2. LITERATURE REVIEW

2.1 INTRODUCTION

Generally the developments in technologies and their applications are drivers of each other. The same is true in case of computational technology and its applications in structural engineering. Since the early age of computers, applications of structural engineering utilized their full potential and applications requiring more computational power motivated the developments in computational technology.

The growing applications of parallel and distributed processing in structural engineering is contributed mainly due to:

- Advances in microelectronics, VLSI etc
- Networking and communication technology
- Advances in algorithms, which can exploit parallelism
- Rise in application requiring more computational power.

To utilize full potential of parallel and distributed processing algorithm specifically designed for hardware and application is needed. Since the development of multiprocessor computers, the papers related to their applications, algorithm to utilize full potential of hardware and development of software tools etc. have published in various journals and proceedings. As the developments in hardware progressed, various parallel processing platforms emerged with different characteristics. Researchers evolved different computation strategies for different kind of problems and reported improvements in computational efficiency in literature. Size of problems solved using parallel and distributed computing environment also increased with latest developments. The available literature covers characteristics of supercomputers or distributed computing environment used, software tools used for implementation, algorithm utilized and type of problem considered. Variation in computational efficiency with size of problem and varying number of processors or computers has also been included in the literature.

The literature review carried out here is divided into following categories and is presented in chronological order:

- General aspects of parallel and distributed processing
- Substructure technique and other algorithms
- Applications of linear analysis of skeletal structures
- Dynamic and nonlinear analysis of skeletal structures
- Static finite element analysis of continuum structures
- Dynamic and nonlinear finite element analysis of continuum structures
- Structural optimization applications
- Parallel processing in Neural Networks
- Literature related to laminated composite structures

2.2 GENERAL ASPECTS OF PARALLEL AND DISTRIBUTED PROCESSING

In literature various aspects of parallel and distributed processing and various applications, which could be implemented, on such environment have been discussed. Development of early supercomputers, their characteristics, and suitable hardware specific algorithms for applications were covered. This section reviews the literature about developments in computer hardware and its impact on structural engineering.

Developments in digital computer technology since early fifties were reviewed by **Utku et al.** in **1982** [11] and the parallelism existed between these developments and developments in analysis and design procedure of structural engineering were identified. The impact of distributed processing on analysis and design practices of structural engineer was assessed by examining these practices from data processing standpoint to identify key operations and databases and then fitting them to the characteristics of distributed processing. The merits and drawbacks of this technology in educating structural engineers were discussed and projections were made for the industry-academia relations in the distributed processing environment.

In a review article in **1987**, parallel processing technique was described by **Adeli et al.** [12] as promising approach to large scale engineering computations. The challenge to obtain high performance using powerful microprocessors in parallel is to reformulate the problem, develop parallel algorithm and device new computational strategies in order to utilize fully the capabilies of parallel machines. Parallel processing was reviewed including various interconnection networks in multi processors, commercial parallel processors, parallel programming languages and algorithms. Various civil engineering applications like finite element static and dynamic analysis, dynamic simulation, structural optimization, seismic migration were reviewed by them.

Rao et al. in **1990** [13] explored the use of supercomputers for solving computationally intensive problems of finite element. In this paper general aspect of parallel processing like architecture, speedup, efficiency etc. discussed. Solution of linear equations using direct and iterative methods along with their advantages w.r.t. parallel processing discussed. Application of parallel processing in various areas like static analysis, dynamic analysis, and nonlinear analysis explored. Also future trends in this area were described.

Rao et al. again in **1991** [14] discussed the role of parallel processing, various parallel processing machines and some commercially available parallel machines. They also reviewed various parallel processing applications in civil engineering. Various parallel processing systems like shared memory and message passing MIMD machines were discussed. They also covered parallel programming languages, parallel compilers and algorithms, which harness properly the capacity of parallel machines. Applications in complex and large size problems of solid mechanics, structural analysis, computational fluid dynamics, optimization etc. were briefly reviewed. Indian initiatives in the field of parallel processing by CDAC and NAL were discussed. Outstanding issues like modification of sequential programs, ideal parallel processing architecture, standardization of parallel compilers etc were raised.

Thangaraj in **1996** [15] discussed applications of parallel processing in real time hydrologic and hydraulic modeling. Parallel computing technology i.e. pipeling, vector, array and multiprocessor system were described briefly. Hydrological system included directly observable parallel pathways like streams in catchments, and obscure pathways in subsurface and ground water flow and also man made pathways like pipeline, storm sewer. Judicious exploitation of these pathways in multiprocessor environment helped in reducing computational requirements. Two types of parallelism i.e. system based and procedure based were identified and described briefly. Other applications in area of ground water flow and transport modeling were also described.

The advances in computer technology likely to impact structural analysis and design were reviewed by **Noor** in **1997** [5]. A brief summary of advances in microelectronics and networking technologies and human-computer interaction paradigms and techniques was given. The major features of new and projected computing systems including high performance computers, parallel processing machines, and small systems were described. The paper also reviewed advances in programming environment, numerical algorithms and computational strategies for new computing systems. The paper summarized some of the recent developments in computing systems during the recent past and near-term future and related them to structural analysis and design. The discussion was intended to give structural analysts and designers some insight into the potential of these systems for providing cost-effective solutions of complex structural problems and to stimulate research and development of necessary algorithms, firmware and software to realize this potential.

Also, the history, recent developments and trends in computational structures technology (CST), which blends the insightful modeling structural response with the computation methods, were summarized by **Noor** in **1999** [16]. Some recent advances in a number of CST area were described including: discretization techniques and element technology, computational material modeling, modeling of composite, sandwich and smart structures, computational tools and methodologies for life management, transient response analysis, strategies and numerical algorithms for new computing systems. Research areas in CST that have high potential for meeting the future technological needs were identified.

Appa Rao in **1999** [17] described parallel processing machines, advances in development of parallel processing algorithms and their application in area of finite element method. Different phases of finite element analysis and their adaptability for parallelization were discussed. Parallel computational strategies for structural analysis like domain-wise, operator-splitting, column-wise or row-wise algorithms, node/element-wise algorithms were covered. <u>Cost effective alternative of expensive supercomputer</u> was suggested with use of network of workstations with message passing libraries. <u>Activities at Structural Engineering</u>

Research Centre, Madras in development of parallel algorithms were described briefly with few applications.

