SOME INVESTIGATIONS INTO THE NUMERICAL SOLUTION OF INITIAL VALUE PROBLEMS IN ORDINARY DIFFERENTIAL EQUATIONS

By

G. N. HAYDEN

Ph.D. Thesis

March 1976

University of Newcastle upon Tyne

Acknowledgements

I would like to express my sincerest appreciation to my supervisor Dr. B. Shaw for his guidance and supervision throughout the preparation of this thesis.

Thanks are also due to Dr. K. Wright and Mr. T. Penna for their many helpful comments and suggestions over the same period.

I would also like to thank Mrs. N. Dowson and Miss J. Watson for their careful typing of this thesis, and Mrs. M. Webster for duplicating copies of this thesis.

This research was supported by the Science Research Council, by the University of Newcastle, and by Newcastle Polytechnic.

Abstract

In this thesis several topics in the numerical solution of the initial value problem in first-order ordinary differential equations are investigated.

Consideration is given initially to stiff differential equations and their solution by stiffly-stable linear multistep methods which incorporate second derivative terms. Attempts are made to increase the size of the stability regions for these methods both by particular choices for the third characteristic polynomial and by the use of optimization techniques while investigations are carried out regarding the capabilities of a high order method.

Subsequent work is concerned with the development of Runge-Kutta methods which include second-derivative terms and are implicit with respect to y rather than k. Methods of order three and four are proposed which are L-stable.

The major part of the thesis is devoted to the establishment of recurrence relations for operators associated with linear multistep methods which are based on a non-polynomial representation of the theoretical solution. A complete set of recurrence relations is developed for both implicit and explicit multistep methods which are based on a representation involving a polynomial part and any number of arbitrary functions. The amount of work involved in obtaining multistep methods by this technique is considered and criteria are proposed to decide when this particular method of derivation should be employed.

The thesis is concluded by using Prony's method to develop one-step methods and multistep methods which are exponentially adaptive and as such can be useful in obtaining solutions to problems which are exponential in nature.

Contents

Chapter One

Introduction

The behaviour patterns of many physical systems are governed by differential equations. These equations may be either ordinary or partial differential equations and in general it is impossible to obtain an analytical solution for them. One way of obtaining an approximate solution is to make use of numerical methods. In this thesis we will be concerned solely with the numerical solution of initial-value problems for first-order ordinary differential equations. This type of problem occurs commonly in areas such as chemistry [Robertson (1966)], electrical engineering [Calahan (1968)], astronomy [Bettis and Stiefel [(1969)], and biology [Cooper (1969)].

1.1 Initial-Value Problems for First-Order Ordinary Differential Equations

The problem is described by the equation

$$y' = f(x,y) \qquad y(a) = y_0. \qquad (1.1)$$

We will attempt to find a numerical solution to this problem at various points in the interval $[a, b]$.

Note that we are effectively restricting the problem since not all first-order ordinary differential equations can be rearranged into the form (1.1). Before attempting to find a solution (theoretical or numerical) of an initial value problem it is important to know whether a solution does exist and whether this solution is unique. The following theorem which is given in Henrici (1962) guarantees the existence of a unique solution to the problem (1.1).

Theorem 1.1

If (i) $f(x,y)$ is defined and continuous for $a \leq x \leq b$,

$-\infty < y < \infty$ where a and b are finite and

(ii) $\exists$ a constant $L(\geq 0)$ such that for $x\epsilon[a,b]$ and any two numbers y_1, y_2

$$|f(x,y_1) - f(x,y_2)| \leq L|y_1-y_2| \tag{1.2}$$

then $\exists$ a function y satisfying

(a) y is continuous and differentiable for $x\epsilon[a,b]$

(b) $y'(x) = f(x,y(x))$, $x\epsilon[a,b]$

(c) $y(a) = y_0$

(d) y is a unique solution of the initial-value problem.

The condition (1.2) is known as a Lipschitz condition and the constant L as a Lipschitz constant. The hypothesis that f(x,y) be defined and continuous in the infinite interval $-\infty < y < \infty$ is very restrictive. Many functions f(x,y) are only continuous over a finite interval in y. In these circumstances we can employ a modified version of Theorem 1.1 as given by Burkill (1962) to guarantee a unique solution to the problem (1.1).

1.2 Numerical Solution of the Initial-Value Problem

To obtain a numerical solution the problem is discretized, that is, we define the mesh points $x_0, x_1, x_2, \ldots, x_N$ by $x_0 = a$, $x_n = x_0 + nh$ $(n=1, 2, \ldots N)$, $x_N = b$ and the parameter h, which for the moment is regarded as constant, is called the step length. A numerical solution is then sought on this discrete point set $\{x_n | n = 0, 1, 2, \ldots, N\}$.

The exact value of the theoretical solution at the mesh point x_n is denoted by $y(x_n)$ and the numerical solution at this point by y_n. In addition we define the numerical solution f_n which is an approximation to the first derivative of y(x) at $x = x_n$ by

$$f_n = f(x_n,y_n) \tag{1.3}$$

The two numerical methods used most frequently to obtain approximate solutions to the first-order initial-value problem are linear multistep methods and Runge-Kutta methods. In the following two sections we outline the background theory behind these methods so that we can refer to it when necessary in later chapters.

1.3 Linear Multistep Methods

If the numerical values y_{n+j}, f_{n+j}, $j = 0, 1, 2, \ldots, k$ are combined linearly in the form

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j} \qquad n = 0, 1, 2, \ldots$$

$$\alpha_k \neq 0 \quad |\alpha_0| + |\beta_0| > 0 \tag{1.4}$$

where the α_j and β_j are constants then (1.4) is called a linear multistep method with stepnumber k. Since (1.4) is effectively unaltered by multiplying throughout by a constant, the coefficients α_j, β_j are arbitrary to the extent of a constant multiplier. One commonly used convention which we will follow unless otherwise stated is to normalize (1.4) by setting $\alpha_k = 1$. The differential equation (1.1) is thus replaced by a difference equation and we attempt to find a numerical solution of the differential equation by calculating a sequence $\{y_n\}$ which is a solution of the difference equation (1.4) for the starting values $y_0, y_1, \ldots, y_{k-1}$.

The difference equation is non-linear and may not have a solution while even if it does possess a solution this may not be unique. When $\beta_k = 0$ however the linear multistep method is called explicit and (1.4) can then be rearranged into the form

$$y_{n+k} = [-\alpha_{k-1}y_{n+k-1} - \ldots - \alpha_0 y_n + h(\beta_{k-1}f_{n+k-1} + \ldots + \beta_0 f_n)]$$

and this has a unique solution since the right hand side does not depend on y_{n+k}. When $\beta_k \neq 0$ the linear multistep method is called implicit and (1.3) can then be arranged as

$$y_{n+k} = h\beta_k f_{n+k} + C \tag{1.5}$$

where

$$C = [-\alpha_{k-1}y_{n+k-1} - \ldots - \alpha_0 y_n + h(\beta_{k-1}f_{n+k-1} + \ldots + \beta_0 f_n)] \tag{1.6}$$

Since $f_{n+k} = f(x_{n+k}, y_{n+k})$, y_{n+k} occurs on both sides of (1.5) which in general is a non-linear equation for y_{n+k} and may have one, several or no solutions. Rewriting (1.5) as

$$y_{n+k} = F(y_{n+k}) \tag{1.7}$$

where

$$F(y_{n+k}) = h\beta_k f_{n+k} + C \tag{1.8}$$

and iterating this formula by

$$y^{(\nu+1)} = F(y^{(\nu)}) \qquad \nu = 0, 1, 2, \ldots \tag{1.9}$$

where $y^{(0)}$ is a suitable first approximation then the following theorem as given in Henrici (1962) guarantees the existence of a unique solution to (1.7) and the convergence of the iterative scheme (1.9) to this solution.

Theorem 1.2

Let the function $F(y)$ be defined for $-\infty < y < \infty$ and let there exist a constant K such that $0 \leq K < 1$ and

$$|F(y^*) - F(y)| \leq K|y^*-y| \tag{1.10}$$

for arbitrary y^* and y. Then the following hold

(i) equation (1.7) has a unique solution y

(ii) for arbitrary $y^{(0)}$ the sequence defined by (1.10) converges to y.

If $F(y)$ is defined by (1.8) and, if $f(x,y)$ satisfies a

Lipschitz condition with respect to y and possesses a Lipschitz constant L, then (1.10) is satisfied with

$$K = Lh|\beta_k| \tag{1.11}$$

which is less than one for small enough h.

We also require that the unique solution $\{y_n\}$ generated by the multistep method 'converges' to the theoretical solution $y(x)$ as the steplength h tends to zero. Before formally defining convergence we state certain definitions which will be required later.

Definition 1.1

The associated linear difference operator L is defined by

$$L[z(x);h] = \sum_{j=0}^{k} [\alpha_j z(x+jh) - h\beta_j z'(x+jh)] \tag{1.12}$$

where $z(x)$ is an arbitrary test function infinitely differentiable on $[a,b]$.

It would be preferable to use the notation $(L_h z)(x)$ since we are discussing the application of an operator to a function but we follow the commonly used notation in this particular field of research and use $L[z(x);h]$.

Expanding $z(x+jh)$ and $z'(x+jh)$ by Taylor series and collecting terms gives

$$L[z(x);h] = \sum_{r=0}^{\infty} C_r h^r z^r(x) \tag{1.13}$$

where

$$\left.\begin{aligned} C_0 &= \sum_{j=0}^{k} \alpha_j \\ C_r &= \frac{1}{r!} \sum_{j=0}^{k} (j^r \alpha_j - r j^{r-1} \beta_j) \quad r = 1, 2, \ldots \end{aligned}\right\}. \tag{1.14}$$

Definition 1.2

The difference operator and associated linear multistep method are said to be of order p if in (1.13)

$$C_0 = C_1 = C_2 = \ldots = C_p = 0 \qquad \text{but } C_{p+1} \neq 0. \tag{1.15}$$

Notice that by using an arbitrary test function which is infinitely differentiable we can define the order of a linear multistep method without making any assumptions on the number of derivatives possessed by the theoretical solution.

Definition 1.3

The local error at x_{n+k} of the linear multistep method is defined to be

$$L[y(x_n);h] = \sum_{j=0}^{k} \alpha_j y(x_n+jh) - h \sum_{j=0}^{k} \beta_j y'(x_n+jh). \tag{1.16}$$

The local error at x_{n+k} is often denoted by T_{n+k}. If we now assume that the theoretical solution $y(x)$ is sufficiently differentiable and that no previous local errors have been made, that is

$$y_{n+j} = y(x_{n+j}) \qquad j = 0, 1, \ldots, k-1 \tag{1.17}$$

then it can be shown that for linear multistep methods of order p

$$y(x_{n+k}) - y_{n+k} = C_{p+1} h^{p+1} y^{(p+1)}(x_n) + O(h^{p+2}) \tag{1.18}$$

where the left hand side of (1.18) is equal to T_{n+k}. C_{p+1} and $C_{p+1}h^{p+1}y^{(p+1)}(x_n)$ are called the error constant and principal local error respectively.

Definition 1.4

The global error is defined as

$$e_{n+k} = y(x_{n+k}) - y_{n+k} \tag{1.19}$$

where no localizing assumptions are made and roundoff errors are ignored.

Definition 1.5

Associated with the linear multistep method are the first characteristic polynomial given by

$$\rho(\xi) = \sum_{j=0}^{k} \alpha_j \xi^j \tag{1.20}$$

and the second characteristic polynomial given by

$$\sigma(\xi) = \sum_{j=0}^{k} \beta_j \xi^j . \tag{1.21}$$

We now formally define the convergence of a linear multistep method by

Definition 1.6

The linear multistep method (1.4) is said to be convergent if for all initial-value problems subject to the hypotheses of Theorem 1.1

$$\lim_{\substack{h \to 0 \\ n \to \infty \\ nh = x-a}} y_n = y(x_n)$$

holds for all $x \in [a,b]$ and for all solutions $\{y_n\}$ of the difference equation (1.3) which satisfy the starting conditions

$$y_\mu = \eta_\mu(h)$$

where

$$\lim_{h \to 0} \eta_\mu(h) = y_0 \qquad \mu = 0, 1, 2, \ldots, k-1.$$

This definition means that when $k > 1$ the extra starting values required need not be exact solutions of the initial-value problem (1.1) at the appropriate values for x but only that regarded as functions of h they must all tend to the given initial value y_0 as $h \to 0$.

To obtain convergence we have to control both the magnitude of the local error and the way in which this error is propagated step by step as the calculation proceeds. This is achieved if the method satisfies the following two conditions called consistency and zero-stability.

Definition 1.7 (Consistency)
The linear multistep method (1.4) is said to be consistent if the order of the associated linear difference operator (1.12) is at least one, that is, $p \geq 1$ and hence $C_0 = C_1 = 0$.

Definition 1.8 (Zero-Stability)
The linear multistep method (1.4) is said to be zero-stable if no root of the first characteristic polynomial $\rho(\xi)$ has modulus greater than one and if every root of modulus one is simple.

Consistency ensures that the multistep method approximates the given differential equation while also attempting to keep the magnitude of the local error small at each step. Zero-stability ensures that any extra solutions of the difference equation for y_n (parasitic solutions), which occur because a first-order differential equation is being replaced by a higher order difference equation, will be damped out in the limit as $h \to 0$. Zero-stability thus attempts to control the way in which local errors are propagated. Convergence is then guaranteed by the following theorem due to Dahlquist (1956).

Theorem 1.3
The necessary and sufficient conditions for a linear multistep method to be convergent are that it be both consistent and zero-stable.

In addition to the method being convergent we would like it to have as high an order as possible to reduce the local error. The general k-step linear multistep method (1.4) has 2k+2 coefficients α_j, β_j, $j = 0, 1, \ldots, k$. We have normalized by setting $\alpha_k = 1$ so for an implicit method there are therefore $2k + 1$ free parameters while for an explicit method there are only $2k$ parameters. For an implicit method we might hope to achieve order $2k$ by requiring $C_0 = C_1 = \ldots = C_{2k} = 0$ thus giving us a set of $2k + 1$ linear equations for the same number of unknowns $\alpha_0, \ldots, \alpha_{k-1}, \beta_0, \ldots \beta_k$. Unfortunately the solution of these equations gives a method which is not zero-stable. The following theorem, again due to Dahlquist (1956), gives us a bound on the maximum order attainable by a zero-stable method.

Theorem 1.4

No zero-stable linear multistep method of stepnumber k can have order exceeding $k + 1$ when k is odd, or exceeding $k + 2$ when k is even. A necessary and sufficient condition for p to equal $k + 2$ is that k be even and all zeros of $\rho(\xi)$ lie on the unit circle in the complex plane.

It is common practice to choose an implicit formula of order $k + 1$. Implicit formulae have in general a much smaller local error than an explicit method of the same order and thus they are preferable even though more computation is involved. At each step however an implicit formula needs a starting value for $y_{n+k}^{(0)}$ and one way of obtaining this is to use an explicit formula to 'predict' a value for $y_{n+k}^{(0)}$. The implicit formula is then used

iteratively to 'correct' the predicted value and the combination is known as a Predictor-Corrector method.

1.4 Runge-Kutta Methods

The general one-step explicit method is defined as

$$y_{n+1} - y_n = h\Phi(x_n,y_n;h) \tag{1.22}$$

where $\Phi(x,y;h)$ is called the increment function and depends not only on x and y but also on the stepsize h and the differential equation.

We introduce the exact relative increment Δ by

$$\Delta(x,z^*;h) = \begin{cases} \dfrac{z(x+h) - z(x)}{h} & h \neq 0 \\ f(x,z^*) & h = 0 \end{cases} \tag{1.23}$$

where z satisfies the same differential equation as the theoretical solution y(x) but has a different initial condition $z(x) = z^*$.

Order, local error, global error and consistency are then defined in the following manner.

Definition 1.9

The method (1.22) is said to have order p if p is the largest integer for which

$$\Phi(x,z^*;h) - \Delta(x,z^*;h) = O(h^p). \tag{1.24}$$

Definition 1.10

The local error, T_{n+1}, at x_{n+1} of the method (1.22) is given by

$$T_{n+1} = y(x_{n+1}) - y(x_n) - h\Phi(x_n,y(x_n);h). \tag{1.25}$$

If we make the localizing assumption that $y_n = y(x_n)$ then

$$T_{n+1} = y(x_{n+1}) - y_{n+1} \tag{1.26}$$

while if in addition we assume that the theoretical solution

$y(x)$ is sufficiently differentiable then

$$T_{n+1} = \psi(x_n, y(x_n))h^{p+1} + O(h^{p+2}) \tag{1.27}$$

where $\psi(x,y)$ and $\psi(x_n, y(x_n))h^{p+1}$ are known as the principal error function and principal local error respectively.

Definition 1.11

The global error e_{n+1} of the method (1.22) at x_{n+1} is defined by

$$e_{n+1} = y(x_{n+1}) - y_{n+1} \tag{1.28}$$

where no localizing assumptions are made and roundoff errors are ignored.

Definition 1.12

The method (1.22) is said to be consistent if

$$\Phi(x,y;0) = f(x,y). \tag{1.29}$$

The following theorem as given in Henrici (1962) guarantees the convergence of (1.22).

Theorem 1.5

If (i) the function $\Phi(x,y;h)$ is continuous jointly as a function of its three arguments, in the region D defined by $x \in [a,b]$, $y \in (-\infty,\infty)$, $h \in [0,h_o]$ where $h_o > 0$

(ii) $\Phi(x,y;h)$ satisfies a Lipschitz condition of the form

$$|\Phi(x,y^*;h) - \Phi(x,y;h)| \leq M|y^*-y| \tag{1.30}$$

then the method (1.22) is convergent if and only if it is consistent.

Since the difference equation (1.22) is first-order then it has no parasitic solutions. There is therefore no need for zero-stability and so only consistency is required to ensure convergence.

If $f(x,y)$ is assumed to be sufficiently differentiable then by a Taylor series expansion we have

$$y(x+h) - y(x) = h\left[f+\frac{h}{2!}f^{(1)} + \frac{h^2}{3!}f^{(2)} + \frac{h^2}{4!}r^{(3)} + \ldots \right.$$
$$\left. \ldots + \frac{h^{p-1}}{p!}f^{(p-1)} + O(h^p)\right] \qquad (1.31)$$

where f and its derivatives $f^{(1)}, \ldots, f^{(p-1)}$ have arguments $(x,y(x))$.

One choice for the increment function is

$$\Phi(x,y;h) = f(x,y)+ \frac{h}{2!}f^{(1)}(x,y) + \frac{h^2}{3!}f^{(2)}(x,y)+\ldots+ \frac{h^{p-1}}{p!}f^{p-1}(x,y) \qquad (1.32)$$

This is known as a Taylor series method of order p. To apply this method it is necessary to calculate not only f but also all the higher derivatives $f^{(1)}, f^{(2)}, \ldots f^{(p-1)}$. Obtaining the necessary higher derivatives is usually complex and laborious. Hence we look at an alternative choice for Φ which does not require derivatives of f but is a weighted combination of values of f where the weights and the points at which f is evaluated are chosen to make the order as high as possible.

The general R-stage explicit Runge-Kutta method is thus defined by taking Φ to be of the form

$$\Phi(x,y;h) = \sum_{r=1}^{R} c_r k_r \qquad (1.33)$$

where

$$k_r = f(x+ha_r, y+h\sum_{s=1}^{R} b_{rs}k_s) \qquad r = 1,2,\ldots R \qquad (1.34)$$

and

$$a_r = \sum_{s=1}^{R} b_{rs} \qquad r = 1,2,\ldots R. \qquad (1.35)$$

These methods fall into two sub-classifications. If $b_{rs} = 0$ for $r \leq s$ then the functions k_r are defined explicitly. This type of method will be called k-explicit. Butcher (1965)

has shown that for a k-explicit method the maximum order p(R) which can be obtained by an R-stage method is given by

$$\left.\begin{aligned} p(R) &= R && R = 1,2,3,4 \\ p(R) &= R-1 && R = 5,6,7 \\ p(R) &= R-2 && R = 8,9 \\ p(R) &\leq R-2 && R \geq 10. \end{aligned}\right\} \tag{1.36}$$

If $b_{rs} \neq 0$ for $r \leq s$ then the functions k_r are defined by a set of R implicit non-linear equations. This type of method will be called k-implicit. In another paper Butcher (1963) has shown that an R-stage k-implicit method can have order 2R and that the system of R implicit non-linear equations for the k's can be solved by using the iteration:-

$$k_r^{(\nu+1)} = f\left(x+ha_r, y+h\sum_{s=1}^{r-1} b_{rs}k_s^{(\nu+1)} + h\sum_{s=r}^{R} b_{rs}k_s^{(\nu)}\right)$$

$$\nu = 0,1,2,\ldots \quad r = 1,2,\ldots R \tag{1.37}$$

provided

$$h < 1/L(u+v) \tag{1.38}$$

where L is the Lipschitz constant of f with respect to y and

$$\left.\begin{aligned} u &= \max\left\{\sum_{j=1}^{R}|b_{1j}|, \sum_{j=2}^{R}|b_{2j}|, \sum_{j=3}^{R}|b_{3j}|, \ldots, \sum_{j=R}^{R}|b_{Rj}|\right\} \\ v &= \max\left\{\sum_{j=1}^{1}|b_{2j}|, \sum_{j=1}^{2}|b_{3j}|, \sum_{j=1}^{3}|b_{4j}|, \ldots, \sum_{j=1}^{R-1}|b_{Rj}|\right\} \end{aligned}\right\} \tag{1.39}$$

1.5 Contents of the Thesis

In chapter two we will present a short review of the most important papers published since 1955 on the numerical solution of the initial-value problem by linear multistep methods and Runge-Kutta methods. The aim of this review is

to show the different lines of investigation followed by the various researchers interested in these particular methods.

The third chapter contains an outline of the background stability theory for linear multistep methods. In particular the theory of weak stability is developed to illustrate the necessary requirements for errors in the numerical solution to be damped out when a non-zero stepsize is used. This is followed by a formal definition of 'stiff' systems of first-order ordinary differential equations. These are equations whose solution contains negative exponentials with widely differing time constants. Many multistep methods can only be used with a very small stepsize on this type of problem, even when the magnitude of these exponential terms is negligible, if the criteria imposed by weak stability are to be met. We then discuss the appropriate types of stability region necessary for such problems and suggest the use of multistep methods which are known as 'stiffly-stable.

In the fourth chapter we start with a discussion of a particular class of stiffly-stable multistep methods due to Gear (1968) and the modifications made by Jain and Srivastava (1970) to these methods by taking different choices for the second characteristic polynomial in order to increase the size of the stability region. It is then shown that when linear multistep methods are applied to stiff differential equations difficulties are encountered in using the normal corrector iteration and this leads to the use of a modified corrector iteration which incorporates $\partial f/\partial y$. Enright (1972) therefore proposed the use of stiffly stable multistep methods which employ not only values of y and f but also include the second derivative of y, $f^{(1)}$. (This

type of method will often subsequently be referred to as a second-derivative linear multistep method whereas the method (1.4) will be termed a first-derivative multistep method). We continue chapter four by discussing the work of Enright. A logical progression, comparable with that made to the first derivative methods of Gear, is to investigate the modifications that can be made to the class of second derivative methods proposed by Enright by suitable alterations to the third characteristic polynomial associated with second-derivative methods. This technique however, although it does allow us to increase the stability region of our stiffly-stable second-derivative methods, is rather limiting in the number of choices that can be made for the third characteristic polynomial and so we then move on to show how the problem of maximizing the size of the stability region can be viewed as a problem of constrained optimization. A method, based on an algorithm of Rosenbrock (1960), is outlined for determining solutions to this optimization problem and numerous results are given to illustrate the increases that can be obtained in the size of the stability region. Finally at the end of the chapter we investigate the possibility of obtaining second-derivative multistep methods having an order one higher than those proposed by Enright and carry out some limited numerical experiments to see how the higher order methods behave in comparison.

In chapter five the application of Runge-Kutta methods to solve stiff differential equations is considered and it is shown that implicit Runge-Kutta

methods suffer the same restrictions on the stepsize, due to weak stability and convergence of the iteration process, as do linear multistep methods. We then discuss the desirable stability regions for Runge-Kutta methods and suggest the use of methods which are 'L-stable'. In an effort to obtain L-stable methods we investigate the use of Runge-Kutta methods which include the second derivative of y and we propose third and fourth-order L-stable methods which are implicit with respect to y rather than the k's. We conclude the chapter by comparing this fourth-order method numerically with an alternative fourth-order method due to Butcher (1964).

The content of the sixth chapter is divorced from any of the work detailed in chapters three, four or five. In this chapter we consider the use of linear multistep methods based on non-polynomial interpolants rather than the conventional polynomial interpolant. It is shown how multistep methods based on a general non-polynomial interpolant can be derived by the use of recurrence relations. The feasibility of using this technique for the derivation is investigated and criteria, based mainly on the number of arithmetic operations involved, are presented for deciding when multistep methods should be derived by recurrence. We illustrate numerically the solution of several problems by multistep methods which are based on non-polynomial interpolants and have been derived by the recurrence relations.

In chapter seven we continue with methods based on non-polynomial interpolants. Continuously adaptive single-step methods and multistep methods, based on

applying non-linear exponential fitting to the solution, are derived by a technique which makes use of an idea originally proposed by Prony [Hildebrand (1956)]. Sufficient numerical results are quoted to show that these types of method warrant further consideration for those problems having a solution which is exponential in nature.

The final chapter, the eighth, summarises the results and conclusions drawn from the previous four chapters. We discuss the implications of these conslusions and finish by suggesting possible extensions to various aspects of the work carried out in this thesis.

Chapter Two

Introduction

Although various forms of the linear multistep methods and Runge-Kutta methods had existed from before 1900 subsequent development was comparatively slow mainly due to the lack of calculating facilities. The widespread introduction of computers in the early 1950's meant an enormous increase in the size and complexity of problems that could be tackled and this spurred on researchers into developing both the background theory and the numerical techniques involved. In this chapter we give a brief bibliography of the more important papers published from 1955 onwards concerning the numerical solution of the first-order initial-value problem. This bibliography, which is not claimed to be comprehensive, is divided into two sections. The first section deals with linear multistep methods and the second with Runge-Kutta methods.

2.1 Developments in Linear Multistep Methods

Perhaps the most significant papers concerning linear multistep methods were those produced by Dahlquist (1956, 1959) in which he formalized the definitions of convergence, produced necessary and sufficient conditions for convergence and obtained results on the maximum order attainable by convergent explicit and implicit methods. This material was then included in a book by Henrici (1962) which for many years was commonly regarded as the definitive work on the subject. Henrici's book was not restricted to linear multistep methods and also included chapters on the convergence and accuracy of Runge-Kutta methods. In a

subsequent book Henrici (1963) generalized the theory behind the use of linear multistep methods so that theoretical results were directly applicable to the solution of systems of first-order differential equations.

Much of Dahlquist's work was concerned with the behaviour of the numerical solution in the limiting case as the stepsize h tended to zero. Interest was then focused on ensuring that, when working with a finite stepsize h greater than zero, any errors introduced at a particular step would be damped out as the solution advanced. The forms of numerical solution being employed were variants of predictor - corrector methods and people who published work on these aspects include Hamming (1959), Milne and Reynolds (1959, 1960), Wilf (1959 , 1960) Hull and Newberry(1959, 1961, 1962), Ralston (1961 , 1965), Chase (1962), Crane and Lambert (1962), Hull and Creemer (1963), Crane and Klopfenstein (1965), Stetter (1965a, 1965b, 1968), Hall (1967), Klopfenstein and Millman (1968) and Lambert (1971).

A different topic considered by various people was whether a way could be found to circumvent the restrictions imposed by the demands of zero-stability on the maximum order of linear multistep methods. One technique suggested was to use one or more off-step points which meant finding the values of y_r and f_r at points such as $x + rh$ where r is not an integer. This idea was proposed independently by Gear (1964), Gragg and Stetter (1964), and Butcher (1965) while further developments were made by Kohfeld and Thompson (1967, 1968), Brush, Kohfeld and Thompson (1967), and Lyche (1969).

An alternative suggestion for obtaining higher order methods was to alter previously computed values of y and f and this was investigated by Nordsieck (1962), Gear (1967) and Beaudet and Feagin (1975). The methods of both Nordsieck and Gear involved making use of derivatives of y higher than the first. Other people who have considered this possibility include Lambert and Mitchell (1962), Liniger and Willoughby (1967), and Makinson (1968).

Yet another technique is that of using the so called 'block methods' where the solution is advanced by more than one step at a time and involves the cycling of corrector formulae. This idea has been investigated by Rosser (1967), Shampine and Watts (1969) and Donaldson and Hansen (1970).

The use of splines in deriving multistep formulae has been considered by the following, Loscalzo (1968a, 1968b), Loscalzo and Schoenberg (1969), Loscalzo and Talbot (1967a, 1967b), Callender (1971), Byrne and Chi (1972), Andria, Byrne and Hill (1973), and Hill (1975).

Modifying linear multistep methods so that the coefficients are no longer constants but depend on the stepsize h and $\partial f/\partial y$ is another idea for increasing the maximum value of the stepsize that can safely be used. This has received attention from Rahme (1969, 1970), Lambert (1970) and Lambert and Sigurdsson (1972).

One of the most important techniques suggested for the solution of the initial-value problem by multistep methods is that of employing various step sizes to carry out a step and then using extrapolation to obtain a more accurate result. This strategy was originally proposed by Gragg (1964) who put forward a method which was based on a

modification of a two-step explicit multistep method. The resultant method had a global error expansion which ascended in powers of h^2 and was therefore eminently suitable for use with extrapolation. Bulirsch and Stoer (1966) improved on Gragg's original method by proposing that rational extrapolation be used rather than polynomial extrapolation. Their method which can vary both the order and stepsize as necessary turns out to be one of the most efficient methods currently available when the cost of making function evaluations is small.

Other multistep methods which are regarded as being efficient are those developed by Krogh (1969, 1970, 1971), Gear (1971a, 1971b) and Sedgwick (1973). All these formulations are variants of the well known Adams methods with Krogh and Sedgwick making use of divided differences to store information about the solution while Gear uses scaled derivatives. All three methods can vary both the order and stepsize as necessary and the difference between them in addition to the way they store their information about the solution is in the mode of application, the maximum order available and their strategies for error control.

Over the last few years the development of new methods and modifications to improve older techniques has become less important and interest has turned to setting up test standards by which methods might be compared for efficiency and reliability and the comparison by these standards of the various methods currently available. Test criteria have been proposed by Hull (1967), Stewart (1968) and Hull et al (1972). The first attempts at comparison of methods were those of Clark (1968) and Crane and Fox (1969).

These early efforts were rather limited in scope and it was left to Hull et al (1972, 1974) to provide a full scale comparison of methods to a common test standard. Hull and his colleagues tested various types of methods including multistep methods over a wide range of problems where all the methods were forced to conform to a local error tolerance per unit step. For each method on each problem extensive statistics are taken relating to the time taken by the method and the level of accuracy it attains. Among the multistep methods considered the conclusion drawn is that the method of Bulirsch and Stoer is the most efficient if function evaluations are cheap to compute and that otherwise the methods of Krogh or Sedgwick are preferable.

No mention has been made so far concerning the application of linear multistep methods to types of problem which may create special difficulties such as problems with a singularity in the solution or problems where the value of the stepsize has to be kept very small to stop the propogation of errors even though there is no excessive need for accuracy. The solutions of these types of problem are the subjects of the chapters following this one and discussion of the relevant papers is incorporated in these later chapters and hence no references are given at this stage.

2.2 Developments in Runge-Kutta Methods

We now consider the development of the Runge-Kutta methods over the period of the last twenty years and we concentrate exclusively in this section on explicit methods. In the late 1950's and early 1960's when storage was still an important

consideration attention was focused on deriving methods which required the minimum amount of storage. This approach originally proposed by Gill (1951) was followed by Conte and Reeves (1956), Blum (1962) and Fyfe (1966) where these authors made choices for the free parameters present in most Runge-Kutta methods so as to reduce the storage and in certain cases round-off error. An alternative approach was to choose the free parameters in order to minimize the local error. This has been considered by , Kuntzmann (1959), , Ralston (1962), Hull and Johnson (1964) and King (1966). A further development of this idea is the proposal of methods which incorporate direct estimates of the local error and hence provide a mechanism for stepsize control. This has been suggested by Merson (1957), Scraton (1964), Fehlberg (1964), Shintani (1965, 1966a, 1966b) and England (1969). This technique can itself be extended by including or 'embedding' lower order methods within higher order methods to obtain an error control strategy. Fehlberg (1964, 1968, 1969), Sarafyan (1966a, 1966b, 1967) and Nasser (1967) are among the people who have investigated this particular approach.

Multistep versions of the Runge-Kutta methods, where we incorporate function evaluations outside the interval $[x_n, x_{n+1}]$, is a topic that has received attention from various authors including Byrne and Lambert (1966), Ceshino and Kuntzmann (1966), Byrne (1967) and Rosen (1968).

Another approach considered by Lawson (1966, 1967), and Lomax (1968) has been to choose the free parameters in various Runge-Kutta methods in an attempt to increase the maximum value of the stepsize that can safely be used.

Most attention over the period has been focused on the development of methods of ever increasing order. The results of Butcher (1965, 1975) give theoretical bounds on the minimum number of function evaluations necessary to obtain a given order while Cooper and Verner (1972) show that methods of order p can be obtained with $(p^2-7p+14)/2$ function evaluations. Other people who have produced high-order methods include Luther and Konen (1965), Luther (1966, 1968), Shanks (1966), Rosser (1967) and Fehlberg (1968, 1969, 1970).

Various Runge-Kutta methods were considered by Hull et al (1972, 1974) in their comparison of methods. They looked at the fourth, sixth and eighth order formulae of Runge (1895), Butcher (1964) and Shanks (1966) respectively and the formulae of orders four up to eight developed by Fehlberg (1968, 1969, 1970). In their conclusions they recommended the methods of Fehlberg as being the best of the various Runge-Kutta methods that were tested with the lower order methods being the most competitive at less stringent error tolerances and higher order methods the most competitive at more stringent tolerances. This is essentially due to the fact that none of the methods can vary the order and this has to be regarded as a disadvantage. In comparison with the multistep methods the methods of Fehlberg are equally as competitive as the extrapolation method of Bulirsch and Stoer when function evaluations are inexpensive although if this is not the case then the Adams multistep methods of Krogh and Sedgwick are preferable to any Runge-Kutta method.

Again no mention has been made in this section to those problems which give rise to special difficulties. One application of Runge-Kutta methods to a special problem is given in Chapter Five and the relevant literature will be quoted there. Finally it

should be pointed out that there are cases when the linear multistep methods and the Runge-Kutta methods overlap each other. One example of this is in some of the 'block' methods which can be expressed either in terms of Runge-Kutta formulae or in terms of linear multistep methods.

Chapter Three
Introduction

The background theory of convergence outlined in the first chapter is concerned with the behaviour of the numerical solution as the stepsize h tends to zero. In practice however we are working with a non-zero stepsize and would like to have some measure of the error involved and ideally a guarantee for damping out the error as the calculation proceeds. In this chapter and the next one we will be dealing only with linear multistep methods and for these Henrici (1962) has proved the following theorem.

Theorem 3.1

If $L[y(x);h]$ is a difference operator of order $p > 0$ then for all functions $y(x)$ which have $p + 1$ continuous derivatives in $[a, b]$ the bound on the local error is given by

$$|L[y(x_n);h]| \leq h^{p+1} GY \tag{3.1}$$

where

$$Y = \max_{x\in[a,\, b]} |y^{(p+1)}(x)| \tag{3.2}$$

and

$$G = \frac{1}{p!} \int_0^k |G(s)|ds \tag{3.3}$$

with

$$G(s) = \sum_{j=0}^{k} [\alpha_j (j-s)_+^p - p\beta_j (j-s)_+^{p-1}]. \tag{3.4}$$

Henrici also proves that under suitable assumptions the global error is of order p. Thus the bound for the global error is an order of magnitude greater than that of the local error due to a process of accumulation. Therefore by increasing the order of the local error we can raise the order of the global error. However this alone is not enough to stop the global error increasing. We have also to ensure that the errors are not allowed to build up as we proceed step by step with the calculation. To do this another restriction must be placed on the method and this we now discuss.

3.1 The Theory of Weak Stability

Consider the linear multistep method (1.4) and assume it to be both consistent and zero-stable. The theoretical solution $y(x)$ of the initial value problem (1.1) satisfies

$$\sum_{j=0}^{k} \alpha_j y(x_{n+j}) = h \sum_{j=0}^{k} \beta_j f(x_{n+j}, y(x_{n+j})) + T_{n+k} \quad . \tag{3.5}$$

Let $\{\tilde{y}_n\}$ be a solution of (1.4) when a roundoff error R_{n+k} is committed at the n-th application. Then

$$\sum_{j=0}^{k} \alpha_j \tilde{y}_{n+j} = h \sum_{j=0}^{k} \beta_j f(x_{n+j}, \tilde{y}_{n+j}) + R_{n+k} \quad . \tag{3.6}$$

If the total error is defined by

$$\tilde{e}_n = y(x_n) - \tilde{y}_n \tag{3.7}$$

subtracting (3.6) from (3.5) gives

$$\sum_{j=0}^{k} \alpha_j \tilde{e}_{n+j} = h \sum_{j=0}^{k} \beta_j \left[f(x_{n+j}, y(x_{n+j})) - f(x_{n+j}, \tilde{y}_{n+j}) \right] + \phi_{n+k} \tag{3.8}$$

where

$$\phi_{n+k} = T_{n+k} - R_{n+k} \quad . \tag{3.9}$$

Assuming that $\partial f/\partial y$ exists for all $x \in [a,b]$ and applying the mean-value theorem gives

$$f(x_{n+j}, y(x_{n+j})) - f(x_{n+j}, \tilde{y}_{n+j}) = \tilde{e}_{n+j} \frac{\partial f}{\partial y} (x_{n+j}, \xi_{n+j}) \tag{3.10}$$

where ξ_{n+j} lies between $y(x_{n+j})$ and $\tilde{y}_{n+j}$.

Hence

$$\sum_{j=0}^{k} \alpha_j \tilde{e}_{n+j} = h \sum_{j=0}^{k} \beta_j \frac{\partial f}{\partial y} (x_{n+j}, \xi_{n+j}) \tilde{e}_{n+j} + \phi_{n+k} \quad . \tag{3.11}$$

Consider the representative linear differential equation

$$y' = \lambda y \tag{3.12}$$

where λ is a real constant.

From (3.12) we have

$$\frac{\partial f}{\partial y} = \lambda \quad . \tag{3.13}$$

Making the additional assumption

$$\phi_{n+k} = \phi \tag{3.14}$$

where ϕ is a constant, (3.11) becomes

$$\sum_{j=0}^{k} (\alpha_j - h\lambda\beta_j) \tilde{e}_{n+j} = \phi \tag{3.15}$$

which has as its general solution

$$\tilde{e}_n = \sum_{s=1}^{k} d_s r_s^n - \frac{\phi}{h\lambda \sum_{j=0}^{k} \beta_j} \tag{3.16}$$

where the d_s are arbitrary constants depending on the initial errors, and the r_s are the roots, assumed distinct, of the equation

$$\sum_{j=0}^{k} (\alpha_j - h\lambda\beta_j) r^j = 0 . \tag{3.17}$$

Using the first and second characteristic polynomials as defined in (1.20) and (1.21) respectively (3.17) can be written as

$$\pi(r,\bar{h}) = 0 \tag{3.18}$$

where

$$\pi(r,\bar{h}) = \rho(r) - \bar{h}\ \sigma(r) \tag{3.19}$$

and

$$\bar{h} = h\lambda. \tag{3.20}$$

$\pi(r,\bar{h})$ is known as the stability polynomial of the linear multistep method.

From (3.16) it can be seen that the total error will grow only if any of the roots of (3.17) have modulus greater than one. Hence we have the following definitions as given in Lambert (1973).

Definition 3.1

The linear multistep method (1.4) is said to be absolutely stable for a given $\bar{h}$ if for that $\bar{h}$ all the roots r_s of (3.17) satisfy $|r_s| < 1$, $s = 1, 2, \ldots k$, and to be absolutely unstable otherwise.

Definition 3.2

An interval (α, β) of the real line is said to be an interval of absolute stability if the method is absolutely stable for all $\bar{h} \in (\alpha, \beta)$. If the method is absolutely unstable for all $\bar{h}$ it is said to have no interval of absolute stability.

We assumed that the roots of (3.17) were distinct but the restriction $|r_s| < 1$ in definition 3.1 will still cause the total error to decay as n increases even if the roots were multiple.

If the linear multistep method (1.4) is applied directly to the representative linear differential equation (3.12) we obtain

$$\sum_{j=0}^{k} (\alpha_j - h\lambda\beta_j)\, y_{n+j} = 0 . \tag{3.21}$$

Hence the numerical solution $\{y_n\}$ satisfies the same difference equation as the total error $\{\tilde{e}_n\}$. Thus if the numerical solution is increasing then so must the total error and we can therefore only achieve absolute stability if the solution is decreasing.

In developing the theory of weak stability we have used a linear differential equation (3.12) to represent the differential equation (1.1) which may or may not be linear. If (1.1) is linear then absolute stability will guarantee the damping out of the total error. We hope also that the behaviour of the total error as determined from the representative linear equation will give us an indication of the likely error behaviour when solving a non-linear differential equation. How good this indication is obviously depends on the degree of non-linearity of the differential equation (1.1).

