

ФОРМИРОВАНИЕ ПОСЛОЙНЫХ КОНТУРОВ 3D-МОДЕЛЕЙ ДЛЯ АДДИТИВНОГО ПРОИЗВОДСТВА

Ракитин С.Ю., Илькубаев А.А.

Оренбургский государственный университет, г. Оренбург

Формирование управляющей программы для аддитивного производства существенно отличается от генерации программы для станков с числовым программным управлением (ЧПУ) традиционного производства.

Процесс генерации управляющей программы при традиционном производстве заключается в том, что станок с ЧПУ убирает от заготовки лишний материал. Это происходит при использовании механической обработки (сверлении, точении, фрезеровании и т.д.). Управляющая программа содержит команды G-кода.

G-код – условное наименование языка программирования устройств с числовым программным управлением [1]. Используя эти операторы, система ЧПУ выдает команды станку – насколько необходимо переместиться, по каким именно осям, какой привод или механизм включить/выключить.

Процесс формирования управляющей программы при аддитивном производстве выглядит несколько иначе. Аддитивное производство – процесс послойного «выращивания» физического объекта, параметры которого заданы виртуальной моделью. Как правило, при таком производстве используются не станки с ЧПУ, а 3D – принтеры. Это периферийное устройство, использующее метод послойного создания физического объекта по цифровой 3D – модели [2]. Как видно из определения аддитивного производства, управляющая программа должна содержать команды, которые позволяют 3D-принтеру «вырастить, напечатать» модель.

За формирование программ для станков, 3D-принтеров и т.д. используются САМ-системы. САМ (англ. Computer-aided manufacturing) – автоматизированная система, либо модуль автоматизированной системы, предназначенный для подготовки управляющих программ для станков с ЧПУ, ориентированная на использование ЭВМ [3].

При генерации управляющей программы для аддитивного производства необходимо проанализировать 3D – модель на каждом слое. Дискретность слоя зависит от требуемой точности и технической возможности реализации, например, 0.3 и 0.125 мм. При анализе слоя, требуется найти контур от пересечения текущей плоскости с 3D-моделью. Математическое описание этого процесса приведено ниже.

В качестве формата 3D – модели выбран формат STL. STL (от англ. stereolithography) – формат файла, широко используемый для хранения трехмерных моделей объектов для использования в технологиях быстрого прототипирования, обычно, методом стереолитографии [4]. Информация об объекте хранится как список треугольных граней, которые описывают его поверхность, и их нормалей. STL – файл может быть текстовым (ASCII) или двоичным (бинарным).

ASCII STL файл начинается со строки: `solid name`,

Где `name` – необязательная строка (если `name` опущено, всё равно должен быть пробел после `solid`). Файл продолжается с любым числом треугольников, описываемых следующим способом:

```
facet normal  $n_x$   $n_y$   $n_z$ 
```

```
outer loop
```

```
vertex  $v1_x$   $v1_y$   $v1_z$ 
```

```
vertex  $v2_x$   $v2_y$   $v2_z$ 
```

```
vertex  $v3_x$   $v3_y$   $v3_z$ 
```

```
endloop
```

```
endfacet
```

где каждый `n` или `v` число с плавающей запятой в виде знака–мантиссы или 'e'–знак – экспоненты, то есть "`-2.648000e-002`". Файл завершается строкой:

```
endsolid name
```

Наиболее простым в реализации является считывание данных о модели из текстового формата файла (ASCII), хотя он обладает в среднем в 3 раза большим объемом.

Перед тем как начать искать точки пересечения необходимо найти высоту будущей модели. Для этого необходимо перебрать все координаты треугольников и найти минимальное и максимальное значения по координате `Z`, после чего требуется найти их разность.

$$H = Z_{max} - Z_{min}, \quad (1)$$

где Z_{max} – максимальная координата по `Z`;

Z_{min} – минимальная координата по `Z`;

H – высота будущей детали.

Значения H определяет высоту детали, но не указывает, сколько раз требуется выполнить сечение плоскостью. Для этого требуется разделить это число на высоту одного слоя.

$$K = \frac{H}{S}, \quad (2)$$

где H – высота будущей детали;

S – высота одного слоя;

K – количество слоев.

Секущая плоскость должна быть параллельна той плоскости, на которой располагается 3D-модель при печати. В примере модель расположена на плоскости XY . Тогда уравнение секущей плоскости будет иметь вид [5]:

$$C_z + D = 0, \quad (3)$$

где C_z – составляющая нормального вектора плоскости;

D – высота от начала координат до секущей плоскости.

