

Geom Module V1.4

- User Guide -

C. Benoit, G. Jeanfaivre, S. Peron et P. Raud
Onera / DSNA

September 23, 2009

1 Geom : Geometry definition module

1.1 Preamble

A geometry is assumed here to be defined discretely with a great number of points. A geometry is a *i*-array or a *BAR*-array for 2D simulations and a *i,j*-array, a *QUAD*-array or a *TRI*-array for 3D simulations. A **polyline** is defined as a *C0* *i*-array which contains only the polyline points (with no extra discretization points). Geometry is stored in a *Converter* array or in a *pyTree* zone node, following the selected interface.

For use with the *Converter* array interface, you must import the *Geom* module:

```
import Geom as D
```

Then, in the following, *a* is an array, and *A* a list of arrays.

For use with the *pyTree* interface, you must import the module:

```
import Geom.PyTree as D
```

Then, in the following, *a* is a zone node and *A* is a list of zone nodes or a complete *pyTree*.

1.2 Analytical geometries

Create a *naca* *i*-array with less than *N* points and of thickness *e* (*e*=12, 15...) in the (*x,y*) plane:

```
a = D.naca(e, N)
```

(See : [Examples/geom/naca.py](#)) (See : [Examples/geom/nacaPT.py](#))

Create a line *i*-array between point (*x1,y1,z1*) and point (*x2,y2,z2*), discretized with *N* points:

```
a = D.line((x1,y1,z1), (x2,y2,z2), N)
```

(See : [Examples/geom/line.py](#)) (See : [Examples/geom/linePT.py](#))

Create a polyline *i*-array from a list of points [(*x1,y1,z1*),(*x2,y2,z2*),...,(*xN,yN,zN*)], containing no extra discretization points:

```
a = D.polyline([(x1,y1,z1),(x2,y2,z2),...,(xN,yN,zN)])
```

(See : Examples/geom/polyline.py) (See : Examples/geom/polylinePT.py)

Create an i-array made of a portion of circle with N points of center C, radius R, between angles tetas and tetae (in degrees) in the (x,y) plane:

```
a = D.circle((xc,yc,zc), R, tetas, tetae, N)
```

(See : Examples/geom/circle.py) (See : Examples/geom/circlePT.py)

Create a Bezier i-array of N points or i,j-array of NxM points using control points. Control points coordinates are defined in a i-array or a i,j-array designated here by c:

```
a = D.bezier(c, N, M)
```

(See : Examples/geom/bezier.py) (See : Examples/geom/bezierPT.py)

Create a spline i-array of N points or i,j-array of NxM points using control points. Control points coordinates are defined in a i-array or i,j-array. Order is the order of spline:

```
a = D.spline(c, order, N, M)
```

(See : Examples/geom/spline.py) (See : Examples/geom/splinePT.py)

Create a NURBS i-array of N points or i,j-array of NxM points using control points and associated weights. Control points coordinates and Weight are defined in a i-array or i,j-array. Order is the order of basis functions:

```
a = D.nurbs(c, order, N, M)
```

(See : Examples/geom/nurbs.py) (See : Examples/geom/nurbsPT.py)

Create an i-array from a user defined parametric function:

```
a = D.curve(f, N)
```

(See : Examples/geom/curve.py) (See : Examples/geom/curvePT.py)

Create a i,j-array made of a cone with NxN points of center C, basis Radius Rb, top Radius Rt and height H:

```
a = D.cone((xc,yc,zc), Rb, Rt, H, N)
```

(See : Examples/geom/cone.py) (See : Examples/geom/conePT.py)

Create an i,j-array made of a sphere with NxN points of center C and radius R:

```
a = D.sphere((xc,yc,zc), R, N)
```

(See : Examples/geom/sphere.py) (See : Examples/geom/spherePT.py)

Create a sphere made of 6 i,j-arrays with 6xNxN points of center C and radius R:

```
A = D.sphere6((xc,yc,zc), R, N)
```

(See : Examples/geom/sphere6.py) (See : Examples/geom/sphere6PT.py)

Create a TRI-array made of a single triangle with 3 points P1, P2, P3:

```
a = D.triangle((x1,y1,z1), (x2,y2,z2), (x3,y3,z3))
```

