

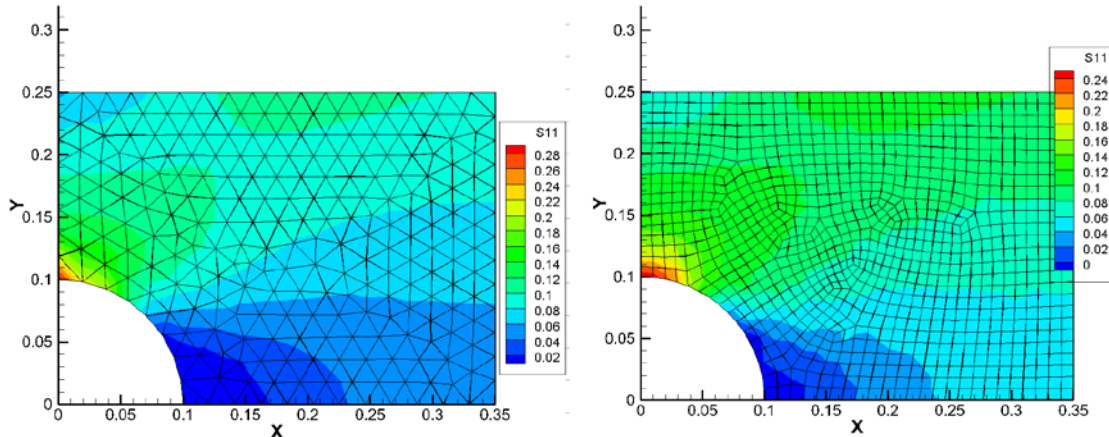


EN234: Computational methods in Structural and Solid Mechanics

Homework 3: Introduction to coding elements

Due Fri Oct 6, 2017

School of Engineering
Brown University



The purpose of this homework is to develop a very simple user element for ABAQUS. If you complete this assignment you should (1) learn how to code the calculations that compute the stiffness and internal force vector for a 2D linear elastic finite element; (2) learn how to run ABAQUS with user subroutines; (3) learn enough Fortran to be able to work with ABAQUS and (4) learn how to work with a software development environment and GitHub.

ABAQUS does not provide a development environment for user subroutines, and also will not run on a Mac. To make writing and debugging user subroutines easier, we have provided a development environment EN234FEA that duplicates enough of the behavior of ABAQUS to be able to write and debug user element / material subroutines outside of ABAQUS. Once you have your user-subroutine codes working and tested, you can run them with ABAQUS itself.

1. You can run this, and all other ABAQUS user subroutine development homeworks on the remote desktop server (you can use the Intel Parallel Studio fortran compiler)
2. If you would like to run everything locally on your own computer, install either Eclipse or Microsoft Visual studio 2013, a FORTRAN compiler (Intel parallel studio XE if you are using VS2013, or gfortran if you are using Eclipse), TECPLOT, create yourself a GitHub account, and fork and download EN234_FEA from Github. If you have a Windows computer running Win 10 you can also install ABAQUS. There are detailed instructions for all this on the EN234 website.
3. Work through the EN234FEA tutorial (first link on the programming page) to make sure you can run EN234FEA successfully.
4. Then use EN234FEA to code and test an ABAQUS format user subroutine (UEL) that implements a general 2D fully integrated (i.e. use enough integration points to evaluate the stiffness exactly) static plane elasticity element in the code. The easiest way to do this is:
 - a. Edit the file called user_codes/src/abaqus_uel_3d.for to rename the subroutine called UEL to UEL_3d (you will need to change the name at the top and bottom of the subroutine)

- b. Edit the file called `user_codes/src/abaqus_uel_2d.for` to rename the UEL_2D to UEL (Both EN234FEA and ABAQUS assume that the user element is in a subroutine called UEL. EN234FEA will now ignore the code in the `abaqus_uel_3d` file, and use the `uel_2d` file instead.
- c. Now you can add the code in the `abaqus_uel_2d.for` file to implement a 2D element. You can find detailed descriptions of the input and output arguments of the function in the ABAQUS manuals (copies are linked on the EN2340 website). To save you some tedious busy-work, the example file provided in EN234FEA includes some subroutines that will calculate integration points, shape functions and their derivatives for most 2D elements. The 3D version of the code will be a helpful guide.

