Structural Engineering and Mechanics, Vol. 38, No. 4 (2011) 405-416
DOI: http://dx.doi.org/10.12989/sem.2011.38.4.405

A two-level parallel algorithm for material nonlinearity problems

Jeeho Lee* and Min Seok Kim

Department of Civil and Environmental Engineering, Dongguk University-Seoul, Seoul 100-715, Korea

(Received April 28, 2010, Accepted November 24, 2010)

Abstract. An efficient two-level domain decomposition parallel algorithm is suggested to solve large-DOF structural problems with nonlinear material models generating unsymmetric tangent matrices, such as a group of plastic-damage material models. The parallel version of the stabilized bi-conjugate gradient method is developed to solve unsymmetric coarse problems iteratively. In the present approach the coarse DOF system is solved parallelly on each processor rather than the whole system equation to minimize the data communication between processors, which is appropriate to maintain the computing performance on a non-supercomputer level cluster system. The performance test results show that the suggested algorithm provides scalability on computing performance and an efficient approach to solve large-DOF nonlinear structural problems on a cluster system.

Keywords: parallel algorithm; domain decomposition; material nonlinearity; plastic-damage model.

1. Introduction

As engineering problems become more complicated and coupled over solid, fluid and thermal fields, computing power demands for their numerical simulations increase extremely. The speed of these computing demands is often much faster than modern hardware performance enhancement, even though it is dramatically and amazingly improved every day.

Parallel algorithms have been suggested to solve large structural engineering and/or its coupled problems from two approaches: the system matrix decomposition approach (Balay *et al.* 2002, Blackford *et al*. 1997, Brill and Pinder 2002) and the domain decomposition approach (Mandel 1993, Smith *et al.* 1996, Toselli and Widlund 2004). Due to the limitation of communication bandwidth between processors, domain decomposition methods based on the substructuring concept are often preferred for developing a parallel structural analysis algorithm on non-supercomputer level cluster systems.

In this paper, an efficient two-level domain decomposition algorithm is suggested to parallelly solve large structural problems with the nonlinear material model generating unsymmetric tangent matrices on a cluster system. The coarse problem (Smith *et al.* 1996, Toselli and Widlund 2004) is set based on coarse nodes consisting of boundaries of subdomains and solved using a parallel iterative solver. Each subdomain problem becomes a Dirichlet boundary value problem and solved

*Corresponding author, Professor, E-mail: jeeholee@dgu.edu

by a multi-frontal sparse algorithm (Davis 2004). The Schur complement is constructed and assembled in multiple subdomain problems and used to derive a preconditioner for solving coarse problems. Once the coarse problem is solved, on each (physical and/or logical) processor the local problem is computed as a Dirichlet problem with given boundary values from the computed coarse nodal data. For the efficient parallel iterative solver that can handle unsymmetric system, the stabilized bi-conjugate gradient method (Van der Vorst 1992) is modified.

After describing a two-level domain decomposition method and a suggested parallel version of the stabilized bi-conjugate gradient method, the unsymmetric tangent matrix derived from the plastic-damage model is explained in the context of the Newton-Raphson-type nonlinear solving algorithm. Then, large-DOF nonlinear structural problem with the plastic-damage material model is analyzed to test and discussed the performance of the suggested parallel algorithm.

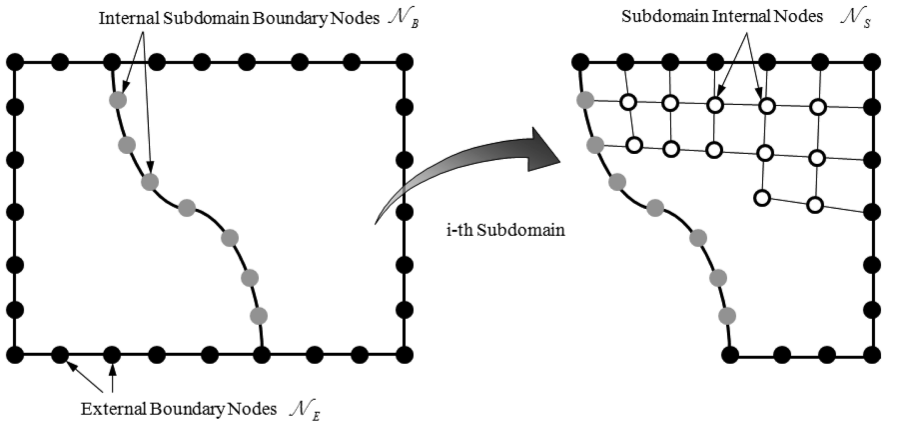
2. Two-level domain decomposition method

