# Omega3P & S3P Tutorials

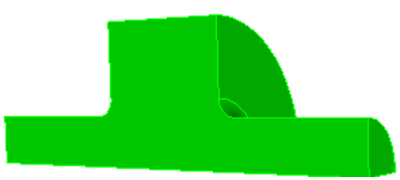
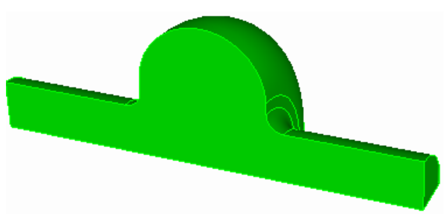
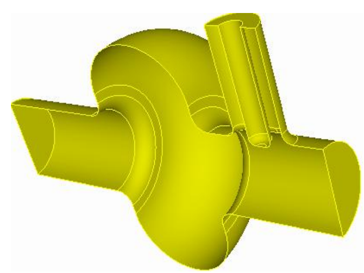
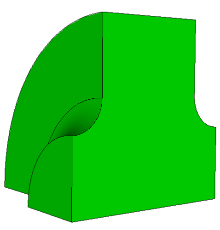
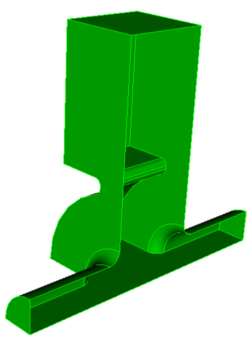
SLAC CW10  9/20-22/2010

# Workflow For ACE3P Codes

1. CUBIT – generate model
2. CUBIT – mesh the model
3. Acdtool – convert “GEN” mesh to “netcdf” format (& check)
4. Run Omega3P/S3P …
5. Postprocessing
    1. Paraview
    2. Acdtool rf postprocess

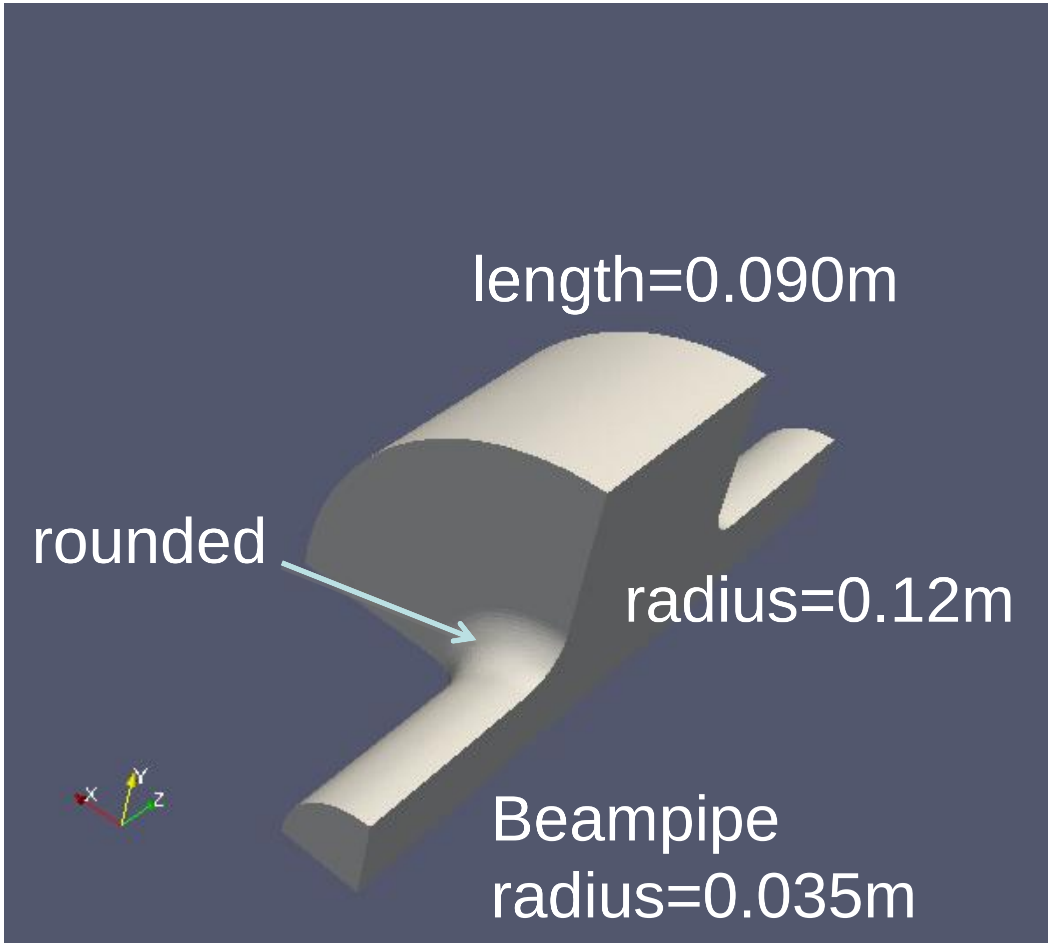
# Omega3P Examples

- Pillbox cavity
- Rounded top cavity
- Cavity with coax coupler
- Traveling wave structure
- Cavity with waveguide coupler

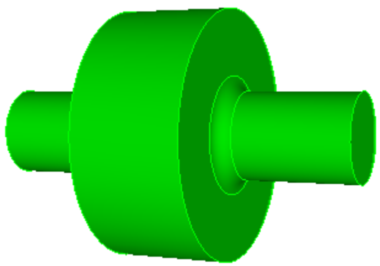






# Example 1: Pillbox

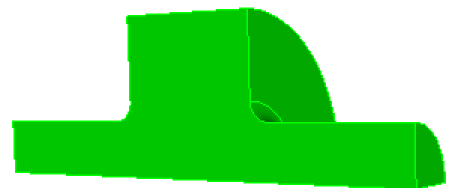
- A quarter geometry with following specification
- Lossless



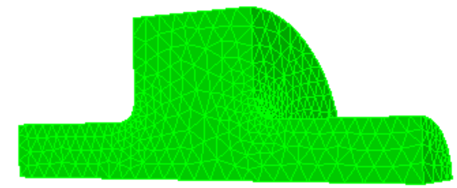
# Pillbox – sw1



1) model



2) symmetry cut



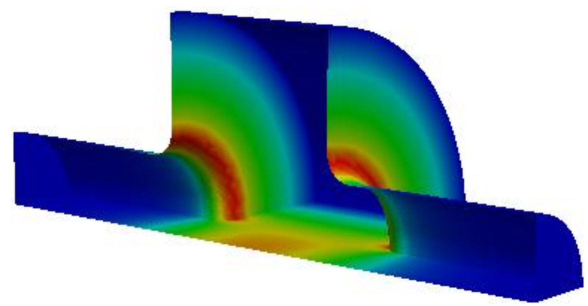
3) mesh

“sw1.ncdf”

**Run Omega3P**

```
  3642 Jul 22 09:32 logfile.output
233544 Sep  3 11:59 mode.l0.1.191206E+09.m0
233544 Sep  3 11:59 mode.l0.2.064482E+09.m1
388212 Sep  3 10:41 model1.ncdf
   450 Jul 10 08:47 o3p.in
  7511 Sep  3 11:59 output
  5285 Jul 22 09:26 sample.input
   512 Sep  3 11:59 vector1/
```

4) Solver



5) postprocess

# Creating Input File for Pillbox

```
ModelInfo : {
 File: sw1.ncdf

 BoundaryCondition : {
   Magnetic: 1, 2
   // Electric: 3, 4
   Exterior: 6
 }
 SurfaceMaterial : {
   ReferenceNumber: 6
   Sigma: 5.8e7
 }
}

FiniteElement: {
 Order:          2
 CurvedSurfaces: on
}

EigenSolver : {
 NumEigenvalues: 2
 FrequencyShift:  0.5e9
}

PostProcess : {
 Toggle: on
 ModeFile: m
 SymmetryFactor: 4
}

CheckPoint: {
  Directory: vector1
  Act:       save
}
```

# ModelInfo

* Mesh file
* Boundary Conditions
* Material properties

# ModelInfo - File & BoundaryCondition

* Specify the file name of the mesh (in netcdf format)

File: sw1.ncdf

* Specify the boundary conditions on each set of reference surfaces – in Cubit, they are called sideset.

```
BoundaryCondition : {
  Magnetic: 1, 2
  ##Electric: 3, 4
  Exterior: 6
}
```

# ModeInfo – Material Properties

* Surface Material

```
SurfaceMaterial : {
  ReferenceNumber: 6  //identified by sideset ID in cubit
  Sigma: 5.8e7
}
```

* Volume Material

```
Material : {
  Attribute: 1   //identified by the block ID in cubit
  Epsilon: 1.0
  Mu: 1.0
 // EpsilonImag: -0.2 //lossy material
}
```

# Finite Element Parameters

```
FiniteElement: {
 Order:          2              //finite element basis order
 CurvedSurfaces: on  //use curved elements
}
```

# Eigen Solver Options

```
EigenSolver : {
 NumEigenvalues: 2          //Number of modes
 FrequencyShift:  0.5e9   //Frequency of interest
}
```

# Eigen Solver Options (2)

```
EigenSolver : {
 NumEigenvalues: 2          //Number of modes
 FrequencyShift:  0.5e9   //Frequency of interest
 Preconditioner: MP //this use p-version of
                                        //multilevel preconditioner.
}
```

# Eigen Solver Options (3)

```
EigenSolver : {
 NumEigenvalues: 2          //Number of modes
 FrequencyShift:  0.5e9   //Frequency of interest
 Memory: 1000 //if the memory usage of the matrix
                              // factor in any process is larger than
                              // 1000MBytes, switch to
                             //use out-of-core solver.
}
```

# Postprocess

```
PostProcess : {
 Toggle: on              //switch, set off to turn off postprocessing
 ModeFile: mode    //mode file prefix
 SymmetryFactor: 1
}
```

# CheckPoint

```
CheckPoint: {
  Directory: vector1 //save eigenvectors for later use
  Act:       save
}
```

# Different Options in Input File

* https://confluence.slac.stanford.edu/display/AdvComp/Omega3P+Sample+Inputs

* https://confluence.slac.stanford.edu/display/AdvComp/S3P+Sample+Inputs

* https://confluence.slac.stanford.edu/display/AdvComp/T3P+Sample+Inputs

# Run Omega3P

* Run omega3p in serial

  omega3p sw1.o3p

# Screen Output

```
*****************************************************************
****          Omega3P V8.2.0 10/30/2009 $             ****
---------------------------------------------------------

        Copyright 2009, Stanford University

---------------------------------------------------------

 We are grateful if you cite the following reference in your
 publications that benefited from using Omega3P:

 Lie-Quan Lee, Zenghai Li, Cho Ng, and Kwok Ko,
 Omega3P: A Parallel Finite-Element Eigenmode Analysis Code
 for Accelerator Cavities, Tech. Rep., SLAC-PUB-13529, 2009

*****************************************************************

Read Mesh: ./sw1.ncdf
Time for calculating adjacency: 0.09636712074279785
Partitioning Method: parmetis

***************************************************************
*      Total Number of Elements read:  13319
*      Total Number of Elements used:  13319
*      Total Number of DOFs:   82256
***************************************************************

Total Volume of the structure is : 0.00091256004594559 91
No.          Sum     Average       Max     Min     Std_dev
Diagonal:     3318528  3.32e+06      3318528      3318528       0.00e+00
Offdiagonal:   0       0.00e+00      0      0      0.00e+00
Nonlocal v:   0       0.00e+00      0      0      0.00e+00

Number of Grad DOFs: 14903
```

```
*************************************************************
 ARPACK Loop:
Shift = 1.098141589009911e+02
*************************************************************
factorizing the matrix using MUMPS ...
Using ParMETIS for ordering...
Before call METIS ...
After call METIS ...


Analysis step: 0.379155 seconds

      Maximal per-core estimated memory        356 MB
      Aggregated estimated memory       356 MB
      Maximal per-core estimated memory if OOC        104 MB
      Aggregated estimated memory if OOC       104 MB


Factorization step: 21.178623 seconds

      ncv=6   nev=2
 Number of converged eigenpairs = 2
eigenvalue:  6.232943440154033e+02      Frequency:
1.191207523118691e+09      Residual:   1.18e-11
eigenvalue:  1.872153402017007e+03      Frequency:
2.064484137364606e+09      Residual:   1.21e-08
COMMIT MODE: 0 FREQ = 1.191207523118691e+09      k=
2.496586357439701e+01      norm(v[0]) = 2.266668141020167e+01
COMMIT MODE: 1 FREQ = 2.064484137364606e+09      k=
4.326838802193822e+01      norm(v[1]) = 2.107696817608093e+01
Number of TriSolve: 43. Average time for one TriSolv: 0.307922
Computed Total Energy (normalized by Epsilon0/2): 1.000000000000008e+00
Computed Total Energy (normalized by Epsilon0/2): 1.000000000000008e+00
```

