n^2}\max\left(1, \frac{n+1}{2n}\right)$$

$$= \frac{1}{4n^2}.$$

Finally, using the above estimates in (38), we have

$$\|M_h^{-1}\| \leq \frac{\frac{1}{4n^2}}{\frac{1}{4n^2}\min\left(\sigma, \frac{\delta}{2}\right)} = \frac{1}{\min\left(\sigma, \frac{\delta}{2}\right)}.$$

Since this is independent of n, and therefore of h, we can conclude that $\|M_h^{-1}\| \leq C$; where C is a finite constant independent of h.

Thus F_h is stable and Theorem 1 of Chapter 2 allows us to conclude that (9) has a unique solution $X(h)$ which converges discretely to $\varphi(t)$.

Clearly, the assumptions (33) are required by our method of proof. We believe that there is nothing inherent to the operator F_h which would prevent a more general proof, but at this time we are unable to construct one.

In the next section we will return to the study of the general operator F_h, without the assumptions (33), and examine its global discretization error.

Section 2.3. The Global Discretization Error

In this section we investigate $E(h) \equiv X(h) - \omega_h \varphi$, the global discretization error for our numerical method. Assuming the discrete convergence of our method, we know that $E(h) \to 0$ as $h \to 0$. We seek more information as to the nature of $E(h)$ as a function of h. Ideally we would like $E(h)$ to have an asymptotic expansion in even powers of h. That is, we would like to show that $\forall t \in [(L-1)r, Lr]$.

$$(39) \qquad E(h) = \omega_h \left(\sum_{k=1}^{M} e_k(t) h^{2k} \right) + \mathcal{O}(h^{2M+2})$$

with the functions $e_k(t)$ independent of h. Equivalently, we want to show how to determine $e_k(t)$ for $k=1,2,\ldots,M$ and $t \in [(L-1)r, Lr]$ so that

$$(40) \quad S(h) \equiv E(h) - \omega_h \left(\sum_{k=1}^{M} e_k(t) h^{2k} \right) = \mathcal{O}(h^{2M+2}).$$

If we write

$$\mu(h) = \sum_{k=1}^{M} e_k(t) h^{2k}$$

then we can rewrite (40) as

$$S(h) = E(h) - \omega_h(\mu(h)).$$

Let $I = F_h(\omega_h(\varphi + \mu(h)))$, then since $F_h(X(h)) = 0$ we have

$$\|I\| = \|F_h(\omega_h(\varphi + \mu(h))) - F_h(X(h))\|.$$

Using the linearity of ω_h and the stability of F_h, we can write

$$\|I\| \geq \frac{1}{C} \|X(h) - \omega_h\varphi - \omega_h\mu(h)\|$$

$$= \frac{1}{C} \|E(h) - \omega_h\mu(h)\|$$

$$= \frac{1}{C} \|S(h)\|.$$

Consequently, if we can show how to choose $e_k(t)$ for $k=1,\ldots M$ so that $\|I\| = \mathcal{O}(h^{2K+2})$ then (40) will be valid and we will have the desired asymptotic expansion.

Theorem 1. Let $F_h(V)$ have $M+1$ continuous Fréchet derivatives with respect to V. Then,

$$X(h) - \omega_h\varphi = \sum_{k=1}^{M} h^{2k} \omega_h e_k(t) + \mathcal{O}(h^{2M+2})$$

where the vectors $\omega_h e_k(t)$ are solutions of

$$F_h'(\omega_h\varphi)\,\omega_h e_k = \omega_h B_k. \tag{41}$$

The vectors $\omega_h B_k$ will be defined in the proof.

Proof. Expand $I = F_h(\omega_h(\varphi+\mu(h)))$ in a Taylor series about the vector $\omega_h\varphi$ to get

$$I = F_h(\omega_h\varphi) + F_h'(\omega_h\varphi)\,\omega_h\mu(h) + \sum_{k=2}^{M} \frac{1}{k!} F_h^{(k)}(\omega_h\varphi)\,(\omega_h\mu(h))^k + R_{M+1}. \tag{42}$$

Here $F_h^{(k)}(\omega_h\varphi)$ is the k^{th} Fréchet derivative of F_h evaluated at $\omega_h\varphi$. The remainder term R_{M+1} involves $[\mu(h)]^{M+1}$ which is $\mathcal{O}(h^{2M+2})$ and therefore R_{M+1} is $\mathcal{O}(h^{2M+2})$.

