Wavenology BHA Manual with Tutorial Examples

Wave Computation Technologies, Inc.

Dec. 2025

Outline

- Source definition: elastic wave sources (2)
  - Curve type sources
    - Line source
    - Arc source
      - The difference between the arc source and the ring source
  - Face type sources
    - Rectangle source
    - Circle source
    - Circular cylinder source
  - Volume type sources
    - Box source
    - Sphere source

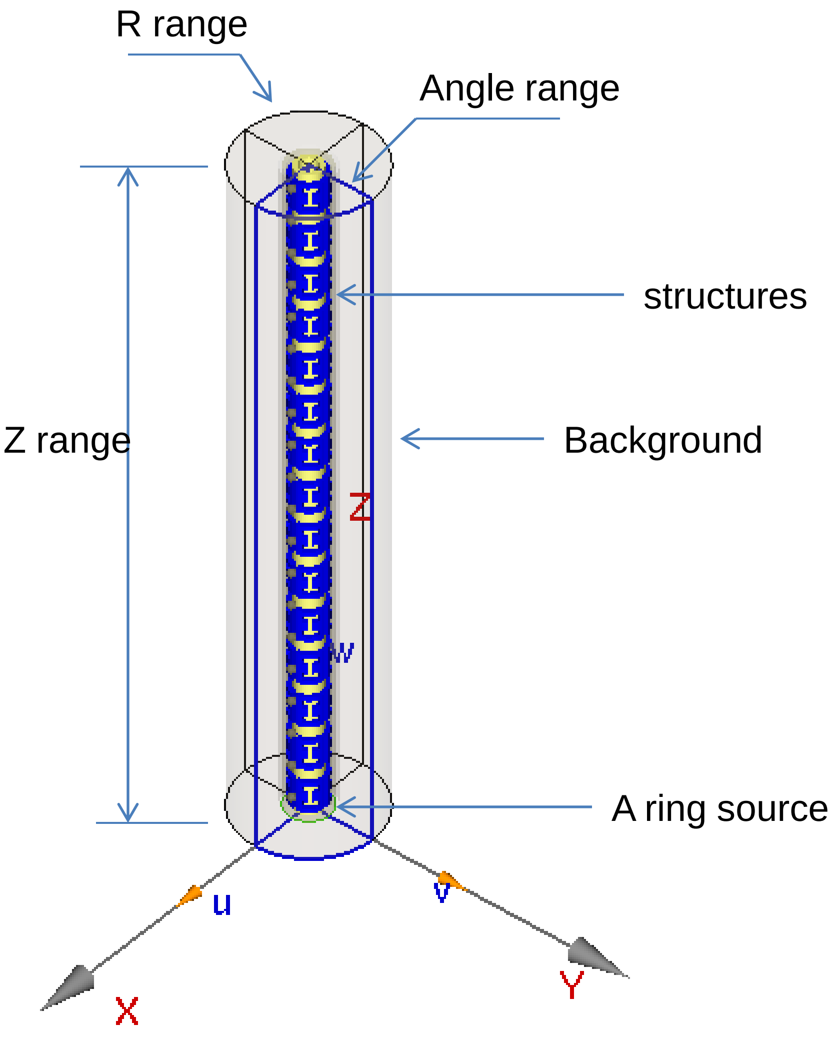
- Source definition: elastic wave sources (3)
  - The amplitude distribution for the ***curve, face, volume*** types source
    - The constant distribution
    - The linear distribution
    - The Triangle distribution
    - The Gauss distribution
    - The Cosine distribution
  - Individual excitation pulse for each source
  - Reduce project size by dealing source type with B.C. in ϕ
  - Create source by data file

- Appendix 227
  - I : mesh generation and displaying
    - The exported mesh data format
  - II: the possible result difference caused by mesh difference
  - III: capture the hole of solids in the mesh
  - IV : the PML setting
    - Special PML setting for the background with small Q factor
  - V : geometry clash test
  - VI: tips to increase the simulation Δt: R_{min} setting & background material
  - VII: Q & A for licensing problems

- Tutorial cases 256
  - 1. A steel pipe in borehole
  - 2. An insulator
  - 3. Import 3D Solids from a SAT file
  - 4. Generating Cylindrical Mesh and export mesh data
  - 5. Setup a simulation project with imported structures from SAT file
  - 6. Simplify existing project to a new model

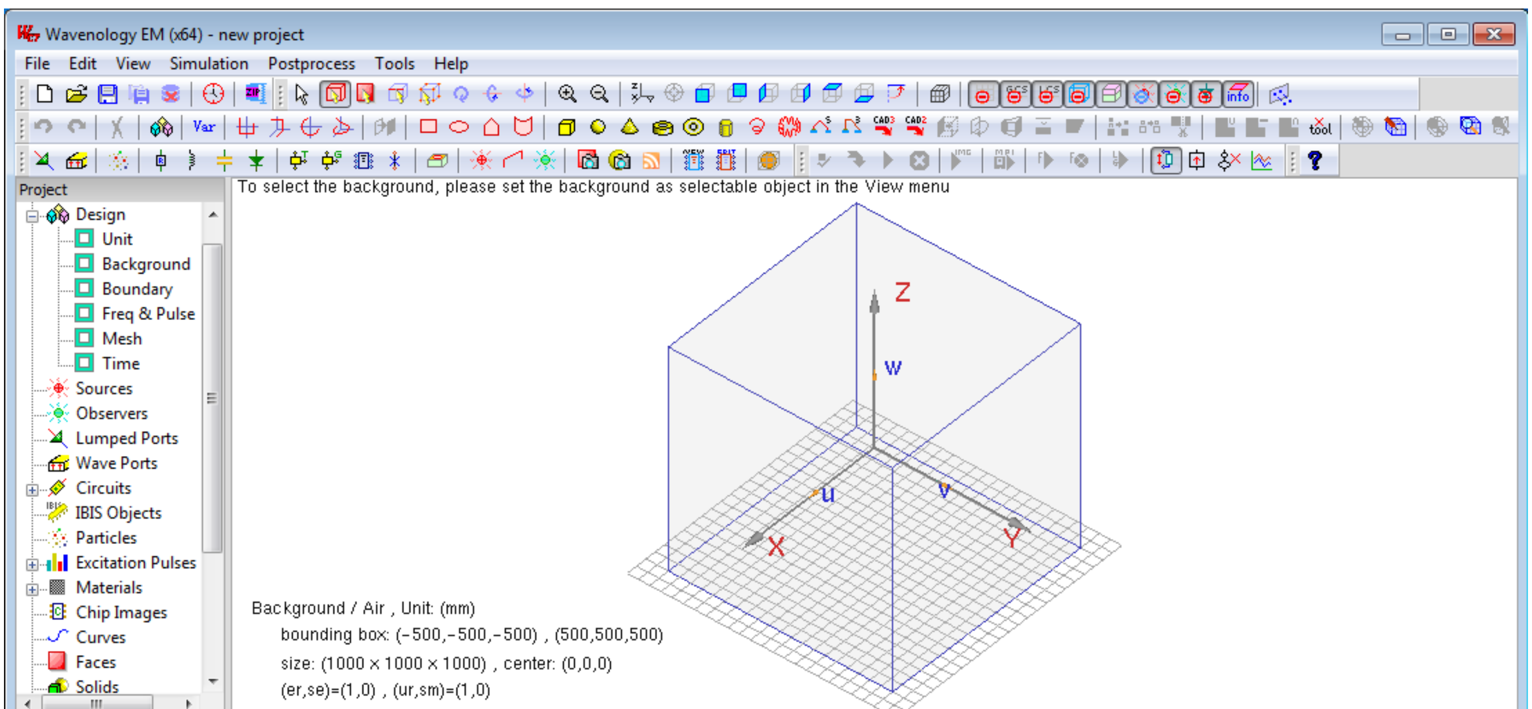
Start a BHA Project & GUI Information

A typical BHA project looks like the figure



Set up a BHA Project

- WCT EM & EL GUI is a general GUI for multiple solvers, including the Cartesian EM, the Cartesian EL, the Cylindrical EL (BHA), etc.
    1) For a new user, the default GUI after installation is for the Cartesian EM solver, as following



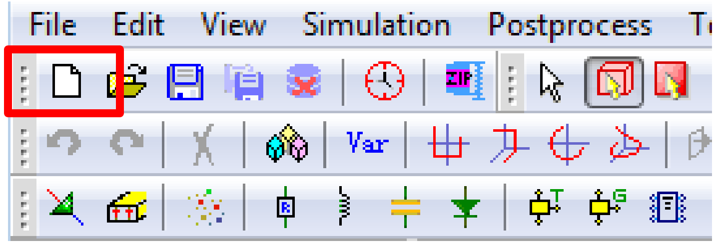
2) If the GUI has been used for some kind of projects, the default project type will be the same as previous project when user open the GUI in the next time
   - for example, if user closes the GUI with a BHA project, the default project type will be recorded as a BHA project automatically.

In order to switch from the EM solver to the BHA solver, there are 3 ways

Method 1) Load an existing WCT BHA project.

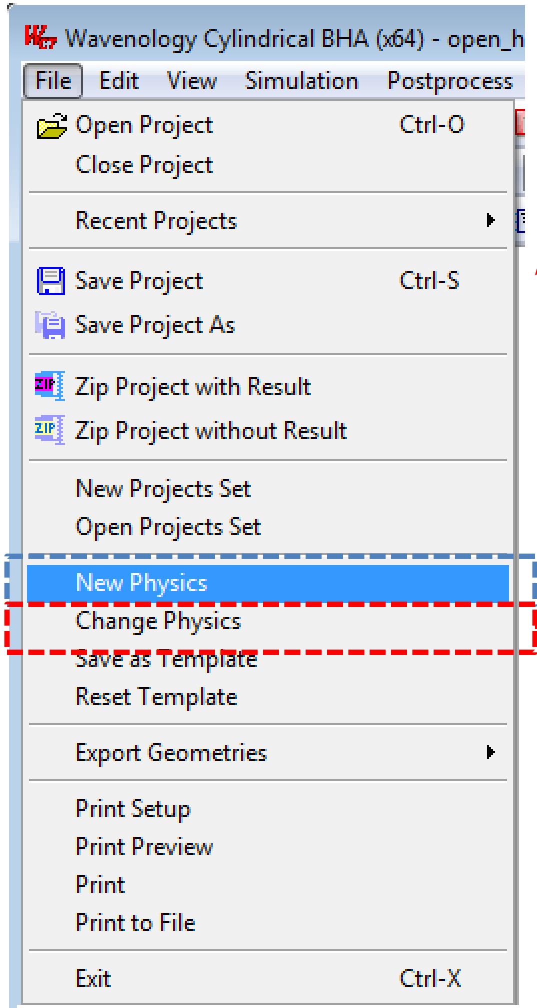
- for example, load a BHA project from the BHA demo package

Method 2) If a BHA project is editing, “New” a project will create a new BHA project



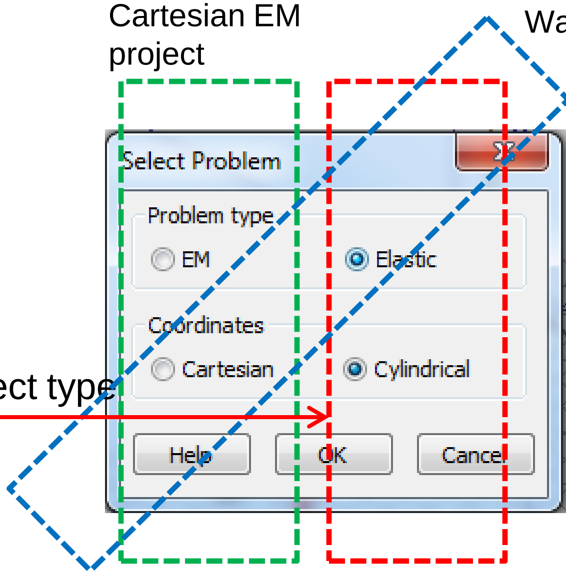
Method 3) Switch to a new BHA project from menu

There are two menu items to define a project as a Cylindrical Elastic Wave project, a Cartesian Elastic Wave project, or a Cartesian EM project.



Create a new project

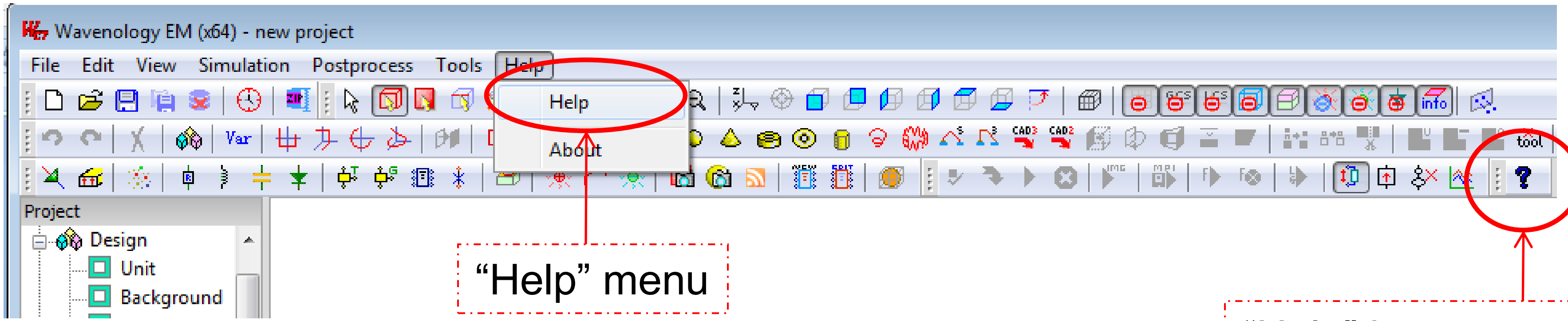
Change an existing project type



Cylindrical Elastic Wave project (BHA)

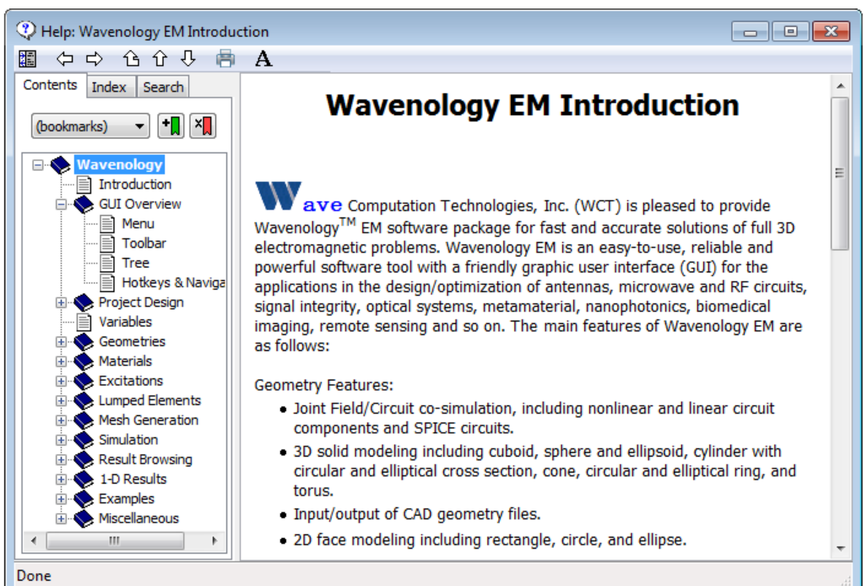
GUI Layout

- For the general GUI information, please refer to the embedded manual in Wavenology GUI (“Help”). The elastic wave solver will share most GUI system of Wavenology EM Package.



“Help” menu

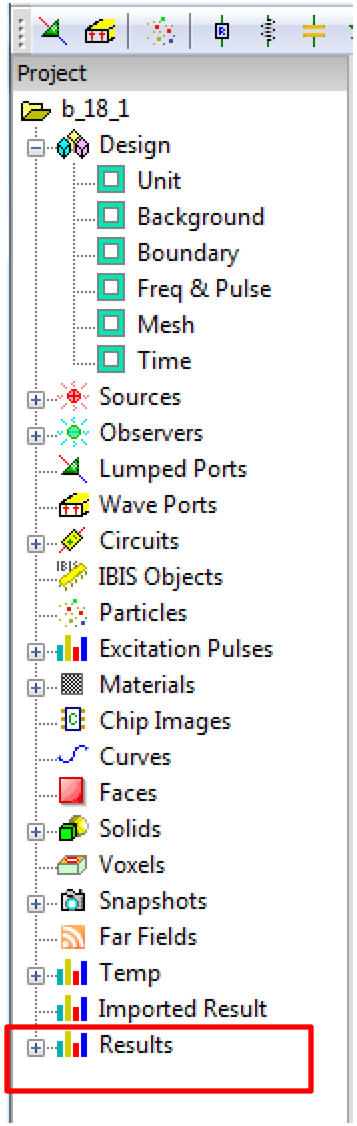
“Help” button



Wavenology EM Package manual

the Project Tree

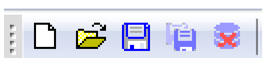
All project contents can be viewed through the project tree



Project setup

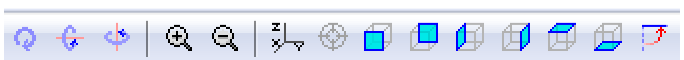
The result nodes will be added after simulation result is available

Toolbars



File operation

Solid, face, edge & point selection

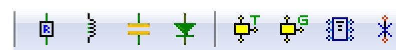


View operation

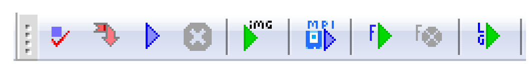
Extended view control

Redo & Undo

Port in EM solver

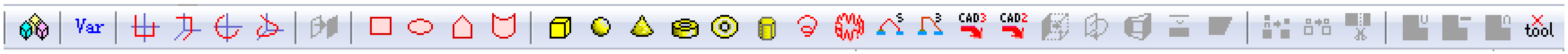


Lumped circuit in EM solver



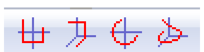
Simulation controls for different solvers

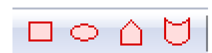
Curve, Face & Solid creation

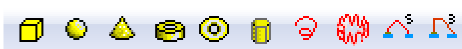


 Project setting

 Variable creation and editing

 Curve creation

 Face creation

 Simple 3D solid creation

 Create 3D solid by sweep, loft, shear, split etc.

 Insert (1) 3D object from ***STEP***, ***SAT, IGES,*** or ***STL*** file
(2) 2D object from ***Gerber*** file

 3D solid BOOLEAN operation, including
UNION, SUBTRACTION & INTERSECTION

Cylindrical and Cartesian mesh generation and display

 Generate cylindrical grid and export the data file

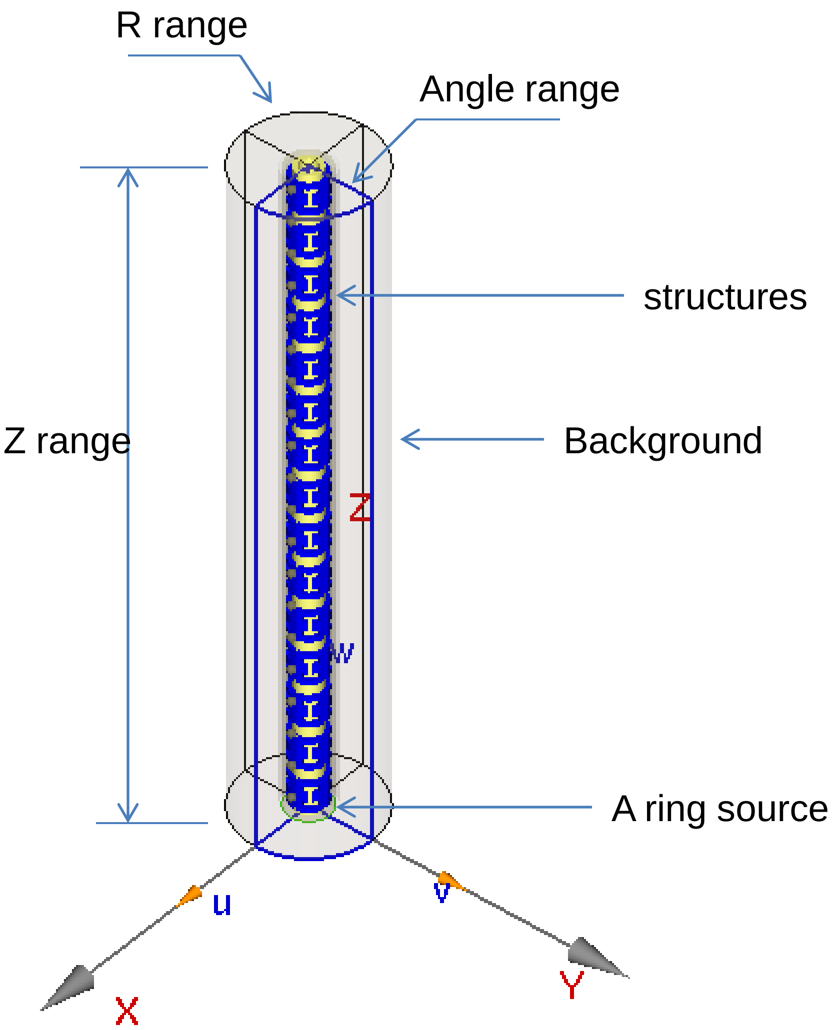
 Generate Cartesian grid and export the data file

 Display the cylindrical mesh

 Display the Cartesian mesh

Project Setup

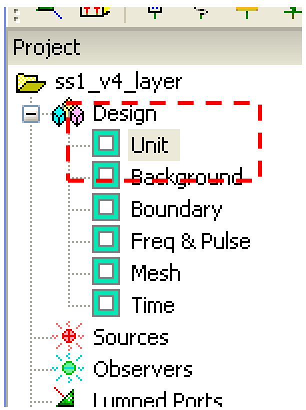
A typical BHA project looks like the figure.



Project Setting

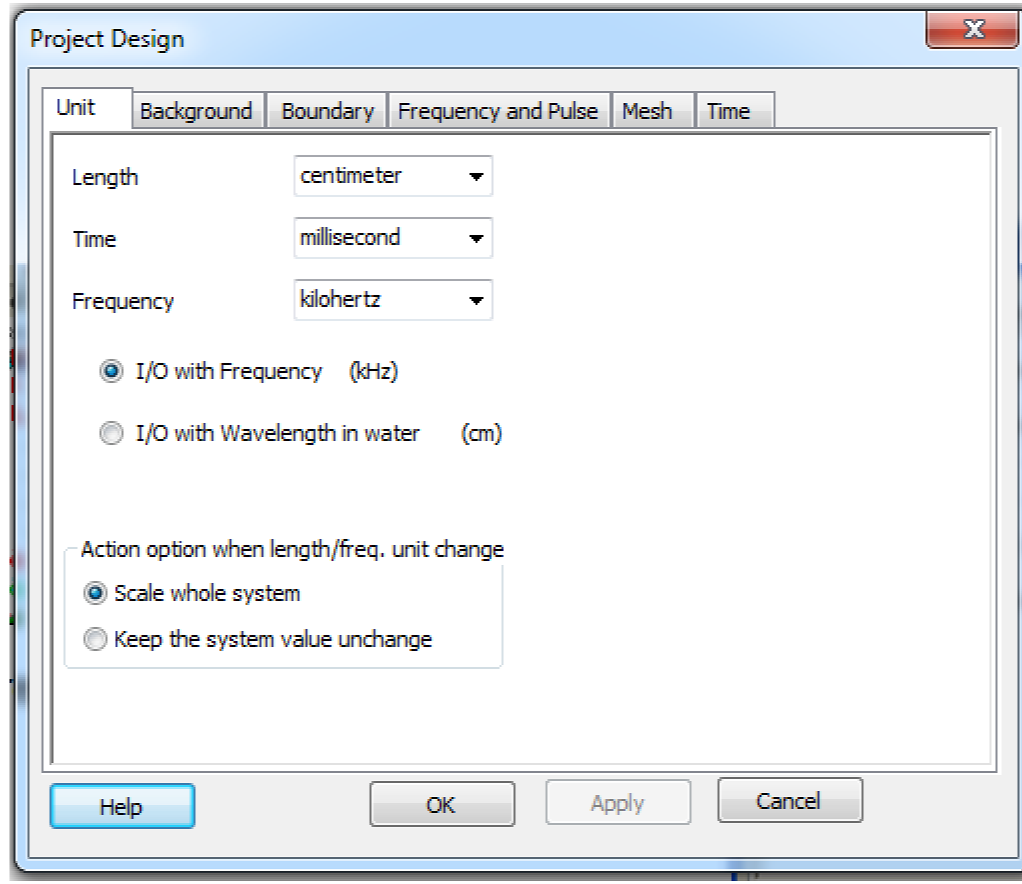
- In Wavenology GUI, we define a simulation case as "a project"
- A project includes
  - Units
  - Materials
  - 3D Solids
  - Excitation pulse & frequency range
  - Mesh settings
    - Automatic mesh by sampling density: PPW (point per λ)
    - Manual uniform mesh
    - User input grid
    - Advanced mesh: each axis can be one of 3 mesh types as above

Project Unit



Treenode ***Unit***

toolbar button "***Design***"



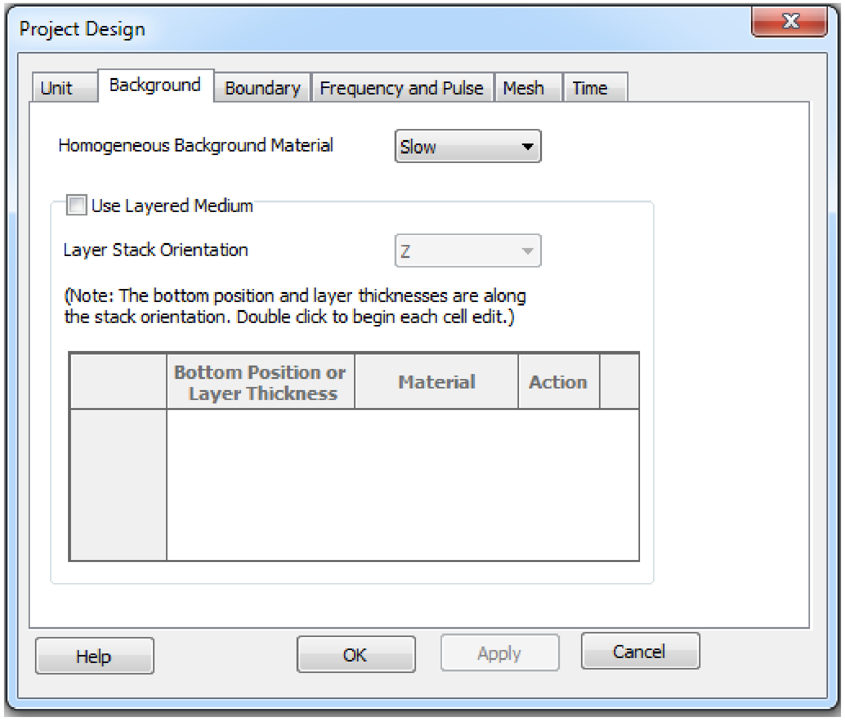
By double click the treenode "Unit" or click toolbar button "Design" to enter the ***Unit*** page of ***Project Setting dialog***

The default frequency unit is **Kilohertz**, for a BHA application.

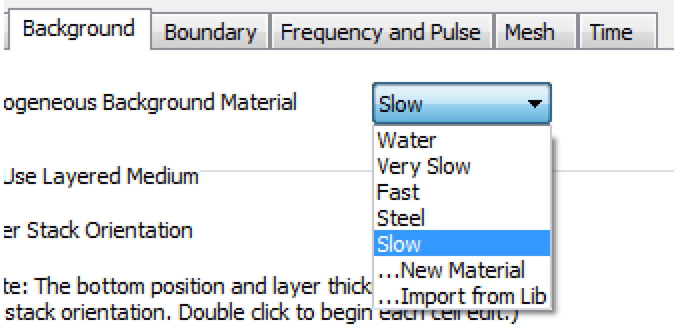
Project Background Material

The project requires a background material.

The default background for an EM problem is air. For borehole acoustic, the background is water.



The background can be changed to other material by using the ***material combobox***



Project Boundary Conditions (B.C.)

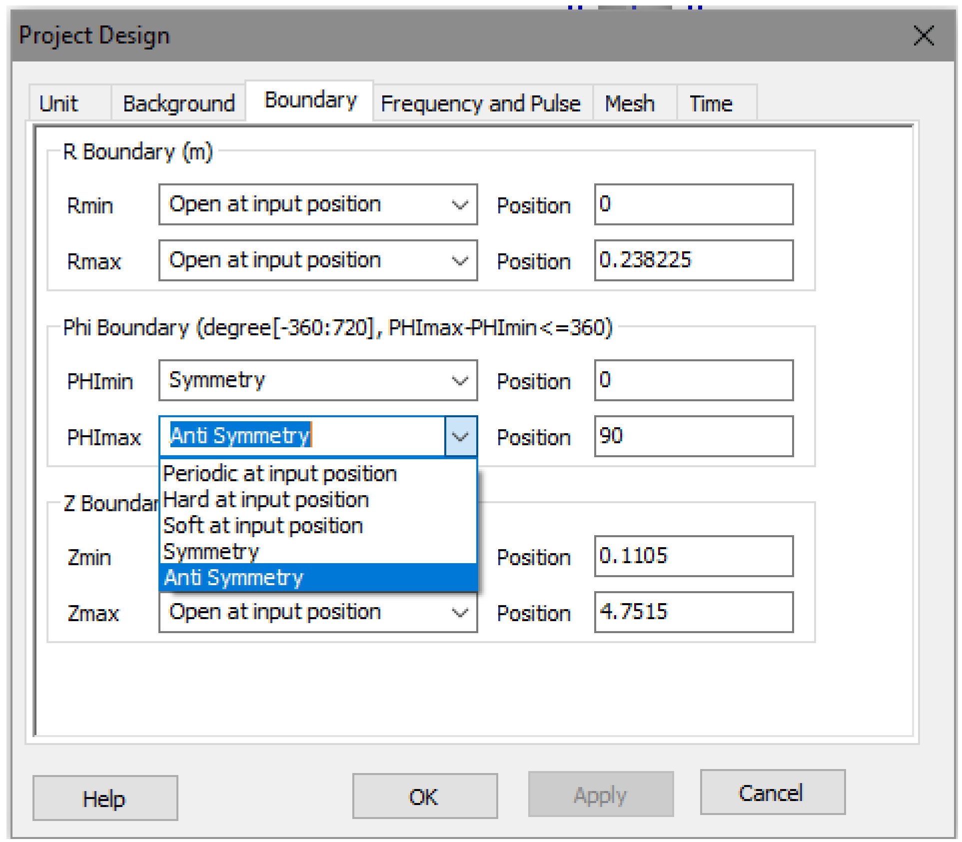
The boundary conditions should be set up correctly for the project.

General boundary conditions can be absorbing B.C. (Open at input position), soft, hard and periodic B.C. etc.

If R_{min}=0, the B.C. at R_{min} must be ***OPEN***, means that it is transparent for wave propagation.

In ϕ direction, if it covers 360^0, WCT prefer the range as [**-180^0, 180^0**]. The range can be [0, 360^0] also, but has limitations. The detail B.C. types in ϕ include *periodic, hard, soft, symmetry, anti-symmetry*. For a 360^0 range in ϕ, the boundary condition in ϕ must be *periodic*.

For more details, please refer to the section "***Dealing source type with boundary conditions***"

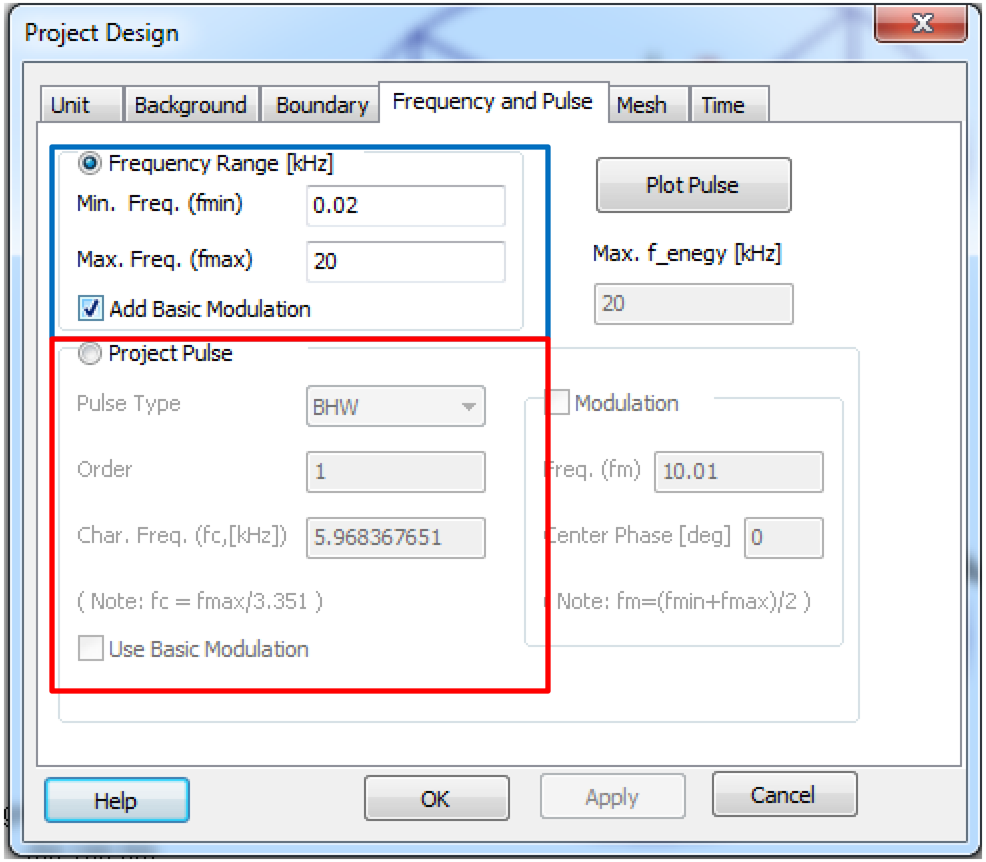


Project Frequency Range and Excitation Pulse for Source(s)

There are 2 choices in defining the excitation pulse for the source(s):

1) Using BHW pulse by defining f_{min} & f_{max} only

2) Using one of WCT intrinsic pulse types, for example, Ricker wave. User need to define the parameters of this pulse, the corresponding f_{min} & f_{max} will be calculated by the input parameters



The bandwidth (f_{min} & f_{max}) of the excitation signal is required for following purposes:

1) The spectrum of the simulation result will be limited to this range
2) The discretized cell size in the simulation is determined by $f_{\mathbf{max}}$ if the automatic mesher is requested.

In general, $f_{\mathbf{min}}$ is used for defining the lower bound of spectrum data only.

One consideration is, for a fixed f_{max}, a smaller f_{min} will make the bandwidth larger, the corresponding time window of the excitation will become shorter, which can reduce the simulation time. However, f_{min}=0 will introduce DC component, it is not good to generate spectrum data. So, define f_{min}=1%*$f_{\mathbf{max}}$ will be good for most cases.

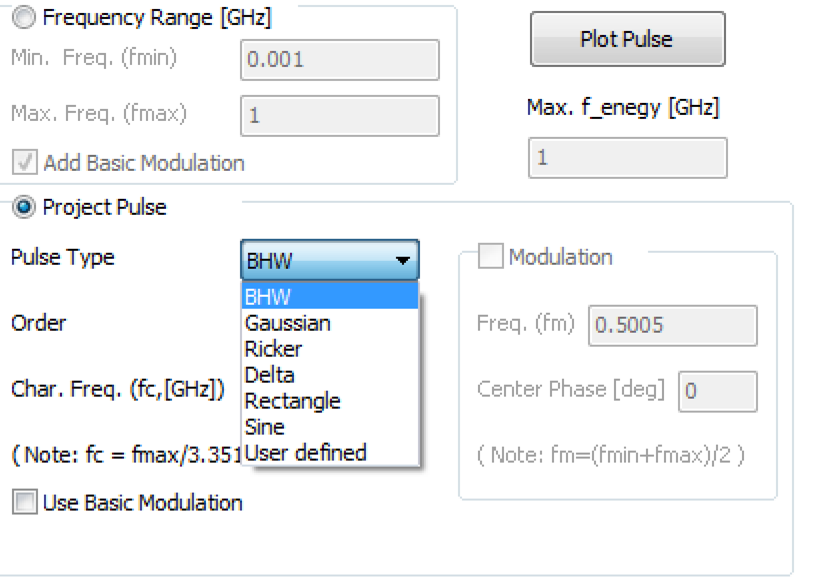
For the excitation pulse, the default pulse type is the modulated ***BHW*** pulse. But user can switch to other waveforms.

In current version, WCT support following intrinsic pulse types:

1) BHW (Blackman-Harris Window)
2) Gaussian – order 0, 1 & 2
3) Ricker
4) Delta
5) Rectangle – finite period(s) or infinite period
6) Sin - finite period(s) or infinite period

User can also load an waveform with arbitrary shape from an ASCII file.

For more details, please refer to the **Wavenology manual** in GUI.



(Note: the excitation pulse defined here is the default pulse for all sources in the project. But a source can use individual pulse instead of the project pulse. More detail can be referred to section "**Individual excitation pulse for each source**")

Cylindrical Mesh Setting

There are 3 basic mesh options:

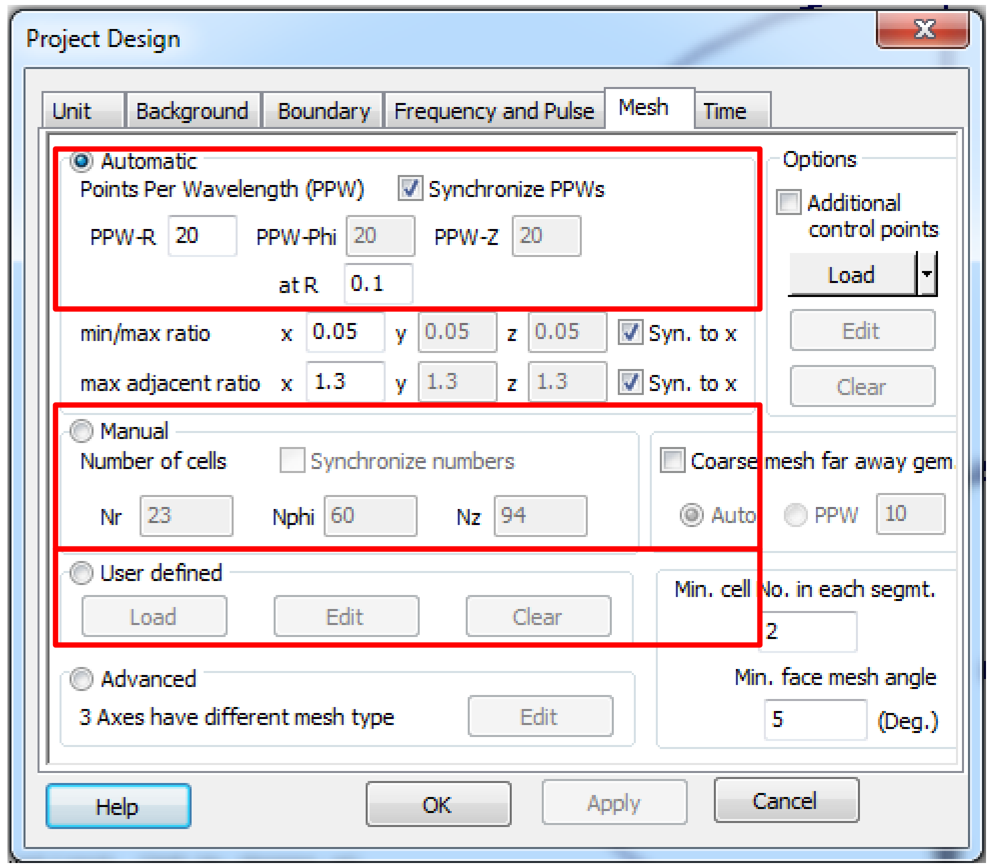
1) ***Automatic*** – the mesh density is based on λ

2) ***Manual*** – uniform mesh

3) ***User defined*** – the mesh grid is defined by user

For these 3 options, if they are employed, all 3 axes will use the same type of option, can't use mixed options.

If user want to use different mesh option in different axis, the "Advanced" mesh control should be used.



“**Automatic**” Mesh Setting

From version 2.0.8, the default mesh option is “Automatic”.

This option will generate a **non-uniform** mesh.

This mesh control is based on

1) The critical point of geometry – in general, it is the bounding box of solids

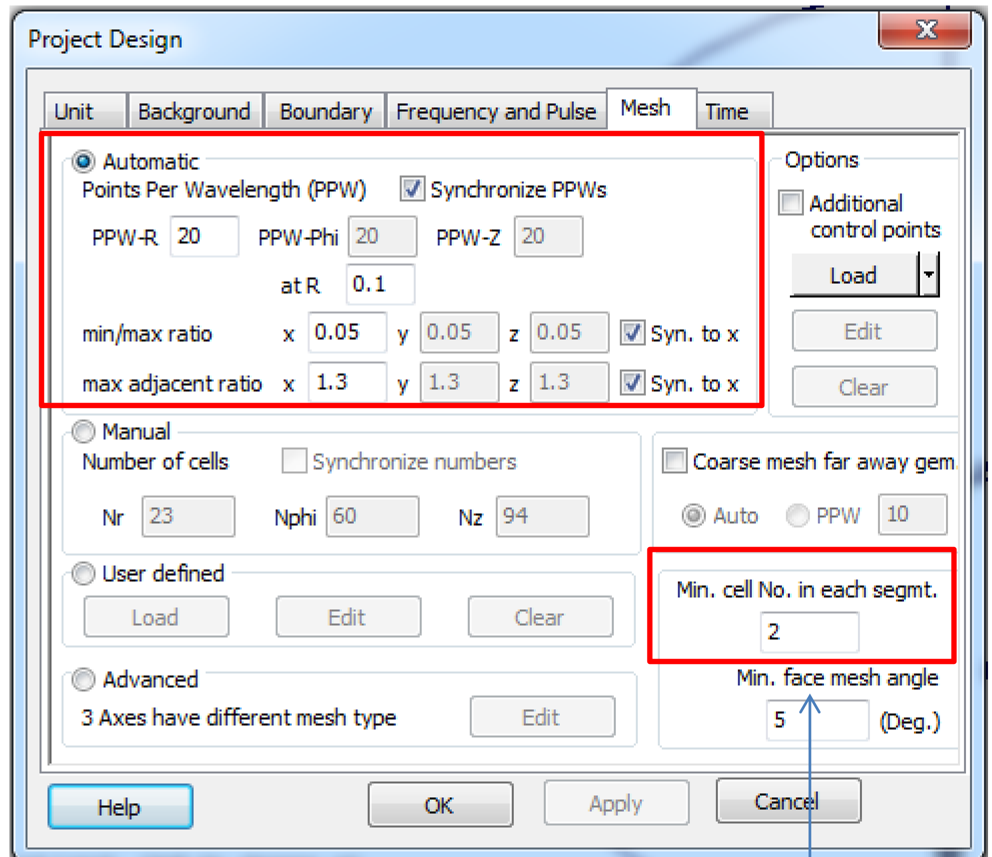
2) the wavelength (λ) – it is determined by project f_{max} & the material property

3) PPW – the number of cells per λ. A typical value is 20 for BHA applications

4) The ratio between the minimum cell size and the maximum cell size

5) The adjacent cell size ratio in the non-uniform grid

6) The minimum cell number for a mesh segment (if the input is a single number, for example, “2”, means that all 3 axis use the same input. however, user can input 3 numbers, for example, “2;3;4”, to let each axis has a different minimum cell number for a mesh segment)



The impact of Min/Max Ratio

this ratio = min-cell-size / max-cell-size.

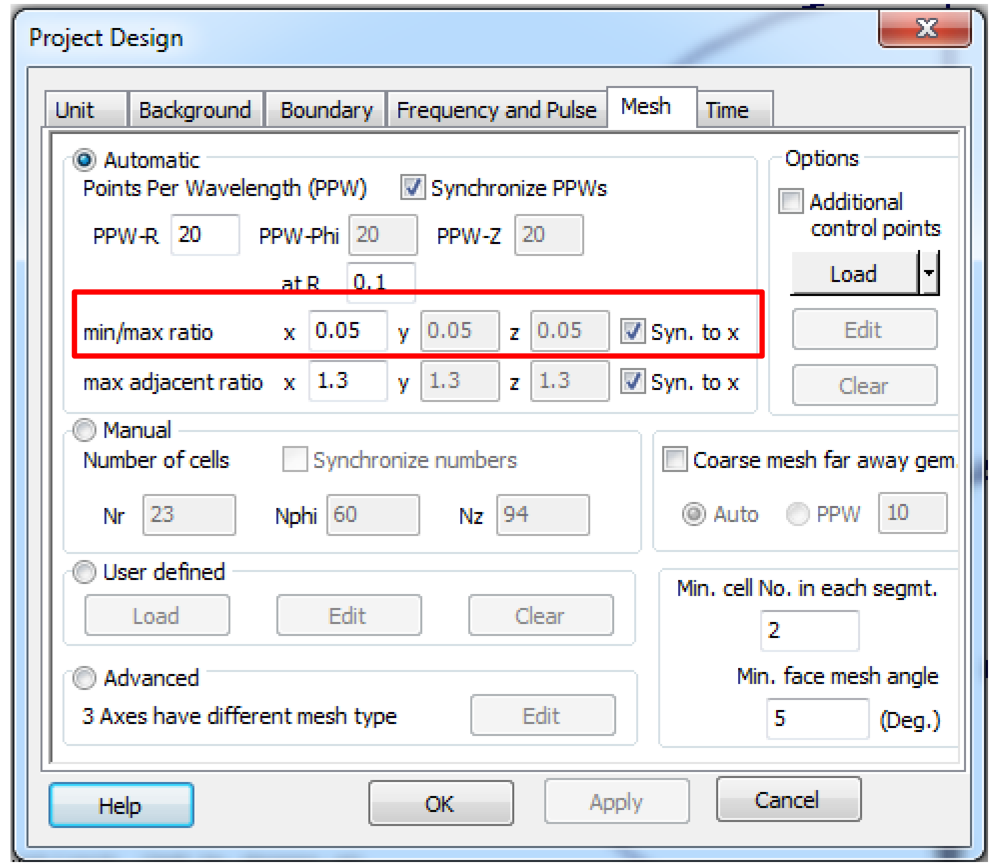
In general, for a BHA application, the maximum cell size can be determined by: λ/PPW. So, with this ratio, the smallest cell will be: λ/PPW*Ratio. This means that, user can use this ratio to define the possible smallest cell in the simulation. The fine details in the solid will be skipped if the size of the detail is smaller than the smallest cell size.

The smallest cell is very critical in the simulation.

1) **Good part**: smaller cell means that the details of solid can be captured correctly
2) **Bad part**: smaller cell means more simulation load
    1) more cells
    2) a smaller Δt

Therefore, a reasonable ratio will be very important for fine structure to obtain a balance between the simulation accuracy and the simulation time.

According to our experience, 0.05 can satisfy most situations.

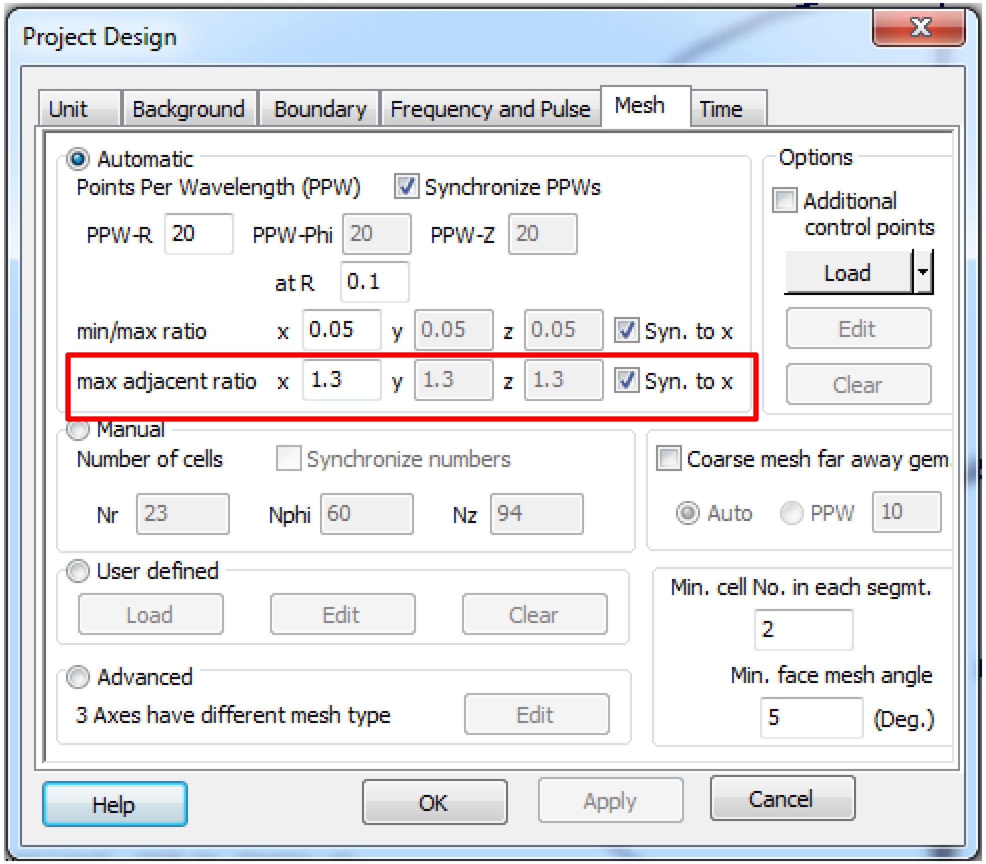


The impact of Adjacent Cell Ratio

For any 2 adjacent cells in the project, this ratio = max_size(these 2 cells) / min_size (these 2 cells)

WCT BHA solver employs the FDTD method. In principle, a uniform cell size will provide a 2nd order accuracy. If the cell size become non-uniform, it will introduce differential error and degrade the accuracy. In order to keep a good balance between the accuracy and the non-uniform cell, the adjacent cell ratio should be in a range.

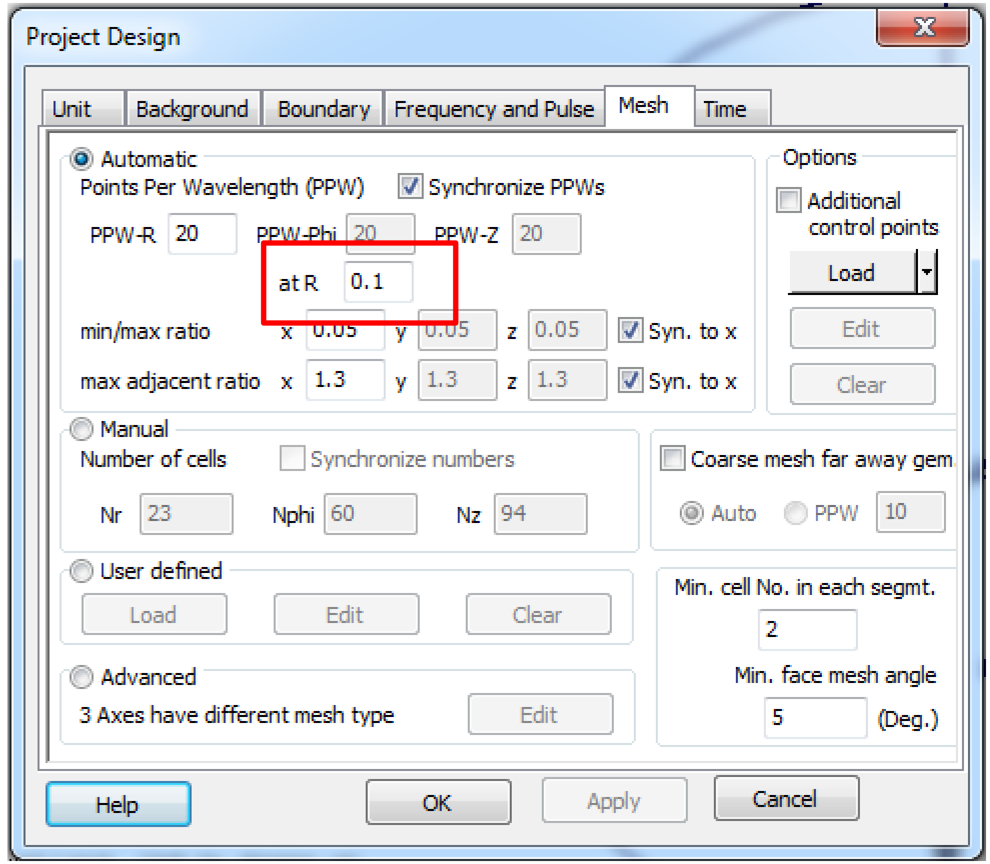
According to our experience, the default value 1.3 can produce an accurate enough result for most situations. If user want to obtain a higher accuracy, he need to lower this ratio, or use an uniform mesh.



Special Input to define PPW in ϕ

It requires special attention to the PPW in ϕ.

In the cylindrical coordinates system, the circle length will change by the radius R, which will make PPW_{ϕ} changed by R. For example, at Rmax, Lcir=2πRmax; if we define PPW=20 at Rmax. At 0.5Rmax, due to L'cir=0.5Lcir, the PPW = 40. So, user need to define a R to calculate PPW in ϕ as the reference.



In most cylindrical applications, the wave will become a cylindrical wave when it is close to Rmax. Or, in a borehole structure, the wave outside the borehole will be very weak. So, it is unnecessary to use high PPW near Rmax.

In default, the PPW_{ϕ} =20 at R=0.1 m is good for typical borehole applications. However, for a ring monopole source with circular structures, due to the ϕ component of wave is 0, we can use 1 cell in ϕ axis. More information is available in the section "***Dealing source type with boundary conditions***".

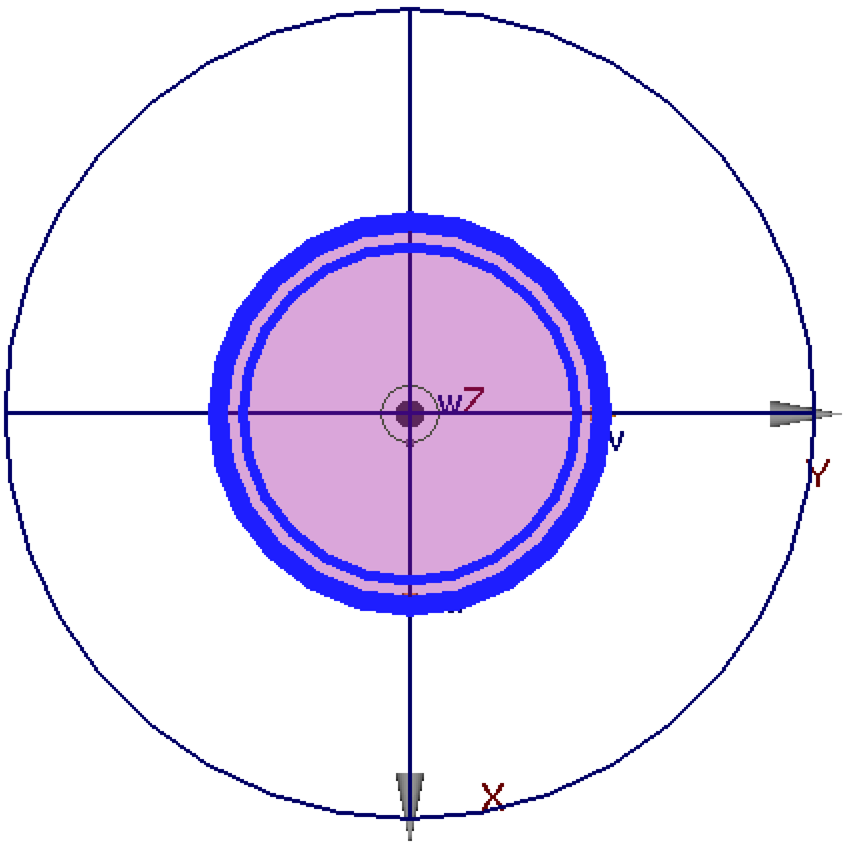
(Note: for some cases, PPW_{ϕ} =20 is oversampled, $\Delta\phi=15^0$ can provide enough accuracy, for detail information, check "***Space Sampling Density in ϕ***")

Following is an example of “Automatic” mesh

The figure is the top view of 2 steel pipes placed in a borehole (the **blue** color rings). As can be seen, the inner pipe is thin. In order to get an accurate simulation result, in principle, there will be at least 2 cells in R direction in this thin steel pipe.

Another requirement is, in order to get an accurate simulation result, the mesh grid should be exactly on the interface between 2 different materials.

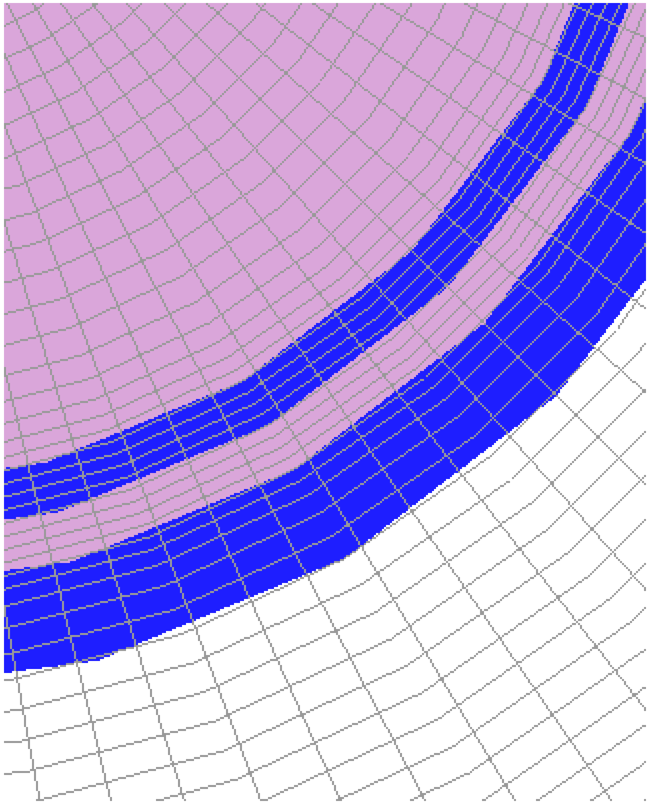
However, due to the dimension of borehole and the steel pipes can be arbitrary values by real situations, a uniform grid is very hard to satisfy this purpose; or user need to use very fine mesh to get it - this will cause huge simulation memory & time.



The “Automatic” mesh control can make a good mesh for thin structures

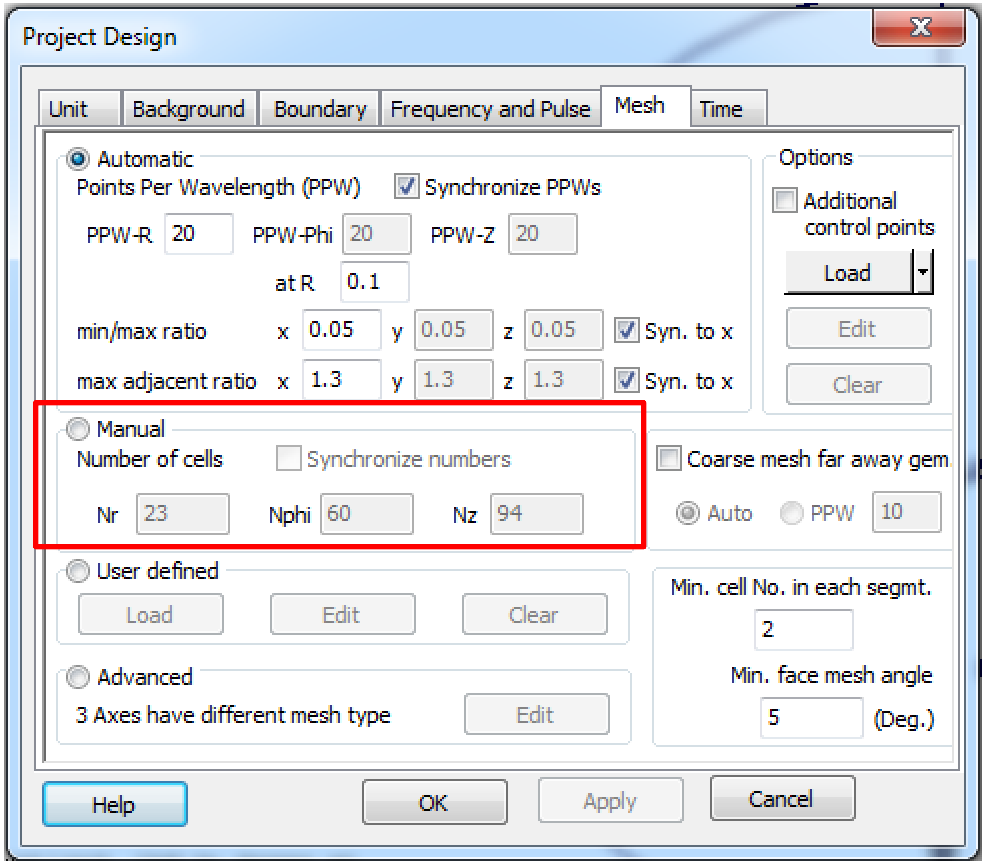
1) The mesh grid is exactly on the interface between 2 materials
2) Very fine mesh in fine structure
3) Coarse mesh in big geometries
4) The smooth variation in cell size will keep the differential error small
5) In general, the cell size close to absorbing boundary has a large cell size, which will make the absorbing boundary have a better performance

Right figure is the mesh grid generated by the “Automatic” mesh control. Here, we set “***the minimum cell number for a segment***” as ***4***.



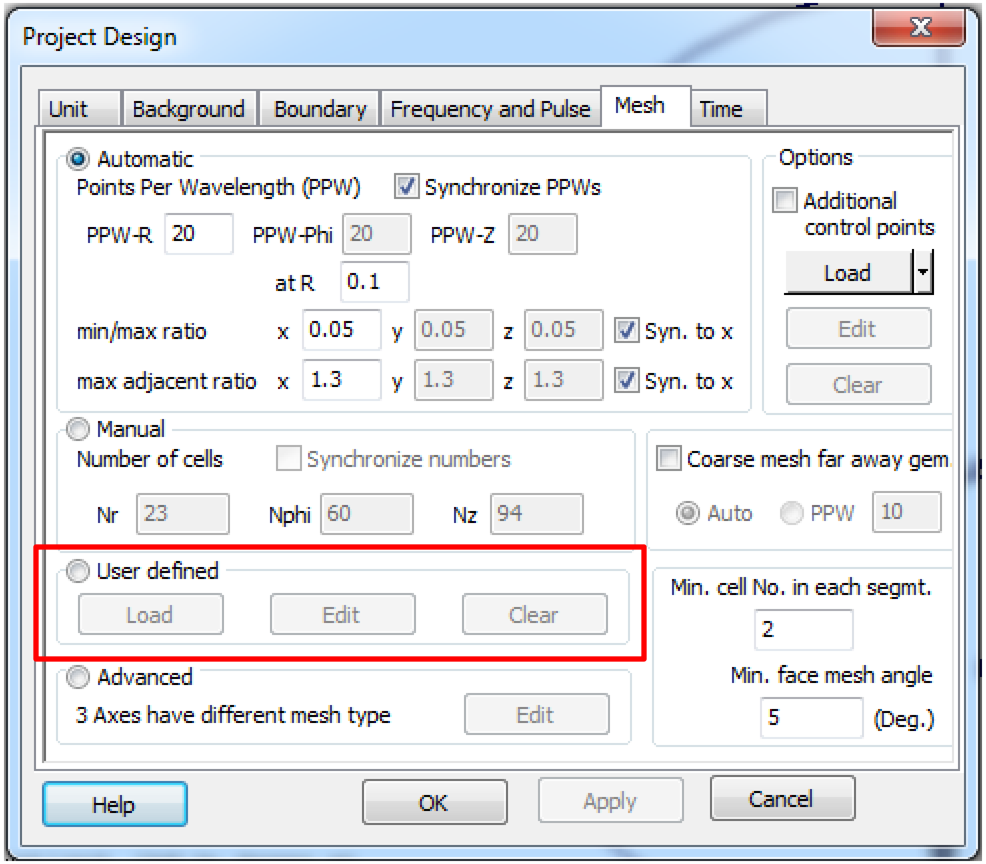
"**Manual**" Mesh Setting (Uniform Grid)

This option will generate a **uniform** mesh.



“**User defined**” Mesh Setting

The grids in this option come from user defined data array.

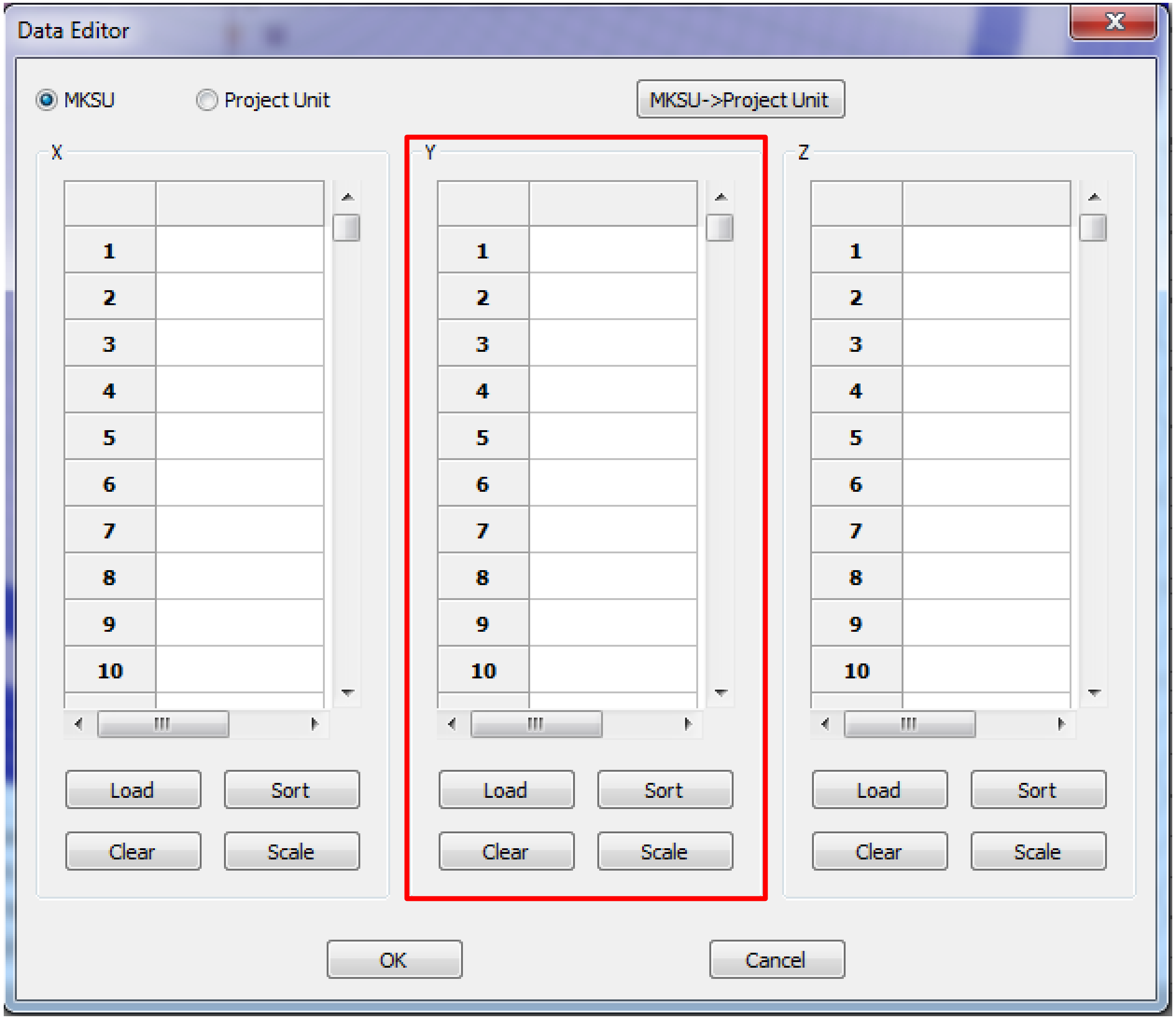


User can define an arbitrary grid array for each axis, not matter it is uniform or non-uniform.

The grid can be loaded from data file, or input by dialog. All data are editable after input.

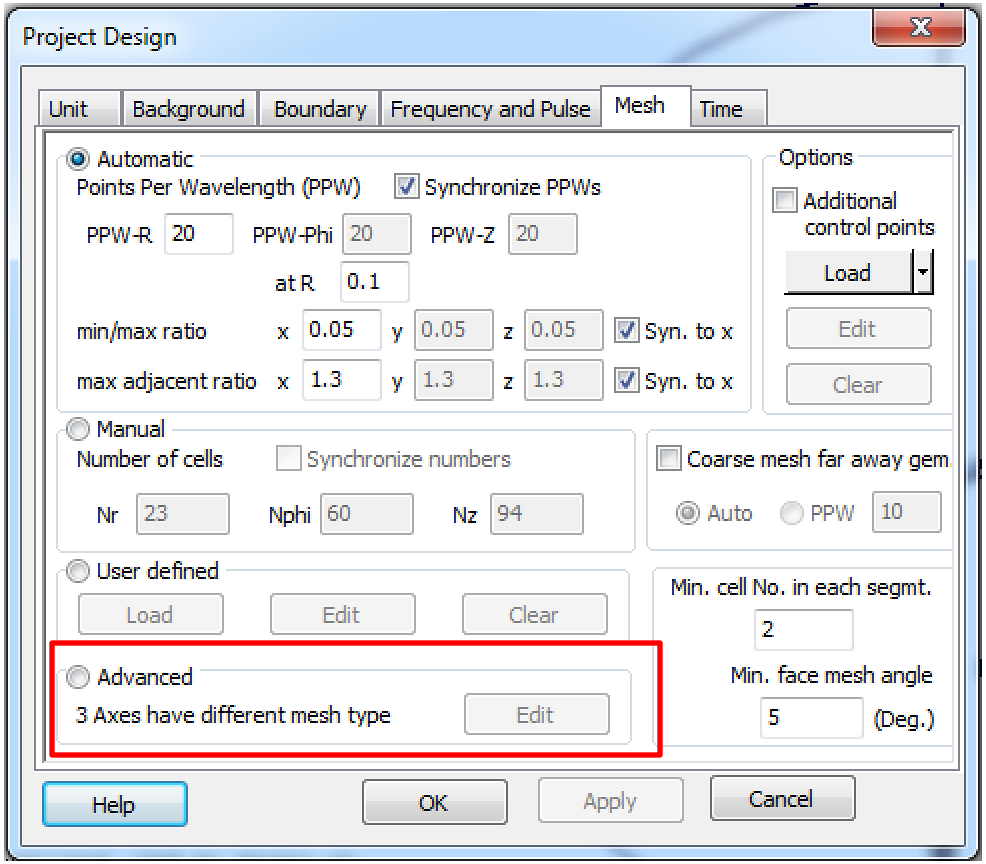
(Note: the 1st & the last grid position must match the project boundary position.)

Editor for the user defined grid



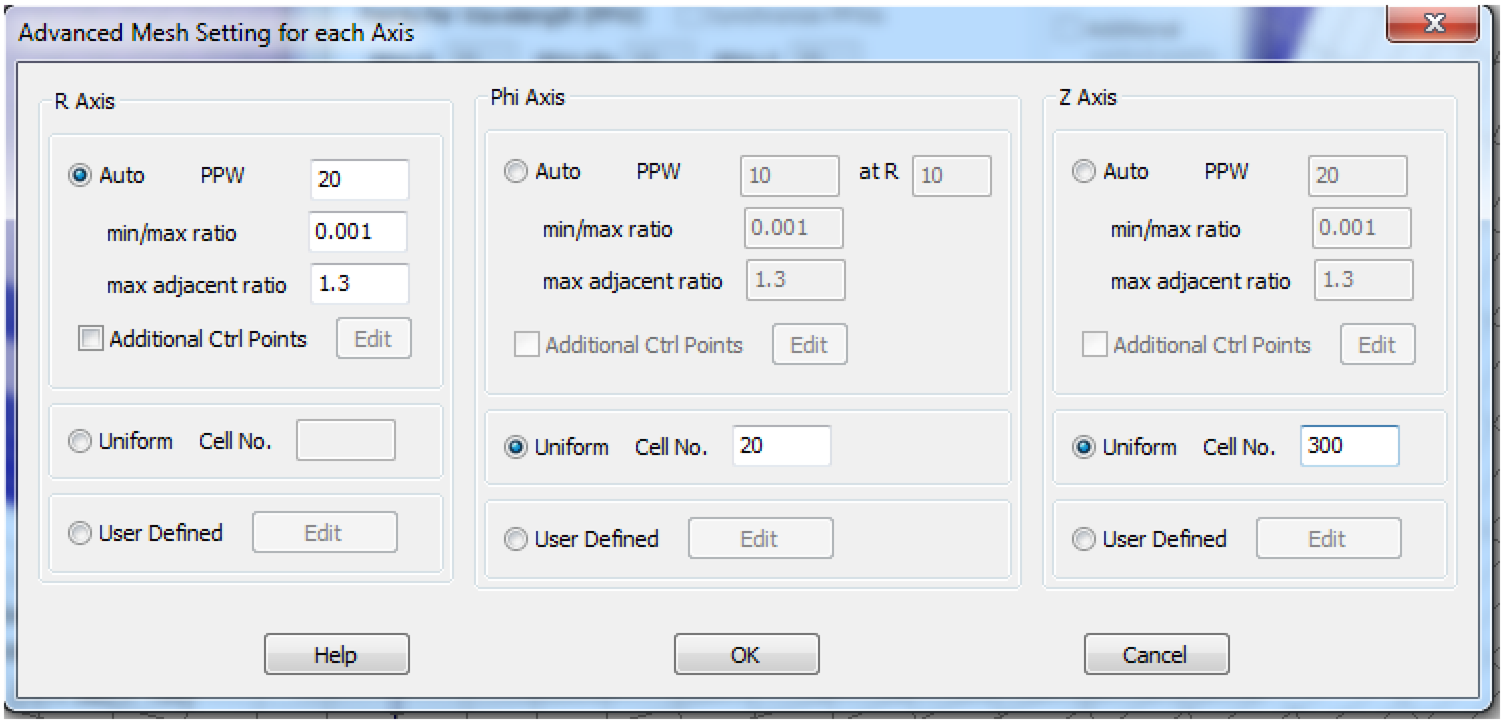
For ϕ grid, the unit is degree (^{o}).

“**Advanced**” Mesh Option



With this option, different axis can use different mesh type.

For example, it is “Automatic” mesh in R axis, uniform grid in ϕ & Z axes. As shown in the following figure



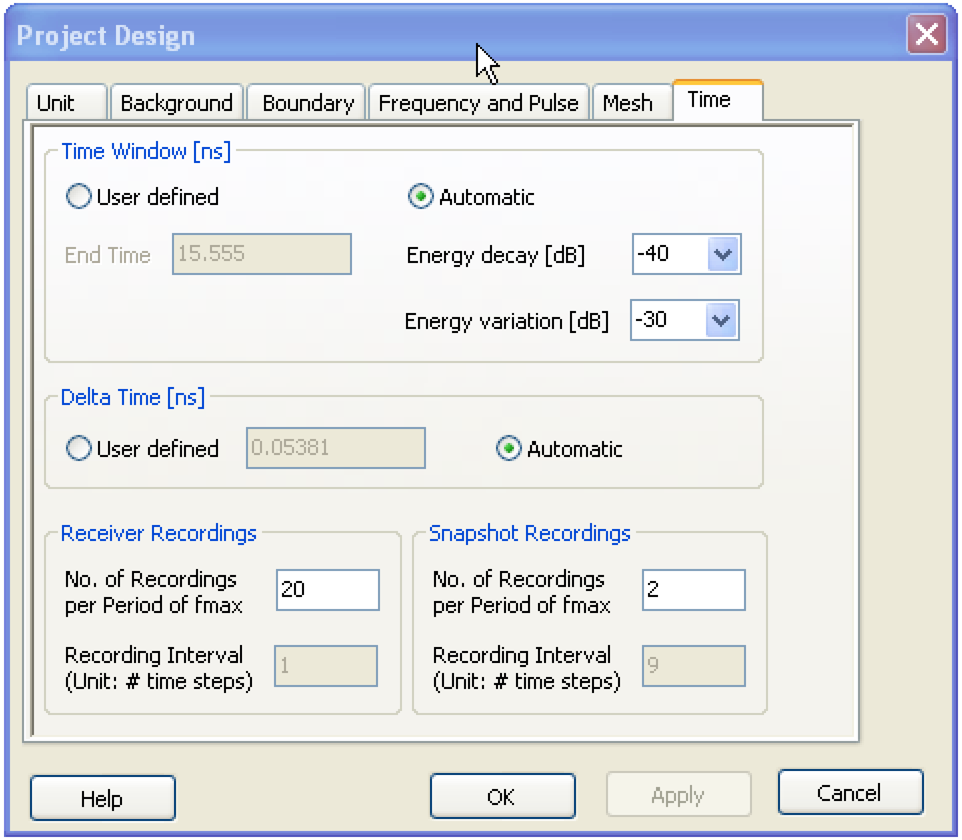
Simulation Timing Setup

- There are 3 timing parameters for a simulation
  1) **The total simulation time window**

     User can specify a fixed time window or using a automatic time window by energy decay level.
  2) **Stepping Δt**

     User can specify a Δt or use automatic Δt. The user defined Δt should be less than automatic Δt. Otherwise, the solver will prevent the simulation.
  3) **Receiver and snapshot recording intervals**

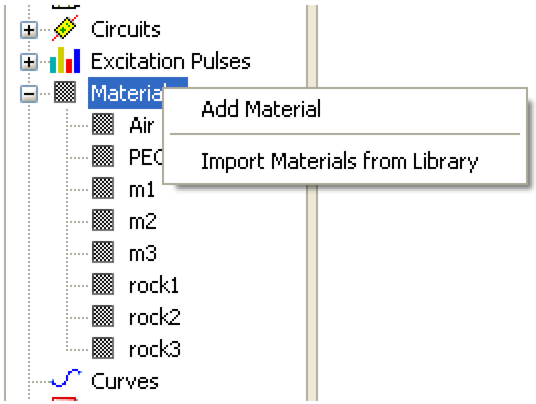
     The sampling intervals can be modified to any reasonable values.

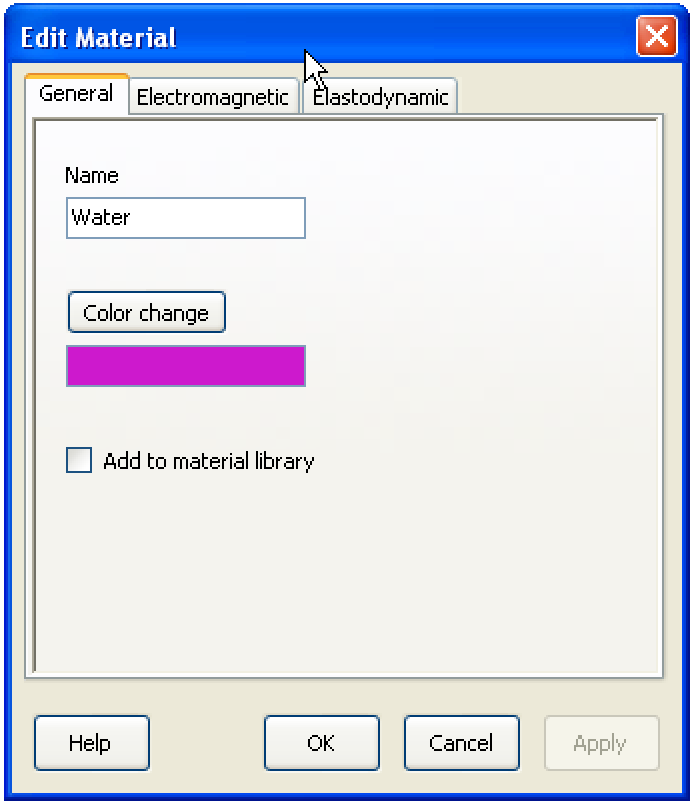


Materials in the Project

Material definition

In general, different solid uses different material. There is a menu attached on the tree-node “Material” to create new materials.

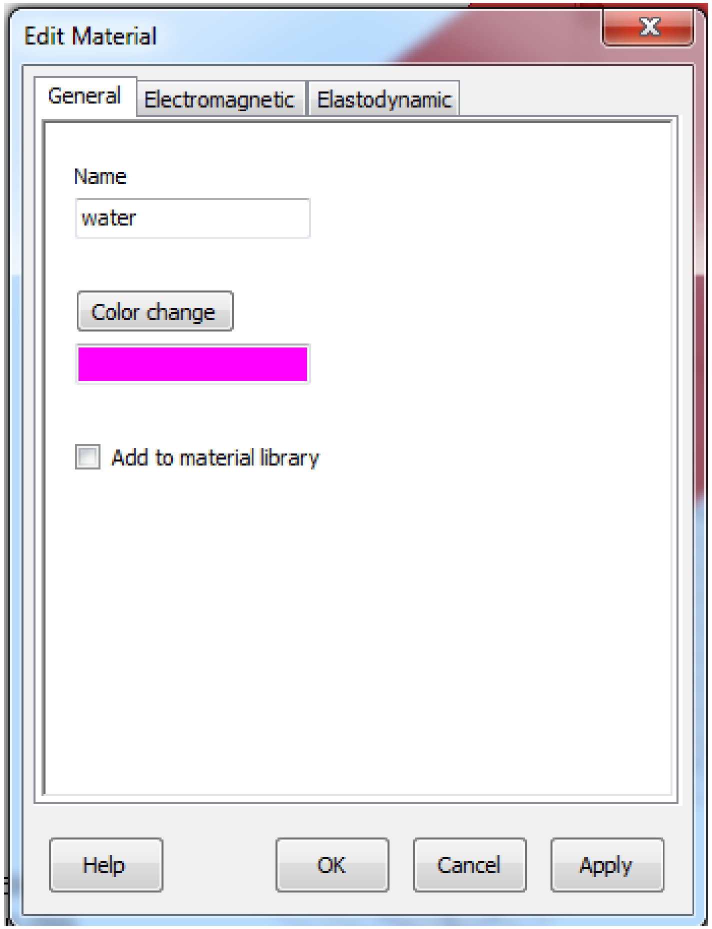




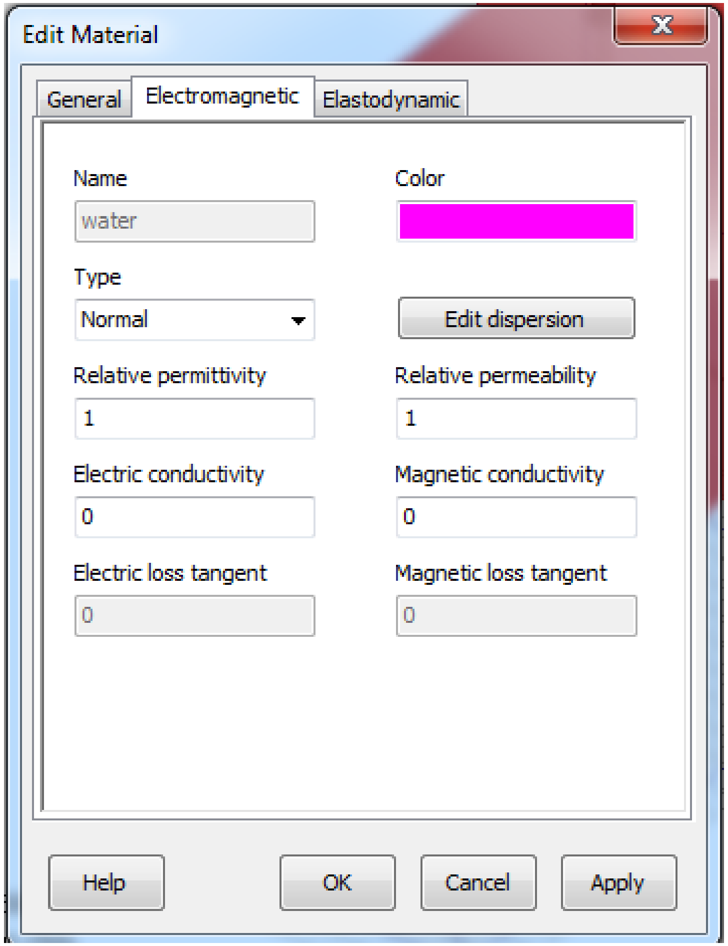
The menu appears by right click mouse button on the treenode **Material**, the **Material Editor** will be shown by choosing “**Add Material**” menu.

There are three kinds of properties for a material:

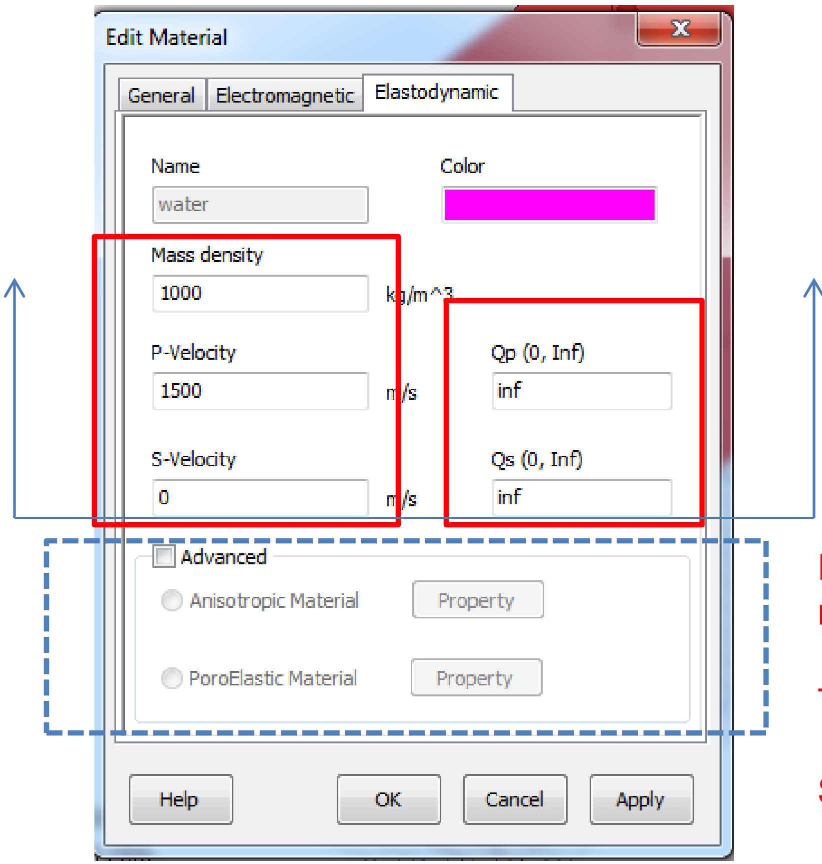
1. ***General definition:*** name and displaying color



2. Parameters for **electromagnetic profile**: relative permittivity and electric conductivity, etc. They are not used in a BHA project, so, user can skip this page.



3. Parameters for **elastic wave**: mass density, velocities (Vp & Vs) and Q factors.



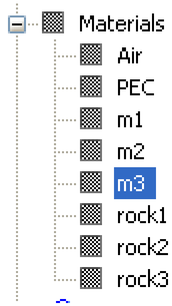
BHA project use these properties only

Note: WCT BHA does not support anisotropic material & Poro-elastic material in current version.

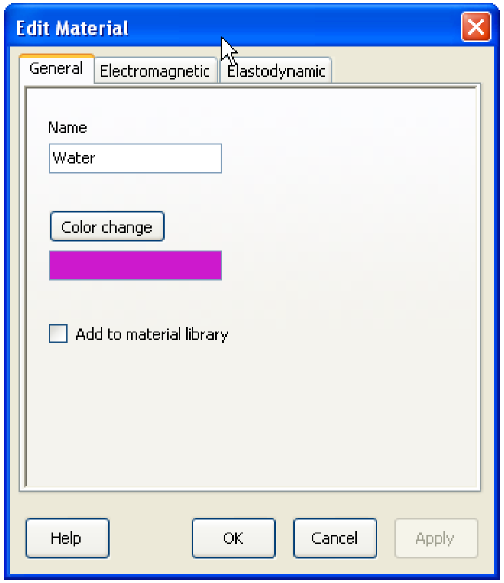
They are supported by WCT Cartesian EL Only.

So, don't define any parameters in this part.

Material Property Modification

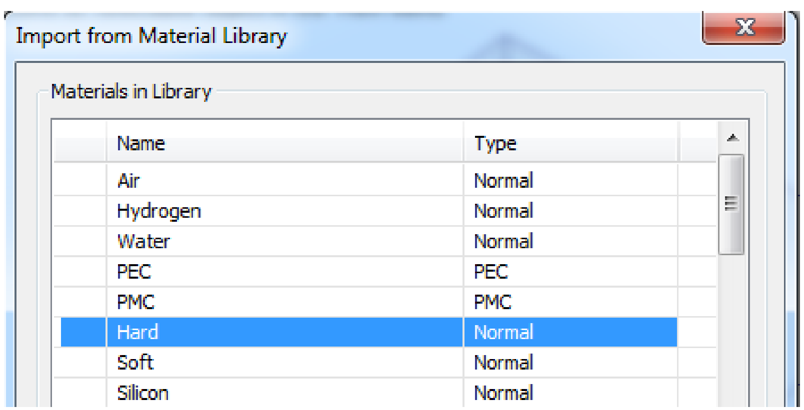


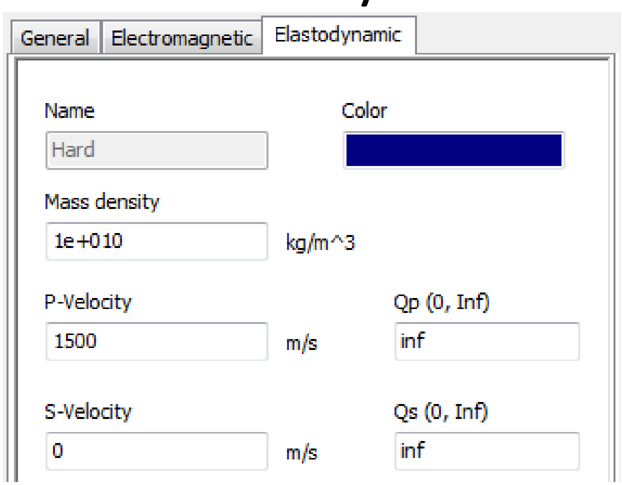
To modify a material’s property, user needs to select the material (the highlighted material tree-node item), then double click this item. The **Material Editor** will be open.



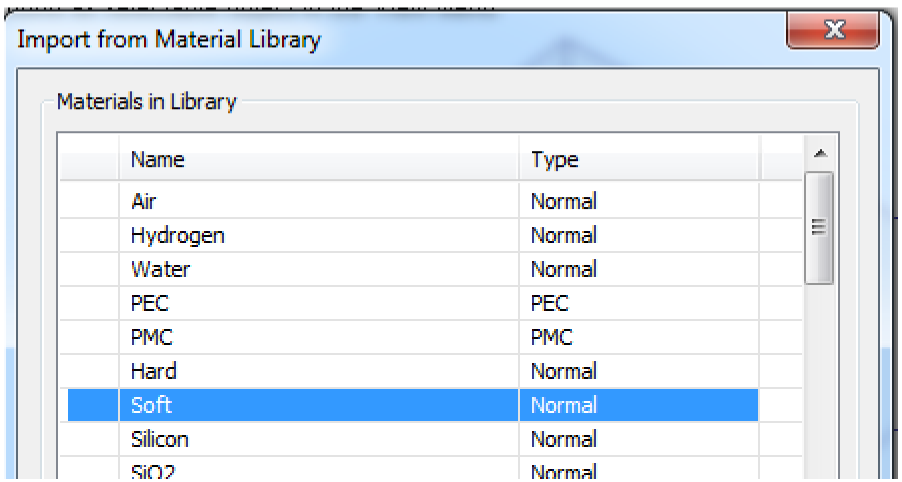
Special Materials

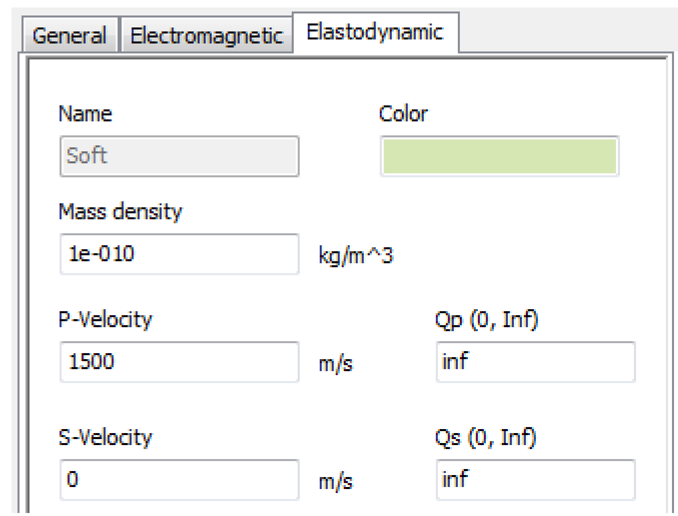
- HARD material
  - The particle velocity in the material should be always 0
  - In the WCT BHA solver, if the mass density of a material >= $1e^{10}$ kg/m^3, the solver will treat it as the hard material
    - user can import it from the material library as following





- SOFT material
  - The stress in the material should be always 0
  - In the WCT BHA solver, if the mass density of a material <= $1e^{-10}$ kg/m^3, the solver will treat it as the soft material
    - user can import it from the material library as following



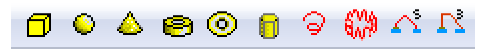


3D Modeling

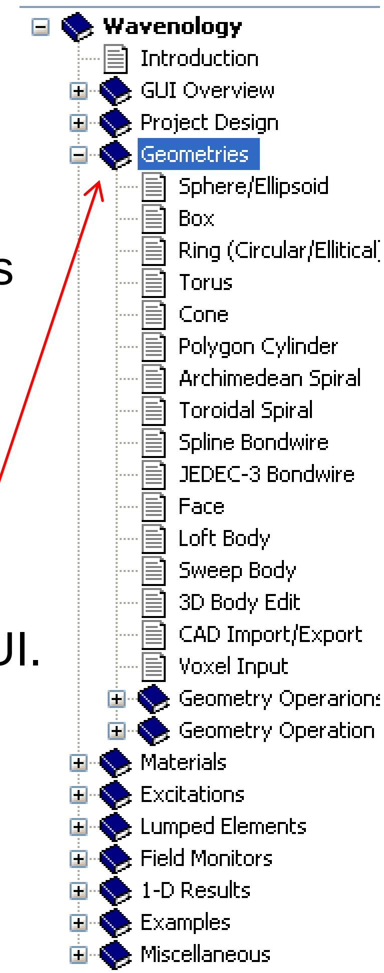
3D Solid Definition

Wavenology GUI supports following basic geometries:

Sphere/Ellipsoid
Box (Brick)
Ring (Circular/Elliptical)
Torus
Cone
Polygon
Cylinder Archimedean
Spiral
Toroidal
Spiral
Spline
Bondwire
JEDEC-3 Bondwire
CAD solid Import/Export



Basic solid creation toolbar buttons



For creating and editing these solids please refer to the "Geometries" section in the embedded manual of Wavenology GUI.

Wavenology GUI also supports complicated solid by sweeping or lofting faces.