(See : Examples/geom/triangle.py) (See : Examples/geom/trianglePT.py)

Create a QUAD-array made of a single quadrangle with 4 points P1, P2, P3, P4:

```
a = D.quadrangle((x1,y1,z1), (x2,y2,z2), (x3,y3,z3), (x4,y4,z4))
```

(See : Examples/geom/quadrangle.py) (See : Examples/geom/quadranglePT.py)

Create a i,j-array from a user defined parametric function:

```
a = D.surface(f, N)
```

(See : Examples/geom/surface.py) (See : Examples/geom/surfacePT.py)

Create i-arrays describing a given text, offset is the space between letters, font is the font name ('text1'), smooth is an integer indicating letter smoothness (0-4):

```
A = D.text1D(text, font, smooth, offset)
```

(See : Examples/geom/text1D.py)

Create TRI-array describing given text in 2D:

```
a = D.text2D(text, font, smooth, offset)
```

(See : Examples/geom/text2D.py)

Create TRI-array describing given text in 3D:

```
a = D.text3D(text, style, smooth, offset)
```

(See : Examples/geom/text3D.py)

1.3 Simple operations on geometries

Add a separation line defined in b to a. b and a must have a common point. Return a list of arrays:

```
A = T.addSeparationLine(a, b)
```

(See : Examples/geom/addSeparationLine.py) (See : Examples/geom/addSeparationLinePT.py)

1.4 Surface mesh from curve mesh

Create a surface geometry by driving an i-array with an i-array curve. The initial i-array is in m, the driving curve is c, the resulting i,j-array is in a:

```
a = D.lineGenerate(m, c)
```

(See : Examples/geom/lineGenerate.py) (See : Examples/geom/lineGeneratePT.py)

Create an axisymmetric mesh from one of its borders. The input array a can be an (i,j) structured array, a "TRI" or "QUAD" unstructured array. Output array is respectively an (i,j,k) array, a "PENTA" or an "HEXA" array. Rotation center and axis are (xo,yo,zo) and (nx,ny,nz), teta is the azimuthal sector, nteta is the number of discretisation points in the azimuthal direction:

```
b = D.axisym(a, (xo,yo,zo), (nx,ny,nz), teta, Nteta )
```

(See : Examples/geom/axisym.py) (See : Examples/geom/axisymPT.py)

Create a volume mesh from cross-section curves in the (x,y) plane. One curve (defined by a BAR-array) per cross-section is required. C must be a list of those curves corresponding to the different cross-sections of the solid body at different elevation z. a is the volume of the solid body:

```
a = D.volumeFromCrossSections(C)
```

(See : Examples/geom/volumeFromCrossSections.py) (See : Examples/geom/volumeFromCrossSectionsPT.py)

1.5 Information on geometries

Return the length of an i-array or a BAR-array:

```
l = D.getLength(a) .or: l = getLength(A)
```

In the case of a list of meshes, the total length is returned.

(See : Examples/geom/getLength.py) (See : Examples/geom/getLengthPT.py)

Return the index (index) of the point distant from point of index ind of a distance l in an i-array. Index starts at 1 and index ind:

```
index = D.getDistantIndex(a, ind, l)
```

(See : Examples/geom/getDistantIndex.py) (See : Examples/geom/getDistantIndexPT.py)

Return the curvilinear abscissa for all points of an i-array or a BAR-array:

```
b = D.getCurvilinearAbscissa(a)
```

(See : Examples/geom/getCurvilinearAbscissa.py) (See : Examples/geom/getCurvilinearAbscissaPT.py)

Return the curvature angle (in degrees and in [0-360]) of all points of an i-array or a BAR-array:

```
b = D.getCurvatureAngle(a) .or: B = D.getCurvatureAngle(A)
```

(See : Examples/geom/getCurvatureAngle.py) (See : Examples/geom/getCurvatureAnglePT.py)

Return the curvature radius of all the points in an i-array. This radius is signed positive for convex regions, negative for concave regions:

```
b = D.getCurvatureRadius(a) .or: B = D.getCurvatureRadius(A)
```

