

A Polynomial Representation of Hybrid Methods for Solving Ordinary Differential Equations

By G. K. Gupta

Abstract. A polynomial representation of the hybrid methods for solving ordinary differential equations is presented. The advantages of the representation are briefly discussed. Also it is shown that one step taken using a hybrid method is equivalent to two steps of the usual multistep methods; one step taken using an explicit method and the other taken using an implicit method. Therefore, the hybrid methods are really a special case of cyclic methods.

1. Introduction. A linear k -step multistep formula for the solution of the differential equation

$$(1.1) \quad y' = f(x, y), \quad y(x_0') = y_0$$

is usually written as

$$(1.2) \quad y_{n+1} = \sum_{r=1}^k \alpha_r y_{n+1-r} + h \sum_{r=0}^k \beta_r f_{n+1-r}.$$

The above formula has $2k + 1$ unknown α 's and β 's and, therefore, can be of order up to $2k$. However, it was shown by Dahlquist (1956) that the order of the above formula cannot exceed $k + 1$ (if k is odd) or $k + 2$ (if k is even) for the formula to be stable. It was suggested by several authors, e.g., Butcher (1965), Gear (1965), and Gragg and Stetter (1964), that a slight modification of the above formula can overcome the stability condition imposed. The modified formulas were given the name 'hybrid methods' by Gear (1965), because they seem to combine the features of Runge-Kutta and the linear multistep methods by evaluating the function f at 'off-step' points. Thus, the hybrid method with one 'off-step' point looks like (there is no reason why the number of off-step points should be restricted to one)

$$(1.3) \quad y_{n+1} = \sum_{r=1}^k \alpha_r y_{n+1-r} + h \sum_{r=0}^k \beta_r f_{n+1-r} + h\beta_\theta f_{n+1-\theta}.$$

The order of the above formula can be as high as $2k + 2$ if we include θ as a variable. In practice, it has been shown by Kohfeld and Thompson (1967) that stable formulas of order $2k + 2$ can be obtained only for $k \leq 6$. Stable formulas of order $2k + 1$,

Received November 21, 1978.

AMS (MOS) subject classifications (1970). Primary 65L05; Secondary 65D30.

Key words and phrases. Linear multistep methods, hybrid methods, numerical solution of ordinary differential equations.

$k \leq 7$ exist, but it is not known what maximum orders are achievable for stable formulas for $k > 7$. For further discussion refer to Lambert (1973, Chapter 5).

It is obvious that to implement formula (1.3), a value for $f_{n+1-\theta}$ must be computed. This requires computing $y_{n+1-\theta}$; and therefore, two predictors are required, one for computing $y_{n+1-\theta}$ and another for computing the predicted value of y_{n+1} . This and the problems of error estimation make it somewhat more difficult to design an algorithm using hybrid methods than by using the conventional predictor-corrector methods. This actually may have been the reason why interest in the development of hybrid methods declined, and these methods at present are not considered 'competitive' with the other methods available.

This paper looks at a polynomial representation for the hybrid methods and then suggests an implementation using the Nordsieck representation of a polynomial. It is shown how the problems of predicting $y_{n+1-\theta}$ and estimating the error could possibly be overcome.

2. Polynomial Representation. A polynomial representation of the usual linear multistep methods has been presented in Wallace and Gupta (1973) and was extended to represent the second-derivative methods in Gupta (1978). We now show how this representation can be used to represent the hybrid methods.

Assuming that the step-size h is fixed, we have $x_n = x_0 + nh$; and y_n is the computed solution at x_n . If $P_n(x)$ is the polynomial approximation of degree m at x_n , such that $P_n(x_n) = y_n$, then the new polynomial $P_{n+1}(x)$ is obtained using the following relation

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

where C is a fixed polynomial of degree m characteristic of the particular m -step (usual) multistep method being used. δ_{n+1} is chosen to satisfy

$$(2.2) \quad P'_{n+1}(x_{n+1}) = f(x_{n+1}, P_{n+1}(x_{n+1})).$$

In the hybrid 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 an additional condition which usually is

$$(2.3) \quad P'_{n+1}(x_{n+1-\theta}) = f_{n+1-\theta} = f(x_{n+1-\theta}, y_{n+1-\theta}),$$

where θ is usually chosen so that $0 < \theta < 1$ and $y_{n+1-\theta}$ is an approximation to the solution at $x_{n+1-\theta}$ computed in some way.

