

Parallel Hermite Interpolation on Extended Fibonacci Cubes

B.N.B.Ray

Department of Computer
Science and Application

Utkal University, Vanivihar
Bhubaneswar, Odisha 751004

Alok Ranjan Tripathy

Department of Computer
Science and Application

Utkal University, Vanivihar
Bhubaneswar, Odisha 751004

S.P.Mohanty

Department of Mathematics
College of Engineering and
Technology

Bhubaneswar, Odisha 751003

ABSTRACT

This work suggests a parallel algorithm for Hermite interpolation on Extended Fibonacci Cube $EFC_1(n)$. The proposed algorithm has 3 phases: initialization, main and final. The main phase of the algorithm involves $2N+3$ multiplications, N additions, $2N$ subtractions and N divisions. In final phase we propose an efficient algorithm to accumulate the partial sums of Hermite interpolation in $O(\log_2 N) \leq n-2$ steps as oppose to the earlier algorithm in the literature that involves $n-2$ steps, where N is the number of nodes, n the degree of $EFC_1(n)$.

Keywords

Hermite Interpolation, Extended Fibonacci Cubes, Parallel algorithm.

1. INTRODUCTION

In curve fitting, interpolation is the process of replacing a continuous function by a polynomial that agrees with the function at specified number of data points. Using interpolation polynomial a prediction to a function value is made by an interpolating polynomial at new data point. In numerical analysis interpolation techniques are of great importance as they are used for modeling various scientific and engineering problems, via weather forecasting, portfolio management, air traffic prediction etc. Many numerical interpolation techniques are proposed in literature. Among which Hermite interpolation is one of the most widely used polynomial fitting techniques which is a special case of Birkhoff interpolation [1, 2]. In interpolation when the numbers of data points are large, uniprocessor environment involves a long computation and large storage space to carry out the computation. To get around these difficulties parallel computation is an obvious advantage.

A bunch of parallel algorithms for interpolation have been developed over past years by various workers. Azad et al. [7] developed parallel algorithm for $N=n!$ point Lagrange interpolation for n -star ($n > 2$). Ioana Zelina [4] developed parallel Lagrange interpolation on Extended Fibonacci Cube. Sarbzi-Azad [6] developed parallel algorithm for Hermite interpolation in Pyramid network. Extended Fibonacci Cube was introduced by Jie Wu [8] in the year 1997. Unlike Fibonacci Cube [3], an extended version of Fibonacci Cube is Hamiltonian, whereas less than $1/3$ of Fibonacci Cubes are not Hamiltonian. Another advantage of Extended Fibonacci Cubes is that it could be stretched to construct parallel machines of any size. Further Extended Fibonacci Cube

behaves like faulty hypercube. As Hermite interpolation is a higher degree polynomial for a given data set and its error bound is less than Newton and Lagrange interpolation. It is interesting to investigate the computational efficiency of Hermite interpolation in extended version of Fibonacci Cube. These features of Extended Fibonacci Cube and Hermite interpolation prompted us to develop parallel algorithm for Hermite interpolation on Extended Fibonacci Cube. To the best of our knowledge this is the first and efficient parallel algorithm for Hermite interpolation in Extended Fibonacci Cube. In short our contributions in this paper are as follows:

Developed parallel algorithm for Hermite interpolation in extended version of Fibonacci Cube.

Proposed an efficient algorithm for accumulating the partial sums of Hermite interpolation in $O(\log_2 N) \leq n-2$ steps (See Theorem 2 below). Whereas the algorithm discussed in [4] involves $n-2$ steps. Where ' n ' and ' N ' are degree and number of nodes of Extended Fibonacci Cube respectively. Determined computational statistics viz. number of data communications, multiplications, additions and subtractions required for the proposed parallel algorithm.

The rest of the paper is organized as follows: Section 2 reviews the Extended Fibonacci Cubes. Section 3 discusses the formulation of Hermite interpolation required for the parallel algorithm. In section 4 we present the proposed parallel Hermite interpolation algorithm. Section 5 discusses analysis of the parallel algorithm. Finally, conclusion is drawn in section 6.

