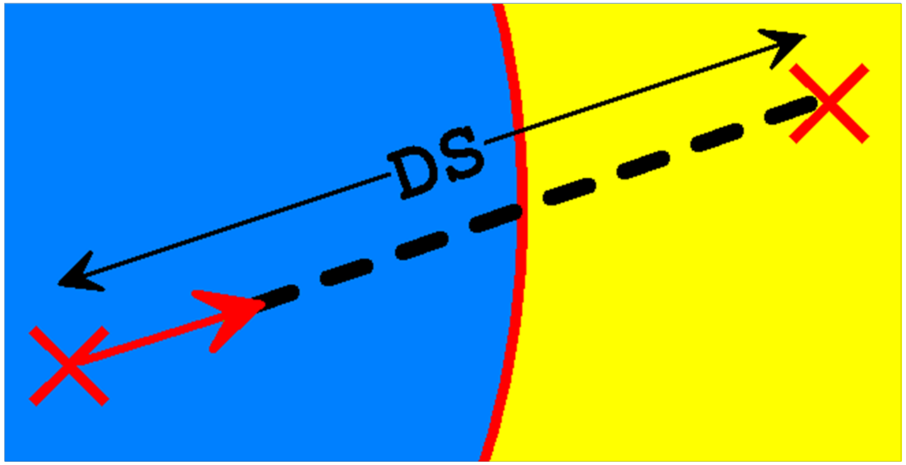
PENGEOM

EUTEMPE-RX module 03

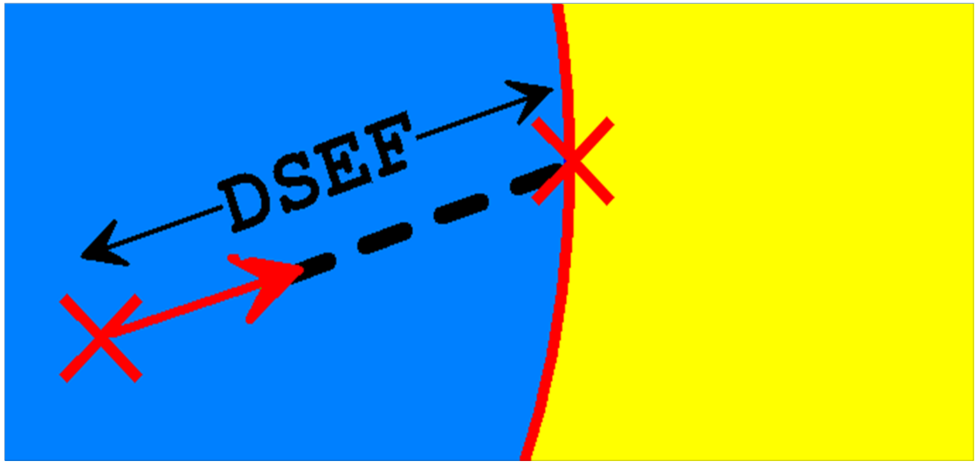
Monte Carlo simulation of x-ray imaging and dosimetry

Barcelona, June 2017

Particle tracking



Physics routines determine the distance to travel in an infinite medium

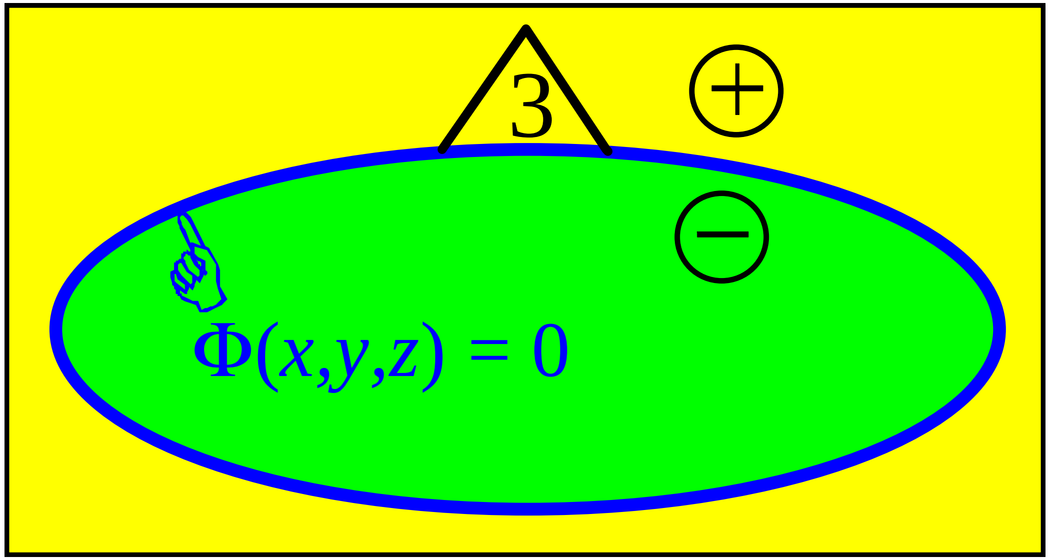


Geometry routines move the particle and stop it when the trajectory intersects and interface

Geometry routines...

- Move particles along straight trajectories
- Find the intersections with the surfaces that limit the body
- Determine the body and material after the interface
- Tell where the particle is (material and body)

Surfaces: basic concepts



- The surface divides the space in three mutually exclusive regions:
  1) $\Phi(\mathrm{r}) = 0$ the surface itself.
  2) $\Phi(\mathrm{r}) > 0$ “outside”.
  3) $\Phi(\mathrm{r}) < 0$ “inside”.

- The “sign” of a region is not an absolute property.
  The eq. $-\Phi(\mathrm{r}) = 0$ reverses the signs of the regions + and –.

