Отчет по лабораторной работе №5

Построение графиков

Легиньких Галина Андреевна

Содержание

# 1 Цель работы

Основная цель работы — освоить синтаксис языка Julia для построения графиков.

# 2 Задание

1. Используя Jupyter Lab, повторите примеры из раздела 5.2. При этом дополните графики обозначениями осей координат, легендой с названиями траекторий, названиями графиков и т.п.
2. Выполните задания для самостоятельной работы (раздел 5.4).

# 3 Выполнение лабораторной работы

**1.** Для начала я повторила примеры и дробавила, где это было необходимо, обозначения осей координат, легенду с названиями траекторий, названия графиков и т.п. Больше я это нигде прописывать не буду.

Julia поддерживает несколько пакетов для работы с графиками. Использование того или иного пакета зависит от целей, преследуемых пользователем при построении. Стандартным для Julia является пакет Plots.jl.

**2.** Далее рассмотрела построение графика функции f(x) = (3x2 + 6x - 9)exp(-0,3x) разными способами. Фактически для построения графика функции требуется иметь массив соответствующих значений x и y.

(рис. 1) (рис. 2) (рис. 3)

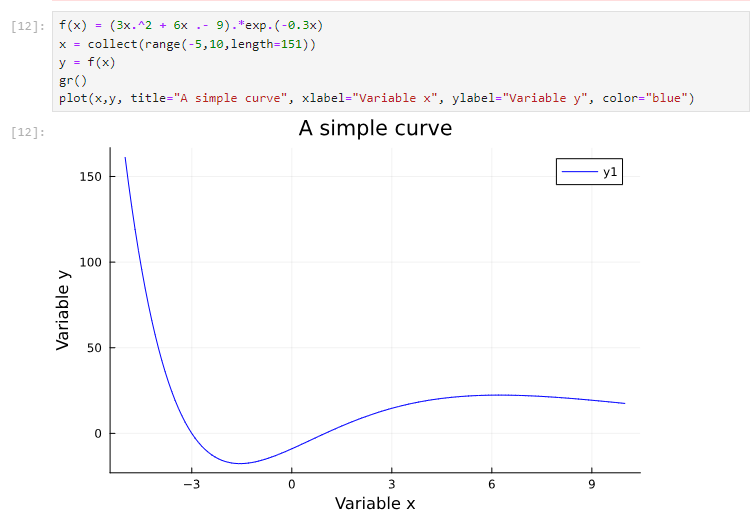


Рис. 1: Способ 1

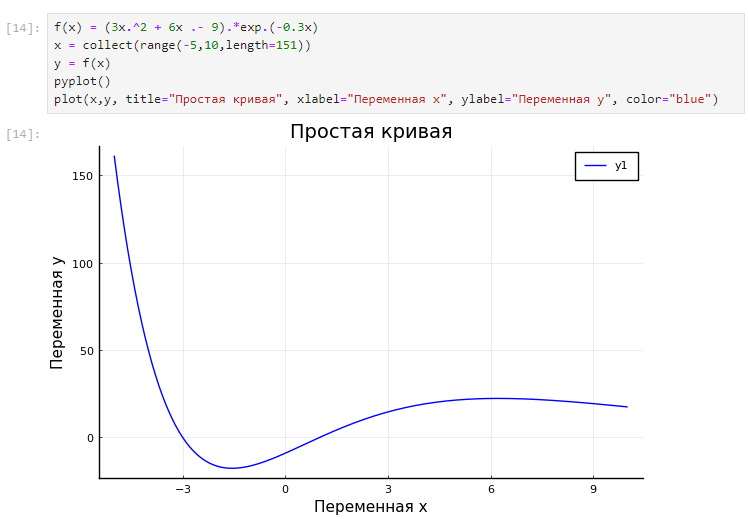


Рис. 2: Способ 2

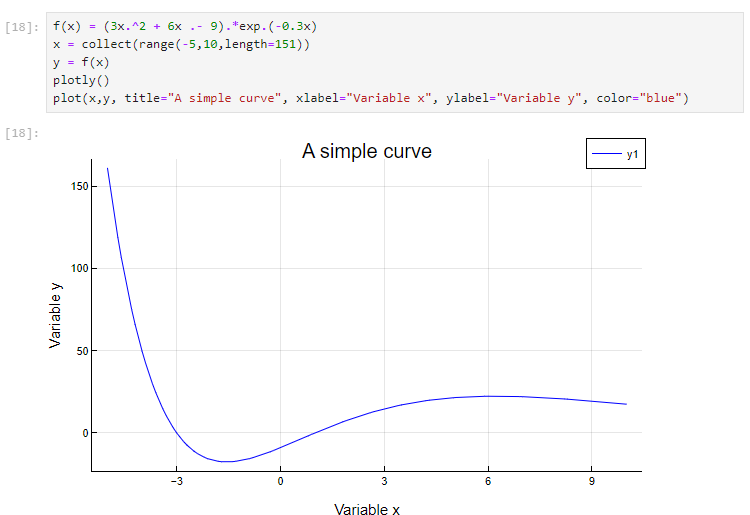


Рис. 3: Способ 3

**3.** Далее на примере графика функции sin(x) и графика разложения этой функции в ряд Тейлора рассмотрела дополнительные возможности пакетов для работы с графикой. В примерах рассматривается поэтапное улучшение графика. Добавлю скриншот конечного результата. (рис. 4) (рис. 5)



