I. Troch, F. Breitenecker, eds. ISBN 978-3-901608-35-3

THE SOLUTION OF IMPLICIT DIFFERENTIAL EQUATIONS VIA A LOBATTO IIIC RUNGE-KUTTA METHOD

Brian C. Fabien and Jason P. Frye, University of Washington

Corresponding author: Brian C. Fabien
Department of Mechanical Engineering
University of Washington, Seattle, WA 98195
Email: fabien@u.washington.edu

Abstract. This paper presents a numerical algorithm for the solution of implicit differential equations. The algorithm is based the three stage Lobatto IIIC implicit Runge-Kutta method. A novel feature of the algorithm is that the local truncation error is estimated via a third-order L-stable two-step error estimator, which is more efficient than Richardson extrapolation. In that, the proposed algorithm requires fewer Newton's iterations to compute a solution to the differential equations and an error estimate, when compared to Richardson extrapolation. An implementation of the algorithm is used to solve some benchmark test problems that involve index-3 differential-algebraic equations written in stabilized index-1 form. We compare the performance our implementation with the MATLAB program `ode15i`, as well as the MATLAB program `ride`, which solves implicit differential equations using the three stage Radau IIA method.

1 Introduction

This paper considers the numerical solution of the implicit differential equations (IDEs)

$$\Phi(y, \dot{y}, t) = 0, \tag{1}$$

where the scalar t is the time, $y(t) \in \mathscr{R}^n$ is the state, $\dot{y}(t) \in \mathscr{R}^n$ is the state derivative, and $\Phi(y, \dot{y}, t) \in \mathscr{R}^n$. It is assumed that the initial conditions $y(t_0)$, $\dot{y}(t_0)$ at time $t = t_0$ are consistent, i.e., $\Phi(y(t_0), \dot{y}(t_0), t_0) = 0$. It should be clear that the formulation (1) can accommodate ordinary differential equations (ODEs), as well as differential-algebraic equations (DAEs) [6].

A motivation for considering the implicit differential equations (1) is that the models of complex dynamic systems can be written in this form. In fact, references [3] and [9] have shown that a Lagrangian approach to modeling multidiscipline systems leads to a set of differential-algebraic equations. Moreover, in [4] it is shown that using a coordinate partitioning technique to reduce the index of the DAEs leads to a system of implicit differential equations that are index-1.

A number of algorithms have been developed for the solution of IDEs. Some specific implementations that are based on the backward differentiation formula (BDF) include; DASSL [1], IDA [7], and `ode15i` [12]. Numerical solvers based on implicit Runge-Kutta methods include; `ride` [3] and PSIDE [14]. The program `ride` uses the three stage Radau IIA method and is based on the code RADAU5 [6], with the local error estimator developed in [13]. The program PSIDE uses the four stage Radau IIA method and utilizes a parallel iterative linear system solver.

Recently, some researchers have shown that Lobatto IIIC implicit Runge-Kutta methods can be used to obtain efficient numerical methods for the solution of ODEs and DAEs. For example, these methods are used in [2] to simulate the behavior of multidiscipline dynamic systems, in [8] to solve index-2 DAEs, and in [11] to simulate multibody dynamic systems.

Motivated by these successful applications, in this paper, we describe an implementation of the three stage Lobatto IIIC method for the solution of the IDEs (1). The main contribution of this paper is the development of the two-step error estimator formula which is believed to be new. In section 3 some index-3 DAEs are used to compare the performance of the `l3cide` algorithm with `ride` and `ode15i`.

2 Local error estimation

Successful termination of the Newton's iterations yields the numerical solution $y^{(k+1)}$ at time $t^{(k+1)}$. Before advancing the solution to the next time step, $t^{(k+2)}$, it is desirable to ensure that $y^{(k+1)}$ is a sufficiently accurate solution. This requires that we compute an estimate of the local error in the numerical solution. In the numerical integration algorithm proposed here we compute the local error estimate using two different solutions to the differential equations at $t^{(k+1)}$.

