6. DISTRIBUTED STATIC FINITE ELEMENT ANALYSIS

6.1 GENERAL REMARKS

Analytical solutions can be obtained only for certain simplified situations but for problems having complex material properties and boundary conditions, numerical methods provides approximate but acceptable solutions [104,105]. The growth in finite element method which one of the most popular numerical methods is because of continuous developments in high-speed electronic digital computers and growing need of numerical methods for solving complex problems.

The finite element method (FEM) is based on representing a body or a structure as an assemblage of subdivisions known as finite elements. These elements are interconnected at joints, known as nodes or nodal points. Simple functions are chosen to approximate the distribution of actual displacements over each finite element, which are known as displacement functions. A variational principle of mechanics i.e. principle of minimum potential energy is used to obtain equilibrium equation of each element. After combining equilibrium equation of individual elements, solution yields the approximate displacements at discrete locations in the body and finally the strains and stresses at desired points.

A variety of examples of static finite element analysis have been implemented over distributed computing environment in this Chapter to understand application development using WebDedip, accuracy of solution, computational efficiency in various types of problems, and factors affecting speedup. Plane stress and plate bending problems are considered in subsequent sections. As the finite element formulations for these problems is simple and is available in standard books [106 to 113], it is not repeated here.

6.2 THE STEP-BY-STEP FEM PROCEDURE

- Divide the continuum, which is physical body or structure or solid, to be analyzed into an equivalent system of finite elements known as discretization. The element can be one dimensional (spring, bar, beam), two dimensional (triangular, rectangular, quadrilateral) or three dimensional (tetrahedral, hexahedral).

- Assume suitable displacement function to represent approximately the distribution of displacement within an element. The displacement function can be in the form of polynomial depending on type of problem and satisfying certain requirements known as convergence criteria.
- Derive element properties, like stiffness matrix and load vector, by applying principle of virtual work or minimization of potential energy. The stiffness matrix for an element depends on displacement model, geometry of element and material properties.
- Assemble stiffness matrix and load vector of all elements to have overall equilibrium of structure. The overall equilibrium relation between total stiffness matrix, total load vector and nodal displacements are expressed as a set of simultaneous equations.
- Apply geometric boundary conditions arising from supports of structure to overall equilibrium equations using a suitable method.
- Solve the algebraic equations to get displacements at nodal points using any one of the suitable solution techniques such as banded, skyline or frontal solution technique. Displacements are primary unknowns in solution of structural mechanics problem.
- Calculate element strains and stresses, which are known as secondary unknowns, from nodal displacements. The stresses are generally used for design purpose.

6.3 PARALLEL COMPUTING IN FINITE ELEMENT METHOD

Evaluation of structural response of large and complex structures modeled with several thousand finite elements involves millions of number crunching operations. Static, dynamic or nonlinear analysis of such structures requires large computational time. Historically, the applicability of the finite element method to complex three dimensional domains has been greatly restricted by two physical constrains of the computing environment, namely, the large physical memory required for storing a complex discretized model and large solution time required due to limited computational speed of the machine. At present, because of the availability of inexpensive physical memory the problem of memory size is not perceived to be as crucial as the one of computational speed. In a large size finite element problem, the solution algorithm often takes about 90-95% of the total computational time , hence efforts to reduce this time may greatly reduce the computational cost as well as solution turnaround time. Accordingly current research is increasingly focusing on adopting proper strategy for the various stages of FEM processing that take advantage of new computer architectures to achieve higher computational speed.

Machine that provides computational speed higher than those available on standard, single processor machine incorporate one or more of the following:

(a) A very fast CPU
(b) Special hardware for vectorization, and
(c) Special hardware for concurrent processing

Efficient algorithmic design can significantly reduce the computational time by taking advantage of vectorization and concurrent processing. Vectorization can be exploited whenever the same operation is performed on all elements of an array e.g. multiplication of an array by a scalar. Concurrent or parallel processing applies when a particular task can be subdivided in a set of subtasks to be executed in parallel on separate processors. Vectorization involves lower level of modification while concurrent processing requires changes at higher level.

Consider an example of solution of linear equations **K X** = **P**, where **K** is stiffness matrix, **P** is force vector and **X** is the vector of the unknown displacements. If direct implementation of Gaussian elimination is used, because of its sequence of operations on vector each representing a row or column of **K**, vectorization can be exploited with little or no change. In contrast, to take advantage of parallel processing, not only the algorithm must be drastically changed but also the problem must be recast in a different form.

The analysis of large structures using finite element, involving thousands of unknowns, can be expedited by subdividing it into smaller parts referred to as substructures. In substructure technique, each substructure is analyzed separately and the results are combined to yield the displacements and stresses in actual structure. To speed up further the analysis time, the parallel processing approach can be used.

Parallel implementation requires parallel machine consisting of multiple processors connected with each other and sharing the same memory. As parallel processing hardware designed exclusively for dedicated parallel processing is more expensive, an inexpensive alternative is to develop application software, which can run on network of workstations known as distributed processing. This arrangement employs a number of computers physically linked to permit online computer-to-computer communication. Using this approach of distributed processing, independent workstations interconnected by network and message passing for the communication of data between the computers can be transformed into cost effective parallel computing resource.

In the present chapter application of distributed processing in finite element analysis over network of computer is discussed with a variety of examples. To implement distributed computing in finite element method using LAN, WebDedip environment as discussed in Chapter 4 is used which is developed using client-server paradigm of distributed processing and JAVA technology. Substructure technique is used in finite element analysis [104] and entire application is divided into small tasks like preparation of substructure data, calculation of stiffness matrix and load vector corresponding to interface DOF, assembly of stiffness matrices and load vector of all substructure and calculation of displacements of interface DOF, calculation of internal DOF and element stresses from interface DOF. These tasks are executed on different computers and communication between them is done through FTP. Derivation of substructure stiffness matrix and load vector depends on number of internal DOF and interface DOF, while communication between computers depends on number of interface DOF. The effects of these parameters have been discussed by subdividing entire structure into number of substructures.

6.4 SUBSTRUCTURE TECHNIQUE IN FINITE ELEMENT ANALYSIS

Application of substructure technique in the analysis of skeletal structures was discussed in chapter 5. Similarly, the substructuring technique can be implemented for Finite Element Analysis of structures [114]. Using basic approach of the substructuring technique analysis can be carried out in five phases. Various processes related with these phases can be summarized as follows:

1) Data generation for individual substructure.
2) Calculation of substructure stiffness matrix and load vector corresponding to boundary degrees of freedom (DOF), using concept of static condensation of internal DOF.
3) Calculation of displacements at boundary nodes for each substructure after assembling substructure stiffness matrices and load vectors.
4) Calculation of displacements at internal degree of freedoms and element stresses for each substructure using boundary displacements.
5) Collection of results for each substructure and giving final result.

Out of the above five processes, first process of data generation is a sequential process so carried out on single computer. After generating data for each substructure, stiffness matrices and load vectors for each substructure can be calculated simultaneously on different computers. Each computer is having required data corresponding to the substructure allotted to that computer. Again the third process of assembly of matrices is a sequential process (i.e. carried out on single computer). So the completion of second process on all computers is necessary before beginning of third process. After calculation of displacements corresponding to boundary nodes, these displacements are distributed again to corresponding computers. So the fourth process is again the parallel process. And finally, after calculating displacements corresponding to internal DOFs, results are collected form each computer to have final results. This step-by-step procedure is explained as below with the help of suitable example.

STEP 1: Fig 6.1 show a plate, which is fixed at one end and subjected to tensile force on the other end. It is required to carry out the finite element analysis. For analysis CST element is used. The Fig 6.1 shows the discretization of plate into CST elements. For illustration plate is discretized into 32 number of elements and resulting in total 25 number of nodes.

In order to carryout the analysis using substructure technique, the plate is divided into 4 number of substructures. Fig 6.2 shows the plate divided into four substructures. For calculation purpose it is necessary to differentiate internal and boundary nodes. Here, each substructure is having 1 internal node and 8 boundary nodes. Node 7, 9, 17 and 19 is the internal node for 1st, 2nd, 3rd and 4th substructures respectively. Along with the internal and boundary nodes, other data such as the restrained conditions and the load data are also supplied individually for each substructure.

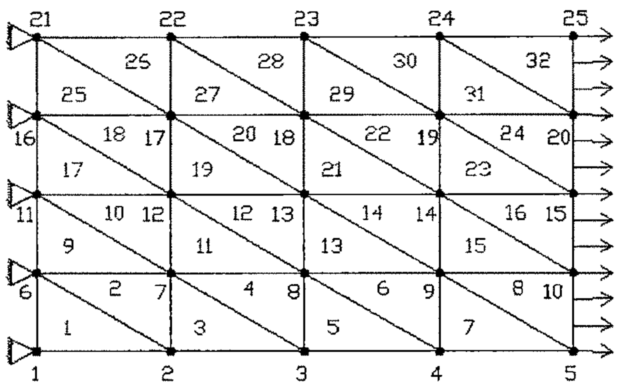


FIG. 6.1 STRUCTURE DISCRETIZED INTO 32 CST ELEMENTS

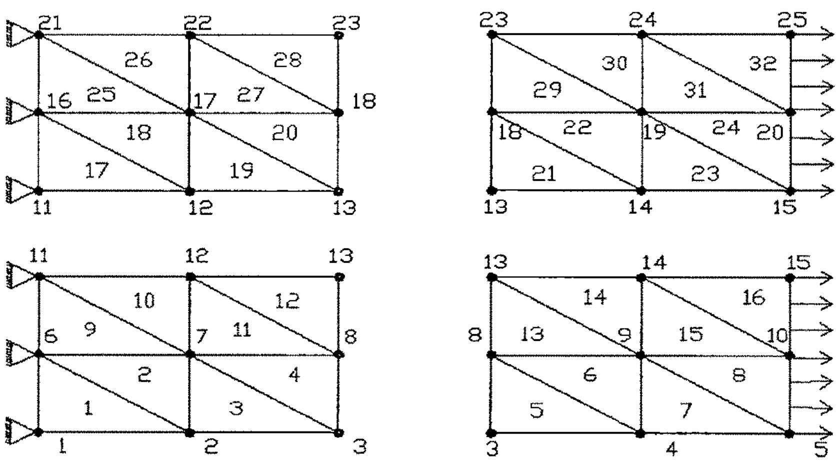


FIG. 6.2 SUBSTRUCTURE DIVIDED INTO FOUR SUBSTRUCTURES

STEP 2: As discussed in the theory of the substructure technique, first the analysis is carried out by restraining the boundary of each substructure. Fig 6.3 shows all substructures with restrained boundaries. In this step of analysis, nodes are renumbered. As shown in Figure, first all the internal nodes are numbered and then boundary nodes are numbered.

After renumbering of nodes, static condensation is carried out in which internal nodes are eliminated from each substructure. Calculation of condensed stiffness matrix and load vector is carried out. So the whole substructure will be treated as a single element. Fig 6.4 shows DOF corresponding to boundary nodes for each substructure. Here actual restrained conditions are also implemented as all substructure stiffness matrices and load vectors are to be assembled to get global stiffness matrix.

