Sandia National Laboratories

Operated for the U.S. Department of Energy by
Sandia Corporation

Albuquerque, New Mexico 87185-

date: January 22, 2001

Original January 15, 2001; Revised January 22, 2001, DAL.

to: CRMPC

from: Tom Baer, Multiphase Transport Processes, Dept. 9114

subject: ROT card tutorial (GT-018.1)

keywords: Rotation conditions, ROT, 3D, surface, edge, vertex, tangents, solver, seed, troubleshooting

input records: ROT, ROT = MESH SURFACE, PLANE, PLANEX, PLANEY, PLANEZ, KINEMATIC

INTRODUCTION

This tutorial is intended as an introduction to the use of ROT cards to solve a simple three-dimensional flow problem. The reader is assumed to be familiar with the structure of GOMA input decks and material files as well as the more common boundary condition types. We will consider the simple expansion of a low Re fluid as it leaves a square nozzle. The starting point is a fixed mesh solution and the input deck attached to this memo. The latter has all the requisite boundary conditions, but lacks ROT cards in order to solve the 3D problem. As the tutorial progresses, the reader will add more and more ROT cards and judge the effect on each. The end result will be an input deck that results in a convergent result.

To ROTATE or Not!

To begin this tutorial requires some brief comments on the nature of rotation and the need for ROT cards in 3D. The need for rotation of equations arises because some boundary conditions are applied in a strong sense, that is, they completely replace one component of the conservation equations (fluid or mesh momentum). On surfaces that are curved or even angled with respect to the principal axes there is some indeterminacy as to which equation component the boundary condition should replace. Replacing, for example, the x MESH equation with a KINEMATIC boundary condition on a curved free surface implies a force is being applied solely in the x direction. Unfortunately, such a force could potentially have a significant component tangential to the mesh which would result in spurious distortion of the mesh. A similar situation arises for the VELO_NORMAL card; although, now the tangential component of fluid force would result in spurious motion of the fluid tangential to the surface.

The rotation procedure alleviates these problems. In brief, on a surface feature the three mesh or momentum components are projected onto a “rotated” local basis consisting of the normal to the surface and two tangents to the surface (one tangent in 2D). These projected mesh or momentum components in the normal and tangential directions substitute the original components. Typically, the boundary condition will now replace the normal component leaving it and the two tangential components being applied at the nodes on the surface. This procedure results in forces being applied only in the normal direction to the surface and zero tangential forces thereby solving the problem discussed previously.

There still remains some indeterminacy however. First, the exact way the rotated components will replaced the original equations needs to be determined, that is, will the component in the first tangent direction replace the x y or z component etc. While unimportant when using a direct solution method, this mapping becomes crucial when using iterative solvers (see comments on pg 3). Second, for three dimensional problems there is not a unique way of picking the two tangent directions; there are an infinite number of valid choices. The manner of this choice must be specified.

The Basics

For two-dimensional problems, the tasks outlined above can be done by an automated algorithm. For this reason, the user need not worry about ROT cards when solving 2D problems. In three dimensional problems, however, the topological complexity that is possible prohibits development of an automated algorithm. We must rely upon the users judgement in resolving these uncertainties. Hence the need for the ROT card.

There are a number of functions associated with each ROT card:

1. Identifying that a surface, edge or vertex will have its equation components rotated.
2. Which boundary conditions are to be applied at the affected nodes
3. Which rotated component (normal, first tangent, second tangent) or boundary condition is to replace which of the original equation components, x, y, or z.
4. How the two tangent vectors are to be determined.

As you proceed through this tutorial and add ROT cards to the input deck, these issues will be discussed in reference to the problem at hand. But to begin…..

