JOURNAL OF COMPUTATIONAL PHYSICS 17, 420–424 (1975)

# Note

## Note on Laplace Transforms of Osculatory and Hyperosculatory Interpolation Polynomials

### INTRODUCTION

The numerical evaluation of $F(p)$, the Laplace transform of $f(t)$, for prescribed values of $p > 0$, involves the quadrature

$$F(p) = \int_0^\infty e^{-pt} f(t)\, dt. \tag{1}$$

When $f(t)$ is approximable by polynomials, even if the required degree is very high in order to obtain $F(p)$ accurately, it is usually most convenient to employ Gauss–Laguerre quadrature

$$F(p) = \frac{1}{p} \int_0^\infty e^{-t} \phi(t)\, dt \sim \frac{1}{p} \sum_{j=1}^{n} A_j \phi(t_j), \tag{2}$$

where $\phi(t) = f(t/p)$, which is exact when $f$ or $\phi$ is a $(2n-1)$th degree polynomial. Tables of $t_j$ and $A_j$, the zeros and Christoffel numbers of the Laguerre polynomials, are available in multiple precision up to very high $n$, the most extensive of which are for $n = 2(1)\,32(4)\,68$, to $30S$ [5, pp. 253–275].

Now often $f(t_j/p)$ is not easily calculated and values of both $f(i)$ and $f'(i)$ happen to be known at the integral points $i = 0(1)\, n-1$.[1] For example, some functions occur naturally only at integral points, or are available from previously calculated tables where the arguments are at equal intervals. Also when $f$ and $f'$ satisfy fairly simple difference equations, it is frequently easier to generate $f(i)$ and $f'(i)$ than to calculate $f(t_j/p)$. A sufficiently good approximation to $F(p)$ is often obtainable by replacing $f(t)$ in (1) by $L_{2n-1}(t)$, the $(2n-1)$th degree osculatory interpolation polynomial based upon $f(i)$ and $f'(i)$, $i = 0(1)\, n-1$, where $n$ is not too large. An earlier table which was calculated from the Laplace transforms of only ordinary $n$-point interpolation polynomials of degree $n-1$, for the nodes $i = 0(1)\, n-1$, $n = 2(1)\, 11$, proved to be quite accurate in a variety of test examples [1].[2]

[1] When given $f(ih)$ and $f'(ih)$, instead of $f(i)$ and $f'(i)$, change the variables in (1) to $t' = t/h$ and $p' = ph$. Then if $g(t') = f(t) = f(ht')$, we have $g(i) = f(ih)$, $g'(i) = hf'(ih)$, and $F(p) = hG(p') = h\int_0^\infty e^{-p't}g(t)dt$.

[2] In [1], the fourth paragraph on p. 1 and the second paragraph on p. 4 are not entirely correct and should be deleted.

From the success of [1], it was natural to consider the approximation

$$F(p) = \int_0^\infty e^{-pt} f(t)\, dt \sim \int_0^\infty e^{-pt} L_{2n-1}(t)\, dt$$
$$= \sum_{i=0}^{n-1} \{A_i^{(n)}(p) f(i) + B_i^{(n)}(p) f'(i)\}, \tag{3}$$

where $A_i^{(n)}(p)$ and $B_i^{(n)}(p)$ are the Laplace transforms of the coefficients of $f(i)$ and $f'(i)$ in $L_{2n-1}(t)$, namely,

$$A_i^{(n)}(p) = \int_0^\infty e^{-pt}\{(L_i^{(n)}(t))^2 (1 - 2L_i^{(n)\prime}(i)(t-i))\}\, dt, \tag{4}$$