The finite element formulation is used to numerically model the physical domains of problems. Then, two-level domain decomposition based on the substructuring method is applied as shown in Fig. 1: the coarse problem level and subdomain level (Toselli and Widlund 2004). The subdomain level problem is also called the local problem. The coarse problem is constructed on the DOF of the coarse problem nodal set, $\mathcal{N}_C$, which is defined as

$$\mathcal{N}_C = \mathcal{N}_E \cup \mathcal{N}_B \tag{1}$$

where $\mathcal{N}_E$ and $\mathcal{N}_B$ are the external boundary nodal set and internal subdomain boundary nodal set of a problem domain, respectively, as depicted in Fig. 1(a). The i-th subdomain problem is defined on the local DOF of the i-th local nodal set, $\mathcal{N}_L^i$, which consists of the internal nodes set, $\mathcal{N}_S^i$, and all the boundary nodes related to the i-th subdomain (Fig. 1(b))

$$\mathcal{N}_L^i = \mathcal{N}_S^i \cup \mathcal{N}_E^i \cup \mathcal{N}_B^i \tag{2}$$



(a) Coarse problem (b) Subdomain (local) problem

Fig. 1 Two-level domain decomposition

A system of equilibrium equations with the i-th DOF vector on $\mathcal{N}_S^i$, $\mathbf{U}_I^{(i)}$, and the coarse DOF vector, $\mathbf{U}_\Gamma$, can be written in the matrix algebraic context of the finite element formulation

$$\begin{bmatrix} \mathbf{K}_{II}^{(1)} & \mathbf{0} & \cdots & \mathbf{K}_{I\Gamma}^{(1)} \\ \mathbf{0} & \mathbf{K}_{II}^{(2)} & & \mathbf{K}_{I\Gamma}^{(2)} \\ \vdots & & \ddots & \vdots \\ \mathbf{K}_{\Gamma I}^{(1)} & \mathbf{K}_{\Gamma I}^{(2)} & \cdots & \mathbf{K}_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{U}_I^{(1)} \\ \mathbf{U}_I^{(2)} \\ \vdots \\ \mathbf{U}_\Gamma \end{bmatrix} = \begin{bmatrix} \mathbf{F}_I^{(1)} \\ \mathbf{F}_I^{(2)} \\ \vdots \\ \mathbf{F}_\Gamma \end{bmatrix} \tag{3}$$

where $\mathbf{F}_I^{(i)}$ is the corresponding force or residual vector. The last row of the submatrix blocks and the rest ones in Eq. (3) can be separated into the following equations

$$\sum_{i=1}^{ND} [\mathbf{K}_{\Gamma I}^{(i)} \mathbf{U}_I^{(i)}] + \mathbf{K}_{\Gamma\Gamma} \mathbf{U}_\Gamma = \mathbf{F}_\Gamma \tag{4a}$$

$$\mathop{\mathbf{A}}_{i=1}^{ND} [\mathbf{K}_{II}^{(i)} \mathbf{U}_I^{(i)} + \mathbf{K}_{I\Gamma}^{(i)} \mathbf{U}_\Gamma] = \mathbf{F}_I \tag{4b}$$

Substituting Eq. (4a) into Eq. (4b) leads to the following coarse problem equation with the Schur complement

$$\mathbf{K}_S \mathbf{U}_\Gamma = \mathbf{g}_\Gamma \tag{5}$$

where the Schur complement matrix and the effective force vector are given respectively

$$\mathbf{K}_S = \mathop{\mathbf{A}}_{i=1}^{ND} [\mathbf{K}_{\Gamma\Gamma}^{(i)} - \mathbf{K}_{\Gamma I}^{(i)} (\mathbf{K}_{II}^{(i)})^{-1} \mathbf{K}_{I\Gamma}^{(i)}] \tag{6a}$$

$$\mathbf{g}_\Gamma = \mathop{\mathbf{A}}_{i=1}^{ND} [\mathbf{F}_\Gamma^{(i)} - \mathbf{K}_{\Gamma I}^{(i)} (\mathbf{K}_{II}^{(i)})^{-1} \mathbf{F}_I^{(i)}] \tag{6b}$$

In the next section, it is discussed in depth how to set up and solve Eq. (5) efficiently using a cluster-based parallel computing system. Once the coarse DOF is computed by solving Eq. (5), the local DOF in the i-th subdomain is calculated backward using Eq. (4a)

$$\mathbf{U}_I^{(i)} = (\mathbf{K}_{II}^{(i)})^{-1} [\mathbf{F}_I^{(i)} - \mathbf{K}_{I\Gamma}^{(i)} \mathbf{U}_\Gamma] \tag{7}$$

It is noted that the local problem in each subdomain is a Dirichlet boundary value problem with known boundary values, $\mathbf{U}_\Gamma$ in the present method and Eq. (7) can be solved efficiently by a multi-frontal sparse algorithm without computing the inverse matrix.

