Available online at www.sciencedirect.com
ScienceDirect
Journal of the Franklin Institute 347 (2010) 53–68
Journal of The Franklin Institute
www.elsevier.com/locate/jfranklin

Parallel computation approaches for flexible multibody dynamics simulations

Olivier A. Bauchau

Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA, USA

Received 14 August 2009; received in revised form 24 September 2009; accepted 2 October 2009

Abstract

Finite element based formulations for flexible multibody systems are becoming increasingly popular and as the complexity of the configurations to be treated increases, so does the computational cost. It seems natural to investigate the applicability of parallel processing to this type of problems; domain decomposition techniques have been used extensively for this purpose. In this approach, the computational domain is divided into non-overlapping sub-domains, and the continuity of the displacement field across sub-domain boundaries is enforced via the Lagrange multiplier technique. In the finite element literature, this approach is presented as a mathematical algorithm that enables parallel processing. In this paper, the divided system is viewed as a flexible multibody system, and the sub-domains are connected by kinematic constraints. Consequently, all the techniques applicable to the enforcement of constraints in multibody systems become applicable to the present problem. In particular, it is shown that a combination of the localized Lagrange multiplier technique with the augmented Lagrange formulation leads to interesting solution strategies.

Keywords: Multibody dynamics; Parallel algorithms; Constrained dynamical systems

1. Introduction

It is often imperative to simulate the dynamic response of multibody systems in real time. In response to this challenge, efficient algorithms have been developed for modeling multirigid body dynamical systems presenting chain or tree-like topologies. By taking

E-mail address: olivier.bauchau@aerospace.gatech.edu

0016-0032/$32.00
doi:10.1016/j.jfranklin.2009.10.001

advantage of specific system topologies, parallel $\mathcal{O}(\log(n))$ formulations have been developed by Fijany et al. [1]. Anderson [2] developed an $\mathcal{O}(n)$ parallel algorithm for the simulation of systems presenting tree topologies. Featherstone proposed a divide-and-conquer parallel $\mathcal{O}(\log(n))$ strategy for articulated bodies [3], which he then generalized to tree and loop configurations [4]. Anderson and Duan [5] considered general systems of rigid bodies, which may contain arbitrary joint types, multiple branches, and/or closed loops. Their proposed scheme allows a substantially higher degree of parallelization than is generally obtainable using the more conventional recursive $\mathcal{O}(n)$ procedures. Critchley and Anderson [6] further refined the approach by introducing a new recursive coordinate reduction parallel algorithm that provides improved computation performance.

In recent years, the field of flexible multibody dynamics has embraced the finite element approach for modeling flexible structures, as described in the textbook of Géradin and Cardona [7] and numerous references therein. In those formulations, the kinematic constraints found in multibody systems are typically enforced via the Lagrange multiplier technique, leading to differential algebraic governing equations. In most cases, complex flexible multibody systems do not present the tree-like topology that characterizes traditional multirigid body dynamical systems. For instance, when modeling a shell structure, all degrees of freedom are coupled through the two-dimensional finite element mesh. On the other hand, the finite element modeling of flexible multibody systems involves orders of magnitude more degrees of freedom than those typically involved in the modeling of multirigid body systems. Consequently, detailed and accurate time simulation of flexible multibody systems using the finite element approach can typically not be performed in real time. The use of a finite element approach, however, does not preclude the development of parallel solution algorithms.

In fact, a voluminous body of literature deals with the development and implementation of parallel computational strategies for finite element models. A complete review of the field is beyond the scope of this paper, but the many approaches that have been proposed fall into the following categories.

First, because the single most expensive operation in a finite element simulation is the factorization of the stiffness matrix, strategies for parallel implementation of factorization algorithms have been developed. The use of vector computers was proposed to speed up the factorization algorithm. This approach is very robust, but seems to have met limited success in terms of scalability [8]. Multi-frontal solvers [9] have also been proposed for the task of factorization [10]. This approach seems to be very arduous to implement; furthermore, for multibody systems, the presence of Lagrange multipliers as additional solution variables would require the development of appropriate multi-frontal solvers. This helps explain why this approach does not seem to have been applied to the types of problems appearing in flexible multibody dynamics.

The second approach relies on iterative solvers, such as the conjugate gradient or generalized minimal residual algorithms [11]. This approach has been applied to the solution of structural dynamics problems. It must be noted, however, that their efficiency crucially depends on system conditioning. Hence, good pre-conditioners are required for the efficient implementation of this approach. Flexible multibody systems often involve rigid body modes associated with zero frequencies, and the Lagrange multipliers used to enforce kinematic constraints introduce numerous "infinite frequencies." Consequently, flexible multibody problems are typically ill-conditioned, more so than their structural dynamics counterparts, and the use of iterative solvers does not seem to be desirable.

Finally, domain decomposition techniques have been the approach of choice for the last two decades. In particular, the FETI method developed by Farhat and coworkers has received considerable attention [12–15]. A distinctive feature of this approach is that an auxiliary problem appears as a byproduct of the solution process. It is based on the rigid body modes of the floating sub-domains, and because its size is small compared to that of the overall problem, it is called the "coarse problem." For plate and shell problems, imposing the continuity of the transverse displacement at sub-domain cross-points, also called "corner points," is found to be beneficial and is enforced via an additional set of Lagrange multipliers. This leads to a two-level procedure [16–19]. Finally, the second generation, dual–primal FETI method [20–23] combines many of the techniques developed earlier in a unified manner.

Limited work has been done to apply the algorithms described in the previous paragraphs to flexible multibody systems. Chiou et al. [24] combined the null space formulation with a preconditioned conjugate gradient algorithm to obtain a natural partitioning scheme for multibody systems. Coulon et al. [25] investigated the application of the FETI method to flexible multibody systems. Finally, Quaranta [26] used a domain decomposition method based on the Schur complement matrix to perform parallel simulation of multibody systems.

The FETI method is an approach by which the computational domain is divided into non-overlapping sub-domains. At the interface between the sub-domains, kinematic constraints are imposed to enforce the continuity of the displacement field over the entire structure. These kinematic constraints are enforced via fields of Lagrange multipliers that act at the interface between the sub-domains and can be interpreted as the interface connection forces. The method then proceeds in two steps. First, an interface problem is solved that yields the Lagrange multipliers. Second, the sub-domains are now independent structures subjected to known boundary forces, the fields of Lagrange multipliers. The displacement fields in each sub-domain can be computed in parallel because each sub-domain is now independent of the others.

