Coupling FE Software through Adapter Elements: A Novel Use of User-Defined Elements

Yuli Huang
Graduate Student Research Associate
Andreas Schellenberg
Graduate Student Research Associate
Stephen A. Mahin
Byron and Elvira Nishkian Professor of Structural Engineering
Gregory L. Fenves
T.Y. and Margaret Lin Professor of Engineering

Department of Civil and Environmental Engineering
University of California at Berkeley
Berkeley, CA 94720

Abstract

An adapter element provides a versatile and computationally efficient method for coupling several finite element (FE) analysis programs so that the unique modeling and analysis capabilities of each can be utilized simultaneously in the simulation of a complete system. FE software is coupled by (1) using each software's programming interface to embed an adaptor element, and (2) connecting the adapter elements using OpenFresco (Open-source Framework for Experimental Setup and Control). The theory underlying the adapter element is based on the penalty method and communication of information at the nodes of the adapter element connecting a master with a slave program. The implementation and accuracy are then demonstrated using a dynamic analysis of a structural model from earthquake engineering.

Introduction

The use of state-of-the-art finite element software provides an effective method for simulating the static and dynamic response of structures. However, such programs are often highly specialized, providing excellent modeling and analysis tools for certain research and engineering fields, but lacking necessary features in other areas. For the analysis and design of civil structures it is necessary to model and simulate complete systems, which may include complex behavior, not just individual components. Thus, there is increasing demand for coupling specialized programs to take advantage of their unique modeling capabilities. The ability to couple finite element analysis software provides additional flexibility and greater realism in simulating large engineering systems than may be possible with a single program.

This paper discusses a new simulation method that employs more than one displacement-based structural finite element (FE) analysis program. Coupled simulations using multiple instances of the same finite element program is a special case. Although the theory presented in the paper uses symbols that are common for the finite element analysis using displacement fields, the idea can be generalized to finite element analysis for other problems, such as heat transfer, electrical potential, etc. The main part of this paper presents the theory for the adapter element and describes its implementation with LS-DYNA®'s new user-defined element interface (LSTC 2007) in conjunction with the Open-source Framework for Experimental Setup and Control

(OpenFresco) (Takahashi and Fenves, 2006; Schellenberg et al., 2007). An illustrative example is then presented from the field of earthquake engineering in which LS-DYNA is coupled with the Open System for Earthquake Engineering Simulation (OpenSees) (McKenna, 1997; Fenves et al., 2007).

Philosophy of FE Software Coupling

Where multiple finite element codes are coupled together, one of the programs is selected as the master, solving the complete structural system, while the other programs model and analyze different structural subassemblies, and thus, act as slaves. The master program can contain parts of the complete system, or *none* of the system. Each subassembly is acting as a superelement and is connected to the master program via interface degrees-of-freedom. The master program sends nodal displacements, at the interface degrees-of-freedom, and obtains forces and possibly the stiffness matrix for the same degrees-of-freedom. From the slave programs' points of view, the displacements at the interface are prescribed, and they return the corresponding nodal reactions.

To connect the master and slave programs in terms of prescribed nodal displacements and corresponding reaction forces, an adapter element is developed based on the penalty method. In addition, a central coordinator program can be used to accommodate all the overhead, including data storage, communication methods, system control, optimization and transformation.

The coordinator program or middleware used in this paper is the Open-source Framework for Experimental Setup and Control (Takahashi and Fenves, 2006; Schellenberg et al., 2007), a transparent and extensible framework for the research and deployment of hybrid simulation. The framework was originally developed to couple structural finite element analysis programs with multiple experimental specimens, in a modular and highly structured manner. For example, in an analysis, OpenFresco's network communication protocols allow different parts of the structure to be analyzed numerically or tested experimentally while being geographically distributed at different sites. Extending this concept, it is feasible to analyze Part A of a large system in Program A, Part B in Program B, while Part C is tested at Experimental Site C, and Part D at Experimental Site D. This means that the complete structure consisting of Parts A, B, C, and D is analyzed collaboratively in Programs A & B and physically tested at Experimental Sites C & D. The remainder of this paper focuses on discussing the coupling of multiple FE-programs. For a discussion of the coupling between analytical and experimental parts of a structure please refer to Schellenberg et al. (2007).

Several approaches may be used to exchange data between coupled codes. Most researchers utilize a file system (Wang et al. 2006; Kwon 2007). When data exchange is required, the slave programs write the model definition and states to a file, and their analysis is terminated. The master program then reads from the file to complete the analysis step. The slaved programs must restart and read pertinent information about past states and the current step from the file before continuing to the next analysis step. Others have made program-specific modifications to the software to be coupled, with communication via network socket connections (Fraunhofer SCAI 2007; Dassault Systèmes, 2007). This requires access to the source code for the software being coupled. For the approach proposed herein, data exchange takes place between adapter elements, embedded by the user into each finite element program using their published programming interfaces (e.g., user-defined elements), by means of open-source coordination

middleware. Thus, all of the connected programs run concurrently. This increases computational efficiency significantly. It also makes it possible for users to customize the coupling.

Theory of Adapter Element

A linear structure, consisting of n_d free degrees-of-freedom, has been discretized in space, and thus, leads to the following system of equations in matrix form.

$$\mathbf{K}\,\mathbf{d} = \mathbf{F} \tag{1}$$

where vector $\mathbf{d} \in \mathbf{R}^{n_d}$, $\mathbf{F} \in \mathbf{R}^{n_d}$ and matrix $\mathbf{K} \in \mathbf{R}^{n_d \times n_d}$. For the finite element approximation

$$\mathbf{K} = \mathop{\mathbf{A}}_{el=1}^{n_{el}} \mathbf{k}_{el}$$

$$\mathbf{F} = \mathop{\mathbf{A}}_{el=1}^{n_{el}} \left[\mathbf{f}_{el,ext} - \mathbf{f}_{el,int}\right]$$

where the symbol $\mathbf{A}$ refers to the direct assembly operator, $\mathbf{k}_{el}$ is the element stiffness matrix, $\mathbf{f}_{el,ext}$ is the external element force vector, and $\mathbf{f}_{el,int}$ is the internal element force vector.

$$\mathbf{k}_{el} = \int_{\Omega_{el}} \mathbf{B}^T \mathbf{D}\, \mathbf{B}\, d\Omega$$

$$\mathbf{f}_{el,int} = \int_{\Omega_{el}} \mathbf{B}^T \boldsymbol{\sigma}_0\, d\Omega$$

where $\mathbf{D}$ is the elasticity matrix, $\mathbf{B}$ is the strain-displacement matrix, $\boldsymbol{\sigma}_0$ is the initial stress, and Ω_{el} is the domain of Element $\mathbf{E}_{el}$.

Assuming that matrix $\mathbf{K}$ is symmetric positive definite, the original problem in Equation (1) can be written in variational form, i.e., find displacement vector $\boldsymbol{d}$ that makes the potential stationary

$$\Pi(\mathbf{d}) = \frac{1}{2}\mathbf{d}^T\mathbf{K}\,\mathbf{d} - \mathbf{d}^T\mathbf{F}$$

In order to couple the slaved subassemblies to the master program, the displacements at the interface degrees-of-freedom are to be prescribed, i.e., for degree of freedom i, $d_i = \bar{d}_i,\ \forall i \in \mathbf{N}_{adpt}$. The size of the set of degrees-of-freedom $\mathbf{N}_{adpt}$ is n_{adpt}. This means that a constraint needs to be appended to the system. The additional linear constraint can be expressed as

$$\mathbf{g}(\mathbf{d}) = \mathbf{Q}\,\mathbf{d} - \bar{\mathbf{d}}_g = \mathbf{0} \tag{2}$$

for a specified vector $\bar{\mathbf{d}}_g \in \mathbf{R}^{n_{adpt}}$ and matrix $\mathbf{Q} \in \mathbf{R}^{n_{adpt} \times n_d}$, and

$$Q_{ij} = \begin{cases} 1 & i = j \in \mathbf{N}_{adpt} \\ 0 & \text{otherwise} \end{cases}$$

