This is a post-peer-review, pre-copyedit version of an article published in *Computational Mechanics*.
The final authenticated version is available online at: https://doi.org/10.1007/s00466-018-1579-4

Scalable parallel elastic–plastic finite element analysis using a quasi-Newton method with a balancing domain decomposition preconditioner

Yasunori Yusa · Hiroshi Okada · Tomonori Yamada · Shinobu Yoshimura

Received: 11 February 2018 / Accepted: 17 April 2018

Abstract A domain decomposition method for large-scale elastic–plastic problems is proposed. The proposed method is based on a quasi-Newton method in conjunction with a balancing domain decomposition preconditioner. The use of a quasi-Newton method overcomes two problems associated with the conventional domain decomposition method based on the Newton–Raphson method: (1) avoidance of a double-loop iteration algorithm, which generally has large computational complexity, and (2) consideration of the local concentration of nonlinear deformation, which is observed in elastic–plastic problems with stress concentration. Moreover, the application of a balancing domain decomposition preconditioner ensures scalability. Using the conventional and proposed domain decomposition methods, several numerical tests, including weak scaling tests, were performed. The convergence performance of the proposed method is comparable to that of the conventional method. In particular, in elastic–plastic analysis, the proposed method exhibits better convergence performance than the conventional method.

Keywords Elastic–plastic analysis · Nonlinear finite element method · Domain decomposition method · Quasi-Newton method · Balancing domain decomposition preconditioner

Y. Yusa · H. Okada
Department of Mechanical Engineering, Faculty of Science and Technology, Tokyo University of Science, 2641 Yamazaki, Noda, Chiba 278-8510, Japan
Tel.: +81 4 7124 1501
E-mail: yyusa@rs.tus.ac.jp

T. Yamada · S. Yoshimura
Department of Systems Innovation, School of Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo, Tokyo 113-8656, Japan

1 Introduction

Large-scale nonlinear finite element analysis on complex and realistic structural models is useful to identify the global and localized structural responses in the processes of design and maintenance, as well as accident investigation. When a source of stress concentration such as a crack, a hole or an inclusion exists in the structure, material in the vicinity of the source suffers from severe stress concentration, resulting in plastic deformation. In contrast, a most portion far from the stress concentration source remains in elastic deformation. This problem would be significant especially in large-scale analysis. Moreover, such a problem can also be observed in other engineering fields such as stress concentration at structurally discontinuous parts and welding with a moving heat source. However, a conventional nonlinear finite element method that is based on the Newton–Raphson method cannot consider this problem. In the conventional method, a huge linear system of equations should be solved at every Newton–Raphson iteration step. Large amount of computational complexity is devoted evenly to the whole analysis model, regardless of whether the region is elastic–plastic or not. For this problem, the authors have proposed the partitioned coupling method [43,45,46], in which an analysis model is decomposed into two nonoverlapping domains, i.e., global and local domains. The local domain, which contains the crack, is modeled as an elastic–plastic body, whereas the uncracked global domain is modeled as an elastic body. The two domains are solved iteratively in the context of a nonlinear solution method such as a quasi-Newton method. Similar approaches can be found in the literature. Nishikawa et al. [32] analyzed a welding problem by using the iterative substructure method, in which the vicinity of a welding heat source is sepa-

rated from an analysis model, and the two domains are analyzed iteratively. Nikishkov and Atluri [31] and Pyo et al. [36] presented the elastic–plastic finite element alternating method, in which a mesh without cracks and analytical solutions of an elastic–plastic crack are solved iteratively. Yumoto et al. [41, 42] and Yusa et al. [44] analyzed stress concentration and crack problems by using the coupling-matrix-free iterative s-version finite element method, which is a modification of the s-version finite element method [16]. A local mesh that models a local feature such as a hole or a crack and a global model without the local feature are solved iteratively. This iteration is accelerated by a linear or nonlinear solution method such as the Gauss–Seidel method or a quasi-Newton method [44]. From aforementioned studies, it can be stated that the use of a local domain with an iterative method appears to be a key in reducing computational time to solve large-scale elastic–plastic problems involving local nonlinearity.

However, such a global–local iteration approach has a difficulty such that the analyst should determine the local domain manually before the analysis. The size of the local domain must be affected by the size of a plastic zone, which is unknown before the analysis. To overcome this difficulty, the authors propose a multi-domain iteration approach in the present study. In this approach, an analysis domain is decomposed into multiple subdomains. Some subdomains experience plastic deformation, whereas other subdomains do not. This approach appears to be related strongly to the domain decomposition method (DDM) [37,38]. The domain decomposition method has been studied for many years in the field of mathematics [3, 4, 8, 11–13, 22–25, 37–39], and has been applied to large-scale computational solid mechanics problems in engineering [1, 6, 7, 18, 28, 34, 40]. The DDM decomposes an analysis domain into multiple subdomains, which are assigned to processing elements (PEs) of a parallel computer. From the point of view of a domain decomposition procedure, the DDM can be classified into two methods, i.e., overlapping and nonoverlapping methods [37,38]. The overlapping DDM [3, 4, 8, 24, 37, 38] allows overlapping between adjacent subdomains, whereas the nonoverlapping DDM [11,22,23,25,37–39] does not. In both methods, nonlinear problems have also been studied in the mathematical literature [3, 4, 8, 12, 13, 22–24]. In the present study, the nonoverlapping DDM is considered, due to the compatibility with the aforementioned partitioned coupling method [43, 45, 46]. In practical large-scale analysis, scalability is indispensable in solving a problem in computational time that the analyst can wait for. For example, design analysis generally requires smaller computational time than one night (off hours). In order to ensure scalability, several coarse-grid-correction-based preconditioners for the nonoverlapping DDM, such as balancing domain decomposition (BDD) [25], BDD with diagonal scaling (BDD-DIAG) [33], BDD by constraints (BDDC) [11], finite element tearing and interconnecting (FETI) [15], and dual–primal FETI (FETI-DP) [14], have been proposed. These preconditioners have been used to solve large-scale computational solid mechanics problems in engineering. Approaches of the DDM in computational solid mechanics applications including the use of these preconditioners were summarized by Gosselet and Rey [18]. BDD and BDD-DIAG have been implemented in the software modules of the ADVENTURE System [1, 40], which is a computational mechanics system for large-scale analysis and design. Ogino et al. [34] solved a dynamic problem involving a nuclear pressure vessel under a seismic load by using the ADVENTURE System on the Earth Simulator. Akiba et al. [2] were selected as one of the ACM Gordon Bell Prize finalists in Supercomputing Conference 2006 (SC2006) by using ADVENTURECluster, which is a commercial version of the ADVENTURE System, on the Blue Gene/L supercomputer. FETI and FETI-DP have been implemented in Salinas [6,7], which is a massively parallel implicit structural mechanics/dynamics software. Bhardwaj et al. [7] earned a Gordon Bell Award for special accomplishment in Supercomputing Conference 2002 (SC2002) by using Salinas on the ASCI Red and White supercomputers.

Although the capability of the DDM with a coarse-grid-correction-based preconditioner for the practical computational solid mechanics applications has been demonstrated as described above, most previous studies examined linear elastic problems. In this sense, the DDM can be regarded as a linear system solver. When a nonlinear problem, such as an elastic–plastic problem, is to be solved by the DDM, the Newton–Raphson method is usually used to linearize the nonlinear system of equations of the discretized principle of virtual work [28]. At every Newton–Raphson iteration step, the linearized system of equations, i.e., a linear elastic problem, is solved by the DDM. However, there exists a problem such that a double-loop iteration algorithm is required in this method. The outer loop is a Newton–Raphson iteration, whereas the inner loop is an iteration to solve a linear system of equations, e.g., a conjugate gradient iteration. In general, a double-loop algorithm has large computational complexity. This problem is usually not encountered in small-scale nonlinear finite element analysis due to the use of a direct linear system solver. In practice, the use of an iterative linear system solver is indispensable in large-scale analy-

sis. In order to avoid a double-loop iteration algorithm, other nonlinear system solvers, such as quasi-Newton methods, may be useful. Although pure quasi-Newton methods also require a double-loop iteration algorithm due to the necessity of linear system solution, limited-memory implementation [21] of quasi-Newton methods with an easily invertible initial Jacobian matrix (e.g., the identity matrix) does not require linear system solution. Quasi-Newton methods for nonlinear finite element methods were investigated early in nonlinear computational solid mechanics research [5, 17, 26]. Moreover, the modified Newton method, which is simpler than quasi-Newton methods, was investigated in the same period [9,29]. At present, in the field of nonlinear computational solid mechanics, quasi-Newton methods appear to be advantageous for problems in which the tangent stiffness matrix cannot be computed explicitly. Quasi-Newton methods have also been used in the field of fluid–structure interaction [10,27], in which a nonlinear system of equations on the fluid–structure interface is defined from continuity and equilibrium. This nonlinear system of equations is solved by a nonlinear solution method, such as a quasi-Newton method. Quasi-Newton methods have been numerically demonstrated to provide better performance than Newton–Raphson method in some fluid–structure interaction problems.

Another problem with using the Newton–Raphson method is the local nonlinearity. Some nonlinear problems, such as stress concentration problems, crack problems, and welding problems, produce the local concentration of nonlinear deformation. High-stress regions experience strongly nonlinear deformation, whereas low-stress regions remain in linearly elastic or weakly nonlinear deformation. This feature appears to be significant, especially in large-scale problems. However, in the DDM using the Newton–Raphson method, a global linear system of equations is solved several times, even in linear elastic subdomains. Furthermore, the number of Newton–Raphson iteration steps would be influenced by the region that has the strongest nonlinearity. This problem was also indicated in the literature of the DDM [3,8,22,23,30,35]. To overcome this problem, several approaches have been proposed so far. In the additive Schwartz preconditioned inexact Newton (ASPIN) method [3,8], a nonlinear system of equations is preconditioned nonlinearly by the additive Schwartz preconditioning technique, and then solved by the inexact Newton method. In nonlinear nonoverlapping DDMs [22, 23, 30, 35], a nonlinear system of equations is preconditioned by a coarse-grid-correction-based preconditioner such as BDDC or FETI-DP.

In the present study, for large-scale elastic–plastic problems, a DDM that is based on a quasi-Newton method with a BDD preconditioner is proposed. The use of a quasi-Newton method overcomes two problems associated with the conventional DDM. First, a quasi-Newton method based on limited-memory implementation [21] is able to eliminate linear system solution, resulting in the avoidance of a double-loop iteration algorithm. Several vector operations rather than linear system solution are computed at every quasi-Newton iteration step. Second, this method requires to solve subdomain-wise nonlinear systems of equations. Some subdomains experience elastic–plastic deformation, whereas other subdomains remain in linear elastic deformation. An optimal number of nonlinear iteration steps is automatically devoted to each subdomain, enabling the consideration of local concentration of nonlinear deformation. This idea is an extension of the global–local iteration approach in the partitioned coupling method [46]. Moreover, a BDD preconditioner ensures scalability, which is indispensable in large-scale analysis. In addition, the methodology includes two original elemental techniques. The first is the derivation of an interface nonlinear system of equations for a quasi-Newton method from the principle of virtual work. This nonlinear system of equations is equivalent to the linear system of equations of the conventional DDM when the problem is linearly elastic. In this sense, the proposed DDM can be regarded as an extension of the conventional DDM. The second technique is the application of a BDD preconditioner to a quasi-Newton method. To the best of the authors' knowledge, BDD preconditioners have not been used in quasi-Newton methods. Although the idea and the methodology of the present study is partly similar to some DDMs that are modified for nonlinear problems, such as the ASPIN method [3,8] and nonlinear nonoverlapping DDMs [22, 23, 30, 35], there are two major contributions in the present study. The first one is the use of a quasi-Newton method with a balancing domain decomposition preconditioner, and the second one is its engineering application to a realistic structural model.

In the present paper, the conventional DDM and the proposed DDM are first explained. This discussion includes the derivation of the nonlinear and linear systems of equations to be solved, the application of the BDD preconditioner, and the description of algorithms. Several numerical tests are then performed. Linear elastic and elastic–plastic problems are presented, and weak scaling tests are performed. The results of these numerical tests provide a basic understanding of the convergence performance and scalability of the proposed method.

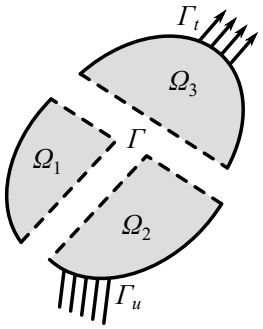

Fig. 1 Decomposed analysis domain.

2 Conventional domain decomposition method based on the Newton–Raphson method in conjunction with conjugate gradient method

Before presenting the proposed DDM, we explain the conventional DDM [1,28,34,40]. The conventional DDM mentioned in the present study is a nonoverlapping DDM that is based on the Newton–Raphson method with the conjugate gradient method. In general, the DDM is a linear system solver due to the use of static condensation. Therefore, the Newton–Raphson method should be used to linearize a nonlinear system of equations. At each Newton–Raphson iteration step, a linear system of equations is statically condensed by the DDM. The condensed linear system of equations is then solved using the conjugate gradient method. The conjugate gradient method is frequently used in the field of computational solid mechanics, whereas other Krylov subspace methods can be used for unsymmetric and/or non-positive-definite problems.