3.2 Applicability of Weak Stability Theory to Systems of Differential Equations

With a system of m first-order ordinary differential equations the representative linear equation is

$$\underline{y}' = A\underline{y} \tag{3.22}$$

where $\underline{y} = ({}_1y, {}_2y, {}_3y, \ldots, {}_my)^T$ and $\underline{y}' = ({}_1y', {}_2y', {}_3y', \ldots {}_my')^T$.

The linear multistep method as applied to a system of m equations becomes

$$\sum_{j=0}^{k} \alpha_j \underline{y}_{n+j} = h \sum_{j=0}^{k} \beta_j \underline{f}_{n+j} \tag{3.23}$$

where $\underline{y}_{n+j} = ({}_1y_{n+j}, {}_2y_{n+j}, \ldots, {}_my_{n+j})^T$ and $\underline{f}_{n+j} = ({}_1f_{n+j}, {}_2f_{n+j}, \ldots {}_mf_{n+j})^T$

while the equation for the total error is

$$\sum_{j=0}^{k} \alpha_j \tilde{\underline{e}}_{n+j} = h \sum_{j=0}^{k} \beta_j \left[\underline{f}(x_{n+j}, \underline{y}(x_{n+j})) - \underline{f}(x_{n+j}, \tilde{\underline{y}}_{n+j})\right] + \underline{\phi}_{n+k} \qquad (3.24)$$

where $\tilde{\underline{e}}_{n+j} = \underline{y}(x_{n+j}) - \tilde{\underline{y}}_{n+j}$ and $\underline{\phi}_{n+k} = \underline{T}_{n+k} - \underline{R}_{n+k}$.

Applying the mean value theorem to (3.24) gives

$$\sum_{j=0}^{k} \alpha_j \tilde{\underline{e}}_{n+j} = h \sum_{j=0}^{k} \beta_j \frac{\partial \underline{f}}{\partial \underline{y}} \tilde{\underline{e}}_{n+j} + \underline{\phi}_{n+k} \qquad (3.25)$$

where $\partial \underline{f}/\partial \underline{y}$ is the (m x m) Jacobian matrix which has as its (i,j) element the partial derivative of ${}_i f(x, {}_1y, {}_2y, \ldots, {}_m y)$ with respect to ${}_j y$ evaluated at some point between ${}_i y(x_{n+j})$ and ${}_i \tilde{y}_{n+j}$.

Making the assumptions

$$\frac{\partial \underline{f}}{\partial \underline{y}} = J \qquad (3.26)$$

where J is an (m x m) matrix with constant coefficients and

$$\underline{\phi}_{n+k} = \underline{\phi} \qquad (3.27)$$

where $\underline{\phi}$ is a vector of constants, (3.25) can be written as

$$\sum_{j=0}^{k} (\alpha_j I - h\beta_j J) \tilde{\underline{e}}_{n+j} = \underline{\phi} \qquad (3.28)$$

where I is the (m x m) identity matrix.

If we also assume that the eigenvalues λ_j, j=1,2, ..., m of J are distinct then there exists a similarity transformation such that

$$P^{-1} J P = D$$

where $D = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)$.

Define

$$\tilde{\underline{e}}_n = P \underline{d}_n \,. \qquad (3.29)$$

Premultiplying (3.28) by P^{-1} and substituting for $\tilde{\underline{e}}_n$ from (3.29) gives

$$\sum_{j=0}^{k} (\alpha_j I - h\beta_j D) \underline{d}_{n+j} = \underline{g} \qquad (3.30)$$

where

$$\underline{g} = P^{-1}\underline{\phi}. \tag{3.31}$$

Since I and D are both diagonal matrices (3.30) can be rewritten as

$$\sum_{j=0}^{k} (\alpha_j - h\beta_j\lambda_i)_i d_{n+j} = {}_ig \qquad i = 1,2,\ldots,m. \tag{3.32}$$

Each of these equations is independent of all the others and each has the same form as the linearized error equation (3.15) for a single differential equation with ${}_id_n$ and λ_i corresponding to $\tilde{e}_n$ and λ respectively. From (3.29) $\underline{d}_n$ will grow or decay with n if and only if $\underline{\tilde{e}}_n$ does and so all the theory developed for a single equation will hold if λ is taken as an eigenvalue of the Jacobian matrix J.

Since the Jacobian is usually unsymmetric its eigenvalues are not necessarily real. Hence for a system the parameter $\bar{h}$ appearing in the stability polynomial (3.18) might be complex. We then have to consider a region for $\bar{h}$ rather than an interval while in addition any criteria involving $\bar{h}$ must be satisfied for all the eigenvalues of the Jacobian. Therefore to extend the concept of absolute stability to systems of differential equations we again follow Lambert (1973) and modify the earlier definition 3.1 of absolute stability to:-

Definition 3.3

A linear multistep method is said to be absolutely stable in a region R of the complex plane if, for all $\bar{h} \in R$, all roots of the stability polynomial $\pi(r,\bar{h})$ associated with the method satisfy $|r_s| < 1$, $s = 1, 2, \ldots, k$.

3.3 Calculation of the Region of Absolute Stability.

The roots of a polynomial are continuous functions of

the coefficients. Thus the roots of the stability polynomial $\pi(r,\bar{h}) = 0$ are continuous functions of $\bar{h}$. If we denote the boundary of the stability region by ∂R then $\bar{h}$ will lie on ∂R when one of the roots of $\pi(r,\bar{h}) = 0$ lies on the boundary of the unit circle. This happens when

$$\rho(\exp(i\theta)) - \bar{h}\sigma(\exp(i\theta)) = 0$$

that is,

$$\bar{h}(\theta) = \frac{\rho(\exp(i\theta))}{\sigma(\exp(i\theta))}. \tag{3.33}$$

Hence we can plot the boundary ∂R by evaluating $\bar{h}(\theta)$ from (3.33) for various values of θ running from zero to 2π radians. This technique for determining the stability region is known as the boundary locus method.

3.4 'Stiff' Systems of Ordinary Differential Equations

Consider the following linear system of m first-order ordinary differential equations

$$\underline{y}' = A\underline{y} + \underline{\alpha}(x) \tag{3.34}$$

where the (m x m) matrix A is again assumed to have m distinct eigenvalues λ_i and corresponding eigenvectors $\underline{c}_t$, $t = 1,2,\ldots, m$. This system has a general solution of the form

$$\underline{y}(x) = \sum_{t=1}^{m} k_t e^{\lambda_t x} \underline{c}_t + \underline{\beta}(x). \tag{3.35}$$

If in addition

$$\text{Re } \lambda_t < 0 \qquad t = 1,2,\ldots,m \tag{3.36}$$

then

$$\sum_{t=1}^{m} k_t e^{\lambda_t x} \underline{c}_t \to 0 \quad \text{as} \quad x \to \infty.$$

The term $\sum_{t=1}^{m} k_t e^{\lambda_t x} \underline{c}_t$ is called the transient solution while $\underline{\beta}(x)$ is known as the steady state solution. If any

one of the eigenvalues of A, say λ_ν, is large in magnitude then to achieve absolute stability h will have to be kept very small throughout the range of integration even though the magnitude of the term involving λ_ν rapidly decays to zero. This type of problem is usually termed 'stiff'. We quote the following formal definition of 'stiff' as given by Lambert (1973).

Definition 3.4

The linear system $y' = A\underline{y} + \underline{\alpha}(x)$ is said to be stiff if

(i) $\mathrm{Re}\,\lambda_t < 0$, $t = 1,2,\ldots,m$ and

(ii) $\max_{t=1,2,\ldots,m} |\mathrm{Re}\lambda_t| >> \min_{t=1,2,\ldots,m} |\mathrm{Re}\lambda_t|$ where $\lambda_t, t=1,2,\ldots,m,$ are the eigenvalues of A. The ratio $\left[\max_{t=1,2,\ldots,m} \left|\mathrm{Re}\lambda_t\right|\right] : \left[\min_{t=1,2,\ldots,m} \left|\mathrm{Re}\lambda_t\right|\right]$ is called the stiffness ratio.

If the non-linear system $\underline{y}' = \underline{f}(x,\underline{y})$ can be expanded by a Taylor series about the point $(\bar{x}, \bar{\underline{y}})$ which is assumed near to $(x,\underline{y})$ then

$$\underline{y}' = \frac{\partial \underline{f}}{\partial \underline{y}}(\bar{x},\bar{\underline{y}})\,\underline{y} + \underline{f}(\bar{x},\bar{\underline{y}}) - \bar{x}\,\frac{\partial \underline{f}}{\partial x}(\bar{x},\bar{\underline{y}}) - \frac{\partial \underline{f}}{\partial \underline{y}}(\bar{x},\bar{\underline{y}})\bar{\underline{y}} + x\frac{\partial \underline{f}}{\partial x}(\bar{x},\bar{\underline{y}}) \qquad (3.37)$$

provided we ignore higher order terms. (3.37) is of the same form as (3.34) with A corresponding to $\partial \underline{f}(\bar{x}, \bar{\underline{y}})/\partial \underline{y}$. Thus the non-linear problem can be approximated by a linear problem over sufficiently small intervals in x and an estimate of the stiffness ratio obtained by considering the eigenvalues of the Jacobian matrix $\partial \underline{f}(\bar{x},\bar{\underline{y}})/\partial \underline{y}$.

For a system of equations the Lipschitz condition of Theorem 1.1 is replaced by

$$||\underline{f}(x,\underline{y}) - \underline{f}(x,\underline{y}^*)|| \leq L|\underline{y}-\underline{y}^*| \qquad (3.38)$$

where $||\ .\ ||$ denotes a vector norm. If the partial derivatives occurring in the Jacobian $\partial \underline{f}/\partial \underline{y}$ are continuous and bounded in a region R, $a \leq x \leq b$, $-\infty < {}_i y < \infty$, $i = 1,2,\ldots,m$ then the Lipschitz constant of the system $\underline{y}' = \underline{f}(x,\underline{y})$ may be taken as

$$L = \sup_{(x,y \in R)} \left|\left| \frac{\partial \underline{f}}{\partial \underline{y}} \right|\right| \tag{3.39}$$

where $||\ .\ ||$ denotes the matrix norm subordinate to the vector norm in (3.38). For any (m x m) matrix A and any matrix norm

$$||\ A\ || \geq \max_{t=1,2,\ldots,m} |\ \lambda_t\ |\ .$$

Stiff systems of equations have $\max_{t=1,2,\ldots m} |\mathrm{Re}\ \lambda_t| >> 0$

and hence $L >> 0$ and are consequently often referred to as systems with large Lipschitz constants.

3.5 Desirable Stability Regions for Stiff Systems of Equations

To solve a stiff system of ordinary differential equations we need a linear multistep method which is not so restricted by the demands of absolute stability that we have to use a small stepsize even after the initial transients have decayed. Dahlquist (1963) proposed that methods for stiff equations should be 'A-stable', a property he defined by:-

Definition 3.5

A numerical method is said to be A-stable if its region of absolute stability contains the whole of the left-hand half-plane $\mathrm{Re}\ \lambda h < 0$.

In addition Dahlquist subsequently proved the following theorem.

Theorem 3.2

(i) An explicit linear multistep method cannot be A-stable.

(ii) The order of an A-stable implicit linear multistep method cannot exceed two.

(iii) The second order A-stable implicit linear multistep method with the smallest error constant is the Trapezoidal Rule.

A-stability thus turns out to be a very demanding requirement and so Gear (1968) proposed the following:-

Definition 3.6

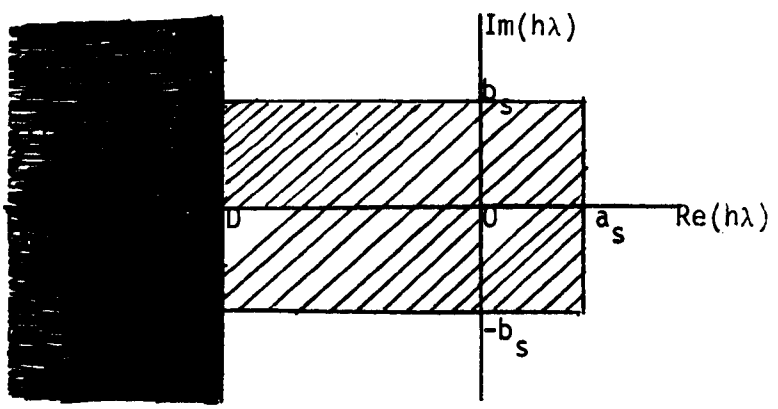
A numerical method is stiffly-stable if (i) its region of absolute stability contains R_1 and R_2 and (ii) it is accurate for all $h\lambda \in R_2$ when applied to the linear representative equation $y' = \lambda y$, λ a complex constant with $\mathrm{Re}\ \lambda < 0$ where

$$R_1 = \{h\lambda \mid \mathrm{Re}\ h\lambda < -D\}$$

$$R_2 = \{h\lambda \mid -D \leq \mathrm{Re}\ h\lambda \leq a_s,\ -b_s \leq \mathrm{Im}\ h\lambda \leq b_s\}$$

and where D, a_s and b_s are positive constants.

We illustrate this definition by the following figure



■ Denotes the region R_1 //// Denotes the region R_2

The eigenvalues with negative real parts and with large moduli correspond to terms in the transient solution which decay rapidly. After the initial transients have died away we no longer insist on an accurate representation for these terms but only that they are not allowed to cause instability. Hence $h\lambda$ for these rapid transients will eventually come in the region R_1 where we only require stability. All the other terms should be represented both accurately and stably and by a suitable choice for h these terms can have values of $h\lambda$ which come in the region R_2 where we insist on both accuracy and stability.

An additional complication arises with some multistep methods whereby although the method is either A-stable or stiffly-stable when it is applied with a constant stepsize h to the linear representative equation $y' = \lambda y$, λ a complex constant with $\mathrm{Re}\ \lambda < 0$, we find

$$\lim_{\substack{n\to\infty \\ h\lambda\to-\infty}} \left|\frac{y_{n+1}}{y_n}\right| \to 1.$$

Thus y_n will decay to zero very slowly. Methods for which this ratio tends to one (for example, the trapezoidal rule) when applied to stiff differential equations can cause some of the transients that decay rapidly to zero in the analytical solution to be represented in the numerical solution as slowly damped or oscillatory components and this can restrict the stepsize. This leads us on to a further definition as given by Enright (1972).

Definition 3.7

A method is stable at infinity if, when it is applied to $y' = \lambda y$, λ a real negative constant, with a constant

stepsize h, there exists real $W < 0$ such that

$$\sup_{h\lambda<W} \lim_{n\to\infty} \left|\frac{y_{n+1}}{y_n}\right| < 1. \tag{3.40}$$

The condition (3.40) implies that $y_n \to 0$ as $n \to \infty$. Since the total error and the numerical solution satisfy the same difference equation this implies that the error is decreasing as $n \to \infty$. The roots of a polynomial are continuous functions of its coefficients therefore as $h\lambda \to -\infty$ so the roots of the stability polynomial tend to those of $\sigma(\xi) = 0$. Hence for a linear multistep method to be stable at infinity we require that the roots of $\sigma(\xi)$ be less than one in magnitude.

Ideally we would like to have methods which were both A-stable and stable at infinity but since A-stability is so demanding we will settle for methods which are stiffly stable and stable at infinity.

In this chapter we have defined the concept of absolute stability for a single equation and for a system of equations. Additionally we have defined 'stiff' differential equations and outlined the desirable stability criteria for this type of problem. In the next chapter we will detail some of the stiffly stable methods that have already been proposed by various researchers and suggest alternative methods.

Chapter Four

4.1 Stiffly-Stable First-Derivative Multistep Methods

Following his definition of stiff stability Gear (1968) went on to propose a class of stiffly-stable multistep methods. He suggested that for a k-step method the second characteristic polynomial should be taken as

$$\sigma(\xi) = \xi^k. \tag{4.1}$$

This choice for $\sigma(\xi)$ ensures that the method is stable at infinity since the roots of $\sigma(\xi) = 0$ are all zero. Thus Gear normalizes his methods by taking $\beta_k = 1$ while by setting $\beta_0, \beta_1, \ldots, \beta_{k-1}$ to be zero the remaining k+1 coefficients $\alpha_0, \alpha_1, \ldots, \alpha_k$ can be chosen to make the method have order k without violating the condition of zero-stability. He then showed that for this class of methods a k-step method could have order k and be stiffly-stable for k = 1,2,3,...,6 but that methods with $k \geq 7$ were not stiffly-stable.

In an attempt to derive stiffly-stable methods of higher order Dill (1969) proposed alternative choices for $\sigma(\xi)$. To obtain a seventh-order method he took k = 7 and $\sigma(\xi) = \xi^6(\xi-0.99)$ while for an eighth-order method he took k = 8 and $\sigma(\xi) = \xi^6(\xi^2-1.8\xi+.81)$.

Jain and Srivastava (1970) subsequently carried out a more systematic approach by considering various classes of methods based on different choices for $\sigma(\xi)$. Their choices were

$$\sigma(\xi) = \xi^{k-r}(\xi-c)^r \qquad r = 1,2,\ldots k \quad -1<c<1 \tag{4.2}$$

$$\sigma(\xi) = \xi^{k-r}(\xi^r-c^r) \qquad r = 2,\ldots,k \quad -1<c<1 \tag{4.3}$$

$$\sigma(\xi) = \xi^{k-r} \sum_{i=0}^{r} \xi^{r-i}c^i \qquad r = 2,\ldots,k \quad -1<c<1. \tag{4.4}$$

With suitable choices for r and c (4.2) gives stiffly stable methods up to order eleven while (4.3) and (4.4)

only give stiffly-stable methods up to order seven and eight respectively. Some of these methods give a larger stability region than Gear's original choice for $\sigma(\xi)$. Jain and Srivastava give the coefficients of their more interesting methods and plots of the corresponding stability regions in their report.

4.2 Stiffly-Stable Second-Derivative Multistep Methods

When we apply an implicit first-derivative linear multistep method to solve the first-order ordinary differential equation (1.1) and attempt to calculate y_{n+k} by using the iteration defined by (1.7), (1.8) and (1.9) then from Theorem 1.2 this iteration will only converge if

$$h < \frac{1}{L|\beta_k|} .$$

For a stiff differential equation L is large and we are therefore restricted to using a small stepsize throughout $[a,b]$ to ensure convergence of the iteration. One way to avoid this restriction is by applying the Newton-Raphson method to (1.9) to give

$$y_{n+k}^{(\nu+1)} - y_{n+k}^{(\nu)} = \left[1-h\beta_k \frac{\partial f}{\partial y}(x_{n+k}, y_{n+k}^{(\nu)})\right]^{-1} \left[-y_{n+k}^{(\nu)} + h\beta_k f(x_{n+k}, y_{n+k}^{(\nu)}) + C\right] \qquad (4.5)$$

where $y_{n+k}^{(\nu)}$ is the value obtained for y_{n+k} after the νth iteration. This iteration is exact for a linear differential equation and converges adequately for most stiff equations. For a system of equations (4.5) takes the form

$$\underline{y}_{n+k}^{(\nu+1)} - \underline{y}_{n+k}^{(\nu)} = \left[I-h\beta_k \frac{\partial \underline{f}}{\partial \underline{y}}(x_{n+k}, \underline{y}_{n+k}^{(\nu)})\right]^{-1} \left[-\underline{y}_{n+k}^{(\nu)} + h\beta_k \underline{f}(x_{n+k}, \underline{y}_{n+k}^{(\nu)}) + \underline{C}\right]. \qquad (4.6)$$

Since the Jacobian matrix $\partial \underline{f}/\partial \underline{y}$ is needed for the iteration the possibility of developing a class of methods which make use of the Jacobian has been considered by Enright

(1972). He was interested in autonomous systems of first-order ordinary differential equations, that is, systems of the form

$$\underline{y}' = \underline{f}(\underline{y}) \tag{4.7}$$

as many stiff systems are encountered in an autonomous form while those that are not can be made autonomous by adding an extra equation ${}_{m+1}y' = 1$, ${}_{m+1}y(a) = a$ (which has the solution $y = x$) and replacing all occurrences of x by ${}_{m+1}y$. For an autonomous system

$$\underline{y}'' = \left[\frac{\partial \underline{f}}{\partial \underline{y}}\right] \underline{y}' \tag{4.8}$$

and Enright proposed using multistep formulae which incorporate the second derivative of y.

The general k-step second-derivative multistep method for a single differential equation is

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j} + h^2 \sum_{j=0}^{k} \gamma_j f^{(1)}_{n+j} \tag{4.9}$$

where $f^{(1)}_{n+j} = \frac{d}{dx}\left[f(x,y)\right]_{\substack{x=x_{n+j} \\ y=y_{n+j}}}$.

Associated with the multistep method (4.9) is the linear difference operator L defined by

$$L[z(x);h] = \sum_{j=0}^{k} \left[\alpha_j z(x+jh) - h\beta_j z'(x+jh) - h^2\gamma_j z''(x+jh)\right] \tag{4.10}$$

where z is an arbitrary test function infinitely differentiable on $[a,b]$.

Expanding $z(x+jh)$, $z'(x+jh)$ and $z''(x+jh)$ by Taylor series and collecting terms gives

$$L[z(x);h] = \sum_{r=0}^{\infty} C_r h^r z^{(r)}(x) \tag{4.11}$$

where

$$\left.\begin{aligned} C_0 &= \sum_{j=0}^{k} \alpha_j \\ C_1 &= \sum_{j=0}^{k} (j\alpha_j - \beta_j) \\ C_r &= \sum_{j=0}^{k} [j^r \alpha_j - r j^{r-1} \beta_j - r(r-1) j^{r-2} \gamma_j] \qquad r=2,3,.... \end{aligned}\right\} \quad (4.12)$$

The difference operator and associated linear multistep method (4.9) are then said to be of order p if in (4.11)

$$C_0 = C_1 = ... = C_p \text{ but } C_{p+1} \neq 0. \quad (4.13)$$

C_{p+1} is again referred to as the error constant.

The local error at x_{n+k} of the linear multistep method (4.9) is given by

$$L[y(x_n);h] = \sum_{j=0}^{k} [\alpha_j y(x+jh) - h\beta_j y'(x+jh) - h^2 \gamma_j y''(x+jh)] \quad (4.14)$$

where this is again often denoted by T_{n+k}.

Associated with the multistep method (4.9) are three characteristic polynomials. The first and second are given by (1.20) and (1.21) respectively while the third is defined by

$$\psi(\xi) = \sum_{j=0}^{k} \gamma_j \xi^j. \quad (4.15)$$

Consistency and zero-stability are then necessary and sufficient conditions for the convergence of the multistep method (4.9).

If we follow the weak stability analysis of section 3.1 we find that for the second-derivative multistep method (4.9) the error equation corresponding to (3.15) is

$$\sum_{j=0}^{k} (\alpha_j - h\lambda\beta_j - h^2\lambda^2 \gamma_j) \tilde{e}_{n+j} = \phi \quad (4.16)$$

and hence the stability polynomial now becomes

$$\pi(r,\bar{h}) = \rho(r) - \bar{h}\sigma(r) - \bar{h}^2\psi(r). \tag{4.17}$$

Thus if the multistep method (4.9) is to be stable at infinity the roots of $\psi(\xi) = 0$ must be less than one in magnitude.

To ensure zero-stability Enright has chosen

$$\rho(\xi) = \xi^k - \xi^{k-1} \tag{4.18}$$

while for stability at infinity he takes

$$\psi(\xi) = \gamma_k \xi^k \tag{4.19}$$

This enables him to determine $\beta_0,\beta_1,\ldots,\beta_k$ and γ_k so that his resultant method has order k+2. His methods are stiffly stable for k = 1,2,...,7.

In Table 4.1 we quote for the methods of Gear and Enright the appropriate values for D and b_s (Definition 3.6) for each value of k. In addition for future comparison purposes we give the error constant C_{p+1} for Enright's methods after they have been normalized so that $\gamma_k = 1$.

Table 4.1

GEAR				ENRIGHT			
D	b_s	k	order	D	b_s	C_{p+1}	k
0	A-stable	1	1				
0	A-stable	2	2				
0.1	0.75	3	3	0	A-stable	.083	1
0.7	0.75	4	4	0	A-stable	.039	2
2.4	0.75	5	5	0.1	2.0	.022	3
6.1	0.5	6	6	0.52	2.0	.014	4
			7	1.4	2.0	.010	5
			8	2.7	2.0	.0074	6
			9	5.3	1.9	.0057	7

It would seem a logical progression to look at alternative classes of second-derivative multistep methods to see whether there are methods which have larger stability regions. Analogously to the manner in which Jain and Srivastava investigated possible modifications to Gear's first-derivative multistep methods we will attempt initially to follow a similar approach with respect to Enright's second-derivative methods.

The general second-derivative linear multistep method (4.9) has one arbitrary coefficient. We normalize the method by choosing γ_k as the arbitrary coefficient and then setting it equal to one. For consistency it is necessary that

$$\alpha_{k-1} = -\alpha_k - \sum_{j=0}^{k-2} \alpha_j . \tag{4.20}$$

To ensure zero-stability while satisfying the consistency condition (4.20) we set

$$\alpha_j = 0 \qquad j = 0,1,\ldots,k-2$$

and take

$$\alpha_k = -\alpha_{k-1}$$

so that

$$\rho(\xi) = \alpha_k(\xi^k - \xi^{k-1}). \tag{4.21}$$

Since all the roots of $\psi(\xi) = 0$ must be less than one in magnitude to ensure stability at infinity we will investigate in turn each of the following choices for $\psi(\xi)$:

$$\psi(\xi) = \xi^{k-r}(\xi-c)^r \qquad r = 1,2,\ldots,k \qquad -1<c<1 \tag{4.22}$$

$$\psi(\xi) = \xi^{k-r}(\xi^r - c^r) \qquad r = 2,\ldots,k \qquad -1<c<1 \tag{4.23}$$

$$\psi(\xi) = \xi^{k-r} \sum_{i=0}^{r} \xi^{r-i} c^i \qquad r = 2,\ldots,k \qquad -1<c<1 \tag{4.24}$$

Note that if c is zero the three choices each reduce to Enright's method, while r starts at two in (4.23) and (4.24)

since taking r = 1 would give the same method as taking r = 1 in (4.22). The coefficients α_k, $\beta_0, \ldots, \beta_k$ can then be determined for each of the above choices for $\psi(\xi)$ so that the method has order k + 2.

Although we have obtained stiffly-stable methods which have an order of twelve and it may well be possible to obtain still higher-order methods, we only give results for those methods which have an order of nine or less. Hence since each of these classes of k-step method have order k + 2 the maximum value of k will be taken as seven. For each of the three choices for $\psi(\xi)$ results have been taken for values of c running from -0.9 to 0.9 in steps of 0.2 except that when r is even (4.23) is symmetric in c and thus we need only consider c going from -0.9 to -0.1 in steps of 0.2. The quantities D and b_s are determined by using the boundary locus method usually with an interval in θ of five degrees although if it appeared from the results that this interval was too large we repeated the calculation using an interval in θ of one degree.

In Appendix 1 results are only given for those values of c for which we actually obtained a stiffly-stable multistep method having a value of less than ten for D. We quote in tables A1.1, A1.2, A1.3, each table corresponding to one of the three classes of method, the value of c and the corresponding values for D, b_s and the error constant C_{p+1}. To make an easier comparison of the various results we include three additional tables, tables A1.4, A1.5, and A1.6, each again corresponding to a particular class of method, which give the values of c and C_{p+1} at the

minimum value of D for each pair of values of r and k. Each of these last three tables start from k equal to three since for all three classes of multistep method k equal to one or two and c equal to zero give A-stable methods which are stable at infinity, and we therefore omit these from our comparison.

Although it is impossible to draw any specific conclusions from the results we can make some general observations. From tables A1.1 and A1.4 we can see that the choice of (4.22) for $\psi(\xi)$ can give us stiffly-stable methods up to order nine at least. It can also be seen that for this particular choice, r equal to one usually gives both the largest stability region, as measured by the value of D, and the smallest error constant although the value of c at which these are obtained tends towards one as k increases. Intuitively we feel that a value of c too near one will give us larger errors when working with bigger values of $\bar{h}$. It is certainly possible however with r equal to one and an appropriate choice for c, without taking it too close to one, to obtain stiffly-stable methods with larger stability regions and smaller error constants than those of Enright. Higher values of r in general give smaller stability regions than when r is one and the error constant tends to be larger although the value of c at which we obtain the largest stability region decreases.

If we now consider the second choice for $\psi(\xi)$, given in (4.23), it can be seen from the results in tables A1.2 and A1.5 that the largest stability regions occur when r is equal to k. Although the values of c are not too close to one, the truncation errors are however much greater than

when r is one.

Taking $\psi(\xi)$ as defined by (4.24) the results in tables A1.3 and A1.6 indicate again that r being one gives the largest stability regions and the smallest error constants.

Comparing between the three choices for $\psi(\xi)$ we observe that the best alternative choice of method between (4.22) and (4.24) is the method for which

$$\psi(\xi) = \xi^{k-1}(\xi-c) \tag{4.25}$$

where an appropriate value for c can be found from table A1.1. The best choice of method from (4.23) is obtained when

$$\psi(\xi) = \xi^{k} - c^{k} \tag{4.26}$$

where an appropriate value for c can be found from table A1.2.

For the same stepnumber k the method determined by the choice (4.26) for $\psi(\xi)$ always has a larger stability region than the method obtained from the choice (4.25) but it also has a much larger error constant. Recalling also that when $\bar{h}$ is large the roots of the error equation (4.16) tend towards those of $\psi(\xi)$ we note that $\xi = c$ is a root of $\psi(\xi)$ for both choices but,whereas the remaining roots of $\psi(\xi)$ in (4.25) are all zero,the other k - 1 roots of $\psi(\xi)$ in (4.26) have magnitude c. We therefore suspect that the choice of (4.26) for $\psi(\xi)$ may lead to much larger errors than (4.25).

With all three choices for $\psi(\xi)$ we have obtained stiffly-stable methods up to order nine which have larger stability regions than Enright's methods. We suggest that the best alternative method based on the choice of (4.22), (4.23) or (4.24) for $\psi(\xi)$ is the method which

has $\psi(\xi)$ given by (4.25).

4.3 The Use of Optimization Techniques to Increase the Stability Region of Stiffly-Stable Second-Derivative Multistep Methods

In each of the three choices (4.22), (4.23) and (4.24) for $\psi(\xi)$ we are imposing a particular form on the third characteristic polynomial. It seems very unlikely that any one of those three arbitrary choices for $\psi(\xi)$ from the infinite number that could be made would turn out to be a best possible choice from the viewpoint of stability, since the coefficients of $\psi(\xi)$ are constrained to vary in a very restrictive manner. Ideally we would like to be able to allow some of the coefficients to vary in such a way that they were more directly related to increasing the stability region. This leads us to an alternative viewing of the problem as being essentially one of optimization since we have an objective, the size of the stability region, which we wish to maximize and this objective depends on the coefficients $\gamma_{k-1}, \ldots, \gamma_0$. One can therefore attempt to apply optimization techniques to this problem. The idea of using optimization to maximize stability regions has been proposed by Crane and Klopfenstein (1965) who successfully used a steepest descent method to increase the stability region of a particular predictor-corrector algorithm. Schoen (1971) has also used optimization to extend stability regions.

Our objective then is to maximize the stability region of stiffly-stable second-derivative multistep methods and since D is our best measure of the size of this region the problem is effectively to minimize D; D of course depends on the coefficients and the order. If we still require

that the order be k + 2 and again choose $\rho(\xi)$ as in (4.21) so ensuring zero-stability without violating the consistency conditions we can then write

$$D = F(\gamma_0, \gamma_1, \ldots\ldots, \gamma_{k-1}, k+2) \tag{4.27}$$

This function is calculated numerically for a particular set of coefficients $\gamma_0, \gamma_1, \ldots, \gamma_{k-1}$ and order k + 2 by first determining the other coefficients $\alpha_k, \beta_0, \ldots, \beta_k$ and then using the boundary locus method with an interval in θ of five degrees, this having previously been found adequate. It does mean however that the function values are not calculated to very great precision. In addition the multistep method should satisfy the constraint that it be stable at infinity. Hence the zeros of $\psi(\xi)$, which are denoted by $\xi_{3,1}, \ldots, \xi_{3,k}$, must be less than one in modulus.

Thus our problem is

$$\text{minimize } D = F(\gamma_0, \gamma_1, \ldots, \gamma_{k-1}, k+2) \tag{4.28}$$

subject to the constraints

$$|\xi_{3,j}| < 1 \qquad j = 1,2,\ldots,k. \tag{4.29}$$

The problem is therefore one of constrained optimization of a function which is calculated numerically to low precision. It is impossible to calculate analytically the derivatives of the function (4.27) and while it would be possible to calculate the derivatives numerically by using differences this idea is rejected because the function values are not very accurate. These considerations drastically restrict the possible methods open to us to carry out the optimization process. We need therefore to look for a method which is applicable to

constrained problems and uses only function values. This virtually reduces the choice to two methods: one due to Powell (1964) and the other due to Rosenbrock (1960). Powell's method which makes use of conjugate directions is more modern and almost certainly the more efficient. Unfortunately to use it for constrained optimization really requires the use of penalty functions. This approach was considered but it was found impossible to develop the right type of penalty function to suit this particular problem. The problem is in fact even more difficult than it may appear since there is no guarantee that the resultant stability region will remain connected and the multistep method stiffly-stable even when the coefficients are varied in such a manner as to satisfy all the constraints.

Because of these difficulties it was decided to use the optimization method of Rosenbrock. We will first describe his basic technique for minimization and then show how it has been adapted to finding a solution for our problem.

At the start of the ℓth iteration let us suppose we have an approximation $\underline{\gamma}^{(\ell)} = (\gamma_0^{(\ell)}, \gamma_1^{(\ell)}, \ldots, \gamma_{k-1}^{(\ell)})$ to the solution together with a set of k mutually orthogonal search directions $q_1, q_2, \ldots, q_k$ and a set of k associated step lengths $\delta_1, \delta_2, \ldots, \delta_k$. Starting from $\underline{\gamma}^{(\ell)}$ a step δ_1 is taken along the direction $q_1^{(\ell)}$. If this does not result in an increase in the value of the function it is considered to be a successful step and the new improved estimate for the minimum retained and δ_1 multiplied by a constant $\alpha > 1$. If however the function value does increase then the step is regarded as a failure and is consequently rejected while δ_1 is multiplied by a constant β where $0 > \beta > -1$. A

common choice is to take $\alpha=3$ with $\beta = -\frac{1}{2}$ which will be the values we use for α and β. The search considers each direction in turn until all the k directions have been explored, and then returns to the first direction $q_1^{(\ell)}$ and recycles through all the directions until a success followed by a failure has been recorded at some time during the iteration for every direction. At this stage a new set of orthogonal directions is calculated. The first of these is taken to be the resultant of the steps taken with the previous directions while the remaining directions are calculated by a method due to Powell (1968). The aim of this repeated orthogonalization is to align the first direction q_1 along the principal axis of the contours and in general q_{r+1} along the best direction which can be found normal to q_r. Initially the search vectors are taken along the coordinate directions.

To apply Rosenbrock's method we need to be able to calculate the function value D for any given coefficient vector $\underline{\gamma}$. This is carried out by first calculating the coefficients α_k, β_0,..., β_k and the error constant C_{p+1} and then using an algorithm due to Duffin (1970) for testing whether or not a polynomial is a Schur polynomial and hence whether the constraints on $\psi(\xi)$ have been violated. If the constraints have been violated then that particular step is simply regarded as a failure otherwise the boundary locus method is used to calculate the value of D. As a starting point for the whole process we take $\psi(\xi) = \xi^k$ and thus $\gamma_0^{(1)} = \gamma_1^{(1)} = \ldots = \gamma_{k-1}^{(1)} = 0$ since we know this certainly gives us stiffly-stable methods for $k \leq 7$. The possibility that the stability region becomes disconnected

during the optimization process always exists but it seems reasonable to make the assumption that if this happened the value for D would then be greater than the current lowest estimate. Hence we take no special precautions to cope with this other than to check that the final solution does give us a connected stiffly-stable region.

The constraints (4.29) which we have applied in this problem are those that are necessary for the multistep method to be stable at infinity. We might however wish to impose more stringent constraints on the roots of $\psi(\xi) = 0$ in the hope that when $\bar{h}$ is very large the errors will be less. It might therefore be thought desirable to make the constraints

$$|\xi_{3,j}| < w \qquad j = 1,2,\ldots,k \tag{4.30}$$

where w must certainly be less than one. It is then straightforward to modify Duffin's algorithm for a given value of w so that we can use this type of constraint in the optimization process.

In Appendix 1 in Tables A1.7 we give the results obtained from the optimization process. We have again taken results with values of k running from three up to and including seven. With each value of k we have in turn allowed there to be from one up to four variable coefficients, except for k being three when a maximum of three varying coefficients is allowed. If there is to be only one varying coefficient we take it to be γ_{k-1} since we have previously concluded from the results in the first appendix that with a suitable choice for γ_{k-1} in (4.22) we could obtain larger stability regions than those of Enright. In general if there are i varying

coefficients where $i \leq 4$ these are taken to be $\gamma_{k-1}, \ldots, \gamma_{k-i}$ while the remaining coefficients of $\psi(\xi)$, $\gamma_{k-1-i}, \ldots, \gamma_0$ are set equal to zero. With each value of k and i we have used constraints of the form (4.30) and we quote results for values of w running from 1.0 down to 0.1 in steps of 0.1.

It is mentioned previously that our intended starting point for the optimization process was to be $\psi(\xi) = \xi^k$. With most optimization problems it is very difficult if not impossible to guarantee that we have attained a global minimum rather than just a local minimum. One possible partial check on this is to take an alternative starting point and see whether in fact we do get the same minimum. This is however rather a negative test since even if we do get the same or a worse estimate then we can not draw any positive conclusions as to whether the original estimate is a global minimum, while if we obtain a better minimum then there is still no guarantee that this new estimate is a global minimum.

Computations were performed for various starting points selected from some of the results in the first appendix. It was found that there was often a large difference in the solutions obtained from the differing starting points illustrating the previous comments on local minima. In the results for each pair of values of k and c we quote only the best estimate we obtained for the minimum value of D, the error constant C_{p+1}

and the values of the coefficients $\gamma_{k-1}, \ldots, \gamma_{k-i}$ at the minimum, and the starting point for $\psi(\xi)$. We do not however claim that the estimate is necessarily a global minimum.

The results show that as the number of varying coefficients used in the optimization process increases so does the size of the stability region. It is certainly possible using this optimization technique to generate stiffly-stable second-derivative multistep methods which have much larger stablity regions than either those of Enright or any method resulting from the choice of (4.25) for $\psi(\xi)$.

4.4 Increasing the Order of Stiffly-Stable Second-Derivative Multistep Methods

All the stiffly-stable second-derivative multistep methods we have considered so far have had order k+2 for a k-step method. We might wish to question whether in fact this is the highest order we can obtain for a k-step method and still have a reasonably large stability region. As a first move towards looking at this problem we propose to investigate the possibility of obtaining stiffly-stable methods of order k + 3 where we will consider values of k between one and six thus giving us orders running from four to nine. To achieve this increase in order we must sacrifice one of the coefficients

$\alpha_0, \ldots, \alpha_{k-2}, \gamma_0, \ldots, \gamma_{k-1}$. Since there is no obvious choice of coefficient to use, for each value of k we take each one of the coefficients $\alpha_0, \ldots, \alpha_{k-2}, \gamma_0, \ldots, \gamma_{k-1}$ in turn setting the remainder to be zero and calculate the other coefficients $\alpha_k, \alpha_{k-1}, \beta_0, \ldots, \beta_k$ so that the resultant method is of order k + 3. We then determine whether this resultant method is zero-stable by determining the roots of the first characteristic polynomial, whether it is stable at infinity by determing the roots of the third characteristic polynomial, and whether it is stiffly-stable. Results of these tests for the various values of k and choices of coefficients are given in Table A1.8 in Appendix 1.

From the results we see that one particular choice for the additional coefficients always gives us the largest stability region. If we make use of γ_0 to increase the order then we find that the resultant methods of order k + 3 are always zero-stable, have a connected stability region and are stable at infinity when $k \geq 2$. When k is one the resultant multistep method, although A-stable, is not stable at infinity since a root of $\psi(\xi)$ is one. The following table gives the values of D, b_s and the error constant C_{p+1} for values of k from one to six when the coefficient γ_0 is used to increase the order.

Table 4.2

k	order	D	b_s	C_{p+1}	$D_{(ENR)}$	$C_{p+1(ENR)}$
1	4	A-stable		0.017	A-stable	0.039
2	5	0.27	2.02	0.014	0.1	0.022
3	6	0.89	1.83	0.011	0.52	0.014
4	7	1.87	1.86	0.008	1.4	0.010
5	8	3.45	1.96	0.0062	2.7	0.0074
6	9	6.16	1.94	-0.0015	5.3	0.0057

We have also given for comparison purposes the values of D and C_{p+1} for Enright's method of comparable order. The values of D for the new higher order method are larger than those of Enright although this is counterbalanced to a certain extent by the smaller error constant and use of a method which always has one step less than that of Enright.

