## Paper

Int'l J. of Aeronautical & Space Sci. 16(2), 177–189 (2015) DOI: http://dx.doi.org/10.5139/IJASS.2015.16.2.177



## Domain Decomposition Approach Applied for Two- and Three-dimensional Problems via Direct Solution Methodology

## Jun Young Kwak\*

Combustion Chamber Team, Korea Aerospace Research Institute, 169-84, Gwahak-ro, Yuseong-gu, Daejeon 305-806, Republic of Korea

## Haeseong Cho\*\*

Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-744, Republic of Korea

## Tae Young Chun\*\*\*

LIGNex1, 702, Sampyeong-dong, Bundang-gu, Seongnam-City, Gyeonggi-do 463-400, Republic of Korea

## SangJoon Shin\*\*\*\*

Department of Mechanical and Aerospace Engineering, Institute of Advanced Aerospace Technology, Seoul National University, Seoul 151-744, Republic of Korea

## Olivier A. Bauchau\*\*\*\*\*

Department of Mechanical and Aerospace Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong.

## Abstract

This paper presents an all-direct domain decomposition approach for large-scale structural analysis. The proposed approach achieves computational robustness and efficiency by enforcing the compatibility of the displacement field across the subdomain boundaries via local Lagrange multipliers and augmented Lagrangian formulation (ALF). The proposed domain decomposition approach was compared to the existing FETI approach in terms of the computational time and memory usage. The parallel implementation of the proposed algorithm was described in detail. Finally, a preliminary validation was attempted for the proposed approach, and the numerical results of two- and three-dimensional problems were compared to those obtained through a dual-primal FETI approach. The results indicate an improvement in the performance as a result of the implementing the proposed approach.

**Key words:** Finite Element Tearing and Interconnecting (FETI), Augmented Lagrangian Formulation, Domain Decomposition Method, Lagrange multipliers, Parallel computing

## 1. Introduction

Advances in computer hardware and software for computational fluids and structural dynamics allow for multi-physics problems and engineering problems of ever increasing size and impact, such as fluid-structure interactions, to be solved. Consequently, it is of increasing importance to develop effective methodologies to solve large-scale structural problems in mechanical and aerospace engineering.

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/bync/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

\* Senior Researcher, Ph.D

- \*\*\* Researcher, MS
- \*\*\*\* Professor, Corresponding author: ssjoon@snu.ac.kr

\*\*\*\* Professor

Received: February 16, 2015 Accepted: May 11, 2015 Copyright © The Korean Society for Aeronautical & Space Sciences

177

(cc)

http://ijass.org pISSN: 2093-274x eISSN: 2093-2480

<sup>\*\*</sup> Graduate Student

High-precision stress predictions require general finite element methods with a large number of elements, and the computational costs increase as a result of the increase in CPU time and memory size that are required. Practical problems may require a significant amount of computational time and gigabytes of memory, so it is important to carefully select the grid size and the solution methodology. Numerical algorithms that can be used in such large-scale problems may be classified into the groups that implement a direct approach, iterative approach, or a combination of both.

The active column solver is a direct solution approach that is most commonly used in commercial finite element software [1], and it is based on the tri-factorization of a global stiffness matrix. This approach is very efficient when multiple right-hand side vectors exist, which is common in structural analysis problems that involve multiple load cases. Typically, the factorized matrix is used for all iterations at a given time step during dynamic analysis, leading to an improvement in the computational efficiency. Over the last decade, the sparse factorization technique has been widely adopted for use in commercial finite element packages.

When these are implemented on parallel computers, the active column solver shows limited success in terms of its simplicity and scalability. This approach typically requires a much larger storage than iterative algorithms do. The operation count increases significantly faster as the problem size increases than for iterative solvers. Multi-frontal solvers [2, 3] have also been proposed for factorization. Although multi-frontal solvers allow a parallel solution for large-scale problems by using distributed memory, the approach does not exhibit an adequate scalability [4]. A direct sparse matrix solver [5] has been developed since the early 1990s and is based on explicit factorization, i.e., Cholesky factorization. The PARDISO package is a direct sparse matrix solver [6] that was developed with both serial and parallel versions [7]. Such a sparse matrix solver is efficient for large and sparse matrix computations, and it provides generality and robustness in the solution. However, its performance is sensitive to the "sparsity" of the system matrix, which might result in poor performance.

The second approach involves the use of iterative algorithms, such as the conjugate gradient or generalized minimal residual algorithms [8]. Since the active column solvers have scalability issues, iterative solvers have been proposed for a parallel implementation of large-scale finite element problems. Iterative algorithms will repeatedly perform matrix-vector or vector-vector computations until the solution converges. However, the convergence rate depends on the condition number of the system matrix, and good pre-conditioners are required to achieve efficiency. Since it is difficult to construct good pre-conditioners, iterative solvers have exhibited limited success when applied to large structural problems.

The finite element tearing and interconnecting (FETI) method is probably the most commonly used domain decomposition approach for large-scale structural analysis. Farhat and Roux [9, 10, 11] proposed FETI as a parallel solution algorithm for elliptic partial differential equations, and it is one of the first non-overlapping domain decomposition methods to demonstrate scalability in terms of both element size and the number of sub-domains. In the FETI method, the computational domain is divided into non-overlapping sub-domains, and Lagrange multipliers are used to enforce continuity across the sub-domain interfaces. The overall computational efficiency of the original FETI method was improved by using a regulation procedure based on a "balanced" perturbed Lagrangian formulation [12]. In addition, Dostal et al. [13] applied the augmented Lagrangian formulation (ALF) to the FETI method to solve the contact problems and proved that the use of the ALF would be efficient in spite of the fact that it reintroduced ill-conditioning into the auxiliary problems. After that, the approach was extended to fourth-order elasticity problems (two-level FETI), and a dual-primal FETI (FETI-DP) method [14, 15] was recently proposed.

The kernels of the floating sub-domains stiffness matrices had to be evaluated for the implementation of the original and the two-level FETI methods. In contrast, all sub-domain stiffness matrices are invertible for the FETI-DP method. The FETI-DP is a dual sub-structuring method that introduces both Lagrange multipliers and a small number of coarse mesh nodes to enforce continuity at the interfaces of the subdomain. The resulting problem is then solved by seeking a saddle-point of the relevant Lagrangian functional. The difference between the original and the FETI-DP methods is the following:

The original FETI method is a projected conjugate gradient algorithm, and hence, cannot be started with an arbitrary initial guess. In contrast, the FETI-DP is a standard preconditioned conjugate algorithm that can be started from an arbitrary initial guess. For both methods, the solution of the interface problem is obtained by using an iterative process that requires an adequate pre-conditioner. Thus, both FETI methods combine the direct solution approach for each of the sub-domains and iterative solvers for the interface problem.

