Elmer FSI Module Users' Guide
Version 0.0.1

June 23, 2015

www.illinoisrocstar.com

License

Contents

1 Introduction

Illinois Rocstar's (IR) *Elmer FSI Module* was developed to serve as a module integrating the *Elmer* finite element software with IR's multiphysics infrastructure *IMPACT*. The integration was done in order to solve a fluid-structure interaction (FSI) problem using the structures solver from *Elmer* and the fluids solver from *OpenFoam-Extend* thus demonstrating the capabilities of *IMPACT* to integrate disparate pieces of software to solve a multiphysics problem. This guide discusses only the installation and use of the *Elmer* module and its own associated test driver (not *OpenFoam-Extend* or an orchestrator for the final FSI problem).

2 Capabilities

Elmer FSI Module has the functionality of *Elmer*, but works in tandem with IR's *IMPACT* in order to share data and function cooperatively with another piece of software. There are, therefore, two pieces of software present to discuss in this guide: the *Elmer* module itself which resides within the original *Elmer* source and the driver of the *Elmer* module which utilizes *IMPACT* to communicate with the module and loads it as a shared library. The module was created to utilize the functionality of the *Elmer* nonlinear-elastic solver, enabling it to share data, communicate with an outside software, and have its time stepping externally controlled. The driver was created to be an initial test orchestrator for the module. It enables the module to run by calling its functions like an orchestrator would do when finally solving the FSI problem at hand. The most obvious difference between the final orchestrator and this test driver is that the test driver does not work with two modules (one for *Elmer* and one for *OpenFoam-Extend*), moderating their actions and relaying information between them. Rather, the driver only serves to help test the functionality of the one *Elmer* module and run it as a single piece of software.

3 Installation

Installation of *Elmer FSI Module* requires installation of several pieces of software with different requirements depending on the user's desired outcome. Installation information for the most common different possibilities of interest to users should be touched upon in this section. The most IMPORTANT piece to note for the build is that a consistent toolchain must be used throughout the entire process. Mainly, this consistency refers to the versions of `mpicc`, `mpicxx`, and `mpif90` used. The steps for installation are outlined below.

3.1 Obtain the source code

- Download the *Elmer* source from the appropriate links on https://www.csc.fi/web/elmer.
- Obtain the *Elmer FSI Module* source from IR.

  - IR internal users can check out the source from svn under svn://irsvn/Source Repository/MPInfra/Third_Party_Modules/ElmerFSI/.

- Obtain source code for *IMPACT* and *IRAD* software from IR.
  - IR internal users can check out the source code for these programs from svn under svn://irsvn/SourceRepository/IMPACT/branches/IMPACT-ICED/ and svn://irsvn/SourceRepository/IRAD/trunk, respectively.
  - Additionally, the appropriate version of these programs may already be built on /Projects, in which case, they can be accessed by linking and need not be rebuilt. However, ensure that they are in fact built with the appropriate toolchain!!!

- If the user is building the module with the intention of solving an FSI problem by integrating with *OpenFoam-Extend*, they must download the source for *OpenFoam-Extend Version 3.1* from the appropriate links on http://www.extend-project.de/.

3.2 Build *OpenFoam-Extend*

- If the user plans to integrate with *OpenFoam-Extend*, then *OpenFoam-Extend* must be built and installed **prior** to installing *Elmer FSI Module*.

- To install *OpenFoam-Extend* in the manner appropriate for integration with and installing of *Elmer FSI Module* see the IR *Users' Guide* for the *OpenFoam-Extend* module

NOTE: *GNU gcc 4.9* or greater must be used for this project!

3.3 Ensure a consistent toolchain

- The same installation of `mpicc`, `mpicxx`, and `mpif90` should be used for building all the software mentioned in this guide (*Elmer FSI Module*, *OpenFoam-Extend*, *IMPACT*, and *IRAD*).

