Stability Properties of Adams Codes

LAWRENCE F. SHAMPINE
Sandia Laboratories

Although there are a great many variants of Adams methods, only two possibilities are represented among the very best pieces of mathematical software. The behavior of this software when solving mildly stiff initial value problems is studied. Substantial differences between various implementations of the same method, as well as those between different methods, are explained.

Key Words and Phrases. Adams methods, Gear, stability, implicit methods, stiff problems
CR Categories: 5.17

The Adams methods are among the most effective [1, 6, 11] and popular for the solution of the initial value problem for nonstiff ordinary differential equations. The term *Adams method* includes quite a few possibilities. All involve a predictor (explicit, Adams-Bashforth) formula and a corrector (implicit, Adams-Moulton) formula. The orders of these formulas have to be specified, and they need not be the same. For a predictor of order k the only possibilities for a corrector which have been seriously considered are those of orders k and $k + 1$. The number of iterations to be made with the corrector must be specified. Alternatively, a variable number of iterations could be made with the object of solving the corrector equation "exactly." If a fixed number of iterations is made, one must decide whether or not to end the computation of the step with a final evaluation of the derivative. Among the very best codes based on Adams methods, only two methods are represented here [1, 6, 11]. Method I uses a corrector of order one higher than the predictor (also known as local extrapolation when viewed differently), corrects only once, and ends with a final evaluation. This method is represented by Krogh's DVDQ and by Shampine and Gordon's STEP. Method II iterates the corrector to "convergence" and hence is the Adams-Moulton method. It is represented by Gear's DIFSUB and its variants GEAR written by Hindmarsh and STIFF written by Kahaner and Sutherland [7].

Naturally the authors of the codes cited tried to select the best method from the class of Adams methods just described, but they came to different conclusions. Generally speaking, this author has seen no advantage to one method or the other. A striking exception reported in [11] occurs when the codes are confronted with a mildly stiff equation. It is of obvious practical value to understand such differences so as to improve the codes or to see that a choice of method (code?) appropriate to the application is made.

This work was supported by the U.S. Energy Research and Development Administration (ERDA) under Contract AT(29-1)-789
Author's address: Numerical Mathematics Division 5642, Sandia Laboratories, Albuquerque, NM 87185

There are several phenomena we would like to understand. The first thing we should like to clear up is why the results of [11] show that the various codes based on method II behave differently from one another despite the fact that their algorithmic details are simple variations. After doing this we should like to understand the differences between methods I and II. It is puzzling to some why method II shows any stability restriction at all in view of the facts that the first-order formula (alias backward Euler) is stiffly A-stable and the second (alias trapezoidal rule) is A-stable. We explain this and the even more puzzling fact that the codes show a more severe stability restriction at these orders than at higher ones with finite stability regions. The formulas of method I all have finite stability regions and they are smaller than those of method II, so it is puzzling that the codes based on method I perform substantially better when confronted with stiffness. Our explanation for this suggests lines of future development for the codes based on method II.

To account for the difference between the codes based on method II we first note that the main algorithmic changes in going from DIFSUB to its descendant GEAR (or STIFF or another we cite below) are that the latter permits higher orders and uses a different way of deciding when convergence to the Adams-Moulton value has been achieved. The first change accounts, we believe, for most of the improved efficiency observed with GEAR for nonstiff problems. Gear himself wrote a variant of DIFSUB with these changes which was tested in [1] with the name NEW DIFSUB. It was concluded that the changes led to "only a modest improvement" by the criteria of that study. However, in the presence of mild or severe stiffness the performance of GEAR was reported to be much better in [11, Fig. 2b and Table 4]. The availability of higher order formulas cannot explain this effect because the stability of the higher order formulas is much worse than that of the orders implemented in DIFSUB. In the first place we do not think the higher order formulas were being used by the order selection algorithm for reasons we examine later. However, if they had been used, the reduced stability would have made GEAR *less* efficient than DIFSUB rather than more efficient. Thus to account for the behavior observed we must turn to the improved convergence test suggested by A. R. Curtis [4]. The iteration for the solution of the implicit equations is linearly convergent. The idea is to estimate this convergence rate and to use the rate to decide whether convergence has occurred one iteration faster than with the old test. We believe this is an important development which has not received the attention it deserves. Each iteration requires an evaluation of the differential equation, and essentially all evaluations are made for this purpose. Because failed steps are relatively unusual we may approximate the average number of iterations per step from the data reported in [1, 6] for the number of evaluations and the number of successful steps. According to this data, DIFSUB averages 2.8 iterations per step and NEW DIFSUB averages 1.8. This agrees very well with our expectation, based on the nature of the change, that the latter should average one fewer iterations per step. In the presence of mild stiffness when both codes should use the same order, we thus expect that NEW DIFSUB, or equivalently GEAR or STIFF, should be more efficient than DIFSUB by a factor of about $1.8/2.8 = 0.6$. This expectation accounts for the differences seen in [11, Fig. 2b]. Table 4 of [11] reports what happens when a stiff problem is solved. In this case the ratio of costs is 9153/