- sweep along path
- sweep along axis
- loft face
- thicken face
- shear
- split solid

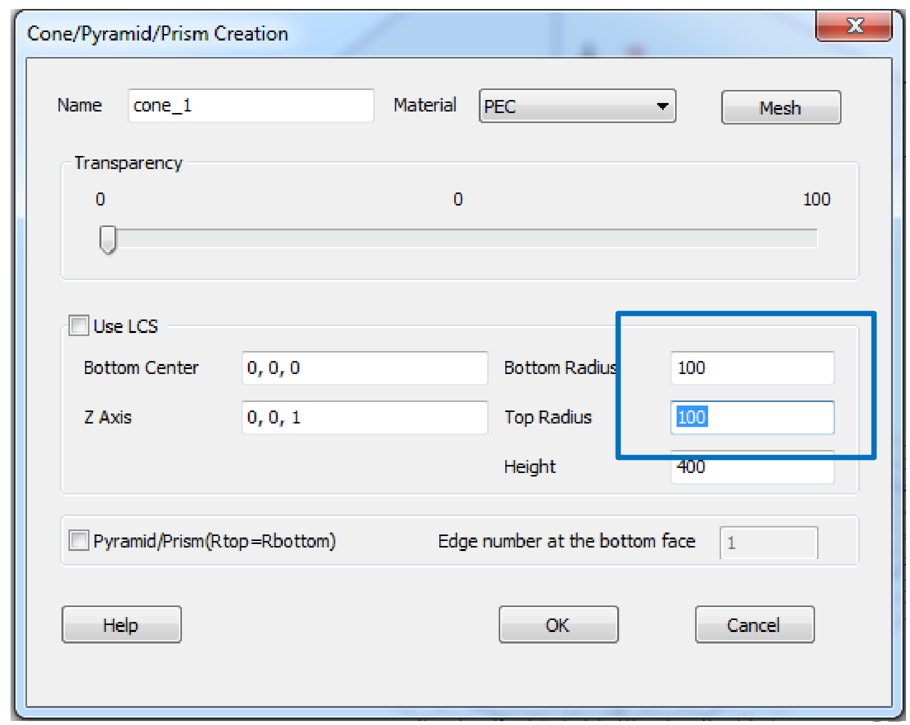
Most Popular 3D Solids in BHA Applications

- Cylinder (or partial Cylinder)

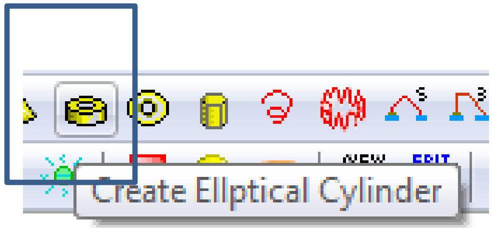
A full cylinder can be created by "***Cone***" or "***Elliptical Cylinder***"

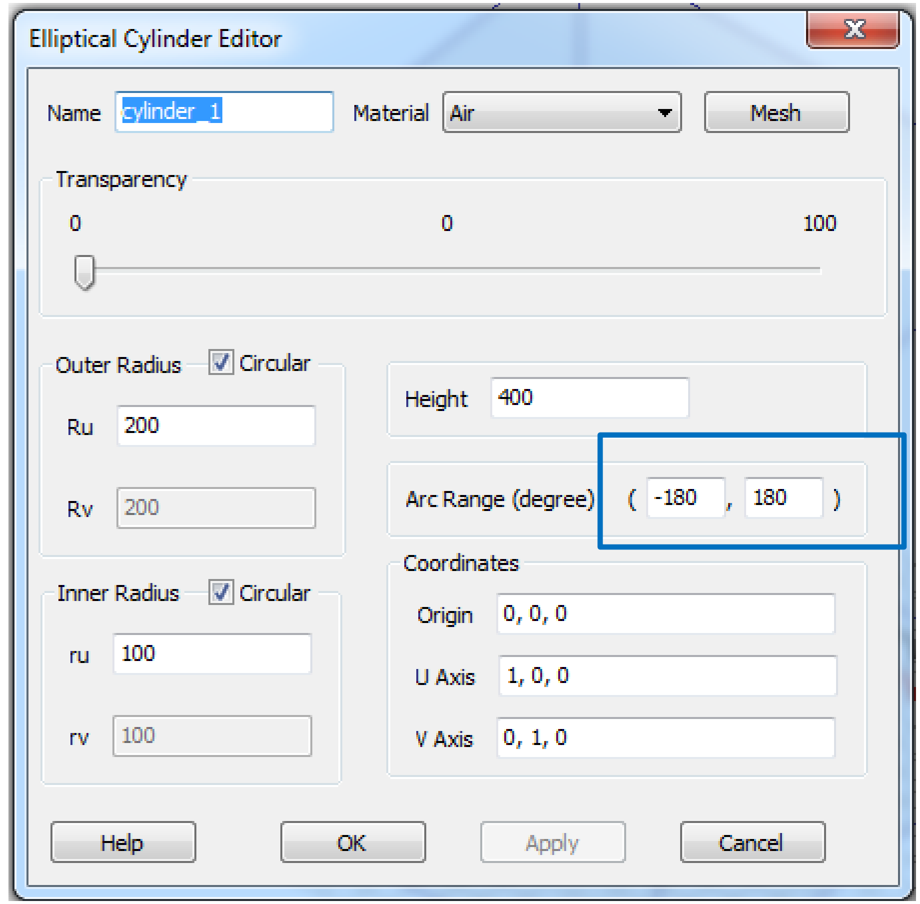
"***Cone***"

The top radius should be the same as the bottom radius

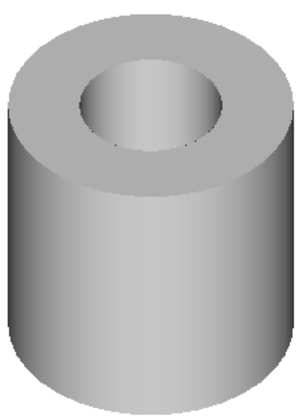


“*Elliptical Cylinder*”



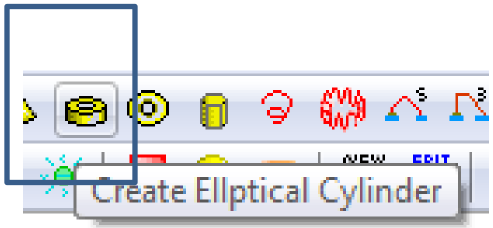


A full cylinder need to cover the 360^{o} range

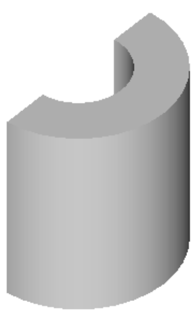


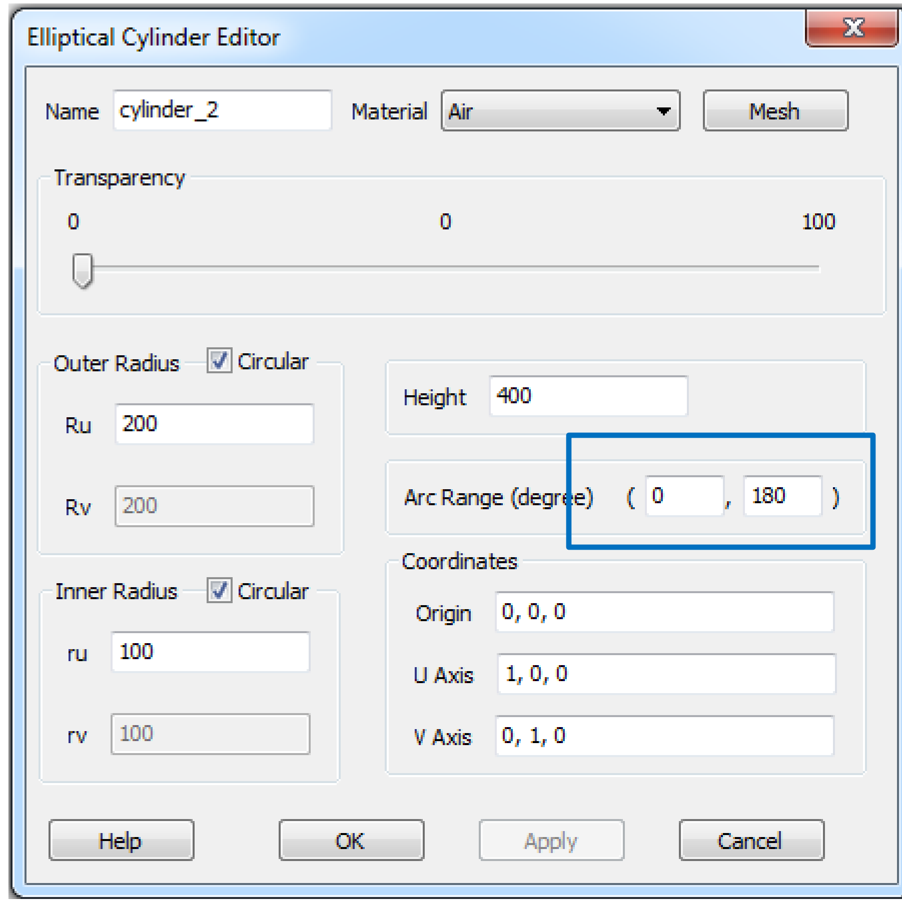
A Partial Cylinder can be created by "***Elliptical Cylinder***"

"***Elliptical Cylinder***"



A partial cylinder cover 180^{o} range

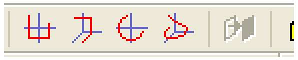




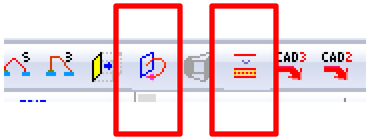
Another way to build a Partial Cylinder

Building Partial Cylinder from Line or Face

This procedure will use 2D & 3D curve creation, and cover planar closed curves as a face.



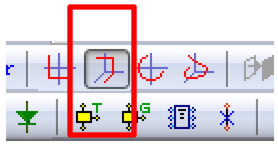
Then sweep this face along an axis with specified angle, or thicken the face directly.

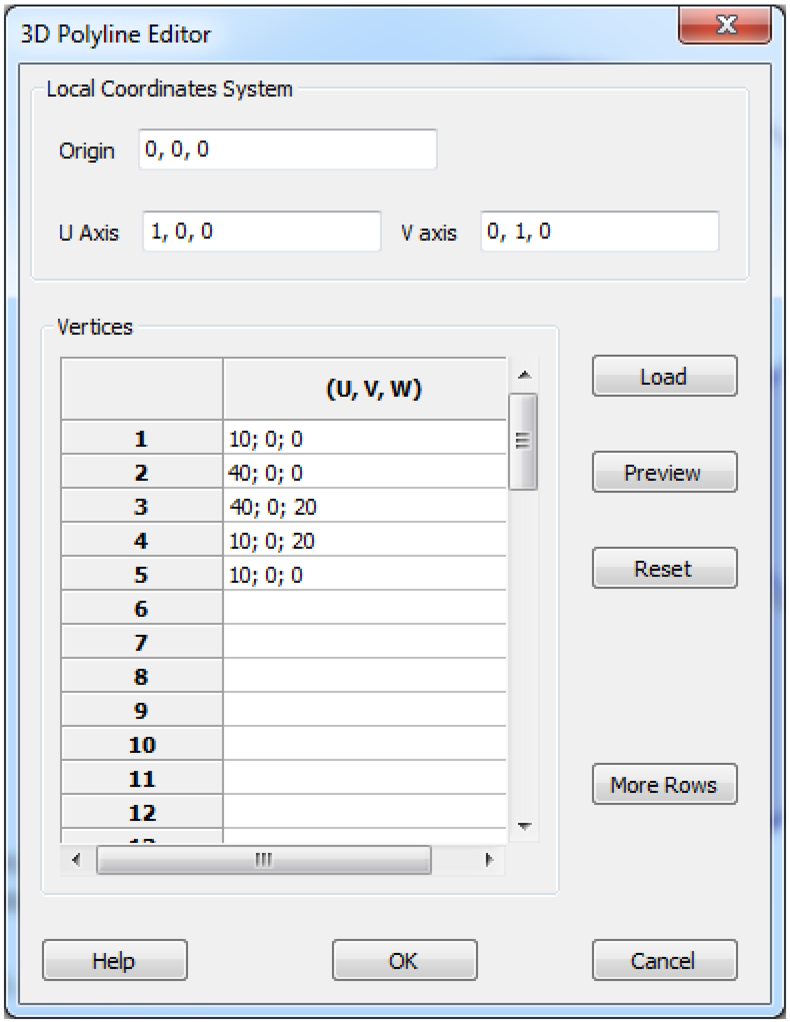


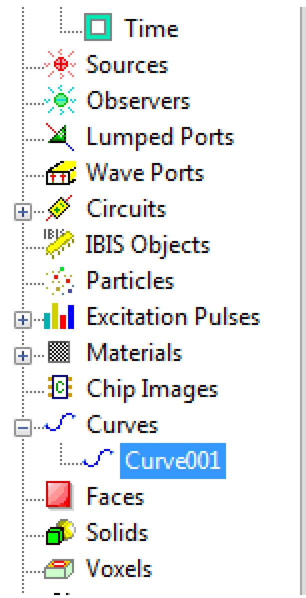
(Note: In the BHA solver, only the 3D solid will be used in mesh generation. The curves & the faces are not used in meshing. They are the tools to model 3D solid only.)

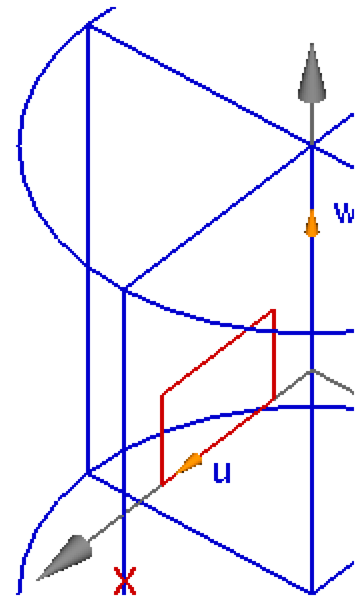
Following is an example to build a cylinder with 60º arc only

Define a closed 3D polyline



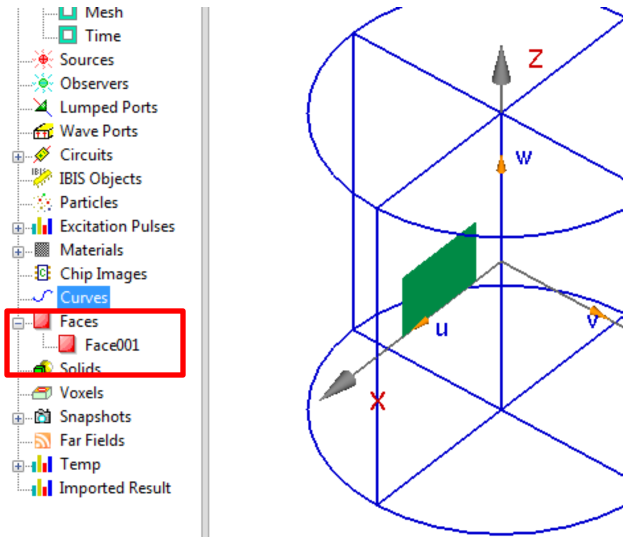




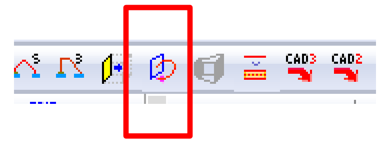


Select this closed polyline to enable "Cover Curves as Face" button

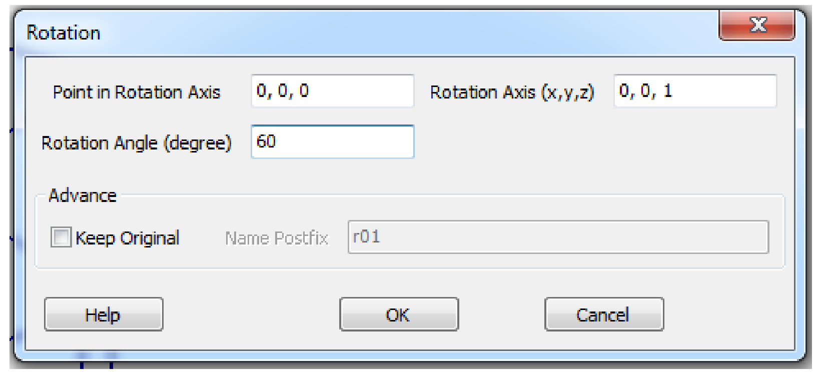
After the face is built, select this face



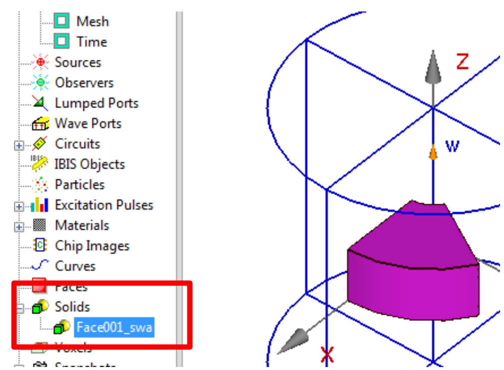
The toolbar item “Sweep along axis” will be enabled



Define rotation axis & angle

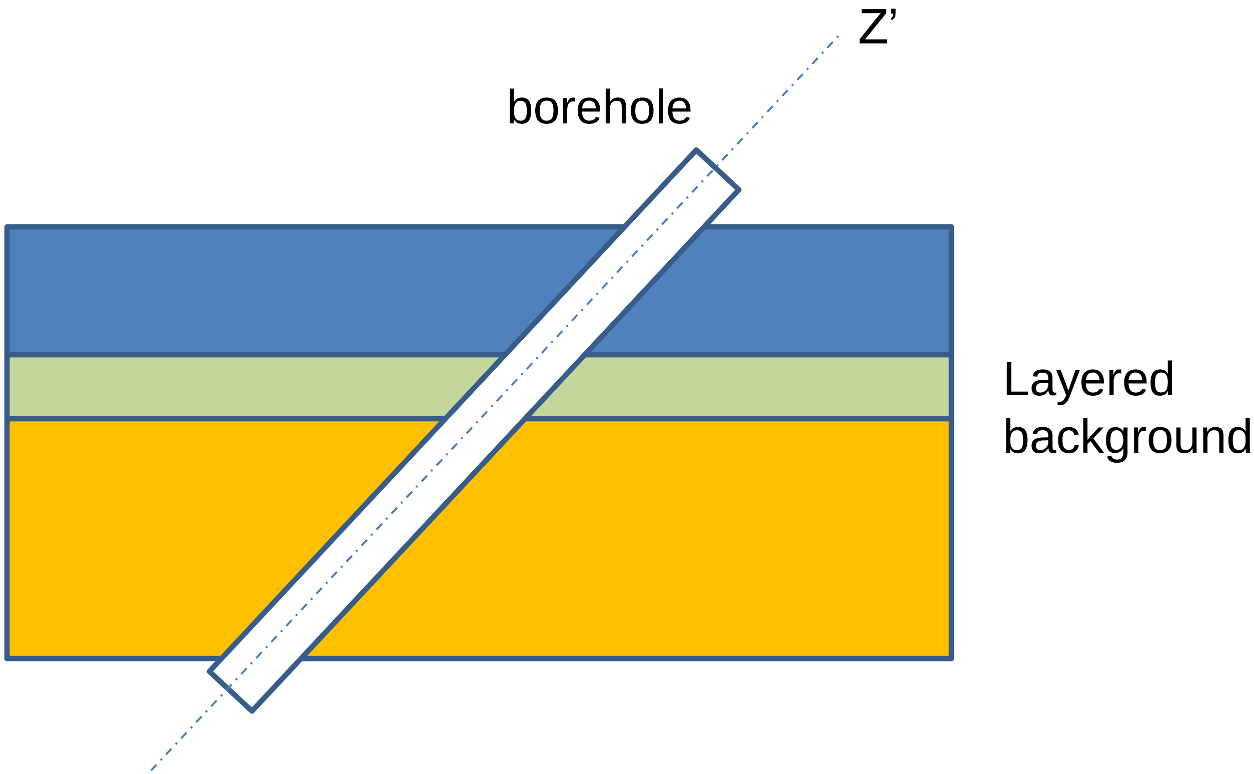


The new solid

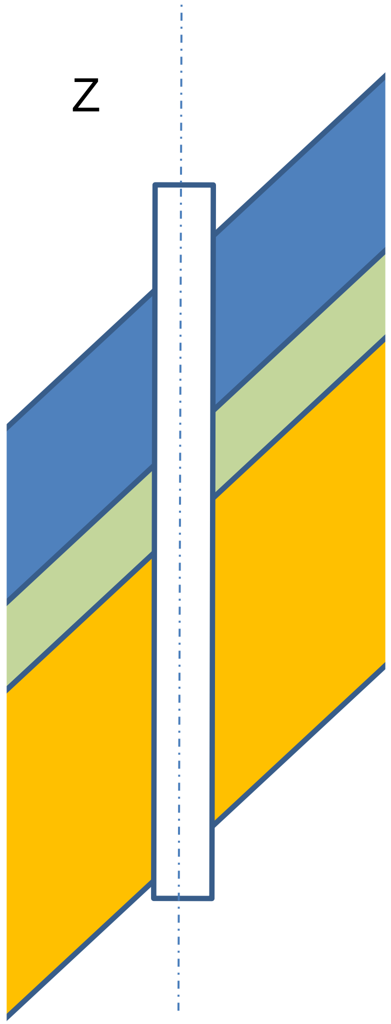


Tilted Layered Background Setup

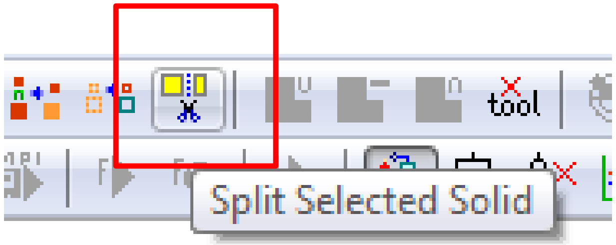
In order to simulate following situation



We can define the axis (Z’) of borehole as the Z axis of the cylindrical coordinates system, the case is equal to: a vertical well in a tilted layered background

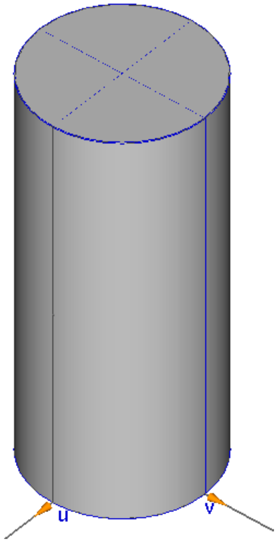


In WCT GUI, the tilted layered can be built by operation “Split Selected Solid”

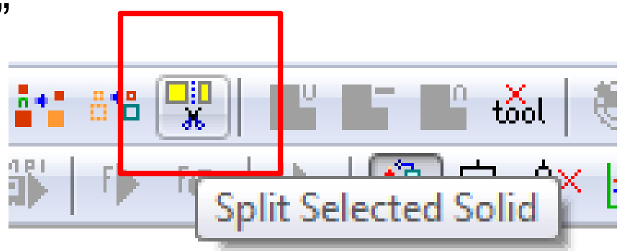


steps

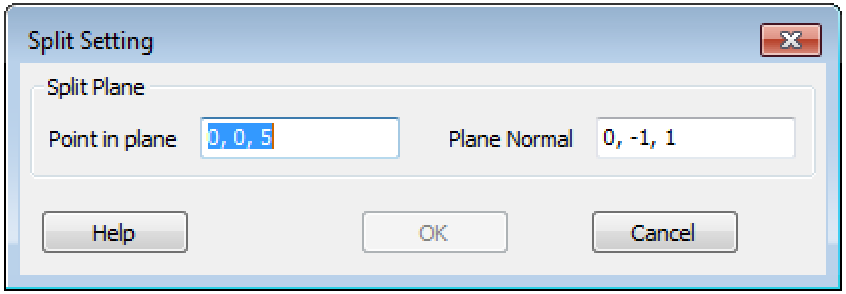
1. Define a cylinder as the background



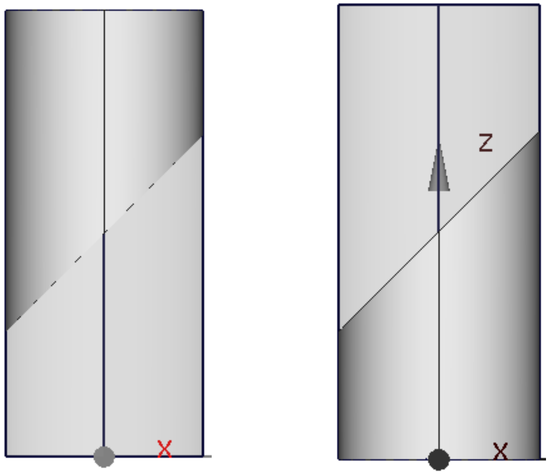
2. Select the cylinder and click “Split Selected Solid”



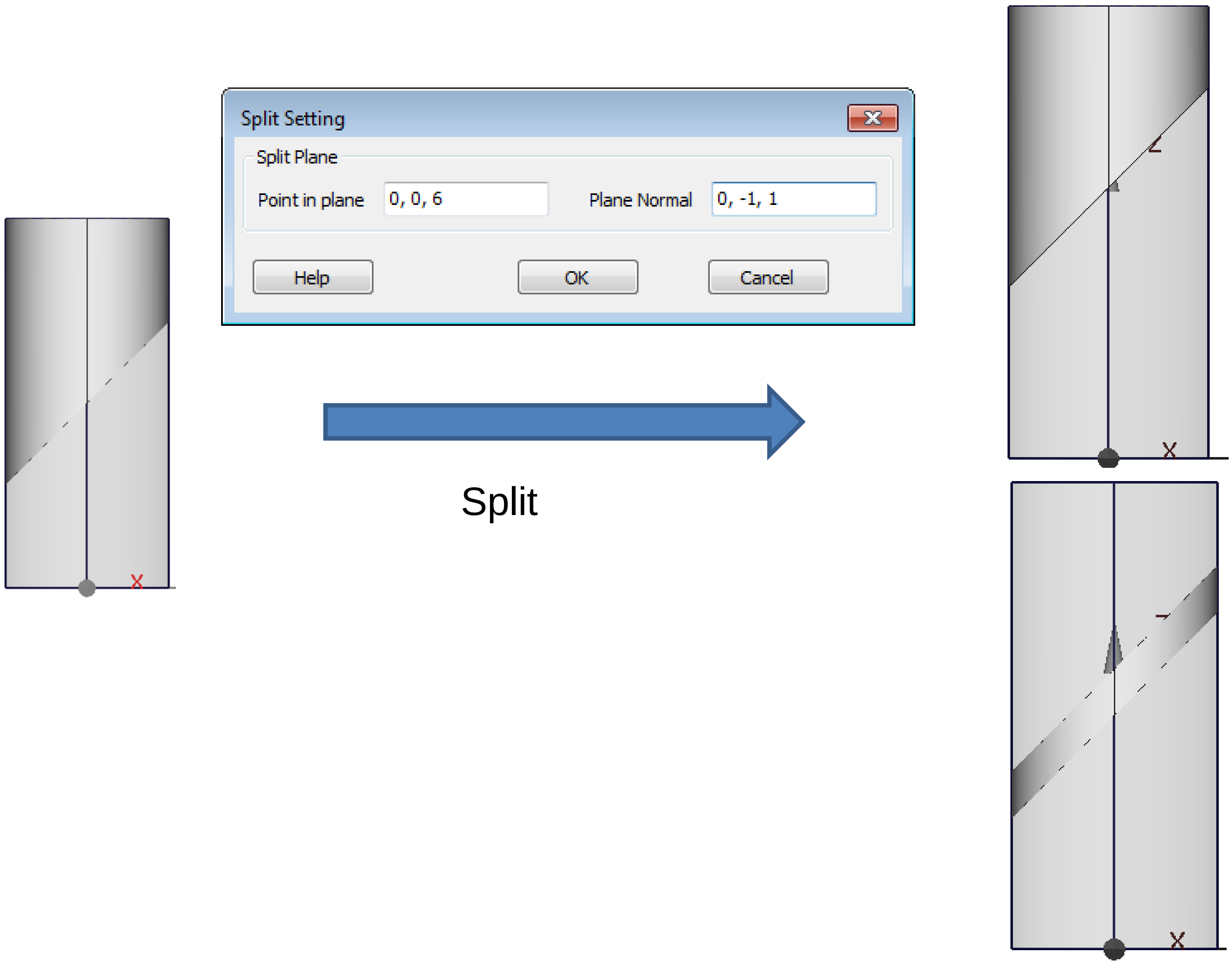
Define the split plane, as following



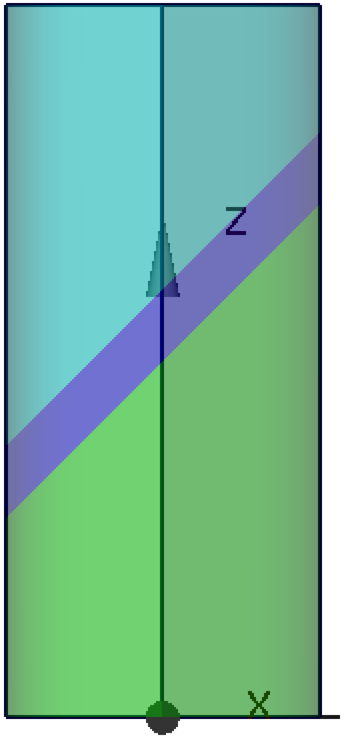
The cylinder will be split into 2 parts as following



Split this part again by following plane



Assign different material for each object, we can get the tilted layered background as following

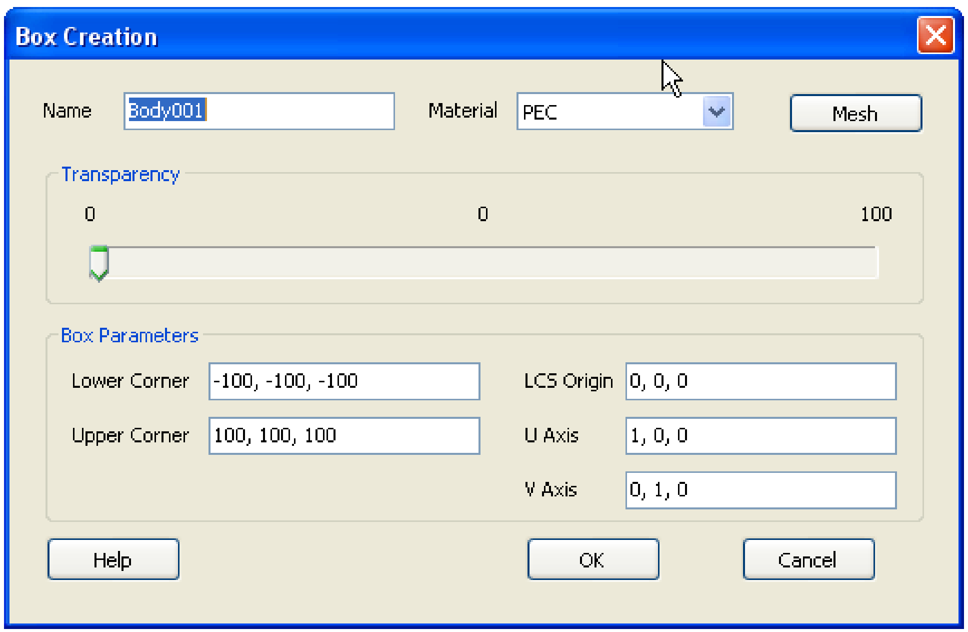


Solid with Parts

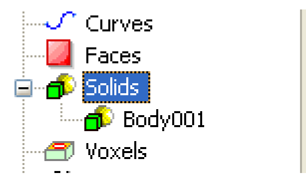
Wavenology GUI supports a solid with multiple parts without external BOOLEAN operations.

For example, we would like to create a solid with a box stacking with a half sphere.

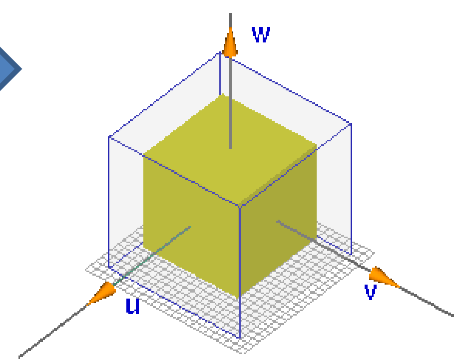
1. Create a box first



box parameters

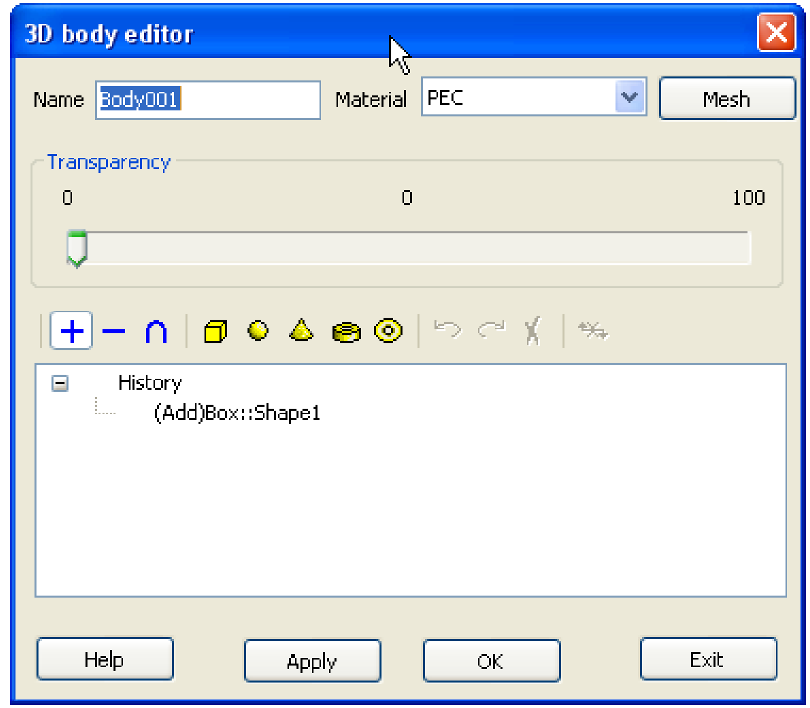


box listed in tree
the box has name "Body001"

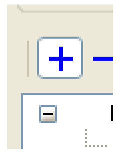


Box shown in the 3D canvas

2. Double click treenode “Body001” to enter solid editor dialog

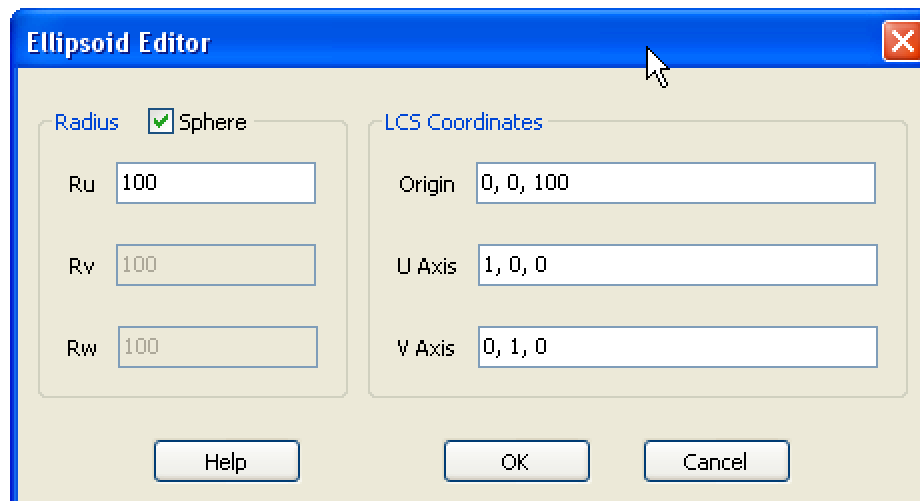


3. Check the “Add” button

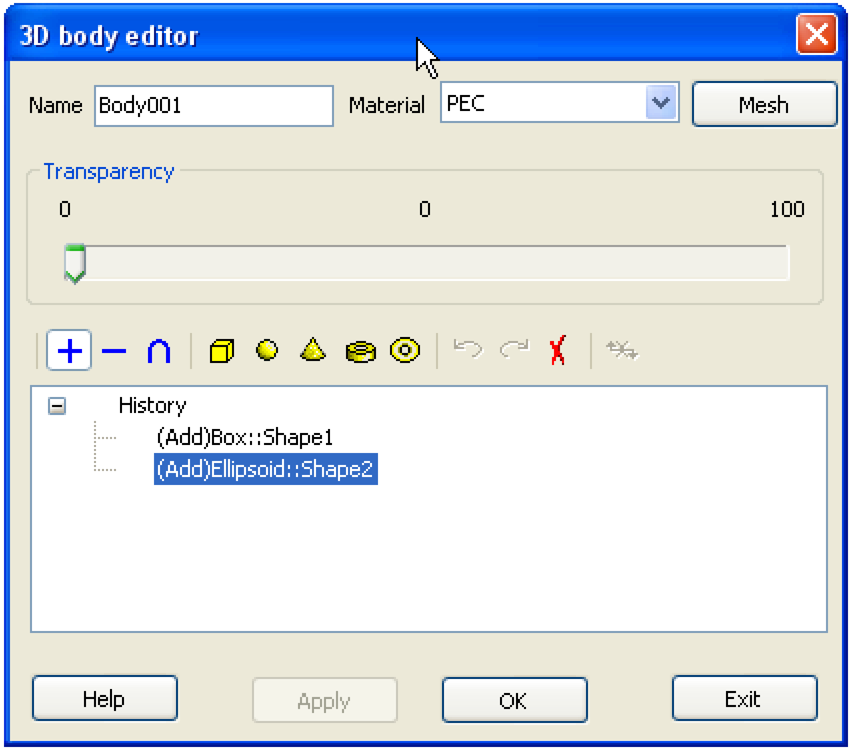


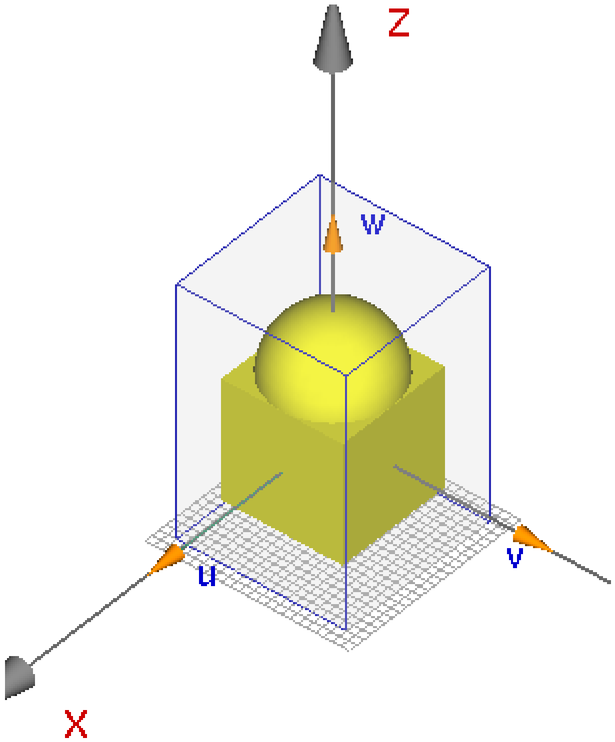
4. Press “sphere” button

to add a sphere on the original box. Here, we make the sphere radius 100 with the center at (0,0,100)

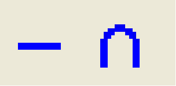


Following is the final structure





Similarly, user can SUBTRACT or INTERSECTION other shape on an existing 3D solid by “subtract” and “intersect” functions



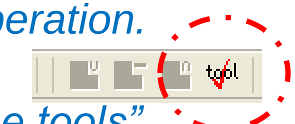
Boolean Operations on 3D Solids

Sometimes, there is clash among several solids. For example, the two solids in following case have space conflict as shown in the red dashed-wire-frame.

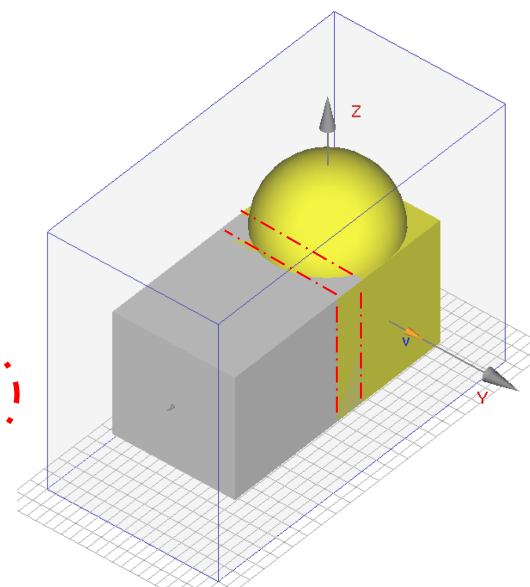
Before mesh generation, user needs to determine which solid occupy the clashed region. Otherwise, the mesh could be wrong.

In this example, we decide to let the gray box “Body002” occupy the clashed region. We use Boolean operation to implement it.

*Note: there must be at least two solids selected to make a Boolean operation. We define the first selected solid as “**Blank**”, the other solids as “**Tools**”. There are two options for a Boolean operation: whether delete the **Tools** after operation.*
The options are set by toolbar button:

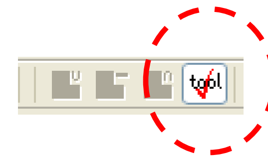


The default setting (unchecked) is “delete the tools”.
If this button is checked, means “keep the tools”



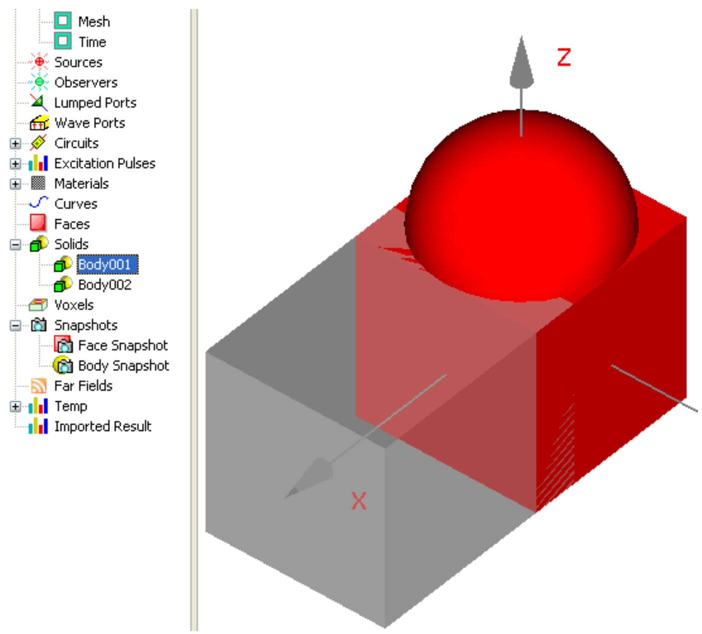
We show two methods to implement a Boolean operation. One is using treenodes to select the target solids, another is using canvas pickup to select the target solids.

Both methods use the Boolean operation **subtraction**: *Body001*= *Body001* **-** *Body002*

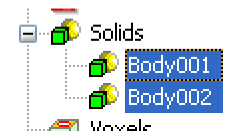
In these two methods, we will keep two solids. Therefore, make sure button  is checked before the operation.

Method 1: operation through treenodes

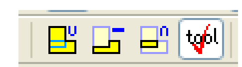
Expand treenode "Solids" to show all solids



Select treenode "Body001" as **Blank**



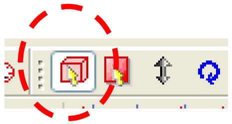
Press keyboard "Ctrl" and select "Body002" as **Tool**



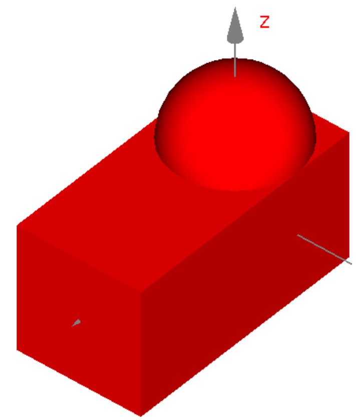
Now, the Boolean operation toolbar buttons become enabled

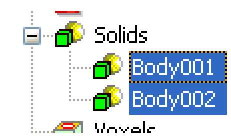
Press "subtract" button. The Boolean operation is done.

Method 2: operation through canvas pickup

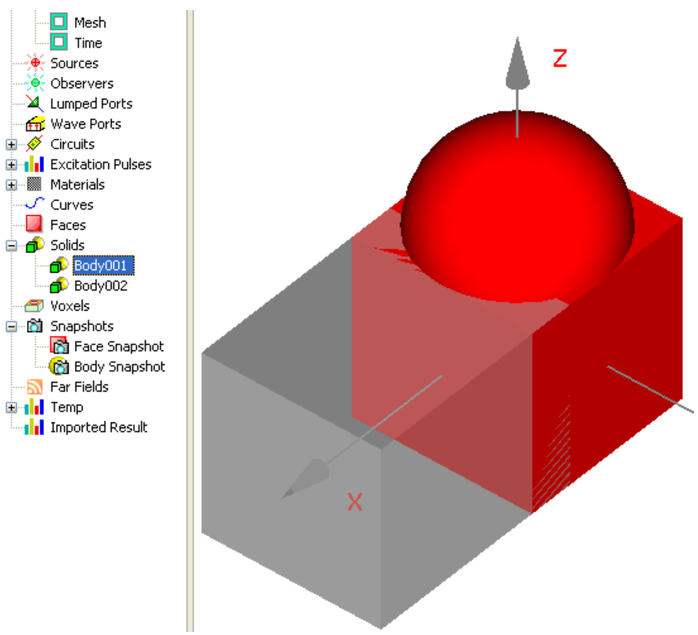


Make sure the display canvas enter the “select body” mode by “Check” this button.

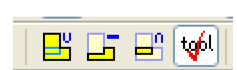




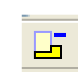
Press keyboard “Ctrl” and use mouse to select “Body002” as **Tool**.
Meanwhile, you can find the “Body002” treenode also selected.



Move mouse arrow on “Body001” and then left click mouse to select it as **Blank**.
Meanwhile, you can find the “Body001” treenode also selected.

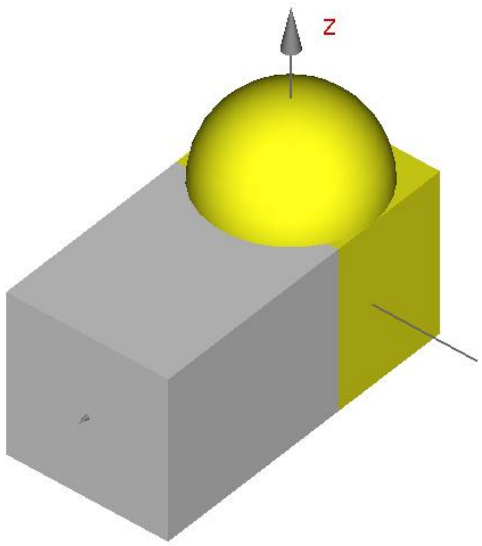


Now, the Boolean operation toolbar buttons become enabled



Press “subtract” button. The Boolean operation will be done.

The operation result is shown in following figure



Note: if the operation result is not as user expects, normally it means the operation setting is wrong. User can use “**Undo**” function to recover the system prior to the Boolean operation

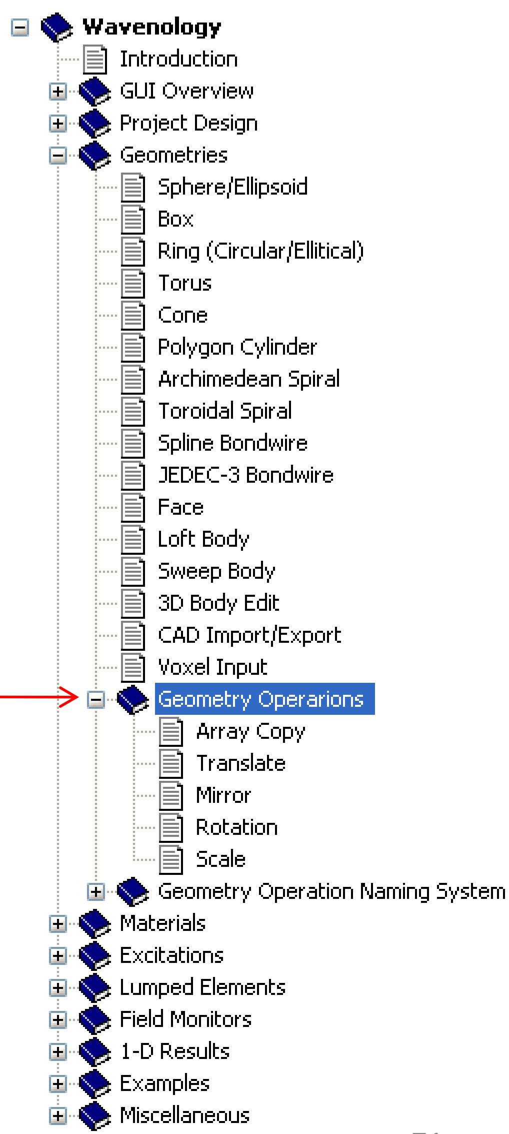
Redo & Undo

Mirror, Translation, Rotation, Scale & Array Copy

Wavenology GUI supports following transform operations on all 3D solids:

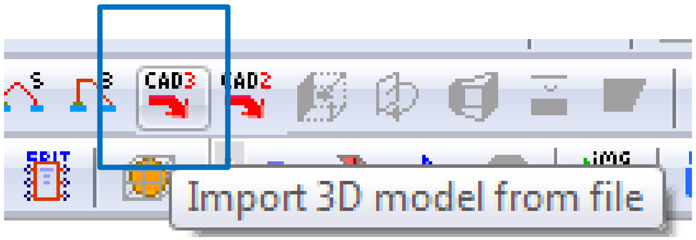
Mirror
Translation
Rotation
Scale
Array Copy

The detail operation please refer to the "**Geometry Operations**" section in the manual of Wavenology GUI.



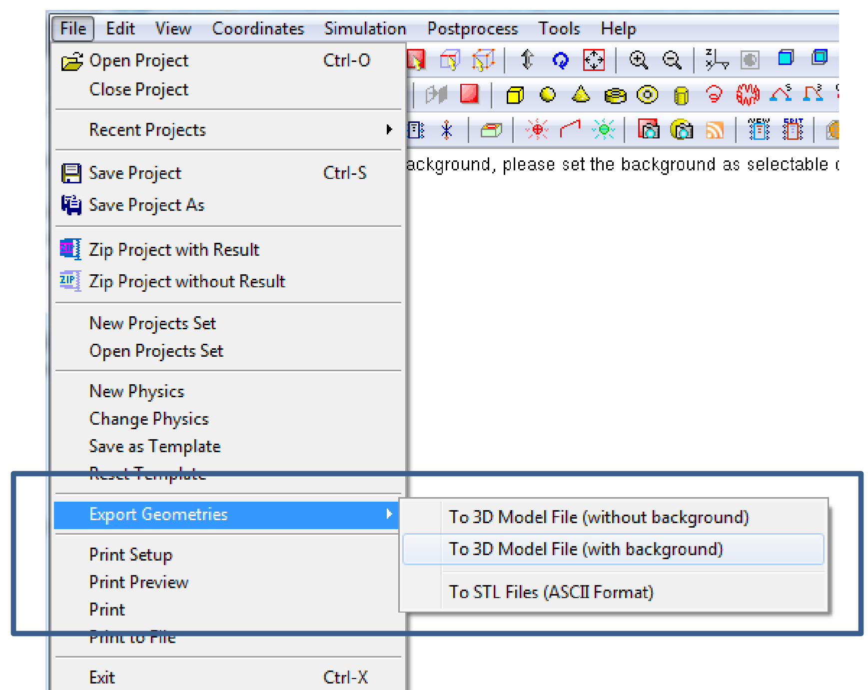
Import CAD models from Data File

- Wavenology GUI can import 3D models generated from other CAD software by following formats
  - STEP
  - SAT
  - STL
  - IGES
  - OpenCascade BREP



Export Geometries into Data File

- Wavenology GUI can export the geometries in the project to file for data exchange purpose, including following formats
  - STEP
  - SAT
  - STL
  - IGES
  - OpenCascade BREP



Special Treatment for Importing SAT Models for Elastic Wave Project

For an elastic wave project, there is a special treatment to let user can import 3D models from SAT file with material name and profiles.

1. The "ProductId" part of the SAT file must be the string "ElasticWave".
2. There must be two accompany files to define the material usage information and material profile. For example, the SAT file is A.sat, there are 5 models in the file.
   2.1 User must define a material usage file as name A. SatMat. The file is an ASCII text format. Each line lists the material name for a model. For example:

   water
   steel s1
   …

   2.2 User must define a material profile file as name A. MatDef. The file is a ASCII text format. Each line lists the profile for a material. The name, velocities are separated by ','. For example,

   water, 1000,1500, 0
   steel s1, 7890, 5790, 3235

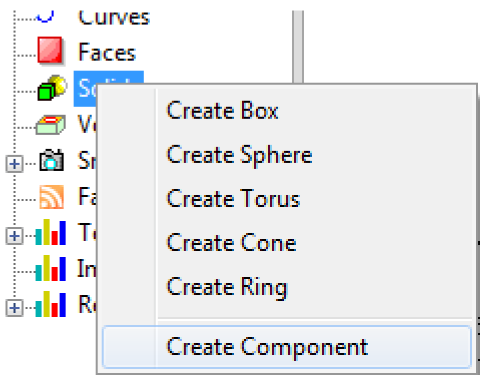
   The first column is the material name; 2nd is the mass density (kg/m3); 3rd is the P speed and 4^{th} is the S speed (m/s).

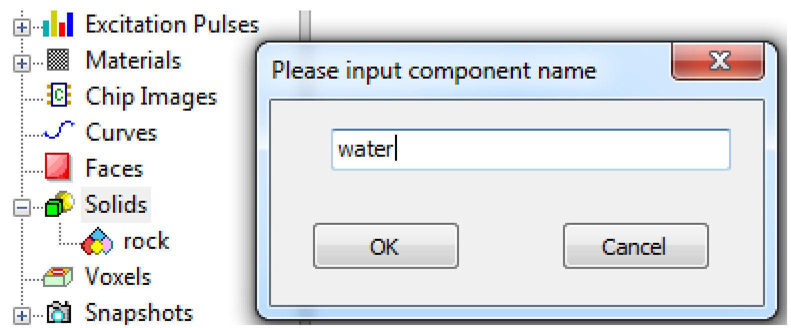
Organize Solids into different Components

In some cases, there are lots of solids in a project, it will be difficult to distinguish a solid from a single object list.

WCT provide a component system to organize solids.

As following procedure, components “Rock” & “Water” are created under Solids treenode. Sub-component can be created under a component also.

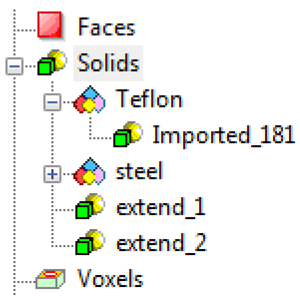


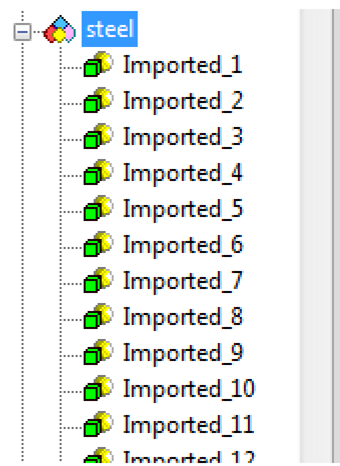


After a component is created, user can

- ➢ create solids in this component
- ➢ move solids between components

Following is an example, there are more than180 solids in the project, part of them are imported from other CAD software, part of them are built by WCT GUI. So, we put the imported solids into 2 components by the material types, and all WCT solids are placed under the root.



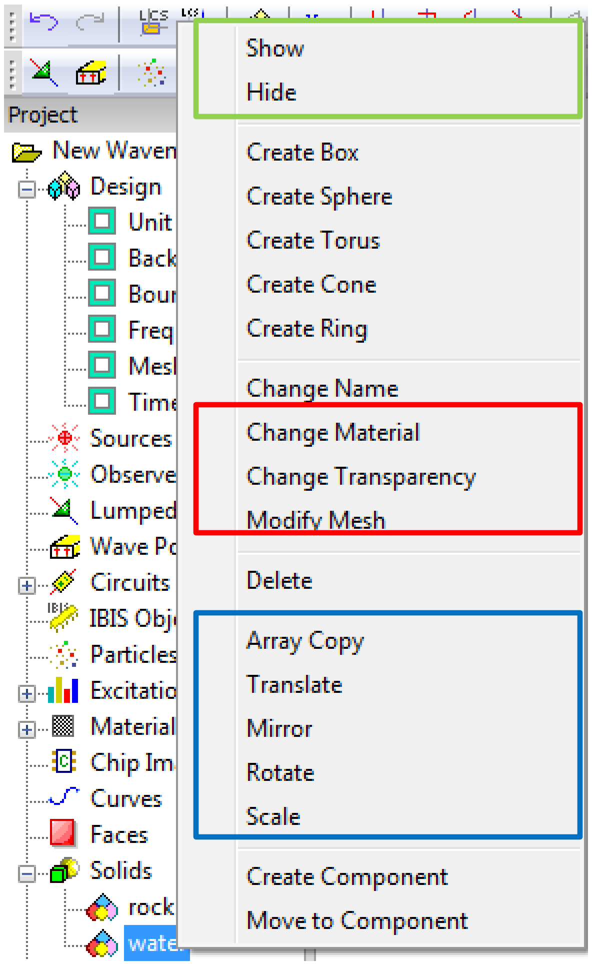


With the component system, user can

➢ Show/hide all solids in a component with one action

➢change the property of all solids in a component with one action
- Material
- Transparency
- Mesh setting

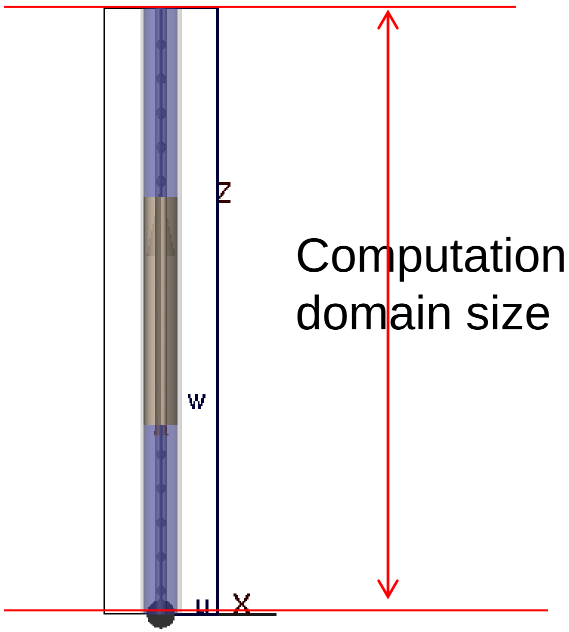
➢ transform all solids in a component with one action
- Array copy
- Translate
- Mirror
- Rotate
- Scale



A Special Setting to allow Geometry larger than the Computation Domain

In principle, all geometries in a project should be inside the computation domain to satisfy the boundary conditions.

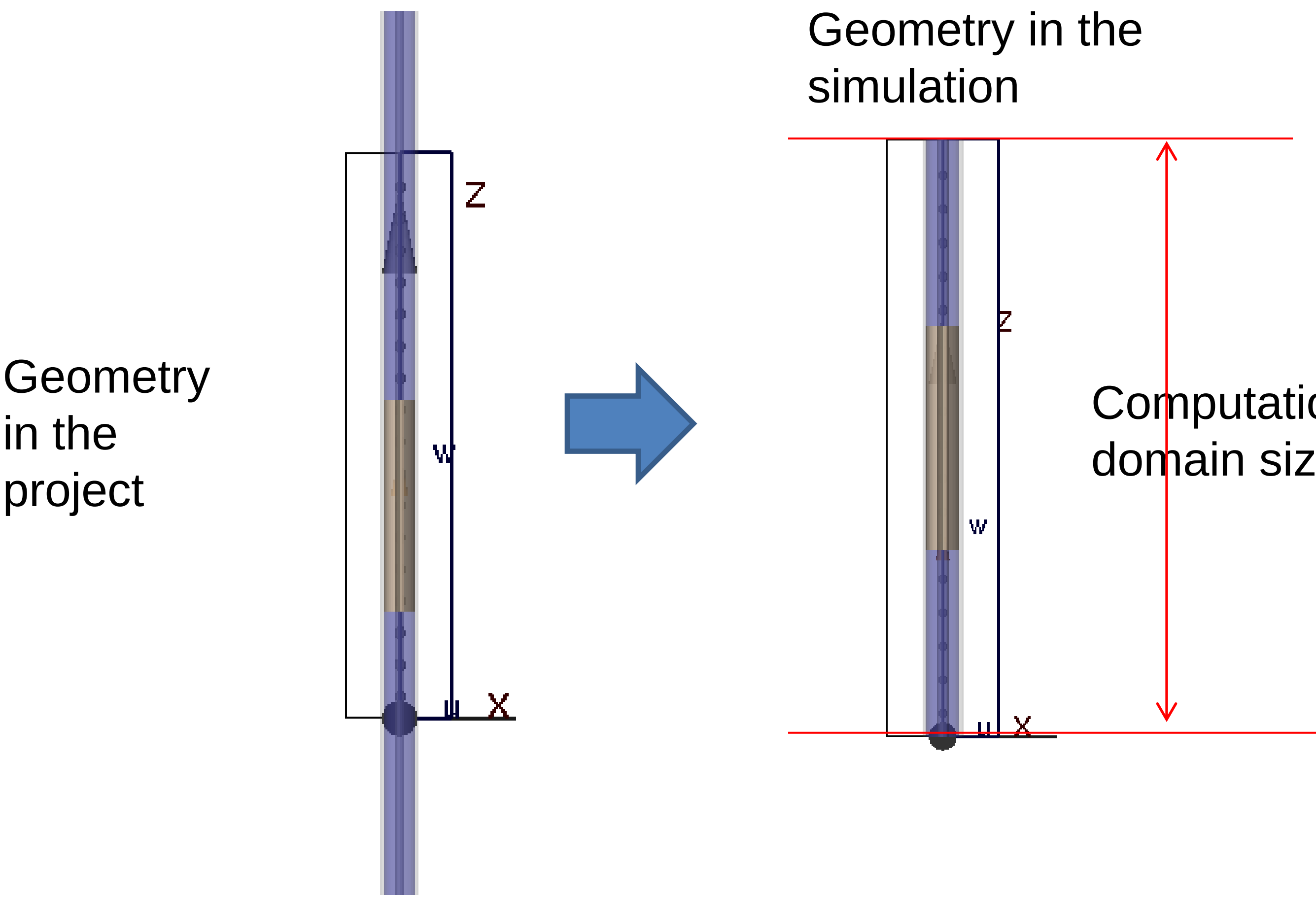
As the default setting, before a simulation, WCT GUI will check to make sure that all geometries are in the domain.



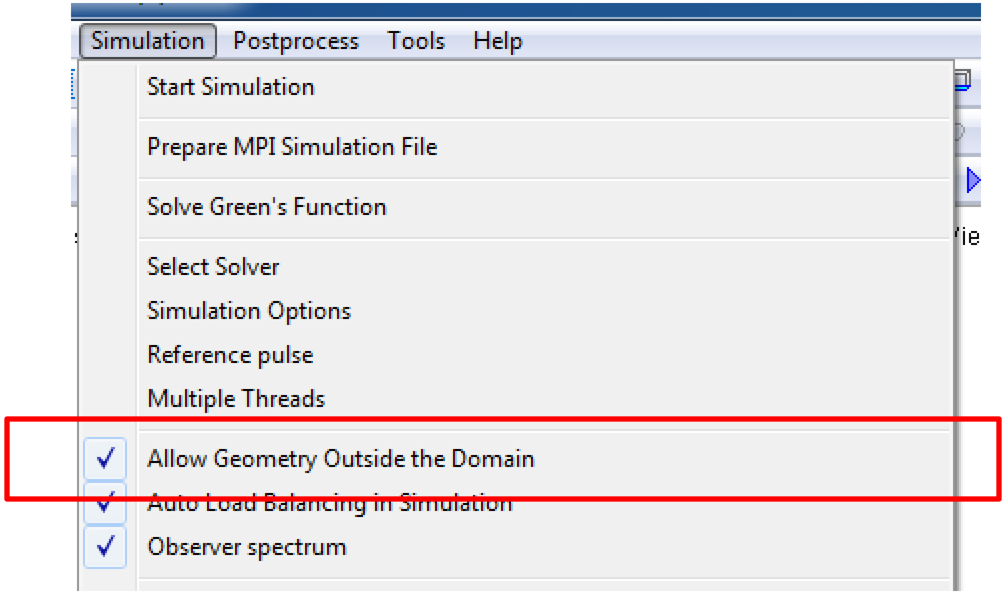
A regular project

In the WCT simulation engine, the geometry out of domain will be cut off automatically.

Except the OPEN boundary conditions, the simulation result will be always correct with the cut operation.



From v2.2.2, WCT GUI adds an option to allow the geometry larger than the computation domain, as following



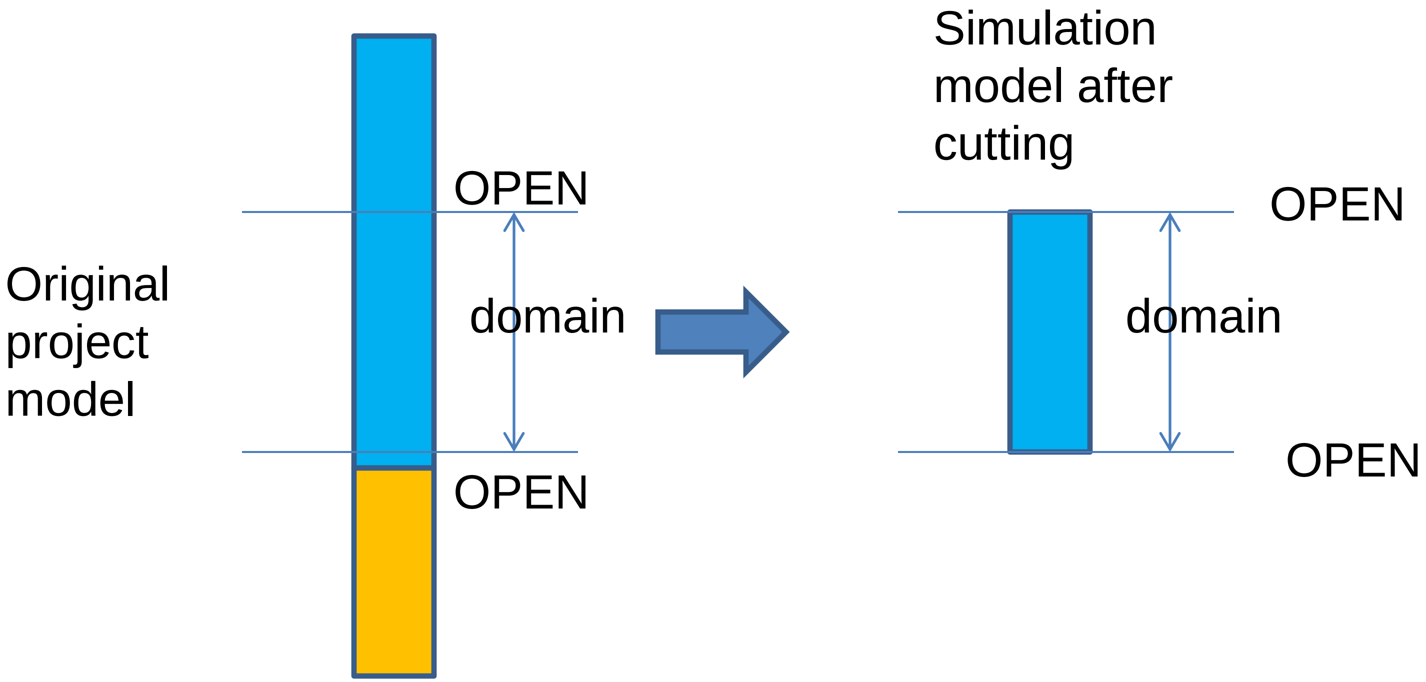
The **default setting** of this option is ***OFF*** (unchecked), which means that all geometries must be in the computation domain. This is the same behavior as the old version.

If this option is set as ***ON*** (checked), which means that the geometries can be larger than the computation domain.

With this option, the parametric sweeping will be more easy to use. We will show it in the section of the ***Parametric Sweeping***.

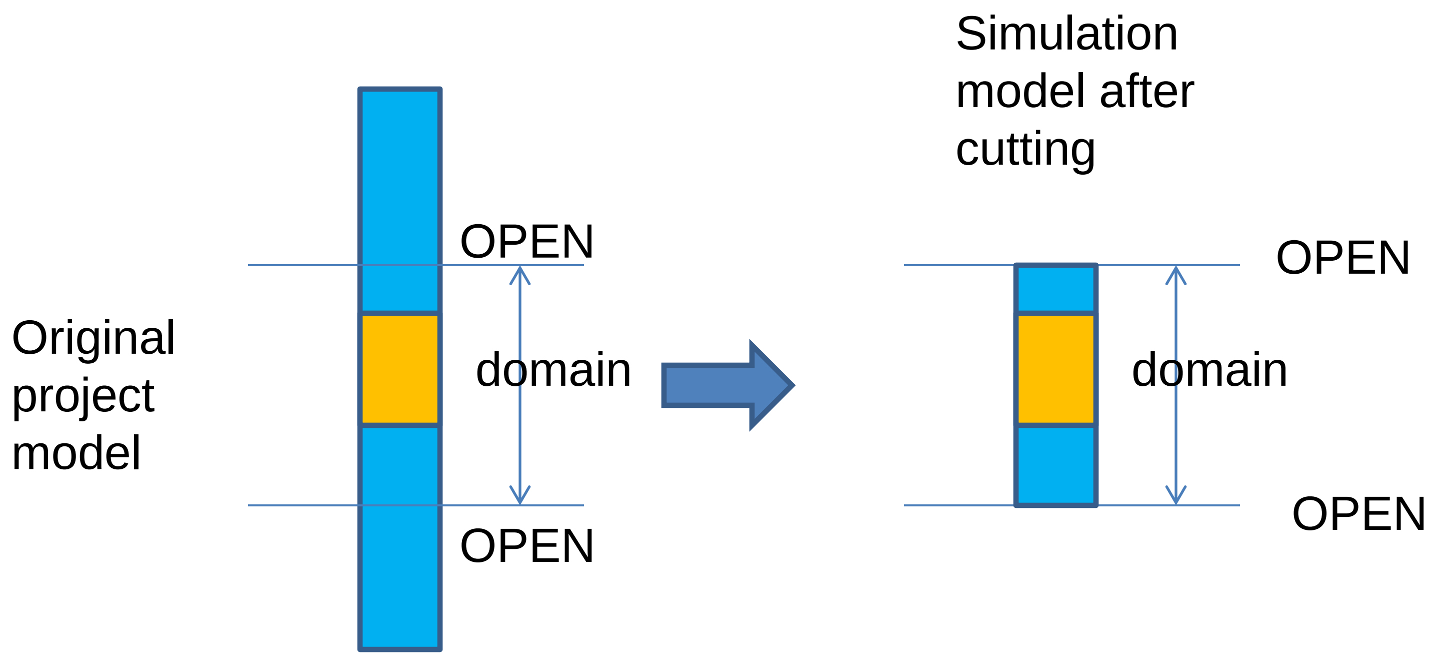
If this option is ***ON***, for the project with **OPEN** boundary, user need to make sure whether the “cut off geometry” operation will generate different simulation results.

For example, for following project,



The model in the simulation will have a different simulation result compared to the original model for most situations. Except the case that the wave can’t reach the boundary.

On the contrary, for following project,



The model in the simulation will have a correct simulation result compared to the original model.

So, if this option is ***ON***, WCT GUI does not know whether the simulation result is correct or not, it is user responsibility to make sure whether the simulation is acceptable or not. Therefore, user need to know the project very clear before enable this option.

Variable System & Usage

Variable System

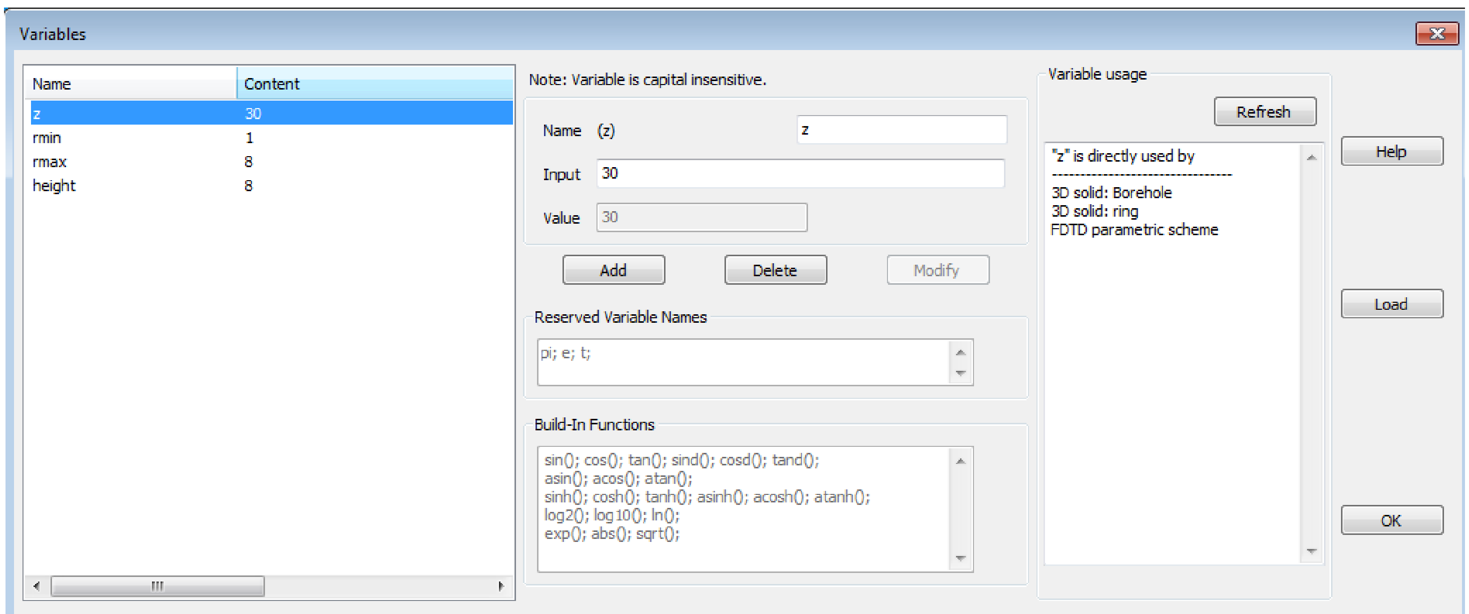
Wavenology GUI embeds a variable system.

Most project parameters can be defined through variable.

- Project boundary position
- Solid dimension parameters
- Source position
- Receiver position

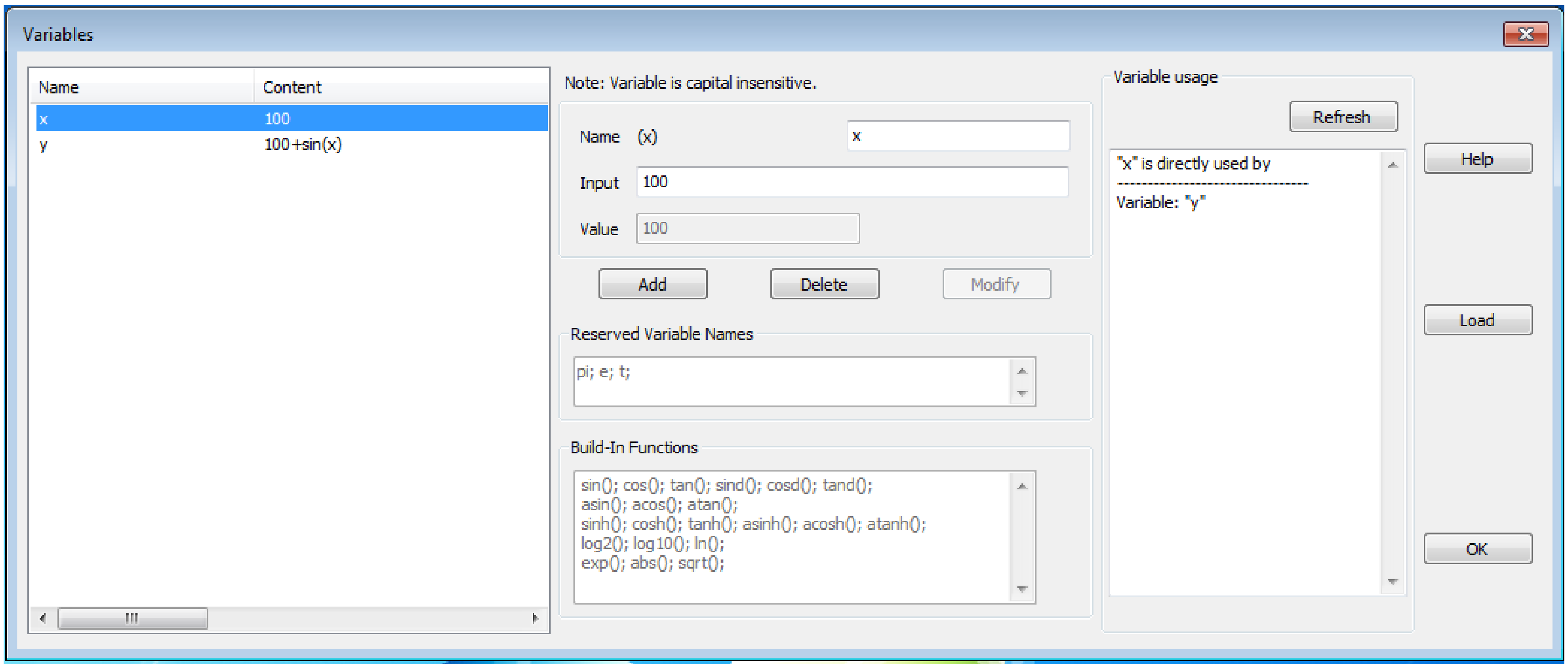
Variable Editor

Through toolbar button “Var”

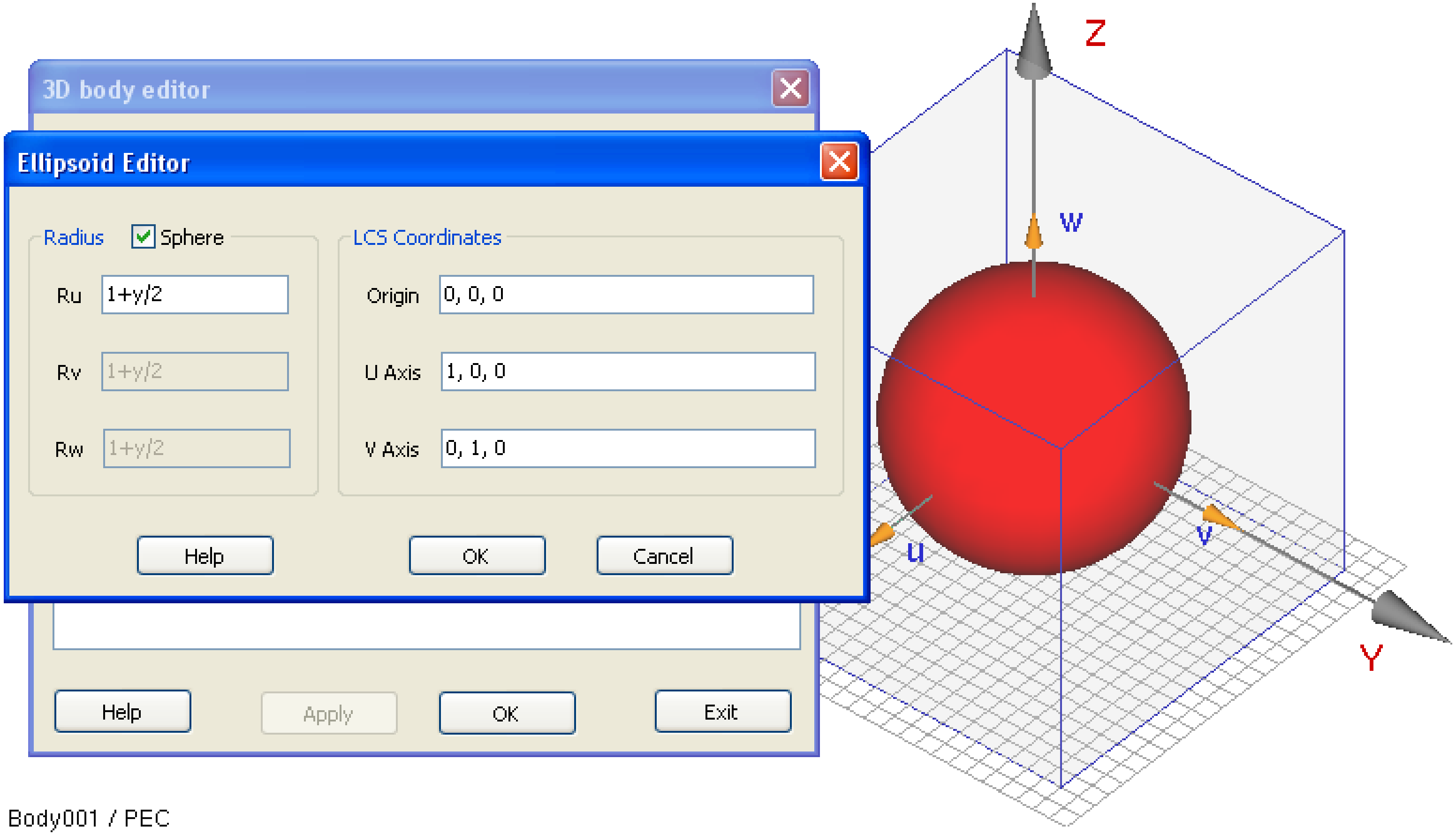


All 3D geometries in Wavenology GUI can be defined by variables.
For example, user can define a sphere with radius as variable “x”. The sphere’s shape can change as the value of “x” changes.

In following example, we define variable “***x***” as value *100* and variable “***y***” equal to “*100 + sin(x)*”



Then, we can define a sphere with radius as “1+y/2”



Body001 / PEC

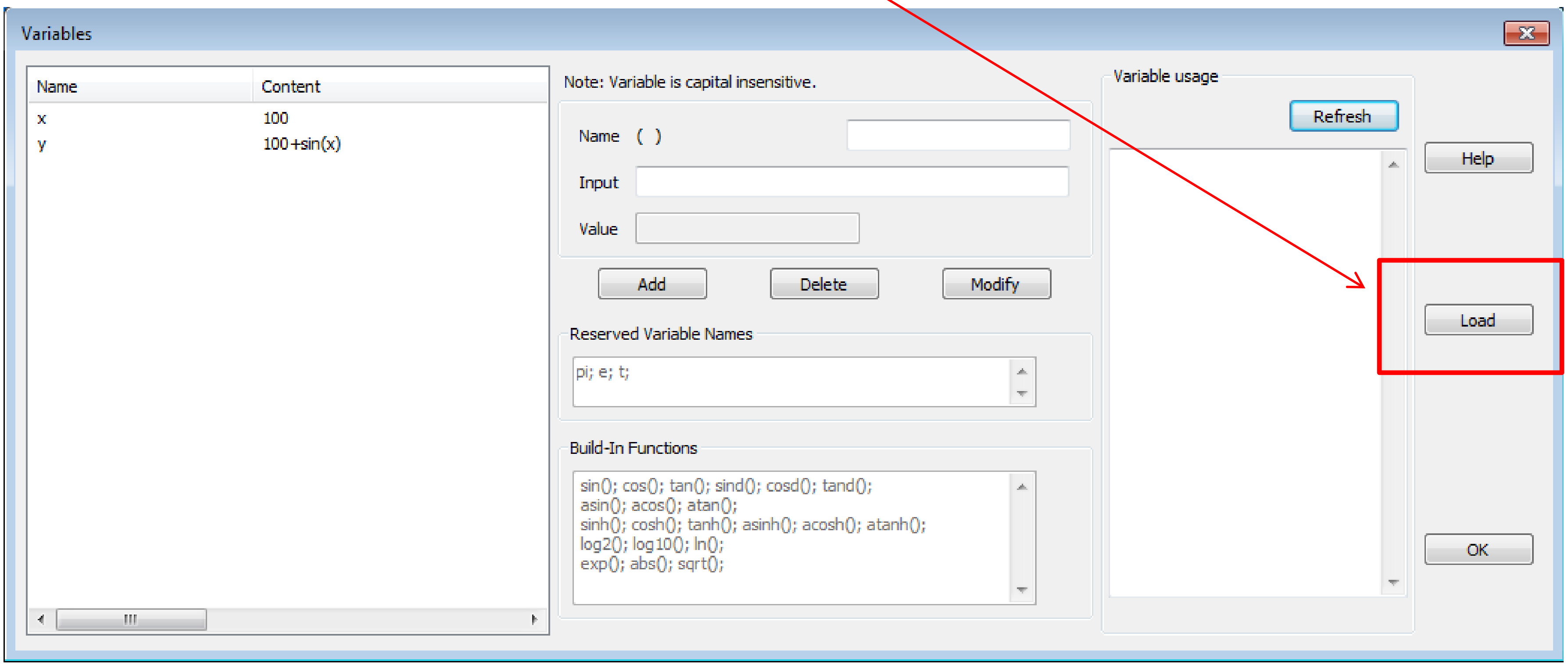
bounding box: (-50.7468,-50.7468,-50.7468),(50.7468,50.7468,50.7468)

size: (101.494 x 101.494 x 101.494) , center: (0,0,0)

PEC

Load Variables From Data File

User can load variables through data file



Variable File Format

- ASCII format
- Each line is the definition for one variable, format as
  - “variable_name”=“variable_definition”
- Example of file

```
“a”=“1.33”
“b”=“cos(a)”
“c”=“a+b”
```

var.txt

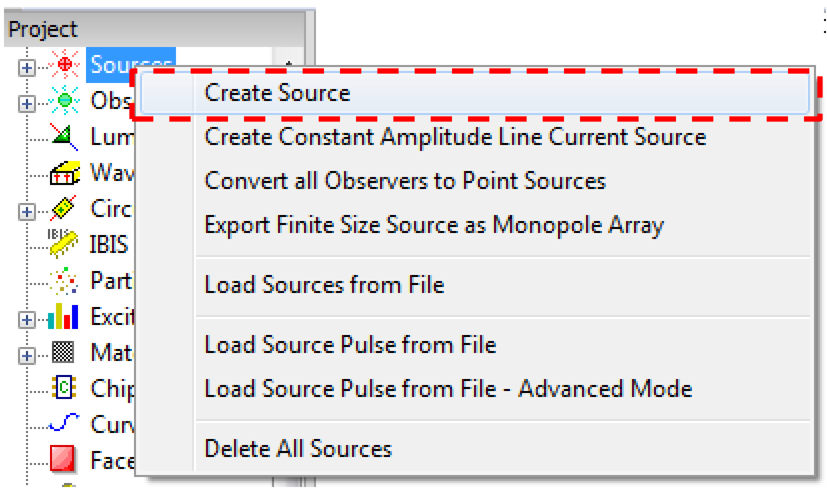
Note:

➢ if the name of a variable in the file conflict with an existing variable in the project, the loaded variable will replace the existing one, a warning information will be shown in the log

➢ The name of a variable can’t include any SPACE.

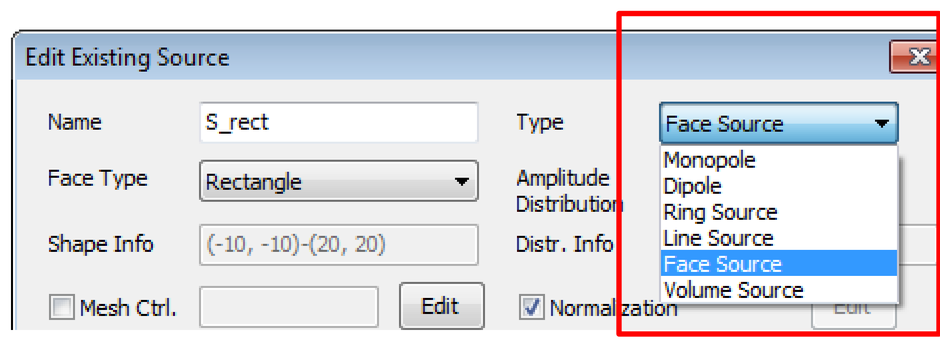
Sources & Excitation

Source & Excitation Pulse



There are 6 types of excitation sources for BHA simulations:

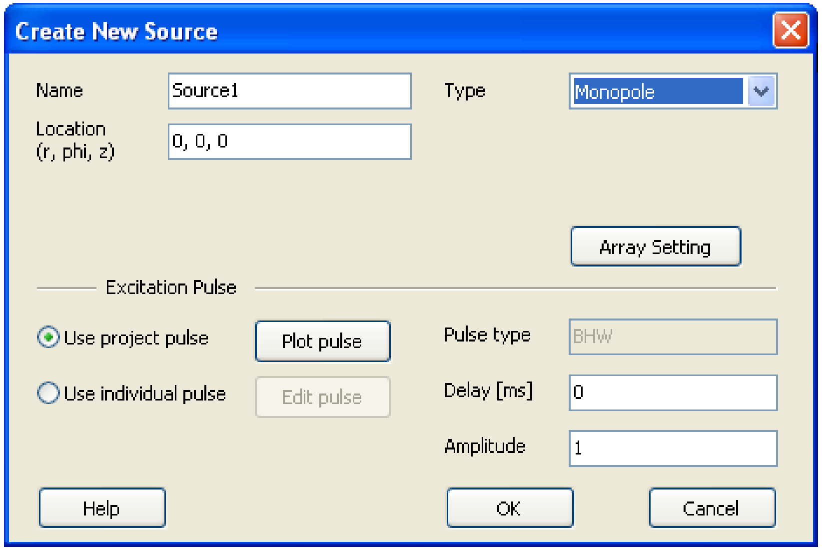
- **Ring source**
- **Point dipole source**
- **Point monopole source**
- **Curve type source**
- **Face type source**
- **Volume type source**



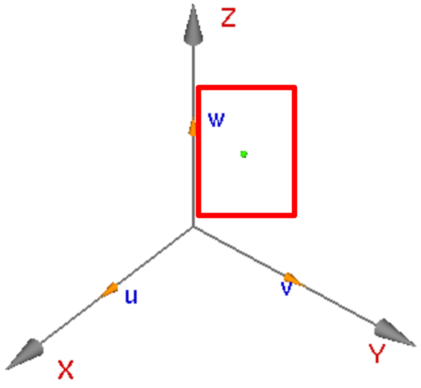
Point Monopole Source

A point monopole source will excite spherically at source position.

The parameter is the **position of the source** : <r, ϕ^o, z>**.**



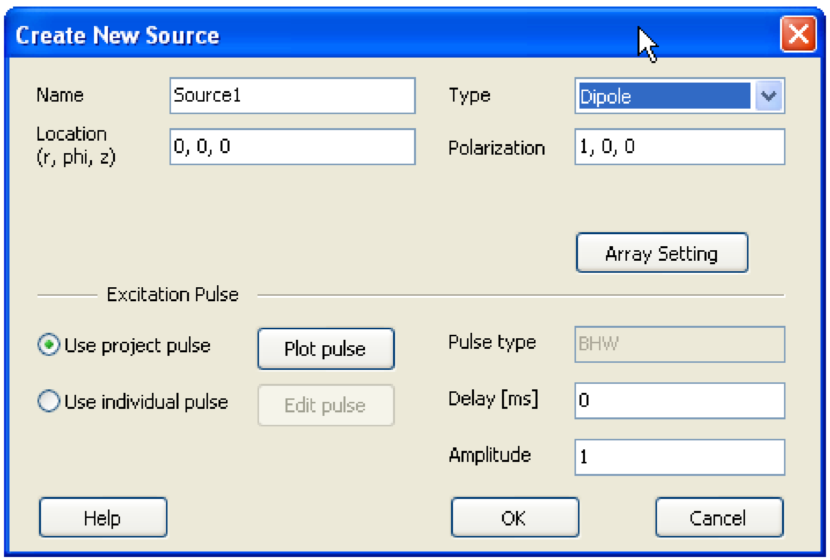
It will be shown as a small point the project canvas



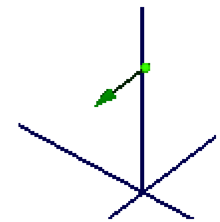
Point Dipole Source

A point dipole source will excite with specified polarization at the source position.

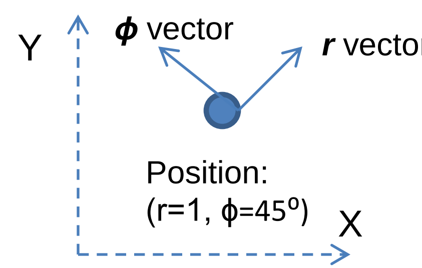
The parameters include:
1) the position: <r, ϕ^o, z>
2) the polarization: <***r***, ***ϕ***, ***z***>



It will be shown as a point with vector in the project canvas



2D r-ϕ plane



Note: the polarization of a point dipole is <***r, ϕ, z***>, and the corresponding polarization in the Cartesian coordinates system need to couple to the source position. For example, a point dipole source at position <r=1, ϕ =45^0, z=0>,

a) With a polarization as <***r***=1, ***ϕ*** =0, ***z***=0>, the corresponding polarization in X-Y-Z system is <1, 1, 0>
b) With a polarization as <***r***=1, ***ϕ*** =-1, ***z***=0>, the corresponding polarization in X-Y-Z system is <1, 0, 0>

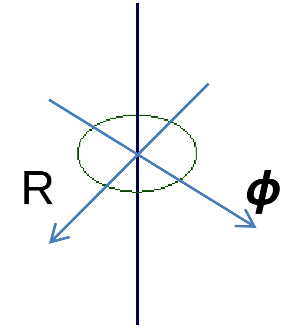
Ring Source (1)

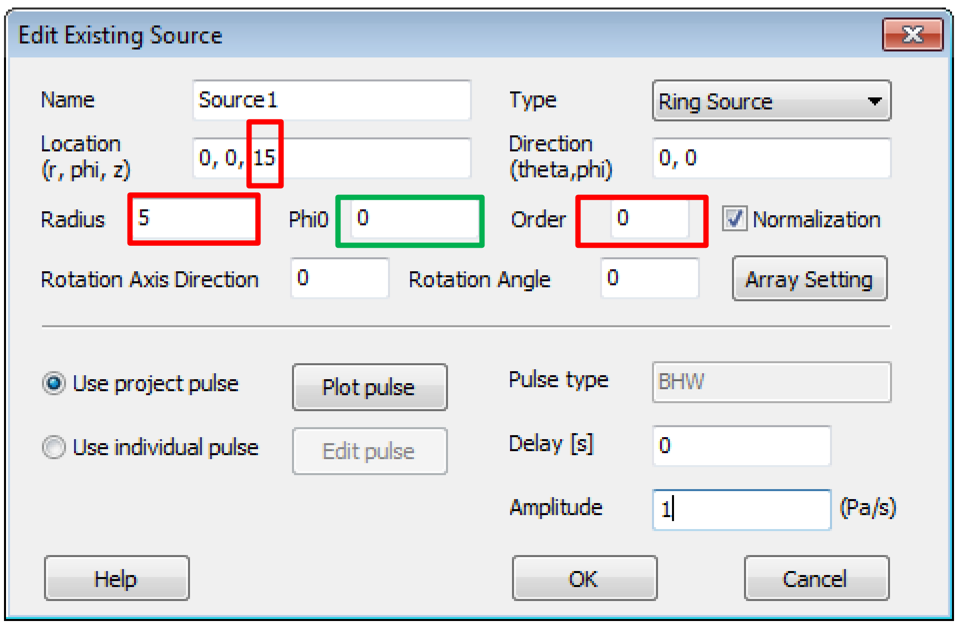
A ring source will excite on a circle with a specified distribution.

The basic parameters include:

1) the ring's z position in axis *Z*
2) the ring *radius*
3) The *order* of distribution

It will be shown as a circle in the project canvas (in R-ϕ plane)





If we define the **order** of a ring source as ***n***, the ring source will have a magnitude distribution in a circle as: $\boldsymbol{\cos(n^* \phi + \phi_0)}$.

So,

1) n=0 : the magnitude distribution will be a constant in the circle, this is a ring monopole source. In this case, ϕ_0 will be ignored;
2) n=1 : the magnitude distribution will be $\cos(\phi + \phi_0)$ in the circle, this is a ring dipole source.
3) n=2 : the magnitude distribution will be $\cos(2\phi + \phi_0)$ in the circle, this is a ring quadrupole source.

