COMPARATIVE STUDY OF TERMINATING NEWTON ITERATIONS: IN SOLVING ODES

Ronald Tshelametse*

ABSTRACT:

The paper deals with the numerical solution of IVP's for systems of stiff ODE's with particular emphasis on implicit linear multistep methods (LMM), particularly the backward differentiation formulae (BDF). In thispaper we investigate the current strategies that are used to terminate the Newton iterations in the Matlab Code ode15s.We analyse the algorithms for terminating the Newton iterations as implemented in the code ode15s. We conductnumerical experiments to investigate the levels of usage of each strategy in solving various test problems. The experiments reveal the displacement test is oftenmore stringent that other termination strategies.

* Department of Mathematics, University of Botswana, Private Bag 0022, Gaborone, Botswana.

1 Introduction

The paper is concerned with the numerical solution of initial value problems (IVP) for systems of ordinary differential equations (ODEs). These are usuallywritten in the form

$$\frac{du}{dt} = f(t,u),\ \ 0 < t \leq T,\qquad u(0) = u_0,\ \ f: R \times R^m \to R^m \qquad (1)$$

In the literature some initial value problems (1) are referred to as stiff. A prominent feature for these problems is that they are extremely difficult tosolve by standard explicit methods. The time integration of stiff systems isusually achieved using implicit methods, and for many codes by linear multistepmethods. A linear multistep method aims at producing a sequence of values$\{u_n\}$which approximates the true solution of the IVP on the discrete points$\{t_n\}$. Thus the linear multistep formula is a difference equation involving anumber of consecutive approximations $u_{n-i},\ i = 0,1,\dots,k$, from which it willbe able to compute sequentially the sequence $\{u_n\},\ n = 1,2,\dots,N$.Theinteger $k$is called the step number of the method and for a linear multistepmethod $k > 1$. When $k = 1$, the method is called a 1-step method. Linearmultistep methods are also called linear k-step methods [3], [5], [7], [8], [9]. Instandard constant stepsize form a linear multistep or k-step method is defined thus:

$$\sum_{i=0}^{k} \alpha_i u_{n-i} = h \sum_{i=0}^{k} \beta_i f_{n-i}\,, \qquad (2)$$

where $\alpha_i$and β_i are constants and α_0 = 1. f_{n-i}denotes $f(t_{n-i}, u_{n-i}),\ t_{n-i} = t_n - ih,\ i = 0,1,\dots,k$ and $h$is the stepsize. The condition that α_0 = 1removes the arbitrariness that arises from the fact that both sides of the IVPcould be multiplied by the same constant without altering the method. Thelinear multistep method (2) is said to be explicit if$\beta_0 = 0$and implicit if$\beta_0 \neq 0$.

Now let $\beta_i = 0, i = 1,2,\dots,k$in (2) then the result is a class of methods known as the backward differentiation formulae, BDFs [15]. We concentrate onBDFs which take the form

$$u_n + \sum_{i=1}^{k} \alpha_i u_{n-i} = h_n \beta_0 f(u_n) = 0, \qquad (3)$$

Whereh_n is the stepsize, $k$is the order and the coefficients $\alpha_i$depend on k only. In practice codes for integrating stiff IVPs vary the stepsize $h_n$and/or order $k$resulting in variable step variable order BDF implementations [1], [4], [13], [17],[23]. At each integration step $t_n$we must solve the nonlinear equation

$$F(u_n) \equiv u_n + \varphi_n - h_n\beta_0 f(u_n) = 0, \qquad (4)$$

where $\varphi_n = \sum_{i=1}^{k} \alpha_i\, u_{n-i}$is a known value.

To solve for $u_n$most codes use the Newton iterative method and its variantsin the following form

$$W_n^{(l)}\varepsilon_n^{(l)} = -F\left(u_n^{(l)}\right), \qquad u_n^{(l+1)} = u_n^{(l)} + \varepsilon_n^{(l)} \qquad l = 0,12,\ldots \qquad (5)$$

with the starting value $u_n^{(0)}$known and "fairly" accurate. For the full Newtonmethod

$$W_n^{(l)} = F'\left(u_n^{(l)}\right) = I - h_h\beta_0 f'\left(u_n^{(l)}\right) \qquad (6)$$

The use of the Newton method is due to the stiffness phenomenon. For largeproblems evaluating the Jacobian, $f'(u_n^{(l)})$ (and hence the Newton iteration matrix $W_n^{(l)}$) and solving the linear algebraic system are by far the most computationally expensive operations in the integration. There are various strategiesused in practice to try and minimise the cost of computing the Jacobian and theNewton matrix [4], [6], [14], [18]. These measures are mainly centredonadministering the iteration matrix in (6). Other cost saving measures in practical codes include options of using analytical or finite difference Jacobians and at times taking advantage of special structures (banded or sparse) for the linearsolves described by (5) and (6).

