MATHEMATICS OF COMPUTATION, VOLUME 32, NUMBER 141
JANUARY 1978, PAGES 13–18

Implementing Second-Derivative Multistep Methods Using the Nordsieck Polynomial Representation

By G. K. Gupta

Abstract. A polynomial representation for the second-derivative linear multistep methods for solving ordinary differential equations is presented. This representation leads to an implementation of the second-derivative methods using the Nordsieck polynomial representation. Possible advantages of such an implementation are then discussed.

1. Introduction. In this paper we are concerned with the second-derivative linear multistep methods (formulae). The differential equation being solved is

$$y' = f(x, y), \qquad y(0) = y_0. \tag{1.1}$$

The second-derivative k-step multistep formula may be written as

$$y_{n+1} = \sum_{r=1}^{k} \alpha_r y_{n+1-r} + h \sum_{r=0}^{k} \beta_r y'_{n+1-r} + h^2 \sum_{r=0}^{k} \gamma_r y''_{n+1-r}. \tag{1.2}$$

Several authors have recently studied such formulae and other formulae which include higher derivatives, for example, Makinson (1968), Genin (1974), Makela et al. (1974), Enright (1974a, b) and Liniger and Willoughby (1970). Also Lambert (1973, Sections 7.2 and 8.11) discusses such methods and calls them Obrechkoff methods. The motivation for studying second-derivative and higher derivative formulae is that the usual multistep methods cannot be A-stable for orders higher than 2 (Dahlquist (1963)), while A-stable multi-derivative formulae of higher orders exist as shown by Genin (1974) and Jeltsch (1975). Therefore, higher derivative multistep formulae may be suitable for solving stiff equations. Also while solving stiff equations, the Jacobian $\partial f/\partial y$ is required in the corrector iterations anyway; and therefore, $y'' = (\partial f/\partial y)f + \partial f/\partial x$ can be computed quite easily.

Enright (1974a, b) has presented a subroutine SDBASIC for solving stiff equations and it was shown by Enright et al. (1975) that this subroutine is efficient and reliable for solving a wide range of stiff test problems.

In this paper, we present a polynomial representation of the second-derivative multistep methods and discuss how this representation may be used in implementing the second-derivative methods using the Nordsieck representation. Advantages of such implementations are then briefly discussed. The discussion in this paper is easily extended to higher derivative multistep methods.

Received March 31, 1977.
AMS (MOS) subject classifications (1970). Primary 65L05; Secondary 65D30.
Key words and phrases. Linear multistep methods, stiff differential equations, multiderivative methods, numerical solutions of ordinary differential equations.

2. Polynomial Representation. A polynomial representation of the usual multistep methods has been presented in Wallace and Gupta (1973). We now show how this representation can be extended to include the second-derivative multistep methods.

We assume that the step-size is fixed and, therefore, $x_n = nh$ and y_n is the approximate solution at x_n. Now suppose that the solution after the step to x_n is approximated by a polynomial $P_n(x)$ of degree m, with $P_n(x_n) = y_n$. To advance the solution from x_n to x_{n+1}, using a usual multistep method, we obtain a new degree m approximating polynomial $P_{n+1}(x)$ from the previous polynomial $P_n(x)$ by the relation

$$P_{n+1}(x) = P_n(x) + \delta_{n+1} C((x - x_{n+1})/h), \tag{2.1}$$

where C is a fixed polynomial of degree m characteristic of the particular m-step method employed and δ_{n+1} is chosen to satisfy

$$P'_{n+1}(x_{n+1}) = f(x_{n+1}, P_{n+1}(x_{n+1})).$$

In the second-derivative multistep methods we require that, in addition to $P_{n+1}(x)$ satisfying the differential equation (1.1) at x_{n+1}, it must also satisfy

$$P''_{n+1}(x_{n+1}) = Sf'(x_{n+1}, P_{n+1}(x_{n+1})), \tag{2.2}$$

where S is some constant, usually equal to 1.

Obviously the polynomial $P_{n+1}(x)$ in (2.1) is not capable of satisfying condition (2.2) in addition to satisfying the differential equation at x_{n+1} if we assume that C is a fixed polynomial. However, if we assume that C is not a fixed polynomial then both the conditions at x_{n+1} can be satisfied by the approximating polynomial $P_{n+1}(x)$.

Before we proceed further, we present two examples. The first example illustrates the use of representation (2.1) for Adams-Moulton formulae, and the second example shows how representation (2.1) can be extended to include the second-derivative methods.

