Contenido

[Procedimiento para crear geometría GTS 1](#_Toc99281447)

[Ejemplo 1: 2](#_Toc99281448)

[1-Tamaño de malla 2](#_Toc99281449)

[2-Vertices 2](#_Toc99281450)

[3-Aristas 2](#_Toc99281451)

[4-Caras 2](#_Toc99281452)

[Script completo: 2](#_Toc99281453)

[Procedimiento para visualizar geometría en Yade. 3](#_Toc99281454)

[Paquete 1 3](#_Toc99281455)

[Paquete 2 3](#_Toc99281456)

[Transformación archivo. geo a archivo.stl 3](#_Toc99281457)

[Transformación de archivo.stl a archivo.gts 4](#_Toc99281458)

[Calibración archivo gts 4](#_Toc99281459)

[Visualización de geometría en yade 5](#_Toc99281460)

[Ejemplo 2 6](#_Toc99281461)

[Visualización de geometría ejemplo 2 en yade 9](#_Toc99281462)

# Procedimiento para crear geometría GTS

A continuación, se describe el procedimiento para crear una geometría GTS para posteriormente ejecutarla en yade.

El procedimiento es sencillo, pero es necesario seguir ciertos ya que comenzamos creando una geometría en un archivo con extensión geo , luego se transforma el archivo a extensión.stl y finalmente a extensión.gts los pasos para este procedimiento se describen continuación:

1-Definir grosor de malla.

2- Definir ubicación espacial de cada vértice que compone la figura a crear.

3-Crear aristas uniendo los vértices creados en el paso anterior, es necesario que los vértices sigan una orientación, esto es unir los vértices de forma tal de crear una línea continua, esto se logra creando todas las aristas en ya sea en sentido horario o antihorario, sino se sigue esta lógica las caras de la figura que queremos crear no quedara bien definida y provocara errores.

4-Crear caras a partir de las líneas anteriores.

## Ejemplo 1:

### 1-Tamaño de malla

En este caso de define como: tm = 2/100

### 2-Vertices

Gráfico

Descripción generada automáticamenteSe creará un cuadrado de lado 1, por lo tanto, es necesario hacer 4 vértices.

Point(1)={0,0,0,tm};

Point(2)={1,0,0,tm};

Point(3)={1,1,0,tm};

Point(4)={0,1,0,tm};

### 3-Aristas

Se crear las aristas siguiendo un sentido antihorario a la hora de unir los vértices.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteLine(1) = {1,2};

Line(2) = {2,3};

Line(3) = {3,4};

Line(4) = {4,1};

### 4-Caras

Se crean las caras considerando que aristas componen la cara que se quiere visualizar.

Line Loop(1) = {1,2,3,4};

Plane Surface(1) = {1};

Finalmente el script se debe guardar en un archivo de texto y a la hora de guardarlo cambiar su extensión a .geo , luego se recomienda descargar gmsh para visualizar la figura (<http://gmsh.info//>).

### Script completo:

tm = 2/100;

Point(1)={0,0,0,tm};

Point(2)={1,0,0,tm};

Point(3)={1,1,0,tm};

Point(4)={0,1,0,tm};

Line(1) = {1,2};

Line(2) = {2,3};

Line(3) = {3,4};

Line(4) = {4,1};

Line Loop(1) = {1,2,3,4};

Plane Surface(1) = {1};

Visualización en Gmsh:

Imagen que contiene Gráfico

Descripción generada automáticamente

Visualizando la figura nos aseguramos de que la geometría que exportaremos para trabajar en yade será la correcta.

## Procedimiento para visualizar geometría en Yade.

En yade no es posible visualizar directamente un archivo de extensión .geo como el que creamos anteriormente por lo que es necesario transformar el formato, para esto será necesario ejecutar antes en yade los dos comandos que se muestran a continuación cada uno por separado, los cuales instalaran los paquetes necesarios.

### Paquete 1

sudo apt install gmsh

### Paquete 2

sudo apt install libgts-bin

## Transformación archivo. geo a archivo.stl

Instalados los paquetes anteriores se procede a la transformación del ejemplo creado en este caso se nombro el archivo como “ejemplo.geo”, para cambiar el formato simplemente se ejecuta el siguiente comando en la terminal de yade correspondiente a la carpeta que contiene tal ejemplo.

gmsh -2 ejemplo.geo -o ejemplo.stl

Creándose así un nuevo archivo de nombre ejemplo.stl.