As mentioned above, huge linear systems of equations should be solved several times, even if nonlinear phenomena such as plasticity occur locally. Furthermore, as the nonlinearity becomes strong, the number of Newton–Raphson iteration steps tends to increase. Moreover, the conventional DDM requires a double-loop iteration algorithm. The outer loop is a Newton–Raphson iteration, whereas the inner loop is a conjugate gradient iteration. In general, a double-loop algorithm has large computational complexity.

2.1 Nonlinear and linear systems of equations to be solved

First, an analysis domain, Ω, is decomposed into multiple non-overlapping subdomains, Ω_i $(i = 1, 2, \ldots, N)$, for parallel computing, as shown in Fig. 1. The subdomain interface, which is represented by the dashed lines, is Γ. Traction forces and enforced displacements are prescribed on Γ_t and Γ_u, respectively.

Then, the principle of virtual work is introduced as

$$\int_\Omega \delta\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\sigma}\mathrm{d}\Omega - \int_{\Gamma_t} \delta\boldsymbol{u}^{\mathrm{T}}\boldsymbol{t}\mathrm{d}\Gamma - \int_\Omega \delta\boldsymbol{u}^{\mathrm{T}}\boldsymbol{b}\mathrm{d}\Omega = 0. \tag{1}$$

In this equation, $\boldsymbol{\sigma}$, $\boldsymbol{t}$, and $\boldsymbol{b}$ are stresses, traction forces, and body forces, respectively. Moreover, $\delta\boldsymbol{u}$ and $\delta\boldsymbol{\varepsilon}$ are variations of displacements and strains, respectively. Although Eq. (1) assumes infinitesimal strain, large-strain problems can be dealt with in a similar manner. Equation (1) can be discretized by a shape function matrix, $\boldsymbol{N}$, and a strain–displacement matrix, $\boldsymbol{B}$, as

$$\int_\Omega \boldsymbol{B}^{\mathrm{T}}\boldsymbol{\sigma}\mathrm{d}\Omega - \int_{\Gamma_t} \boldsymbol{N}^{\mathrm{T}}\boldsymbol{t}\mathrm{d}\Gamma - \int_\Omega \boldsymbol{N}^{\mathrm{T}}\boldsymbol{b}\mathrm{d}\Omega = \boldsymbol{0}. \tag{2}$$

This is the nonlinear system of equations to be solved by the Newton–Raphson method. A residual vector, $\boldsymbol{r}$, an internal force vector, $\boldsymbol{f}^{\mathrm{int}}$, and an external force vector, $\boldsymbol{f}^{\mathrm{ext}}$, are defined as

$$\boldsymbol{r} = \boldsymbol{f}^{\mathrm{int}} - \boldsymbol{f}^{\mathrm{ext}}, \tag{3}$$

$$\boldsymbol{f}^{\mathrm{int}} = \int_\Omega \boldsymbol{B}^{\mathrm{T}}\boldsymbol{\sigma}\mathrm{d}\Omega \tag{4}$$

and

$$\boldsymbol{f}^{\mathrm{ext}} = \int_{\Gamma_t} \boldsymbol{N}^{\mathrm{T}}\boldsymbol{t}\mathrm{d}\Gamma + \int_\Omega \boldsymbol{N}^{\mathrm{T}}\boldsymbol{b}\mathrm{d}\Omega, \tag{5}$$

respectively. In the Newton–Raphson method, the unknown displacement vector, $\boldsymbol{u}$, is updated at the k-th iteration step, as

$$\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \Delta\boldsymbol{u}^{(k)}. \tag{6}$$

In this equation, $\Delta\boldsymbol{u}^{(k)}$ is the solution of a linear system of equations:

$$\boldsymbol{K}^{(k)}\Delta\boldsymbol{u}^{(k)} = -\boldsymbol{r}^{(k)}, \tag{7}$$

where

$$\boldsymbol{K}^{(k)} = \frac{\partial\boldsymbol{r}^{(k)}}{\partial\boldsymbol{u}}. \tag{8}$$

If $\boldsymbol{K}^{(k)}$ cannot be computed explicitly, another nonlinear system solver, such as the inexact Newton method, the modified Newton method, or the quasi-Newton method, is used to solve Eq. (2).

In order to solve Eq. (7) by the DDM in conjunction with the conjugate gradient method, Eq. (7) is permutated for subdomain internal degrees of freedom, Ω_i $(i = 1, 2, \ldots, N)$, and subdomain interface degrees of freedom, Γ, as

$$\boldsymbol{P}\boldsymbol{K}^{(k)}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{P}\Delta\boldsymbol{u}^{(k)} = -\boldsymbol{P}\boldsymbol{r}^{(k)}. \tag{9}$$

Here, $\boldsymbol{P}$ is a permutations matrix. The entries of the permutated linear system of equations can be described as

$$\boldsymbol{P}\boldsymbol{K}^{(k)}\boldsymbol{P}^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{K}^{(k)}_{\Omega_1\Omega_1} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{K}^{(k)}_{\Omega_1\Gamma} \\ \boldsymbol{0} & \boldsymbol{K}^{(k)}_{\Omega_2\Omega_2} & \ddots & \vdots & \boldsymbol{K}^{(k)}_{\Omega_2\Gamma} \\ \vdots & \ddots & \ddots & \boldsymbol{0} & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{K}^{(k)}_{\Omega_N\Omega_N} & \boldsymbol{K}^{(k)}_{\Omega_N\Gamma} \\ \boldsymbol{K}^{(k)^{\mathrm{T}}}_{\Omega_1\Gamma} & \boldsymbol{K}^{(k)^{\mathrm{T}}}_{\Omega_2\Gamma} & \cdots & \boldsymbol{K}^{(k)^{\mathrm{T}}}_{\Omega_N\Gamma} & \boldsymbol{K}^{(k)}_{\Gamma\Gamma} \end{bmatrix}, \tag{10}$$

$$\boldsymbol{P}\Delta\boldsymbol{u}^{(k)} = \begin{Bmatrix} \Delta\boldsymbol{u}^{(k)}_{\Omega_1} \\ \Delta\boldsymbol{u}^{(k)}_{\Omega_2} \\ \vdots \\ \Delta\boldsymbol{u}^{(k)}_{\Omega_N} \\ \Delta\boldsymbol{u}^{(k)}_{\Gamma} \end{Bmatrix} \tag{11}$$

and

$$-\boldsymbol{P}\boldsymbol{r}^{(k)} = \begin{Bmatrix} -\boldsymbol{r}^{(k)}_{\Omega_1} \\ -\boldsymbol{r}^{(k)}_{\Omega_2} \\ \vdots \\ -\boldsymbol{r}^{(k)}_{\Omega_N} \\ -\boldsymbol{r}^{(k)}_{\Gamma} \end{Bmatrix}. \tag{12}$$

This equation can be statically condensed as

$$\boldsymbol{S}^{(k)}\Delta\boldsymbol{u}^{(k)}_{\Gamma} = \boldsymbol{g}^{(k)}, \tag{13}$$

where

$$\boldsymbol{S}^{(k)} = \boldsymbol{K}^{(k)}_{\Gamma\Gamma} - \sum_{i=1}^{N} \boldsymbol{K}^{(k)^{\mathrm{T}}}_{\Omega_i\Gamma}\boldsymbol{K}^{(k)^{-1}}_{\Omega_i\Omega_i}\boldsymbol{K}^{(k)}_{\Omega_i\Gamma} \tag{14}$$

and

$$\boldsymbol{g}^{(k)} = -\boldsymbol{r}^{(k)}_{\Gamma} + \sum_{i=1}^{N} \boldsymbol{K}^{(k)^{\mathrm{T}}}_{\Omega_i\Gamma}\boldsymbol{K}^{(k)^{-1}}_{\Omega_i\Omega_i}\boldsymbol{r}^{(k)}_{\Omega_i}. \tag{15}$$

Equation (13) is referred to as a Schur complement system, and $\boldsymbol{S}^{(k)}$ is referred to as a Schur complement matrix. This Schur complement system is to be solved by the conjugate gradient method. At each iteration step of the conjugate gradient method, the product of the Schur complement matrix and a search direction vector is computed using Eq. (14). For the summation term in Eq. (14), every subdomain, Ω_i, is analyzed in parallel under assumed enforced displacements on the subdomain interface, Γ.

2.2 Balancing domain decomposition preconditioning

In order to accelerate convergence and ensure scalability in large-scale analysis, a coarse-grid-correction-based preconditioner for the DDM, such as BDD [25], BDD-DIAG [33], BDDC [11], FETI [15], or FETI-DP [14], should be applied to the Schur complement system. In the present study, BDD and BDD-DIAG are used. Moreover, a diagonal scaling preconditioner, which is not based on coarse grid correction, is used for comparison. At each iteration step of the conjugate gradient method, a residual vector, $\boldsymbol{r}^{(k,i)}_{\mathrm{Schur}} = \boldsymbol{g}^{(k)} - \boldsymbol{S}^{(k)}\Delta\boldsymbol{u}^{(k,i)}_{\Gamma}$, is preconditioned as

$$\boldsymbol{z}^{(k,i)}_{\mathrm{Schur}} = \boldsymbol{M}^{(k)^{-1}}\boldsymbol{r}^{(k,i)}_{\mathrm{Schur}}, \tag{16}$$

where i is an iteration step of the conjugate gradient method, $\boldsymbol{M}^{(k)}$ is a preconditioning matrix, and $\boldsymbol{z}^{(k,i)}_{\mathrm{Schur}}$ is a preconditioned residual vector. Note that i is an iteration step of the conjugate gradient method, whereas k is that of the Newton–Raphson method.

The explanation begins with diagonal scaling preconditioning. The preconditioning matrix of diagonal scaling is

$$\boldsymbol{M}^{(k)}_{\mathrm{DIAG}} = \mathrm{diag}\left(\boldsymbol{K}^{(k)}_{\Gamma\Gamma}\right), \tag{17}$$

which assumes

$$\mathrm{diag}\left(\boldsymbol{S}^{(k)}\right) \approx \mathrm{diag}\left(\boldsymbol{K}^{(k)}_{\Gamma\Gamma}\right). \tag{18}$$

In these equations, the diag() operator extracts diagonal entries from the input matrix. Non-diagonal entries are zero. Diagonal scaling preconditioning can be performed completely in parallel because there is no interaction between subdomains.

BDD preconditioning ensures scalability by using coarse grid correction. The inverse of the BDD preconditioning matrix is

$$\begin{aligned} \boldsymbol{M}^{(k)^{-1}}_{\mathrm{BDD}} =& \boldsymbol{R}^{\mathrm{T}}_0\boldsymbol{S}^{(k)^{-1}}_0\boldsymbol{R}_0 \\ &+ \left(\boldsymbol{I} - \boldsymbol{R}^{\mathrm{T}}_0\boldsymbol{S}^{(k)^{-1}}_0\boldsymbol{R}_0\boldsymbol{S}^{(k)}\right)\boldsymbol{M}^{(k)^{-1}}_{\mathrm{NN}} \\ &\times \left(\boldsymbol{I} - \boldsymbol{R}^{\mathrm{T}}_0\boldsymbol{S}^{(k)^{-1}}_0\boldsymbol{R}_0\boldsymbol{S}^{(k)}\right)^{\mathrm{T}}, \end{aligned} \tag{19}$$

where $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{R}_0$ is a restriction operator to a coarse grid, $\boldsymbol{S}^{(k)}_0$ is a coarse matrix, and $\boldsymbol{M}^{(k)}_{\mathrm{NN}}$ is a Neumann–Neumann preconditioning matrix. First, the restriction operator, $\boldsymbol{R}_0$, is

$$\boldsymbol{R}_0 = \begin{bmatrix} \boldsymbol{Z}^{\mathrm{T}}_{\Omega_1}\boldsymbol{D}_{\Omega_1}\boldsymbol{R}_{\Omega_1\Gamma} \\ \vdots \\ \boldsymbol{Z}^{\mathrm{T}}_{\Omega_N}\boldsymbol{D}_{\Omega_N}\boldsymbol{R}_{\Omega_N\Gamma} \end{bmatrix}. \tag{20}$$

In this expression, $\boldsymbol{R}_{\Omega_i\Gamma}$ is a restriction operator from the subdomain interface, Γ, to the i-th subdomain, Ω_i, and $\boldsymbol{D}_{\Omega_i}$ is a weight matrix that satisfies

$$\sum_{i=1}^{N} \boldsymbol{R}_{\Omega_i\Gamma}^{\mathrm{T}} \boldsymbol{D}_{\Omega_i} \boldsymbol{R}_{\Omega_i\Gamma} = \boldsymbol{I}. \tag{21}$$

Here, $\boldsymbol{Z}_{\Omega_i}$ is a matrix that represents a coarse space based on rigid-body motion. Then, the coarse matrix, $\boldsymbol{S}_0^{(k)}$, is

$$\boldsymbol{S}_0^{(k)} = \boldsymbol{R}_0 \boldsymbol{S}^{(k)} \boldsymbol{R}_0^{\mathrm{T}}. \tag{22}$$

Finally, the inverse of the Neumann–Neumann preconditioning matrix is

$$\boldsymbol{M}_{\mathrm{NN}}^{(k)^{-1}} = \sum_{i=0}^{N} \boldsymbol{R}_{\Omega_i\Gamma}^{\mathrm{T}} \boldsymbol{D}_{\Omega_i}^{\mathrm{T}} \boldsymbol{S}_{\Omega_i}^{(k)^{\dagger}} \boldsymbol{D}_{\Omega_i} \boldsymbol{R}_{\Omega_i\Gamma}, \tag{23}$$

where $\boldsymbol{S}_{\Omega_i}^{(k)^{\dagger}}$ is the generalized inverse of the local Schur complement matrix, $\boldsymbol{S}_{\Omega_i}^{(k)}$. The Schur complement matrix, $\boldsymbol{S}^{(k)}$, in Eq. (14) can also be represented as

$$\boldsymbol{S}^{(k)} = \sum_{i=1}^{N} \boldsymbol{R}_{\Omega_i\Gamma}^{\mathrm{T}} \boldsymbol{S}_{\Omega_i}^{(k)} \boldsymbol{R}_{\Omega_i\Gamma}. \tag{24}$$

Since $\boldsymbol{S}_{\Omega_i}^{(k)}$ is not a regular matrix, a regularization technique should be applied to $\boldsymbol{S}_{\Omega_i}^{(k)}$ in order to obtain its inverse. In the present study, the detailed computational procedures for obtaining $\boldsymbol{R}_0$, $\boldsymbol{S}_0^{(k)}$, and $\boldsymbol{M}_{\mathrm{NN}}^{(k)}$ follow Ogino et al. [33]. Here, the computational procedure of BDD preconditioning can be divided into two main parts. The first is coarse grid correction, and the second is Neumann–Neumann preconditioning. First, coarse grid correction is the solution of a linear system of equations for which the coefficient matrix is $\boldsymbol{S}_0^{(k)}$. This part is difficult to efficiently parallelize. A serial or parallel direct linear system solver has been used for this part [34]. Second, the Neumann–Neumann preconditioning is the multiplication of $\boldsymbol{M}_{\mathrm{NN}}^{(k)^{-1}}$ and a vector. In this part, every subdomain is analyzed in parallel under assumed traction forces on the subdomain interface.