2. EXTENDED FIBONACCI CUBES

Extended Fibonacci Cubes (EFC) topology is an interconnection network generalized by Wu. J. [8] from Fibonacci Cube proposed by Hsu [3]. The initial conditions used by both topologies are different from the initial conditions of Fibonacci series. An Extended Fibonacci Cube of order k ($k=1,2$) is denoted by $EFC_k(n)$, where $n-2$ is the length of bit string representing the address of nodes in EFC . It may be mentioned that $EFC_k(n)$ is a sub-graph of the corresponding hypercube and a node of $EFC_k(n)$ is addressed by Fibonacci Code (FC). The simplest version of these cubes series is the Fibonacci Cubes.

Let $EFC_1(n) = (V_1(n), E_1(n))$ [4], where $V_1(n)$ is the set of nodes and $E_1(n)$ is the set of edges in $EFC_1(n)$ and $EFC_1(n-1) = (V_1(n-1), E_1(n-1))$,

$EFC_1(n-2) = (V_1(n-2), E_1(n-2))$. Using these notations

$EFC_1(n)$ can be defined recursively by using $EFC_1(n-1)$ and $EFC_1(n-2)$ as follows:
 $V_1(n) = 0 \parallel V_1(n-1) \cup 10 \parallel V_1(n-2)$ where \parallel denotes the concatenation of two strings. The initial condition for recursion is $V_1(3) = \{0, 1\}$ and $V_1(4) = \{00, 10, 11, 01\}$. Two nodes in $EFC_1(n)$ are connected if and only if their address representations differ in exactly 1-bit position. Some $EFC_1(n)$ are shown in figure 1 for $n = 3, 4, 5$. Note that each $EFC_1(n)$ consists of an $EFC_1(n-1)$ and an $EFC_1(n-2)$. It is known that around less than one third of Fibonacci cubes is Hamiltonian, whereas all $EFC_k(n)$ are Hamiltonian. The diameter of $EFC_1(n)$ is given by $n-2$ and the degrees of nodes lie between $\left\lceil \frac{n}{3} \right\rceil$ and $n-2$.

3. HERMITE INTERPOLATION

The Hermite interpolating polynomial in x for the data set $(x_0, y_0, y'_0) \dots (x_{N-1}, y_{N-1}, y'_{N-1})$ is a polynomial of degree $2N-1$ and is given by [5]

$$H_{2N-1}(x) = \sum_{i=0}^{N-1} [1 - 2(x - x_i)l'_i(x_i)]l_i^2(x)y_i + \sum_{i=0}^{N-1} (x - x_i)l_i^2(x)y'_i \quad (3.1)$$

Where $l_i(x)$ is the Lagrange polynomial in x and

$$H_{2N-1}(x_i) = y_i, H'_{2N-1}(x_i) = y'_i, \text{ for } i = 0, 1, \dots, N-1$$

From (3.1) we have

$$H_{2N-1}(x) = \sum_{i=0}^{N-1} [1 - 2(x - x_i)l'_i(x_i)]l_i^2(x)y_i + \sum_{i=0}^{N-1} (x - x_i)l_i^2(x)y'_i \\ = \sum_{i=0}^{N-1} h_i(x)l_i^2(x) \quad (3.2)$$

Where $h_i(x) = [1 - 2(x - x_i)l'_i(x_i)y_i + (x - x_i)y'_i]$

$$l_i(x) = \frac{(x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_{N-1})}{(x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_{N-1})}$$

$$l'_i(x_i) = \sum_{\substack{j=0 \\ j \neq i}}^{N-1} \frac{1}{(x_i - x_j)} \text{ for } i = 0, 1, \dots, N-1$$

We need the following lemma for the development of the proposed parallel algorithm.

