STRUCTURAL FE-SOFTWARE COUPLING THROUGH THE EXPERIMENTAL SOFTWARE FRAMEWORK, OPENFRESCO

A. Schellenberg[1], Y. Huang[1] and S.A. Mahin[2]

[1] *Ph.D. Candidate, Dept. of Civil and Env. Eng., Univ. of California, Berkeley, CA, USA*
[2] *Byron and Elivra Nishkian Prof., Dept. of Civil and Env. Eng., Univ. of California, Berkeley, CA, USA*
Email: andreas.schellenberg@gmx.net, yulee@berkeley.edu, mahin@berkeley.edu

ABSTRACT:

This paper describes the continuing evolution of the Open-source Framework for Experimental Setup and Control (OpenFresco), an object oriented, environment independent software framework that can be utilized to couple any number of finite element analysis software packages in a modular and highly structured manner. The exchange of data between the coupled codes takes place through the connection of OpenFresco with special adapter elements, which are added to the finite element applications using their programming interface. 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. This approach provides the important advantage that all of the connected codes run continuously without the need to shutdown and restart, decreasing analysis time consumptions significantly. Speed of coupled simulations where one or more FE-software codes are used together, including cases where physical substructures are incorporated in the FE model for a hybrid test, are at least an order of magnitude faster than conventional approaches relying on shutting down and restarting the analysis in the coupled programs at each integration time step. The implementation and accuracy of this novel FE-software coupling are demonstrated using a dynamic analysis of a structural model from earthquake engineering.

KEYWORDS: Software Coupling, Adapter Element, Hybrid Simulation, OpenFresco

1. INTRODUCTION

The use of state-of-the-art commercial and research finite element software provides a valuable and cost-effective method for simulating the static and dynamic response of structures analytically. However, most of such software packages are often highly specialized, providing excellent modeling and analysis tools for certain research and engineering fields, but lack necessary features in other areas. Recent advancements in the analysis and design of civil structures try to model and simulate entire systems and not just isolated structures or individual components thereof. These systems are generally more complex than their individual parts. Thus, there is an increasing demand for coupling specialized software packages together to take advantage of their unique modeling and analysis capabilities. The coupling of the most suitable finite element analysis software packages from each necessary discipline provides more flexibility and greater realism in simulating large engineering systems than may be possible with a single program.

A new simulation method that couples two or more displacement-based structural finite element analysis programs together is discussed here. It is important to notice that a coupled simulation that utilizes multiple instances of the same finite element program is simply a special case of the proposed approach. Although the theory presented in this paper is specialized for the finite element analysis of structural problems, the same idea is applicable to the finite element analysis of partial differential equations that arise in other environments, such as fluid, thermal and electromagnetic problems. After the discussion of the basic concept of finite element analysis software coupling, the theory for the proposed generic adapter element approach is presented next. The employment of the Open-source Framework for Experimental Setup and Control (OpenFresco) (Takahashi and

Fenves, 2006; Schellenberg et al., 2007) as the middleware among the coupled codes and the implementation of the new adapter element into the Open System for Earthquake Engineering Simulation (OpenSees) (McKenna, 1997; Fenves et al., 2007) and the LS-DYNA® finite element software packages is described as well. Finally, to demonstrate and validate the novel approach to coupling finite element codes, a three-story five-bay frame model is analyzed using two coupled instances of OpenSees.

2. SOFTWARE COUPLING CONCEPT

Whenever multiple finite element software packages are coupled together, one of the programs is selected to act as the master, solving the complete system, while the other linked programs model and analyze different structural subassemblies, and thus, act as slaves. The master program can model subassemblies of the complete system in its own environment, but this is not a requirement for the coupling to function properly. Each slaved subassembly is acting as a super-element and is connected to the master program via interface degrees-of-freedom. During an analysis the master program imposes boundary conditions on all the subassemblies and the slave programs return the corresponding work conjugates and possibly their stiffness or flexibility matrices. The boundary conditions that are to be imposed at the interface degrees-of-freedom can be prescribed displacements, tractions or a combination of the two. However, for the adapter element theory that is presented herein it was assumed that displacements are prescribed and forces and stiffness matrices are returned.

