INTEGRATION OF HETEROGENEOUS SIMULATION MODELS FOR NETWORK-DISTRIBUTED SIMULATION

by

Geun Soo Ryu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in The University of Michigan
2009

Doctoral Committee:

Professor Gregory M. Hulbert, Co-chair
Research Scientist Zheng-Dong Ma, Co-Chair
Professor Carlos E. S. Cesnik
Professor Jeffrey L. Stein

To my family

ACKNOWLEDGMENTS

I want to express my sincere appreciation to my advisors, Professor Gregory M. Hulbert and Dr. Zheng-Dong Ma. This research would not have been possible without their patient guidance and generous support. Their experience and vision always helped me come through the difficulties and move forward in the right direction. My sincere thanks go to Professor Jeffrey L. Stein and Professor Carlos E. S. Cesnik for his interest in this research and willingness to serve as my committee members. I also would like to thank Dr. Han, who helped me with his academic and industrial knowledge.

I would like to thank Jong-Hun, Jae-wook, Brooke, Kyoungjun, Sungik and all members of Computational Mechanics Laboratory.

Special thanks to my wife, Soo-Jin Yang. She is always with me as my friend and encouraging me as my family and gave a great gift, Jeffrey Sungmin to me.

This research is supported by the Automotive Research Center Thrust Area V. I greatly appreciate the support from the ARC.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

ABSTRACT

Integration of Heterogeneous Simulation Models for Network-Distributed Simulation

by

Geun Soo Ryu

Co-Chair: Gregory M. Hulbert and Zheng-Dong Ma

Distributed simulation is close to reaching its potential to fulfill the demands of industrial CAE by harnessing nearly unlimited computing power across network environments and by efficiently reusing and integrating already constructed simulation models. A distributed simulation platform, denoted as D-Sim, has been under development in our research group since 2001. The present work focuses on the integration of heterogeneous subsystem models, including multibody dynamics (MBD) and finite element (FEM) subsystem models, and conducting seamlessly integrated simulation for design tasks in a distributed computing environment.

Under the guise of a gluing algorithm, the Partitioned Iteration Method (PIM) was developed, which can be used to integrate distributed deformable bodies while allowing large rigid body motions among the bodies or subsystems. The PIM is based upon a

floating frame of reference, in which the global motion of the flexible body can be expressed with linearized elastic deformations by assumption of infinitesimal strains and reference frame as large overall motion. When embedded in D-Sim, it also enables using independent simulation servers, in which each server can run commercially available or research-based MBD and/or FEM codes to minimize the information exchange across the different platforms yet still obtain results within engineering accuracy. Examples are provided which integrate FEM and MBD models and which demonstrate the performance of the PIM. The examples also highlight how to decouple and integrate rigid body motion and elastic deformation using the enhanced gluing algorithm.

A gluing algorithm plays a critical role in integrating the distributed subsystems and components. It is one of the research objectives to apply the gluing algorithm to general simulation models, which may be assembled by diverse connecting methods, including spot welds, bolts, bushings, and other physical connections. The gluing algorithm concept has been extended by creating flexible gluing joints, which can deal with various connections between subsystems, and can account for linear and non-linear flexibility at these connections. This not only improves the accuracy of the simulation to represent the real physical system, but also can improve the convergence of multibody dynamics simulation.

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

The major concern of past and present industry is to find how to make and supply the products for fulfilling customer needs with minimal cost and time. The answer to the above question becomes motivation in the most human activities in industry. Engineers and researchers are now using Computer Aided Design, Manufacturing and Engineering (CAD/CAM/CAE) in everyday tasks with the dramatic changes in computing power and wider availability of software tools for design and production. International competition, increased emphasis on quality, and efficiency of developing product, are also forcing manufacturers to use CAD/CAM/CAE systems to automate their design and product processes.

Presently, CAE is a useful design and evaluation tool for reducing product development cost with the development of various simulation software and powerful computer resources. Engineers and researchers can build high-performance simulation models of complex systems and can perform simulations with multi-processing technologies to reduce the number of physical prototyping and testing. The demands placed on present CAE tools are the support of variant design and an efficient usability, so that engineers can more easily and fast access to existing design projects in the early stage of product development like product life management (PLM) solutions [1]. These PLM solutions successfully provide services in business operation, information management and exchange during the production development. In the computational mechanics field, however, simulation process has become a bottleneck and has not delivered the efficiency and productivity gains envisaged.

Distributed simulation techniques can have great potential to fulfill the present demand of CAE tools with unlimited computing power in network environment and an efficient reusability of integrating already constructed simulation models. Therefore, the focus of this dissertation is developing a new integration methodology, not only to provide an alternative to the uniprocessor simulation, but also to integrate diverse simulation models and simulation schemes.

1.1.1 Present and Future Trend of Computer Aided Engineering(CAE)

Shortening of product life cycle, increasing cost pressure in the global competition and the heterogeneity of computer-aided development systems are determining the trends of today's manufacturing industry. Computer Aided Engineering (CAE) is a concept that was first proposed by J. Lemon in 1980[2], the founder of SDRC, as a way to provide analytical information in a timely manner in the product development process, and through doing this, products and production processes with great improvement are made possible. The broad concept of CAE includes CAD, FE modeling, Finite Element Analysis(FEA) and Design with significant reliance on graphic display systems [3]. CAE is a useful design and evaluation tool with the development of various simulation software and powerful computer resources.

However, CAE has been developed stand alone and independent with CAD/CAM because of analysis oriented simulation with standard FE codes for high accuracy. CAE only support occasional different lines in the development process. But present CAE is more toward design analysis and evaluation by numerical analysis. With growing the demand for virtual prototype, which can reduce the need for costly physical prototyping and testing in product development, CAE should not be just for computer aided engineering analysis, but have large extent of design analysis, evaluation, optimization and process simulation as the connector of CAD and CAM as shown Figure 1.1 [4].

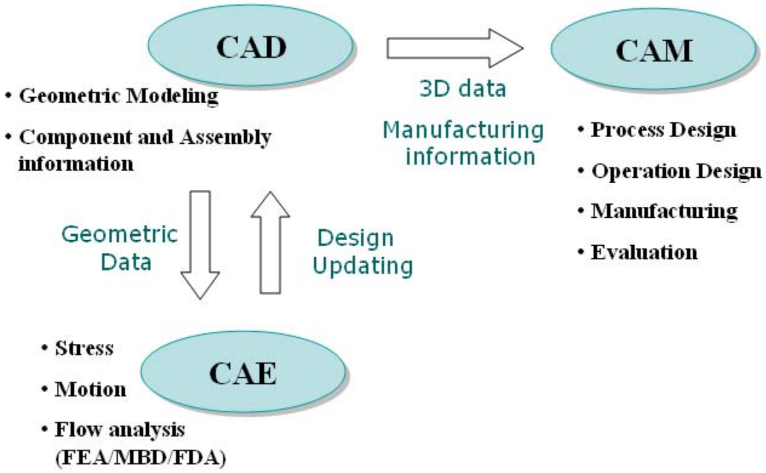


Figure 1.1 Interactions of CAD/CAM and CAE in product manufacturing process

The role of CAE has changed from technical computation which was more or less a service for the different development departments into a virtual product development process. The present demand of CAE should be embedded in CAD for design oriented simulation [5]. For the cost and time reduction in product development processes, CAD has shown quite the success to make change of engineering, but CAE is still regarded to be expensive because the process of modeling is high time consuming work, analysis results are difficult to be reflected to design change [6]. For the CAD embedded CAE, FE modeling methods such as automatic mesh generation should be simplified and improved. The model generation process time on a systems or subsystems level may be as high as 80% of the total engineering simulation effort (person power), with such tasks as linking with CAD data, de-featuring CAD data for adequate mesh quality, input load/support condition [7]. This yields a painful and time consuming work for FEA and becomes barrier for CAE to be full integrated with CAD. Automatic model generation has the potential to significantly increase simulation productivity and allow sparse engineering resources to concentrate on their vital tasks.

To have the most impact on the product design process, future CAE should have the following trends

Standard CAE program

Integration simulation tools for multi-disciplinary simulation

Parallel/Distributed Computing with Intra/Internet (Grid Computing)

Web based modeling and model management

1.1.2 Standard CAE Program for Being Imbedded in CAD

Shortening of product life cycle, increasing cost pressure in the global competition and the heterogeneity of computer-aided development systems are determining the trends of today's automotive development. In the automotive industry, the average number of CAE-tools is more than 40. Furthermore there are at least 14 different CAE-simulation models in use by an automotive OEM [7]. This fact is a big bottleneck for CAE to be imbedded in CAD, since only simulation specialists can solve the complex system and can access to their models.

For CAD integrated CAE in the early design stage, standardization of processes will be one of the most important factors and responsibilities to support the integration of CAE-simulation in the future. Most users become designers rather than analysts, consequently simulation tools should be user oriented / user friendly.

1.1.3 Integration Simulation Tools for Multi-disciplinary Simulation

For complex systems or subsystems in industry, design is a very complex optimization task often involving multi-disciplines, multi-objectives, and computationally intensive processes for product simulation. As manufacturers in industry as well as military customers try to incorporate multidisciplinary design methods in the conceptual design phase, a systematic approach needs to be introduced. Modeling and computer simulation have become tools in all engineering disciplines. Krueger et. al [8] presented two modeling philosophies for multidisciplinary simulation. One approach is that all model components are implemented in a single modeling or simulation tool, using common libraries or a common modeling language, and creating a single model comprising elements of all involved disciplines. Second, the coupling of specialized tools by the means of interfaces is performed. This is especially suited for systems where sub-models already exist in specialized tools and where those models are too large and complex to be transferred into a single simulation tool. Presently many researches in [9-11] deal only with the second approach, i.e. with the coupling of tools via interfaces. While the simulation tools have become very sophisticated in their own domains, the simulation of complex systems calls for multidisciplinary simulation. This can be achieved by the coupling of the existing codes. For this purpose, interfaces between the codes have to be developed. These interfaces have to take into consideration the nature of the description of the physical model, numerical properties of the respective simulation methods, and software and hardware implementation issues. The analysis of complex product is generally concerned with multi-disciplinary modules exchanging physical parameters.

1.1.4 Parallel/ Distributed Computing with Intranet/Internet (Grid Computing)

Despite continuous advances in computing power, increasing complexity of analysis tools and simulation model size seem to keep pace with computing performance enhancements. With high performance computing (HPC), the power of simulation software tools can be greatly increased in their application, accuracy and computational cost. However, this development of computer resources can be a barrier for CAE to be embedded in CAD in the early product design stage. Grid computing (or the use of a computational grid) technology present in [12] is applying the resources of many computers in a network to a single problem at the same time - usually to a scientific or technical problem that requires a great number of computer processing cycles or access to large amounts of data. Grid computing can be thought of as distributed and large-scale computing and as a form of network-distributed parallel processing. It can be confined to the network of computer workstations within a corporation or it can be a public collaboration (in which case it is also sometimes known as a form of peer-to-peer computing).

1.1.5 Web Based Modeling and Model Data Management

One of the most effective approaches to force the integration of simulation into the design process will be a powerful simulation data management tool. CAD/CAM/CAE system vendors are all moving forward integrating their systems with the Internet. Thus Internet capabilities from inside geometric modeling systems are either already in place or about to become available. By using the Internet as an extension to a geometric modeling system through the use of browsers and browser plug-ins, an entire project team and its clients can view and manipulate models and drawings in various web formats early in the design process. Similar simulation data are stored at different organizational units connected by network and able to be easily accessed by users. With the proportion of virtual development methods, the communication between used systems for the virtual vehicle development will become increasingly more complicated.

1.2 DISTRIBUTED SIMULATION PLATFORM (D-SIM)

In a virtual prototyping environment to support product development, the component models usually are functionally and geographically distributed due to the adoption of multi-layer supply chains as well as the increased collaborations among different engineering and business units. It is difficult to bring all the component models together and to analyze such a monolithic model using a uniprocessor simulation; e.g., simulation of an automotive vehicle that includes all subsystem models. Thus, it is important to have a capability to simulate the system-based distributed models. A Distributed Simulation Platform (D-Sim)[13-15] was developed to support the design tasks in multi-layered, distributed supply chains in modern manufacturing systems. In a distributed simulation platform, an engineer just needs to choose right subsystem and send the information of interface and model information to the coordinator of distributed simulation platform and then the coordinator executes the simulations simultaneously and checks whether the results are converged.

There are two different perspectives on the decomposition and coupling of complex systems, as laid out by Tseng[16], which are, “divide-and-conquer” and “integrate-and-collaborate,” or, in other words, decomposition and gluing. The former focuses on how to actively partition a large problem in order to take advantage of parallel computing. In contrast, the gluing perspective starts from the fact that many systems are already partitioned and distributed and does not involve active decomposition. Rather than attempting to partition an existing model, the goal is to glue together an already partitioned model that may contain, for example, many different finite element and multibody dynamics models. Figure 1.2 depicts an example of assembling distributed subsystem models into an Army ground vehicle. Each component was already constructed and simulated for its own function by supplier.

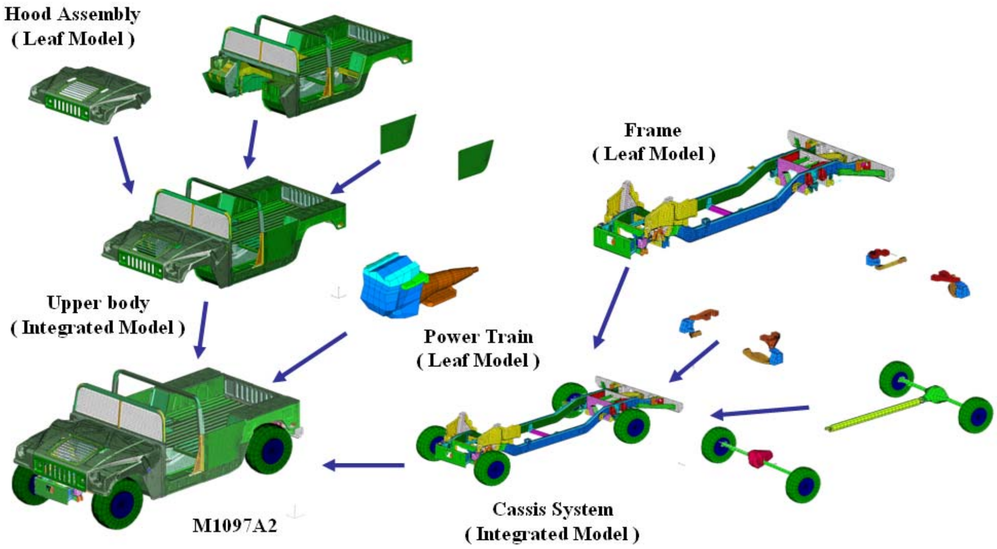


Figure 1.2 Example of the distributed model of Army Ground Vehicle

Distributed simulation, as a research area in computer science primarily focuses on issues of communication, synchronization and time warping of the different components or entities in the simulation. Wang [17] address the need for, and develop, a simulation platform that can incorporate distributed mechanical systems models and couple them together to perform dynamics simulations within the framework of a general virtual prototyping process. The distributed subsystems models might be developed independently, use different software packages, run on different computers, and/or reside at different geographical locations. And also in order to protect proprietary information, the simulation platform may not be willing to share their models directly but to share only minimal information during the coupled simulation.

In scientific computing and military area, many efforts have been made to develop distributed computing technologies and its application. Distributed computing application started to flourish due to the previously limited availability of computers and suitable hardware platforms. NS-2[18], INSANE [19] and NetSim [20] provide rich APIs tools and also provides an extensible simulation environment and a rich graphical user-interface written in Java which can run regardless of computer architecture. These codes represent successful efforts in distributed simulation and offer valuable guidance.

HLA (High Level Architecture), defined in IEEE 1516 [21], is a general-purpose architecture for simulation interoperability and reuse, which was originally developed by US Department of Defense[22, 23]. Gan, B.P. [24] and Rajjev, S. [25] use HLA simulation for distributed supply chain simulation in manufacturing and transportation area. HLA also supports language-independent and platform-independent composition of a simulation. HLA is a generic framework and is a good candidate for implement the development methodology.

Many distributed simulation efforts have been built on top of CORBA (Common Object Request Broker Architecture) [26] in commercialized distributed technologies. CORBA is produced by the OMG to allow applications to communicate with one another independent of location, platform, or vendor. DCOM [27] is the competing standard delivered by Microsoft. RMI [28] is developed by SUN Microsystems. It enables garbage collection of distributed objects by extending the built-in garbage collection functionality of Java Virtual Machine to network. This functionality is helpful for objects life management in a distributed environment. However, RMI is limited only to the Java language. These technologies all achieved limited success before they were adapted for the Web [29]. While the interoperability among these technologies is hard to come by, SOAP-based web services [29, 30] are drawing more and more attentions. SOAP is a protocol specification that defines a uniform way of passing the Extensible Markup Language (XML) encoded data. XML provides a meta-language to express complex interactions between clients and services or between components of a composite service. Due to the platform independence of XML and HTTP, web services provide an ideal platform to interoperate the legacy applications and serve as a new promising distributed technology which has gained support from competitors. Dieckman, D. et al. [31] proposed the idea of protect intellectual property by wrapping the model with some simulation interface. However, the work presented did not provide any general couple simulation algorithm, which is necessary to solve practical engineering problems. Distributed simulation platforms or frameworks as represented in [32, 33] expose some standard APIs for the users to put their simulations in the platforms.

The widely spread availability of parallel computers and their potential for the numerical solution of difficult to solve partial differential equations have led to a large amount of research in sub-structuring [34] or domain decomposition methods [35, 36]. Adeli, H. [37] presents a primitive version of a distributed finite element simulation, in which stiffness matrices and load vectors are generated concurrently on clients and sent to a central server to be assembled. Other researchers adopted the decomposition of large mechanical systems with a primary focus on the decomposition strategy. Farhat, C. , Wilson E.[38] and Farhat, C., F.X. Roux [39] have studied the parallelization of both direct solution methods and iterative methods. A sub-structuring method called FETI (Finite Element Tearing and Interconnecting) [40] introduces extra traction variables and exhibiting more flexibility for model reduction and coupling, compared to competing schemes. In the multibody dynamics arena, researchers also have studied how to partition and parallelize systems[14, 41-50]. Kim S. [41] proposed a subsystem synthesis method for dynamic analysis of vehicle multibody systems, in which each subsystem is independently analyzed with a virtual reference body and the overall vehicle system analysis is formed by synthesizing the effective inertia matrix and force vector from the virtual reference body of each subsystem. Featherstone, R. [42, 43] proposed "a divide-and-conquer" algorithm for rigid body dynamics, which reduces the system to an "articulated-body" by recursively applying a formula Another approach was given by Anderson, K. et. al. [44-46], in which the equations of the subsystem models are evaluated in parallel, and the results are loaded into a single system wide equation to explicitly calculate the constraint forces.

The most general distributed simulation reported in the literature is multi-disciplinary integration [51-54]. Usually, each discipline is wrapped as an object and the simulation only involves single-direction information flow between different domains. This strategy is widely used in multidisciplinary optimization application. There are several simulation multi-disciplinary optimization tools in the market that have some limited distributed simulation capability. iSIGHT / FIPER [55] provide platforms that can automate most optimization problems and can extensively save time on pre- and post-processing; thus, it is gaining popularity within the industrial design community. DOME (Distributed Object-based Modeling Environment) is a prototype implementation of the "simulation service marketplace" concept proposed in [56], which enables distributed collaborative design through parallel heterogeneous simulations. The above tools mainly focus on the simple parallel execution of simulation tasks of a design problem rather than the difficulty of coupled simulation in a distributed simulation system. ModelCenter of Phoenix Integration [57] provides a Simulink-like environment to assemble component models based on the data flow among them. It requires the simulation models to be modeled as control blocks and has several limitations for application to the real distributed simulation problems in the automotive industry. The critical problem is to couple already decomposed and distributed mechanical system models rather than the control box-based simulation.

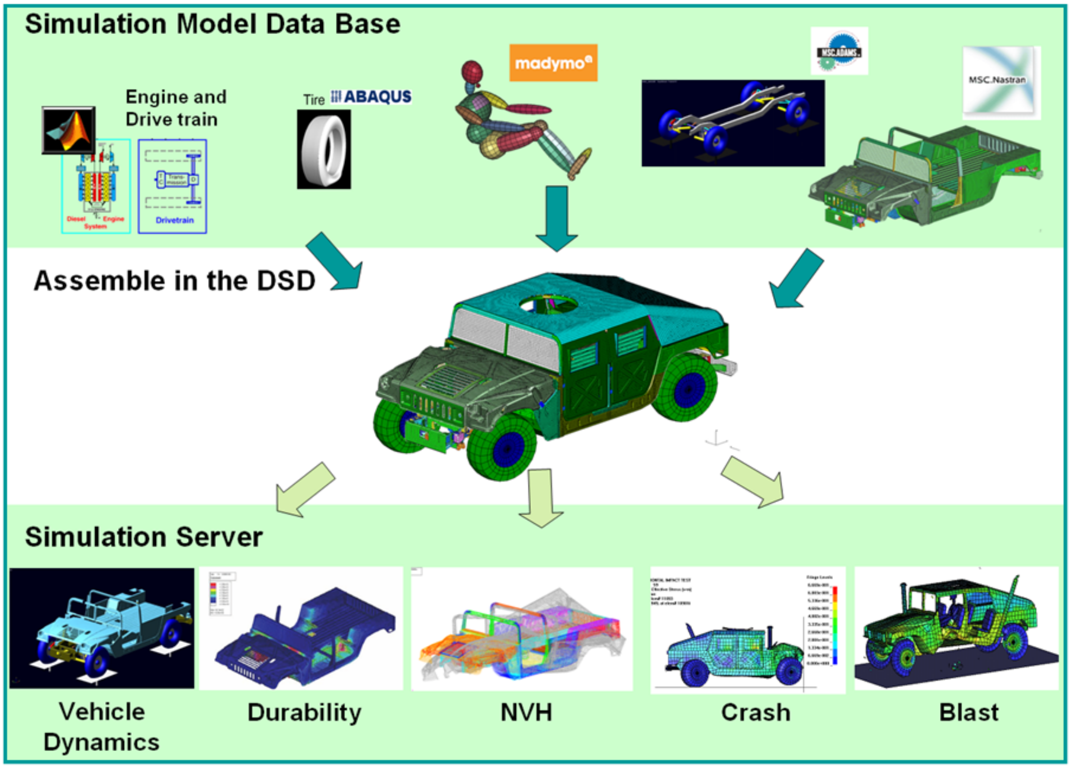


Figure 1.3 Vision of the distributed simulation for army ground vehicle

The vision of D-Sim for producing a digital vehicle based on distributed subsystems models is presented in the Figure 1.3. In this concept, we envision a library of component models, which are built and executed at different locations using different software systems in a multi-layered simulation platform. For example, a detailed vehicle model that includes tires, leaf springs, suspension and steering components and joints, chassis frame, etc., can be developed. Large deformation nonlinear finite element algorithms can be used to model the tires and leaf springs as well as the chassis in the case of crash simulation, while multibody system algorithms are used to model the suspension and steering components and mechanical joints as well as other rigid bodies or bodies that experience small deformations. Each component can be solved in D-Sim using distributed independent solver that wrap commercial software in D-Sim. A detailed tire model may be run in ABAQUS, a flexible chassis frame model may be run in MSC/NASTRAN, an engine combustion model may be run in MATLAB/SIMULINK, and a multibody dynamics model of the rest of the components may be run in ADAMS. D-Sim provides valuable tools to support distributed collaborative design. Engineers in different locations can create different component models for the vehicle, and models of the different vehicle designs can be easily assembled and simulated in the simulation platform.

A commercialized version of the distributed simulation system can be of great benefit to the industry. It could be embedded in the existing PLM systems to integrate simulation capacity in the system. When building such a simulation system, attention should be paid to the enrichment of the Extensible Markup Language (XML) model description, inclusion of other potential gluing algorithms, the storage and processing technique of model data, and introduction of multi-disciplinary simulation capability.

1.3 INTEGRATION OF MULTIBODY DYNAMIC SIMULATION AND LINEAR ELASTIC FINITE ELEMENT ANALYSIS IN D-SIM

The distributed simulation platform (D-Sim) was developed to cater to the distributed nature of collaborative design inside the supply chain of modern product development, and to significantly accelerate the product development process. It provides great potential in laying out innovative and new products in a cost-effective and timely manner. However, one of the remaining challenges in developing such distributed simulation systems, is to integrate heterogeneous subsystem models, such as multibody dynamics subsystems models and finite element subsystems models for practical engineering problems, e.g., detailed durability or NVH simulation, and to conduct seamlessly integrated simulation and design tasks in a distributed computing environment. Ideally, a practical distributed simulation environment should allow using existing commercial packages, including a combination of multibody dynamics codes, such as MSC/ADAMS, and finite element codes, such as MSC/NASTRAN and ABAQUS, without modifying their solvers and user interfaces. Rigid body motion of the overall system (or a subsystem) should be solved with the numerical integrators in a multibody dynamics code, while deformation of each individual component should be solved with the solvers in an existing finite element code.

Integration of multibody dynamic analysis and Finite Element analysis is one of challenges of multi-disciplinary simulation in the Distributed Simulation. For instance, as shown in the Figure 1.5, the multibody dynamics model of frame and tire assembly on the experimental equipments and Finite element body model are assembled with a gluing algorithm for durability and NVH analysis.

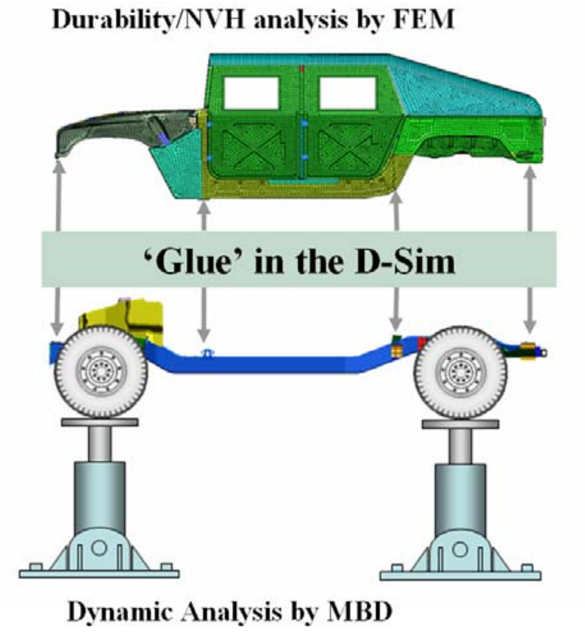


Figure 1.4 Coupling MBD and FE analysis in D-Sim

The multibody dynamics model is built using a commercial preprocessor and solved in a nonlinear dynamics solver, and the flexible body model is constructed for durability analysis and/or NVH analysis and solved in a linear FE solver. It is impossible or very difficult to simulate stand-alone model with nonlinear dynamics solver or nonlinear FE solver unless engineer/researcher would make a new assembled simulation model. Thus, it becomes necessary to successfully integrate large/small deformation finite element formulations and multibody systems to be able to develop a new generation of computer algorithms and codes that can be used to solve the new challenging problems being encountered. In particular, it is important to develop methods to integrate these two models in distributed simulation without building a new simulation model.

The effect of structural flexibility has become an important issue for more accurate kinematics and dynamic analysis in industrial practice. Many investigations for a flexible multi-body system which consists of elastic and rigid components have focused on how the flexible components can be formulated and combined with other component equations. Particular effort has focused on the development of simulation method for coupling the large displacement problem in FEM in which it is assumed that the deformation is small and rotations may be large.

Formulations of the dynamics of mechanical systems that include large deformations (geometrically nonlinear FE) usually lead to a constant mass matrix and nonlinear stiffness matrix, Zienkiewicz and Taylor[58]. Standard FE codes for large deformation dynamic analysis primarily are based on this formulation. Typically, the nonlinear stiffness matrix has to be updated frequently (when using a standard Newton iteration method); consequently, the computational cost can be high.

During the past several decades, flexible multi-body dynamics has emerged as an important research field emanating from the need to simulate many industrial and technological systems. Flexible multi-body dynamics is the subject concerned with the computer modeling and analysis of constrained deformable bodies that undergo large displacements, including large rotations. The large displacements can include rigid body motion as well as elastic deformations. This research has lead to numerous different formulation approaches. Shabana [59-61] introduced the floating frame concept, and coupled this with linearized elasticity theory. Earlier, Huston [62, 63] suggested the Finite Segment Method in which the deformable body is assumed to consist of a set of rigid bodies which are connected by springs and/or dampers. The system elasticity, represented by spring coefficients can be determined using the finite element method. Belytschko

[64] employed a convected coordinate technique and a direct nodal force computational scheme for the transient analysis of large-displacement, small-strain problems with material non-linearity. Simo and Vu-Quoc [65] derived an approach based on the large rotation vector instead of formulating conventional beam elements with respect to a moving frame. The limitations, that conventional finite element formulations employ infinitesimal rotations as nodal coordinates and can not be used to describe correctly a finite rigid body rotation, can be overcome with co-rotational nodal coordinates in [64, 65]. More recently, Shabana [66, 67] proposed using absolute displacements and global slopes as element coordinates for flexible bodies, known as the Absolute Nodal Coordinate formulation. Simeon [68] used stabilization techniques in order to solve a solid 2D flexible multibody system with contact. For the 2D and 3D flexible body, Gerstmyr and Schőberl [69] derived an absolute coordinate formulation principally based on the geometrical nonlinear FE having a constant mass matrix and a nonlinear stiffness matrix in a flexible body formulation. The new feature of their approach is that the stiffness matrix is decomposed into the rotation term resulting from spatial integration over whole body and a constant stiffness matrix derived from the assumption of small deformation of the flexible body. Pedersen [70] described a flexible body using only the position of the nodes in the inertial frame; the mass matrix can be formulated as constant matrix with this description instead of nonlinear mass matrix. Other methods for flexible body modeling –Incremental finite element approach, Augmented methods, Recursive and projection methods- are reviewed in Shabana [59]

It becomes necessary to successfully integrate large/small deformation finite element formulations and multibody system to be able to develop a new generation of computer algorithms and codes that can be used to solve the new challenging problems being encountered. Most existing general-purpose multi-body system computer codes are designed to solve systematically and efficiently rigid body systems including small deformation problems. Shabana [71] reviews current integration methods for FEM and MBD with three approaches. These are mainly the gluing algorithms as co-simulation method, the finite element based direct integration method, and the multi-body system based direct integration method. Industry and commercial needs led to a further development of the well known component mode synthesis based on Craig-Bampton modes [34]. Carlbom [72] combined multibody models and finite element models of a rail vehicle which is reduced by eigenmode representation using the modal participation factor or modal contribution factor to determine the most important modes to keep. The introduction of ADAMS/Flex [73] among the commercial simulation codes has overcome many of the problems associated with analysis of flexible multibody dynamics by employing flexible elements whose component modes-based data are based on finite element codes. It is possible to combine MBD model with discrete finite element models of flexible body without modally-based reduced order models. The resultant equations can be difficult to solve with general DAEs solvers. Oghbaei and Anderson [74] proposed a time finite element implicit scheme for stable solution of a large set of equations and constraints. The philosophy adopted in this dissertation is to utilize a distributed simulation environment which should permit using existing commercial or research based software without modifying their solvers and simulation models. In this research, a new integration methodology of distributed general purpose MBD and FEM simulation models with a gluing algorithm [17, 50] is proposed, so that independent commercial or research based solvers for integration of each simulation can be employed.

The small deformation assumption of a flexible body for structural dynamic analysis is important to avoid high computational cost. The separation of reference motion and deformed body motion was proposed by Winfrey [75], Popp [76], Offner [77]. In this approach, the coupling force terms of flexible body motion and rigid body motion equations are neglected due to the assumption of relatively small coupling terms. This assumption is efficient for eliminate coupling terms in the formulation; however, accurate simulation results cannot be guaranteed in high speed or highly flexible problems. Park et. al. [78] separated rigid and flexible body motion using d'Alembert-Lagrange principal equations for partitioning a coupled mechanical system. The principal solution of the rigid body motion is infinitesimal and not assumed to include large motion. Therefore, the rigid body principal solution may be different if the flexible system has large rotational movement. A Partitioned Iteration Method (PIM) is proposed in [79], which decouples the rigid body motion from elastic deformation of the simulated system using an iteration scheme. The PIM employs mean axis reference condition [80], and also a center of gravity (CG) following reference frame for each deformable body in the distributed simulation of flexible multibody systems. The PIM can be considered within the distributed simulation system [15, 17], and accounts for the full coupling of rigid body motion and elastic deformation, The PIM allows, in general, linear, nonlinear, and plastic deformations in each deformable body. Due to mass redistribution, the CG position of a deformed body changes with time. Employing a floating frame that follows the CG position of each deformable body allows decoupling of the rigid body motion from the overall deformation.

1.4 ENHANCED GLUING ALGORITHM

A gluing algorithm should only rely on the information at the interfaces of the models that are to be coupled. An interface can be represented by a set of interface nodes in a FEM or by a set of connecting joints in a multibody dynamics model. The typical information available at the interface can be classified as kinematic information and force information. The kinematic information may contain displacements, velocities, and/or accelerations of the interface. Force information refers to action-reaction forces at the interface. Mechanics principles require that at any interface the force quantities, namely, action-reaction forces, satisfy the equilibrium equations and the kinematic quantities satisfy the compatibility conditions, where it is assumed that the equilibrium and compatibility conditions in the internal domain of each subsystem are satisfied *a priori*. The gluing algorithm employs an iterative process, starting with an initial guess of some of the interface quantities. These interface quantities are then updated using a prescribed iteration process to satisfy the equilibrium and/or compatibility conditions at the interface. In general, if a proper set of interface force variables is defined such that the equilibrium conditions are satisfied, then only the compatibility conditions need to be considered during the iteration process. In this case, the interface force variables can be considered as functions of the interface kinematic quantities, and these interface force variables can be updated using the kinematic information and compatibility conditions. Similarly, if a proper set of the interface kinematic variables is defined such that the compatibility conditions are satisfied, then only the equilibrium conditions need to be considered during the iteration process. In this latter case, the interface kinematic variables are functions of the force quantities at the interface, and they can be updated by satisfying the equilibrium conditions.

The gluing algorithm plays a critical role in coupling the distributed subsystems and components in the distributed simulation. However, it is a challenge to apply the gluing algorithm to general models, which may be assembled by diverse connecting methods, including spot welding, bolting, bushing, and other physical connections. In this research, an improved gluing algorithm is proposed for assembling the subsystems models with general rigid or flexible connections. Figure 1.4 depicts the vehicle under body parts connected with flexible components like rubber bushes. Flexible gluing joints can deal with various connections between subsystems, and can account for linear and non-linear flexibility at these connections. This not only improves the accuracy of the simulation to represent the real physical system, but also can improve the convergence of multibody dynamics simulation.

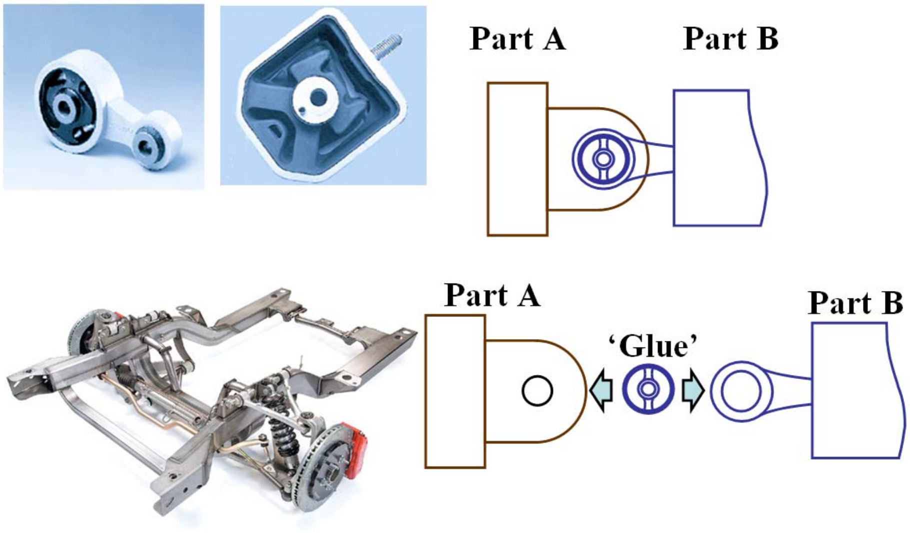


Figure 1.5 Flexible joints at the interface – rubber mounts

The objective of a Gluing Algorithm is to find the interface forces that satisfy compatibility conditions or the interface kinetic variables that satisfies equilibrium condition. In the case of coupling MBD and FEM, it is difficult to satisfy kinematic variables simultaneously at the interface, since each simulation results at the interface is computed with different simulation schemes: multibody dynamic system to solve the nonlinear differential and algebraic equations (DAEs) and classical finite element formulations, connected with modeling the inertial and stiffness of structural systems, employ infinitesimal rotations and displacements as nodal coordinates. In the previous Gluing Algorithm, only one kind of variable such as acceleration can be used for the checking compatibility condition because other variables are calculated with the same time integrator. In the case of gluing different simulation subsystem such as MBD and FEM models, satisfying compatibility conditions in other kinematic variables can not be guaranteed with one variable, and the instability problem with interface error accumulation can make a distributed simulation diverge. Therefore, it can be another issue to finding method how to satisfying compatibility conditions in other kinematic variables such as penalized method in gluing algorithm.

A gluing algorithm was developed and applied to both finite element and multibody dynamics models. The developed algorithm relies only on the interface information of the component models and helps protect the proprietary information of the models [17]. Treating the subsystem models as control blocks and taking advantage of many sophisticated control-based simulation software packages is another common modeling approach. In [47], a modular formulation for multibody systems is proposed, based on the block representation of a multibody system with corresponding input and output quantities. This “block diagram” representation of the system can then be embedded into appropriate simulation packages, e.g., SIMULINK. In [48], “Co-simulation,” is presented, which employs a new discrete time sliding mode controller (DTSM) to satisfy the algebraic constraints among the subsystem models and to solve the causal conflicts associated with the algebraic constraints. The methods reviewed above either involve the active decomposition of the full system and require more information than just that associated with the subsystem interfaces or mandate specific requirements or structure on the formulation of the subsystems. In the context of coupling already distributed subsystems, the gluing perspective is preferred. A study is presented in [16], in which the terminology “gluing algorithm” is first suggested to describe a class of algorithms that can be used to glue distributed component models for use in dynamics simulations. Several gluing algorithms are studied in [16, 49, 50], including MEPI (Maggi’s Equations with Perturbed Iteration) and MOP (Manifold Orthogonal Projection Method). The word “glue” will be used hereafter instead of “couple” in places where the gluing perspective is implied. In [14], a case study of distributed design and simulation of an Army tank road arm is presented. The case study established a distributed framework based on CORBA and involved several simulation and analysis tools.

1.5 RESEARCH OBJECTIVES

The primary objective of this research is to develop ***the integration methodology of heterogeneous simulation models***, which are models for a general multibody dynamic simulation, linear/nonlinear FE simulation for a Distributed Simulation Platform, so that the distributed simulation toolkit for design can be improved in order to solve more general problem in engineering area.

Preliminary steps in achieving these goals are building a model data base of subsystems of Army ground vehicle (High Mobility and Multipurpose Wheeled Vehicle - HMMWV) and a demonstration of distributed simulation.

One of the main difficulties encountered in integration process of MBD and FEM in D-Sim is attributed to the fact that the solution procedures used in finite element codes differ significantly from those used in general-purpose multibody system codes. Finite element methods employ the corotational formulations that are often used with incremental solution procedures. Multibody computer codes, on the other hand, do not, in general, use incremental solution procedures. Especially, the deformed body solution of FE simulation model should have an accurate rotational solution to integrate with MBD solution with a gluing algorithm in a distributed simulation environment which should permit using existing commercial or research based software without modifying their solvers and simulation models.

A Gluing algorithm plays role in integrating the distributed subsystems and components. It is one of challenges to apply the gluing algorithm to general simulation models, which may be assembled by diverse connecting methods, including spot welding, bolting, bushing, and other physical connections having compliance between subsystem parts. Flexibility of interfaces between subsystems can account for linear and non-linear characteristics of the connections, and can be important issue to improve the accuracy of the simulation to represent the real physical system and the convergence of distributed simulation.

1.6 CONTRIBUTIONS

This dissertation presents an integration methodology of heterogeneous simulation models, which are models for a general multibody dynamic simulation, linear/nonlinear FE simulation in a distributed simulation platform that is essential to support the product development in the multi-layered supply chains and the modularized design in the automotive industry. The main contributions of this research include:

1. Model data base of army ground vehicle is built in a distributed simulation platform
   A. the component models are identified with XML model descriptions in the Web Based User Interface
   B. Component subsystem models are built with layered structure in Model Server
   C. Distributed Simulations are demonstrated in D-Sim platform
2. Partitioned Iteration Method is proposed for coupling FE and MBD in D-Sim
   A. Formulation of general equation of motion of FE flexible body including mean axis reference condition
   B. The separation of flexible body formulation in the floating frame of reference into rigid motion and flexible body motion
   C. Flexible body is solved with simulation model already built for the linear elastic simulation
   D. The quadratic velocity correctors in rigid and flexible motion are updated with iteration scheme
3. The integration of MBD and FE simulation model within D-Sim
   A. Demonstration using three dimensional elements- beam, shell

B. Demonstration of coupling MBD and FE with vehicle simulation model

4. Linear/Nonlinear Flexible Gluing Connector are added to previous Gluing Algorithm

   A. Formulation of flexible gluing matrix with linear stiffness and/or damping properties

   B. Validation of Flexible gluing connector with simple plate models and upper body and main frame of HMMWV

5. Penalized Method For Using Baumgarte Stabilization

1.7 DISSERTATION ORGANIZATION

Following this introductory chapter, Chapter 2 briefly introduces the Distributed Simulation Platform and presents the application of the distributed methodology to army ground vehicle, layered simulation model data base, and case study with Gluing algorithm in distributed architecture. Chapter 3 proposes Partitioned Iteration Method for deformed body having large displacement and rotational problem. Chapter 4 presents several application examples that demonstrate the developed methodologies and Chapter 5 follows to provide Enhanced Gluing Algorithm including Flexible Gluing Connector. Chapter 6 finally concludes the dissertation.

CHAPTER 2

DISTRIBUTED SIMULATION PLATFORM AND CASE STUDIES FOR ITS APPLICATION

The overall objective of this research is to develop a methodology for distributed simulation of mechanical system models. This chapter provides an overview for distributed simulation platform and its three key ingredients with case studies for distributed simulation application. The Distributed Simulation Platform(D-Sim) is introduced using army ground vehicles - HMWWV family model. Simulation results using these assembled models provide a concrete introduction of distributed simulation.

Figure 2.1 shows a HMMWV family of vehicles, which shares a common chassis subsystem. By adding different modules onto the common chassis, varieties of vehicles are obtained for different military usages ranging from troop carrier to battle field ambulance. Figure 2.2 depicts component reuse becoming more prevalent in the military and commercial sector. There is a great driving force for the industry to reuse existing parts in the design a new vehicle system. The design and testing of a new part created from scratch is much more expensive in terms of time and money than reusing or modifying a design from a validated existing part. In the platform-based design, the design is classified based on vehicle size and configuration. A simulation capacity that can easily couple component modules to assess the behavior of the whole system will be essential to the success of modern product development processes.