Intersections

- The intersection between $\Phi(\mathbf{r}) = 0$ and a straight line $\mathbf{r} = \mathbf{r}_0 + \hat{\mathbf{v}}s$ is found by solving the equation $\Phi(\mathbf{r}_0 + \hat{\mathbf{v}}s) = 0$ in the unknown *s*. This limits the possible functional forms of $\boldsymbol{\Phi}$ in practice.

- There is a family of functions versatile enough to describe most of the objects found in real-life problems and for which the intersection equation can always be solved in a reasonable amount of time: the **quadrics.**

- Some non-quadric surfaces produce also a solvable intersection equation, but require much more CPU time. This is the case of a torus,

$$\left[\sqrt{x^2 + y^2} - a\right]^2 + z^2 - b^2 = 0$$

which gives rise to a 4th order polynomial equation.

Quadrics

$$\Phi(\mathbf{r}) = A_{xx}x^2 + A_{yy}y^2 + A_{zz}z^2 + A_{xy}xy + A_{yz}yz + A_{zx}zx +$$
$$+ A_x x + A_y y + A_z z + A_0 = 0$$

or, in matrix form,

$$\Phi(\mathbf{r}) = \mathbf{r}^{\mathrm{t}} A_2 \mathbf{r} + \mathbf{r}^{\mathrm{t}} A_1 + A_0 = 0$$

The intersection equation turns out to be (with $\mathbf{r} = \mathbf{r}_0 + \hat{\mathbf{v}} s$)

$$s^2 \left(\hat{\mathbf{v}}^{\mathrm{t}} A_2 \hat{\mathbf{v}} \right) + s \left(2\mathbf{r}_0^{\mathrm{t}} A_2 \hat{\mathbf{v}} + \hat{\mathbf{v}}^{\mathrm{t}} A_1 \right) + \left(\mathbf{r}_0^{\mathrm{t}} A_2 \mathbf{r}_0 + \mathbf{r}_0^{\mathrm{t}} A_1 + A_0 \right) = 0$$

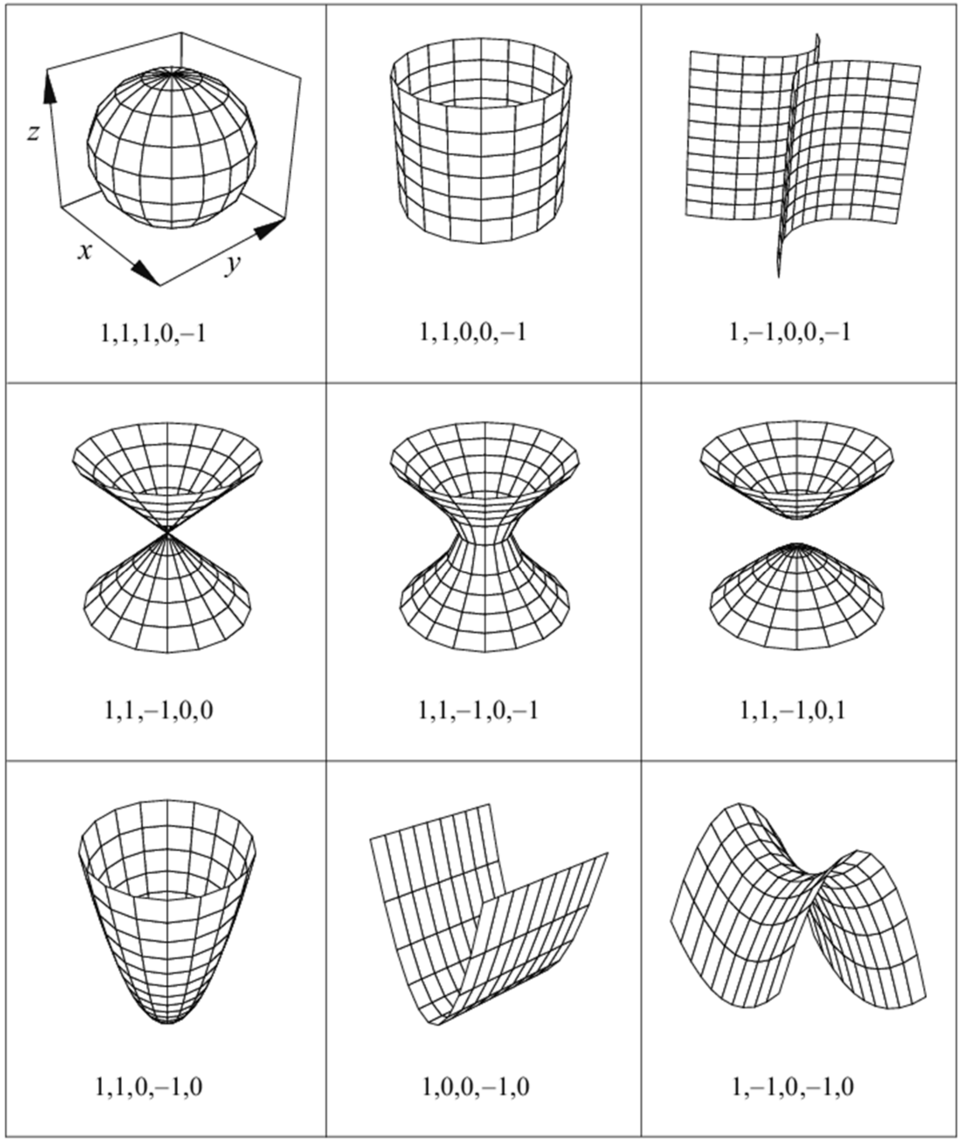
a second order polynomial in *s.*

Reduced form

Any quadric can be generated from its reduced form:

