MATHEMATICS OF COMPUTATION, VOLUME 33, NUMBER 148
OCTOBER 1979, PAGES 1265–1271

Recurrence Relations for Computing With Modified Divided Differences*

By Fred T. Krogh

Abstract. Modified divided differences (MDD) provide a good way of representing a polynomial passing through points with unequally spaced abscissas. This note gives recurrence relations for computing coefficients in either the monomial or Chebyshev basis from the MDD coefficients, and for computing the MDD coefficients for either the differentiated or the integrated polynomial. The latter operation is likely to be useful if MDD are used in a method for solving stiff differential equations.

1. Introduction. A modification of divided differences was first suggested by Blanch [1] as a means of getting some of the desirable characteristics of differences when working with unequal intervals. Modified divided differences (MDD) of the form used here were introduced in [2] and developed further in [3]. Shampine and Gordon use them in [4]; and Jackson [5] has made a careful study of related computational approaches, concluding that the form used here is best. Jackson has also shown that the round-off characteristics of these modified divided differences are excellent.

This work is the result of requests from users of a code for solving differential equations which uses MDD. Several users of this code have wanted to convert MDD to Chebyshev coefficients in order to get results in a form compatible with existing programs which are used in further processing of the data. For similar reasons another user was interested in converting differences formed from accelerations to the equivalent differences based on velocities. We believe it worthwhile to call attention to the fact that computationally efficient procedures are available for getting to various representations of a polynomial, starting from MDD's.

The polynomial of lowest degree passing through the points $(t_i, w(t_i))$, $i = n, n-1, \ldots, n-q+1$ (notation here reflects the fact that MDD are used primarily for the step by step solution of ordinary differential equations) can be expressed in terms of divided differences, [6, p. 45], using Newton's interpolation formula

$$P_{q-1,n}(t) = w(t_n) + (t - t_n)w[t_n, t_{n-1}] + \cdots + (t - t_n)(t - t_{n-1}) \cdots (t - t_{n-q+2})w[t_n, t_{n-1}, \ldots, t_{n-q+1}] \tag{1}$$

Received August 29, 1977; revised April 7, 1978.
AMS (MOS) subject classifications (1970). Primary 65L05; Secondary 65D05, 65D15.
*This paper presents the results of one phase of research carried out at the Jet Propulsion Laboratory, California Institute of Technology, under Contract No. NAS 7-100, sponsored by the National Aeronautics and Space Administration.

0025-5718/79/0000-0159/$02.75

or in terms of MDD as

$$P_{q-1,n}(t) = P_{q-1,n}(t_n + h_n\tau) = \sum_{i=0}^{q-1} \phi_i(n) c_{i,n}(\tau), \tag{2}$$

where we use notation similar to that in [3]:

$$\begin{aligned}
h_i &= t_i - t_{i-1}, \\
\tau &= (t - t_n)/h_n, \\
\xi_i(n) &= h_n + h_{n-1} + \cdots + h_{n-i} = t_n - t_{n-i-1} = \xi_{i-1}(n-1) + h_n, \\
\alpha_i(n) &= h_n/\xi_i(n), \\
\beta_0(n) &= 1, \\
\beta_{i+1}(n) &= (\xi_i(n)/\xi_i(n-1))\beta_i(n), \qquad (3) \\
\phi_0(n) &= w(t_n), \\
\phi_{i+1}(n) &= \phi_i(n) - \beta_i(n)\phi_i(n-1)\ (= \xi_0\xi_1 \cdots \xi_i w[t_n, t_{n-1}, \ldots, t_{n-i-1}]), \\
c_{0,n} &= 1, \\
c_{i,n} &= [\alpha_{i-1}(n)\tau + \xi_{i-2}(n)/\xi_{i-1}(n)]c_{i-1,n}, \quad i \geqslant 1\ (\xi_{-1}(n) \equiv 0).
\end{aligned}$$

Here we think of the $\phi_i(n)$ as coefficients of a polynomial expressed in terms of the basis polynomials $c_{i,n}(\tau)$. In [3], the presentation focused more on the $\phi_i(n)$, which are the modified divided differences of w computed at $t = t_n, t_{n-1}, \ldots, t_{n-i+1}$. To simplify notation, we will no longer explicitly include n when referring to the various parameters defined in (1)–(3).

Note that for the case of constant stepsize, $\beta_i \equiv 1$ and the MDD reduce to backward differences. In most applications ϕ_i is a vector and computational savings can be realized by taking advantage of constant steps when they occur. More details can be found in [3] or [4].

