# Optimum-Point Formulas for Osculatory and Hyperosculatory Interpolation

By Herbert E. Salzer

**Abstract.** Formulas are given for $n$-point osculatory and hyperosculatory (as well as ordinary) polynomial interpolation for $f(x)$, over $(-1, 1)$, in terms of $f(x_i)$, $f'(x_i)$ and $f''(x_i)$ at the irregularly-spaced Chebyshev points $x_i = -\cos \{(2i - 1)\pi/2n\}$, $i = 1, \cdots, n$. The advantage over corresponding formulas for $x_i$ equally spaced is in the squaring and cubing, in the respective osculatory and hyperosculatory formulas, of the approximate ratio of upper bounds for the remainder in ordinary interpolation using Chebyshev and equal spacing (e.g., for $n = 10$, the 15 per cent ratio for ordinary interpolation becoming 2.4 per cent and 0.37 per cent for osculatory and hyperosculatory interpolation). The upper bounds for the remainders in these optimum $n$-point $r$-ply confluent formulas (here $r = 1$ and 2) are around $2^r$ times those of the optimum $\{(r + 1)n\}$-point non-confluent formulas. But these present confluent formulas may require fewer computations for irregular arguments when $f(x)$ satisfies a simple first or second-order differential equation. To facilitate computation, for $n = 2(1)10$, auxiliary quantities $a_i$, $b_i$ and $c_i$, $i = 1, \cdots, n$, independent of $x$, are tabulated exactly or to 15S, not precisely for the optimum points, but for those Chebyshev arguments rounded to 2D ("near-optimum" points). At the very worst ($n = 9$, hyperosculatory) this change about doubles the remainder, which is still less than $(\frac{1}{50})$th of the remainder in the corresponding equally-spaced formula.

**1. Advantage Over Equal-Interval Formulas.** Formulas are given here for $n$-point osculatory and hyperosculatory polynomial interpolation for $f(x)$, from prescribed values of $f(x)$ with its first, or first and second derivatives at the irregularly-spaced Chebyshev points $x_{n-i+1} = \cos \{(2i - 1)\pi/2n\}$, $i = 1, 2, \cdots, n$, instead of equally-spaced points. In this notation, $x_i = -x_{n-i+1}$ and $x_i$ increases with $i$. For the sake of completeness, the ordinary Lagrangian interpolation formulas are also given for these Chebyshev points. All $n$-point ordinary, osculatory and hyperosculatory formulas given here are exact for $f(x)$ a polynomial of degree $n - 1$, $2n - 1$ and $3n - 1$ respectively.

The advantage of Chebyshev-point over equal-interval polynomial interpolation formulas is apparent from the factor $\Pi(x) \equiv \Pi_{i=1}^{n}(x - x_i)$ in the remainder term, which is $\Pi(x)f^{(n)}(\xi)/n!$ for $n$-point ordinary Lagrangian interpolation, $\{\Pi(x)\}^2 f^{(2n)}(\xi)/(2n)!$ for $n$-point osculatory interpolation and $\{\Pi(x)\}^3 f^{(3n)}(\xi)/(3n)!$ for $n$-point hyperosculatory interpolation. At the moment, in order to compare Chebyshev-point with equal-interval formulas, let the range of $x$ be $(-1, 1)$, since the relative improvement of the former over the latter is unchanged under any linear transformation. For $x_i$ at the Chebyshev points, $|\Pi(x)| \leqq (\frac{1}{2})^{n-1}$, which is a fraction of the upper bound of $|\Pi(x)|$ for equally-spaced $x_i$'s. However, that fraction is not impressively small, decreasing rather slowly with increasing $n$ (except

Received August 28, 1961.