$F_h(\omega_h\varphi)$ is the vector of local truncation errors for the discrete operator F_h. By equation (22) each component of $F_h(\omega_h\varphi)$ is given by

$$\tau_h(t_i) = F(\varphi)(t_i) + \sum_{j=1}^{M} h^{2j} g_{2j}(\cdot)\big|_{\varphi, t_i} + \mathcal{O}(h^{2M+2}).$$

Using the space discretization ω_h, we can write

$$(43)\quad F_h(\omega_h\varphi) = \omega_h\tau_h = \omega_h F(\varphi) + \sum_{j=1}^{M} h^{2j}\omega_h(g_{2j}(\cdot)\big|_{\varphi}) + \mathcal{O}(h^{2M+2}).$$

Here we have used the linearity of ω_h. Also, the $\mathcal{O}(h^{2M+2})$ term in (43) is understood to be a vector each of whose components is $\mathcal{O}(h^{2M+2})$.

Noting that $F(\varphi) \equiv 0$ and substituting (43) into (42), we obtain

$$(44)\quad I = \sum_{j=1}^{M} h^{2j}\omega_h g_{2j}(\cdot)\big|_{\varphi} + F_h'(\omega_h\varphi)\,\omega_h\mu(h) + \sum_{k=2}^{M}\frac{1}{k!}F_h^{(k)}(\omega_h\varphi)[\omega_h\mu(h)]^k + \mathcal{O}(h^{2M+2}).$$

Using the definition of $\mu(h)$, it is immediate that the coefficient of h^2 in (44) is

$$\omega_h g_2 + F_h'(\omega_h\varphi)\,\omega_h e_1.$$

In order to eliminate this term from the expansion (44), define $\omega_h B_1 = -\omega_h g_2$ and take $\omega_h e_1$ to be the solution of

$$F_h'(\omega_h\varphi)\,\omega_h e_1 = \omega_h B_1.$$

With $\omega_h e_1$ so determined we can write (44) as

$$I = \sum_{j=2}^{M} h^{2j} \omega_h g_{2j} + F'(\omega_h \varphi)\, \omega_h \Big(\sum_{j=2}^{M} h^{2j} e_j \Big) + \sum_{k=2}^{M} \frac{1}{k!} F_h^{(k)}(\omega_h \varphi) [\omega_h u(h)]^k + \mathcal{O}(h^{2M+2}).$$

Collecting the terms involving h^4 we have

$$h^4 \omega_h g_4 + F'(\omega_h \varphi) h^4 \omega_h e_2 + \frac{1}{2} F_h^{(2)}(\omega_h \varphi) h^2 \omega_h e_1 h^2 \omega_h e_1.$$

Define $\omega_h B_2 = -\omega_h g_4 - \frac{1}{2} F_h^{(2)}(\omega_h \varphi)\, \omega_h e_1 \omega_h e_1$ and obtain $\omega_h e_2$ by solving

$$F_h'(\omega_h \varphi)\, \omega_h e_2 = \omega_h B_2.$$

Substituting $\omega_h e_2$ in the expression for I we can eliminate the terms involving h^4.

Suppose $\omega_h e_1, \omega_h e_2, \ldots, \omega_h e_J$ have all been determined as solutions to equations of the form (41), so that the expression for I is

$$I = \sum_{j=J+1}^{M} h^{2j} \omega_h g_{2j}(\cdot)\big|_{\varphi} + F'(\omega_h \varphi)\, \omega_h u(h) + \sum_{k=J+1}^{M} \frac{1}{k!} F_h^{(k)}(\omega_h \varphi) [\omega_h u(h)]^k + \sum_{k=J+1}^{JM} G_{2k}(\cdot) h^{2k} + \mathcal{O}(h^{2M+2}).$$

The arguments of the functions $G_{2k}(\cdot)$ involve various error vectors $\omega_h e_j$ for $j=1,\ldots,J$ and various Fréchet derivatives

$F_h^{(j)}(\omega_h\varphi)$ for $j=2,\ldots,J$. If

$$\sum_{j=2}^{J} \frac{1}{j!} F_h^{(j)}(\omega_h\varphi)[\omega_h u(h)]^j$$

is expressed as a power series in h^2 then for $k=J+1,\ldots,JM$ the function $G_{2k}(\cdot)$ can be obtained as the coefficient of h^{2k} in this expansion.