3. Parallelized Bi-CGStab algorithm for domain decomposition

For accelerated and scalable computation at the coarse problem, a parallelized algorithm based on the stabilized bi-conjugate gradient method (Bi-CGStab) is developed. Because many nonlinear material models including plasticity with non-associative flow rules and plastic-damage models give

Initialize $\hat{\mathbf{r}}_0 = \mathbf{r_0}$; $\rho = \hat{\mathbf{r}}_0^T \cdot \mathbf{r}$; $\hat{\mathbf{p}} = \hat{\mathbf{r}}_0$

FOR $i = 1$, *max_ iteration*

$\hat{\mathbf{p}}_i = \mathbf{M}^{-1}\hat{\mathbf{p}}_i$; $\mathbf{v}_i = \mathbf{K}\hat{\mathbf{p}}_i$; $\alpha_i = \dfrac{\rho_{i-1}}{\hat{\mathbf{r}}_0^T \mathbf{v}_i}$

$\mathbf{s}_i = \mathbf{r}_{i-1} - \alpha_i \mathbf{v}_i$

IF $\|\mathbf{s}_i\| <$ *tolerance*

THEN $\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i \hat{\mathbf{p}}_i$, EXIT

$\hat{\mathbf{s}}_i = \mathbf{M}^{-1}\mathbf{s}_i$; $\mathbf{t}_i = \mathbf{K}\hat{\mathbf{s}}_i$; $\omega = \dfrac{\mathbf{t}_i^T \mathbf{s}_i}{\mathbf{t}_i^T \mathbf{t}_i}$

$\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i \hat{\mathbf{p}}_i + \omega_i \hat{\mathbf{s}}_i$

$\mathbf{r}_i = \mathbf{s}_i - \omega_i \mathbf{t}_i$

IF $\|\mathbf{r}_i\| <$ *tolerance*, EXIT

$\rho_i = \hat{\mathbf{r}}_0^T \mathbf{r}_i$; $\beta = \dfrac{\rho_i \rho_{i-1}}{\alpha_i \omega_i}$

$\hat{\mathbf{p}}_{i+1} = \mathbf{r}_i - \beta\omega_i \mathbf{v}_i + \beta\hat{\mathbf{p}}_i$

END

Fig. 2 Parallelized Bi-CGStab algorithm

unsymmetric tangent stiffness matrices, symmetric iterative methods including the conjugate gradient method are not appropriate to solve material nonlinearity problems. It is also considered that the domain decomposition approach is more efficient in computing element tangent stiffness matrices than the system matrix decomposition approach (Balay *et al.* 2002, Blackford *et al.* 1997, Brill and Pinder 2002).

The original algorithm of Bi-CGStab (Van der Vorst 1992) is modified as listed in Fig. 2 for a parallelized version of Bi-CGStab algorithm in the context of the domain decomposition. In the present approach the coarse DOF system Eq. (5) is solved parallelly on each processor rather than the whole system equation (Brill and Pinder 2002) to minimize the data communication between processors, which is appropriate to maintain the computing performance on a non-supercomputer level cluster system. In Fig. 2 the iteration algorithm is supposed to solve the linear system: $\mathbf{Kx} = \mathbf{r}_0$ with the Jacobi pre-conditioner $\mathbf{M}$. A group of the vectors $\{\hat{\mathbf{P}}, \hat{\mathbf{s}}, \hat{\mathbf{t}}, \hat{\mathbf{r}}_0\}$ is the assembled ones at shared coarse DOF, while $\{\mathbf{P}, \mathbf{s}, \mathbf{t}, \mathbf{r}_0\}$ is its counterpart of the contribution from each subdomain. The relationship between these two groups are schematically explained in Fig. 3. The other directional vectors and scalars are defined as in the original Bi-CGStab method (Van der Vorst 1992).

It is noted in Fig. 3 that each logical processor for one subdomain needs to hold $\{\mathbf{P}, \mathbf{s}, \mathbf{t}, \mathbf{r}_0\}$ and its assembled counterpart $\{\hat{\mathbf{P}}, \hat{\mathbf{s}}, \hat{\mathbf{t}}, \hat{\mathbf{r}}_0\}$ only for the corresponding coarse DOF instead of all coarse DOF. This enhances the computing performance by saving significant amount of data communication time on a cluster system, mostly having low bandwidth connection.

The suggested parallel algorithm can be implemented with MPI (Gropp *et al.* 1994) for two or more processors located in distributed cluster nodes. In Fig. 4, the computing procedure and necessary data exchanges in the suggested parallel Bi-CGStab algorithm exampled on a two-processor system is diagramed.

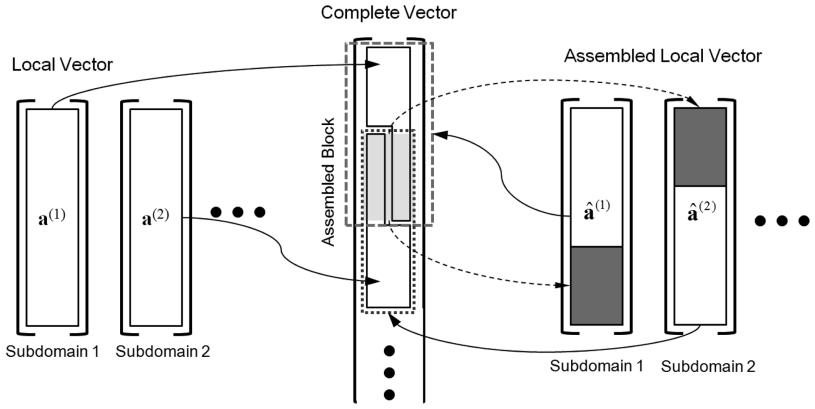


Fig. 3 Local vector update: $\mathbf{a}^i \in \{\mathbf{P}, \mathbf{s}, \mathbf{t}, \mathbf{r}_0\}$, $\hat{\mathbf{a}}^i \in \{\hat{\mathbf{P}}, \hat{\mathbf{s}}, \hat{\mathbf{t}}, \hat{\mathbf{r}}_0\}$

