Pergamon

Computers Math. Applic. Vol. 36, No. 10–12, pp. 59–69, 1998
© 1998 Elsevier Science Ltd. All rights reserved
Printed in Great Britain
0898-1221/98 $19.00 + 0.00

PII: S0898-1221(98)00200-4

Lobatto Deferred Correction for Stiff Two-Point Boundary Value Problems

Z. BASHIR-ALI* AND J. R. CASH
Department of Mathematics, Imperial College of Science, Technology and Medicine
South Kensington, London SW7 2BZ, England

H. H. M. SILVA
Departamento de Ciências de Computação e Estatística
IBILCE-UNESP, C.P. 136, 15054-000, S.J. Rio Preto, SP, Brasil

Abstract—An iterated deferred correction algorithm based on Lobatto Runge-Kutta formulae is developed for the efficient numerical solution of nonlinear stiff two-point boundary value problems. An analysis of the stability properties of general deferred correction schemes which are based on implicit Runge-Kutta methods is given and results which are analogous to those obtained for initial value problems are derived. A revised definition of symmetry is presented and this ensures that each deferred correction produces an optimal increase in order. Finally, some numerical results are given to demonstrate the superior performance of Lobatto formulae compared with mono-implicit formulae on stiff two-point boundary value problems. © 1998 Elsevier Science Ltd. All rights reserved.

Keywords—Deferred correction, Lobatto formulae, Symmetry, Two-point boundary value problems.

1. INTRODUCTION

Iterated deferred correction is one of the most widely used methods for the numerical solution of the general first-order system of nonlinear two-point boundary value problems

$$\frac{dy}{dx} = f(x,y), \qquad a \le x \le b, \; g(y(a),\, y(b)) = 0. \tag{1.1}$$

The technique of deferred correction, which was originally proposed by Fox [1], now has many variants and practical applications. In what follows, we will discuss a particular form, namely, deferred difference correction, which will be of interest to us in this paper. To explain deferred difference correction, let ϕ be an approximate finite difference method for solving the equation

$$\bar{\phi}(\bar{\eta}) = 0,$$

and let η be the solution of

$$\phi(\eta) = 0. \tag{1.2}$$

A single step of a deferred difference correction comes from solving

$$\phi(\bar{\eta}) = \psi(\eta), \tag{1.3}$$

*The author wishes to express her gratitude to the Engineering and Physical Sciences Research Council for their financial support.

Typeset by $\mathcal{A}_\mathcal{M}\mathcal{S}$-TEX

where $\psi \equiv \phi - \bar{\phi}$. Here the operator ψ serves to compute a discrete approximation to the principal local error term in ϕ. It can be seen that the deferred difference correction schemes of this class are characterized by specifying particular finite difference methods ϕ and $\bar{\phi}$.

The original ideas of Fox [1] were put on a firm theoretical basis by Pereyra [2] and were further generalized by Stetter [3] and Lindberg [4]. In particular, Lindberg proposed using the approach

$$\begin{aligned} &\phi_p(\eta) = 0, \\ &\phi_p(\bar{\eta}) = \phi_p(\eta) - \phi_{p+r}(\eta), \end{aligned} \tag{1.4}$$

where ϕ_p, ϕ_{p+r} are numerical methods of order $p, p+r$, respectively. This approach is a very powerful one since it provides a general way of defining the correction operator ψ. A general framework for providing accuracy results for deferred correction schemes of the form (1.2),(1.3) was given in an influential paper by Skeel [5]. In what follows, we present his main theorem.

Consider the approximate numerical solution of (1.1) on a mesh

$$\pi : a = x_1 < x_2 < \cdots < x_{N+1} = b.$$

Denote by Δy the restriction of the continuous solution $y(x)$ to the finite grid π. Then, we have the following theorem.

THEOREM. *(See [5].) Let ϕ be a stable numerical method and assume that the following conditions hold for the deferred correction scheme (1.2),(1.3):*

(i) $\|\eta - \Delta y\| = O(h^p)$,
(ii) $\|\psi(\Delta y) - \phi(\Delta y)\| = O(h^{r+p})$,
(iii) $\psi(\Delta w) = O(h^r)$,

for arbitrary functions w having at least r continuous derivatives. Here $\|.\|$ is a suitable finite norm defined in [5]. If $\phi(\bar{\eta}) = \psi(\eta)$, then

$$\|\bar{\eta} - \Delta y\| = O\left(h^{r+p}\right).$$

It is easy to see that for the deferred correction scheme (1.4), the first two conditions of Skeel's theorem are trivially satisfied. In general, it is Condition (iii) that is the difficult one to satisfy in practice and it is this that provides the barrier to deriving efficient high-order schemes. In effect, Conditions (i) and (ii) ensure that we are using sufficiently high-order operators ϕ and ψ to obtain the desired increase in order. The purpose of Condition (iii) is to ensure that we annihilate sufficiently many terms in the local error expansion of $\phi(\eta)$ to allow us to obtain a solution $\bar{\eta}$ with order $r + p$.

