TR-2009-[ID]

OpenFresco Framework for Hybrid Simulation: Installation and Getting Started Manual

Andreas Schellenberg, Hong K. Kim, Yoshikazu Takahashi, Gregory L. Fenves, and Stephen A. Mahin

Department of Civil and Environmental Engineering,
University of California, Berkeley

Last Modified: 2009-08-21 Version: 2.6

Acknowledgment: This work was supported by the George E. Brown, Jr. Network for Earthquake Engineering Simulation (NEES) Program of the National Science Foundation under Award Number CMS-0402490. Visit http://it.nees.org/ for more information.

Table of Contents

Table of Figures

1 About This Document

The Open-source Framework for Experimental Setup and Control (OpenFresco) enables local and distributed hybrid simulation of structural systems through an object-oriented software framework. OpenFresco provides an environment-independent software that is modular, transparent, and easily extensible. It can be modified to accommodate various experimental elements, setups and controls.

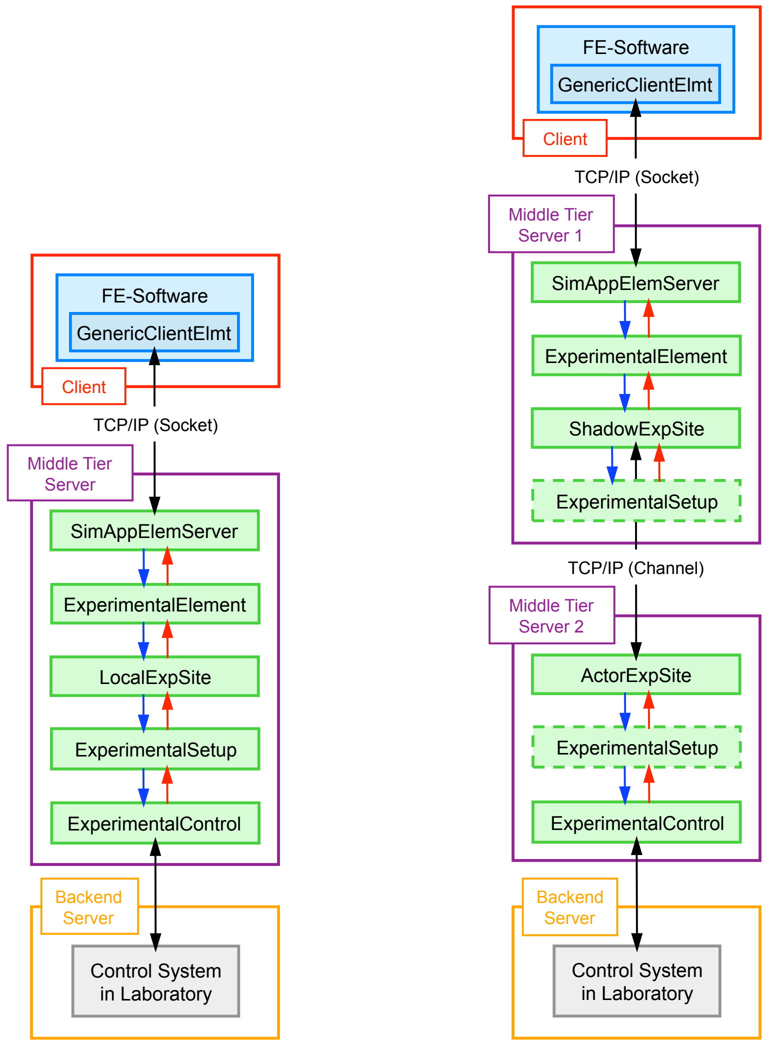


Figure 1.1: OpenFresco Software Architecture for Local (Left) and Distribute Simulation (Right).

OpenFresco is based on a three- or multi-tier client-server architecture, which is composed of a client, one or more middle-tier servers, and a backend server (Figure 1.1)[1]. The blue box represents the finite element software. The green boxes represent OpenFresco processes, and the gray box represents the laboratory control and data acquisition systems. The red box shows the processes that occur on the client side, the purple boxes the middle-tier server side, and the orange box the backend server side. The finite

[1] For further information on the client-server software architect, visit http://www.sei.cmu.edu/str/descriptions/clientserver.html.

element software in Fig. 1.1 uses the GenericClient Element in order to provide a general interface with OpenFresco. The user does not have to create an experimental element in the finite element software when the GenericClient Element is used. However, OpenFresco also provides the user with the option of using a special experimental element in the finite element software. In a distributed test, the Experimental Setup can be executed on either the first or second middle-tier server side depending on the needs and preferences of the user.

This document contains instructions for installing OpenFresco and getting started with hybrid simulation. For the purpose of getting started, the Open Systems for Earthquake Engineering Simulations, OpenSees (http://opensees.berkeley.edu), is used as the computational simulation software. The installation section provides instructions on running the OpenFresco Windows Installer and setting up SVN and OpenSSL. The example section provides information about the model, the ground motion, and the Tcl commands for defining the model and hybrid simulation. It contains step-by-step instructions on running the examples as well as the results for comparison. The OpenFresco Example Manuals contain more comprehensive examples that highlight some of OpenFresco's power and flexibility. They show how OpenFresco can be used with a wide variety of computational software packages such as Matlab/Simulink, LS-DYNA, Abaqus, OpenSees, and UI-SimCor 2.6.

2 What is New and Improved in OpenFresco Version 2.6

2.1 *Modeling of Experimental Elements*

The OpenFresco experimental elements act within the finite element (FE) software to represent the portion of the structure that is physically tested in an experiment. Moreover, the experimental elements provide the necessary interface to the analysis procedures in the FE analysis software.

In addition to the experimental elements in Version 2.5, Version 2.6 has the following features:

1. The twoNodeLink experimental element has been modified to incorporate coupled shear and moment behavior if the length of the element is larger than zero. In addition, if the element does not have zero length, the user can now optionally specify how the P-Delta moments around the local x- and y-axis are distributed among a moment at node i, a moment at node j and a shear couple. The sum of these three ratios is always equal to 1.
2. A corotTruss experimental element is now available to test truss elements considering large displacements. A corotational formulation adopts a set of corotational axes, which rotate with the element, thus taking into account an exact geometric transformation between local and global frames of reference.

2.2 *Representing the Setup of Experiments*

The experimental setup objects in OpenFresco are software abstractions of an experiment's actual setup in the laboratory. As a result, they are responsible for transforming the response quantities between the experimental element degrees of freedom and the actuator degrees of freedom, utilizing the geometry and the kinematics of the loading and instrumentation system, and back again.

Several improvements and additions are introduced in Version 2.6:

1. A FourActuators3d experimental setup has been added to test columns in 3D. As the name suggests, this setup utilized four actuators to control the two translational and the two rotational degrees of freedom of a frame specimen in 3D. The axial and the torsional degrees of freedom are not controlled. However, the setup accounts for the effect of a constant axial load (which is applied to the specimen with a spreader beam and two prestress rods) on the element resisting forces. All transformations in this experimental setup are implemented as non-linear transformations accounting for large displacements.
2. The response modification factors in the experimental setups can now be applied to the trial response quantities before they are transformed to control signals and to the output response quantities after they have been converted from the daq signals. This flexibility to apply factors to trial and output response quantities in addition to control and daq response quantities was necessary to correctly account for similitude scaling operations.

2.3 Interfacing with Controllers and Data Acquisition Systems

OpenFresco provides a wide range of functions for communicating with the laboratory control and data acquisition systems. The functions provide command actions in the actuator coordinate system and they obtain measured data from the data acquisition system.

In Version 2.6, significantly new capability is provided for:

1. The deadlock problems that were encountered with xPC-Target versions newer than 3.1 have been resolved. The xPCtarget experimental control is working flawlessly with the current xPC-Target version 4.1 (Matlab R2009a).
2. The experimental control objects are now able to use signal filter objects to filter control and daq response quantities and to simulate experimental errors such as overshoot, undershoot, lead, lag or random noise.

2.4 Simulating Specimens (including Controllers and Data Acquisition Systems)

In order to check a hybrid model and estimate the response of a structure before running an actual test, specimens can be simulated using the OpenFresco simulation experimental control and data acquisition objects. Besides the existing SimUniaxialMaterial and SimDomain experimental control classes the SimFEAdapter class has been added in Version 2.6. The SimFEAdpater object enables users to simulate physical subassemblies and specimens in any finite element software of their choice. Adapter elements that act as the interfaces to the slave finite element software, in which the subassemblies are simulated, have so far been implemented for OpenSees, LS-DYNA and Abaqus. Because the adapter element approach is not utilizing any file system to exchange data between the master and the slave FE-software and such programs are not required to repetitively be shutdown and restarted, hybrid simulations can be performed continuously and concurrently.

2.5 Utilizing different Computational Drivers

To interface with a variety of finite element software packages, OpenFresco uses the three-tier software architecture concept and a generic client element in the FE-software. A generic client element can be implemented into any FE-software, which provides an application programming interface (API) that allows for the addition of user-defined elements. In addition, a generic client element only needs to be implemented once for any FE-software that is desired to be used as a computational driver for performing hybrid simulations.

Generic client elements have been implemented for OpenSees, Matlab, LS-DYNA, UI-SimCor, Abaqus and Simulink. The Abaqus and the Simulink interfaces are new in OpenFresco Version 2.6.

2.6 Recording Experimental Data

The recorder classes provide means for recording response quantities from all the experimental objects (such as elements, sites, setups and controls) that are of interest to the experimentalist. Data can be recorded in many different formats including, ASCII format without headers, ASCII comma separated format without headers, ASCII format including xml metadata and binary format. In addition, response data can be recorded to databases such as SQL or Berkeley DB.

2.7 Installing the OpenFresco Software

Because of the quickly expanding library of interfaces with controllers and data acquisition systems supported by OpenFresco, the number of required dynamic link libraries is growing rapidly as well. In contrast, the installation of OpenFresco for users and/or developers should be as straightforward and flexible as possible.

In Version 2.6, the OpenFresco installer for Windows platforms, which was introduced in Version 2.5 using the NSIS (Nullsoft Scriptable Install System) professional open-source software, is employed again. The installer provides installation options for general users as well as developers of OpenFresco.

2.8 Documentation for OpenFresco

Several major new documents are provided in public review form for support and explanations for different aspects of OpenFresco. These will be finalized following a brief public review period. The new documentation accompanying Version 2.6 describes:

1. Installation of the software and getting started,
2. OpenFresco specific TCL scripting language commands,
3. Hybrid simulation examples,
4. Adaptation of FE-software packages OpenSees, Abaqus, LS-Dyna and Matlab for hybrid testing, and
5. Generation of certificates needed for the secure communication channels.

3 Installing OpenFresco

The OpenFresco Windows Installer is posted on the NEESforge OpenFresco website (http://openfresco.neesforge.nees.org). This section provides step-by-step instructions for installing OpenFresco. After the installation is complete, the user should be able to run the example in Section 5 of this document and the examples described in the OpenFresco Example Manuals.

3.1 System Requirements

3.1.1 For Users

The OpenFresco Windows Installer installs OpenFresco only on *Microsoft Windows XP* and *Vista*. However, OpenFresco is platform independent. It can run on Mac OS and other Unix/Linux based operating systems with adjustments to the make file. The Tcl/Tk scripting language is used as an interpreter for OpenFresco[2]. Tcl/Tk has many features that make running OpenFresco more convenient such as loops, expressions, and input/output functions. OpenSees and Tcl/Tk version 8.5.x can be downloaded from the OpenSees website (http://opensees.berkeley.edu/OpenSees/user/download.php). Follow the instruction on this website carefully.

3.1.2 For Developers

OpenFresco is written in C++. Therefore, the developer needs a C++ compiler to make extensions or modifications to the software framework. The provided executable and DLLs were generated with Microsoft Visual Studio 2008. So preferably, such compiler should be used to take advantage of the existing Microsoft Visual Studio 2008 project files that come with the source code. Since OpenFresco is taking advantage of and utilizing many existing OpenSees classes, the OpenSees source code needs to be installed and built prior to compiling and building OpenFresco. In addition the TCP_SocketSSL files and the path to ..\OpenSSL\include need to be added to the OpenSees Actor project before building OpenSees. Finally, Tcl/Tk 8.5.x and OpenSSL need to be installed in order to build and execute OpenSees and OpenFresco.

OpenFresco 2.6 is compatible with the most recent xPC-Target version 4.1 (MATLAB® R2009a toolbox).

3.2 Instructions for Installing OpenFresco on Windows XP and Vista

Two types of installations are available with the OpenFresco Windows Installer. The first type is the **Full** (Developers) installation, which includes the source code, examples, executables, and DLLs. This installation is recommended for those who are planning to customize or extend OpenFresco. The steps for the **Full** installation are outlined in Section 3.2.1. The second type of installation is the **Executable, DLLs, and Examples** (Users) installation. As described in the title, this option installs only the executables, DLLs and examples. This is recommended for those who plan to use OpenFresco but not modify or extend the software. The steps for this installation are in Section 3.2.2. To uninstall OpenFresco, use the uninstaller that comes with the OpenFresco installation package. Section 3.2.3 shows how to uninstall OpenFresco.

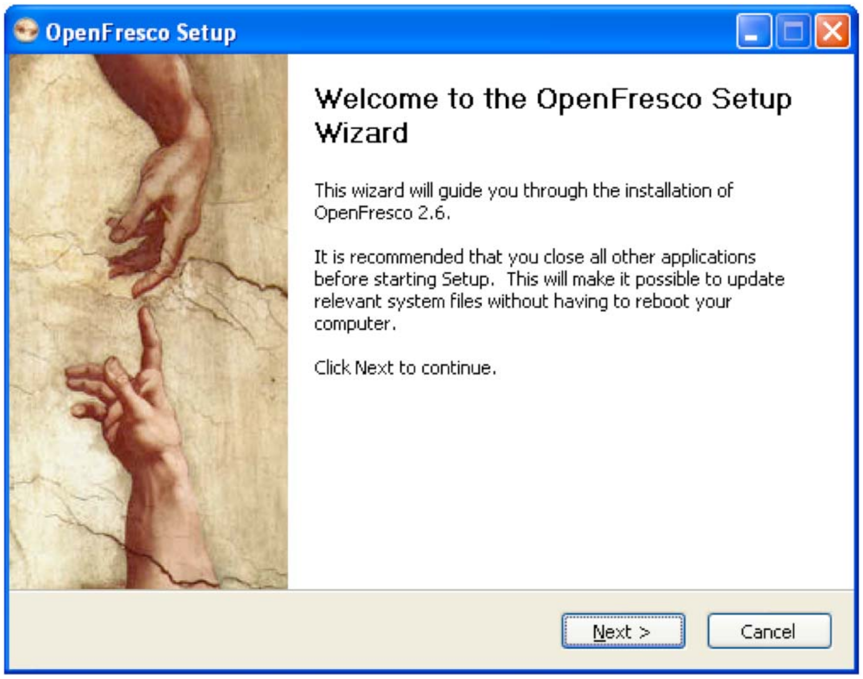
3.2.1 Full Installation (Developers)

1. Download Tcl/Tk Version 8.5.x. The instructions for downloading Tcl/Tk are found on OpenSees website (http://opensees.berkeley.edu/OpenSees/user/download.php) or on the Tcl/Tk website (http://www.tcl.tk).