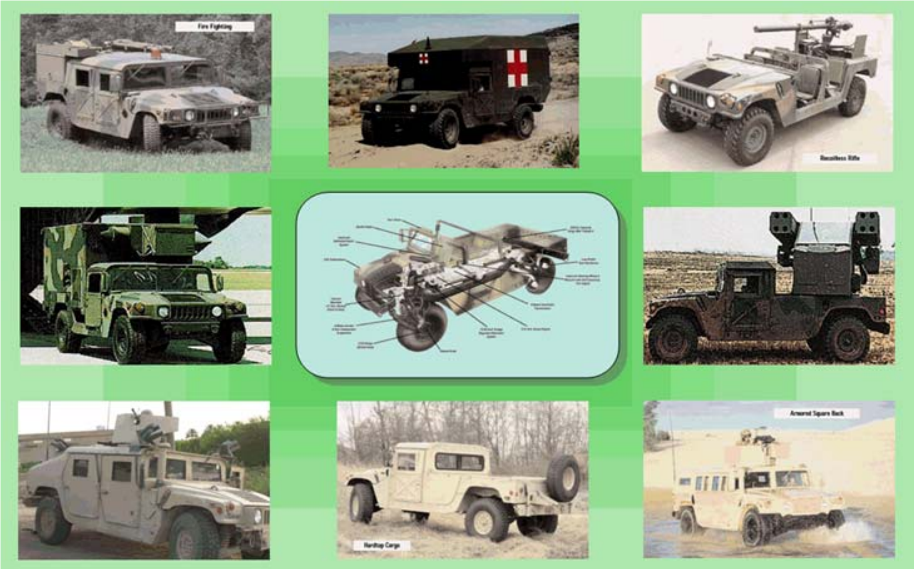


Figure 2.1 HMMWV family

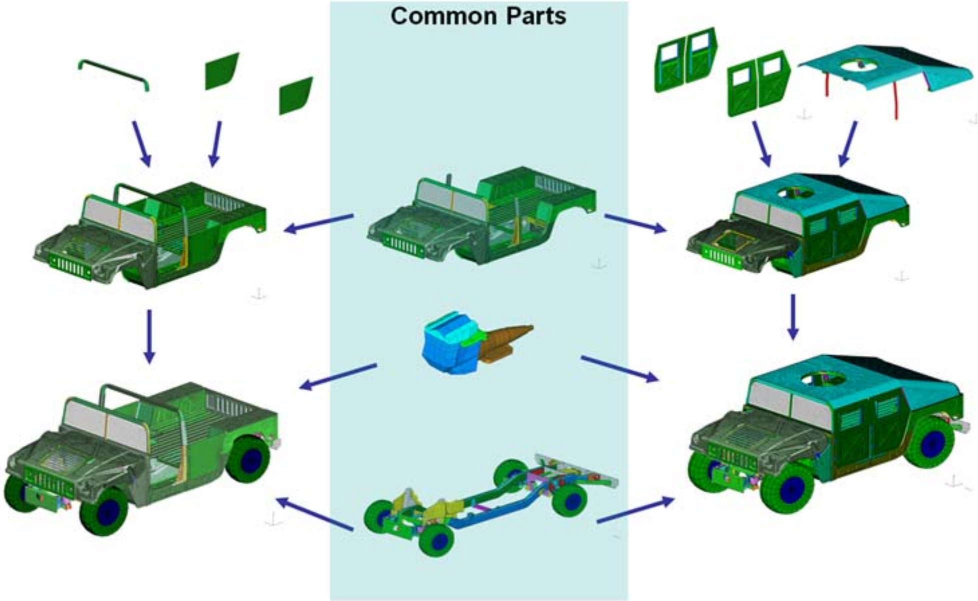


Figure 2.2 Component reuses in HMMWV distributed simulation

2.1 DISTRIBUTED SIMULATION FRAMEWORK

A distributed simulation platform, called D-Sim, was proposed in previous research, and comprises three essential attributes: a general XML description for models suitable for both leaf and integrated models, a gluing algorithm with the T-T method, which only relies on the interface information to integrate subsystem models, and a logical distributed simulation architecture that can be realized using distributed technology. The D-Sim paradigm fits the distributed nature of collaborative design inside the supply chain of modern product development, and can be used to significantly accelerate the product development process. It also provides great potential in laying out innovative and new products in a cost-effective and timely manner. Figure 2.3 depicts the concept of distributed simulation of assembled vehicle upper-body and under-body models. One way to run the simulation of a full vehicle is to bring all the component models together and form a monolithic, stand-alone simulation model and analyze it on a single simulation server, which is a very difficult task. Using a distributed simulation platform, an engineer just needs to choose right subsystem and send the information of interface and model information to the distributed simulation platform simulation corrdinator. Then the coordinator executes the simulations simultaneously and checks whether the results are converged. Subsequently, the engineer can check simulation results in a web based user interface after (or during) the simulation solution process.

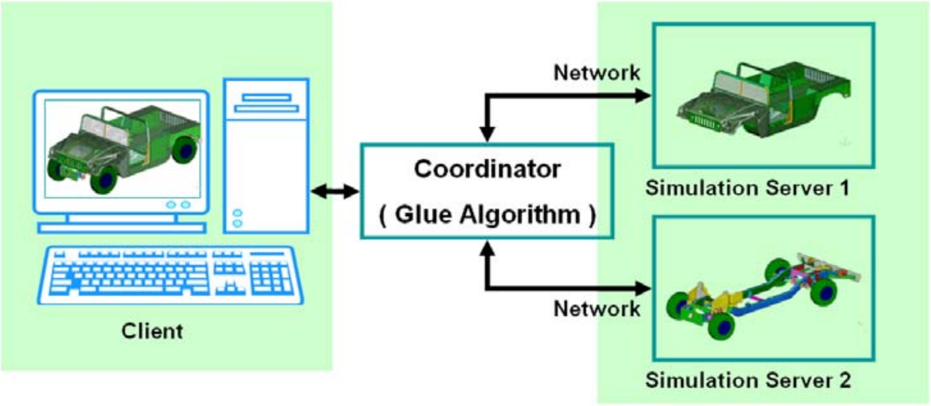


Figure 2.3 Concept of distributed simulation platform

The distributed architecture design provides a solution for realizing the gluing algorithm over the distributed simulation platform and for optimally utilizing distributed computing resources. Figure 2.4 is an illustration of the logical architecture of a distributed simulation platform. Each ellipse represents a model simulation server, which is either a wrapped simulation code that can be accessed through the network or an implementation of the gluing algorithm. Basically, a user accesses the system through a web browser and sends the XML description of a model to the simulation management server to conduct a simulation based upon the model description. The simulation management server first creates an integrated model object based on the XML file. Then the integrated model object parses the XML file and connects the proper subsystem model objects according to the description.

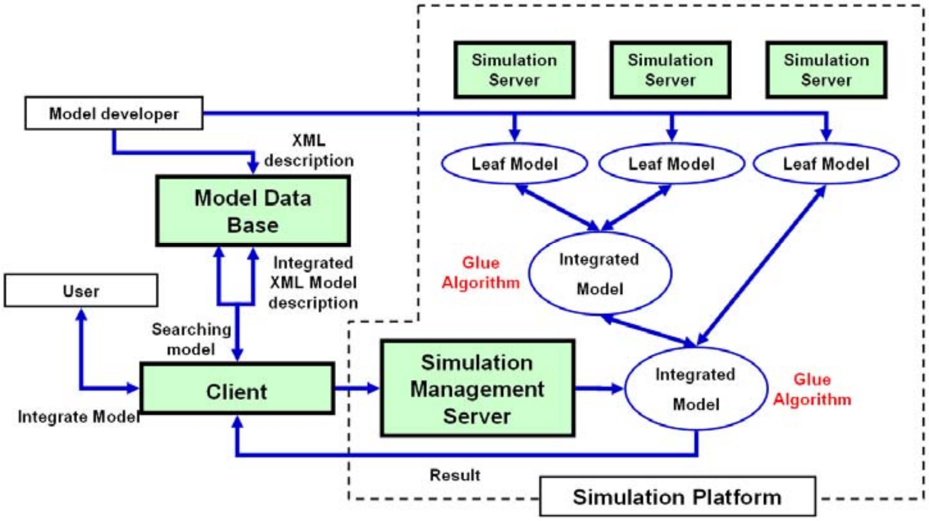
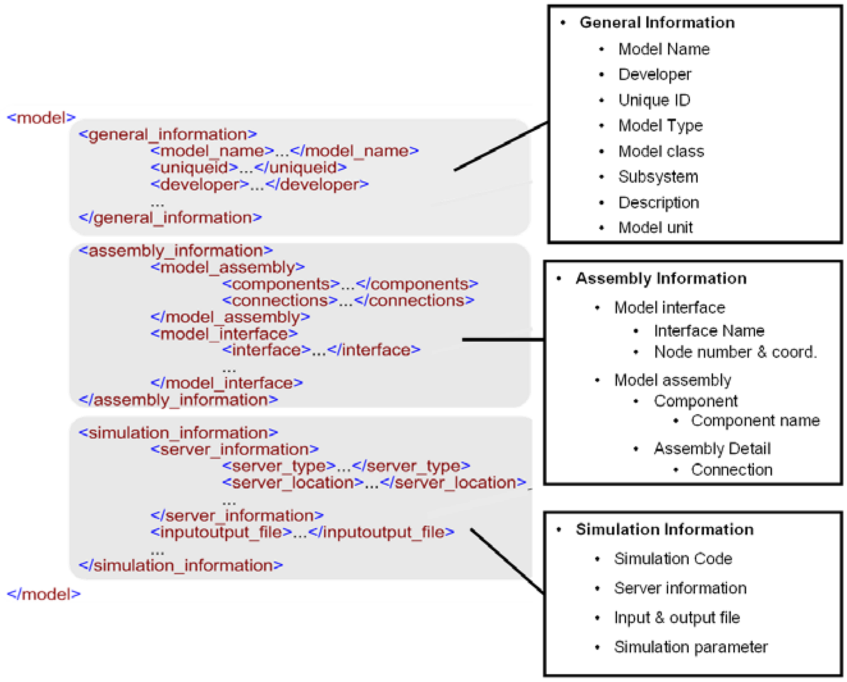


Figure 2.4 Distributed simulation architecture

2.1.1 XML Model Description

The Extensible Markup Language (XML) is currently the standard in web applications and many other applications that require data exchange. XML provides a basic syntax that can be used to share information between different kinds of computers, different applications, and different organizations. In D-Sim, XML is used to describe the mechanical simulation models. The model description needs to be general enough to represent the possible simulation models encountered in the simulation. It should also be flexible so that it can be expanded to describe new models as the simulation platform is extended.



```
<model>
    <general_information>
            <model_name>...</model_name>
            <uniqueid>...</uniqueid>
            <developer>...</developer>
            ...
    </general_information>
    <assembly_information>
            <model_assembly>
                    <components>...</components>
                    <connections>...</connections>
            </model_assembly>
            <model_interface>
                    <interface>...</interface>
                    ...
            </model_interface>
    </assembly_information>
    <simulation_information>
            <server_information>
                    <server_type>...</server_type>
                    <server_location>...</server_location>
                    ...
            </server_information>
            <inputoutput_file>...</inputoutput_file>
            ...
    </simulation_information>
</model>
```

- General Information
    - Model Name
    - Developer
    - Unique ID
    - Model Type
    - Model class
    - Subsystem
    - Description
    - Model unit

- Assembly Information
    - Model interface
        - Interface Name
        - Node number & coord.
    - Model assembly
        - Component
            - Component name
        - Assembly Detail
            - Connection

- Simulation Information
    - Simulation Code
    - Server information
    - Input & output file
    - Simulation parameter

Figure 2.5 Structure of the XML model description

Figure 2.5 shows an example of a standardized model description using XML for describing subsystem models in the gluing simulation. The root element has three child elements, which correspond to the three categories of information for a gluing simulation, namely, General Information, Assembly Information, and Simulation Information. More details of the model description are given in [13]

2.1.2 Gluing Algorithm

The key component to realize the proposed methodology is the gluing algorithm. A gluing methodology not only must be efficient but also must maintain the “privacy” of the individual component models among potentially competing supply chain units. Figure 2.6 illustrates the gluing of a two component system to simulate a chassis frame. In the simulation, both equilibrium conditions and compatibility conditions should be satisfied at the interface. Here, $\mathbf{f}$ and $\mathbf{u}$ refer to the interface force information and the kinematic information (displacement, velocity or acceleration) at the interface. Compatibility conditions refer to the spatial continuity of the displacement, velocity or acceleration fields at the interfaces or joints. A gluing algorithm named the T-T method, was developed and applied to both finite element and multibody dynamics models. The developed algorithm relies only on the interface information of the component models and helps protect the proprietary information of the models.

The developed T-T gluing algorithm has been applied to different simulation models, including finite element and multibody dynamics models and has achieved satisfactory results. However, to apply the gluing algorithm to general simulation problems, e.g., using commercial software packages, further study regarding its convergence, accuracy and efficiency is needed to exploit the full potentials of the gluing algorithm. In Chapter 5, an enhanced gluing algorithm is presented that improves upon the previous T-T gluing algorithm in terms of stability, accuracy and efficiency.

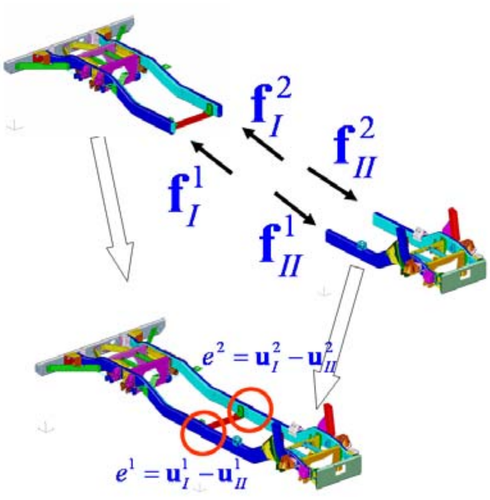

$\mathbf{f}_I^2$
$\mathbf{f}_{II}^2$
$\mathbf{f}_I^1$
$\mathbf{f}_{II}^1$
$e^2 = \mathbf{u}_I^2 - \mathbf{u}_{II}^2$
$e^1 = \mathbf{u}_I^1 - \mathbf{u}_{II}^1$

Figure 2.6 Gluing algorithm illustration

2.1.3 Model Database in Web Based User Interface

In a virtual prototyping environment to support product development, the component models usually are functionally and geographically distributed due to the adoption of multi-layer supply chains as well as the increased collaborations among different engineering and business units. Computer aided system engineering vendors are all moving forward integrating their systems with the Internet. Thus Internet capabilities from inside geometric modeling systems are either already in place or about to become available. By using the Internet as an extension to a geometric modeling system through the use of browsers and browser plug-ins, an entire project team and its clients can view and manipulate models and drawings in various web formats early in the design process.

Fig. 2.7 is an illustration of multi-layer supply chains in the automotive industry and the Army procurement process. The modern automobile industry is formed as a distributed supply chain to save manufacturing cost and to be proficient in individual manufacturing units. The modularization of vehicle design has continuously progressed according to this model of multi-layered manufacturing supply chains. Thus, many digital tools and computing platforms for design and manufacturing have been developed to support these trends. Also the model database in distributed simulation platforms has been multi-layered. As the supply chains are naturally distributed, the CAE models of the components or subsystems of a vehicle are also distributed among different suppliers.

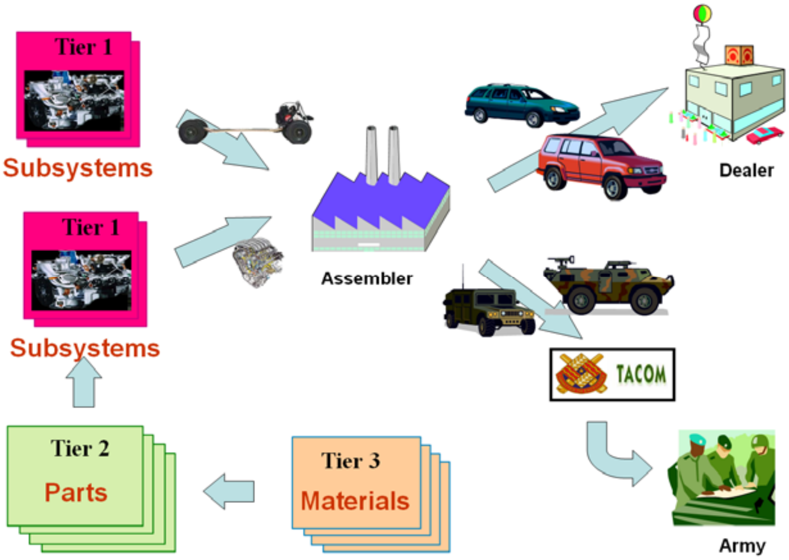


Figure 2.7 Commercial and army vehicle supply chain in modern manufacturing

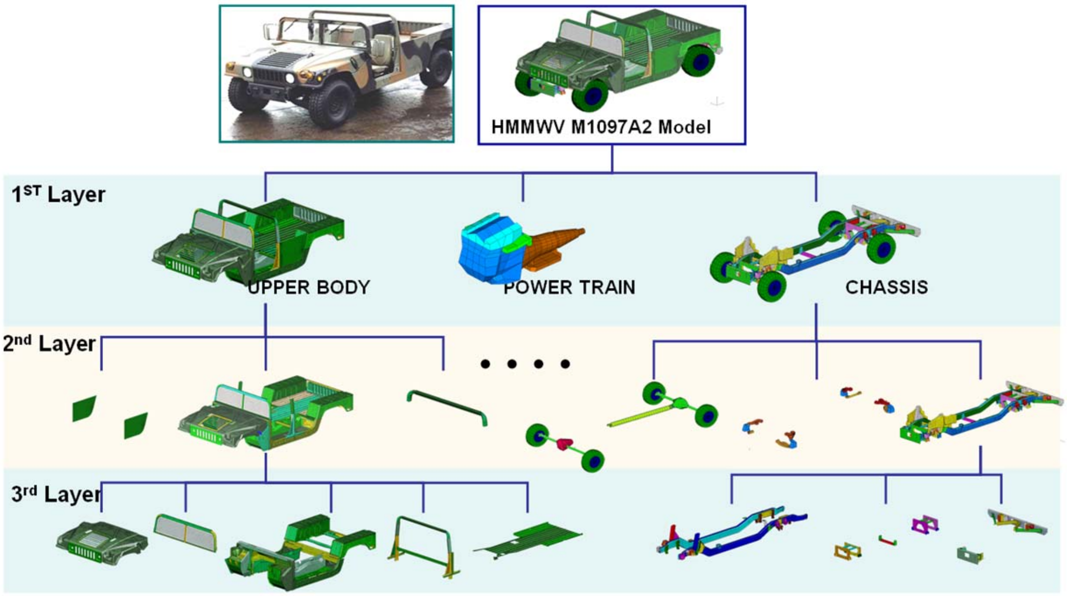
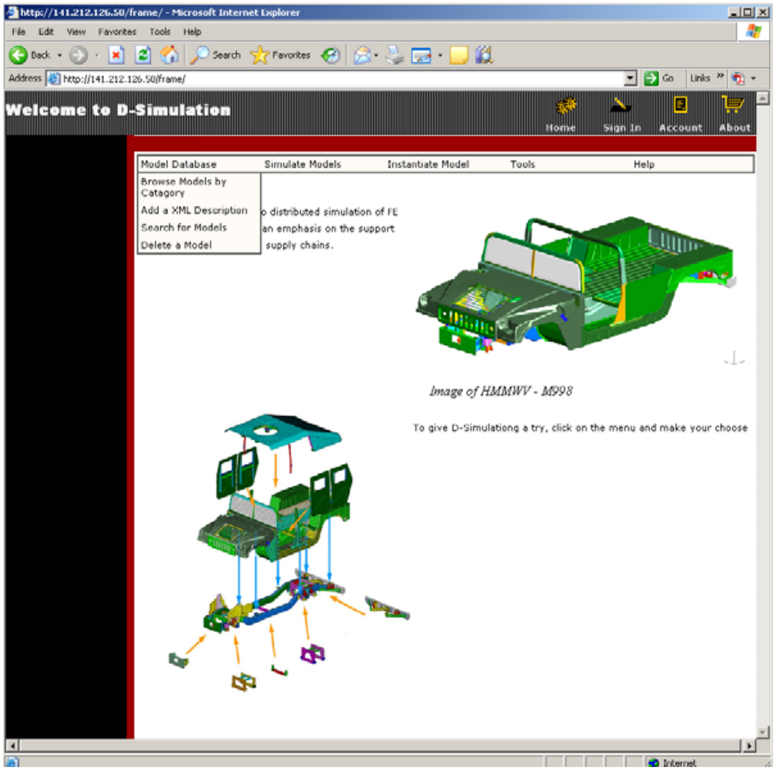


Figure 2.8 Layered system/component model structure in the model server of distributed simulation platform

Figure 2.8 depicts a modularized simulation model that could be parsed across a distributed model server. Each component model and XML model description is stored in the model server on the network, which is accessible through a web server. Each model is uniquely identified by the model ID, simulation server location, interface information, simulation type and so on. Figure 2.9 is a snapshot of the demonstration system, in which a user can browse the model database, assemble and simulate new integrated models as well as view the results. It can be further extended to form a new simulation tool for supporting multilayered simulation and modularized vehicle design.



o distributed simulation of FE
an emphasis on the support
supply chains.

Image of HMMWV - M998

To give D-Simulationg a try, click on the menu and make your choose

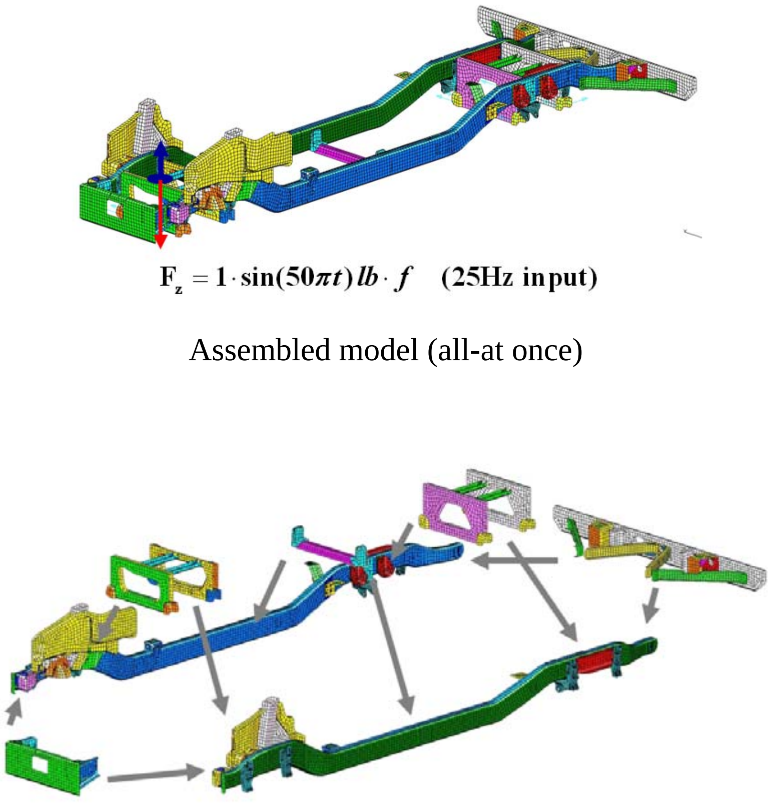
Figure 2.9 Web Based User Interface in D-Sim

2.2 CASE STUDIES OF HMMWV DISTRIBUTED SUBSYSTEM MODES

Case studies of the application of distributed simulation of an Army ground vehicle are shown in this session. Three examples of distributed simulation are provided in order to demonstrate the original gluing algorithm and distributed simulation. First, the mainframe model of HMMWV is demonstrated for verification of original T-T gluing algorithm. Second, the 1st layered distributed simulation model result and 2nd layered distributed model of HMMWV 1097A2 composed body and frame model are compared. Finally, the case of non-matched interface is presented.

2.2.1 Main Frame Model of HMMWV for Verifying Gluing Algorithm

The first case study is a distributed simulation of a frame assembly model, which is a one-layer gluing simulation, assembling six leaf models: Frame rail, front bumper, front cross member, mid cross member, rear cross member and rear bumper model. The integrated main frame model is assembled in the simulation platform by selecting the six component models and matching the interfaces of these components. Figure 2.10 shows the details of the integrated HMMWV main frame model including the mentioned component models and the matching information of the interfaces. Each model server resides on independent computer and is a wrapped FEAP (A Finite Element Analysis Program [81]) code as simulation solver, and the model server is published using HTTP protocol. The total element number of the integrated model is 24989, and nodes number 25156. A sinusoidal loading excites the center of front cross member and the response is reported at same position as the excitation point.



$$F_z = 1 \cdot \sin(50\pi t)\, lb \cdot f \quad \text{(25Hz input)}$$

Assembled model (all-at once)

(b) Distribued models

Figure 2.10 Distributed simulation model of main frame

Table 2.1 The number of interface nodes between subsystem parts in frame model

INTERFACE				NODE NUM
FRAME RAIL	↔	FRT C/MEBER	LH	12
			RH	12
FRAME RAIL	↔	MID C/MEMBER	LH	2
			RH	2
FRAME RAIL	↔	RR C/MEMBER	LH	12
			RH	12
FRAME RAIL	↔	FRT BUMPER	LH	3
			RH	3
FRAME RAIL	↔	RR BUMPER	LH	12
			RH	12
SUM				82

The interface and the number of joints of each interface are presented in Table 2.1. The total number of interfaces is ten and the total number of interface nodes in the simulation model is 82. Figure 2.11 shows a comparison of the results obtained using the gluing simulations with the results obtained using an “all-in-one” finite element analysis. It is seen that the gluing process induces no additional error (beyond the round-off errors) compared to the standard all-at-once simulation result.

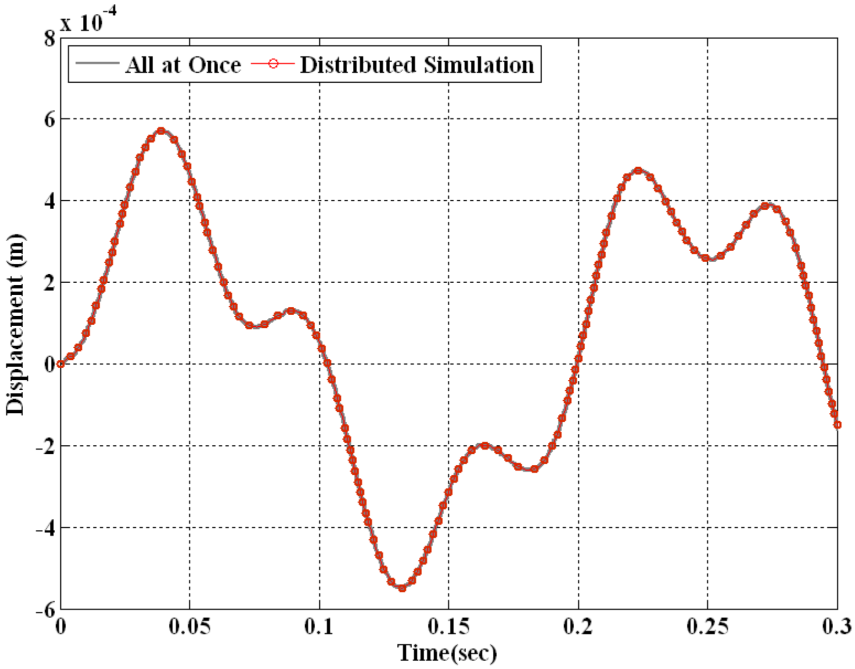


Figure 2.11 Comparison of the simulation results

2.2.2 Multi-Layered Simulation Model

In this example, the layered distributed models composed of main frame and upper body are compared with two gluing layers. The 1st layered distributed model consists of the leaf models of main frame and M1098-A2 upper body. In the second layered simulation model, the integrated main frame is formed with six leaf models, which is the same as the previous example. An integrated upper body is formed by the hood and main body assembly The total number of nodes and elements number is 59920 and 60482, respectively, in the upper body model. Also, the loading condition is same as in the previous example. The simulation system couples the second layer components first to form a higher-level subsystem model, and then integrates the first layer subsystems' models to form the body and frame assembly.

Figure 2.13 shows a comparison of the results obtained using the gluing simulations with 1st and 2nd layered finite element analyses. Here, dynamic loads of $f = 100\sin(50\pi t)\ N$ are applied at the middle point of front cross member The result shown here is the displacement at a selected node at the middle of body floor(node 12093) along the vertical direction. It is seen that the gluing process induces no additional error (beyond the round-off errors) in 1st and 2nd layered distributed simulation results.

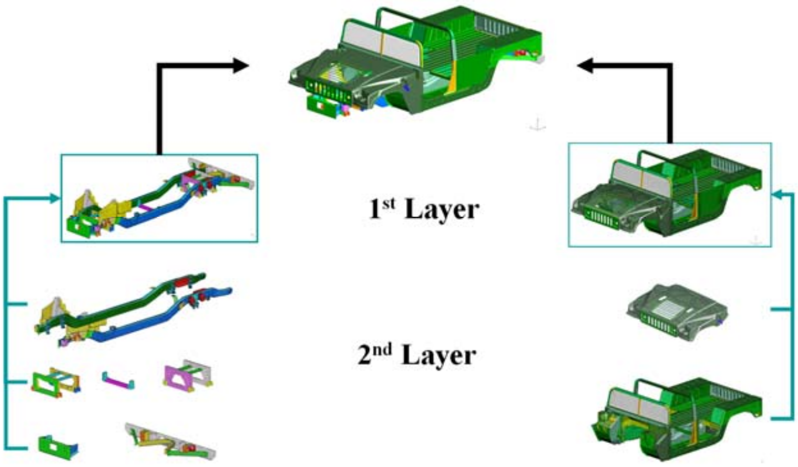


Figure 2.12 2nd layered component model of frame and upper body

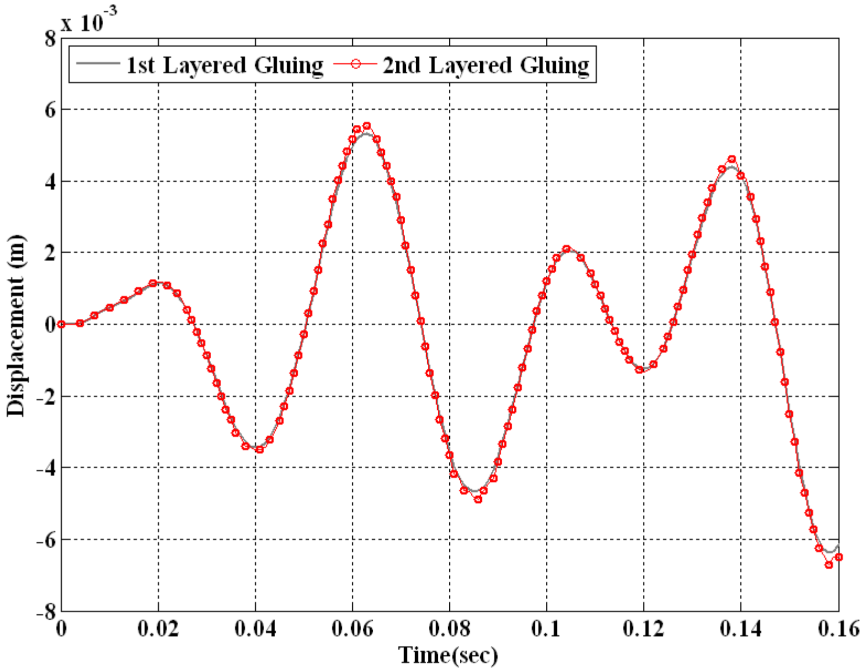


Figure 2.13 Time response of vertical direction at the body floor

2.2.3 Non-matched Interface Treatment

It is very common that the interfaces among the distributed finite element models may not match each other due to the model creation usually being done by different engineers at different organizations. This is an important problem that should be addressed in order to apply a gluing algorithm to practical engineering design problems.

In previous research [17], the treatment of unmatched interfaces was addressed using the Moving Least Square Method. In Finite Element Methods and Meshfree Methods, the continuous field data are represented by interpolating the values on a finite number of discrete points using selected shape functions. More specifically, the interpolation is a linear combination of the shape functions weighted by the values on the discrete points. The Moving Least Square (MLS) method is an approximation method originally proposed for data fitting and has been studied widely recently because it can be used to generate shape functions for the Meshfree method.

To validate the MLS-based non-matched gluing problem, the main frame model in the first example is set up as shown in Fig. 2.14. To compare the gluing results with the solution obtained the matched interfaces of the two finite element models, the interface nodal coordinates of two finite element models are intentionally constructed to be un-matched with each other. Figure 2.15 illustrates a comparison of the vertical-displacement at the middle of front cross member, the same position of loading position. It can be seen that the result of MLS virtual interfaces introduce negligible error in the simulation results.

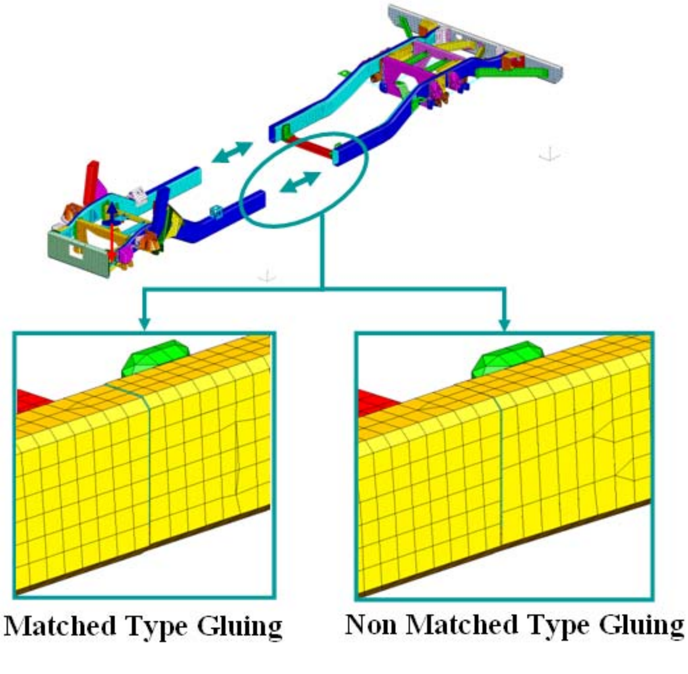


Figure 2.14 Matched and Non-matching type interface of main frame model

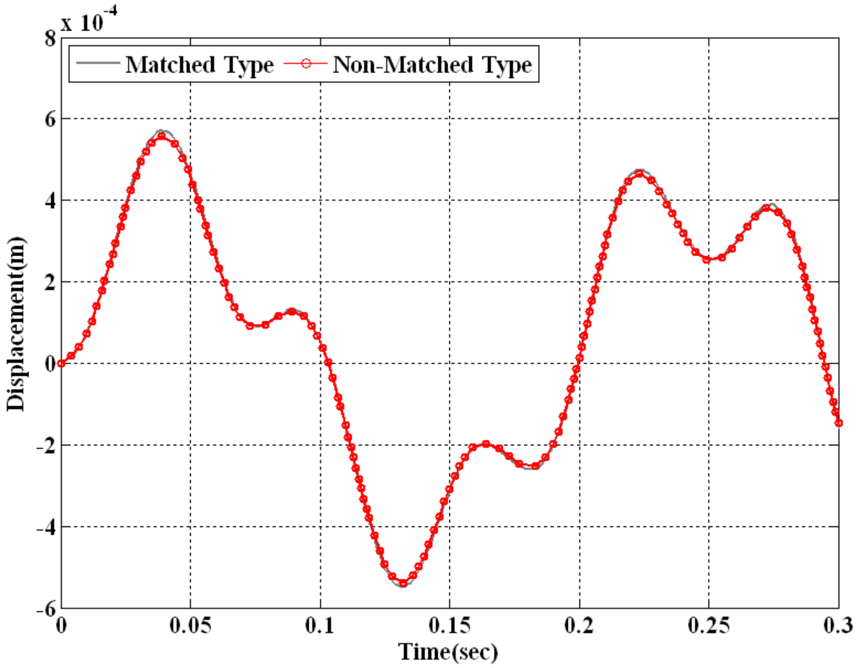


Figure 2.15 Comparison of the responses at the front cross member with matched and non-matched type distributed simulations

CHAPTER 3

PARTITIONED ITERATION METHOD FOR SOLVING FLEXIBLE BODIES HAVING LARGE ROTATIONS AND DISPLACEMENTS

The effect of structural flexibility has become an important issue for more accurate kinematic and dynamic analysis in industrial practice. Many investigations for a flexible multibody system which consists of elastic and rigid components have focused on how the flexible components can be formulated and combined with other component equations.

In this chapter, a Partitioned Iteration Method (PIM) is proposed, which decouples the rigid body motion from elastic deformation of the simulated system using an iteration scheme. This method leads to a floating frame of reference, in which the global motion of body can be expressed by linearized elastic deformations and large overall motion of a reference frame. To decouple the reference motion and relative deformable body motion, the PIM employs mean axis reference conditions, a set of reference conditions defining the body axis and a unique displacement of deformable body. Consequently, one can more easily separate the equations of motion into reference motion and deformable body motion. The PIM has an advantage in that rigid body motion is solved with general DAE solvers, while the deformable body motion is solved with linear FE solvers which are useful in solving an elastic body model having a large number of degrees of freedom. Examples are provided to demonstrate the performance of the method and also how to decouple and integrate rigid body motion and elastic deformation.

3.1 FLOATING FRAME OF REFERENCE

The floating frame of reference method in [61] currently is the most widely used method in the computer simulation of flexible multibody systems. It is implemented in several commercial as well as research general purpose multibody computer programs. In the floating frame of reference, the configuration of a deformable body is identified by using two sets of coordinates: reference and elastic coordinates. Reference coordinates define the location and orientation of a selected body reference. Elastic coordinates, on the other hand, describe the body deformation with respect to the body reference.

The motion of the body is then defined as the motion of its reference plus the motion of the material points on the body with respect to its reference, as depicted in Figure 3.1.

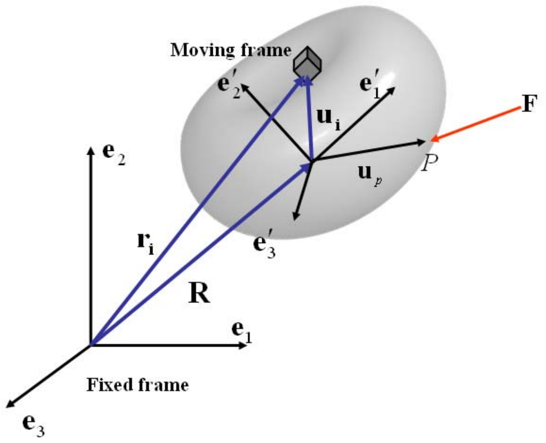


Figure 3.1 Deformable body coordinates

Define a body reference $\mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3$ whose location and orientation with respect to the global coordinates are given by $\mathbf{q}_R$, where $\mathbf{q}_R$ can be written in a partitioned form as

$$\mathbf{q}_R = \begin{bmatrix} \mathbf{R}^T & \boldsymbol{\theta}^T \end{bmatrix}^T \tag{3.1}$$

The vector, $\mathbf{R}$ is a set of Cartesian coordinates that define the location of the origin of the body reference. The vector, $\boldsymbol{\theta}$ is a set of rotational coordinates that describe the orientation of the selected body reference. The body coordinate system $\mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3$ is the floating frame of reference. The global position vector $\mathbf{r}$ of arbitrary point in the deformable body can be written as

$$\mathbf{r} = \mathbf{R} + \mathbf{A}\bar{\mathbf{u}} \tag{3.2}$$

where $\bar{\mathbf{u}}$ is the position vector with respect to the body coordinate, which can be written as

$$\bar{\mathbf{u}} = \bar{\mathbf{u}}_0 + \delta\bar{\mathbf{u}} \tag{3.3}$$

in which $\bar{\mathbf{u}}_0$ is the undeformed position vector and $\delta\bar{\mathbf{u}}$ is the elastic deformation vector of the point. **A** is a transformation matrix, where the orientation of the body reference can be identified using, e.g., three independent Euler angles, Rodriquez parameters or the four dependent Euler parameters which are provide in Appendix A.

