BIT 32 (1992), 464–480.

NEW ALGORITHMS FOR POLYNOMIAL AND TRIGONOMETRIC INTERPOLATION ON PARALLEL COMPUTERS

ILAN BAR-ON and AVRAM SIDI

Department of Computer Science, Technion, Haifa 32000, Israel

Abstract.

An interpolation polynomial of order N is constructed from p independent subpolynomials of order $n \sim N/p$. Each such subpolynomial is found independently and in parallel. Moreover, evaluation of the polynomial at any given point is done independently and in parallel, except for a final step of summation of p elements. Hence, the algorithm has almost no communication overhead and can be implemented easily on any parallel computer. We give examples of finite-difference interpolation, trigonometric interpolation, and Chebyshev interpolation, and conclude with the general Hermite interpolation problem.

Categories and Subject Descriptor (CR): G.1.0, G.1.1.

AMS Subject Classifications (1985): 65D05, 65W05, 68C25.

Key words: Parallel algorithms, polynomial interpolation, trigonometric interpolation, Chebyshev interpolation, the general Hermite interpolation.

1. Introduction.

In this paper we provide new formulas and algorithms for polynomial and trigonometric interpolation that are especially useful for vector and parallel machines. Unlike previous works on this subject, we do not consider ways of parallelizing the classical methods such as Lagrange, Newton, Finite difference etc., for polynomial interpolation, and FFT methods for trigonometric and Chebyshev interpolation [5, 8]. Such works, most recently by Eğecioğlu, Gallopoulos and Koç [3, 2] for the parallel construction of the Newton and Hermite formulas, Reif [7] for the construction of the Lagrange formula, Munro and Paterson [6] and Dowling [1] for parallel evaluation, are inefficient for large parallel systems because of their high communication overhead. Our new formulas on the other hand require almost no communication.

Received December 1990. Revised June 1991 and October 1991.

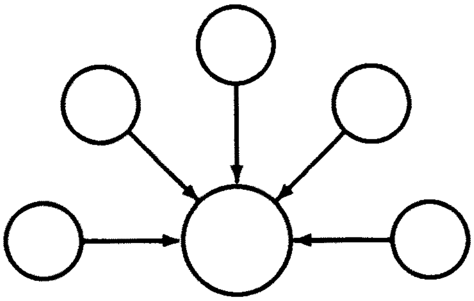

Fig. 1. A star architecture.

Given a set of interpolation points, we partition them into smaller sets of points, and accordingly to smaller independent interpolation problems. That is, given p processors, we break our interpolation problem into p smaller independent interpolation problems similar in kind to the original one. The corresponding subpolynomials are then found in parallel with no communication overhead. For that purpose, each processor, operating sequentially, may use the best numerical and computational software tools that exist. There is no need for all processors to use the same interpolation method or the same number of points, although it is recommended that they all solve their subproblems in approximately the same time. To evaluate the polynomial we combine the values of the corresponding subpolynomials. We observe that in case $p = 1$ we may use any sequential method, whereas in case $p = N$, where N equals the degree of the interpolation problem, we get the Lagrange method. Hence, our algorithm adapts itself to the number of processors available.

We assume a star-shape model of computation, see figure 1. Here, there is one main processor with direct communication channels with the rest of the processing elements (PE's) and therefore with insignificant communication delay. For example, let $N = np$, then each PE computes in parallel the function values at its interpolation points and constructs the corresponding subpolynomial. Given a point for evaluation, each PE evaluates its subpolynomial and sends the result to the main processor which combines the individual results suitably. Taking $p \ll N$ the communication overhead will contribute little to the overall complexity of the algorithm, and we obtain an efficient and practical algorithm.

In Section 2 we present our new mathematical formulas for the interpolation polynomial as well as for the modified barycentric form. We then obtain specific formulas and algorithms for the finite difference interpolation problem in Section 3, for the trigonometric interpolation problem in Section 4, for the Chebyshev interpolation problem in Section 5, and for the general Hermite interpolation problem in Section 6.

2. The interpolation polynomial.

Let $X = \{x_0, x_1, \ldots, x_N\}$ be a set of given distinct points in the interval $[a, b]$, and let $f(x)$ be a function defined on $[a, b]$, whose values $f_j \equiv f(x_j)$, $j = 0, 1, \ldots, N$, are given. We are interested in constructing a representation of the polynomial $P(x)$ of degree at most N that interpolates $f(x)$ on X and is most suitable for parallel computation.

Let $\{X_1, X_2, \ldots, X_p\}$ be a partition of X, i.e.,

$$X = \cup_1^p X_i \quad \text{and} \quad X_i \cap X_j = \emptyset \quad \text{for} \quad i \neq j. \tag{1}$$

The following theorem indicates how $P(x)$ can be constructed independently and in parallel by p processors, each solving a smaller interpolation problem on one of the subsets X_i.

THEOREM 2.1. *For $i = 1, \ldots, p$, define*

$$w_{i,j}^{-1} = \prod_{x_k \notin X_i} (x_j - x_k), \qquad x_j \in X_i, \tag{2}$$

and let $Q_i(x)$ be the polynomial of degree at most $|X_i| - 1$ that satisfies the following interpolation conditions:

$$Q_i(x_j) = w_{i,j} f_j, \qquad x_j \in X_i. \tag{3}$$

Then $P(x)$, the interpolation polynomial on X, is given by

$$P(x) = \sum_{i=1}^{p} Q_i(x) \prod_{x_k \notin X_i} (x - x_k). \tag{4}$$

PROOF. First, it is clear that the right hand side of (4) is a sum of polynomials, each of degree at most N. Next, with the help of (2) and (3), it can be shown that $P(x)$ satisfies the interpolation conditions

$$P(x_j) = f(x_j) \quad \text{for all} \quad j. \tag{5}$$

The result now follows from the uniqueness of $P(x)$. ■

It is known [4, 9, 10] that the barycentric representation for Lagrange interpolation enjoys a large degree of numerical stability. Therefore we look for a generalization of the barycentric representation that is appropriate for the formula given in (4). Precisely this is achieved in Theorem 2.2 below.