In an attempt to test the possible usefulness of the new higher order methods some small scale preliminary tests were made with them on three linear problems: one quoted by Enright and the other two quoted in a report by Hull et al (1972). The three problems are:-

A) $$\underline{y}' = \begin{bmatrix} -0.1 & 0 & 0 & 0 \\ 0 & -10 & 0 & 0 \\ 0 & 0 & -100 & 0 \\ 0 & 0 & 0 & -1000 \end{bmatrix} \underline{y} \quad , \quad \underline{y}(0) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{range } [0,20]$$

This linear problem has real distinct eigenvalues (-.1,-10, -100,-1000) with a modest spread.

B) $$\underline{y}' = \begin{bmatrix} -1 & 10 & 0 & 0 \\ -10 & -1 & 0 & 0 \\ 0 & 0 & -100 & 100 \\ 0 & 0 & -100 & -100 \end{bmatrix} \underline{y}\ ,\ \underline{y}(0) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$ range $[0,20]$

This linear problem has complex eigenvalues $(-1\pm 10i, -100\pm 100i)$ with a small spread.

C) $$\underline{y}' = \begin{bmatrix} -10^4 & 10^3 & 0 & 0 \\ -10^3 & -10^4 & 0 & 0 \\ 0 & 0 & -10 & 100 \\ 0 & 0 & -100 & -10 \end{bmatrix} \underline{y}\ ,\ \underline{y}(0) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$ range $[0,20]$

This linear problem has complex eigenvalues $(-10^4\pm 10^3 i, -10\pm 100i)$ with a large spread.

Results were obtained for each of these three problems using both Enright's method and the new higher order method. The new method employs the same formulation and error strategies as Enright's published program in which his methods are incorporated in a variable-step variable-order manner with the maximum order restricted to six. Following the type of comparison statistics which are given by Hull et al we quote for each method the error tolerance τ, which is meant to bound the local error/unit step, the maximum local error/unit step M.L.E./U.S, the average error/unit step AE/US, and the total problem state time T.T. taken in seconds when using double precision on the IBM 360/67 at Newcastle University. In addition we use h to denote the stepsize, ENR to denote Enright's method, and NEW to denote the new higher order method. The following table, Table 4.3 gives the results obtained with these methods at three error tolerances, where both the maximum local error/unit step and the average error/unit step are given in units of the error tolerance τ.

Table 4.3

	τ	h	Method	M.I.E./U.S.	A.E./U.S.	T.T.
Problem A	10^{-2}	1	ENR	.611	.037	.15
		1	NEW	95.3	13.3	.30
		.0001	ENR	.217	.0058	1.19
		.0001	NEW	.111	.0038	.95
	10^{-4}	$.122\ 10^{-3}$	NEW	.171	.0083	1.52
		$.305\ 10^{-4}$	ENR	.477	.014	1.76
		$.305\ 10^{-4}$	NEW	.156	.0075	1.55
	10^{-6}	$.610\ 10^{-4}$	NEW	1.13	.021	2.44
		$.763\ 10^{-5}$	ENR	.764	.016	2.67
		$.763\ 10^{-5}$	NEW	.359	.012	2.54
Problem B	10^{-2}	$.781\ 10^{-2}$	NEW	.856	.066	1.56
		$.195\ 10^{-2}$	ENR	.495	.049	1.64
		$.195\ 10^{-2}$	NEW	.568	.073	1.57
	10^{-4}	$.195\ 10^{-2}$	NEW	.816	.043	2.90
		$.488.10^{-3}$	ENR	.806	.041	3.01
		$.488\ 10^{-3}$	NEW	.422	.036	2.98
	10^{-6}	$.488\ 10^{-3}$	NEW	.654	.037	5.66
		$.122\ 10^{-3}$	ENR	1.26	.040	5.80
		$.122\ 10^{-3}$	NEW	.654	.036	5.70
Problem C	10^{-2}	1.0	ENR	.214	.012	.14
		$.1\ 10^{-7}$	NEW	.490	.026	2.90
		$.1\ 10^{-7}$	ENR	.400	.025	3.01
	10^{-4}	$.191\ 10^{-5}$	ENR	1.57	.035	4.70
		$.1\ 10^{-7}$	NEW	.545	.031	4.89
		$.1\ 10^{-7}$	ENR	.539	.034	4.97
	10^{-6}	$.238\ 10^{-6}$	ENR	.552	.029	9.40
		$.1\ 10^{-7}$	NEW	.607	.040	8.92
		$.1\ 10^{-7}$	ENR	.620	.037	9.34

From this limited set of results it is impossible to draw any definite conclusions but we can make certain observations. In problem A we see that at the largest error tolerance 10^{-2} and using the largest stepsizes that the methods will accept, which in this case is h = 1 for both methods, the newer method compares poorly with Enright's method in terms of time taken and accuracy. If we take a smaller stepsize, h = .0001, then the new method is both quicker and more accurate than that of Enright. Considering next the behaviour of the two methods at the two smaller error tolerances we see that the new method can take a larger initial stepsize than that of Enright and is faster in time. At the tolerance of 10^{-4} the new method is more accurate but at the tolerance of 10^{-6} it is less accurate. If however we run the new method at these two smaller tolerances taking the same starting stepsize as Enright's method we find that the new method is both faster and more accurate.

On problem B the new method can take a larger starting stepsize than Enright's method at all three tolerances but although the new method is always quicker, at the largest tolerance 10^{-2} it is less accurate, at the middle tolerance 10^{-4} both methods are almost equally accurate while at the smallest tolerance 10^{-6} the new method is much more accurate. If we then run the new method at each tolerance taking the same starting stepsize as Enright's method we find that at the largest tolerance Enright's method is still more accurate but at both the smaller tolerances the new method is much more accurate and in all three cases it is faster.

On the third problem, which has a much larger stiffness ratio, Enright's method can take a larger stepsize at all three tolerances and gives good accuracy at this larger stepsize. If however we reduce the stepsize to a value which is acceptable to both methods then at all three tolerances the behaviour of the two methods is almost exactly similar in accuracy although the new method is slightly quicker.

From these results it would appear that the new method is worthy of further consideration particularly for problems where high accuracy is demanded. It is thought that the reason the new method behaves less well at the larger tolerance and on the third problem is due to the fact that the fourth-order formulae of the new method, which is the one it starts with, is not stable at infinity since the root of $\psi(\xi) = 0$ for this order is $\xi_{3,1} = 1$. If too large a stepsize is used for starting, this causes oscillations to occur in the solution. This type of behaviour is of course similar to that which can occur with the trapezoidal rule. Hence at larger error tolerances we would start with a larger stepsize and might then expect oscillations in the solution which is in fact the case. At the smaller error tolerances however, where we would start with a smaller stepsize, we tend to avoid this oscillatory behaviour except on the third problem which is fairly stiff and hence would require the use of a small stepsize to avoid these oscillations.

There are two ways by which one might overcome this problem. One way is to consider the possible use of a

smoothing type of formula in conjunction with the new method. This is a similar approach to that employed by Lindberg (1971) with the trapezoidal rule. An alternative way might be to combine the new method with that of Enright by making use of Enright's third and fourth-order formulae and then switching to the higher-order formulae in the new method because of their smaller error constant.

Chapter Five

Introduction

In the preceding chapter it was shown how the necessity to keep the total error from growing and the problems associated with the convergence of the corrector iteration led us to the use of second-derivative stiffly-stable multistep methods for solving stiff ordinary differential equations. We will illustrate in this chapter how the same line of reasoning leads us to propose the use of second-derivative Runge-Kutta formulae which are implicit with respect to y rather than k.

5.1 Weak Stability as applied to Runge-Kutta Formulae

The conventional explicit Runge-Kutta formulae suffer from the same weak stability disadvantages as explicit linear multistep methods. If we consider, as an example, the fourth-order explicit Runge-Kutta method

$$y_{n+1} - y_n = \frac{h}{6}(k_1+2k_2+2k_3+k_4) \tag{5.1}$$

where

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f(x_n+\tfrac{h}{2}, y_n + \tfrac{h}{2}k_1) \\ k_3 &= f(x_n+\tfrac{h}{2}, y_n + \tfrac{h}{2}k_2) \\ k_4 &= f(x_n+h, y_n+hk_3) \end{aligned} \tag{5.2}$$

and apply this method to solve the representative linear equation

$$y' = \lambda y \tag{5.3}$$

where λ is a complex constant with Re $\lambda < 0$, we obtain

$$y_{n+1} = R(\bar{h})y_n \tag{5.4}$$

where

$$\bar{h} = h\lambda \tag{5.5}$$

and

$$R(\bar{h}) = 1 + \bar{h} + \frac{\bar{h}^2}{2!} + \frac{\bar{h}^3}{3!} + \frac{\bar{h}^4}{4!} \tag{5.6}$$

The solution of (5.3) tends to zero as x increases and this can only occur in the numerical solution (5.4) if $R(\bar{h})$ is less than one in magnitude. It is easy to show that the region of $\bar{h}$ for which $|R(\bar{h})| < 1$ is small. On the negative real axis, for instance, the interval of absolute stability is (-2.7,0). The requirements of absolute stability again restrict us to the use of a small stepsize if $|\mathrm{Re}\ \lambda|$ is large.

A one-step method will always give us an equation of the form (5.4) for y_{n+1} where $R(\bar{h})$ is determined from the particular method. For an explicit method $R(\bar{h})$ will be a polynomial in $\bar{h}$ while for a method, which is implicit in y or k, $R(\bar{h})$ will be a rational function of $\bar{h}$.

The definitions of the various types of stability regions as given in Chapter Three are all applicable to a one-step method. The type of stability region which is usually sought for a one-step method however is given in the following definition proposed by Ehle (1969).

Definition 5.1

A one-step numerical method is said to be L-stable if it is A-stable and when applied to the linear representative equation $y' = \lambda y$, where λ is a complex constant with $\mathrm{Re}\ \lambda < 0$ it yields $y_{n+1} = R(\bar{h})y_n$, where $|R(\bar{h})| \to 0$ as $\mathrm{Re}\ \bar{h} \to -\infty$.

In terms of our previous definitions L-stability

corresponds to combining A-stability with stability at infinity. Since Lomax (1968) has shown that no explicit Runge-Kutta method can be A-stable we are therefore restricted in the search for L-stable methods to implicit Runge-Kutta methods.

The quantity $R(\bar{h})$ appearing in (5.4) is, for an implicit Runge-Kutta method of order p, a rational approximation to $e^{\bar{h}}$. Since the stability of the method is dependent upon the value of $R(\bar{h})$ we now quote several definitions and theorems related to rational approximants.

Definition 5.2

If $\bar{h}$ is a complex scalar and $R_{u,v,p}(\bar{h})$, $u \geq 0$, $v \geq 0$ is given by

$$R_{u,v,p}(\bar{h}) = \frac{\sum_{i=0}^{u} s_i (\bar{h})^i}{\sum_{i=0}^{v} t_i (\bar{h})^i} \quad \text{with } s_0 = t_0 = 1$$

where all the s_i and t_i are real then $R_{u,v,p}(\bar{h})$ is said to be a (u,v) rational approximation of order p to the exponential $e^{\bar{h}}$ if

$$R_{u,v,p}(\bar{h}) = e^{\bar{h}} + O(\bar{h}^{p+1})$$

The highest possible order is when $p = u + v$ and $R_{u,v,u+v}(\bar{h})$ is then called the (u,v) Padé approximant to $e^{\bar{h}}$ and will be denoted by $P_{u,v}(\bar{h})$. We have the following definition due to Ehle (1969).

Definition 5.3

A rational approximant $R_{u,v,p}$ is said to be

(i) A-acceptable if $|R_{u,v,p}(\bar{h})| < 1$ whenever $\mathrm{Re}\ \bar{h} < 0$

(ii) L-acceptable if it is A-acceptable and also satisfies

$|R_{u,v,p}(\bar{h})| \to 0$ as $Re(\bar{h}) \to -\infty$.

The following theorem, whose first part is due to Birkhoff and Varga (1965), and second part to Ehle (1969), gives some of the known information on Padé approximants.

Theorem 5.1

Let $P_{u,v}(\bar{h})$ be the (u,v) Padé approximant to $e^{\bar{h}}$. Then

(i) if $u = v$, $P_{u,v}$ is A-acceptable

(ii) if $v = u + 1$ or $v = u + 2$ then $P_{u,v}$ is L-acceptable.

If for a given implicit method we can show that $R(\bar{h})$ matches an appropriate Padé approximant then we can use Theorem 5.1 to establish whether the method is A -stable or L-stable.

Butcher (1963) has shown that there exist k-implicit Runge-Kutta methods of order 2R which Ehle subsequently proved were A-stable. Alternative high-order k-implicit methods have been proposed by Ehle (1969) and Chipman (1971). Ehle developed R-stage implicit methods of order 2R-1 which are L-stable and methods of order 2R-2 which are A-stable while Chipman proposed methods of order 2R-2 which are L-stable.

5.2 Iteration Processes for k-Implicit Runge-Kutta Formulae

Although one can obtain much higher-order formulae with k-implicit methods than with explicit methods,at each step of the calculation it is necessary to solve a set of R implicit non-linear equations for the k's. Butcher showed that these equations could be solved by the direct iteration method (1.37) provided condition (1.38) held. For stiff equations the Lipschitz constant will be large and so the

iteration (1.37) will only converge if h is very small. In an attempt to avoid this restriction on the stepsize an alternative form of iteration based on the Newton-Raphson method has been suggested. If we let $k_r^{(\nu)}$ be the approximation to k_r after the ν-th iteration the iteration is defined by

$$\left[1 - hb_{rr}\frac{\partial f}{\partial y}\left(x+ha_r, y+h\sum_{s=1}^{R} b_{rs}k_s^{(\nu)}\right)\right](k_r^{(\nu+1)} - k_r^{(\nu)})$$
$$- \frac{\partial f}{\partial y}\left(x+ha_r, y+h\sum_{s=1}^{R} b_{rs}k_s^{(\nu)}\right)\sum_{\substack{s=1\\ s\neq r}}^{R} hb_{rs}(k_s^{(\nu+1)} - k_s^{(\nu)})$$
$$= f\left(x+ha_r, y+h\sum_{s=1}^{R} b_{rs}k_s^{(\nu)}\right) - k_r^{(\nu)} \qquad r = 1,\ldots,R \qquad (5.7)$$

When we are dealing with a system of m first-order differential equations the iteration takes the form

$$\begin{bmatrix} I - hb_{11}J_1^{(\nu)} & - hb_{12}J_1^{(\nu)} & \cdots & - hb_{1R}J_1^{(\nu)} \\ - hb_{21}J_2^{(\nu)} & I - hb_{22}J_2^{(\nu)} & & \\ \vdots & \vdots & & \vdots \\ - hb_{R1}J_R^{(\nu)} & - & - & I - hb_{RR}J_R^{(\nu)} \end{bmatrix}$$

$$\begin{bmatrix} \underline{k}_1^{(\nu+1)} - \underline{k}_1^{(\nu)} \\ \vdots \\ \underline{k}_R^{(\nu+1)} - \underline{k}_R^{(\nu)} \end{bmatrix} = \begin{bmatrix} \underline{f}(x+ha_1, \underline{y}+h\sum_{s=1}^{R} b_{1s}\underline{k}_s^{(\nu)}) - \underline{k}_1^{(\nu)} \\ \vdots \\ \underline{f}(x+ha_R, \underline{y}+h\sum_{s=1}^{R} b_{Rs}\underline{k}_s^{(\nu)}) - \underline{k}_R^{(\nu)} \end{bmatrix} \qquad (5.8)$$

where $J_r^{(\nu)}$ is the matrix whose (i,j)th element is

$$\frac{\partial_i f}{\partial_j y}\left(x+ha_r, \underline{y}+h\sum_{s=1}^{R} b_{rs}\underline{k}_s^{(\nu)}\right)$$

Hence at each iteration we have to solve a set of mR linear simultaneous equations.

If $b_{rs} = 0$ for $r < s$ the resultant methods are called semi-explicit and the equations (5.8) can then be split into m sets of k linear equations. Rosenbrock (1963) proposed an alternative method which could be applied to stiff differential equations in autonomous form. His method which is based on the linearization of a semi-explicit method is

$$y_{n+1} = y_n + c_1k_1 + c_2k_2 + c_3k_3 \tag{5.9}$$

where

$$\left.\begin{aligned} k_1 &= h\left[f(y_n)+a_1\frac{\partial f}{\partial y}(y_n)k_1\right] \\ k_2 &= h\left[f(y_n+b_1k_1)+a_2\frac{\partial f}{\partial y}(y_n+c_1k_1)k_2\right] \\ k_3 &= h\left[f(y_n+b_1k_1+d_1k_2)+a_3\frac{\partial f}{\partial y}(y_n+c_2k_1+e_1k_2)k_3\right] \end{aligned}\right\} \tag{5.10}$$

Since each of the equations for k_i can be rearranged to give k_i explicitly there is no need to iterate. Rosenbrock gave two possible choices for the coefficients, one of which gives a method of order two while the other yields a third-order method. Haines (1969) and Calahan (1968) have also investigated this type of method.

5.3 Derivation of y-Implicit Second Derivative Runge-Kutta Formulae

In his method Rosenbrock made use of $\partial f/\partial y$. For an autonomous system we know that

$$\underline{y}'' = \frac{\partial \underline{f}}{\partial \underline{y}}\,\underline{y}'$$

and we now consider whether implicit Runge-Kutta methods

which incorporate the second derivative of y can be developed. In unpublished work Penna has developed some implicit Runge-Kutta methods which although they only employ values of f are implicit with respect to y rather than k while similar types of formulae have been suggested by Cash (1975). Since these methods are explicit with respect to k, at each step we need only solve a set of m simultaneous non-linear equations to obtain the value of y_{n+1}. Our aim is to derive a similar class of y-implicit Runge-Kutta methods which incorporate both the first and second derivatives of y. We will assume throughout that the differential equation, or system of equations, is in autonomous form.

Consider initially the method

$$y_{n+1} = y_n + c_1 h k_1 + c_2 h^2 \ell_1 + c_3 h k_2 \qquad c_1 \neq 0, c_2 \neq 0, c_3 \neq 0 \tag{5.11}$$

where

$$\left.\begin{aligned} k_1 &= f(y_{n+1}) \\ \ell_1 &= f^{(1)}(y_{n+1}) \\ k_2 &= f(y_n + a_2 h k_1 + a_3 h^2 \ell_1) \end{aligned}\right\} \tag{5.12}$$

We associate with the method defined by (5.11) and (5.12) the difference operator M defined by

$$\begin{aligned} M[y(x);h] = y(x+h) - y(x) - c_1 h f(y(x+h)) - c_2 h^2 f^{(1)}(y(x+h)) \\ - c_3 h f[y(x) + a_2 h f(y(x+h)) + a_3 h^2 f^{(1)}(y(x+h))] \end{aligned} \tag{5.13}$$

Assuming that y(x) is sufficiently differentiable and and recalling that we are interested in autonomous equations we can expand each of the terms on the right hand side of (5.13) in a Taylor series about x. Hence we have

$$y(x+h) = y(x) + h y'(x) + \frac{h^2}{2!} y^{(2)}(x) + \frac{h^3}{3!} y^{(3)}(x) + \frac{h^4}{4!} y^{(4)}(x) + \ldots\ldots \tag{5.14}$$

$$f(y(x+h))=y'(x)+hy^{(2)}(x)+\frac{h^2}{2!}y^{(3)}(x)+\frac{h^3}{3!}y^{(4)}(x)+\ldots\ldots \tag{5.15}$$

$$f^{(1)}(y(x+h))=y^{(2)}(x)+hy^{(3)}(x)+\frac{h^2}{2!}y^{(4)}(x)+\ldots\ldots \tag{5.16}$$

$$\begin{aligned} f[y(x)+a_2hf(y(x+h))+a_3h^2f^{(1)}(y(x+h))] \\ = y'(x)+[a_2hy'(x+h)+a_3h^2y^{(2)}(x+h)]f_y(y(x)) \\ +\frac{1}{2}[a_2hy'(x+h)+a_3h^2y^{(2)}(x+h)]^2f_{yy}(y(x)) \\ +\frac{1}{3!}[a_2hy'(x+h)+a_3h^2y^{(2)}(x+h)]^3f_{yyy}(y(x))+\ldots\ldots \end{aligned} \tag{5.17}$$

Substituting in (5.17) for $y'(x+h)$ and $y''(x+h)$ from (5.15) and (5.16) respectively and simplifying yields

$$\begin{aligned} f[y(x)+a_2hf(y(x+h))+a_3h^2f^{(1)}(y(x+h))] \\ = f+ha_2f^{(1)}+h^2[(a_2+a_3)f^{(1)}f_y+\frac{a_2^2}{2}f^2f_{yy}] \\ h^3[(\frac{a_2}{2}+a_3)f^{(1)}f_y^2+(\frac{a_2}{2}+a_3+a_2^2+a_2a_3)f^2f_yf_{yy}]+\frac{a_2^3}{6}f^3f_{yyy} \end{aligned} \tag{5.18}$$

where each term on the right hand side has $(y(x))$ as its argument.

Using (5.14), (5.15), (5.16) and (5.18) we obtain

$$\begin{aligned} M[y(x);h] = [1-c_1-c_3]hf + [\frac{1}{2}-c_1-c_2-c_3a_2]h^2f^{(1)} \\ +[\frac{1}{6}-\frac{c_1}{2}-c_2-c_3(a_2+a_3)]h^3f^{(1)}f_y \\ +[\frac{1}{6}-\frac{c_1}{2}-c_2-c_3\frac{a_2^2}{2}]h^3f^2f_{yy}+\ldots\ldots \end{aligned} \tag{5.19}$$

where again each term on the right hand side has $(y(x))$ as its argument. Defining the order of the method as being one less than the power of h in the first non-vanishing term in (5.19) we seek a third-order method by equating the coefficients of h, h^2 and h^3 to zero. This gives us the following equations

$$\left.\begin{aligned} c_1 \qquad + c_3 &= 1 \\ c_1 + c_2 + c_3 a_2 &= \frac{1}{2} \\ \frac{c_1}{2} + c_2 + c_3(a_2+a_3) &= \frac{1}{6} \\ \frac{c_1}{2} + c_2 + c_3\frac{a_2^2}{2} &= \frac{1}{6} \end{aligned}\right\} \qquad (5.20)$$

If we take a_2 as a parameter we can then solve for c_1, c_2, a_3 and c_2 in terms of a_2 to give us

$$\left.\begin{aligned} c_3 &= \frac{1}{3(a_2-1)^2} \\ c_1 &= 1 - \frac{1}{3(a_2-1)^2} \\ a_3 &= \frac{a_2^2 - 2a_2}{2} \\ c_2 &= -\frac{1}{2} - \frac{1}{3(a_2-1)} \end{aligned}\right\} \qquad (5.21)$$

Hence in terms of the parameter a_2 equations (5.12) become

$$\left.\begin{aligned} k_1 &= f(y_{n+1}) \\ \ell_1 &= f^{(1)}(y_{n+1}) \\ k_2 &= f(y_n + a_2 h k_1 + \frac{(a_2^2-2a_2)}{2} h^2 \ell_1) \end{aligned}\right\} \qquad (5.22)$$

and the resultant method is third-order.

Applying the method given by equations (5.11), (5.21) and (5.22) to the representative linear equation (5.3) yields

$$y_{n+1} = R(\bar{h})\, y_n$$

where

$$R(\bar{h}) = \frac{\left[1 + \frac{\bar{h}}{3(a_2-1)^2}\right]}{1-\bar{h}(1-\frac{1}{3(a_2-1)^2}) + \bar{h}^2(\frac{1}{2} + \frac{1}{3(a_2-1)^2}) - \bar{h}^3\frac{(a_2^2-2a_2)}{6(a_2-1)^2}} \qquad (5.23)$$

Since the (1,3) Padé approximant to $e^{\bar{h}}$ is

$$\frac{1+\bar{h}/4}{1-3\bar{h}/4+\bar{h}^2/4-\bar{h}^3/24} \tag{5.24}$$

we attempt to choose a value for the parameter a_2 such that (5.23) agrees with (5.24). If we compare coefficients of $\bar{h}$ in the numerator we find

$$a_2 = 1 + \frac{2}{3}\sqrt{3} \tag{5.25}$$

and this choice for a_2 also gives agreement for each of the coefficients of $\bar{h}$ in the denominator. Hence taking a_2 as given by (5.25) we can match the (1,3) Padé approximant which from Theorem 5.1 is known to be L-acceptable. Our third-order L-stable method is thus

$$y_{n+1}=y_n+\frac{3}{4}hk_1-\left(\frac{1}{2}+\frac{1}{\sqrt{12}}\right)h^2\ell_1+\frac{1}{4}hk_2 \tag{5.26}$$

where

$$\left.\begin{aligned} k_1 &= f(y_{n+1}) \\ \ell_1 &= f^{(1)}(y_{n+1}) \\ k_2 &= f\left(y_n+\left(1+\frac{2\sqrt{3}}{3}\right)hk_1+\frac{1}{6}h^2\ell_1\right) \end{aligned}\right\} \tag{5.27}$$

This method is explicit with respect to k but implicit in terms of y_{n+1}.

We now turn our attention to deriving a fourth-order method. If we consider

$$y_{n+1}=y_n+c_1hk_1+c_2h^2\ell_1+c_3hk_2+c_4hk_3 \tag{5.28}$$

where

$$\left.\begin{aligned} k_1 &= f(y_{n+1}) \\ \ell_1 &= f^{(1)}(y_{n+1}) \\ k_2 &= f(y_n+a_2hk_1+a_3h^2\ell_1) \\ k_3 &= f(y_n+b_2hk_1+b_3hk_2+b_4h^2\ell_1) \end{aligned}\right\} \tag{5.29}$$

We associate with the method (5.28) the difference operator M defined by

$$M[y(x);h] = y(x+h)-y(x)-c_1hf(y(x+h))-c_2h^2f^{(1)}(y(x+h))$$
$$- c_3hf[y(x)+a_2hf(y(x+h))+a_3h^2f^{(1)}(y(x+h))]$$
$$- c_4hf[y(x)+b_2hf(y(x+h))+b_3hf(y(x)+a_2hf(y(x+h))$$
$$+a_3h^2f^{(1)}(y(x+h)))+b_4h^2f^{(1)}(y(x+h))] \qquad (5.30)$$

Assuming that $y(x)$ is sufficiently differentiable and again recalling that we are interested in autonomous equations we proceed as before and expand each of the terms on the right hand side of (5.30) in a Taylor series about x. Since we already have the expansions of most of these terms the only new expansion we need is the one for the last term on the right hand side. Thus we have after simplification

$$f[y(x)+b_2hf(y(x+h))+b_3hf(y(x)+a_2hf(y(x+h))+a_3h^2f^{(1)}(y(x+h)))$$
$$+b_4h^2f^{(1)}(y(x+h))]$$
$$= f+h(b_2+b_3)f^{(1)}+h^2[(b_2+b_3a_2+b_4)f^{(1)}f_y+\frac{(b_2+b_3)^2}{2}f^2f_{yy}]$$
$$+ h^3[f^{(1)}f_y^2(\frac{b_2}{2}+b_3(a_2+a_3)+b_4)+f^2f_yf_{yy}(\frac{b_2}{2}+b_3\frac{a_2^2}{2}+b_4$$
$$+(b_2+b_2)(b_2+b_3a_2+b_4))+\frac{(b_2+b_2)^3}{6}f^3f_{yyy}]+..... \qquad (5.31)$$

where again each term on the right hand side has argument $(y(x))$.

Using (5.14), (5.16), (5.18) and (5.31) we obtain

$$
\begin{aligned}
M[y(x);h] = {} & [1-c_1-c_3-c_4]hf + [\tfrac{1}{2}-c_1-c_2-c_3a_2-c_4(b_2+b_3)]h^2f^{(1)} \\
& + [\tfrac{1}{6}-\tfrac{c_1}{2}-c_2-c_3(a_2+a_3)-c_4(b_2+b_3a_2+b_4)]h^3f^{(1)}f_y \\
& + [\tfrac{1}{6}-\tfrac{c_1}{2}-c_2-c_3\tfrac{a_2^2}{2}-\tfrac{c_4}{2}(b_2+b_3)^2]h^3f^2f_{yy} \\
& + [\tfrac{4}{24}-\tfrac{4}{6}c_1-4\tfrac{c_2}{2}-c_3(\tfrac{a_2}{2}+a_3+a_2^2+a_2a_3) \\
& \quad -c_4(\tfrac{b_2}{2}+b_3\tfrac{a_2^2}{2}+b_4+(b_2+b_3)(b_2+b_3a_2+b_4))]h^4f^2f_yf_{yy} \\
& + [\tfrac{1}{24}-\tfrac{c_1}{6}-\tfrac{c_2}{2}-c_3(\tfrac{a_2}{2}+a_3)-c_4(\tfrac{b_2}{2}+b_3(a_2+a_3)+b_4)]h^4f^{(1)}f_y^2 \\
& + [\tfrac{1}{24}-\tfrac{c_1}{6}-\tfrac{c_2}{2}-c_3\tfrac{a_2^3}{6}-\tfrac{c_4}{6}(b_2+b_3)^3]h^4f^3f_{yyy}+\ldots
\end{aligned}
\tag{5.32}
$$

with each term on the right hand side having the argument $(y(x))$.

We seek a fourth-order method by equating the coefficients of h, h^2, h^3 and h^4 to zero. This gives us the following equations:

$$
\left.
\begin{aligned}
& c_1+c_3+c_4=1 \\
& c_1+c_2+c_3a_2+c_4(b_2+b_3)=\tfrac{1}{2} \\
& \tfrac{c_1}{2}+c_2+c_3\tfrac{a_2^2}{2}+\tfrac{c_4}{2}(b_2+b_3)^2=\tfrac{1}{6} \\
& \tfrac{c_1}{2}+c_2+c_3(a_2+a_3)+c_4(b_2+b_3a_2+b_4)=\tfrac{1}{6} \\
& 4\tfrac{c_1}{6}+4\tfrac{c_2}{2}+c_3(\tfrac{a_2}{2}+a_3+a_2^2+a_2a_3)+c_4(\tfrac{b_2}{2}+b_3\tfrac{a_2^2}{2}+b_4+(b_2+b_3)(b_2+b_3a_2+b_4))=\tfrac{4}{24} \\
& \tfrac{c_1}{6}+\tfrac{c_2}{2}+c_3(\tfrac{a_2}{2}+a_3)+c_4(\tfrac{b_2}{2}+b_3(a_2+a_3)+b_4)=\tfrac{1}{24} \\
& \tfrac{c_1}{6}+\tfrac{c_2}{2}+c_3\tfrac{a_2^3}{6}+\tfrac{c_4}{6}(b_2+b_3)^3=\tfrac{1}{24}
\end{aligned}
\right\}
\tag{5.33}
$$

Thus we have a set of seven non-linear equations in nine unknowns. If we assume

$$\frac{a_2^2}{2} = a_2 + a_3 \tag{5.34}$$

and

$$(b_2+b_3)^2 = b_2+b_3a_2+b_4 \tag{5.35}$$

then setting

$$d_1 = b_2 + b_3 \tag{5.36}$$

and using the conditions (5.34) and (5.35) to simplify equations (5.33) we obtain

$$\left.\begin{aligned}
&c_1 + c_3 + c_4 = 1 \\
&c_1 + c_2 + c_3 a_2 + c_4 d_1 = \tfrac{1}{2} \\
&\tfrac{c_1}{2} + c_2 + c_3 a_2^2 + \tfrac{c_4}{2} d_1^2 = \tfrac{1}{6} \\
&\tfrac{c_1}{6} + c_2 + c_3 \tfrac{a_2^3}{6} + \tfrac{c_4}{6} d_1^3 = \tfrac{1}{24} \\
&b_2 + b_3(a_2+a_3) + b_4 = \frac{\left[\tfrac{1}{24} - \tfrac{c_1}{6} - \tfrac{c_2}{2} - c_3\left(\tfrac{a_2^2 - a_2}{2}\right)\right]}{c_4} \\
&b_2 + b_3 a_2 + b_4 = \tfrac{d_1^2}{2} \\
&b_2 + b_3 = d_1
\end{aligned}\right\} \tag{5.37}$$

Our interest lies in obtaining a method which is L-stable. If we consider applying the method defined by equations (5.28), (5.29) and (5.37) to the representative linear equation (5.3) it can be shown that

$$y_{n+1} = R(\bar{h})y_n$$

where

$$R(\bar{h}) = \frac{\left[1+\bar{h}(c_3+c_4)+\bar{h}^2 c_4 b_3\right]}{\left[1-(\bar{h}c_1+\bar{h}^2(c_3a_1+c_4b_2+c_2)+\bar{h}^3(c_3a_3+c_4b_3a_1+c_4b_4)+\bar{h}^4 c_4 b_3 a_3)\right]} \tag{5.38}$$

while the (2.4) Pade approximant to $e^{\bar{h}}$ is

$$\frac{1+\frac{\bar{h}}{3}+\frac{\bar{h}^2}{20}}{1-\frac{2}{3}\bar{h}+\frac{\bar{h}^2}{5}-\frac{\bar{h}^3}{30}+\frac{\bar{h}^4}{360}} \tag{5.39}$$

If (5.38) and (5.39) are to correspond we must have

$$c_1 = 2/3 \tag{5.40}$$

and

$$a_3 = -1/12 \tag{5.41}$$

while from (5.34) we can deduce that

$$a_2 = 1 + \sqrt{5/6} \tag{5.42}$$

Taking the first four equations in (5.37) it can be shown after some algebraic manipulation that

$$d_1^2[12(c_1-1)(a_2-1)^2+4] + d_1[-24(c_1-1)(a_2-1)^2+4a_2-9] + 12c_1(a_2-1)^2+4a_2^2-9a_2+6 = 0 \tag{5.43}$$

With c_1 and a_2 given by (5.40) and (5.42) we can solve the quadratic equation (5.43) for d_1. The first three equations of (5.37) are then linear simultaneous equations in the unknowns c_2, c_3 and c_4 and can be solved as such. Finally the last three equations in (5.37) can be solved as linear simultaneous equations for the unknowns b_2, b_3 and b_4. We can then check whether the solution gives agreement between (5.38) and (5.39) and this procedure has been carried out numerically using double precision. If we take the negative square root for a_2 in (5.42) then equation (5.43) has

complex roots for d_1. We therefore take the positive square root for a_2 and obtain agreement between (5.38) and (5.39) with the following set of coefficients.

$$\left.\begin{array}{llll} a_3 = -0.8333 & 3333 & 3333 & 3333\times10^{-1} \\ a_2 = 1.9128 & 7092 & 9175 & 276 \\ b_2 = -0.1362 & 7939 & 3451 & 9903 \\ b_3 = 0.1198 & 6226 & 6084 & 0889 \\ b_4 = -0.9286 & 6889 & 8098 & 2830\times10^{-1} \\ c_1 = 0.6666 & 6666 & 6666 & 6667 \\ c_2 = -0.2677 & 6114 & 1824 & 5271 \\ c_3 = 0.5523 & 6360 & 6801 & 6865\times10^{-1} \\ c_4 = 0.2780 & 9697 & 2653 & 1645 \end{array}\right\} \quad (5.44)$$

Since (5.43) is a quadratic equation it must have another real solution for d_1 but this solution is omitted because some of the associated coefficients are large in magnitude (for example, b_2 would be less than -55).

We have thus succeeded in deriving an L-stable fourth order Runge-Kutta method which matches the (2.4) Padé approximant and we note that this same technique can be employed to derive methods for which $R(\bar{h})$ matches the (2.3) Padé approximant.

5.4 Iteration Process for y-Implicit Runge-Kutta Formulae

The iteration process employed in the y-implicit formulae to find y_{n+1} is the Newton-Raphson method. We illustrate this by showing how it is applied to the fourth-order method. From (5.28) we set

$$F(y_{n+1}) = y_{n+1} - y_n - (c_1hk_1+c_2h^2\ell_1+c_3hk_2+c_4hk_3) \quad (5.45)$$

Differentiating (5.45) with respect to y_{n+1} we obtain

$$\frac{dF}{dy}(y_{n+1}) = 1-(c_1 h\frac{dk_1}{dy_{n+1}} + c_2 h^2\frac{d\ell_1}{dy_{n+1}} + c_3 h\frac{dk_2}{dy_{n+1}} + c_4 h\frac{dk_3}{dy_{n+1}}) \tag{5.46}$$

If we set

$$\begin{aligned} J_1 &= \left[\frac{df(y)}{dy}\right]_{y=y_{n+1}} \\ J_2 &= \left[\frac{df(y)}{dy}\right]_{y=y_n+a_2hk_1+a_3h^2\ell_1} \\ J_3 &= \left[\frac{df(y)}{dy}\right]_{y=y_n+b_2hk_1+b_3hk_2+b_4h^2\ell_1} \end{aligned} \tag{5.47}$$

then using (5.29) and (5.47) gives

$$\begin{aligned} \frac{dk_1}{dy_{n+1}} &= J_1 \\ \frac{d\ell_1}{dy_{n+1}} &= J_1^2 \\ \frac{dk_2}{dy_{n+1}} &= J_2(a_2hJ_1+a_3h^2J_1^2) \\ \frac{dk_3}{dy_{n+1}} &= J_3(b_2hJ_1+b_3hJ_2(a_2hJ_1+a_3h^2J_1^2)+b_4h^2J_1^2) \end{aligned} \tag{5.48}$$

where we follow Liniger and Willoughby (1968) and Enright (1972) and ignore all derivatives higher than the first. Substituting from (5.48) into (5.46) and simplifying gives

$$\begin{aligned} \frac{dF}{dy}(y_{n+1}) = 1-[&c_1hJ_1+h^2(c_2J_1^2+c_3a_2J_1J_2+c_4b_2J_1J_3) \\ &+h^3(c_3a_3J_1^2J_2+c_4b_3a_2J_1J_2J_3+c_4b_4J_1^2J_3)+h^4J_1^2J_2J_3c_{4}b_{3}a_{3}] \end{aligned} \tag{5.49}$$

Hence our iteration scheme is

$$\frac{dF}{dy}(y_{n+1}^{(\nu)}) \cdot (y_{n+1}^{(\nu+1)}-y_{n+1}^{(\nu)}) = -F(y_{n+1}^{(\nu)}) \tag{5.50}$$

where $y_{n+1}^{(\nu)}$ is the value obtained for y_{n+1} after the ν-th iteration.

5.5 Numerical Testing of the y-Implicit Runge-Kutta Formulae

Although the theory in the previous two sections has been developed in terms of a single differential equation it is straightforward to show that the theory carries across to systems of equations. In order to test the y-implicit formulae we will attempt to find a numerical solution to the same three systems of stiff differential equations given in Chapter Four (p. 55-56). For comparison purposes we will also find a solution using the fourth-order k-implicit Runge-Kutta method proposed by Butcher (1964). Both methods are programmed so that they employ a one full step, two half step strategy to estimate the local error and hence regulate the stepsize. Thus if E_{est} denotes the absolute value of the difference between the two estimates for y_{n+1} and τ is a prescribed local error tolerance then the stepsize is halved if

$$\frac{E_{est}}{30h} > \tau \,, \tag{5.51}$$

the stepsize is doubled if

$$\frac{E_{est}}{30h} < \frac{\tau}{64} \,, \tag{5.52}$$

and otherwise the stepsize is left unchanged. The iteration process is judged to have converged if the maximum difference between any two corresponding components in the latest two iterates is less than $2h\tau$ while we attempt to keep $\frac{\partial F}{\partial y}_{n+1}$ constant for as long as possible by allowing up to four iterations before it is re-evaluated.

In the following table, Table 5.1, we quote for each of the three problems the error tolerance τ, the starting stepsize

the maximum local error/unit step M.L.E./U.S. and the average error/unit step A.E./U.S. both in units of the error tolerance and the total problem state time T.T. taken in seconds using double precision on the I.B.M. 370/168 at Newcastle University. We use y-IMP to denote the fourth-order y-implicit method and BUT to denote Butcher's method.

Table 5.1

	τ	h	Method	M.L.E./U.S.	A.E./U.S.	T.T.
Problem A						
	10^{-2}	1	y-IMP	$.743\times10^{-2}$	$.299\times10^{-3}$	.0316
		1	BUT	95.3	17.7	.0669
	10^{-4}	$.244\times10^{-3}$	y-IMP	.136	$.266\times10^{-2}$	.135
		$.122\times10^{-3}$	BUT	.387	$.208\times10^{-1}$	.374
	10^{-6}	$.244\times10^{-3}$	y-IMP	.290	$.853\times10^{-2}$	.186
		$.610\times10^{-4}$	BUT	1.13	$.441\times10^{-1}$	.807
Problem B						
	10^{-2}	.125	y-IMP	.772	$.275\times10^{-1}$	.0777
		$.781\times10^{-2}$	BUT	.856	$.502\times10^{-1}$	.242
	10^{-4}	$.195\times10^{-2}$	y-IMP	.268	$.159\times10^{-1}$	.215
		$.195\times10^{-2}$	BUT	.816	$.510\times10^{-1}$	.667
	10^{-6}	$.195\times10^{-2}$	y-IMP	.378	$.239\times10^{-1}$	.415
		$.488\times10^{-3}$	BUT	.588	$.593\times10^{-1}$	1.92
Problem C						
	10^{-2}	1	y-IMP	$.656\times10^{-2}$	$.172\times10^{-2}$	.0244
		$.305\times10^{-4}$	BUT	.765	$.429\times10^{-1}$	.506
	10^{-4}	1	y-IMP	.359	$.241\times10^{-1}$	.0249
		$.763\times10^{-5}$	BUT	.593	$.493\times10^{-1}$	1.29
	10^{-6}	$.153\times10^{-4}$	y-IMP	.469	$.229\times10^{-1}$	.705
		$.191\times10^{-5}$	BUT	.673	$.563\times10^{-1}$	2.45

From the results in Table 5.1 it can be seen that on all three problems the y-implicit method is much quicker, has a smaller maximum local error/unit step, has a smaller average error/unit step and usually takes a much larger starting stepsize than Butcher's k-implicit method.

