

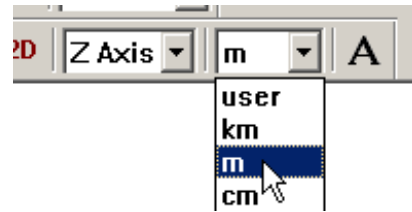
# COULOMB Sample – Power Lines

In this sample you will construct and analyze a power pole supporting wires carrying 3 phases of current at 60 Hz. The goal is to familiarize you with a variety of construction methods and program features.

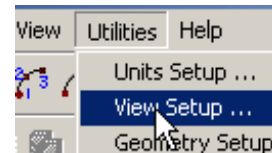
## Step 1 – The pole and cross beams

### 1) Reset the view limits:

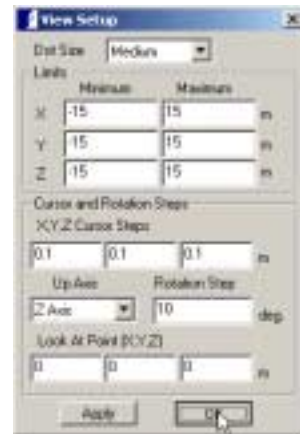
The pole will be approximately 10 m high. For easy viewing from multiple angles and for plotting over the surrounding area it is worth giving the model generous view limits. Begin by ensuring the model units are set as meters:



Next, enter the View Setup Dialog box:

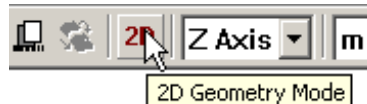


and set all limits to +/- 15 then Click “Ok”:



### 2) Drawing the pole

Select the 2D mode using either the toolbar:



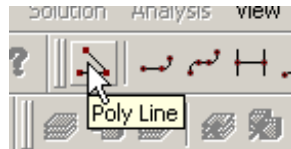
or using the menu:



In the Plane Section Dialog Box select a plane perpendicular to the y-axis, at a y depth of zero. Then click “Ok”.



We will be creating a pole that is 10 m tall and tapered from a diameter of 40 cm at the base to a diameter of 15 cm at the top. Select the Polyline drawing tool:



and enter the coordinates:

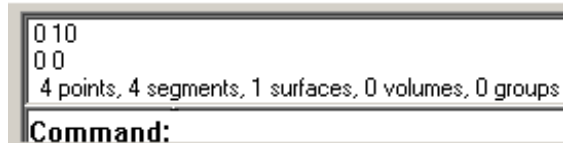
0 0 {Enter}  
 0.20 0 {Enter}  
 0.15 10 {Enter}  
 0 10 {Enter}  
 0 0 {Enter}

**Note:** Any kind of space between the numbers will parse them for Coulomb. For clarity in reading the numbers many people prefer to include commas like this:

0, 0 {Enter}  
 0.20, 0 {Enter}  
 0.15, 10 {Enter}  
 0, 10 {Enter}  
 0, 0 {Enter}

Now you are finished with the **Poly Line** tool press {Enter} again, or press {Esc}, or right-click. Any of these will terminate the command.

Now exit the 2D mode by toggling the switch. It is a good habit to check the geometry summary that appears when you do so. The number of point, lines, and surfaces can help you spot if a mistake is made:



Set the sweep type to “Circular” and Select the **Sweep Surface** tool. Click on the pole cross section to select it, then Right-click to end selection. You will



be prompted for coordinates for the rotation axis. The way this has been drawn the z-axis is the center. So, give 0 0 0 {Enter} as the first point, then 0 0 1 {Enter} as the second

point. Any two points along the axis are sufficient, just keep in mind that the direction of the sweep is determined by the “Right-hand Rule” with the axis pointing in the direction of the first point to the second point. In situations where there is existing geometry on the central axis you can also simply click on two axial points.

After providing the axis you will be prompted for the angle of rotation. We will draw the entire pole, so enter 360 as the angle. Once you end the command you can again check the number of lines, surfaces, and volumes in the message area to make sure it is correct.

***Note:** zoom in close on the bottom surface (e.g. using Alt-LeftMouse to zoom and Shift-LeftMouse to pan). Since this surface was created by sweeping about the center point (which is a part of the surface) you see some gray “isocurves” radiating outwards from the center. Because the center is a part of the surface, as you approach the center where a single point wraps around 360 degrees. This can lead to a dense set of elements in this region and can make calculations difficult near this point. While Coulomb will deal with a variety of singularities such as this one, they do make the analysis more difficult and susceptible to errors.*



*So the safe approach to modeling is to take an extra minute to redefine the top and bottom surfaces. To do so, choose the **Undefine Surface** tool and select the bottom surface. This tool does not delete any points or segments, it simply undefines the existence of surfaces bounded by some segments. After this surface is undefined, display it in 2D selecting the +z plane at a depth of zero. (It will probably be necessary to use the Ctrl-clicking method of selecting a surface hidden behind another surface.) This recognizes the displayed circle as a closed region. When you return the 3D the surface will be defined from the 2D region. However, the radial isocurves will not be there because rather than being a swept line the surface is recognized as a plane with a circular boundary. This is a much more robust surface definition type to work with. Do the same thing with the top surface. Now choose the volume define tool and select any surface for the pole volume. Since there is only one possible volume you won't be presented with volume options, simply a prompt to Right-Click to confirm that you want to define the highlighted surfaces as a volume. You might notice that the original surface you swept to create the pole does not highlight. It doesn't form part of the volume boundary, so it is not part of the volume definition.*




Save the model so far as “pole”.

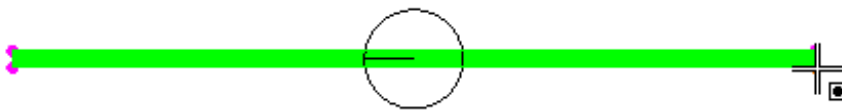
### 3) The top of the pole and the cross-pieces.

The top of the pole presents some challenges and decisions. We need to avoid singularities such as “Overlapping Surfaces” (surfaces which touch without sharing common boundaries), tangent surfaces, intersection volumes/surfaces (vertical surfaces on the cross-piece intersecting the tapered pole volume). For the purposes of this model we will reconcile these problems by assuming that nailing the cross-piece onto the pole will slightly flatten it.

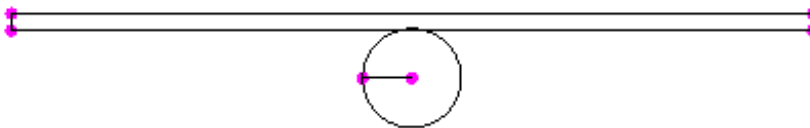
To begin we draw a cross-section of the very top surface, shared by the pole and the cross-pieces. Again, select the 2D plane at a depth of  $+z=10$ . We will draw the top of an 8' long 2x4 board. Change the units to inches. Now select the geometry tool **Rectangle**. Enter the coordinates  $4*12, 1$  and  $-x, -y$ . (Note the ability to do math such as the inch to feet conversion and to reference the previous  $x, y$  coordinates when entering data.)




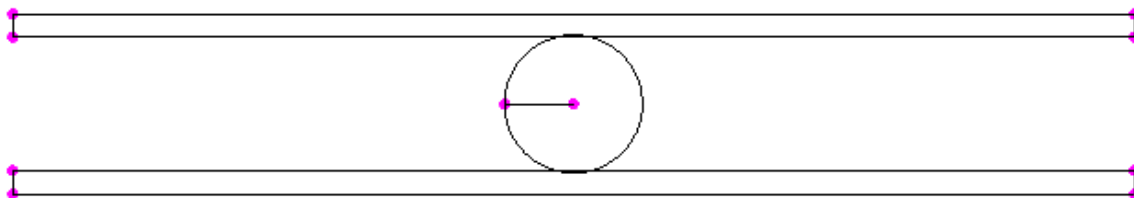
We now want to put the edge of this board almost at the edge of the pole. Since the pole has a radius of 15 cm, change the units to cm. Now, making sure that “Copy Off” is set, choose the displacement tool , select the rectangle, then end selection. When prompted for a reference point choose the lower right corner.



