European Society of Computational Methods in Sciences and Engineering (ESCMSE)

Journal of Numerical Analysis, Industrial and Applied Mathematics (JNAIAM)
vol. 1, no. 1, 2006, pp. 13-25
ISSN 1790–8140

Lobatto-Obrechkoff Formulae for 2^{nd} Order Two-Point Boundary Value Problems

S. D. Capper, J. R. Cash and D. R. Moore[1]

Department of Mathematics, Imperial College of Science, Technology and Medicine, London, SW7 2AZ, United Kingdom

Received 27 November, 2005; accepted in revised form 18 January, 2006

Abstract: A substantial increase in efficiency may be obtained by numerical integration methods which take advantage of the special second order forms $y'' = f(x,\ y)$ or $y'' = f(x,\ y,\ y')$ in systems of second order two-point boundary value problems, while retaining the MIRK structure. In particular, for these special second order equations, we derive high order methods which require considerably fewer function evaluations than are required by methods intended for general first order systems. Methods based on a Lobatto-MIRK formula for finding y'_n and an Obrechkoff type formula for finding y_n using the same values of y'' are derived and these methods are of sixth and of eighth order accuracy.

Keywords: MIRK, Runge Kutta, ODE, Boundary Value Problem, Lobatto Formulae, Obrechkoff Formulae

Mathematics Subject Classification: 65L10, 65L12

1 Introduction

Many systems of two point boundary value problems occur in one of the two special forms:

$$\frac{\mathrm{d}^2y}{\mathrm{d}x^2} = f(x,\ y) \qquad \text{or} \qquad \frac{\mathrm{d}^2y}{\mathrm{d}x^2} = f\left(x,\ y,\ \frac{\mathrm{d}y}{\mathrm{d}x}\right). \tag{1}$$

Differential equations of this type are typically derived from physical systems having either spatial invariance or symmetry. The standard technique for solving equation (1) is to expand it into a larger first order system and then to solve it using an integration method intended for general first order systems. This approach does have serious drawbacks in that the conversion of (1) to a first order system doubles the size of the system to be solved and typically increases the cost of the linear algebra by a factor of eight! There do, however, exist codes which can deal with (1) directly, i.e. without reducing the equations to first order form ([1, 2]).

[1]Corresponding author: e-mail: dan.moore@imperial.ac.uk, Phone: (+44) 2075948510, Fax: (+44) 2075948517

In what follows we derive sixth and eighth order accurate pairs of finite difference formulae for y_n and y'_n. Accurate mesh interval interior values of y and y' are found from interpolants [9]. The formulae pairs derived in this paper require fewer function evaluations per mesh interval than MIRK formulae of the same order of accuracy. Because of this, the resulting Jacobians for solving the non-linear equations using Newton iteration require fewer matrix multiplies to calculate.

Extensive numerical tests, which we report on in this paper, show that these formulae achieve greater accuracy for most problems than the standard MIRK formulae and require significantly less computational effort. We suggest that these formulae should be incorporated into any general purpose finite difference two-point boundary value package as an option for this class of problems, especially for problems where the right hand side of the differential equation system is expensive to calculate.

2 The Integration Schemes

Lobatto integration formulae are useful templates for the calculation of high order MIRK formulae. As they are the most accurate formulae specifically including the end points of the integration interval, they minimize the number of internal values necessary to achieve a given order of accuracy. They transform the determination of a high order MIRK formula from an initial value problem (*find* a_{ij}, b_i, c_i *in the Runge-Kutta tableaux* [3] (page 340) *satisfying the implicit Runge Kutta order conditions*) to a boundary value problem (*find* a_{ij}, *given* b_i *and some* c_i). The determination of the a_{ij} and of the remaining c_i becomes an exercise in creating a succession of more accurate interpolants designed to give the most accurate possible values for y (and y') required by the Lobatto formula. Cash ([6]) exploited this strategy when he used the 5-point 8^{th} order Lobatto integration formula as the final stage of his 8^{th} order MIRK formula.

In the next two subsections we follow this methodology to construct 6^{th} and 8^{th} order accurate finite difference formulae for y'_n and y'_{n+1} using the appropriate Lobatto formulae for each order. We then construct finite difference formulae for finding y_n and y_{n+1} using y'_n and y'_{n+1} and the same interior values of $y''(x)$, $x_n \le x \le x_{n+1}$ using the Lobatto based finite difference formulae for y'_n and y'_{n+1}. We term these new formulae Lobatto-Obrechkoff (or LOB) formulae.

2.1 A Sixth Order Method

In this subsection we derive MIRK formulae specifically designed for the solution of second order systems. A sixth order MIRK formula may be derived for y' using values of y'' at the sixth order Lobatto integration formula ordinates and using the Lobatto weights:

$$y'_{n+1} = y'_n + \frac{h_n}{12}\left(\left(y''_{n+1} + y''_n\right) + 5\left(y''_{n+\frac{1}{2}+\frac{\sqrt{5}}{10}} + y''_{n+\frac{1}{2}-\frac{\sqrt{5}}{10}}\right)\right). \tag{2a}$$

A companion Obrechkoff type formula using the same values of y'' may be derived:

$$y_{n+1} = y_n + \frac{h_n}{2}\left(y'_{n+1} + y'_n\right) - \frac{h_n^2}{24}\left(\left(y''_{n+1} - y''_n\right) + \sqrt{5}\left(y''_{n+\frac{1}{2}+\frac{\sqrt{5}}{10}} - y''_{n+\frac{1}{2}-\frac{\sqrt{5}}{10}}\right)\right), \tag{2b}$$