On these three problems therefore the y-implicit method gives a significant improvement on Butcher's method. We note however that all these three problems are linear and it would be very dangerous to draw any firm conclusions without testing the y-implicit method on various non-linear problems, since it is only on non-linear problems that we will be able to evaluate the effects of the iteration processes. This is a topic on which we are currently engaged in further investigations.

Chapter Six

6.1 A Review of Methods based on Non-Polynomial Interpolants

The underlying assumption in linear multistep methods is that the behaviour of the theoretical solution can be adequately represented by a polynomial. When we try to raise the order of a method this is equivalent to approximating the theoretical solution by a higher degree polynomial in the hope that we will obtain a better representation and hence a more accurate numerical solution. For many practical problems the polynomial assumption is not realistic. An instance of this occurs when the solution is oscillatory and we might then profit by assuming a representation involving sines and cosines. A similar situation occurs when the solution appears to be increasing or decreasing exponentially. In this case a representation involving exponential functions might be beneficial. Another time when a non-polynomial approximant may be used is when the solution appears to be approaching a singularity.

Representation of the solution in terms of sines and cosines was first proposed by Gautschi (1961) who was interested in problems which he knew had an oscillatory solution and where an estimate of the frequency was available in advance. For this type of problem he therefore derived trigonometric multistep methods which integrated exactly trigonometric polynomials of a given order and in addition he proved existence theorems for these methods. Later work in this same field was carried out by Bettis and Stiefel (1969). They were concerned with an orbit problem in celestial mechanics

and developed a technique applicable to second-order differential equations of the form $y^{(2)} = f(x,y)$ where by choosing in advance a set of frequencies w_k, $k = 1,2,3$ they derived a trigonometric multistep method of Störmer type [Henrici (1962)] which would integrate exactly $\cos w_k t$ and $\sin w_k t$, $k = 1,2,3$. By considering various special cases, such as confluent or zero frequencies, methods are also derived which will integrate exactly the product of algebraic polynomials with trigonometric polynomials. In a subsequent paper Bettis (1969) developed the technique both by increasing the total number of frequencies one could incorporate and by extending the idea to modify other types of multistep method to trigonometric multistep methods with some of the modified trigonometric methods being applicable to first-order ordinary differential equations. Lyche (1972) extended some of the basic theory of linear multistep methods to include smooth, stepsize-dependent coefficients. In particular he dealt with methods based on trigonometric and exponential functions and also the product of these functions with algebraic polynomials.

Exponential fitting of the solution had first been suggested by Brock and Murray (1952). It has subsequently been proposed by Fowler and Warten (1967), and Liniger and Willoughby (1967) all of whom thought that it might be used in the numerical solution of stiff equations. Makela et al (1971) have also proposed the use of exponentially fitted

multistep methods and have proved the existence and convergence of certain classes of method.

There appears to have been little work previously done on those problems which have a singularity in the solution. Multistep methods tend to give poor results if they are used too close to a singularity. Lambert and Shaw (1965) therefore proposed representation of the solution by a rational function in the belief that this would give a better representation near the singularity and they derived various explicit and implicit methods based on rational functions. In a later paper, Lambert and Shaw (1966a), they suggested a representation using a more general function of the form

$$\left.\begin{array}{ll} \sum_{i=0}^{P} a_i x^i + b|A+x|^N & N\notin\{0,1,2,\ldots,P\} \\ \sum_{i=0}^{P} a_i x^i + b|A+x|^N \log|A+x| & N\in\{0,1,\ldots,P\} \end{array}\right\} \qquad (6.1)$$

where P is a positive integer, $a_i, i=0,1,\ldots,P$ and b are regarded as undetermined coefficients, and A and N are regarded as parameters which can if necessary be determined adaptively by their method while carrying out the numerical solution. Their resultant methods are one-step explicit or implicit methods which necessitate the use of higher derivatives up to order (P+2) if both A and N are known in advance, up to order (P+3) if only one of A and N is known and up to order (P+4) if neither is known. To avoid the labour involved in calculating these higher derivatives Shaw (1967) subsequently derived multistep methods based on the same representation (6.1).

In yet another paper Lambert and Shaw (1966b) proposed explicit and implicit multistep methods based on a representation of the form

$$\sum_{i=0}^{P-1} a_i x^i + b\pi(x) \tag{6.2}$$

where b is constant and $\pi(x)$ is any arbitrary function assumed sufficiently differentiable. The resultant multistep methods can be generated from polynomial based formulae by means of recurrence relations.

This chapter is concerned with extending this last piece of work by Lambert and Shaw so that the basic representation takes the form

$$\sum_{i=0}^{P-N} a_i x^i + \sum_{j=1}^{N} b_j \pi_j(x) \tag{6.3}$$

where again each b_j is a constant and each $\pi_j(x)$ is an arbitrary function assumed sufficiently differentiable.

We will be considering the class of multistep difference formula

$$\sum_{s=0}^{\ell} \sum_{j=0}^{k} a_{sj} h^s y_{n+j}^{(s)} = 0 \qquad k \geq 1,\ \ell \geq 1. \tag{6.4}$$

To ensure consistency we specify

$$\sum_{j=0}^{k} a_{0j} = 0. \tag{6.5}$$

With the class (6.4) we associate the operator $\mathcal{L}_{k\ell}$ given by

$$\mathcal{L}_{k\ell}[z(x); a_{sj}] = \sum_{s=0}^{\ell} \sum_{j=0}^{k} a_{sj} h^s z^{(s)}(x+jh) \qquad k \geq 1, \ell \geq 1. \tag{6.6}$$

where $z(x)$ is an arbitrary function assumed sufficiently differentiable.

The operator (6.6) is then said to have order p if the coefficients a_{sj} are such that

$$\mathcal{L}_{k\ell}[z(x); a_{sj}] = C_{p+1} h^{p+1} z^{(p+1)}(x) + O(h^{p+2}) \qquad C_{p+1} \neq 0 \tag{6.7}$$

where C_i is the coefficient of $h^i z^{(i)}(x)$ in the Taylor expansion of (6.6) about the point x.

Noting that

$$\mathcal{L}_{k\ell}[x^p; a_{sj}] = 0 \qquad p \geq 0 \tag{6.8}$$

we can then say that the multistep method (6.4) will give us the exact answer (in the absence of roundoff errors) if the solution is a polynomial of degree less than or equal to p.

We also define the linear operators $\mathcal{P}_k$ and $\mathcal{R}_{k\ell}$ by

$$\mathcal{P}_k[z(x); a_{oj}] = -z(x+kh) + (1 - \sum_{j=0}^{k-2} a_{oj}) z(x+(k-1)h) + \sum_{j=0}^{k-2} a_{oj} z(x+jh) \tag{6.9}$$

and

$$\mathcal{R}_{k\ell}[z(x); a_{sj}] = \sum_{s=1}^{\ell} \sum_{j=0}^{k} a_{sj} h^s z^{(s)}(x+jh) \qquad a_{sk} \neq 0,\ s = 1,2,\ldots,\ell \tag{6.10}$$

where again $z(x)$ is assumed sufficiently differentiable.

6.2 Implicit Non-Polynomial Operators

Initially we will be considering implicit operators and we therefore define the implicit operator $\mathcal{A}_{k\ell}$ applied to any function $z(x)$ by

$$\mathcal{A}_{k\ell}[z(x); a_{sj}] = \mathcal{P}_k[z(x); a_{oj}] + \mathcal{R}_{k\ell}[z(x); a_{sj}] \tag{6.11}$$

where the $\ell(k+1)$ coefficients $a_{sj}, 1 \leq s \leq \ell, 0 \leq j \leq k$ can be determined as functions of the stability

parameters $a_{0j}, 0 \leq j \leq k-2$ by requiring

$$\mathcal{A}_{k\ell}[x^r; a_{sj}] = 0 \qquad r = 1,2,\ldots,P \tag{6.12}$$

where

$$P = \ell(k+1) \tag{6.13}$$

The values a_{sj} which satisfy (6.12) are written as

$$\bar{a}_{sj}(a_{0j}, k, \ell) \qquad 1 \leq s \leq \ell,\ 0 \leq j \leq k \tag{6.14}$$

which for notational simplicity we will write as $\bar{a}_{sj}$. The resultant operator $\bar{\mathcal{A}}_{k\ell}$, which is zero-stable, is then referred to as the stable (k,ℓ) implicit operator. Thus

$$\bar{\mathcal{A}}_{k\ell}[z(x)] = \mathcal{A}_{k\ell}[z(x); \bar{a}_{sj}] \qquad \ell \geq 1 \tag{6.15}$$

If however we set

$$\mathcal{A}_{k\ell}[x^r; a_{sj}] = 0 \qquad r = 1,2,\ldots,Q-2 \tag{6.16}$$

where

$$Q = (\ell+1)(k+1) \tag{6.17}$$

then we could have solved for $a_{0j}, 0 \leq j \leq k-2$ in addition to the a_{sj}, $1 \leq s \leq \ell$, $0 \leq j \leq k$ and the values of a_{0j} and a_{sj} which satisfy (6.16) are written as

$$\bar{\bar{a}}_{sj}(k,\ell) \qquad 0 \leq s \leq \ell,\ 0 \leq j \leq k \tag{6.18}$$

which for notational simplicity we will write as $\bar{\bar{a}}_{s_j}$. The resultant operator $\bar{\bar{\mathcal{A}}}_{k\ell}$, which is generally not zero-stable, is then referred to as the optimum (k,ℓ) implicit operator. Thus

$$\bar{\bar{\mathcal{A}}}_{k\ell}[z(x)] = \mathcal{A}_{k\ell}[z(x); \bar{\bar{a}}_{s_j}] \qquad \ell \geq 0,\ k \geq 1. \tag{6.19}$$

We would now like to derive multistep difference formulae based on the non-polynomial representation (6.3).

For methods which are based on polynomials the coefficients a_{s_j} are real constants but for a non-polynomial representation the coefficients will usually be functions of x. Since the derivation of the recurrence formulae which will appear later in this chapter are long and involve a complicated notational system we attempt to prepare the reader for what will follow by first giving a derivation of the recurrence formulae necessary to determine a particular non-polynomial based implicit operator.

Consider then the zero-stable, implicit, non-polynomial operator $\bar{\mathcal{A}}^{(2)}_{31}$ where

$$\bar{\mathcal{A}}^{(2)}_{31}[z(x); \bar{a}^{(2)}_{s_j}] = \mathcal{P}_3[z(x); \bar{a}_{o_j}] + \mathcal{R}_{31}[z(x); \bar{a}^{(2)}_{i_j}] \tag{6.20}$$

and the coefficients $\bar{a}^{(2)}_{i_j}$, $0 \leq j \leq k$ are determined as functions of the stability parameters $\bar{a}_{o_j}$ by making

$$\bar{\mathcal{A}}_{31}^{(2)}[x^r; \bar{a}_{sj}^{(2)}] = 0 \qquad r = 1,2 \tag{6.21}$$

and

$$\bar{\mathcal{A}}_{31}^{(2)}[\pi_i(x); \bar{a}_{sj}^{(2)}] = 0 \qquad i = 1,2. \tag{6.22}$$

Similarly the zero-stable implicit non-polynomial operator $\bar{\mathcal{A}}_{31}^{(1)}$ is defined by

$$\bar{\mathcal{A}}_{31}^{(1)}[z(x); \bar{a}_{sj}] = \mathcal{P}_3[z(x); \bar{a}_{oj}] + \mathcal{R}_{31}[z(x); \bar{a}_{ij}^{(1)}] \tag{6.23}$$

where the coefficients $\bar{a}_{ij}^{(1)}$, $0 \leq j \leq 3$ are determined as functions of the stability parameters $\bar{a}_{oj}$ by requiring

$$\bar{\mathcal{A}}_{31}^{(1)}[x^r; \bar{a}_{sj}^{(1)}] = 0 \qquad r = 1,2,3 \tag{6.24}$$

and

$$\bar{\mathcal{A}}_{31}^{(1)}[\pi_i(x); \bar{a}_{sj}^{(1)}] = 0. \tag{6.25}$$

In addition we will also need the optimum implicit non-polynomial operator $\bar{\bar{\mathcal{A}}}_{30}^{(1)}$ as defined by

$$\bar{\bar{\mathcal{A}}}_{30}^{(1)}[z(x); \bar{\bar{a}}_{oj}^{(1)}] = \mathcal{P}_3[z(x); \bar{\bar{a}}_{oj}^{(1)}] \tag{6.26}$$

where the coefficients $\bar{\bar{a}}_{oj}^{(1)}$ are determined by making

$$\bar{\bar{\mathcal{A}}}^{(1)}_{30}[x;\bar{\bar{a}}^{(1)}_{oj}] = 0 \tag{6.27}$$

and

$$\bar{\bar{\mathcal{A}}}^{(1)}_{30}[\pi_1(x);\bar{\bar{a}}^{(1)}_{oj}] = 0\,. \tag{6.28}$$

Setting

$$\bar{a}^{(2)}_{sj} = \bar{a}^{(1)}_{sj} + \bar{c}_{sj} \qquad 0 \le j \le 3 \tag{6.29}$$

where

$$\bar{c}_{oj} = 0 \qquad 0 \le j \le 3 \tag{6.30}$$

and substituting for $\bar{a}^{(2)}_{ij}$ in (6.20) gives

$$\begin{aligned}\bar{\mathcal{A}}^{(2)}_{31}[z(x);\bar{a}^{(2)}_{sj}] &= \mathcal{P}_3[z(x);\bar{a}_{oj}] + \mathcal{R}_{31}[z(x);\bar{a}^{(1)}_{ij}+\bar{c}_{ij}] \\ &= \bar{\mathcal{A}}^{(1)}_{31}[z(x);\bar{a}^{(1)}_{sj}] + \mathcal{R}_{31}[z(x);\bar{c}_{ij}]\,.\end{aligned} \tag{6.31}$$

Since $\bar{\mathcal{A}}^{(1)}_{31}[x^\tau;\bar{a}^{(1)}_{sj}] = 0$ for $\tau = 1,2$ (6.21) implies

$$\mathcal{R}_{31}[x^\tau;\bar{c}_{ij}] = 0 \qquad \tau = 1,2 \tag{6.32}$$

while from (6.22) we have

$$\mathcal{R}_{31}[\pi_1(x);\bar{c}_{ij}] = 0\,. \tag{6.33}$$

Defining

$$\bar{d}_{0j} = -\bar{c}_{1j} / \bar{c}_{13} \qquad 0 \le j \le 3 \ , \ c_{13} \neq 0 \tag{6.34}$$

means that (6.32) and (6.33) are equivalent to the conditions

$$-\bar{c}_{13} h \sum_{j=0}^{3} \bar{d}_{0j} z(x+jh) = 0 \qquad z(x) = 1, x, \pi_1'(x) . \tag{6.35}$$

Since $\bar{d}_{03} = -1$ from (6.34) while (6.35) with $z(x) = 1$ implies $\bar{d}_{02} = 1 - \sum_{j=0}^{1} d_{0j}$ then from (6.35) we have

$$\mathcal{R}_{31}[z(x); \bar{c}_{1j}] = -\bar{c}_{13} h \bar{\bar{\mathcal{A}}}^{(1)}_{30}[z'(x); \bar{\bar{a}}'^{(1)}_{0j}] . \tag{6.36}$$

We note that $\bar{\bar{\mathcal{A}}}^{(1)}_{30}[z'(x); \bar{\bar{a}}'^{(1)}_{0j}]$ is defined by replacing each of the arbitrary π functions in the definition (6.28) of $\bar{\bar{\mathcal{A}}}^{(1)}_{30}$ by their derivatives.

Thus we can now write

$$\bar{\mathcal{A}}^{(2)}_{31}[z(x); \bar{a}^{(2)}_{sj}] = \bar{\mathcal{A}}^{(1)}_{31}[z(x); \bar{a}^{(1)}_{sj}] - h\bar{c}_{13} \bar{\bar{\mathcal{A}}}^{(1)}_{30}[z'(x); \bar{\bar{a}}'^{(1)}_{0j}] . \tag{6.37}$$

The condition (6.22) with $i = 2$ will be satisfied if we choose

$$\bar{c}_{13} = \frac{\bar{\mathcal{A}}^{(1)}_{31}[\pi_2(x); \bar{a}^{(1)}_{sj}]}{h \bar{\bar{\mathcal{A}}}^{(1)}_{30}[\pi_2'(x); \bar{\bar{a}}'^{(1)}_{0j}]} \tag{6.38}$$

assuming the denominator is non-zero.

We have finally arrived at a recurrence formula for the zero-stable implicit non-polynomial operator $\bar{\mathcal{A}}^{(2)}_{31}$. Thus

$$\bar{\mathcal{A}}^{(2)}_{31}[z(x);\bar{a}^{(2)}_{sj}] = \bar{\mathcal{A}}^{(1)}_{31}[z(x);\bar{a}^{(1)}_{sj}] - \frac{\bar{\mathcal{A}}^{(1)}_{31}[\pi_2(x);\bar{a}^{(1)}_{sj}]}{h\bar{\bar{\mathcal{A}}}^{(1)}_{30}[\pi_2'(x);\bar{\bar{a}}'^{(1)}_{oj}]}\, h\bar{\bar{\mathcal{A}}}^{(1)}_{30}[z'(x);\bar{\bar{a}}'^{(1)}_{oj}] \qquad (6.39)$$

provided the denominator is non-zero.

To make use of the recurrence formula (6.39) we also need recurrence formulae for the operators $\bar{\mathcal{A}}^{(1)}_{31}$ and $\bar{\bar{\mathcal{A}}}^{(1)}_{30}$. By a similar process to that given previously it is straightforward to derive the following recurrence formula for $\bar{\mathcal{A}}^{(1)}_{31}$. Thus

$$\bar{\mathcal{A}}^{(1)}_{31}[z(x);\bar{a}^{(1)}_{sj}] = \bar{\mathcal{A}}^{(0)}_{31}[z(x);\bar{a}^{(0)}_{sj}] - \frac{\bar{\mathcal{A}}^{(0)}_{31}[\pi_1(x);\bar{a}^{(0)}_{sj}]}{h\bar{\bar{\mathcal{A}}}^{(0)}_{30}[\pi_1'(x);\bar{\bar{a}}'^{(0)}_{oj}]}\, h\bar{\bar{\mathcal{A}}}^{(0)}_{30}[z'(x);\bar{\bar{a}}'^{(0)}_{oj}] \qquad (6.40)$$

again assuming the denominator to be non-zero. The recurrence formula (6.40) can also be obtained directly from formula (2.15) in Lambert and Shaw (1966b). The operators $\bar{\mathcal{A}}^{(0)}_{31}$ and $\bar{\bar{\mathcal{A}}}^{(0)}_{30}$ correspond to the polynomial based operators $\mathcal{A}_{31}$ and $\bar{\mathcal{A}}_{30}$ respectively.

We now attempt to derive a recurrence formula for $\bar{\bar{\mathcal{A}}}^{(1)}_{30}$. The operator $\bar{\bar{\mathcal{A}}}^{(1)}_{30}$ is already normalized so that

$$\bar{\bar{a}}^{(1)}_{03} = \bar{\bar{a}}_{03}. \tag{6.41}$$

Setting

$$\bar{\bar{a}}^{(1)}_{0j} = \bar{\bar{a}}_{0j} + \bar{\bar{c}}_{0j} \qquad 0 \le j \le 2 \tag{6.42}$$

and substituting for $\bar{\bar{a}}^{(1)}_{0j}$ in (6.26) we obtain

$$\bar{\bar{\mathcal{A}}}^{(1)}_{30}[z(x); \bar{\bar{a}}^{(1)}_{0j}] = \bar{\bar{\mathcal{A}}}^{(0)}_{30}[z(x); \bar{\bar{a}}^{(0)}_{0j}] + \mathcal{A}_{20}[z(x); \bar{\bar{c}}_{0j}]\,. \tag{6.43}$$

Defining

$$\bar{\bar{d}}_{0j} = -\bar{\bar{c}}_{0j} / \bar{\bar{c}}_{02} \qquad 0 \le j \le 2\,,\ \bar{\bar{c}}_{02} \neq 0 \tag{6.44}$$

(6.43) becomes

$$\bar{\bar{\mathcal{A}}}^{(1)}_{30}[z(x); \bar{\bar{a}}^{(1)}_{0j}] = \bar{\bar{\mathcal{A}}}^{(0)}_{30}[z(x); \bar{\bar{a}}^{(0)}_{0j}] - \bar{\bar{c}}_{02}\, \mathcal{A}_{20}[z(x); \bar{\bar{d}}_{0j}] \tag{6.45}$$

Since $\bar{\bar{\mathcal{A}}}^{(0)}_{30}[x^{\tau}; \bar{\bar{a}}^{(0)}_{0j}] = 0$ for $\tau = 0,1$ (6.27) together with (6.45) implies

$$\mathcal{A}_{20}[x^{\tau}; \bar{\bar{d}}_{0j}] = 0 \qquad \tau = 0,1 \tag{6.46}$$

From (6.44) $\bar{\bar{d}}_{02} = -1$ and so the condition (6.46) is the definition of $\bar{\bar{\mathcal{A}}}^{(0)}_{20}[z(x); \bar{\bar{a}}^{(0)}_{0j}]$. Hence (6.45) becomes

$$\bar{\bar{\mathcal{A}}}^{(1)}_{30}[z(x);\bar{\bar{a}}^{(1)}_{oj}] = \bar{\bar{\mathcal{A}}}^{(0)}_{30}[z(x);\bar{\bar{a}}^{(0)}_{oj}] - \bar{\bar{c}}_{02}\bar{\bar{\mathcal{A}}}^{(0)}_{20}[z(x);\bar{\bar{a}}^{(0)}_{oj}] \tag{6.47}$$

The condition (6.28) is then satisfied if we take

$$\bar{\bar{c}}_{02} = \frac{\bar{\bar{\mathcal{A}}}^{(0)}_{30}[\pi_1(x);\bar{\bar{a}}^{(0)}_{oj}]}{\bar{\bar{\mathcal{A}}}^{(0)}_{20}[\pi_1(x);\bar{\bar{a}}^{(0)}_{oj}]} \tag{6.48}$$

provided the denominator is non-zero.

We have therefore the following recurrence formula for the optimum implicit non-polynomial operator $\bar{\bar{\mathcal{A}}}^{(1)}_{3,0}$. Thus

$$\bar{\bar{\mathcal{A}}}^{(1)}_{30}[z(x);\bar{\bar{a}}^{(1)}_{oj}] = \bar{\bar{\mathcal{A}}}^{(0)}_{30}[z(x);\bar{\bar{a}}^{(0)}_{oj}] - \frac{\bar{\bar{\mathcal{A}}}^{(0)}_{30}[\pi_1(x);\bar{\bar{a}}^{(0)}_{oj}]}{\bar{\bar{\mathcal{A}}}^{(0)}_{20}[\pi_1(x);\bar{\bar{a}}^{(0)}_{oj}]}\bar{\bar{\mathcal{A}}}^{(0)}_{20}[z(x);\bar{\bar{a}}^{(0)}_{oj}] \tag{6.49}$$

again assuming the denominator to be non-zero.

Since

$$\bar{\bar{\mathcal{A}}}^{(0)}_{k0}[z(x);\bar{\bar{a}}^{(0)}_{oj}] = -\Delta^k z(x) \tag{6.50}$$

then (6.40) can be simplified to

$$\bar{\mathcal{A}}^{(1)}_{31}[z(x);\bar{a}^{(1)}_{sj}] = \bar{\mathcal{A}}^{(0)}_{31}[z(x);\bar{a}^{(0)}_{sj}] - \frac{\bar{\mathcal{A}}^{(0)}_{31}[\pi_1(x);\bar{a}^{(0)}_{sj}]}{h\,\Delta^3\pi_1'(x)}\,h\Delta^3 z'(x) \tag{6.51}$$

while (6.49) becomes

$$\bar{\bar{A}}^{(1)}_{30}[z(x);\bar{\bar{a}}^{(1)}_{oj}] = -\left(\Delta^3 z(x) - \frac{\Delta^3 \pi_1(x)}{\Delta^2 \pi_1(x)}\Delta^2 z(x)\right) \qquad (6.52)$$

If we obtain from tables the coefficients of the polynomial based operator $\bar{A}^{(0)}_{31}$ then the three recurrence formulae (6.39),(6.51) and (6.52) enable us to determine the coefficients of the operator $\bar{A}^{(2)}_{31}$ in the following manner. We use (6.51)

$$\bar{A}^{(1)}_{31}[z(x);\bar{a}^{(1)}_{sj}] = \bar{A}^{(0)}_{31}[z(x);\bar{a}^{(0)}_{sj}] - \frac{\bar{A}^{(0)}_{31}[\pi_1(x);\bar{a}^{(0)}_{sj}]}{h\Delta^3\pi_1'(x)} h\Delta^3 z'(x)$$

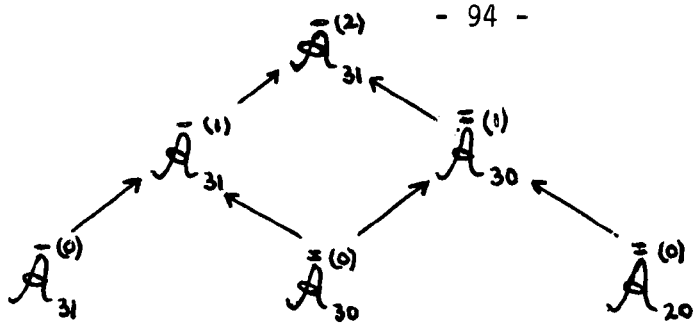
to find the coefficients of $\bar{A}^{(1)}_{31}$. Formula (6.52) is then employed in the form

$$\bar{\bar{A}}^{(1)}_{30}[z'(x);\bar{\bar{a}}'^{(1)}_{oj}] = -\left(\Delta^3 z'(x) - \frac{\Delta^3 \pi_1'(x)}{\Delta^2 \pi_1'(x)}\Delta^2 z'(x)\right)$$

to determine the coefficients $\bar{\bar{a}}'^{(1)}_{oj}$ of $\bar{\bar{A}}^{(1)}_{30}$. Finally we use (6.39)

$$\bar{A}^{(2)}_{31}[z(x);\bar{a}^{(2)}_{sj}] = \bar{A}^{(1)}_{31}[z(x);\bar{a}^{(1)}_{sj}] - \frac{\bar{A}^{(1)}_{31}[\pi_2(x);\bar{a}^{(1)}_{sj}]}{h\bar{\bar{A}}^{(1)}_{30}[\pi_2'(x);\bar{\bar{a}}'^{(0)}_{oj}]} h\bar{\bar{A}}^{(1)}_{30}[z'(x);\bar{\bar{a}}'^{(1)}_{oj}]$$

to calculate the coefficients $\bar{\bar{a}}^{(2)}_{sj}$ of $\bar{A}^{(2)}_{31}$.
Diagrammatically we represent this as



where the three elements at the base of the pyramid are the polynomially-based operators.

This example illustrates certain points that can be employed to simplify the notation relating to the general (k,ℓ) case. Thus we observe that the same superscripts are often attached to both the operator and the coefficients in the operator, and that when the operator is applied to the derivative of a function the coefficients also contain a dash superscript. We will therefore delete any reference to the coefficients when it is thought obvious what is intended. In passing it is noted that an alternative way of writing the operator $\bar{\bar{A}}^{(0)}_{20}[z(x)]$ is ${}_1\bar{\bar{A}}^{(0)}_{30}[z(x)]$ where the subscript before the operator ${}_1\bar{\bar{A}}^{(0)}_{30}$ indicates that the term $z(x+3h)$ is omitted. Similarly the operator ${}_2\bar{\bar{A}}^{(0)}_{30}$ would have the terms $z(x+3h)$ and $z(x+2h)$ missing. This use of the alternative form was not necessary in the particular example we considered but it will be required to indicate missing terms in the general (k,ℓ) case.

We now proceed to the general case and define the N-zero-stable implicit non-polynomial operator by

$$\bar{\mathcal{A}}^{(N)}_{k\ell}[z(x)] = \mathcal{P}_k[z(x);\bar{a}_{oj}] + \mathcal{R}_{k\ell}[z(x);\bar{a}^{(N)}_{sj}] \qquad \ell \geq 1, k \geq 1 \tag{6.53}$$

where the $\bar{a}^{(N)}_{sj}$, $1 \leq s \leq \ell$, $0 \leq j \leq k$ are determined as functions of the stability parameters $\bar{a}_{oj}$ by making

$$\bar{\mathcal{A}}^{(N)}_{k\ell}[x^r] = 0 \qquad r = 1,2,\ldots,P-N \quad 0 \leq N \leq P \tag{6.54}$$

and if $N > 0$

$$\bar{\mathcal{A}}^{(N)}_{k\ell}[\pi_i(x)] = 0 \qquad i = 1,\ldots,N \quad 1 \leq N \leq P \tag{6.55}$$

with P given by (6.13). The superscript N on the operator $\bar{\mathcal{A}}^{(N)}_{k\ell}$ denotes the number of arbitrary functions $\pi_1(x),\ldots,\pi_N(x)$ incorporated in the non-polynomial base.

Similarly we also have the N-optimum implicit non-polynomial operator defined by

$$\bar{\bar{\mathcal{A}}}^{(N)}_{k\ell}[z(x)] = \mathcal{P}_k[z(x);\bar{\bar{a}}_{oj}] + \mathcal{R}_{k\ell}[z(x);\bar{\bar{a}}^{(N)}_{sj}] \qquad \ell \geq 0, k \geq 0 \tag{6.56}$$

where the $\bar{\bar{a}}^{(N)}_{sj}$, $0 \leq s \leq \ell$, $0 \leq j \leq k$ are determined by requiring

$$\bar{\bar{\mathcal{A}}}^{(N)}_{k\ell}[x^r] = 0 \qquad r = 1,\ldots,Q-2-N \quad 0 \leq N \leq Q-2 \tag{6.57}$$

and if N > 0

$$\bar{\bar{A}}^{(N)}_{k\ell}[\pi_i(x)] = 0 \qquad i = 1,\ldots,N \quad 1 \le N \le Q-2 \tag{6.58}$$

with Q given by (6.17). For k = 1 the operators $\bar{\bar{A}}^{(N)}_{k\ell}$ and $\bar{A}^{(N)}_{k\ell}$ are identical while we define

$$\bar{\bar{A}}^{(0)}_{00}[z(x)] = -z(x) \tag{6.59}$$

Consider now the (N+1)-zero-stable implicit operator $\bar{A}^{(N+1)}_{k\ell}$ where the coefficients $\bar{\alpha}^{(N+1)}_{sj}$, $1 \le s \le \ell$, $0 \le j \le k$ are determined as functions of the stability parameters $\bar{\alpha}_{0j}$ by making

$$\bar{A}^{(N+1)}_{k\ell}[x^\tau] = 0 \qquad \tau = 1,\ldots,P-N-1 \tag{6.60}$$

and

$$\bar{A}^{(N+1)}_{k\ell}[\pi_i(x)] = 0 \qquad i = 1,\ldots,N+1. \tag{6.61}$$

If we then set

$$\bar{\alpha}^{(N+1)}_{sj} = \bar{\alpha}^{(N)}_{sj} + \bar{c}_{sj} \qquad 1 \le s \le \ell,\ 0 \le j \le k \tag{6.62}$$

and substitute for $\bar{\alpha}^{(N+1)}_{sj}$ in $\bar{A}^{(N+1)}_{k\ell}[z(x)]$ we obtain

$$\bar{A}^{(N+1)}_{k\ell}[z(x)] = \bar{A}^{(N)}_{k\ell}[z(x)] + \mathcal{R}_{k\ell}[z(x); \bar{c}_{sj}] \tag{6.63}$$

Since $\bar{A}^{(N)}_{k\ell}[x^\tau] = 0$ for $\tau = 1,\ldots,P-N-1$, (6.60) is equivalent to

$$\mathcal{R}_{k\ell}[x^\tau; \bar{c}_{sj}] = 0 \qquad \tau = 1,\ldots,P-N-1 \tag{6.64}$$

Substituting for $\bar{a}_{sj}^{(N+1)}$ in (6.61) from (6.62) and simplifying shows that (6.61) is equivalent to the conditions

$$\mathcal{R}_{k\ell}[\pi_i(x); \bar{c}_{sj}] = 0 \qquad i = 1, \ldots, N \tag{6.65}$$

and

$$\bar{\mathcal{A}}_{k\ell}^{(N+1)}\left[\pi_{N+1}(x); \bar{a}_{sj}^{(N)} + \bar{c}_{sj}\right] = 0 \tag{6.66}$$

Defining

$$\bar{d}_{s-1,j} = -\bar{c}_{sj} / \bar{c}_{1k} \qquad 1 \le s \le \ell,\ 0 \le j \le k,\ \bar{c}_{1k} \neq 0 \tag{6.67}$$

means that (6.64) implies

$$-\bar{c}_{1k} \sum_{s=0}^{\ell-1} \sum_{j=0}^{k} \bar{d}_{sj} h^s z^{(s)}(x+jh) = 0 \qquad z(x) = 1, x, \ldots, x^{p-N-2} \tag{6.68}$$

while (6.65) implies

$$-\bar{c}_{1k} \sum_{s=0}^{\ell-1} \sum_{j=0}^{k} \bar{d}_{sj} h^s z^{(s)}(x+jh) = 0 \qquad z(x) = \pi_1'(x), \ldots, \pi_N'(x) \tag{6.69}$$

Since $\bar{d}_{0k} = -1$ from (6.67) while (6.68) with $z(x) = 1$ implies $\bar{d}_{0,k-1} = 1 - \sum_{j=0}^{k-2} \bar{d}_{0j}$ then from (6.68) and (6.69) we have

$$\mathcal{R}_{K\ell}[z(x);\ \bar{c}_{sj}] = -\bar{c}_{1K}\, h\, \bar{\bar{\mathcal{A}}}^{(N)}_{K,\ell-1}[z'(x);\ \bar{\bar{a}}'^{(N)}_{sj}] \tag{6.70}$$

where the $\bar{\bar{a}}'^{(N)}_{sj}$ $0 \le s \le \ell-1$, $0 \le j \le k$ are determined by making

$$\bar{\bar{\mathcal{A}}}^{(N)}_{K,\ell-1}[z'(x)] = 0 \qquad z(x) = x, \ldots x^{p-N-1}, \pi_1(x), \ldots \pi_N(x) \tag{6.71}$$

Hence we can write

$$\bar{\mathcal{A}}^{(N+1)}_{K\ell}[z(x)] = \bar{\mathcal{A}}^{(N)}_{K\ell}[z(x)] - h\bar{c}_{1K}\bar{\bar{\mathcal{A}}}^{(N)}_{K,\ell-1}[z'(x)] \tag{6.72}$$

The condition (6.66) will then be satisfied if we choose

$$\bar{c}_{1K} = \frac{\bar{\mathcal{A}}^{(N)}_{K\ell}[\pi_{N+1}(x)]}{\bar{\bar{\mathcal{A}}}^{(N)}_{K,\ell-1}[\pi'_{N+1}(x)]} \qquad \ell \ge 1 \tag{6.73}$$

assuming that the denominator in (6.73) is non-zero.

Thus finally we arrive at the general recurrence formula for the N-zero-stable implicit non-polynomial operators.

$$\bar{\mathcal{A}}^{(N+1)}_{K\ell}[z(x)] = \bar{\mathcal{A}}^{(N)}_{K\ell}[z(x)] - \frac{\bar{\mathcal{A}}^{(N)}_{K\ell}[\pi_{N+1}(x)]}{h\,\bar{\bar{\mathcal{A}}}^{(N)}_{K,\ell-1}[\pi'_{N+1}(x)]}\, h\bar{\bar{\mathcal{A}}}^{(N)}_{K,\ell-1}[z'(x)] \tag{6.74}$$

$\ell \ge 1, k \ge 1$

again assuming the denominator in (6.74) is non-zero.

Setting N = 0 in (6.74) corresponds to the recurrence formula (2.15) given in Lambert and Shaw (1966b). We note also that when N = 0 the coefficients $\bar{\bar{a}}'^{(0)}_{s_j}$ in $\bar{\bar{\mathcal{A}}}^{(0)}_{k,\ell-1}[z'(x)]$ are the same as the $\bar{\bar{a}}^{(0)}_{s_j}$ occurring in $\bar{\bar{\mathcal{A}}}^{(0)}_{k,\ell-1}[z(x)]$ since there are then no arbitrary functions present.

We would now like to derive a general recurrence formula for the N-optimum implicit non-polynomial operator $\bar{\bar{\mathcal{A}}}^{(N)}_{k\ell}$. Before doing this it is necessary to introduce another optimum operator ${}_M\bar{\bar{\mathcal{A}}}^{(N)}_{k\ell}$ defined by

$$
\begin{aligned}
{}_M\bar{\bar{\mathcal{A}}}^{(N)}_{k\ell}[z(x)] = &\sum_{s=0}^{\ell-q+1}\sum_{j=0}^{k} {}_M\bar{\bar{a}}^{(N)}_{s_j} h^s z^{(s)}(x+jh) \\
&- \sum_{j=0}^{M-(q-1)(k+1)-1} {}_M\bar{\bar{a}}^{(N)}_{\ell-q+1,k-j} h^{\ell-q+1} z^{(\ell-q+1)}(x+(k-j)h)
\end{aligned}
\qquad (6.75)
$$

$$0 \le N \le Q-M-2, \quad \ell \ge 0, \quad k \ge 1$$

where

$$q = \text{entier}[(M-1)/(k+1)] + 1 \qquad 1 \le M \le Q-2 \qquad (6.76)$$

and the coefficients ${}_M\bar{\bar{a}}^{(N)}_{s_j}$ are determined by requiring

$${}_M\bar{\bar{\mathcal{A}}}^{(N)}_{k\ell}[x^r] = 0 \qquad r = 0,1,\ldots,Q-2-M-N \qquad (6.77)$$

and

$${}_M\bar{\bar{\mathcal{A}}}^{(N)}_{k\ell}[\pi_i(x)] = 0 \qquad i = 1,\ldots,N \qquad (6.78)$$

The right hand side of (6.75) is a linear combination of values of z and its first $(\ell-q+1)$ derivatives at $x,\ldots x + kh$ except that there are no $z^{(\ell-q+1)}$ terms at $x + kh,\ldots,x + (k-M+(q-1)(k+1)+1)h$. An example of this would be the operator $_2\bar{\bar{A}}^{(1)}_{31}$ given by

$$_2\bar{\bar{A}}^{(1)}_{31}[z(x)] = {}_2\bar{\bar{a}}^{(1)}_{00}z(x) + {}_2\bar{\bar{a}}^{(1)}_{01}z(x+h) + {}_2\bar{\bar{a}}^{(1)}_{02}z(x+2h) + {}_2\bar{\bar{a}}^{(1)}_{03}z(x+3h)$$
$$+ {}_2\bar{\bar{a}}^{(1)}_{10}hz'(x) + {}_2\bar{\bar{a}}^{(1)}_{11}hz'(x+h)$$

where the coefficients $_2\bar{\bar{a}}^{(1)}_{sj}$ are obtained from the conditions

$$_2\bar{\bar{A}}^{(1)}_{31}[x^r] = 0 \qquad r = 0,1,2,3$$

and

$$_2\bar{\bar{A}}^{(1)}_{31}[\pi_1(x)] = 0.$$

Returning to the (N+1)-optimum implicit non-polynomial operator $\bar{\bar{A}}^{(N+1)}_{K\ell}$, $k \geq 1$, which is determined by

$$\bar{\bar{A}}^{(N+1)}_{K\ell}[x^r] = 0 \qquad r = 0,\ldots,Q-3-N \tag{6.79}$$

and

$$\bar{\bar{A}}^{(N+1)}_{K\ell}[\pi_i(x)] = 0 \qquad i = 1,\ldots,N+1. \tag{6.80}$$

we will now assume that this operator has been normalized so that

$$\bar{\bar{a}}^{(N+1)}_{\ell K} = \bar{\bar{a}}^{(N)}_{\ell K}. \tag{6.81}$$

In addition we can set

$$\bar{\bar{a}}_{sj}^{(N+1)} = \bar{\bar{a}}_{sj}^{(N)} + \bar{\bar{c}}_{sj} \qquad 0 \le s \le \ell,\ 0 \le j \le k \tag{6.82}$$

where

$$\bar{\bar{c}}_{\ell k} = 0 . \tag{6.83}$$

Substituting for $\bar{\bar{a}}_{sj}^{(N+1)}$ in $\bar{\bar{\mathcal{A}}}_{K\ell}^{(N+1)}[z(x)]$ we obtain

$$\bar{\bar{\mathcal{A}}}_{K\ell}^{(N+1)}[z(x)] = \bar{\bar{\mathcal{A}}}_{K\ell}^{(N)}[z(x)] + \mathcal{A}_{K\ell}[z(x);\bar{\bar{c}}_{sj}] . \tag{6.84}$$

Defining

$$\bar{\bar{d}}_{sj} = -\bar{\bar{c}}_{sj} / \bar{\bar{c}}_{\ell,k-1} \qquad 0 \le s \le \ell,\ 0 \le j \le k\ ,\ \bar{\bar{c}}_{\ell,k-1} \neq 0 \tag{6.85}$$

(6.84) becomes

$$\bar{\bar{\mathcal{A}}}_{K\ell}^{(N+1)}[z(x)] = \bar{\bar{\mathcal{A}}}_{K\ell}^{(N)}[z(x)] - \bar{\bar{c}}_{\ell,k-1}\,\mathcal{A}_{K\ell}[z(x);\bar{\bar{d}}_{sj}] . \tag{6.86}$$

Since $\bar{\bar{\mathcal{A}}}_{K\ell}^{(N)}[x^{\tau}] = 0$ for $\tau = 0,\ldots,Q-2-N$, using (6.79) and (6.84) implies

$$\mathcal{A}_{K\ell}[x^{\tau};\bar{\bar{d}}_{sj}] = 0 \qquad \tau = 0,\ldots,Q-3-N \tag{6.87}$$

while $\bar{\bar{\mathcal{A}}}_{K\ell}^{(N)}[\pi_i(x)] = 0$ for $i = 1,\ldots,N$ together with (6.80) and (6.84) implies

$$\mathcal{A}_{K\ell}[\pi_i(x);\bar{\bar{d}}_{sj}] = 0 \qquad i = 1,\ldots,N. \tag{6.88}$$

Conditions (6.87) and (6.88), with $\bar{d}_{\ell k} = 0$ from (6.85), are precisely the definition of the operator $_1\bar{\bar{A}}^{(N)}_{K\ell}$.