Please start by writing code to solve 2D plane strain problems with displacement boundary conditions. This means you must initialize the stiffness matrix and internal force vector for the element, but don't need to write code to apply distributed forces to the element faces. In addition, to plot or print stresses, (both in EN234FEA and ABAQUS) you must initialize the SVARS() vector in the UEL subroutine to store the stresses at each integration point. Specifically SVARS(1:4) should store $\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}$ at the first integration point (in that order); SVARS(5:9) should store $\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}$ at the second integration point, and so on. You can comment out the calculations that apply distributed forces to the element faces to begin with.

To test your code, you can:

- (1) Run your user element in EN234FEA with the input file `abaqus_uel_linear_elastic_2d.in` provided (just uncomment the relevant lines of code in `main.f90`). Use the CHECK STIFFNESS option in the input file to make sure the stiffness is consistent with the residual force – check the `abaqus_uel_linear_elastic_2d.out` file in the Output_files directory. The code will also create a file called `contourplots.dat` that can be read by TECPLOT to visualize the deformed mesh and stress contours.
- (2) If you would like to solve a more interesting problem, will find input files for EN234FEA that set up a simple boundary value problem (plate with a hole, as usual), with 3 and 6 noded triangular elements, and 4 and 8 noded rectangular elements, in the `input_files/` directory of EN234FEA.

Once your user element code works in EN234FEA, please try running it in ABAQUS (you can run ABAQUS via remote desktop). Your user subroutine should be compatible with ABAQUS without modification. A sample input file (which runs a simple test by stretching 2 4 noded quadrilateral elements) has been provided for you on the homework website. To run the code with ABAQUS

- (1) Copy the `.inp` file and the `Abaqus_uel_linear_elastic_2d.for` files to the ABAQUS working directory
- (2) Follow the instructions on the course website or abaqus tutorial to run abaqus with the user subroutine
- (3) To see the results, open the file with the `.dat` extension in the ABAQUS working directory. Scroll down to see a record of the displacements at the nodes and the element state variables.

Unfortunately there is no easy way to plot the output from user elements in ABAQUS. You can save the data to a file (you can open the file with a `.fil` extension to see what the data looks like) and then read it into your own code to plot the mesh and variables that you might be interested in. Sometimes you can also overlay a mesh with regular ABAQUS elements over the top of your user elements (and give the ABAQUS elements a very small elastic modulus, so they don't influence the solution at all). There is a way to transfer the state variables from user elements to the ABAQUS elements (use the UVARM subroutine), which then enables you to see the results in ABAQUS/CAE. This involves quite a bit of coding of the most tedious kind. You don't often need to use user elements in ABAQUS (user materials are more common), but you will need to do so if you want to solve some sort of coupled multi-physics problem. For problems like this,

you might find that an open-source code designed with multi-physics in mind is preferable to ABAQUS, however.

As a solution to this homework, please hand in:

1. A one or two page description of the tests you ran to verify your code in EN234FEA and ABAQUS (please upload this to canvas). **Please include the link to your GitHub repository.**
2. Push your code up to your GitHub fork of EN234FEA. I will pull your codes and run them. Please don't upload a code that takes six weeks to run!

Optional: - if this was too easy, you could

- (1) Add the calculations to apply distributed forces on the element faces
- (2) code your element to solve plane strain, plane stress, and axisymmetric problems. To allow the user to control the choice through the input file, you can either add a variable to the list of element properties (this has to be a number, strings can't be used as properties); alternatively you can use the integer valued 'element identifier' to control the type of analysis.