Так как STL-формат, аппроксимирует модель в виде треугольников, то для получения текущего контура модели необходимо искать точки пересечения сторон треугольника с секущей плоскостью (рисунок 1).

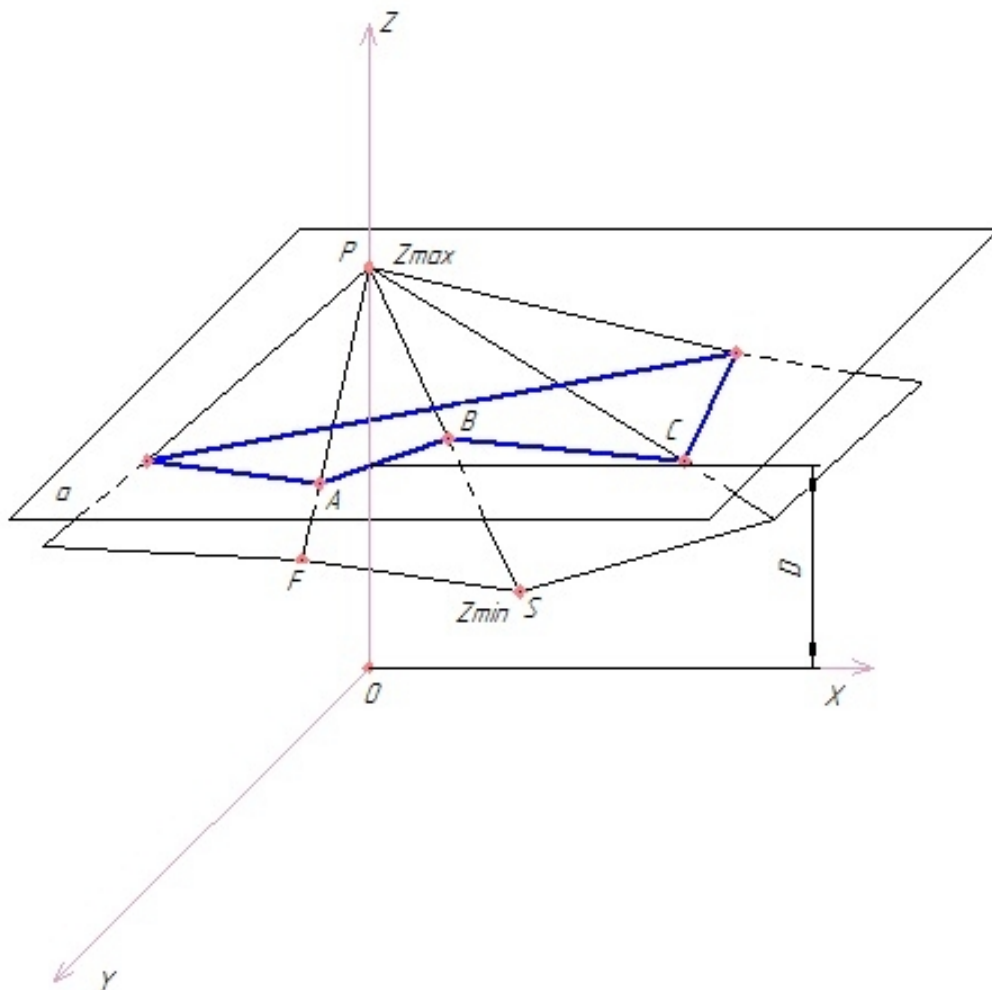


Рисунок 1 – Пример формирования контура

Под контуром понимается будущий слой модели, без учета внутреннего заполнения, состоящий из замкнутых отрезков. Рассмотрим треугольник FPS .

Как видно из рисунка 1 он пересекает секущую плоскость в точках A и B . Прежде чем искать точки пересечения отрезка с секущей плоскостью необходимо проверить условия (данные условия действуют для отрезка PF):

$$\left((D < P_z) \text{ и } (D > F_z) \right) \text{ или } \left((D > P_z) \text{ и } (D < F_z) \right), \quad (4)$$

где D — высота от начала координат до секущей плоскости;

P_z — координата z точки P ;

F_z — координата z точки F .

Эти условия позволяют определить, будет ли иметь данный отрезок точку пересечения с секущей плоскостью. Для определения координат точки пересечения A необходимо воспользоваться формулами:

$$X_A = X_P + (X_F - X_P) \cdot \frac{D - Z_P}{Z_F - Z_P}, \quad (5)$$

$$Y_A = Y_P + (Y_F - Y_P) \cdot \frac{D - Z_P}{Z_F - Z_P}, \quad (6)$$

$$Z_A = Z_P + (Z_F - Z_P) \cdot \frac{D - Z_P}{Z_F - Z_P}, \quad (7)$$

где X_A, Y_A, Z_A — координаты будущей точки A которая принадлежит контуру;

X_P, Y_P, Z_P — координаты точки P , которая является началом отрезка, над которым происходит операция;

X_F, Y_F, Z_F — координаты точки F , которая является концом отрезка, над которым происходит операция;

D — высота от начала координат до секущей плоскости.