[2] For information on Tcl/Tk scripting language see B.Welch, K. Jones, and J. Hobbs, *Practical Programming in Tcl and Tk (4th Edition).*

2. The executable file for the OpenFresco Windows Installer can be downloaded from NEESforge OpenFresco website (http://openfresco.neesforge.nees.org).

3. Start the OpenFresco Windows Installer by double clicking on **OPF_Installer.exe**. The Setup Wizard window will open. In this window, click the **Next** button.



Welcome to the OpenFresco Setup Wizard

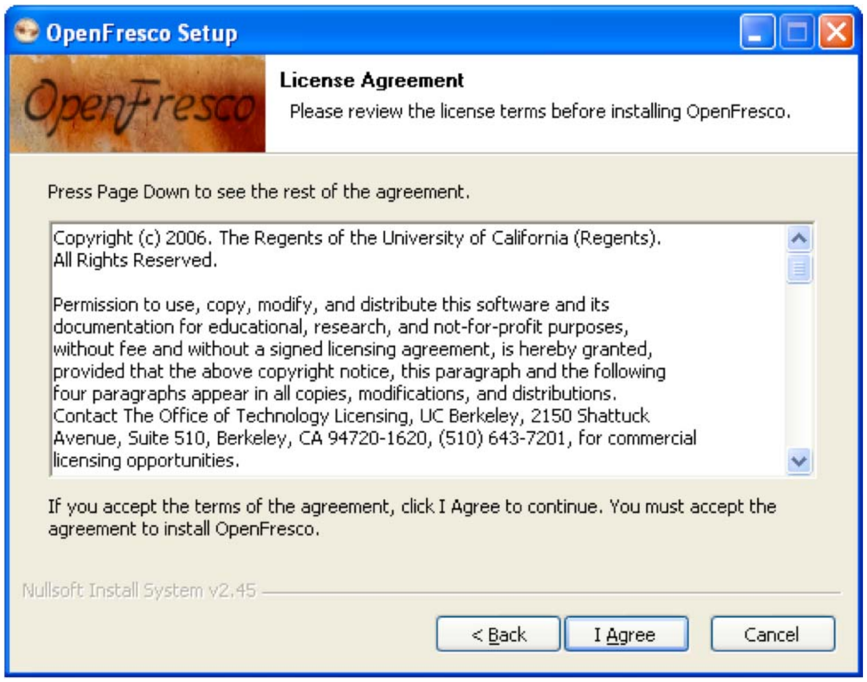
This wizard will guide you through the installation of OpenFresco 2.6.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Figure 3.1: OpenFresco Setup Wizard Window.

4. In the License Agreement screen use the scroll bar to read the entire agreement. When you are ready, click the **I Agree** button.



License Agreement

Please review the license terms before installing OpenFresco.

Press Page Down to see the rest of the agreement.

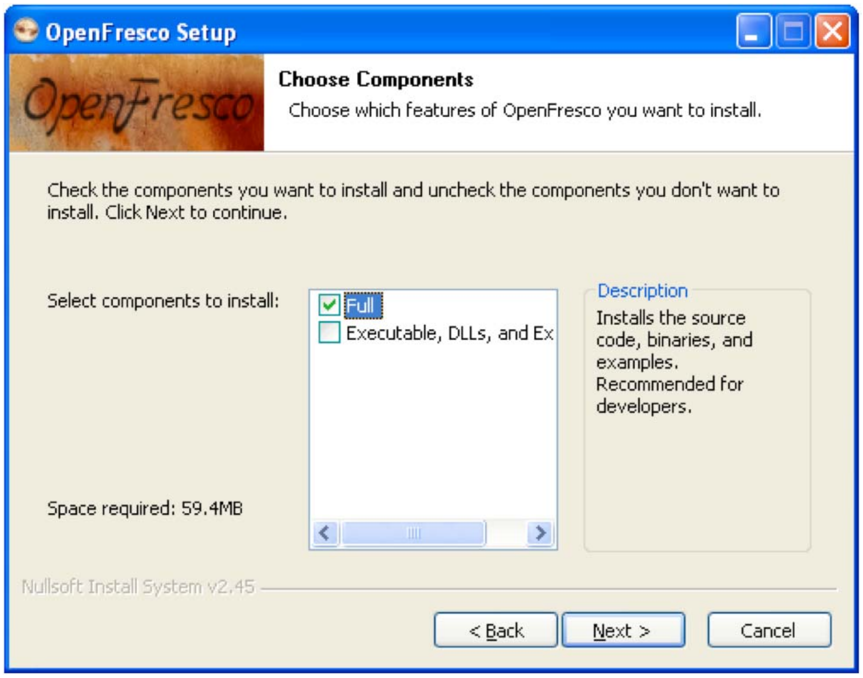
Copyright (c) 2006. The Regents of the University of California (Regents).
All Rights Reserved.

Permission to use, copy, modify, and distribute this software and its documentation for educational, research, and not-for-profit purposes, without fee and without a signed licensing agreement, is hereby granted, provided that the above copyright notice, this paragraph and the following four paragraphs appear in all copies, modifications, and distributions. Contact The Office of Technology Licensing, UC Berkeley, 2150 Shattuck Avenue, Suite 510, Berkeley, CA 94720-1620, (510) 643-7201, for commercial licensing opportunities.

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install OpenFresco.

Figure 3.2: License Agreement Window.

5. Next you'll see the Choose Components screen. Check the **Full** installation option and click **Next**.



Choose Components

Choose which features of OpenFresco you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

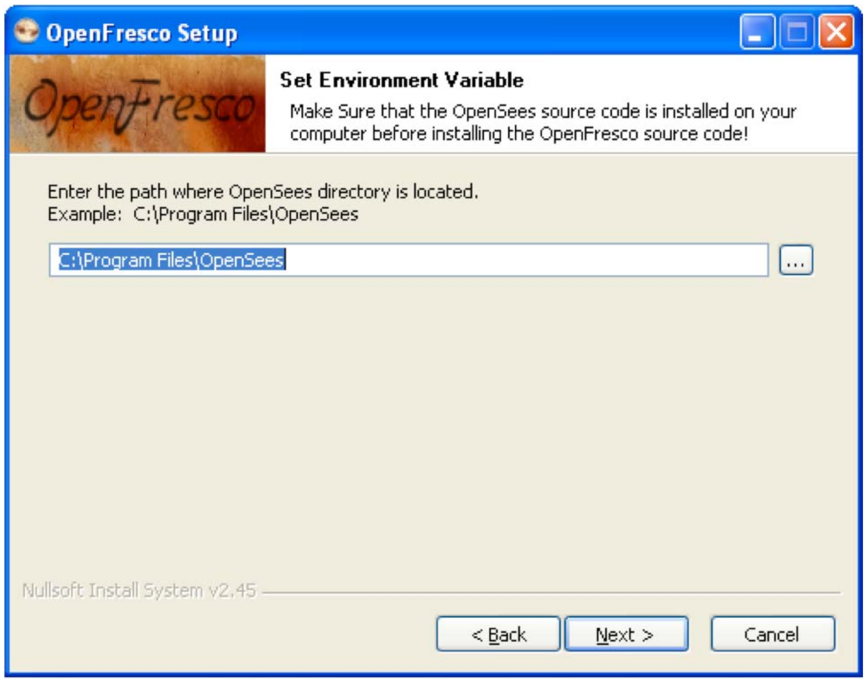
Select components to install: [✓ Full] Executable, DLLs, and Ex

Description: Installs the source code, binaries, and examples. Recommended for developers.

Space required: 59.4MB

Figure 3.3: Installation Component Window.

6. In the Set Environment Variable screen, enter the path to where the OpenSees source code is located. This must be set correctly for OpenFresco to compile. Click **Next**.



OpenFresco Setup

Set Environment Variable

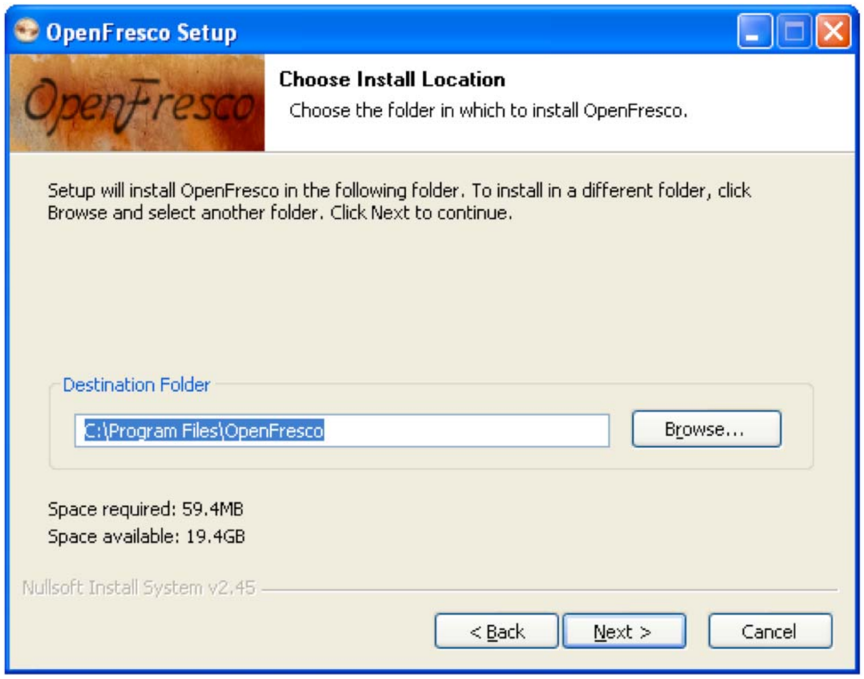
Make Sure that the OpenSees source code is installed on your computer before installing the OpenFresco source code!

Enter the path where OpenSees directory is located.
Example: C:\Program Files\OpenSees

C:\Program Files\OpenSees

Figure 3.4: Set Environment Variable Window.

7. Choose the installation location by clicking on **Browse**. Then click **Next**.



OpenFresco Setup

Choose Install Location

Choose the folder in which to install OpenFresco.

Setup will install OpenFresco in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

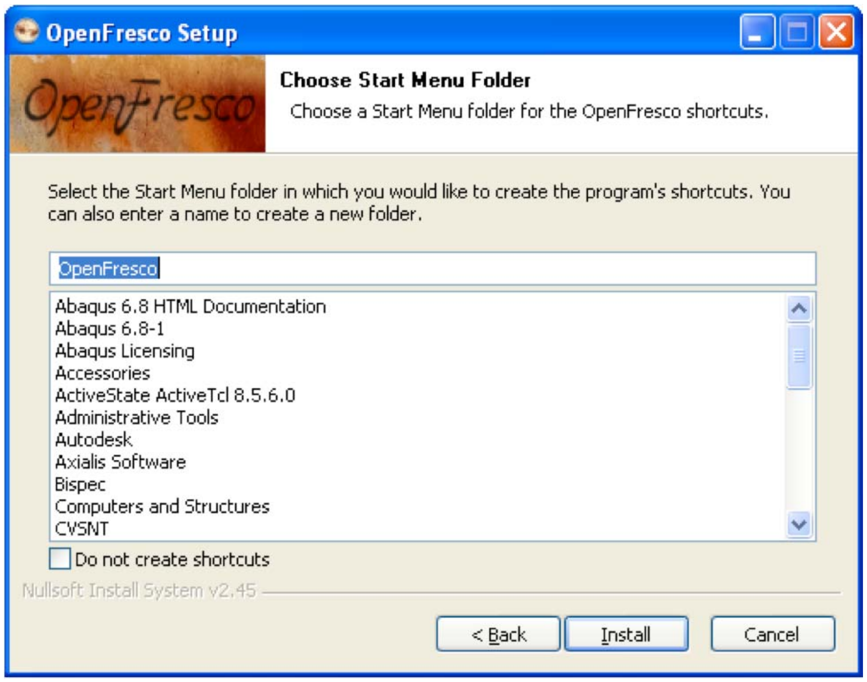
Destination Folder

C:\Program Files\OpenFresco

Space required: 59.4MB
Space available: 19.4GB

Figure 3.5: Choose Installation Location Window.

8. To create a shortcut on the Start Menu, click **Install**. Optionally, check the **Do not create shortcuts** box to bypass this step. Then click **Install**.


Choose Start Menu Folder
Choose a Start Menu folder for the OpenFresco shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

OpenFresco

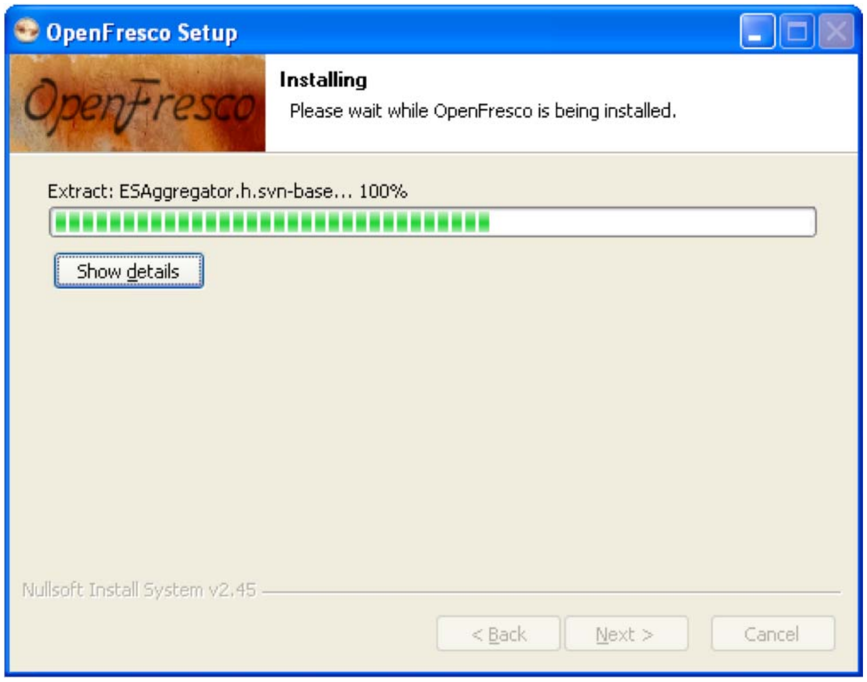
Abaqus 6.8 HTML Documentation
Abaqus 6.8-1
Abaqus Licensing
Accessories
ActiveState ActiveTcl 8.5.6.0
Administrative Tools
Autodesk
Axialis Software
Bispec
Computers and Structures
CVSNT

☐ Do not create shortcuts

Nullsoft Install System v2.45

Figure 3.6: Create Shortcut in Start Menu Window.