THEOREM 2.2 *Let $Q_i(x)$ be as in Theorem* (2.1), *and let $R_i(x)$, $i = 1, \ldots, p$, be the polynomial of degree at most $|X_i| - 1$ that satisfies the interpolation conditions*

$$R_i(x_j) = w_{i,j}, \qquad x_j \in X_i. \tag{6}$$

Then $P(x)$ can be expressed in the form

$$P(x) = \frac{\sum_{i=1}^{p} Q_i(x)/\prod_{x_k \in X_i}(x - x_k)}{\sum_{i=1}^{p} R_i(x)/\prod_{x_k \in X_i}(x - x_k)} \equiv \frac{\sum_{i=1}^{p} \phi_i(x)}{\sum_{i=1}^{p} \psi_i(x)}. \tag{7}$$

PROOF. Comparing (3) and (6), and employing (4), we see that

$$1 \equiv \sum_{i=1}^{p} R_i(x) \prod_{x_k \notin X_i}(x - x_k) \quad \text{for all} \quad x, \tag{8}$$

which we rewrite in the form

$$1 \equiv \prod_{x_k \in X}(x - x_k) \sum_{i=1}^{p} R_i(x)/\prod_{x_k \in X_i}(x - x_k). \tag{9}$$

Similarly,

$$P(x) = \prod_{x_k \in X}(x - x_k) \sum_{i=1}^{p} Q_i(x)/\prod_{x_k \in X_i}(x - x_k). \tag{10}$$

The result now follows by dividing (10) by (9). ■

Given p processors, we assign processor i ($i = 1, \ldots, p$), to computing the corresponding terms $\phi_i(x)$ and $\psi_i(x)$. The computation of $w_{i,j}$ takes $O(n_i(N - n_i))$ additions and multiplications in the worst case, where $n_i = |X_i|$. However, as will be seen in the following sections, in many cases of interest these values can be computed analytically in much fewer operations. Assuming that $w_{i,j}$ are known, and that $n_i \sim n \sim N/p$, $i = 1, \ldots, p$, each processor is faced with an interpolation problem of order n that can be solved simultaneously without any need of interprocessor communication. Once the interpolation polynomial is known, its value at points not in the set is given by summing and dividing the corresponding subvalues in (7).

In Sections 3, 4, 5 we consider the problems of finite difference interpolation, trigonometric interpolation, and Chebyshev interpolation. For ease of representation we will use a slightly different notation as follows: We assume that the function $f(x)$ is given at $N = np$ distinct points and that each of the p subsets X_i, which are now numbered $i = 0, \ldots, p - 1$, contains exactly n points. We denote the points in the subset X_i by $x_{i,j}$, $j = 0, \ldots, n - 1$.

3. Finite difference interpolation.

Let X be a set of equally spaced points in the interval $[a, b]$,

$$x_i = a + ih, \quad i = 0, 1, \ldots, N - 1, \quad h = (b - a)/(N - 1), \tag{11}$$

where we assume for simplicity that $N = np$ and p is the number of processors available. We here consider two partitions.

In the first partition we assign the ith group of n consecutive points to the ith processor, i.e.,

$$X_i = \{x_{i,j} = x_{in+j},\ \ j = 0,\dots,n-1\}, \qquad i = 0,\dots,p-1. \tag{12}$$

Hence,

$$\begin{aligned} w_{i,j}^{-1} &= \prod_{k=0}^{in-1} (x_{in+j} - x_k) \prod_{k=(i+1)n}^{N-1} (x_{in+j} - x_k) \\ &= C_1(-1)^{in}\binom{n-1}{j}\Big/\binom{N-1}{in+j}, \end{aligned} \tag{13}$$

where $C_1 = (-h)^{N-n}(N-1)!/(n-1)!$ is independent of i and j. We note that if the same constant C multiplies all $w_{i,j}$, it follows from (3), (6) that the subpolynomials $Q_i(x)$ and $R_i(x)$ are also multiplied by the same constant C but the interpolation polynomial $P(x)$ remains invariant by (7). In view of this, each processor has to compute the corresponding polynomials $Q_i(x)$ and $R_i(x)$ that satisfy the interpolation conditions

$$Q_i(x_{i,j}) = (-1)^{in} f_{in+j}\binom{N-1}{in+j}\Big/\binom{n-1}{j}, \tag{14}$$

$$R_i(x_{i,j}) = (-1)^{in}\binom{N-1}{in+j}\Big/\binom{n-1}{j}, \tag{15}$$

for $j = 0,\dots,n-1$, and this computation can be carried out using any finite difference formula.

In the second partition the subsets X_i are formed according to

$$X_i = \{x_{i,j} = x_{i+jp},\ \ j = 0,\dots,n-1\}, \qquad i = 0,\dots,p-1. \tag{16}$$

Consequently,

$$w_{i,j}^{-1} = \frac{\prod_{k=0}^{i+jp-1} (x_{i+jp} - x_k) \prod_{k=i+jp+1}^{N-1} (x_{i+jp} - x_k)}{\prod_{l=0}^{j-1} (x_{i+jp} - x_{i+lp}) \prod_{l=j+1}^{n-1} (x_{i+jp} - x_{i+lp})} \tag{17}$$

$$= C_2(-1)^{i+j(p-1)}\binom{n-1}{j}\Big/\binom{N-1}{i+jp} \tag{18}$$

where $C_2 = C_1/p^{n-1}$ is independent of i and j. Each processor has to compute the corresponding polynomials $Q_i(x)$ and $R_i(x)$ that satisfy the interpolation conditions

$$Q_i(x_{i,j}) = (-1)^{i+j(p-1)} f_{i+jp}\binom{N-1}{i+jp}\Big/\binom{n-1}{j}, \tag{19}$$

$$R_i(x_{i,j}) = (-1)^{i+j(p-1)}\binom{N-1}{i+jp}\Big/\binom{n-1}{j}, \tag{20}$$

for $j = 0,\dots,n-1$. As before, this computation can be carried out using any finite difference formula.