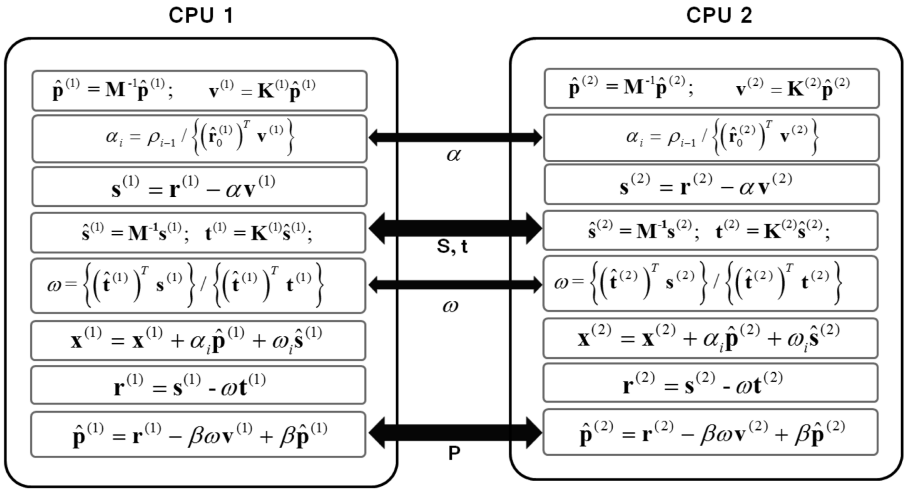


Fig. 4 Vector and scalar transfer between two processors

4. Numerical algorithm for nonlinear material models

4.1 Plastic-damage model

The plastic-damage model (ABAQUS 2005, Lee and Fenves 1998, Lubliner *et al.* 1989) is a nonlinear material model derived from classical plasticity and continuum damage mechanics to represent concrete-like materials. The model can be used as a generalized constitutive form of plasticity and continuum damage models (Mazars and Pijaudier-Cabot 1989, Simo and Ju 1987). Like other material models associated with the non-associative flow rule, most of the suggested plastic-damage models produce unsymmetric tangent stiffness matrices, which play important roles in the Newton-Raphson iteration scheme. In a plastic-damage model with the small strain assumption (Lee and Fenves 1998), the constitutive equations are expressed as follows.

(1) Stress-strain relation and stress admissibility

$$\boldsymbol{\sigma} = (1 - D(\kappa))\overline{\boldsymbol{\sigma}} \tag{8a}$$

$$\overline{\boldsymbol{\sigma}} = \mathbf{E}_0 : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p) \tag{8b}$$

where $\boldsymbol{\sigma}$ and $\overline{\boldsymbol{\sigma}}$ are the stress and effective stress tensors, respectively; D is the degradation variable as a function of damage variable vector $\boldsymbol{\kappa}$; $\mathbf{E}_0$ is the elastic moduli tensor; $\boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}^p$ are the total strain and its plastic strain part. Using the yield surface function F, the stress admissibility is defined by the following inequality equation

$$\overline{\boldsymbol{\sigma}} = \mathbf{E}_0 : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p) \in \{\overline{\boldsymbol{\sigma}} | F(\overline{\boldsymbol{\sigma}}, \kappa) \leq 0\} \tag{9}$$

(2) Flow rule: the plastic strain rate is assumed to be generated by the following non-associative flow rule equation and the plastic potential function Φ

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\gamma} \nabla_{\overline{\boldsymbol{\sigma}}} \Phi(\overline{\boldsymbol{\sigma}}) \tag{10}$$

where γ is the plastic consistency parameter and its rate does not give a physical meaning.

(3) Damage evolution rule

$$\dot{\boldsymbol{\kappa}} = \dot{\gamma} \mathbf{H}(\hat{\overline{\boldsymbol{\sigma}}}, \boldsymbol{\kappa}) \tag{11}$$

where $\mathbf{H}$ is a generalized form of the hardening/softening relations and associated with the plastic dissipation.

(4) Loading/unloading conditions: the following inequality equations are used to define the present loading state.

$$\dot{\gamma} \geq 0; \;\; F(\overline{\boldsymbol{\sigma}}, \kappa) \leq 0; \;\; \dot{\gamma} F(\overline{\boldsymbol{\sigma}}, \kappa) = 0 \tag{12a,b,c}$$

Those equations can be derived as the Kuhn-tucker conditions in nonlinear optimization.

