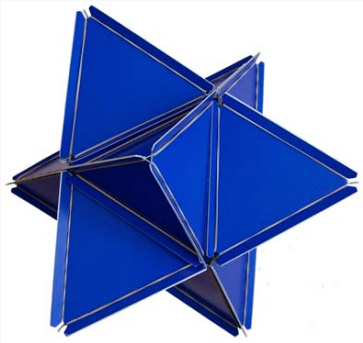
GEMC Tutorial

GEMC

This Tutorial and Support Files Can be Found at gemc.jlab.org

Please send any inaccuracies, improvements or features requests to ungaro@jlab.org

Lesson 5:
Geometry I/O (part II)



In this lesson:

Geometry:

- Creating Mother Volume
- Using Loop to create Paddles
- Making Paddles Sensitive
- Basics of Sensitivity and Hit Process Routine

Pre-requisites:

geo2.tar, available at gemc.jlab.org
or at /home/gemc/geo2.tar

Database I/O

First, we need a name for your new geometry. Untar geo2.tar inside a directory of your choice (`tar –xvf geo2.tar`)

Open go_table.
Change "mygeo" at the top to something unique. Your name for example:

```
set tables = (mestayer)
```

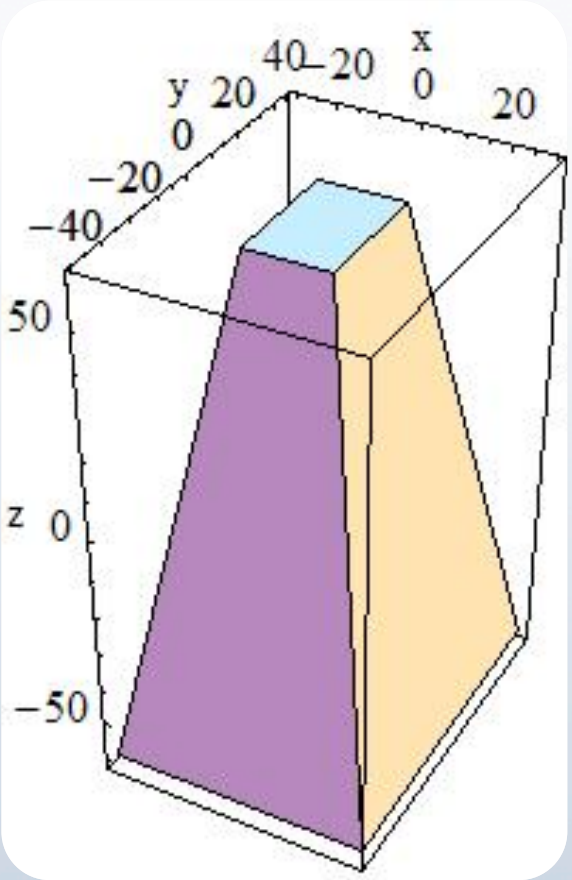
Open my_geometry.pl. This is the geometry script.
Change "mygeo" at the top to the same name you used in go_table.

```
my $envelope = 'mestayer';
my $file     = ' mestayer.txt';
```

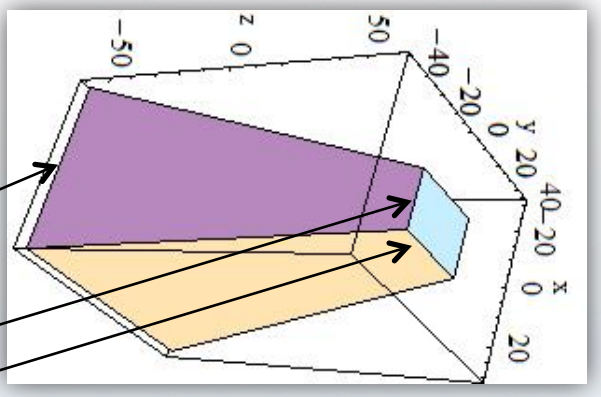
Database I/O

Wanted: Create many paddles with cylindrical symmetry. We could use Box but… Box would create gaps at the top.

We need Trd!