Collecting terms that include h^{2J+2} we have

$$\omega_h g_{2J+2} h^{2J+2} + G_{2J+2}(\cdot) h^{2J+2} + F'(\omega_h\varphi)\omega_h e_{J+1}.$$

Because the function G_{2J+2} includes only the error vectors $\omega_h e_1,\ldots,\omega_h e_J$ as arguments of the various Fréchet derivatives it contains, we are able to define $\omega_h B_{J+1}$ as

$$\omega_h B_{J+1} = -\omega_h g_{2J+2} - G_{2J+2}.$$

Therefore, we can determine $\omega_h e_{J+1}$ as the solution to

$$F_h'(\omega_h\varphi)\omega_h e_{J+1} = \omega_h B_{J+1}.$$

Hence by induction Theorem 1 is valid. □

Thus the global discretization error will have an expansion of the form (39) provided $F_h(V)$ has M+1 continuous Fréchet derivatives and we can solve equations of the form (41).

In the finite dimensional case, the Fréchet derivative of a discrete differential operator $F_h(V)$ is just the Jacobian matrix of the operator considered as a vector-valued

function of the R-dimensional vector $V = (V_1,\ldots,V_R)^T$. For our particular difference operator defined by (9) the j^{th} row of the Jacobian matrix is obtained by computing the partials with respect to the components of V of equation (9) with $i=j-1$.

We will denote the j^{th} row of $F_h'(V)$ by $[F_h'(V)]_{j\cdot}$. Introducing the notation

$$f^k(V) = f(t_k, V_k, V_{k-R}, \frac{V_{k+1}-V_{k-1}}{2h}, \frac{V_{k+1-R}-V_{k-1-R}}{2h})$$

and using it for the partial derivatives of $f(t,u,v,y,z)$, we can write $F_h'(\omega_h\varphi)$ as

$$[F_h'(\omega_h\varphi)]_{1\cdot} = h^{-2}(1 + \frac{h}{2}f_y^\circ(\omega_h\varphi), 0,\ldots,0);$$

$$[F_h'(\omega_h\varphi)]_{2\cdot} = h^{-2}(-2+h^2 f_u^1(\omega_h\varphi), 1 + \frac{h}{2}f_y^1(\omega_h\varphi), 0,\ldots,0);$$

and for $3 \le j \le R$,

$$[F_h'(\omega_h\varphi)]_{j\cdot} = h^{-2}(0,\ldots,0,1 - \frac{h}{2}f_y^{j-1}(\omega_h\varphi), -2+h^2 f_u^{j-1}(\omega_h\varphi),$$

$$1 + \frac{h}{2}f_y^{j-1}(\omega_h\varphi), 0,\ldots,0)$$

with the non zero entries in the j^{th} row for $3 \le j \le R$ occurring in the $(j-2)$nd, $(j-1)$st and j^{th} positions.

Since $F_h'(\omega_h\varphi)$ is lower triangular, it will be nonsingular provided none of the diagonal entries vanish. For any j the $(j,j)^{th}$ entry of $F_h'(\omega_h\varphi)$ is

$$h^{-2} + \frac{1}{2h} f_y(\cdot).$$

This will be non zero provided

$$f_y(\cdot) \neq \frac{-2}{h} . \tag{45}$$

If we assume that $f_y(t,u,v,y,z)$ is bounded, then as $h \to 0$ the left hand side of (45) is bounded while the right hand side diverges to $-\infty$. Thus for sufficiently small h, (45) will be valid and $F_h'(\omega_h\varphi)$ will be nonsingular. Consequently, the equation (41) will be uniquely solvable.

Now let us consider the system of equations

$$F_h'(\omega_h\varphi)\,\omega_h e_k = 0 \tag{46}$$

in more detail. The j^{th} equation in (46) is given by

$$h^{-2}\left(1 - \frac{h}{2}f_y^{j-1}(\omega_h\varphi)\right)e_k(t_{j-2}) + h^{-2}\left(-2+h^2 f_u^{j-1}(\omega_h\varphi)\right)e_k(t_{j-1})$$
$$+h^{-2}\left(1 + \frac{h}{2}f_y^{j-1}(\omega_h\varphi)\right)e_k(t_j) = 0.$$

Equivalently, we can write this as

$$\frac{e_k(t_{j-2})-2e_k(t_{j-1})+e_k(t_j)}{h^2} + f_y^{j-1}(\omega_h\varphi)\frac{e_k(t_j)-e_k(t_{j-2})}{2h}$$
$$+ f_u^{j-1}(\omega_h\varphi)e_k(t_{j-1}) = 0. \tag{47}$$

This equation is the discrete analog of

$$e_k''(t) + f_y(\varphi)e_k'(t) + f_u(\varphi)e_k(t) = 0 \tag{48}$$

at $t=t_{j-1}$, where $f_y(\varphi) = f_y(t,\varphi(t),\varphi(t-r),\varphi'(t),\varphi'(t-r))$.

