STL : qu’est-ce que ce format de fichier 3D ?

https://www.iteration3d.fr/fr/blog/what-is-the-stl-3d-file-format.php

Vous utilisez souvent des fichiers STL sans vraiment savoir ce qu’ils contiennent ? Ce format emblématique de l’impression 3D existe depuis 1987… mais reste encore aujourd’hui le standard incontournable.

Découvrez l’origine du format, son fonctionnement basé sur un maillage de triangles, et pourquoi il est si universel.

Savez-vous qu’un simple arrondi peut multiplier la taille d’un fichier par 400 ? Ou que même un fichier de quelques lignes texte peut produire un cube parfait ?

Enfin, comprenez ses limites… et pourquoi malgré tout, il continue de dominer face aux formats plus récents. Un article complet, illustré et accessible.

L'origine du format STL

Le format de fichier 3D STL a été créé en 1987 par la société 3D Systems, aujourd'hui encore acteur majeur de l'impression 3D. A l'époque la société avait besoin de définir un format d'échange pour sa technologie d'impression 3D dite “stéréolithographique”, aujourd'hui plus connue sous le nom d'impression 3D à résine.

Et c'est bien ici que l'acronyme “STL” prend son origine, cette succession de trois lettres étant en fait l'abbréviation pour “**STereoLithography**”. On trouve aussi comme décryptage de l'acronyme “STL” “**Standard Triangle Language**”. Cette version de l'acronyme STL a également beaucoup de sens comme nous le verrons plus tard…

A l'époque donc, la firme américaine 3D Systems avait besoin d'un nouveau format de fichiers répondant aux contraintes techniques de la toute première imprimante 3D commerciale à résine. Les ordinateurs étaient peu puissants et les ressources mémoire très limitées. Il fallait donc un fichier aussi léger et simple que possible, capable de décrire uniquement la forme extérieure d’un objet sans surcharger la machine. C'est ainsi qu'est né le format STL, un format minimaliste, définissant la surface du modèle 3D, sans texture ni couleur. Une forme 3D définit de manière aussi simple que possible, à l'époque.

Bien que de nouveaux formats plus riches pour l'impression 3D ont été récemment développés (comme le format 3MF), le format STL est encore très utilisé aujourd'hui, aussi bien pour l'impression 3D résine que l'impression 3D à dépôt de filament fondu (dite impression 3D “FFF”).

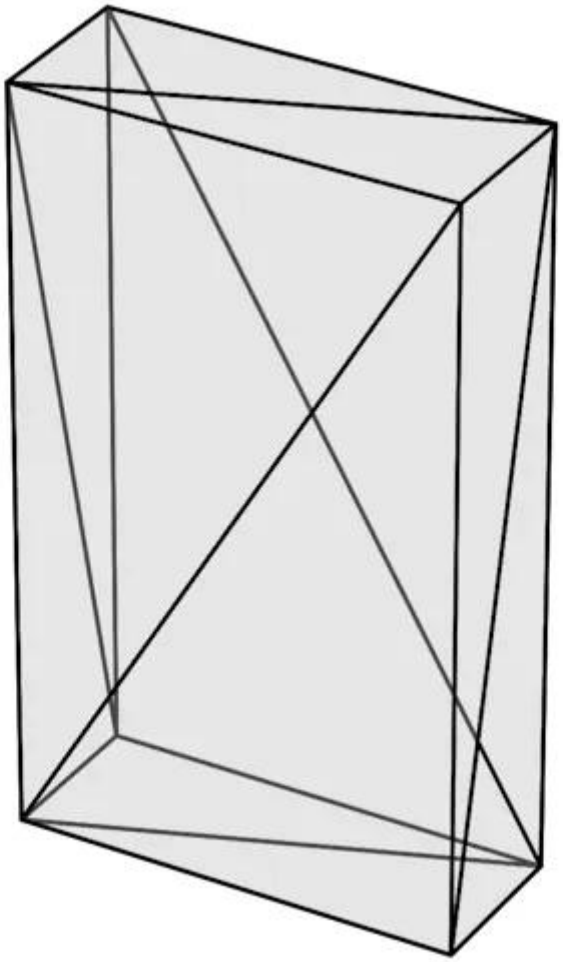
Qu'est-ce qui caractérise le format 3D STL ?