$P_{n+1}(x)$ as expressed in (2.1) assuming a constant polynomial C , has only one variable δ_{n+1} . Therefore, $P_{n+1}(x)$ can satisfy only one of the above two conditions. To satisfy both conditions C must include a variable and not be a constant polynomial. We now consider one example showing how a variable C can represent a hybrid method. The treatment here is similar to that for the second-derivative methods presented in Gupta (1978).

Example. Consider a one-step hybrid method

$$(2.4) \quad y_{n+1} = y_n + \frac{h}{6}(f_{n+1} + f_n + 4f_{n+\frac{1}{2}}).$$

The above formula is derived using a polynomial $P_{n+1}(x)$ approximating the differential equation such that

$$P_{n+1}(x_n) = y_n, \quad P'_{n+1}(x_n) = f_n, \quad P'_{n+1}(x_{n+\frac{1}{2}}) = f_{n+\frac{1}{2}}, \quad P'_{n+1}(x_{n+1}) = f_{n+1}.$$

The above four conditions imply that $P_{n+1}(x)$ is of degree 3 and also that $C(t)$ must satisfy the following conditions: $C(x = x_n) = 0$, $C'(x = x_n) = 0$, $C'(x = x_{n+1}) = 1$ (say) and $C'(x = x_{n+\frac{1}{2}}) = s$. If we put $t = (x - x_{n+1})/h$, then we want $C(-1) = 0$, $C'(-1) = 0$, $C'(0) = 1$, and $C'(-\frac{1}{2}) = s$, where s 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 conditions give us

$$(2.5) \quad \begin{aligned} C(t) &= \left(\frac{1}{6} + \frac{2}{3}s\right) + t + \left(\frac{3}{2} - 2s\right)t^2 + \frac{2}{3}(1 - 2s)t^3 \\ &= \frac{1}{6} + t + \frac{3}{2}t^2 + \frac{2}{3}t^3 + s\left(\frac{2}{3} - 2t^2 - \frac{4}{3}t^3\right) = p(t) + sq(t). \end{aligned}$$

Note that $p(t)$ satisfies the conditions $p'(0) = 1$, $p'(-1) = 0$, $p'(-\frac{1}{2}) = 0$, $p(-1) = 0$ and $q(t)$ satisfies $q'(0) = 0$, $q'(-1) = 0$, $q'(-\frac{1}{2}) = 1$, $q(-1) = 0$. Therefore, $p(t)$ and $q(t)$ are easily obtained.

The above example shows how the representation (2.1) can be extended to represent the hybrid methods. $P_{n+1}(x)$ for a hybrid method may be written as follows:

$$(2.6) \quad P_{n+1}(x) = P_n(x) + \delta_{n+1,1}p((x - x_{n+1})/h) + \delta_{n+1,2}q((x - x_{n+1})/h),$$

where $\delta_{n+1,1}$ and $\delta_{n+1,2}$ are chosen to satisfy conditions (2.2) and (2.3). We note that the above representation is similar to the representation for second-derivative methods obtained in Gupta (1978) except in that case $\delta_{n+1,1}$ and $\delta_{n+1,2}$ are evaluated to satisfy condition (2.2) and a second-derivative condition. Also, it follows that a hybrid method using two off-step points can be represented by

$$(2.7) \quad \begin{aligned} P_{n+1}(x) &= P_n(x) + \delta_{n+1,1}p((x - x_{n+1})/h) + \delta_{n+1,2}q((x - x_{n+1})/h) \\ &\quad + \delta_{n+1,3}r((x - x_{n+1})/h), \end{aligned}$$

and so on. Again, the above relation is similar to a relation which would be obtained for a multiderivative formula using second and third derivatives of y .