Adeli in **2000** [18] again reviewed research papers published on parallel processing, supercomputing, distributed computing since 1994 with focus on three areas: analysis, optimization and control. The majority of journal articles published since 1987 dealt with fundamental issues and academic problems.

(1) Parallel Processing on shared and distributed memory machines:
A number of parallel algorithms for automatic partitioning, elastic and nonlinear analysis and optimization of 2-D framed structure subjected to static and dynamic loading using microtasking, vectorization were implemented over supercomputers like Cray, network of transputers etc. Problems of varying sizes were solved over different environments by researchers and computational efficiency and speedup reported. The contribution of researchers in development of efficient substructuring and domain decomposition techniques was described in the paper. The work done by researchers in areas of boundary element analysis, aeroelasticity and coupled fluid/structure problems, direct versus indirect equation solvers, meshless methods etc. were reported in this review paper. It also included references of work in vectorization and parallelization of conventional optimization technique as well as biologically inspired algorithms over supercomputers. Optimal active control for large structures, to minimize the damage to bridges, buildings and other structures due to earthquake, wind and other environmental loads, is computationally intensive due to complex eigen value problems. The work of researchers in this direction was also reported in the paper.

(2) Distributed Computing:
As cluster of personal computers or workstations networked together provided cost effective environment for high performance computing, researchers implanted various applications like finite element analysis, direct and iterative solution methods, seismic response of large structures, finite strip method, boundary element analysis, thermo-hydro-mechanical analysis, optimization using Genetic Algorithm etc. over network of computers using message passing libraries like PVM and MPI.

(3) Parallel computing and Object-oriented programming:

Due to increasing popularity of object-oriented programming in development of flexible, maintainable, reusable CAD/CAE software systems, researchers implemented the same over supercomputers.

In final comments, for researchers working in this area, list of important journals and contributors were given by **Adeli**.

2.3 SUBSTRUCTURE TECHNIQUE AND OTHER ALGORITHMS

Review of articles related to algorithms to implement structural engineering problems on parallel and distributed computing platform is carried out in this section. Mainly application of substructure technique in decomposing problem for static, dynamic and nonlinear analysis is covered. For large size problems of finite element, review of mesh generation and decomposition algorithms covered in literature is also included.

When the number of independent variables describing response of structure is large, it is usually more convenient to do analysis using substructures involving fraction of total number of independent variables at a time. For large scale computer one may still resorts to substructuring to minimize computing cost by reducing high core memory usage. Also for many analysis problems partitioning may be unavoidable. **Utku** in **1975** [19] suggested systematic substructuring, in which computer was instructed to partition the problem depending on available core memory with minimum number of partitions by means of the binary connectivity matrices. Two algorithms were outlined to obtain the connectivity matrix of the mesh and the supported structure systematically and economically. He claimed that this approach relieves the analyst from the burden of additional input.

Noor in **1975** [20] presented a simple and accurate reanalysis technique based on the reduced basis method in conjunction with the substructuring concept. He pointed out relative merits of the Taylor series expansion and reduced basis-substructuring techniques when applied to reanalysis of large and complex structures. The potential of proposed reanalysis technique was demonstrated by numerical example of transmission tower.

Kluttz et al. in **1977** [21] suggested that for every analysis problem requiring partitioning, there would be atleast one best option which produces least cost of computing in a given computing environment. Such partitioning may be identified from a table listing the required number of partitions and their sizes as a function of available rapid access memory (RAM) and the knowledge of the prevailing unit cost of RAM and peripheral processing units in the computing facility. This table may be generated by software, which require to scan only once the usual input data of the analysis problem. The generation of such tables cost nothing, yet, if the structural analysis software is designed to take advantage of the information provided by them, the analysis cost may be reduced drastically. The paper presented a computer program RAMPART that was developed to extract the partitioning information.

Noor in **1978** [22] discussed the status and developments of substructuring techniques and their applications to structural analysis and design. Various aspects like multilevel substructuring algorithm, use of hypermatrix and other sparse matrix schemes, use of substructuring in automated design systems and application of substructuring in elasto-plastic problems were described. Numerical examples were presented by him to demonstrate the reduction in the number of arithmetic operations and disk storage requirements using multilevel substructuring.

Owen et al. in **1982** [23] discussed application of substructuring technique to analysis of quasi-static elastoplastic problems. In this technique, the part of structure which remain elastic during deformation was defined as one substructure while the remainder of structure which undergo plastic deformation was defined as another substructure. The incremental / iterative process for the solution of elastoplastic problem and its application to substructure technique was described. To illustrate the savings in CPU time two numerical examples were included. It was shown that with small number of unknowns in plastic region compared to total number of unknowns, better economy could be achieved.

Farhat et al. in **1988** [24] presented general purpose parallel Fortran subroutines for the solution of sparse and dense symmetric systems of linear equations, which can be implemented easily on already existing sequential out-of-core direct solver. They were designed to run on a class of multiprocessors with local and/or shared memory. The idea was based on Doolittle reduction 'active column' solver. Numerical experiments were carried out on both the Intel Personal Supercomputer (local memory) and Encore Multimax (Shared Memory). The impact of hardware architecture on software implementation and performance was discussed.

Goehlich et al. in **1989** [25] reported a study on development, evaluation and implementation of parallel equation solver in the context of finite element method. A triangular decomposition approach was used and good parallel processing performance was obtained on a FLEX/32 multicomputer using upto seven processors. The parallel program was incorporated in MSC/NASTRAN and tested for several static stress analysis demonstration problems on CRAY X-MP, IBM 3090 and ALLIANT computers with upto four processors. It was shown that a parallel processing approach could significantly reduce execution time for large-scale finite element problems where the computational efficiency depends on size of problem, matrix bandwidth and type of computers.

The application of multi-level substructuring technique to the analysis of elasto-plastic and local nonlinear problems was studied by **DeRoeck et al.** in **1989** [26]. After review of principles and formulations, a practical implementation and typical flowchart of the procedure for technique was presented. The multilevel substructuring technique was found powerful and useful tool for solving complex structures consisting various isolated nonlinear portions and components. Modification was limited to those substructures that require a reformation of stiffness matrix. For simple problems where the rupture mechanism can be easily outlined, a two level substructuring scheme was found to be sufficient.

Saxena et al. in **1992** [27] discussed an automatic meshing schemes suitable for parallel processing, through recursive spatial decompositions, which inherit the hierarchical organization and the spatial addressablility of the underlying grid. The concept of a meshing operator for parallel processing was defined and algorithms for various stages of automatic meshing scheme were presented. A systematic simulation of fine- and coarse- grain parallel configurations was used to evaluate the performance of the meshing scheme.