In his original papers, Fox considered the approach whereby each deferred correction step is iterated to convergence. In such an approach, successive iterates would typically be defined by

$$\begin{aligned} \phi\left(\eta^1\right) &= 0, \\ \phi\left(\eta^{i+1}\right) &= \psi\left(\eta^i\right), \qquad i = 1, 2, \dots . \end{aligned}$$

However, more recent deferred correction algorithms [6–8] have been based on an approach whereby

(i) equation (1.2) is solved exactly (to a predetermined precision) for η;
(ii) equation (1.3) is solved exactly for $\bar{\eta}$ (again to the prescribed precision);
(iii) $\bar{\eta}$ is accepted as the final solution for the current grid.

This implies that when the algorithm defined by (i)–(iii) is used, it is not normally the case that

$$\bar{\phi}(\bar{\eta}) = 0.$$

This general approach is often referred to as linearized deferred correction and for an analysis of this, together with an overview of several other forms of deferred correction, the reader is referred to [3].

An early code which implemented deferred correction was PASVA due to Lentini and Pereyra [9]. In this particular approach, the basic low-order method ϕ is the trapezium rule, the solution of which gives an $O(h^2)$ approximation η^1 to the exact solution of (1.1). It is well known that, with a mesh spacing $h_i \equiv x_{i+1} - x_i$, the local truncation error of the trapezium rule has an asymptotic expansion of the form

$$\tau_i[y] = \sum_{j=1}^{r} h_i^{2j} T_j \left[y\left(x_{i+1/2}\right)\right] + O\left(h_i^{2r+2}\right), \tag{1.5}$$

i.e., containing only even powers of h_i. In the Lentini-Pereyra approach, successively higher-order difference schemes are obtained using

$$\phi\left(\eta^{k+1}\right) = \psi_k\left(\eta^k\right), \qquad k = 1, 2, \ldots, \tag{1.6}$$

where the ψ_k are $O(h^{2k+2})$ difference approximations to the leading terms in the expansion (1.5). Despite suffering from the so-called "end of the net catastrophe", PASVA has proved to be an excellent code for solving general nonstiff problems of the form (1.1). Furthermore, Pereyra and his coworkers have established many of the important principles that are used in more recent deferred correction schemes. However, PASVA has not in general proved to be nearly as effective for the numerical solution of stiff problems. This is due in part to the fact that higher-order corrections involve function evaluations at many adjacent grid points. This means that, when applying a correction to a mesh interval lying outside the stiff region, data from inside the stiff region may be used to compute the correction term and this is undesirable. Pereyra's deferred correction approach (1.6) was extended to special second-order equations of the form

$$y'' = f(x, y), \qquad y(a) = \alpha, \qquad y(b) = \beta,$$

by Daniel and Martin [10]. Here, the basic numerical method used is Numerov's method which, for uniform grid spacing, also has a local truncation error with asymptotic expansion containing only even powers of h.

A rather more effective deferred correction approach for solving stiff problems of the general form (1.1) was developed in [7,11]. This approach, which has the desirable property that deferred corrections are localized to a single mesh interval, is based on the use of Mono-Implicit Runge-Kutta (MIRK) formulae. MIRK formulae are characterized by the special property that, for initial value problems, they can be written in terms of the single unknown y_{n+1}. This makes them particularly cheap for the solution of boundary value problems in a deferred correction framework since the correction terms are now explicit. A FORTRAN code, TWPBVP, which is based on MIRK formulae and is available from NETLIB, has been developed and this code has proved to be effective for a large class of both stiff and nonstiff problems of the general form (1.1) [11]. However, experiments with a continuation algorithm [12] have shown that for extremely stiff problems, the code TWPBVP is not as efficient as might at first be expected. Extensive practical experience has shown that for mildly stiff and stiff problems, the code TWPBVP is normally able to adapt the mesh very quickly to the physical characteristics of the solution and this normally allows the solution to be computed in an efficient manner. However, for extremely stiff problems, the TWPBVP code often removes points from a grid on the grounds of accuracy, but then has to include them again at the next remeshing due to loss of stability. Of course, the main reason for this loss of stability is due to the fact that the correction term is explicit and the effects of this were analysed in [7].