In the literature cited above, the FETI method is seen as a purely computational algorithm, and researchers have focused on the efficiency and scalability of the approach. In this paper, domain decomposition methods are investigated from the viewpoint of constrained dynamical system. Once the system is decomposed into sub-domains for the purpose of parallel computation, these sub-domains can be viewed as flexible bodies connected by kinematic constraints, as is typically found in flexible multibody systems. Clearly, the tools and techniques developed for the analysis of such systems become relevant. In rigid and flexible multibody dynamics, the method used to enforce the kinematic constraints plays a pivotal role in the formulation. Bauchau and Laulusa [27,28] have reviewed the literature on this topic. Some widely used techniques are not desirable in this case. For instance, many approaches seek a minimum set of variable and eliminate the Lagrange multipliers from the formulation. While this might be a computationally efficient way to proceed for sequential processing, it is not an option for parallel computations. On the other hand, the augmented Lagrangian approach is a well established procedure for solving constrained dynamical systems and seems to be very much applicable to the problem at hand.

The present paper layout is as follows. A general approach to domain decomposition is presented in Section 2. The approach is novel because two different Lagrange multiplier techniques are used to enforce the continuity condition between sub-domain: the global

and the local multiplier techniques. The governing equations of the system are derived in Section 3. It is shown that the augmented Lagrange procedure can be used when constraints are enforced via local Lagrange multipliers, but is not desirable when used in conjunction with global Lagrange multipliers. Two extreme cases are investigated next: the case when all Lagrange multipliers are global, and that when all are local, see Sections 4 and 5, respectively. The conditioning of the interface problem is studied for the latter case. Finally, Section 6 investigates the mixed, global–local Lagrange multiplier approach.

2. Domain decomposition

Consider the planar solid depicted in Fig. 1. To develop a parallel solution algorithm for this problem, the solid is partitioned into N_s non-overlapping sub-domains. Each of these sub-domains could themselves be multibody systems comprising both elastic elements and nonlinear kinematic constraints. For convenience, Fig. 1 depicts a planar system, but all the developments presented here are applicable to general, three-dimensional problems. The degrees of freedom (dofs) for each sub-domain are collected in arrays denoted by $\underline{u}_i$, $i = 1, 2, \ldots N_s$. Array $\underline{u}_i$ stores the dofs of sub-domain i, i.e., the displacement components at all the nodes of the sub-domain. This array is of size n_i^u, which is the total number of dofs for sub-domain i. The global array of dofs is defined as

$$\underline{u}^T = \{\underline{u}_1^T, \underline{u}_2^T, \ldots \underline{u}_{N_s}^T\}. \tag{1}$$

Array $\underline{u}$ is of size $n^u = \sum_{i=1}^{N_s} n_i^u$, which is the total number of dofs for the complete structure. As 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: all interface dofs appear twice or more times. The variables stored in array $\underline{u}$ should be called "generalized coordinates" because they do not form a minimum set, but the term "dofs," more widely used in the finite element literature, will be used here.

The dofs of each sub-domain can be split into two mutually exclusive groups, the internal and boundary dofs, respectively. The boundary dofs are those that are exposed in

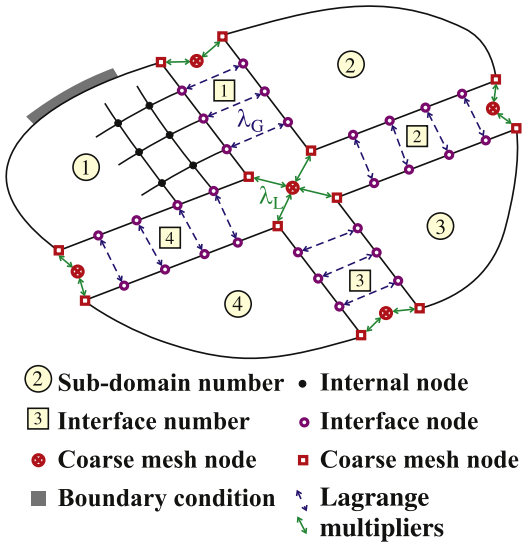

Fig. 1. Planar solid separated into four non-overlapping sub-domains.

the process of dividing the original problem into sub-domains, whereas the remaining dofs are internal. Kinematic constraints will be imposed at the boundary nodes to enforce the continuity of the displacement field, thereby ensuring that the behavior of the connected sub-domains is identical to that of the original, un-partitioned solid.

The continuity of the displacement field across sub-domain boundaries is enforced by imposing linear constraints, the equality of the dofs of corresponding nodes in adjacent sub-domains. Fig. 1 illustrates this approach: the double-headed dashed arrows symbolize the Lagrange multipliers used to impose the linear constraints on the dofs of nodes belonging to adjacent sub-domains.

This approach corresponds to the classical Lagrange multiplier method, but it is not the only possible approach. An alternative implementation is to first define independent interface nodes, then imposing two kinematic constraints: the displacement components at the boundary nodes in the two sub-domains adjacent to the interface must equal those at the independent interface nodes. In finite element formulations, this approach has been used to enforce the continuity of displacement fields between adjacent incompatible elements [29]. The same approach, called "localized version of the method of Lagrange multipliers," has been advocated by Park et al. [30,31]. Fig. 1 also illustrates this approach: the double-headed solid arrows symbolize the two sets of Lagrange multipliers used to impose the equality of the displacement components at nodes belonging to adjacent sub-domains with those of the independent interface nodes. At a corner node, i.e., at sub-domain cross-points, a single interface node is defined and Lagrange multipliers are used to enforce equality of the interface node dofs with those of all adjacent nodes. This feature of local Lagrange multipliers was emphasized by Park et al. [31].

In this paper, continuity conditions enforced through the classical Lagrange multiplier method will be called "global constraints" and the associated Lagrange multiplier "global Lagrange multipliers;" in contrast, when using the localized version of the method of Lagrange multipliers, the corresponding quantities are called "local constraints" and "local Lagrange multipliers," respectively.

The configuration shown in Fig. 1 is just an illustration. Clearly, the continuity conditions could be enforced exclusively through global Lagrange multipliers, exclusively through local Lagrange multipliers, or through a combination of both. These three approaches will be investigated in this paper. As stated in Fig. 1, the independent interface nodes will be called the "coarse mesh nodes," because they define a coarse mesh for the problem.

Boolean matrices can be defined to partition the dofs listed in array $\underline{u}_i$. For instance, array $\underline{\ell}_i$, of size n_i^{ℓ}, stores the internal dofs of sub-domain i,

$$\underline{\ell}_i = L_i \underline{u}_i, \tag{2}$$

where L_i is the Boolean matrix that extract the internal degrees of freedom from array $\underline{u}_i$.