Ring Source (2)

A ring source can rotate in a R-ϕ plane with following controls

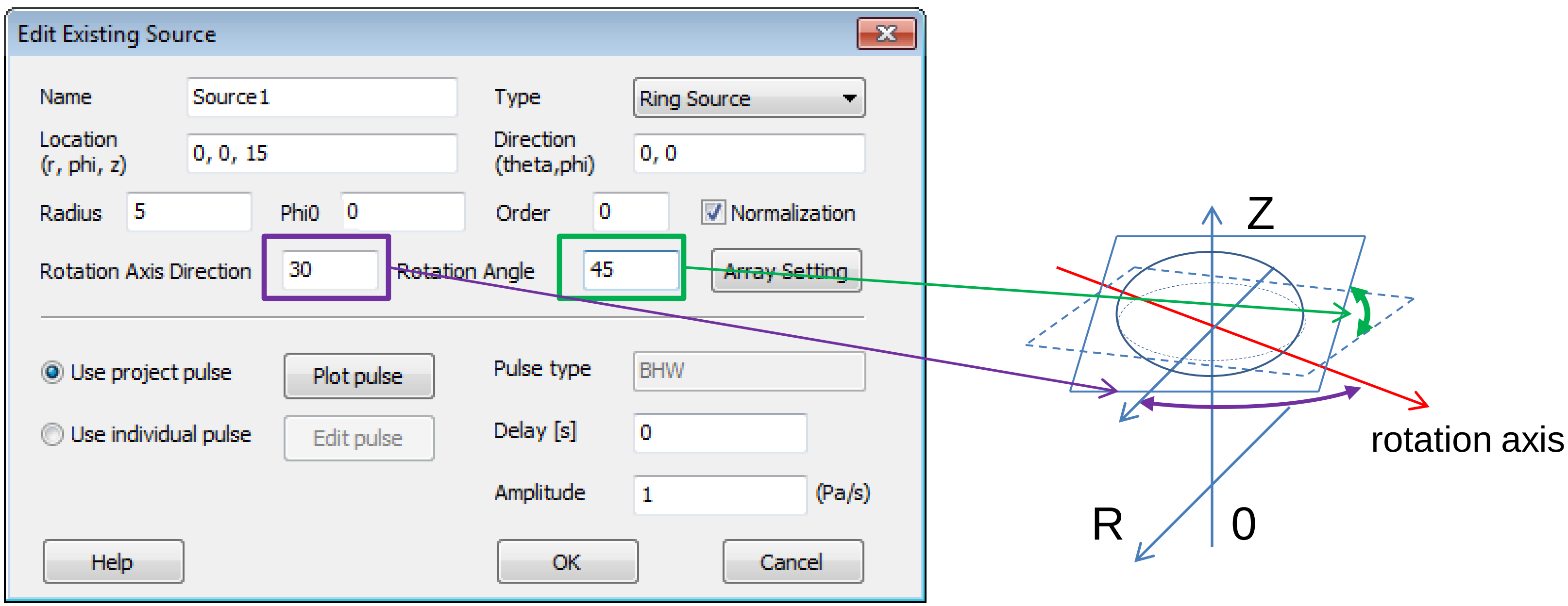
1) the direction of the ring rotation axis
2) the rotation angle

(Unit: degree)

These control will apply following rotation on the ring

➢ the rotation is in a R-ϕ plane at Z position

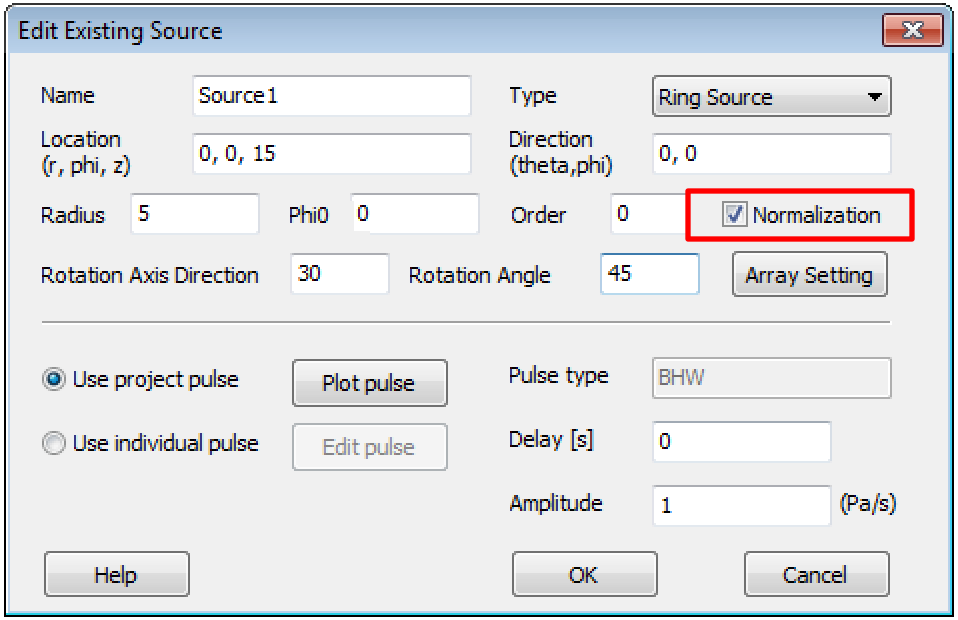
➢ the rotation axis is defined by the angle from +R axis



Ring Source (3)

The *amplitude normalization* of the ring source

➢ An option is added to the ring source from WCT version v2.2.6, as shown in the figure.



If it is enabled, the amplitude of the ring source will be normalized by a factor which is proportional to the length of the ring.

Before v2.2.6, this flag is always enabled.
After v2.2.6, this flag is set as the default setting.

Basic Comment for Curve, Face & Volume types of Source

- These 3 types of source require following settings
  - *Shape type and parameters* – i.e. they are finite size
  - User need to define *an amplitude distribution* in the shape
    - The default distribution type is the *Constant* distribution with the value 1
  - The excitation amplitude will be the *amplitude distribution* multiple with the source amplitude
  - These sources will be converted to array of point monopole sources, user can define a *discretization granularity*
    - If user does not define it, the GUI will use a AUTO method, which depends on the cell size of system grid
  - There is "*Normalization*" option for the source
    - It means, whether the amplitude of source will be normalized by a factor, which is proportional to the size of the shape
    - The default value is "Not"

Curve Type Sources

The source term of the curve type source has following definition

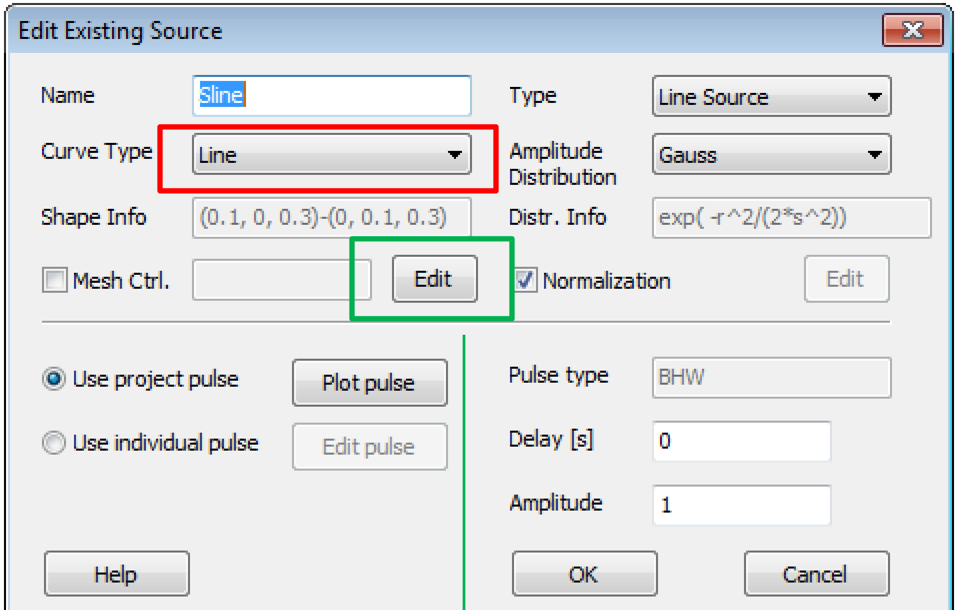
$$f(r_l) \cdot \delta_s(r-r_l)$$

Here, $r_l \in L$, L is the curve with a finite size; δ_s is the δ function in **2D** for the cross-section of curve at position l, $f(r_l)$ is the source amplitude distribution in the curve.

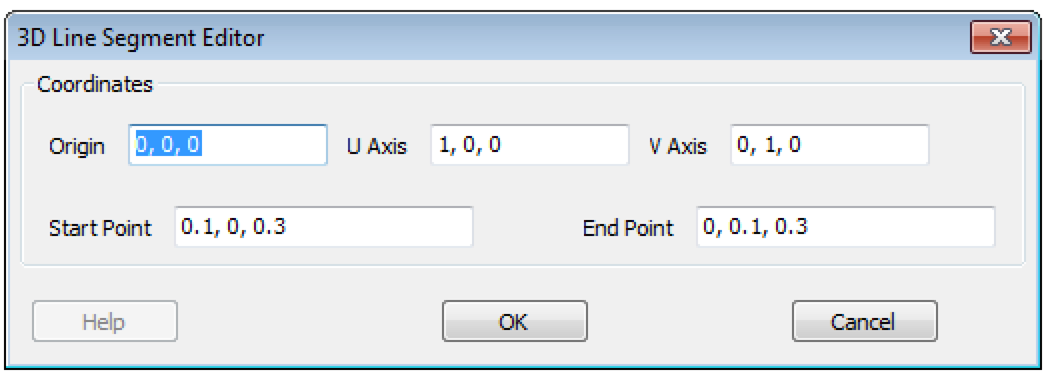
In current WCT BHA solver, it supports following types of curve source

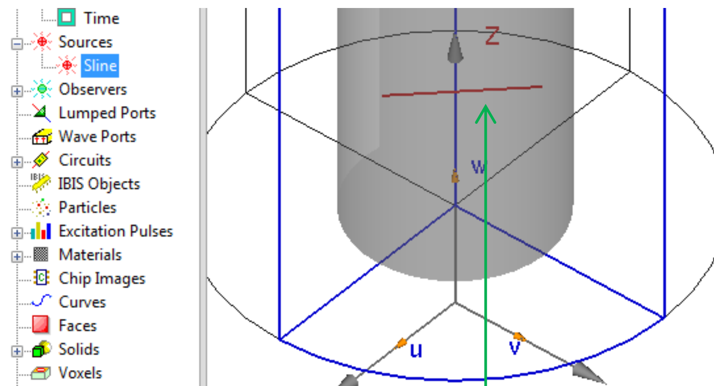
- *line source*
- *arc source*

Line Source



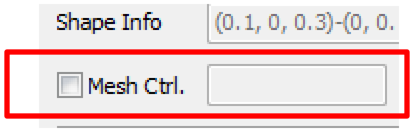
The line parameters can be edited as following



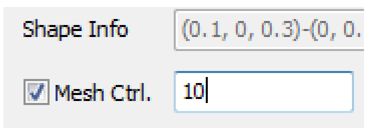


The line source shown in GUI

The default setting of *discretization* is *AUTO*

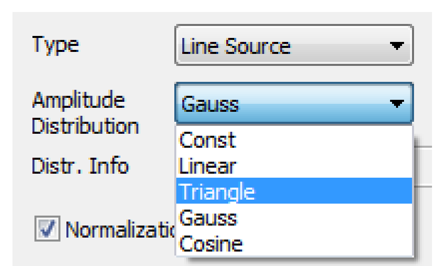


User can define a customized *discretization granularity* for this source. Here, “10” means that, this source will be converted to 10 point monopole sources in the engine.



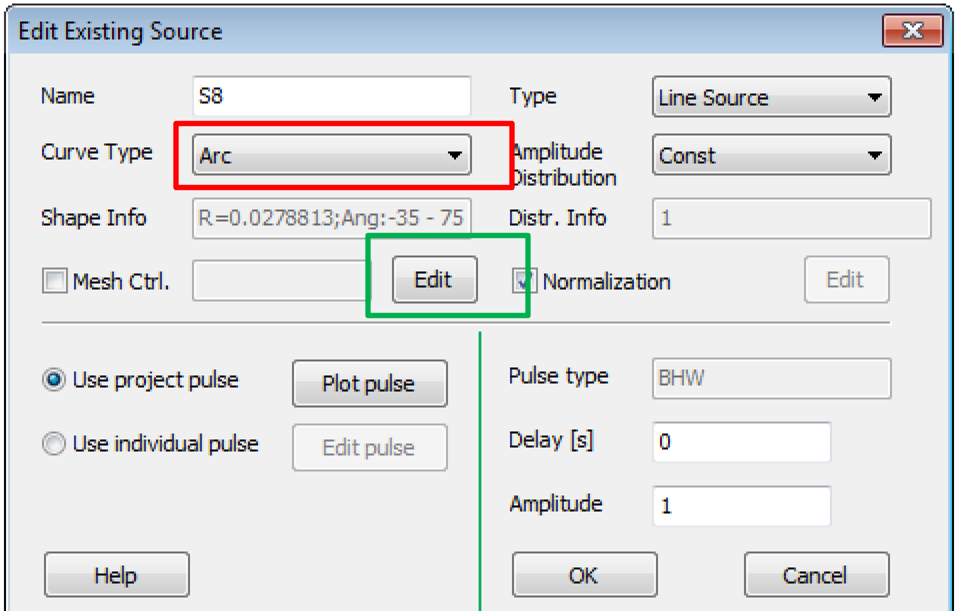
For the line source, it supports following amplitude distributions

- Constant
- Linear
- Triangle
- Gauss
- Cosine

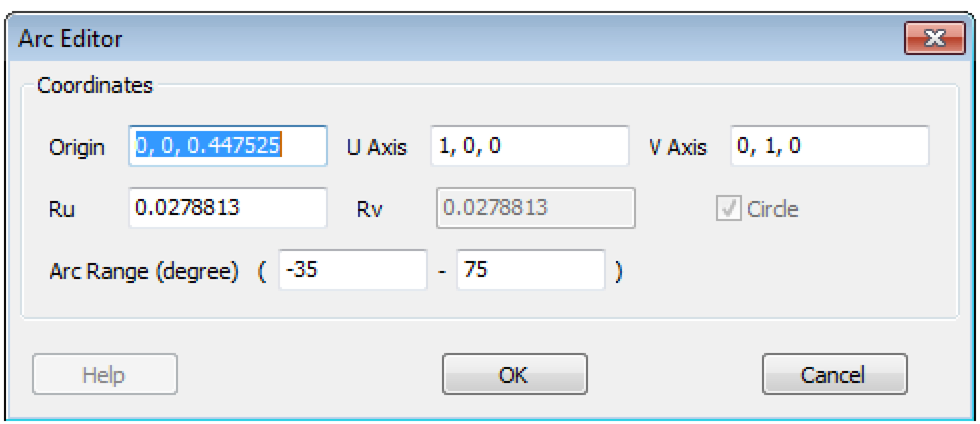


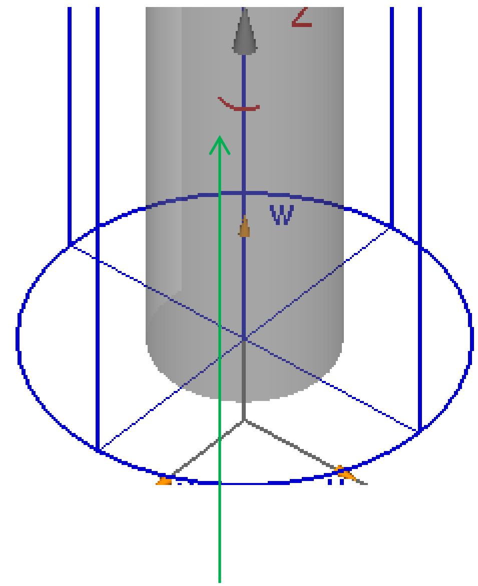
For more details for the definition of these amplitude distributions, please check section - *The amplitude distribution for the curve, face, volume types source*

Arc Source



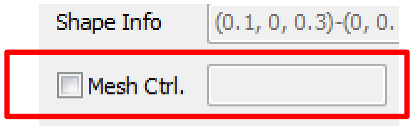
The arc parameters can be edited as following



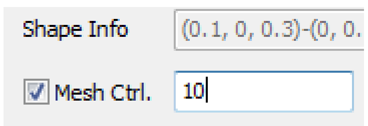


The arc source shown in GUI

The default setting of *discretization* is *AUTO*

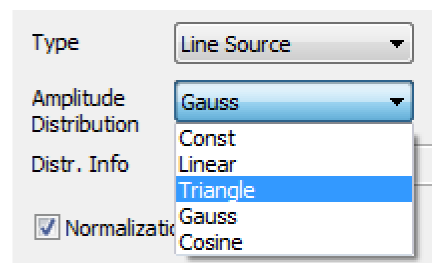


User can define a customized *discretization granularity* for this source. Here, “10” means that, this source will be converted to 10 point monopole sources in the engine.



For the arc source, it supports following amplitude distributions

- Constant
- Linear
- Triangle
- Gauss
- Cosine



For more details for the definition of these amplitude distributions, please check section - *The amplitude distribution for the curve, face, volume types source*

The difference between the Arc source and the Ring source

Arc source

- the Local coordinates system of the arc can be defined by user
  - therefore, user can change the arc to arbitrary location and direction
- the angle range is arbitrary
  - For example, [-35°, 75°]
- the arc source must be in the computational domain
- the normalization is based on the real length of the arc
- the amplitude distribution includes
  - constant
  - linear
  - triangle
  - gauss
  - cosine

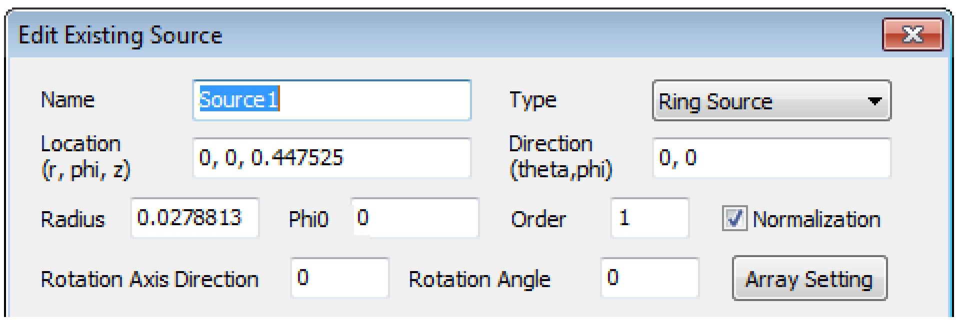
Ring source

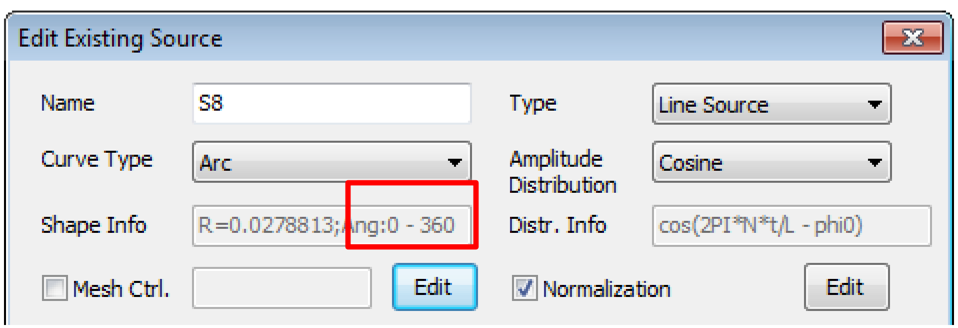
- the axis of ring is fixed: the Z axis
  - user can only change the Z position of the ring.
- the angle range is fixed: [0, 360°]
- the ring source can be out of the computational domain in the θ range
- the normalization is based on the length of the full ring
- the amplitude distribution is fixed to the cosine function only

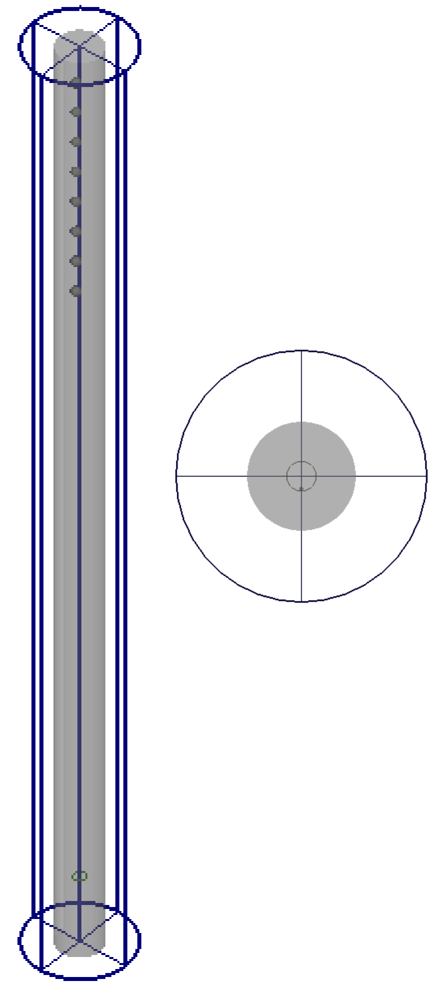
Following is a simulation comparison between 2 sources settings

A Ring dipole source with domain θ=[-180°, 180°] setting

With the same radius, with normalization enabled, the ring source has **the same simulation result** as the arc source



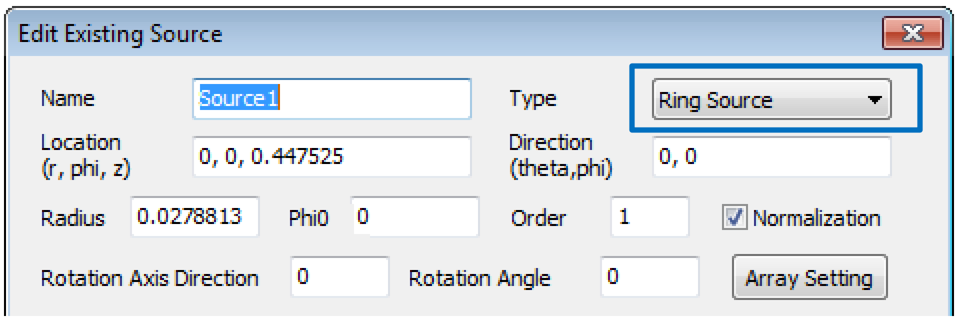


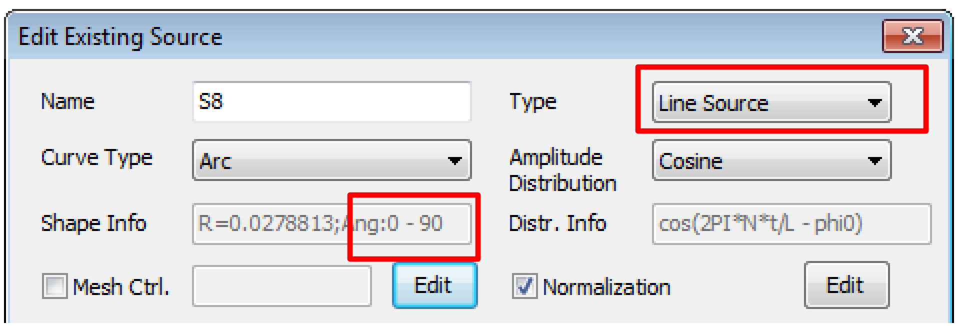


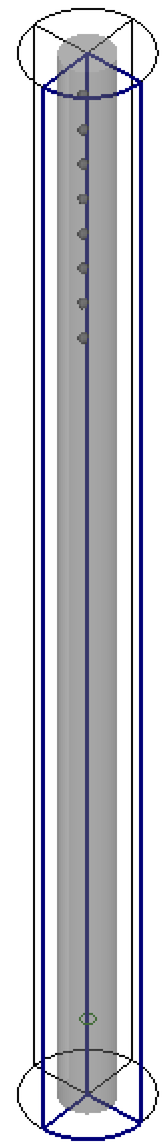
A Ring dipole source with domain θ=[0, 90°] setting

With the same radius, with normalization enabled, the simulation result of the arc source is **4 times** larger than that from the ring source

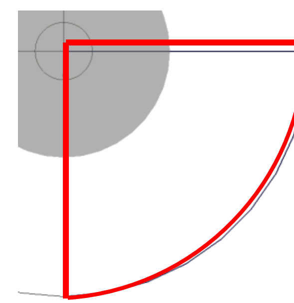
- because the length of the full ring is 4 times of the length of the arc (the arc must be in the domain)







Domain size in θ



One more consideration for the setting of the **arc** source

➢ the amplitude distribution of the arc source is based on the length of curve to the starting point of curve. For a circle, the length is proportional to the $\Delta\theta$

➢ the cosine amplitude distribution of the ring source is based on the θ value

If the arc source uses the cosine amplitude distribution, in order to make these 2 cosine functions match → requires $\Delta\theta = \theta$

➢ Therefore, the angle range of the arc source should be defined as starting from 0

Face Type Sources

The source term of the face type source has following definition

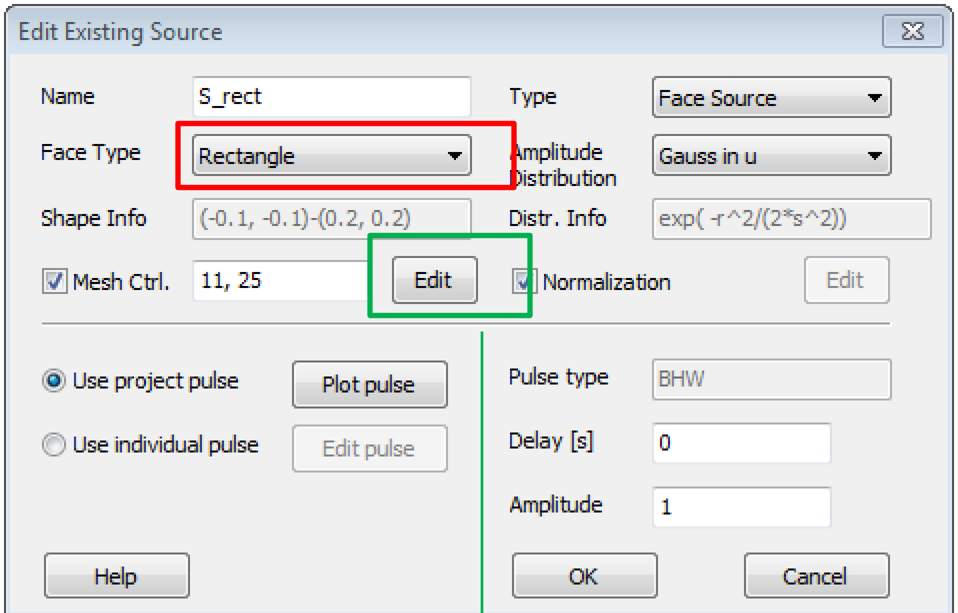
$$f(r_s) \cdot \delta_l(r - r_s)$$

Here, $r_s \in S$, S is the face with a finite size, δ_l is the δ function in **1D** for the thickness of the face at position s, $f(r_s)$ is the source distribution in the face.

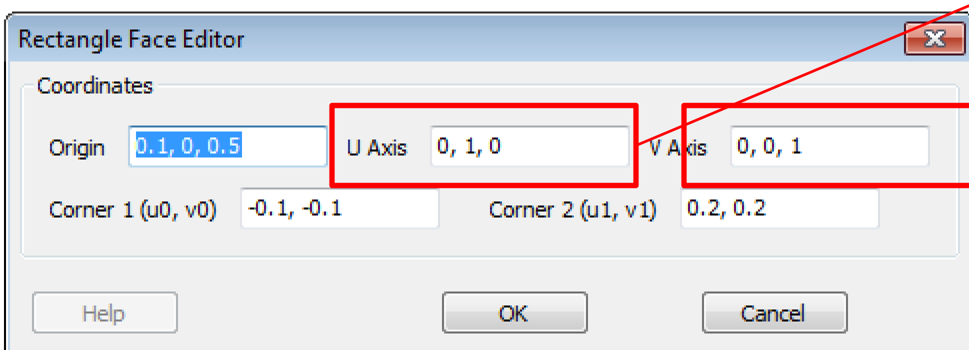
In current WCT BHA solver, it supports following types of face source

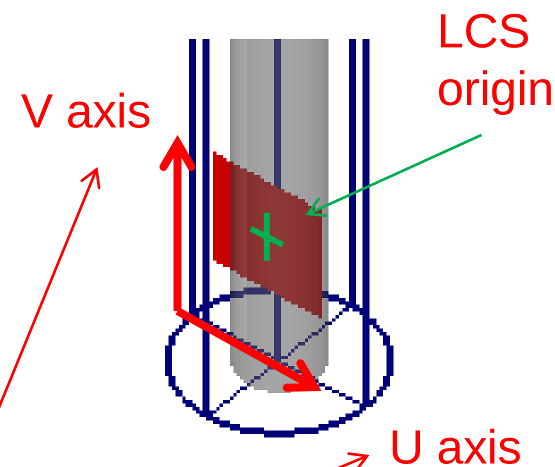
- *rectangle source*
- *circle (or ellipse) source*
- *circular cylinder source*

Rectangle Source



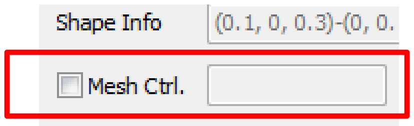
The parameters can be edited as following



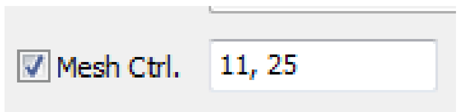


The rectangle source shown in GUI

The default setting of *discretization* is *AUTO*

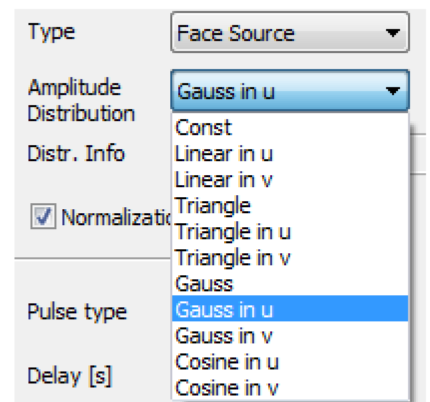


User can define a customized *discretization granularity* for this source. Here, “11, 25” means that, this source will be converted to 11x25 point monopole sources in the engine.



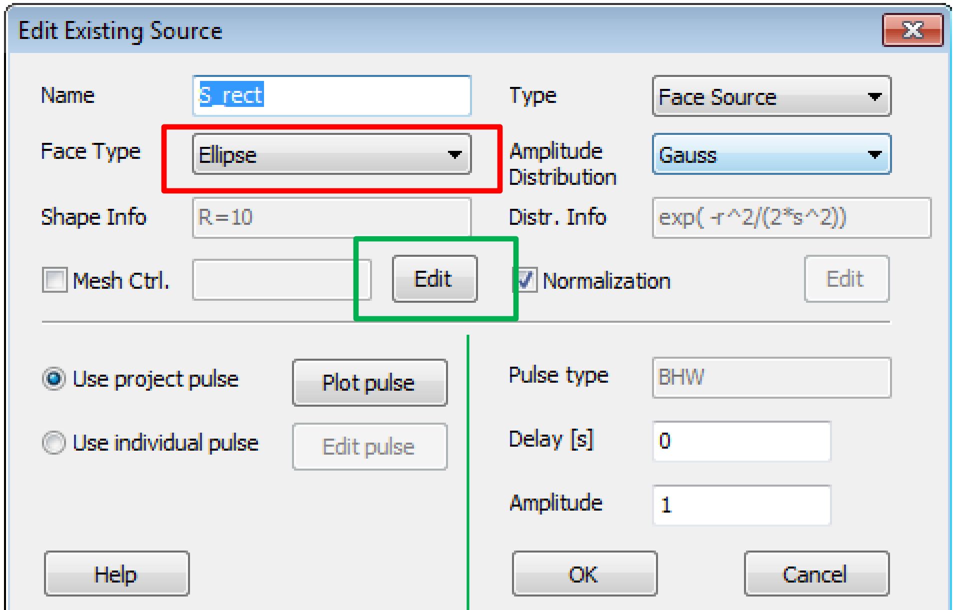
For the rectangle source, it supports following amplitude distributions

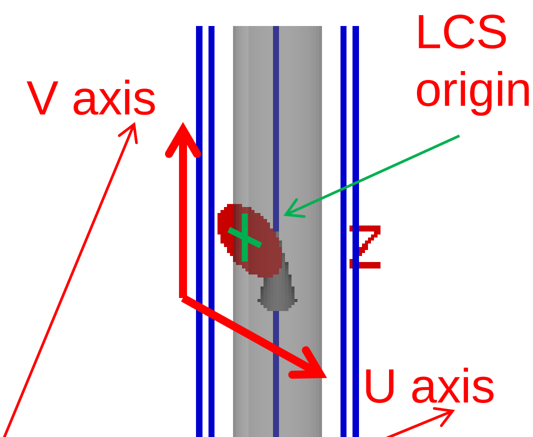
- Constant
- Linear in U
- Linear in V
- Triangle
- Triangle in U
- Triangle in V
- Gauss
- Gauss in U
- Gauss in V
- Cosine in U
- Cosine in V



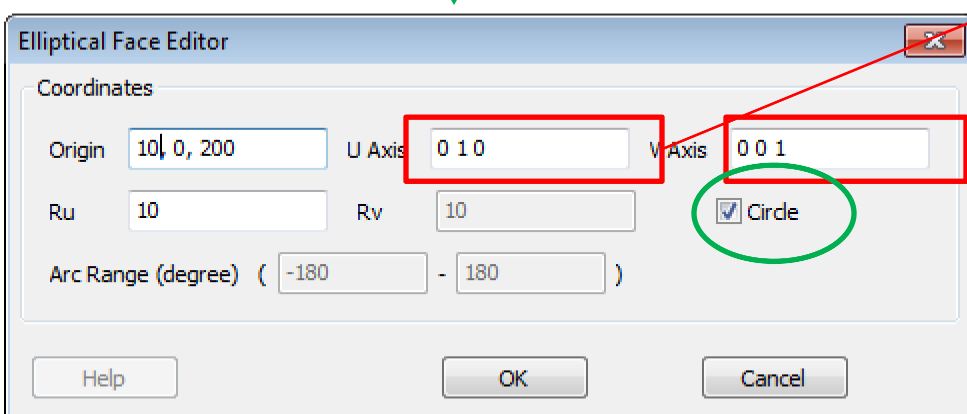
For more details for the definition of these amplitude distributions, please check section - *The amplitude distribution for the curve, face, volume types source*

Circle (Ellipse) Source



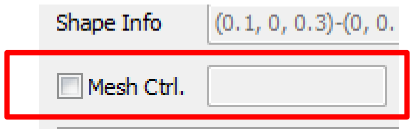


The parameters can be edited as following

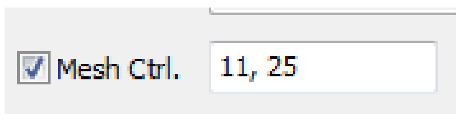


The circle source shown in GUI

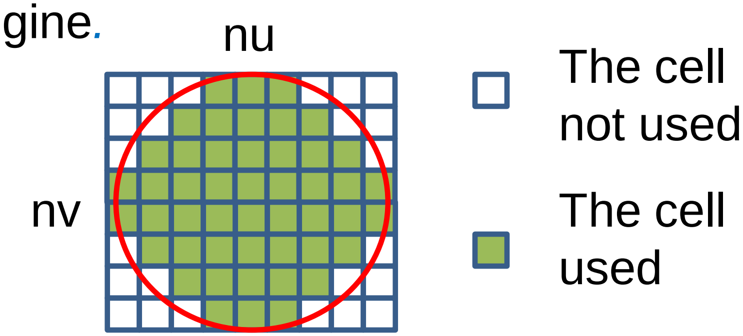
The default setting of *discretization* is *AUTO*



User can define a customized *discretization granularity* for this source. Here, “11, 25” means that, this source will be converted to 11x25 point monopole sources in the engine.

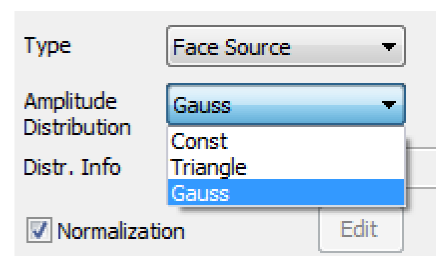


For the *discretization* definition of this shape, it is as this figure



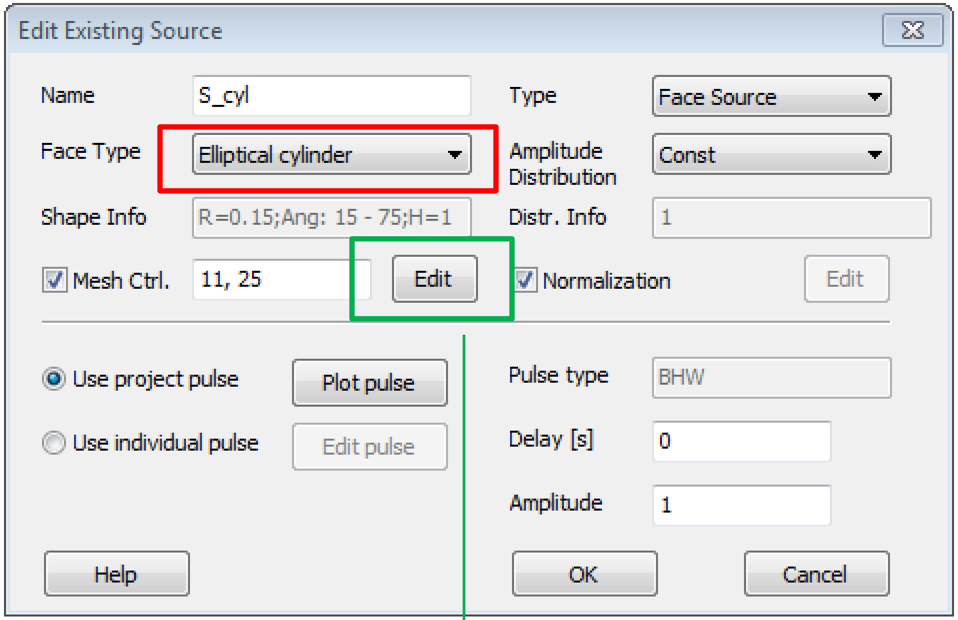
For the circle source, it supports following amplitude distributions

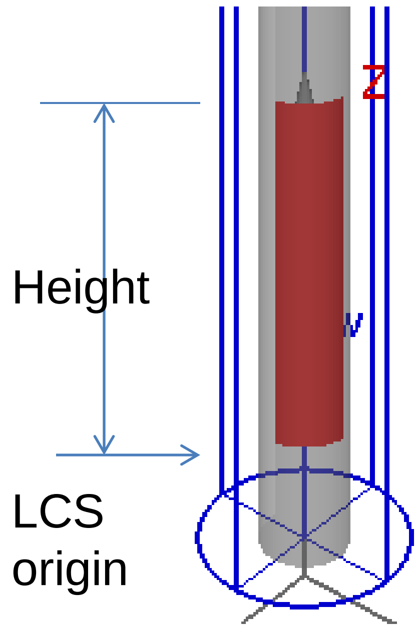
- Constant
- Triangle
- Gauss



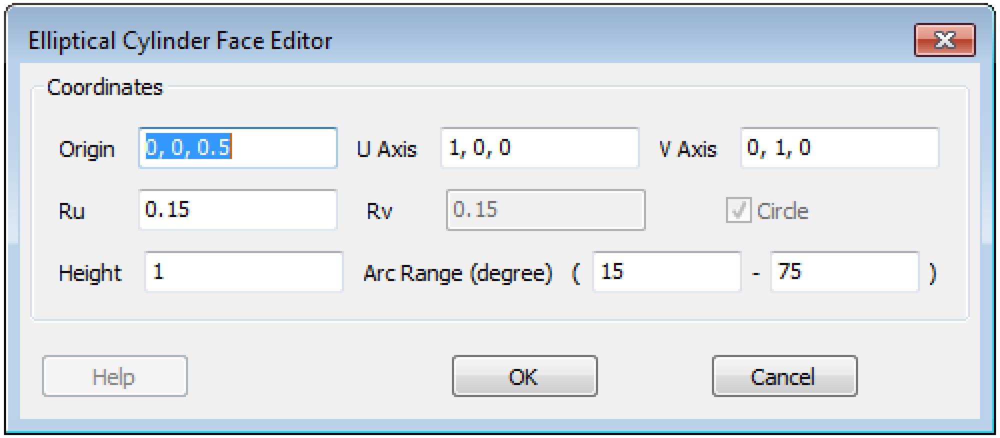
For more details for the definition of these amplitude distributions, please check section - *The amplitude distribution for the curve, face, volume types source*

Circular Cylinder Face Source



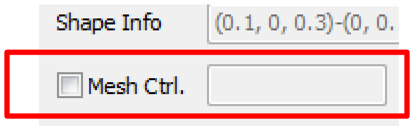


The parameters can be edited as following

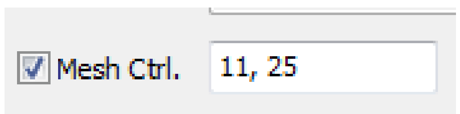


The rectangle source shown in GUI

The default setting of *discretization* is *AUTO*



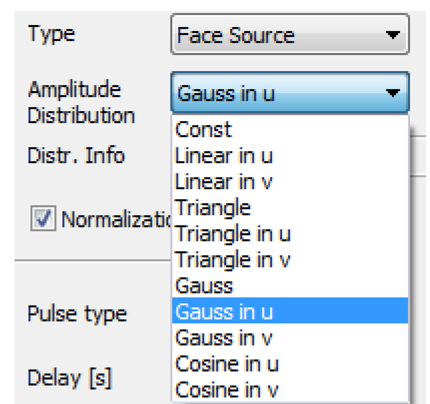
User can define a customized *discretization granularity* for this source. Here, “11, 25” means that, this source will be converted to 11x25 point monopole sources in the engine.



Here, V discretization = along the arc length
U discretization = along Z

For the circular cylinder source, it supports following amplitude distributions

- Constant
- Linear in U
- Linear in V
- Triangle
- Triangle in U
- Triangle in V
- Gauss
- Gauss in U
- Gauss in V
- Cosine in U
- Cosine in V



For more details for the definition of these amplitude distributions, please check section - *The amplitude distribution for the curve, face, volume types source*

Volume Type Sources

The source term of the volume type source has following definition

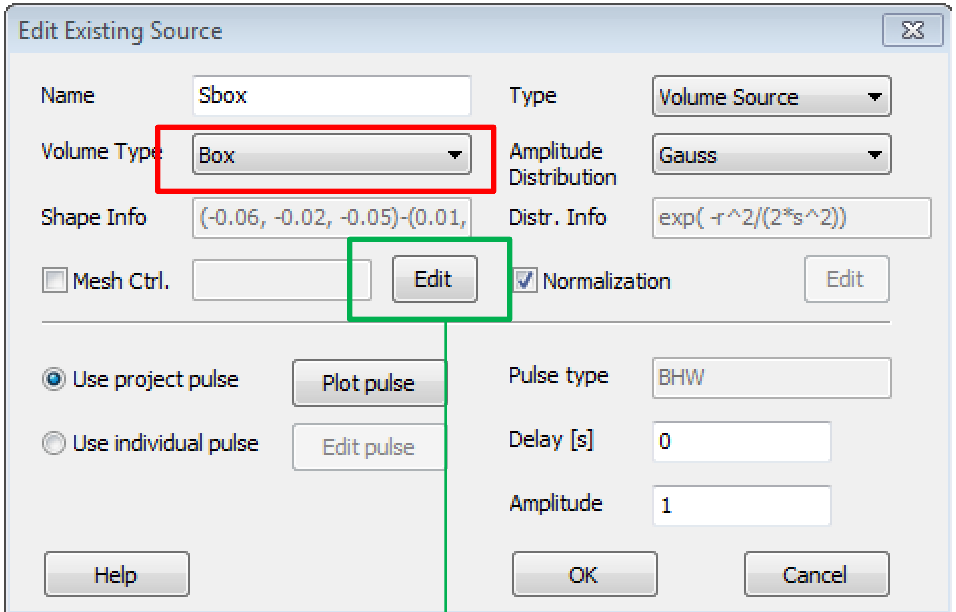
$$f(r_v)$$

Here, $r_v \in V$, V is the volume with a finite size, $f(r_v)$ is the source distribution in the volume. Here, there is not δ function.

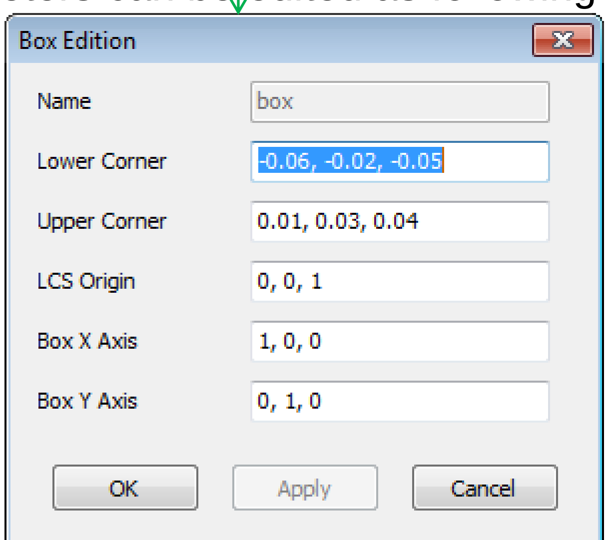
In current WCT BHA solver, it supports following types of volume source

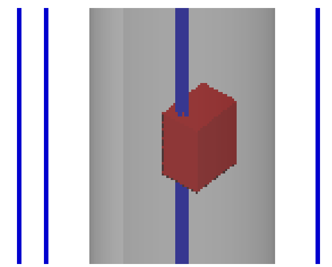
- *box source*
- *sphere (or ellipsoid) source*

Box Source



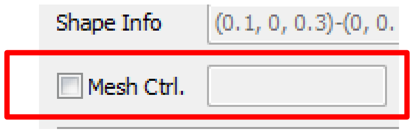
The parameters can be edited as following



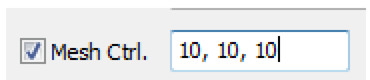


The box source shown in GUI

The default setting of *discretization* is *AUTO*

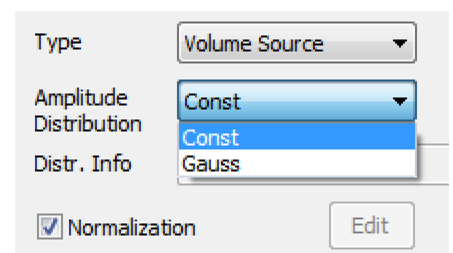


User can define a customized *discretization granularity* for this source. Here, “10, 10, 10” means that, this source will be converted to 10x10x10 point monopole sources in the engine.



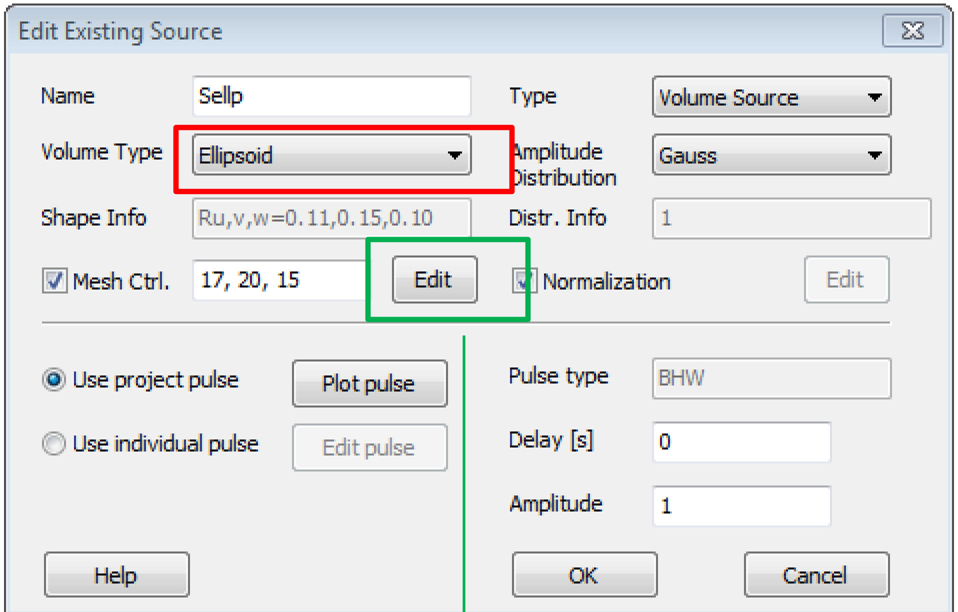
For the box source, it supports following amplitude distributions

- Constant
- Gauss

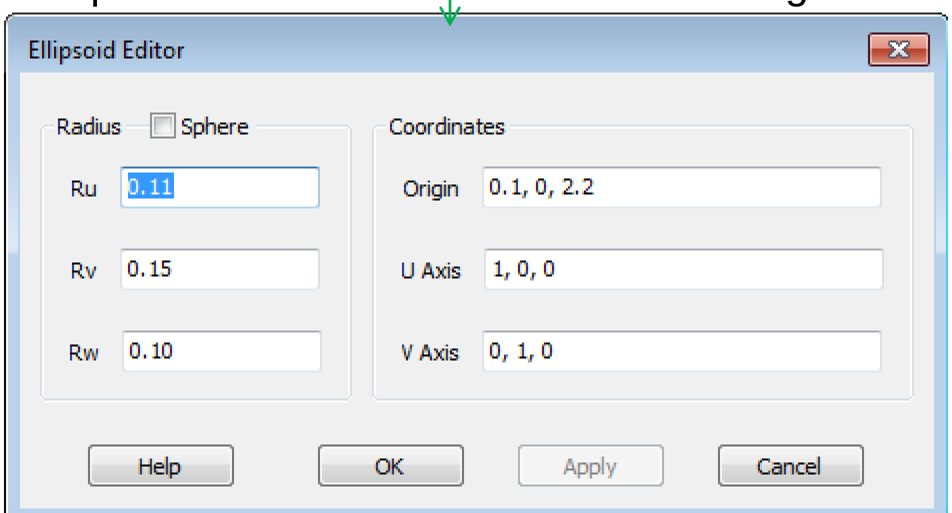


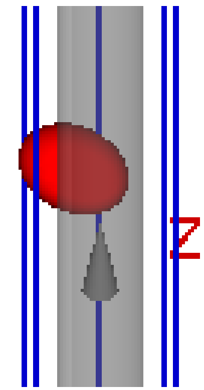
For more details for the definition of these amplitude distributions, please check section - *The amplitude distribution for the curve, face, volume types source*

Sphere (Ellipsoid) Source



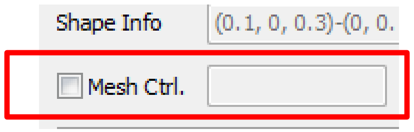
The parameters can be edited as following





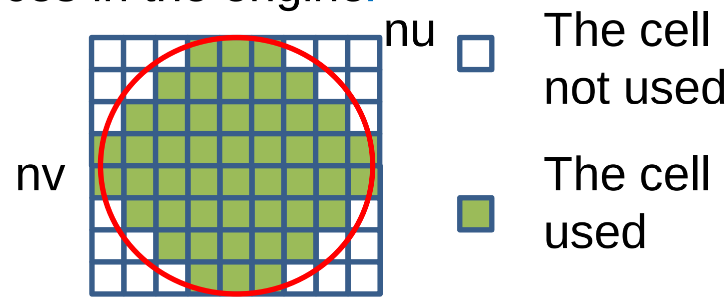
The sphere source shown in GUI

The default setting of *discretization* is *AUTO*



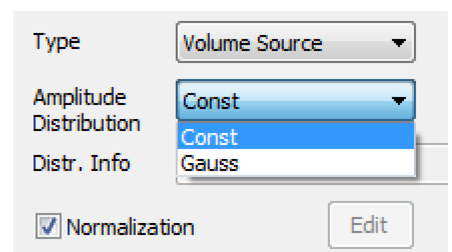
User can define a customized *discretization granularity* for this source. Here, “17, 20, 15” means that, this source will be converted to 17x20x15 point monopole sources in the engine.

For the *discretization* definition of this shape, it is as this figure



For the sphere (ellipsoid) source, it supports following amplitude distributions

- Constant
- Gauss



For more details for the definition of these amplitude distributions, please check section - *The amplitude distribution for the curve, face, volume types source*

The amplitude distribution for the ***curve, face, volume*** types source

Basically, there are following amplitude distributions

- ***Constant*** – always is 1
- ***Linear*** – the definition is: $K*t+C$
- ***Triangle*** -- max value is 1, at the edge of shape, the value is 0
- ***Gauss*** -- max value is 1 at the center; at the edge of shape, the value is that of at 3 times of RMS
- ***Cosine*** -- the definition is: $cos(2\pi*n*t/L+\theta_0)$

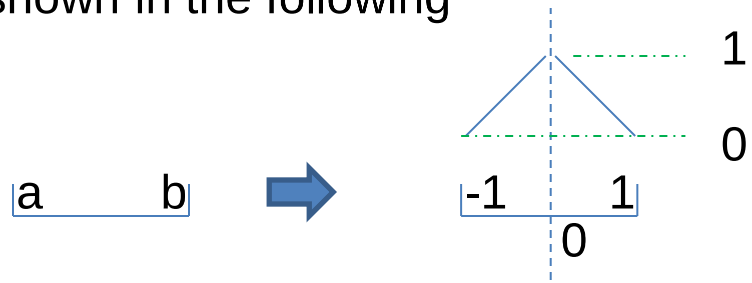
Due to the shape has different dimensions, there are different sub-types of above functions

The ***Constant*** Distribution - Not editable

- for all types of shape, the value is always 1.

The ***Triangle*** Distribution - Not editable

- **1D type** - for a finite range [a, b], will be normalized to the range of [-1, 1]. For the new range, it has the max value=1 at 0, and the value=0 at -1 & +1. As shown in the following



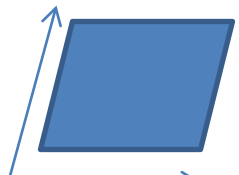
For all kind of lines: *straight line*, or *curve*, we use the length of curve, range as [0, L] to process as above.

I.e. for a 1D shape, we define the U axis is the length of curve-segment (related to the curve starting point).

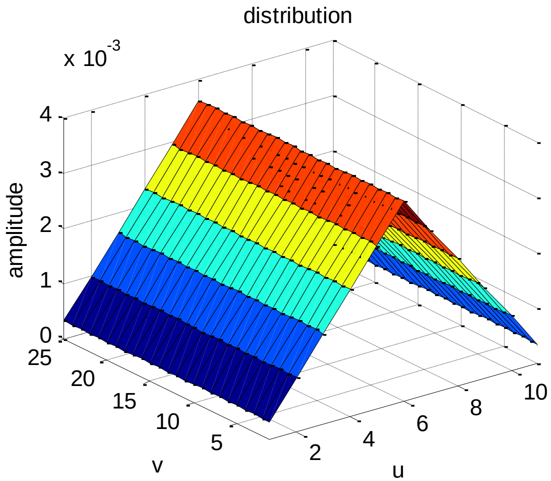
The ***Triangle*** Distribution (cont.)

- **2D type** – The shape has 2 dimensions, the face shape can be expressed as face(u, v). So, it include 3 sub-types
  - Triangle in U only
  - Triangle in V only
  - Triangle

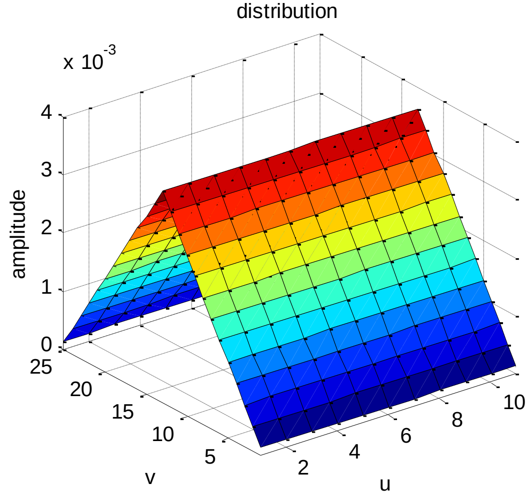
- for Triangle in U or V only, it is supported by *Rectangle* and *Circular cylinder* because these 2 shapes are rectangular in 2D U, V plane. The distribution is shown in following figures



Triangle in U only

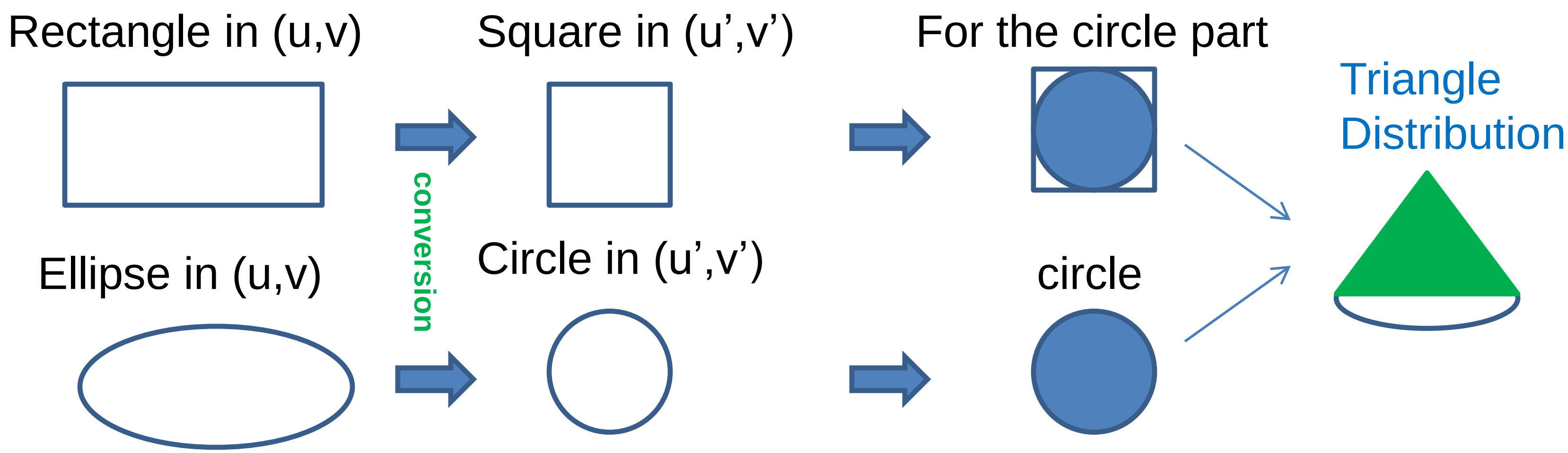


Triangle in V only



- for the Triangle Distribution in 2D shapes, it is supported by *Rectangle, circle (ellipse)* and *Circular cylinder*

    - here,
        - both U & V will be converted to the range [-1, 1],
        - then the distance to the new shape center is: $r=\mathrm{sqrt}(\Delta u^2 + \Delta v^2)$
        - the distribution value will be
            - Triangle Distribution, if $r \in [0, 1]$
            - 0, if $r>1$

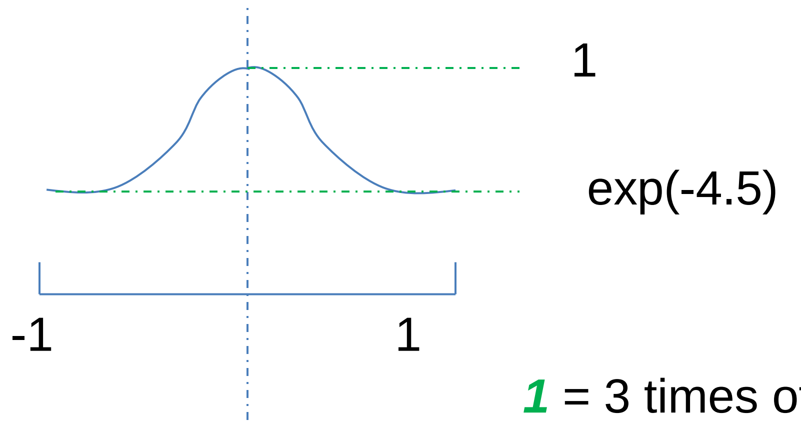


The ***Gauss*** Distribution - Not editable

- it include following sub-types
    - Gauss Distribution
    - Gauss Distribution in U only
    - Gauss Distribution in V only

- similar to the Triangle distribution, the shape transformation method is the same, only the distribution method is changed to the Gauss function.

- The Gauss Distribution has following shape

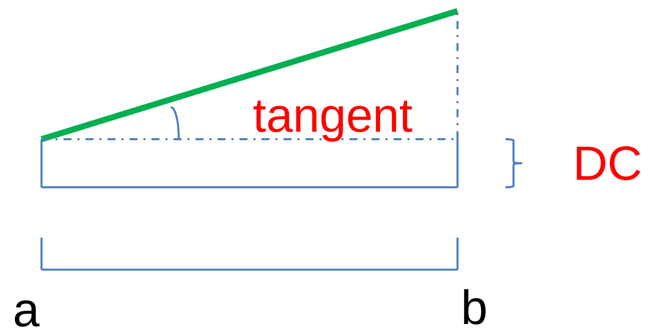


1 = 3 times of the RMS (or variance) of the Gauss function

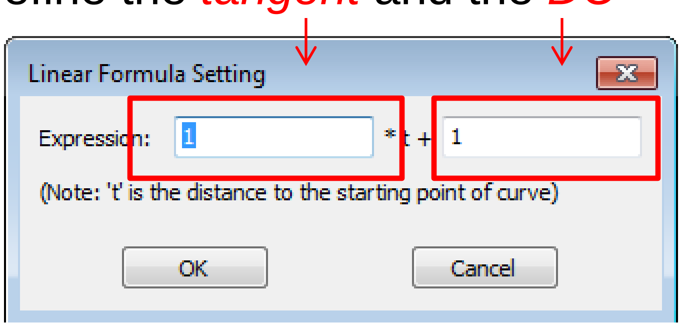
The ***Linear*** Distribution - editable

- it include following sub-types
  - Linear Distribution - for 1D shape only
  - Linear Distribution in U only - for 2D & 3D shape
  - Linear Distribution in V only - for 2D & 3D shape
  - Linear Distribution in W only - for 3D shape only

- The Linear Distribution, for any U, V, W axis, has following shape



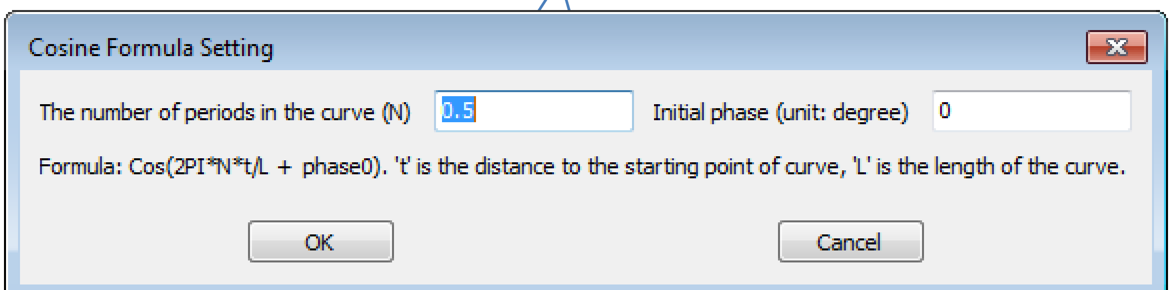
- User need to define the *tangent* and the *DC*



The ***Cosine*** Distribution - editable

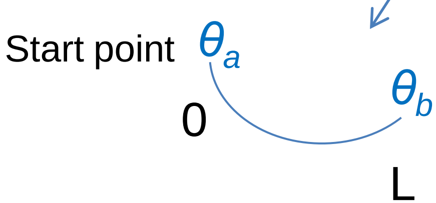
- it include following sub-types
  - Cosine Distribution - for 1D shape only
  - Cosine Distribution in U only - for 2D & 3D shape
  - Cosine Distribution in V only - for 2D & 3D shape
  - Cosine Distribution in W only - for 3D shape only

- The Cosine Distribution, for any U, V, W axis, has following definition
  - $cos(2\pi * n * t/L + \theta_0)$
    - which means that, for a 1D range [**0, L**], it has **n** periods for the cosine function.



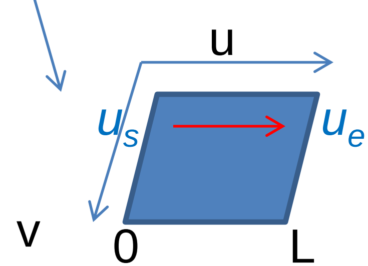
- User need to define the period number n (it is a floating number, for example, 0.25, 0.5, 1, 2, etc.) and the *initial phase - θ_0*

For an arc

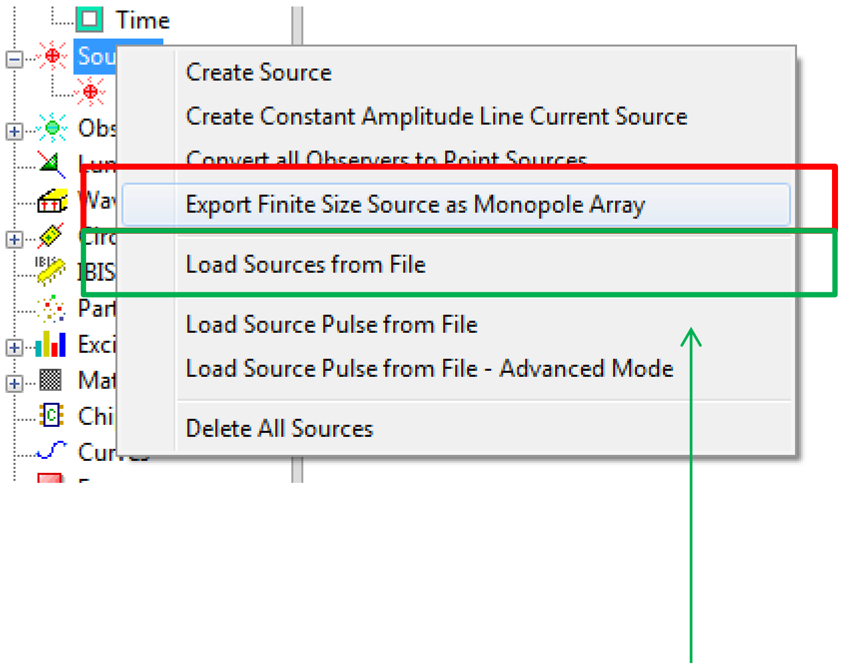


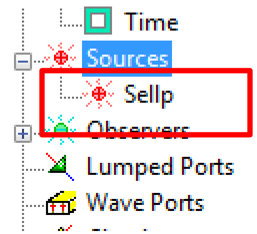
For a rectangle

Cosine Distribution in U



A Tool to Check the Discretization of Curve, Face & Volume Source



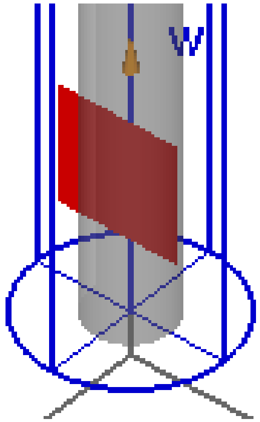


For each curve, face or volume source, a text file will be created for the source, the file name is the same as the source name. Here, the file name is "Sellp.txt".
The file format is the same as that in section "*Create Sources by Data File*"

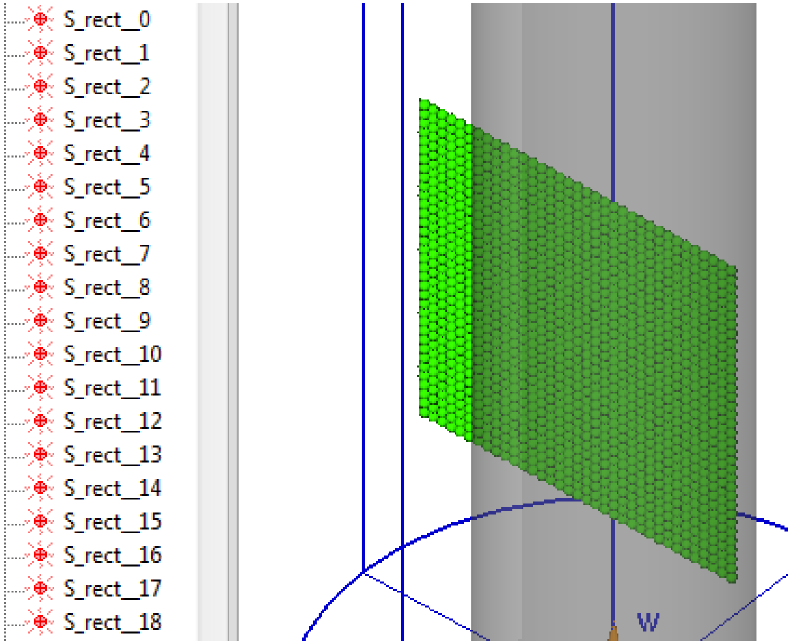
Then user can load the exported data file by "Load Sources from file" to check the monopole array.

Following is the example of a rectangle source

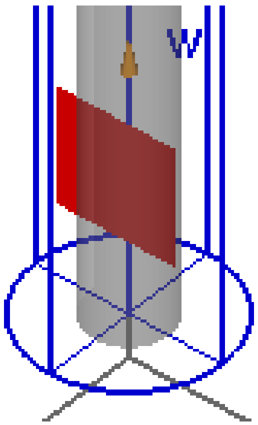
Original rectangle source



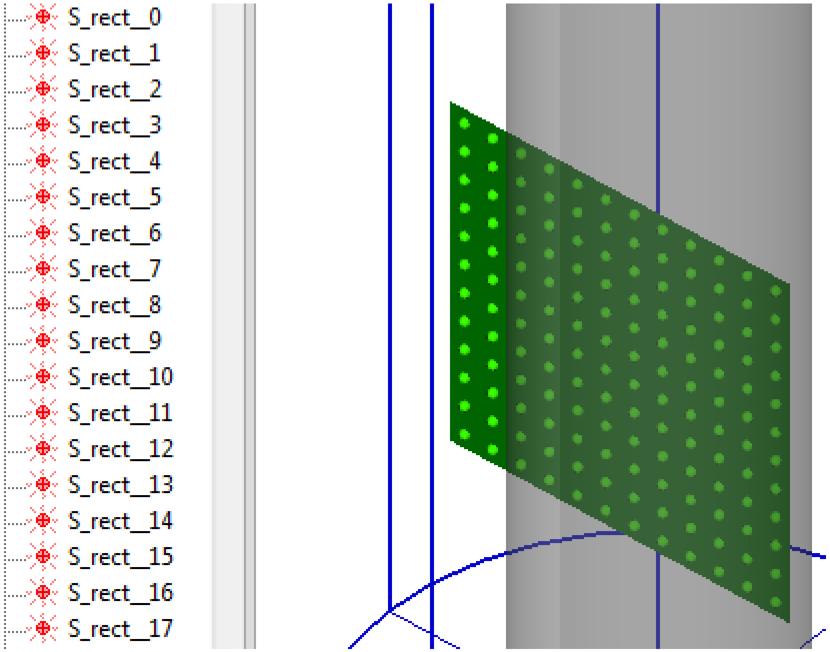
Loaded discretized source with AUTO mesh



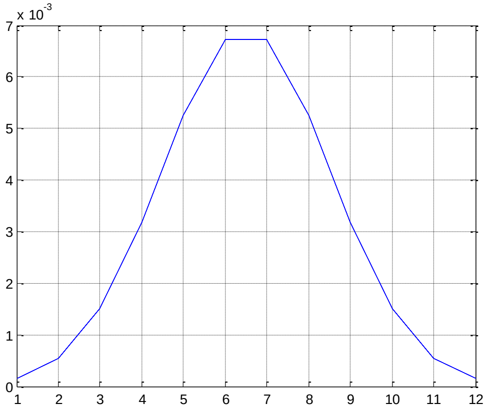
Original rectangle source



Loaded source with user defined discretization=12x12



For this setting, user can check amplitude distribution by the 5th column of the export data file.
Total 12x12=144 sources, 12 rows and 12 columns, respectively.
For this demo, the distribution is "Gauss in U"

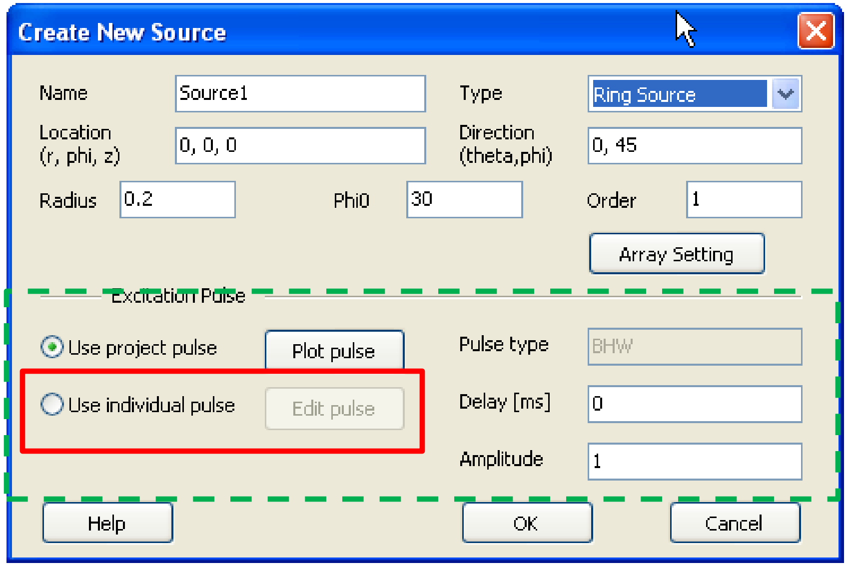


Individual Excitation Pulse

In the default setting, all sources in a project will use the same pulse type defined in the project setting, but with individual magnitude & delay.

However, each source could use individual excitation pulse instead of the common project pulse defined in the section "**Project Frequency Range and Excitation Pulse for Source(s)**"

Choose the "Use individual pulse" option, then edit the pulse

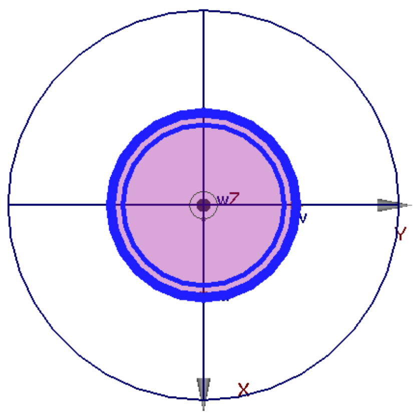


Dealing Source Type and Boundary Conditions in ϕ

For a ring source, it has a magnitude distribution on a circle as cos(nϕ).

1) n=0, it is a **ring monopole**, having a const magnitude in the circle.

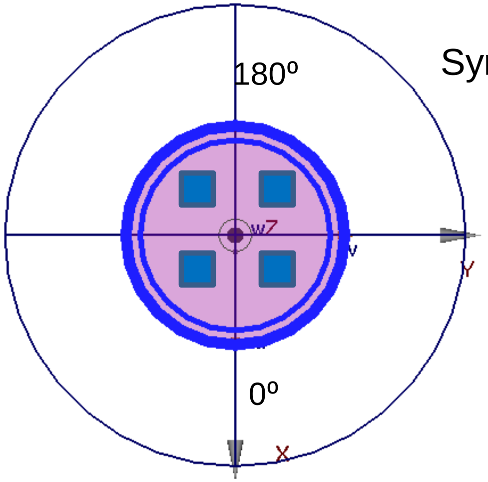
For this type of source, if the solids in the project is circular uniform, i.e., all solids are ring with Z axis as the center axis, as following



There is not any circular field in wave propagation. So, the space sampling density in ϕ is useless in the simulation, user can set the total cell number in ϕ as 1.

2) n=1, it is a **ring dipole**, the magnitude distribution in the circle is cos(ϕ).

For this type of source, if the solids in the project has some kinds of symmetry distribution along X or Y plane, as following



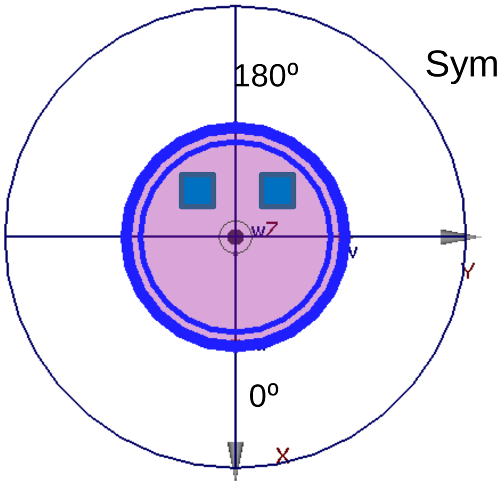
Symmetry along X & Y planes

Due to the magnitude distribution of source is cos(ϕ), it is symmetry at 0^0 & 180^0, anti-symmetry at 90^0.

So, user can define a smaller ϕ range instead of 360^0 to reduce the project size

a) a quarter of ϕ space to represent the whole project:
   - ϕ range=[**0 ^{0}, 90 ^{0}**]
   - **Symmetry** B.C. type at ϕ_{min}; **Anti-Symmetry** B.C. type at ϕ_{max}.

b) Half ϕ space to represent the whole project:
   - ϕ range=[**0 ^{0}, 180 ^{0}**]
   - **Symmetry** B.C. type at ϕ_{min}; **Symmetry** B.C. type at ϕ_{max}.



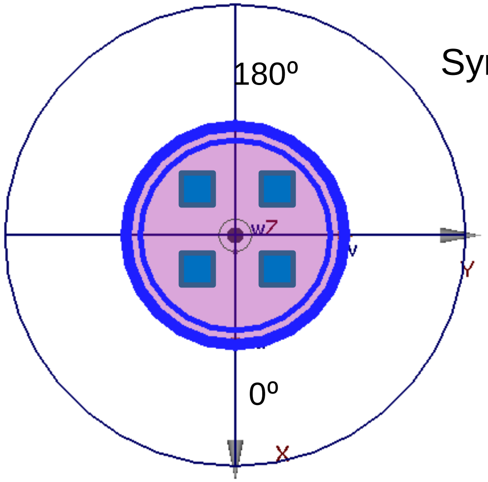
Symmetry along X plane

In this situation, user can define half ϕ space to represent the whole project:

- ϕ range=[**0 ^{0}, 180 ^{0}**]
- **Symmetry** B.C. type at ϕ_{min}; **Symmetry** B.C. type at ϕ_{max}.

3) n=2, it is a **ring quadrupole**, the magnitude distribution in the circle is cos(2ϕ).

For this type of source, if the solids in the project has some kinds of symmetry distribution along X or Y plane, as following



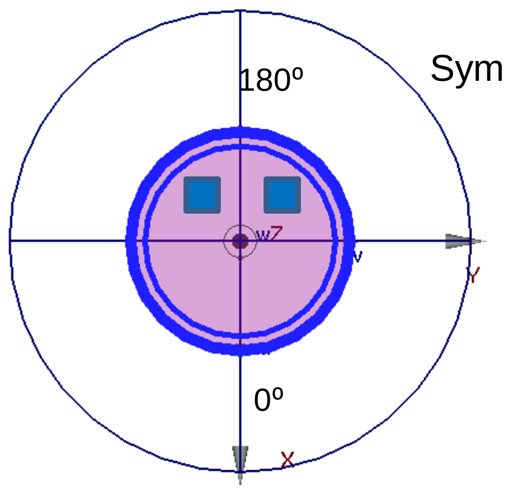
Symmetry along X & Y planes

Due to the magnitude distribution of source is cos(2ϕ), it is symmetry at 0^0 , 90^0 & 180^0.

So, user can define a smaller ϕ range instead of 360^0 to reduce the project size

a) a quarter of ϕ space to represent the whole project:
- ϕ range=[**0^0, 90^0**]
- **Symmetry** B.C. type at ϕ_{min}; **Symmetry** B.C. type at ϕ_{max}.

b) Half ϕ space to represent the whole project:
- ϕ range=[**0^0, 180^0**]
- **Symmetry** B.C. type at ϕ_{min}; **Symmetry** B.C. type at ϕ_{max}.

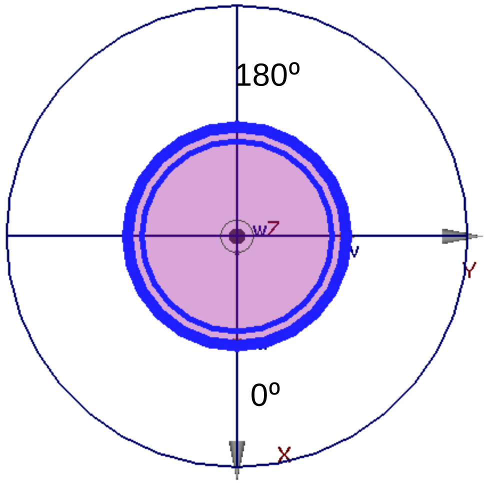


Symmetry along X plane

In this situation, user can define half ϕ space to represent the whole project:

- ➢ ϕ range=[**0 ^{0}, 180 ^{0}**]
- ➢ **Symmetry** B.C. type at ϕ_{min}; **Symmetry** B.C. type at ϕ_{max}.

Moreover, if the structure of the project is pure circular, or is symmetry by every 45^0 planes,

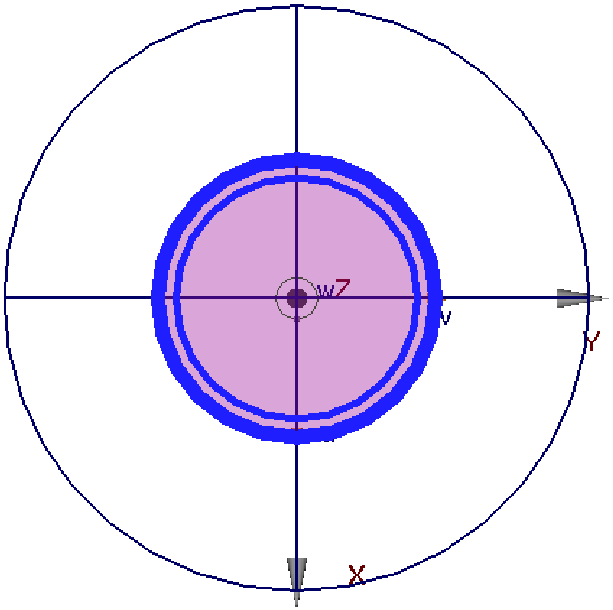


for a ring quadrupole source, user can use a 1/8th circle range instead of the full circle in the simulation.

- ϕ range=[**0 ^{0}, 45 ^{0}**]
- **Symmetry** B.C. type at ϕ_{min}; Anti-**Symmetry** B.C. type at ϕ_{max}.

Space Sampling Density in ϕ

For a cylindrical project as following



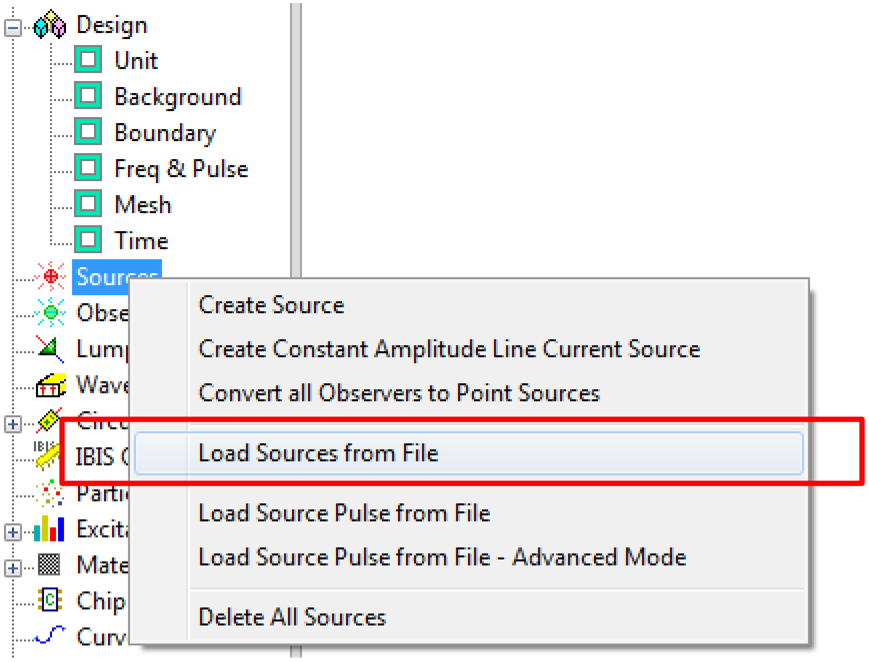
if the source is a ring dipole source, according to our experience, $\Delta\phi=15^0$ can provide accurate enough result. Therefore, for this kind of structure, we suggest employ "Advanced" mesh option

- R & Z direction use Automatic mesh
- ϕ direction uses uniform mesh to make $\Delta\phi\approx15^0$

Create Sources by Data File

In some cases, user need to define array of sources, for example, 1000 point monopole sources at specified positions (those positions are not suitable for using “Array Creation”). In this situation, it is hard to define these sources manually in GUI.

GUI provides a function to load these sources from a data file, as following



The data file is a text file in ASCII format, each line defines a source.

The format of each line can be one of following:

1) type R ϕ Z magnitude pr pϕ pz

2) type R ϕ Z magnitude pr pϕ pz "src name"

Type: 0 – point monopole; 1 – point dipole; 2 – ring source

R, ϕ, Z: the position of the source. For ϕ, the unit is degree. For R and Z, it is the project unit in length.

Magnitude: the magnitude

pr, pϕ, pz: the polarization of the dipole source. For monopole source and ring source, they are not used, can be set as "0 0 0"

For the format (2), there is a string enclosed by a pair of "", it is the name of the source. But it may conflict with existing source in the project, GUI will append "_1" or so to solve the problem

User can also add comment in the end of each line. The comment start from "//". For example,

type x y z amplitude px py pz *// this is the comment*

Both format (1) & (2) support variable or expression as input, for example,

```
0    r_src*1.05*cosd(10)/cosd(5)   -5    0.14        1       0  0  0
```

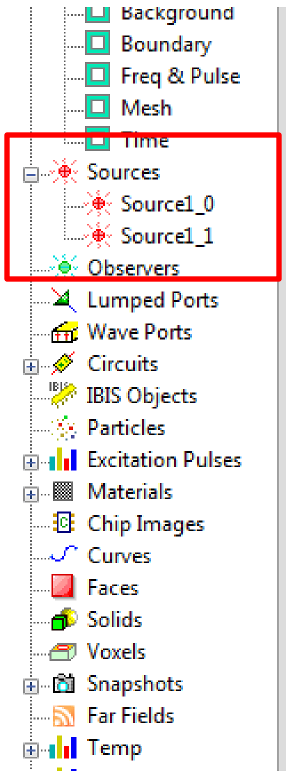
This line define a source, the x (or r) position is an expression using a variable in the project and an intrinsic function of the GUI.
(Note: for the variable and the expression, SPACE is not allowed)

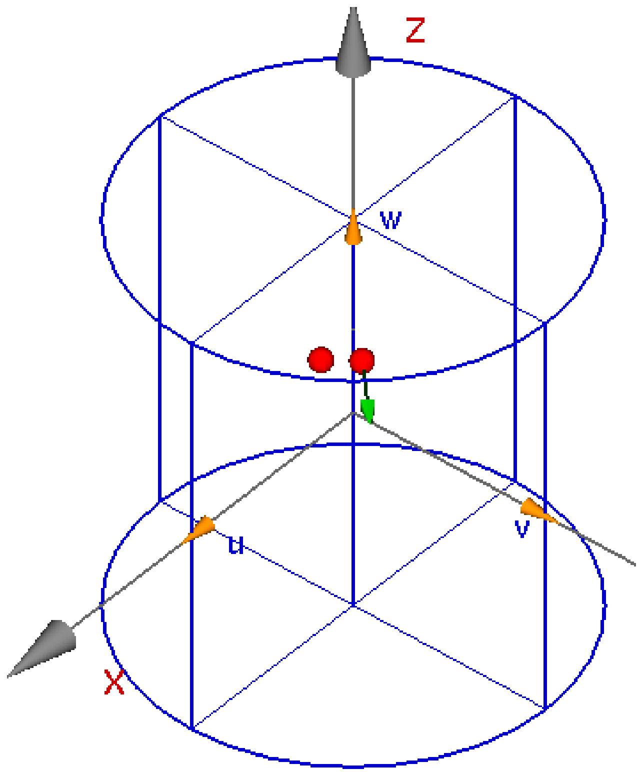
Following are examples of the data file

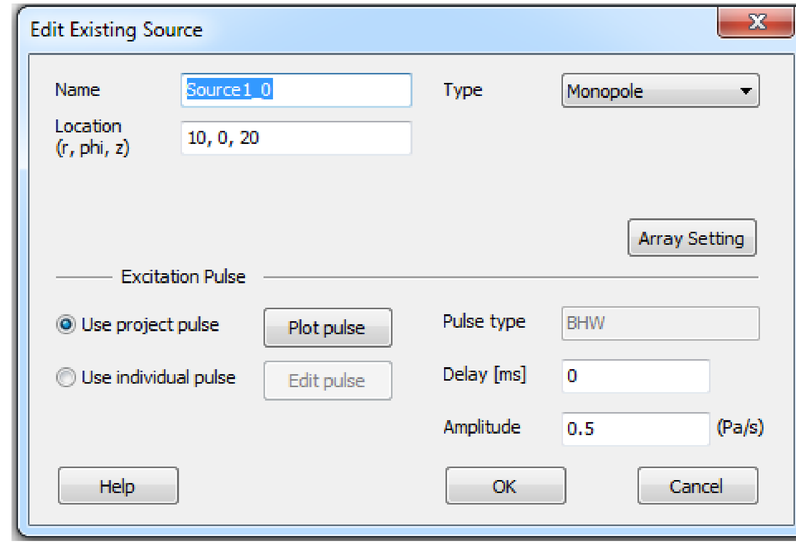
(1) All lines use format (1)

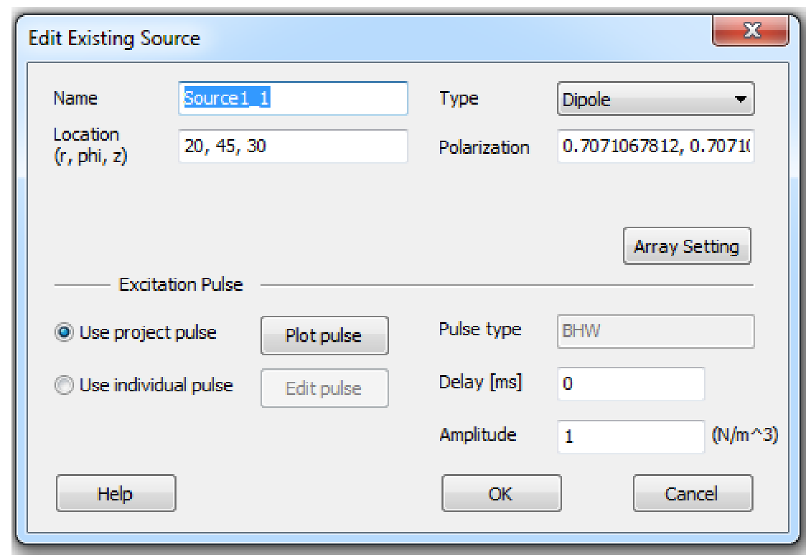
```
0     10     0    20     0.5      0    0     0
1      20    45  30     1          1    1     0
```

After loading, 2 sources are created as following









(2) Mix format (1) & (2)

The comment in the file

```
0  1  30  1  1.1  0  0  0           // define a point monopole source
                                      // at(r=1, phi=30°, z=1); amplitude=1.1; auto name

0  1  45  1  1  0  0  0  “src1”   // define a point monopole source
                                     // at(r=1, phi=45°, z=1); amplitude=1; name=“src1”

1  2  30  1  1  1  0  0            //  define a point dipole source
                                     // at(r=2, phi=30°, z=1); amplitude=1;  polarization=(1,0,0); auto name

2  0.5  0  a+1  1  0  0  0  “src2”  // define a ring source
                                        // at z=“a+1”; radius=0.5, amplitude=1; name=“src2”
```

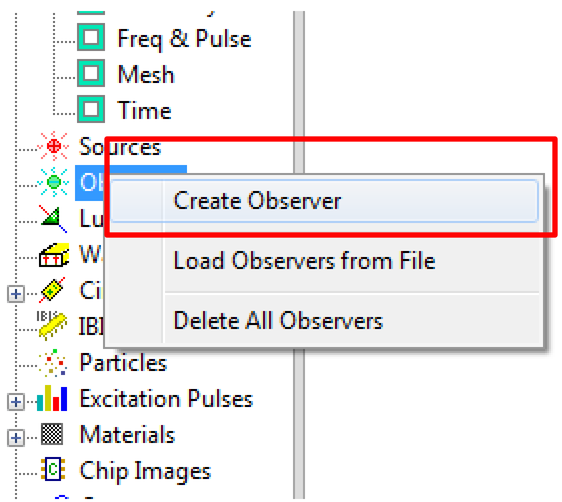
Here, the 4th source uses an expression and the variable “a” to define the Z position. In order to make it work, the project should have defined the variable “a” before loading. Otherwise, the load operation will fail. Meanwhile, the expression can’t include SPACE.