12787 = 0.7, which is in reasonable agreement with our prediction. (We note that the availability of higher orders may play a role in the difficult transient of this problem by affecting the order selection (as we discuss below).) In conclusion, we believe that the principal difference in the stability properties of the Adams-Moulton method as implemented in the GEAR, STIFF, and NEW DIFSUB codes and in DIFSUB is explained by the more efficient convergence test in the more recent codes.

An explanation for the observed differences between methods I and II involves a number of factors. The first thing we explain is why the stability regions of the Adams-Moulton methods differ from what might be expected. The stability regions expected are those which assume the implicit equations are solved exactly. There are a number of schemes for solving these equations built into the codes, but the standard way is to use simple or functional iteration for nonstiff problems. All the tests cited suppose this scheme. The Adams-Moulton formulas have the form $y_n = \beta_0^* h f(y_n) + \phi_{n,k}$ when solving $y' = f(y)$ [3, p. 112]. Applying these formulas to the test equation used to define the stability region, $y' = \lambda y$ with $\mathrm{Re}(\lambda) \leq 0$, it is well known [3, p. 114] that convergence of simple iteration for arbitrary guesses $y_{n,0}$ holds if and only if $|h\beta_0^*\lambda| < 1$.

The point here is that the *solution technique* imposes a restriction on the product $h\lambda$ of exactly the same kind as that posed by stability. We conclude that the effective stability region of an Adams-Moulton method based on simple iteration is the intersection of the usual region of absolute stability and the disc centered at the origin of radius $1/|\beta_0^*|$. The radii for the low orders are

order	1	2	3	4	5
radius	1	2	12	24	720/19

The interesting thing here is that the regions of absolute stability decrease rapidly as the order increases, but the iteration restriction becomes less severe. We see then that, *as implemented,* the A-stable Adams-Moulton methods of orders 1 and 2 are not actually A-stable. For higher orders the discs reported above do not restrict the stability regions of method II plotted in [3, p. 131]. On comparison we find that the effective stability restriction of a number of these higher order formulas is less severe than that of the "A-stable" formulas.

A simple numerical example will serve to emphasize the effect of the solution technique. The code STIFF allows one to specify either simple iteration or a variant of Newton's method (the chord method). When using the latter the backward Euler method and the trapezoidal rule *are* A-stable, but the Newton iteration is not ordinarily used for nonstiff problems because it is unnecessary and it requires more storage, more overhead, and the expense of forming a Jacobian. We solved the test problem B2 of [2] which has a constant Jacobian with eigenvalues -0.1, -0.5, -1, -4, $-10 \pm 3i$ and is posed on [0, 20]. A pure absolute error tolerance of 10^{-2} was selected with the intention that a crude tolerance lead to the use of low order (1 and 2) formulas where the differences in stability would be prominent. Using simple iteration the problem was solved in 296 steps, and the maximum magnitude of the true (global) error seen at any step was 1.5×10^{-2}. Using the chord method with an analytical Jacobian, the problem was solved in 26 steps with a maximum error of 1.5×10^{-2}. The stability restriction

due to the use of simple iteration clearly must be taken into account to understand results obtained for mildly stiff problems by codes based on method II.

The efficiency of a code depends on both the step size used and the cost of the step. To compare relative efficiency when the step size is restricted to lie in the stability region, we must scale the region by the number of function evaluations. The difficulty which arises here is that we do not know how many evaluations will be needed to solve the corrector equation. Because of this we have noticed no fair comparisons of the implicit and explicit Adams methods in the literature. The direct comparisons that are seen assume, in effect, that the costs are the same. However, we have already observed that a figure of 2.8 evaluations per step is a reasonable one for one way of implementing the implicit formulas, and a figure of 1.8 is a reasonable average value for a more efficient (but somewhat less reliable) way. These figures are of considerable importance in the present context of stiffness as we have already observed in comparing implementations of method II. In what follows we assume the more efficient convergence test is used.