One way of removing these stability problems is to use deferred correction schemes which are such that the correction is defined implicitly. Of course, the disadvantage will be that such schemes are normally more expensive per step than those based on MIRK formulae. However, if these implicit schemes are chosen carefully, then this extra cost will be more than compensated for by the improved stability obtained. We have analyzed and implemented several different deferred correction schemes for the numerical solution of linear problems of the form (1.1) and we have found that, in the linear case, formulae based on Lobatto schemes are the most promising for extremely stiff problems, while for problems which are not extremely stiff, the code TWPBVP is normally the most efficient. Of course, the terms nonstiff, stiff, and extremely stiff are not at all precise (due to the imprecise definition of stiffness itself), but these terms will become clearer when we consider the numerical results given in Section 4. We will also see from these results that our expectations concerning the relative performance of MIRK and Lobatto codes are justified for nonlinear, as well as for linear problems.

2. A SYMMETRY PROPERTY FOR DEFERRED CORRECTION FORMULAE

In this section, we consider the symmetry properties that our Runge-Kutta formulae need to possess to guarantee that each deferred correction achieves an extra two orders of accuracy. To investigate this, it is sufficient to consider the deferred correction algorithm

$$\begin{aligned} \phi_4(\eta) &= 0, \\ \phi_4(\bar{\eta}) &= \phi_4(\eta) - \phi_6(\eta), \end{aligned} \tag{2.1}$$

where ϕ_4, ϕ_6 are symmetric Runge-Kutta formulae of order 4 and 6, respectively. It is straightforward to prove order of accuracy results for (2.1) by appealing to the theorem presented by Skeel [5]. In particular, if we appeal to Skeel's main result which was discussed in the previous section, it follows that $\bar{\eta}$ is an order 6 approximation to Δy provided that

$$\phi_4(\Delta w) - \phi_6(\Delta w) = O\left(h^2\right), \tag{2.2}$$

for arbitrary functions Δw having the necessary smoothness properties. An analysis of condition (2.2) was carried out in [13] for the case where ϕ_4, ϕ_6 are Lobatto IIIa formulae (see, for example [14,15] for a description of these formulae). It was shown that in this case, condition (2.2) is not satisfied and an explicit computation verified that (2.1) only raised the order from 4 to 5. At first sight, this may seem a little surprising since both ϕ_4 and ϕ_6 are symmetric Runge-Kutta formulae. However, closer examination reveals that the usual definition of symmetry is inappropriate if we wish to ensure that (2.2) holds. This was first pointed out in [8], and in what follows, we formally give a revised definition of the concept of symmetry which is sufficient to ensure that (2.2) is satisfied. It will turn out that this revised definition will reduce to the normal definition of symmetry if the higher-order formula is used in the conventional way, i.e.,

$$\phi_6\left(\bar{\eta}\right) = 0.$$

To motivate our new definition of symmetry, we consider the standard s-stage Runge-Kutta formula

$$\begin{aligned} y_{n+1} &= y_n + h \sum_{i=1}^{s} b_i k_i, \\ k_i &= f\left(x_n + c_i h,\ y_n + h \sum_{j=1}^{s} a_{ij} k_j\right), \qquad 1 \le i \le s. \end{aligned} \tag{2.3}$$

Replacing h by $-h$, we have

$$y_{n-1} = y_n - h\sum_{i=1}^{s} b_i k_i, \tag{2.4a}$$

$$k_i = f\left(x_n - c_i h,\ y_n - h\sum_{j=1}^{s} a_{ij} k_j\right) \tag{2.4b}$$

$$= f\left(x_{n-1} + (1 - c_i)\,h,\ y_{n-1} + h\sum_{j=1}^{s} (b_j - a_{ij})\,k_j\right). \tag{2.4c}$$

Equation (2.4c) is known as the reflection [14] of (2.3). Formulae (2.3) and (2.4) are equivalent, that is formula (2.3) is symmetric, if

$$1 - c_i = c_{s+1-i}, \qquad 1 \le i \le s,$$
$$a_{ij} + a_{s+1-i,s+1-j} = b_{s+1-j} = b_j, \qquad 1 \le i, j \le s.$$