Database I/O




```
my $NUM_BARS = 10;
my $inches   = 2.54;

# Paddle Dimensions
my $dx1     =   2.0*$inches;        # width at top,cm
my $dx2     =   1.0*$inches;        # width at bottom, cm
my $dy      =   1.0*$inches;        # length, cm
my $dz      =  10.0*$inches;        # heigth, cm
my $theta0 = 360./$NUM_BARS;        # angle of one of the trapezoid sides
```

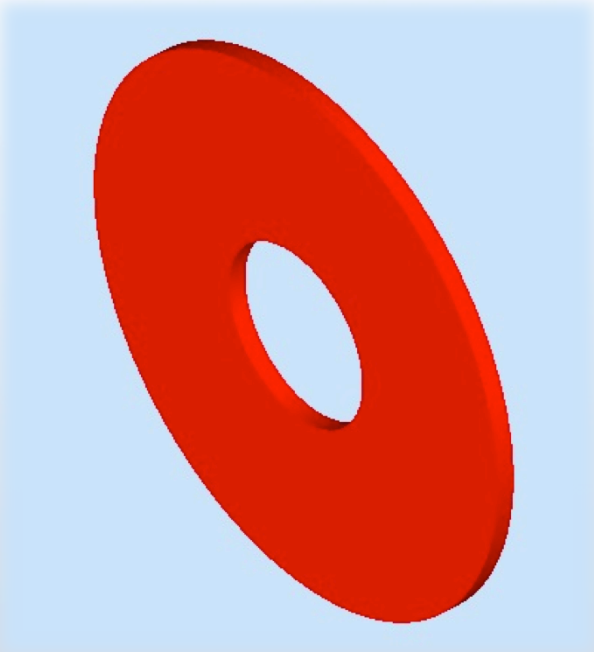
Database I/O

Let's first create the mother volume:

```
my $R      =  25.0 + $dz + 0.1;    # midway between R_outer and R_inner - cm
```

```
# Mother Volume
sub make_mother
{
      $detector{"name"}        = "mygeo";
      $detector{"mother"}      = "root";
      $detector{"description"} = "my geometry";
      $detector{"pos"}         = "0*cm 0.0*cm 0*mm";
      $detector{"rotation"}    = "0*deg 0*deg 0*deg";
      $detector{"color"}       = "ff0000";
      $detector{"type"}        = "Tube";
      my $Rin        = $R - $dz - 0.1;
      my $Rout       = $R + $dz + 0.1;
      my $DZ         = $dy;
      $detector{"dimensions"}  = "$Rin*cm $Rout*cm $DZ*cm 0*deg 360*deg";
      $detector{"material"}    = "Air";
      $detector{"mfield"}      = "no";
      $detector{"style"}       = 1;
      $detector{"sensitivity"} = "no";
      $detector{"hit_type"}    = "";
      $detector{"identifiers"} = "";
      print_det(\%detector, $file);
}
```

Database I/O



Mother Volume

Run

```
go_table  user_geometry
```

And then:

```
gemc -GT=mygeo -DATABASE=user_geometry
```

Remember to change that to whatever name you choose for your table

Making the Paddles

Uncomment the last line:

#make_paddles(); becomes
make_paddles();

And take a look at the routine make_paddles.

Not the "mother" volume name becomes what you created with

Making the Paddles

```
my $theta  = ($n-1)*$theta0;
my $theta2 = $theta + 90;
my $x     = sprintf("%.3f", $R*cos(rad($theta)));
my $y     = sprintf("%.3f", $R*sin(rad($theta)));
my $z     = "0";
```

For each paddle, this calculates the rotation and placement

```
$detector{"pos"}      = "$x*cm $y*cm $z*cm";
$detector{"rotation"}  = "90*deg $theta2*deg 0*deg";
$detector{"color"}     = "66bbff";
$detector{"type"}      = "Trd";
```

This places each Trd object with the proper rotation

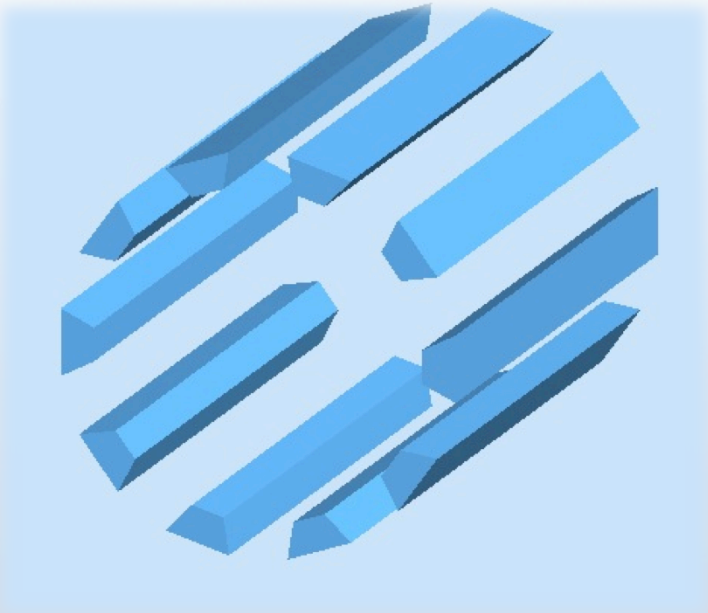
Now you can play with the parameters….
For example, make the tube longer!

```
my $dy    =   20.0*$inches;      # length
```

run "go_table user_geometry"