(See : Examples/geom/getCurvatureRadius.py) (See : Examples/geom/getCurvatureRadiusPT.py)

1.6 Example files

Example file : Examples/geom/naca.py

```
# - naca (array) -
import Geom as D
import Converter as C

a = D.naca(12.)
C.convertArrays2File([a], 'out.plt', 'bin_tp')
```

Example file : Examples/geom/nacaPT.py

```
# - naca (pyTree) -
import Geom.PyTree as D
import Converter.PyTree as C

a = D.naca(12.)
t = C.newPyTree(['Base',1]); t[1][2].append(a)
C.convertPyTree2File(t, 'out.cgns', 'bin_cgns')
```

Example file : Examples/geom/line.py

```
# - line (array) -
import Geom as D
import Converter as C

a = D.line((0,0,0), (1,0,0))
C.convertArrays2File([a], 'new.plt', 'bin_tp')
```

Example file : Examples/geom/linePT.py

```
# - line (pyTree) -
import Geom.PyTree as D
import Converter.PyTree as C

a = D.line((0,0,0), (1,0,0))
t = C.newPyTree(['Base',1]); t[1][2].append(a)
C.convertPyTree2File(t, 'out.cgns', 'bin_cgns')
```

Example file : Examples/geom/polyline.py

```
# - polyline (array) -
import Geom as D
import Converter as C

a = D.polyline([(0.,0.,0.), (1.,1.,0.), (2.,0.,0.)])
C.convertArrays2File([a], "out.plt", "bin_tp")
```

Example file : Examples/geom/polylinePT.py

```
# - polyline (pyTree) -
import Geom.PyTree as D
import Converter.PyTree as C

a = D.polyline([(0.,0.,0.), (1.,1.,0.), (2.,0.,0.)])
t = C.newPyTree(['Base',1]); t[1][2].append(a)
C.convertPyTree2File(t, "out.cgns", "bin_cgns")
```

Example file : Examples/geom/circle.py

```
# - circle (array) -
import Geom as D
import Converter as C

a = D.circle((0,0,0), 1. , 0., 360.)
C.convertArrays2File([a], "out.plt", "bin_tp")
```

Example file : Examples/geom/circlePT.py

```
# - circle (pyTree) -
import Geom.PyTree as D
import Converter.PyTree as C

a = D.circle((0,0,0), 1. , 0., 360.)
t = C.newPyTree(['Base',1]); t[1][2].append(a)
C.convertPyTree2File(t, "out.cgns", "bin_cgns")
```

Example file : Examples/geom/bezier.py

```
# - bezier (array) -
import Geom as D
import Converter as C
import Generator as G

# Bezier 1D
pts = D.polyline([(0.,0.,0.), (0.,1.,0.), (2.,1.,0.), (2.,0.,0.)],\
                 (4.,-1.,0.), (5.,6.,0.),])
a = D.bezier(pts, 100)
C.convertArrays2File([pts]+[a], 'out.plt', 'bin_tp')

# Bezier 2D
ni = 2 ; nj = 3
a = G.cart((0,0,0), (1,1,1), (ni,nj,1))
C.setValue(a, (1,1,1), [1.,1.,2.])
C.setValue(a, (1,2,1), [1.,2.,5.])
C.setValue(a, (1,3,1), [1.,3.,2.])
C.setValue(a, (2,1,1), [2.,1.,2.])
C.setValue(a, (2,2,1), [2.,2.,5.])
C.setValue(a, (2,3,1), [2.,3.,2.])
b = D.bezier(a, 10, 10)

C.convertArrays2File([a]+[b], 'out2.plt', 'bin_tp')
```

Example file : Examples/geom/bezierPT.py

```
# - bezier (pyTree) -
import Geom.PyTree as D
import Converter.PyTree as C

# Bezier 1D
pts = D.polyline([(0.,0.,0.), (0.,1.,0.), (2.,1.,0.), (2.,0.,0.)],\
                 (4.,-1.,0.), (5.,6.,0.),])
a = D.bezier(pts, 100); a[0] = 'bezier'
t = C.newPyTree(['Base',1]); t[1][2].append(a)
C.convertPyTree2File(t, "out.cgns", "bin_cgns")
```