The boundary dofs to be connected to coarse mesh dofs through local Lagrange multipliers are denoted by $\underline{s}_i$ and represented by open square symbols in Fig. 1. The dofs of sub-domain i associated with these nodes are stored in array $\underline{s}_i$, of size n_i^s, and extracted by Boolean matrix S_i,

$$\underline{s}_i = S_i \underline{u}_i. \tag{3}$$

The remaining boundary dofs of sub-domain i, denoted by $\underline{r}_i$, are represented by open circle symbols in Fig. 1, and are each associated with a specific interface; the dofs

associated with interface j, denoted by $\underline{r}_{ij}$, are extracted by Boolean matrix R_{ij},

$$\underline{r}_{ij} = R_{ij}\underline{u}_i. \tag{4}$$

Finally, array $\underline{c}$ collects the dofs at the N_c coarse mesh nodes and those nodes to be connected to sub-domain i are stored in array $\underline{c}_i$ of size n_i^c and extracted by Boolean matrix C_i,

$$\underline{c}_i = C_i\underline{c}. \tag{5}$$

To enforce the continuity conditions, two types of constraints are defined. First, at the boundary nodes to be connected to coarse mesh nodes, the local continuity conditions are $\underline{\mathcal{C}}_{Li} = S_i\underline{u}_i - C_i\underline{c}_i = 0$ for $i = 1, 2, \ldots, N_s$, and are written in a compact form as

$$\underline{\mathcal{C}}_L = S\underline{u} - C\underline{c} = 0, \tag{6}$$

where the subscript $(\cdot)_L$ indicates that these are local constraints, $S = \text{diag}(S_\alpha)$, $\underline{\mathcal{C}}_L^T = \{\underline{\mathcal{C}}_{L1}^T, \underline{\mathcal{C}}_{L2}^T, \ldots, \underline{\mathcal{C}}_{LN_s}^T\}$, and $C^T = [C_1^T, C_2^T, \ldots, C_{N_s}^T]$. In this paper, the notation $\text{diag}(M_\alpha)$ is used to denote a block diagonal matrix whose diagonal blocks are $M_1, M_2, \ldots, M_{N_s}$. Because the constraints are linear, matrices S and C form the constraint Jacobian matrix $S = \partial\underline{\mathcal{C}}_L/\partial\underline{u}$ and $C = \partial\underline{\mathcal{C}}_L/\partial\underline{c}$. Note that $S_i^T S_i$ is a diagonal matrix whose diagonal entries are unit for each boundary dof connected to a coarse mesh node and zero for all other entries.

Next, the global continuity conditions at the remaining boundary nodes along interface j are $R_{pj}\underline{u}_p - R_{qj}\underline{u}_q = 0$, where p and q are the indexes of the two sub-domains adjacent to interface j. When expressed for all interfaces, these continuity conditions can be written in a compact form as

$$\underline{\mathcal{C}}_G = R\underline{u} = 0, \tag{7}$$

where the subscript $(\cdot)_G$ indicates that these are global constraints. Here again, because the constraints are linear, matrix R forms the constraint Jacobian matrix $R = \partial\underline{\mathcal{C}}_G/\partial\underline{u}$.

3. Governing equations

The total strain energy in the structure, A, can be evaluated by summing the strain energies, A_i, of the various sub-domains

$$A = \sum_{i=1}^{N_s} A_i = \frac{1}{2}\sum_{i=1}^{N_s} \underline{u}_i^T K_i \underline{u}_i = \frac{1}{2}\underline{u}^T \text{diag}(K_\alpha)\underline{u}, \tag{8}$$

where K_i is the stiffness matrix for sub-domain i. The stiffness matrix of each sub-domain is obtained from the stiffness matrices of each finite element of the sub-domain using standard finite element assembly procedures. The block diagonal, global stiffness matrix of the system, $\text{diag}(K_\alpha)$, is of size $(n^u \times n^u)$. If the original solid is suitably constrained by a set of boundary conditions that prevent overall rigid body motions, the global stiffness matrix will not be singular. This property is not shared by the stiffness matrices of individual sub-domains: indeed, due to the partitioning of the solid into possibly unconstrained or "floating sub-domains," the stiffness matrix of each sub-domain is potentially singular.

Domain decomposition methods exploit the special structure of the global stiffness matrix. Because it is block-diagonal, its inverse is readily computed as $\text{diag}(K_\alpha^{-1})$. Since the

sub-domain stiffness matrices are independent of each other, their inverses, K_α^{-1}, can be computed independently by N_s independent processors.

The total potential of the externally applied loads, Φ, is found by summing up the potentials for each of the sub-domains

$$\Phi = \sum_{i=1}^{N_s} \Phi_i = -\sum_{i=1}^{N_s} \underline{u}_i^T \underline{Q}_i = -\underline{u}^T \underline{Q}, \tag{9}$$

where $\underline{Q}_i$ is the load array for sub-domain i. The system's global load array is defined as $\underline{Q}^T = \{\underline{Q}_1^T, \underline{Q}_2^T, \ldots, \underline{Q}_{N_s}^T\}$.

The kinematic constraints of the problem give rise to the potential of the constraints, V^c, which is written as

$$V^c = \underline{\lambda}_G^T \mathrm{diag}(p_{G\alpha}) \underline{\mathcal{C}}_G + \frac{1}{2} \underline{\mathcal{C}}_G^T \mathrm{diag}(\overline{p}_{G\alpha}) \underline{\mathcal{C}}_G + \sum_{i=1}^{N_s} \left[p_{Li} \underline{\lambda}_{Li}^T \underline{\mathcal{C}}_{Li} + \frac{1}{2} \overline{p}_{Li} \underline{\mathcal{C}}_{Li}^T \underline{\mathcal{C}}_{Li} \right]. \tag{10}$$

In this expression, $\underline{\lambda}_G$ and $\underline{\lambda}_L^T = \{\underline{\lambda}_{L1}^T, \underline{\lambda}_{L2}^T, \ldots, \underline{\lambda}_{LN_s}^T\}$ are the global and local Lagrange multipliers used to enforce the global and local constraints, respectively. Matrix $\mathrm{diag}(p_{G\alpha})$ is a block diagonal matrix of scaling factors for the global Lagrange multipliers; matrix p_{Gi} is itself a diagonal matrix, which means that p_{Gi} is the scaling factor for sub-domain i. Similarly, p_{Li} are the scaling factors for the local Lagrange multipliers. Coefficients $\overline{p}_{Gi}$ and $\overline{p}_{Li}$ are penalty coefficients for the global and local constraints, respectively. The first and third terms of Eq. (10) are the classical potential associated with Lagrange multipliers, while the second and fourth terms are characteristic of penalty methods.

This approach, which combines the traditional Lagrange multiplier approach with the penalty method is known as the augmented Lagrangian formulation and has been studied extensively [32,33]. It is an effective approach for the enforcement of kinematic constraints in multibody dynamics, as proposed by Bayo et al. [34,35]. Furthermore, scaling of the Lagrange multipliers and the addition of the penalty terms was shown to help the solution of differential algebraic [36,37].

