Journal of Computational Science 24 (2018) 24–33

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Parallel computation for three-dimensional shell analysis of curved configuration based on domain decomposition method

HyunShig Joo[a], Haeseong Cho[b], Seil Kim[a], SangJoon Shin[c,*], JunYoung Kwak[d]

[a] Department of Mechanical and Aerospace Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, 151-744, Republic of Korea
[b] BK21 Plus Transformative Training Program for Creative Mechanical and Aerospace Engineers, Institute of Advanced Machines and Design, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, 151-744, Republic of Korea
[c] Department of Mechanical and Aerospace Engineering, Institute of Advanced Aerospace Technology, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, 151-744, Republic of Korea
[d] Korea Aerospace Research Institute 169-84, Gwahak-ro, Yuseong-Gu, Daejeon 34133, Republic of Korea

ARTICLE INFO

Article history:
Received 23 June 2017
Received in revised form 5 November 2017
Accepted 11 November 2017

Keywords:
Finite element tearing and interconnecting (FETI)
Lagrange multiplier
Shell element
Parallel algorithm

ABSTRACT

In this paper, a parallel computational algorithm is developed based on finite element tearing and interconnecting (FETI) method, specifically, localized Lagrange multipliers. The proposed FETI method decomposes large-size structures into non-overlapping subdomains via localized Lagrange multipliers. To consider the curved configuration of large-size structures, a facet shell element created by combining an optimal triangle membrane and discrete Kirchhoff triangle bending plate (OPT-DKT) is suggested and used by introducing rotational operators. Moreover, practical performance of the present OPT-DKT facet shell element is evaluated through static and dynamic analysis. Finally, parallel computation is implemented for the proposed approach using message passing interface (MPI).

1. Introduction

With the enhancement of computational hardware and software over the past few decades, it became more and more capable of conducting large-size computational analysis for complex configurations. As the number of elements and grids increases, computational costs will also increase, such as hundreds of hours for computational time and gigabytes of memory usage. To relieve such increased time and memory usage, the domain decomposition techniques were suggested. The main purpose of that approach was to divide the entire domain into the number of smaller domains so that each domain might be processed in parallel environment. The domain decomposition method can be classified into the overlapping and non-overlapping methods. In the over-lapping method, sub-domains on the interface regions and unknown variables are analyzed by iterative methods while prescribed by Dirichlet boundary conditions. In the non-overlapping methods, however, sub-domains on the interfaces with Lagrange multipliers are solved by either iterative or direct solvers depending on the condition number of the relevant matrices [2,3]. One of the most successful non-overlapping methods is the finite element tearing and interconnecting (FETI) method. The original FETI method utilized a parallel algorithm for the second-order partial differential equations (PDEs) [5]. This method was further extended to consider the original second- and fourth-order PDEs [6,8]. The computational domains were decomposed into various non-overlapping subdomains, while Lagrange multipliers were used to enforce the compatibility of the displacements along the interconnecting subdomains. Recently, the dual-primal FETI (FETI-DP) method was developed that made it feasible to obtain a standard preconditioned conjugate gradient algorithm (PCG), which was not used in the original FETI method [7]. The basic idea of FETI-DP method is to introduce Lagrange multipliers into the coarse nodes. By estimating a saddle-point of the Lagrangian functional, the resulting equation can be solved by iterative methods. But both original FETI and FETI-DP method require a good preconditioner related with iterative methods to solve the interface problem, which may degrade efficiency of the entire solution procedure.

In flexible multibody dynamic formulations, finite element analysis with Lagrange multipliers has been used [1]. Lagrange multipliers were used to enforce various kinematic constraints among the multiple bodies. To solve the resulting nonlinear problems, augmented Lagrange formulation (ALF) was established to enhance the flexibility matrix conditioning. In addition, localized

* Corresponding author.
E-mail addresses: joohyunshig@snu.ac.kr (H. Joo), nicejjo@snu.ac.kr (H. Cho), seill49kim@snu.ac.kr (S. Kim), ssjoon@snu.ac.kr (S. Shin), kjy84@kari.re.kr (J. Kwak).

https://doi.org/10.1016/j.jocs.2017.11.007
1877-7503/

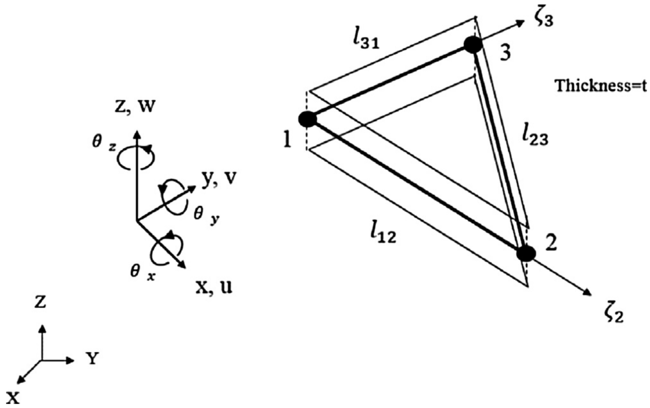

Fig. 1. Geometry of the present OPT-DKT shell element.

