# SHARED MEMORY PARALLEL COMPUTING PROCEDURES FOR NONLINEAR DYNAMIC ANALYSIS OF SUPER HIGH-RISE BUILDINGS

Shaojun FU, Zheng HE*, Zhiquan LI, Xiang TU, Qian TAO
Department of Civil Engineering, Dalian University of Technology, Dalian 116024, China
*E-mail: hezheng1971@126.com (Z. He)

## ABSTRACT

The proposed parallel state transformation procedures (PSTP) of fiber beam-column elements and multi-layered shell elements, combined with the parallel factorization of Jacobian (PF), are incorporated into a finite element program. In PSTP, elements are classified into different levels of workload prior to state determination in order to balance workload among different threads. In PF, the multi-threaded version of OpenBLAS is adopted to compute super-nodes. A case study on four super high-rise buildings, i.e. S1~S4, has demonstrated that the combination of PSTP and PF does not have any observable influence on computational accuracy. As number of elements and DOFs increases, the ratio of time consumed in the formation of the Jacobian to that consumed in the solution of algebraic equations tends to decrease. The introduction of parallel solver can yield a substantial reduction in computational cost. Combination of PSTP and PF can give rise to a further significant reduction. The acceleration ratios associated with PSTP do not exhibit a significant decrease as PGA level increases. Even PGA level is equal to 2.0g, PSTP still can result in acceleration ratios of 2.56 and 1.92 for S1 and S4, respectively. It is verified that it is more effective to accelerate analysis by reducing the time spent in solving algebraic equations rather than reducing that spent in the formation of the Jacobian for super high-rise buildings.

## KEYWORDS

State transformation procedures, super-nodal Cholesky factorization, super high-rise building, parallel computing, OpenBLAS, nonlinear dynamic analysis, computational efficiency.

## INTRODUCTION

Nonlinear dynamic analysis is a powerful tool for the assessment of seismic performance of structures under strong dynamic excitations, e.g. earthquakes. The standard simulation approach is to discretize the time-continuous governing equations by time integration methods then solve them via iterative algorithm (Bathe 1996). Since analytical time step or time interval, Δ*t*, is generally small and iterations may be required within a step, the formation of real-time tangent stiffness matrix (also called the Jacobian) and solution process of system equations usually causes expensive computational cost. For large structures with a huge number of elements and degrees of freedom (DOFs), e.g. super high-rise buildings, such cost might be too high to be tolerable for design purposes although computer hardware is updated contemporarily.

Some parallel computing techniques were adopted to reduce computational cost from two aspects. Parallel direct (Cho and Hall 2012) or iterative solvers (Balay et al. 2014) are applied to accelerate the solution of system equations while the parallelized state determination of elements (Liu et al. 2007) is used to accelerate the formation of the Jacobian. Computational efficiency and convergence of iterative solvers are highly dependent on a pre-conditioner selected which is also closely related with the characteristics of the Jacobian. For super high-rise structures having various structural systems, direct solvers are more attractive because of excellent stability. Some parallel direct solver packages have been developed, e.g. MUMPS (Amestoy et al. 2001), SuperLU (Li 2005), and PaStiX (Hénon et al. 2002). The parallelized state determination of elements can be easily implemented (Yang et al. 2014). However, extra considerations should be taken in the treatment of unbalanced computational workload among different types of elements. Even within a certain type of elements, unbalance in workload can also exist between processed elements and unprocessed elements, if the state transformation procedures (STP) (He et al. 2015) are applied. Other than the above-mentioned two approaches, the domain decomposition methods (DDMs) (Smith et al. 1996; Sotelino 2003) directly divide the solution domain of concern into several subdomains and then assign one thread to each subdomain. In this way, the time consumed both in the process of the formation of the Jacobian and the solution of system equations can be saved simultaneously (Yang and Hsieh 2002; Yang et al. 2012). Nevertheless, sometimes it is difficult to balance the workload among subdomains.

Performance of all the parallel computing techniques are highly related to the configuration of computer hardware, e.g. multi-core processors, multiple local computers (McKenna and Fenves 2007), high performance graphics processing units (GPU) (Papadrakakis et al. 2011), etc. Inter-process communication between threads may be an obstacle in parallel computing, especially on multiple local computers and GPUs. Data transmission between threads can be avoided in shared memory multi-processing on multi-core computers, which is not difficult for computerized implementation. Moreover, shared memory multi-processing is based on either open Multi-Processing (OpenMP) (Amritkar et al. 2012) programming language or POSIX Threads (Pthreads) (IEEE Std 1003.1c-1995) programming language. These accessible programming languages can be embedded in source codes and have been applied in many areas (Brown and Sharapov 2007).

This paper aims at developing some optimized shared memory computing procedures that to be implemented on personal computers for nonlinear dynamic time-history analysis for super high-rise buildings. An optimized STP-based strategy for the parallel state determination of elements is proposed to accelerate the formation process of the Jacobian, while the super-nodal Cholesky solver from CHOLMOD (Chen et al. 2008) is accelerated with multi-threaded OpenBLAS (Zhang et al. 2012) in order to solve the linear algebraic equations more efficiently. The characteristics of nonlinear dynamic analysis for super high-rise buildings will be discussed prior to the introduction of these procedures. The effects of some factors, e.g. peak ground acceleration (PGA) level, the number of DOFs and the number of elements, on computational efficiency are investigated.