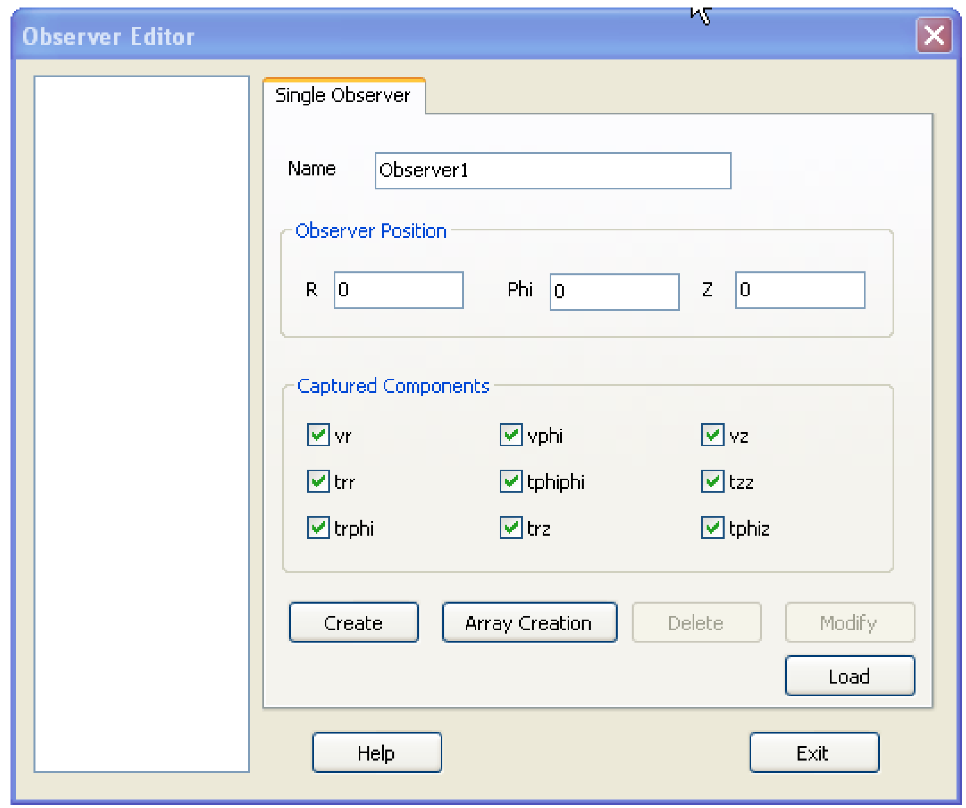
Field Monitoring

Observer

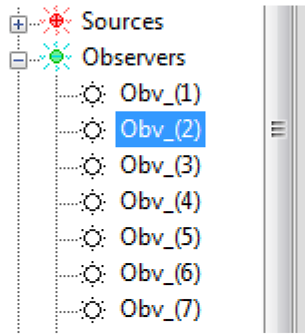
User can define observer to record the simulation data. The recording data are

1) velocity: vr, vphi, vz
2) stress: trr, tphiphi, tzz, trphi, trz, tphiz

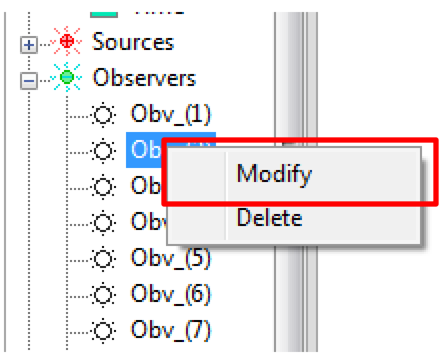


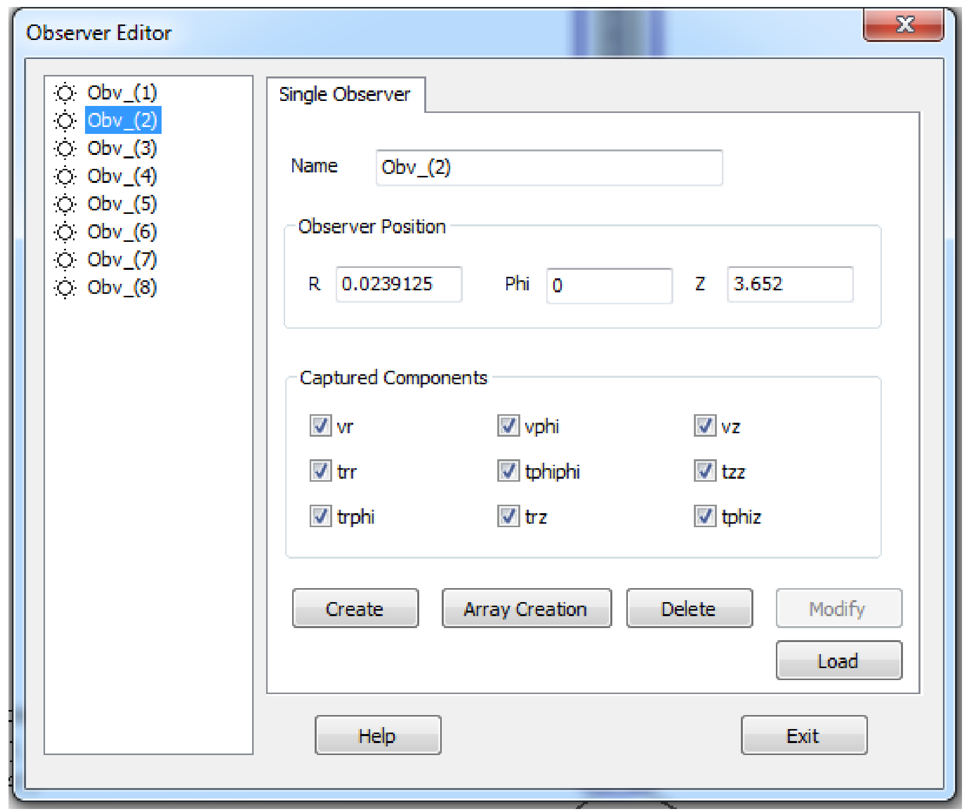


After receiver is created, it can be modify by double clicking the selected observer, or use the popup menu



Double click the selected item

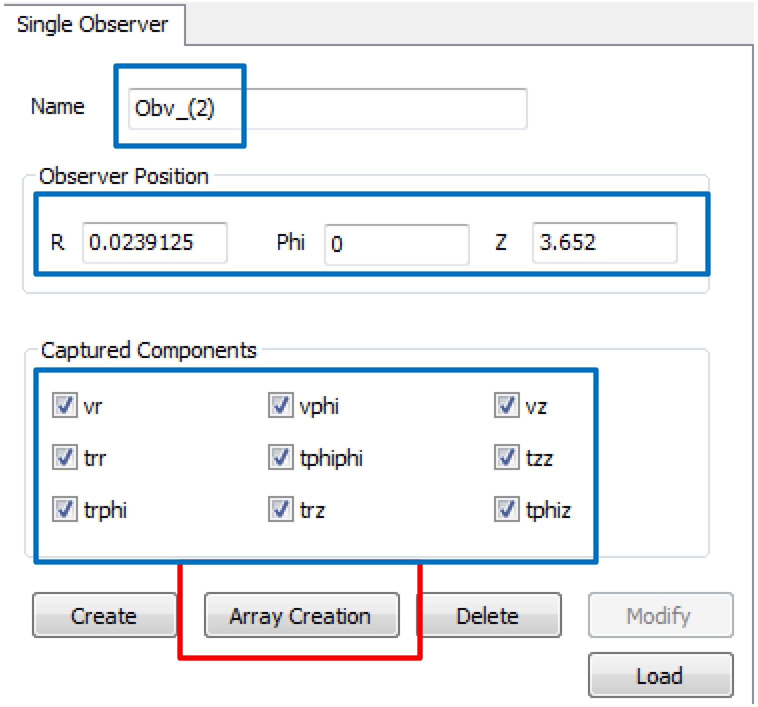




Generate an Observer Array

WCT GUI provides a convenient way to generate observer array.

In the observer editor, complete the content for the 1st item in the array. The press “Array Creation” button.



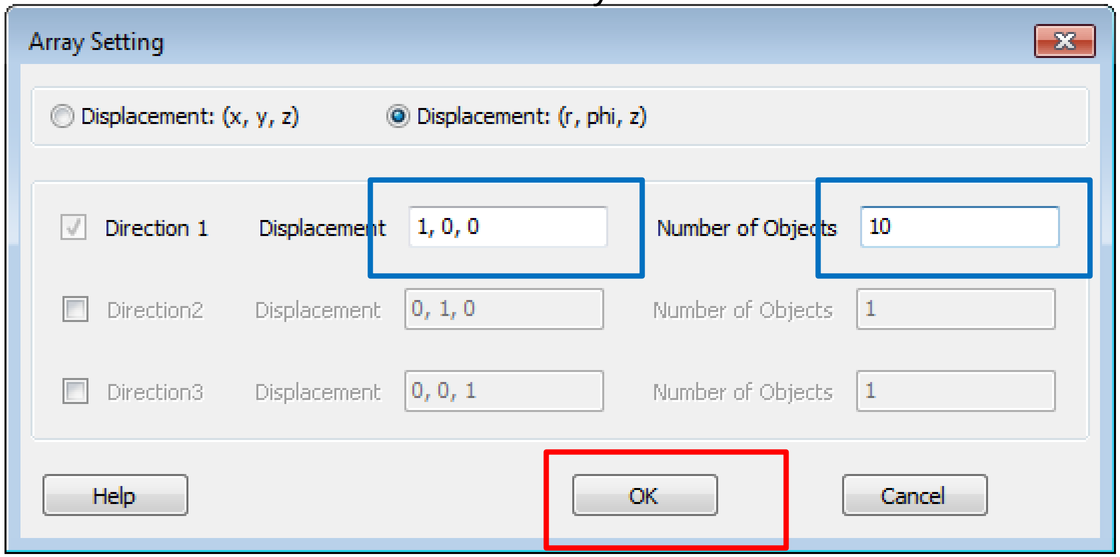
It will be the base name of the observers in the array

It will be the start position of the array

The receivers in the array will record these fields

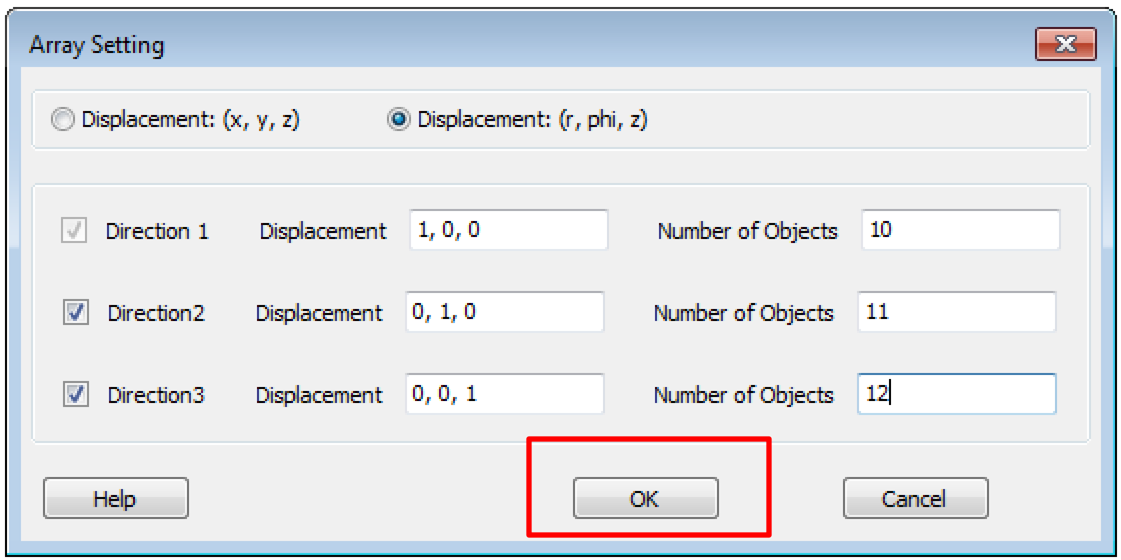
Distance between 2 items in the array

How many items in this dimension



1D array

The array can be 1D, 2D or 3D.

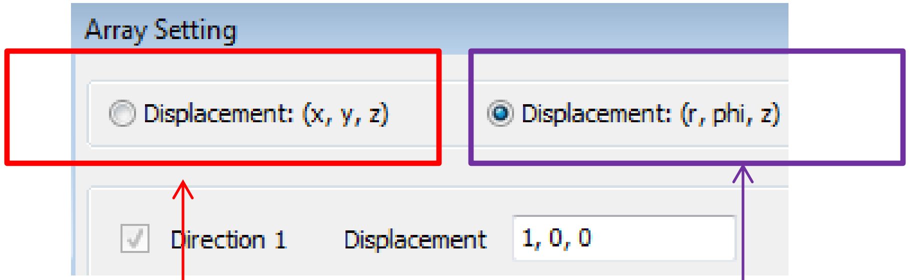


3D array

Press “**OK**” button, the observer array will be generated.

Cylindrical or Planar Array

There is an option for the unit of displacement values



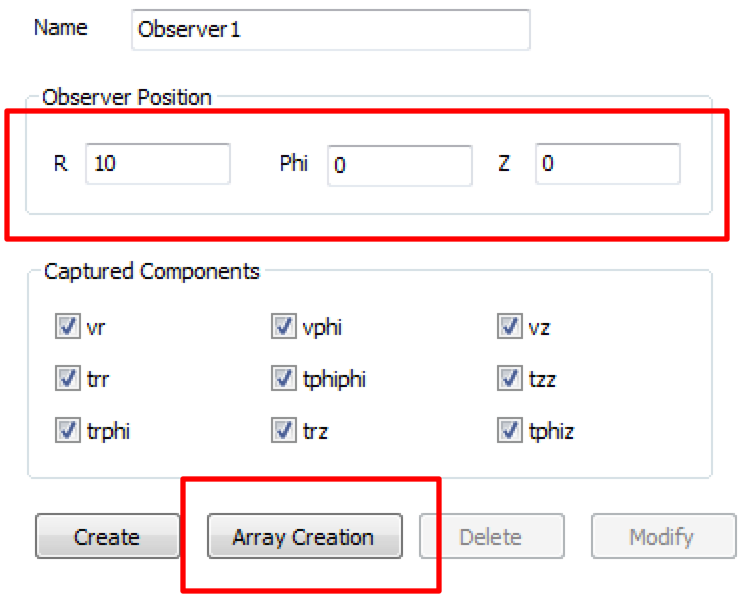
It means that not matter the project is in the Cylindrical coordinates system or in the Cartesian coordinates system, the displacement value is (x, y, z)

It means that not matter the project is in the Cylindrical coordinates system or in the Cartesian coordinates system, the displacement value is (r, θ, z)

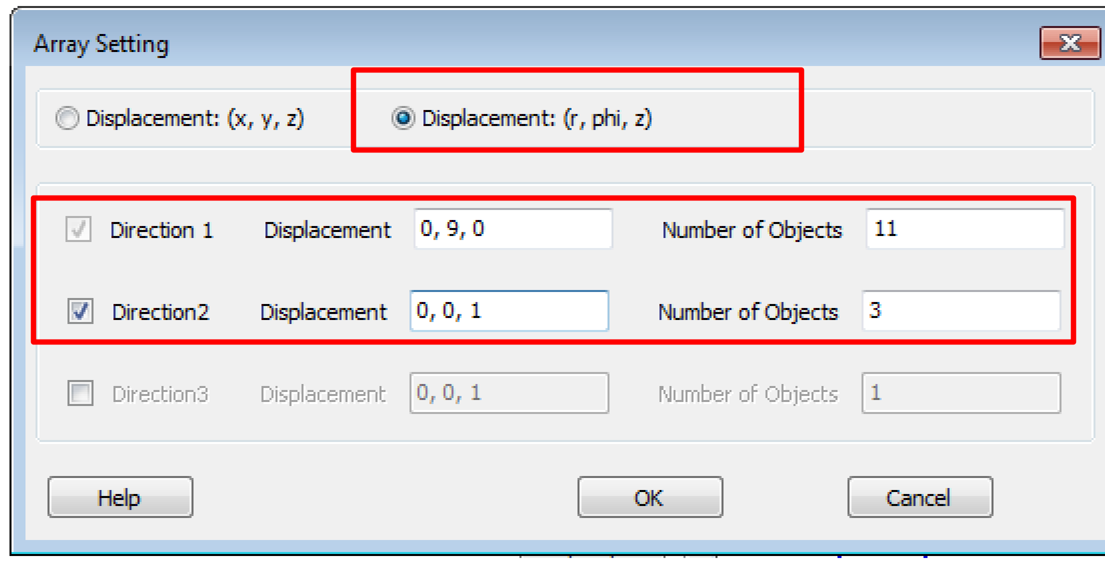
The default choice is the same as the project coordinates system

Example (1) 2D cylindrical observer array in a BHA project

(1) Define the start point of the array

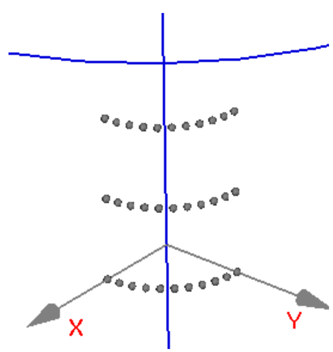


(3) With default option



(2) Click this button

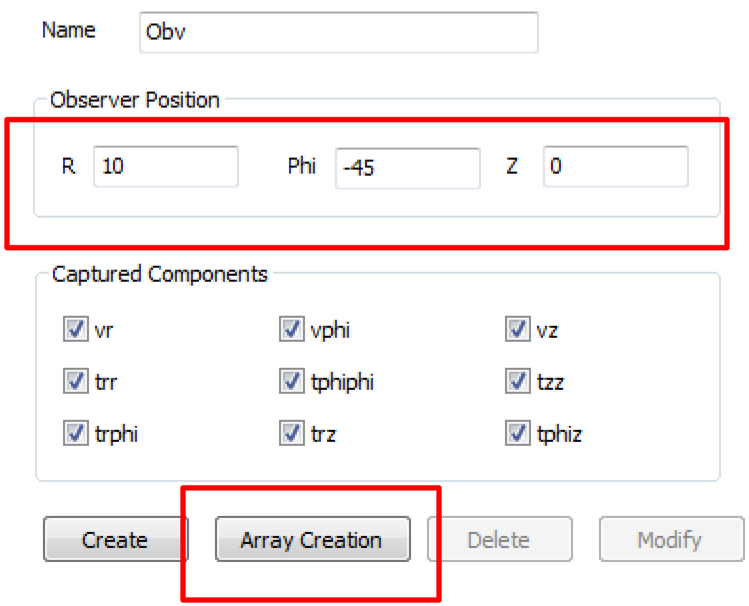
(4) Define the displacement and observer number in each direction



This is the cylindrical 2D array

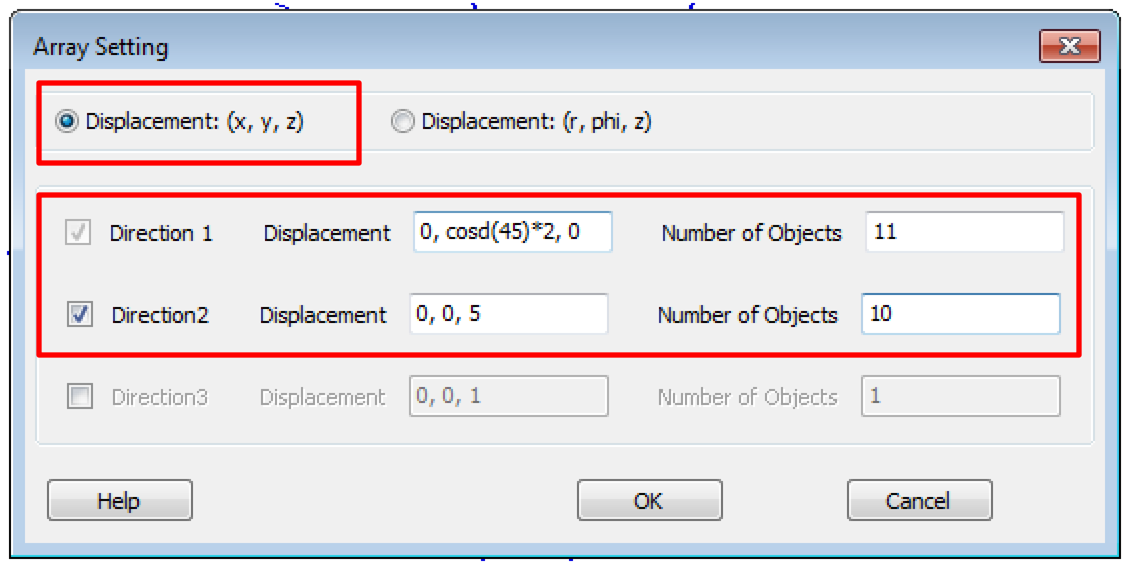
Example (2) 2D planar observer array in a BHA project

(1) Define the start point of the array by (r, θ, z) value

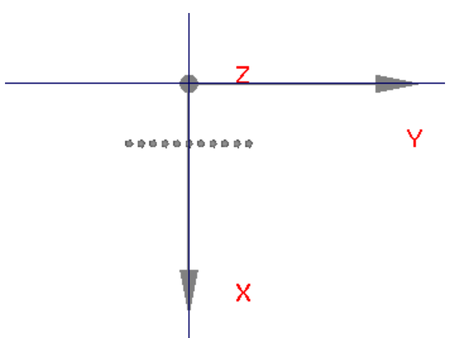


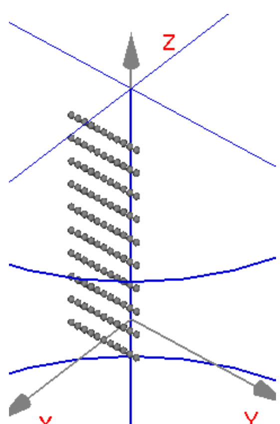
(2) Click this button

(3) Set as (x, y, z) displacement



(4) Define the displacement and observer number in each direction



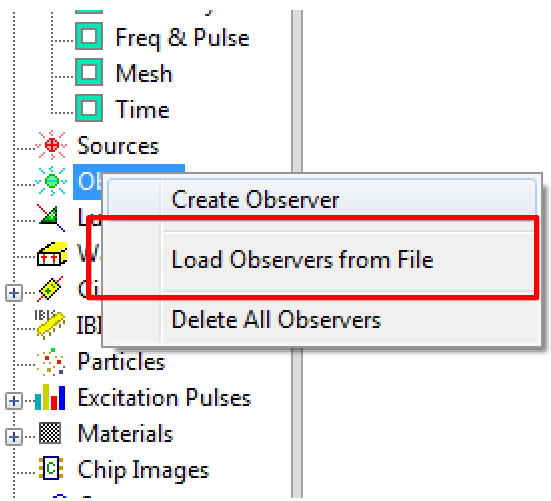


This is the planar 2D array

Create Observers by Data File

In some cases, user need to define array of observers, for example, 1000 observers at specified positions (those positions are not suitable for using “Array Creation”). In this situation, it is hard to define these observers manually in GUI.

GUI provides a function to load these observers from a data file, as following



The data file is a text file in ASCII format, each line defines a observer.

The format of each line can be one of following:

(a) ***r phi z vr vphi vz trr tphiphi tzz trphi trz tphiz***
(b) ***r phi z vr vphi vz trr tphiphi tzz trphi trz tphiz "name"***

For the format (a), the observer in the file has not name, they will use the name generated by the GUI

For the format (b), the observer name is defined in the file. But it may conflict with existing source in the project, GUI will append "_1" or so to solve the problem

User can also add comment in the end of each line. The comment start from "//". For example,

- *r phi z vr vphi vz trr tphiphi tzz trphi trz tphiz* *// this is the comment*

- The value and the meaning for *vr vphi vz trr tphiphi tzz trphi trz tphiz*
  - Each input can be 0 or 1
    - 0 – do not record this field component
    - 1 - record this field component
  - Example for a observer in a BHA project
    - 1 30 1 1 1 1 0 0 0 0 0 0

Position=(r=1, phi=30°, z=1)

Record (vr, vphi, vz)

Do not record any Tau component

Note: if all components are not listed as recorded, the 1st field component will be set as recorded component

- Additionally, each line start from “//” or , ‘*’, or ‘#’ will be considered as a pure comment line and it will be skipped.
- Any empty line will be skipped automatically

- For the position (r, phi, z), user can define them by variable or an expression with variables
  - But each value must use a string without SPACE
  - Example
    - x ang+30 z 1 0 0 0 0 0 0 0 0
      - Here, 'x' is the variable to calculate the x (or r) value
      - "ang+30" is the expression to calculate the y (or phi)
      - 'z' is the variable to calculate the z value

Note: the definition uses variable "x", "ang" and "z". In order to make it work, the project should define these variables before loading. Otherwise, the load operation will fail.

- Example of the content of a file to define the observers for a BHA project

obv_demo.txt

```
1    30  1  1  1  1 0  0  0 0  0  0
1    45  1  0  0  0 1  0  0 0  0  0
2    30  1  0  0  0 1  0  0 0  0  0
0.5  0   3  0  0  0 0  0  1 0  0  0
```

Simplest version

Line #1 define a observer at(r=1, phi=30°, z=1); record (vr, vphi, vz); auto name

Line #3 define a observer at(r=2, phi=30°, z=1); record trr; auto name

Line #4 define a observer at(r=0.5, phi=0°, z=3); record tzz; auto name

obv_demo2.txt A complicated version

```
1 30  1  1  1  1 0  0  0 0  0  0      // define a observer at (r=1, phi=30°, z=1);
                                        // record (vr, vphi, vz); auto name

1 45  1  0  0  0 1  0  0 0  0  0 "obv1"   // define a observer at (r=1, phi=45°, z=1);
                                            // record trr; name="obv1"

2  30  1  0  0  0 1  0  0 0  0  0              //  define a observer at (r=2, phi=30°, z=1);
                                                // record trr; auto name

0.5  0   a+1  0  0  0 0  0  1 0  0  0  "obv2"  // define a observer at (r=2, phi=30°, z ="a+1");
                                                  // record tzz; name="src2"
```

The comment in the file

Here, the 4th observer uses an expression and the variable "a" to define the Z position. In order to make it work, the project should have defined the variable "a" before loading. Otherwise, the load operation will fail.

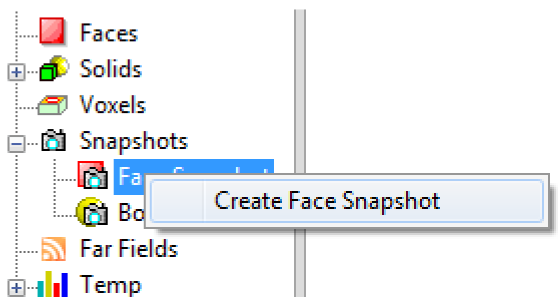
Following format is still work, it add ';' as the separator to make it more readable.

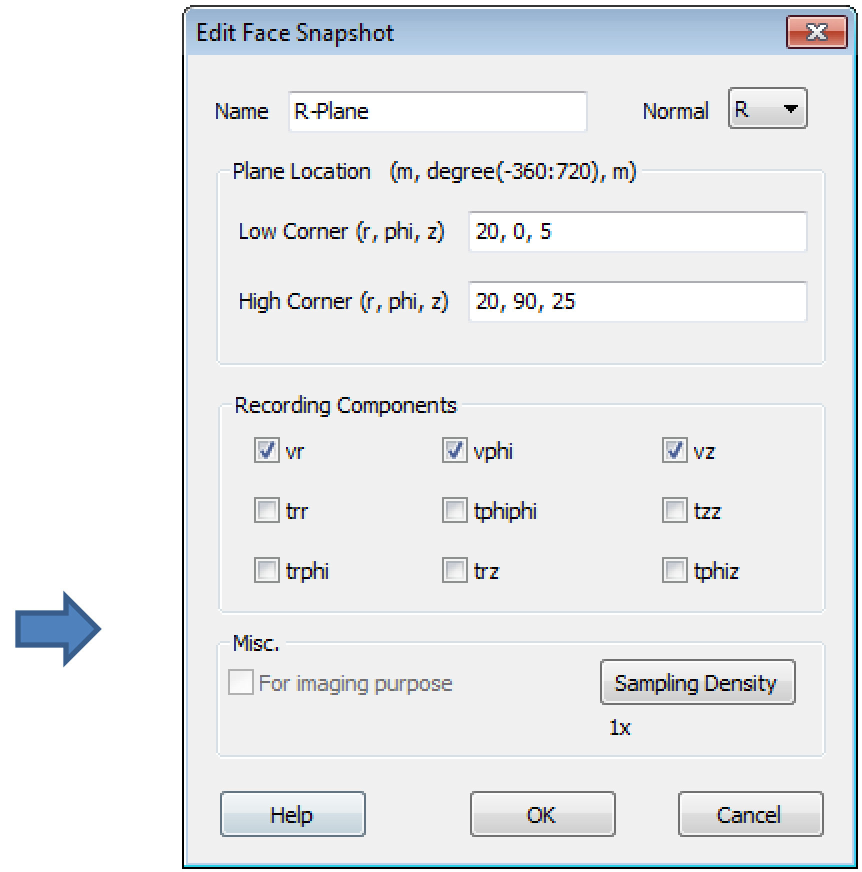
obv_demo3.txt

```
1;    30;  1;  1;  1;  1; 0;  0;  0; 0;  0;  0; "obv1"
1;    45;  1;  0;  0;  0; 1;  0;  0; 0;  0;  0; "obv2"
2;    30;  1;  0;  0;  0; 1;  0;  0; 0;  0;  0; "obv3"
0.5 ; 0;   3;  0;  0;  0; 0;  0;  1; 0;  0;  0; "obv4"
```

Snapshot

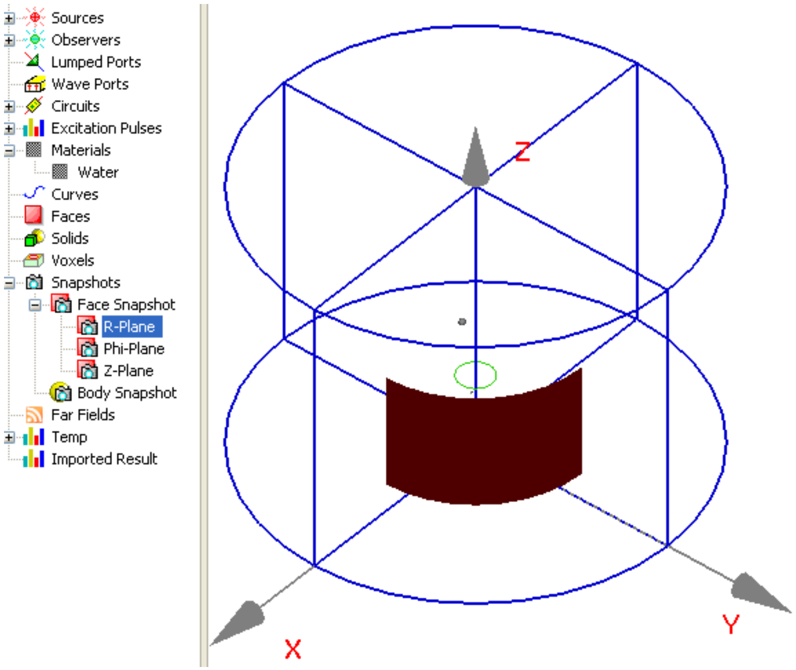
In this version, there are only **regular 2D face snapshot** is available for WCT BHA project. User can define ***R*** planes, ***Phi*** planes and ***Z*** planes to record
velocity: vr, vphi, vz and stress: trr, tphiphi, tzz, trphi, trz, tphiz





(Note: we suggest that do not record mixed field type in the same snapshot, i.e., don't record V field & Tau field in the same snapshot. For example, record V field in one snapshot, record Tau file in another snapshot.)

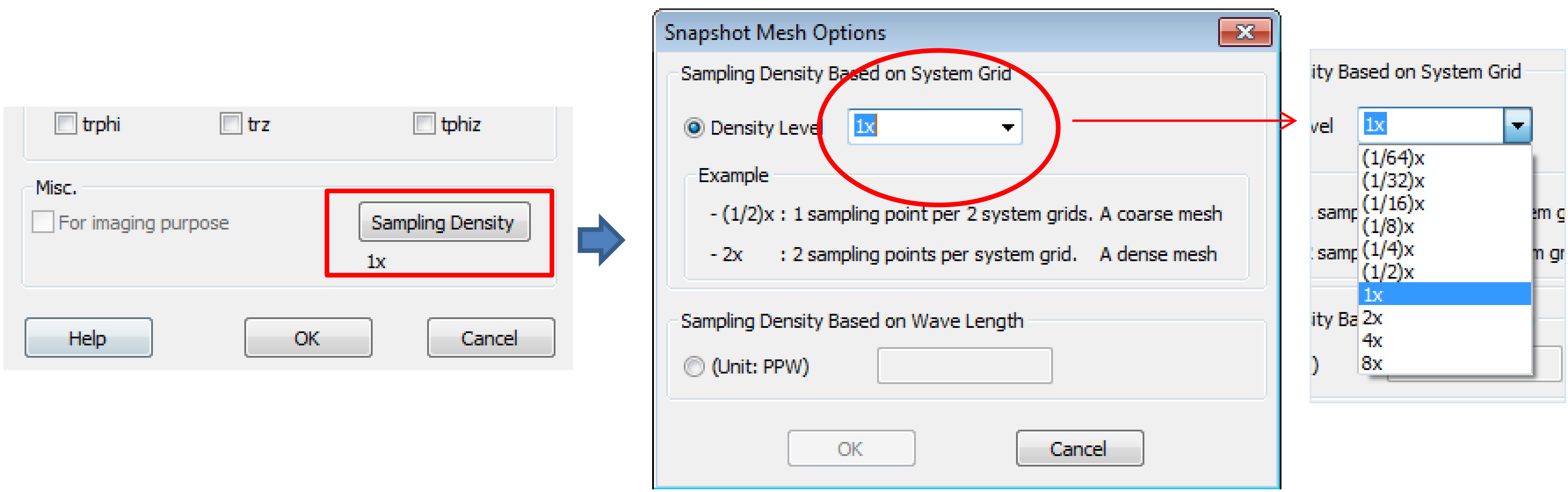
After defining a snapshot, use can highlight the treenode item to display the snapshot's shape in the project.



The Sampling Density of a Snapshot

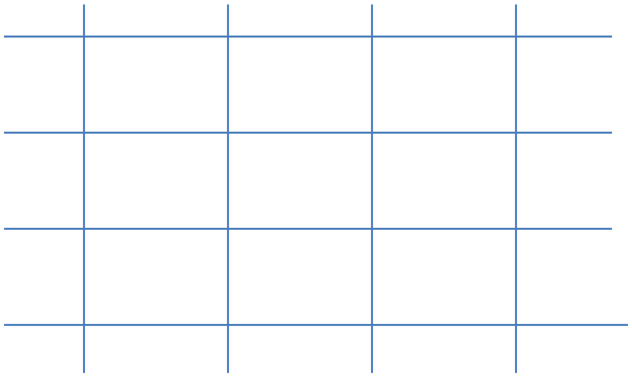
The default sampling positions of a snapshot is the simulation grid point.

User can choose a denser or coarser sampling density by the requirement, as following

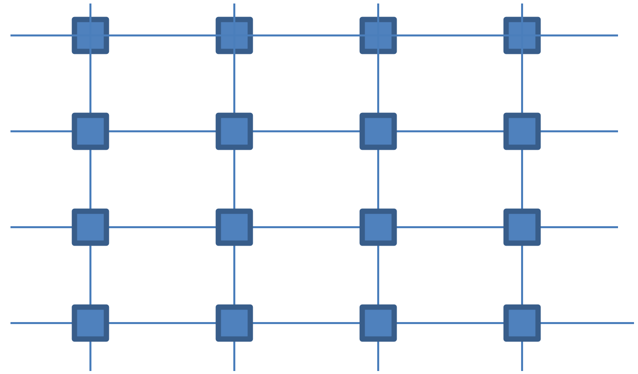


Following are the examples of the sampling density

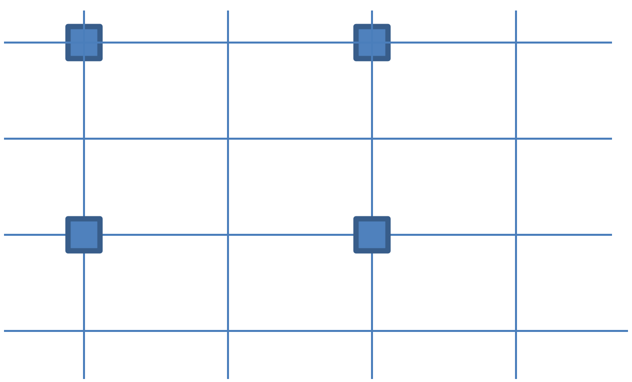
System grid



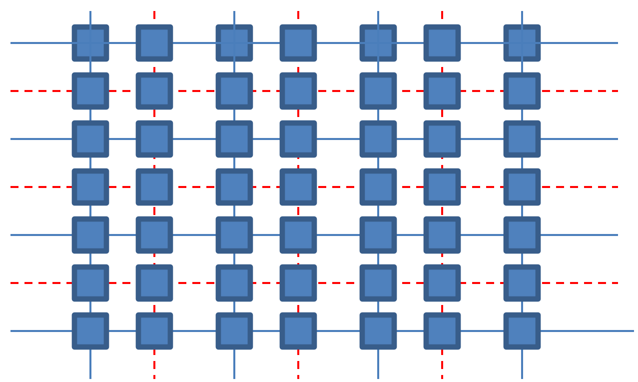
Default sampling density: 1x



Sampling density: (1/2)x

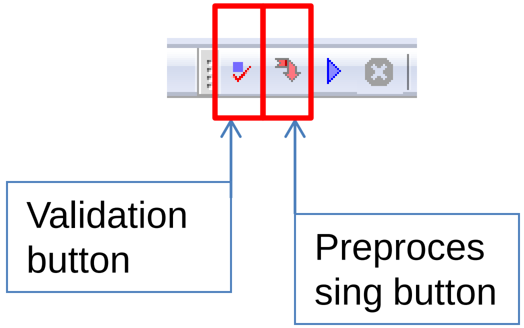


Sampling density: 2x



 the sampling point in the snapshot area

Project Validation & Preprocessing



Before a simulation, GUI will verify whether a project is set up correctly for the simulation, including

- boundary conditions
- solid position and clash test
- source, observer & snapshot position

The validation information will be shown in the log

- whether the project pass the validation
- if fail, the reasons

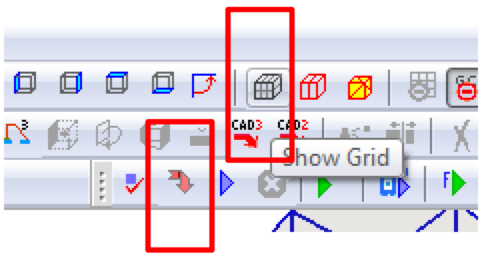
User can also make a preprocessing before the simulation to

- verify the project
- synchronize the mesh grid and obtain cells number in simulation
- calculate the Δt

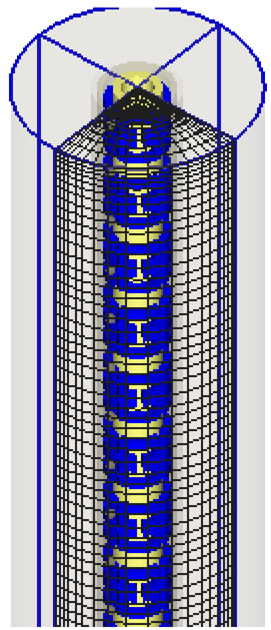
Grid Displaying

Before a simulation, user can use the grid displaying to check the mesh quality of this project. As following,

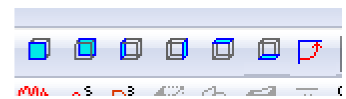
Press “Show/Hide Grid” button

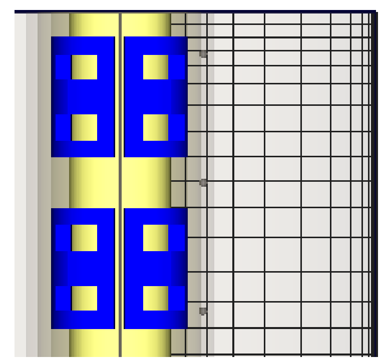


If the grid is not shown after using “Show/Hide” grid button, let GUI make a “Pre-processing” to generate grid. Then show grid.



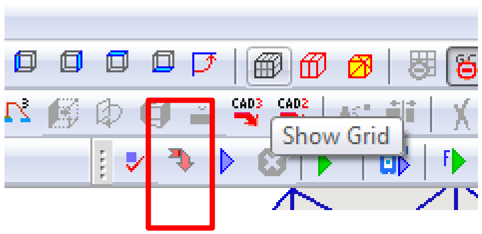
Switch to different View angle





Here, we can see that the Z grid does not capture the hole of solids

Note: to show a grid correctly, we strongly suggest to make a “Pre-process” before showing a grid. This is because the grid in GUI is not always synchronized with the change in solid.



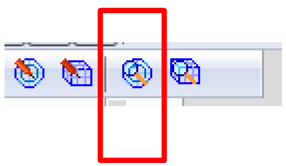
Pre-processing

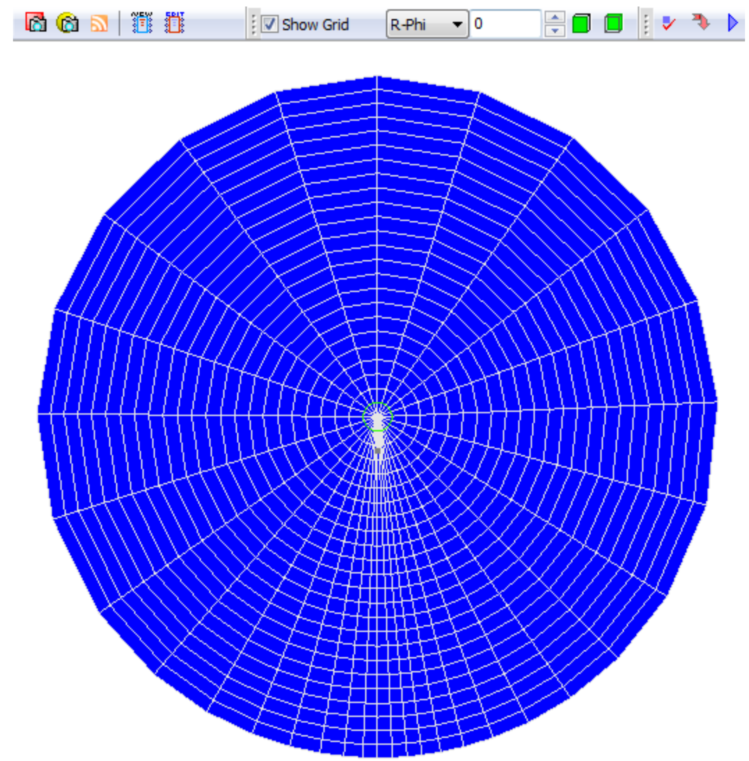
The pre-processing will verify whether the case setup, solids and other parameters are correct or not. It also reports the cells number, Δt etc. information, as following

```
Log
Preprocessing...
        Domains: 1 x 1 x 1,  Cells: 12 x 14 x 182,  Delta time: 2.84731e-007 sec,  Mesher version: 2,   CPU Time: 0.64 sec

Validating the design
......Cylindrical EL simulation.....
    ** Warning: clash testing for solid is skipped by project setting. Can't sure whether the mesh in engine is correct or not.
       It is suggested to double check solids before simulation.
            Body positions, face positions, layer positons, observer positons, user defined control points, source combination are verified.

Preprocessing...
        Domains: 1 x 1 x 1,  Cells: 20 x 20 x 260,  Delta time: 2.42195e-007 sec,  Mesher version: 2,   CPU Time: 0.655 sec
```

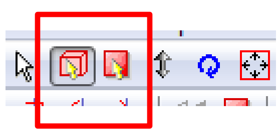
Mesh checking

Before a simulation, user can also use mesh viewer to check the mesh quality of this project. As following,



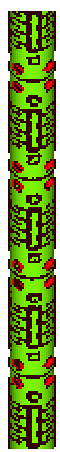


Use “Select Body” or “Select Face” to quit mesh viewer.

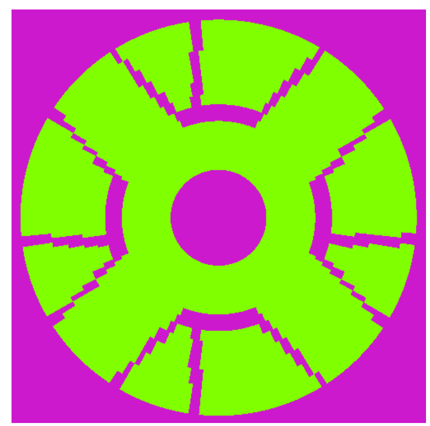


We strongly suggest user to check the quality of mesh for a complicated project.

For example, following sonic logging device, there are more than 100 parts and there are many slots in the solids. It is better to check whether the slot is captured correctly in the mesh.



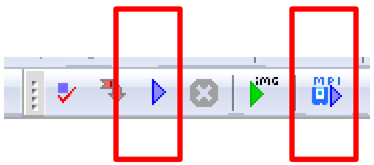
A cross-section of mesh data



Simulation & Tools

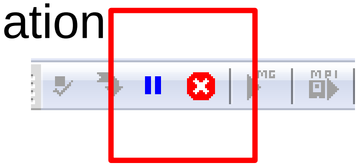
Simulation

User can use “Start” button to start the simulation

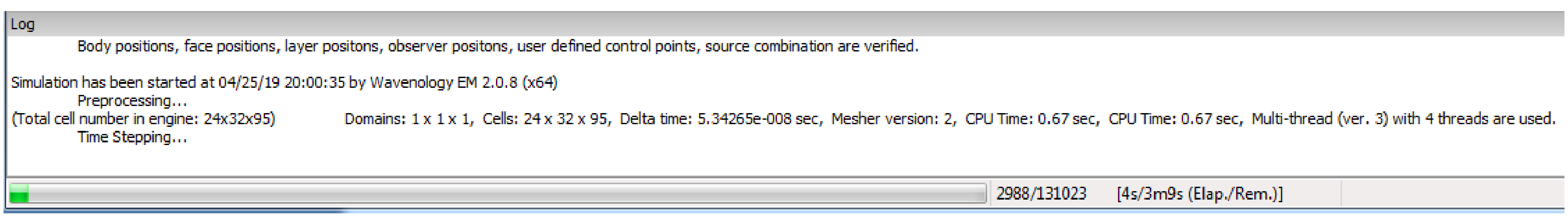


If the user has a WCT BHA HPC solver, this button can be used to generated the data files for WCT BHA HPC solver

After a simulation start, user can “Pause” or “Stop” this simulation

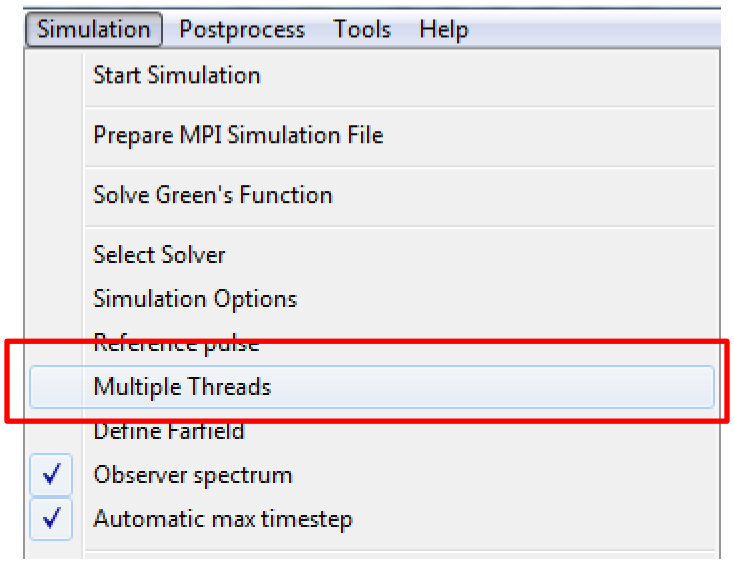


The simulation information and progress will be shown in the log and the status bar.

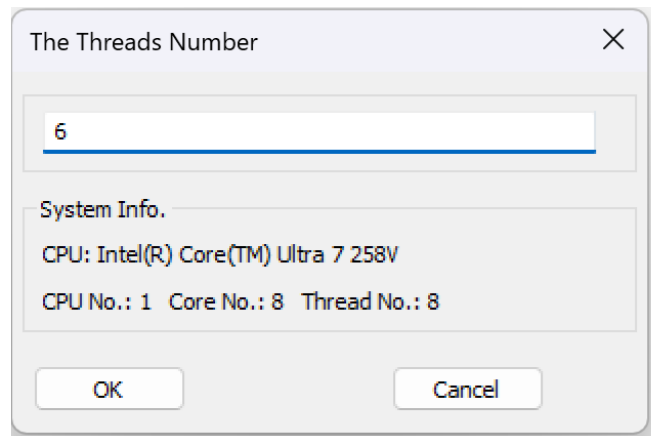


Multi-threading Control

WCT GUI employs multi-threading to speed up the simulation, the default thread number will be determined by GUI. In general, this number can provide the best parallel efficiency, but user can change this thread number to other values.



Define the thread number



Note: WCT GUI will automatically detect the physical core number of CPU(s) and determine the optimized thread number based on the physical core number.

From WCT GUI v2.2.2, the WCT Cylindrical Elastic Wave solver will split the whole domain into pseudo sub-domains with fine granularity. So, even the domain with 1 single ϕ plane can be simulated by multiple threads.

WCT GUI employs a dynamic load balancing method to re-allocate the load for each thread in the simulation. Therefore, the simulation can get a stable and high CPU usage.

Note: to obtain the best parallel efficiency, the number of threads depends on the hardware.

According to our experience, for a single CPU system, with 2 memory channels, 4 threads is enough; with 4 memory channels, 6 threads is enough.
User can use 1 or 2 more threads than above number to reduce the simulation time a little bit, but the parallel efficiency will become lower.

Following is our test based on different systems for a case with 400x200x2000 cells.

System with 2 memory channels

CPU	Memory		Thread No. and Simulation Time (minutes)							
	channels	type	1	2	3	4	5	6	7	8
I9-10900 (10 cores)	2	ddr4-3000	1122	669	534	504	496	471		
I9-10900 (10 cores)	2	ddr4-3200	1076	608	500	473	468	464		

System with 4 memory channels

CPU	Memory		Thread No. and Simulation Time (minutes)							
	channels	type	1	2	3	4	5	6	7	8
I7-5960x (8 cores)	4	ddr4-2400	1613	828	594	460	393	365	349	353
I7-5960x (8 cores)	4	ddr4-2666	1598	830	589	454	384	345	329	337

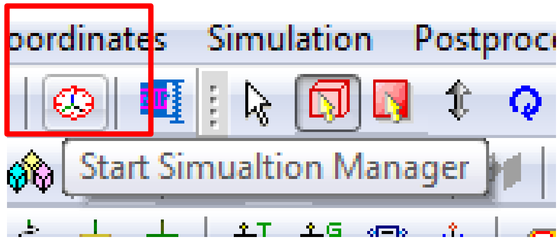
As can be seen, for a system with 4 memory channels, the speedup factor can be up to 5; but with 2 memory channels, the speedup factor will be around 2.5. Therefore, if there are enough cores in CPU, the system with 4 memory channels can obtain double speedup compared to the system with 2 memory channels.

Batch Simulation through the Simulation Manager

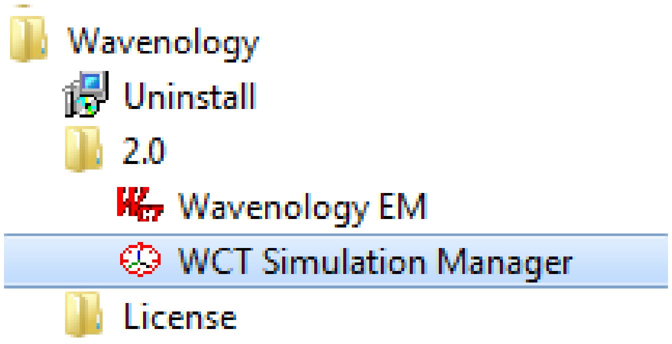
User can simulate multiple projects sequentially through WCT simulation manager

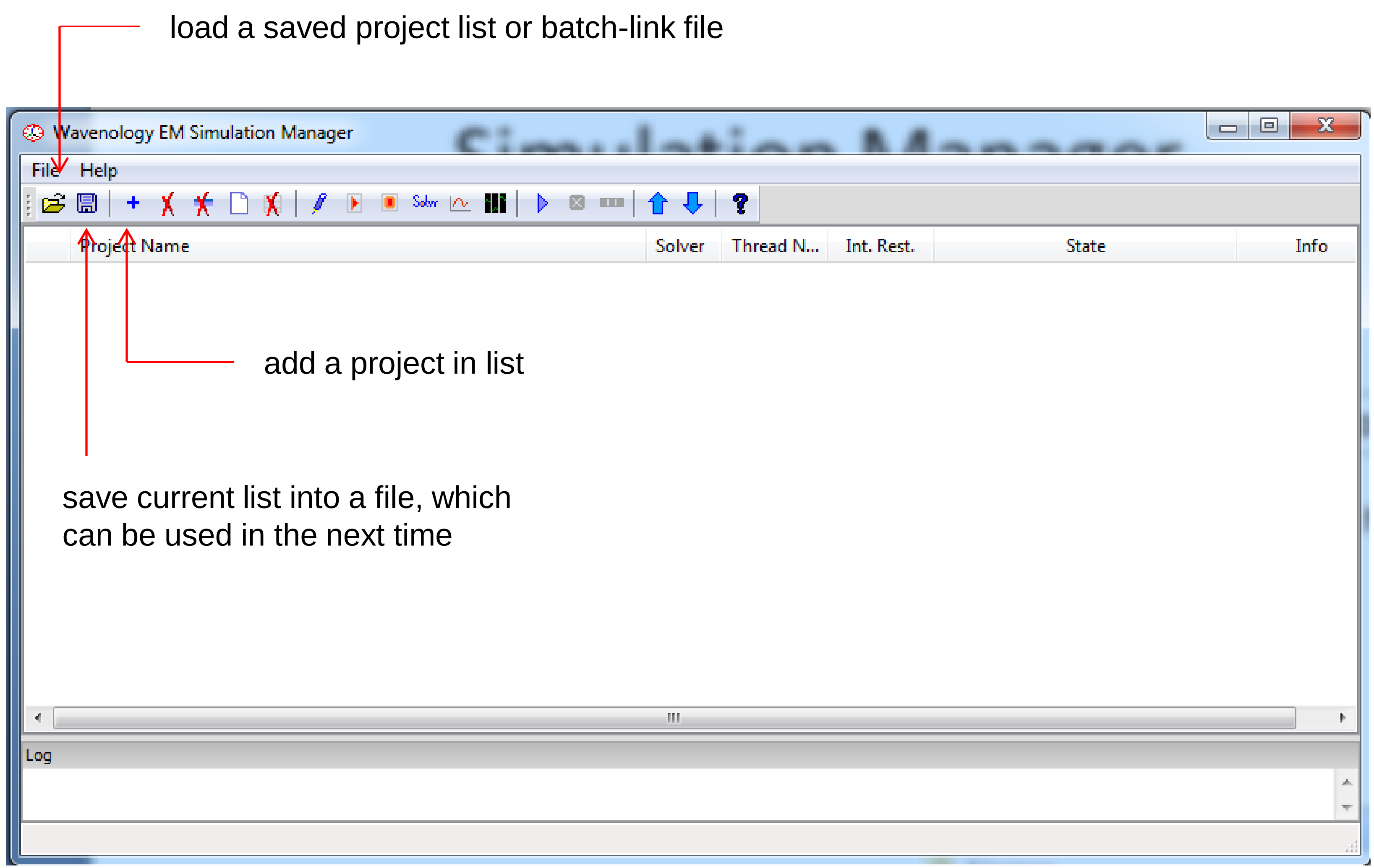
First, a or several WCT Cylindrical Elastic Projects need to be built and stored.

In WCT GUI, use this button to start the simulation manager

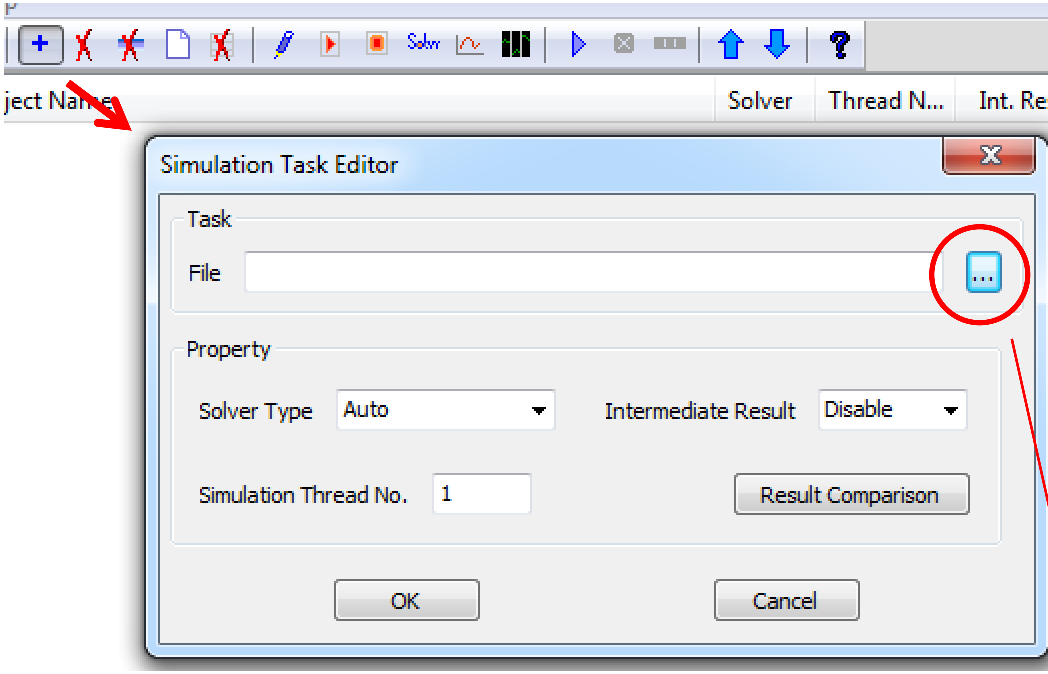


Or, in Windows menu, expand here to start the simulation manager

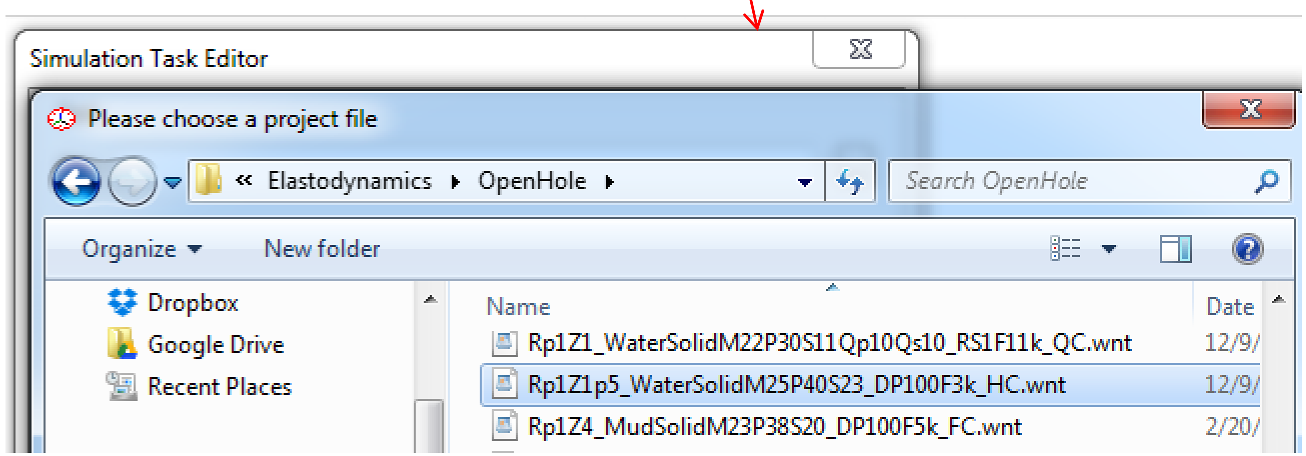



load a saved project list or batch-link file
Wavenology EM Simulation Manager
File
Help
Project Name
Solver
Thread N...
Int. Rest.
State
Info
add a project in list
save current list into a file, which
can be used in the next time
Log

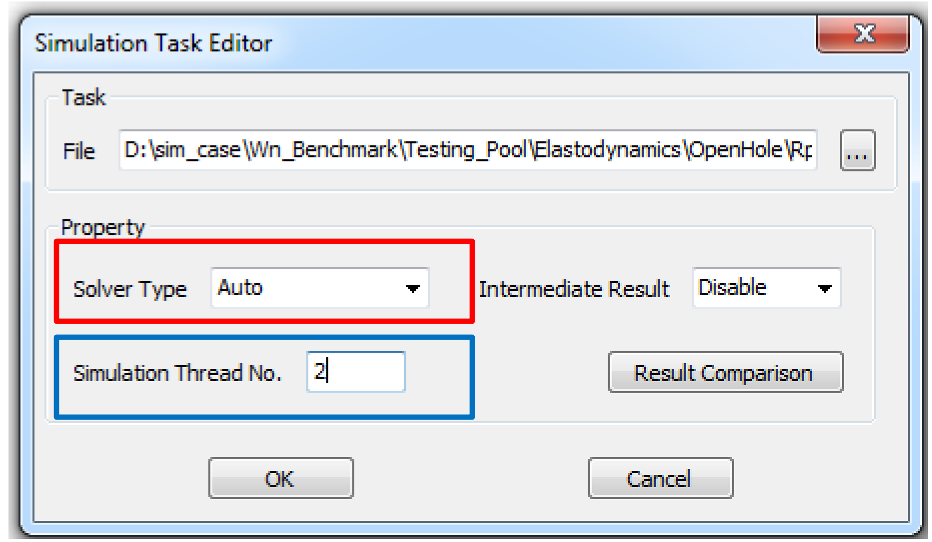
Add a single Project into List



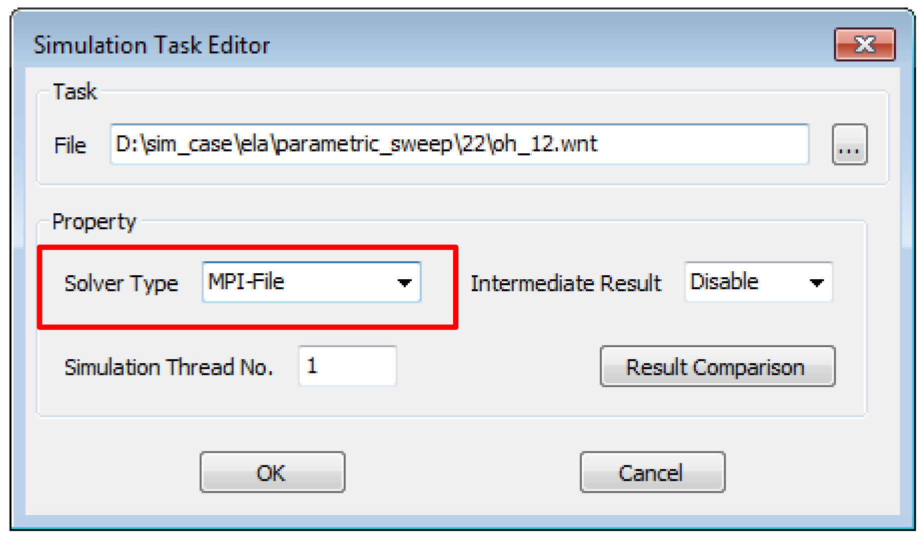
select project file



For a BHA project for **simulation purpose**, set solver type as "Auto" and user and define the thread number in the simulation.



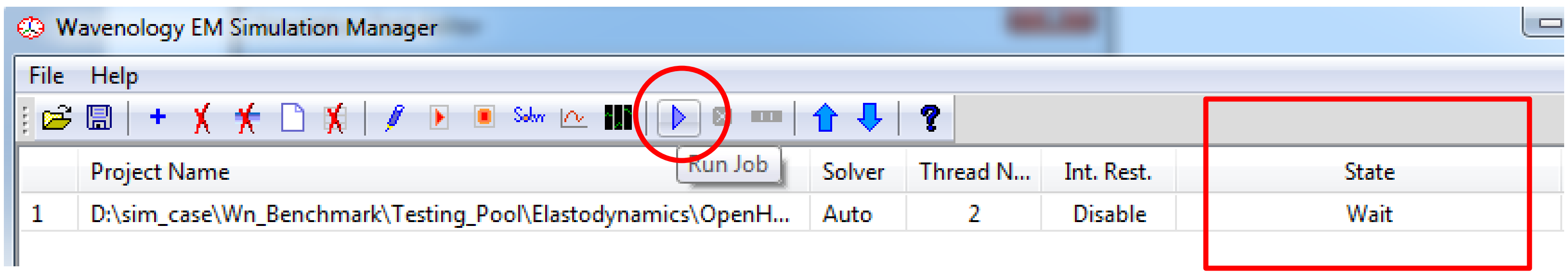
For a BHA project for **generating HPC project files purpose**, set solver type as "MPI-File".



After all projects are loaded in the list, user can start simulations

Click here to run the project in the list

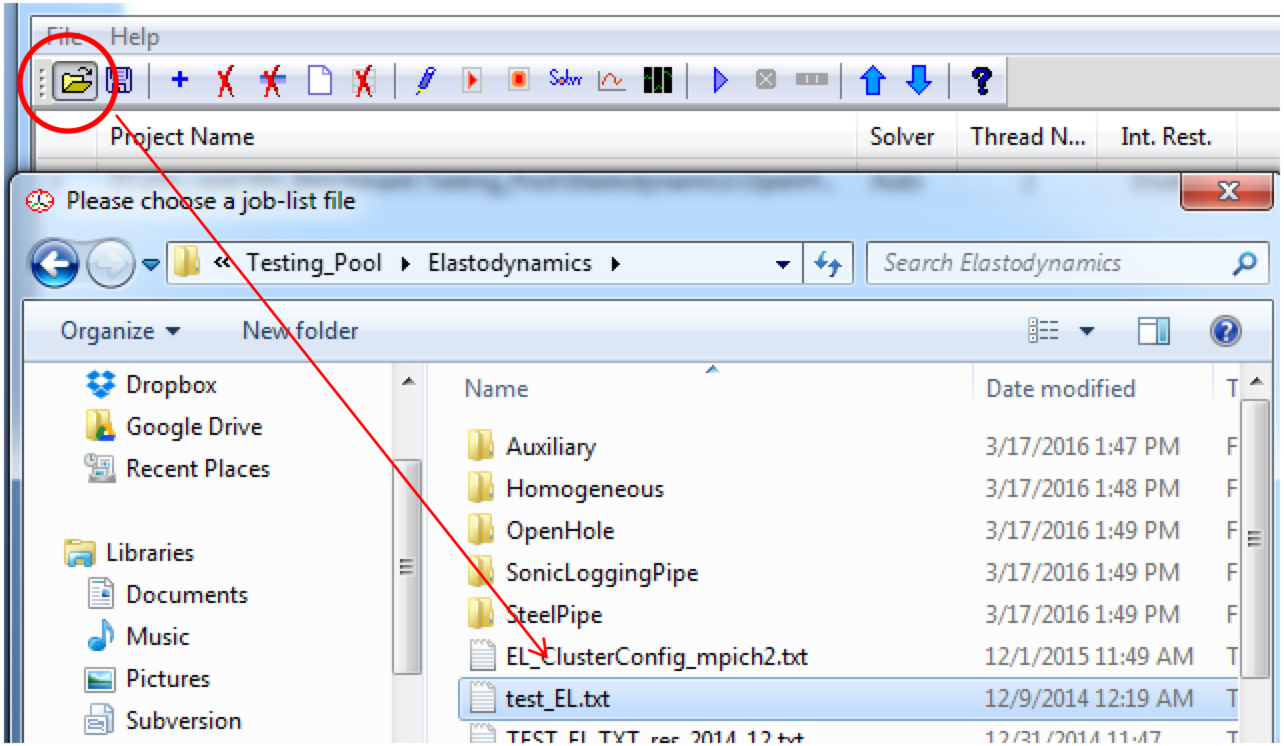
The project status is listed here to shown whether it is finished or not



Load projects through a Link File

If user have multiple projects to simulate and they are in different folders, user can use a link file to include these projects.

Then load this link file into the WCT simulation manager



WCT Simulation Link file format

Test_EL.txt

```
.include Homogeneous\test_Homogeneous.txt
.include OpenHole\test_OpenHole.txt
.include SteelPipe\test_SteelPipe.txt
.include SonicLoggingPipe\test_SonicLoggingPipe.txt
#########
```

comment line
start from '#'

test_Homogeneous.txt

```
"Rp1Z1_WaterSolidM22P30S11_RS0F3k_FC.wnt" SIMTYPE=auto  SIMTHREAD=1 INTRESULT=0
STATUS=wait
"Rp1Z1_WaterSolidM22P30S11_RS0F3k_HC.wnt" SIMTYPE=auto  SIMTHREAD=1 INTRESULT=0
STATUS=wait
"Rp1Z1_WaterSolidM22P30S11_RS0F3k_QC.wnt" SIMTYPE=auto  SIMTHREAD=1 INTRESULT=0
STATUS=wait
```

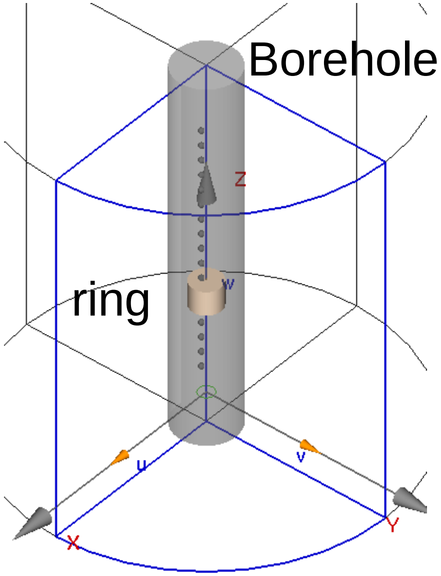
Parametric Sweeping

As mentioned in the section **Variable System**, the geometry or system parameter can be defined by variables. Therefore,

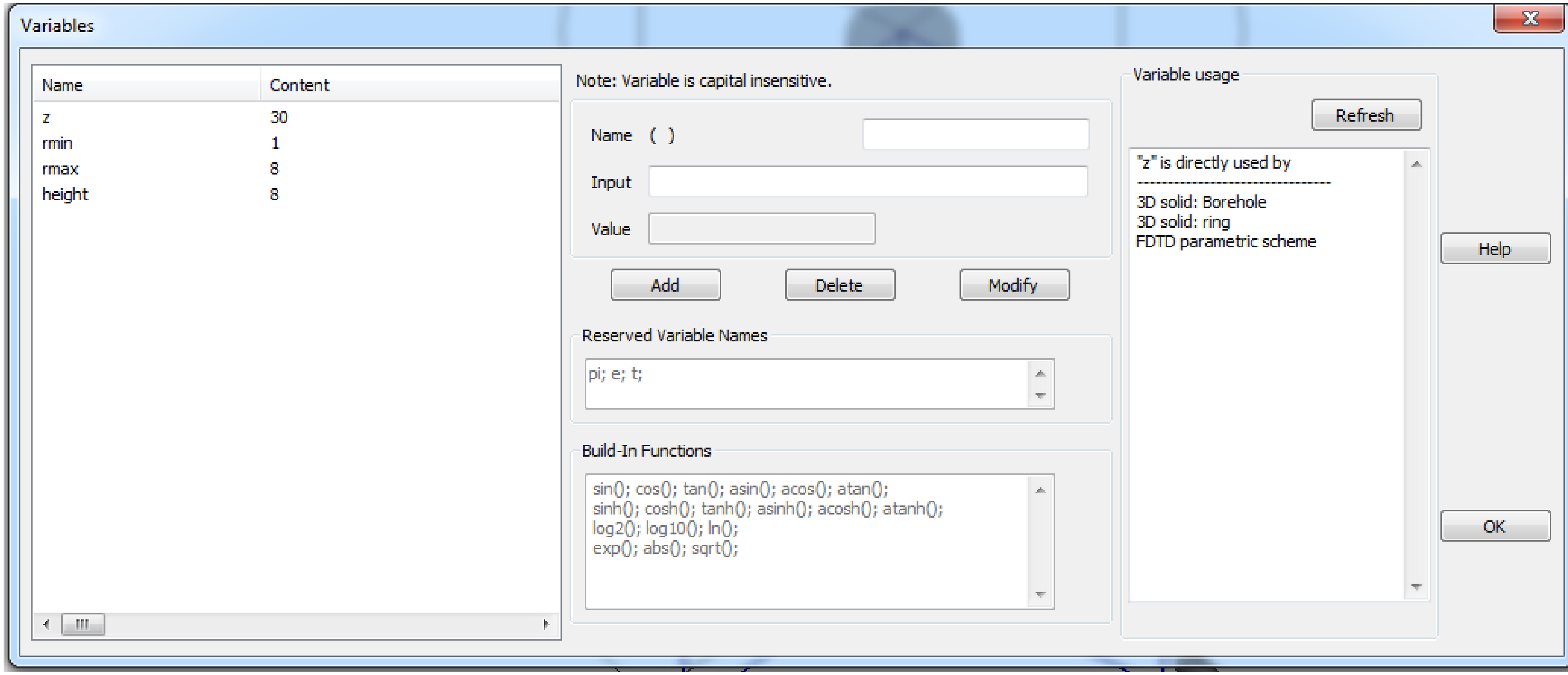
➢ user can sweep the variable in a range to design the geometries dimensions to obtain a desired performance.

➢ or, user can sweep source and receiver array in different positions of a system to investigate the response.

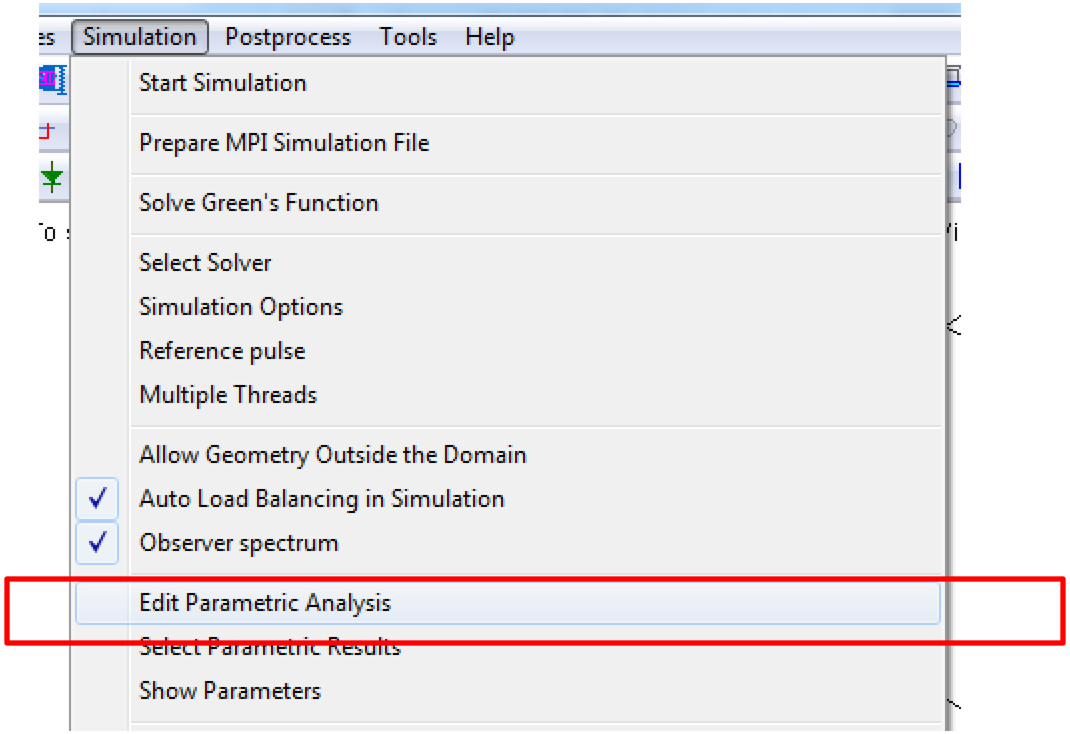
Following is an example to sweep 2 variables.



In this project, “Borehole” and inner object “ring” both use variables “z” and “height”.

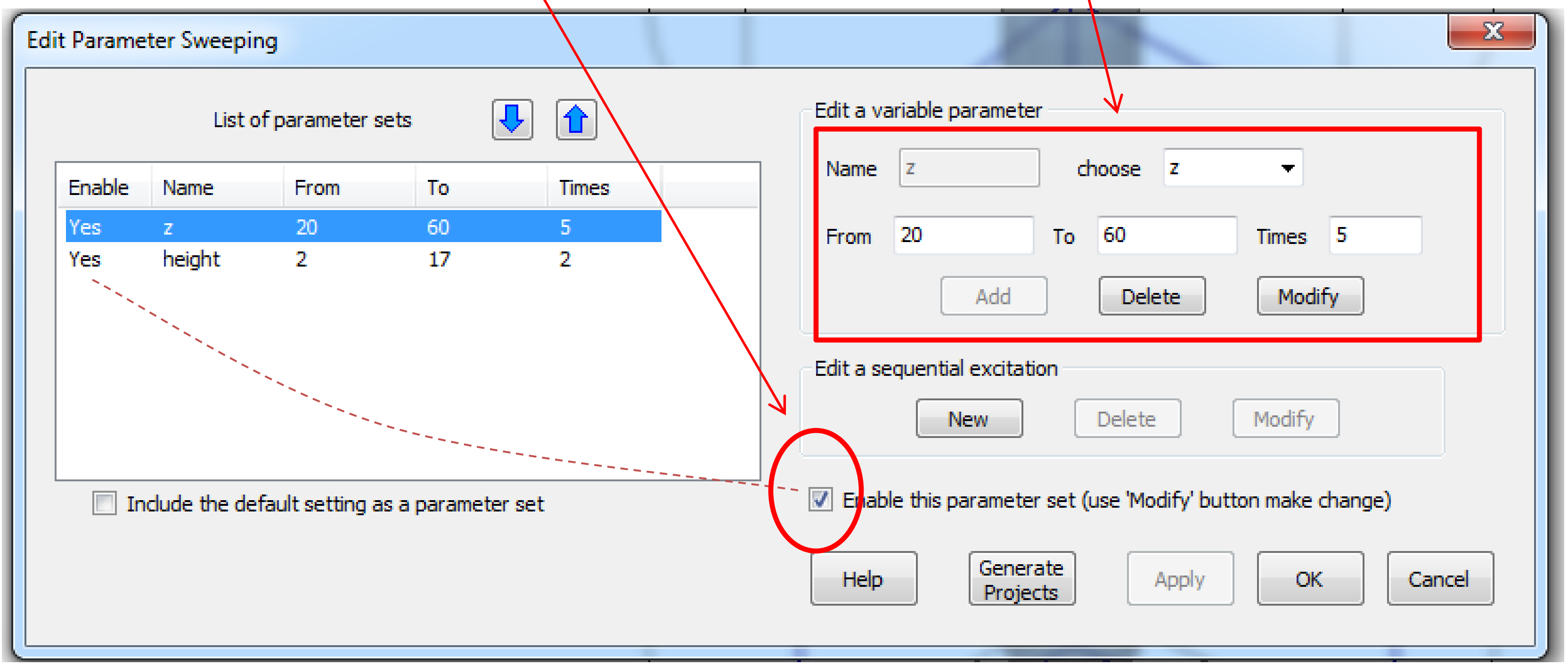


Use “**Edit Parametric Analysis**” to define which parameters will be swept.



Add/remove the variable from sweeping

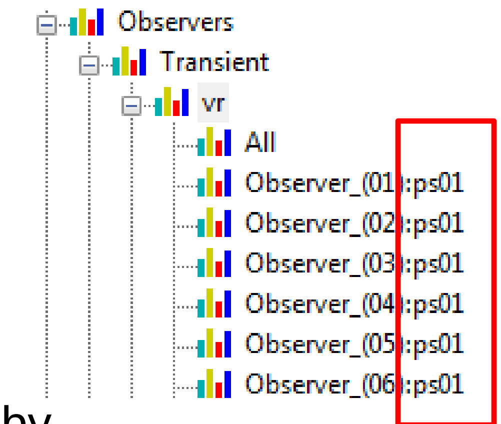
define the sweeping range for a variable



Options:

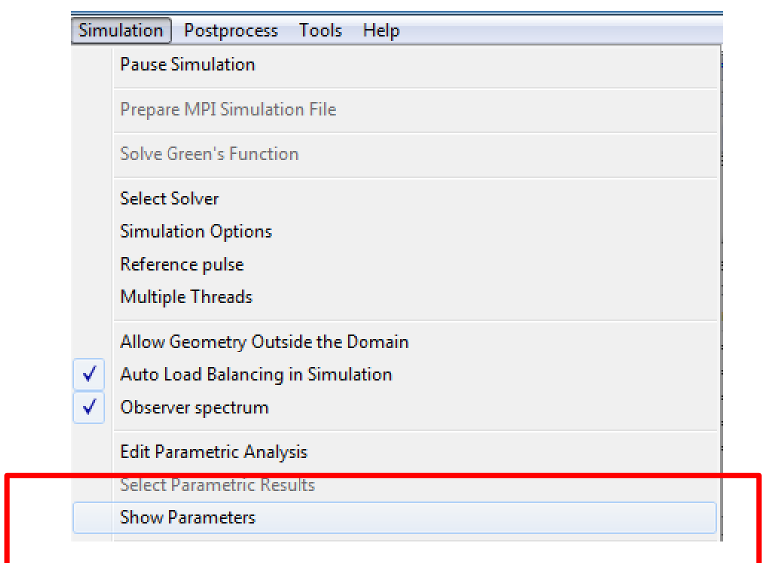
- ☐ Enable this parameter set
  - → Enable or Disable the parameter when parametric analysis.
- ☐ Include the default setting as a parameter set
  - → Add the default parameter set to the last parameter set.

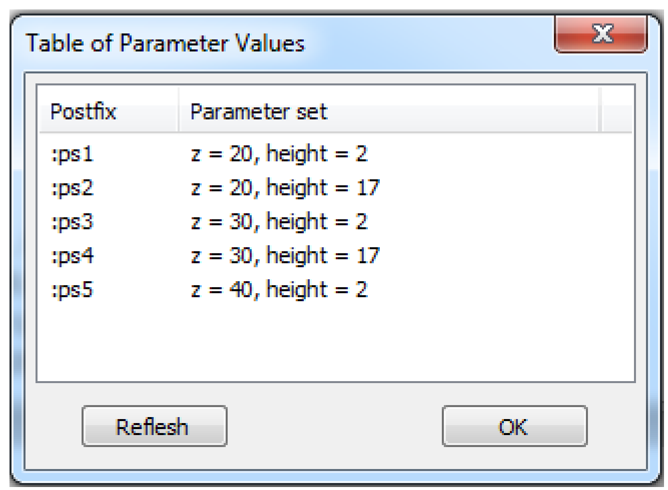
If there are sweeping parameters defined in a project, the simulation will be automatically switched to the sweeping mode



Each defined value in the sweeping range will be applied to the project and a corresponding simulation will be run.

The final results will be shown with suffix: psxx. The corresponding variable value for each psxx can be referred by

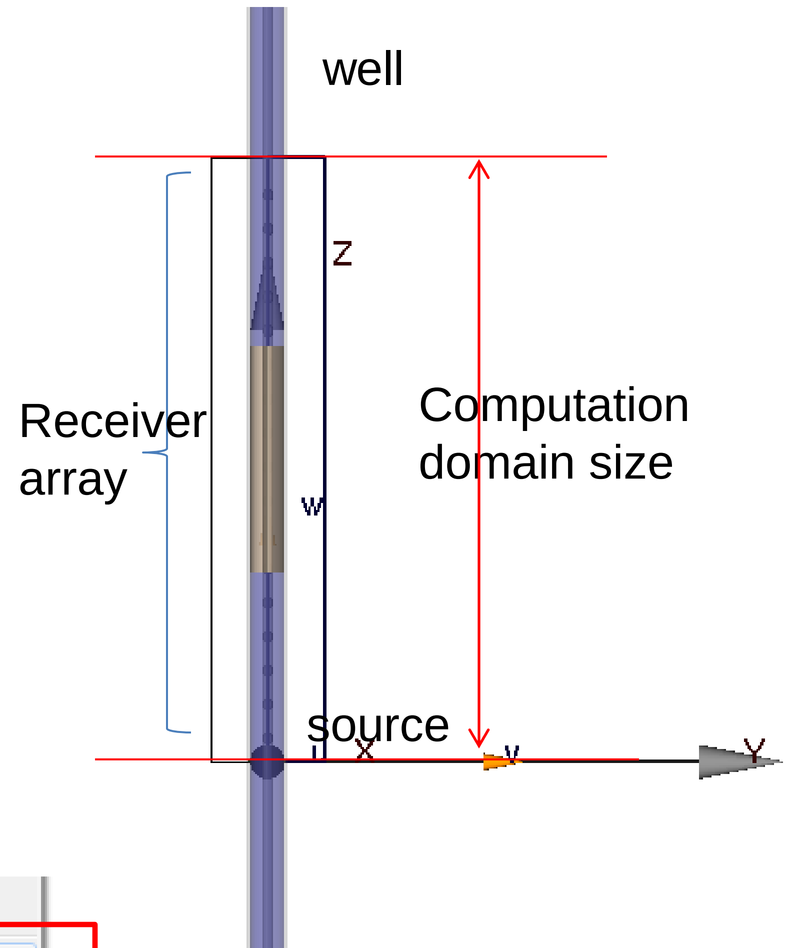




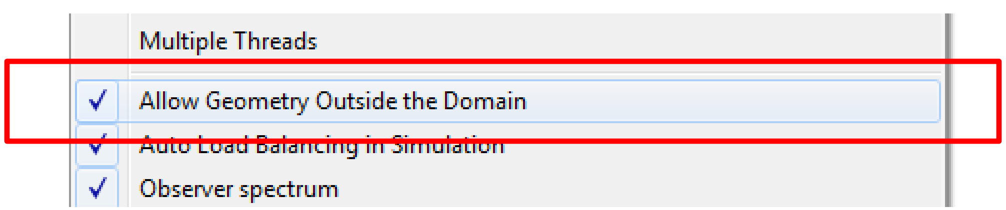
Example of sweeping along a long well to investigate the response at different locations

Here, a long well is defined.

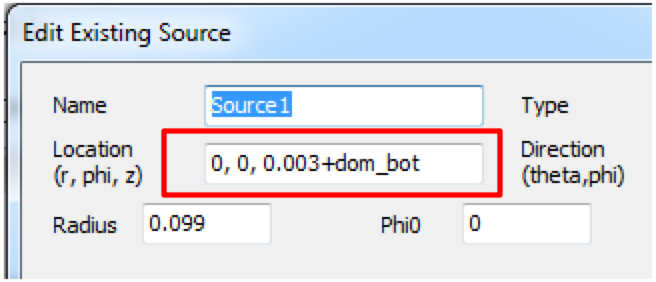
- ➢ A source and array of receiver will be swept along the well to check the response at different location of the well.
- ➢ We know that the wave do not propagate for a long distance. So, define the computation domain enclosing the source and the receiver will be enough.

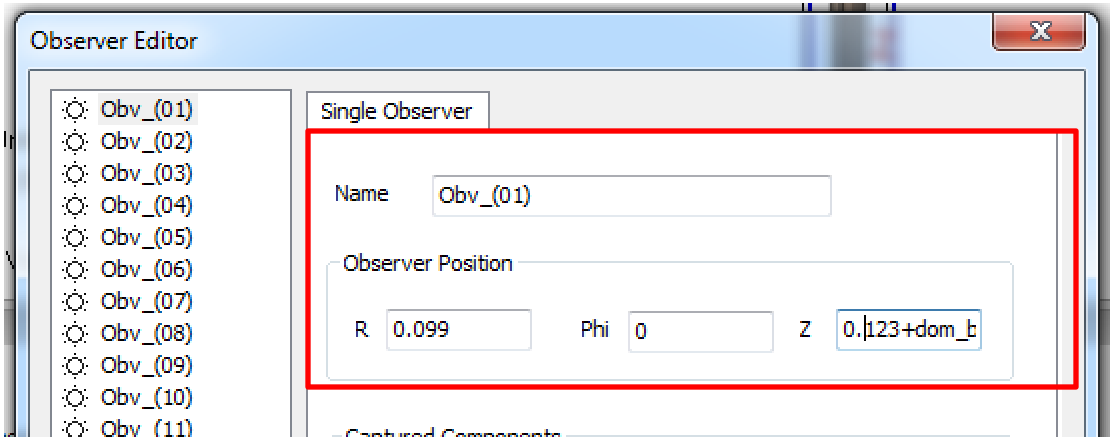


For this case, we need to allow geometry larger than the computation domain

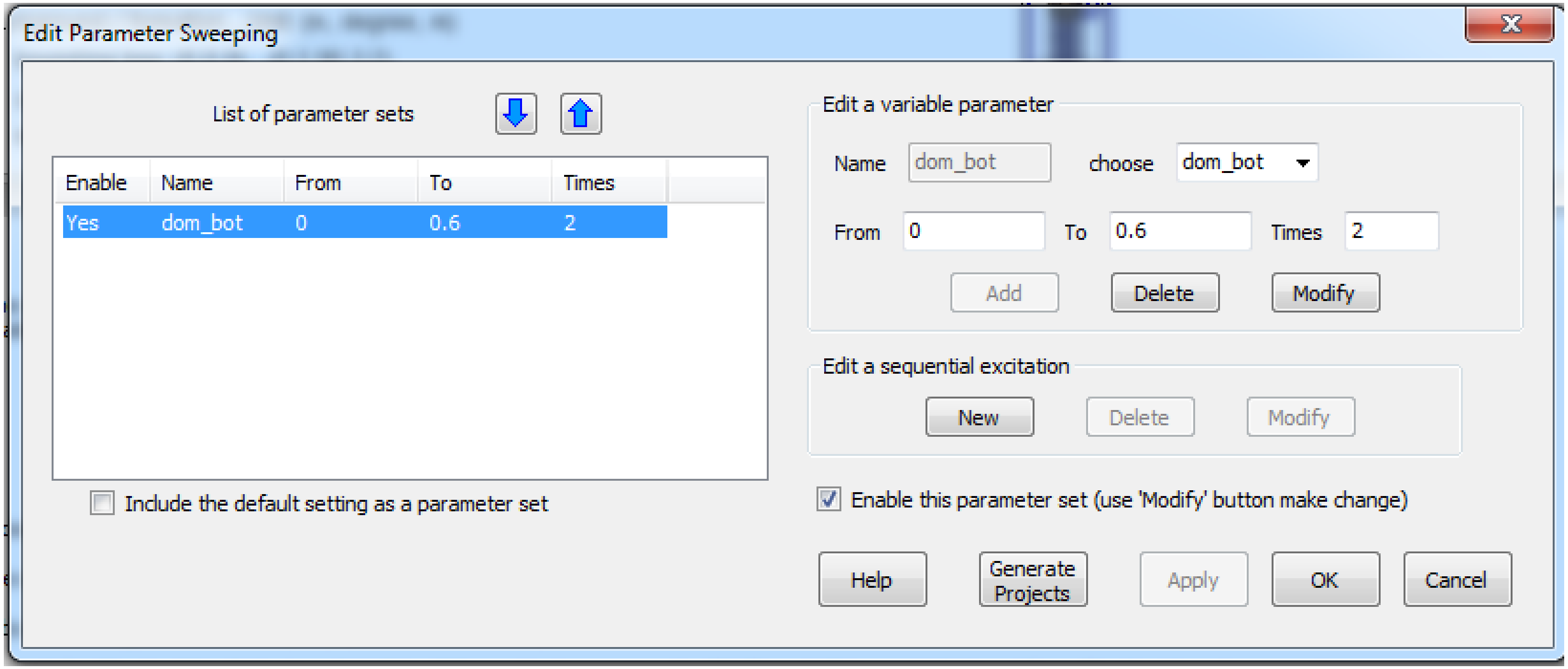


Meanwhile, the source and receiver positions depend on the variable “dom_bot”





We define the sweeping on variable as following and start the simulation.

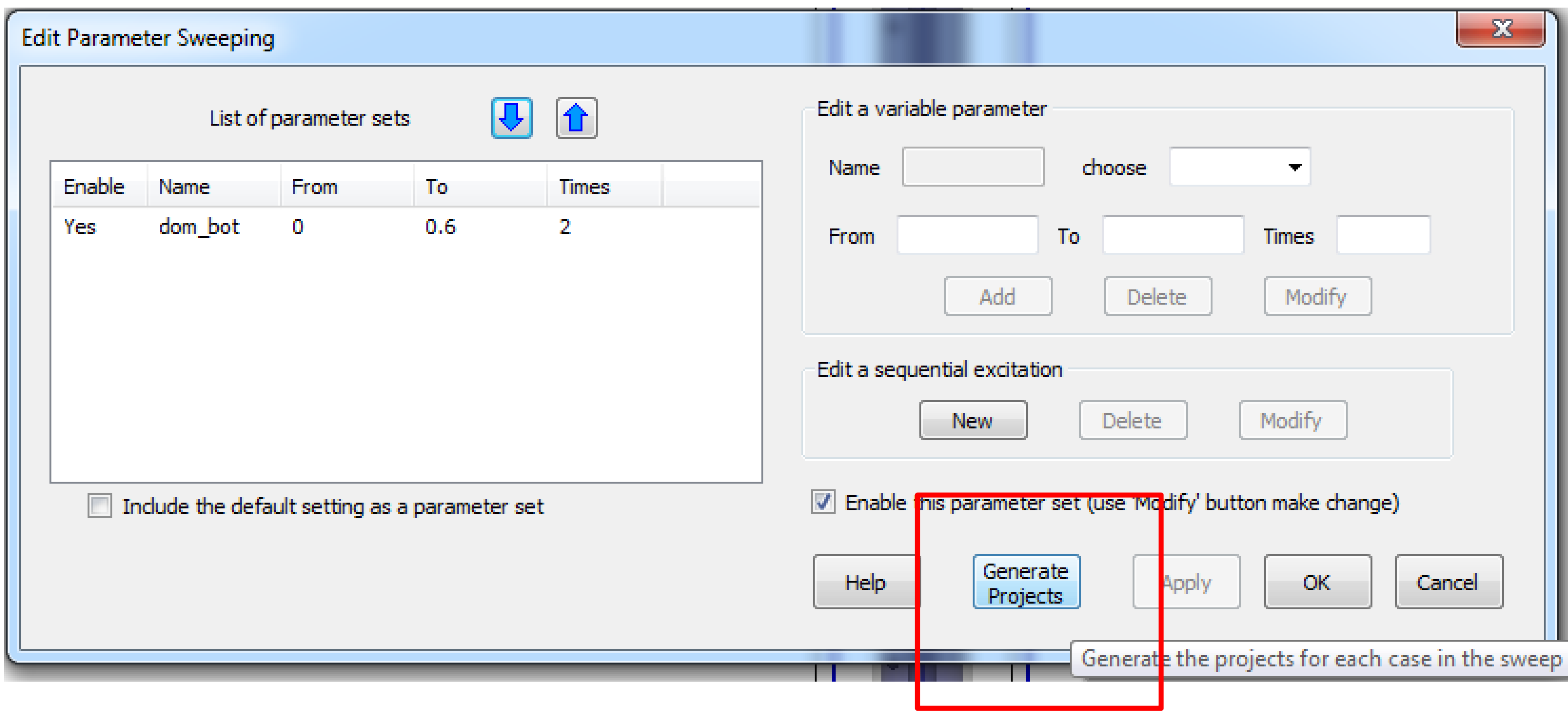


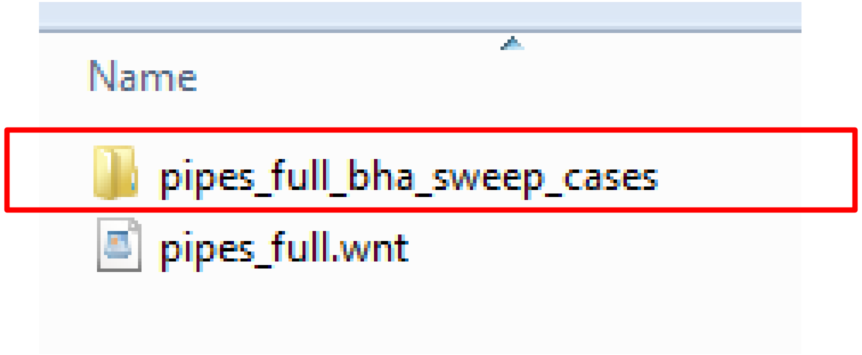
With this method, user can design different sweeping schemes for different purposes with a single geometry layout. It can significantly reduce the work for setup the project.

Generate Batch Projects from Parametric Sweeping

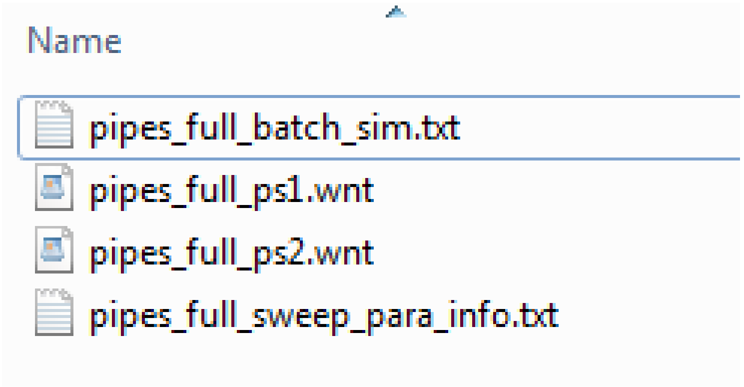
The parametric sweeping in GUI does not support snapshot to avoid huge storage space. However, for some cases, the snapshot in the parametric sweeping is required.

Following functionality can generate batch projects with specified parameters. If the project has snapshot, the snapshot definition will be kept in the batch projects.





