Workflow for OMAE Barge

The folder OMAEbarge contains input files for the barge problem described in Riggs, H. R., Niimi, K. M., and Huang, L. L. (2007). “Two benchmark problems for three-dimensional, linear hydroelasticity.” Journal of Offshore Mechanics and Arctic Engineering, 129, 149-157. If you do not have ready access to the paper, you may contact me via ResearchGate.

There is some latitude in the workflow of a hydroelastic analysis with HYDRAN-XR. The following describes one workflow for the OMAE barge. There are variations on the workflow described here, but there are 4 basic steps: 1) create the reduced basis; 2) determine the hydrostatic stiffness (if not done in step 1); 3) determine the hydroelastic response; and 4) determine the transfer functions (response amplitude operators, RAOs) of desired output quantities as well as any other desired quantities.

For these files to work, either HYDRAN-XR is restarted or a new project is started for each of the following steps. Also, the project names are the names of the corresponding input files (without the .txt).

1. **Define finite element structural model – input file eigen.txt**
   Creating a structural finite element model is the first step in obtaining the ‘dry’ (in-air) natural modes that are used for the reduced basis in the hydroelastic analysis, as well as for the determination of the hydrostatic stiffness. The natural modes for the structure ‘floating’ in air are to be determined. (An alternative is to use the modes of the structure floating in water but ignoring any ‘added mass’ from the water, i.e., include the hydrostatic stiffness in the eigenvalue analysis. I do not use this approach.)

   The first steps are to define the mesh (specify the nodes and the elements). After that, we do some important error checking: viewing the mesh and comparing the rigid body mass matrix with the expected values.

   The file **eigen.txt** does the following:
   a. Define some variables for future use. This is optional but could be useful if the values were to change later. Commands **seti** and **setr** define integer and real scalars, respectively. My personal convention is to use all upper-case names for such “constants”, but it is not necessary.
   b. Defining the structural nodal coordinates and the structural elements is typically a lot of data, so I put that in a separate file, feamesh.txt. **filein** that file to process it.
   c. Set the boundary conditions (all nodes are free). Command **bcid**.
   d. Form mass and stiffness. First, we optimize the equation numbering to minimize storage using the command **node_order**. Then **form_k** forms the stiffness (.kstr) and **form_m** forms the mass (.mstr). Profile (also known as active column) storage is used for efficiency, so these are stored in vectors.
   e. Determine the rigid body mass matrix and compare with expected values. First, we use the command **rigid_modes** to define the 6 rigid body modes with respect

to the CG. Then we take the double matrix product to obtain the 6x6 rigid body mass matrix. Note that the command **pmult** is used to multiply a profile-stored matrix with another matrix. The command **tmult** is used to multiply by the transpose of a vector or matrix.

f. View the mesh. The command **export_graphics -Gmsh** will export the mesh to be viewed in the GUI or in Gmsh (*.msh). **-Tecplot** will export to Tecplot format (*.dat). Default file name is project_name.msh (or .dat).

g. Note that there is a **return** after this step to prevent further processing until the model is checked out. Stopping here saves time if there are errors because determining the natural frequencies and mode shapes is somewhat computationally intensive and it is advisable to check the mesh first. Also, if there are problems with the mesh, then the eigensolver may fail.

2. **Create reduced basis – input file eigen.txt**

   Comment out the previous **return** to continue processing. Note that we could do this step in a different file, but inputting the mesh is quick and once the mesh is fine we might as well do it in this same file.

   We now solve the eigenvalue problem to obtain the natural frequencies (.omega) and mode shapes (.phi). For efficiency, first we convert the profile matrices to sparse format with the command **ptosparse** (profile-to-sparse).

   The command **eigval** puts the natural frequencies in the vector .omega. The mode shapes are in the matrix .phi, which is ndof × #freq and where column *i* is the nodal displacements in mode shape *i*. We print the frequencies and periods for inspection. We then save the database for subsequent processing and proceed to the next step, which is to consider the natural frequencies and view the mode shapes. The stiffness and mass matrices are first deleted to reduce the size of the database file, as these can be very large arrays for large structures.

3. **View mode shapes – input file eigen_view.txt**

   In this step we need to evaluate the natural frequencies and mode shapes to make sure they seem reasonable, a further error check. This will also help us to determine how many mode shapes we are going to use for the hydroelastic analysis. This will be based on the natural frequencies and the mode shapes.

   The command **export_graphics -Gmsh -deformed** will export the displacements in.disp. Do this for each mode shape to be viewed if using the GUI. If using Gmsh or Tecplot, an alternative is to export them all at the same time. This file does both.

   If everything looks good, continue to the next step.

4. **Process modes in preparation for hydrodynamic analysis – input file eigen_post.txt**

   This step writes the modal data in the format needed for hydroelastic analysis.

