Text For **True*Grid*®** Short Introductory Training Course Vol. 1
by
Robert Rainsberger

This is the first in a series of videos forming the short introductory course in **True*Grid*®**. In this video we demonstrate **True*Grid*®** by building a simple part from a set of IGES files. We begin by importing the CAD geometry using the IGES command.

There are various ways to launch **True*Grid*®** based on the type of operating system on your computer. I am using WINDOWS 7. I will double click on the **True*Grid*®** icon on my desktop. On WINDOWS, a window pops up so that I can choose from any existing **True*Grid*®** command file. I have prepared one that contains the IGES command an a couple graphics commands.

Each surface in this model is a different color and there are 19 surfaces here. Our first task is to form composite surfaces: the upper, lower, left, and right side. I will remove many of the surfaces temporarily to form the upper surface. We use the cosurf command to form the composite surface and we will make this composite surface number 50.

To select the lower surfaces, we first show all of the surfaces and remove surface 50. Then we carefully remove the other surfaces not forming the lower composite surface. Once again we use the cosurf command to form composite surface number 51.

We repeat this procedure until we have isolated the left surfaces. Again we use the cosurf command and name this surface 52.

Again repeating this procedure, we form the right side and make it surface number 53 using the cosurf command.

I am showing all of the surfaces. We can label the surfaces. By clicking the number 50, for example, you see the composite surface 50 being highlighted.

After selecting a surface and clicking on the projection button, we can learn the coordinates of points on surface 50.

The next step is to build a 3 block part for the corner region of our mesh. Using the block part, I start with a 2 block by 2 block part with indices in the x-direction starting at 1 and adding 2 elements for each block. I do the same in the z-direction because we have symmetry with respect to x and z. There will only be one block in the y-direction to capture the thickness of this object.

We need coordinates for the start and end of each block which we can get by moving the mouse close to surface 50 and clicking. For example, the corner has a 0 x and z coordinate. Also the radius of this circle is 14, approximately. So I choose the 3 x-coordinates to be 0, 7, and 14. Because of the symmetry, I use 0, -7, -14, but in order to maintain a right handed local coordinate system for the z-coordinate, I will reverse the order of these coordinates. The y-coordinate goes from 0 to 2.

You can see two versions of the mesh. On the left hand side is the physical mesh showing all of the elements. On the right hand side you can see the computational mesh. It is an iconic representation of each of the blocks and how they are connected to each other. The purpose of this second view of the mesh will become clear as we begin to use it. Suffice it to say that it is our way of selecting regions of the mesh for various purposes.

Notice the correspondence between the two representations of the mesh. I can select a face that changes the color to yellow in both views. Or I can move the mouse to the I-index bar and as it gets close to a node along this bar, the corresponding face is lit up in both the computational and physical window. I now select one of the blocks in the mesh and delete it by clicking on the delete button.

The gap resulting from this deletion must be closed. The pb command is used to move the 2 opposing edges together to close this gap. One edge is selected from the computational mesh with a click-and-drag from one end of the edge to the other. Then the pb dialogue box is selected from the menus. The region of the mesh selected from the computational window is transferred to the dialogue box using the F1 function key from the key pad. The x and z-coordinates will be changed so that this edge is along the 45 degree diagonal. The coordinates along the diagonal do not have to be accurate. I will use 10 and a -10.

When I draw the picture, you can see the edge that is now along the diagonal. The same procedure is followed for the opposing edge.

The next step is to attach block edges to surface boundaries. The surface boundaries are labeled. One of the block edges is selected from the computational window, the corresponding surface boundary is selected, and the attach button is clicked. Now you can see that the edge takes the shape of the surface boundary. This is repeated for the other 3 block edges.

Now I will project the upper and lower faces of the mesh to the upper and lower composite surfaces. The surfaces are labeled. The upper face of the mesh is selected, the upper composite surface is selected, and the project button is clicked. In the same manner, the lower face of the mesh is projected. Next, I select the left side of the mesh, the left composite surface, and project. Finally, I select the right side of the mesh, select the right composite surface, and project.

I now remove the surfaces from the picture.

Element counts can be changed using the mseq command. I start by increasing the number of elements in the x-direction by specifying 4 additional elements in both blocks. I do the same in the z-direction to maintain symmetry.

The middle node where the three blocks come together is not ideal and can be improved using a smoothing feature. This is an iterative method that makes a small improvement in the smoothness and uniformity each iteration. Before the mesh can be smoothed, the miter in the mesh must be glued together using the block boundary interface command. I select the bb command from the menus, select one of the faces along the miter, and complete the entries in the dialogue box. I do the same for the opposing face of the miter. Now the two faces are glued together.

I select the unifm command dialogue box from the menus and select one face of the mesh from the computational window and click the F1 function key. Then I select a second remaining face and complete the dialogue box. This smoothing will take 20 iterations. The improvement in the mesh quality is obvious.

The same procedure is applied to smooth the bottom face of the mesh.

The block boundary interface command will also be used to glue this part to a second part yet to be made. I will number these block boundary interfaces 2 and 3.

The top face of this mesh will have a pressure applied to it. This will be done by forming a face set. I will call this face fdface.

Finally, this part will be replicated 3 times to form a full 360 degrees. This is done in 2 steps. The first step is to define the 3 transformations by rotation about the y-axis. The second step is to apply these transformations to form replications. We will not see these replications until we go into the merge phase. The 0 means keep the original part without a transformation applied to it, the 1 means make a copy of the part and apply transformation number 1 to it, and subsequently 2 and 3.