Typically, a parallel implementation of direct solvers is robust, but such an implementation is complex and the scalability remains to be addressed issue. Moreover, the associated memory and computational costs increase rapidly with the problem size. In contrast, iterative algorithms efficiently utilize memory even though they exhibit poor performance when confronted with poorly conditioned matrices. Therefore, iterative solvers rely on various types of preconditioning techniques to improve the convergence of the solution.

In view of these limitations, it is appropriate to investigate a combination of domain decomposition methods with direct solvers. This approach seems to have received little attention thus far [16, 17, 18, 19]. Bauchau [20, 21] suggested the use of ALF in conjunction with both global and local Lagrange multipliers and demonstrated an improvement in the conditioning of the interface flexibility matrix. For this approach, the stiffness matrices of all sub-domains become nonsingular, bypassing the need to evaluate pseudoinverses and enabling the use of standard direct solvers. The authors proposed an improved domain decomposition approach as a preliminary step to the present effort [22], and more recently, a FETI-type algorithm was developed for use with multibody structures [23]. However, a serial version of the column solver has to be adopted to address each of the sub-domains as well as the interface problem, and doing so is not efficient. In the present paper, the computational domain will be divided into non-overlapping sub-domains by following the authors' previous study. A column solver will then be used in each of the sub-domains, and finally, a parallel version of the column solver will be used to address the interface problem. The present approach was originally developed for use in large-scale structural analysis. However, this paper consists of a preliminary validation that applies the method to two- and three-dimensional problems, such as planar and shell elements.

The paper is organized as follows. The original FETI method will be summarized in the first section, and the proposed algorithm will be compared to the original FETI method. Next, the proposed algorithm is implemented on parallel hardware, and finally, the computational cost and the results of the scalability obtained by the proposed approach will be presented for a two-dimensional planar problem and a three-dimensional cylindrical structure modelled with shell elements.

## 2. Description of the algorithms

Section 2.3 describes the original FETI method, and the role of the rigid-body modes is underlined. In Section 2.4, an augmented Lagrangian formulation will be introduced, and its impact is highlighted in terms of the characteristics of the problem. Section 2.5 provides a brief description of

the interface connection strategy for FETI and the proposed approach.

#### 2.1 Domain decomposition

Consider the planar solid shown in Fig. 1. A parallel solution algorithm can be developed for this problem by partitioning the solid into Ns non-overlapping sub-domains. Fig. 1 depicts a planar system for convenience, but all developments presented here are generally applicable to three-dimensional problems. The degrees of freedom (dofs) for each sub-domain are collected in arrays denoted as  $u^{(i)}$ ,  $i = 1, 2, N_s$ . The array stores the dofs of the sub-domain i, i.e., the displacement components for all nodes of the sub-domain i. (•)<sup>(i)</sup> indicates the quantities that are associated with sub-domain i, and the global array of dofs is defined as

$$\underline{\underline{u}}^{T} = \left\{ \underline{\underline{u}}^{(1)T}, \ \underline{\underline{u}}^{(2)T}, \ \dots \ \underline{\underline{u}}^{(N_{s})T} \right\}$$
(1)

Array  $\underline{u}$  has a size of  $n_u = \sum_{i=1}^{N_s} n_u^{(i)}$ , which is the total number of dofs of a complete structure. Since the original domain is divided into sub-domains, the nodes along the interfaces and the associated dofs are duplicated. Consequently, array  $\underline{u}$  contains a large number of redundant dofs with all interface dofs appearing two or more times. The variables stored in array  $\underline{u}$  should be referred to as "generalized coordinates" because they do not form a minimum set, but the term "dofs," which is more widely used in the finite element literature, will be used here.

The dofs of each sub-domain can be split into two



Fig. 1. Planar solid separated into four non-overlapping sub-domains



Fig. 2. Classical and localized Lagrange multipliers

mutually exclusive groups: the internal and the boundary dofs. The boundary dofs are those that are exposed when dividing the original problem into sub-domains while the remaining dofs are internal. Kinematic constraints will be imposed at the boundary nodes to enforce continuity in the displacement field, thereby ensuring that the behavior of the connected sub-domains is identical to that of the original, un-partitioned solid.

#### 2.2 Lagrange multipliers

Linear constraints are imposed to enforce the continuity of the displacement field across the sub-domain boundaries, and an equality in the dofs corresponds to nodes in adjacent sub-domains. This was achieved in the original FETI approach by using a classical Lagrange multiplier technique, as conceptually illustrated in Fig. 2. Let the displacement vectors of two nodes belonging to two adjacent sub-domains be denoted as  $\underline{u}_1$  and  $\underline{u}_2$ . The continuity of the displacement field across the interface of the two sub-domains implies  $\underline{C} = \underline{u}_1 - \underline{u}_2$ , where  $\underline{C}$  is the constraint that is imposed.

An alternative approach involves defining an independent interface node, which is denoted as  $\underline{c}$ . Two kinematic constraints are then imposed with displacement components at the boundary nodes in the two sub-domains adjacent to the interface that must equal those at the independent interface nodes. For the simple connection illustrated in Fig. 2, the two kinematic constraints become  $\underline{C}^{[1]} = \underline{u}_1 - \underline{c} = \underline{0}$  and  $\underline{C}^{[2]} = \underline{u}_2 - \underline{c} = \underline{0}$ , and the corresponding constraint potential is  $V_c = \underline{\lambda}^{[1]T} \underline{C}^{[2]}$ . (•)<sup>[j]</sup> indicates the quantities that are associated with constraint *j*.

For this approach, Lagrange multipliers  $\underline{\lambda}^{[1]}$  enforce the constraint between the boundary dofs of sub-domain 1, denoted as  $u_1$ , and the interface dofs, c. Similarly, the Lagrange multipliers  $\lambda^{[2]}$  enforce the constraint between the boundary dofs of the adjacent sub-domain 2, denoted as  $\underline{u}_2$ , and the same interface dofs, c. No direct constraint is written between the dofs of the two sub-domains. Consequently, Lagrange multipliers  $\lambda^{[1]}$  and  $\lambda^{[2]}$  become "localized," i.e.,  $\underline{\lambda}^{[1]}$  and  $\underline{\lambda}^{[2]}$  are local variables of subdomains 1 and 2, respectively. The name "localized Lagrange multiplier technique" stems from this feature of this approach. Note that the constraints are localized as well with constraints  $C^{[1]}$  and  $C^{[2]}$  associated with sub-domains 1 and 2, respectively. A single interface node is introduced when multiple connections occur for single nodes. In finite element formulations, this approach has been used to enforce continuity in the displacement fields between adjacent incompatible elements [24]. The same approach is referred to as a "localized version of the method of Lagrange multipliers," as proposed by Park et al. [25, 26].