Equations (2a) and (2b) are solved for $\left\{y_n,\ y_{n+1},\ y'_n,\ y'_{n+1}\right\}$ using the same modified Newton method typically used to solve MIRK integration formulae. Before we can solve these equations, we require values of $y\left(x_{n+\frac{1}{2}\pm\frac{\sqrt{5}}{10}}\right)$; and for systems of the form $y'' = f(x,\ y,\ y')$, the additional values of $y'\left(x_{n+\frac{1}{2}\pm\frac{\sqrt{5}}{10}}\right)$. We use the superscript notation $y^{(a)}_{n+\frac{1}{2}+\omega}$ to denote a point at $x = x_n + h_n(\frac{1}{2}+\omega)$ with final $O(h_n^a)$ truncation error (taking into account the truncation errors of the intermediate points). Fitting a fifth order polynomial $y^{(6)}_{n+\omega} = y(x_n + \omega h_n) + O(h_n^6)$ through the points $\{y_n,\ y'_n,\ y''_n,\ y_{n+1},\ y'_{n+1},\ y''_{n+1}\}$ gives:

$$y^{(6)}_{n+\frac{1}{2}\pm\frac{\sqrt{5}}{10}} = \frac{125 \pm 41\sqrt{5}}{250} y_{n+1} + \frac{125 \mp 41\sqrt{5}}{250} y_n - h_n \left(\frac{15 \pm 4\sqrt{5}}{125} y'_{n+1} - \frac{15 \mp 4\sqrt{5}}{125} y'_n \right) + h_n^2 \left(\frac{5 \pm \sqrt{5}}{500} y''_{n+1} + \frac{5 \mp \sqrt{5}}{500} y''_n \right), \tag{3}$$

where:

$$y^{(6)}_{n+\frac{1}{2}\pm\frac{\sqrt{5}}{10}} - y\left(x_n + h_n \left(\frac{1}{2} \pm \frac{\sqrt{5}}{10} \right) \right) = \frac{h_n^6}{90000} \frac{\mathrm{d}^6 y}{\mathrm{d}x^6} + O(h_n^7). \tag{4}$$

For problems of the form $y'' = f(x,\ y,\ y')$, we can obtain the required values of y' at $x = x_n + h_n \left(\frac{1}{2} \pm \frac{\sqrt{5}}{10} \right)$ from $\frac{\mathrm{d}}{\mathrm{d}\omega} y^{(6)}_{n+\omega} = y'^{(5)}_{n+\omega}$:

$$y'^{(5)}_{n+\frac{1}{2}\pm\frac{\sqrt{5}}{10}} = \frac{6(y_{n+1} - y_n)}{5\, h_n} - \frac{5 \mp 7\sqrt{5}}{50} y'_{n+1} - \frac{5 \pm 7\sqrt{5}}{50} y'_n \mp \frac{h_n \sqrt{5}}{50} \left(y''_{n+1} + y''_n \right), \tag{5}$$

where:

$$y'^{(5)}_{n+\frac{1}{2}\pm\frac{\sqrt{5}}{10}} - y'\left(x_n + h_n \left(\frac{1}{2} \pm \frac{\sqrt{5}}{10} \right) \right) = \mp \frac{h_n^5 \sqrt{5}}{30000} \frac{\mathrm{d}^6 y}{\mathrm{d}x^6} + O(h_n^6). \tag{6}$$

We are now able to compute $y''_{n+\frac{1}{2}\pm\frac{\sqrt{5}}{10}}$:

$$y''_{n+\frac{1}{2}\pm\frac{\sqrt{5}}{10}} = \begin{cases} f\left(x_{n+\frac{1}{2}\pm\frac{\sqrt{5}}{10}},\ y^{(6)}_{n+\frac{1}{2}\pm\frac{\sqrt{5}}{10}} \right) & \text{for } y'' = f(x,\ y) \text{ problems.} \\ f\left(x_{n+\frac{1}{2}\pm\frac{\sqrt{5}}{10}},\ y^{(6)}_{n+\frac{1}{2}\pm\frac{\sqrt{5}}{10}},\ y'^{(5)}_{n+\frac{1}{2}\pm\frac{\sqrt{5}}{10}} \right) & \text{for } y'' = f(x,\ y,\ y') \text{ problems.} \end{cases} \tag{7}$$

For both classes of problem $y'' = f(x,\ y)$ and $y'' = f(x,\ y,\ y')$, the interior points $\left\{ y''_{n+\frac{1}{2}\pm\frac{\sqrt{5}}{10}} \right\}$ are entered into (2), and the resulting equations are solved for y_n and y_{n+1}.

For problems of the form $y'' = f(x,\ y,\ y')$, we can track the error contributions from our interior points to compute the local truncation errors of the sixth order scheme:

$$y'_{n+1} - y'_n - \frac{h_n}{12} \left(\left(y''_{n+1} + y''_n \right) + 5 \left(y''_{n+\frac{1}{2}+\frac{\sqrt{5}}{10}} + y''_{n+\frac{1}{2}-\frac{\sqrt{5}}{10}} \right) \right) = \frac{h_n^7}{504000} \left[2 \frac{\mathrm{d}^7 y}{\mathrm{d}x^7} - 7 \frac{\partial f}{\partial y'} \frac{\mathrm{d}^6 y}{\mathrm{d}x^6} \right] + O(h_n^8), \tag{8a}$$

and,

$$y_{n+1} - y_n - \frac{h_n}{2} \left(y'_{n+1} + y'_n \right) + \frac{h_n^2}{24} \left(\left(y''_{n+1} - y''_n \right) + \sqrt{5} \left(y''_{n+\frac{1}{2}+\frac{\sqrt{5}}{10}} - y''_{n+\frac{1}{2}-\frac{\sqrt{5}}{10}} \right) \right) = \frac{h_n^7}{1512000} \left[\left(21 \frac{\mathrm{d}}{\mathrm{d}x} \frac{\partial f}{\partial y'} - 14 \frac{\partial f}{\partial y} \right) \frac{\mathrm{d}^6 y}{\mathrm{d}x^6} + \frac{\partial f}{\partial y'} \frac{\mathrm{d}^7 y}{\mathrm{d}x^7} - \frac{\mathrm{d}^8 y}{\mathrm{d}x^8} \right] + O(h_n^8), \tag{8b}$$

we thus obtain sixth order accuracy when solving BVPs with this scheme. For the simpler class of problem $y'' = f(x,\ y)$, one can calculate the local truncation error of the finite difference scheme by substituting $\frac{\partial f}{\partial y'} = 0$, into (8).

It is worth noting that our sixth order integration scheme requires one less right hand side evaluation per mesh interval than the sixth order MIRK formula, for first order systems, of Cash and Singhal (1982), ([10]). In addition, the analytic expression for the Jacobian required for solving the resulting non-linear system is composed of fewer matrix multiplications.