Un fichier STL décrit une forme 3D. Cette forme est en fait une maillage de triangles. Lorsque vous visionnez un fichier STL dans un trancheur (ou “slicer”) vous voyez en fait une forme globale composée d'une juxtaposition de triangles. Etonnant au premier abord !

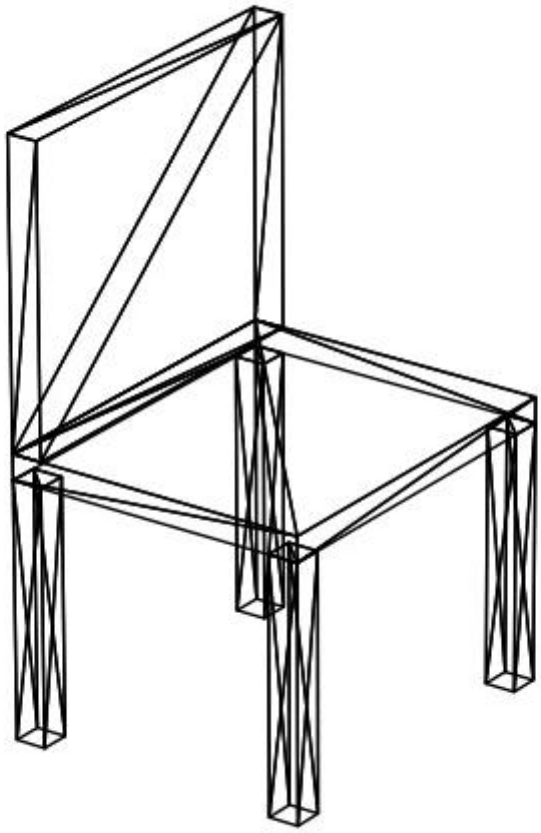
Dans le cas d'un pavé (ou parallélépipède rectangle) la forme3D est composée de 12 triangles, soit 2

triangles par face. Quel est alors le volume 3D le plus simple défini en STL ? Il s'agit d'un tétraèdre composé de 4 triangles.

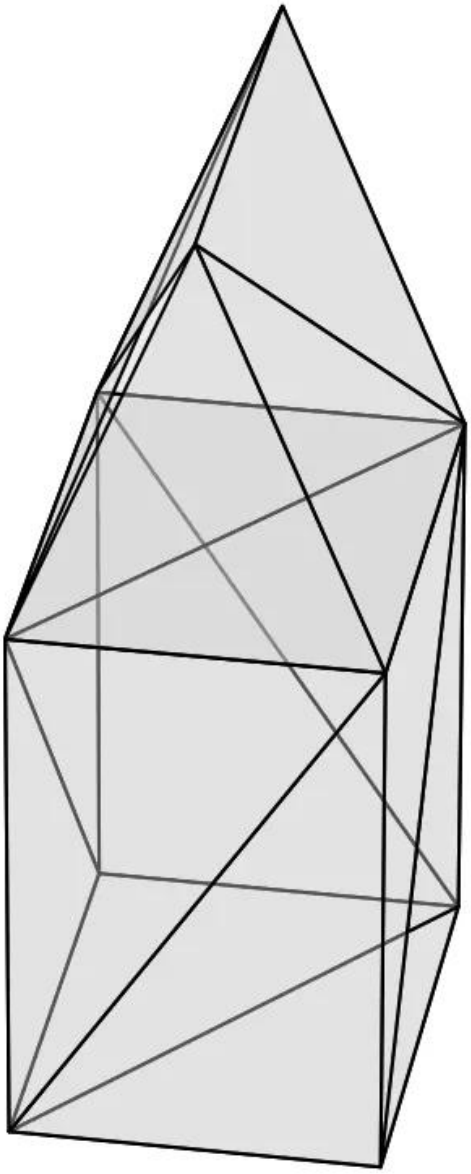
Voici quelques images illustratives représentées avec les triangles visibles :