Equation (48) is the linear variational equation associated

with the continuous operator $F(x) = 0$ and is often denoted by $F'(\varphi)e_k = 0$.

By examining the first and second equations in (46) we see that $e_k(t_{-1}) = e_k(t_0) = 0$. In fact, since $E(h) \equiv X(h) - \omega_h\varphi$ and we have taken $X_{-j} = \varphi(t_{-j})$ for all $j=0,1,\ldots,R+1$ we must have $e_k(t_{-j}) = 0 \quad \forall k$ and $\forall j = 0,\ldots,R+1$.

One method for obtaining (48) is to let $h \to 0$ in (47). If we do this, the points t_{-j} become dense in the interval $[(L-2)r,(L-1)r]$ and we therefore must have $e_k(t) \equiv 0$ on $[(L-2)r,(L-1)r]$. This implies that $e_k'(t) \equiv 0$ on $[(L-2)r,(L-1)r]$. Since the solution to (48) will have a continuous first derivative, we see that the discrete problem (47) with the initial conditions $e_k(t_{-1}) = e_k(t_0) = 0$ corresponds to the continuous problem (48) with the initial conditions $e_k(t_0) = e_k'(t_0) = 0$.

Consequently, the vectors $\omega_h e_k$ in (41) are actually discretizations of differentiable functions $e_k(t)$ which satisfy

$$e_k''(t) + f_y(\varphi)e_k'(t) + f_u(\varphi)e_k(t) = B_k(\cdot)$$
$$e_k(t_0) = e_k'(t_0) = 0. \tag{48'}$$

To summarize, we have established that the global discretization error has an asymptotic expansion in even powers of h:

$$X(h)(t_i) = \varphi(t_i) + \sum_{k=1}^{M} e_k(t_i)h^{2k} + \mathcal{O}(h^{2M+2}).$$

Moreover, the functions $e_k(t)$ are independent of h and satisfy $e_k(t_0) = e_k'(t_0) = 0$.

In the next section we will discuss the implementation of this method and the modifications necessary to obtain a solution on $[0,r]$.

Section 2.4. Implementing the Second Order Method

In order to obtain a numerical solution to

$$\ddot{x}(t)+f(t,x(t),x(t-r),\dot{x}(t),\dot{x}(t-r)) = 0, \quad r>0 \tag{8}$$

on $[0,r]$, some modifications of the discrete method are needed. The first modification is based on the fact that on $[0,r]$ we have more information available to us concerning the behavior of the theoretical solution than we do later.

To obtain the theoretical solution to (8) on $[0,r]$ we start with a given initial function $\varphi(t)$ whose derivative $\varphi'(t)$ is known on $[-r,0]$. The extra information we have in this case is knowledge of $\varphi'(t)$. This is incorporated into the discrete version of (8) as added initial conditions.

Let $R>0$ be any given natural number and define $h=r/R$. Construct the uniform grid $\{t_i=0+ih\}_{i=-R}^{R}$. Since we are given $\varphi(t)$ and $\varphi'(t)$ for $t\in[-r,0]$ we may define a discrete version of (8) for $t\in[0,r]$ by

$$\frac{X_{i+1}-2X_i+X_{i-1}}{h^2}+f\left(t_i,X_i,\varphi_{i-R},\frac{X_{i+1}-X_{i-1}}{2h},\varphi'_{i-R}\right) = 0, \quad i=0,1,\ldots,R-1;$$

$$\varphi_{i-R} = \varphi(t_{i-R}) \equiv \varphi(t_i-r), \tag{49}$$

$$\varphi'_{i-R} = \varphi'(t_{i-R}) \equiv \varphi'(t_i-r), \quad i=0,1,\ldots,R-1;$$

$$X_{-1} = \varphi(t_{-1}) \equiv \varphi(-h),$$

$$X_0 = \varphi(t_0) \equiv \varphi(0).$$

The other modification that we must make is based on the fact that the theoretical solution to (8) will, in general, not have a continuous second derivative at the origin.

Since the expression $\frac{X_{i+1}-2X_i+X_{i-1}}{h^2}$ for $i=0$ in (49) is an approximation to the second derivative of the solution at $t=0$ we must take steps to insure that the second derivative exists and is continuous at $t=0$. However, we want to accomplish this in such a manner that the initial information X_0, X_{-1}, φ_{i-R} and φ'_{i-R} for $i=0,1,\ldots,R-1$ is not altered.

