



School of Engineering
Brown University

EN234: Computational methods in Structural and Solid Mechanics

Homework 10: ABAQUS user subroutines

Due Wed Dec 2, 2015

The purpose of this homework is to develop some experience with coding ABAQUS user-subroutines. In the first problem you will code an ABAQUS UMAT; in the second you will code a user element.

1. ABAQUS UMAT

Write an ABAQUS user-material subroutine to implement the simple viscoplastic constitutive equation discussed in Lecture 13. As discussed in class, the user-subroutine will provide you with the increment in strain $\Delta \varepsilon_{ij}$ and the stress and state variables (just the accumulated plastic strain for this material) $\sigma_{ij}^n, \varepsilon_e^n$ at the end of the preceding increment; you must calculate and return the stress at the end of the increment, the tangent stiffness matrix, and the state variable at the end of the increment. You just need to do the calculation for a single integration point – ABAQUS handles all the element stiffness assembly and so on.

The stress update is summarized below for convenience (by now you will be very familiar with the procedure!):

1. Compute the deviatoric strain increment and stress $\Delta e_{ij} = \Delta \varepsilon_{ij} - \Delta \varepsilon_{kk} \delta_{ij} / 3$ $S_{ij}^n = \sigma_{ij}^n - \sigma_{kk}^n \delta_{ij} / 3$
2. Compute the elastic predictors $S_{ij}^* = S_{ij}^n + \frac{E}{(1+\nu)} \Delta e_{ij}$ $\sigma_e^* = \sqrt{3 S_{ij}^* S_{ij}^*} / 2$
3. Solve the following equation for the plastic strain increment (using Newton-Raphson – there is no need to check for yield with the over-stress material model used here)

$$\sigma_e^* - Y \left(1 + \frac{\varepsilon_e^n + \Delta \varepsilon_e}{\varepsilon_0} \right)^{1/n} \left(\frac{\Delta \varepsilon_e}{\Delta t \dot{\varepsilon}_0} \right)^{1/m} - \frac{3E}{2(1+\nu)} \Delta \varepsilon_e = 0$$

4. Calculate the stress $\sigma_{ij}^{n+1} = \left(1 - \frac{3E \Delta \varepsilon_e}{2(1+\nu) \sigma_e^*} \right) S_{ij}^* + \left(\sigma_{kk}^n + \frac{E \Delta \varepsilon_{kk}}{(1-2\nu)} \right) \frac{\delta_{ij}}{3}$
5. Calculate $\gamma = \frac{3E}{2(1+\nu) \sigma_e^*} + \left(1 - \frac{3E \Delta \varepsilon_e}{2(1+\nu) \sigma_e^*} \right) \left(\frac{1}{n(\varepsilon_0 + \varepsilon_e + \Delta \varepsilon_e)} + \frac{1}{m \Delta \varepsilon_e} \right)$
6. Find the tangent stiffness

$$\frac{\partial \sigma_{ij}}{\partial \Delta \varepsilon_{kl}} = C_{ijkl}^{ep} = \frac{E}{1+\nu} \left(1 - \frac{3E \Delta \varepsilon_e}{2(1+\nu) \sigma_e^*} \right) \left(\frac{1}{2} (\delta_{ik} \delta_{jl} + \delta_{jk} \delta_{il}) - \frac{1}{3} \delta_{ij} \delta_{kl} \right) + \frac{E}{(1+\nu)} \frac{9E (\Delta \varepsilon_e - 1/\gamma) S_{ij}^* S_{kl}^*}{4(1+\nu) \sigma_e^* \sigma_e^* \sigma_e^*} + \frac{E}{3(1-2\nu)} \delta_{ij} \delta_{kl}$$

As usual the tangent stiffness can be calculated quickly using outer products. The term $\frac{1}{2}(\delta_{ik}\delta_{jl} + \delta_{jk}\delta_{il})$ is the usual diagonal matrix with 1 on the top three rows and 0.5 in the second three (for a 3D problem).

