Royal Holloway University of London

PH4100: Major Project

BDSIM
&
pyg4ometry

Ben Shellswell

Abstract

Supervised by
Prof. S Boogert

January 16, 2020

Contents

1 Introduction

1.1 BDSIM

BDSIM (or Beam Delivery SIMulation) is a software package written by the John Adams institute for accelerator science (JAI), for the use of modelling particle beam interactions. BDSIM has many applications such as modelling complex particle accelerators such as the LHC and concepts magnets for MRI medical scanners.

1.2 pyg4ometry

pyg4ometry is a python packaged also generated by JAI, its purpose it to convert 3D CAD models between different representations to allow compatibility with BDSIM for the testing of new concepts. The '4' in 'pyg4ometry' comes from the consitencey the package has with Geant4 1.3.

1.3 Geant4

Geant4 (or GEometry ANd Tracking) is a software developed for the simulation and tracking of particles traveling through matter.

1.4 Project Aims

The aims of this project are to optimize the pyg4ometry package to improve and performance test the results. The main areas for improvement and where most of the computational energy in wasted is in the meshing of the primitive Geant4 solids.

2 Primitive Meshing

This section will describe the work done to optimize the python scipts that generate the three dimensional meshing for the primitive solids. All the solids used are constructed such that they are compatible with Geant4's solids.

2.1 Co-ordinate Systems

maybe append code snippet
redraw all coord systems in inkscape

2.1.1 Cylindrical Co-ordinate System

$$\begin{aligned} x &= r\cos\theta \\ y &= r\sin\theta \\ z &= z \end{aligned} \tag{1}$$

```python
for j0 in range(nslice):
    j1 = j0
    j2 = j0 + 1
```