We now plan to build the second part which will form the bulk of this model. Since the minimum radius of the second part is 14 (it only needs to be greater than zero) and the part has an axis of symmetry, we can use the cylinder command to generate the mesh in cylindrical coordinates. The first coordinate will be the radius. We start with 20 elements in this radial direction. The minimum radius is 14. By picking a point near the largest radius, we see that we can use 34 as the maximum radius for this part.

The second coordinate is the angular coordinate. The mesh in the angular direction must match the first part. We will use twice the number of elements in the angular direction that were used in the first part with the intent of using the transition block boundary to make the two parts match at the interface. The angular coordinates range from 0 to 90.

We will continue to use 2 elements through the thickness.

As you can see, the default local coordinate system for the cylinder part does not match the geometry. So we need to adjust the local coordinate system using the cycorsy command. By aligning the picture so that the x-axis points towards us in the picture, we can see that the cylindrical part should be rotated about the x-axis -90 degrees.

Next, we need to move the mesh to get it into position for best projections to the surfaces. I choose the top edge with the maximum radius and select x and z coordinates for change and then I select a point from the geometry and attach. The lower edge is also moved.

Notice the amount of curvature that must be bridged by this one block. This is too much curvature for the projection to the surface to form a good mesh. So I will insert a partition and move the new partition closer to the geometry. This is done with the insprt command in the mesh menu. The insertion is in the i-direction (that is in the radial direction). I will place this new partition in the middle so that the blocks on both sides have 10 elements. By selecting each of the edges and moving them, we can get better control of the projection to the surfaces.

We now project four faces of the mesh to 4 composite surfaces.

Notice the nodal distribution in the cross section. This distortion is due to the default equal spacing along the upper and lower surfaces. We can adjust this by changing the nodal distribution rule along the upper surface so that it matches more closely the distribution along the lower surface. This is done with the res command.

I use a ratio of .95 along the left side and the reciprocal on the right side using square brackets so that I do not have to make the necessary calculation.

I now want to attach this part to part 1 using block boundary interfaces 2 and 3. This requires an additional partition through the middle in the angular direction.

With that we can now attach this part to part 1 using the transition block boundary. Just as with part replications, we will not see the results of the transitional block boundaries until we enter the merge phase.

Also, I will define block boundary number 4 to be used later in part 3 to attach to this part.

Finally, we replicate this part just as we did in part 1. That completes part 2.

The third and final part will be created using the cylinder command. Once again we will use the transition block boundary command to interface with part 2 by using 4 elements in the radial direction. The radial coordinates will range from 34 to 42. The angular direction will be the same as in part 2. We will also use 4 elements in the local z-direction.

This part, like part 2, must have its local coordinate system transformed by rotating -90 degrees about the global x-axis.

The 4 edges in the angular direction are moved into position by choosing points from the geometry and changing the x and z-coordinates.

Now we can attach the minimum z-face to part 2 using the transition block boundary command by attaching to block boundary 4.

So we have run into an error here. It is telling me that the number of elements is incorrect at the interface. So I page up in the text window and I notice that I intended to have 4 elements in the I-direction. In fact I only have 3. So I need to add 1 element and we use the mseq command in the mesh in the i-direction. We are going to add 1 element.

Now, when we draw, we no longer have the error and we are now connected to the block boundary interface.

There are many features in TrueGrid that allow you to undo and change things in a command if you run into difficulties.

Now, we project the remaining 5 faces to the 4 composite surfaces and the surface 25.

Notice that the elements at the ends of this part have less than ideal shapes. This is due to the simple default interpolation. We can do better by smoothing the mesh with the unifm command. This can be done in one command to smooth both ends using what is referred to as index progressions.

Before finishing this part, I will set nodal boundary conditions at the top face using the b command. All translational degrees of freedom will be constrained.

Finally, this part has to be replicated, like the other 2 parts.

And with that we have completed the model and we can enter the merge phase.

In the first part, we create a face set called fdface. We use this face set to apply a pressure using the pr command. This will require a load curve which we define using the lcd command in the 2D curves menu. We will need a second load curve for the material model. Now we can apply the pressure of 5.

Since there are coincident nodes in this model, we need to merge nodes with the stp command with a tolerance of .01.

We go back to the control phase to select the output format, material properties, and options for analysis. The simplest way to do this is to type in the control command. Under the output menu, I select the Lsdyna output.

Under the Options menu I select the lsdyopts command. There are 5 parameters to set.

For “Controls for energy dissipation”, choose “hourglass energy option” “hourglass energy is included in energy balance”.

For “Set hourglass control default”, choose “Hourglass viscosity type” “standard”. Also set “Hourglass coefficient” of .05.

For “Job termination”, choose “termination time” to be 20.

Finally, for “Time step control”, choose “scale for computed time step” to be .5.

This model requires a rubber material model. We go to the Materials menu and choose lsdymats. From there, choose the Frazer-Nash rubber material model which is also referred to as “Slightly Compressible Rubber Model”. From here we select different parameters.

For the element class, we choose the “constant stress brick”, “Density” 1.0e-06, “Poison’s Ratio” .49, and for all of the constants: 1,0,0,0,1,0,1,1. For “Exit” we choose “stop if strain limts are exceeded”, “Maximum strain limit” of .9, “Minimum strain limit” of -.9, and for all of the dimensions for specimen gauge use 1, load curve 2.

We can now return to the merge phase and write the LSDYNA output file. This has created the full deck for LSDYNA and that concludes this example.