For its destination give  $x, 14.5$ . This keeps the same  $x$  coordinate and moves the geometry such that this point is at 14.5 in the  $y$  direction. This will enable you to get an effective 0.5 cm indent into the pole.

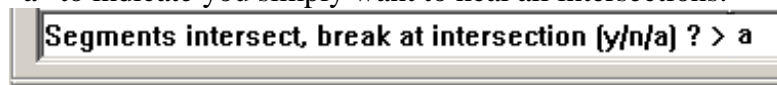


To get a mirror image on the opposite side select “Copy On”, then the mirror tool . For the mirror line select the middle as the points 0 0 and 1 0. (Or, click on the two pink dots inside the circle, which also defines the line along which  $y=0$ .) Your geometry should now look like this:

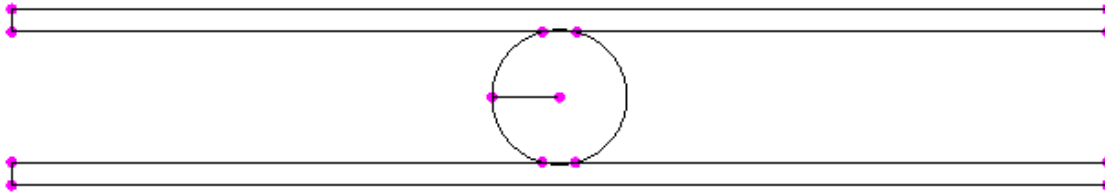


For easy reference, save the model as “pole with cross pieces”.

Choose the menu **Geometry>Healing>IntersectingSegments**. When prompted select “a” to indicate you simply want to heal all intersections:



In a few seconds your geometry should look like this:



Note the circle and rectangles now share common end points for connecting segments.

Return to 3D mode. We have just modified geometry on a 2D plane which is connected to 3D structures (surfaces/volumes) that don't display completely on that plane. Coulomb has logic built in that will attempt to retain and update the definitions of such geometry, but it is possible that some information will be lost. It is a good idea to check the model. You can again check the number of surfaces displayed in the message area, but we are now getting enough surfaces that you may not be sure how many to expect.

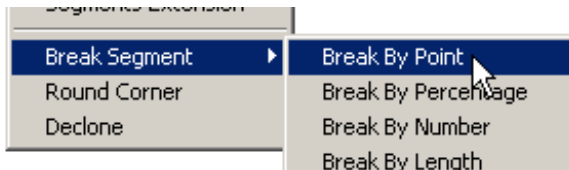
Another good method is to inspect visually with solid viewing. In the present case with the Mar 12<sup>th</sup> build of Coulomb 6.1 we can see that the vertical sides of the cylinder have become undefined.



However, you may also notice that the original vertical surface that was swept by 360 degrees is still there. Since this surface is no longer required you may want to undefine it and delete the radius lines at the top and bottom as well as the line along the z-axis so the model is a bit simpler to look at.



If the cylinder surface has been undefined you won't want to redefine it with the sweep command... that would recreate the overlap problem. It is possible to use one of the **Break Segment** commands to create end points that will make it easy to draw new lines that can create vertical surfaces.



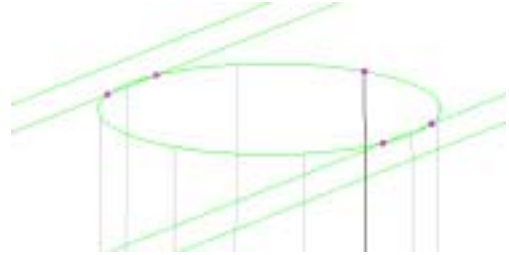
This is work that we would rather avoid, so instead open up the model as you had saved it on the 2D plane ("pole with cross pieces.dbs"). Instead of healing the intersection in 2D simply return to 3D mode. Because of the intersecting geometry problem, the 2 rectangles are not automatically defined as surfaces. So use the **Define Surface** tool to create them one at a time. (Choose the 4 bounding segments for one rectangle then right-click twice, then do the same for the other. Do not use this tool for complex surfaces.)

Once the rectangle surfaces are defined, use the 3D Healing to fix the problem. Choose

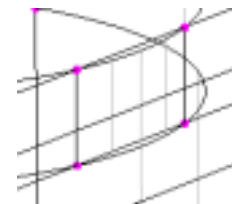


**Geometry>Healing>HealAllOverlappingSurfaces.** This works for planar overlap such as we have at the top of the pole. Answer yes to the prompt to automatically heal. Again the surfaces at the top will be broken to fix the intersection/overlapping problem, but now the cylinder wall doesn't get undefined.

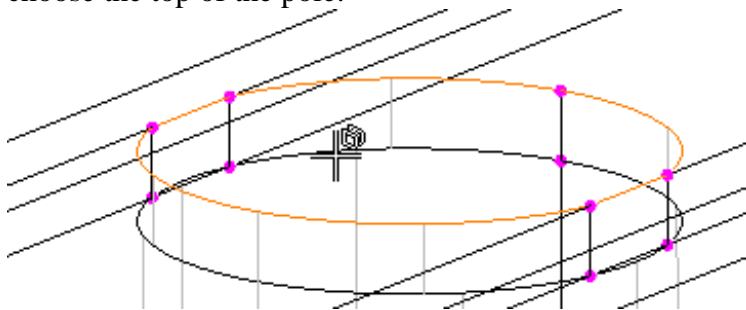
For the final step: switch the units to inches, reset the sweep setting from "Circular" to "Linear", and choose the **Sweep Surface** tool. Select all 5 top surfaces then right-click. For the reference point choose 0 0 0 and for the destination choose 0 0 4. This relative referencing says to sweep by 4 inches in the z direction.



Notice that a very thin volume is created as part of the board. This is a result of the skinny surface where the bottom of the board pushes half a cm into the pole. This is necessary at the bottom, but not at the top. The model can be left as is, but you must remember then to always select these tiny volumes when selecting the board. Also, the small surface will force a higher density of elements on this part of the board for no necessary reason. So for model simplicity and to make the analysis a bit better, we will redefine this. Delete the small arc on each board. This will also undefine the top of the board since the arc was part of the surface boundary.

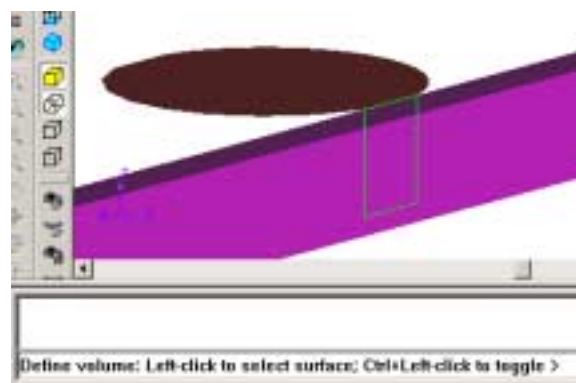


To redefine these surfaces we will again use the 2D plane. However, we will use the "Reference Geometry" option. When prompted, make sure you are on surface selection and choose the top of the pole:



Return to 3D and look at the model in solid mode. It now looks fine, but we need to do some final work on the volumes that have become undefined.

In fact, sweeping the top surface of the pole means that the top of the pole is now its own volume. For simplicity undefine that volume so that the whole pole can be defined as a single volume.



Choose the **Define Volume** tool and select a surface on one of the boards. This time there are multiple possibilities for which set of closed surfaces you may wish to use. Therefore, COULOMB will offer you a choice and give you the option of switching your chosen surface by Ctrl-clicking. Whichever surface is highlighted as green is the one that will be selected when you Left-click. Make sure the vertical rectangular surface shown to the left is the one highlighted before you Left-click. Keep looking at the model and responding to the prompts until the board is defined as a volume, then do the same for the pole and other board. If you are uncertain what surface options are being presented at any step, use Ctrl-Left-click to toggle the options and it should become more obvious. Save the model.

*Note: Most people will find it easiest to follow the prompts when using solid viewing so that only relevant geometry is being shown.*

## Step 2: The Insulators

Choose **File>New** to begin a new model. Change the units to cm and select working on the 2D  $+x=0$  plane. Select the **Rectangle** drawing tool. Enter the points 0, 0 and 0.5, 3.5 to define one rectangle, then 0, 12.5 and 0.5, 15 to define a second rectangle.