BDD-DIAG preconditioning simplifies BDD preconditioning by replacing the Neumann–Neumann preconditioning matrix, $\boldsymbol{M}_{\mathrm{NN}}^{(k)}$, in Eq. (19) with the diagonal scaling preconditioning matrix, $\boldsymbol{M}_{\mathrm{DIAG}}^{(k)}$, i.e.,

$$\begin{aligned}\boldsymbol{M}_{\text{BDD-DIAG}}^{(k)^{-1}} = {} & \boldsymbol{R}_0^{\mathrm{T}} \boldsymbol{S}_0^{(k)^{-1}} \boldsymbol{R}_0 \\ & + \left(\boldsymbol{I} - \boldsymbol{R}_0^{\mathrm{T}} \boldsymbol{S}_0^{(k)^{-1}} \boldsymbol{R}_0 \boldsymbol{S}^{(k)} \right) \boldsymbol{M}_{\mathrm{DIAG}}^{(k)^{-1}} \\ & \times \left(\boldsymbol{I} - \boldsymbol{R}_0^{\mathrm{T}} \boldsymbol{S}_0^{(k)^{-1}} \boldsymbol{R}_0 \boldsymbol{S}^{(k)} \right)^{\mathrm{T}}. \end{aligned} \tag{25}$$

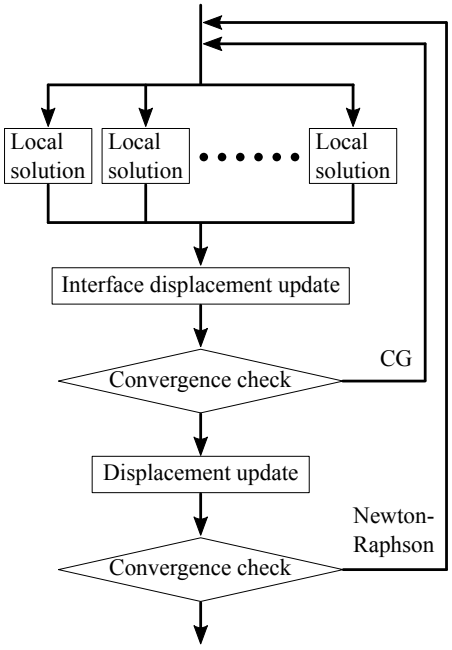

Fig. 2 Flowchart of the conventional domain decomposition method.

BDD-DIAG has an advantage in that its algorithm is simpler than that of BDD. Moreover, its computational time at each iteration step is shorter than that of BDD. However, the number of iteration steps of BDD-DIAG tends to be larger than that of BDD [33].

2.3 Algorithm

In the conventional DDM, a double-loop iteration algorithm consisting of Newton–Raphson iteration and conjugate gradient iteration is required. The computational procedures of the conventional DDM are summarized in Fig. 2. In this algorithm, first, subdomain-wise local problems (the multiplication of Eq. (14) and a vector) are solved in parallel. Second, based on the results of the subdomain-wise local problems, the interface displacement vector is computed by preconditioning (Eq. (16)) and several vector operations in the context of the conjugate gradient method. Some parts of these procedures, such as diagonal scaling preconditioning, Neumann–Neumann preconditioning, and some vector operations, can be parallelized. The convergence of conjugate gradient iteration is then confirmed. After conjugate gradient iteration converges, the displacement vectors at whole degrees of freedom are updated using Eq. (6). Finally, the convergence of Newton–Raphson iteration is checked.

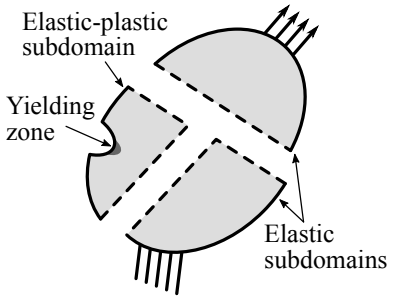

Fig. 3 Decomposed analysis domain of a stress concentration problem in elastic–plastic analysis.

3 Proposed domain decomposition method based on a quasi-Newton method

In this section, we propose a DDM that is based on a quasi-Newton method. The proposed DDM can consider the local concentration of nonlinear deformation, which is caused by stress concentration. Here, a schematic view of a stress concentration problem is illustrated in Fig. 3. In this figure, the analysis domain is decomposed into multiple subdomains. Then, we assume that some subdomains experience elastic–plastic deformation with yielding zones, whereas other subdomains remain in linear elastic deformation. For these subdomains, subdomain-wise nonlinear finite element analyses are performed in parallel at each quasi-Newton iteration step. When a subdomain is in nonlinear deformation, such as elastic–plastic deformation, the subdomain is analyzed by nonlinear iteration, such as Newton–Raphson iteration, modified Newton iteration, or quasi-Newton iteration. When a subdomain is in linear elastic deformation, the number of nonlinear iteration steps becomes one. Moreover, due to the use of a quasi-Newton method, there exists another advantage, namely, the avoidance of a global double-loop iteration algorithm, which is required in the Newton–Raphson method in conjunction with the conjugate gradient method. Note that we consider the case in which many subdomains are in linear elastic deformation. This is due to the local nonlinearity. If the problem is totally nonlinear, then the proposed DDM should also be regarded as involving a double-loop iteration algorithm.

3.1 Nonlinear system of equations to be solved

First, a nonlinear system of equations to be solved by a quasi-Newton method is introduced. This nonlinear system of equations is defined for the subdomain interface degrees of freedom. This idea is similar to the partitioned coupling method [46] and a quasi-Newton-based fluid–structure interaction analysis method [10, 27]. The former method defines a nonlinear system of equations on the interface of global and local domains, whereas the latter method defines a nonlinear system on the fluid–structure interface. However, the mathematical procedure of the proposed method for deriving the nonlinear system of equations is different. Moreover, when the problem is linearly elastic, the proposed nonlinear system of equations becomes linear and equivalent to Eq. (13) in the conventional DDM.

The explanation begins with the principle of virtual work (Eq. (1)). Although Eq. (1) assumes infinitesimal strain, geometrical nonlinear problems with finite strain theory are available, as far as the principle of virtual work can be defined. Moreover, additional terms such as an inertia term and a damping term in dynamic analysis with a time discretization method can be considered in the methodology. Moreover, a constitutive equation is not assumed. The proposed method assumes that nonlinear phenomena are observed locally. Thus, the target problems of the present study are elastic–plastic problems. Then, in Eq. (1), the variation of displacements, $\delta\boldsymbol{u}$, is decomposed additively as

$$\delta\boldsymbol{u} = \delta\boldsymbol{u}_{\Omega_1} + \cdots + \delta\boldsymbol{u}_{\Omega_N} + \delta\boldsymbol{u}_{\Gamma}. \tag{26}$$

Here, $\delta\boldsymbol{u}_{\Omega_i}$ is nonzero in the i-th subdomain, Ω_i, and on its traction-prescribed boundary, Γ_{t_i}. However, $\delta\boldsymbol{u}_{\Omega_i}$ vanishes in other subdomains, Ω_j $(j = 1, \ldots, i-1, i+1, \ldots, N)$, on their traction-prescribed boundaries, Γ_{t_j} $(j = 1, \ldots, i-1, i+1, \ldots, N)$, and on the subdomain interface, Γ. Then, $\delta\boldsymbol{u}_{\Gamma}$ is nonzero on the subdomain interface, Γ. Using Eq. (26), Eq. (1) can be decomposed into N equations, as

$$\int_{\Omega_i} \delta\boldsymbol{\varepsilon}_{\Omega_i}^{\mathrm{T}} \boldsymbol{\sigma} \mathrm{d}\Omega - \int_{\Gamma_{t_i}} \delta\boldsymbol{u}_{\Omega_i}^{\mathrm{T}} \boldsymbol{t} \mathrm{d}\Gamma - \int_{\Omega_i} \delta\boldsymbol{u}_{\Omega_i}^{\mathrm{T}} \boldsymbol{b} \mathrm{d}\Omega = 0 \quad (i = 1, \ldots, N). \tag{27}$$

These equations are then discretized by a shape function matrix, $\boldsymbol{N}$, and a strain–displacement matrix, $\boldsymbol{B}$, as

$$\boldsymbol{r}_{\Omega_i} = \int_{\Omega_i} \boldsymbol{B}^{\mathrm{T}} \boldsymbol{\sigma} \mathrm{d}\Omega - \int_{\Gamma_{t_i}} \boldsymbol{N}^{\mathrm{T}} \boldsymbol{t} \mathrm{d}\Gamma - \int_{\Omega_i} \boldsymbol{N}^{\mathrm{T}} \boldsymbol{b} \mathrm{d}\Omega = \boldsymbol{0} \quad (i = 1, \ldots, N). \tag{28}$$

Here, $\boldsymbol{r}_{\Omega_i}$ is a residual vector in the i-th subdomain, Ω_i. Equation (28) can be solved by subdomain-wise nonlinear finite analyses under assumed enforced displacements, $\boldsymbol{u}_{\Gamma}$, on the subdomain interface, Γ. As a result, displacements, $\boldsymbol{u}$, at whole degrees of freedom, $\Omega = \Omega_1 \cup \cdots \cup \Omega_N \cup \Gamma$, are obtained. Using these displacements, $\boldsymbol{u}$, the residual vector, $\boldsymbol{r}$, at whole degrees

of freedom in Eq. (3) can be evaluated. This residual vector is then permutated by a permutation matrix, $\boldsymbol{P}$, for subdomain internal degrees of freedom, Ω_i $(i = 1, \ldots, N)$, and subdomain interface degrees of freedom, Γ, as

$$\boldsymbol{P}\boldsymbol{r} = \begin{Bmatrix} \boldsymbol{r}_{\Omega_1} \\ \vdots \\ \boldsymbol{r}_{\Omega_N} \\ \boldsymbol{r}_{\Gamma} \end{Bmatrix}. \tag{29}$$

Since Eq. (28) has already been solved, $\boldsymbol{r}_{\Omega_i} = \boldsymbol{0}$ $(i = 1, \ldots, N)$ under the displacements, $\boldsymbol{u}$. Hence, in the present study, the nonlinear system of equations to be solved is defined as

$$\boldsymbol{r}_{\Gamma} = \boldsymbol{0}. \tag{30}$$

The unknown vector of this nonlinear system of equations is $\boldsymbol{u}_{\Gamma}$.