Lemma1. For Lagrange function $l_i(x)$ with dataset

$x_0, x_1 \dots x_{N-1}$, $l'_i(x_i)$ is given by

$$l'_i(x_i) = \sum_{\substack{j=0 \\ j \neq i}}^{N-1} \frac{1}{(x_i - x_j)}$$

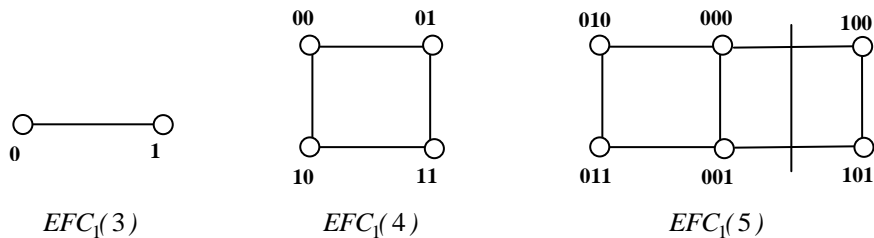


Fig 1: $EFC_1(n)$, where $n = 3, 4, 5$

Proof.

$$l_i(x) = \frac{(x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_{N-1})}{(x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_{N-1})}$$

On differentiating with respect to x we have

$$l'_i(x) = ((x - x_1) \dots (x - x_{N-1}) + (x - x_0) \dots (x - x_{N-1}) + \dots + (x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_{N-2})) / \Delta$$

$$\text{Where } \Delta = (x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_{N-1})$$

Now

$$l'_i(x_i) = \frac{1}{(x_i - x_0)} + \frac{1}{(x_i - x_1)} + \frac{1}{(x_i - x_2)} + \dots + \frac{1}{(x_i - x_{N-1})}$$

$$= \sum_{\substack{j=0 \\ i \neq j}}^{N-1} \frac{1}{(x_i - x_j)} \quad \square$$

4. PROPOSED PARALLEL ALGORITHM

The proposed parallel algorithm is based on the parallel algorithm described for Lagrange interpolation in Extended Fibonacci Cube [4] using equation (3.2) and the result of Lemma 1. Let we are given a dataset

$(x_0, y_0, y'_0) \dots (x_{N-1}, y_{N-1}, y'_{N-1})$ and value x , where N is the number of nodes of $EFC_1(n)$. The parallel algorithm computes Hermite interpolations in three phases: initialization, main and final phases. Initially, each processor

$P_i (0 \leq i < N)$ is given a data point (x_i, y_i, y'_i) which runs in parallel with other processors. Each processor calculates $l_i(x)$

, $l'_i(x_i)$ and the product $h_i(x) \times l_i^2(x)$ for a given value of x . Then the partial products at all processors are added to obtain the final result.

Let P_w be a processor corresponds to a node of the Extended Fibonacci Cube $EFC_1(n)$ with the binary representation w . Let us assume each processor node has five registers R_1, R_2, R_3, R_4, R_5 . The registers R_1 and R_2 hold the terms required for calculating $l_i(x)$ and the register R_5 holds the term to calculate $l'_i(x)$. The registers R_3 and R_4 will be used to implement all-to-all broadcast algorithm [4] in $EFC_1(n)$.

When constructing a Hamiltonian cycle in $EFC_1(n)$, two arrays, Next[w] and Previous[w] are used to indicate nodes before and after the nodes $w, w \in V(n)$. For any node P_w in the Hamiltonian ring, the next and previous nodes are $P_{\text{Next}[w]}$ and $P_{\text{Previous}[w]}$ respectively. Those arrays should have been set to their values before starting the initialization phase.