We now wish to extend this definition to find conditions under which the deferred correction scheme (2.1) is symmetric. The crucial point is that it is not

$$\phi_4(\bar{\eta}) = -\phi_6(\bar{\eta}),$$

which we require to be symmetric (in this case, the usual definition of symmetry would do) but instead we require the deferred correction step

$$\phi_4(\bar{\eta}) = -\phi_6(\eta)$$

to be symmetric. Since the intermediate solution η is generated by a fourth-order Runge-Kutta method, we cannot go from step (2.4b) to (2.4c) in rewriting ϕ_6. It is precisely for this reason that we require a revised definition of symmetry which is appropriate for (2.1). There are several ways in which this revised definition could be formulated. One would be to replace all the k_i's appearing in (2.3) by approximations involving only y_n and y_{n+1}. However, this is only possible when ϕ_4 and ϕ_6 are mono-implicit Runge-Kutta formulae [16]. Clearly, what we need is a revised definition of symmetry which is applicable to (2.1) when ϕ_4, ϕ_6 are arbitrary Runge-Kutta formulae. A suitable definition is the following.

DEFINITION 2.1. Consider the general s-stage Runge-Kutta formula (2.3). Rewrite this in the form

$$\begin{aligned} y_{n+1} &= y_n + h\sum_{i=1}^{s} b_i k_i, \\ k_i &= f\left(x_n + c_i h,\ \frac{y_n + y_{n+1}}{2} + h\sum_{j=1}^{s}\left(a_{ij} - \frac{b_j}{2}\right) k_j\right). \end{aligned} \tag{2.5}$$

This Runge-Kutta formula is defined to be BV-symmetric if

$$c_i = 1 - c_{s+1-i}, \quad b_i = b_{s+1-i}, \qquad 1 \le i \le s, \tag{2.6a}$$

$$a_{ij} - \frac{1}{2}b_j = \frac{1}{2}b_{s+1-j} - a_{s+1-i,s+1-j}, \qquad 1 \le i, j \le s. \tag{2.6b}$$

This definition is more transparent if we express it in terms of the coefficients of the modified Runge-Kutta formula (2.5). Thus, if we rewrite (2.5) in the form

$$\begin{aligned} y_{n+1} &= y_n + h\sum_{i=1}^{s} b_i k_i, \\ k_i &= f\left(x_n + c_i h,\ \frac{y_n + y_{n+1}}{2} + h\sum_{j=1}^{s} \hat{a}_{ij} k_j\right), \end{aligned} \tag{2.7}$$

then the condition for BV-symmetry is (2.6a) and

$$\hat{a}_{ij} = -\hat{a}_{s+1-i,s+1-j}, \qquad 1 \le i, j \le s.$$

We can now give the following lemma.

LEMMA 2.1.

(i) *The deferred correction scheme (2.1) is symmetric if ϕ_4 and ϕ_6 are BV-symmetric.*
(ii) *A symmetric deferred correction scheme satisfies condition (2.2).*
(iii) *The Runge-Kutta scheme (2.5) is BV-symmetric if and only if the underlying formula (2.3) is symmetric.*

PROOF. The proof of (i) is immediate and comes from writing the second of (2.1) as

$$\begin{aligned}\bar{\eta}_{n+1} - \bar{\eta}_n - h\sum_{i=1}^{s} b_i f\left(x_n + c_i h,\ \frac{\bar{\eta}_n + \bar{\eta}_{n+1}}{2} + h\sum_{j=1}^{s}\hat{a}_{ij}k_j\right)& \\ = -\left[\eta_{n+1} - \eta_n - h\sum_{i=1}^{r}\tilde{b}_i f\left(x_n + \tilde{c}_i h,\ \frac{\eta_n + \eta_{n+1}}{2} + h\sum_{j=1}^{r}\tilde{a}_{ij}\tilde{k}_j\right)\right]&.\end{aligned}$$

Replacing h by $-h$, we see that this formula is equal to its reflection and so is symmetric.

To show (ii), we note that

$$\begin{aligned}\psi(\Delta w) &= \phi_4(\Delta w) - \phi_6(\Delta w)\\ &= \frac{\Delta w_{n+1} - \Delta w_n}{h} - \sum_{i=1}^{s} b_i\hat{k}_i - \left[\frac{\Delta w_{n+1} - \Delta w_n}{h} - \sum_{i=1}^{r}\tilde{b}_i\tilde{k}_i\right]\\ &= \sum_{i=1}^{r}\tilde{b}_i f\left(x_n + \tilde{c}_i h,\ \frac{\Delta w_n + \Delta w_{n+1}}{2} + h\sum_{j=1}^{r}\tilde{a}_{ij}\tilde{k}_j\right)\\ &\quad - \sum_{i=1}^{s} b_i f\left(x_n + c_i h,\ \frac{\Delta w_n + \Delta w_{n+1}}{2} + h\sum_{j=1}^{s}\hat{a}_{ij}\hat{k}_j\right)\\ &= \sum_{i=1}^{r}\tilde{b}_i f\left(x_{n+1/2},\ \frac{\Delta w_n + \Delta w_{n+1}}{2}\right)\\ &\quad - \sum_{i=1}^{s} b_i f\left(x_{n+1/2},\ \frac{\Delta w_n + \Delta w_{n+1}}{2}\right) + O\left(h^2\right) = O\left(h^2\right).\end{aligned}$$