We compare the stability plots for method II given in [3, p. 131] and those for method I given in [10, pp. 135–140]. The latter method costs two evaluations per step which we must account for in scaling the plots. The stiffly A-stable Adams-Moulton method of order 1 has an effective stability region which is the intersection of the disc of radius 1 centered at the origin with the left-half complex plane. Scaling this to account for equal work we see the radius should be taken 1*2/1.8 = 1.1. One must examine the regions to assess their relative merits because their shapes are different, but we think that method I is quite a lot better at order 1—up to a factor of almost 2. The A-stable Adams-Moulton method of order 2 has an effective region which is the intersection of the disc of radius 2 with the left-half complex plane. This should be scaled to 2.2 to compare with the figure for method I. In this case method II is better, although it ranges from a little worse along the real axis to better by a factor of about 1.3 along the imaginary axis. Comparing the (scaled) figures of method II for higher orders to those of method I, it is clear that method II is quite a lot better for orders 3 through 6 with the biggest advantages being along the real axis and the smallest (if any) along the imaginary axis. Because the codes we are discussing vary their orders, we must also consider the relative size of the *best* choice in either case. The best choice for method II (order 3) has a region which ranges up to three *times* the size of the best choices for method I.

Comparison of the scaled stability regions shows that if the step size is limited by stability, the variable order codes based on method II should perform better than those based on method I and for some problems, *much* better. We have cited evidence that among the codes being studied, those based on method II perform worse, not better, than those based on method I. Below we report other computations supporting this contention. It must, then, be the case that the codes based on method I do a better job of selecting their order in the presence of stiffness. In what follows we examine aspects of the order selection algorithms. Briefly, we believe that the codes based on method I do a pretty good job of selecting their order, that those based on method II do not lower their order when they should, and that those based on method II do not adopt the correct strategy for the Adams methods.

Because of the shapes of the regions, the best strategy for the formulas of method I in the presence of stiffness is not unambiguous, but a good approach is to lower the order. This was a design goal for STEP and it does so rather well. Indeed it performs well enough that using a low order is the basis of a pretty successful detector of stiffness in the companion code DE [10, ch. 8]. Krogh [8, 9] has also noted that this should happen and that it does in DVDQ although he has told the author that he does not consider this to be a reliable detector of stiffness in his code. Theoretical and numerical results given in the references cited show that these codes are doing a reasonably efficient job of selecting their order in the presence of stiffness.

It is important that the reader understand that the Gear family of codes implements two kinds of formulas, the Adams and the backward differentiation formulas. The same basic algorithms are used in both cases. While it is true that this approach is simple and saves a few statements in the code, it is evident that it must degrade the performance of one set of formulas in any situation in which the two sets behave differently. For example, it is obvious that considerations of efficiency are quite different as to the consequences of a change of step size when using simple iteration, as is typical with Adams methods, and when using the chord method with a Jacobian computed by differences, as is typical with backward differentiation formulas. The situation at hand is that of the order selection. As the stability regions given in [3, p. 216] for the backward differentiation formulas show, the proper action in the presence of stability restrictions is to lower the order. However, we have seen that with the Adams-Moulton formulas the proper strategy is to move the order towards 3. This conflict, we believe, accounts in large measure for the relatively poor performance of the Adams formulas in this family of codes.

At orders higher than 3 the strategy is the same for the two sets of formulas implemented in the Gear family of codes and we might hope that the order selection algorithm would function properly. There is evidence in the literature based on computations done with the backward differentiation formulas that it does not. Gear [5, p. 14] has recommended that the sixth-order formula not be used at all because of its relatively poor stability region and the fact that the code will sometimes fail to lower the order when it should. This recommendation was adopted in GEAR, STIFF, and NEW DIFSUB. It ameliorates the stability problem but does not eliminate it as example B5 of the tests [2] shows. This example has a Jacobian with eigenvalues relatively near the imaginary axis, and the fifth-order formula exhibits a stability restriction. To be more specific, when B5 is solved with STIFF at a pure absolute error tolerance of 10^{-6}, it takes 99 percent of its steps at order 5 and requires 4167 function evaluations to solve the problem. Restricting the maximum order to 4 enables the code to solve the problem with 3107 evaluations. The proper action, if possible, is to improve the order selection rather than drop the higher order formulas. Dropping the high order formulas must degrade the efficiency of the codes when sufficient accuracy is requested and this has been remarked in experiments with a modified DIFSUB [2, p. 36]. We were attempting to improve the algorithm when we learned of the work of Skelboe [12] who concurs that the order selection is faulty and proposes a remedy. He gives further numerical results illuminating the situation.

The substantial body of experiment reported shows that the Gear family may

not lower the order in the presence of stiffness as it should. We now demonstrate that when using the Adams formulas it does not raise the order when it should.