To illustrate and demonstrate application of ROT to a three-dimensional free surface flow, we have chosen to model a jet of very low Re fluid from a square nozzle. This is a relatively simple problem which can be readily solved and also allows for reasonable demonstration of ROT cards. In the swell3D directory, you will find three files: swell3D.inp, newt.mat, contin.exoII. The first is the input deck for problem, the second the material file for the fluid, and the last is a restart exodus II file that we have computed previously for the fixed mesh. If you blot this file you can inspect the geometry. The starting configuration is simply a box that has been meshed. You should inspect the velocity and pressure contours to satisfy yourself that a valid fixed mesh solution is present. More importantly you should *inspect the labels and locations of each of the sidesets by typing*:

```
DETOUR> ssets
DETOUR> p
```

Rotate the object using rotate x, rotate y and rotate z so you get a picture of the sidesets. The ROT cards refer to the geometry via the sidesets so it is important to have a good idea of the orientation and purpose of each one. Figure 1 of this tutorial shows an exploded view of the sidesets and should be referred to often when writing the ROT cards.

Next you should inspect the input file, swell3D.inp. You should try to connect the boundary conditions cards you see there with the side sets shown in Figure 1. In preparation for writing the ROT cards, the first thing that must be done is to identify those boundary conditions where rotation of the equation components should be applied. In the case of swell3D.inp, we can identify three rotated boundary conditions: KINEMATIC 55, PLANE 20, and PLANE 30. There are other plane-class boundary conditions, for example, PLANEZ 70 and PLANEX 50, however, these are not rotated boundary conditions. That is the distinction between plain PLANE cards and a PLANEX, PLANEY, PLANEZ. Hence, the latter do not need ROT cards. You can identify whether a BC is from Tables 6.2, 6.3, 6.4, and 6.5 in the GOMA manual. If the boundary condition doesn't appear there, you can also look in the file mm_names.h. If in the information associated with the boundary condition, you find descriptors like R_MOM_NORMAL, R_MESH_NORMAL, R_MESH_TANG1, the boundary conditions is rotated. If you find a NO_ROT descriptor, the BC is unrotated.

For solution, those sidesets on which rotate boundary conditions are specified require, at the minimum, a SURFACE ROT card to be written. In this case, you must write ROT cards for three sidesets 55, 20 and 30. GOMA searches for all ROT cards after the "Rotation Specifications =" cards. When the "END OF ROT" card is encountered GOMA quits search for ROT cards. If you inspect swell3D.inp, you will see that there are currently no ROT cards for GOMA to read. We shall add the necessary cards as part of the tutorial.

We'll begin this discussion by writing the complete SURFACE card for sideset 20 and then discussing it in detail:

Edit swell3D.inp and type the following ROT card after "Rotation Specifications =".

```
ROT = MESH SURFACE 20 PLANE 20 T2 0 T1 0 SEED 0 0 1
```

The "ROT=" is a required construct and alerts GOMA to parse what follows the equals sign as a ROT card. The string "MESH" identifies this card as being applied to the pseudo-solid mesh motion equations. Hence, all rotation of equations or replacement of equations will occur to the mesh equations.

The string "SURFACE" identifies this card as applying to nodes on a sideset identified by the integer read in after it, in this case, 20. Following the sideset id, GOMA expects to read three {string} {integer} pairs. In this case, PLANE 20 T2 0 T1 0. This information is used in two ways. First, it identifies exactly what is going to be applied at each node affected by this ROT cards. In this case, the mesh equations at each of these nodes will be rotated, the first and second tangential components (this is what is meant by T1 0 and T2 0, respectively) will be applied, but the normal component will be replaced by a PLANE 20 constraint equation.

The order of the string-integer pairs is also important. It determines which of the original unrotated equation components is replaced by which rotated equation component or boundary condition. The first of the string-integer pairs is associated with x equation (in this case, the x mesh displacement), the second string-integer pair is associated with y equation (y mesh displacement) and the last pair is associated with z equations (z displacement). So for the card above, the x equation is replaced by the PLANE 20 constraint, the y equation by the second tangential component and the z equation by the first tangential rotated component.