Also in **1993, Saxena et al.** [28] discussed an automatic substructuring scheme based on RSD of the domain that could be closely integrated with the RSD based automatic meshing procedure. The hierarchical data structures used to represent RSD-based automatically derived meshes provided a one-to-one mapping between spatially decomposed sub-domains and analytical substructures. Such a hierarchical organization of substructures and inherent parallelism of RSD were exploited to design the substructuring scheme suitable for parallel processing. The CPU time for various stages of analysis was recorded on the ALLIANT FX/8 machine running as a serial machine. Speedup for examples from vectorization alone were reported.

Rao in **1994** [29] described concurrent finite element analysis using concurrent version of popular frontal solver known as multi frontal scheme. The suitability of frontal solver coupled with mesh decomposition for implementation on parallel processors was discussed. As this strategy need does not require assembly of global stiffness matrix, it can be applied more efficiently to very large and complex structures. This multi frontal based finite element algorithm was implemented on Magnum IV, a shared memory parallel processing platform. By considering a numerical example of plate bending problem it was demonstrated that proposed method is adaptive to concurrent processing.

Adeli et al. in **1995** [30] presented distributed algorithms for the finite element analysis of large structures on loosely coupled multicomputer such as a cluster of inexpensive workstations. The focus was on the development of a coarse grained preconditioned conjugate gradient (PCG) solver based on element-by-element approach to solve resulting system of linear equations. The problem of slow communication speed of ethernet network connected workstations was reduced by techniques such as redundant computations to eliminate communication, efficient data distribution and algorithmic restructuring to reduce communication frequency. Data distribution and data movement strategy based on set theory were presented.

Also in **1995, Farhat et al.** [31] developed TOP/DOMDEC - a totally Object-Oriented Program for automatic Domain Decomposition, a software tool and a software environment for mesh partitioning and parallel processing. It offered several state-of-art graph decomposition algorithm in user friendly environment. Generated mesh partitions could be smoothened and optimized for minimum interface and maximum load balance using non-deterministic optimization algorithm. The user interface included high speed three dimensional graphics, an interprocessor communication simulator for massively parallel systems and an output function with parallel I/O data structures.

Topping et al. in **1996** [32] reviewed some recent developments in computational technology applied to structural engineering like parallel and distributed processing, neural networks and genetic algorithms. General introduction of these developments was given. To increase the computational efficiency in finite element analysis, the mesh should be divided into a number of subdomains for distributed processing on multiple processors. It was demonstrated that neural networks and genetic algorithms provide useful tools for the analyst in decomposing finite element problems and these tools could be implemented in parallel. The parallel Subdomain Generation Method was presented and its superiority over sequential version was demonstrated.

For large size dam-foundation problems requirements of computer memory and hard disk space increases, which limits the use of finite element method for such problems. **Saini et al.** in **1997** [33] suggested substructuring technique of large dam-foundation system to provide solution to problem faced in one-stage finite element analysis. Two dimensional analysis of concrete gravity dam was carried out using substructure technique and comparison of execution time and hard disk storage requirements studied. It was concluded that substructure technique leads to significant reduction in both the computational time and hard disk storage requirements without any compromise in accuracy of the solution.

Watkins et al. also in **1997** [34] presented an efficient parallel procedure for the triangulation of real symmetric matrices. The objective of developing a parallel form of the Gauss-Doolittle method was to reduce overall solution time for the large problems and to increase computational efficiency. The method was implemented on distributed MIMD parallel computers. Speedup and efficiency results demonstrated that the method could be parallelized efficiently using appropriate number of processors

Wriggers et al. (1998) [35] presented an algebraic multigrid solver which could be applied as preconditioner for the conjugate gradient method. The solver was implemented in a parallel version of finite element program FEAP. The aim was to show the performance of these solvers on MIMD computers and to present a concept of porting finite element code to a parallel machine. Numerical examples presented to discuss speedup on T800 transputer system and PowerXplorer system.

Zucchini in **2000** [36] presented parallel preconditioned conjugate gradient iterative solver for finite element problems with coarse -mesh / fine-mesh formulation over IBM-RISC590 and Quadrics-QH1, a massively parallel SIMD machine with 128 processors. An efficient preconditioner was derived from the multigrid stiffness matrix. The application field was limited to topologically rectangular finite element grids. The results of test showed improvement over monogrid algorithm and a good parallel scalability and efficiency.

The work of **Sotelino (2003)** [37] was concerned with some of the parallel algorithms developed in the field of structural engineering in last two decades to take advantage of parallelism offered by the multiprocessor computers. The paper was a survey of parallel algorithms in which structural engineers were interested. Such algorithms included parallel solver (direct and iterative) for linear systems of algebraic equations, techniques for parallelization of finite element method and concurrent time-stepping algorithms for the solution of equations of evolution arising in structural dynamic problems.

2.4 APPLICATIONS OF LINEAR ANALYSIS OF SKELETAL STRUCTURES

This section reviews work carried out by researchers in area of static linear analysis of skeletal structures using parallel and distributed processing environment. Review includes characteristics of hardware, type of problems solved, algorithm used and efficiency observed during implementation.

Adeli et al. in **1989** [38] demonstrated the feasibility of parallel processing through the use of the notion of cheap concurrency and the concept of threads in the area of structural analysis. In this work, portion of the structural analysis problem was parallelized using Encore Parallel Threads (EPT) that operates on the top of UNIX operating system and C programming language. The issues of racing condition, synchronization and mappig were considered and discussed. Two synchronization mechanisms semaphores and monitors were employed and compared. Two different mapping strategies were implemented and studied. Applications to geodesic dome and 72-bar truss were considered for studying speedup and overall computational time performance. An overall efficiency of 90-95% was reported using 11 processors.

Again **Adeli et al. in 1992** [39] attempted parallel structural analysis of framed structures on shared memory multiprocessors with emphasis on parallelizing solution process. Parallel algorithms were developed to maintain workload balance using substructuring technique throughout the solution procedure and to assure best concurrent performance and speedup. Concurrency was achieved through the concept of threads and minimal storage schemes like skyline and compact form scheme were introduced to allow for large-scale structures. To avoid racing condition synchronization of threads or introduction of additional step were presented.