The patch projects are stored in the sub-folder “***project_name***_bha_sweep_case”



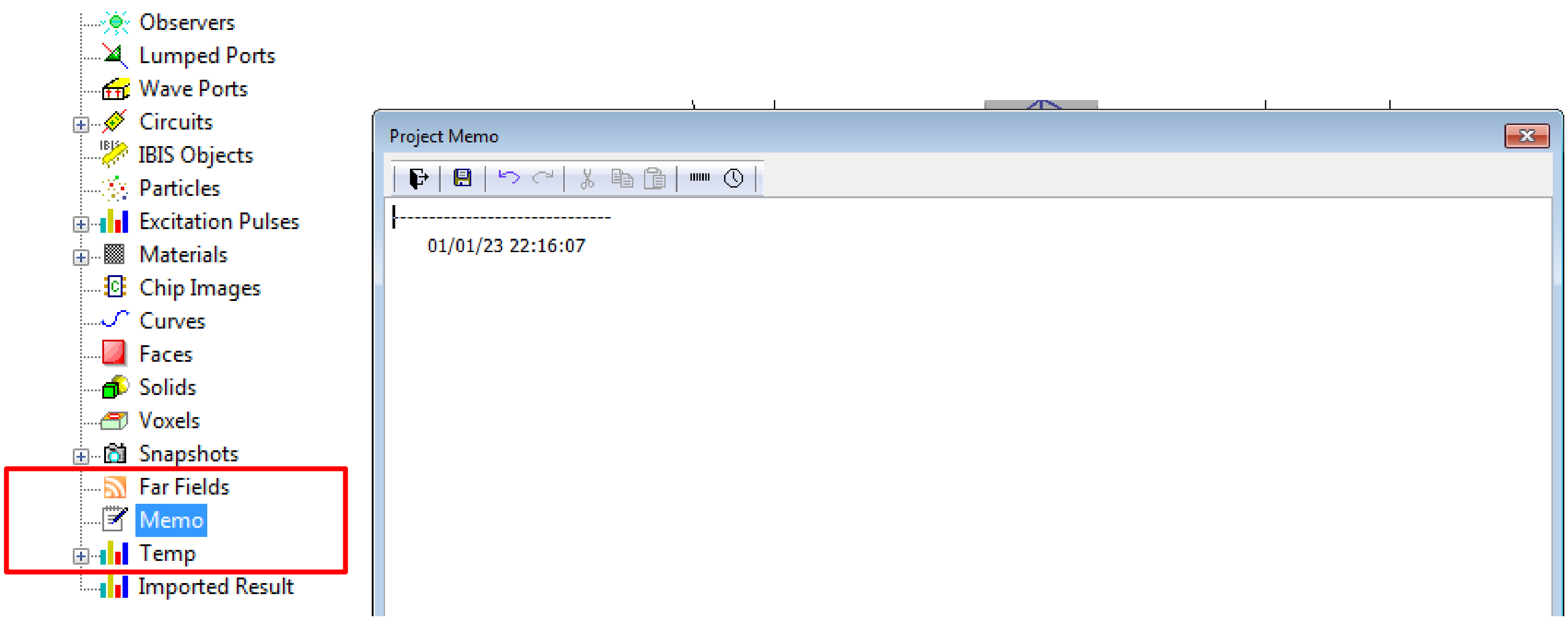
The files in the sub-folder

The file “*pipes_full_batch_sim.txt*” is a batch simulation file, which is supported by the WCT simulation manager .

The “*pipes_full_sweep_para_info.txt*” has the values for the parameters for each sweep.

Project Memo

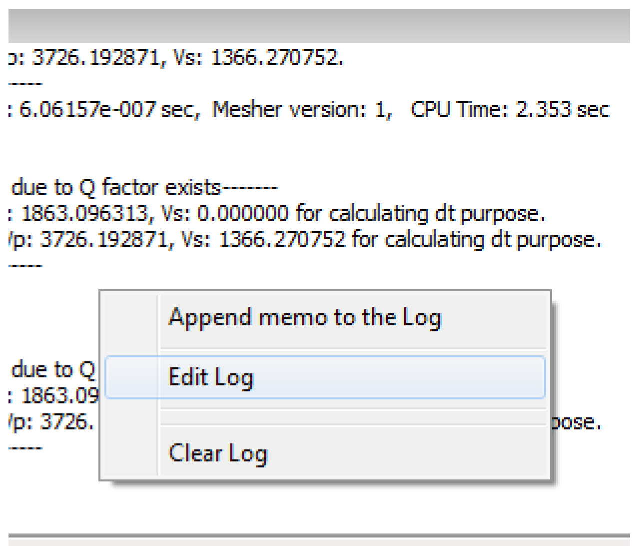
- From version v2.2.4, built in Dec., 2022, user can add memo for the project as following



Project Log Editing

- From version v2.2.4, built in Dec., 2022, user can edit project log to add or modify the information for each simulation

In the log region, right click mouse to popup following menu



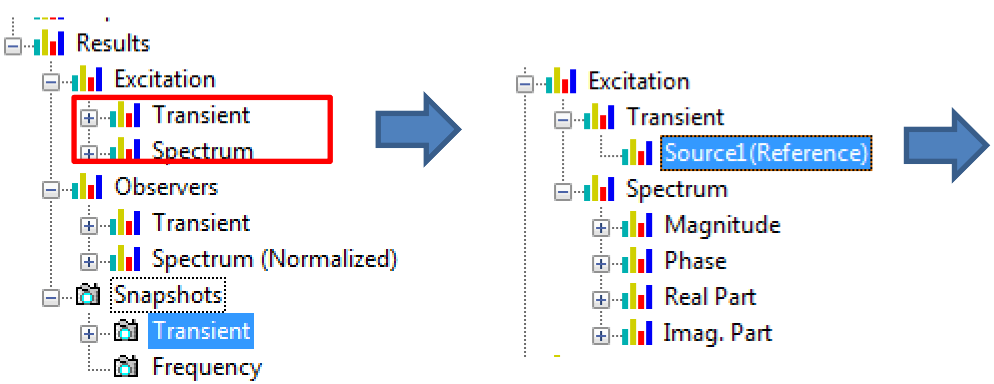
Result Displaying

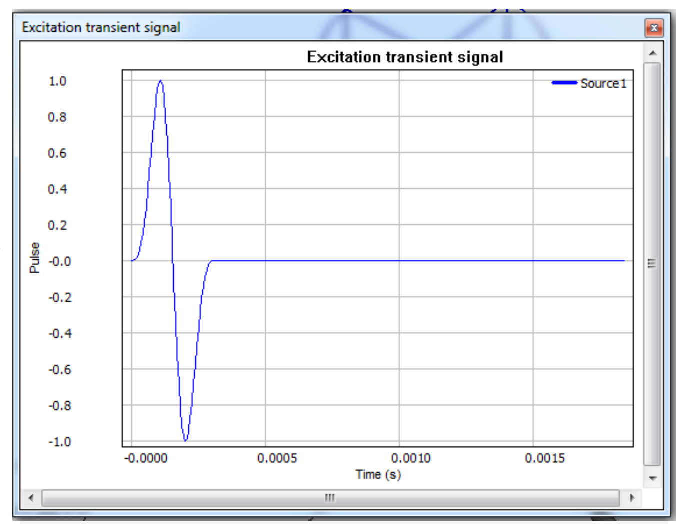
After simulation finish, the result will be stored and viewed as transient waveform or spectrum distribution. It includes

1) The excitation pulse: transient pulse & spectrum

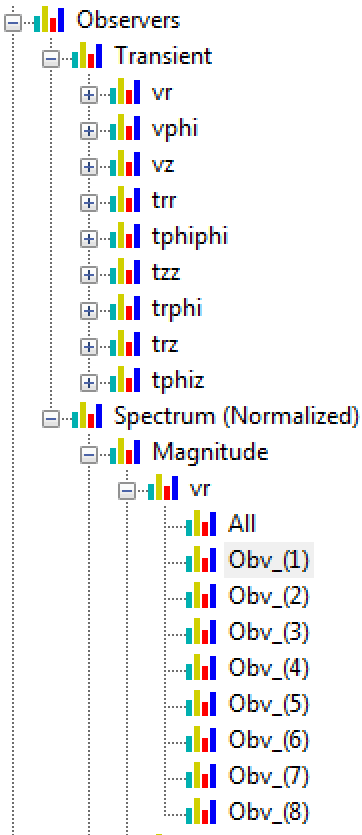
2) The field on observer: transient signal & normalized spectrum

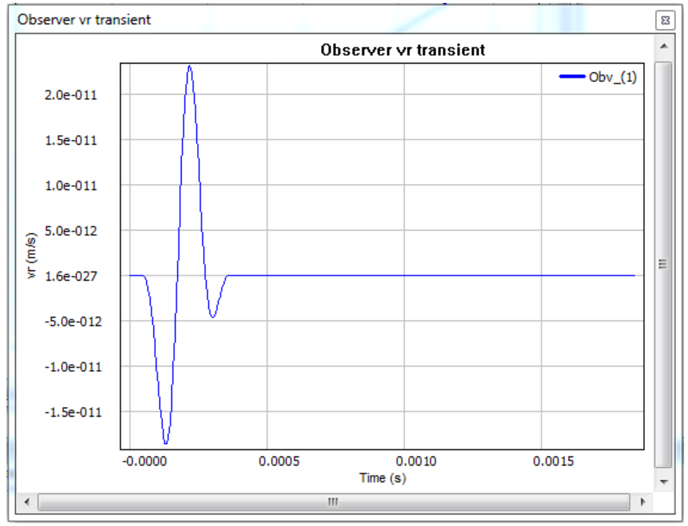
3) The field on snapshot: transient signal

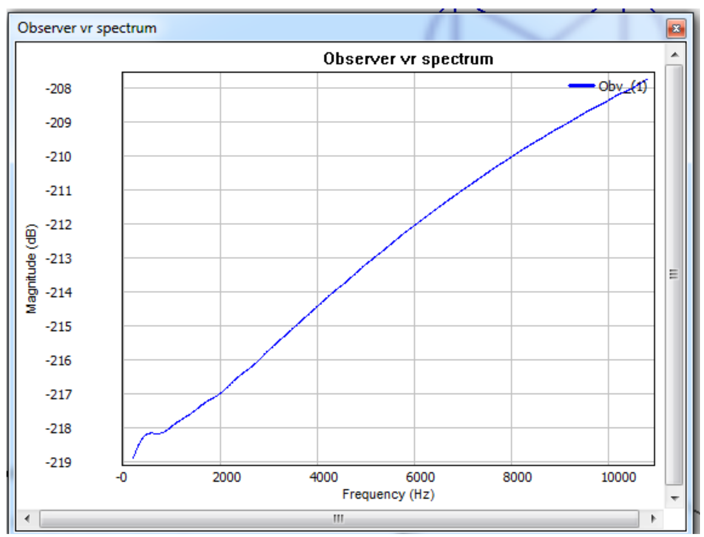




All registered components at observer are listed.

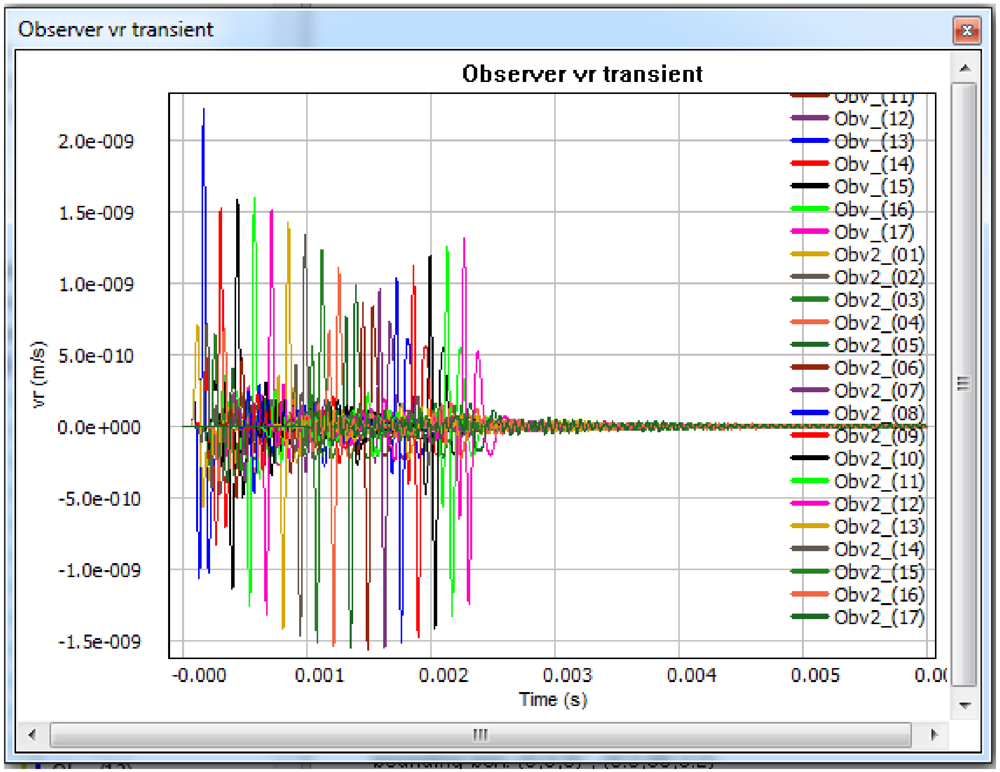






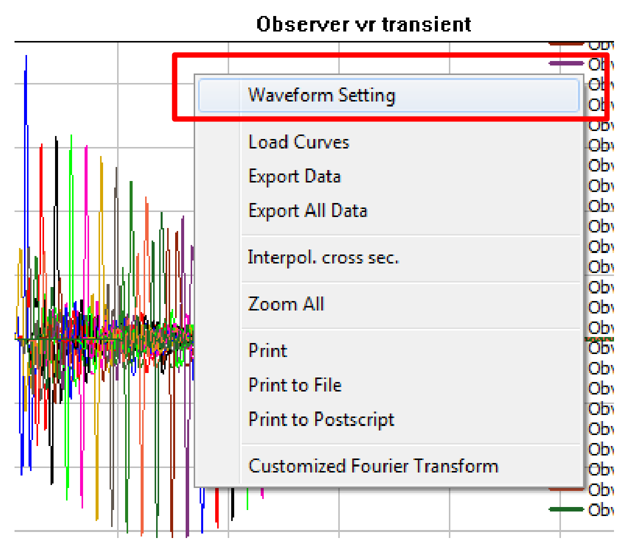
(Note: for the observer spectrum, it is normalized by the spectrum of the excitation. So, for a single source simulation, not matter what type of excitation pulse is used, the spectrum result for receiver will be the same.)

Manage 2D Displaying

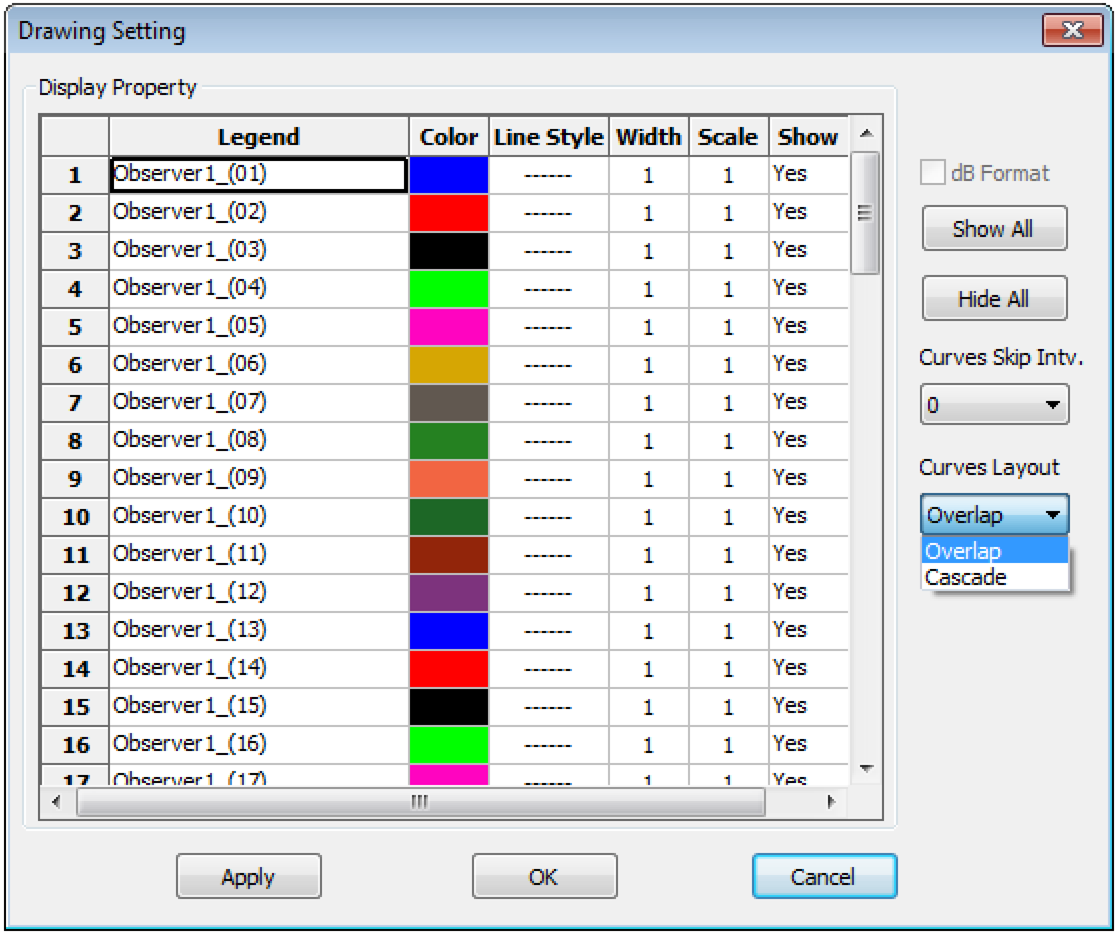


In the 2D canvas, user can popup a menu to

- control the displaying property of the trace
- Load curves from external data file to compare with existing trace
- Export traces to data file



Control the Displaying Property of Traces

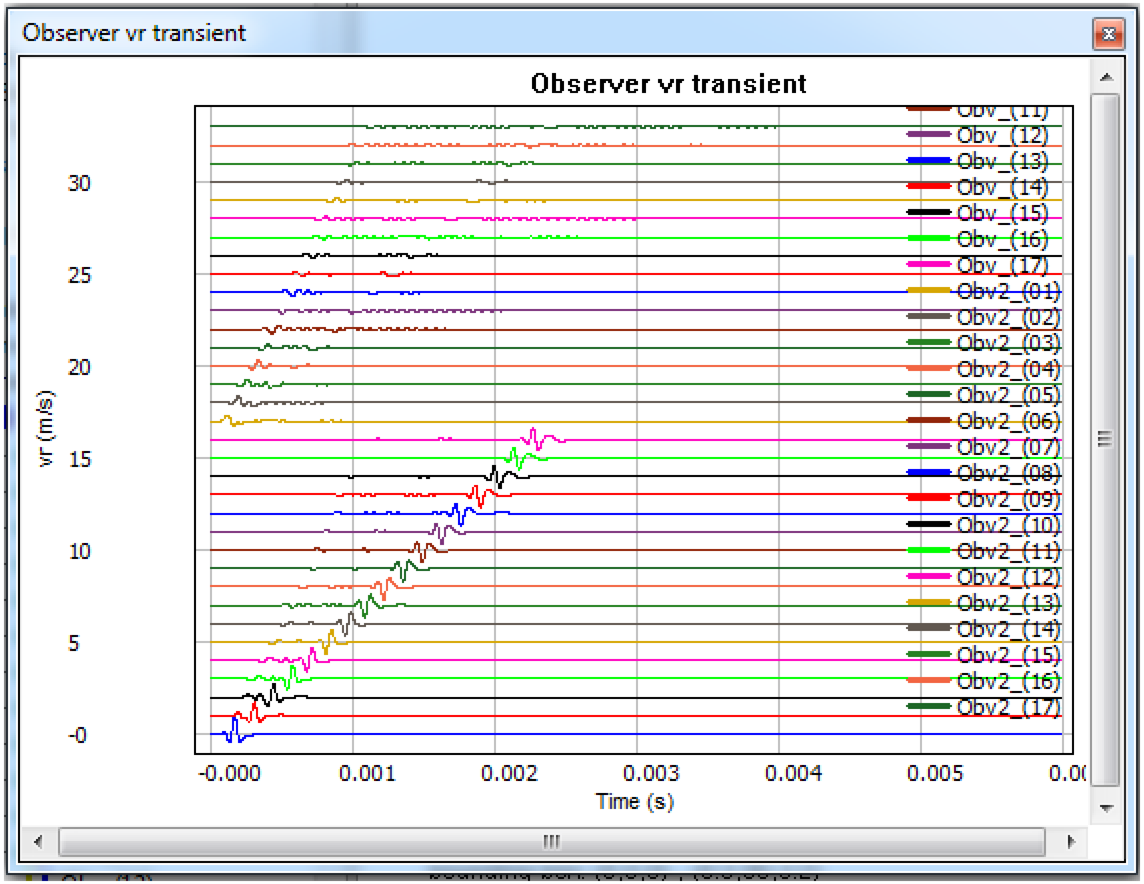


	Legend	Color	Line Style	Width	Scale	Show
1	Observer1_(01)		------	1	1	Yes
2	Observer1_(02)		------	1	1	Yes
3	Observer1_(03)		------	1	1	Yes
4	Observer1_(04)		------	1	1	Yes
5	Observer1_(05)		------	1	1	Yes
6	Observer1_(06)		------	1	1	Yes
7	Observer1_(07)		------	1	1	Yes
8	Observer1_(08)		------	1	1	Yes
9	Observer1_(09)		------	1	1	Yes
10	Observer1_(10)		------	1	1	Yes
11	Observer1_(11)		------	1	1	Yes
12	Observer1_(12)		------	1	1	Yes
13	Observer1_(13)		------	1	1	Yes
14	Observer1_(14)		------	1	1	Yes
15	Observer1_(15)		------	1	1	Yes
16	Observer1_(16)		------	1	1	Yes
17	Observer1_(17)		------	1	1	Yes

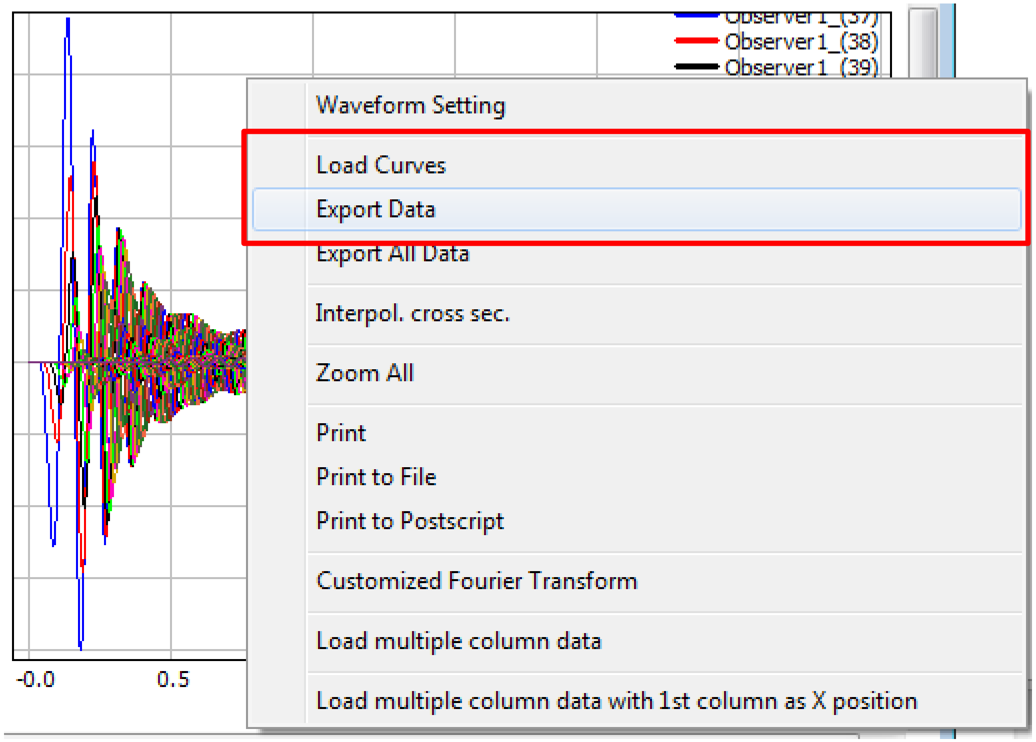
Default displaying layout is overlapping, but it can be changed to cascade style

Trace name   Trace color   Hide or Show

Example of displaying Traces in Cascade Style



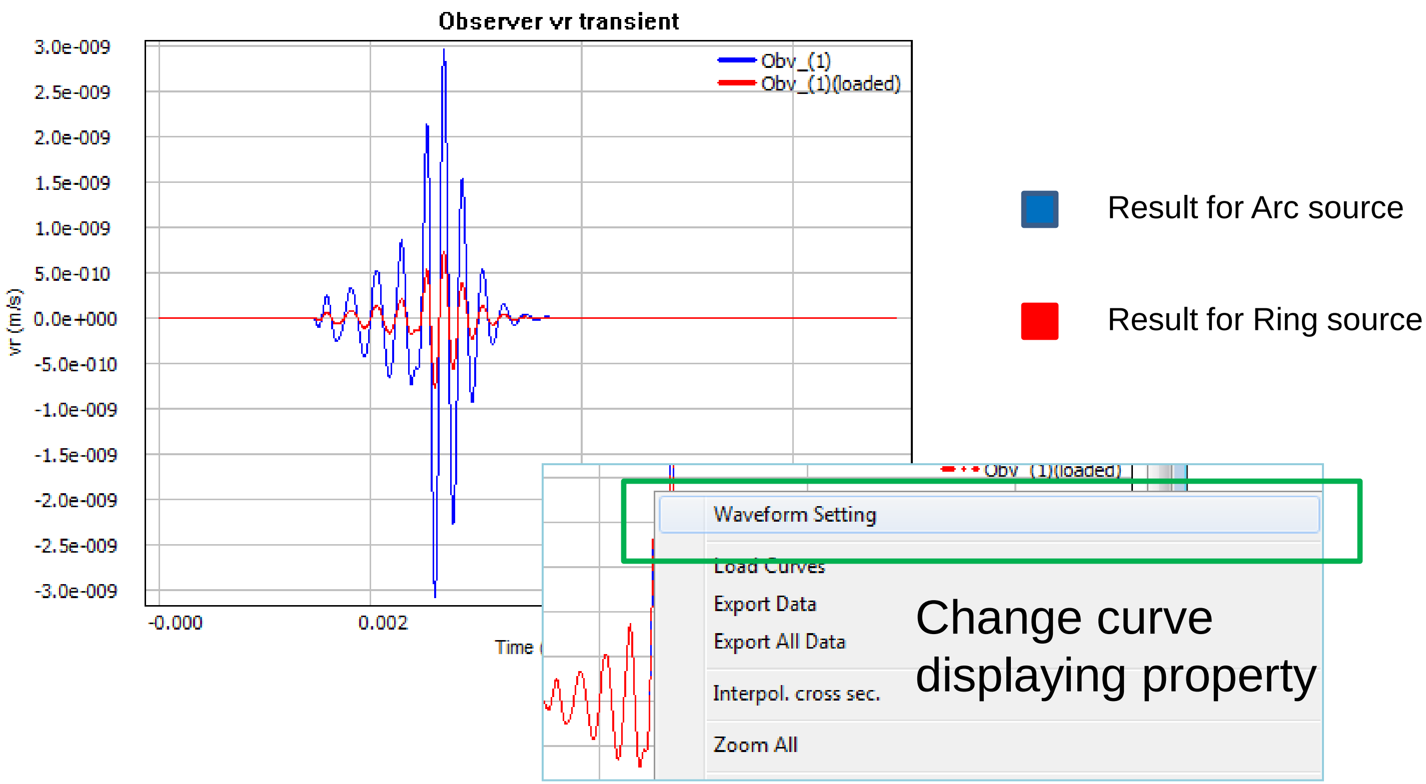
User can export the simulation result from one project, and load it in the 2D canvas of another project to compare the difference

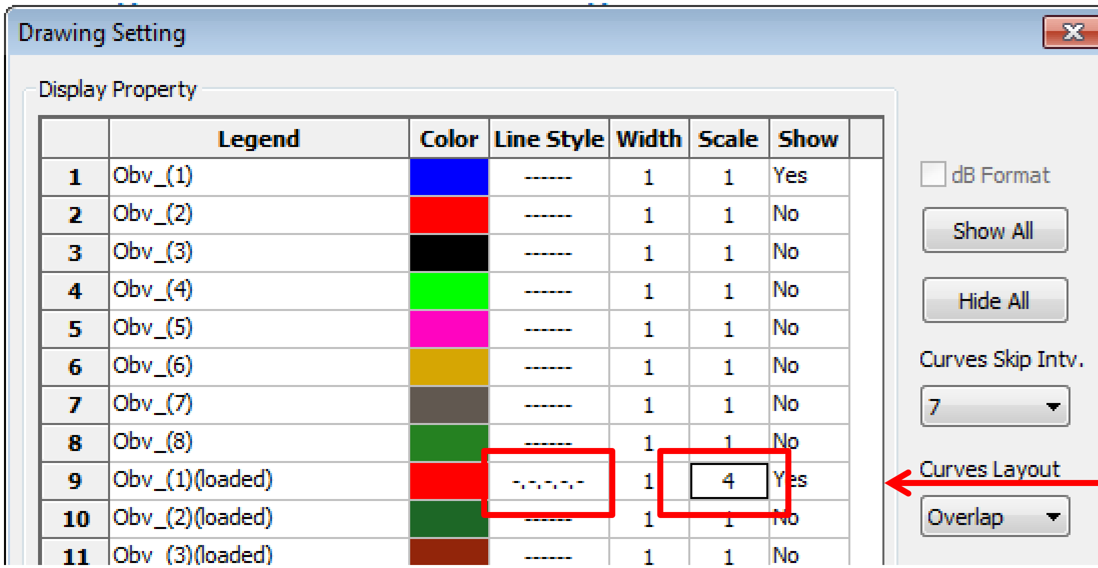


For example, as mention in the section “*the difference between the arc source and the ring source*”

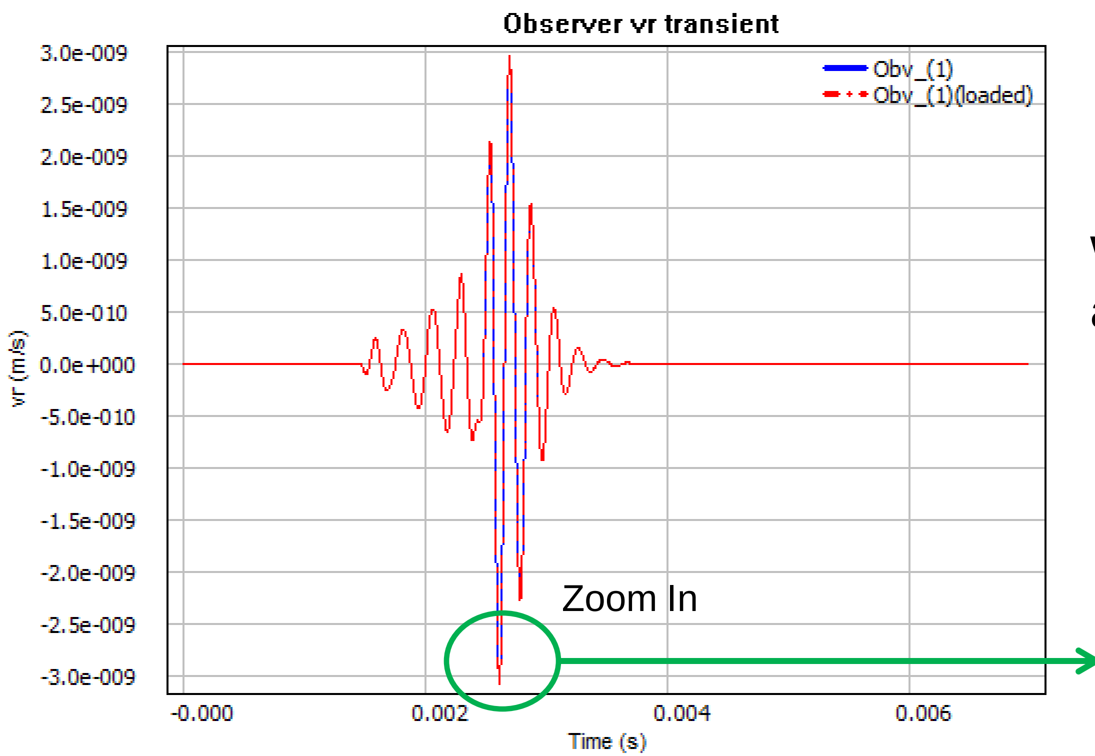
➢ For the domain with θ=[0, 90°], with the same radius, with normalization enabled, the simulation result of the arc source is **4 times** larger than that from the ring source

The origin 2 curves

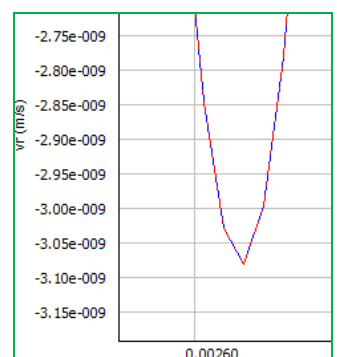




Change the loaded data with these settings



We can see these 2 curves are exactly match



Customized Fourier Transform on Transient Data

As mentioned in previous pages, the spectrum of observer shown in WCT GUI is normalized to a source already. But in some cases, user want to:

1) get the direct spectrum of the signal
2) the spectrum of partial transient signal
3) switch the normalization to other source: multiple sources situation

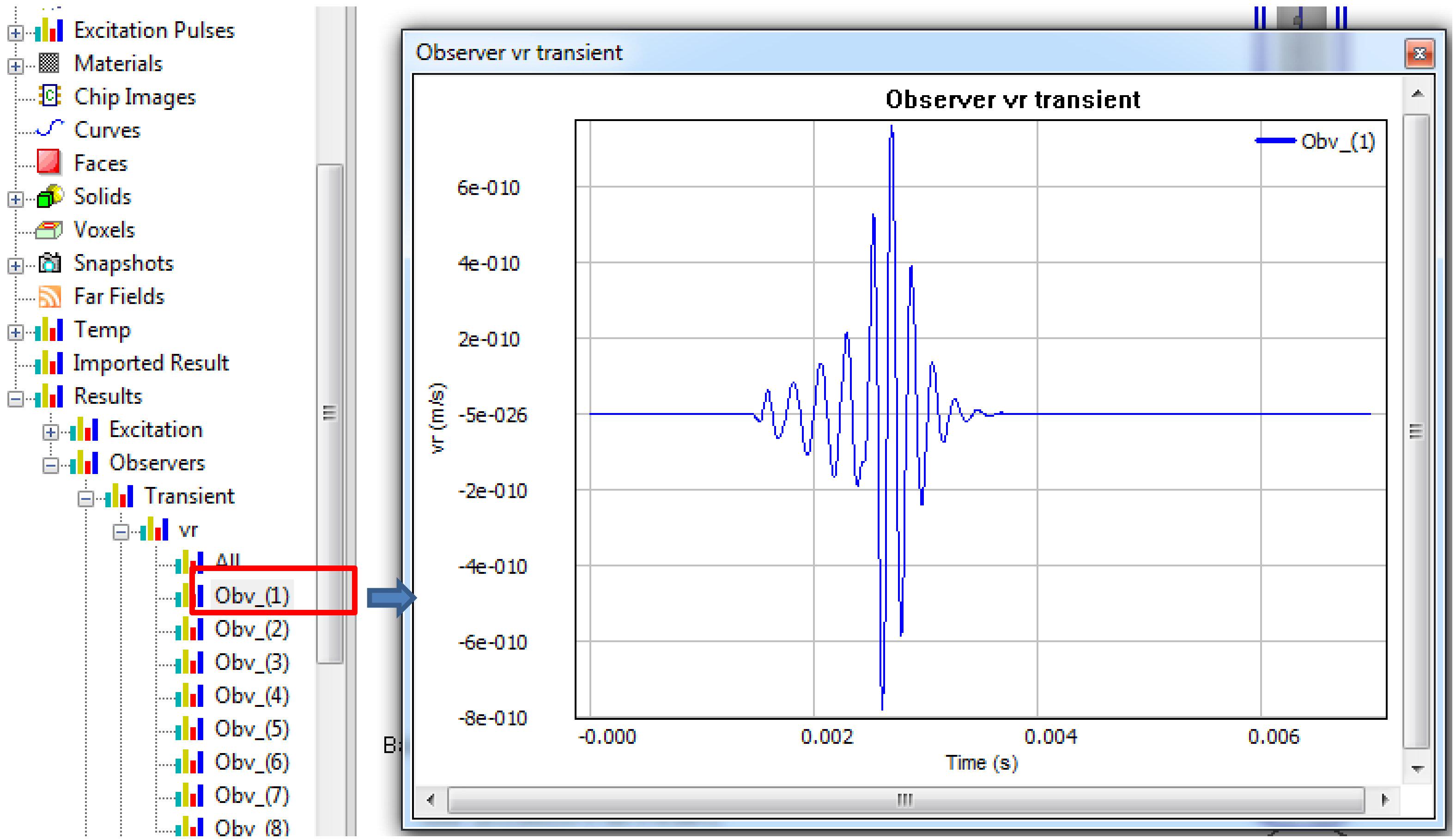
WCT GUI provides a tool: "Customized Fourier Transform on Transient Data" to satisfy above purposes.

WCT Customized Fourier Transform (FT) supports

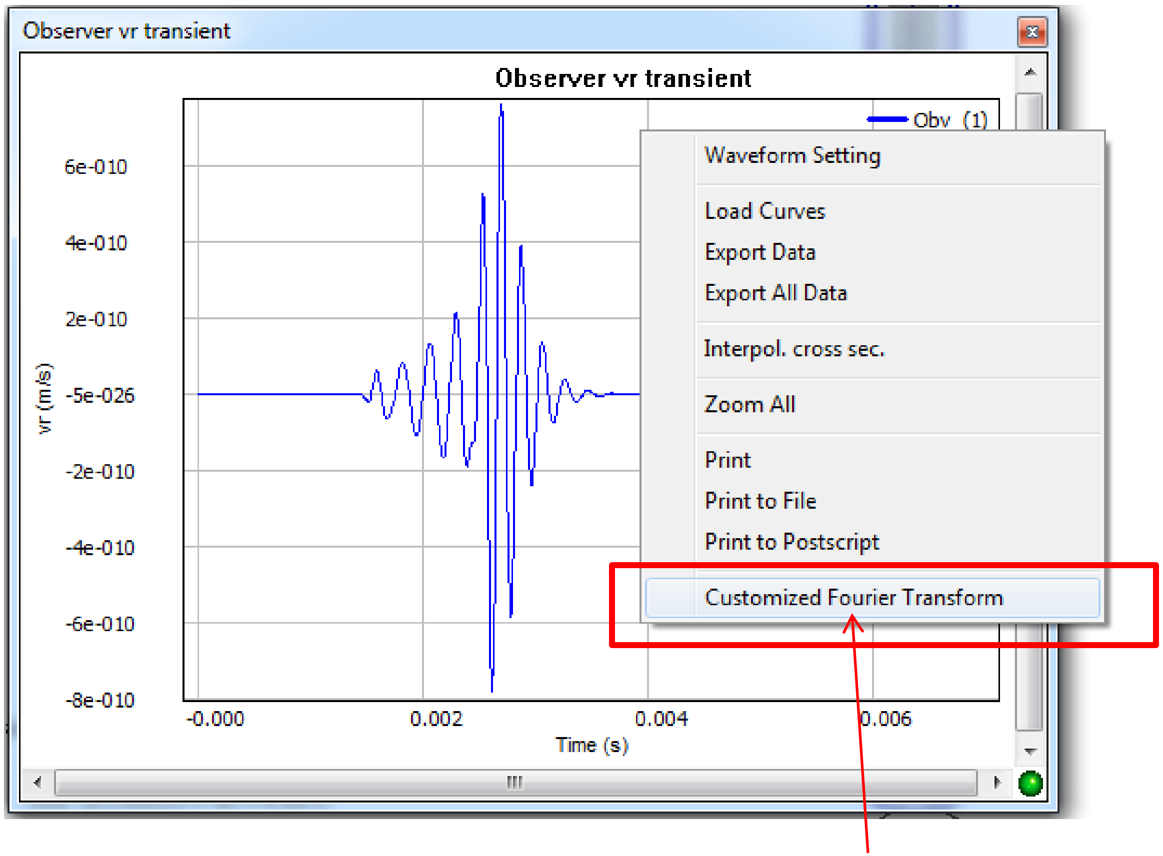
- Customized time window in the FT
- Customized frequency range for the FT result
- Customized normalization in FT
  - without normalization
  - can be the spectrum of any source pulse in that WCT simulation

Steps

1. Open one transient WCT simulation result.

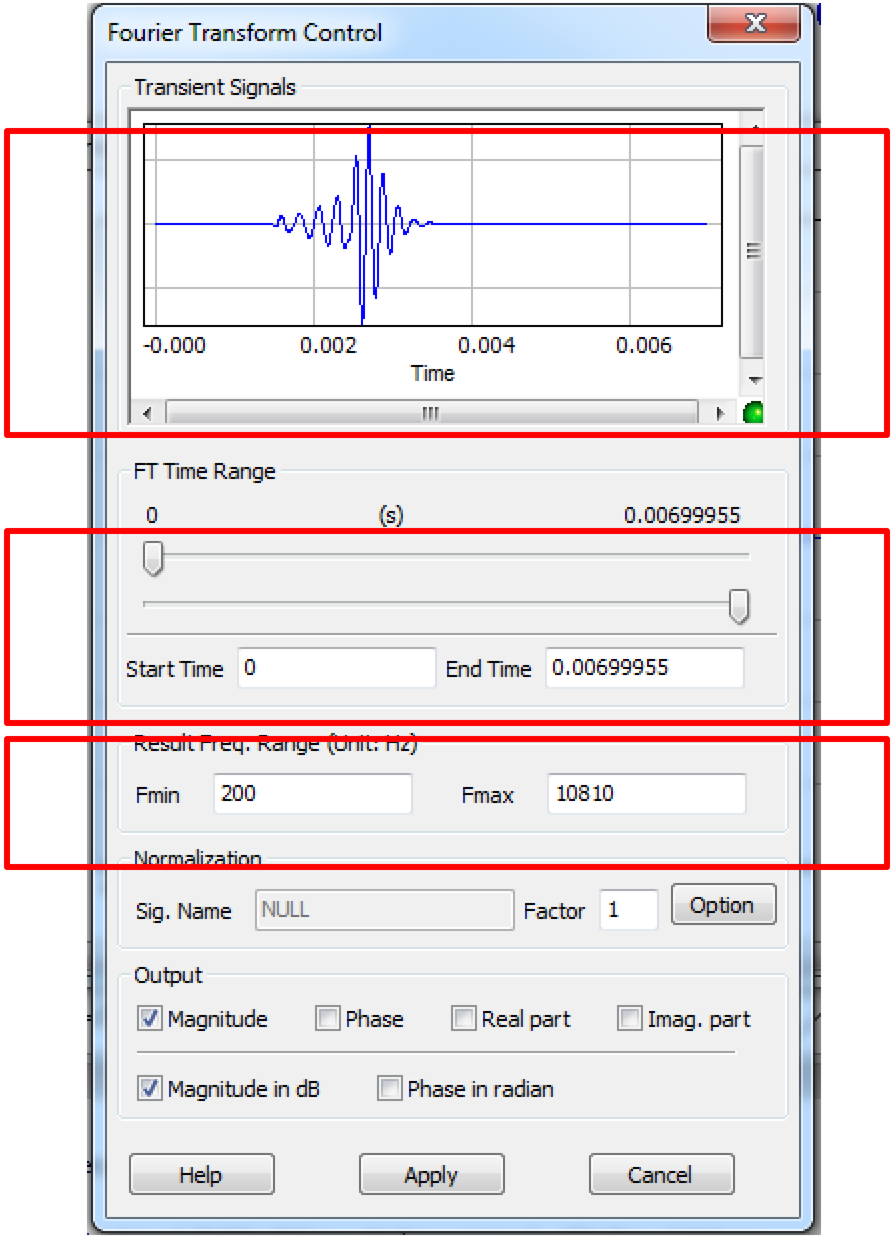


2. In the 2D canvas, right click mouse to popup the menu as following.



Click this “Customized Fourier Transform” menu item.

3. In the "Fourier Transform Control" dialog, setup the control as you want

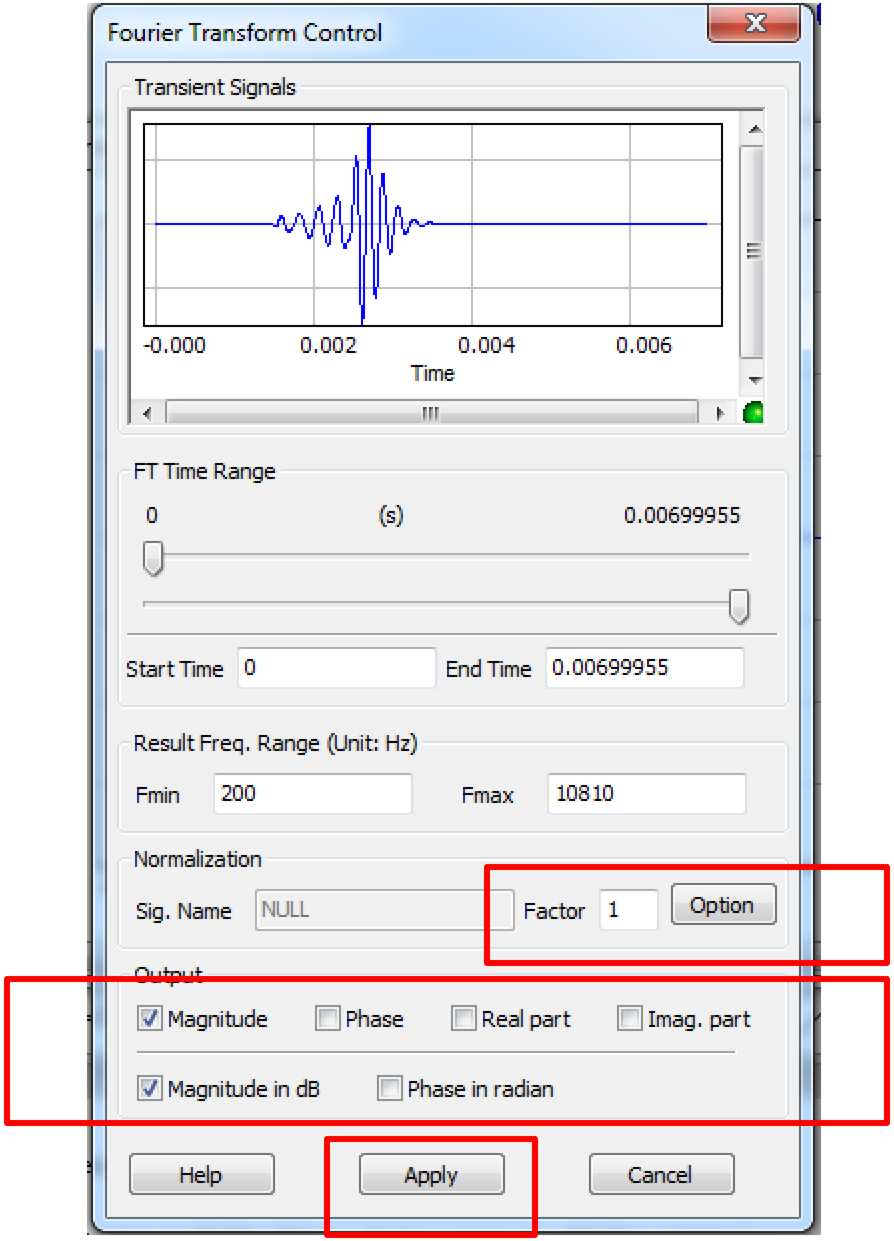


A down sampled transient trace for view purpose only (this is not the real transient data in FT, down sampled to speedup displaying only)

The time range will be used in FT.
Default is the real simulation time window in the simulation, but user can change it.

The exported spectrum range for the FT result.
Default is the f_{min} & f_{max} in the project.
The minimum value of f_{min} is 0; The maximum value of f_{max} is 200 times of f_{max} in the project.

cont.

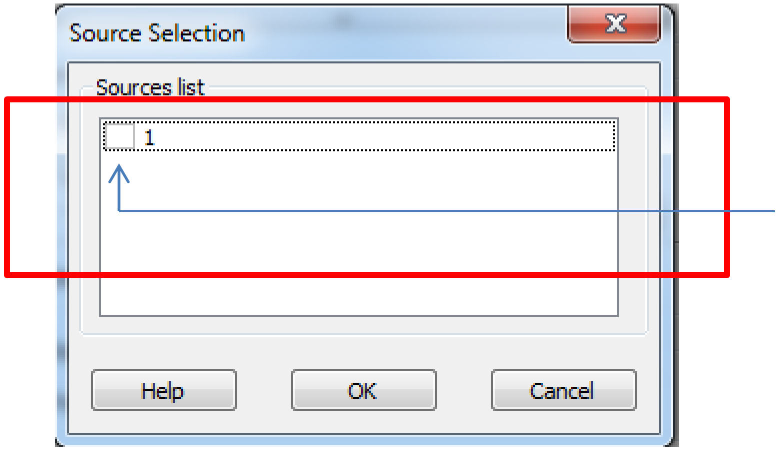


Normalization factor setup.
Please check the next page for detail setup.

What kind of FT results will be shown.

After all setup are OK, "Apply" to show FT result.

Set up the normalization trace in the FT



The list of all excitations in the simulation.

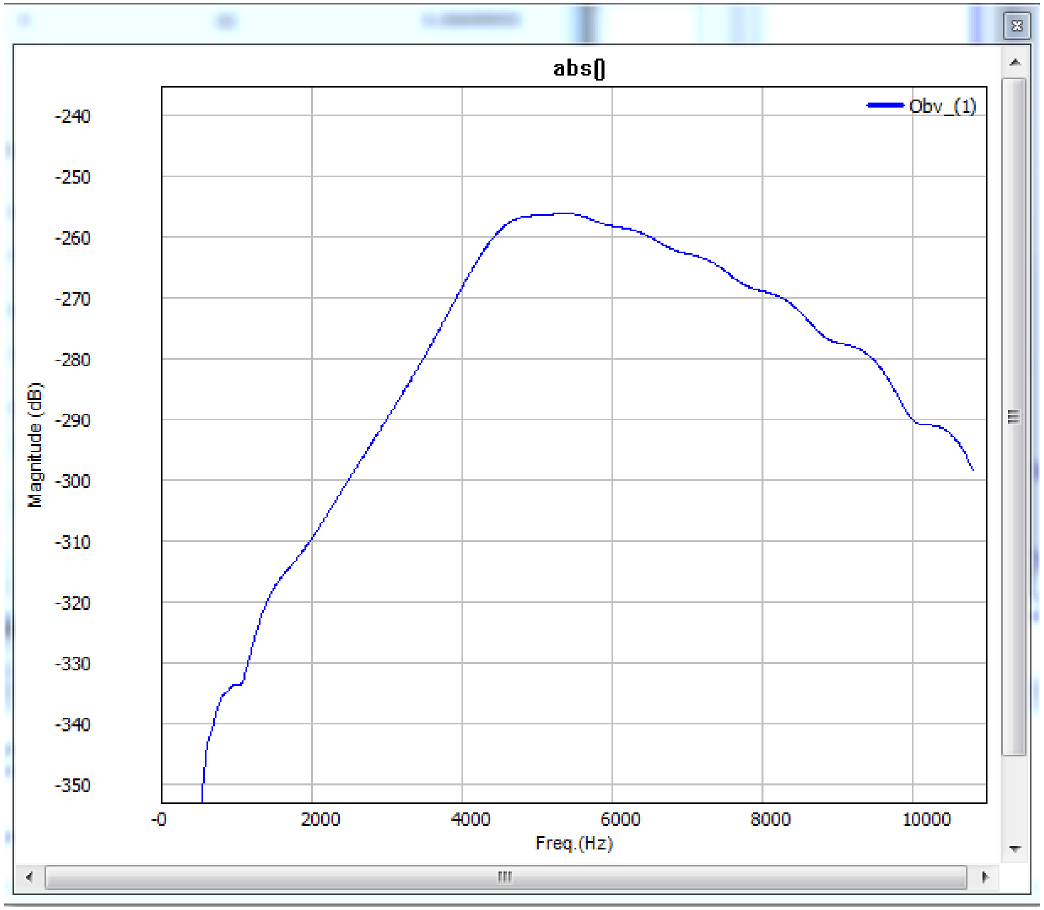
If one trace is selected (***check*** the trace), the spectrum of this trace will be working as the normalization trace in the FT.

The normalization method:

A. if there is not any excitation trace are selected, the FT result will not be normalized:
 FT_result = FT(Transient data)

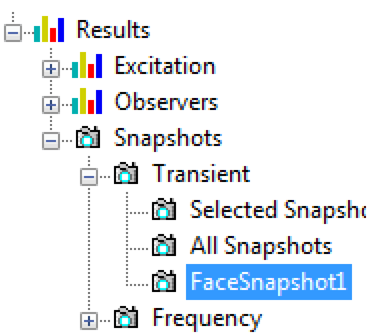
A. if there is one excitation trace selected, the FT result will be:
 FT_result = FT(Transient data) / FT(factor * excitation)

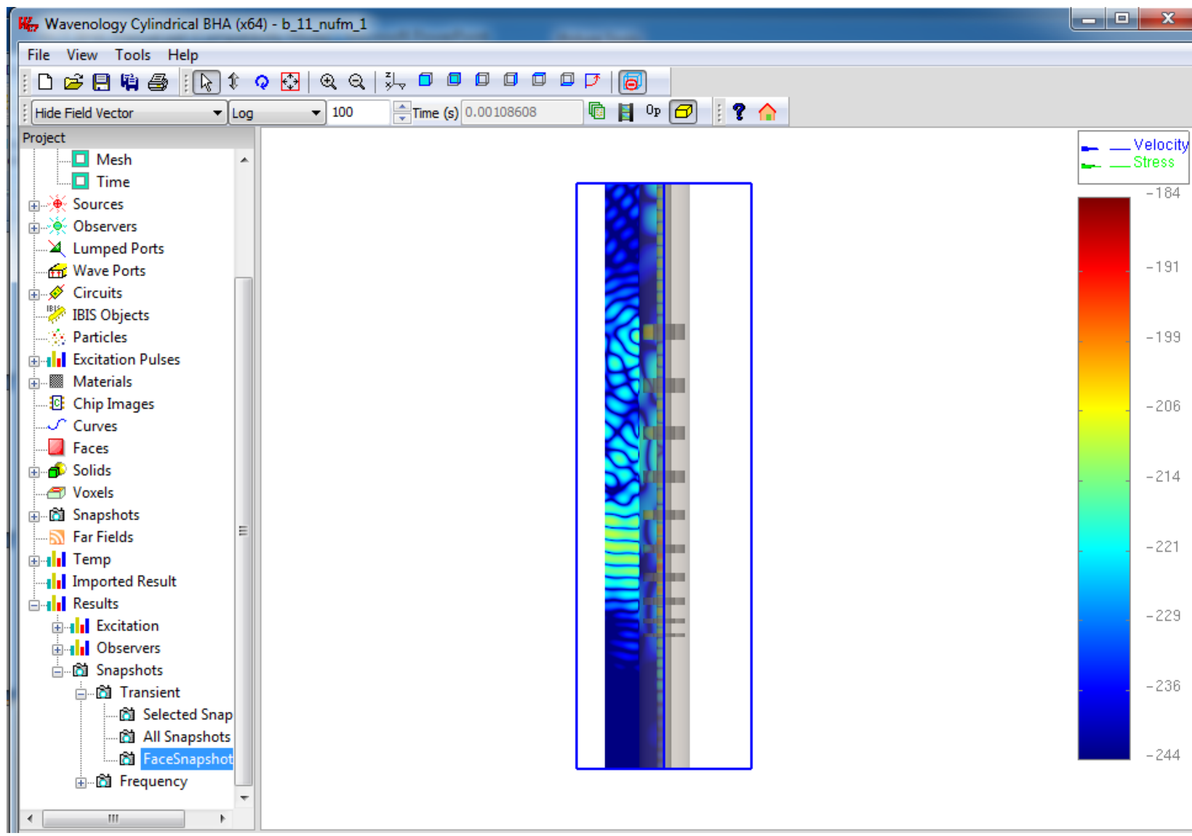
Following is the FT result without normalization.



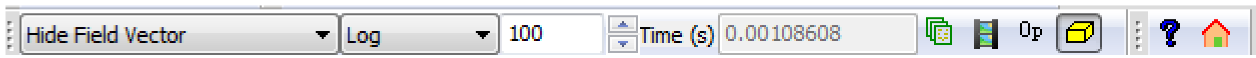
Display the Field Recorded in Snapshot

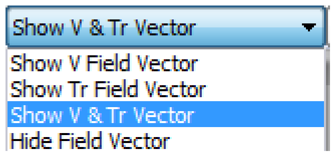
Double click any snapshot treenode in the result system, the 3D canvas will be switched to the snapshot displaying mode

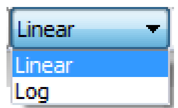




Following is the toolbar in snapshot displaying mode







Return to regular 3D canvas

Whether need to show the 3D solids as background

Manually switch shown frame

Automatic displaying snapshot frames

Whether need to display fields in the vector form

Whether the displayed data is in dB mode

Export all snapshot frames as a MPEG movie

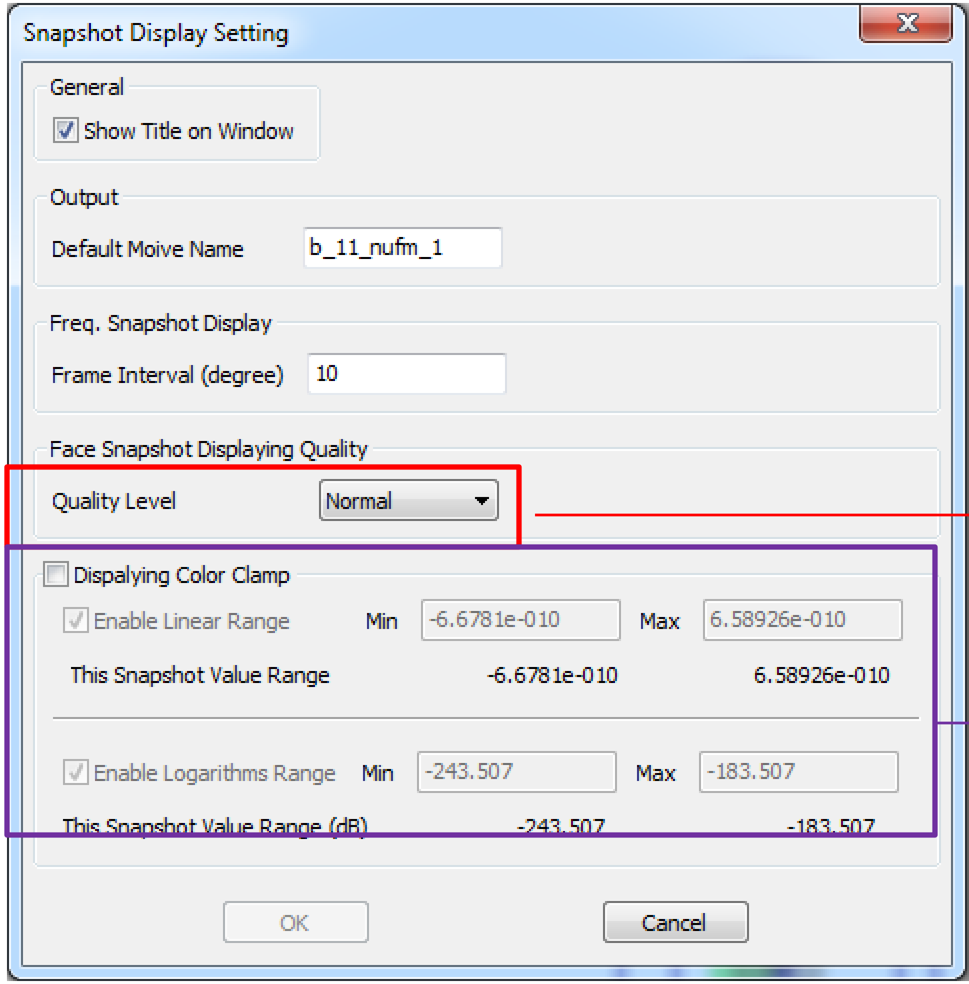
The frame index

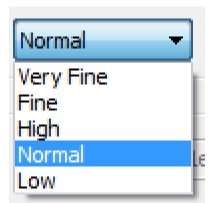
The corresponding time for the shown frame

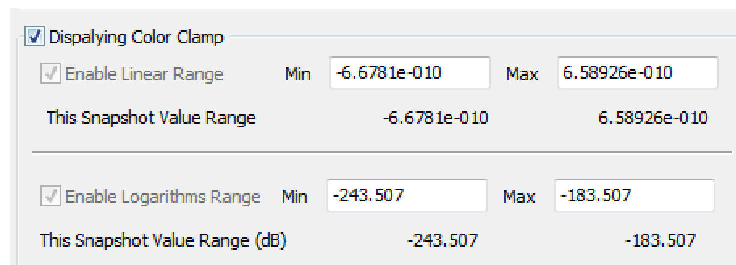
Options, see the next page

(Note: for a snapshot with a single component, the linear data range is ($data_{min}$, $data_{max}$); for a snapshot with multiple components, the linear data range is ($|vector|_{min}$, $|vector|_{max}$))

Snapshot displaying property

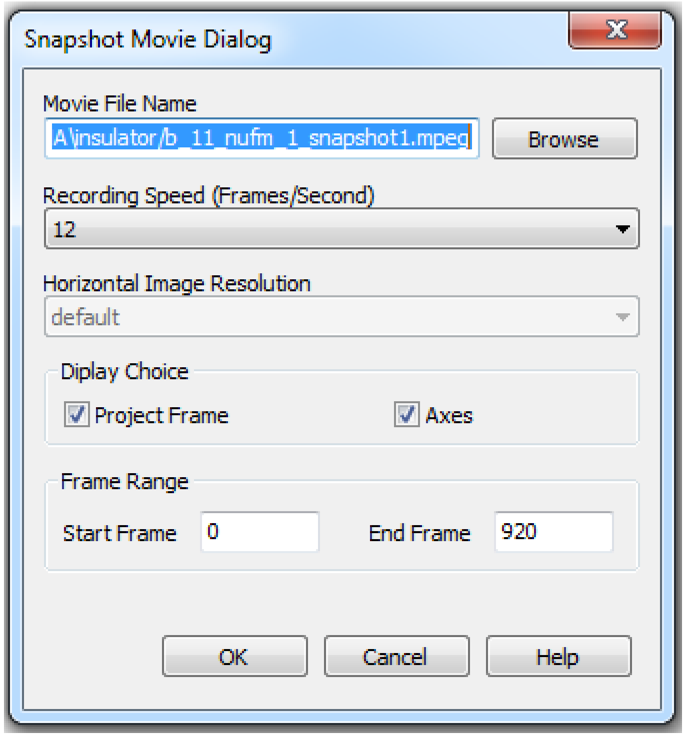






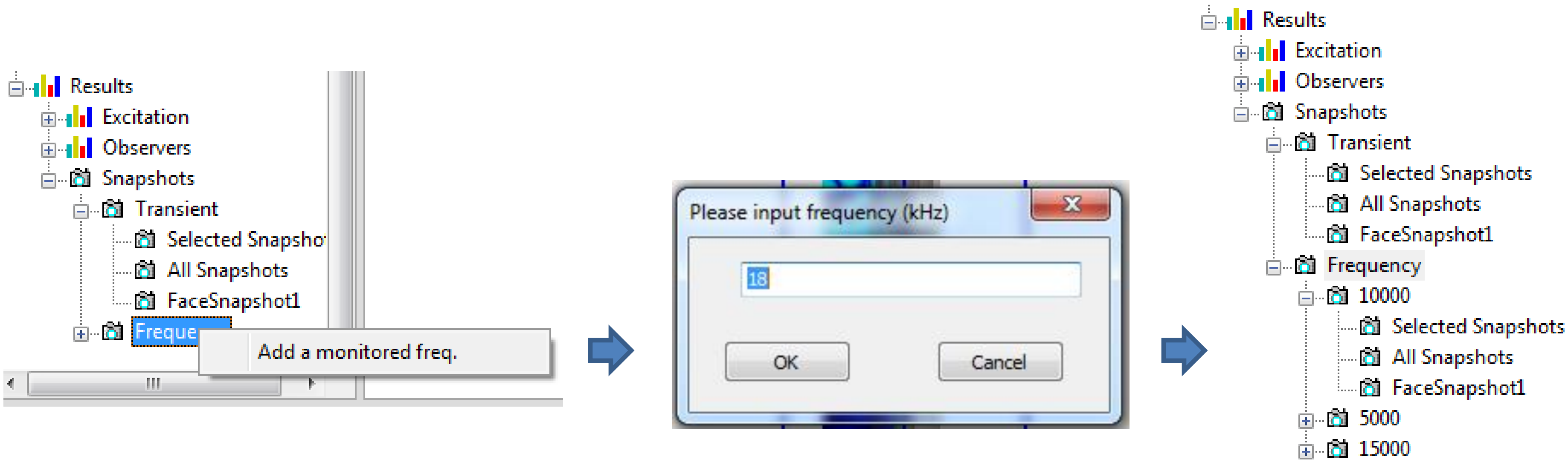
User can define a data range to change image displaying color

Movie property in exporting frames to Movie



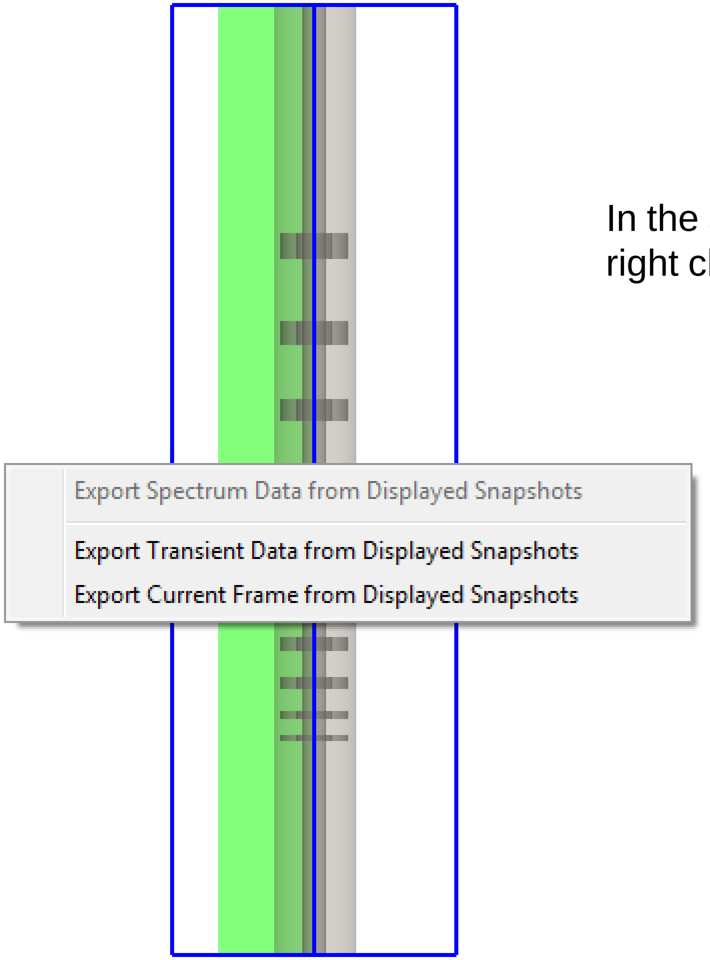
As the default setting, the transient data for a snapshot will be generated after simulation automatically.

User can generate the snapshot in frequency domain by following operations.



(Note: for a loaded project, the menu item "Add a monitored freq." will be enabled after the transient snapshot is loaded and shown)

Export the snapshot to external data file



In the snapshot displaying canvas,
right click mouse to popup this menu

For freq. domain snapshot data

For transient snapshot data

For the exported data format, please refer to ***the manual of WCT snapshot data file***.

Process Snapshot Data

The original snapshot data saved by GUI in the following place

- Snapshot data file position:
  - Project\project_res\snapshot\project_snaps.dat

  or
  - Project\project_res\snapshot\project_snaps.bin

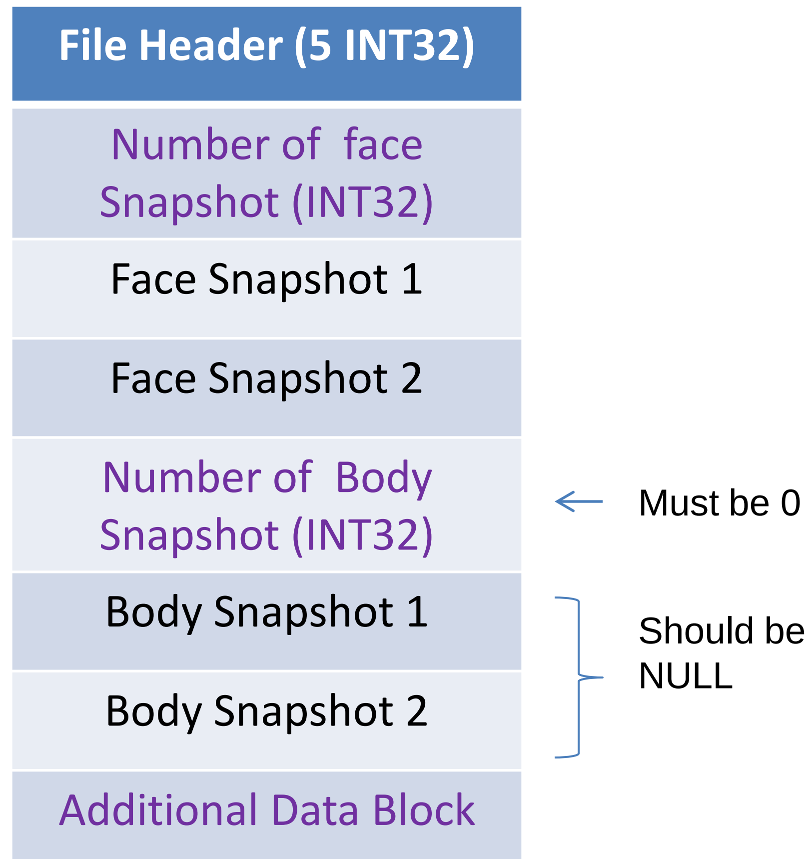
(Note: all snapshots share 1 data file)

- The data file use binary format, mix with ***char***, ***int*** and ***floating numbers***.

To process the snapshot result comes for the WCT BHA HPC solver, please refer the manual for the WCT BHA snapshot data. It is in the sub-folder "*process_snapshot_data*" of the demo package.

Data file basic structure

- In WCT EL Package, we support free face snapshot with transient data only.
- User need to know how many components recorded for each snapshot
- The V or Tau data are scaled in the file.
  - If real_V > real_Tau, scale_factor = max_real_v/max_real_Tau. Then all recorded V is real V, recorded Tau is real_Tau* scale_factor.
  - Otherwise, scale_factor = max_real_Tau/max_real_v, recorded Tau is real Tau, recorded V is real_v*scale_factor
- File header has the version info of the file
  - version 1: the file created by the GUI before v2.2.7
  - version 2: the file created by the GUI from v2.2.7



(Note: for the version 1, there is not "Additional Data Block"; for the version 2, "Additional Data Block" exists . If user only want to process the data before the "Additional Data Block", he can use the same method to process version 1 & 2 data.)

an example is

version 1

File Header (5 INT32)
2 (INT32)
Face Snapshot 1
Face Snapshot 2
0 (INT32)

version 2

File Header (5 INT32)	
2 (INT32)	← Number of Face Snapshot
Face Snapshot 1	
Face Snapshot 2	
0 (INT32)	← Number of Body Snapshot
Additional Data Block	← Additional Data

Binary Snapshot Data File Header

Data Type	Length in Byte	Meaning
int	4	0
int	4	0
int	4	0
int	4	0
int	4	Version number, must $\geq$ 1

Data structure for a Face Snapshot Data

Data Type	Length in Byte	Meaning
int	4	Block header ***1 – free face*** ***2 – solid face***
Structured data		Sampling information
Face Snapshot Data Block		

Structured sampling information for a Free Face Snapshot

Data Type	Length in Byte	Meaning

int	4	How many points in 1st dimension
		Free face is 2D planar face, r, θ or z plane only.
		The 2D plane is always use 3D for-loop to define sampling position as,
		for z
		    for θ
		        for r
		For z plane, 1st dimension is r; θ plane's 1st dimension is r; r's is θ.

Face Snapshot Data Block

Data Type	Length in Byte	Meaning
int	4	Number of characters for name
char	N	Snapshot name (N is from above)
int	4	Number of sampling points
Block i	M	Sampling data for point 1^{st}
	…	
Block j	M	Sampling data for point jth

Data structure for a block

Data Type	Length in Byte	Meaning
double	8	Sampling point x position
double	8	Sampling point y position
double	8	Sampling point z position
int	4	Trace length for all components
float	4*Len	Trace for component 1
…		
float	4*Len	Trace for component k

The additional data block

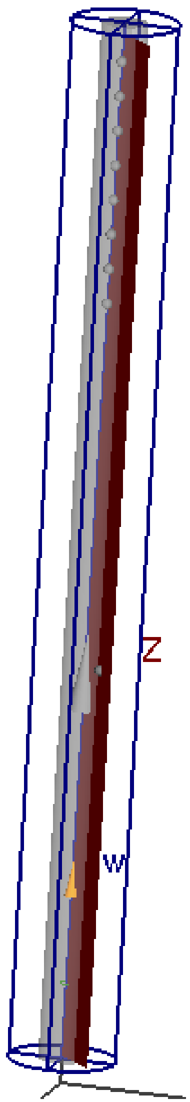
Data Type	Length in Byte	Meaning
int	4	Version number. Value=1 : version 1 For version 1, following data has a fixed length of 196 bytes. The format is as following
double	8	Recording time step for the snapshot. Unit: second
…..	188	Reserved

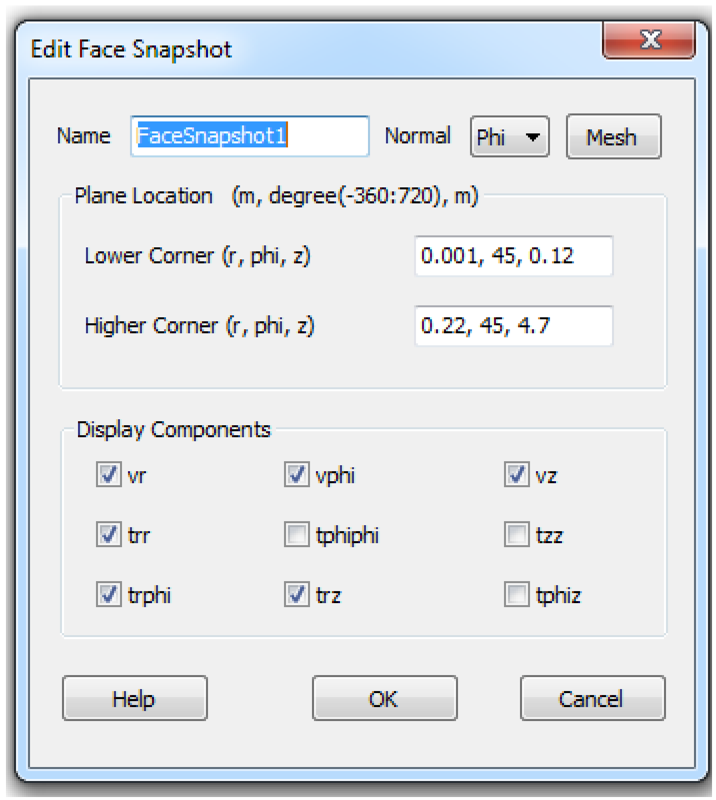
Total 200 bytes

The total length of the additional data block is 200 bytes.

Example by the Open-Hole Case

(the demo case & Matlab code are in the sub-folder “process_snapshot_data” of demo package)





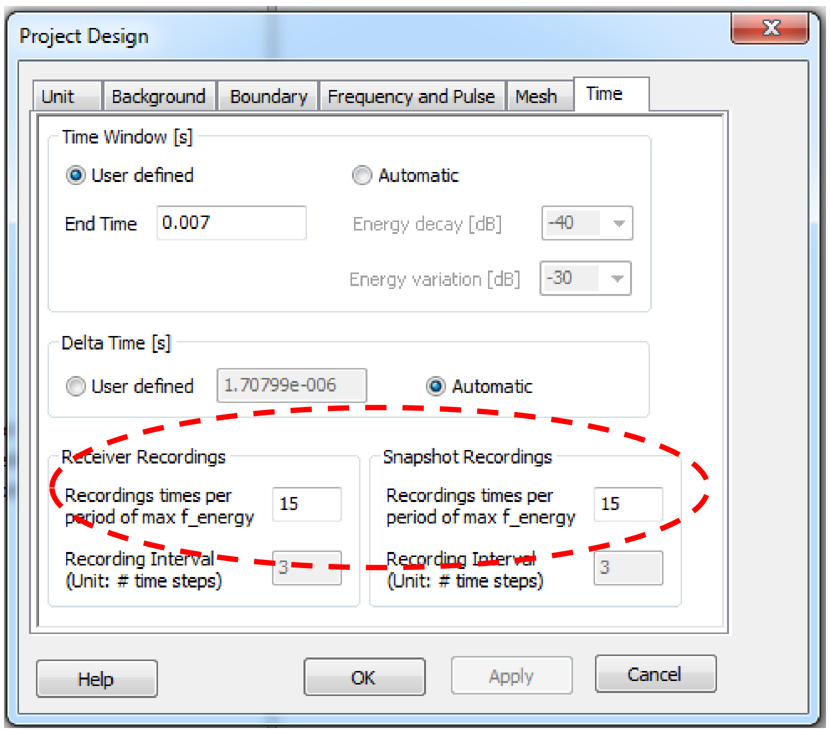
- A face snapshot is defined as left figure.
- This snapshot includes 6 components
- If we say the project is “open-hole\a2.wnt”, then the snapshot data file is “open-hole\a2_res\snapshots _res\ a2_snap.dat”

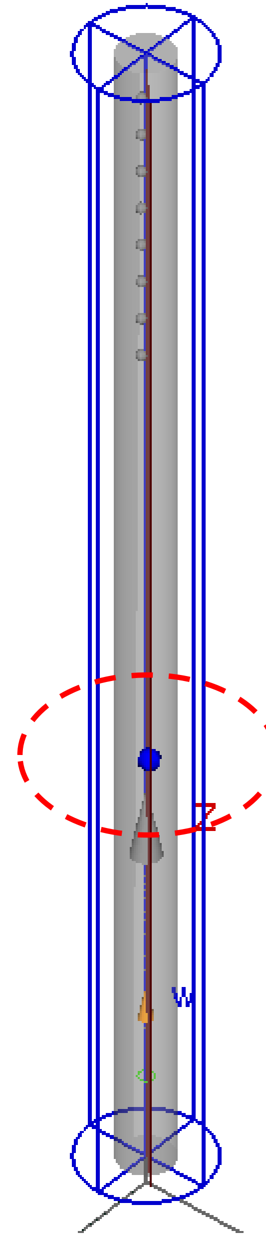
- In order to make a comparison, we define a receiver at (0.1191, 45, 1.8785), which has a equivalent point in the snapshot shot's grid 17-rows and 6 column.

- The sampling point of snapshot can be read from file as the demo matlab code.

- After simulation, we export the Vr on this receiver by data file 'vr_a1.txt' to be the reference data file.

- In general, WCT GUI samples 2 points per period for snapshot; however, regular receiver samples 20 points per period. In order to make two sampling can be comparable, we set two sampling density as the same,





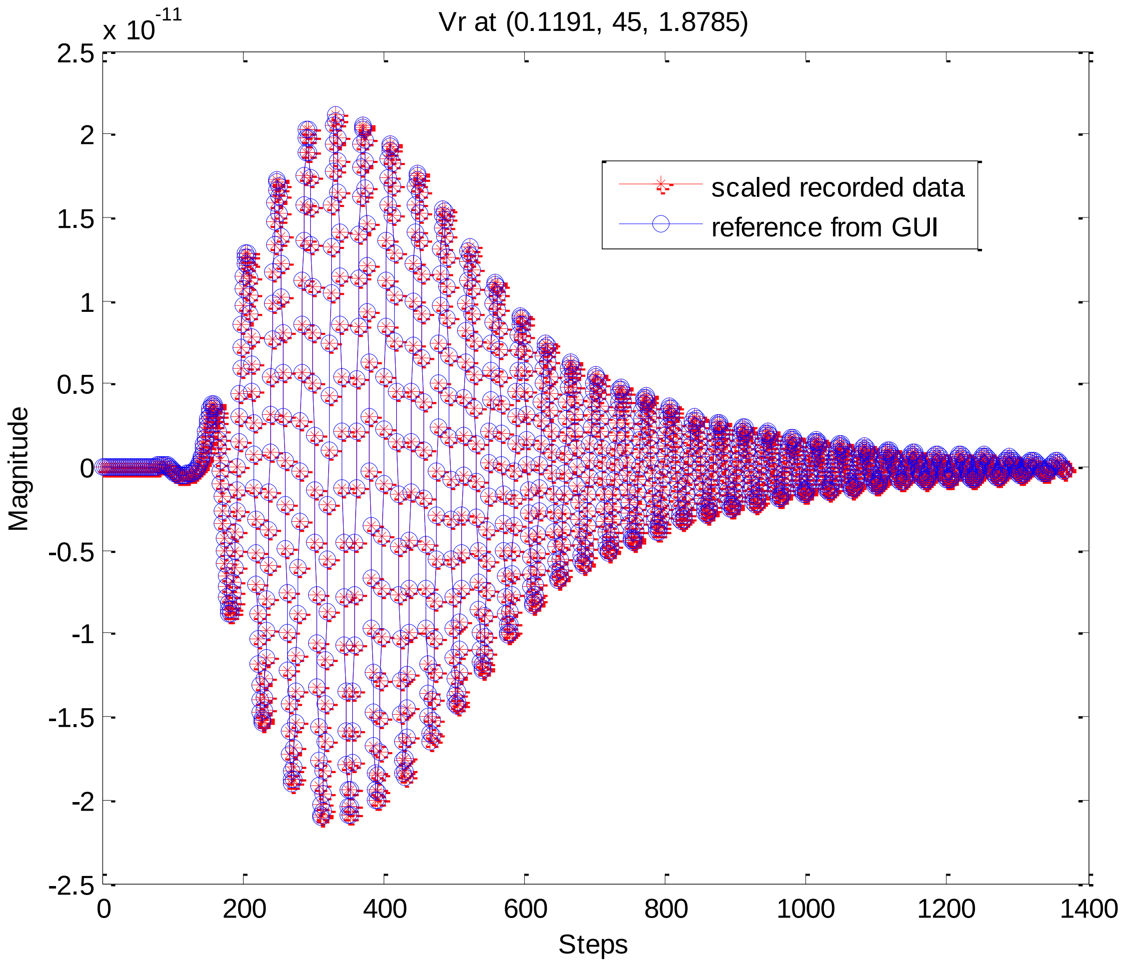
Demo matlab code to read data except "Additional Data Block"

```
clear all;
%%%%%%
nComp = 6;  %%%% firstly, you need to determine how many
        possible field components will be in the data file
         %%%% here, we assume this is only one snapshot in the file,
        this snapshot has 6 components

% open file
fid = fopen( 'a2_snap.dat', 'rb' ); % target file
if( fid == -1 )    return;  end;

%%% read header
n0 = fread( fid, 1, '*int32' );
n1 = fread( fid, 1, '*int32' );
n2 = fread( fid, 1, '*int32' );
n3 = fread( fid, 1, '*int32' );
ver = fread( fid, 1, '*int32' );

%%% make sure it is the new version
if( ~((ver == 1) || (ver == 2)) )
    fclose( fid );
    return;
 end;

%%% face snapshot part
%%% the number of face snapshot in the data file
nFS = fread( fid, 1, '*int32' );

%%%% following code assume there is only one snapshot in the file.
%%%% if there are more, you need to use your data structure to
        store information
```

```
if nFS > 0,
   %%% there is face snapshot

   for k = 1 : nFS,    %%% read face snapshot one by one
      nType = fread( fid, 1, '*int32' ); %%% face type (free face or solid
face)
      n1D = fread( fid, 1, '*int32' );   %%% 1st dimension length
      nNameLen = fread( fid, 1, '*int32' );   %%% snapshot length
      sFsName = fread( fid, double(nNameLen), '*char' );   %%%
snapshot name
      nPoint = fread( fid, 1, '*int32' );  %%% total point number, so the
array will be (n1D) * (nPoint/n1D)

      %%%% all point will include a structured information
      for j = 1 : nPoint
         %%% r, theta, z position
         x = fread( fid, 1, '*double' );   %%% point r position
         y = fread( fid, 1, '*double' );   %%% point theta position
         z = fread( fid, 1, '*double' );   %%% point z position

         pos(j, :) = [x y z];
         len = fread( fid, 1, '*int32' );  %%% trace length for this receiver
         len = len / 4;      %%% this length is the size of 'char', need to
convert to 'float'

         for iComp = 1 : nComp,   %%% for all possible components
            v = fread( fid, double(len), '*float' );
            fss(j, iComp, :) = v;
         end;
      end;
   end;

end;
```

```
%%% volume snapshot part
nBS = fread( fid, 1, '*int32' );

%%%% for EL solver, we use face snapshot only, then we skip following
data file
fclose( fid );

%%%% display
%%% for this 2D array, the inner point number is 'n1D'
n2D = nPoint / n1D;

xPos = pos(:, 1);    xPos = reshape( xPos, n1D, n2D );
yPos = pos(:, 2);    yPos = reshape( yPos, n1D, n2D );
zPos = pos(:, 3);    zPos = reshape( zPos, n1D, n2D );

%%%% we get a obv. at (r=6, z=17)
idr = 6;
idz = 16;
id = idz * n1D + idr;
vr = squeeze( fss( id, 1, : ) );
max_vr = max( abs(vr) );

%%% reference data
aa = load( 'vr_a1.txt' );
vr_ref = aa( 2:end, 2 );
max_vr_ref = max( abs(vr_ref) );
scale = max_vr_ref / max_vr;

%%%%%%%%%%%%%%%%%%%%%

figure;
plot( vr, 'r*-' );
hold on;
plot( vr_ref/scale, 'bo-' );
```

As can be seen, the scaled back data is exactly match the original receiver result directly from GUI.



Demo matlab code to read data with "Additional Data Block"

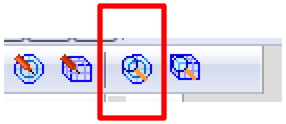
```
clear all;

%%%%%%%
nComp = 6;  %%%%% firstly, you need to
determine how many possible field
components will be in the data file
        %%%%% here, we assume this is only
one snapshot in the file, this snapshot has 6
components

% open file
fid = fopen( 'a2_snap.dat', 'rb' ); % target file
if( fid == -1 )
   return;
end;

%%%% read header
n0 = fread( fid, 1, '*int32' );
n1 = fread( fid, 1, '*int32' );
n2 = fread( fid, 1, '*int32' );
n3 = fread( fid, 1, '*int32' );
ver = fread( fid, 1, '*int32' );

%%%% make sure it is the version #2
if( ~( (ver == 2)) )
   fclose( fid );
   return;
end;

%%%% face snapshot part
%%%% the number of face snapshot in the
data file
nFS = fread( fid, 1, '*int32' );

%%%%% following code assume there is only
one snapshot in the file.
%%%%% if there are more, you need to use
your data structure to store
%%%%% information

if nFS > 0,
   %%%% there is face snapshot

   for k = 1 : nFS,    %%%% read face snapshot
one by one
      nType = fread( fid, 1, '*int32' ); %%%%
face type (free face or solid face)
      n1D = fread( fid, 1, '*int32' );   %%%% 1st
dimension length
      nNameLen = fread( fid, 1, '*int32' );
%%%% snapshot length
      sFsName = fread( fid,
double(nNameLen), '*char' );   %%%%
snapshot name
      nPoint = fread( fid, 1, '*int32' );  %%%%
total point number, so the array will be (n1D) *
(nPoint/n1D)

      %%%%% all point will include a structured
information
      for j = 1 : nPoint
         %%%% r, theta, z position
         x = fread( fid, 1, '*double' );   %%%%
point r position
         y = fread( fid, 1, '*double' );   %%%%
point theta position
         z = fread( fid, 1, '*double' );   %%%%
point z position

         pos(j, :) = [x y z];
         len = fread( fid, 1, '*int32' );  %%%%
trace length for this receiver
         len = len / 4;      %%%% this length is
the size of 'char', need to convert to 'float'

         for iComp = 1 : nComp,   %%%% for all
possible components
            v = fread( fid, double(len), '*float' );
            fss(j, iComp, :) = v;
         end;
      end;
   end;

end;

%%%% volume snapshot part
nBS = fread( fid, 1, '*int32' );

%%%% additional data block
add_ver = fread( fid, 1, '*int32' );
if( add_ver == 1 )
   % the next value is the dt of the snapshot
   dt_ss = fread( fid, 1, '*double' );

   % show this value
   dt_ss
end;

%%%%% for EL solver, we use face snapshot
only, then we skip following data file
fclose( fid );
```

Appendix

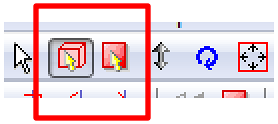
Mesh Data

(1) Display & Check Mesh Data

User can use Wavenology GUI to check the mesh directly by displaying the mesh in the canvas.



Show the cylindrical mesh



Use “Select Body” or “Select Face” to quit mesh viewer.

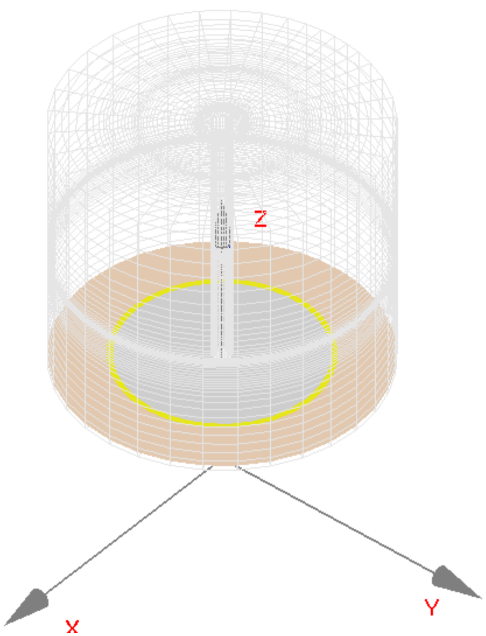
After click one of the “**Show mesh**” buttons, the main canvas will switch to “**Mesh Displaying**” mode. The two coordinate system modes has similar toolbar system, and only the definition of plane-normal is different.

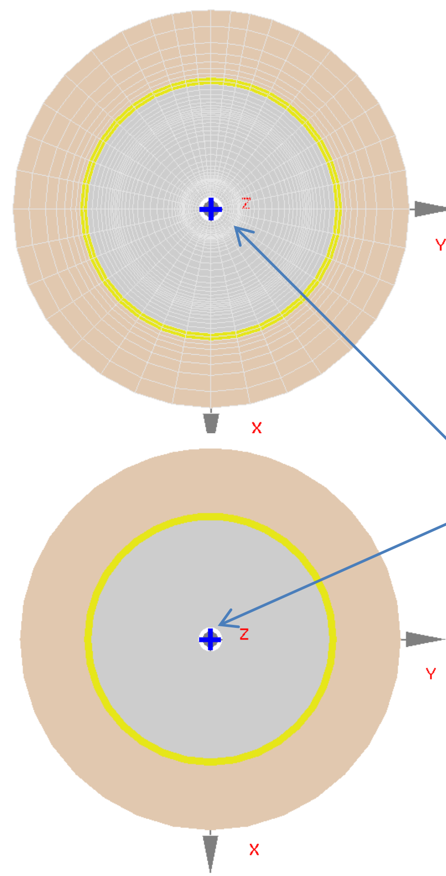
In following pages, we demonstrate how to display the cylindrical mesh.

Mesh displaying controls

Click top view mode

User can check whether the cell-material is correct or not





The grid can be hidden by uncheck

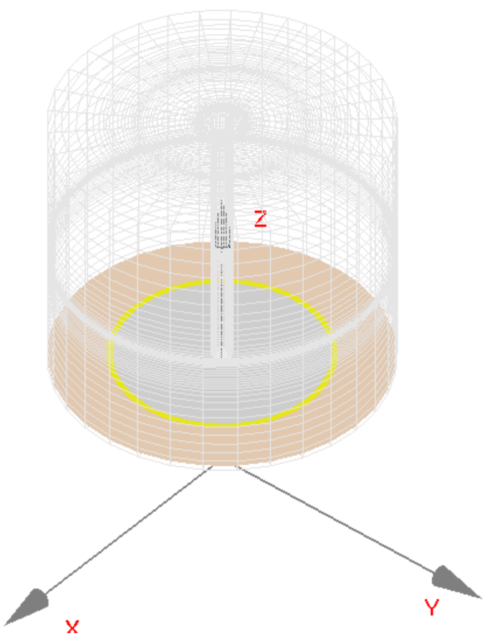
Note: the blue cross is the view rotation center, not a real object.

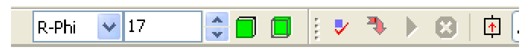
The default displaying mode is lowest Z (R-Phi plane) cells with grid line shown.

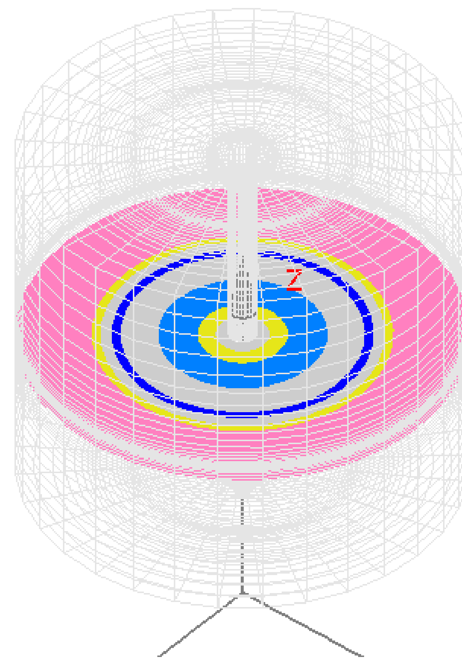
Click
reset view mode

Change the section index.
A different mesh-section is displayed

The canvas return back the
angle view mode.

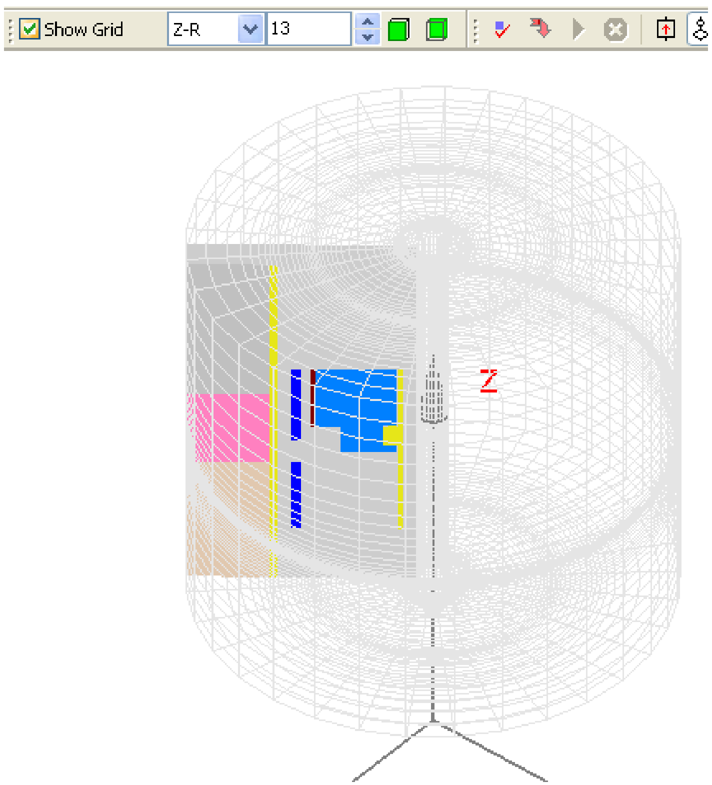




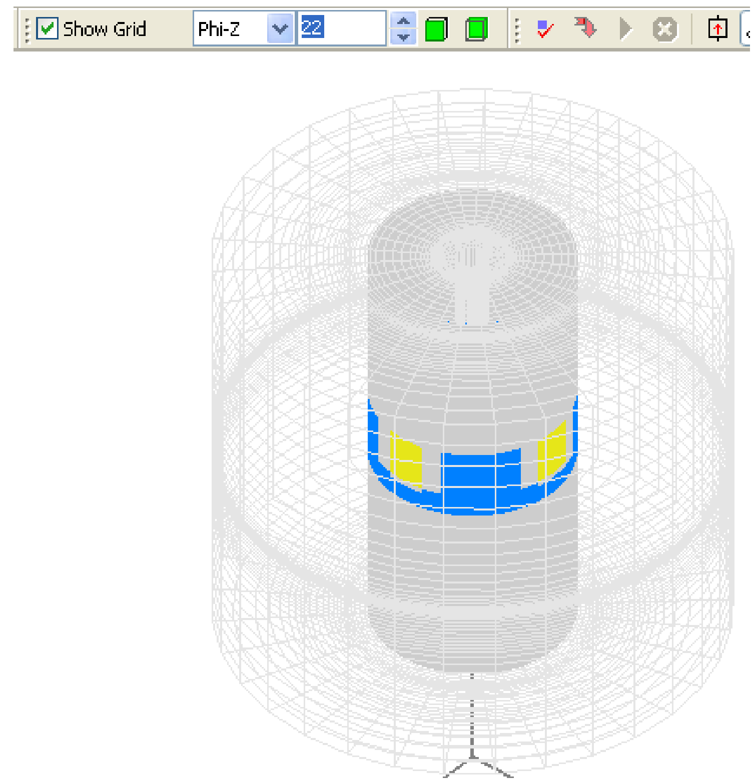


Note: the displaying color for each cell is determined by the material in that cell. The material color is defined as one of the material property in the **Material Editor**

Switch to different displaying plane to check mesh.