4.1 Initialization Phase

In this phase the values x , x_i, y_i, y_i' are given to the processor P_w for storing in its local memory. Recall that i is the order of the Fibonacci number $w = f_i$, $i = 1, \bar{N}$

$$P_w^{(0)}(R_1) = 1, P_w^{(0)}(R_2) = 1, P_w^{(0)}(R_3) = x_i, P_w^{(0)}(R_4) = x_i, P_w^{(0)}(R_5) = 0$$

The initialization phase requires no communication steps and no computation steps. Let $P_{a_1, a_2, \dots, a_n}(R_m)$ denotes the content of register R_m ($m = 1, 2, 3, 4, 5$) of the node P_{a_1, a_2, \dots, a_n} . Further, let $P_{a_1, a_2, \dots, a_n}^{(t)}(R_m)$ indicates the content of register R_m of P_{a_1, a_2, \dots, a_n} after step t , where a step may involve a set of communication and computational operations. Symbol ' \Leftarrow ' denotes a communication operation between two adjacent nodes.

4.2 Main Phase

Algorithm 1. Parallel Algorithm to calculate $h_i(x) \times l_i^2(x)$

for $t = 0$ to $N/2 - 2$

$$P_w^{(t+1)}(R_3) \Leftarrow P_{Next(w)}^{(t)}(R_3)$$

$$P_w^{(t+1)}(R_4) \Leftarrow P_{Prev(w)}^{(t)}(R_4)$$

$$P_w^{(t+1)}(R_1) = P_w^{(t)}(R_1) \times (x - P_w^{(t+1)}(R_3)) \times (x - P_w^{(t+1)}(R_4))$$

$$P_w^{(t+1)}(R_2) = P_w^{(t)}(R_2) \times (x - P_w^{(t+1)}(R_3)) \times (x - P_w^{(t+1)}(R_4))$$

$$P_w^{(t+1)}(R_5) = P_w^{(t)}(R_5) + \frac{1}{(x_w - P_w^{(t+1)}(R_3))} + \frac{1}{(x_w - P_w^{(t+1)}(R_4))}$$

end for

$$P_w^{(N/2)}(R_3) \Leftarrow P_{Next(w)}^{(N/2-1)}(R_3)$$

$$P_w^{(N/2)}(R_1) = P_w^{(N/2-1)}(R_1) \times (x - P_w^{(N/2)}(R_3))$$

$$P_w^{(N/2)}(R_2) = P_w^{(N/2-1)}(R_2) \times (x_w - P_w^{(N/2)}(R_3))$$

$$P_w^{(N/2)}(R_5) = P_w^{(N/2-1)}(R_5) + \frac{1}{(x_w - P_w^{(N/2)}(R_3))}$$

After the execution above instruction sequence, we have

$$P_w(R_1) = (x - x_0)(x - x_1)(x - x_2) \dots (x - x_{w-1})(x - x_{w+1}) \dots (x - x_{N-1})$$

$$P_w(R_2) = (x_w - x_0)(x_w - x_1)(x_w - x_2) \dots (x_w - x_{w-1})(x_w - x_{w+1}) \dots (x_w - x_{N-1})$$

$$P_w(R_5) = \frac{1}{(x_w - x_0)} + \frac{1}{(x_w - x_1)} + \frac{1}{(x_w - x_2)} + \dots + \frac{1}{(x_w - x_{w-1})} +$$

$$\frac{1}{(x_w - x_{w+1})} + \dots + \frac{1}{(x_w - x_{N-1})}$$

Now to this end all processors shall execute the following instruction to calculate i^{th} term of $h_i(x) \times l_i^2(x)$ for $i = 0, 1, \dots, N-1$ of the Hermite interpolation.

$$P_w^{(\frac{N}{2}+1)}(R_1) = \frac{P_w^{(\frac{N}{2})}(R_1)}{P_w^{(\frac{N}{2})}(R_2)}$$

$$P_w^{(\frac{N}{2}+1)}(R_5) = P_w^{(\frac{N}{2})}(R_5)$$

$$P_w^{(\frac{N}{2}+2)}(R_5) = P_w^{(\frac{N}{2}+1)}(R_5)$$

$$P_w^{(\frac{N}{2}+2)}(R_1) = P_w^{(\frac{N}{2}+1)}(R_1) \times P_w^{(\frac{N}{2}+1)}(R_1)$$

$$P_w^{(\frac{N}{2}+3)}(R_1) = P_w^{(\frac{N}{2}+2)}(R_1) \times \left\{ (1 - 2(x - x_w) P_w^{(\frac{N}{2}+2)}(R_5) y_w) + (x - x_0) \times y_w' \right\}$$