## PARALLEL COMPUTING STRATEGIES

### *Basic Procedures*

The dynamic equilibrium of a nonlinear system can be expressed as the followings:

$$\boldsymbol{p}(t)-\boldsymbol{f}_{\mathrm{S}}\left(\boldsymbol{x}(t)\right)-\boldsymbol{f}_{\mathrm{D}}\left(\boldsymbol{x}'(t)\right)-\boldsymbol{f}_{\mathrm{I}}\left(\boldsymbol{x}''(t)\right)=0 \tag{1}$$

Where, $\boldsymbol{x}$, $\boldsymbol{x}'$ and $\boldsymbol{x}''$ are the vectors of displacements, velocities and accelerations of a system, respectively; $\boldsymbol{f}_{\mathrm{S}}$, $\boldsymbol{f}_{\mathrm{D}}$ and $\boldsymbol{f}_{\mathrm{I}}$ are the vectors of inertia forces, damping force and resisting forces, respectively; $\boldsymbol{p}$ is the vector of external excitations. Discretizing Eq. (2) with the $i$th analytical time step, $\Delta t$, will give the following incremental form,

$$\Delta\boldsymbol{p}_i-\Delta\boldsymbol{f}_{\mathrm{S}i}\left(\Delta\boldsymbol{x}_i\right)-\Delta\boldsymbol{f}_{\mathrm{D}i}\left(\Delta\boldsymbol{x}_i'\right)-\Delta\boldsymbol{f}_{\mathrm{I}i}\left(\Delta\boldsymbol{x}_i''\right)=0 \tag{2}$$

Eq. (2) can be simplified as the following general form by using any step-by-step time integration method,

$$F\left(\Delta\boldsymbol{x}_i\right)=0 \tag{3}$$

If the Newton-Raphson algorithm is applied to solve Eq. (4) and to obtain $\Delta\boldsymbol{x}_i$, the iterative procedures within an analytical time step can be expressed as:

$$\text{Obtain } \boldsymbol{s}_i^k \text{ from : } F'\left(\Delta\boldsymbol{x}_i^k\right)\cdot\boldsymbol{s}_i^k=-F\left(\Delta\boldsymbol{x}_i^k\right) \tag{4}$$

$$\text{Set } \Delta\boldsymbol{x}_i^{k+1}=\Delta\boldsymbol{x}_i^k+\boldsymbol{s}_i^k \tag{5}$$

Where, $F'(\Delta\boldsymbol{x}_i^k)$ is the Jacobian at the $k$th iteration during the $i$th analytical step, which can be computed by the time integration method selected; $\boldsymbol{s}_i^k$ is the increment of displacement vector after the $k$th iteration. For most time integration methods, e.g. the Newmark-beta method, $F'(\Delta\boldsymbol{x}_i^k)$ is a linear combination of tangent stiffness matrix, $\boldsymbol{K}$, damping matrix, $\boldsymbol{C}$ and mass matrix, $\boldsymbol{M}$, i.e.

$$F'\left(\Delta\boldsymbol{x}_i^k\right)=a_1\boldsymbol{K}_i^k+a_2\boldsymbol{C}_i^k+a_3\boldsymbol{M}_i^k \tag{6}$$

Where, constants, $a_1$, $a_2$ and $a_3$ are dependent on analytical step, $\Delta t$, and time integration method selected. A complete nonlinear dynamic time history analysis usually involves thousands of analytical steps and even more iterations, as illustrated in Figure 1. Thus, it is necessary to reduce computing time in the two time-consuming steps (see Figure 1) via some proper approaches.

### *Parallel Computing Strategy for STP*

Fiber beam-column element (Spacone et al. 1996) and multi-layer shell element (Lu et al. 2015) are capable of capturing dynamic time-history nonlinear responses of structures excited by strong earthquakes. These elements can bridge microscopic material-level properties (e.g. stresses and strains) and macroscopic structure-level responses (e.g. deflection, story drift, base shear, overturning moment etc.) with desirable accuracy. However, computational accuracy is achieved at the cost of significant computation workload in material-level state determination. It is prove to an efficient way to reduce the workload by using the state transformation procedures (STP) developed by He et al (2015), as illustrated in Figure 2. The STP is fully based on the understanding of sparse nonlinearity distribution in structures when they are undergoing strong earthquakes. It assumes that sections at integration points in fiber beam-column elements and multi-layer shell elements have three states, i.e. initial state, elastic state and nonlinear state. The state determination of initial state and nonlinear state at integration points is carried out in a traditional way, i.e. from material level. Once initial state is determined, the subsequent determination of sections can override material-level computation in elastic state. All sections in this state are simply simulated by predetermined linear resisting force models until nonlinearity occurs. Once nonlinearity develops, the linear models will be replaced with those determined by corresponding nonlinear sections. Initial tensile cracking, nonlinear compression or yielding of steel can be all regarded as the critical limit of elastic cross sections. Beyond the limit, the sections will not behave linear elastic any more. Such nonlinearities developed in a cross section always start from some specific locations, e.g. corner point in a rectangular section or the outmost layer in a shell element. These locations can act as monitoring points for the determination of critical elastic limit.

