Chapter 5. Computer aspects of additive processes

November 25, 2024

Outline

5.1.1 Management of information

Common aspect of additive processes

In most of the presented additive processes (SLA, FDM, 3DP, SLS etc...) the part is built layer by layer. There is only one exception (DMD) which will not be adressed in this section.

For layer manufacturing processes it is necessary

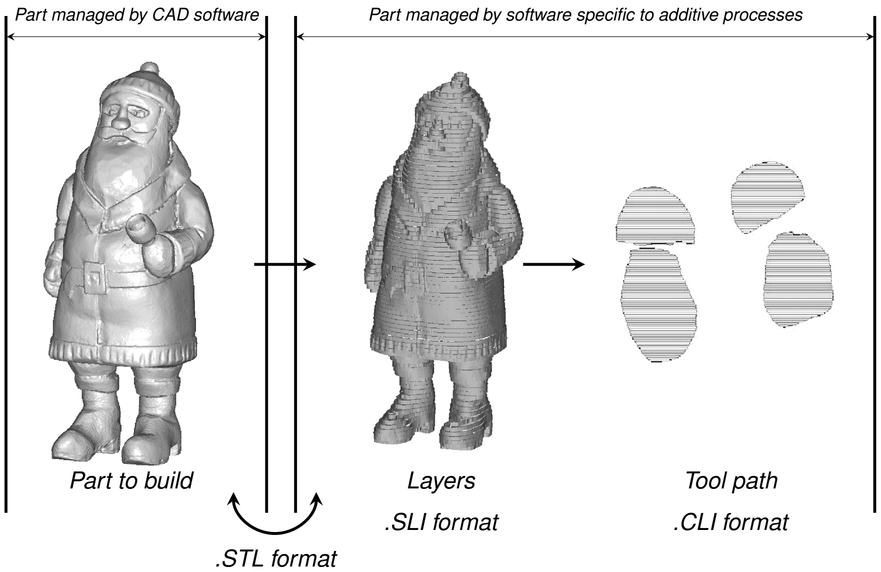
- *to get a computer description of each layer,*
- *to define the tool path (laser, printhead, nozzle) to generate each layer.*

Type of problem

- *The geometric problems to solve are essentially posed in **two-**dimensions only,*
- *They are simpler than the problems encountered in traditional manufacturing processes which are usually posed in a space of dimension ≥ 3:*
  - *In conventional machining, one has to manage the simultaneous movements of n axes ($n = 3, 4, 5 \ldots$).*

5.1.2 Information transfer

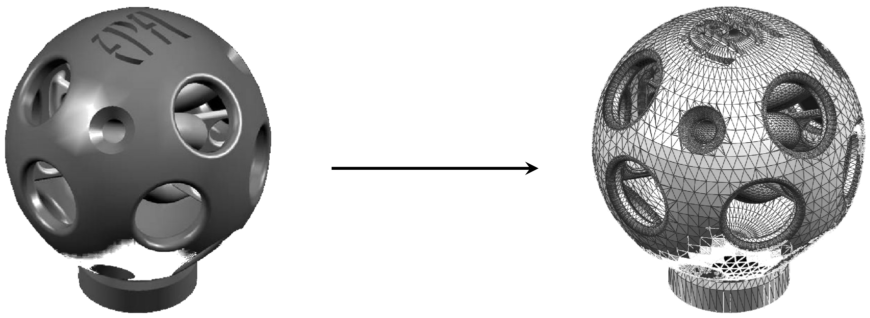
Layers and tool path



5.2.1 .STL format (Standard Tesselation Langu.)

Definition

- *The .STL representation of a part P is a computer description of its surface.*
- *This surface is approximated by a surface made of triangular facets.*



The Part *Its .STL representation*

Conclusion

- *A .STL file is a list of triangles in 3d-space (with their outwards pointing normal).*

