Machine Independent Algorithm for Concurrent Finite-Element Problems

W.J. BUCHANAN & N.K. GUPTA.

Department of Electrical, Electronic and Computer Engineering, Napier University, Edinburgh, UK.

ABSTRACT

This paper describes how parallel processing applies to finite-element simulations. The methods discussed are the Parallel Gaussian Elimination and the Parallel Active Column Solver. Both methods reduce the time taken to determine the global coefficient matrix. The paper discusses the suitability of row- and column-based approaches when applying the Parallel Gaussian method to parallel processing. The Parallel Active Column Solver uses a skyline storage technique.

Introduction

The finite-element method was initially developed for mechanical and civil engineering applications. Over the years it has since been applied to electromagnetics [1-3]. One problem with it is that the simulation time becomes relatively long when applied to large complex problems. This paper discusses methods in reducing this time by applying parallel processing.

A system may seem complex when viewed over a large region, but when split into smaller sections its behaviour can be easily approximated. In the finite-element method the total region divides into a number of non-overlapping sub-regions, called finite elements. In two dimensions, simple polygons, such as triangles and/or squares, make-up the elements. Figure 1 shows a region divided into triangles.

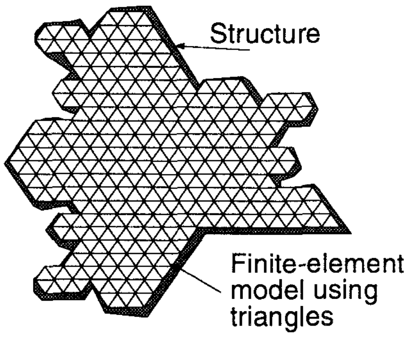

Figure 1: Conversion of a structure into finite-elements

Regardless of the element shape, the field is approximated with a different expression for each element. Where adjoining edges meet, the field representations must maintain field continuity. Normally the equations to be solved are stated in terms not of the field variables but of an integral-type function such as energy. The field solution then makes this function stationary. The total function is the sum of the integral over each element.

This paper investigates two methods that solve finite-element problems using parallel processing. The techniques used are Gaussian elimination using banded storage and active column equation solver using skyline storage.

The banded storage is a row-oriented and the skyline method uses a column-oriented approach.

Finite element method

The finite-element method uses small sub-regions to simplify the solution. Each sub-region is an element of the global structure.

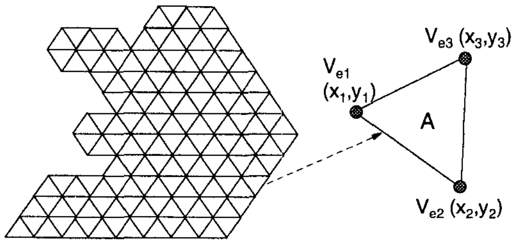

Figure 2: Triangular finite-elements

The potential V_e within an element, e, is the summation of the effect of each element on that element. As an approximation over the whole region, V(x,y) can be given by:

$$V(x,y) \approx \sum_{e=1}^{N} V_e(x,y)$$

where N is the number of triangular elements. The approximation of V for a triangular element can be expressed as:

$$V_e(x,y) = a + bx + cy$$

It can be shown [1-3] that the energy associated with an element e is

$$W_e = \frac{1}{2}\varepsilon[V_e][C^{(e)}][V_e]^T$$

where, regarding Figure 2, $[V_e]$ represents the voltage matrix:

$$[V_e] = \begin{bmatrix} V_{e1} \\ V_{e2} \\ V_{e3} \end{bmatrix}$$

Computation in Electromagnetics, 10–12 April 1996,
Conference Publication No. 420, © IEE, 1996

and the element matrix is:

$$\left[C^{(e)}\right]=\begin{bmatrix} C_{11}^{(e)} & C_{12}^{(e)} & C_{13}^{(e)} \\ C_{21}^{(e)} & C_{22}^{(e)} & C_{23}^{(e)} \\ C_{31}^{(e)} & C_{32}^{(e)} & C_{33}^{(e)} \end{bmatrix}$$

