PARALLEL ALGORITHMS FOR DIRECT SOLUTION OF LARGE SYSTEMS OF EQUATIONS

Jian-She Wang   Nathan Ida

Electrical Engineering Department
The University of Akron
Akron, Ohio 44325

ABSTRACT

A number of direct solution algorithms have been parallelized for use in conjunction with finite element analysis of large engineering problems. Parallel solution algorithms based on the Gauss-Jordan and Gauss elimination were implemented and compared. These parallel solvers are applied to large, dense or banded systems of equation arising from finite element analysis of 2-D and 3-D electromagnetic field problems. Both real and complex matrices are considered with emphasis on very large systems. The speedup obtained by parallelization on the MPP compared to sequential computers is almost three orders of magnitude. Although the MPP is used for implementation, most aspects of parallelization are general.

INTRODUCTION

In engineering applications it is often necessary to solve large systems of equations that are either too large or require too much computer resources to be economically feasible on standard computers. For this type of problem a parallel machine is very attractive. The type of systems considered are those arising from the application of the finite element method (FEM) to engineering applications. The FEM is particularly computationally intensive, yet its various parts are either intrinsically parallel or can be parallelized. By using a parallel processor, considerably faster solution times can be achieved or, alternatively, larger problems can be solved.

The Gauss elimination and the Gauss-Jordan methods have been chosen for this work because of their extensive use in finite element applications. In most cases, dense, nonsymmetric, real systems are solved but similar methods for banded and complex systems are presented. Sparse systems are not considered here although, these can obviously be handled.

The MPP has been described elsewhere [1,2] in detail. For the purpose of this work, the MPP is configured as an 128*128 array with a 32 bit word length. For the solution of linear systems, the two most important aspects related to the MPP are the number of memory planes in the ARray Unit (ARU) and the size of the staging memory. The ARU contains 900 usable bit planes of memory. This limits the number of real arrays (128*128, 32 bit) in the ARU to 28. The staging memory is limited to 512 real arrays. Parallel Pascal callable I/O procedures can transfer only one 128*128 array in or out of the ARU at any one time. This makes it necessary for any array larger than 128*128 to be blocked into sub-arrays of 128*128. Thus, the smallest system considered is a 128*128 system of equations.

A PARALLEL GAUSS-JORDAN ALGORITHM

For a system of equations of the form [A]{X}={B}, the parallel implementation of the Gauss-Jordan algorithm begins by loading [A] into one array and the right hand side (RHS) {B} into the first column of a second array. Assuming that the first column in [A] has been eliminated, these arrays look as:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ & a'_{22} & a'_{23} & \cdots & a'_{2n} \\ & a'_{32} & a'_{33} & \cdots & a'_{3n} \\ & \vdots & \vdots & & \vdots \\ & a'_{n2} & a'_{n3} & \cdots & a'_{nn} \end{bmatrix} \quad (1a) \qquad \begin{bmatrix} b_1 & 0 & 0 & \cdots & 0 \\ b'_2 & 0 & 0 & \cdots & 0 \\ b'_3 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ b'_n & 0 & 0 & & 0 \end{bmatrix} \quad (1b)$$

where n=128. The prime indicates that the corresponding coefficients have been modified during elimination of the first column. To eliminate the second column, a pivot row array and a pivot element array are created using row and column broadcasting routines.

$$\begin{bmatrix} 0 & a'_{22} & a'_{23} & \cdots & a'_{2n} \\ 0 & a'_{22} & a'_{23} & \cdots & a'_{2n} \\ 0 & a'_{22} & a'_{23} & \cdots & a'_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & a'_{22} & a'_{23} & \cdots & a'_{2n} \end{bmatrix} \quad (2a) \qquad \begin{bmatrix} a'_{22} & a'_{22} & \cdots & a'_{22} \\ a'_{22} & a'_{22} & \cdots & a'_{22} \\ a'_{22} & a'_{22} & \cdots & a'_{22} \\ \vdots & \vdots & & \vdots \\ a'_{22} & a'_{22} & \cdots & a'_{22} \end{bmatrix} \quad (2b)$$