Listing 1: Python example

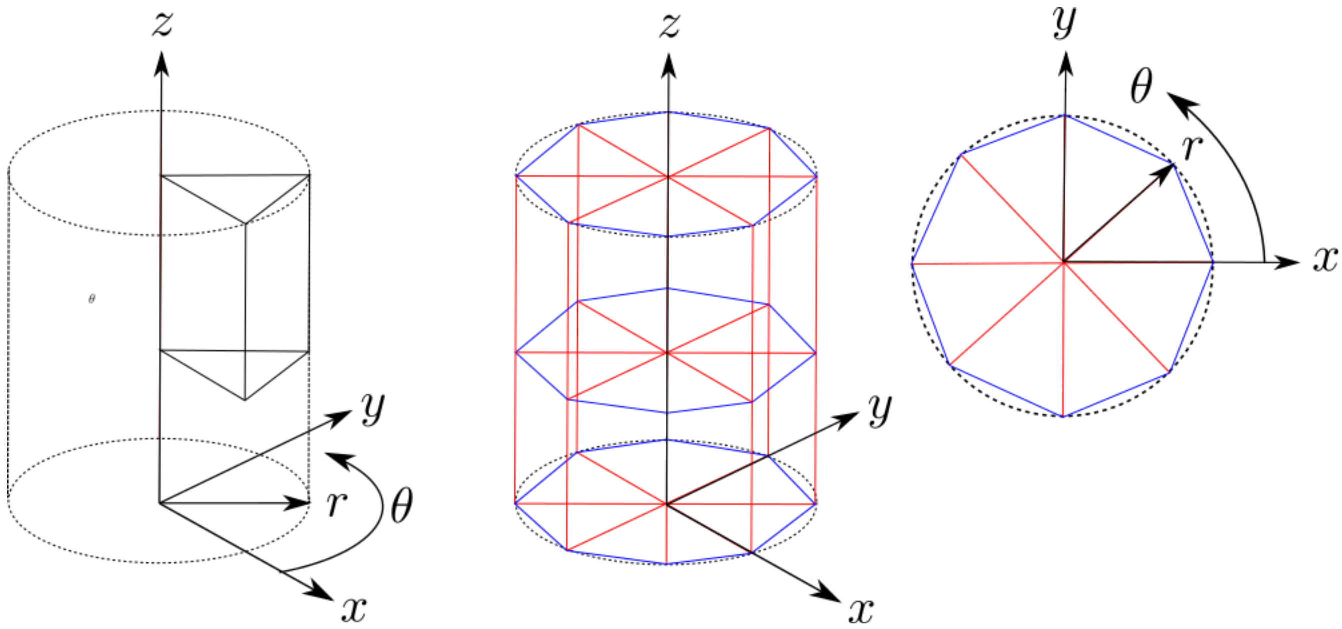

Figure 1: CylindricalCoordinate System

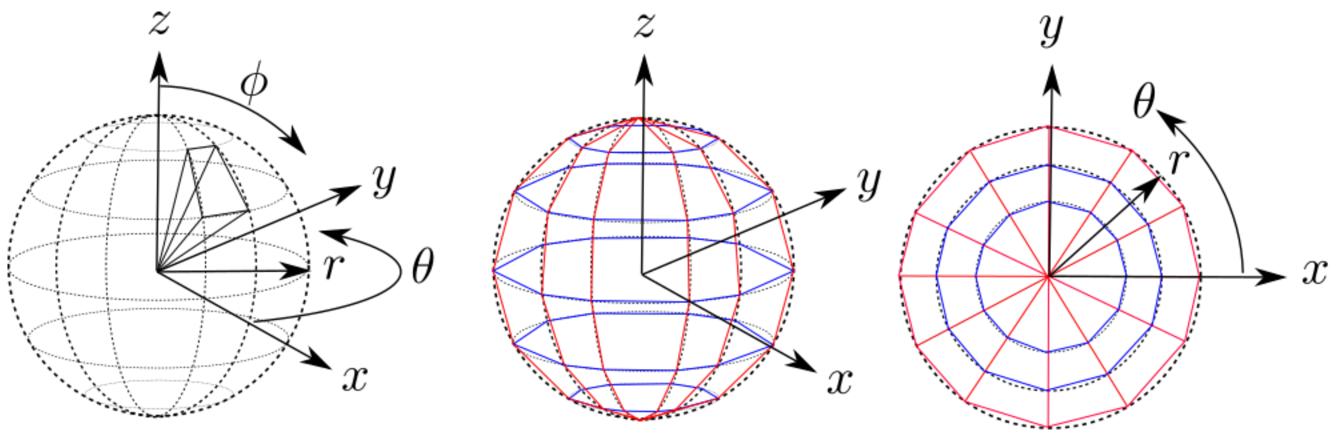

Figure 2: Spherical Coordinate System

2.1.2 Spherical Co-ordinate System

$$\begin{aligned} x &= r\cos\theta\sin\phi \\ y &= r\sin\theta\sin\phi \\ z &= z \end{aligned} \tag{2}$$

```
for j0 in range(nslice):
    j1 = j0
    j2 = j0 + 1

    for i0 in range(nstack):
        i1 = i0
        i2 = i0 + 1
```

Listing 2: Python example

2.1.3 Toroidal Co-ordinate System

$$\begin{aligned} x &= R_{Torus} + R\cos\theta\cos\phi \\ y &= R_{Torus} + R\cos\theta\sin\phi \\ z &= R\sin\theta \end{aligned} \tag{3}$$