4.2 Numerical algorithm with unsymmetric tangent matrix

The constitutive equations for the plastic-damage model are numerically discretized along with time using the backward-Euler scheme:

$$\begin{aligned} \overline{\boldsymbol{\sigma}}_{n+1} &= \mathbf{E}_0 : (\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}^p_{n+1}) \\ &= \boldsymbol{\sigma}^{tr}_{n+1} - \mathbf{E}_0 : \Delta\boldsymbol{\varepsilon}^p \end{aligned} \tag{13a}$$

$$\Delta\boldsymbol{\varepsilon}^p = \gamma_{n+1} \frac{\partial \Phi}{\partial \overline{\boldsymbol{\sigma}}_{n+1}} \tag{13b}$$

The return-mapping algorithm is implemented with the trial effective stress $\boldsymbol{\sigma}^{tr}_{n+1} = \mathbf{E}_0 : (\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}^p_n)$ as a predictor and following plastic and damage correctors computation to satisfy the stress admissibility and the total stress relation written in the form

$$\boldsymbol{\sigma}_{n+1} = (1-D_{n+1})\overline{\boldsymbol{\sigma}}_{n+1} \tag{14}$$

An appropriate algorithmic tangent matrix is required to have quadratic convergency in a Newton-Raphson-type nonlinear solving algorithm (Lee and Fenves 2001, Simo and Taylor 1986). The consistent algorithmic tangent stiffness matrix is derived from the following system of residual equations

$$\mathbf{E}_0^{-1}:\overline{\boldsymbol{\sigma}}_{n+1} = \boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_{n+1}^{p} \tag{15a}$$

$$\kappa_{n+1} = \kappa_n + \gamma_{n+1}\mathbf{H}(\hat{\overline{\boldsymbol{\sigma}}}_{n+1}, \boldsymbol{\kappa}_{n+1}) \tag{15b}$$

$$F(\overline{\boldsymbol{\sigma}}_{n+1}, \kappa_{n+1}) = 0 \tag{15c}$$

The relation between the stress and strain increments is given

$$d\overline{\boldsymbol{\sigma}}_{n+1} = \left[\mathbf{S} - \frac{\mathbf{S}:\nabla\Phi\otimes\mathbf{T}_{f\sigma}:\mathbf{S}}{\mathbf{T}_{f\sigma}:\mathbf{S}:\nabla\Phi - T_{f\gamma}}\right]:d\boldsymbol{\varepsilon}_{n+1} \tag{16}$$

which leads to the algorithmic tangent matrix for the plastic-damage model

$$\frac{\partial\boldsymbol{\sigma}_{n+1}}{\partial\boldsymbol{\varepsilon}_{n+1}} = [(1-D_{n+1})\mathbf{I} - \overline{\boldsymbol{\sigma}}_{n+1}\otimes\mathbf{T}_{D\sigma}]:\left[\mathbf{S} - \frac{\mathbf{S}:\nabla\Phi\otimes\mathbf{T}_{f\sigma}:\mathbf{S}}{\mathbf{T}_{f\sigma}:\mathbf{S}:\nabla\Phi - T_{f\gamma}}\right] - \left(\overline{\boldsymbol{\sigma}}_{n+1}\otimes\frac{\partial D_{n+1}}{\partial\boldsymbol{\kappa}_{n+1}}\right)\left(\frac{\mathbf{T}_{\kappa\gamma}\otimes\mathbf{T}_{f\sigma}:\mathbf{S}}{\mathbf{T}_{f\sigma}:\mathbf{S}:\nabla\Phi - T_{f\gamma}}\right) \tag{17}$$

where $\mathbf{S} = (\mathbf{E}_0^{-1} + \gamma\partial^2\Phi/\partial\overline{\boldsymbol{\sigma}}_{n+1}^2)^{-1}:\mathbf{T}_{D\sigma}, \mathbf{T}_{f\sigma}, \mathbf{T}_{\kappa\gamma}$ and $T_{f\gamma}$ are the derivative vectors(scalar) defined by Lee and Fenves (2001).

The present algorithmic tangent of the plastic-damage material model is used to construct element stiffness matrices, and then, $\mathbf{K}_{II}^{(i)}$ and $\mathbf{K}_{\Gamma}$, local subdomain stiffness matrices in Eq. (3), which are also unsymmetric. The overall nonlinear solving algorithm in the context of the Newton-Raphson iteration scheme is shown in Fig. 5.