While the adapter elements provide the interfaces to the slave programs, a generic super-element is utilized as the interface to the master program. Additionally, to connect the master and slave programs through their generic super- and adapter element interfaces a middleware or coordinator software is employed. Such software provides essential functionalities such as data storage, communication methods, system control, optimization and data transformations. The 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 software framework for the research and deployment of hybrid simulation. The object-oriented framework was originally developed to couple any structural finite element analysis software with experimental specimens in one or more laboratories, in a modular and highly structured manner. OpenFresco's software objects such as sites, communication channels, transfer system setups, control and data acquisition interfaces allow different parts of the structure to be analyzed numerically or tested experimentally, while they are geographically distributed at different sites. Extending this concept, it is possible to analyze Part A of a large system in Program A, Part B in Program B, while Part C is tested in Laboratory C, and Part D in Laboratory D. This means that the complete structure consisting of Parts A, B, C, and D is analyzed collaboratively in Programs A and B and physically tested in Laboratories C and D. The remainder of this paper focuses on the coupling of multiple finite-element programs for purely analytical simulations without experimental subassemblies in laboratories. For a discussion of the modular and structured coupling of analytical and experimental subassemblies in hybrid simulations, the interested reader is referred to (Schellenberg et al., 2007).

Several other approaches may be used to exchange data between coupled codes. Most researchers utilize a file system (Wang et al. 2006; Kwon 2007) in the following manner. Whenever the coupled codes are required to exchange data, the master program writes the trial response quantities at the interface degrees-of-freedom to files. All the slave programs are then started and read pertinent information about past states and the current state from the files. Once they finish the analysis of the current step, the slave programs write their results back to files and terminate execution. Finally, the master program reads the information from these files and completes its own analysis step. Others have made program-specific modifications to the software to be coupled, using network socket connections for the data exchange (Fraunhofer SCAI, 2007; Dassault Systèmes, 2007). However, this requires access to the entire source code of the software being coupled. On the other hand, in the approach proposed in this paper, generic super- and adapter elements realize the data exchange. These elements are implemented once into each finite element program using their published programming interfaces (e.g., user-defined elements) and OpenFresco, the open-source middleware, manages the communication. Thus, all of the connected codes run continuously and concurrently, without the need to shutdown and restart, increasing computational efficiency significantly. Furthermore, it also makes it possible for users to customize the coupling.

3. ADAPTER ELEMENT THEORY

If a structural subassembly, which is analyzed in one of the slave programs, is discretized in space, the following semi-discrete equations of motion in matrix form are obtained.

$$\mathbf{M}\ddot{\mathbf{U}}(t)+\mathbf{P}_{\mathbf{r}}\left(\mathbf{U}(t),\dot{\mathbf{U}}(t)\right)=\mathbf{P}(t)-\mathbf{P}_{\mathbf{0}}(t) \tag{3.1}$$

The global mass matrix $\mathbf{M}$, the global resisting force vector $\mathbf{P}_{\mathbf{r}}$ and the global element load vector $\mathbf{P}_{\mathbf{0}}$ are given by the following expressions.

$$\begin{aligned}
\mathbf{M} &= \mathbf{M}_{\mathbf{nd}}\ddot{\mathbf{U}}(t)+\mathop{\mathbf{A}}_{el}\mathbf{m}_{el}\ddot{\mathbf{u}}_{el}(t)\\
\mathbf{P}_{\mathbf{r}} &= \mathop{\mathbf{A}}_{el}\mathbf{p}_{\mathbf{r},el}\left(\mathbf{u}_{el}(t),\dot{\mathbf{u}}_{el}(t)\right)\\
\mathbf{P}_{\mathbf{0}} &= \mathop{\mathbf{A}}_{el}\mathbf{p}_{\mathbf{0},el}(t)
\end{aligned} \tag{3.2}$$

In the above Equations (3.1) and (3.2), $\mathbf{M}_{\mathbf{nd}}\ddot{\mathbf{U}}$ are the concentrated inertia forces due to the nodal masses, $\mathbf{A}$ is the direct assembly operator, $\mathbf{m}_{el}\ddot{\mathbf{u}}_{el}$ are the element inertia force contributions, $\mathbf{p}_{\mathbf{r},el}$ are the element resisting forces due to internal stresses, $\mathbf{P}$ are the externally applied nodal loads and $\mathbf{p}_{\mathbf{0},el}$ are element forces due to externally applied loads like body forces, boundary tractions, non-mechanical strains and initial stresses.

$$\begin{aligned}
\mathbf{m}_{el} &= \int_{V_{el}}\mathbf{N}^{T}\rho\mathbf{N}\,dV\\
\mathbf{p}_{\mathbf{r},el} &= \int_{V_{el}}\mathbf{B}^{T}\boldsymbol{\sigma}(\boldsymbol{\varepsilon})\,dV\\
\mathbf{p}_{\mathbf{0},el} &= -\int_{V_{el}}\mathbf{N}^{T}\mathbf{b}\,dV-\int_{\partial V_{t,el}}\mathbf{N}^{T}\mathbf{t}\,dS-\int_{V_{el}}\mathbf{B}^{T}\mathbf{D}\boldsymbol{\varepsilon}_{\mathbf{0}}\,dV+\int_{V_{el}}\mathbf{B}^{T}\boldsymbol{\sigma}_{\mathbf{0}}\,dV
\end{aligned} \tag{3.3}$$

In the above equations $\mathbf{N}$ is a matrix of shape functions, $\mathbf{B}$ is the strain-displacement matrix and $\mathbf{D}$ is the elasticity matrix. In order to couple the slaved subassemblies to the master program, the displacements at the interface degrees-of-freedom are to be prescribed and the corresponding resisting forces are to be measured. To achieve this goal a linear-elastic adapter element that is connected to all the nodal interface degrees-of-freedom is added to the structural subassembly that is modeled in the slave program. To guarantee that the interface degrees-of-freedom move by the prescribed displacement values, the stiffness of such element should be two to three orders of magnitude larger than the stiffness of the subassembly. This concept is comparable to a hybrid simulation where a transfer system (actuator) with a large stiffness (high oil-column stiffness) is imposing boundary conditions on an experimental subassembly. Given the stiffness matrix of the adapter element and assuming that no body forces, boundary tractions, non-mechanical strains and initial stresses act on the adapter element, the externally applied load vector due to the imposed displacements is given by.

$$\mathbf{p}_{\mathbf{0},adpt}=-\mathbf{k}_{adpt}\,\mathbf{u}_{\mathbf{imp}}(t) \tag{3.4}$$

Comparing Equations (3.3c) with (3.4) it can be seen that the externally applied loads of the adapter element are similar to the loads due to non-mechanical strains, but vary over time instead of being constant. Finally, the nodal force vector of the adapter element, which is required by the slave program to assemble its unbalanced force vector, is a combination of the forces due to the adapter element deformations and the ones due to the imposed displacements.

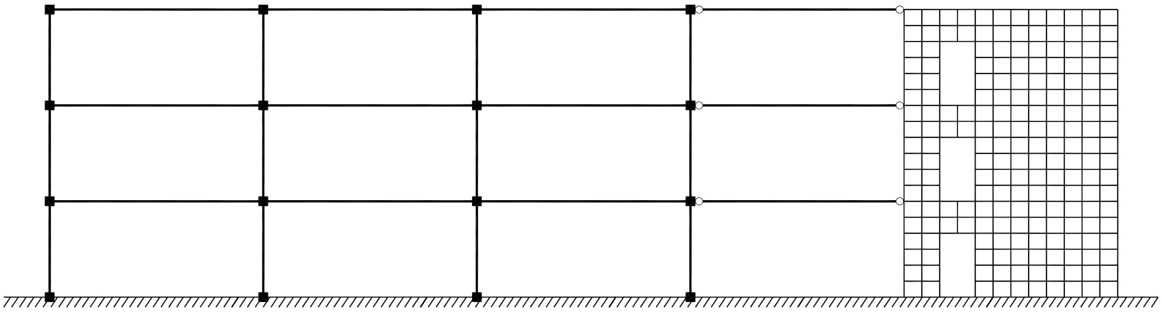
$$\mathbf{p}_{adpt}=\mathbf{p}_{\mathbf{r},adpt}+\mathbf{p}_{\mathbf{0},adpt}=\mathbf{k}_{adpt}\left(\mathbf{u}_{adpt}(t)-\mathbf{u}_{\mathbf{imp}}(t)\right) \tag{3.5}$$