2 Current termination strategies

2.1The underlying theory

In solving the IVP

$$\frac{du}{dt} = f(t,u), \quad 0 < t \le T, \qquad u(0) = u_0, \quad f: R \times R^m \to R^m \qquad (7)$$

these codes compute an approximate solution, u_n, of the implicit equation

$$F(u_n) \equiv u_n + \varphi_n - h_n\beta_0 f(t_n, u_n) = 0, \qquad (8)$$

to satisfy, in principle

$$\left|\left|u_n^* - u_n\right|\right| \leq k_1 \times tol, \qquad (9)$$

where $tol$is a user specified tolerance, k_1 is a constant usually less than unityand u_n^*denotes the true solution of (8). Alternatively, a test will be to accept$u_n$if the residual satisfies

$$\left|\left|F(u_n)\right|\right| \leq k_1 \times tol. \qquad (10)$$

In practice most codes that use iterative methods to solve the implicit equations(8) accept the approximation when

$$\left|\left|u_n^{(l)} - u_n^{(l-1)}\right|\right| \leq k_1 \times tol, \qquad (11)$$

where $u_n^{(l)}$and $u_n^{(l-1)}$, $l = 0,1,2,$ ...are the successive iterates, or

$$\left|\left|F\left(u_n^{(l)}\right)\right|\right| \leq k_1 \times tol. \qquad (12)$$

where $u_n^{(l)}$is the current iterate. The tests are called the displacement test andthe residual test respectively. Houbak et al [11] conduct a comparative studyand reveal that it often takes more computational work to satisfy (12) than (11)with little or no gain in the accuracy of the numerical solution of the associatedinitial value problem.

We are mainly interested in how to terminate the iterations

$$W_n\left(u_n^{(l+1)} - u_n^{(l)}\right) = -F\left(u_n^{(l)}\right), \quad l = 0,1,2, \ldots \qquad (13)$$

where $W_n$is an approximation to the Newton iteration matrix, in order to obtain a good approximation, $u_n^{(l+1)}$ to the solution u_n^*. It is common practice toterminate the iterations based on the norm of the difference,

$$d_l = \left\|u_n^{l+1} - u_n^{(l)}\right\|$$

alone. The iterations are terminated as soon as d_l is small enough, but Shampine [21] argued that a small difference $d_l$says nothing about how close $u_n^{(l+1)}$is tou_n^*, nor even that the iteration process is converging. But if the convergencerate factor of the iterative process$\eta < 1$ then a

small difference $d_l$implies that$u_n^{(l+1)}$is an acceptable approximation tou_n^*. This is discussed in [2, p612] whereit is shown that under appropriate the assumptions and for $u_n^{(0)}$ a sufficientlygood approximations to the solution u_n^*, then the simplified Newton methodconverges linearly, that is, $u_n^{(l)} \to u_n^*$with factor $\eta \in (0,1)$. In the stiff ODE applications $u_n^{(0)}$is generally a good approximation.

It can be shown that

$$\left\| u_n^{(l+1)} - u_n^{(l)} \right\| \leq \|\tilde{G}\| \left\| u_n^{(l)} - u_n^{(l-1)} \right\|$$

where $G(u) = u - W^{-1}F(u)$, $u^{(l+1)} = G(u^{(l)})$ and $W$is the simplifiedNewton Iteration matrix. Now assume throughout the region of interest that$\|\tilde{G}\|$is bounded above by someη. It is hoped that $\eta < 1$. In fact for the infinitynorm if $\eta < 1$then by a theorem in [12, p111] the iterates u_n^lconverge to thetrue solution u_n^*if $u_n^{(0)}$is sufficiently close to u_n^*. A similar theorem is discussedin [16, p119] for the general norm. There follows

$$\left\| u_n^{(l+1)} - u_n^{(l)} \right\| \leq \eta \left\| u_n^{(l)} - u_n^{(l-1)} \right\| \qquad (14)$$

and so at the time $u_n^{(l+1)}$is computed,ηcan be estimated as

$$\eta^{(l)} = \left\| u_n^{(l+1)} - u_n^{(l)} \right\| / \left\| u_n^{(l)} - u_n^{(l-1)} \right\| \qquad (15)$$