Further algorithms developed for concurrent analysis of framed structures on shared memory multiprocessor computers were implemented using C language and Encore Multimax multiprocessor and were applied to various truss and framed structures by **Adeli et al. (1992)** [40]. Balancing the internal nodes proved to be efficient for the steps of static condensation and calculation of non-interface displacements. The approach of switching between 'initial partitioning' based on a balance of elements and 'final partitioning' based on a balance of internal nodes within the subdomains resulted in an efficient concurrent processing paradigm. Overall efficiency of 90-100% was achieved for the case of 1200 degree-of-freedom structure.

Umesha et al. in **2000** [41] presented parallel version of Computer Adaptive Language (CAL) to analyse large structural mechanics problems on message passing systems. Several functions for inter-processor communication, domain decomposition, static condensation, parallel data structures, parallel equation solver were developed and incorporated in the parallel version of CAL. This allowed understanding the concept of parallel computing techniques in structural engineering. The paper explained computational strategies for finite element analysis suitable for parallel structural analysis and use of PARCAL was explained with numerical examples of 72 bar truss and 630 bar transmission tower. Master slave concept was used on network of transputers and cluster of workstations. An efficiency of about 60% was reported.

Foley (2001) [42] illustrated application of plastic zone analysis on large steel frames using advanced computational methods. A plastic zone analysis algorithm capable of using parallel processor and vector computation was discussed. Applicable measures for evaluating program speedup and efficiency on a Cray-YMP C90 multiprocessor supercomputer were described and program performance was evaluated for parallel and vector processing. Nonlinear response including postcritical branches of three large scale fully restrained and partially restrained steel framework was computed and the study indicated advanced analysis of steel frames could be accomplished using plastic zone analysis methods and alternate computational strategies.

In 2002 **Umesha et al.** [43] presented algorithms for parallel finite element analysis using substructuring technique on message passing parallel computer using PARCAL. Divide and conquer algorithm of parallel computing was discussed. A numerical example of microwave tower was solved by considering 2, 3, 4 subdomains and speedup in each case studied. Parametric study of degrees of freedom, number of processors on efficiency of computing was carried out.

2.5 DYNAMIC AND NONLINEAR ANALYSIS OF SKELETAL STRUCTURES

This section reviews literature about implementation of dynamic and nonlinear analysis of skeletal structures on parallel and distributed processing. Type of hardware and algorithm used, problems solved and computational efficiency observed are included in review.

Utku et al. in **1982** [44] reviewed numerical solution of nonlinear equilibrium problems of structures using Newton-Raphson type iterations. Organization and flow of data for various types of digital computers like, single-processor/ single-level memory, single-processor/two-level memory, vector-processor/two-level memory and parallel processors with and without substructuring was discussed. The effects of the relative costs of computation, memory and data transfer on substructuring were mentioned. The idea of assigning comparable size substructures to parallel processors was exploited.

Hajjar et al. in **1988** [45] presented a strategy for the solution of fully nonlinear transient structural dynamics problem in a coarse grained parallel processing environment. Emphasis was placed on the analysis of three-dimensional framed structures subjected to arbitrary dynamic loading and in particular on steel building frames subjected to earthquake loading. The implicit domain decomposition method was described using substructuring technique with a preconditioned conjugate gradient algorithm for the iterative solution of the reduced set of unknowns along the substructure interfaces. Substructuring was shown to provide a natural preconditioner for effective parallel iterative solution. Numerical examples of transmission line tower and 30 storey structure were discussed for timing study of the domain decomposition algorithm. The algorithm was implemented on a bus architecture. A feasibility domain decomposition algorithm for nonlinear structural dynamic problems using FEM was mentioned.

Chien et al. in **1989** [46] developed methods for parallel processing of finite element solutions of large truss structures. The parallel processing techniques were implemented in two stages, i.e. repeated forming of the nonlinear global stiffness matrix and the solving of the global system of equations. Between the two parallel forming methods for global stiffness matrix, a special nodal numbering scheme was proposed to avoid a synchronization problem. The parallel Gaussian elimination procedure was found very efficient on multiprocessor computers and its efficiency remained high for matrices with large bandwidths.

In **1993 Shieh** [47] formulated two fully implicit and two semi-implicit sets of finite element method-based numerical algorithms for transient response analysis of space frame and truss structures in a massively parallel processing (MPP) environment. All algorithm sets used an implicit force calculation/vector equation of motion assembly procedure. The semi-implicit algorithms were based on the explicit central difference (CD) and the fourth-order Runge-Kutta (RK4) schemes, respectively, in conjunction with the use of mass lumping techniques so that solution of the recurrence equations for unknown displacements was reduced to a trivial diagonal matrix inversion operation. The fully implicit algorithm sets were based on the Newmark Beta (NB) and CD schemes, respectively, in conjunction with use of the (iterative) preconditioned conjugate gradient (PCG) method for solving the linear algebraic recurrence equations. The semi-implicit algorithm sets were fully implemented and assessed on a Massively Parallel Processor CM-2 computer.

The method of substructuring and the parallel processing technique of multitasking were applied in **1994** by **Foley et al.** [48] in the analysis of high rise structural frameworks. The performance of the proposed method was measured in both wall-clock time and connect time when executed in batch environment on a Cray Y-MP C90 supercomputer. An attempt was made to quantify the optimum number of processors that should be used in the analysis of rectangular frameworks based on the partitioning algorithm. Based on the analysis of several high-rise planer structures, load-deformation response subjected to propotional and nonproportional load was given. In addition speedup for the parallel code verses vector code was presented and performance characteristics were discussed.

Farhat et al. (1997) [49] highlighted some key elements of approach for the solution of large-scale three dimensional nonlinear aeroelastic problems on high performance computational platforms, which included three field arbitrary Lagrangian-Eulerian (ALE) finite volume / element formulation for the coupled fluid/structure problems, the derivation of a geometric conservation law for three dimensional fluid problems with moving boundaries and unstructured deformable meshes and solution of corresponding coupled semi-discrete equations with a partitioned manner. They presented a family of mixed explicit / implicit staggered solution algorithms, and discussed them with reference to accuracy, stability, subcycling and parallel processing. A general framework for the solution of coupled aeroelastic problems on heterogeneous and/or parallel computational platforms was described and some preliminary numerical investigations on iPSC-860, Paragon XP/S and CRAY T3D massively parallel systems were illustrated with aeroelasic and aerodynamic transient response of three dimensional wing in transonic airstreams.