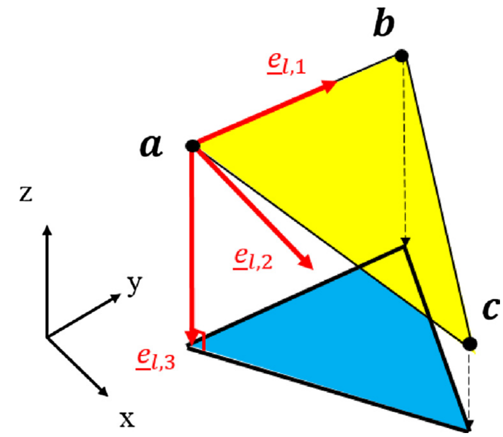

Fig. 2. Re-defined new three-nodes in x-y plane by using rotational operator.

Lagrange multipliers were adopted as penalty formulations to enhance the solution accuracy. Both serial and parallel algorithms for Lagrange multipliers based on the FETI method were developed [17,18]. Moreover, the scalability was evaluated using the localized Lagrange multipliers applied to both large-size structural analysis and flexible multibody dynamics [14,15]. The localized Lagrange multipliers as penalty term improve the condition number of the matrices to approach unity, which allows the direct solver to be applicable. Effectiveness of the localized Lagrange multipliers was already proved rigorously in Ref. 17 and the present authors demonstrated numerically for the case of the static, two-dimensional plane stress structure situation in [15].

In this study, a three-dimensional OPT-DKT shell element is used to analyze the curved configuration of large-size structure [4,9,10,12,19]. A rotational operator is used to obtain elemental stiffness matrix and the internal load vector for the curved configuration [11]. FETI-type algorithm (hereafter referred to as FETI-local) is also proposed by adopting localized Lagrange multipliers to enhance the compatibility of displacements along the boundary nodes in each subdomain. Isotropic/anisotropic property along with the thin-wall configuration is used. All the localized Lagrange multipliers are additionally defined in each boundary node along the interconnecting subdomain. All the penalty terms are equally applied as the spring coefficients for the multiplication results of Lagrange multipliers, while the penalty term affects reaction forces in each subdomain. Validation of the present OPT-DKT shell elements is completed by comparing with the commercial software in terms of both static and dynamic analysis. Because of the improved condition number for the matrices, it becomes possible to use a parallel sparse direct solver library, such as PARDISO to solve the issues of the sparse stiffness matrices induced by the shell elements. Finally, the present FETI-local methodology is implemented in parallel computing environments based on FORTRAN 90 with the message passing interface (MPI), to achieve similar results of reduced computational time and memory usage.

2. OPT-DKT shell elements

An OPT-DKT shell element is created by combining an OPT membrane element and a DKT plate bending element. Each element exhibits eighteen degrees-of-freedom; moreover, each element is divided into three nodes, each exhibiting six degrees-of-freedom (three translations and three rotations).

2.1. Definition of the geometric parameters

Fig. 1 illustrates the present OPT-DKT shell element geometry. The flat triangular element (OPT) has three nodes with three degrees-of-freedom at each node. Global and local coordinate systems are denoted as (X, Y, Z) and (x, y, z), respectively. Based on the local coordinates $\left(x_{ij}, y_{ij}\right)$, the elemental geometric parameters length $\left(l_{ij}\right)$, area(A), and volume (V) are defined as

$$\left(x_{ij}, y_{ij}\right) = \left(x_i - x_j, y_i - y_j\right) \tag{1}$$

$$l_{ij} = \sqrt{x_{ij}^2 + y_{ij}^2} \tag{2}$$

$$A = \frac{1}{2}\left(y_{21}x_{13} - x_{21}y_{13}\right) \tag{3}$$

$$V = At \tag{4}$$

2.2. Rotational operators

For a curved configuration, it is necessary to define the local coordinates with respect to three orthogonal vectors: the *x-y* plane and its normal vector. In order to obtain the exact elemental stiffness matrix and internal load vector based on the shell element, it is necessary to use a rotational operator that adjusts the normal vector of out-of-plane with respect to the three nodes of the *x-y* plane at each element. Fig. 2 illustrates the result of projecting the configuration onto the *x-y* plane using the rotational operator, which defines new three-nodes in the local coordinates based on the physical three-nodes in the fixed coordinates, to exploit the conventional OPT-DKT shell formulation.

Let $\underline{e}_{l,i}$ denote the i^{th} axis vector on the shell element in the local coordinates. The rotational operator can be described as $\underline{\underline{r}} = \left\{ \underline{e}_{l,1} \quad \underline{e}_{l,2} \quad \underline{e}_{l,3} \right\}$. For the first axis vector on the shell element, $\underline{e}_{l,1}$ can be established by selecting the first basis vector $\underline{r}_{ba} = (X_b - X_a, Y_b - Y_a, Z_b - Z_a)$.

$$\underline{e}_{l,1} = \frac{\underline{r}_{ba}}{\|\underline{r}_{ba}\|} \tag{5}$$

To define new three-nodes on the *x-y* plane in the local coordinates, the third coordinate axis orthogonal to the *x-y* plane in the local coordinates can be obtained by using $\underline{r}_{ba}$ and $\underline{r}_{ca}$:

$$\underline{e}_{l,3} = \frac{\underline{r}_{ba} \times \underline{r}_{ca}}{\|\underline{r}_{ba} \times \underline{r}_{ca}\|} \tag{6}$$

The remaining coordinate axis is described as

$$\underline{e}_{l,2} = \underline{e}_{l,3} \times \underline{e}_{l,1} \tag{7}$$

By multiplying the rotational operator, newly defined three-nodes are always imposed on the *x-y* plane in the local coordinates.

$$\left\{\underline{x}\right\} = \underline{\underline{r}}\left\{\underline{X}\right\} \tag{8}$$