flair for FLUKA geometry editor

Vasilis.Vlachoudis@cern.ch

FLUKA Meeting 09.07.2010

What's new in Version 0.8.3 [1/2]

- Multiple frames – fully customizable
- FLUGG support
- Input Editor improvements with most important
  - Tip help for every item value (short description + default) bodies (definition) in the region
  - Indentation of cards (towards integration of #include)
  - Accelerated display
  - Multiple editing, by selecting a range of similar cards
  - Expansion of parenthesis
- Customized file dialog to easier searching, deleting, renaming files as well creation of new folders
- MCNP exporting to macro bodies + importing (basic)

What's new in Version 0.8.3 [2/2]

- Improved Customize dialog and Gnuplot definitions
- Multiple selection for rules editing in "Data merge"
- Improvements in the plotting:
  - Use of styles for full customization of plots
  - Rebinning of USRBINs
  - Gnuplot reference in the manual

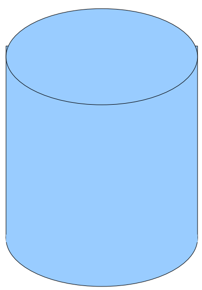
- Integration of the Geometry Editor

2D Geometry Editor

- Working on 2D cross sections of the geometry
- Creating and editing bodies/regions in a graphical way
- Most of the objects are 2D extruded in the 3rd dimension
- Pros
  - Fast display of complex geometries
  - Visual selection and editing of zones
  - Use real curve of bodies with no conversion to vertices/edges
  - Interactive debugging with information of problematic body regions and zones
  - No use of any additional hardware (plain X11 libraries)
- Cons
  - No interactive 3D display
  - Blind in 3rd dimension
    [could be compensated with raytracing]
  - Difficult to orientate in an unknown geometry

How it works

- All bodies are converted to a set (up to 6) quadratic equations:
  $c_x x^2+c_y y^2+c_z z^2 + c_{xy} xy+c_{xz} xz+c_{yz} yz + c_x x+c_y y+c_z z + c \le 0$



RCC → 3 quadratic equations

1. $x^2 + y^2 - R^2 \le 0$
2. $-z - 0 \le 0$
3. $z - h \le 0$

Sign defines the location **+** = outside, **0** = on surface, **-** = inside

Then it is transformed to the direction of the H-vector

- Quadratic can be represented in 4x4 matrix format

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}^T \begin{bmatrix} C_x & C_{xy}/2 & C_{xz}/2 & C_x/2 \\ C_{xy}/2 & C_y & C_{yz}/2 & C_y/2 \\ C_{xz}/2 & C_{yz}/2 & C_z & C_z/2 \\ C_x/2 & C_y/2 & C_z/2 & C \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0 \quad \text{or} \quad X^T \cdot Q \cdot X = 0$$

- Any transformation of the system $X = R \cdot X'$
  will modify the quadratic as

$$X^T \cdot R^T \cdot Q \cdot R \cdot X = 0 \text{ with } Q' = R^T \cdot Q \cdot R$$

the new equation of the quadratic

Conics

- The quadratic equations/matrices are rotated/translated to the viewport location and then are converted to conic section assuming z'=0

Rotation Translation

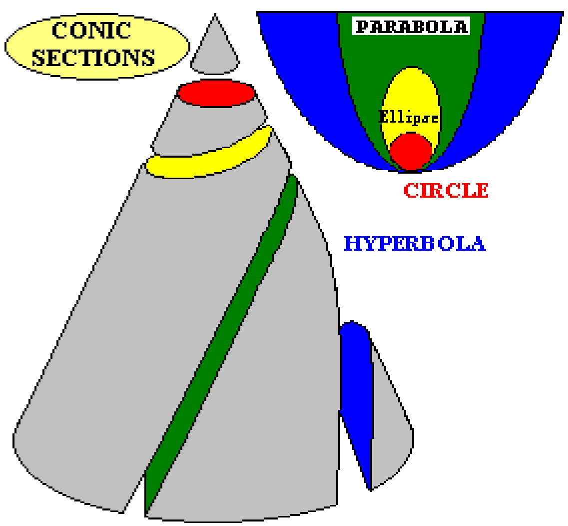
$$ax^2 + 2hxy + by^2 + 2gx + 2fy + c = 0$$

- The conics can be represented in matrix format as:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}^T \begin{bmatrix} a & h & g \\ h & b & f \\ g & f & c \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

- Similarly to the quadratics the conics can be transformed (rotated/translated) using matrix operations.
- Under these operations the following quantities are invariant

$$\Delta = \begin{vmatrix} a & h & g \\ h & b & f \\ g & f & c \end{vmatrix} \qquad I=a+b \qquad J=ab-h^2$$