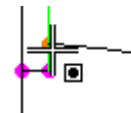
Select the **Free Form Curve** tool and enter the points (0.5, 13), (4, 11), (3, 8), (5, 5) and (0.5, 3).

Select the Poly Line tool and draw a line connecting the dots to finish enclosing the insulator cross-section.

There is one remaining problem indicated by the ends of the curve being blue rather than pink – the segments which the curve is intended to attach to need to be broken in 2 at the intersection point. This could be done with the healing tool, but another method is to select the menu **Modify>Break Segment>Break By Point**.



Select a vertical segment, then click on the connecting blue dot. Do this for both segments and the ends of the curves will be pink, indicating that the segments and curve share common end points.



Return to 3D mode and save the model as “insulator.dbs”.

Perform a circular sweep of the 3 surfaces. Select point that will make the z axis the axis of rotation and sweep through 360 degrees.

*In solid viewing you model might look rather coarsely defined. This is not a geometry problem, it is the surface shading mesh. The amount of computation involved in calculating and displaying the surfaces will impact the response speed of the program for selection, rotation etc. The*

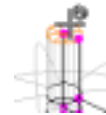


default viewing is rather coarse looking in some models. The real visual test is what the model looks like under the finer shading modes available under the menu **View>Shading**. Selecting the first three of these will require some computation to generate a new mesh. The “System Alternate” option uses built in OpenGL routines to display the model. It is thus very quick to switch in and out of this viewing mode and it will generally produce a very nice looking model for presentation purposes. However, the user interface is normally very sluggish in this mode.

As with the pole model, you may wish to undefine and redefine any surfaces containing a central point with a 360 degree singularity. Save the model.

### Step 3: The connector on top of the insulator.

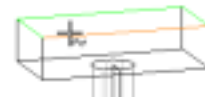
Enter 2D mode using the planar geometry option, choosing the top of the connector.



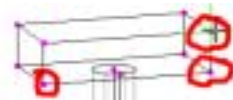
Draw a rectangle from 2, 1 to  $-x, -y$ . Return to 3D mode.

Set the sweep type to “Linear” and select the **Segment Sweep** tool. Sweep the 4 segments bounding the new rectangle upwards in the +z direction by 1 cm.

Define the 4 segments bounding the top as a surface. (This was not defined because we only swept the segments, not the surface. The alternative was to sweep the surface but then get rid of the hole running through the volume.)



Enter a 2D plane using 3 point selection. Create a circle in the center of the rectangle (if you use the 3 points shown this should be 2, 0.5) of radius 0.4 cm.



Use the menu **Modify>Break Segment>Break By Number** to break the circle into 4 pieces. (This is so the cable that will be generated could be used for an induction calculation.)



Return to 3D and do the same thing with the corresponding rectangle on the opposite side.

Define the rectangular box as a volume. Save the model.

### Step 4: Merge and Finish Making the Model

Select the menu **File>Merge** and pick the “pole with cross piece.dbs” model. In the merged model the insulator will be at the bottom of the pole. Switch to

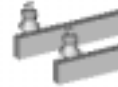




box selection so you can easily select this. Use the **Displace** tool to move it from (0, 0, 0) to (115, 17, 1009). Switch to solid viewing and adjust the view to the top, which should now look as shown to the right.

Save the model as “power lines”.

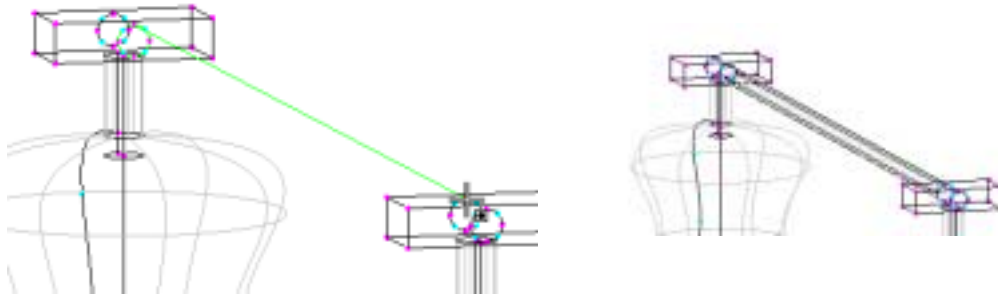
Switch to “Copy On” mode. Use the **Mirror** tool to create a copy of the insulator on the opposite board (across  $y=0$ ).



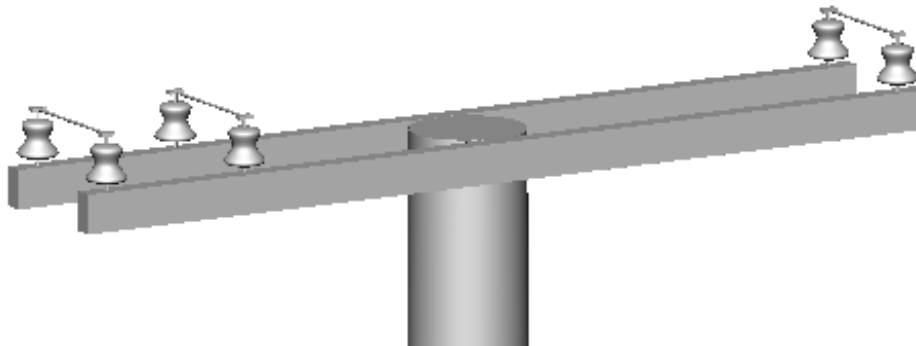
For easier manipulation through zooming and rotation it would be convenient to move the “Look At Point” to this area. Select the tool shown and click on any point in this area. That point will become the new center of rotation.



To create the cable joining the two insulators, use Ctrl-left-mouse to orient the model such that you can readily select one of the circles and sweep it from one of its points to the corresponding point on the opposite insulator.

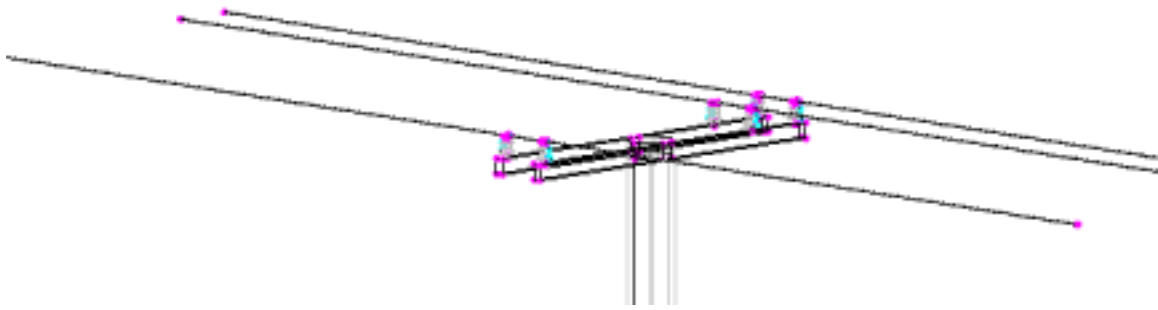


Mirror/copy this whole insulator/cable group across  $x=0$ , then displace/copy it by 40 cm in the  $-x$  direction. The top of the model should now look like this:

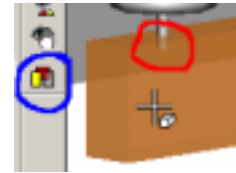


Save the model.

Sweep the circles on the opposite side of the insulator connectors to create the rest of the high voltage wires. Sweep to the planes  $y=\pm 5$  m.



The final geometry editing is to deal with the intersection of the insulator support with the boards. These intersections are demonstrated in the picture (circled in red). The portion inside shows up nicely because of using the transparency tool (blue) to make the board semi-transparent.



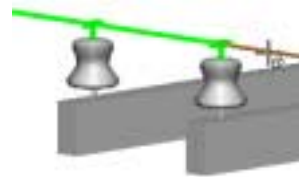
The problem is that the cylindrical surface needs to be divided where it meets the top of the board, and the top of the board needs to contain a hole where this surface passes through. This is accomplished with the intersection tool found on the menu at **Geometry>Intersection>Intersection of Surfaces**. With this tool you can match the posts with the top surfaces. You will have 6 of these intersections to do.