Part (iii) is straightforward to verify from the basic definitions. ∎

The important practical implication of this lemma is that if we take two standard symmetric Runge-Kutta formulae ϕ_4 and ϕ_6, of order 4 and 6, respectively, and rewrite them in the modified form (2.5), then the deferred correction scheme (2.1) based on these modified formulae is of order 6. In the next section, we examine this approach in the case where ϕ_4 and ϕ_6 are Lobatto IIIa formulae.

3. STABILITY OF DEFERRED CORRECTION FORMULAE

In previous papers on iterated deferred correction, the stability properties of the schemes presented have been derived essentially from first principles. In this section, we present a more general stability analysis which will allow the stability properties of any deferred correction scheme to be investigated providing that it is based on Runge-Kutta formulae. The deferred correction scheme which we will investigate, and the one which is used to give the numerical results of the next section is

$$\begin{aligned} \phi_4\left(\eta^1\right) &= 0, \\ \phi_4\left(\eta^2\right) &= -\phi_6\left(\eta^1\right), \\ \phi_4\left(\eta^3\right) &= -\phi_6\left(\eta^1\right) - \phi_8\left(\eta^2\right), \\ \phi_4\left(\eta^4\right) &= -\phi_6\left(\eta^1\right) - \phi_8\left(\eta^2\right) - \phi_{10}\left(\eta^3\right). \end{aligned} \tag{3.1}$$

Here, each of the Lobatto Runge-Kutta formulae ϕ_i is written in the boundary value form (2.7) and this ensures that η^i has order $2(1+i)$ for $1 \le i \le 4$. To examine the stability of (3.1), we define the matrix $\hat{A}_s$ as

$$\hat{\mathbf{A}}_s = \begin{pmatrix} 0 & 0 & \dots & 0 \\ \hat{a}_{21} & \hat{a}_{22} & \dots & \hat{a}_{2s} \\ \vdots & \vdots & & \vdots \\ \hat{a}_{s-11} & \hat{a}_{s-12} & \dots & \hat{a}_{s-1s} \\ 0 & 0 & \dots & 0 \end{pmatrix}. \tag{3.2}$$

It is straightforward to show that if we apply (2.1), which forms the first two steps of (3.1), to the scalar test equation $y' = \lambda y$, we obtain

$$y_{n+1}\left(1 - z\hat{b}_s^{\mathrm{T}}\left(I - z\hat{A}_s\right)^{-1} e_2^s\right) - y_n\left(1 + z\hat{b}_s^{\mathrm{T}}\left(I - z\hat{A}_s\right)^{-1} e_1^s\right) = 0, \tag{3.3}$$

where

$$z \equiv h\lambda, \qquad \hat{b}_s = \left(b_1, b_2, \dots, b_s\right)^{\mathrm{T}},$$
$$e_1^s = \left(1, \frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2}, 0\right)^{\mathrm{T}} \quad \text{and} \quad e_2^s = \left(0, \frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2}, 1\right)^{\mathrm{T}}.$$

We note the similarity between the result given by (3.3) and the corresponding one for initial value problems in [14, p. 240]. Having obtained the expression (3.3), it is now straightforward to derive the stability properties of our deferred correction scheme. If we consider (2.1), for example, the first step of this scheme applied to $y' = \lambda y$ gives

$$\left[1 - z\hat{b}_3^{\mathrm{T}}\left(I - z\hat{A}_3\right)^{-1} e_2^3\right]\eta_{n+1} - \left[1 + z\hat{b}_3^{\mathrm{T}}\left(I - z\hat{A}_3\right)^{-1} e_1^3\right]\eta_n = 0. \tag{3.4}$$