$$\Phi(\mathbf{r}) = I_1 x^2 + I_2 y^2 + I_3 z^2 + I_4 z + I_5 = 0$$

Reduced form	Indices					Quadric
$z - 1 = 0$	0	0	0	1	-1	plane
$z^2 - 1 = 0$	0	0	1	0	-1	pair of parallel planes
$x^2 + y^2 + z^2 - 1 = 0$	1	1	1	0	-1	sphere
$x^2 + y^2 - 1 = 0$	1	1	0	0	-1	cylinder
$x^2 - y^2 - 1 = 0$	1	-1	0	0	-1	hyperbolic cylinder
$x^2 + y^2 - z^2 = 0$	1	1	-1	0	0	cone
$x^2 + y^2 - z^2 - 1 = 0$	1	1	-1	0	-1	one sheet hyperboloid
$x^2 + y^2 - z^2 + 1 = 0$	1	1	-1	0	1	two sheet hyperboloid
$x^2 + y^2 - z = 0$	1	1	0	-1	0	paraboloid
$x^2 - z = 0$	1	0	0	-1	0	parabolic cylinder
$x^2 - y^2 - z = 0$	1	-1	0	-1	0	hyperbolic paraboloid



Non-planar
reduced quadrics

Transformations

Any quadric can be generated by starting from its reduced form and applying a combination (in the quoted order) of the following transformations:

- Scaling of the coordinates axis

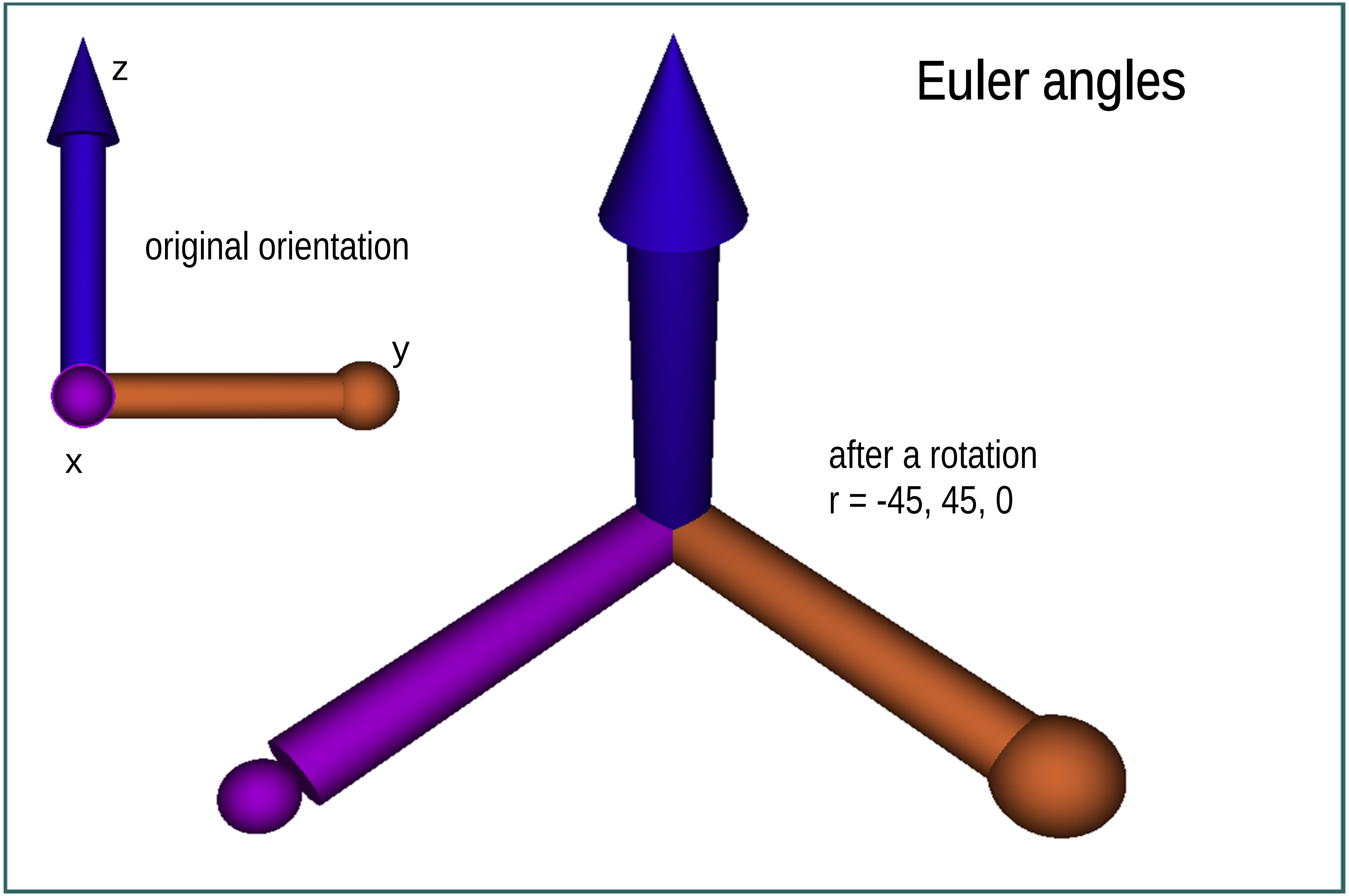
$$x \longrightarrow ax \qquad\qquad y \longrightarrow by \qquad\qquad z \longrightarrow cz$$