SCHEDULE 1: *Upper Bound for Absolute Value of Coefficient of* $f^{(m)}(\xi)$

| $n$ | Ordinary: $m = n$ | | Osculatory: $m = 2n$ | | Hyperosculatory: $m = 3n$ | |
|---|---|---|---|---|---|---|
| | *U.B.* | *Ratio to U.B. for equal spacing* | *U.B.* | *Ratio to U.B. for equal spacing* | *U.B.* | *Ratio to U.B. for equal spacing* |
| 2 | .250 | 50% | .(1)104 | 25% | .(3)174 | $12\frac{1}{2}$% |
| 3 | .(1)417 | 65% | .(4)868 | 42% | .(7)431 | 27% |
| 4 | .(2)521 | 63% | .(6)388 | 40% | .(11)408 | 25% |
| 5 | .(3)521 | 55% | .(8)108 | 30% | .(15)187 | 17% |
| 6 | .(4)434 | 45% | .(11)204 | 20% | .(20)477 | 9.2% |
| 7 | .(5)310 | 36% | .(14)280 | 13% | .(25)747 | 4.5% |
| 8 | .(6)194 | 27% | .(17)292 | 7.6% | .(30)769 | 2.1% |
| 9 | .(7)108 | 21% | .(20)238 | 4.3% | .(35)547 | 0.90% |
| 10 | .(9)538 | 15% | .(23)157 | 2.4% | .(40)281 | 0.37% |

for a slight increase from $n = 2$ to $n = 3$) being somewhat larger than $\frac{1}{9}$ for $n = 11$. Thus ordinary Lagrangian interpolation at Chebyshev points, even for $n = 9$ or 10, gains less than one full decimal place accuracy over interpolation at equally-spaced points. But in the osculatory and hyperosculatory cases, the $\{\Pi(x)\}^2$ and $\{\Pi(x)\}^3$ in the remainder term squares and cubes the relative improvement of the Chebyshev-point formulas. For instance, when $n = 10$ the approximately 15 per cent ratio in the upper bounds of $|\Pi(x)|$ for the Chebyshev and equally-spaced points is now replaced by only around 2 per cent and 0.4 per cent in the ratios of the upper bounds of $\{\Pi(x)\}^2$ and $|\{\Pi(x)\}^3|$ respectively.

In Schedule 1, we give the upper bound for the absolute value of the coefficient of $f^{(m)}(\xi)$, $-1 \leqq \xi \leqq 1$, $m = n$, $2n$ and $3n$, in the remainder term of the $n$-point ordinary, osculatory and hyperosculatory interpolation formulas, for $n = 2(1)10$ to 3S. These bounds are, of course, $1/2^{n-1}n!$, $1/2^{2n-2}(2n)!$ and $1/2^{3n-3}(3n)!$ respectively. Next to each upper bound is the ratio, in per cent, of that quantity to the corresponding upper bound when the $n$ points $x_i$ are equally-spaced over $(-1, 1)$. The quantity in parentheses indicates the number of zeros between the decimal point and the first significant digit.

**2. Comparison with Non-Osculatory Chebyshev-Point Formulas.** The upper bounds for $\{\Pi(x)\}^2$ and $|\{\Pi(x)\}^3|$ in the $n$-point Chebyshev osculatory and hyperosculatory formulas are $(\frac{1}{2})^{2n-2}$ and $(\frac{1}{2})^{3n-3}$ respectively, which is only twice and four times the upper bounds of $(\frac{1}{2})^{2n-1}$ and $(\frac{1}{2})^{3n-1}$ for $|\Pi(x)|$ in the $2n$- and $3n$-point optimum-point (non-confluent) formulas of the same degree of accuracy, namely, for $x_i$ at the zeros of the Chebyshev polynomials $T_{2n}(x) = (\frac{1}{2})^{2n-1} \cos(2n \cos^{-1} x)$ and $T_{3n}(x) = (\frac{1}{2})^{3n-1} \cos(3n \cos^{-1} x)$. This two-and four-ratio is unchanged, of course, under a linear transformation to any range $(a, b)$ other than $(-1, 1)$, because the factor of $\{(b - a)/2\}^{2n}$ or $\{(b - a)/2\}^{3n}$ which then enters the remainder term is the same for both confluent and non-confluent forms of the interpolation formulas.