The resisting force vector that corresponds to the imposed displacement vector $\mathbf{u}_{\mathbf{imp}}$, and which is returned to the master program once the equilibrium solution process of the slave program has converged, is the negative of the above equation $-\mathbf{p}_{adpt}$. The displacements that are actually imposed on the subassembly are equal to the adapter element deformations $\mathbf{u}$. Such deformations are generally not exactly equal to the prescribed displacement targets received from the master program. Their accuracy depends on the stiffness of the adapter element with respect to the stiffness of the subassembly. This concludes the derivation of the adapter element theory. An alternative derivation of the theory presented here that is based on the penalty method and leads to the same results is discussed in (Huang et al., 2008).

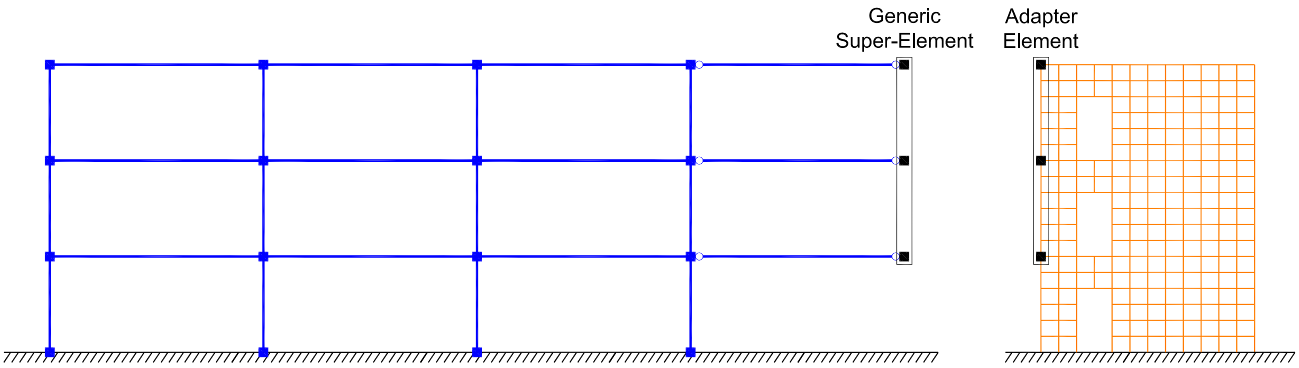
Because the equilibrium Equations (3.1) of the subassembly were written in their most general form, including external loads as well as inertia and energy dissipation effects, it is very important that the employed integration methods in the master and slave programs are compatible with each other. This is necessary so that time in the master and slave programs advance at the same rate. Explicit or Operator-Splitting methods, which only require one data exchange per integration time step, or the specialized constant iteration hybrid simulation integrators, which require a fixed number of data exchanges per time step, need to be employed for the dynamic analysis in the master program. For the dynamic analysis in the slave programs any transient integration method can be used, as long as the time step sizes are compatible with the master integrator. On the other hand, if the behavior of the subassembly in the slave program is time independent (no inertia and energy dissipation effects) a static integrator should be employed in the slave program and any integration method can be utilized for the static or dynamic analysis in the master program.

4. IMPLEMENTATION DETAILS

Next the operations of the different modules of a coupled simulation and the interactions among them are discussed in terms of a three-story example structure. As can be seen from Figure 1 below, the structure consists of a three-story, four-bay steel moment resisting frame that is combined with a three-story concrete shear wall.



Complete Structure



Moment Resisting Frame Subassembly in Master FE-Software

Shear Wall Subassembly in Slave FE-Software

Figure 1: Coupled simulation of three-story example structure.