Redefining surfaces will undefine the enclosed volumes. After all surface intersecting is done, use the volume define tool to redefine the posts and boards.

Save the model.

## Step 5: Defining Groups

For easy selection of each high voltage line and connecting conductive material, click the **Define Group** tool (next to **Define Volume**). Using volume selection pick all parts of a high voltage line, Right-click to end selection, then Right-click to define the group. Do the same for the other two high voltage lines.



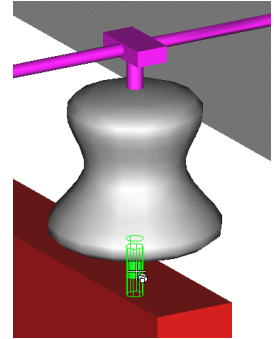
## Step 6: Assign Physical Properties

### Voltages:

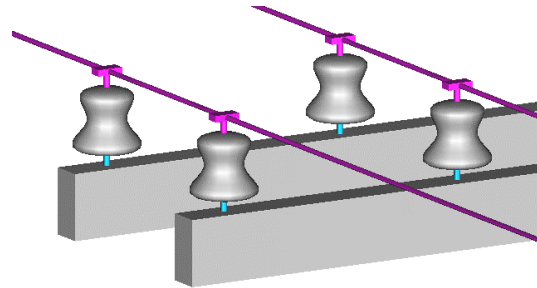
Set the selection type to “Group” and use the menu **Physics>Boundary Conditions>Voltage** to assign one high voltage line as 100,000 V at 0 degrees, then the other two as also 100,000 V, but at phase angles of 120 and 240 degrees. If your

command line prompt doesn't mention the phase, exit the command ("Esc") and go to **Physics>Physics Global Setup** to select Phasor as the "Operation Mode".

The posts attaching the insulator to the boards will be made of a metal, hence they will be at a constant voltage. However, that voltage is unknown. Therefore, these should be assigned a "floating" boundary condition. Go to **Physics>BoundaryConditions>Floating** and using "Volume" as the selection type choose each of these one at a time and make them floating. Each of these may be at a different voltage, so they should be numbered 1,2,3,4,5,6 when prompted for the conductor number. The order of the numbering isn't important. The numbering is a means of associating conductors so that they float to the same potential. However, the numbering must be sequential without missing integers. For example, the solver will object if you assign 1,2,3,5,6,7 since there is a number 5 but no number 4 in the model.

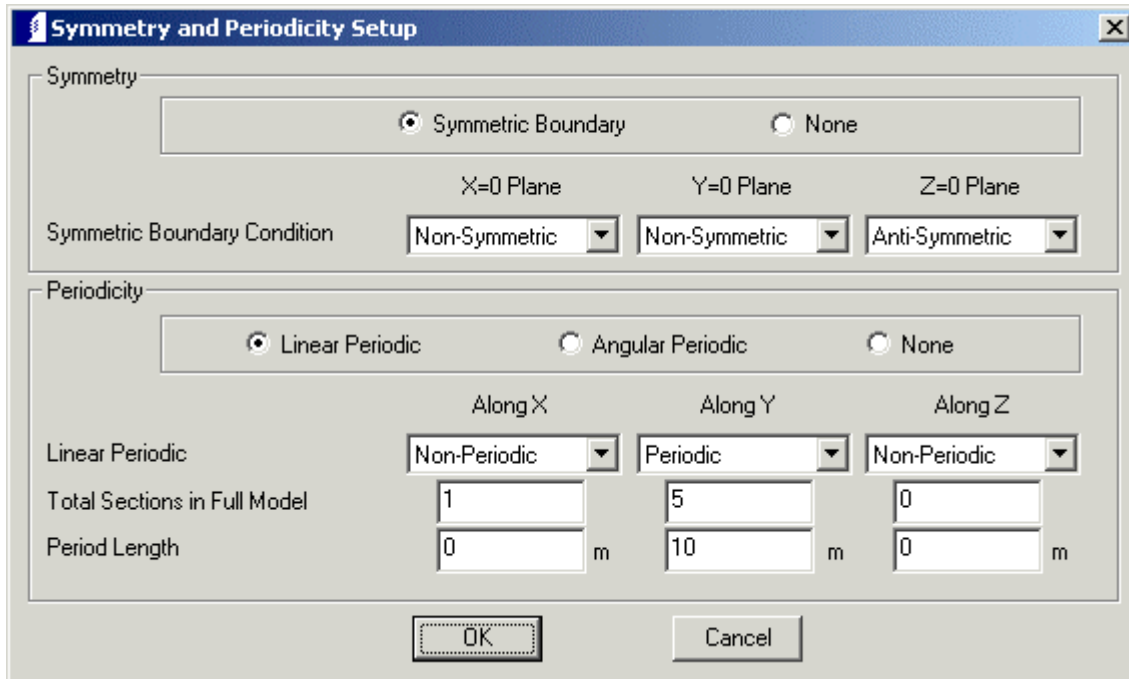


Notice that whenever the mouse goes down the Physics menu to **BoundaryConditions** the floating conductor surfaces appear blue and the voltage surfaces appear violet. This is a feature of the Physics menu – as you scroll down it you will see the model change color to indicate the properties associated with each item. The exception is the **MaterialTable** entry. Material colors don't display until you open up the material table.



## Symmetry and Periodicity Setup

Click on the menu item Physics>Symmetry and Periodicity Setup. This will open up the following dialog box:



In a real model of this type you would likely make the pole come out of a large rectangular volume assigned a material to represent the ground. For the present purposes we will consider the ground to act as a very good conductor. Because of this the model is considered to be antisymmetric in the z direction. (A perfect planar conductor problem is equivalent to having the exact same problem mirrored across the conductor, but with the signs of all sources reversed.)

Using linear periodicity has enabled us to model a row of identical poles with the connecting wires by drawing a single central pole and wires. With linear periodicity in COULOMB you draw a single central unit and apply the properties to it. You then specify how many total sections you wish to have. An equal number of images are placed on each side of the central unit at the spacing given by the “Period Length”. Hence, the total number of sections must always be an odd number.

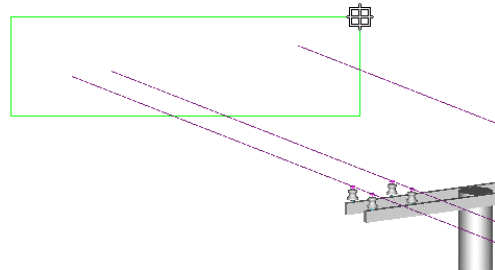
Using symmetry and periodicity (or angular periodicity) the solution time is determined by the central unit that is drawn, not the number of images. This means that basic solver time can be reduced dramatically. In the present case, to draw the negative pole under the “ground” plus 4 images of the central part would make the model 10 times larger. The result would be that the normal BEM matrix to be solved would be  $10 \times 10 = 100$  times larger. So, we are solving the equivalent problem approximately 100 times faster by using symmetry and periodicity with the normal BEM method.

Although the symmetric and periodic images don’t affect the time taken to solve a model, they do affect the time taken to perform computations such a force, torque, field plots, etc where contributions must be added from the entire model – the part drawn and its images.

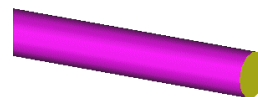
This is why only 5 sections were chosen in the dialog box above. For this model that should be sufficient to give the same answer around the center as having a much larger number, but a much larger number make plotting the results much more time consuming.

Linear periodicity puts an identical solution on the drawn model and all its images. Hence fringing effect at the ends of the model won't be quite correct. The extent problem this creates will depend on the model. In a case like this, assuming our interest is in the behavior around a "typical" pole, linear periodicity works very well.

At the ends of the wires and at the bottom of the pole there exist surfaces which are really internal. They must be drawn so that volumes can be assigned, but they do not really exist. Because of this, they shouldn't be assigned elements. To let COULOMB know not to assign elements to these surfaces you should use the Boundary Conditions for Symmetry and Linear Periodicity (see the submenu **Physics>Boundary Conditions**). To save a lot of rotating, panning and zooming to display these surfaces in a way that is convenient for selection by surface, set the selection type to "Box". Since these surfaces are at the ends of long cylindrical surfaces you can easily draw selection boxes which include these surfaces without enclosing any others.