A pivot column array is created from (1a) as

$$\begin{bmatrix} a_{12} & a_{12} & a_{12} & \cdots & a_{12} \\ a'_{22} & a'_{22} & a'_{22} & \cdots & a'_{22} \\ a'_{32} & a'_{32} & a'_{32} & \cdots & a'_{32} \\ \vdots & \vdots & \vdots & & \vdots \\ a'_{n2} & a'_{n2} & a'_{n2} & \cdots & a'_{n2} \end{bmatrix} \quad (3)$$

Eq. (3) is divided by Eq. (2b) and multiplied by Eq. (2a) to create a modifier array

$$\begin{bmatrix} 0 & a_{12}a'_{22}/a'_{22} & a_{12}a'_{23}/a'_{22} & \cdots & a_{12}a'_{2n}/a'_{22} \\ 0 & a'_{22}a'_{22}/a'_{22} & a'_{22}a'_{23}/a'_{22} & \cdots & a'_{22}a'_{2n}/a'_{22} \\ 0 & a'_{32}a'_{22}/a'_{22} & a'_{32}a'_{23}/a'_{22} & \cdots & a'_{32}a'_{2n}/a'_{22} \\ & \vdots & \vdots & & \vdots \\ 0 & a'_{n2}a'_{22}/a'_{22} & a'_{n2}a'_{23}/a'_{22} & \cdots & a'_{n2}a'_{2n}/a'_{22} \end{bmatrix} \quad (4)$$

This array, with the exception of the pivot row, is subtracted from the original array in (1a). The result is a new coefficient array

$$\begin{bmatrix} a_{11} & 0 & a''_{13} & \dots & a''_{1n} \\ 0 & a'_{22} & a'_{23} & \dots & a'_{2n} \\ 0 & 0 & a''_{33} & \dots & a''_{3n} \\ 0 & 0 & a''_{43} & \dots & a''_{4n} \\ \cdot & \cdot & \cdot & & \cdot \\ 0 & 0 & a''_{n3} & \dots & a''_{nn} \end{bmatrix} \quad (5)$$

The modification of the RHS during elimination is similar. Eq. (3) is divided by Eq. (1a) and multiplied by Eq. (1b) to generate an RHS modifier array This is subtracted from Eq. (1b) to obtain the new RHS array.

$$\begin{bmatrix} b_1 - a_{12}b'_2/a'_{22} & 0 & 0 & \dots & 0 \\ b'_2 - a'_{32}b'_2/a'_{22} & 0 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & & \cdot \\ b'_n - a'_{n2}b'_2/a'_{22} & 0 & 0 & \dots & 0 \end{bmatrix} \quad (6)$$

After n elimination steps, the original coefficient matrix is reduced to a diagonal system. To obtain the solution, an array of the diagonals is constructed

$$\begin{bmatrix} a_{11} & a_{11} & a_{11} & \dots & a_{11} \\ a'_{22} & a'_{22} & a'_{22} & \dots & a'_{22} \\ a''_{33} & a''_{33} & a''_{33} & \dots & a''_{33} \\ \cdot & \cdot & \cdot & & \cdot \\ a''_{nn} & a''_{nn} & a''_{nn} & \dots & a''_{nn} \end{bmatrix} \quad (7)$$

Eq. (6) is divided by Eq. (7), to obtain the unknowns x1 through x128:

A PARALLEL GAUSS ELIMINATION ALGORITHM

The Gauss elimination algorithm follows similar steps. The steps in Eq. (1) through (4) are identical. In subtracting the modifier array in Eq. (4) from Eq. (1a), only the rows below the pivot row are modified. After (n-1) elimination steps, the original system (1a) is reduced to an equivalent upper triangular system: The right hand side is similarly modified.