9. After clicking **Install**, the Installation Status screen will appear.

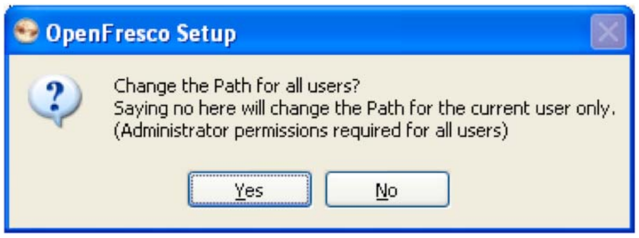

Installing
Please wait while OpenFresco is being installed.

Extract: ESAggregator.h.svn-base... 100%

Nullsoft Install System v2.45

Figure 3.7: Installation Status Window.

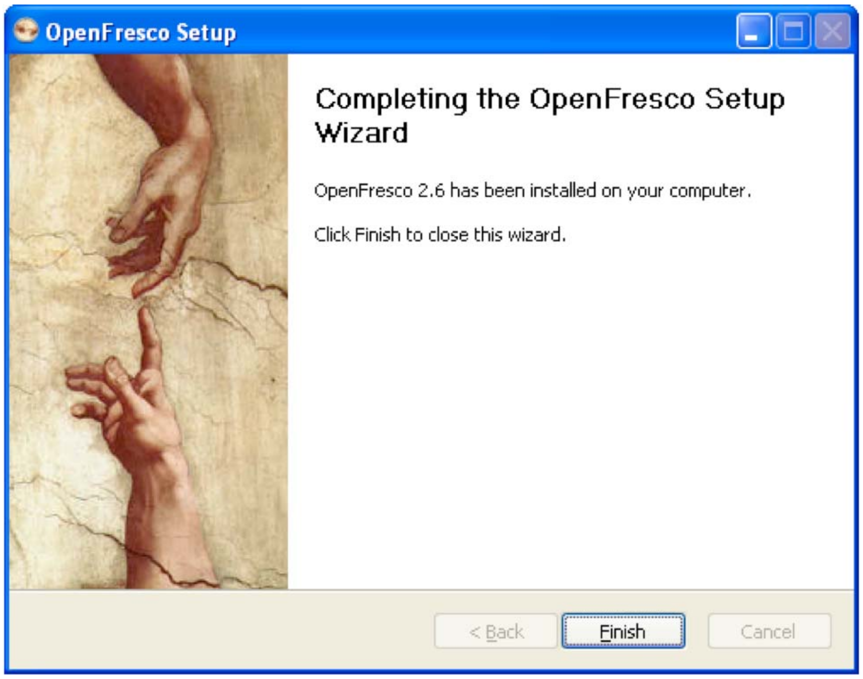
10. During the installation process, a dialogue window as below will appear. Click **Yes** to complete the installation. The OpenFresco Windows Installer adds a directory path (`User's Directory\OpenFresco\trunk\WIN32\bin`) that contains the OpenFresco executable and all the DLLs. This allows the user to start OpenFresco from the MS-DOS prompt without having to be in the directory that contains the executable and the DLLs. This effect can be achieved with OpenSees by placing the OpenSees executable in this directory.


OpenFresco Setup

Change the Path for all users?
Saying no here will change the Path for the current user only.
(Administrator permissions required for all users)

Figure 3.8: Changing Path Dialogue Window.

11. Click **Finish** to complete the installation.


OpenFresco Setup

Completing the OpenFresco Setup Wizard

OpenFresco 2.6 has been installed on your computer.

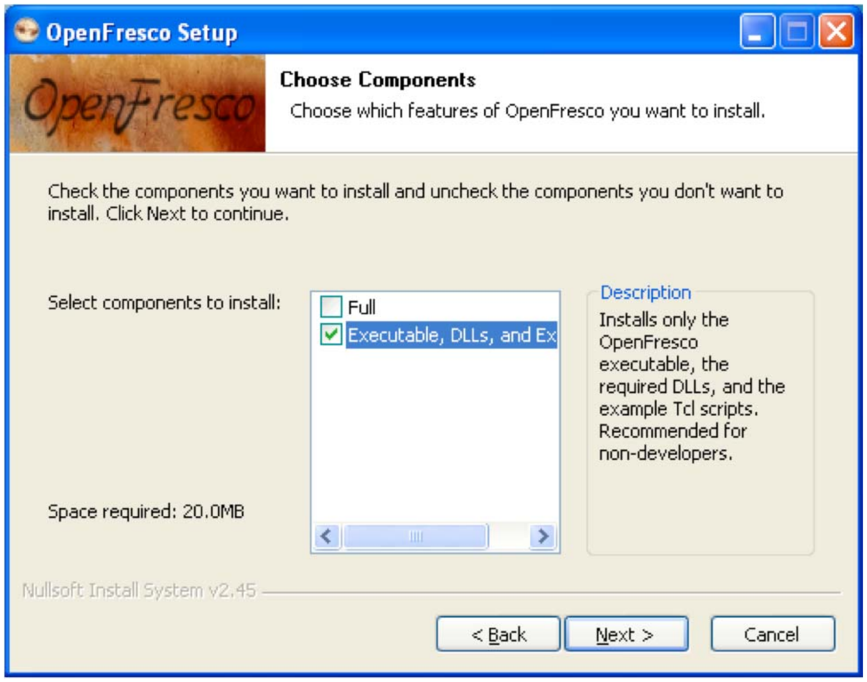
Click Finish to close this wizard.

Figure 3.9: Installation Complete Window.

3.2.2 Executable, DLLs, and Examples Installation (Users)

1. The first four steps are the same as Section 3.2.1.

2. Next you'll see the Choose Components screen. Check the Executable, DLLs, and Examples installation option and click **Next**.



Choose Components

Choose which features of OpenFresco you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

- [ ] Full
- [✓] Executable, DLLs, and Ex

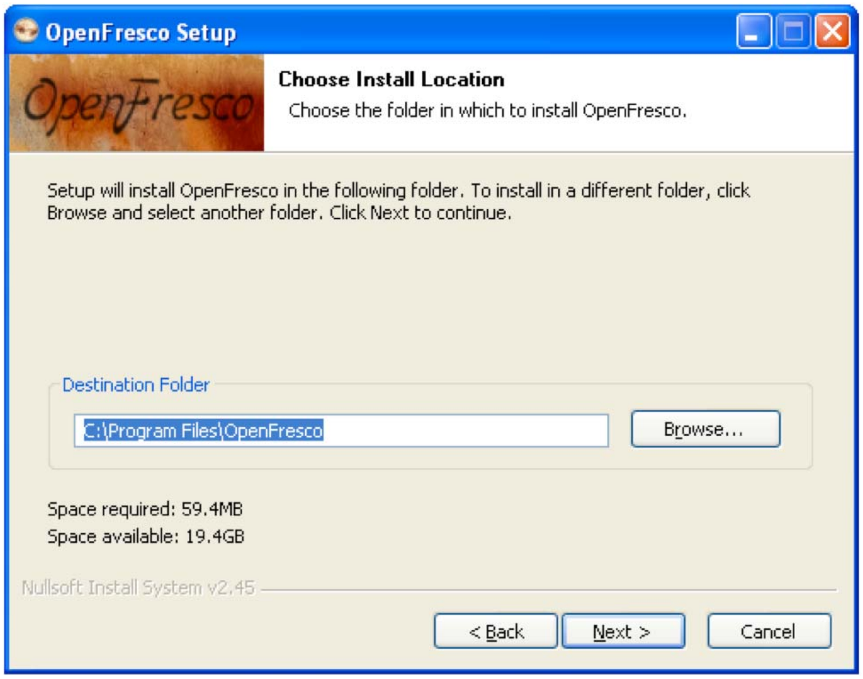
Description

Installs only the OpenFresco executable, the required DLLs, and the example Tcl scripts. Recommended for non-developers.

Space required: 20.0MB

Figure 3.10: Installation Component Window.

3. Choose the installation location by clicking on **Browse**. Then click **Next**.



Choose Install Location

Choose the folder in which to install OpenFresco.

Setup will install OpenFresco in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

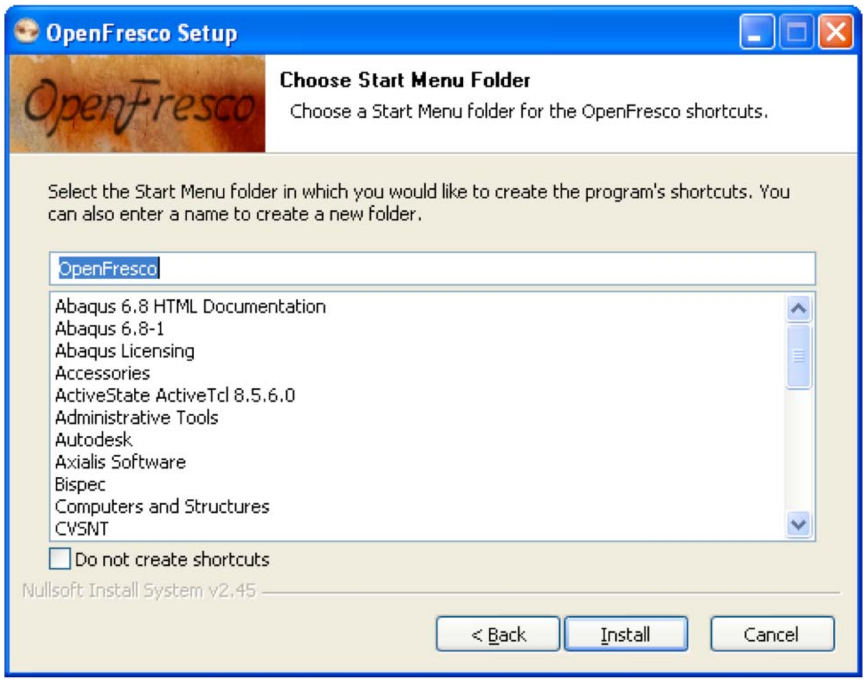
C:\Program Files\OpenFresco

Space required: 59.4MB

Space available: 19.4GB

Figure 3.11: Choose Installation Location Window.

4. To create a shortcut on the Start Menu, click **Install**. Optionally, check the **Do not create shortcuts** box to bypass this step. Then click **Install**.



OpenFresco Setup

Choose Start Menu Folder

Choose a Start Menu folder for the OpenFresco shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

OpenFresco

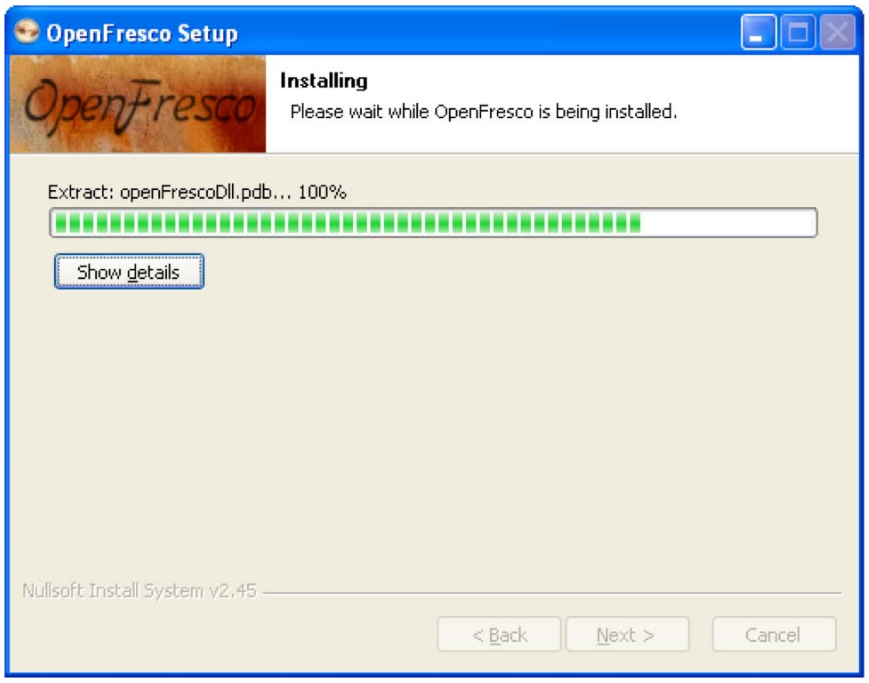
Abaqus 6.8 HTML Documentation
Abaqus 6.8-1
Abaqus Licensing
Accessories
ActiveState ActiveTcl 8.5.6.0
Administrative Tools
Autodesk
Axialis Software
Bispec
Computers and Structures
CVSNT

☐ Do not create shortcuts

Nullsoft Install System v2.45

Figure 3.12: Create Shortcut in Start Menu Window.

5. After clicking **Install**, the Installation Status screen will appear.



OpenFresco Setup

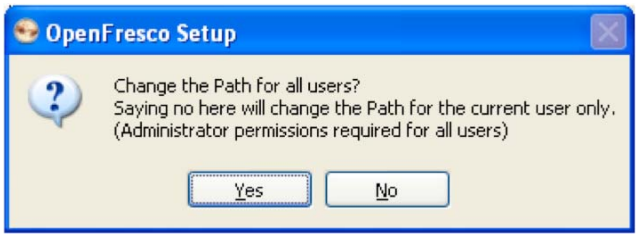
Installing

Please wait while OpenFresco is being installed.

Extract: openFrescoDll.pdb... 100%

Figure 3.13: Installation Status Window.

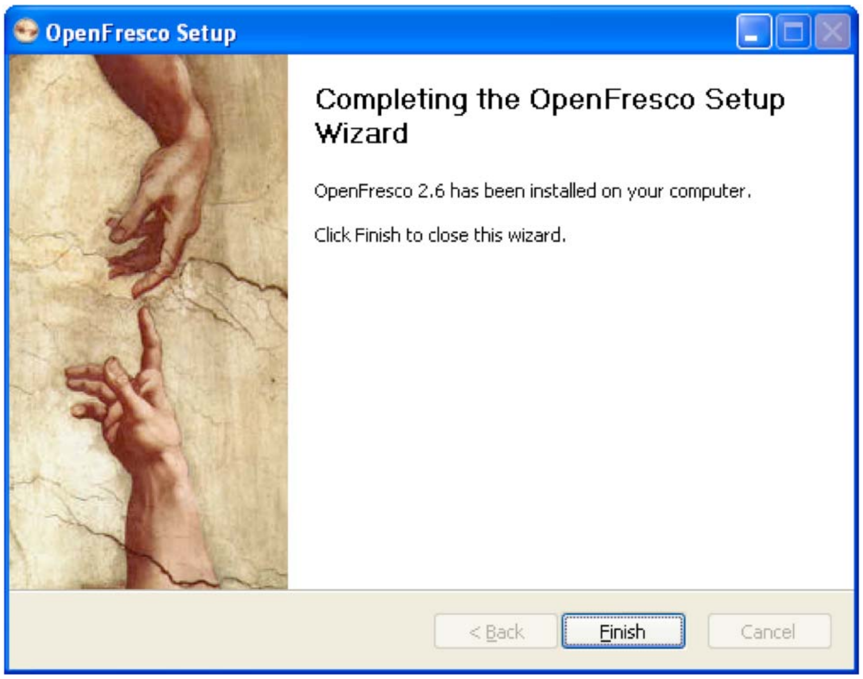
6. During the installation process, a dialogue window as below will appear. Click **Yes** to complete the installation. The OpenFresco Windows Installer adds a directory path (`User's Directory\OpenFresco\trunk\WIN32\bin`) that contains the OpenFresco executable and all the DLLs. This allows the user to start OpenFresco from the MS-DOS prompt without having to be in the directory that contains the executable and the DLLs. The effect can be achieved with OpenSees by placing the OpenSees executable in this directory.