To implement this procedure:

1. Download the [sample UMAT code](#) from the website (you can delete the elasticity stuff). You can edit the file with a text editor, with Eclipse, or (now that you have installed it) Visual Studio
2. The properties for the material are $E, \nu, Y, \varepsilon_0, n, \dot{\varepsilon}_0, m$
3. The material has a single state variable (per integration point) – the accumulated plastic strain ε_e . If you like, you can also store the increment $\Delta\varepsilon_e$ as a second state variable – you can then use the value at the end of the preceding step as the initial guess for the Newton iterations. You don't need to store the stresses as state variables, because ABAQUS does this for you automatically. Note that ABAQUS will apply the rotation to the stress components before calling your subroutine, as discussed in class, so even though these equations were derived using a small-strain model, they will work for large strains (as long as the modulus is large compared to the yield stress).
4. Note that ABAQUS will call your UMAT with a zero state of stress and strain in the first increment, so just as in your hypoelasticity code you will need to trap for this state – return zero for the updated stress and state variables, and the elastic D matrix for
5. This material model is implemented in the sample MATLAB code discussed in Lecture 13, so you can more-or-less cut and paste the Newton iteration loop from the MATLAB.

To test your UMAT you can use a simple .inp file provided on the [Homework page](#) of the website that subjects a single 2D plane strain element (with reduced integration) to a prescribed stretch parallel to the x_2 direction. You can also use ABAQUS/CAE to generate a complicated .inp file and edit it to use your UMAT.

ABAQUS UEL

In this problem you will code a simple UEL that will make ABAQUS do a quasi-static analysis in a rotating reference frame (ABAQUS has a similar procedure built in, but it is less general than the one you will implement here).

The figure illustrates the problem to be solved – a solid of interest rotates at angular rate Ω (with counterclockwise convention) about an axis parallel to a unit vector \mathbf{n} , which passes through some point \mathbf{x}^0 . We can solve the governing equations in a basis that rotates with the solid. Mathematically this means we specify the position vector of a point in the solid using

$$\mathbf{y} = y_i \mathbf{e}_i = x_i \mathbf{e}_i + u_i \mathbf{e}_i$$

We assume that the components y_i are independent of time, but the basis vectors have time derivatives $\dot{\mathbf{e}}_i = \Omega \mathbf{n} \times \mathbf{e}_i$.

It is straightforward to show that the linear momentum balance equation reduces to

$$\frac{\partial \sigma_{ij}}{\partial y_j} + \rho \Omega^2 [(y_i - x_i^0) - (y_k - x_k^0) n_k n_i] = 0$$

where ρ is the mass density per unit deformed volume. We can go through the usual machinery to write this in weak form

$$\int_{V_0} J \sigma_{ij} \frac{\partial \eta_i}{\partial y_j} dV_0 - \int_{V_0} \rho_0 \Omega^2 \left[(y_i - x_i^0) - (y_k - x_k^0) n_k n_i \right] \eta_i dV_0 = \int_{A_0} t_i^0 \eta_i dA_0$$

Here ρ_0 is the mass density per unit undeformed volume. Inserting the finite element interpolation functions will reduce this to a system of nonlinear equations

$$R_i^a(u_k^b) + B_i^a(u_k^b) = F_i^a$$

Where

$$\begin{aligned} R_i^a &= \int_{V_0} J \sigma_{ij} \frac{\partial N^a}{\partial y_j} dV_0 \\ F_i^a &= \int_{A_0} t_i^0 N^a dA_0 \\ B_i^a &= \int_{V_0} -\rho_0 \Omega^2 \left[(x_i - x_i^0) - (x_k - x_k^0) n_k n_i + (u_i - x_i^0) - (u_k - x_k^0) n_k n_i \right] N^a dV_0 \end{aligned}$$