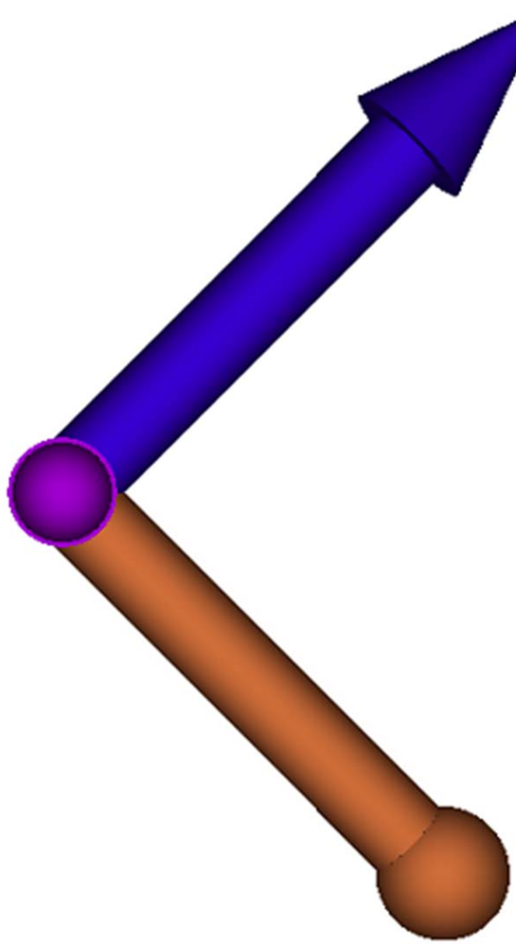
- Rotation

$$\mathbf{r} \longrightarrow R(\omega, \theta, \phi)\ \mathbf{r} = R_z(\phi)R_y(\theta)R_z(\omega)\ \mathbf{r}$$

- Translation

$$\mathbf{r} \longrightarrow \mathbf{r} + \mathbf{a}$$


Euler angles
z
original orientation
y
x
after a rotation
r = -45, 45, 0

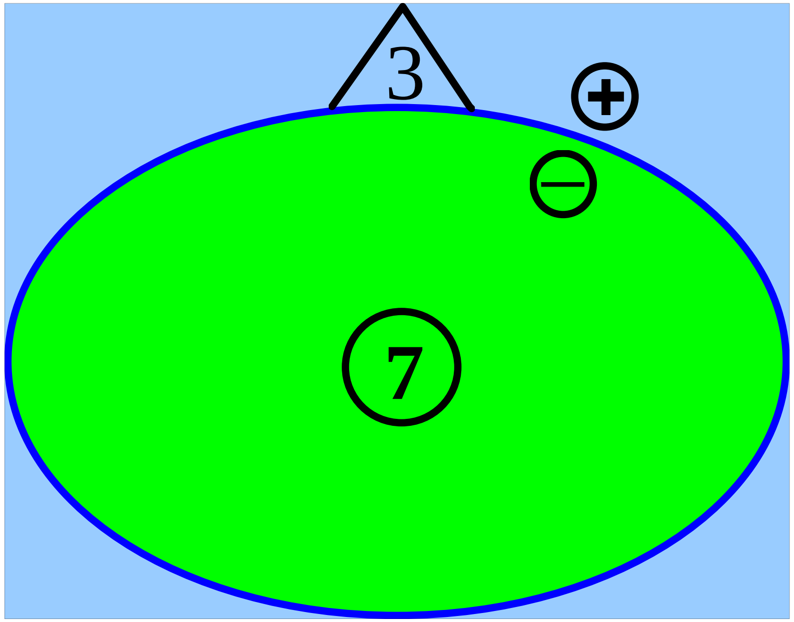


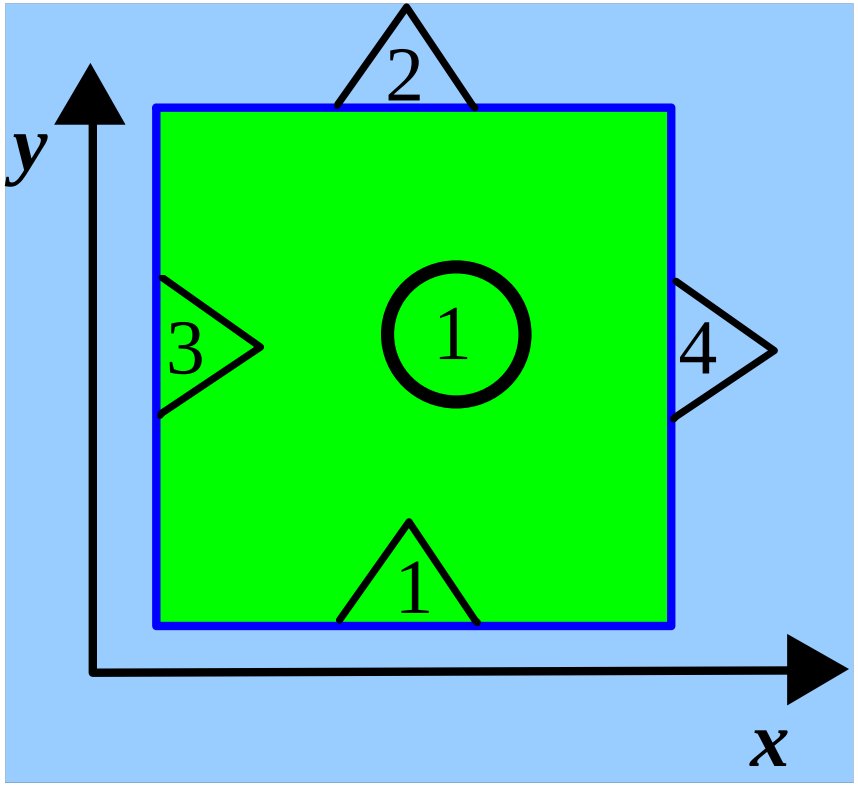
$$R_x(\theta) = R_z(90^o)\ R_y(-\theta)\ R_z(-90^o)$$