# Files In Run Directory

* Input files:
  – sw1.ncdf  sw1.o3p
* Mode files:  (Electric and magnetic fields on mesh nodes)
  – mode.l0.m0000.1.1912075e+09.mod
  – mode.l0.m0001.2.0644841e+09.mod
* Simple log file:
  – output
* Directory: (containing complete eigenvectors information)
  – vector1
  – Can be used in Track3P for further computations, or acdtool for postprocessing

# Postprocessing Using ACDTOOL

- Shunt impedance
- Kick factor
- Surface fields
- Field map
- …

# ACDTOOL usage

- acdtool postprocess rf
  produce a "sample.input" file
  containing inputs for all functions

  Modify and rename it

- acdtool postprocess rf  input.in
  perform calculations specified in
  "input.in"

# Sample.input (see details next)

```
RFField
{
  O3PMode      = vector1
  ModeRangeID1=     -1
  ModeRangeID2=     -1
  ModeID       =     0
  xsymmetry  =     none   // [none, electric, magnetic]
  ysymmetry  =     none   // [none, electric, magnetic]
  gradient   = -1
  cavityBeta  =     1.00000     //for R/Q, V integral
  reversePowerFlow=      0   // [1=charging 0=decaying]
  x0          =     0.00000
  y0          =     0.00000
  gz1         =     -0.057
  gz2         =      0.057
  fx1         = -10000000.00000
  fx2         = -10000000.00000
  fy1         = -10000000.00000
  fy2         = -10000000.00000
  fz1         = -10000000.00000
  fz2         = -10000000.00000
  npoint      =    300
  fmnx        =     10
  fmny        =     10
  fmnz        =     50
}
RoverQ
{
  ionoff      =     1
  modelD1     =     -1
  modelD2     =     -1
  x1          =     0.00000
  x2          =     0.00000
  y1          =     0.00100
  y2          =     0.00100
  z1          = 100000000.00000
  z2          = 100000000.00000
}

RoverQT
{
  ionoff      =     0
  modelD1     =     -1
  modelD2     =     -1
  x0          =     0.00000
  y0          =     0.00000
  z1          = 100000000.00000
  z2          = 100000000.00000
}

kickFactor
{
  ionoff      =     0
  modelD1     =     -1
  modelD2     =     -1
  x0          =     0.00000
  y0          =     0.00000
  z1          = 100000000.00000
  z2          = 100000000.00000
}

FieldMap
{
  ionoff      =     0
  nx          =     20
  ny          =     20
  nz          =     50
  x1          =     0.00000
  x2          =     0.00000
  y1          =     0.00100
  y2          =     0.00100
  z1          = 100000000.00000
  z2          = 100000000.00000
}

FieldAtPoint
{
  ionoff      =     0
  x0          =     0.00000
  y0          =     0.00000
  z0          =     0.00000
}

ALLFieldAtPoint
{
  ionoff      =     0
  modelD1     =     -1
  modelD2     =     -1
  x0          =     0.00000
  y0          =     0.00000
  z0          =     0.00000
}

FieldOnLine
{
  ionoff      =     0
  npoint      =    300
  filename    = filename1
  rfphase     =     0.00000
  x1          =     0.00000
  y1          =     0.00100
  z1          = 100000000.00000
  x2          =     0.00000
  y2          =     0.00100
  z2          = 100000000.00000
}

ALLFieldOnLine
{
  ionoff      =     0
  modelD1     =     -1
  modelD2     =     -1
  npoint      =    300
  filename    = filename1
  rfphase     =     0.00000
  x1          =     0.00000
  y1          =     0.00100
  z1          = 100000000.00000
  x2          =     0.00000
  y2          =     0.00100
  z2          = 100000000.00000
}

fieldOn2DBoundary
{
  ionoff      =     0
  filename    = filename1
  surfaceID   =     6
}

fieldOnSurface
{
  ionoff      =     0
  filename    = filename1
  surfaceID   =     6
  output      = amplitude   // [component, amplitude]
}

maxFieldsOnSurface
{
  ionoff      =     1
  surfaceID   =     6
}
```

# ACDTOOL Postprocess RF Functions

RoverQ
RoverQT
kickFactor

FieldMap
FieldAtPoint
ALLFieldAtPoint

FieldOnLine
ALLFieldOnLine

fieldOn2DBoundary
fieldOnSurface
maxFieldsOnSurface

Multipole field analysis
….

# Sample.input in detail

```
RFField
{
  O3PMode      = vector1
  ModeRangeID1=     -1
  ModeRangeID2=     -1
  ModeID       =      0
  xsymmetry    =      magnetic    // [none, electric, magnetic]
  ysymmetry    =      magnetic    // [none, electric, magnetic]
  gradient     =  -1
  cavityBeta   =      1.00000     //for R/Q, V integral
  reversePowerFlow=       0   // [1=charging 0=decaying]
  x0           =      0.00000
  y0           =      0.00000
  gz1          =     -0.057
  gz2          =      0.057
  fx1          = -10000000.00000
  fx2          = -10000000.00000
  fy1          = -10000000.00000
  fy2          = -10000000.00000
  fz1          = -10000000.00000
  fz2          = -10000000.00000
  npoint       =    300
  fmnx         =     10
  fmny         =     10
  fmnz         =     50
}
```

RFField

Mode information

Always be the first input

# Shunt Impedance & Kickfactor

```
RoverQ
{
  ionoff      =     1
  modeID1     =    -1
  modeID2     =    -1
  x1          =     0.00000
  x2          =     0.00000
  y1          =     0.00100
  y2          =     0.00100
  z1          = 100000000.00000
  z2          = 100000000.00000
}

RoverQT
{
  ionoff      =     0
  modeID1     =    -1
  modeID2     =    -1
  x0          =     0.00000
  y0          =     0.00000
  z1          = 100000000.00000
  z2          = 100000000.00000
}

kickFactor
{
  ionoff      =     0
  modeID1     =    -1
  modeID2     =    -1
  x0          =     0.00000
  y0          =     0.00000
  z1          = 100000000.00000
  z2          = 100000000.00000
}
```

# Field Map

```
FieldMap
{
  ionoff     =    0
  nx         =   20
  ny         =   20
  nz         =   50
  x1         =    0.00000
  x2         =    0.00000
  y1         =    0.00100
  y2         =    0.00100
  z1         = 100000000.00000
  z2         = 100000000.00000
}
```

```
FieldAtPoint
{
  ionoff     =    0
  x0         =    0.00000
  y0         =    0.00000
  z0         =    0.00000
}

ALLFieldAtPoint
{
  ionoff     =    0
  modelD1    =   -1
  modelD2    =   -1
  x0         =    0.00000
  y0         =    0.00000
  z0         =    0.00000
}
```

# Field on a line

```
FieldOnLine
{
  ionoff      =     0
  npoint      =   300
  filename    = filename1
  rfphase     =     0.00000
  x1          =     0.00000
  y1          =     0.00100
  z1          = 100000000.00000
  x2          =     0.00000
  y2          =     0.00100
  z2          = 100000000.00000
}
```

```
ALLFieldOnLine
{
  ionoff      =     0
  modeID1     =    -1
  modeID2     =    -1
  npoint      =   300
  filename    = filename1
  rfphase     =     0.00000
  x1          =     0.00000
  y1          =     0.00100
  z1          = 100000000.00000
  x2          =     0.00000
  y2          =     0.00100
  z2          = 100000000.00000
}
```

# Field on a Surface

```
fieldOn2DBoundary
{
  ionoff      =     0
  filename    = filename1
  surfaceID   =     6
}

maxFieldsOnSurface
{
  ionoff      =     1
  surfaceID   =     6
}

fieldOnSurface
{
  ionoff      =     0
  filename    = filename1
  surfaceID   =     6
  output      = amplitude   // [component,
    amplitude]
}
```

# Output (sample)

```
*************************** Omega3P Mode Summary **********************************
**
** Omega3P Mode Summary:
**
**   Total number of modes :      1
**      ModeID        Frequency(r,i)                        Qext
**         0          1.51823e+10, 4.83426e-06           1.57028e+15
**
**   Model bounding box: ( 0.00000e+00,  0.00000e+00 -5.46716e-03)
**                       ( 1.11055e-02,  1.11055e-02  5.46716e-03)
**
** ------------
**   Mode used:
**      ModeID    =      0
**      Frequency =  1.51823e+10,    4.83426e-06
**
*******************************************************************************

[scaling]
{
  E,B m_factor = 1.00000e+00
}
```

# Output – kick factor & surface field

```
RoverQ
{
  ionoff      =     1
  modeID1     =    -1
  modeID2     =    -1
  x1          =     0.00000
  x2          =     0.00000
  y1          =     0.00100
  y2          =     0.00100
  z1          =    -0.15000
  z2          =     0.15000
}
[RoverQ]
{  // RoverQ=V^2/(omega*U)
  Integral:  x1  = 0.0000e+00,  y1  = 1.0000e-03,  z1  =-1.5000e-01
             x2  = 0.0000e+00,  y2  = 1.0000e-03,  z2  = 1.5000e-01
 ModeID  Frequency        (V_r, V_i)              |V|          RoQ(ohm/cavity)
   0   1.19121e+09  (-3.4990e+00, -2.4096e+00)   4.24845e+00     1.36181e+02
   1   2.06448e+09  (-5.1684e-01,  2.4588e+00)   2.51254e+00     2.74824e+01
}
maxFieldsOnSurface
{
  ionoff      =     1
  surfaceID   =     6
}
[maxFieldsOnSurface]
{
  surfaceID :   6
     Emax :  3.26104e+03 (V.m)      at ( 4.8285e-03,  2.7420e-03, -3.7607e-03)
     Hmax :  3.61287e+01 (A/m)      at ( 1.9729e-03,  4.0967e-03,  5.4672e-03)
}
```

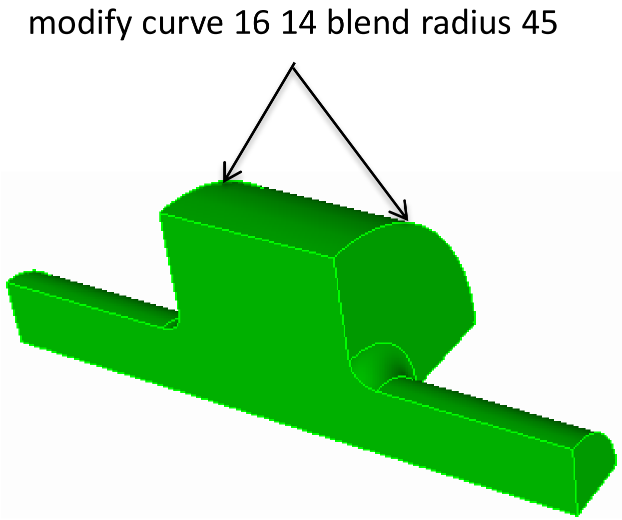
Surface field value corresponds to gradient specified in RFField

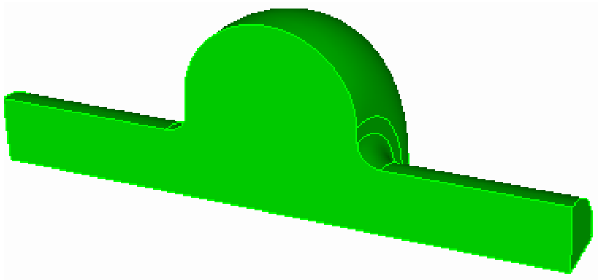
# Omega3P Practice - 1

* Round the pillbox out surface to half circle
  – modify curve 16 14 blend radius 45

* Mesh the model
  – modify the meshing file, surface number may have changed
  – Start with a very coarse mesh