The total potential energy of the system, $\Pi = A + \Phi + V^c$, is found by combining Eqs. (8)–(10) to find

$$\begin{aligned} \Pi = {} & \frac{1}{2} \underline{u}^T \mathrm{diag}(K_\alpha) \underline{u} - \underline{u}^T \underline{Q} + \underline{\lambda}_G^T \mathrm{diag}(p_{G\alpha}) \underline{\mathcal{C}}_G + \frac{1}{2} \underline{\mathcal{C}}_G^T \mathrm{diag}(\overline{p}_{G\alpha}) \underline{\mathcal{C}}_G \\ & + \sum_{i=1}^{N_s} \left[p_{Li} \underline{\lambda}_{Li}^T \underline{\mathcal{C}}_{Li} + \frac{1}{2} \overline{p}_{Li} \underline{\mathcal{C}}_{Li}^T \underline{\mathcal{C}}_{Li} \right]. \end{aligned} \tag{11}$$

The principle of minimum total potential energy then yields the equations of motion as

$$\begin{bmatrix} K^* & -\mathrm{diag}(\overline{p}_{L\alpha} S_\alpha^T) C & R^T \mathrm{diag}(p_{G\alpha}) & \mathrm{diag}(p_{L\alpha} S_\alpha^T) \\ -C^T \mathrm{diag}(\overline{p}_{L\alpha} S_\alpha) & C^T \mathrm{diag}(\overline{p}_{L\alpha}) C & 0 & -C^T \mathrm{diag}(p_{L\alpha}) \\ \mathrm{diag}(p_{G\alpha}) R & 0 & 0 & 0 \\ \mathrm{diag}(p_{L\alpha} S_\alpha) & -\mathrm{diag}(p_{L\alpha}) C & 0 & 0 \end{bmatrix}$$

$$\begin{Bmatrix} \underline{u} \\ \underline{c} \\ \underline{\lambda}_G \\ \underline{\lambda}_L \end{Bmatrix} = \begin{Bmatrix} \underline{Q} \\ 0 \\ 0 \\ 0 \end{Bmatrix}, \tag{12}$$

where $K^* = \text{diag}(K_\alpha) + R^T \text{diag}(\overline{p}_{G\alpha})R + \text{diag}(S_\alpha^T \overline{p}_{L\alpha} S_\alpha)$. The solution of this system of equations yields the complete solution of the problem.

The computational efficiency of domain decomposition methods stems from the fact that the lead matrix of this system is block diagonal. Unfortunately, with the present formulation, K^* is the sum of three matrices: the first and third matrices are block diagonal, matrix $R^T \text{diag}(\overline{p}_{G\alpha})R$ is not. From now on, the penalty terms associated with global constraints will be ignored, i.e., $\overline{p}_{Gi} = 0$. The lead matrix now reduces to $K^* = \text{diag}(K_\alpha + S_\alpha^T \overline{p}_{L\alpha} S_\alpha)$, which is a block diagonal matrix. For simplicity, the remaining scaling and penalty coefficients are selected as $p_i = \overline{p}_{Li} = p_{Li} = p_{Gi}$, where p_i is the common value of the penalty and scaling factors for sub-domain i.

4. All Lagrange multipliers are global

This section focuses on a special case of the general formulation developed in Section 3. The coarse mesh is assumed to vanish, which implies that all constraints are global. The equations of motion are easily obtained by eliminating $\underline{c}$, the coarse mesh, $\underline{\lambda}_L$, the associated Lagrange multipliers, and matrices S and C, defining the local constraints, see Eqs. (5) and (3). Eq. (12) now reduces to

$$\begin{bmatrix} \text{diag}(K_\alpha) & R^T \text{diag}(p_\alpha) \\ \text{diag}(p_\alpha)R & 0 \end{bmatrix} \begin{Bmatrix} \underline{u} \\ \underline{\lambda}_G \end{Bmatrix} = \begin{Bmatrix} \underline{Q} \\ 0 \end{Bmatrix}. \tag{13}$$

This approach corresponds to the well-known FETI method. The solution strategy proceeds as follows. First, the sub-domain dofs are evaluated as $\underline{u} = \text{diag}(K_\alpha^{-1})[\underline{Q} - R^T \text{diag}(p_\alpha)\underline{\lambda}_G]$. Introducing this result in the global constraint then leads to

$$F_G \underline{\lambda}_G = \underline{Q}_G, \tag{14}$$

where $F_G = \text{diag}(p_\alpha)R\,\text{diag}(K_\alpha^{-1})R^T \text{diag}(p_\alpha)$ is the interface flexibility matrix and $\underline{Q}_G = \text{diag}(p_\alpha)R\,\text{diag}(K_\alpha^{-1})\underline{Q}$ the interface load array. In this approach, the sub-domain stiffness matrices must be invertible, i.e., K_α^{-1} must exist. In the FETI method, this problem is overcome by defining a coarse problem based on the rigid body motions of the sub-domains. This approach will not be developed further here as all the details can be found in the literature.

5. All Lagrange multipliers are local

This section focuses on a special case of the general formulation developed in Section 3. All constraints are assumed to be local, which means that the coarse mesh is defined along the entire interface. The equations of motion are easily obtained by eliminating $\underline{\lambda}_G$ and

matrix R defining the global constraints. Eq. (12) now reduce to

$$\begin{bmatrix} \text{diag}(\overline{K}_\alpha) & -\text{diag}(\overline{S}_\alpha^T)C & \text{diag}(\overline{S}_\alpha^T) \\ -C^T\,\text{diag}(\overline{S}_\alpha) & C^T\,\text{diag}(p_\alpha)C & -C^T\,\text{diag}(p_\alpha) \\ \text{diag}(\overline{S}_\alpha) & -\text{diag}(p_\alpha)C & 0 \end{bmatrix} \begin{Bmatrix} \underline{u} \\ \underline{c} \\ \underline{\lambda}_L \end{Bmatrix} = \begin{Bmatrix} \underline{Q} \\ 0 \\ 0 \end{Bmatrix}, \tag{15}$$

where $\overline{K}_i = K_i + p_i S_i^T S_i$ and $\overline{S}_i = p_i S_i$. Because $S_i^T S_i$ is a diagonal matrix, the penalty coefficient, p_i, is added to each diagonal entry of K_i corresponding to a boundary dof to be connected to a coarse mesh node. Physically, this corresponds to adding springs of stiffness constant p_i to each of these dofs. Whereas stiffness matrix K_i is singular for floating domains, matrix $\overline{K}_i$ is not, provided that a sufficient number of coarse mesh nodes are associated with each sub-domain.

Two sets of variables, $\underline{c}$ and $\underline{\lambda}_L$, now define the interface problem. First, the sub-domain displacement dofs, $\underline{u}$, are evaluated by solving the first equation of system (15). Introducing the result in the other two equations then yields the interface problem

$$\begin{bmatrix} C^T\,\text{diag}(p_\alpha - \overline{F}_\alpha)C & -C^T\,\text{diag}(p_\alpha - \overline{F}_\alpha) \\ \text{diag}(p_\alpha - \overline{F}_\alpha)C & \text{diag}(\overline{F}_\alpha) \end{bmatrix} \begin{Bmatrix} \underline{c} \\ \underline{\lambda}_L \end{Bmatrix} = \begin{Bmatrix} C^T \\ I \end{Bmatrix} \text{diag}(\overline{S}_\alpha \overline{K}_\alpha^{-1})\underline{Q}, \tag{16}$$

where $\overline{F}_i = \overline{S}_i \overline{K}_i^{-1} \overline{S}_i^T$ is the coarse mesh flexibility matrix for sub-domain i.