This hybrid structure is an ideal candidate to be analyzed by two finite element analysis codes, where one of them has excellent frame element modeling capabilities and the other one has superior shell element modeling capabilities. The steel moment resisting frame subassembly is analyzed in the master FE-software and the concrete shear wall subassembly is analyzed in the slave FE-software. Because the moment resisting frame is connected to the shear wall through three interface nodes, a generic 3-node super-element is added to the master program. Hence, this element represents the wall in the master model. As can also be seen from Figure 1 above, the 3-node adapter element connects to the interface nodes of the shear wall subassembly in the slave program and is responsible for imposing trial displacements on such subassembly.

The sequence of operations and the data exchange necessary to achieve the coupling between the master and slave FE-software is shown in Figure 2. Starting on the side of the master program, the super-element receives a vector of global trial displacements $\mathbf{u}_{super} = \mathbf{u}_{el}$ for all its degrees-of-freedom form the master integration method. It then sends these displacements using a TCP/IP socket to the OpenFresco simulation application server. The experimental site and setup modules are responsible for storing and transforming the response quantities if necessary. However, for the example structure presented here, no transformations of the trial displacements and the resisting forces are required, meaning that the NoTransformation experimental setup is utilized. The trial displacements are next passed to the SimFEAdapter experimental control object that provides the connection to the adapter element utilizing a TCP/IP socket. The adapter element then combines the received displacements $\mathbf{u}_{\mathbf{imp}}$ with its own element displacements $\mathbf{u}_{adpt} = \mathbf{u}_{el}$ from the subassembly. Subsequently, the element force vector $\mathbf{p}_{el} = \mathbf{p}_{adpt}$ of the adapter element is updated using Equation (3.5) and returned to the subassembly. Once the equilibrium solution process of the slave program has converged, the negative of the element force vector $-\mathbf{p}_{adpt}$ is returned to the SimFEAdapter experimental control object across the TCP/IP socket. The experimental site and setup modules are again responsible for storing and transforming the response quantities. The simulation application server then returns the force vector through the TCP/IP socket to the super-element in the master program. Finally, the super-element saves them as element forces $\mathbf{p}_{el} = \mathbf{p}_{super}$ and returns them to the master integration method, which is then capable to determine the new trial displacements and proceed to the next time step. It is important to notice that both the super-element in the master subassembly and the adapter element in the slave subassembly can be implemented as user-defined elements into each finite element program using their published programming interfaces.

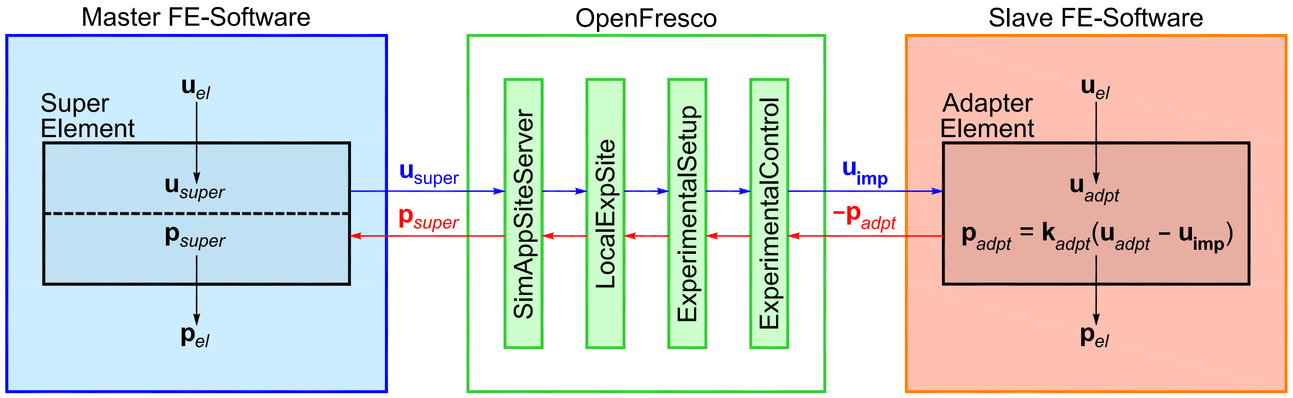


Figure 2: Sequence of operations and data exchange.