Example file : Examples/geom/spline.py

```
# - spline (array) -
import Generator as G
import Converter as C
import Geom as D

# Spline 1D
c = D.polyline([(0.,0.,0.), (1.,1.,0.), (2.,1.,0.), \
               (3.,0.,0.), (4.,-1.,0.), (5.,6.,0.), \
               (6.,1.,0.), (7.,2.,0.), (8.,1.,0.), \
               (9.,-1.,0.), (10.,1.,0.), (11.,-1.,0.)])
d = D.spline(c,3,100)
C.convertArrays2File([c]+[d], 'out.plt', 'bin_tp')

# Spline 2D
ni = 4 ; nj = 4
a = G.cart((0,0,0), (1,1,1), (ni,nj,1))

C.setValue(a, (1,1,1), [1.,1.,2.])
C.setValue(a, (1,2,1), [1.,2.,5.])
C.setValue(a, (1,3,1), [1.,3.,5.])
C.setValue(a, (1,4,1), [1.,4.,2.])
C.setValue(a, (2,1,1), [2.,1.,2.])
C.setValue(a, (2,2,1), [2.,2.,5.])
C.setValue(a, (2,3,1), [2.,3.,5.])
C.setValue(a, (2,4,1), [2.,4.,2.])
C.setValue(a, (3,1,1), [3.,1.,2.])
C.setValue(a, (3,2,1), [3.,2.,5.])
C.setValue(a, (3,3,1), [3.,3.,5.])
C.setValue(a, (3,4,1), [3.,4.,2.])
C.setValue(a, (4,1,1), [4.,1.,2.])
C.setValue(a, (4,2,1), [4.,2.,5.])
C.setValue(a, (4,3,1), [4.,3.,5.])
C.setValue(a, (4,4,1), [4.,4.,2.])

b = D.spline(a, 4, 30, 30)

C.convertArrays2File([a]+[b], 'out2.plt', 'bin_tp')
```

Example file : Examples/geom/splinePT.py

```
# - spline (pyTree) -
import Generator.PyTree as G
import Converter.PyTree as C
import Geom.PyTree as D

# Spline 1D
c = D.polyline([(0.,0.,0.), (1.,1.,0.), (2.,1.,0.), \
               (3.,0.,0.), (4.,-1.,0.), (5.,6.,0.), \
               (6.,1.,0.), (7.,2.,0.), (8.,1.,0.), \
               (9.,-1.,0.), (10.,1.,0.), (11.,-1.,0.)])
d = D.spline(c,3,100); d[0] = 'spline'
t = C.newPyTree(['Base',1]); t[1][2].append(d)
C.convertPyTree2File(t, "out.cgns", "bin_cgns")
```

Example file : Examples/geom/nurbs.py

```
# - nurbs (array) -
import Geom as D
import Converter as C
import Generator as G

a = D.polyline([(4.1,0.1,1.1),(1.1,0.2,1.2),(1.1,1.3,1.3),(1.1,1.5,1.4),(4.5,2.5,1.5),(5.6,1.5,1.6),(6.7,1.6)])
```

```

w = C.array('W', 10, 1, 1)
w[1][:] = 1
b = D.nurbs(a, w, 4, 2000)
C.convertArrays2File([a,b], 'out.plt', 'bin_tp')

ni = 10 ; nj = 10
a = G.cart((0,0,0), (1,1,1), (ni,nj,1))

C.setValue(a, (1,1,1), [1.,1.,1.])
C.setValue(a, (1,2,1), [1.,2.,1.])
C.setValue(a, (1,3,1), [1.,3.,1.])
C.setValue(a, (1,4,1), [1.,4.,1.])
C.setValue(a, (2,1,1), [2.,1.,2.])
C.setValue(a, (2,2,1), [2.,2.,5.])
C.setValue(a, (2,3,1), [2.,3.,5.])
C.setValue(a, (2,4,1), [2.,4.,2.])
C.setValue(a, (3,1,1), [3.,1.,2.])
C.setValue(a, (3,2,1), [3.,2.,5.])
C.setValue(a, (3,3,1), [3.,3.,12.])
C.setValue(a, (3,4,1), [3.,4.,2.])
C.setValue(a, (4,1,1), [4.,1.,2.])
C.setValue(a, (4,2,1), [4.,2.,5.])
C.setValue(a, (4,3,1), [4.,3.,5.])
C.setValue(a, (4,4,1), [4.,4.,2.])
C.setValue(a, (6,8,1), [4.,6.,14.])
C.setValue(a, (8,6,1), [4.,6.,-4.])

w = C.array('W', 10, 10, 1)

w[1][0][:] = 1
w[1][0][7] = 7
w[1][0][15] = 9

d = D.nurbs(a,w,4,100,100)

C.convertArrays2File([a]+[d], 'out2.plt', 'bin_tp')