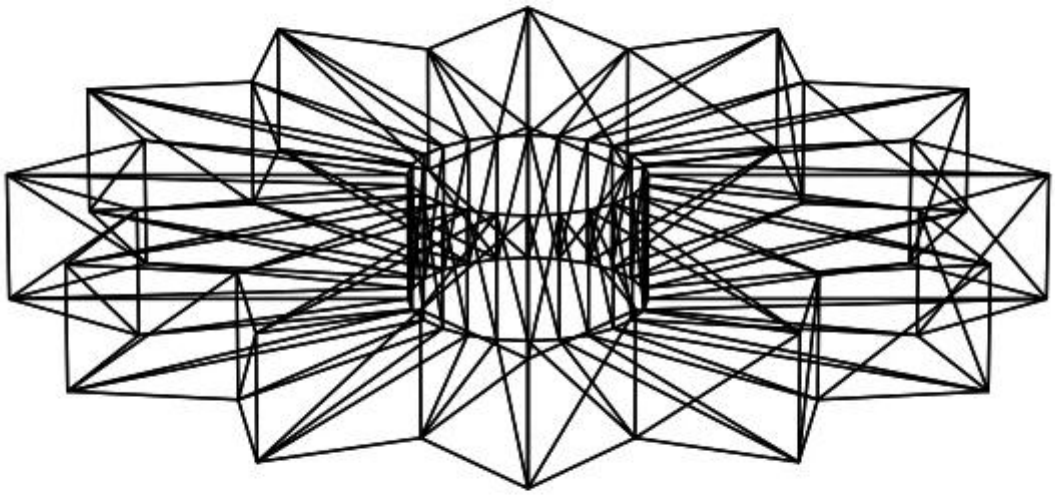
```
                                                <note>*pavé composé de 12
triangles</note>
```



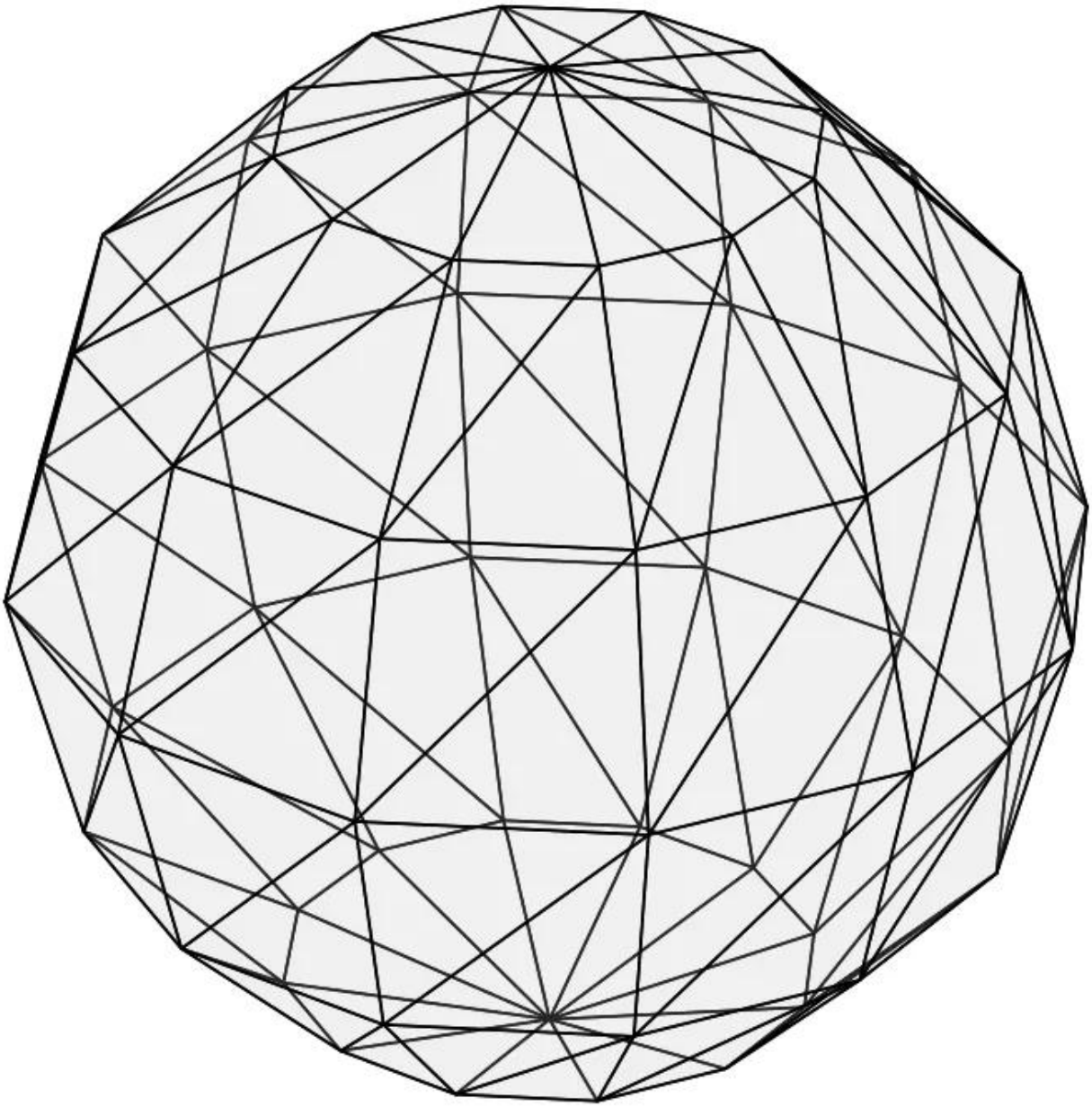
```
                                                    <note> *principe d'une
chaise</note>
```