После того как деталь будет полностью проанализирована, получаем массив точек для каждого слоя (таблица 1).

Таблица 1 – Массив точек для каждого слоя

Номер треугольника	1		2		3	
Номер точки пересечения	1	2	1	2	1	2
X , мм	15.20	15.17	15.17	15.07	15.07	14.86
Y , мм	15.00	15.10	15.10	15.18	15.18	15.13
Z , мм	0.50	0.50	0.50	0.50	1.00	1.00
Номер слоя	1	1	1	1	2	2

Рассмотрим особенности массива точек. Как видно из таблицы 1, координата Z для каждого слоя является константой. Это объясняется тем, что все точки слоя находятся на одинаковом удалении от начала координат. Вторая особенность заключается в том, что у двух рядом лежащих треугольников одна общая сторона, поэтому у первого треугольника 2 точка пересечения имеет одинаковые координаты с 1 точкой пересечения второго треугольника.

Теперь необходимо из этих точек построить контур для каждого слоя. В качестве примера рассмотрим модель, вид сверху которой представлен на рисунке 2.

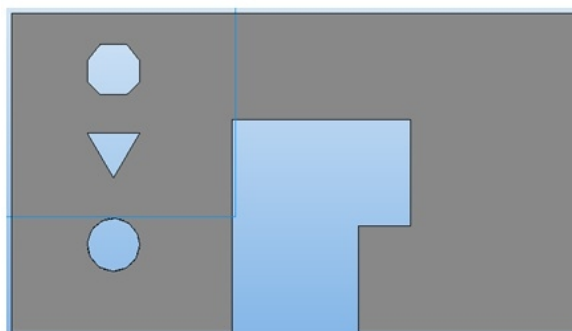


Рисунок 2 – Модель детали со сквозными отверстиями разной формы

Для этого требуется соединить точки в определенной последовательности. Рассмотрим последовательность, при которой точки соединяются по следующей формуле:

$$A(X_n, Y_n), B(X_{n+1}, Y_{n+1}), \quad (8)$$

где A – начало отрезка;

B – конец отрезка;

n – число всех точек пересечения.

На рисунке 3 показано как будет выглядеть контур слоя, если его соединить по формуле описанной выше.

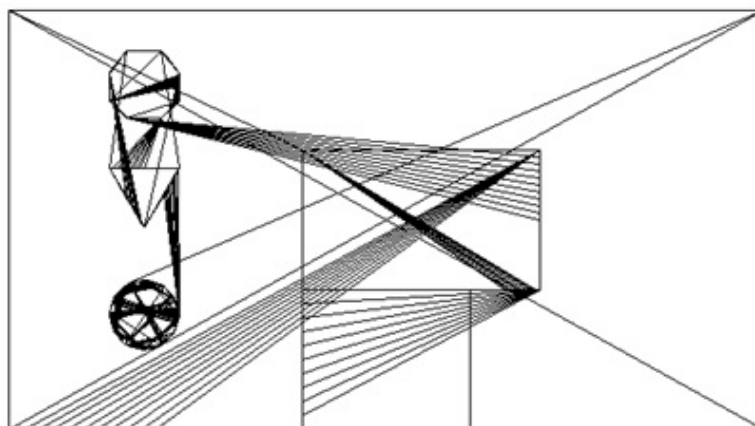


Рисунок 3 – Контур, полученной по первой последовательности

Как видно из рисунка 3 на контуре появляются лишние отрезки. Это связано с тем, что при формировании контура необходимо знать не только все его точки, но и последовательность их соединения, а так же принадлежность к внешнему или одному из возможных внутренних. Таким образом, соединение точек в порядке их нахождения недопустимо.

Для решения данной проблемы воспользуемся тем, что секущая плоскость пересекает треугольник в двух точках, а у двух рядом лежащих треугольников, есть общее ребро, это значит, что точка пересечения этого отрезка с секущей плоскостью будет одинаковой. Исходя из этих особенностей, получаем последовательность:

$$A(X_{v*2-1}, Y_{v*2-1}), B(X_{v*2}, Y_{v*2}), \quad (9)$$

где **A** – начало отрезка;

B – конец отрезка

$$v = n/2, \quad (10)$$

где **n** – число всех точек пересечения.