If the penalty method is adopted, the penalized potential is defined as

$$\Pi_\kappa(\mathbf{d}) = \Pi(\mathbf{d}) + \frac{1}{2}\mathbf{g}^T \boldsymbol{\kappa}\ \mathbf{g}$$

where $\boldsymbol{\kappa}$ is a *positive definite* penalty matrix, which can physically be interpreted as very stiff spring constants. An intuitive choice for the penalty matrix is

$$\kappa_{ij} = \begin{cases} k_i & i = j \\ 0 & \text{otherwise} \end{cases}$$

where $k_i > 0$ is the penalty stiffness at the $\lfloor \mathbf{N}_{adpt} \rfloor_i$-th degree of freedom whose displacement is prescribed.

Note that the penalized potential is still quadratic in $\mathbf{d}$. Finding a displacement vector $\mathbf{d}$ that makes the penalized potential Π_κ stationary provides a solution to the constrained problem. The stationary point of $\Pi_\kappa(\mathbf{d})$ is referred to as $\mathbf{d}_\kappa$ since it will generally not coincide with the solution $\mathbf{d}$ of the exact constrained system. The stationary condition requires the first variation of Π_κ to vanish

$$\frac{\partial \Pi_\kappa}{\partial \mathbf{d}} = \mathbf{K}\ \mathbf{d}_\kappa - \mathbf{F} + \mathbf{Q}^T \boldsymbol{\kappa}\ \mathbf{g}(\mathbf{d}_\kappa) = \mathbf{K}\ \mathbf{d}_\kappa - \mathbf{F} + \mathbf{Q}^T \boldsymbol{\kappa}\ \left(\mathbf{Q}\ \mathbf{d}_\kappa - \bar{\mathbf{d}}_g\right) = 0$$

or, equivalently,

$$\left[\mathbf{K} + \mathbf{Q}^T \boldsymbol{\kappa}\ \mathbf{Q}\right]\mathbf{d}_\kappa = \mathbf{F} + \mathbf{Q}^T \boldsymbol{\kappa}\ \bar{\mathbf{d}}_g \tag{3}$$

If the errors in the approximate solution are defined as $\mathbf{e}_d = \mathbf{d}_\kappa - \mathbf{d}$, then by rigorous analysis of the penalty method, it can be shown that $\|\mathbf{e}_d\| \to 0$ as $\kappa_{min} \to \infty$, with linear convergence in $1/\kappa_{min}$, where κ_{min} is the smallest eigenvalue of the penalty matrix $\boldsymbol{\kappa}$.

Alternatively, starting from the original problem statement of Equation (1), one can apply *no* constraint (Equation 2), but define an *additional* adapter element $\mathbf{E}_{adpt}$ connecting all the nodes that are related to the degrees-of-freedom in the set $\mathbf{N}_{adpt}$. Its stiffness matrix is defined as $\left[\mathbf{k}_{el}\right]_{adpt} = \boldsymbol{\kappa}$, and its element internal force vector is defined as $\left[\mathbf{f}_{el,int}\right]_{adpt} = -\boldsymbol{\kappa}\ \bar{\mathbf{d}}_g$. The adapter element and all other n_{el} elements form a new system, for which the global stiffness matrix and force vector are assembled as:

$$\tilde{\mathbf{K}} = \mathop{\mathbf{A}}_{el=1}^{n_{el}+1} \mathbf{k}_{el} = \mathbf{K} + \mathbf{Q}^T \boldsymbol{\kappa}\ \mathbf{Q}$$

$$\tilde{\mathbf{F}} = \sum_{el=1}^{n_{el}+1} \left[\mathbf{f}_{e,ext} - \mathbf{f}_{e,int} \right] = \mathbf{F} + \mathbf{Q}^T \boldsymbol{\kappa} \; \bar{\mathbf{d}}_g$$

Finally, since $\tilde{\mathbf{K}} \; \mathbf{d}_\kappa = \tilde{\mathbf{F}}$ from Equation (3), the solution for this new system is exactly $\mathbf{d}_\kappa$. In other words, the adapter element method is equivalent to the penalty method.