Hence (6.86) becomes

$$\bar{\bar{A}}^{(N+1)}_{K\ell}[z(x)] = \bar{\bar{A}}^{(N)}_{K\ell}[z(x)] - \bar{\bar{c}}_{\ell,k-1}\,{}_1\bar{\bar{A}}^{(N)}_{K\ell}[z(x)]. \tag{6.89}$$

We can then satisfy (6.80) by taking

$$\bar{\bar{c}}_{\ell,k-1} = \frac{\bar{\bar{A}}^{(N)}_{K\ell}[\pi_{N+1}(x)]}{{}_1\bar{\bar{A}}^{(N)}_{K\ell}[\pi_{N+1}(x)]} \qquad \ell \geq 0,\ k \geq 1 \tag{6.90}$$

assuming the denominator in (6.90) is non-zero.

Thus the general recurrence formula for the optimum implicit non-polynomial operator is

$$\bar{\bar{A}}^{(N+1)}_{K\ell}[z(x)] = \bar{\bar{A}}^{(N)}_{K\ell}[z(x)] - \frac{\bar{\bar{A}}^{(N)}_{K\ell}[\pi_{N+1}(x)]}{{}_1\bar{\bar{A}}^{(N)}_{K\ell}[\pi_{N+1}(x)]}\,{}_1\bar{\bar{A}}^{(N)}_{K\ell}[z(x)] \tag{6.91}$$

$$\ell \geq 0, k \geq 1$$

again assuming the denominator is non-zero.

When N = 0 (6.91) becomes

$$\bar{\bar{A}}^{(1)}_{K\ell}[z(x)] = \bar{\bar{A}}^{(0)}_{K\ell}[z(x)] - \frac{\bar{\bar{A}}^{(0)}_{K\ell}[\pi_1(x)]}{{}_1\bar{\bar{A}}^{(0)}_{K\ell}[\pi_1(x)]}\,{}_1\bar{\bar{A}}^{(0)}_{K\ell}[z(x)] \tag{6.92}$$

$$\ell \geq 0, k \geq 1$$

The formula (2.16) in Lambert and Shaw (1966b) should correspond to (6.92), but does not because their

formula (2.16) is in error.

Formula (6.91) holds when $\ell = 0$ in which case it is helpful to use the relation

$$ {}_M\bar{\bar{A}}^{(N)}_{K0}[z(x)] = \bar{\bar{A}}^{(N)}_{K-M,0}[z(x)] . \tag{6.93} $$

To complete the recurrence relations for the implicit operators we now derive a recurrence formula for the operator ${}_M\bar{\bar{A}}^{(N+1)}_{K\ell}$. Consider then the operator ${}_M\bar{\bar{A}}^{(N+1)}_{K\ell}$ whose coefficients ${}_M\bar{\bar{a}}^{(N+1)}_{sj}$ are determined by making

$$ {}_M\bar{\bar{A}}^{(N+1)}_{K\ell}[x^\tau] = 0 \qquad \tau = 0,1,\ldots,Q-3-M-N \tag{6.94} $$

and

$$ {}_M\bar{\bar{A}}^{(N+1)}_{K\ell}[\pi_i(x)] = 0 \qquad i = 1,\ldots,N+1 \tag{6.95} $$

where we assume that this operator has been normalized so that

$$ {}_M\bar{\bar{a}}^{(N+1)}_{\ell-q+1,\,q(K+1)-M-1} = {}_M\bar{\bar{a}}^{(N)}_{\ell-q+1,\,q(K+1)-M-1} . \tag{6.96} $$

Setting

$$ {}_M\bar{\bar{a}}^{(N+1)}_{sj} = {}_M\bar{\bar{a}}^{(N)}_{sj} + {}_Mc_{sj} \qquad 0 \le s \le \ell,\ 0 \le j \le k \tag{6.97} $$

where

$$ {}_Mc_{sj} = 0 \qquad s > \ell-q+1 $$

and

$$ {}_M C_{\ell-q+1,j} = 0 \qquad j \geq q(k+1) - M - 1 \qquad (6.98) $$

we can then substitute for ${}_M\bar{\bar{a}}^{(N+1)}_{sj}$ in ${}_M\bar{\bar{\mathcal{A}}}^{(N+1)}_{k\ell}[z(x)]$

to obtain

$$ {}_M\bar{\bar{\mathcal{A}}}^{(N+1)}_{k\ell}[z(x)] = {}_M\bar{\bar{\mathcal{A}}}^{(N)}_{k\ell}[z(x)] + \mathcal{A}_{k\ell}[z(x); {}_M C_{sj}] . \qquad (6.99) $$

Defining

$$ {}_M d_{sj} = -{}_M C_{sj} / {}_M C_{\ell-q+1, q(k+1)-M-1} \qquad (6.100) $$

and using (6.100) to simplify (6.99) we get

$$ {}_M\bar{\bar{\mathcal{A}}}^{(N+1)}_{k\ell}[z(x)] = {}_M\bar{\bar{\mathcal{A}}}^{(N)}_{k\ell}[z(x)] - \gamma \mathcal{A}_{k\ell}[z(x); {}_M d_{sj}] \qquad (6.101) $$

where

$$ \gamma = {}_M C_{\ell-q+1, q(k+1)-M-2} \qquad \text{if } q(k+1) \geq M-2 $$

and

$$ \gamma = {}_M C_{\ell-q,k} \qquad \text{otherwise} \qquad (6.102) $$

Since ${}_M\bar{\bar{\mathcal{A}}}^{(N)}_{k\ell}[x^\tau] = 0$ for $\tau = 0,1,\ldots,Q-2-M-N$

using (6.94) and (6.101) implies

$$ \mathcal{A}_{k\ell}[x^\tau; {}_M d_{sj}] = 0 \qquad \tau = 0,1,\ldots,Q-3-M-N \qquad (6.103) $$

while ${}_M\bar{\bar{\mathcal{A}}}^{(N)}_{k\ell}[\pi_i(x)] = 0$ for $i = 1,\ldots,N$ together

with (6.92) and (6.98) implies

$$\mathcal{A}_{K\ell}[\pi_i(x); {}_M d_{s_j}] = 0 \qquad i = 1, \ldots, N \tag{6.104}$$

Conditions (6.103) and (6.104) together with ${}_M d_{s_j}$ $= 0$, $s > \ell - q + 1$ and ${}_M d_{\ell-q+1,j} = 0$, $j \geq q(k+1) - M$ correspond to the definition of the operator ${}_{M+1}\overset{=(N)}{\mathcal{A}}_{K\ell}$. Hence (6.101) becomes

$${}_M\overset{=(N+1)}{\mathcal{A}}_{K\ell}[z(x)] = {}_M\overset{=(N)}{\mathcal{A}}_{K\ell}[z(x)] - \gamma_{M+1}\overset{=(N)}{\mathcal{A}}_{K\ell}[z(x)] \tag{6.105}$$

while (6.95) is satisfied if we take

$$\gamma = \frac{{}_M\overset{=(N)}{\mathcal{A}}_{K\ell}[\pi_{N+1}(x)]}{{}_{M+1}\overset{=(N)}{\mathcal{A}}_{K\ell}[\pi_{N+1}(x)]} \tag{6.106}$$

provided the denominator is non-zero.

Hence the general recurrence relation for the operator ${}_M\overset{=(N+1)}{\mathcal{A}}_{K\ell}$ is

$${}_M\overset{=(N+1)}{\mathcal{A}}_{K\ell}[z(x)] = {}_M\overset{=(N)}{\mathcal{A}}_{K\ell}[z(x)] - \frac{{}_M\overset{=(N)}{\mathcal{A}}_{K\ell}[\pi_{N+1}(x)]}{{}_{M+1}\overset{=(N)}{\mathcal{A}}_{K\ell}[\pi_{N+1}(x)]}\ {}_{M+1}\overset{=(N)}{\mathcal{A}}_{K\ell}[z(x)] \tag{6.107}$$

$\ell \geq 0, k \geq 1$

again provided the denominator is non-zero.

We note that by setting $M = 0$ in (6.107) we obtain (6.91). Hence (6.91) is a particular case of (6.107). Thus the two formulae (6.74) and (6.107) enable us to construct any implicit non-polynomial

based operator solely from implicit polynomial based operators. To illustrate this we now give two examples.

Example 1. $\bar{\mathcal{A}}^{(3)}_{41}$

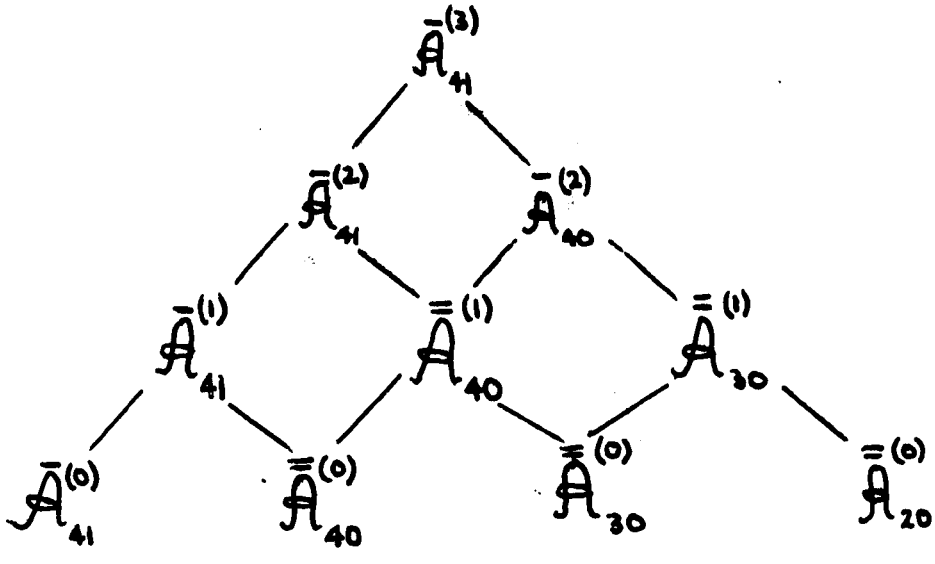
This operator is calculated in three stages starting from the polynomial based operators $\bar{\mathcal{A}}^{(0)}_{41}$, $\bar{\bar{\mathcal{A}}}^{(0)}_{40}$, $\bar{\bar{\mathcal{A}}}^{(0)}_{30}$ and $\bar{\bar{\mathcal{A}}}^{(0)}_{20}$ where $\bar{\bar{\mathcal{A}}}^{(0)}_{k0}$ is Δ^k.

Stage 1. Calculate $\bar{\mathcal{A}}^{(1)}_{41}[z(x)]$ from (6.74) and $\bar{\bar{\mathcal{A}}}^{(1)}_{40}[z'(x)]$, $\bar{\bar{\mathcal{A}}}^{(1)}_{30}[z'(x)]$ from (6.91).

Stage 2. Calculate $\bar{\mathcal{A}}^{(2)}_{41}[z(x)]$ from (6.74) and $\bar{\bar{\mathcal{A}}}^{(2)}_{40}[z'(x)]$ from (6.91).

Stage 3. Calculate $\bar{\mathcal{A}}^{(3)}_{41}[z(x)]$ from (6.74).

Diagrammatically we represent this by



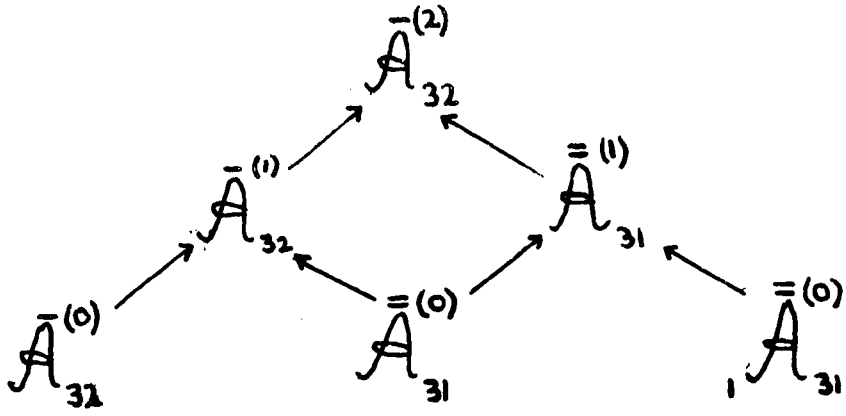
Example 2. $\bar{\mathcal{A}}^{(2)}_{32}$

This operator is calculated in two stages starting from the polynomial based operators $\bar{\mathcal{A}}^{(0)}_{32}$ $\bar{\bar{\mathcal{A}}}^{(0)}_{31}$ and ${}_1\bar{\bar{\mathcal{A}}}^{(0)}_{31}$

Stage 1. Calculate $\bar{\bar{A}}^{(1)}_{32}[z(x)]$ from (6.74) and $\bar{\bar{A}}^{(1)}_{31}[z'(x)]$ from (6.107).

Stage 2. Calculate $\bar{\bar{A}}^{(2)}_{32}[z(x)]$ from (6.74).

Diagrammatically we represent this by



We do not give an example for the construction of an N-optimum implicit formula since it is thought unlikely that such a formula would ever be used in practice other than in obtaining zero-stable formulae by use of the recurrence relations.

6.3 Explicit Non-Polynomial Operators

Our aim is to derive recurrence formulae for the explicit non-polynomial operators and we will proceed in a similar fashion to the implicit operators. We define the explicit operator $\mathcal{B}_{k\ell}$ applied to any function $z(x)$ by

$$\mathcal{B}_{k\ell}[z(x);b_{sj}] = \mathcal{P}_k[z(x);b_{oj}] + \mathcal{R}_{k-1,\ell}[z(x);b_{sj}] \qquad (6.108)$$

where the ℓk coefficients b_{sj}, $1 \leq s \leq \ell$, $0 \leq j \leq k-1$

can be determined as functions of the stability parameters b_{0j}, $0 \leq j \leq k-2$ by requiring

$$\mathcal{B}_{k\ell}[x^{r}; b_{sj}] = 0 \qquad r = 1, \ldots, R \tag{6.109}$$

where

$$R = \ell k . \tag{6.110}$$

The values b_{sj} which satisfy (6.109) are written as

$$\bar{b}_{sj}(b_{oj}, k, \ell) \qquad 1 \leq s \leq \ell, \; 0 \leq j \leq k \tag{6.111}$$

which for notational simplicity we denote by $\bar{b}_{sj}$.
The resultant operator $\mathcal{B}_{k\ell}$, which is zero-stable, is referred to as the stable (k,ℓ) explicit operator. That is

$$\bar{\mathcal{B}}_{k\ell}[z(x)] = \mathcal{B}_{k\ell}[z(x); \bar{b}_{sj}] . \tag{6.112}$$

If however we set

$$\mathcal{B}_{k\ell}[x^{r}; b_{sj}] = 0 \qquad r = 1, 2, \ldots, S \tag{6.113}$$

where

$$S = (\ell + 1)k \tag{6.114}$$

then we could have solved for b_{0j}, $0 \leq j \leq k-2$ in addition to the b_{sj}, $1 \leq s \leq \ell$, $0 \leq j \leq k-1$ and the

values of b_{0j} and b_{sj} which satisfy (6.110) are written as

$$\bar{\bar{b}}_{sj}(k,\ell) \qquad 0 \le s \le \ell, \; 0 \le j \le k \tag{6.115}$$

which we denote by $\bar{\bar{b}}_{sj}$ for convenience. The resultant operator $\mathcal{B}_{k\ell}$, which is generally not zero-stable, is then referred to as the optimum (k,ℓ) explicit operator:

$$\bar{\bar{\mathcal{B}}}_{k\ell}[z(x)] = \mathcal{B}_{k\ell}[z(x); \bar{\bar{b}}_{sj}] . \tag{6.116}$$

For the non-polynomial based operators we have the N-zero-stable explicit non-polynomial operator $\bar{\mathcal{B}}^{(N)}_{k\ell}$ defined by

$$\bar{\mathcal{B}}^{(N)}_{k\ell}[z(x)] = \mathcal{P}_k[z(x); \bar{b}_{0j}] + \mathcal{R}_{k-1,\ell}[z(x); \bar{b}^{(N)}_{sj}] \qquad \ell \ge 1, k \ge 1 \tag{6.117}$$

where the coefficients $\bar{b}^{(N)}_{sj}$, $1 \le s \le \ell$, $0 \le j \le k-1$ are determined as functions of the stability parameters $\bar{b}_{0j}$ by making

$$\bar{\mathcal{B}}^{(N)}_{k\ell}[x^r] = 0 \qquad r = 1, 2, \ldots, R-N \tag{6.118}$$

and

$$\bar{\mathcal{B}}^{(N)}_{k\ell}[\pi_i(x)] = 0 \qquad i = 1, \ldots, N \tag{6.119}$$

while the N-optimum explicit non-polynomial operator

is defined by

$$\bar{\bar{B}}^{(N)}_{K\ell}[z(x)] = P_k[z(x); \bar{b}_{oj}] + \mathcal{R}_{K-1,\ell}[z(x); \bar{\bar{b}}^{(N)}_{sj}] \qquad \ell \geq 1, k \geq 1 \quad (6.120)$$

where the $\bar{\bar{b}}^{(N)}_{sj}$, $0 \leq s \leq \ell$, $0 \leq j \leq k-1$ are determined from

$$\bar{\bar{B}}^{(N)}_{K\ell}[x^\tau] = 0 \qquad \tau = 1, \ldots, S-N-1 \quad 0 \leq N \leq S \quad (6.121)$$

and

$$\bar{\bar{B}}^{(N)}_{K\ell}[\pi_i(x)] = 0 \qquad i = 1, \ldots, N \quad 1 \leq N \leq S \quad (6.122)$$

For $k = 1$ the operators $\bar{B}^{(N)}_{K\ell}$ and $\bar{\bar{B}}^{(N)}_{K\ell}$ are identical.

Since the derivation of the recurrence formulae for the explicit non-polynomial operators is very similar to that of the implicit we relegate the proofs to Appendix 2 and restrict ourselves here to quoting the following results

$$\bar{B}^{(N+1)}_{K\ell}[z(x)] = \bar{B}^{(N)}_{K\ell}[z(x)] - \frac{\bar{B}^{(N)}_{K\ell}[\pi_{N+1}(x)]}{h\bar{\bar{A}}^{(N)}_{K-1,\ell-1}[\pi'_{N+1}(x)]}\, h\bar{\bar{A}}^{(N)}_{K-1,\ell-1}[z'(x)] \quad (6.123)$$

$$\ell \geq 1, k \geq 1, N+1 \leq R$$

and

$$\bar{\bar{B}}^{(N+1)}_{K\ell}[z(x)] = \bar{\bar{B}}^{(N)}_{K\ell}[z(x)] - \frac{\bar{\bar{B}}^{(N)}_{K\ell}[\pi_{N+1}(x)]}{\bar{\bar{A}}^{(N)}_{K-1,\ell}[\pi_{N+1}(x)]} \bar{\bar{A}}^{(N)}_{K-1,\ell}[z(x)] \qquad (6.124)$$

$$\ell \geq 1,\ k \geq 1,\ N+1 \leq S$$

where we assume that the denominators in (6.123) and (6.124) are both non zero

When using the recurrence formula (6.123) with k = 1 we find that the operator $\bar{\bar{A}}^{(N)}_{0,\ell-1}$ occurs. The general recurrence formula (6.91) for the optimum implicit non-polynomial operator is however only valid for $k \geq 1$. Hence we need an additional recurrence formula to cover the case when k = 0. The proof of this recurrence formula is relegated to Appendix 2 and we only quote the result:

$$\bar{\bar{A}}^{(N+1)}_{0\ell}[z(x)] = \bar{\bar{A}}^{(N)}_{0\ell}[z(x)] - \frac{\bar{\bar{A}}^{(N)}_{0\ell}[\pi_{N+1}(x)]}{\bar{\bar{A}}^{(N)}_{0,\ell-1}[\pi_{N+1}(x)]} \bar{\bar{A}}^{(N)}_{0,\ell-1}[z(x)] \qquad (6.125)$$

$$\ell \geq 1.$$

where again the denominator is assumed non-zero.

The two formulae (6.123) and (6.124) together with (6.91), (6.107) and (6.125) enable us to construct any explicit non-polynomial operator directly from explicit and implicit polynomial based operators. To illustrate this construction we give two examples.

Example 1. $\bar{\bar{B}}^{(2)}_{31}$

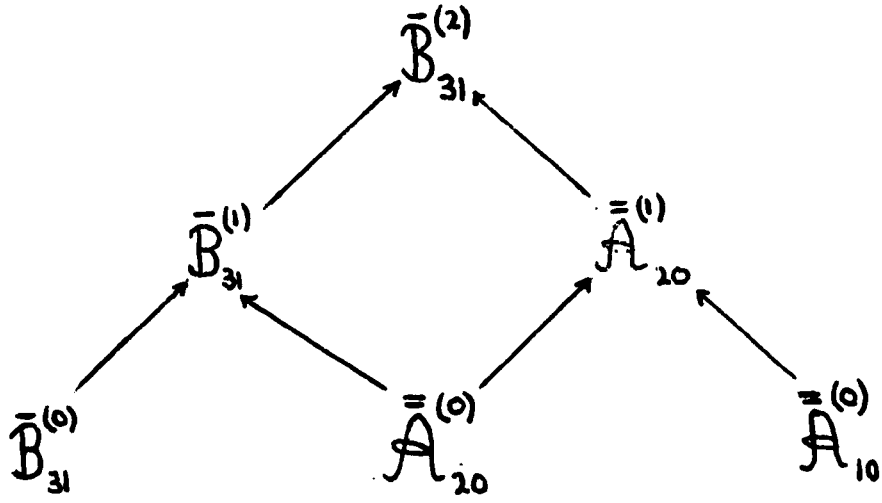
This operator is calculated in two stages starting

from the polynomial based operators , , .

Stage 1. Calculate $\bar{\mathcal{B}}_{31}^{(1)}[z(x)]$ from (6.123) and $\bar{\bar{\mathcal{A}}}_{20}^{(1)}[z'(x)]$ from (6.91)

Stage 2. Calculate $\bar{\mathcal{B}}_{31}^{(2)}[z(x)]$ from (6.123).

Diagrammatically we represent this by



Example 2. $\bar{\mathcal{B}}_{32}^{(3)}$

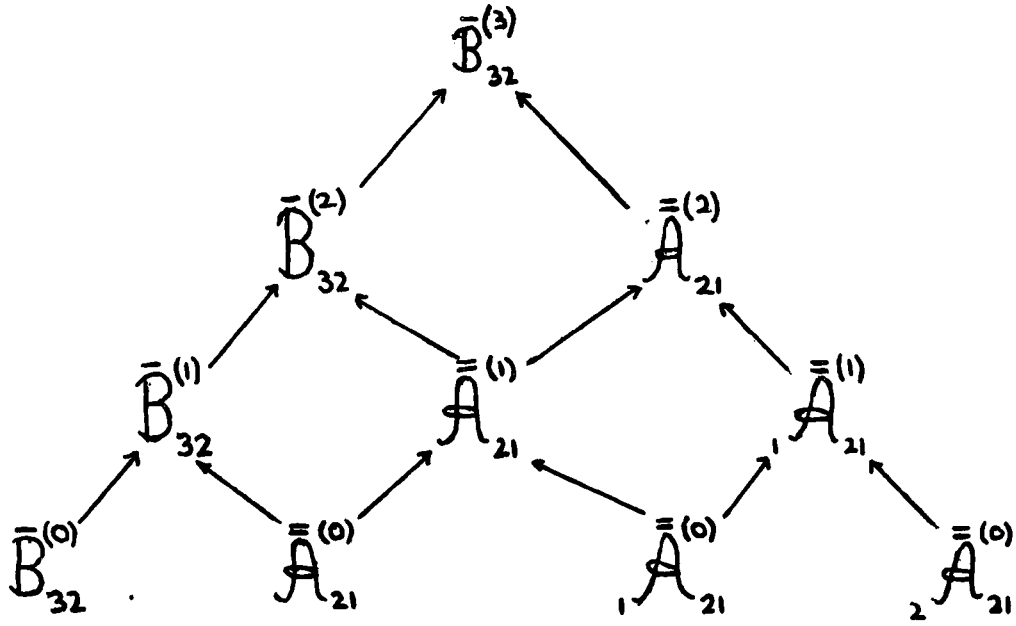
This operator is calculated in three stages starting from the polynomial based operators $\bar{\mathcal{B}}_{32}^{(0)}$, $\bar{\bar{\mathcal{A}}}_{21}^{(0)}$, ${}_2\bar{\bar{\mathcal{A}}}_{21}^{(0)}$.

Stage 1. Calculate $\bar{\mathcal{B}}_{32}^{(1)}[z(x)]$ from (6.123) and $\bar{\bar{\mathcal{A}}}_{21}^{(0)}[z'(x)]$, ${}_1\bar{\bar{\mathcal{A}}}_{21}^{(0)}[z'(x)]$ from (6.107).

Stage 2. Calculate $\bar{\mathcal{B}}_{32}^{(2)}[z(x)]$ from (6.123) and $\bar{\bar{\mathcal{A}}}_{21}^{(1)}[z'(x)]$ from (6.91).

Stage 3. Calculate $\bar{\mathcal{B}}_{32}^{(3)}[z(x)]$ from (6.123).

Diagrammatically we represent this by



6.4 Stability Characteristics of the Non-Polynomial Based Formulae

From their definitions the orders of the various non-polynomial based operators are as follows

Operator	Order		
$\bar{A}^{(N)}_{k\ell}[z(x)]$	P-N	$\ell \geq 1,\ k \geq 1$	(6.126.1)
$\bar{\bar{A}}^{(N)}_{k\ell}[z(x)]$	Q-N-2	$\ell \geq 0,\ k \geq 1$	(6.126.2)
${}_M\bar{\bar{A}}^{(N)}_{k\ell}[z(x)]$	Q-N-M-2	$\ell \geq 0,\ k \geq 1$	(6.126.3)
$\bar{B}^{(N)}_{k\ell}[z(x)]$	R-N	$\ell \geq 1,\ k \geq 1$	(6.126.4)
$\bar{\bar{B}}^{(N)}_{k\ell}[z(x)]$	S-N-1	$\ell \geq 1,\ k \geq 1$	(6.126.5)
$\bar{\bar{A}}^{(N)}_{0\ell}[z(x)]$	ℓ-N-1	$\ell \geq 0,\ k \geq 0$	(6.126.6)

} (6.126)

where the order of $\bar{\bar{A}}^{(0)}_{00}$ is defined to be -1.

Our aim is to determine the stability characteristics of the non-polynomial based operators. Initially we will only consider the implicit zero-stable non-polynomial based

operator $\bar{\mathcal{A}}^{(N)}_{K\ell}$. From (6.74) we have the recurrence relation

$$\bar{\mathcal{A}}^{(N+1)}_{K\ell}[z(x)] = \bar{\mathcal{A}}^{(N)}_{K\ell}[z(x)] - \frac{\bar{\mathcal{A}}^{(N)}_{K\ell}[\pi_{N+1}(x)]}{h\bar{\bar{\mathcal{A}}}^{(N)}_{K,\ell-1}[\pi'_{N+1}(x)]}\, h\bar{\bar{\mathcal{A}}}^{(N)}_{K,\ell-1}[z'(x)]$$

Using the order relations given in (6.126) we have

$$\frac{\bar{\mathcal{A}}^{(N)}_{K\ell}[\pi_{N+1}(x)]}{h\bar{\bar{\mathcal{A}}}^{(N)}_{K,\ell-1}[\pi'_{N+1}(x)]} = O(h)$$

This implies that the (N+1)-zero-stable implicit operator is an $O(h^2)$ perturbation of the N-zero-stable implicit operator. It follows therefore that for any N > 0 the N-zero-stable implicit operator is an $O(h^2)$ perturbation of the corresponding polynomial based operator.

Similar arguments can be used to show that, relative to the corresponding polynomial based operators, the N-optimum implicit operator is an O(h) perturbation, the N-zero-stable explicit operator is an $O(h^2)$ perturbation and the N-optimum explicit operator is an O(h) perturbation.

Thus for the zero-stable operators the stability characteristics of the associated non-polynomial based formulae are the same as those of the corresponding polynomial based formulae. No conclusions can be drawn regarding the stability characteristics of the optimum operators since the non-polynomial based operators are only an O(h) perturbation of the corresponding polynomial based operator.

6.5 Local Error Bound for the Non-Polynomial Based Formulae

The formulae (3.1), (3.2), (3.3) and (3.4) in the third chapter give the local error bound for a first-derivative operator. For the general operator $\mathcal{L}_{k\ell}$ we have

$$\mathcal{L}_{k\ell}[z(x);a_{sj}] = h^{p+1}\int_0^k G(\sigma)z^{(p+1)}(x+h\sigma)\,d\sigma \tag{6.127}$$

where

$$G(\sigma) = \sum_{s=0}^{\ell}\sum_{j=0}^{k} a_{sj}\frac{(j-\sigma)_+^{p-s}}{(p-s)!} \tag{6.128}$$

Since the function z(x) is assumed to be sufficiently differentiable and continuous then if the kernel $G(\sigma)$ is of constant sign throughout $0 \leq \sigma \leq k$ we can apply the first mean-value theorem for integrals to obtain

$$\mathcal{L}_{k\ell}[z(x)] = h^{p+1}z^{(p+1)}(\xi)\int_0^k G(\sigma)\,d\sigma \qquad \xi\in[x,x+kh]. \tag{6.129}$$

For $r \geq \ell + 1$ we have

$$C_r = \sum_{s=0}^{\ell}\sum_{j=1}^{k} a_{sj}\frac{j^{r-s}}{(r-s)!} \tag{6.130}$$

Setting r = p + 1 in (6.130) and using this in (6.128) and (6.129) gives

$$\mathcal{L}_{k\ell}[z(x)] = C_{p+1}h^{p+1}z^{(p+1)}(\xi) \tag{6.131}$$

If we denote the kernels of the non-polynomial based operators $\bar{\mathcal{A}}^{(N)}_{k\ell}$, $\bar{\bar{\mathcal{A}}}^{(N)}_{k\ell}$, ${}_M\bar{\bar{\mathcal{A}}}^{(N)}_{k\ell}$, $\bar{\mathcal{B}}^{(N)}_{k\ell}$, $\bar{\bar{\mathcal{B}}}^{(N)}_{k\ell}$, $\bar{\bar{\mathcal{A}}}^{(N)}_{0\ell}$ by $\bar{G}^{(N)}_{k\ell\mathcal{A}}(\sigma)$, $\bar{\bar{G}}^{(N)}_{k\ell\mathcal{A}}(\sigma)$, ${}_M\bar{\bar{G}}^{(N)}_{k\ell\mathcal{A}}(\sigma)$, $\bar{G}^{(N)}_{k\ell\mathcal{B}}(\sigma)$, $\bar{\bar{G}}^{(N)}_{k\ell\mathcal{B}}(\sigma)$, $\bar{\bar{G}}^{(N)}_{0\ell\mathcal{A}}(\sigma)$ respectively then for the zero-stable implicit operator $\bar{\mathcal{A}}^{(N+1)}_{k\ell}$ we can write

$$\bar{\mathcal{A}}^{(N+1)}_{K\ell}[z(x)] = h^{P-N+1}\Big[\int_0^K \bar{G}^{(N)}_{K\ell A}(\sigma) z^{(P-N-1)}(x+h\sigma)$$

$$- \frac{\int_0^K \bar{G}^{(N)}_{K\ell A}(\sigma)\pi^{(P-N-1)}(x+h\sigma)\,d\sigma}{\int_0^K \bar{\bar{G}}^{(N)}_{K,\ell-1,A}(\sigma)\pi^{(Q-N)}(x+h\sigma)\,d\sigma}\int_0^K \bar{\bar{G}}^{(N)}_{K,\ell-1,A}(\sigma) z^{(Q-N)}(x+h\sigma)\,d\sigma\Big] \tag{6.132}$$

When $\bar{G}^{(N)}_{K\ell A}(\sigma)$ and $\bar{\bar{G}}^{(N)}_{K,\ell-1,A}(\sigma)$, $0 \le \sigma \le k$ each have the same sign (6.132) reduces to

$$\bar{\mathcal{A}}^{(N+1)}_{K\ell}[z(x)] = C_{P-N+1}h^{P-N+1}\left[z^{(P-N+1)}(\xi_1) - \frac{\pi^{(P-N+1)}_{N+1}(\xi_2)}{\pi^{(Q-N)}_{N+1}(\eta_1)} z^{(Q-N)}(\eta_2)\right] \tag{6.133}$$

$k \ge 1$

where $\xi_j \in [x, x+kh]$, $\eta_j \in [x, x+kh]$, $j = 1,2$.

Similar bounds to (6.132) and (6.133) exist for the optimum implicit, zero-stable explicit and optimum explicit non-polynomial operators.

6.6 Work Involved when using the Recurrence Formulae to Derive the Non-Polynomial Based Operators

6.6.1 Arithmetic Operations

Using the recurrence formulae, given in the previous sections, to derive a non-polynomial based operator is not the only way of carrying out the derivation. One could make the operator exact for any given set of functions by substituting these functions in the operator, equating to zero, and solving the resultant set of linear simultaneous equations directly for the coefficients. Obviously we would wish to use the most efficient method to derive the operator and we will therefore try to estimate the work involved

in deriving an operator both by using recurrence formulae and by the direct solution of the linear simultaneous equations. In this section consideration is given to the number of arithmetic operations involved and discussions regarding the amount of storage necessary are left to a later section. Since it is unlikely that anyone would use the optimum operators in practice we omit these completely from the rest of the chapter.

Initially we consider the N-zero-stable implicit operator $\bar{\mathcal{A}}^{(N)}_{k\ell}$ and its derivation by recurrence formulae. There are essentially N stages in the derivation of $\bar{\mathcal{A}}^{(N)}_{k\ell}$. In the J-th stage we have to calculate

$$\bar{\mathcal{A}}^{(J-1)}_{k\ell}[\pi_J(x)], \bar{\bar{\mathcal{A}}}^{(J-1)}_{K,\ell-1}[\pi'_J(x)], {}_1\bar{\bar{\mathcal{A}}}^{(J-1)}_{K,\ell-1}[\pi'_J(x)], \ldots\ldots, {}_{N-J}\bar{\bar{\mathcal{A}}}^{(J-1)}_{K,\ell-1}[\pi'_J(x)] \quad (6.134)$$

in order to obtain

$$\bar{\mathcal{A}}^{(J)}_{k\ell}[z(x)], \bar{\bar{\mathcal{A}}}^{(J)}_{K,\ell-1}[z'(x)], {}_1\bar{\bar{\mathcal{A}}}^{(J)}_{K,\ell-1}[z'(x)], \ldots\ldots, {}_{N-J-1}\bar{\bar{\mathcal{A}}}^{(J)}_{K,\ell-1}[z'(x)] \quad (6.135)$$

The work involved in this is

$$(\ell+1)(k+1) + \ell(k+1) + \{\ell(k+1)-1\} + \ldots\ldots\ldots + \{\ell(k+1)-N+J\}$$
$$+ \ell(k+1) + \{\ell(k+1)-1\} + \ldots\ldots\ldots + \{\ell(k+1)-N+J\}$$

operations where an operation is regarded as equivalent to a multiplication and an addition. There are also an additional N-J+1 divisions which we overestimate as being equivalent to N-J+1 operations. Hence at the J-th stage the

total number of operations is

$$(\ell+1)(k+1) + (N-J+1)(2\ell(k+1) - N+J+1) \tag{6.136}$$

We must now sum (6.136) over J running from one to N.

Hence we have

$$\sum_{J=1}^{N} (\ell+1)(k+1) + \sum_{J=1}^{N} (N-J+1)(2\ell(k+1) - N+J+1) \tag{6.137}$$

Evaluating (6.137) gives a total of

$$N(k+1)(\ell N + 2\ell + 1) - \frac{N}{6}(N+1)(2N-5) \tag{6.138}$$

operations for the calculation of the N-zero-stable implicit operator $\bar{A}^{(N)}_{k\ell}$ by means of recurrence formulae.

By a direct method using Gaussian elimination we would need

$$\frac{1}{3}[\ell(k+1)]^3 + [\ell(k+1)]^2 - \frac{1}{3}[\ell(k+1)] \tag{6.139}$$

operations in carrying out the elimination and back substitution. Hence we need to compare (6.138) and (6.139) for any given set of k, ℓ and N to determine the more efficient method. The following table, Table 6.1, gives the values of (6.138) and (6.139) for ℓ = 1, 2; N = 1,2,3,4 and k = 1,2,...,5 subject to the condition that $\ell(k+1) \geq N$.

Table 6.1

ℓ	N	k	Recurrence (6.138)	Direct (6.139)
1	1	1	9	6
1	1	2	13	17
1	1	3	17	36
1	1	4	21	65
1	1	5	25	106
1	2	1	21	6
1	2	2	31	17
1	2	3	41	36
1	2	4	51	65
1	2	5	61	106
1	3	2	52	17
1	3	3	70	36
1	3	4	88	65
1	3	5	106	106
1	4	3	102	36
1	4	4	130	65
1	4	5	158	106
2	1	1	15	36
2	1	2	22	106
2	1	3	29	232
2	1	4	36	430
2	1	5	43	716
2	2	1	37	36
2	2	2	55	106
2	2	3	73	232
2	2	4	91	430
2	2	5	109	716
2	3	1	64	36
2	3	2	97	106
2	3	3	130	232
2	3	4	163	430
2	3	5	196	716
2	4	1	94	36
2	4	2	146	106
2	4	3	198	232
2	4	4	250	430
2	4	5	302	716

We summarise these results in Table 6.2 where we quote the minimum value of k, for a given N and ℓ, at which the recurrence derivation becomes more efficient.

Table 6.2

ℓ	N	k
1	1	2
	2	4
	3	5
	4	7
2	1	1
	2	2
	3	2
	4	3

Hence when ℓ is one the minimum value of k increases faster than linearly with N so that the recurrence derivation would probably only be used when N equals one and k is at least two. It is possible that when N equals two or three we might employ the recurrence derivation provided the value of k was at least four or five respectively. If ℓ is two however, the recurrence derivation becomes more attractive to use since the minimum value of k increases less than linearly with respect to N. Larger values of ℓ would make use of the recurrence derivation more efficient still in comparison to the direct method.

We now consider the N-zero-stable explicit operator $\bar{B}^{(N)}_{k\ell}$ which also has N stages in its derivation by recurrence formulae. In the J-th stage we need to calculate

$$\bar{\bar{B}}_{k\ell}^{(J-1)}[\pi_J(x)], \bar{\bar{A}}_{k-1,\ell-1}^{(J-1)}[\pi_J'(x)], \bar{\bar{A}}_{k-1,\ell-1}^{(J-1)}[\pi_J'(x)], \ldots, \bar{\bar{A}}_{k-1,\ell-1}^{(J-1)}[\pi_J'(x)] \quad (6.140)$$

in order to obtain

$$\bar{\bar{B}}_{k\ell}^{(J)}[z(x)], \bar{\bar{A}}_{k-1,\ell-1}^{(J)}[z'(x)], \bar{\bar{A}}_{k-1,\ell-1}^{(J)}[z'(x)], \ldots, \bar{\bar{A}}_{k-1,\ell-1}^{(J)}[z'(x)] \quad (6.141)$$

The number of operations involved in this stage is

$$k(\ell+1)+1+\ell k+(\ell k-1)+\ldots+[\ell k-(N-J)] + \ell k+(\ell k-1)+\ldots+\{\ell k-(N-J)\}+N-J+1 \quad (6.142)$$

Summing (6.142) for J running from 1 to N yields

$$\sum_{J=1}^{N}(k(\ell+1)+1)+\sum_{J=1}^{N}(N-J+1)(2\ell k-N+J+1) \quad (6.143)$$

Evaluating (6.143) gives a total of

$$Nk(\ell N+2\ell+1)-\frac{N}{6}(2N^2-3N-11) \quad (6.144)$$

operations for the calculation of the N-zero-stable explicit operator $\bar{B}_{k\ell}^{(N)}$ by means of recurrence formula. By the direct method we would use

$$\frac{1}{3}(\ell k)^3+(\ell k)^2-\frac{1}{3}(\ell k) \quad (6.145)$$

Table 6.3 compares the number of operations as given in (6.144) and (6.145) for ℓ = 1,2;N=1,2,3,4,k=1,2,3,4,5 subject to the condition $\ell k \geq N$.