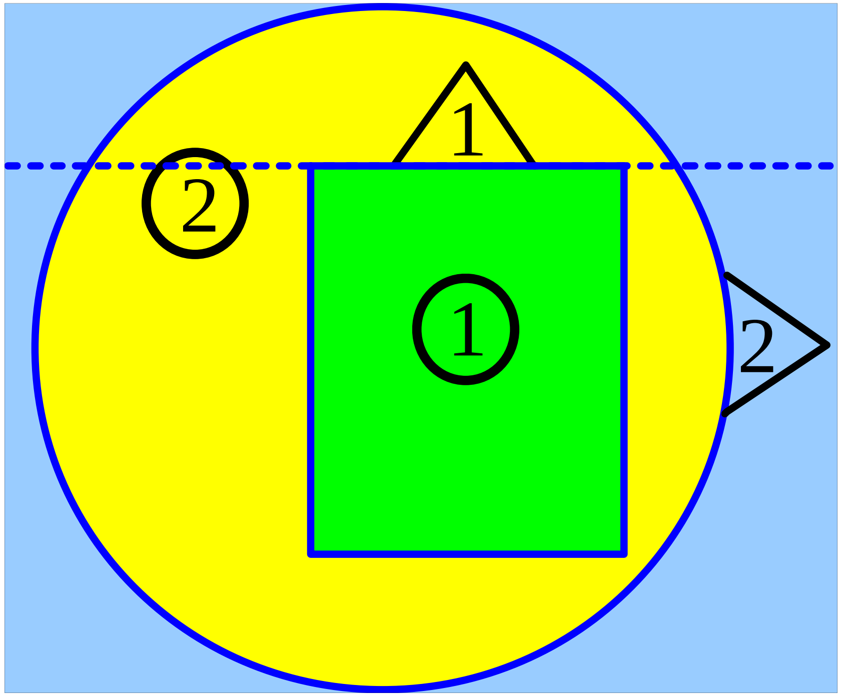
Syntax

```
0000000000000000000000000000000000000000000000000000000000000000
SURFACE (    )   reduced form
INDICES=( 1, 1, 1, 1, 1)
X-SCALE=(+1.000000000000000E+00,   0)              (DEFAULT=1.0)
Y-SCALE=(+1.000000000000000E+00,   0)              (DEFAULT=1.0)
Z-SCALE=(+1.000000000000000E+00,   0)              (DEFAULT=1.0)
  OMEGA=(+0.000000000000000E+00,   0) DEG          (DEFAULT=0.0)
  THETA=(+0.000000000000000E+00,   0) DEG          (DEFAULT=0.0)
    PHI=(+0.000000000000000E+00,   0) RAD          (DEFAULT=0.0)
X-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
Y-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
Z-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
0000000000000000000000000000000000000000000000000000000000000000
SURFACE (    )   implicit form
INDICES=( 0, 0, 0, 0, 0)
    AXX=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
    AXY=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
    AXZ=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
    AYY=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
    AYZ=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
    AZZ=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
     AX=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
     AY=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
     AZ=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
     A0=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
```

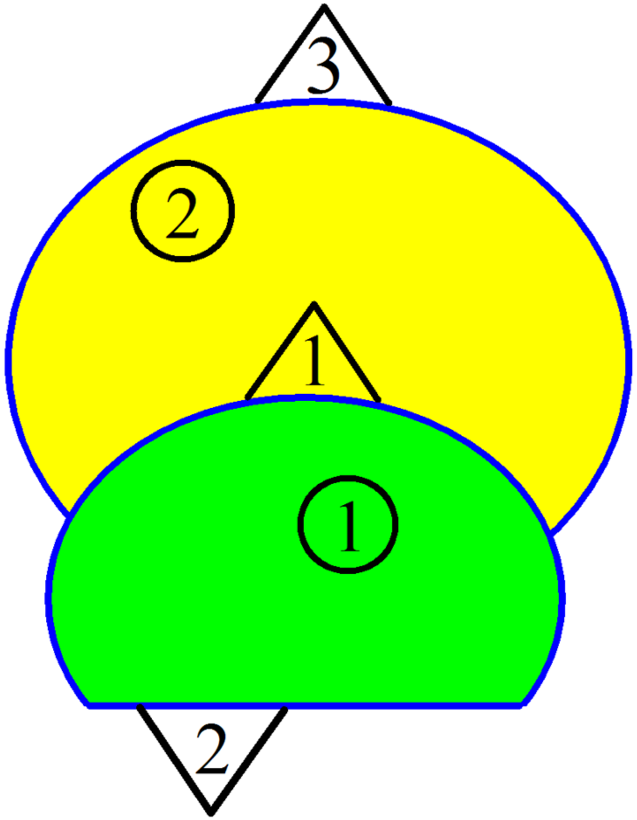
Definition of bodies







Some bodies require more than surfaces



Avoid substractions if there are simpler definitions

Syntax

```
0000000000000000000000000000000000000000000000000000000000000000
BODY     (    )   text
MATERIAL(    )
SURFACE (    ), SIDE POINTER=( 1)
BODY     (    )
MODULE   (    )
```

etc...