In the finite element method, the motions of arbitrary points are related to the position of element nodes, with which we can calculate the displacement field and stress field in the finite elements using the associated element shape functions. Consequently, for an arbitrary point within the *i-th* element, we can write the position vector as

$$\bar{\mathbf{u}}_i = \bar{\mathbf{u}}_{i0} + \delta\bar{\mathbf{u}}_i = \mathbf{N}_i\left(\mathbf{q}_{i0} + \mathbf{q}_{fi}\right) \tag{3.4}$$

where $\mathbf{q}_{i0}$ is the undeformed nodal vector, $\mathbf{N}_i$ is element shape function of *i-th* lement, which transformed in the body coordinates system, and $\mathbf{q}_{fi}$ is nodal displacement vector. Consequently, global location vector $\mathbf{r}_i$ can be written as

$$\mathbf{r}_i = \mathbf{R} + \mathbf{A}\mathbf{N}_i\left(\mathbf{q}_{i0} + \mathbf{q}_{fi}\right) \tag{3.5}$$

or as

$$\mathbf{r}_i = \mathbf{R} + \mathbf{A}\underline{\mathbf{N}}_i\left(\mathbf{q}_0 + \mathbf{q}_f\right) = \mathbf{R} + \mathbf{A}\underline{\mathbf{N}}_i\mathbf{u}_f \tag{3.6}$$

where $\underline{\mathbf{N}}_i$ is the connectivity matrix concatenation of $\mathbf{N}_i$ for generalized coordinates $\mathbf{q}_f$, and $\mathbf{q}_f$ is nodal vector including all degrees of freedom of the deformed body. The velocity vector of an arbitrary point in *i-th* element is given by

$$\dot{\mathbf{r}}_i = \dot{\mathbf{R}} + \dot{\mathbf{A}}\underline{\mathbf{N}}_i\mathbf{u}_f + \mathbf{A}\underline{\mathbf{N}}_i\dot{\mathbf{q}}_f \tag{3.7}$$

where the superposed dot denotes differentiation with respect to time. The middle term, $\dot{\mathbf{A}}\underline{\mathbf{N}}_i\mathbf{u}_f$, can be written as

$$\dot{\mathbf{A}}\underline{\mathbf{N}}_i\mathbf{u}_f = \mathbf{A}_{\theta i}\left(\mathbf{u}_f, \boldsymbol{\theta}\right)\dot{\boldsymbol{\theta}} \tag{3.8}$$

in order to isolate velocity terms. The matrix $\mathbf{A}_{\theta i}\left(\mathbf{u}_f, \boldsymbol{\theta}\right)$ is a function of the reference rotational coordinates and the elastic coordinate of the body. If the transformation matrix **A** is described by Euler parameters, then

$$\mathbf{A} = \begin{bmatrix} 1-2(\theta_2)^2-2(\theta_3)^2 & 2(\theta_1\theta_2-\theta_0\theta_3) & 2(\theta_1\theta_3+\theta_0\theta_2) \\ 2(\theta_1\theta_2+\theta_0\theta_3) & 1-2(\theta_1)^2-2(\theta_3)^2 & 2(\theta_2\theta_3-\theta_0\theta_1) \\ 2(\theta_1\theta_3-\theta_0\theta_2) & 2(\theta_2\theta_3+\theta_0\theta_1) & 1-2(\theta_1)^2-2(\theta_2)^2 \end{bmatrix} \tag{3.9}$$

where

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 & \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^T$$
$$\theta_0 = \cos\frac{\theta}{2}, \quad \theta_1 = v_1 \sin\frac{\theta}{2}, \quad \theta_2 = v_2 \sin\frac{\theta}{2}, \quad \theta_3 = v_3 \sin\frac{\theta}{2} \tag{3.10}$$

The vector $\begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}^T$ is the axis vector of rotation and then, the matrix $\mathbf{A}_{\theta i}\left(\mathbf{u}_f, \boldsymbol{\theta}\right)$ can be written as

$$\mathbf{A}_{\theta i}\left(\mathbf{u}_f, \boldsymbol{\theta}\right) = -\mathbf{A}\tilde{\bar{\mathbf{u}}}_i \bar{\mathbf{G}} \tag{3.11}$$

where

$$\bar{\mathbf{G}} = 2\begin{bmatrix} -\theta_1 & \theta_0 & \theta_3 & -\theta_2 \\ -\theta_2 & -\theta_3 & \theta_0 & \theta_1 \\ -\theta_3 & \theta_2 & -\theta_1 & \theta_0 \end{bmatrix}, \quad \bar{\mathbf{u}} = \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \\ \bar{u}_3 \end{bmatrix}, \quad \tilde{\bar{\mathbf{u}}} = \begin{bmatrix} 0 & -\bar{u}_3 & \bar{u}_2 \\ \bar{u}_3 & 0 & -\bar{u}_1 \\ -\bar{u}_2 & \bar{u}_1 & 0 \end{bmatrix}$$

Thus, the velocity vector of Equation (3.7) can be written in partitioned form as

$$\dot{\mathbf{r}}_i = \begin{bmatrix} \mathbf{I} & -\mathbf{A}\tilde{\bar{\mathbf{u}}}_i \bar{\mathbf{G}} & \mathbf{A}\underline{\mathbf{N}}_i \end{bmatrix} \begin{bmatrix} \dot{\mathbf{R}} \\ \dot{\boldsymbol{\theta}} \\ \dot{\mathbf{q}}_f \end{bmatrix} \tag{3.12}$$

Once the velocity vector of an arbitrary point on *i-th* element is defined, one can write the kinetic energy of body as

$$T = \sum_{i=1}^{N_e} \frac{1}{2} \int_{\Omega_e} \rho \dot{\mathbf{r}}_i^T \cdot \dot{\mathbf{r}}_i \; d\Omega_e \tag{3.13}$$

where N_e is total number of elements and ρ is material density. By substituting Equation (3.12) into Equation (3.13), we obtain

$$T=\frac{1}{2}\begin{bmatrix}\dot{\mathbf{R}}^T & \dot{\boldsymbol{\theta}}^T & \dot{\mathbf{q}}_f{}^T\end{bmatrix}\sum_{i=1}^{N_e}\int_{\Omega_e}\rho\begin{bmatrix}\mathbf{I} & -\mathbf{A}\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}} & \mathbf{A}\underline{\mathbf{N}}_i\\ -\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\mathbf{A}^T & \bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}} & -\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\underline{\mathbf{N}}_i\\ \underline{\mathbf{N}}_i{}^T\mathbf{A}^T & -\underline{\mathbf{N}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}} & \underline{\mathbf{N}}_i{}^T\underline{\mathbf{N}}_i\end{bmatrix}d\Omega_e\begin{bmatrix}\dot{\mathbf{R}}\\ \dot{\boldsymbol{\theta}}\\ \dot{\mathbf{q}}_f\end{bmatrix} \quad (3.14)$$

The kinetic energy, given by Equation (3.14), can be written in a partitioned form with the body reference generalized coordinates and the elastic coordinates as

$$T=\frac{1}{2}\begin{bmatrix}\dot{\mathbf{q}}_R{}^T & \dot{\mathbf{q}}_f{}^T\end{bmatrix}\sum_{i=1}^{N_e}\int_{\Omega_e}\rho\begin{bmatrix}\mathbf{M}_{RR} & \mathbf{M}_{Rf}\\ \mathbf{M}_{fR} & \mathbf{M}_{ff}\end{bmatrix}d\Omega_e\begin{bmatrix}\dot{\mathbf{q}}_R\\ \dot{\mathbf{q}}_f\end{bmatrix} \quad (3.15)$$

Matrices $\mathbf{M}_{Rf}$ and $\mathbf{M}_{fR}$ $(\mathbf{M}_{Rf}=\mathbf{M}_{fR}^T)$, are the coupling mass matrices between the reference and elastic coordinates, which are dependent on the generalized coordinates and body reference conditions. A decoupling between reference and elastic coordinates can be achieved either by neglecting the off-diagonal matrices ($\mathbf{M}_{Rf}$ and $\mathbf{M}_{fR}$), or by choosing a proper set of reference conditions. Using energy and virtual work expressions, one can write the variational form of the system equation of motion for the flexible body which is provided in Appendix B.

$$\begin{aligned}&\sum_{i=1}^{N_e}\int_{\Omega_e}\rho\begin{bmatrix}\mathbf{I} & -\mathbf{A}\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}} & \mathbf{A}\underline{\mathbf{N}}_i\\ -\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\mathbf{A}^T & \bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}} & -\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\underline{\mathbf{N}}_i\\ \underline{\mathbf{N}}_i{}^T\mathbf{A}^T & -\underline{\mathbf{N}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}} & \underline{\mathbf{N}}_i{}^T\underline{\mathbf{N}}_i\end{bmatrix}d\Omega_e\begin{bmatrix}\ddot{\mathbf{R}}\\ \ddot{\boldsymbol{\theta}}\\ \ddot{\mathbf{q}}_f\end{bmatrix}+\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \mathbf{F}^{\mathrm{int}}\end{bmatrix}\\ &=\begin{bmatrix}\mathbf{F}_r{}^{\mathbf{ext}}\\ \mathbf{F}_\theta{}^{\mathbf{ext}}\\ \mathbf{F}_f{}^{\mathbf{ext}}\end{bmatrix}+\sum_{i}^{N_e}\int_{\Omega_e}\rho\begin{bmatrix}2\mathbf{A}\dot{\tilde{\bar{\mathbf{u}}}}_i\bar{\mathbf{G}}\dot{\boldsymbol{\theta}}+\mathbf{A}\tilde{\bar{\boldsymbol{\omega}}}\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}\dot{\boldsymbol{\theta}}\\ -2\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}^T\dot{\tilde{\bar{\mathbf{u}}}}\bar{\mathbf{G}}\dot{\boldsymbol{\theta}}-2\dot{\bar{\mathbf{G}}}^T\tilde{\bar{\mathbf{u}}}^T\tilde{\bar{\mathbf{u}}}\bar{\mathbf{G}}\dot{\boldsymbol{\theta}}+2\dot{\bar{\mathbf{G}}}^T\tilde{\bar{\mathbf{u}}}^T\dot{\mathbf{u}}\\ 2\underline{\mathbf{N}}_i{}^T\dot{\tilde{\bar{\mathbf{u}}}}_i\bar{\mathbf{G}}\dot{\boldsymbol{\theta}}+\underline{\mathbf{N}}_i{}^T\tilde{\bar{\boldsymbol{\omega}}}^T\tilde{\bar{\boldsymbol{\omega}}}\bar{\mathbf{u}}_i\end{bmatrix}^T d\Omega_e\end{aligned} \quad (3.16)$$

where $\begin{bmatrix}\mathbf{F}_r{}^{\mathbf{ext}T} & \mathbf{F}_\theta{}^{\mathbf{ext}T} & \mathbf{F}_f{}^{\mathbf{ext}T}\end{bmatrix}^T$ is generalized external force vector. The virtual work of all external forces acting on flexible body in compact form as

$$\delta W = \begin{bmatrix} \mathbf{F}_r^{\mathbf{ext}T} & \mathbf{F}_\theta^{\mathbf{ext}T} & \mathbf{F}_f^{\mathbf{ext}T} \end{bmatrix} \delta \mathbf{r} = \begin{bmatrix} \mathbf{F}_r^{\mathbf{ext}T} & \mathbf{F}_\theta^{\mathbf{ext}T} & \mathbf{F}_f^{\mathbf{ext}T} \end{bmatrix} \begin{bmatrix} \delta \mathbf{R} \\ \delta \boldsymbol{\theta} \\ \delta \mathbf{q}_f \end{bmatrix} \tag{3.17}$$

In the Figure 3.1, the force $\mathbf{F}$ acts at point P of the deformable body, and has three components, which can be defined in the global coordinate system. Then, one can write as

$$\mathbf{F}_r^{\mathbf{ext}} = \mathbf{F}\,, \quad \mathbf{F}_\theta^{\mathbf{ext}} = \bar{\mathbf{G}}^T \tilde{\bar{\mathbf{u}}}_p{}^T \mathbf{A}^T \mathbf{F} \qquad \text{and,} \qquad \mathbf{F}_f^{\mathbf{ext}} = \underline{\mathbf{N}}_p{}^T \mathbf{A}^T \mathbf{F} \tag{3.18}$$

A partitioned form of the equations of motion, using the body reference generalized coordinates and the elastic coordinates, is given by

$$\begin{bmatrix} \mathbf{M}_{RR} & \mathbf{M}_{Rf} \\ \mathbf{M}_{fR} & \mathbf{M}_{ff} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}}_R \\ \ddot{\mathbf{q}}_f \end{Bmatrix} + \begin{Bmatrix} 0 \\ \mathbf{F}^{\mathrm{int}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_R^{\mathbf{ext}} \\ \mathbf{F}_f^{\mathbf{ext}} \end{Bmatrix} + \begin{Bmatrix} \mathbf{Q}_R \\ \mathbf{Q}_f \end{Bmatrix} \tag{3.19}$$

where $\mathbf{Q}_R$ and $\mathbf{Q}_f$ are, respectively, the quadratic velocity vector of the rigid (reference) and the flexible body. These terms include the effect of the Coriolis and centrifugal forces. The quadratic velocity terms are coupled with $\dot{\mathbf{q}}_R$, $\dot{\mathbf{q}}_f$ and $\mathbf{q}_f$ resulting in a nonlinear function of the generalized coordinates and velocity coordinates.

One cannot directly solve Equation (3.19), because the number of equations is larger than the number of degree of freedom of flexible body, which is the same as the size of $\mathbf{q}_f$, as $\mathbf{q}_f$ already includes rigid body modes. Thus, a reference condition is necessary in order to eliminate the rigid body modes in the assumed displacement field. In the rigid-body analysis, a fixed body axis that is rigidly attached to a point on the body or an extension thereof is commonly employed, whereas a moving reference body is suggested in the analysis of flexible bodies as shown in Figure 3.2.

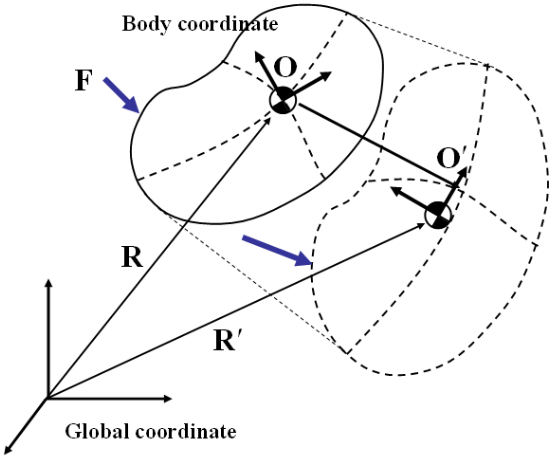


Figure 3.2 Moving frame of reference

The choice of the deformable-body reference and the associated reference conditions, which are required to define a unique displacement field, are of major concern when modeling a constrained component in multibody systems. The mean-axis conditions[80], which are the result of minimizing the kinetic energy of the elastic motion with respect to an observer stationed on the deformable body, imply that the linear and angular momentum due to elastic deformation are zero. The relative kinetic energy of the *i-th* element with respect to body coordinate is defined by

$$T_i^{rel} = \frac{1}{2}\int_{\Omega_e} \rho_i \dot{\mathbf{u}}_i^T \cdot \dot{\mathbf{u}}_i \; d\Omega_e \tag{3.20}$$

The total relative kinetic energy of all elements of a body with respect to its body fixed coordinate system can be written as

$$T^{rel} = \sum_{i=1}^{N_e} \frac{1}{2} \int_{\Omega_e} \rho_i \left(\dot{\mathbf{r}}_i - \dot{\mathbf{R}} - \dot{\mathbf{A}}\underline{\mathbf{N}}_i \mathbf{u}_f \right)^T \cdot \left(\dot{\mathbf{r}}_i - \dot{\mathbf{R}} - \dot{\mathbf{A}}\underline{\mathbf{N}}_i \mathbf{u}_f \right) d\Omega_e \tag{3.21}$$

For minimum relative kinetic energy, the partial derivatives with respect to $\dot{\mathbf{R}}$ and $\dot{\boldsymbol{\theta}}$ should be zero, i.e.,

$$\frac{\partial T_{rel}}{\partial \dot{\mathbf{R}}} = \mathbf{0}, \quad \frac{\partial T_{rel}}{\partial \dot{\boldsymbol{\theta}}} = \mathbf{0} \tag{3.22}$$

These result in

$$\left(\sum_{i=1}^{N_e} \int_{\Omega_e} \rho_i \underline{\mathbf{N}}_i d\Omega_e \right) \dot{\mathbf{q}}_f = \left(\sum_{i=1}^{N_e} \int_{\Omega_e} \rho_i \dot{\bar{\mathbf{u}}}_i d\Omega_e \right) = \mathbf{0} \tag{3.23}$$

$$\left(\sum_{i=1}^{N_e} \int_{\Omega_e} \rho_i \mathbf{A}_{\theta i}{}^T \mathbf{A} \underline{\mathbf{N}}_i d\Omega_e \right) \dot{\mathbf{q}}_f = \left(\sum_{i=1}^{N_e} \int_{\Omega_e} \rho_i \bar{\mathbf{u}}_i \times \dot{\bar{\mathbf{u}}}_i d\Omega_e \right) = \mathbf{0} \tag{3.24}$$

Equation (3.23) is the positional condition of reference such that the body reference frame always moves with the mass center, which is not fixed in the body during analysis as shown in the Figure 3.2. Equation (3.24) is the rotational condition of reference such that the axis of body reference is always collinear with the axis which makes no rigid rotation of flexible body with respect to reference coordinates during analysis. These equations result in a set of constraint equations that are sufficient to uniquely define the displacement field.

$$\begin{bmatrix} \mathbf{M}_{rr} & \mathbf{M}_{r\theta} & \mathbf{M}_{rf} \\ \mathbf{M}_{\theta r} & \mathbf{M}_{\theta\theta} & \mathbf{M}_{\theta f} \\ \mathbf{M}_{fR} & \mathbf{M}_{f\theta} & \mathbf{M}_{ff} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{R}} \\ \ddot{\boldsymbol{\theta}} \\ \ddot{\mathbf{q}}_f \end{Bmatrix} + \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{F}^{\mathbf{int}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_r{}^{\mathbf{ext}} \\ \mathbf{F}_\theta{}^{\mathbf{ext}} \\ \mathbf{F}_f{}^{\mathbf{ext}} \end{Bmatrix} + \begin{Bmatrix} \mathbf{Q}_r \\ \mathbf{Q}_\theta \\ \mathbf{Q}_f \end{Bmatrix} \tag{3.25}$$

In Equation (3.25), the mass matrix and the quadratic velocity force terms **Q** are functions of the nodal displacement and velocity vectors. Applying the mean axis reference conditions of Equations (3.23) and (3.24) to Equation (3.25), provided the initial position of reference is attached to mass center of body, results in

$$\begin{bmatrix} \mathbf{M}_{rr} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\theta\theta} & \mathbf{0} \\ \mathbf{M}_{fR} & \mathbf{M}_{f\theta} & \mathbf{M}_{ff} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{R}} \\ \ddot{\boldsymbol{\theta}} \\ \ddot{\mathbf{q}}_f \end{Bmatrix} + \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{F}^{\mathbf{int}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_r^{\mathbf{ext}} \\ \mathbf{F}_\theta^{\mathbf{ext}} \\ \mathbf{F}_f^{\mathbf{ext}} \end{Bmatrix} + \begin{Bmatrix} \mathbf{0} \\ \mathbf{Q}_\theta \\ \mathbf{Q}_f \end{Bmatrix} \tag{3.26}$$

The equation of motion includes the reference condition which implies the body coordinates lie on the point at which the average deformation across deformable body is be zero. In Equation (3.26), the coupling terms in the mass and quadratic velocity terms can be removed using the mean axis reference condition. The mass matrix in this equation is non-singular. Therefore, if all the forces are known, this equation can be solved. But there is no guarantee that the solution of Equation (3.26) satisfies the mean axis reference condition, because numerical error in mean axis condition can accumulate and have adverse effects on the relative deformed body equation part during integration. In other words, Equation (3.26) is a necessary but not sufficient condition to satisfy mean axis reference conditions. Nikravesh P. E. [82] noted the reason is that the mean-axis conditions at the acceleration level are not explicitly present. Although the amount of violation could be small, it is possible that during forward integration of the equations of motion, the error could cause numerical instability. To understand the physical meaning of this instability, one can extract deformed body equation of motion from Equation (3.26) as follows:

$$\mathbf{M}_{ff}\ddot{\mathbf{q}}_f + \mathbf{F}^{\mathbf{int}} = \mathbf{F}_f^{\mathbf{ext}} + \mathbf{Q}_f - \mathbf{M}_{fr}\ddot{\mathbf{R}} - \mathbf{M}_{f\theta}\ddot{\boldsymbol{\theta}} \tag{3.27}$$

The right hand side of Equation (3.27) implies that from the external forces are subtracted the inertial forces calculated with the motion of reference frame. To satisfy the mean axis reference conditions, there should be no rigid body motion in the solution of the deformable body of Equation (3.27). Consequently, the force terms should be in static and/or dynamic equilibrium in six directions in the three dimensional problem. Since the geometric stiffening terms generated by geometric non-linearity [83] are not considered in equation (3.26), there might be slight errors and consequent non-zero forces generating rigid body displacement during forward integration of the equations of motion.

In order to eliminate this particular error, Equations (3.23) and (3.24) can be incorporated into Equation (3.26) or Equation (3.25) with the aid of Lagrange multipliers to obtain

$$\begin{bmatrix} \mathbf{M}_{rr} & \mathbf{0} & \mathbf{0} & \mathbf{C}_{qr}{}^{T} \\ \mathbf{0} & \mathbf{M}_{\theta\theta} & \mathbf{0} & \mathbf{C}_{q\theta}{}^{T} \\ \mathbf{M}_{fr} & \mathbf{M}_{f\theta} & \mathbf{M}_{ff} & \mathbf{C}_{qf}{}^{T} \\ \mathbf{C}_{qr} & \mathbf{C}_{q\theta} & \mathbf{C}_{qf} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{R}} \\ \ddot{\boldsymbol{\theta}} \\ \ddot{\mathbf{q}}_{f} \\ \underline{\boldsymbol{\lambda}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{F}^{\mathbf{int}} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{r}{}^{\mathbf{ext}} \\ \mathbf{F}_{\theta}{}^{\mathbf{ext}} \\ \mathbf{F}_{f}{}^{\mathbf{ext}} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{Q}_{\theta} \\ \mathbf{Q}_{f} \\ \mathbf{0} \end{bmatrix} \qquad (3.28)$$

where,

$$\mathbf{C}_{qr} = \overset{6\times3}{[\mathbf{0}]},\ \mathbf{C}_{qt} = \overset{6\times4}{[\mathbf{0}]},\quad \mathbf{C}_{qf} = \overset{6\times N_e}{\begin{bmatrix} \sum_{i}^{N_e} \int_{\Omega} \underline{\mathbf{N}}\, d\Omega \\ \sum_{i}^{N_e} \int_{\Omega} \bar{\mathbf{u}} \times \underline{\mathbf{N}}\, d\Omega \end{bmatrix}} = \begin{bmatrix} \mathbf{M}_{rf} \\ \mathbf{M}_{\theta f} \end{bmatrix}$$

In Equation (3.28), the mean axis reference condition is expressed through constraint equation form. Lagrange multipliers only affect the deformed body equation of motion with respect to reference frame in equation. The summation of external forces and generalized force ($\mathbf{F}_f^{\mathbf{ext}} + \mathbf{Q}_f - \mathbf{M}_{fr}\ddot{\mathbf{R}} - \mathbf{M}_{f\theta}\ddot{\boldsymbol{\theta}}$ in Equation 3.27) implies that the external forces on the elastic body are extracted with the inertial forces calculated from the response of reference frame. Also, constraint force terms including Lagrange multipliers make sure that the right side of Equation (3.27) induces no rigid body motion in the elastic deformed body. When we derived the formulation, basically we assumed linear elastic deformation of flexible body. However, the equation of motion in Equation (3.25) or (3.26) is coupled with the non-linear multibody dynamic equation. Therefore, the solution of Equation (3.26) slightly violates the mean axis reference condition. Consequently, the error in the deformable body equation of motion may be numerically instable. Equation (3.28), in which coupling terms of relative deformed body and reference variables can be removed, is able to be easily separated into a rigid body motion and a relative deformable body motion with respect to rigid motion.

3.2 PARTITIONED ITERATION METHOD (PIM)

The Partitioned Iteration Method (PIM) is proposed for a solution strategy that can decouple the solutions of the rigid body and elastic body equations of motion, by decoupling the rigid body motion from the elastic deformation using an iteration scheme. As depicted in Figure 3.3, it employs a CG-following reference frame for each deformable body in the system. The axis and position of this CG are not fixed at the initial CG point, but are updated with respect to the body deformation. This approach was proposed as the Linear elasto-dynamic method(Winfrey [31] and Popp [76]), but in this Linear elasto-dynamics approach, the coupling force terms $\mathbf{Q}_f$ in Equation (3.28) of flexible body motion and rigid body motion equations are neglected due to the assumption of relatively small coupling terms. PIM has the advantage that rigid body motion is solved with general DAE solvesr, while the deformable body motion is solved with linear ODE solvers. The fact that the deformable body is solved with a linear FE solver is of direct benefit for distributed simulation of flexible multibody systems because the deformable body model already is contained on the linear elastic deformation model server.

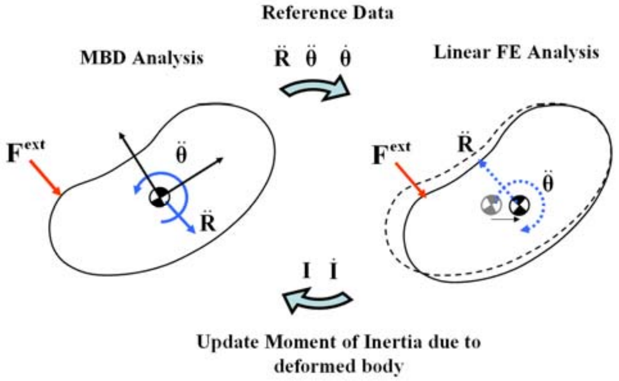


Figure 3.3 Partition of reference motion and body deformation analysis

3.2.1 Approximated Multipliers with Mean Axis Reference

To develop a Partitioned Iteration Method, we use the floating frame of reference approach,

$$\begin{bmatrix} \mathbf{M}_{rr} & \mathbf{0} & \mathbf{0} & \mathbf{C}_{qr}{}^{T} \\ \mathbf{0} & \mathbf{M}_{\theta\theta} & \mathbf{0} & \mathbf{C}_{q\theta}{}^{T} \\ \mathbf{M}_{fr} & \mathbf{M}_{f\theta} & \mathbf{M}_{ff} & \mathbf{C}_{qf}{}^{T} \\ \mathbf{C}_{qr} & \mathbf{C}_{q\theta} & \mathbf{C}_{qf} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{R}} \\ \ddot{\boldsymbol{\theta}} \\ \ddot{\mathbf{q}}_{f} \\ \underline{\boldsymbol{\lambda}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{F}^{\mathbf{int}} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{r}{}^{\mathbf{ext}} \\ \mathbf{F}_{\theta}{}^{\mathbf{ext}} \\ \mathbf{F}_{f}{}^{\mathbf{ext}} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{Q}_{\theta} \\ \mathbf{Q}_{f} \\ \mathbf{0} \end{bmatrix} \tag{3.28}$$

Then, Equation (3.28) can be partitioned to separate the reference body and deformable body equations:

$$\begin{bmatrix} \mathbf{M}_{rr} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\theta\theta} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{R}} \\ \ddot{\boldsymbol{\theta}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_{r}{}^{\mathbf{ext}} \\ \mathbf{F}_{\theta}{}^{\mathbf{ext}} \end{Bmatrix} + \begin{Bmatrix} \mathbf{0} \\ \mathbf{Q}_{\theta} \end{Bmatrix} \tag{3.29}$$

$$\begin{aligned} \mathbf{M}_{ff}\ddot{\mathbf{q}}_{f} + \mathbf{F}^{\mathbf{int}} &= \mathbf{F}_{General} - \mathbf{C}_{qf}^{T}\underline{\boldsymbol{\lambda}} \\ \mathbf{F}_{General} &= \mathbf{F}_{f}{}^{ext} + \mathbf{Q}_{f} - \mathbf{M}_{fr}\ddot{\mathbf{R}} - \mathbf{M}_{f\theta}\ddot{\boldsymbol{\theta}} \end{aligned} \tag{3.30}$$

In Equations (3.29) and (3.30), the vectors $\mathbf{Q}_{\theta}$ and $\mathbf{Q}_{f}$ are assumed, for the initial iteration, to be independent of the elastic deformations of the body. Equation (3.29) can be solved for the reference coordinates, velocities, and accelerations in the MBD program. The results obtained from the rigid body analysis can then be substituted into Equation (3.30) and the linear structural problem can be solved, including the influence of the rigid body response on the elastic deformation. The Lagrange multiplier is still unknown because the constraint equation was not included in Equation (3.30). In the previous section, multipliers contribute to the solution of deformable body to satisfy the mean axis reference condition, which means that the elastic body with respect to the reference frame deforms without rigid translational motion or rotational motion, i.e., the applied force $\mathbf{F}_{General}$ has to be force equilibrated during integration. For force equilibrium, rigid body forces, which are defined in Equation (3.31), should be zero.

$$\mathbf{F}_{General} = \begin{bmatrix} \mathbf{F}^{T\,1}_{General} \\ \mathbf{F}^{R\,1}_{General} \\ \vdots \\ \mathbf{F}^{T\,N}_{General} \\ \mathbf{F}^{R\,N}_{General} \end{bmatrix} \qquad \mathbf{F}^{Ti}_{General} = \begin{bmatrix} F^{x\,i}_{general} \\ F^{y\,i}_{general} \\ F^{z\,i}_{general} \end{bmatrix} \qquad \mathbf{F}^{R\,i}_{General} = \begin{bmatrix} F^{\theta x\,i}_{general} \\ F^{\theta y\,i}_{general} \\ F^{\theta z\,i}_{general} \end{bmatrix} \tag{3.31}$$

$$\mathbf{F}_{Rigid\_t} = \left[\sum_{i}^{N} \mathbf{F}^{T\,i}_{general} \right] \qquad \mathbf{F}_{Rigid\_r} = \left[\sum_{i}^{N} (\mathbf{F}^{R\,i}_{general} + \overline{\mathbf{u}}_i \times \mathbf{F}^{T\,i}_{general}) \right] \tag{3.32}$$

Therefore, in Equation (3.31), $\mathbf{F}_{General}$ should be corrected with constraint forces such that the deformable body can deform without rigid body motion. As shown in Equation (3.31), using the rigid body force defined in Equation (3.32), approximated multipliers can be calculated by multiplying the summation forces in each direction of $\mathbf{F}_{General}$ by the inverse of the mass matrix in Equation (3.30) without additional constraint equations.

$$\underline{\boldsymbol{\lambda}} = \begin{bmatrix} \underline{\boldsymbol{\lambda}}_t \\ \underline{\boldsymbol{\lambda}}_r \end{bmatrix} \approx \begin{bmatrix} \mathbf{M}_{rr} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\theta\theta} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{F}_{Rigid\_t} \\ \mathbf{F}_{Rigid\_r} \end{bmatrix} \tag{3.33}$$

With the approximated multipliers of Equation (3.33), partitioned deformable body equation of motion can be rewritten as follows:

$$\mathbf{M}_{ff}\ddot{\mathbf{q}}_f + \mathbf{F}^{\mathbf{int}} = \mathbf{F}_f^{ext} + \mathbf{Q}_f - \mathbf{M}_{fr}(\ddot{\mathbf{R}} + \underline{\boldsymbol{\lambda}}_t) - \mathbf{M}_{f\theta}(\ddot{\boldsymbol{\theta}} + \underline{\boldsymbol{\lambda}}_r) \tag{3.34}$$

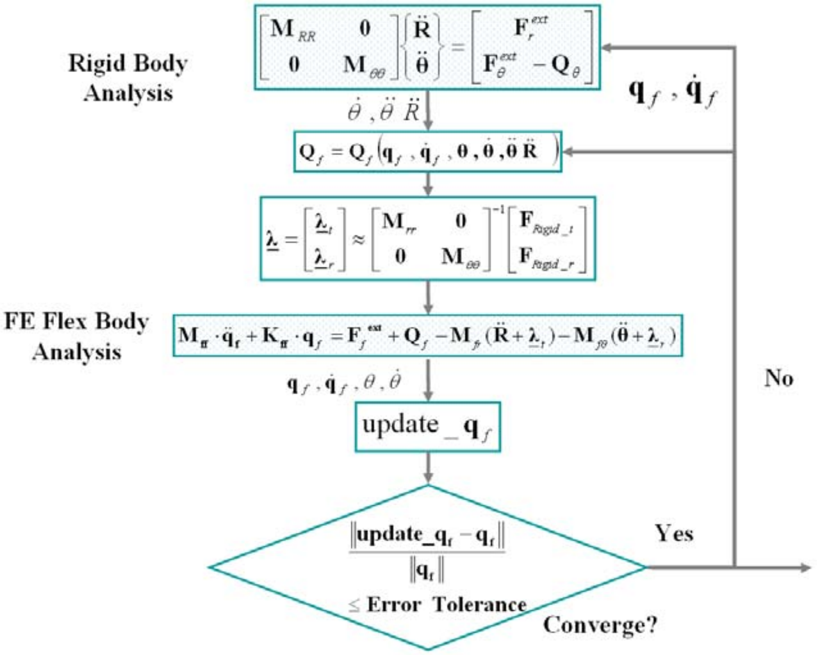

Rigid Body Analysis
$\begin{bmatrix} \mathbf{M}_{RR} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\theta\theta} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{R}} \\ \ddot{\boldsymbol{\theta}} \end{Bmatrix} = \begin{bmatrix} \mathbf{F}_r^{ext} \\ \mathbf{F}_\theta^{ext} - \mathbf{Q}_\theta \end{bmatrix}$
$\mathbf{q}_f , \dot{\mathbf{q}}_f$
$\dot{\theta} , \ddot{\theta} \ \ddot{R}$
$\mathbf{Q}_f = \mathbf{Q}_f \left(\mathbf{q}_f , \dot{\mathbf{q}}_f , \boldsymbol{\theta} , \dot{\boldsymbol{\theta}} , \ddot{\boldsymbol{\theta}} \ \ddot{\mathbf{R}} \right)$
$\underline{\boldsymbol{\lambda}} = \begin{bmatrix} \underline{\boldsymbol{\lambda}}_t \\ \underline{\boldsymbol{\lambda}}_r \end{bmatrix} \approx \begin{bmatrix} \mathbf{M}_{rr} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\theta\theta} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{F}_{Rigid\_t} \\ \mathbf{F}_{Rigid\_r} \end{bmatrix}$
FE Flex Body Analysis
$\mathbf{M}_{ff} \cdot \ddot{\mathbf{q}}_f + \mathbf{K}_{ff} \cdot \mathbf{q}_f = \mathbf{F}_f^{ext} + \mathbf{Q}_f - \mathbf{M}_{fr} (\ddot{\mathbf{R}} + \underline{\boldsymbol{\lambda}}_t) - \mathbf{M}_{f\theta} (\ddot{\boldsymbol{\theta}} + \underline{\boldsymbol{\lambda}}_r)$
$\mathbf{q}_f , \dot{\mathbf{q}}_f , \theta , \dot{\theta}$
No
update _ $\mathbf{q}_f$
$\frac{\| \text{update\_}\mathbf{q}_f - \mathbf{q}_f \|}{\| \mathbf{q}_f \|}$
$\leq$ Error Tolerance
Yes
Converge?

Figure 3.4 Procedure of Partitioned Iteration Method

As shown in Figure 3.4, the quadratic velocity terms are considered in rigid body analysis employing updated quadratic velocity terms from the flexible body analysis nodal variables. If the deformable body stiffness is sufficiently high, the quadratic velocity terms may be neglected. Note that due to mass redistribution, the CG position of the deformed body changes with time. Employing a floating frame that follows the CG position of each deformable body allows decoupling of the rigid body motion from the overall deformation, which is of immediate benefit for incorporation into the D-Sim.

3.2.2 Small Deformation Assumption in the Rigid Body Rotational Equation (CG Following Reference Condition)

In the previous section, it is mentioned that numerical instability comes from geometric nonlinearity which generate errors in the relative flexible body equation of motion. Consequently, when one solves Equation (3.26), the solution slightly violates the mean axis reference condition so that it may cause numerical instability. To remove this instability, the approximated multipliers method of inertial forces is suggested to remove rigid body motion in the relative deformable body equation. However, in this method, approximated multipliers should be calculated by the summation forces of each direction of $\mathbf{F}_{general}$. In this section, an alternative method is provided such that the relative deformable body solution satisfies the mean axis reference condition, in which the average relative deformation should be zero with respect to the body coordinate system.

In the case of Euler parameters for the definition of rotational position, one has

$$\bar{\boldsymbol{\omega}} = \bar{\mathbf{G}}\dot{\boldsymbol{\theta}}\,,\ \ \dot{\bar{\mathbf{G}}}\dot{\boldsymbol{\theta}} = 0 \qquad \text{and} \ \ \bar{\boldsymbol{\alpha}} = \bar{\mathbf{G}}\ddot{\boldsymbol{\theta}} \tag{3.35}$$

where $\bar{\boldsymbol{\omega}}, \bar{\boldsymbol{\alpha}}$ are the angular velocity vector and the angular acceleration vector defined in the body coordinate system, and $\bar{\mathbf{G}}$ is the matrix that relates the angular velocity to the time derivatives of orientation coordinates. For Euler parameters, $\bar{\mathbf{G}}\bar{\mathbf{G}}^T = 4$, $\bar{\mathbf{G}}\dot{\bar{\mathbf{G}}}^T = 2\tilde{\bar{\boldsymbol{\omega}}}$, one can rewrite rotational equations of motion and relative deformed body motion in Equation (3.26) including the mean axis reference conditions as follows,

$$\begin{bmatrix} \mathbf{M}_{ff}\ddot{\mathbf{R}} \\ \rho\sum_{i=1}^{N_e}\int_{\Omega_e}\bar{\mathbf{u}}_i\times\left(\bar{\boldsymbol{\alpha}}\times\bar{\mathbf{u}}_i+2\bar{\boldsymbol{\omega}}\times\dot{\bar{\mathbf{u}}}_i+\bar{\boldsymbol{\omega}}\times\bar{\boldsymbol{\omega}}\times\bar{\mathbf{u}}_i\right)d\Omega_e \end{bmatrix}=\begin{bmatrix} \mathbf{F} \\ \bar{\mathbf{u}}_p\times\mathbf{A}^T\mathbf{F} \end{bmatrix}$$

$$\begin{aligned} \mathbf{M}_{ff}\ddot{\mathbf{q}}_f+\mathbf{K}_{ff}\mathbf{q}_f &= \mathbf{F}_f^{ext}-\rho\sum_{i}^{N_e}\int_{\Omega_e}\underline{\mathbf{N}}_i^T\mathbf{A}^T\ddot{\mathbf{R}}d\Omega_e \\ &\quad -\rho\sum_{i}^{N_e}\int_{\Omega_e}\underline{\mathbf{N}}_i^T(\bar{\boldsymbol{\alpha}}\times\bar{\mathbf{u}}_i+2\bar{\boldsymbol{\omega}}\times\dot{\bar{\mathbf{u}}}_i+\bar{\boldsymbol{\omega}}\times\bar{\boldsymbol{\omega}}\times\bar{\mathbf{u}}_i)d\Omega_e \end{aligned} \tag{3.36}$$

In the relative deformable body motion in Equation (3.36), the nodal force acts on the deformable body with discrete form. This nodal force acts at the initial nodal position without considering body deformation. But, in the rotational equation for reference motion, the generalized force $\bar{\mathbf{u}}_p\times\mathbf{A}^T\mathbf{F}$ is the cross product of the deformed position vector and the external force described in the body coordinate system. Also, the solution of reference rotational motion, $\bar{\boldsymbol{\alpha}}$, $\bar{\boldsymbol{\omega}}$ are computed with including body deformation. In the relative deformed body part in Equation (3.36), the nodal force vector subtracted inertial force vector from reference equation of motion should have no rigid body force. One can know that force equilibrium can be satisfied in relative deformed body equation if the position vector of generalized force acting on the reference body is the same with initial position vector with small deformation assumption in the rotational reference equation of motion as follows

$$\bar{\mathbf{u}}_i=\bar{\mathbf{u}}_{i0}+\delta\bar{\mathbf{u}}_i\approx\bar{\mathbf{u}}_{i0} \tag{3.37}$$

$$\rho\sum_{i=1}^{N_e}\int_{\Omega_e}\bar{\mathbf{u}}_{i0}\times\left(\bar{\boldsymbol{\alpha}}\times\bar{\mathbf{u}}_i+2\bar{\boldsymbol{\omega}}\times\dot{\bar{\mathbf{u}}}_i+\bar{\boldsymbol{\omega}}\times\bar{\boldsymbol{\omega}}\times\bar{\mathbf{u}}_i\right)d\Omega_e=\bar{\mathbf{u}}_{p0}\times\mathbf{F} \tag{3.38}$$

Employing this small deformation assumption in the rotational body reference equation, we can get a stable solution without additional calculation of approximated multipliers in Equation (3.33).

The constraint force in the mean axis reference condition redistributes the mass center and body coordinates axis to the point satisfying the reference conditions so that deformable body solutions should be shifted to the same magnitude of difference of reference coordinates and redistributed axis and center of mass, while the reference coordinates in the CG following reference conditions are updated to the point of redistributed axis and mass center as shown in Figure 3.5.

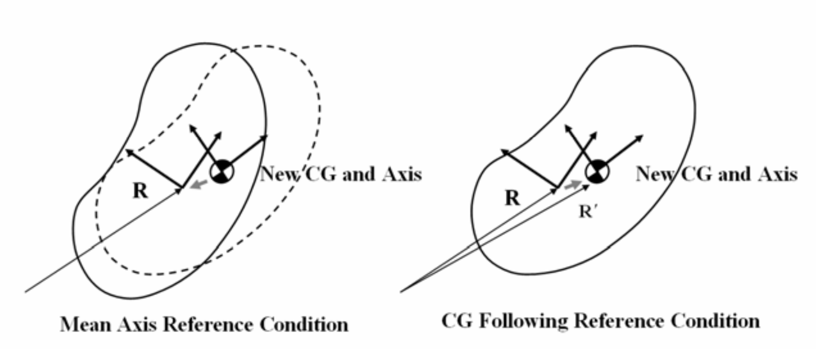

R
New CG and Axis
Mean Axis Reference Condition
R
R′
New CG and Axis
CG Following Reference Condition

Figure 3.5 The difference of mean axis and CG following reference conditions

3.3 EXAMPLES

3.3.1 Evaluation of Partitioned Iteration Method: 2D Beam Problem

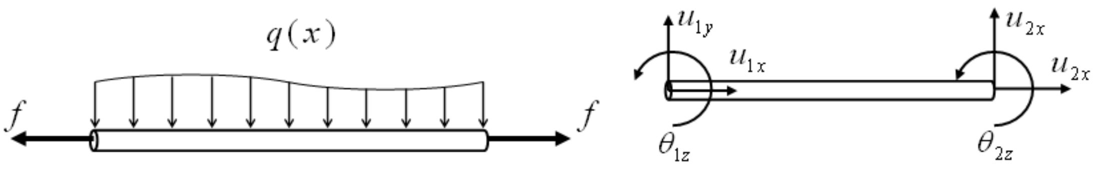


Figure 3.6 2-Dimensional beam element

For validation of PIM, a simple planar frame model is considered, as depicted in Figure 3.6 The axial deformation of each frame element is described by linear polynomial as,

$$\rho A \frac{d^2 w_x}{dt^2} + \frac{d}{dx}\left(EA \frac{dw_x}{dx} \right) = f \tag{3.39}$$

while the transverse deformation is described using Euler-Bernoulli beam theory and Hermite cubic polynomial interpolations.

$$\rho A \frac{d^2 w_y}{dt^2} + \frac{d^2}{dx^2}\left(EI \frac{d^2 w_y}{dx^2} \right) = q \tag{3.40}$$

For this study the axial and transverse equations are uncoupled. The displacement field within the element is given by the following polynomials

$$\begin{aligned} w_x &= a_0 + a_1 \xi \\ w_y &= b_0 + b_1 \xi + b_2 \xi^2 + b_3 \xi^3 \end{aligned} \tag{4.27}$$

The number of coordinates for this element is 6, and the nodal displacement vector for the ith element can be written as

$$\mathbf{q}_{fi} = \begin{bmatrix} q_{1i} & q_{2i} & q_{3i} & q_{4i} & q_{5i} & q_{6i} \end{bmatrix}^T \tag{4.28}$$

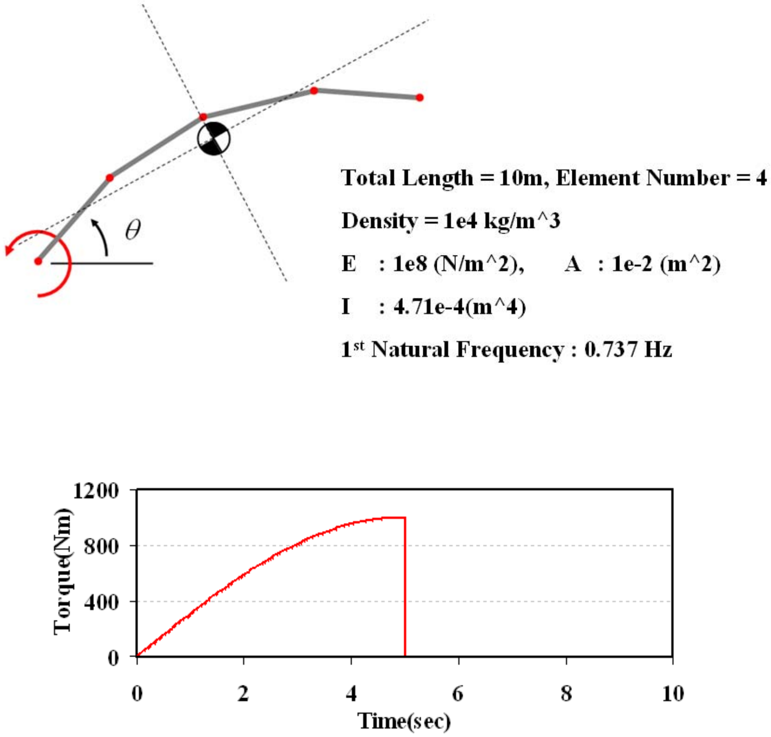
The exact location of an arbitrary point on the *i-th* element can be obtained using the element shape function and the vector of nodal coordinate as

$$\mathbf{u}_i = \mathbf{N}\left(\mathbf{u}_{i0} + \mathbf{q}_{fi}\right) \tag{4.29}$$

where, $\mathbf{u}_{i0}$ is undeformed body nodal coordinate of *i-th* element and $\mathbf{N}$ is the element shape function matrix given by

$$\mathbf{N} = \begin{bmatrix} \frac{l_e - \xi}{l_e} & 0 & 0 & \frac{\xi}{l_e} & 0 & 0 \\ 0 & 1 - 3\frac{\xi^2}{l_e^2} + 2\frac{\xi^3}{l_e^3} & 1 - 2\frac{\xi^2}{l_e} + \frac{\xi^3}{l_e^2} & 0 & 3\frac{\xi^2}{l_e^2} - 2\frac{\xi^3}{l_e^3} & 2\frac{\xi^2}{l_e} - \frac{\xi^3}{l_e^2} \end{bmatrix} \tag{4.30}$$

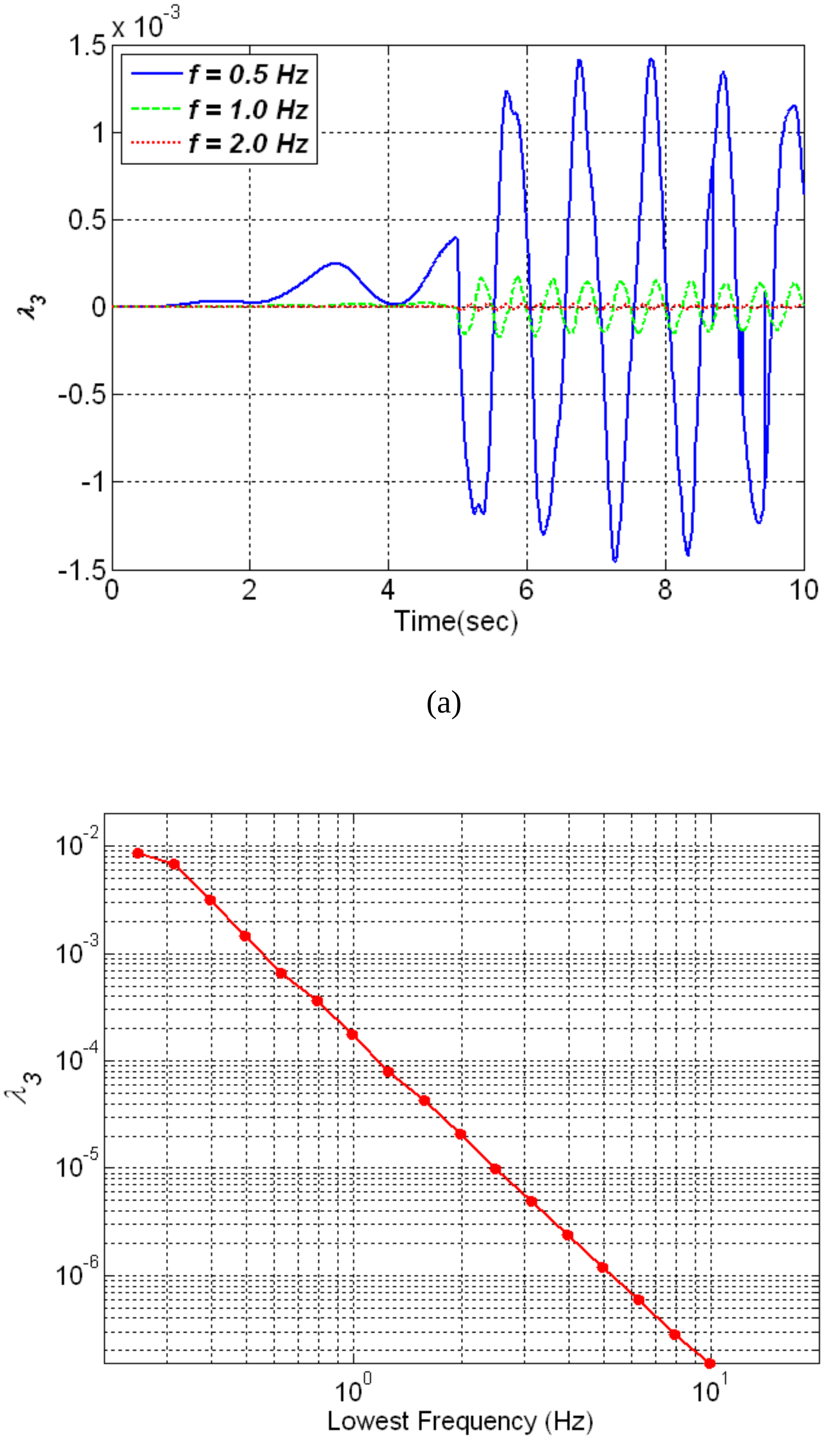
For this example, the loading condition applied is a sinusoidal torque at the end of bar, with time history as shown in Figure 3.7. The simulation results provided are the solutions from the partitioned iteration method and floating frame of reference method using the mean axis reference condition. The linear elastic FE model is solved using the well-known trapezoidal integration rule, while the rigid body equations are solved using a general ODE solver suite (ode45, ode15s in MATLAB).