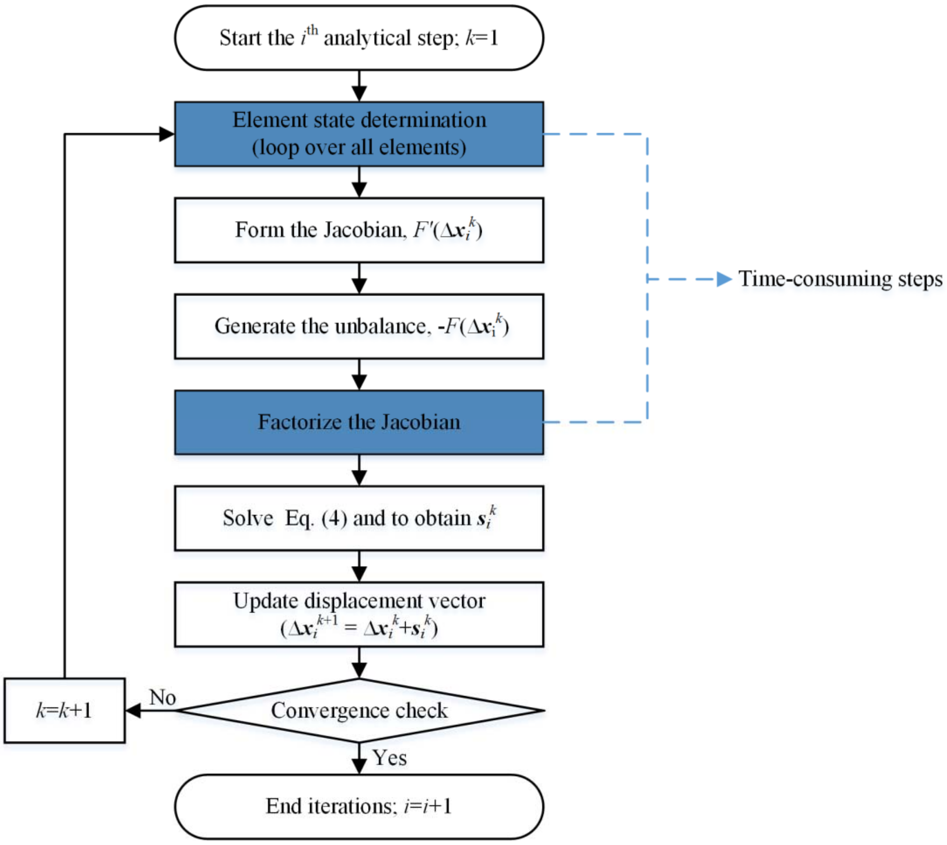

Start the $i^{th}$ analytical step; $k$=1
Element state determination (loop over all elements)
Form the Jacobian, $F'(\Delta x_i^k)$
Generate the unbalance, $-F(\Delta x_i^k)$
Factorize the Jacobian
Solve Eq. (4) and to obtain $s_i^k$
Update displacement vector ($\Delta x_i^{k+1} = \Delta x_i^k + s_i^k$)
Convergence check
No
$k$=$k$+1
Yes
End iterations; $i$=$i$+1
Time-consuming steps

Figure 1 Iteration within an analytical step

The above-mentioned STP can be combined with parallel processing technique (PSTP) to achieve higher acceleration. However, thread allocation should be optimized to avoid thread wait. The unbalance in computational workload arises from two aspects when the STP to accelerate the state determination process of elements. One is from different time consumed in the determination of fiber beam-column element and multi-layered shell elements. The other is from different number of nonlinear sections among elements. Thus, the so-called level-to-level parallel processing scheme is proposed. That is, elements are classified into different levels of workload, as illustrated in Figure 3, prior to state determination according to the number of nonlinear sections at integration points develop within an element. $n$, $n_f$ and $n_s$ shown in Figure 3 denote the number of super-modes (to be introduced later), nonlinear sections in a fiber beam-column element and a multi-layered shell element. For instance, a fiber beam-column element with three nonlinear sections can be classified as Level 3 while Level 0 indicates zero nonlinear sections within an element. Parallel computing is carried out by using the dynamic schedule of OpenMP. In this way, workload among different threads is basically balanced.

***Parallel Computing Strategy for LL$^T$ Factorization***

In most cases, the Jacobian is a sparse and symmetric positive definite matrix. Thus, the sparse Cholesky factorization (the $\boldsymbol{LL}^{\mathrm{T}}$ factorization) method can be applied to determine the increment of displacement vector after the $k$th iteration, $\boldsymbol{s}_i^k$, by the following equations:

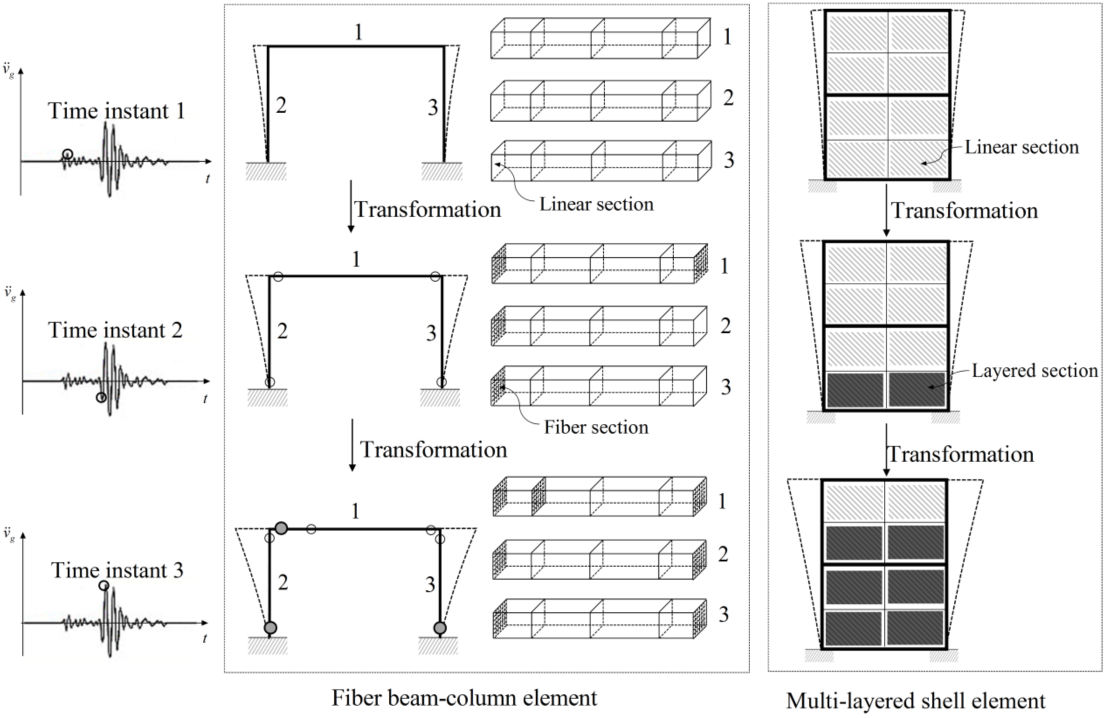


Figure 2 Concept of element state transformation (STP)