Table 6.3

ℓ	N	k	Recurrence (6.144)	Direct (6.145)
1	1	2	10	6
	1	3	14	17
	1	4	18	36
	1	5	22	65
	2	2	23	6
	2	3	33	17
	2	4	43	36
	2	5	53	65
	3	3	55	17
	3	4	73	36
	3	5	91	65
	4	4	106	36
	4	5	134	65
2	1	2	16	36
	1	3	23	106
	1	4	30	232
	1	5	37	430
	2	2	39	36
	2	3	57	106
	2	4	75	232
	2	5	93	430
	3	2	67	36
	3	3	100	106
	3	4	133	232
	3	5	166	430
	4	2	98	36
	4	3	150	106
	4	4	202	232
	4	5	254	430

These results are summarised in the following table which quotes the minimum value of k for a given N and ℓ at which the recurrence derivation becomes more efficient.

Table 6.4

ℓ	N	k
1	1	3
	2	4
	3	7
	4	8
2	1	2
	2	3
	3	3
	4	4

The results for the N-zero-stable explicit operator follow the same pattern as those of N-zero-stable implicit operator. From Table 6.4 we can see that when ℓ is one the minimum value of k increases faster than linearly with N so that the recurrence derivation would probably only be used when N is either one or two in which case k must be at least three or four respectively. If ℓ is two however then the minimum value of k increases less than linearly with respect to N and the recurrence derivation becomes more advantageous while larger values for ℓ make use of the recurrence derivation still more efficient in comparison with the direct method.

A tentative conclusion for the values of N, k and ℓ that have been considered and based solely on the number of arithmetic operations involved in the derivation of both implicit and explicit zero-stable operators is that the method employing recurrence relations should only be used when either ℓ is one and N is sufficiently small with respect to k as given in Tables 6.2, 6.4, or if ℓ is two then k must be greater than or equal to N.

6.6.2 Storage Space

With the vast development in computer technology over the last twenty years and the consequent increase in the available storage space in most computers one does not usually have to worry unduly about storage space required since this is no longer the restriction it once used to be. However for completeness we will compare the amount of storage space needed for both the recurrence derivation and the direct method and show that the same conclusions that were drawn from a comparison of the arithmetic operations can also be drawn from a consideration of the storage space required.

Consider first the N-zero-stable implicit operator $\bar{A}_{ke}^{(N)}$. Since the recurrence technique can be used so that each stage overwrites the previous stage, from (6.135) it is easy to see that we need

$$(\ell+1)(k+1) + \ell(k+1) + \ldots\ldots + [\ell(k+1) - (N-2)]$$
$$+(\ell+1)(k+1) + \ell(k+1) + \ldots\ldots + [\ell(k+1) - (N-2)] + [\ell(k+1) - (N-1)] \qquad (6.146)$$

storage locations. This is not the total number of locations needed but it is judged to be the most relevant quantity to be compared with the storage needed for the solution of the linear simultaneous equations in the direct method. Both methods do require a small amount of additional temporary storage space but this is small in comparison and is approximately equal in both methods and hence we ignore it. The direct solution will require

$$[\ell(k+1)][\ell(k+1)+1] \qquad (6.147)$$

storage locations. Simplifying (6.146) gives

$$(k+1)(2\ell N+\ell+2)-(N-1)^2 \qquad (6.148)$$

storage locations for the recurrence technique. Table 6.5 then gives a comparison of (6.148) and (6.147) for $\ell = 1,2$; N=1,2,3,4, k=1,2,3,4,5 subject to the condition that $\ell(k+1) \geq N$.

Table 6.5

ℓ	N	k	Recurrence (6.148)	Direct (6.147)
1	1	1	10	6
	1	2	15	12
	1	3	20	20
	1	4	25	30
	1	5	30	42
	2	1	13	6
	2	2	20	12
	2	3	27	20
	2	4	34	30
	2	5	41	42
	3	2	23	12
	3	3	32	20
	3	4	41	30
	3	5	50	42
	4	3	35	20
	4	4	46	30
	4	5	57	42
2	1	1	16	20
	1	2	24	42
	1	3	32	72
	1	4	40	110
	1	5	48	156
	2	1	23	20
	2	2	35	42
	2	3	47	72
	2	4	59	110
	2	5	71	156
	3	2	44	42
	3	3	60	72
	3	4	76	110
	3	5	92	156
	4	3	71	72
	4	4	91	110
	4	5	111	156

We summarize these results in Table 6.6 where we quote

the minimum value of k for a given N and ℓ at which the recurrence derivation becomes more efficient.

Table 6.6

ℓ	N	k
1	1	3
	2	5
	3	7
	4	8
2	1	1
	2	2
	3	3
	4	3

Using these results for the N-zero-stable implicit operator we can draw similar conclusions from the storage space requirements to those drawn from the number of arithmetic operations. Hence when ℓ is one the value of N has to be sufficiently small with respect to k to make the recurrence derivation more efficient while when ℓ is 2 the recurrence derivation becomes more attractive since k need only be as large as N. Higher values of ℓ reflect better on the recurrence derivation.

For the N-zero-stable explicit operator $\bar{B}_{k\ell}^{(N)}$ we restrict ourselves to quoting the storage requirements for both methods of derivation, recurrence and direct, since it is straightforward but tedious to show that the conclusions one obtains are the same as in the previous section. The operator $\bar{B}_{k\ell}^{(N)}$ can be seen from (6.141) to require

$$\{(\ell+1)k+1\} + \ell k + \ldots\ldots + [\ell k-(N-2)] + \{(\ell+1)k+1\} + \ell k + \ldots\ldots + \{\ell k-(N-2)\} + [\ell k-(N-1)] \qquad (6.149)$$

storage locations for the recurrence derivation. This can be simplified to

$$k(2\ell N+\ell+2)-(N^2-2N-1) \tag{6.150}$$

while the direct solution requires

$$\ell k(\ell k+1) \tag{6.151}$$

storage locations.

Although the results from this section back up the evidence obtained from the comparison of the arithmetic operations it is important to remember that the additional storage space required for the recurrence derivation is relatively small and hence our judgement of the method to use should be based completely on the number of arithmetic operations involved. This is particularly important for methods based on arbitrary functions which necessitate recalculating the operator at each step, x_{n+k}, of the solution.

Finally it should be also noted that the recurrence derivation of each of the operators can be suitably modified to cope with second-order ordinary differential equations of the form

$$y'' = f(x,y) \tag{6.152}$$

with appropriate initial conditions. In fact the recurrence derivation can be applied to cope with any order differential equation of the form

$$y^{(r)} = f(x,y) \tag{6.153}$$

where $r \geq 1$.

6.7 Numerical Testing of Methods Derived by Recurrence Formulae

As a numerical test of methods derived by recurrence formulae we will attempt to find a solution to the following two problems given in Bettis and Stiefel (1969):

A) Harmonic Oscillator

$$y'' = -y \qquad \text{Range } [0,40\pi]$$

where

$$y = {}_1y + i\ {}_2y$$

while at $x = 0$

$${}_2y = {}_1y' = 0$$

$${}_1y = {}_2y' = 1$$

This problem has the solution

$${}_1y = \cos x$$

$${}_2y = \sin x$$

B) Perturbed Harmonic Oscillator

$$y'' + y = 0.001 \exp(ix) \qquad \text{Range } [0,40\pi]$$

where

$$y = {}_1y + i\ {}_2y$$

while at $x = 0$

$${}_2y = {}_1y' = 0$$

$${}_1y = {}_2y' = 1$$

This problem has the solution

$$_1y = \cos x + 0.005 \times \sin x$$

$$_2y = \sin x - 0.005 \times \cos x$$

Both of these problems are second-order differential equations. We need therefore a multistep method which is directly applicable to second-order differential equations. If we consider the N-zero-stable implicit operator $\bar{C}_{k2}^{(N)}$ defined by

$$\bar{C}_{K2}^{(N)}[z(x);\bar{c}_{sj}] = P_K[z(x);\bar{c}_{oj}] + R_{K1}[z'(x);\bar{c}_{1j}^{(N)}] \tag{6.154}$$

where the $\bar{c}_{1j}^{(N)}$, $0 \le j \le k$, are determined as functions of the stability parameters $\bar{c}_{oj}$ by setting

$$\bar{C}_{K2}^{(N)}[x^r] = 0 \qquad r = 2,\ldots,k+2-N \quad N \le k \tag{6.155}$$

and

$$\bar{C}_{K2}^{(N)}[\pi_i(x)] = 0 \qquad i = 1,\ldots,N \tag{6.156}$$

then it can be shown that the recurrence formula for the N-zero-stable implicit operator is

$$\bar{C}_{K2}^{(N+1)}[z(x)] = \bar{C}_{K2}^{(N)}[z(x)] - \frac{\bar{C}_{K2}^{(N)}[\pi_{N+1}(x)]}{h^2\bar{\bar{A}}_{KO}^{(N)}[\pi''_{N+1}(x)]}\, h^2\bar{\bar{A}}_{KO}^{(N)}[z''(x)] \tag{6.157}$$

provided the denominator is non-zero.

We note that the multistep method which has $\bar{C}_{22}^{(0)}$ as its associated operator is a fourth-order method of Störmer type:

$$y_{n+2} - 2y_{n+1}+y_n = \frac{h^2}{12}(f_{n+2}^{(1)}+10f_{n+1}^{(1)}+f_n^{(1)})$$

while one multistep method which has $\bar{C}_{42}^{(0)}$ as its associated operator is the sixth-order method of Störmer type:

$$y_{n+4}-2y_{n+3}+y_{n+2}=\frac{h^2}{240}(19f_{n+4}^{(1)}+204f_{n+3}^{(1)}+14f_{n+2}^{(1)}+4f_{n+1}^{(1)}-f_n^{(1)})$$

In problem A we seek a multistep method which is exact for x, x^2, $\sin x$ and $\cos x$. We can thus employ the recurrence formulae (6.157), (6.91) and (6.107) to construct the appropriate operator $\bar{C}_{22}^{(2)}$. In problem B we seek a multistep method which is exact for x, x^2, $\sin x$, $\cos x$, $x \sin x$ and $x \cos x$ and so we can construct the appropriate operator $\bar{C}_{42}^{(4)}$ again using the recurrence formulae (6.157), (6.91) and (6.107).

We apply the multistep methods determined from the operators $\bar{C}_{22}^{(2)}$ and $\bar{C}_{42}^{(4)}$ to problems A and B respectively taking the stepsize alternately as $\pi/18$, $\pi/9$, $\pi/6$, $\pi/3$ while the extra starting values required are taken from the theoretical solution. The calculations were performed using double precision on the IBM 360/67 at Newcastle University. In the following table, Table 6.7, we quote for each problem and each stepsize the error in the

numerical solution of $_1y$ and $_2y$ denoted by E_1 and E_2 respectively

Table 6.7

h	Problem A		Problem B	
	E_1	E_2	E_1	E_2
$\pi/18$	$-.806\times10^{-13}$	$.293\times10^{-12}$	$-.980\times10^{-13}$	$.264\times10^{-12}$
$\pi/9$	$-.289\times10^{-13}$	$.353\times10^{-13}$	$-.022\times10^{-13}$	$.347\times10^{-13}$
$\pi/6$	$-.363\times10^{-13}$	$.218\times10^{-13}$	$-.105\times10^{-14}$	$.288\times10^{-13}$
$\pi/3$	$-.999\times10^{-15}$	$-.236\times10^{-13}$	$-.135\times10^{-13}$	$.110\times10^{-13}$

The errors incurred in both problems are negligible and we observe that on these two problems we do not experience any loss of accuracy when deriving our methods by recurrence formulae.

Chapter Seven

Introduction

In the previous chapter we have outlined the idea of using non-polynomial based formulae in the solution of the initial value problem. It was shown how any linear combination of arbitrary functions could be used as the non-polynomial base and a method of obtaining non-polynomial based formulae from polynomial based formulae by means of recurrence formulae was derived. The use of this technique is however restricted to those problems where we can surmise what the appropriate combination of arbitrary functions is to be before we start the numerical solution. If however we are faced with a problem which has special difficulties, such as a singularity in the solution, then we may prefer to work with an adaptive formulae based on the representation (6.1) as proposed by Lambert and Shaw (1966a) and Shaw (1967) where the parameters, A and N, which control the position and nature of any possible singularity are continuously modified throughout the solution.

In this chapter our aim is to investigate whether an adaptive type of formula can be derived for an interpolant of the form

$$Y(x) = \sum_{i=0}^{L} a_i x^i + \sum_{m=1}^{N} b_m e^{\lambda_m x} \qquad b_m \neq 0, \lambda_m \neq 0, m=1,\ldots,N \qquad (7.1)$$

where the $L + 1$ constants $a_i (i=0,1\ldots,L)$ are regarded as undetermined coefficients while the $2N$ coefficients $b_1,\ldots,b_N,\lambda_1,\ldots,\lambda_N$ are to be determined by the behaviour of the particular differential equation whose solution is sought.

7.1 One-Step Explicit Methods

Analogously to Lambert and Shaw (1966a) we start by considering one-step explicit methods over the range $[x_n, x_{n+1}]$. To determine the L+1 undetermined coefficients $a_0, \ldots, a_L$ in (7.1) we need L + 2 equations. These equations are obtained by observing that if the theoretical solution $y(x)$, which we approximate by the numerical solution, is to be represented by the interpolant $Y(x)$ in the range $[x_n, x_{n+1}]$ the following must hold

$$\left.\begin{aligned} Y(x_n) &= y_n \\ Y(x_{n+1}) &= y_{n+1} \\ \left[\frac{d^s}{dx^s} Y(x)\right]_{x=x_n} &= f_n^{(s-1)} \qquad s = 1,2,\ldots,L \end{aligned}\right\} \tag{7.2}$$

where

$$f_n^{(s)} = \left[\frac{\partial f}{\partial x}^{(s-1)}(x,y) + f(x,y)\frac{\partial f}{\partial y}^{(s-1)}(x,y)\right]_{\substack{x=x_n \\ y=y_n}} \tag{7.3}$$

If we set $$P(x) = \sum_{i=0}^{L} a_i x^i \tag{7.4}$$

then from (7.1) we must have

$$P^{(s)}(x) = Y^{(s)}(x) - \sum_{m=1}^{N} b_m \lambda_m^s e^{\lambda_m x} \qquad s=0,1,\ldots,L \tag{7.5}$$

which using (7.2) yields

$$P^{(s)}(x_n) = f_n^{(s-1)} - \sum_{m=1}^{N} b_m \lambda_m^s e^{\lambda_m x_n} \qquad s=1,2,\ldots,L. \tag{7.6}$$

Now

$$Y(x_{n+1}) = P(x_{n+1}) + \sum_{m=1}^{N} b_m e^{\lambda_m x_{n+1}} . \tag{7.7}$$

Expanding $P(x_{n+1})$ by a Taylor series about x_n gives

$$Y(x_{n+1}) = P(x_n) + hP^{(1)}(x_n) + \frac{h^2}{2!}P^{(2)}(x_n) + \ldots + \frac{h^L}{L!}P^{L}(x_n) + \ldots$$
$$\ldots + \sum_{m=1}^{N} b_m e^{\lambda_m x_n} e^{\lambda_m h} \tag{7.8}$$

Substituting for the derivatives of P(x) from (7.6) and using (7.2) we obtain after simplification the following equation

$$y_{n+1} = y_n + \sum_{s=1}^{L} \frac{h^s}{s!} f_n^{(s-1)} + \sum_{m=1}^{N} b_m e^{\lambda_m x_n} \left[e^{\lambda_m h} - \sum_{s=0}^{L} h^s \frac{\lambda_m^s}{s!} \right] \tag{7.9}$$

Since the term in the square brackets is the difference between $e^{\lambda_m h}$ and the first L + 1 terms of the Taylor series for the same expression it follows that the perturbation term is of order at least h^{L+1}. Our definition of the local error under the usual localizing and differentiability assumptions is

$$T_{n+1} = y_{n+1} - y(x_{n+1}) \tag{7.10}$$

where

$$y(x_{n+1}) = \sum_{s=0}^{\infty} \frac{h^s}{s!} y^{(s)}(x_n) = y_n + \sum_{s=1}^{\infty} h^s f_n^{(s-1)} \tag{7.11}$$

Hence subtracting (7.11) from (7.9) gives

$$T_{n+1} = \sum_{p=1}^{\infty} \frac{T_p h^{L+p}}{L+p!} \tag{7.12}$$

where

$$T_p = -f_n^{(L+p-1)} + \sum_{m=1}^{N} b_m \lambda_m^{L+p} e^{\lambda_m x_n} \tag{7.13}$$

In order to calculate $b_1, \ldots, b_N, \lambda_1, \ldots, \lambda_N$ we set

$$T_1 = T_2 = \ldots = T_{2N} = 0 \tag{7.14}$$

This gives us a set of 2N simultaneous non-linear equations for 2N unknowns λ_m and b_m. Writing these equations out fully we have

$$f_n^{(L)} = \sum_{m=1}^{N} b_m \lambda_m^{L+1} e^{\lambda_m x_n} \tag{7.15.1}$$

$$f_n^{(L+1)} = \sum_{m=1}^{N} b_m \lambda_m^{L+2} e^{\lambda_m x_n} \tag{7.15.2}$$

$$
\left.
\begin{aligned}
f_n^{(L+N)} &= \sum_{m=1}^{N} b_m \lambda_m^{L+N+1} e^{\lambda_m x_n} && (7.15.N+1)\\
f_n^{(L+N+1)} &= \sum_{m=1}^{N} b_m \lambda_m^{L+N+2} e^{\lambda_m x_n} && (7.15.N+2)\\
&\vdots \qquad\qquad \vdots\\
f_n^{(L+2N-1)} &= \sum_{m=1}^{N} b_m \lambda_m^{L+2N} e^{\lambda_m x_n} && (7.15.2N)
\end{aligned}
\right\} \quad (7.15)
$$

If we let $\lambda_1, \lambda_2, \ldots, \lambda_N$ be the roots of the polynomial

$$\lambda^N - g_1\lambda^{N-1} - g_2\lambda^{N-2} - \ldots - g_{N-1}\lambda - g_N = 0 \qquad (7.16)$$

then multiplying equation (7.15.1) by $-g_N$, (7.15.2) by $-g_{N-1}, \ldots,$ (7.15.N) by $-g_1$ and adding the resulting equations to (7.15.N+1) gives

$$
\begin{aligned}
&-g_N f_n^{(L)} - g_{N-1} f_n^{(L+1)} - \ldots - g_1 f_n^{(L+N-1)} + f_n^{(L+N)} =\\
&\qquad \sum_{m=1}^{N} b_m \lambda_m^{L+1} e^{\lambda_m x_n} (-g_N - g_{N-1}\lambda_m - \ldots - g_1\lambda_m^{N-1} + \lambda_m^N)
\end{aligned}
\qquad (7.17)
$$

From (7.16) the value of the bracket on the right hand side is zero. Hence rearranging we have

$$g_N f_n^{(L)} + g_{N-1} f_n^{(L+1)} + \ldots + g_1 f_n^{(L+N-1)} = f_n^{(L+N)} \qquad (7.18)$$

If we now multiply (7.15.2) by $-g_N$, (7.15.3) by $-g_{N-1}\ldots$, (7.15.N+1) by $-g_1$, add the resultant equations to (7.15.N+2) and simplify using (7.16) we obtain

$$g_N f_n^{(L+1)} + g_{N-1} f_n^{(L+2)} + \ldots + g_1 f_n^{(L+N)} = f_n^{(L+N+1)} \qquad (7.19)$$

Starting one equation further down the set (7.15) each time and repeating the above technique, which is known as Prony's Method [Hildebrand (1956)], gives us the following set of equations written in matrix form as

$$\begin{bmatrix} f_n^{(L)} & f_n^{(L+1)} & \cdots & f_n^{(L+N-1)} \\ f_n^{(L+1)} & f_n^{(L+2)} & \cdots & f_n^{(L+N)} \\ \vdots & \vdots & & \vdots \\ f_n^{(L+N-1)} & \cdots & & f_n^{(L+2N-2)} \end{bmatrix} \begin{bmatrix} g_N \\ g_{N-1} \\ \vdots \\ g_1 \end{bmatrix} = \begin{bmatrix} f_n^{L+N} \\ f_n^{L+N+1} \\ \vdots \\ f_n^{L+2N-1} \end{bmatrix} \tag{7.20}$$

By inspection it can be seen that the resultant set of N simultaneous linear equations for the g's are symmetric about the leading diagonal. Hence once we have solved (7.20) for the g's this enables us to use (7.16) to find the values of λ_m, m=1,...,N. The values of the b's can then be obtained by solving the last N equations in (7.15) which are linear simultaneous equations for the b's or alternatively one could use a least squares technique on all the 2N equations.

It is possible that when solving a particular problem some of the values obtained from (7.16) for the λ_m,m=1,...,N may be complex. Since the coefficients of (7.16) are real all complex roots must occur in conjugate pairs. We must therefore allow the values of the b's to be complex as well as the λ's. In practice this corresponds to modifying the form of the interpolant (7.1) to include sine and cosine terms.

7.2 One-Step Implicit Methods

We now consider whether any one-step implicit methods can be derived. In this section it will be shown that various implicit methods can be derived by a similar procedure to those for explicit methods. If (7.1) is again taken to be the underlying interpolant we start by attempting to satisfy the following equations

$$\left.\begin{aligned} Y(x_n) &= y_n \\ Y(x_{n+1}) &= y_{n+1} \\ \left[\frac{d^s}{dx^s}Y(x)\right]_{x=x_{n+1}} &= f_{n+1}^{(s-1)} \qquad s = 1,2,\ldots,L. \end{aligned}\right\} \tag{7.21}$$

Proceeding as before we can develop the formula

$$y_{n+1} - y_n = \sum_{s=1}^{L} (-1)^{s-1} \frac{h^s}{s!} f_{n+1}^{(s-1)} + \sum_{m=1}^{N} b_m e^{\lambda_m x_{n+1}} \sum_{q=1}^{\infty} (-1)^{L+q-1} \frac{h^{L+q}}{(L+q)!} \lambda^{L+q} \tag{7.22}$$

with a local error given by

$$T_{n+1} = \sum_{p=1}^{\infty} \frac{T_p h^{L+p}}{(L+p)!} \tag{7.23}$$

where

$$T_p = \left[-f_n^{(L+p-1)} + \sum_{m=1}^{N} b_m e^{\lambda_m x_n} \lambda_m^{L+p}\right] \sum_{r=0}^{L} \frac{(-1)^{r+1}}{(L+p-r)!\,r!} \,. \tag{7.24}$$

If we then set

$$T_1 = T_2 = \ldots\ldots = T_{2N} = 0 \tag{7.25}$$

in order to calculate $b_1,\ldots,b_N,\lambda_1,\ldots\lambda_N$ we obtain the same set of equations given in (7.15). We can thus employ the same technique used in the derivation of the explicit methods to derive various implicit methods.

The methods obtained by making (7.1) satisfy the equations (7.21) are not the only one-step implicit methods that can be derived by this technique. Instead of (7.21) we might make (7.1) satisfy

$$\left.\begin{aligned} Y(x_n) &= y_n \\ Y(x_{n+1}) &= y_{n+1} \\ \frac{d}{dx}\left[Y(x)\right]_{x=x_n} &= f_n \\ \frac{d}{dx}\left[Y(x)\right]_{x=x_{n+1}} &= f_{n+1} \end{aligned}\right\} \qquad (7.26)$$

We can then carry out a similar procedure to those previously indicated to derive a one-step implicit method which effectively turns out to be a modification of the trapezoidal rule.

7.3 Explicit and Implicit Multistep Methods

The disadvantage of the one-step methods derived in the previous two sections is the necessity to calculate higher derivatives. Since this can be very time consuming for even the simplest of functions f we now investigate whether we can derive multistep methods based on (7.1) as the underlying interpolant. The approach we follow is comparable to that in Shaw (1967).

We will assume that the theoretical solution y(x)which we again approximate by the numerical solution, is to be represented by the interpolant Y(x) in the range $[x_n, x_{n+k}]$. The following equations must then be satisfied

$$\left.\begin{aligned} y_{n+t} &= Y(x_{n+t}) \qquad t = 0,1,\ldots,k \\ f_{n+t} &= \left[\frac{d}{dx} Y(x)\right]_{x=x_{n+t}} \qquad t = 0,1,\ldots,k' \end{aligned}\right\} \qquad (7.27)$$

where $k' = k - 1$ for explicit methods while $k' = k$ for implicit methods. If we set

$$Q_i = y_i - \sum_{m=1}^{N} b_m e^{\lambda_m x_i} \tag{7.28}$$

and

$$Q_i' = f_i - \sum_{m=1}^{N} b_m \lambda_m e^{\lambda_m x_i} \tag{7.29}$$

then equations (7.28) and (7.29) can be combined with (7.27) to give

$$Q_{n+t} = \sum_{i=0}^{L} a_i x_{n+t}^i \qquad t = 0,1,\ldots,k \tag{7.30}$$

and

$$Q'_{n+t} = \sum_{i=1}^{L} i a_i x_{n+t}^{i-1} \qquad t = 0,1,\ldots,k' \tag{7.31}$$

We will choose $L = k' - 2N + 1$ so that since $L \geq 0$, $k' \geq 2N-1$ while if L is zero the right hand side of (7.31) is identically zero.

Introducing the real numbers α_t, $t = 0,1,\ldots,k$ and β_t, $t = 0,1,\ldots,k'$ where

$$\left.\begin{aligned} \alpha_k &= 1 \\ \alpha_{k-1} &= \sum_{t=0}^{k} \alpha_t^{-1} \end{aligned}\right\} \tag{7.32}$$

implies

$$\sum_{t=0}^{k} \alpha_t = 0 \tag{7.33}$$

Multiplying Q_{n+t} in (7.30) by α_t, $t = 0,1,\ldots,k$, Q'_{n+t} in (7.31) by $-h\beta_t$, $t = 0,1,\ldots,k'$ and adding gives

$$\sum_{t=0}^{k} \alpha_t Q_{n+t} - h \sum_{t=0}^{k'} \beta_t Q'_{n+t} = \sum_{i=1}^{L} a_i \left[\sum_{t=0}^{k} \alpha_t x_{n+t}^i - ih \sum_{t=0}^{k'} \beta_t x_{n+t}^{i-1} \right] \tag{7.34}$$

where (7.33) has been used to eliminate a_o in (7.34) while if L is zero the right hand side of (7.34) is zero. Equating the coefficients a_i, $i = 1,\ldots, L$ to zero yields a system of $k'-2N+1$ equations in the $k'+1$ variables $\beta_0,\ldots,\beta_k$ which can be solved in terms of the parameters α_t, $t = 0,1,\ldots,k-2$. Since x_n may be taken as zero without any loss of generality we obtain the following

equations for the β_t, $t = 0,1,\ldots,k'$

$$\sum_{t=0}^{k'} t^{i-1}\beta_t = \frac{1}{i}\sum_{t=0}^{k} t^i\alpha_t \qquad i = 1,\ldots,k'-2N+1 \tag{7.35}$$

In solving (7.35) any 2N of the $\beta_t, t=0,1,\ldots,k'$ may be chosen arbitrarily and we will choose $\beta_0,\beta_1,\ldots,\beta_{2N-1}$. Equation (7.35) is then solved for $\beta_t, t = 2N,\ldots,k'$ in terms of the parameters $\alpha_0,\alpha_1,\ldots,\alpha_k,\beta_0,\ldots,\beta_{2N-1}$. Finally $\beta_0,\ldots,\beta_{2N-1}$ are chosen to be the solution of the 2N equations obtained by setting $i = k' - 2N+2,\ldots,k'+1$ in (7.35). Hence (7.34) becomes

$$\sum_{t=0}^{k} \alpha_t Q_{n+t} - h\sum_{t=0}^{k'}\beta_t Q'_{n+t} = 0 \tag{7.36}$$

(7.36) is thus a multistep method whose associated operator is

$$L[Q(x);h] = \sum_{t=0}^{k}\alpha_t Q(x+th) - h\sum_{t=0}^{k'}\beta_t Q'(x+th) \tag{7.37}$$

where the α_t are the stability parameters and the β_t are obtained from

$$\sum_{t=0}^{k'} t^{i-1}\beta_t = \frac{1}{i}\sum_{t=0}^{k} t^i\alpha_t \qquad i = 1,\ldots,k'+1 \tag{7.38}$$

Since (7.38) implies that the order is k'+1 then the local error of (7.37) is given by

$$L[Q(x);h] = \sum_{s=1}^{\infty} C_{k'+1+s}h^{k'+1+s}Q^{(k'+1+s)}(x) \tag{7.39}$$

where

$$C_r = \frac{1}{r!}\left(\sum_{j=0}^{k} j^r\alpha_j - \sum_{j=0}^{k'} rj^{r-1}\beta_j\right) \qquad r=k'+s+1 \tag{7.40}$$

Substituting in (7.36) from (7.28) and (7.29) we obtain

$$\sum_{t=0}^{k}\alpha_t y_{n+t} = h\sum_{t=0}^{k'}\beta_t f_{n+t} + \sum_{m=1}^{N} b_m\left\{\sum_{t=0}^{k}\alpha_t e^{\lambda_m x_{n+t}} - h\sum_{t=0}^{k'}\beta_t\lambda_m e^{\lambda_m x_{n+t}}\right\} \tag{7.41}$$

If we now introduce the forward difference operator Δ where $\Delta z_r = z_{r+1} - z_r$ and use the fact that $\Delta^L z_r = 0$ if z is a polynomial of degree less than or equal to L - 1 then we can return to (7.31) to obtain

$$\Delta^L Q'_{n+t} = 0 \qquad t = 0,1,2,\ldots,2N-1 \tag{7.42}$$

This gives us a set of 2N equations from which we can determine the 2N unknowns $b_1,\ldots,b_N,\lambda_1,\ldots,\lambda_N$. From (7.29) we have

$$\Delta^L Q'_i = \Delta^L f_i - \Delta^L \sum_{m=1}^{N} b_m \lambda_m e^{\lambda_m x_i}$$

Hence (7.42) implies

$$\Delta^L f_{n+t} = \Delta^L \sum_{m=1}^{N} b_m \lambda_m e^{\lambda_m x_{n+t}} \qquad t = 0,1,\ldots,2N-1 \tag{7.43}$$

It is straightforward to show that

$$\Delta^L \sum_{m=1}^{N} b_m \lambda_m e^{\lambda_m x_{n+t}} = \sum_{m=1}^{N} b_m \lambda_m e^{\lambda_m x_n} (e^{\lambda_m h} - 1)^L \tag{7.44}$$

Combining (7.43) and (7.44) together gives

$$\Delta^L f_{n+t} = \sum_{m=1}^{N} b_m \lambda_m e^{\lambda_m x_n} (e^{\lambda_m h} - 1)^L \qquad t = 0,1,\ldots,2N-1 \tag{7.45}$$

Writing these equations out fully we have

$$\left.\begin{aligned}
\Delta^L f_n &= \sum_{m=1}^{N} b_m \lambda_m e^{\lambda_m x_n} (e^{\lambda_m h}-1)^L && (7.46.1)\\
\Delta^L f_{n+1} &= \sum_{m=1}^{N} b_m \lambda_m e^{\lambda_m x_{n+1}} (e^{\lambda_m h}-1)^L && (7.46.2)\\
&\\
\Delta^L f_{n+N} &= \sum_{m=1}^{N} b_m \lambda_m e^{\lambda_m x_{n+N}} (e^{\lambda_m h}-1)^L && (7.46.N+1)\\
\Delta^L f_{n+N+1} &= \sum_{m=1}^{N} b_m \lambda_m e^{\lambda_m x_{n+N-1}} (e^{\lambda_m h}-1)^L && (7.46.N+2)\\
&\\
\Delta^L f_{n+2N-1} &= \sum_{m=1}^{N} b_m \lambda_m e^{\lambda_m x_{n+2N-1}} (e^{\lambda_m h}-1)^L && (7.46.2N)
\end{aligned}\right\} \quad (7.46)$$

If we let $e^{\lambda_1 h}, e^{\lambda_2 h}, \ldots, e^{\lambda_N h}$ be the roots of

$$e^{\lambda N h} - g_1 e^{\lambda (N-1) h} - g_2 e^{\lambda (N-2) h} \ldots - g_N = 0 \qquad (7.47)$$

then multiplying (7.46.1) by $-g_N$, (7.46.2) by $-g_{N-1}, \ldots,$ (7.46.N) by $-g_1$ and adding the resultant equations to (7.46.N+1) gives

$$-g_N \Delta^L f_n - g_{N-1} \Delta^L f_{n+1} - \ldots - g_1 \Delta^L f_{n+N-1} + \Delta^L f_{n+N}$$

$$= \sum_{m=1}^{N} b_m \lambda_m e^{\lambda_m x_n} (e^{\lambda_m h}-1)^L \{-g_N - g_{N-1} e^{\lambda_m h} - \ldots - e^{\lambda_m N h}\} \qquad (7.48)$$

From (7.47) the value of the bracket on the right hand side is zero. Hence rearranging we have

$$g_N \Delta^L f_n + g_{N-1} \Delta^L f_{n+1} + \ldots + g_1 \Delta^L f_{n+N-1} = \Delta^L f_{n+N} \qquad (7.49)$$

If we now multiply (7.46.2) by $-g_N$, (7.46.3) by $-g_{N-1}, \ldots,$ (7.46.N+1) by $-g_1$, add the resultant equations to (7.46.N+2) and simplify using (7.47) we obtain

$$g_N \Delta^L f_{n+1} + g_{N-1} \Delta^L f_{n+2} + \ldots + g_1 \Delta^L f_{n+N} = \Delta^L f_{n+N+1} \qquad (7.50)$$

Starting one equation further down the set (7.46) each time and repeating the same technique gives us the following set of equations written in matrix form as

$$\begin{bmatrix} \Delta^L f_n & \Delta^L f_{n+1} & \cdots & \Delta^L f_{n+N-1} \\ \Delta^L f_{n+1} & \Delta^L f_{n+2} & \cdots & \Delta^L f_{n+N} \\ \vdots & \vdots & & \vdots \\ \Delta^L f_{n+N-1} & \Delta^L f_{n+N} & \cdots & \Delta^L f_{n+2N-1} \end{bmatrix} \begin{bmatrix} g_N \\ g_{N-1} \\ \vdots \\ g_1 \end{bmatrix} = \begin{bmatrix} \Delta^L f_{n+N} \\ \Delta^L f_{n+N+1} \\ \vdots \\ \Delta^L f_{n+2N-1} \end{bmatrix} \tag{7.51}$$

We note that this set of N simultaneous linear equations for the g_m, m=1,2,...N, is symmetric about the leading diagonal. Hence once (7.51) has been solved for the g_m,m=1,2,...N we can find the roots of (7.47) which is an N-th degree polynomial in $e^{\lambda h}$ and hence determine the value of each λ_m. The values of the b_m, m=1,2,...N can then be obtained either from the last N equations in (7.46) which are linear simultaneous equations for the b's or alternatively by using a least squares technique on all the 2N equations.

The formula (7.41) can thus be used as modified multistep method with a local error of order k for an explicit method and order k+1 for an implicit method.

7.4 Numerical Testing of Exponentially Adaptive One-Step Methods and Multistep Methods

As a test of the methods derived in this chapter we will attempt to find numerical solutions for each of the following three problems.

Problem A

$$y' = x - y + 2 \qquad \text{range } [0,1]$$

with

$$y(0) = 2$$

This problem has the theoretical solution

$$y = 1 + x + e^{-x}$$

Problem B

$$y' = \frac{y \log y}{x} \qquad \text{range } [1,2]$$

with

$$y(1) = e^3$$

This problem has the theoretical solution

$$y = e^{3x}$$

Problem C

$$y' = y + 2e^{3x} - 2 \qquad \text{range } [3,5]$$

with

$$y(3) = e^3 + e^3 + 2$$

This problem has the theoretical solution

$$y = e^x + e^{3x} + 2$$

Thus problem A is a linear differential equation with a decreasing exponential in the solution while problem B is a non-linear equation with an increasing exponential in the solution. The third problem is a linear equation with two increasing exponentials in the solution.

Both the one-step and multistep methods we have outlined require us to find at each step, the solution of two sets of N simultaneous linear equations (one for the g_m, and one for the b_m) and the roots of an N-th degree polynomial. If, in addition, our method is implicit we must repeat all this work on each iteration. We therefore only consider explicit methods and keep the value of N as

small as possible by setting it equal to one.

Solutions for the three problems A, B and C have been obtained by each of the following methods:

the one-step adaptive method,

$$y_{n+1}=y_n+hf_n+\frac{h^2}{2!}f_n^{(1)}+be^{\lambda x_n}(e^{\lambda h}-1-h\lambda-\frac{h^2\lambda^2}{2!}) \tag{7.52}$$

the Taylor series fourth-order method,

$$y_{n+1}=y_n+hf_n+\frac{h^2}{2!}f_n^{(1)}+\frac{h^3}{3!}f_n^{(2)}+\frac{h^4}{4!}f_n^{(3)} \tag{7.53}$$

the multistep adaptive method,

$$\begin{aligned} y_{n+4}=y_{n+3}&+\frac{h}{24}[55f_{n+3}-59f_{n+2}+37f_{n+1}-9f_n] \\ &+b[e^{\lambda x_{n+4}}-e^{\lambda x_{n+3}} \\ &+\frac{h}{24}(55e^{\lambda x_{n+3}}-59e^{\lambda x_{n+2}}+37e^{\lambda x_{n+1}}-9e^{\lambda x_n})] \end{aligned} \tag{7.54}$$

and the Adams fourth-order method

$$y_{n+4}=y_{n+3}+\frac{h}{24}(55f_{n+3}-59f_{n+2}+37f_{n+1}-9f_n) \tag{7.55}$$

The calculations were carried out in double precision on the IBM 360/67 at Newcastle University and the extra starting values necessary for the multistep methods (7.54) and (7.55) were taken from the theoretical solution.

For each problem we quote in Table 7.1 the relative error at the end of the range in the numerical solution obtained by the methods (7.52), (7.53), (7.54) and (7.55) and the corresponding value of the theoretical solution.

Table 7.1

	Method				
Problem	(7.52)	(7.53)	(7.54)	(7.55)	Theor. Soln.
A	$.469\ 10^{-16}$	$.141\ 10^{-6}$	$.469\ 10^{-16}$	$.444\ 10^{-5}$	$0.23679\ 10^{1}$
B	$.141\ 10^{-14}$	$.211\ 10^{-3}$	$.705\ 10^{-15}$	$.451\ 10^{-2}$	$0.40343\ 10^{3}$
C	$.503\ 10^{-8}$	$.117\ 10^{-2}$	$.179\ 10^{-6}$	$.258\ 10^{-1}$	$0.32692\ 10^{7}$

From these results we can see that in problems A and B, where we can exactly represent the theoretical solution by an interpolant of the form (7.1) then the errors are negligible. In problem C however where we only include one exponential in our interpolant we observe that the numerical solution, obtained by either the one-step method (7.52) or the multistep method (7.54) is clearly more accurate than the solution obtained with either the Taylor Series fourth-order method or the Adams fourth-order method (7.56).

With such limited results it is impossible to draw any definite conclusions. The results however do indicate that this technique of adaptive exponential fitting may be beneficial if the theoretical solution is exponential in nature and we recommend that further investigations should be made on this particular topic.

Chapter Eight

Conclusions and Possible Future Extensions

8.1 Second-Derivative Stiffly-Stable Multistep Methods

In chapter four we have shown that particular choices for the third characteristic polynomial of second-derivative stiffly-stable multistep methods can result in a significant increase in the size of the stability region. Larger increases can be obtained by the use of optimization techniques where usually the size of the stability region increases with the number of coefficients we allow in the optimization process.

Subsequent work on this aspect might include alternative choices for both the third characteristic polynomial and the coefficients employed in the optimization. There is however no need to restrict this to the third characteristic polynomial and we could instead vary the first characteristic polynomial subject always to the constraints imposed by zero-stability. In addition we should like to test numerically many of the methods we have obtained with larger stability regions. The optimization technique could also be employed to maximize stability regions for first derivative methods and in particular Gear's method.

The method of order k + 3 which we have developed would appear, from the numerical results obtained, to have no real advantage over the method of order k + 2 proposed by Enright and indeed with very stiff problems it may be worse since the starting formula is not stable at infinity. This can restrict us to a very small starting stepsize. It may however be possible to combine both the order (k+2) and order (k+3) method to overcome this difficulty and take advantage of the smaller error constants of the higher order method.