The confluent Chebyshev-point formulas given here, while not quite as accurate as the non-confluent Chebyshev-point formulas of the same degree, have this advantage: For irregularly-spaced values of $x_i$, it is often less work to compute $n$ values of $y_i \equiv f(x_i)$ together with $y_i' \equiv f'(x_i)$, or with $y_i'$ and $y_i'' \equiv f''(x_i)$, instead of $2n$ or $3n$ values of $y_i$. For instance, in the osculatory case $y = f(x)$ might satisfy a rather simple first-order differential equation $y' = \phi(x, y)$ where it is easier to obtain $n$ values of $y_i' = \phi(x_i, y_i)$ after $y_i$ has been calculated than to compute $n$ more values of $y_i$. The most obvious example is when $\phi(x, y) = y$, where $y = e^x$ and obtaining $y_i' = y_i$ involves no extra work at all. In the hyperosculatory case $y$ might satisfy a simple second-order differential equation from which $y_i''$ is readily obtained from $y_i$ and $y_i'$.

**3. Interpolation Formulas.** We shall not repeat here the derivations of the interpolation formulas, since they have been given a number of times, as well as a full discussion of their advantages, efficient arrangement, remainder terms, extension to inverse and complex interpolation, etc., in previous articles [1]-[3]. In (1)–(14) below, $n$ is understood, $i$ ranges from 1 to $n$, $f \equiv f(x)$, $f_i \equiv f(x_i)$, $f_i' \equiv f'(x_i)$, $f_i'' \equiv f''(x_i)$ and $\sum$ denotes $\sum_{i=1}^{n}$. We employ quantities $p_{ij}$, $q_i$, $r_i$ and $s_i$ given by

$$(1) \qquad \begin{cases} p_{ij} = 1/(x_i - x_j),\, j \neq i; \qquad q_i = \sum_{j=1, j\neq i}^{n} p_{ij}; \\ r_i = q_i^2; \qquad s_i = \sum_{j=1, j\neq i}^{n} p_{ij}^2. \end{cases}$$

For each $n$ we define first

$$(2) \qquad A_i = \prod_{j=1, j\neq i}^{n} p_{ij}.$$

For ordinary interpolation we define

$$(3) \qquad a_i = k_1(n) A_i.$$

For osculatory interpolation we define

$$(4) \qquad \begin{cases} a_i = k_2(n) A_i^2, \\ b_i = -2q_i a_i = -2k_2(n) q_i A_i^2. \end{cases}$$

For hyperosculatory interpolation we define

$$(5) \qquad \begin{cases} a_i = k_3(n) A_i^3, \\ b_i = -3q_i a_i = -3k_3(n) q_i A_i^3, \\ c_i = a_i[\tfrac{9}{2} r_i + \tfrac{3}{2} s_i] = k_3(n)[\tfrac{9}{2} r_i + \tfrac{3}{2} s_i] A_i^3. \end{cases}$$

In (3)–(5), the $k_m(n)$, $m = 1, 2, 3$, denote suitably chosen constants that do not affect the results of the interpolation in formulas (7), (10) and (14), but which might (and this depends upon the values and functional nature of the arguments $x_i$) facilitate appreciably the calculation and use of the auxiliary quantities $a_i$, $a_i$ and $b_i$, or $a_i$, $b_i$ and $c_i$ in (6)–(14).