and

$$B_i^{(n)}(p) = \int_0^\infty e^{-pt}\{(L_i^{(n)}(t))^2 (t-i)\}\, dt, \tag{5}$$

where $L_i^{(n)}(t) = \prod_{j=0, j\neq i}^{n-1} (t-j)/\prod_{j=0, j\neq i}^{n-1} (i-j)$. Some time ago, $A_i^{(n)}(p)$ and $B_i^{(n)}(p)$, which are polynomials of degree $2n$ in $1/p$, without a constant term, were obtained for $n = 2(1)\,9$, the coefficients given as exact rational numbers, with the original purpose of producing numerical tables of $A_i^{(n)}(p)$ and $B_i^{(n)}(p)$ for argument $p$, similar to those in [1]. That project was discontinued in view of the widespread number of presently available computer programs for generating polynomials to multiple precision, and $A_i^{(n)}(p)$ and $B_i^{(n)}(p)$ were deposited in the Unpublished Mathematical Tables File in *Mathematics of Computation* [2]. For $n = 2(1)\,9$, the rightmost member of (3) is obtainable as the matrix product $ABC^T$, where $A$ is the $1 \times 2n$ row-matrix $\| f(0) f(1) \ldots f(n-1) f'(0) f'(1) \ldots f'(n-1) \|$, $B$ is the $2n \times 2n$ square matrix of coefficients of $p^{-m}$, $m = 1(1)\,2n$, in both $A_i^{(n)}(p)$ and $B_i^{(n)}(p)$, $i = 0(1)\,n-1$, and $C$ is the $1 \times 2n$ row-matrix $\| p^{-1} p^{-2} \ldots p^{-2n} \|$. For fixed or few values of $A(C)$ it might be economical to precompute and store $AB(BC^T)$ to use for many different values of $C(A)$. Printing out $BC^T$ for a large number of regularly spaced $p$'s would fulfill the original intention of having tables of $A_i^{(n)}(p)$ and $B_i^{(n)}(p)$ available for wide distribution and handy in desk calculation.

The purpose of this note is to show how another way of computing the Laplace transforms of the interpolation polynomials in the osculatory and hyperosculatory cases, which employs the barycentric form of the interpolation polynomial in conjunction with Gauss–Laguerre quadrature, is much more efficient and better adapted to present-day computers.

*Osculatory Case.* The rightmost member of (3) is calculated for a given $p$ by a method that bypasses the computation of $A_i^{(n)}(p)$ and $B_i^{(n)}(p)$. The method employs a variation of (2) where $\phi(t_j) = f(t_j/p)$ is replaced by $L_{2n-1}(t_j/p)$. Since (2) is an exact formula when $\phi(t)$ is any polynomial of the $(2n-1)$th degree, no accuracy is lost when $t_j/p$ falls far outside the interpolation interval $[0, n-1]$, although that will probably require extra places in the computation. The calculation of $L_{2n-1}(t_j/p)$ is facilitated by employing the barycentric form of $L_{2n-1}(t)$ [3]. First we obtain

$$\beta_{i,j} = a_i/(t_j/p - i) \qquad \text{and} \qquad \alpha_{i,j} = (\beta_{i,j} + b_i)/(t_j/p - i),$$
$$i = 0(1)\, n-1, \qquad j = 1(1)\, n, \tag{6}$$

where $a_i$ and $b_i$, which are exact integers, have been previously tabulated up to $n = 11$ for 21th degree accuracy in $L_{2n-1}(t)$ [3, *p*. 215].[3] Both $a_i$ and $b_i$ have few digits, at the most eight. Next we find

$$L_{2n-1}(t_j/p) = \sum_{i=0}^{n-1} \{\alpha_{i,j} f(i) + \beta_{i,j} f'(i)\} \Big/ \sum_{i=0}^{n-1} \alpha_{i,j}, \qquad j = 1(1)n. \tag{7}$$

Finally we obtain

$$F(p) \sim \frac{1}{p} \int_0^\infty e^{-t} L_{2n-1}(t/p)\, dt = \frac{1}{p} \sum_{j=1}^{n} A_j L_{2n-1}(t_j/p). \tag{8}$$

The main advantage in (6)–(8) over the previous $ABC^T$ method using the UMT File in [2] is that there is much less storage. We discount the storage of $f(i)$ and $f'(i)$ that is common to both methods. For (6) and (8) we store the auxiliary quantities $a_i$ and $b_i$, $i = 0(1)\, n-1$, and $t_j$ and $A_j$, $j = 1(1)\, n$. Since $a_i = a_{n-1-i}$ and $b_i = -b_{n-1-i}$, actually only $2[(n+1)/2] + 2n \sim 3n$ distinct quantities are stored. The $ABC^T$ method involves the storage of $4n^2$ numerators of fractions (exceeding 15 digits at $n = 9$) and $2n$ l.c.d.'s, one for each $A_i^{(n)}(p)$ and $B_i^{(n)}(p)$. The number of $m-d$'s (multiplication-divisions) in (6)–(8) is $4n^2 + 3n + 1$, which is only slightly less than the $4n^2 + 4n$[4] $m - d$'s in using [2]. But in (6)–(8) the $m-d$ operations are more compact in arrangement and more efficiently performed because of the much smaller number of stored constants, and also because of the smaller average number of digits per constant.

*Hyperosculatory Case.* Here also the $(3n-1)$th degree interpolation polynomial $L_{3n-1}(t)$ which is based upon $f(i)$, $f'(i)$ and $f''(i)$, $i = 0(1)\, n-1$ (often

[3] In [3] and [4], for $n > 2$, add $[(n-1)/2]$ to the subscript $i$ in $a_i$, $b_i$ and $c_i$ to agree with the present notation. $[\cdots]$ denotes the nearest integer throughout this article.