$$\text{Obtain } \boldsymbol{L}_i^k \text{ from: } \quad F'\left(\Delta \boldsymbol{x}_i^k\right) = \boldsymbol{L}_i^k \boldsymbol{L}_i^{k\mathrm{T}} \tag{7}$$

$$\text{Obtain } \boldsymbol{y}_i^k \text{ from: } \quad \boldsymbol{L}_i^k \boldsymbol{y}_i^k = -F\left(\Delta \boldsymbol{x}_i^k\right) \tag{8}$$

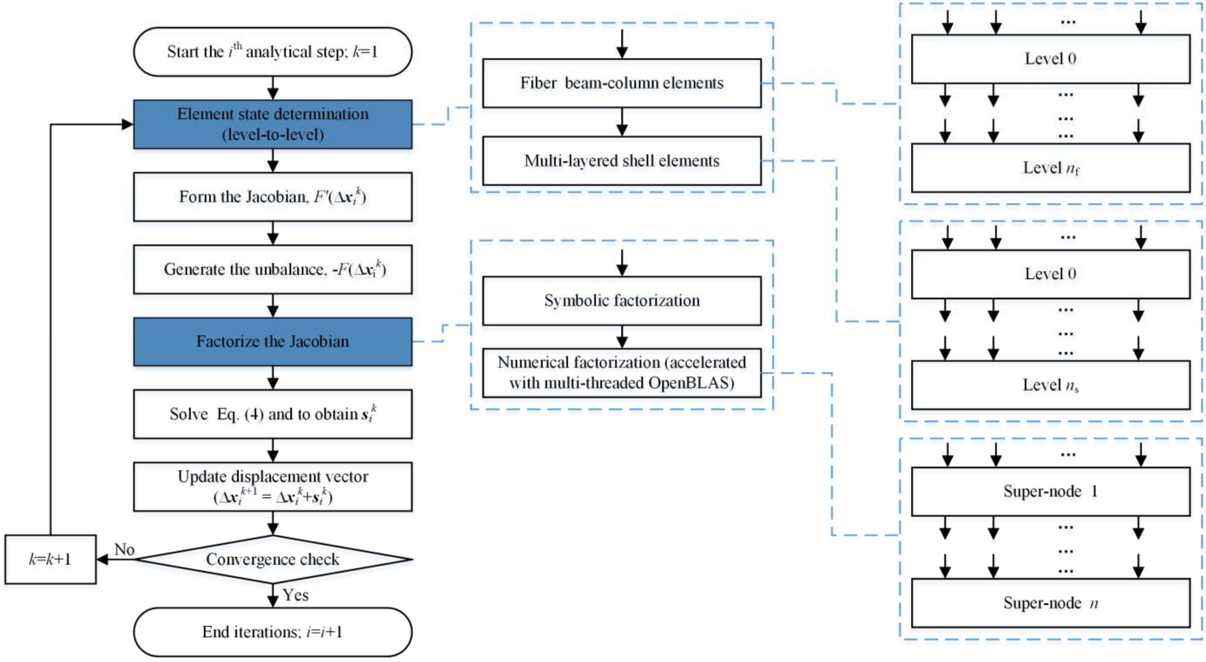
$$\text{Obtain } \boldsymbol{s}_i^k \text{ from: } \quad \boldsymbol{L}_i^{k\mathrm{T}} \boldsymbol{s}_i^k = \boldsymbol{y}_i^k \tag{9}$$

The super-nodal method (Rotkin and Toledo 2004) can be utilized to accelerate the factorization in Eq. (7), in which a set of super-nodes are computed through exploiting dense matrix kernels, i.e. the BLAS (Lawson et al. 1979). Some efficient BLAS libraries, e.g. Intel® Math Kernel Library (Intel® MKL) (Intel Corporation 2015), OpenBLAS and cuBLAS (NVIDIA Corporation 2015) are available. The former two are fully optimized for multiprocessing in Intel® CPUs. The last developed based on CUDA (NVIDIA Corporation 2015) is the GPU-accelerated version of complete standard BLAS library. With proper configuration of computer hardware, the three BLAS libraries have better performance than the standard serial BLAS. CHOLMOD is chosen herein to perform the super-nodal $\boldsymbol{LL}^{\mathrm{T}}$ factorization and the multi-threaded version of OpenBLAS is adopted to compute super-nodes. Thus, such factorization process can be parallelized (PF). As depicted by Figures 3, the step of numerical factorization is accelerated with multi-threaded OpenBLAS.

