

Local Interface Formulation for Structural Nonlinear Dynamic Finite Element Analysis

Chao-Jiang FU

College of Civil Engineering, Fujian University of Technology, Fuzhou, 350108, China

cjfu@163.com

Keywords: Finite Element Method; Nonlinear Dynamics; Domain Decomposition; Local Interface Formulation; Parallel Computing

Abstract. The local interface formulation for nonlinear dynamic analysis of large structures is presented using domain decomposition and preconditioned conjugate gradient (PCG) technique. In the formulation, the PCG algorithm is formulated using the unassembled local Schur complement matrices of subdomains. Incomplete Cholesky preconditioner is employed. Newmark- β average acceleration technique is employed for time integration. Numerical results indicate that the proposed parallel formulation is superior in performance when compared to the conventional domain decomposition algorithms with parallel direct solver.

Introduction

The computational complexities of structural dynamic nonlinear finite element analysis may become very large for many practical engineering problems. Implicit formulation of finite element method provides a viable approach to resolve the complete dynamic response history of structures. Repeated solution of the linearised equilibrium equations for each time step within Newton iterative method for solving nonlinear equations takes a lot of computation time. Parallel computing can significantly reduce the time of structure analysis, so it has important significance to study and develop effective parallel algorithm.

In recent years, a number of methods based on domain decomposition procedures have been proposed for parallel solution of dynamic equilibrium equations. These domain decomposition methods can be classified as conventional domain decomposition methods, iterative domain decomposition methods. In conventional domain decomposition method[1], direct solvers are employed. Iterative domain decomposition method[2] invariably employs parallel iterative solvers like linear preconditioned conjugate gradient method (LPCG)[3]. Iterative methods generally have better scalability for parallel execution and are well suited for distributed environment and require far less memory. Unfortunately, these methods do not always converge within a reasonable amount of time frame and, for systems with large condition numbers, may not converge at all. The choice of an appropriate preconditioner often accelerates convergence. The linear preconditioned conjugate gradient (LPCG) algorithm represents one of the most commonly used iterative methods.

The formulation is proposed by combining domain decomposition with LPCG algorithm for solving large-scale nonlinear dynamic analysis problems on network cluster environment. The parallel implementation for LPCG algorithm is presented. That is, local interface formulation (LIF). Numerical example is implemented to validate as well as to evaluate the performance of the proposed formulation.

Implicit Finite Element Analysis

In the implicit finite element method for structural nonlinear analyses, application of standard finite element procedure yields a set of nonlinear equilibrium equations at $n+1$ and are of the form[4]

$$\mathbf{r}_{n+1} = \mathbf{f}_{n+1} - \mathbf{p}_{n+1} - \mathbf{M}\mathbf{a}_{n+1} \quad (1)$$

where \mathbf{a}_{n+1} denotes the nodal accelerations, \mathbf{M} is the mass matrix for the structure and \mathbf{f}_{n+1} defines the effective nodal forces due to applied force. \mathbf{p}_{n+1} denotes the forces exerted on element nodes consistent with the stresses. \mathbf{r}_{n+1} defines the unbalanced nodal forces.

For time integration of nonlinear system of equations given by Eq. (1), Newmark's constant average acceleration technique has been adopted with Newmark parameters β as 1/4 and γ as 1/2, which yields

$$\mathbf{r}_{n+1} = \mathbf{f}_{n+1}^d - \mathbf{p}_{n+1} - \frac{1}{\beta \Delta t^2} \mathbf{M} \mathbf{u}_{n+1} \quad (2)$$

where Δt defines the analysis time step and \mathbf{f}_{n+1}^d represents the effective nodal forces as modified by the terms from the Newmark integrator[5]. Application of a Newton Raphson procedure to eliminate the residual nodal forces at $n+1$ yields the following system of linear equations at Newton iteration i to advance the solution from n to $n+1$.