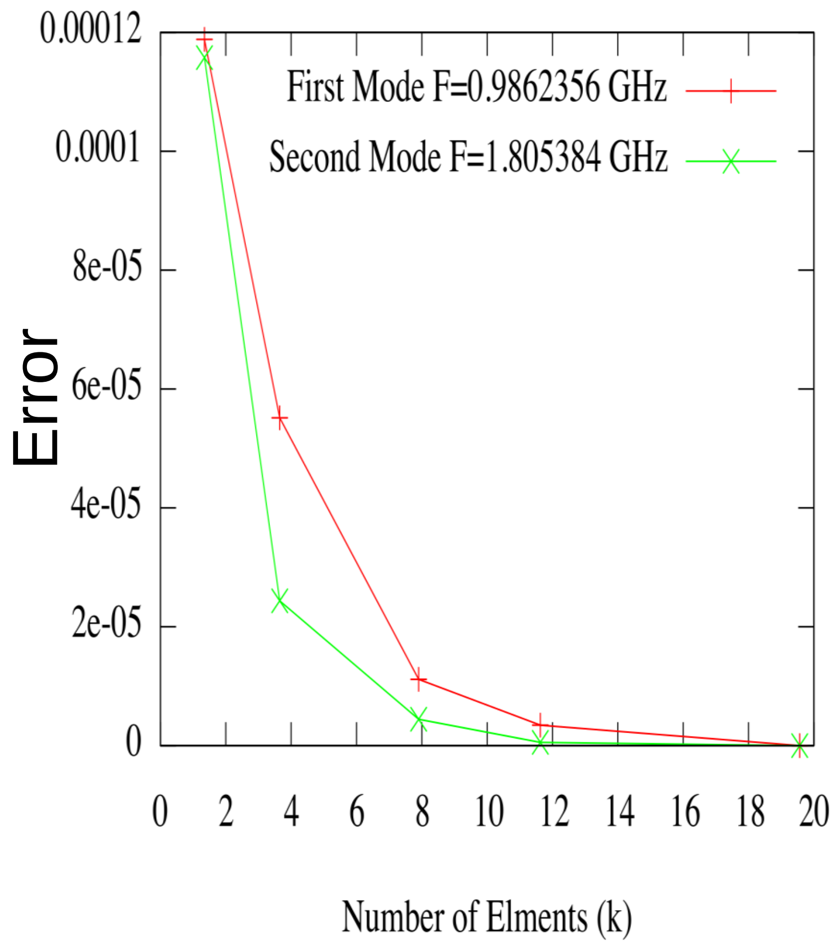
* Refine the mesh to study convergence
  – Just need to do couple of mesh levels
  – Don’t make too large a mesh (cpu, memory limit)



# Convergence Study - example

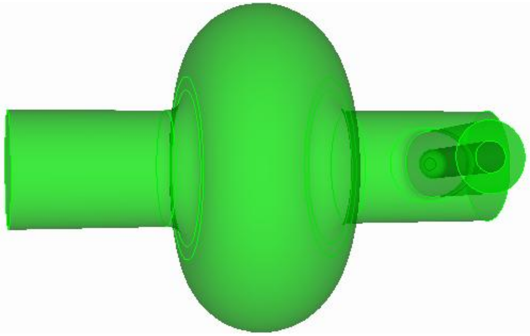
* Generate a set of meshes for convergence study
* Cubit journal files
  – m.jou, m0.jou, m1.jou, m2.jou, m3.jou
* Mesh files
  – m.ncdf, m0.ncdf, m1.ncdf, m2.ncdf, m3.ncdf
  – Number of elements:
  1.3k, 3.7k, 7.9k, 11.6k, 19.6k

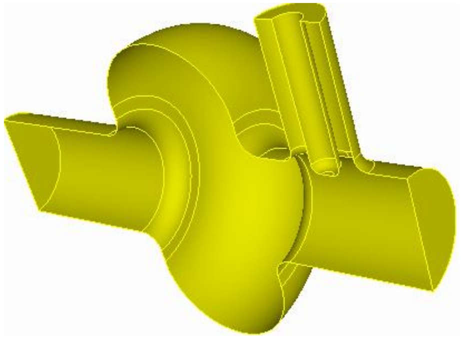


# Omega3P Practice - 2

Add coax coupler to the rounded pillbox cavity

- Coax inner conductor radius: 8.7 mm
- Outer conductor radius: 20.0 mm
- At z location 100 mm
- Inner conductor tip 30 mm from the beam axis
- Round the inner conductor tip with 5 mm radius
- Round the outer conductor and beampipe connection with 7 mm radius





Set ID for the "Port" surface

# Omega3P Practice – 2 Build the model

* Build rtop+coax model DEMO

* Build rtop+coax model

* Mesh with rtop+coax
  – Remember to define sideset for the port surface

* Omega3P input with “port”

* Run omega3p

* Run postprocessing

# Omega3P Input with Port

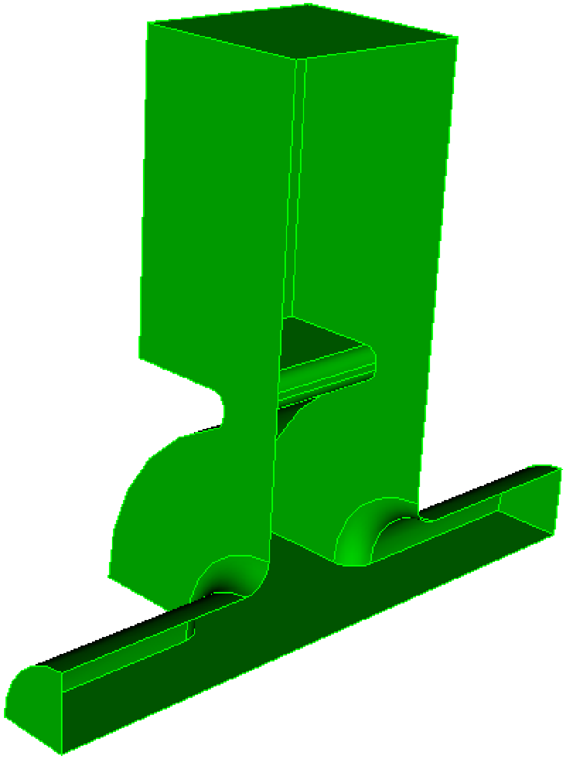
```
CheckPoint: {
  Directory: vector1
  Act:      save
}
ModelInfo : {
 File: ./rtop+coax2.ncdf
 BoundaryCondition : {
   Magnetic:  2
   Electric:  3, 4
   Exterior:  6
   Absorbing: 7
 }
 SurfaceMaterial : {
   ReferenceNumber: 6
   Sigma: 5.8e7
 }
}
FiniteElement: {
 Order:        2
 CurvedSurfaces: on
}
EigenSolver : {
 NumEigenvalues: 1
 FrequencyShift:  1.0e9
}
PostProcess : {
 Toggle: on
 Compute Total Energy: 1
 ModeFile: mode
 SymmetryFactor: 1
}
```

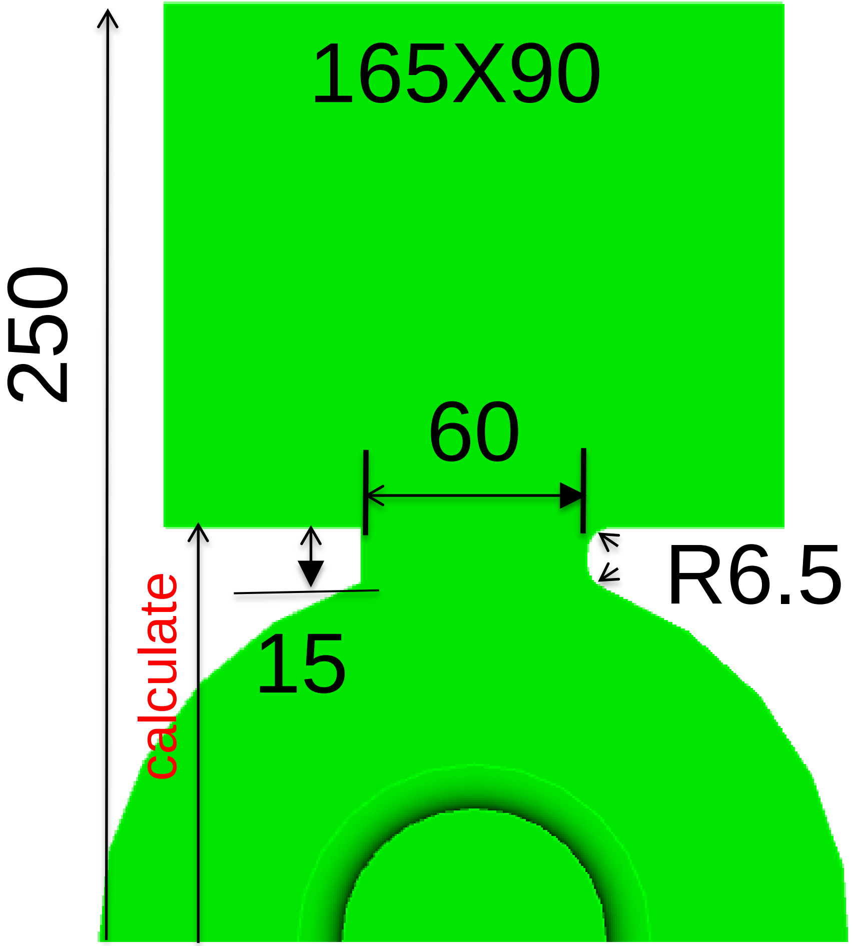
```
Port: {
 ReferenceNumber: 7  // HOM Coupler Port
   Origin: -0.120 , 0.0 , 0.100
   ESolver:
   {
     Type: Analytic
     Mode:
     {
       Mode number: 1
       Waveguide type: Coax
       Mode type: TEM
       A: 0.0087  // A is smaller radius
       B: 0.0200  // B is larger radius
     }
   }
}
```

https://confluence.slac.stanford.edu/display/AdvComp/Omega3P+Sample+Inputs

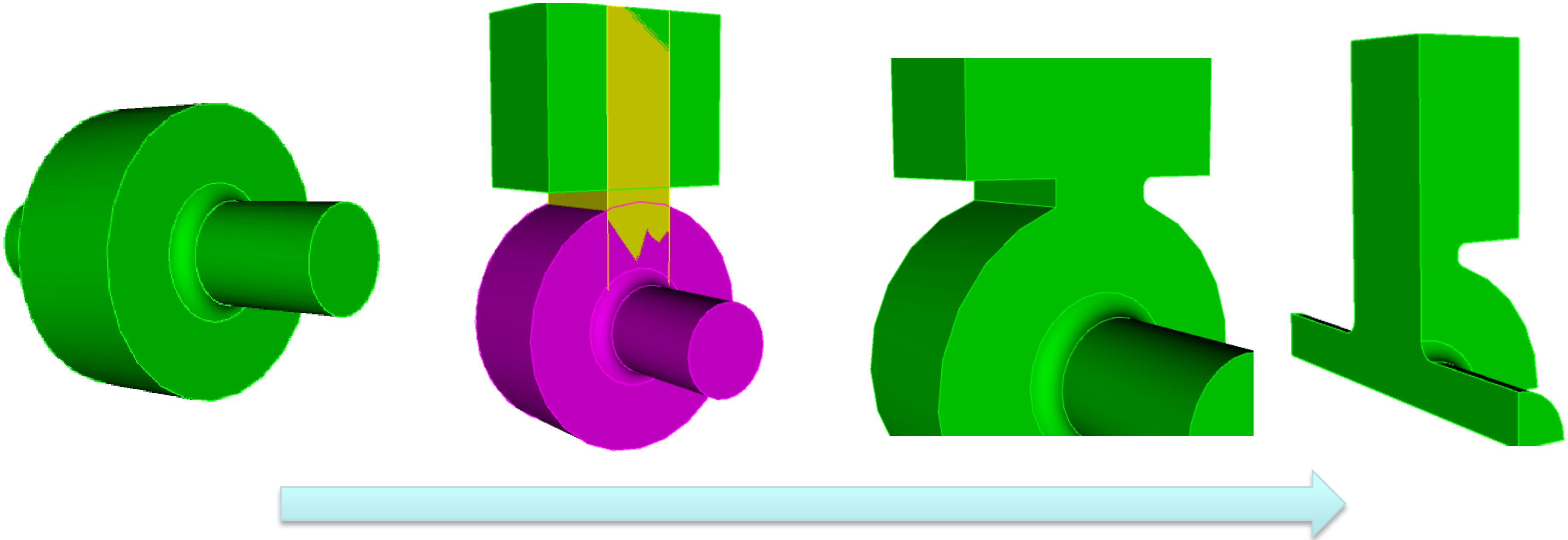
# Omega3P - Practice - 3

* Add rectangular coupler to pillbox cavity





# Model Steps



Pillbox cavity

Add coupler blocks

Unite all Round iris corners

Webcut 1/4 model

# Practice 3 - Step1-Make-sw1.jou

```
reset
#{cav_length = 90}
#{cav_radius = 100}
#{bp_length  = 300}
#{bp_radius  = 35 }
#{iris_rounding = 12}

create Cylinder height {cav_length} radius {cav_radius}
create Cylinder height {bp_length} radius {bp_radius}
unite volume all
compress ids

modify curve 6 5  blend radius {iris_rounding}
export acis "sw0.sat" overwrite
```