## COMPUTERIZED IMPLEMENTATION

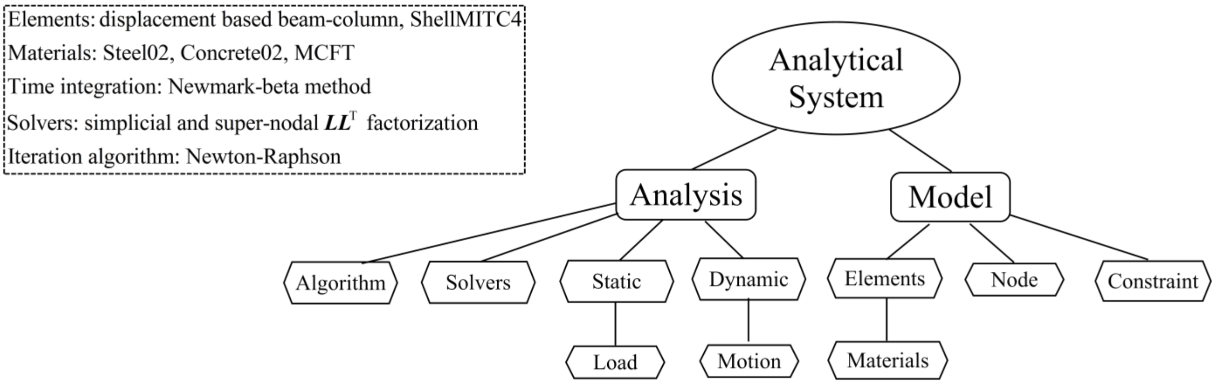
As one of open-source finite element programs, OpenSees (Mazzoni et al 2015) has been shown to be a powerful tool for performing nonlinear dynamic analysis of buildings and bridges and has become more and more popular in the field of earthquake engineering. However, some classes associated with elements in OpenSees have static member variables which are not compatible with shared memory parallel computing. Due to some reason, part of code in OpenSees has not been released so far, making an obstacle to realize PSTP. Thus, the above features are implemented into a new parallel finite element program using C++ program language. Some open-source codes available are incorporated into the program. The uniaxial materials of concrete and steel, i.e. Concrete02 (Hisham and Yassin 1994) and Steel02 (Filippou et al. 1983), as well as shell element ShellMITC4 (Dvorkin and Bathe 1984), are adopted from OpenSees (Mazzoni et al 2015). Eigen (Benoit and Guennebaud 2015) is chosen to handle basic matrix manipulations. Both the simplicial method and the super-nodal method in CHOLMOD are selected to perform sparse Cholesky factorization and to solve algebraic equations. Multi-threaded OpenBLAS is used to accelerate super-nodal sparse Cholesky factorization in parallel. In addition, the well-known MCFT for 2D concrete material developed by Vecchio and Collins (1986), and Vecchio (1999) is combined with the ShellMITC4 to develop a nonlinear multi-layered shell element (Li 2015).

The framework and some basic modules of the program are shown Figure 4. The program is compiled and fully optimized with open-source GNU compilers. In addition, the Newmark-beta time integration and the Newton-Raphson iteration algorithm are applied.



Start the $i^{th}$ analytical step; $k$=1
Element state determination (level-to-level)
Form the Jacobian, $F'(\Delta x_i^k)$
Generate the unbalance, $-F(\Delta x_i^k)$
Factorize the Jacobian
Solve Eq. (4) and to obtain $s_i^k$
Update displacement vector ($\Delta x_i^{k+1} = \Delta x_i^k + s_i^k$)
Convergence check
No
$k$=$k$+1
Yes
End iterations; $i$=$i$+1
Fiber beam-column elements
Multi-layered shell elements
Symbolic factorization
Numerical factorization (accelerated with multi-threaded OpenBLAS)
Level 0
Level $n_f$
Level 0
Level $n_s$
Super-node 1
Super-node $n$

Figure 3 Basic procedures for PSTP and PF during an analytical step



Elements: displacement based beam-column, ShellMITC4
Materials: Steel02, Concrete02, MCFT
Time integration: Newmark-beta method
Solvers: simplicial and super-nodal $LL^T$ factorization
Iteration algorithm: Newton-Raphson
Analytical System
Analysis
Model
Algorithm
Solvers
Static
Dynamic
Elements
Node
Constraint
Load
Motion
Materials