Now applying the triangle inequality to

$$u_n^{(l+1)} - u_n^* = (u_n^{(l+1)} - u_n^{(l+2)}) + (u_n^{(l+2)} - u_n^{(l+3)}) + \cdots$$

We get

$$\begin{aligned} \left\| u_n^{(l+1)} - u_n^* \right\| &\leq \left\| (u_n^{(l+1)} - u_n^{(l+2)}) \right\| + \left\| u_n^{(l+2)} - u_n^{(l+3)} \right\| + \cdots \\ &\leq \eta \left\| (u_n^{(l+1)} - u_n^{(l)}) \right\| + \eta^2 \left\| (u_n^{(l+1)} - u_n^{(l)}) \right\| + \cdots \\ &\leq \frac{\eta}{1-\eta} \left\| (u_n^{(l+1)} - u_n^{(l)}) \right\| \qquad (16) \end{aligned}$$

It is clear that the iteration error should not be larger than the required tolerance. Therefore the iteration can be stopped when

$$\frac{\eta^{(l)}}{1-\eta^{(l)}}\left\|(u_n^{(l+1)}-u_n^{(l)})\right\| \leq k.tol, \qquad (17)$$

Where $k > 0$ is a suitable constant and $(u_n^{(l+1)})$ accepted as u_n, an approximationu_n^*,where$\eta^{(l)}$is given by (15). It is clear from (15) that at least two iterationsare required to estimate the rate of convergence and hence apply (17). The rateof convergenceη at the previousintegration step can be approximated by takingthere the largest observed$\eta^{(l)}$. This can then be used to judge if in the currentstep, the iterate after 1 (one) Newton iteration, $u_n^{(1)}$is acceptable. Note that therate at the previous integration step, is only applicable to the current step if thesolution and the factor $h_n\beta_0$remain much unchanged. The above analysis is also found in Tshelametse [23] and further discussedby Shampine in [21] and Hairer and Wanner in [10, pp119-121].

The estimate of the convergence rate,$\eta^{(l)}$,at the current iterate is also used todecide when to terminate the Newton iterations. If $\eta^{(l)} > \delta$for some,$\delta < 1$,thenthe iteration is regarded as being too slowly convergent and is then terminated and restarted with a different $u_n^{(0)}$obtained using a different stepsize/order andpossibly an updated Jacobian matrix. In practice we set the maximum numberof iterations, $maxit$If the number is reached before the iteration convergesthen the iterations are terminated and the process restarted.

2.2Terminating Newton Iterations in ODE15s

2.2.1The pseudo code

In ode15s the Newton iterations are terminated as follows. We give each optiona case number for ease of discussion.

- IF $\left\|(u_n^{(l+1)}-u_n^{(l)})\right\| \leq 100 * eps$where $eps$is the machine epsilon, the current $u_n^{(l+1)}$is accepted. This is a typical (relative) displacement test. All norms areweightednorms (CASE1).
- ELSE IF the current iteration is the first (CASE2)
    - IF the convergence rate, η_{n-1} from the previous step is available. That is, if the current time step is not the first time step, then the first iterate $u_n^{(1)}$ is accepted If

$$\frac{\eta_{n-1}}{1-\eta_{n-1}}\left\|(u_n^{(1)}-u_n^{(0)})\right\| \leq 0.05 \times rtol, \qquad (18)$$

where $rtol$is now a scalar (CASE2(A)).

- ELSE the convergence rate is set to zero $(\eta_0 = 0)$ (CASE2(B)).
- ENDIF

- ELSE IF the convergence rate at the current iterate

$$\eta^{(l)} = \frac{\left\|u_n^{(l+1)} - u_n^{(l)}\right\|}{\left\|u_n^{(l)} - u_n^{(l-1)}\right\|} > 0.9 \qquad (19)$$

then the iteration is regarded as too slow and is terminated and restarted with a different$u_n^{(0)}$obtained using a different stepsize/order and possibly an updated Jacobian matrix(CASE3).

- ELSE the convergence rate at the current time step is set to

$$\eta_n = max\left[0.9 * \eta_n, \eta_n^{(l)}\right]$$

(CASE4) and

- IF

$$\frac{\eta_n}{1-\eta_n}\left\|(u_n^{(l+1)} - u_n^{(l)})\right\| \leq 0.5 \times rtol \qquad (20)$$

then the iterate $u_n^{(L+1)}$is accepted. Note the test (18) is more stringent than (20)because we are using the old rate, η_{n-1} (CASE4(A)).