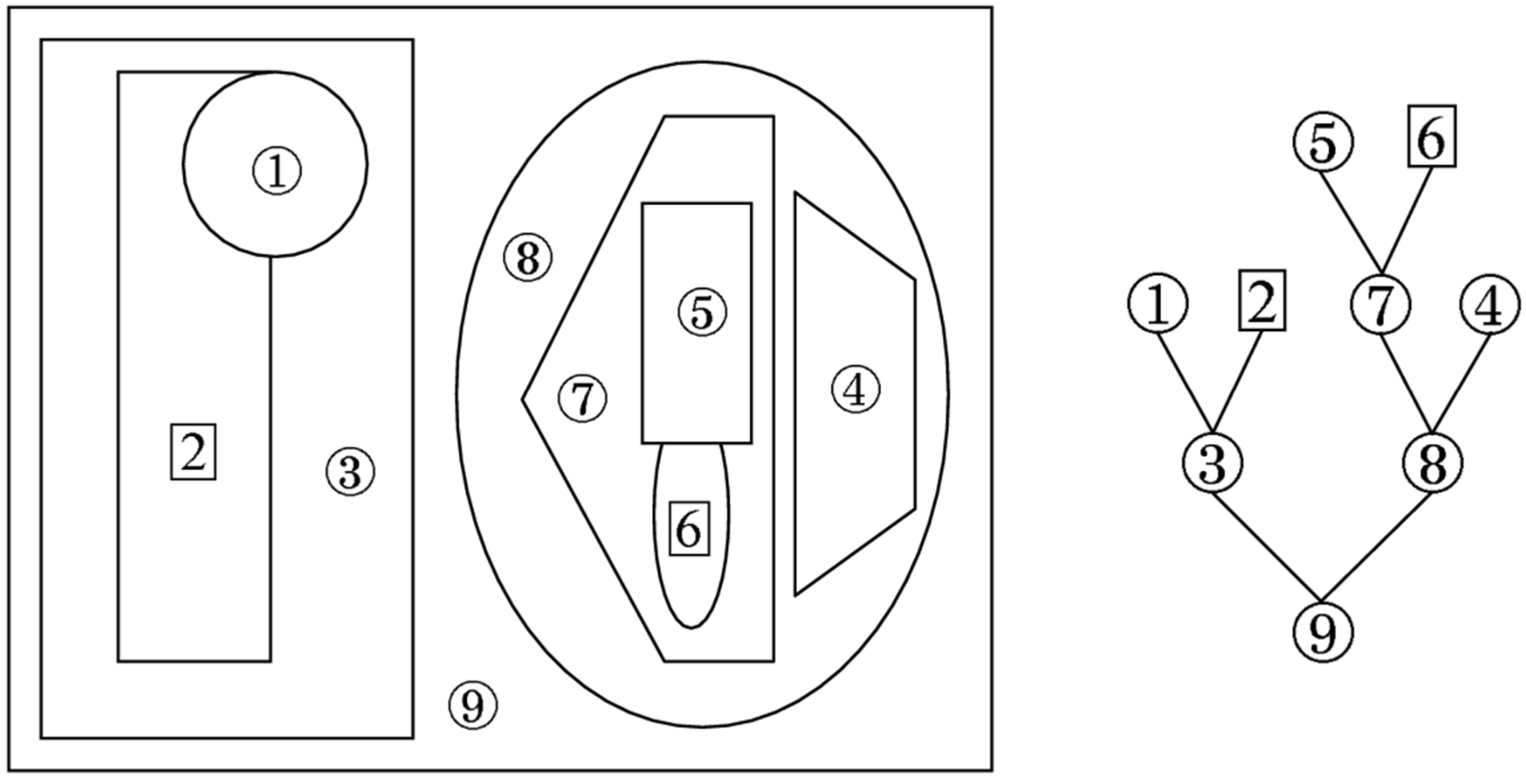
Lines defining default values can be removed from the definition file

Modules

Module ≡ Connected volume limited by quadrics.

- Think of a module as a “super-body” that defines a part of a more complex system.
- It may contain bodies and other modules. These have to be fully contained in the parent module.
- Modules cannot overlap with modules or bodies other than the ones contained in it.
- The “cavities” of a module can be filled with a default material.
- A module can be rotated/translated as a whole.

Hierarchical structure



Syntax

```
0000000000000000000000000000000000000000000000000000000000000000
MODULE  (    )   text
MATERIAL(    )
SURFACE (    ), SIDE POINTER=( 1)
BODY    (    )
MODULE  (    )
1111111111111111111111111111111111111111111111111111111111111111
  OMEGA=(+0.000000000000000E+00,   0) DEG          (DEFAULT=0.0)
  THETA=(+0.000000000000000E+00,   0) DEG          (DEFAULT=0.0)
    PHI=(+0.000000000000000E+00,   0) RAD          (DEFAULT=0.0)
X-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
Y-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
Z-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
```

Other options

```
0000000000000000000000000000000000000000000000000000000000000000
CLONE    (    )   copies one module and moves it
MODULE   (    )   original module
1111111111111111111111111111111111111111111111111111111111111111
  OMEGA=(+0.000000000000000E+00,   0) DEG          (DEFAULT=0.0)
  THETA=(+0.000000000000000E+00,   0) DEG          (DEFAULT=0.0)
    PHI=(+0.000000000000000E+00,   0) RAD          (DEFAULT=0.0)
X-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
Y-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
Z-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
```

```
0000000000000000000000000000000000000000000000000000000000000000
INCLUDE
FILE=   (filename.ext)
```