θ
Total Length = 10m, Element Number = 4
Density = 1e4 kg/m^3
E : 1e8 (N/m^2), A : 1e-2 (m^2)
I : 4.71e-4(m^4)
1st Natural Frequency : 0.737 Hz
1200
800
400
0
Torque(Nm)
0
2
4
6
8
10
Time(sec)

Loading Torque profile versus time

Figure 3.7 Flexible body model with finite beam elements and loading condition

In the floating frame of reference formulation, Lagrange multipliers contribute to the solution of deformable body so also satisfy mean axis reference condition, i.e., the errors generated by the assumption of small deformation or linear elastic behavior in deformable body are corrected by these constraint forces. Therefore, if body deformation is small enough to be considered as linear elastic behavior, the constraint forces should be close to zero. As shown Figure 3.8-(a), the Lagrange multipliers decrease as the body stiffness increases. Also, the relationship of maximum peak values of Lagrange multipliers and the lowest frequency of deformable body can be seen in Figure 3.8-(b).



(a)

(b)

Figure 3.8 Rotational Lagrange Multiplier according to changing dynamic stiffness

The multiplier for satisfaction of the mean axis reference condition (MARC) in the floating frame of reference is compared with the approximated multiplier of PIM in Figure 3.9. The partitioned deformable body equation of motion in equation (3.36) can be solved using a linear elastic simulation solver without instability arising from the accumulation of numerical error during forward integration. Figure 3.10 show the difference of angular velocities of reference coordinates of mean axis reference condition and CG following reference condition. The reference of PIM with Approximated multipliers is in excellent agreement with the MARC reference coordinates. As seen in Figure 3.5, the difference between the CG following and Mean axis reference condition results in a different reference motion. Figure 3.10 shows the difference of the angular velocity of these reference conditions.

The angular speeds and vertical displacement of the beam end point are compared in Figure 3.11 and 3.12. It can be seen that the results are in agreement with the floating frame of reference results.

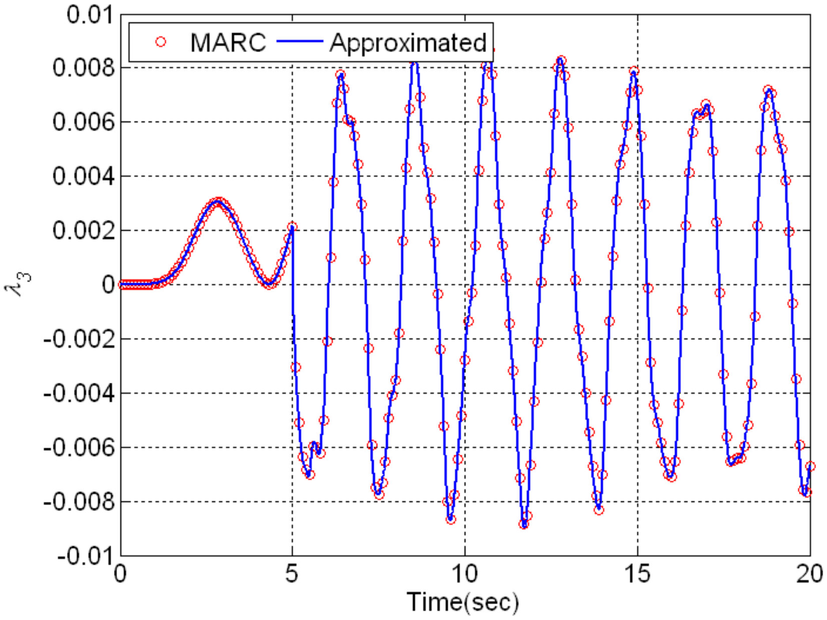


Figure 3.9 Results of rotational Lagrange Multiplier

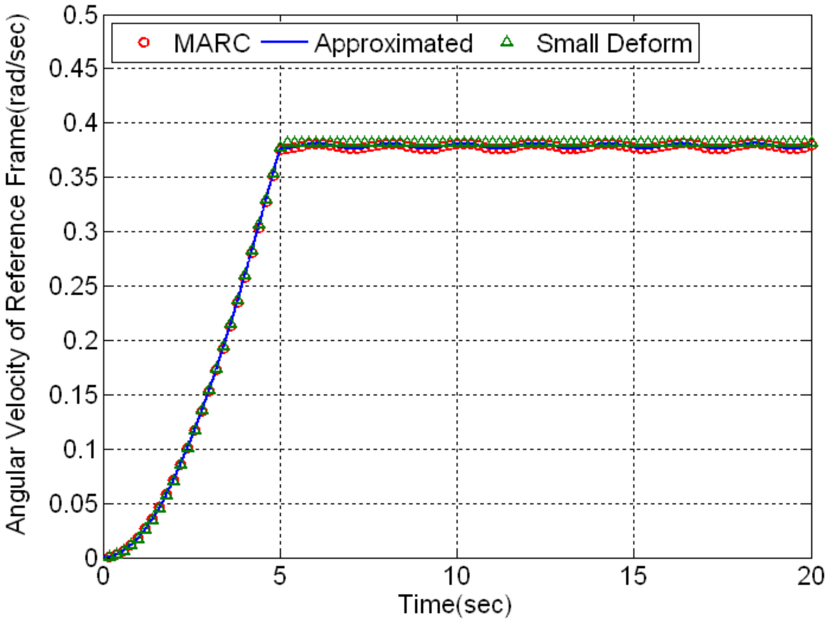


Figure 3.10 Angular velocities of reference frame

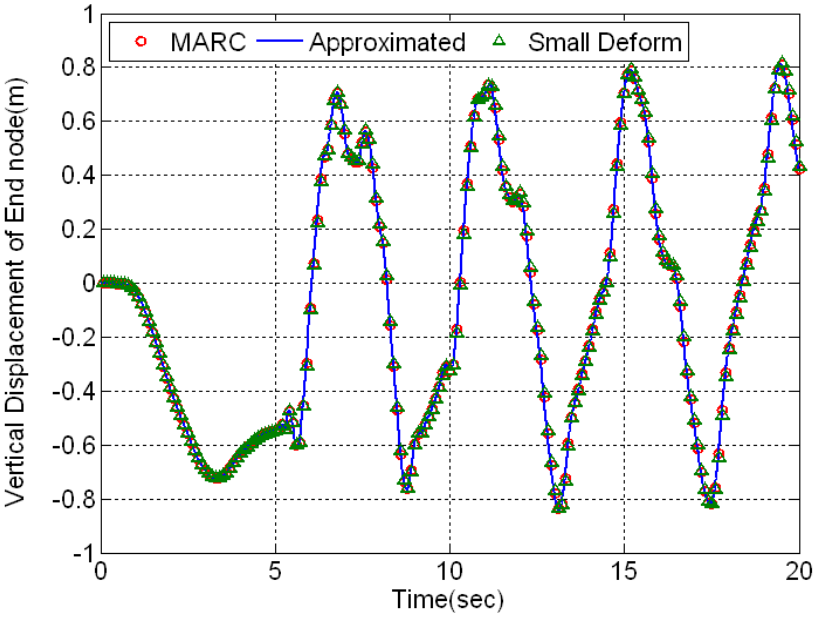


Figure 3.11 Relative motion comparison: vertical displacement of beam end node

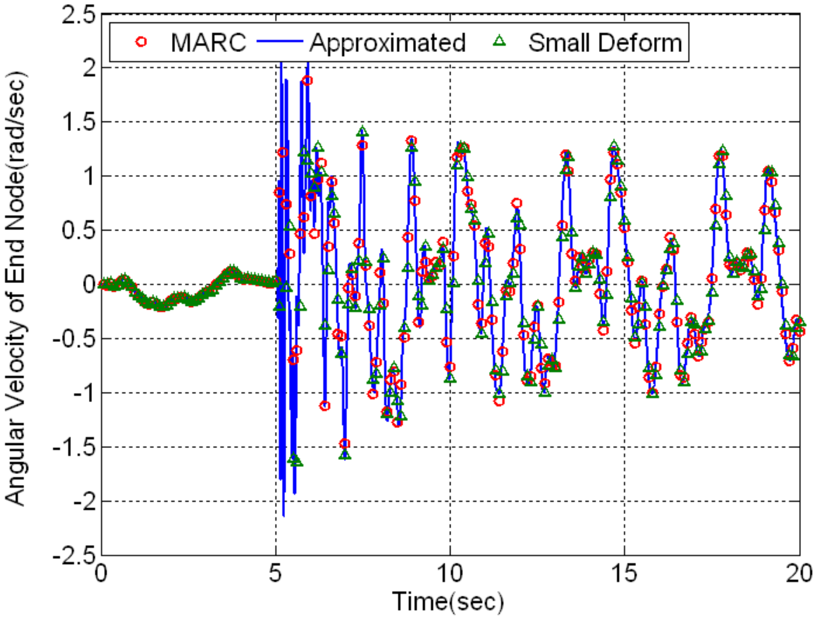


Figure 3.12 Relative motion comparison: angular velocities of end node

Of prime importance is specifying an appropriate error tolerance for iteration convergence. The simulation result is deemed satisfactory to that of floating reference frame if the error tolerance is sufficiently small. However, the number of iterations increases with smaller error tolerances, which is directly related to the simulation cost. Figures 3.13 and 3.14 show respectively the average iteration count and Converged Value Error as a function of error tolerance. The Converged Value Error is defined as

$$\text{Converged Value Error: } \mathbf{CVE} = \sum_{n=1}^{N_s} \frac{\left\| \mathbf{u}_\mathbf{e}^\mathbf{n} - \mathbf{u}^\mathbf{n} \right\|}{\left\| \mathbf{u}^\mathbf{n} \right\|} \cdot \mathit{\Delta t} \tag{3.41}$$

where, n is time step, N_s total time step number, $\mathbf{u}_\mathbf{e}^\mathbf{n}$ is deformation vector with error tolerance e and $\mathbf{u}^\mathbf{n}$ is converged deformation vector with error tolerance 10^{-20}. Figure 3.14 shows that for the range of error tolerance values from 10^{-8} to 10^{-18}, the average iteration number changes little. Also, converged value error is closed to zero for the range of error tolerance, which is smaller than10^{-8}. The adequate error tolerance can be between 10^{-8} and 10^{-10} for the simulation accuracy and cost as shown in Figure 3.13 and 3.14.

Figure 3.15 shows the average iteration number in accordance with the changing of a maximum norm of body deformation defined as

$$\frac{\max\left(\int_\Omega \left\| \mathbf{u}_n \right\| d\Omega\right)}{L} = \frac{\max\left(\sqrt{\int_\Omega \mathbf{u}_n^T \mathbf{u}_n d\Omega}\right)}{L} = \frac{\max\left(\sqrt{\sum_i^{N_e} \int_{\Omega_e} \mathbf{q}_{\mathbf{f}n}^T \underline{\mathbf{N}}_i^T \underline{\mathbf{N}}_i \mathbf{q}_{\mathbf{f}n} d\Omega_e}\right)}{L} \tag{3.42}$$

where L is total length of beam model. Note that the average iteration number reduces to one when the norm of body deformation reaches 10^{-8}, because the coupling forces is function of the magnitude of body deformation. Above this value, the problem is effectively that of linear elasto-dynamics in which the body deformation has no effect on the reference motion. Figure 3.16 shows the iteration number comparison of ABAQUS non-linear and PIM results during 1000 simulation steps. As shown Figure 3.15, PIM needs 2350 iterations which is a smaller number of iterations than the 3008 iterations used by ABAQUS. Especially, in ABAQUS non-linear simulation, the stiffness matrix needs to be updated for the expressing with geometric nonlinearities with respect to the changing of body position at every time step,

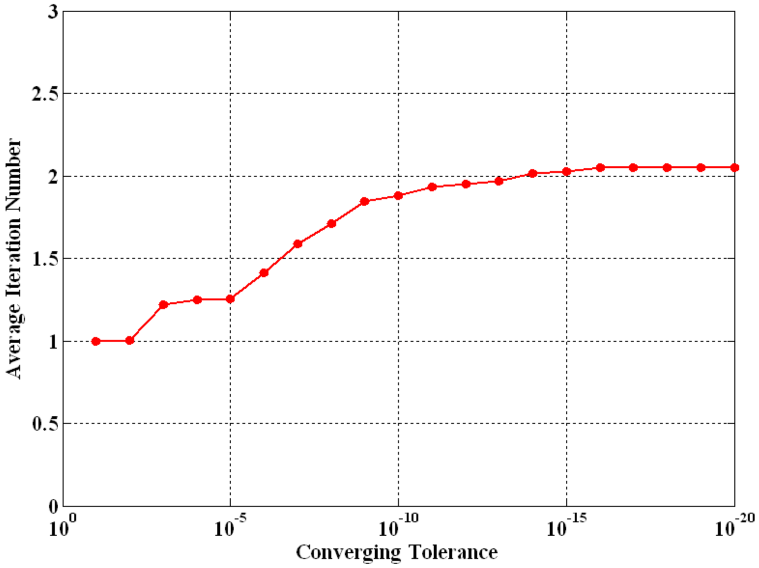


Figure 3.13 Average iteration number with respect to error tolerance

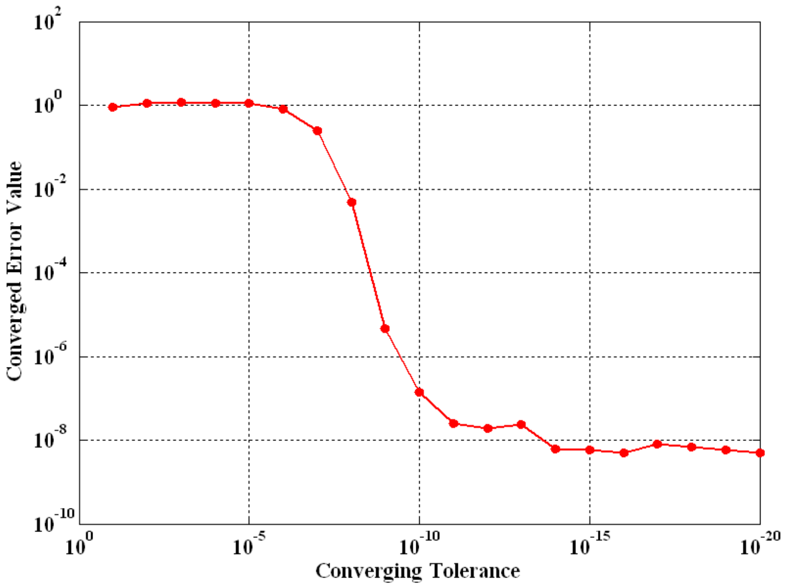


Figure 3.14 Converged value error with respect to error tolerance

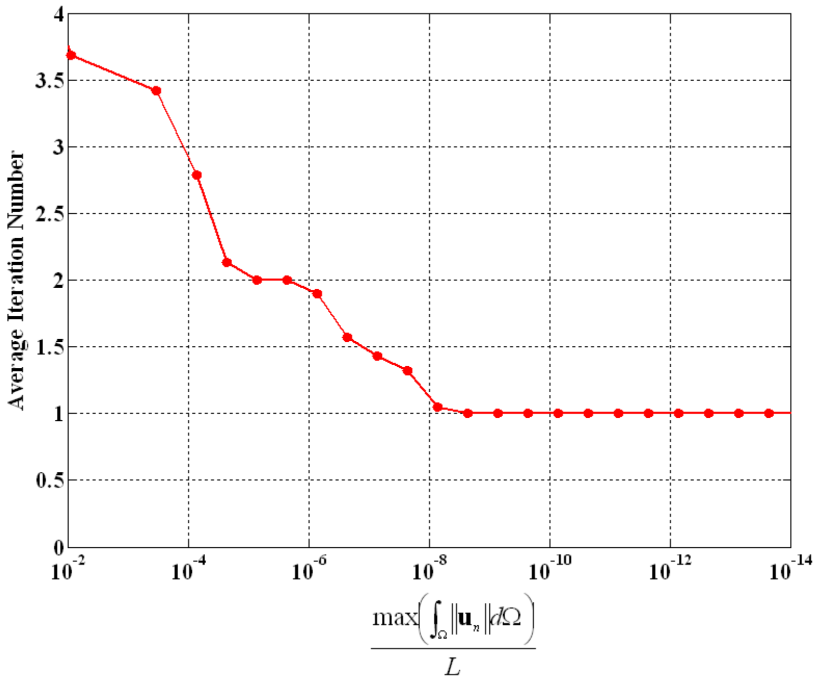


Figure 3.15 Average iteration number with respect to the norm of deformation

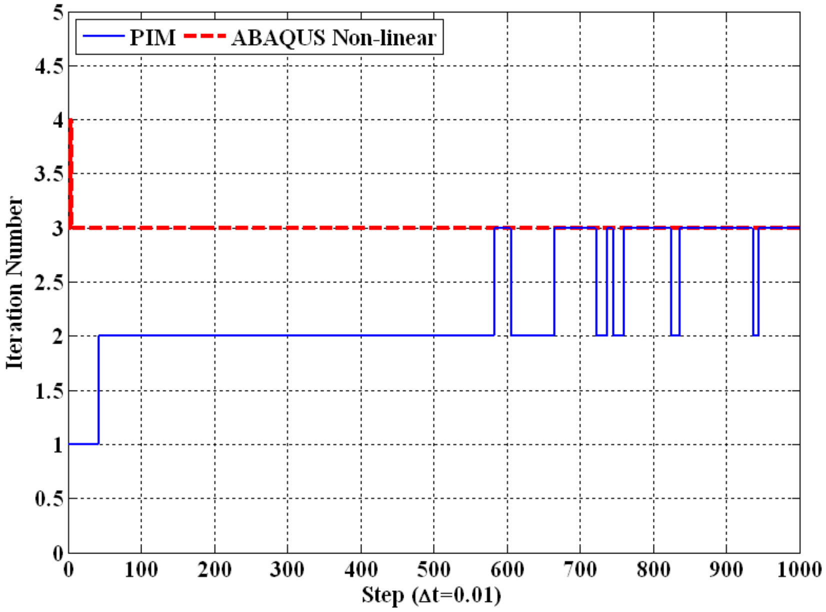


Figure 3.16 Iteration number comparison of ABAQUS Non-linear and PIM

3.3.2 Plate Model with 3D Shell Elements

A three dimensional shell element model having a center slot, depicted in Figure 3.17, is solved using a partitioned iteration method and compared with that from the commercial multibody dynamics code ADAMS/FLEX. The plate uses the elastic shell element of FEAP. The model comprises 1196 nodes and 1116 elements. The coupling force is applied at the corner as depicted in Figure 3.16, which engenders a complex motion having three directional rotations in space. ADAMS/FLEX employs modes-based reduced order modeling for flexible components. These reduced order models are embedded in DAEs and solved with other rigid body components. This plate example shows the difference of flexible body results and the result of reduced order finite element model in ADAMS. The tolerance for the compatibility condition is 10^{-10} and time increment is selected as 0.001 second. Figure 3.18 shows the global body motion of reference frame which is attached to center of mass of plate elements. It is seen to be in a good agreement with the ADAMS/FLEX results.

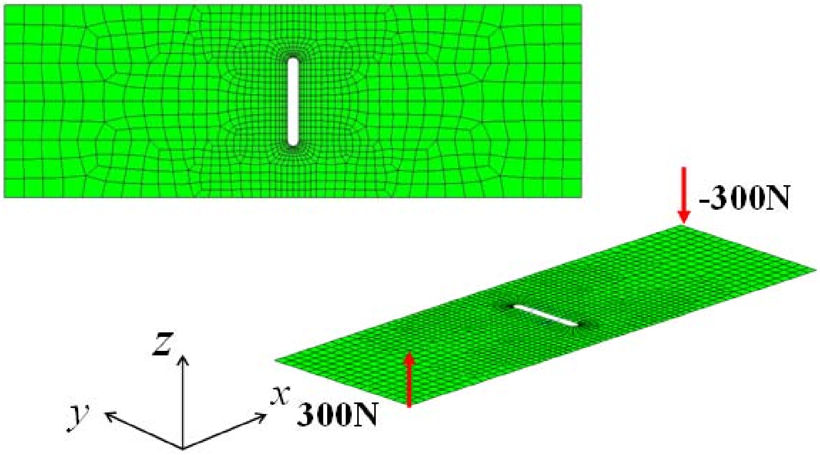


Figure 3.17 Plate model with 3D shell elements and loading condition

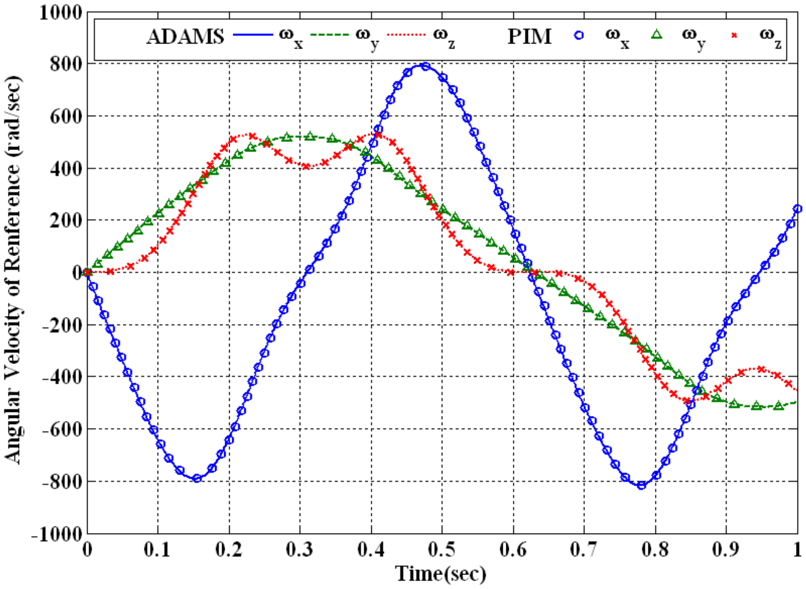


Figure 3.18 Angular velocities of reference frame in global coordinates

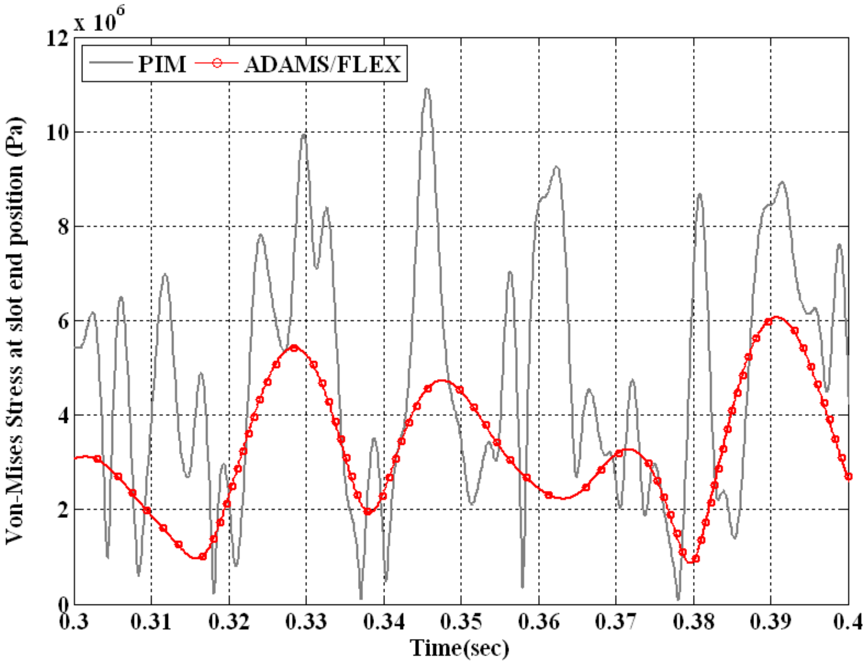


Figure 3.19 Von-Mises stress at the slot end position element (including 15 modes)

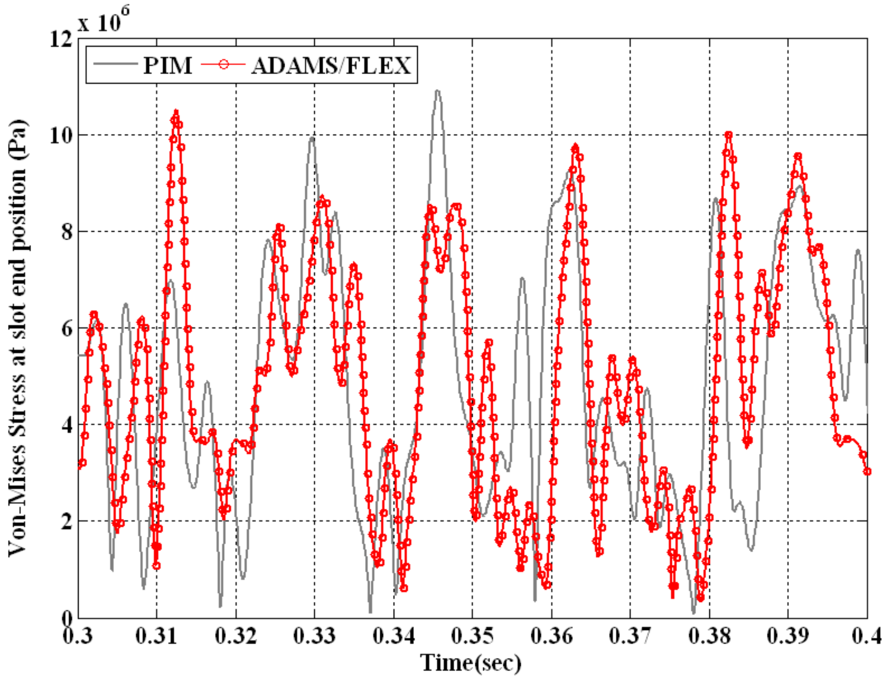


Figure 3.20 Von-Mises stress at the slot end position element (including 100 modes)

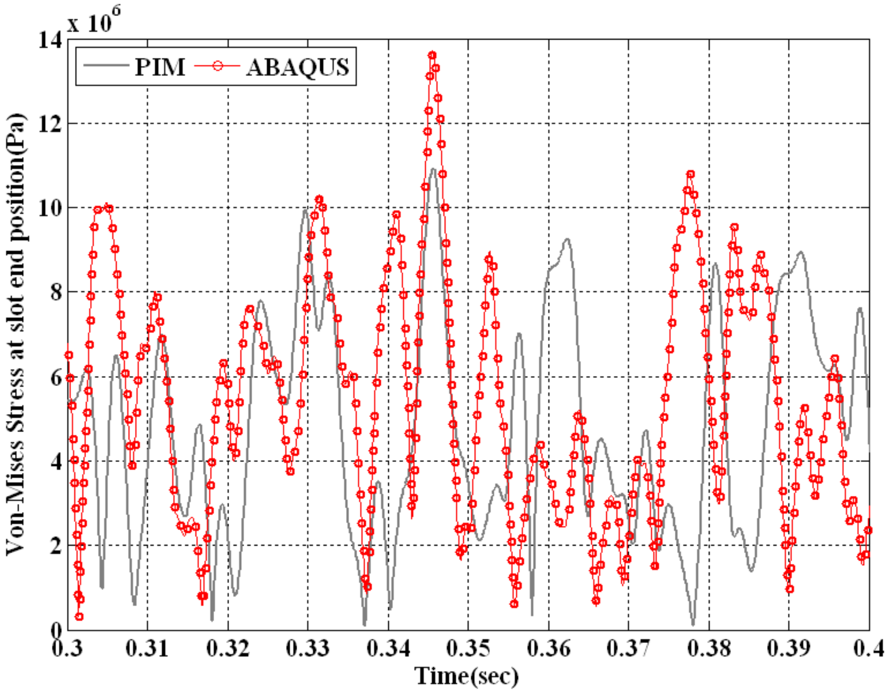


Figure 3.21 Von-Mises stress comparison at the slot end position element

The flexible body in ADAMS/FLEX can be constructed with modal based data imported using a special formatted file having file extension MNF (modal neutral file). This type of modal data has mode shapes, nodal point stress and natural frequencies including constraint modes and can be exported by commercial finite element analysis software such as NASTRAN, and/or ABAQUS. The modal data of the plate model was exported from NASTRAN normal mode analysis (solver 103). Figures 3.19 and 3.20 illustrate and zoom in the Von-Mises stress time histories around a corner of slot area predicted using PIM and ADAM/FLEX between 0.3 and 0.4 seconds, in which the flexible body is solved with reduced order model including respectively 15 normal modes and 100 normal modes. It is seen that the ADAMS/FLEX stress result including 100 modal data is closer to PIM result, which is obtained by FEAP with linear elastic finite element solver. Note that the results obtained using the partitioned iteration method is different from the ADAMS results. The difference of shell element formulation in NASTRAN and FEAP can make a difference of time responses and stress magnitude between two simulation codes. Also, the difference of time histories of Von-Mises stress can be seen in Figure 3.21, the comparison of ABAQUS and PIM simulation results which include full nodal degree of freedom to solve dynamic response. As the results in Figure 3.20 and 3.21, one can know that the differences of element formulation make different simulation results in stress time histories.

Figure 3.22 depicts the modified plate model reinforced with plate and bolt components. The result depicted in Figure 3.23 of this example also shows the difference of stress fields in PIM and ADAMS results. In order to express more exact stress concentration results at the area of bolt and reinforce plate, ADAMS/FLEX has to import modal data having high frequency mode, it means that additional normal modes analysis and reloading flexible body model for updating modal data. The PIM can be a useful analysis method and allow more accurate solutions for complex model having large rotational movement.

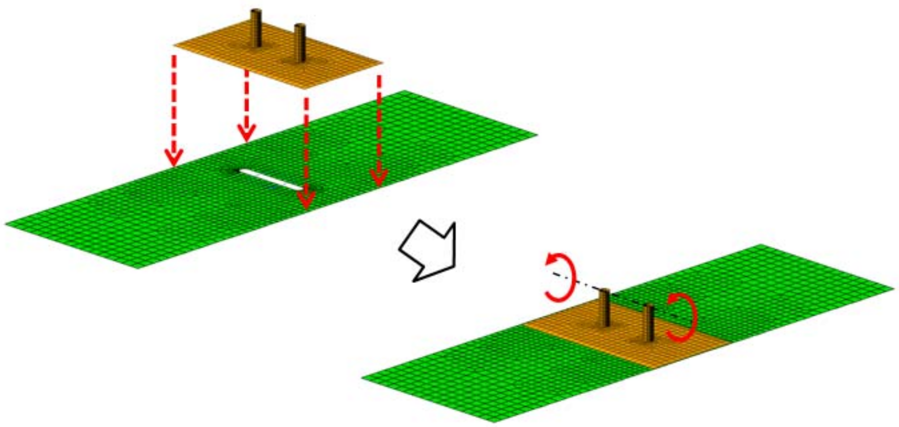
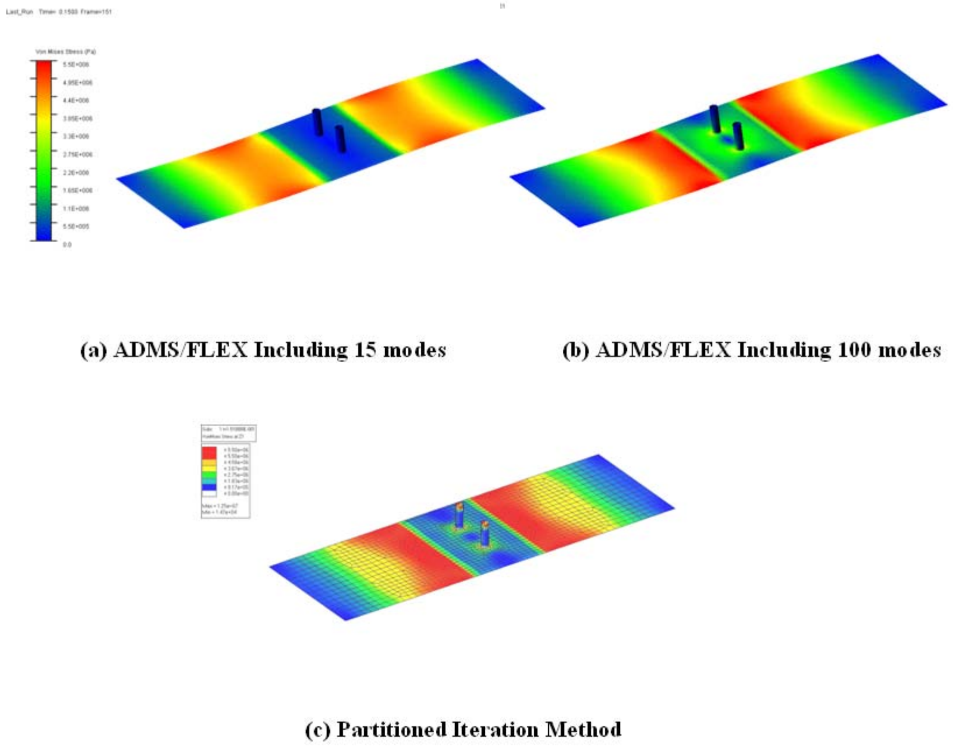


Figure 3.22 Reinforced plate model with throughout bolts



(a) ADMS/FLEX Including 15 modes

(b) ADMS/FLEX Including 100 modes

(c) Partitioned Iteration Method

Figure 3.23 Stress concentration differences of ADAMS/FLEX result and PIM at bolt and attached reinforced plate area

CHAPTER 4

INTEGRATION FEM AND MBD CODES USING GLUING ALGORITHM

The distributed simulation platform (D-Sim) was developed to cater to the distributed nature of collaborative design inside the supply chain of modern product development, and to significantly accelerate the product development process. It provides great potential in laying out innovative and new products in a cost-effective and timely manner. One of the challenges in developing such distributed simulation systems, is to integrate heterogeneous subsystem models, such as multibody dynamics subsystems models and finite element subsystems models for practical engineering problems. In this chapter, a practical distributed simulation environment enables using existing commercial packages, including a combination of multibody dynamics codes, such as MSC/ADAMS, and finite element codes without modifying their solvers and user interfaces. Rigid body motion of the overall system (or a subsystem) should be solved with the numerical integrators in a multibody dynamics code, while deformation of each individual component should be solved with the solvers in an existing finite element code. The flexible part of FEM is simulated using PIM and then coupled with a simple MBD model, which is simulated with a different computation processor in distributed simulation. The simulation results are combined using gluing algorithm.

4.1 GLUING ALGORITHM

In the case of coupling MBD and FEM, it is difficult to satisfy all kinematic variables simultaneously at the interface, since each set of simulation results at the interface may be computed with different simulation schemes; for example, multibody dynamic system to solve the nonlinear differential and algebraic equations (DAEs) and classical finite element formulations, connected with modeling the inertial and stiffness of structural systems, employ infinitesimal rotations and displacements as nodal coordinates.

One approach to code integration is to undertake cosimulation, in which the finite element program and the multibody dynamics program run concurrently, but separately. The need exists to exchange appropriate information, for example, kinematic and force data, between the two codes. In addition, the execution of the codes must be coordinated so that their combined solutions are identical or comparable to solutions that would be obtained from "all-at-once" approaches. The foundation of the gluing approach arises from the already disparate simulation models and programs in use within the industry. Rather than attempting to partition an existing model, the goal is to glue together an already partitioned model that may contain many different finite element and multibody dynamics models. For example, a detailed vehicle model that includes tires, leaf springs, suspension and steering components and joints, chassis frame, etc., can be developed.

Different Gluing Algorithms ensue, depending on which group of interface quantities is considered as the defined input. Detailed descriptions of different Gluing Algorithms are given in Wang 2005 [13] . In general, Gluing Algorithms can be classified into three groups: *T*-*T*, *X*-*T*, and *X*-*X*, where *X* denotes kinematic quantities and *T* denotes force quantities. In the T-T algorithm, each system provides kinematic quantities to the gluing coordination module. The coordinator then returns updated force quantities to the distributed systems; the systems update their dynamic states using these new force estimates. This process is iterated until convergence is reached. This algorithm provides the least intrusive change to existing finite element and multibody dynamics codes as the *T-T* algorithm has the structure of a general force element for most codes. The *X-X* Gluing Algorithm is the inverse of the *T-T* approach, whereby force quantities are provided to the gluing coordination module from each system. The coordinator then provides updated kinematic states to the systems. This approach may be viewed as a time-dependent prescribed kinematic boundary condition to be specified for each system.

4.1.1 Formulation of Gluing Algorithm (T-T Method)

In general, for the T-T Gluing Algorithm, if a proper set of interface force variables is defined such that the equilibrium conditions are satisfied, then only the compatibility conditions need to be considered during the iteration process. In this case, the interface force variables can be considered as functions of the interface kinematic quantities, and these interface force variables can be updated using the kinematic information and compatibility conditions. In this session, T-T gluing strategies are summarized.

In the general case of a distributed system, let us consider that the equations of motion of each distributed subsystem in a dynamic finite element method can be written each as:

$$\mathbf{M}^i\ddot{\mathbf{u}}^i + \mathbf{C}^i\dot{\mathbf{u}} +^i \mathbf{K}^i\mathbf{u}^i = \mathbf{f}^i, (i = 1, 2, \cdots, n) \tag{4.1}$$

in which $\mathbf{M}^i$, $\mathbf{C}^i$ and $\mathbf{K}^i$ are the respective mass, damping, and stiffness matrices, $\mathbf{f}^i$ denotes the external forces, and $\mathbf{u}^i$, $\mathbf{v}^i$ and $\mathbf{a}^i$ are the nodal displacement, velocity and acceleration vectors, respectively. In multibody dynamics, equation of motion can be written as a Differential Algebraic Equations (DAEs)

$$\begin{cases} \mathbf{M}_R^i \mathbf{a}^i + \mathbf{\Phi}_q^{i\,T} \boldsymbol{\lambda}^i = \mathbf{Q}^i \\ \mathbf{\Phi}^i = \mathbf{0} \end{cases} \tag{4.2}$$

where, $\mathbf{M}_R^i$ denotes the inertia matrix, $\mathbf{a}^i$ denotes the acceleration vector, $\mathbf{Q}^i$ denotes the generalized force vector, $\boldsymbol{\lambda}^i$ is the vector of Lagrange multipliers, $\mathbf{\Phi}^i$ represents the internal constraints, which are assumed to be holonomic.

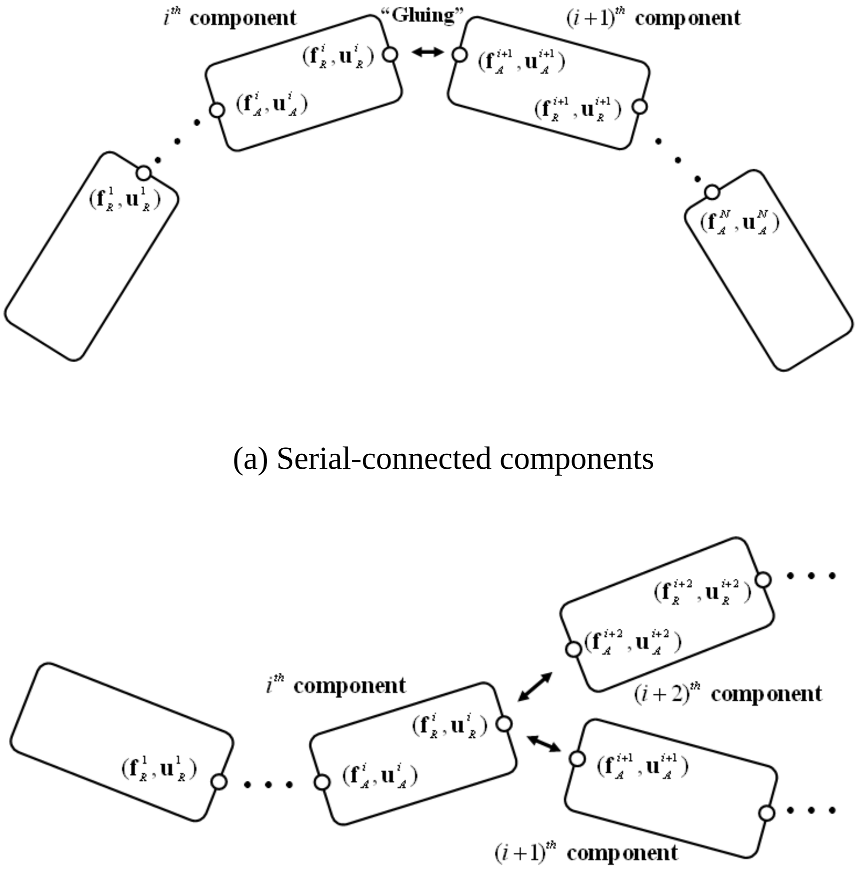
A Gluing Algorithm, named the T-T method, depends on only the interface information of subsystem models. Consequently, it can enable plug-and-play of the simulation models in a distributed simulation environment. In the general case, the *i-th* subsystem equation of motion Equation (4.1) and (4.2) can be written by interface forces and kinetic information as

$$\mathbf{\Psi}^i(\ddot{\mathbf{u}}_c^i, \dot{\mathbf{u}}_c^i, \mathbf{u}_c^i) = \mathbf{f}_c^i \quad (i = 1, 2, \cdots, n) \tag{4.3}$$

where, $\ddot{\mathbf{u}}_c^i$, $\dot{\mathbf{u}}_c^i$, $\mathbf{u}_c^i$ denote accelerations, velocities and displacements of the interface nodes or points, $\mathbf{f}_c^i$ denotes action-reaction forces at the interface, and n is the total number of subsystems to be integrated. The problem here is to couple the subsystems' equations in Equation (4.3) so that the equilibrium and compatibility conditions at the subsystem interfaces can be satisfied while individual equation (4.3) are solved independently. Figure 4.1 illustrates a general connection of subsystems, which can have serial type connections and multiple type connections. In these two types of connections, compatibility and equilibrium conditions can be different with each other.