- Set the environment variables CC, CXX, and FC to be `mpicc`, `mpicxx`, and `mpif90`, respectively.
  - Example (*C shell*):
    ```
    > setenv CC mpicc
    ```
  - Example (*Bash*):
    ```
    > export CXX=mpicxx
    ```

- If building with *OpenFoam-Extend*, extra steps must be taken.
  - Following the installation procedures for the *OpenFoam-Extend* module from the IR *Users' Guide*, *OpenFoam-Extend* will download and install its own versions of `mpicc`, `mpicxx`, and `mpif90`.

– In order to maintain a consistent toolchain *OpenFoam-Extend*'s versions of `mpicc`, `mpicxx`, and `mpif90` must be used for the building of all the software pieces. To use them, navigate to the main source directory for *OpenFoam-Extend*. Then issue the following command:

```
> source ./etc/cshrc
```

– The above command will ensure that *OpenFoam-Extend's* installation of `mpicc`, `mpicxx`, and `mpif90` are used. Double check that they are the defaults by issuing a command like

```
> which mpicc
```

3.4 Build *IRAD*

- Create a build directory for *IRAD*, and navigate into it.
- *IRAD* needs to be installed in an accessible location so that when building the following pieces of software, they may link to it. Set the environment variable CMAKE_INSTALL_PREFIX to the desired path for installation (or pass the value to *CMake* on the command line).
- Issue the `cmake` command with two arguments: the path to your *IRAD* source and the installation path set in the CMAKE_INSTALL_PREFIX variable (If not done already in the environment).

```
> cmake /PATH/TO/IRAD/SOURCE -DCMAKE_INSTALL_PREFIX=/PATH/TO/IRAD/INSTALL
```

- `> make`
- `> make install`
- For more information on using *CMake* see http:www.cmake.org.

3.5 Build *IMPACT*

- *IMPACT* needs to be installed in an accessible location so that when building the following pieces of software, they may link to it. Set the installation path for *IMPACT* when issuing the `cmake` command or by setting the environment variable CMAKE_INSTALL_PREFIX to the desired path.
- *IMPACT* requires the libraries and include files from *IRAD*. Set the path to the *IRAD* install directory when issuing the `cmake` command or by setting the environment variable CMAKE_PREFIX_PATH to the desired path.
- Create a build directory for *IMPACT*, and navigate into it.

- Issue the `cmake` command with the path to your *IMPACT* source as the argument. Also, pass the installation path set in the CMAKE_INSTALL_PREFIX variable and the path to the *IRAD* installation in the CMAKE_PREFIX_PATH variable (If not done already in the environment).
  ```
  > cmake /PATH/TO/IMPACT/SOURCE -DCMAKE_INSTALL_PREFIX=/PATH/TO/IMPACT/
  INSTALL -DCMAKE_PREFIX_PATH=/PATH/TO/IRAD/INSTALL
  ```
- `> make`
- `> make install`
- For more information on using *CMake* see http:www.cmake.org.

3.6 Preparing *Elmer* and *Elmer FSI Module* for building

- Building and installing *Elmer FSI Module* requires altering a few files within the *Elmer* source. These changes are due to the fact that *Elmer FSI Module* is built along with *Elmer* itself.
- The entire directory titled native within the *Elmer FSI Module* source needs to be copied into the following location within the *Elmer* source:

  ELMERSOURCE/fem/src/
- A patch file has been provided to assist with the changes that need to be made to the *Elmer* source. Run the provided patch file from within ELMERSOURCE/fem/src

  ```
  > patch < native/ElmerPatch
  ```
- Edit the CMakeLists.txt file within ELMERSOURCE/fem/src/ by adding the following line to the file:

  ADD_SUBDIRECTORY(native)

3.7 Build *Elmer* and *Elmer FSI Module*

- The *Elmer* libraries and its associated module library need to be installed in an accessible location so that they can be loaded at runtime when required. Set the installation path when issuing the `cmake` command or by setting the environment variable CMAKE_INSTALL_PREFIX to the desired path.
- *Elmer FSI Module* requires the libraries and include files from *IMPACT* . Set the path to the *IMPACT* install directory when issuing the `cmake` command or by setting the environment variable CMAKE_PREFIX_PATH to the desired path.
- Create a build directory for *Elmer*, and navigate into it.