One way of accomplishing this is to modify the given initial function $\varphi(t)$. Select $\varepsilon < 0$ such that $-h < \varepsilon < 0$ and define a new initial function $\bar{\varphi}(t)$ by using an interpolating polynomial in place of $\varphi(t)$ on $[\varepsilon,0]$. More precisely, let $H(t)$ be the fourth degree Hermite interpolating polynomial which satisfies $P(\varepsilon) = \varphi(\varepsilon)$, $P'(\varepsilon) = \varphi'(\varepsilon)$, $P(0) = \varphi(0)$, $P'(0) = \varphi'(0)$ and $P''(0) = f(0,\varphi(0),\varphi(-r),\varphi'(0),\varphi'(-r))$. Define a new initial function $\bar{\varphi}(t)$ on $[-r,0]$ by

$$(50) \qquad \bar{\varphi}(t) = \begin{cases} \varphi(t) & -r \le t \le \varepsilon \\ H(t) & \varepsilon \le t \le 0. \end{cases}$$

By construction $\bar{\varphi}(t)$ will be continuously differentiable on $[-r,0]$ and since $\bar{\varphi}''(0) = f(0,\varphi(0),\varphi(-r),\varphi'(0),\varphi'(-r))$, the second derivative of the solution obtained with the initial

function $\bar{\varphi}$ will be continuous at $t=0$. Now $\bar{\varphi}(t)$ agrees with $\varphi(t)$ for $-r \leq t \leq \epsilon$ and also for $t=0$. This includes all points in $[-r,0]$ at which initial information is needed in (49). Since, as mentioned earlier, the solution to (8) depends continuously on the initial function, the use of $\bar{\varphi}(t)$ in place of $\varphi(t)$ will not radically alter the character of the solution to (8). It will however make (49) a reasonable discretization for (8).

Note that, as we move to the right solving (8) with the method of steps, we gain differentiability of the solution (see the discussion for first order equations in Section 1 of this chapter). Thus the solution to (8) will have a continuous second derivative for all $t > 0$ and the approximation $\frac{X_{i+1}-2X_i+X_{i-1}}{h^2}$ will not cause any difficulties at other multiples of r.

The proof that the discrete problem (49) is consistent is a special case of Section 2.1. The only difference is that now we are using the actual value of $\varphi'(t-r)$ instead of an approximation. This will simplify the expression for the local truncation error but does not alter the fact that the truncation error has an asymptotic expansion in even powers of h. Also, note that the proof of stability with the added hypotheses (33), and the analysis of the global discretization error given for (9) carry over directly to (49).

Using the appropriate discrete scheme, (9) or (49), we can compute a solution to (8) whose global discretization error has an asymptotic expansion in even powers of h. Since the coefficient error functions in this expansion are independent of h, extrapolation can be performed to obtain a more accurate solution.

Let R be any natural number and let $h_0 = r/R$ be the basic steplength. For each $k=1,\ldots,M$ define $h_k = h_0/2^k = \frac{r}{2^k R}$. Define grids G_k by

$$G_k = \{t_i^k : t_i^k = 0+ih_k, i=-2^kR,-2^kR+1,\ldots,0,1,\ldots,2^kR\} \text{ for } k=0,1,\ldots M.$$

As before, the grids G_k are nested; $G_k \subset G_{k+1}$.

Given an initial function $\varphi(t)$ such that $\varphi(t)$ and $\varphi'(t)$ are continuous on $[-r,0]$, select $\varepsilon < 0$ such that $-h_M < \varepsilon < 0$. With this ε define a new initial function, again denoted by φ, as in (50).

On each grid G_k compute the solution $X(h_k)$ to (49). Denoting the solution to (8) on $[0,r]$ also by $\varphi(t)$ we know from our previous work that

$$(51) \qquad [X(h_k)]_i = \varphi(t_i^k) + \sum_{j=1}^{M} h^{2j} e_j(t_i^k) + \mathcal{O}(h_k^{2M+2}), \text{ for } i=1,\ldots,2^kR.$$

If $R=1$, then we can use pullback interpolation as described in Chapter 1 to obtain a solution, $X(h)$, which satisfies

$$(52) \qquad [X(h)]_i = \varphi(t_i) + \mathcal{O}(h_O^{2M+2})$$

for each $t_i \in G_M$. Actually the pullback interpolation method does not involve as much work in this case as it does for first order initial value problems. This is so because from (48') we have $e_j^{\prime}(0) = 0$ for $j=1,\ldots,M$. Therefore we do not have to go through all the work of constructing a table analogous to Table 4 of Chapter I to obtain $e_j^{\prime}(0)$.