Z-R plane displaying example



Phi-Z plane displaying example

(2) Export Mesh Data

The mesh can be exported to external data file also.

Generate cylindrical mesh and export the data file:

cyl_mesh.txt	cell material index, ascii format
material_info.txt	material name and index , ascii format
cyl_mesh_grid.m	grid position, matlab m file

Note: the exported files has the same names for all cases. The files from a new generation will replace the old data files.
It is suggested to use different folders to save different cases.

Exported Mesh Data File Format

material_info.txt material name and index , ascii format

```
% id  name   eps_r    e_cond      mu_r       m_cond
0     Air          1    0    1    0
1     PEC          1    0    1    0
2     m1           2    0    1    0
3     m2           1    0    1    0
4     m3           1    0    1    0
5     rock1        2    0    1    0
6     rock2        4    0    1    0
7     rock3        3    0    1    0
```

1st column is the material index in this case, not matter this material has been used for any solid or background

2nd column is the material name

other columns are electric-profile of the material. We will add more parameters to support elastic wave solver in the future.

cyl_mesh.txt cell material index, ascii format

data	meaning
N	How many cells along R(cylindrical mesh:cyl_mesh.txt)
next N rows	Cell centers along R or X axis
M	Cells number along Phi or Y axis
next M rows	Cell centers along Phi or Y axis
K	Cells number along Z axis
next K rows	Cell centers along Z axis

next N*M*K rows	Material index for each cell, the index is the same as the file *material_info.txt*
	
	The data is created by the "for-loop"
	For R
	    For Phi
	        For Z
	        …
	        end Z
	    end Phi
	End R

cyl_mesh_grid.m grid position, matlab m file

User can directly run these files in matlab. After running, user can get following variables:

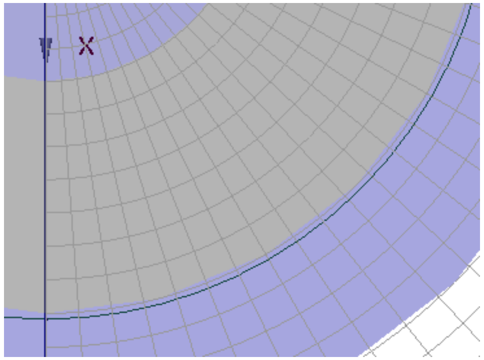
NR or NX:	grid number in R or X axis
NPhi or NY:	grid number in Phi or Y axis
NZ:	grid number in Z axis
Rgrid or Xgrid:	grid position in R or X axis
Phigrid or Ygrid:	grid position in Phi or Y axis
Zgrid:	grid position in Z axis

The content of the file is easy to understand even with Notepad.

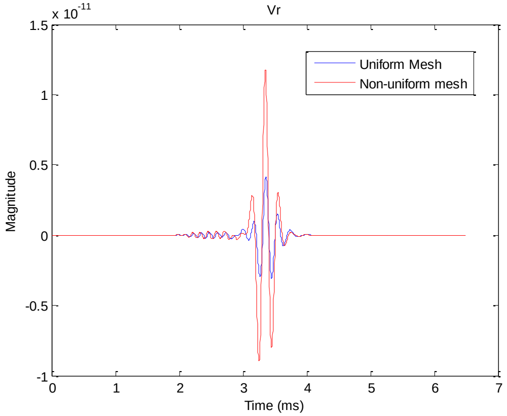
the Possible Result Difference caused by Mesh Difference

In a typical BHA application, ring source will be excited on the interface of device, for example, sonic logging device, steel pipe, cement pipe etc.

Ring source



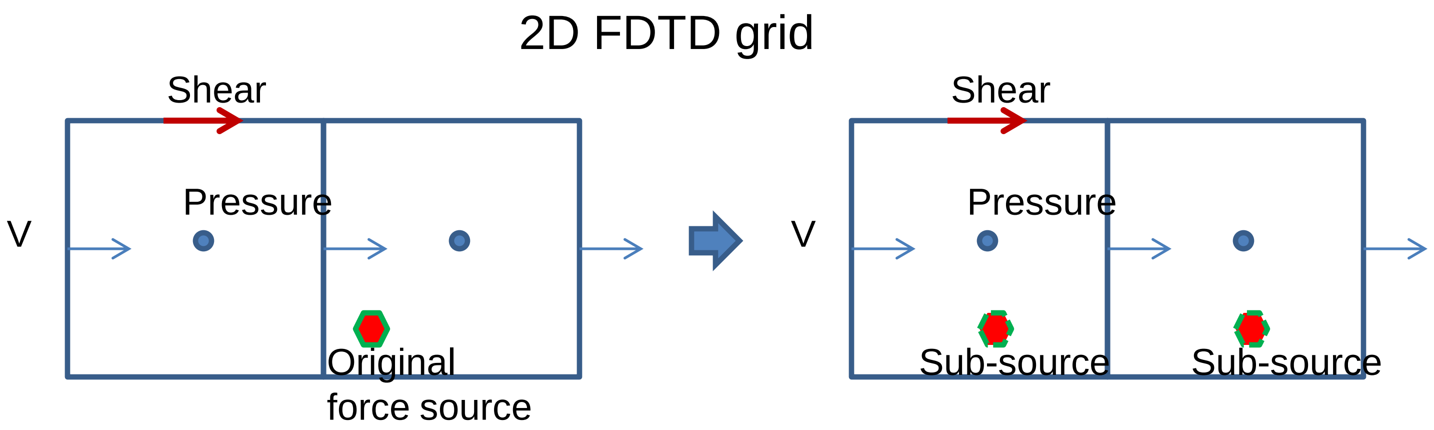
Sometimes, user will find that, for the same simulation project, if the mesh setting change, the simulation results may have an almost the same shape, but with a significant change in magnitude, as following figure.



Actually, all simulation results are correct, the difference comes from the implementation of the source.

The basic implementation of a source in the FDTD method.

Following is a 2D FDTD grid and the components' position in the grid.



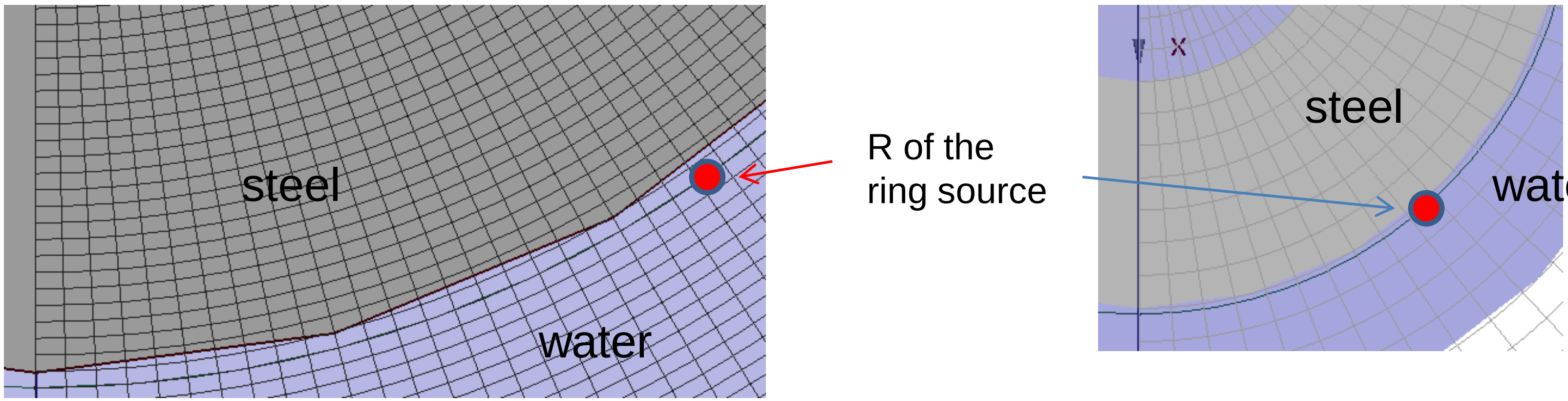
As can be seen, in general, a source (force or velocity source) will not be exactly at a component's position. In order to provide a consistent result with different meshes, this source will be split into multiple sub-sources with corresponding magnitude.

So, we can know

a) If all sub-sources are in the same type of material, the simulation result will be consistent not matter how to change the mesh.

b) If sub-sources are in the different type of materials, the magnitude of the response will change

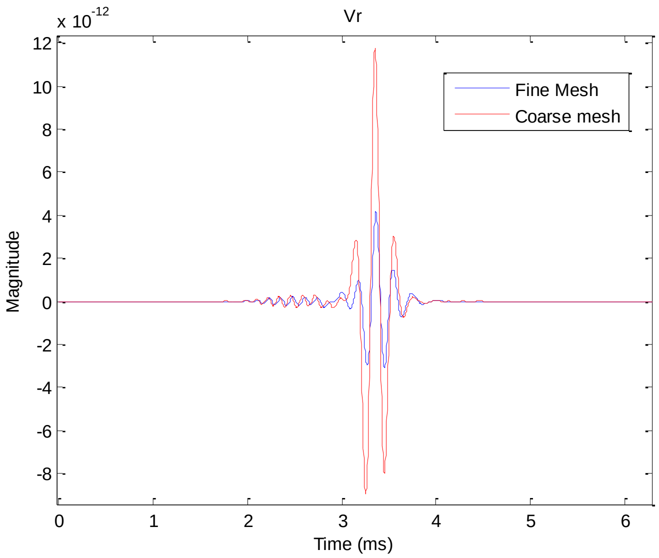
Following is the grid for the same physical model, but the left case uses very fine uniform mesh to capture the interface of solid, the right case use non-uniform mesh.



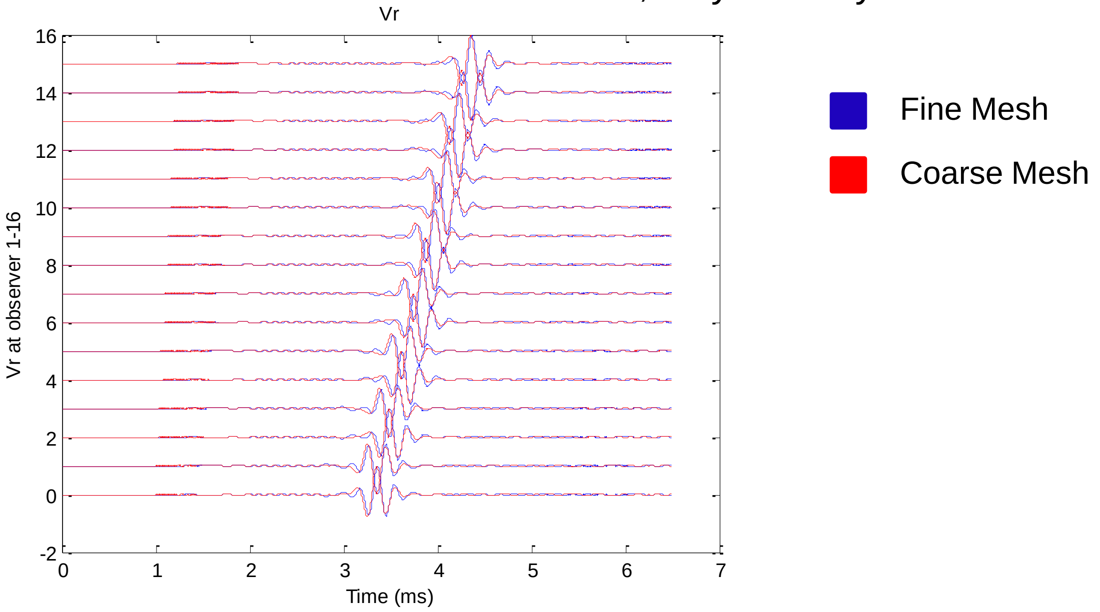
It can be seen, due to very fine mesh, the source in this case is 1 cell away from steel.
Therefore, in the source implementation, all sub-sources are in the water.

It can be seen, in this case, the source is about 0.1 cell away from steel.
Therefore, in the source implementation, some sub-sources are in the water, others are in the steel.

By comparing the simulation result, we can see big difference in magnitude.

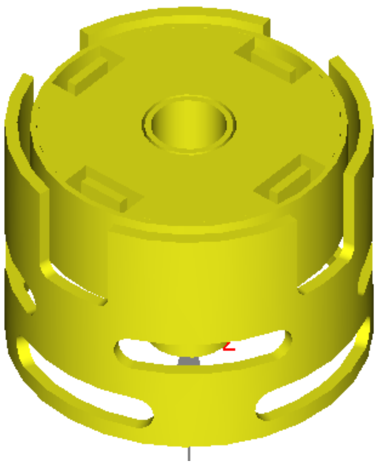


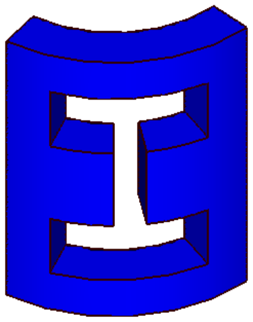
However, if we normalize the waveform to the same level, they are very close.



Capture the Hole of Solids in Mesh

In order to simplify the mesh generation and save the preprocessing time, the default mesh control for a solid is to capture the bounding box.
However, for a complicated solid, there are hole or slot inside the structure, as following solids

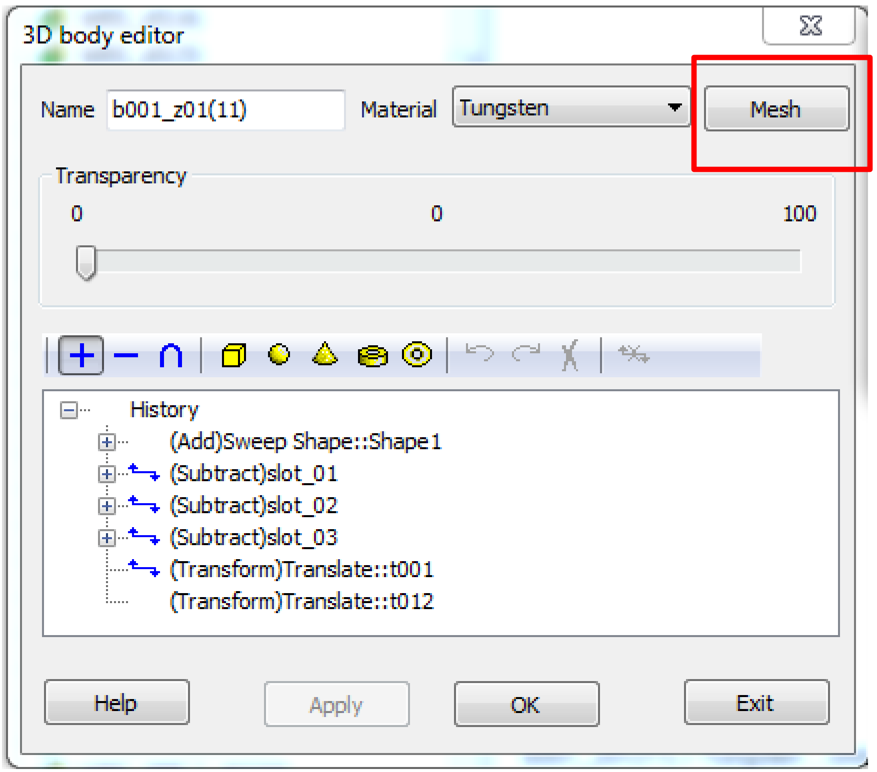


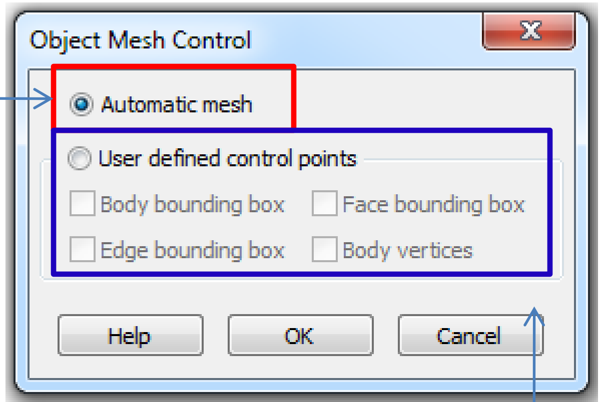


The hole or slot is design for special purpose and will affect the wave propagation significantly. Therefore, in the mesh, these details must be presented correctly. In order to obtain this goal, more mesh controls on the solid must be enabled.

Here, we show how to change the mesh control to affect the detail presentation in mesh.

Solid Editor





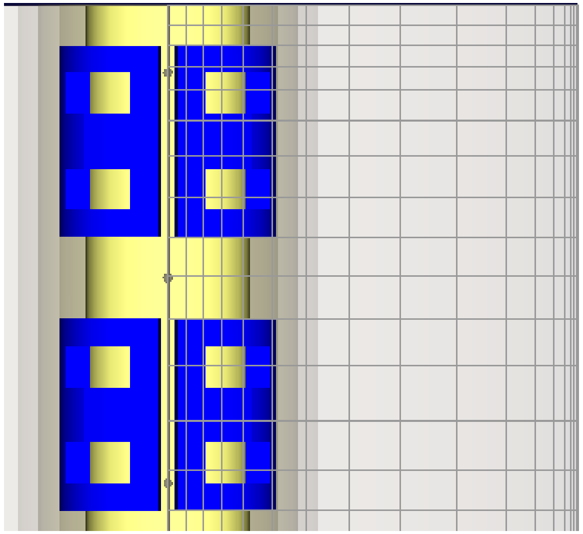
Default setting, GUI will check whether the solid is convex.

For a convex solid, the grid will capture the solid bounding box only.

Otherwise, the grid will capture the bounding box of each face of the solid.

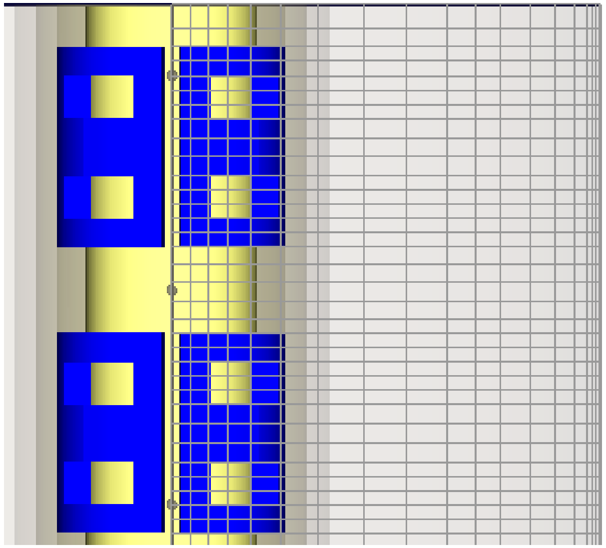
The grid will capture which part of the solid.

This is the mesh capturing the solid bounding box only



This mesh data can't represent solids' hole with accurate position.

This is the mesh by capturing solid faces' bounding box (or using ***Auto*** mesh option)



This mesh data can exactly capture solids' hole.

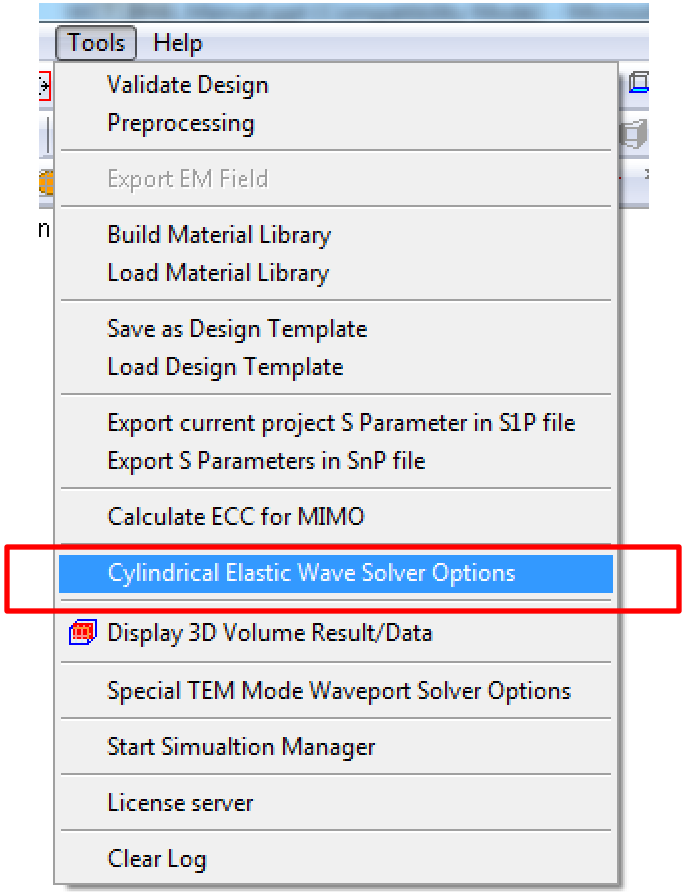
the PML Setting

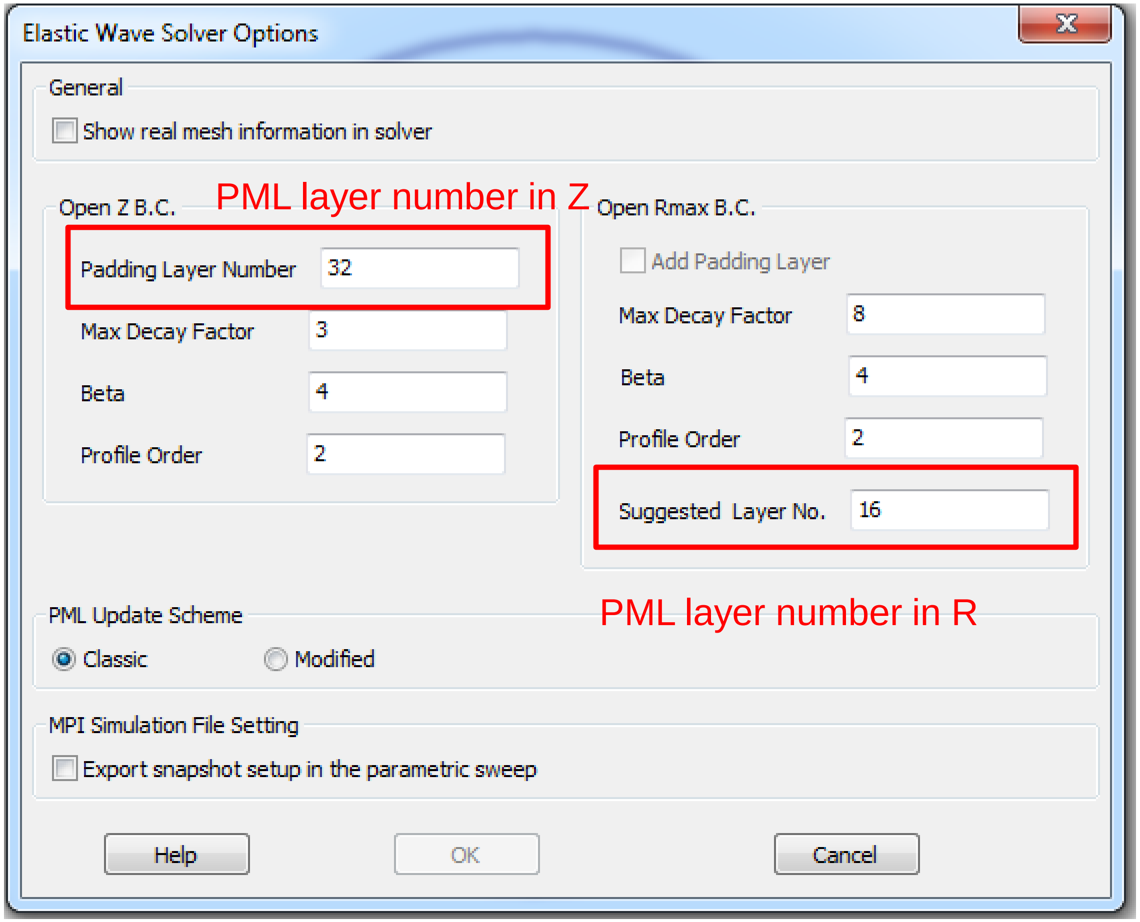
WCT BHA solver employs PML at the absorbing boundary.

In principle, in order to let the PML works well, the total thickness of PML layer will be at least half wavelength. We suggest it should be about 1 wavelength in the BHA solver.

If the project uses the automatic non-uniform mesh with PPW=20, the default PML layer will satisfy above requirement already.

However, if user define a very high PPW, or user define a very fine mesh, the default PML layer will not satisfy above requirement.
In this situation, it is better to change PML layer number.





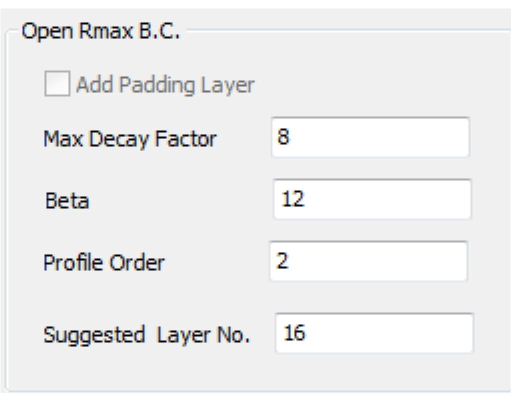
Others are the PML coefficients. Please do not change these values if user don't know the meaning of these parameters.

Special PML Setting for the Background with Small Q values

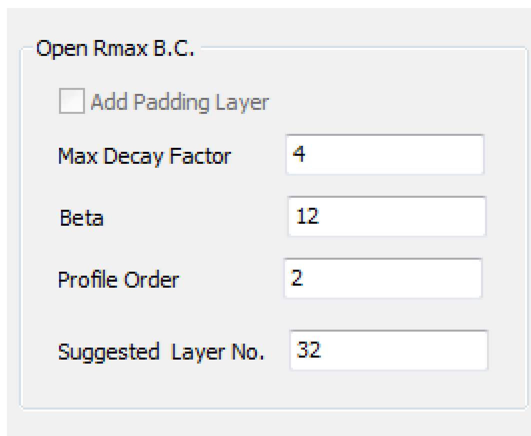
The Default PML setting may be bad if the project set up the background material with a small Q value, for example, Q=20, or Q=5.

Following are the examples of the special PML settings for Q=10, and Q=5.

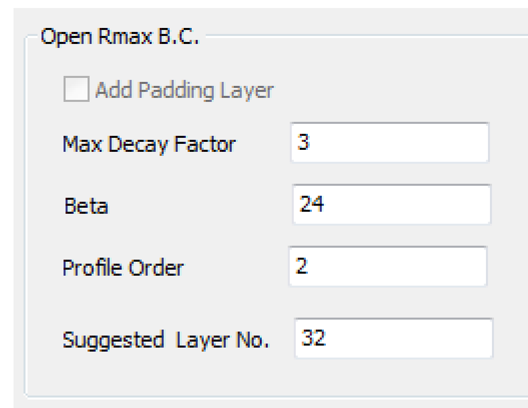
Background Q=20



Background Q=10



Background Q=5



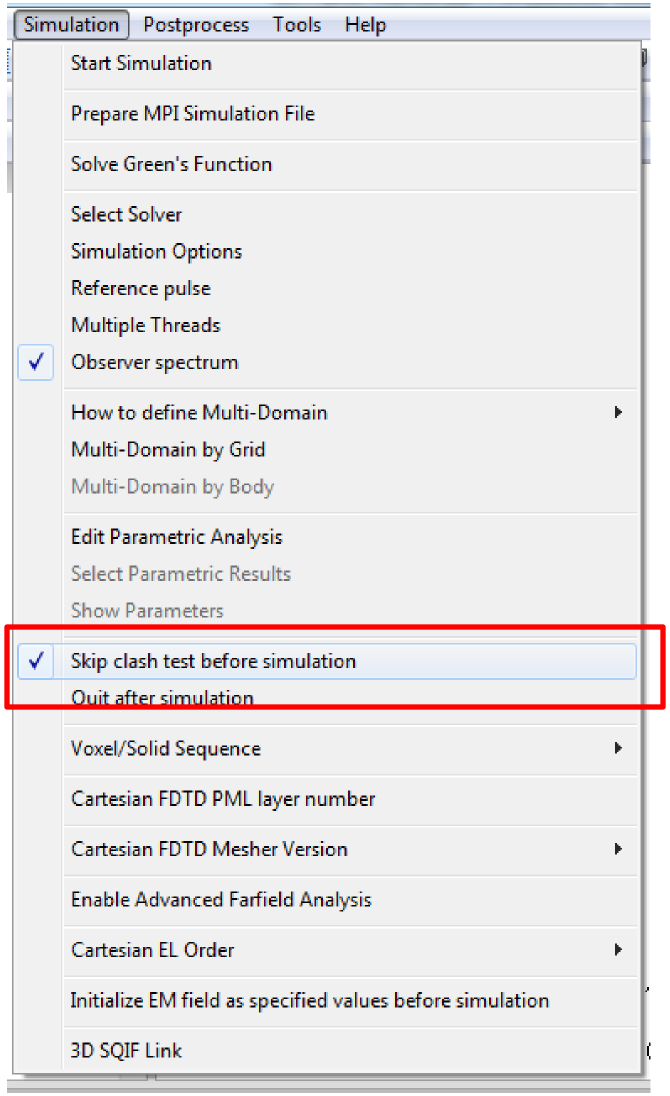
Geometry Clash Test

In some cases, all (or most) solids are imported from other CAD software. In principle, all solids are designed correctly and will not need to be modified.

However, due to the error tolerance in different software is different, for example, a gap or clash in 1e-7 level will be acceptable in other software, but it is not acceptable in WCT. Therefore, WCT will report that there is geometry clash before simulation and ask user to fix it – this means tedious boolean operations.

Actually, this small error will not affect the simulation result. If user can make sure the solids in the project are correct, he can skip the clash test before simulation, as the figure shows.

(Note: if there is solids imported from external CAD data file, this option will be automatically enabled)

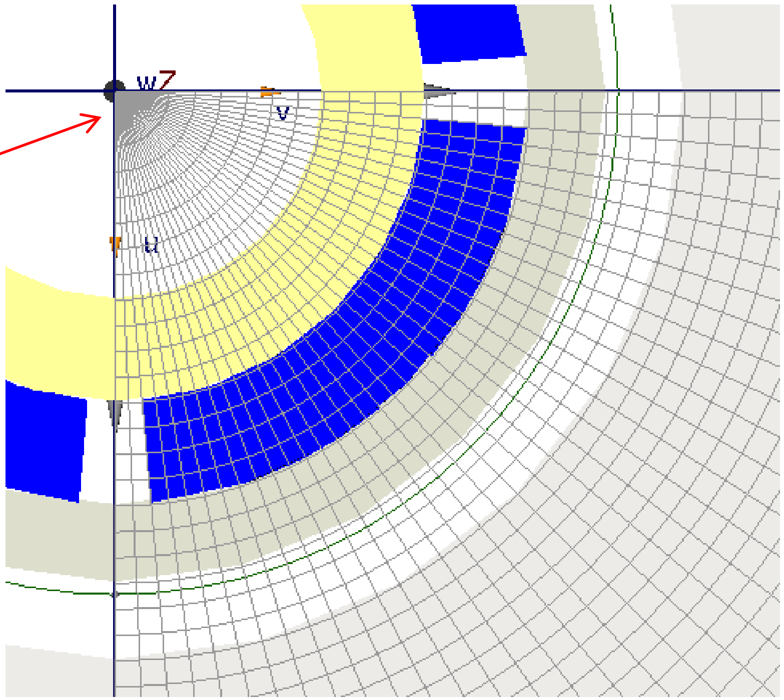


Tips to Increase the Simulation Δt

The WCT BHA solver employs the FDTD method. Therefore, it requires a max Δt in the simulation. Basically, the Δt is determined by the combination of the cell size and the wave speed in this cell.

- larger cell size will increase Δt
- faster the wave speed in a cell will decrease Δt

A typical uniform mesh grid. As can be seen, the cell size at R=0 is very small



1) In a cylindrical system, the cell size is proportional to the R and ΔR of a cell. So, except extreme thin pipes (ΔR will be very tiny), in general, smallest cell is located at R_{min}, especially for R_{min}=0 situation. Therefore, a relative larger R_{min} can increase Δt. According to our experience, for a uniform R grid setup, if we set the $R_{min}=\Delta R$ with Hard boundary at R_{min}, the Δt will be 1.5 times bigger than R_{min}=0 setup. And the simulation result has about 1.5% difference only. But due to Δt increases with 1.5 times, for a fixed time window, the total simulation time will be 67% of that in R_{min}=0 setup.

2) If possible, don't place solid with high speed material at R_{min} region, because it will decrease the Δt. For example, if there is a solid steel pipe cross R=0, the Δt will be 3 times smaller than a small water hole inserted into this solid steel pipe. In general, due to the wave in R=0 region is very weak, and the wavelength of the wave is very large compared to the R of the inserted water hole, the simulation result will be almost the same. But the total simulation time has 3 times difference.

Q & A for Licensing Problems

- What is WCT EM & EL package?
  - WCT EM & EL package is a general GUI to simulate multiple physics. It supports following solvers by the same GUI
    - Cartesian EM
    - Cartesian EM Imaging
    - Particle in cell
    - Green's Function for Layer media
    - SEM solver for Lithography
    - Cartesian EL
    - Cartesian EL Imaging
    - Cylindrical EL for Borehole acoustic (BHA)

- OS requirement
  - It only pass the tests in Window 7 x64 & Windows 10 x64, and it is x64 version only.

- Where can get WCT EM & EL package?
  - Please go to: http://www.wavenology.com/download.html and download the installation package with the download key provided by Wave Computation Technology.

- What is in the WCT EM & EL package?
  - setup.exe & Data.msi

- How to install WCT EM & EL package?
  - Unzip the installation package and run “setup.exe”. In most cases, the installation has not other requirements. If there is error, please refer to the “Installation” section in the WCT download page.

- What is WCT license?
  - To start a simulation in GUI, the corresponding engine need to be unlocked. WCT GUI uses a license to unlock the engine.
  - A license is bound to a MAC provide by user and can be used on the computer with that MAC only.

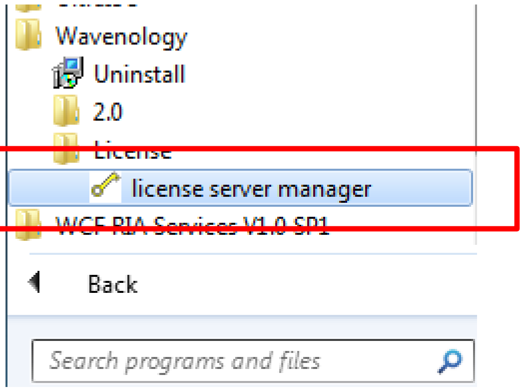
- Does WCT GUI requires license?
  - Not, WCT GUI does not require license. After installation, the GUI can be used for modeling purpose, or load the existing WCT project to check existing simulation result.

- How to obtain a license?
  - Please contact WCT to purchase a license.
  - Or register in WCT website to obtain a free license for 30 days evaluation purpose.

- Whether there is multiple WCT license types?
  - Yes. There are two types of license: fixed node or floating
    - A fixed node license can be used on the computer with a specified MAC.
    - A floating license should be installed on the computer with a specified MAC, but other computers can link to this computer to unlock the engine.
  - Basically, one WCT license file can unlock one or multiple engines, not matter it is fixed node or floating.

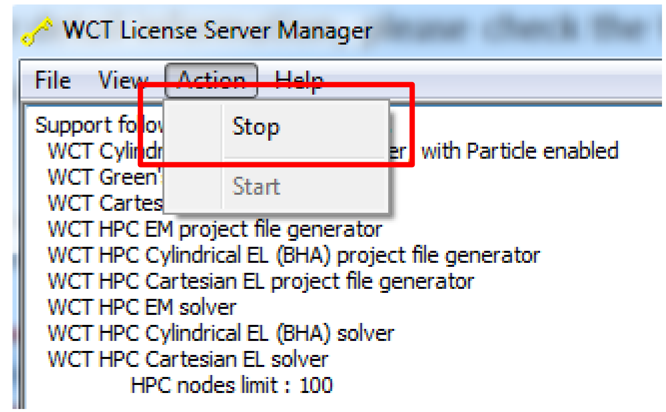
- What is the WCT license server?
  - WCT license server is a service installed in OS after WCT EM & EL package is installed.
  - It is used to manage the license.
  - The WCT GUI need to communicate with it to unlock the engine.
    - Without the WCT license server, the simulation can't start.

- How to load a license?
  - For detail information, please check **the WCT licensing Manual**
  - A simple procedure is:

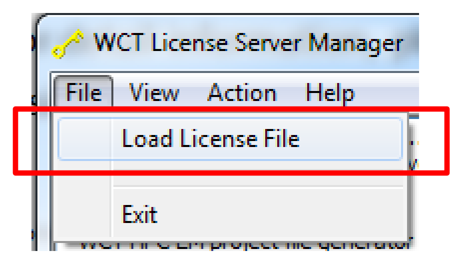


Run as administrator

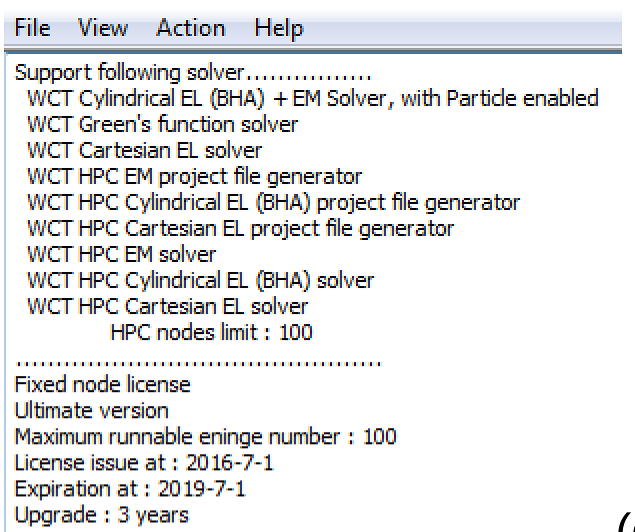
Stop the license manager



Load the license file provided from WCT

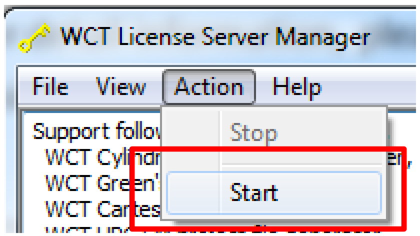


Make sure similar information shown

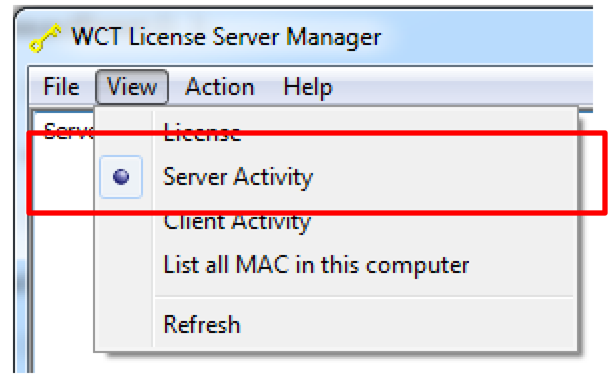


(go to next page)

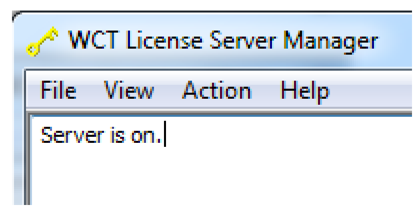
Restart the license manager



check the server status



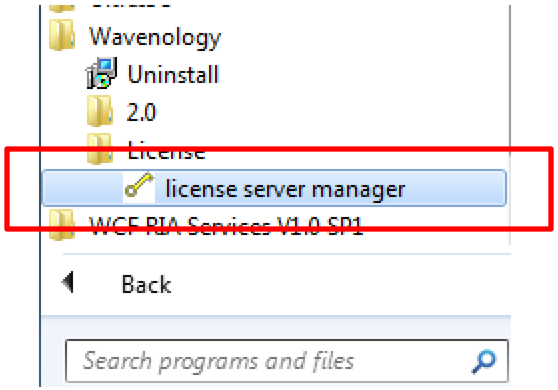
Make sure the serve is ON



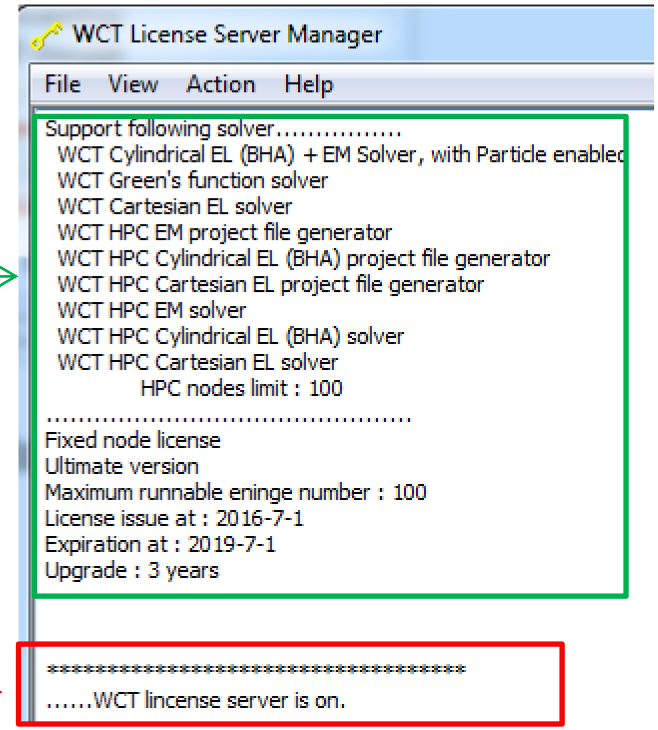
If reach this step, the license is loaded correctly and the license server works well.

User can quit the license server manager.

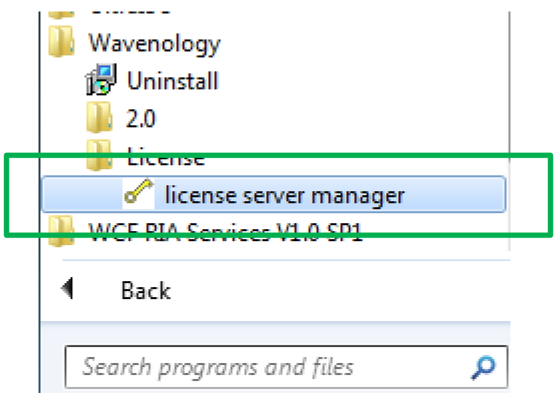
- How to check the status of WCT license server?



Run as administrator



- How to check the status of a license?



Run as administrator

- Make sure the solver type and the time are correct.

- What is the meaning of following information?
  - Sometimes, user can’t start a simulation and get following information in the log.

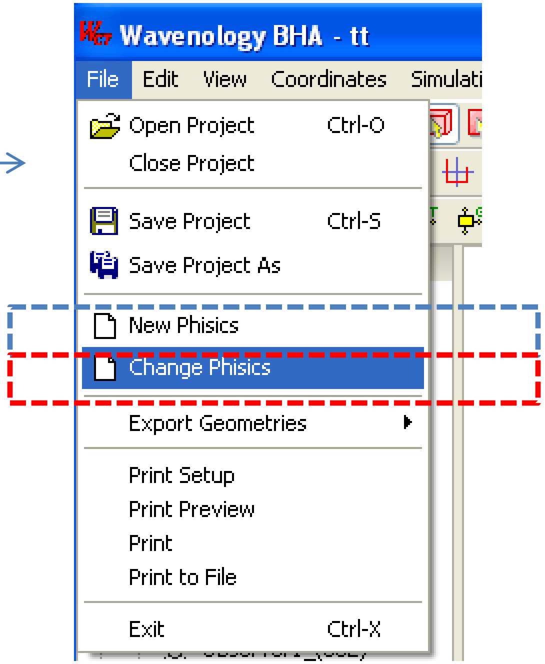
```
..........Acquiring a license for --the BHA solver-- to start job (simulation or others) at Sun Apr 28 22:55:11 2019 ..........

**************!!!!!!!!!!!  Failure in license request, Job (simulation or others) can't start. Sun Apr 28 22:55:12 2019   !!!!!!!!!!!**************
.........It may be due to one of the following reasons: 1. Other copies of WCT are running simualtion; 2. the IP of the WCT license server is wrong; 3. the WCT license server is turned off; 4. the requested solver is not supported by the license
.......
.........Corresponding solutions: 1. Wait until other copies of WCT finish simulations; 2. set a correct IP for the WCT license server; 3. turn on the WCT license server........

.........*** More detail information about the failure: Fail to connect to server. ***.........
```

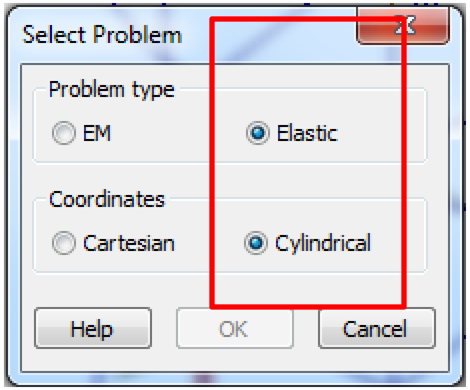
It means that the GUI can’t unlock the engine to start the simulation, the reasons could be:

- The license is not loaded correctly.
- The requested engine does not match the license.
- The license expires already.
- The requested simulation number excess the register number in the license.
  - For example, if the license support 2 simulations simultaneously, the request for the 3rd simulation will be rejected. User must wait until one of existing simulation finish.
- The license server is OFF, or does not exist.
  - An known problem is, in Windows 10, after OS upgrade, the WCT license server may be removed by OS. In this situation, user need to un-install WCT and re-install it. But in this situation, user does not need to reload the license.
- For a floating license, the GUI can’t connect to the license server.
- If the GUI is terminated manually, the license server need to wait 15 seconds to make sure whether the GUI is still alive and just lost connection. So, a new simulation should wait 15 seconds to start a new simulation.

Tutorial Cases

To build a BHA project, please make sure the GUI has been switched to the Cylindrical EL mode (BHA mode)





Or, user can load a existing BHA project from the BHA demo package, then **"Close"** it to make a new BHA project.

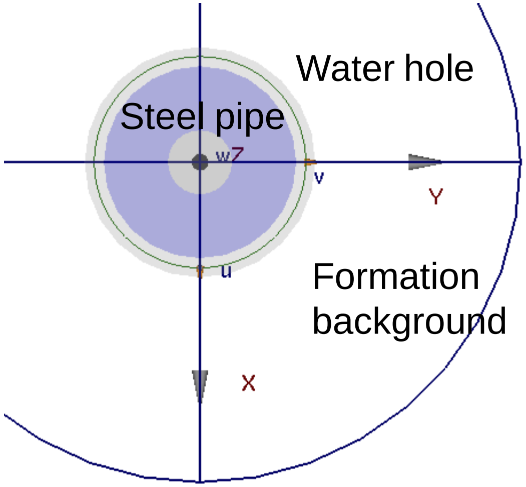
Case1
A steel Pipe in Borehole

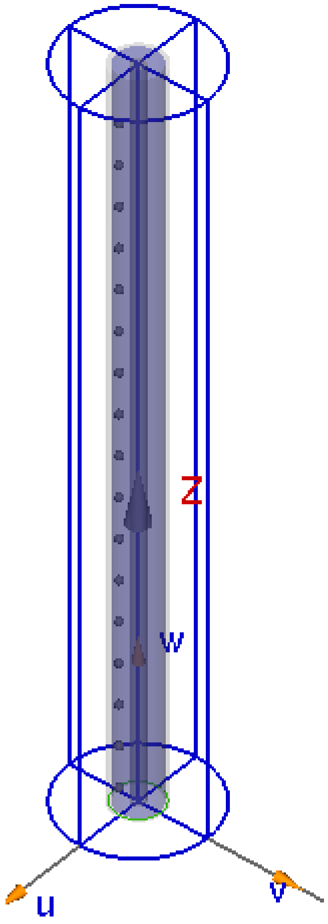
This case is used to show how to build a basic borehole with a steel pipe inserted inside, then a ring dipole source is excited and check the wave propagation

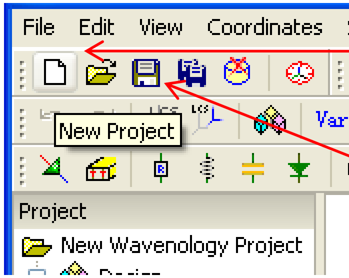
The idea to build this case is
- formation background
- a water hole as the borehole
- a steel pine is placed in the borehole

For the borehole & the steel pipe, there are 2 ways to build them
1) One water cylinder as the borehole, a steel cylinder is placed into the borehole – we need to let borehole **Subtract** the steel pipe
2) Because the water borehole is split by the steel pipe into 2 cylinders, so, there are 3 independent cylinders in the project.

Here, we use method 2 to build the project.

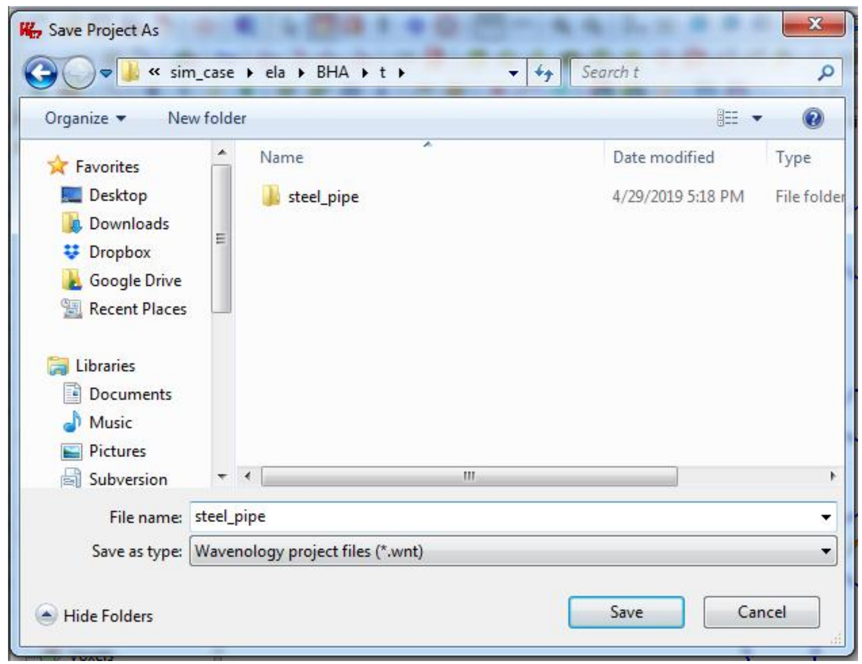


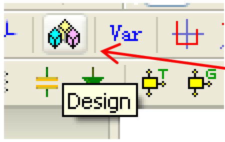


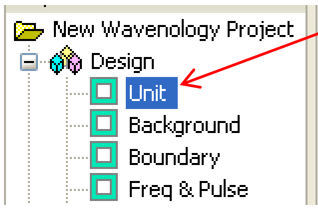


Use “New Project” button create a new project

It is better, then, to save the project with a name under the target folder. Here, we define the project name as “steel_pipe” under folder “steel_pipe”. Because in order to generate the mesh data file, the software need a predefined folder to export them. Without this predefined folder, the data file will be saved in some system default folder and will be hard to find.

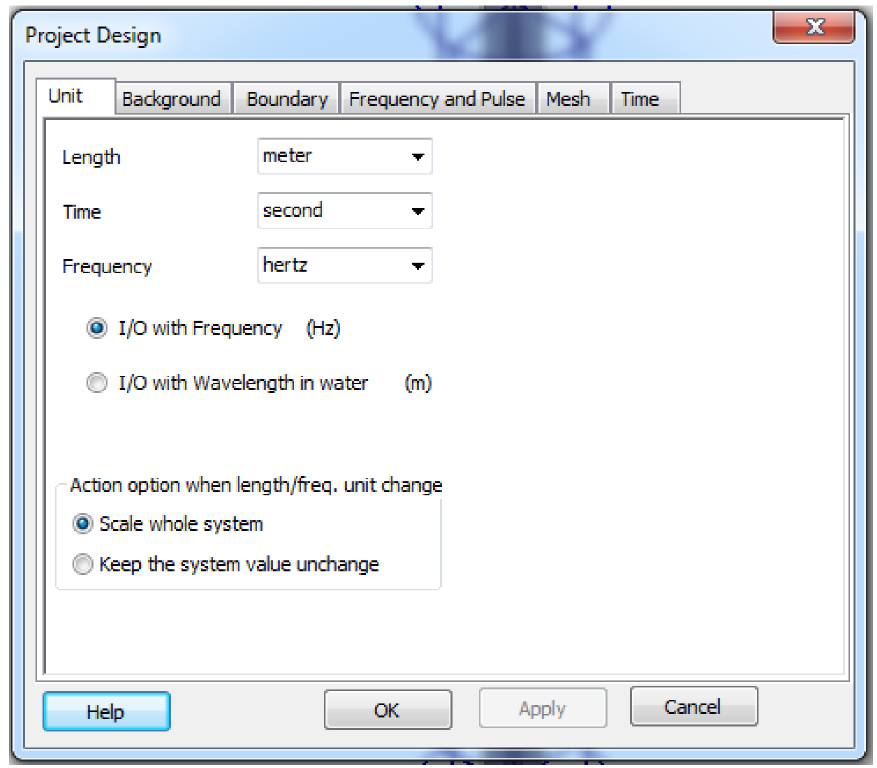




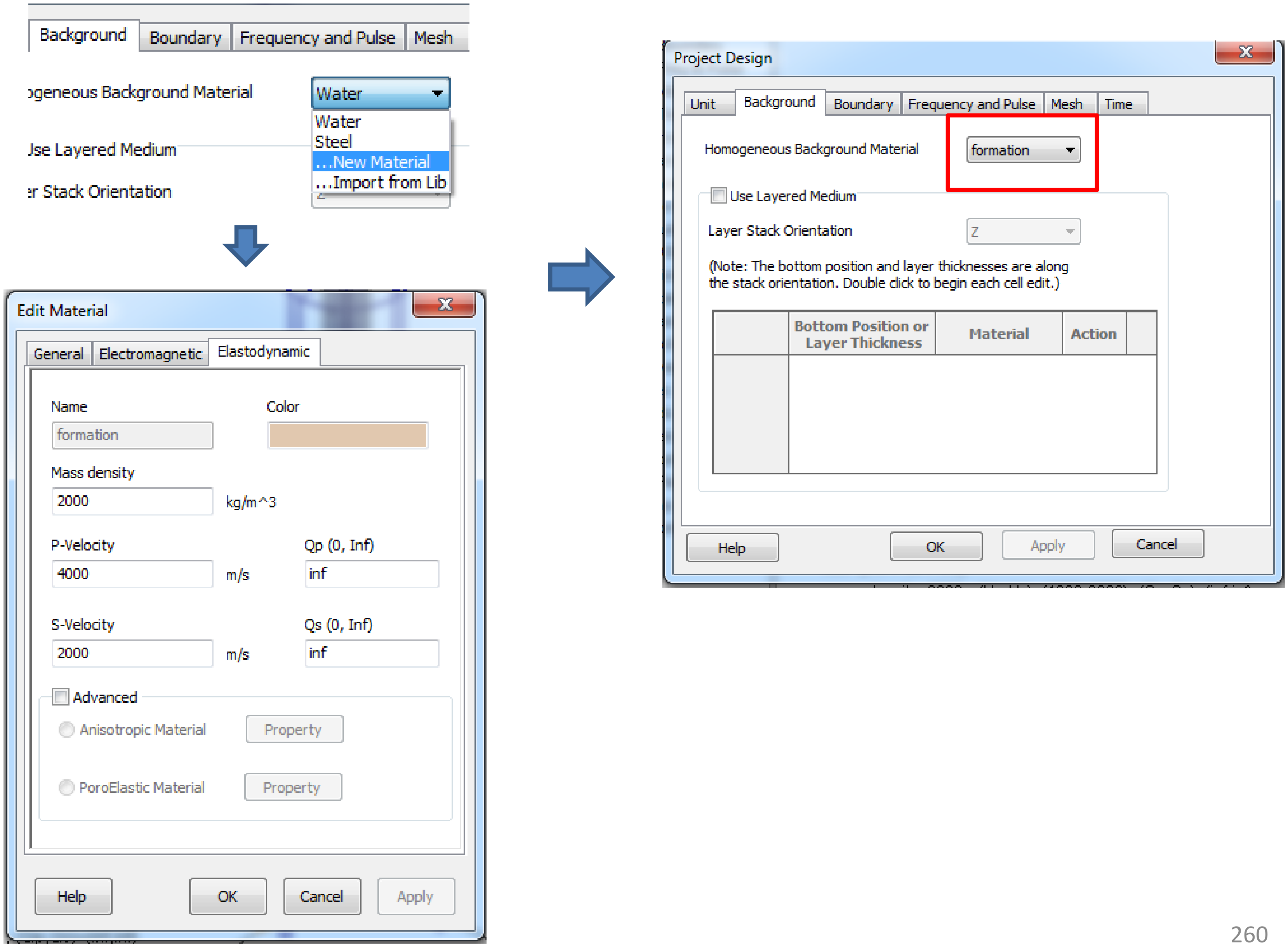


Use “**Project Design**” button or “**Unit**” treenode to modify project unit

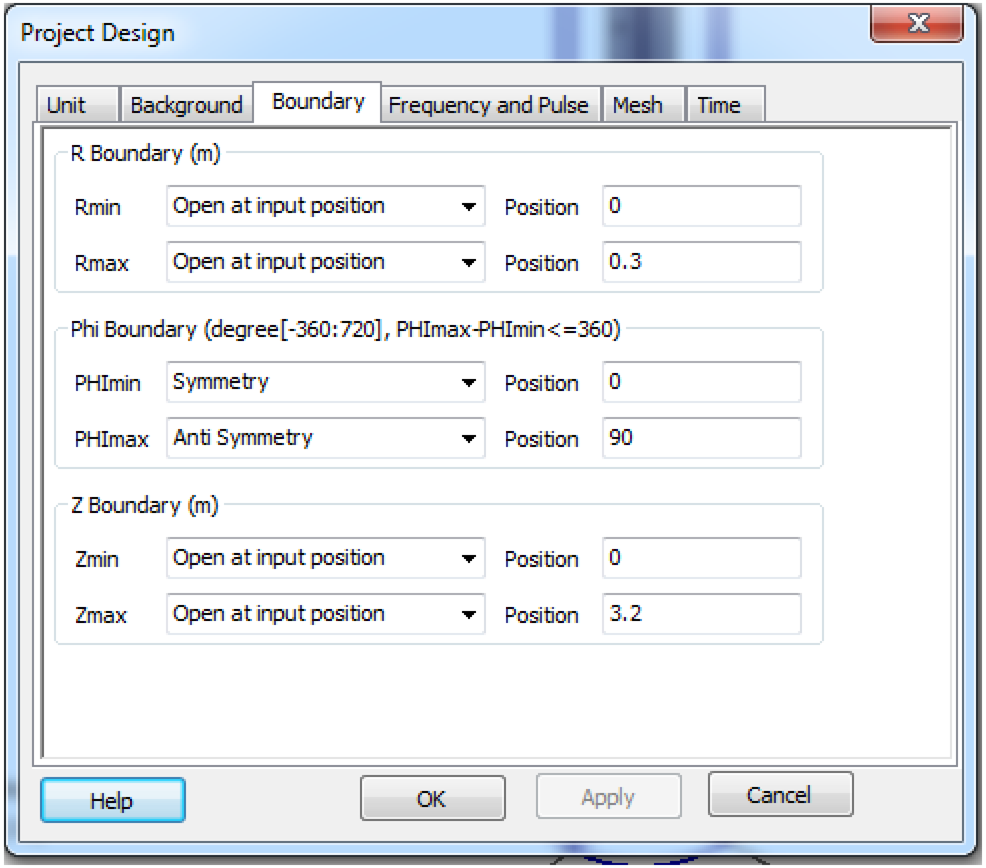
Define project unit as “m”, “s” and “Hz”. Other settings use ***Default*** values



Define the background as a new material "Formation"

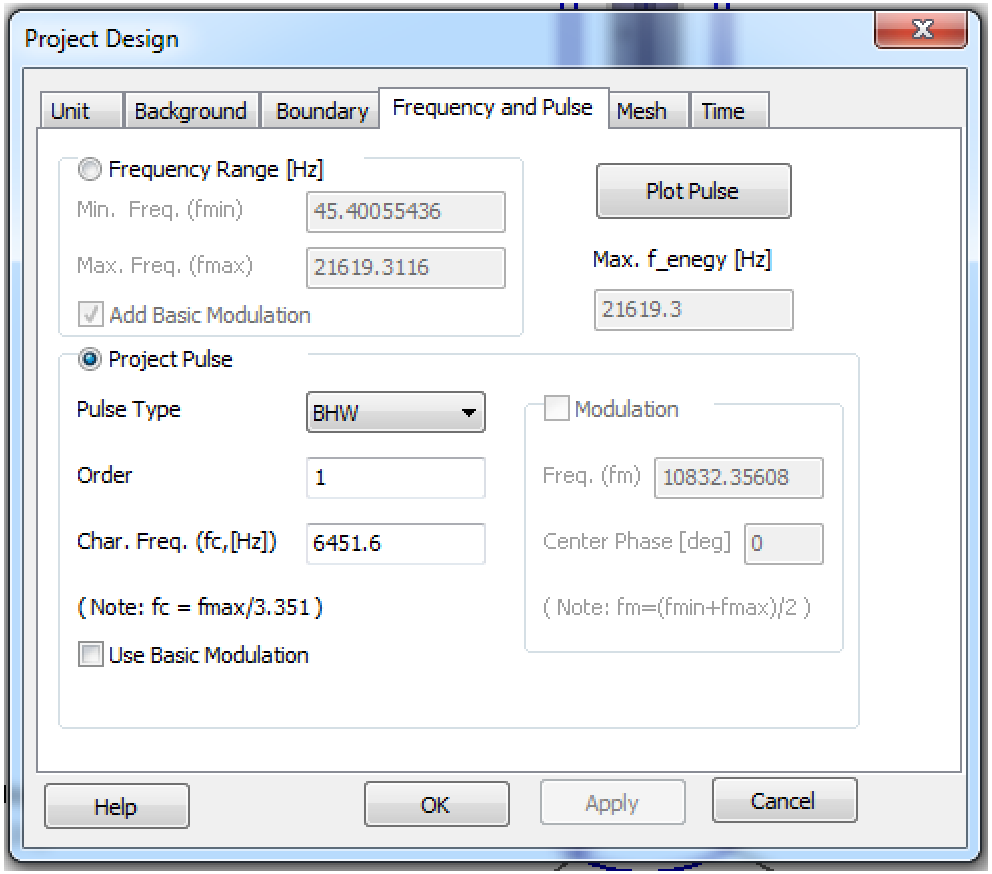


The boundary position & the boundary conditions are defined as following.



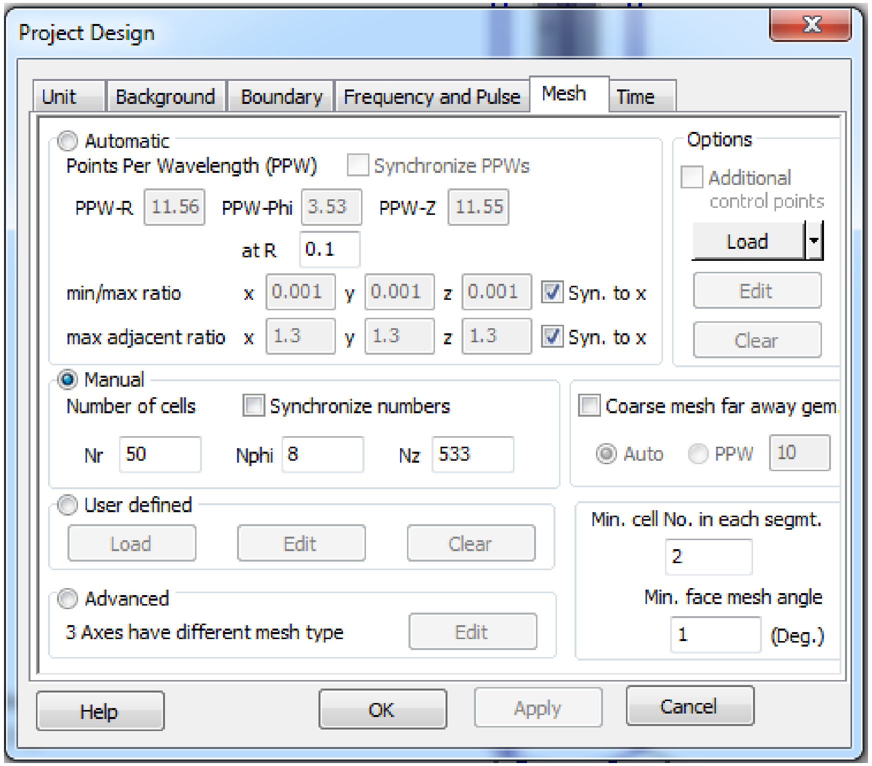
Because we will use a ring dipole to simulate a cylindrical pipe, based on previous explanation, we can use a quarter circle [0, 90] to represent the full circle.

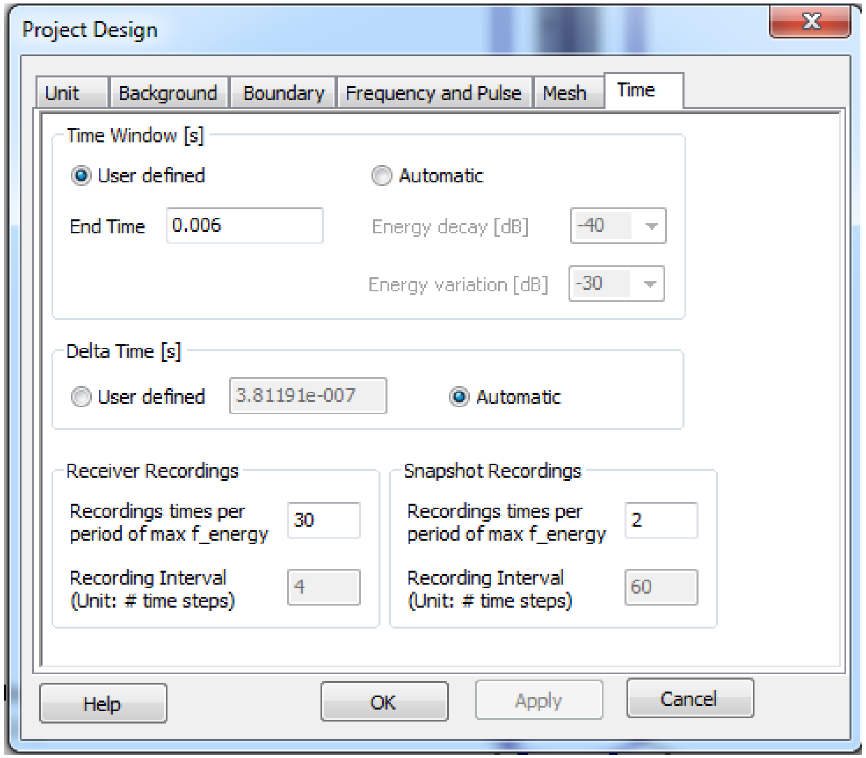
Define project pulse as 1st order BHW pulse with f_{max}=21.6 KHz, f_{min}=45 Hz



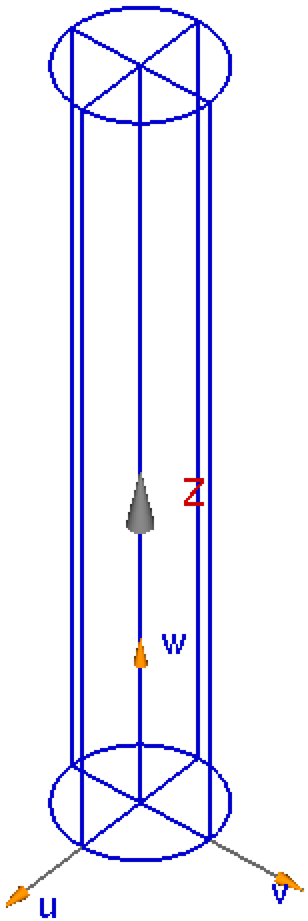
Uniform mesh

Automatic Δt to run 0.006 s

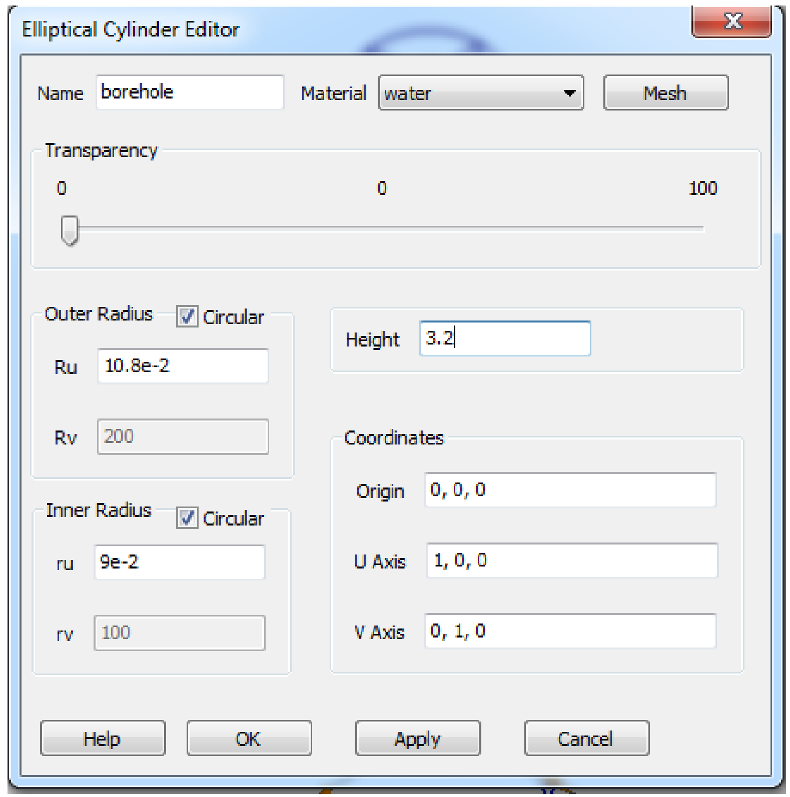


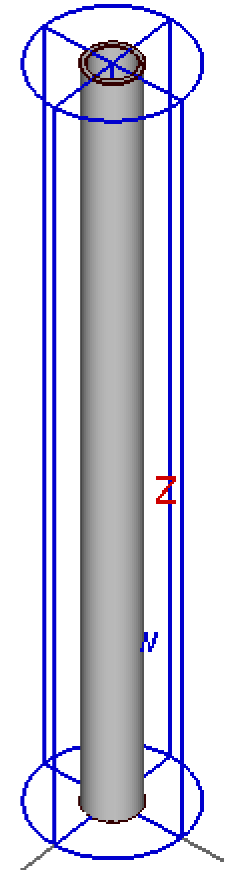


We get the project space shown as

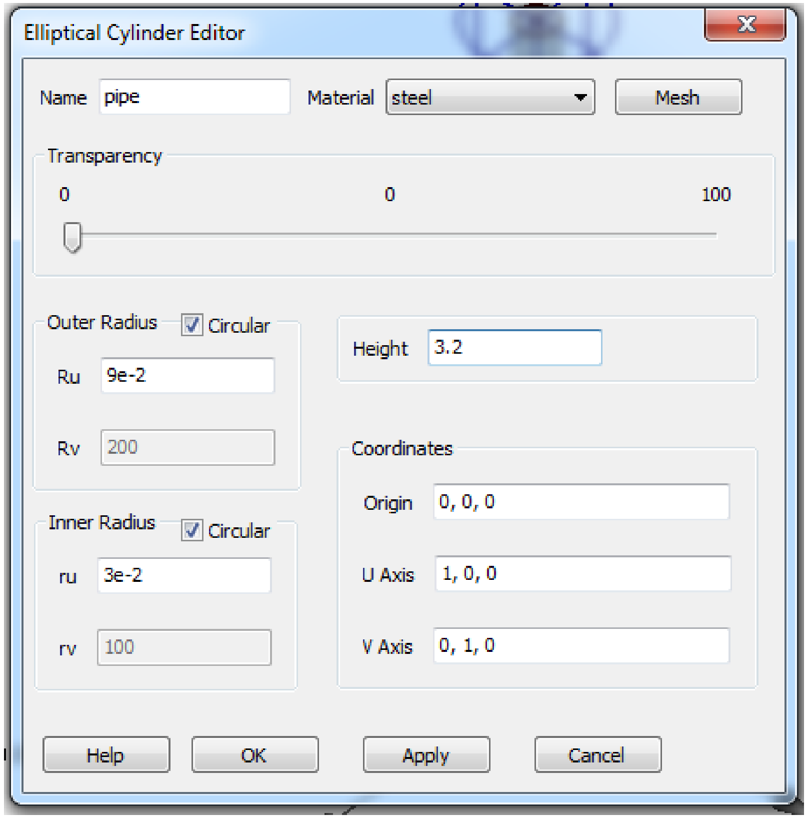


Then we build the borehole's outer part

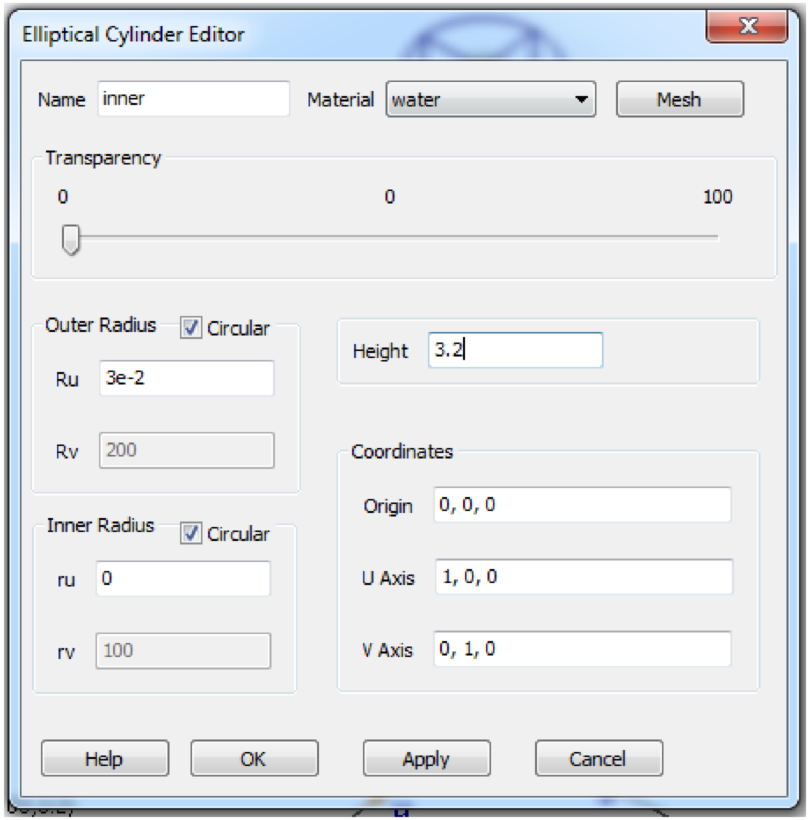




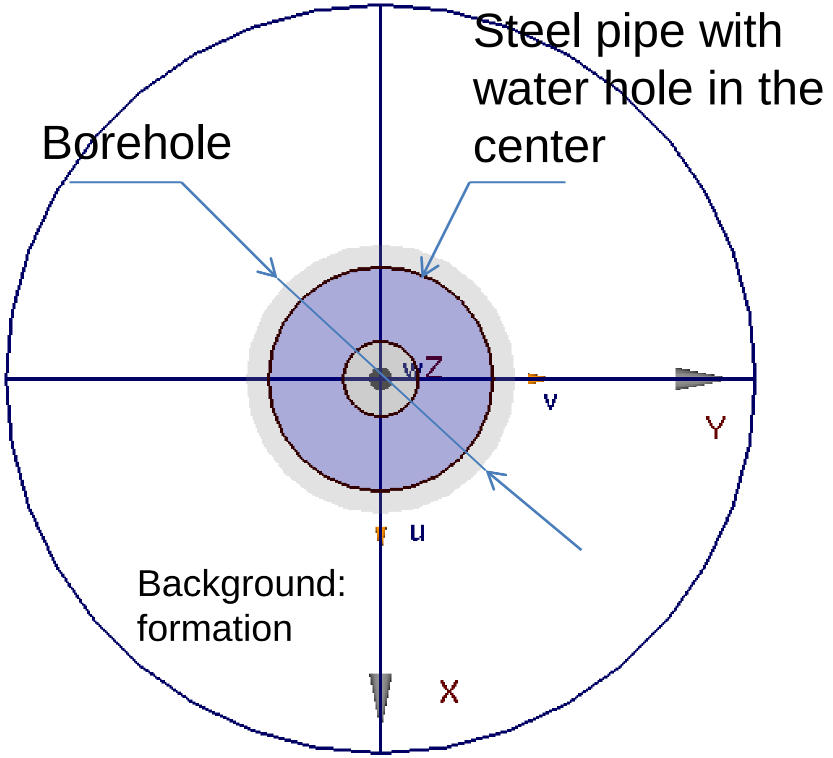
The steel pipe



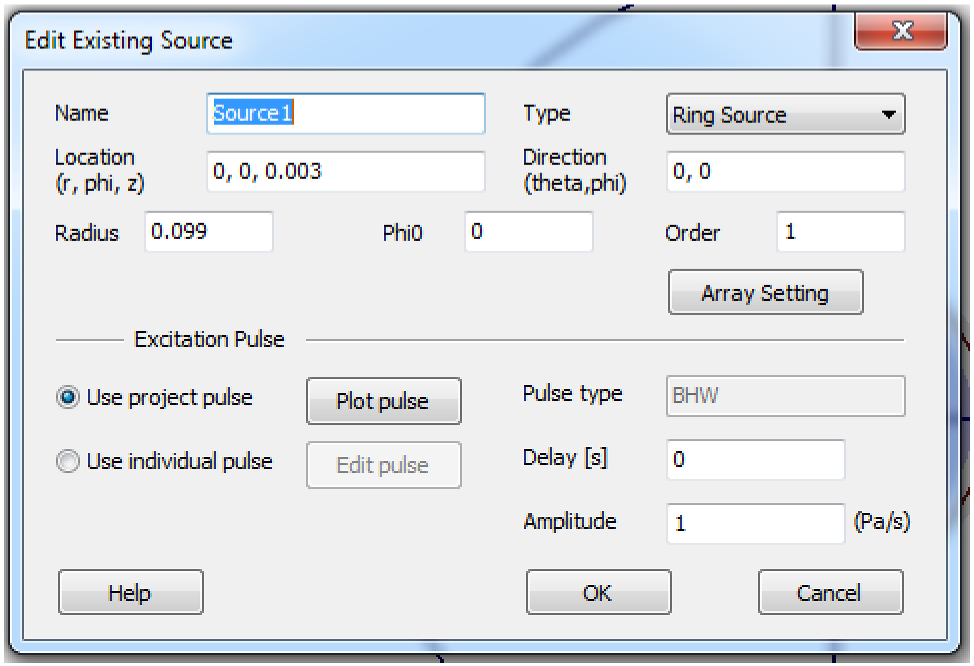
The water hole inside the steel pipe



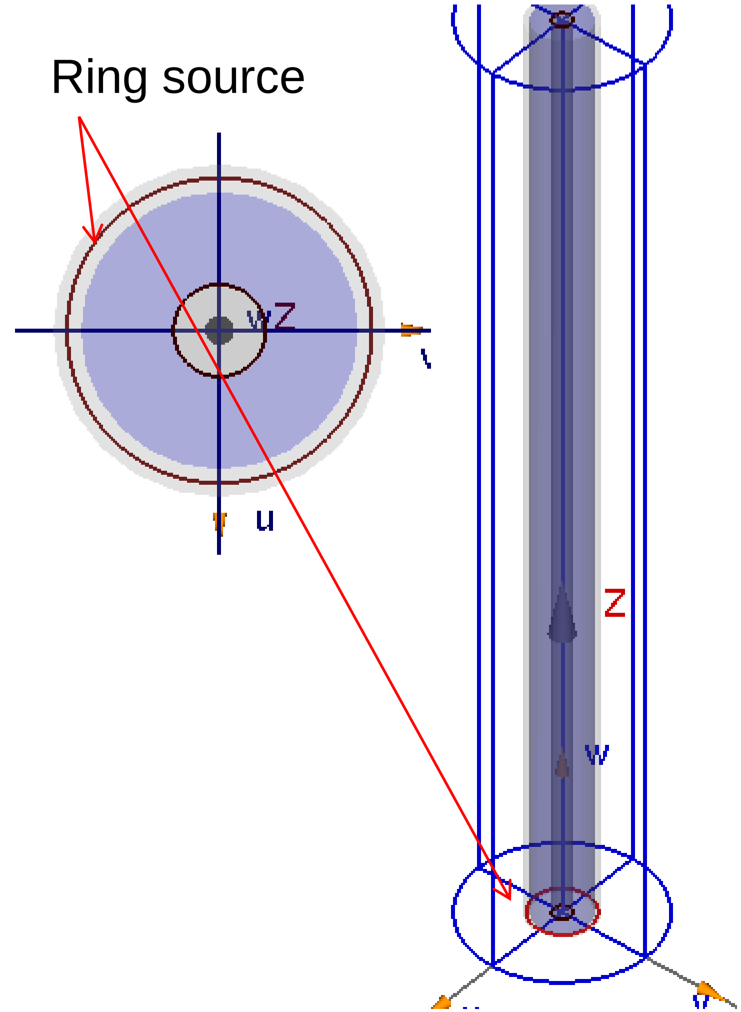
The layout of the structures



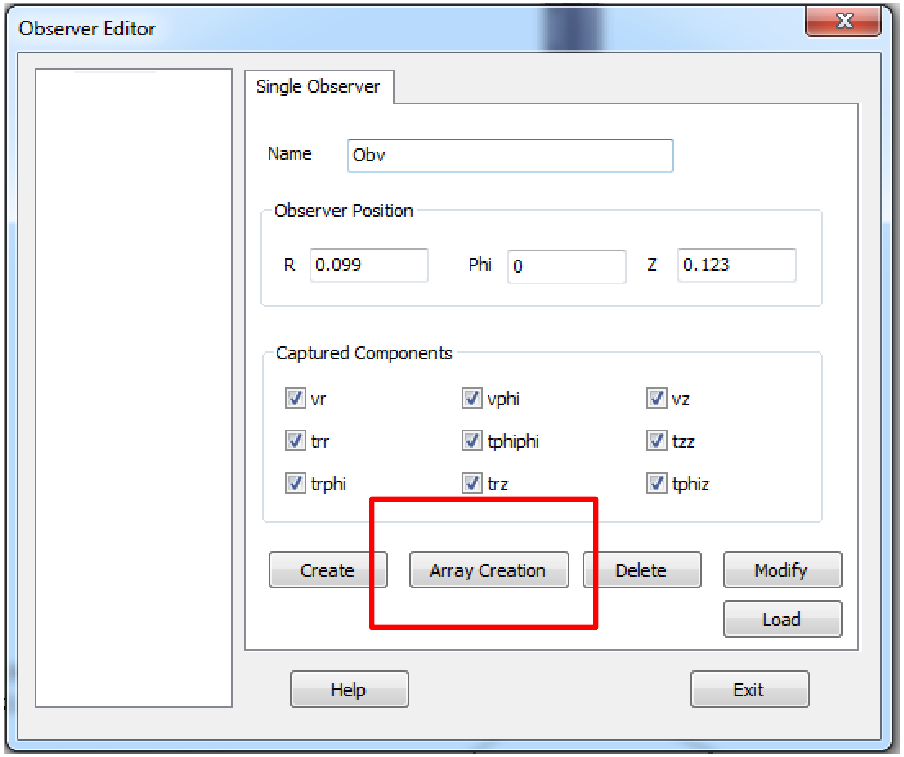
Define a ring dipole source

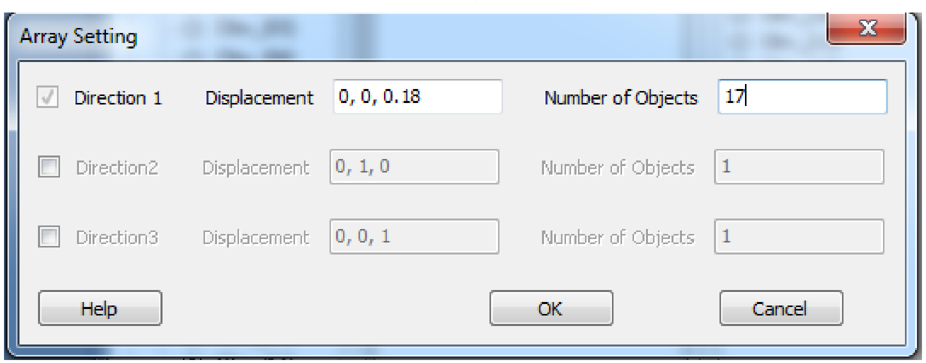


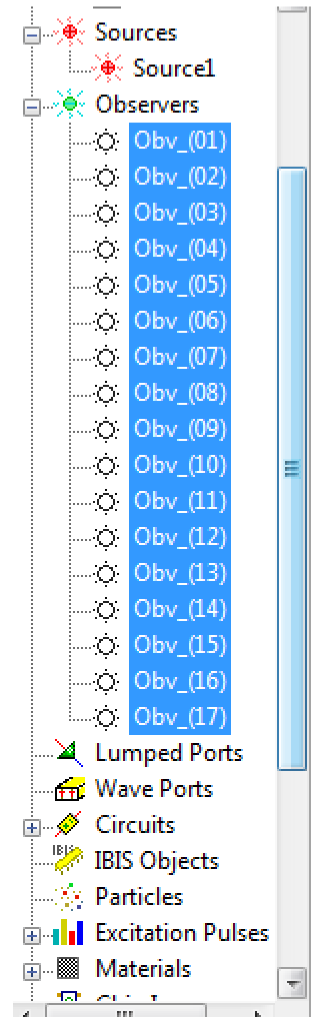
Ring source

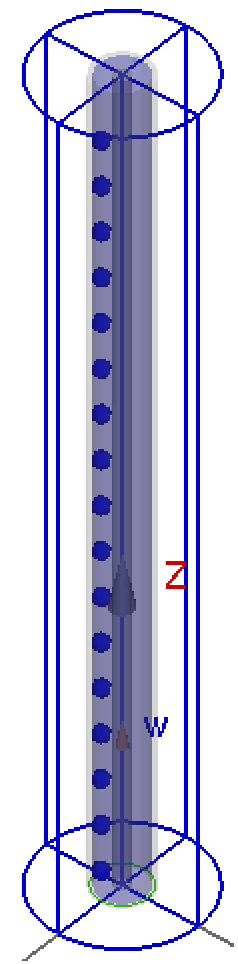


Define an array of observers to record the field

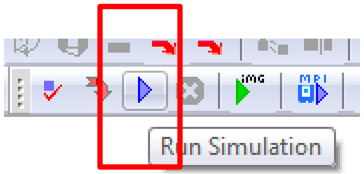


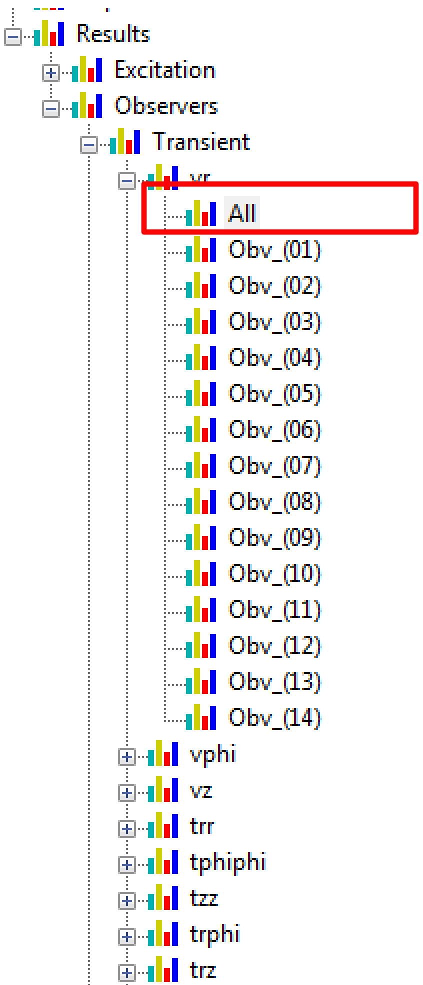






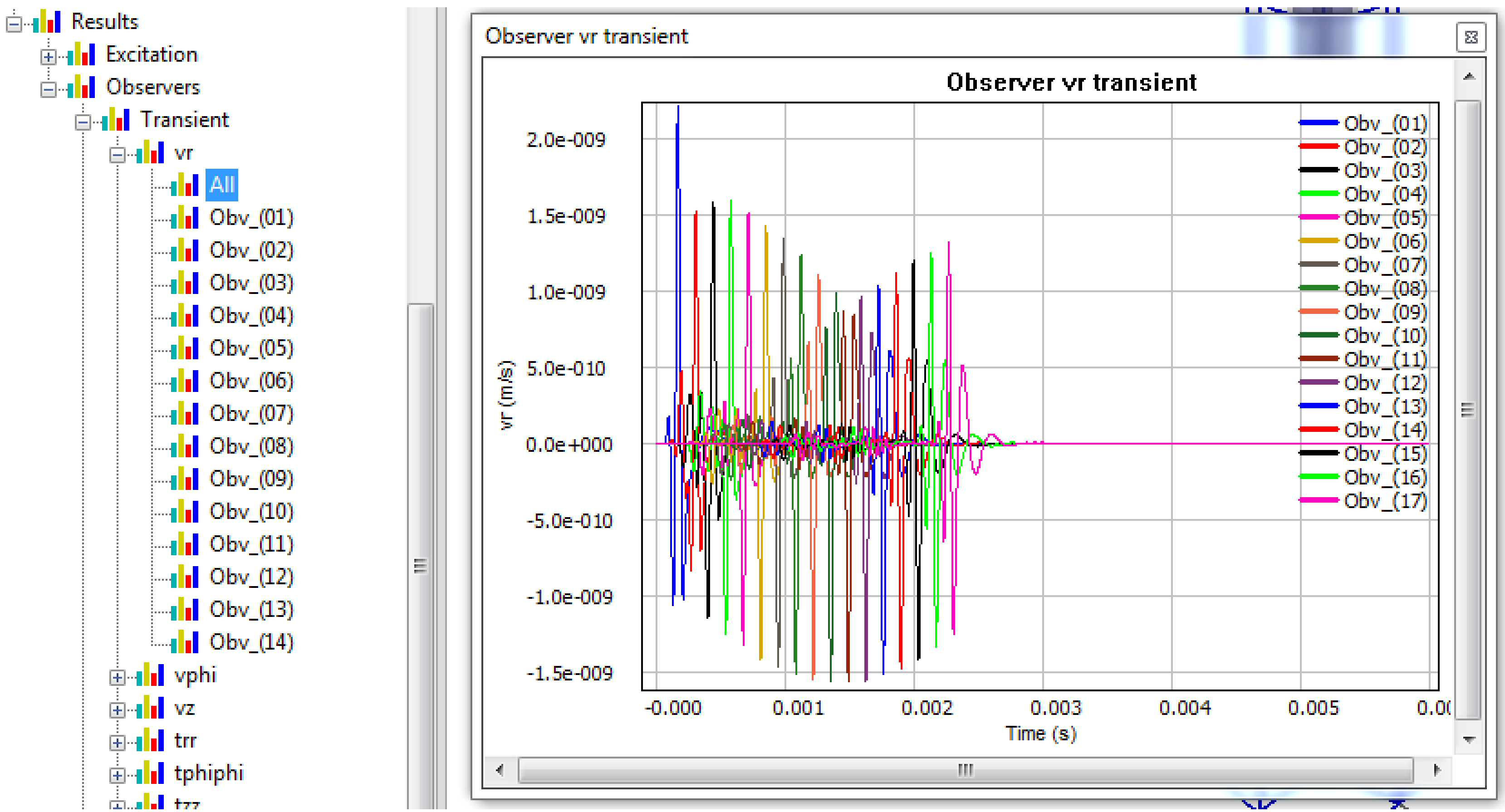
Then, we can start the simulation & check the result.





Double click the "All" node under Vr to show all Vr traces in the canvas

All transient Vr shown in one figure

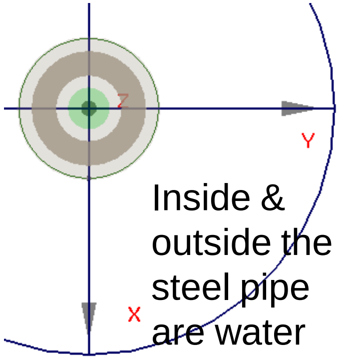


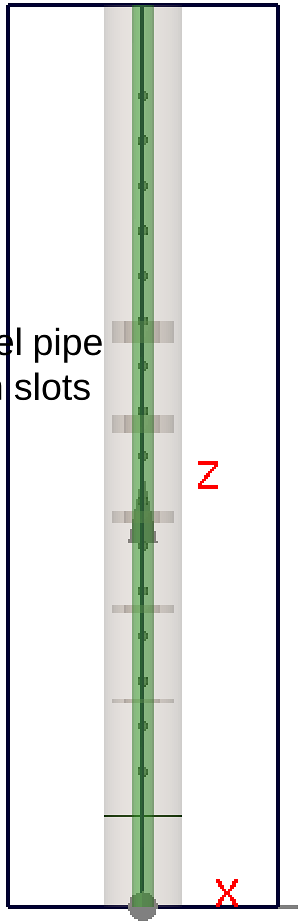
Case2
An insulator

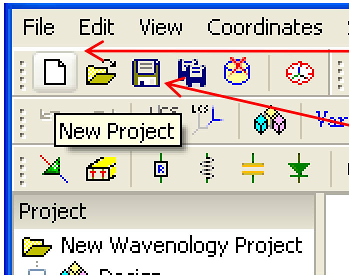
This case is used to show how to build a steel pipe with multiple slot working an insulator, then place this insulator in the water to check the wave propagation.