Figure 4 Basic modules of the program

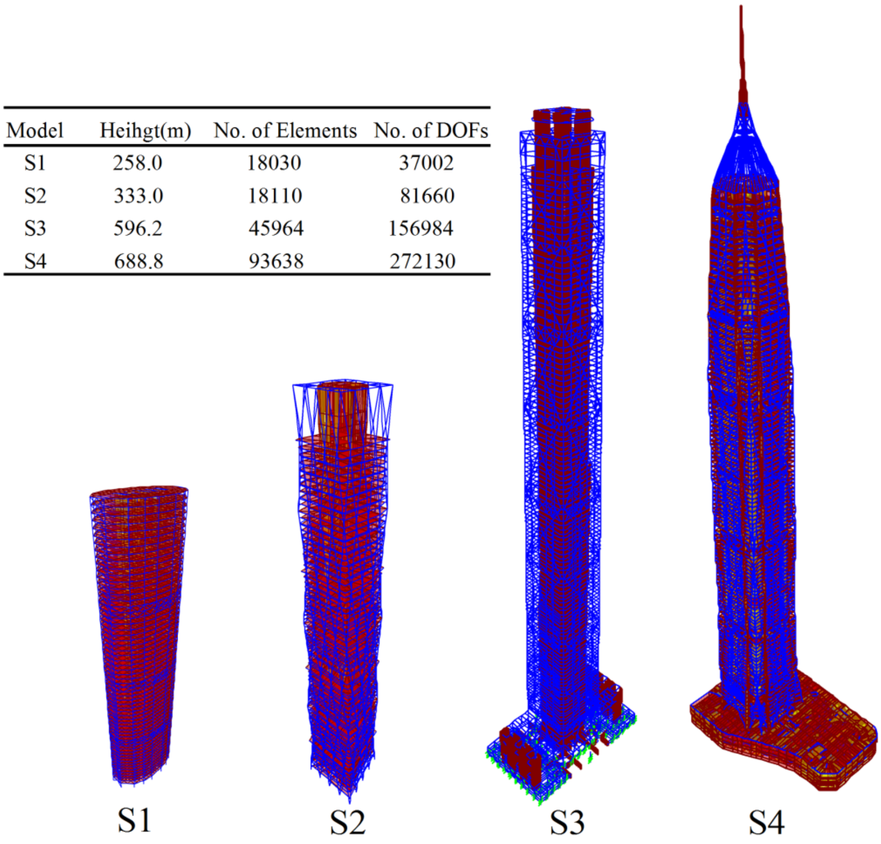
## CASE STUDY

Four super high-rise buildings, S1~S4 as illustrated in Figure 5, are selected to assess the performance of PSTP and PF. Both PSTP and PF have four threads. Structural walls, floors and ramps in S1~S4, and mega columns in S4 are modeled by multi-layered shells (red areas in Figure 5), while the rest components in S1~S4 are modeled by fiber beam-column elements (blue lines in Figure 5). A Dell Optiplex 9010 personal computer with a quad-core Inter I7-3770, 32GB main memory, and Linux Ubuntu 14.04 operating system, is used in the study. Three strong ground motions, i.e. Northridge earthquake at Station MUL009, Duzce earthquake at Station BOL000 and Hector earthquake at Station HEC000 (PEER 2015), with two horizontal components are selected as external excitations.

### *Computational Accuracy*

Figures 6 and 7 illustrate some results of S1 from nonlinear dynamic time history analysis by using the program with PSTP and PF, and OpenSees (Mazzoni et al 2015) that combines the PlaneStressUserMaterial with ShellMITC4 (Dvorkin and Bathe 1984; Lu et al. 2015). The PGA levels of the principal components of three are scaled to 1.0g. Figures 6 and 7 show the top displacement time history response and the distribution of the maximum inter-story drift ratios along structural height, respectively. The results between the program and OpenSees agree very well with each other.

***Computational Efficiency (PGA=1.0g)***



| Model | Heihgt(m) | No. of Elements | No. of DOFs |
|---|---|---|---|
| S1 | 258.0 | 18030 | 37002 |
| S2 | 333.0 | 18110 | 81660 |
| S3 | 596.2 | 45964 | 156984 |
| S4 | 688.8 | 93638 | 272130 |

Figure 5 Numerical models

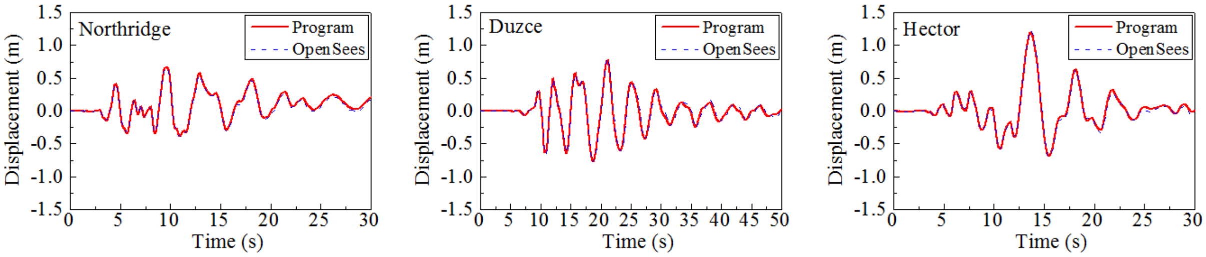


Figure 6 Top displacement time histories of Structure S1

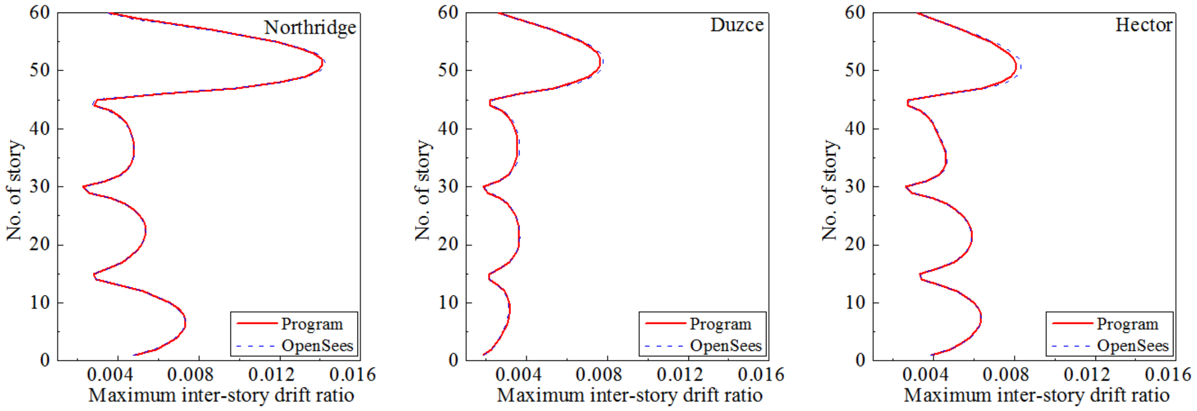
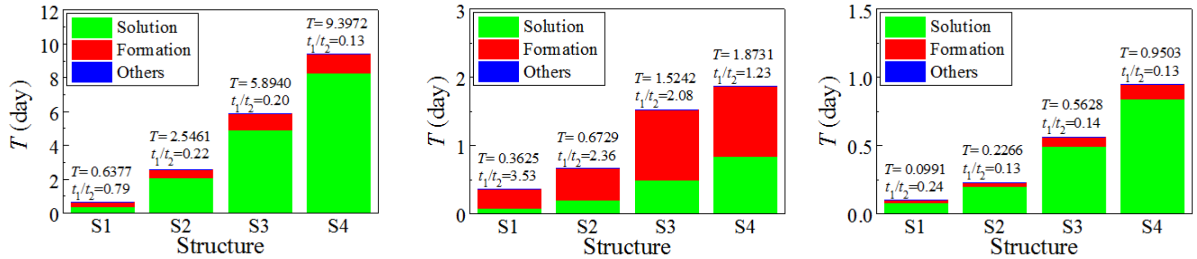