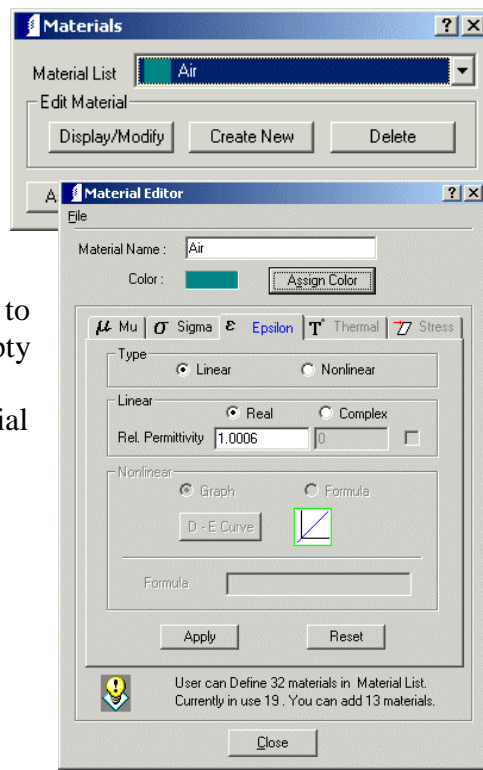


After setting the symmetry and periodicity Boundary Conditions you should zoom in on one of these surfaces and note that these conditions are indicated by yellow coloring.



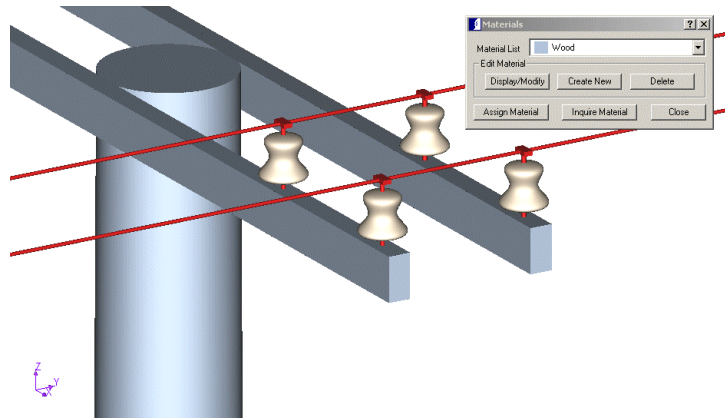
## Materials

Choose Physics>Materials Table to open up the material table. This table provides a list of up to 32 materials which can be assigned to volumes in the model. For demonstration you can select any dielectric materials you wish to assign to the wood and insulators. If you want to work with specific wood or ceramic materials for these you can use Display/Modify to rename and change to properties of a couple of entries, or use Create New to make two new ones (there is typically about 10 empty spaces in the table). In either case you will do the material naming and properties editing in the "Material Editor" dialog box that pops up.



In Coulomb the magnetic permeability  $\mu$  is irrelevant, hence it can be left as 1. By default you are in “Permittivity Only” mode, so any conductivities you enter under Sigma will have no effect. You can change this mode under **Physics>Physics Global Setup**, however, this is usually not a good idea. You should only use “Conductivity+Permittivity” when you are operating at a frequency where the real and imaginary parts of the dielectric constant of a material are comparable. Otherwise you are adding a lot of unknowns and slowing down the computations for no good reason. In any other case everything can either be assigned a voltage or floating condition, or considered a perfect insulator.

You may wish to assign copper to the wires etc. for display purposes. Since these are assigned floating and voltage conditions, the material inside is irrelevant to the computations. To get a good image for presentation you may also change the coarseness of the viewing mesh. By default Coulomb uses a coarse mesh to display the models.



Under the menu **View>Shading** you will find 3 COULOMB generated display modes (Coarse, Medium, Fine). When you switch between these modes COULOMB recalculates a mesh of shaded triangles. The finer the mesh the better the display will look. However, the mesh requires memory and needs to be recalculated when changes are made to the geometry. Hence, fine meshing can slow down the responsiveness of COULOMB for a complex model. The other shading option – “System Alternate” – turns control of the display over to an OpenGL viewing mode available on your system. This is usually a very nice looking display of the model. Furthermore switching between this display and the previous COULOMB generated viewing mesh does not require re-computing the mesh, so it is faster than switching between COULOMB modes.

However, COULOMB gives up a lot of control to use this mode with 2 side-effects:

- 1) any translucency assigned to surfaces doesn't work
- 2) the whole interface tends to become very sluggish in this mode, sometimes taking several minutes for such task as highlighting a selection in a complex model

Due to these disadvantages, this mode is usually only used for the purpose of getting a good look at the model or taking a screenshot, but one of the other modes is selected for working with the model.

## Step 7: Setting up and Running the Solver

We will solve this model using the default solver settings, but for your information consider the dialog box opened up from **Solution > Solver Setup**

**Method of solution:** As of version 6.1 the only solution method in Coulomb is Boundary Element.

**Matrix Solver Type:** This will determine whether the matrix solution uses a direct or an iterative process. The Auto setting chooses whichever is expected to solve faster.

**Iterative Accuracy:** Can only be set when the Solver Type is set to Iterative. This is a measure of how well the solution satisfies the matrix equation. To get a more accurate equation solution you can decrease this number, but accuracy problems are usually due to not enough or poorly distributed elements.

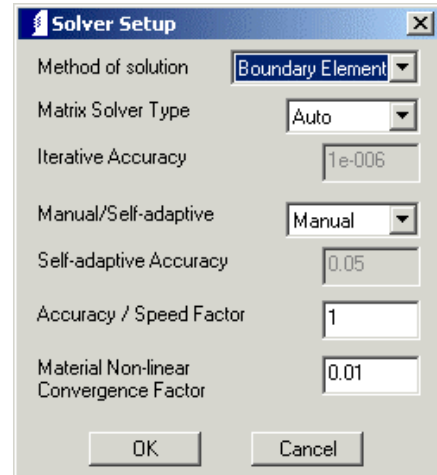
**Manual/Self-adaptive:** With the manual solver you setup elements and then solve the model. With the self-adaptive solver the elements are setup automatically, the problem is solved, then the boundary condition on each element is assessed. An overall error number is computed and compared with the **Self-adaptive Accuracy** number. If the error is larger than this number the elements with the largest error are subdivided and the problem is re-solved. This repeats until the overall error is low enough. This has the benefit of giving the most strategic distribution elements. It has a disadvantage that the multiple solutions can become time consuming and especially so with complex models (such as this one). So, it is worth experimenting with this solver, but as of version 6.1 our default in 3D is still the Manual solver.

**Accuracy/Speed Factor:** Coulomb – with Boundary Elements – solves models by assigning charge which satisfies all boundary conditions to the nodes of the elements. To make computations from this solution (e.g. electric field values, voltages) the charge is distributed over the element and the contribution is integrated. Far from an element the charge can be considered a point charge, nearer the element various integration methods are used depending on the element size and the distance. Increasing the accuracy/speed factor will cause Coulomb to choose a more accurate integration at any given distance. A factor of 2 means “use the integration that would be used at half this distance”. More accurate integrations will mean the model takes longer to solve. This number is normally increased to values up to 10 in model which are having difficulty converging.

**Material Nonlinear Convergence Factor:** A measure of how well the D-E relationship is following a nonlinear curve. In version 6.1 this is irrelevant in Coulomb.

## Assigning Elements

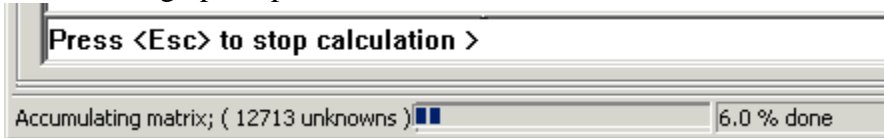
Click on the menu **Solution > Triangular Elements > Automatic All**. You will be prompted for the number of elements to assign to the model. Request 1 to get the



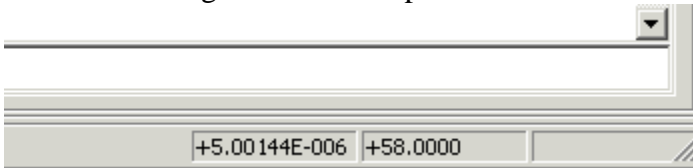
minimum number of elements that COULOMB can use to solve. This should produce approximately 12,000-13,000 elements.

