Numerical solutions to a biharmonic problem with clamped plate boundary conditions can be generated using the IFISS software package.

1. Start by reproducing the worked example in the lecture notes. This can be done by running the driver `square_plate` with the grid parameter set to `3`. You can display the solution values as a square array by simply typing `disp(u'), disp(uxy')` after computing the solution. What are the computed values of $\partial u/\partial x$ on the boundary of the plate? (Hint: `type square_plate` and check the last few lines.)

2. Next, by running the scriptfile `eigs_plate`, reproduce the results for $h = 1/16$ (grid parameter 6) and $h = 1/32$ tabulated in the lecture notes. If you compare the associated eigenfunctions for these two values of h then you should notice something strange. (Hint: having the solved the problem on the coarse grid, save the first eigenvector plot by typing `savefig('plate_eigs16')`. Then, having solved the problem on the refined grid you can then regenerate the original figure by typing `openfig('plate_eigs16')`.)

   By recomputing results with an even smaller value of h, verify that the first six eigenfunctions are the ones illustrated in the lecture notes.

3. Finally, you might like to explore the solution of the driven-cavity Stokes flow problem discussed in one of the videos. To do this you simply need to (a) set the forcing function to zero and (b) set the normal (vertical) derivative boundary condition to unity on the top of the plate.

   This can be done by modifying two files in the `biharmonic` directory:

   (a) change `ones` to `zeros` on line 11 in the file `plate_rhs`,

   (b) change `bc(k2)=0.0` to `bc(k2)=1.0` on line 21 in the file `plate_bc`.

   Test your changes by running `square_plate` with the grid parameter set to `7`. You should compute reference solution values `u = -0.1179173` and `u_xy* = 3.7082e-03`.

   You can visualise the computed streamlines by typing the following:

```
>> maxphi=max(max(u)); minphi=min(min(u));
>> contour(u',[minphi:-minphi/24:0,maxphi/6:maxphi/6:maxphi])
>> axis square, axis off
>> colormap('copper'), title('streamfunction'), shg
```

   Note the quality of the resolution of the *Moffatt eddies* in the bottom two corners.