Figure 7 Distribution of maximum inter-story drift ratios of S1

The computational efficiency of PF and PSTP is compared with traditional serial computing on four super high-rise buildings under Northridge earthquake with a scaled PGA level equal to 1.0g in principal component. Time consumed in the formation of the Jacobian, i.e. $t_1$, including mainly element state determination, and the solution of algebraic equations, $t_2$, are recorded during time history analyses, as illustrated in Figure 8 in which $T$ represents total time consumed. As discussed previously, most of time is consumed in the formation of the Jacobian and the solution of algebraic equations in any case. As number of elements and number of DOFs increases, $T$, or total workload obviously increases. Because $t_1$ increases approximately linearly with the number of elements while $t_2$ increases nonlinearly with the DOFs. That is, difficulties in solving algebraic equations using direct solvers rise super linearly, the ratio of $t_1/t_2$ tends to decrease as structural height increases. If time consumed in other aspects is ignored, i.e. $T$=$t_1$+$t_2$, the introduction of parallel solver can reduce $t_2$ from 3.91d to 0.40d on average. Combination of PSTP and PF can give rise to further significant reduction in computational cost, more than 10 times faster than traditional serial computing on average. Every element is much less computationally expensive due to the combination of the STP and parallel computing technique. However, the workload of solving algebraic equations, $t_2$, becomes dominant again with dual application of PF and PSTP although total workload has been dramatically reduced.



(a) Serial computing

(b) PF

(c) PF & PSTP

Figure 8 Time consumed using three computing strategies (Northridge earthquake)

### *Computational Efficiency (PGA=0.2g~2.0g)*

The efficiency of STP is highly related with PGA level. To fully demonstrate the efficiency of PSTP, incremental dynamic analyses are performed on S1 and S4. Three other earthquakes, i.e. an artificial earthquake wave (AEW), Tabas earthquake and Landers earthquake are selected, with the PGA level of principal component varies from 0.20g to 2.0g. Although significant difference is seen from Figure 9 between S1 and S4, the acceleration ratios caused by PSTP, $S$=$T_{\mathrm{PF}}$/$T_{\mathrm{PF\&PSTP}}$, decrease within a relatively small range as PGA increase from 0.2g to 2.0g. The PSTP still can result in averaged ratios of 2.56 and 1.92 for S1 and S4, respectively. The ratios of S4 are generally smaller than those of S1, because $t_1$ of S4 occupies less portion than that of S1, as can be observed from Figure 8(b). Thus, for large structures like S4, it is more effective to accelerate analysis by reducing the time spent in solving algebraic equations rather than reducing that spent in the formation of the Jacobian.

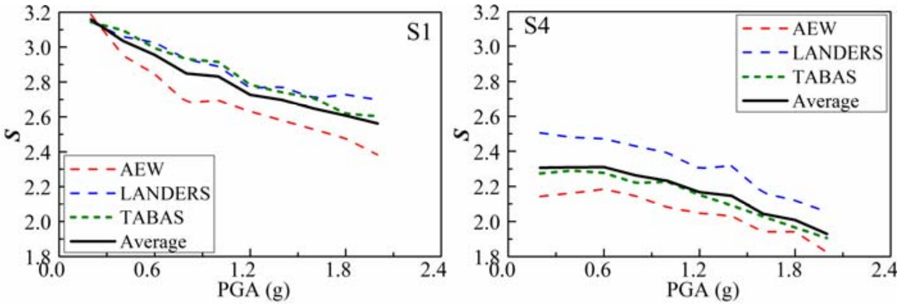


Figure 9 Acceleration ratios (PGA=0.2g~2.0g)

## CONCLUSIONS

The proposed parallel state transformation procedures (PSTP) of elements combined with the parallel factorization of the Jacobian (PF) using multi-threaded OpenBLAS are incorporated into a finite element program. From the case study on the nonlinear dynamic time history analyses of four super high-rise buildings, i.e. S1~S4, the combination of PSTP and PF does not have any observable influence on computational accuracy. Most of time is consumed in the formation of the Jacobian and the solution of algebraic equations in all cases. As number of elements and DOFs increases, the ratio of time consumed in the formation of the Jacobian to that consumed in the solution of algebraic equations tends to decrease. Combination of PSTP and PF can give rise to significant reduction in computational cost, more than 10 times faster than traditional serial computing on average. The workload of solving algebraic equations will become dominant again although total workload is reduced dramatically with the dual application of PSTP and PF. The acceleration ratios associated with PSTP do not exhibit a significant decrease as PGA level increases from 0.2g to 2.0g. Even PGA level is equal to 2.0g, PSTP still can result in averaged ratios of 2.56 and 1.92 for S1 and S4, respectively. For super high-rise buildings, it is more effective to accelerate analysis by reducing the time spent in solving algebraic equations rather than reducing that spent in the formation of the Jacobian.

## ACKNOWLEDGEMENTS

This research was financially supported by the National Natural Science Foundation of China (Grant No. 91315301 and 51261120376).

## REFERENCES

Amestoy, P.R., Duff, I.S., L'Excellent, J.Y., Koster, J. (2001) "MUMPS: a general purpose distributed memory sparse solver." *In Applied Parallel Computing: New Paradigms for HPC in Industry and Academia Lectures Notes in Computer Science*, Bergen, Norway.

Amritkar, A., Tafti, D., Liu, R., Kufrin, R., Chapman, B. (2012). "OpenMP parallelism for fluid and fluid-particulate systems." *Parallel Computing*, 38(9), 501-517.

Bathe, K. J. (1996). *Finite element procedures*, Prentice-Hall, Upper Saddle River, N.J., USA.

Balay, S. et al. (2014). "PETSc Users Manual". *Technical Report ANL-95/11 – Revision 3.5*, Argonne National Laboratory.