## Solving

After assigning elements click on the menu **Solution > Solve**. The solution proceeds in 2 distinct steps. During the first step you see a status indicator for accumulating the matrix. The is setting up the problem that needs to be solved.



In the second step the matrix is actually solved using either the direct or the iterative solver. The direct solver is normally used for small problems, so with a problem where you wait a long time for an answer you will see this information in the right corner of the status bar during the second step:



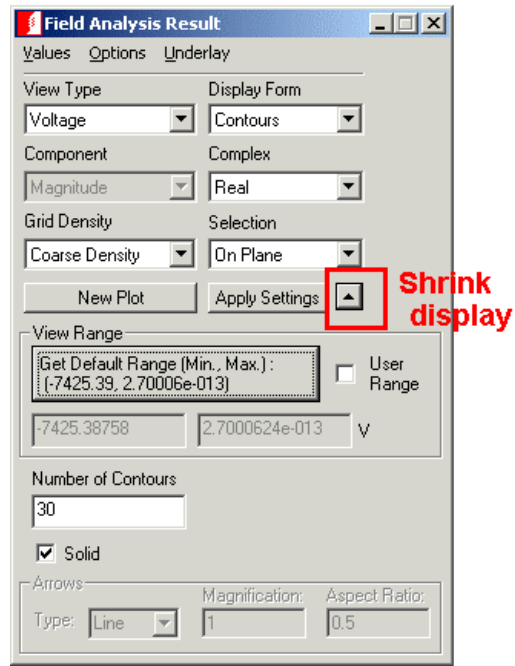
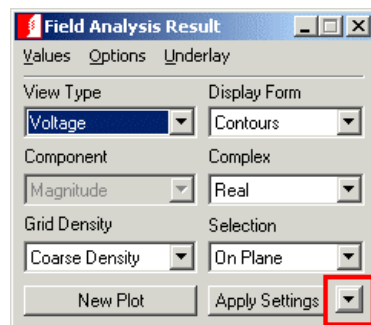
The first number is the iterative accuracy obtained so far – which is close to the target value of 1E-6. The second number shows that 58 iterations have been performed so far.

On a 2.4 GHz computer this problem took approximately 1 hour to solve.

## Displaying Results

When the solver finishes Coulomb pop up the Field Analysis Results dialog box. (When it is closed you can always open this box for a solved model from the menu **Analysis > Field Results**.) This table is used for obtaining numbers associated with points. This may involve plotting voltage, E, charge density etc., or querying the values of these at specific locations. In the latter case, the Values menu has options for obtaining the values at individual points either by typing them in on the **Command** line or by reading them from a file.

The initial view of the Field Analysis Results dialog box only shows some basic plotting





options. This can be expanded by clicking the down arrow in the lower right hand corner.

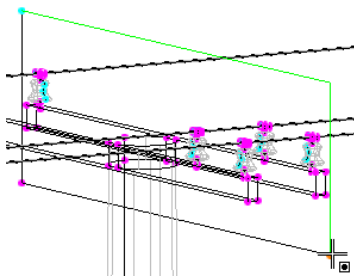
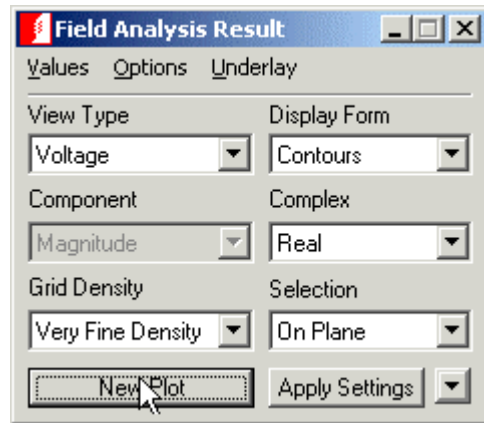
The other items on the **Analysis** menu are related to bulk values associated with parts of the model, rather than points in space (force, torque etc.)

To create plots you must specify end points of the a line, corners of a plane or box, or select given geometry to plot on. It is often convenient to create geometry without any physical properties applied purely for the purpose of plotting. Select the segment drawing tool and enter the following points:

```
>1.3 0.17 9.75
>-x y z
>x y 10.5
```

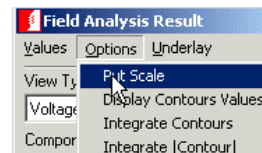
This produces two lines approximately centered about one of the boards. Next setup the plot as shown to the right and click on New Plot.

The Field Analysis Results box will disappear while you define the plot. Select the 3 dots on the corners of the 2 lines you just drew as the border of the plotting plane.

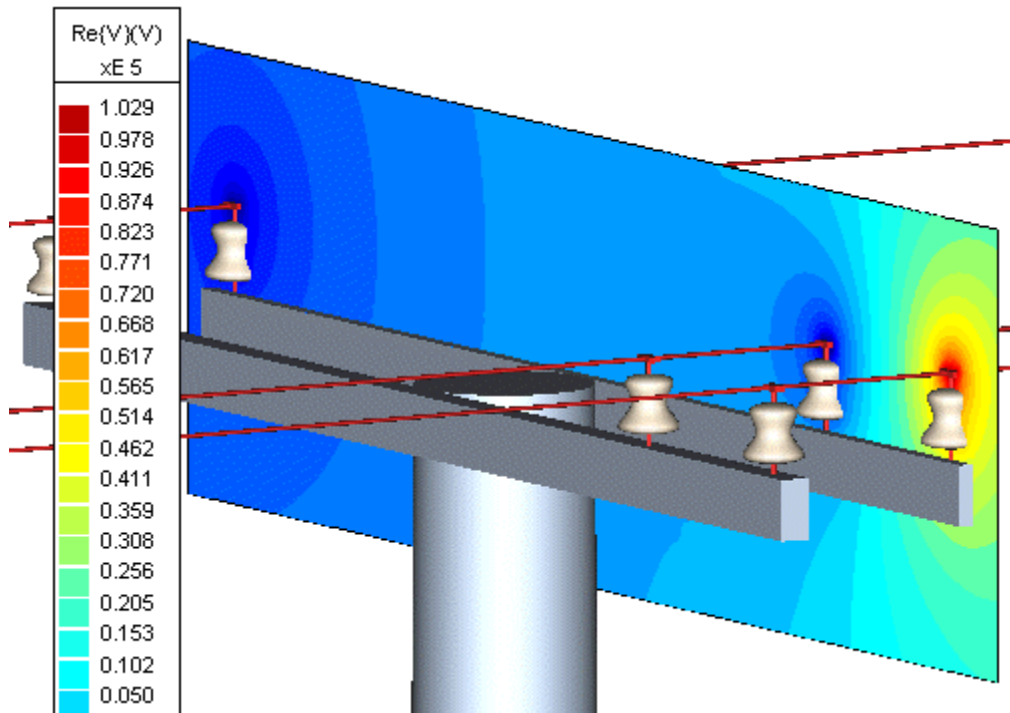


COULOMB will now create a “Very Fine Density” grid on that plane and calculate Voltage at each point on the grid.