Now, applying the correction term, we obtain

$$\begin{aligned} &\left[1 - z\hat{b}_3^{\mathrm{T}}\left(I - z\hat{A}_3\right)^{-1} e_2^3\right]\bar{\eta}_{n+1} - \left[1 + z\hat{b}_3^{\mathrm{T}}\left(I - z\hat{A}_3\right)^{-1} e_1^3\right]\bar{\eta}_n \\ &\quad = -\left[1 - z\hat{b}_4^{\mathrm{T}}\left(I - z\hat{A}_4\right)^{-1} e_2^4\right]\eta_{n+1} + \left[1 + z\hat{b}_4^{\mathrm{T}}\left(I - z\hat{A}_4\right)^{-1} e_1^4\right]\eta_n. \end{aligned} \tag{3.5}$$

Using (3.4) in (3.5), it follows that

$$\begin{aligned} \bar{\eta}_{n+1} = {} & \frac{\left[1 + z\hat{b}_3^{\mathrm{T}}\left(I - z\hat{A}_3\right)^{-1} e_1^3\right]}{\left[1 - z\hat{b}_3^{\mathrm{T}}\left(I - z\hat{A}_3\right)^{-1} e_2^3\right]}\bar{\eta}_n \\ & - \left[\frac{\left[1 - z\hat{b}_4^{\mathrm{T}}\left(I - z\hat{A}_4\right)^{-1} e_2^4\right]\left[1 + z\hat{b}_3^{\mathrm{T}}\left(I - z\hat{A}_3\right)^{-1} e_1^3\right]}{\left[1 - z\hat{b}_3^{\mathrm{T}}\left(I - z\hat{A}_3\right)^{-1} e_2^3\right]\left[1 - z\hat{b}_3^{\mathrm{T}}\left(I - z\hat{A}_3\right)^{-1} e_2^3\right]} - \frac{\left[1 + z\hat{b}_4^{\mathrm{T}}\left(I - z\hat{A}_4\right)^{-1} e_1^4\right]}{\left[1 - z\hat{b}_3^{\mathrm{T}}\left(I - z\hat{A}_3\right)^{-1} e_2^3\right]}\right]\eta_n. \end{aligned}$$

All of these expressions can be expanded using an algebraic package, Mathematica, for example, and from this we can examine the stability properties of (2.1). Using this approach, we have been able to verify all the stability functions that were given in [13] and were derived from first principles.

This general approach is straightforward to extend to allow the stability functions of higher-order deferred correction schemes to be derived. We have done this for (3.1) and have been able to completely analyse its stability properties in this way. In particular, we have been able to show that this scheme is stable in the sense that

$$\lim_{z\to\infty} \frac{\eta_n^4}{\eta_0^1} = 1, \tag{3.6}$$

where η_n^4 denotes that component of η^4 which is evaluated at the n^{th} grid point. This is a much more satisfactory result than is obtained for the corresponding scheme based on MIRK formulae, since in the latter case, the ratio (3.6) becomes unbounded [7].

The conclusion of our stability analysis, therefore, is that a deferred correction scheme of the form (3.1) based on Lobatto IIIa formulae is much more stable than the corresponding one based on MIRK formulae. As a result, we expect the Lobatto formulae to be much more effective on extremely stiff systems. We have developed a general program based on (3.1), and in the next section, we will present some general results to compare this program with TWPBVP from NETLIB which is based on MIRK formulae.

4. NUMERICAL RESULTS

In this section, we present some numerical results to compare our Lobatto code based on (3.1) with an existing deferred correction code. We have run Lobatto on the challenging test set described in [11] and in what follows, we give a representative subset of results to demonstrate the overall performance of our algorithm. The test problems considered are the following.

PROBLEM 1.

$$\varepsilon y'' + \exp(y)y' - \frac{\pi}{2}\sin\left(\frac{\pi x}{2}\right)\exp(2y) = 0, \qquad 0 \le x \le 1, \quad y(0) = y(1) = 0.$$

This has a boundary layer at $x = 0$.

PROBLEM 2.

$$\varepsilon y'' + (y')^2 = 1, \qquad 0 \le x \le 1,$$
$$y(0) = 1 + \varepsilon \ln\cosh\left(\frac{-0.745}{\varepsilon}\right), \qquad y(1) = 1 + \varepsilon \ln\cosh\left(\frac{0.255}{\varepsilon}\right).$$

The true solution is $y(x) = \alpha + \varepsilon \ln\cosh(x - \beta)/\varepsilon$.

As $\varepsilon \to 0$, $y(x) \to \alpha + |x - \beta|$ and so, for small ε, the solution has a corner layer at $x = \beta$.

PROBLEM 3. Troesch's equation

$$y'' = \mu \sinh \mu y, \qquad y(0) = 0, \quad y(1) = 1.$$

This problem, which has been considered as a test problem by many authors, has a sharp boundary layer at $x = 1$.