Рис. 4: Код для графика исходной функции и её разложения в ряд Тейлора с опциями

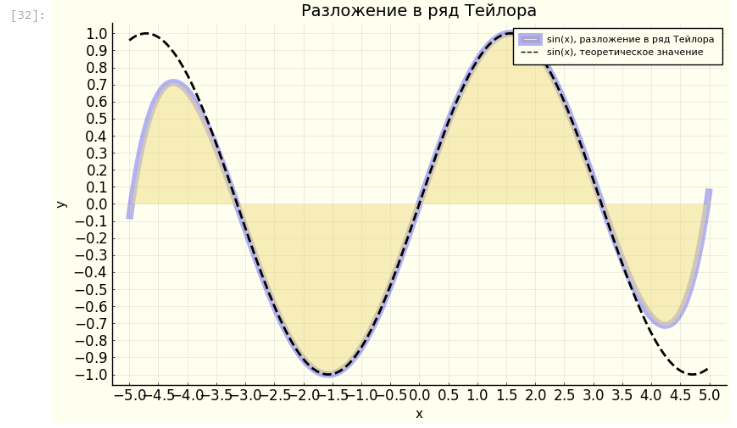


Рис. 5: Графики исходной функции и её разложения в ряд Тейлора с опциями

**4.** Как и построении обычного графика для точечного графика необходимо задать массив значений x, посчитать или задать значения 𝑦, задать опции построения графика. Примеры 2-мерного и 3-мерного графиков. (рис. 6) (рис. 7)

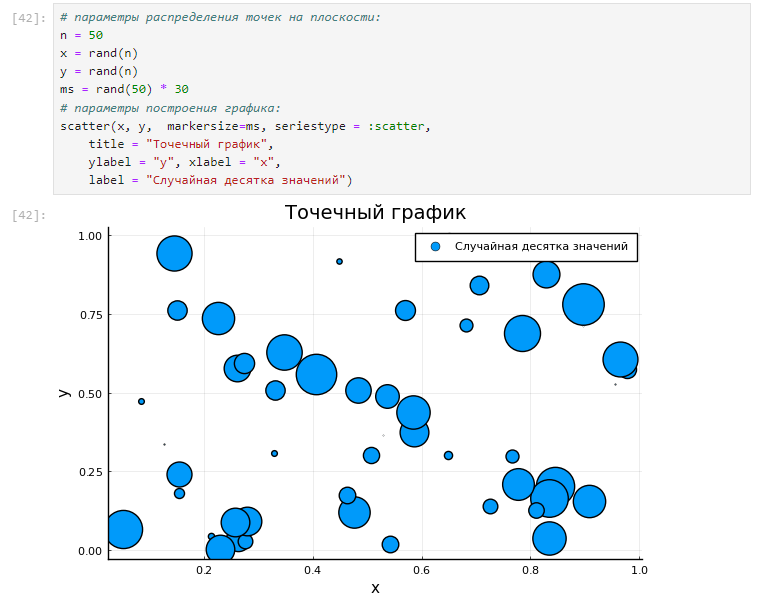


Рис. 6: График пятидесяти случайных значений на плоскости с различными опциями отображения

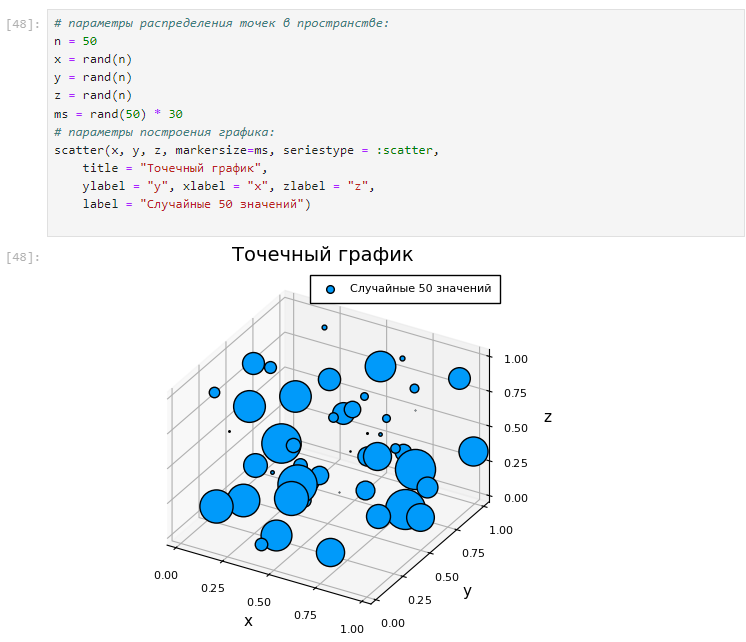


Рис. 7: График пятидесяти случайных значений в пространстве с различными опциями отображения