For nonlinear structures, the theory is similar but the equilibrium equations are written in the residual form.

$$\mathbf{0} = \tilde{\mathbf{R}} = \tilde{\mathbf{F}} - \tilde{\mathbf{F}}_\mathbf{r}(\mathbf{d}_\kappa) = \mathbf{F} + \mathbf{Q}^T \boldsymbol{\kappa} \; \bar{\mathbf{d}}_g - \mathbf{F}_\mathbf{r}(\mathbf{d}_\kappa) - \mathbf{Q}^T \boldsymbol{\kappa} \; \mathbf{Q} \; \mathbf{d}_\kappa$$

where $\tilde{\mathbf{F}}_\mathbf{r}(\mathbf{d}_\kappa)$ is the assembled resisting force vector from all elements, including the adapter element, and $\mathbf{F}_\mathbf{r}(\mathbf{d}_\kappa)$ is the assembled resisting force vector from all regular elements. Alternatively, the equilibrium equations can be expressed as follows:

$$\tilde{\mathbf{R}} = \sum_{el=1}^{n_{el}+1} \mathbf{r}_{el}$$

where $\mathbf{r}_{el}$ is the residual force vector of Element $\mathbf{E}_{el}$,

$$\mathbf{r}_{el} = \mathbf{f}_{el,ext} - \mathbf{f}_{el,int}$$

For the adapter element, its internal force vector is defined as

$$\left[\mathbf{f}_{el,int} \right]_{adpt} = \boldsymbol{\kappa} \left(\mathbf{Q} \; \mathbf{d}_\kappa - \bar{\mathbf{d}}_g \right) = \boldsymbol{\kappa} \left(\left[\mathbf{d}_{el,\kappa} \right]_{adpt} - \bar{\mathbf{d}}_g \right) \tag{4}$$

where $\left[\mathbf{d}_{el,\kappa} \right]_{adpt}$ is the displacement vector of the adapter element.

If there is no external load applied at the nodes of the adapter element, which is the case for coupled simulation by design, then zero residual leads to

$$\tilde{\mathbf{R}} = -\sum_{el=1}^{n_{el}} \mathbf{f}_{el,int} - \left[\mathbf{f}_{el,int} \right]_{adpt} = \mathbf{0}$$

so the reaction force of all the regular elements can be calculated as

$$\sum_{el=1}^{n_{el}} \mathbf{f}_{el,int} = -\left[\mathbf{f}_{el,int} \right]_{adpt} \tag{5}$$

In other words, if n_{el} regular elements are treated as a superelement, the element reaction force vector can be extracted using Equation (5). The adapter element method for coupling different FE-codes solves two issues essential for coupled simulation: prescribing nodal displacements and obtaining nodal resisting forces.

Table 1: Coupled simulation examples

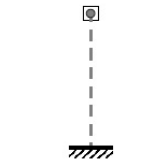
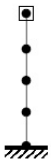
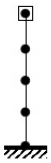
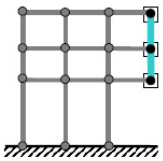
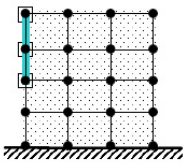
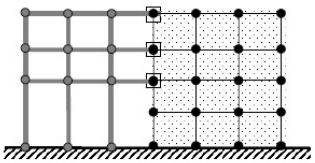
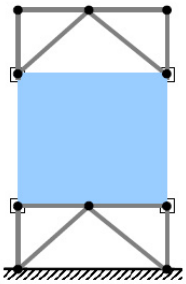
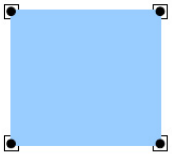
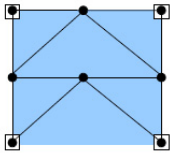
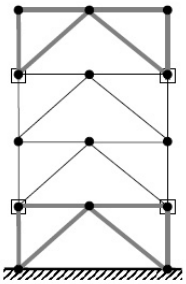
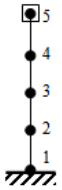
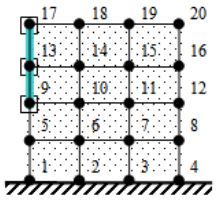
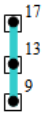
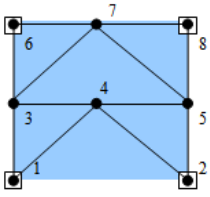
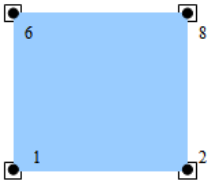
Master structure	Adapter element	Subassembly	Equivalent complete structure
			