2.2 An Eighth Order Method

If we consider a 5 point Lobatto formula as a template for an eighth order accurate finite difference formula for y' we get $\left(\text{where } \alpha = \sqrt{\frac{3}{28}}\right)$:

$$y'_{n+1} = y'_n + \frac{h_n}{180}\left(9\left(y''_{n+1} + y''_n\right) + 49\left(y''_{n+\frac{1}{2}+\alpha} + y''_{n+\frac{1}{2}-\alpha}\right) + 64\, y''_{n+\frac{1}{2}}\right). \tag{9a}$$

The companion Obrechkoff like finite difference formula integrating y to eighth order accuracy is:

$$y_{n+1} = y_n + \frac{h_n}{2}\left(y'_{n+1} + y'_n\right) - \frac{h_n^2}{360}\left(9\left(y''_{n+1} - y''_n\right) + 7\sqrt{21}\left(y''_{n+\frac{1}{2}+\alpha} - y''_{n+\frac{1}{2}-\alpha}\right)\right), \tag{9b}$$

These formulae for y_n and y'_n can be shown to be eighth order accurate if the interior values of y'' can be computed to sufficient accuracy. We note also that equation (9a) is the basis of the eighth order MIRK formulae of Cash (2000) [6]. To generate approximations to $y_{n+\frac{1}{2}\pm\alpha}$, we use the seventh order interpolation formulae of [9] evaluated at the Lobatto interior points.

$$y^{(6)}_{n+\frac{1}{2}} = \frac{1}{2}\left(y_{n+1} + y_n\right) - \frac{5h_n}{32}\left(y'_{n+1} - y'_n\right) + \frac{h_n^2}{64}\left(y''_{n+1} + y''_n\right), \tag{10a}$$

$$y''^{(6)}_{n+\frac{1}{2}} = f\left(x_n + \frac{h_n}{2},\ y^{(6)}_{n+\frac{1}{2}}\right), \tag{10b}$$

$$y^{(7)}_{n+\frac{1}{2}\pm\alpha} = a_7^{\pm}\, y_{n+1} + a_7^{\mp}\, y_n - h_n\left(b_7^{\pm} y'_{n+1} - b_7^{\mp} y'_n\right) + h_n^2\left(c_7^{\pm} y''_{n+1} + d_7 y''^{(6)}_{n+\frac{1}{2}} + c_7^{\mp} y''_n\right), \tag{10c}$$

$$y''^{(7)}_{n+\frac{1}{2}\pm\alpha} = f\left(x_n + h_n\left(\frac{1}{2} \pm \alpha\right),\ y^{(7)}_{n+\frac{1}{2}\pm\alpha}\right). \tag{10d}$$

(10e)

The coefficients $\{a_7^{\pm},\ b_7^{\pm},\ c_7^{\pm},\ d_7\}$ are taken from the seventh order accurate interpolant described in [9] *(formulae 3.1 and 3.4)* evaluated at the two points $\omega^{\pm} = \frac{1}{2} \pm \alpha$. These numbers are expressed in subtraction free form to minimize rounding error when calculating them numerically:

$$\begin{aligned}
a_7^{+} &= A_7(\omega^{+}) = \frac{343 + 69\sqrt{21}}{686}, & a_7^{-} &= A_7(\omega^{-}) = \frac{1262}{49(343 + 69\sqrt{21})},\\
b_7^{+} &= B_7(\omega^{+}) = \frac{5\sqrt{21} + 24}{343}, & b_7^{-} &= B_7(\omega^{-}) = \frac{51}{343(5\sqrt{21} + 24)},\\
c_7^{+} &= C_7(\omega^{+}) = \frac{13 + 3\sqrt{21}}{4116}, & c_7^{-} &= C_7(\omega^{-}) = \frac{-5}{1029(13 + 3\sqrt{21})},\\
d_7 &= D_7(\omega^{\pm}) = \frac{-8}{1029}.
\end{aligned} \tag{11}$$

The seventh order accurate interpolant [9] using second derivative data evaluated at the interior Lobatto points supplies the value of $y^{(7)}_{n+\frac{1}{2}\pm\alpha}$ which is then used to calculate $y''^{(7)}_{n+\frac{1}{2}\pm\alpha}$ to sufficient accuracy to preserve the overall order of the Lobatto formula (9). However, a more accurate value of $y_{n+\frac{1}{2}}$ than that available from the Hermite-Birkhoff interpolant using just the data set $\{y_n,\ y'_n,\ y''_n,\ y_{n+1},\ y'_{n+1},\ y''_{n+1}\}$, (10a), is required to compute $y''_{n+\frac{1}{2}}$ to a sufficient accuracy to use in (9a). The $O(h_n^6)$ value (10a) can, however, be used to calculate a more accurate value of $y_{n+\frac{1}{2}}$, labelled $y^{(8)}_{n+\frac{1}{2}}$:

$$y^{(8)}_{n+\frac{1}{2}} = \frac{1}{2}\left(y_{n+1} + y_n\right) - \frac{3h_n}{32}\left(y'_{n+1} - y'_n\right) + \frac{h_n^2}{192}\left(y''_{n+1} - 8y''^{(6)}_{n+\frac{1}{2}} + y''_n\right), \tag{12}$$

$$y''^{(8)}_{n+\frac{1}{2}} = f\left(x_n + \frac{h_n}{2},\ y^{(8)}_{n+\frac{1}{2}}\right). \tag{13}$$

The interior points $\left\{y''^{(7)}_{n+\frac{1}{2}\pm\alpha}, y''^{(8)}_{n+\frac{1}{2}}\right\}$ are entered into (9), and the resulting equations are solved for y_n and y_{n+1}.