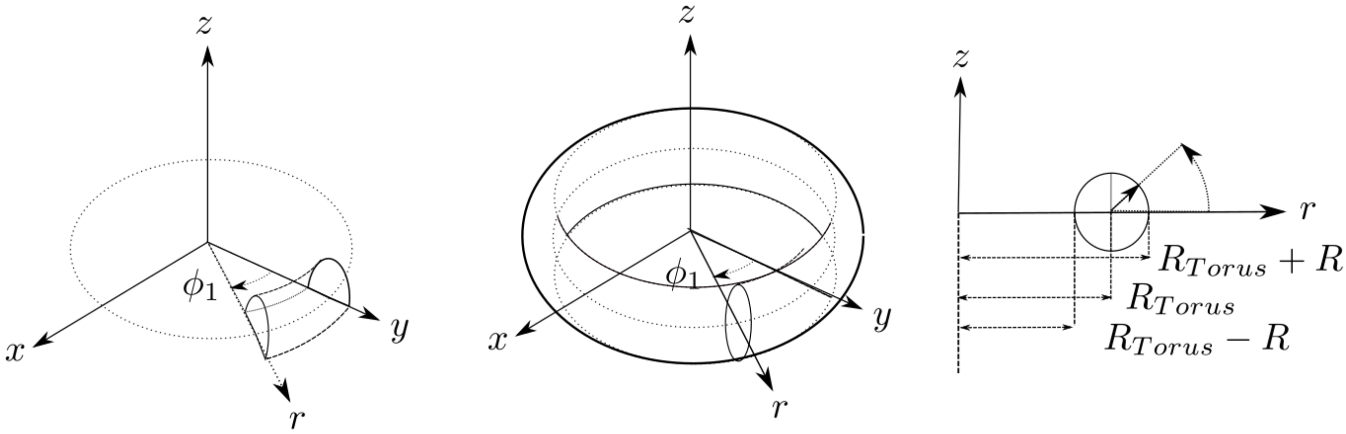

Figure 3: Toroidal Coordinate System

2.2 Meshing performance testing

2.3 Curved primitive solids

for each shape
stack and slice ?
radially mesh ?
which coord system

2.3.1 Cons

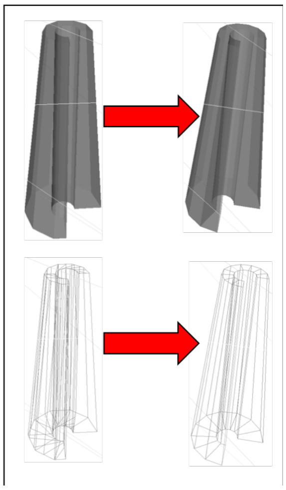

Figure 4: Toroidal Coordinate System

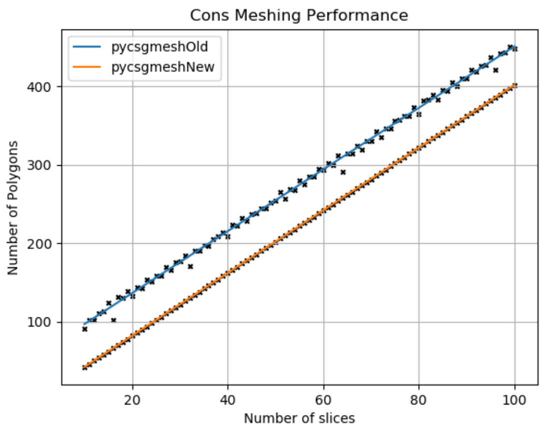

Figure 5: Spherical Coordinate System

2.3.2 CutTubs

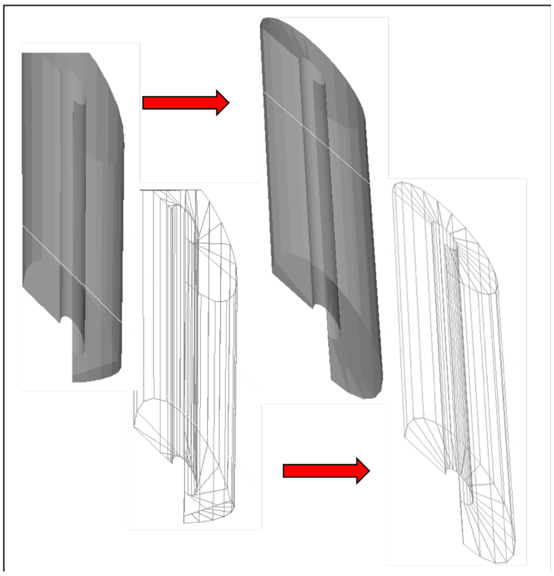

Figure 6: Toroidal Coordinate System

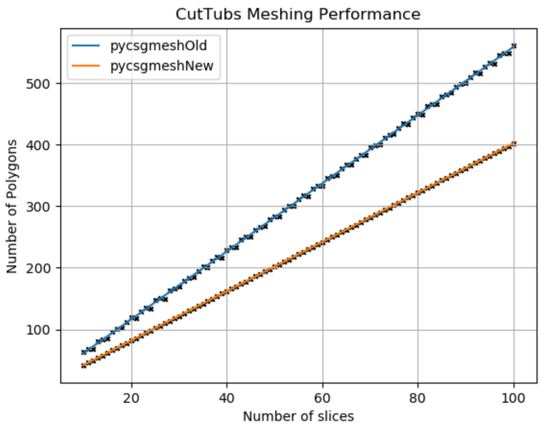

Figure 7: Spherical Coordinate System