5. COUPLED SIMULATION EXAMPLE

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, two instances of the software framework Open System for Earthquake Engineering Simulation (OpenSees) were employed. The first instance acted as the master program and the second one as the slave program. As shown in Figure 3, the complete structural model consisted of a three-story, four-bay steel moment resisting frame and a three-story concrete shear wall. The bay widths and story heights of the structure are 30 ft. and 13.5 ft., respectively. All the columns of the moment resisting frame are fixed at the base. The beams in the bay adjacent to the shear walls are not moment-connected and were therefore modeled as truss elements. The concrete shear wall has a thickness of t = 24 in. and was modeled by plane-stress, four-node, quadrilateral elements. For each story of the wall 12x6 such elements were utilized and the three door openings are 5x9 ft. in size. The total weights of the first, second and third floors are 3950 kip, 3950 kip and 3250 kip, respectively. Lumped masses were assigned to all the nodes according to those floor weights. Since it was assumed that the floor diaphragms are rigid in plane equal degree-of-freedom multipoint constraints were utilized among the nodes of each floor to constrain horizontal translations together. For this simple demonstrative example it was assumed that all the elements exhibit linear-elastic behavior with the moduli of elasticity of 29000 kip/in. and 3600 kip/in. for steel and concrete, respectively.

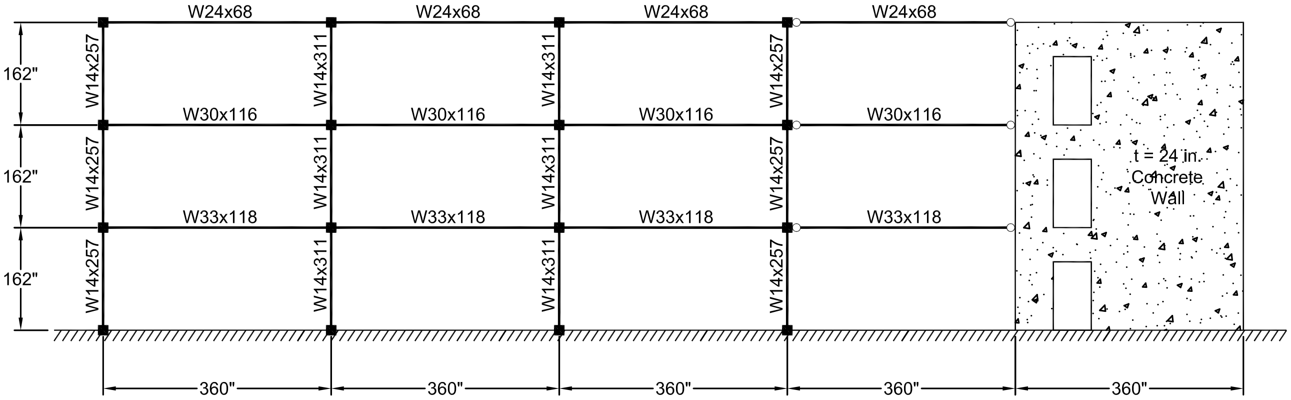


Figure 3: Three-story five-bay frame example structure.

As shown in Figure 1, the four bays of the steel moment resisting frame were analyzed in the master program and the three-story concrete shear wall was analyzed in the slave program. The generic super-element, which represented the shear wall in the master FE-software, connected the three interface nodes at the ends of the truss elements. For simplicity only the horizontal degrees-of-freedom at such nodes were utilized, meaning that the vertical and rotational ones were restrained. The 3x3 initial stiffness matrix of the super-element needs to be specified by the user. Such matrix can be determined from the shear wall subassembly by imposing unit displacements at one interface degree-of-freedom at a time, while restraining the remaining interface degrees-of-freedom. Applying this procedure the following super-element stiffness matrix was determined.