To illustrate how this is accomplished let $y^{(k+1)}$ be the numerical solution determined by one integration method, and let $\hat{y}^{(k+1)}$ be the numerical solution determined by a second integration method. In both cases we assume that the initial condition is exact i.e., $y^{(k)} = \hat{y}^{(k)} = y(t^{(k)})$. Now, if $y^{(k+1)}$ is determined by a method with consistency of order p, and $\hat{y}^{(k+1)}$ is determined by a method with consistency of order $p-1$, then we have

$$\begin{aligned} y(t^{(k+1)}) - y^{(k+1)} &= Ch^{p+1} + O(h^{p+2}), \\ y(t^{(k+1)}) - \hat{y}^{(k+1)} &= \hat{C}h^{p} + O(h^{p+1}), \end{aligned}$$

where C and $\hat{C}$ are constants that depend on the numerical method and the differential equation. Thus, an estimate of the local error can be obtained as

$$\eta^{(k+1)} = \|y^{(k+1)} - \hat{y}^{(k+1)}\| = O(h^p). \tag{2}$$

In the following section we will use the Lobatto IIIC method to design an integration technique that yields a solution $y^{(k+1)}$ with consistency of order $p = 2s-2$, and second solution $\hat{y}^{(k+1)}$ with consistency of order $p-1$.

2.1 A two-step error estimator

Consider the integration of the IDEs (1) from $t^{(k)}$ to $t^{(k+1)}$ in two steps with increments $h/2$. Thus, we determine $y^{(k+1)}$ by applying an s-stage implicit Runge-Kutta method from $t^{(k)}$ to $t^{(k)} + h/2$, and from $t^{(k)} + h/2$ to $t^{(k)} + h = t^{(k+1)}$. In which case $y^{(k+1)}$ is given by

$$y^{(k+1)} = y^{(k)} + \frac{h}{2}\sum_{i=1}^{s} b_i Y_i' + \frac{h}{2}\sum_{i=1}^{s} b_i Y_{s+i}', \tag{3}$$

where Y_1', Y_2', $\cdots, Y_s'$ are the stage derivatives computed in the interval $[t^{(k)}, t^{(k)} + h/2]$, and Y_{s+1}', Y_{s+2}', $\cdots$, Y_{2s}' are the stage derivatives computed in the interval $[t^{(k)} + h/2, t^{(k+1)}]$. Using this approach with the Lobatto IIIC methods it can be shown that (3) has consistency of order $p = 2s-2$ when applied to ODEs [6].

To obtain an estimate of the local error we can compute a second solution to the differential equations at $t^{(k+1)}$ using the formula

$$\hat{y}^{(k+1)} = y^{(k)} + h\sum_{i=1}^{2s} \hat{b}_i Y_i', \tag{4}$$

where the coefficients $\hat{b}_i$, $i = 1, 2, \cdots, 2s$ are determined so that (4) has consistency of order $p-1$. Note that (4) uses the same stage derivatives that are used to compute $y^{(k+1)}$.

The composite implicit Runge-Kutta method described by (3) and (4) can be put in the form of the Butcher tableau

$$\begin{array}{c|c} \bar{c} & \bar{A} \\ \hline & \bar{b}^T \\ & \hat{b}^T \end{array}, \quad \bar{A} = \frac{1}{2}\begin{bmatrix} A & 0 \\ eb^T & A \end{bmatrix}, \quad \bar{b} = \frac{1}{2}\begin{bmatrix} b \\ b \end{bmatrix}, \quad \bar{c} = \frac{1}{2}\begin{bmatrix} c \\ c+e \end{bmatrix},$$

where $e = [1, 1, \cdots, 1]^T \in \mathscr{R}^s$. Written in this form we see that (4) can be viewed as an embedded formula in a $2s$-stage Runge-Kutta method.

Next we select the coefficients $\hat{b}$ so that (4) satisfies certain consistency and stability properties. The design methodology follows a similar approach used in [5] to determined error estimators for Radau IIA methods. Here we will focus on the development of coefficients for the three stage Lobatto IIIC method.