			
			

Table 2: Adapter element details

Regular Element Type	Subassembly	Adapter Element	Adapter Element Number of Nodes	Adapter Element Connectivity
beam	 Cantilever column		1	[5]
solid/shell	 Shear wall		3	$\begin{bmatrix} 9 \\ 13 \\ 17 \end{bmatrix}$
beam/truss	 Multiple story		4	$\begin{bmatrix} 1 \\ 2 \\ 8 \\ 6 \end{bmatrix}$

Implementation of Coupled Simulation Using Adapter Elements

Table 1 shows several examples of coupled simulations. The first is a cantilever column. The master structure contains no parts of the complete structure except for the node on top of the column. The column is a subassembly, which is divided into four elements, and it is analyzed in a slave program. The interface from the master program to the subassembly is the top node, where the adapter element attaches. The other three nodes are interior nodes to the subassembly.

The second example is a moment frame combined with a shear wall. The moment frame is analyzed in the master program; and the shear wall is a subassembly analyzed in the slave program. The moment frame is connected to the shear wall through three interface nodes. Therefore, a generic 3-node superelement representing the shear wall is introduced in the master structure. The adapter element is also a 3-node element, but in the substructure space.

The third example is a 4-story braced frame. The second and third stories represent the subassembly to be analyzed in the slave program. This means that in the master structure, there is a generic 4-node superelement representing these two stories. The two top nodes of such element are connected to the fourth story, and the two bottom nodes are connected to the first story. The adapter element is then a 4-node element in the substructure space connecting the four corners.

Table 2 provides details about the corresponding adapter elements used in the coupled simulations illustrated in Table 1. For example, the 4-story braced frame, as illustrated in Figure 1, has a superelement in the master structure representing the subassembly that is being analyzed in the slave program. For given element displacements, the superelement returns element forces. To impose the same displacements on the subassembly, an adapter element is connected to the nodes that are related to these prescribed degrees-of-freedom. Since the adapter is an element, it will also obtain element displacements and return element forces.

The sequence of operations and data exchange necessary to achieve the coupling is shown in Figure 2. First the superelement obtains element displacements $\left[\mathbf{d}_{el}\right]_{super}$. Then it sends these displacements via a TCP/IP connection to the adapter element through the OpenFresco middleware. The adapter element then combines the received displacements $\bar{\mathbf{d}}_g$ with its own element displacements $\left[\mathbf{d}_{el,\kappa}\right]_{adpt}$ from the subassembly. Subsequently, the element force vector of the adapter element $\left[\mathbf{f}_{el,int}\right]_{adpt}$ is updated using Equation (4) and returned to the subassembly. The element internal forces of the subassembly comprising all the regular elements $\mathbf{A}_{el=1}^{n_{el}}\mathbf{f}_{el,int}$ are updated using Equation (5) and sent via an OpenFresco TCP/IP connection back to the superelement. Finally, the superelement saves them as element forces $\left[\mathbf{f}_{el,int}\right]_{super}$ and returns them to the master structure. Note that both the superelement in the master structure and the adapter element in the subassembly can be implemented as user-defined elements (LSTC 2007).