- ELSEIF the iteration has reached the maximum allowed iterations,$l + 1$then it is regarded as too slow and restartedη_{n-1} (CASE4(B)).
- ELSE IF

$$0.5 * tol < \left[\frac{\eta_n}{1-\eta_n}\left\|(u_n^{(l+1)} - u_n^{(l)})\right\|\right]\eta_n^{(maxit-(l+1))} \qquad (21)$$

the iteration is also regarded as too slow and restarted (CASE4(C)). This is be-cause the size of $\left\|(u_n^{(maxit)} - u_n^{*})\right\|$ after $l$iterations can be estimated by

$$\frac{\eta_n^{(maxit-l)}}{1-\eta_n}\left\|(u_n^{(l+1)} - u_n^{(l)})\right\|$$

see Hairer and Wanner [10].

- ENDIF

ENDIF

The norms are either the weighted 2-norm or the weighted in infinity normwith the weights

$$\frac{1.}{\max\left(|u_{n-1}|, \left|u_n^{(l)}\right|, \frac{atol.}{rtol}\right)}, \quad l = 0,1,2, \ldots$$

where for a vector $x$the notation $1./x$ denotes a vector of reciprocals of eachelement x_i, $i = 1,2, \ldots, m$,that is a vector whose elements are $1/x_i$and $|x|$denotes a vector whose elements are $|x_i|$. The parameters $atol$and$rtol$arethe user supplied vectors of absolute and relative tolerances respectively and$atol./rtol$is a vector whose elements are$atol_i/rtol_i$. Note that the weights donot depend upon $atol$and$rtol$unless the magnitude of the elements of u_{n-1} and $u_n^{(l)}$are small compared to $l./rtol$, referred to as the threshold. The defaultvalue of the threshold is $\frac{10^{-6}}{10^{-3}} = 10^{-3}$.

It is clear from the above algorithm that ode15s implements two terminationstrategies, the relative displacement test (CASE1) and the test (23)(CASE4(A))to decide whether to accept the Newton iterate$u_n^{(l)}$. Note that forthe first iteration (CASE2(A)) the code implements the same strategy as in(CASE4(A)) except that it is using the convergence rate from the previous integration step and hence is more stringent. Also see Tshelametse [23].

3Numerical Experiments

In Table 1 (linear problems) and Table 4 (nonlinear test problems) we show theresults of the experiments to investigate which of the two strategies is mostlyused and possibly comment about the strictness of the tests. Most of the testproblems we use are adopted from the Matlab ODE suite. We use the weightedinfinity norm.

Test Problem	Test A (Displacement Test)	Test B (23)
A2ode	0	119
A3ode	138	0
B5ode	1383	0
Fem1ode	0	66

Fem2ode	0	57
Hb3ode	1007	0

Table 1: The total number of times the displacement test, CASE1, (Test A) andthe test (23), CASE2(A) and CASE4(A), (Test B) were used in terminating theNewton iterations for each linear test problem (default tolerances).

Test Problem	Test A (Displacement)	Test B (23)
Ds1ode	0	52
Ds2ode	63	84
Ds4ode	0	143
Fem1ode	0	66
Will1ode	8	109

Table 2: The total number of times the displacement test, CASE1, (Test A) andthe test (23), CASE2(A) and CASE4(A), (Test B) were used in terminating theNewton iterations for each dissipative test problem (default tolerances).

Test Problem	Test A (Displacement)	Test B (23)
Fem1ode	0	66
Fem2ode	0	57
Brussode	0	100
Will1ode	8	109

Table 3: The total number of times the displacement test, CASE1, (Test A) andthe test (23), CASE2(A) and CASE4(A), (Test B) were used in terminating theNewton iterations for each large test problem (default tolerances).

Test Problem	Test A (Displacement)	Test B (23)
Buiode	0	71
Brussode	0	100
Chm6ode	0	171
Chm7ode	0	55
Chm9ode	0	1039
D1ode	0	74
Ds1ode	0	52
Ds2ode	63	84
Ds4ode	0	143
Gearode	0	19
Hb1ode	0	218
Hb2ode	36	590
Vdpode	13	1024
Will1ode	8	109

Table 4: The total number of times the displacement test, CASE1, (Test A) andthe test (23), CASE2(A) and CASE4(A), (Test B) were used in terminating theNewton iterations for each nonlinear test problem (default tolerances).