Why is this association important? In the case of using a direct solver, it isn't. The problem should work equally well no matter what association is chosen. However, it is rare for a 3D free surface problem to be small enough for a direct solver to appropriate. More often iterative matrix solvers are employed, and in these cases the nature of this association can be critically important. The issue is diagonal dominance. Iterative solvers work best when the largest terms in any equation are near the diagonal of the matrix. We can use the associations allowed by the ROT cards to influence the diagonal dominance of the resulting matrix. The rule is:

> **"Try to associate the boundary conditions with the components they are most sensitive to. Never associate a boundary condition with a component to which it is completely insensitive."**

This rule is why we chose to associate the PLANE 20 boundary condition with the x equation pair. If you consider Figure 1, you will see that the x-axis is normal to sideset 20. Therefore, PLANE 20 has been set up with the following coefficient list: 1.0 0.0 0.0 0.0 (check the input deck!). Clearly, the constraint is only sensitive to changes in the x displacement of a node (nodes displacing in the x direction change the value of the constraint, node displacing in the y or z direction do not change the value of the constraint). We have no other choice but to make the PLANE 20 the x component string-integer pair.

Why the first and second tangential components are associated as they are related to the last portion of the ROT card: the SEED specification method. Choosing a seeding method is necessary for solution of three dimensional problems because, unlike two dimensional problems, choice of two unique tangents to accompany the normal vector in the rotated basis system is not a unique process, that is, there are an infinite number of choices. By choosing a seeding method, the user helps GOMA to make this choice. Seeding can be done manually as in the preceding example. The syntax for this method is SEED sx sy sz. When this method is chosen, GOMA projects the vector s = (sx, sy, sz), which doesn't have to be normalized, into the surface to get the first tangential direction. For those of you excited by tensor notation (and those who aren't), the operation is as follows:

$$t_1 = (I - nn) \cdot s \qquad \text{(EQ 1)}$$

where n is the well-defined outward-pointing normal to the surface. The second tangential direction is determined from the cross-product of the normal and the first tangential direction:

$$t_2 = n \times t_1 \qquad \text{(EQ 2)}$$

The one very major caveat when using this manually seeding method is that at no point on the surface of interest should the seed vector be normal, or even close to normal, to it. The reason is obvious from the first equation: the projection of a vector nearly normal to a surface into the surface is not a very well defined vector. Of course, for some surfaces it can be hard to specify a seed vector that meets the criteria. In these cases, GOMA offers the BASIS_RESEED option. This seeding method relies upon the element boundaries in the surface to compute the first tangent vector. Averaging is performed over all elements that share a specific node, so that each node will have a consistent first tangent direction. The second tangent direction is still determined from the normal and first tangent as noted above.

In the example we have given, however, we have chosen to use the manual seeding method because it is easy to find a vector that is never normal to a planar surface. In this case, we have chosen the vector (0,0,1) or the z basis vector. Since sideset 20 is a plane perpendicular to the x axis, the seed vector projects unchanged to (0,0,1). As a result, the first tangent direction is the z direction itself. The second tangent direction is therefore the y direction. Thus, you should be able to see why T2 0 was chosen as the y component string-integer pair, and T1 0 was the z component integer pair. Because of the seed vector choice, the first rotated tangent component is effectively the z component of the mesh equations and it makes sense that it should be associated with the z component string-integer pair (this would be the case if no rotation where applied at this node after all). Similar reasoning holds for the second tangent direction.

Now add to swell3D.inp the following ROT for sideset 30 after the card for sideset 20:

```
ROT = MESH SURFACE 30 T2 0 PLANE 30    T1 0   SEED 0 0 1
```

Hopefully, the syntax here will be a little clearer. The only thing that has changed from the previous cards is that PLANE 30 BC has been associated with the y direction and T2 0 with the x direction. Once again if you inspect the input deck or look at Figure 1, you will see that in this case sideset 30 is perpendicular to the y direction and hence the PLANE condition associated with it is most sensitive to y mesh displacements. We have used the same SEED vector as before, so T1 0 should be placed as before at the z position, but now because the normals to the sideset are in the y direction the second tangent direction is in the x direction, and hence its placement at the x position.