Change the Path for all users?
Saying no here will change the Path for the current user only.
(Administrator permissions required for all users)

Figure 3.14: Changing Path Dialogue Window.

7. Click **Finish** to complete the installation.



Completing the OpenFresco Setup Wizard

OpenFresco 2.6 has been installed on your computer.

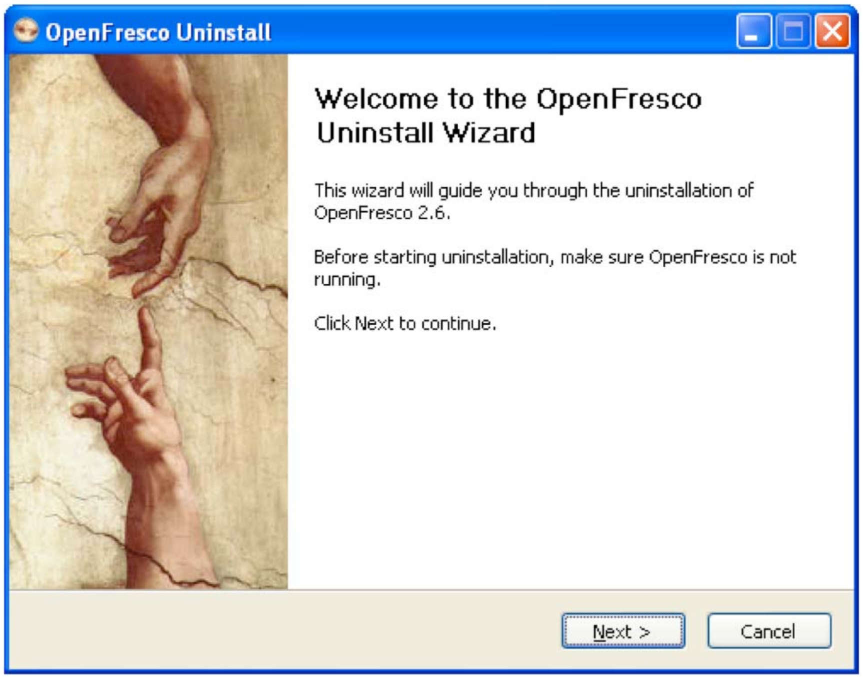
Click Finish to close this wizard.

Figure 3.15: Installation Complete Window.

3.2.3 Uninstalling OpenFresco

The Uninstaller can be found in the `OpenFresco` folder. It can also be accessed through the Start Menu if the shortcut was created during installation.

1. Start the Uninstaller by double clicking on the **Uninstall.exe**. The Uninstall Wizard Window will open. Click the **Next** button.



OpenFresco Uninstall

Welcome to the OpenFresco Uninstall Wizard

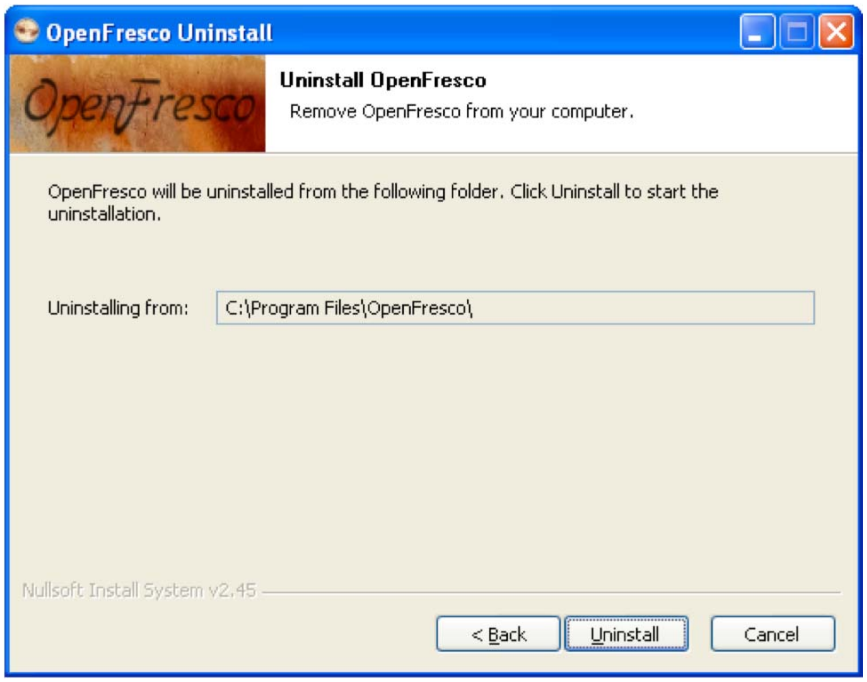
This wizard will guide you through the uninstallation of OpenFresco 2.6.

Before starting uninstallation, make sure OpenFresco is not running.

Click Next to continue.

Figure 3.16: OpenFresco Uninstall Wizard Window.

2. Then, click **Uninstall**.



OpenFresco Uninstall

Uninstall OpenFresco

Remove OpenFresco from your computer.

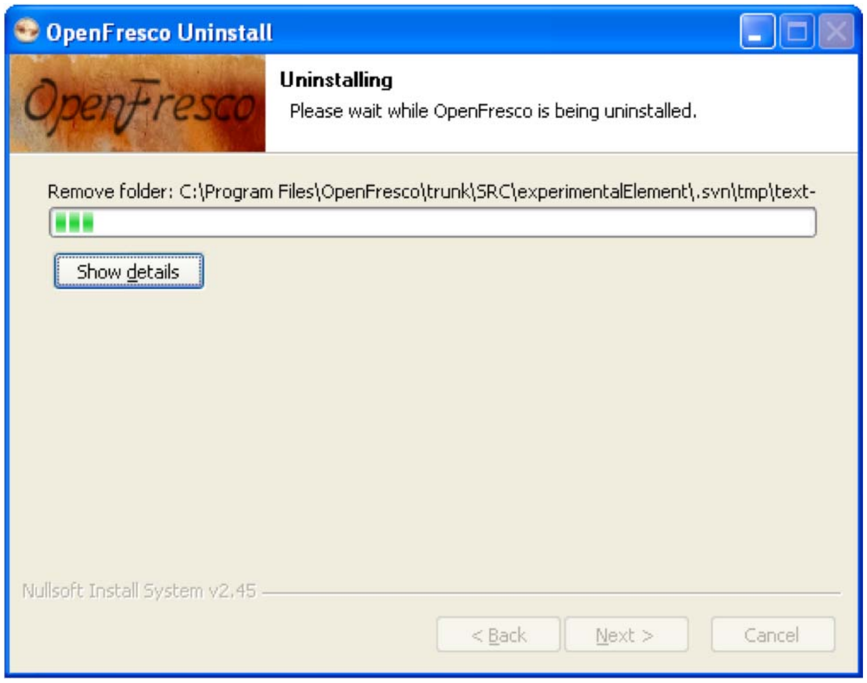
OpenFresco will be uninstalled from the following folder. Click Uninstall to start the uninstallation.

Uninstalling from: C:\Program Files\OpenFresco\

Nullsoft Install System v2.45

Figure 3.17: OpenFresco Uninstall Location Window.

3. After clicking **Uninstall**, the Uninstaller Status screen will appear.


OpenFresco Uninstall

Uninstalling

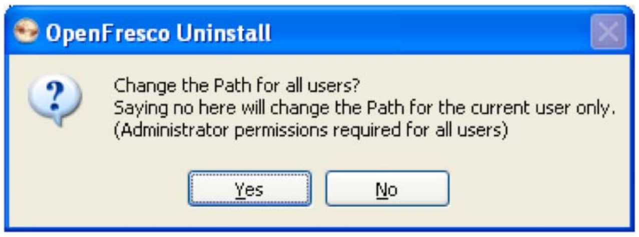
Please wait while OpenFresco is being uninstalled.

Remove folder: C:\Program Files\OpenFresco\trunk\SRC\experimentalElement\.svn\tmp\text-

Nullsoft Install System v2.45

Figure 3.18: OpenFresco Uninstaller Status Window.

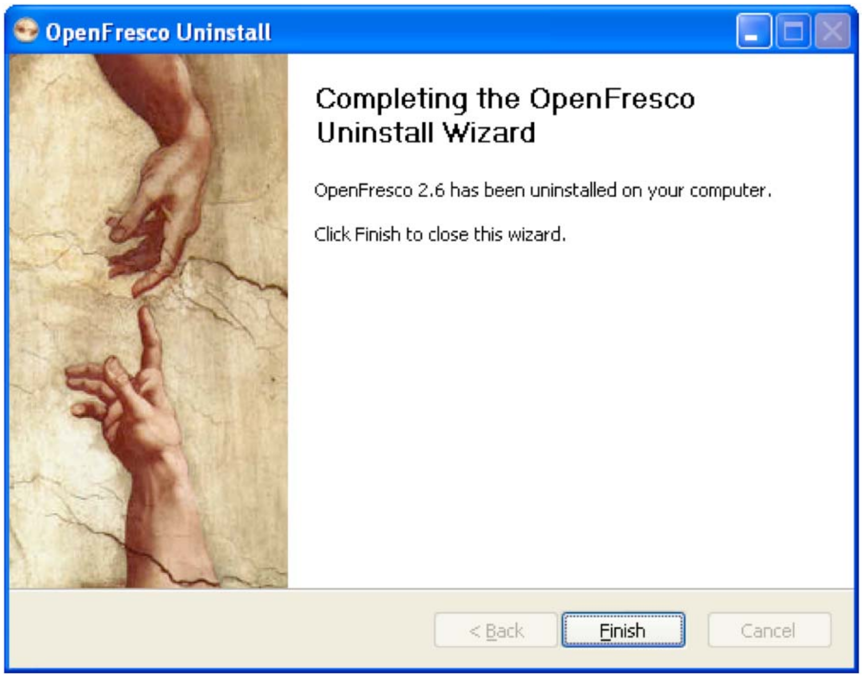
4. During the uninstalling process, a dialogue window will appear. Click **Yes** to complete the uninstalling process.


OpenFresco Uninstall

Change the Path for all users?
Saying no here will change the Path for the current user only.
(Administrator permissions required for all users)

Figure 3.19: Change Dialogue Window.

5. Click **Finish** to complete the uninstalling of OpenFresco.



Completing the OpenFresco Uninstall Wizard

OpenFresco 2.6 has been uninstalled on your computer.

Click Finish to close this wizard.

Figure 3.20: Uninstalling Complete Location Window.

3.3 Staying Current Using SVN

SVN[3] or Subversion is a popular source code version control system for developing and updating software. The OpenFresco team uses SVN for all updates and modifications. If the OpenFresco source code has been installed using the **Full** installation in Section 3.2.1, SVN can be used to keep the source code up to date with the latest version from the OpenFresco team through the NEESforge site.

3.3.1 Installing SVN

There are two ways of installing SVN on *Windows XP* and *Vista*. The first option is to download the setup.exe from the SVN website (http://subversion.tigris.org/servlets/ProjectDocumentList?folderID=91). Any SVN versions 1.5.x or newer will work. Start the setup.exe and follow the directions. After the installation is complete, the SVN commands will be accessible through the MS-DOS prompt.

A user-friendlier alternative is to download Tortoise SVN from the Tortoise SVN website (http://tortoisesvn.tigris.org/). Tortoise SVN is a GUI interface to SVN. All SVN commands are conveniently available through the *Windows Explorer*.

3.3.2 Updating Source Code

Currently, the anonymous SVN access is not working at NEESforge. A NEESforge user account needs to be setup to gain access to the SVN repository. If not done so already, sign up for an account at the NEES Forge website (https://neesforge.nees.org/account/register.php). If the MS-DOS prompt is being used, go to the `OpenFresco` directory. Then, type `svn update` at the prompt. If Tortoise SVN is being

[3] For information on SVN go to the SVN website (http://subversion.tigris.org).

used, right click on the `OpenFresco` folder. SVN commands will appear in the drop-down menu. Choose `SVN Update`. In both cases, SVN will prompt the user for a NEESforge login and password.

4 Setting Up OpenSSL

To enable the OpenSSL[4] feature in OpenFresco for secure communications, download OpenSSL (win 32 version) from the OpenSSL website (http://www.slproweb.com/products/Win32OpenSSL.html). Install OpenSLL in `C:\Program Files\OpenSSL\` by following the onscreen instructions. There are two options for setting up OpenSSL for OpenFresco. The first option is to create self-signed certificates (Sec 4.1). The other option is to create certificates using your own certificate authority (Sec 4.2).

4.1 Creating Self-Signed Certificates

1. Run openssl.exe from the MS-DOS command prompt: `C:\Program Files\OpenSSL\bin> openssl`

2. Generate a private key for the server:
   a. With password: `OpenSSL> genrsa -des3 -out server.key 4096`
   b. Without password: `OpenSSL> genrsa -out server.key 4096`

3. Generate a self-signed certificate from the private key: `OpenSSL> req -new -x509 -days 365 -key server.key -out server.crt`

4. Provide the required information. An example is shown below, but you should use the information for your location, organization, and identification:

```
Country Name: US
State or Province Name: California
Locality Name: Berkeley
Organization Name: NEES
Organizational Unit Name: UCB
Common Name: OpenFresco
Email Address: .
```

5. Exit OpenSSL: `OpenSSL> exit`

6. Generate a client CA certificate by making a copy: `> copy server.crt client_ca.crt`

7. Repeat steps 1 to 6 to generate the following files.
   a. `client.key`
   b. `client.crt`
   c. `server_ca.crt`

8. Place the server files in the folder where server-input scripts reside and the client files in the folder where client input scripts reside.

[4] For more information on OpenSLL, go to http://www.openssl.org/docs/HOWTO/

4.2 Creating Certificates using your own CA (Certificate Authority)

1. Run openssl.exe from the MS-DOS command prompt: `C:\Program Files\OpenSSL\bin> openssl`

2. Generate a private key for your own local CA:
   a. With password (preferably): `OpenSSL> genrsa -des3 -out ca.key 4096`
   b. Without password: `OpenSSL> genrsa -out ca.key 4096`

3. Generate a CA certificate from the private key (copies will be made in step 11): `OpenSSL> req -new -x509 -days 3650 -key ca.key -out ca.crt`

4. Provide the required information. An example is shown below, but you should use the information for your location, organization, and identification:

```
Country Name: US
State or Province Name: California
Locality Name: Berkeley
Organization Name: NEES
Organizational Unit Name: UCB
Common Name: OpenFrescoCA
Email Address: .
```

5. Create a directory called `localCA` with a subdirectory `private` and move `ca.key` into `..\localCA\private\` and `ca.crt` into `..\localCA\`. This concludes the generation of the local certificate authority. In the next few steps, the server and client certificate requests are generated and signed by the CA.