In this paragraph, it is briefly proved that Eq. (30) becomes equivalent to Eq. (13) when the problem is linearly elastic. The subdomain-wise residual vector (Eq. (28)) under enforced displacements, $\boldsymbol{u}_{\Gamma}$, on the subdomain interface, Γ, can be expressed as

$$\boldsymbol{r}_{\Omega_i} = \begin{bmatrix} \boldsymbol{K}_{\Omega_i\Omega_i} & \boldsymbol{K}_{\Omega_i\Gamma} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_{\Omega_i} \\ \boldsymbol{u}_{\Gamma} \end{Bmatrix} - \boldsymbol{f}^{\mathrm{ext}}_{\Omega_i} = \boldsymbol{0}. \tag{31}$$

In this equation, $\boldsymbol{u}_{\Omega_i}$ is an unknown displacement vector in the i-th subdomain. $\boldsymbol{f}^{\mathrm{ext}}_{\Omega_i} = \int_{\Gamma_{t_i}} \boldsymbol{N}^{\mathrm{T}}\boldsymbol{t}\mathrm{d}\Gamma + \int_{\Omega_i} \boldsymbol{N}^{\mathrm{T}}\boldsymbol{b}\mathrm{d}\Omega$ is an external force vector. $\boldsymbol{K}_{\Omega_i\Omega_i} = \int_{\Omega_i} \boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}\mathrm{d}\Omega$ is a stiffness matrix of subdomain internal degrees of freedom, Ω_i, and $\boldsymbol{K}_{\Omega_i\Gamma}$ is a stiffness matrix of subdomain interface degrees of freedom, Γ. Here, $\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{B}\boldsymbol{u}$, where $\boldsymbol{D}$ is an elasticity matrix. Note that, in Eq. (28), enforced displacements are not expressed explicitly, because $\delta\boldsymbol{u}_{\Omega_i}$ vanishes on Γ. Enforced displacements are processed algebraically. Then, Eq. (31) can be rewritten as

$$\boldsymbol{u}_{\Omega_i} = \boldsymbol{K}^{-1}_{\Omega_i\Omega_i} \left(\boldsymbol{f}^{\mathrm{ext}}_{\Omega_i} - \boldsymbol{K}_{\Omega_i\Gamma}\boldsymbol{u}_{\Gamma} \right). \tag{32}$$

Using this equation, the interface residual vector, $\boldsymbol{r}_{\Gamma}$, can be evaluated as

$$\begin{aligned} \boldsymbol{r}_{\Gamma} &= \begin{bmatrix} \boldsymbol{K}^{\mathrm{T}}_{\Omega_1\Gamma} & \cdots & \boldsymbol{K}^{\mathrm{T}}_{\Omega_N\Gamma} & \boldsymbol{K}_{\Gamma\Gamma} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_{\Omega_1} \\ \vdots \\ \boldsymbol{u}_{\Omega_N} \\ \boldsymbol{u}_{\Gamma} \end{Bmatrix} - \boldsymbol{f}^{\mathrm{ext}}_{\Gamma} \\ &= \sum_{i=1}^{N} \boldsymbol{K}^{\mathrm{T}}_{\Omega_i\Gamma}\boldsymbol{u}_{\Omega_i} + \boldsymbol{K}_{\Gamma\Gamma}\boldsymbol{u}_{\Gamma} - \boldsymbol{f}^{\mathrm{ext}}_{\Gamma} \\ &= \boldsymbol{S}\boldsymbol{u}_{\Gamma} - \boldsymbol{g} \\ &= \boldsymbol{0}. \end{aligned} \tag{33}$$

Here,

$$\boldsymbol{S} = \boldsymbol{K}_{\Gamma\Gamma} - \sum_{i=1}^{N} \boldsymbol{K}^{\mathrm{T}}_{\Omega_i\Gamma}\boldsymbol{K}^{-1}_{\Omega_i\Omega_i}\boldsymbol{K}_{\Omega_i\Gamma} \tag{34}$$

and

$$\boldsymbol{g} = \boldsymbol{f}^{\mathrm{ext}}_{\Gamma} - \sum_{i=1}^{N} \boldsymbol{K}^{\mathrm{T}}_{\Omega_i\Gamma}\boldsymbol{K}^{-1}_{\Omega_i\Omega_i}\boldsymbol{f}^{\mathrm{ext}}_{\Omega_i}. \tag{35}$$

Therefore, Eq. (30) becomes equivalent to Eq. (13) when the problem is linearly elastic.

3.2 Balancing domain decomposition preconditioning

To accelerate convergence and ensure scalability in large-scale analysis, a coarse-grid-correction-based preconditioner, such as BDD [25] or BDD-DIAG [33], is applied to the proposed DDM. In addition, a diagonal scaling preconditioner is applied for comparison. Although BDD and BDD-DIAG have been used for linear systems of equations in previous studies, they are applied to a nonlinear system of equations in the present study. The reason why BDD and BDD-DIAG can be applied to a nonlinear system of equations is its equivalence to Eq. (13), which was proved in the previous subsection. Here, a preconditioning matrix, $\boldsymbol{M}$, operates on the interface residual vector, $\boldsymbol{r}_{\Gamma}$, in Eq. (30), as

$$\boldsymbol{z}_{\Gamma} = \boldsymbol{M}^{-1}\boldsymbol{r}_{\Gamma}, \tag{36}$$

where $\boldsymbol{z}_{\Gamma}$ is a preconditioned residual vector.

In nonlinear analysis, $\boldsymbol{M}_{\mathrm{BDD}}$ (Eq. (19)), $\boldsymbol{M}_{\mathrm{BDD\text{-}DIAG}}$ (Eq. (25)), and $\boldsymbol{M}_{\mathrm{DIAG}}$ (Eq. (17)) can change at each quasi-Newton iteration step, due to the change of the stiffness matrix. However, regenerating the preconditioning matrix at every quasi-Newton iteration step takes very long computational time. Thus, we generate the preconditioning matrix by assuming linear elastic deformation. Once a preconditioning matrix is generated, it remains throughout the quasi-Newton iteration.

3.3 Quasi-Newton-based solution

In the proposed DDM, the preconditioned interface nonlinear system of equations,

$$\boldsymbol{z}_{\Gamma} = \boldsymbol{0}, \tag{37}$$

is solved by a quasi-Newton method. A quasi-Newton method is adopted in order to avoid a double-loop iteration algorithm and because of the consideration of the local nonlinearity. Moreover, the Jacobian matrix, $\frac{\partial \boldsymbol{z}_{\Gamma}}{\partial \boldsymbol{u}_{\Gamma}}$,

cannot be computed explicitly. Among various quasi-Newton methods, the Broyden method, as well as the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method, are used in the present study due to their popularity. Since a pure quasi-Newton method requires a double-loop iteration algorithm due to linear system solution, limited-memory implementation [21] with the identity matrix as an initial Jacobian matrix is applied.

In a quasi-Newton method, the unknown displacement vector, $\boldsymbol{u}$, is updated at the k-th iteration step, as

$$\boldsymbol{u}_\Gamma^{(k+1)} = \boldsymbol{u}_\Gamma^{(k)} + \Delta\boldsymbol{u}_\Gamma^{(k)}, \tag{38}$$

where

$$\Delta\boldsymbol{u}_\Gamma^{(k)} = -{\boldsymbol{J}^{(k)}}^{-1}\boldsymbol{z}_\Gamma^{(k)}. \tag{39}$$

$\boldsymbol{J}$ is an approximate Jacobian matrix, i.e.,

$$\boldsymbol{J} \approx \frac{\partial \boldsymbol{z}_\Gamma}{\partial \boldsymbol{u}_\Gamma}. \tag{40}$$

By the Broyden method, the approximate Jacobian matrix, $\boldsymbol{J}$, is updated as

$$\boldsymbol{J}^{(k+1)} = \boldsymbol{J}^{(k)} + \frac{\boldsymbol{z}_\Gamma^{(k)} {\Delta\boldsymbol{u}_\Gamma^{(k)}}^{\mathrm{T}}}{\left\|\Delta\boldsymbol{u}_\Gamma^{(k)}\right\|^2}. \tag{41}$$

Since solving Eq. (39) by a linear iterative solver requires a double-loop iteration algorithm, limited-memory implementation [21] is used. In this implementation, only several vector operations are required rather than linear system solution. Here, the Sherman–Morrison formula is introduced as

$$\left(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}\right)^{-1} = \left(\boldsymbol{I} - \frac{\boldsymbol{A}^{-1}\boldsymbol{u}}{1 + \boldsymbol{v}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{u}}\boldsymbol{v}^{\mathrm{T}}\right)\boldsymbol{A}^{-1}, \tag{42}$$

where $\boldsymbol{A}$ is an arbitrary square matrix, and $\boldsymbol{u}$ and $\boldsymbol{v}$ are arbitrary vectors. Moreover, $\boldsymbol{I}$ is the identity matrix. By applying this formula to Eq. (41), we can obtain two recursive expressions:

$$\Delta\boldsymbol{u}_\Gamma^{(k)} = -{\boldsymbol{J}^{(k)}}^{-1}\boldsymbol{z}_\Gamma^{(k)} = \frac{\boldsymbol{p}^{(k,k-1)}}{1 - \frac{{\Delta\boldsymbol{u}_\Gamma^{(k-1)}}^{\mathrm{T}}\boldsymbol{p}^{(k,k-1)}}{\left\|\Delta\boldsymbol{u}_\Gamma^{(k-1)}\right\|^2}}, \tag{43}$$

$$\boldsymbol{p}^{(k,i+1)} = -{\boldsymbol{J}^{(i+1)}}^{-1}\boldsymbol{z}_\Gamma^{(k)} = \boldsymbol{p}^{(k,i)} + \frac{{\Delta\boldsymbol{u}_\Gamma^{(i)}}^{\mathrm{T}}\boldsymbol{p}^{(k,i)}}{\left\|\Delta\boldsymbol{u}_\Gamma^{(i)}\right\|^2}\Delta\boldsymbol{u}_\Gamma^{(i+1)}. \tag{44}$$

Their initial values are

$$\Delta\boldsymbol{u}_\Gamma^{(0)} = -\boldsymbol{z}_\Gamma^{(0)} \tag{45}$$

and

$$\boldsymbol{p}^{(k,0)} = -\boldsymbol{z}_\Gamma^{(k)}. \tag{46}$$

The initial value of the approximate Jacobian matrix, $\boldsymbol{J}^{(0)}$, is assumed to be the identity matrix. In the limited-memory Broyden method, $\Delta\boldsymbol{u}_\Gamma^{(i)}$ $(i = 0, \ldots, k)$ should be stored in the memory.

Similarly, by the BFGS method, the approximate Jacobian matrix, $\boldsymbol{J}$, is updated as

$$\boldsymbol{J}^{(k+1)} = \boldsymbol{J}^{(k)} - \frac{\boldsymbol{J}^{(k)}\Delta\boldsymbol{u}_\Gamma^{(k)}\left(\boldsymbol{J}^{(k)}\Delta\boldsymbol{u}_\Gamma^{(k)}\right)^{\mathrm{T}}}{{\Delta\boldsymbol{u}_\Gamma^{(k)}}^{\mathrm{T}}\boldsymbol{J}^{(k)}\Delta\boldsymbol{u}_\Gamma^{(k)}} + \frac{\boldsymbol{y}^{(k)}{\boldsymbol{y}^{(k)}}^{\mathrm{T}}}{{\Delta\boldsymbol{u}_\Gamma^{(k)}}^{\mathrm{T}}\boldsymbol{y}^{(k)}}, \tag{47}$$

where

$$\boldsymbol{y}^{(k)} = \boldsymbol{z}_\Gamma^{(k)} - \boldsymbol{z}_\Gamma^{(k-1)}. \tag{48}$$

By applying Eq. (42) to Eq. (47), we can obtain three recursive expressions:

$$\Delta\boldsymbol{u}_\Gamma^{(k)} = -\boldsymbol{q}^{(k,k-1)}, \tag{49}$$

$$\boldsymbol{q}^{(k,i+1)} = \boldsymbol{q}^{(k,i)} + \left(\gamma^{(i)} - \beta^{(i)}\right)\Delta\boldsymbol{u}_\Gamma^{(i)}, \tag{50}$$

$$\boldsymbol{p}^{(k,i)} = \boldsymbol{p}^{(k,i+1)} - \gamma^{(i)}\boldsymbol{y}^{(i+1)}. \tag{51}$$

In these equations,

$$\beta^{(i)} = \frac{{\boldsymbol{y}^{(i+1)}}^{\mathrm{T}}\boldsymbol{q}^{(k,i)}}{{\boldsymbol{y}^{(i+1)}}^{\mathrm{T}}\Delta\boldsymbol{u}_\Gamma^{(i)}} \tag{52}$$

and

$$\gamma^{(i)} = \frac{{\Delta\boldsymbol{u}_\Gamma^{(i)}}^{\mathrm{T}}\boldsymbol{p}^{(k,i+1)}}{{\Delta\boldsymbol{u}_\Gamma^{(i)}}^{\mathrm{T}}\boldsymbol{y}^{(i+1)}}. \tag{53}$$

The initial values of $\Delta\boldsymbol{u}_\Gamma$, $\boldsymbol{q}$, and $\boldsymbol{p}$ are

$$\Delta\boldsymbol{u}_\Gamma^{(0)} = -\boldsymbol{z}_\Gamma^{(0)}, \tag{54}$$

$$\boldsymbol{q}^{(k,0)} = \boldsymbol{p}^{(k,0)}, \tag{55}$$

and

$$\boldsymbol{p}^{(k,k)} = \boldsymbol{z}_\Gamma^{(k)}. \tag{56}$$

In the limited-memory BFGS method, $\Delta\boldsymbol{u}_\Gamma^{(i)}$ $(i = 0, \ldots, k)$, as well as $\boldsymbol{y}^{(i)}$ $(i = 0, \ldots, k)$ and $\gamma^{(i)}$ $(i = 0, \ldots, k)$, should be stored in the memory.

3.4 Convergence criterion

A general convergence criterion of a quasi-Newton method is

$$\frac{\left\|\boldsymbol{z}_{\Gamma}^{(k)}\right\|}{\left\|\boldsymbol{z}_{\Gamma}^{(0)}\right\|} \leq \varepsilon. \tag{57}$$

The left-hand side is a relative residual, whereas the right-hand side, ε, is a tolerance. However, this sort of convergence criterion depends upon the selection of preconditioners, as well as the number of subdomains. Hence, we adopt a preconditioner- and subdomain-independent convergence criterion:

$$\frac{\left\|\boldsymbol{r}^{(k)}\right\|}{\left\|\boldsymbol{r}^{(0)}\right\|} = \frac{\left\|\boldsymbol{r}^{(k)}\right\|}{\left\|\boldsymbol{f}^{\mathrm{ext}}\right\|} \leq \varepsilon. \tag{58}$$

This convergence criterion has another advantage in that comparing the proposed DDM with the conventional DDM becomes fair. Note that, in the conventional DDM, we use $\frac{\left\|\boldsymbol{g}^{(k)}-\boldsymbol{S}^{(k)}\Delta\boldsymbol{u}_{\Gamma}^{(k,i)}\right\|}{\left\|\boldsymbol{g}^{(0)}\right\|}$ as a relative residual in the convergence criterion of inner-loop conjugate gradient iteration, because convergence is guaranteed by outer-loop Newton–Raphson iteration.

3.5 Algorithm

In the proposed DDM, a double-loop iteration algorithm, which is required in the conventional DDM (Fig. 2), is not required. The computational procedures of the proposed DDM are summarized in Fig. 4. First, subdomain-wise local problems (Eq. (28)) are solved in parallel by a nonlinear system solver such as the Newton–Raphson method, a modified Newton method, or a quasi-Newton method. Based on the results of the subdomain-wise local solutions, the interface residual vector is evaluated by Eq. (3). Note that, in this procedure, an iteration algorithm is required in subdomains that are in nonlinear deformation. Second, from the interface residual vector, the interface displacement vector is computed by preconditioning (Eq. (36)) and quasi-Newton-based updating (Eq. (38)). A part of these procedures, such as diagonal scaling preconditioning, Neumann–Neumann preconditioning, and some vector operations, can be parallelized. Finally, convergence is conformed using Eq. (58).

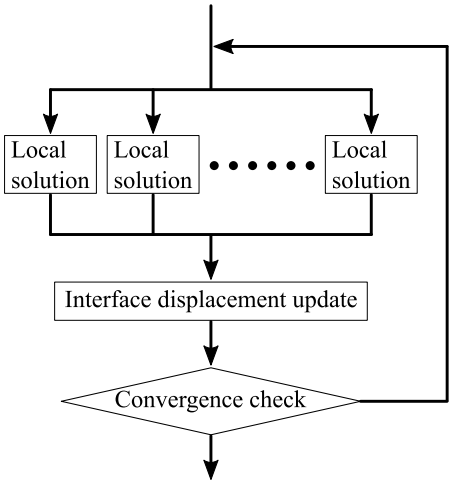

Fig. 4 Flowchart of the proposed domain decomposition method.

4 Numerical tests

Several numerical tests were performed in order to numerically demonstrate the effectiveness of the proposed DDM. First, in addition to performing a weak scaling test, linear elastic problems were analyzed in order to investigate the performance of the proposed DDM as a linear system solver. These numerical tests offer a basic understanding of convergence performance, as well as scalability. Second, elastic–plastic problems were analyzed in order to investigate the performance of the proposed DDM as a nonlinear system solver. These problems are more practical than linear elastic problems. In elastic–plastic analysis, some subdomains experience nonlinear elastic–plastic deformation, whereas other subdomains remain in linear elastic deformation. This feature is also demonstrated numerically.

4.1 Linear elastic problem

Before the investigation with nonlinear problems, a linear elastic problem was analyzed by the conventional and proposed DDMs. The problem involves a flat plate with a circular hole under a tensile load. In linear problems, the number of Newton–Raphson iteration steps of the conventional DDM is always one. Convergence histories of the conventional DDM (conjugate gradient method) and the proposed DDM (quasi-Newton method) are compared. Based on this numerical test, a basic understanding of the convergence performance of the proposed DDM can be obtained.

The dimensional parameters and boundary conditions of the problem are depicted in Fig. 5. The mesh with linear hexahedral elements is visualized in Fig. 6. The numbers of elements and nodes are 4,096 and 5,440, respectively. This mesh was decomposed by ADVENTURE_Metis [1, 40], which is based on the METIS and ParMETIS graph partitioning libraries [20]. The decomposed mesh is visualized in Fig. 7. The number of

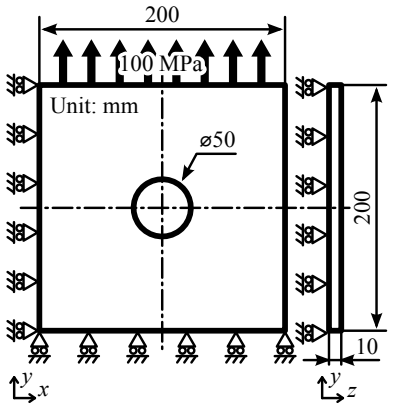

Fig. 5 Dimension parameters and boundary conditions of a plate with a hole.

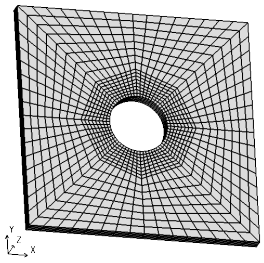

Fig. 6 Mesh of a plate with a hole.

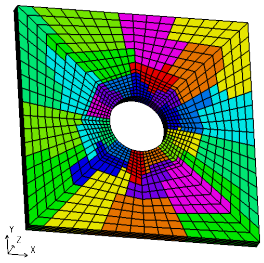

Fig. 7 Subdomains of a plate with a hole.

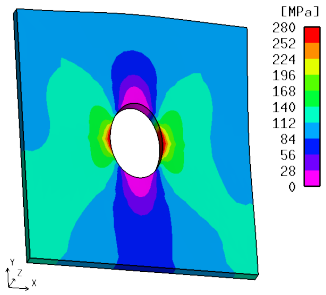

Fig. 8 Distribution of stress y with the deformation of a plate with a hole in linear elastic analysis.

subdomains is 32. Young's modulus and Poisson's ratio were set to be 200 GPa and 0.3, respectively. These are typical material constants of steel. The tolerance of the conjugate gradient method (conventional DDM), as well as the Broyden and BFGS methods (proposed DDM), was set to be 10^{-6}.

The distribution of stress y computed by the proposed DDM of the Broyden method with a BDD-DIAG preconditioner is visualized in Fig. 8. Deformation is magnified by 200 times. Stress concentration is observed in the vicinity of the hole.

Convergence histories of the conventional DDM of the conjugate gradient method with diagonal scaling

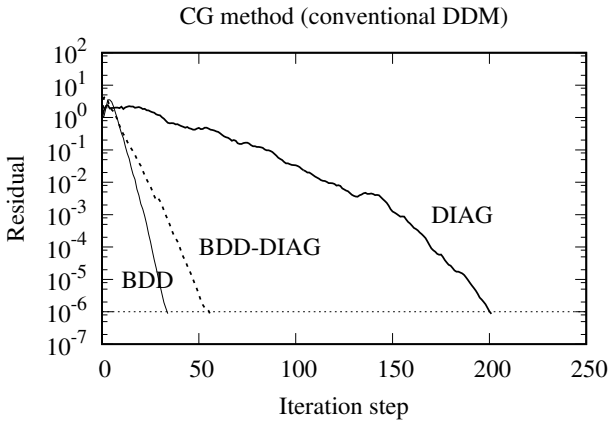

Fig. 9 Convergence histories of the conjugate gradient method (conventional DDM) of a hole problem in linear elastic analysis.

Table 1 Numbers of iteration steps of the conventional and proposed DDMs in linear elastic analysis.

	DIAG	BDD	BDD-DIAG
CG (conventional DDM)	201	34	56
Broyden (proposed DDM)	474	178	99
BFGS (proposed DDM)	241	244	96

(DIAG), BDD, and BDD-DIAG preconditioners are plotted in Fig. 9. Those of the proposed DDM of the Broyden method are plotted in Fig. 10, and those of the BFGS method are plotted in Fig. 11. In these figures, the horizontal axes represent the iteration step, whereas the vertical axes represent the relative residual. The numbers of iteration steps are summarized in Table 1. In the conjugate gradient method, the BDD preconditioner exhibits the best convergence performance. The BDD-DIAG preconditioner showed the second-best performance, and the diagonal scaling preconditioner showed the worst performance. This tendency is similar to that described by Ogino et al. [33]. However, in the Broyden and BFGS methods, a BDD preconditioner did not exhibit good convergence performance. The BDD preconditioner exhibited worse convergence performance than the BDD-DIAG preconditioner and diagonal scaling in the BFGS method. Thus, Neumann–Neumann preconditioning appears not to be effective in the proposed DDM. This would be affected by the computational procedure of the generalized inverse of the local Schur complement matrix. In contrast, with the BDD-DIAG preconditioner, the numbers of iteration steps of the Broyden and BFGS methods are only twice as large as that of the conjugate gradient method.

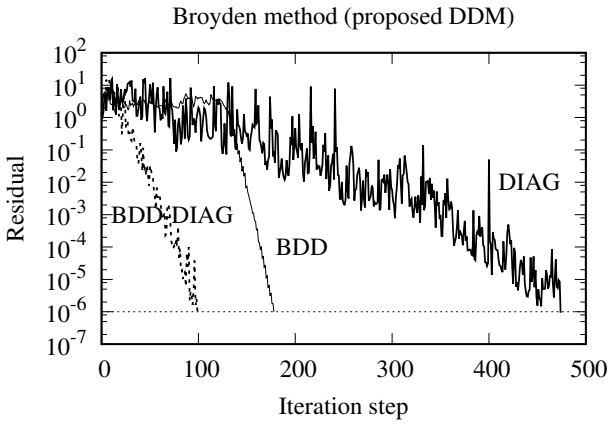

Fig. 10 Convergence histories of the Broyden method (proposed DDM) of a hole problem in linear elastic analysis.

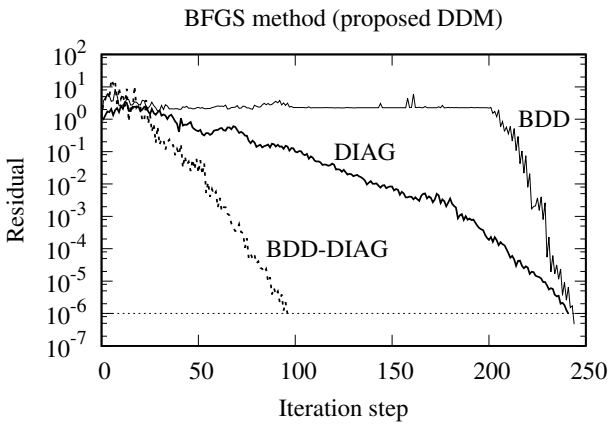

Fig. 11 Convergence histories of the BFGS method (proposed DDM) of a hole problem in linear elastic analysis.

4.2 Weak scaling test of the linear elastic problem

Generally, an iterative method for large-scale analysis should be scalable. In order to investigate the scalability of the proposed DDM for a linear elastic problem, a weak scaling test was performed. In this test, several problems, for which the numbers of elements and subdomains are varied, are analyzed. However, the number of elements of each subdomain remains approximately the same. Then, the number of iteration steps is carefully examined. If an iterative method is scalable, then the number of iteration steps remains approximately constant, even though the number of subdomains increases.

The problem involves a flat plate with multiple circular holes under a tensile load. Four problems were analyzed. Some dimensional parameters and boundary conditions of the problems are depicted in Fig. 12. Other dimensional parameters and the numbers of elements, nodes, and subdomains are described in Table 2. In these problems, the stress concentration should be observed in the vicinity of every hole. The mesh of the problem with 64 holes is shown in Fig. 13. This mesh was generated by mirroring the mesh of Fig. 6. This mesh was decomposed by ADVENTURE_Metis [1, 40] into 2,048 subdomains, as shown in Fig. 14. Note that the size of each subdomain slightly changes as the problem scales, because domain decomposition procedures depend upon the graph partitioning algorithm. Only if the size of subdomains per the size of elements remains, the BDD preconditioner ensures scalability [25].

The distribution of stress y computed by the proposed DDM of the Broyden method with a BDD-DIAG preconditioner is visualized in Fig. 15. Deformation is magnified by 200 times. Stress concentration is observed in the vicinity of every hole.

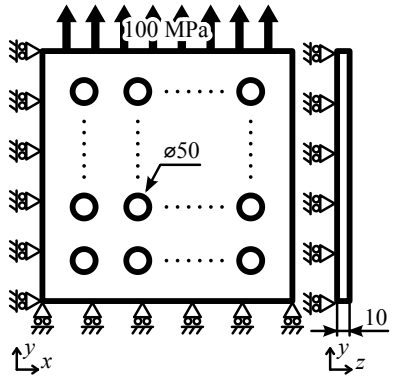

Fig. 12 Dimension parameters and boundary conditions of a plate with multiple holes.

Table 2 Parameters of the weak scaling test of multi-hole problems in linear elastic analysis.

	#1	#2	#3	#4
Plate size [mm]	200	400	800	1,600
# of holes	1	4	16	64
# of elements	4,096	16,384	65,536	262,144
# of nodes	5,440	21,425	85,045	338,885
# of subdomains	32	128	512	2,048

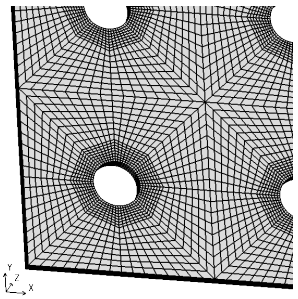

Fig. 13 Mesh of a plate with 64 holes.