# Practice 3 – ADD-rec-to-sw1.jou (1)

```
reset
#{cav_radius = 100}
#{wguide  = 165.0 }
#{hguide  = 90.0 }
#{lguide  = 250}

#{wiris   = 60 }
#{diris = 15}
#{rround  = 6.5 }
#{sth = 0.5*wiris/cav_radius}
#{cth = sqrt(1.0 - sth*sth)}
#{rguide  = cth*cav_radius + diris}

brick x {wguide} y {lguide-rguide} z {hguide}
body 1 move 0 {0.5*(lguide-rguide)+rguide} 0
brick x {wiris} y {lguide} z {hguide}
body 2 move 0 {0.5*lguide} 0
```

# Practice 3 - ADD-rec-to-sw1.jou (2)

```
import acis "sw0.sat”
unite all
compress ids

modify curve 30 17 blend radius {rround}

webcut volume 1 with plane yplane offset 0 noimprint nomerge
delete volume 1
compress ids

webcut volume 1 with plane xplane offset 0 noimprint nomerge
delete volume 2
compress ids

export acis "sw1+rec.sat" overwrite
```

# Practice 3 - Mesh-sw1+rec.jou

```
reset
import acis "sw1+rec.sat"
volume all scheme Tetmesh
volume all sizing function type skeleton scale 7 time_accuracy_level 2 min_size
    auto max_size 12 max_gradient 1.2

mesh volume all
Sideset 1 surface 1
Sideset 2 surface 7
Sideset 3 surface 2
Sideset 4 surface 8
Sideset 7 surface 6
Sideset 6 surface all except 1 7 2 8 6
block 1 volume all
block 1 element type tetra10
volume all scale 0.001
export Genesis  "sw1+rec.gen" block all overwrite
```

# Practice 3 - sw1-rec.o3p (1)

```
CheckPoint: {
  Directory: vector1
  Act:       save
}
ModelInfo : {
 File: ./sw1+rec.ncdf
 BoundaryCondition : {
   Magnetic:  1, 2
   Electric:  3, 4
   Exterior:  6
   Absorbing: 7
 }
 SurfaceMaterial : {
   ReferenceNumber: 6
   Sigma: 5.8e7
 }
}
FiniteElement: {
 Order:         1
 CurvedSurfaces: on
}

EigenSolver : {
 NumEigenvalues: 1
 FrequencyShift:  0.9e9
}
PostProcess : {
 Toggle: on
 Compute Total Energy: 1
 ModeFile: mode
 SymmetryFactor: 4
}
```

# Practice 3 - sw1-rec.o3p (2)

```
Port:
{
    ReferenceNumber: 7
    Origin:     0.0, 0.25, 0.0
    XDirection: -1.0, 0.0, 0.0
    YDirection: 0.0, 0.0, 1.0
    ESolver:
    {
        Type: Analytic
        Mode:
        {
            Mode number: 1
            Waveguide type: Rectangular
            Frequency: 1.e+9
            Mode type: TE, 1, 0
            A: 0.165
            B: 0.090
        }
    }
}
```

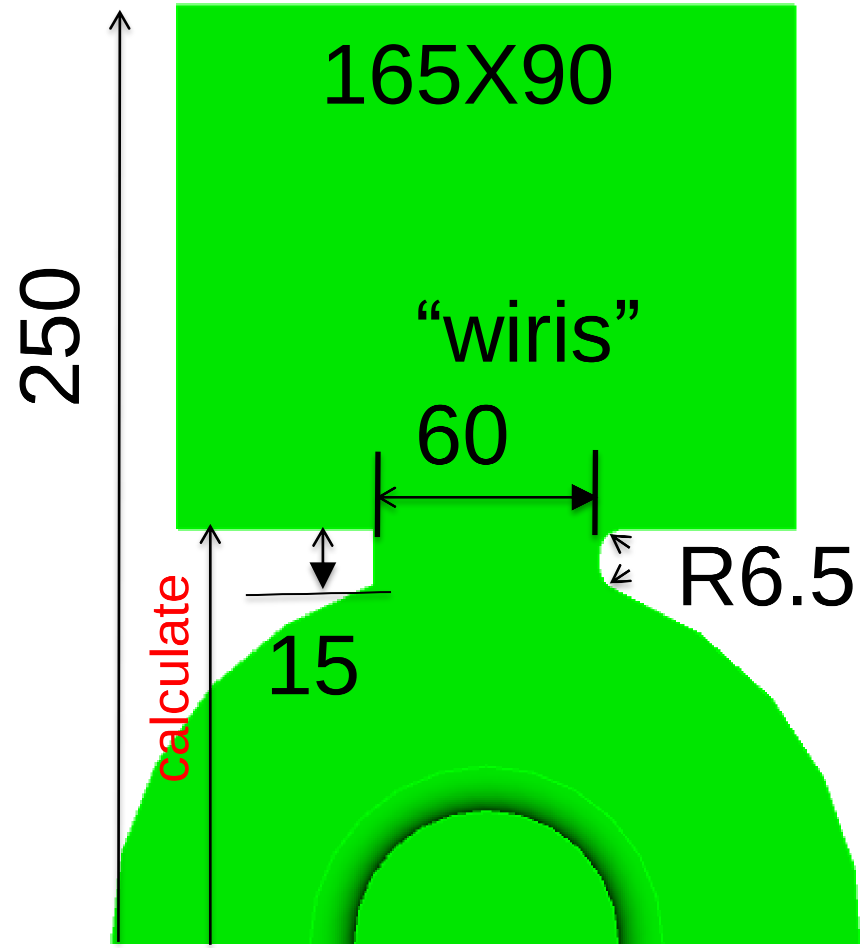
# Practice 3 – Omega3p Results

```
------------------------------------------------------
Iteration: 12
Solving Projected System...
The 0th ritz pair:
     Frequency:        (1160285342.547472,1630154.204100033)
     Residual:         3.392976528129805e-08
     norm(v):          20.61363492852002
     required tolerance:          1e-08
     reached tolerance:           1.645986522947221e-09
=================================================
COMMIT MODE: 0 FREQ = (1160285342.547472,1630154.204100033)          k =
    (24.31778259221621,0.0341655057367666)    Q = 355.8820814709478

Postprocess Mode 0 (1160285342.547472,1630154.204100033).  Q is
    355.8820814709478
Computed Total Energy (normalized by Epsilon0/2): (1.000003127024341,0)
```

# Omega3P - Practice 3 - Hands on

* Change “wiris” in “ADD-rec-to-sw1.jou”
* Regenerate & mesh the coupler model
* Run omega3p to obtain Qext

* Find “wiris” such that Qext ~ 4000



# Omega3P - practice 4 - with load

sw+rec+load directory

Run: Step1-Make-sw1.jou & ADD-rec-to-sw1.jou

to make sw+rec.sat

Run: ADD-load.jou

webcut at y=200 to 2 volumes, one will be the load

# Omega3P - with load - Mesh (1)

“imprint, merge”
The interface surfaces between volumes become ONE single surface

Reset
import acis "sw1+rec+load.sat”

imprint body 1 2
merge body 1 2

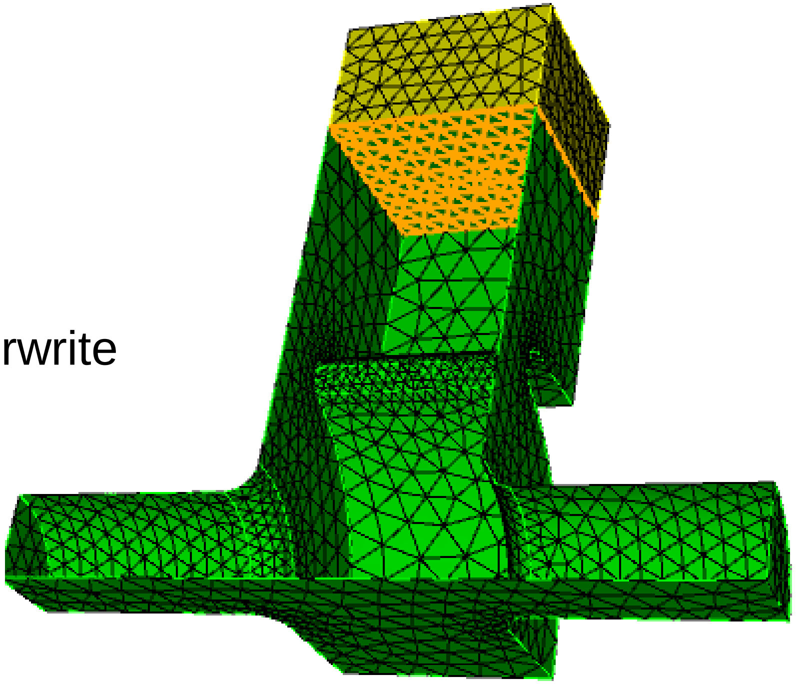
volume all scheme Tetmesh
volume 1 sizing function type skeleton scale 7 time_accuracy_level 2 min_size auto max_size 15 max_gradient 1.2
volume 2 sizing function type skeleton scale 7 time_accuracy_level 2 min_size auto max_size 10 max_gradient 1.2

mesh volume all

# Omega3P - with load - Mesh (2)

Sideset 1 surface 4 20
Sideset 2 surface 9
Sideset 6 surface all except 4 20 9 1

block 1 volume 1
block 2 volume 2
block all element type tetra10
volume all scale 0.001
export Genesis  "sw1+rec+load.gen" block all overwrite



Check if the interface surface a single surface

# Omega3P - with load - Input

```
CheckPoint: {
  Directory: vector1
  Act:       save
}
ModelInfo : {
 File: ./sw1+rec+load.ncdf
 BoundaryCondition : {
   Magnetic:  1, 2
   Exterior:  6
 }
  Material : {
   Attribute: 1
   Epsilon:   1.0
   Mu:        1.0
  }
  Material : {
   Attribute: 2
   Epsilon:   10.0
   EpsilonImag: -1.52
   Mu:        1.0
  }
 SurfaceMaterial : {
   ReferenceNumber: 6
   Sigma: 5.8e7
 }
}
FiniteElement: {
 Order:          1
 CurvedSurfaces: on
}
EigenSolver : {
 NumEigenvalues: 2
 FrequencyShift:  1.1e9
}
PostProcess : {
 Toggle: on
 Compute Total Energy: 1
 ModeFile: mode
 SymmetryFactor: 4
}
```

# Omega3P - with load - result

Number of converged eigenpairs = 2
eigenvalue: (591.1599013854861,0.1189265487860311)
Frequency: (1160094374.35769,116690.9481590217) Residual: 6.162633507886794e-12
eigenvalue: (601.9239405107943,90.58412207268904)
Frequency: (1173899163.930948,87836082.86096708) Residual: 1.890395558726409e-10

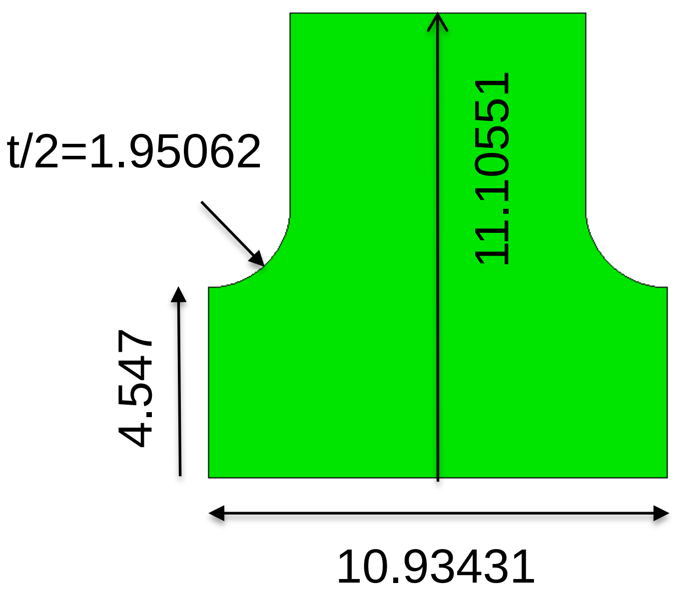
COMMIT MODE: 0 FREQ = (1160094374.35769,116690.9481590217)
k = (24.31378019491716,0.002445661428059073) Q = 4970.798475202896
COMMIT MODE: 1 FREQ = (1173899163.930948,87836082.86096708)
k = (24.60310718997917,1.840908170118933) Q = 6.682328752006592