6. Generate a private key for the server:
   a. With password: `OpenSSL> genrsa -des3 -out server.key 4096`
   b. Without password: `OpenSSL> genrsa -out server.key 4096`

7. Generate a certificate request from the private key: `OpenSSL> req -new -key server.key -out server.csr`

8. Provide the required information. An example is shown below, but you should use the information for your location, organization, and identification:

```
Country Name: US
State or Province Name: California
Locality Name: Berkeley
Organization Name: NEES
Organizational Unit Name: UCB
Common Name: OpenFrescoServer
Email Address: .
A challenge password: *******
An optional company name:
```

9. Sign the certificate request with the CA: `OpenSSL> x509 -req -days 365 -in server.csr -CA ..\localCA\ca.crt -CAkey ..\localCA\private\ca.key -set_serial 01 -out server.crt`

10. Repeat steps 6 to 9 to generate the following files.
    a. `client.key`
    b. `client.csr`
    c. `client.crt`

11. Place `server_ca.crt` (a copy of `ca.crt`), `server.key` & `server.crt` files in the folder where server-input files reside and `client_ca.crt` (a copy of `ca.crt`), `client.key` & `client.crt` files in the folder where the client input files reside.

5 One-Bay Frame Example Using OpenSees

After completing the installation, run the One-Bay Frame example to make sure that OpenFresco has been installed correctly. This example demonstrates several basic features and capabilities of the software. It shows how OpenFresco can be used to run both local and distributed hybrid simulations using the multi-tier client-server architecture described in Section 1. The One-Bay Frame example is a fully simulated test, meaning that the experimental control is set to simulation mode. It does not require a physical specimen to run. OpenSees is used as the computational driver for the hybrid simulation. The response plots from the simulation are provided for comparison.

5.1 Downloading OpenSees

The OpenSees executable and Tcl/Tk 8.5.x are required to run this example. They can freely be downloaded from the OpenSees website (http://opensees.berkeley.edu/OpenSees/user/download.php). Follow the directions carefully on the website.

5.2 Required Files

The files in the following directory are necessary.

```
User's Directory\OpenFresco\trunk\EXAMPLES\OneBayFrame\OpenSees
```

if OpenFresco was installed in the default location, the `User's Directory` is `C:\Program Files`.

The file list for the One-Bay Frame example is as follows:

- `OneBayFrame_Local_Client.tcl`
- `OneBayFrame_Local_SimAppServer.tcl`
- `OneBayFrame_Distr_Client.tcl`
- `OneBayFrame_Distr_SimAppServer.tcl`
- `OneBayFrame_Distr_LabServer.tcl`
- `elcentro.txt`

5.3 Structural Model

The model consists of two columns and a spring. The columns are element 1 connected by nodes 1 and 3 and element 2 connected by nodes 2 and 4. The spring is element 3 connected by nodes 3 and 4. A lumped mass is placed at the top of each column. The bases of the two columns are fixed, and the columns are considered axially rigid. Imperial units are used [inches, kips, sec]. Both columns are simulated experimentally. Element 1 is simulated locally during the local test and is simulated on the simulation application server side during the distributed test. However, element 2 is simulated locally during both the local and the distributed tests. The following Tcl script from `OneBayFrame_Local_Client.tcl` and `OneBayFrame_Distr_Client.tcl` defines the geometry of the model:

```
# Define geometry for model
# -------------------------
set mass3 0.04
set mass4 0.02
# node $tag $xCrd $yCrd $mass
node  1     0.0   0.00
node  2   100.0   0.00
node  3     0.0  54.00  -mass $mass3 $mass3
node  4   100.0  54.00  -mass $mass4 $mass4

# set the boundary conditions
# fix $tag $DX $DY
fix 1   1  1
fix 2   1  1
fix 3   0  1
fix 4   0  1
```

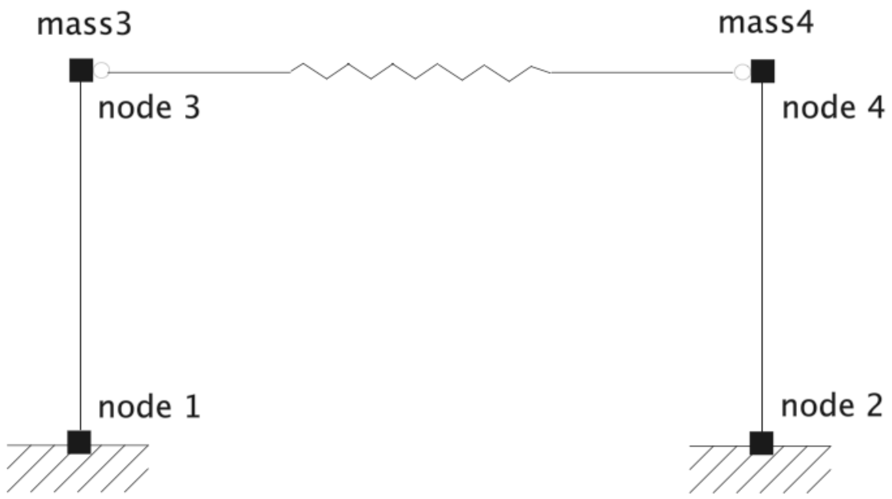


Figure 5.1: OpenSees One-Bay Frame Model.

5.4 Ground Motion

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[5]. The file, elcentro.txt, contains the acceleration data recorded at every 0.02 seconds (Figure 5.2).

[5] Chopra, A.K., "Dynamics of Structures, Theory and Applications to Earthquake Engineering", 3rd edition, Prentice Hall, 2006, 912 pp.

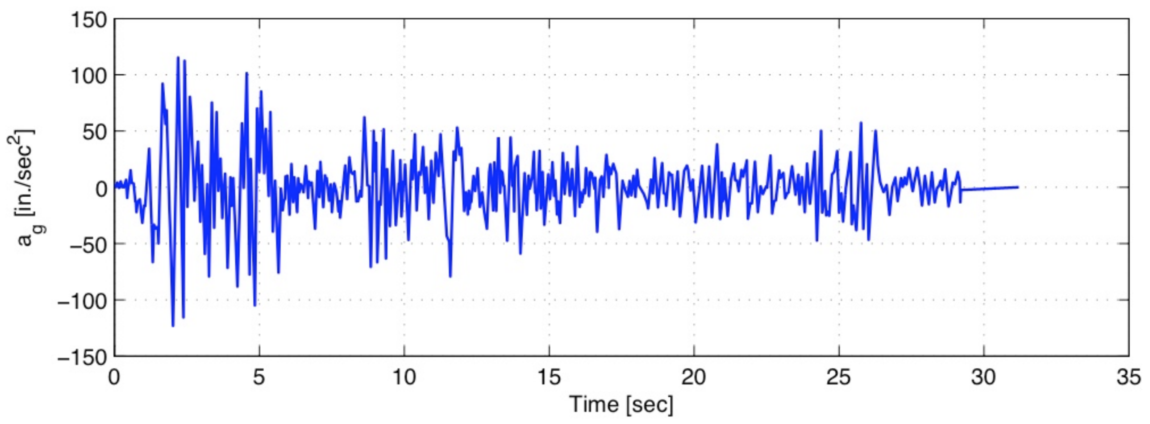


Figure 5.2: 1940 El Centro Ground Motion.

5.5 *OpenFresco Tcl Commands*

This section contains explanations of the common OpenFresco Tcl commands used in both the local and the distributed tests for element 1. The test specific commands are explained in Section 5.6. Each subsection highlights a Tcl command and the script that contains the command.

```
# Load OpenFresco package
# -----------------------
# (make sure all dlls are in the same folder as openSees.exe)
loadPackage OpenFresco
```

The above script is found in both files the `OneBayFrame_Local_Client.tcl` and the `OneBayFrame_Distr_Client.tcl`. The `loadPackage OpenFresco` is necessary for the examples to execute properly.

5.5.1 Experimental Control

The experimental control is set to be `SimUniaxialMaterials` for this example. `SimUniaxialMaterials` uses the Steel02 material, which has a matTag of 1, to simulate the response of the experimental element. The script below is in `OneBayFrame_Local_SimAppServer.tcl` for the local test and `OneBayFrame_Distr_LabServer.tcl` for the distributed test. The experimental control is defined by the Tcl command `expControl`.

```
# Define materials
# ----------------
# uniaxialMaterial Steel02 $matTag $Fy $E $b $R0 $cR1 $cR2 $a1 $a2 $a3 $a4
#uniaxialMaterial Elastic 1 2.8
uniaxialMaterial Steel02 1 1.5 2.8 0.01 18.5 0.925 0.15 0.0 1.0 0.0 1.0

# Define experimental control
# ---------------------------
# expControl SimUniaxialMaterials $tag $matTags
expControl SimUniaxialMaterials 1 1
```

The `expControl` command parameters for `SimUniaxialMaterials` are:

- `$tag` is the unique control tag.
- `$matTags` are the tags of previously defined uniaxial material objects.

5.5.2 Experimental Setup

The `OneActuator` setup is being used for the experimental setup (Figure 5.3). The script below is located in `OneBayFrame_Local_SimAppServer.tcl` for the local test and `OneBayFrame_Distr_LabServer.tcl` for the distributed test. The Tcl command for the experimental setup is `expSetup`.

```
# Define experimental setup
# -------------------------
# expSetup OneActuator $tag <-control $ctrlTag> $dir -sizeTrialOut $sizeTrial $sizeOut
<-trialDispFact $f> ...
expSetup OneActuator 2 -control 2 1 -sizeTrialOut 1 1
```

The `expSetup` command parameters for `OneActuator` are:

- `$tag` is the unique setup tag.
- `$ctrlTag` is the tag of a previously defined control object. In this case, it is SimUniaxialMaterials.
- `$dir` is the direction of the imposed displacement in the element basic reference coordinate system.
- `$sizeTrial` and `$sizeOut` are the sizes of the element trial and output data vectors, respectively.
- `$f` are trial displacement factor, output displacement factor, and output force factor, respectively. These optional fields are used to factor the imposed and the measured data. The default values are 1.0.

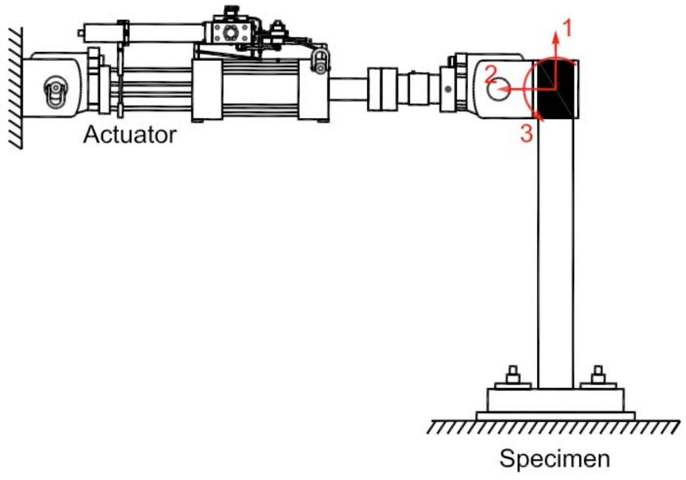


Figure 5.3: OneActuator Experimental Setup.

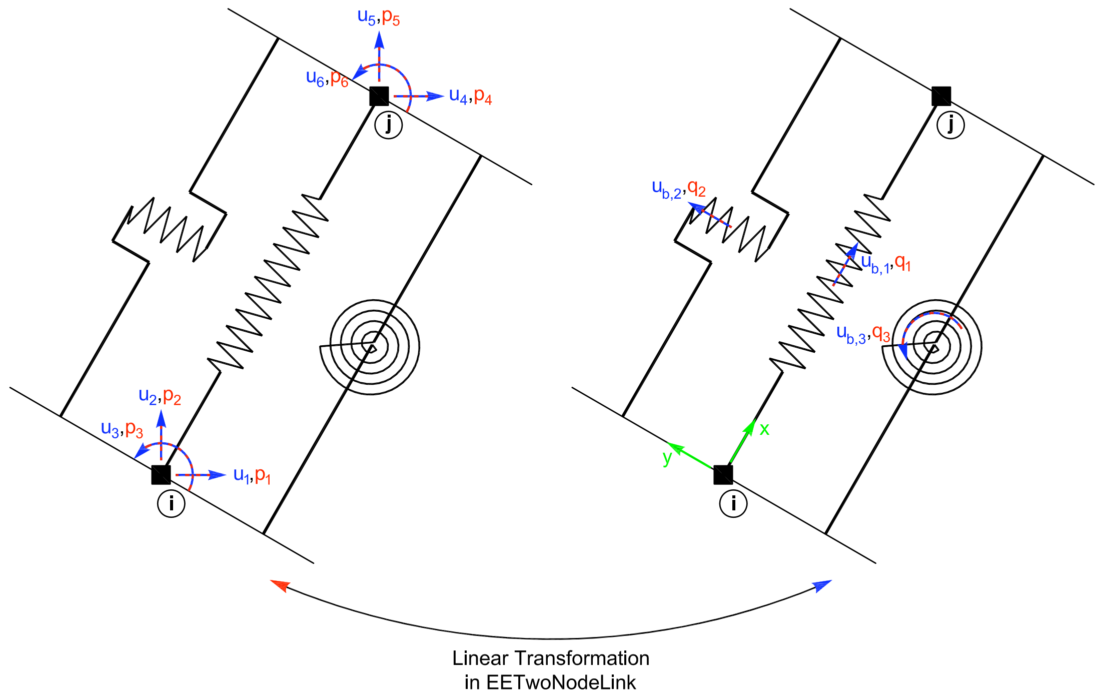
5.5.3 Experimental Element

The experimental element is set to be a `twoNodeLink` element (Figure 5.4). The script below is located in `OneBayFrame_Local_SimAppServer.tcl` for the local test and `OneBayFrame_Distr_SimAppServer.tcl` for the distributed test. Note that the experimental element is defined by the Tcl command `expElement`.

```
# Define experimental element
# ---------------------------
# left column
# expElement twoNodeLink $eleTag $iNode $jNode -dir $dirs -site $siteTag -initStif $Kij
<-orient <$x1 $x2 $x3> $y1 $y2 $y3> <-pDelta (4 $Mratio)> <-iMod> <-mass $m>
expElement twoNodeLink 1 1 3 -dir 2 -site 1 -initStif 2.8 -orient -1 0 0
```

The `expElement` command parameters for `twoNodeLink` are:

- `$eleTag` is the unique element tag.
- `$iNode` and `$jNode` are the end nodes that the element connects to.
- `$siteTag` is the tag of a previously defined site object. In this example, it is the `LocalSite` for the local test and the `RemoteSite` for the distributed test.
- `$dirs` are the force-deformation directions in the element local reference coordinate system.
- `$Kij` are the (row wise) initial stiffness matrix components of the element.
- `$x1 $x2 $x3 $y1 $y2 $y3` set the orientation vectors for the element. `x1`, `x2`, and `x3` are vector components in the global coordinates defining the local x-axis. `y1`, `y2`, and `y3` are the same except that they define the y vector which lies in the local x-y plane for the element. `<-orient <$x1 $x2 $x3> $y1 $y2 $y3>` field is optional with default being the global X and Y.
- $Mratio are the optional P-Delta moment contribution ratios. The size of the ratio vector is 4 (entries: [My-$iNode, My-$jNode, Mz-$iNode, Mz-$jNode]) My-$iNode + My-$jNode <= 1.0, Mz-$iNode + Mz-$jNode <= 1.0. The remaining P-Delta moments are resisted by shear couples. (default = [0.0 0.0 0.0 0.0])
- `-iMod` and `$m` are also optional fields. `-iMod` allows for error correction using Nakashima's initial stiffness modification. Default for `-iMod` is false. `$m` is used to assign mass to the element. Its default is zero.



$$u_{b,1} = u_4 - u_1$$
$$u_{b,2} = u_5 - u_2$$
$$-0.5L(u_3 + u_6)$$
$$u_{b,3} = u_6 - u_3$$

Figure 5.4: twoNodeLink Experimental Element.