If $R \geq 2$, then we can obtain uniform accuracy at all grid points of G_M without using any information about $e_j^{\prime}(0)$. The reason for this is that with $R \geq 2$ we have enough data points at each stage to do Lagrange interpolation with accuracy comparable to that of the data points. That is, when performing Lagrange interpolation to accomplish the pullback with $R \geq 2$, it is the accuracy to which we know the error functions that dominates (see the discussion in Section 5 of Chapter 1). Thus we do not need to do Hermite interpolation in this case and therefore the information about $e_j^{\prime}(0)$ is superfluous.

Once the solution $X(h)$, satisfying (52), is obtained we can use the discrete method (9) to obtain a numerical solution on $[r,2r]$. This process, theoretically at least, can be repeated indefinitely on intervals of length r, to obtain a solution which is $\mathcal{O}(h^{2M+2})$.

It should be noted however, that when using (9) to solve (8) on intervals of the form $[(L-1)r, Lr]$, $L \geq 1$, the initial data is taken from the computed solution on the previous interval. Since the initial data will not be precise (the error in the initial data is $\mathcal{O}(h^{2M+2})$) an accumulation or errors is unavoidable. Of course, this also occurs when any numerical method is used on successive intervals.

A variety of other schemes, based on (9), (48) and pullback interpolation, to solve (8) are possible. We could, for instance, use (49) and pullback interpolation on $[0, h_0]$ to obtain a solution. Then by modifying the initial concitions in (49) we could repeat this procedure on $[h_0, 2h_0]$. Repeating this R times we would have the solution on all of $[0, r]$. We could then follow the same format, using (9) in place of (49), to obtain a solution on $[r, 2r]$, etc.

We do not make a comparison of these schemes here. In the next section we will merely exhibit some numerical examples computed using the first described solution technique.

Section 2.5. Numerical Results for Second Order Equations

This section consists of two examples. The first of these is

Example 3: Solve $\ddot{x}(t) = x(t-1)$, $0 \leq t \leq 1$ numerically for the initial function $\varphi(t) = 12(t+1)^2 + 24$ on $[-1,0]$. The method of solution is the discretization (49), extrapolation and pullback interpolation with four grids. The steplengths employed are $h_k = 1/2^k$ for $k=0,\ldots,3$ and the grids are

$$G_k = \{t_i^k = 0 + ih_k: \quad i=0,1,\ldots,2^k\}$$

for $k=0,\ldots,3$.

Note that the initial function $\varphi(t)$ satisfies $\varphi(-1) = \varphi''(0-)$, so the second derivative of the solution is continuous at the origin. Thus, the initial function does not need to be modified as in (50) for this example.

The theoretical solution is given by

$$\varphi(t) = t^4 + 12t^2 + 24t + 36$$

for $t \in [0,1]$. The numerical results presented in Table 15 are for the nine equally spaced points in $[0,1]$ which belong to the finest grid G_3. The error reported is once again the computed solution minus the theoretical solution.

TABLE 15

i	Error
0	0.0
1	0.0×10^{-12}
2	$-.23 \times 10^{-12}$
3	$-.23 \times 10^{-12}$
4	$-.23 \times 10^{-12}$
5	$-.23 \times 10^{-12}$
6	$-.23 \times 10^{-12}$
7	0.0×10^{-12}
8	0.0×10^{-12}

The high accuracy of the numerical solution in this example is due to the fact that the theoretical solution is a low degree polynomial. Thus, extrapolation and consequently pullback interpolation are quite accurate in this case. This accuracy is not to be expected for more general equations as the following example illustrates.

Example 4: Solve $\ddot{x}(t) + \frac{1}{2}x(t) - \frac{1}{2}x(t-\pi) = 0$ on $[0,\pi]$ for the initial function $\varphi(t) = 1-\sin t$, $-\pi \leq t \leq 0$. This example is from Norkin [23] and the theoretical solution is also given by $\varphi(t) = 1-\sin t$, $0 \leq t \leq \pi$. Thus, the second derivative of the solution is continuous at the origin and no modification of the initial function is required.

The method of solution is to use (49), extrapolation and pullback interpolation with four grids to obtain a numerical solution on $[0,\pi/4]$. Using this numerical solution as initial data, the discretization (9), extrapolation and pullback interpolation with four grids is employed to obtain a solution on $[\frac{\pi}{4},\frac{\pi}{2}]$. This process is repeated twice more to obtain numerical solutions on $[\frac{\pi}{2},\frac{3\pi}{4}]$ and $[\frac{3\pi}{4},\pi]$. On each interval of length $\pi/4$ the stepsizes used are $h_k = \frac{\pi}{4\cdot 2^k}$ for $k=0,\ldots,3$. Thus, on each of the four subintervals the finest grid contains nine equally spaced points which we number as $t_0,\ldots,t_8$.