$$\begin{bmatrix} a_{11} + & a_{12} + & a_{13} + \dots + & a_{1n} \\ & a'_{22} + & a'_{23} + \dots + & a'_{2n} \\ & & a''_{33} + \dots + & a''_{3n} \\ & & & \cdot \\ & & & a^{n}_{nn} \end{bmatrix}_{8a} \begin{bmatrix} b_1 & 0 & 0 & \dots & 0 \\ b'_2 & 0 & 0 & \dots & 0 \\ b''_3 & 0 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & & \cdot \\ b^{n}_n & 0 & 0 & & 0 \end{bmatrix}_{8b}$$

The solution of the system in Eq. (8) is performed using the following algorithm

$$x_i = b_i/a_{ii}, \quad b_k = b_k - a_{ki}x_i \quad (9)$$

where i=n,n-1,...,1 and k=i-1,i-2,...,1. In this algorithm, once an unknown is backsubstituted, the upper triangular system is reduced in order by one and then the RHS is modified.

A pivot column and a pivot element array are created as

$$\begin{bmatrix} a_{1i} & a_{1i} & a_{1i} & \dots & a_{1i} \\ a_{2i} & a_{2i} & a_{2i} & \dots & a_{2i} \\ a_{3i} & a_{3i} & a_{3i} & \dots & a_{3i} \\ \cdot & \cdot & \cdot & & \\ a_{ii} & a_{ii} & a_{ii} & \dots & a_{ii} \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}_{10a} \quad \begin{bmatrix} a_{ii} & a_{ii} & a_{ii} & \dots & a_{ii} \\ a_{ii} & a_{ii} & a_{ii} & \dots & a_{ii} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ a_{ii} & a_{ii} & a_{ii} & \dots & a_{ii} \end{bmatrix}_{10b}$$

The RHS is divided by the pivot element array (masked operation) to solve for the ith unknown. From this, an RHS pivot array is generated. These arrays are:

$$\begin{bmatrix} b_1 & 0 & 0 & \dots & 0 \\ b_2 & 0 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ x_i & 0 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & & \cdot \\ x_n & 0 & 0 & & 0 \end{bmatrix} \quad \begin{bmatrix} x_i & 0 & 0 & \dots & 0 \\ x_i & 0 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ x_i & 0 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ x_i & 0 & 0 & \dots & 0 \end{bmatrix} \quad (11)$$

Multiplication of Eq. (11b) by Eq. (10b) results in a modifier array:

$$\begin{bmatrix} a_{1i}x_i & 0 & 0 & \dots & 0 \\ a_{2i}x_i & 0 & 0 & \dots & 0 \\ a_{3i}x_i & 0 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ a_{ii}x_i & 0 & 0 & \dots & \cdot \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad (12)$$

The modifier array in Eq. (12) is now subtracted from the RHS. After n=128 steps, the RHS array contains the n unknowns in its first column.

BLOCK GAUSS-JORDAN AND GAUSS ELIMINATION

For the solution of any system with order larger than 128, the coefficient matrix is blocked in subarrays of 128*128. For each subarray the algorithm described in Eq. (1) through (7) is applied. A 512*512 system is chosen as an example since this is the largest array the ARU can handle. Any larger matrices will have to utilize the stager. In Fig. 1, the 512*512 coefficient matrix is blocked into 4*4 subarrays, while the RHS vector is stored in the first column of 4 corresponding subarrays or in the first four columns of one sub-array.

The Gauss elimination solution for a 512*512 system is similar to that of the Gauss-Jordan method described above other than the obvious changes described in Eq. (8) through (12).

Table 1 summarizes the number of operations required for solution on a sequential machine and on the MPP. Table 2 summarizes the solution times for a 128*128 and a 512*512 system of linear equations using the Gauss-Jordan and Gauss elimination methods on the MPP. The results are compared with those obtained for the same systems on a MicroVaxII computer. The highest speedup is achieved for a 512*512 system (largest problem that can reside in the ARU). The backsubstitution is the slowest of the two parts (essentially a sequential operation).