In the general case, the subsystem interface force vector $\mathbf{f}_c^i$ in Equation (4.3) can be represented by a subset of the variables in $\mathbf{F}$, and therefore $\mathbf{f}_c^i$ can be written as

$$\mathbf{f}_c^i = \mathbf{f}_c^i(\mathbf{F}), (i = 1, 2, \cdots, n) \tag{4.4}$$


i^{th} component
"Gluing"
$(i+1)^{th}$ component
$(\mathbf{f}_R^i, \mathbf{u}_R^i)$
$(\mathbf{f}_A^{i+1}, \mathbf{u}_A^{i+1})$
$(\mathbf{f}_A^i, \mathbf{u}_A^i)$
$(\mathbf{f}_R^{i+1}, \mathbf{u}_R^{i+1})$
$(\mathbf{f}_R^1, \mathbf{u}_R^1)$
$(\mathbf{f}_A^N, \mathbf{u}_A^N)$
$(\mathbf{f}_R^{i+2}, \mathbf{u}_R^{i+2})$
$(\mathbf{f}_A^{i+2}, \mathbf{u}_A^{i+2})$
i^{th} component
$(i+2)^{th}$ component
$(\mathbf{f}_R^i, \mathbf{u}_R^i)$
$(\mathbf{f}_A^{i+1}, \mathbf{u}_A^{i+1})$
$(\mathbf{f}_R^1, \mathbf{u}_R^1)$
$(\mathbf{f}_A^i, \mathbf{u}_A^i)$
$(i+1)^{th}$ component

(a) Serial-connected components

(b) Multiple-connected components

Figure 4.1 Gluing types of components

Assume $\mathbf{U}$ is an assembly of the interface displacements of all subsystems, namely,

$$\mathbf{U} = \left\{ \mathbf{u}_c^i \right\}^T = \begin{Bmatrix} \mathbf{u}_c^1 \\ \vdots \\ \mathbf{u}_c^n \end{Bmatrix} \tag{4.5}$$

Then, an error measure $\mathbf{e}$ can be written as the function of interface kinetic quantity $\mathbf{U}$

$$\mathbf{e} = \mathbf{e}(\mathbf{U}) \tag{4.6}$$

In the general case, if the number of subsystems is *n*, the number of interfaces is *n-1*. Then, we can define the interface force vector

$$\mathbf{F} = \left\{ \left(\mathbf{f}_A^2 \right)^T, \left(\mathbf{f}_A^3 \right)^T, \cdots, \left(\mathbf{f}_A^n \right)^T \right\}^T \tag{4.7}$$

where $\mathbf{f}_A^i$ is the subset of independent force variables selected from $\mathbf{f}_c^i$, and thus

$$\mathbf{f}_c^i = \begin{Bmatrix} \mathbf{f}_A^i \\ \mathbf{f}_R^i \end{Bmatrix} \tag{4.8}$$

where $\mathbf{f}_R^i$ contains the remaining force components in $\mathbf{f}_c^i$. For convenience, we call $\mathbf{f}_A^i$ “active” or “action” forces and $\mathbf{f}_R^i$ “passive” or “reaction” forces. Passive forces are in general determined by the active forces of the adjacent subsystems. $\mathbf{F}$ can represent the force space at the interfaces considered and $\mathbf{F}$ is self-balanced, i.e., the equilibrium conditions at the interfaces are automatically satisfied For example, for a system with the serial type connections shown in Figure 3.1, we have

$$\mathbf{f}_R^i = -\mathbf{f}_A^{i+1}, \left(i = 1, 2, \cdots, n-1 \right) \tag{4.9}$$

and the equilibrium conditions of multiple type connections can be

$$\begin{gathered}\mathbf{f}_R^i = -\mathbf{f}_A^{i+1} - \mathbf{f}_A^{i+2} \cdots - \mathbf{f}_A^{i+m} \\ \left(\text{m : multiple connected subsystem number with the i-th subsystem}\right)\end{gathered} \tag{4.10}$$

Let $\mathbf{e}$ be an error measure vector that represents the violation of the compatibility conditions at the interfaces, where $\mathbf{e} = \mathbf{0}$ indicates that the compatibility conditions are fully satisfied. In the general case, $\mathbf{e}$ can be considered as

$$\mathbf{e} = \begin{Bmatrix} \mathbf{u}_A^2 - \mathbf{u}_R^1 \\ \vdots \\ \mathbf{u}_A^{i+1} - \mathbf{u}_R^i \\ \vdots \\ \mathbf{u}_A^n - \mathbf{u}_R^{n-1} \end{Bmatrix} \}\, i-th\ interface \tag{4.11}$$

In multiple type connections, $\mathbf{e}$ can be

$$\mathbf{e} = \begin{Bmatrix} \vdots \\ \mathbf{u}_A^{i+1} - \mathbf{u}_R^i \\ \mathbf{u}_A^{i+2} - \mathbf{u}_R^i \\ \vdots \\ \mathbf{u}_A^{i+m} - \mathbf{u}_R^i \\ \vdots \end{Bmatrix} \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} i-th\ interface \tag{4.12}$$

In T-T Gluing Method, if a proper set of interface force variables is defined such that the equilibrium conditions are satisfied, then only the compatibility conditions need to be considered during the simulation process. In this case, the interface force variables can be considered as functions of the interface kinematic quantities, and these interface force variables can be updated using the kinematic information and compatibility conditions. The target of the T-T Gluing Method is to finding interface forces, which satisfy compatibility conditions. Therefore, for the sake of finding interface forces satisfying the compatibility conditions, the Gluing Algorithm employs a Newton-

Raphson iteration method [58]. For the general solution of distributed simulation using Newton-Raphson method, the compatibility condition can be set as residue **Re,** where

$$\mathbf{Re}\left(\mathbf{F}\right) = \mathbf{e}(\mathbf{F}) = 0 \tag{4.13}$$

or

$$\mathbf{Re}\left(\mathbf{U}\right) = \mathbf{e}(\mathbf{U}) = 0 \tag{4.14}$$

The general problem is therefore formulated in terms of the discretized parameter as solution at the *n+1* step of

$$\mathbf{Re}_{n+1} \equiv \mathbf{Re}\left(\mathbf{F}_{n+1}\right) = \mathbf{e}(\mathbf{F}_{n+1}) = 0 \tag{4.15}$$

Equation (4.15) defines a set of linear or nonlinear equations, which can be solved by a properly chosen algorithm of equation solvers. Then in the general case, the solution of Equation (4.15) arises due to the changing of interface forces

$$\mathbf{F}_{n+1}^{k+1} = \mathbf{F}_{n+1}^{k} + \Delta\mathbf{F}_{n+1}^{k+1} \tag{4.16}$$

Here, k is the iteration counter starting from *n* step solution. In the Newton-Raphson method, we note that, to the first order, Equation (4.15) can be approximated as

$$\mathbf{Re}\left(\mathbf{F}_{n+1}^{k+1}\right) \approx \mathbf{Re}\left(\mathbf{F}_{n+1}^{k}\right) + \left(\frac{\partial \mathbf{Re}}{\partial \mathbf{F}}\right)_{\mathbf{F}=\mathbf{F}_{n+1}^{i}} \Delta\mathbf{F}_{n+1}^{k+1} = 0 \tag{4.17}$$

Equation (4.17) gives immediately the iterative corrector as

$$\Delta\mathbf{F}_{n+1}^{k+1} = -\left(\frac{\partial \mathbf{Re}}{\partial \mathbf{F}}\right)_{\mathbf{F}=\mathbf{F}_{n+1}^{k}}^{-1} \cdot \mathbf{Re}\left(\mathbf{F}_{n+1}^{k}\right) \tag{4.18}$$

A series of successive approximation gives

$$\mathbf{F}_{n+1}^{k+1} = \mathbf{F}_{n+1}^{k} - \left(\frac{\partial \mathbf{Re}}{\partial \mathbf{F}} \right)_{\mathbf{F}=\mathbf{F}_{n+1}^{k}}^{-1} \cdot \mathbf{Re}\left(\mathbf{F}_{n+1}^{k} \right) \tag{4.19}$$

Then in the general case, the interface force of the T-T Gluing algorithm can be updated as

$$\mathbf{F}^{k+1} = \mathbf{F}^{k} - \mathbf{\Lambda} \cdot \mathbf{e}^{k} \tag{4.20}$$

where $\mathbf{\Lambda}$ is called the *gluing matrix* or *lambda matrix*, which will be a constant matrix if Equation (4.15) is linear, or a function of **F** if Equation (4.15) is nonlinear. A gluing matrix can be obtained as

$$\mathbf{\Lambda} = \left(\frac{\partial \mathbf{Re}}{\partial \mathbf{F}} \right)_{\mathbf{F}=\mathbf{F}_{n+1}^{k}}^{-1} = \left(\frac{\partial \mathbf{e}}{\partial \mathbf{F}} \right)_{\mathbf{F}=\mathbf{F}_{n+1}^{k}}^{-1} \tag{4.21}$$

Equation (4.20) simply implies that the interface forces can be updated (to satisfy the compatibility conditions) using only the kinematic information at the interface with an update rule such as that shown in Equation (4.20) provided the gluing matrix is obtained. Therefore, the key issue becomes how to obtain the $\mathbf{\Lambda}$ matrix in Equation (4.21) in a systematic and efficient way. Equation (4.21) can be rewritten as

$$\left(\frac{\partial \mathbf{e}}{\partial \mathbf{F}} \right)^{-1} = \left(\sum_{i}^{n} \frac{\partial \mathbf{e}}{\partial \mathbf{u}_{c}{}^{i}} \frac{\partial \mathbf{u}_{c}{}^{i}}{\partial \mathbf{f}_{c}{}^{i}} \frac{\partial \mathbf{f}_{c}{}^{i}}{\partial \mathbf{F}} \right)^{-1} \tag{4.22}$$

and

$$\mathbf{G}^{i} = \frac{\partial \mathbf{u}_{c}^{i}}{\partial \mathbf{f}_{c}^{i}} \tag{4.23}$$

where $\mathbf{G}^{i}$ is called interface flexibility matrix of subsystem *i*, which can be calculated by solving Equation (4.3) independently for each subsystem, where $i = 1, 2, \cdots, n$. It is important to note that $\mathbf{\Lambda}$ is essentially the inverse matrix of an assembly of the subsystem interface flexibility matrices defined in Equation(4.23). Therefore, $\frac{\partial \mathbf{e}}{\partial \mathbf{u}_c{}^i}$ and $\frac{\partial \mathbf{f}_c{}^i}{\partial \mathbf{F}}$ are called *assembly matrices* of subsystem *i*, and we will discuss them further below.

With Equation (4.5) and (4.11) assembly matrices for serial type connections can be written as

$$\begin{aligned}
\mathbf{B}_1 &= \frac{\partial \mathbf{e}}{\partial \mathbf{u}_c^1} = \begin{bmatrix} -\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}^T \; \textit{1-st subsystem} \\
\mathbf{B}_i &= \frac{\partial \mathbf{e}}{\partial \mathbf{u}_c^i} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \vdots & \vdots \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \\ \vdots & \vdots \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \left.\begin{matrix} \\ \\ \end{matrix}\right\} i-th \textit{ subsystem} \qquad (4.24) \\
\mathbf{B}_n &= \frac{\partial \mathbf{e}}{\partial \mathbf{u}_c^n} = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{bmatrix}^T \; n\textit{-th subsystem}
\end{aligned}$$

Then from Equation (4.7), (4.9), we have

$$\mathbf{C}^i = \frac{\partial \mathbf{f}_c{}^i}{\partial \mathbf{F}} = \mathbf{B}^{iT} = \begin{matrix} \begin{bmatrix} -\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} & \textit{1-st subsystem} \\ \left.\begin{bmatrix} \mathbf{0} & \cdots & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & -\mathbf{I} & \cdots & \mathbf{0} \end{bmatrix}\right\} & \textit{i-th subsystem} \\ \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{bmatrix} & \textit{n-th subsystem} \end{matrix} \qquad (4.25)$$

In the case of multiple type connections, assembly matrices of subsystem *i* (multiple connection at reactive interface of *i-th* subsystem) from Equation (4.12) can be written as

$$\mathbf{B}_i = \frac{\partial \mathbf{e}}{\partial \mathbf{u}_c^i} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \vdots & \vdots \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \\ \vdots & \vdots \\ \mathbf{0} & -\mathbf{I} \\ \vdots & \vdots \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \left.\left.\begin{matrix} m\ connetion \\ of\ \ subsystems \end{matrix}\right\}\right\} i-th\ subsystem \tag{4.26}$$

In the same manner of serial type gluing, we have

$$\mathbf{C}^i = \frac{\partial \mathbf{f}_c^{\ i}}{\partial \mathbf{F}} = \mathbf{B}^{iT} \tag{4.27}$$

Usually, the number of interface degrees of freedom is much smaller than the number of total degrees of freedom of the subsystem model. Therefore, $\mathbf{\Lambda}$ can be easily calculated when $\mathbf{G}^i, (i = 1, 2, \cdots, n)$ are obtained. The proposed approach treats each subsystem as a black box without accessing its internal information. Subsystem interface matrices, $\mathbf{G}^i$, can be calculated by calling the independent solvers associated with the subsystem models, and the subsystems can be glued together using only the interface information.

The gluing algorithm described above can be extended for solving finite element and multibody dynamics problems. For integration of finite element method based dynamics and multibody dynamics problem, the error measure can include accelerations, velocities and displacements at the interfaces or any combination of them. A similar process can be conducted to calculate the gluing matrix and thus couple the subsystem models. Examples show the integration of multibody dynamics and finite element simulations by a gluing algorithm in the next section.

4.2 EXAMPLES

4.2.1 Coupling Rigid Body Link and Flexible Link of 2-D Beam FE Models

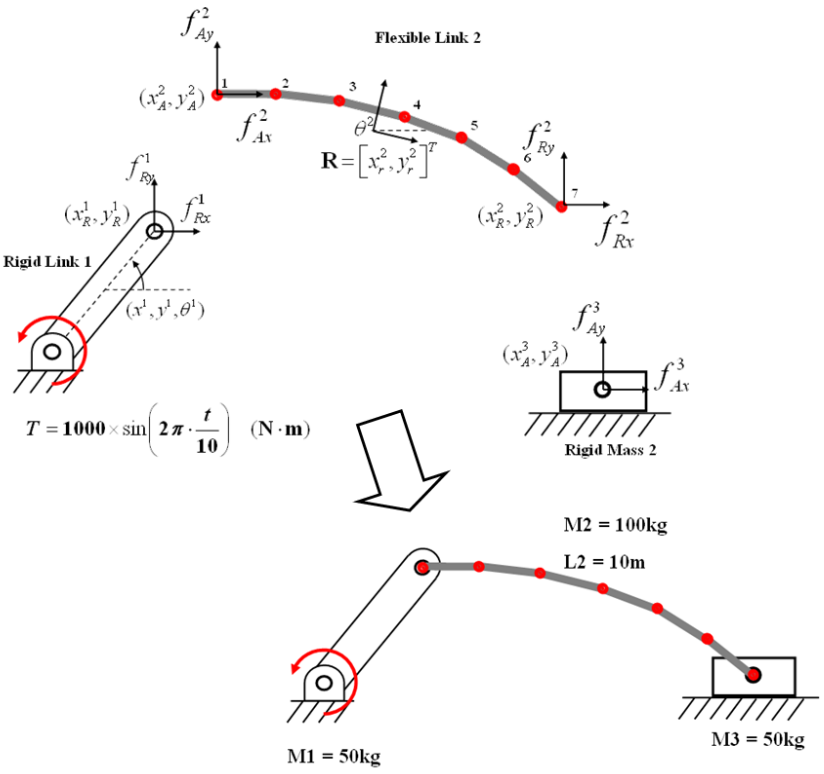


Figure 4.2 Coupling rigid link of MBD and flexible link of finite beam elements

This example demonstrates coupling a general multibody simulation and FE simulation, using the partitioned iteration method. This example also demonstrates how to decouple and integrate the rigid multibody motion and elastic deformation in the D-Sim using the developed gluing algorithm. As shown in Figure 4.2, the flexible body is the same model as two dimensional beam element model used in Chapter 3. The rigid links are solved using distributed ODE solvers. Each part is solved on its own independent processor and then glued together. The prescribed input is a sinusoidal speed of the first link.

The equation of motion of a multibody dynamics model for subsystems 1 and 3 can be written as:

$$\begin{cases} \mathbf{M}^i\mathbf{a}^i + \mathbf{\Phi}_q^{i\,T}\boldsymbol{\lambda}^i = \mathbf{Q}^i \\ \mathbf{\Phi}^i = \mathbf{0} \end{cases} \tag{4.28}$$

where, $\mathbf{M}^i$ denotes the inertia matrix, $\mathbf{a}^i$ denotes the acceleration vector, $\mathbf{Q}^i$ denotes the generalized force vector, $\boldsymbol{\lambda}^i$ is the vector of Lagrange multipliers, $\mathbf{\Phi}^i$ represents the internal constraints, which are assumed to be holonomic. In Fig. 4.2, the revolute joint connecting rigid link 1, flexible link 2 and rigid mass 3 are cut to form three subsystems. Then, the T-T gluing algorithm is applied as follows.

The equations of motion for first subsystems is

$$\mathbf{a}^1 = \ddot{\mathbf{q}}^1 = \begin{Bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{\theta}_1 \end{Bmatrix}, \ \mathbf{M}^1 = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_1 & 0 \\ 0 & 0 & I_1 \end{bmatrix} \tag{4.29}$$

where $\mathbf{q}^1 = \{x_1, y_1, \theta_1\}^T$, $I_1 = \frac{1}{12}m_1 l_1^2$, and $m_1 = 50$ and $l_1 = 3$ are the mass and length of link 1. For generalized force and constraint equations of the first linkage, we have

$$\mathbf{Q}^1 = \begin{Bmatrix} f_{Rx}^1 \\ f_{Ry}^1 \\ T + \frac{l_1}{2}(-f_{Rx}^1\cos\theta_1 + f_{Ry}^1\sin\theta_1) \end{Bmatrix}$$

$$\mathbf{\Phi}^1 = \begin{Bmatrix} x_1 - \frac{l_1}{2}\cos\theta_1 \\ y_1 - \frac{l_1}{2}\sin\theta_1 \end{Bmatrix}, \mathbf{\Phi}_q^1 = \begin{bmatrix} 1 & 0 & -\frac{l^1}{2}\sin\theta_1 \\ 0 & 1 & -\frac{l_1}{2}\cos\theta_1 \end{bmatrix} \tag{4.30}$$

For the third system, rigid mass, we have

$$\mathbf{a}^3 = \ddot{\mathbf{q}}^3 = \begin{Bmatrix} \ddot{x}_3 \\ \ddot{y}_3 \end{Bmatrix}, \ \mathbf{M}^3 = \begin{bmatrix} m_3 & 0 \\ 0 & m_3 \end{bmatrix} \tag{4.31}$$

$$\begin{aligned} &\mathbf{Q}^3 = \begin{Bmatrix} f_{Ax}^3 \\ f_{Ay}^3 \end{Bmatrix} \\ &\mathbf{\Phi}^2 = y_3 \ \text{ and } \mathbf{\Phi}_q^2 = \begin{bmatrix} 0 & 1 \end{bmatrix} \end{aligned} \tag{4.32}$$

For the flexible body of second link, two-dimensional beam elements are solved using the partitioned iteration method in the previous Chapter 3.

$$\begin{bmatrix} \mathbf{M}_{rr}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\theta\theta}^2 \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{R}}_2 \\ \ddot{\boldsymbol{\theta}}_2 \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_r^{\mathbf{ext}} \\ \mathbf{F}_\theta^{\mathbf{ext}} \end{Bmatrix} + \begin{Bmatrix} \mathbf{0} \\ \mathbf{Q}_\theta \end{Bmatrix} \tag{4.33}$$

$$\mathbf{M}_{ff}\ddot{\mathbf{q}}_f + \mathbf{F}^{\mathbf{int}} = \mathbf{F}_f^{\,ext} + \mathbf{Q}_f - \mathbf{M}_{fr}(\ddot{\mathbf{R}} + \underline{\boldsymbol{\lambda}}_t) - \mathbf{M}_{f\theta}(\ddot{\boldsymbol{\theta}} + \underline{\boldsymbol{\lambda}}_r) \tag{4.34}$$

$$\begin{bmatrix} x_A^2 \\ y_A^2 \end{bmatrix} = \mathbf{R}_2 + \mathbf{A}_2\bar{\mathbf{u}}_1 \quad \begin{bmatrix} x_R^2 \\ y_R^2 \end{bmatrix} = \mathbf{R}_2 + \mathbf{A}_2\bar{\mathbf{u}}_7 \quad \left(where \quad \mathbf{A}_2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 \\ \sin\theta_2 & \cos\theta_2 \end{bmatrix} \right) \tag{4.35}$$

The compatibility condition for the system is the continuity of displacements at joint A and joint B which can be written as

$$\mathbf{e} = \begin{Bmatrix} x_A^2 - x_R^1 \\ y_A^2 - y_R^1 \\ x_A^3 - x_R^2 \\ y_A^3 - y_R^2 \end{Bmatrix} = \mathbf{0} \tag{4.36}$$

Here we could also employ acceleration or velocity in the compatibility condition. The equilibrium condition at the joint becomes

$$\begin{Bmatrix} f_{Rx}^{1} + f_{Ax}^{2} \\ f_{Ry}^{1} + f_{Ay}^{2} \\ f_{Rx}^{2} + f_{Ax}^{3} \\ f_{Rx}^{2} + f_{Ax}^{3} \end{Bmatrix} = \mathbf{0} \tag{4.37}$$

As before, we define an interface force vector,

$$\mathbf{F} = \begin{Bmatrix} f_{Ax}^{2} \\ f_{Ay}^{2} \\ f_{Ax}^{3} \\ f_{Ay}^{3} \end{Bmatrix} = \begin{Bmatrix} -f_{Rx}^{1} \\ -f_{Ry}^{1} \\ -f_{Ax}^{2} \\ -f_{Ay}^{3} \end{Bmatrix} \tag{4.38}$$

Applying the T-T method, the $\mathbf{\Lambda}$ matrix can be calculated as

$$\mathbf{\Lambda} = \left[\bar{\mathbf{G}}^{1} + \bar{\mathbf{G}}^{2} + \bar{\mathbf{G}}^{3} \right]^{-1} \tag{4.39}$$

where,

$$\left(\frac{\partial \mathbf{e}}{\partial \mathbf{F}} \right)^{-1} = \left(\sum_{i}^{n} \frac{\partial \mathbf{e}}{\partial \mathbf{u}_{c}{}^{i}} \frac{\partial \mathbf{u}_{c}{}^{i}}{\partial \mathbf{f}_{c}{}^{i}} \frac{\partial \mathbf{f}_{c}{}^{i}}{\partial \mathbf{F}} \right)^{-1} \tag{4.40}$$

In Equation (4.40), interface displacements $\mathbf{u}_{c}{}^{i}$ and interface forces $\mathbf{f}_{c}{}^{i}$ of *i-th* subsystem can be defined as:

$$\mathbf{u}_{c}^{1} = \begin{Bmatrix} x_{R}^{1} \\ y_{R}^{1} \end{Bmatrix}, \quad \mathbf{u}_{c}^{2} = \begin{Bmatrix} x_{A}^{2} \\ y_{R}^{2} \\ x_{A}^{2} \\ y_{A}^{2} \end{Bmatrix}, \quad \mathbf{u}_{c}^{3} = \begin{Bmatrix} x_{A}^{3} \\ y_{A}^{3} \end{Bmatrix} \tag{4.41}$$

$$\mathbf{f}_{c}^{1} = \begin{Bmatrix} f_{Rx}^{1} \\ f_{Ry}^{1} \end{Bmatrix}, \quad \mathbf{f}_{c}^{2} = \begin{Bmatrix} f_{Ax}^{2} \\ f_{Ry}^{2} \\ f_{Ax}^{2} \\ f_{Ay}^{2} \end{Bmatrix}, \quad \mathbf{f}_{c}^{3} = \begin{Bmatrix} f_{Ax}^{3} \\ f_{Ay}^{3} \end{Bmatrix} \tag{4.42}$$

The interface flexibility matrix of subsystem *i*, which can be calculated by solving Equation (4.3) independently for each subsystem:

$$\mathbf{G}_1 = \frac{\partial \mathbf{u}_c^1}{\partial \mathbf{f}_c^1} = \begin{bmatrix} \frac{\partial x_R^1}{\partial f_{Rx}^1} & \frac{\partial x_R^1}{\partial f_{Ry}^1} \\ \frac{\partial y_R^1}{\partial f_{Rx}^1} & \frac{\partial y_R^1}{\partial f_{Ry}^1} \end{bmatrix}, \quad \mathbf{G}_3 = \frac{\partial \mathbf{u}_c^3}{\partial \mathbf{f}_c^3} = \begin{bmatrix} \frac{\partial x_A^3}{\partial f_{Ax}^3} & \frac{\partial x_A^3}{\partial f_{Ay}^3} \\ \frac{\partial y_A^3}{\partial f_{Ax}^3} & \frac{\partial y_A^3}{\partial f_{Ay}^3} \end{bmatrix}$$

$$\mathbf{G}_2 = \frac{\partial \mathbf{u}_c^2}{\partial \mathbf{f}_c^2} = \begin{bmatrix} \frac{\partial x_R^2}{\partial f_{Rx}^2} & \frac{\partial y_R^2}{\partial f_{Rx}^2} & \frac{\partial x_A^2}{\partial f_{Rx}^2} & \frac{\partial y_A^2}{\partial f_{Rx}^2} \\ \frac{\partial x_R^2}{\partial f_{Ry}^2} & \frac{\partial y_R^2}{\partial f_{Ry}^2} & \frac{\partial x_A^2}{\partial f_{Ry}^2} & \frac{\partial y_A^2}{\partial f_{Ry}^2} \\ \frac{\partial x_R^2}{\partial f_{Ax}^2} & \frac{\partial y_R^2}{\partial f_{Ax}^2} & \frac{\partial x_A^2}{\partial f_{Ax}^2} & \frac{\partial y_A^2}{\partial f_{Ax}^2} \\ \frac{\partial x_R^2}{\partial f_{Ay}^2} & \frac{\partial y_R^2}{\partial f_{Ay}^2} & \frac{\partial x_A^2}{\partial f_{Ay}^2} & \frac{\partial y_A^2}{\partial f_{Ay}^2} \end{bmatrix} \tag{4.43}$$

Then, assembly matrices of subsystem from Eqaution (4.24) and (4.25) can be written as:

$$\mathbf{C}^1 = \mathbf{B}^{1T} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}, \quad \mathbf{C}^3 = \mathbf{B}^{3T} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{C}^2 = \mathbf{B}^{2T} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.44}$$

For a nonlinear subsystem we approximate

$$\frac{\partial x_m}{\partial f_n} \approx \frac{\Delta x_m}{\Delta f_n} \tag{4.45}$$

where $\frac{\partial x_m}{\partial f_n}$ represents a component flexible matrix. Obviously, the amplitude of the Δf_n will affect the outcome of the $\mathbf{\Lambda}$ matrix, and in general, the $\mathbf{\Lambda}$ matrix will be a function of the interface forces calculated at the previous step. In this paper, Δf_n is determined by a percentage of the corresponding interface force component, namely

$$\Delta f_n = \varepsilon f_n \tag{4.46}$$

where, ε is a small ratio, which can be typically selected as $\varepsilon = 0.01$.

The reaction force time histories between the first link and the coupler are shown in Figure 4.3 for both a flexible and rigid coupler model. Figure 4.3 shows the magnitude of converged interface force and the link position versus simulation time is depicted in the Figure 4.4. The time of peaks when 1st and 2nd link are collinear with each other(1st link position is 0° or 180°) are different, as expected.

Figure 4.6 and 4.7 show comparisons of a distributed simulation results with ADAMS/Flex[73] simulation. The interface force and angular velocity of the flexible links are in a good agreement in global response. However, the flexible body responses differ slightly. ADAMS/Flex employs flexible elements whose component modes-based data are based on finite element codes such as NASTRAN. Table 4.1 shows the natural frequencies of two FE code beam models which result in the slight differences in simulation results.

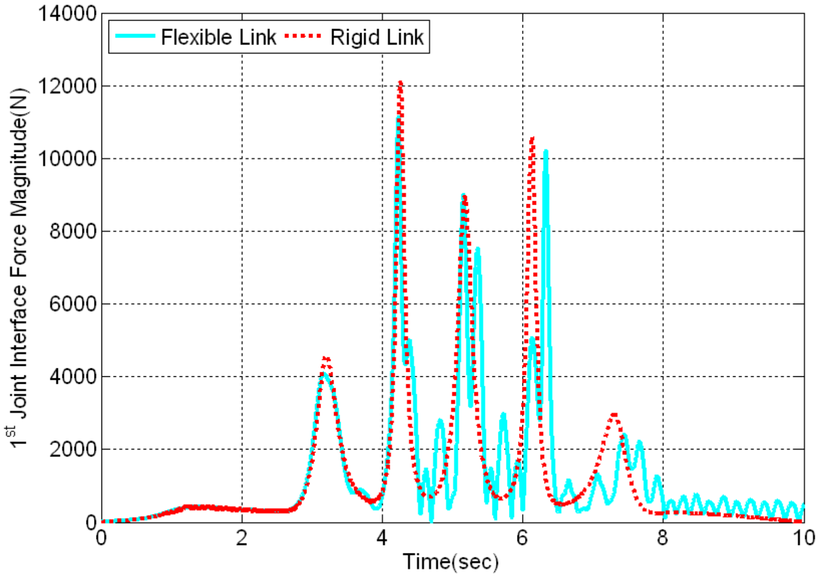


Figure 4.3 Interface Force Magnitude between 1st and 2nd Link

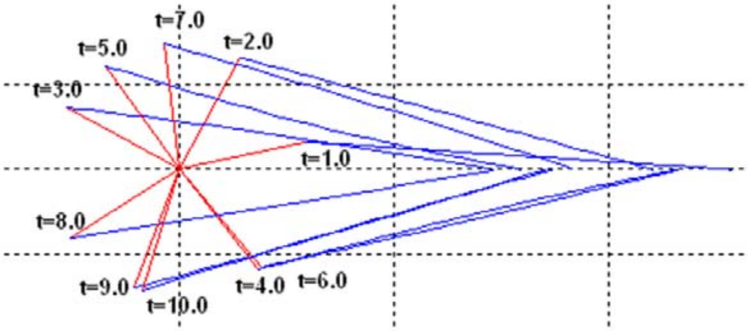


Figure 4.4 Link Position at each time

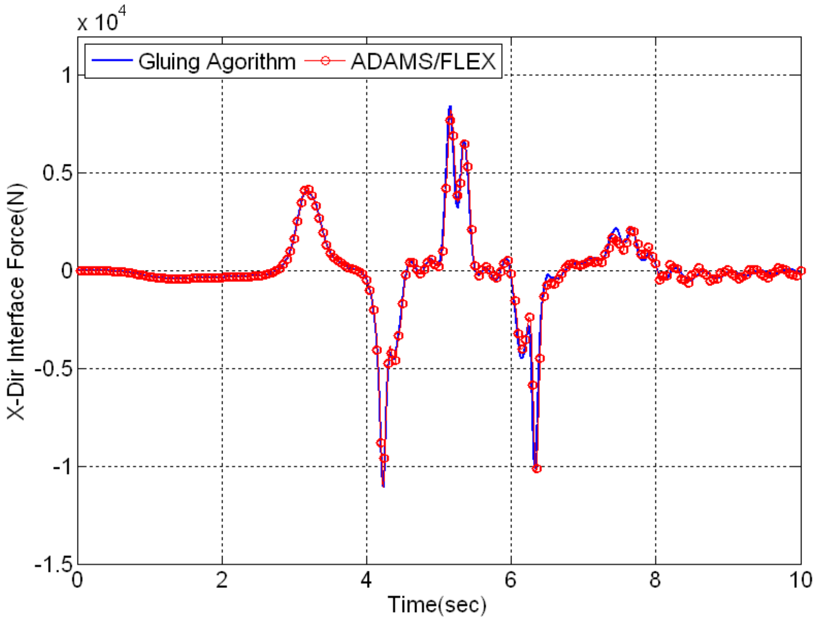


Figure 4.5 X directional Interface Force of 1st and 2nd Link

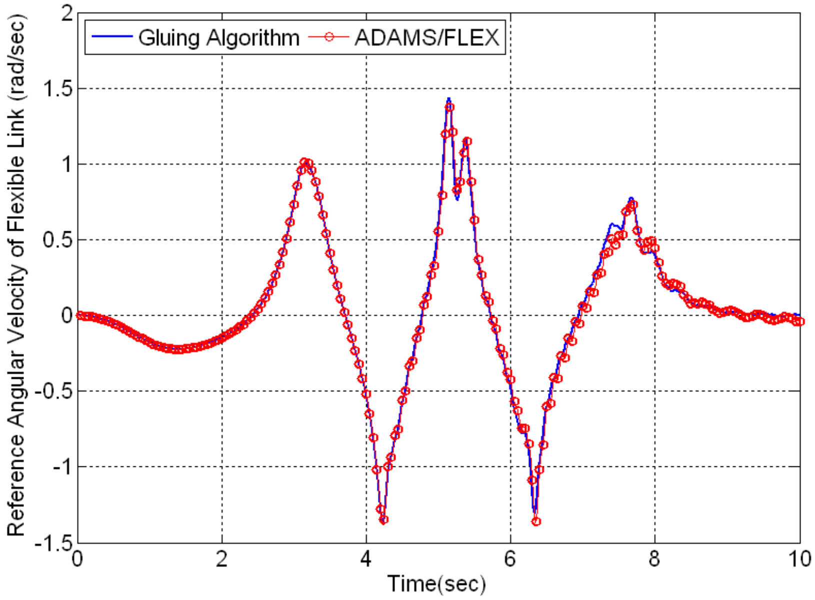


Figure 4.6 Angular velocity of Reference frame of Flexible Link

Table 4.1 Natural Frequency of Beam Model (Hz)

	NASTRAN	Flexible body		NASTRAN	Flexible body
1st	1.52E+01	1.54E+01	6th	1.32E+02	1.39E+02
2nd	2.96E+01	3.02E+01	7th	1.69E+02	1.78E+02
3rd	4.87E+01	4.99E+01	8th	2.09E+02	2.23E+02
4th	7.22E+01	7.46E+01	9th	2.53E+02	2.72E+02
5th	1.00E+02	1.04E+02	10th	2.71E+02	2.73E+02

4.2.2 Distributed Simulation of Coupling MBD Model in ADAMS and FEM model in FEAP

In this example, an upper body model and 4-Post excitation test model are solved using the gluing algorithm to demonstrate the coupled simulation of a multibody dynamics and a finite element model. The models are shown in Figure 4.7. A 4-Post is a MBD model built in ADAMS/View and each post is modeled with a rigid body and a spring-damper assembly. The body model uses the elastic shell element in FEAP, which has been wrapped in the demonstration simulation platform developed for D-Sim. The model comprises 6546 nodes and 6586 elements. Each post is glued at the body mount position and excited in the vertical direction using the time history shown in Figure 4.8

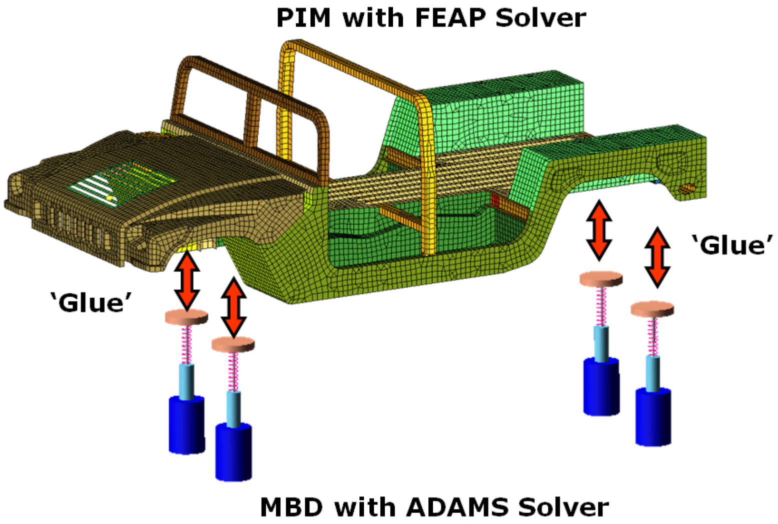


Figure 4.7 Simulation Model

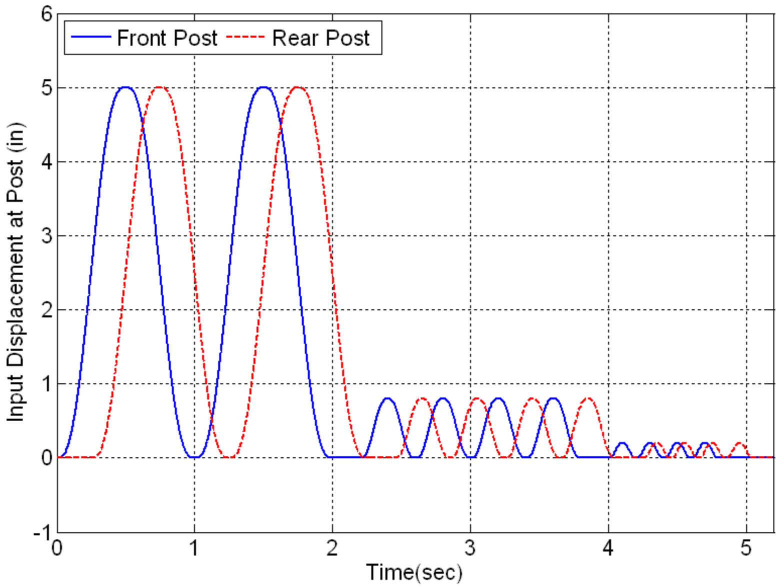


Figure 4.8 Input profiles of displacement at the front and rear post simulation model

The simulation is conducted with a global gluing time step of 0.001 s for a 5.2 s total simulation time. The average iteration number in each time step is 2.96 for the 5200-step simulation, which is quite small for such a non-linear problem. Figure 4.9 shows the normal interface force at the front body mount. Figure 4.10 shows the snapshot of body stress contour and Figure 4.11 plots the stress time history at the area of side armor for the comparing the results using ADAMS/FLEX simulation code. As shown Figures 4.9 and 4.11, the results of interface force and stress magnitude are in good agreement with each simulation results.

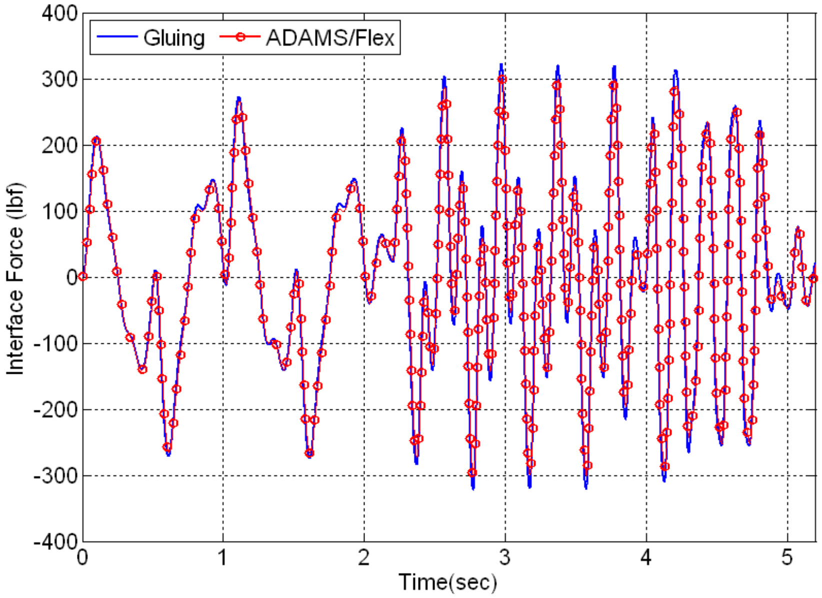


Figure 4.9 Vertical directional interface forces at front body mount

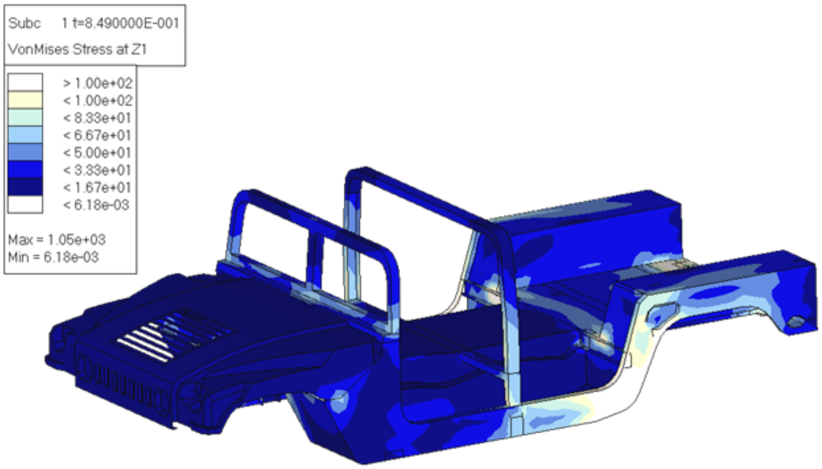


Figure 4.10 Upper body Stress Contour (Von Misses)

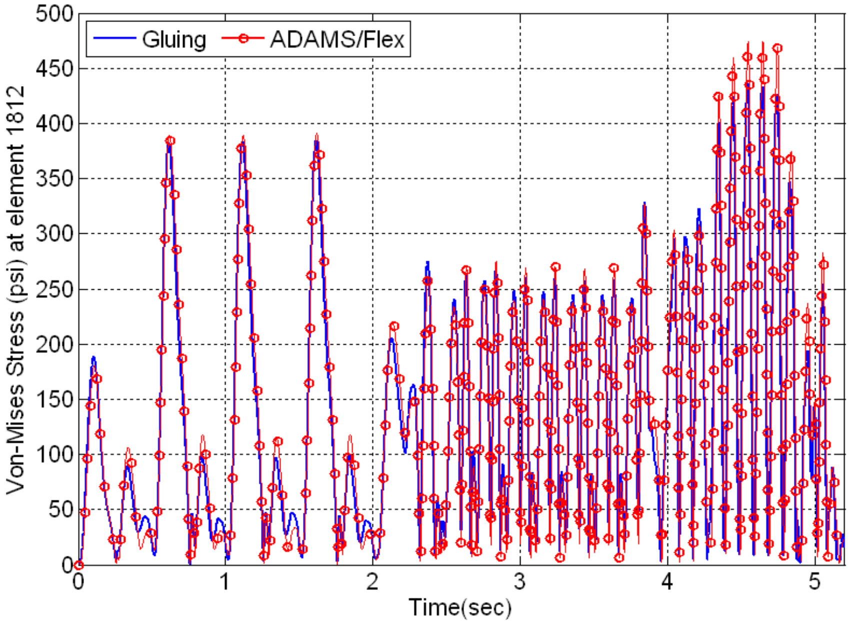


Figure 4.11 Von-Mises stress time history of gluing and ADAM/Flex results

CHAPTER 5

ENHANCED GLUING ALGORITHM FOR DISTRIBUTED SIMULATION

A Distributed Simulation Platform was developed to support the design tasks in multi-layered, distributed supply chains in modern manufacturing systems. Functionally and geographically distributed subsystems can be integrated only with the information at the interfaces of each subsystem without converting or sharing full subsystem models and without putting them together into one monolithic, assembled model. A Gluing Algorithm is a key component to realize distributed simulation to integrate different subsystem parts. In order to apply the gluing algorithm to general simulation problem, further study regarding its convergence, accuracy and efficiency is needed to exploit the full potentials of the gluing algorithm.

One issue is an application of the gluing algorithm to integrate general models, which may be assembled with diverse connecting methods, including spot welds, bolts, bushings, and other physical connections. A proper modeling of flexible joint forces plays an important role in predicting the dynamic behavior of assembled system. Another issue is a stable gluing algorithm in the case of integrating models solved with different simulation schemes such as general multibody dynamic codes and linear elastic finite element codes.

In this chapter, improved gluing algorithms are proposed for assembling the subsystem models to apply to general problems in distributed simulation. A flexible gluing algorithm which subsystem parts can improve the accuracy of the simulation to represent the real physical system and also can improve the convergence of distributed simulation.

5.1 FLEXIBLE GLUING ALGORITHM

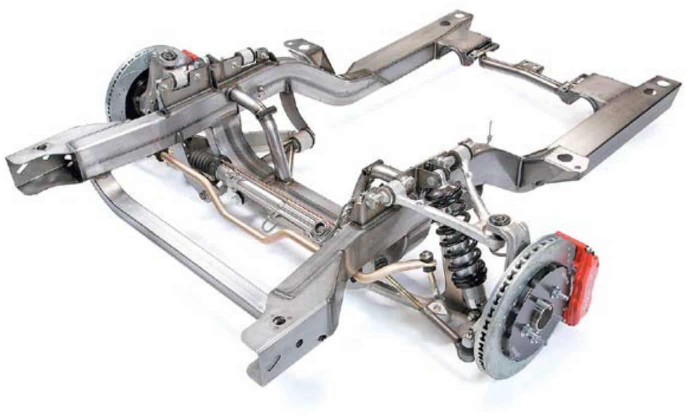


Figure 5.1 Vehicle underbody assembly connected with diverse joint elements

Current research in multibody dynamic and finite element analysis, especially in the automotive industry evolves around the formulation of the governing equations that would include joint and link flexibility effects during motion, and specified motions to certain links and end-effecters. Figure 5.1 depicts a chassis frame of a passenger vehicle including suspensions, axle assembly and brake system. It is also possible to model flexible connectors such as the rubber bushes so commonly used to isolate vibration in vehicle suspensions. Proper modeling of flexible joint forces plays an important role in predicting the dynamic behavior of the assembled system. System elements such as springs and dampers with non-linear force characteristics are also important to model properly.

The primary role of the bushes in a suspension system of a passenger vehicle is to isolate the vehicle and driver from small amplitude, high frequency road inputs, so as to improve the ride quality of the vehicle. Blundell [84] and Amirouche [85] described the influence of rubber bush compliance on changes in suspension geometry during vertical movement relative to the vehicle body. The effects of the bushes on vehicle handling depend on whether the bushes have any influence on geometric changes in the suspension and road wheel as the wheel moves vertically relative to the vehicle body. To make good prediction regarding the characteristics of a flexible joint for vibration isolation in the design process, numerical models can be used. However, for a reliable prediction of the dynamic behavior of the isolator, the rubber material parameters need to be known. Mackerle [86] reviews the finite-element methods applied to the analysis and simulation of rubber and rubber-like materials. Beijers et. al. [87] proposed a method based on an optimization procedure in which the material parameters of a numerical model are updated in such a way that a fit is accomplished with experimentally determined measurement data with static force-displacement material properties and the dynamic material parameters. Ledesma et. al. [88] presented a formulation of nonlinear viscoelastic bushing forces as massless force elements between two bodies in a multibody system.

The flexible gluing algorithm can deal with various connections between subsystems, and can account for flexibility at these connections as shown in the Fig. 5.2. These flexible joints embedded in a gluing algorithm can improve the accuracy of the simulation to represent the real physical system and can ensure convergence at singular points in the multibody dynamic simulation. In the distributed simulation, it is important to accurately predict the dynamic loads that act on the subsystem components at the gluing interface because these loads feed directly into the fatigue life prediction of the components, in the durability analyses.