8.2 Second-Derivative y-Implicit Runge-Kutta Methods

In Chapter five we have proposed L-stable second-derivative Runge-Kutta methods of orders three and four which are implicit with respect to y. This means that at each step of the calculation we need only solve a set of m simultaneous non-linear equations in contrast to the conventional k-implicit methods where we have to solve a set of mR simultaneous non-linear equations. The m non-linear equations for y_{n+1} are then solved by an iteration process based on the Newton-Raphson method.

Our fourth-order method is compared numerically with a fourth-order method of Butcher and we have shown that on the linear problems tested the new method performs considerably better than Butcher's method. We are currently engaged in further numerical testing of this new method in order to see how it compares on non-linear problems and we see this topic as a suitable area for further research.

8.3 Recurrence Relations for Non-Polynomially Based Operators

In Chapter six we have developed a complete set of recurrence relations for explicit and implicit operators associated with those linear multistep methods which are based on a representation involving a polynomial part and any number of arbitrary functions. Estimates of the work incurred in deriving these operators are formed in terms of the number of arithmetic operations involved and the amount of storage required. These estimates were compared with the corresponding quantities for obtaining the operator directly by the solution of a set of linear simultaneous equations and we showed that the conclusions are the same whether we consider arithmetic operations or storage. Tables are given to enable us to decide which method of derivation should be employed.

Although the work in the chapter is almost complete possible

extensions might include examining whether the structure of the recurrence formulae can be simplified in those cases when the non-polynomial part is comprised of a set of similar functions such as exponentials.

8.4 Exponentially Adaptive One Step Methods and Multistep Methods

In chapter seven we have derived, using Prony's method, one-step methods and multistep methods which are exponentially adaptive. We note that the amount of work involved at each step if we are using N exponentials is large since it includes the solution of two sets of N linear simultaneous equations and the determination of the roots of an N-th degree polynomial. It is therefore recommended that N be kept as small as possible and preferably set equal to one.

The numerical results we have taken appear to indicate that this type of method may be useful for problems which are exponential in nature. Any further work on this topic should initially consist of taking additional numerical results in order to properly evaluate the worth of this type of method.

References

G.D. Andria, G.D. Byrne and D.R. Hill (1973), Integration formulas and schemes based on g-splines. Math.Comp. 27 pp831-838.

P. Beaudet and T. Feagin (1975), The use of back correctors in multistep methods of numerical integration. Siam J.Numer.Anal. 12 pp895-918.

D.G. Bettis (1970), Numerical integration of products of Fourier and ordinary polynomials. Numer.Math. 14 pp421-434.

D.G. Bettis and E. Stiefel (1969), Stabilization of Cowell's Method. Numer.Math. 13 pp154-175.

G. Birkhoff and R.S. Varga (1965), Discretization errors for well-set Cauchy problems. Int.J.Maths and Phys. 44 pp1-23.

E.K. Blum (1962), A modification of the Runge-Kutta fourth-order method. Math.Comp. 16 pp176-187.

P. Brock and F.J. Murray (1952), The use of exponential sums in step by step integration. Math.Comp. 6 pp63-78 and 138-150.

D.G. Brush, J.J. Kohfeld and G.T. Thompson (1967), Solution of ordinary differential equations using two 'off-step' points. JACM 14 pp769-784.

R. Bulirsch and J. Stoer (1966), Numerical treatment of ordinary differential equations by extrapolation methods. Numer.Math. 8 pp1-13.

J.C. Burkill (1962), The Theory of Ordinary Differential Equations. University Mathematical Texts, Oliver and Boyd.

J.C. Butcher (1963), Coefficients for the study of Runge-Kutta integration processes. J.Austral.Math.Soc. 3 pp185-201.

J.C. Butcher (1964a), On Runge-Kutta processes of high order. J.Austral.Math.Soc. 4 pp179-194.

J.C. Butcher (1964b), Implicit Runge-Kutta processes. Math.Comp. 18 pp50-64.

J.C. Butcher (1965a), On the attainable order of Runge-Kutta methods. Math.Comp. 19 pp408-417.

J.C. Butcher (1965b), A modified multistep method for the numerical integration of ordinary differential equations. JACM 12 pp124-135.

J.C. Butcher (1975), An order bound for Runge-Kutta methods. Siam.J.Numer.Anal. 12 pp304-315.

G.D. Byrne (1967), Parameters for pseudo-Runge-Kutta methods. CACM 10 pp102-110.

G.D. Byrne and D.N. Chi (1972), Linear Multistep formulae based on g-splines. Siam.J.Numer.Anal. 9 pp316-324.

G.D. Byrne and R.J. Lambert (1966), Pseudo-Runge-Kutta methods involving two points JACM 13 pp114-123.

D.A. Calahan (1968), A-stable accurate method of numerical integration for nonlinear systems. Proc.IEEE 56 pp744.

E.D. Callender (1971), Single Step methods and low order splines for solutions of ordinary differential equations. Siam.J.Numer.Anal. 8 pp61-66.

J.W. Carr (1958), Error bounds for Runge-Kutta single step integration processes. JACM 5 pp39-44.

J.R. Cash (1975), A class of implicit Runge-Kutta methods for the numerical integration of stiff ordinary differential equations. JACM 22 pp504-511.

F. Ceschino and J. Kuntzmann (1966), Numerical solution of initial value problems, Prentice-Hall.

P.E. Chase (1962), Stability properties of predictor-corrector methods for ordinary differential equations. JACM 9 pp456-468.

F.H. Chipman (1971), Numerical solution of initial value problems using A-stable Runge-Kutta processes. University of Waterloo Dept. Applied Analysis and Computer Science, Research Report No. CSRR 2042.

N. Clark (1968), A study of some numerical methods for the integration of first-order ordinary differential equations. Argonne National Lab. Report No. 7428.

S.D. Conte and R.F. Reeves (1956), A Kutta third-order procedure for solving differential equations requiring minimum storage. JACM 3 pp22-25.

G.J. Cooper (1969), The numerical solution of stiff differential equations. FEBS Letters 2 (Supplement), pp522-529.

G.J. Cooper and J.H. Vemer (1972), Some explicit Runge-Kutta Methods of high order. Siam.J.Numer.Anal. 9 pp389-405.

P.C. Crane and P.A. Fox (1969), A comparative study of computer programs for integrating differential equations. Numerical Mathematics Program Library Project, Bell Telephone Lab.

R.L. Crane and R.W. Klopfenstein (1965), A predictor-corrector algorithm with an increased range of absolute stability. JACM 12 pp227-241.

R.L. Crane and R.J. Lambert (1962), Stability of a generalized corrector formula JACM 9 pp104-117.

G. Dahlquist (1956), Convergence and stability in the numerical integration of ordinary differential equations. Math.Scand.4 pp33-53.

G. Dahlquist (1959), Stability and error bounds in the numerical integration of ordinary differential equations. Kungl. Tekniska Hogskolans Handlingar No. 130.

G. Dahlquist (1963), A special stability problem for linear multistep methods. BIT 3 pp27-43.

C. Dill (1969), A computer graphic technique for finding numerical methods for ordinary differential equations. University of Illinois Dept. of Computer Science Report No. 295.

J. Donelson and E.H. Hansen (1971), Cyclic composite multistep predictor-corrector methods, J.Siam.Numer.Anal. 8 pp137-157.

R.J. Duffin (1969), Algorithms for classical stability problems. Siam Review 11 pp196-213.

B.L. Ehle (1968), High order A-stable methods for the numerical solution of systems of differential equations. BIT 8 pp276-278.

B.L. Ehle (1969), On Padé approximations to the exponential function and A-stable methods for the numerical solution of initial value problems. University of Waterloo Dept. Applied Analysis and Computer Science, Research Rep. No. CSRR 2010.

R. England (1969), Error estimates for Runge-Kutta type solutions to systems of ordinary differential equations. Comput.J. 12 pp166-170.

W. Enright (1972), Studies in the numerical solution of stiff ordinary differential equations. University of Toronto Dept. of Computer Science Technical Rep. No. 46.

E. Fehlberg (1964), New high-order Runge-Kutta formulas with step size control for systems of first- and second-order differential equations. Z.Angew.Math.Mech. 44 pp17-29.

E. Fehlberg (1968), Classical fifth-, sixth-, seventh- and eighth-order Runge-Kutta formulas with stepsize control. NASA Tech.Rep. No. 287 Huntsville.

E. Fehlberg (1969), Klassische Runge-Kutta-Formeln fünfter und siebenter Ordung mit Schrittweiten-Kontrolle. Computing 4 pp93-106.

E. Fehlberg (1970), Klassische Runge-Kutta-Formeln vierter und niedrigerer Ordung mit Schrittweiten-Kontrolle und ihre Anwendung auf Warmeleitungsprobleme. Computing 6 pp61-71.

M.E. Fowler and R.M. Warten (1967), A numerical integration technique for ordinary differential equations with widely separated eigenvalues IBM J. of Res. and Dev. 11 pp537-543.

D.J. Fyfe (1966), Economical evaluation of Runge-Kutta formulae. Math.Comp. 20 pp392-398.

B.A. Galler and D.P. Rozenberg (1960), A generalization of a theorem of Carr on error bounds for Runge-Kutta procedures. JACM 7 pp57-60.

W. Gautschi (1961), Numerical integration of ordinary differential equations based on trigonometric polynomials. Numer.Math. 3 pp381-397.

C.W. Gear (1964), Hybrid methods for initial value problems in ordinary differential equations. Siam.J.Numer.Anal. 2 pp69-86.

C.W. Gear (1967), The numerical integration of ordinary differential equations. Math.Comp. 21 pp146-156.

C.W. Gear (1968), The automatic integration of stiff ordinary differential equations. Information Processing 68. ed. A.J.H. Morrell, North Holland Publishing Co. pp187-193, 1969.

C.W. Gear (1971a), The automatic integration of ordinary differential equations CACM 14 pp176-179.

C.W. Gear (1971b), Algorithm 407, DIFSUB for solution of ordinary differential equations. CACM 14 pp185-190.

S. Gill (1951), A process for the step-by-step integration of differential equations in an automatic digital computing machine. Proc.Cambridge Philos.Soc. 47 pp95-108.

W.B. Gragg (1964), Repeated extrapolation to the limit in the numerical solution of ordinary differential equations. Ph.D. thesis. University of California.

W.B. Gragg and H.J. Stetter (1964), Generalized multistep predictor-corrector methods JACM 11 pp188-209.

C.F. Haines (1969), Implicit integration processes with error estimates for the numerical solution of differential equations. Comput.J. 12 pp183-187.

G. Hall (1967), The stability of predictor-corrector methods. Comput.J. 9 pp410-412.

R.W. Hamming (1959), Stable predictor-corrector methods for ordinary differential equations JACM 6 pp37-47.

P. Henrici (1962), Discrete Variable Methods in Ordinary Differential Equations. John Wiley and Sons.

P. Henrici (1963), Error Propagation for Difference Methods. John Wiley and Sons.

F.B. Hildebrand (1956), Introduction to Numerical Analysis. McGraw-Hill.

D.R. Hill (1975), On comparing Adams and natural spline multistep formulae Math.Comp. 29 pp741-745.

T.E. Hull (1967), A search for optimum methods for the numerical integration of ordinary differential equations. SIAM Rev. 9 pp647-654.

T.E. Hull and A.L. Creemer (1963), Efficiency of predictor-corrector procedures. JACM 10 pp291-301.

T.E. Hull, R. Bedet, W.H. Enright and J. Farkas (1974), Test results on initial value methods for non-stiff ordinary differential equations. University of Toronto Dept. of Computer Science Tech.Rep. No. 68.

T.E. Hull, W.H. Enright, B.M. Fellen and A.E. Sedgwick (1972), Comparing numerical methods for ordinary differential equations Siam.J.Numer.Anal. 9 pp603-637.

T.E. Hull and R.L. Johnston (1964), Optimum Runge-Kutta methods. Math.Comp. 18 pp306-310.

T.E. Hull and A.C.R. Newberry (1959), Error bounds for three point integration procedures. J.Soc.Indust.Math. 7 pp402-412.

T.E. Hull and A.C.R. Newberry (1961), Integration procedures which minimize propagated errors. J.Soc.Indust.Appl.Math. 9 pp31-47.

T.E. Hull and A.C.R. Newberry (1962), Corrector formulas for multistep integration methods J.Soc.Indust.Appl.Math. 10 pp351-369.

M.K. Jain and V.K. Srivastava (1970), High order stiffly stable methods for ordinary differential equations. University of Illinois Dept. of Computer Science Report No. 394.

R. King (1966), Runge-Kutta methods with constrained minimum error bounds. Math.Comp. 20 pp386-391.

R.W. Klopfenstein and R.S. Milliman (1968), Numerical stability of a one evaluation predictor-corrector algorithm for numerical solution of ordinary differential equations Math.Comp. 22 pp557-564.

J.J. Kohfeld and G.T. Thompson (1967), Multistep methods with modified predictors and correctors JACM 14 pp155-166.

J.J. Kohfeld and G.T. Thompson (1968), A modification of Nordsieck's method using an 'off-step' point JACM 15 pp390-401.

F.T. Krogh (1969), Variable order integrators for the numerical solution of ordinary differential equations. Jet Propulsion Lab.Tech.Memo. California Institute of Technology.

F.T. Krogh (1970), On testing a subroutine for the numerical integration of ordinary differential equations. Jet Propulsion Lab.Tech.Memo. No. 217 California Institute of Technology.

F.T. Krogh (1971), Suggestions on conversion (with listings) of the variable order integrators VODQ, SVDQ and DVDQ. Jet Propulsion Lab.Tech.Memo. No. 278 California Institute of Technology.

J. Kuntzmann (1959), Evaluation de l'erreur sur un pas dans les méthodes a pas séparès. Chiffres 2 pp97-102.

J.D. Lambert (1970), Linear multistep methods with mildly varying coefficients. Math.Comp. 24 pp81-94.

J.D. Lambert (1971), Predictor-corrector algorithms with identical regions of stability Siam.J.Numer.Anal. 8 pp337-344.

J.D. Lambert (1973), Computational Methods in Ordinary Differential Equations. John Wiley and Sons.

J.D. Lambert and A.R. Mitchell (1962), On the solution of $y' = f(x,y)$ by a class of high accuracy difference formulae of high order. Z.Angew.Math.Phys. 13 pp223-232.

J.D. Lambert and B. Shaw (1965), On the numerical solution of $y' = f(x,y)$ by a class of formulae based on rational approximation. Math.Comp. 19 pp456-462.

J.D. Lambert and B. Shaw (1966a), A method for the numerical solution of $y' = f(x,y)$ based on a self-adjusting non-polynomial interpolant. Math.Comp. 20 pp11-20.

J.D. Lambert and B. Shaw (1966b), A generalisation of multistep methods for ordinary differential equations. Numer.Math. 8 pp250-263.

J.D. Lambert and S.T. Sigurdsson (1972), Multistep methods with variable matrix coefficients. Siam.J.Numer.Anal. 9 pp715-733.

J.D. Lawson (1966), An order five Runge-Kutta process with extended region of stability. Siam.J.Numer.Anal. 3 pp593-597.

J.D. Lawson (1967), An order six Runge-Kutta process with extended region of stability. Siam.J.Numer.Anal. 4 pp620-625.

B. Lindberg (1971), On smoothing and extrapolation for the trapezoidal-rule BIT 11 pp29-52.

W. Liniger and R.A. Willoughby (1967), Efficient numerical integration methods for stiff systems of differential equations. IBM Research Report RC-1970.

H. Lomax (1967), An operational unification of finite difference methods for the numerical integration of ordinary differential equations. NASA Tech. Report NASA TRR-262.

H. Lomax (1968), Stable implicit and explicit numerical methods for integrating quasi-linear differential equations with parasitic, stiff and parasitic-saddle eigenvalues. NASA Tech. Note NASA TN D-4703.

F.R. Loscalzo (1968a), Numerical solution of ordinary differential equations by spline functions (SPLINDIF). Tech. Report 842. Math.Res.Center. Madison.

F.R. Loscalzo (1968b), On the use of spline functions for the numerical solution of ordinary differential equations. Tech. Report 869. Math.Res.Center. Madison.

F.R. Loscalzo and J.J. Schoenberg (1967), On the use of spline functions for the approximation of solutions of ordinary differential equations. Tech. Report 723 Math.Res.Centre. Madison.

F.R. Loscalzo and T.D. Talbot (1967a), Spline function approximations for solutions of ordinary differential equations. Bull.Amer.Math.Soc. 73 pp438-442.

F.R. Loscalzo and T.D. Talbot (1967b), Spline function approximations for solutions of ordinary differential equations. Siam.J.Numer.Anal. 4 pp433-445.

H.A. Luther (1966), Further explicit fifth-order Runge-Kutta formulas. SIAM Rev. 8 pp374-380.

H.A. Luther (1968), An explicit sixth-order Runge-Kutta formula. Math.Comp. 22 pp434-436.

H.A. Luther and H.P. Konen (1965), Some fifth-order classical Runge-Kutta formulas, SIAM Rev. 7 pp551-558.

T. Lyche (1969), Optimal order methods with an arbitrary number of nonstep points. Conference on the Numerical Solution of Differential Equations. Dundee ed. J.L. Morris. Springer.

T. Lyche (1972), Chebyshevian Multistep methods for ordinary differential equations. Numer.Math. 19 pp65-75.

M. Makelä, O, Nevanlinna, A. Sipilä (1973), On exponentially fitted multistep methods generated by generalized Hermite-Birkhoff interpolation. Conference of the Numerical Solution of Differential Equations, University of Dundee.

G.J. Makinson (1968), Stable high order implicit methods for the numerical solution of systems of differential equations. Comput.J. 11 pp305-310.

R.H. Merson (1957), An operational method for the study of integration processes. Proc.Symp. Data Processing, Weapons Research Establishment, Salisbury S. Australia.

W.E. Milne and R.R. Reynolds (1959), Stability of a numerical solution of differential equations. JACM, 6 pp193-203.

W.E. Milne and R.R. Reynolds (1962), Stability of a numerical solution of differential equations JACM 7 pp46-56.

A. Nordsieck (1962), On numerical integration of ordinary differential equations. Math.Comp. 16 pp29-49.

M.J.D. Powell (1964), An efficient method for finding the minimum of a function of several variables without calculating derivatives. Computer J. 7 pp155-162.

M.J.D. Powell (1968), On the calculation of orthogonal vectors. Computer J. 11 pp302-304.

H.S. Rahme (1969), A new look at the numerical integration of ordinary differential equations. JACM 16 pp496-506.

H.S. Rahme (1970), Stability analysis of a new algorithm used for integrating a system of ordinary differential equations. JACM 17 pp284-293.

A. Ralston (1961), Some theoretical and computational matters relating to predictor-corrector methods of numerical integration. Computer J. 4 pp1-4.

A. Ralston (1962), Runge-Kutta methods with minimum error bounds. Math.Comp. 16 pp431-437.

A. Ralston (1965), Relative stability in the numerical solution of ordinary differential equations. Siam.Rev. 7 pp114-125.

H.H. Robertson (1966), 'The solution of a set of reaction rate equations'. Numerical analysis: An introduction, J.Walsh ed., Academic Press pp178-182.

J.S. Rosen (1968), Multistep Runge-Kutta methods. NASA Technical Note, NASA TN D 4400.

H.H. Rosenbrock (1960), An automatic method for finding the greatest or least value of a function. Computer. J 3 pp175-184.

H.H. Rosenbrock (1963), Some general implicit processes for the numerical solution of differential equations. Computer J. pp329.

J.B. Rosser (1967), A Runge-Kutta for all seasons. Siam.Rev. 9 pp417-452.

C. Runge (1895), Uber die numerische Auflosung von Differentialgleichungen Math.Ann. 46 pp167-178.

D. Sarafyan (1966a), Error estimation for Runge-Kutta methods through Pseudo-iterative formulas. Louisiana State University Technical Report No. 14.

D. Sarafyan (1966b), Estimation of errors for the approximate solution of differential equations and their systems. Louisiana State University Technical Report No. 15.

D. Sarafyan (1967), Multi-order property of Runge-Kutta formulas and error estimation. Louisiana State University Technical Report No. 29.

K. Schoen (1971), Fifth and sixth order PECE algorithms with improved stability properties. Siam.J.Numer.Anal. 8 pp244-248.

R.E. Scraton (1964), Estimation of the truncation error in Runge-Kutta and allied processes. Computer J. 7 pp246-248.

A.E. Sedgwick (1973), An effective variable order variable step Adams method. Ph.D. thesis, Dept. of Computer Science Tech. Report No. 53 University of Toronto.

L.F. Shampine and H.A. Watts (1969), Block implicit one-step methods. Math.Comp. 23 pp731-740.

E.B. Shanks (1966), Solutions of differential equations by evaluation of functions Math.Comp. 20 pp21-38.

B. Shaw (1967), Modified multistep methods based on a non-polynomial interpolant. JACM 14 pp143-154.

H. Shintani (1965), Approximate computation of errors in numerical integration of ordinary differential equations by one-step methods. J.Sci. Hiroshima Univ. Ser. A-1 Math. 29 pp97-120.

H. Shintani (1966a), On a one-step method of order 4. J.Sci. Hiroshima Univ. Ser. A-1 Math. 30 pp91-107.

H. Shintani (1966b), Two-step processes by one-step methods of order 3 and of order 4. J.Sci. Hiroshima Univ. Ser. A-1 Math 30. pp183-195.

H.J. Stetter (1965a), Stabilizing predictors for weakly unstable correctors. Math.Comp. 9 pp84-89.

H.J. Stetter (1965b), A study of strong and weak stability in discretization algorithms. Siam.J.Numer.Anal. 2 pp265-280.

H.J. Stetter (1968), Improved absolute stability of predictor-corrector schemes. Computing 3 pp286-296.

N.F. Stewart (1968), The comparison of numerical methods for ordinary differential equations. University of Toronto Dept. of Computer Science Tech. Rep. No. 3

H.S. Wilf (1959), A stability criterion for numerical integration. JACM 6 pp363-366.

H.S. Wilf (1960), Maximally stable numerical integration. J.Soc.Indust.Appl.Math. 8 pp537-540.

Appendix 1

Stability Regions and Error Constants for Second-Derivative Multistep Methods

Table A1.1

Stability Regions and Error Constants for Second-Derivative Multistep Methods with a Third Characteristic Polynomial given by

$$\psi(\xi) = \xi^{k-r}(\xi - c)^r \qquad r = 1,2,\ldots,k \qquad -1<c<1$$

Table A1.1.1

$k = 2 \quad r = 1 \quad \psi(\xi) = \xi(\xi - c)$ A.S. denotes A-stable

c	-0.9	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	0.013	A.S.	A.S.	A.S.	A.S.	A.S.	A.S.	A.S.	A.S.	A.S.
b_s	2.62									
C_{p+1}	0.059	0.054	0.050	0.045	0.041	0.037	0.032	0.027	0.023	0.019

Table A1.1.2

$k = 2 \quad r = 2 \quad \psi(\xi) = (\xi - c)^2$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5
D	1.80	0.12	0.0014	A.S.	A.S.	0.027	1.54
b_s	1.6	2.03	2.01			2.02	2.04
C_{p+1}	0.0074	0.029	0.041	0.042	0.033	0.014	-0.015

Table A1.1.3

$k = 3 \quad r = 1 \qquad \psi(\xi) = \xi^2 (\xi - c)$

c	-0.9	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	0.87	0.36	0.22	0.15	0.12	0.094	0.081	0.075	0.076	0.090
b_s	1.60	1.70	1.77	1.86	1.95	2.05	2.16	2.27	2.37	2.43
C_{p+1}	0.031	0.029	0.027	0.025	0.023	0.021	0.020	0.018	0.016	0.014

Table A1.1.4

$k = 3 \quad r = 2 \qquad \psi(\xi) = \xi(\xi - c)^2$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5
D	2.41	0.63	0.26	0.13	0.088	0.12	1.41
b_s	1.57	1.64	1.74	1.90	2.10	2.25	2.24
C_{p+1}	0.023	0.025	0.026	0.024	0.020	0.015	0.0067

Table A1.1.5

$k = 3 \quad r = 3 \qquad \psi(\xi) = (\xi - c)^3$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3
D	2.83	0.56	0.35	0.15	0.089	1.21
b_s	1.24	1.51	1.68	1.85	2.16	2.13
C_{p+1}	0.047	0.033	0.027	0.024	0.019	0.0036

Table A1.1.6

$k = 4 \quad r = 1 \qquad \psi(\xi) = \xi^3 (\xi - c)$

c	-0.9	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	3.95	1.68	1.04	0.75	0.59	0.48	0.41	0.36	0.33	0.32
b_s	1.61	1.65	1.68	1.71	1.76	1.80	1.84	1.89	1.94	2.01
C_{p+1}	0.019	0.018	0.017	0.016	0.015	0.014	0.013	0.012	0.011	0.010

Table A1.1.7

$k = 4 \quad r = 2 \qquad \psi(\xi) = \xi^2 (\xi - c)^2$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5
D	8.69	2.40	1.13	0.66	0.44	0.39	1.46
b_s	1.61	1.63	1.68	1.74	1.80	1.93	2.11
C_{p+1}	0.017	0.017	0.017	0.015	0.013	0.011	0.0076

Table A1.1.8

$k = 4 \quad r = 3 \qquad \psi(\xi) = \xi(\xi - c)^3$

c	-0.5	-0.3	-0.1	0.1	0.3
D	5.39	1.71	0.74	0.41	1.25
b_s	1.55	1.65	1.73	1.84	2.04
C_{p+1}	0.017	0.017	0.016	0.013	0.0072

Table A1.1.9

$k = 4 \quad r = 4 \qquad \psi(\xi) = (\xi - c)^4$

c	-0.3	-0.1	0.1	0.3
D	2.68	0.83	0.40	4.81
b_s	1.60	1.71	1.88	2.15
C_{p+1}	0.017	0.016	0.012	0.0008

Table A1.1.10

$k = 5 \quad r = 1 \qquad \psi(\xi) = \xi^4 (\xi - c)$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	4.81	2.82	1.97	1.50	1.20	0.99	0.85	0.74	0.66
b_s	1.73	1.75	1.77	1.80	1.82	1.85	1.88	1.90	1.94
C_{p+1}	0.012	0.011	0.011	0.010	0.010	0.009	0.009	0.008	0.008

Table A.1.1.11

$k = 5 \quad r = 2 \qquad \psi(\xi) = \xi^3 (\xi - c)^2$

c	-0.5	-0.3	-0.1	0.1	0.3	0.5
D	7.05	3.02	1.69	1.09	0.80	1.50
b_s	1.71	1.73	1.79	1.84	1.90	1.99
C_{p+1}	0.012	0.011	0.011	0.0095	0.0081	0.0064

Table A1.1.12

$k = 5 \quad r = 3 \qquad \psi(\xi) = \xi^2 (\xi - c)^3$

c	-0.3	-0.1	0.1	0.3
D	4.82	1.91	0.99	1.30
b_s	1.70	1.77	1.84	1.98
C_{p+1}	0.012	0.011	0.0091	0.0054

Table A1.1.13

$k = 5 \quad r = 4 \qquad \psi(\xi) = \xi(\xi - c)^4$

c	-0.3	-0.1	0.1	0.3
D	7.91	2.16	0.92	4.82
b_s	1.70	1.77	1.86	2.14
C_{p+1}	0.012	0.011	0.0087	0.0040

Table A1.1.14

$k = 5 \quad r = 5 \qquad \psi(\xi) = (\xi - c)^5$

c	-0.1	0.1
D	2.45	0.88
b_s	1.76	1.89
C_{p+1}	0.011	0.0083

Table A1.1.15

$k = 6 \quad r = 1 \qquad \psi(\xi) = \xi^5(\xi - c)$

c	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	6.58	4.26	3.09	2.40	1.94	1.62	1.37	1.18
b_s	1.85	1.86	1.88	1.90	1.92	1.94	1.97	1.98
C_{p+1}	0.0083	0.0080	0.0076	0.0072	0.0070	0.0066	0.0062	0.0059

Table A1.1.16

$k = 6 \quad r = 2 \qquad \psi(\xi) = \xi^4(\xi - c)^2$

c	-0.3	-0.1	0.1	0.3	0.5
D	7.01	3.55	2.13	1.44	1.50
b_s	1.85	1.87	1.90	1.94	2.00
C_{p+1}	0.0083	0.0077	0.0070	0.0062	0.0052

Table A1.1.17

$k = 6 \quad r = 3 \qquad \psi(\xi) = \xi^3(\xi - c)^3$

c	-0.1	0.1	0.3
D	4.08	1.92	1.29
b_s	1.87	1.91	2.00
C_{p+1}	0.0079	0.0068	0.0053

Table A1.1.18

$k = 6 \quad r = 4 \quad \psi(\xi) = \xi^2(\xi - c)^4$

c	-0.1	0.1	0.3
D	4.71	1.73	4.92
b_s	1.83	1.94	2.10
C_{p+1}	0.0080	0.0066	0.0040

Table A1.1.19

$k = 6 \quad r = 5 \quad \psi(\xi) = \xi(\xi - c)^5$

c	-0.1	0.1
D	5.46	1.56
b_s	1.85	1.93
C_{p+1}	0.0081	0.0063

Table A1.1.20

$k = 6 \quad r = 6 \quad \psi(\xi) = (\xi - c)^6$

c	-0.1	0.1
D	6.34	1.45
b_s	1.85	1.98
C_{p+1}	0.0082	0.0060

Table A1.1.21

$k = 7 \quad r = 1 \quad \psi(\xi) = \xi^6(\xi - c)$

c	-0.1	0.1	0.3	0.5	0.7	0.9
D	5.94	4.47	3.53	2.86	2.37	1.99
b_s	1.92	1.94	1.96	1.98	2.00	2.02
C_{p+1}	0.0058	0.0056	0.0053	0.0051	0.0049	0.0046

Table A1.1.22

$k = 7 \quad r = 2 \qquad \psi(\xi) = \xi^5(\xi - c)^2$

c	-0.1	0.1	0.3	0.5
D	7.01	3.92	2.46	1.67
b_s	1.90	1.95	2.01	2.05
C_{p+1}	0.0059	0.0054	0.0049	0.0042

Table A1.1.23

$k = 7 \quad r = 3 \qquad \psi(\xi) = \xi^4(\xi - c)^3$

c	0.1	0.3
D	3.44	1.79
b_s	1.96	2.04
C_{p+1}	0.0053	0.0043

Table A1.1.24

$k = 7 \quad r = 4 \qquad \psi(\xi) = \xi^3(\xi - c)^4$

c	0.1	0.3
D	3.03	5.04
b_s	1.98	2.10
C_{p+1}	0.0051	0.0036

Table A1.1.25

$k = 7 \quad r = 5 \qquad \psi(\xi) = \xi^2(\xi - c)^5$

c	0.1
D	2.67
b_s	1.98
C_{p+1}	0.0050

Table A1.1.26

$k = 7 \quad r = 6 \qquad \psi(\xi) = \xi(\xi - c)^6$

c	0.1
D	2.36
b_s	2.00
C_{p+1}	0.0048

Table A1.1.27

$k = 7 \quad r = 7 \qquad \psi(\xi) = (\xi - c)^7$

c	0.1
D	2.13
b_s	2.00
C_{p+1}	0.0046

Table A1.2

Stability Regions and Error Constants for Second-Derivative Multistep Methods with a Third Characteristic Polynomial given by

$$\psi(\xi) = \xi^{k-r}(\xi^r - c^r) \qquad r = 2,\ldots,k \qquad -1<c<1$$

Table A1.2.1

$k = 2 \quad r = 2 \quad \psi(\xi) = \xi^2 - c^2$

c	-0.9	-0.7	-0.5	-0.3	-0.1
D	A.S.	A.S.	A.S.	A.S.	A.S.
b_s					
C_{p+1}	0.14	0.10	0.071	0.050	0.040

Table A1.2.2

$k = 3 \quad r = 2 \quad \psi(\xi) = \xi(\xi^2 - c^2)$

c	-0.9	-0.7	-0.5	-0.3	-0.1
D	0.25	0.12	0.098	0.097	0.10
b_s	1.73	1.83	1.92	1.99	2.00
C_{p+1}	0.043	0.035	0.029	0.025	0.023

Table A1.2.3

$k = 3 \quad r = 3 \quad \psi(\xi) = \xi^3 - c^3$

c	-0.9	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7
D	0.052	0.029	0.034	0.071	0.10	0.11	0.16	0.57	2.44
b_s	1.60	1.67	2.06	2.10	1.99	2.05	1.96	1.89	1.76
C_{p+1}	0.12	0.066	0.038	0.026	0.022	0.022	0.019	0.066	-0.021

Table A1.2.4

$k = 4 \quad r = 2 \quad \psi(\xi) = \xi^2(\xi^2 - c^2)$

c	-0.9	-0.7	-0.5	-0.3	-0.1
D	1.54	0.78	0.60	0.54	0.53
b_s	1.66	1.70	1.74	1.77	1.79
C_{p+1}	0.022	0.019	0.017	0.015	0.015

Table A1.2.5

$k = 4 \quad r = 3 \quad \psi(\xi) = \xi(\xi^3 - c^3)$

c	-0.9	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	0.55	0.40	0.43	0.50	0.53	0.53	0.56	0.74	1.68	8.35
b_s	1.58	1.68	1.74	1.77	1.78	1.78	1.78	1.80	1.84	1.90
C_{p+1}	0.030	0.022	0.017	0.015	0.014	0.014	0.014	0.012	0.007	-0.001

Table A1.2.6

$k = 4 \quad r = 4 \quad \psi(\xi) = (\xi^4 - c^4)$

c	-0.9	-0.7	-0.5	-0.3	-0.1
D	0.79	0.18	0.24	0.47	0.52
b_s	1.40	1.50	1.70	1.78	1.80
C_{p+1}	0.092	0.043	0.022	0.015	0.014

Table A1.2.7

$k = 5 \quad r = 2 \quad \psi(\xi) = \xi^3(\xi^2 - c^2)$

c	-0.9	-0.7	-0.5	-0.3	-0.1
D	5.07	2.17	1.62	1.42	1.34
b_s	1.74	1.76	1.79	1.79	1.80
C_{p+1}	0.013	0.012	0.011	0.010	0.010

Table A1.2.8

$k = 5 \quad r = 3 \quad \psi(\xi) = \xi^2(\xi^3 - c^3)$

c	-0.9	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	2.27	1.42	1.30	1.33	1.33	1.33	1.35	1.46	2.13	8.23
b_s	1.72	1.75	1.80	1.81	1.81	1.81	1.81	1.82	1.85	1.91
C_{p+1}	0.015	0.012	0.011	0.010	0.010	0.010	0.010	0.009	0.008	0.006

Table A1.2.9

$k = 5 \quad r = 4 \quad \psi(\xi) = \xi(\xi^4 - c^4)$

c	-0.9	-0.7	-0.5	-0.3	-0.1
D	1.17	0.87	1.15	1.31	1.33
b_s	1.57	1.72	1.78	1.81	1.82
C_{p+1}	0.021	0.014	0.011	0.010	0.010

Table A1.2.10

$k = 5 \quad r = 5 \quad \psi(\xi) = (\xi^5 - c^5)$

c	-0.7	-0.5	-0.3	-0.1
D	0.46	0.82	1.29	1.33
b_s	1.54	1.72	1.80	1.80
C_{p+1}	0.028	0.013	0.010	0.010

Table A1.2.11

$k = 6 \quad r = 2 \quad \psi(\xi) = \xi^4(\xi^2 - c^2)$

c	-0.7	-0.5	-0.3	-0.1
D	5.10	3.56	2.96	2.73
b_s	1.87	1.88	1.88	1.88
C_{p+1}	0.008	0.008	0.008	0.007

Table A1.2.12

$k = 6 \quad r = 3 \quad \psi(\xi) = \xi^3(\xi^3 - c^3)$

c	-0.9	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	6.97	3.35	2.88	2.74	2.71	2.70	2.68	2.68	3.04	8.92
b_s	1.83	1.86	1.88	1.88	1.89	1.89	1.90	1.90	1.91	1.93
C_{p+1}	0.009	0.008	0.008	0.007	0.007	0.007	0.007	0.007	0.007	0.006

Table A1.2.13

$k = 6 \quad r = 4 \quad \psi(\xi) = \xi^2(\xi^4 - c^4)$

c	-0.9	-0.7	-0.5	-0.3	-0.1
D	3.12	2.59	2.65	2.70	2.71
b_s	1.80	1.85	1.85	1.87	1.88
C_{p+1}	0.010	0.008	0.008	0.007	0.007

Table A1.2.14

$k = 6 \quad r = 5 \quad \psi(\xi) = \xi(\xi^5 - c^5)$

c	-0.9	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7
D	2.78	1.79	2.52	2.69	2.71	2.71	2.72	2.95	4.19
b_s	1.66	1.82	1.87	1.88	1.90	1.89	1.89	1.90	1.96
C_{p+1}	0.015	0.010	0.008	0.007	0.007	0.007	0.007	0.007	0.005

Table A1.2.15

$k = 6 \quad r = 6 \quad \psi(\xi) = (\xi^6 - c^6)$

c	-0.7	-0.5	-0.3	-0.1
D	0.86	2.13	2.68	2.71
b_s	1.63	1.84	1.88	1.89
C_{p+1}	0.008	0.008	0.007	0.007

Table A1.2.16

$k = 7 \quad r = 2 \quad \psi(\xi) = \xi^5(\xi^2 - c^2)$

c	-0.5	-0.3	-0.1
D	7.39	5.72	5.15
b_s	1.92	1.92	1.92
C_{p+1}	0.006	0.006	0.006

Table A1.2.17

$k = 7 \quad r = 3 \quad \psi(\xi) = \xi^4(\xi^3 - c^3)$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	7.65	5.79	5.20	5.10	5.10	5.03	4.77	4.49	9.27
b_s	1.92	1.93	1.93	1.94	1.94	1.94	1.94	1.94	1.95
C_{p+1}	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.005	0.005

Table A1.2.18

$k = 7 \quad r = 4 \quad \psi(\xi) = \xi^3(\xi^4 - c^4)$

c	-0.7	-0.5	-0.3	-0.1
D	5.96	5.26	5.10	5.10
b_s	1.91	1.93	1.93	1.93
C_{p+1}	0.006	0.006	0.006	0.006

Table A1.2.19

$k = 7 \quad r = 5 \quad \psi(\xi) = \xi^2(\xi^5 - c^5)$

c	-0.9	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7
D	5.71	4.95	5.05	5.09	5.10	5.10	5.11	5.23	5.78
b_s	1.85	1.91	1.92	1.93	1.93	1.93	1.93	1.93	1.94
C_{p+1}	0.008	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.005

Table A1.2.20

$k = 7 \quad r = 6 \quad \psi(\xi) = \xi(\xi^6 - c^6)$

c	-0.9	-0.7	-0.5	-0.3	-0.1
D	4.50	3.77	4.89	5.09	5.10
b_s	1.74	1.89	1.92	1.93	1.93
C_{p+1}	0.011	0.007	0.006	0.006	0.006

Table A1.2.21

$k = 7 \quad r = 7 \quad \psi(\xi) = (\xi^7 - c^7)$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5
D	1.36	4.47	5.09	5.10	5.10	5.12	5.76
b_s	1.75	1.91	1.93	1.93	1.93	1.93	1.93
C_{p+1}	0.013	0.006	0.006	0.006	0.006	0.006	0.005

Table A1.3

Stability Regions and Error Constants for Second-Derivative Multistep Methods with a Third Characteristic Polynomial given by

$$\psi(\xi) = \xi^{k-r} \sum_{i=o}^{r} \xi^{r-i} c^i \qquad r = 2,\ldots,k \qquad -1<c<1$$

Table A1.3.1

$k = 2 \quad r = 2 \qquad \psi(\xi) = \sum_{i=o}^{2} \xi^{2-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	2.87	0.42	0.002	A.S.	A.S.	A.S.	0.067	0.65	4.32
b_s	1.76	2.08	2.17				2.31	1.72	1.56
C_{p+1}	-0.040	-0.042	0.020	0.035	0.040	0.034	0.018	-0.008	-0.044

Table A1.3.2

$k = 3 \quad r = 2 \qquad \psi(\xi) = \xi \sum_{i=o}^{2} \xi^{2-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	2.40	0.38	0.11	0.095	0.12	0.16	0.24	0.50	2.68
b_s	2.10	2.14	2.16	2.08	1.97	1.88	1.83	1.81	1.78
C_{p+1}	0.003	0.011	0.017	0.021	0.023	0.022	0.020	0.016	0.010

Table A1.3.3

$k = 3 \quad r = 3 \qquad \psi(\xi) = \sum_{i=o}^{3} \xi^{3-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	5.55	1.24	0.19	0.097	0.12	0.11	0.064	0.057	0.34
b_s	1.78	1.92	2.06	2.08	1.98	1.90	1.93	1.68	0.96
C_{p+1}	-0.041	-0.005	0.014	0.021	0.023	0.026	0.036	0.060	0.10

Table A1.3.4

$k = 4 \quad r = 2 \qquad \psi(\xi) = \xi^2 \sum_{i=o}^{2} \xi^{2-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	2.44	0.59	0.43	0.48	0.58	0.71	0.88	1.18	3.36
b_s	2.05	1.93	1.84	1.80	1.76	1.74	1.73	1.72	1.73
C_{p+1}	0.007	0.010	0.012	0.014	0.015	0.015	0.015	0.014	0.012

Table A1.3.5.