We now consider the solution of problems of the form $y'' = f(x,\ y,\ y')$. As we are tracking errors from both y and y' expressions, our superscript notation used to denote the error at a point, $y''_{n+\frac{1}{2}\pm\alpha}$, needs adapting in order to better reflect the error contributions from both the intermediate y and y' values. We use the following notation:

$$y''^{(r,s)}_{n+\frac{1}{2}\pm\alpha} = f\left(x_n,\ y^{(r)}_{n+\frac{1}{2}\pm\alpha},\ y'^{(s)}_{n+\frac{1}{2}\pm\alpha}\right) = y''\left(x_n + h_n\left(\frac{1}{2}\pm\alpha\right)\right) + h_n^r\frac{\partial f}{\partial y}(\cdots)R + h_n^s\frac{\partial f}{\partial y'}(\cdots)S, \tag{14}$$

to express the truncation error orders from both the y and y' values.

For $y'' = f(x,\ y,\ y')$ problems, we compute the following interior points:

$$y^{(6)}_{n+\frac{1}{2}} = \frac{1}{2}\left(y_{n+1} + y_n\right) - \frac{5h_n}{32}\left(y'_{n+1} - y'_n\right) + \frac{h_n^2}{64}\left(y''_{n+1} + y''_n\right), \tag{15a}$$

$$y'^{(6)}_{n+\frac{1}{2}} = \frac{15}{8h_n}\left(y_{n+1} - y_n\right) - \frac{7}{16}\left(y'_{n+1} + y'_n\right) + \frac{h_n}{32}\left(y''_{n+1} - y''_n\right), \tag{15b}$$

$$y''^{(6,6)}_{n+\frac{1}{2}} = f\left(x_n + \frac{h_n}{2},\ y^{(6)}_{n+\frac{1}{2}},\ y'^{(6)}_{n+\frac{1}{2}}\right), \tag{15c}$$

$$y^{(7)}_{n+\frac{1}{2}\pm\alpha} = a_7^{\pm}\, y_{n+1} + a_7^{\mp}\, y_n - h_n\left(b_7^{\pm}y'_{n+1} - b_7^{\mp}y'_n\right) + h_n^2\left(c_7^{\pm}y''_{n+1} + d_7 y''^{(6,6)}_{n+\frac{1}{2}} + c_7^{\mp}y''_n\right), \tag{15d}$$

$$y'^{(6)}_{n+\frac{1}{2}\pm\alpha} = \frac{a'_7}{h_n}\left(y_{n+1} - y_n\right) + b'^{\pm}_7 y'_{n+1} + b'^{\mp}_7 y'_n - h_n\left(c'^{\pm}_7 y''_{n+1} \mp d'_7 y''^{(6,6)}_{n+\frac{1}{2}} - c'^{\mp}_7 y''_n\right) \tag{15e}$$

$$y''^{(7,6)}_{n+\frac{1}{2}\pm\alpha} = f\left(x_n + h_n\left(\frac{1}{2}\pm\alpha\right),\ y^{(7)}_{n+\frac{1}{2}\pm\alpha},\ y'^{(6)}_{n+\frac{1}{2}\pm\alpha}\right), \tag{15f}$$

$$y^{(8)}_{n+\frac{1}{2}} = \frac{1}{2}\left(y_{n+1} + y_n\right) - \frac{3h_n}{32}\left(y'_{n+1} - y'_n\right) + \frac{h_n^2}{192}\left(y''_{n+1} - 8y''^{(6,6)}_{n+\frac{1}{2}} + y''_n\right), \tag{15g}$$

$$y''^{(8,6)}_{n+\frac{1}{2}} = f\left(x_n + \frac{h_n}{2},\ y^{(8)}_{n+\frac{1}{2}},\ y'^{(6)}_{n+\frac{1}{2}}\right), \tag{15h}$$

(15i)

where,

$$a'_7 = \left.\frac{\mathrm{d}A_7(\omega)}{\mathrm{d}\omega}\right|_{\omega^\pm} = \frac{30}{49},$$
$$b'^{+}_7 = \left.\frac{\mathrm{d}B_7(\omega)}{\mathrm{d}\omega}\right|_{\omega^+} = \frac{133+39\sqrt{21}}{686} \qquad b'^{-}_7 = \left.\frac{\mathrm{d}B_7(\omega)}{\mathrm{d}\omega}\right|_{\omega^-} = \frac{-1018}{49(133+39\sqrt{21})},$$
$$c'^{+}_7 = \left.\frac{\mathrm{d}C_7(\omega)}{\mathrm{d}\omega}\right|_{\omega^+} = \frac{14+3\sqrt{21}}{686}, \qquad c'^{-}_7 = \left.\frac{\mathrm{d}C_7(\omega)}{\mathrm{d}\omega}\right|_{\omega^-} = \frac{1}{98(14+3\sqrt{21})},$$
$$d'_7 = \left.\frac{\mathrm{d}D_7(\omega)}{\mathrm{d}\omega}\right|_{\omega^+} = \frac{8\sqrt{21}}{343}. \tag{16}$$

The interior points $\left\{y''^{(7,6)}_{n+\frac{1}{2}\pm\alpha},\ y''^{(8,6)}_{n+\frac{1}{2}}\right\}$ are entered into (9), and the resulting equations are solved for y_n and y_{n+1}.

Although the interpolated values of y' are sometimes less accurate than those of y, at the interior Lobatto points, these errors cancel in both (9a) and (9b) and the resulting scheme is still eighth order.

Overall, these formulae require just four interior evaluations of the differential equation per mesh interval. The eighth order MIRK formulae of Cash and Singhal (1982) [10] and of Cash (2000) [6] for two point boundary value problems of the form $y' = f(x,\ y)$ require seven interior evaluations per mesh interval.

Economies using these new formulae in place of existing MIRK 8 formulae for problems of the form $y'' = f(x,\ y)$ are achieved in two ways: (i) fewer function evaluations are required per mesh point and (ii) fewer matrix multiplications are required to construct the Jacobians necessary to solve the non-linear system of the MIRK formulae.

3 Implementation

The application of these finite difference formulae to the solution of two-point boundary value problems involves several separate steps. Amongst these are choice of a mesh of independent variable values covering the domain of the problem and the choice of the method for solving the finite difference equations approximating the differential equation across each interior mesh interval. A meaningful discussion of the best mesh selection strategy is beyond the scope of this paper, but it is an essential part of any useful numerical package for solving two point boundary value problems intended for general use.