Conics Types

Conic	Form	Parametric	Δ	J	Δ/I	$ca-g^2 + bc-f^2$
Real Ellipse	$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$	$x=c_1 + c_2\cos t + c_3\sin t$ $y=c_4 + c_5\cos t + c_6\sin t$	$\neq 0$	+	-	
Virtual Ellipse	-//-		$\neq 0$	+	+	
Hyperbola	$\frac{x^2}{a^2}-\frac{y^2}{b^2}=1$	$x=c_1 + c_2\sec t + c_3\tan t$ $y=c_4 + c_5\sec t + c_6\tan t$	$\neq 0$	-		
Parabola	$y^2=4ax$	$x=c_1 + c_2t + c_3t^2$ $y=c_4 + c_5t + c_6t^2$	$\neq 0$	0		
Real intersecting lines	$(l_1x+m_1y+n_1).$ $(l_2x+m_2y+n_2)=0$	$x=c_1 + c_2t$ $y=c_4 + c_5t$ x2	0	-		
Conjugate complex intersecting lines	-//-		0	+		
Real distinct parallel lines	-//-	$x=c_1 + c_2t$ $y=c_4 + c_5t$ x2	0	0		-
Conjugate complex parallel lines	-//-		0	0		+
Coincident lines	$l_1x+m_1y+n_1=0$	$x=c_1 + c_2t$ $y=c_4 + c_5t$	0	0		0

Intersection of Conics

Intersect all body conics that are visible in the current viewport with each other. There are two ways of calculating the intersection of conics

Using a pencil of conics

- Given two conics C1 and C2
- Consider the pencil of conics $\lambda C1 + \mu C2$
- Identify the homogeneous parameters (λ,μ) which corresponds to the degenerate conic of the pencil (lines).

$$\det(\lambda C1 + \mu C2) = 0$$

  a 3rd degree equation.
- Decompose the degenerate conic C0 into two lines
- Intersect each line with one of the initial conics

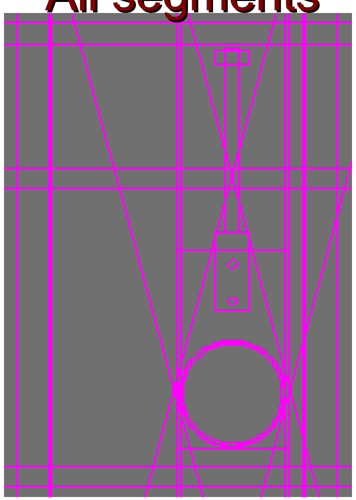
Direct substitution

- Solve the 2^{nd} degree equation of conic C1 for y
- Substitute in the C2 => generate a 4^{th} degree equation on x
- Solve the quartic equation
- Find the y coordinates for every x solution for C1 and C2
- Find the common (x,y) points

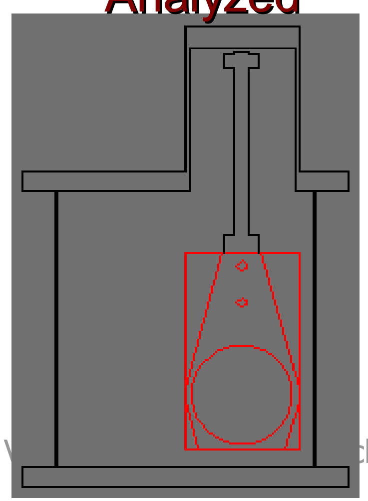
Drawing conics

- Having calculated all intersections of all conics, and with the window borders,
  Calculate the parametric t corresponding to every intersection
- Sort the the intersections according to t
- Inspect segment on actual geometry if it belongs to zero, one, two or more regions.
  - If it belongs to only one region then ignore
  - If it belongs to two different regions then plot as normal
  - If it belongs to zero or more than two regions then plot as error