The table below gives rough estimates on the total operation count for $p \ll N$:

	sequential	parallel	speed-up
Construction	$(N-1)N/2$	$(N-n)n$	$\sim p/2$
Evaluation	N	$2(n+p)$	$\sim p/2$

We point out that $w_{i,j}$, being dependent only on N and p, can be computed in advance for a wide range of applications. The corresponding construction phase is then reduced to n^2, a saving by a factor p.

4. Trigonometric interpolation.

Let θ_j, $j = 0,1,\dots,N-1$, be equally spaced points in $[0,2\pi]$ given by

$$\theta_j = 2\pi j/N, \qquad j = 0,1,\dots,N-1, \tag{21}$$

and let $f(\theta)$ be a function defined on $[0,2\pi]$ whose values $f_j \equiv f(\theta_j)$, $j = 0,1,\dots,N-1$, are given. Furthermore, let $N = 2M$. Then there exists a unique balanced trigonometric polynomial $T(\theta)$ of degree M,

$$T(\theta) = \tfrac{1}{2}a_0 + \sum_{k=1}^{M-1} (a_k \cos k\theta + b_k \sin k\theta) + \tfrac{1}{2}a_M \cos M\theta, \tag{22}$$

interpolating $f(\theta)$ at the points θ_j, $j = 0,1,\dots,N-1$, see [4]. A complex interpretation of $T(\theta)$ in terms of the variable $z = e^{i\theta}$ yields the balanced complex trigonometric polynomial $P(z) \equiv T(\theta)$ of degree M,

$$P(z) = \tfrac{1}{2}c_{-M}z^{-M} + \sum_{k=-M+1}^{M-1} c_k z^k + \tfrac{1}{2}c_M z^M, \quad c_{-M} = c_M, \tag{23}$$

whose coefficients c_k are related to the a_k and b_k in (22) through

$$a_k = c_k + c_{-k}, \quad b_k = i(c_k - c_{-k}), \quad k = 0,1,\dots,M. \tag{24}$$

Of course, $P(z)$ satisfies the interpolation conditions

$$P(z_j) = T(\theta_j) = f_j, \qquad j = 0,1,\dots,N-1, \tag{25}$$

where z_j are given by

$$z_j = z_1^j, \quad j = 0,1,\dots,N-1, \quad \text{with} \quad z_1 = e^{2\pi i/N}. \tag{26}$$

The coefficients c_l of $P(z)$ can be computed from

$$c_l = \frac{1}{N}\sum_{k=0}^{N-1} f_k z_k^{-l}, \qquad l = -M, -M+1, \ldots, M, \tag{27}$$

and this, with the help of the FFT, can be done in $O(N \log N)$ operations. We note that the parallel FFT algorithm requires approximately $O(\log p)$ steps of interprocessor communication, and further, that the resulting polynomial needs further processing for parallel evaluation. In this section we introduce a new representation for $P(z)$ that can be computed on a parallel computer with almost no interprocessor communication. Specifically, we divide the original trigonometric interpolation problem into p trigonometric interpolation problems of smaller size, each of which is of the same type as the original problem. Furthermore, the FFT can be employed in each of these problems.

Let $N = np$ with $n = 2m$, and consider the partition $\{Z_0, Z_1, \ldots, Z_{p-1}\}$ of the set of points $Z = \{z_0, z_1, \ldots, z_{N-1}\}$, where

$$Z_l = \{z_{l,r} = z_{l+rp} = z_l z_p^r, \quad r = 0, 1, \ldots, n-1\}, \qquad l = 0, 1, \ldots, p-1. \tag{28}$$

THEOREM 4.1. *For $l = 0, 1, \ldots, p-1$, define*

$$w_{l,r}^{-1} = z_{l,r}^{-M+m} \prod_{k \neq l} (z_{l,r} - z_{k,t}), \quad r = 0, 1, \ldots, n-1, \tag{29}$$

and let $Q_l(s)$ be the balanced complex trigonometric polynomial of degree m that satisfies the interpolation conditions

$$Q_l(z_p^r) = f_{l+rp} w_{l,r}, \qquad r = 0, 1, \ldots, n-1, \tag{30}$$

on the subset of points Z_0. Then $P(z)$, the balanced trigonometric polynomial that satisfies the interpolation conditions in (25), can be expressed in the form

$$P(z) = z^{-M+m} \sum_{l=0}^{p-1} Q_l(z/z_l) \prod_{k \neq l} (z - z_{k,t}). \tag{31}$$

PROOF. First, it is clear from (29) and (30) that $P(z)$ in (31) satisfies the interpolation conditions in (25). Next, by (23), $Q_l(s)$ is of the form

$$Q_l(s) = \tfrac{1}{2} d_{-m} s^{-m} + \sum_{k=-m+1}^{m-1} d_k s^k + \tfrac{1}{2} d_m s^m, \qquad d_{-m} = d_m. \tag{32}$$

Also

$$z^{-M+m} \prod_{k \neq l} (z - z_{k,t}) = \sum_{k=-M+m}^{M-m} g_k z^k \tag{33}$$

with

$$g_{-M+m} = \prod_{k \neq l} (-z_{k,t}) = \frac{\prod_{j=0}^{N-1}(-z_j)}{\prod_{r=0}^{n-1}(-z_{l,r})} = z_l^{-n} \quad \text{and} \quad g_{M-m} = 1. \tag{34}$$

The first result in (34) follows from the fact that z_j are Nth roots of unity, while the z_p^r are the nth roots of unity. Substituting (32) in (31), and using (33) and (34), we see that each of the terms $z^{-M+m}Q_l(z/z_l)\prod_{k\neq l}(z-z_{k,t})$ in (31) is of the form $\sum_{k=-M}^{M} h_k z^k$ with

(35) $$h_{-M} = \tfrac{1}{2}d_{-m}z_l^m g_{-M+m} = \tfrac{1}{2}d_{-m}z_l^{-m} = \tfrac{1}{2}d_m z_l^{-m} g_{M-m} = h_M,$$

ensuring that $P(z)$ is balanced. The rest follows from the uniqueness of $P(z)$. ■

As before, we look for a generalized barycentric formula. This formula is developed in Theorem 4.2 below.