Example 1. Consider the Adams-Moulton formula of order three. We require that C has a zero at x_n and that C' has zeros at x_n and x_{n-1}. If we define $t = (x - x_{n+1})/h$, then we require that $C(t)$ be such that $C(-1) = 0$ and $C'(-k) = 0$; $k = 1, 2$. This gives us

$$C(t) = \frac{t^3}{3} + \frac{3t^2}{2} + 2t + \frac{5}{6}.$$

Example 2. Consider the one-step 3rd order second-derivative formula presented by Enright (1973). In the usual notation, the formula is

$$y_{n+1} = y_n + \frac{h}{3}(2f_{n+1} + f_n) - \frac{h^2}{6} f'_{n+1}.$$

This formula is derived by using a polynomial $P_{n+1}(x)$ approximating the solution of the differential equation such that

$P_{n+1}(x_n) = y_n$, $P'_{n+1}(x_n) = f_n$, $P_{n+1}(x_{n+1}) = f_{n+1}$ and $P''_{n+1}(x_{n+1}) = f'_{n+1}$. These conditions imply that $P_{n+1}(x)$ is of degree 3 and also that $C(t)$ must satisfy the following conditions:

$$C(-1) = 0, \quad C'(-1) = 0, \quad C'(0) = 1 \quad \text{(say)}, \quad C''(0) = u, \tag{2.3}$$

where u is yet to be determined. We have put $C'(0) = 1$ arbitrarily because the scaling factor gets included in δ_{n+1} in representation (2.1).

The above four conditions give us

$$C(t) = \frac{(4-u)}{6} + t + \frac{1}{2}ut^2 + \frac{(u-1)}{3}t^3. \tag{2.4}$$

Using this $C(t)$ in representation (2.1) means that we now have to compute u and δ_{n+1} to find $P_{n+1}(x)$. The two conditions which determine u and δ_{n+1} are

$$P'_{n+1}(x_{n+1}) = f(x_{n+1}P_{n+1}(x_n)) \quad \text{and} \quad P''_{n+1}(x_{n+1}) = f'(x_{n+1}, P_{n+1}(x_{n+1})).$$

$C(t)$ may be written as $C(t) = p(t) + uq(t)$, where

$$p(t) = \frac{2}{3} + t - \frac{1}{3}t^3, \qquad q(t) = \frac{1}{6} + \frac{1}{2}t^2 + \frac{1}{3}t^3.$$

Note that $p(t)$ satisfies $p(-1) = 0$, $p'(-1) = 0$, $p'(0) = 1$ and $p''(0) = 0$. Also, $q(t)$ satisfies $q(-1) = 0$, $q'(-1) = 0$, $q'(0) = 0$ and $q''(0) = 1$. This shows that $p(t)$ and $q(t)$ are easily obtained.

We may now rewrite representation (2.1) for the second-derivative method as

$$P_{n+1}(x) = P_n(x) + \delta_{n+1}\{p((x - x_{n+1})/h) + uq((x - x_{n+1})/h)\}. \tag{2.5}$$

It is interesting to note that representation (2.5) presents an equivalent variable-coefficient multistep formula representation similar to that of Lambert and Sigurdsson (1972) for all second-derivative methods. Also, the above representation indicates that the method of 'averaging' used by Liniger and Odeh (1972) is related to the multi-derivative multistep methods.

In the next section we discuss how this representation is helpful in implementing second-derivative methods using the Nordsieck representation.

3. Implementation. In the representation suggested by Nordsieck (1962), a polynomial $P_n(x)$ of degree m at x_n is represented by the following vector:

$$[P_n(x_n), hP'_n(x_n), h^2P''_n(x_n)/2!, \ldots, h^mP_n^{(m)}(x_n)/m!]^T.$$

Using this representation, Gear (1971, p. 217) suggested that the predictor-corrector algorithm may be expressed as follows

$$a_{n+1} = Aa_n + lw, \tag{3.1}$$

where a_{n+1} is the vector of scaled derivatives of $P_{n+1}(x)$ at x_{n+1}, and a_n is the vector of scaled derivatives of $P_n(x)$ at x_n. A is the Pascal triangle, l is the vector of

scaled derivatives of C at x_{n+1} and w is a scalar.

In using (3.1) to represent the second-derivative method, we find that l is not a constant vector. For example, the vector l corresponding to the 3rd order second-derivative method discussed in the last section is

$$\left[\frac{(4-u)}{6}, 1, \frac{u}{2}, \frac{(u-1)}{3}\right]^T.$$

This however can be represented by two vectors, viz.

$$\left[\frac{2}{3}, 1, 0, -\frac{1}{3}\right]^T \quad \text{and} \quad \left[-\frac{1}{6}, 0, \frac{1}{2}, \frac{1}{3}\right]^T;$$

and therefore, we can express the second-derivative methods as follows

$$a_{n+1} = Aa_n + (g + uz)w, \tag{3.2}$$

where g and z are (constant) vectors of scaled derivatives of polynomials p and q in (2.5) at x_{n+1}.