At the end of this phase

$$P_w(R_1) = h_w(x) \times l_w^2(x)$$

4.3 Final Phase

In this phase the contents of register R_1 of all processors are added together to obtain the final result. For parallel Lagrange interpolation in $EFC_I(n)$, the authors in [4] have used an algorithm for adding contents of register R_1 of all nodes in $n-2$ steps. In the following a new parallel algorithm (Algorithm 2) for all-to-one-reduction is suggested for adding the contents of register R_1 of all processors. Algorithm 2 involves $\lceil \log_2 N \rceil \leq (n-2)$ steps (See Theorem 2 below) as oppose to $n-2$ steps discussed in [4]. This is evident from the results of Theorem 1, Theorem 2 and Theorem 3.

Theorem 1. For $n \geq 3$, the number of nodes N of $EFC_I(n)$ is given by

$$N(n) = \left(\frac{\sqrt{5}+3}{\sqrt{5}} \right) \phi^{n-3} + \left(\frac{\sqrt{5}-3}{\sqrt{5}} \right) \hat{\phi}^{n-3},$$

$$\text{Where } \phi = \frac{1+\sqrt{5}}{2}, \quad \hat{\phi} = \frac{1-\sqrt{5}}{2}$$

Proof. The recurrence relation for $EFC_I(n)$ [4] is given by

$$f_n = f_{n-1} + f_{n-2} \quad n \geq 5 \quad (4.1)$$

$$f_3 = 2 \quad f_4 = 4$$

The characteristic equation of equation (4.1) is given by

$$t = \frac{1}{t} + \frac{1}{t^2}$$

$$\Rightarrow t^2 - t - 1 = 0$$

$$\Rightarrow t = \frac{1 \pm \sqrt{5}}{2}$$

$$\text{Let } \phi = \frac{1+\sqrt{5}}{2}, \quad \hat{\phi} = \frac{1-\sqrt{5}}{2}$$

The solution of equation of (4.1) is given by

$$f_n = A\phi^n + B\hat{\phi}^n \quad (4.2)$$

Where A and B are constants to be determined using boundary conditions

$$f_3 = 2 \text{ and } f_4 = 4.$$

Now using boundary conditions, we have

$$2 = A\phi^3 + B\hat{\phi}^3 \quad (4.3)$$

$$4 = A\phi^4 + B\hat{\phi}^4 \quad (4.4)$$

$$2\phi = A\phi^4 + B\hat{\phi}^3\phi$$

$$4 = A\phi^4 + B\hat{\phi}^4$$

$$2\phi - 4 = B\hat{\phi}^3(\phi - \hat{\phi})$$

$$\Rightarrow B = \frac{(2\phi - 4)}{\hat{\phi}^3(\phi - \hat{\phi})} = \frac{(1 + \sqrt{5} - 4)}{\hat{\phi}^3 \times \sqrt{5}} = \frac{(\sqrt{5} - 3)}{\hat{\phi}^3 \times \sqrt{5}}$$

$$A = \frac{2 - B\hat{\phi}^3}{\phi^3} = \frac{2 - \left(\frac{(\sqrt{5} - 3)}{\hat{\phi}^3 \times \sqrt{5}} \times \hat{\phi}^3 \right)}{\phi^3}$$

$$= \frac{2 - \frac{(\sqrt{5} - 3)}{\sqrt{5}}}{\phi^3}$$

$$= \frac{(\sqrt{5} + 3)}{\phi^3 \times \sqrt{5}}$$

$$\therefore A = \frac{(\sqrt{5} + 3)}{\phi^3 \times \sqrt{5}} \quad B = \frac{(\sqrt{5} - 3)}{\hat{\phi}^3 \times \sqrt{5}}$$