The work of **Modak et. al. (2002)** [50] addressed the time history analysis of structures subjected to dynamic loads using high performance computing environment. Structural mechanics, parallel computing and object-oriented programming methodologies were integrated to design and implement frameworks for parallel and sequential transient finite element analysis. For parallel transient analysis, the finite element framework FE++ was extended to PTFE++ by integrating parallel transient solution algorithms. An example of 30 storey three-bay structural frame was solved on the IBM 9076 model 304 Scalable POWER-parallel (SP2) supercomputer.

2.6 STATIC FINITE ELEMENT ANALYSIS OF CONTINUUM STRUCTURES

Applications of parallel and distributed processing in static finite element analysis published by various researchers are reviewed in this section. Review includes hardware specific issues, tools used for implementations and computational efficiency observed in solving different size of problems.

Vanluchene et al. in **1986** [51] presented linear and nonlinear finite element software development considerations for vector processors. Various issues like performance measurement, data management, element level calculations and nonlinear problem solution were discussed. All work was implemented on the Control Data CYBER-205 at Purdue university which was the largest and fastest computer at that time. Thorough knowledge was required of special algorithm development techniques to exploit the inherent parallel or independent nature of numerical calculations which allow finite element system to show substantial performance on vector processor supercomputer. A sample problem was presented to demonstrate software performance.

In **1988, Zois** [52] presented parallel processing techniques for generation of element stiffnesses, element loads, assembly process and evaluation of element stresses. A portable and adaptable Parallel Finite Element System (PARFES) for linear static analysis of structures was proposed for MIMD parallel computer as well as parallel computational model using programming language (CSP and OCCAM). For illustration eight nodded plane stress / strain element under uniformly distributed load was considered. PARFES Consists of three major parts as: Distribution of input data to P processors (serial to parallel), FE structural analysis with displacement model using P processors (parallel) and Collection of output data from P processors (parallel to serial).

Also **Zois** in **1988** [53] included parallel system solution method called Asynchronous Systolic Multiple Gauss Jordon (ASMGJ) on a message passing MIMD parallel computer. The ASMGJ method consisted a computational algorithm, a communication algorithm and a storage-handling algorithm. These algorithms were presented along with their application examples on a parallel computer. The ASMGJ method was implemented as part of PARFES on the Transputer Development system (TDS) executing on VAX8600.

El-Sayed et al. (1990) [54] presented an approach for the parallel computation of finite elements with a small number of processors. The basic concept was to divide the analyzed structure into substructures. Each substructure analysed on a separate processor concurrently. To demonstrate the approach an algorithm for the linear static finite element analysis of two substructures was developed. The formulation of necessary information to be communicated and the numerical solution steps for the parallel computation were discussed. The general speedups and efficiencies for some test cases on the CRAY X-MP were presented.

The design of complex engineering systems such as advanced aircraft structures and offshore platforms require large order finite element and/or finite difference model and excessive computation demands both calculation speed and fast information management. The driving forces for computer simulation of the nonlinear dynamic response of structures and implementation of parallel FEM systems on high-speed multiprocessors were reliable simulation of automotive and aircraft crash phenomena and the increased performance of the computers. Two algorithms for parallel stiffness generation based on subdomain parallelism and element by element were discussed by **Chiang et al.** in **1990** [55]. Based on numerical examples it was shown that, both the approaches for parallel/vector generation of element matrices were feasible for shared / local memory computers.

Adeli et al. (1995) [56] implemented distributed algorithms for the finite element analysis of structures on a cluster of six IBM RC/6000 workstations. A common class of object oriented data structures developed for generation and display of data distribution among the workstations and interprocess communication scheme. In spite of the high latency and low bandwidth of the Ethernet networks, an overall parallelization efficiency of 75-90% was observed for large structural models discretized in few thousand elements. The research suggested that with the emergence of high speed communication networks, workstation clusters could be efficient alternative to high performance computing due to cost / performance over supercomputers.

Adaptive finite element method can improve reliability of finite element solution, which is computationally expensive and requires parallel and economic high performance computing environments. **Annamalai et al.** in **1999** [57] presented parallel implementation of AFEA on a cluster of workstation and its efficiency was illustrated by examples. A library PAVE (PArallel Virtual Environment) was developed to take care of all the essential needs of message passing/communication and to relieve the programmer from the complexities of PVM and MPI.

The accuracy and reliability of finite element analysis can be improved based on error estimates, refinement strategies and adaptive meshing techniques leading to adaptive finite element analysis. As Adaptive FEA was much compute intensive, **Krishnamoorthy** in **2000** [58] suggested use of high performance computing. For distributed computing, network of workstations or cluster of computers also known as commodity supercomputing, was used. To cater the needs of software engineers, a library PAVE (Parallel Virtual Environment) was developed at IIT Madras. Examples using an efficient AFEA software on homogeneous and heterogeneous computing environment were discussed.

DeSantiago et al. (2000) [59] investigated use of network of workstations for solving the incompressible Navier-Stokes equations using finite element method. A 'pool of tasks' strategy was proposed and employed for achieving proper load balance in case of varying computational resources across the network, which resulted dynamic load balancing that allowed the program to adjust the changing user loads on the network. A fault recovery scheme that permit the program to proceed in the event of failure was also presented. The implementation was a hybrid of the SPMD and master-slave paradigms using message passing libraries of PVM. It was noted that as the number of workstations employed increases, the efficiency in comparison of ideal speedup of distributed model decreases.

Gullerud et al. in **2001** [60] described a coarse –grain parallel implementation of a linear preconditioned conjugate gradient solver using an element-by-element architecture and prconditioner for computation. The solver was implemented within a nonlinear, implicit finite element code using MPI for message passing to provide portable parallel execution on shared, distributed and distributed-shared memory computers. The implementation coupled an unstructured dependency graph with new balanced graph-coloring algorithm to schedule parallel computations within and across domains. Three example problems with upto 158000 elements and 180000 nodes were analyzed on SGIcray Origin 2000 machine. The implementation of Hughes-Winget predonditioner showed 80% parallel efficiency over 48 processors.

Parallel implementation of an unstructured finite-element solver using the preconditioned conjugate gradient (PCG) method was described by **Thiagarajan et al.** in **2002** [61]. High performance Fortran was used with the implementation based on 32-node Pentium II 350 Mhz cluster running Linux. The PCG solver was set up for the element-by-element method which is suitable for very large problems. Two algorithms, one of which store the stiffness matrix of all elements in every processor while other store the stiffness matrix of only those elements that contribute to equations allotted to that processors were described. For the case studies, it was observed that the optimal number of processors for a given problem size was achieved when CPU time and C&S time balanced which depended on problem size.