SOLUTION OF BANDED SYSTEMS OF EQUATIONS

Fig. 2 shows the coefficient data structure of a 512*512 system of equations arising from finite element analysis, where only the shaded area has non-zero terms (semi-bandwidth less than or equal to 128). In Fig. 2a, 37.5% of the memory storage can be saved by considering only the non-zero blocks. In fig. 2b, 65.6% of the memory can be saved. With this storage scheme, a 1024*1024 matrix with a semi-bandwidth of 128 can reside in the ARU.

The solution times for a parallel, banded elimination algorithm are summarized in Table 3 and compared with those for full coefficient matrix of the same order (512*512).

SOLUTION OF SYSTEMS WITH COMPLEX COEFFICIENTS

Application of the finite element method to the solution of eddy current problems in electromagnetic fields results in the following system of complex linear equations:

$$(A + jB)(X + jY) = (C + jD) \quad (13)$$

On the MPP, complex data is stored in two sets of arrays. Complex calculations are resolved into two or more real parallel array operations. The basic operations required are implemented as:

$$\begin{aligned} &P+Q=(P1+Q1)+j(P1+Q1) \quad P-Q=(P1-Q1)+j(P1-Q1) \\ &P*Q=(P1*Q1-P2*Q2)+j(P1*Q2+P2*Q1) \quad (14) \\ &P/Q=((P1*Q1+P2*Q2)/(Q1*Q1+Q2*Q2)) \\ &\quad +j((P2*Q1-P1*Q2)/(Q1*Q1+Q2*Q2)) \end{aligned}$$

For the solution of a system of complex linear equations with order higher than 128, the complex coefficient matrix is blocked into subarrays of 128*128.

The solution times for a 128*128 and a 256*256 system of complex linear equations by the Gauss-Jordan and Gauss elimination methods are summarized and compared with those for solution of the same order system of real equations on the MPP. The results are shown in Table 4.

The solution time for a system of complex linear equations by Gauss' and Jordan's methods is about 4 to 5 times that needed to solve the same order system of real equations.

SOLUTION OF LARGE SYSTEMS OF EQUATIONS

For problems of size larger than the capacity of the ARU, the stager must be used. The matrix is again subdivided into blocks of 128*128. Once all subarrays in the matrix are in correct stager addresses, part of the arrays are sent to the ARU for processing. The results are returned to the same stager addresses. This is repeated until the system is solved.

The division into subarrays is the same as in Fig. 2b for banded systems and as in Fig. 1 for dense systems except for the larger number of subarrays required. The RHS is placed in columns of a single array to save space.

Several banded systems with bandwidth≤128 of selected order ranging from 1024 to 16,384 have been solved on the MPP using the stager. The solution time (including data transfer between stager and ARU) is shown in Table 5. Table 6 gives the largest banded systems with different bandwidth that can be solved on the MPP under the limit of the stager size (32 Mb). Table 7 summarizes the solution times for two large, dense systems of equations (nonsymmetric).

CONCLUSIONS

The implementation of solution algorithms on a massively parallel processor is quite efficient as long as the system fits in the ARU. Larger systems can also be solved with reduced efficiency. Even so, the solution is as fast or faster than on vector machines. An increase in size of the array and local memory could significantly improve performance.

REFERENCES

[1] K.E. Batcher, "Design of a massively parallel processor ", IEEE Transactions on Computers, Vol. C-29, No.9, pp. 837-840, September 1980.

[2] K. E. Batcher, "Architecture of the MPP" IEEE Computer Society on Computer Architecture for Pattern Analysis and Image Database Management Proceedings, pp. 170-174, October 1983.

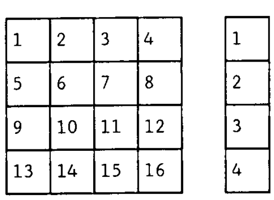

Figure 1 Blocking of a 512*512 system.