```

Example file : Examples/geom/nurbsPT.py

```

# - nurbs (pyTree) -
import Geom.PyTree as D
import Converter.PyTree as C
import Generator.PyTree as G

ni = 10 ; nj = 10
a = G.cart((0,0,0), (1,1,1), (ni,nj,1));
w = G.cart((0,0,0), (1,1,1), (ni,nj,1)); w[0]='weight'
w[2][1][2][0][1][:] = 1
w[2][1][2][0][1][7] = 7;
w[2][1][2][0][1][9] = 9;

d = D.nurbs(a,w,4,100,100); d[0] = 'nurbs'
t = C.newPyTree(['Base',2]); t[1][2].append(d)
C.convertPyTree2File(t, "out.cgns", "bin_cgns")

```

Example file : Examples/geom/curve.py

```

# - curve (array) -
import Converter as C
import Geom as D

# User definition of parametric curve
def f(t):
    x = t
    y = t*t+1

```

```

    z = 0.
    return (x,y,z)

# Array definition of geometry
a = D.curve(f)
C.convertArrays2File([a], "new.plt", "bin_tp")

```

Example file : Examples/geom/curvePT.py

```

# - curve (pyTree) -
import Converter.PyTree as C
import Geom.PyTree as D

# User definition of parametric curve
def f(t):
    x = t
    y = t*t+1
    z = 0.
    return (x,y,z)

a = D.curve(f)
t = C.newPyTree(['Base',1]); t[1][2].append(a)
C.convertPyTree2File(t, "out.cgns", "bin_cgns")

```

Example file : Examples/geom/cone.py

```

# - cone (array) -
import Geom as D
import Converter as C

a = D.cone((0,0,0), 1. , 0.5, 1.)
C.convertArrays2File([a], "out.plt", "bin_tp")

```

Example file : Examples/geom/conePT.py

```

# - cone (pyTree) -
import Geom.PyTree as D
import Converter.PyTree as C

a = D.cone((0,0,0), 1. , 0.5, 1.)
t = C.newPyTree(['Base',2]); t[1][2].append(a)
C.convertPyTree2File(t, "out.cgns", "bin_cgns")

```

Example file : Examples/geom/sphere.py

```

# - sphere (array) -
import Geom as D
import Converter as C

a = D.sphere((0,0,0), 1., 20)
C.convertArrays2File([a], "out.plt", "bin_tp")

```

Example file : Examples/geom/spherePT.py

```

# - sphere (pyTree) -
import Geom.PyTree as D
import Converter.PyTree as C

a = D.sphere((0,0,0), 1., 20)
t = C.newPyTree(['Base',2]); t[1][2].append(a)
C.convertPyTree2File(t, "out.cgns", "bin_cgns")

```

Example file : Examples/geom/sphere6.py


```
# - sphere6 (array) -
import Geom as D
import Converter as C
```

```
a = D.sphere6((0,0,0), 1., 20)
C.convertArrays2File(a, "out.plt", "bin_tp")
```

Example file : Examples/geom/sphere6PT.py

```
# - sphere6 (pyTree) -
import Geom.PyTree as D
import Converter.PyTree as C
```

```
A = D.sphere6((0,0,0), 1., 20)
t = C.newPyTree(['Base',2]); t[1][2] = t[1][2] + A
C.convertPyTree2File(t, "out.cgns", "bin_cgns")
```

Example file : Examples/geom/triangle.py

```
# - triangle (array) -
import Geom as D
import Converter as C
```

```
a = D.triangle((0,0,0), (0.1,0.,0.1), (0.05, 0.08, 0.1))
C.convertArrays2File([a], "out.plt", "bin_tp")
```