For ordinary $n$-point interpolation, of $(n - 1)$th degree accuracy, we obtain

$$(6) \qquad \alpha_i = a_i/(x - x_i), \text{ from which}$$

$$(7) \qquad f \sim \Sigma \alpha_i f_i / \Sigma \alpha_i.$$

For $n$-point polynomial osculatory interpolation of $(2n - 1)$th degree accuracy, we obtain

$$\beta_i = a_i/(x - x_i), \tag{8}$$

$$\alpha_i = (\beta_i + b_i)/(x - x_i), \quad \text{from which} \tag{9}$$

$$f \sim \Sigma(\alpha_i f_i + \beta_i f_i')/\Sigma\alpha_i . \tag{10}$$

For $n$-point polynomial hyperosculatory interpolation of $(3n - 1)$th degree accuracy, we obtain

$$\gamma_i = a_i/2(x - x_i), \tag{11}$$

$$\beta_i = (2\gamma_i + b_i)/(x - x_i), \tag{12}$$

$$\alpha_i = (\beta_i + c_i)/(x - x_i), \quad \text{from which} \tag{13}$$

$$f \sim \Sigma(\alpha_i f_i + \beta_i f_i' + \gamma_i f_i'')/\Sigma\alpha_i . \tag{14}$$

**4. Use of "Near-Optimum" Points.** Instead of taking the $x_i$ precisely equal to the zeros of $T_n(x)$, we now round them off to two decimal places. This makes the osculatory and hyperosculatory formulas "near-optimum" rather than "optimum" point formulas. Three reasons for such a choice are: 1) easier calculation and checking of the table of the auxiliary quantities $a_i$, $b_i$ and $c_i$ occurring in the interpolation formulas (7), (10) and (14); 2) some of the $a_i$, for the lower values of $n$, can be given exactly with much fewer than 15 significant figures; 3) for many functions $f(x)$, it is less work to calculate $f(x_i)$ when $x_i$ is an exact two-decimal argument.

The employment of rounded-off zeros of $T_n(x)$ as the arguments $x_i$ was suggested by Lanczos's use of rounded zeros of Legendre polynomials for a modification of Gaussian quadrature. [4] In this present case, the slight shift in the $x_i$ from exact to rounded Chebyshev points does not produce too great a change in the upper bound for the remainder, (the changes for $n = 7$ and $n = 9$ being appreciably greater than the rest, as seen in Schedule 2). This justifies the terminology "near-optimum", which contrasts sharply with the experience of Lanczos with rounded Gaussian points for quadrature formulas. Thus, quoting his comment on an example [4, p. 410]: "Compared with the Gaussian error, the error has increased by the factor 71, which shows the great sensitivity of the Gaussian method to even small shifts of the zeros." Here, at the worst, for 9-point hyperosculatory interpolation, the choice of the near-optimum instead of optimum points causes the maximum error to be slightly more than doubled. But even then it is less than $(\frac{1}{50})$th of the maximum error in the corresponding equally-spaced formula.

In attempting to estimate the sensitivity in the upper bound of the absolute value of $\Pi(x) = T_n(x)$ for a slight change of $\Delta x_i$ in every $x_i$, we differentiate $T_n(x) = \Pi_{i=1}^{n}(x - x_i)$ partially with respect to each $x_i$, obtaining for $D_n(x)$, the dominant part of the deviation in $\Pi(x)$, the expression

$$D_n(x) = -\sum_{i=1}^{n} \frac{\Pi_{i=1}^{n}(x - x_i)}{x - x_i} \Delta x_i , \tag{15}$$