2. Converting to the Monomial Basis. In order to get to the monomial basis, we represent the polynomial in the right member of (2) as

$$\begin{aligned}
P_{q-1}(\tau) = &\sum_{i=0}^{j} \phi_i \prod_{k=0}^{i-1} \left(\alpha_k\tau + \frac{\xi_{k-1}}{\xi_k}\right) \\
&+ \prod_{k=0}^{j} \left(\alpha_k\tau + \frac{\xi_{k-1}}{\xi_k}\right) \sum_{k=0}^{q-j-2} M_{j,k}\tau^k
\end{aligned} \tag{4}$$

where an empty product has the value 1, $j = q - 1$ gives a representation as in the right member of Eq. (2), and $j = 0$ gives the desired monomial basis. Equating coefficients in Eq. (4) for the cases j and $j + 1$ yields

$$(5)\qquad M_{j,k} = \begin{cases} \phi_{j+1} + \dfrac{\xi_j}{\xi_{j+1}} M_{j+1,0}, & k = 0, \\ \alpha_{j+1} M_{j+1,k-1} + \dfrac{\xi_j}{\xi_{j+1}} M_{j+1,k}, & k = 1, 2, \ldots, q-3-j, \\ \alpha_{j+1} M_{j+1,k-1}, & k = q-2-j. \end{cases}$$

The operation count for this recurrence can be reduced by introducing $\hat{M}_{j,k} = \alpha_j M_{j,k}$, $k \geqslant 0$, $\hat{M}_{j+1,-1} = \phi_{j+1}$. Since $\alpha_0 = 1$, $\hat{M}_{0,k} = M_{0,k}$ = the desired coefficient of τ^{k+1}.

$$(6)\qquad \hat{M}_{j,k} = \begin{cases} \alpha_j \hat{M}_{j+1,k-1} + \hat{M}_{j+1,k}, & k = 0, 1, \ldots, q-3-j, \\ \alpha_j \hat{M}_{j+1,k-1}, & k = q-2-j, \end{cases}$$

where $j = q-2, q-3, \ldots, 0$.

The operations in (6) require only a single vector of storage. For example, with $\hat{M}_{j,k}$ always stored in location ψ_{j+k+1}, (6) becomes

$$(7)\qquad \psi_{j+k+1} = \begin{cases} \alpha_j \psi_{j+k+1} + \psi_{j+k+2}, & k = 0, 1, \ldots, q-3-j, \\ \alpha_j \psi_{j+k+1} = \alpha_j \psi_{q-1}, & k = q-2-j, \end{cases}$$

$j = q-2, q-3, \ldots, 0$. A similar recurrence can be derived from Eq. (5.11) of [3].

3. Converting to the Chebyshev Basis. Proceeding as above for the monomial basis, we write

$$(8)\qquad \begin{aligned} P_{q-1}(\tau) = \sum_{i=0}^{j} \phi_i \prod_{k=0}^{i-1} \left(\alpha_k \tau + \frac{\xi_{k-1}}{\xi_k}\right) \\ + \prod_{k=0}^{j} \left(\alpha_k \tau + \frac{\xi_{k-1}}{\xi_k}\right) \sum_{k=0}^{q-j-2} C_{j,k} T_k(\tau), \end{aligned}$$

where T_k is the Chebyshev polynomial of degree k, $j = q-1$ is the starting point, and in this case to get the final result we need $j = -1$. The step from $j = 0$ to $j = -1$ is a special case and will be treated separately. As before, we equate coefficients in Eq. (8) for the cases j and $j+1$, except this time using the identities: $\tau T_k(\tau) = [T_{k+1}(\tau) + T_{k-1}(\tau)]/2$, $k > 0$, $\tau T_0(x) = T_1(x)$.

(9)
$$C_{j,k}=\begin{cases}\phi_{j+1}+\tfrac{1}{2}\alpha_{j+1}C_{j+1,1}+\dfrac{\xi_j}{\xi_{j+1}}C_{j+1,k}, & k=0,\\ \tfrac{1}{2}\alpha_{j+1}[2C_{j+1,k-1}+C_{j+1,k+1}]+\dfrac{\xi_j}{\xi_{j+1}}C_{j+1,k}, & k=1,\\ \tfrac{1}{2}\alpha_{j+1}[C_{j+1,k-1}+C_{j+1,k+1}]+\dfrac{\xi_j}{\xi_{j+1}}C_{j+1,k}, & k=2,3,\ldots,q-4-j,\\ \tfrac{1}{2}\alpha_{j+1}C_{j+1,k-1}+\dfrac{\xi_j}{\xi_{j+1}}C_{j+1,k}, & k=q-3-j,\\ \tfrac{1}{2}\alpha_{j+1}C_{j+1,k-1}, & k=q-2-j.\end{cases}$$

The operation count for this recurrence is reduced by introducing $\hat{C}_{j,k}=\alpha_j C_{j,k}$ for $k>1$, $\hat{C}_{j,0}=2\alpha_j C_{j,0}$.