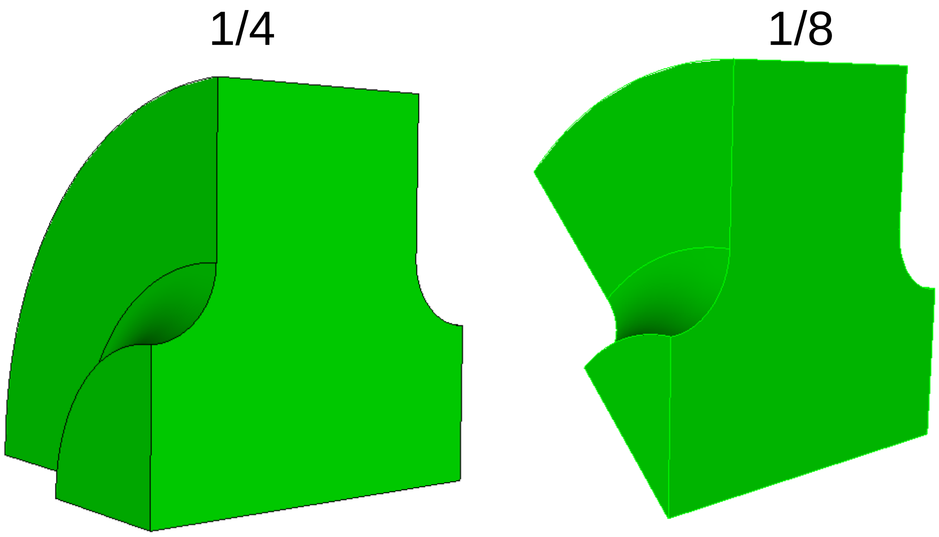
Postprocess Mode 0 (1160094374.35769,116690.9481590217). Q is 4970.798475202896
Computed Total Energy (normalized by Epsilon0/2): (0.9999540206050256,-0.0001005828364993335).
Q from energy: 4970.798475203329

Postprocess Mode 1 (1173899163.930948,87836082.86096708). Q is 6.682328752006592
Computed Total Energy (normalized by Epsilon0/2): (1.083167550785069,-0.08104716117564639).
Q from energy: 6.682328752006592

# Omega3P - practice 5: PBC

## Disk loaded waveguide structure



# Omega3P – PBC - MESH

reset

import acis "ds-cell-1fourth.sat"

volume 1 scheme tetmesh

volume 1 size 2.0

surface 5 6 size 1.

surface 9   size 1.

mesh surface 2

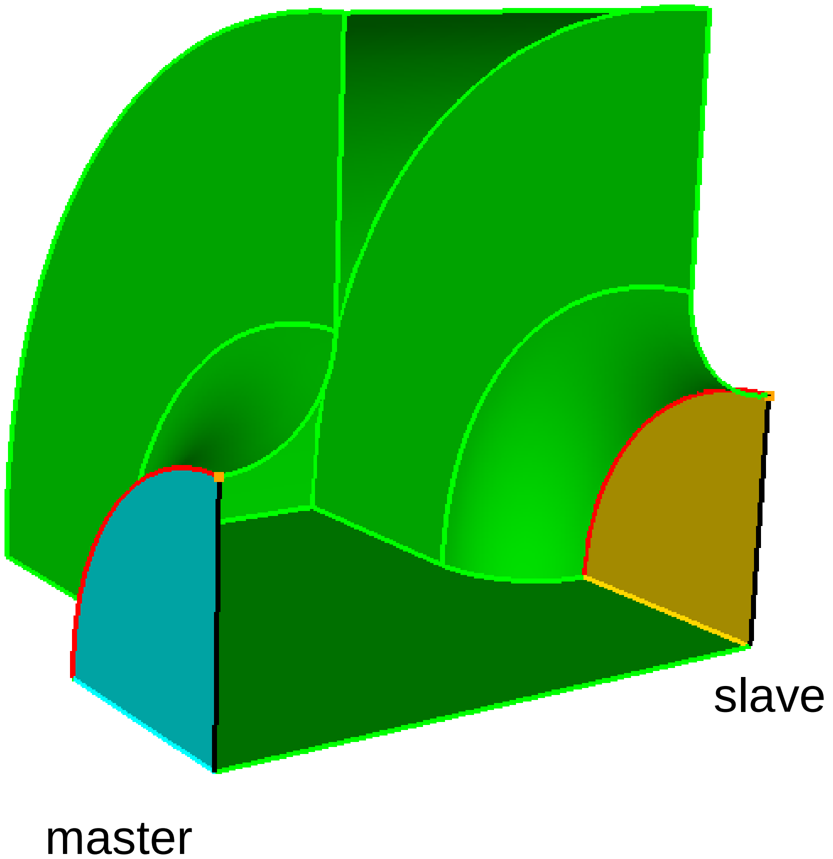
copy Mesh Surface 2 Onto Surface 4 source curve 10 target curve 17 source curve 9 target curve 11 source vertex 9 target vertex 11 Nosmoothing

mesh volume 1

##smooth vol 1

##smooth surface all except 5 7

# Omega3P – PBC - MESH




Mesh “master” first, then copy mesh to “slave”
(1 surface, 1 or 2 curves, 1 vertex)

# Omega3P – PBC - input

```
CheckPoint: {
  Directory: vector1
  Act:      save
}
ModelInfo : {
 File: ../pbc-4.ncdf
 BoundaryCondition : {
   HFormulation: 0
   Exterior: 6
   Magnetic: 2,1
   Periodic_M: 3
   Periodic_S: 4
   Theta: -150
 }
 SurfaceMaterial: {
   ReferenceNumber: 6
   Sigma: 5.8e7
 }
}
```

```
FiniteElement: {
 Order:        2
 CurvedSurfaces: on
}

EigenSolver : {
 NumEigenvalues: 1
 FrequencyShift:  1.10e9
}

PostProcess : {
 Toggle: on
 Compute Total Energy: 1
 ModeFile: mode
 SymmetryFactor: 1
}
```

# Time to run all examples

Run omega3P practice examples

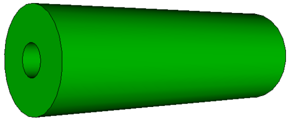
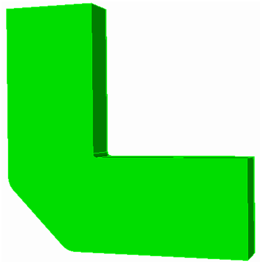
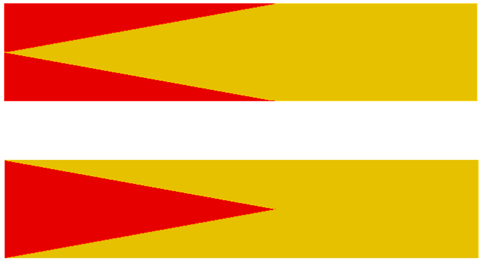
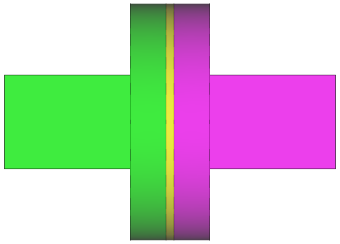
Mesh convergence (do not make too dense)

Acdtool postprocessing

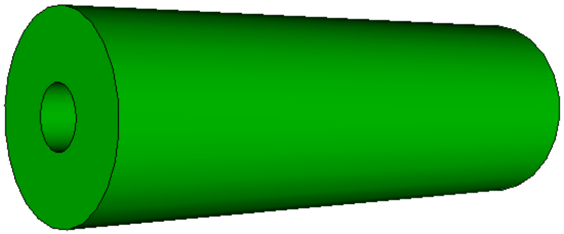
Paraview visualization

# S3P Examples

- Coax
- 90 degree bend
- Waveguide load1
- Waveguide load2
- Window

# S3P - Coax - model



```
reset
#{rin  =  6.26 }
#{rout = 20.0  }
#{wlengthz = 120}
create cylinder height {wlengthz} radius {rout}
create cylinder height {wlengthz} radius {rin}
subtract body 2 from 1
compress ids
export acis "coax.sat" overwrite
webcut volume all with plane xplane offset 0
delete volume 2
webcut volume all with plane yplane offset 0
delete volume 1
compress ids
export acis "coax4.sat" overwrite
```

# S3P - Coax - Mesh

```
reset

import acis "coax4.sat"
volume all scheme tetmesh
volume 1 sizing function type skeleton scale 5
      time_accuracy_level 2 min_size auto
      max_size 6 max_gradient 1.2

mesh volume all
smooth vol all
smooth surface all

group "port7" add surface 3
group "port8" add surface 6
group "sym1"  add surface 4
group "sym2"  add surface 1
group "ss6" add surface all
```

```
ss6 subtract sym1   from ss6
ss6 subtract sym2   from ss6
ss6 subtract port7  from ss6
ss6 subtract port8  from ss6

sideset 1 surface in sym1
sideset 2 surface in sym2
sideset 7 surface in port7
sideset 8 surface in port8
sideset 6 surface in ss6

block 1 vol all

block all element type tetra10
transform mesh output scale 0.001

export genesis "coax4.gen" overwrite
```

# S3P - Coax - S3P input

```
ModelInfo: {
 File: ../coax4.ncdf
 BoundaryCondition: {
  HFormulation: 0
  Magnetic: 1, 2
  Exterior: 6
  Waveguide: 7, 8
 }
 Material : {
  Attribute: 1
  Epsilon:   1.0
  Mu:        1.0
 }
 SurfaceMaterial: {
  ReferenceNumber: 6
  Sigma: 5.8e7
 }
}
FiniteElement: {
 Order: 1
 CurvedSurfaces: on
}

VerifyLinearSolver: yes
```

```
LinearSolver: {
  Solver:         SuperLU
}
WaveguideFrequency: 1.3e+9
/*
FrequencyScan: {  //enable frequency scan of S
      parameter
   Start: 2.456e+9
   End:   3.056e+9
   Interval: 0.02e+9
}
*/

PostProcess : {
 Toggle: on
 Compute Total Energy: 1
 ModeFile: mode
}

CheckPoint: {
 Action: save
 Directory: vector1
}
```

# S3P - Coax - S3P Input (2)

```
Port:
{
    ReferenceNumber: 7
    Origin:     0.0, 0.00, -0.060
    XDirection: -1.0, 0.0, 0.0
    YDirection: 0.0, 1.0, 0.0
    ESolver:
    {
        Type: Analytic
        Mode:
        {
            WaveguideType: Coax
            ModeType: TEM
            A: 0.00625  //smaller radius
            B: 0.020  //larger radius
        }
    }
}
```

```
Port:
{
    ReferenceNumber: 8
    Origin:     0.0, 0.00, 0.060
    XDirection: 1.0, 0.0, 0.0
    YDirection: 0.0, 1.0, 0.0
    ESolver:
    {
        Type: Analytic
        Mode:
        {
            WaveguideType: Coax
            ModeType: TEM
            A: 0.00625  //smaller radius
            B: 0.020  //larger radius
        }
    }
}
```