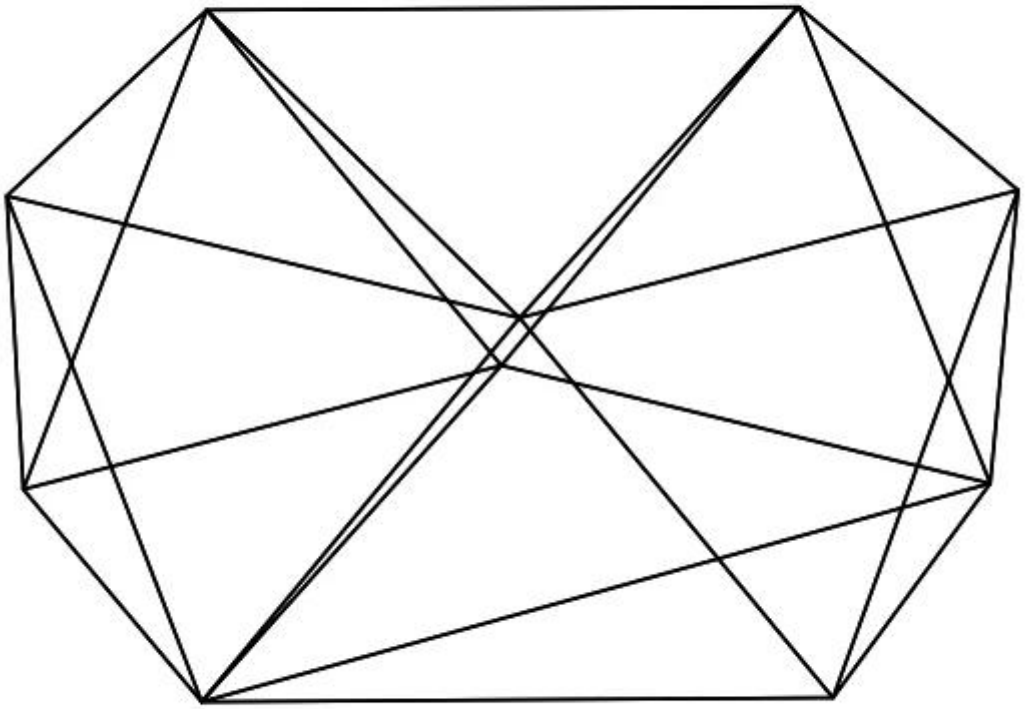
- principe d'une maison



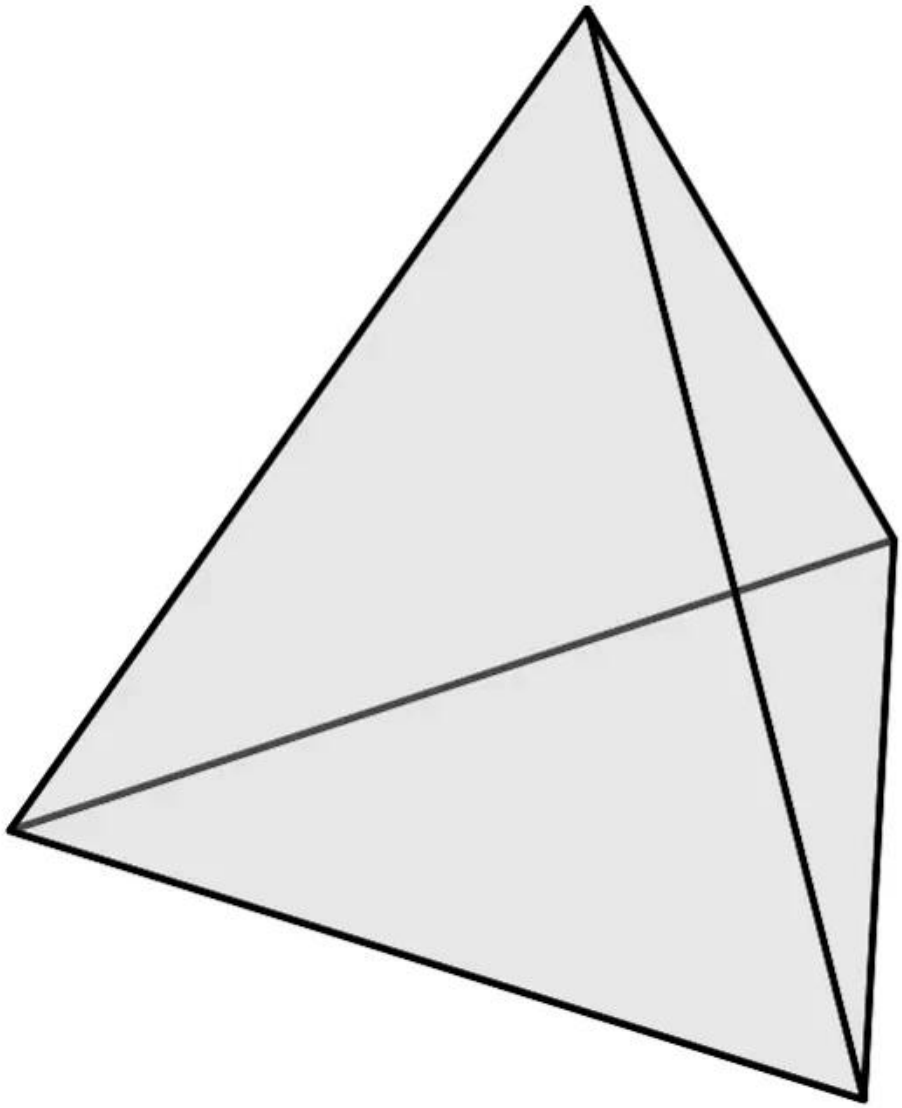
principe d'un engrenage



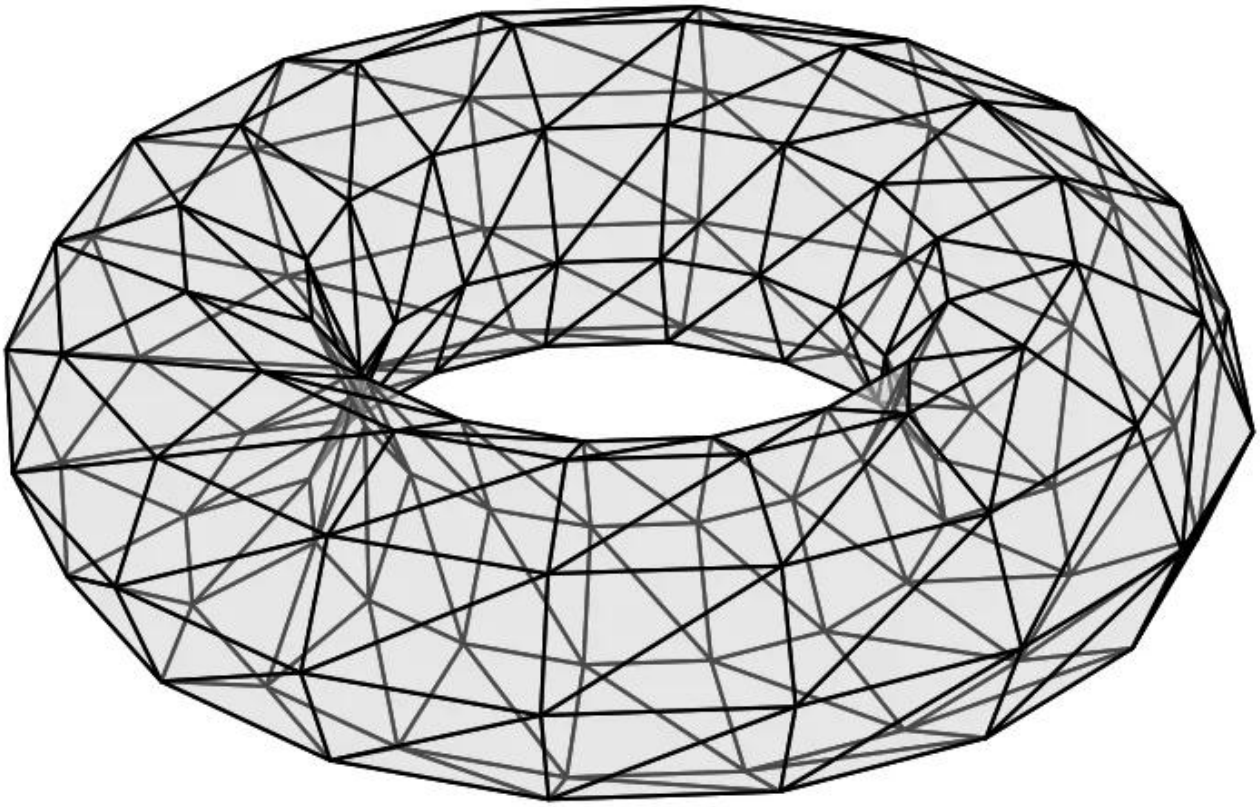
sphère en mode lowpoly



antiprisme pentagonal



tétraèdre composé de 4 triangles, le volume 3D STL le plus simple



tore

Si vous avez déjà travaillé sous SkecthUp vous avez sans doute remarqué ce type de structure en triangles. Maintenant vous savez pourquoi ! Pourquoi donc un triangle comme forme de base de tout volume 3D ? Les raisons sont multiples :

- Trois points forment toujours un plan, contrairement à quatre points par exemple. (aparte de la vie réelle : un tabouret à 3 pieds aura toujours la totalité de ses pieds au sol. Il sera toujours stable contrairement à un tabouret à 4 pieds qui peut être bancal car tous ces pieds ne sont pas forcément sur le même plan !).
- un triangle est “simple” à traiter par les algorithmes.
- Si une surface complexe comporte des erreurs dues à une mauvaise triangulation, on obtiendra quand même un rendu simplifié, mais encore exploitable.