For the purposes of isolating the questions of the accuracy and the efficiency of these LOB formulae in comparison with MIRK formulae of the same order of accuracy from issues of mesh selection, we have chosen to use a uniform mesh covering the domain of each test problem. We use a uniform mesh of 1025 points for the accuracy comparisons and we use a mesh of 4096 points for the timing comparisons presented below. Where a test problem does not have a known analytic solution (problems 19, 22-32), we generate a numerical solution on a finer uniform mesh which we are confident is of the required accuracy. Higher order MIRK methods [5] have been utilised to verify this assertion.

Newton's method is most commonly used to solve the finite difference equations to determine the values of y and y' at each mesh point x_n [1]. This method requires the calculation of the Jacobian of the full system of finite difference equations and boundary conditions with respect to all of the y_n and y'_n. Both the LOB methods presented here and the MIRK methods we are comparing them with create a highly structured Jacobian that is tightly banded. The finite difference equations for each mesh interval $\{x_n, x_{n+1}\}$ depend only upon $\{y_n, y'_n, y_{n+1}, y'_{n+1}\}$.

However, the direct calculation of the local Jacobian of the finite difference equations for each mesh interval can still require many operations [10]. These appear to make complicated high order MIRK and LOB methods unattractive (and difficult to code).

Four strategies have evolved to deal with these difficulties; (i) Direct calculation [8], (ii) Economical approximation of the Jacobian [10], (iii) Numerical differences approximating the Jacobian [13], (iv) Deferred correction mode with a low order MIRK formula with a simple Jacobian calculation being solved at each stage and using the high order formulae for the deferred correction [11].

The LOB formulae presented here appear to be superior to MIRK formulae of the same order of accuracy for all of the above strategies. Because they require fewer stages per mesh interval, they require fewer matrix multiplications to calculate the Jacobian directly or to calculate an economical approximation as in strategies (i) and (ii). They require fewer function evaluations overall per mesh interval and hence should be more efficient if either strategy (iii) or (iv) is adopted. The advantage the LOB formulae should have over the MIRK formulae should become more marked the more expensive the right hand side of the differential equations becomes.

Actual numerical comparisons on the model set of problems [12] bear out these predictions. For both the accuracy and timing comparisons described below, we have adopted strategy (iii), although we have tested the LOB formulae in strategies (i) and (iv) and found the expected efficiencies there as well. The choice of these strategies affects only the speed of execution, not the final accuracy achieved by solving the finite difference equations.

A Fortran 95 code *NewNRK* [4], solved the test problems using either the LOB or the MIRK formulae. The finite difference equations are solved using Newton's method and the required Jacobians are generated by numerical differences (strategy (iii) described above).

The times taken to solve 31 of the 32 test problems with uniform meshes of 4096 points using both MIRK and LOB methods of orders 6 & 8 can be found tabulated in table 1, and plotted in figure 1. For some problems (e.g. problems 1 & 24) we get an 8^{th} order accurate solution to a test problem using the LOB 8 method in roughly the same amount of time required to obtain one using MIRK 6!

4 Accuracy

In [12] Cash & Wright assembled 32 test problems to assist in evaluating the accuracy and efficiency of various two-point boundary value problem solving codes. Thirty one of these test problems take the form $y'' = f(x,\ y)$ or $y'' = f(x,\ y,\ y')$; of these, 21 have known analytic solutions. Most of these problems contain a parameter, ϵ, that as $\epsilon \to 0$ a singular perturbation problem is created [2]. Very small values of ϵ create problems that are stiff with sharp boundary layers or interior regions where the solution changes rapidly. As we are going to compare our new formulae with the MIRK formulae intended for non-stiff problems, we will choose values of ϵ for each test problem that make it at worst only mildly stiff. The actual values of ϵ used for each problem are set out in column 2 of table 1.

Both L_∞ and L_2 norms of the errors for each test problem were calculated both for y_n and y'_n. The L_2 error norm data for y & y' computed using MIRK 6 & 8 (Cash 2000) and LOB 6 & 8 can be found plotted in figures 2, 3, 4 & 5. One would expect the accuracy of the LOB methods to be greater than that of the MIRK methods of the same order as 2^{nd} derivative information is used when computing the internal points.

[2] Problems 23 & 32 become singular perturbation problems when $\frac{1}{\epsilon} \to 0$.

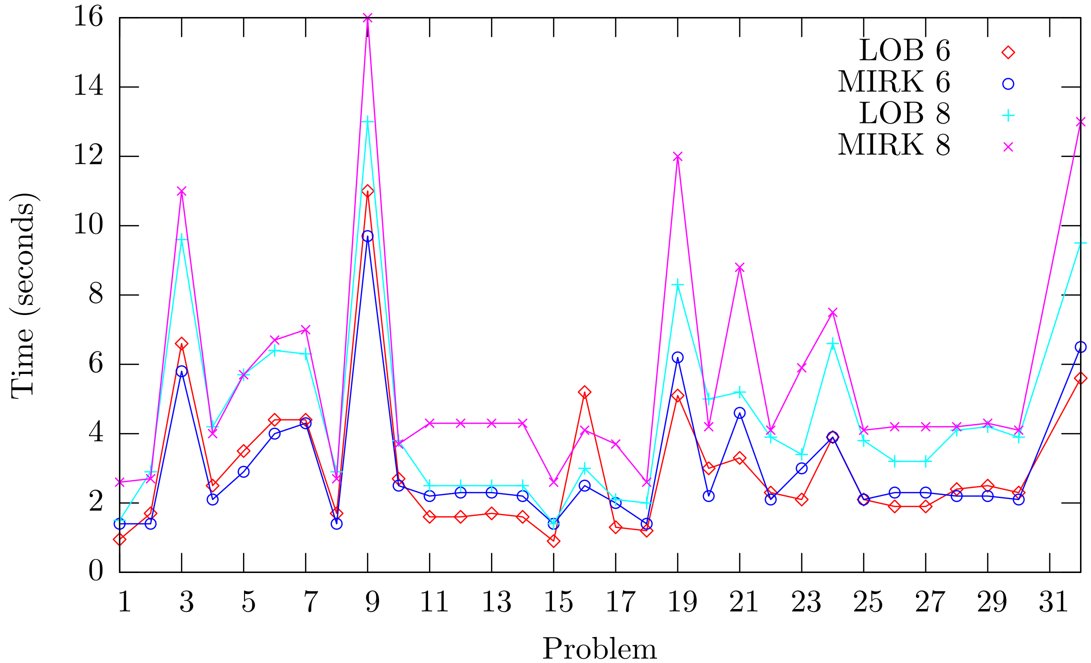

Figure 1: Graph showing the time taken to solve 31 test problems. The Cash 2000 [6] MIRK 8 scheme is used.