Tables of $a_i$, $b_i$ and $c_i$

| | | | Ordinary Interpolation | Osculatory Interpolation | |
|---|---|---|---|---|---|
| $n$ | $i$ | $x_i$ | $a_i$ | $a_i$ | $b_i$ |
| 2 | 1, 2 | ∓0.71 | ∓1 | 1 | ±1.40845 07042 2535 |
| 3 | 1, 3 | ∓0.87 | 1 | 1 | ±3.44827 58620 6897 |
| 3 | 2 | 0 | −2 | 4 | 0 |
| 4 | 1, 4 | ∓0.92 | ∓1.9 | 0.361 | ±2.28481 29567 6948 |
| 4 | 2, 3 | ∓0.38 | ±4.6 | 2.116 | ±0.98676 86309 79157 |
| 5 | 1, 5 | ∓0.95 | 0.3481 | 0.12117 361 | ±1.21320 85521 6070 |
| 5 | 2, 4 | ∓0.59 | −0.9025 | 0.81450 625 | ±0.67432 28058 42933 |
| 5 | 3 | 0 | 1.1088 | 1.22943 744 | 0 |
| 6 | 1, 6 | ∓0.97 | ∓1.5 | 0.225 | ±3.23024 16119 5097 |
| 6 | 2, 5 | ∓0.71 | ±4.1 | 1.681 | ±2.37511 73717 7847 |
| 6 | 3, 4 | ∓0.26 | ∓5.6 | 3.136 | ±0.85512 42401 72493 |
| 7 | 1, 7 | ∓0.97 | 0.37810 201 | 0.14296 11299 66040 | ±2.84410 18250 1904 |
| 7 | 2, 6 | ∓0.78 | −1.04383 446 | 1.08959 03798 8349 | ±1.99381 25356 4272 |
| 7 | 3, 5 | ∓0.43 | 1.51061 495 | 2.28195 75271 6350 | ±1.46095 20152 4591 |
| 7 | 4 | 0 | −1.68976 500 | 2.85530 57552 2500 | 0 |
| 8 | 1, 8 | ∓0.98 | ∓0.97536 56688 | 0.09513 38187 87367 1 | ±2.45239 17122 6555 |
| 8 | 2, 7 | ∓0.83 | ±2.81517 71648 | 0.79252 22469 21137 | ±2.32927 47108 1650 |
| 8 | 3, 6 | ∓0.56 | ∓4.15391 20805 | 1.72549 85572 5238 | ±0.93364 45598 54158 |
| 8 | 4, 5 | ∓0.20 | ±4.72726 03686 | 2.23469 90592 5362 | ∓0.05814 57965 08793 1 |
| 9 | 1, 9 | ∓0.98 | 0.46316 76707 68 | 0.21452 42912 44654 | ±7.31124 44403 2311 |
| 9 | 2, 8 | ∓0.87 | −1.22781 71566 08 | 1.50753 49700 6095 | ±2.70477 66013 7086 |
| 9 | 3, 7 | ∓0.64 | 1.82850 70803 23 | 3.34343 81427 9134 | ±4.60073 50158 6742 |
| 9 | 4, 6 | ∓0.34 | −2.28761 18102 88 | 5.23316 77945 6914 | ±2.44461 28094 1959 |
| 9 | 5 | 0 | 2.44750 84316 10 | 5.99029 75228 0204 | 0 |
| 10 | 1, 10 | ∓0.99 | ∓0.41223 53180 2154 | 0.16993 79574 24320 | ±6.73542 63847 8561 |
| 10 | 2, 9 | ∓0.89 | ±1.24470 77696 4339 | 1.54929 74318 1062 | ±8.10439 99566 8827 |
| 10 | 3, 8 | ∓0.71 | ∓1.92977 33728 6557 | 3.72402 52706 2096 | ±3.47307 39064 2641 |
| 10 | 4, 7 | ∓0.45 | ±2.46258 91833 0950 | 6.06434 54857 5295 | ±6.43767 06135 5020 |
| 10 | 5, 6 | ∓0.16 | ∓2.73564 36743 5008 | 7.48374 63130 1161 | ∓1.57793 67089 3871 |