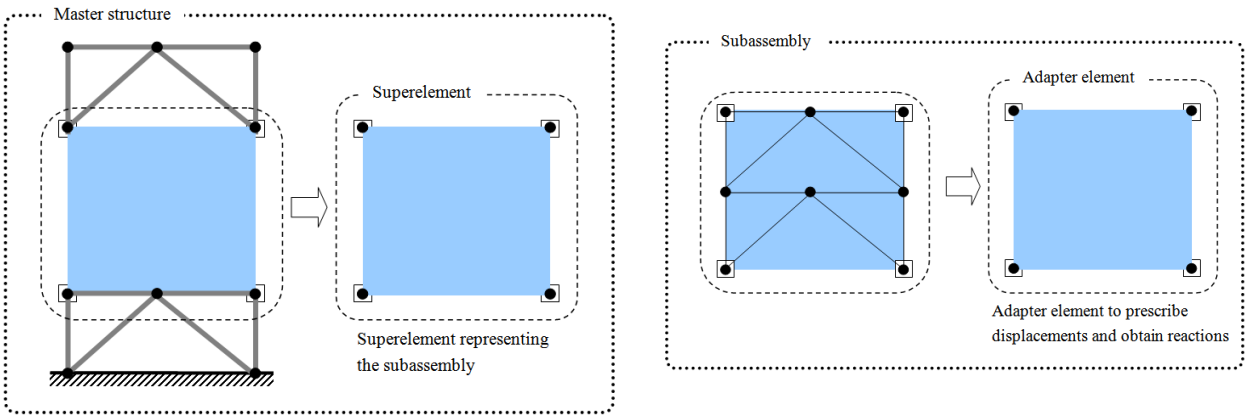


Figure 1: Master structure and subassembly in a coupled simulation

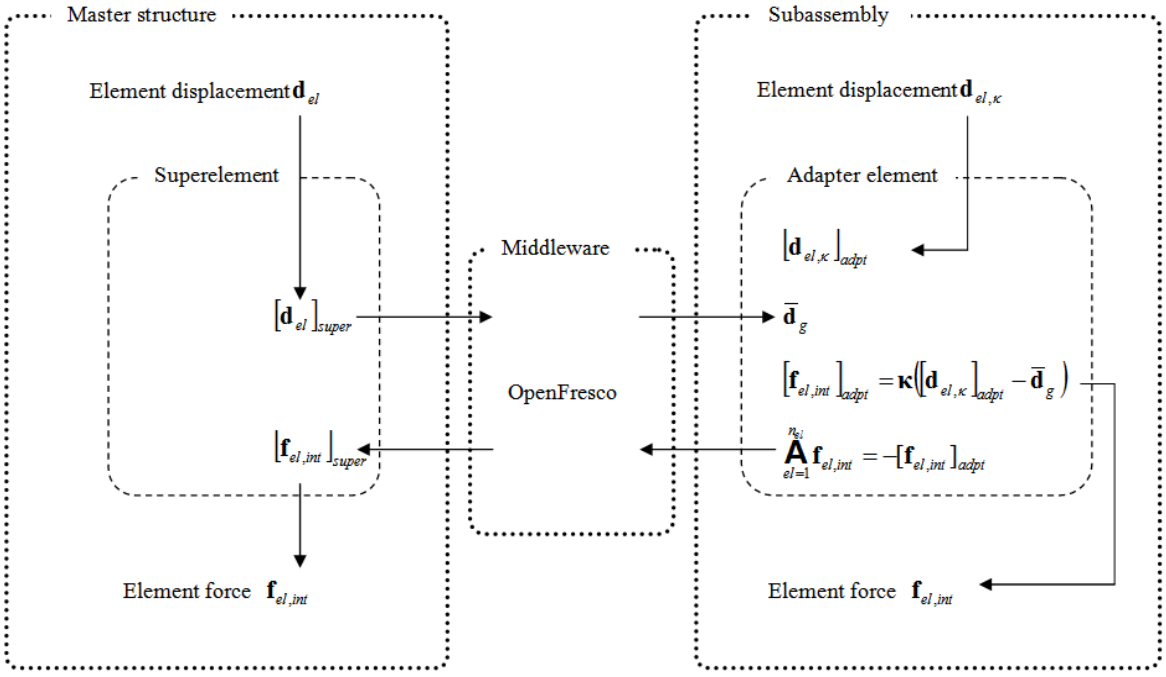


Figure 2: Implementation of coupled simulation using adapter elements

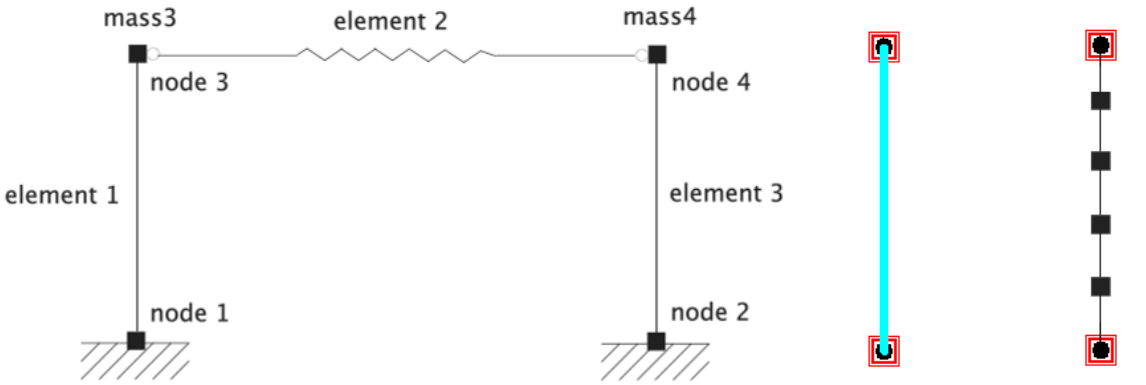
Demonstration of a Coupled Simulation Using Adapter Elements

This section demonstrates how adapter elements can be used in a coupled simulation and assesses the accuracy of such approach. In the illustrative example presented, the software framework Open System for Earthquake Engineering Simulation (OpenSees), is used as the master program. Other FE-programs can be employed in a similar manner. LS-DYNA v971 (LSTC 2007) is used as the slave program, analyzing the subassembly.

As shown in Figure 3a, the model consists of two columns, Element 1 and 3, connected by a spring, Element 2. Lumped masses are placed at the top of each column. The bases of the two columns are fixed. Element 1 is the superelement for the column representing the 6-node, 5-

element subassembly in Figure 3c. The adapter element (the red square in Figure 3b) is a 2-node element connecting to the top and bottom nodes of the subassembly. For comparison, the complete structure is also analyzed with LS-DYNA. The structure is subjected horizontally to the north-south component of the ground motion recorded at a site in El Centro, California during the Imperial Valley earthquake of May 18, 1940. The ground acceleration history is shown in Figure 4 with a time interval of 0.02 sec and a peak acceleration of 0.32 g.

From the simulations, the displacement histories in the horizontal direction at Node 3 are compared and shown in Figure 5: (1) from the coupled simulation (OpenSees + LS-DYNA), and (2) from the single-program simulation (LS-DYNA). As can be seen, the two displacement histories are essentially identical, which illustrates the feasibility and accuracy of coupled simulations using adapter elements.