# S3P - Coax - result

PortRef7.out

PortRef8.out

mode.m0.1.300000E+09

mode.m1.1.300000E+09

output

s3p.in

vector1/

SParameter.out

eigens.out

postprocess.in

s3p.R1.30000000E+09

vector1/SParameter.out

Frequency: 1300000000

NPortMode: 2

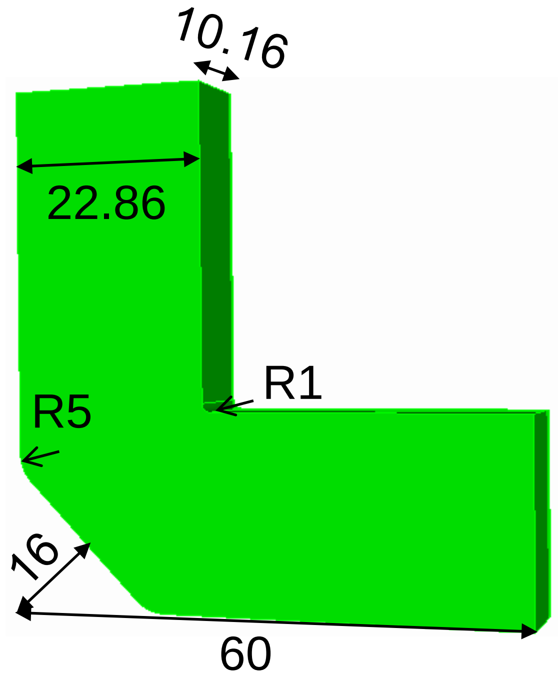
(-6.036907691853166e-06,-0.0007652805304632082) (-0.9911752841378469,0.1325555371289022)

(-0.9911752841378392,0.1325555371289013) (-0.0001952693057193922,-0.0007399734000657077)

A(0,0)=0.0007653043411352959 A(0,1)=0.9999997071545937

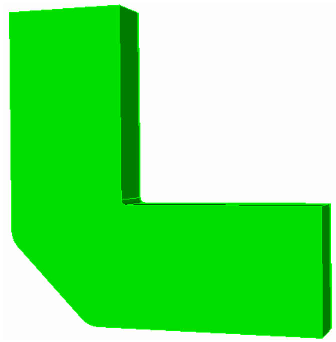
A(1,0)=0.999999707154586 A(1,1)=0.0007653043411355625

# S3P Example – 90-degree Bend

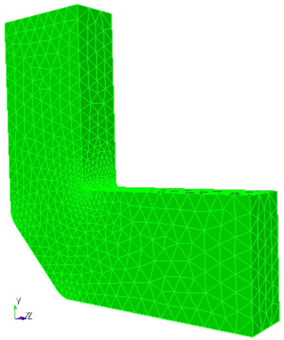


# S3P Example

## X-Band mitrered bend
## Waveguide dimension (WR90): 22.86X10.16



1) model

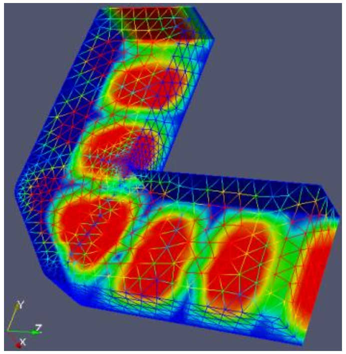


2) mesh

Run S3P

```
Frequency: 11424000000
NPortMode: 2
(-0.0917076674973023,0.0668223302828 1259) (0.5850967326943307,0.8029858612005931)
(0.5850967326944134,0.8029858612007068) (-0.0917071101119568 7,0.0668230952376 0815)
              A(0,0)=0.113470349000173              A(0,1)=0.993541383082 5767
              A(1,0)=0.993541383082 7172             A(1,1)=0.1134703490001729
```

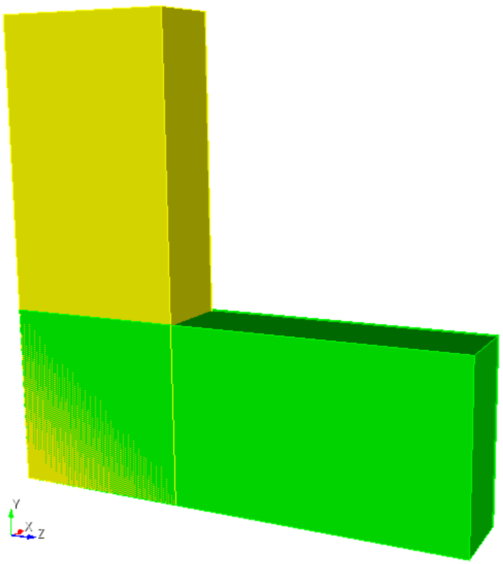
3) Solver



4) postprocess

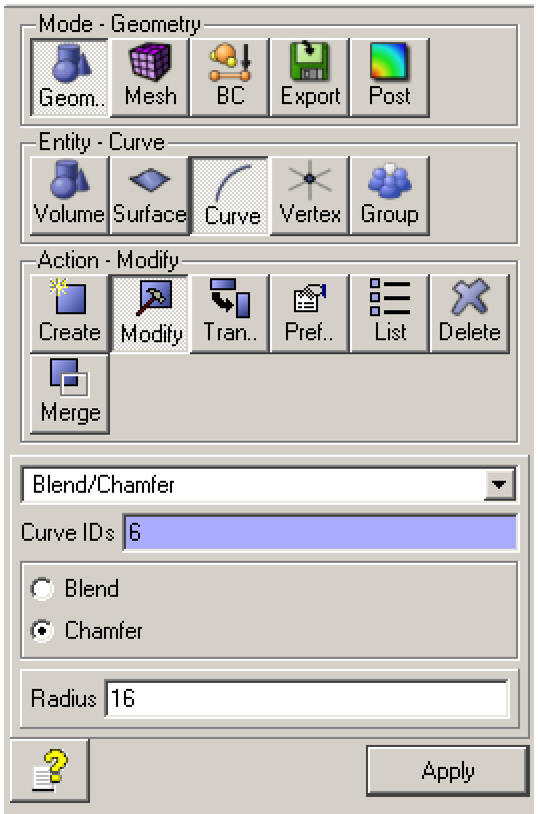
# Create The Model

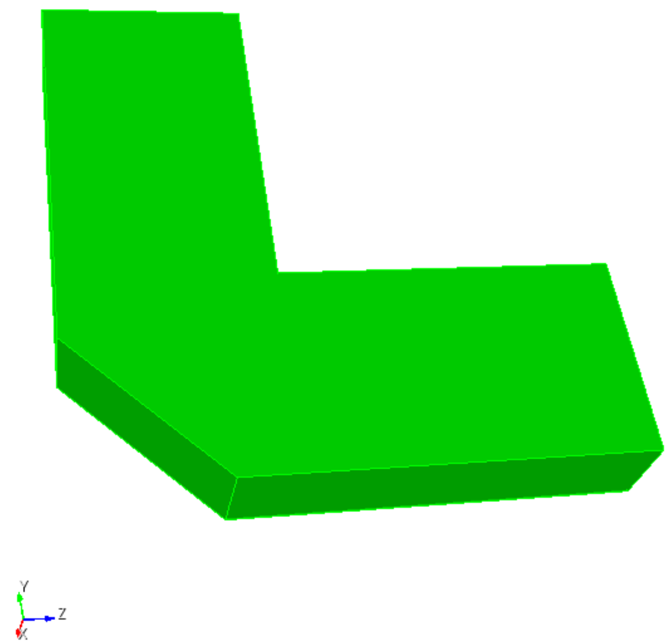
1. Create “brick”: 10.16 X 22.86 X 60
2. Move (dx,dy,dz): 0, 0, {0.5*60-0.5*22.86}
3. Create “brick”: 10.16 X 60 X 22.86
4. Move (dx,dy,dz): 0, {0.5*60-0.5*22.86},0



# Now Add Chamfer and Rounding

1. Unite all (remember to "compress ids"
2. Cut corner (chamfer) with "radius" 16



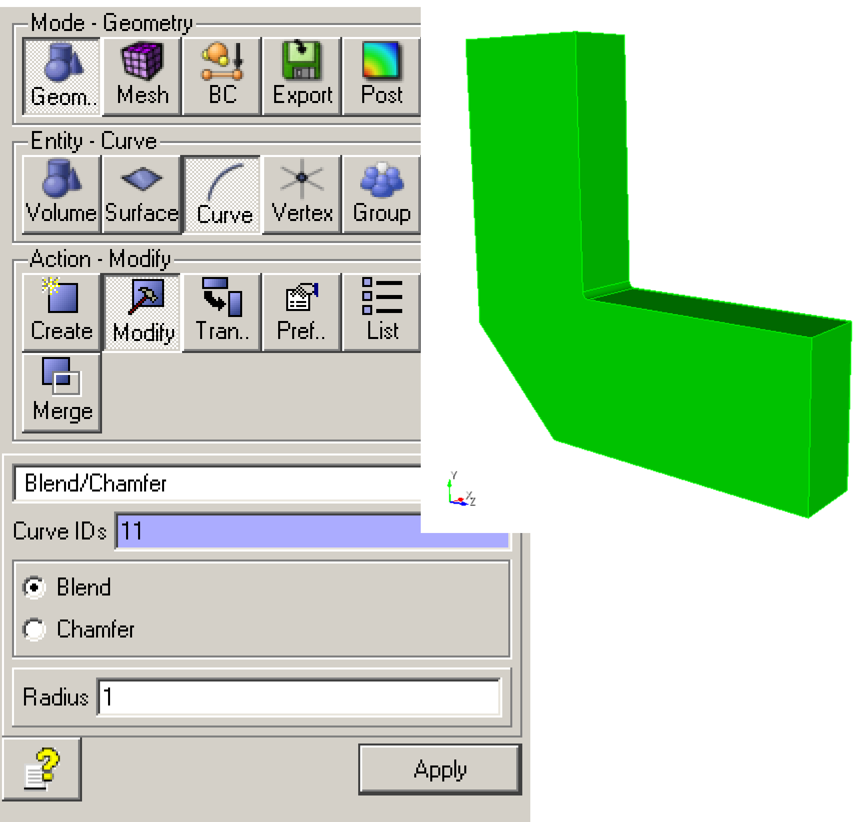



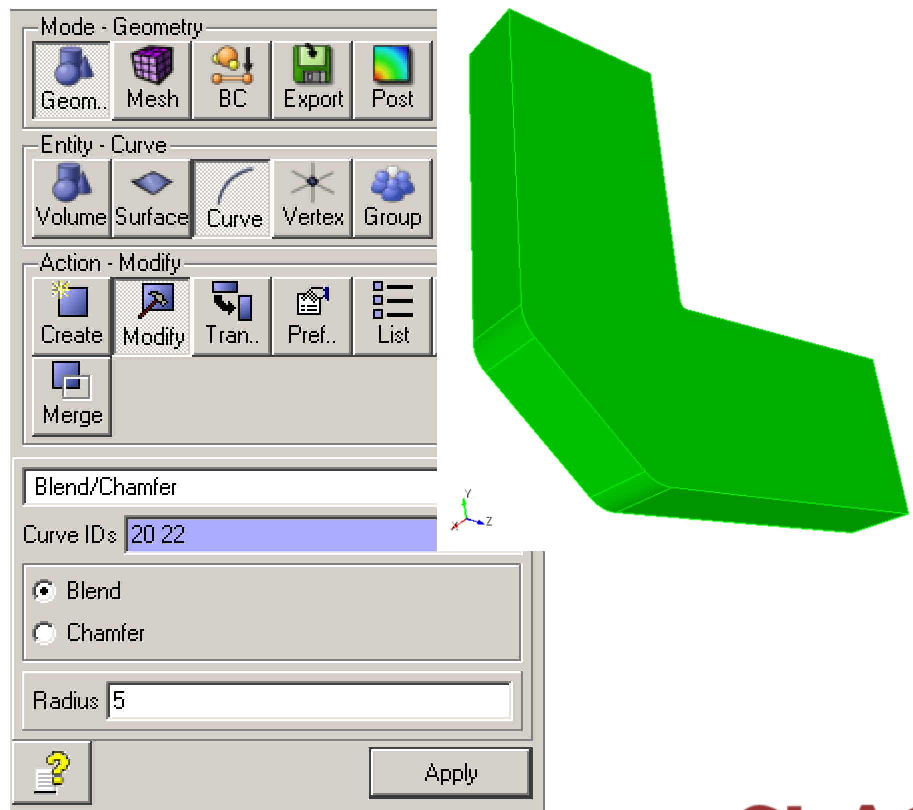


SLAC NATIONAL ACCELERATOR LABORATORY

# Add “Tool” Roundings

* Blend “sharp corner” with radius 1
* Blend “chamfered corners” with radius 5
* Then “export” the model