Solving the second equation of system (16) for the coarse mesh dofs and introducing the result into the first equation yields

$$\sum_{i=1}^{N_s} C_i^T p_i \underline{\lambda}_{Li} = 0. \tag{17}$$

This equation simply expresses the fact that the pair of Lagrange multipliers enforcing the equality of the dofs of two adjacent sub-domains with a common coarse mesh dof must be equal and of opposite sign, as expected from Newton's third law.

Because the local Lagrange multipliers are sub-domain variables, it is convenient to combine the sub-domain dofs and local Lagrange multipliers into a single array, $\check{\underline{u}}_i^T = \{\underline{u}_i^T, \underline{\lambda}_{Li}^T\}$, and system (15) further reduces to

$$\begin{bmatrix} \text{diag}(\check{K}_\alpha) & -\text{diag}(\check{S}_\alpha^T)C \\ -C^T\,\text{diag}(\check{S}_\alpha) & C^T\,\text{diag}(p_\alpha)C \end{bmatrix} \begin{Bmatrix} \underline{u} \\ \underline{c} \end{Bmatrix} = \begin{Bmatrix} \underline{Q} \\ 0 \end{Bmatrix}, \tag{18}$$

where $\check{S}_i = [\overline{S}_i \;\; p_i]$, $\check{\underline{Q}}^T = \{\underline{Q}^T, 0\}$, and

$$\check{K}_i = \begin{bmatrix} \text{diag}(\overline{K}_i) & \text{diag}(\overline{S}_i^T) \\ \text{diag}(\overline{S}_i) & 0 \end{bmatrix}. \tag{19}$$

Solving the first equation of system (18) for the sub-domain variables, $\check{\underline{u}}$, and introducing the result into the second equation yields the coarse mesh dofs as

$$\check{K}_c \underline{c} = \check{\underline{Q}}_c, \tag{20}$$

where $\check{K}_c$ and $\underline{\check{Q}}_c$ are the interface problem stiffness matrix and load array, respectively. The stiffness matrix obtained by assembling the sub-domain coarse mesh stiffness matrices, $k_i = p_i - \check{S}_i \check{K}_i^{-1} \check{S}_i^T$, as $\check{K}_c = \sum_{i=1}^{N_s} C_i^T k_i \; C_i$. This is a typical finite element assembly procedure where C_i is the connectivity matrix for the coarse problem, as expected from its definition, Eq. (5). The load array is obtained by assembling the sub-domain coarse mesh load arrays, $\underline{f}_i = \check{S}_i \check{K}_i^{-1} \underline{\check{Q}}_i$, as $\underline{\check{Q}}_c = \sum_{i=1}^{N_s} C_i^T \underline{f}_i$.

5.1. Effect of the penalty terms

Combining the sub-domain and coarse mesh dofs into a single array, system (15) is now recast as

$$\begin{bmatrix} \hat{K} & \hat{B}^T \operatorname{diag}(p_\alpha) \\ \operatorname{diag}(p_\alpha)\hat{B} & 0 \end{bmatrix} \begin{Bmatrix} \underline{\hat{u}} \\ \underline{\lambda}_L \end{Bmatrix} = \begin{Bmatrix} \underline{\hat{Q}} \\ 0 \end{Bmatrix}, \tag{21}$$

where $\hat{K} = \operatorname{diag}(K_\alpha) + \hat{B}^T \operatorname{diag}(p_\alpha)\hat{B}$, $\hat{B} = [\operatorname{diag}(S_\alpha) \;\; -C]$, $\underline{\hat{u}}^T = \{\underline{u}^T, \underline{c}^T\}$, and $\underline{\hat{Q}}^T = \{\underline{Q}^T, 0\}$. To define an interface problem, the sub-domain displacements and coarse mesh are computed from the first equation and the result is introduced into the second equation to find

$$F_L \underline{\lambda}_L = \operatorname{diag}(p_\alpha)\hat{B}\hat{K}^{-1}\underline{\hat{Q}}, \tag{22}$$

where the flexibility matrix is

$$F_L = \operatorname{diag}(p_\alpha)\hat{B}\hat{K}^{-1}\hat{B}^T \operatorname{diag}(p_\alpha). \tag{23}$$

It is interesting to look at the behavior of the flexibility matrix in the limiting case where $p_i \to \infty$, i.e.,

$$\lim_{p\to\infty} F_L = \lim_{p\to\infty} \operatorname{diag}(p_\alpha)\hat{B}\hat{K}^{-1}\hat{B}^T \operatorname{diag}(p_\alpha) = \operatorname{diag}(p_\alpha)\hat{B}(\hat{B}^T\hat{B})^{-1}\hat{B}^T. \tag{24}$$

Consider now the following singular value decomposition of matrix $\hat{B}^T$, assumed to be of full rank,

$$\hat{B}^T = U \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^T, \tag{25}$$

where U and V are orthogonal matrices, and Σ the diagonal matrix of singular values. Matrices $(\hat{B}^T\hat{B})$ and its Moore–Penrose inverse, denoted by $(\hat{B}^T\hat{B})^\dagger$, now become

$$(\hat{B}^T\hat{B}) = U \begin{bmatrix} \Sigma^2 & 0 \\ 0 & 0 \end{bmatrix} U^T, \quad (\hat{B}^T\hat{B})^\dagger = U \begin{bmatrix} \Sigma^{-2} & 0 \\ 0 & 0 \end{bmatrix} U^T, \tag{26}$$

respectively. Replacing the inverse of matrix $(B^T B)$ by its Moore–Penrose inverse in Eq. (24), the limiting behavior of the flexibility matrix now becomes

$$\lim_{p\to\infty} F_L = \mathrm{diag}(p_\alpha) V [\Sigma \;\; 0] \begin{bmatrix} \Sigma^{-2} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^T = \mathrm{diag}(p_\alpha). \tag{27}$$

If the penalty coefficients for all sub-domains are selected to be equal, the flexibility matrix converges to an ideally preconditioned, identity matrix as the magnitude of the penalty coefficient increases. This observation is very important if the interface problem is solved using iterative methods. In this case, the conditioning of the system matrix dictates the convergence rate of the method. If the system matrix has a condition number of unity, iterative methods will converge in one single iteration. Clearly, the use of penalty terms provides the ideal conditioning of the flexibility matrix.