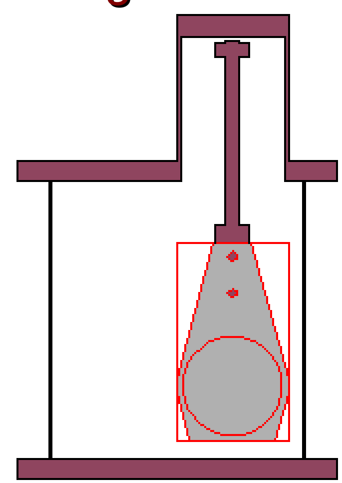
All segments



Analyzed



Region filled



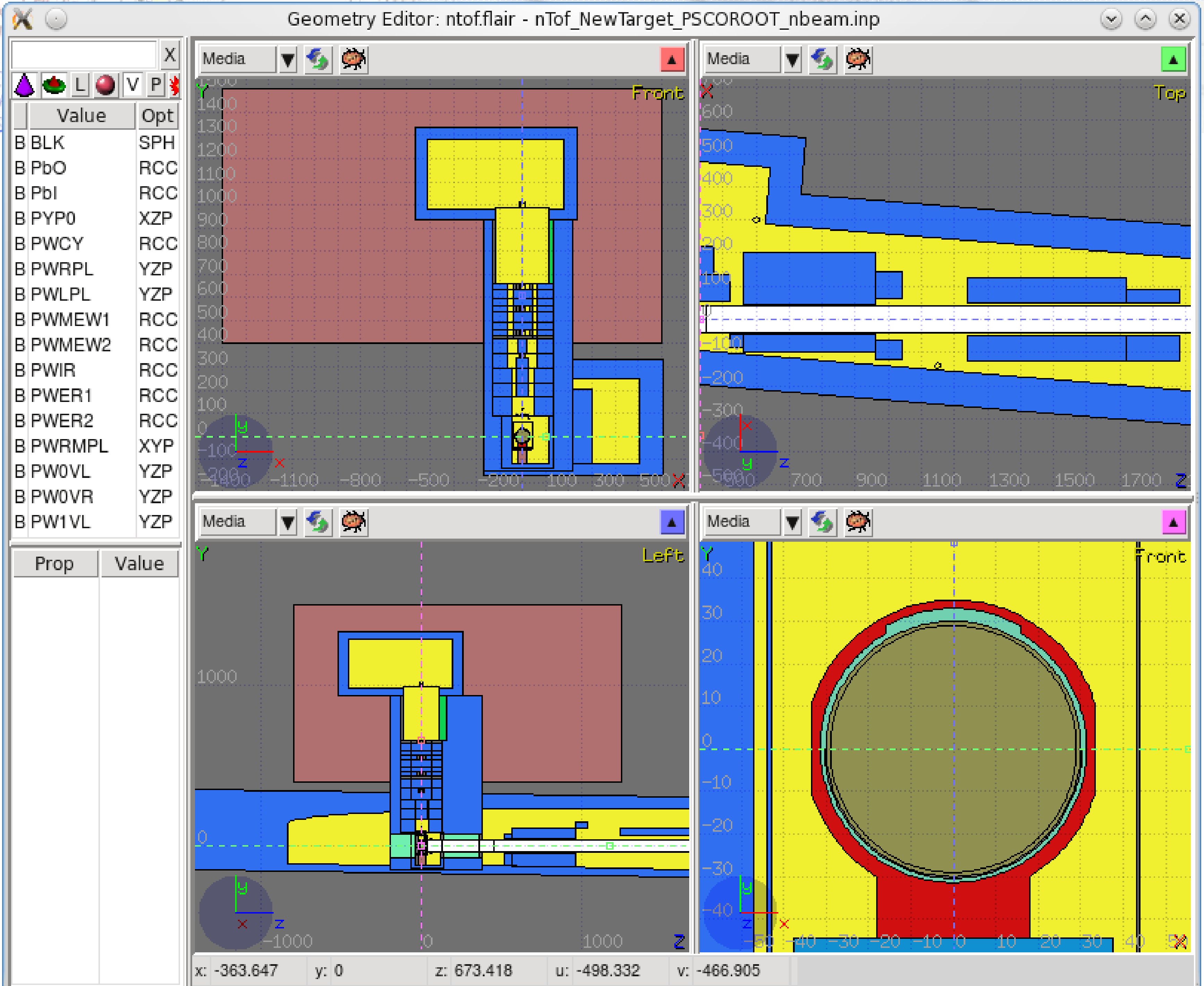
Program

Plotting engine

- Language: “simple” C++ (as portable as possible)
- No use of ANY external library
- Drawing directly in a bitmap array
- All graphic operations with home source code
- Fully re-entrant and threaded
- Modestly robust in numerical precision.
  Accuracy of operations eps: 10^{-8} up to 10^{9}
- Heavily optimized

Interface (integrated into flair)

- High level interface is written in python with tk
- Low level interface with C++, tcl/tk and x11 libraries


Geometry Editor: ntof.flair - nTof_NewTarget_PSCOROOT_nbeam.inp
Value
Opt
B BLK SPH
B PbO RCC
B PbI RCC
B PYP0 XZP
B PWCY RCC
B PWRPL YZP
B PWLPL YZP
B PWMEW1 RCC
B PWMEW2 RCC
B PWIR RCC
B PWER1 RCC
B PWER2 RCC
B PWRMPL XYP
B PW0VL YZP
B PW0VR YZP
B PW1VL YZP
Prop
Value
Media
Front
Top
Left
Front
x: -363.647
y: 0
z: 673.418
u: -498.332
v: -466.905

Status & Future

Plotting engine

- Geometry engine operates reasonably
- Quite robust for debugging geometries
- Could be further optimized while scanning regions for errors
- To be added a 3D ray tracing for vacuum/low density regions
- Exporting to various formats (dxf, eps, png)

Interface

- A lot of work for a user friendly interface
- Will allow editing of regions by simply drawing/selecting the zones. Then the program will construct the logical operations

Maybe first release in autumn 2010