PROBLEM 4. Lagerstrom-Cole equation

$$\varepsilon y'' + yy' - y = 0, \qquad 0 \le x \le 1, \quad y(0) = -\frac{7}{6}, \quad y(1) = \frac{3}{2}.$$

This problem has a shock layer at $x = 1/3$.

PROBLEM 5.

$$y'''' = R(y'y'' - yy'''), \qquad y(0) = y'(0) = 0, \quad y(1) = 1, \quad y'(1) = 0.$$

This arises from fluid injection through one side of a long vertical channel.

For a discussion of these problems and for the original references, the reader is referred to [11].

We will use the test problems just described to compare the performance of our Lobatto code, which is based on (3.1), with that of the deferred correction code TWPBVP which appears in NETLIB. TWPBVP is based on mono-implicit Runge-Kutta formulae. An extensive comparison between TWPBVP and COLSYS already exists [11] and so, we do not feel it is necessary to give any further results for COLSYS. Instead, what we wish to do in this section is to show that we obtain a performance which is in line with that predicted by the theory and in particular, to show that Lobatto formulae are superior for excessively stiff problems. Our method of testing has been fully described in [11] and we will not elaborate on it in this section. However, we do need to describe briefly the way in which our Lobatto code is implemented. This code is based on four Lobatto IIIa formulae of orders 4, 6, 8, and 10 which are implemented in a deferred correction framework as described by (3.1). The equations defining the deferred corrections, which are implicit, are solved to machine accuracy using a Newton iteration scheme.

If this iteration scheme fails to converge for less than four mesh points, then the grid is refined (four extra equally spaced points are added) only at these mesh points. If nonconvergence is obtained at more than three mesh points, then the grid is halved. Apart from these changes in dealing with the implicit rather than explicit deferred corrections, the implementation of the Lobatto code is as described in [11].

Table 1. Results for Problem 1.

		LOBATTO		TWPBVP	
ε	Tol	Max Mesh	Time	Max Mesh	Time
10^{-1}	10^{-4}	11	0.05	11	0.03
	10^{-8}	17	0.11	23	0.05
10^{-2}	10^{-4}	21	0.15	33	0.16
	10^{-8}	46	0.32	54	0.21
10^{-3}	10^{-4}	100	0.55	87	0.49
	10^{-8}	100	1.04	113	0.64
10^{-4}	10^{-4}	340	3.06	289	2.36
	10^{-8}	340	3.40	303	2.91
10^{-5}	10^{-4}	1319	10.61	4219	35.49
	10^{-8}	1319	11.28	4219	39.41

Before considering our numerical results, we first summarize briefly what the theory predicts will happen. We would expect from the analysis of the previous section that for ε reasonably large (or for μ and R reasonably small), when the problem is not very stiff, that TWPBVP would be superior since the excellent stability of Lobatto is not important and TWPBVP has the advantage that the deferred correction is explicit and so cheap. However, as the parameter ε is decreased, and the problem becomes increasingly stiff, then good stability is crucial and we would expect LOBATTO, even though its deferred correction terms are expensive to compute, to be more efficient. An examination of Tables 1–5 shows that this is exactly what happens on all problems apart from on Problem 2 where the results for TWPBVP and LOBATTO are comparable. This confirms our expectations that the highly stable LOBATTO code is likely to be superior on very stiff problems. This in turn suggests two exciting possibilities which we now mention. First, since an expensive part of the deferred correction code is in actually computing the deferred corrections, and since these evaluations can all be done in parallel, we would expect very large speed ups if this code was implemented on a parallel computer. Second, it has been shown in [12] that it is often very efficient to solve extremely stiff problems using an automatic continuation algorithm. Since we have now developed a code which has the excellent stability to deal with extremely stiff problems, we plan to investigate this possibility. Preliminary experience with a nonlinear continuation algorithm based on Lobatto formulae indicates that it is extremely effective for dealing with excessively stiff problems and we hope to report on this, as well as on a parallel code, in forthcoming papers.

Table 2. Results for Problem 2.
Max Mesh = maximum number of mesh points used.

		LOBATTO		TWPBVP	
ε	Tol	Max Mesh	Time	Max Mesh	Time
$\frac{1}{6}$	10^{-4}	11	0.05	10	0.03
	10^{-8}	20	0.12	36	0.10
$\frac{1}{36}$	10^{-4}	21	0.16	21	0.13
	10^{-8}	33	0.30	42	0.18
$\frac{1}{216}$	10^{-4}	41	0.36	55	0.32
	10^{-8}	84	0.68	127	0.68

Table 3. Results for Problem 3.