$$(\mathbf{K}_T + \frac{1}{\beta \Delta t^2} \mathbf{M}) \Delta \tilde{\mathbf{u}}_{n+1}^i = \mathbf{K}_T^d \Delta \tilde{\mathbf{u}}_{n+1}^i = \mathbf{f}_{n+1}^d - \mathbf{p}_{n+1}^{i-1} - \frac{1}{\beta \Delta t^2} \mathbf{M} \Delta \mathbf{u}_{n+1}^{i-1} \quad (3)$$

The right hand side represents the force imbalance (residual) from previous iteration, where $\Delta \mathbf{u}_{n+1}^{i-1}$ is the displacement increment for step $n+1$ as of the previous iteration. On the left hand side, \mathbf{K}_T^d defines the dynamic (consistent) tangent stiffness matrix for the current iteration, and $\Delta \tilde{\mathbf{u}}_{n+1}^i$ denotes the correction to the displacement increment.

Local Interface Formulation

In order to impart parallelism into the implicit finite element method for nonlinear dynamic analysis, domain decomposition technique is employed in the paper. A high quality domain partitioning algorithm balances the computational load among processors, while simultaneously minimising the communication costs between processors. The genetic algorithm[6] is employed for partitioning the finite element meshes in the paper. The finite element mesh is partitioned into desired number of submeshes and assigns one submesh to each processor.

The PCG algorithm is applied on the local submesh interface matrices and force vectors after eliminating the internal degrees of freedom. The global interface stiffness coefficient matrix is not formed. The preconditioners are constructed locally using the local schur complement matrix of the assigned submesh[7]. The interface stiffness coefficient matrix, \mathbf{K}_B and interface force vector, \mathbf{f}_B of submesh m can be written as

$$\mathbf{K}_b^m = \mathbf{K}_{bb}^m - \mathbf{K}_{bi}^m \mathbf{K}_{ii}^{m-1} \mathbf{K}_{ib}^m, \quad \mathbf{f}_b^m = \mathbf{f}_b^m - \mathbf{K}_{bi}^m \mathbf{K}_{ii}^{m-1} \mathbf{f}_i^m \quad (4)$$

The global interface stiffness matrix can be obtained by assembling all the submesh interface stiffness matrices. It can be written as

$$\mathbf{S} \mathbf{u}_b = \hat{\mathbf{f}}_b \quad (5)$$

where $\mathbf{S} = \sum_{m=1}^n \mathbf{B}_b^{m'} \mathbf{s}^m \mathbf{B}_b^m$, $\mathbf{S}^m = \mathbf{K}_{bb}^m - \mathbf{K}_{bi}^m \mathbf{K}_{ii}^{m-1} \mathbf{K}_{ib}^m$, $\hat{\mathbf{f}}_b = \mathbf{f}_b - \sum_{m=1}^n \mathbf{B}_b^{m'} \mathbf{K}_{bi}^m \mathbf{K}_{ii}^{m-1} \mathbf{f}_i^m$.

Matrix \mathbf{S} is the schur complement of \mathbf{K}_{bb} in \mathbf{K} and each submatrix \mathbf{S}^m corresponds to a local static condensation operator or a local schur complement. The incomplete factorization preconditioners are constructed using the local schur complement matrix \mathbf{S}^m . The parallel PCG implementation in the local interface formulation is presented in Table 1.