Our first example is the solution of B2 with a pure absolute error tolerance of 10^{-4}. The code GEAR solves this problem in 331 steps with a cost of 529 evaluations to achieve a solution with a maximum error of 4.8×10^{-4}. The code STEP solves the problem in 188 steps with a cost of 387 evaluations and a maximum error of 4.4×10^{-4}. The latter requires two evaluations for a successful step. The ratio 2.06 of evaluations to steps should instill some caution as to our scheme used earlier of estimating the average number of iterations for method II. Furthermore, the ratio 1.6 for GEAR shows the variation possible in the average of 1.8 we computed earlier from extensive computations. Returning to the matter at hand, we observe that steps are being taken more cheaply by the GEAR code and we recall that its stability regions are (much) better but that it is substantially more expensive. A table of the orders used shows why:

	percent steps at orders							
	1	2	3	4	5	6	7	8
GEAR	31	59	5	5	0	0	0	0
STEP	1	10	20	16	23	21	9	1

The steps taken at order 1 are especially inefficient relative to STEP, but the fact that 90 percent of the steps were at inefficiently low orders explains why the code did not realize its potential.

The variable order codes start at order 1 so one might think that the low orders used by GEAR in the last example originated in this way. We constructed an example to show that this is not the explanation. Consider

$$y' = g'(x) - x[y - g(x)], \quad y(0) = 1$$

where $g(x) = [(20 - x)/20]^{10}$. Obviously the solution is $g(x)$ and the eigenvalue of the Jacobian is real. We solved this problem with a pure absolute error tolerance of 10^{-9} and gathered statistics on the two intervals [0, 2] and [2, 40]. At $x = 1$ the solution is already down to 9.5×10^{-10} so the numerical solution on [2, 40] is purely a question of stability. The stiffness increases on this interval and the fact that the eigenvalue is real means that GEAR might do three times as well as STEP if it chose a good order. Both codes do about the same on [0, 2] and both are at order 6 when they reach $x = 2$. On [2, 40] GEAR took 1202 steps using 2064 function evaluations and resulted in a maximum error of 3×10^{-9}. On [2, 40] STEP took 613 steps using 1271 evaluations and had a maximum error of 7×10^{-9}. Again the code based on method I performed rather better than that based on method II. The orders used were

	percent steps at orders							
	1	2	3	4	5	6	7	8
GEAR	14	20	46	7	2	2	0	0
STEP	0	6	14	40	25	10	6	0

The order selection is better when dropping the order but the algorithm still went too far and took a third of the steps at an order too low.

In conclusion, we have been able to explain a variety of phenomena exhibited by Adams codes in the presence of stiffness. In addition, we have shown that the codes implementing method II perform worse than those implementing method

I but that they are potentially *more* efficient. To achieve this potential in the codes studied, the evidence we have displayed or cited says that the order selection algorithm must be improved in general and that in particular it must distinguish between the Adams formulas and the backward differentiation formulas.

REFERENCES

1. ENRIGHT, W H , AND HULL, T E Test results on initial value methods for non-stiff ordinary differential equations. *SIAM J. Numer. Anal 13* (1976), 944-961.
2. ENRIGHT, W H , HULL, T.E , AND LINDBERG, B Comparing numerical methods for stiff systems of O.D.E.'s *BIT 15* (1975), 10-48.
3. GEAR, C.W. *Numerical Initial Value Problems in Ordinary Differential Equations.* Prentice-Hall, Englewood Cliffs, N.J , 1971
4. GEAR, C.W. Private communication.
5. HINDMARSH, A C. GEAR· Ordinary differential equation system solver. UCID-30001, Rev. 1, Lawrence Livermore Lab., Livermore, Calif., 1972
6. HULL, T.E., ENRIGHT, W.H., FELLEN, B.M., AND SEDGWICK, A.E. Comparing numerical methods for ordinary differential equations. *SIAM J. Numer. Anal 9* (1972), 603-637.
7. KAHANER, D., AND SUTHERLAND, D Code D207 of the program library at Los Alamos Scientific Lab., Los Alamos, N. Mex. The author gratefully acknowledges the generosity and assistance of the writers of this code for privately communicating it to him for purposes of testing, evaluation, and development
8. KROGH, F.T Changing stepsize in the integration of differential equations using modified divided differences Proc Conf. on the Numerical Solution of Ordinary Differential Equations, Lecture Notes in Mathematics No 362, Springer, New York, 1974
9. KROGH, F.T. On testing a subroutine for the numerical integration of ordinary differential equations. *J. ACM 20* (1973), 545-562.
10. SHAMPINE, L.F., AND GORDON, M.K. *Computer Solution of Ordinary Differential Equations The Initial Value Problem.* W.H. Freeman, San Francisco, 1975.
11. SHAMPINE, L.F., WATTS, H.A., AND DAVENPORT, S.M Solving nonstiff ordinary differential equations—the state of the art *SIAM Rev 18* (1976), 376-411.
12. SKELBOE, S. The control of order and steplength for backward differentiation formulas. *BIT 17* (1977), 91-107

Received September 1977, revised January 1978