The accuracies of the numerical results are reported as Table 16, with the error being computed in the usual manner.

TABLE 16

i	Error on $[0,\pi/4]$	Error on $[\pi/4,\pi/2]$
0	0.0	$.5 \times 10^{-8}$
1	15.9×10^{-8}	-1.0×10^{-6}
2	37.6×10^{-8}	-4.5×10^{-6}
3	30.2×10^{-8}	-6.4×10^{-6}
4	-1.6×10^{-8}	-6.6×10^{-6}
5	-33.2×10^{-8}	-6.7×10^{-6}
6	-40.0×10^{-8}	-8.4×10^{-6}
7	-17.1×10^{-8}	-11.5×10^{-6}
8	$.5 \times 10^{-8}$	-11.9×10^{-6}

i	Error on $[\pi/2,3\pi/4]$	Error on $[3\pi/4,\pi]$
0	-11.9×10^{-6}	-27.2×10^{-6}
1	-13.7×10^{-6}	-28.6×10^{-6}
2	-18.7×10^{-6}	-31.9×10^{-6}
3	-21.2×10^{-6}	-33.2×10^{-6}
4	-21.0×10^{-6}	-32.4×10^{-6}
5	-20.6×10^{-6}	-31.4×10^{-6}
6	-22.6×10^{-6}	-32.0×10^{-6}
7	-26.6×10^{-6}	-34.4×10^{-6}
8	-27.2×10^{-6}	-34.1×10^{-6}

The accuracy of the computed solution is considerably higher on the first subinterval than on the last three subintervals. Also the errors are steadily increasing at a very low rate over the last three subintervals. This suggests an accumulation of truncation errors.

This accumulation is to be expected for the following reason. For a linear equation, the discretizations (9) or (49) are similar to a well known linear multistep method for solving second order initial value problems (see Lambert [18]). The only difference is the manner in which the discretizations are initialized. The stability theory for linear multistep methods for solving second order equations predicts an accumulation of truncation errors. Thus, based on comparisons with similar methods, our numerical results are consistent with how our method should behave.

Note that the equation $\ddot{x}(t) + \frac{1}{2}x(t) - \frac{1}{2}x(t-\pi) = 0$ is not covered by our proof of stability. Yet the numerical results seem to indicate that the method is stable when applied to this equation. Consequently, we again reiterate our belief that stability is valid for a much larger class of functions than our proof indicates.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Aitken, A.C., On interpolation by iteration of proportional parts, Proc. Edinburgh Math. Soc., 2, 56-76 (1932).

2. Björck, A., and V. Pereyra, Solutions of Vandermonde systems of equations, Math. Comp., 24, 893-903 (1970).

3. Cooke, K.L., and S.E. List, The numerical solution of integro-differential equations with retardation, Department of Electrical Engineering, University of Southern California, Los Angeles, Technical Report No. 72-4 (1972).

4. Cooke, K.L. and J.A. Yorke, Equations modelling population growth, economic growth and gonorrhea epidemiology, preprint.

5. Corliss, G.F., Parallel Rootfinding Algorithms, Ph.D. thesis, Michigan State University, East Lansing, Michigan (1974).

6. Cullen, C.G., Matrices and Linear Transformations, Addison-Wesley Publishing Company, Reading, Mass. (1967).

7. Cunningham, W.J., A non-linear differential-difference equation of growth, Proc. Natl. Acad. Sci., U.S., 40, 709-713 (1954).

8. Dahlquist, G., Convergence and stability in the numerical integration of ordinary differential equations, Kungl. Tekniska Hogskolans Handlingar No. 130 (1959).

9. Dahlquist, G., A special stability problem for linear multistep methods, BIT, 3, 27-43 (1963).

10. Feldstein, A., Discretization Methods for Retarded Ordinary Differential Equations, Ph.D. thesis, University of California, Los Angeles (1964).

11. Gragg, W.B., Repeated Extrapolation to the Limit in the Numerical Solution of Ordinary Differential Equations, Ph.D. thesis, University of California, Los Angeles (1964).

12. Gragg, W.B., On extrapolation algorithms for ordinary initial value problems, SIAM J. Numer. Anal., 2, 384-403 (1965).