The experimental element works in conjunction with a user-defined element in the FE software. For OpenSees, the user-defined element is the `genericClient` element. The following script is located in `OneBayFrame_Local_Client.tcl` for the local test and `OneBayFrame_Distr_Client.tcl` for the distributed test.

```
# Define client element
# ----------------------
# left column
# element genericClient $eleTag -node $Ndi -dof $dofNdi -dof $dofNdj ... -server $ipPort
<$ipAddr>  <-ssl> <-dataSize $size>
element genericClient 1 -node 1 3 -dof 1 2 -dof 1 2 -server 8090
```

The OpenFresco Example Guide contains more information on how to program a user-defined client element for other FE software packages supported by OpenFresco.

5.6 Running the Example

5.6.1 Local Hybrid Simulation

In the One-Bay Frame local hybrid simulation example, the experimental site is set to `LocalSite`. There is a client to middle-tier-server communication in this example. The script below is found in `OneBayFrame_Local_SimAppServer.tcl`.

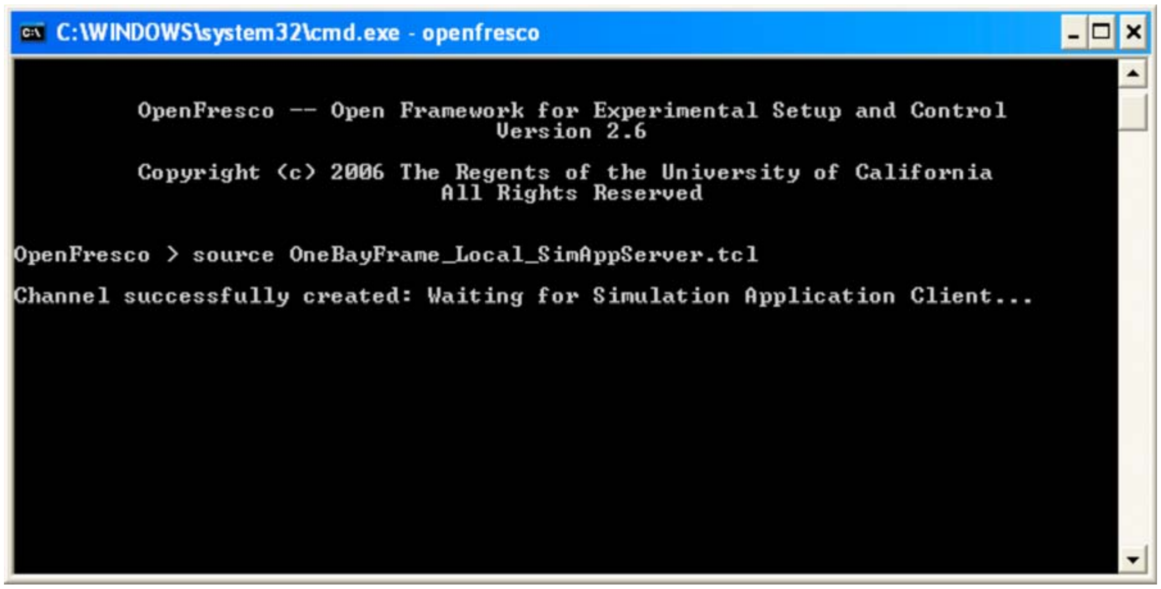
```
# Define experimental site
# -------------------------
# expSite LocalSite $tag $setupTag
expSite LocalSite 1 1
```

The `expSite` command parameters for `LocalSite` are:

- `$tag` is the unique site tag.
- `$setupTag` is the tag of a previously defined experimental setup object.

To run this simulation perform the following steps:

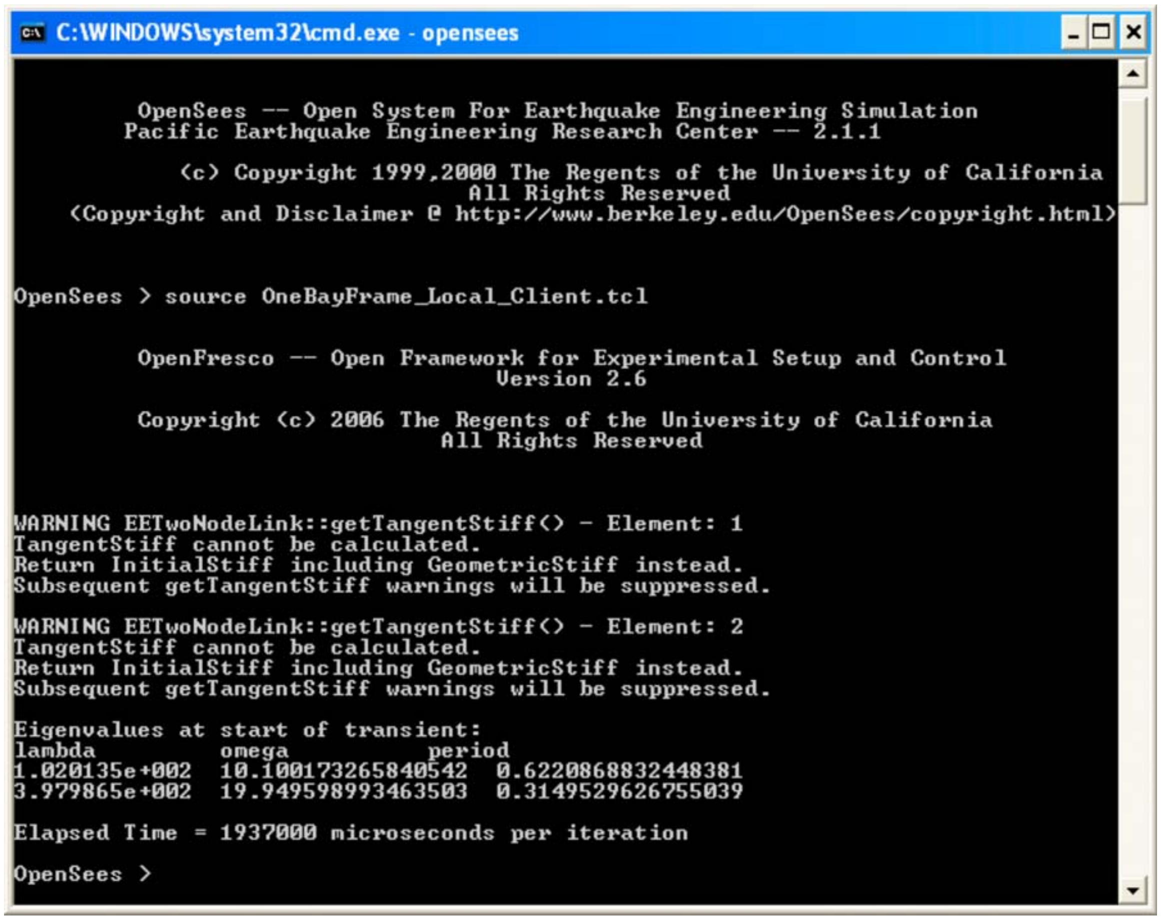
- Start the OpenFresco executable file (OpenFresco.exe) from the directory where the `OneBayFrame_Local_SimAppServer.tcl` resides.
- At the prompt, type **`source OneBayFrame_Local_SimAppServer.tcl`** and hit enter (Figure 5.5).



```
        OpenFresco -- Open Framework for Experimental Setup and Control
                                  Version 2.6

        Copyright (c) 2006 The Regents of the University of California
                              All Rights Reserved

OpenFresco > source OneBayFrame_Local_SimAppServer.tcl

Channel successfully created: Waiting for Simulation Application Client...
```

Figure 5.5: OpenFresco Window for Local Test.

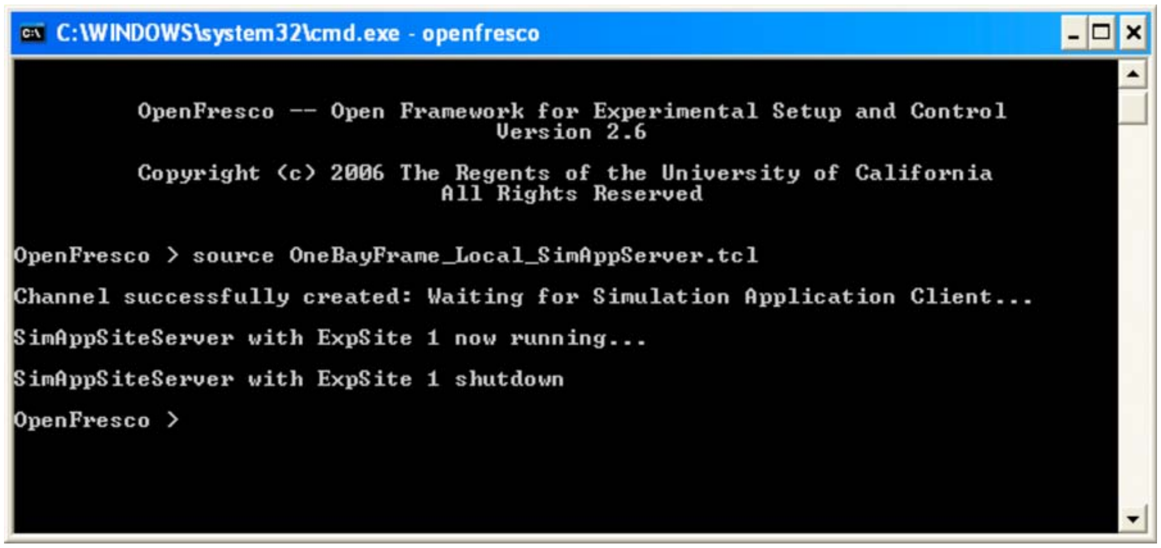
- Start the OpenSees executable file (openSees.exe) from the directory where the `OneBayFrame_Local_Client.tcl` resides.
- At the command window prompt, type **`source OneBayFrame_Local_Client.tcl`** and hit enter (Figure 5.6). This runs the simulation.



```
C:\WINDOWS\system32\cmd.exe - opensees

         OpenSees -- Open System For Earthquake Engineering Simulation
        Pacific Earthquake Engineering Research Center -- 2.1.1

            (c) Copyright 1999,2000 The Regents of the University of California
                                 All Rights Reserved
    (Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)


OpenSees > source OneBayFrame_Local_Client.tcl

         OpenFresco -- Open Framework for Experimental Setup and Control
                                Version 2.6

         Copyright (c) 2006 The Regents of the University of California
                               All Rights Reserved


WARNING EETwoNodeLink::getTangentStiff() - Element: 1
TangentStiff cannot be calculated.
Return InitialStiff including GeometricStiff instead.
Subsequent getTangentStiff warnings will be suppressed.

WARNING EETwoNodeLink::getTangentStiff() - Element: 2
TangentStiff cannot be calculated.
Return InitialStiff including GeometricStiff instead.
Subsequent getTangentStiff warnings will be suppressed.

Eigenvalues at start of transient:
lambda         omega          period
1.020135e+002  10.100173265840542  0.6220868832448381
3.979865e+002  19.949598993463503  0.3149529626755039

Elapsed Time = 1937000 microseconds per iteration

OpenSees >
```

Figure 5.6: OpenSees Window for Local Test after Simulation.

- After the simulation is complete, the OpenFresco command window should look like Figure 5.7.



```
C:\WINDOWS\system32\cmd.exe - openfresco

         OpenFresco -- Open Framework for Experimental Setup and Control
                                Version 2.6

         Copyright (c) 2006 The Regents of the University of California
                               All Rights Reserved

OpenFresco > source OneBayFrame_Local_SimAppServer.tcl

Channel successfully created: Waiting for Simulation Application Client...

SimAppSiteServer with ExpSite 1 now running...

SimAppSiteServer with ExpSite 1 shutdown

OpenFresco >
```

Figure 5.7: OpenFresco Window for Local Test after Simulation.

5.6.2 Distributed Hybrid Simulation

OpenFresco can be used to run a distributed test. A distributed test consists of a multi-tier client-server architecture. The client runs the finite element program, OpenSees, while the lab or backend server controls the physical experiment. The middle-tier servers contain information about the experimental element and moderate the communications between the client and the lab server. The script below, `OneBayFrame_Distr_SimAppServer.tcl`, shows that the `expSite` on the first middle-tier server is set to `ShadowSite`:

```
# Define experimental site
# ------------------------
# expSite ShadowSite $tag <-setup $setupTag> $ipAddr $ipPort <-ssl> <-dataSize $size>
expSite ShadowSite 1 "127.0.0.1" 8091
```

The `expSite` command parameters for `ShadowSite` are:

- `$tag` is the unique site tag.
- `$setupTag` is the optional tag of a previously defined experimental setup object.
- `$ipAddr` is the IP address of the corresponding `ActorSite`.
- `$ipPort` is the IP port number of the corresponding `ActorSite`.
- `-ssl` is an option that uses OpenSSL. The default is off.
- `$size` is the optional data size being sent.