Un fichier 3D est donc un maillage de triangles, qu'il s'agisse d'un cube ou d'une sphère. Dans le cas d'un cube pur, le nombre de triangles est fixe : 2 triangles par face x 6 faces = 12 triangles. Dans le

cas d'une sphère, cela dépendra de la résolution souhaitée. Plus la sphère est lisse, plus le nombre de triangles nécessaires est importante et donc plus le poids du fichier sera importante.

Vous l'aurez compris, les formes “angulaires” produiront un fichier moins lourd que les formes sphériques. Prenons un exemple avec les modèles 3D disponibles sur iteration3d et plus précisément un cube de 100x100x100 mm. Le fichier STL du modèle 745, cube pur de 100x100x100 mm pur pèse 1ko. Ajoutons un filet (bord arrondi) de 2 mm sur toutes les arêtes : le modèle 744 pèse 462 ko ! En effet, les bords arrondis nécessitent une très grande quantité de triangles qui se traduit par un fichier plus lourd.

Que se passe t'il si l'on compare un cube de 100x100x100 mm et un autre de 10xx10x10 mm. Plus lourd ? Non, quel que soit les dimensions de ce cube, le nombre de triangles est identique (12) et donc la taille de fichier identique !

C'est bien le nombre de triangles qui détermine le poids du fichier. Prenons un minuscule joint torique de diamètre extérieur de 10,1 mm et d'épaisseur 2,4 mm, celui-ci pèse 1550ko ! Bien entendu, toutes ces comparaisons sont basées sur une définition (résolution) identique. Concrètement, que comporte un fichier STL ?

Très bien, un fichier 3D STL est un maillage de triangles. Mais concrètement, comment cela prend forme ? Et bien, logiquement, un fichier STL définit… des triangles. Le code ci-dessous définit un cube :

exemplestl.stl

```
solid cube
facet normal 0 0 -1
    outer loop
    vertex 0 0 0
    vertex 100 0 0
    vertex 100 100 0
    endloop
endfacet
facet normal 0 0 -1
    outer loop
    vertex 0 0 0
    vertex 100 100 0
    vertex 0 100 0
    endloop
endfacet

facet normal 0 0 1
    outer loop
    vertex 0 0 100
    vertex 100 100 100
    vertex 100 0 100
    endloop
endfacet
facet normal 0 0 1
    outer loop
    vertex 0 0 100
```

```
        vertex 0 100 100
        vertex 100 100 100
        endloop
    endfacet

    facet normal 0 -1 0
        outer loop
        vertex 0 0 0
        vertex 100 0 100
        vertex 100 0 0
        endloop
    endfacet
    facet normal 0 -1 0
        outer loop
        vertex 0 0 0
        vertex 0 0 100
        vertex 100 0 100
        endloop
    endfacet

    facet normal 0 1 0
        outer loop
        vertex 0 100 0
        vertex 100 100 0
        vertex 100 100 100
        endloop
    endfacet
    facet normal 0 1 0
        outer loop
        vertex 0 100 0
        vertex 100 100 100
        vertex 0 100 100
        endloop
    endfacet

    facet normal -1 0 0
        outer loop
        vertex 0 0 0
        vertex 0 100 100
        vertex 0 0 100
        endloop
    endfacet
    facet normal -1 0 0
        outer loop
        vertex 0 0 0
        vertex 0 100 0
        vertex 0 100 100
        endloop
    endfacet

    facet normal 1 0 0
```

```
                    outer loop
                    vertex 100 0 0
                    vertex 100 0 100
                    vertex 100 100 100
                    endloop
                endfacet
                facet normal 1 0 0
                    outer loop
                    vertex 100 0 0
                    vertex 100 100 100
                    vertex 100 100 0
                    endloop
                endfacet
                endsolid cube
```

Copiez ce code et collez-le dans Notepad par exemple. Enregistrez-le avec le nom “test.stl”. Ouvrez ce STL dans un trancheur comme PrusaSlicer par exemple. Qu'obtenez-vous ? Un cube de 100x100x100 mm !