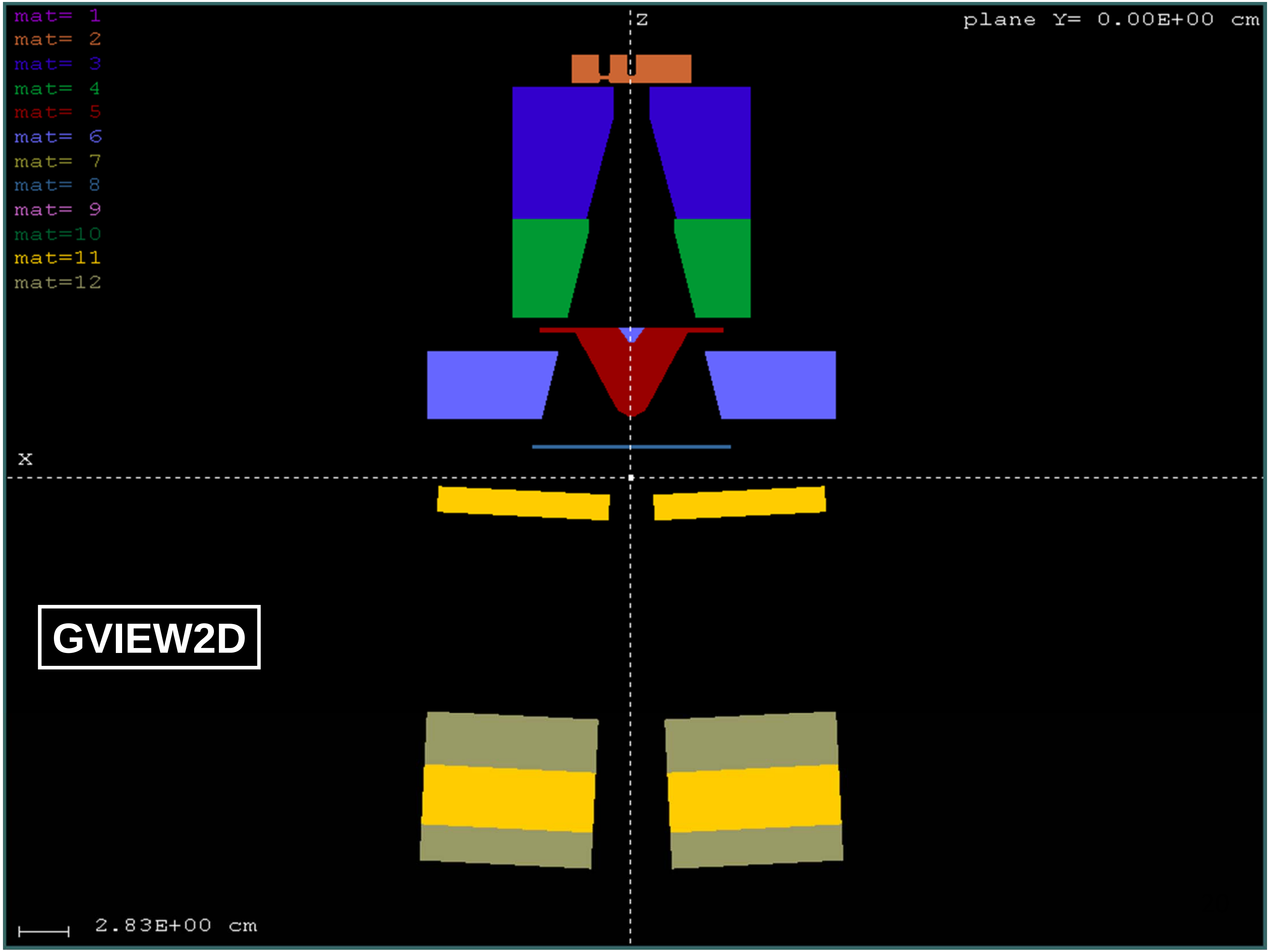
Viewers

Tools available (for free)

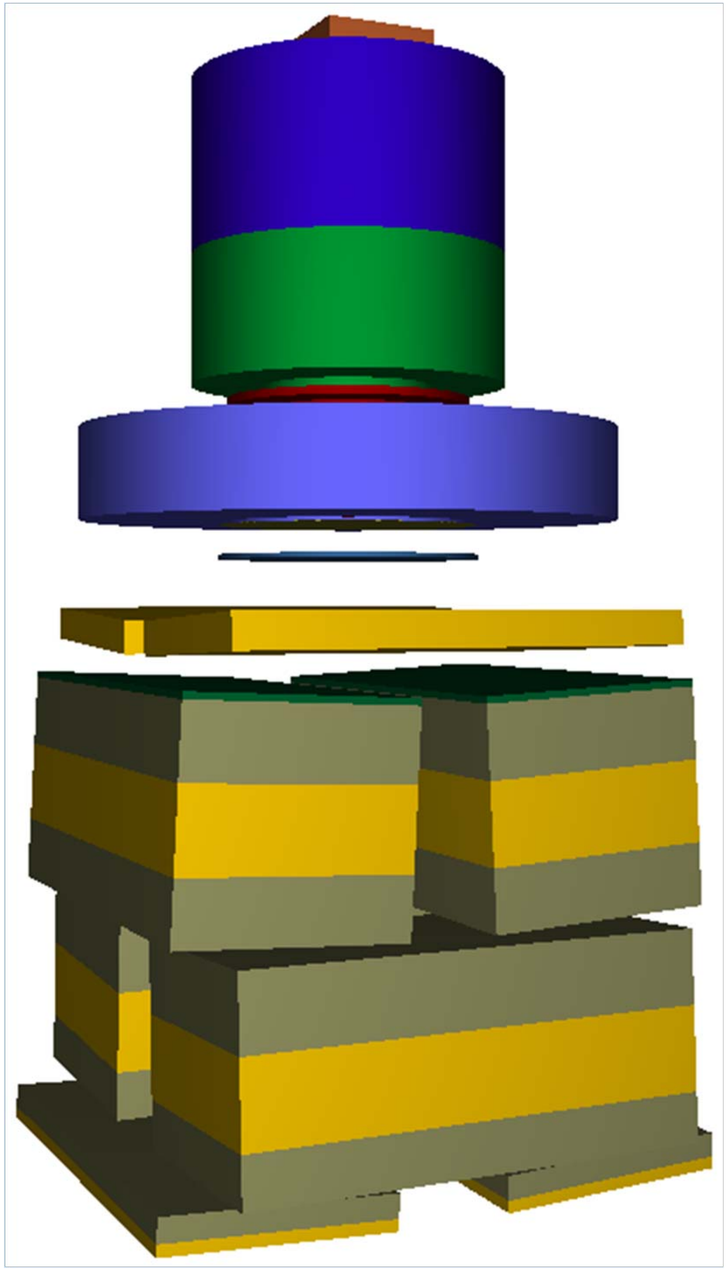
From the PENELOPE package (under ~/other/gview/.)