These can be solved with the usual Newton-Raphson procedure: the equations for a Newton correction dw_k^b are

$$\left[K_{aibk}^S + K_{aibk}^B \right] dw_k^b = -R_i^a - B_i^a + F_i^a$$

Where

$$\begin{aligned} K_{aibk}^S &= \int_{V_0} \frac{\partial J \sigma_{ij}}{\partial \Delta \varepsilon_{kl}} \frac{\partial N^b}{\partial y_l} \frac{\partial N^a}{\partial y_j} dV_0 \\ K_{aibk}^B &= \int_{V_0} -\rho_0 \Omega^2 N^b N^a [\delta_{ik} - n_k n_i] \end{aligned}$$

To implement this in ABAQUS, we can use ABAQUS to assemble the usual element residual force R_i^a and stiffness K_{aibk}^S (just set up an ABAQUS problem in the usual way), and write a UEL to calculate K_{aibk}^B, B_i^a .

The relevant element vectors that must be computed by the UEL can be expressed as follows

$$\mathbf{B} = \begin{bmatrix} b_1 N^1 \\ b_2 N^1 \\ b_3 N^1 \\ b_1 N^2 \\ b_2 N^2 \\ \vdots \end{bmatrix} \quad \mathbf{K}^B = -\rho_0 \Omega^2 \begin{bmatrix} N^1 N^1 (1 - n_1^2) & -N^1 N^1 n_1 n_2 & -N^1 N^1 n_1 n_3 & N^1 N^2 (1 - n_1^2) & \cdots \\ -N^1 N^1 n_1 n_2 & N^1 N^1 (1 - n_2^2) & & & \\ -N^1 N^1 n_1 n_3 & \vdots & \ddots & & \\ N^2 N^1 (1 - n_1^2) & & & & \\ -N^2 N^1 n_1 n_2 & & & & \end{bmatrix}$$

Where

$$b_i = -\rho_0 \Omega^2 \left[(x_i - x_i^0) - (x_k - x_k^0) n_k n_i + (u_i - x_i^0) - (u_k - x_k^0) n_k n_i \right]$$

For 2D plane problems rows/columns with 3 in them should be omitted. It is straightforward to code these with brute-force loops (other ways are possible too but are a headache).

Note that ABAQUS uses the same scheme for assembling element matrices as EN234FEA: the UEL should calculate $r^{el} = -\mathbf{B}$ (ABAQUS expects the user to take care of the negative sign); and \mathbf{K}^B .

Your mission is to write an ABAQUS UEL that will implement this procedure. Some notes on the procedure:

1. A template for the UEL can be downloaded from the course website. Some of the subroutines for shape functions, integration points, etc have been coded for you to save time.
2. Note that there is one important difference between EN234FEA and ABAQUS. In EN234FEA the variable called 'dof_total' stores the accumulated displacement at the *start* of a time step. In ABAQUS, the variable called U stores the accumulated displacement at the *end* of a timestep. In EN234FEA this would be equivalent to dof_total+dof_increment.
3. Implement your code to make the angular speed Ω increase with time (we assume that the rate of increase is slow enough for a quasi-static solution to be relevant). You can do this by specifying the time derivative of Ω in the input file and multiplying by time (in ABAQUS the total elapsed time is stored in TIME(2)). The purpose of doing this is to ensure that the body force will reduce if ABAQUS cuts back the time step – if you attempt to apply a large change in angular speed over a single increment the Newton iterations may not converge.
4. A simple input file has been provided that will test the procedure on a plane stress mesh with 2 elements, as shown in the figure below. The left hand boundary is prevented from moving in the x_1 direction; the base is prevented from moving in the x_2 direction, and a body force with angular acceleration 100, axis parallel to the x_2 direction passing through the bottom right corner of the block is applied (the arrow shows the rotation axis – you can think of the elements as a spinning plate). Read through the .inp file to learn how it works (in particular note the way that elements were created to apply the body force – we simply take a copy of the connectivity of the regular ABAQUS mesh, and define another set of elements that sit exactly over the top of this mesh, with the same nodes, same connectivity, etc.
5. Two additional input files for 3D problems are also provided: one sets up a small rotating hyperelastic bar meshed with 3 elements; the other does a full 3D analysis with a large number of elements, with an obscenely large deformation (the solution shown below is not the last increment).