When performing an iterative solution of the interface problem, Eq. (22), a "matrix times vector" operation of the type $\underline{x} = F_L \underline{a}$, where $\underline{a}$ is an arbitrary array, will be required. In turn, this requires the ability to evaluate $\underline{y} = \hat{K}^{-1} \underline{b}$, which corresponds to the solution of the linear system $\hat{K}\underline{y} = \underline{b}$, where matrix $\hat{K}$ is not block-diagonal. This problem is recast as

$$\begin{bmatrix} \mathrm{diag}(\overline{K}_\alpha) & -\mathrm{diag}(\overline{S}_\alpha^T) C \\ -C^T \mathrm{diag}(\overline{S}_\alpha) & C^T \mathrm{diag}(p_\alpha) C \end{bmatrix} \begin{Bmatrix} \underline{u} \\ \underline{c} \end{Bmatrix} = \begin{Bmatrix} \underline{q}_u \\ \underline{q}_c \end{Bmatrix}. \tag{28}$$

Because the lead matrix of this system is block-diagonal, an efficient procedure is to solve for the sub-domain displacements first, then introduce the results in the second equation to defined a coarse mesh problem of the form

$$K_c \underline{c} = \underline{Q}_c, \tag{29}$$

where K_c and $\underline{Q}_c$ are the interface problem stiffness matrix and load array, respectively. The stiffness matrix obtained by assembling the sub-domain coarse mesh stiffness matrices, $k_i = p_i - \overline{S}_i \overline{K}_i^{-1} \overline{S}_i^T$, as $K_c = \sum_{i=1}^{N_s} C_i^T k_i C_i$. The load array is obtained by assembling the sub-domain coarse mesh load arrays, $\underline{f}_i = \overline{S}_i \overline{K}_i^{-1} \underline{q}_{ui}$, as $\underline{Q}_c = \underline{q}_c - \sum_{i=1}^{N_s} C_i^T \underline{f}_i$.

In conclusion, the use of penalty terms in the formulation of the constraints, Eq. (10), leads to a natural conditioning of the interface problem, see Eq. (27). This has profound implications on the efficiency of iterative methods used for the solution of the interface problem. On the other hand, this advantage comes at the expense of the solution of the coarse mesh problem, Eq. (29). When all Lagrange multipliers are local, as discussed in this section, the size of the coarse mesh problem could be large (it is not necessarily a "coarse mesh"). The advantage of the natural conditioning of the system stemming from the use of penalty terms is offset by the cost of solving for a "large" coarse mesh problem. This observation prompts the investigation of the mixed case, where some of the constraints are local, and others are global constraints.

6. Mixed local–global Lagrange multipliers

In this section, the strategy illustrated in Fig. 1 is investigated. Some of the constraints used to enforce the continuity of the displacement fields across sub-domain interfaces are local, and the remaining are global. The governing equations of the problem are now given by Eq. (12).

Because the local Lagrange multipliers are sub-domain variables, it is convenient to combine the sub-domain dofs and local Lagrange multipliers into a single array,

$$\begin{bmatrix} \mathrm{diag}(\breve{K}_\alpha) & -\mathrm{diag}(\breve{S}_\alpha^T)C & \breve{R}^T \\ -C^T\,\mathrm{diag}(\breve{S}_\alpha) & C^T\,\mathrm{diag}(p_\alpha)C & 0 \\ \breve{R} & 0 & 0 \end{bmatrix} \begin{Bmatrix} \breve{\underline{u}} \\ \underline{c} \\ \underline{\lambda}_G \end{Bmatrix} = \begin{Bmatrix} \breve{\underline{Q}} \\ 0 \\ 0 \end{Bmatrix}, \tag{30}$$

where $\breve{R} = [\mathrm{diag}(p_\alpha)R \;\; 0]$, and matrix $\breve{K}_i$ is defined by Eq. (19).

To define an interface problem, the sub-domain displacements and local Lagrange multipliers are computed from the first equation and the result is introduced into the second equation and third equation to find the interface system

$$\begin{bmatrix} \breve{K}_c & (\breve{R}\breve{C})^T \\ (\breve{R}\breve{C}) & -\breve{R}\,\mathrm{diag}(\breve{K}_\alpha^{-1})\breve{R}^T \end{bmatrix} \begin{Bmatrix} \underline{c} \\ \underline{\lambda}_G \end{Bmatrix} = \begin{Bmatrix} \breve{C}^T & \breve{\underline{Q}} \\ -\breve{R}\,\mathrm{diag}(\breve{K}_\alpha^{-1}) & \breve{\underline{Q}} \end{Bmatrix}, \tag{31}$$

where $\breve{C}^T = [p_1 C_1^T \breve{S}_1 \breve{K}_1^{-1}, p_2 C_2^T \breve{S}_2 \breve{K}_2^{-1}, \ldots, p_{N_s} C_{N_s}^T \breve{S}_{N_s} \breve{K}_{N_s}^{-1}]$. The coarse mesh stiffness matrix, $\breve{K}_c$, is obtained by assembling the sub-domain coarse mesh stiffness matrices, $k_i = p_i - \breve{S}_i \breve{K}_i^{-1} \breve{S}_i^T$, as $\breve{K}_c = \sum_{i=1}^{N_s} C_i^T k_i C_i$.

Finally, an interface problem expressed in terms of the sole global Lagrange multipliers is easily obtained by elimination to find

$$\breve{R}\breve{F}_G\breve{R}^T\underline{\lambda}_G = \breve{R}\breve{F}_G\breve{\underline{Q}}, \tag{32}$$

where the interface compliance matrix is $\breve{F}_G = \mathrm{diag}(\breve{K}_\alpha^{-1}) + \breve{C}\breve{K}_c^{-1}\breve{C}^T$. The inverse of the coarse problem stiffness matrix, $\breve{K}_c^{-1}$, is now part of the flexibility matrix of the interface problem, which it preconditions. When all constraints are enforced via local Lagrange multipliers, the penalty terms provide the ideal preconditioning because the interface flexibility matrix converge to the identity matrix for large penalty coefficients, see Section 5.1. On the other hand, when all constraints are enforced via global Lagrange multipliers, the penalty terms alter the bandwidth of the system matrix, destroying its block diagonal structure; it is not desirable to use penalty terms in this case. Finally, when both local and global Lagrange multipliers are used to enforce the constraints, the inverse of the coarse problem stiffness matrix must be evaluated. Because the coarse problem is of small size, this operation does not incur high computational cost, but it provides preconditioning for the iterative solution of the interface problem expressed in terms of the sole global Lagrange multipliers.