Next add after the previous two ROTs the following card for the free surface sideset 55:

```
ROT = MESH SURFACE 55 KINEMATIC 55 T2 0 T1 0   SEED 0 0 1
```

In this card, instead of a PLANE card we have the KINEMATIC card for sideset 55. Otherwise, the card is identical to that for sideset 20. Note that sideset 55 consists initially of two perpendicular surfaces, one a constant x surface, the other a constant y surface. The sideset resembles a piece of angle iron. As for a PLANE command, the sensitivity of the KINEMATIC constraint is greatest in the direction of the outward pointing normal. In this case, we have a choice of whether to put the KINEMATIC card at the x position or the y position since it is potentially sensitive to both. Actually it doesn't matter because the x and y portions of the sideset are of similar size. It would be different if one or the other where more extensive. You might also wonder about the fact that putting the KINEMATIC card at the x position would introduce poorly conditioned equations (not diagonally dominant) for those nodes on the y portion of the sideset. The answer to this is yes. In this case weare relying on the robust of the iterative solver preconditioners to save the day and produce a result eventhough the matrix is not optimally conditioned. Note, of course, that in many problems, they are not up to this task. Note also that the KINEMATIC 55 BC is not sensitive at all to the z mesh displacements so it should not be put at the z position.

Run GOMA on the problem using the command:

```
% goma –i swell3D.inp
```

You should see that GOMA completes on iteration and then halts with a mesh folding error. We indicated that we wanted the intermediate solutions to be written to the exodus II file so the deformed mesh should reside in the file out.exoII.

Blot out.exoII. Move into DETOUR and plot the result.

You should see the very deformed mesh. This offers little information in diagnosing the problem so *scale down the deformation by using the mag DETOUR option.*

```
DETOUR> mag 0.01
```

Rotate the figure so you can see the free surface side of the mesh. Paint the DMX contours. What is immediately apparent is that there is a large displacement of the mesh at the seam where the free surface intersects the solid nozzle surfaces. This occurs because we have been careless in use of the SURFACE ROT card for sideset 55. This ROT card applies to all nodes on sideset 55 which includes those along this seam. Therefore nodes on this seam are having a KINEMATIC boundary condition plus two rotated components applied to them. That is bad because we really would like to have these nodes stay in the planes defined by PLANEX 50 or PLANEY 40 whichever is appropriate. Worse still, the velocity at these nodes has been set to zero by the no slip conditions on nodesets 40 and 50 so the KINEMATIC boundary condition is ill-defined there because there is no velocity field. The result is the mesh folding seen.

What we need to do is override the boundary conditions specified by the surface card with some that are specific to the nodes only on that seam. This is done using a ROT EDGE card.

Type the following cards after the SURFACE 55 ROT card:

```
ROT = MESH EDGE 40 55  T 0 PLANEY 40 PLANEZ 70 NONE
ROT = MESH EDGE 50 55  PLANEX 50 T 0 PLANEZ 70 NONE
```

To save on typing, you may copy stub cards that lack the boundary conditions from the end of the input deck.

These are EDGE ROT cards and apply only to those nodes that lie on the intersection of two surface sidesets which are specified following the EDGE string, in this case, 40 and 55 or 50 and 55. If you inspect Figure 1 you will see this is the seam where the free surface intersects the nozzle boundaries.

As in the case of the SURFACE cards the boundary conditions listed are those that will be applied to those nodes. So in the case of the 40 55 EDGE this is the PLANEY 40 BC and the PLANEZ 70 BC. Look at Figure 1 to locate sideset 70. It is the plane that is transverse to the flow through the plane of the die exit.

The new string/integer pair, T 0, is specific to the EDGE card. It tells GOMA to use a rotated component of the equations which is the projection of the original equations in the direction of the tangent vector that lies along the curve of the seam. For this reason, the seed method chosen is NONE because this tangent direction is well defined from the geometry itself and needs not the user to intervene.