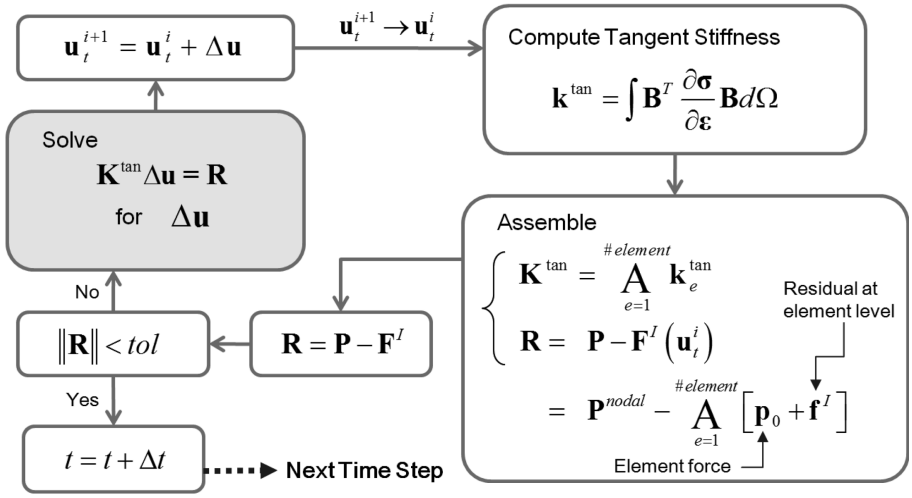


Fig. 5 Nonlinear solving algorithm in the context of Newton-Raphson scheme

The structural tangent stiffness $\mathbf{K}^{\tan}$ becomes unsymmetric if any part of the problem domain undergoes nonlinear material behavior range. The domain decomposition formulation described in Section 2 can be associated directly with the solving algorithm in Fig. 5, and the suggested parallel Bi-CGStab algorithm is applied on solving the coarse problem derived from the structural problem: $\mathbf{K}^{\tan}\Delta\mathbf{u} = \mathbf{R}$.

5. Parallel performance evaluation

The suggested parallel algorithm is implemented for material nonlinearity problems including the plastic-damage model described in Section 4. A program structure diagram of the coded finite element program with the present parallel algorithm is shown in Fig. 6. In addition to Bi-CGStab algorithm at the coarse problem level, to handle high computing-time-consuming nonlinear problems at the subdomain level, a multi-frontal sparse solver is used (Davis 2004). To test the performance of the suggested two-level parallel domain decomposition algorithm, a structural

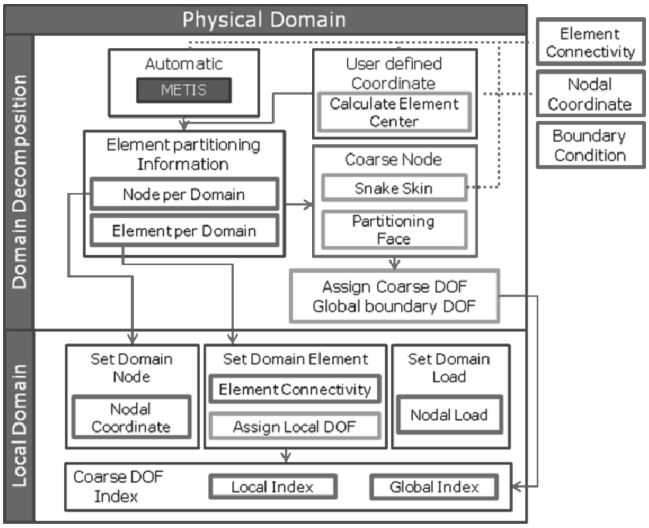


Fig. 6 Two-level domain decomposition

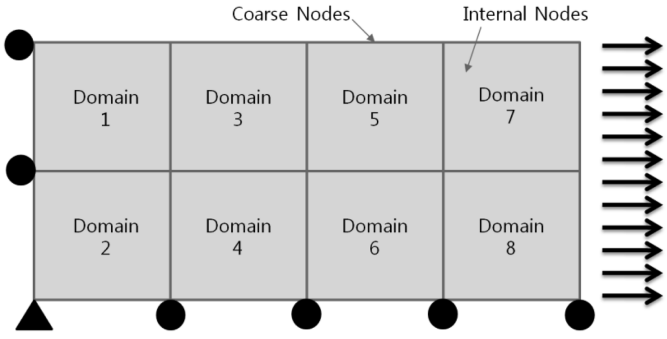


Fig. 7 Eight-subdomain example for performance test

problem as shown in Fig. 7 is solved with one to 16 subdomains, which are assigned to each physical processor. The plastic-damage material model is used with standard plain concrete material properties, which are not presented here because specific material behavior is not interested in this study. Total 10 time steps are applied along the right side: first 5 steps result in linear behavior and the rest 5 steps simulate nonlinear material behavior.

The test problem size is as follows: number of nodes: 31,626; number of plane stress elements: 31,250; number of DOF: 63,252; number of active (non-prescribed) DOF: 62,876. The computation is performed on the cluster system consisting of 8-node Apple X-Server with two 64bit PowerPC processors per cluster node, which provides 16 physical processors. Each cluster node has 2 Gbyte memory and communicated through dual channel Gigabit Ethernet.