Table 1 Parallel PCG algorithm for LIF

1 construct preconditioning matrix \mathbf{P} using local schur complement matrix	11 if($i=0$) $\bar{\mathbf{w}}_i = \bar{\mathbf{z}}_i$
2 initialise the vector $\mathbf{d}_0 = \mathbf{d}^0$, $\mathbf{r}_0 = \mathbf{f}$	12 else $\bar{\mathbf{w}}_i = \bar{\mathbf{z}}_i + (\alpha_i / \alpha_{i-1}) \bar{\mathbf{z}}_{i-1}$ /SAXPYoperation/
3 send \mathbf{r}_0^b , receive \mathbf{r}_0^{eb} from neighbouring processors	13 $\mathbf{z}_i = \mathbf{S} \bar{\mathbf{w}}_i$ /matrix vector operation /
4 form $\bar{\mathbf{r}}_0 = \mathbf{r}_0 + \mathbf{r}_0^{eb}$ /vector global assembly /	14 $\beta_i = (\bar{\mathbf{w}}_i)^T \mathbf{z}_i$ / vector operation /
5 loop over iterations for $i=0,1,2,\dots$	15 reduce $\beta_i = \sum \beta_i$ / communication /
6 compute $\mathbf{z}_i = \mathbf{P}^{-1} \bar{\mathbf{r}}_i$ /preconditioning step/	16 send \mathbf{z}_i^b , receive \mathbf{z}_i^{eb} from neighbouring processors
7 compute $\alpha_i = (\bar{\mathbf{r}}_i)^T \mathbf{z}_i$ / vector operation /	17 compute $\bar{\mathbf{z}}_i = \mathbf{z}_i + \mathbf{z}_i^{eb}$ / vector global assembly/
8 reduce α_i , $\alpha_i = \sum \alpha_i$ /communication /	18 $\bar{\mathbf{d}}_{i+1} = \bar{\mathbf{d}}_i + (\alpha_i / \beta_i) \bar{\mathbf{w}}_i$ / vector operation /
9 send \mathbf{z}_i^b , receive \mathbf{z}_i^{eb} from neighbouring processors	19 $\bar{\mathbf{r}}_{i+1} = \bar{\mathbf{r}}_i - (\alpha_i / \beta_i) \bar{\mathbf{z}}_i$ / vector operation /
10 compute $\bar{\mathbf{z}}_i = \mathbf{z}_i + \mathbf{z}_i^{eb}$ / vector global assembly /	20 if(α_i / α_0) < ε then exit
	21 End for

Numerical Example

The developed parallel computing for structural nonlinear finite element dynamic analysis is performed on DELL workstation cluster. It is a cluster with 8 processors arranged in 4 dual-processor nodes with 2.4GHz Intel Xeon chips (512KB cache) and 1.0GB of memory per node. These nodes are connected with a 100.0Mbps Ethernet interconnect. The MPI communication libraries have used to manage the message passing[8].

One quarter of the cylindrical shell shown in Fig.1 is considered as a numerical example. Its length is 1000mm and its two sides are fixed boundary. Uniform load $p=2.0\text{KN/m}^2$, the way of imposed load is given in Fig.1. The shell thickness is 2mm, the material properties Young's modulus $E=2.06 \times 10^5 \text{Mpa}$, Poisson's ratio $\nu=0.3$, Yield stress $f=235\text{Mpa}$, Mass density $\rho=7.8 \times 10^3 \text{Kg/m}^3$. Geometric nonlinearity is considered.

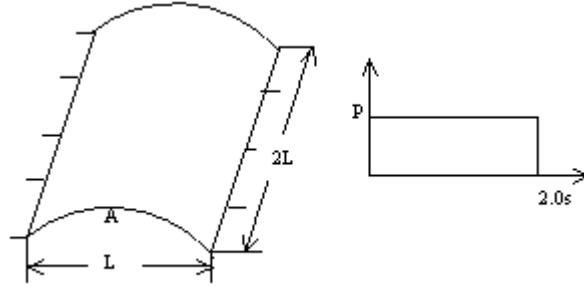


Fig.1 Description of the one quarter of the cylindrical shell

The shell is discretised into a 12×24 structured mesh. The 8-node shell element is employed. The stability and accuracy are examined by the time step size 0.5×10^{-3} . The A nodal displacement time history response computed by the Newmark method and the proposed LIF are shown in Fig.2. It can be observed that the proposed parallel algorithm yields acceptable results and the solution is in good agreement with the conventional Newmark's solution. This illustrates that the proposed algorithm possess highly the stability and accuracy.