We call formulas of type (2.6) a 2-C multistep method because it is a method requiring two corrections (or needs to satisfy two conditions). Similarly, formulas of type (2.7) could be called 3-C multistep methods.

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

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

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

$$(3.1) \quad a_{n+1} = Aa_n + lw,$$

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 hybrid methods, we find that l is not a constant vector. Similar to the second-derivative methods, hybrid methods can also be represented by the following modification of (3.1):

$$(3.2) \quad a_{n+1} = Aa_n + gw_1 + zw_2,$$

where g and z are (constant) vectors of scaled derivatives of polynomials p and q in (2.6) and w_1 and w_2 are scalars computed to satisfy (2.2) and (2.3). Vectors g and z for the example in the last section are

$$\left[\frac{1}{6}, 1, \frac{3}{2}, \frac{2}{3} \right]^T \quad \text{and} \quad \left[\frac{2}{3}, 0, -2, -\frac{4}{3} \right]^T,$$

We have already commented that in representation (3.2) scalars w_1 and w_2 must be computed by an iterative method to satisfy the two conditions (2.2) and (2.3).

These iterations for hybrid methods turn out to be quite simple. In the example we considered in the last section we have

$$(3.3) \quad p'(x = x_{n+1-\theta}) = p'(-\frac{1}{2}) = 0 \quad \text{and} \quad q'(x = x_{n+1-\theta}) = q'(-\frac{1}{2}) = 1.$$

If p and q satisfy these two conditions, then condition (2.3) will be met if

$$(3.4) \quad P'_{n+1}(x_{n+1-\theta}) = P'_n(x_{n+1-\theta}) + \delta_{n+1,2} = f_{n+1-\theta}.$$

Here we have evaluated the derivative of P_{n+1} given by (2.6). Since $\delta_{n+1,1}$ and $\delta_{n+1,2}$ in representation (2.6) are really the same scalars as w_1 and w_2 in (3.2), the above equation (3.4) immediately gives w_2 . Now only w_1 needs computing, and this is computed iteratively so that P_{n+1} satisfies the differential equation at x_{n+1} . This, of course, is exactly what is done in the usual predictor-corrector scheme. The algorithm for hybrid methods can, therefore, be written as follows: (We have a_n at x_n and we want to compute a_{n+1} at x_{n+1} .)

- (a) Find $P_n(x_{n+1-\theta})$ and $P'_n(x_{n+1-\theta})$.

Since a_n is given the above quantities can be obtained by computing the inner products of a_n with vectors

$$[1, 1 - \theta, (1 - \theta)^2, \dots, (1 - \theta)^m] \quad \text{and} \quad [0, 1 - \theta, 2(1 - \theta)^2, \dots, m(1 - \theta)^m]^T.$$

We have assumed that the ratio $(x_{n+1-\theta} - x_n)/(x_{n+1} - x_n)$ is equal to $1 - \theta$ and used the Taylor series to evaluate $P_n(x_{n+1-\theta})$ and $P'_n(x_{n+1-\theta})$ given the scaled derivatives of P_n at x_n . Note that both vectors used in computing the inner products with a_n are constant for any hybrid formula.

- (b) Evaluate $f_{n+1-\theta} = f(x_{n+1-\theta}, P_n(x_{n+1-\theta}))$.
- (c) Compute $w_2 = f_{n+1-\theta} - P'_n(x_{n+1-\theta})$.
- (d) Compute $Aa_n + zw_2 = a_{n+1}^{(1)}$ (say).
- (e) Correct iterations to compute w_1
 $a_{n+1,1}^{(1)} + g_1 w_1 = hf(x_{n+1}, a_{n+1,0}^{(1)} + g_0 w_1)$
 $a_{n+1,0}^{(1)}$ and g_0 are first elements of vectors $a_{n+1}^{(1)}$ and g , respectively,
 and $a_{n+1,1}^{(1)}, g_1$ are second elements.
- (f) We now have a_{n+1} .