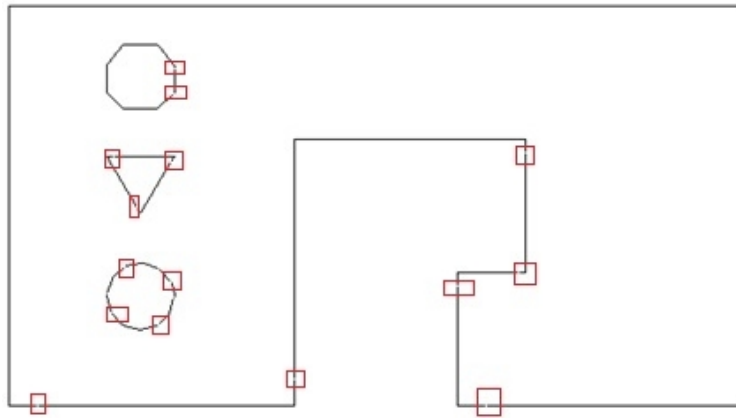


Рисунок 4 – Контур, полученный по второй последовательности

Как видно из рисунка 4, лишние отрезки отсутствуют. Но возникла следующая проблема. Прямоугольниками на рисунке 4 выделены пустые точки, то есть контур не замкнутый. Это объясняется ошибками в округлении при операциях с числами с плавающей точкой, а так же вероятностью расположения вершины треугольника непосредственно на плоскости. Для решения этой проблемы необходимо изменить условия (4) следующим образом:

$$\left((D \leq P_z) \text{ и } (D \geq F_z) \right) \text{ или } \left((D \geq P_z) \text{ и } (D \leq F_z) \right), \quad (11)$$

где D – высота от начала координат до секущей плоскости;

P_z – координата z точки P ;

F_z – координата z точки F .

Теперь под условия попадают отрезки, которые пересекают плоскость, и отрезки начало или конец которых лежит в плоскости. Для правильного соединения массива точек необходимо ввести принадлежность точек к контуру. То есть, построение будет осуществляться для каждого слоя отдельно. Таким образом, получаем третью последовательность:

$$A_{ij}(X_{n_j}, Y_{n_j}), B_{ij}(X_{n_{j+1}}, Y_{n_{j+1}}), \quad (12)$$

где A_{ij} – начало i – го отрезка, на j слое;

B_{ij} – конец i – го отрезка, на j слое;

n_j – число точек пересечения на j слое.

На рисунке 5 показано как будет выглядеть контур слоя, если его соединить по формуле описанной выше.

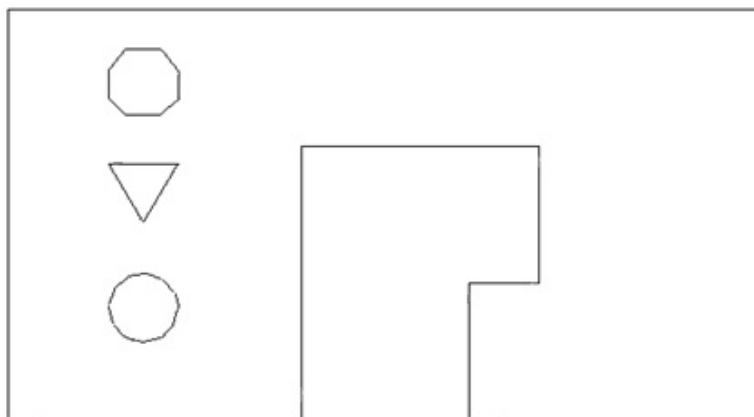


Рисунок 5 – Контур, полученный по третьей последовательности

Третья последовательность позволяет удалить лишние отрезки и обеспечивает необходимое условие замкнутости контура. Полученный контур полностью идентичен модели, показанной на рисунке 2.

Список литературы

1. *G-code* / Свободная энциклопедия – Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/G-code>. – 30.11.2015.
2. *3D-принтер* / Свободная энциклопедия – Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/3D-принтер>. – 30.11.2015.
3. *CAM* / Свободная энциклопедия – Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/CAM>. – 30.11.2015.
4. *STL (формат файла)* / Свободная энциклопедия – Википедия. – Режим доступа: [https://ru.wikipedia.org/wiki/STL_\(формат_файла\)](https://ru.wikipedia.org/wiki/STL_(формат_файла)). – 30.11.2015.
5. *Плоскость. Прикладная математика, справочник математических формул, примеры и задачи с решениями.* – Режим доступа: <http://www.pm298.ru/plosk.php>. – 30.11.2015.