If we apply the method (4) to the scalar ODE $\dot{y} = \lambda y$, then we can show that $\hat{y}^{(k+1)} = R(z)y^{(k)}$, where $z = \lambda h$, and $R(z)$ is called the stability function. Moreover,

$$\begin{aligned} R(z) &= 1 + z\hat{b}^T(I - z\bar{A})^{-1}e \\ &= 1 + z\hat{b}^Te + z^2\hat{b}^T\bar{A}e + z^3\hat{b}^T\bar{A}^2e + z^4\hat{b}^T\bar{A}^3e + z^5\hat{b}^T\bar{A}^4e + z^6\hat{b}^T\bar{A}^5e + O(z^7) \\ &= \frac{\det(I - z\bar{A} + ze\hat{b}^T)}{\det(I - z\bar{A})} = \frac{P(z)}{Q(z)}. \end{aligned} \tag{5}$$

Here, I is the identity matrix, and $P(z)$ and $Q(z)$ are polynomials of degree $2s$.

We also note that the stability function approximates the exact solution to the ODE $\dot{y} = \lambda y$, i.e., $R(z) \approx e^z$. In the case of the three stage Lobatto IIIC method we would like (4) to have consistency of order $p-1 = 3$. This implies that the terms in a Taylor series expansion of $R(z)$ must coincide with with the exact solution e^z up to order z^3. Hence, from (5) we have that the coefficients $\hat{b}$ must satisfy the three conditions

$$\hat{b}^Te = 1, \quad \hat{b}^T\bar{A}e = \frac{1}{2!}, \quad \text{and} \quad \hat{b}^T\bar{A}^2e = \frac{1}{3!}. \tag{6}$$

If these conditions are met then the stability function can be written as

$$R(z) = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \rho_4 z^4 + \rho_5 z^5 + \rho_6 z^6 + O(z^7), \tag{7}$$

where $\rho_4 = \hat{b}^T \bar{A}^3 e$, $\rho_5 = \hat{b}^T \bar{A}^4 e$ and $\rho_6 = \hat{b}^T \bar{A}^5 e$. Using (5) and (7) we can show that

$$\begin{aligned} P(z) &= R(z)Q(z) = R(z)\det(I - z\bar{A}) \\ &= 1 + \frac{z}{4} + \frac{z^2}{64} + \left(\rho_4 - \frac{1}{24}\right) z^4 + \left(\rho_5 - \frac{3}{4}\rho_4 + \frac{35}{1536}\right) z^5 + \left(\rho_6 - \frac{3}{4}\rho_5 + \frac{17}{64}\rho_4 - \frac{77}{12288}\right) z^6. \end{aligned} \tag{8}$$

Thus far we have specified three conditions that must be satisfied by the six terms in $\hat{b}$, i.e., the conditions in (6). We can specify three other conditions as follows. (i) Set the coefficient of z^6 in $P(z)$ equal to zero, i.e., $\rho_6 - \frac{3}{4}\rho_5 + \frac{17}{64}\rho_4 - \frac{77}{12288} = 0$; (ii) Set the coefficient of z^5 in $P(z)$ equal to a free parameter β, i.e., $\rho_5 - \frac{3}{4}\rho_4 + \frac{35}{1536} = \beta$; (iii) Set the coefficient of z^4 in $P(z)$ equal to a free parameter α, i.e., $\rho_4 - \frac{1}{24} = \alpha$.

Note that the condition (i) ensures that $R(\infty) = 0$, which is a necessary condition for L-stability. The free parameters α and β will be selected to ensure that the method (4) is A-stable, and the error estimate is a good approximation at least for ODE $\dot{y} = \lambda y$.

The six conditions in (6), (i), (ii) and (iii) lead to the linear equation

$$\hat{b}^T \left[e,\ \bar{A}e,\ \bar{A}^2 e,\ \left(\bar{A}^5 - \frac{3}{4}\bar{A}^4 + \frac{17}{64}\bar{A}^3\right) e,\ \left(\bar{A}^4 - \frac{3}{4}\bar{A}^3\right) e,\ \bar{A}^3 e \right] = \left[1,\ \frac{1}{2!},\ \frac{1}{3!},\ \frac{77}{12288},\ \beta - \frac{35}{1536},\ \alpha + \frac{1}{24} \right].$$