Note also the association of the PLANEX 50 card with the x position, the PLANEY 40 with the y position, and the PLANEZ 70 card with the z position. This again is following the sensitivity rule described above.

Run GOMA again using the same command.

This time GOMA should run for a few completed iterations before finally giving up with another folded mesh error. *Run blot on out.exoII. Invoke DETOUR. Paint DMZ.*

Without even rotating the figure, we can tell that there is still something wrong. Our first view is of the exit plane and it shows that at the edges of this plane there is significant displacement in the z direction. But this exit plane (SS 60) is a constant z plane. The mesh nodes should remain in it and not displace out of it. What gives?

Again it has to do with the SURFACE ROT cards. The nodes at the edge of sideset 60 are alternatively also parts of sidesets 20, 30 and 55. Therefore, they are subject to the SURFACE cards we wrote for those sidesets. The PLANEZ 60 card does not appear on these SURFACE ROT cards so it is not applied to nodes on those edges. The solution to this is as previously: add an EDGE ROT card.

Ty*pe or paste the following EDGE ROT cards:*

```
ROT = MESH EDGE 60 55 KINEMATIC 55 T 0 PLANEZ 60 NONE
ROT = MESH EDGE 60 20 PLANE 20 T 0 PLANEZ 60 NONE
ROT = MESH EDGE 60 30 T 0 PLANE 30 PLANEZ 60 NONE
```

If you compare the assignments made on these cards with those made on the SURFACE cards for 55, 20, and 30 you will see they are quite similar. Only the T1 0 is replaced by the PLANEZ 60 and the T2 0 is replaced by the edge tangent string/integer pair, T 0. With these cards in place, now the nodes on the edges of sideset 60 will receive a PLANEZ 60 constraint. Zero shear stress along the edge direction will also be maintained.

Before running GOMA again, we might reflect a bit on what we have learned and whether there are some other edges that might also benefit from EDGE ROT cards. Well, there are three that might pose a problem: 55 20, 55 30, and 20 30. Taking the 55 20 edge as an example, since there are no

EDGE cards yet assigned to it, it's boundary conditions will be those assigned by the first SURFACE ROT card that includes those nodes. In the 55 20 case, this is the SURFACE 20 ROT card. Therefore, nodes on this edge will be assigned a PLANE 20 BC and two rotated mesh equation components. But these nodes are certainly on the free surface so they should have a KINEMATIC BC applied to them. So we need an EDGE card to permit this.

Type or paste after the other EDGE cards the following:

```
ROT = MESH EDGE 55 20 PLANE 20 KINEMATIC 55 T 0 NONE
```

This is what we want on this edge because it will cause the nodes on this edge to stay in PLANE 20 but respond to the KINEMATIC 55 BC. Further, zero mesh stress is imposed in the z direction by the T 0 pair at the z position. Note that on this card the KINEMATIC 55 is associated with the y position. Following our rule, we should never associate PLANE 20 with the y position, that leaves only KINEMATIC 55 for the y position, since neither BC should be associated with the z position.

By similar reasoning, we note that the 55 30 and 20 30 edges will also need EDGE ROTS.

Type or paste after the other EDGE cards the following:

```
ROT = MESH EDGE 55 30 KINEMATIC 55 PLANE 30 T 0 NONE
ROT = MESH EDGE 30 20 PLANE 20 PLANE 30 T 0 NONE
```

You should be getting quite good at these by now. So it is left as an exercise to figure out these EDGE cards.

Run GOMA again.

This time GOMA struggles for a bit longer but eventually gives up in another mesh folding episode.

Blot out.exoII, invoke DETOUR, mag 0.01, rotate y –45. and paint dmx.

What you should see is something odd going on at the corners of the exit plane sideset 60. The nodes at these points are not behaving well at all. Again the reason goes back to the fact that they are not constrained sufficiently, this time by the EDGE cards that apply to them.