2.3.3 Ellipsoid

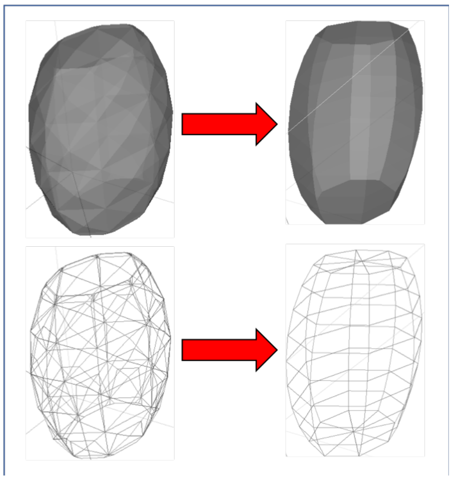

Figure 8: Toroidal Coordinate System

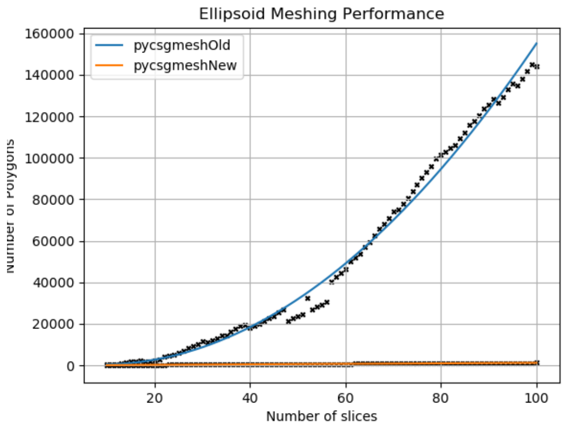

Figure 9: Spherical Coordinate System

2.3.4 EllipticalCone

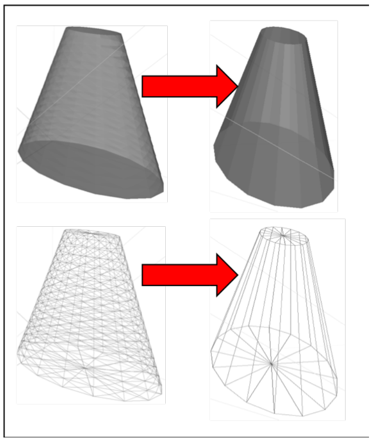

Figure 10: Toroidal Coordinate System

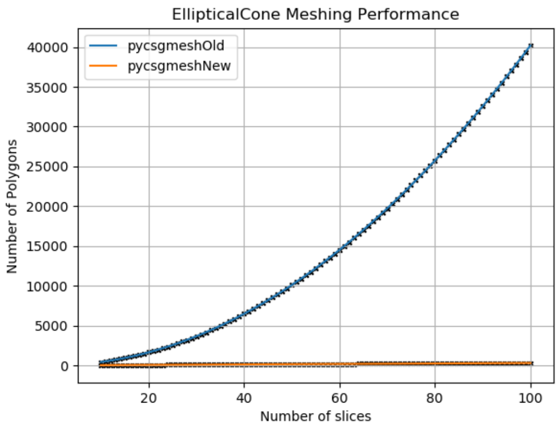

Figure 11: Spherical Coordinate System

2.3.5 EllipticalTube

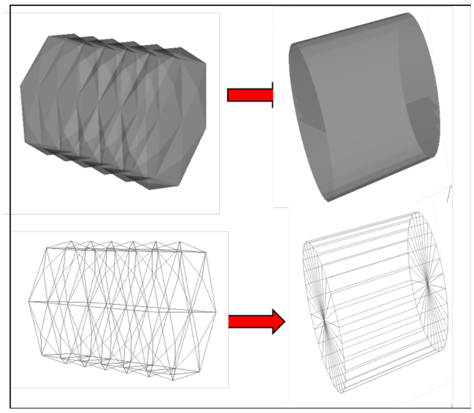

Figure 12: Toroidal Coordinate System

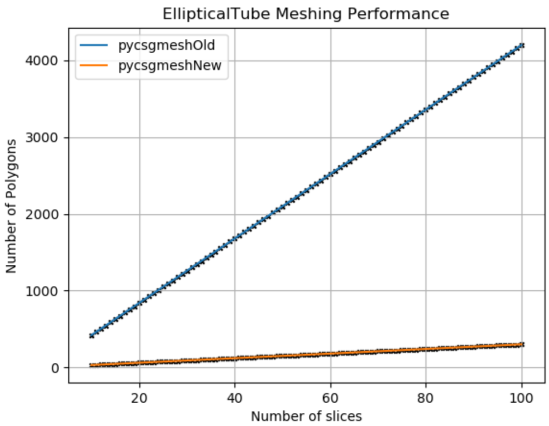

Figure 13: Spherical Coordinate System