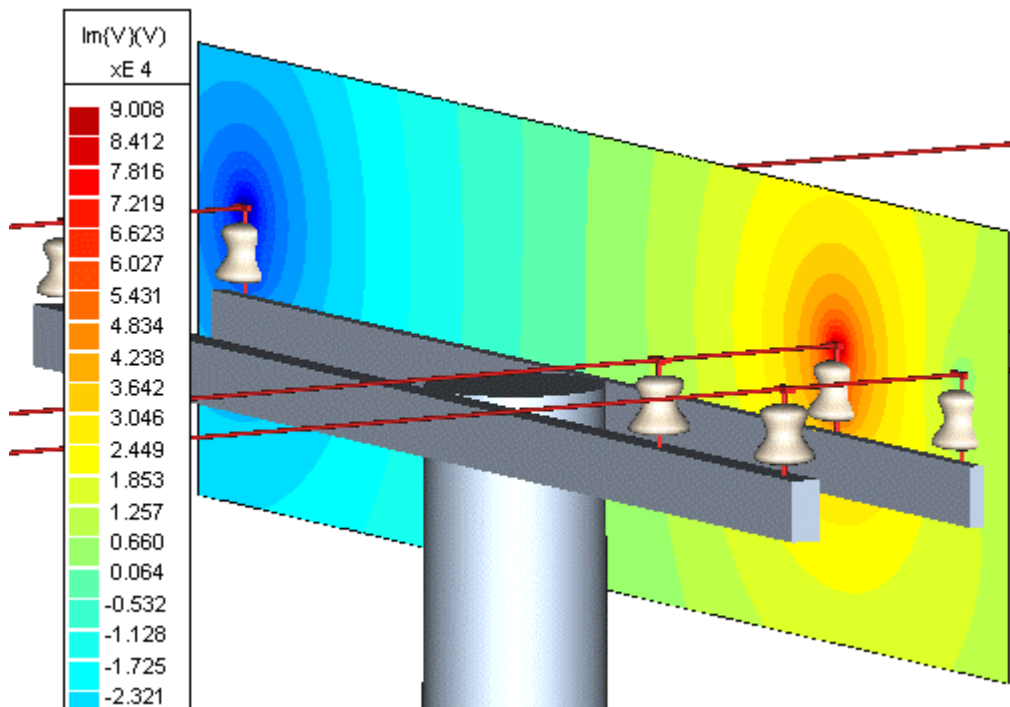
This will likely take a few minutes to complete. When it is done a solid contour plot will appear using values interpolated between the grid points. You can then put a



reference scale in place from the menu shown. Use the mouse to locate the scale at a convenient location. After that, in solid viewing mode your model should look somewhat like this:

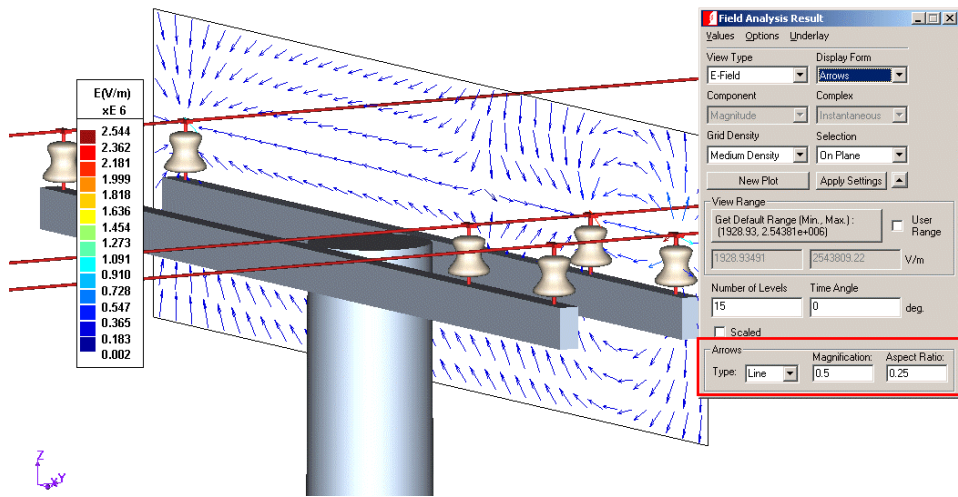
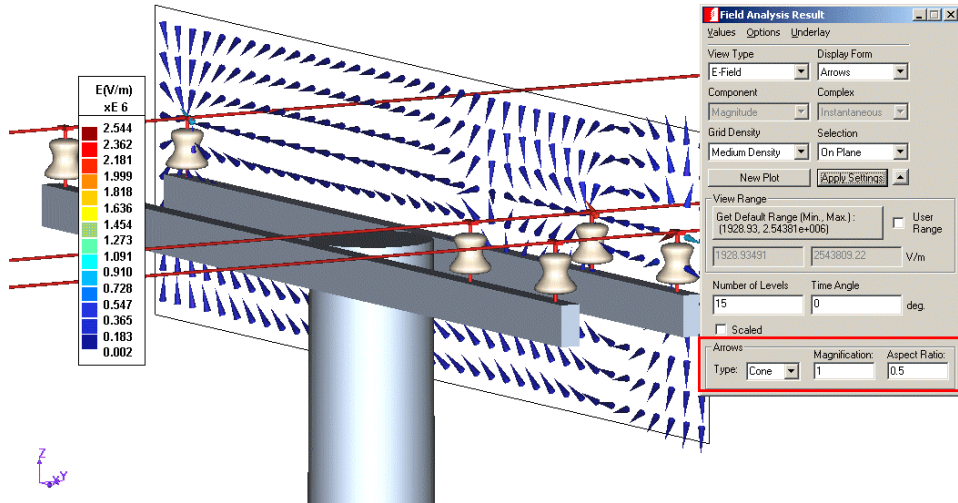


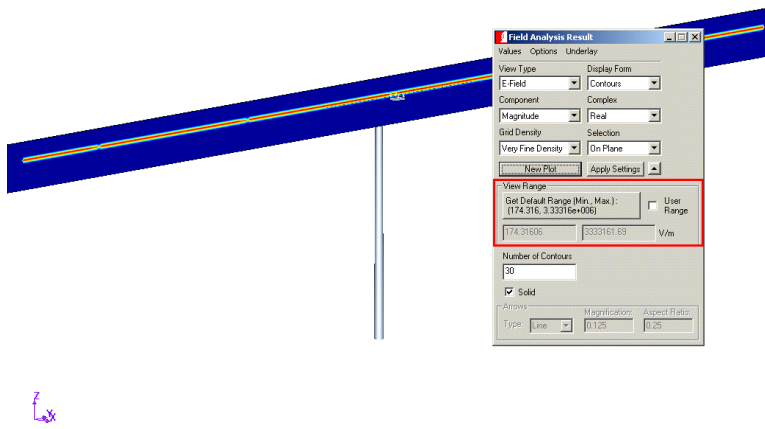
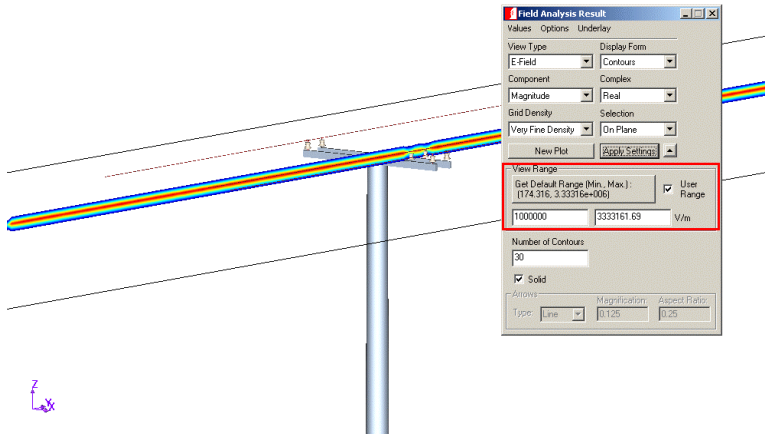
Changing the plot type – for example the density of points or switch to an E field component – will require recomputing. However, if you are switching between line and solid contours, changing the component plotted, etc. you can just make the change then press **Apply Settings**. For example, change the **Complex** setting to “Imaginary” and then press **Apply Settings**. The display should change to this instantly:



You can choose what physical aspect of the model determines the coloring with the **Underlay** menu.