4.1 Sixth Order Formulae

Looking at the results comparing the sixth order Lobatto-Obrechkoff pair set out in Section 2 with the sixth order MIRK formula [10] in figure 2, we see that our pair of formulae are noticeably less accurate than the MIRK 6 results for problems 7, 17 and 23. If we compare the accuracy of the y' results, shown in figure 3, our formulae fare worse only for problems 23 and 30. One possible cause for this could be the fact that LOB 6 uses a 4 point integration formula, whilst MIRK 6 uses a 5 point formula which has a smaller local truncation error.

4.2 Eighth Order Formulae

In figure 4 the results comparing the Lobatto-Obrechkoff pair (9a) and (9b) with the eighth order results of [6] can be found. We see that the former pair of formulae yield more accurate results than the MIRK8 (Cash 2000) formulae for all 31 cases! The next figure, 5, shows a similar improvement in accuracy for the L_2 norm for the values of y'.

Overall, we conclude that the Lobatto-Obrechkoff finite difference pairs are more accurate than the MIRK formulae of the same order of accuracy for most problems. Also the eighth order Lobatto-Obrechkoff pair is consistently more accurate than the eighth order MIRK formula.

5 Conclusions

The sixth order and eighth order Lobatto-Obrechkoff finite difference approximations for solving the second order systems $y'' = f(x,\ y)$ and $y'' = f(x,\ y,\ y')$ appear to be more accurate and more efficient when used to solve two-point boundary value problems than using MIRK formulae

Problem	ϵ	LOB 6	MIRK 6	LOB 8	MIRK 8 †
1 *	0.0010	0.95E+00	0.14E+01	0.15E+01	0.26E+01
2	0.0100	0.17E+01	0.14E+01	0.29E+01	0.27E+01
3	0.0500	0.66E+01	0.58E+01	0.96E+01	0.11E+02
4	0.0250	0.25E+01	0.21E+01	0.42E+01	0.40E+01
5	0.0100	0.35E+01	0.29E+01	0.57E+01	0.57E+01
6	0.0220	0.44E+01	0.40E+01	0.64E+01	0.67E+01
7	0.0250	0.44E+01	0.43E+01	0.63E+01	0.70E+01
8	0.0100	0.17E+01	0.14E+01	0.29E+01	0.27E+01
9	0.0550	0.11E+02	0.97E+01	0.13E+02	0.16E+02
10	0.0220	0.27E+01	0.25E+01	0.38E+01	0.37E+01
11 *	0.0010	0.16E+01	0.22E+01	0.25E+01	0.43E+01
12 *	0.0025	0.16E+01	0.23E+01	0.25E+01	0.43E+01
13 *	0.0025	0.17E+01	0.23E+01	0.25E+01	0.43E+01
14 *	0.0025	0.16E+01	0.22E+01	0.25E+01	0.43E+01
15 *	0.0050	0.90E+00	0.14E+01	0.14E+01	0.26E+01
16 *	0.0525	0.52E+01	0.25E+01	0.30E+01	0.41E+01
17 *	0.0005	0.13E+01	0.20E+01	0.21E+01	0.37E+01
18	0.0100	0.12E+01	0.14E+01	0.20E+01	0.26E+01
19	0.0300	0.51E+01	0.62E+01	0.83E+01	0.12E+02
20	0.0500	0.30E+01	0.22E+01	0.50E+01	0.42E+01
21 *	0.0008	0.33E+01	0.46E+01	0.52E+01	0.88E+01
22	0.0250	0.23E+01	0.21E+01	0.39E+01	0.41E+01
23 *	5.0000	0.21E+01	0.30E+01	0.34E+01	0.59E+01
24	0.0300	0.39E+01	0.39E+01	0.66E+01	0.75E+01
25	0.0025	0.21E+01	0.21E+01	0.38E+01	0.41E+01
26	0.0200	0.19E+01	0.23E+01	0.32E+01	0.42E+01
27	0.0200	0.19E+01	0.23E+01	0.32E+01	0.42E+01
28	0.0300	0.24E+01	0.22E+01	0.41E+01	0.42E+01
29	0.0150	0.25E+01	0.22E+01	0.42E+01	0.43E+01
30	0.0420	0.23E+01	0.21E+01	0.39E+01	0.41E+01
32	100.0000	0.56E+01	0.65E+01	0.95E+01	0.13E+02

*This problem is of the form $y'' = f(x, y)$, therefore we are able to omit the computation of the y' terms when formulating the LOB system.

†The Cash 2000 [6] MIRK scheme is used.

Table 1: Time taken, in seconds, to solve 31 of the 32 test problems with uniform meshes of 4096 points using both MIRK and LOB methods.