To evaluate the performance of the proposed formulation, the problem size(i.e., degree of freedom) is 17322 dof. is solved using 1, 2, 4, 6, 8 processors respectively. The speedups are shown in Fig.3. It can be observed that the speedup of the proposed parallel algorithm is larger than the parallel direct solvers. The proposed parallel algorithm is superior in performance when compared to the conventional domain decomposition algorithm with a direct solver. The proposed parallel formulation exhibits superior performance.

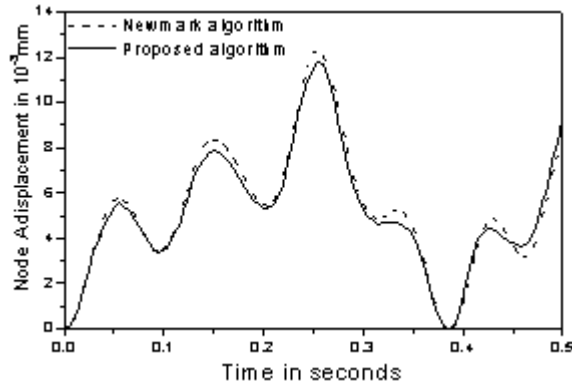


Fig.2 Displacement time history response of node A

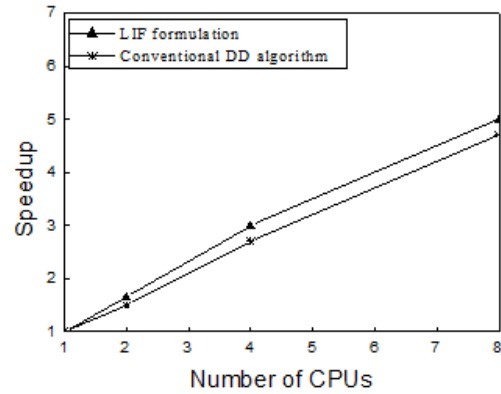


Fig.3 Parallel speedup

Conclusions

In this paper, parallel algorithm for implicit nonlinear dynamic analysis employing finite element method is studied. The Global Interface Formulation combining domain decomposition based finite element method with linear preconditioned conjugate gradient solvers are proposed. Newmark- β average acceleration technique is employed for time integration. Numerical studies indicate that the proposed parallel formulation is good in performance.

Acknowledgments

The work was financially supported by the National Natural Science Foundation of China(Grand No.51378124).

References

- [1] D. M. Day, M. K. Bhardwaj. Mechanism free domain decomposition. Computer Methods in Applied Mechanics and Engineering. Vol. 192(2003),p.763
- [2] A. Rama Mohan Rao, B Dattaguru. Parallel finite element method for nonlinear dynamic analysis with optimized communications using sparse PCG solver//Jenkins L. and Ivengar. Proceedings of the International Conference on High Performance Computing. USA: IEEE Press, (2002), p.157
- [3] J. F. Hajjar, J. F. Abel. Parallel processing for transient nonlinear structural dynamics of three dimensional framed structures using domain decomposition. Computers and Structures. Vol. 30(1988),p.1237
- [4] A. Rama Mohan Rao. MPI-based parallel finite element approaches for implicit nonlinear dynamic analysis employing sparse PCG solvers. Advances in Engineering Software. Vol.36(2005), p.181
- [5] N. M. Newmark. A method of computation for structural dynamics. Journal of the Engineering Mechanics Division.vol.32(1959),p.67
- [6] A. Rama Mohan Rao, Appa Rao TVSR and B. Dattaguru. Generating optimised partitions for parallel finite element computations employing float-encoded genetic algorithms. Computer Modeling in Engineering and Science. Vol.5(2007),p.213
- [7] N. Bitoulas, M. Papadrakakis. An optimised computer implementation of incomplete Cholesky factorization. Computing Systems in Engineering. Vol.5(2006),p.265
- [8] Chaojiang Fu. The research on parallel computation of finite element structural analysis based on MPI cluster. Shanghai: Shanghai University, (2006), (in Chinese)