Benoit, J., Guennebaud, G. http://eigen.tuxfamily.org (accessed 7.07.2015).

Brown, R., Sharapov, I. (2007). "High-scalability parallelization of a molecular modeling application: performance and productivity comparison between OpenMP and MPI implementations." *International Journal of Parallel Programming*, 35(5), 441–458

Chen, Y., Davis, T. A., Hager, W. W., Rajamanickam, S. (2008). "Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate," *ACM Transactions on Mathematical Software*, 35(3) 22:1–22:14

Cho, I. H., and Hall, J. F. (2012). "A parallelized implicit nonlinear FEA program for real scale RC structures under cyclic loading." *Journal of Computing in Civil Engineering*, 26(3), 356–365.

David, R. B. *Programming with POSIX Threads*. Addison-Wesley Professional, New York, NJ, USA.

Dvorkin, E. N., Bathe, K.J. (1984). "A continuum mechanics based four node shell element for general nonlinear analysis." *Engineering Computing*, 1, 77-88.

Filippou, F. C., Popov, E. P., Bertero, V. V. (1983). "Effects of bond deterioration on hysteretic behavior of reinforced concrete joints." *EERC Report 83-19*, Earthquake Engineering Research Center, University of California, Berkeley, CA, USA.

He, Z., Fu S.J., Ou, J. P. (2015) “State transformation procedures for fiber beam-column element in inelastic dynamic time history analysis for moment-resisting frames.” *Journal of Computing in Civil Engineering*, under review.

Hénon, P., Ramet, P., Roman, J. (2002) “PaStiX: a high-performance parallel direct solver for sparse symmetric definite systems.” *Parallel Computing*, 28(2), 301-321.

Hisham, M., and Yassin, M. (1994). “Nonlinear analysis of prestressed concrete structures under monotonic and cycling loads.” PhD dissertation, University of California, Berkeley, CA, USA.

Intel Math Kernel Library. http://software.intel.com/en-us/intel-mkl (accessed 7.07.2015).

Kurc, O. (2010). “Workload Distribution Framework for the Parallel Solution of Large Structural Models on Heterogeneous PC Clusters.” *Journal of Computing in Civil Engineering*, 24(2), 151-160.

Kurc, O., and Will, K. (2007). “An iterative parallel workload balancing framework for direct condensation of substructures.” *Computer Methods in Applied Mechanics and Engineering*, 196(17-20), 2084–2096.

Li, X.S. (2005). “An overview of SuperLU: algorithms, implementation, and user interface.” *Journal of ACM Transactions on Mathematical Software*, 31(3), 302-325.

Li, Z.Q. (2015). “Incorporation of modified compression field theory model into a nonlinear laminated reinforced concrete shear wall element.” Master thesis, Dalian University of Technology, Dalian, China. (in Chinese)

Liu, Y.R., Zhou, W.Y., Yang, Q. (2007). “A distributed memory parallel element-by-element scheme based on Jacobi-conditioned conjugate gradient for 3D finite element analysis.” *Finite Element in Analysis and Design*, 43(6-7), 494-503.

Lu, X. Z., Xie, L.L., Guan, H., Huang, Y.L., Lu, X. (2015). “A shear wall element for nonlinear seismic analysis of super high-rise buildings using OpenSees.” *Finite Element in Analysis and Design*, 98, 14-25.

Mazzoni, S., McKenna, F., Scott, M.H., Fenves, G.L. (2015). Open system for earthquake engineering simulation command language manual. http://OpenSees.berkeley.edu/wiki/index.php/Command_Manual (accessed 7.07.2015).

McKenna, F., and Fenves, G. (2007). “Using the OpenSees interpreter on parallel computers.” *NEESit Report No.TN-2007-16*, NEES Cyberinfrastructure Center; La Jolla, CA.

NVIDIA CUDA Basic Linear Algebra Subroutines (cuBLAS) Library. https://developer.nvidia.com/cublas (accessed: 7.07.2015).

Papadrakakis, M., Stavroulakis, G., Karatarakis, A. (2011). “A new era in scientific computing: Domain decomposition methods in hybrid CPU–GPU architectures.” *Computer Methods in Applied Mechanics and Engineering*, 200(13-16), 1490-1508.

PEER Strong Motion Database. (2015). Pacific Earthquake Engineering Research Center. http://peer.berkeley.edu/smcat/ (accessed 7.07.2015).

Smith, B., Bjorstad, P., Gropp, W. (1996). *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, London, UK.

Sotelino, E. D. (2003). “Parallel processing techniques in structural engineering applications.” *Journal of Structural Engineering*, 129(12), 1698-1706.

Spacone, E., Filippou, F. C., Taucer, F. (1996). “Fiber beam-column model for nonlinear analysis of R/C frames.” *Earthquake Engineering and Structural Dynamics*, 25(7), 711-725.

Yang, Y. S., and Hsieh, S. H. (2002) “Iterative mesh partitioning optimization for parallel nonlinear dynamic finite element analysis with direct substructuring.” *Computational Mechanics*, 28(6), 456-468.

Yang, Y. S., Hsieh, S. H., Hsieh, T. J. (2012). “Improving parallel substructuring efficiency by using a multilevel approach.” *Journal of Computing in Civil Engineering*, 2012, 26(4), 457–464.

Vecchio, F. J. (1999) “Towards cyclic load modeling of reinforced concrete.” *ACI Structural Journal*, 96(2), 193-202.

Vecchio, F. J., Collins, M. P. (1986) “The modified compression field theory for reinforced concrete elements subjected to shear.” *ACI Journal*, 83(2), 219-231.

Zhang, X.Y., Wang, Q., Zhang Y.Q., (2012). “Model-driven level 3 BLAS performance optimization on Loongson 3A processor”. *In IEEE 18th International Conference on Parallel and Distributed Systems*, Singapore.