#### 2.3 The original FETI method

In the original FETI method, the computational domain is divided into non-overlapping sub-domains, as shown in Fig. 1. The Lagrange multipliers can be interpreted as connection forces and are introduced to enforce continuity in the displacement field at the interface nodes. The general form of the linear static structural problems in the computational domain  $\Omega$  is cast as

$$\underline{\underline{K}}\underline{\underline{u}} = \underline{\underline{f}} \tag{2}$$

where  $\underline{K}$ ,  $\underline{u}$  and  $\underline{f}$  are the global stiffness matrix, displacement vector, and prescribed force vector, respectively. The computational domain is then divided into  $N_s$  non-overlapping regions or sub-domains, and Lagrange multipliers are introduced to enforce continuity in the displacement problem, which becomes a saddle-point problem.

Once these are divided into sub-domains, the governing equation from Eq. (2) takes the following form

$$\underline{\underline{K}}^{(s)}\underline{\underline{\mu}}^{(s)} = \underline{\underline{f}}^{(s)} + \sum_{s=1}^{a^{(s)}} \underline{\underline{B}}^{(s)T}_{j} \underline{\underline{\lambda}}, \quad s = 1, 2, \dots N_s$$
(3a)

$$\underline{\underline{B}}_{j}^{(s)}\underline{\underline{u}}^{(s)} = \underline{\underline{\underline{B}}}_{s}^{(j)}\underline{\underline{u}}^{(j)}, \quad s = 1, 2, \cdots N_{s}$$
(3b)

where  $\underline{K}_{\underline{i}}^{(s)}$ ,  $\underline{u}^{(s)}$ , and  $\underline{f}^{(s)}$  are the stiffness matrix, displacement vector, and prescribed force vector, respectively, for subdomain s. The  $\underline{B}_{\underline{j}}^{(s)}$  matrix is the Boolean connectivity matrix,  $\underline{\lambda}$  is the array of Lagrange multipliers and  $a^{(s)}$  is the number of sub-domains that are connected to sub-domain *s*. If all subdomains are constrained by adequate boundary conditions, stiffness matrices  $\underline{K}_{\underline{i}}^{(s)}$  are non-singular and the solution of Eq. (3) becomes

$$\underline{u}^{(s)} = \underline{\underline{K}}^{(s)-1} \left( \underline{\underline{f}}^{(s)} + \underline{\underline{\underline{B}}}^{(s)T} \underline{\underline{\lambda}} \right), \quad s = 1, 2, \cdots N_s$$
(4a)

$$\left[\sum_{s=1}^{N_s} \underline{\underline{B}}^{(s)} \underline{\underline{K}}^{(s)-1} \underline{\underline{B}}^{(s)T}\right] \underline{\lambda} = \left[\sum_{s=1}^{N_s} \underline{\underline{\underline{B}}}^{(s)} \underline{\underline{K}}^{(s)-1} \underline{\underline{f}}^{(s)}\right]$$
(4b)

The problem is solved in two steps. First, the solution of the interface problem (4b) yields Lagrange multipliers, and second, the displacement field in each sub-domain is evaluated using Eq. (4a).

As shown in Fig. 1, typical configurations include floating sub-domains with stiffness matrices that are singular and the kernel of which are sub-domain rigid-body modes. The solution procedure is modified as follows. First, a direct factorization scheme is used in each sub-domain to Jun Young Kwak Domain Decomposition Approach Applied for Two- and Three-dimensional Problems via Direct Solution ...

compute the Moore-Penrose inverse of the stiffness matrix, denoted as  $K_{\equiv}^{(s)\ell}$ , knowing the sub-domain rigid-body modes. Second, the contributions of the rigid-modes are related to the Lagrange multipliers through an orthogonality condition. The solution procedure described in Eq. (4) is then modified, leading to

$$\underline{u}^{(s)} = \underline{\underline{K}}^{(s)\ell} \left( \underline{\underline{f}}^{(s)} + \underline{\underline{B}}^{(s)T} \underline{\underline{\lambda}} \right) + \underline{\underline{R}}^{(s)} \underline{\underline{\alpha}}^{(s)}, \quad s = 1, 2, \dots N_s$$
 (5a)

$$\underline{\underline{R}}^{(s)T}\left(\underline{f}^{(s)} - \underline{\underline{\underline{B}}}^{(s)T}\underline{\underline{\lambda}}\right) = 0, \quad s = 1, 2, \cdots N_s$$
(5b)

where the matrix stores the sub-domain rigid-body modes and a set of unknown coefficients.

The interface problem is obtained by combining Eqs. (3b), (5a), and (5b) to find

$$\begin{bmatrix} \underline{F}_{I} & -\underline{G}_{I} \\ -\underline{G}_{I}^{T} & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{\lambda} \\ \underline{\alpha} \end{bmatrix} = \begin{bmatrix} \underline{d} \\ -\underline{e} \end{bmatrix}$$
(6)

where  $\underline{F}_{I} = \sum_{s=1}^{N_s} \underline{B}^{(s)} \underline{K}^{(s)\ell} \underline{B}^{(s)T}$ ,  $\underline{G}_{I} = \left[\underline{B}^{(1)} \underline{R}^{(1)}, \dots, \underline{B}^{(N_s)} \underline{R}^{(N_s)}\right]$ ,  $\underline{\alpha}^{T} = \left[\underline{\alpha}_{1}^{T}, \dots, \underline{\alpha}_{N_s}^{T}\right]$ ,  $\underline{d} = \sum_{s=1}^{N_s} \underline{B}^{(s)} \underline{K}^{(s)\ell} \underline{f}^{(s)}$  and  $\underline{e} = \underline{R}^{(s)T} \underline{f}^{(s)}$ . The pseudo-inverse stiffness matrix is defined as  $\underline{K}^{(s)\ell} = \underline{K}^{(s)-1}$  for non-floating sub-domains and for floating sub-domains, and  $\underline{K}^{(s)\ell}$  indicates the Moore-Penrose inverse. Eq. (6) can be solved by adopting iterative methods such as the preconditioned conjugate projected gradient (PCPG) method. Either lumped or Dirichlet preconditioner is used for the PCPG algorithm, and the lumped pre-conditioner is more efficient than its Dirichlet counterpart but is not mathematically optimal. On the other hand, Dirichlet preconditioner exhibits optimal convergence characteristics. Farhat et al. introduced one of the basic pre-conditioners used in FETI [14, 15]. Finally, the sub-domain displacement fields,  $\underline{u}^{(s)}$ , can be obtained from Eq. (5).