Making the Paddles

```
gemc -GT=mygeo -DATABASE=user_geometry
```



Play with number of paddles, their dimensions, etc. Make the mother invisible by setting its attribute "visibility" to zero.

Run some tracks through the scintillator…

What happens? Looks like nothing happens actually: We need to make those paddles sensitive….

You can tell they're not sensitive also by looking at the detector inspector (they're blue)

Remember: every time you change the geometry, run `go_table user_geometry`.
Every time you do that, the geometry is uploaded to the database and everybody can use it immediately

Sensitivity

In my_geometry.pl associate the paddles to the FLUX bank, and FLUX hit process routine… This is a special Sensitive Detector that record each track passing into the detector, and record Edep, etc.

```
$detector{"sensitivity"} = ”FLUX";
$detector{"hit_type"}    = ”FLUX";
```

```
$detector{"identifiers"} = ”id manual $n";
```

✓ What’s this line? This will associate a number to each paddle

✓ This number is the ID in the FLUX bank

Sensitivity

The Sensitive Detector is an object defined in the database. It describe the detector time window, threshold, etc.
The Hit Process routine is used to decide what to store in the output bank and to provide digitization.

$detector{"sensitivity"} = ”DC";
$detector{"hit_type"} = ”DC";

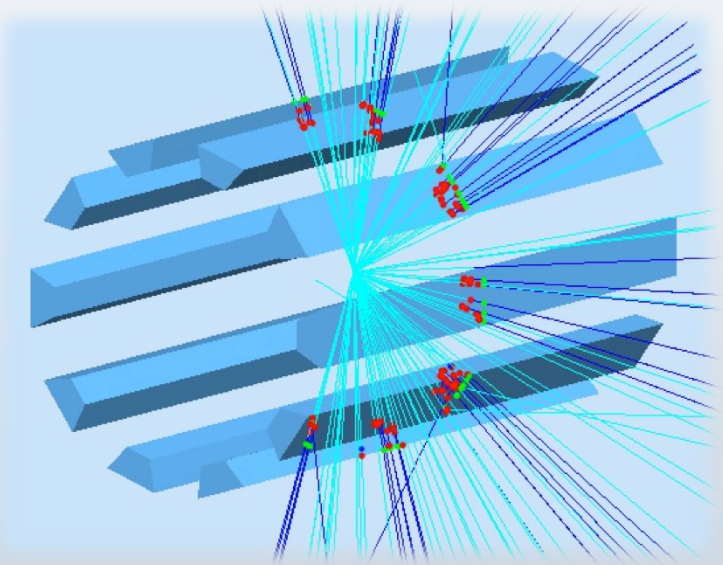
```
$detector{"identifiers"} = "sector ncopy 0 superlayer manual $isup layer manual $nlayer wire manual 1";
```

This line describe the hierarchy of the sensitive elements
“ncopy” means take the copy number
“manual” assign the number
This assignment can be done at run time as well
(example: the wire # in the drift chambers is assigned based on the track position)

Sensitivity

Run `go_table user_geometry`

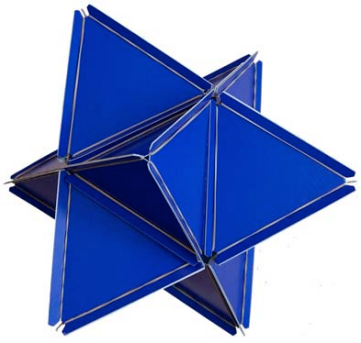
```
gemc -GT=mygeo -DATABASE=user_geometry –OUTPUT="txt, out.txt"
```



Send tracks to the paddles: now we have hits.

Run some event and look at the out.txt. Info is stored in the FLUX bank.

In the Detector Inspector, the paddles are now RED because they're sensitive.

End of Lesson 5:

Geometry I/O (part II)