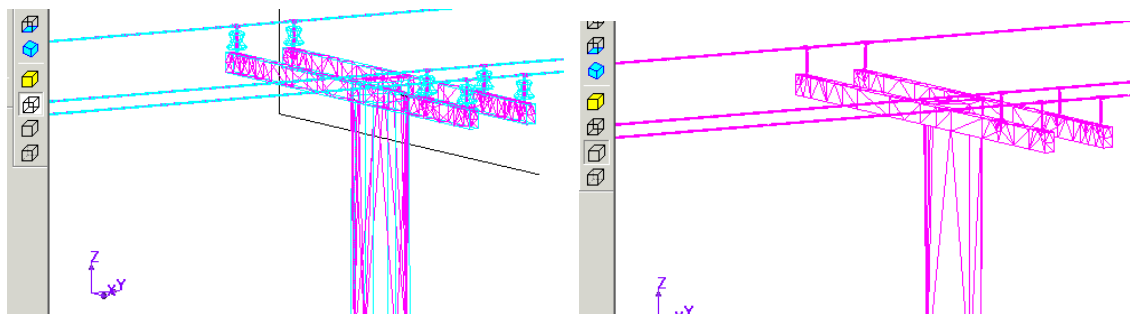
You should experiment with producing plots of various types. Here are some samples:



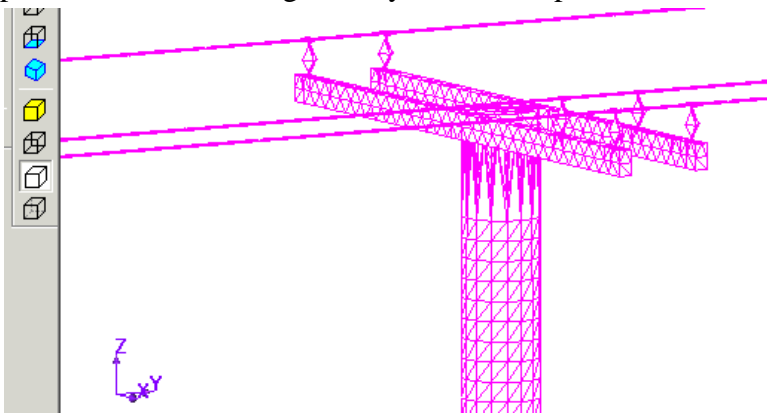


## More Solution Options

Consider the element distribution seen by requesting “1” element.



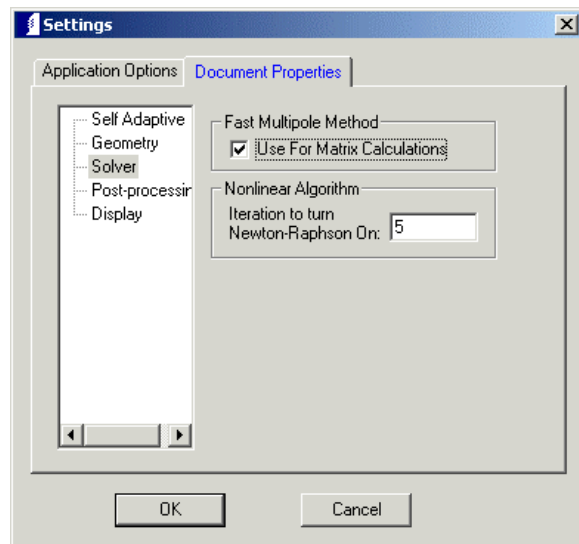
By requesting “1” element you are asking Coulomb to produce the minimum number of elements it considers to be acceptable based on some geometry rules for aspect ratio of the sides and matching nodes at edges. The wires get a very good number of elements because they have 4 sides wrapping around the outside. If you request a higher number, as seen to the right, you will get a better mapping of elements onto the pole and insulators. The extra elements will have little impact on the elements on the wires until you request an extremely large number.



When looking at elements keep in mind that the triangle element represents a section of the surface, but is only displayed by the boundaries drawn with straight edges. This will make it appear that the elements in the first case form a thin cylinder, and a pyramid type shape in the second case. In both cases the elements follow the curved surface, however, the more elements there are the shorter the (straight) sides, hence the closer they can be seen to be following the surface.

More elements will provide a more accurate solution, however, it will also take more time. With the Boundary Element Method, for large problems (more than a few minutes to solve) the time taken to get the solution is approximately proportional to the square of the number of elements on a given model. The time taken to do post-processing such as producing contour plots is approximately proportional to the number of elements.

To reduce the time taken to solve large problems, in version 6.2 and later a Fast Multipole Method (FMM) can be invoked in the Solver and/or Post-processing. FMM makes some approximations that reduce the time taken to make calculations. Theoretically this method will make the solver time grow as  $N \cdot \ln(N)$  instead of  $N^2$ . Experience with Coulomb so far shows that significant time savings can be accomplished for greater than about 5,000 elements with a resulting error of approximately 2%.



Besides the number of elements and FMM vs BEM, you can also experiment with the settings under **Solution>Solver Setup**. When solving the matrix equation that produces the charge distribution on the elements you

have a choice of a Direct solver which solves exactly, or an Iterative solver which adjusts trial solutions until the equation is satisfied within some given accuracy criteria. “Direct” is superior for results but becomes very slow for large problems. On the “Auto” setting Coulomb will switch to the iterative solver at approximately 1000 elements. Near the change this might mean more elements produce a less accurate answer. To get better accuracy you can change the iterative accuracy to a smaller number to force more iterations, or if the problem is not too large switch to the Direct solver.

COULOMB also has a Self-Adapt option for producing elements. This method automatically assigns minimal elements, solves, computes how well the boundary condition is satisfied on each element, subdivides those with the largest error, then solves again. This repeats until the average error is smaller than some specified quantity. The self-adapt procedure is relatively new and may require some experience to work well on a given problem. The biggest danger is that the elements will subdivide so much that the problem size become impractical. However, it will theoretically produce a better answer by producing an optimized element distribution. For now, “Manual” is the default.

Computations in Coulomb are based on integrating the effect of the charge on each element. The charge is distributed over the elements surface. To perform this integration Coulomb makes a choice of how much to divide the element surface based on how far away the effect is being computed compared to the element size. In the “Solver Setup” you can affect this choice with the “Accuracy/Speed” factor. For example, a value of 2 means “perform this computation as if the distance was twice as small”. This will lead to more accurate computations, but at the expense of taking more time. In most cases this is a last resort after using the options above, but it is known to be quite helpful in a variety of problems where elements are very close (thin films, small gaps).

## Circuit Parameters

Besides solving for fields, voltages, force, torque, streamlines, etc. COULOMB can also compute Capacitance and free space Inductance. This is set under **Physics>Physics Global Setup** where you select the Solver Type mode as “Fields”, “Capacitance”, or “Inductance”. In the latter two modes you assign conductor numbers rather than voltages. Coulomb will then compute the matrix of partial inductance or capacitance based on those numbers.

Choose Capacitance mode and the Physics menu now includes “Surface Conductor”. Select one group of cables/connectors and assign it as conductor number 1, another group as conductor number 2, and the third group as conductor number 3. Now when you solve, instead of computing fields etc. according to your specified voltages, the voltages on these groups will be varied in order to determine

$$\begin{vmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{vmatrix}$$

The results will display in the message area and can be queried from the Analysis menu. Results can also be saved to a file using the menu **File>SaveDataToFile**.

Next choose Inductance mode and undefine the Surface Conductors (these can be used for inductance from a thin material, but here we want inductance of a cable). Assign the 3 sets of cables as Volume Conductor number 1, 2 and 3. Note that we didn't make the effort to draw the connectors such that we could use them in the inductance calculation. (They are so small compared to the cables that they will have very little effect, so this is not a bad approximation.) Besides assigning the volume conductor numbers, you also need to assign volume current to each volume. The magnitude and phase you assign are unimportant. For simplicity assign 1 A at 0 degrees. The important thing is getting the direction of current flow. Coulomb can assign current flow to 6-sided volumes (this is why the circle swept to create the cable was broken into 4 segments), an entry surface, and exit surface, and 4 surfaces that guide the flow. When you are prompted to select a surface, choose the surface where current flows in to the volume.

With inductance the symmetry and periodicity aspects of a problem might change. The source now is current rather than voltage. In this model the periodicity is the same, but the ground has no effect on the inductance, so remove the vertical antisymmetry and remove the symmetry condition at the bottom of the pole.

Reassign 2D elements and you will have 0 assigned. This mode is much faster to solve because all the dielectric materials have no effect. With the volume current flow you do need to assign 3D brick elements. These will be assigned to any volume assigned volume current. Coulomb cannot allow for any magnetic material properties, so the solution will be a free-space inductance matrix:

$$\begin{vmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{vmatrix}$$