This case include

- ➢use the Boolean operations to build a complicated solid
- ➢ use Automatic mesh to capture the detail of a complicated solid
- ➢ use snapshot to check the wave propagation

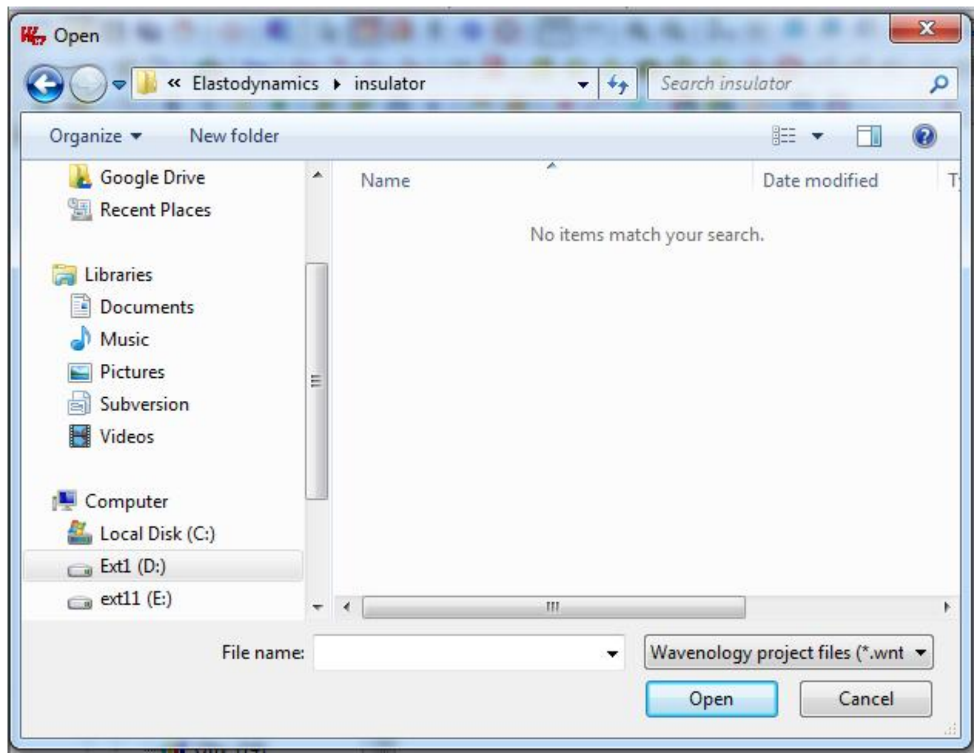


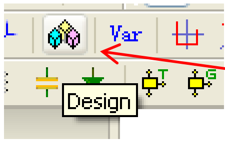




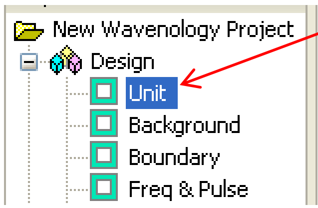
Use “New Project” button create a new project

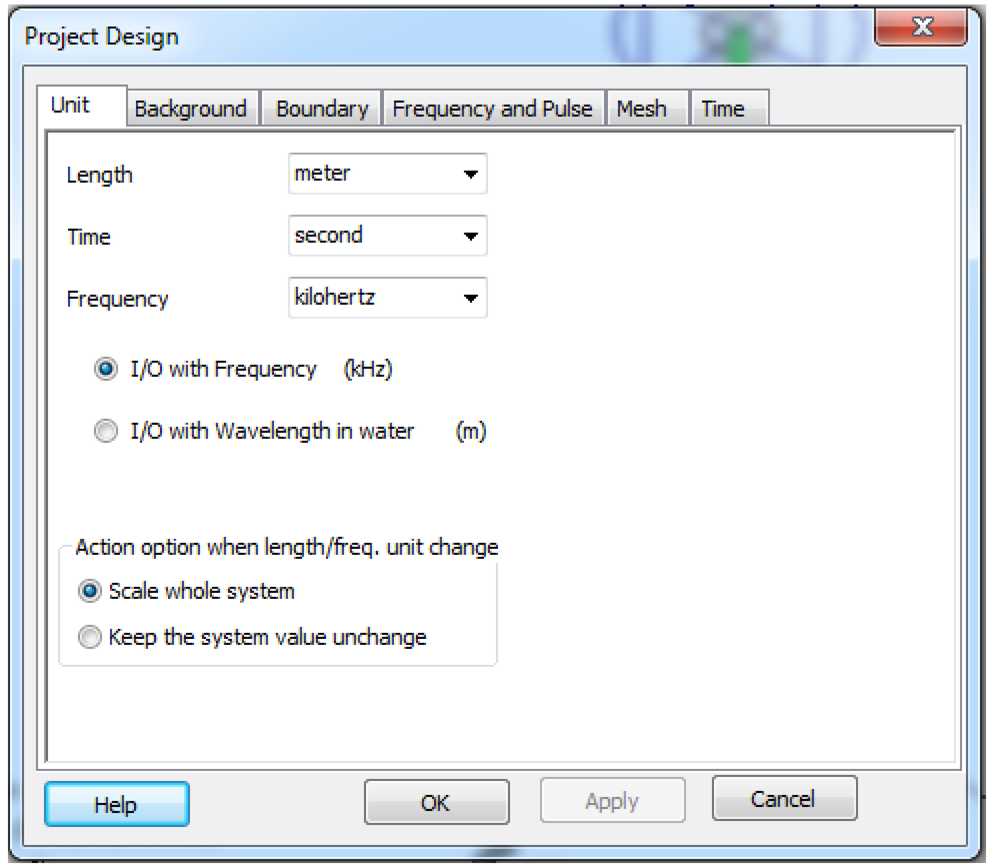
Then save the project with a name under the target folder. Here, we define the project name as “insulator” under folder “insulator”.





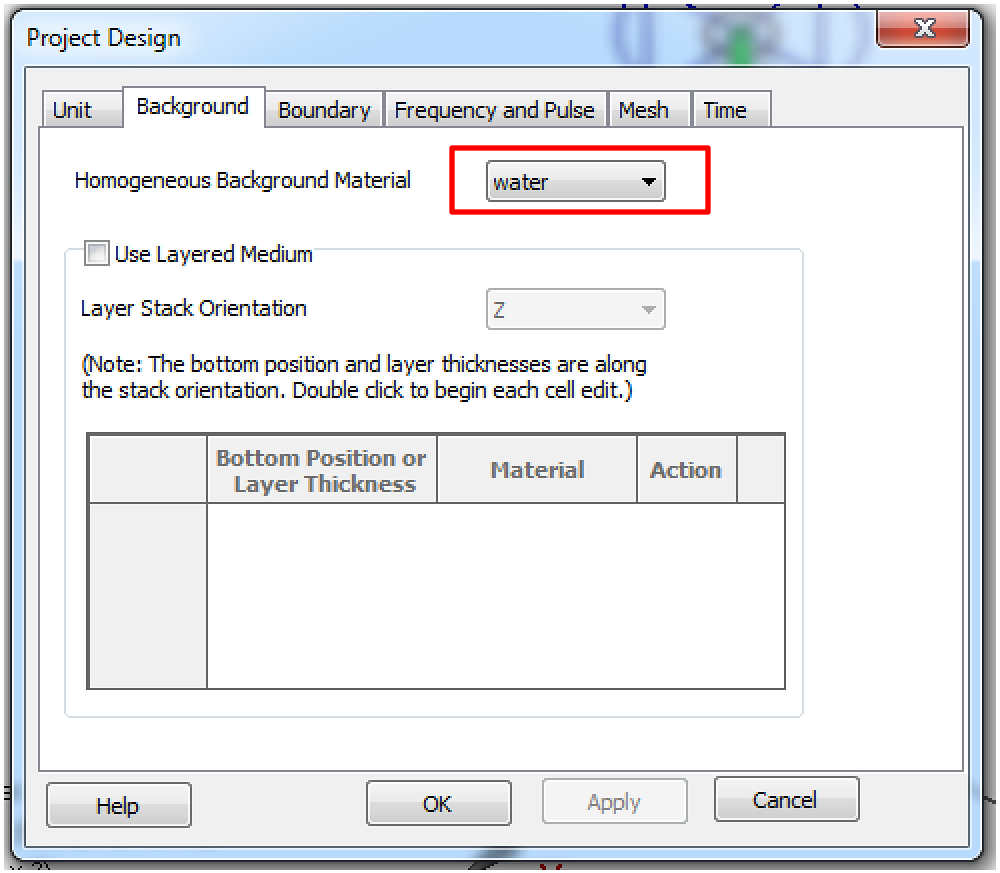
Use “**Project Design**” button or “**Unit**” treenode to modify project unit



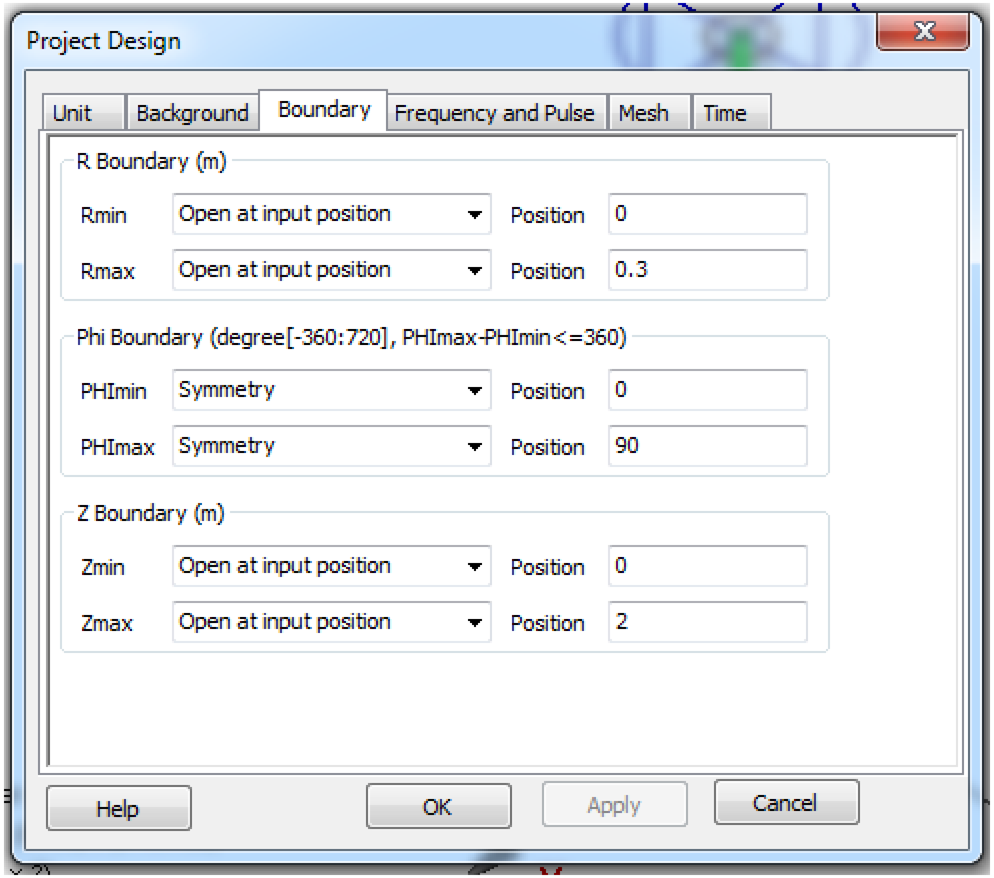


Define project unit as “m”, “s” and “KHz”. Other settings use ***Default*** values

Define the background as “water”

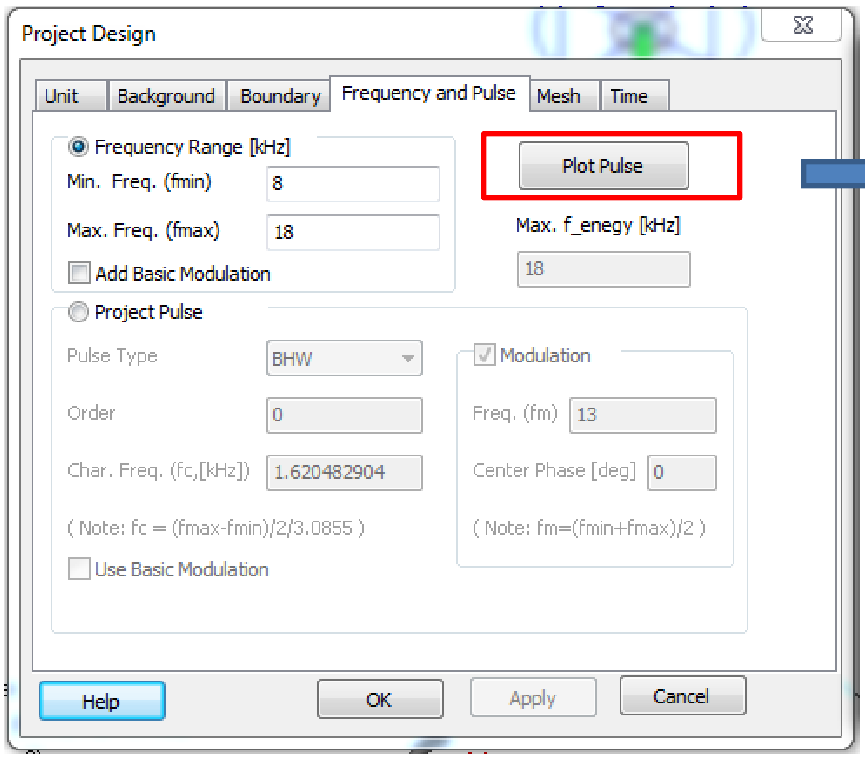


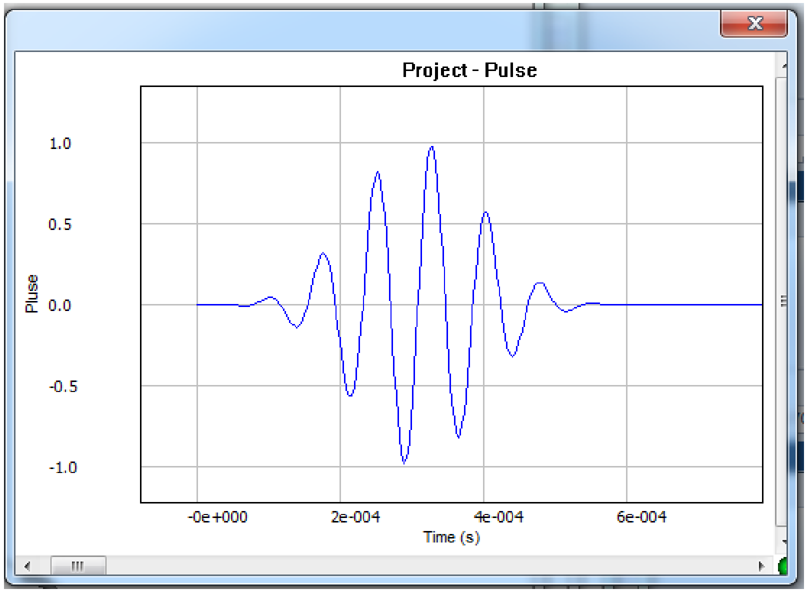
The boundary position & the boundary conditions are defined as following.



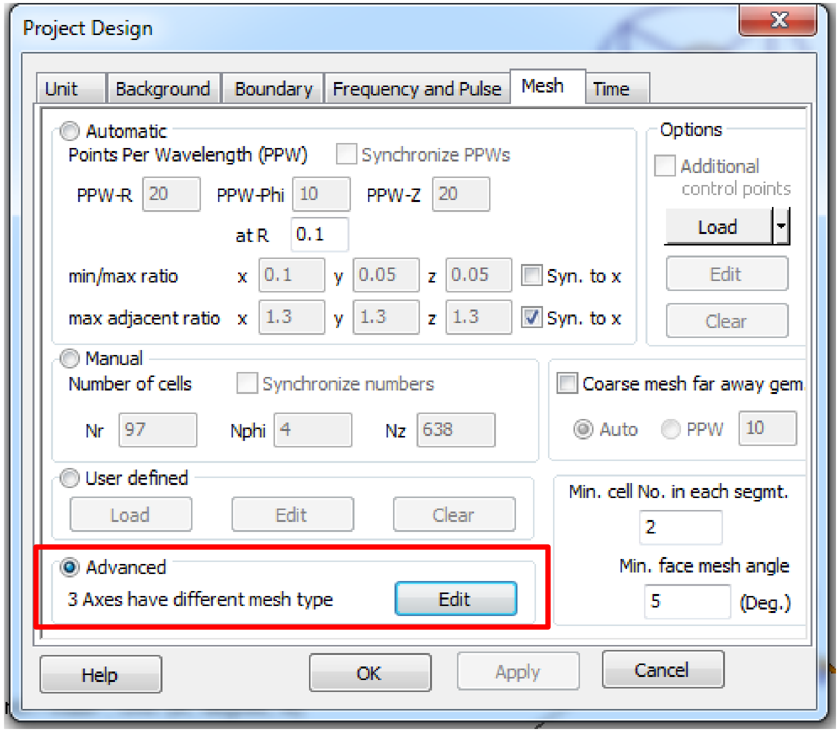
Because we will use a ring dipole to simulate a cylindrical pipe, based on previous explanation, we can use a quarter circle [0, 90] to represent the full circle.

Define project pulse as 1st order BHW pulse with f_{max}=18 KHz, f_{min}=8 KHz

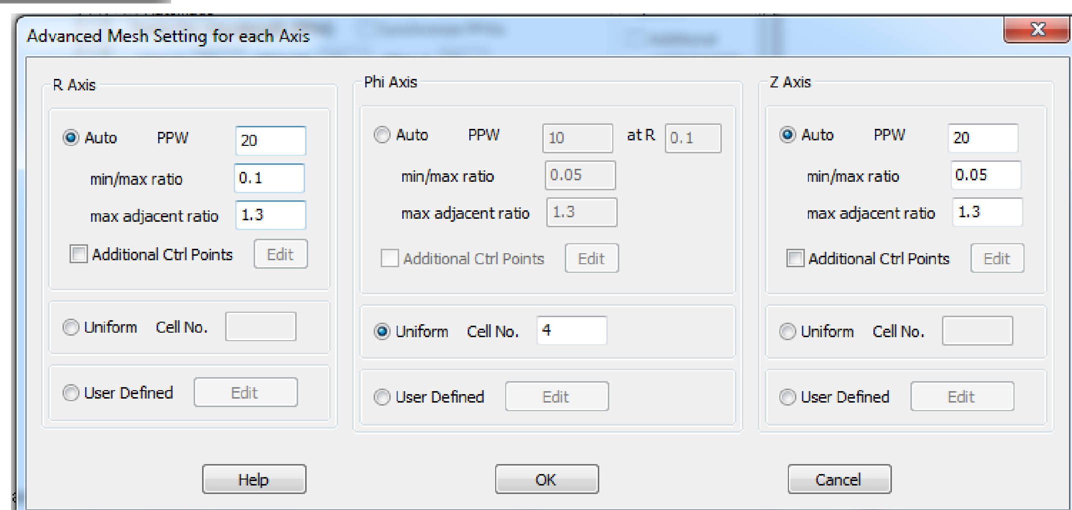




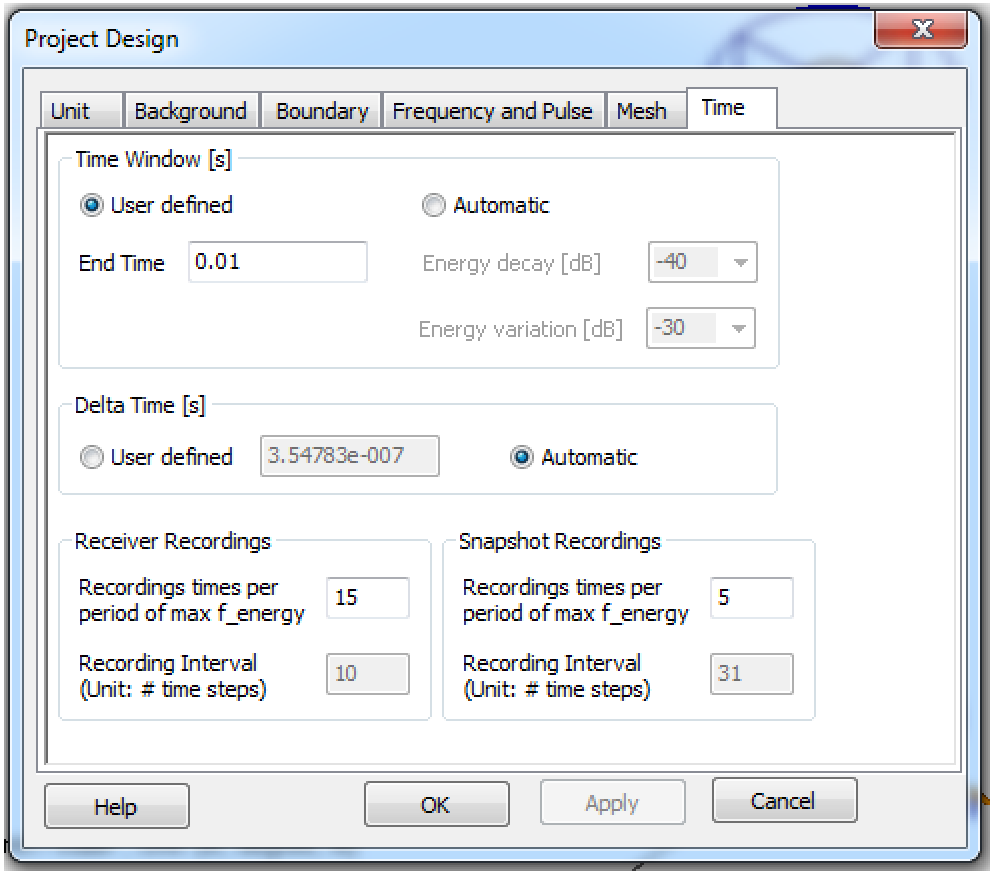
Advanced mesh



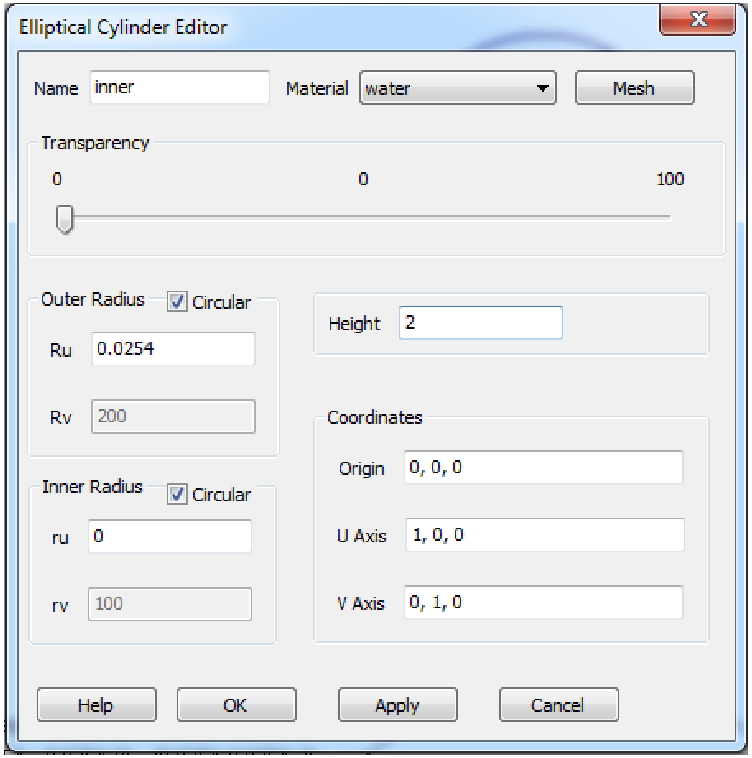
1) Automatic mesh with PPW=20 in R & Z axis
2) nϕ=4 with uniform mesh

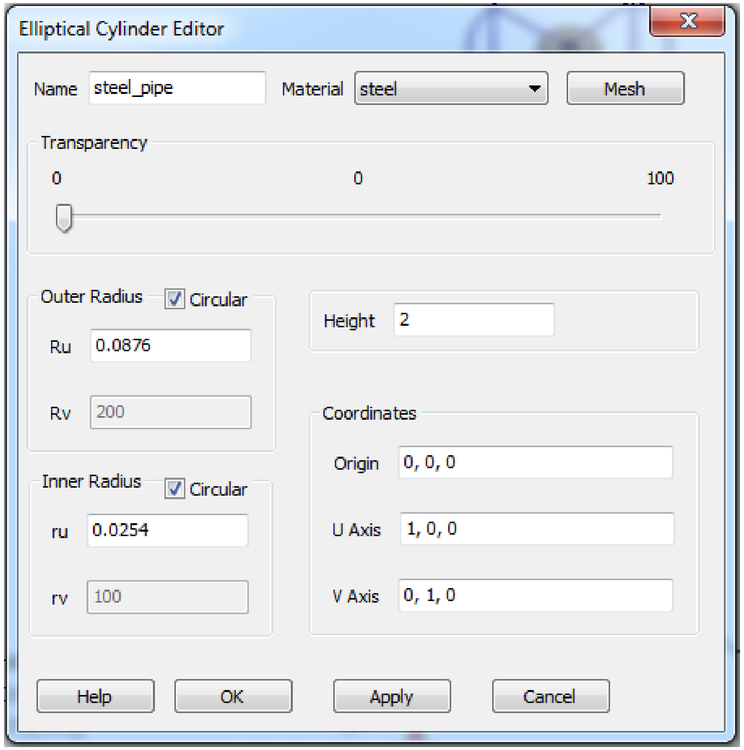


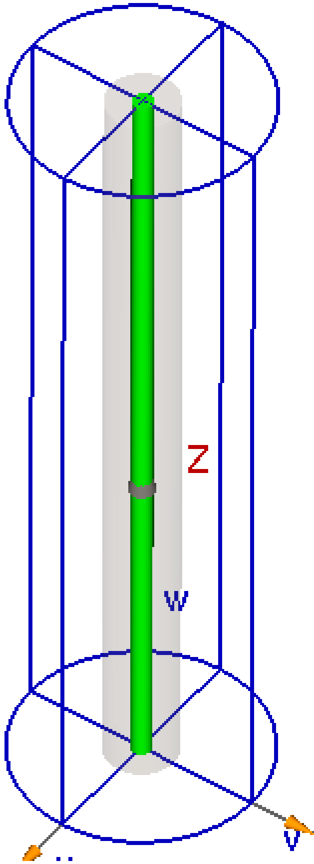
Automatic Δt to run 0.006 s



Define an inner water hole & a steel pipe

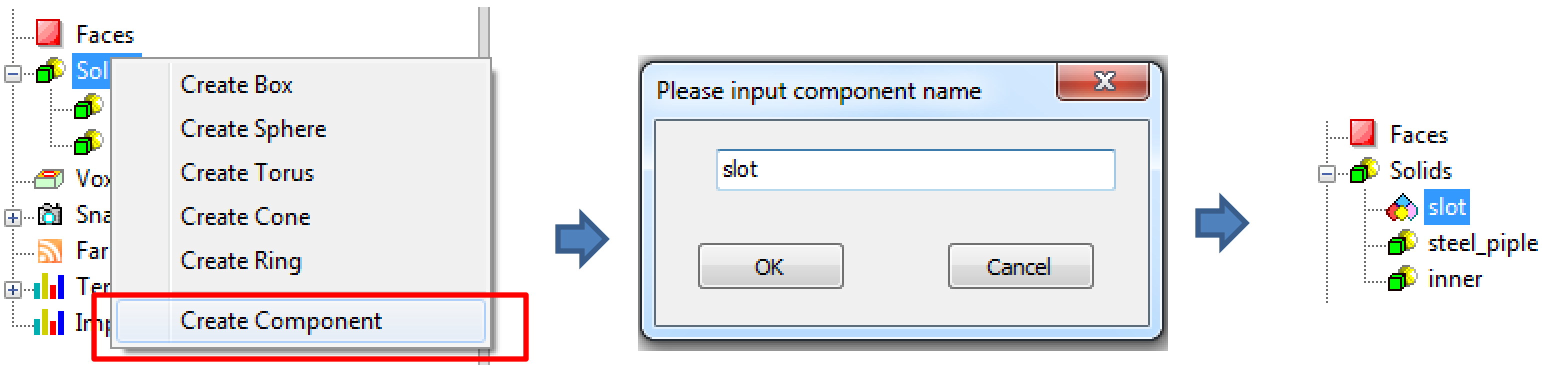






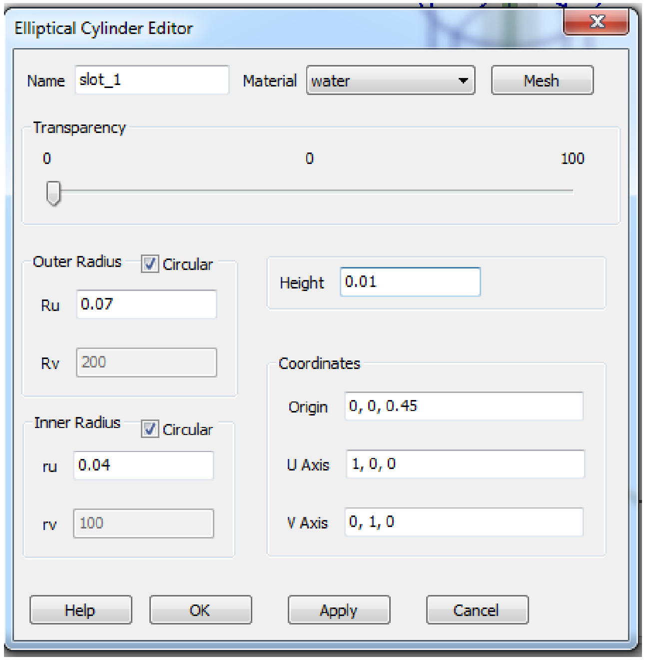
Then, we will define several water slots.

1) Define a “Slot” component to place slots

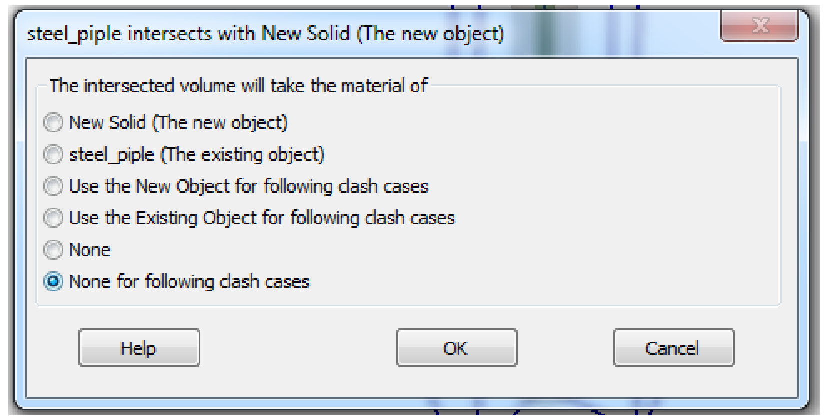


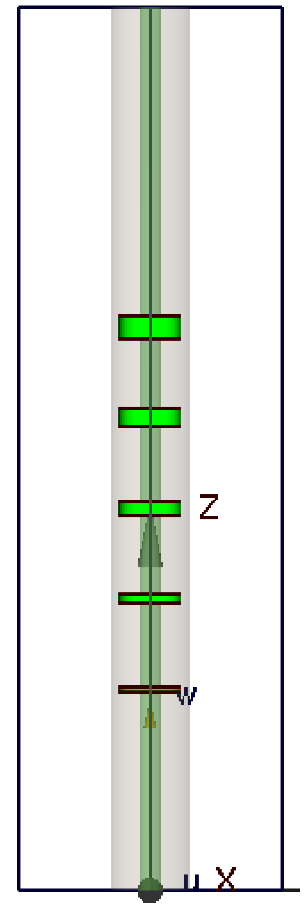
2) Define 5 slots in this component

Slot_1 Parameters



Slot_1 will clash with steel pipe, here, we do nothing, just keep the clash status.





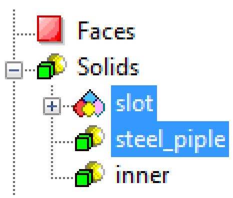
Slot_2, 3, 4 & 5 have the same radius, but the positions and heights are:

Slot_2: z=0.65, h=0.02
Slot_3: z=0.85, h=0.03
Slot_4: z=1.05, h=0.04
Slot_5: z=1.25, h=0.05

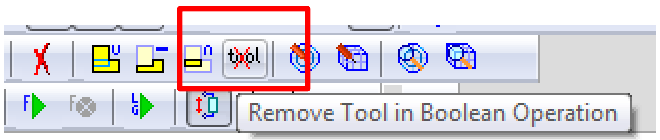
3) Subtract component from steel pipe

Select “steel_pipe” as the 1st object (this is the **Blank** object in the boolean operation)

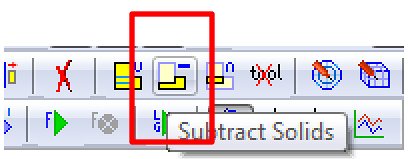
Press “Ctrl” key and select component “slot” as the 2nd object (this is the **Tool** object in in the Boolean operation)

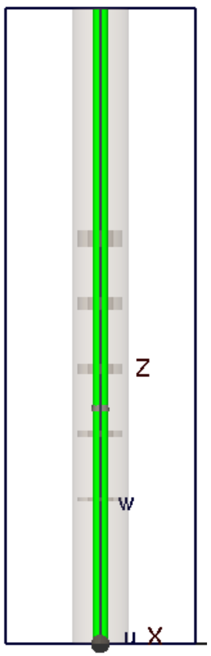


Make sure the button “Remove Tools in Boolean Operation” is not pressed

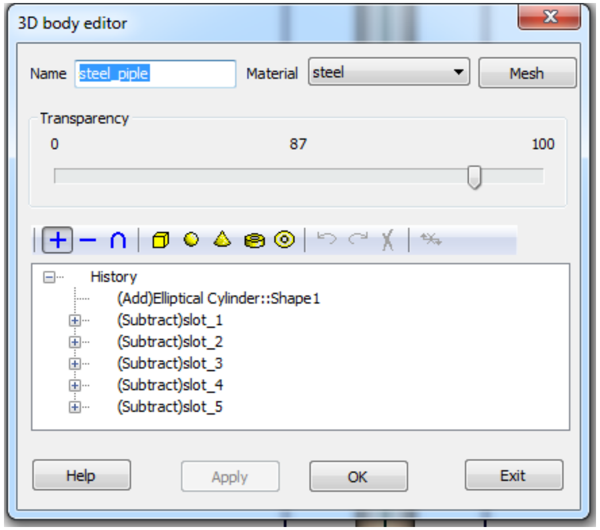


Then press “Subtract Solids” button

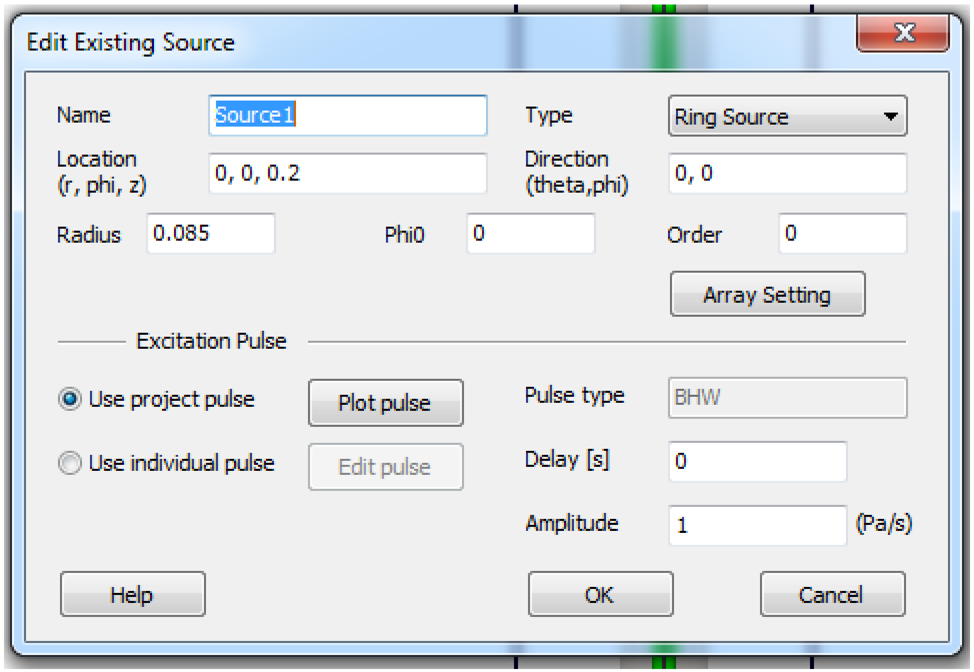




5 slots are subtracted from steel pipe



Define a ring dipole source


Edit Existing Source

Name	Source1	Type	Ring Source
Location (r, phi, z)	0, 0, 0.2	Direction (theta,phi)	0, 0

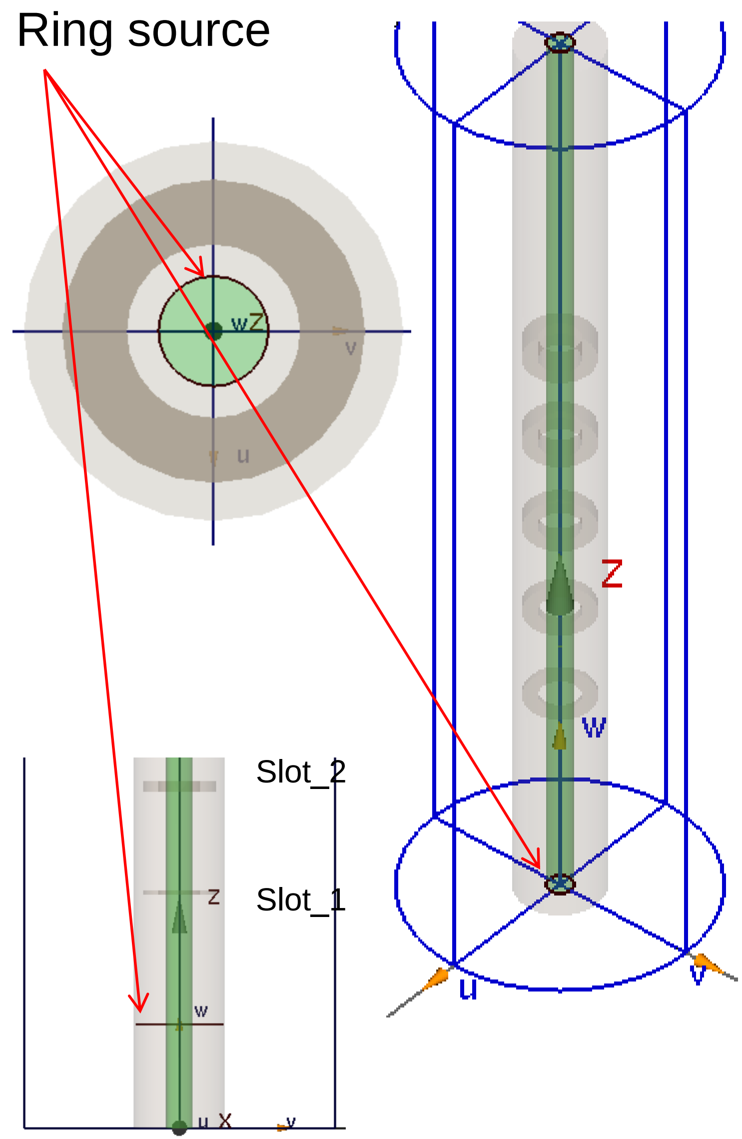
Radius 0.085 Phi0 0 Order 0

Array Setting

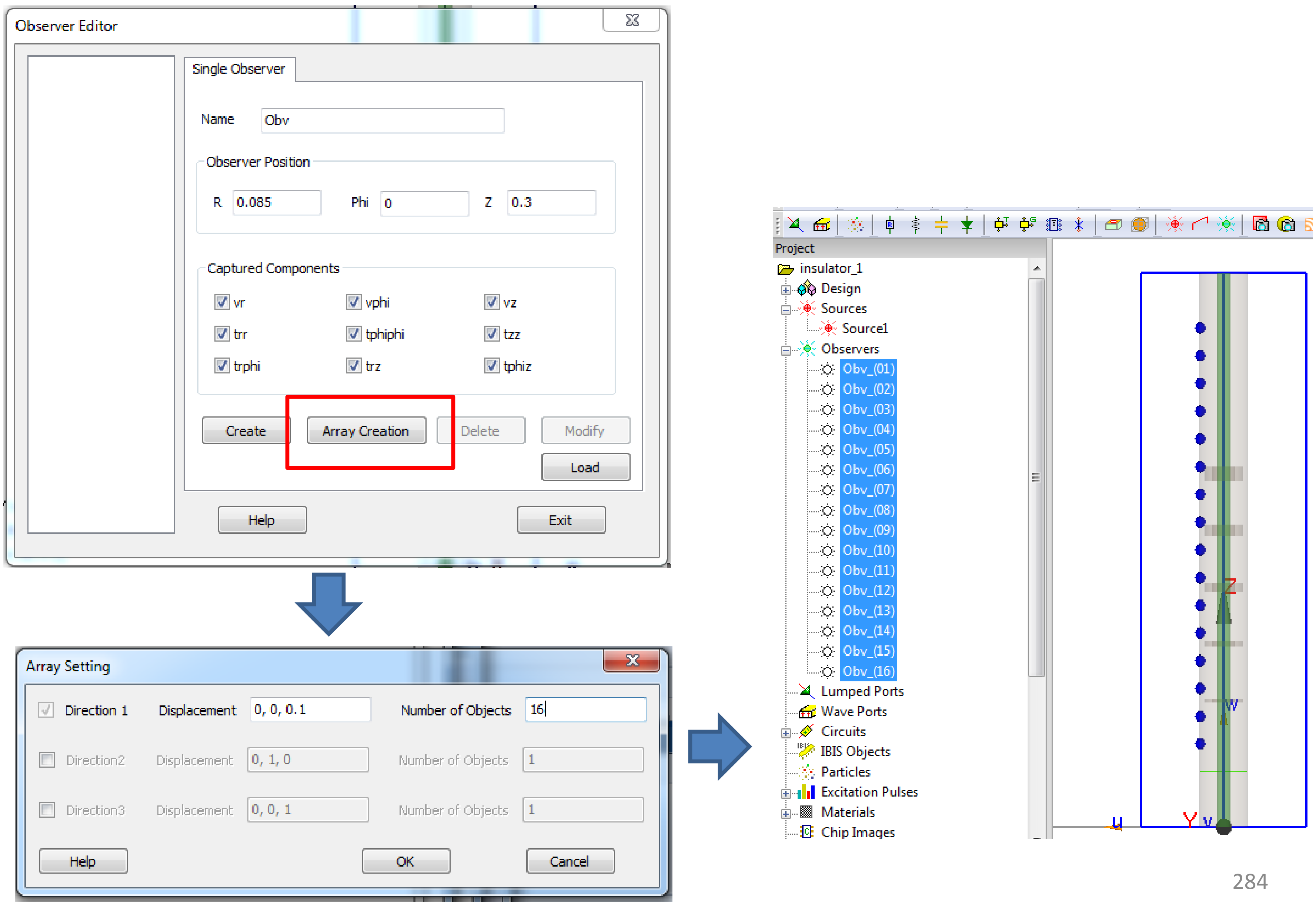
Excitation Pulse

Use project pulse	Plot pulse	Pulse type	BHW
Use individual pulse	Edit pulse	Delay [s]	0
		Amplitude	1 (Pa/s)

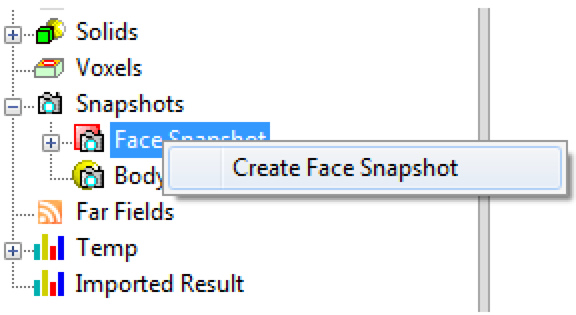
Help OK Cancel

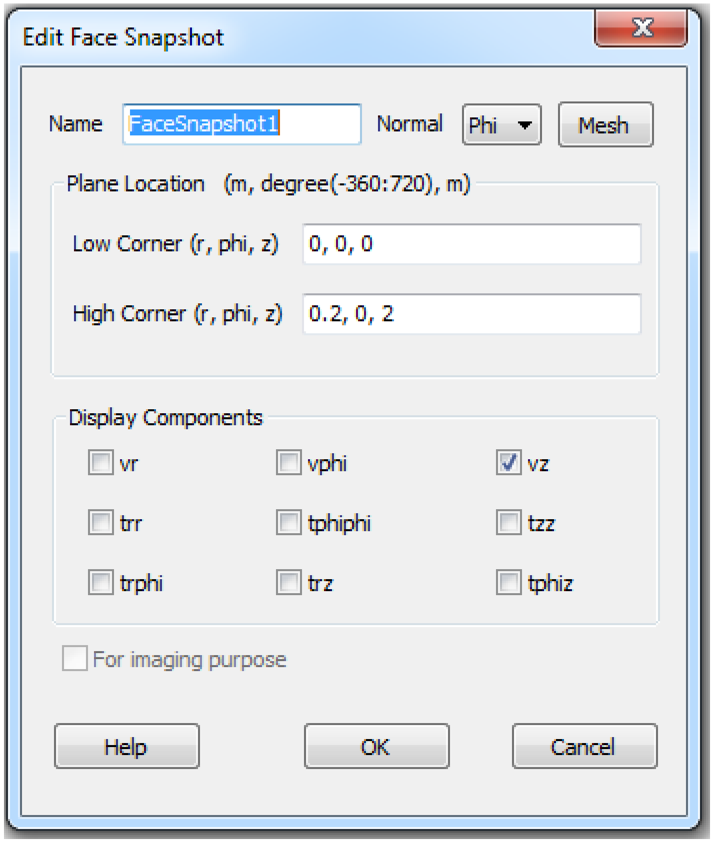


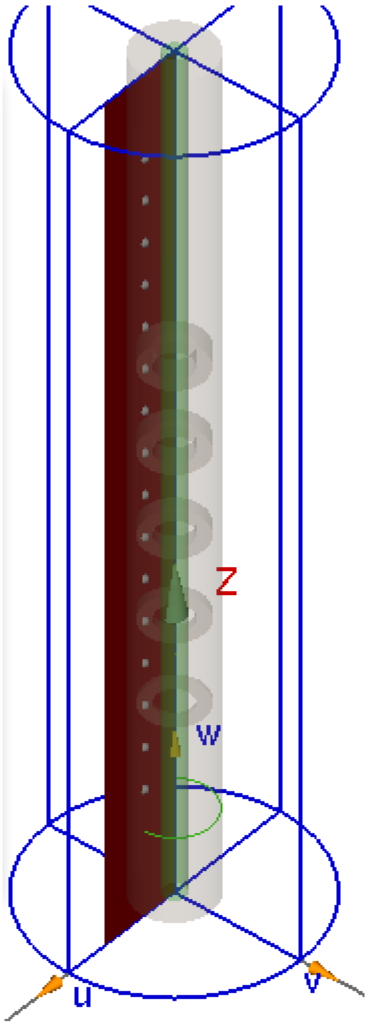
Define an array of observers to record the field



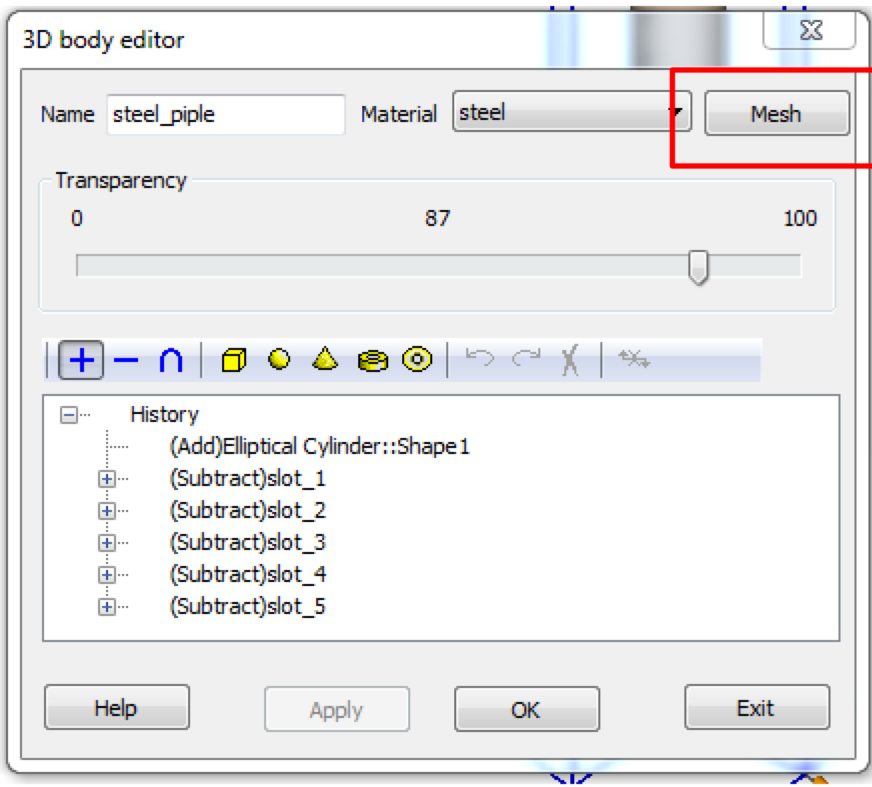
Define an snapshot to observe the field propagation

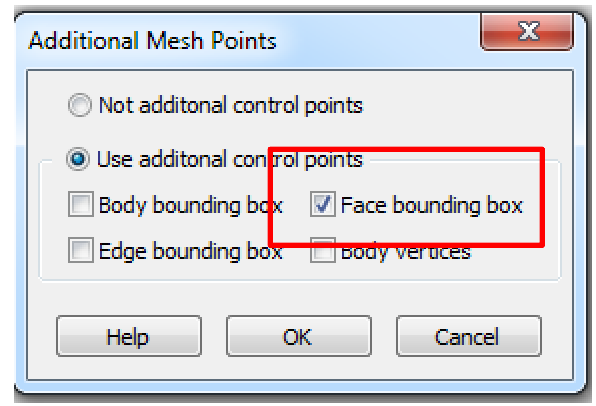






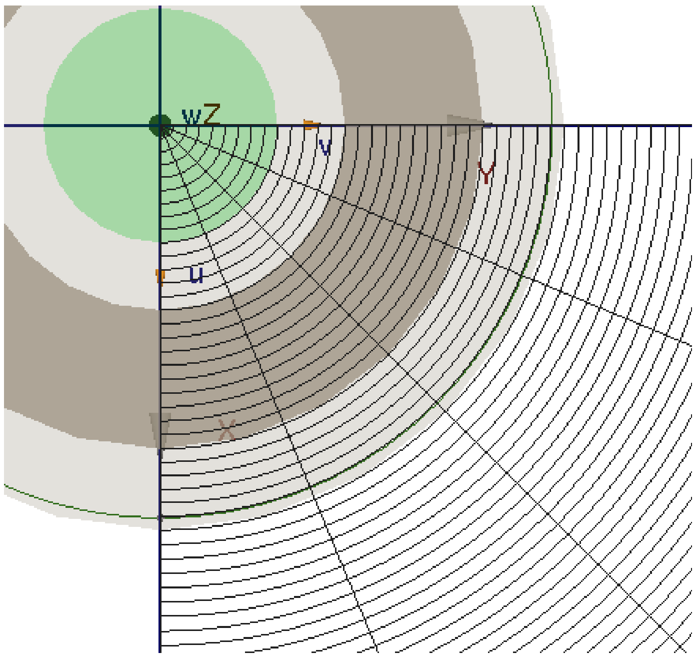
In order to let the mesher can capture the slot in the steel pipe, we need to set the steel_pipe's mesh option as "Face bounding box"

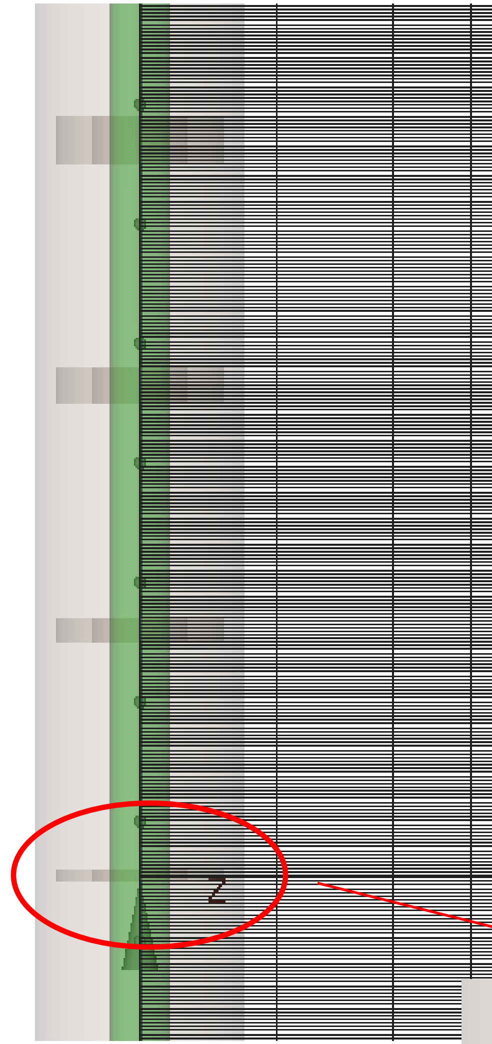




Then we check the system grid, we can see it captures the slot correctly

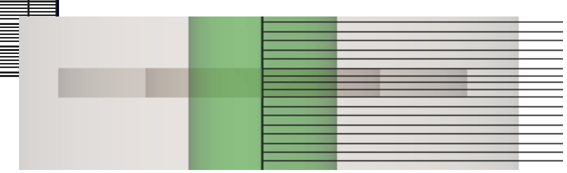
R grid



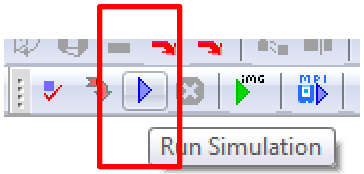


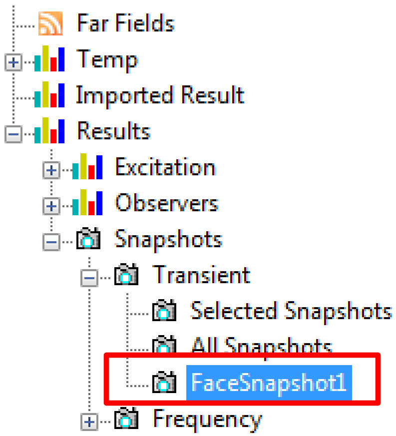
X grid

Zoom in



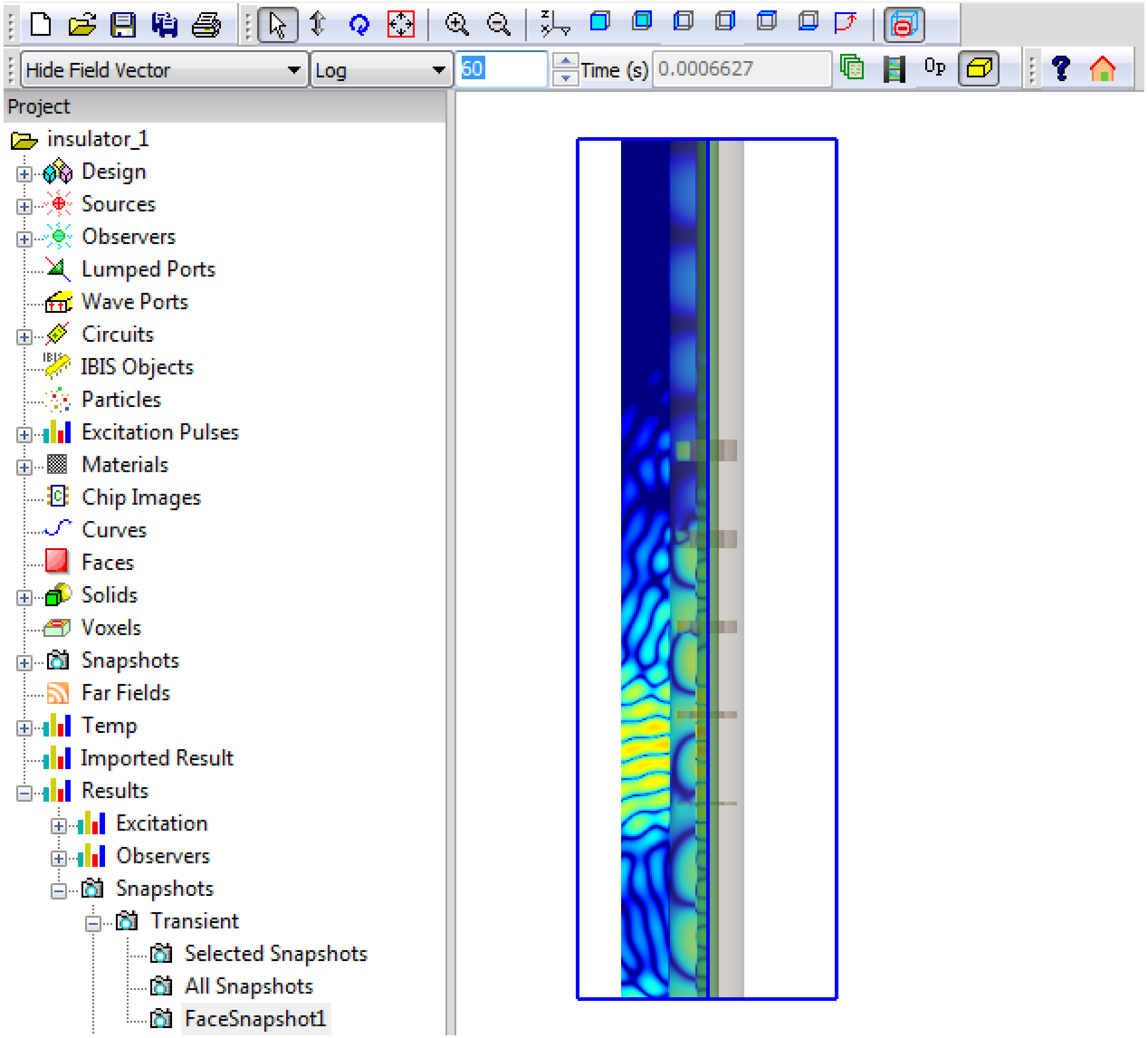
Then, we can start the simulation & check the result.



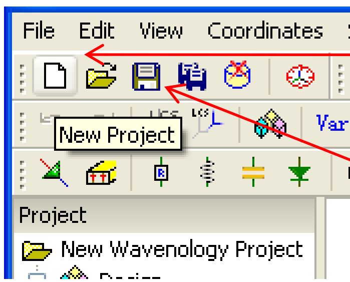


Double click the name of snapshot to show it in the canvas

Hide field vector & using log scale, switch frame manually, following is the 60^{th} frame.

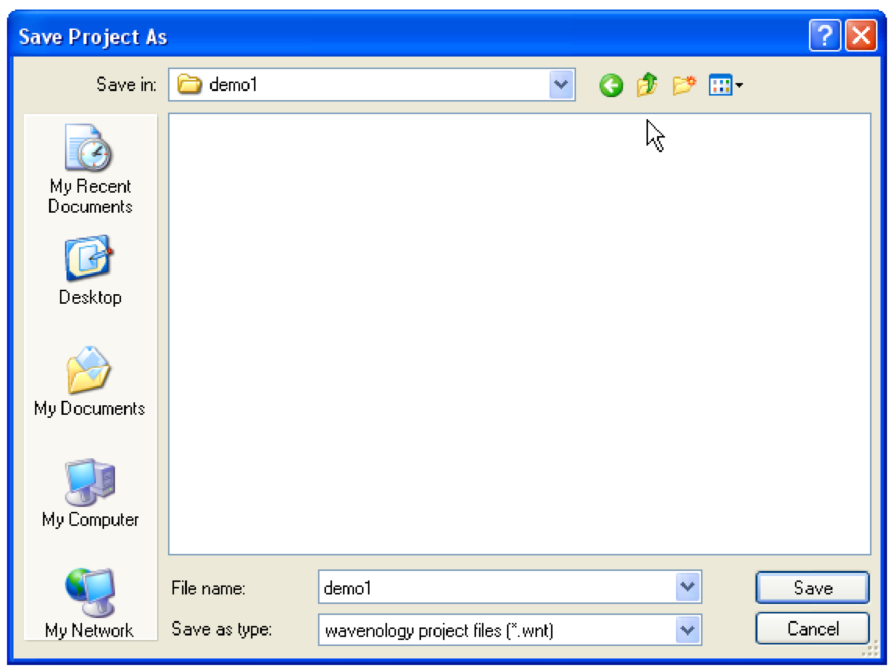


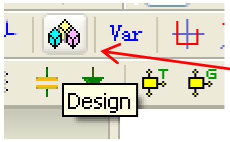
Case3
Import 3D Solids from a SAT file



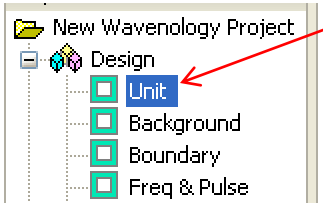
Use “New Project” button create a new project

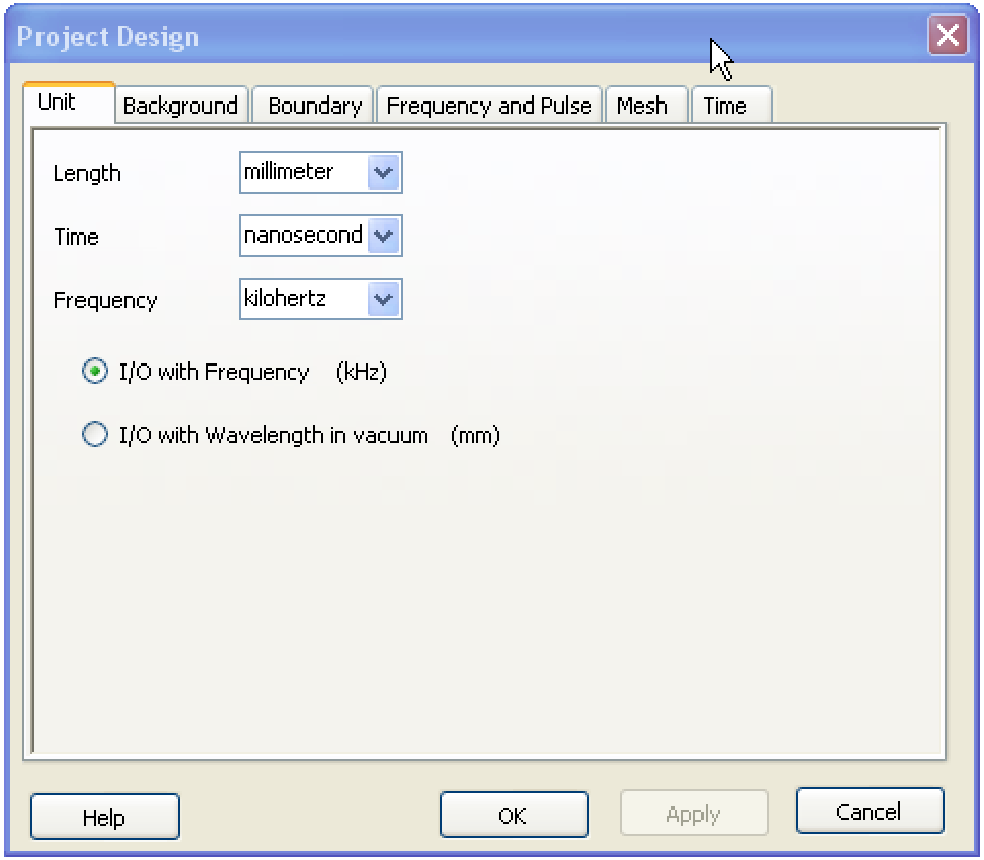
It is better, then, to save the project with a name under the target folder. Here, we define the project name as “demo1” under folder “demo1”. Because in order to generate the mesh data file, the software need a predefined folder to export them. Without this predefined folder, the data file will be saved in some system default folder and will be hard to find.





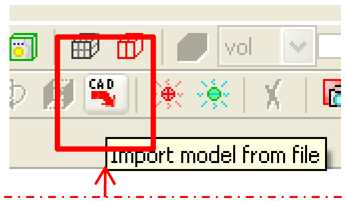
Use “**Project Design**” button or “**Unit**” treenode to modify project unit



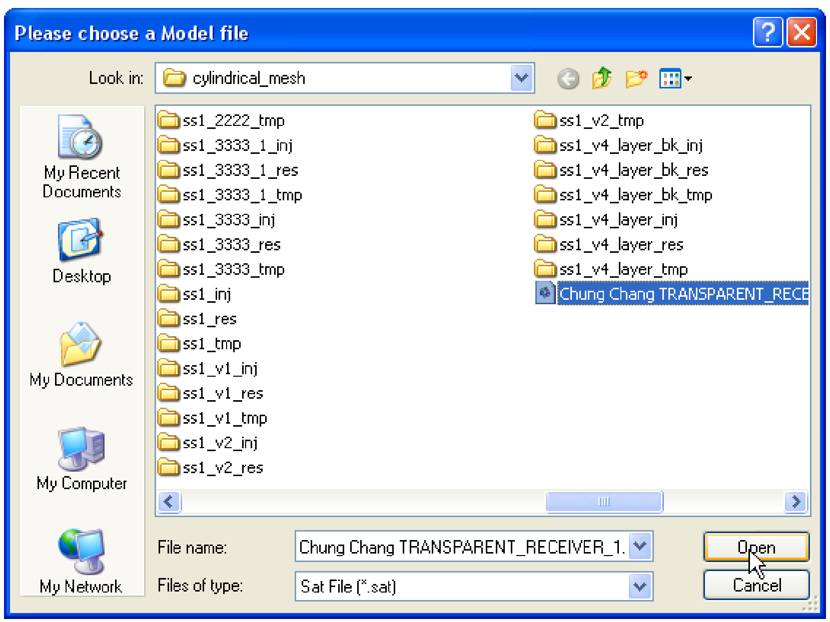


Define project unit as “mm”, “ns” and “KHz” . Other settings use ***Default*** values

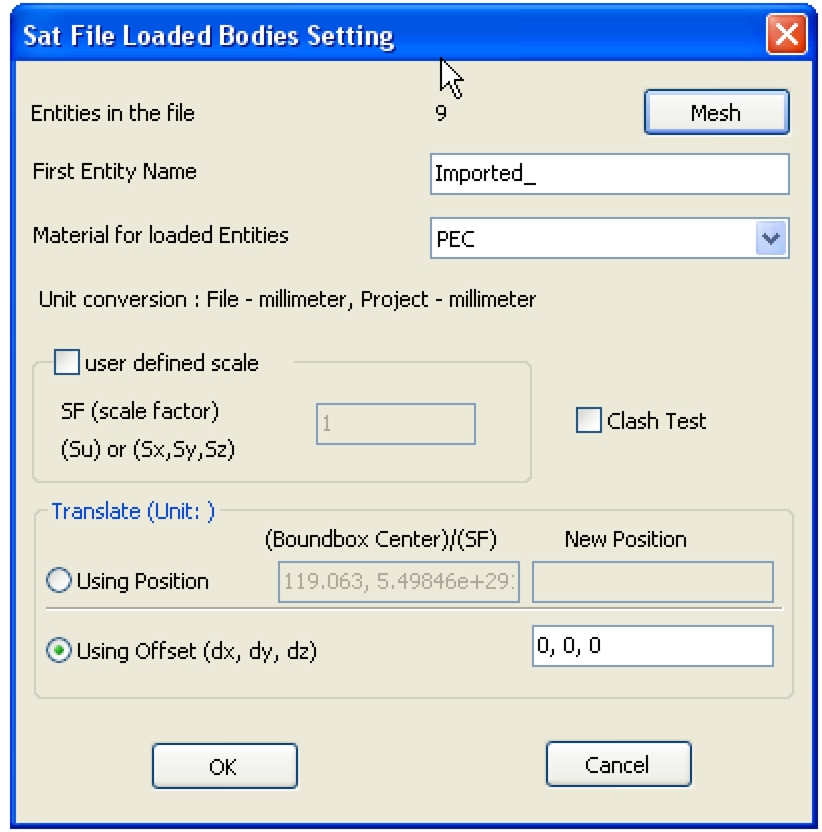
Insert 3D solids from an SAT file. For example, “TRANSPARENT_RECEIVER_1.SAT”



Use this Toolbar button



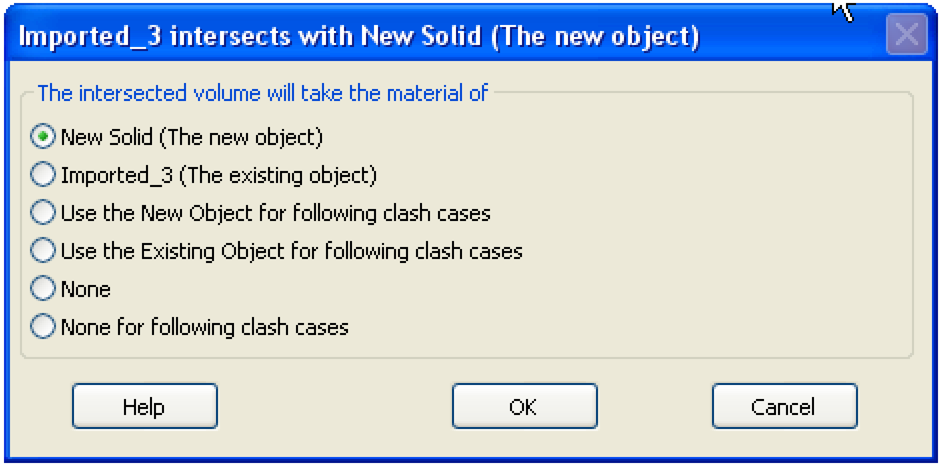
Select the SAT file



Define the name, default material and position for all imported solids

Due to the following reasons:

1. Different 3D modeling software has different tolerance. Not-intersect solids could be treated as intersect in porting among different software packages
2. The solids are actually intersect due to the original 3D modeling software that permits such an intersection.

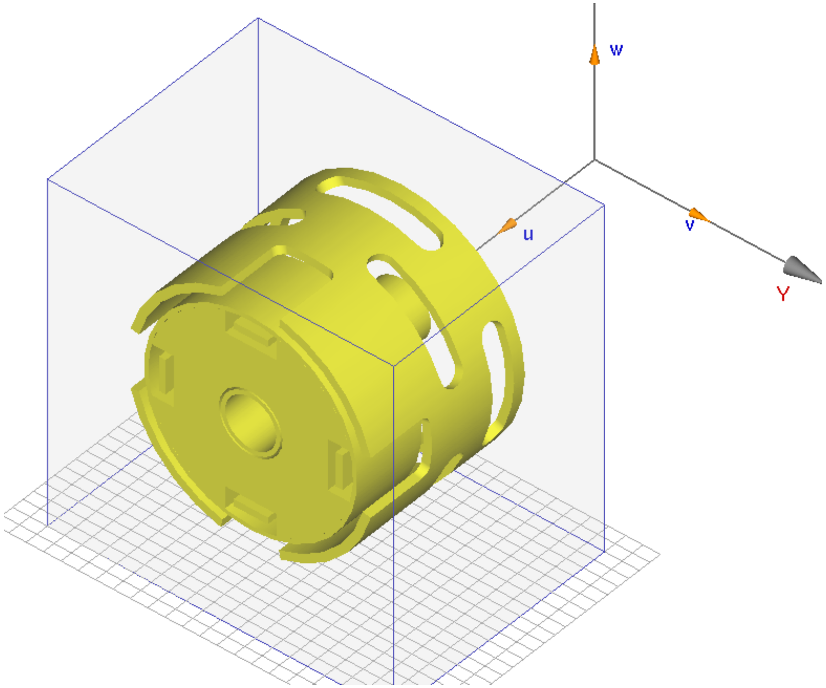


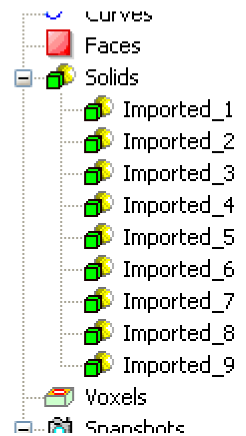
In importing this file, Wavenology finds there is an intersection among some solids and reports a warning.

User needs to decide how to handle intersections. Here, we let the solid ***Imported_3*** occupy the intersected region.

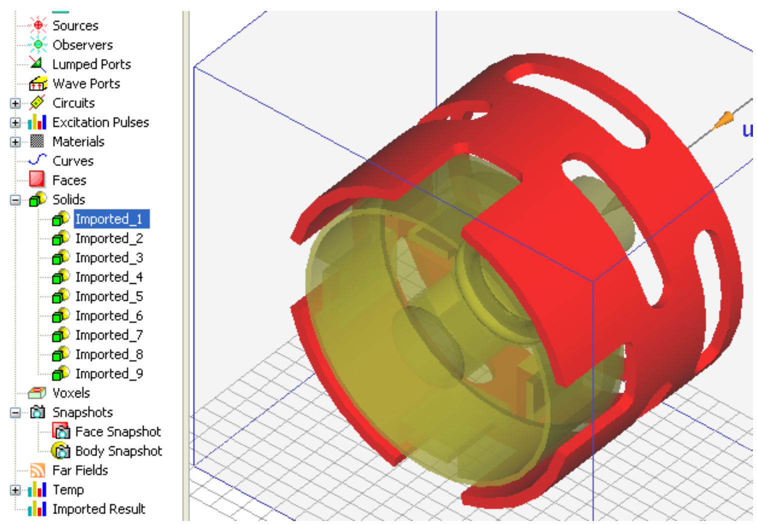
Note: WCT provide a special treatment for the cylindrical mesh generation and elastic wave simulation. If user consider the imported models should not be reported as clashing and these clashes will not effect the final mesh and simulation. User can choose "***None for following clash cases***" to let WCT skip clash test.

Following is the shape of the imported solids and their names

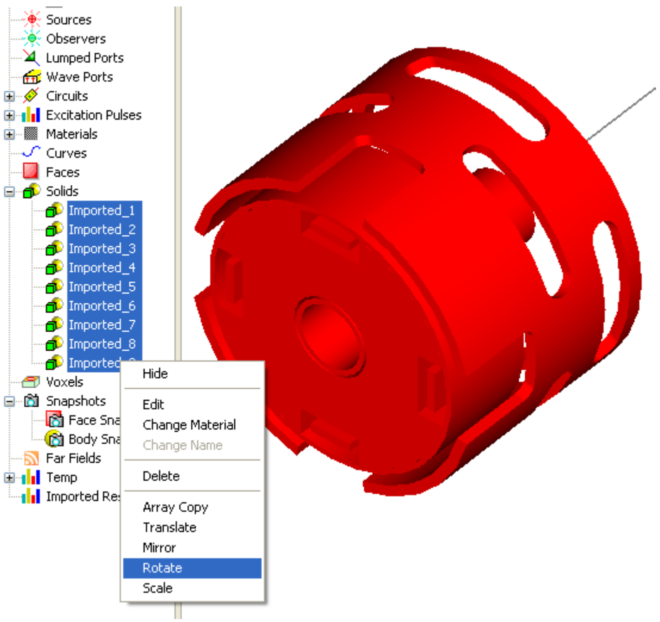




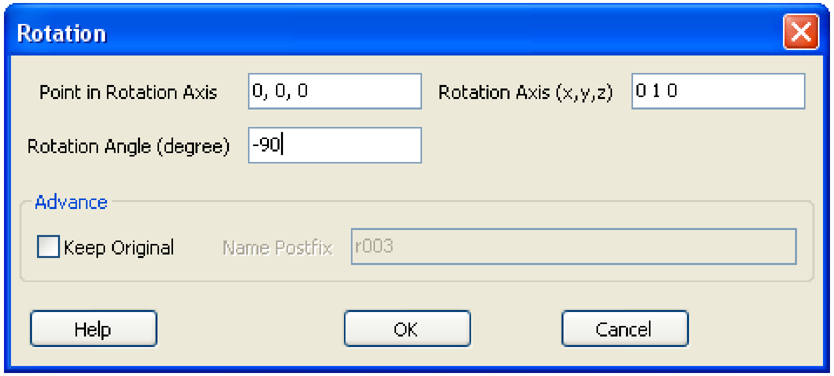
User can check the name-shape mapping by select a solid in the tree.



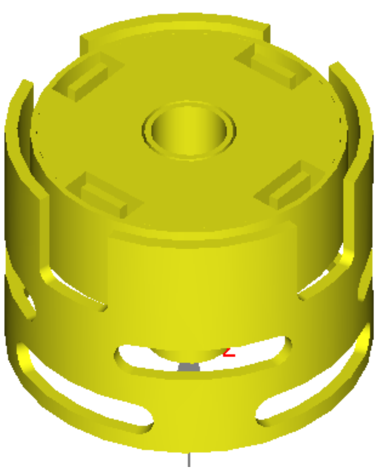
The current Cylindrical mesh generator only supports Z axis as the rotation axis, but the imported solids can be X aligned. Therefore, we need to align all solids to Z axis by **Rotation**.



Press “ctrl” key and use mouse left button to select all solids, then press mouse right button to popup ***Solid Transform Menu***. Press “**Rotate**” menu.

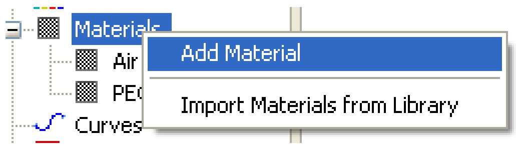


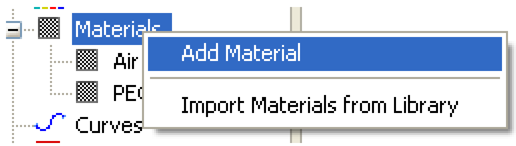
Let all solids rotate -90 degree along Y axis.
The input is shown in above dialog.
Note: CCW is positive degree, CW is negative degree.

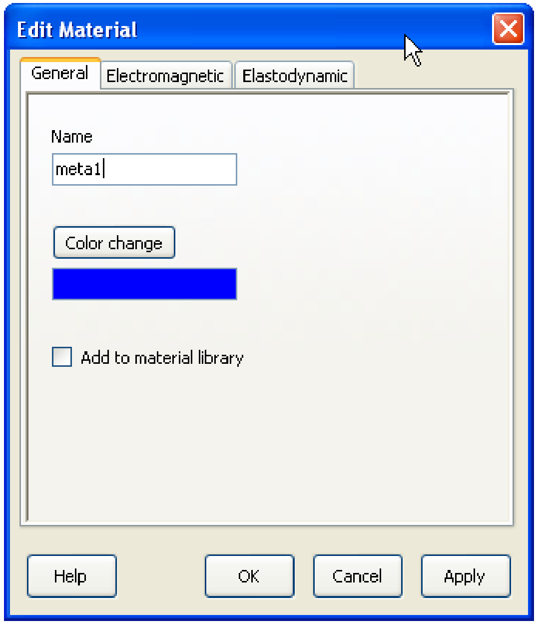


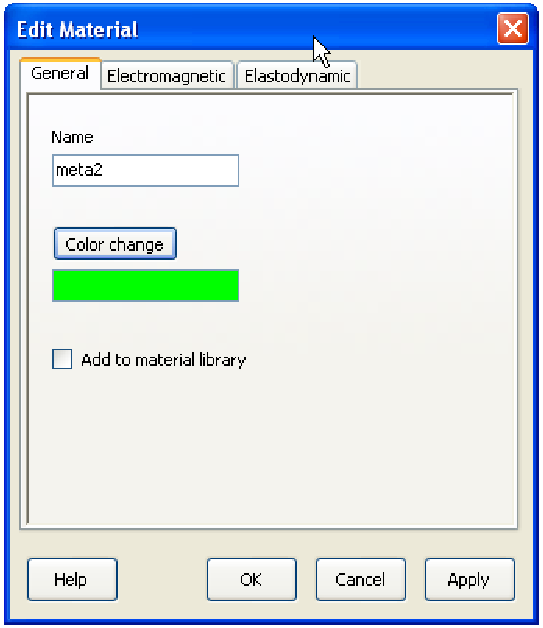
Define different materials for different components.

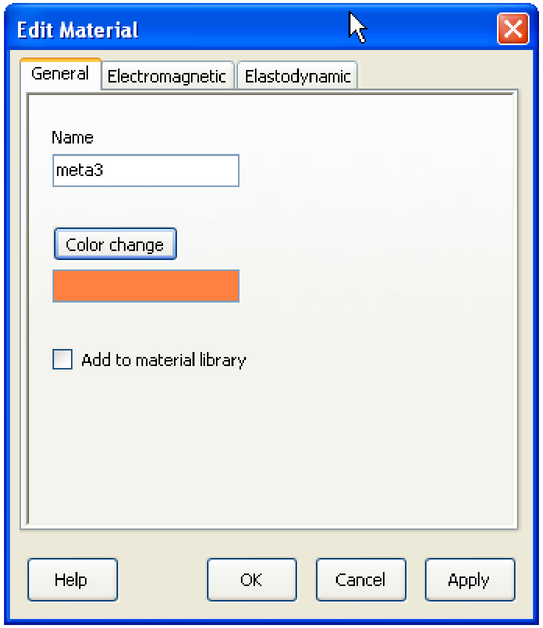
Here, we define metal1, metal2 and metal3.











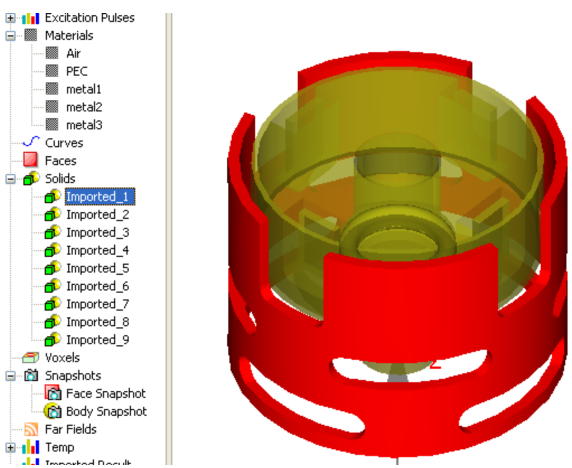
Assign different solids by different materials

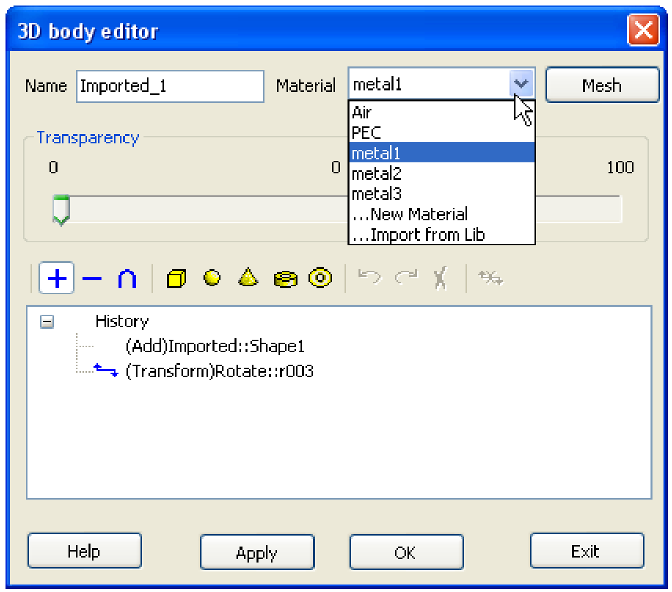
For example, assign component “Imported_1” by material metal1

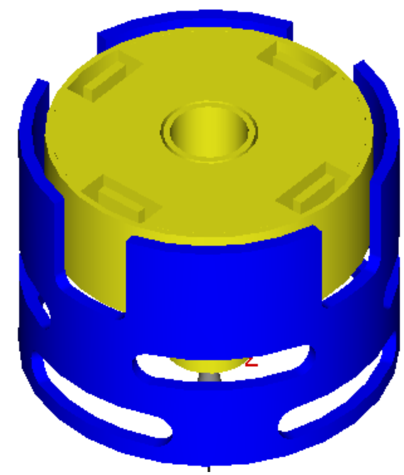
double click treenode “Imported_1”

Modify material to metal1, then press “OK”

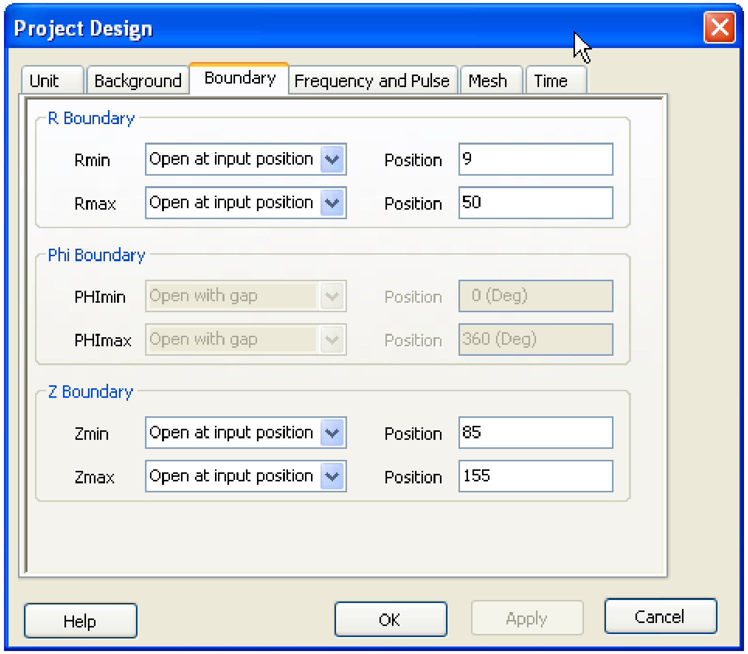
The result



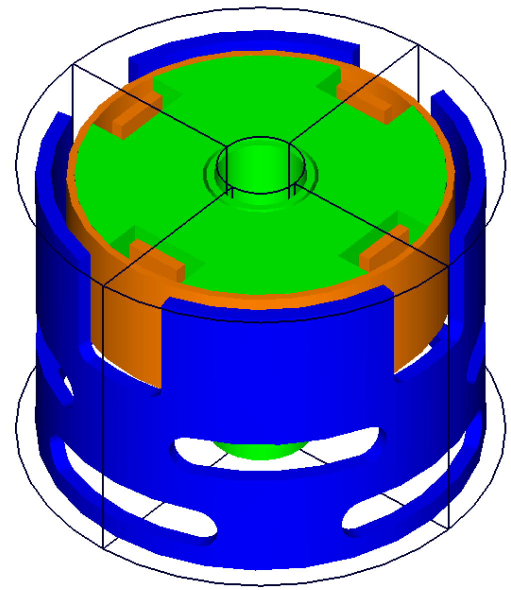


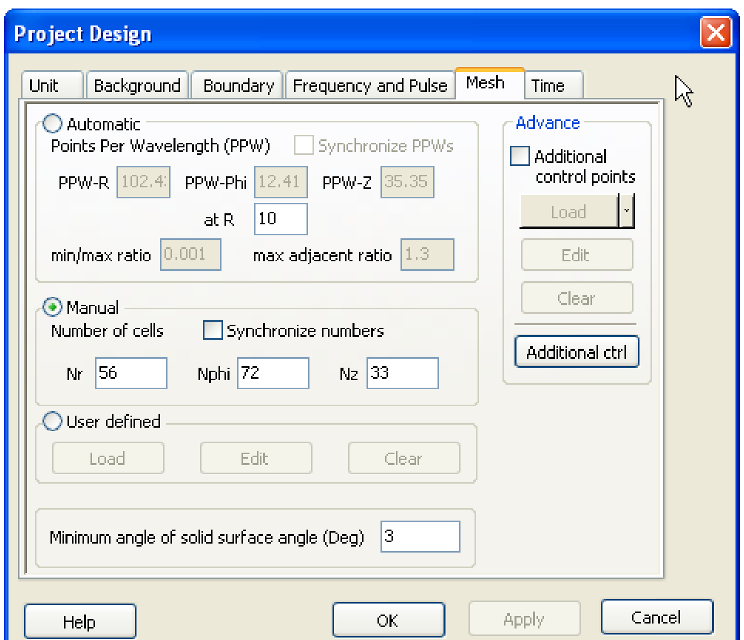


Set up cylindrical mesh control



Computation domain boundary and size



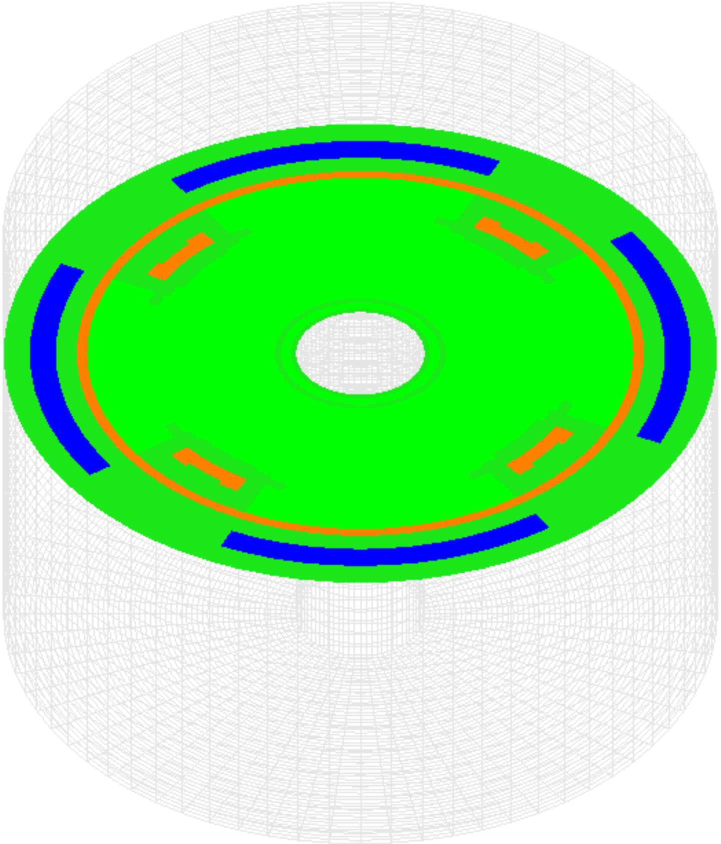


Unifrom mesh setup

After setting up cylindrical mesh control, User can export or display cylindrical mesh by

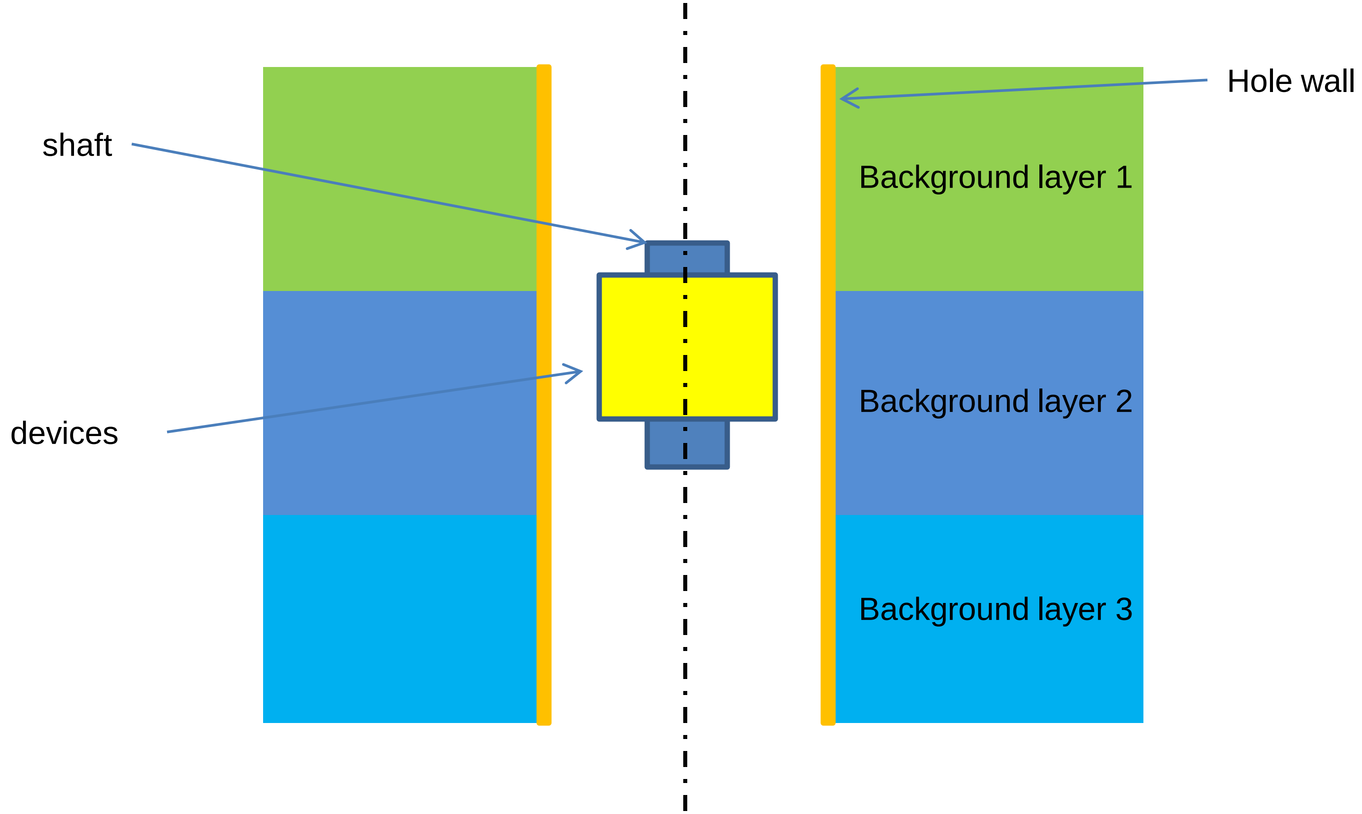
 Generate cylindrical mesh and export the data file

 Begin to show the cylindrical mesh

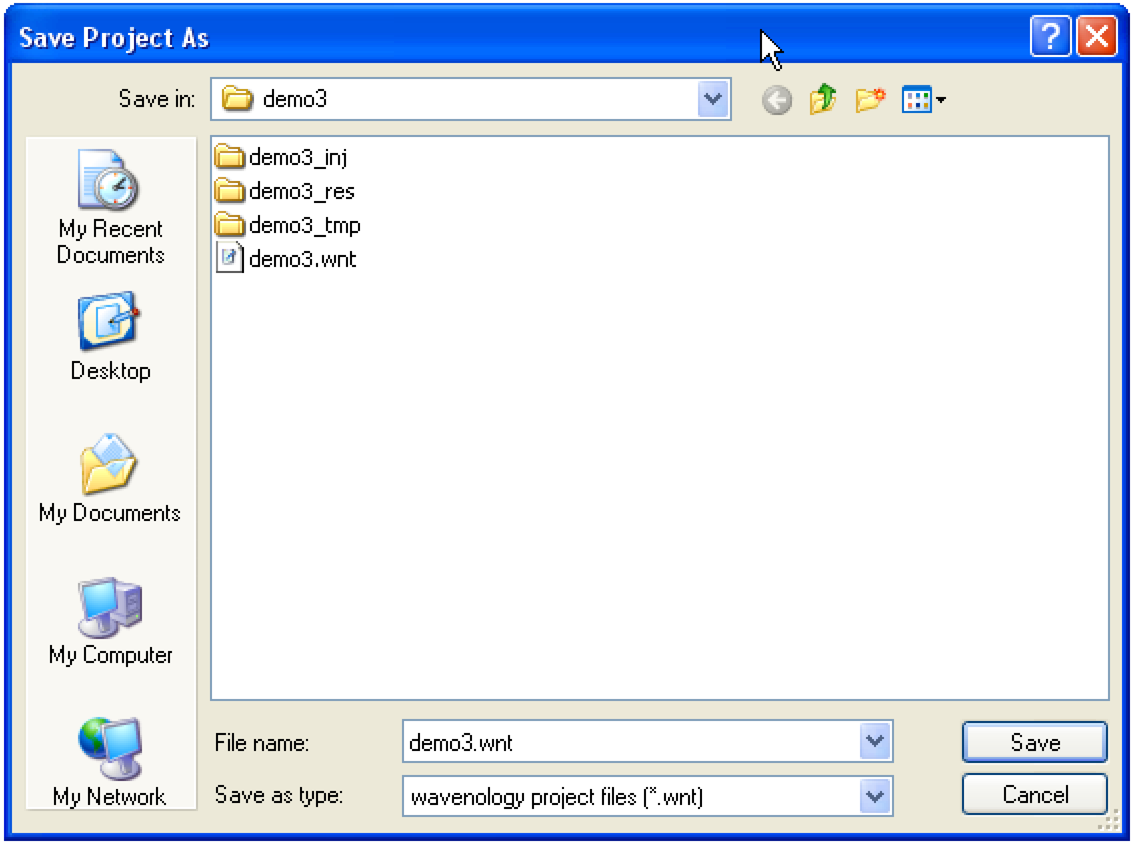


Case4
Generating Cylindrical Mesh and Export Mesh Data

This case will generate cylindrical mesh for the devices in layered media.
The solids in this case represent a general setup for borehole environment.