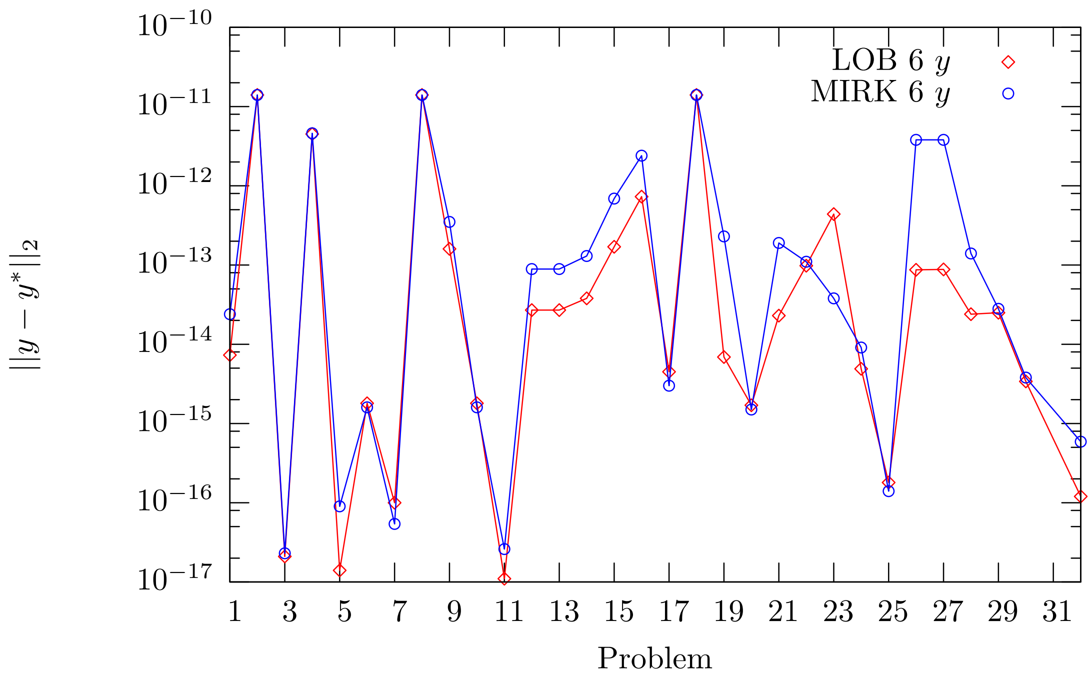

Figure 2: L_2 norm for y_n for 6^{th} order pair applied to 31 of the 32 test problems in [12].

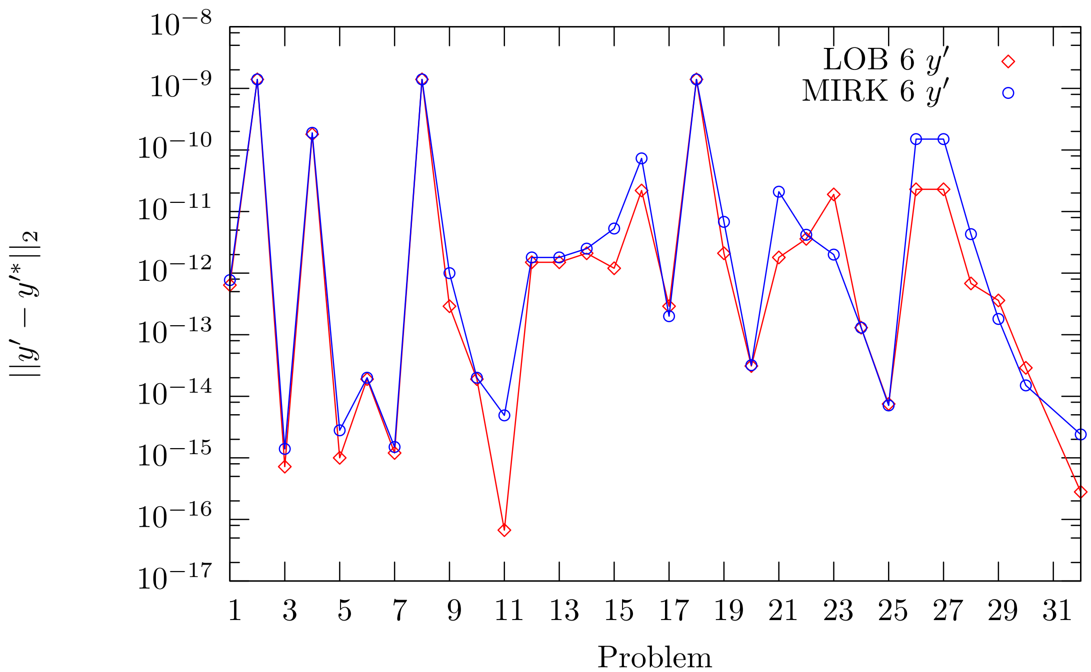

Figure 3: L_2 norm for y'_n for 6^{th} order pair applied to 31 of the 32 test problems in [12].

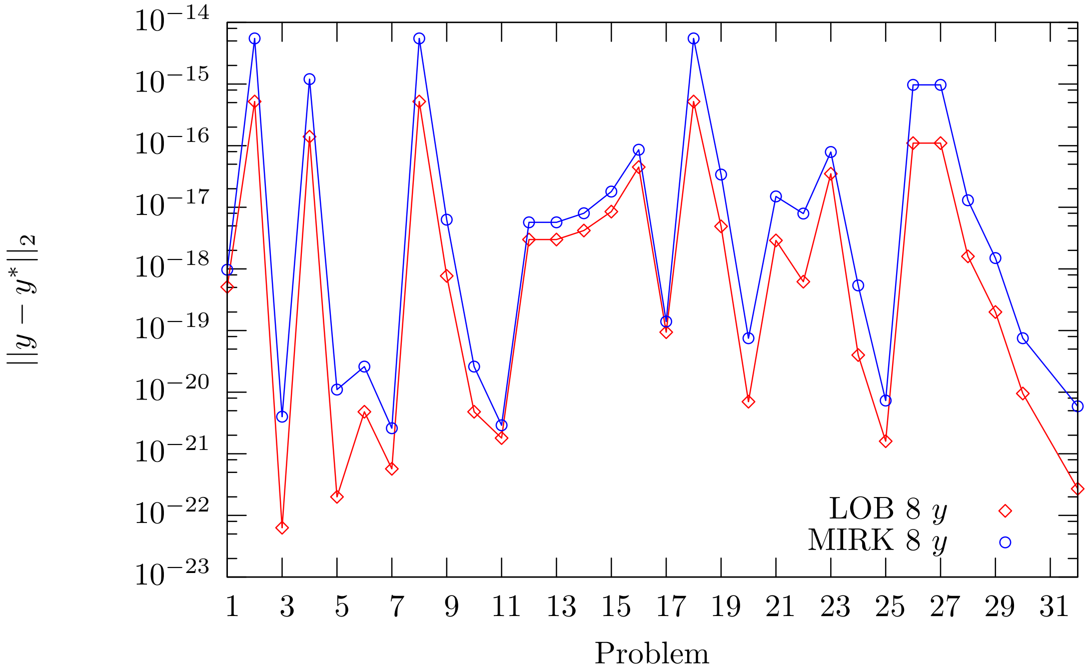

Figure 4: L_2 norm for y_n for 8^{th} order pair applied to 31 of the 32 test problems in [12]. The Cash 2000 [6] MIRK 8 scheme is used.