Also in **2002, Hsieh et al.** [62] integrated general sparse matrix and parallel computing technologies as a finite element solution of large-scale structural problems in a PC cluster environment. The general sparse matrix technique was first employed to reduce execution time and storage requirements for solving the simultaneous equilibrium equations and to further reduce computational time, two parallel processing approaches for sharing computational workloads among collaborating processors were investigated. One approach adopted publicly available parallel equation solver called SPOOLES, to directly solve the sparse FE equations while the other employed parallel substructure method for large-scale finite element solution. Numerical studies were conducted to investigate effectiveness in reducing computational time and storage requirements in large scale FE analysis.

Gupta et al. (2003) [63] presented analysis of stresses developed in anchorage zone in prestressed post-tensioned concrete beam using finite element method on supercomputer PARAM10000. A computer code developed using MPI and C capable of solving plane stress and strain problems. Anchorage zone of post-tensioned beam was modeled using as 2 dimensional plane stress problem and analysis for different loadings was carried out. CST element was used for the analysis. The beam was discretized using 1136 triangular elements and 613 nodes resulted in 1226 unknowns. The most computationally expensive process of solving linear equations using matrix inversion method was parallelized to reduce time. The different components of computational time and their variation with number of processors were studied and maximum speedup of 4.6 was reported with 7 processors.

Gupta et al. (2004) [64] presented analysis of stresses developed in spalling zone in prestressed post-tensioned concrete beam using supercomputer PARAM10000. A computer code was developed for structural analysis using finite element method on the platform of PARAM10000. The prestressed post-tensioned concrete beam was discretized using 4800 three nodded triangular elements with 2501 nodes resulting in 5002 unknowns. The problem was analysed by increasing the number of processors from one to five. Computation and communication time were presented. A maximum speedup of 4.59 was observed for five number of processors.

2.7 DYNAMIC & NONLINEAR FE ANALYSIS OF CONTINUUM STRUCTURES

Implementations of dynamic and nonlinear finite element analysis over parallel processing environment are reviewed in present section. Various parameters considered in the implementations like type of hardware and software tools, size of problems and speedup for different number of processors are covered in the review.

In **1990, Malone** [65] developed a parallel nonlinear dynamic finite element analysis code (PANDA) to predict response of three-dimensional shell structures for execution on a hypercube computer. This code was implemented in conjunction with available mesh decomposition algorithm and the performance results of the parallel code was tested on a 32-processor Intel iPSC/d5 hypercube. Performance results were obtained for a range of problems, which exhibit large deformation, elastic-plastic dynamic response, including an idealized automotive lower side rail subjected to impact loading. Efficiency of 99% was obtained by assigning as few as eight shell elements to each processors. It was demonstrated that this high efficiency decreases as the number of processor increases. A massively parallel hypercube machine have enormous potential for solving complex structural analysis problems which may be discretized in 10000 or more elements.

The use of the substructuring technique for the solution of two-dimensional nonlinear problems of dynamic response with the direct time integration method was presented by **Sheu et al. (1990)** [66]. The object was to develop schemes that considerably reduce the computational expense in analysis of locally and fully nonlinear dynamic problems as compared with a traditional method. After a review of the principle of the incremental equilibrium equation of motion, some practical aspects were dealt with in detail. Two computer programs for elasto-plastic dynamic analysis of axisymmetric shell, plane stress / strain structures were developed using the substructuring technique and Newmark step-by-step integration method in a predictor-corrector form.

As in the area of crash impact, research was urgently required on the development and evaluation of parallel methods for crash dynamics analysis of complex nonlinear finite element and/or finite difference problems, an investigation of selected nonlinear dynamic algorithms using appropriate parallel computers was carried out by **Chiang et al.** in **1990** [67]. Implicit methods as those of the Newmark type which build on the Cholesky decomposition strategy and explicit methods such as the central difference time integration method were discussed. Both the implicit and explicit dynamic algorithms were investigated on FLEX/32 shared memory multicomputer and the intel iPSC Hypercube local memory computer.

Hsieh et al. (1993) [68] investigated, developed and demonstrated coarse grained parallel-processing strategies for nonlinear dynamic simulations of rotating bladed-disk assemblies. Numerical algorithms for parallel nonlinear solutions and techniques to effect load balancing among processors were suggested. The parallel environment employed was a distributed-memory, coarse grained one consisting of networked workstations. A parallel explicit time integration method was employed and automatic domain partitioning techniques were investigated for load balancing among processors.

Jaques et al. in **1996** [69] explored the use of parallel processing techniques in the form of transputer technology to reduce the processing time for non-linear finite element problems. Also the levels of inherent parallelism within finite element method were identified and results of applying these concepts in terms of processing time reduction compared to linear processing and domain decomposition were reported. Two programs were written in OCCAM, one which used the divide and conquer approach and the other which used the inherent parallelism approach. The resultant parallel processing structures were implemented on a Meiko computing surface and used four T800 25 MHz transputers. It was observed that inherent parallelism approach give better efficiency than divide and conquer approach.

In the year **2000, Sziveri et al.** [70] presented research on the use of homogeneous parallel and heterogeneous distributed computers for finite element analysis of transient dynamic problems using MPI. Based on the discussion of appropriate computer architecture a definition of computational efficiency was defined for heterogeneously distributed finite element analysis. A program for the dynamic explicit transient analysis of reinforced concrete plates was implemented in different parallel environments, on a shared memory and a distributed memory MIMD architecture. It was demonstrated that the code for homogeneous computing system needs to be revised for heterogeneously distributed systems.

Also in year **2000, Namazifard et al.** [71] presented use of standard message-passing interface (MPI) to parallelize Newmark's method. The linear matrix equation encountered at each time step was solved using a preconditioned conjugate gradient algorithm. On a basis of degree-of-freedom, data were distributed over the processors of a given parallel computer, which produced effective load balance between the processors and lead to highly parallelized code. The portability of the implementation of the proposed scheme was tested by solving some simple problems on two different machines: an SGI Origin2000 and an IBM SP2. The measured times demonstrated the efficiency of the approach and highlighted the maintenance advantages arised from parallel library like MPI.

