

## Практическая работа №5

### Влияние параметров STL файла 3D модели на размерную и геометрическую точность образцов, полученных методом послойного формирования

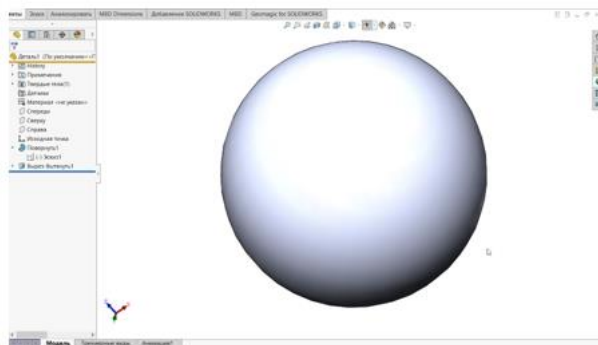
**Цель работы:** ознакомление с основными форматами файлов 3D моделей, применяемых для 3D печати, изучение параметров STL файлов 3D моделей влияющих на размерную и геометрическую точность образцов, полученных методом послойного формирования

#### Теоретическая часть:

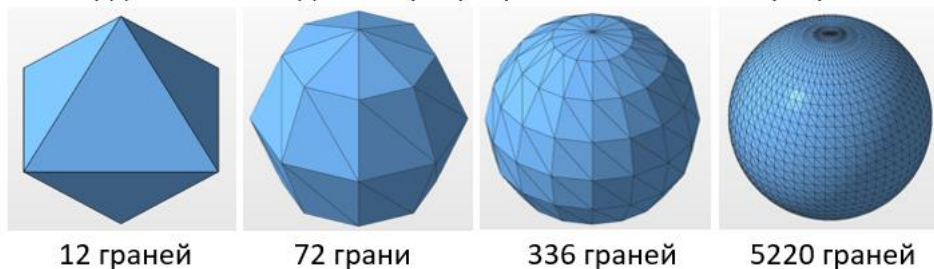
Размерная, геометрическая точность образцов, шероховатость поверхности, получаемых методом послойного выращивания, напрямую зависит от используемого программного обеспечения, от его функциональных возможностей, от корректности работы программного пакета и компьютерной грамотности проектирования 3D моделей.

Существует более 30 форматов файлов для 3D-печати: STL, OBJ, PLY, 3MF, AMF, VRML и др. Наиболее распространенным файлом 3D модели для последующей 3D печати безусловно является STL. Наименование данного файла происходит от сокращения слова Stereolithography (данного 3D Systems), однако иногда его называют «стандартным языком треугольников» или «стандартным языком тесселяции».

Файл STL хранит информацию 3D-модели. Формат представляет собой комбинацию элементов для двумерной внешней поверхности 3D-моделей с использованием полигонов (треугольников) в заданной последовательности. Например, квадрат может быть описан как два треугольника; куб (шесть квадратов) как 12 треугольников. Искривленные поверхности нуждаются в большем количестве треугольников для их описания. Чем выше отклонение (для гладких поверхностей), тем большее количество треугольников необходимо (рис. 1). В результате, описание высококачественных объектов требует достаточно объемный файл.



Твердотельная модель шара, разработанная в CAD программе



12 граней

72 грани

336 граней

5220 граней

Модели шара, сохранённые в формате STL с различным количеством полигонов

Рисунок 1 – Сравнение твердотельной модели шара с различными вариантами его тесселяции в STL файле

Формат файла STL аппроксимирует поверхность модели CAD треугольниками. Аппроксимация никогда не бывает идеальной, а грани вносят отклонения от геометрических показателей качества в модель. Использование небольшого числа треугольников приводит к формированию грубой модели и, наоборот, большее количество треугольников формирует более точную поверхность (с более плавным приближением). Идеальная сферическая поверхность аппроксимируется тесселяциями.

Практически все программные продукты CAD способны генерировать файл STL из разработанной 3D-модели. При сохранении файла 3D модели в данный формат необходимо установить требуемое разрешение. Для этого необходимо настроить высоту хорды (отклонение, допуск) и угловое отклонение (угловой допуск).

Высота хорды – это максимальное расстояние от поверхности оригинального дизайна 3D модели и сетки STL (рис. 2 а).



