하계학술대회
논문86-12-3

행렬연산을 위한 병렬계산 알고리즘

Parallel Computing Algorithm for Matrix Operations

이 장 규 서울대학교
김 형 중* 제어계측공학과

I. Introduction

Parallel computing algorithm should be devised to fully exploit the concurrency of matrix operations. To enhance the performance and utilization, computation loads should be distributed over the processors.

If one requires only the solution of the linear system

$$A\underline{x} = b \qquad (1)$$

where A is a nonsingular matrix of order n, one need not compute A^{-1}. Procedure involving A^{-1} always requires more work than the elimination reduction method. However, it happens in a problem that the elements of A^{-1} have some physical meaning.

Next, the Lyapunov equation

$$A^TX + XA = Q \qquad (2)$$

is frequently encountered in control engineering. The steady-state solution of the time-invariant Riccati equation

$$0 = A^TX + XA - XBR^{-1}B^TX^T + Q \qquad (3)$$

is of the type of Eq. (2), for which efficient methods of solution exist [1]. If A is asymptotically stable, a safe initial value is $X_o = 0$ for iterative method. If A is not asymptotically stable sophisticated methods [2-3] for choosing an initial value p_o are required.

Tridiagonal matrix can be decomposed by a cyclic reduction algorithm [4]. However, it requires much computation before decomposition. Therefore, one-shot decomposition algorithm is required.

All the algorithms proposed in the following Sections enhance the utilization and require minimum storage. Matrix inversion and decomposed linear system solver are suitable for SIMD machine. Lyapunov equation solver is suitable for synchronous MIMD machine.

Some algorithms are decomposed naturally for parallel processing over processors. Natural decomposition is inherent in a problem. Matrix inversion and Lyapunov equation are good example. However, linear system equations should be decomposed forcibly.

2. Matrix Inversion Algorithm.

Let X be the inverse matrix of a nonsingular matrix A of order n. Then the following relation holds.

$$AX = I \tag{4}$$

We denote by $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n$ the vectors formed by the first, second, ..., nth column of X, and analogously we define the unit vectors. Now the equation AX=I can be transformed by a linear system of the following form

$$\begin{bmatrix} A & & & \\ & A & & \\ & & \ddots & \\ & & & A \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \vdots \\ \underline{x}_n \end{bmatrix} = \begin{bmatrix} \underline{i}_1 \\ \underline{i}_2 \\ \vdots \\ \underline{i}_n \end{bmatrix} \tag{5}$$

Therefore, matrix inversion problem is replaced by n linear systems of equations, all with the same coefficient matrix :

$$A\underline{x}_r = \underline{i}_r \quad , \qquad r = 1, \ldots , n . \tag{6}$$

Above algorithm is due to Fröberg [5]. Eq. (4) can be decomposed naturally by Eq. (6) to be distributed over SIMD processors. Number of resultant linear system equations is dependent on the order of matrix A. All these systems in Eq. (6) have unique solution, since A is nonsingular.

3. Lyapunov Matrix Equation Solver

Lyapunov equation for algebraic Riccati equation is of the form

$$A^T X + XA = Q$$

where order of X is n. Lyapunov equation can be transformed into

$$\left(\begin{bmatrix} A & & & & \\ & A & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & A \end{bmatrix} + \begin{bmatrix} B_{11} & B_{21} & \cdots & B_{n1} \\ B_{12} & B_{22} & & \cdot \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ B_{1n} & \cdots\cdots & & B_{nn} \end{bmatrix} \right) \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \cdot \\ \cdot \\ \underline{x}_n \end{bmatrix} = \begin{bmatrix} \underline{q}_1 \\ \underline{q}_2 \\ \cdot \\ \cdot \\ \cdot \\ \underline{q}_n \end{bmatrix} \tag{7}$$

where

$$B_{ij} = \begin{bmatrix} b_{ij} & & \\ & b_{ij} & \\ & & b_{ij} \end{bmatrix}, \quad \underline{x}_i = \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \cdot \\ \underline{x}_n \end{bmatrix}, \quad \underline{q}_i = \begin{bmatrix} \underline{q}_{1i} \\ \underline{q}_{2i} \\ \cdot \\ \underline{q}_{ni} \end{bmatrix}$$

Eq. (7) can be converted into an equivalent system of the form

$$\begin{bmatrix} \underline{x}_1^{(k+1)} \\ \underline{x}_2^{(k+1)} \\ \cdot \\ \cdot \\ \cdot \\ \underline{x}_n^{(k+1)} \end{bmatrix} = \left(\begin{bmatrix} \tilde{A} & & & & \\ & \tilde{A} & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \tilde{A} \end{bmatrix} + \begin{bmatrix} \tilde{B}_{11} & \tilde{B}_{21} & \cdots & \tilde{B}_{n1} \\ \tilde{B}_{12} & \tilde{B}_{22} & \cdots & \cdot \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ B_{1n} & \cdots\cdots & & B_{nn} \end{bmatrix} \right) \begin{bmatrix} \underline{x}_1^{(k)} \\ \underline{x}_2^{(k)} \\ \cdot \\ \cdot \\ \cdot \\ \underline{x}_n^{(k)} \end{bmatrix} - \begin{bmatrix} \tilde{\underline{q}}_1 \\ \tilde{\underline{q}}_2 \\ \cdot \\ \cdot \\ \cdot \\ \tilde{\underline{q}}_n \end{bmatrix} \tag{8}$$

with the initial vector $\underline{x}^{(o)}$. Each processor only computes the following matrix equation in a synchronous fashion.

$$\underline{x}_r^{(k+1)} = \tilde{A}\underline{x}_r^{(k)} + \sum_{j=1}^{n} \tilde{b}_{jr}\underline{x}_j^{(k)} - \tilde{\underline{q}}_r \ , \qquad r=1, \ldots , n \tag{9}$$

Lyapunov equation can be decomposed naturally, and the number of resultant equations depends on the order of square matrix X.

4. Tridiagonal System Decomposition

Linear System

$A\underline{x} = b$

can be triangularized by the multiplication of P such as

$$P A \underline{x} = P \underline{b} . \qquad (10)$$

If we use P of the form

$$p = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & \ddots & & & & \\ & & & 1 & p_{ij} & & \\ & & & p_{ji} & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}, \; i < j.$$

where

$$p_{ij} = -\frac{a_{ij}}{a_{jj}} , \quad p_{ji} = -\frac{a_{ji}}{a_{ii}} ,$$

linear system of Eq. (1) can be decomposed into n sublinear system equations. If a_{jj} is zero, elementary row operation is required before decomposition. Then they are suitable for SIMD machine. Tridiagonal linear system equation can be forcibly decomposed with P.

REFERENCES

[1] H. Kwakernaak and R. Shivan, Linear Optimal Control Systems, John Wiley & Sons, 1972.

[2] F.T. Man and H.W. Smith, " Design of Linear Regulators Optimal for Time-Multiplied Performance Indices, " IEEE Trans. Autom. Control, Vol. 14, No. 5, pp. 527-529, 1969.

[3] D.L. Kleinman, " An Easy Way to Stabilize a Linear Constant System," IEEE Trans. Autom. Control, Vol. 15, No. 6, pp. 692, 1970.

[4] R.W. Hockney, " A Fast Direct Solution of Poisson's Equation Using Fourier Analysis, " J. ACM, Vol. 12, No. 1, pp. 95-113, 1965.

[5] C. Fröberg, Introduction to Numerical Analysis, Addison-Wesley, 1970.