The solution to this equation is

$$\hat{b} = \begin{bmatrix} 1/12 \\ 192(\beta - 35/1536) - 24(\alpha + 1/24) + 137/24 \\ -1344(\beta - 35/1536) - 24(\alpha + 1/24) - 709/24 \\ 768(\beta - 35/1536) + 96(\alpha + 1/24) + 163/12 \\ 576(\beta - 35/1536) - 72(\alpha + 1/24) + 395/24 \\ -192(\beta - 35/1536) + 24(\alpha + 1/24) - 127/24 \end{bmatrix}, \tag{9}$$

and the Taylor series expansion of the resultant stability function is

$$R(z) = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \frac{24\alpha + 1}{4!} z^4 + \frac{1536\beta + 1152\alpha + 13}{1536} z^5 + \frac{9216\beta + 3648\alpha + 19}{12288} z^6 + O(z^7). \tag{10}$$

Note that selecting $\alpha = 0$ and $\beta = 0$ yields $\hat{b} = \bar{b}$ which is not useful for error estimation. Interestingly however, we note that by selecting $\alpha = 0$ and $\beta = 1/5! - 13/1536$, the stability function, $R(z)$, then approximates e^z up to the term z^5 which is two orders higher that we hoped for. Moreover, this choice of α and β ensures that (4) is A-stable. A-stability and the fact that $R(\infty) = 0$ implies that (4) is also L-stable. This choice for α and β gives the coefficients $\hat{b}^T = [1/12,\ 37/120,\ 31/120,\ -1/60,\ 31/120,\ 13/120]$.

3 Test problems

Several test problems were used for comparing the capabilities of the solver `l3cide`. Here we report partial results for four of these problems, i.e., (i) a quick-return mechanism [3]; (ii) an electromechanical system [9]; (iii) the car axis problem [10]; and (iv) the Andrews' squeezing mechanism problem [10]. All these problems involve index-3 DAEs that are written in stabilized index-1 form using the technique described in [4].

Solutions for each example problem were computed using the three solvers `ride`, `l3cide`, and `ode15i` over a range of tolerance values (refer to [10] for details). The resulting numerical computation statistics were used to produce work-precision diagrams in a similar fashion as in [6]. Computations were performed using MATLAB 7.0.1.15 (R14) on a 1.60 GHz Pentium IV processor.

Figure 1 shows the CPU time versus desired solution tolerance for these four test problems. It can be seen that at modest tolerances ($< 10^{-6}$) all of the solvers require about the same amount of time to compute solutions to the problems. At higher tolerances `ride` and `ode15i` perform better than `l3cide`. This is to be expected since `ride` and `ode15i` are 5-th order methods and `l3cide` is a 4-th order method.

It should be noted that in both `l3cide` and `ride` many parts of the code are written to emphasize clarity rather than speed of execution. Nonetheless, both of these programs have been found to be competitive with `ode15i` when considering the time required to solve a variety of ODEs and DAEs.

The computer codes used to obtain the results reported here can be found at

`http://abs-5.me.washington.edu/`.