We now note that the simple algorithm above will work only if conditions (3.3) are satisfied. But this must be possible always because if some p and q do not satisfy the conditions, linear combinations of p and q could be found which do.

The presentation in this section is different from the Nordsieck representation of hybrid methods discussed by Kohfeld and Thompson (1968), who do not discuss the polynomial representation.

4. Discussion. The advantages of the polynomial representation are clear. The problem of finding predictors for computing $y_{n+1-\theta}$ is immediately solved and the order of the error term for $y_{n+1-\theta}$ is the same as for the hybrid method. Also, it should be relatively easy to estimate the local error at each step of the hybrid method since the last element of $(gw_1 + zw_2)$ is an approximation to $h^{m+1}y^{(m+1)}/m!$.

The algorithm presented in the last section also gives us further insight into hybrid methods. The algorithm clearly shows that one step taken using a hybrid method is really two steps of the usual multistep methods taken together; first step taken from x_n to $x_{n+1-\theta}$ using an explicit method (given by z) and the second step from $x_{n+1-\theta}$ to x_{n+1} using an implicit method (given by g). For example, if $\theta = \frac{1}{2}$, we can say that one step taken using a hybrid method with a step-size h is equivalent to first advancing the solution to $x_{n+\frac{1}{2}}$ using an explicit method with step-size $h/2$ and then computing y_{n+1} using an implicit method with step-size $h/2$. Therefore, a k -step hybrid method with step-size h could be looked at as a $2k$ -step method with step-size $h/2$. Since $2k$ is even, order $2k + 2$ can be achieved.

Although further investigation is required, it is possible that the hybrid methods may turn out to be efficient for solving stiff equations. Also, it is interesting to note that the hybrid methods are similar to the cyclic methods proposed by Donelson and Hansen (1971) and Stetter (1974).

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

1. J. C. BUTCHER (1965), "A modified multistep method for the numerical integration of ordinary differential equations," *J. Assoc. Comput. Mach.*, v. 12, pp. 124-135.
2. G. G. DAHLQUIST (1956), "Numerical integration of ordinary differential equations," *Math. Scand.*, v. 4, pp. 33-50.
3. J. DONELSON III & E. HANSEN (1971), "Cyclic composite multistep predictor corrector methods," *SIAM J. Numer. Anal.*, v. 8, pp. 137-157.
4. C. S. GEAR (1965), "Hybrid methods for initial value problems in ordinary differential equations," *SIAM J. Numer. Anal.*, v. 2, pp. 69-86.

5. C. S. GEAR (1971), *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, N. J.
6. W. GRAGG & H. J. STETTER (1964), "Generalized multistep predictor-corrector methods," *J. Assoc. Comput. Mach.*, v. 11, pp. 188-209.
7. G. K. GUPTA (1978), "Implementing second-derivative multistep methods using the Nordsieck polynomial representation," *Math. Comp.*, v. 32, pp. 13-18.
8. J. J. KOHFELD & G. T. THOMPSON (1967), "Multistep methods with modified predictors and correctors," *J. Assoc. Comput. Mach.*, v. 14, pp. 155-166.
9. J. J. KOHFELD & G. T. THOMPSON (1968), "A modification of Nordsieck's method using an 'off-step' point," *J. Assoc. Comput. Mach.*, v. 15, pp. 390-401.
10. J. D. LAMBERT (1973), *Computational Methods in Ordinary Differential Equations*, Wiley, New York.
11. A. NORDSIECK (1962), "On the numerical integration of ordinary differential equations," *Math. Comp.*, v. 16, pp. 22-49.
12. H. J. STETTER (1974), "Cyclic finite-difference methods for ordinary differential equations," in *Conference on the Numerical Solution of Differential Equations* (G. A. Watson, Ed.), Lecture Notes in Math., No. 363, Springer-Verlag, Berlin and New York, pp. 134-143.
13. C. S. WALLACE & G. K. GUPTA (1973), "General linear multistep methods to solve ordinary differential equations," *Austral. Comput. J.*, v. 5, pp. 62-69.