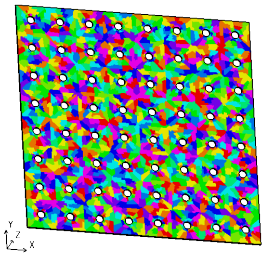

Fig. 14 Subdomains of a plate with 64 holes.

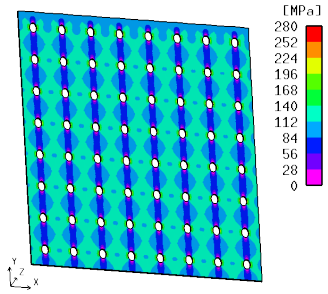

Fig. 15 Distribution of stress y with the deformation of a plate with 64 holes in linear elastic analysis.

The numbers of iteration steps of the conventional DDM (CG method) and the proposed DDM (Broyden and BFGS methods) are plotted in Fig. 16. An enlarged view is shown in Fig. 17. The horizontal axes represent the number of subdomains, whereas the vertical axes represent the number of iteration steps. Note that the proposed DDM with a BDD preconditioner is omitted because it did not show good convergence performance in the previous subsection. Actually, the proposed DDM could not achieve a converged solution in cases involving ≥ 512 subdomains. In these figures, the results of the conventional and proposed DDMs with a diagonal scaling (DIAG) preconditioner are not scalable. The number of iteration steps increases rapidly as the number of subdomains increases. Then, with the BDD-DIAG preconditioner, the conventional DDM, as well as the proposed DDM, provided almost scalable results. In addition, the conventional DDM with a BDD preconditioner showed a scalable result. The number of iteration steps of the Broyden and BFGS methods with a BDD-DIAG preconditioner is only two to three times larger than that of the CG method with a BDD-DIAG preconditioner. The proposed DDM is comparable to the conventional DDM in linear elastic analysis.

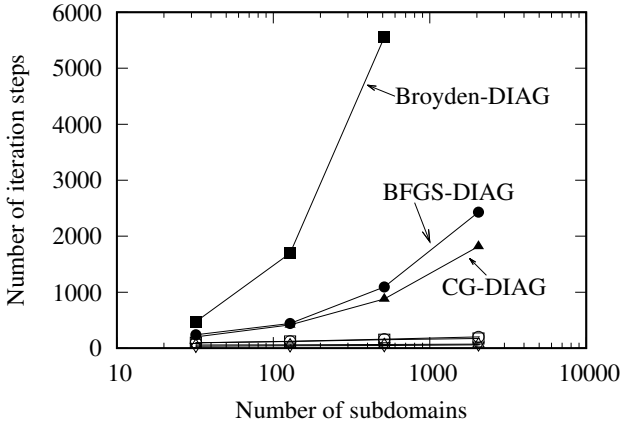

Fig. 16 Number of iteration steps of the conjugate gradient (CG), Broyden, and BFGS methods of multi-hole problems in linear elastic analysis.

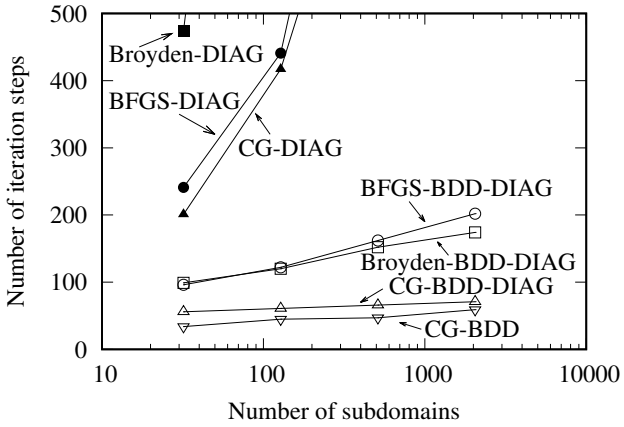

Fig. 17 Enlarged view of Fig. 16.

4.3 Elastic–plastic problem

An elastic–plastic problem was analyzed to investigate the performance of the proposed DDM in nonlinear analysis. In nonlinear analysis, the conventional DDM requires a double-loop algorithm of Newton–Raphson iteration and conjugate gradient iteration, as shown in Fig. 2, whereas the proposed DDM does not require such an algorithm. The proposed DDM uses a single-loop algorithm of quasi-Newton iteration, as shown in Fig. 4, even in nonlinear analysis.

The dimensional parameters and boundary conditions of the problem are the same as those in Section 4.1 and are depicted in Fig. 5. The mesh of the problem is also the same. The mesh and the decomposed mesh are shown in Figs. 6 and 7, respectively. As an elastic–plastic model, von Mises' yield criterion, an associated flow rule and linear isotropic hardening are used in this numerical test. Young's modulus and Poisson's ratio were set to be 200 GPa and 0.3, respectively. The initial yield stress and the hardening modulus were set to be 200 MPa and 20 GPa, respectively. These are respectively one thousandth and one tenth of Young's modulus, although real materials behave with a variety of initial yield stresses and hardening moduli.

In the conventional DDM, the tolerance of the Newton–Raphson method (outer iteration loop) and that of the conjugate gradient method (inner iteration loop) were set to be 10^{-6} and 10^{-7}, respectively. In order to obtain a converged solution in the outer iteration loop, the tolerance of the inner iteration loop should probably be smaller than that of the outer iteration loop. In contrast, in the proposed DDM, the tolerance of the Broyden and BFGS methods was set to be 10^{-6}. As

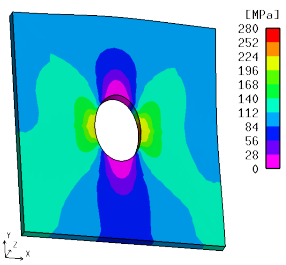

Fig. 18 Distribution of stress y with the deformation of a plate with a hole in elastic–plastic analysis.

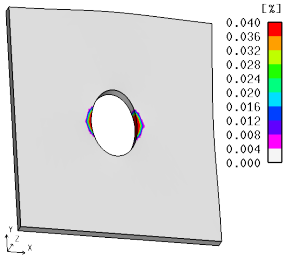

Fig. 19 Distribution of equivalent plastic strain with the deformation of a plate with a hole in elastic–plastic analysis.

the subdomain-wise local solver, the Broyden method was used, although any nonlinear system solver such as the Newton–Raphson method can be used in the methodology. The initial approximate Jacobian matrix of the Broyden method was a stiffness matrix of a linear elastic body. This matrix was factorized once by the LDL method before the global quasi-Newton iteration. Then, in the global quasi-Newton iteration, many forward and backward substitutions were performed. This sort of nonlinear system solver can be faster than the Newton–Raphson method [19]. The tolerance of this local solver was set to be 10^{-7}.

The distribution of stress y computed by the proposed DDM of the Broyden method with a BDD-DIAG preconditioner is visualized in Fig. 18. Deformation is magnified by 200 times. In this figure, stress concentration is observed. The distribution of equivalent plastic strain is shown in Fig. 19. Yielding zones are observed in the vicinity of the hole.

The convergence histories of the conventional DDM of the Newton–Raphson method are plotted in Fig. 20. The horizontal axis represents the Newton–Raphson iteration step, whereas the vertical axis represents the relative residual. Additionally, the results with a very small CG tolerance (10^{-15}) are plotted to numerically demonstrate potential quadratic convergence. Practically, a CG tolerance of 10^{-7} seems to be sufficiently small for a Newton–Raphson tolerance of 10^{-6}. For any preconditioners of the conjugate gradient method, the Newton–Raphson method required five iteration steps until convergence. At each Newton–Raphson iteration step, a linear system of equations was solved by the conjugate gradient method with a preconditioner. The convergence histories of the conjugate gradient method are plotted in Fig. 21. The horizontal axis represents the accumulated CG iteration step, whereas the vertical axis represents the interface relative residual. Moreover, the convergence histories of the proposed DDM of the Broyden method are plotted in Fig. 22, and those

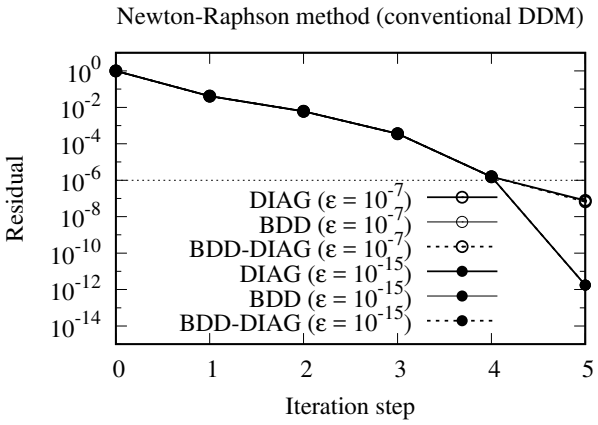

Fig. 20 Convergence histories of the Newton–Raphson method (conventional DDM) of a hole problem in elastic–plastic analysis.

Table 3 Numbers of iteration steps of the conventional and proposed DDMs in elastic–plastic analysis.

	DIAG	BDD	BDD-DIAG
CG (conventional DDM)	747	132	193
Broyden (proposed DDM)	641	NA	83
BFGS (proposed DDM)	206	NA	75

of the BFGS method are plotted in Fig. 23. The horizontal axes represent the quasi-Newton iteration step, whereas the vertical axes represent the relative residual. Note that the BDD preconditioner is omitted in the Broyden and BFGS methods, because the preconditioner did not show good convergence performance, even in the linear elastic analysis of Section 4.1. The numbers of iteration steps are summarized in Table 3. In the proposed DDM, the initial values of the quasi-Newton iteration were set to be the solutions in elastic deformation, due to the consideration of extrapolation-based initial value prediction in incremental analysis. Each number of iteration steps of the proposed DDM is smaller than that of the conventional DDM. The speedup from the conventional DDM to the proposed DDM with a BDD-DIAG preconditioner is between two and three times. These results are different from those of the linear elastic analysis in Section 4.1 because the conventional DDM requires Newton–Raphson iteration in nonlinear analysis. The number of iteration steps in the conventional DDM is larger than that for the linear elastic analysis in Table 1, whereas those of the proposed DDM remain approximately unchanged.

In the proposed DDM, subdomain-wise nonlinear finite element analyses are performed in parallel at each quasi-Newton iteration step. In order to demonstrate the suitability of the proposed DDM for the local nonlinearity, the number of linear system solutions of each

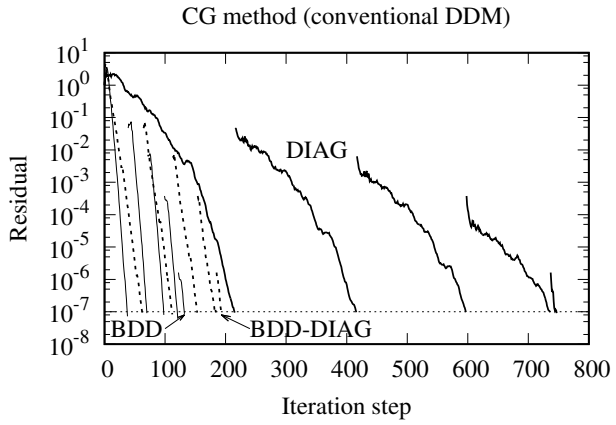

Fig. 21 Convergence histories of the conjugate gradient method (conventional DDM) of a hole problem in elastic–plastic analysis.

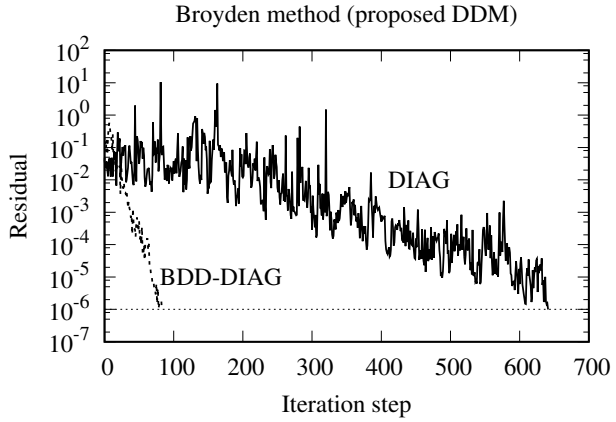

Fig. 22 Convergence histories of the Broyden method (proposed DDM) of a hole problem in elastic–plastic analysis.

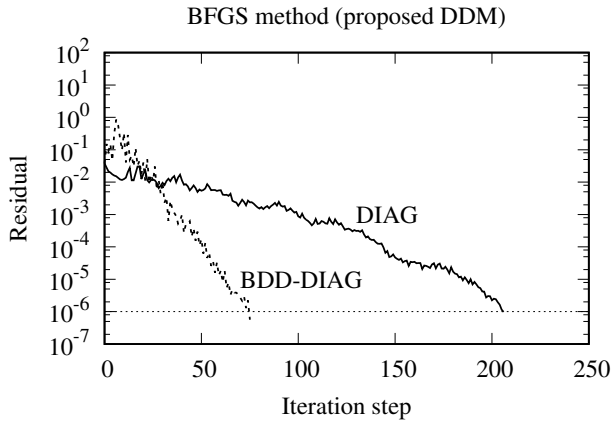

Fig. 23 Convergence histories of the BFGS method (proposed DDM) of a hole problem in elastic–plastic analysis.

