

# Python lab

by Bruno DENIS, ENS Paris-Saclay

## Sommaire

### General objective

- Put yourself in the context of the PoC project to clarify expectations
  - start from a scientific paper ()
  - Understand the approach proposed in the paper
  - Collect and generate representative input data to benchmark the paper approach
  - Coding to reproduce the paper approach
  - Manage the progress and narration of your work with a Github repository

### Operational objective

- Python
  - object-based data modeling (as in the previous exercise)
  - display 3D graphics (ex: matplotlib module)
  - Using a graph theory library (ex: networkx module)
- Git
  - commit your code in `python_lab` directory
- Jupyter
  - create a narrative of your work (include it on git repository)
  - use basics of markdown format

### Lab statement

- Read the following "tiny" paper : *Triangular representation of a 3-dimensional surface geometry - plan intersection and geodesic*
- Create a `python_lab` directory in your github depository
- Develop a Python code to reproduce paper approach
  - design data structure (UML class diagram could be help)
  - code a Python module able to calculate plan/solid intersection and geodesic
- Evaluate the complexity of your algorithms
- Benchmark paper approach with a set of input data
- Write de narrative of your work (Python module design and usage, paper benchmark, data set description)

*Four last points will be addressed in interactive nanner*

# Tiny paper (not a real one)

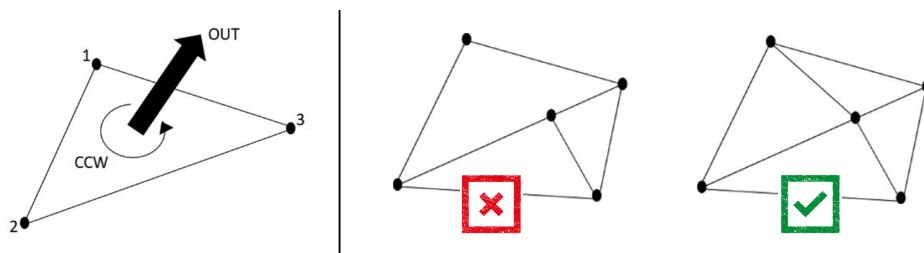
## Triangular representation of a 3-dimensional surface geometry - plan intersection and geodesic

by Someboby ;-)

Abstract: Introduction of STL format to describe 3D solids and how to compute intersection plan/solid dan how to compute geodesic between two points of solid surface.

STL file format is perhaps the main standard for 3D printing. The format is specified as both an ASCII (printable character) format as well as a binary format. This format provides a **triangular representation of a 3-dimensional surface geometry**. The surface is broken down into a series of small triangles call facets. Each facet is identified by a normal unit vector (perpendicular to the facet and with a length of 1.0) and by three vertices (corners of the triangle).

- Orientation of a facet is determined by the direction of the unit normal and the order in which the vertices are listed (see next fig., left side). This two ways to specify facet orientation are redundant and must be consistent.
- Each facet (triangle) must share two vertices (corners) with each of its neighboring facets. Next figure (right side) shows a violation of this rule and a correct configuration.



**ASCII format:** example of file with two facets

```
solid MySolidName

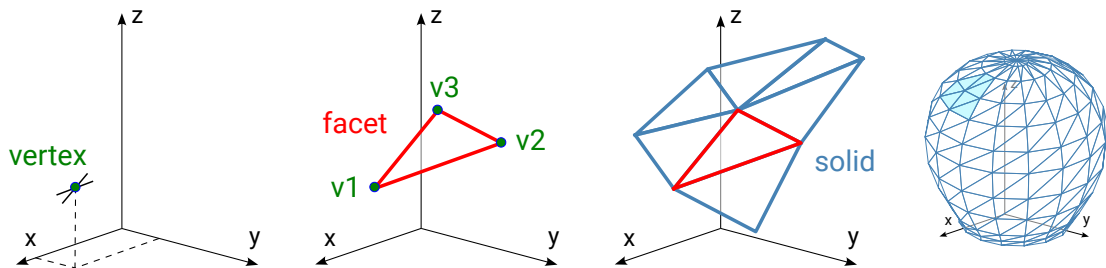
facet normal 0.035587 -0.0022915 0.999364
outer loop
vertex 0.486688 0.030433 8.55517
vertex 0.490792 0.093757 8.55517
vertex 0.031389 0.093757 8.57153
endloop
endfacet

facet normal 0.0349306 -0.0069426 0.999366
outer loop
vertex 0.474708 -0.03 8.55517
vertex 0.486688 0.030433 8.55517
vertex 0.031389 0.093757 8.57153
endloop
endfacet

endsolid MySolidName
```

Keywords `solid` , `endsolid` , `facet normal` , `endfacet` , `outer loop` , `endloop` and `vertex` are in lower case. The numerical data are single precision floats, for example, 1.23456e+789 or 1.23456

Illustration



but let's see how it actually works. Are you planning to use 3D printing for the first time? Or are you looking for more details about file formats suitable for additive manufacturing? Then, this blog post will certainly help to answer some of your questions.