Example file : Examples/geom/trianglePT.py

```
# - triangle (pyTree) -
import Geom.PyTree as D
import Converter.PyTree as C
```

```
a = D.triangle((0,0,0), (0.1,0.,0.1), (0.05, 0.08, 0.1))
t = C.newPyTree(['Base',2]); t[1][2].append(a)
C.convertPyTree2File(t, "out.cgns", "bin_cgns")
```

Example file : Examples/geom/quadrangle.py

```
# - quadrangle (array) -
import Geom as D
import Converter as C
```

```
a = D.quadrangle((0,0,0.1), (0.1,0.,0.1), (0.05, 0.08, 0.1), (0.02,0.05,0.1))
C.convertArrays2File([a], "out.plt", "bin_tp")
```

Example file : Examples/geom/quadranglePT.py

```
# - quadrangle (PyTree) -
import Geom.PyTree as D
import Converter.PyTree as C
```

```
a = D.quadrangle((0,0,0.1), (0.1,0.,0.1), (0.05, 0.08, 0.1), (0.02,0.05,0.1))
t = C.newPyTree(['Base',2]); t[1][2].append(a)
C.convertPyTree2File(t, "out.cgns", "bin_cgns")
```

Example file : Examples/geom/surface.py

```
# - surface (array) -
import Converter as C
import Geom as D
```

```
# User definition of parametric curve
def f(t,u):
    x = t+u
    y = t*t+1+u*u
    z = u
    return (x,y,z)
```

```
# Array definition of geometry
a = D.surface(f)
C.convertArrays2File([a], "new.plt", "bin_tp")
```

Example file : Examples/geom/surfacePT.py

```
# - surface (PyTree) -
import Converter.PyTree as C
import Geom.PyTree as D

# User definition of parametric curve
def f(t,u):
    x = t+u
    y = t*t+1+u*u
    z = u
    return (x,y,z)

a = D.surface(f)
t = C.newPyTree(['Base',2]); t[1][2].append(a)
C.convertPyTree2File(t, "out.cgns", "bin_cgns")
```

Example file : Examples/geom/text1D.py

```
# - text1D (array) -
import Geom as D
import Converter as C

a = D.text1D("ABCDEFGHIJKLMNOPQRSTUVWXYZ")
b = D.text1D("A0123456789")
c = D.text1D("Aabcdefghijklmnopqrstuvwxy");
d = D.text1D("ABCDEFGHIJKLMNOPQRSTUVWXYZ", smooth=4, offset=1.)
e = D.text1D("A.1+2=3. 2-1=1.")

C.convertArrays2File(e, 'out.plt', 'bin_tp')
```

Example file : Examples/geom/text2D.py

```
# - text2D (array) -
import Geom as D
import Converter as C

a = D.text2D("ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789", smooth=0, offset=1.)
C.convertArrays2File([a], 'out.plt', 'bin_tp')
```

Example file : Examples/geom/text3D.py

```
# - text3D (array) -
import Geom as D
import Converter as C

a = D.text3D("CASSIOPEE", smooth=1)
C.convertArrays2File([a], 'out.plt', 'bin_tp')
```

Example file : Examples/geom/addSeparationLine.py

```
# - addSeparationLine (array)-
import Geom as D
import Converter as C

# Create a cartesian mesh
a1 = D.circle((0,0,0), 1, 0., 360, 1000)
a2 = D.line((0.,1.,0.), (0.,2.,0), 100)
arrays = D.addSeparationLine(a1, a2)
C.convertArrays2File(arrays, "out.plt", "bin_tp")
```

Example file : Examples/geom/addSeparationLinePT.py

```

# - addSeparationLine (pyTree)-
import Geom.PyTree as D
import Converter.PyTree as C

a1 = D.circle((0,0,0), 1, 0., 360, 1000)
a2 = D.line((0.,1.,0.), (0.,2.,0), 100)
zones = D.addSeparationLine(a1, a2)
t = C.newPyTree(['Base',1])
t[1][2] = t[1][2] + zones
C.convertPyTree2File(t, "out.cgns", "bin_cgns")

```

Example file : Examples/geom/lineGenerate.py

```

# - lineGenerate (array) -
import Geom as D
import Converter as C

a = D.naca(12.)
b = D.line((0,0,0), (0.,0.,1.))
c = D.lineGenerate(a, b)
C.convertArrays2File([c], 'out.plt', 'bin_tp')

```

Example file : Examples/geom/lineGeneratePT.py

```

# - lineGenerate (pyTree)-
import Geom.PyTree as D
import Converter.PyTree as C

a = D.naca(12.)
l = D.line((0,0,0), (0,0.,1.))
a = D.lineGenerate(a, l)
t = C.newPyTree(['Base',2]); t[1][2].append(a)
C.convertPyTree2File(t, 'out.cgns', 'bin_cgns')

```

Example file : Examples/geom/axisym.py

```

# - axisym (array) -
import Generator as G
import Converter as C
import Transform as T
import Geom as D

a0 = G.cart((0.,0.,0.), (0.1,0.1,0.2),(10,10,1))
a1 = D.axisym(a0,(1.,0.,0.), (0.,1.,0.),30.,2)
C.convertArrays2File([a1], "out1.plt", "bin_tp")

a = C.convertStruct2Tetra(a0)
a1 = D.axisym(a,(1.,0.,0.), (0.,1.,0.),30.,30)
C.convertArrays2File([a1], "out2.plt", "bin_tp")

```

```

# QUAD
a = C.convertStruct2Hexa(a0)
a = C.addVars(a,'cellN')
a = C.initVars(a,'cellN',1.)
ax1 = D.axisym(a,(1.,0.,0.), (0.,1.,0.),30.,20)
C.convertArrays2File([ax1], "out3.plt", "bin_tp")

```

Example file : Examples/geom/axisymPT.py

```

# - axisym (pyTree) -
import Generator.PyTree as G
import Converter.PyTree as C
import Geom.PyTree as D

a = G.cart((0.,0.,0.), (0.1,0.1,0.2),(10,10,1))
a = D.axisym(a,(1.,0.,0.), (0.,1.,0.),30.,2)
t = C.newPyTree(['Base',3]); t[1][2].append(a)
C.convertPyTree2File(t, "out.cgns", "bin_cgns")

```

Example file : Examples/geom/volumeFromCrossSections.py

```
# - volumeFromCrossSections (array) -
import Converter as C
import Generator as G
import Transform as T
import Geom as D
import Post as P

a = C.convertFile2Arrays( 'Data/cube.fig', 'fmt_xfig' )
a = T.symetrize(a, (0,0,0), (1,0,0), (0,0,1))

c = 0
contours = {}
for i in a:
    b = C.convertArray2Tetra(i)
    z = C.getValue(b, 0)[2]
    if (z not in contours):
        contours[z] = b
    else:
        contours[z] = T.join(contours[z], b)

keys = contours.keys()
contour = []
for i in keys:
    b = contours[i]
    b = G.close(b)
    contour.append(b)

vol = D.volumeFromCrossSections(contour)
C.convertArrays2File([vol], 'out.plt', 'bin_tp')
```

Example file : Examples/geom/volumeFromCrossSectionsPT.py

```
# - volumeFromCrossSection (pyTree) -
import Converter.PyTree as C
import Geom.PyTree as D
import Generator.PyTree as G
import Transform.PyTree as T

t = C.convertFile2PyTree('Data/cube.fig', 'fmt_xfig')
t = T.symetrize(t, (0,0,0), (1,0,0), (0,0,1))
c = 0
contours = {}
for i in t[1][2]:
    b = C.convertArray2Tetra(i)
    z = C.getValue(b, 'CoordinateZ',0)
    if (z not in contours):
        contours[z] = b
    else:
        contours[z] = T.join(contours[z], b)

keys = contours.keys()
contour = C.newPyTree(['Base',1])
for i in keys:
    b = contours[i]
    b = G.close(b)
    contour[1][2].append(b)

zone = D.volumeFromCrossSections(contour)
t = C.newPyTree(['Base',3]); t[1][2].append(zone)
C.convertPyTree2File(t,"out.cgns","bin_cgns")
```

Example file : Examples/geom/getLength.py

```
# - getLength (array) -
import Geom as D
```

```
a = D.line((0,0,0), (1,0,0))
print D.getLength(a)
```

Example file : Examples/geom/getLengthPT.py

```
# - getLength (pyTree)-
import Geom.PyTree as D
```

```
a = D.line((0,0,0), (1,0,0))
print D.getLength(a)
```

Example file : Examples/geom/getDistantIndex.py

```
# - getDistantIndex (array) -
import Geom as D
```

```
a = D.line((0.,0.,0.), (1.,0.,0), 100)
print 'distant Index : ', D.getDistantIndex(a,25,0.2)
```

Example file : Examples/geom/getDistantIndexPT.py

```
# - getDistantIndex (pyTree)-
import Geom.PyTree as D
```

```
a = D.line((0.,0.,0.), (1.,0.,0), 100)
print 'distant Index : ', D.getDistantIndex(a,25,0.2)
```

Example file : Examples/geom/getCurvilinearAbscissa.py

```
# - getCurvilinearAbscissa (array) -
import Converter as C
import Geom as D
import Transform as T
```

```
a = D.line((0.,0.,0.), (1.,0.,0), 100)
a2 = D.line((1.,0.,0.), (1.,1,0), 100)
a = T.join (a, a2)
a3 = D.getCurvilinearAbscissa( a )
a = C.addVars([a, a3])
C.convertArrays2File([a], "out.plt", "bin_tp")
```

Example file : Examples/geom/getCurvilinearAbscissaPT.py

```
# - getCurvilinearAbscissa (pyTree)-
import Converter.PyTree as C
import Geom.PyTree as D
```

```
a = D.line((0.,0.,0.), (1.,0.,0), 100)
a = D.getCurvilinearAbscissa( a )
t = C.newPyTree(['Base',1]); t[1][2].append(a)
C.convertPyTree2File(t,"out.cgns","bin_cgns")
```

Example file : Examples/geom/getCurvatureAngle.py

```
# - getCurvatureAngle (array) -
import Converter as C
import Geom as D
import Transform as T
```

```
a1 = D.line((0.,0.,0.), (1.,0.,0), 100)
a2 = D.line((1.,0.,0.), (1.,1,0), 100)
a = T.join (a1, a2)
a3 = D.getCurvatureAngle( a )
a = C.addVars([a, a3])
C.convertArrays2File([a], "out.plt", "bin_tp")
```

Example file : Examples/geom/getCurvatureAnglePT.py

```
# - getCurvatureAngle (pyTree) -
import Converter.PyTree as C
import Geom.PyTree as D

a = D.polyline([(0.,0.,0.), (1.,1.,0.), (2.,0.,0.)])
a = D.getCurvatureAngle( a )
t = C.newPyTree(['Base',1]); t[1][2].append(a)
C.convertPyTree2File(t, "out.cgns", "bin_cgns")
```

Example file : Examples/geom/getCurvatureRadius.py

```
# - getCurvatureRadius (array) -
import Geom as D

# ligne : courbure infinie
print 'Droite : '
a = D.line((0,0,0), (1,0,0), 3)
rad = D.getCurvatureRadius(a) ; print rad

# cercle
print 'Cercle : '
a = D.circle((0,0,0), 1, 10, 0, 10)
rad = D.getCurvatureRadius(a) ; print rad

# bezier
print 'Bezier : '
pts = D.polyline([(6,0.01,1), (5.4,0.036,1), (4.8,0.064,1), (2.5,0.21,1),
                 (0.3,0.26,1), (0,0.047,1), (0,0,0)])
a = D.bezier( pts, 100 )
rad = D.getCurvatureRadius(a) ; print rad
```

Example file : Examples/geom/getCurvatureRadiusPT.py

```
# - getCurvatureRadius (pyTree) -
import Geom.PyTree as D
import Converter.PyTree as C

a = D.circle((0,0,0), 1, 10, 0, 10)
a = D.getCurvatureRadius(a)
t = C.newPyTree(['Base',1]); t[1][2].append(a)
C.convertPyTree2File(t, "out.cgns", "bin_cgns")
```