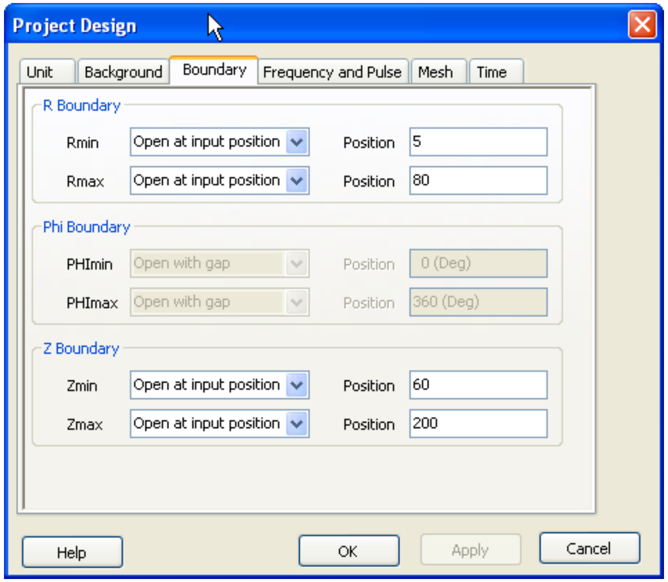


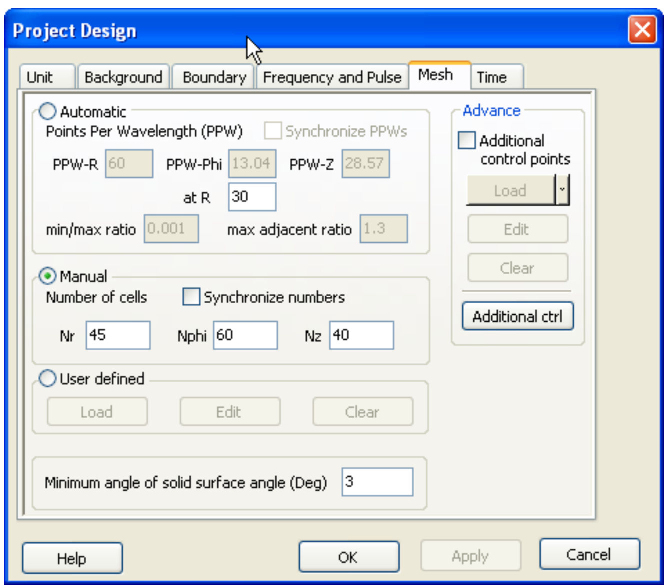
In this case, we use the imported solids in **demo1** as the shaft (**Imported_2**) and devices (others solids).
So, open **demo1** and ***Save As*** **demo3** under folder “**demo3**”.



In order to mesh bigger space including layered background media, we need to increase space size.

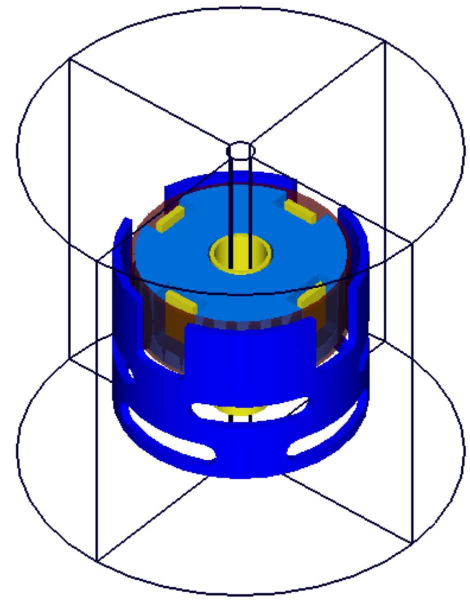
Set up cylindrical mesh control

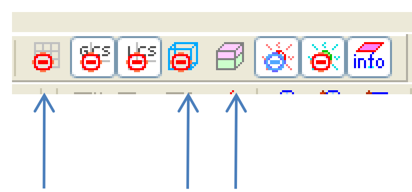




Here, we define mesh space:
rmin=5 mm, rmax = 80 mm;
zmin=60 mm, zmax=200 mm

The mesh bound box is shown as

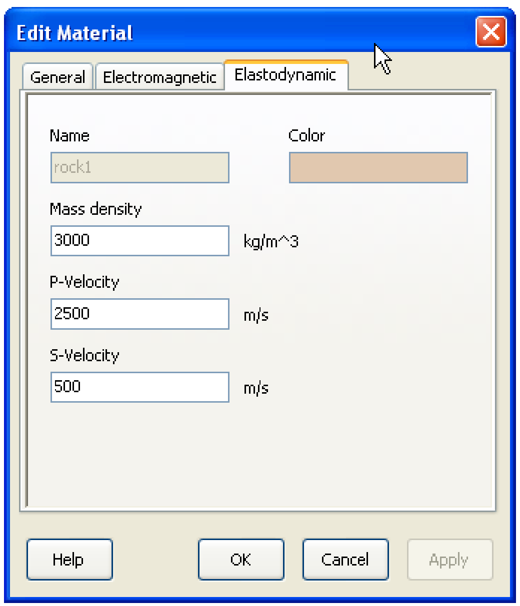


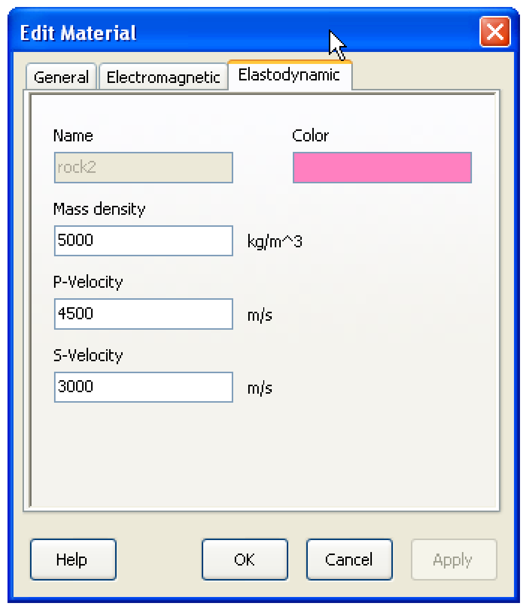


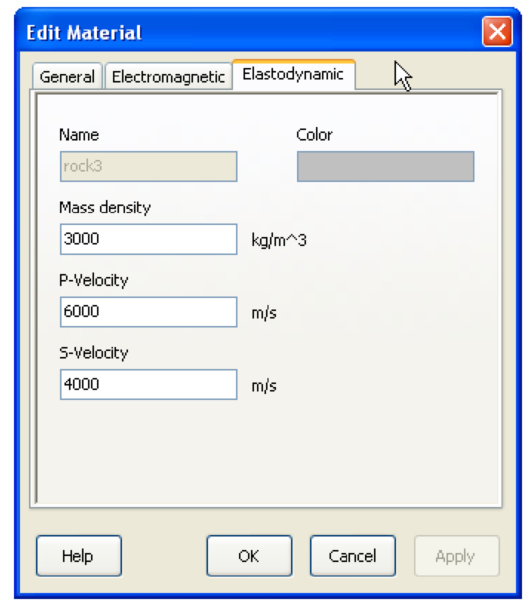
Note: in order to show this bound box only, user need to turn-off these displaying options

Define different materials for different layered background media.

Here, we define rock1, rock2 and rock3.

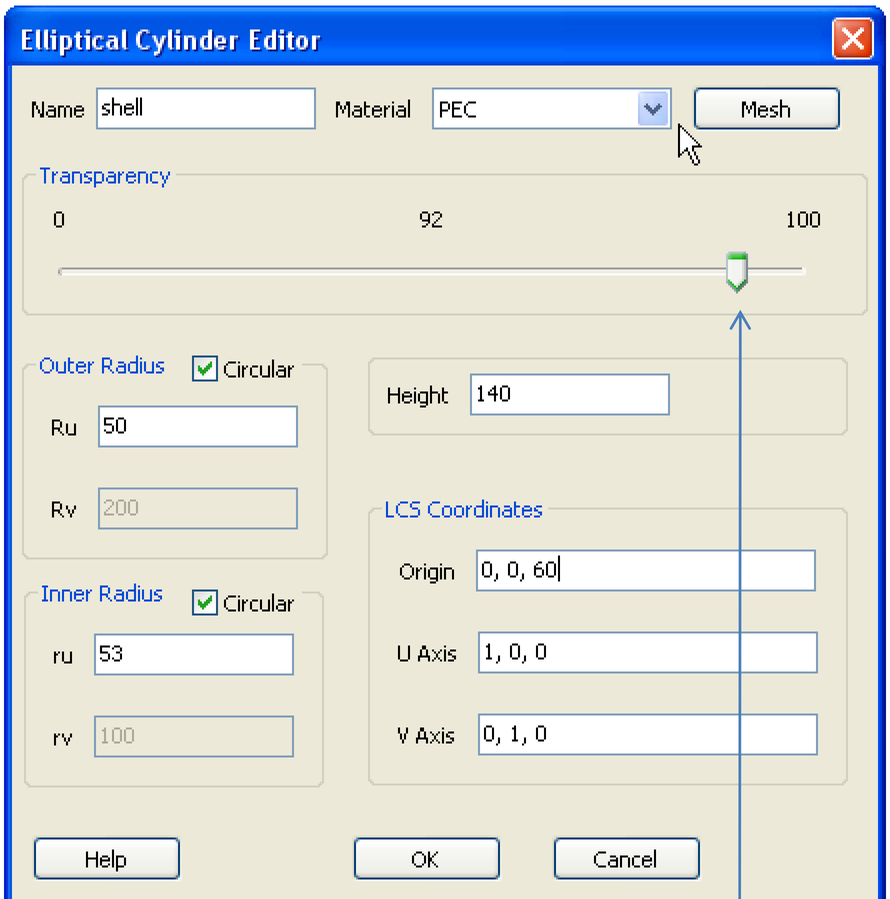




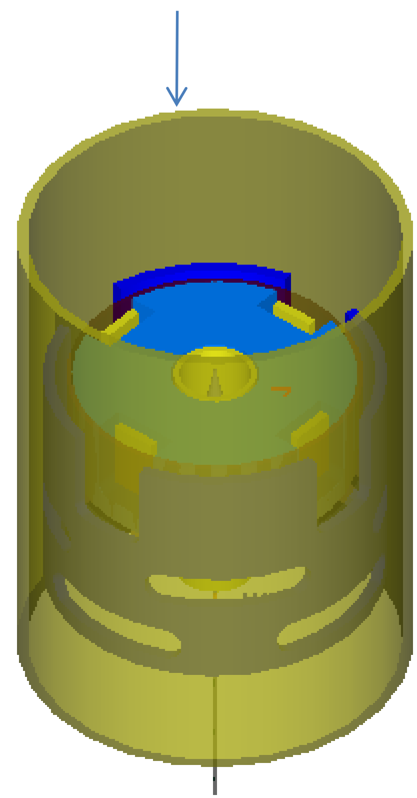


Create a metal shell as hole wall, the material is PEC.

“**Cylinder**” toolbar button

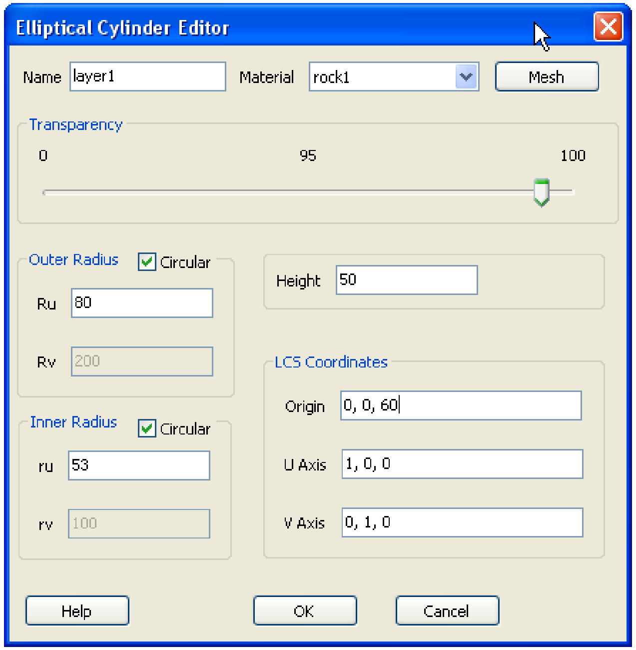


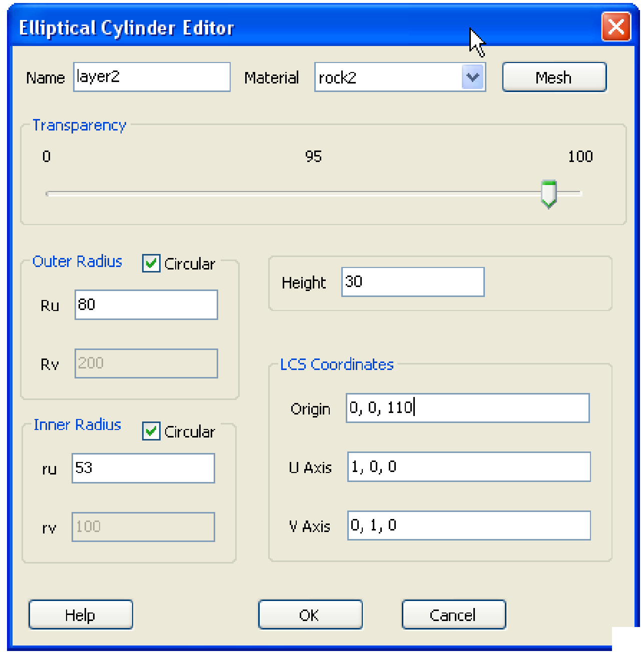
After **Shell** creation

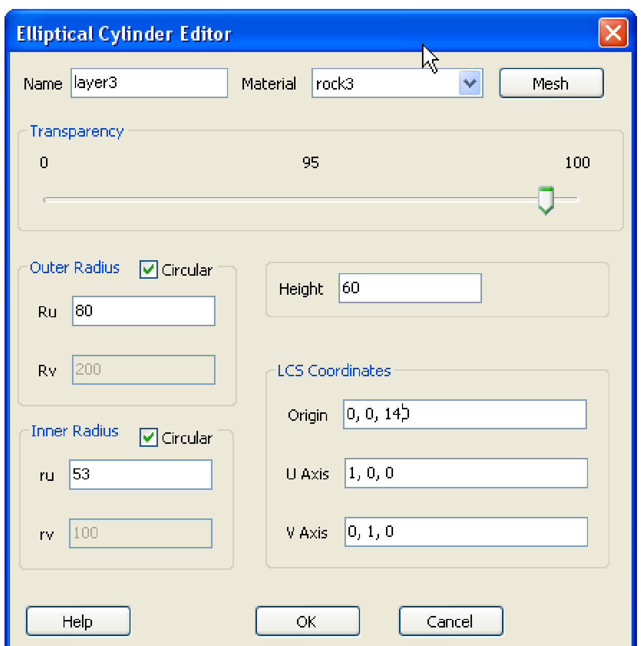


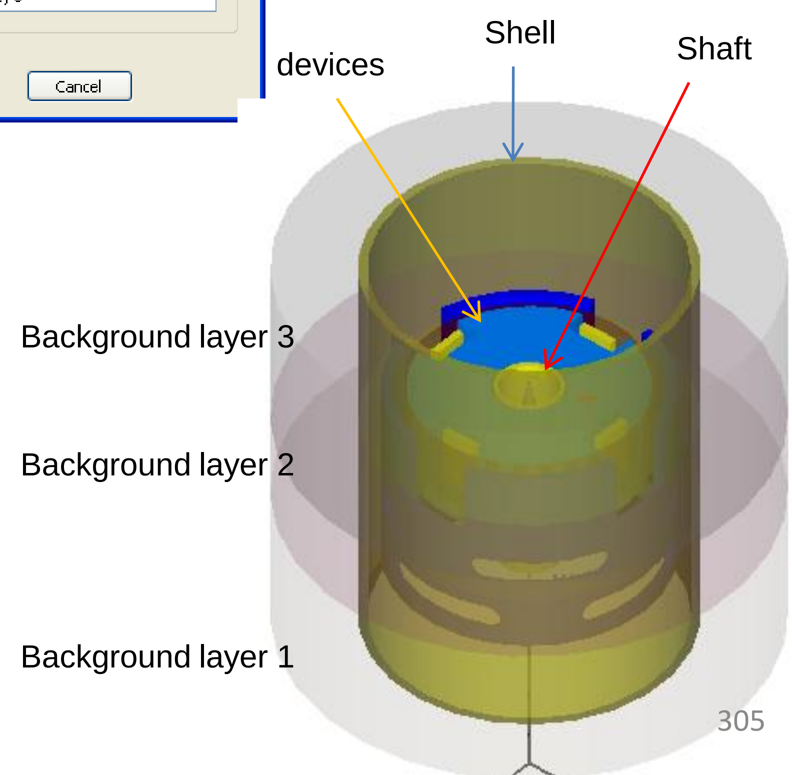
Note: in order to let the inner solids is still visible, it is suggested to let the **Shell** with some transparency.

Create 3 rings to represent 3 layers.

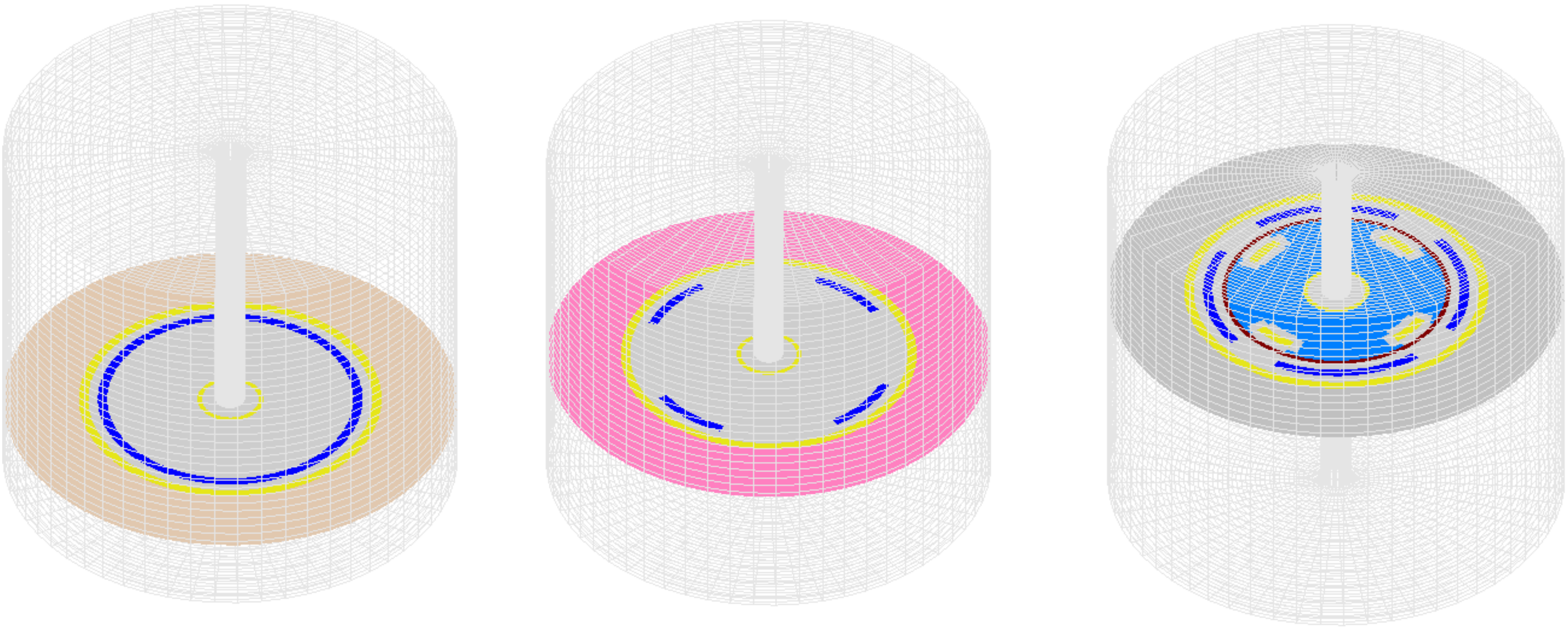








Then generate mesh or display it.



Case5

Setup a simulation project with imported structures from SAT file

This case will import 85 structures from a SAT file into a water background. Then use a ring dipole source to excite a BHW 1^{st} order pulse with a 12 ms time window. There are 100 observers along Z direction to record all velocities and stresses. The simulation is running with multiple threads.

The imported models are in file: **5-Transparent_Receiver_T-BONE_HSG_SWI(LG 8Win With Mass).SAT**

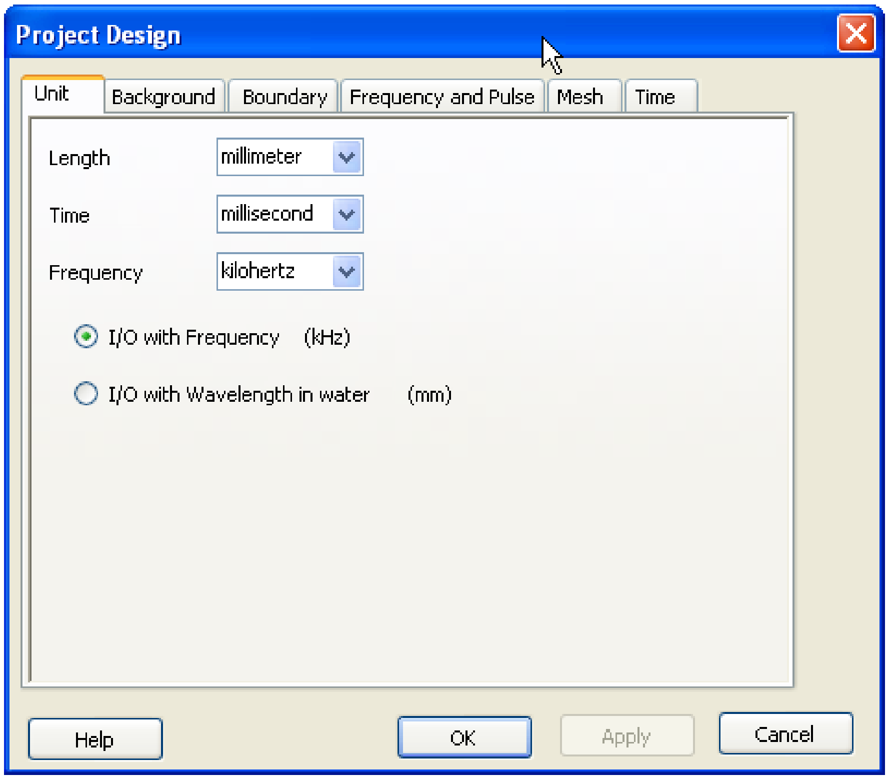
There are: **5-Transparent_Receiver_T-BONE_HSG_SWI(LG 8Win With Mass).MatDef**

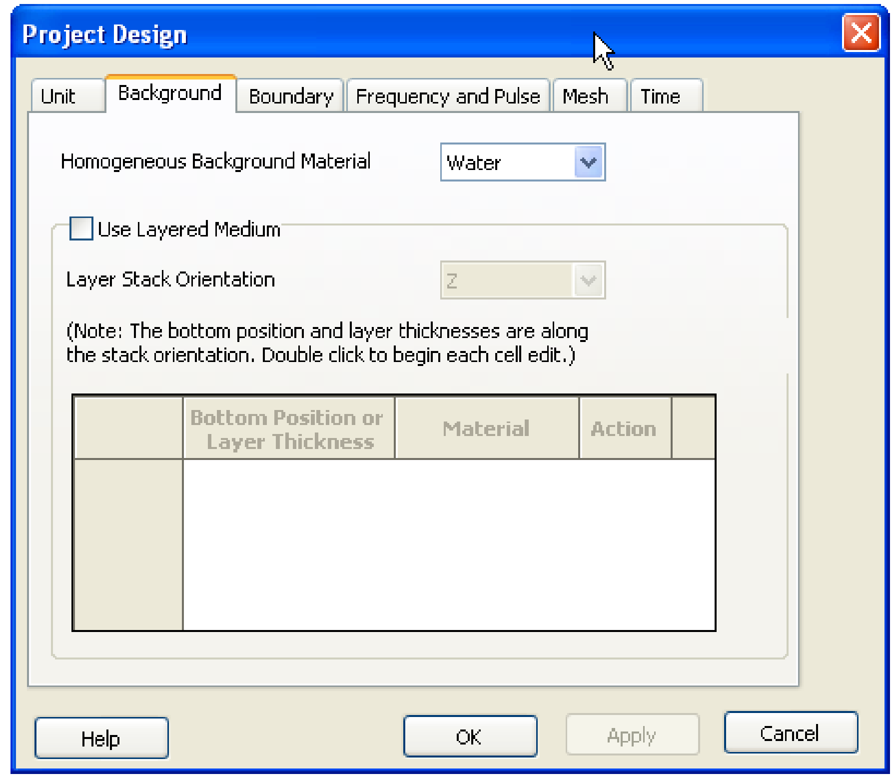
5-Transparent_Receiver_T-BONE_HSG_SWI(LG 8Win With Mass).SATMat

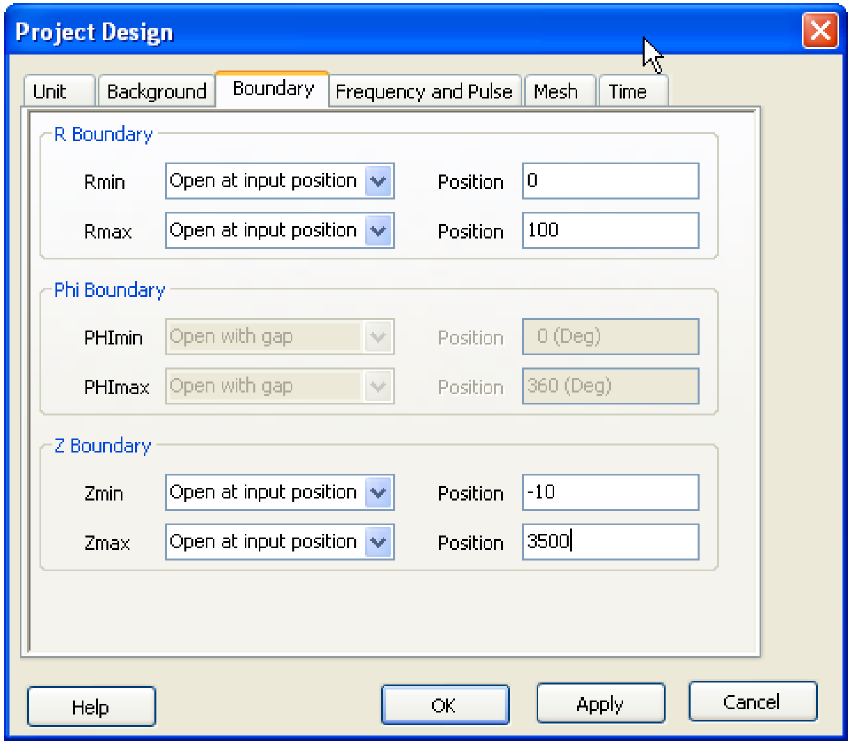
define the material.

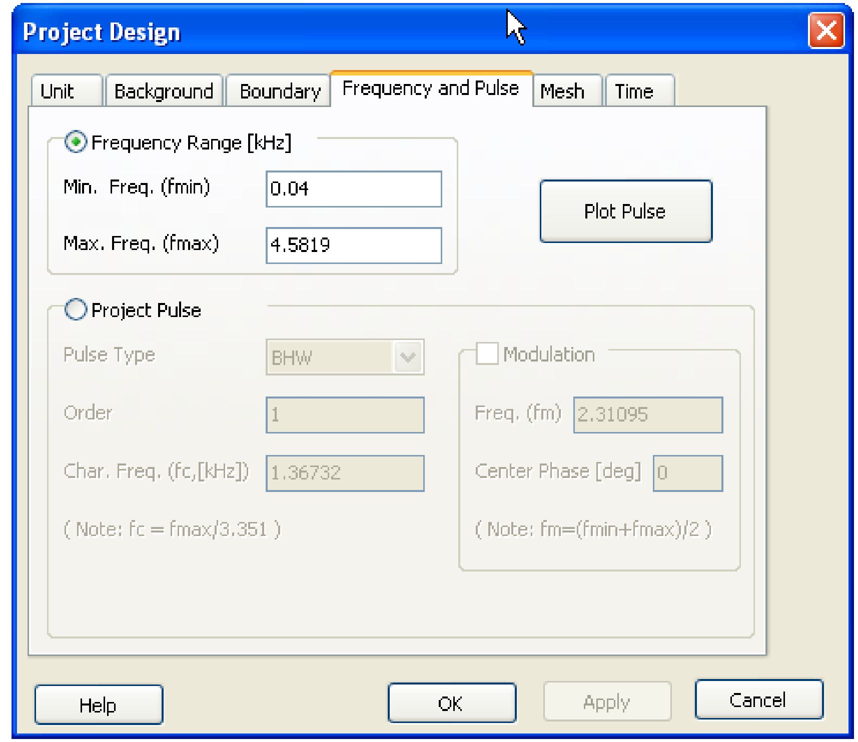
Please make sure the material name in **MatDef** and **SATMat** must be exactly the same. For example, for a name "Water 1", do not use "Water_1" in one file and use "Water 1" in another. Two files must use "Water 1".

Before loading the SAT file, please check whether all unit and project setting is reasonable.



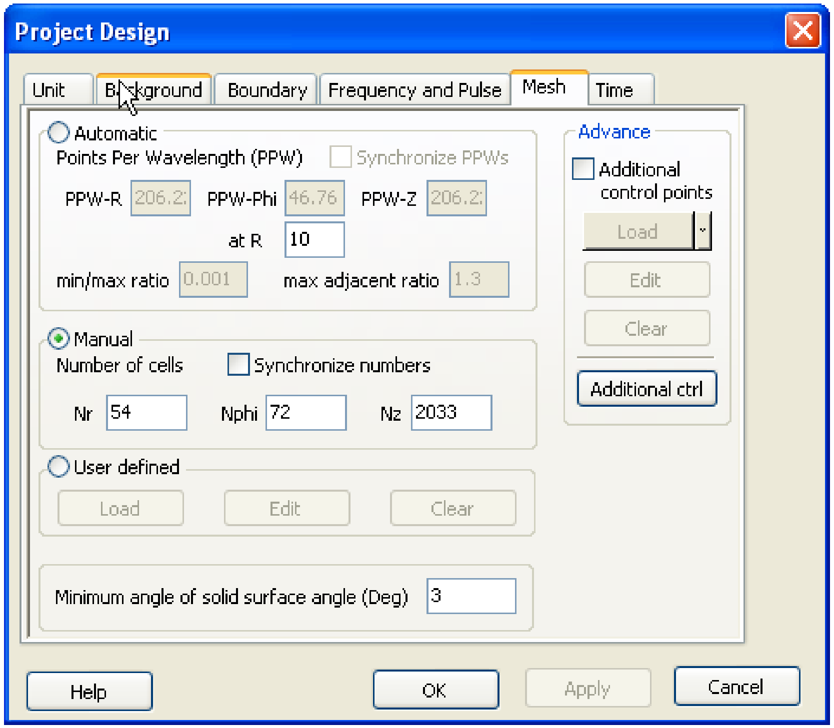


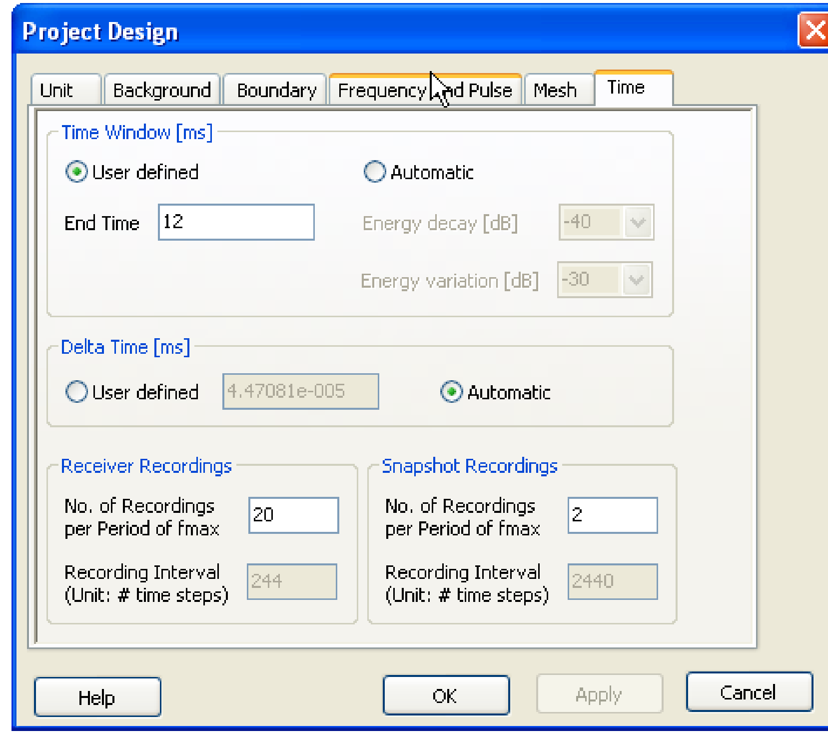




User can use default value, and adjust them to this correct values after loading.

User can use default value, and change them to design value before simulation.

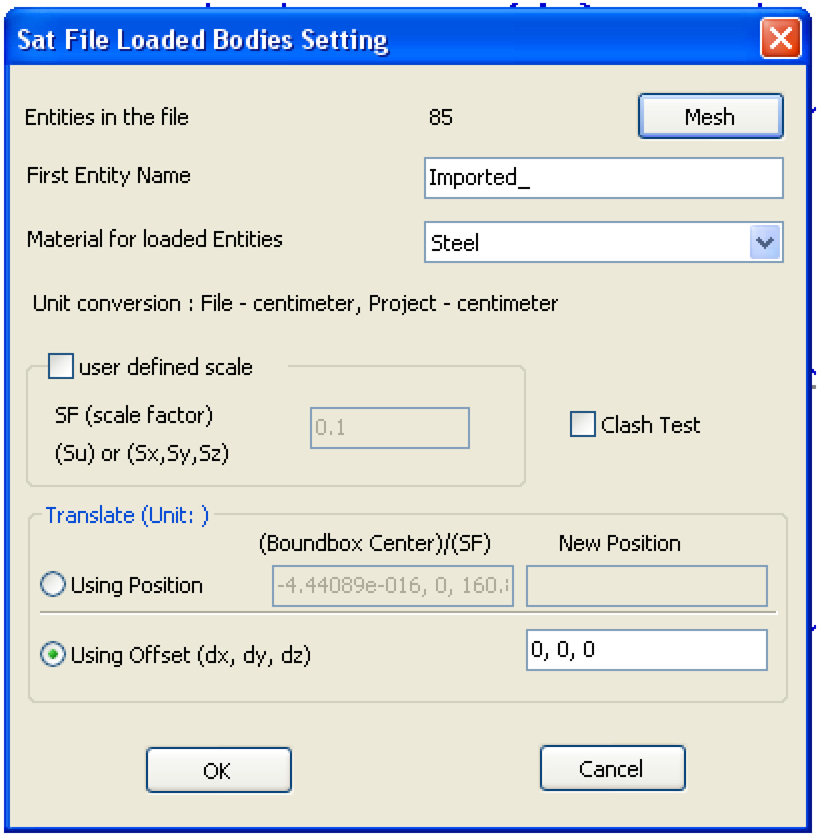




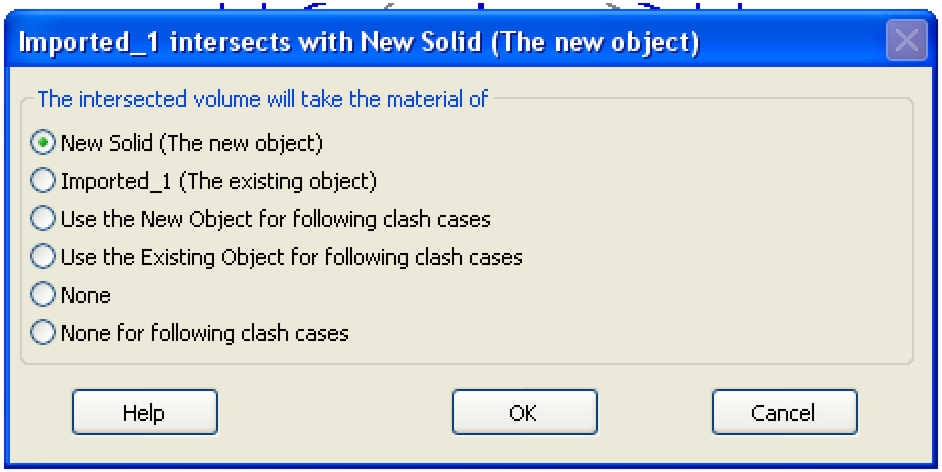
User can use default value, and change them to design value before simulation.

User can use default value, and change them to design value before simulation.

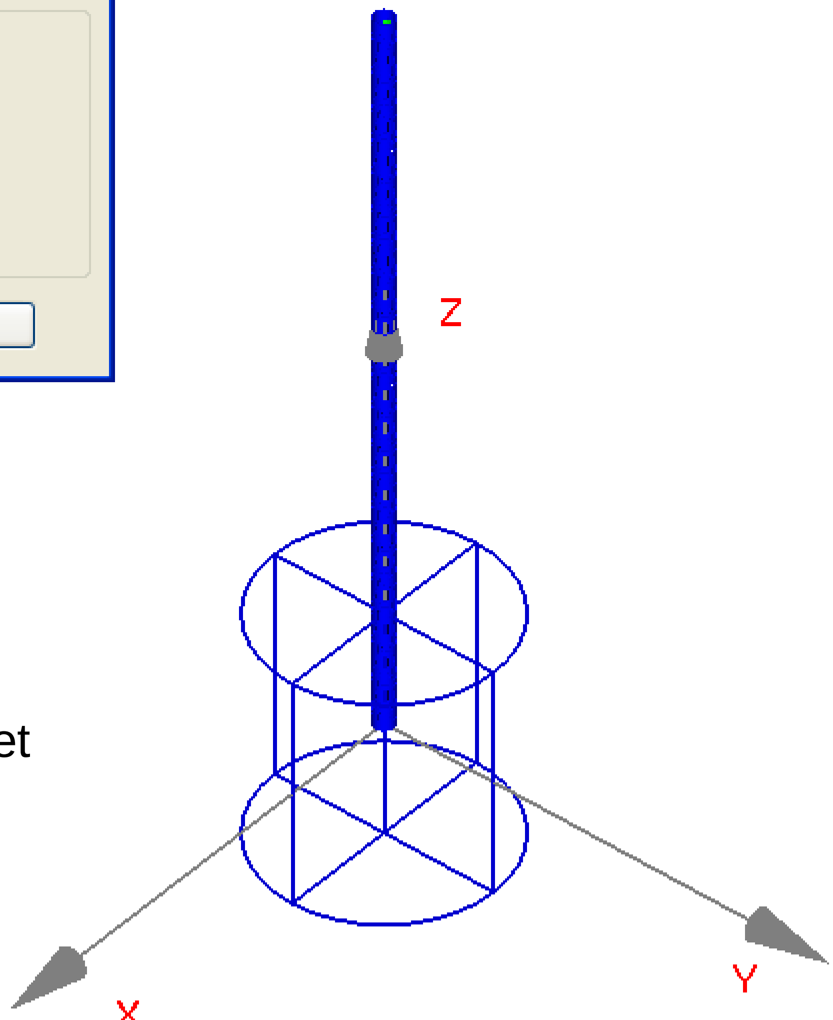
Insert SAT models, find 85 solids in the file



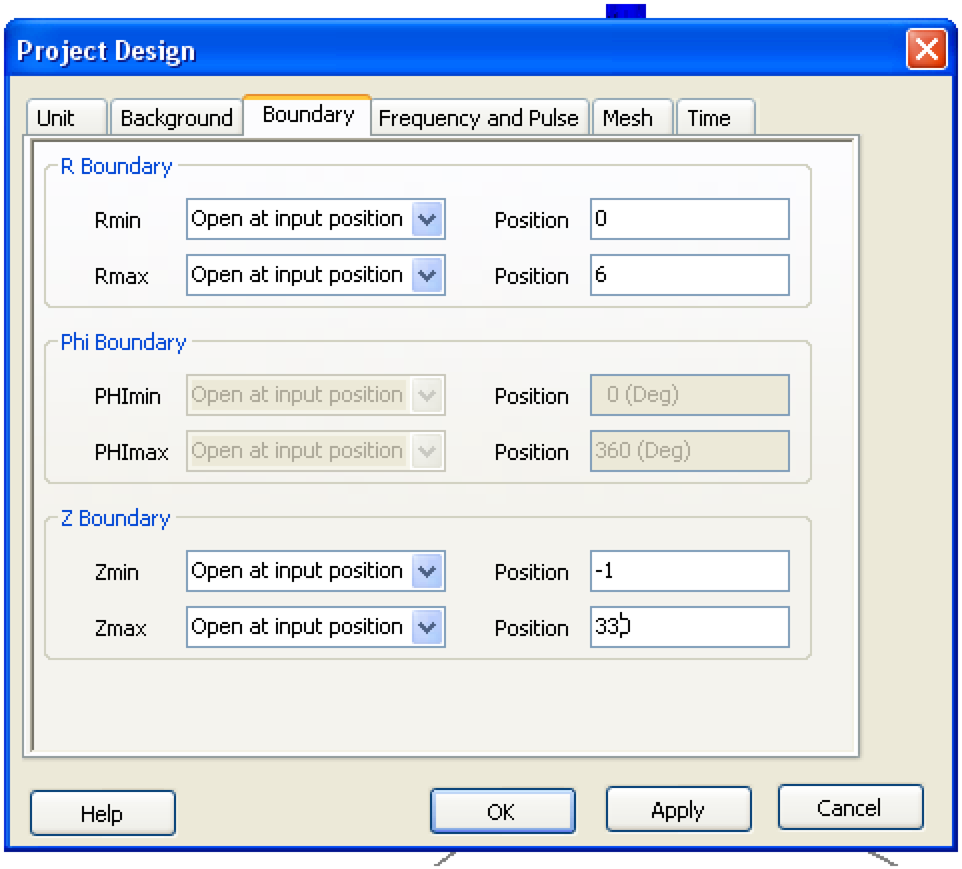
Find clash, but we consider it is due to tolerance. So, we use “**None for following clash cases**”



After Insert, we get

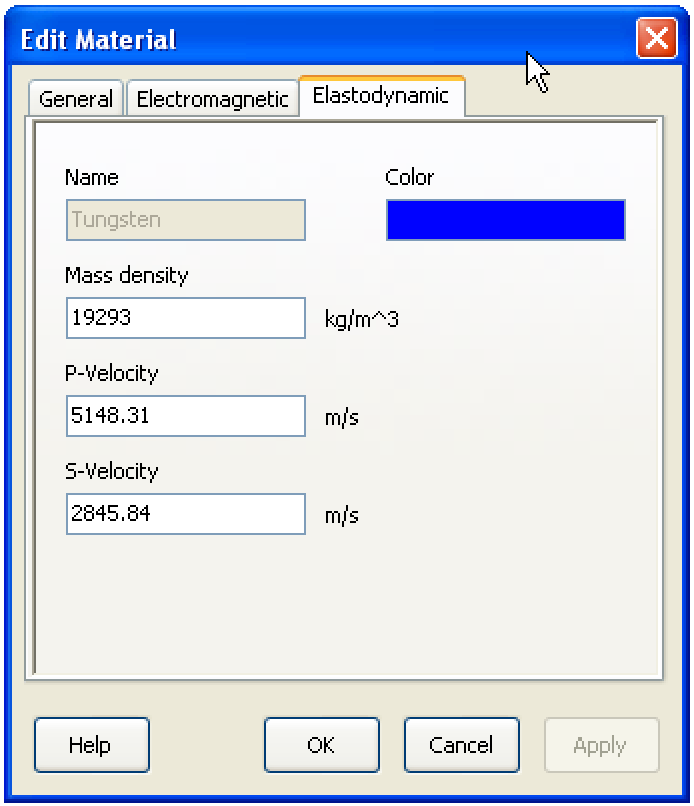


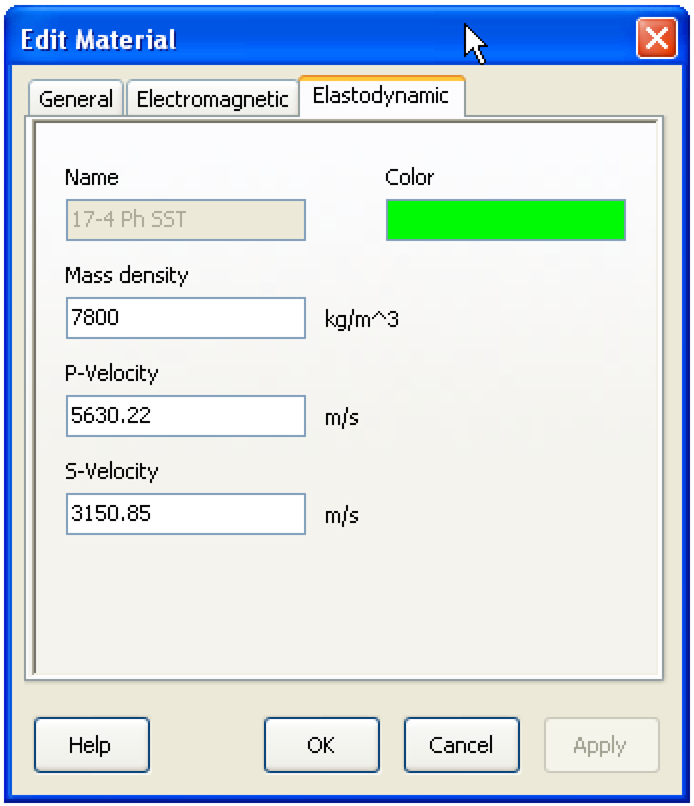
Adjust computation domain size to enclose all solids



Checking the Material property

After loading the SAT file with materials. User need to check the material to make sure the loaded material profile is correct.





Checking the Material-Solid mapping

User also need to check whether the imported models has correct material. For this case, it seems Imported_1 and Imported_3 has wrong material.

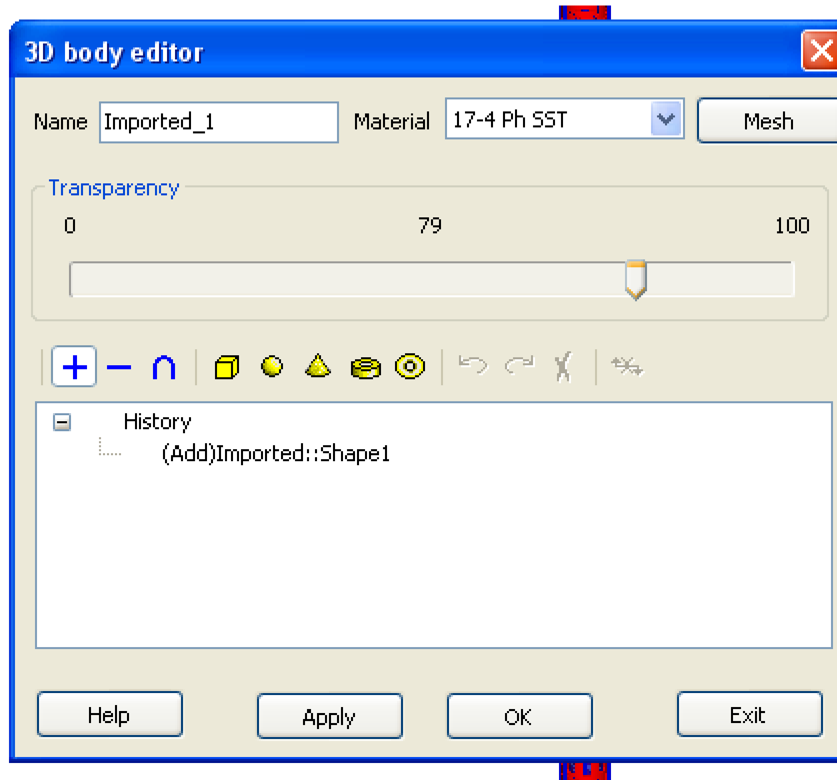
It is also suggested to change outside bar's transparency to make the case more viewable.

Imported_1 / 17-4 Ph SST

bounding box: (-46.0375,-46.0375,0),(46.0375,46.0375,3216.28

size: (92.075 x 92.075 x 3216.28) , center: (-3.55271e-015,0,16

(er,se)=(1,0) , (ur,sm)=(1,0)



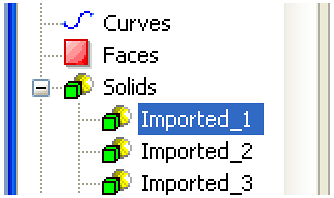
Set up cylindrical mesh control

This button is pressed

Firstly, design real computation size.

1) Make sure you are in the “Select Body” mode.

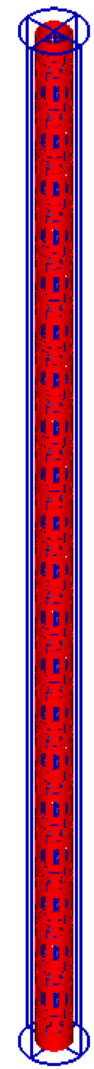
2) Select the outer bar to know the size of bar.
(select the **Imported_1** in the tree or use the mouse directly hit the bar. The bar will be high-lighted in the canvas)



The bounding box of bar is shown at the left corner of the main canvas.

As can be seen, the X, Y and Z Range of bar is x=[-46.0375, 46.0375] y=[-46.0375, 46.0375], z=[0, 3216.28]

Imported_1 / 17-4 Ph SST
bounding box: (-46.0375,-46.0375,0),(46.0375,46.0375,3216.28)
size: (92.075 x 92.075 x 3216.28) , center: (-3.55271e-015,0,1608.14)
(er,se)=(1,0) , (ur,sm)=(1,0)



So, in cylindrical system, the bounding box of the bar is r=[0, 46.0375], theta=[0, 360], z=[0, 3216.28]

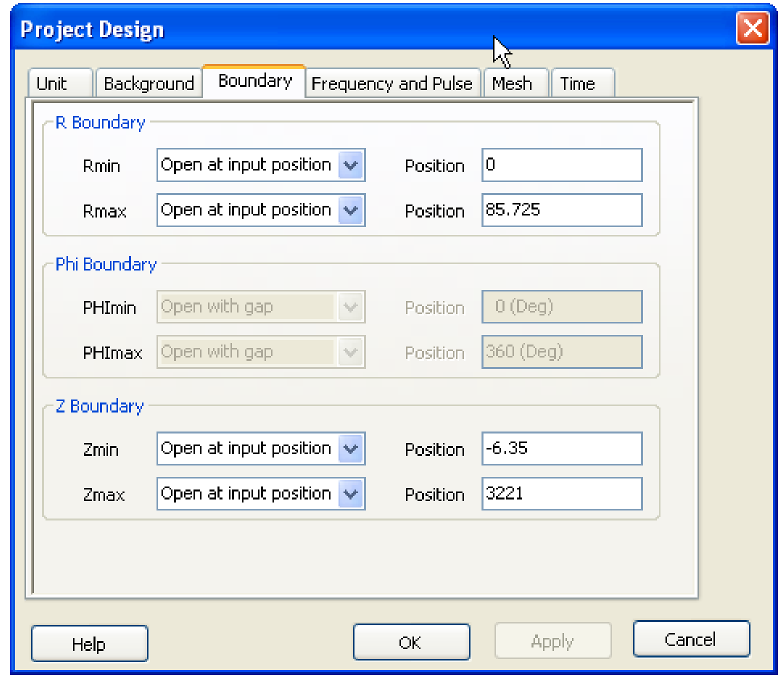
In this example, we use the uniform mesh control.

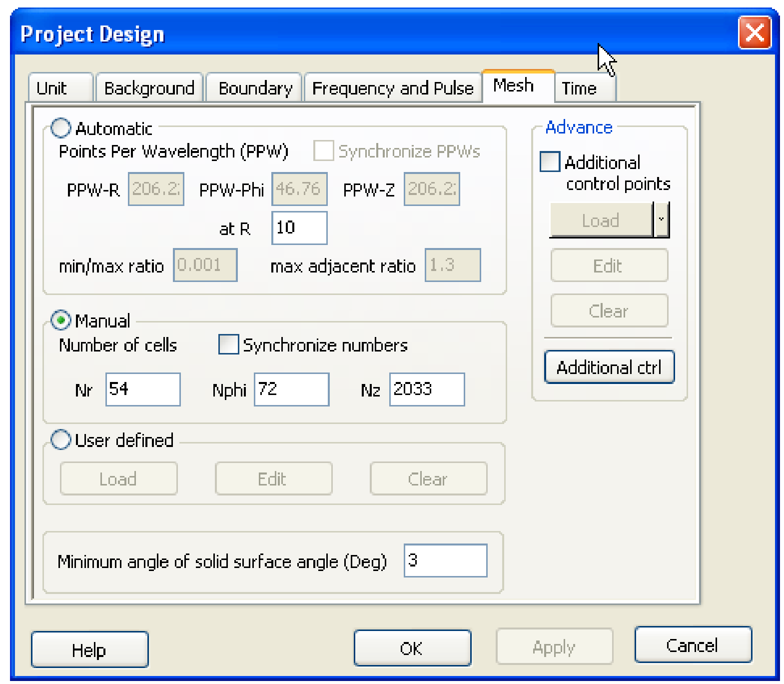
In the project, we want to use absorbing boundary conditions at Zmin, Zmax and Rmax.

We want to have at least 3 cells of water from the models to boundary at Z direction, and at least 15 cells of water from model to the Rmax boundary.

And we want the $\Delta r = \Delta z = 1.5875$ mm.

So, after calculation, we will setup following computation boundary and mesh control.

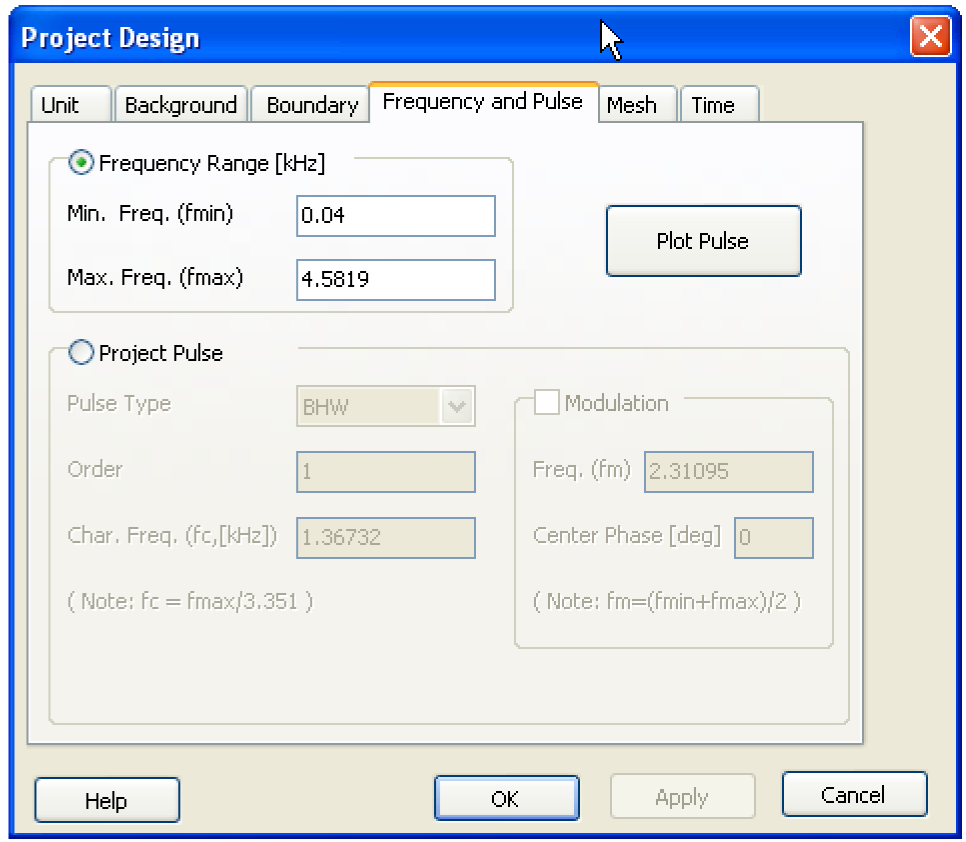




Set up fmax

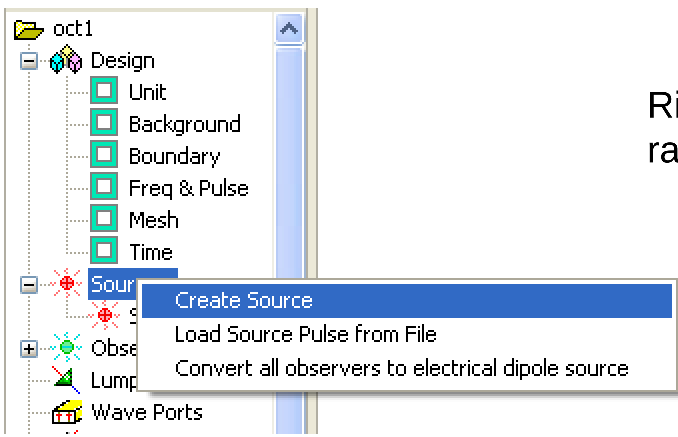
WCT use fmin and fmax to generate excitation signal. If user want to match their requirement, user need to convert their f_center or f_character to fmax. In order to reduce the time window of excitation, it is suggested to let fmin≈0.01fmax.

In this case, we define fmax=4.5819 KHz.
fmin ≈ 0.01fmax ≈0.04 KHz.



Define source

Here, we want to use a ring dipole source at the bottom of bar to excite.

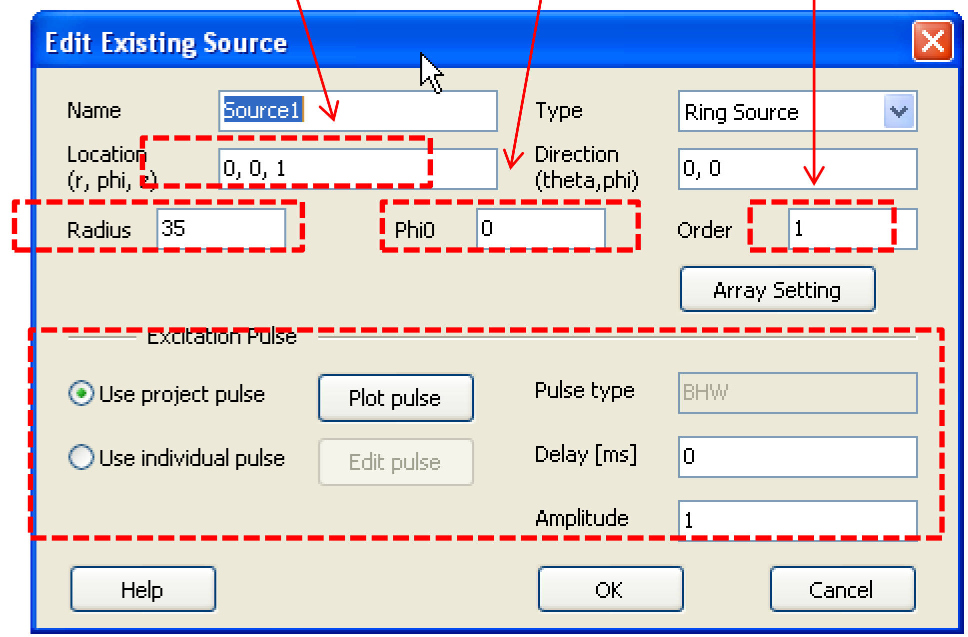


Ring radius

Ring axis

Reference angle

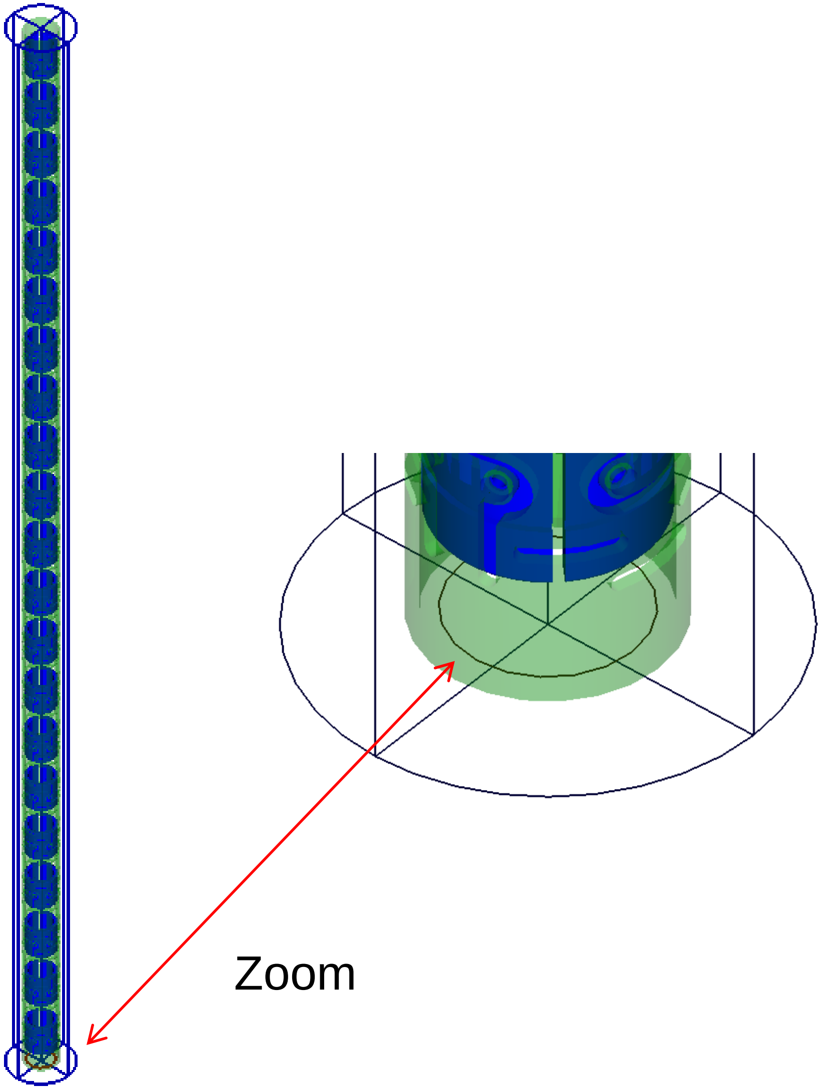
0: ring monopole
1: ring dipole



Default pulse is 1st BHW already

Check source position

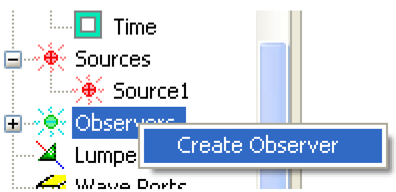
By select the tree item in the tree. The source will be highlighted in the canvas.



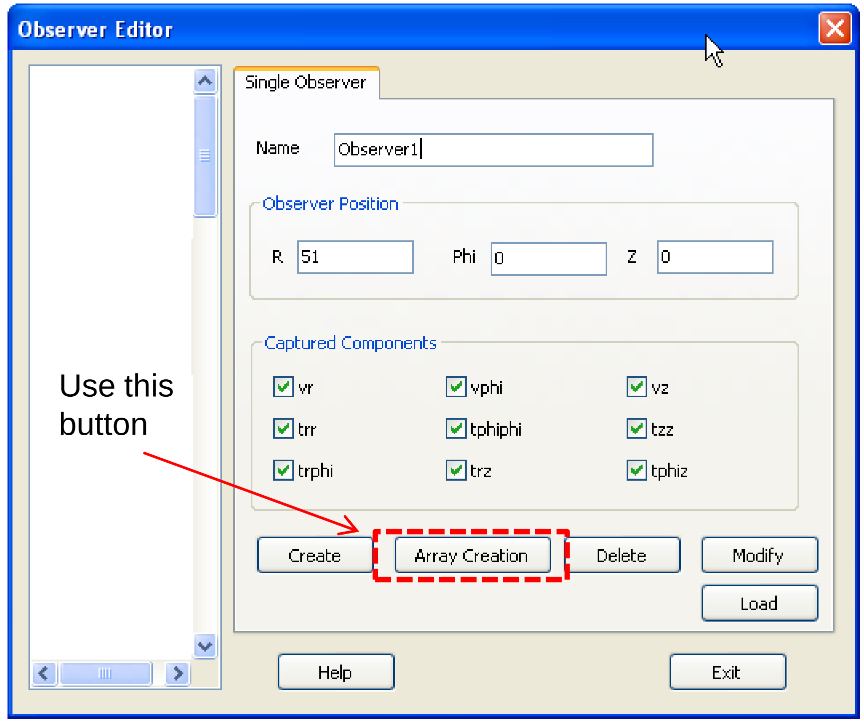
Define observers

Use **Array Creation** to simplify the work.

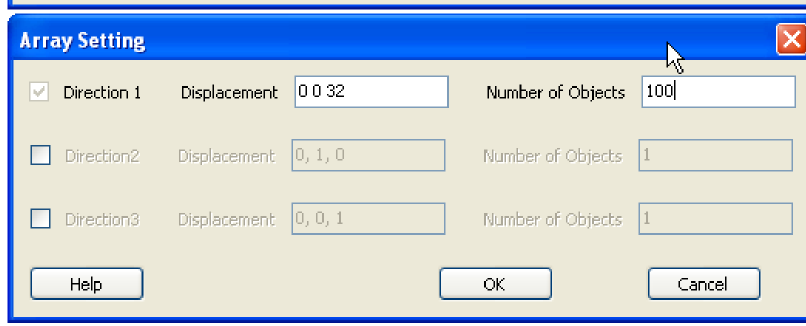
Here, we will place 100 observers 3 cells away from the outer range of bar. The 100 observers will along Z direction.



Define the prefix of observers and the start position of the observer array.



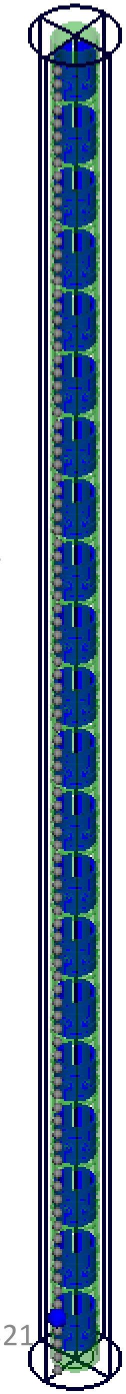
Use this button



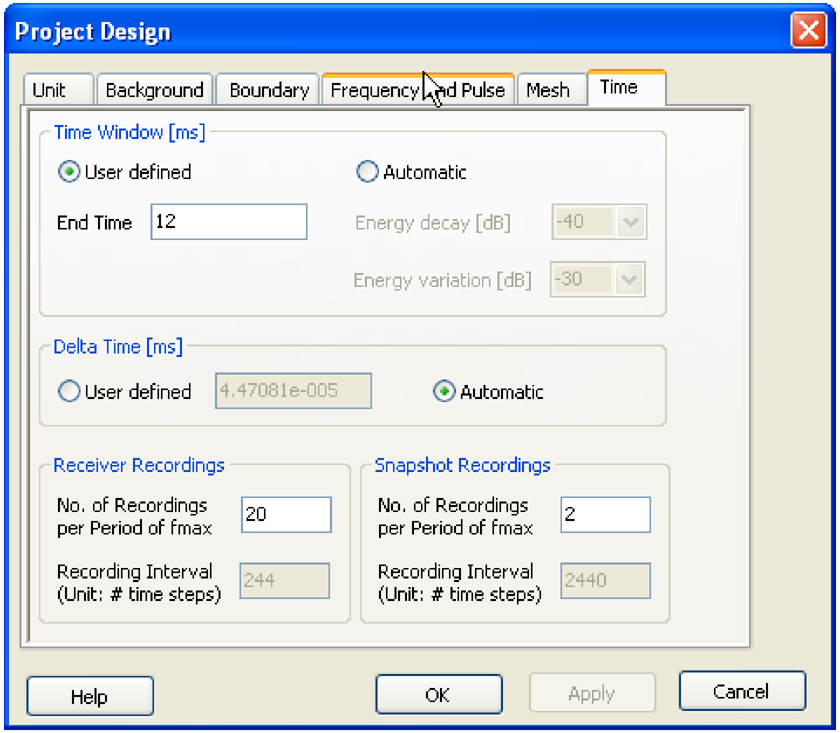
Adjacent observers distance

Number of observers in array

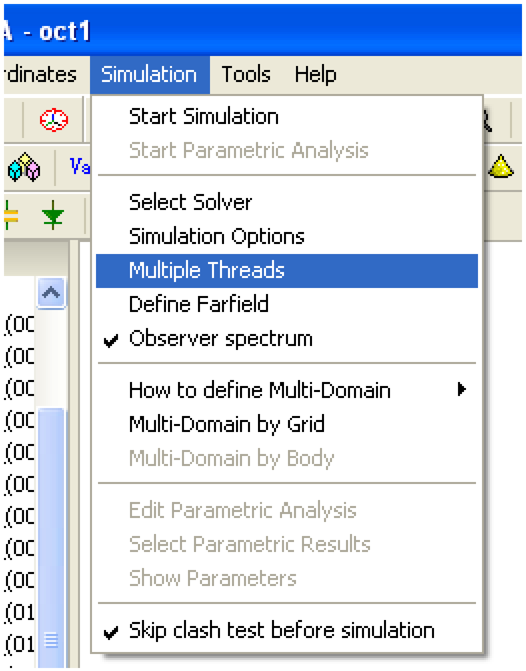
The gray spheres are observers

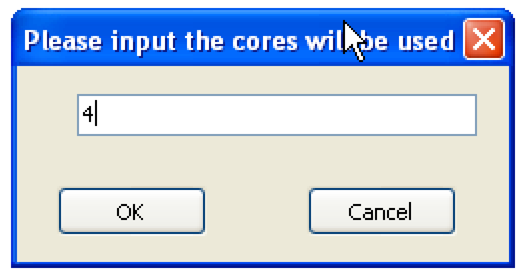


Define the simulation time window & observer recording interval

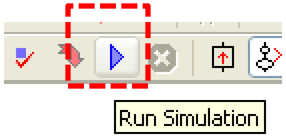


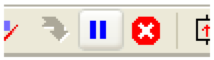
Define the many threads will be used to run simulation parallelly



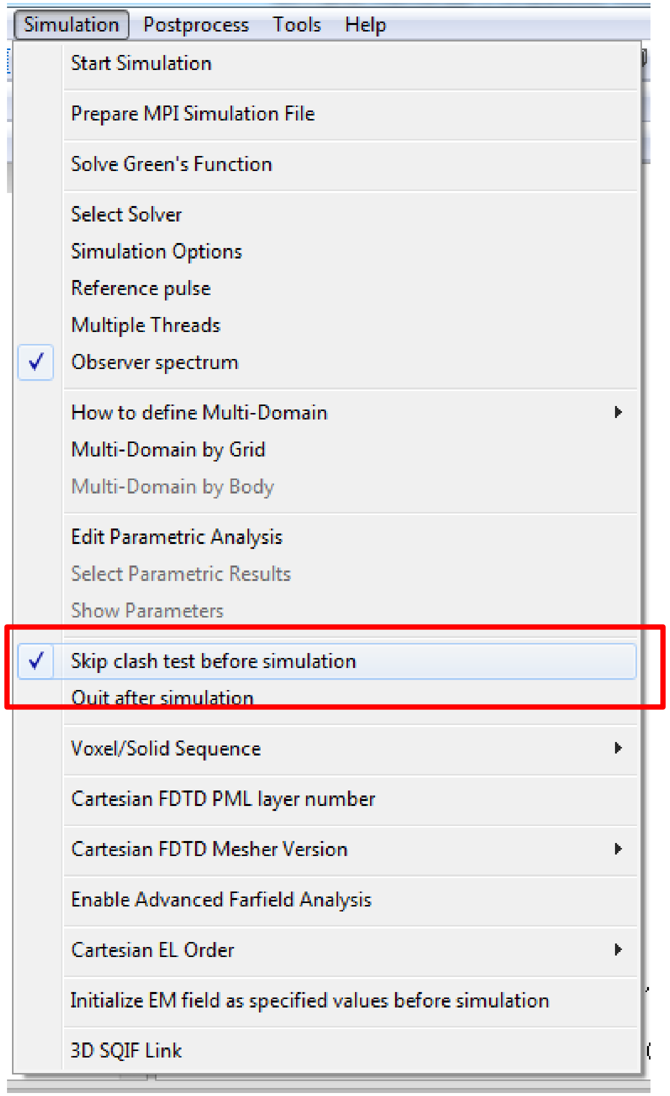


Run the simulation





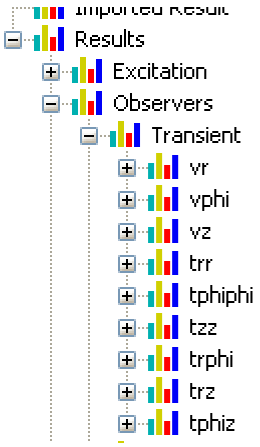
User can ***Pause*** or ***Stop simulation***

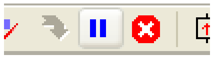


Note: if the clash test is ignored in loading the SAT file, and the solids have not significant clash problem (means even if they have, but these are due to tolerance, will not effect final simulation result, then user can enable "***skip clash test before simulation***" option to skip project clash validation. Otherwise, the project will not pass the validation and can not run before user eliminate all clashes by "Boolean" operation.

Watch the result

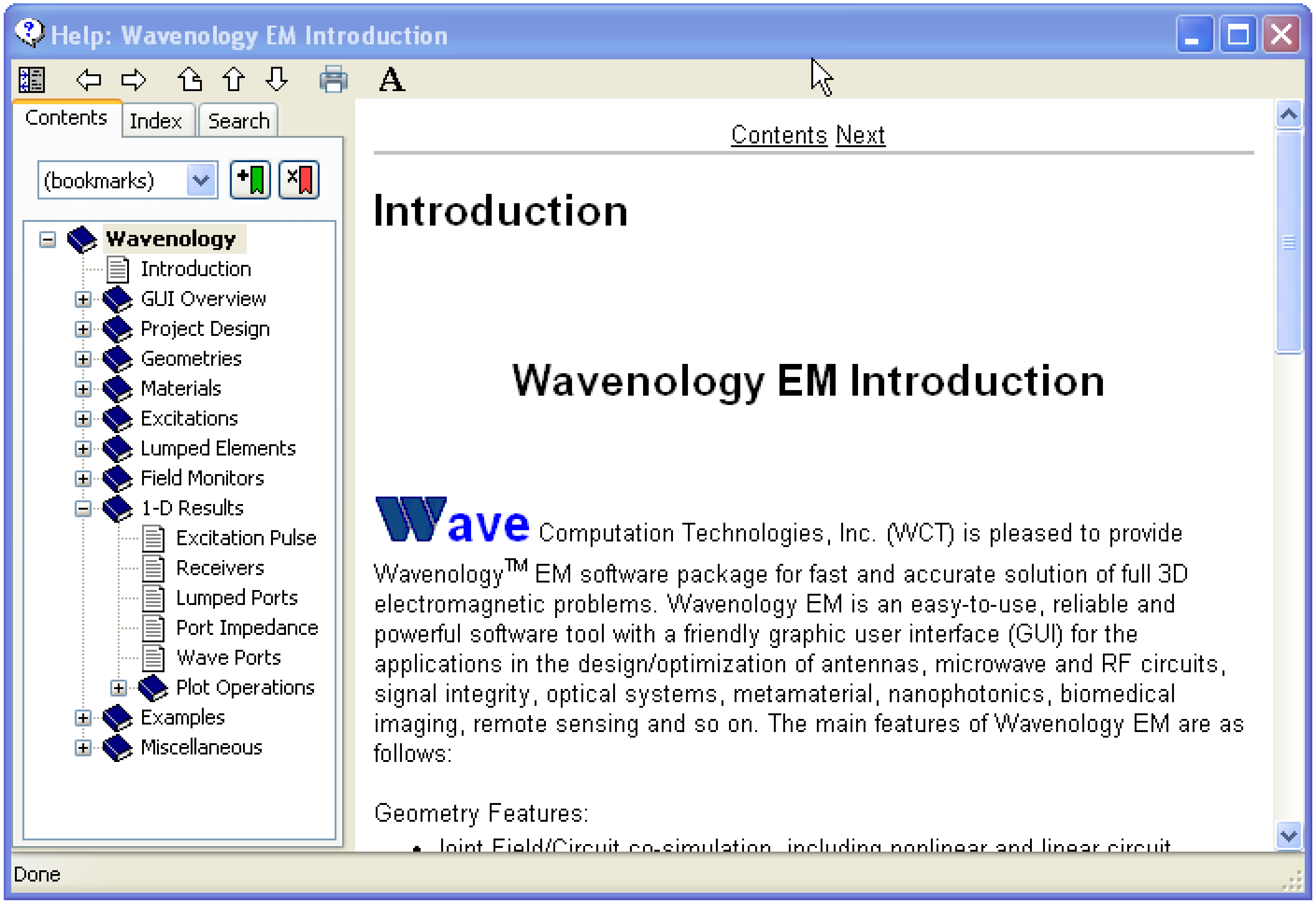
User can watch the result after the simulation or in the middle of simulation.





Normally, WCT will automatically **refresh** result at GUI. So, in the middle of simulation, user can watch result without waiting utill the end of simulation. But this refresh interval depends on the simulation speed. If user want to watch the result before the **refresh**, user can press “Pause simulation” button to force GUI to refresh result then press “resume simulation” button to continue simulation.

For displaying and IO the 1D observer result, please refer to the “1-D Results” section in the manual of Wavenology EM package



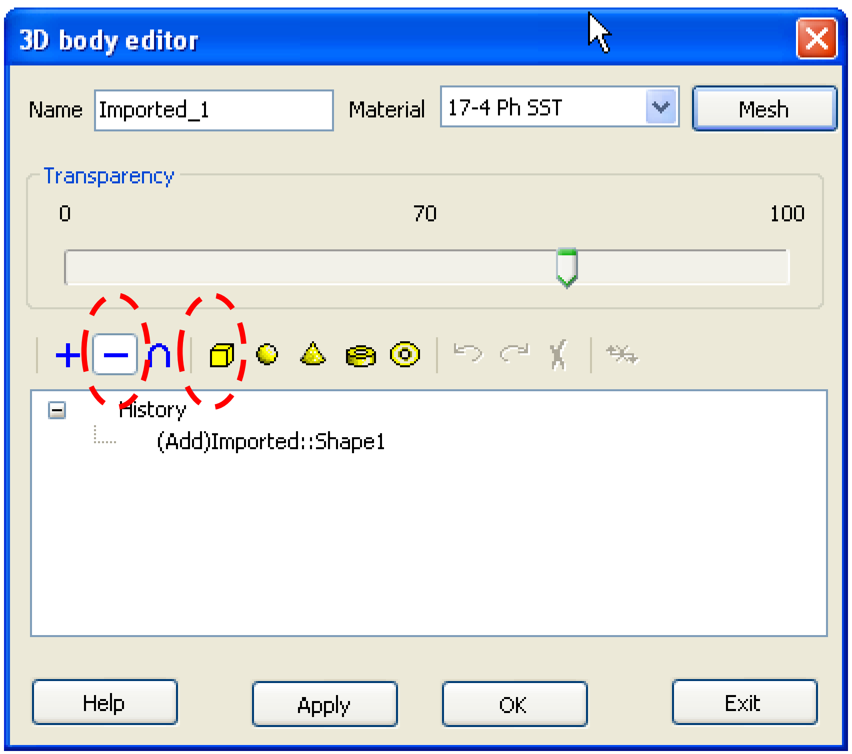
Case 6
Simplify Existing Project to a New Model

The case 6 is long in Z direction, need a long simulation time. Sometimes, user want to run part of the structures in case 3 in a shorter time window. In this example, we will demonstrate to run 0-500 mm (in Z) part of the full models.

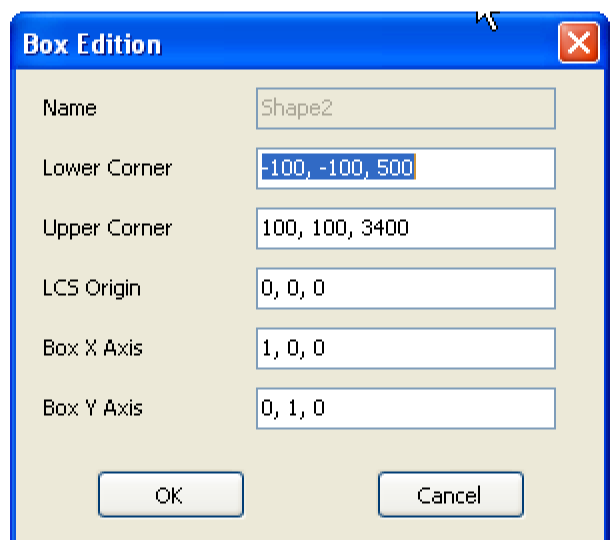
Save last case to another folder, for example, ***Case4***.

Delete all observers.

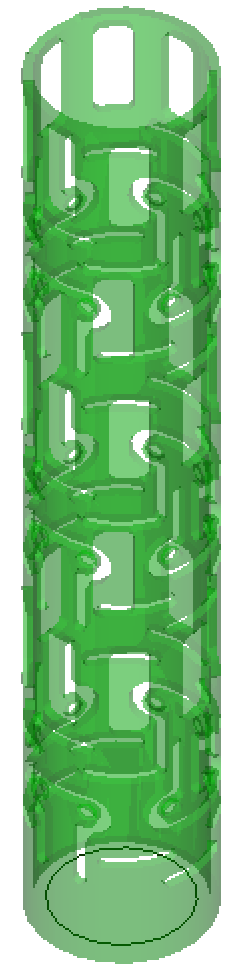
Then cut off the 500-3300 mm part from **bar** by editing “Imported_1”:



Cut off top part by subtract a **BOX** from this solid.
Check “**Subtract shape**” button, then press “**Box**” button to define subtract volume.



After subtraction

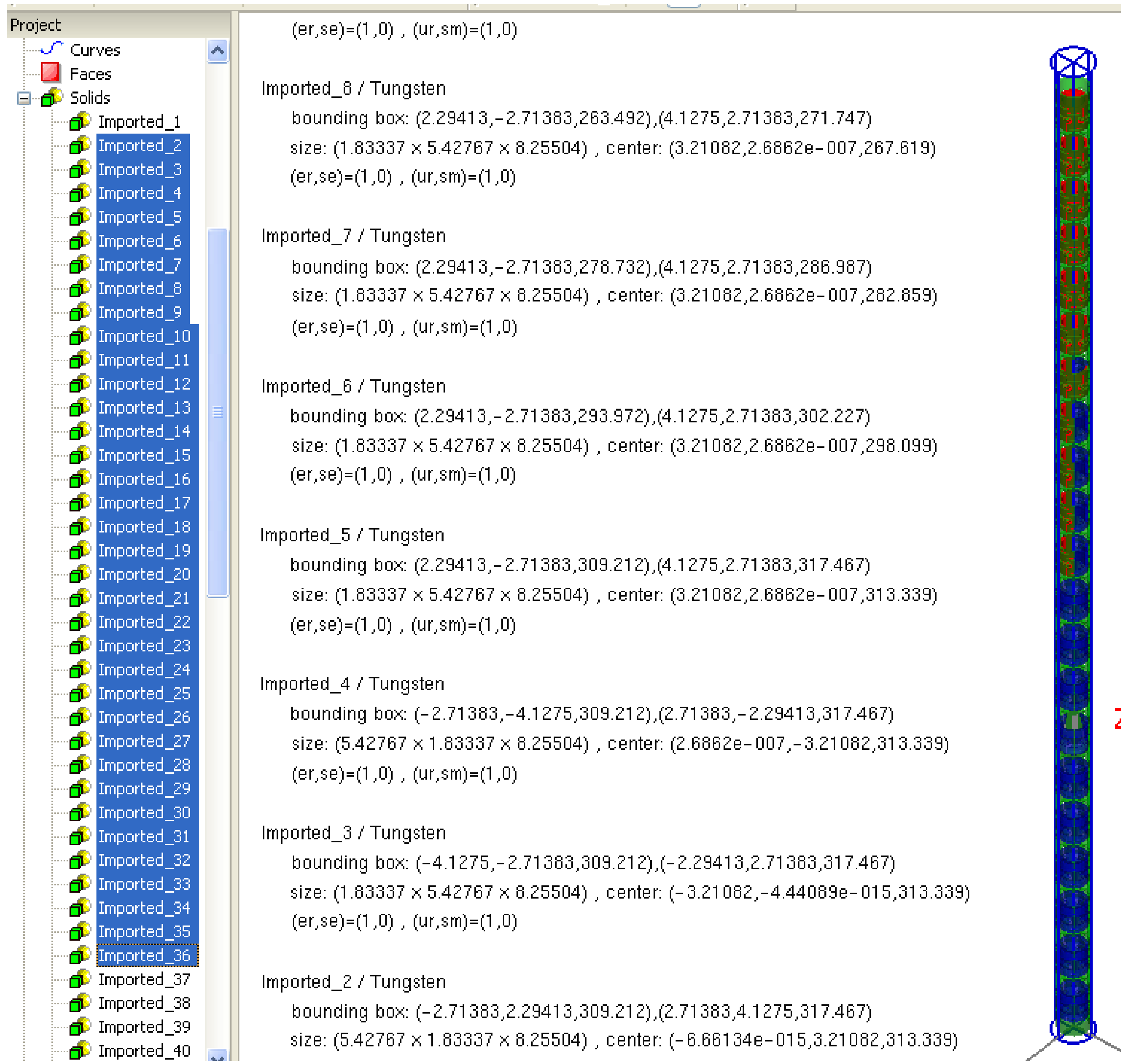


Delete solids outside the bar.

Use “Ctrl” key can multiple select solids.

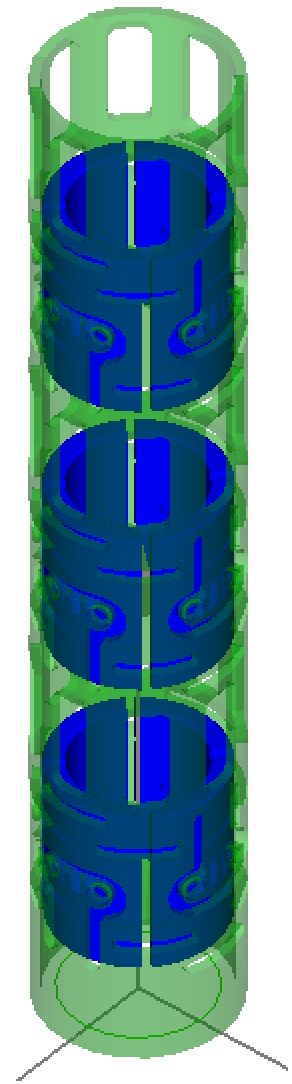
Then use “Del” key or “delete” button

to delete selected solids

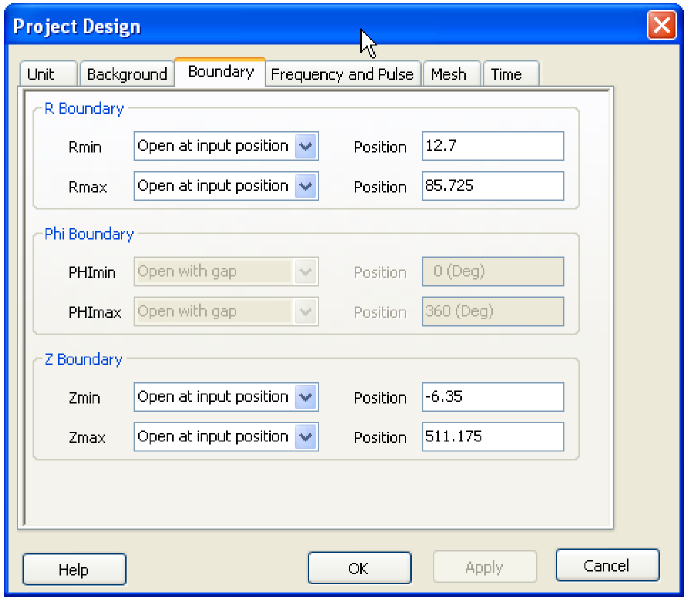


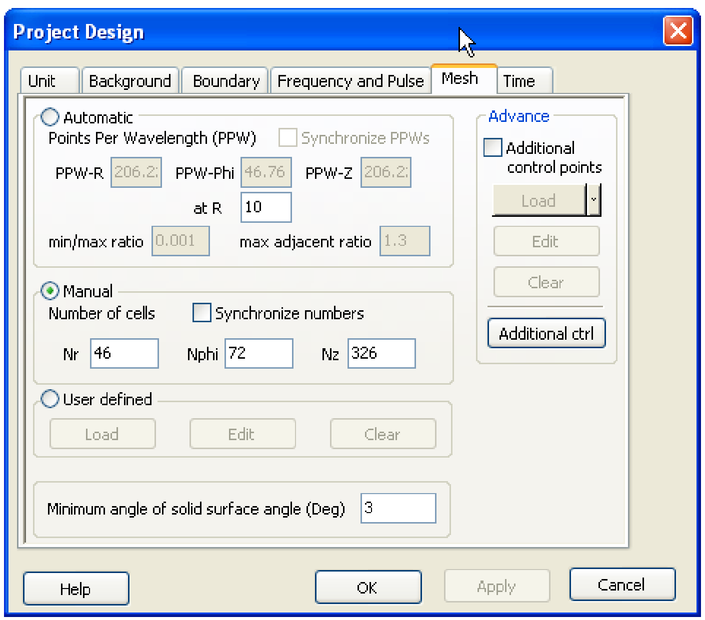
Final solids

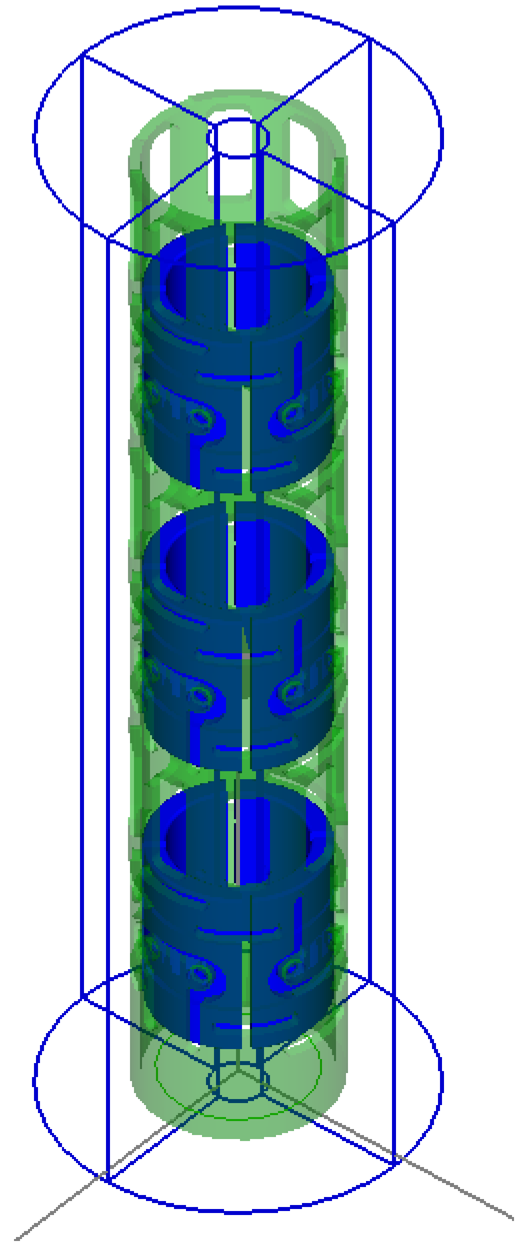
- Solids
  - Imported_1
  - Imported_53
  - Imported_54
  - Imported_55
  - Imported_63
  - Imported_64
  - Imported_65
  - Imported_73
  - Imported_74
  - Imported_75
  - Imported_83
  - Imported_84
  - Imported_85



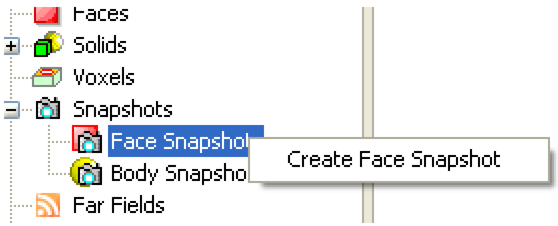
Adjust Z size of computation domain

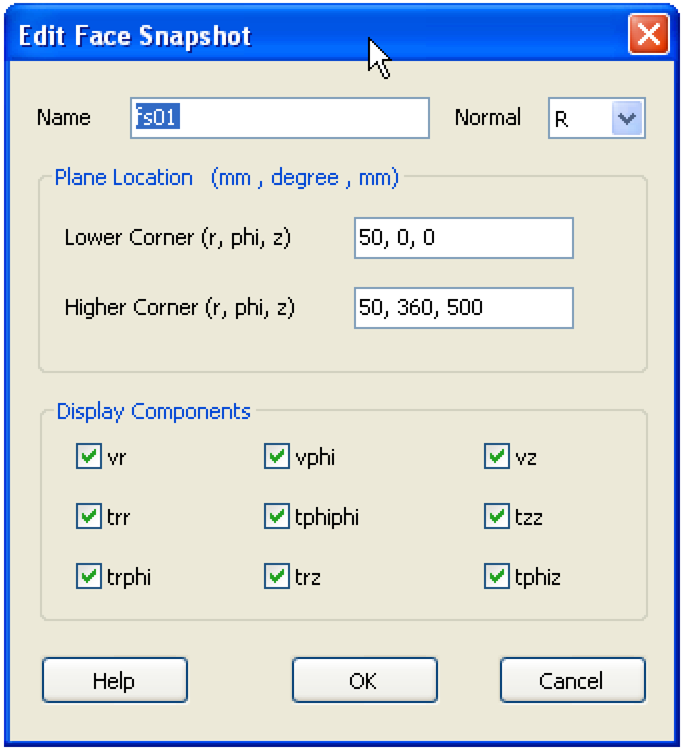




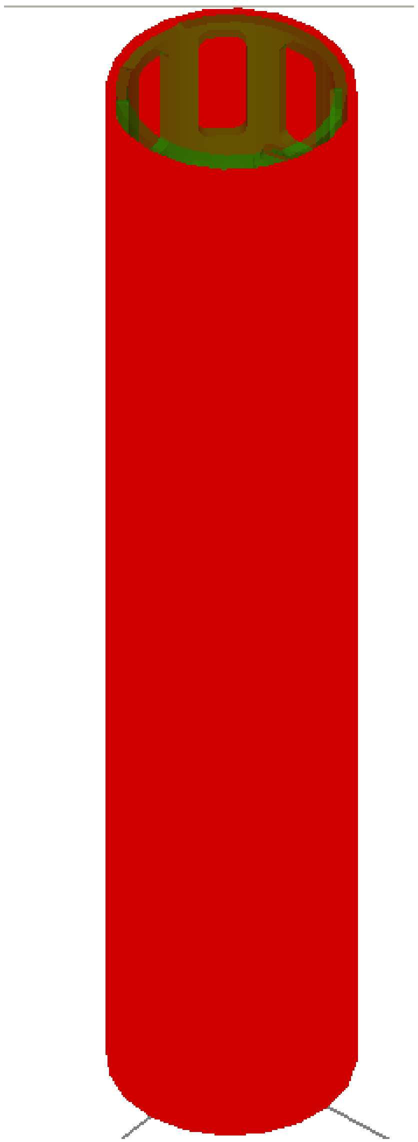


Define a R plane snapshot





Snapshot position



Define 15 observers

Then define the threads number and run the simulation.

End