(10)
$$\hat{C}_{j,k}=\begin{cases}2\alpha_j\phi_{j+1}, & k=0, j=q-2,\\ 2\alpha_j\phi_{j+1}+\hat{C}_{j+1,0}, & k=0, j=q-3,\\ \alpha_j[2\phi_{j+1}+\hat{C}_{j+1,1}]+\hat{C}_{j+1,0}, & k=0,\\ \tfrac{1}{2}\alpha_j[\hat{C}_{j+1,k-1}+\hat{C}_{j+1,k+1}]+\hat{C}_{j+1,k}, & k=1,2,\ldots,q-4-j,\\ \tfrac{1}{2}\alpha_j\hat{C}_{j+1,k-1}+\hat{C}_{j+1,k}, & k=q-3-j,\\ \tfrac{1}{2}\alpha_j\hat{C}_{j+1,k-1}, & k=q-2-j,\end{cases}$$

$j=q-3, q-4, \ldots, 0$. This step from $j=0$ to $j=-1$ is obtained by equating coefficients in

$$\phi_0+\tau\sum_{k=0}^{q-2}C_{0,k}T_k(\tau)=\sum_{k=0}^{q-1}C_{-1,k}T_k(\tau),$$

using $C_{0,k}=\hat{C}_{0,k}$, $k>0$, $C_{0,0}=\tfrac{1}{2}\hat{C}_{0,0}$.

(11)
$$C_{-1,k}=\begin{cases}\phi_0+\tfrac{1}{2}\hat{C}_{0,1}, & k=0,\\ \tfrac{1}{2}(C_{0,k-1}+C_{0,k+1}), & k=1,2,\ldots,q-3,\\ \tfrac{1}{2}\hat{C}_{0,k-1}, & k=q-2,q-1.\end{cases}$$

As before, the calculations in (10) and (11) can be done in a single vector of storage. Starting with $\psi_k=\phi_k$, and with $\hat{C}_{j,k}$ stored in ψ_{j+k+1}, we obtain

$$\psi_{q-1} = 2\alpha_{q-2}\psi_{q-1} \qquad (j = q-2, k = 0),$$

$$\psi_{q-2} = 2\alpha_{q-3}\psi_{q-2} + \psi_{q-1} \qquad (j = q-3, k = 0),$$

$$\psi_{q-1} = \tfrac{1}{2}\alpha_{q-3}\psi_{q-1} \qquad (j = q-3, k = 1),$$

(12)
$$\psi_{j+k+1} = \begin{cases} \alpha_j[2\psi_{j+1} + \psi_{j+3}] + \psi_{j+2}, & k = 0, \\ \tfrac{1}{2}\alpha_j[\psi_{j+k+1} + \psi_{j+k+3}] + \psi_{j+k+2}, & k = 1, 2, \ldots, q-4-j, \\ \tfrac{1}{2}\alpha_j\psi_{j+k+1} + \psi_{j+k+2}, & k = q-3-j, \\ \tfrac{1}{2}\alpha_j\psi_{j+k+1}, & k = q-2-j, \end{cases}$$

$j = q-4, q-3, \ldots, 0$. And from (11) the final Chebyshev coefficients are obtained from

(13)
$$\begin{cases} \psi_0 = \psi_0 + \tfrac{1}{2}\psi_2, \\ \psi_k = \tfrac{1}{2}(\psi_k + \psi_{k+2}), & k = 1, 2, \ldots, q-3, \\ \psi_k = \tfrac{1}{2}\psi_k, & k = q-2, q-1. \end{cases}$$

It is sometimes convenient to get a scaled value of τ in the T_k. That is to obtain the Chebyshev coefficients for the case $P_{q-1}(\tau) = \Sigma\, C_{-1,k}\, T_k(\lambda\tau)$. Such C's can be obtained by replacing α_j with α_j/λ in the recurrence (12) and replacing all ½'s in (13) with $\tfrac{1}{2}\lambda$. For example, $\lambda = h$ gives an expansion using $T_k(t - t_n)$ in place of $T_k(t - t_n)/h$.

If one is using the method outlined in [3] or the code in [4], the α's are available at the time of integration. Except for the case $\lambda = \tfrac{1}{2}$, a multiplication can be saved in the innermost loop of (12) by storing $\alpha_j/2\lambda$ in a temporary location when starting a new value of j.