$$\mathbf{k}_{super} = \begin{bmatrix} 93204.7 & -51063.1 & 10190.1 \\ -51063.1 & 82381.2 & -36235.3 \\ 10190.1 & -36235.3 & 23304.8 \end{bmatrix} \tag{5.1}$$

On the other hand the stiffness matrix of the adapter element in the slave finite element software is not determined from the physical properties of the master subassembly. Instead, high stiffness values for the diagonal elements of the matrix are assigned arbitrarily, while keeping in mind that very large stiffness values could cause numerical problems and smaller stiffness values will reduce the accuracy of the imposed displacements. For the example presented here, 1E12 kip/in. stiffness values were assigned to the three diagonal entries of the adapter element stiffness matrix. These were found to produce good accuracies without causing any numerical problems during the analysis.

With the stiffness matrix of the super-element available, it was possible to perform an eigenvalue analysis to determine the periods and mode shapes of the entire structure. The periods of the three horizontal mode shapes turned out to be $T_1 = 0.3119$ sec, $T_2 = 0.0888$ sec and $T_3 = 0.0514$ sec, respectively. For the dynamic analysis the viscous energy dissipation was modeled considering 5% mass proportional damping. The first two mode shapes are shown in Figure 4. The structure was subjected horizontally to the SACNF01 near-fault ground motion of the SAC steel project, which corresponds to the 1978 Tabas earthquake scaled to a peak ground acceleration of 0.755g. The integration time step was chosen to be $\Delta t = 0.01$ seconds.

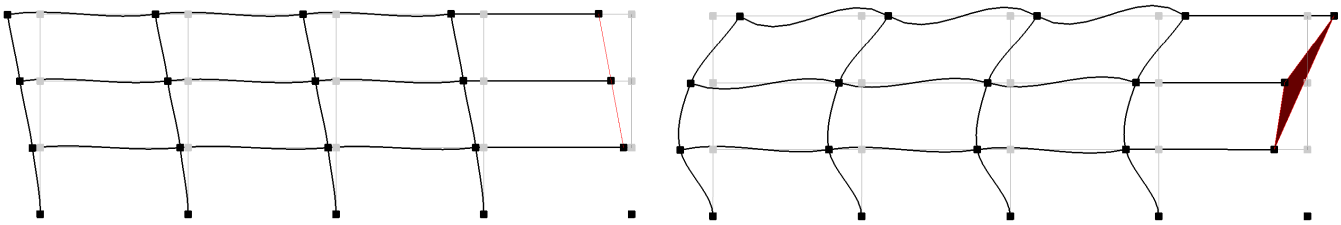
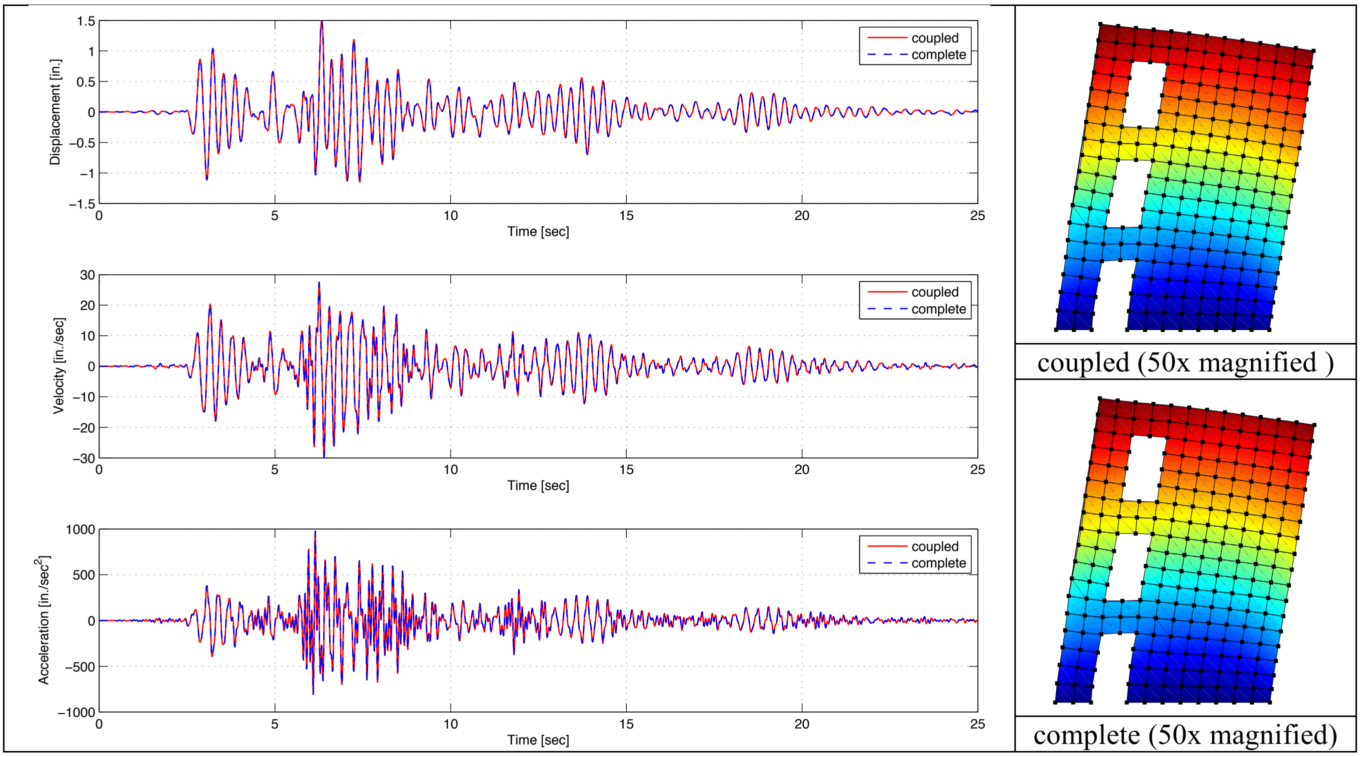