In Table 1 the test results for all 10 steps in computing time are summarized with the used number of processors (CPUs). It is observed that larger number of processors increases the overall computing performance up to 8 processors, and then the performance does not increase any more on 16 processors. This flat performance is mainly due to the increase of the coarse problem size.

It is also observed that the used memory size on each processor is consistently decreased with the increase of processors despite the flat performance, which means increase of processors and the corresponding subdomains is still effective in the aspect of computable problem size expansion. From all results the model build time for partitioning the domain into subdomains take less than 1.5% in the total computing time (Model Build and Solve in Table 1), which tends to be more negligible with more time steps, while it consumes significant portion of the total computing time in one-step linear problems. Scalability of computing performance of the present algorithm with more processors is shown visually in Fig. 8.

Table 1 Computing Time (10 total time steps: 5 linear and 5 non-linear steps)

CPUs	Model Build (sec)	Solve (sec)	Ratio*	Number of Coarse DOF	Number of Local DOF	Used Memory Size
2	17.44	1149.09	1.50%	1405	30740	1,500Mb
4	3.16	393.10	0.80%	1925	15246	500Mb
8	1.56	228.43	0.68%	2631	7500	200Mb
16	2.54	224.93	1.12%	3677	3700	100Mb

*Ratio of Model Build time to total computing time (Model Build & Solve)

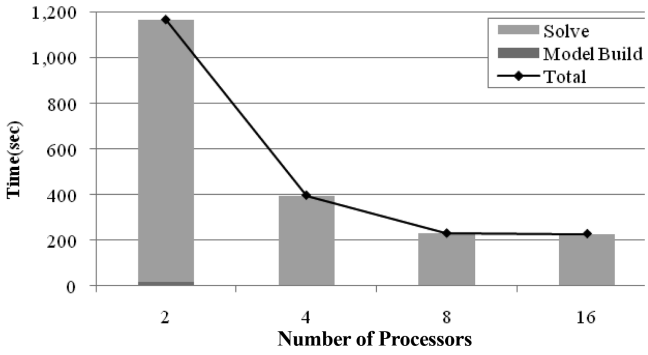


Fig. 8 Scalability of computing performance with number of processors

Table 2 Computing time for solving equation at one linear time step (sec)

CPUs	Assemble Procedure	Schur Complement	Coarse Problem	Local Problem	Total
2	1.80	39.93	4.20	0.15	46.07
4	0.30	11.96	6.52	0.06	18.84
8	0.13	3.37	7.42	0.02	10.94
16	0.08	1.62	8.77	0.01	10.48

Table 3 Computing time for solving problem at one nonlinear time step (sec)

CPUs	Assemble Procedure	Schur Complement	Coarse Problem	Local Problem	Total
2	23.37	119.78	12.60	0.44	156.19
4	3.87	35.89	19.57	0.17	59.49
8	1.70	10.11	22.27	0.06	34.14
16	1.08	4.85	26.32	0.03	32.28

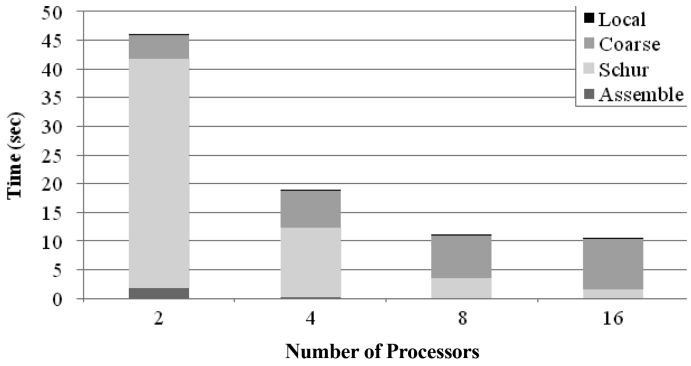


Fig. 9 Computing time for solving problem (Linear time step)

The averaged computing time results are analyzed in depth for linear and nonlinear time steps separately in Table 2 and 3. Four computing parts are measured: Assemble Procedure, Schur Complement Computation, Coarse Problem and Local Problem. At both linear and nonlinear steps total computing time trends with the number of processors are the same as seen in Table 1. Comparing Table 2 and 3, it is observed that each computing part except Assemble Procedure shows the same performance pattern at both linear and nonlinear time steps considering nonlinear step requires approximately 3 iterations, which explains why nonlinear steps take about 300% of linear step computing time.

Assemble Procedure takes 12 times more at a nonlinear step than a linear step, because construction of element tangent stiffness matrix for material models with strong nonlinearity consumes 4 or more times computing power than linear material models. This computing part shows strong scalability and its portion in the total computing time drops rapidly as the number of processors increases.