		LOBATTO		TWPBVP	
μ	Tol	Max Mesh	Time	Max Mesh	Time
10	10^{-4}	30	0.22	47	0.23
	10^{-8}	51	0.52	67	0.44
20	10^{-4}	68	0.53	66	0.50
	10^{-8}	135	1.26	99	0.77
30	10^{-4}	115	1.22	527	13.50
	10^{-8}	115	1.71	1359	18.19

Table 4. Results for Problem 4.

		LOBATTO		TWPBVP	
ε	Tol	Max Mesh	Time	Max Mesh	Time
$\frac{1}{5}$	10^{-4}	11	0.03	10	0.02
	10^{-8}	11	0.06	20	0.05
$\frac{1}{25}$	10^{-4}	17	0.12	16	0.06
	10^{-8}	37	0.28	90	0.29
$\frac{1}{125}$	10^{-4}	825	4.86	721	4.42
	10^{-8}	825	5.42	1873	11.65

Table 5. Results for Problem 5.

		LOBATTO		TWPBVP	
μ	Tol	Max Mesh	Time	Max Mesh	Time
10	10^{-4}	11	0.10	20	0.15
	10^{-8}	11	0.16	20	0.12
100	10^{-4}	21	0.19	34	0.29
	10^{-8}	22	0.58	34	0.46
1000	10^{-4}	30	0.30	34	0.33
	10^{-8}	30	1.13	72	0.57
10000	10^{-4}	30	0.42	47	0.32
	10^{-8}	30	0.65	79	0.83
100000	10^{-4}	39	0.93	145	2.07
	10^{-8}	53	1.49	185	2.09

REFERENCES

1. L. Fox, Some improvements in the use of relaxation methods for the solution of ordinary and partial differential equations, *Proc. Roy. Soc. London Ser. A* **190**, 31–59 (1947).
2. V. Pereyra, On improving an approximate solution of a functional equation by deferred corrections, *Numer. Math.* **8**, 376–391 (1966).
3. H.J. Stetter, The defect correction principle and discretization methods, *Numer. Math.* **29**, 425–443 (1978).
4. B. Lindberg, Error estimation and iterative improvement for discretization algorithms, *BIT* **20**, 486–500 (1980).
5. R.D. Skeel, A theoretical framework for proving accuracy results for deferred corrections, *SIAM J. Numer. Anal.* **19**, 171–196 (1982).
6. J.R. Cash, On the numerical integration of nonlinear two-point boundary value problems using iterated deferred corrections, Part 1: A survey and comparison of some one-step formulae, *Computers Math. Applic.* **12A** (10), 1029–1048 (1986).
7. J.R. Cash, Part 2: The development and analysis of highly stable deferred correction formulae, *SIAM J. Numer. Anal.* **21**, 862–882 (1988).
8. H.H.M. Silva and J.R. Cash, Iterated deferred correction for linear two-point boundary value problems, *Comp. Appl. Math.* **15**, 55–75 (1996).
9. M. Lentini and V. Pereyra, A variable order finite difference method for nonlinear multipoint boundary value problems, *M. Comp.* **28**, 981–1004 (1974).
10. J.W. Daniel and A.J. Martin, Numerov's method with deferred corrections for two-point boundary value problems, *SIAM J. Numer. Anal.* **14**, 1033–1050 (1977).
11. J.R. Cash and M.H. Wright, A deferred correction method for nonlinear two-point boundary value problems. Implementation and numerical evaluation, *SIAM J. Sci. Stat. Comput.* **12**, 971–989 (1991).
12. R.W. Wright, An automatic continuation strategy for the solution of stiff two-point boundary value problems, Ph.D. Thesis, Imperial College, University of London, (1995).
13. H.H.M. Silva, Iterated deferred correction schemes for the numerical solution of two-point boundary value problems, Ph.D. Thesis, Imperial College, University of London, (1994).
14. J.C. Butcher, *The Numerical Analysis of Ordinary Differential Equations*, J. Wiley, (1987).
15. J.D. Lambert, *Numerical Methods for Ordinary Differential Equations*, Wiley, (1991).
16. J.R. Cash and A. Singhal, High order methods for the numerical solution of two-point boundary value problems, *BIT* **22**, 184–199 (1982).
17. U. Ascher, J. Christiansen and R.D. Russel, Collocation software for boundary value ODEs, *ACM Trans. Math. Software*, 209–222 (1981).
18. R.D. Skeel, Thirteen ways to estimate global error, *Numer. Math.* **48**, 1–20 (1986).