Hyperosculatory Interpolation

| $n$ | $i$ | $x_i$ | $a_i$ | $b_i$ | $c_i$ |
|---|---|---|---|---|---|
| 2 | 1, 2 | ∓0.71 | ∓1 | −2.11267 60563 3803 | ∓2.97560 00793 4934 |
| 3 | 1, 3 | ∓0.87 | 1 | ±5.17241 37931 0345 | 15.85414 18945 700 |
| 3 | 2 | 0 | −8 | 0 | −31.70828 37891 399 |
| 4 | 1, 4 | ∓0.92 | ∓0.6859 | −6.51171 69267 9301 | ∓35.35105 61593 570 |
| 4 | 2, 3 | ∓0.38 | ±9.7336 | 6.80870 35537 5619 | ±86.36831 12988 727 |
| 5 | 1, 5 | ∓0.95 | 0.04218 05336 41 | ±0.63347 68455 10712 | 5.35936 14929 5057 |
| 5 | 2, 4 | ∓0.59 | −0.73509 18906 25 | ∓0.91286 44984 09870 | −13.49924 33461 453 |
| 5 | 3 | 0 | 1.36320 02334 72 | 0 | 16.27976 37063 895 |
| 6 | 1, 6 | ∓0.97 | ∓0.03375 | −0.72680 43626 88967 | ∓8.74001 29929 3812 |
| 6 | 2, 5 | ∓0.71 | ±0.68921 | 1.46069 71836 4376 | ±23.92400 98678 306 |
| 6 | 3, 4 | ∓0.26 | ∓1.75616 | −0.71830 43617 44894 | ∓32.66402 28607 687 |
| 7 | 1, 7 | ∓0.97 | 0.00540 53890 59203 10 | ±0.16130 40925 02655 | 2.67672 79073 0026 |
| 7 | 2, 6 | ∓0.78 | −0.11373 51985 80688 | ∓0.31218 15347 22577 | −7.06970 10338 4325 |
| 7 | 3, 5 | ∓0.43 | 0.34471 59155 79822 | ±0.33104 03933 19465 | 10.26580 84638 704 |
| 7 | 4 | 0 | −0.48247 95729 47777 | 0 | −11.74567 06746 548 |

TABLES of $a_i$, $b_i$ and $c_i$—(*Continued*)

Hyperosculatory Interpolation

| $n$ | $i$ | $x_i$ | $a_i$ | $b_i$ | $c_i$ |
|---|---|---|---|---|---|
| 8 | 1, 8 | ∓0.98 | ∓0.00927 90260 78703 83 | −0.35879 68023 89020 | ∓7.68100 77077 0057 |
| 8 | 2, 7 | ∓0.83 | ±0.22310 90532 12837 | 0.98359 81464 65512 | ±23.18826 27433 073 |
| 8 | 3, 6 | ∓0.56 | ∓0.71675 69301 85600 | −0.58174 16124 10694 | ∓33.10310 66120 670 |
| 8 | 4, 5 | ∓0.20 | ±1.05640 04298 5573 | −0.04123 05479 15504 7 | ±34.10349 18155 463 |
| 9 | 1, 9 | ∓0.98 | 0.00993 60716 29894 27 | ±0.50794 98086 75992 | 14.41869 83299 256 |
| 9 | 2, 8 | ∓0.87 | −0.18509 77300 42737 | ∓0.49814 56673 93253 | −30.70431 01740 749 |
| 9 | 3, 7 | ∓0.64 | 0.61135 00316 71595 | ±1.26187 14826 8053 | 41.26425 87109 550 |
| 9 | 4, 6 | ∓0.34 | −1.19714 56452 0752 | ∓0.83884 87701 61438 | −54.56736 40851 665 |
| 9 | 5 | 0 | 1.46613 03694 9105 | 0 | 59.17743 44367 215 |
| 10 | 1, 10 | ∓0.99 | ∓0.00700 54427 92274 56 | −0.41648 70956 61415 | ∓13.63892 25685 320 |
| 10 | 2, 9 | ∓0.89 | ±0.19284 22550 86323 | 1.51314 14391 5812 | ±46.53720 81301 798 |
| 10 | 3, 8 | ∓0.71 | ∓0.71865 24807 12282 | −1.00533 68319 9238 | ∓70.78787 62690 287 |
| 10 | 4, 7 | ∓0.45 | ±1.49339 91597 0670 | 2.37800 07027 9572 | ±93.69814 23293 455 |
| 10 | 5, 6 | ∓0.16 | ∓2.04728 63261 6309 | 0.64750 08864 49945 | ∓104.39709 18091 15 |

so that

$$|D_n(x)| \leqq 2^{-n+1} \sum_{i=1}^{n} \frac{|\Delta x_i|}{|x - x_i|}. \tag{16}$$