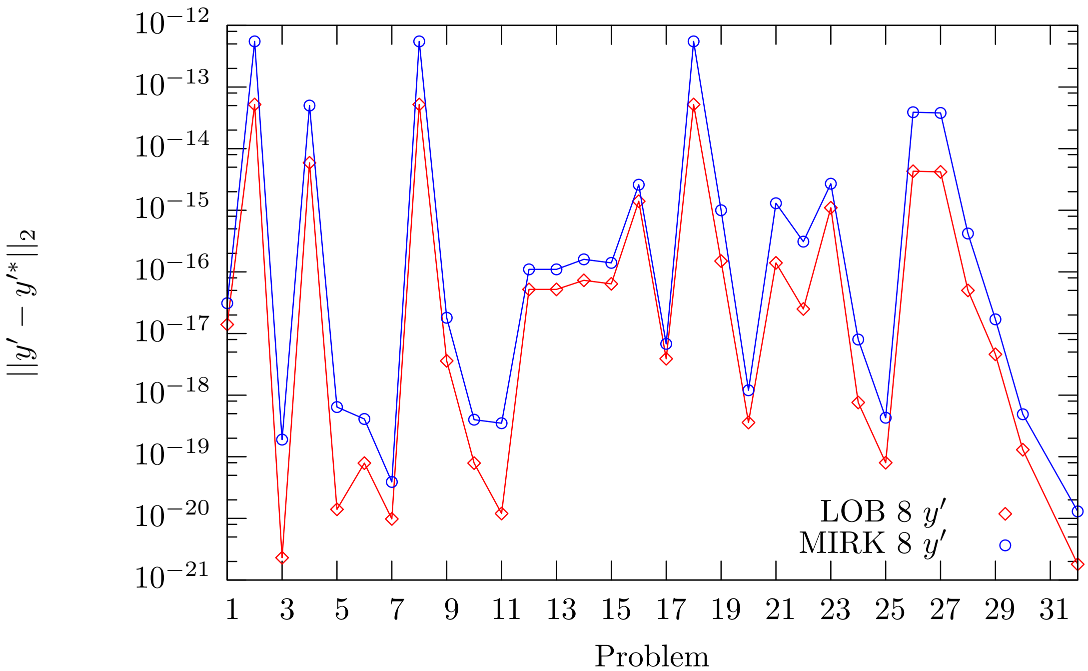

Figure 5: L_2 norm for y'_n for 8^{th} order pair applied to 31 of the 32 test problems in [12]. The Cash 2000 [6] MIRK 8 scheme is used.

of the same orders of accuracy on most of our test problems. This certainly motivates us to carry out research on higher order LOB formulae, which is a topic we hope to report on in a future paper. Indeed if we examine the results of Table 1 we see that for special second order equations of the form $y'' = f(x,\ y)$ the time taken by the eighth order formula LOB 8 is very close to that taken by the sixth order formula MIRK 6.

We recommend these formulae pairs be added to any general purpose finite difference two-point boundary value package where the order of the method is a user choice. The extra accuracy and efficiency of these methods make them very attractive for the special classes of two-point boundary value problems considered in this paper. They can be used both directly and as deferred corrections. The solutions between the mesh points may be constructed economically to the required accuracy using the interpolants described for second order problems in [9].

References

[1] U.M. Ascher, R.M.M. Mattheij & R.D. Russell Numerical solution of boundary value problems for ordinary differential equations *Prentice Hall,* New Jersey (1988).

[2] G. Bader and U. Ascher A new basis implementation for a mixed order boundary value ODE solver *SISSC*, **8**, (1987) 483–500.

[3] J. C. Butcher The Numerical Analysis of Ordinary Differential Equations *John Wiley & Sons* (1987)

[4] S.D. Capper & D.R. Moore Fortran 95 Software for the solution of 2 point boundary value problems, online `http://www.ma.ic.ac.uk/~sdc99/bvp`

[5] S.D. Capper & D.R. Moore On High Order MIRK Schemes and Hermite-Birkhoff Interpolants ICNAAM 2005/6

[6] J.R. Cash On the Derivation of High Order Symmetric MIRK Formulae with Interpolants for Solving Two-Point Boundary Value Problems *New Zealand Journal of Mathematics* **29**, (2000) 129–150.

[7] J.R. Cash, M.P. Garcia & D.R. Moore Mono-implicit Runge-Kutta formulae for the numerical solution of second order nonlinear two-point boundary value problems *J. Computational and Applied Mathematics* **143**, (2002) 275–289.

[8] J. R. Cash & A. D. R. Moore A High order methods for the Numerical Solution of two–point boundary value problems *BIT* **20**, (1980) 44-53.

[9] J.R. Cash & D.R. Moore High Order Interpolants for solutions of Two-Point Boundary Value Problems using MIRK Methods *J. Computational and Applied Mathematics* **48**, (2004) pp 1749-1763.

[10] J. R. Cash & A. Singhal High order methods for the numerical solution of two–point boundary value problems *BIT* **22**, (1982) 184-199.

[11] J.R. Cash & M.H. Wright A deferred correction method for nonlinear two–point boundary value problems. Implementation and numerical evaluation. *SIAM J. Sci. Statistical Computing* **12**, (1991) 971–989.

[12] J.R. Cash & R. W. Wright Set of 32 test problems for BVP codes, online `http://www.ma.ic.ac.uk/~jcash/BVP_software/problems.ps`

[13] D. R. Moore & N. O. Weiss Resonant interactions in thermosolutal convection *Proc. Roy. Soc. Lond.* **456** ,(2000) 39–62.

[14] M. Van Daele & J. R. Cash Superconvergent Deferred Correction Methods for First Order Systems of Nonlinear Two-Point Boundary Value Problems *SIAM J. Scientific Computing* **22**, No. 5, (2000) pp 1697-1716.