Finite Element Updating (FEMU) with Leveling

This example illustrates how to use FEMU to estimate model parameters using DIC data from a short beam shear specimen. There are four subdirectories:

- dic: Contains the DIC images and a project file
- fe: Contains the fe mesh and node locations for three load steps.
- scripts: The python code.
- femu: Contains a config.yaml file to run the process.

Quickstart

Open the sbs_coupon.z3d file in VIC-3D and analyze the images to create the DIC output files.

In an anaconda shell on windows, change to the femu directory and run

$ python ..\scripts\demo_estimate_parameters.py -c config_win.yaml -g comparison_data

On linux, simply type

$ ..\scripts\demo_estimate_parameters.py -c config.yaml -g comparison_data

Note that you may have to chmod +x the python files after unzipping. In either case, you should see an output similar to:

Building interpolation table, this might take a while...

```
Initial error: 110.2270, std. dev. 0.03219
iteration  1:    28.0979 (better)
iteration  2:     1.9092 (better)
iteration  3:     0.0922 (better)
iteration  4:     0.0690 (better)
iteration  5:     0.0588 (better)
iteration  6:     0.0402 (better)
iteration  7:     0.0233 (better)
iteration  8:     0.0173 (better)
iteration  9:     0.0167 (better)
iteration 10:     0.0167 (better)
iteration 11:     0.0167 (better)
Final error: 0.0167, std. dev.: 0.00040
Final parameters: -0.0000 0.0014 -0.0014 0.0001
```

The command above generates VIC-3D output files *comparison_data_??.out*, that can be imported into VIC-3D for visualization in the *iris* workspace. Use the *iris* sequence manager to add the generated output files as a new sequence. The output files contain the original DIC data as well as corresponding displacements and strains from the FE data. In addition, the difference between the two is available as a contour variable for all displacements and strains.

Details

The demo is a Levenberg-Marquardt optimization routine that minimizes the differences in strain between the DIC data and FE simulation by varying model parameters. For the demo, there is no finite element computation. Instead, there are four model parameters that define a 2D affine transformation. This affine transformation is applied to the example data, and the parameters that minimize the error are computed.

To use this for actual FEMU, a script or executable has to be provided that can generate the FE mesh for a set of model parameters. The optimizer stores the model parameters in a file before calling the FE program. The entire process can be controlled through the config.yaml file. The example config.yaml file is documented and most options should be self-explanatory.

There are some built-in assumptions that are worth mentioning. The mapping between DIC data and FE data is computed at the beginning and stored in an interpolation table in a file specified in the configuration. If the DIC data is changed, e.g., through a coordinate transformation, different step size, etc., the interpolation table needs to be recomputed. The interpolation table can simply be deleted from the hard drive and will be recomputed the next time it is needed.

The static interpolation table also means that the FE mesh has to remain static, i.e., the code will not work without modification if the mesh is modified.

If you have any questions or need help, please do not hesitate to call or email us at support@correlatedsolutions.com.