Take as an example the node at the intersection of sidesets 20 30 and 60. Because of the ROT card precedence rules, the BCs that are applied to it are those associated with the first EDGE card in the input deck that would include that node. If you go back through the ROT cards, you will find that the first EDGE card that applies to this node is:

```
ROT = MESH EDGE 60 20 PLANE 20 T 0 PLANEZ 60 NONE
```

Which means that this node is constrained by the PLANE 20 and PLANEZ 60 cards to remain in those planes, but zero mesh force is applied to it in the y direction. Hence, we see the node here drifting upwards in the y direction in response to loads applied elsewhere to the pseudo-solid.

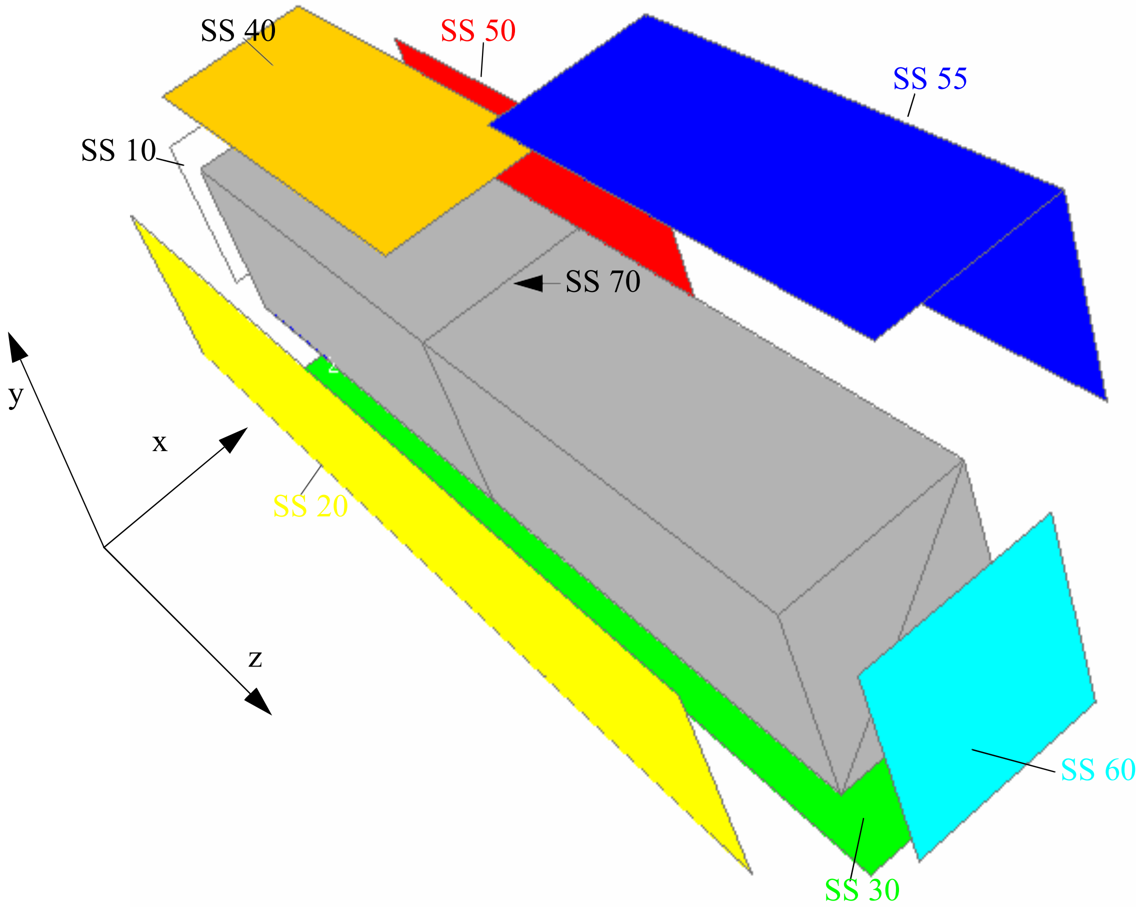

Figure 1. Exploded view of sideset configuration for 3D die swell problem. Note that sideset 70 is embedded in the mesh and is transverse to the z at the exit of the die.