$$\therefore f_n = \left(\frac{\sqrt{5} + 3}{\sqrt{5}} \right) \frac{1}{\phi^3} \phi^n + \left(\frac{\sqrt{5} - 3}{\sqrt{5}} \right) \frac{1}{\hat{\phi}^3} \hat{\phi}^n$$

$$f_n = \left(\frac{\sqrt{5} + 3}{\sqrt{5}} \right) \phi^{n-3} + \left(\frac{\sqrt{5} - 3}{\sqrt{5}} \right) \hat{\phi}^{n-3} \quad (4.5)$$

Thus the number of nodes $N = N(n)$ of $EFC_I(n)$ is given by equation (4.5).

Hence the theorem. \square

Theorem 2. If N is the number of nodes of $EFC_I(n)$ with $n \geq 3$. Then $\lceil \log_2 N \rceil \leq n - 2$

Proof. From Theorem 1.

$$N = \left(\frac{\sqrt{5} + 3}{\sqrt{5}} \right) \phi^{n-3} + \left(\frac{\sqrt{5} - 3}{\sqrt{5}} \right) \hat{\phi}^{n-3}$$

We claim that $\log_2 N \leq n - 2$

$$\Leftrightarrow 2^{n-2} \geq N$$

$$\Leftrightarrow 2^{n-2} \geq \left(\frac{\sqrt{5} + 3}{\sqrt{5}} \right) \phi^{n-3} + \left(\frac{\sqrt{5} - 3}{\sqrt{5}} \right) \hat{\phi}^{n-3} \quad (4.6)$$

We will prove (4.6) by induction on n .

Basis: For $n = 3$, left hand side of (4.6) is $2^{n-2} = 2^{3-2} = 2$ and right hand side of (4.6) is

$$\left(\frac{\sqrt{5} + 3}{\sqrt{5}} \right) \phi^{3-3} + \left(\frac{\sqrt{5} - 3}{\sqrt{5}} \right) \hat{\phi}^{3-3} = 1$$

Since $2 > 1$, the induction is true for $n = 3$.

Inductive Hypothesis: Let the inductive hypothesis is true for n

$$\text{i.e. } 2^{n-2} \geq \left(\frac{\sqrt{5} + 3}{\sqrt{5}} \right) \phi^{n-3} + \left(\frac{\sqrt{5} - 3}{\sqrt{5}} \right) \hat{\phi}^{n-3}$$

Inductive step: We will prove (4.6) for $n + 1$

$$\text{i.e. to show } 2^{n+1-2} \geq \left(\frac{\sqrt{5} + 3}{\sqrt{5}} \right) \phi^{n+1-3} + \left(\frac{\sqrt{5} - 3}{\sqrt{5}} \right) \hat{\phi}^{n+1-3}$$

Now

$$2^{n+1-2} = 2(2^{n-2}) \geq 2 \left[\left(\frac{\sqrt{5} + 3}{\sqrt{5}} \right) \phi^{n-3} + \left(\frac{\sqrt{5} - 3}{\sqrt{5}} \right) \hat{\phi}^{n-3} \right]$$

$$\geq \left[\left(\frac{\sqrt{5} + 3}{\sqrt{5}} \right) \phi \cdot \phi^{n-3} + \left(\frac{\sqrt{5} - 3}{\sqrt{5}} \right) \hat{\phi} \cdot \hat{\phi}^{n-3} \right]$$

$$\geq \left[\left(\frac{\sqrt{5} + 3}{\sqrt{5}} \right) \phi^{n+1-3} + \left(\frac{\sqrt{5} - 3}{\sqrt{5}} \right) \hat{\phi}^{n+1-3} \right]$$

Using inductive hypothesis and the fact that $2 \geq \phi, \hat{\phi}$

Hence the prove. \square

4.4 Proposed All-to-One Reduction

We discuss the algorithm for a $EFC_I(6)$. Following figure 2 shows the diagram for $EFC_I(6)$.