Рисунок 2 – Параметры, влияющие на разрешение 3D модели

Чем меньше высота хорды, тем точнее грани представляют фактическую поверхность модели. Рекомендуется установить допуск от 0,01 до 0,001 мм. Это обычно приводит к оптимальным показателям качества 3D печати. Более низкий диапазон допуска не оказывает существенное влияние на разрешение 3D модели в формате STL, поскольку 3D-принтеры не могут печатать с таким уровнем детализации.

Угловой допуск – это угол между нормальными векторами соседних треугольников (рис. 2 б). Данный показатель ограничивает угол между нормальными векторами соседних треугольников. Угол по умолчанию обычно устанавливается в 10-15 градусов. Уменьшение допуска улучшает разрешение печати.

Формат файла STL обеспечивает два разных способа хранения информации о треугольных гранях, которые описывают поверхность объекта. Они называются кодировкой ASCII и двоичным кодированием. В обоих форматах хранится информация о каждом полигоне: координаты вершин, компоненты единичного вектора нормали к треугольнику. Вектор нормали должен указывать наружу относительно трехмерной модели.

Формат файла ASCII STL

Файл ASCII STL начинается с обязательной строки: `solid <name>`, где `<name>` – это название 3D-модели. Файл продолжается информацией о покрывающих треугольниках. Информация о вершинах и векторе нормали представлена следующим образом:

- `facet normal nx ny nz`
- `outer loop`
- `vertex v1x v1y v1z`
- `vertex v2x v2y v2z`
- `vertex v3x v3y v3z`
- `endloop`

- endfacet

Здесь  $n$  – нормаль к треугольнику, а  $v1$ ,  $v2$  и  $v3$  — вершины треугольника. Значения координат представлены в виде числа с плавающей, например, «3.245000e-002». Файл заканчивается обязательной строкой:

- endsolid <name>

Бинарный формат файла STL

Если в тесселяции много маленьких треугольников, файл ASCII STL может стать огромным. Вот почему существует более компактная двоичная версия. Двоичный файл STL начинается с 80-символьного заголовка. После заголовка общее количество треугольников указывается с использованием 4-байтового целого числа без знака.

- UINT8[80] – Заголовок
- UINT32 – Количество треугольников

Каждый треугольник представлен двенадцатью 32-битными числами с плавающей запятой. Как и в файле ASCII STL, 3 числа относятся к трехмерным декартовым координатам нормали к треугольнику. Остальные 9 чисел предназначены для координат вершин (по три в каждой). Вот как это выглядит:

- foreach triangle
- REAL32[3] – Нормальный вектор
- REAL32[3] – Вершина 1
- REAL32[3] – Вершина 2
- REAL32[3] – Вершина 3
- UINT16 – конец подсчета байтов атрибута

После каждого треугольника есть 2-байтовая последовательность, называемая «счетчиком байтов атрибута». В большинстве случаев это значение равно нулю и действует как разделитель между двумя треугольниками.

Бинарный формат всегда рекомендуется для 3D-печати, поскольку он приводит к уменьшению размера файла. Однако, если необходимо проверить файл STL на предмет отладки, тогда ASCII предпочтительнее, поскольку его легче читать.

Спецификация STL имеет некоторые специальные правила для тесселяции и для хранения информации.

1. Правило вершин – каждый треугольник должен иметь две вершины с соседними треугольниками (рис. 3). Это правило должно соблюдаться при проверке поверхности трехмерного объекта.

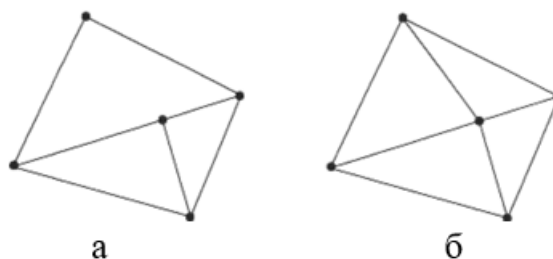


Рисунок 3 – Варианты тесселяции области 3D модели

Вариант объединения полигонов на рисунке 3 а является недопустимой тесселяцией, тогда как рисунок 3 б приемлем.

2. Правило ориентации – ориентация фасета должна быть указана двумя способами. Во-первых, направление нормали должно указывать наружу. Во-вторых, вершины перечислены в порядке против часовой стрелки, если смотреть на объект снаружи (правило правой руки) (рис. 4).

Ориентация каждой грани задается двумя способами: направлением вектора нормали и упорядочением вершин, что помогает обеспечить согласованность данных и выявлять поврежденные данные.