#### 2.4 Proposed domain decomposition approach

The continuity of the displacement field across the subdomain interfaces is enforced in the proposed approach by using the augmented Lagrangian formulation, which provides natural conditioning for the problem [21]. Direct solvers are used for both the sub-domain and interface stiffness matrix factorizations, thereby eliminating the need for pre-conditioners.

The total potential energy of the system, denoted as  $\prod$ , is expressed as  $\prod = A + \Phi + V_c$ , where A is the strain energy,  $\Phi$  is the potential of the externally applied loads, and  $V_c$  is the potential of the constraints. The total strain energy is evaluated by summing up the contributions from each subdomain.

$$A = \sum_{i=1}^{N_s} A^{(i)} = \frac{1}{2} \sum_{i=1}^{N_s} \underline{u}^{(i)T} \underline{\underline{K}}^{(i)} \underline{\underline{u}}^{(i)} = \frac{1}{2} \underline{\underline{u}}^T \operatorname{diag}\left(\underline{\underline{K}}^{(\alpha)}\right) \underline{\underline{u}}$$
(7)

where  $\underline{K}^{(i)}$  denotes the stiffness matrix of sub-domain *i* and, diag $\left(\underline{K}^{(\alpha)}\right)$  represents the global stiffness matrix of the system. If the structure is constrained by suitable boundary conditions, the global stiffness matrix will not be singular even though the stiffness matrices of the floating sub-domains are singular.

The total work performed by the externally applied loads is  $\Phi = \sum_{i=1}^{N_s} \Phi^{(i)} = -\sum_{i=1}^{N_s} \underline{u}^{(i)T} \underline{Q}^{(i)} = -\underline{u}^T \underline{Q}$ , where  $\underline{Q}^{(i)}$  is the load vector for a sub-domain and the global load array of the system is defined as  $\underline{Q}^T = \{\underline{Q}^{(1)T}, \underline{Q}^{(2)T}, \dots, \underline{Q}^{(N_s)T}\}$ .

The kinematic continuity conditions across the subdomain interfaces are enforced using the localized Lagrange multiplier technique. Let  $\underline{u}_{b}^{[J]}$  and  $\underline{c}^{[J]}$  denote the arrays of the dofs at a boundary node and at an interface node, respectively. The kinematic constraint *j* is written as  $C^{[J]} = u_{b}^{[J]} - c^{[J]} = 0$ , and the associated potential is

$$V_{c}^{[j]} = s\underline{\lambda}^{[j]T}\underline{C}^{[j]} + \frac{p}{2}\underline{C}^{[j]T}\underline{C}^{[j]}$$
(8)

where  $\lambda^{[j]}$  is the array of the Lagrange multipliers that are used to enforce the constraint, and s the scaling factor for those multipliers. The first term on the right-hand side of Eq. (8) provides the enforcement of the constraints by using the Lagrange multiplier technique. Note the presence of the scaling factor, s. The second term on the right-hand side of Eq. (8) can be interpreted as the constraint enforcement via the penalty method, and the combination of the two approaches is known as the Augmented Lagrangian Formulation [27, 28], which is a technique that finds a general solution for boundary value problems and is particularly useful when dealing with constrained dynamic systems [29, 30, 31]. Bottasso et al. [32] proposed using augmented Lagrangian formulations to solve differential-algebraic equations. These aspects of the formulation become indispensable when the present work is extended to address structural and multibody dynamics problems.

The variation in the constraint potential, Eq. (8), leads to

$$\delta V_c^{[j]} = \left(\delta \underline{u}_b^{[j]} - \delta \underline{c}^{[j]}\right)^T \left[s \underline{\lambda}^{[j]} + p \underline{C}^{[j]}\right] + \delta \underline{\lambda}^{[j]T} \left[s \underline{C}^{[j]}\right]$$
(9)

and gives rise to the following generalized forces of the constraint,

$$\underline{f}^{[j]} = \begin{cases} s\underline{\lambda}^{[j]} + p\underline{C}^{[j]} \\ s\underline{C}^{[j]} \\ -\left(s\underline{\lambda}^{[j]} + p\underline{C}^{[j]}\right) \end{cases}$$
(10)

http://ijass.org

#### Int'l J. of Aeronautical & Space Sci. 16(2), 177-189 (2015)

Taking the derivative of these forces for the constraint with respect to the dofs yields the stiffness matrix of the constraint,

$$\underline{k}_{\underline{z}}^{[j]} = \begin{bmatrix} p\underline{I} & s\underline{I} & -p\underline{I} \\ s\underline{I} & \underline{0} & -s\underline{I} \\ -p\underline{I} & -s\underline{I} & p\underline{I} \end{bmatrix}$$
(11)

where *I* denotes the identity matrix.

Since the Lagrange multipliers are localized in the proposed approach, the constraint forces and the stiffness matrix are partitioned as follows

$$\underline{f}^{[j]} = \left\{ \underbrace{\underline{f}_{b}^{[j]}}_{\underline{f}_{c}^{[j]}} \right\}, \quad \underline{\underline{k}}^{[j]} = \left[ \underbrace{\underline{\underline{k}}_{bb}}_{\underline{\underline{k}}_{bc}^{[j]}}, \quad \underline{\underline{k}}_{bc}^{[j]} \right]$$
(12)

where the partitions are indicated by the horizontal and vertical lines in Eqs. (10) and (11). Subscripts  $(\bullet)_b$  and  $(\bullet)_c$  denote the dofs associated with the boundary and interface nodes, respectively. In summary, each of the kinematic constraints generates an array of constraint forces and a constraint stiffness matrix. Each kinematic constraint can clearly be viewed as a finite element, and in the sequel, the terms "kinematic constraint" and "constraint element" will be used interchangeably, and Kwak et al. provided a physical interpretation of the ALF formulation [23].