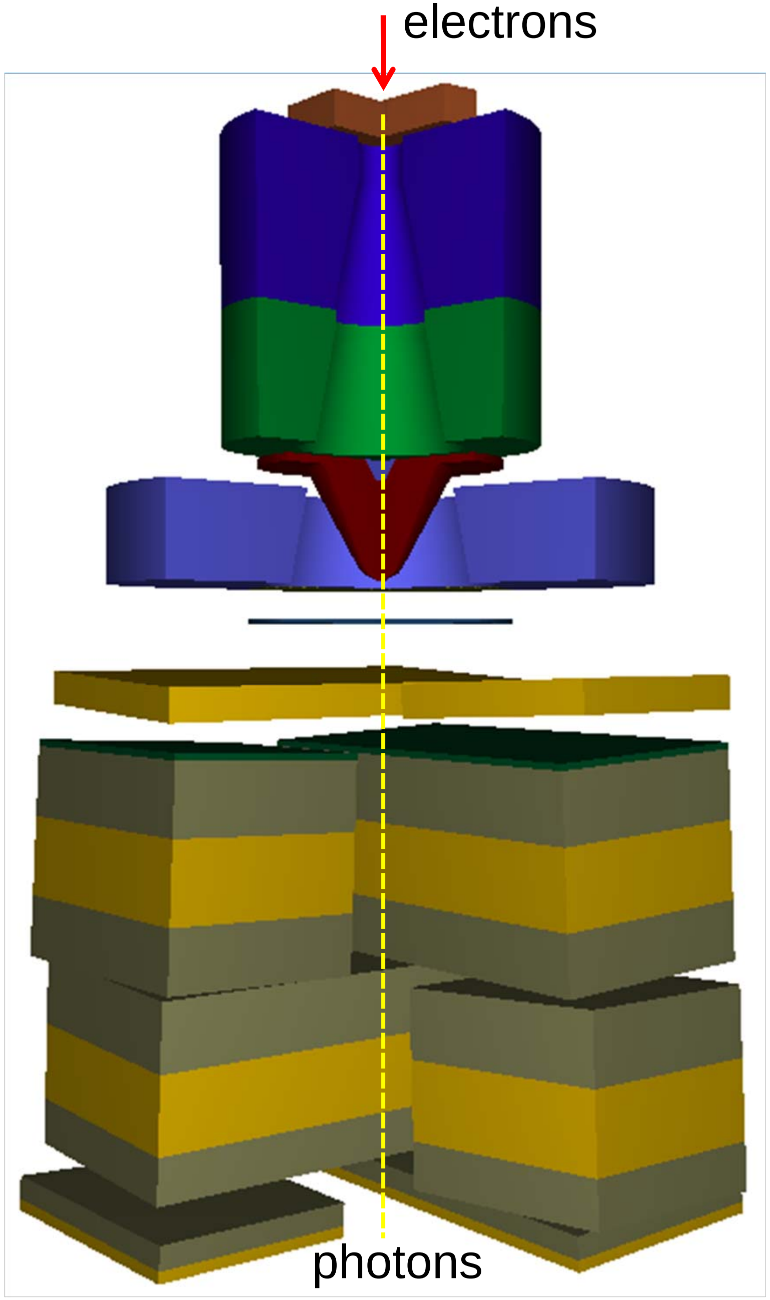
- GVIEW2D
- GVIEW3D

Both run under MS Windows. They can also be executed on Linux using a Windows emulator like wine.


mat= 1
mat= 2
mat= 3
mat= 4
mat= 5
mat= 6
mat= 7
mat= 8
mat= 9
mat=10
mat=11
mat=12
Z
plane Y= 0.00E+00 cm
X
GVIEW2D
2.83E+00 cm

GVIEW3D





More viewers

From other sources:

- POV-Ray™ (Persistence Of Vision)
  High-quality tool for creating 3D graphics. Available for various OSs.
  http://www.povray.org

- PenGeom.jar
  F Salvat et al.
  Written in Java, runs on all common OSs
  Contact the authors to get a copy

- PENGL
  Julia Riede
  Written in C using OpenGL for Windows
  http://www.winsite.com/electron/electron+transport+system/index3.html

Debugging

GVIEW generates a debugging file named geometry.rep.

- GVIEW stops whenever a syntax error or an insurmountable inconsistency are found.
- The offending input datum appears in the last printed lines of the debugging file.
- Errors or inconsistencies in the definition of the geometry are usually identified by visual inspection of the rendering created by GVIEW2D.
- PENGEOM admits up to 10,000 surfaces and 5,000 bodies (and modules). The maximum number of bodies in a module or the number of surfaces defining a body is set to 250. These parameters can be changed by editing the source file pengeom.f (`module PENGEOM_mod`) and recompiling the code.

Thank you.