The Hamiltonian path of the network is given by

$$0000 \rightarrow 0100 \rightarrow 0101 \rightarrow 0001 \rightarrow 1001 \rightarrow 1000 \rightarrow 1010$$

$$\rightarrow 1011 \rightarrow 0011 \rightarrow 0010 \rightarrow 0000$$

Let $P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5 \rightarrow P_6 \rightarrow P_7 \rightarrow P_8 \rightarrow P_9 \rightarrow P_0$ it be

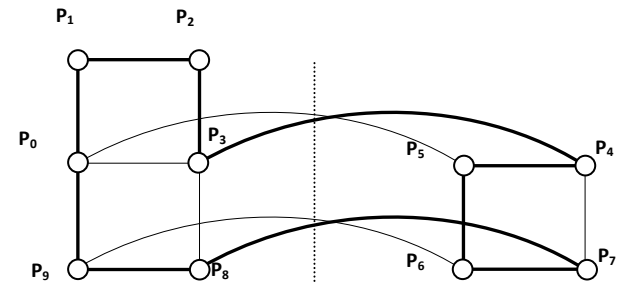


Fig. 2. $EFC_I(6)$

Our reduction algorithm uses following steps.

Let P_0, P_1, \dots, P_9 hold data $(x_0, x_1, x_2, \dots, x_9)$ in register R_1 respectively.

Algorithm 2. All-to-One Reduction

Step1. Processors P_1, P_3, P_5, P_7, P_9 send the content of the register R_1 in parallel to P_2, P_4, P_6, P_8, P_0 and at the end of the stage P_2, P_4, P_6, P_8, P_0 hold partial sum of content of R_1 and the received data. The register R_1 of P_2 contains the sum $(x_1 + x_2)$, P_4 holds $(x_3 + x_4)$, P_6 holds $(x_5 + x_6)$, P_8 holds $(x_7 + x_8)$ and P_0 holds $(x_9 + x_0)$ so on.

Step2. At this step P_2, P_6 send their content to P_4, P_8

At the end of this step P_4 holds $(x_1 + x_2 + x_3 + x_4)$ and P_8 holds $(x_5 + x_6 + x_7 + x_8)$.

Step3. At this step P_4 sends their content to P_8 and it holds $(x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8)$.

Step4. At the end of this step $\sum_{i=1}^8 x_i$ for P_8 is added with the partial sum $(x_9 + x_0)$ in P_0 .

It is interesting to note that the proposed accumulation algorithm involves $\log N$ data communication steps.

Theorem 3. The proposed All-to-One-Reduction algorithm i.e. Algorithm 2 in final phase of Hermite interpolation, for $EFC_I(n)$ involves less than $(n-2)$ communication operations as oppose to the earlier algorithm [4] that involves $(n-2)$ operations. Thus the proposed Algorithm 2 is communication optimal.

Proof. Since the Algorithm 2 involves $\log N$ steps. Theorem 3 follows from Theorem 2. \square

Following Table1 compares the number of steps taken by the proposed Algorithm2 and the algorithm discussed in [4] for values of $n \geq 4$. It is evident from Table1 and the plots shown in Fig.3 that the steps taken by Algorithm2 is less than the algorithm discussed in [4].

Table 1. Comparison of the number of steps.