5.2.2 Example of .STL file

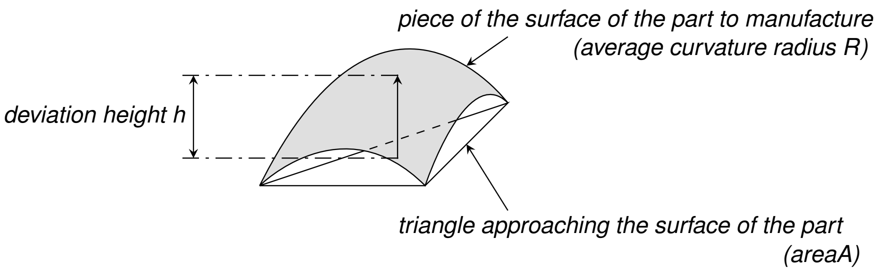
```
                solid sample.stl
                        facet normal   -1.00000   0.00000      0.00000                                  norm ext.
Triangle No 1                          outer loop
                                                  vertex       140.502634   233.993075   -38.310362     1st vertex
                                                  vertex       140.502634   229.424780   -38.310362     2nd vertex
                                                  vertex       140.502634   242.525774   -27.097848     3rd vertex
                                       end loop
                        end facet
                        facet normal   0.903689   0.004563     0.428166
Triangle No 2                          outer loop
                                                  vertex       134.521310   273.427837   30.342009
                                                  vertex       134.521310   308.505852   30.715799
                                                  vertex       140.502634   334.576026   18.369396
                                       end loop
                        end facet
                        facet normal   -0.903689  0.004563     0.428166
Triangle No 3                          outer loop
                                                  vertex       140.502634   334.576026   18.369396
                                                  vertex       140.502634   294.929752   17.946926
                                                  vertex       140.502634   273.427873   30.342009
                                       end loop
                        end facet
                . . .
                end solid sample.stl
```

(see Append. 2)

5.2.3 Control of .STL format

Deviation height and minimal angle

- *The accuracy of the .STL approximation is controlled by the maximal deviation height F_{max}.*
- *The quality of the .STL approximation is controlled by the angle tolerance α_{min}: none of the triangles can have angles smaller than the angle tolerance $\alpha_{\min}$.*



- *If $h \le F_{max} \Longrightarrow$ keep triangle,*
- *If $h > F_{max} \Longrightarrow$ add triangles.*

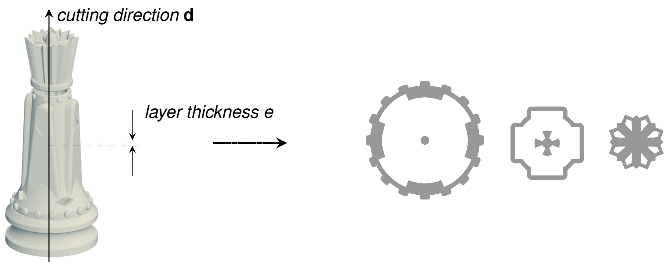
- *$h \simeq C\frac{A}{R}$ where C dep. on $\alpha_{\min}$,*
- *$C = \frac{2}{3\sqrt{3}} = min!$ for $\alpha_{\min} = 60°$.*

(see Append. 1, 3)

5.3.1 .SLI format (Slice Layer Interface)

Definition

- *The .SLI representation of a part P is connected to a .STL approximation, to a direction* **d** *called* ***cutting direction*** *and to a length e called* ***layer thickness***.
- *It contains the polygonal sections obtained by intersecting the STL-polyedral approximation by planes perpendicular to* **d** *and spaced by the distance e.*



The Part *.SLI representation*

- *As a consequence, a .SLI file is a list of segments in two dimensions oriented with the inside at left hand with respect to the cutting direction.*

(see Append. 4)

5.3.2 Example of .SLI files

Header:

```
$$HEADERSTART
$$ASCII
$$UNITS/0.01
$$LAYERS/50
$$HEADEREND
```

File format
0.01 → File resolution in mm/pixel
50 → Number of layers

```
$$GEOMETRYSTART
```

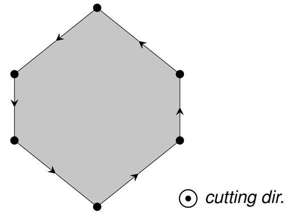
1st layer:

```
$$LAYER/5
$$POLYLINE/6
            VERTEX     653,    1493
            VERTEX    1306,    1158
            VERTEX    1594,     504
            VERTEX    1673,     504
. . . . . . . . .
. . . . . . . . .
```

5 → height of 1st layer($5 \times 0.01 = 0.05$ mm)
6 → nb. of edges
$(653, 1493)$ → start vertex 1st edge:(6.53 mm, 14.93 mm)
$(1306, 1158)$ → end vertex 1st edge: (13.06 mm, 11.58 mm)
$(1594, 504)$ → start vertex 2nd edge: (15.04 mm, 5.04 mm)
$(1673, 504)$ → end vertex 2nd edge: (16.73 mm, 5.04 mm)

2nd layer:

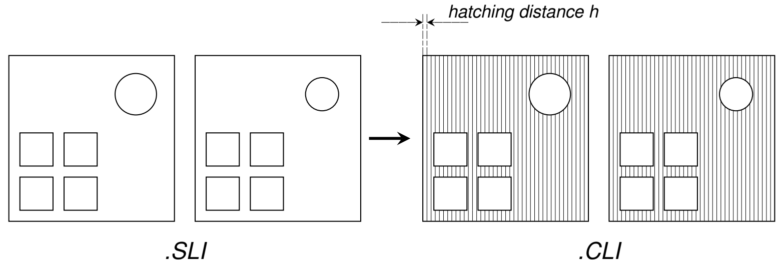
```
$$LAYER/10
$$POLYLINE/8
. . . . . . . . .
. . . . . . . . .
```



5.3.3 .CLI format (Common Layer Interface)

Definition

- *The CLI representation of a part P is connected to a .SLI slicing in sections $S_1, S_2 \ldots S_N$ and to a length h called* **hatching distance**.
  It contains a list of broken lines (possibly discontinuous) that fills the sections $S_1, S_2 \ldots S_N$ for the distance h.



- *As a consequence, a .CLI is a list of ordered points in 2d (the vertices of the broken lines)*

(see Append. 5)

5.3.4 Example of .CLI files

Header:

```
$$HEADERSTART
$$ASCII
$$UNITS/0.01
$$LAYERS/50
$$HEADEREND
```

File format
0.01 → File resolution in mm/pixel
50 → Number of layers

```
$$GEOMETRYSTART
```

1st layer, 1st br.l.cont.:

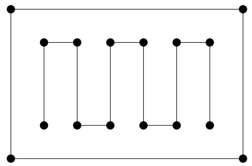
```
$$LAYER/5
      $$HATCHES/0,12
            VERTEX      5,      5
            VERTEX      5,    995
            VERTEX     10,    995
            VERTEX     10,      5
            . . . . . . . . .
```

0 → open curve, 12 → nb. of points
$(5, 5)$ → 1st point: (0.05 mm, 0.05 mm)
$(5, 995)$ → 2nd point:(0.05 mm, 9.95 mm)
$(10, 995)$ → 3rd point: (0.1 mm, 9.95 mm)
$(10, 5)$ → 4th point: (0.1 mm, 0.05 mm)

2nd br.l.cont.:

```
      $$HATCHES/1,4
            VERTEX      0,      0
            VERTEX     35,      0
            VERTEX     35,   1000
            VERTEX      0,   1000
            . . . . . . . . .
      . . . . . . . . .
```

1 → closed curve, 4 → nb. of points
$(0, 0)$ → 1st point: (0.00 mm, 0.00 mm)



2nd layer:

```
$$LAYER/10
      $$HATCHES/0,8
      . . . . . . . . .
      . . . . . . . . .
```


5.3.5 Transformation of .STL format to .SLI format

Type of problem

To transform a .STL file into a .SLI file, we have to intersect a polyhedron by a set of planes:

- *spaced by a layer thickness e,*
- *parallel (by a rotation, one reduces the situation to the case where planes are horizontal)*

Algorithm

- *A loop on the cutting planes Π.*
  - *A loop on the triangles T (oriented) contained in the .STL file*

  ⟹ *Computation of $\gamma = \Pi \cap T$, If γ is*

a segment (oriented),	*install it in the .SLI file,*	*go back to the loop,*
a point,	______	*go back to the loop,*
the empty set,	______	*go back to the loop,*
the entire triangle T,	______ *exceptional sit. to handle.*	______

(see Append. 6, 7)

5.3.6 Transformation of .SLI format to .CLI format

Type of problem (coloring problem)

To transform a .SLI file into a .CLI file, we have to fill polygons by broken lines for a hatching distance h .

Standard algorithm

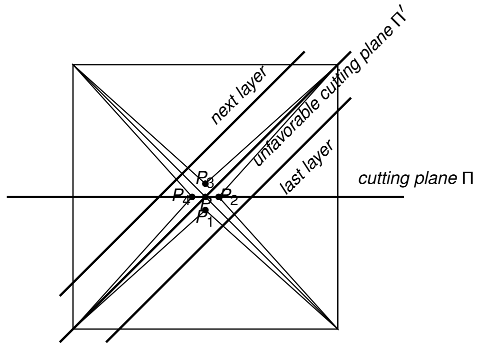
- *We set the polygon Σ on a grid of step h, then*
  - *One performs a loop over the grid points P.*
    - *Loop over the segments γ (oriented) contained in the .SLI file $\Longrightarrow$ computation of the angle ϑ under which P sees γ*
    - *If $\sum \vartheta = 2\pi$, install P in the .CLI file* — *back to the loop on P*
    - *If $\sum \vartheta = 0$, ____________* — *back to the loop on P*
- *When the process uses a pointlike tool (e.g the laser SLA, SLS, or the nozzle in FDM), one applies a suitable algorithm to create a trajectory (broken line) by ordering the points in the .CLI file.*
- *For processes applying global consolidation (e.g. DLP, SGC, JetFusion), the step of connecting points together is not needed. The position of the points inside the polygonal layer is the only relevant information in that case.*

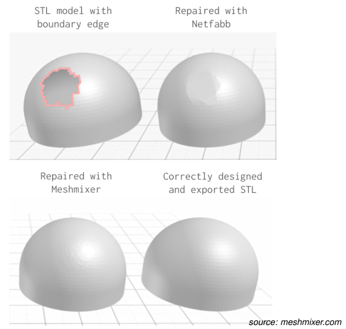
(see Append. 8, 9, 10, 11)

5.4.1 Known problems related to .STL format

The **redundancy** issue $\Longrightarrow$ non closed surface

- *If a point P is a vertex of N triangles it appears N times in the .STL file.*
- *Because of rounding errors, the computer considers that it has to do with N points $P_1 \dots P_n$ close but* ***separate****,*
- *This can result in catastrophic errors when layering the part (non closed polygons!).*
- *Such errors are identified by an examination of the shape changes in the successive layers.*





5.4.2 Alternative solution to store .STL-like infos

The surface of the polyhedral part is represented by:

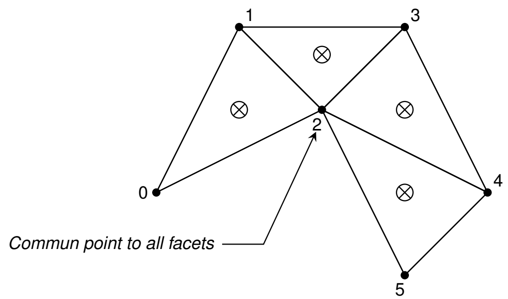
The list of vertices

Node 0:	*140.502*	*233.993*	*-38.310*
Node 1:	*140.502*	*229.424*	*-38.359*
Node 2:	*140.502*	*242.525*	*-27.097*
Node 3:	*134.521*	*273.427*	*30.342*
Node 4:	*134.521*	*308.505*	*30.715*
Node 5:	*140.502*	*334.576*	*18.369*

The list of facets

Facet 0:	*0*	*1*	***2***
Facet 1:	*1*	*3*	***2***
Facet 2:	***2***	*3*	*4*
Facet 3:	***2***	*4*	*5*

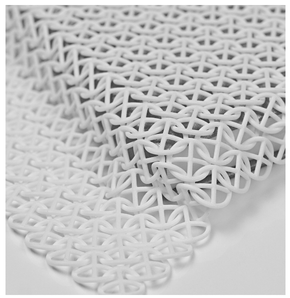
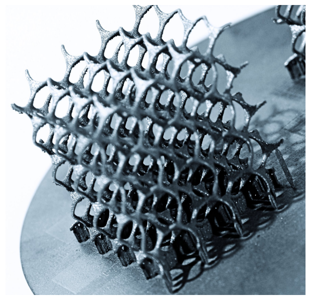
N.B. *The outwards pointing normals are recovered by applying the corkscrew rule:*



5.4.3 Known problems related to .STL format

Parts with high aspect ratio, multimaterial parts

- *The STL format is not adapated to parts with large aspect ratio i.e large surface for small volume. They have very low local radius of curvature and their polyhedral approximation requests a lot of triangles. Popular examples of such situations are knitted fabrics, lattice structures, etc....*

knitted fabrics	***lattice structures***
 source: 3dSystems™	 source: SLMSolution™

- *The STL format is not adapated to multimaterial parts either.*

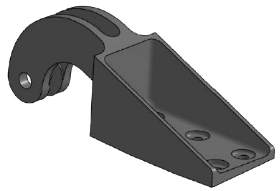
(see Append. 14)

5.5.1 Design for additive manufacturing (DFAM)

Typical mistake to be avoided

(1) *The part is designed traditionally*

(2) *The part is eventually AM'd for some reasons*



source: EADS innovation team

→ *The reasons could be: small serie, high complexity, etc..*

A more efficient solution exists:

(1) *AM is choosen for some reasons (same as above)*

(2) *The part is designed for AM.*

source: EADS innovation team

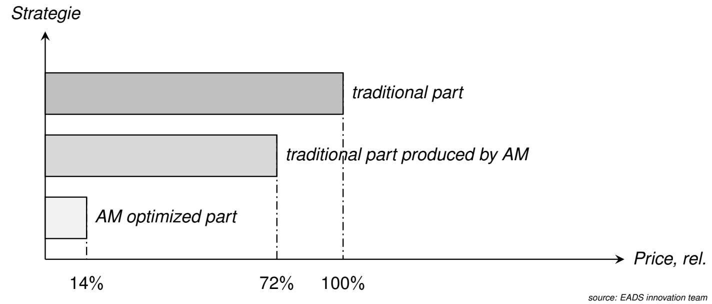
(see Append. 15, 16, 17)