a) Master structure b) Adapter c) Subassembly

Figure 3: Master structure, adapter and subassembly in demonstration simulation

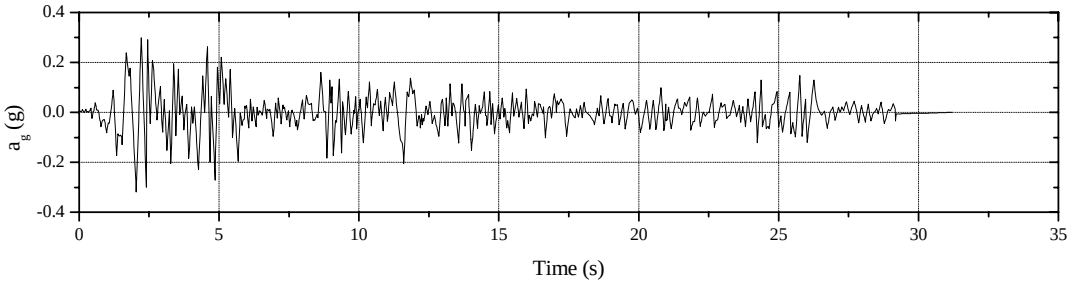


Figure 4: Ground acceleration history

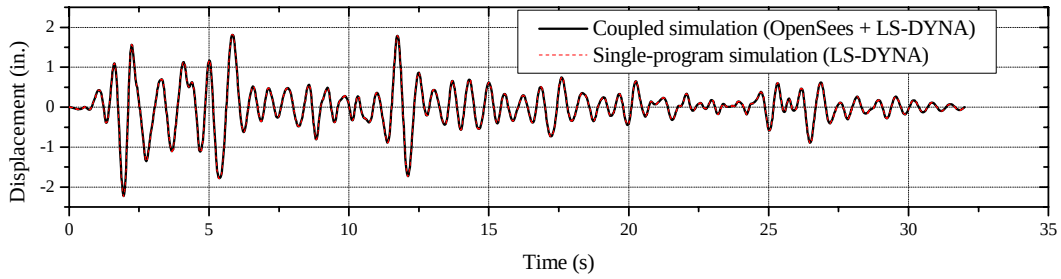


Figure 5: Comparison of displacement histories

Concluding Remarks

The OpenFresco framework middleware for communication between finite element analysis programs provides a useful and effective set of modules for coupling structural analysis software. The adapter elements developed herein illustrate a novel application of user-defined elements, offering an effective means for users to couple different FE-programs. Coupling is important in the simulation of large structural systems that require the unique modeling capabilities of different analysis programs. An example demonstrates the implementation and accuracy of the adapter element concept. The approach avoids the need to repetitively shutdown/restart programs and read/write data files, thereby substantially reducing the time needed for coupled simulations compared to current approaches used for coupling software.

Acknowledgements

Funding for this work was provided in part by the National Science Foundation through a subaward from the NEES Inc consortium under Cooperative Agreement CMS0402490. The authors appreciate the assistance of Prof. Yoshikazu Takahashi of Kyoto University, Dr. Frank McKenna of UC Berkeley, and Dr. Jim Day of LSTC. The findings and conclusions are those of the authors alone, and may not reflect those of the sponsors.

References

Dassault Systèmes (2007). *Abaqus Analysis User's Manual,* ABAQUS, Inc., Dassault Systèmes.

Fenves, G. L., McKenna, F., Scott, M. H., and Takahashi, Y. (2004). "An object-oriented software environment for collaborative network simulation. " *Proceedings of 13th World Conference on Earthquake Engineering*, Vancouver, Canada.

Fraunhofer SCAI (2007). *MpCCI Documentation, Part V, User Manual,* Fraunhofer Institute for Algorithms and Scientific Computing SCAI, Germany.

Kwon, O.-S., Nakata, N., Park, K.-S., Elnashai, A., and Spencer, B. (2007). User Manual and Examples for UI-SIMCOR and NEES-SAM, University Of Illinois, Urbana-Champaign.

LSTC (2007). *LS-DYNA® Keyword User's Manual*, Livermore Software Technology Corporation, CA, US.

McKenna, F. T. (1997). "Object-oriented finite element programming: frameworks for analysis, algorithms and parallel computing." Ph.D. Thesis, University of California, Berkeley.

Schellenberg, A., Mahin, S. and Fenves, G. (2007). "Software Framework for Hybrid Simulation of Large Structural Systems." *Proceedings, Structures Congress, ASCE, Long Beach, CA.*

Takahashi, Y. and Fenves, G. (2006). "Software Framework for Distributed Experimental-Computational Simulation of Structural Systems," *Earthquake Engineering and Structural Dynamics*, 35(3), 267-291.

Wang, T., Nakashima, M., and Pan, P. (2006). "On-line hybrid test combining with general-purpose finite element software." *Earthquake Engineering and Structural Dynamics*, 35, 1471–1488.