The solution to this problem is yet another ROT type, the VERTEX ROT card. VERTEX ROTs are applied generally to single nodes that are present at the intersection of three sidesets. The VERTEX node is defined by specifying the three sidesets which share it. In this case, we need to write a ROT card for the node where sidesets 20 30 and 60 come together. It's syntax is as follows:

```
ROT = VERTEX 60 20 30 PLANE 20 PLANE 30 PLANEZ 60 NONE
```

Type this into the input deck after the other ROT cards.

It should be clear that this card leaves nothing to chance about what is to be applied at this node. Note also the position assignment is consistent with the sensitivity of the boundary conditions as always.

We probably also need VERTEX ROTs for the other two VERTEX nodes on the out flow plane.

Type the following into the GOMA input deck.

```
ROT = VERTEX 60 20 55 PLANE 20 KINEMATIC 55 PLANEZ 60 NONE
ROT = VERTEX 60 55 30 KINEMATIC 55 PLANE 30 PLANEZ 60 NONE
```

Run GOMA again.

This time GOMA struggles for a bit but then converges smartly to a result. Hurrah!

blot out.exoII, invoke DETOUR, rotate y -45, and plot.

It looks like we have a good solution. No more of that unfortunate mesh folding. But we are not quite done. Look at the plane of the exit and you will see that the mesh is distorting there inappropriately. We still need a few more VERTEX cards to hold those corners in place.

Type the following ROT cards into your file.

```
ROT = VERTEX 70 30 20 PLANE 20 PLANE 30 PLANEZ 70 NONE
ROT = VERTEX 70 30 50 PLANEX 50 PLANE 30 PLANEZ 70 NONE
ROT = VERTEX 70 20 40 PLANE 20 PLANEY 40 PLANEZ 70 NONE
ROT = VERTEX 70 40 50 PLANEX 50 PLANEY 40 PLANEZ 70 NONE
```

By now you should recognize we are specifying the BCs that are applied at the four corners where sideset 70 intersects 30 20, 40 50, 30 50, and 20 40.

Run GOMA again. This will be the last time.

In five iterations, we get good convergence. Look at the solution in blot. All the bad distortion of the mesh is now taken care of and we have a good solution. Finally.

We are done.