Matrix [C] is the element coefficient matrix. The matrix element $C_{ij}^{(e)}$ of the coefficient matrix is the coupling between nodes i and j. Its value is obtained from the location of the points of the triangle, for example for a triangular element:

$$C_{21}^{(e)}=C_{12}^{(e)}=\frac{1}{4A}\left[(y_2-y_3)(y_3-y_2)+(x_3-x_2)(x_1-x_3)\right]$$

$$C_{13}^{(e)}=C_{31}^{(e)}=\frac{1}{4A}\left[(y_2-y_3)(y_1-y_2)+(x_3-x_2)(x_2-x_1)\right]$$

$$C_{23}^{(e)}=C_{32}^{(e)}=\frac{1}{4A}\left[(y_3-y_1)(y_1-y_2)+(x_1-x_3)(x_2-x_1)\right]$$

$$C_{11}^{(e)}=\frac{1}{4A}\left[(y_2-y_3)^2+(x_3-x_2)^2\right]$$

$$C_{22}^{(e)}=\frac{1}{4A}\left[(y_2-y_3)^2+(x_3-x_2)^2\right]$$

$$C_{33}^{(e)}=\frac{1}{4A}\left[(y_1-y_2)^2+(x_2-x_1)^2\right]$$

The energy associated with the summation of all the mesh elements is:

$$W=\sum_{e=1}^{N}W_e=\frac{1}{2}\varepsilon[V][C][V]^T,$$

where, $$\left[V\right]=\begin{bmatrix} V_1 \\ . \\ . \\ . \\ V_n \end{bmatrix}$$

and n is the number of nodes, N is the number of elements, and [C] is the global coefficient matrix. This matrix contains the individual assemblage of element coefficients. The example in the Figure 3 has 5 points on the mesh. This leads to a 5x5 global coefficient matrix in the form:

$$[C]=\begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} \end{bmatrix}$$

The C_{ij} factor is the coupling between the nodes i and j. The first five coefficients of the global coefficient matrix are:

$$C_{11}=C_{11}^{(1)}+C_{11}^{(2)},\ C_{12}=C_{13}^{(1)},\ C_{13}=C_{12}^{(2)}$$

$$C_{14}=C_{12}^{(1)}+C_{13}^{(2)},\ C_{15}=0$$

Matrix C has certain qualities that makes it easy to fill and some of the terms become zero, these are:

1. that the matrix is symmetrical ($C_{ij}=C_{ji}$) just as the element matrix;
2. the matrix is sparse and banded because Cij is zero when no coupling exists between nodes i and j.

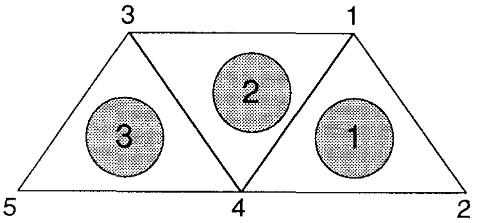

Figure 3: Connection of finite-elements

The solution of the voltage potentials can either be iterative or can use the band matrix method. With the band matrix method, a node has either a fixed potential (f) or a free potential (p). The applied electric field sets the fixed potentials. To solve for the potentials the energy is written in terms of the fixed potentials (V_f) and the free potentials (V_p), to give:

$$W=\frac{1}{2}\varepsilon[V_f \quad V_p]\begin{bmatrix} C_{ff} & C_{fp} \\ C_{pf} & C_{pp} \end{bmatrix}\begin{bmatrix} V_f \\ C_p \end{bmatrix}$$

The energy change with respect to the voltage potential will tend to zero and since the fixed potentials (V_f) are constant then:

$$\left[C_{ff} \ C_{fp}\right]\begin{bmatrix} V_f \\ V_p \end{bmatrix}=0$$

and thus

$$\left[C_{ff}\right]\left[V_f\right]=-\left[C_{fp}\right]\left[V_p\right]$$

This is in the form of simultaneous equations, $[A][V]=[B]$, and therefore can be solved using a Gaussian elimination technique. The global coefficient matrices will both be symmetrical, as illustrated in Figure 4.

A solution could also be found by determining the inverse of the matrix C_{ff} and using

$$\left[V_f\right]=-\left[C_{ff}\right]^{-1}\left[C_{fp}\right]\left[V_p\right]$$

This method may be impractical if the global coefficient matrix C_{ff} is large.

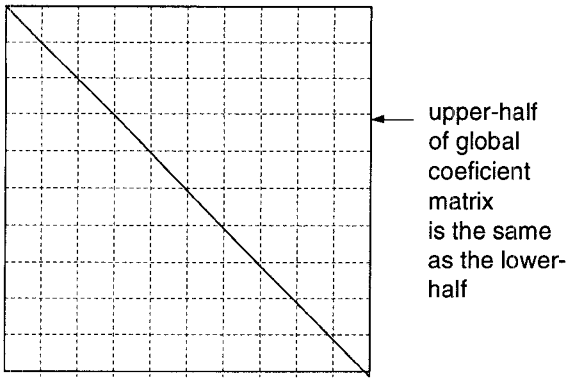

Figure 4: Symmetrical coefficient matrix

Parallel Gaussian Elimination Technique

The Gaussian elimination technique allows the use of parallelism with row- or column-oriented algorithms. For the row-oriented approach, each processor holds a set of rows. In the column-oriented approach each processor holds a set of columns. For both methods a round-robin assignment allocates each processor with either a row or column. This helps to equalize the computational load on them.