Rao (2002) [72] presented a parallel mixed time integration algorithm formulated by synthesizing the implicit and explicit time integration techniques as an extension of mixed time integration algorithm. The parallel algorithm for nonlinear dynamic response of structures employing mixed time integration technique developed within broad framework of domain decomposition. Parallel finite element code was developed using portable Message Passing Interface software development environment. Numerical examples were attempted on PARAM10000 to test accuracy and performance of the algorithm and found to be adaptive to parallel processing.

Romero et al. in **2002** [73] discussed the parallelization of complex three dimensional software for nonlinear analysis of R.C. building structures. The aim was the improvement of run time in a nonlinear analysis program for large 3-D structure, evaluating computational time used by appropriate parallel algorithm and to develop scalable and portable computational strategies for nonlinear RC 3-D frames. The nonlinear finite element model adopted fiber decomposition approach for the cross-section of beam elements to capture nonlinear behavior of concrete. The parallelization strategy was designed for following three items: the numerical stability of nonlinear procedure, the parallel sparse equation solver and application of heterogeneous hardware. Significant improvement in the runtime was observed and Data transmission through the switch MIRINET was noted faster than in a previous analysis.

The efficient finite element analysis of shell structures with highly nonlinear behavior was presented by **Rottner et al. in 2002** [74]. The coupled nonlinear system of equations resulting from the FE discretization were solved using Newton-like procedures. Parallel computers offered major capabilities to reduce the CPU time needed for the fine discretization which resulted in very large system of equations. A geometrical approach for parallelization was used, and standard methods for the graph partitioning were employed. In the analysis of shell structures with tendencies to buckle, static and dynamic approaches were discussed considering both physical and computational aspects.

A new parallel algorithm based on domain decomposition technique and Newmark-β method was presented by **Rao et al. (2003)** [75]. The parallel overlapped domain decomposition algorithm proposed by splitting the mass, damping and stiffness matrices arises out of finite element discretisation of a given structure. A predictor-corrector scheme was formulated for iteratively improving solution in each step. A computer program developed and implemented with MPI as software development environment and PARAM10000 MIMD parallel computer was used to evaluate performance. Numerical studies indicated the better performance of proposed algorithm than the conventional domain decomposition algorithm.

2.8 STRUCTURAL OPTIMIZATION APPLICATIONS

Various techniques like gradient based methods, biologically inspired genetic algorithms for optimization of structures and their implementation over different types of parallel and distributed processing hardware are reviewed in this section. Suitable algorithms for specific hardware, implementation of various problems and efficiency observed as covered in the literature are included in the review.

The real benefit of structural optimization for large structures like full vehicle and full aircraft could not be taken due to computational time and memory requirements, which could be achieved by employing multiprocessor computers. **Hsiung et al.** in **1993** [76] discussed parallel processing of structural optimization problems with parallel structural analysis on Cray X-MP. Two methods for interfacing the finite element analysis software with the optimization software for parallel structural optimization were developed and tested. In these methods, a parallel finite element computation technique with separate substructures was utilized. The optimized structure was decomposed into several substructures. One processor on the Cray X-MP system was chosen to execute the optimization calculation and the finite element analysis of one substructure. The other processors were used to perform the structural analysis on the assigned substructures.

As Genetic-Algorithm based structural optimization could be parallelized on new generation multiprocessors, **Adeli** [77] presented a mixed computational model for GA based structural optimization of large space structures on massively parallel supercomputers. He exploited in **1995** Parallelism at both coarse-grained design optimization level in genetic search using the MIMD model of computing and fine grained fitness function evaluation level using SIMD model of computing. The later model involved the development of data parallel iterative PCG algorithm for solution of linear algorithm. The model was implemented on Connection Machine CM-5 and applied to optimization of space steel structures subjected to the constraints as per AISC specifications.

Kumar et al. (1995) [78] discussed the distributed algorithm for minimum weight design of large structures on a network of IBM RISC/6000 workstations using biologically inspired genetic algorithms. For message passing between the workstations, communication construct from Parallel Virtual Machine (PVM) was used. Two examples of space trusses were discussed and performance estimates were provided based on the granularity and parallelization efficiency of the distributed model. For large structures a parallelization efficiency of 90% was observed. Based on the scalability of the distributed GA cluster of workstation demonstrated a cost effective alternative for high performance computing.

Again in **1995, Adeli et al.** [79] presented distributed Genetic Algorithm for optimization of large structures on cluster of workstation connected via local area network (LAN). The selection of genetic algorithm was based on its adaptability to a high degree of parallelism. Two different approaches were used to transform the constrained structural optimization to an unconstrained problem: Penalty function method and Augmented lagrangian approach. For the solution of the resulting simultaneous linear equations, the iterative preconditioned conjugate gradient (PCG) method was used because of its low memory requirement. A dynamic load-balancing mechanism was developed to account for the unpredictable multi-user multitasking environment of network of workstations, heterogeneity of machines, and indeterminate nature of the iterative PCG equation solver. The algorithm was applied to space structure subjected to horizontal and vertical loads and constraints of the AISC specification.

Weinert et al. in **1996** [80] presented decomposition and parallelization strategies for optimization of large scale structures and applied to the shape optimization of complex shells of revolution. PARDEC, Parallel Decomposition strategy, for solution of complex optimization, which cannot be solved efficiently on sequential optimization, was discussed. Global optimization problem was divided into smaller parallel solvable subproblems which were optimal. The subtasks 'structural analysis' and 'sensitivity analysis' within optimization process were parallelized for better computational efficiency. All realized parallel strategies were implemented on parallel computing systems and were verified by optimization of an automotive wheel.

Thierauf et al. (1998) [81] presented parallelization strategy for solving structural optimization problems with discrete variables using evolution strategies, which combine the concept of artificial survival of the fittest with evolutionary operators to form robust search mechanism. As evolution strategies work simultaneously with a population of design points in the space of variables, it could be implemented in parallel computing environment. Two ways of parallel implementation, a directly parallelized evolution strategy and parallel subevolution strategy, on a Power-Xplorer parallel system and on a workstation cluster using PVM were described. Two numerical examples were presented to study speedup.

Also in **1998, Soegiarso et al.** [82] presented an efficient parallel-vector algorithm for optimization of large space truss steel structures subjected to realistic code specified constraints and multiple loading conditions. The implementation was carried out on shared memory Cray YMP8/864 supercomputer with eight processors. The algorithm was used for several high-rise building structures and it was shown that parallel processing and vectorization performance improve with increase in size of structure. For a largest 81 storey structure, speedup of 6.58 with eight processors was achieved.