THEOREM 4.2. *For $l = 0, 1, \ldots, p-1$, let $\hat{Q}_l(s)$, be the balanced complex trigonometric polynomial of degree m that satisfies the interpolation conditions*

(36) $$\hat{Q}_l(z_p^r) = (-1)^{r(p-1)} f_{l+rp}, \quad r = 0, 1, \ldots, n-1,$$

on the subset of points Z_0. Then the balanced trigonometric interpolation polynomial $P(z)$ of Theorem (4.1) *can be expressed in the form*

(37) $$P(z) = \frac{\sum_{l=0}^{p-1}(-1)^l z_l^{-m}\hat{Q}_l(z/z_l)/((z/z_l)^n - 1)}{\sum_{l=0}^{p-1}(-1)^l z_l^{-m}\hat{R}(z/z_l)/((z/z_l)^n - 1)},$$

where, depending on whether p is even or odd, $\hat{R}(s)$ assumes the simple forms

(38) $$\hat{R}(s) = \begin{cases} 1 & \text{if } p \text{ is odd} \\ \tfrac{1}{2}(s^m + s^{-m}) & \text{if } p \text{ is even.} \end{cases}$$

PROOF. For $l = 0, 1, \ldots, p-1$, we let $R_l(s)$ be the balanced complex trigonometric polynomial of degree m that satisfies the interpolation conditions

(39) $$R_l(z_p^r) = w_{l,r}, \qquad r = 0, 1, \ldots, n-1,$$

on the subset of points Z_0. Comparing (39) with (30), it is obvious from Theorem 4.1 that

(40) $$1 \equiv z^{-M+m}\sum_{l=0}^{p-1} R_l(z/z_l)\prod_{k\neq l}(z - z_{k,t}).$$

Dividing now (31) by (40), we obtain the barycentric formula for $P(z)$

(41) $$P(z) = \frac{\sum_{l=0}^{p-1} Q_l(z/z_l)/\prod_{r=0}^{n-1}(z - z_{l,r})}{\sum_{l=0}^{p-1} R_l(z/z_l)/\prod_{r=0}^{n-1}(z - z_{l,r})}.$$

Next, we observe that

(42) $$\begin{aligned}\prod_{k\neq l}(z_{l,r} - z_{k,t}) &= \prod_{k,t\neq l,r}(z_{l,r} - z_{k,t})/\prod_{t\neq r}(z_{l,r} - z_{l,t}) \\ &= (Nz_{l,r}^{N-1})/(nz_{l,r}^{n-1}) = pz_{l,r}^{-n},\end{aligned}$$

so that (29) becomes

$$w_{l,r}^{-1} = pz_{l,r}^{-M-m} = p(-1)^{l+r(p-1)}z_l^{-m}. \tag{43}$$

Furthermore,

$$\prod_{r=0}^{n-1}(z - z_{l,r}) = z_l^n \prod_{r=0}^{n-1}(z/z_l - z_p^r) = z_l^n((z/z_l)^n - 1). \tag{44}$$

Comparing (36) with (30), and invoking (43), we see that

$$Q_l(s) = p^{-1}(-1)^l z_l^m \hat{Q}_l(s), \qquad l = 0, 1, \ldots, p-1. \tag{45}$$

Similarly, if we define $\hat{R}(s)$ to be the balanced complex trigonometric polynomial that satisfies the interpolation conditions

$$\hat{R}(z_p^r) = (-1)^{r(p-1)}, \qquad r = 0, 1, \ldots, n-1, \tag{46}$$

then

$$R_l(s) = p^{-1}(-1)^l z_l^m \hat{R}(s), \qquad l = 0, 1, \ldots, p-1. \tag{47}$$

Combining (44), (45), and (47) in (41), we obtain (37). Finally, (38) can be seen to hold by inspection. ■

Now that we have obtained the barycentric form of $P(z)$, we can obtain that of $T(\theta)$, the real form of $P(z)$, very easily as follows:

$$T(\theta) = \frac{\sum_{l=0}^{p-1}(-1)^l \hat{U}_l(\theta - \theta_l)/\sin m(\theta - \theta_l)}{\sum_{l=0}^{p-1}(-1)^l \hat{V}_l(\theta - \theta_l)/\sin m(\theta - \theta_l)}, \tag{48}$$

where $\hat{U}_l(\phi) \equiv \hat{Q}_l(s)$, with $s = e^{i\phi}$, is the balanced trigonometric polynomial of degree m that satisfies the interpolation conditions

$$\hat{U}_l(\theta_{pr}) = (-1)^{r(p-1)} f_{l+rp}, \qquad r = 0, 1, \ldots, n-1, \tag{49}$$

and $\hat{V}(\phi)$ is given by

$$\hat{V}(\phi) = \begin{cases} 1 & \text{if } p \text{ is odd.} \\ \cos m\phi & \text{if } p \text{ is even.} \end{cases} \tag{50}$$

We have the following operation count for constructing and evaluating the polynomial when $p \ll N$:

	sequential	parallel	speed-up
Construction	$N \log N$	$n \log n$	$\sim p$
Evaluation	N	$n + 2p$	$\sim p$

We obtain a speed-up of order p both in the construction and evaluation of the polynomial as compared to the sequential FFT algorithm.