Now for $x$ in the neighborhood of the extrema of $T_n(x)$ not close to the ends $\pm 1$, the $|x - x_i|$ stays large enough for (16) to furnish upper bounds for $|D_n(x)|/2^{-n+1}$ of the order of just several per cent when $\Delta x_i$ is the roundoff error in employing $x_i$ to 2D. However (16) breaks down as a practical formula, for larger $n$ and $x$ either at $\pm 1$ or at an extremum close to $\pm 1$ since there $|x - x_i|$ is quite small. This might also be expected from the very large derivative of $2^{n-1}T_n(x)$ at $x = \pm 1$, its magnitude being $n^2$. Thus, to be on the safe side, to provide for every $x$ in the range $(-1, 1)$, instead of using (15) or (16), the factor $\Pi(x) = \Pi_{i=1}^{n}(x - x_i)$ for the chosen near-optimum $x_i$'s was calculated for every $n$ from 2 to 10, for $x = -1(.001)1$, and its greatest deviation from zero was found. The percentage increase in the upper bound for the absolute value of the coefficient of $f^{(m)}(\xi)$ (see Schedule 1), due to the use of these near-optimum points $x_i$ instead of optimum points, is given in Schedule 2.

SCHEDULE 2: *Increase in Schedule 1 When Using Near-Optimum Points*

| $n$ | Ordinary | Osculatory | Hyperosculatory |
|---|---|---|---|
| 2 | 0.82% | 1.65% | 2.5% |
| 3 | 1.4% | 2.8% | 4.2% |
| 4 | 5.1% | 10.5% | 16% |
| 5 | 1.7% | 3.4% | 5.2% |
| 6 | 2.6% | 5.3% | 8.0% |
| 7 | 21% | 46% | 76% |
| 8 | 6.2% | 13% | 20% |
| 9 | 29% | 66% | 113% |
| 10 | 7.6% | 16% | 25% |

**5. Tables of Auxiliary Coefficients $a_i$, $b_i$ *and* $c_i$.** To facilitate the use of (6)–(14) for these near-optimum points $x_i$, the auxiliary quantities $a_i$, $b_i$ and $c_i$ are tabulated here for $n = 2(1)10$, $i = 1, \cdots, n$. It reduced the work considerably to choose the constants $k_m(n)$, $m = 1, 2, 3$, in (3)–(5), as products of powers of selected prime numbers < 200. As a result of this choice, it was easy to give exact values of all the quantities $a_i$ for ordinary interpolation, and of $a_i$ for $n = 2(1)6$ for osculatory and hyperosculatory interpolation. The remaining quantities $a_i$ and all quantities $b_i$ and $c_i$ are given to 15S, believed to be correct to within a unit in the last place. In reading entries prefixed by $\pm$ or $\mp$ signs, the upper sign corresponds to the negative $x_i$.

General Dynamics/Astronautics
San Diego, California

1. H. E. Salzer, "New formulas for facilitating osculatory interpolation," *J. Res. Nat. Bur. Standards*, v. 52, 1954, p. 211–216.
2. H. E. Salzer, "Formulae for hyperosculatory interpolation, direct and inverse," *Quart. J. Mech. Appl. Math.*, v. 12, 1959, p. 100–110.
3. H. E. Salzer, "Alternative formulas for osculatory and hyperosculatory inverse interpolation," *Math. Comp.*, v. 14, 1960, p. 257–261.
4. C. Lanczos, *Applied Analysis*, Prentice Hall, Englewood Cliffs, New Jersey, 1956, p. 408–410, table on p. 529.