2.3.6 Hyperboloid

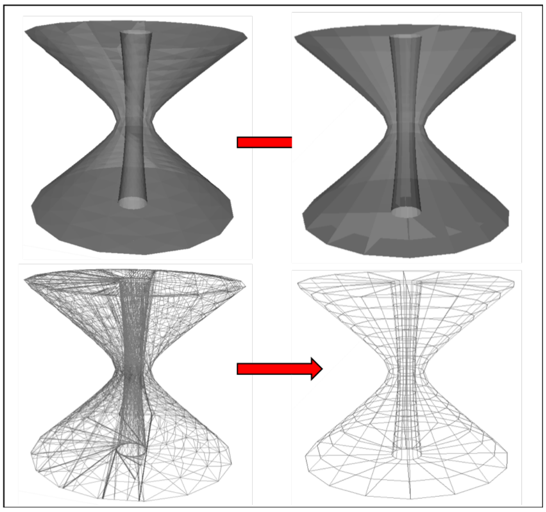

Figure 14: Toroidal Coordinate System

2.3.7 Orb

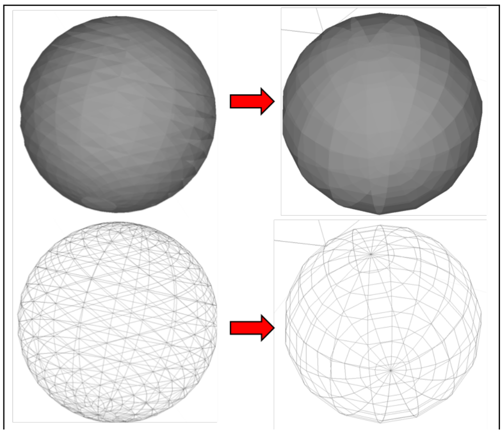

Figure 15: Toroidal Coordinate System

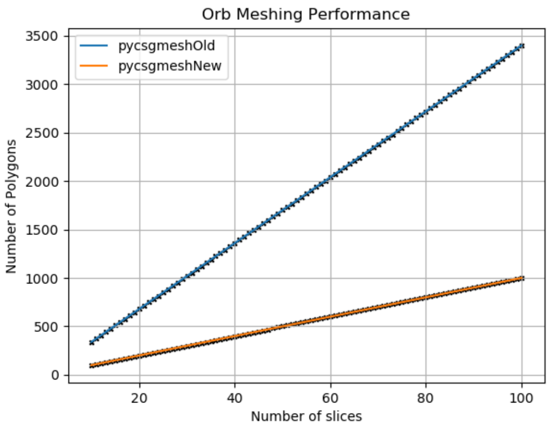

Figure 16: Spherical Coordinate System

2.3.8 Paraboloid

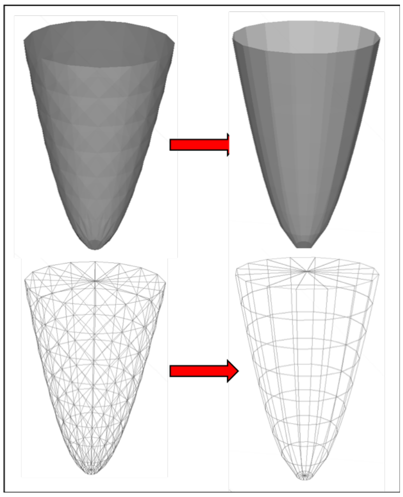

Figure 17: Toroidal Coordinate System

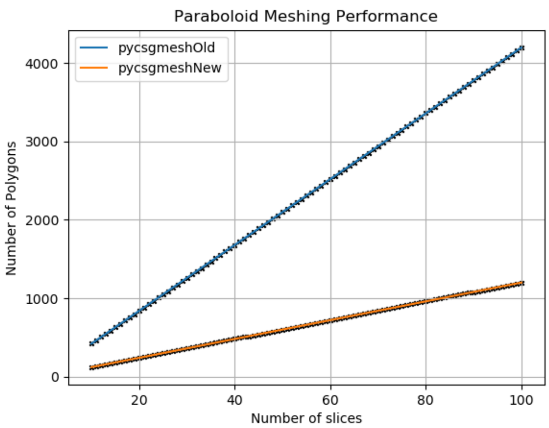

Figure 18: Spherical Coordinate System

2.3.9 Polycone

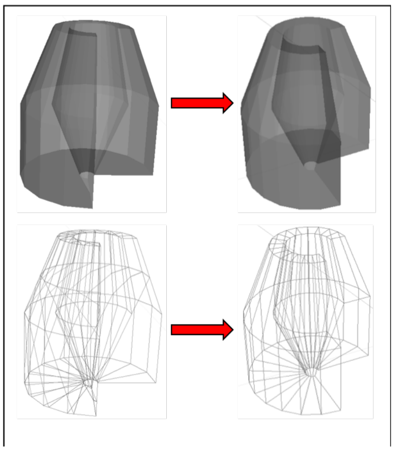

Figure 19: Toroidal Coordinate System