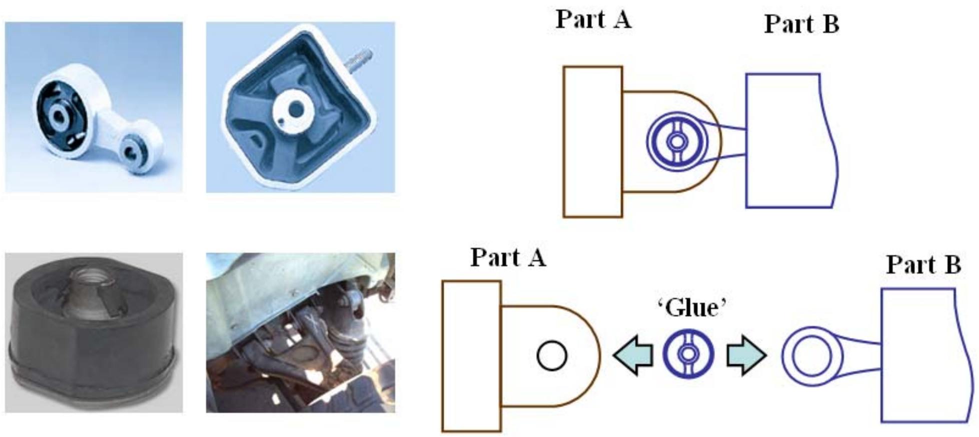
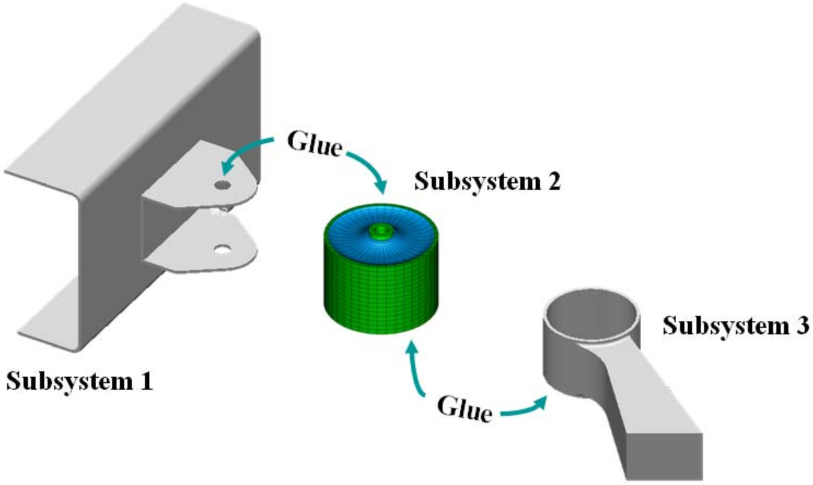


Figure 5.2 Flexible joints at the interface – rubber mounts

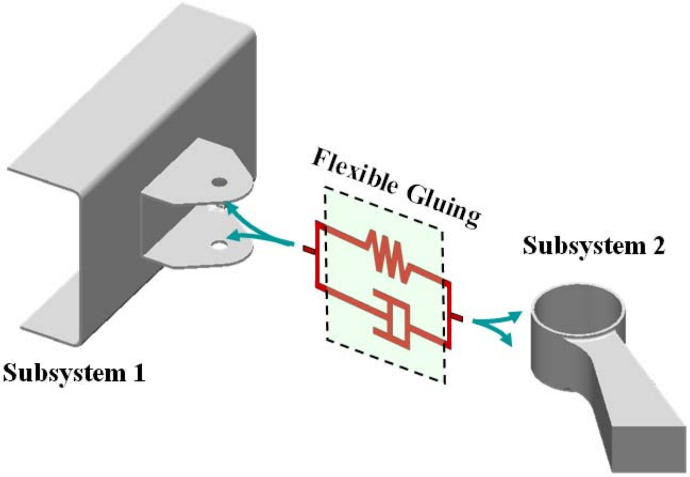
5.1.1 Expression of Flexible Joints in D-Sim

The key component to realize distributed simulation is the gluing algorithm. A gluing algorithm aims at performing successful simulations by establishing an interface between existing codes and must efficiently maintain the “privacy” of the individual component models. Expressing sthe flexible effect of an interface is another challenge of distributed simulation. For the numerical expression of interface or joint flexibilities between subsystem components using gluing algorithm, it is necessary to consider the gluing formulations and the characteristics of a gluing methodology. In chapter 4, we reviewed a general gluing methodology and its formulations for application to distributed simulation.

For expressing flexible effects of interfaces or joints in distributed simulation, there are two types of approaches. First, the flexible joint is modeled the same as other component parts and simulated with an independent solver in D-Sim, as shown in the Figure 5.3-(a). This method can be useful in the case of considering the dynamic response of the flexible component itself. Thus, joint models can be solved in D-Sim without changing of original gluing algorithm. Second, flexible joints are embedded in the gluing algorithm using the simplified model in the Figure 5.3-(b). This method is more efficient in the case when considering the global motion of assembled system. Since the first case can be sufficiently simulated in the proposed D-Sim platform, only the,second case using a flexible gluing algorithm will be discussed in this dissertation.



(a) Integration of flexible joint model



(b) Flexible gluing algorithm

Figure 5.3 Gluing methods of integrating components with a flexible joint in D-Sim

5.1.2 Flexible Gluing Formulation

The objective of the T-T gluing algorithm during iteration is to find interface forces while satisfying the compatibility condition. It is obvious that the simulation results at the interface are the same as the result of the assembled model using a fixed joint at the interface. Figure 5.4 illustrates an integration of subsystems with flexible joints which can be modeled with spring and damper systems to represent the dynamic characteristics of a flexible joint. A flexible joint such as the rubber bushing element in the commercial multibody dynamic programs e.g., ADAMS [89] and DADS [90], can be expressed with a Kelvin-Voight model. This Kelvin-Voight model is superposed with force components of translational and rotational springs and dampers. In this dissertation, in order to express the forces of flexible joints, a Kelvin-Voight model is considered. As shown in the Figure 5.4, a flexible joint can be represented by a spring in parallel with a viscous damper.

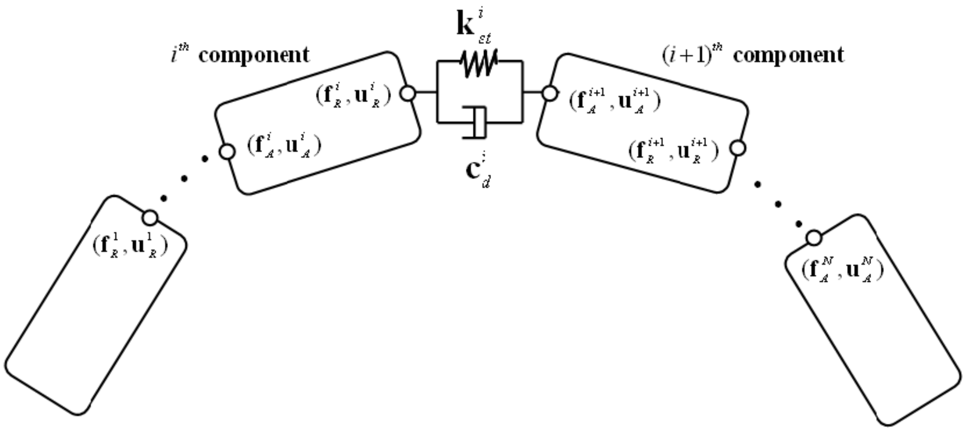

$\mathbf{k}^i_{st}$
i^{th} component
$(\mathbf{f}^i_B, \mathbf{u}^i_B)$
$(\mathbf{f}^i_A, \mathbf{u}^i_A)$
$(\mathbf{f}^1_B, \mathbf{u}^1_B)$
$\mathbf{c}^i_d$
$(i+1)^{th}$ component
$(\mathbf{f}^{i+1}_A, \mathbf{u}^{i+1}_A)$
$(\mathbf{f}^{i+1}_B, \mathbf{u}^{i+1}_B)$
$(\mathbf{f}^N_A, \mathbf{u}^N_A)$

Figure 5.4 Flexible gluing connectors

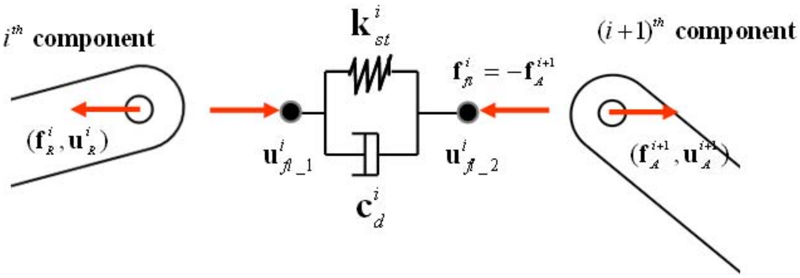

i^{th} component
$\mathbf{k}^{i}_{st}$
$(i+1)^{th}$ component
$\mathbf{f}^{i}_{fl} = -\mathbf{f}^{i+1}_{A}$
$(\mathbf{f}^{i}_{R}, \mathbf{u}^{i}_{R})$
$\mathbf{u}^{i}_{fl\_1}$
$\mathbf{u}^{i}_{fl\_2}$
$(\mathbf{f}^{i+1}_{A}, \mathbf{u}^{i+1}_{A})$
$\mathbf{c}^{i}_{d}$

Figure 5.5 Force equilibrium at the interface between two subsystems

Figure 5.5 depicts the force equilibrium condition at the *i-th* interface between *i-th* and (*i+1*)-*th* subsystem as

$$\mathbf{f}^{i}_{R} = \mathbf{f}^{i}_{fl} = -\mathbf{f}^{i+1}_{A}, \left(i = 1, 2, \cdots, n-1\right) \tag{5.1}$$

To represent the flexible joint, it is necessary that the compatibility condition should be change into constraint condition defining between the relation of interface forces and displacements. The number of flexible joint is the same with the number of interfaces. One can define the displacement of the *i-th* flexible joint as:

$$\mathbf{q}^{i}_{fl} = \mathbf{u}^{i}_{fl\_2} - \mathbf{u}^{i}_{fl\_1} \tag{5.2}$$

where $\mathbf{u}^{i}_{fl\_1}$ and $\mathbf{u}^{i}_{fl\_2}$ are virtual interface nodes for the expressing a relation between forces and displacements of flexible joint:

$$\mathbf{c}^{i}_{d}\dot{\mathbf{q}}^{i}_{fl} + \mathbf{k}^{i}_{st}\mathbf{q}^{i}_{fl} = \mathbf{f}^{i}_{fl} \tag{5.3}$$

where $\mathbf{q}_{fl}^{i}$, $\dot{\mathbf{q}}_{fl}^{i}$ are respectively displacements and velocities of the flexible joint at the i-th interface, and where $\mathbf{c}_{d}^{i}$ and $\mathbf{k}_{st}^{i}$ are respective viscous damping and stiffness matrices to describe the dynamic characteristics of joint elements. These stiffness and damping parameters can be fit to experimental data. These stiffness and damping properties can be linear and/or nonlinear according to the displacements and velocities characteristics. Non-linearity of flexible joints can be important for accurately predicting the dynamic behaviors of an assembled system. Then, the displacement vector of flexible joint can be defined as

$$\mathbf{q}^{T} = \left[\mathbf{q}_{fl}^{1\ T} \ldots \mathbf{q}_{fl}^{i\ T} \cdots \mathbf{q}_{fl}^{n-1T}\right] \tag{5.4}$$

For satisfaction of the compatibility condition at the interface, the interface error measured between subsystems and the displacement of flexible joint must satisfy

$$\mathbf{e} - \mathbf{q} = \mathbf{0} \tag{5.5}$$

The goal of T-T Gluing Method is to finding interface forces, which satisfy compatibility conditions. Therefore, when using Newton-Raphson iteration method, above compatibility condition can be set as residue as a function of interface forces

$$\mathbf{Re} = \mathbf{e} - \mathbf{q} = \mathbf{e}(\mathbf{F}) - \mathbf{q}(\mathbf{F}) \tag{5.6}$$

The problems is therefore formulated as a solution in terms of the discretized parameter at the *n+1* step

$$\mathbf{Re}\left(\mathbf{F}_{n+1}\right) = \mathbf{e}(\mathbf{F}_{n+1}) - \mathbf{q}(\mathbf{F}_{n+1}) \approx \mathbf{0} \tag{5.7}$$

In Chapter 4, the interface errors $\mathbf{e}$ in Equation (5.7) are defined as a set of linear or nonlinear equations, which can be solved by a properly chosen equation solver.

The displacement vector $\mathbf{q}$ of the flexible joint can be solved with an independent solver embedded in the gluing algorithm because the equation of motion of the flexible components is simpler and has fewer degrees of freedoms than other subsystems. In the general case, the solution of Equation (5.7) arises due to the changing of interface forces

$$\mathbf{F}_{n+1}^{k+1} = \mathbf{F}_{n+1}^{k} + \Delta\mathbf{F}_{n+1}^{k+1} \tag{5.8}$$

Here, *k* is the iteration counter starting from *n* step solution (for $k = 0$). In the Newton-Raphson method, we note that, to the first order, Equation (5.7) can be approximated as

$$\mathbf{Re}\left(\mathbf{F}_{n+1}^{k+1}\right) \approx \mathbf{Re}\left(\mathbf{F}_{n+1}^{k}\right) + \left(\frac{\partial\mathbf{Re}}{\partial\mathbf{F}}\right)_{\mathbf{F}=\mathbf{F}_{n+1}^{i}} \Delta\mathbf{F}_{n+1}^{k+1} = 0 \tag{5.9}$$

Equation (4.17) gives immediately the iterative corrector as

$$\Delta\mathbf{F}_{n+1}^{k+1} = -\left(\frac{\partial\mathbf{Re}}{\partial\mathbf{F}}\right)_{\mathbf{F}=\mathbf{F}_{n+1}^{k}}^{-1} \cdot \mathbf{Re}\left(\mathbf{F}_{n+1}^{k}\right) \tag{5.10}$$

A series of successive approximation gives

$$\mathbf{F}_{n+1}^{k+1} = \mathbf{F}_{n+1}^{k} - \left(\frac{\partial\mathbf{Re}}{\partial\mathbf{F}}\right)_{\mathbf{F}=\mathbf{F}_{n+1}^{k}}^{-1} \cdot \mathbf{Re}\left(\mathbf{F}_{n+1}^{k}\right) \tag{5.11}$$

Then in the general case, the interface force of the T-T Gluing algorithm can be updated as

$$\mathbf{F}^{k+1} = \mathbf{F}^{k} - \mathbf{\Lambda}_{fl} \cdot \mathbf{e}^{k} \tag{5.12}$$

where $\mathbf{\Lambda}_{fl}$ is called the *flexible gluing matrix* or *lambda matrix*, which is a function of **F.** A flexible gluing matrix can be obtained as

$$\mathbf{\Lambda}_{fl} = \left(\frac{\partial \mathbf{Re}}{\partial \mathbf{F}}\right)^{-1}_{\mathbf{F}=\mathbf{F}^k_{n+1}} = \left(\frac{\partial \mathbf{e}}{\partial \mathbf{F}} - \frac{\partial \mathbf{q}}{\partial \mathbf{F}}\right)^{-1}_{\mathbf{F}=\mathbf{F}^k_{n+1}} \tag{5.13}$$

In Equation (4.21), $\frac{\partial \mathbf{e}}{\partial \mathbf{F}}$ is already defined in original gluing algorithm as

$$\begin{aligned} \left(\frac{\partial \mathbf{e}}{\partial \mathbf{F}}\right) &= \left(\frac{\partial \mathbf{e}}{\partial \mathbf{u}_c{}^i}\frac{\partial \mathbf{u}_c{}^i}{\partial \mathbf{f}_c{}^i}\frac{\partial \mathbf{f}_c{}^i}{\partial \mathbf{F}}\right) \\ &= \sum_{i=1}^{n} \mathbf{B}^i \mathbf{G}^i \mathbf{C}^i \quad \mathit{(i=1\cdots n)} \end{aligned} \tag{5.14}$$

where $\frac{\partial \mathbf{u}_c{}^i}{\partial \mathbf{f}_c{}^i}$ in the Equation (5.14) is interface flexibility matrix of subsystem *i*, which is same as the flexibility matrix in original gluing algorithm($i = 1, 2, \cdots, n$). Then, flexibility matrix of flexible joints is given by

$$\left(\frac{\partial \mathbf{q}}{\partial \mathbf{F}}\right) = \frac{\partial \mathbf{q}}{\partial \mathbf{q}^i_{fl}}\frac{\partial \mathbf{q}^i_{fl}}{\partial \mathbf{f}^i_{fl}}\frac{\partial \mathbf{f}^i_{fl}}{\partial \mathbf{F}} \tag{5.15}$$

with Equation (5.4), and assembly matrices can be written as

$$\frac{\partial \mathbf{q}}{\partial \mathbf{q}^i_{fl}} = \mathbf{B}^i_{fl} = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{I} \\ \vdots \\ \mathbf{0} \end{bmatrix} \tag{5.16}$$

Using the force equilibrium condition of Equation (5.1), we have

$$\left(\frac{\partial \mathbf{f}^i_{fl}}{\partial \mathbf{F}}\right) = \mathbf{C}^i_{fl} = -\mathbf{B}^i_{fl}{}^T = \begin{bmatrix} \mathbf{0} & \cdots & -\mathbf{I} & \cdots & \mathbf{0} \end{bmatrix} \tag{5.17}$$

Then, the flexible matrix of flexible joints for the gluing matrix can be written as

$$\left(\frac{\partial \mathbf{q}}{\partial \mathbf{F}}\right) = \bar{\mathbf{G}}_{fl} = \sum_{i}^{N-1} \mathbf{B}^{i}_{fl}\mathbf{G}^{i}_{fl}\mathbf{C}^{i}_{fl} = -\begin{bmatrix} \frac{\partial \mathbf{q}^{1}_{fl}}{\partial \mathbf{f}^{1}_{fl}} & \mathbf{0} & \cdots & & \mathbf{0} \\ \mathbf{0} & \ddots & & & \vdots \\ \vdots & & \frac{\partial \mathbf{q}^{i}_{fl}}{\partial \mathbf{f}^{i}_{fl}} & & \\ & & & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & & \mathbf{0} & \frac{\partial \mathbf{q}^{n-1}_{fl}}{\partial \mathbf{f}^{n-1}_{fl}} \end{bmatrix} \tag{5.18}$$

In the case of multiple type connections, force equilibrium condition of Equation (5.1) should be changed to

$$\begin{gathered} \mathbf{f}^{i}_{R} = \mathbf{f}^{i}_{fl} = -\mathbf{f}^{i+1}_{A} - \mathbf{f}^{i+2}_{A} \cdots - \mathbf{f}^{i+m}_{A} \\ \left(\textit{m : multiple connected subsystem number with the i-th subsystem}\right) \end{gathered} \tag{5.19}$$

and the compatibility condition can be written as

$$\mathbf{q} = \left\{\begin{matrix} \vdots \\ \mathbf{q}^{i}_{fl} \\ \mathbf{q}^{i}_{fl} \\ \vdots \\ \mathbf{q}^{i}_{fl} \\ \vdots \end{matrix}\right\} \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} i-th\ interface \tag{5.20}$$

Then, the assembly matrices of subsystem *i* from Equation (5.19), (5.21) can be written as

$$\frac{\partial \mathbf{q}}{\partial \mathbf{q}^{i}_{fl}} = \mathbf{B}^{i}_{fl} = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{I} \\ \vdots \\ \mathbf{I} \\ \vdots \\ \mathbf{0} \end{bmatrix} \left.\begin{matrix} \left.\begin{matrix} m\ connetions \\ of\ \ subsystems \end{matrix}\right\} \end{matrix}\right\} i-th\ subsystem \tag{5.22}$$

From same manner of serial type gluing, we have

$$\mathbf{C}^{i} = \frac{\partial \mathbf{f}_{fl}{}^{i}}{\partial \mathbf{F}} = -\mathbf{B}^{iT} \tag{5.23}$$

Then, the flexible matrix of flexible joints for the gluing matrix can be written as

$$\left(\frac{\partial \mathbf{q}}{\partial \mathbf{F}}\right) = \sum_{i}^{n-m-1} \mathbf{B}^{i}_{fl}\mathbf{G}^{i}_{fl}\mathbf{C}^{i}_{fl} = -\begin{bmatrix} \frac{\partial \mathbf{q}^{1}_{fl}}{\partial \mathbf{f}^{1}_{fl}} & \mathbf{0} & & \cdots & \cdots & & \mathbf{0} \\ \mathbf{0} & \ddots & & & & & \\ & & \frac{\partial \mathbf{q}^{i}_{fl}}{\partial \mathbf{f}^{i}_{fl}} & & & & \vdots \\ \vdots & & & \ddots & & & \vdots \\ \vdots & & & & \frac{\partial \mathbf{q}^{i}_{fl}}{\partial \mathbf{f}^{i}_{fl}} & & \\ & & & & & \ddots & \mathbf{0} \\ \mathbf{0} & & \cdots & \cdots & & \mathbf{0} & \frac{\partial \mathbf{q}^{n-1}_{fl}}{\partial \mathbf{f}^{n-1}_{fl}} \end{bmatrix} \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} i-th\ interface \tag{5.24}$$

5.2 PENALIZED METHOD FOR GLUING ALGORITHM

To integrate subsystem parts with gluing algorithms in D-Sim requires finding the interface forces that satisfy compatibility conditions, when using the T-T gluing method. In this section, the T-T gluing algorithm is improved using a stabilized constraint approach. Consequently, the new algorithm can be applied to dynamics simulation of both finite element models and multibody dynamics system models.

5.2.1 Strategies of Gluing Algorithm

A gluing algorithm is the fundamental and essential part of a distributed simulation. Thus it is necessary to the understand gluing algorithm to improve its application to the integration of general purpose simulation models. In [16], the author investigated five different gluing algorithms namely, BCG (Baumgarte's Stabilization with CG solver), RAL (Relaxed Augmented Lagrange Method), PSS (Park's Staggered Stabliliation Scheme) and MOP (Manifold Orthogonal Projection Method) MEPI (Maggi's Equations with Perturbed Iteration), which are suitable for systems that can be modeled by semi-discrete DAEs of the form

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{Q} \\ \mathbf{\Phi} = \mathbf{0} \end{cases} \tag{5.25}$$

where $\mathbf{q}$ represents the generalized position coordinates of the system, $\mathbf{M}$ the generalized inertia matrix, $\boldsymbol{\lambda}$ the Lagrange multiplier vector, $\mathbf{\Phi}$ the holonomic constraints in the system, in which $\mathbf{\Phi}_{\mathbf{q}} = \frac{\partial \mathbf{\Phi}}{\partial \mathbf{q}}$ and which is assumed to have full row rank. Equation (5.25) is essentially the equation of motion for a general multibody dynamics system. When the system is decomposed into several sub-domains, the inertia matrix $\mathbf{M}$ and the generalized force vector $\mathbf{Q}$ can be decoupled easily into sub-domains. However, the constraint force, $\mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda}$, usually cannot be easily decoupled.

In the case of coupling MBD and FEM models using the T-T gluing algorithm, it is difficult to satisfy kinematic variables (accelerations, velocities and displacements) simultaneously at the interface, since each set of simulation results at the interface is computed, in general, with different simulation schemes. For example, if the accelerations of each subsystem part are matched at the interface using a gluing algorithm, velocity or displacement errors can be generated during the distributed simulation. In the case of gluing different simulation subsystems such as MBD and FEM, satisfying compatibility conditions in other kinematic variables cannot be guaranteed with single variable gluing, and instability problems induced by interface error accumulation can cause distributed simulations to diverge.

5.2.2 Stabilized Constraint Approach

One can reduce the index of DAEs by repeatedly differentiating the constraint equations in Equation (5.25) to take advantage of their better numerical properties for general time integration schemes. Differentiation of the constraint will transform the original DAEs to lower index system with invariants. Some stabilization measures must be taken if longer time simulation is required. Baumgarte's stabilization [91] is probably the most widely used scheme for practical engineering applications. It replaces the holonomic constraint in Equation (5.25) by linear combination of the constraint and its time derivatives in such a way that the differential equations for the constraints are stable.

The acceleration constraint, $\ddot{\mathbf{\Phi}} = 0$, is obtained by differentiating Equation (5.25), and the resultant index-1 DAEs are

$$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_{\mathbf{q}}^{T} \\ \mathbf{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{bmatrix} Q \\ -\left(\dot{\mathbf{\Phi}}_{\mathbf{q}} \dot{\mathbf{q}} + \dot{\mathbf{\Phi}}_{t} \right) \end{bmatrix} \tag{5.26}$$

An analysis of $\ddot{\mathbf{\Phi}} = 0$ leads to two repeated poles at the origin. This is a marginally stable system. A small perturbation, such as round-off errors from numerical time integration, will accumulate violations in the position invariant $\mathbf{\Phi}$, which grow at a quadratic rate with respect to time step size. To damp out these violations, Baumgrate proposed using

$$\ddot{\mathbf{\Phi}} + 2\alpha\dot{\mathbf{\Phi}} + \beta^{2}\mathbf{\Phi} = 0 \tag{5.27}$$

By carefully choosing the α and β, we can obtain a stable solution of DAEs. However, these parameters can only be tuned on a case-by-case basis in such a way that the dynamics of the system matches the decay time constant.

5.2.3 Penalized Method Using Baumgarte's Stabilization

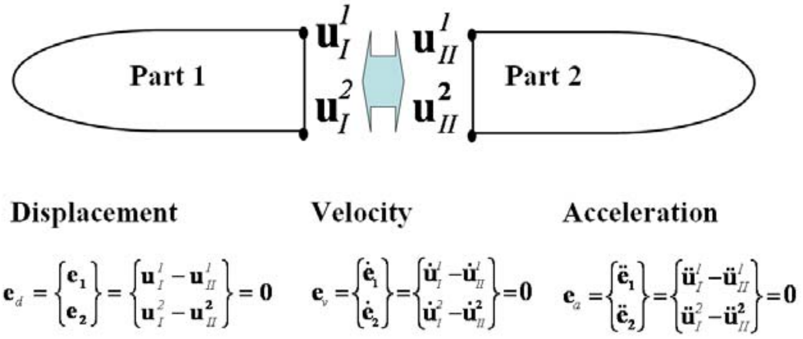

$\mathbf{u}_I^1$
$\mathbf{u}_{II}^1$
Part 1
Part 2
$\mathbf{u}_I^2$
$\mathbf{u}_{II}^2$
Displacement
Velocity
Acceleration
$\mathbf{e}_d = \begin{Bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{Bmatrix} = \begin{Bmatrix} \mathbf{u}_I^1 - \mathbf{u}_{II}^1 \\ \mathbf{u}_I^2 - \mathbf{u}_{II}^2 \end{Bmatrix} = \mathbf{0}$
$\mathbf{e}_v = \begin{Bmatrix} \dot{\mathbf{e}}_1 \\ \dot{\mathbf{e}}_2 \end{Bmatrix} = \begin{Bmatrix} \dot{\mathbf{u}}_I^1 - \dot{\mathbf{u}}_{II}^1 \\ \dot{\mathbf{u}}_I^2 - \dot{\mathbf{u}}_{II}^2 \end{Bmatrix} = \mathbf{0}$
$\mathbf{e}_a = \begin{Bmatrix} \ddot{\mathbf{e}}_1 \\ \ddot{\mathbf{e}}_2 \end{Bmatrix} = \begin{Bmatrix} \ddot{\mathbf{u}}_I^1 - \ddot{\mathbf{u}}_{II}^1 \\ \ddot{\mathbf{u}}_I^2 - \ddot{\mathbf{u}}_{II}^2 \end{Bmatrix} = \mathbf{0}$

Figure 5.6 Compatibility conditions at the interface

As shown in the Figure 5.6, the compatibility condition should be satisfied with all kinetic variables at the interface. Since each subsystem simulation result is computed with a different simulation scheme, a penalized compatibility condition can provide more stable results at every time step using a Baumgarte stabilization.

$$\mathbf{e} = \mathbf{e}_a + 2\xi\omega\mathbf{e}_v + \omega^2\mathbf{e}_d \approx 0 \tag{5.28}$$

where ω and ξ are the respective natural frequency and damping ratio of penalized compatibility condition. Consequently, the Gluing Matrix should be calculated as same manner with the definition of interface error.

$$\mathbf{\Lambda} = \left(\frac{\partial \mathbf{e}_a}{\partial \mathbf{F}} + 2\xi\omega \frac{\partial \mathbf{e}_v}{\partial \mathbf{F}} + \omega^2 \frac{\partial \mathbf{e}_d}{\partial \mathbf{F}} \right)^{-1} \tag{5.29}$$

5.3 EXAMPLES

In this section, four examples are presented to demonstrate the developed simulation methodology with an emphasis on the flexible gluing algorithm. First, a one dimensional spring mass system connected with linear and nonlinear flexible gluing models is used for verification of proposed flexible gluing algorithm. Second, a four bar link model of a rigid multibody dynamics simulation is simulated to demonstrate that the flexible gluing algorithm can improve the convergence of the fixed type gluing algorithm for multibody dynamics simulation that may have singularity problems. Third, the four-bar link model with deformable body solved with PIM in Chapter 4 is used to illustrate a penalized method for gluing algorithm application. Finally an integration application of a FE upper body model and a multibody dynamics model of a frame and suspension on a 4 post shaker test system is used to demonstrate the flexible gluing algorithm to incorporate the effects of body mounts.

5.3.1 One Dimensional Spring Mass System Connected with Linear and Non-Linear Flexible Gluing Joints

To demonstrate the proposed flexible gluing algorithm, simple distributed subsystem models are set as shown in the Figure 5.7. A one dimensional spring-mass assembled model is solved using the proposed flexible gluing algorithm and compared with the result of an all-at-once model including linear and non-linear spring dampers. The parameters for the spring-mass subsystem are

$$\begin{aligned} &m_1 = m_2 = m_3 = 10\ kg \\ &k = 100\ N/m \\ &F = 10N\ \textit{(constant)} \end{aligned}$$

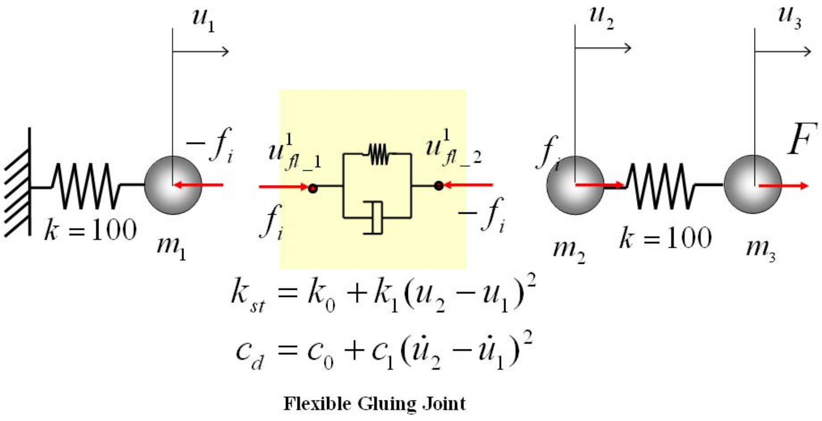


Figure 5.7 One dimensional spring mass model connected with flexible joints

A constant force F_{ext} excites m_3 of the second subsystem. The first subsystem is a one degree of freedom mass and spring system described as

$$m_1\ddot{u}_1+ku_1=-f_i \tag{5.30}$$

The second subsystem is two degree of freedom mass and spring system written as

$$\begin{bmatrix} m_2 & 0 \\ 0 & m_3 \end{bmatrix}\begin{Bmatrix} \ddot{u}_2 \\ \ddot{u}_3 \end{Bmatrix}+\begin{bmatrix} k & -k \\ -k & k \end{bmatrix}\begin{Bmatrix} u_2 \\ u_3 \end{Bmatrix}=\begin{bmatrix} f_i \\ F \end{bmatrix} \tag{5.31}$$

The number of subsystems is two, and the number of interface is one. The flexible joint is embedded in the gluing algorithm, where the interface force/displacement relation can be written as

$$c_d\dot{q}_{fl}+k_{st}q_{fl}=-f_1 \tag{5.32}$$

in which,

$$q_{fl}=u^1_{fl\_2}-u^1_{fl\_1} \tag{5.33}$$

Because there is only one interface, the assembly matrix to obtain the gluing matrix is simple using Equation (4.24) and (5.16) as

$$\mathbf{B}^1 = \mathbf{C}^1 = [-1] \quad \mathbf{B}^2 = \mathbf{C}^2 = [1] \quad \mathbf{B}^1_{fl} = -\mathbf{C}^1_{fl} = [1] \tag{5.34}$$

The flexible gluing joint properties are given as

Linear Flexible Gluing Case:

$$\begin{aligned} k_{st} &= k_0 \left(k_0 = 100\right) \\ c_d &= c_0 \left(c_0 = 2 \times 0.707 \times 10\right) \end{aligned} \tag{5.35}$$

Non-Linear Flexible Gluing Case:

$$\begin{aligned} k_{st} &= k_0 + k_1 q^1_{fl} \left(k_0 = 100, k_1 = 1000\right) \\ c_d &= c_0 + c_c \dot{q}^1_{fl} \left(c_0 = 2 \times 0.707 \times 10, c_1 = 2 \times 0.707 \times 10^4\right) \end{aligned} \tag{5.36}$$

For the guidelines for selecting Flexible Gluing parameters in the general case Zeid [92] proposed setting the parasitic spring stiffnesses by the largest imaginary part of the eigenvalues of the system with coupling springs was computed as a function of the parasitic stiffnesses, which were then set so that the eigenvalues were ten times the highest natural frequency of the original system.

The equations of motion for all-at-once model are written as

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{Bmatrix} \ddot{u}_1 \\ \ddot{u}_2 \\ \ddot{u}_3 \end{Bmatrix} + \begin{bmatrix} c_d & -c_d & 0 \\ -c_d & c_d & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \dot{u}_3 \end{Bmatrix} \begin{bmatrix} k + k_{st} & -k_{st} & 0 \\ -k_{st} & k + k_{st} & -k \\ 0 & -k & k \end{bmatrix} \begin{Bmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \dot{u}_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ F \end{Bmatrix} \tag{5.37}$$

Each subsystem, flexible gluing joint and all-at-once model are solved using the same time integration method, namely the well-known trapezoidal integration rule and a Newton-Raphson iteration method is employed for non-linear case. The tolerance for convergence is 1e-10 and a time step size of 1e-2 was selected. Figures 5.8 and 5.9 show comparisons of the results obtained using flexible gluing simulations with the results of all-at-once simulation for the linear flexible gluing case. Figures 5.10 and 5.11 show comparisons of the results obtained for the non-linear flexible gluing case. The gluing simulations are in good agreement with the all-at-once simulation results.

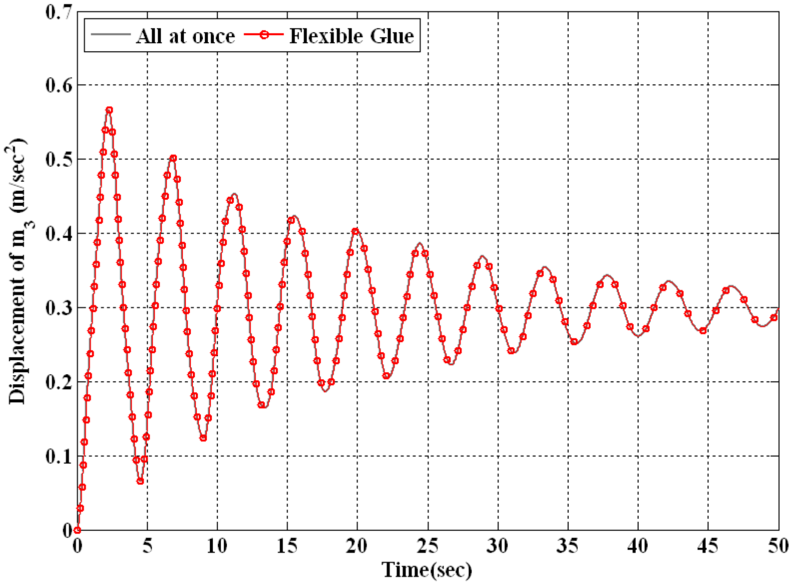


Figure 5.8 Displacement of m_3 (linear flexible gluing)

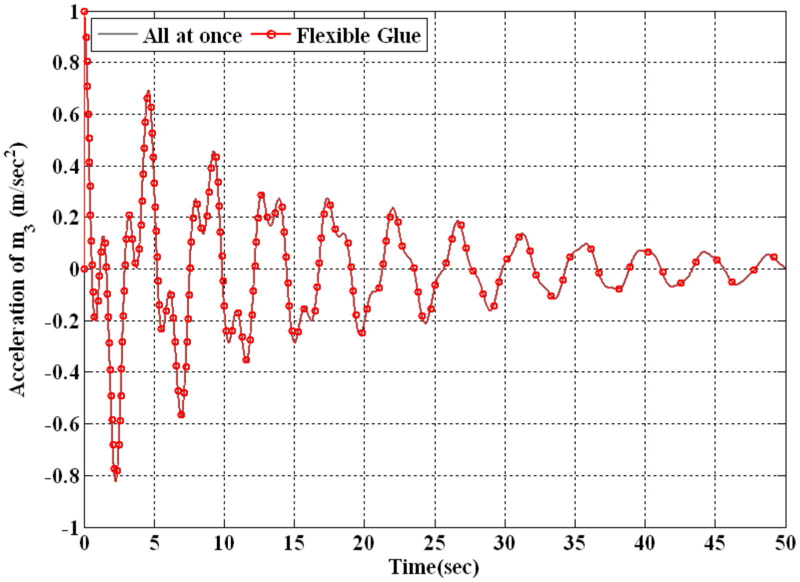


Figure 5.9 Acceleration of m_3 (linear flexible gluing)

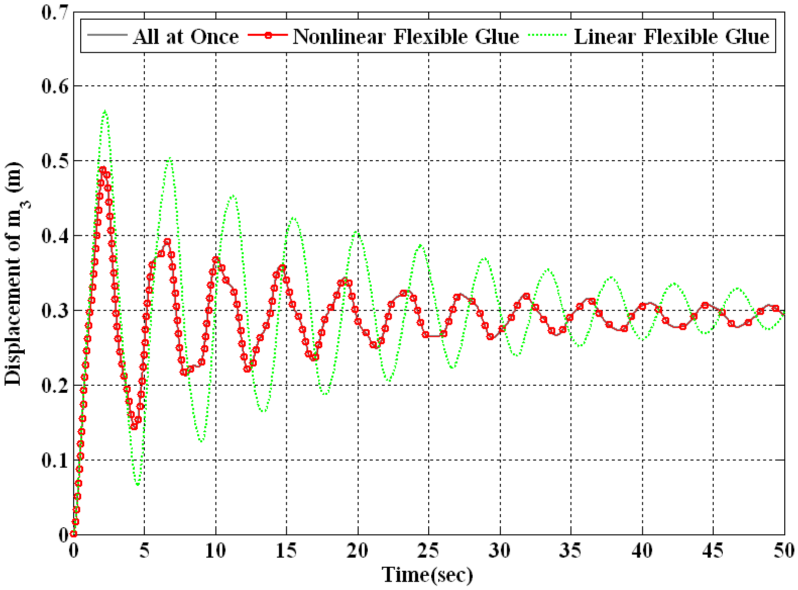


Figure 5.10 Displacement of m_3 (Non-linear flexible gluing)

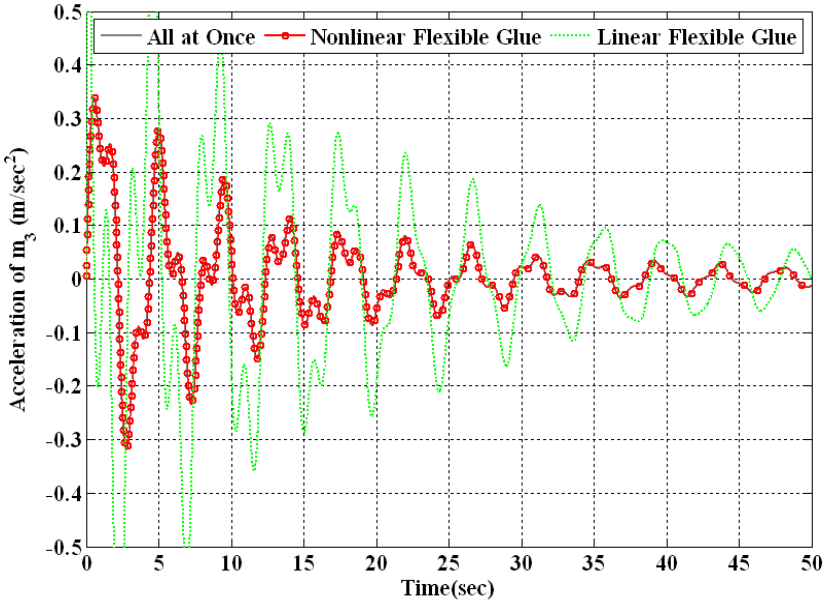


Figure 5.11 Acceleration of m_3 (Non-linear flexible gluing)

5.3.2 Planar Four-Bar Link Mechanism with A Flexible Gluing Joint

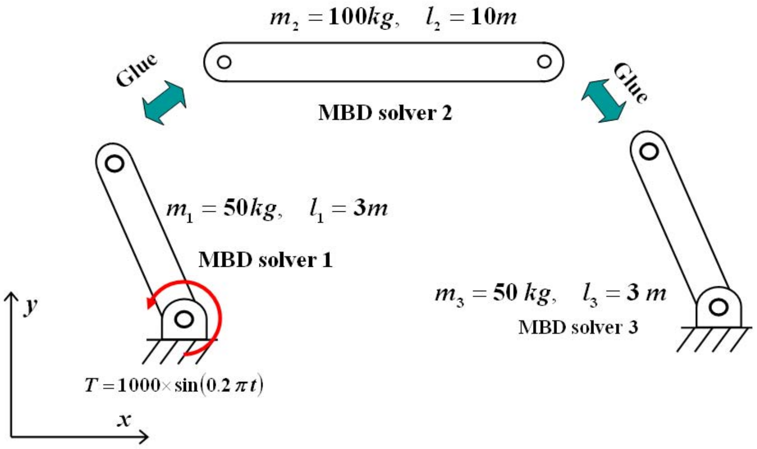

$m_2 = 100kg, \quad l_2 = 10m$
Glue
Glue
MBD solver 2
$m_1 = 50kg, \quad l_1 = 3m$
MBD solver 1
$m_3 = 50\ kg, \quad l_3 = 3\ m$
MBD solver 3
y
x
$T = 1000 \times \sin(0.2\pi t)$

Figure 5.12 Planar Four-Bar Link Mechanism

In this example, a planar four-bar link mechanism is considered for verification of the flexible gluing algorithm for simulating a multibody dynamics problem. Figure 5.12 depicts the four-bar link model separated into three subsystems. The original T-T gluing algorithm dealing with fixed type gluing may fail to pass the singular point of a mechanism, as encountered in previous research [17], when all the bars lie on the same line. In these singular points in multibody dynamics problems, the interface forces (constraint forces at joints) are ideally of infinite value to satisfy the compatibility condition (constraint equations) because all subsystems are considered as perfectly rigid body and none of the joints has compliance. Commercial multibody dynamics program such as ADAMS can deal with these kind of singular points during solving multibody dynamics models by intentionally changing the simulation time increment in order to quickly pass over the singular points within a specified convergence error [89]. Because the real subsystem components are not perfectly rigid bodies and real joint elements have compliance, it is necessary to consider component and joint flexibility for good accuracy and good convergence of simulation results. Instead, using the proposed flexible gluing algorithm, one can deal with singular points in multibody dynamic problem and more accurately predict joint element dynamic behaviors.

As shown Figure 5.12, the separated subsystem parts are modeled as rigid bodies and two interfaces are defined as gluing points, where each part is simulated using the Matlab solver(ode45, ode15s). The simulation results of each part are glued together at 0.01 time increments. The parameter values of both flexible joints are $k_{st} = 10^6 (N/m)$ and $c_d = 2 \cdot 0.707 \cdot 10^3 (N \cdot \sec/m)$. Table 5.1 presents a comparison of the total number of iterations and the number of times the gluing matrix was updated, where gluing matrix was updated if the iteration number exceeded 10, with different interface error tolerances. As shown in Table 5.1, the flexible gluing algorithm provides better convergence than the fixed type gluing algorithm. Figures 5.13 and 5.14 plot the comparisons of interface forces and angular velocities of first link. As shown in Figure 5.13, the peaks generated at the singular point and interface force oscillation remain and effect the simulation results after the singular point for the fixed type gluing, but the flexible gluing result show the oscillation is dissipated after the singular points are passed.