- Issue the `cmake` command with the path to your *Elmer* source as the argument. Also, pass the installation path set in the CMAKE_INSTALL_PREFIX variable and the path to the *IMPACT* installation in the CMAKE_PREFIX_PATH variable (if not done already in the environment).

  ```
  > cmake /PATH/TO/ELMER/SOURCE -DCMAKE_INSTALL_PREFIX=/PATH/TO/ELMER/
  INSTALL -DCMAKE_PREFIX_PATH=/PATH/TO/IMPACT/INSTALL
  ```

- `> make`
- `> make install`
- For more information on using *CMake* see http:www.cmake.org.

As mentioned the *Elmer* libraries will need to loaded at runtime by the module driver. Ensure that these libraries are accessible to the driver by adding them to the LD_LIBRARY_PATH environment variable.

Example for *C shell*:

```
> setenv LD_LIBRARY_PATH PATH/TO/ELMER/INSTALL/lib:{$LD_LIBRARY_PATH}
```

3.8 Build driver for *Elmer FSI Module*

- If the user desires to run the test driver and the associated tests then the driver must be built. The test driver is not required, however, for running the final FSI problem with *OpenFoam-Extend* and the orchestrator.
- The *Elmer FSI Module* driver requires the libraries and include files from *IMPACT* and *IRAD*. Set the path to the install directories for these programs when issuing the `cmake` command or by setting the environment variable CMAKE_PREFIX_PATH to the desired paths.
- Create a build directory for the *Elmer FSI Module* driver, and navigate into it.
- Issue the `cmake` command with the path to the *Elmer FSI Module* driver source as the argument. Also, pass the paths to the *IMPACT* and *IRAD* installations in the CMAKE _PREFIX_PATH variable (If not done already in the environment).

  ```
  > cmake /PATH/TO/MODULE_DRIVER/SOURCE -DCMAKE_PREFIX_PATH=/PATH/TO/
  IMPACT/INSTALL\;/PATH/TO/IRAD/INSTALL
  ```

  NOTE the use of \; with no spaces to separate multiple paths.

- `> make`
- Using `make install` and CMAKE_INSTALL_PREFIX are optional for the driver.

- To ensure that the build and installation worked correctly run the provided tests for the driver and library.

  ```
  > make test
  ```

  All the tests should pass except for those regarding scaling (these tests pass and fail depending on parallel convergence and are not required).

- For more information on using *CMake* see http:www.cmake.org.

3.9 Build *OpenFoam-Extend* module and FSI orchestrator

The building and using of the *OpenFoam-Extend* module and the FSI orchestrator are not discussed in this manual. See IR's *Users' Manuals* for each of these for help.

4 Preprocessing

The test driver in essence functions as a wrapper for *Elmer* with changes made for transferring FSI data. Therefore, the preprocessing and input file requirements can be found in the *Elmer* documentation at https://www.csc.fi/web/elmer/documentation. Note that as discussed in Section 5, final time for the simulation put in the input file will be overwritten by the time passed to the driver executable on the command line. Also note that any pre-processing tools built with *Elmer* may need to be compiled and built separately from the *Elmer FSI Module* build. The driver does not wrap additional executables for pre-processing (it only calls the main *Elmer* solvers).

If running *Elmer FSI Module* for the purpose of integrating with *OpenFoam-Extend* for a coupled FSI problem, an extra step must be taken in the generation of the input files. After reviewing the *Elmer* documentation notice that there are sections within the main input file, with one entitled "Boundary Condition #," where # is the number of the boundary condition. For the FSI boundary, this section has a few added requirements in order for *Elmer FSI Module* to properly transfer the displacement and load data in and out of *Elmer*. The following line must be present in the section:

Fsi BC = True

Additionally, for any forces which need to be passed from the associated flow solver to *Elmer* the following lines must be present

Force # = Variable Time

  Real Procedure "MyLibrary" "MyFunction#"

where # is 1,2, and/or 3 for the force in the $x-$,$y-$, or $z-$direction.