To connect the  $N_s$  sub-domains, a total of  $N_c$  interface nodes are defined, and the following array stores the dofs at all these interface nodes,

$$\underline{c}^{T} = \left\{ \underline{c}_{1}^{T}, \underline{c}_{2}^{T}, \cdots, \underline{c}_{N_{c}}^{T} \right\}$$
(13)

The array  $\underline{c}$  has a size  $n_c$ . The total potential for all constraints associated with the sub-domain *i*, denoted  $V_c^{(i)}$ , is found by summing up the potentials of the corresponding constraint,  $V_c^{(i)} = \sum_{j=1}^{N_b^{(i)}} V_c^{(j)}$ . Finally, the total potential of all kinematic constraints is  $V_c = \sum_{i=1}^{N_s} V_c^{(i)}$ . Each constraint element contributes the constraint forces and stiffness matrices defined in Eq. (12). A standard assembly procedure is used to assemble the finite element method [1] by generating force arrays and stiffness matrices according to all constraint elements associated with sub-domain i into the following sub-domain arrays and matrices

$$\underline{\breve{F}}_{b}^{(i)} = \sum_{j=1}^{N_{b}^{(i)}} \underline{\underline{B}}_{b}^{[j]T} \underline{f}_{b}^{[j]}, \quad \underline{\breve{K}}_{bb}^{(i)} = \sum_{j=1}^{N_{b}^{(i)}} \underline{\underline{B}}_{b}^{[j]T} \underline{\underline{k}}_{bb}^{[j]T} \underline{\underline{B}}_{b}^{[j]}$$
(14)

where  $\underline{\underline{B}}_{b}^{[j]}$  represents the Boolean matrices used for the assembly process, i.e.,  $\underline{\underline{u}}_{b}^{[j]} = \underline{\underline{B}}_{b}^{[j]} \underline{\underline{u}}_{b}^{(i)}$ . Of course, the assembly procedure can be performed in parallel for all sub-domains. Similarly, the constraint elements contribute force arrays and stiffness matrices to the interface problem,

$$\underline{F}_{c}^{(i)} = \sum_{j=1}^{N_{b}^{(i)}} \underline{\underline{B}}_{c}^{[j]T} \underline{f}_{c}^{[j]}, \quad \underline{\underline{K}}_{cc}^{(i)} = \sum_{j=1}^{N_{b}^{(i)}} \underline{\underline{B}}_{c}^{[j]T} \underline{\underline{k}}_{cc}^{[j]T} \underline{\underline{B}}_{c}^{[j]}$$
(15)

where  $\underline{\underline{B}}_{c}^{[j]}$  represents the Boolean matrices used for the assembly process, i.e.,  $\underline{c}_{b}^{[j]} = \underline{\underline{B}}_{c}^{[j]} \underline{c}$ . Finally, the constraint coupling stiffness is assembled to find

$$\underline{\underline{K}}_{bc}^{(i)} = \sum_{j=1}^{N_{b}^{(i)}} \underline{\underline{B}}_{b}^{[j]T} \underline{\underline{k}}_{bc}^{[j]} \underline{\underline{B}}_{c}^{[j]}$$
(16)

The total potential energy of the system,  $\prod = A + \Phi + V_c$ , is evaluated, and the principle of the minimum total potential energy yields the governing equations as

$$\begin{bmatrix} \operatorname{diag}\left(\underline{\breve{K}}^{(\alpha)}_{=} + \underline{\breve{K}}^{(\alpha)}_{bb}\right) & \underline{\breve{K}}_{bc} \\ \underline{\breve{K}}^{T}_{=bc} & \underline{\breve{K}}_{cc} \end{bmatrix} \left\{ \underline{\breve{U}} \right\} = \left\{ \underline{\breve{Q}} - \underline{\breve{F}}_{b} \\ -\underline{F}_{c} \end{bmatrix}$$
(17)

where arrays  $\underline{\underline{F}}_{b}$  and  $\underline{\underline{F}}_{b}$  are the assembly of their sub-domain counterparts,  $\underline{\underline{F}}_{b}^{(i)}$  and  $\underline{\underline{F}}_{c}^{(i)}$ , respectively,  $\underline{\underline{K}}_{cc} = \sum_{i=1}^{N_{s}} \underline{\underline{K}}_{cc}^{(i)}$  and  $\underline{\underline{K}}_{bc}^{T} = \left[\underline{\underline{K}}_{bc}^{(1)T}, \underline{\underline{K}}_{bc}^{(2)T}, \cdots, \underline{\underline{K}}_{bc}^{(N_{s})T}\right]$  (18)

The block-diagonal nature of the leading entry of the system matrix makes this approach amenable to parallel solution algorithms.

Note that in Eq. (17) the sub-domain stiffness matrices  $\underline{\breve{K}}^{(\alpha)} + \underline{\breve{K}}^{(\alpha)}_{bb}$  are never singular, even for floating sub-domains, even though stiffness matrices  $\underline{\breve{K}}^{(\alpha)}$  are indeed singular. Stiffness matrices  $\underline{\breve{K}}^{(\alpha)}_{bb}$  stem from the use of an augmented Lagrangian formulation and can be interpreted as connecting each of the sub-domains to ground by means of springs with a stiffness constant p, thereby removing all singularities. Hence, the first level Schur complement is well conditioned, and a direct factorization algorithm, such as LU factorization, can be used. On the other hand, the sparsity in Eq. (17) resulting from the effects of Eq. (8) can be manifested by applying the sparse matrix solver, such as PARDISO and MUMPS.

The sub-domain interface node stiffness matrices are defined as

$$\underline{\underline{K}}_{I}^{(i)} = p^{(i)} \underline{\underline{I}} - \underline{\underline{\hat{S}}}^{(i)} \underline{\underline{\hat{K}}}^{(i)-1} \underline{\underline{\hat{S}}}^{(i)T}$$
(19)

where  $\underline{\hat{S}}^{(i)} = \left[ p^{(i)} \underline{\underline{B}}^{(i)}_{\underline{b}}, p^{(i)} \right]$  and

$$\underline{\hat{K}}^{(i)} = \begin{bmatrix} \operatorname{diag}\left(\underline{\breve{K}}^{(i)} + p^{(i)}\underline{B}^{(i)T}_{b}\underline{B}^{(i)}\right) & \operatorname{diag}\left(\underline{\hat{S}}^{(i)T}\right) \\ \operatorname{diag}\left(\underline{\hat{S}}^{(i)}\right) & 0 \end{bmatrix}$$
(20)

The sub-domain interface node load arrays are

DOI: http://dx.doi.org/10.5139/IJASS.2015.16.2.177

Jun Young Kwak Domain Decomposition Approach Applied for Two- and Three-dimensional Problems via Direct Solution ...