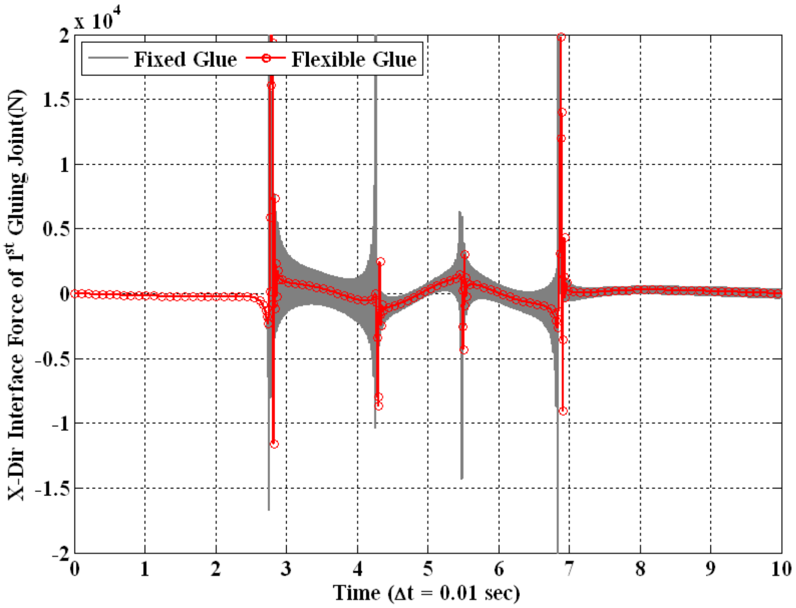


Figure 5.13 Comparison of interface forces at the 1st joint of fixed and flexible type results

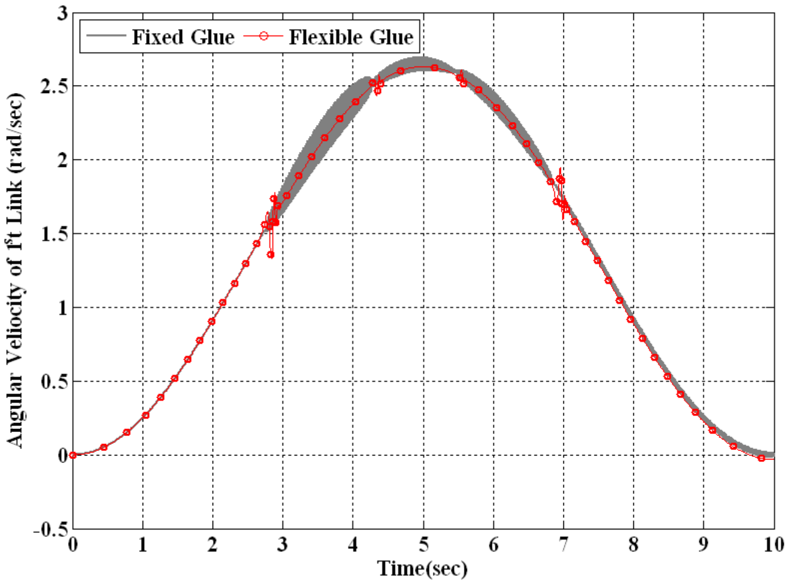


Figure 5.14 Comparison of angular velocity of the 1st link of fixed and flexible type results

Table 5.1 Comparison of fixed type gluing and flexible type gluing algorithm results

Interface Error Tolerance	Gluing Matrix Updating Number		Total Iteration Number	
	Fixed Glue	Flexible Glue	Fixed Glue	Flexible Glue
1e-7	**-**	**0**	**-**	**2616**
1e-8	**16**	**0**	**4935**	**3016**
1e-9	**18**	**0**	**5217**	**3457**
1e-10	**-**	**0**	**-**	**3943**

(- : failure to pass singular points)

5.3.3 Four-Bar Crank Slider Glued with Penalized Method.

This example employs the same model used for integration of multibody dynamic and finite element simulation codes from Chapter 4. In the case of coupling MBD and FEM, it is difficult to satisfy all kinematic variables simultaneously at the interface, since each set of simulation results at the interface may be computed with different simulation schemes.

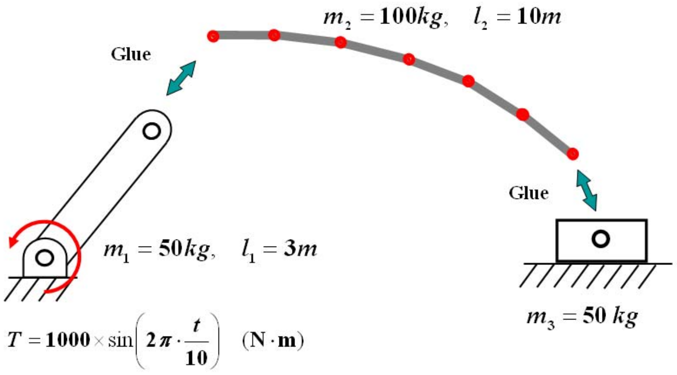

$m_2 = 100kg, \quad l_2 = 10m$
Glue
Glue
$m_1 = 50kg, \quad l_1 = 3m$
$m_3 = 50\ kg$
$T = 1000 \times \sin\left(2\pi \cdot \frac{t}{10}\right) \quad (\mathrm{N \cdot m})$

Figure 5.15 Four-bar crank slider distributed simulation model

In the subsystem parts depicted in Figure 5.15, the rigid link is solved using Matlab (ode45) and the flexible link is solved using the PIM method of the previous chapter. If one integrates distributed subsystems with original T-T gluing algorithm using acceleration information to check the compatibility condition at the interface, other variables such as velocity and displacement may be not satisfied the compatibility condition. Figure 5.15 shows the interface error of acceleration and Figure 5.15 shows displacement error for different gluing parameters. In the previous gluing algorithm, only one class of variable such as acceleration or displacement could be used for the checking compatibility condition. As shown Figure 5.16 and 5.17, the errors of compatibility conditions are more stable when using the penalized compatibility condition.

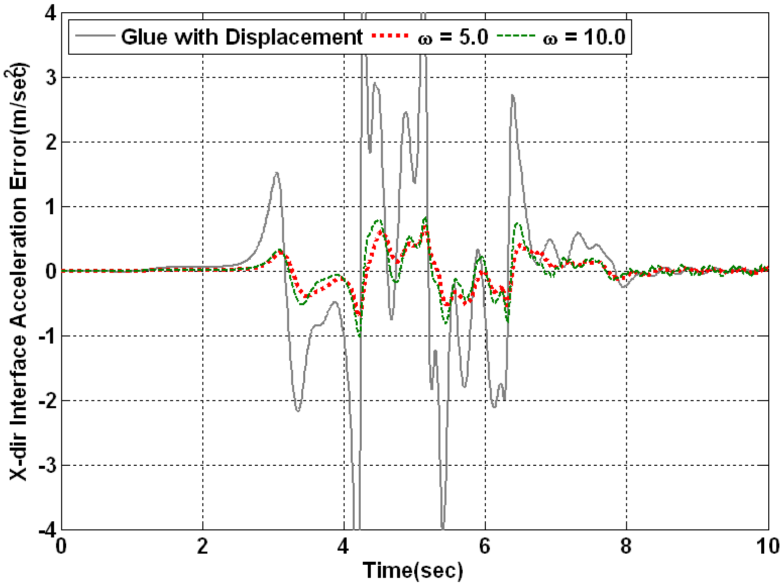


Figure 5.16 Comparison of acceleration error at the interface between 1st & 2nd Link

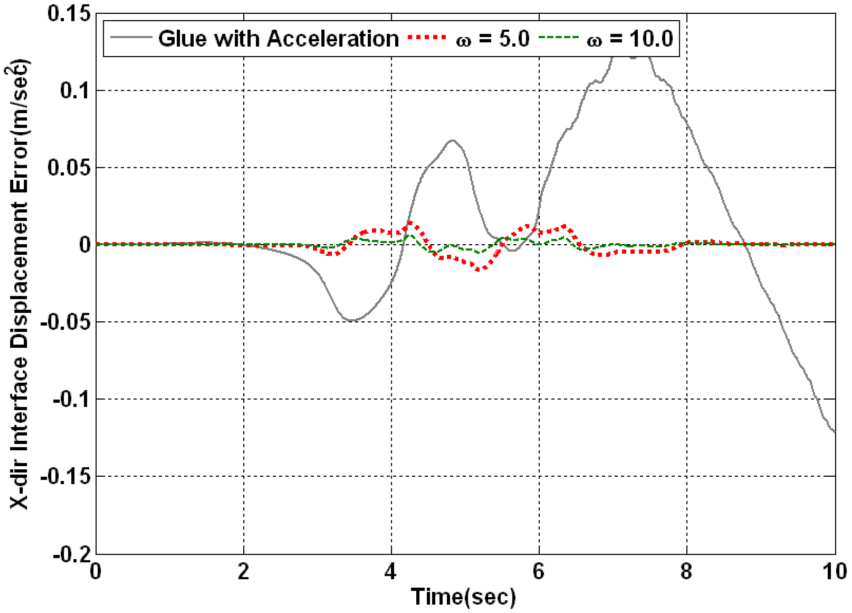


Figure 5.17 Comparison of displacement error at the interface between 1st & 2nd Link

5.3.4 Integration of Compliant Connectors with Flexible Gluing Algorithm

The flexible gluing connector was formulated for considering both linear and non-linear springs and dampers. Assumption of linear response of connectors such as rubber bushings, mounst and other joints or connectors can be sufficient for many noise and vibration simulations since the whole simulation model is assumed linear. However, for specially focused simulations like durability and vibration in low frequency range, it is necessary that the rubber mounts be modeled with their non-linear properties. In this case, the assembled model should be solved with non-linear dynamics and/or nonlinear transient FE analysis. Using the non-linear flexible gluing connector within the distributed simulation platform, the component models can be computed simply with linear elastic solver. Thus, a nonlinear simulation can be computed efficiently in distributed simulation platform.

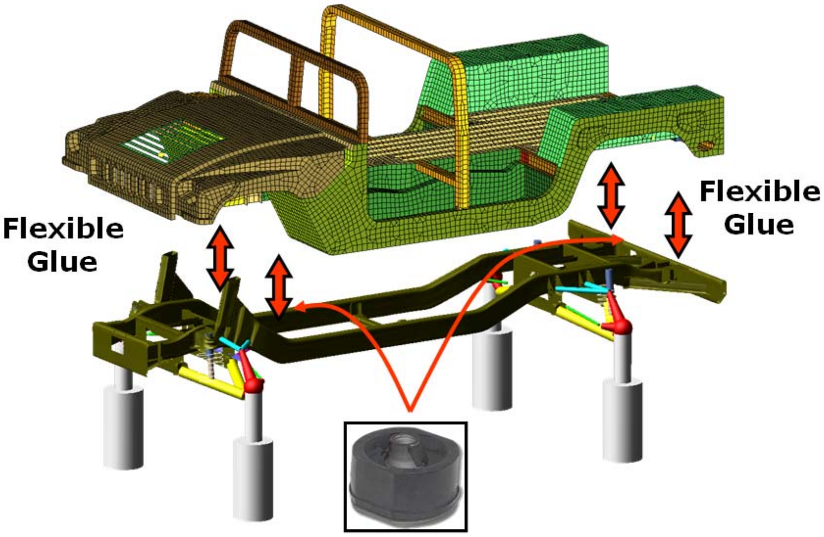


Figure 5.18 Gluing under body of MBD model and upper body of FE model with Flexible gluing algorithm

Figure 5.18 depicts a distributed simulation for the case of a multibody dynamics subsystem model for frame and suspension system on a four post shaker system and a Finite Element model of a HMMWV upper body, which is same model used in integrating rigid body and linear elastic deformed body in the previous chapter. The four-post excitation inputs are also same with the previous verification example 4.3.2. The interfaces of gluing algorithm are set at the body mounts positions between the upper body and frame. A flexible gluing algorithm is used to model the rubber bushing components at the body mount, where the parameters of Equation (5.36) are chosen as

$$\begin{array}{ll} k_0 = 3.0e5 & k_1 = 1.0e4 \\ c_0 = 1.5e3 & c_1 = 1.0e2 \end{array}$$

These,parameters were arbitrarily selected to show the effect of body mountswithin the flexible joint. Figure 5.19 plots the interface forces at the front body mount compared with fixed type gluing results. We see the effect of flexible gluing algorithm in the interface force, which can make different dynamic response in the flexible subsystem components.

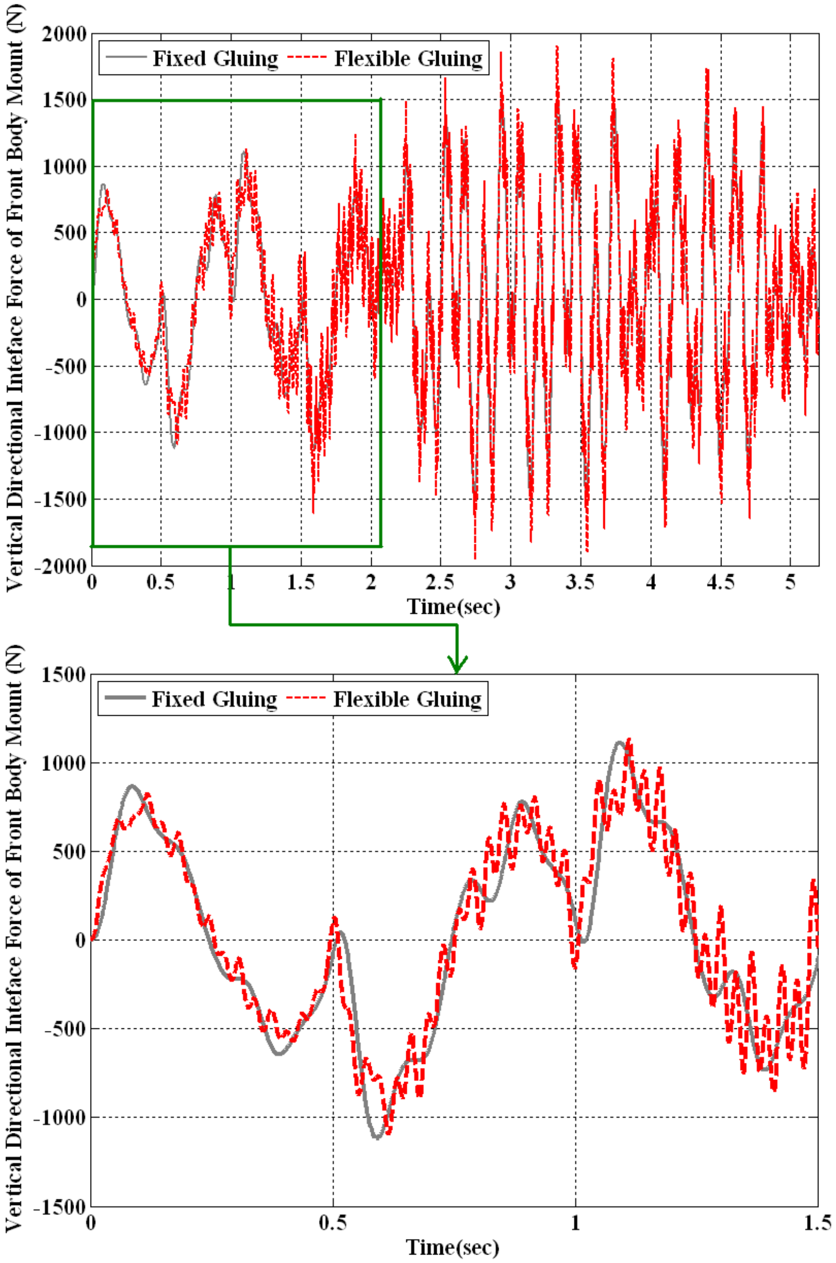


Figure 5.19 Comparison of vertical directional interface forces at front body mount

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSIONS

The need for a distributed simulation methodology to integrate heterogeneous simulation models, which are models for a general multibody dynamic (MBD) simulation, linear/nonlinear Finite Element (FE) simulation, is addressed in this dissertation. Integration of multibody dynamic analysis and Finite Element analysis using a Partitioned Iteration Method (PIM) is demonstrated in the Distributed Simulation Platform, so that the distributed simulation can be improved in order to solve more general problems in engineering.

First, The Partitioned Iteration Method addresses the motion of flexible bodies represented by coupling FE and MBD models and codes. The PIM is an efficient method to embed in a distributed simulation environment because the deformation of body can be solved using a linear elastic FE solver. This method is formulated employing the mean axis reference condition and also a C.G. following reference frame with an iteration scheme. The reference motion and the flexible body are solved using separate solvers. This approach overcomes the weakness of linear elastodynamics, which is based on the assumption that the effect of flexible body deformation on the reference motion can be neglected. Examples were given that demonstrate good correlation between the PIM and the floating frame of reference approach.

Second, integration of subsystem models, such as multibody dynamics subsystems models and finite element subsystems models for practical engineering problems, e.g., detailed durability or NVH simulation, was demonstrated with seamlessly integrated simulation and design tasks in a distributed computing environment. Ideally, a practical distributed simulation environment should allow using existing commercial packages, including a combination of multibody dynamics codes, such as MSC/ADAMS, and finite element codes, such as FEAP, which has been wrapped in the demonstration simulation platform without modifying their solvers and user interfaces. Rigid body motion of the overall system (or a subsystem) should be solved with the numerical integrators in a multibody dynamics code, while deformation of each individual component should be solved with the solvers in an existing finite element code. This coupling method of FEM and MBD can be easily extended to consider nonlinear and large deformation of the flexible body and other nonlinearities in the flexible multibody systems and has great promise for practical engineering use.

Third, the key component to realize the distributed simulation is the gluing algorithm. Improved gluing algorithms were proposed for assembling the subsystems models with general rigid or flexible connections. The expressing methods of the flexible joint effects in D-Sim are addressed in this study. The flexible joints embedded in gluing algorithm can improve the accuracy of the simulation to represent the real physical system and the convergence at singular points in the multibody dynamic simulation. A distributed simulation can provide an accurate prediction of the dynamic loads that act on the subsystem components at the gluing interface because these loads feed directly into the fatigue life prediction of the components.

Overall, the main contributions of this research include:

1. Partitioned Iteration Method for coupling FE and MBD in D-Sim
   A. Formulation of general equation of motion of FE flexible body including mean axis reference condition.
   B. Separation of flexible body formulation in the floating frame of reference into rigid motion and flexible body motion.
   C. Flexible body is solved with simulation model already built for the linear elastic simulation.
   D. The quadratic velocity correctors in rigid and flexible motion are updated with iteration.
2. Demonstration of Integration MBD and FE within D-Sim
   A. Application to three dimensional elements- beam, shell
   B. Demonstration of coupling MBD and FE with vehicle simulation model
3. Linear/Nonlinear Flexible Gluing Connector are added to previous Gluing Algorithm
   A. Formulation of flexible gluing matrix with linear stiffness and/or damping properties
   B. Validation of Flexible gluing connector with upper body and main frame of HMMWV
4. Penalized Method using Baumgarte Stabilizaton

6.2 FUTURE WORK

To apply the proposed methodology to solve practical engineering problems, improvements should be made in both the academic research and the engineering implementation.

6.2.1 Improve Gluing Algorithm

The developed T-T gluing algorithm has been applied to the integration of different simulation models, including both finite element and multibody dynamics models and has achieved satisfactory results. However, to apply the gluing algorithm to a very general simulation problem, convergence, accuracy and efficiency will be needed to exploit the full potentials of the gluing algorithm. The integration of individual and high fidelity simulation models together in a distributed way to study the behaviors of complex systems will greatly benefit both the science and the engineering fields.

The flexible gluing algorithm and penalized method in Chapter 5, are examples of improving gluing algorithm to relax the compatibility condition and to give numerical damping effects for convergence at special range during distributed simulation. In the view of numerical efficiency, it has no limitation to use distributed computer resources, however, their is a disadvantage since iterative scheme is used to obtain the converged simulation results. Thus, it can be a good study issue to develop integration method without trial simulation for finding interface variables.

6.2.2 Improve Partitioned Iteration Method

The methods of large rotation and displacement problem which are floating frame of reference, linear elastodynamics and partitioned iteration method are presented in

Chapter 3. The Partitioned Iteration Method facilitates the integration of finite element models and multibody dynamics models in the D-Sim Platform since the reference motion and the deformation of flexible body are separately simulated in the distributed independent solver, and also the flexible body models already constructed for the linear elastic simulation can be used.

The partitioned iteration method uses iteration, which might have a relative high computing cost in the case of solving a flexible body having a large degree of freedom. The improvements of iteration procedure to obtain the coupling forces (the quadratic velocity terms or gyro effects) between the relative deformed body motion and the reference motion can be another issue to develop an efficient computational method to solving flexible body in D-Sim.

6.2.3 Extend for the Integration Other Problems

Presently the design process in industry is a very complex optimization task often involving multi-disciplines, multi-objectives, and computationally intensive processes for product simulation. Many researchers deal with the coupling of specialized tools by the means of interfaces. Interfaces between the codes have to be developed for considering the nature of the description of the physical model, numerical properties of the respective simulation methods, and software and hardware implementation issues. The analysis of complex product is generally concerned with multi-disciplinary modules exchanging physical parameters. The current D-Sim can deal only with the interface parameters of forces and kinematic quantities. For a commercialized version of D-Sim, the platform has to provide valuable integration tools to support distributed collaborative design tasks.

For example, in the structural dynamic problems induced by friction or contact and heat transfer problems, we have to consider the exchanging parameters of interfaces and various boundaries of interfaces.

APPENDICES

ROTATIONAL MATRIX

FINITE ROTATION

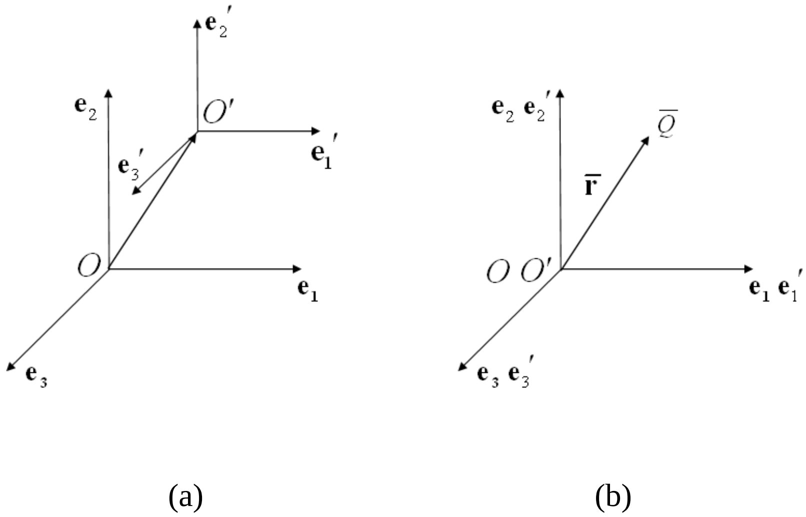
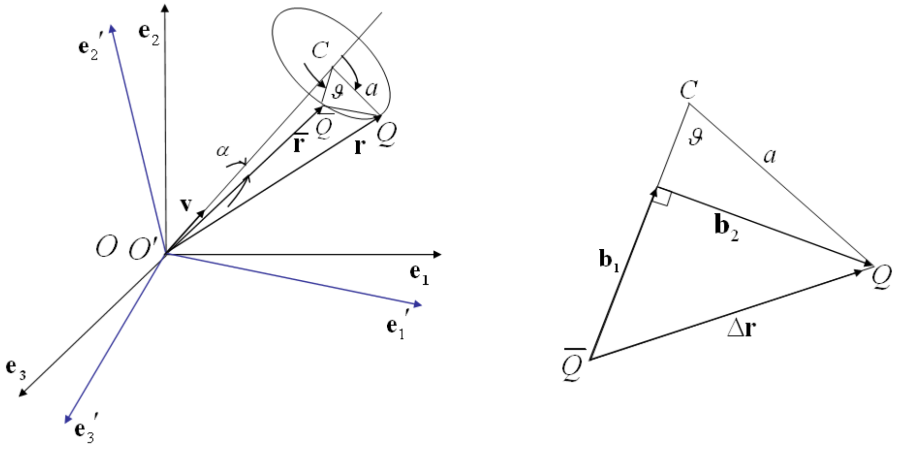


Figure A.6.1 Coordinates systems

In multibody systems, the components may undergo large relative translational and rotational displacements. To define the configuration of a body in the multibody system space, one must be able to determine the location of every point on the body respect to a selected inertial fame of reference. A body reference in which the position vectors of the material points can be described. The position vectors of these points can then be found in other coordinate systems by defining the relative position and orientation of the body coordinate system with respect to other coordinate systems. As shown in Figure A.1-(a), three variables define the relative translational motion between two coordinate systems. This relative translational motion can be measured by the position vector of the origin O' of the coordinate system $\mathbf{e}_1', \mathbf{e}_2', \mathbf{e}_3'$ with respect to coordinate system $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$.

To develop the transformation that defines the relative orientation between two coordinate systems between $\mathbf{e}_1', \mathbf{e}_2', \mathbf{e}_3'$ and $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$, we first assume that the origins of the two coordinate systems coincide as shown in Figure A.1-(b). We also assume that the axes of these two coordinates systems initially parallel. Let the vector $\bar{\mathbf{r}}$ be the position vector of point $\bar{Q}$ whose coordinates are assumed to be fixed in the $\mathbf{e}_1', \mathbf{e}_2', \mathbf{e}_3'$ coordinate system. Let the reference $\mathbf{e}_1', \mathbf{e}_2', \mathbf{e}_3'$ rotate an angle θ about axis OC as shown in the Figure A.2-(a).



(a) (b)

Figure A.6.2 Finite rotations

As the result of this rotation, point $\bar{Q}$ is translated to point Q. The position vector of point Q in the $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ coordinate system is denoted by $\mathbf{r}$. The change of point $\bar{Q}$ due to the rotation θ is defined by the vector $\Delta\mathbf{r}$ as shown in the Figure A.2-(b). new vector can be written as

$$\mathbf{r} = \bar{\mathbf{r}} + \Delta\mathbf{r} \tag{A.1}$$

The vector $\Delta\mathbf{r}$ can be written as the sum of the two vectors

$$\Delta\mathbf{r} = \mathbf{b}_1 + \mathbf{b}_2 \tag{A.2}$$

where the vector $\mathbf{b}_1$ is drown perpendicular to the plane $OC\bar{Q}$ and thus has a direction $(\mathbf{v} \times \bar{\mathbf{r}})$, where $\mathbf{v}$ is a unit vector along the axis of OC. The magnitude of vector $\mathbf{b}_1$ is given by

$$|\mathbf{b}_1| = a \sin\theta \tag{A.3}$$

From Figure A.2-(b), one can see that

$$a = |\bar{\mathbf{r}}| \sin\alpha = |\mathbf{v} \times \bar{\mathbf{r}}| \tag{A.4}$$

Therefore

$$\mathbf{b}_1 = a \sin\theta \frac{\mathbf{v} \times \bar{\mathbf{r}}}{|\mathbf{v} \times \bar{\mathbf{r}}|} = (\mathbf{v} \times \bar{\mathbf{r}}) \sin\theta \tag{A.5}$$

$$\mathbf{b}_2 = 2a \sin^2 \frac{\theta}{2} \cdot \frac{\mathbf{v} \times (\mathbf{v} \times \bar{\mathbf{r}})}{a} = 2[\mathbf{v} \times (\mathbf{v} \times \bar{\mathbf{r}})] \sin^2 \frac{\theta}{2} \tag{A.6}$$

Using Eqaution (A.1), one can write

$$\mathbf{r} = \bar{\mathbf{r}} + (\mathbf{v} \times \bar{\mathbf{r}}) \sin\theta + 2[\mathbf{v} \times (\mathbf{v} \times \bar{\mathbf{r}})] \sin^2 \frac{\theta}{2} \tag{A.7}$$

Using skew symmetric matrices of $\mathbf{v}$ define as

$$\tilde{\mathbf{v}} = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \tag{A.8}$$

Equation (A.7) can be rewritten as:

$$\begin{aligned} \mathbf{r} &= \bar{\mathbf{r}} + \tilde{\mathbf{v}}\bar{\mathbf{r}}\sin\theta + 2(\tilde{\mathbf{v}})^2\bar{\mathbf{r}}\sin^2\frac{\theta}{2} \\ &= \left[\mathbf{I} + \tilde{\mathbf{v}}\sin\theta + 2(\tilde{\mathbf{v}})^2\sin^2\frac{\theta}{2}\right]\bar{\mathbf{r}} \end{aligned} \tag{A.9}$$

where $\mathbf{I}$ is a 3×3 identity matrix. Equation (A.9) can be written as

$$\mathbf{r} = \mathbf{A}\bar{\mathbf{r}} \tag{A.10}$$

where $\mathbf{A} = \mathbf{A}(\theta)$ is the 3×3 rotation matrix given by

$$\mathbf{A} = \left[\mathbf{I} + \tilde{\mathbf{v}}\sin\theta + 2(\tilde{\mathbf{v}})^2\sin^2\frac{\theta}{2}\right] \tag{A.11}$$

This rotation matrix, referred to as the *Rodriquez-formula*, is expressed in terms of the angle of rotation and a unit vector along the axis of rotation.

EULER PARAMETERS

Using the trigonometric identity

$$\sin\theta = 2\sin\frac{\theta}{2}\cdot\cos\frac{\theta}{2} \qquad \text{(A.12)}$$

one can rewrite the transformation matrix of Equation (A.11) as

$$\mathbf{A} = \mathbf{I} + 2\tilde{\mathbf{v}}\sin\frac{\theta}{2}\left(\mathbf{I}\cos\frac{\theta}{2} + \tilde{\mathbf{v}}\sin\frac{\theta}{2}\right) \qquad \text{(A.13)}$$

The transformation matrix of Equation (A.13) can be expressed in terms of the following four *Euler parameters*:

$$\left.\begin{array}{ll} \theta_0 = \cos\dfrac{\theta}{2} & \theta_1 = v_1\sin\dfrac{\theta}{2} \\ \theta_2 = v_2\sin\dfrac{\theta}{2} & \theta_3 = v_3\sin\dfrac{\theta}{2} \end{array}\right\} \qquad \text{(A.14)}$$

in which v_1, v_2 and v_3 are the components of the unit vectors $\mathbf{v}$. If one defines with Euler parameter as $\bar{\boldsymbol{\theta}} = \begin{bmatrix}\theta_1 & \theta_2 & \theta_3\end{bmatrix}^T$, the transformation matrix $\mathbf{A}$ can be written as

$$\mathbf{A} = \mathbf{I} + 2\tilde{\bar{\boldsymbol{\theta}}}\left(\mathbf{I}\theta_0 + \tilde{\bar{\boldsymbol{\theta}}}\right) \qquad \text{(A.15)}$$

where the four Euler parameters given by (A.14) satisfy the relation

$$\sum_{k-0}^{3}(\theta_k)^2 = \boldsymbol{\theta}^T\boldsymbol{\theta} = 1 \qquad \text{(A.16)}$$

where $\boldsymbol{\theta}$ is the vector as

$$\boldsymbol{\theta} = \begin{bmatrix}\theta_0 & \theta_1 & \theta_2 & \theta_3\end{bmatrix}^T \qquad \text{(A.17)}$$

The transformation matrix $\mathbf{A}$ can be written explicitly in terns of the four Euler parameters of Equation (A.14) as

$$\mathbf{A} = \begin{bmatrix} 1-2(\theta_2)^2-2(\theta_3)^2 & 2(\theta_1\theta_2-\theta_0\theta_3) & 2(\theta_1\theta_3+\theta_0\theta_2) \\ 2(\theta_1\theta_2+\theta_0\theta_3) & 1-2(\theta_1)^2-2(\theta_3)^2 & 2(\theta_2\theta_3-\theta_0\theta_1) \\ 2(\theta_1\theta_3-\theta_0\theta_2) & 2(\theta_2\theta_3+\theta_0\theta_1) & 1-2(\theta_1)^2-2(\theta_2)^2 \end{bmatrix} \quad \text{(A.18)}$$

Using the identity of Equation (A.16), an alterantive form of the transformation matrix can be obtained as

$$\mathbf{A} = \begin{bmatrix} 2[(\theta_0)^2+(\theta_1)^2]-1 & 2(\theta_1\theta_2-\theta_0\theta_3) & 2(\theta_1\theta_3+\theta_0\theta_2) \\ 2(\theta_1\theta_2+\theta_0\theta_3) & 2[(\theta_0)^2+(\theta_2)^2]-1 & 2(\theta_2\theta_3-\theta_0\theta_1) \\ 2(\theta_1\theta_3-\theta_0\theta_2) & 2(\theta_2\theta_3+\theta_0\theta_1) & 2[(\theta_0)^2+(\theta_3)^2]-1 \end{bmatrix} \quad \text{(A.19)}$$

Note that Euler parameters does not depend on the components of the vector $\overline{\mathbf{r}}$. It depend on only on the components of the unit vector $\mathbf{v}$ along the axis of rotation as well as the angle of rotation θ.

RODRIGUEZ PARAMETERS

The transformation matrix (A.11) developed in the preceding section is expressed in terms of four parameters, that is, one more than the number of degrees of freedom. In this section, an alterantive representation, which uses three parameters called *Rodriguez parameters*, is developed

For convenience, we reproduced the transformation matrix of Equation (A.11), and now denfine the vector γ of Rodriguez parameters as

$$\boldsymbol{\gamma} = \mathbf{v} \tan\frac{\theta}{2} \tag{A.20}$$

that is

$$\gamma_1 = v_1 \tan\frac{\theta}{2}, \quad \gamma_2 = v_1 \tan\frac{\theta}{2}, \quad \gamma_3 = v_3 \tan\frac{\theta}{2} \tag{A.21}$$

Note that the Rodriguez parameter representation has the disadvantage of becoming infinite when the angle of rotation θ is equal to π. Using the trigonometric identity

$$\begin{gathered} \sin\theta = 2\sin\frac{\theta}{2}\cdot\cos\frac{\theta}{2}, \quad \sin\frac{\theta}{2} = \tan\frac{\theta}{2}\cdot\cos\frac{\theta}{2} \\ \sec^2\frac{\theta}{2} = 1 + \tan^2\frac{\theta}{2} \end{gathered} \tag{A.22}$$

one can write $\sin\theta$ as

$$\sin\theta = 2\sin\frac{\theta}{2}\cdot\cos\frac{\theta}{2} = 2\tan\frac{\theta}{2}\cdot\cos^2\frac{\theta}{2} = \frac{2\tan(\theta/2)}{1+\tan^2(\theta/2)} \tag{A.23}$$

Since $\mathbf{v}$ is unit vector, one has

$$\boldsymbol{\gamma}^T\boldsymbol{\gamma} = \tan^2\frac{\theta}{2} \tag{A.24}$$

Therefore, Equation (A.23) leads to

$$\sin\theta = \frac{2\tan(\theta/2)}{1+\boldsymbol{\gamma}^T\boldsymbol{\gamma}} \tag{A.25}$$

Similary,

$$\sin^2\frac{\theta}{2} = \frac{\tan^2(\theta/2)}{1+\boldsymbol{\gamma}^T\boldsymbol{\gamma}} \tag{A.26}$$

Substituting Eqaution (A.25) and (A.26) into (A.13) yields

$$\mathbf{A} = \mathbf{I} + \frac{2}{1+\boldsymbol{\gamma}^T\boldsymbol{\gamma}}\left(\tilde{\mathbf{v}}\tan\frac{\theta}{2} + \tilde{\mathbf{v}}^2\tan^2\frac{\theta}{2}\right) \tag{A.27}$$

which, on using Equation (A.20), yields

$$\mathbf{A} = \mathbf{I} + \frac{2}{1+\boldsymbol{\gamma}^T\boldsymbol{\gamma}}\left(\tilde{\boldsymbol{\gamma}} + \tilde{\boldsymbol{\gamma}}^2\right) \tag{A.28}$$

In a more explicit form, the transformation matrix **A** can be written in terms of the three parameters as

$$\mathbf{A} = \frac{1}{1+\boldsymbol{\gamma}^T\boldsymbol{\gamma}}\begin{bmatrix} 1+\gamma_1^2-\gamma_2^2-\gamma_3^2 & 2(\gamma_1\gamma_2-\gamma_3) & 2(\gamma_1\gamma_3+\gamma_2) \\ 2(\gamma_1\gamma_2+\gamma_3) & 1-\gamma_1^2+\gamma_2^2-\gamma_3^2 & 2(\gamma_2\gamma_3-\gamma_1) \\ 2(\gamma_1\gamma_3-\gamma_2) & 2(\gamma_2\gamma_3+\gamma_1) & 1-\gamma_1^2-\gamma_2^2+\gamma_3^2 \end{bmatrix} \tag{A.29}$$

Using the definition of Euler parameter, one can show that Rodriguez parameters can be written in terms of Euler parameters as follows

$$\left.\begin{array}{c} \gamma_1 = v_1\dfrac{\sin(\theta/2)}{\cos(\theta/2)} = \dfrac{\theta_1}{\theta_0} \\ \gamma_2 = \dfrac{\theta_2}{\theta_0}, \quad \gamma_1 = \dfrac{\theta_3}{\theta_0} \end{array}\right\} \text{ and } \theta_0 = \frac{1}{\sqrt{1+\boldsymbol{\gamma}^T\boldsymbol{\gamma}}} \tag{A.30}$$

EULER ANGLES

The three independent *Euler angles* is one of the most common and widely used parameters in describing reference orientations. Euler angles involve three successive rations about three axes that are not orthogonal in general, Euler angles, however, are not unique. To this end we carry out the transformation between two coordinate systems by means of three successive rotations performed in a given sequence. For instance, the coordinates system $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ and $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3$ initially be coincide. The sequence starts by rotating the system $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3$ and angle ϕ about $\mathbf{e}_3$ axis. The result of this rotation is shown in the Figure A.3-(a). Since Figure 2.13 shows a comparison of the results obtained using the gluing simulations with 1st and 2nd layered finite element analyses. Here, dynamic loads of $f = 100\sin(50\pi t)\ N$ are applied at the middle point of front cross member The result shown here is the displacement at a selected node at the middle of body floor(node 12093) along the vertical direction. It is seen that the gluing process induces no additional error (beyond the round-off errors) in 1st and 2nd layered distributed simulation results. is the angle of rotation in the plane $\mathbf{e}_1, \mathbf{e}_2$, we have

$$\boldsymbol{\xi} = \mathbf{D}_1\mathbf{e} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}\mathbf{e} \qquad \text{(A.31)}$$

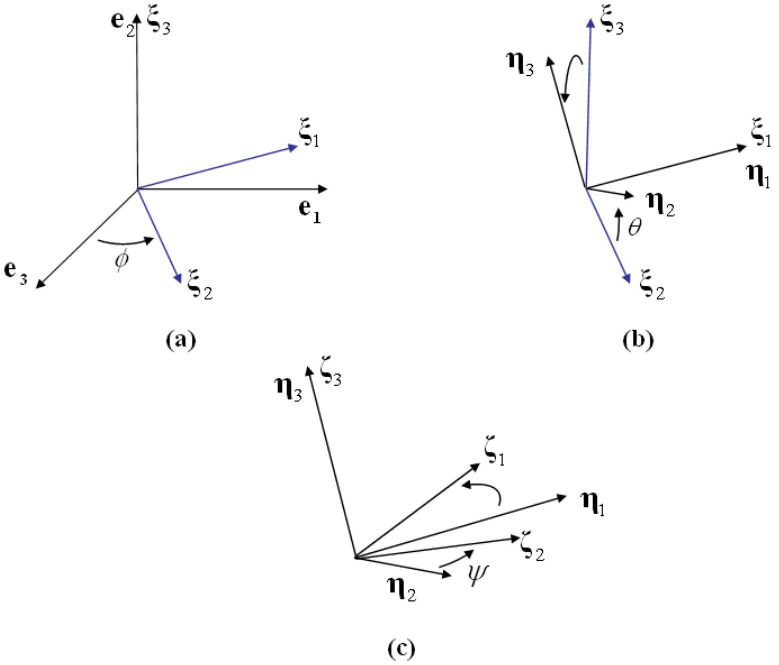


Figure A.6.3 Euler angle (3-1-3 Sequence)

Next, we consider coordinate system $\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \boldsymbol{\eta}_3$, which coincides with the system $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3$ and rotate this system an angle θ is in the plane Figure 2.13 shows a comparison of the results obtained using the gluing simulations with 1st and 2nd layered finite element analyses. Here, dynamic loads of $f = 100\sin(50\pi t)\ N$ are applied at the middle point of front cross member The result shown here is the displacement at a selected node at the middle of body floor(node 12093) along the vertical direction. It is seen that the gluing process induces no additional error (beyond the round-off errors) in 1st and 2nd layered distributed simulation results., we have

$$\boldsymbol{\eta} = \mathbf{D}_2\boldsymbol{\xi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \boldsymbol{\xi} \qquad \text{(A.32)}$$

Finally, we have consider the coordinate system $\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \boldsymbol{\zeta}_3$, which coincides with the system $\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \boldsymbol{\eta}_3$ and rotate this system an angle ψ is in the plane $\boldsymbol{\eta}_1, \boldsymbol{\eta}_2$, we have

$$\boldsymbol{\zeta} = \mathbf{D}_3\boldsymbol{\eta} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}\boldsymbol{\eta} \tag{A.33}$$

Using Equation (A.31), (A.32) and (A.33), one can write the transformation between the initial coordinate system $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ and the final system $\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \boldsymbol{\zeta}_3$ as follow:

$$\boldsymbol{\zeta} = \mathbf{D}_3\mathbf{D}_2\mathbf{D}_1\mathbf{e} = \mathbf{A}^T\mathbf{e} \tag{A.34}$$

Also, one can write

$$\mathbf{e} = \mathbf{A}\boldsymbol{\zeta} \tag{A.35}$$

where $\mathbf{A}$ is the transformation matrix

$$\mathbf{A} = \begin{bmatrix} \cos\psi\cos\phi - \cos\theta\sin\phi\sin\psi & -\sin\psi\cos\phi - \cos\theta\sin\phi\cos\psi & \sin\theta\sin\phi \\ \cos\psi\sin\phi + \cos\theta\cos\phi\sin\psi & -\sin\psi\sin\phi + \cos\theta\cos\phi\cos\psi & -\sin\theta\cos\phi \\ \sin\theta\sin\psi & \sin\theta\cos\psi & \cos\theta \end{bmatrix} \tag{A.36}$$

The three angles ϕ, θ and ψ are called the Euler angles. The matrix $\mathbf{A}$ is the the transformation matrix expressed in terms of Euler angles.

DYNAMIC EQUATION OF MOTION FOR LINEAR ELASTIC FINITE ELEMENTS MODEL FROM KINETIC ENERGY

In Chapter 3, the dynamic equation of motion of the flexible body is set using energy and virtual work expressions. The global position of an arbitrary point on i^{th} element of the deformable body is defined by using a coupled set of reference and elastic coordinates as

$$\mathbf{r}_i = \mathbf{R} + \mathbf{A}\bar{\mathbf{u}}_i = \mathbf{R} + \mathbf{A}\underline{\mathbf{N}}_i\mathbf{q}_f \tag{B.1}$$

The velocity vector can be written in partitioned form as

$$\dot{\mathbf{r}}_i = \begin{bmatrix} \mathbf{I} & -\mathbf{A}\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}} & \mathbf{A}\underline{\mathbf{N}}_i \end{bmatrix} \begin{bmatrix} \dot{\mathbf{R}} \\ \dot{\boldsymbol{\theta}} \\ \dot{\mathbf{q}}_f \end{bmatrix} \tag{B.2}$$

Then, we can write the coordinates of body with portioned form as

$$\mathbf{q}^T = \begin{bmatrix} \mathbf{R}^T & \boldsymbol{\theta}^T & \mathbf{q}_f^T \end{bmatrix} \tag{B.3}$$

Once the velocity vector of an arbitrary point on i^{th} element is defined, one can write the kinetic energy of body as

$$T = \sum_{i=1}^{N_e} \frac{1}{2} \int_{\Omega_e} \rho_i \dot{\mathbf{r}}_i^T \cdot \dot{\mathbf{r}}_i \, d\Omega_e \tag{B.4}$$

By substituting Equation (B.2) into Equation (B.4), we obtain

$$T = \frac{\mathbf{1}}{\mathbf{2}} \begin{bmatrix} \dot{\mathbf{R}}^T & \dot{\boldsymbol{\theta}}^T & \dot{\mathbf{q}}_f{}^T \end{bmatrix} \sum_{i=\mathbf{1}}^{N_e} \int_{\Omega_e} \rho_i \begin{bmatrix} \mathbf{I} & -\mathbf{A}\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}} & \mathbf{A}\underline{\mathbf{N}}_i \\ -\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\mathbf{A}^T & \bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}} & -\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\underline{\mathbf{N}}_i \\ \underline{\mathbf{N}}_i{}^T\mathbf{A}^T & -\underline{\mathbf{N}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}} & \underline{\mathbf{N}}_i{}^T\underline{\mathbf{N}}_i \end{bmatrix} d\Omega_e \begin{bmatrix} \dot{\mathbf{R}} \\ \dot{\boldsymbol{\theta}} \\ \dot{\mathbf{q}}_f \end{bmatrix} \tag{B.5}$$

For the equation of motion of deformable body, Lagrange's Equation takes the form in terms of partial derivative of variable $\mathbf{q}$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}}\right)^T - \left(\frac{\partial L}{\partial \mathbf{q}}\right)^T = \mathbf{F}^{ext} \tag{B.6}$$

where, $\mathbf{F}^{ext}$ is generalized external forces, L is Lagrangian such that $L = T - V$. If the deformed body can be assumed by a linear elastic deformable body, the potential energy V is same with the strain energy of body since the rigid body motion corresponds to the case of constant strains. In this dissertation, we consider a linear isotropic material. The more general case of nonlinear elastic, orthotropic material can also be formulated by changing the form of the body stiffness matrix. Using the virtual work principle, the virtual potential energy due to the elastic forces in the body can be written as

$$\delta V = \sum_{i}^{N_e} \int_{\Omega_e} \boldsymbol{\sigma}_i^T \delta\boldsymbol{\varepsilon}_i \, d\Omega_e \tag{B.7}$$

where $\boldsymbol{\sigma}_i$ and $\boldsymbol{\varepsilon}_i$ are, respectively, the stress and strain vectors of *i-th* element. The strain displacement relation in terms of nodal displacement vector in the following form

$$\boldsymbol{\varepsilon}_i = \underline{\mathbf{B}}_i \mathbf{q}_f \tag{B.8}$$

where, $\underline{\mathbf{B}}_i$ is a differentiated matrix of the shape function $\underline{\mathbf{N}}_i$. For a linear isotropic material, the constitutive equations relating the stress and strain can be written as

$$\boldsymbol{\sigma}_i = \mathbf{D}\boldsymbol{\varepsilon}_i \tag{B.9}$$

where $\mathbf{D}$ is athe symmetric elasticity matrix containing the appropriate material properties. Substituting Equation (B.8) into Equation (B.9) yields

$$\boldsymbol{\sigma}_i = \mathbf{D}\underline{\mathbf{B}}_i\mathbf{q}_f \tag{B.10}$$

in which the stress vector is written in terms of the nodal displacement vector. Substituting Equation (B.8) and (B.10) into Equation (B.7) yields

$$\delta V = \sum_i^{N_e} \int_{\Omega_e} \mathbf{q}_f^T \underline{\mathbf{B}}_i^T \mathbf{D}\underline{\mathbf{B}}_i \delta\mathbf{q}_f \, d\Omega_e \tag{B.11}$$

because $\mathbf{q}_f$ depends only on time, Equation (B.11) can be written as

$$\begin{aligned} \delta V &= \mathbf{q}_f^T \left[\sum_i^{N_e} \int_{\Omega_e} \underline{\mathbf{B}}_i^T \mathbf{D}\underline{\mathbf{B}}_i d\Omega_e \right] \delta\mathbf{q}_f \\ &= \mathbf{q}_f^T \mathbf{K}_{ff} \delta\mathbf{q}_f \end{aligned} \tag{B.12}$$

where $\mathbf{K}_{ff}$ is the symmetric positive definite stiffness matrix associated with the nodal displacement vector.

In the equation of motion of Lagrange's Equation (B.6) we take Lagrangian as the form in terms of partial derivative of variables $\mathbf{R}$, $\boldsymbol{\theta}$ and $\mathbf{q}_f$.

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}}\right) &= \left[\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{R}}}\right) \quad \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\boldsymbol{\theta}}}\right) \quad \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}_f}\right) \right] \\ &= \left[\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{R}}}\right) \quad \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\boldsymbol{\theta}}}\right) \quad \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{q}}_f}\right) \right] \end{aligned} \tag{B.13}$$

where $\frac{d}{dt}\left(\frac{\partial V}{\partial \dot{\mathbf{q}}}\right) = \mathbf{0}$ since the potential energy depends only on the nodal displacement as shown in Equation (B.12), and

$$\begin{aligned}\left(\frac{\partial L}{\partial \mathbf{q}}\right) &= \left[\left(\frac{\partial L}{\partial \mathbf{R}}\right) \quad \left(\frac{\partial L}{\partial \boldsymbol{\theta}}\right) \quad \left(\frac{\partial L}{\partial \mathbf{q}_f}\right)\right] \\ &= \left[\left(\frac{\partial T}{\partial \mathbf{R}}\right) \quad \left(\frac{\partial T}{\partial \boldsymbol{\theta}}\right) \quad \left(\frac{\partial T}{\partial \mathbf{q}_f} - \frac{\partial V}{\partial \mathbf{q}_f}\right)\right]\end{aligned} \tag{B.14}$$

the equation of motion of deformable body using Lagrange’s Equation can be rewritten as

$$\begin{aligned}\left[\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{R}}}\right)\right]^T &= \mathbf{F}_r^{\mathbf{ext}} \\ \left[\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\boldsymbol{\theta}}}\right) - \left(\frac{\partial T}{\partial \boldsymbol{\theta}}\right)\right]^T &= \mathbf{F}_\theta^{\mathbf{ext}} \\ \left[\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{q}}_f}\right) - \left(\frac{\partial T}{\partial \mathbf{q}_f} - \frac{\partial V}{\partial \mathbf{q}_f}\right)\right]^T &= \mathbf{F}_f^{\mathbf{ext}}\end{aligned} \tag{B.15}$$

where $\left[\mathbf{F}_r^{\mathbf{ext}T} \quad \mathbf{F}_\theta^{\mathbf{ext}T} \quad \mathbf{F}_f^{\mathbf{ext}T}\right]^T$ is generalized external force vector. The virtual work of all external forces acting on flexible body in compact form as

$$\delta W = \left[\mathbf{F}_r^{\mathbf{ext}T} \quad \mathbf{F}_\theta^{\mathbf{ext}T} \quad \mathbf{F}_f^{\mathbf{ext}T}\right]\delta \mathbf{r} = \left[\mathbf{F}_r^{\mathbf{ext}T} \quad \mathbf{F}_\theta^{\mathbf{ext}T} \quad \mathbf{F}_f^{\mathbf{ext}T}\right]\begin{bmatrix}\delta \mathbf{R} \\ \delta \boldsymbol{\theta} \\ \delta \mathbf{q}_f\end{bmatrix}$$

Then, one can obtain the each component in Equation (B.13). First, in terms of the postion $\mathbf{R}$ of body reference frame

$$\begin{aligned}\frac{\partial T}{\partial \dot{\mathbf{R}}} &= \frac{1}{2}\rho\sum_i^{N_e} \int_\Omega \frac{\partial\left(\dot{\mathbf{r}}_i^T \cdot \dot{\mathbf{r}}_i\right)}{\partial \dot{\mathbf{R}}} d\Omega \\ &= \frac{1}{2}\rho\sum_i^{N_e} \int_{\Omega_e} \left\{2\dot{\mathbf{R}}^T - 2\dot{\boldsymbol{\theta}}^T \bar{\mathbf{G}}^T \tilde{\bar{\mathbf{u}}}_i^T \mathbf{A}^T + 2\dot{\bar{\mathbf{u}}}_i^T \mathbf{A}^T\right\} d\Omega_e\end{aligned} \tag{B.16}$$

The derivative of Equation (B.16) with repect to time can be written as

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{R}}}\right)=\rho\sum_{i}^{N_e}\int_{\Omega_e}\left\{\ddot{\mathbf{R}}^T-\ddot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\mathbf{A}^T-\dot{\boldsymbol{\theta}}^T\dot{\bar{\mathbf{G}}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\mathbf{A}^T-\dot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\dot{\tilde{\bar{\mathbf{u}}}}_i{}^T\mathbf{A}^T\right.$$
$$\left.-\dot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\dot{\mathbf{A}}^T+\ddot{\mathbf{u}}_i{}^T\mathbf{A}^T+\dot{\mathbf{u}}_i{}^T\dot{\mathbf{A}}^T\right\}d\Omega_e \qquad \text{(B.17)}$$

using the relation of $\dot{\bar{\mathbf{G}}}\dot{\boldsymbol{\theta}}=0$, Equation (B.17) can be simple form as

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{R}}}\right)=\rho\sum_{i}^{N_e}\int_{\Omega_e}\left\{\ddot{\mathbf{R}}^T-\ddot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\mathbf{A}^T-2\dot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\dot{\tilde{\bar{\mathbf{u}}}}_i{}^T\mathbf{A}^T-\dot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\boldsymbol{\omega}}}^T\mathbf{A}^T+\ddot{\mathbf{u}}_i{}^T\mathbf{A}^T\right\}d\Omega_e$$

(B.18)

Second, the partial derivative of kinetic energy with repect to variable $\dot{\boldsymbol{\theta}}$ can be written as

$$\begin{aligned}\frac{\partial T}{\partial \dot{\boldsymbol{\theta}}}&=\frac{1}{2}\rho\sum_{i}^{N_e}\int_{\Omega_e}\frac{\partial\left(\dot{\mathbf{r}}^T\cdot\dot{\mathbf{r}}\right)}{\partial\dot{\boldsymbol{\theta}}}d\Omega_e\\&=\frac{1}{2}\rho\sum_{i}^{N_e}\int_{\Omega_e}\left\{-2\dot{\mathbf{R}}^T\mathbf{A}\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}+2\dot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}-2\dot{\mathbf{u}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}\right\}d\Omega_e\\&=\rho\sum_{i}^{N_e}\int_{\Omega_e}\left\{-\dot{\mathbf{R}}^T\mathbf{A}\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}+\dot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}-\dot{\mathbf{u}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}\right\}d\Omega_e\end{aligned} \qquad \text{(B.19)}$$

The derivative of Equation (B.19) with repect to time can be written as

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\boldsymbol{\theta}}}\right)&=\rho\sum_{i}^{N_e}\int_{\Omega_e}\left\{\begin{array}{l}-\ddot{\mathbf{R}}^T\mathbf{A}\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}-\dot{\mathbf{R}}^T\dot{\mathbf{A}}\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}-\dot{\mathbf{R}}^T\mathbf{A}\dot{\tilde{\bar{\mathbf{u}}}}_i\bar{\mathbf{G}}-\dot{\mathbf{R}}^T\mathbf{A}\tilde{\bar{\mathbf{u}}}_i\dot{\bar{\mathbf{G}}}\\+\ddot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}+\dot{\boldsymbol{\theta}}^T\dot{\bar{\mathbf{G}}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}+\dot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\dot{\tilde{\bar{\mathbf{u}}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}+\dot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\dot{\tilde{\bar{\mathbf{u}}}}_i\bar{\mathbf{G}}+\dot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\dot{\bar{\mathbf{G}}}\\-\ddot{\mathbf{u}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}-\dot{\mathbf{u}}_i{}^T\dot{\tilde{\bar{\mathbf{u}}}}_i\bar{\mathbf{G}}-\dot{\mathbf{u}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\dot{\bar{\mathbf{G}}}\end{array}\right\}d\Omega_e\\&=\rho\sum_{i}^{N_e}\int_{\Omega_e}\left\{\begin{array}{l}-\ddot{\mathbf{R}}^T\mathbf{A}\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}-\dot{\mathbf{R}}^T\dot{\mathbf{A}}\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}-\dot{\mathbf{R}}^T\mathbf{A}\dot{\tilde{\bar{\mathbf{u}}}}_i\bar{\mathbf{G}}-\dot{\mathbf{R}}^T\mathbf{A}\tilde{\bar{\mathbf{u}}}_i\dot{\bar{\mathbf{G}}}\\+\ddot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}+2\dot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\dot{\tilde{\bar{\mathbf{u}}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}+\dot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\dot{\bar{\mathbf{G}}}\\-\ddot{\mathbf{u}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}-\dot{\mathbf{u}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\dot{\bar{\mathbf{G}}}\end{array}\right\}d\Omega_e\end{aligned}$$

(B.20)

Third, the partial derivative of kinetic energy with repect to variable $\dot{\mathbf{q}}_f$ can be written

as

$$\frac{\partial T}{\partial \dot{\mathbf{q}}_{\mathbf{f}}} = \rho \sum_{i}^{N_e} \int_{\Omega_e} \left\{ \dot{\mathbf{R}}^T \mathbf{A} \underline{\mathbf{N}}_i - \dot{\boldsymbol{\theta}}^T \bar{\mathbf{G}}^T \tilde{\bar{\mathbf{u}}}_i{}^T \underline{\mathbf{N}}_i + \dot{\bar{\mathbf{u}}}_i{}^T \underline{\mathbf{N}}_i \right\} d\Omega_e \tag{B.21}$$

The time derivative of Equation (B.21) can be written as

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{q}}_{\mathbf{f}}} \right) &= \rho \sum_{i}^{N_e} \int_{\Omega_e} \Big\{ \ddot{\mathbf{R}}^T \mathbf{A} \underline{\mathbf{N}}_i + \dot{\mathbf{R}}^T \dot{\mathbf{A}} \underline{\mathbf{N}}_i - \ddot{\boldsymbol{\theta}}^T \bar{\mathbf{G}}^T \tilde{\bar{\mathbf{u}}}_i{}^T \underline{\mathbf{N}}_i \\ &\qquad - \dot{\boldsymbol{\theta}}^T \dot{\bar{\mathbf{G}}}^T \tilde{\bar{\mathbf{u}}}_i{}^T \underline{\mathbf{N}}_i - \dot{\boldsymbol{\theta}}^T \bar{\mathbf{G}}^T \dot{\tilde{\bar{\mathbf{u}}}}_i{}^T \underline{\mathbf{N}}_i + \ddot{\bar{\mathbf{u}}}^T \underline{\mathbf{N}}_i \Big\} d\Omega_e \\ &= \rho \sum_{i}^{N_e} \int_{\Omega_e} \left\{ \ddot{\mathbf{R}}^T \mathbf{A} \underline{\mathbf{N}}_i + \dot{\mathbf{R}}^T \mathbf{A} \tilde{\bar{\boldsymbol{\omega}}} \underline{\mathbf{N}}_i - \ddot{\boldsymbol{\theta}}^T \bar{\mathbf{G}}^T \tilde{\bar{\mathbf{u}}}_i{}^T \underline{\mathbf{N}}_i - \dot{\boldsymbol{\theta}}^T \bar{\mathbf{G}}^T \dot{\tilde{\bar{\mathbf{u}}}}_i{}^T \underline{\mathbf{N}}_i + \ddot{\bar{\mathbf{u}}}_i{}^T \underline{\mathbf{N}}_i \right\} d\Omega_e \end{aligned} \tag{B.22}$$

Next, one can obtain the each component in Equation (B.14), the partial derivative of Lagrangian with repect to $\mathbf{R}, \boldsymbol{\theta}$ and $\mathbf{q}_f$ can be written as

$$\frac{\partial T}{\partial \mathbf{R}} = 0 \tag{B.23}$$

For the derivative of kinetic energy with repect to variable $\boldsymbol{\theta}$, square norm of velocity can be expressed with respect to $\boldsymbol{\theta}$ as

$$\begin{aligned} \dot{\mathbf{r}}_i{}^T \cdot \dot{\mathbf{r}}_i = {} & \dot{\mathbf{R}}^T \dot{\mathbf{R}} - \dot{\mathbf{R}}^T \mathbf{A} \tilde{\bar{\mathbf{u}}}_i \bar{\mathbf{G}} \dot{\boldsymbol{\theta}} + \dot{\mathbf{R}}^T \mathbf{A} \dot{\bar{\mathbf{u}}}_i \\ & - \dot{\boldsymbol{\theta}}^T \bar{\mathbf{G}}^T \tilde{\bar{\mathbf{u}}}_i{}^T \mathbf{A}^T \dot{\mathbf{R}} + \dot{\boldsymbol{\theta}}^T \bar{\mathbf{G}}^T \tilde{\bar{\mathbf{u}}}_i{}^T \tilde{\bar{\mathbf{u}}}_i \bar{\mathbf{G}} \dot{\boldsymbol{\theta}} - \dot{\boldsymbol{\theta}}^T \bar{\mathbf{G}}^T \tilde{\bar{\mathbf{u}}}_i{}^T \dot{\bar{\mathbf{u}}}_i \\ & + \dot{\bar{\mathbf{u}}}_i{}^T \mathbf{A}^T \dot{\mathbf{R}} - \dot{\bar{\mathbf{u}}}_i{}^T \tilde{\bar{\mathbf{u}}}_i \bar{\mathbf{G}} \dot{\boldsymbol{\theta}} + \dot{\bar{\mathbf{u}}}_i{}^T \dot{\bar{\mathbf{u}}}_i \end{aligned} \tag{B.24}$$

where, the relation of $\bar{\mathbf{G}}\boldsymbol{\theta} = 0$, $\dot{\bar{\mathbf{G}}}\boldsymbol{\theta} + \bar{\mathbf{G}}\dot{\boldsymbol{\theta}} = 0$ leads to $\dot{\bar{\mathbf{G}}}\boldsymbol{\theta} = -\bar{\mathbf{G}}\dot{\boldsymbol{\theta}}$, Equation (B.24) becomes

$$\begin{aligned} \dot{\mathbf{r}}_i{}^T \cdot \dot{\mathbf{r}}_i = {} & \dot{\mathbf{R}}^T \dot{\mathbf{R}} + \dot{\mathbf{R}}^T \mathbf{A} \tilde{\bar{\mathbf{u}}}_i \dot{\bar{\mathbf{G}}} \boldsymbol{\theta} + \dot{\mathbf{R}}^T \mathbf{A} \dot{\bar{\mathbf{u}}}_i \\ & + \boldsymbol{\theta}^T \dot{\bar{\mathbf{G}}}^T \tilde{\bar{\mathbf{u}}}_i{}^T \mathbf{A}^T \dot{\mathbf{R}} + \boldsymbol{\theta}^T \dot{\bar{\mathbf{G}}}^T \tilde{\bar{\mathbf{u}}}_i{}^T \tilde{\bar{\mathbf{u}}}_i \dot{\bar{\mathbf{G}}} \boldsymbol{\theta} + \boldsymbol{\theta}^T \dot{\bar{\mathbf{G}}}^T \tilde{\bar{\mathbf{u}}}_i{}^T \dot{\bar{\mathbf{u}}}_i \\ & + \dot{\bar{\mathbf{u}}}_i{}^T \mathbf{A}^T \dot{\mathbf{R}} + \dot{\bar{\mathbf{u}}}_i{}^T \tilde{\bar{\mathbf{u}}}_i \dot{\bar{\mathbf{G}}} \boldsymbol{\theta} + \dot{\bar{\mathbf{u}}}_i{}^T \dot{\bar{\mathbf{u}}}_i \end{aligned} \tag{B.25}$$

The partial derivative of kinetic energy with repect to $\boldsymbol{\theta}$ can be written as

$$\begin{aligned}\frac{\partial T}{\partial \boldsymbol{\theta}} &= \frac{1}{2}\rho\sum_{i}^{N_e}\int_{\Omega_e}\Big\{-2\dot{\mathbf{R}}^T\dot{\mathbf{A}}\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}} - 2\dot{\mathbf{R}}^T\mathbf{A}\tilde{\bar{\mathbf{u}}}_i\dot{\bar{\mathbf{G}}} - 2\dot{\mathbf{R}}^T\mathbf{A}\dot{\tilde{\bar{\mathbf{u}}}}_i\bar{\mathbf{G}} \\ &\qquad +2\boldsymbol{\theta}^T\dot{\bar{\mathbf{G}}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\dot{\bar{\mathbf{G}}} + 2\dot{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\dot{\bar{\mathbf{G}}}\Big\}\,d\Omega_e \\ &= \rho\sum_{i}^{N_e}\int_{\Omega_e}\Big\{-\dot{\mathbf{R}}^T\dot{\mathbf{A}}\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}} - \dot{\mathbf{R}}^T\mathbf{A}\tilde{\bar{\mathbf{u}}}_i\dot{\bar{\mathbf{G}}} - \dot{\mathbf{R}}^T\mathbf{A}\dot{\tilde{\bar{\mathbf{u}}}}_i\bar{\mathbf{G}} - \dot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\dot{\bar{\mathbf{G}}} + \dot{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\dot{\bar{\mathbf{G}}}\Big\}d\Omega_e\end{aligned} \tag{B.26}$$

For the partial derivative of kinetic energy with respect to variable $\mathbf{q}_f$, square norm of velocity can be express with $\bar{\mathbf{u}}_i$

$$\begin{aligned}\dot{\mathbf{r}}_i{}^T\cdot\dot{\mathbf{r}}_i &= \left(\dot{\mathbf{R}} + \mathbf{A}\tilde{\bar{\boldsymbol{\omega}}}\bar{\mathbf{u}}_i + \mathbf{A}\dot{\bar{\mathbf{u}}}_i\right)^T\left(\dot{\mathbf{R}} + \mathbf{A}\tilde{\bar{\boldsymbol{\omega}}}\bar{\mathbf{u}}_i + \mathbf{A}\dot{\bar{\mathbf{u}}}_i\right) \\ &= \dot{\mathbf{R}}^T\dot{\mathbf{R}} + \dot{\mathbf{R}}^T\mathbf{A}\tilde{\bar{\boldsymbol{\omega}}}\bar{\mathbf{u}}_i + \dot{\mathbf{R}}^T\mathbf{A}\dot{\bar{\mathbf{u}}}_i \\ &\quad + \bar{\mathbf{u}}_i{}^T\tilde{\bar{\boldsymbol{\omega}}}^T\mathbf{A}^T\dot{\mathbf{R}} + \bar{\mathbf{u}}_i{}^T\tilde{\bar{\boldsymbol{\omega}}}^T\tilde{\bar{\boldsymbol{\omega}}}\bar{\mathbf{u}}_i + \bar{\mathbf{u}}_i{}^T\tilde{\bar{\boldsymbol{\omega}}}^T\dot{\bar{\mathbf{u}}}_i \\ &\quad + \dot{\bar{\mathbf{u}}}_i{}^T\mathbf{A}^T\dot{\mathbf{R}} + \dot{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\boldsymbol{\omega}}}\bar{\mathbf{u}}_i + \dot{\bar{\mathbf{u}}}_i{}^T\dot{\bar{\mathbf{u}}}_i\end{aligned} \tag{B.27}$$

and, the partial derivative of kinetic energy with respect to $\mathbf{q}_f$ can be written as

$$\begin{aligned}\frac{\partial T}{\partial \mathbf{q}_f} &= \frac{1}{2}\rho\sum_{i}^{N_e}\int_{\Omega_e}\Big\{2\dot{\mathbf{R}}^T\mathbf{A}\tilde{\bar{\boldsymbol{\omega}}}\underline{\mathbf{N}}_i + 2\bar{\mathbf{u}}^T\tilde{\bar{\boldsymbol{\omega}}}^T\tilde{\bar{\boldsymbol{\omega}}}\underline{\mathbf{N}}_i + 2\dot{\bar{\mathbf{u}}}^T\tilde{\bar{\boldsymbol{\omega}}}\underline{\mathbf{N}}_i\Big\}d\Omega_e \\ &= \rho\sum_{i}^{N_e}\int_{\Omega_e}\Big\{\dot{\mathbf{R}}^T\mathbf{A}\tilde{\bar{\boldsymbol{\omega}}}\underline{\mathbf{N}}_i + \bar{\mathbf{u}}^T\tilde{\bar{\boldsymbol{\omega}}}^T\tilde{\bar{\boldsymbol{\omega}}}\underline{\mathbf{N}}_i + \dot{\bar{\mathbf{u}}}^T\tilde{\bar{\boldsymbol{\omega}}}\underline{\mathbf{N}}_i\Big\}d\Omega_e\end{aligned} \tag{B.28}$$

also the partial derivative of potential energy with respect to $\mathbf{q}_f$ using Equation (B.12) can be written as

$$\frac{\partial V}{\partial \mathbf{q}_f} = \mathbf{q}_f^T\mathbf{K}_{ff} \tag{B.29}$$

The left hand side of Equation (B.15) can be written as

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{R}}}\right) = \rho\sum_{i}^{N_e}\int_{\Omega_e}\Big\{\ddot{\mathbf{R}}^T - \ddot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\mathbf{A}^T - 2\dot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\dot{\tilde{\bar{\mathbf{u}}}}_i{}^T\mathbf{A}^T - \dot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\boldsymbol{\omega}}}^T\mathbf{A}^T + \ddot{\bar{\mathbf{u}}}_i{}^T\mathbf{A}^T\Big\}d\Omega_e \tag{B.30}$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial\dot{\boldsymbol{\theta}}}\right)-\frac{\partial T}{\partial\boldsymbol{\theta}}=\rho\sum_{i}^{N_e}\int_{\Omega_e}\Big\{-\ddot{\mathbf{R}}^T\mathbf{A}\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}+\ddot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}+2\dot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\dot{\tilde{\bar{\mathbf{u}}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}$$

$$+2\dot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\dot{\bar{\mathbf{G}}}-\ddot{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}-2\dot{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\dot{\bar{\mathbf{G}}}\Big\}d\Omega_e \tag{B.31}$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial\dot{\mathbf{q}}_f}\right)-\left(\frac{\partial T}{\partial\mathbf{q}_f}-\frac{\partial V}{\partial\mathbf{q}_f}\right)=\rho\sum_{i}^{N_e}\int_{\Omega_e}\Big\{\ddot{\mathbf{R}}^T\mathbf{A}\underline{\mathbf{N}}_i-\ddot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\underline{\mathbf{N}}_i-2\dot{\boldsymbol{\theta}}^T\bar{\mathbf{G}}^T\dot{\tilde{\bar{\mathbf{u}}}}_i{}^T\underline{\mathbf{N}}_i$$

$$+\ddot{\bar{\mathbf{u}}}_i{}^T\underline{\mathbf{N}}_i-\bar{\mathbf{u}}_i{}^T\tilde{\bar{\boldsymbol{\omega}}}^T\tilde{\bar{\boldsymbol{\omega}}}\Big\}d\Omega_e+\mathbf{q}_f^T\mathbf{K}_{ff} \tag{B.32}$$

one can write equation of motion for the flexible body.

$$\sum_{i=1}^{N_e}\int_{\Omega_e}\rho_i\begin{bmatrix}\mathbf{I} & -\mathbf{A}\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}} & \mathbf{A}\underline{\mathbf{N}}_i\\ -\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\mathbf{A}^T & \bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}} & -\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}_i{}^T\underline{\mathbf{N}}_i\\ \underline{\mathbf{N}}_i{}^T\mathbf{A}^T & -\underline{\mathbf{N}}_i{}^T\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}} & \underline{\mathbf{N}}_i{}^T\underline{\mathbf{N}}_i\end{bmatrix}d\Omega_e\begin{bmatrix}\ddot{\mathbf{R}}\\ \ddot{\boldsymbol{\theta}}\\ \ddot{\mathbf{q}}_f\end{bmatrix}+\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{K}_{ff}\end{bmatrix}\begin{bmatrix}\mathbf{R}\\ \boldsymbol{\theta}\\ \mathbf{q}_f\end{bmatrix}$$

$$=\begin{bmatrix}\mathbf{F}_r^{\,\mathbf{ext}}\\ \mathbf{F}_\theta^{\,\mathbf{ext}}\\ \mathbf{F}_f^{\,\mathbf{ext}}\end{bmatrix}+\sum_{i}^{N_e}\int_{\Omega_e}\rho_i\begin{bmatrix}\left\{2\mathbf{A}\dot{\tilde{\bar{\mathbf{u}}}}_i\bar{\mathbf{G}}\dot{\boldsymbol{\theta}}+\mathbf{A}\tilde{\bar{\boldsymbol{\omega}}}\tilde{\bar{\mathbf{u}}}_i\bar{\mathbf{G}}\dot{\boldsymbol{\theta}}\right\}^T\\ \left\{-2\bar{\mathbf{G}}^T\tilde{\bar{\mathbf{u}}}^T\dot{\tilde{\bar{\mathbf{u}}}}\bar{\mathbf{G}}\dot{\boldsymbol{\theta}}-2\dot{\bar{\mathbf{G}}}^T\tilde{\bar{\mathbf{u}}}^T\tilde{\bar{\mathbf{u}}}\bar{\mathbf{G}}\dot{\boldsymbol{\theta}}+2\dot{\bar{\mathbf{G}}}^T\tilde{\bar{\mathbf{u}}}^T\dot{\bar{\mathbf{u}}}\right\}^T\\ \left\{2\underline{\mathbf{N}}_i{}^T\dot{\tilde{\bar{\mathbf{u}}}}_i\bar{\mathbf{G}}\dot{\boldsymbol{\theta}}+\underline{\mathbf{N}}_i{}^T\tilde{\bar{\boldsymbol{\omega}}}^T\tilde{\bar{\boldsymbol{\omega}}}\bar{\mathbf{u}}_i\right\}^T\end{bmatrix}d\Omega_e \tag{B.33}$$

BIBLIOGRAPHY

[1] Lee, S.G., Y.S. Ma, G.L. Thimm, and J. Verstraeten, "Product lifecycle management in aviation maintenance, repair and overhaul". *Computers in Industry*, 2008, vol **59**(2-3), p296-303.

[2] Lemon, J.R., S.K. Tolani, and A.L. Klosterman, "Integration and Implementation of Computer Aided Engineering and Related Manufacturing Capabilities into Mechanical Product Development Process". *CAD-Fachgespräch*, 1980, p161-183.

[3] Coticchia, E., G.W. Crawford, and E.J. Preston, "CAD/CAM/CAE systems ". 2nd ed. 1993, Marcel Dekker.

[4] Kunwoo, L., "Principles of CAD/CAM/CAE Systems". 1999, Addison-Wesley Longman Publishing Co., Inc. 582.

[5] Gordon, L., "CAE inside of CAD ups the ante.(FE update)". *Machine Design*, 2009, vol **81**(4), p74(2).

[6] Kikuchi, N., Y.J. Song, and S. Min, "A Structural Optimization Software for Topology, Shape, and Sizing Optimization*". Series of Lectures at Institute of Advanced Engineering* 1998, October 12-17. Daewoo Group, Korea.

[7] Alfred, M. and S. Raimund, "Prospects and Barriers For Up-Front CAE-Simulation in the Automotive Development*". AutoSim technology workshop*. 2007, Autosim.

[8] Wolf.Krueger, Ondrei.Vaculin, and Willi.Kortuem, "Multi-Disciplinary Simulation of Vehicle System Dynamics*". RTO AVT Symposium*. 2002, 22-25 April. Paris, France.

[9] "MD Nastran Multidiscipline Simulation System for Advanced Engineering Analysis *"*. 2006, MSC Software.

[10] Wang, H., S.-w. Liao, and H. Zhang, "Multi-disciplinary simulation and analysis of complex product in service oriented environment*". IEEE International Conference on Service-Oriented Computing and Applications*. 2007, 19-20 June 2007.

[11] Veitl, A., et al., "Methodologies for coupling simulation models and codes in mechatronic system analysis and design". *Vehicle System Dynamics*, 1999, vol **33**, p231-243.

[12] Foster, I. and C. Kesselman, "The grid : Blueprint for a New Computing Infrastructure ". 2nd ed. 2004, Elsevier.

[13] Wang, J.Z., "Development of Advanced Methodology for Network-Distributed Simulation" *University of Michigan*, 2005,Ph. D. Dissertation

[14] Hulbert, G.M., et al., "Case study for network-distributed collaborative design and simulation: Extended life optimization for M1 Abrams tank road arm". *Mechanics of Structures and Machines*, 1999, vol **27**(4), p423-451.

[15] Wang, J.Z., Z.D. Ma, and G.M. Hulbert, "A gluing algorithm for distributed simulation of multibody systems". *Nonlinear Dynamics*, 2003, vol **34**(1-2), p159-188.

[16] Tseng, F., "Multibody dynamics simulation in network-distributed environments" *University of Michigan*, 2000,Ph. D. Dissertation

[17] Wang, J.Z., Z.D. Ma, and G.M. Hulbert, "A distributed mechanical system simulation platform based on a "gluing algorithm"". *Journal of Computing and Information Science in Engineering*, 2005, vol **5**(1), p71-76.

[18] "The Network Simulator - NS-2"; http://www.isi.edu/nsnam/ns/.

[19] "An Internet Simulated ATM Networking Environment "; http://www.kitchenlab.org/www/bmah/Software/Insane/.

[20] Veith, T.L., J.E. Kobza, and C.P. Koelling, "Netsim: Java (TM)-based simulation for the World Wide Web". *Computers & Operations Research*, 1999, vol **26**(6), p607-621.

[21] Dahmann, F.R. J., and R. Weatherly, "the Department of Defense High Level Architecture". *Proc. of 1997 Winter Simulation Conference*. 1997, p142-149.

[22] Judith S. Dahmann, R.M.F.R.M.W., "The Department Of Defense High Level Architecture". *Proceedings of the 1997 Winter Simulation Conference* 1997,

[23] Kuhl, F., R. Weatherly, and J. Dahmann, "Creating computer simulation systems: An Introduction to the High Level Architecture". 2000, Prentice Hall.

[24] Gan, B.P., et al., "Distributed supply chain simulation across enterprise boundaries ". *Proceedings of the 2000 Winter Simulation Conference* 2000, p1245-1251.

[25] Rajeev, S., J.E.T. Simon, and J. Tharumasegaram, "Distributed supply chain simulation in GRIDS". *Proceedings of the 32nd conference on Winter simulation*. 2000. Orlando, Florida, Society for Computer Simulation International,

[26] Siegel, J., "CORBA 3 Fundamentals and Programming". 2nd ed. 2000, John Wiley & Sons.

[27] Horstmann, M. and M. Kirtland, "DCOM Architecture". July 23, 1997, Microsoft Corporation.

[28] "Remote Method Invocation (RMI)"; http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp.

[29] Prescod, P., "Second Generation Web Services"; http://webservices.xml.com/pub/a/ws/2002/02/06/rest.html. February 06, 2002

[30] Chatterjee, S. and J. Webber, "Developing Enterprise Web Services: An Architect's Guide". 2003 Prentice Hall.

[31] Dieckman, D., D.E. Martin, L. Moore, and P.A. Wilsey, v 3696, 1999, p 44-53., "Distributed web-based simulation for protecting intellectual property*". In Proceedings of SPIE - The International Society for Optical Engineering*. 1999, **v 3696,** p44-53.

[32] Cholkar, A. and P. Koopman, "Widely deployable web-based network simulation framework using CORBA IDL-based APIs*". Winter Simulation Conference Proceedings*. 1999, **2,** p1587-1594.

[33] Fitzgibbons, J.B., et al., "IDSim: an extensible framework for Interoperable Distributed Simulation*". Web Services, 2004. Proceedings. IEEE International Conference on*. 2004, p532-539.

[34] Craig, R.R. and M.C.C. Bampton, "Coupling of Substructures for Dynamic Analyses". *Aiaa Journal*, 1968, vol **6**(7), p1313-1319.

[35] Smith, B.F., P. Bjorstad, and W. Gropp, "Domain decomposition: parallel multilevel methods for elliptic partial differential equations". 1st ed. 2004, Cambridge University Press.

[36] Tallec, P.L., "Domain decomposition methods in computational mechanics". *Computational Mechanics Advances*, 1994, vol **1**, p121-220.

[37] Adeli, H. and S. Kumar, "Distributed Finite-Element Analysis on Network of Workstations - Algorithms". *Journal of Structural Engineering-Asce*, 1995, vol **121**(10), p1448-1455.

[38] Farhat, C. and E. Wilson, "A Parallel Active Column Equation Solver". *Computers & Structures*, 1988, vol **28**(2), p289-304.

[39] Farhat, C.H. and F.X.-. Roux, "Implicit parallel processing in structural mechanics". *Computational Mechanics Advances*, 1994, vol **2**, p1-124.

[40] Farhat, C. and F.X. Roux, "A method of finite element tearing and interconnecting and its parallel solution algorithm". 1991, vol **32**(6), p1205-1227.

[41] Kim, S.S., "A subsystem synthesis method for efficient vehicle multibody dynamics". *Multibody system dynamics*, 2002, vol **7**(2), p189.

[42] Featherstone, R., "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*, 1999, vol **18**(9), p867-875.

[43] Featherstone, R., "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*, 1999, vol **18**(9), p876-892.

[44] Anderson, K.S. and S. Duan, "Highly parallelizable low-order dynamics simulation". *Journal of Guidance, Control, and Dynamics*, 2000, vol **23**(2), p355-364.

[45] Duan, S. and K.S. Anderson, "Parallel implementation of a low order algorithm for dynamics of multibody systems on a distributed memory computing system". *Engineering with Computers*, 2000, vol **16**(2), p96-108.

[46] Sharf, I. and G.M.T. Deleuterio, "Parallel Simulation Dynamics for Elastic Multibody Chains". *Ieee Transactions on Robotics and Automation*, 1992, vol **8**(5), p597-606.

[47] Kubler, R. and W. Schiehlen, "Modular simulation in multibody system dynamics". *Multibody System Dynamics*, 2000, vol **4**(2-3), p107-127.

[48] Gu, B. and H.H. Asada, "Co-Simulation of Algebraically Coupled Dynamic Subsystems Without Disclosure of Proprietary Subsystem Models". *Journal of Dynamic Systems, Measurement, and Control*, 2004, vol **126**(1), p1-13.

[49] Tseng, F.C. and G.M. Hulbert, "A gluing algorithm for network-distributed dynamics simulation. Multibody System Dynamics". 2001, vol **6**(4), p377-396.

[50] Tseng, F.C., Z.D. Ma, and G.M. Hulbert, "Efficient numerical solution of constrained multibody dynamics systems." *Computer Methods in Applied. Mechanics and Engineering*, 2003, vol **192**, p439-472.

[51] Sistla, R., A.R. Dovi, and P. Su, "A distributed, heterogeneous computing environment for multidisciplinary design and analysis of aerospace vehicles". *Advances in Engineering Software*, 2000, vol **31**(8-9), p707-716.

[52] Seifert, M., P. Parkhi, V.T. Sistla, and K.L. Lawrence, "Developing and distributing network based engineering solutions". *Advances in Engineering Software*, 2000, vol **31**(7), p453-465.

[53] Goldfinger, A., et al., "A knowledge-based approach to spacecraft distributed modeling and simulation". *Advances in Engineering Software*, 2000, vol **31**(8-9), p669-677.

[54] Anand V. Hudli, R.M.V.P., "Distributed finite element structural analysis using the client-server model". *Communications in Numerical Methods in Engineering*, 1995, vol **11**(3), p227-233.

[55] "Isight 3.5 Getting Started Guide". 2009, Simulia.

[56] Andrew May, C.C.S.J., "Virtual team working in the european automotive industry: User requirements and a case study approach". *International Journal of Human Factors in Manufacturing*, 2000, vol **10**(3), p273-289.

[57] Malone, B., "A Simulation Based Design Environment for Network Centric Warfare Using Distributed Analysis*"*. 2009, April. 06, Phoenix Integration.

[58] Zienkiewicz, O.C. and R.L. Taylor, "The finite element method. 2: Solid and fluid mechanics dynamics and non-linearity. repr". 4th ed. 1998, Mcgraw-Hill.

[59] Shabana, A.A., "Flexible Multibody Dynamics: Review of Past and Recent Developments". *Multibody system dynamics*, 1997, vol **1**(2), p189.

[60] Shabana, A.A., "Computational Dynamics". 2nd ed. 1994, John Wiley & Sons.

[61] Shabana, A.A., "Dynamics of Multibody Systems". 2nd ed. 1997, Cambridge university press.

[62] Huston, R.L., "Multi-Body Dynamics Including the Effects of Flexibility and Compliance". *Computers & Structures*, 1981, vol **14**(5-6), p443-451.

[63] Huston, R.L., "Computer Methods in Flexible Multibody Dynamics". *International Journal for Numerical Methods in Engineering*, 1991, vol **32**(8), p1657-1668.

[64] T. Belytschko, B.J.H., "Non-linear transient finite element analysis with convected co-ordinates". *International Journal for Numerical Methods in Engineering*, 1973, vol **7**(3), p255-271.

[65] Simo, J.C. and L. Vu-Quoc, "On the dynamics of flexible beams under large overall motions – the plane case: Part I". *Journal of Appl. Mech.*, 1986, vol **53**, p849–863.

[66] Shabana, A.A., "Definition of the Slopes and the Finite Element Absolute Nodal Coordinate Formulation". *Journal of Multibody Dynamics*, 1997, vol **1**(3), p339-348

[67] Shabana, A.A., "Finite element incremental approach and exact rigid body inertia". *Journal of Mechanical Design-Transactions of the Asme*, 1996, vol **118**(2), p171-178.

[68] Simeon, B., "Numerical analysis of flexible multibody systems". *Multibody System Dynamics*, 2001, vol **6**(4), p305-325.

[69] Gerstmayr, J. and J. Schoberl, "A 3D finite element method for flexible multibody systems". *Multibody System Dynamics*, 2006, vol **15**(4), p309-324.

[70] Pedersen, N.L., "On the Formulation of Flexible Multibody Systems with Constant Mass Matrix". *Multibody System Dynamics*, 1997, vol **1**(3), p323-337.

[71] Shabana, A.A., O.A. Bauchau, and G.M. Hulbert, "Integration of Large Deformation Finite Element and Multibody System Algorithms". *Journal of Computational and Nonlinear Dynamics*, 2007, vol **2**(4), p351-359.

[72] Carlbom, P.F., "Combining MBS with FEM for rail vehicle dynamics analysis". *Multibody System Dynamics*, 2001, vol **6**(3), p291-300.

[73] Ottarson, G., G. Moore, and D. Minen, "MDI/ADAMS-MSC/NASTRAN Integration Using Component Mode Synthesis*". MSC Americas Users' Conference*. 1998. Universal City, CA.

[74] Oghbaei, M. and K.S. Anderson, "A new time-finite-element implicit integration scheme for multibody system dynamics simulation". *Computer Methods in Applied Mechanics and Engineering*, 2006, vol **195**(50-51), p7006-7019.

[75] Winfrey, R.C., "Elastic Link Mechanism Dynamics". *Journal of Engineering for Industry*, 1971, vol **93**(1), p268-&.

[76] Popp, K., B. Dirr, and M. Jahnke, "Analysis of flexible parts in multibody systems using finite element methods". *Dynamics and control of large structures; Proceedings of the 8th VPI&SU Symposium*. 1991, p393-404.

[77] Offner, G., T. Eizenberger, and H. Priebsch, "Separation of reference motions and elastic deformations in an elastic multi-body system". *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, 2006, vol **220**(1), p63-75.

[78] Park, K.C., C.A. Felippa, and R. Ohayon, "The d'Alembert-Lagrange principal equations and applications to floating flexible systems". 2009, vol **77**(8), p1072-1099.

[79] Ryu, G., Z.-D. Ma, and G. Hulbert, "A Partitioned Iteration Method Employing CG-Following Reference Frames for Distributed Simulation of Multibody Systems". *Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. 2007. Las Vegas, Nevada, DETC2007-34711,

[80] Agrawal, O.P. and A.A. Shabana, "Application of deformable-body mean axis to flexible multibody system dynamics". *Computer Methods in Applied Mechanics and Engineering* 1986, vol **56**(2), p217-245.

[81] "FEAP : A Finite Element Analysis Program "; http://www.ce.berkeley.edu/projects/feap/. January 2009

[82] Nikravesh, P., "Understanding Mean-Axis Conditions as Floating Reference Frames", in *Advances in Computational Multibody Systems*. 2005. p. 185-203.

[83] Wallrapp, O. and R. Schwertassek, "Representation of geometric stiffening in multibody system simulation". 1991, vol **32**(8), p1833-1850.

[84] Blundell, M.V., "The influence of rubber bush compliance on vehicle suspension movement". *Materials & Design*, 1998, vol **19**(1-2), p29-37.

[85] Amirouche, F.M.L. and T. Jia, "Modelling of clearances and joint flexibility effects in multibody systems dynamics". *Computers & Structures*, 1988, vol **29**(6), p983-991.

[86] Mackerle, J., "Rubber and rubber-like materials, finite-element analyses and simulations: a bibliography (1976-1997)". *Modelling and Simulation in Materials Science and Engineering*, 1998, vol **6**(2), p171-198.

[87] Beijers, C., B. Noordman, and A. Boer de, "Numerical Modelling of Rubber Vibration Isolators: identification of material parameters*". Eleventh International Congress on Sound and Vibration, ICSV 11*. 2004. St. Petersburg, Russia.

[88] Ledesma, R., Z.D. Ma, G. Hulbert, and A. Wineman, "A nonlinear viscoelastic bushing element in multibody dynamics". *Computational Mechanics*, 1996, vol **17**(5), p287-296.

[89] *ADAMS User's Guide*. 2008, MSC Software Corporation.

[90] *DADS(Dynamic Analysis and Design System) User's Manual for Revision 9.6*. 1997, CADSI(Computer Aided Design Software Inc.).

[91] Baumgarte, J., "Stabilization of constraints and integrals of motion in dynamical systems". *Computer Methods in Applied Mechanics and Engineering*, 1972, vol **1**(1), p1-16.

[92] Zeid, A., "Bond Graph Modeling of Planar Mechanisms With Realistic Joint Effects". *Journal of Dynamic System Measurement and Control*, 1989, March, p15-23.