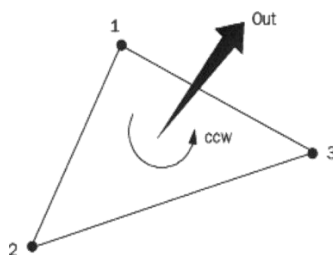


Рисунок 4 – Правило ориентации

Программное обеспечение может, например, рассчитать ориентацию по нормали, а затем по вершинам и проверить, совпадают ли они. Если это не так, то он может объявить файл STL поврежденным!

3 Правило положительных октантов – координаты вершин треугольника должны быть положительными. Это подразумевает, что трехмерный объект находится в полностью положительном октанте трехмерной декартовой системы координат. Октант I (красный) — полностью положительный октант (рис. 5).

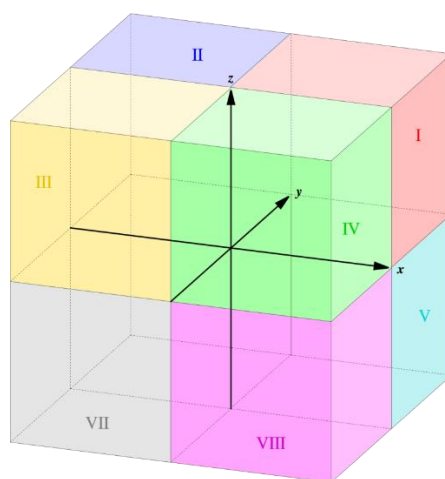


Рисунок 5 – Октанты трехмерной декартовой системы координат

Основанием для этого правила является экономия места. Если бы трехмерному объекту было позволено находиться где-либо в координатном пространстве, пришлось бы иметь дело с отрицательными координатами. Для хранения отрицательных координат необходимо использовать числа с плавающей запятой со знаком, требующих один дополнительный бит для хранения знака (+/-). Так как все координаты положительны, это правило гарантирует, что можем использовать числа без знака для координат и сохранять бит для каждого значения координаты, которое мы храним.

Преимущества формата данных STL:

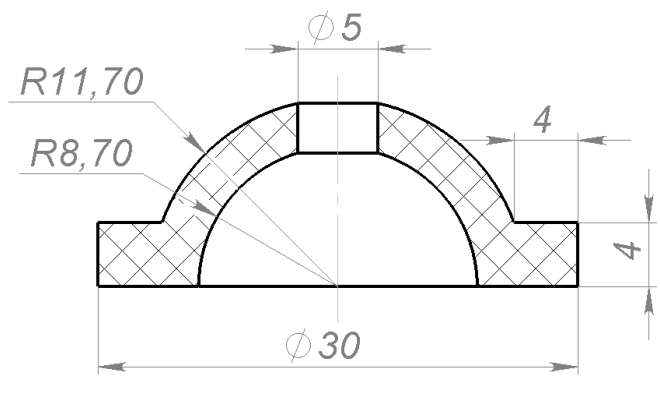
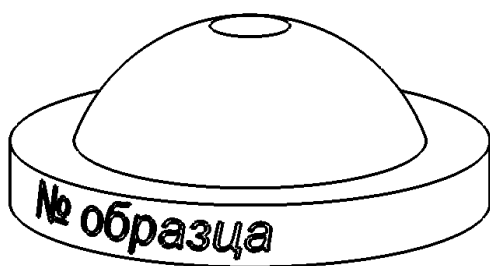
1. Простота структуры данных формата – меньшие размеры файлов, более быстрая обработка
2. Универсальность – поддерживается почти всеми 3D-принтерами.
3. Доступность – практически все программные пакеты позволяют тесселировать твердотельную модель в данный формат
4. Зрелая экосистема – множество программных пакетов работающих с данным форматом, громадная база 3D моделей в данном формате в ресурсах интернет.

### Недостатки формата данных STL:

1. Формат файла STL не может хранить дополнительную информацию, такую как цвет, материал и т. д., в отличие от формата файла OBJ, в котором можно хранить цветовые и текстурные профили. Однако существуют нестандартные версии формата STL, которые действительно способны передавать информацию о цвете. Например, программные пакеты VisCAM и Solidview используют «число байтов атрибута» в конце каждого треугольника для хранения 15-битного цвета RGB
2. В случае, когда точность печати должна соответствовать разрешению в микронном масштабе, количество треугольников, необходимое для описания гладких изогнутых поверхностей, может привести к огромным размерам файлов.
3. Формат файла не позволяет включить в базу данных метаданные (информация об авторстве и авторских правах).