5.5.2 Design for additive manufacturing (DFAM)

Expected gains

- *For the same functions and mechanical properties, an AM optimized part*
  (1) *is lighter and uses less material than the traditional part,*
  (2) *is much cheaper than the traditional part when it is produced by AM:*
  (3) *makes additive manufacturing even more profitable (see Fig. below).*

Price analysis over an average part - aerospace application



5.5.3 Design for additive manufacturing (DFAM)

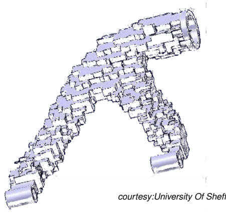
Paradigm modification

***Traditional manufacturing**:*	*Remove material only where it is **really** superfluous*
***Additive manufacturing**:*	*Put material only where it is **really** necessary*

Consequences of the paradigm modification

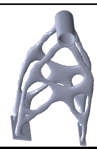
***Change in the designer's** job:*

- *More time for topological optimization of parts*
- *Less time to fulfill the construction constraints*

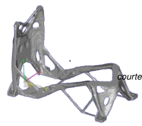


courtesy:University Of Sheffield

***Change in the user's** mentality*



courtesy:IRCCYN

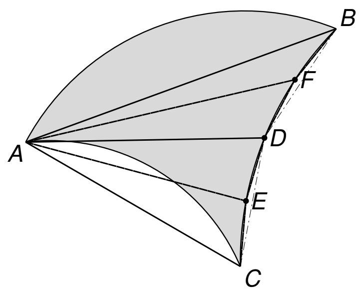


courtesy:CNES

APPENDICES

A 1: Dividing triangles

The natural tendency to divide triangles is to cut angles

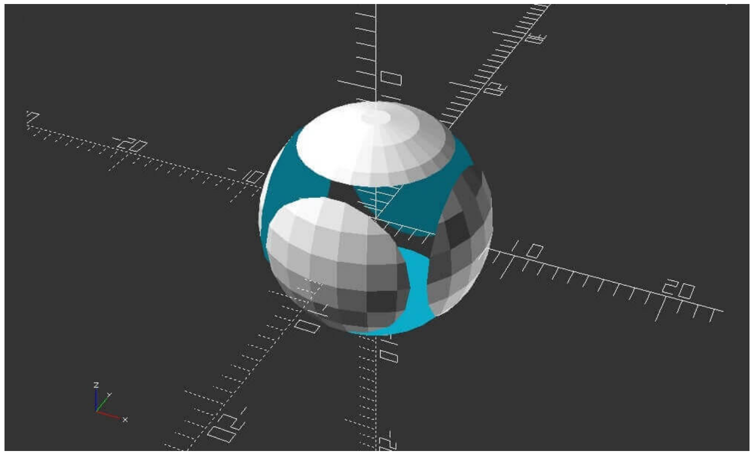


- *It results in the creation of almost degenerated triangles (looking like segments).*
- *It is the source for numerical difficulties, like instabilities in **unit normal** computation:*

$$\mathbf{n} = \frac{\mathbf{AB} \wedge \mathbf{AF}}{||\mathbf{AB} \wedge \mathbf{AF}||} \quad \textit{\textbf{but}} \quad ||\mathbf{AB} \wedge \mathbf{AF}|| \simeq 0$$

A 2: Origin of the .STL format

The .STL format has been developpped for rendering purposes

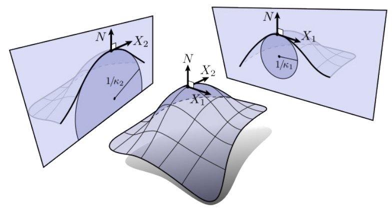


- *The outwards pointing normal was necessary to determine the brightness level of each triangles : the amount of light diffracted to you depends on the angle between this normal and the incoming light .*

A 3: Radius of curvature and deviation height

Radius of curvature of a surface: definition

- *The local radius of curvature of a surface is R if, locally, the surface is well approached by a sphere with radius R.*