The `expSite` is set to `ActorSite` on the second middle-tier server. The script below is found in `OneBayFrame_Distr_LabServer.tcl`:

```
# Define experimental site
# ------------------------
# expSite ActorSite $tag -setup $setupTag $ipPort <-ssl>
expSite ActorSite 1 -setup 1 8091
```

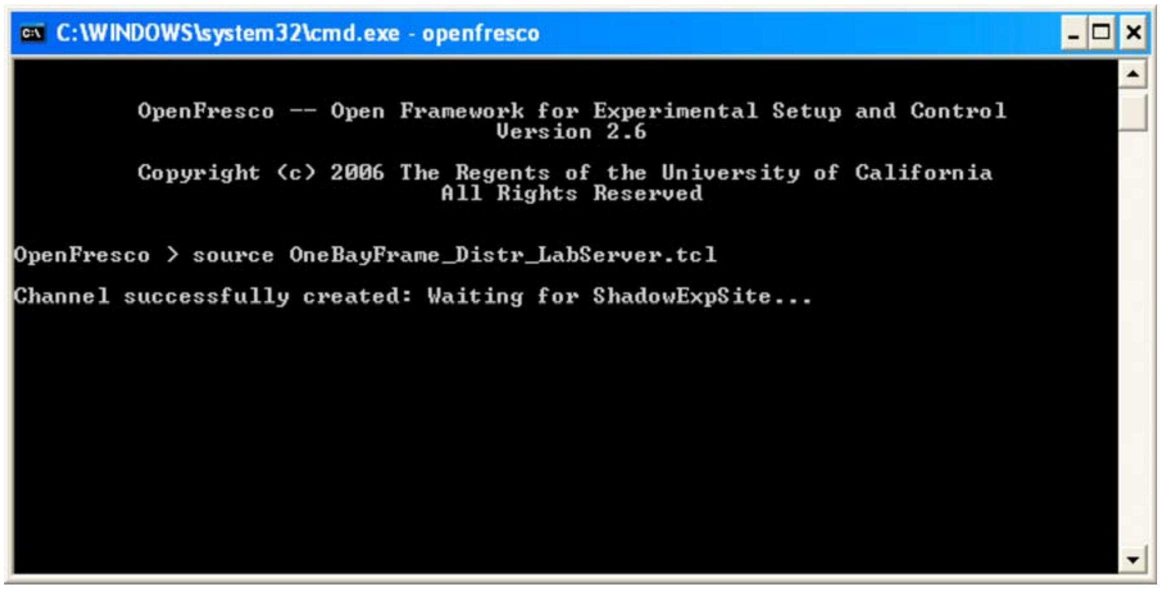
The `expSite` command parameters for `ActorSite` are:

- `$tag` is the unique site tag.
- `$setupTag` is the tag of a previously defined experimental setup object.
- `$ipPort` is the IP port number of the `ActorSite`.
- `-ssl` is an option that uses OpenSSL. The default is off.

To run this simulation perform the following steps:

- Start the OpenFresco executable file (OpenFresco.exe) from the directory where the `OneBayFrame_Distr_LabServer.tcl` resides.

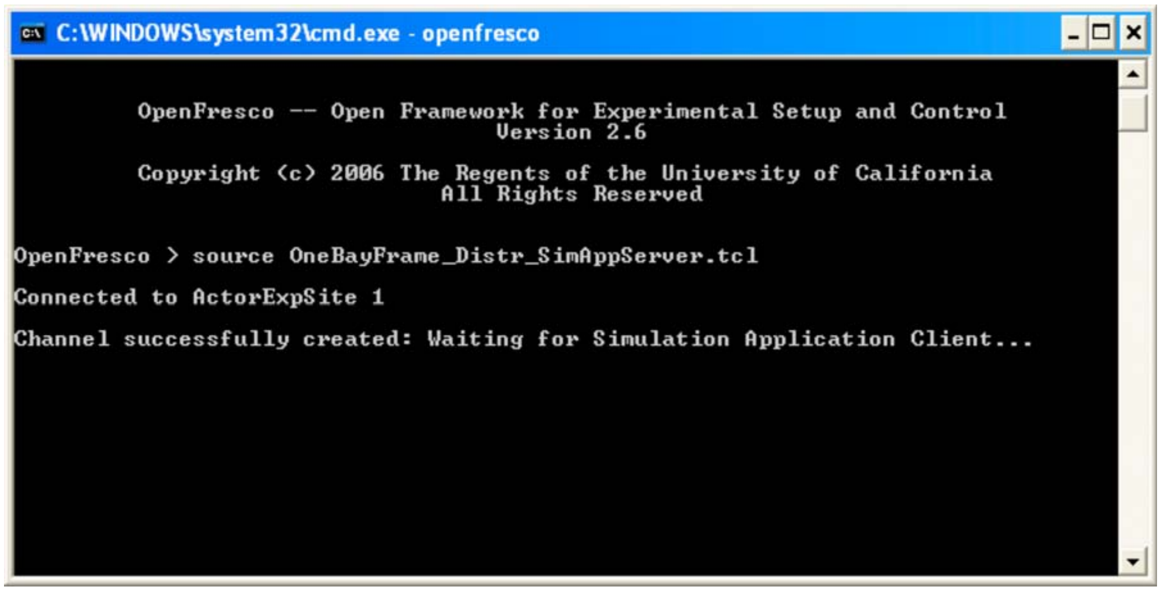
- At the prompt, type **`source OneBayFrame_Distr_LabServer.tcl`** and hit enter (Figure 5.8).



```
        OpenFresco -- Open Framework for Experimental Setup and Control
                                  Version 2.6

        Copyright (c) 2006 The Regents of the University of California
                              All Rights Reserved

OpenFresco > source OneBayFrame_Distr_LabServer.tcl

Channel successfully created: Waiting for ShadowExpSite...
```

Figure 5.8: OpenFresco Lab Server Window for Distributed Test.

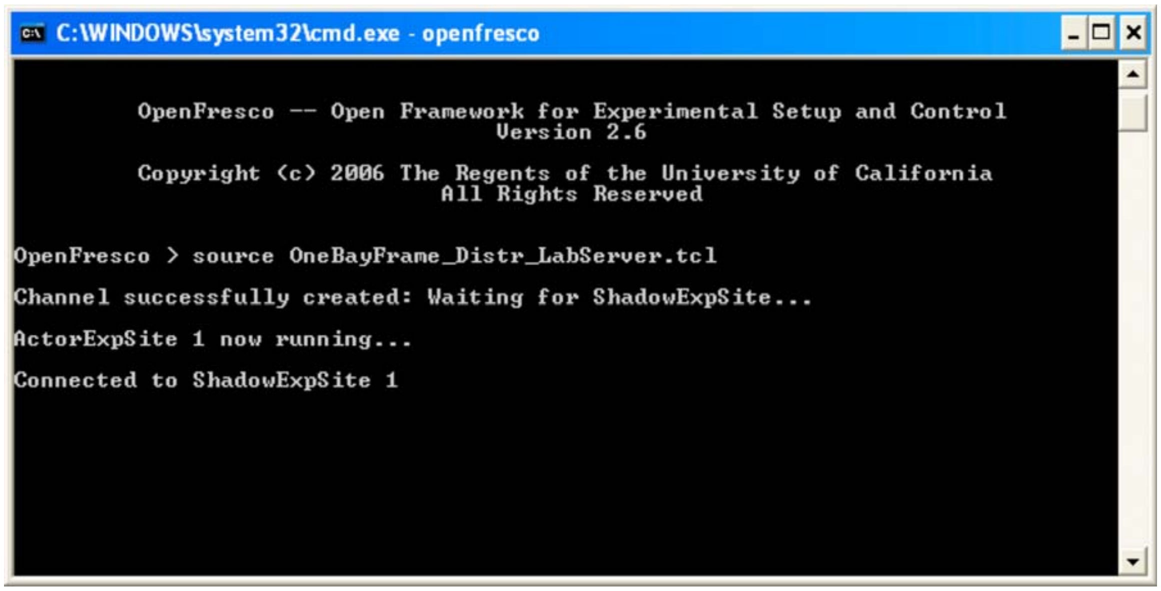
- Start the OpenFresco executable again (OpenFresco.exe) from the directory where the `OneBayFrame_Distr_SimAppServer.tcl` resides. This opens another OpenFresco command window.
- At the prompt, type **`source OneBayFrame_Distr_SimAppServer.tcl`** and hit enter (Figure 5.9).



```
        OpenFresco -- Open Framework for Experimental Setup and Control
                                  Version 2.6

        Copyright (c) 2006 The Regents of the University of California
                              All Rights Reserved

OpenFresco > source OneBayFrame_Distr_SimAppServer.tcl

Connected to ActorExpSite 1

Channel successfully created: Waiting for Simulation Application Client...
```

Figure 5.9: OpenFresco Simulation Application Server Window for Distributed Test.

- The OpenFresco lab server window now looks like Figure 5.10.



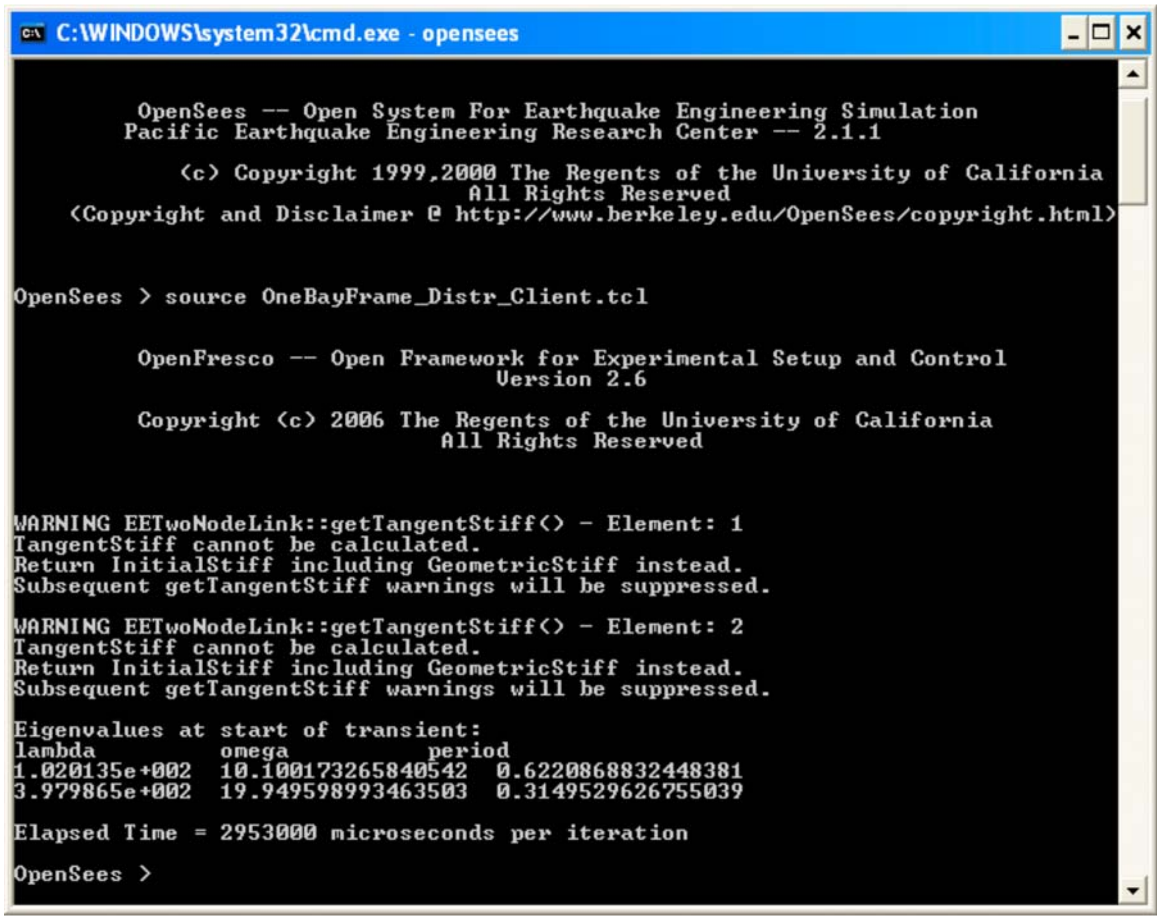
C:\WINDOWS\system32\cmd.exe - openfresco

```
         OpenFresco -- Open Framework for Experimental Setup and Control
                                Version 2.6

         Copyright (c) 2006 The Regents of the University of California
                              All Rights Reserved

OpenFresco > source OneBayFrame_Distr_LabServer.tcl

Channel successfully created: Waiting for ShadowExpSite...

ActorExpSite 1 now running...

Connected to ShadowExpSite 1
```

Figure 5.10: OpenFresco Lab Server Window for Distributed Test during Simulation.

- Start the OpenSees executable file (openSees.exe) from the directory where the `OneBayFrame_Distr_Client.tcl` resides.

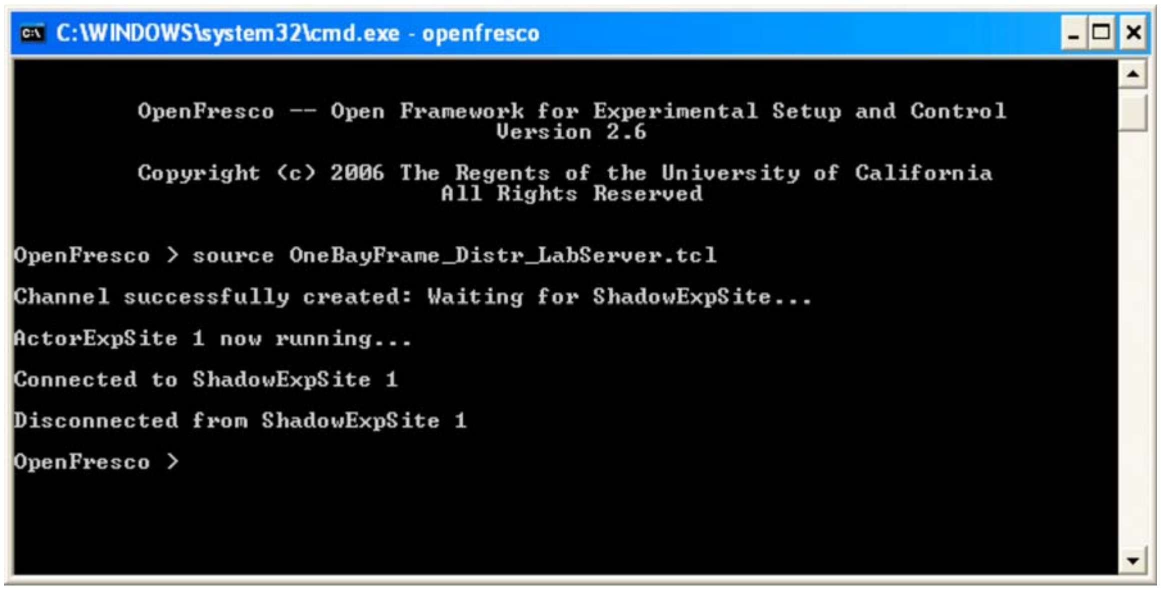
- At the command prompt, type **`source OneBayFrame_Distr_Client.tcl`** and hit enter (Figure 5.11). This runs the simulation.



```
         OpenSees -- Open System For Earthquake Engineering Simulation
        Pacific Earthquake Engineering Research Center -- 2.1.1

            (c) Copyright 1999,2000 The Regents of the University of California
                                 All Rights Reserved
    (Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)


OpenSees > source OneBayFrame_Distr_Client.tcl

         OpenFresco -- Open Framework for Experimental Setup and Control
                                   Version 2.6

         Copyright (c) 2006 The Regents of the University of California
                               All Rights Reserved


WARNING EETwoNodeLink::getTangentStiff() - Element: 1
TangentStiff cannot be calculated.
Return InitialStiff including GeometricStiff instead.
Subsequent getTangentStiff warnings will be suppressed.

WARNING EETwoNodeLink::getTangentStiff() - Element: 2
TangentStiff cannot be calculated.
Return InitialStiff including GeometricStiff instead.
Subsequent getTangentStiff warnings will be suppressed.

Eigenvalues at start of transient:
lambda          omega           period
1.020135e+002  10.100173265840542  0.6220868832448381
3.979865e+002  19.949598993463503  0.3149529626755039

Elapsed Time = 2953000 microseconds per iteration

OpenSees >
```

Figure 5.11: OpenSees Client Window for Distributed Test after Simulation.