Figure 4: 1st and 2nd horizontal mode shapes determined from coupled analysis.

To validate the coupled dynamic analysis results and assess their accuracies, the complete structure is also analyzed in OpenSees. The horizontal displacement, velocity and acceleration time histories of the third floor diaphragm are shown in Figure 5. As can be seen, the displacement, velocity and acceleration histories of the coupled simulation and the complete simulation are essentially identical, which illustrates the feasibility and accuracy of the adapter element approach. Also, the wall deformations at the maximum horizontal roof displacement of $d_{r,max} = 1.495$ in. at time $t = 6.33$ sec, are essentially identical.



coupled (50x magnified)

complete (50x magnified)

Figure 5: Comparison of third floor response histories and wall deformations at t = 6.33 sec.

Finally the analysis time consumptions are compared between the complete simulation in OpenSees and the coupled simulation using two instances of OpenSees. In both cases elapsed times were measured for the 2500-step-long transient analyses. For the coupled analysis this corresponded to a total of 9990 network transactions across the TCP/IP connection between the two programs. As can be seen from Table 5.1, the coupled simulation is about three times slower than the simulation of the complete structure.

Table 5.1: Comparison of analysis time consumptions.

	complete simulation	coupled simulation
elapsed time	28.7 sec	84.4 sec
number of analysis steps	2500	2500/4995 (master/slave)

6. CONCLUSIONS

The OpenFresco middleware, combined with the adapter element approach, 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 technique for users to couple different finite element analysis software packages. Coupling is important in the simulation of large structural systems that require the unique modeling capabilities of different analysis programs. An example demonstrated 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.

ACKNOWLEDGMENTS

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, France.

Fenves, G. L., McKenna, F., Scott, M. H., and Takahashi, Y. (2004). An object-oriented software environment for collaborative network simulation. Proceedings, 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.

Huang, Y., Schellenberg, A., Mahin, S. A., and Fenves, G. L. (2008). Coupling FE Software through Adapter Elements: A Novel Use of User-Defined Elements. Proceedings, 10th International LS-DYNA Users Conference, Dearborn, MI, United States.

Hughes, T. J. R. (2000). The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Dover, Toronto, Ontario, Canada.

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, IL, United States.

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, CA, United States.

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

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.