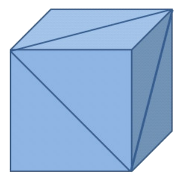


Radius of curvature and deviation height

- *If the radius of curvature of Σ is infinite $R = \infty$, then the deviation height is*

$$h \simeq C\frac{A}{R} = 0$$

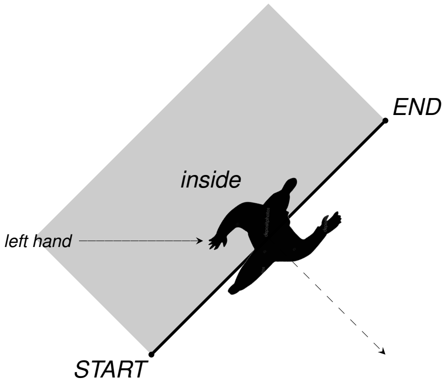
- *If $R = \infty$, Σ is locally a part of sphere of radius ∞. But a sphere of radius ∞ is a plane. A surface which is locally a plane is a polyhedron and its polyhedral approximation is the surface itself and there is no approximation error.*



A 4: Oriented edges

Rule to be applied

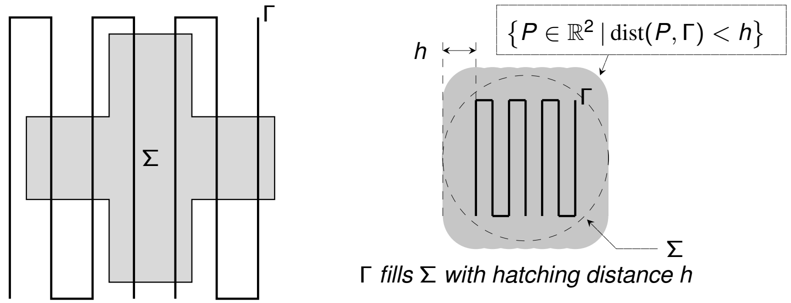
- *A man with the head in the $z-$ direction (build direction) has the inside at **left hand** when he walks from START to END!*



A 5: A curve fills a surface

Definition

- *We say that a curve Γ (possibly discontinuous) fills a surface Σ with a hatch distance h if and only if*
  (i) $\Gamma \subset \Sigma$
  (ii) $\Sigma \subset \left\{P \in \mathbb{R}^2 \mid \mathrm{dist}(P,\Gamma) < h\right\}$



Γ does not fill Σ because $\Gamma \not\subset \Sigma$

Γ fills Σ with hatching distance h

A 6: Choosing the manufacturing direction

General remarks

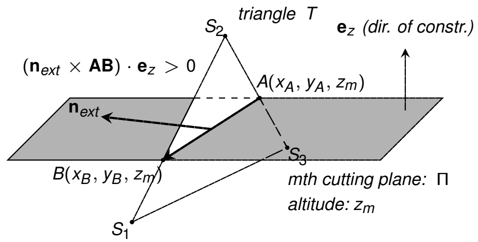
- *By default the layering of a .STL representation is made according to horizontal plans.*
- *Ideally, the manufacturing direction must be determined at the CAD level (z axis).*
- *The choice of the manufacturing direction is not a trivial decision. This choice influences:*
  - *the need to build a support structure,*
  - *the location of the supports,*
  - *the manufacturing time, which increases with the building height.*

A 7: Conversion from .STL to .SLI format

Installing a segment into a .SLI file

.STL file sequence corresponding to the current triangle T

facet normal	*-1.0*	*0.0*	*0.0*	
outer loop				
	vertex	*140*	*233*	*-38*
	vertex	*140*	*229*	*-38*
	vertex	*140*	*242*	*-27*
end loop				

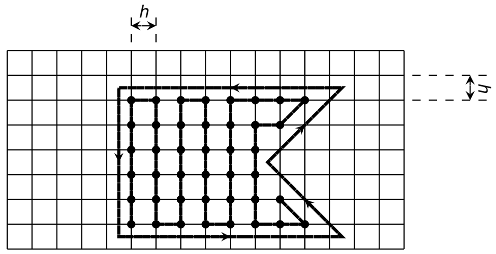


Current sequence of the .SLI file corresponding to layer no m***: list of oriented segments***

$$POLYLINE/n+1			
VERTEX	...		*start segment no 1*
VERTEX	...		*end segment no 1*
⋮			
VERTEX	...		*end segment no n*
VERTEX	x_A	y_A	*start segment no n+1*
VERTEX	x_B	y_B	*end segment no n+1*

A 8: Transformation of .SLI format to .CLI format

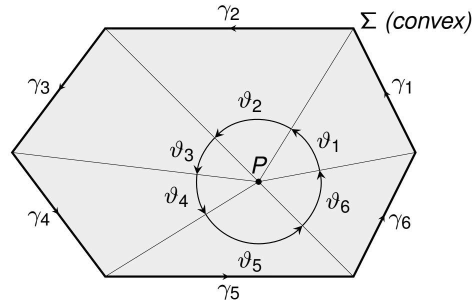
Points in a polygon, travelling algorithms