In design optimization of large structural system with gradient based optimization methods, design sensitivity analysis is time consuming process, which can be made faster using parallel processing, **Umesha et al. (1999)** [83] described two types of approaches for parallel implementation of design sensitivity analysis i.e. single level parallelism and multi level parallelism. In single level parallelism, coefficients of gradient matrices were calculated for entire structure while in multi level parallelism, structure was decomposed in to substructures and gradient matrices of all substructures were calculated in parallel. Examples were presented based on both the approaches and good efficiency was observed. For parallel processing, cluster of workstation and transputer network systems with message passing interface were used. Parallel algorithms for sensitivity analysis were developed within software platform of Computer Adaptive Language (CAL).

Chan et al. in the year **2000** [84] described new workload distribution strategies which were used with fine grain parallel analysis method in order to parallelise individual steps of the gradient based design optimization procedure. A simple plane frame program was adapted to act as a test bed for all strategies presented. The complete parallel design method was implemented and successfully tested on a relatively coarse grain Transtech Paramid computer. Reasonably high efficiencies were maintained even when a high number of processors were used. The results showed that the efficiency falls relatively sharply with increasing number of processors.

Patnaik et al. (2000) [85] considered design optimization of large structures through a substructure strategy, in which structure was divided into smaller substructures that were clustered to obtain a sequence of sub problems. Substructure strategies, in sequential and parallel computational environments on Cray-YMP supercomputer were implemented in a test bed CometBoards. The issues encountered during substructure solution and their resolution were discussed under: (1) coupling and constraint formulation, (2) differences in optimal solutions, and (3) amount of computation. Coupling between subproblems and separating constraints into local and global sets promoted convergence of the iterative process. The substructure optimization could be computation-intensive, but in a parallel computational mode it could effectively use assigned processors.

Due to high cost associated with supercomputers and inability to test complicated mathematical model on Personal Computers (PCs), there appeared gap in technology transfer to the industry. **Balla et al.** in **2000** [86] reported results from an exploratory research that implemented a complicated optimization model based on distributed genetic algorithm on a network of PCs. In network, PCs were hardwired using 16 bit 10Base-T Ethernet cards and were made accessible using Peer-to-Peer networking capabilities of Windows 95/NT operating system. The inherent parallelism associated with genetic algorithms coupled with relatively small data exchange between the computers resulted in significant reduction in computer time. The proposed framework was adopted for water distribution network.

Sarma et al. (2001) [87] described optimization of very large steel structures subjected to AISC-ASD and LRFD specifications on high-performance multiprocessor machines using biologically inspired genetic algorithms. Fuzzy genetic algorithms were presented for optimization using distributed memory Message Passing Interface (MPI) with two different schemes: processor farming scheme and migration scheme. Subsequently two bilevel parallel GAs were presented for large-scale structural optimization through judicious combination of shared memory data parallel processing using OpenMP Application Programming Interface and distributed memory message passing parallel processing using MPI. Speedup results indicated that bilevel algorithms were superior to MPI algorithm.

Park et al. in **2002** [88] presented distributed computational model for structural analysis of large-scale structures on cluster of personal computers based on two different levels of substructuring techniques. They were implemented on a collection of Pentium processors connected via a 10 Mbit/s Ethernet LAN. The effect of different levels and sizes of substructures on performance of distributed iplementation was investigated. The effectiveness of algorithms was accessed by applying them to static analysis of moment resisting frames and space structures. This algorithm allowed a larger granularity of parallel tasks, which could improve performance.

2.9 PARALLEL PROCESSING IN NEURAL NETWORKS

As training of neural networks using large set of input and output data is computational intensive, the efforts of researcher for better efficiency using parallel processing are reviewed in this section. Type of problem considered, hardware used, computational strategy used and results obtained are covered in review.

Topping et al. in **1997** [89] described a parallel processing implementation for neural computing and its application to finite element mesh decomposition. The parallelized neural network software developed was based on the public domain NASA developed program NETS 2.01, which was based on the back propagation algorithm. The principal focus of the research was on parallel implementation on an array of transputers. A comparison between sequential and parallel versions was given and finally a structural design problem concerned with finite element mesh generation was solved using the parallel neural network software.

2.10 LITERATURE RELATED TO LAMINATED COMPOSITE STRUCTURES

Applications of parallel and distributed processing in analysis of laminated composites under various loading condition over parallel processing environment are included in this section. Type of hardware platform used, algorithm utilized and speedup observed for different problem are reviewed.

Watson et al. (1996) [90] presented a computational strategy for calculating sensitivity coefficients for the non-linear large-deflection and postbuckling response of laminated composite structures on distributed-memory parallel computers. The key elements of the proposed strategy were: (1) multiple-parameter reduced basis technique, (2) parallel sparse equation solver based on multilevel substructuring, and (3) multilevel parallel procedure for evaluating hierarchical sensitivity coefficients. These hierarchical sensitivity coefficients measured the sensitivity of composite structure response to variation in parameters like laminate, layer and micromechanical properties. The effectiveness of the strategy was accessed by solving problems of stiffened composite panels with cutouts subjected to mechanical and thermal load on distributed memory computers like Intel Paragon, Cray T3D and IBM SP2.

Fiber reinforced plastic composites (FRPC) consisting of number of layers is heterogeneous material and stress analysis of real life problem imposes great demands on computational requirements. **Shah et al.** in **2000** [91] reviewed higher order shear deformation theory for stress analysis of laminated composite shells. For improving computational efficiency, supercomputer developed at

CDAC (PARAM 10000) was used. The most time consuming part of FEM i.e. solution of linear equation was parallelized. For factorization of stiffness matrix Choleskey method was used. For parallel implementation of Cholesky solver master-worker concept was used. Three examples were solved with different number of processors. Time required for I/O, communication and total elapsed time were compared for different number of processors.

Kant et al. (2003) [92] implemented thermo-mechanical analysis of FRP structures on distributed memory high performance parallel machine PARAM10000. A higher order facet element was used and lamina stresses under thermal loading were validated with existing analytical results. Two different parallel solvers, Cholesky and Preconditioned Conjugate Gradient were used to solve the system of linear equations. Both the solvers exploited the sparsity of global matrix and carried out load distribution across the number of processors in column oriented fashion. The elemental stiffness generation and assembly were also carried out in parallel. Though the Cholesky solver gave better absolute time compared to Conjugate solver, the latter showed more scalability on parallel machine. Good scalability upto 16 processors was observed because inter-processor communication was less in iterative solver than in direct solver.