5. Chebyshev interpolation.

Let x_j, $j = 0, 1, \ldots, N-1$, be N Chebyshev points in $[-1,1]$ given by

$$x_j = \cos\theta_j, \qquad \theta_j = \frac{2j+1}{2N}\pi, \qquad j = 0, 1, \ldots, N-1, \tag{51}$$

and let $f(x)$ be a function defined on $[-1,1]$ whose values $f_j \equiv f(x_j)$, $j = 0, 1, \ldots, N-1$, are given. Let $N = np$, and consider the partition $X = \{X_0, X_1, \ldots, X_{p-1}\}$, where

$$X_l = \{x_{l,r} = x_{l+rp}, \quad r = 0, 1, \ldots, n-1\}, \quad l = 0, 1, \ldots, p-1. \tag{52}$$

We define a new partition, $Y = \{Y_0, Y_1, \ldots, Y_{q-1}\}$, $q = \lfloor (p+1)/2 \rfloor$ as follows:

$$Y_l = X_l \cup X_{l'}, \quad l' = p - 1 - l, \qquad l = 0, 1, \ldots, q-1. \tag{53}$$

LEMMA 5.1. *For $l = 0, 1, \ldots, q-1$, define*

$$w_{m,r}^{-1} = \prod_{k \neq l, l'} (x_{m,r} - x_{k,t}), \qquad m = l, l', \quad r = 0, 1, \ldots, n-1, \tag{54}$$

and let $Q_l(x)$ be the polynomial of degree $|Y_l| - 1$ that satisfies the interpolation conditions

$$Q_l(x_{k,r}) = f_{k+rp} w_{k,r}, \quad k = l, l', \quad r = 0, 1, \ldots, n-1, \tag{55}$$

on the subset of points Y_l. Then $P(x)$, the interpolation polynomial on X, can be expressed in the form

$$P(x) = \sum_{l=0}^{q-1} Q_l(x) \prod_{k \neq l, l'} (x - x_{k,t}). \tag{56}$$

PROOF. The result in (56) follows from Theorem 2.1. ∎

In developing the barycentric formula we distinguish between the cases in which p is even and odd.

THEOREM 5.2. *Let p be even, and for $l = 0, 1, \ldots, q-1$, let $\hat{Q}_l(x)$, be the polynomial of degree $2n-1$ that satisfies the interpolation conditions*

$$\hat{Q}_l(x_{k,r}) = f_{k+rp}, \quad k = l, l', \quad r = 0, 1, \ldots, n-1. \tag{57}$$

Then $P(x)$ of Lemma 5.1 can be expressed in the form

(58) $$P(x) = \frac{\sum_{l=0}^{q-1}(-1)^l \sin 2n\theta_l \hat{Q}_l(x)/(T_{2n}(x) - T_{2n}(x_l))}{\sum_{l=0}^{q-1}(-1)^l \sin 2n\theta_l/(T_{2n}(x) - T_{2n}(x_l))},$$

where $T_k(x)$ is the Chebyshev polynomial of the first kind of degree k.

PROOF. For $l = 0, 1, \ldots, q-1$, we let $R_l(x)$ be the polynomial of degree $2n-1$ that satisfies the interpolation conditions

(59) $$R_l(x_{k,r}) = w_{k,r}, \quad k = l, l', \quad r = 0, 1, \ldots, n-1,$$

on the subset of points Y_l. We then obtain the barycentric formula for $P(x)$

(60) $$P(x) = \frac{\sum_{l=0}^{q-1} Q_l(x)/\prod_{r=0}^{n-1}(x - x_{l,r})(x - x_{l',r})}{\sum_{l=0}^{q-1} R_l(x)/\prod_{r=0}^{n-1}(x - x_{l,r})(x - x_{l',r})}$$

from Theorem 2.2. Observing that

(61) $$\prod_{k=l,l'}(x - x_{k,t}) = 2^{-2n+1}(T_{2n}(x) - T_{2n}(x_l)),$$

we obtain

(62) $$\prod_{k \neq l,l'}(x_{l,r} - x_{k,t}) = \prod_{k,t \neq l,r}(x_{l,r} - x_{k,t}) \bigg/ \left(\prod_{t \neq r}(x_{l,r} - x_{l,t}) \prod_{t=0}^{n-1}(x_{l,r} - x_{l',t})\right)$$
$$= \left(\frac{N \sin N\theta_{l,r}}{2^{N-1}\sin\theta_{l,r}}\right) \bigg/ \left(\frac{2n \sin 2n\theta_{l,r}}{2^{2n-1}\sin\theta_{l,r}}\right)$$

and hence,

(63) $$w_{k,r} = c_0(-1)^l \sin 2n\theta_l, \quad k = l, l', \quad r = 0, 1, \ldots, n-1,$$

where $c_0 = 2^{N-2n+1}/p$ is a constant independent of l and r. Comparing (55), (57) and (59) with (63) we obtain

(64) $$Q_l(x) = c_0(-1)^l \sin 2n\theta_l \hat{Q}_l(x),$$

(65) $$R_l(x) = c_0(-1)^l \sin 2n\theta_l,$$

for $l = 0, 1, \ldots, q-1$. Combining (64), (65) and (61) in (60), we obtain (58). ■

THEOREM 5.3. *Let p be odd, and for $l = 0, 1, \ldots, q-2$, let $\hat{Q}_l(x)$, be the polynomial of degree $2n-1$ that satisfies the interpolation conditions*

(66) $$\hat{Q}_l(x_{k,r}) = \begin{cases} (-1)^r f_{l+rp}, & k = l \\ (-1)^{r+1} f_{l'+rp}, & k = l' \end{cases} \quad r = 0, 1, \ldots, n-1.$$

Furthermore, let $\hat{Q}_{q-1}(x)$, be the polynomial of degree $n-1$ that satisfies the interpolation conditions

(67) $$\hat{Q}_{q-1}(x) = f_{q-1+rp}, \qquad r = 0, 1, \ldots, n-1.$$

Then $P(x)$ of Lemma 5.1 can be expressed in the form

$$(68)\quad P(x) = \frac{\sum_{l=0}^{q-2}(-1)^l \sin 2n\theta_l \hat{Q}_l(x)/(T_{2n}(x) - T_{2n}(x_l)) + \frac{1}{2}(-1)^{q-1}\hat{Q}_{q-1}(x)/T_n(x)}{\sum_{l=0}^{q-2}(-1)^l 2\sin n\theta_l T_n(x)/(T_{2n}(x) - T_{2n}(x_l)) + \frac{1}{2}(-1)^{q-1}/T_n(x)}.$$

PROOF. We start by observing from (62) that for $l = 0, 1, \ldots, q-2$

$$(69)\quad w_{l,r} = c_0(-1)^{l+r}\sin 2n\theta_l,$$

$$(70)\quad w_{l',r} = c_0(-1)^{l+r+1}\sin 2n\theta_l.$$

Furthermore,

$$(71)\quad \prod_{t=0}^{n-1}(x - x_{q-1,t}) = 2^{-n+1}T_n(x),$$

and therefore,

$$(72)\quad \prod_{k\neq q-1}(x_{q-1,r} - x_{k,t}) = \prod_{k,t\neq q-1,r}(x_{q-1,r} - x_{k,t})\Big/\prod_{t\neq r}(x_{q-1,r} - x_{q-1,t})$$
$$= \left(\frac{N\sin N\theta_{q-1,r}}{2^{N-1}\sin\theta_{q-1,r}}\right)\Big/\left(\frac{n\sin n\theta_{q-1,r}}{2^{n-1}\sin\theta_{q-1,r}}\right).$$

Finally

$$(73)\quad w_{q-1,r} = c_1(-1)^{q-1},$$

where $c_1 = c_0 2^{n-1}$ is a constant independent of $q-1$ and r. Comparing (55) and (59) with (66), (69) and (70) we obtain for $l = 0, 1, \ldots, q-2$,

$$(74)\quad Q_l(x) = c_0(-1)^l \sin 2n\theta_l \hat{Q}_l(x),$$

$$(75)\quad R_l(x) = c_0(-1)^l \sin 2n\theta_l T_n(x)/T_n(x_l).$$

Similarly, from (67) and (73) we obtain

$$(76)\quad Q_{q-1}(x) = c_1(-1)^{q-1}\hat{Q}_{q-1}(x),$$

$$(77)\quad R_{q-1}(x) = c_1(-1)^{q-1}.$$

Combining (74), (75), (76) and (77) in (60), we obtain (68). ■

We next show how to find the corresponding polynomials $\hat{Q}_l(x)$, $l = 0, 1, \ldots, q-1$ using the FFT algorithm. We give explicit formulas for the case where p is odd. The case where p is even is solved similarly. Let $Q(x)$ be a polynomial of degree $m-1$. Then $Q(x)$ has a unique representation in terms of the Chebyshev polynomials of order less than m, i.e.,

$$(78)\quad Q(x) = \tfrac{1}{2}a_0 + \sum_{k=1}^{m-1} a_k T_k(x).$$

Rewriting the series in terms of $x = \cos\theta$, $z = e^{i\theta}$ we get the corresponding "symmetric" complex polynomial of degree $m-1$

(79) $$C(z) = \sum_{k=-m+1}^{m-1} c_k z^k, \quad c_k = c_{-k} = \tfrac{1}{2}a_k, \quad k = 0, 1, \ldots, m-1.$$

Let $Q(x)$ satisfy the interpolation conditions

(80) $$Q(x_j) = g_j, \quad x_j = \cos\theta_j, \quad \theta_j = \frac{2j+1}{2m}, \quad j = 0, 1, \ldots, m-1,$$

then $C(z)$ satisfies the interpolation conditions

(81) $$C(v_j) = P(x_j) = g_j, \quad v_j = e^{i\theta_j}, \quad j = 0, 1, \ldots, m-1,$$

and vice versa. Hence, $Q(x)$ can be obtained from $C(z)$.

THEOREM 5.4. *Let $\hat{C}(s)$ be the balanced complex trigonometric polynomial of degree n that satisfies the interpolation conditions*

(82) $$\begin{cases} \hat{C}(z_1^j) = \tfrac{1}{2}f_{q-1+jp} \\ \hat{C}(z_1^{n+j'}) = \tfrac{1}{2}f_{q-1+j'p} \end{cases} \quad j' = n-1-j, j = 0, 1, \ldots, n-1,$$

where $z_1 = e^{i\pi/n}$. Then $C(z)$, the symmetric complex polynomial of degree $n-1$ corresponding to $\hat{Q}_{q-1}(x)$ in (68), *can be expressed in the form*

(83) $$C(z) = \hat{C}(z/z_1^{1/2}) + \hat{C}(1/(zz_1^{1/2})).$$

PROOF. First, it is clear from (83) that $C(z)$ is a symmetric complex polynomial. Furthermore, from (23) $\hat{C}(s)$ is of the form

(84) $$\hat{C}(s) = \tfrac{1}{2}c_{-n}s^{-n} + \sum_{k=-n+1}^{n-1} c_k s^k + \tfrac{1}{2}c_n s^n, \qquad c_{-n} = c_n.$$

Consequently, the coefficients of z^n and z^{-n} in (83) are

(85) $$\tfrac{1}{2}c_n(z_1^{-n/2} + z_1^{n/2}) = 0,$$

and $C(z)$ is of degree at most $n-1$. Next, $Y_{q-1} = X_{q-1} = \left\{\cos\frac{2j+1}{2n}\pi,\ j = 0, 1, \ldots, n-1\right\}$ is a set of n Chebyshev points, and therefore from (82)

(86) $$C(v_j) = \hat{C}(z_1^j) + \hat{C}(z_1^{n+(n-1-j)}) = f_{q-1+jp}, \qquad j = 0, 1, \ldots, n-1,$$

and $C(z)$ satisfies the interpolation conditions. ■

THEOREM 5.5. *Let $\hat{C}(s)$ be the balanced complex trigonometric polynomial of degree n that satisfies the interpolation conditions*

(87) $$\begin{cases} \hat{C}(z_1^j) = f_{l+jp} \\ \hat{C}(z_1^{n+j'}) = f_{l'+j'p} \end{cases} \quad j' = n-1-j, \quad j = 0, 1, \ldots, n-1,$$

where $z_1 = e^{i\pi/n}$ *and* $l < q - 1$. *Then* $C(z)$, *the symmetric complex polynomial of degree* $2n - 1$ *corresponding to* $(-2i \sin 2n\theta_l \hat{Q}_l(x))$ *in* (68), *can be expressed in the form*

$$(88)\quad C(z) = \left(\left(\frac{z}{v_{l'}}\right)^n - \left(\frac{z}{v_{l'}}\right)^{-n}\right)\hat{C}\left(\frac{z}{v_l}\right) + \left(\left(\frac{1}{zv_{l'}}\right)^n - \left(\frac{1}{zv_{l'}}\right)^{-n}\right)\hat{C}\left(\frac{1}{zv_l}\right)$$
$$\equiv \tilde{C}(z) + \tilde{C}(z^{-1}).$$

PROOF. First, it is clear from (88) that $C(z)$ is a symmetric complex polynomial. Furthermore, from (23) $\hat{C}(z/v_l)$ is of the form

$$(89)\quad \hat{C}\left(\frac{z}{v_l}\right) = \tfrac{1}{2}c_{-n}\left(\frac{z}{v_l}\right)^{-n} + \sum_{k=-n+1}^{n-1} c_k\left(\frac{z}{v_l}\right)^k + \tfrac{1}{2}c_n\left(\frac{z}{v_l}\right)^n, \qquad c_{-n} = c_n.$$

Consequently the coefficients of z^{2n} *and* z^{-2n} in (88) are

$$(90)\quad \tfrac{1}{2}c_n((v_{l'}v_l)^{-n} - (v_{l'}v_l)^n) = 0,$$

and $C(z)$ is of degree at most $2n - 1$. Next, we observe that

$$(91)\quad \tilde{C}(z) = 2i \sin n(\theta - \theta_{l'})\hat{C}\left(\frac{z}{v_l}\right)$$

and therefore from (87) and (66)

$$(92)\quad C(v_{l,j}) = (-1)^{j-1} 2i \sin 2n\theta_l \hat{C}(z_1^j) = -2i \sin 2n\theta_l \hat{Q}(x_{l,j}),$$

$$(93)\quad C(v_{l',j}) = (-1)^j 2i \sin 2n\theta_l \hat{C}(z_1^{2n-1-j}) = -2i \sin 2n\theta_l \hat{Q}_l(x_{l',j}).$$

and $C(z)$ satisfies the interpolation conditions. ■

We have the following operation count for constructing and evaluating the polynomial when $N \sim n(2p)$, $p << N$:

	sequential	parallel	speed-up
Construction	$N \log N$	$2n \log(2n)$	$\sim p$
Evaluation	N	$2(n + p)$	$\sim p$

Again we obtain a speed-up of order p both for the construction and evaluation of the polynomial as compared to the sequential FFT algorithm.

6. The general Hermite interpolation problem.

Let $X = \{x_0, x_1, \ldots, x_M\}$ be given distinct points in the interval $[a, b]$, and let $f(x)$ be a function defined on $[a, b]$, for which

$$(94)\quad f_j^t \equiv f^{(t)}(x_j), \quad t = 0, 1, \ldots, k_j - 1, \quad j = 0, 1, \ldots, M,$$

are given. We are interested in constructing a representation of the general Hermite interpolation polynomial $P(x)$ of degree at most N, $N+1=\sum_{j=0}^{M} k_j$, that interpolates $f(x)$ on X, i.e.,

$$P^{(t)}(x_j)=f_j^t,\quad t=0,1,\dots,k_j-1,\quad j=0,1,\dots,M, \tag{95}$$

and is most suitable for parallel computation.

Let $\{X_1,X_2,\dots,X_p\}$ be a partition of X, i.e.,

$$X=\cup_1^p X_i \quad\text{and}\quad X_i\cap X_j=\emptyset \quad\text{for}\quad i\neq j. \tag{96}$$

The following theorem indicates how $P(x)$ can be constructed independently and in parallel by p processors, each solving a smaller general Hermite interpolation problem on one of the subsets X_i.

THEOREM 6.1. *For $i=1,\dots,p$, and $x_j\in X_i$, define*

$$v_{i,j}^0=\prod\nolimits_{x_r\notin X_i}(x_j-x_r)^{k_r}, \tag{97}$$

$$z_{i,j}^t=(-1)^t\sum_{x_r\notin X_i}\frac{k_r}{(x_j-x_r)^{t+1}},\qquad t=0,1,\dots,k_j-2, \tag{98}$$

$$v_{i,j}^t=\frac{1}{t}\sum_{s=0}^{t-1}v_{i,j}^s z_{i,j}^{t-1-s},\quad t=1,\dots,k_j-1, \tag{99}$$

$$q_{i,j}^t=\left(\frac{f_j^t}{t!}-\sum_{s=0}^{t-1}q_{i,j}^s v_{i,j}^{t-s}\right)\Big/ v_{i,j}^0,\qquad t=0,1,\dots,k_j-1. \tag{100}$$

Let $Q_i(x)$ be the polynomial of degree at most n_i-1, $n_i=\sum_{x_j\in X_i}k_j$, that satisfies the following interpolation conditions:

$$Q_i^{(t)}(x_j)=t!q_{i,j}^t,\qquad t=0,1,\dots,k_j-1. \tag{101}$$

Then $P(x)$, the interpolation polynomial on X, is given by

$$P(x)=\sum_{i=1}^{p}Q_i(x)\prod\nolimits_{x_r\notin X_i}(x-x_r)^{k_r}\equiv\sum_{i=1}^{p}Q_i(x)l_i(x). \tag{102}$$