$k = 4 \quad r = 3 \qquad \psi(\xi) = \xi \sum_{i=o}^{3} \xi^{3-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	4.40	1.15	0.48	0.48	0.58	0.68	0.71	0.68	0.84
b_s	2.06	1.98	1.86	1.80	1.76	1.72	1.68	1.61	1.51
C_{p+1}	-0.001	0.007	0.012	0.014	0.015	0.016	0.017	0.021	0.027

Table A1.3.6

$k = 4 \quad r = 4 \qquad \psi(\xi) = \sum_{i=0}^{4} \xi^{4-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7
D	6.15	1.67	0.55	0.48	0.58	0.76	1.46	4.93
b_s	1.90	1.96	1.88	1.78	1.75	1.74	1.75	1.80
C_{p+1}	-0.029	-0.001	0.011	0.014	0.015	0.015	0.010	-0.008

Table A1.3.7

$k = 5 \quad r = 2 \qquad \psi(\xi) = \xi^3 \sum_{i=0}^{2} \xi^{2-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	2.53	0.94	0.98	1.19	1.48	1.82	2.14	2.50	4.67
b_s	1.98	1.91	1.86	1.82	1.80	1.78	1.77	1.76	1.77
C_{p+1}	0.006	0.008	0.009	0.010	0.010	0.011	0.011	0.010	0.010

Table A1.3.8

$k = 5 \quad r = 3 \qquad \psi(\xi) = \xi^2 \sum_{i=0}^{3} \xi^{3-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	4.39	1.38	1.02	1.19	1.48	1.81	2.13	2.56	4.05
b_s	2.05	1.93	1.86	1.82	1.80	1.79	1.74	1.71	1.66
C_{p+1}	0.004	0.007	0.009	0.010	0.010	0.011	0.011	0.012	0.014

Table A1.3.9

$k = 5 \quad r = 4 \qquad \psi(\xi) = \xi \sum_{i=0}^{4} \xi^{4-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7
D	4.79	1.66	1.05	1.19	1.48	1.84	2.43	3.88
b_s	2.08	1.96	1.88	1.83	1.80	1.78	1.78	1.82
C_{p+1}	0.001	0.006	0.009	0.010	0.010	0.011	0.010	0.008

Table A1.3.10

$k = 5 \quad r = 5 \qquad \psi(\xi) = \sum_{i=0}^{5} \xi^{5-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	6.03	1.98	1.09	1.20	1.48	1.78	1.55	0.71	5.71
b_s	1.92	1.99	1.88	1.84	1.80	1.78	1.71	1.54	1.24
C_{p+1}	-0.018	-0.003	0.008	0.010	0.010	0.011	0.014	0.026	0.067

Table A1.3.11

$k = 6 \quad r = 2 \qquad \psi(\xi) = \xi^4 \sum_{i=0}^{2} \xi^{2-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	2.55	1.52	1.86	2.38	3.06	3.76	4.36	4.67	6.03
b_s	1.99	1.95	1.92	1.90	1.87	1.84	1.82	1.83	1.84
C_{p+1}	0.005	0.006	0.007	0.007	0.008	0.008	0.008	0.008	0.007

Table A1.3.12

$k = 6 \quad r = 3 \qquad \psi(\xi) = \xi^3 \sum_{i=0}^{3} \xi^{3-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7
D	4.47	1.75	1.87	2.37	3.06	3.85	4.83	6.53
b_s	2.03	1.95	1.92	1.90	1.88	1.86	1.84	1.83
C_{p+1}	0.004	0.006	0.007	0.007	0.008	0.008	0.008	0.008

Table A1.3.13

$k = 6 \quad r = 4 \qquad \psi(\xi) = \xi^2 \sum_{i=0}^{4} \xi^{4-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7
D	4.79	2.00	1.88	2.37	3.06	3.85	4.74	6.14
b_s	2.08	1.98	1.92	1.90	1.88	1.87	1.86	1.87
C_{p+1}	0.004	0.006	0.007	0.007	0.008	0.008	0.008	0.008

Table A1.3.14

$r = 6 \quad r = 5 \qquad \psi(\xi) = \xi \sum_{i=0}^{5} \xi^{5-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	4.73	2.16	1.90	2.37	3.06	3.83	4.49	4.17	2.68
b_s	2.10	2.00	1.93	1.90	1.88	1.87	1.86	1.80	1.70
C_{p+1}	0.001	0.005	0.007	0.007	0.008	0.008	0.008	0.010	0.014

Table A1.3.15

$k = 6 \quad r = 6 \qquad \psi(\xi) = \sum_{i=o}^{6} \xi^{6-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5
D	5.86	2.40	1.92	2.37	3.06	3.87	5.52
b_s	1.84	2.02	1.92	1.90	1.88	1.87	1.88
C_{p+1}	-0.011	0.004	0.007	0.007	0.008	0.008	0.007

Table A1.3.16

$k = 7 \quad r = 2 \qquad \psi(\xi) = \xi^5 \sum_{i=o}^{2} \xi^{2-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3	0.5	0.7	0.9
D	2.51	2.45	3.29	4.43	5.85	7.43	8.39	8.44	8.01
b_s	2.03	2.01	1.97	1.95	1.92	1.90	1.88	1.86	1.85
C_{p+1}	0.004	0.005	0.005	0.006	0.006	0.006	0.006	0.006	0.006

Table A1.3.17

$k = 7 \quad r = 3 \qquad \psi(\xi) = \xi^4 \sum_{i=o}^{2} \xi^{2-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3
D	4.41	2.48	3.26	4.43	5.86	7.73
b_s	2.05	2.01	1.96	1.94	1.92	1.90
C_{p+1}	0.004	0.005	0.005	0.006	0.006	0.006

Table A1.3.18

$k = 7 \quad r = 4 \qquad \psi(\xi) = \xi^3 \sum_{i=o}^{4} \xi^{4-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3
D	4.89	2.62	3.26	4.43	5.86	7.66
b_s	2.07	2.00	1.96	1.94	1.92	1.90
C_{p+1}	0.004	0.005	0.005	0.006	0.006	0.006

Table A1.3.19

$k = 7 \quad r = 5 \qquad \psi(\xi) = \xi^2 \sum_{i=o}^{5} \xi^{5-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3
D	4.81	2.75	3.27	4.43	5.86	7.67
b_s	2.09	2.01	1.96	1.94	1.92	1.90
C_{p+1}	0.003	0.005	0.005	0.006	0.006	0.006

Table A1.3.20

$k = 7 \quad r = 6 \qquad \psi(\xi) = \xi \sum_{i=o}^{6} \xi^{6-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3
D	4.78	2.85	3.28	4.43	5.86	7.68
b_s	2.05	2.02	1.97	1.94	1.92	1.90
C_{p+1}	0.002	0.004	0.005	0.006	0.006	0.006

Table A1.3.21

$k = 7 \quad r = 7 \qquad \psi(\xi) = \sum_{i=o}^{7} \xi^{7-i} c^i$

c	-0.7	-0.5	-0.3	-0.1	0.1	0.3
D	6.11	3.09	3.29	4.43	5.86	7.65
b_s	1.74	2.02	1.97	1.94	1.92	1.90
C_{p+1}	0.006	0.004	0.005	0.006	0.006	0.006

Table A1.4

Summary of the Largest Stability Regions obtained for Second-Derivative Multistep Methods when the Third Characteristic Polynomial is given by

$$\psi(\xi) = \xi^{k-r} (\xi - c)^r$$

k	r	c	D	C_{p+1}
3	1	.5	.075	.018
	2	.1	.088	.020
	3	.1	.089	.019
4	1	.9	.32	.010
	2	.3	.39	.011
	3	.1	.41	.013
	4	.1	.40	.012
5	1	.9	.66	.008
	2	.3	.80	.008
	3	.1	.99	.009
	4	.1	.92	.009
	5	.1	.88	.008
6	1	.9	1.18	.006
	2	.3	1.44	.006
	3	.3	1.29	.005
	4	.1	1.73	.007
	5	.1	1.56	.006
	6	.1	1.45	.006
7	1	.9	1.99	.005
	2	.5	1.67	.004
	3	.3	1.79	.004
	4	.1	3.03	.005
	5	.1	2.67	.005
	6	.1	2.36	.005
	7	.1	2.13	.005

Table A1.5

Summary of the Largest Stability Regions obtained for Second-Derivative Multistep Methods when the Third Characteristic Polynomial is given by

$$\psi(\xi) = \xi^{k-r}\,(\xi^r - c^r)$$

k	r	c	D	C_{p+1}
3	1	.5	.075	.018
	2	-.3	.097	.025
	3	-.3	.071	.026
4	1	.9	.32	.010
	2	-.1	.53	.015
	3	-.7	.40	.022
	4	-.7	.18	.043
5	1	.9	.66	.008
	2	-.1	1.34	.010
	3	.1	1.33	.010
	4	-.7	0.87	.014
	5	-.7	.46	.028
6	1	.9	1.18	.006
	2	-.1	2.73	.007
	3	.3	2.68	.007
	4	-.7	2.59	.008
	5	-.7	1.79	.010
	6	-.7	0.86	.008
7	1	.9	1.99	.005
	2	-.1	5.15	.006
	3	.7	4.49	.005
	4	-.1	5.10	.006
	5	-.7	4.95	.006
	6	-.7	3.77	.007
	7	-.7	1.36	.013

Table A1.6

Summary of the Largest Stability Regions obtained for Second-Derivative Multistep Methods when the Third Characteristic Polynomial is given by

$$\psi(\xi) = \xi^{k-r} \sum_{i=0}^{r} \xi^{r-i} c^{i}$$

k	r	c	D	C_{p+1}
3	1	.5	.075	.018
	2	-.1	.095	.021
	3	-.1	.097	.021
4	1	.9	.32	.010
	2	-.3	.43	.012
	3	-.1	.48	.014
	4	-.1	.48	.014
5	1	.9	.66	.008
	2	-.5	.94	.008
	3	-.3	1.02	.009
	4	-.3	1.05	.009
	5	.7	.71	.026
6	1	.9	1.18	.006
	2	-.5	1.52	.006
	3	-.5	1.75	.006
	4	-.3	1.88	.007
	5	-.3	1.90	.007
	6	-.3	1.92	.007
7	1	.9	1.99	.005
	2	-.5	2.45	.005
	3	-.5	2.48	.005
	4	-.5	2.62	.005
	5	-.5	2.75	.005
	6	-.5	2.85	.004
	7	.7	2.29	.013

Table A1.7

Optimized Stability Regions for Second-Derivative Multistep Methods.

We will use i to denote the number of coefficients in $\psi(\xi)$ that were used in the optimization process and $\psi^{(0)}(\xi)$ to denote the starting value of $\psi(\xi)$.

Table A1.7.1

k = 3 i = 1

w	γ_2	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-.61	.074	.017	ξ^3
.9	-.61	.074	.017	ξ^3
.8	-.61	.074	.017	ξ^3
.7	-.61	.074	.017	ξ^3
.6	-.597	.074	.017	ξ^3
.5	-.497	.075	.018	ξ^3
.4	-.397	.077	.019	ξ^3
.3	-.297	.081	.020	ξ^3
.2	-.197	.086	.021	ξ^3
.1	-.097	.094	.021	ξ^3

Table A1.7.2

k = 3 i = 2

w	γ_2	γ_1	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-.743	-.257	.0376	.022	$\xi(\xi-0.15)^2$
.9	-.655	-.220	.0423	.022	$\xi(\xi-0.15)^2$
.8	-.523	-.222	.0482	.023	ξ^3
.7	-.489	-.148	.0539	.022	$\xi(\xi-0.15)^2$
.6	-.397	-.122	.0605	.022	ξ^3
.5	-.343	-.078	.0676	.021	$\xi(\xi-0.15)^2$
.4	-.272	-.051	.0745	.021	ξ^3
.3	-.298	-.000	.0806	.020	$\xi(\xi-0.15)^2$
.2	-.312	-.022	.0853	.019	$\xi(\xi-0.15)^2$
.1	-.099	-.000	.0941	.021	ξ^3

Table A1.7.3

k = 3 i = 3

w	γ_2	γ_1	γ_0	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-.696	.122	.172	.0089	.035	ξ^3
.9	-.696	.122	.172	.0089	.035	ξ^3
.8	-.696	.122	.172	.0089	.035	ξ^3
.7	-.696	.122	.172	.0089	.035	ξ^3
.6	-.677	.0894	.102	.013	.027	$(\xi-0.15)^3$
.5	-.559	-.163	.105	.018	.035	$\xi^3-.343\xi^2-78\xi$
.4	-.464	-.0641	.047	.030	.026	$(\xi-0.15)^3$
.3	-.358	-.046	.021	.049	.023	$\xi^3-.337\xi^2+.011\xi$
.2	-.325	-.016	.0025	.079	.019	$(\xi-0.15)^3$
.1	-.130	-.004	.0007	.094	.021	ξ^3

Table A1.7.4

k = 4 i = 1

w	γ_3	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-.859	.316	.010	ξ^3
.9	-.859	.316	.010	ξ^3
.8	-.797	.320	.010	ξ^3
.7	-.697	.328	.011	ξ^3
.6	-.597	.341	.012	ξ^3
.5	-.497	.358	.012	ξ^3
.4	-.397	.381	.013	ξ^3
.3	-.297	.407	.013	ξ^3
.2	-.197	.438	.014	ξ^3
.1	-.097	.478	.014	ξ^3

Table A1.7.5

k = 4 i = 2

w	γ_3	γ_2	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-.845	-.155	.265	.012	ξ^4
.9	-.765	-.121	.283	.012	ξ^4
.8	-.678	-.098	.302	.012	ξ^4
.7	-.658	-.029	.322	.012	ξ^4
.6	-.586	-.0085	.339	.012	ξ^4
.5	-.516	.0080	.356	.012	$\xi^2(\xi-0.25)^2$
.4	-.525	.050	.371	.012	ξ^4
.3	-.506	.062	.379	.012	$\xi^2(\xi-0.25)^2$
.2	-.388	.038	.391	.012	ξ^4
.1	-.176	.0076	.446	.014	ξ^4

Table A1.7.6

k = 4 i = 3

w	γ_3	γ_2	γ_1	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-.542	.138	.309	.174	.017	$\xi(\xi-0.15)^3$
.9	-.542	.138	.309	.174	.017	$\xi(\xi-0.15)^3$
.8	-.542	.138	.309	.174	.017	$\xi(\xi-0.15)^3$
.7	-.786	.067	.154	.182	.013	$\xi(\xi-0.15)^3$
.6	-.860	-.045	.122	.186	.013	$\xi(\xi-0.15)^3$
.5	-.636	-.110	.090	.244	.014	$\xi(\xi-0.15)^3$
.4	-.540	-.007	.035	.307	.013	$\xi(\xi-0.15)^3$
.3	-.519	.042	.007	.358	.012	$\xi(\xi-0.15)^3$
.2	-.395	.038	.0002	.389	.012	ξ^4
.1	-.176	.0076	.000	.446	.014	ξ^4

Table A1.7.7

k = 4 i = 4

w	γ_3	γ_2	γ_1	γ_0	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-.429	.094	-.165	-.157	.081	.027	$(\xi-0.1)^4$
.9	-.429	.094	-.165	-.157	.081	.027	$(\xi-0.1)^4$
.8	-.439	.116	-.121	-.147	.081	.026	$(\xi-0.1)^4$
.7	-.483	.157	.206	-.105	.098	.028	$(\xi-0.1)^4$
.6	-.742	.061	.116	-.061	.092	.020	$(\xi-0.1)^4$
.5	-.771	.053	.128	-.043	.103	.018	$(\xi-0.1)^4$
.4	-.492	-.037	-.079	-.020	.208	.016	$(\xi-0.1)^4$
.3	-.493	-.014	.320	-.107	.273	.014	$(\xi-0.1)^4$
.2	-.443	.057	-.0007	-.0002	-387	.012	$(\xi-0.1)^4$
.1	-.176	.0076	.000	.000	.446	.012	ξ^4

Table A1.7.8

k = 5 i = 1

w	γ_4	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-.997	.634	.0073	ξ^5
.9	-.897	.664	.0077	ξ^5
.8	-.797	.701	.0079	ξ^5
.7	-.697	.741	.0082	ξ^5
.6	-.597	.793	.0084	ξ^5
.5	-.497	.853	.0087	ξ^5
.4	-.397	.921	.0090	ξ^5
.3	-.297	1.00	.0093	ξ^5
.2	-.197	1.09	.0095	ξ^5
.1	-.097	1.20	.0098	ξ^5

Table A1.7.9

k = 5 i = 2

w	γ_4	γ_3	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-.974	-.026	.631	.0075	ξ^5
.9	-.893	-.0059	.663	.0077	ξ^5
.8	-.842	.034	.692	.0076	ξ^5
.7	-.783	.058	.722	.0077	ξ^5
.6	-.779	.108	.747	.0075	ξ^5
.5	-.739	.120	.766	.0076	$\xi^3(\xi 10.3)^2$
.4	-.747	.139	.778	.0075	ξ^5
.3	-.480	.054	.861	.0086	ξ^5
.2	-.313	.040	.981	.0091	ξ^5
.1	-.318	.0099	1.15	.0097	ξ^5

Table A1.7.10

k = 5 i = 3

w	γ_4	γ_3	γ_2	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-.909	-.062	.136	.556	.0087	$\xi^2(\xi-0.1)^3$
.9	-.909	-.062	.136	.556	.0087	$\xi^2(\xi-0.1)^3$
.8	-.909	-.062	.136	.556	.0087	$\xi^2(\xi-0.1)^3$
.7	-.835	.030	.173	.560	.0087	$\xi^2(\xi-0.1)^3$
.6	-.864	-.040	.120	.581	.0086	ξ^5
.5	-.639	-.065	.082	.719	.0091	$\xi^2(\xi-0.1)^3$
.4	-.693	.074	.017	.741	.0080	$\xi^2(\xi-0.1)^3$
.3	-.539	.054	.0054	.822	.0084	$\xi^2(\xi-0.1)^3$
.2	-.409	.047	-.0007	.909	.0088	$\xi^2(\xi-0.1)^3$
.1	-.158	.006	.0002	1.13	.0096	ξ^5

Table A1.7.11

k = 5 i = 4

w	γ_4	γ_3	γ_2	γ_1	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-.595	.099	.212	-.143	.481	.012	$\xi(\xi-0.1)^4$
.9	-.595	.099	.212	-.143	.481	.012	$\xi(\xi-0.1)^4$
.8	-.595	.099	.212	-.143	.481	.012	$\xi(\xi-0.1)^4$
.7	-.595	.099	.212	-.143	.481	.012	$\xi(\xi-0.1)^4$
.6	-.435	.107	.044	-.101	.614	.010	ξ^5
.5	-.629	.044	.126	-.058	.576	.0099	$\xi(\xi-0.1)^4$
.4	-.686	.061	.076	-.022	.632	.0088	$\xi(\xi-0.1)^4$
.3	-.520	.069	.018	-.006	.805	.0086	$\xi(\xi-0.1)^4$
.2	-.445	.060	-.003	.000	.886	.0086	$\xi(\xi-0.1)^4$
.1	-.158	.006	.0002	.000	1.13	.0096	ξ^5

Table A1.7.12

k = 6 i = 1

w	γ_5	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-.997	1.103	.0057	ξ^6
.9	-.897	1.183	.0058	ξ^6
.8	-.797	1.274	.0060	ξ^6
.7	-.697	1.376	.0062	ξ^6
.6	-.597	1.490	.0064	ξ^6
.5	-.497	1.621	.0065	ξ^6
.4	-.397	1.772	.0067	ξ^6
.3	-.297	1.947	.0069	ξ^6
.2	-.197	2.151	.0071	ξ^6
.1	-.097	2.404	.0072	ξ^6

Table A1.7.13

k = 6 i = 2

w	γ_5	γ_4	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-1.076	.093	1.048	.0054	ξ^6
.9	-1.056	.141	1.071	.0053	ξ^6
.8	-1.018	.175	1.102	.0053	$\xi^4(\xi-0.3)^2$
.7	-0.990	.204	1.131	.0053	ξ^6
.6	-0.920	.192	1.165	.0054	$\xi^4(\xi-0.3)^2$
.5	-0.946	.224	1.164	.0053	$\xi^4(\xi-0.3)^2$
.4	-0.786	.155	1.256	.0058	$\xi^4(\xi-0.3)^2$
.3	-0.330	.090	1.813	.0067	ξ^6
.2	-0.313	.040	1.882	.0068	ξ^6
.1	-0.138	.010	2.277	.0072	ξ^6

Table A1.7.14

k = 6 i = 3

w	γ_5	γ_4	γ_3	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-1.076	.093	.000	1.048	.0054	ξ^6
.9	-0.979	.126	.064	1.057	.0056	ξ^6
.8	-0.979	.126	.064	1.057	.0056	ξ^6
.7	-0.979	.126	.064	1.057	.0056	ξ^6
.6	-0.979	.126	.071	1.061	.0057	ξ^6
.5	-0.921	.239	.003	1.181	.0054	$\xi^3(\xi-0.3)^3$
.4	-0.916	.254	-.019	1.237	.0053	$\xi^3(\xi-0.3)^3$
.3	-0.764	.204	-.018	1.299	.0057	ξ^6
.2	-0.348	.030	.001	1.830	.0068	ξ^6
.1	-0.171	.009	.000	2.199	.0071	ξ^6

Table A1.7.15

k = 6 i = 4

w	γ_5	γ_4	γ_3	γ_2	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-1.076	.093	.000	.000	1.048	.0054	ξ^6
.9	-0.979	.126	.064	.000	1.057	.0056	ξ^6
.8	-0.979	.126	.064	.000	1.057	.0056	ξ^6
.7	-0.979	.126	.064	.000	1.057	.0056	ξ^6
.6	-0.970	.126	.071	.000	1.061	.0057	ξ^6
.5	-0.806	.178	.075	-.042	1.110	.0060	$\xi^2(\xi-0.1)^4$
.4	-0.916	.254	-.019	.000	1.237	.0053	$\xi^3(\xi-0.3)^3$
.3	-0.764	.204	-.018	.000	1.299	.0057	ξ^6
.2	-0.494	.072	.0001	-.0005	1.582	.0064	$\xi^2(\xi-0.1)^4$
.1	-0.171	.009	.000	.000	2.199	.0071	ξ^6

Table A1.7.16

k = 7 i = 1

w	γ_6	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-.997	1.843	.0045	ξ^7
.9	-.897	1.999	.0045	ξ^7
.8	-.797	2.175	.0047	ξ^7
.7	-.697	2.375	.0049	ξ^7
.6	-.597	2.604	.0050	ξ^7
.5	-.497	2.868	.0051	ξ^7
.4	-.397	3.176	.0052	ξ^7
.3	-.297	3.539	.0053	ξ^7
.2	-.197	3.971	.0054	ξ^7
.1	-.097	4.492	.0056	ξ^7

Table A1.7.17

k = 7 i = 2

w	γ_6	γ_5	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-1.098	.160	1.602	.0042	ξ^7
.9	-1.090	.172	1.605	.0042	ξ^7
.8	-1.058	.215	1.621	.0042	ξ^7
.7	-1.003	.263	1.660	.0042	$\xi^5(\xi-0.5)^2$
.6	-1.003	.263	1.660	.0042	$\xi^5(\xi-0.5)^2$
.5	-0.997	.248	1.675	.0042	ξ^7
.4	-0.745	.159	2.003	.0046	ξ^7
.3	-0.330	.090	3.190	.0052	ξ^7
.2	-0.313	.040	3.371	.0053	ξ^7
.1	-0.138	.010	4.228	.0055	ξ^7

Table A1.7.18

k = 7 i = 3

w	γ_6	γ_5	γ_4	D	C_{p+1}	$\psi^{(0)}(\xi)$
1.0	-1.098	.160	.000	1.602	.0042	ξ^7
.9	-1.090	.172	.000	1.605	.0042	ξ^7
.8	-1.056	.245	-.028	1.618	.0041	ξ^7
.7	-1.034	.249	-.010	1.635	.0042	ξ^7
.6	-1.025	.222	.021	1.649	.0043	$\xi^4(\xi-0.3)^3$
.5	-0.997	.263	-.003	1.667	.0042	ξ^7
.4	-0.939	.282	-.025	1.734	.0042	$\xi^4(\xi-0.3)^3$
.3	-0.407	.126	-.003	2.857	.0050	ξ^7
.2	-0.313	.040	.000	3.371	.0053	ξ^7
.1	-0.234	.018	-.0005	3.743	.0054	ξ^7

Table A1.7.19

k = 7 $\hat{t}$ = 4

w	γ_6	γ_5	γ_4	γ_3	D	C_{p+1}
1.0	-1.100	.215	-.033	.001	1.585	.0041
.9	-1.100	.215	-.033	.001	1.585	.0041
.8	-1.056	.245	-.028	.000	1.618	.0041
.7	-1.034	.249	-.010	.000	1.635	.0042
.6	-1.025	.222	.021	.000	1.649	.0043
.5	-0.997	.263	-.003	.000	1.667	.0042
.4	-0.939	.282	-.025	.000	1.734	.0042
.3	-0.407	.126	-.003	.000	2.857	.0050
.2	-0.313	.040	.000	.000	3.371	.0053
.1	-0.252	.021	.000	.000	3.660	.0054

Table A1.8

A Search for Order k+3 Stiffly-Stable Second-Derivative Multistep Methods.

Once anything is found which would preclude use of the method the remaining information is omitted. We use A.P to denote the additional parameter used in fitting the increased order while N.S.S. denotes not stiffly-stable.

k	A.P.	$\max\lvert\xi_{1j}\rvert$	$\max\lvert\xi_{3j}\rvert$	D	b_s
1	γ_0	1.0	1.0	A-stable	
2	α_0	1.0	0.0	.51	2.01
	γ_0	1.0	0.55	.27	2.02
	γ_1	1.0	1.75		
3	α_0	1.0	0.0	1.21	1.87
	α_1	1.0	0.0	1.80	2.08
	γ_0	1.0	0.56	0.89	1.83
	γ_1	1.0	0.93	9.67	1.95
	γ_2	1.0	2.43		
4	α_0	1.0	0.0	2.12	1.89
	α_1	1.6			
	α_2	1.0	0.0	3.60	2.25
	γ_0	1.0	0.59	1.87	1.86
	γ_1	1.0	0.88	6.25	1.92
	γ_2	1.0	1.29		
	γ_3	1.0	3.07		
5	α_0	1.0	0.0	4.05	1.95
	α_1	1.0	0.0	4.78	2.07
	α_2	5.4			
	α_3	1.0	0.0	N.S.S.	
	γ_0	1.0	0.62	3.45	1.96
	γ_1	1.0	0.88	8.03	1.97
	γ_2	1.0	1.18		
	γ_3	1.0	1.64		
	γ_4	1.0	3.67		

Table A1.8 (Cont'd.)

k	A.P.	$\max\lvert\xi_{1j}\rvert$	$\max\lvert\xi_{3j}\rvert$	D	b_s
6	α_0	1.0	0.0	6.66	1.93
	α_1	1.5			
	α_2	1.0	0.0	N.S.S.	
	α_3	14.7			
	α_4	1.1			
	γ_0	1.0	0.0	6.16	1.94
	γ_1	1.0	0.89	10.64	1.92
	γ_2	1.0	1.15		
	γ_3	1.0	1.48		
	γ_4	1.0	1.98		
	γ_5	1.0	4.26		

Appendix 2

Additional Recurrence Relations for Non-Polynomial Based Operators

A2.1 Recurrence Relation for the N-Zero-Stable Explicit Operator

Consider the (N+1)-zero-stable implicit operator $\bar{B}^{(N+1)}_{K\ell}$ where the coefficients $\bar{b}^{(N+1)}_{sj}$ $1 \le s \le \ell$, $0 \le j \le k-1$ are determined as functions of the stability parameters $\bar{b}_{0j}$ by making

$$\bar{B}^{(N+1)}_{K\ell}[x^r] = 0 \qquad r = 1, \ldots, R-N-1 \tag{A2.1}$$

and

$$\bar{B}^{(N+1)}_{K\ell}[\pi_i(x)] = 0 \qquad i = 1, \ldots, N+1 . \tag{A2.2}$$

If we set

$$\bar{b}^{(N+1)}_{sj} = \bar{b}^{(N)}_{sj} + \bar{c}_{sj} \qquad 1 \le s \le \ell, 0 \le j \le k-1 \tag{A2.3}$$

and substitute for $\bar{b}^{(N+1)}_{sj}$ in $\bar{B}^{(N+1)}_{K\ell}[z(x)]$ we obtain

$$\bar{B}^{(N+1)}_{K\ell}[z(x)] = \bar{B}^{(N)}_{K\ell}[z(x)] + \mathcal{R}_{K-1,\ell}[z(x); \bar{c}_{sj}] . \tag{A2.4}$$

Since $\bar{B}^{(N)}_{K\ell}[x^r] = 0$ for r = 1,..., R-N-1 (A2.1) is equivalent to

$$\mathcal{R}_{K-1,\ell}[x^r; \bar{c}_{sj}] = 0 \qquad r = 1, \ldots R-N-1 \tag{A2.5}$$

Substituting for $\bar{b}^{(N+1)}_{sj}$ in (A2.2) from (A2.3) and simplifying shows that (A2.2) is equivalent to the conditions

$$\mathcal{R}_{K-1,\ell}[\pi_i(x); \bar{c}_{sj}] = 0 \qquad i = 1, \ldots, N \tag{A2.6}$$

and

$$\bar{B}^{(N+1)}_{K\ell}[\pi_{N+1}(x); \bar{b}_{sj} + \bar{c}_{sj}] = 0 . \tag{A2.7}$$

Defining

$$\bar{d}_{s-1,j} = -\bar{c}_{sj} / \bar{c}_{1,K-1} \qquad 1 \le s \le \ell, 0 \le j \le k-1, \bar{c}_{1,K-1} \neq 0 \tag{A2.8}$$

means that (A2.5) implies

$$-\bar{c}_{1,K-1}\sum_{S=0}^{\ell-1}\sum_{j=0}^{K-1}\bar{d}_{sj}h^s z^{(s)}(x+jh)=0 \qquad z(x)=1,x,\ldots,x^{R-N-2} \tag{A2.9}$$

while (A2.6) implies

$$-\bar{c}_{1,K-1}\sum_{s=0}^{\ell-1}\sum_{j=0}^{K-1}\bar{d}_{sj}h^s z^{(s)}(x+jh)=0 \qquad z(x)=\pi_1'(x),\ldots,\pi_N'(x). \tag{A2.10}$$

Since $\bar{d}_{0,k-1}=-1$ from (A2.8) while (A2.9) with $z(x)=1$ implies $\bar{d}_{0,k-1}=1-\sum_{j=0}^{k-2}\bar{d}_{0j}$ then from (A2.9) and (A2.10) we have

$$\mathcal{R}_{K-1,\ell}[z(x);\bar{c}_{sj}]=-\bar{c}_{1,K-1}\,h\bar{\bar{A}}^{(N)}_{K-1,\ell-1}[z'(x);\bar{\bar{\alpha}}'^{(N)}_{sj}]\,. \tag{A2.11}$$

Hence we can write

$$\bar{B}^{(N+1)}_{K\ell}[z(x)]=\bar{B}^{(N)}_{K\ell}[z(x)]-h\bar{c}_{1,K-1}\bar{\bar{A}}^{(N)}_{K-1,\ell-1}[z'(x)]\,. \tag{A2.12}$$

The condition (A2.7) will then be satisfied if we choose

$$\bar{c}_{1,K-1}=\frac{\bar{B}^{(N)}_{K\ell}[\pi_{N+1}(x)]}{h\bar{\bar{A}}^{(N)}_{K-1,\ell-1}[\pi'_{N+1}(x)]} \tag{A2.13}$$

assuming the denominator is non-zero.

Thus finally we arrive at the general recurrence formula for the N-zero-stable explicit non-polynomial operators.

$$\bar{B}^{(N+1)}_{K\ell}[z(x)]=\bar{B}^{(N)}_{K\ell}[z(x)]-\frac{\bar{B}^{(N)}_{K\ell}[\pi_{N+1}(x)]}{h\bar{\bar{A}}^{(N)}_{K-1,\ell-1}[\pi'_{N+1}(x)]}\,h\bar{\bar{A}}^{(N)}_{K-1,\ell-1}[z'(x)] \tag{A2.14}$$

$$\ell\geq 1,\ K\geq 1,\ N+1\in R$$

assuming again the denominator is non-zero.

A2.2 Recurrence Relation for the N-Optimum Explicit Operator

Consider the (N+1)-optimum explicit operator $\bar{\bar{B}}_{K\ell}^{(N+1)}$ where the coefficients $\bar{b}_{sj}^{(N+1)}$ $0 \le s \le \ell$, $0 \le j \le k-1$ are determined by making

$$\bar{\bar{B}}_{K\ell}^{(N+1)}[x^r] = 0 \qquad r = 0, 1, \ldots, S-N-2 \tag{A2.15}$$

and

$$\bar{\bar{B}}_{K\ell}^{(N+1)}[\pi_i(x)] = 0 \qquad i = 1, \ldots, N+1. \tag{A2.16}$$

If we set

$$\bar{\bar{b}}_{sj}^{(N+1)} = \bar{\bar{b}}_{sj}^{(N)} + \bar{\bar{c}}_{sj} \qquad 0 \le s \le \ell,\ 0 \le j \le k-1 \tag{A2.17}$$

and substitute for $\bar{b}_{sj}^{(N+1)}$ in $\bar{\bar{B}}_{sj}^{(N+1)}$ we obtain

$$\bar{\bar{B}}_{K\ell}^{(N+1)}[z(x)] = \bar{\bar{B}}_{K\ell}^{(N)}[z(x)] + \bar{\bar{A}}_{K-1,\ell}[z(x); \bar{\bar{c}}_{sj}] . \tag{A2.18}$$

Defining $$\bar{d}_{sj} = -\bar{\bar{c}}_{sj}/\bar{\bar{c}}_{0,k-1} \qquad 0 \le s \le \ell,\ 0 \le j \le k-1 \tag{A2.19}$$

(A.29) becomes

$$\bar{\bar{B}}_{K\ell}^{(N+1)}[z(x)] = \bar{\bar{B}}_{K\ell}^{(N)}[z(x)] - \bar{\bar{c}}_{0,K-1}\, \bar{\bar{A}}_{K-1,\ell}[z(x); d_{sj}] . \tag{A2.20}$$

Since $\bar{\bar{B}}_{K\ell}^{(N)}[x^r] = 0$ for $r = 0, \ldots, S-N-2$ using (A2.15) and (A2.20) implies

$$\bar{\bar{A}}_{K-1,\ell}[x^r; \bar{\bar{d}}_{sj}] = 0 \qquad r = 0, \ldots, S-N-2 \tag{A2.21}$$

while $\bar{\bar{B}}_{K\ell}^{(N)}[\pi_i(x)] = 0$ for $i = 1, \ldots, N$ together with (A2.16) and (A2.20) implies

$$\bar{\bar{A}}_{K-1,\ell}[\pi_i(x); \bar{\bar{d}}_{sj}] = 0 \qquad i = 1, \ldots, N. \tag{A2.22}$$

Conditions (A2.21) and (A2.22) together with $\bar{d}_{0k} = -1$ from (A2.20) are precisely the definition of the operator $\bar{\bar{A}}_{K-1,\ell}^{(N)}$.

Hence (A2.20) becomes

$$\bar{\bar{B}}^{(N+1)}_{K\ell}[z(x)] = \bar{\bar{B}}^{(N)}_{K\ell}[z(x)] - \bar{\bar{c}}_{0,K-1}\,\bar{\bar{A}}^{(N)}_{K-1,\ell}[z(x)] \quad . \tag{A2.23}$$

We can satisfy (A2.16) by taking

$$\bar{\bar{c}}_{0,K-1} = \frac{\bar{\bar{B}}^{(N)}_{K\ell}[\pi_{N+1}(x)]}{\bar{\bar{A}}^{(N)}_{K-1,\ell}[\pi_{N+1}(x)]} \tag{A2.24}$$

assuming the denominator is non zero.

Thus we have the general recurrence formula for the N-optimum explicit non-polynomial operator

$$\bar{\bar{B}}^{(N+1)}_{K\ell}[z(x)] = \bar{\bar{B}}^{(N)}_{K\ell}[z(x)] - \frac{\bar{\bar{B}}^{(N)}_{K\ell}[\pi_{N+1}(x)]}{\bar{\bar{A}}^{(N)}_{K-1,\ell}[\pi_{N+1}(x)]}\,\bar{\bar{A}}^{(N)}_{K-1,\ell}[z(x)] \qquad \begin{matrix}\ell \geq 1, K \geq 1 \\ N+1 \leq S\end{matrix} \tag{A2.25}$$

again assuming the denominator is non-zero

2.3 Recurrence Relation for the N-Optimum Implicit Operator $\bar{\bar{A}}^{(N)}_{0\ell}$

Consider the (N+1)-optimum implicit non-polynomial operator $\bar{\bar{A}}^{(N+1)}_{0\ell}$ $\ell \geq 1$ which is determined by

$$\bar{\bar{A}}^{(N+1)}_{0\ell}[x^{\tau}] = 0 \qquad \tau = 0,\ldots,\ell-N-2 \tag{A2.26}$$

and

$$\bar{\bar{A}}^{(N+1)}_{0\ell}[\pi_i(x)] = 0 \qquad i = 1,\ldots,N+1 \tag{A2.27}$$

We will assume that this operator has been normalized so that

$$\bar{\bar{a}}^{(N+1)}_{\ell 0} = \bar{\bar{a}}^{(N)}_{\ell 0} \tag{A2.28}$$

and set

$$\bar{\bar{a}}^{(N+1)}_{S0} = \bar{\bar{a}}^{(N)}_{S0} + \bar{\bar{c}}_{S0} \tag{A2.29}$$

where

$$\bar{\bar{c}}_{\ell 0} = 0 \tag{A2.30}$$

Substituting for $\bar{\bar{a}}^{(N+1)}_{S0}$ in $\bar{\bar{A}}^{(N+1)}_{0\ell}[z(x)]$ we obtain

$$\bar{\bar{A}}_{0\ell}^{(N+1)}[z(x)] = \bar{\bar{A}}_{0\ell}^{(N)}[z(x)] + A_{0,\ell-1}[z(x); \bar{\bar{c}}_{s0}]$$

Defining

$$\bar{\bar{d}}_{s0} = \bar{\bar{c}}_{s0} / \bar{\bar{c}}_{00} \qquad 0 \le S \le \ell - 1 \qquad \text{(A2.32)}$$

(A2.31) becomes

$$\bar{\bar{A}}_{0\ell}^{(N+1)}[z(x)] = \bar{\bar{A}}_{0\ell}^{(N)}[z(x)] - \bar{\bar{c}}_{00} A_{0,\ell-1}[z(x); \bar{\bar{d}}_{s0}] \qquad \text{(A2.33)}$$

Since $\bar{\bar{A}}_{0\ell}^{(N)}[x^r] = 0$ for $r = 0, \ldots, \ell-N-1$ using (A2.26)

and (A2.33) implies

$$A_{0,\ell-1}[x^r, d_{s0}] = 0 \qquad r = 0, \ldots, \ell-N-2 \qquad \text{(A2.34)}$$

while $\bar{\bar{A}}_{0\ell}^{(N)}[\pi_i(x)] = 0$ for $i = 1, \ldots N$ together with (A2.27)

and (A2.33) implies

$$A_{0,\ell-1}[\pi_i(x); \bar{\bar{d}}_{s0}] = 0 \qquad i = 1, \ldots N \qquad \text{(A2.35)}$$

Conditions (A2.34) and (A2.35) are precisely the

definition of the operator $\bar{\bar{A}}_{0,\ell-1}^{(N)}$. Hence (A2.33) becomes

$$\bar{\bar{A}}_{0\ell}^{(N+1)}[z(x)] = \bar{\bar{A}}_{0\ell}^{(N)}[z(x)] - \bar{\bar{c}}_{00} \bar{\bar{A}}_{0,\ell-1}^{(N)}[z(x)] \qquad \text{(A2.36)}$$

We can then satisfy (A2.27) by taking

$$c_{00} = \frac{\bar{\bar{A}}_{0\ell}^{(N)}[\pi_{N+1}(x)]}{\bar{\bar{A}}_{0,\ell-1}^{(N)}[\pi_{N+1}(x)]} \qquad \text{(A2.37)}$$

assuming the denominator is non-zero.

The recurrence formula for the optimum implicit non-

polynomial operator is thus

$$\bar{\bar{A}}_{0\ell}^{(N+1)}[z(x)] = \bar{\bar{A}}_{0\ell}^{(N)}[z(x)] - \frac{\bar{\bar{A}}_{0\ell}^{(N)}[\pi_{N+1}(x)]}{\bar{\bar{A}}_{0,\ell-1}^{(N)}[\pi_{N+1}(x)]} \bar{\bar{A}}_{0,\ell-1}^{(N)}[z(x)] \qquad \text{(A2.38)}$$

$\ell \ge 0 \quad k \ge 1$

again assuming the denominator is non-zero.