Note that when using the *Elmer FSI Module* driver no load values will actually be given to the module for these forces since these would actually be provided from the flow solver. Therefore, if using the test driver and the above lines are entered in the input file, *Elmer* will receive 0.0 as the contribution for these forces on the FSI boundary.

5 Runtime

The *Elmer FSI Module* driver must be called from within the directory containing the input data like *Elmer* itself is called. Unlike *Elmer* , the *Elmer FSI Module* driver requires arguments from the command line. Currently, it must be given the '`-com-mpi`' flag and will not function without it. The other required argument is the final simulation time. This simulation time will be used to overwrite the simulation time given in the *Elmer* input file. Additionally, several simulation times may be given, in which case the driver will call the *Elmer* solvers to run up to each given time, sequentially. The driver does not restart before each new time step so the times must be monotone increasing. The reason for constructing the test driver in this manner is to mock a coupled time stepping done by an orchestrator.

The *Elmer FSI Module* driver can take two optional flags: '`-fsi`' and '`-loads`.' The '`-fsi`' flag indicates to the driver that the problem being solved is an FSI problem. The driver therefore assumes that the module is registering displacement data for it to access. The input file for *Elmer* must have Fsi Bc = True as discussed in Section 4. If this flag is given and the module was not in fact running an FSI problem (therefore registering no displacements) the driver will fail. This flag also prompts the driver to write a .vtk file with the solution data at each timestep.

The '`-loads`' flag indicates to the driver to prescribe loads onto the FSI boundary for the module to use. Like the '`-fsi`' flag, this flag should only be used if the module is in fact solving an FSI problem. The driver will prescribe loads to the FSI boundary as follows:

$$load_n = (3n + t,\ 3n + 1 + t,\ 3n + 2 + t) \tag{5.1}$$

where n is the node index and t is the time for the timestep. These loads are arbitrary and used for testing purposes.

The *Elmer FSI Module* driver is compiled into an executable called `SolverModuleDriver`. An example of calling it is shown below.

```
> SolverModuleDriver -com-mpi 5.0 7.0 9.5 13.0
```

6 Post-processing

As mentioned in Section 4 the driver essentially functions like a wrapper for *Elmer*. Therefore, any post-processing information can be found in the *Elmer* documentation at https://www.csc.fi/web/elmer/documentation. Note that any post-processing tools built with *Elmer* may need to be compiled and built separately from the *Elmer FSI Module* build. The driver does not wrap additional executables for post-processing (it only calls the main *Elmer* solvers).

7 Example Cases

The example cases used for *Elmer* found in the documentation (https://www.csc.fi/web/elmer/documentation) can also be used for the *Elmer FSI Module* driver. Non-GUI examples can be used exactly as the *Elmer* executable would be except for giving the driver the '`-com-mpi`' flag and the final simulation time on the command line. In order to replicate a GUI example, follow the instructions given in the *Elmer Tutorials* document for the example. Before running the case, generate the input file and save the project. All necessary input files and data should then be saved in the designated directory. Simply navigate to the directory and run the driver from within it, giving it the '`-com-mpi`' flag and the final simulation time on the command line. In order to determine the final time for a transient simulation multiply the Timestep intervals by the Timestep sizes found in the main input file for the example and for a steady-state simulation simply use the Steady State Max Iterations variable from the main input file as the final time value.

Example run command for GUI Tutorial 7 (Vortex shedding - von Karman instability):

```
> SolverModuleDriver -com-mpi 8.0
```

Example run command for GUI Tutorial 9 (Interaction between fluid flow and elastic obstacle):

```
> SolverModuleDriver -com-mpi 100.0
```

Additionally, a modified dataset has been created for use as a sample FSI problem for the *Elmer FSI Module* in conjuction with the *OpenFoam-Extend* module. When solved with the full coupling algorithm and the complete orchestrator it will replicate the data from the Hron Turek validation problem. The problem can be solved stand-alone with the *Elmer FSI Module* driver in which case it is almost identical to the beam example provided in the *Elmer* tutorial. The dataset for this problem can be found in the *Elmer FSI Module* source under /tesing/data/HronTurek, and the problem run with the following command

```
> SolverModuleDriver -com-mpi 1.0
```

7.1 HronTurek Beam Calculations

Here the utilization of *Elmer* includes various calculations on a HronTurek dimensioned beam. Figure 7.1 shows a typical cantilever beam of length L. In order to apply these general formulas to a HronTurek beam, $L = 0.351$.