# JOU File to Create Bend Model

```
reset

brick x 10.16 y 22.86 z 60
move volume 1 location x 0 y 0 z {0.5*60-0.5*22.86}
brick x 10.16 y 60 z 22.86
move volume 2 location x 0 y {0.5*60-0.5*22.86} z 0
unite all
compress ids

modify curve 6  chamfer radius 16
modify curve 11  blend radius 1
modify curve 20 22  blend radius 5

export acis "bend.sat" overwrite
```

# Parameterized JOU File

```
reset
#{wgwidth = 22.86}
#{wghight = 10.16}
#{wglength = 60.0}
#{cornercut = 16}
#{rcorner1 = 1}
#{rcorner2 = 5}
brick x {wghight} y {wgwidth} z {wglength}
move volume 1 location 0 0 {0.5*wglength-0.5*wgwidth}
brick x {wghight} y {wglength} z {wgwidth}
move volume 2 location 0 {0.5*wglength-0.5*wgwidth} 0
unite all
compress ids
modify curve 6  chamfer radius {cornercut}
modify curve 11  blend radius {rcorner1}
modify curve 20 22  blend radius {rcorner2}
export acis "bend.sat" overwrite
```

# Modify "Mesh-sw1.jou" to Mesh the Bend

* reset

* undo group begin
* set attribute on
* import acis "model1.sat"
* separate body all
* set attribute off
* undo group end
* volume all scheme Tetmesh
* volume all sizing function type skeleton scale 5 time_accuracy_level 2 min_size auto max_size 12 max_gradient 1.2

* mesh volume all

* Sideset 1 surface 3
* Sideset 2 surface 1
* Sideset 6 surface all except 3 1

* block 1 volume all
* block 1 element type tetra10
* volume all scale 0.001
* undo group begin
* set large exodus file off
* export Genesis "model1.gen" block all overwrite
* undo group end

1 → "bend.sat"

2 → "max_size 3"
(xband geom – smaller)

3 →
Sideset 7 surface 10
Sideset 8 surface 12
Sideset 6 surface all except 10 12

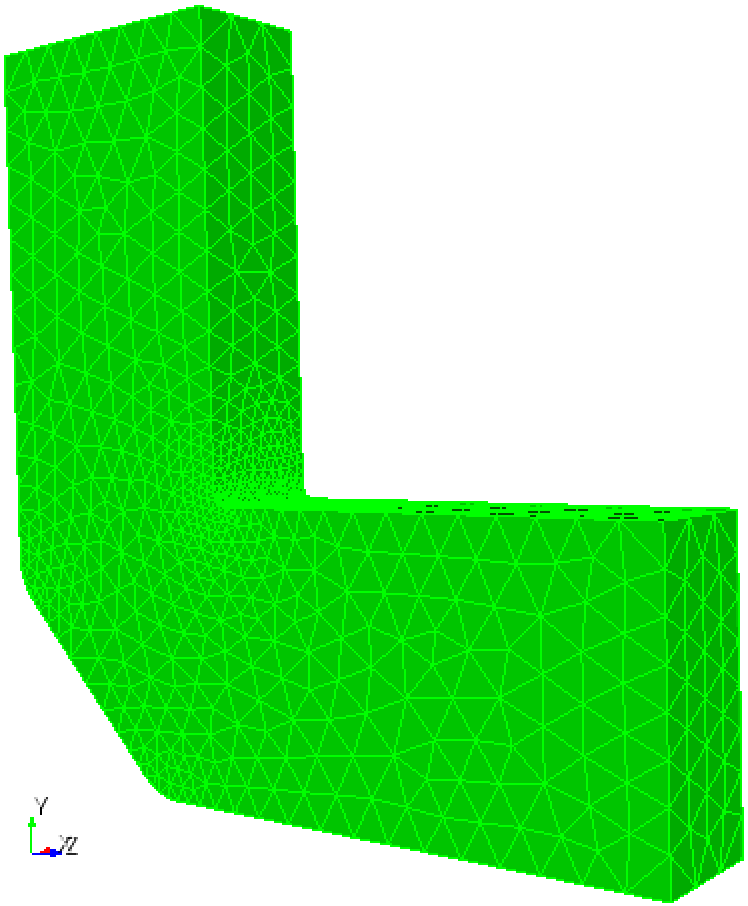
(I normally use 7,8 as ports, 6 as exterior)

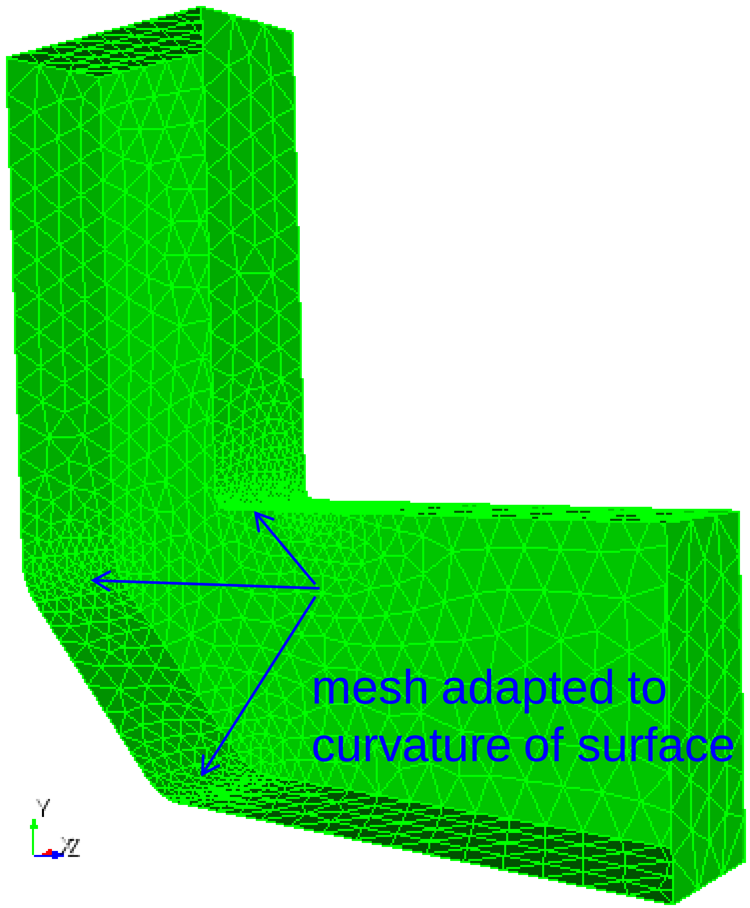
4 → "bend.gen"

5
Save the "jou" file as "Mesh-bend90.jou"
"Play" the jou in CUBIT to generate the mesh

# The Mesh

Front surface visibility off



# Input

```
ModelInfo: {
 File: ./bend.ncdf
 BoundaryCondition: {
  Exterior: 6
  Waveguide: 7, 8
 }
}

FiniteElement: {
 Order:        2
 CurvedSurfaces: on
}

FrequencyScan: {  //enable frequency scan of S
     parameter
  Start: 11.0e+9
  End:   11.8e+9
  Interval: 0.1e+9
}
#WaveguideFrequency: 11.424e+9

PostProcess: {
 Toggle: on
 PortNumber: 0 //input port
 ModeFile: coupla
}
```

```
Port: {
   ReferenceNumber: 7
   Origin: 0.0, 0.04875, 0.0
   XDirection: 0.0, 0.0, 1.0
   YDirection: 1.0, 0.0, 0.0
  ESolver: {
   Type: Analytic
   Mode: {
    WaveguideType: Rectangular
    ModeType: TE 1, 0
    A: 0.02286
    B: 0.01016
   } } }
Port: {
   ReferenceNumber: 8
   Origin: 0.0, 0.0, 0.04875
   XDirection: 0.0, -1.0, 0.0
   YDirection: 1.0, 0.0, 0.0
  ESolver: {
   Type: Analytic
   Mode: {
    WaveguideType: Rectangular
    ModeType: TE 1, 0
    A: 0.02286
    B: 0.01016
   } } }
```

# Input

```
ModelInfo: {
  File: ./bend.ncdf
  BoundaryCondition: {
    Exterior: 6
    Waveguide: 7, 8
  }
}
```

Two ports: reference 7 and 8

# Input

```
FrequencyScan: {  //enable frequency scan of S parameter
  Start: 11.0e+9
  End:   11.8e+9
  Interval: 0.1e+9
}

//WaveguideFrequency: 11.424e+9
//single frequency S parameter computation
```

# Input: Port Information

```
Port: {
    ReferenceNumber: 7                  //cubit sideset ID
    Origin: 0.0, 0.04875, 0.0          //Origin of the 2D port
    XDirection: 0.0, 0.0, 1.0          //the X direction of the port
    YDirection: 1.0, 0.0, 0.0         //the Y direction
  ESolver: {
    Type: Analytic                     //can be Interpolative or
    Numeric
     Mode: {
      WaveguideType: Rectangular //waveguide type
      ModeType: TE 1, 0                 // mode
      A: 0.02286                       //dimension
      B: 0.01016                       //dimension
    }
  }
}
```

# Input

```
PostProcess: {
  Toggle: on
  PortNumber: 0 //the first port is the input port
  ModeFile: coupla
}
```

* Specify the input port and compute the field distributions.
* If not specified, each port will take turn to be the input port and compute field distributions.

# Different Options in Input File

* https://confluence.slac.stanford.edu/display/AdvComp/Omega3P+Sample+Inputs

* https://confluence.slac.stanford.edu/display/AdvComp/S3P+Sample+Inputs
  - Impedance boundary conditions
  - Absorbing boundary conditions
  - Interpolative port modes
  - …

* https://confluence.slac.stanford.edu/display/AdvComp/T3P+Sample+Inputs

# Files In Run Directory

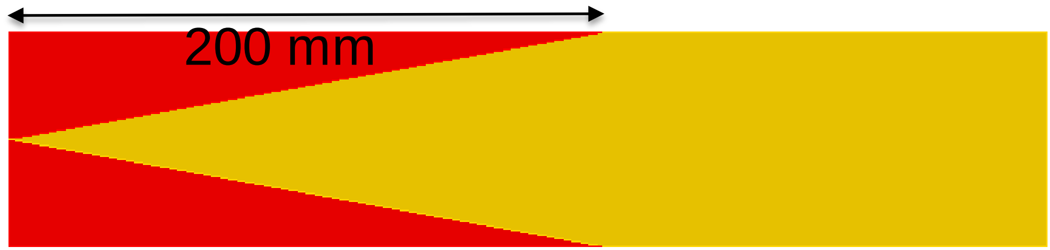
| Filenames | Description |
|---|---|
| PortRef7.out<br>PortRef8.out | Port mode, can be used to examine the waveguide modes |
| bend.ncdf<br>bend.s3p | Input files |
| coupla.m0.1.100000E+10<br>coupla.m0.1.110000E+10<br>coupla.m0.1.120000E+10<br>coupla.m0.1.130000E+10<br>coupla.m0.1.140000E+10<br>coupla.m0.1.150000E+10<br>coupla.m0.1.160000E+10<br>coupla.m0.1.170000E+10<br>coupla.m0.1.180000E+10 | Fields files. coupla is the prefix specified in the input<br>m0 means for the first input port |
| output | Log file |
| vector1 | Directory – field & S-Parameter |

# S Parameter

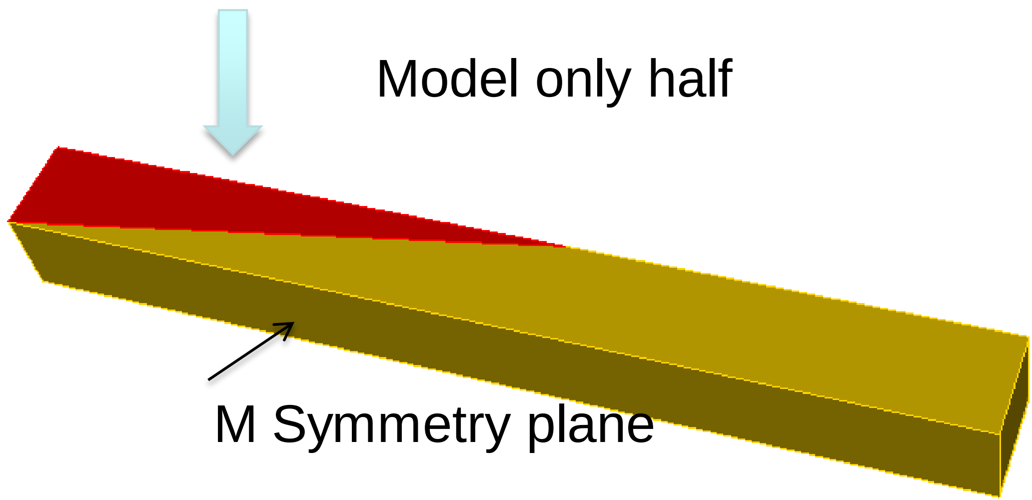
* Saved in vector1/SParameter.out

Frequency: 11000000000
NPortMode: 2
(0.01073444281419905,0.001778045801562309) (0.1670928370598731,-0.9858811257419255)
(0.167092837059825,-0.985881125741642) (0.01072083760495635,0.001858319254573582)
A(0,0)=0.01088070353441187 A(0,1)=0.9999408033933227
A(1,0)=0.9999408033930351 A(1,1)=0.01088070353441197
Frequency: 11100000000
NPortMode: 2
(0.007396890816952828,-0.0006793697559622294) (-0.08576812899370666,-0.9962874346853471)
(-0.08576812899370544,-0.996287434685333) (0.007404169513243131,-0.0005948200084051058)
A(0,0)=0.007428023762969344 A(0,1)=0.9999724118509424
A(1,0)=0.9999724118509282 A(1,1)=0.007428023762969398