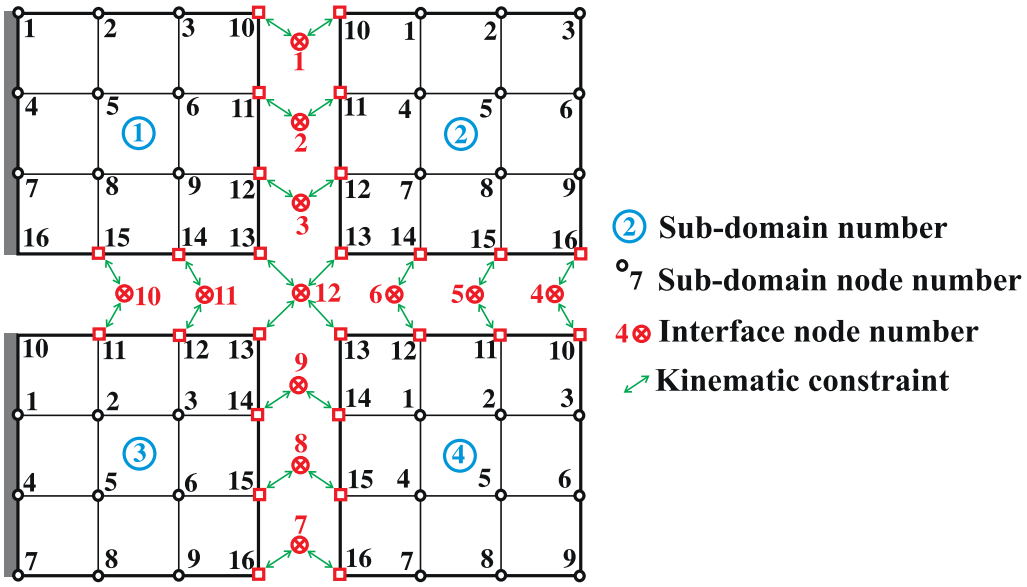

Fig. 2. Two dimensional, plane stress problem divided into four sub-domains.

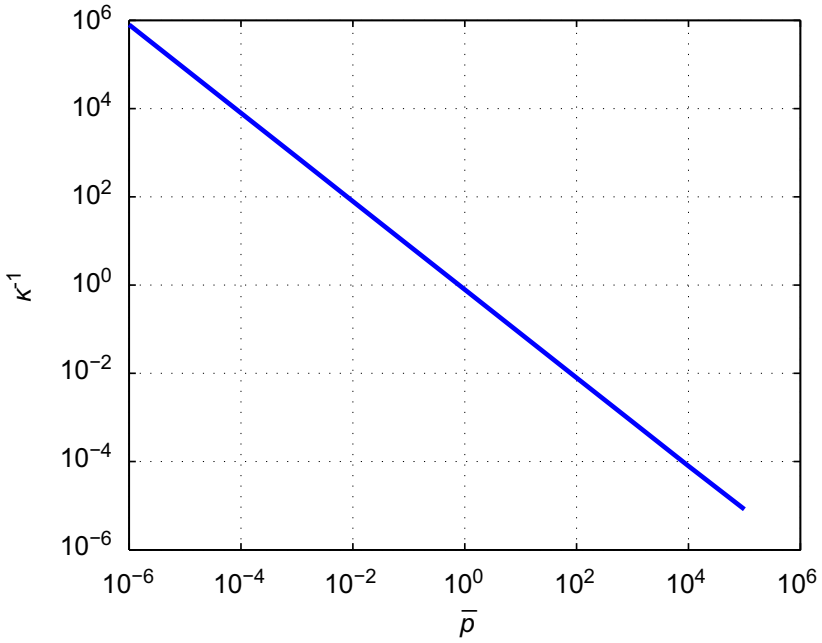

Fig. 3. Conditioning of the flexibility matrix measured as $\kappa(F_I) - 1$.

7. Numerical example

Consider the simple, two dimensional, plane stress elasticity problem depicted in Fig. 2. The complete problem, modeled by 36, four-noded finite elements, was divided into four sub-domains, each comprising nine elements. A total of 12 coarse mesh nodes, i.e., 24 interface degrees of freedom, are necessary to represent the interface displacement field. A total of 52 kinematic constraints are required to enforce the compatibility of the sub-domain nodal displacements with those of the coarse mesh nodes. The material is linearly elastic with Young's modulus $E = 73\,\text{GPa}$ and Poisson's ratio $\nu = 0.3$. The sheet of material is subjected to uniform loading along its right edge.

This particular case was solved using the three approaches described in this paper, i.e., using all global Lagrange multipliers, all local Lagrange multipliers (depicted in Fig. 2), and the mixed approach. In the latter two approaches, penalty terms were added to the potential of the constraint forces to improve the conditioning of the interface flexibility matrix. When using all global Lagrange multipliers, the condition number of the interface flexibility matrix was found to be $\kappa = 9.4 \times 10^5$. If all local Lagrange multipliers are used, this condition number of the flexibility matrix defined by Eq. (23) becomes a function of the penalty coefficient, p. Fig. 3 shows the condition number, $\kappa(F_L) - 1$ as a function of the non-dimensional penalty coefficient, $\overline{p} = p/\|K_{ii}\|_\infty$. As predicted in Section 5.1, this condition number approaches unity as the penalty coefficient increases. For $\overline{p} = 1$, $\kappa(F_I) = 1.7917$, which means that iterative methods will be fast converging even for modest values of the penalty coefficient.

8. Conclusions

A novel approach has been proposed for parallel computation in flexible multibody dynamics. The approach relies on two distinct strategies for the enforcement of the kinematic constraints at the interface between sub-domains. The traditional approach is to use global Lagrange multipliers to enforce all constraints. In the proposed approach, a hybrid strategy is used: some constraint are enforced using local Lagrange multipliers, while the remaining are imposed via global Lagrange multipliers. A coarse mesh is defined as a byproduct of the local Lagrange multiplier technique. Furthermore, an augmented Lagrangian formulation is used in conjunction with the local Lagrange multipliers. If all kinematic constraints are enforced via this technique, the penalty terms stemming from the augmented Lagrangian formulation provide a natural conditioning of the interface problem expressed in terms of the local Lagrange multipliers. In fact, as the penalty factor increases, the condition number of the interface problem flexibility matrix tend to unity. Clearly, this approach is ideally suited for iterative solutions of the interface problem. This advantage, however, comes at the expense of the solution of a large sized coarse mesh problem. When the proposed combination of global and local Lagrange multipliers is used, it is still possible to obtain an interface problem expressed in terms of the sole global Lagrange multipliers and the solution of the coarse mesh problem provides a natural preconditioning of this interface problem.

References

[1] A. Fijany, I. Sharf, G.M.T. D'Eleuterio, Parallel O(log$n$) algorithms for computation of manipulator forward dynamics, IEEE Transactions on Robotics and Automation 11 (3) (1995) 389–400.

[2] K.S. Anderson, An order-n formulation for the motion simulation of general multi-rigid-body tree systems, Computers and Structures 46 (3) (1993) 547–559.

[3] R. Featherstone, A divide-and-conquer articulated body algorithm for parallel O(log(n)) calculation of rigid body dynamics. Part 1: basic algorithm, International Journal of Robotics Research 18 (9) (1999) 867–875.