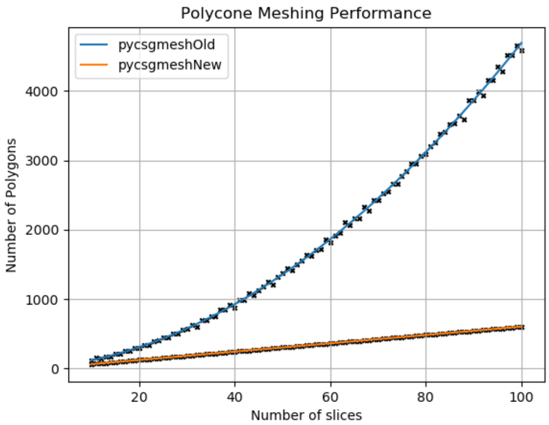

Figure 20: Spherical Coordinate System

2.3.10 Sphere

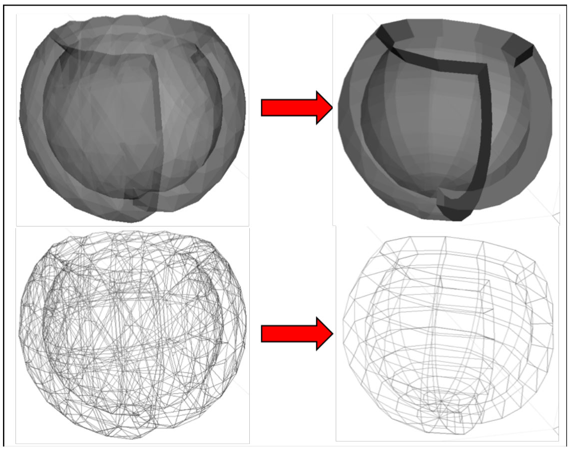

Figure 21: Toroidal Coordinate System

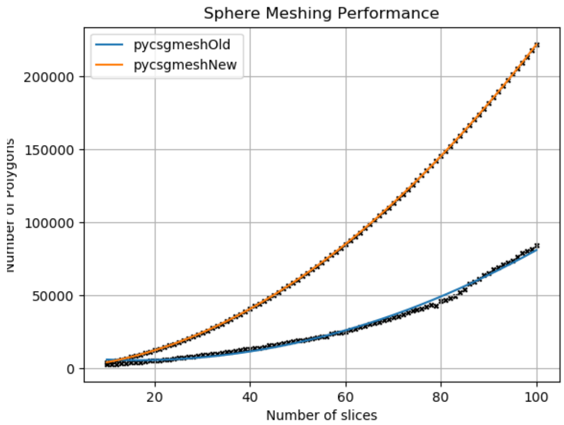

Figure 22: Spherical Coordinate System

2.3.11 Torus

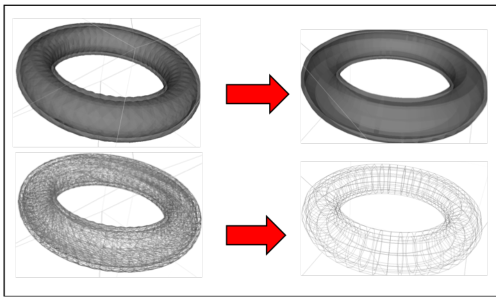

Figure 23: Toroidal Coordinate System

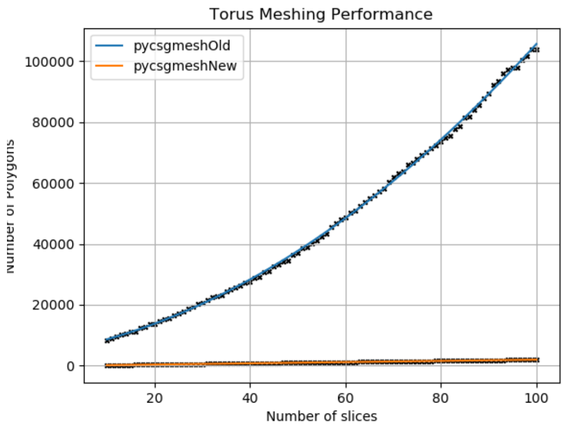

Figure 24: Spherical Coordinate System

2.3.12 Tubs

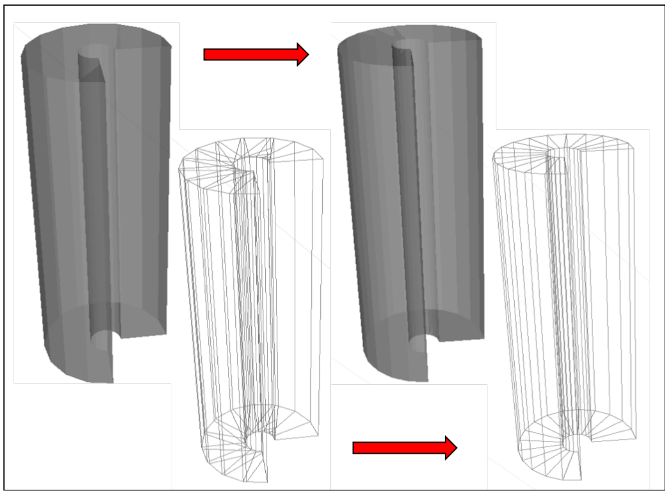

Figure 25: Toroidal Coordinate System