**5.** Аппроксимация — научный метод, состоящий в замене объектов их более простыми аналогами, сходными по своим свойствам. (рис. 8)

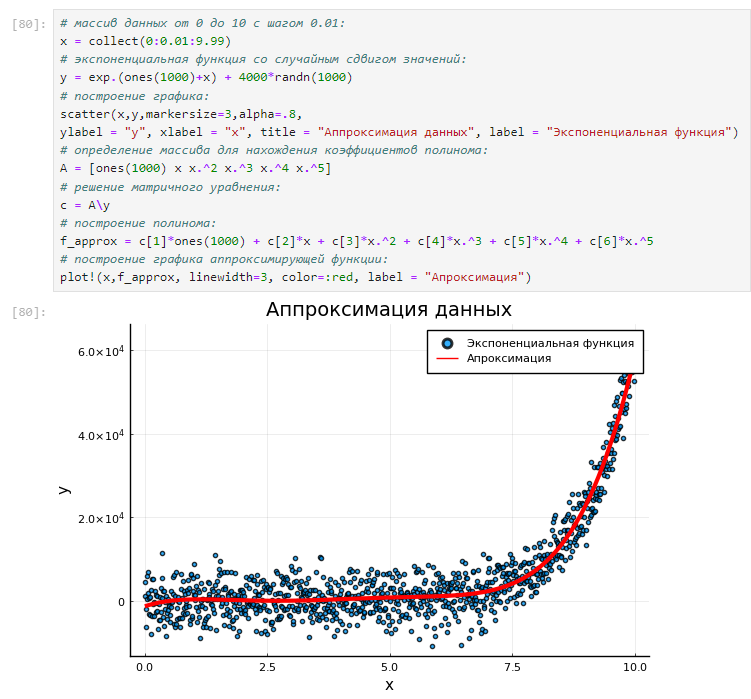


Рис. 8: Пример аппроксимации исходной функции полиномом 5-й степени

**6.** Иногда требуется на один график вывести несколько траекторий с существенными отличиями в значениях по оси ординат. (рис. 9)