- *There are several types of algorithms:*
  - i) *snake filling,*
  - ii) *filling parallel to a direction (uni-directional filling),*
  - iii) *filling with offset from the boundary,*
  - iv) *random filling under constraints (no crossings, minimal length, . . .).*
- *In the particular case of selective laser sintering processes, the type of filling may affect the mechanical properties of the parts. In particular, they condition:*
  - a) *the powder consolidation,*
  - b) *the residual stresses in the finished part.*

A 9: Localisation of a point P in a polygone Σ

A criterion based on the view angles



Observation in the case where the polygon Σ is convex

- *If* $P \in \Sigma$ *then* $\sum_{j=1}^{j=N} \vartheta_j = 2\pi$
- *If* $P \notin \overline{\Sigma}$ *then* $\sum_{j=1}^{j=N} \vartheta_j = 0$
- *If* $P \in \partial\Sigma$ *then* $\sum_{j=1}^{j=N} \vartheta_j = \begin{cases} \pi & \text{if } P \text{ is not a vertex} \\ \alpha & \text{if } P \text{ is a vertex of angle } \alpha \end{cases}$

A 10: Localisation of a point P in a polygone Σ

The rule generalizes to any polygon

- *If* $\sum_{j=1}^{j=N} \vartheta_j = 2\pi$ *then* $P \in \Sigma$,
- *If* $\sum_{j=1}^{j=N} \vartheta_j = 0$ *then* $P \notin \overline{\Sigma}$,
- *If* $\sum_{j=1}^{j=N} \vartheta_j = \alpha$ *with* $0 < \alpha < 2\pi$ *then* $P \in \partial\Sigma$ *is a vertex of angle* α.
  - *This rule is numerically more stable than others (eg. intersection criteria).*
  - *It is an expression of Gauss law in 2d-electromagnetics. If there is a unit charge located at point P, we know that the produced 2d-electrical field* **E** *satisfies:*

$$\int_{\partial\Sigma} \mathbf{E}\cdot\mathbf{n}dl = 2\pi, \text{ if } P \in \Sigma \quad \text{and} \quad \int_{\partial\Sigma} \mathbf{E}\cdot\mathbf{n}dl = 0, \text{ if } P \notin \overline{\Sigma}.$$

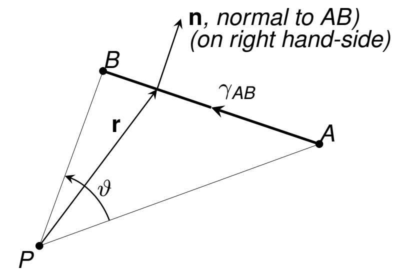
If Σ *is a polygon, the* ***flux*** *of* **E** *is the sum of the view angles* $\sum_{j=1}^{j=N} \vartheta_j$, *therefore:*

$$\sum_{j=1}^{j=N} \vartheta_j = 2\pi, \text{ if } P \in \Sigma \quad \text{and} \quad \sum_{j=1}^{j=N} \vartheta_j = 0, \text{ if } P \notin \overline{\Sigma}, \quad \Sigma : \text{a polyon.}$$

(see Append. 12, 13)

A 11: View angle of an oriented segment

Connection to electric flux and computational rule



- ϑ *is the view angle of segment* γ_{AB} *when it is observed from P. One has:*

$$\vartheta = \int_{\gamma_{AB}} \frac{\mathbf{r}\cdot\mathbf{n}}{r^2}\,\mathrm{d}l$$

where $\frac{\mathbf{r}}{r^2}$ *is the 2d-electric field produced by a unit charge in P.*

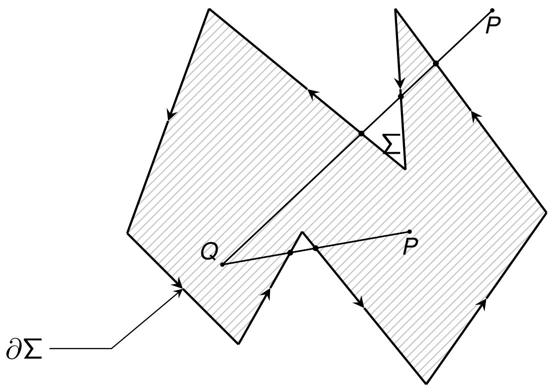
Computational rule: ϑ is computed in four steps