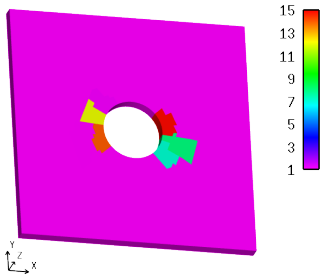

Fig. 24 Distribution of the average number of linear system solutions of each subdomain in elastic–plastic analysis.

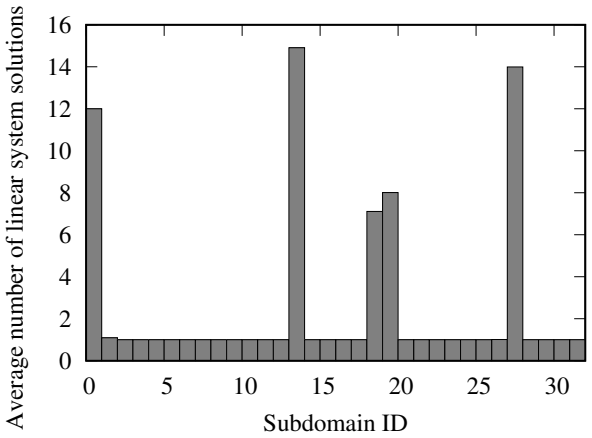

Fig. 25 Average number of linear system solutions of each subdomain in elastic–plastic analysis.

subdomain was investigated. The distribution of the average number of linear system solutions of each subdomain is shown in Fig. 24 and plotted in Fig. 25 as a bar graph. The horizontal axis represents the subdomain ID, whereas the vertical axis represents the total number of linear system solutions divided by the number of quasi-Newton iteration steps. This result was computed by the Broyden method with a BDD-DIAG preconditioner. The subdomain-wise local nonlinear system solver was also a Broyden method, the initial approximate Jacobian matrix of which was a stiffness matrix of a linear elastic body. As shown in Figs. 24 and 25, five subdomains in the vicinity of the hole experience elastic–plastic deformation with a large number of linear system solutions, whereas the other 27 subdomains remained approximately elastic throughout the quasi-Newton iteration.

4.4 Weak scaling test of the elastic–plastic problem

Similarly to Section 4.2, weak scaling in elastic–plastic analysis was investigated. In the weak scaling test, several problems for which the mesh is the same as that

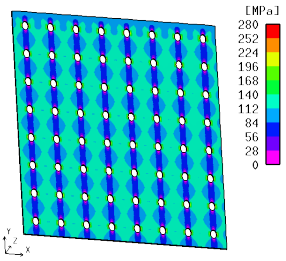

Fig. 26 Distribution of stress y with the deformation of a plate with 64 holes in elastic–plastic analysis.

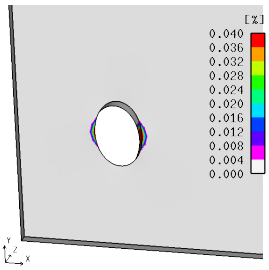

Fig. 27 Distribution of equivalent plastic strain with the deformation of a plate with 64 holes in elastic–plastic analysis.

in Section 4.2. Moreover, the number of subdomains in elastic–plastic deformation divided by the total number of subdomains was set to remain approximately the same as the problem scales.

The dimensional parameters, boundary conditions, and numbers of elements, nodes, and subdomains are the same as in Section 4.2 and are shown in Fig. 12 and Table 2. In these problems, the stress concentration, as well as yielding, should be observed in the vicinity of every hole. The meshes are also the same as those in Section 4.2. The mesh and the decomposed mesh with 64 holes are visualized in Figs. 13 and 14.

The distribution of stress y computed by the proposed DDM of the Broyden method with a BDD-DIAG preconditioner is visualized in Fig. 26, and that of equivalent plastic strain is visualized in Fig. 27. Deformation is magnified by 200 times. Stress concentration and yielding are observed in the vicinity of every hole.

The numbers of iteration steps of the conventional DDM (Newton–Raphson method with the CG method) and the proposed DDM (Broyden and BFGS methods) are plotted in Fig. 28. An enlarged view is shown in Fig. 29. The horizontal axes represent the number of subdomains, whereas the vertical axes represent the number of iteration steps. In these figures, the conventional DDM (Newton-CG) achieved converged solutions for any number of subdomains. Although the Broyden-DIAG, BFGS-DIAG, and BFGS-BDD-DIAG could not be converged in some problems, the Broyden-BDD-DIAG established good convergence performance. The number of iteration steps remained approximately half that of the Newton–Raphson method in conjunction with the CG method with a BDD-DIAG preconditioner. The proposed DDM of the Broyden method with a BDD-DIAG preconditioner was demonstrated to be scalable, and the number of iteration steps for the proposed DDM is smaller than that for the conventional DDM.

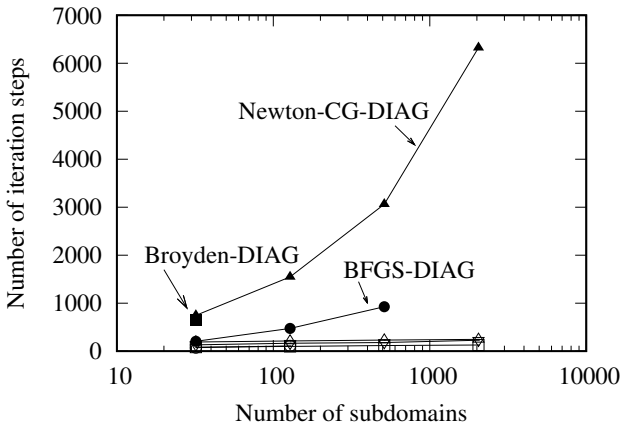

Fig. 28 Number of iteration steps of the Newton–Raphson method with the conjugate gradient (CG) method, the Broyden method, and the BFGS method of multi-hole problems in elastic–plastic analysis.

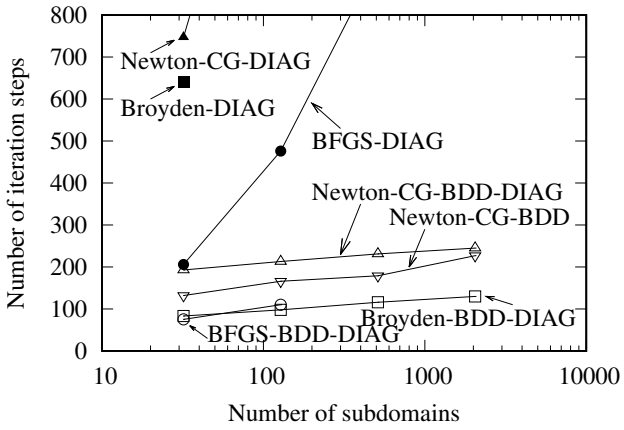

Fig. 29 Enlarged view of Fig. 28.

4.5 Elastic–plastic problem of a structural model

Finally, an elastic–plastic problem of a structural model was analyzed to demonstrate the capability of the proposed DDM for realistic complex-shape structural models. The dimensional parameters and boundary conditions of the structural model are depicted in Fig. 30. The structural model consists of a lower head of a pressure vessel, a nozzle, and a curved pipe. The skirt of the lower head is constrained, and a tensile load is prescribed on the end of the pipe. Stress concentration should occur at the nozzle, resulting in plastic deformation. The mesh near the nozzle with linear hexahedral elements is visualized in Fig. 31. A half model was used due to symmetry. The decomposed mesh is visualized in Fig. 32. The numbers of elements, nodes, and subdomains are 91,200, 129,991, and 256, respectively. Young's modulus and Poisson's ratio were set to be 200 GPa and 0.3, respectively. The stress–strain

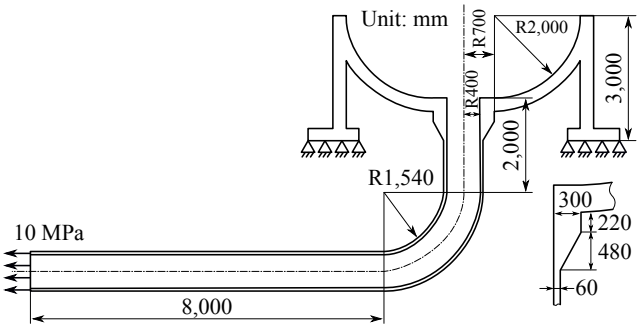

Fig. 30 Dimension parameters and boundary conditions of the structural model.

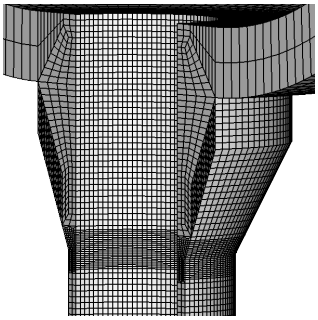

Fig. 31 Mesh near the nozzle of the structural model.

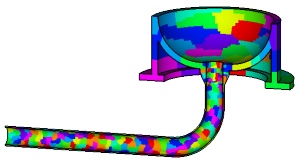

Fig. 32 Subdomains of the structural model.

curve adopted in this numerical test is

$$\sigma_y = \sigma_{y_0} + H\bar{\varepsilon}^{p^n}, \tag{59}$$

where σ_y is yield stress and $\bar{\varepsilon}^p$ is equivalent plastic strain. The initial yield stress, σ_{y_0}, the hardening modulus, H, and the hardening exponent, n, were set to be 200 MPa, 1,300 MPa, and 0.45, respectively. These material constants are typical values of stainless steel. In this numerical test, BDD and BDD-DIAG preconditioners were used in the conventional DDM, and a BDD-DIAG preconditioner was used in the proposed DDM. These three solvers established scalability in the previous subsection. The tolerances of the Newton–Raphson iteration (the conventional DDM) and the quasi-Newton iteration (the proposed DDM) were set to be 10^{-6}. Those of the conjugate gradient iteration (the conventional DDM) and the subdomain-wise local iteration (the proposed DDM) were set to be 10^{-7}. As the subdomain-wise local solver, the Broyden method was used.

The distribution of equivalent stress computed by the proposed DDM of the Broyden method with a BDD-DIAG preconditioner is visualized in Fig. 33. In this figure, stress concentration is observed at the left (compressive) and right (tensile) sides of the nozzle. The right side experiences plastic deformation, which is visualized in Fig. 34. In this figure, the distribution of equivalent plastic strain near the right side of the nozzle is visualized.

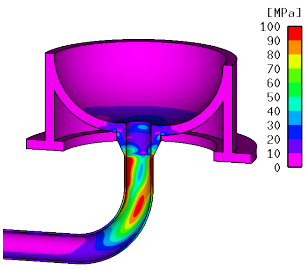

Fig. 33 Distribution of equivalent stress in the structural model analysis.

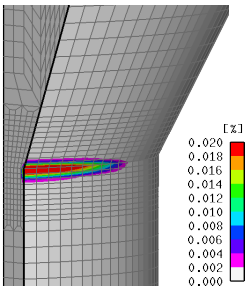

Fig. 34 Distribution of equivalent plastic strain near the nozzle in the structural model analysis.

The convergence histories of the conventional DDM of the Newton–Raphson method are plotted in Fig. 35. The horizontal axis represents the Newton–Raphson iteration step, whereas the vertical axis represents the relative residual. Four iteration steps were required until convergence. Thus, four linear systems of equations were solved by the conjugate gradient method with a BDD or BDD-DIAG preconditioner. The convergence histories of the conjugate gradient method are plotted in Fig. 36. The horizontal axis represents the accumulated CG iteration step, whereas the vertical axis represents the interface relative residual. The numbers of accumulated CG iteration steps of BDD and BDD-DIAG preconditioners were 217 and 333, respectively. Moreover, the convergence histories of the proposed DDM of the Broyden method are plotted in Fig. 37. The horizontal axis represents the quasi-Newton iteration step, whereas the vertical axis represents the residual norm. The number of quasi-Newton iteration steps was 327. Therefore, the proposed method exhibited comparable convergence performance to the conventional DDM in an elastic–plastic analysis on a realistic structural model.

5 Conclusion

In the present study, a domain decomposition method for large-scale elastic–plastic problems was proposed. Unlike conventional DDMs, the proposed method can consider the local concentration of nonlinear deformation, which is observed in elastic–plastic problems. In order to consider the local nonlinearity, we derived a nonlinear system of equations of the subdomain interface degrees of freedom from the principle of virtual work. This nonlinear system of equations is equivalent in linear elastic deformation to the statically condensed linear system of equations of the conventional DDM. In order to solve the interface nonlinear system of equations, a quasi-Newton method such as the

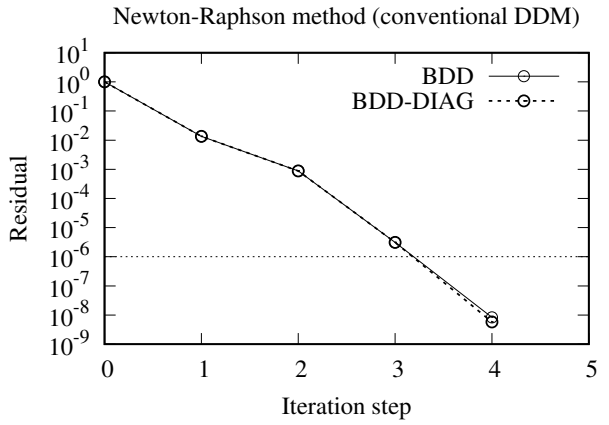

Fig. 35 Convergence histories of the Newton–Raphson method (conventional DDM) in the structural model analysis.