4Conclusion

We realise that the displacement test (test A) is always implemented first butfor most problems the Newton iterates are accepted via test B. We also notethat for the large linear test problems, fem1ode and fem2ode the code uses testB in the entire integration. This implies that the pure displacement test canoften be more strict than the test (23). This is particularly visible for large testproblems, see Table 3. There is no distinctive pattern on the preferred test forlinear (Table 1), dissipative (Table 2) or nonlinear test problems [21](Table 4).Further investigations could be carried out on the nature of solution and thetype of preferred test for terminating the iterations.

References

[1] Peter N. Brown, George D. Byrne, and Alan C. Hindmarsh. VODE: a variable-coefficient ODE solver. *SIAM J. Sci. Stat. Comput.*, 10, No. 5:1038–1051, September 1989.

[2] Peter N. Brown and Alan C. Hindmarsh.Matrix-free methods for stiffsystems of ODE's.*SIAM J. Numer. Anal*., 23, No. 3:610–638, June 1986.

[3] John C. Butcher. Numerical methods for ordinary differential equations.*John Wiley*, 2003

[4] G. D. Byrne and A. C. Hindmarsh. A polyalgorithm for the numericalsolution of ordinary differential equation.*Comm. ACM*, 1, No. 1:71–96,March 1975.

[5] W. H. Enright, T. E. Hull, and B. Linberg.Comparing numerical methodsfor stiff systems of ODEs.*BIT*, 15:10–48, 1975.

[6] G. Gheri and P.Marzulli.Parallel shooting with error estimate for increasing the accuracy.*J. Comput. and Appl. Math*., 115, Issues 1-2:213–227,March 2000.

[7] E. Hairer. Backward error analysis for linear multistep methods.*Numer.Math*., 84:2:199–232, 1999.

[8] E. Hairer. Conjugate-symplecticityof a linear multistep methods. *J. Computational Mathematics*, 26:5:657–659, 2008.

[9] E. Hairer and C. Lubich.Symmetric multistep methods over long times.*Numer. Math*., 97:4:699–723, 2004.

[10] E. Hairer and G. Wanner.Solving Ordinary Differential Equations II - StiffandAlgebraic Problems.*Springer Verlag, Berlin, Germany*, second revisededition, 1996.

[11] N. Houbak, S. P. Nørsett, and P. G. Thomsen.Displacement or residual testin the application of implicit methods for stiff problems.*IMA J. Numer.Anal*., 5:297–305, 1985.

[12] E. Isaacson and H. B. Keller.Analysis of numerical methods.*John Wileyand Sons*, 1966.

[13] Kenneth R. Jackson. The numerical solution of stiff IVPs for ODEs.*J.Applied Numerical Mathematics*, 1995.

[14] C. T. Kelley. Iterative methods for linear and nonlinear equations.*Societyfor Industrial and Applied Mathematics, Philadelphia, PA, USA*, 1995.

[15] J. D. Lambert. Numerical Methods for Ordinary Differential Systems.*JohnWiley and Sons*, 1991.

[16] James M. Ortega and W. C. Rheinboldt.Iterative solution of nonlinearequations in several

variables.*Academic Press, London, England*, 1970.

[17] L. F. Shampine and P. Bogacki.The effect of changing the stepsize inthe linear multistep codes.*SIAM J. Sci. Stat. Comput*., 10:1010–1023,September 1989.

[18] Lawrence F. Shampine. Numerical solution of ordinary differential equations.*Chapman and Hall*, 1994.

[19] L.F. Shampine. Implementation of implicit formulas for the solution ODEs.*SIAM J. Sci. Stat. Comput.*, 1, No.1:103–118, March 1980.

[20] Peter Tischer. A new order selection strategy for ordinary differential equation solvers.*SIAM J. Sci. Stat. Comput*., 10:1024–1037, September 1989.

[21] R. Tshelametse. An extension of the theory of dorsselaer and spijker.*Botswana Journal of Technology*, 18, number 2:pp. 64–68, 2009.

[22] R. Tshelametse. The milne error estimator for stiff problems. *SouthernAfrica Journal of Pure and Applied Mathematics*, 4,:pp. 13–27, 2009.

[23] R. Tshelametse. Terminating simplified newton iterations: A modifiedstrategy.*International Journal of Management, IT and Engineering*, 4,Issue 4:pp. 246–266, 2014.