# S3P - Load



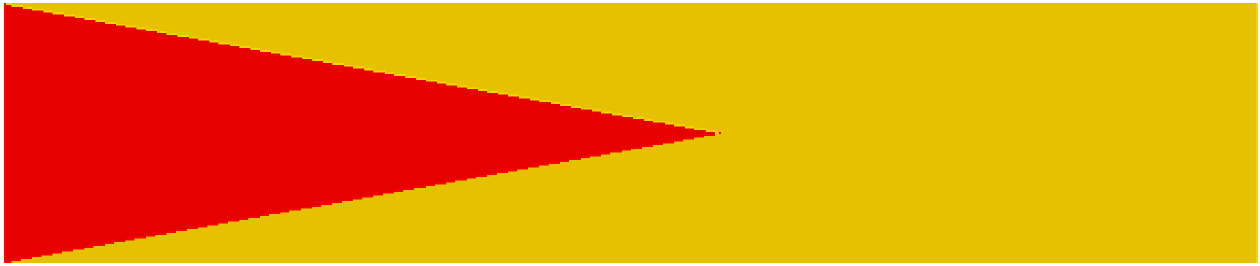

WR284
72.136X34.036

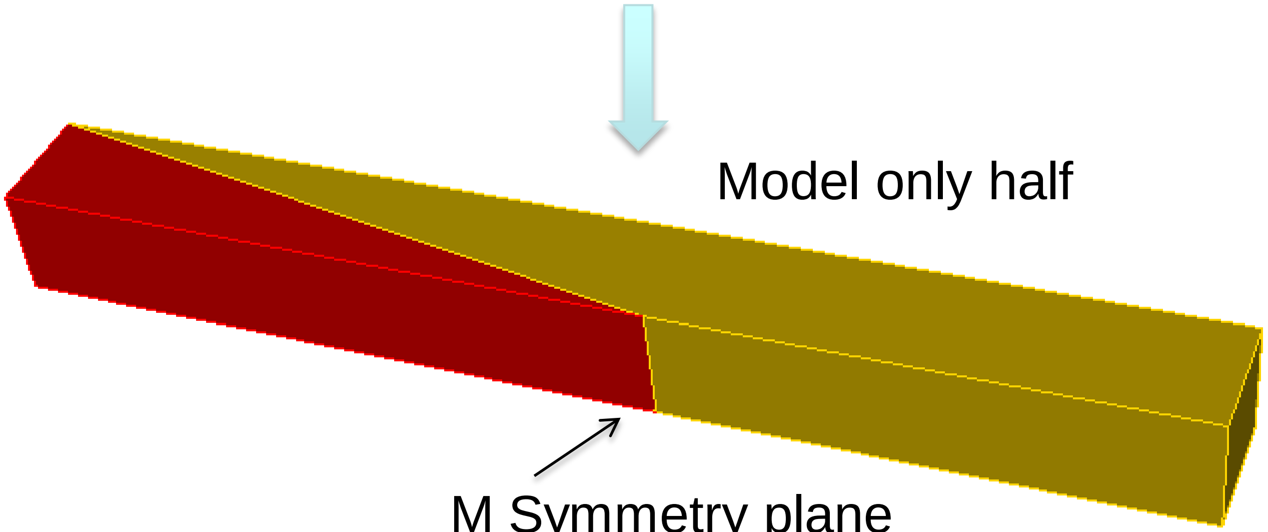


# Mesh with multi-blocks

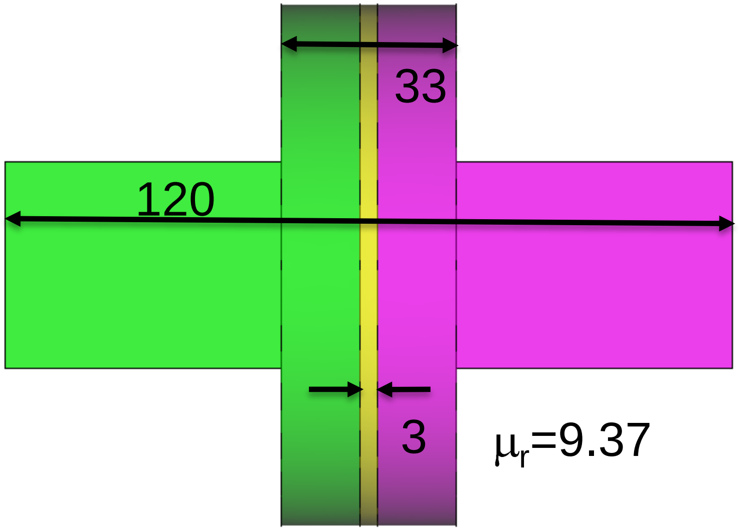
```
reset
import acis "rec-load.sat"

imprint body 1 2
merge body 1 2

volume  all scheme tetmesh
surface all scheme triadvance

volume 1 size 8.0
volume 2 size 8.0

## interface surfaces
surface 1 size 8.0
surface 1 interval hard

mesh volume 1
mesh volume 2
smooth vol all
smooth surface all

sideset 1 surface 11
sideset 7 surface 6
sideset 6 surface all except  11 6 1

block 1 vol 2
block 2 vol 1
block all element type tetra10
transform mesh output scale 0.001
export genesis "recload.gen" overwrite
```

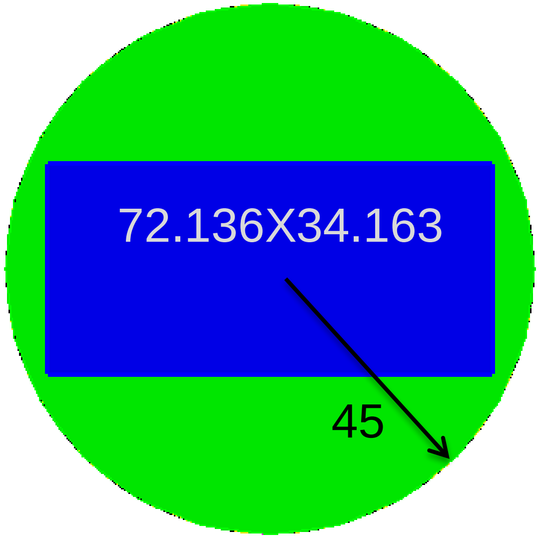
# S3P - Load2





# S3P - Window



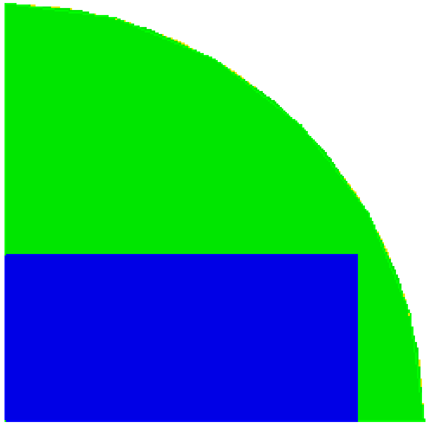

vacuum block

```
Material : {
  Attribute: 1
  Epsilon:   1.0
  Mu:        1.0
}
```

window block

```
Material : {
  Attribute: 2
  Epsilon:   9.37
  Mu:        1.0
}
```

Model ¼ geometry



# S3P – Window - Model

```
reset

#{w284=72.136 }
#{h284=34.163 }
#{cradius = 45 }
#{wdwt    =  3 }
#{cavl    = 15 }
#{wlengthz = 60}
```

```
## ceremic window
brick x {w284} y {h284} z {2*wlengthz}
create cylinder height {2*cavl+wdwt} radius {cradius}
unite all
compress ids
webcut volume 1  with plane zplane offset {0.5*wdwt}
webcut volume 2  with plane zplane offset {-0.5*wdwt}

webcut volume all with plane xplane offset 0
delete volume 6 4 5
webcut volume all with plane yplane offset 0
delete volume 1 2 3
compress ids

export acis "win.sat" overwrite
```

# S3P – Window – Mesh (1)

```
reset
import acis "win.sat”
volume all scheme tetmesh

merge tolerance 0.0001
imprint body all
merge body all

volume 1 3 sizing function type skeleton scale 5 time_accuracy_level 2
    min_size auto max_size 6 max_gradient 1.2
volume 2   sizing function type skeleton scale 3 time_accuracy_level 2
    min_size 1.5 max_size 3 max_gradient 1.2

group 'ms' merged surface
ms expand size 2
ms expand interval hard
```

# S3P – Window – Mesh (2)

```
mesh volume all
smooth vol all
smooth surface all

group "port7" add surface 16
group "port8" add surface 2
group "sym1"  add surface 5 11 17
group "sym2"  add surface 1 9 14

group "ss6" add surface all

ss6 subtract ms     from ss6
ss6 subtract sym1   from ss6
ss6 subtract sym2   from ss6
ss6 subtract port7  from ss6
ss6 subtract port8  from ss6
```

```
sideset 1 surface in sym1
sideset 2 surface in sym2
sideset 7 surface in port7
sideset 8 surface in port8
sideset 6 surface in ss6

block 1 vol 1 3
block 2 vol 2

block all element type tetra10
transform mesh output scale 0.001

export genesis "win.gen" overwrite
```

# S3P - Window - S3P Input (1)

```
ModelInfo: {

 File: ./win.ncdf

 BoundaryCondition: {
  HFormulation: 0
  Magnetic: 1
  Electric: 2
  Exterior: 6
  Waveguide: 7, 8
 }

 Material : {
  Attribute: 1
  Epsilon:   1.0
  Mu:        1.0
 }

 Material : {
  Attribute: 2
  Epsilon:   9.37
  Mu:        1.0
 }

 SurfaceMaterial: {
  ReferenceNumber: 6
  Sigma: 5.8e7
 }
}
```

# S3P - Window - S3P Input (2)

```
FiniteElement: {
  Order: 1
  CurvedSurfaces: on
}

VerifyLinearSolver: yes

LinearSolver: {
   Solver:            SuperLU
}

WaveguideFrequency: 2.856e+9
/*
FrequencyScan: {
     //enable frequency scan of S parameter
    Start: 2.456e+9
    End:   3.056e+9
    Interval: 0.02e+9
}
*/

PostProcess: {
  Toggle: on
  Port Number: 0 //input port
  ModeFile: port0
}

CheckPoint: {
  Action: save
  Directory: vector1
}
```

# S3P - Window - S3P Input (3)

```
Port:
{
    ReferenceNumber: 7
    Origin:     0.0, 0.00, -0.060
    XDirection: -1.0, 0.0, 0.0
    YDirection: 0.0, 1.0, 0.0
    ESolver:
    {
        Type: Analytic
        Mode:
        {
            Mode number: 1
            Waveguide type: Rectangular
            Frequency: 2.856e+9
            Mode type: TE, 1, 0
            A: 0.072136
            B: 0.034036
        }
    }
}
```

```
Port:
{
    ReferenceNumber: 8
    Origin:     0.0, 0.00, 0.060
    XDirection: 1.0, 0.0, 0.0
    YDirection: 0.0, 1.0, 0.0
    ESolver:
    {
        Type: Analytic
        Mode:
        {
            Mode number: 1
            Waveguide type: Rectangular
            Frequency: 2.856e+9
            Mode type: TE, 1, 0
            A: 0.072136
            B: 0.034036
        }
    }
}
```