An Optimal Scheduling Procedure for Matrix Inversion on Linear Array at a Processor Level

M. K. Stojčev, E. I. Milovanović, and I. Ž. Milovanović

Received September 1992; revised October 1993

This paper presents a parallel algorithm for computing the inversion of a dense matrix based on modified Jordan's elimination which requires fewer calculation steps than the standard one. The algorithm is proposed for the implementation on the linear array with a small to moderate number of processors which operate in a parallel-pipeline fashion. A communication between neighboring processors is achieved by a common memory module implemented as a FIFO memory module. For the proposed algorithm we define a task scheduling procedure and prove that it is time optimal. In order to compute the speedup and efficiency of the system, two definitions (Amdahl's and Gustafson's) were used. For the proposed architecture, involving two to 16 processors, estimated Gustafson's (Amdahl's) speedups are in the range 1.99 to 13.76 (1.99 to 9.69).

KEY WORDS: Parallel algorithms; Gauss-Jordan elimination; linear processor array; task scheduling.

1. INTRODUCTION

The problem of inverting a real matrix of order \( n \) is one of the central problems in numerical linear algebra. It arises in many important and diverse fields such as digital signal processing, circuits analysis, speech analysis, optimization, and quantitative study of business and economic problems. The arithmetical computational complexity of the methods for dense matrix inversion is \( O(n^3) \), and therefore computations are time con-
suming for large $n$. Parallel processing has become a valuable candidate to compute the matrix inversion in a significantly faster manner.

One of the frequently used methods to invert a matrix is a method based on Gauss-Jordan elimination. Melhelm(1) presented a computational network which consists of $n^2/2$ cells and performs Gauss-Jordan elimination in $4n - 1$ cycles, i.e. $3n - 1$ cycles if broadcast hardware is added. Cosnard et al.(2) proposed a systolic network which performs Gauss-Jordan elimination for the same time as in Ref. 1, but using $3n^2/8$ processing cells.

For arbitrary $n \times n$ matrix $A$ and $n \times m$ matrix $B$, Chan-Jy Lin (3) has proposed a systolic algorithm based on Gauss-Jordan elimination to solve linear system $AX = B$. A systolic array consists of $n(n + 1)$ PEs. The algorithm requires $4n + m - 2$ time steps to solve the linear systems. Seduhin and Tishenko(4) have considered the realization of Gauss-Jordan algorithm on highly parallel architecture suitable for digital signal processing. A parallel Gauss-Jordan algorithm suitable for implementation on pyramidal multiprocessor system was analyzed by Geus et al.(5) In all mentioned papers the algorithms for matrix inversion are implemented on 2D or 3D arrays of processing elements. Linear arrays are more suitable than two-dimensional arrays because they can balance the computation rate with the available I/O bandwidth to a sequential host for a large class of algorithms which come from linear algebra. Also, they enable the design of fault-tolerant arrays which is indispensable if the technique of wafer-scale integration (WSI) is needed.

This paper describes a parallel algorithm for inversion of dense matrices, of order $n$, based on a modified Jordan's method (see, e.g. Ref. 6), which requires fewer calculation steps than the standard one used by Cosnard et al.(7) The algorithm is proposed for implementation on the linear array with a small to moderate number of homogeneous processors such that the execution cost of a program module is the same on all processors. Processors operate in a parallel-pipeline fashion. The proposed parallel algorithm implies that a number of processors should be $1 < p < n/2$. With this number of processors the total execution time of the proposed algorithm (if communications are omitted) is

$$T_p = \frac{n^2}{p} (2n - 1)$$

Under the same conditions the execution time of the algorithm proposed by Cosnard and Daoudi(8) is

$$T_{JD} = \frac{3n^3}{p} + O \left( \frac{n^3}{p} \right)$$