When the number of processors increases from 4 to 8, the cross between two important parts,

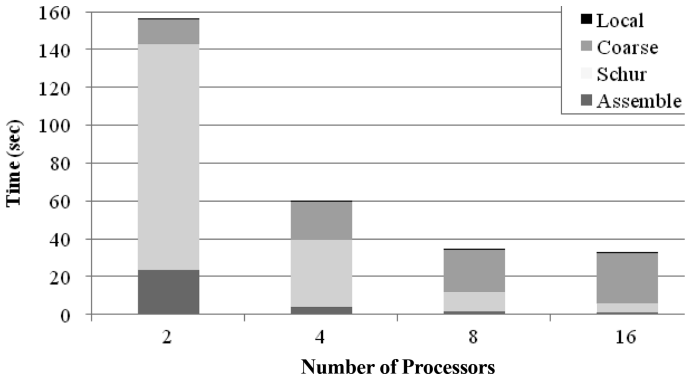


Fig. 10 Computing time for solving problem (Non-linear time step)

Schur Complement and Coarse Problem, over computing time occurs: Coarse Problem part starts to take more computing time than Schur Complement part. Since then, increase of processors does not help to boost the computing performance due to the increase of coarse DOF, which requires more iterations in the parallelized Bi-CGStab procedure. This trend is shown in Fig. 9 for a linear time step and in Fig. 10 for a nonlinear time step. It is noted that the Conjugate Gradient algorithm for symmetric linear problems also requires more iteration with larger coarse DOF vectors.

6. Conclusions

An efficient two-level parallel domain decomposition algorithm is suggested to solve large-DOF structural problems with nonlinear material models, especially with the plastic-damage model, which generates unsymmetric tangent stiffness matrices due to its strong nonlinearity. The parallel version of the stabilized bi-conjugate gradient method is newly developed to solve unsymmetric coarse problems iteratively. The Dirichlet boundary value problem in each subdomain is formulated in the context of the finite element method and can be efficiently computed using a multi-frontal sparse solver. The performance test results from structural problems with the plastic-damage material model show that the suggested algorithmic approach provides scalability on computing performance and an efficient method to solve large-DOF structural problems on a cluster system. It is also observed that once maximum computing performance is achieved with the appropriate amount of processors, using more processors does not give scalability further, while it can expand the problem size to be computed at the maximum performance.

References

ABAQUS (2005), *ABAQUS Theory Manual*, ABAQUS Inc., RI, USA.

Balay, S., Buschelman, K., Gropp, W., Kaushik, D., Knepley, M., McInnes, L.C., Smith, B. and Zhang, H. (2002), *PETSc Users' Manual, Report 95/11-Rev. 2.2.3*, Argonne National Laboratory, IL.

Blackford, L.S., Choi, J., Cleary, A., D'Azevedo, E., Demmel, J., Dhillon, I., Dongarra, J., Hammarling, S., Henry, G., Petitet, A., Stanley, K., Walker, D. and Whaley, R.C. (1997), *ScaLAPACK Users' Guide*, SIAM

Publications, Philadelphia, PA.
Brill, S.H. and Pinder, G.F. (2002), "Parallel implementation of the Bi-CGSTAB method with block red-black Gauss-Seidel preconditioner applied to the Hermite collocation discretization of partial differential equations", *Parallel Comput.*, **28**(3), 399-414.
Davis, T.A. (2004), "A column pre-ordering strategy for the unsymmetric-pattern multifrontal method", *ACM T. Math. Software*, **30**(2), 165-195.
Gropp, W., Lusk, E. and Skjellum, A. (1994), *Using MPI: Portable Parallel Programming with the Message Passing Interface*, The MIT Press.
Lee, J. and Fenves, G.L. (1998), "Plastic-damage model for cyclic loading of concrete structures", *J. Eng. Mech.-ASCE*, **124**(8), 892-900.
Lee, J. and Fenves, G.L. (2001), "A return-mapping algorithm for plastic-damage models: 3-D and plane stress formulation", *Int. J. Numer. Meth. Eng.*, **50**(2), 487-506.
Lubliner, J., Oliver, J., Oller, S. and Onate, E. (1989), "A plastic-damage model for concrete", *Int. J. Solids Struct.*, **25**(3), 299-326.
Mandel, J. (1993), "Balancing domain decomposition", *Commun. Numer. Meth. Eng.,* **9**(3), 233-241.
Mazars, J. and Pijaudier-Cabot, G. (1989), "Continuum damage theory-application to concrete", *J. Eng. Mech.-ASCE*, **115**(2), 345-365.
Simo, J.C. and Ju, J.W. (1987), "Stress and strain based continuum damage models - I. Formulation", *Int. J. Solids Struct.,* **23**(7), 821-840.
Simo, J.C. and Taylor, R.L. (1986), "A return mapping algorithm for plane stress elastoplasticity", *Int. J. Numer. Meth. Eng.*, **22**(3), 649-670.
Smith, B.F., Bjørstad, P. and Gropp, W.D. (1996), *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations,* Cambridge University Press.
Toselli, A. and Widlund, O.B. (2004), *Domain Decomposition Methods - Algorithms and Theory,* Springer.
Van der Vorst, H.A. (1992), "Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems", *SIAM J. Sci. Comput.,* **13**(2), 631-644.