It has been decided to consider only 18 modes. Set the scalar #MODES to 18.

Read in the database from eigen using the **readdb** command.

We use the exact rigid body modes, so we replace the ones calculated by the eigensolver with the exact ones. Copy the #MODES modes to array phi. Also, save the modal displacements in the format required by the hydrodynamic analysis. That analysis expects the x-displacements to be in the matrix modex(#nodes,#modes). Similar for the other 5 nodal displacements. This step is where everything is simpler if the node numbers for the structural mesh and the hydrodynamic mesh coincide.

5. **Determine hydrostatic stiffness – input file kf.txt**
   An iterative static analysis based on gravity loads is carried out. Because small displacements are assumed, the initial position of the structure described by the input nodes **must** be a good approximation to the equilibrium configuration. Usually, this just means the position that would be the case if the structure were rigid.

   We need to redefine the structural mesh to carry out a hydrostatic analysis including hydrostatic stiffness. This was not included in the input for the eigenvalue analysis. The file feamesh-kf.txt has the few modifications required. Using a program that compares text files will quickly show them.

   To carry out the static analysis, the rigid modes of surge, sway, and yaw must be restrained (if freely floating). This requires 3 displacement degrees of freedom to be restrained via the **bcid** command (one DOF per restrained mode).

   The command **nodal_pressure** is used to define the hydrostatic pressure.

   Iterate to obtain the equilibrium configuration, with a maximum of #KF_ITER iterations. A displacement norm is used to define convergence. That is, the maximum difference in nodal displacements between two successive iterations must be less than or equal to TOL.

   Once the equilibrium configuration is determined, one needs to determine the internal structural forces (command **state**). Also, one needs to determine the final hydrostatic stiffness, transformed to the reduced basis. This is written to file modal.kf. The steps are summarized below.

   a. Redefine the mesh by fileing in a slightly modified version of the original mesh file, which includes hydrostatic stiffness.
   b. Make the structure stable by suppressing rigid body surge, sway and yaw.
   c. Define the hydrostatic pressure field.
   d. Iterate to find static equilibrium position and internal forces/stresses
   e. Move the stresses for later use.
   f. Redefine the finite element model, using very small stiffnesses so that $\mathbf{K}_s$ will be nearly zero.

g. Form the modal stiffness matrix.
h. There may be numerical errors in the hydrostatic stiffness matrix because the formulation involves a lot of cancellations. I always check the upper left 6x6 submatrix to make sure it corresponds with the values for a rigid body. If it does not, I replace that 6x6 with the values for a rigid body. Note that this is not done in this file.

6. **Determine hydroelastic response – input file hydro.txt**
   a. Convert the fea mesh to the panel mesh.
   b. Carry out a basic check of the hydrodynamic mesh via the **hyd_body_check** command.
   c. Create the mass matrix. The upper 6x6 is just the rigid body mass matrix, inserted into the mass-normalized mass matrix (identity matrix) for the flexible modes.
   d. The modal stiffness matrix (i.e., the squares of the natural frequencies) is formed.
   e. The hydrodynamic added mass, damping, and exciting forces are then determined via the **hyd_analysis** command.
   f. The hydroelastic response, in terms of the reduced basis, is then determined via the **hyd_analysis_response** command.

7. **Determine modal matrix for stresses – input file stresses.txt**
   Determine the 'stresses' (in this case, plate forces and moments) for each mode shape. Save the ones that will be desired. This file saves the stresses at 5 points. The first point has coordinates (0,0,1), which is the center of the top deck. This node has 8 elements framing into it (4 quadrilateral elements, but each quad is actually 4 triangular elements), so the average is taken. The plate forces/moments are converted to stresses and written to the file modal.stress. This file will be read in file tf when the RAOs are calculated

8. **Determine RAOs – input file tf.txt**
   a. Read the database from hydro.
   b. Define the modal response matrix, hyd_modemat, which has dimensions of #response_components x #modes. Row '*i*' of this matrix is the response of component '*i*' in each of the modes. This is multiplied by the modal coordinates to obtain the transfer functions. The formation of hyd_modemat is documented in the file.
   c. The transfer functions are written to file *.tf. In this case, only the magnitude (RAO) is written. They are also stored in array hyd_tf. Note that this is a three-dimensional array, with rows, columns, and "tables". The rows correspond to wave frequencies, the columns to wave angles, and the tables to the different components. The RAOs are conveniently viewed in the GUI with the View–X-Y Plot command.

9. **Estimate the wet natural frequencies and modes – input file wet.txt**
   For each wave frequency, finds the natural frequencies. Those within "error" of the wave frequency are reported as potentially converged. The frequencies and mode

shapes are written to the file project_name.wet. Note that the modes are expressed in terms of the dry modes.

10. **Determine water surface elevations for visualization – input file surf.txt**
    Creates .msh files for animation.