When using the representation (3.2), w and u must be computed by an iterative method. Since the simple iterations do not converge when solving stiff equations, usually the Newton method is used as discussed by Liniger and Willoughby (1970) and Enright (1974). The Newton method in notation (3.2) works as follows: (We assume that a single differential equation is being solved. Later in this section we shall generalize the results to include a system of equations.)

We require that a_{n+1} satisfy the differential equation at x_{n+1}. Therefore,

$$b_1 + wg_1 - hf(x_{n+1}, b_0 + gw + uz_0w) = 0, \tag{3.3}$$

where $b = Aa_n$ and b_0, b_1 are the first and second elements of b and similarly g_0, g_1 and z_0. To compute w, we define the following iterations (w^r and u^r being approximations to w and u).

$$w^{r+1} = w^r - \frac{[b_1 + w^r g_1 - hf(x_{n+1}, b_0 + w^r g_0 + w^r u^r z_0)]}{g_1 - hg_0\,\partial f/\partial y - hz_0u^r\,\partial f/\partial y}. \tag{3.4}$$

This requires an estimate of u^r so that the second-derivative condition of (2.2) is also satisfied. We proceed as follows. Let $a_{n+1,i}$ and $y_{n+1,i}$, $i = 0, 1, \ldots$, be the successive approximations to a_{n+1} and y_{n+1}, respectively, such that $y_{n+1,i}$ is the first element of $a_{n+1,i}$ and

$$a_{n+1,0} = Aa_n, \qquad a_{n+1,i} = a_{n+1,i-1} + (g + u^i z)\Delta_i, \qquad i = 1, 2, \ldots.$$

We have (assuming $z_1 = 0$ and $g_2 = 0$)*

$$\Delta_i g_1 = hf(x_{n+1}, y_{n+1,i-1}) - hf(x_{n+1}, y_{n+1,i-2}).$$

Therefore,

*In the general case, z_1 and g_2 may be nonzero, but this results in the corrector iterations becoming more complex.

$$\Delta_i g_1 = h\frac{\partial f}{\partial y}(y_{n+1,i-1} - y_{n+1,i-2}). \tag{3.5}$$

Also, we want (assuming $\partial f/\partial y$ constant for iterations)

$$\Delta_i u^i z_2 = \frac{h^2}{2}\frac{\partial f}{\partial y}[f(x_{n+1}, y_{n+1,i-1}) - f(x_{n+1}, y_{n+1,i-2})]$$

or

$$\Delta_i u^i z_2 = \frac{h^2}{2}\left(\frac{\partial f}{\partial y}\right)^2 (y_{n+1,i-1} - y_{n+1,i-2}). \tag{3.6}$$

Comparing (3.5) and (3.6) and assuming $g_1 = 1$ and $z_2 = \frac{1}{2}$, we have

$$u^i = h\frac{\partial f}{\partial y}. \tag{3.7}$$

Now we can rewrite (3.4) as

$$\begin{aligned} &W(w^{r+1} - w^r) = -b_1 + w^r g_1 - hf(x_{n+1}, b_0 + w^r g_0 + w^r u^r z_0) \quad \text{and} \\ &W = I - hg_0\left(\frac{\partial f}{\partial y}\right) - h^2 z_0\left(\frac{\partial f}{\partial y}\right)^2. \end{aligned} \tag{3.8}$$

Therefore, at the rth iteration we apply two corrections, one $(w^{r+1} - w^r)g$ and another $(w^{r+1} - w^r)u^{k+1}z$. u^{r+1} is determined by satisfying the second-derivative condition.

We note that the iterations defined by (3.8) are also applicable when (1.1) is a system of differential equations.

4. Concluding Remarks. Using the notation (1.2), Enright (1974a, b) uses the following strategies in the subroutine SDBASIC. To estimate the error, a one-step two half-steps error estimate is used. This requires that three steps must be taken for advancing the solution from x_n to x_{n+1} and two matrices W in (3.8) corresponding to step-size h and $h/2$ must be retained. Also, the order changing strategy is that starting with a third order method, the order is increased if the step-size has been constant for $k + 1$ steps. It seems no new step-size is computed when changing order, and this strategy continues until the order is equal to the maximum order.