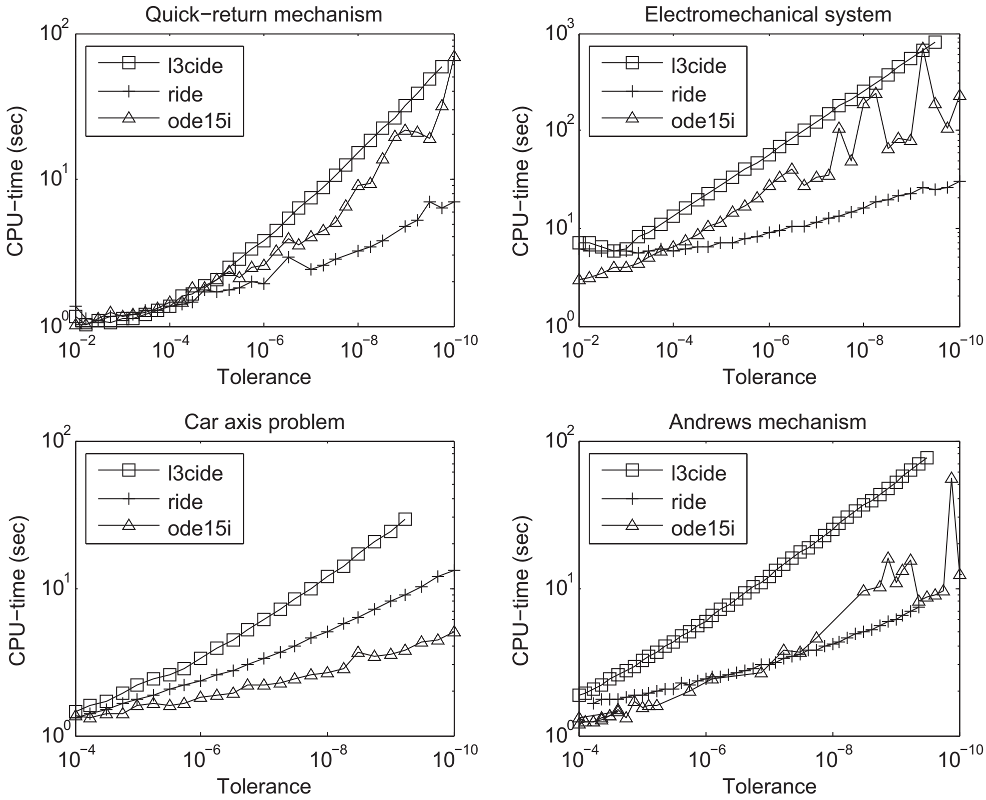

Figure 1: CPU time versus tolerance

4 References

[1] Brenan K. E., Campbell S. L., and Petzold L. R., *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, SIAM Classics in Applied Mathematics **14**, SIAM, Philadelphia, 1996.

[2] Cellier F. E. and Kofman E., *Continuous System Simulation*, Springer, 2006.

[3] Fabien B. C., *Analytical System Dynamics: Modeling and Simulation*, Springer, 2008.

[4] Frye J. P. and Fabien B. C., "A Comparison of Numerical Techniques for the Solution of High-Index Lagrangian DAEs," *Intl. J. Modeling and Simulation*, submitted.

[5] Gonzalez Pinto S., Montijano Torcal J. I. and Perez Rodriguez S., "Two-step error estimators for implicit Runge-Kutta methods applied to stiff systems," *ACM Transactions on Mathematical Software*, **30** (2004), pp. 1–18.

[6] Hairer E. and Wanner G., *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, 2nd ed., Springer-Verlag, Berlin, 1996.

[7] Hindmarsh A. C., Brown P. N., Grant K. E., Lee S. L., Serban R., Shumaker D. E. and Woodward C. S., "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers," *ACM Trans. Math. Softw.*, **31**, (2005), pp. 363–396.

[8] Jay L., "Specialized Runge-Kutta methods for index 2 differential-algebraic equations," *Mathematics of Computation*, **75**, (2005), pp. 641–654.

[9] Layton R. and Fabien B. C., "Systematic Modelling using Lagrangian DAEs," *Mathematical and Computer Modeling of Dynamical Systems*, **7**, (2001), pp. 273–304.

[10] Mazzia F. and Iavernaro F.. "Test Set for Initial Value Problem Solvers," Department of Mathematics, University of Bari, August 2003. Available at `http://www.dm.uniba.it/∼testset`.

[11] Schaub M. and Simeon B., "Blended Lobatto methods in multibody dynamics," *ZAMM*, **83**, (2003), pp. 720–728.

[12] Shampine L. F., "Solving $0 = F(t, y(t), y(t))$ in MATLAB," *J. Numer. Math.*, **10** (2002), pp. 291–310.

[13] de Swart J. and Söderlind G., "On the construction of error estimators for implicit Runge-Kutta methods," *Journal of Computational and Applied Mathematics*, **87**, (1997), pp. 347–358.

[14] de Swart J., Lioen W. M., and van der Veen W. A., *PSIDE*, November 25, 1998. Available at `http://www.cwi.nl/cwi/projects/PSIDE/`.