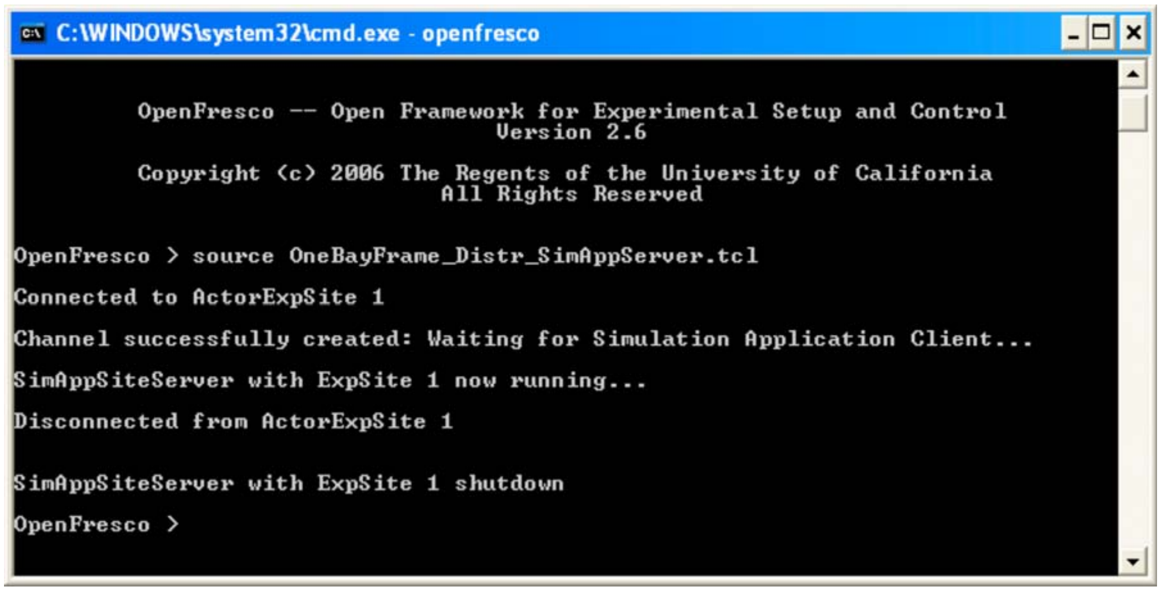
- After the simulation is finished, the OpenFresco lab server and simulation application command windows look like Figures 5.12 and 5.13, respectively.



C:\WINDOWS\system32\cmd.exe - openfresco

```
         OpenFresco -- Open Framework for Experimental Setup and Control
                                  Version 2.6

         Copyright (c) 2006 The Regents of the University of California
                              All Rights Reserved

OpenFresco > source OneBayFrame_Distr_LabServer.tcl

Channel successfully created: Waiting for ShadowExpSite...

ActorExpSite 1 now running...

Connected to ShadowExpSite 1

Disconnected from ShadowExpSite 1

OpenFresco >
```

Figure 5.12: OpenFresco Lab Server Window for Distributed Test after Simulation.



C:\WINDOWS\system32\cmd.exe - openfresco

```
         OpenFresco -- Open Framework for Experimental Setup and Control
                                  Version 2.6

         Copyright (c) 2006 The Regents of the University of California
                              All Rights Reserved

OpenFresco > source OneBayFrame_Distr_SimAppServer.tcl

Connected to ActorExpSite 1

Channel successfully created: Waiting for Simulation Application Client...

SimAppSiteServer with ExpSite 1 now running...

Disconnected from ActorExpSite 1

SimAppSiteServer with ExpSite 1 shutdown

OpenFresco >
```

Figure 5.13: OpenFresco Simulation Application Server Window for Distributed Test after Simulation.

5.7 Results

The OpenSees client window should display the following results:

```
Eigenvalues at start of transient:
lambda           omega          period
1.020135e+002    10.100173      0.622087
3.979865e+002    19.949599      0.314953
```

There are now output files from the simulation in the directory:

```
User's Directory\OpenFresco\trunk\EXAMPLES\OneBayFrame\OpenSees
```

The following are the output files:

- `Elmt_Frc.out`
- `Elmt_ctrlDsp.out`
- `Elmt_daqDsp.out`
- `Node_Acc.out`
- `Node_Dsp.out`
- `Node_Vel.out`

These files are created using the `recorder` command. Below is the script that executes these commands. The OpenFresco Command Language Manual contains more information about the experimental element `recorder` commands. For the node recorder commands refer to the OpenSees Command Language Manual on the OpenSees website (http://opensees.berkeley.edu).

```
# ------------------------------
# Start of recorder generation
# ------------------------------
# create the recorder objects
recorder Node -file Node_Dsp.out -time -node 3 4 -dof 1 disp
recorder Node -file Node_Vel.out -time -node 3 4 -dof 1 vel
recorder Node -file Node_Acc.out -time -node 3 4 -dof 1 accel

recorder Element -file Elmt_Frc.out     -time -ele 1 2 3 forces
recorder Element -file Elmt_ctrlDsp.out -time -ele 1 2   ctrlDisp
recorder Element -file Elmt_daqDsp.out  -time -ele 1 2   daqDisp
# --------------------------------
# End of recorder generation
# --------------------------------
```

The response quantities are plotted in Figures 5.14 to 5.17.

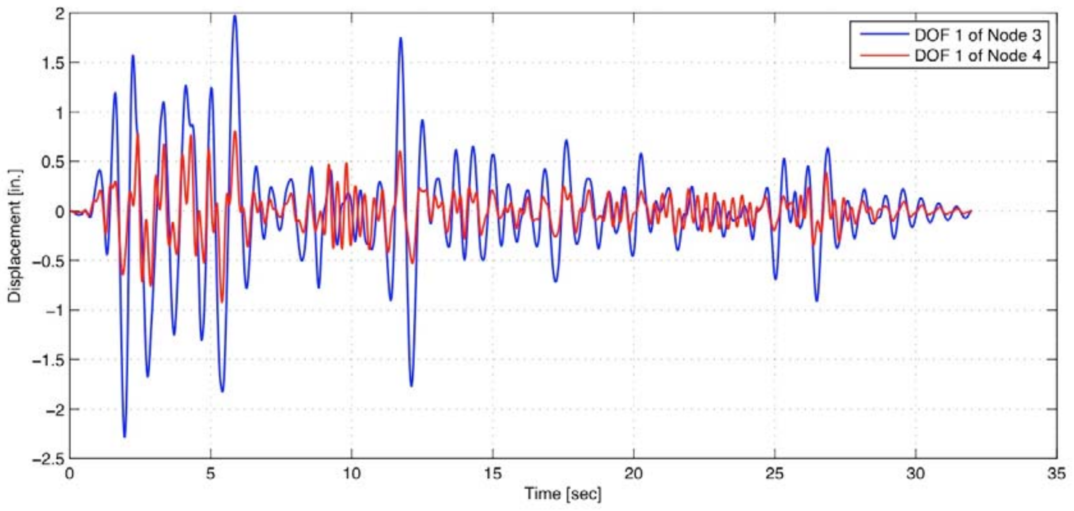


Figure 5.14: Displacement vs. Time for One-Bay Frame Example.

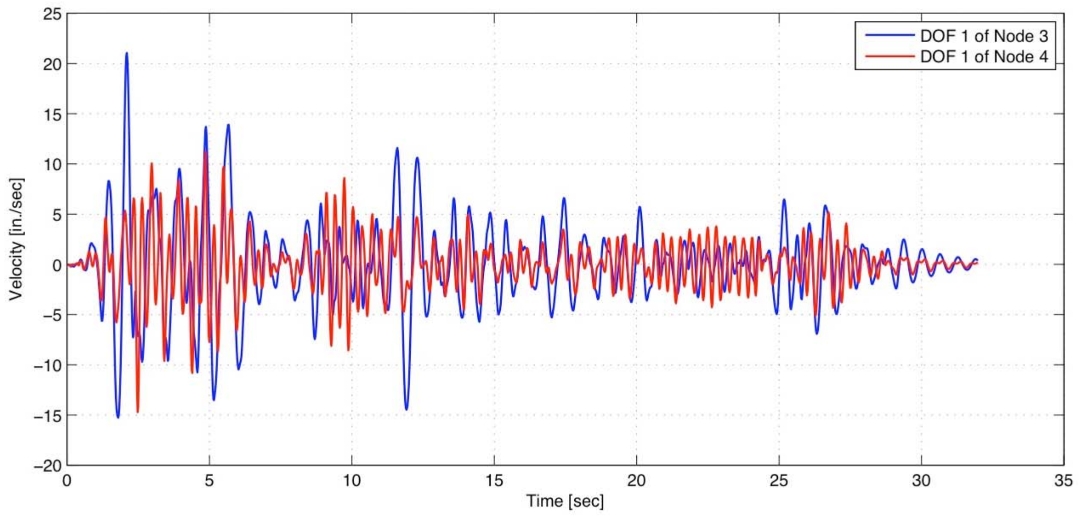


Figure 5.15: Velocity vs. Time for One-Bay Frame Example.

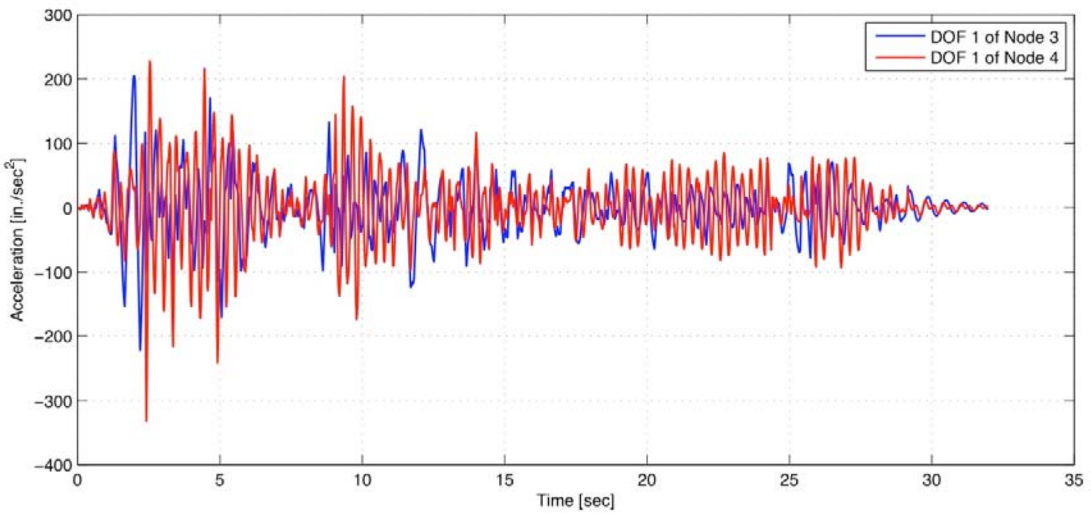


Figure 5.16: Acceleration vs. Time for One-Bay Frame Example.

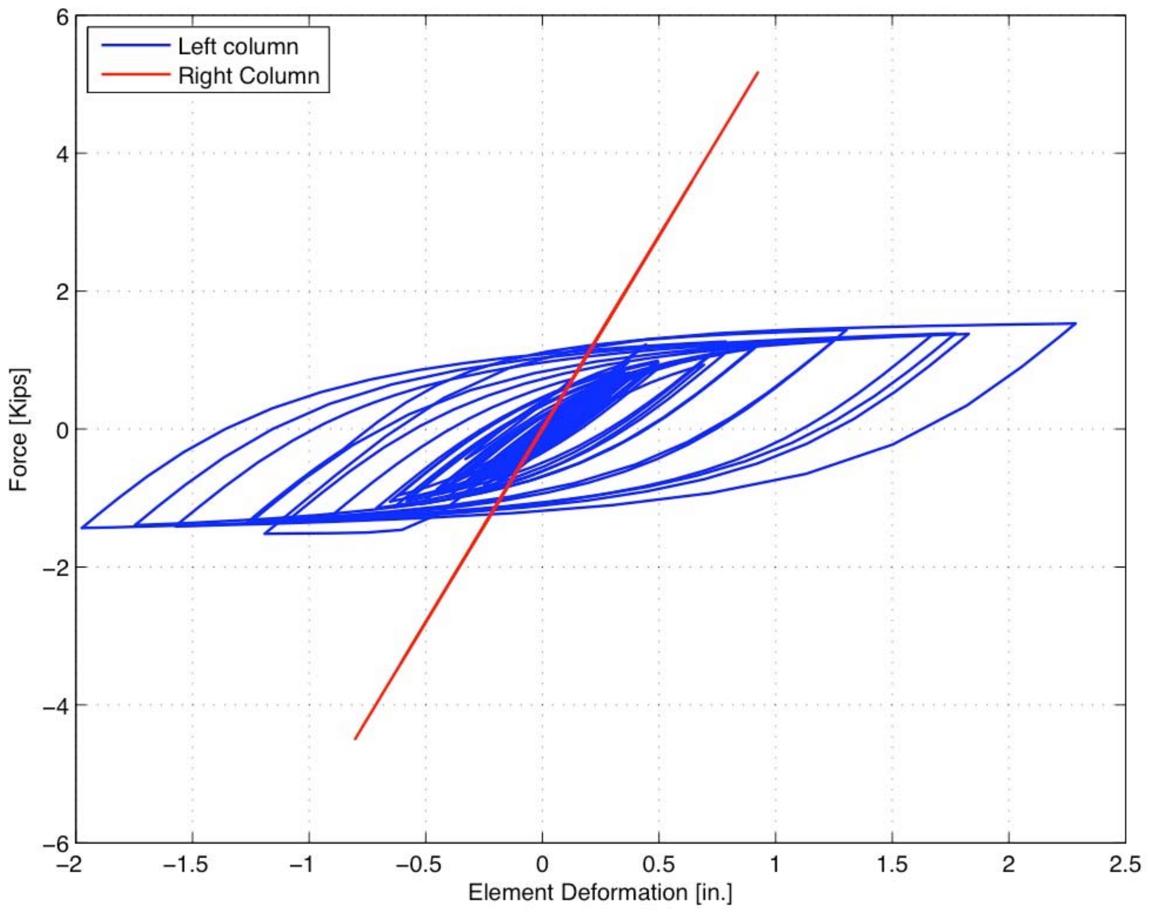


Figure 5.17: Element Hysteresis Loops for One-Bay Frame Example.

6 What To Do Next

OpenFresco is now installed. The OpenFresco Example Manuals contain more comprehensive examples that showcase other features of OpenFresco. Email Andreas Schellenberg (`andreas.schellenberg@gmail.com`) or Hong Kim (`hongkim@ce.berkeley.edu`) with any questions or feedback.