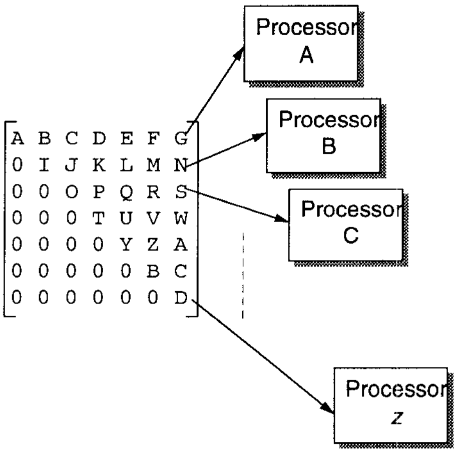

Figure 4: Allocation of rows to processors

The row-oriented approach uses a pivot row. This row contains the pivot diagonal entry of the global coefficient matrix and is used to reduce the rows below it. Processors are assigned the rows below the pivot row and perform arithmetic operations concurrently on their respective rows, as illustrated in Figure 4. The column entries below the pivot diagonal entry become zero simultaneously at the end of each step of forward reduction. The forward reduction is performed from left to right until the global coefficient matrix reduces to an upper triangular form. Unknowns are then determined using back substitution.

In the column-oriented approach the vector of multipliers is calculated for each forward reduction step. Then all processors access this vector and their respective pivotal row element to perform arithmetic operations, column-wise concurrently, using the appropriate multiplier.

Parallel Active Column Solver

This algorithm is based on work carried out by Farhat [4] and uses the upper triangular part of the global coefficient matrix to store columns using a skyline storage scheme. The columns are distributed in round-robin fashion among the processors. The solution takes the form:

$$[A]=[V][B]$$

for the solution the jth column of [A] is

$$A_{ij}=B_{ij}-\sum_{i=1}^{i-1}\frac{A_{ji}}{A_{ii}}A_{jk}$$

The factorization process starts with the first row and calculates the rows of the matrix [*A*] using the above equation. This continues until the final row. The calculation of the elements of row i requires column *j=i* which, at that time, is called the active column. All processors access the elements of the active column and calculate the elements of row *i* concurrently.

The solution of [*V*] can then be found using forward substitution. The algorithm given next shows the steps for determining the free voltage point [*V*] for an equation in the form [*A*][*V*]=[*B*].

```
for i=1 to n do
begin
   for j=1 to i-1 do
   begin
      scalar[j]=A[j][i]/A[i][i]
   end

   mult=B[i]/A[i][i]

   for k=i to n do
   begin
      sum=0
      for j=1 to i-1 do
      begin
         sum=sum+scalar[j]*A[j][k]
      end

      A[j][k]=A[i][k]-sum

      if (k>i) B[k]=B[k]-mult*A[i][k]
   end
end

for k=n downto 2 do
begin
   mult=B[k]/A[k][k]

   for j=k-1 downto 1 do
   begin
      B[j]=B[j]-A[j][k]*mult
   end
end
```

```
for i=1 to n do
begin
  V[i]=B[i]/A[i][i]
end
```

Conclusion

The paper shows the application of parallel processing in finite-element simulations. Parallel processing reduces the time taken to determine the global coefficient matrix.

For the Parallel Gaussian technique, the column-oriented approach works well on shared memory systems but may cause problems on local memory systems. As only the upper half of the global coefficient matrix is usually stored and operated upon, the vector of multipliers cannot be computed by one processor without inter-processor communication. Thus for portability across shared and local memory system the row-oriented approach is preferable.

In initial tests it has been found that the parallel row-oriented technique is slightly faster than the active column solver.

Reference

[1] P.P Silverster and R.L. Ferrari "Finite Elements for Electrical Engineers", Cambridge University Press, New York, 1983.

[2] C.W. Steele, "Numerical Computation of Electric and Magnetic Fields", Van Nostrand Rienhold, New York, 1987.

[3] A.J Davies, "A First Course in the First Element Method", Oxford University Press, Oxford, 1980.

[4] C Farhat, E Wilson, "A parallel active column equation solver", Computers and Structures, Vol. 28, No. 2, 1988.