(21)

$$\underline{f}_{\underline{}}^{(i)} = \underline{\hat{\underline{S}}}_{\underline{=}}^{(i)} \underline{\hat{\underline{K}}}_{\underline{=}}^{(i)-1} \underline{\hat{Q}}_{\underline{=}}^{(i)}$$

where  $\hat{Q}^T = \begin{bmatrix} Q^T, 0 \end{bmatrix}$ 

#### 2.5 Interface structure in each approach

The various approaches described thus far differ primarily in the manner in which they enforce continuity among the various sub-domains, as illustrated in Fig. 3. Lagrange multipliers (dual variables) are used to impose these constraints in the original FETI method. In contrast, a small number of "coarse mesh nodes" are defined in the FETI-DP approach, typically at the corner nodes. At those coarse mesh nodes, the unknowns are displacement (primal) variables. Hence the name FETI-DP, where "DP" stands for Dual-Primal, to indicate that both dual and primal variables are involved in the interface problem. Note that if an adequate number of coarse mesh nodes is used in each sub-domain, the corresponding stiffness matrices are no longer singular, simplifying the solution process. For the proposed approach, the localized Lagrange multipliers and the interface nodes are used to impose continuity across the sub-domain boundaries. A single interface node is defined at the corner nodes, i.e., at the cross-points of the sub-domain.





Fig. 4. Computational step of the proposed approach

## 3. Solution procedure for the proposed approach

For domain decomposition approaches, the overall computational efficiency is often determined by the strategy that is used to solve the interface problem. Indeed, the computations associated with each sub-domain are independent of each other and will scale as the number of processors increases. On the other hand, the solution of the interface problem involves interaction among the processors that can severely slow down the solution process.

## 3.1 Parallel computing algorithm used in the proposed approach

The proposed approach proceeds in the three computational steps, as shown in Fig. 4. Message passing is implemented in the proposed approach, as shown in Fig. 5, and Step I sets up a structural interface problem, Step II evaluates the solution of the structural interface problem, and Step III recovers the solution from each sub-domain. LAPACK and ScaLAPACK are standard and portable libraries that are used in the present implementation. Table 1 shows the specific solvers of the libraries and their numerical features within each step.

The purpose of Step I is to set up an interface problem that involves (1) the evaluation and assembly of the stiffness matrix, (2) the factorization of the stiffness matrix, and (3) the assembly of the interface stiffness matrix for each subdomain.

Step II computes the solution of the interface problem. In this step, the stiffness matrix corresponding to the interface nodes existing in the individual sub-domains needs to be



http://ijass.org

distributed to each processor. The MPI\_REDUCE routine is then used to collect the matrix data to a root process.

The computational load is then distributed to  $N_s$  processes by a parallel solver for linear equations, such as the ScaLAPACK library. ScaLAPACK is a message passing version of LAPACK. It assumes that the tasks are to be distributed according to the two-dimensional block-cyclic data layout scheme. The block cyclic distribution provides a general way to distribute a block-partitioned matrix on concurrent distributed memory computers. In this process, LU decomposition is used with partial pivoting and row interchanges.

In Step III, the final solution is obtained for each subdomain by the linear solver. From Step II, array  $\underline{c}$ , the degrees of freedom at the interface nodes, is obtained. Thus, the displacement of each sub-domain  $\underline{u}^{(i)}$  is easily obtained. The MPI\_BCAST routine sends the value of array  $\underline{c}$  to all other processes first, and then, the LU decomposition routine of the LAPACK library is used to recover the solution for each sub-domain.

## 4. Numerical Results

The proposed approach is implemented in order to solve static, two-dimensional plane stress and three-dimensional shell problems. The present parallel computations were executed in the TACHYON system, a supercomputer operated by Korea Institute of Science and Technology Information. The detailed specifications of the TACHYON system are listed in Table 2.

Section 4.1 will discuss the results for the two-dimensional configuration, and the computational cost and the scalability are examined in the parallel environment, and the proposed approach is compared with FETI-DP to evaluate the computational cost. Section 4.2 will then examine the corresponding results for the three-dimensional structure.

#### 4.1 Two-dimensional plane stress problem

In this section, a two-dimensional structure is examined in the plane stress conditions. Fig. 6 shows the structure



Fig. 5. Parallel implementation of the proposed approach

Table 1. LAPACK [33] and ScaLAPACK [34] solvers involved in the proposed approach

	Step I	Step II	Step III
Library Solver	LAPACK	ScaLAPACK	LAPACK
	DGETRF	PDGESV	DGESV
		LU decomposition	
Feature	LU factorization	Parallel computation	LU decomposition
		Block-cyclic scheme	_

Table 2. Specifications of TACHYON system

	Value
Processor	AMD Operation, 2.0GHz(Barcelona)
Number of nodes	188
Number of CPU core	3008
Memory	DDR2/667MHz 6TB
OS	CentOS 4.6

DOI: http://dx.doi.org/10.5139/IJASS.2015.16.2.177

subjected to a uniform load applied along the right edge. The equations of the planar elasticity are used [1]. Constitutive laws are assumed for a linearly elastic homogeneous and isotropic material, i.e.,  $\underline{\sigma} = \underline{\underline{D}}\underline{\varepsilon}$  as the material stiffness matrix. The element stiffness matrix, denoted as  $\underline{\underline{k}}^e$ , is given as  $\underline{\underline{k}}^e = \int_{\Omega^e} \underline{\underline{B}}^T \underline{\underline{D}}\underline{\underline{B}} d\Omega$ , where  $\underline{\underline{B}}$  is the strain interpolation matrix, and  $\Omega^e$  is the element domain. For the numerical examples shown below, the modulus of elasticity is 73 GPa and the Poisson's ratio is 0.3.

The proposed approach adopts the localized Lagrange multiplier described in Section 2. A relatively small number of dofs was used for the initial runs. The number of sub-domains increased from 4 to 225, but the number of dofs was kept to a total of 7,442. First, the serial LAPACK library was used to solve this problem through the use of an undecomposed domain only, and the computational time was 99.42 s. Then, the PARDISO package, a direct sparse matrix solver, was employed to solve the same undecomposed domain problem. The serial version of PARDISO, which is included in the Intel Math Kernel Library [35], was used, and the computational time was 2.25 s. Then, the parallel ScaLAPACK library was implemented for the same undecomposed domain problem. Finally, the proposed approach was executed. Fig. 7 shows the performance of the proposed approach, which is also summarized in Table 3.