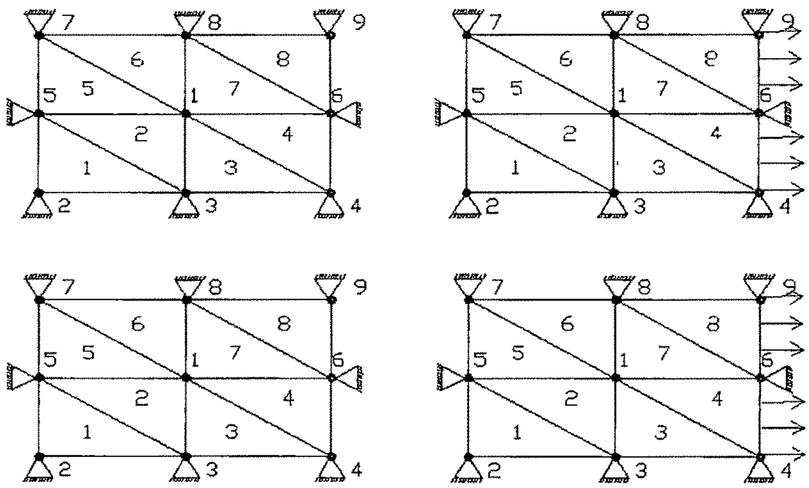


FIG. 6.3 SUBSTRUCTURES WITH ALL BOUNDARIES FIXED

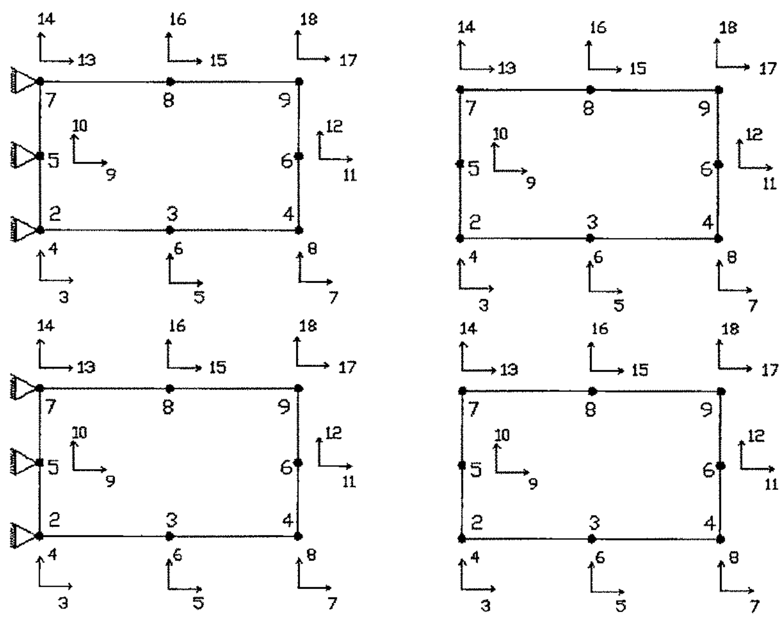


FIG. 6.4 SUBSTRUCTURES AFTER STATIC CONDENSATION

STEP 3: After calculation of condensed stiffness matrices and load vectors for each substructure, these matrices are assembled according to the boundary nodes. Fig 6.5 shows the nodes and corresponding DOF, which are incorporated in the global stiffness matrix. Nodes 1,6,11,16 and 21 are eliminated, as retrained and also internal nodes are not considered. After assembly of stiffness matrix and load vector, displacements corresponding to boundary nodes (shown in fig) are calculated.

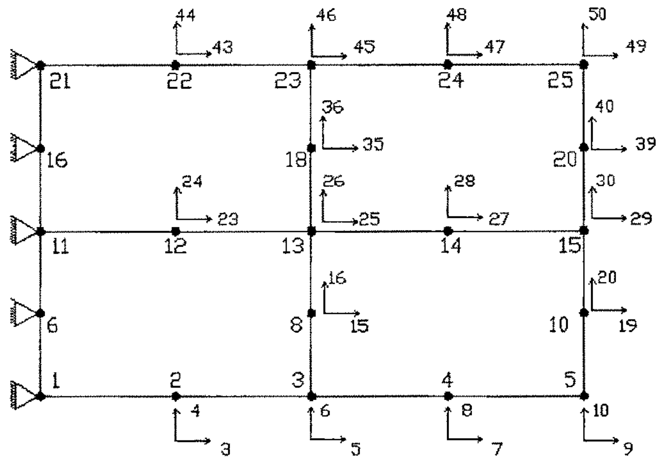


FIG. 6.5 SUBSTRUCTURES COMBINED AFTER STATIC CONDENSATION

STEP 4: After calculating displacements corresponding to boundary nodes, these are again distributed to corresponding substructure to calculate internal displacements. Fig 6.6 shows all substructures with internal and boundary displacements. In this stage of analysis, internal displacements are calculated from boundary displacements and finally element stresses are calculated.

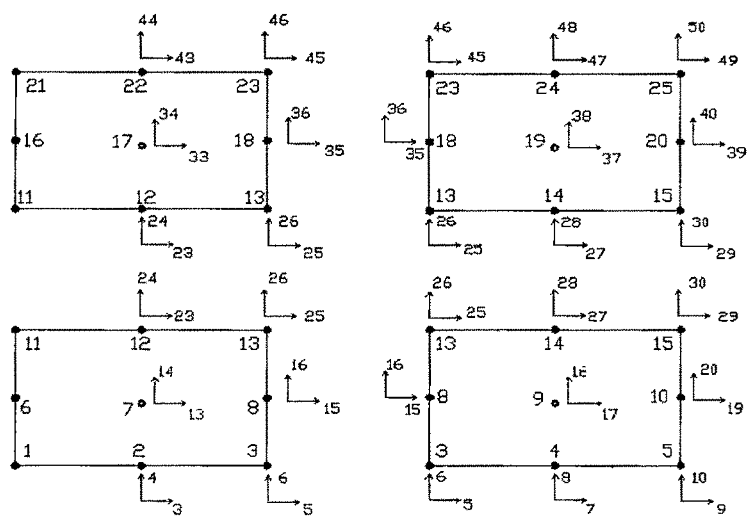


FIG. 6.6 SUBSTRUCTURES WITH KNOWN BOUNDARY DISPLACEMENTS

STEP 5: After calculating all displacements corresponding to internal and boundary nodes and element stresses, all results are assembled in order to have final results. Fig 6.7 shows the whole structure with DOFs corresponding to all nodes.

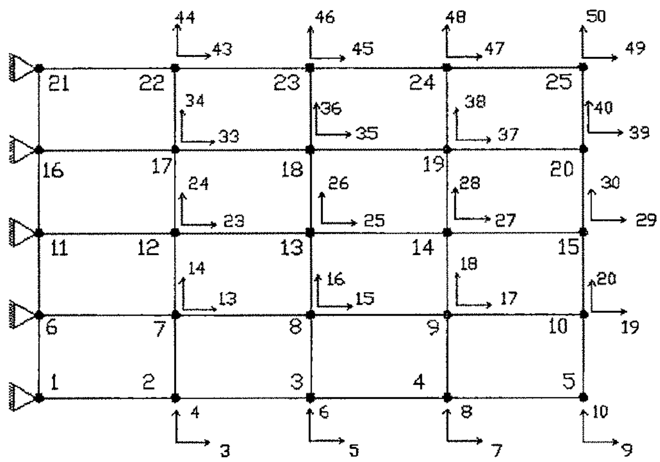


FIG. 6.7 COMPLETE STRUCTURE WITH ALL DISPLACEMENTS

Some of the advantages of the substructure technique can be summarized as follows:

- In substructure technique, the stiffness matrix of each substructure is statically condensed out so that effective stiffness matrix corresponds to only the boundary degrees of freedom.
- The number of nodes in the main structure is reduced because only the boundary nodes of substructure appear there.
- The size of individual substructure is less than that of the main structure being analyzed. Thus at any given instant the main memory required to process the data corresponding to any one substructure is reduced.
- Substructuring technique can be advantageous in case the structure is descritized into identical parts. In such cases, the stiffness matrix of a typical substructure can be formed and condensed only once and can be used as many times as the substructure appears in the main structure.
- Another advantage of substructuring is the reduction in data. But this will be so, only if substructures are identical and repeatedly used.
- The full advantage of substructure technique can be obtained in case of non linear analysis and structural optimization wherein a part of the structure only is modified before the subsequent analysis.

6.5 IMPLEMENTATION OVER DISTRIBUTED COMPUTING ENVIRONMENT

The substructure technique for distributed implementation of finite element method over network of computers using WebDedip is discussed in this section.

In WebDedip for development of distributed application, the solution of a problem is divided in to various tasks, which can run in parallel on different computers and communication between the tasks is carried out through intermediate files using FTP. As WebDedip do not require message-passing functions in program, development of distributed application becomes easy. The finite element method is divided in to number of tasks while using substructure technique as shown in Table 6.1.

TABLE 6.1 TASKS FOR DISTRIBUTED IMPLEMENTATION OF FEM

Sr. No.	Task	Function of task
1.	CSUB1	Read the data of problem. Decompose the finite element mesh in to number of parts known as substructures, equal to number of computers and distribute the data of substructure to different computer.
2.	CSUB2	For each substructure, calculate stiffness matrix and load vector corresponding to boundary or interface nodes using static condensation procedure in parallel on different computers. Subsequently communicate substructure stiffness matrix and load vector for global stiffness matrix and load vector.
3.	CSUB3	Assemble substructure stiffness matrix and load vector of all substructures to form overall stiffness matrix and load vector corresponding to boundary or interface nodes of all substructures. After incorporating boundary conditions of structure, calculate the displacements corresponding to boundary nodes. Communicate the appropriate displacements of boundary nodes corresponding to substructure to different computers.
4.	CSUB4	Compute displacement of internal nodes from that of boundary nodes and secondary unknowns i.e. internal stresses for each substructure in parallel on different computers and communicate the same for overall results.
5.	CSUB5	Combine the results of each substructure i.e. nodal displacements and element stresses of all substructures and prepare the final output.

In first task to form the substructure data, each substructure is assigned an equal number of elements, to have equal computational load among different computers. The process of distributing computational load depending on processing power of computers so that they can complete the allotted task in the same time is known as load balancing. Load balancing carried out in the beginning of process only is known as static load balancing while load adjustment in the beginning and during the process is known as dynamic load balancing. If all computers are of same configuration distribution of equal number of elements to each computer will be a case of static load balancing. The substructure data includes element connectivity, nodal data and boundary nodes. The nodes which are common between more than one substructure is known as boundary nodes or interface nodes. The first task prepares data files for each substructure. These substructures are assigned to different computers, if number of computers are equal to or more than number of substructures. If number of computers are less, more than one substructure may be assigned to some computers, which may violate the load balancing.

In the second task, to calculate substructure stiffness matrix and load vector, internal nodes are numbered first and then external or boundary nodes are numbered. This process of renumbering increases the bandwidth of stiffness matrix and so static condensation requires more computational time. The substructure stiffness matrix and load vector are written on file, which is communicated to master computer for the generation of total stiffness matrix and load vector. To reduce computation time and memory only upper half of banded stiffness matrix is stored. To reduce the size of file to be communicated, only upper half of substructure stiffness matrix is written to file. The intermediate file for second task consists of substructure stiffness matrix, load vector and corresponding boundary degrees of freedom.

The third task assembles substructure stiffness matrix and load vector of all substructures depending on boundary degrees of freedom, in banded form. As the bandwidth and number of unknowns are small, it takes less time to compute displacements of all boundary nodes of entire structure. This task prepares the intermediate file consisting of appropriate boundary displacements for each substructure for communication to different computers, where substructure stiffness matrix and load vector are calculated.

In fourth task the stiffness matrix and load vector are calculated for each substructure. Subsequently substituting the displacements of boundary nodes, displacements of internal nodes are calculated. As in this task, nodes are numbered in sequence and no renumbering is required, the bandwidth remains minimum and it takes less computational time. The intermediate file of this process consists of nodal displacements and element stresses, which are communicated for final results to be prepared by fifth task.

The various tasks carried out by different computers and communication of the data between them is shown in Fig. 6.8.

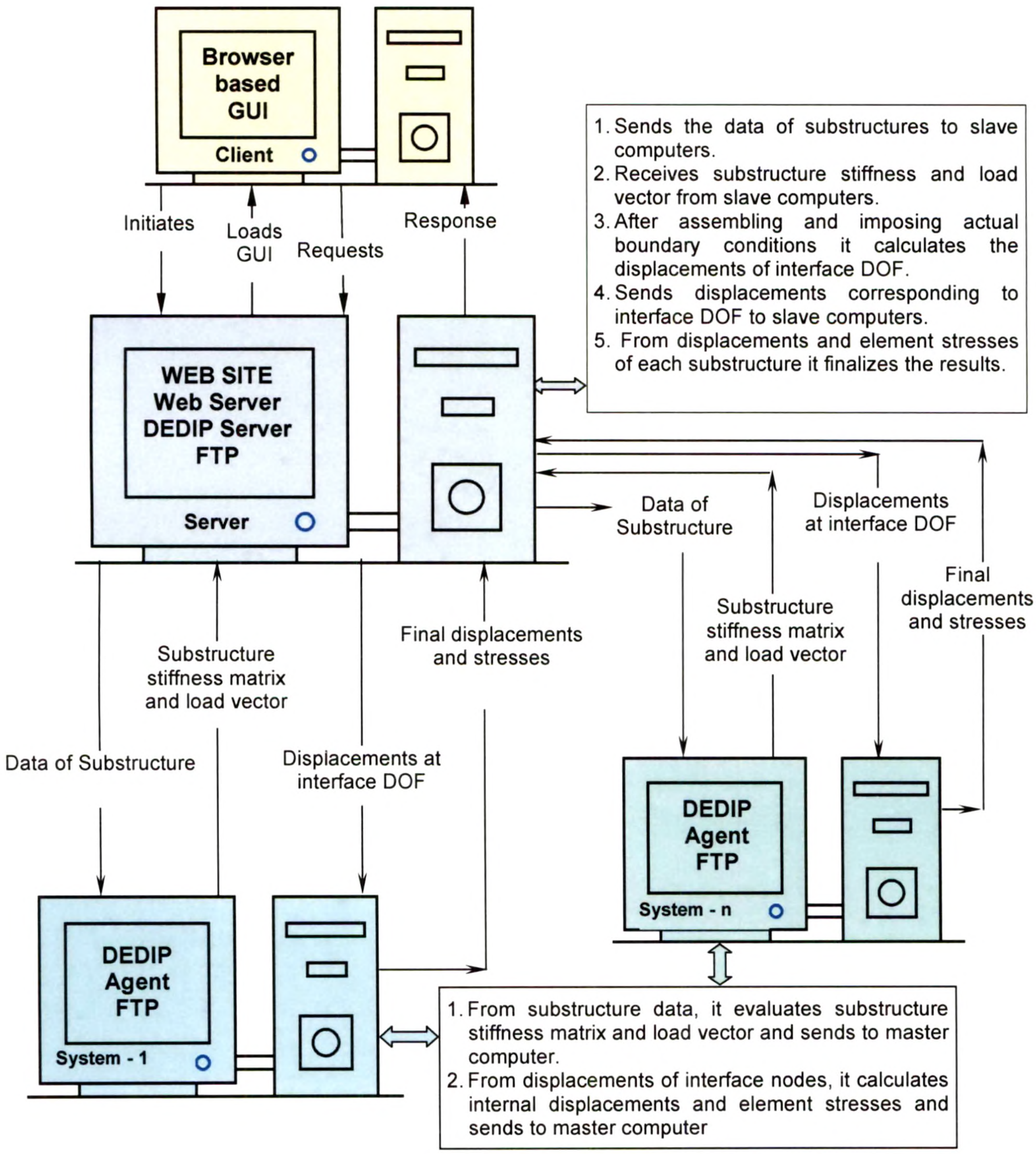


FIG. 6.8 IMPLEMENTATION OF DISTRIBUTED FE ANALYSIS ON WEBDEDIP

The application can be configured using WebDedip environment as explained in Chapter 4. In configuration of application details for process, i.e. node on which it has to run, name of processes on which it is depending and name of dependent processes, and detail of file transfer, i.e. source process, destination process and name of file to be transferred, are given. A screen shot for typical configuration of application over five computers is shown in Fig. 6.9.

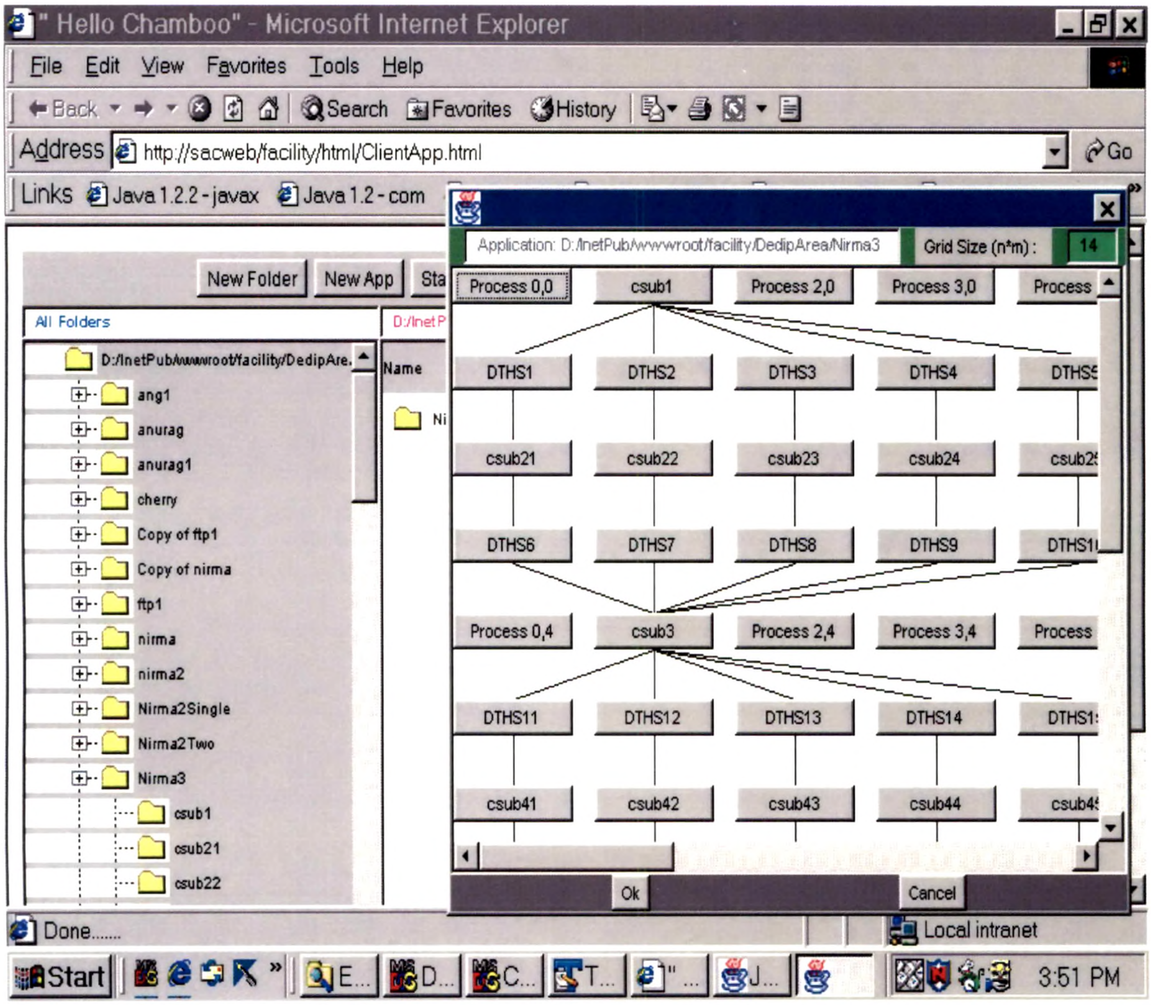
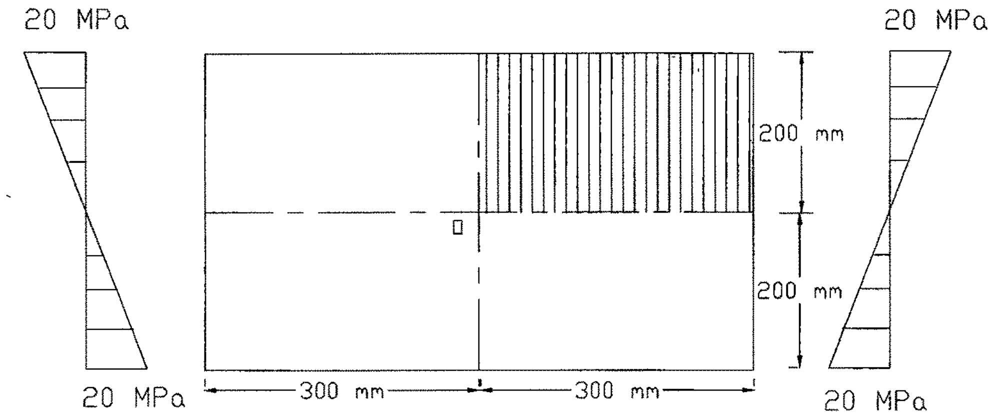


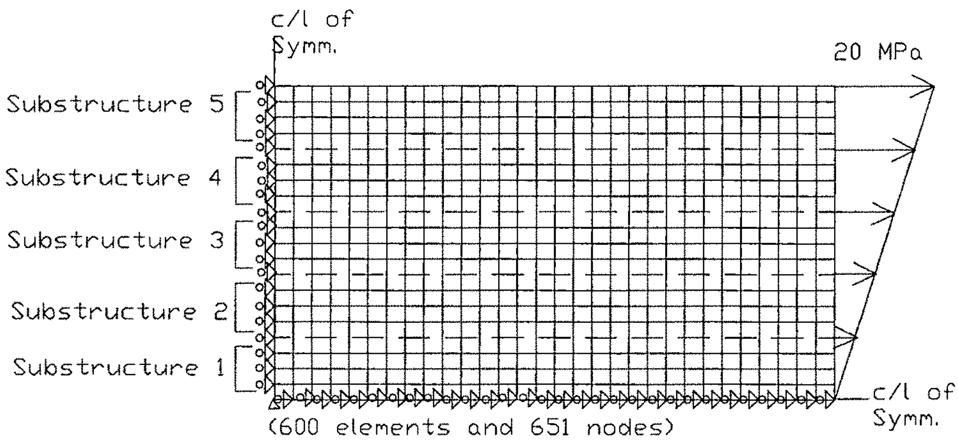
FIG. 6.9 CONFIGURATION OF APPLICATION OVER FIVE COMPUTERS

6.6 BEAM SUBJECTED TO PURE BENDING EXAMPLE

A beam subjected to a bending stress distribution with a maximum intensity of 20 N/mm^2 is shown in Fig. 6.10(a). Young's modulus of elasticity E = 200 kN/mm^2, Poisson's ratio is 0.3 and thickness is 60 mm. Because of symmetry and antisymmetry about the x and y axis respectively, only one quarter of the beam is discretized in to 600 elements (20 rows containing 30 elements) as shown in Fig. 6.10(b). For the discretization plane stress quadrilateral element with linear displacement model is used. Stiffness matrix and load vector of quadrilateral element are obtained by combining properties of four constant strain triangular elements, and finally internal node is condensed out. Further, quarter of the structure is divided into five substructures as shown in Fig. 6.10(b).



(a) Beam under pure bending



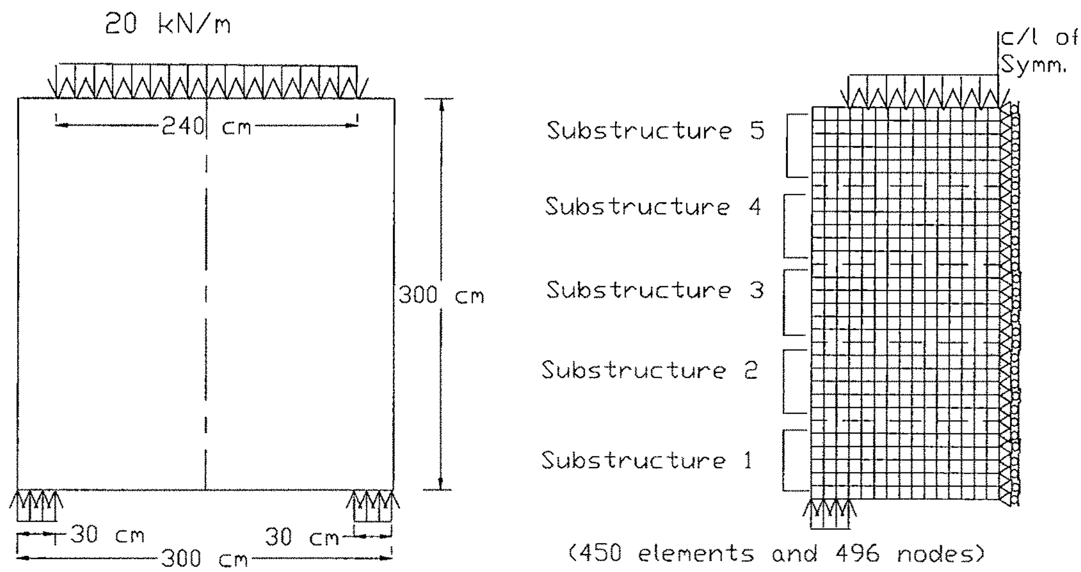
(b) Discretization and substructures

FIG. 6.10 BEAM BENDING EXAMPLE

On the host computer the structure is divided into substructures (csub1.c) and data files corresponding to each substructure are distributed to slave computers. Analysis of each substructure including generation of load vector, and stiffness matrix is carried out on different slave computers (csub2.c). Then substructure matrix and load vector of each substructure are communicated back from slave to host computer. On the host computer substructure stiffness matrix and load vector corresponding to boundary nodes of all sub substructures are assembled to find the displacements at the boundary nodes (csub3.c). These displacements are distributed by host computer to different slave computers as per the boundary nodes of each substructure to find the displacements at the nodes and stresses in each element. After calculating displacements and stresses (csub4.c), slave computers communicate the results of each substructure to host computer. Finally displacements of all nodes and stresses in elements are assembled on host computer (csub5.c) to have final result of analysis. For the problem under consideration the results of analysis are found in good agreement with that of theoretical results given by Desai and Abel [105].

6.7 A DEEP BEAM EXAMPLE

Figure 6.11 (a) shows an example of a deep beam subjected to uniformly distributed load. The Young's modulus of elasticity E = 200 kN/mm^2, Poisson's ratio is 0.2 and thickness is 20 cm. Due to one way symmetry only half of the structure is considered for the analysis. This half portion is discretized in to 450 elements (30 rows of 15 elements each) as shown in Fig. 6.11(b). For discretization plane stress quadrilateral element is considered. Further, for distributed processing it is subdivided in to five substructures and analysed on five different computers. The results of each substructure are combined to get the final result in the form of nodal displacements and element stresses.



(a) Deep beam **(b) Substructures**

FIG. 6.11 DEEP BEAM EXAMPLE

The results of analysis are found to match with the finite element analysis results reported by Krishnamoorthy [102].

6.8 A SQUARE PLATE WITH CIRCULAR HOLE PROBLEM

The geometry of the plate is shown in Fig 6.12. The plate is a square and having a circular hole in the center. This plate is subjected to uniform tension on both edges. Young's modulus of elasticity of the plate material is E = 20000 kN/cm^2 and Poisson's ratio is 0.3. Plate is having thickness of 2.5 cm. As the plate is symmetrical about both x and y-axis, only the quarter plate as shown in Fig 6.12 by hatched area, is descritized into number of elements. For descritization, CST element with linear displacement models is used. Typical discretization of plate with node and element numbering is shown in Fig 6.13.

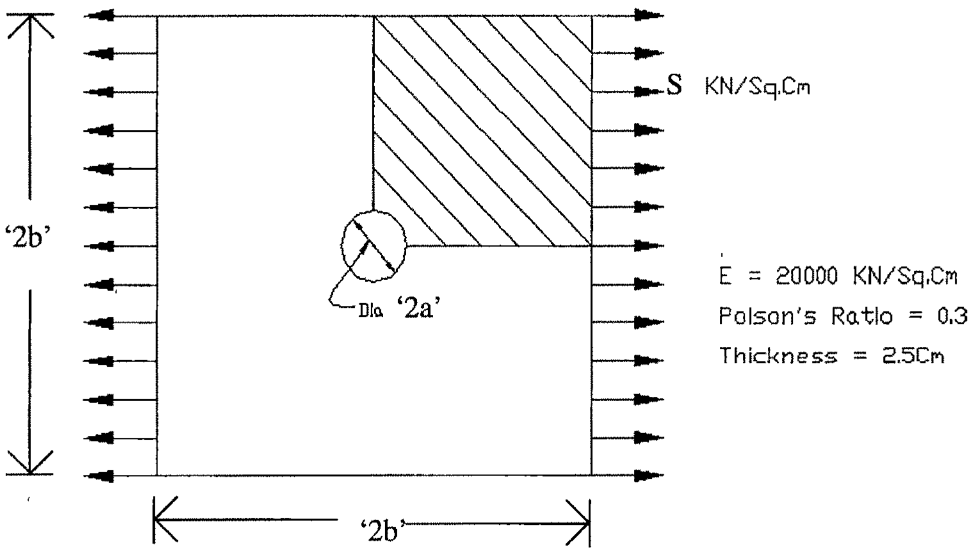


FIG 6.12 RECTANGULAR PLATE WITH CIRCULAR HOLE

Further, this discretization of a quarter plate is divided into number of substructures, having almost equal number of elements to implement over distributed computing environment. Different number of substructures is considered in the study. Fig 6.14 show division of plate into five substructures.

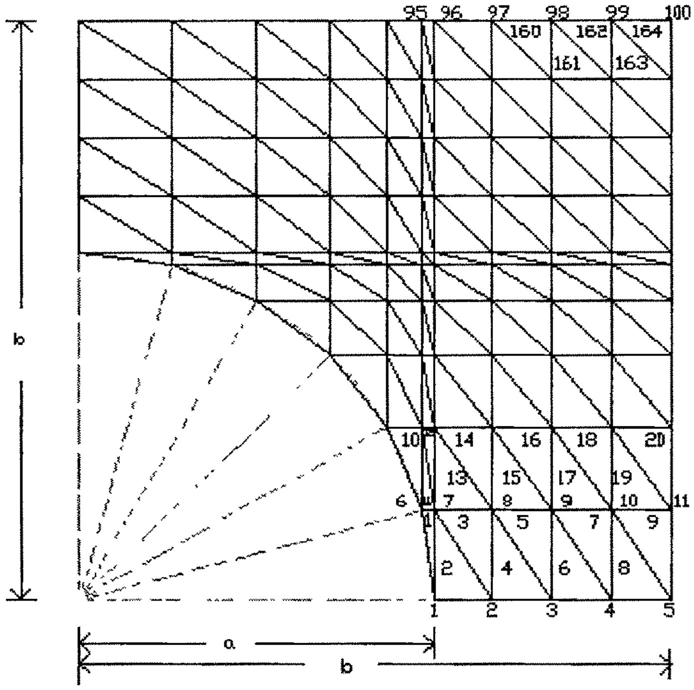


FIG. 6.13 DISCRETIZATION OF QUARTER PLATE USING CST ELEMENTS

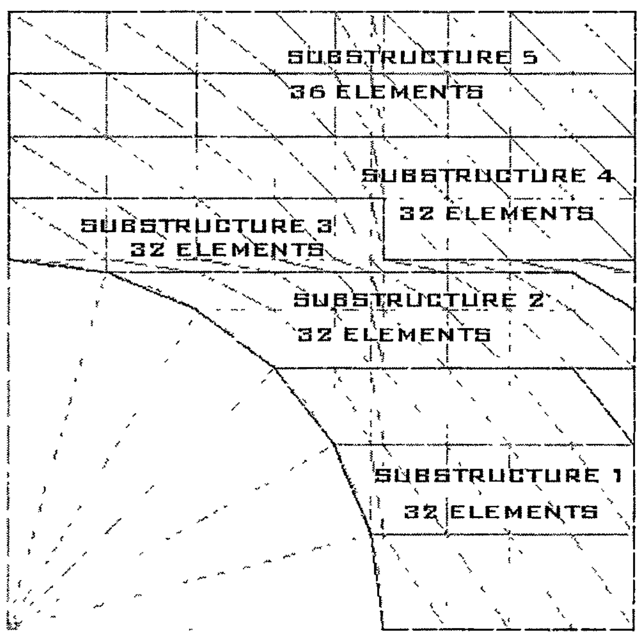


FIG. 6.14 DIVISION OF QUARTER PLATE INTO FIVE SUBSTRUCTURES

Here the quarter plate is discretized into 27504 elements. So as to have 13990 nodes. Each node is having 2 DOF so total number of unknowns are 27980. The screen shot for configuration of the application, when distributed over three computers is reproduced here in Fig 6.15.

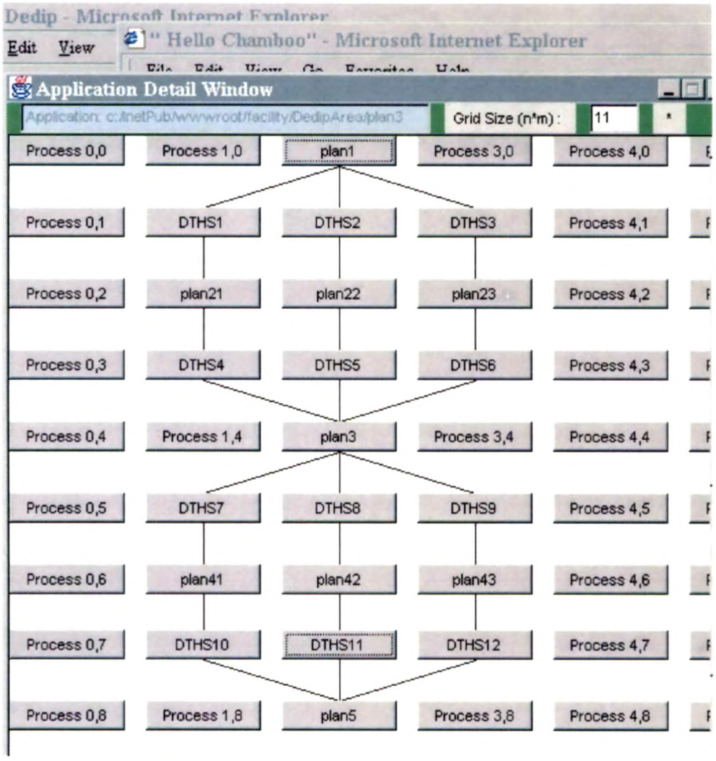


FIG. 6.15 CONFIGURATION FOR APPLICATION OVER THREE COMPUTERS

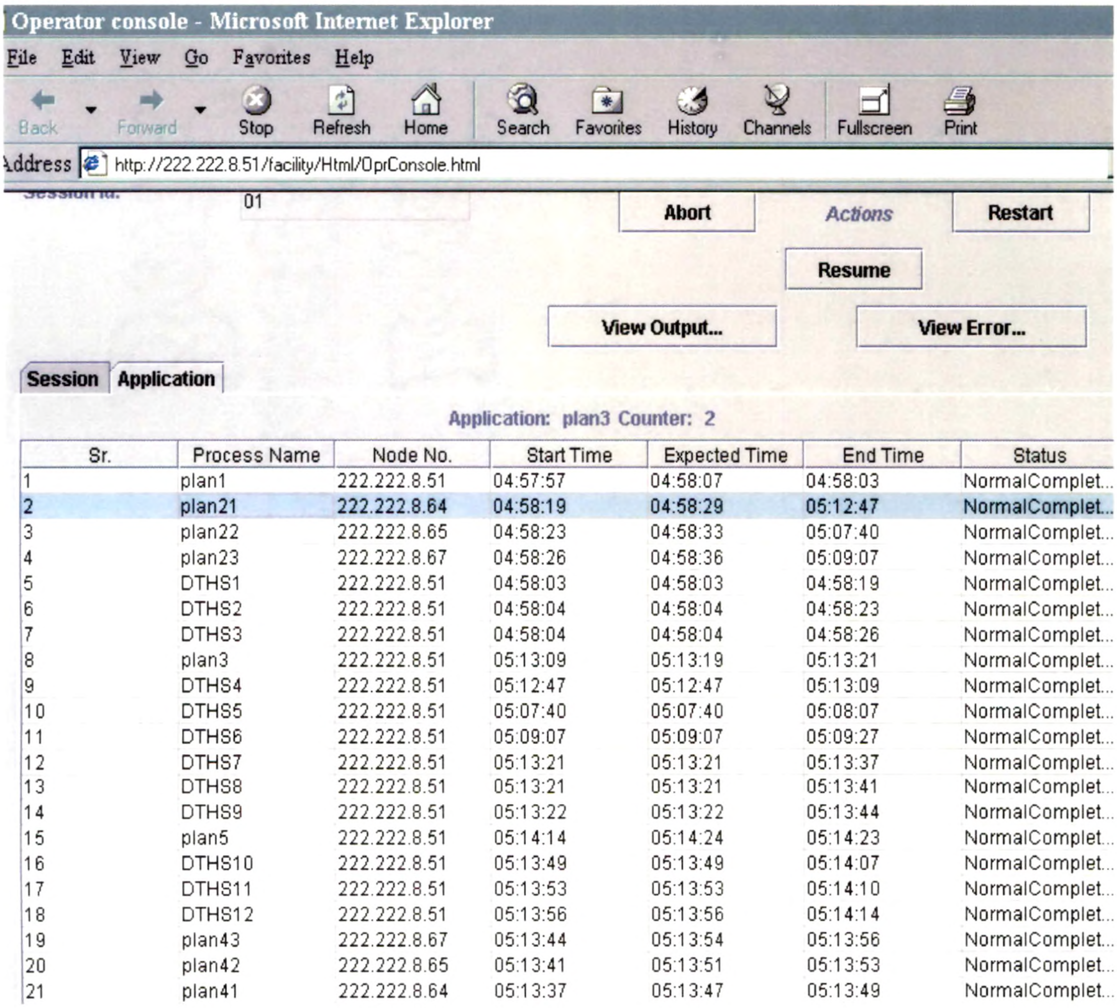
For the problem under consideration good agreement has been observed with the analytical solution for σ_θ given by Timoshenko [115]. Comparison of results of FEM analysis with the classical theory results is given in Table 6.2.

TABLE 6.2 COMPARISON OF ANALYSIS AND REFERENCE RESULTS

Sr. No	a (mm)	b (mm)	R (mm)	S (N/mm²)	σ_x (N/mm²)	σ_y (N/mm²)	τ_{xy} (N/mm²)	σ_θ (N/mm²)	
								Timoshenko	FEM
1	100	500	110	100	221.8	23.44	-40.25	234.58	229.58
2			150		152.9	32.32	4.34	147.05	147.78
3			200		121.9	20.42	5.28	118.29	117.00
4			250		111.7	14.49	4.78	108.59	107.13
5			300		107.7	10.88	3.94	104.28	103.39
6			375		104.3	7.07	2.88	101.25	100.33
7			450		102.4	4.94	2.19	99.80	98.69

In Table 6.2, R is the radial distance of point of interest and S is the intensity of tensile force.

After completion of application, WebDedip gives the summary of application including node number on which application run successfully, start time and end time. If error occurs at any computer either due to communication or due to failure of program, can be known through the summary given by Operator Console. A typical screen shot of summary of an application is depicted in Fig. 6.16.



Sr.	Process Name	Node No.	Start Time	Expected Time	End Time	Status
1	plan1	222.222.8.51	04:57:57	04:58:07	04:58:03	NormalComplet...
2	plan21	222.222.8.64	04:58:19	04:58:29	05:12:47	NormalComplet...
3	plan22	222.222.8.65	04:58:23	04:58:33	05:07:40	NormalComplet...
4	plan23	222.222.8.67	04:58:26	04:58:36	05:09:07	NormalComplet...
5	DTHS1	222.222.8.51	04:58:03	04:58:03	04:58:19	NormalComplet...
6	DTHS2	222.222.8.51	04:58:04	04:58:04	04:58:23	NormalComplet...
7	DTHS3	222.222.8.51	04:58:04	04:58:04	04:58:26	NormalComplet...
8	plan3	222.222.8.51	05:13:09	05:13:19	05:13:21	NormalComplet...
9	DTHS4	222.222.8.51	05:12:47	05:12:47	05:13:09	NormalComplet...
10	DTHS5	222.222.8.51	05:07:40	05:07:40	05:08:07	NormalComplet...
11	DTHS6	222.222.8.51	05:09:07	05:09:07	05:09:27	NormalComplet...
12	DTHS7	222.222.8.51	05:13:21	05:13:21	05:13:37	NormalComplet...
13	DTHS8	222.222.8.51	05:13:21	05:13:21	05:13:41	NormalComplet...
14	DTHS9	222.222.8.51	05:13:22	05:13:22	05:13:44	NormalComplet...
15	plan5	222.222.8.51	05:14:14	05:14:24	05:14:23	NormalComplet...
16	DTHS10	222.222.8.51	05:13:49	05:13:49	05:14:07	NormalComplet...
17	DTHS11	222.222.8.51	05:13:53	05:13:53	05:14:10	NormalComplet...
18	DTHS12	222.222.8.51	05:13:56	05:13:56	05:14:14	NormalComplet...
19	plan43	222.222.8.67	05:13:44	05:13:54	05:13:56	NormalComplet...
20	plan42	222.222.8.65	05:13:41	05:13:51	05:13:53	NormalComplet...
21	plan41	222.222.8.64	05:13:37	05:13:47	05:13:49	NormalComplet...

FIG. 6.16 SUMMARY OF APPLICATION GIVEN BY WEBDEDIP

Average time required for various processes, when entire structure is divided into different number of substructures, is tabulated in Table 6.3. Also the time required in parallel implementation i.e. computation time and communication time are shown separately. Based on the time required for various processes in sequential and parallel implementation, speedup is calculated. The comparison of ideal speedup and observed speedup is shown in Fig 6.17. Comparison of communication and computation time for various substructures is also shown in

Fig.6.18. From calculated speedup and ideal speedup, efficiency is also calculated as a measure of performance. The timing shown in Table 6.3 are observed when application is implemented over network of Pentium-IV computers having 256 MB RAM running at 1.8 GHz. Computers are connected through 100 Mbps Ethernet line and are a part of local area network.

TABLE 6.3 TIME REQUIRED FOR SEQUENTIAL AND PARALLEL PROCESSING

Process	NB	NEQ	time (sec)	Sequential time (sec)	Parallel time		Speed up		Efficiency (%)
					Comp (sec)	comm (sec)	Ideal	Observed	
3 Substructures									
plan1	-	-	6						
plan2	9080	9476	689						
plan3	632	1380	12						
plan4	240	9476	12						
plan5	-	-	6	2127	725	80	3	2.64	88.07
4 Substructures									
plan1	-	-	6						
plan2	6808	7162	268						
plan3	590	1612	7						
plan4	240	7162	9						
plan5	-	-	7	1128	297	87	4	2.94	73.44
5 Substructures									
plan1	-	-	6						
plan2	5442	5774	132						
plan3	572	1852	8						
plan4	240	5774	7						
plan5	-	-	6	854	159	92	5	3.40	68.05
6 Substructures									
plan1	-	-	7						
plan2	4534	4848	68						
plan3	554	2082	8						
plan4	240	4848	6						
plan5	-	-	8	467	97	100	6	2.37	39.51

(NB Band width, NEQ Number of equations)

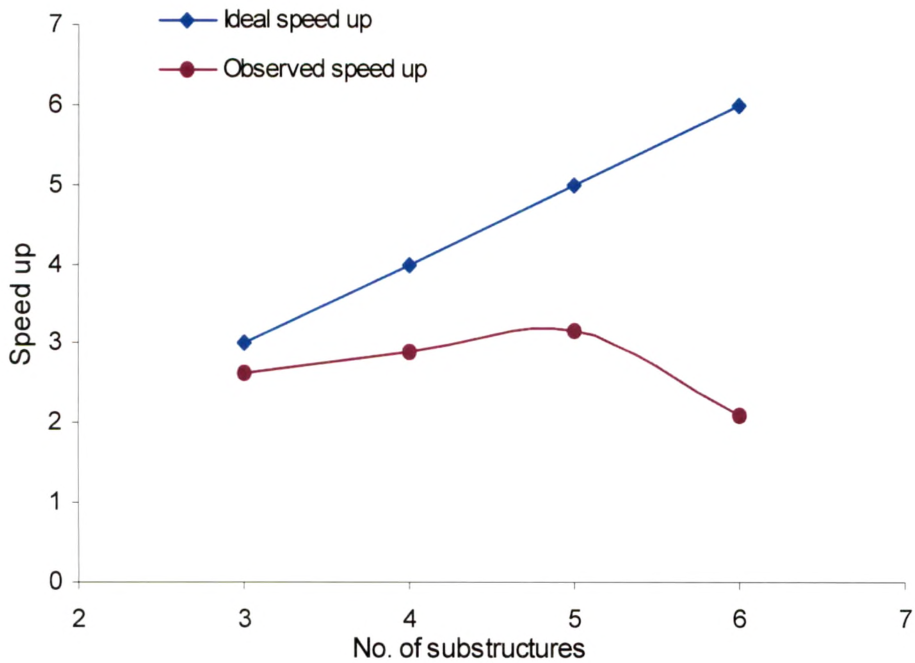


FIG. 6.17 COMPARISON OF IDEAL AND OBSERVED SPEEDUP

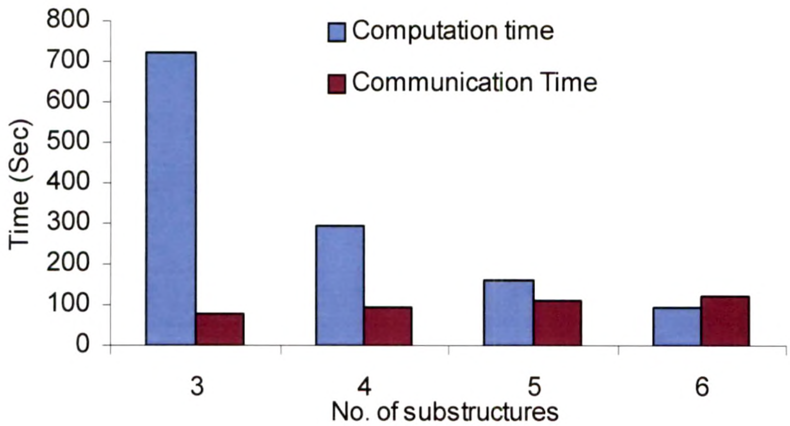


FIG. 6.18 COMPUTATION AND COMMUNICATION TIME

The size of problem considered here is large so as to clearly observe the advantage of parallel processing. As the number of substructures or computers increases, the computation time reduces and communication time increases. From Table 6.3, it is observed that process 'Plan2', for calculation of substructure stiffness matrix and load vector consumes maximum time due to larger bandwidth of the stiffness matrix. But as number of computers increases, time required for second process reduces which lowers the overall computation time. As the number of computer increases, the total time required for solution reduces but the speedup i.e. ratio of sequential time to parallel time increases up to five computers but for six computers it reduces. This reduction in speedup is due to increased communication time. Thus, the optimum number of computers for maximum speedup depends on size of problem so that computational time is greater than that of communication time.

6.9 TRANSVERSELY LOADED ANNULAR PLATE PROBLEM

For plate bending problem, a circular plate having hole in the center, as shown in Fig 6.19, is selected. Such analysis is often useful in case of annular raft or when any circular slab is having hole at the center and subjected to uniformly distributed load. Plate is fixed at the outer edge and free at the inner edge and is subjected to uniformly distributed pressure. Modulus of elasticity of plate is 2×10^8 kN/m^2, Poisson's ratio is 0.3, thickness of the plate is 0.1 m and the intensity of pressure is 20 kN/m^2.

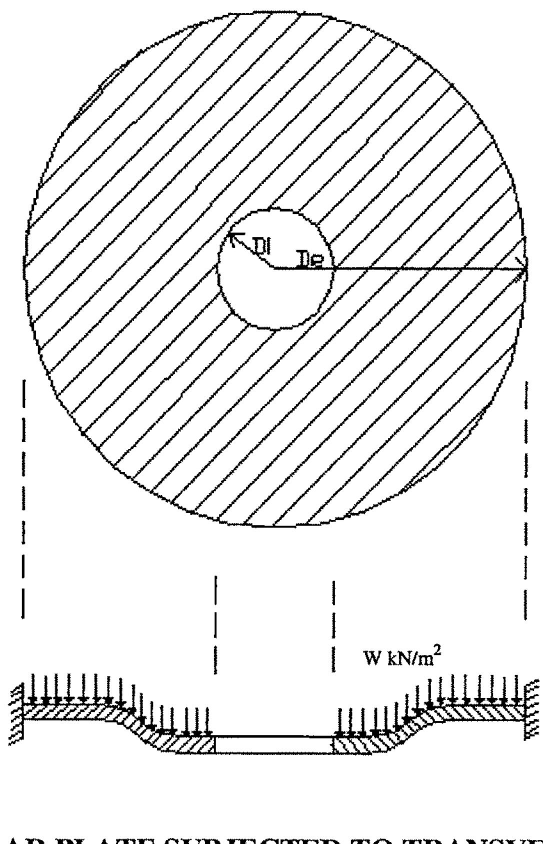


FIG. 6.19 ANNULAR PLATE SUBJECTED TO TRANSVERSE LOADING

For the analysis 8 noded isoparametric plate element is used. Due to symmetry only quarter plate is discretized in to eight noded isoparametric elements. Typical discretizations is shown in Fig 6.20.

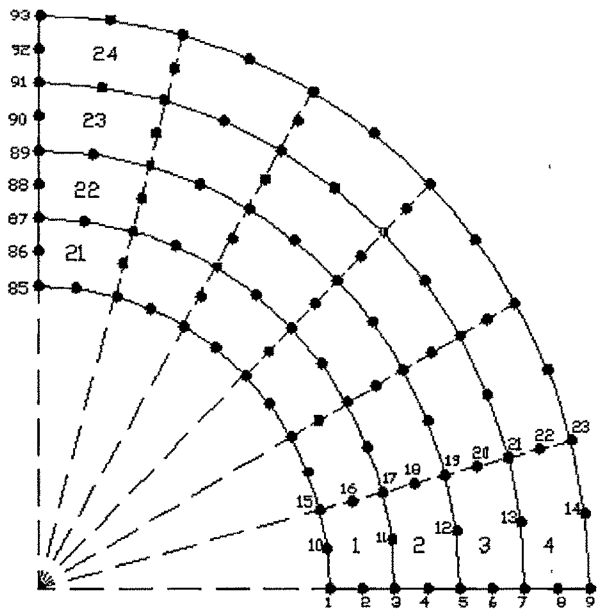


FIG. 6.20 DISCRITEZATION OF QUARTER PLATE

Further the plate is divided into number of substructures. Load balance for each computer is maintained by keeping number of elements same in each substructure. Fig 6.21 shows the typical division of quarter plate into three substructures.

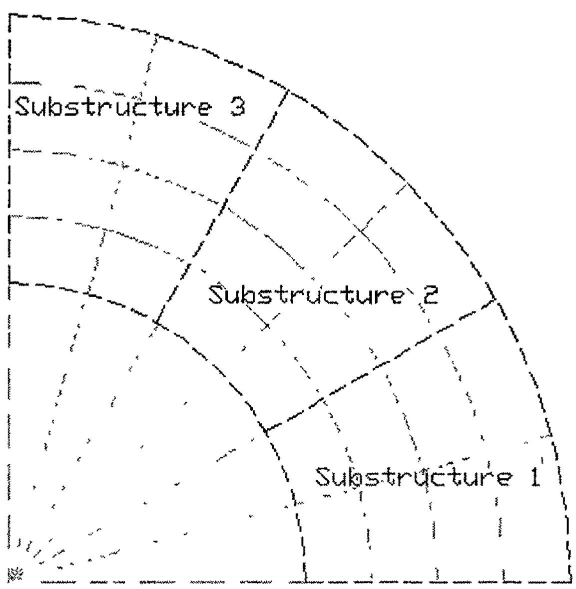


FIG. 6.21 DIVISION OF QUARTER PLATE INTO 3 SUBSTRUCTURES

Results for the deflections are tabulated in Table 6.4. These results are compared with those obtained by analytical methods, and are reported by Timoshenko [116].

TABLE 6.4 COMPARISON OF DISPLACEMENT COEFFICIENT

Sr No	ID = b	ED = a	TNE	TNN	k1	
					FE Analysis	Reference [116]
1	2	4	4	21	0.057	0.057
2	2	6	4	21	0.129	0.130
3	2	8	4	21	0.161	0.162
4	2	10	4	21	0.174	0.175
5	4	8	12	51	0.057	0.057
6	5	15	12	51	0.129	0.130
7	4	16	12	51	0.161	0.162
8	4	20	12	51	0.175	0.175
9	2	4	24	93	0.057	0.057
10	2	6	24	93	0.129	0.130
11	2	8	24	93	0.162	0.162
12	2	10	24	93	0.174	0.175
13	2	4	48	173	0.057	0.057
14	2	6	48	173	0.129	0.130
15	2	8	48	173	0.162	0.162
16	2	10	48	173	0.175	0.175
17	2	4	72	253	0.057	0.057
18	2	6	72	253	0.129	0.130
19	2	8	72	253	0.161	0.162
20	2	10	72	253	0.173	0.175
21	2	4	96	333	0.057	0.057
22	2	6	96	333	0.129	0.130
23	2	8	96	333	0.162	0.162
24	2	10	96	333	0.174	0.175

In Table 6.4 TNE = Total number of elements, TNN = Total number of nodes and coefficient $k_1 = W_{max}\ E\ h^3 / q\ a^4$, where W_{max} is maximum deflection, 'q' is intensity of uniformly distributed load, 'E' is Young's modulus of elasticity, 'h' is thickness of plate and 'a' is external diameter of annular plate.

For distributed implementation this plate bending problem is discretized into 1800 elements and 5725 nodes. Each node is having three DOF resulting into total 17175 unknowns. In this application the finite element analysis of plate using substructure technique is divided in to following five tasks. The first task "**plana1**" prepares separate data file for individual substructure. This data file includes the details such as number of elements, number of nodes, material property, loading data, and the boundary nodes for that particular substructure. Second task "**plana2**" calculates stiffness matrix and load vector corresponding to boundary nodes for a substructure using static condensation. Third task "**plana3**" assembles stiffness matrix and load vector for all substructures and calculates displacements of boundary nodes. In fourth task "**plana4**" internal displacements are calculated using boundary displacements. After calculating all degrees of freedom, element stresses are calculated. Finally, the fifth task "**plana5**" combines results of all substructures to have final result. The configuration of the problem on WebDedip, when divided into four substructures (assigned each substructure on separate computer) is shown in Fig 6.22.

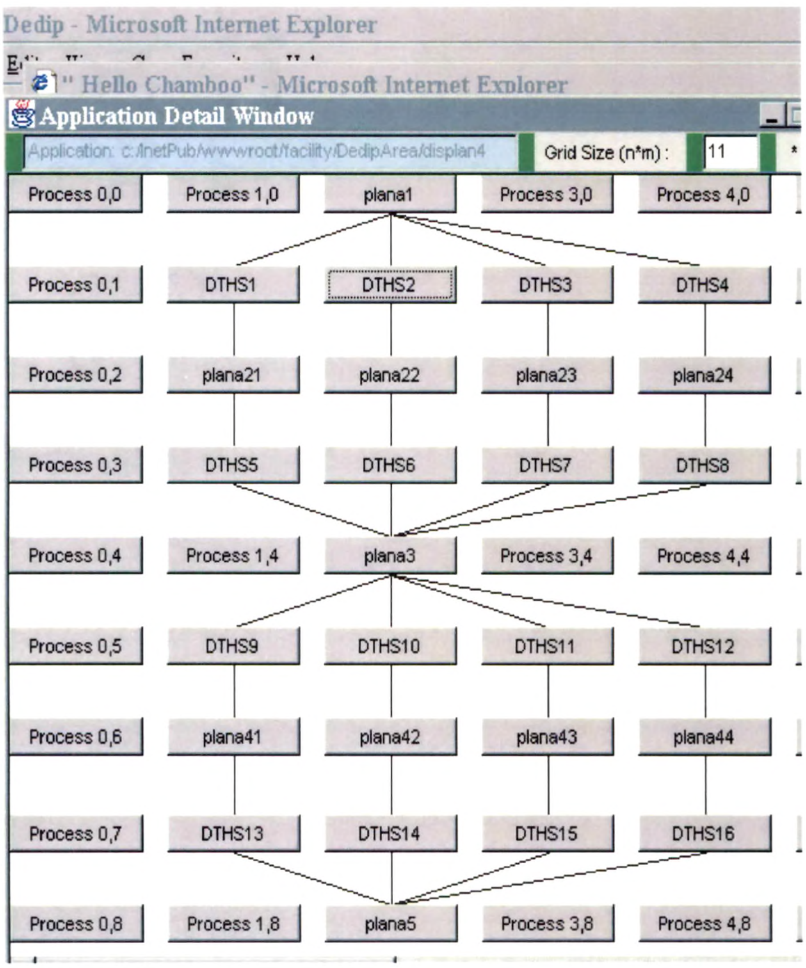


FIG. 6.22 CONFIGURATION OF PLATE BENDING PROBLEM

From the summary of the application as shown in Fig. 6.23 time required for various tasks i.e. computational time and communication time are obtained.

View Output... View Error...

Session | Application

Application: displan4 Counter: 2

Sr.	Process Name	Node No.	Start Time	Expected Time	End Time	Status
1	plana1	222.222.8.51	21:48:01	21:48:11	21:48:08	NormalComplet.
2	plana21	222.222.8.65	21:48:25	21:48:35	21:53:09	NormalComplet.
3	plana22	222.222.8.67	21:48:27	21:48:37	21:53:12	NormalComplet.
4	plana23	222.222.8.68	21:48:31	21:48:41	21:53:11	NormalComplet.
5	plana24	222.222.8.69	21:48:34	21:48:44	21:53:17	NormalComplet.
6	DTHS1	222.222.8.51	21:48:08	21:48:08	21:48:24	NormalComplet.
7	DTHS2	222.222.8.51	21:48:08	21:48:08	21:48:27	NormalComplet.
8	DTHS3	222.222.8.51	21:48:08	21:48:08	21:48:31	NormalComplet.
9	DTHS4	222.222.8.51	21:48:08	21:48:08	21:48:34	NormalComplet.
10	DTHS5	222.222.8.51	21:53:09	21:53:09	21:53:38	NormalComplet.
11	DTHS6	222.222.8.51	21:53:12	21:53:12	21:54:09	NormalComplet.
12	DTHS7	222.222.8.51	21:53:11	21:53:11	21:54:01	NormalComplet.
13	DTHS8	222.222.8.51	21:53:18	21:53:18	21:54:09	NormalComplet.
14	plana3	222.222.8.51	21:54:09	21:54:19	21:56:55	NormalComplet.
15	plana41	222.222.8.65	21:57:12	21:57:22	21:58:18	NormalComplet.
16	DTHS9	222.222.8.51	21:56:55	21:56:55	21:57:11	NormalComplet.
17	DTHS10	222.222.8.51	21:56:55	21:56:55	21:57:14	NormalComplet.
18	DTHS11	222.222.8.51	21:56:55	21:56:55	21:57:18	NormalComplet.
19	DTHS12	222.222.8.51	21:56:55	21:56:55	21:57:21	NormalComplet.
20	plana5	222.222.8.51	21:58:48	21:58:58	21:58:55	NormalComplet.
21	plana42	222.222.8.67	21:57:14	21:57:24	21:58:21	NormalComplet.
22	plana43	222.222.8.68	21:57:18	21:57:28	21:58:24	NormalComplet.
23	plana44	222.222.8.69	21:57:21	21:57:31	21:58:28	NormalComplet.
24	DTHS13	222.222.8.51	21:58:18	21:58:18	21:58:34	NormalComplet.
25	DTHS14	222.222.8.51	21:58:21	21:58:21	21:58:39	NormalComplet.
26	DTHS15	222.222.8.51	21:58:24	21:58:24	21:58:44	NormalComplet.
27	DTHS16	222.222.8.51	21:58:28	21:58:28	21:58:48	NormalComplet.

FIG. 6.23 TIME REQUIREMENT FOR INDIVIDUAL PROCESS

The average time required for various processes, when entire structure is divided into different number of substructure, is included in Table 6.5. Also time required in parallel implementation i.e. computation time and communication time is shown. Based on the time required for various tasks / processes in sequential and parallel implementation, speedup is calculated. The comparison of ideal speedup and observed speedup is depicted in Fig 6.24. From calculated speedup and ideal speedup, efficiency is also calculated as a measure of performance. Fig 6.25 shows the comparison of computation and communication time.

TABLE 6.5 TIME REQUIRED FOR SEQUENTIAL & PARALLEL PROCESSING

Processes	NB	NEQ	Time (Sec)	Sequential Time (Sec)	Parallel time		Speed up		Efficiency (%)
					Comp (Sec)	Comm. (Sec)	Ideal	Calculated	
3 substructures									
plana1	-	-	7						
plana2	5391	6327	556						
plana3	1848	3738	122						
plana4	1365	6327	88						
plana5	-	-	7	2068	780	100	3	2.35	78.33
4 substructures									
plana1	-	-	7						
plana2	4047	4971	358						
plana3	1836	4635	165						
plana4	1365	4971	67						
plana5	-	-	7	1879	604	119	4	2.60	64.97
6 substructures									
plana1	-	-	7						
plana2	2703	3615	115						
plana3	1824	6429	248						
plana4	1365	3615	44						
plana5	-	-	7	1216	421	135	6	2.18	36.45

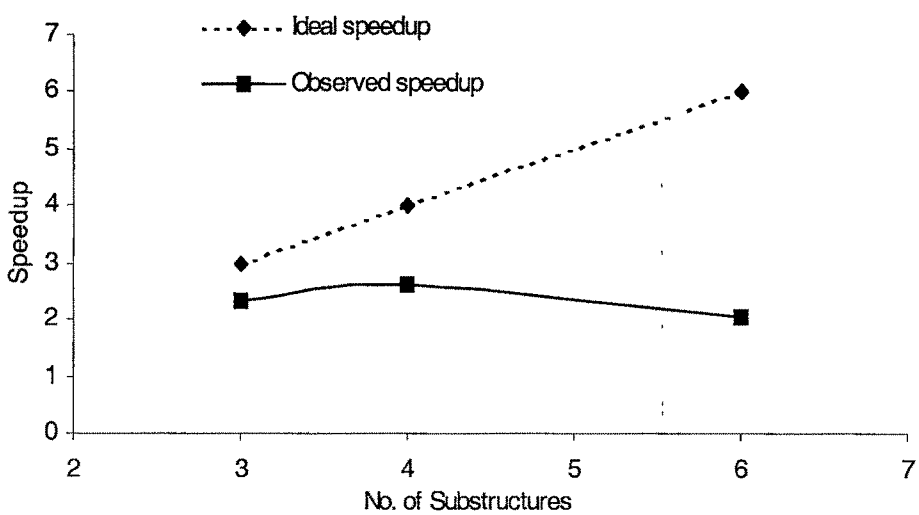


FIG. 6.24 COMPARISON OF SPEEDUP

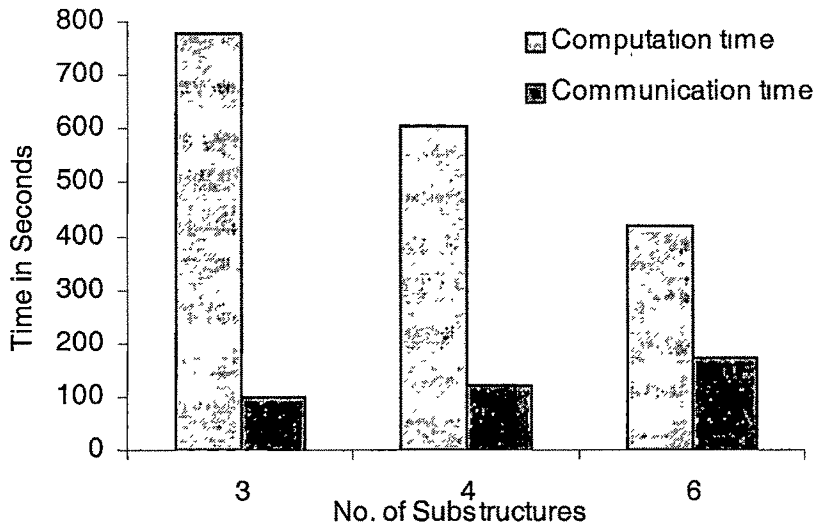


FIG. 6.25 COMPUTATION AND COMMUNICATION TIME

6.10 TRANSVERSELY LOADED SKEW PLATE EXAMPLE

A skew plate (Fig 6.26) subjected to uniformly distributed transverse pressure is simply supported at two edges and free at the remaining two edges. Modulus of elasticity of plate is 2×10^8 kN/m^2, Poisson's ratio is 0.2, thickness of the plate is 0.1 m and the intensity of pressure is 20 kN/m^2. For the analysis eight noded isoparametric plate element is selected. Typical discretization of the plate is shown in Fig 6.26 whereas typical division of plate into four substructures in shov

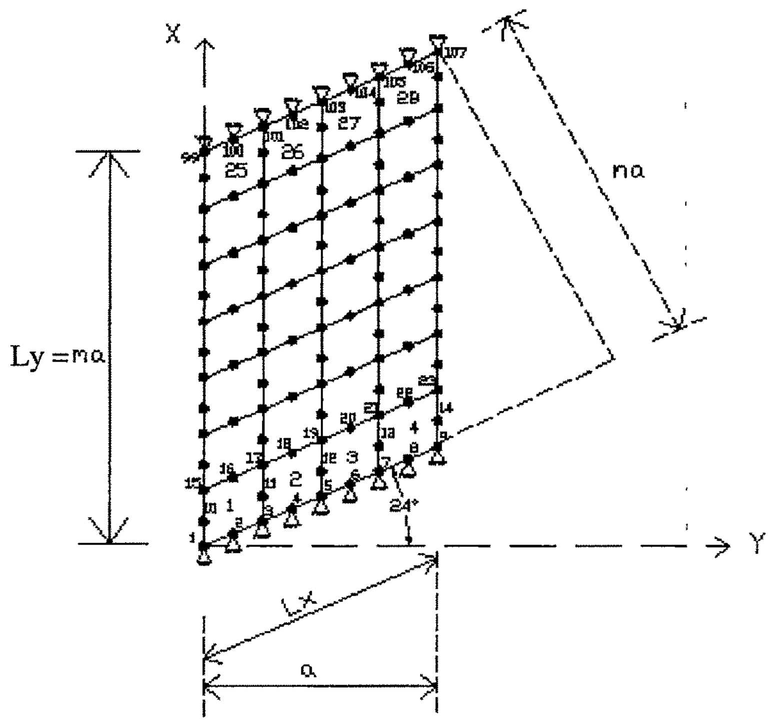


FIG. 6.26 SKEW PLATE PROBLEM AND ITS DISCRETIZATION

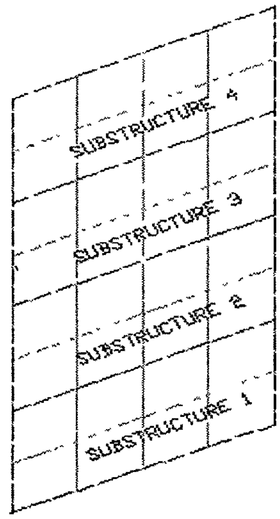


FIG. 6.27 DIVISION OF PLATE INTO 4 SUBSTRUCTURES

Results for the deflections and moments for various skew angles are compared with those obtained by analytical methods and represented by Timoshenko [116]. Variation of maximum deflection for various skew angles and for various discretization of plate is tabulated below in Table 6.6.

TABLE 6.6 DEFLECTION COEFFICIENT FOR SKEW PLATE

Sr No	Angle	Lx	Ly	Nex	Ney	Tne	α_0	
							Present	Timoshenko
1	0	5	10.00	6	12	72	0.212	0.214
2	0	5	10.00	10	20	200	0.211	0.214
3	0	10	20.00	30	40	1200	0.172	0.214
4	0	10	20.00	26	52	1352	0.198	0.214
5	30	5	8.31	6	12	72	0.115	0.118
6	30	5	8.31	10	20	200	0.114	0.118
7	30	10	16.63	30	40	1200	0.103	0.118
8	30	10	16.63	26	52	1352	0.091	0.118
9	45	5	7.07	6	12	72	0.070	0.070
10	45	5	7.07	10	20	200	0.070	0.070
11	45	10	14.14	30	40	1200	0.071	0.070
12	45	10	14.14	26	52	1352	0.068	0.070
13	60	5	5.00	6	12	72	0.018	0.018
14	60	5	5.00	10	20	200	0.018	0.018
15	60	10	10.00	30	40	1200	0.018	0.018
16	60	10	10.00	26	52	1352	0.018	0.018

Where Lx / Ly is Length of plate in x and y-direction respectively, Nex / Ney is number of elements along x and y-direction respectively, TNE is Total number of elements, α_0 = WD/qa^4, W is deflection at the center of plate, q is intensity of uniformly distributed load and D is flexural rigidity of plate = $E h^3 / 12(1-v^2)$.

Variation for maximum moment (My) at the center of plate for various skew angles and for various descritization of plate is tabulated below in Table 6.7

TABLE 6.7 COEFFICIENTS FOR MAXIMUM MOMENT AT CENTER OF PLATE

Sr No	Lx	Ly	Nex	Ney	TNE	Bo	
						Present	Timoshenko
1	5	10	5	11	55	0.497	0.495
2	5	10	11	21	231	0.499	0.495
3	5	10	15	25	375	0.500	0.495
4	5	10	25	51	1275	0.500	0.495
5	5	10	5	11	55	0.364	0.368
6	5	10	11	21	231	0.365	0.368
7	5	10	15	25	375	0.365	0.368
8	5	10	25	51	1275	0.363	0.368
9	5	10	5	11	55	0.275	0.291
10	5	10	11	21	231	0.278	0.291
11	5	10	15	25	375	0.275	0.291
12	5	10	25	51	1275	0.283	0.291

In Table 6.7, $B_0 = M / q a^4$ where M is maximum moment at the center of plate

Variation for maximum moment (My) at the center of unsupported edge of plate for various skew angles and for various descritization is presented in Table 6.8.

TABLE 6.8 COEFFICIENT FOR MAXIMUM MOMENT AT FREE EDGE

Sr No	Lx	Ly	Nex	Ney	TNE	B1	
						FE Analysis	Reference
1	5	10	5	11	55	0.509	0.508
2	5	10	11	21	231	0.511	0.508
3	5	10	15	25	375	0.511	0.508
4	5	10	25	51	1275	0.512	0.508
5	5	10	5	11	55	0.376	0.367
6	5	10	11	21	231	0.373	0.367
7	5	10	15	25	375	0.371	0.367
8	5	10	25	51	1275	0.367	0.367
9	5	10	5	11	55	0.298	0.296
10	5	10	11	21	231	0.296	0.296
11	5	10	15	25	375	0.293	0.296
12	5	10	25	51	1275	0.294	0.296

In Table 6.8, B_1 = M1 / q a^4 where M1 = maximum moment at the center of free edge of plate

For distributed implementation the plate is discretized into 3000 elements and 9221 nodes. Each node is having three DOF resulting in total 27663 unknowns. Average time required for various processes, when entire structure is divided into different number of substructure, is reported in Table 6.9.

TABLE 6.9 TIME FOR SEQUENTIAL AND PARALLEL PROCESSING

Process	NB	NEQ	Time (sec)	Sequential time (sec)	Parallel		speed up		Efficiency (%)
					Comp (sec)	Comm (sec)	Ideal	Calculated	
3 Substructure									
plana1	-	-	7	2545	951	110	3	2.40	79.96
plana2	8895	9423	707						
plana3	840	1914	140						
plana4	465	9423	90						
plana5	-	-	7						
4 substructure									
plana1	-	-	7	2499	747	127	4	2.86	71.48
plana2	6675	7143	502						
plana3	780	2211	149						
plana4	465	7143	82						
plana5	-	-	7						
5 substructure									
plana1	-	-	7	2466	566	143	5	3.48	69.56
plana2	5343	5775	315						
plana3	744	2508	172						
plana4	465	5775	65						
plana5	-	-	7						
6 substructure									
plana1	-	-	7	1648	433	161	6	2.77	46.24
plana2	4455	4863	180						
plana3	720	2805	175						
plana4	465	4863	63						
plana5	-	-	8						

Also time required in parallel implementation i.e. computation time and communication time is indicated. A comparison of ideal speedup and observed speedup is shown in Fig 6.28. Whereas comparison of communication and computation time for number of substructures is depicted in Fig 6.29. From calculated speedup and ideal speedup, efficiency is also calculated as a measure of performance.

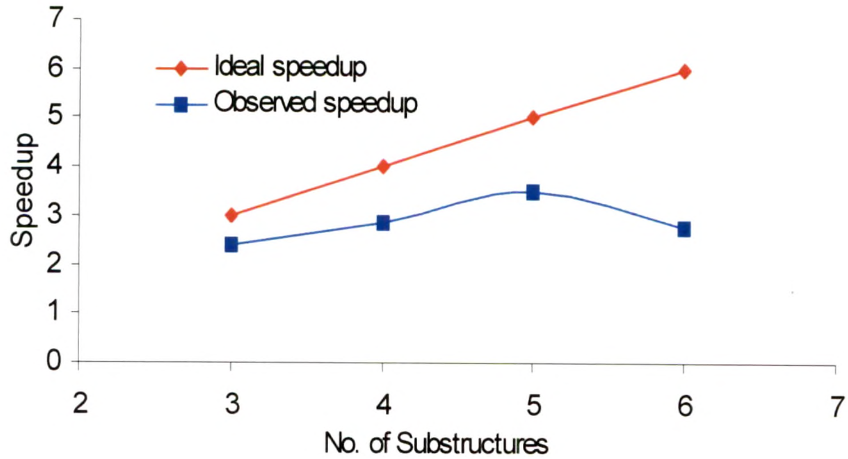


FIG. 6.28 COMPARISON OF SPEEDUP

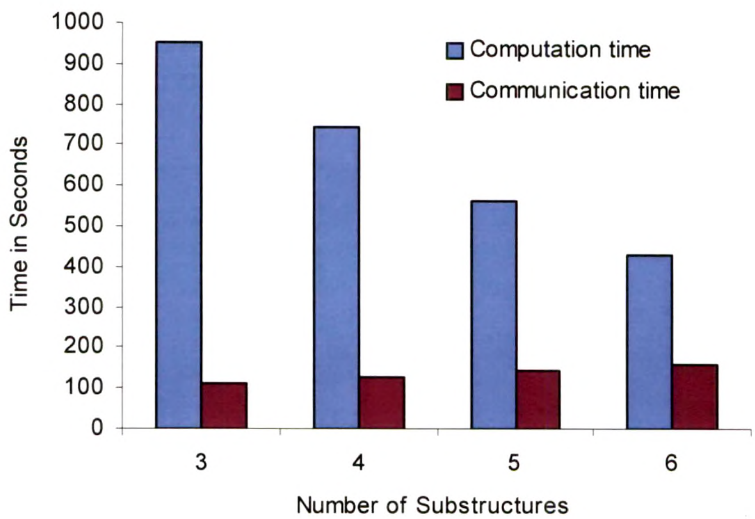


FIG. 6.29 COMPARISON OF COMPUTATION & COMMUNICATION TIME

6.11 EFFECT OF RATIO OF INTERNAL NODE TO INTERFACE NODE ON COMPUTATIONAL TIME

In previous sections it was observed that computational time for calculating substructure stiffness matrix and load vector is maximum due to renumbering of internal nodes and boundary or interface nodes. To study the effect of ratio of internal nodes to interface nodes, an example of beam subjected to pure bending as discussed in section 6.6, is considered. Due to symmetry and anti symmetry about the x and y axis respectively only one quarter of the plate is discretized in to 21600 elements, 21901 nodes and 43802 unknowns. For the discretization, plane stress quadrilateral element with linear displacement model is used. Quarter of the structure is divided into number of substructures for the analysis.

For different number of substructures, the number of internal nodes and interface nodes are shown in Table 6.10. As internal nodes are numbered first and all interface nodes are numbered last, the bandwidth of stiffness matrix generated from equilibrium equation increases which in turn increases the time required for derivation of substructure stiffness matrix and load vector. To observe the time required for substructure stiffness matrix and load vector all the substructures are run on the single computer. The average time required is shown in Table 6.10.

TABLE 6.10 TIME REQUIRED TO DERIVE SUBSTRUCTURE STIFF. MATRIX & LOAD VECTOR

No. of substructures	No. of internal nodes	No. of interface nodes	Ratio of internal to interface Nodes	Average time in Seconds to derive substructure stiffness matrix and load vector
2	10561	480	22.00	3440
3	6981	440	15.87	1764
4	5191	420	12.36	947
5	4117	408	10.09	608.6
6	3401	400	8.50	372.83
8	2506	390	6.43	166.75
10	1969	384	5.13	96.4
12	1611	380	4.24	69.7

Figure 6.30 shows the variation of time spent in deriving substructure stiffness matrix and load vector with various ratios of internal to interface nodes, from which it can be observed that the time increases rapidly with the increase in ratio of internal to interface nodes.

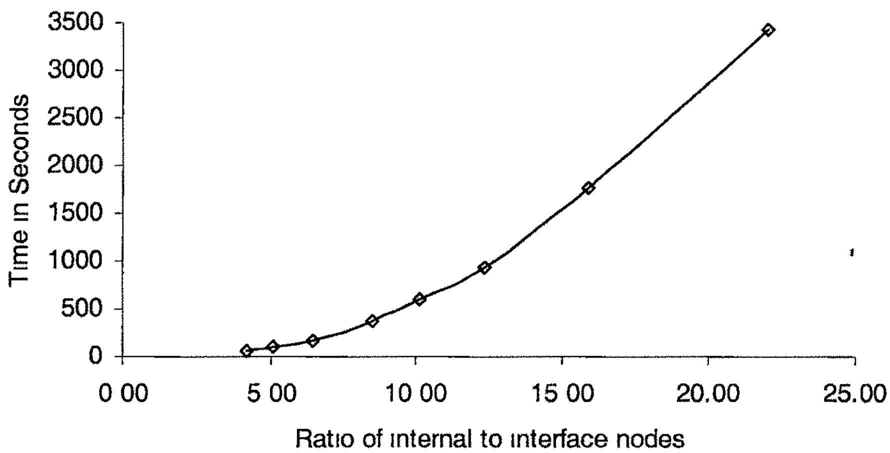


FIG. 6.30 VARIATION OF TIME WITH RATIO OF INTERNAL TO INTERFACE NODES

The number of equations (NEQ), half bandwidth (NB) and average time spent in seconds (T_{AV}) in various processes are presented in Table 6.11.

TABLE 6.11 TIME REQUIRED FOR VARIOUS PROCESSES

No. of Substructures		Process Name				
		CSUB1	CSUB2	CSUB3	CSUB4	CSUB5
2	NEQ	-	22082	1568	22082	-
	NB		21490	960	366	
	T_{AV}	1	3444	10	37	2
3	NEQ	-	14842	1916	14842	-
	NB		14330	880	366	
	T_{AV}	1	1786	14	26	2
4	NEQ	-	11222	2274	11222	
	NB		10750	840	366	
	T_{AV}	1	942	17	20	2
5	NEQ	-	9050	2632	9050	-
	NB		8602	816	366	
	T_{AV}	1	608.6	21	16.6	2
6	NEQ	-	7602	2990	7602	-
	NB		7170	800	366	
	T_{AV}	1	377.833	23	14.5	2
8	NEQ	-	5792	3706	5792	-
	NB		5380	780	366	
	T_{AV}	1	166.75	29	12	2
10	NEQ	-	4706	4422	4706	-
	NB		4303	768	366	
	T_{AV}	1	96.4	33	9.6	2
12	NEQ	-	3982	5138	3982	-
	NB		3590	760	366	
	T_{AV}	1	69.667	39	8.333	2

Table 6.12 shows comparison of time in sequential and distributed processing and speedup observed for different number of substructures or number of computers. The time in distributed processing consists of computation time and communication time on network of Pentium IV computers running at 2.4 GHz speed and having 256 MB RAM. With increase in number of substructures computational time reduces and communication time increases. Due to this after eight numbers of substructures the time speedup starts reducing. Figure 6.31 shows the variation of observed speedup and its comparison with ideal speed up, for various number of computers assigned to carry out the task.

TABLE 6.12 COMPARISON OF TIME IN SEQUENTIAL & DISTRIBUTED PROCESS

No. of Substructures	Sequential Processing	Distributed Processing (Sec)			Speedup
		Computation	Communication	Total	
2	6967	3494	89	3583	1.94
3	5387	1829	90	1919	2.81
4	3888	999	93	1092	3.56
5	3150	694	95	789	3.99
6	2380	436	97	533	4.47
8	1456	219	103	322	4.52
10	1096	143	106	249	4.40
12	978	121	110	231	4.23

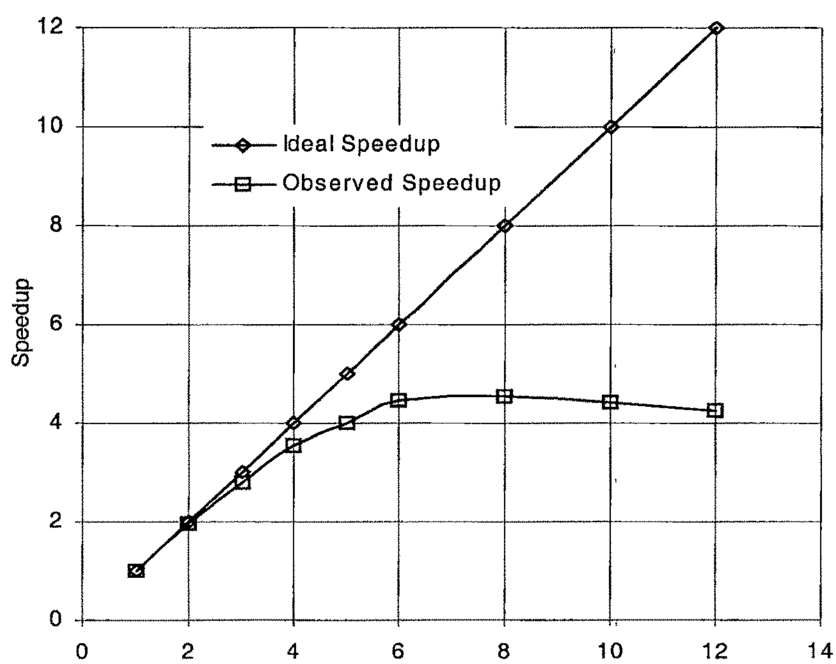


FIG. 6.31 COMPARISON OF SPEEDUP

6.12 CLOSING REMARKS

For large size problem finite element solution requires large memory and good processing speed. When such problems are solved on uni-processor computer, it may require large computational time. To improve computational speed, parallel processing using supercomputers or cluster of high performance workstations can be used but they are costly, not easily available and difficult to use. The economical alternative for high performance computing is to use network of computers for distributed processing. In this chapter finite element method for static analysis was implemented over Local Area Network using WebDedip environment. Here it was demonstrated that the finite element method can be easily implemented over distributed computing environment using substructure technique.

First of all examples discussed in section 6.6 and 6.7 were of smaller size in which use of distributed computing did not prove advantageous due to less computation time and comparatively more communication time required by these applications. However through these examples the application development using WebDedip became clear, the proper functioning of programs for various tasks as described in Table 6.1 was ensured. Also the feasibility of the distributed environment was proved. Otherwise to start with large size problem, difficulties may be faced for debugging of programs, and understanding working with distributed computing environment. In subsequent subsections larger size problems were considered to observe the improvement in computational speed using distributed processing.

For the applications like plane stress and plate bending analysis a computational efficiency of order 70-80% was observed. With increase in size of problem the computational efficiency can be further improved because with smaller problem communication overhead reduces computational efficiency.