[4] This estimate ignores a substantial amount of recent literature on reducing the number of multiplications in evaluating polynomials, by a more complicated arrangement of operations, which appears to be too cumbersome to apply here

$f''(i)$ is had immediately from $f(i)$ and $f'(i)$ from a second-order differential equation), is expressed in barycentric form [4]. Tables of auxiliary coefficients $a_i$, $b_i$ and $c_i$ for calculating $L_{3n-1}(t)$ are available for $n = 2(1)\,7$, providing up to 20th degree accuracy [4, p. 106].[3] As before, $a_i$, $b_i$ and $c_i$ are integers having at most eight digits. We calculate first

$$\gamma_{i,j} = a_i/(t_j/p - i), \qquad \beta_{i,j} = (\gamma_{i,j} + b_i)/(t_j/p - i) \qquad \text{and}$$
$$\alpha_{i,j} = (\beta_{i,j} + c_i)/(t_j/p - i), \qquad i = 0(1)\,n-1, \qquad j = 1(1)[(3n+1)/2], \tag{9}$$

and then

$$L_{3n-1}(t_j/p) = \sum_{i=0}^{n-1} \{\alpha_{i,j} f(i) + \beta_{i,j} f'(i) + \tfrac{1}{2}\gamma_{i,j} f''(i)\} \Big/ \sum_{i=0}^{n-1} \alpha_{i,j},$$
$$j = 1(1)[(3n+1)/2], \tag{10}$$

and finally, employing a Gauss–Laguerre formula to attain at least $(3n-1)$th degree accuracy,

$$F(p) \sim \frac{1}{p} \int_0^\infty e^{-t} L_{3n-1}(t/p)\,dt = \frac{1}{p} \sum_{j=1}^{[(3n+1)/2]} A_j L_{3n-1}(t_j/p). \tag{11}$$

The advantages of (9)–(11) over calculating the Laplace transforms of hyperosculatory interpolation coefficients as polynomials in $1/p$,[5] namely, a big reduction in the number of stored constants, with a smaller average number of digits per constant, making for more compactness and greater efficiency in machine computation, are even more marked than in the preceding osculatory case. Here we discount the storage of $f(i)$, $f'(i)$, and $f''(i)$ that is common to both methods. For (9) and (11) we store the auxiliary constants $a_i$, $b_i$, and $c_i$, $i = 0(1)\,n-1$, and $t_j$ and $A_j$, $j = 1(1)[(3n+1)/2]$. Since here $a_i = (-1)^{n-1}\,a_{n-1-i}$, $b_i = (-1)^n\,b_{n-1-i}$, and $c_i = (-1)^{n-1}\,c_{n-1-i}$, actually only $3\,[(n+1)/2] + 2[(3n+1)/2] \sim 4\frac{1}{2}\,n$ distinct quantities are stored, whereas a method corresponding to the preceding $ABC^T$ would require the storage of $9n^2$ numerators and $3n$ l.c.d.'s, the numerators running into double precision for even moderately large $n$. The number of $m-d$'s in (9)–(11), not counting the factor 1/2 in (10), is $(6n+3)[(3n+1)/2] + 1 \sim$ (averaging even with odd $n$) $9n^2 + 6n$. This is the same as the number of $m-d$'s for an $ABC^T$ method,[4] but as noted before, considerably more efficient in operation.

[5] At present there are no tables of explicit polynomial expressions in terms of $1/p$, similar to $A_i^{(n)}(p)$ and $B_i^{(n)}(p)$ in [2].

## References

1. H. E. Salzer, "Tables of Coefficients for the Numerical Calculation of Laplace Transforms," *National Bureau of Standards Applied Mathematics Series* 30, U.S. Government Printing Office, Washington DC, 1953, ii, 36.
2. H. E. Salzer, "Laplace Transforms of Osculatory Interpolation Coefficients," ms. with six oversized pages, copy deposited in the UMT File in *Math. Comput.*
3. H. E. Salzer, *J. Research Nat. Bur. Stands.* **52** (1954), 211.
4. H. E. Salzer, *Quart. J. Mech. Appl. Math.* **12** (1959), 100.
5. A. H. Stroud and D. Secrest, "Gaussian Quadrature Formulas," pp. x, 374, Prentice-Hall, Englewood Cliffs, N.J., 1966.

Received November 18, 1974

Herbert E. Salzer
*941 Washington Avenue*
*Brooklyn, New York 11225*