In order to calculate the maximum deflection at the end of a beam of length L, it's useful to know the curvature of the beam:

$$\kappa = \frac{M}{EI} = \frac{\partial^2 \delta_{\max}(x)}{\partial x^2} \tag{7.1}$$

Where M is the concentrated load committing the force upon the beam, E is the Elastic

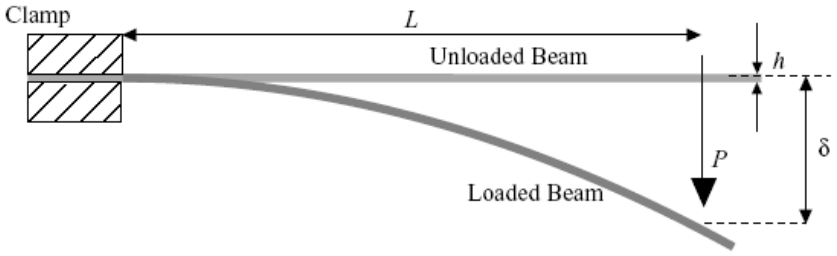

Figure 7.1: Typical cantilever beam showing length L, concentrated load P and transverse displacement δ. Graciously borrowed from: http://www.doitpoms.ac.uk/tlplib/thermal-expansion/printall.php

Modulus and I is the moment of Inertia. Solving for $\delta(x)$ where $x = L$ gives:

$$\delta_{\max}(L) = \frac{PL^3}{3EI} \tag{7.2}$$

Similar equations can be derived for a concentrated load P at any point along the beam:

$$\delta_{\max}(x) = \frac{PL^3}{6EI}(3L - x) \tag{7.3}$$

As well as for an uniformly distributed load ω:

$$\delta_{\max}(\omega) = \frac{\omega L^4}{8EI} \tag{7.4}$$

It is important to note that for the case of the HronTurek Beam with a cross sectional area of bh, the moment of Inertia (I) will be:

$$I = \frac{bh^3}{12} \tag{7.5}$$

Where h is the axis where the bending moment is applied.

7.1.1 Testing the HronTurek Beam in Elmer

Elmer utilizes the finite element method in order to discretize the beam to many sub parts via a mesh. The more nodes the mesh has (in each dimension) the better the approximation, to the analytical solution, will be. There is a trade off of computational time versus accuracy. Understandably, the more dense the mesh, the longer Elmer will take to complete its run. The analytical equations above are only valid once the reduction in error vs. the number of mesh nodes starts to converge.

For this example, it was found that Elmer closely matched the analytical solution once there were: 400 nodes along the length, 16 nodes along the base and 12 nodes along the height of the beam.

A testing script, *HronTurekBeam.csh*, was created to compare the y-component of the displacement vector of any one Elmer run to the "ideal" case (*yvals_gold.txt*) within a tolerance of 1.0E-9.

After running the Solver Module Driver (see Section 5), run the HronTurekBeam script in order to make sure your values meet the minimum tolerance.

```
> ctest -R HronTurekBeam
```

A text file *yvals_txt* should be created from the *hronturektest.ep* file created when the *.sif* file is run. From this point the script will diff the *yvals.txt* file against the *yvals_gold.txt* in order to test for the 1.0E-9 tolerance.

If the test does not pass then the message *"Test data did not pass with tolerance of 1e-9."* will appear. Otherwise, the individual test was successful.

8 Troubleshooting

If users encounter problems or have difficulties, please contact us at Illinois Rocstar.

jkress@illinoisrocstar.com