Fig. 6. Configuration of the two-dimensional problem



Fig. 7. Computational time of the proposed approaches, ScaLAPACK and PARDISO

Table 5. Companson of the computational time	Table 3.	Comparison	of the cor	mputational	time
--	----------	------------	------------	-------------	------

Sub-	Proposed	CDU	Parallel	CDU	Serial	Serial
domains	approach [s]	CPUS	ScalAPACK [s]	CPUS	[s]	LAPACK [s]
4	6.52	4	27.63			
9	1.84	9	15.37			
16	0.64	16	10.07	1	2.25	00.42
25	0.45	25	8.17	1	1 2.25	99.42
36	0.53	36	7.09			
100	1.03	100	5.49			

## http://ijass.org

#### Int'I J. of Aeronautical & Space Sci. 16(2), 177-189 (2015)



Fig. 8. Computational performance of three approaches for the two-dimensional problem



Fig. 9. Memory usage of the proposed approaches in the two-dimensional problem

The results indicate that the proposed parallel implementation achieved a good performance. In addition, the computational time consumed by the proposed algorithm decreased to 0.45 s when using an optimal number of 25 processors. The proposed algorithm based on the domain decomposition outperforms not only the parallel ScaLAPACK library but also the PARDISO sparse matrix solver when it is applied to an undecomposed domain.

The problem size was then increased to 35,378 dofs, and the computational efficiency and the memory usage of the proposed approach were examined. The number of subdomains increased from 4 to 36, and Figs. 8 and 9 depict the respective computational time and memory usage required by FETI-DP and the proposed approach. The proposed approach outperforms FETI-DP in terms of both the computational time and the memory usage.

Figure 10 presents the scalability of the proposed approach, as estimated by the speed-up defined as the ratio between the computational time for the serial processing and for parallel processing with  $N_s$  processors. An ideal speed-up was achieved when the speed-up approaches the number of processors, and the results indicate that good scalability characteristics for the proposed approach.

Finally, Fig 11 illustrates the computational time consumed by Step II in the present computation. In Step II, a parallel version of the linear solver was used to compute the interface problem. As the number of sub-domains increased, the number of dofs increases as well for the interface problem, and the use of the parallel solver becomes increasingly beneficial for the interface problem.



Fig. 10. Speed-up for the two-dimensional problem

# 4.2 Three-dimensional structural problem using a shell element

The shell problem depicted in Fig. 12 is investigated in this section. The material and geometrical properties of the cylinder and the loading condition are also listed in the figure. A small deflection is assumed for this problem. The degenerate shell element proposed by Ahmad et al. [36, 37, 38] was used in the present effort. A total of 86,544 dofs were used, and the number of the sub-domains increased from 10 to 40.

Table 4 provides a summary of the computational time and the memory usage required by the proposed approach. As



Fig. 11. Computational time required by the various steps of the algorithm



Fig. 12. Configuration of the three-dimensional problem

Table 4. Computational time and memory usage in the shell problem

the number of processors increases, the computational time decreases from 466.63 to 33.88 s, and the maximum memory usage increases from 1785 to 179.78 Mb per processor. The scalability was also assessed by evaluating the speed-up, and Fig. 13 shows the good scalability potential of the proposed approach.

## 5. Conclusions

This paper describes the development of a finite element based domain decomposition algorithm that relies only on direct solvers. The proposed approach uses the domain decomposition concept that characterizes classical FETI methods and enforces continuity in the displacement field across the sub-domain interfaces by using a combination of localized Lagrange multiplier methods and the augmented Lagrangian formulation. This approach generates wellconditioned stiffness matrices and allows the use of direct solvers for both sub-domain and interface problems. The preliminary stage of the computation was conducted by using the proposed approach. The computational cost and the scalability of the proposed method were compared to those of FETI-DP for two-and three-dimensional problems. The results indicate that the proposed approach outperformed FETI-DP in those problems. A further comparison with the PARDISO sparse matrix solver revealed that the proposed approach achieved an outstanding performance. The proposed approach was demonstrated to have good scalability for realistic, three-dimensional shell problems. Furthermore, the proposed approach can be improved by employing a sparse matrix solver to handle the sparsity within the governing equation.

The proposed approach will thus be extended for a dynamic solution of nonlinear multibody systems involving nonlinear kinematic constraints and time transient analysis.

## Acknowledgement

This work was supported by the Human Resources

Number of	Computational time [s]	Memory usage per processor [Mb]
sub-domains	Proposed approach	Proposed approach
10	466.63	1785.30
20	114.34	271.81
25	74.93	172.12
30	54.07	177.13
40	33.88	177.13

#### http://ijass.org

Development program (No. 20124030200030) of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government Ministry of Trade, Industry and Energy. This work was also supported by the Advanced Research Center Program (No. 2013073861) through a grant from the National Research Foundation of Korea (NRF), funded by the Korean government (MSIP) as a contract with the Next Generation Space Propulsion Research Center at Seoul National University.

## References

[1] Bathe, K. J., Finite Element Procedures, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1996..

[2] Irons, B. M., "A frontal solution program for finite element analysis." *International Journal for Numerical Methods in Engineering*, Vol. 2, 1970, pp. 5-32.

[3] Duff, I. S. and Reid, J. K., "The multifrontal solution of indefinite sparse symmetric linear equations." *ACM Transactions on Mathematical Software*, Vol. 9, No. 3, 1973, pp. 302-325.

[4] Kim, J. H. and Kim, S. J., "Multifrontal solver combined with graph partitioners." *AIAA Journal*, Vol. 37, No. 8, 1999, pp. 964–970.

[5] Gould, N. I. H. and Scott, J. A., "A Numerical Evaluation of Sparse Direct Solvers for the Solution of Large Sparse Symmetric Linear Systems of Equations." *ACM Transactions on Mathematical*, Vol. 33, No. 2, 2007, pp. 10.

[6] O. Schenk, and K. Gartner. *PARDISO User Guide Verson* 5.0.0., 2014.

[7] Schenk , O., Gartner, K. and Fichtner, W., "Efficient Sparse LU Factorization with Left-Rghit Looking Strategy on Shared Memory Multiprocessors." *BIT Numerical Mathematics*, Vol. 40, No. 1, 1999, pp. 158–176.

[8] Saad, Y. and Schultz, M. H., "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems" *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869.

[9] Farhat, C. and Roux, F.-X., "A method of finite element tearing and interconnecting and its parallel solution algorithm." *International Journal for Numerical Methods in Engineering*, Vol. 32, 1991, pp. 1205–1227.