13. Hale, J., Functional Differential Equations, Springer-Verlag, New York, N.Y. (1971).

14. Huygens, C., De Circuli Magnitudine Inventa, Leiden (1654).

15. Isaacson, E., and H.B. Keller, Analysis of Numerical Methods, John Wiley and Sons, Inc., New York, N.Y (1966).

16. Joyce, D.C., Survey of extrapolation processes in numerical analysis, SIAM Review, 13, 435-490 (1971).

17. Keller, H.B., Numerical Methods for Two-Point Boundary-Value Problems, Blaisdell Publishing Company, Waltham, Massachusetts (1968).

18. Lambert, J.D., Computational Methods in Ordinary Differential Equations, John Wiley and Sons, London (1973).

19. Lindberg, G., A simple interpolation algorithm for improvement of the numerical solution of a differential equation, SIAM J. Numer. Anal., 9, 662-668 (1972).

20. Milne, W.E., and R.R. Reynolds, Stability of a numerical solution of differential equations, J. Assoc. Comput. Mach., 6, 193-203 (1959); Part II, J. Assoc. Comput. Mach., 7, 46-56 (1960).

21. Milne, W.E., and R.R. Reynolds, Fifth-order methods for the numerical solution of ordinary differential equations, J.Assoc. Comput. Mach., 9, 64-70 (1962).

22. Neville, E.H., Iterative interpolation, J. Ind. Math. Soc., 20, 87-120 (1934).

23. Norkin, S.B., Differential Equations of the Second Order with Retarded Argument, American Mathematical Society, Providence, Rhode Island (1972).

24. Pareyra, V., On improving an approximate solution of a functional equation by deferred corrections, Numer. Math., 8, 376-391 (1966).

25. Pereyra, V., Iterated deferred corrections for nonlinear operator equations, Numer. Math., 10, 316-323 (1967).

26. Pereyra, V., Accelerating the convergence of discretization algorithms, SIAM J. Numer. Anal., 4, 508-533 (1967).

27. Pereyra, V., Iterated deferred corrections for nonlinear boundary value problems, Numer. Math., 11, 111-125 (1968).

28. Pereyra, V., High order finite difference solution of differential equations, Seminar Notes, Stanford University, Palo Alto, California (1973).

29. Richardson, L.F., The deferred approach to the limit, I-single lattice, Trans. Roy. Soc. London, 226, 299-349 (1927).

30. Sansone, G., Teorema di esistenza di soluzioni per un sistema di equazioni funzionali differenziali, Ann. Mat. Pura Appl., 39, 65-67 (1955).

31. Stetter, H.J., Asymptotic expansions for the error of discretization algorithms for non-linear functional equations, Numer. Math., 7, 18-31 (1965).

32. Varga, R.S., Matrix Iterative Analysis, Prentice-Hall, Inc., Englewood Cliffs, New Jersey (1962).

33. Widder, D.V., Advanced Calculus, Prentice-Hall, Inc., Englewood Cliffs, New Jersey (1961).

34. Wright, E.M., A non-linear difference-differential equation, J. reine angew. Math., 194, 66-87 (1955).

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) UNIFORMLY ACCURATE NUMERICAL SOLUTIONS TO DIFFERENTIAL EQUATIONS USING EXTRAPOLATION AND INTERPOLATION.		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard Allan Rogers		8. CONTRACT OR GRANT NUMBER(s) AFOSR-72-2271
9. PERFORMING ORGANIZATION NAME AND ADDRESS Michigan State University Department of Mathematics East Lansing, Michigan 48824		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research 1400 Wilson Blvd. Arlington, Virginia 22209		12. REPORT DATE August 1974
		13. NUMBER OF PAGES 154
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In this work we are concerned with numerical methods for solving ordinary differential equations. We consider those methods that have asymptotic error expansions involving all powers of h^q, where h is the steplength and q is a fixed integer. The process of extrapolation can be employed with such methods to obtain highly accurate solutions at grid points belonging to the coarsest mesh. In Chapter I we develop the

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

"pullback interpolation method". This method combines extrapolation with Hermite interpolation of the coefficient functions for the asymptotic error expansion to produce a highly accurate solution at all grid points of the finest mesh. When q is 1 or 2 pullback interpolation yields uniform accuracy at all grid points of the finest mesh.

In Chapter II the pullback interpolation method is modified so as to be applicable to boundary value problems. In addition, an elementary proof of the stability of V. Pereyra's finite difference scheme for solving two point boundary value problems is given.

In Chapter III we consider difference differential equations with constant retardation. The methods of Chapter I are shown to be applicable to first order delay equations. Because of the presence of the delay term, the uniform accuracy obtained through pullback interpolation is indispensible for these problems.

A finite difference scheme for directly solving second order delay equations is constructed and analyzed in Chapter III. The global discretization error is shown to have an asymptotic error expansion in even powers of the steplength.