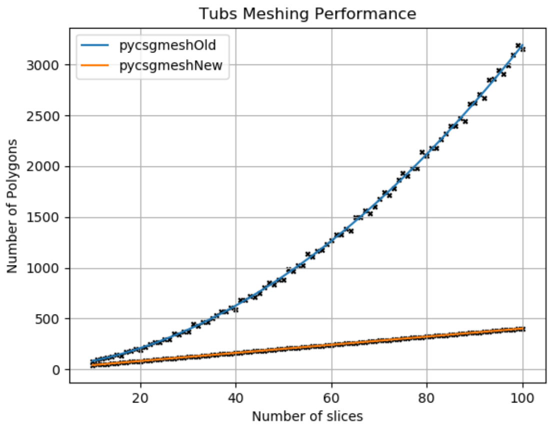

Figure 26: Spherical Coordinate System

2.4 Performance tests

table of perfrmance for each shape
plots

3 BDSIM

3.1 Particle collisions with meshed solids

4 CAD

5 Appendix (Python scripts)

This section lists the Python scripts used to generate some of the figures within this report. Data taken from Online NASA's confirmed exoplanet archive [**?**], downloaded as .cvs file and imported into Python 3.7. "path" is the path to your ".cvs" type file. May be required to delete unnecessary rows explaining column headers.