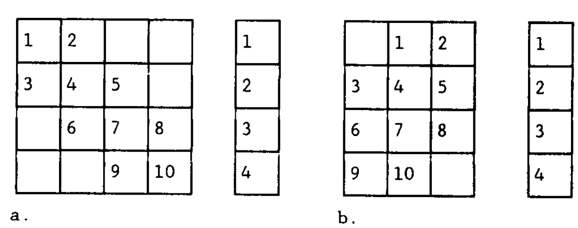

Figure 2. Two methods of blocking a 512*512 system with semi-bandwidth of 128 or smaller

Table 1. Number of operations needed for sequential and parallel solution with the Gauss-Jordan and Gauss elimination algorithms.

		Sequential solution			Parallel Solution		
Method	Oper.	Diag./Triag.	Solution	Total	Diag./Triag.	Solution	Total
Gauss Jordan	A/M D	n(n-1)(n+3)/2 n(n-1)	0 n	n(n-1)(n+3)/2 n^2	amm(m+3)/2 amn	 m	amm(m+3) + a/2 m(am+1)
Gauss Elim.	A/M D	n(nn-1)/3 n(n-1)/2	n(n-1)/2 n	n(n-1)(2n+5)/6 n(n+1)/2	am(m+1)(2m+1)/6-1 am(m+1)/2-1	am(m+1)/2-1 am	am(m+1)(m+2)/3-1 am(m+3)/2-1

A=add, M=multiply, D=divide, a=128, m=n/128, n=# of equations in the system.

Table 2. Comparison of solution times for the Gauss-Jordan and Gauss elimination methods on the Microvax II and the MPP. (Times in seconds).

	Gauss-Jordan			Gauss Elimination		
Order	μVax	MPP	Speedup	μVax	MPP	Speedup
128:						
Elim.	11.57	.07788	148	7.57	.07795	97
Sol.	0.01	.00643	1.6	.12	.0493	2.4
Total	11.58	.08431	137	7.59	.12927	59
512:						
Elim.	3476	1.728	2011	3165	1.231	2572
Sol.	0.23	.02568	8.9	2.01	.25615	5.6
Total	3476	1.754	1982	3169	1.588	1994

Table 3 Banded and full matrix solution times using Gauss elimination on the MPP. (in miliseconds).

Step	Full Matrix	Banded Matrix	Speedup
Elimination	1230.56	724.26	1.699
Solution	356.15	277.44	1.284
Total	1585.70	1001.71	1.584

Table 4. Comparison of solutions in real and complex variables on the MPP. (miliseconds).

	Gauss Jordan			Gauss Elimination		
Order	Real	Complex	Ratio	Real	Complex	Ratio
128:						
Elim.	77.88	334.89	4.3	77.95	334.96	4.3
Sol.	6.43	12.43	1.9	49.30	244.58	4.96
Total	84.31	347.32	4.1	128.27	579.55	4.52
256:						
Elim.	343.68	1513.80	4.4	277.76	1196.30	4.3
Sol.	12.28	24.88	2.0	125.61	576.48	4.59
Total	355.96	1538.68	4.3	403.37	1772.77	4.4

Table 5. Solution times on the MPP for banded systems of different sizes. Semi-bandwidth is 128. (Time is in seconds).

Size	Elimination	Solution	Total
1024	1.556	0.651	2.206
2048	3.300	1.371	4.671
3072	5.015	2.077	7.093
4096	6.731	2.784	9.515
8192	13.594	5.611	19.205
12288	20.458	8.438	28.896
16384	27.321	11.264	38.264

Table 6. Largest systems solvable on the MPP

Semi-bandwidth	Size of Systems
128	21888
256	13184
384	9472
512	7552
1024	4352
2048	2944
2816	2816

Table 7. Solution of large, dense, nonsymmetric systems on the MPP. Time is in seconds.

Size	Elimination	Solution	Total	CRAY X/MP
1024x1024	7.809	1.327	9.136	36.523
2048x2048	51.353	4.638	55.990	——