A Twisted Trapezoidal Shape for Geant4

Based on „Stereo Mini-jet Cells in a Cylindrical Drift Chamber“
(hep-ex/0303014v1, K. Hoshina et al.)

Oliver Link, EP-SFT
Apr 05
**G4VTwistedFacted**

Base class: `G4VSolid`, Similar to `G4TwistedTubs`
- `DistanceToIn`
- `DistanceToOut`
- `Inside ..`
- Constructor calls
  - `G4TwistedTrapAlphaSide`
  - `G4TwistedTrapParallelSide`
  - `G4TwistedTrapBoxSide` as special case for a twisted box
  - `G4FlatTrapSide` (for the endcaps)
The resulting solution contains terms in Sin(\(\phi\)) and Cos(\(\phi\)) which are approximated with Padé expansions.

Polynom: 7\(^{th}\) order.

\[
\begin{cases}
(w(u, \phi) + \Delta x \frac{\phi}{\Delta \phi}) \cos \phi - (u + \Delta y \frac{\phi}{\Delta \phi}) \sin \phi = p_x + tv_x \\
(w(u, \phi) + \Delta x \frac{\phi}{\Delta \phi}) \sin \phi + (u + \Delta y \frac{\phi}{\Delta \phi}) \cos \phi = p_y + tv_y \\
\frac{L\phi}{\Delta \phi} = p_z + tv_z
\end{cases}
\]

\(\varphi \in [-\frac{1}{2} \Delta \phi, +\frac{1}{2} \Delta \phi]\)

\(u \in [-\frac{1}{2} b(\phi), +\frac{1}{2} b(\phi)]\)

\[
w(u) = \frac{a(\phi)}{2} + \frac{d(\phi) - a(\phi)}{4} - u \left[ \frac{d(\phi) - a(\phi)}{2b(\phi)} - \tan \alpha \right]
\]
Planarity condition

An untwisted trapezoid requires planar surfaces.

Planarity is equivalent to

\[ \beta_1 = \beta_2 \]

\[ \frac{d_1 - a_1}{b_1} = \frac{d_2 - a_2}{b_2} \]
**G4TwistedBox**

```
G4TwistedBox(Name, Δφ, fDx, fDy, fDz)
```

Eg: 70*deg, 20*cm, 30*cm, 80*cm
G4TwistedTrd

\[ G4TwistedTrap(\text{Name}, fDx_1, fDx_2, fDy_1, fDy_2, fDz, \Delta \phi) \]

Eg: 15*cm, 25*cm, 30*cm, 20*cm, 80*cm, 70*deg
**G4TwistedTrap**

Twisted trapezoid with equal sized endcaps and no tilt angle.

[G4TwistedTrap("Trap",70*deg,15*cm,25*cm,30*cm,80*cm)]

Twist angle $a$

$\frac{a}{2}$ $\frac{b}{2}$ $\frac{d}{2}$ $L/2$

$G4TwistedTrap("Trap",70*\text{deg},15*\text{cm},25*\text{cm},30*\text{cm},80*\text{cm})$
G4TwistedTrap

General Twisted trapezoid with different sized endcaps and and tilt angle.

$$G4TwistedTrap(Name, \Delta \phi, fDz, \theta, \phi, fDx_1, fDx_2, fDy_1, fDx_3, fDx_4, fDy_2, \alpha)$$
Solved Problems

The visualisation showed cracks in the surface and wrongly tracked events. This issue was successfully solved by a

- division by zero check (added special case)
- new surface-point finder
- introducing G4JTPolynomialSolver
- special treatment for events parallel to the surface

before

after
Two intersections
Testing the Solid

With traditional tools

• test10: successful
• SolidsChecker:
  1 event „Track stuck“ out of 100 M

... and with a new testing tool

• SurfaceChecker: successful
SurfaceChecker

- Generate a random particle position $P$ and a random point $X^{true}$ on the surface of the solid.
- Ask G4 for the intersection point given the point $P$ and its direction $v$ (select the intersection closest to $X^{true}$).

The distance $\delta$ between $X^{true}$ and $X^{reconst.}$ gives us information about the goodness of the reconstruction.
Results I
Results II
Conclusions

• **G4TwistedBox** with equal endcaps is replaced by the new generic version.

• **G4TwistedTrd** with rectangular endcaps of different size (but no tilt angle) added.

• **G4TwistedTrap** with trapezoidal endcaps of different size and tilt angle added. The old version with equal endcaps is covered by the generic version.

• **G4JTPolynomialSolver** to solve the high order polynomials is added to HEPNumerics.