1) *Compute* $l = \mathbf{AB}\cdot\mathbf{AB}$
2) *Compute* $y = \mathbf{PA}\cdot\mathbf{AB}$
3) *Compute* $x = \mathbf{PA}\times\mathbf{AB}$
4) *Compute* $\vartheta = \arctan\frac{y+l}{x} - \arctan\frac{y}{x}$

NB: $\begin{pmatrix} \xi_1 \\ \eta_1 \end{pmatrix} \times \begin{pmatrix} \xi_2 \\ \eta_2 \end{pmatrix} = \xi_1\eta_2 - \eta_1\xi_2,$

A 12: Point localisation: intersection criteria

Point in Σ, brute force algortihm



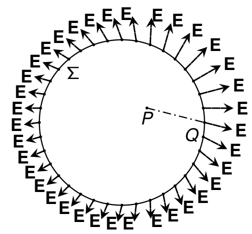
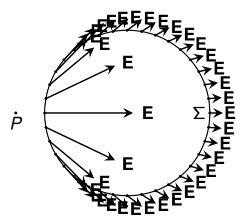
- $P \in \Sigma \iff PQ \cap \Sigma$ ***contains an even number of elements***
- $P \notin \Sigma \iff PQ \cap \Sigma$ ***contains an odd number of elements***

A 13: Gauss Law

Σ is a curve and a unit charge is located at a point P

- *Electric field at Q in 2d:*

$$\mathbf{E}(Q) = \frac{\mathbf{PQ}}{||\mathbf{PQ}||^2}$$

	$P \in \Sigma$	$P \notin \Sigma$
		
Gauss Law:	$\int_{\partial\Sigma} \mathbf{E} \cdot \mathbf{n}\, \mathrm{d}l = 2\pi$	$\int_{\partial\Sigma} \mathbf{E} \cdot \mathbf{n}\, \mathrm{d}l = 0$

A 14: Multimaterial applications

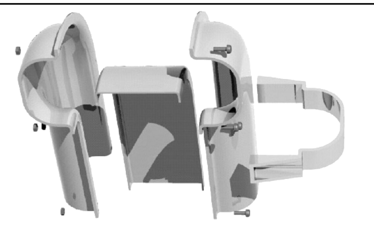
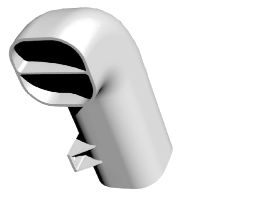
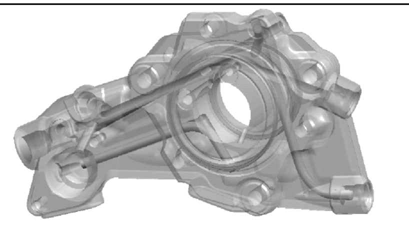
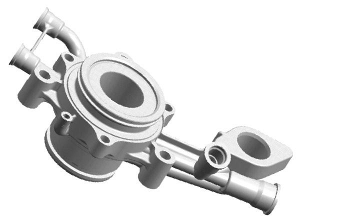
Possible solution: consider different parts

- *Produce **two** .STL files:*
  - (1) *one for the bones,*
  - (2) *one for the soft tissue.*
- *Slice the **two** .STL independently to get **two** .SLI files.*
- *Hatch the **two** .SLI files independently to get **two** .CLI files.*

- *Construct the part layer layer by moving the print-head according:*
  - (1) *to the info in the first .CLI file with the bone material,*
  - (2) *to the info in the second .CLI file with the soft tissue material.*

A 15: Other examples of part designed for AM

New design, new fonctions

	Traditional design	***AM design***
Airducts (aircraft).		
Pump system		

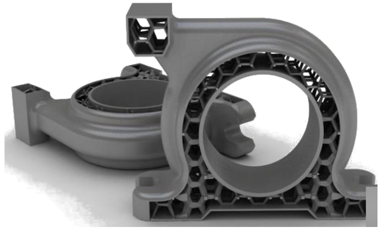
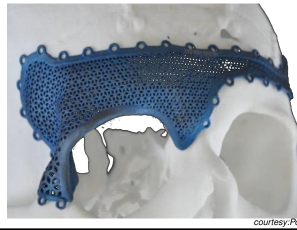
A 16: Other examples of part designed for AM

New design, new fonctions

	Traditional design	***AM design***
Gear box	 *courtesy:3T RPD*	 *courtesy:3T RPD*
Hinge	 *courtesy:University Of Sheffield*	 *courtesy:University Of Sheffield*

A 17: Other examples of part designed for AM

New design, new fonctions

Design for AM only	
 *courtesy:Polyshape™*	 *courtesy:Polyshape™*
 *courtesy:ConceptLaser™*	 *courtesy:Polyshape™*