PROOF. First, it is clear that the right hand side of (102) is a sum of polynomials each of degree at most N. Next, we observe that

$$l_i'(x)=l_i(x)\sum_{x_r\notin X_i}\frac{k_r}{x-x_r}\equiv l_i(x)z_i(x) \tag{103}$$

and

$$l_i^{(t+1)}(x)=\sum_{s=0}^{t}\binom{t}{s}l_i^{(s)}(x)z_i^{(t-s)}(x) \tag{104}$$
$$=t!\sum_{s=0}^{t}\frac{l_i^{(s)}(x)}{s!}\,\frac{z_i^{(t-s)}(x)}{(t-s)!}.$$

Comparing (103) and (104) with (97)–(99), we see that

$$z_i^{(t)}(x_j) = t!z_{i,j}^t, \qquad t = 0, 1, \ldots, k_j - 2, \tag{105}$$

and

$$l_i^{(t)}(x_j) = t!v_{i,j}^t, \qquad t = 0, 1, \ldots, k_j - 1. \tag{106}$$

Differentiating both sides of (102) t times, and setting $x = x_j \in X_i$ there, we finally obtain

$$P^{(t)}(x_j) = \sum_{s=0}^{t} \binom{t}{s} Q_i^{(s)}(x_j) l_i^{(t-s)}(x_j) \tag{107}$$

$$= t! \sum_{s=0}^{t} q_{i,j}^s v_{i,j}^{t-s} = f_j^t, \tag{108}$$

as required. The result now follows from the uniqueness of $P(x)$. ■

We conclude with the barycentric formula for $P(x)$ given in Theorem 6.2 below.

THEOREM 6.2. *The general Hermite interpolation polynomial $P(x)$ of Theorem 6.1 has the barycentric form*

$$P(x) = \frac{\sum_{i=1}^{p} Q_i(x) / \prod_{x_j \in X_i} (x - x_j)^{k_j}}{\sum_{i=1}^{p} R_i(x) / \prod_{x_j \in X_i} (x - x_j)^{k_j}}, \tag{109}$$

where $R_i(x)$, like $Q_i(x)$, is a polynomial of degree at most $n_i - 1$ that satisfies the same interpolation conditions with f_j replaced by 1, and f_j^t by 0, $t = 1, \ldots, k_j - 1$, for all j.

PROOF. As in Theorem 2.2. ■

For example, for the classical Hermite interpolation problem, in which $k_j = 2$ for all j, we get

$$Q_i(x_j) = f_j / v_{i,j}^0, \quad Q_i'(x_j) = f_j' / v_{i,j}^0 - f_j \sum_{x_r \notin X_i} \frac{2}{x_j - x_r}, \tag{110}$$

and

$$R_i(x_j) = 1 / v_{i,j}^0, \quad R_i'(x_j) = - \sum_{x_r \notin X_i} \frac{2}{x_j - x_r}. \tag{111}$$

We can find the formulas for $v_{i,j}^s$, $s = 0, 1, \ldots, k_j - 1$, in

$$O\left(\sum_{x_j \in X_i} k_j |X - X_i| + k_j^2 \right) \leq O(n_i |X - X_i| + n_i^2) \tag{112}$$

operations, and the formulas for the $q_{i,j}^s$, $s = 0, 1, \ldots, k_j$, in

$$O\left(\sum_{x_j \in X_i} k_j^2\right) \le O(n_i^2) \tag{113}$$

operations. Let $n_i \sim n \sim N/p$, $i = 1, \ldots, p$, and assume that the $v_{i,j}^s$ are known. Each processor is then faced with a general Hermite interpolation problem of order n approximately that can be solved in $O(n^2)$ operations.

7. Conclusion.

We have presented new mathematical formulas for polynomial and trigonometric interpolation that are especially useful for parallel computers. These interpolation formulas adjust themselves to the number of processors available, receiving a classical form in case of a single processor, and the Lagrange form in case there are as many processors as needed. The interpolation problem is broken down into smaller independent subproblems, which can be solved independently using currently existing software tools. We have shown that in most cases of interest the formulation of the interpolation subproblems can be done analytically, reducing the problem from order N to order $n \sim N/p$ and achieving optimum speed-ups. Furthermore, the barycentric formula developed in the present work can be seen to enjoy a high degree of numerical stability as in the case with the barycentric formula for the ordinary Lagrange interpolation.

REFERENCES

1. M. L. Dowling, *A fast parallel Horner algorithm*, SIAM J. Comput., 19 (1990), pp. 133–142.
2. O. Eğecioğlu, E. Gallopoulos and Ç. Koç, *Fast computation of divided differences and parallel Hermite interpolation*, J. Complexity, 30 (1990), pp. 268–288.
3. O. Eğecioğlu, E. Gallopoulos and Ç. Koç, *A parallel method for fast and practical high-order Newton interpolation*, BIT, 30 (1990), pp. 268–288.
4. P. Henrici, *Essentials of Numerical Analysis*, John Wiley & Sons, 1982.
5. F. B. Hildebrand, *Introduction to Numerical Analysis*, McGraw-Hill Book Company, 1974.
6. I. Munro and M. Paterson, *Optimal algorithms for parallel polynomial evaluation*, J. Comput. System Sci., 7 (1973), pp. 189–198.
7. J. Reif, *Logarithmic depth circuits for algebraic functions*, SIAM J. Comput., 15 (1986), pp. 231–242.
8. J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer Verlag, 1980.
9. W. J. Taylor, *Method of Lagrangian curvilinear interpolation*, J. Res. of the Nat. Bur. of Standards, 35 (1945), pp. 151–155.
10. W. Werner, *Polynomial interpolation: Lagrange versus Newton*, Math. Comp. 43 (1984), pp. 205–217.