### Практическая часть:

Используя программные продукты для разработки CAD объектов построить четыре 3D модели образцов по геометрическим параметрам, представленных на эскизе.



Построенные модели сохранить в формате списка треугольных граней и нормалей в следующем разрешении:

Образец 1 – грубое

Образец 2 – точное

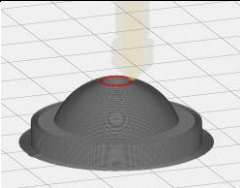
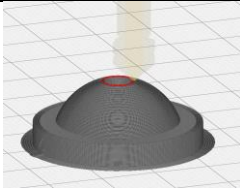
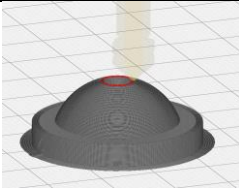
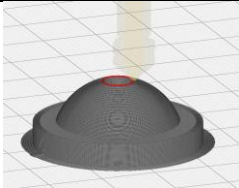
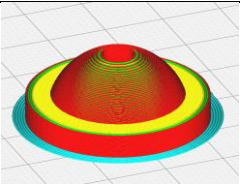
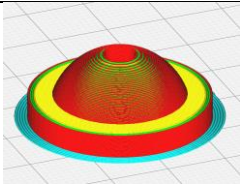
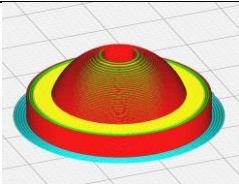
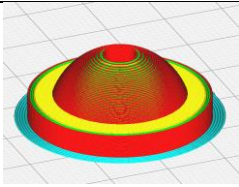




Образец 3 – максимально возможная точность, обеспечиваемая CAD программой

Образец 4 – оптимизированное, основанное на показателях допусков первых трех образцов

В слайсере Ultimaker Cura подготовить файлы к 3D печати с использованием сопла 0,4 мм и построением изделия с чередующейся стенкой для повышения прочности. Высота слоя - 0,2 мм, количество линий стенки - 2, слои крышки - 4, слои дна - 4, нижний шаблон начального слоя - зигзаг, плотность заполнения - 9 %, шаблон заполнения – сетка, скорость печати – 50 мм/с, тип прилипания к столу – кайма, количество линий каймы – 5. Остальные показатели качества построения печатного изделия оставить по умолчанию.

В отчете представить изображения твердотельной CAD модели, stl образца после слайсинга, распечатанного образца. Используя текстовый редактор «Блокнот», программный продукт в котором разрабатывались CAD объекты произвести сравнение четырех моделей образцов.

Представить информацию о первом и последнем фасете 3D модели образца 4 в файле STL в кодировке ASCII.

Параметр	Образец 1	Образец 2	Образец 3	Образец 4
Изображение твердотельной CAD модели				
Изображение stl образца после слайсинга				
Изображение распечатанного образца				
Количество треугольников	1910	1910	384866	10722
Допуск отклонения	0.05251362	0.00218807	0.00218807	0.00758807
Допуск угла	30.0000000	0.50000	0.50000	4.83000
Размер файла в кодировании ASCII	540554	12706954	108917102	3034350
Размер файла в двоичном кодировании	95584	224584	19243384	192443384
Время печати, м.	15	15	15	15
Масса образца, гр.	3	3	3	3

Представлена информация о первом и последнем фасете 3D модели образца 4 в файле STL в кодировке ASCII.:

```
facet normal 4.114027e-02 -9.476155e-01 3.167529e-01
  outer loop
    vertex 1.619281e+01 1.833313e+01 3.826577e+01
    vertex 1.616325e+01 1.833313e+01 3.826961e+01
    vertex 1.608848e+01 1.814606e+01 3.771965e+01
  endloop.....
facet normal 9.947751e-01 0.000000e+00 -1.020903e-01
  outer loop
    vertex 3.144599e+01 1.192668e+01 3.947489e+01
    vertex 3.139274e+01 1.116667e+01 3.895601e+01
    vertex 3.139274e+01 1.268670e+01 3.895601e+01
  endloop
endfacet
endsolid
```

#### Вопросы:

1. Основные форматы файлов для 3D-печати
2. Характеристика файла STL
3. Регулирование разрешения файла STL
4. Что такое высота хорды, угловой допуск?
5. Кодировки файла STL
6. Характеристика файла ASCII STL
7. Характеристика бинарного формата файла STL
8. Правила для тесселяции информации файла STL
9. Преимущества формата данных STL
10. Недостатки формата данных STL