Using the Nordsieck representation presented in this paper, error estimation, step-size and order changing techniques similar to those used in DIFSUB of Gear (1971, Chapter 9) may be used, and these should prove to be more efficient.

Also, just as the polynomial representation (2.1) has facilitated search for new (usual) multistep methods, for example, refer to Wallace and Gupta (1973) and Gupta (1976), the polynomial representation (2.5) should facilitate search for new and possibly better second-derivative multistep methods. In addition, should there be a need for computing variable-step coefficients of the second-derivative methods, the representation (3.2) provides us with a simple algorithm. For example, consider the 4th order method of Enright (1974a) using unequal step-sizes. The approximating polynomial $P_{n+1}(x)$ used in deriving it is such that

$$P_{n+1}(x_n) = y_n, \quad P'_{n+1}(x_{n-1}) = f_{n-1}, \quad P'_{n+1}(x_n) = f_n,$$

$$P'_{n+1}(x_{n+1}) = f_{n+1} \quad \text{and} \quad P''_{n+1}(x_{n+1}) = f'_{n+1}.$$

These conditions imply that $p(t)$ and $q(t)$ in notation (2.5) be such that

$$p'(0) = 1, \qquad q'(0) = 0,$$

$$p''(0) = 0, \qquad q''(0) = 1,$$

$$p(-1) = 0, \qquad q(-1) = 0,$$

$$p'(-1) = 0, \qquad q'(-1) = 0,$$

$$p'(t_{n-1}) = 0, \qquad q'(t_{n-1}) = 0,$$

where $t_{n-1} = (x_{n-1} - x_{n+1})/(x_{n+1} - x_n)$. We can now compute $p(t)$ and $q(t)$.

Note Added in Proof. Recently the author has learned of the work of R. D. Skeel and A. K. Wong, *Blended Linear Multistep Methods*, Report UIUCDGS-R-76-800, Department of Computer Science, University of Illinois, 1976.

Department of Computer Science
Monash University
Clayton, Victoria 3168, Australia

1. G. G. DAHLQUIST (1963), "A special stability problem for linear multistep methods," *BIT*, v. 3, pp. 27–43.

2. W. H. ENRIGHT (1974a), "Second derivative multistep methods for stiff ordinary differential equations," *SIAM J. Numer. Anal.*, v. 11, pp. 321–331.

3. W. H. ENRIGHT (1974b), "Optimal second derivative methods for stiff systems," in *Stiff Differential Systems*, R. A. Willoughby (Editor), Plenum Press, New York, pp. 95–109.

4. W. H. ENRIGHT, T. E. HULL & B. LINDBERG (1975), "Comparing numerical methods for stiff systems of ODE:s," *BIT*, v. 15, pp. 10–48.

5. C. W. GEAR (1971), *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, N. J.

6. Y. GENIN (1974), "An algebraic approach to A-stable linear multistep-multiderivative integration formulas," *BIT*, v. 14, pp. 382–406.

7. G. K. GUPTA (1976), "Some new high-order multistep formulae for solving stiff equations," *Math. Comp.*, v. 30, pp. 417–432.

8. R. JELTSCH (1975), *Note on A-Stability of Multistep Multiderivative Methods*, unpublished report.

9. J. D. LAMBERT (1973), *Computation Methods in Ordinary Differential Equations*, Wiley, New York.

10. J. D. LAMBERT & S. T. SIGURDSSON (1972), "Multistep methods with variable matrix coefficients," *SIAM J. Numer. Anal.*, v. 9, pp. 715–733.

11. W. LINIGER & F. ODEH (1972), "A-stable, accurate averaging of multistep methods for stiff differential systems," *IBM J. Res. Develop.*, v. 16, pp. 335–348.

12. W. LINIGER & R. A. WILLOUGHBY (1970), "Efficient integration methods for stiff systems of ordinary differential equations, *SIAM J. Numer. Anal.*, v. 7, pp. 47–66.

13. M. MAKELA, O. NEVANLINNA & A. H. SIPILA (1974), "Exponentially fitted multistep methods by generalized Hermite Birkhoff interpolation," *BIT*, v. 14, pp. 437–451.

14. G. J. MAKINSON (1968), "Stable high order implicit methods for the numerical solution of systems of differential equations," *Comput. J.*, v. 11, pp. 305–310.

15. A. NORDSIECK (1962), "On the numerical integration of ordinary differential equations," *Math. Comp.*, v. 16, pp. 22–49.

16. C. S. WALLACE & G. K. GUPTA (1973), "General linear multistep methods to solve ordinary differential equations," *Austral. Comput. J.*, v.5, pp. 62–69.