Décortiquons ces quelques lignes de code :

solid et ensolid définissent le modèle. Ici le nom donné est “cube” mais cela n'a aucun impact sur la génération du modèle. On aurait pu mettre “model1” à la place de “cube.”

Chaque facet définit un triangle. Il y en a donc 12 dans le cas d'un cube. Ces facet sont ici regroupées par 2, représentants ainsi les 6 faces du cube.

Notez que vous avez créé là un fichier STL en format ASCII. Souvent les fichiers STL sont disponibles sous format binaire car moins gourmand en espace de stockage. Ces fichiers, même ouverts dans un éditeur comme Notepad ne sont pas “lisibles” pour un humain contrairement aux fichiers au format ASCII.

Si vous souhaitez creuser la question, vous pouvez consulter cet article wikipedia.

Est-ce qu'un fichier STL peut être modifié ?

La première “modification” d'un fichier STL constite à changer son échelle sur l'un ou plusieurs de ces axes X, Y, Z. Vous l'avez probablement déjà fait dans votre logiciel de tranchage. Il ne s'agit pas en réalité d'une véritable modification. Aucun triangle n'est supprimé ou ajouté, on agit juste en fait sur les “proportions” du maillage.

Il est possible de réduire le nombre de triangles. On parle alors de mode polylow pour low polygon. Cela ne s'applique pas à tous les modèles. Réduire le nombre de triangles d'un tétraèdre est impossible, pour d'autres modèles cela n'aura pas de sens. Cette transformation lowpoly est en fait intéressante pour les formes organiques. La réduction du nombre de triangles donne alors un rendu styilisé. La réduction du maillage de triangles peut également servir à réduire le poids du fichier pour un aperçu rapide de l'objet.

D'autres modifications sont possibles comme par exemple la fusion avec d'autres volumes, la déformation, la symétrie, etc. Si vous voulez creuser cet aspect vous pouvez vous tourner vers des logiciels comme Meshlab. Notez que ce logiciel peut aussi permettre de réparer les fichiers STL. Il est en effet assez courant de devoir retraiter un fichier STL afin de pallier à des zones ouvertes dans le modèle, des faces inversées ou encore à des arêtes ouvertes.

Certains logiciels permettent eux de subdiviser un triangle en plusieurs sous triangles. En jouant sur les coordonnées de ces nouveaux triangles il est alors possible de produire un nouveau volume, avant défini par un seul triangle. Cette technique peut être utilisée pour "lisser" une surface. On peut alors parler de mode highpoly.

STL : le format idéal ?

Le format STL est léger et simple. C'est sans doute ce qui explique sa très large diffusion et compatibilité. Cette simplicité explique aussi les limites de ce format. Un fichier STL ne comporte pas de couleur ni de texture. Il ne présente pas d'unité en tant que telle (voir le code ci-dessus).

Autre limite, aussi fine que la résolution soit possible, ce format ne permet pas de définir de véritables courbes au sens mathématique du terme. Toute courbe sera en fait une section de triangles plus ou moins petits.

Pour pallier à ces faiblesses, un nouveau format a récemment vu le jour sous le nom 3MF. Bien que ce format soit mis en avant par les fabricants d'imprimantes 3D et que ces avantages soient réels, le format STL reste encore largement utilisé.

Et ensuite ?

La majorité des imprimantes 3D n'accepte pas directement un fichier au format STL. Pour que l'imprimante 3D puisse imprimer le modèle, le fichier STL doit être transformé en "chemin" de la tête d'impression. C'est le rôle des logiciels de tranchage qui convertissent le fichier stl généralement en gcode. Le fichier gcode comporte des instructions de déplacements (et de chauffe de la buse et du plateau d'impression).

Voilà vous savez tout ou presque sur le format de fichier STL !