## Transformación de archivo.stl a archivo.gts

Similar al paso anterior ejecutamos en la terminal de la carpeta que contiene el ejemplo el siguiente comando:

stl2gts -r < ejemplo.stl > ejemplo.gts

## Calibración archivo gts

Es posible que la orientación de las superficies creadas no sea la correcta esto se corrige con el siguiente scrip, el que se debe guardar con extencion .py en la carpeta que contiene el ejemplo y posteriormente ejecutarlo en la terminal.

with open("ejemplo.gts") as f:

lines = f.readlines()

flag = 0

for i,l in enumerate(lines):

ls = l.split()

if flag == 0 and len(ls)==2:

flag = 1

if flag == 1 and len(ls)==3:

flag = 2

if flag != 2:

continue

lines[i] = "{} {} {}\n".format(ls[0],ls[2],ls[1])

with open("ejemplo.gts","w") as f:

f.writelines(lines)

### Visualización de geometría en yade

Finalmente es posible visualizar en Yade la geometría creada con el siguiente script que debe ser guardado con extencion.py y ejecutado en la temrinal.

from \_\_future\_\_ import print\_function

from yade import pack

import gts, os.path, locale

locale.setlocale(

locale.LC\_ALL, 'en\_US.UTF-8'

) #gts is locale-dependend. If, for example, german locale is used, gts.read()-function does not import floats normally

###################################################################

if not os.path.exists('ejemplo2.coarse.gts'):

if os.path.exists('ejemplo2.gts'):

surf = gts.read(open('ejemplo2.gts'))

surf.coarsen(100)

surf.write(open('ejemplo2.coarse.gts', 'w'))

else:

print(

"""ejemplo2.gts not found, you need to download input data:

wget http://gts.sourceforge.net/samples/ejemplo2.gts.gz

gunzip ejemplo2.gts.gz

"""

)

quit()

surf = gts.read(open('ejemplo2.coarse.gts'))

O.bodies.append(pack.gtsSurface2Facets(surf, wire=True))

#### Figura

Ejecutando el scrip anterior se obtiene la figura creada:

Imagen que contiene Polígono

Descripción generada automáticamente

## Ejemplo 2

El siguiente script muestra la creación de una nueva figura(.geo) pero esta vez mas compleja :

tm = 2/100;

Point(1)={0,0,0,tm};

Point(2)={0,0,1,tm};

Point(3)={2.5,0,1,tm};

Point(4)={3,0,1,tm};

Point(5)={5.5,0,1,tm};

Point(6)={5.5,0,0,tm};

Point(7)={3,0,0,tm};

Point(8)={2.5,0,0,tm};

Point(9)={0,1,0,tm};

Point(10)={0,1,1,tm};

Point(11)={2.5,1,1,tm};

Point(12)={3,0.5,1,tm};

Point(13)={5.5,0.5,1,tm};

Point(14)={5.5,0.5,0,tm};

Point(15)={3,0.5,0,tm};

Point(16)={2.5,1,0,tm};

Line(1) = {1,2};

Line(2) = {2,3};

Line(3) = {3,4};

Line(4) = {4,5};

Line(5) = {5,6};

Line(6) = {6,7};

Line(7) = {7,8};

Line(8) = {8,1};

Line(9) = {9,10};

Line(10) = {10,11};

Line(11) = {11,12};

Line(12) = {12,13};

Line(13) = {13,14};

Line(14) = {14,15};

Line(15) = {15,16};

Line(16) = {16,9};

Line(17) = {10,2};

Line(18) = {11,3};

Line(19) = {12,4};

Line(20) = {13,5};

Line(21) = {14,6};

Line(22) = {15,7};

Line(23) = {16,8};

Line(24) = {9,1};

Line(25) = {11,16};

Line(26) = {12,15};

Line(27) = {3,8};

Line(28) = {4,7};

Line Loop(1) = {1, 2, 27,8};

Line Loop(2) = {-27,3,28,7};

Line Loop(3) = {-28,4,5,6};

Line Loop(4) = {9,10,25,16};

Line Loop(5) = {-25,11,26,15};

Line Loop(6) = {-26,12,13,14};

Line Loop(7) = {-13,20,5,-21};

Line Loop(8) = {9,17,-1,-24};

Line Loop(9) = {17,2,-18,-10};

Line Loop(10) = {18,3,-19,-11};

Line Loop(11) = {19,4,-20,-12};

Line Loop(12) = {24,-8,-23,16};

Line Loop(13) = {23,-7,-22,15};

Line Loop(14) = {22,-6,-21,14};

Plane Surface(1) = {1};

Plane Surface(2) = {2};

Plane Surface(3) = {3};

Plane Surface(4) = {4};

Plane Surface(5) = {5};

Plane Surface(6) = {6};

Plane Surface(7) = {7};

Plane Surface(8) = {8};

Plane Surface(9) = {9};

Plane Surface(10) = {10};

Plane Surface(11) = {11};

Plane Surface(12) = {12};

Plane Surface(13) = {13};

Plane Surface(14) = {14};

### Visualización de geometría ejemplo 2 en yade

Siguiendo el procedimiento del ejemplo 1 (archivo.geo>archivo.stl>archivo.gts) se obtiene la visualización del nuevo ejemplo 2:

Forma, Polígono

Descripción generada automáticamente