Degree of $EFC_I(n)$	No. of nodes of $EFC_I(n)$ N	No. of steps in algorithm[4] $(n-2)$	No. of steps in proposed All-to-One-Reduction $\log_2 N$
5	6	3	2.584
6	10	4	3.321
7	16	5	4.000
8	26	6	4.700
9	42	7	5.392
10	68	8	6.087
11	110	9	6.781

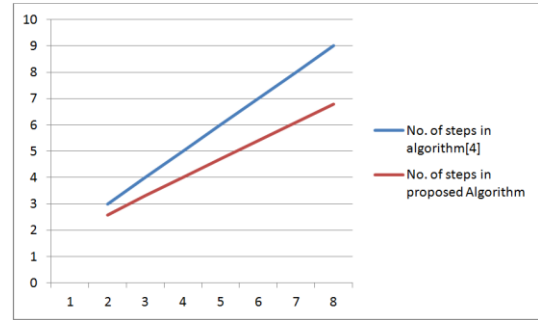


Fig. 3. Comparison between algorithm [4] and proposed algorithm.

5. ANALYSIS OF COMPUTATIONAL STATISTICS

Theorem 4. The main phase of the algorithm for Hermite interpolation involved $(2N+3)$ multiplications N -additions $2N$ -subtractions and N -divisions.

Proof. Let M, S, A and D represents the number of multiplications, subtractions, additions and divisions respectively. Now, a closer look of the algorithm reveals that there are $N/2$ number of data communications at the end of $(N/2+3)^{th}$ step.

At $(\frac{N}{2}-1)$ step we have $M = 4(\frac{N}{2}-1) = 2N-4$,

$A = 2(\frac{N}{2}-1) = N-2$

$S = 4(\frac{N}{2}-1) = 2N-4$ and $D = 2(\frac{N}{2}-1) = N-2$

At $\frac{N}{2}$ step, $M = 2, A = 1, S = 2, D = 1$

$(\frac{N}{2}+1)$ step, $M = 0, A = 0, S = 0, D = 1$

$(\frac{N}{2}+2)$ step, $M = 1, S = 0, A = 0, D = 0$

$(\frac{N}{2}+3)$ step, $M = 4, S = 2, A = 1, D = 0$

Adding number of multiplications, additions, subtractions and divisions the theorem follows. \square

6. CONCLUSION

This work suggests a parallel algorithm for Hermite interpolation in extended version of Fibonacci Cube. The proposed algorithm at final stage accumulates partial sums in less than $n-2$ steps as oppose to the earlier final phase algorithm discussed for Lagrange interpolation in [4], which involves $n-2$ steps. Where ' n ' and ' N ' are degree and number of nodes of Extended Fibonacci Cube respectively. The number of data communications, multiplications, additions, subtractions and divisions required for the proposed parallel algorithm are $(N/2+3), (2N+3), N, 2N$ and N respectively.

7. REFERENCES

- [1] D. Kincaid, W. Cheney, Numerical Analysis: Mathematics of Scientific Computing, 3rd Edition, Brooks/Cole Publisher, 2001.
- [2] G. Lorentz, K. Jetter, S.D. Riemanschneder, Birkhoff Interpolation, Addison-Wesley, 1983.
- [3] Hsu, W.J., "Fibonacci Cubes-A New Interconnection Topology," IEEE Trans. Parallel and Distributed Systems, vol.4, no.1, pp.3-12, Jan.1993
- [4] Ioana Zelina: Parallel Lagrange Interpolation on Extended Fibonacci Cube. STUDIA UNIV. BABES-BOLYAI, INFORMATICA, Volume L, Number 1, pp.105-110, 2005
- [5] Jain M.K., Iyengar S. R. K., Jain R. K., "Numerical Methods for Scientific and Engineering Computation" 3e, Wiley Eastern Limited.
- [6] Larijani, E.; Sarbazi-Azad, H.: Parallel hermite interpolation on the pyramid, 22nd international symposium on Digital Object Identifier, Computer and information sciences, 2007. pp. 1-5, 2007
- [7] Sarbazi-Azad, H., Ould-Khaoma, M., Mackenzie, L.M., Parallel Lagrange Interpolation on the Star Graph, Proc. 14th IPDPS, Cancun, Mexico, pp.777, 2000
- [8] Wu, J., "Extended Fibonacci Cubes", IEEE Trans. on Parallel and Distributed Systems, vol.8, no.12, pp.1203-1210, 1997