[10] Farhat, C., Chen, P. S. Mandel, J. and Roux, F.-X., "The two-level FETI method for static and dynamic plate problems. Part I an optimal iterative solver for biharmonic systems." *Computer Methods and Applied Mechanics and Engineering*, Vol. 155, 1998, pp. 129–151.

[11] Farhat, C., Chen, P. S. Mandel, J. and Roux, F.-X., "The two-level FETI method. Part II extension to shell problems,

parallel implementation and performance results." *Computer Methods and Applied Mechanics and Engineering,* Vol. 155, 1998, pp. 153–179.

[12] Farhat, C., Chen, P. S. and Roux, F.-X., "The dual schur complement method with well-posed local neumann problems: Regularization with a perturbed lagrangian formulation." *SIAM Journal on Scientific Computing*, Vol. 14, No. 3, 1993, pp. 752–759.

[13] Dostal, R. Z., Friedlander, A. and Santos, S. A., "Solution of coercive and semicoercive contact problems by FETI domain decomposition." *Contemporary Mathematics*, Vol. 218, 1998.

[14] Farhat, C., Lesoinne, M., and Pierson, K., "A scalable dual-primal domain decomposition method." *Numerical Linear Algebra with Applications*, Vol. 7, 2000, pp. 687–714.

[15] Farhat, C., M. Lesoinne, P. Tallec, L., Pierson, K. and Rixen, D., "FETI-DP: a dual-primal unified FETI method - Part I: A faster alternative to the two-level FETI method." *International Journal for Numerical Methods in Engineering*, Vol. 50, 2001, pp. 1523–1544.

[16] Hackbusch, R. W., Khoromskij, B. N. and Kriemann, R., "Direct Schur complement method by domain decomposition based on H-matrix approximation." *Computing and Visualization in Science*, Vol. 8, 2005, pp. 179–188.

[17]Li, R.K., "Fastandhighlyscalableparallelcomputations for fundamental matrix problems on distributed memory systems." *The Journal of Supercomputing*, Vol. 54, 2010, pp. 271–297.

[18] Gu`eye, I., Arem S. E., Feyel F., Roux R.X. and Gailletaud G., "A new parallel sparse direct solver: Presentation and numerical experiments in large-scale structural mechanics parallel computing." *International Journal for Numerical Methods in Engineering*, Vol. 88, 2011, pp. 370–384.

[19] Tak M. and Park T., "High scalable non-overlapping domain decomposition method using a direct method for finite element analysis." *Computer Methods in Applied Mechanics and Engineering*, Vol. 264, 2013, pp. 108–128.

[20] O.A. Bauchau, A. Epple, and C.L. Bottasso. Scaling of constraints and augmented Lagrangian formulations in multibody dynamics simulations. *Journal of Computational and Nonlinear Dynamics*, Vol. 4, No. 2, 2009, pp. 1–9.

[21] O.A. Bauchau. "Parallel computation approaches for flexible multibody dynamics simulations", *Journal of the Franklin Institute*, Vol. 347, No. 1, February 2010, pp. 53–68.

[22] Kwak J. Y., Cho H. S., Shin S. J. and Bauchau O. A., "Development of finite element domain decomposition method using local and mixed Lagrange multipliers." *Journal* of the Computational Structural Engineering Institute of Korea, Vol. 25, No. 6, 2012, pp. 469–476. Jun Young Kwak Domain Decomposition Approach Applied for Two- and Three-dimensional Problems via Direct Solution ...

[23] Kwak , J. Y., Chun, T. Y., Shin, S. J. and Bauchau, O.A., "Domain decomposition approach to flexible multibody dynamics simulation." *Computational Mechanics*, Vol. 53, 2014, pp. 147-158.

[24] Tong P. and Pian T. H. H., "A hybid-element approach to crack problems in plane elasticity." *International Journal for Numerical Methods in Engineering*, Vol. 7, 1973, pp. 297– 308.

[25] Park K. C. and Felippa C. A., "A variational framework for solution method developments in structural mechanics." *Journal of Applied Mechanics*, Vol. 65, 1998, pp. 242–249.

[26] Park K. C., Felippa C. A. and Gumaste U.A., "A localized version of the method of Lagrange multipliers and its applications." *Computational Mechanics*, Vol. 24, 2000, pp. 476–490.

[27] Fortin M. and Glowinski R., "Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary- Value Problems." North-Holland, Amsterdam, the Netherlands, 1983.

[28] Gill P. E., Murray W., Saunders M. A. and Wright M. H., "Sequential quadratic programming methods for nonlinear programming." In E.J. Haug, editor, *Computer-Aided Analysis and Optimization of Mechanical System Dynamics*, Springer-Verlag, Berlin, Heidelberg, 1984, pp. 679–697.

[29] Bayo E., Garc'ıa de Jal'on J. and Serna M. A., "A modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems." *Computer Methods in Applied Mechanics and Engineering*, Vol. 71, 1998, pp. 183–195.

[30] Bayo E., Garc´ıa de Jal´on J., Avello A. and Cuadrado J., "An efficient computational method for real time multibody dynamic simulation in fully Cartesian coordinates." *Computer Methods in Applied Mechanics and Engineering*, Vol. 92, 1991, pp. 377–395.

[31] Bayo E. and Ledesma R., "Augmented Lagrangian and mass-orthogonal projection methods for constrained multibody dynamics." *Nonlinear Dynamics*, Vol. 9, 1996, pp. 113–130.

[32] Bottasso C.L., Bauchau O.A. and Cardona A., "Timestep-size-independent conditioning and sensitivity to perturbations in the numerical solution of index three differential algebraic equations." *SIAM Journal on Scientific Computing*, Vol. 29, No. 1, 2007, pp. 397–414.

[33] *Linear Algebra PACKage (LAPACK)*. http://www. netlib.org/lapack, 2014.

[34] *Scalable Linear Algebra PACKage (ScaLAPACK)*. http://www.netlib.org/scalapack, 2014.

[35] *Intel Math Kernel Library (Intel MKL) 11.0.* http://software.intel.com/en-us/intel-mkl, 2014.

[36] Ahmad S., Irons B. H. and Zienkiewicz, O. C., "Analysis of thick and thin shell structures by curved finite elements." *International Journal for Numerical Methods in Engineering*, Vol. 2, 1970, pp. 419–451.

[37] Ashwell D.G. and Gallagher R.H., *Finite Elements for Thin Shells and Curved Members*. JohnWiley & Sons, New York, 1976.

[38] Ugural A.C., *Stresses in Plates and Shells*. McGraw-Hill Book Company, New-York, second edition, 1999.