Today, we are going to give you all the information you need about STL files. What is it exactly and what are the advantages for you? How to create it, and how to use it? Let's discover it right now and explain this file format in depth.

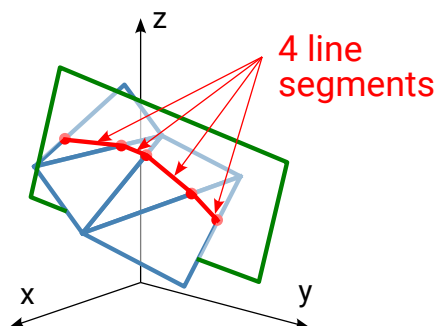
Online conversion from binary STL to ASCII STL: <https://www.meshconvert.com/>,  
<https://fabconvert.com/convert/3d-model>

Online viewer: <https://3dviewer.net/>

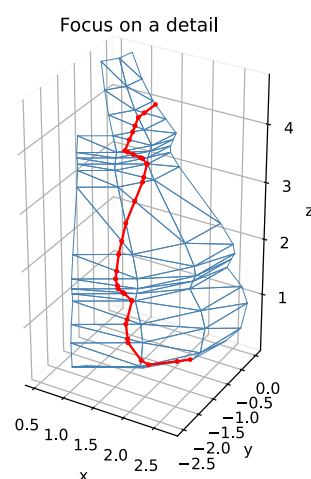
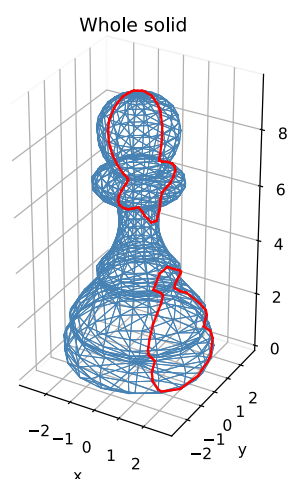
## Intersection with a plan

To get STL model intersection with a plane, the intersection between each facet and the plane is

calculated (nothing, a point or a segment) and the union of all these intersections gives the 3D intersection line



Intersection between a chess pawn and the plan  $7x - y + 2z = 16$



## Intersection plan-line

Notations

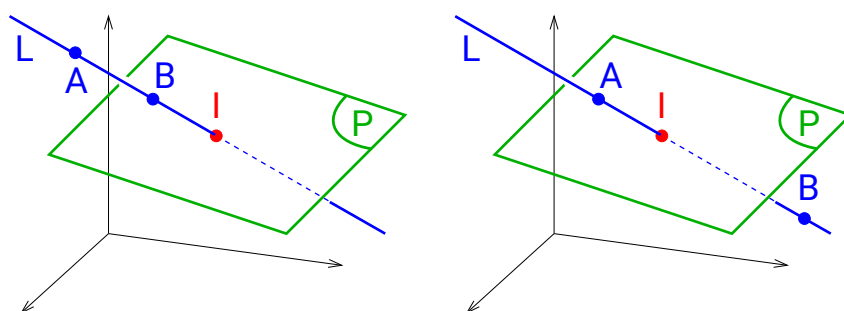
$A(x_A, y_A, z_A)$  and  $B(x_B, y_B, z_B)$  are two points

$L = \overleftrightarrow{AB}$  is a line defined by points A and B

$P$  is a plan defined by the following equation  $M(x, y, z) \in P$  iff  $ax + by + cz = d$

$I(x_I, y_I, z_I)$  is the point intersection common to line  $L$  and plan  $P$

$S = \overline{AB}$  is a line segment bounded by points A and B



$M(x, y, z) \in L$  if it exist  $\exists k \in \mathbb{R}$  such as

$$\overrightarrow{AM} = k. \overrightarrow{AB} \quad (1)$$

$$\begin{pmatrix} x - x_A \\ y - y_A \\ z - z_A \end{pmatrix} = \begin{pmatrix} x_B - x_A \\ y_B - y_A \\ z_B - z_A \end{pmatrix} \quad (2)$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} k(x_B - x_A) + x_A \\ k(y_B - y_A) + y_A \\ k(z_B - z_A) + z_A \end{pmatrix}$$

$$I(x_I, y_I, z_I) \in P \text{ and } I(x_I, y_I, z_I) \in \overleftrightarrow{AB}$$

$$\begin{pmatrix} ax_I + by_I + cz_I \\ x_I \\ y_I \\ z_I \end{pmatrix} = \begin{pmatrix} d \\ k(x_B - x_A) + x_A \\ k(y_B - y_A) + y_A \\ k(z_B - z_A) + z_A \end{pmatrix}$$

$$k = \frac{d - ax_A - by_A - cz_A}{a(x_B - x_A) + b(y_B - y_A) + c(z_B - z_A)}$$

Determine geodesic

Geodesic is a curve representing in some sense the shortest path between two points in a surface.

To get geodesic between two vertices of a solid in STL format, the STL format is viewed as a weighted graph where graph nodes are STL vertices and graph arcs are STL facet segments weighted with facet segments length.

In [ ]: