

*Ильичев В. Ю., к.т.н., доцент кафедр «Тепловые двигатели и гидромашины»,
«Мехатроника и робототехнические системы»*

*Калужский филиал ФГОУ ВО «Московский государственный технический
университет имени Н.Э. Баумана (национальный исследовательский
университет), г. Калуга, Россия*

*Суркова П. В., студент кафедры «Тепловые двигатели и гидромашины»
Калужский филиал ФГОУ ВО «Московский государственный технический
университет имени Н.Э. Баумана (национальный исследовательский
университет), г. Калуга, Россия*

СОЗДАНИЕ ПАРАМЕТРИЧЕСКИХ ЧЕРТЕЖЕЙ В САПР FREECAD С ИСПОЛЬЗОВАНИЕМ ИХ СКРИПТОВОГО ОПИСАНИЯ НА ЯЗЫКЕ PYTHON

Аннотация: В статье описаны средства и методы применения скриптов на языке Python для создания объектов машинной графики разной степени сложности в САПР FreeCAD. Также рассмотрены возможности расширения функций данной системы автоматизированного проектирования как путём связывания её с другими известными программными продуктами, так и с помощью создания и вычисления расчётных моделей в той же программе на Python, с помощью которой выводятся графические результаты работы. Описаны и прочие современные технологии, составляющие рассмотренную новую концепцию автоматизированного проектирования.

Ключевые слова: машинная графика, FreeCAD, библиотека Part, модуль Loft, метод конечных элементов, язык Python.

Annotation: The article describes the means and methods of using Python scripts to create machine graphics objects of varying degrees of complexity in

FreeCAD. Also, the possibilities of expanding the functions of this computer-aided design system are considered, both by linking it with other well-known software products, and with the help of a co-building and calculating models in the same program on Python, with the help of which graphic results of work are displayed. Other modern technologies that make up the new concept of computer-aided design are described.

Keywords: machine graphics, FreeCAD, Part library, Loft module, finite element method, Python language.

Введение

Разработка двух- и трёхмерных чертежей однотипных объектов всё чаще осуществляется с использованием технологий параметрического (с заданием исходных параметров) [1] скриптового (с использованием компьютерных программ) проектирования [8]. В данном случае для получения ряда похожих моделей достаточно изменять лишь некоторые числовые (и возможно логические) значения переменных и после этого перезапускать созданную программу, а получение разных визуальных результатов при этом осуществляется автоматически.

Недавно была создана и всё более широко применяется бесплатная система автоматизированного проектирования (САПР), которая так и называется FreeCAD [7], обладающая уникальным сочетанием достоинств:

- большим набором встроенных средств создания растровой и векторной (являющейся основной) графики [2], как 2D, так и 3D;
- универсальностью создания объектов для разных отраслей: машиностроения, дизайна, искусства и многих других;
- возможностью подготовки деталей для 3D печати [3], расчёта с использованием метода конечных элементов, создания сложных сборок;
- предусмотрено подключение дополнительных программных модулей для выполнения специфических задач, например, можно использовать дополнительное ядро для создания качественной графики OpenCascade [12],

современнейшую библиотеку трёхмерной графики Coin3D [6], средства изменения и дополнения интерфейса пользователя, один из популярнейших языков программирования Python [4];

- наличием множества форматов импорта и экспорта объектов, как популярных, так и довольно экзотических [10];

- существуют версии FreeCAD для всех популярных операционных систем: Windows, Linux, MacOS и других (т.е. данная САПР обладает свойством мультиплатформенности);

- выполнение булевых (логических) операций при организации взаимодействия объектов – выдавливания, вращения, суммирования, вычитания, создания вырезов, «натягивания» текстур и огромного количества прочих.

Кроме всего перечисленного, можно отметить, что создание моделей во FreeCAD позволяет использовать множество новых инновационных подходов, во многом концептуально отличных от традиционных. Целью данного исследования является изучение и демонстрация главных принципов проектирования во FreeCAD с использованием средств программной автоматизации, реализованных на языке Python, так как в имеющейся литературе данная тема рассматривается крайне скудно.

Материал и методы исследования

Команды на языке Python можно выполнять во FreeCAD как пошагово, с использованием программной консоли, так и путём написания и запуска макросов. Хотя специфика использования Python в данной САПР частично описана в документации к программе [5], достаточно успешно можно изучать её путём традиционного создания объектов и произведения операций с ними в документе, с наблюдением и копированием некоторых команд, отображаемых при этом в консоли.

Во FreeCAD используется концепция использования множества программных модулей, в интерфейсе именуемых «верстаками»; для использования возможностей каждого из них в программу на Python

необходимо добавить соответствующую библиотеку функций. Например, для создания нового документа необходимо подключить хотя бы один модуль – главную библиотеку программы FreeCAD, а затем написать строку создания данного документа, и одновременного присваивания его идентификатора созданной переменной (для упрощения дальнейшего кода):

```
import FreeCAD
doc=FreeCAD.newDocument()
```

После запуска данного простейшего скрипта (макроса) на экране появляется фон документа, инструменты навигации и стрелки направления осей координат. Рассмотрим принцип создания в документе новых объектов и действий с ними для получения вначале 2D, а затем на её основе и 3D графики.

При создании наиболее широко используемых в простой графике первичных геометрических объектов (например, плоскости, куба, цилиндра, конуса и др.) можно использовать верстак «Part». Для этого следует импортировать программный модуль верстака и задать параметры объекта. Например, при создании плоскости (Box) по умолчанию (с размерами 10x10 мм в системе координат xy) это выглядит следующим образом:

```
import Part
plate = doc.addObject("Part::Box", "Plate")
doc.recompute()
```

Последняя строка необходима для обновления экрана при выводе изображения (без неё плоскость не появится на чертеже). Результат выполнения представленных команд представлен на рис. 1.

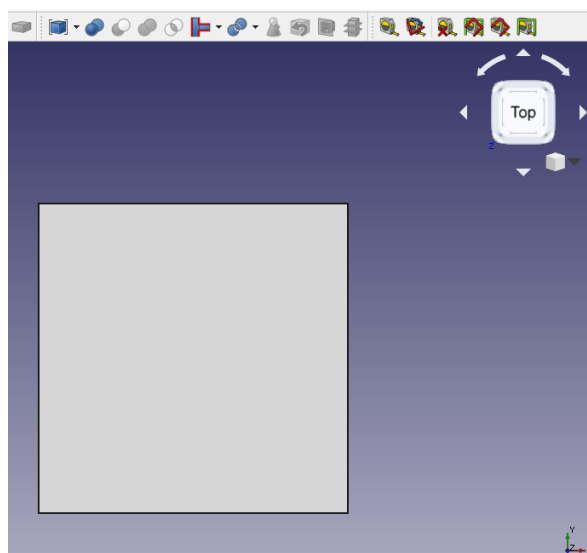


Рис. 1. Результат создания нового документа и отрисовки плоскости

Подобным же образом на чертёж можно добавить другие объекты из верстака Part. Рассмотрим теперь более сложную задачу – создание своих, произвольных 2D объектов, а затем преобразование их сочетания в 3D модель.

В данном случае используется следующая последовательность действий: вначале создаются 2D эскизы (скетчи) в каждом сечении по высоте модели, а затем они объединяются каким-либо из выбранных способов в объёмную модель. Возможность рисования скетчей с помощью скрипта Python осуществляется также с помощью команд библиотеки Part (но в интерфейсе программы необходимо переключиться на верстак Sketcher). В программе Python создание нового эскиза осуществляется с помощью команды:

```
part1=doc.addObject('Sketcher::SketchObject','Sketch')
```

После этого добавляются команды выбора расположения плоскостей эскизов и создания на них примитивов (в интерфейсе FreeCAD для этого существует огромное количество инструментов). Например, на рис. 2 изображён эскиз треугольника.

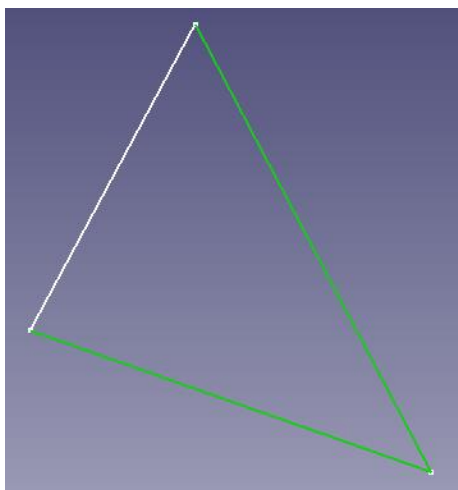


Рис. 2. Эскиз треугольника, объединяющий три отдельных линии с помощью наложения граничных связей.

В программе каждая линия данного примитива задаётся в виде координат начала и конца каждой из трёх составляющих его линий, а также связей между ними (Constraints), которые создаются автоматически. В данном случае связями (ограничениями) являются совпадения координат конца каждого отрезка с координатами начала следующего (точечная связь воспринимается как одна точка). Данные ограничения в концепции FreeCAD являются необходимыми для замыкания контуров фигур и тем самым исключения появления артефактов при их использовании для создания более сложных комплексных объектов.

Подобным же образом в другой плоскости (с иной координатой z) создан эскиз второго треугольника, а затем к двум примитивам применена операция их соединения Loft [3]. Результат, - трёхмерная пирамидальная фигура, приведён на рис. 3.

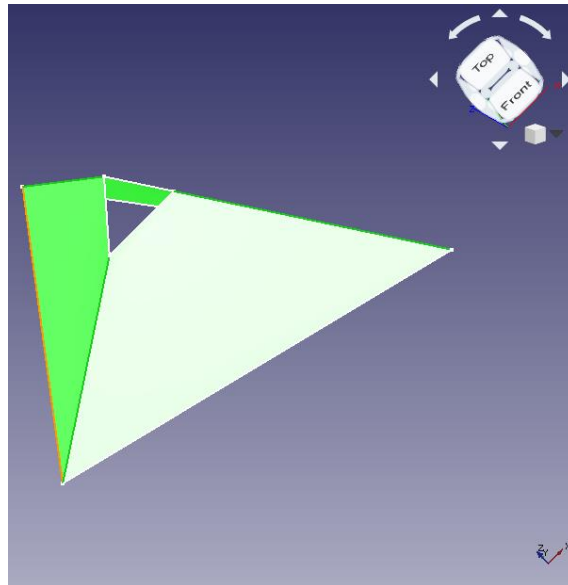


Рис. 3. Результат создания 3D пирамиды из двух треугольников с помощью операции Loft.

Видно, что программа сама соединяет вершины треугольников плавными линиями (это и есть смысл алгоритма Loft), несмотря на то, что треугольники несколько повернуты относительно друг друга. С помощью верхней правой панели созданную фигуру можно повернуть для обозрения произвольным образом.

В более сложном алгоритме преобразования связи между элементами примитивов создаются принудительно. Например, эти операции необходимо совершать для соединения концов дуг при рисовании модели лопатки турбины. Связи дуг (Constraints) обозначены на рис. 4 точками; в результате, после применения к верхнему и нижнему сечению операции Loft, получена результирующая модель лопатки.

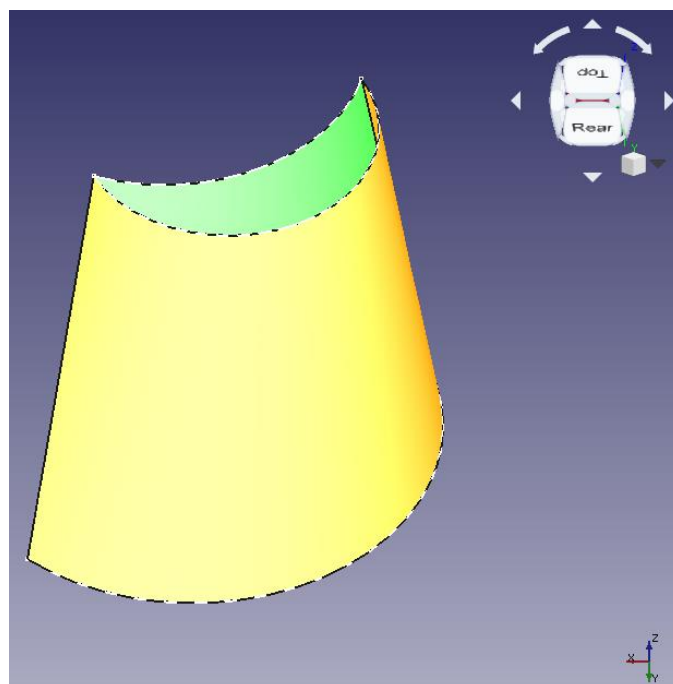


Рис. 4. Модель лопатки турбины, построенной с помощью скрипта на Python в САПР FreeCAD.

Развивая дальше идеи и программные коды, созданные и описанные в статье при создании сравнительно несложных объектов, можно создать модели и гораздо более сложных форм.

Заключение

В результате исследования рассмотрены средства языка Python, позволяющий создавать модели двух- и трёхмерных объектов в САПР FreeCAD. Для каждой модели написан программный код, с изменением значений переменных в котором можно легко изменять размеры и вид объектов. Можно также рассчитывать значения данных переменных по любой математической модели, реализация которой добавлена в программу, а после этого уже строить результирующие модели во FreeCAD.

Кроме того, коды на Python позволяют добавить новые функциональные возможности самой системе FreeCAD, а также соединять её с другими приложениями. Например, данная САПР позволяет организовать дальнейшее применение к созданным геометрическим объектам методов конечно-элементного анализа, например, для расчёта их на прочность или деформацию

[11] (в случае, если для этого не хватит возможностей такого моделирования в самом FreeCAD встроенными средствами FEM Workbench).

Для создания более «продвинутой» и эффектной графики с применением разного типа освещения, к FreeCAD с помощью модуля OpenSCAD Workbench можно подключить другую САПР с открытым исходным кодом OpenSCAD.

Очень полезным при моделировании роботов может оказаться верстак Robot совместно со специализированными модулями [9], для изучения работы, с которыми в документации имеются несколько примеров.

Рассмотренную САПР можно с успехом применять не только для обучения основам машинной графики, но и для создания и исследования сложных объектов. Без сомнения, технологии, представленные во FreeCAD, совместно с языком программирования Python, являются наиболее перспективными, и при этом совершенно бесплатными для использования. Любой исследователь может дополнить рассмотренную САПР библиотеками и модулями Python для достижения своих целей.

Библиографический список:

1. Ильичев В.Ю. Создание параметрических конечно-элементных трехмерных объектов с использованием функций Python. // Системный администратор. 2021. № 5 (222). С. 82-85.

2. Ильичев В.Ю. Использование рекурсивных функций для создания фрактальной графики средствами языка Python. // Системный администратор. 2021. № 3 (220). С. 92-95.

3. Ильичев В.Ю. Использование библиотеки ZenCAD языка Python для разработки универсальной методики создания объемных изделий. // Системный администратор. 2021. № 6 (223). С. 82-85.

4. Ильичев В.Ю. Разработка программы для исследования аттрактора Лоренца и ее использование. // Сложные системы. 2021. № 1 (38). С. 56-63.

5. О FreeCAD. [Электронный ресурс]. URL: https://wiki.freecadweb.org/About_FreeCAD/ru (дата обращения 01.10.2021).

6. Семченко Р.В., Еровлев П.А., Лучанинов Д.В. Обзор программ для создания 3D моделей. // Постулат. 2018. № 5-1 (31). С. 46.
7. Ткаченко О.П. Вычислительное моделирование трубопровода с изломом профиля в пакете прикладных программ FreeCAD. // Современные наукоемкие технологии. 2020. № 11-1. С. 73-78.
8. Федчун Д.О., Глухой Р.Е. Генеративные методы создания малых архитектурных форм. // Архитектура и дизайн: история, теория, инновации. 2018. № 3. С. 321-327.
9. Чухраев И. В., Ильичев В. Ю. Распознавание характерных объектов на изображении с использованием технологий компьютерного зрения. // E-Scio. 2021. № 8 (59). С. 122-131.
10. Astanaliev E.T.O., Baratov D. Raster and vector formats of electronic document of technical documentation. // Universum: технические науки. 2021. № 7-3 (88). С. 35-37.
11. Ilichev V., Nasonov D., Raevsky V., Volkhonskaya A. Methodology of error measurements reduction while determining integrated errors of satellite nodes of planetary gears. // В сборнике: Proceedings of the 52th International JVE Conference in St. Petersburg. 2021. С. 90-94.
12. Yan J., Li J., Lou Y., Li Z. NC machine interference and collision checking system based on OpenCascade. // В сборнике: Proceedings - 2010 International Conference on Digital Manufacturing and Automation, ICDMA 2010. 2010 International Conference on Digital Manufacturing and Automation, ICDMA 2010. С. 976-979.