[4] R. Featherstone, A divide-and-conquer articulated body algorithm for parallel O(log(n)) calculation of rigid body dynamics. Part 2: trees, loops, and accuracy, International Journal of Robotics Research 18 (9) (1999) 876–892.

[5] K.S. Anderson, S. Duan, Highly parallelizable low order algorithm for the dynamics of complex multi rigid body systems, Journal of Guidance, Control and Dynamics 23 (2) (2000) 355–364.

[6] J.H. Critchley, K.S. Anderson, Parallel logarithmic order algorithm for general multibody system dynamics, Multibody Systems Dynamics 12 (1) (2004) 75–93.

[7] M. Géradin, A. Cardona, Flexible Multibody System: A Finite Element Approach, Wiley, New York, 2001.
[8] D. Coulon, Paradigmes Parallèles Appliqués au Cacul de la Dynamique non Linéaire des Mécanismes Flexibles, Ph.D. Thesis, Université de Liège, Belgium, 1998.
[9] I.S. Duff, J.K. Reid, The multifrontal solution of indefinite sparse symmetric linear equations, ACM Transactions on Mathematical Software 9 (3) (1973) 302–325.
[10] J.H. Kim, S.J. Kim, Multifrontal solver combined with graph partitioners, AIAA Journal 37 (8) (1999) 964–970.
[11] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 7 (3) (1986) 856–869.
[12] C. Farhat, F.-X. Roux, A method of finite element tearing and interconnecting and its parallel solution algorithm, International Journal for Numerical Methods in Engineering 32 (1991) 1205–1227.
[13] C. Farhat, F.-X. Roux, An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems, SIAM Journal on Scientific and Statistical Computing 13 (1) (1992) 379–396.
[14] C. Farhat, L. Crivelli, F-X. Roux, A transient FETI methodology for large-scale implicit computations in structural mechanics, International Journal for Numerical Methods in Engineering 37 (11) (1994) 1945–1975.
[15] C. Farhat, L. Crivelli, F.-X. Roux, Extending substructure based iterative solvers to multiple load and repeated analyses, Computer Methods and Applied Mechanics and Engineering 117 (1994) 195–209.
[16] C. Farhat, P.S. Chen, J. Mandel, F.-X. Roux, 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 155 (1998) 129–151.
[17] C. Farhat, P.S. Chen, J. Mandel, F.-X. Roux, The two-level FETI method. Part II: extension to shell problems parallel implementation and performance results, Computer Methods and Applied Mechanics and Engineering 155 (1998) 153–179.
[18] C. Farhat, P.S. Chen, F. Risler, F.-X. Roux, A unified framework for accelerating the convergence of iterative substructuring methods with Lagrange multipliers, International Journal for Numerical Methods in Engineering 42 (1998) 257–288.
[19] C. Farhat, M. Géradin, On the general solution by a direct method of a large-scale singular system of linear equations: application to the analysis of floating structures, International Journal for Numerical Methods in Engineering 41 (4) (1998) 675–696.
[20] C. Farhat, K. Pierson, M. Lesoinne, The second generation FETI methods and their application to the parallel solution of large-scale linear and geometrically non-linear structural analysis problems, Computer Methods and Applied Mechanics and Engineering 184 (2000) 333–374.
[21] C. Farhat, M. Lesoinne, K. Pierson, A scalable dual–primal domain decomposition method, Numerical Linear Algebra with Applications 7 (2000) 687–714.
[22] C. Farhat, M. Lesoinne, P. Le Tallec, K. Pierson, D. Rixen, 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 50 (2001) 1523–1544.
[23] H. Bavestrello, P. Avery, C. Farhat, Incorporation of linear multipoint constraints in domain-decomposition-based iterative solvers—Part II: blending FETI-DP and mortar methods and assembling floating substructures, Computer Methods in Applied Mechanics and Engineering 196 (2007) 1347–1368.
[24] J.C. Chiou, K.C. Park, C. Farhat, A natural partitioning scheme for parallel simulation of multibody systems, International Journal for Numerical Methods in Engineering 36 (1993) 945–967.
[25] D. Coulon, M. Géradin, C. Farhat, Adaptation of a finite element solver for the analysis of flexible mechanisms to parallel processing systems, in: Second International Conference on Computational Structures Technology, Athens, Greece, 30 August–1 September 1994.
[26] G. Quaranta, P. Masarati, P. Mantegazza, Multibody analysis of controlled aeroelastic systems on parallel computers, Multibody System Dynamics 8 (1) (2002) 71–102.
[27] A. Laulusa, O.A. Bauchau, Review of classical approaches for constraint enforcement in multibody systems, Journal of Computational and Nonlinear Dynamics 3 (1) (2008) 011004.
[28] O.A. Bauchau, A. Laulusa, Review of contemporary approaches for constraint enforcement in multibody systems, Journal of Computational and Nonlinear Dynamics 3 (1) (2008) 011005.
[29] P. Tong, T.H.H. Pian, A hybrid-element approach to crack problems in plane elasticity, International Journal for Numerical Methods in Engineering 7 (1973) 297–308.
[30] K.C. Park, C.A. Felippa, A variational framework for solution method developments in structural mechanics, Journal of Applied Mechanics 65 (1998) 242–249.

[31] K.C. Park, C.A. Felippa, U.A. Gumaste, A localized version of the method of Lagrange multipliers and its applications, Computational Mechanics 24 (2000) 476–490.
[32] M. Fortin, R. Glowinski, Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems, North-Holland, Amsterdam, The Netherlands, 1983.
[33] P.E. Gill, W. Murray, M.A. Saunders, M.H. Wright, Sequential quadratic programming methods for nonlinear programming, in: E.J. Haug (Ed.), Computer-Aided Analysis and Optimization of Mechanical System Dynamics, Springer, Berlin, Heidelberg, 1984, pp. 679–697.
[34] E. Bayo, J. García de Jalón, M.A. Serna, A modified Lagrangian formulation for the dynamic analysis of constrained mechanical systems, Computer Methods in Applied Mechanics and Engineering 71 (1988) 183–195.
[35] E. Bayo, J. García de Jalón, A. Avello, J. Cuadrado, An efficient computational method for real time multibody dynamic simulation in fully Cartesian coordinates, Computer Methods in Applied Mechanics and Engineering 92 (1991) 377–395.
[36] C.L. Bottasso, O.A. Bauchau, A. Cardona, Time-step-size-independent conditioning and sensitivity to perturbations in the numerical solution of index three differential algebraic equations, SIAM Journal on Scientific Computing 29 (1) (2007) 397–414.
[37] O.A. Bauchau, A. Epple, C.L. Bottasso, Scaling of constraints and augmented lagrangian formulations in multibody dynamics simulations, Journal of Computational and Nonlinear Dynamics 4 (2) (2009) 021007.