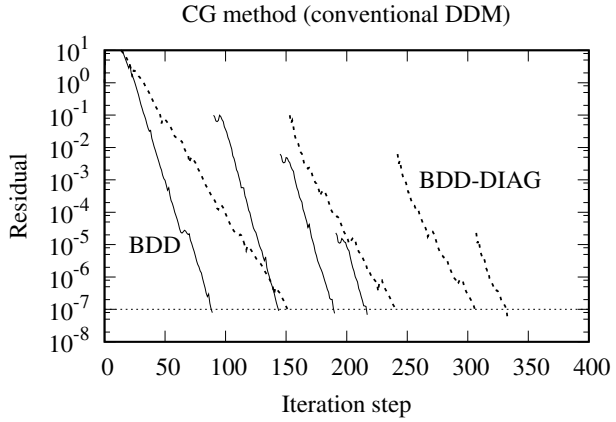

Fig. 36 Convergence histories of the conjugate gradient method (conventional DDM) in the structural model analysis.

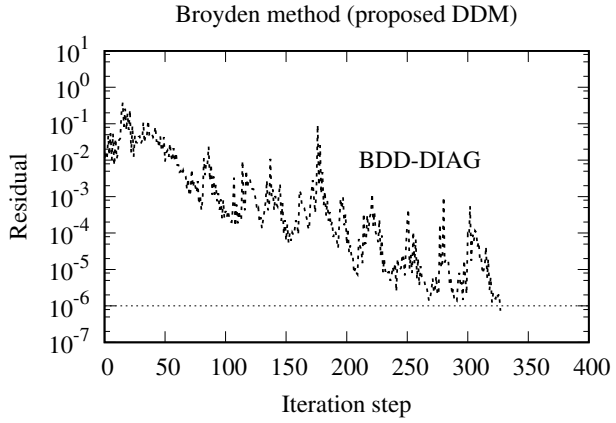

Fig. 37 Convergence histories of the Broyden method (proposed DDM) in the structural model analysis.

Broyden method or the BFGS method was used. At each quasi-Newton iteration step, subdomain-wise nonlinear finite element analyses are performed in parallel. If a subdomain is in linear elastic deformation, the number of subdomain-wise nonlinear iteration steps becomes one. Thus, each subdomain automatically selects its deformation to be linearly elastic or elastic–plastic and invests a number of nonlinear iteration steps that is optimal for its nonlinearity. Moreover, the use of a quasi-Newton method rather than the Newton–Raphson method offers another advantage, namely, the avoidance of a double-loop iteration algorithm, which is required in the standard Newton–Raphson method with a linear iterative solver. In general, a double-loop algorithm has large computational complexity. After that, a BDD preconditioner was applied to the quasi-Newton method in order to ensure scalability. In the literature, BDD preconditioners have been used in linear solution methods, whereas, in the present study, a BDD preconditioner is applied to a nonlinear solution method. Then, several numerical tests of linear elastic and elastic–plastic problems were performed in order to demonstrate the effectiveness of the proposed method. The results of these numerical tests provide a basic understanding of the convergence performance, as well as the scalability, of the proposed method. The convergence performance of the proposed method was comparable to that of the conventional method. In particular, in elastic–plastic analysis, the proposed method exhibited better convergence performance than the conventional method. Moreover, scalability with regard to the number of elements, as well as the number of subdomains, was demonstrated numerically by weak scaling tests of linear elastic and elastic–plastic problems.

In the future, numerical comparison on computational time should be performed after intensive performance tuning on the proposed method, as well as the conventional method. Although the convergence performance of the proposed method was better than that of the conventional method, the computational procedures in an iteration step are totally different. In addition, another problem remains to be overcome. The proposed method produces a significant load imbalance in parallel computing due to the consideration of local nonlinearity.

Acknowledgements The present study was supported in part by MEXT Post-K Project Priority Issue 6: *Accelerated Development of Innovative Clean Energy Systems* and by JSPS KAKENHI Grant Number JP16K05988. The authors would like to thank the members of the ADVENTURE Project for their helpful discussions.

References

1. ADVENTURE Project. URL http://adventure.sys.t.u-tokyo.ac.jp/
2. Akiba, H., Ohyama, T., Shibata, Y., Yuyama, K., Katai, Y., Takeuchi, R., Hoshino, T., Yoshimura, S., Noguchi, H., Gupta, M., Gunnels, J.A., Austel, V., Sabharwal, Y., Garg, R., Kato, S., Kawakami, T., Todokoro, S., Ikeda, J.: Large scale drop impact analysis of mobile phone using ADVC on Blue Gene/L. In: Proc 2006 ACM/IEEE Conf Supercomput, pp. 1–26 (2006)
3. An, H.B.: On convergence of the additive Schwarz preconditioned inexact Newton method. SIAM J Numer Anal **43**(5), 1850–1871 (2005)
4. Badea, L.: On the Schwarz alternating method with more than two subdomains for nonlinear monotone problems. SIAM J Numer Anal **28**(1), 179–204 (1991)
5. Bathe, K.J., Cimento, A.P.: Some practical procedures for the solution of nonlinear finite element equations. Comput Methods Appl Mech Eng **22**(1), 59–85 (1980)
6. Bhardwaj, M., Day, D., Farhat, C., Lesoinne, M., Pierson, K., Rixen, D.: Application of the FETI method to ASCI problems—scalability results on 1000 processors and discussion of highly heterogeneous problems. Int J Numer Methods Eng **47**(1-3), 513–535 (2000)
7. Bhardwaj, M., Pierson, K., Reese, G., Walsh, T., Day, D., Alvin, K., Peery, J., Farhat, C., Lesoinne, M.: Salinas: A scalable software for high-performance structural and solid mechanics simulations. In: Proc 2002 ACM/IEEE Conf Supercomput, pp. 1–19 (2002)
8. Cai, X.C., Keyes, D.E.: Nonlinearly preconditioned inexact Newton algorithms. SIAM J Sci Comput **24**(1), 183–200 (2002)
9. Crisfield, M.: A faster modified Newton–Raphson iteration. Comput Methods Appl Mech Eng **20**(3), 267–278 (1979)
10. Degroote, J., Bathe, K.J., Vierendeels, J.: Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction. Comput Struct **87**(11–12), 793–801 (2009)
11. Dohrmann, C.: A preconditioner for substructuring based on constrained energy minimization. SIAM J Sci. Comput **25**(1), 246–258 (2003)
12. Dolean, V., Gander, M.J., Kheriji, W., Kwok, F., Masson, R.: Nonlinear preconditioning: How to use a nonlinear Schwarz method to precondition Newton's method. SIAM J Sci Comput **38**(6), A3357–A3380 (2016)
13. Dryja, M., Hackbusch, W.: On the nonlinear domain decomposition method. BIT Numer Math **37**(2), 296–311 (1997)
14. Farhat, C., Lesoinne, M., LeTallec, P., Pierson, K., Rixen, D.: FETI-DP: a dual–primal unified FETI method—part I: A faster alternative to the two-level FETI method. Int J Numer Methods Eng **50**(7), 1523–1544 (2001)
15. Farhat, C., Roux, F.X.: A method of finite element tearing and interconnecting and its parallel solution algorithm. Int J Numer Methods Eng **32**(6), 1205–1227 (1991)
16. Fish, J.: The s-version of the finite element method. Comput Struct **43**(3), 539–547 (1992)
17. Geradin, M., Idelsohn, S., Hogge, M.: Computational strategies for the solution of large nonlinear problems via quasi-Newton methods. Comput Struct **13**(1), 73–81 (1981)
18. Gosselet, P., Rey, C.: Non-overlapping domain decomposition methods in structural mechanics. Arch Comput Methods Eng **13**(4), 515–572 (2006)
19. Hisada, T., Noguchi, H.: Foundation and Application of Nonlinear Finite Element Method (in Japanese). Maruzen Publishing (1995)
20. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J Sci Comput **20**(1), 359–392 (1998)
21. Kelley, C.T.: Solving Nonlinear Equations with Newton's Method. Society for Industrial and Applied Mathematics (2003)
22. Klawonn, A., Lanser, M., Rheinbach, O.: Nonlinear FETI-DP and BDDC methods. SIAM J Sci Comput **36**(2), A737–A765 (2014)
23. Klawonn, A., Lanser, M., Rheinbach, O., Uran, M.: Nonlinear FETI-DP and BDDC methods: A unified framework and parallel results. SIAM J Sci Comput **39**(6), C417–C451 (2017)
24. Lui, S.H.: On Schwarz alternating methods for nonlinear elliptic PDEs. SIAM J Sci Comput **21**(4), 1506–1523 (1999)
25. Mandel, J.: Balancing domain decomposition. Commun Numer Methods Eng **9**(3), 233–241 (1993)
26. Matthies, H., Strang, G.: The solution of nonlinear finite element equations. Int J Numer Methods Eng **14**(11), 1613–1626 (1979)
27. Minami, S., Yoshimura, S.: Performance evaluation of nonlinear algorithms with line-search for partitioned coupling techniques for fluid–structure interactions. Int J Numer Methods Fluids **64**(10–12), 1129–1147 (2010)
28. Miyamura, T., Noguchi, H., Shioya, R., Yoshimura, S., Yagawa, G.: Elastic–plastic analysis of nuclear structures with millions of DOFs using the hierarchical domain decomposition method. Nucl Eng Des **212**(1–3), 335–355 (2002)
29. Nayak, G.C., Zienkiewicz, O.C.: Note on the 'alpha'-constant stiffness method for the analysis of non-linear problems. Int J Numer Methods Eng **4**(4), 579–582 (1972)
30. Negrello, C., Gosselet, P., Rey, C., Pebrel, J.: Substructured formulations of nonlinear structure problems—influence of the interface condition. Int J Numer Methods Eng **107**(13), 1083–1105 (2016)
31. Nikishkov, G.P., Atluri, S.N.: An analytical–numerical alternating method for elastic–plastic analysis of cracks. Comput Mech **13**(6), 427–442 (1994)
32. Nishikawa, H., Serizawa, H., Murakawa, H.: Actual application of FEM to analysis of large scale mechanical problems in welding. Sci Tech Weld Join **12**(2), 147–152 (2007)
33. Ogino, M., Shioya, R., Kanayama, H.: An inexact balancing preconditioner for large-scale structural analysis. J Comput Sci Tech **2**(1), 150–161 (2008)
34. Ogino, M., Shioya, R., Kawai, H., Yoshimura, S.: Seismic response analysis of nuclear pressure vessel model with ADVENTURE System on the Earth Simulator. J Earth Simul **2**, 41–54 (2005)
35. Pebrel, J., Rey, C., Gosselet, P.: A nonlinear dual-domain decomposition method: Application to structural problems with damage. Int J Multiscale Comput Eng **6**(3), 251–262 (2008)
36. Pyo, C.R., Okada, H., Atluri, S.N.: An elastic–plastic finite element alternating method for analyzing widespread fatigue damage in aircraft structures. Comput Mech **16**(1), 62–68 (1995)
37. Smith, B., Bjørstad, P., Gropp, W.: Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations. Cambridge University Press (2004)

38. Toselli, A., Widlund, O.: Domain Decomposition Methods: Algorithms and Theory. Springer (2004)
39. Xu, J., Zou, J.: Some nonoverlapping domain decomposition methods. SIAM Rev **40**(4), 857–914 (1998)
40. Yoshimura, S., Shioya, R., Noguchi, H., Miyamura, T.: Advanced general-purpose computational mechanics system for large-scale analysis and design. J Comput Appl Math **149**(1), 279–296 (2002)
41. Yumoto, Y., Yusa, Y., Okada, H.: Element subdivision technique for coupling-matrix-free iterative s-version FEM and investigation of sufficient element subdivision. Mech Eng J **3**(5), 16–00361 (2016)
42. Yumoto, Y., Yusa, Y., Okada, H.: An s-version finite element method without generation of coupling stiffness matrix by using iterative technique. Mech Eng J **3**(5), 16–00001 (2016)
43. Yusa, Y., Kataoka, S., Kawai, H., Yoshimura, S.: Large-scale fracture mechanics analysis using partitioned iterative coupling algorithm (in Japanese). Trans Jpn Soc Mech Eng Ser A **78**(791), 966–975 (2012)
44. Yusa, Y., Okada, H., Yumoto, Y.: Three-dimensional elastic analysis of a structure with holes using accelerated coupling-matrix-free iterative s-version FEM. Int J Comput Methods **in press**, 1850036 (35 pages)
45. Yusa, Y., Yoshimura, S.: Mixed-mode fracture mechanics analysis of large-scale cracked structures using partitioned iterative coupling method. Comput Model Eng Sci **91**(6), 445–461 (2013)
46. Yusa, Y., Yoshimura, S.: Speedup of elastic–plastic analysis of large-scale model with crack using partitioned coupling method with subcycling technique. Comput Model Eng Sci **99**(1), 87–104 (2014)