If one wants to approximate the solution of a differential equation by a Chebyshev polynomial, and thus wants to get the coefficients of the integrated polynomial, one can save a little computation by computing such coefficients from the $C_{0,k}$ instead of the $C_{-1,k}$. Denoting the kth coefficient of the integrated polynomial by $\tilde{C}_k$, and using the identity

$$\int T_k(\tau)\,d\tau = \frac{1}{2}\left[\frac{T_{k+1}(\tau)}{k+1} - \frac{T_{k-1}(\tau)}{k-1}\right], \qquad k \geqslant 2,$$

one has for $2 \leqslant k \leqslant q-4$ (other values of k are special cases)

(14) $\tilde{C}_k = (\tfrac{1}{2}k)[C_{-1,k-1} - C_{-1,k+1}]$, or $\tilde{C}_k = (\tfrac{1}{4}k)[C_{0,k-2} - \hat{C}_{0,k+2}]$.

4. Computing Coefficients for the Integrated or Differentiated Polynomial. When integrating stiff equations, the best integration method depends on whether the most active transient has decayed. If a procedure such as that in [3] is used, changes

in the method involve either getting coefficients for the integrated polynomial when leaving a transient region, or for the differentiated polynomial when a transient is encountered.

The best algorithm we have been able to find for the case of variable stepsize is based on using the recurrence in (7) to get the polynomial in the monomial basis, differentiating or integrating this polynomial (which is trivial) and then doing the reverse of (7) to get back to the original basis. Thus after running through the recurrence in (7), ψ_k is replaced by $(k+1)\psi_{k+1}$, $k = 0, 1, \ldots, q-2$, for differentiation (or with ψ_{k-1}/k, $k = q, q-1, \ldots, 1$, for integration). In the case of integration, ψ_0 is set to the constant of integration. The reverse of (7) is given by

$$\psi_{j+k+1} = \begin{cases} \psi_{q'-1}/\alpha_j, & k = q'-2-j, \\ (\psi_{j+k+2} - \psi_{j+k+1})/\alpha_j, & k = q'-3-j, q'-4-j, \ldots, 0, \end{cases} \tag{15}$$

$j = 0, 1, \ldots, q'-2$, where $q' = q+1$, for integration, and $q' = q-1$, for differentiation. In the case of a constant stepsize, one can accomplish this goal more efficiently by repeated differencing of the corrector equation

$$\nabla y_n = h \sum_{k=0}^{q-1} \gamma_k^* \nabla y_n'. \tag{16}$$

Thus after applying the operator ∇^{j-1} to both sides of Eq. (16) and replacing $\nabla^{k+j-1} y_n'$ by 0 for $k+j-1 \geqslant q$, one obtains

$$\nabla^j y_n = h \sum_{k=0}^{q-j} \gamma_k^* \nabla^{k+j-1} y_n', \qquad j = 1, 2, \ldots, q, \tag{17}$$

and the reverse operation simply involves treating (17) as a triangular system to be solved for the $\nabla^k y_n'$.

Aknowledgement. The results in this paper were first obtained by substituting into the formulas given by Salzer [7]. We are grateful to a referee for the suggestion that the interpolating polynomial be expressed in terms of the coefficients which arise in the recurrence. This suggestion led to the self-contained and cleaner presentation given here.

Programming Development Section
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, California 91103

1. G. BLANCH, "On modified divided differences," *Math. Comp.*, v. 8, 1954, pp. 1–11, 67–75.

2. F. T. KROGH, "A variable step variable order multistep method for the numerical solution of ordinary differential equations," *Information Processing* 68 (Proceedings of the IFIP Congress, 1968), North-Holland, Amsterdam, 1961, pp. 194–199.

3. F. T. KROGH, "Changing stepsize in the integration of differential equations using modified divided differences," *Proceedings of the Conference on the Numerical Solution of Ordinary Differential Equations,* October 1972, Lecture Notes in Math., vol. 362, Springer-Verlag, New York, 1974, pp. 22–71.

4. L. F. SHAMPINE & M. K. GORDON, *Computer Solution of Ordinary Differential Equations, The Initial Value Problem,* Freeman, San Francisco, Calif., 1975.

5. L. W. JACKSON, *The Computation of Coefficients of Variable-Step Adams Methods,* Technical Report No. 94, Dept. of Comput. Sci., Univ. of Toronto, 1976.

6. F. B. HILDEBRAND, *Introduction to Numerical Analysis,* McGraw-Hill, New York, 1956.

7. H. E. SALZER, "A recurrence scheme for converting from one orthogonal expansion into another," *Comm. ACM,* v. 16, 1973, pp. 705–707.