Input Deck

Input deck for 3D die swell problem

```
FEM File Specifications
-- ---- --------------
#FEM file                        = swell3D.exoII
FEM file                        = contin.exoII
Output EXODUS II file            = out.exoII
GUESS file                       = contin.dat
SOLN file                        = soln.dat
Write intermediate results       = yes
Anneal Mesh on Output            = yes

General Specifications
------- --------------
Number of processors             = 1
Output Level                     = 0
Debug                            = 0
#Initial Guess                    = zero
Initial Guess                    = read_exoII

Time Integration Specifications
---- ----------- --------------
Time integration                 = steady
delta_t                          = 5.0e-6
Maximum number of time steps     = 200
Maximum time                     = 1
Minimum time step                = 1e-9
Maximum time step                = .008
Time step parameter              = 0.5
Time step error                  = 1e-1 0 1 1 0 0 0 0
Printing Frequency               = 10
Initial Time                     = 0.0




Solver Specifications
------ --------------
Solution Algorithm              = gmres
```

```
Preconditioner                  = dom_decomp
Matrix subdomain solver         = ilut
Matrix Scaling                  = row_sum
Matrix reorder                  = rcm
#Matrix drop tolerance          = 1.e-9
#Matrix Scaling                 = sym_diag
Matrix residual norm type       = r0
#Matrix output type             = 10
Matrix factorization overlap    = none
#Matrix factorization reuse     = calc
Matrix drop tolerance           = 0
Matrix polynomial order         = 3
Size of Krylov subspace         = 300
Orthogonalization               = modified
Maximum Linear Solve Iterations = 300
Number of Newton Iterations     = 10
Newton correction factor        = 1
Normalized Residual Tolerance   = 1e-9
Residual Ratio Tolerance        = 1e-9
Pressure Stabilization          = yes
Pressure Stabilization Scaling  = 0.01

Boundary Condition Specifications
-------- --------- --------------
Number of BC = -1

#inlet boundary
BC = W NS 10 1.0
BC = U NS 10 0.0
BC = V NS 10 0.0

#symmetry surfaces
BC = U NS 20 0.0
BC = V NS 30 0.0

No slip surfaces
BC = U NS 40 0.
BC = V NS 40 0.
BC = W NS 40 0.

BC = U NS 50 0.
BC = V NS 50 0.
BC = W NS 50 0.

#inlet and outlet PSPG boundary conditions
BC = PSPG SS 60
```

```
BC = PSPG SS 10


#free surface
BC = KINEMATIC SS 55 0.0
BC = CAPILLARY SS 55 1.0 0.0 0.0 0.0

#inlet plane
BC = PLANEZ SS 10 0 0 1 0

# x symmetry
BC = PLANE SS 20 1 0 0 0

# y symmetry
BC = PLANE SS 30 0 1 0 0

# y no slip
BC = PLANEY SS 40 0 1 0 -1

# x no slip
BC = PLANEX SS 50 1 0 0 -1

# outlet
BC = PLANEZ SS 60 0 0 1 -6

# exit plane
BC = PLANEZ SS 70 0 0 1 -3

END OF BC

Rotation Specifications =
#
#insert the ROT cards here
#




END OF ROT



Problem Description
------- -----------
Number of Materials = 1
MAT = newt 1
   Coordinate System   = CARTESIAN
```

```
    Element Mapping     = isoparametric
    Mesh Motion         = ARBITRARY
    Number of bulk species = 0
    Number of EQ           = 7

    EQ = momentum1   Q1 U1 Q1   1 1 1 1 1 0
    EQ = momentum2   Q1 U2 Q1   1 1 1 1 1 0
    EQ = momentum3   Q1 U3 Q1   1 1 1 1 1 0
    EQ = continuity  Q1 P  Q1   1         0
    EQ = mesh1       Q1 D1 Q1   0 0 0 1 0 0
    EQ = mesh2       Q1 D2 Q1   0 0 0 1 0 0
    EQ = mesh3       Q1 D3 Q1   0 0 0 1 0 0
                                - - - - - - -
                                d m a b d s p

Post Processing Specifications
---- ---------- --------------
Stream Function = no
Streamwise normal stress= no
Pressure contours = no
Second Invarient of Strain = no
Mesh Dilatation = no
Navier Stokes Residuals = no
Moving Mesh Residuals = no
Mass Diffusion Vectors = no
Mass Fluxlines = no
Energy Conduction Vectors = no
Energy Fluxlines = no
Time Derivatives = no
Mesh Stress Tensor = no


Post Processing Fluxes =
    FLUX = VOLUME_FLUX 60 1 0 area.dat
    FLUX = VOLUME_FLUX 10 1 0 area.dat
END OF FLUX




ROT = MESH SURFACE 20
ROT = MESH SURFACE 30
ROT = MESH SURFACE 55

ROT = MESH EDGE 10 30
ROT = MESH EDGE 10 20
```

```
ROT = MESH EDGE 20  40
ROT = MESH EDGE 30  50
ROT = MESH EDGE 20  30

ROT = MESH EDGE 60  55
ROT = MESH EDGE 60  20
ROT = MESH EDGE 60  30

ROT = MESH EDGE 55  20
ROT = MESH EDGE 55  30

ROT = MESH VERTEX 10 30 20
ROT = MESH VERTEX 10 30 50
ROT = MESH VERTEX 10 20 40

ROT = MESH VERTEX 70 30 20
ROT = MESH VERTEX 70 30 50
ROT = MESH VERTEX 70 20 40
ROT = MESH VERTEX 70 50 40

ROT = MESH VERTEX 20 30 60
ROT = MESH VERTEX 20 55 60
ROT = MESH VERTEX 30 55 60
```