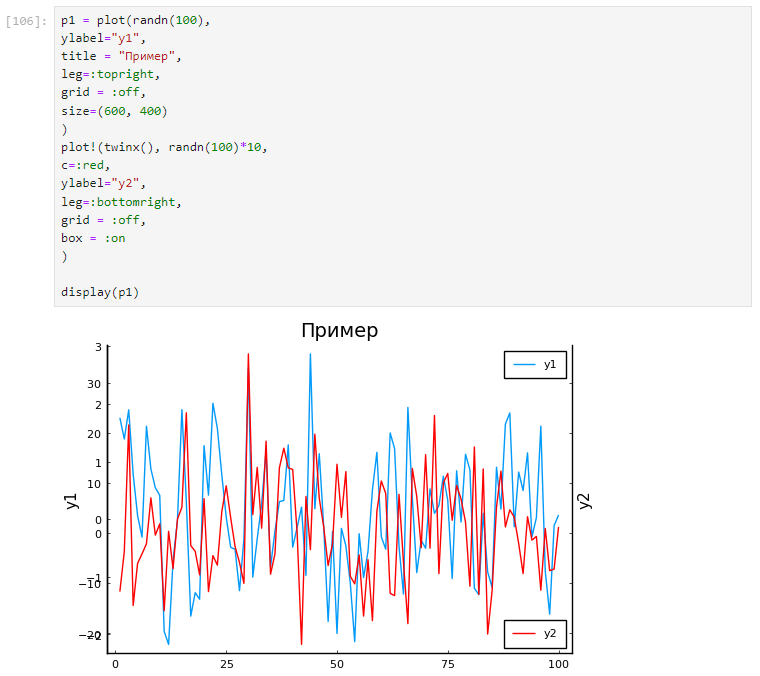


Рис. 9: Пример двух траекторий на одном графике с двумя осями ординат

**7.** Повторила пример построения графика функции в полярных координатах. (рис. 10)

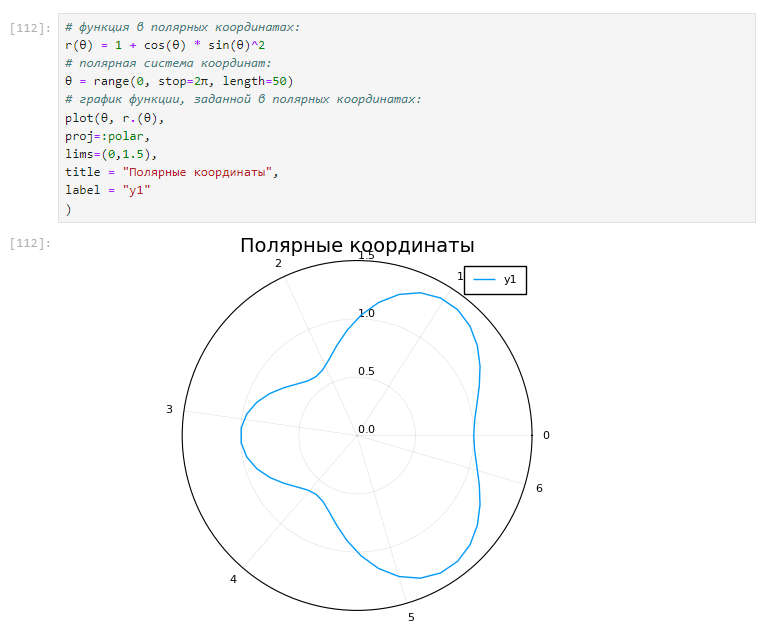


Рис. 10: График функции, заданной в полярных координатах

**8.** Повторила пример построения графика параметрически заданной кривой на плоскости. (рис. 11)

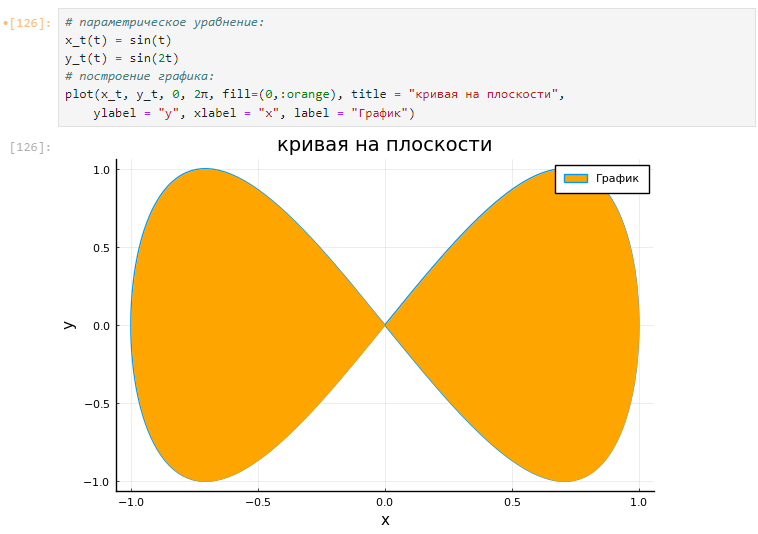


Рис. 11: Параметрический график кривой на плоскости

**9.** Приведём пример построения графика параметрически заданной кривой в пространстве. (рис. 12)

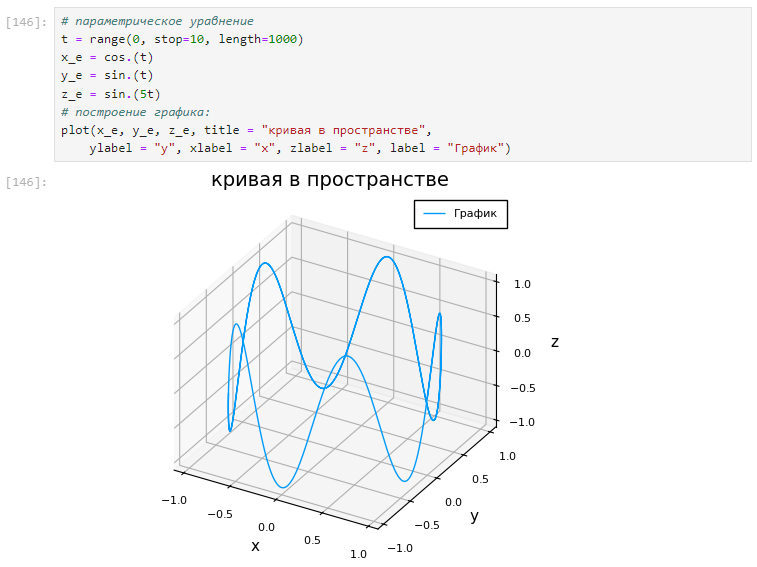


Рис. 12: Параметрический график кривой в пространстве

**10.** Перешла к графикам поверхности. (рис. 13) (рис. 14) (рис. 15) (рис. 16)

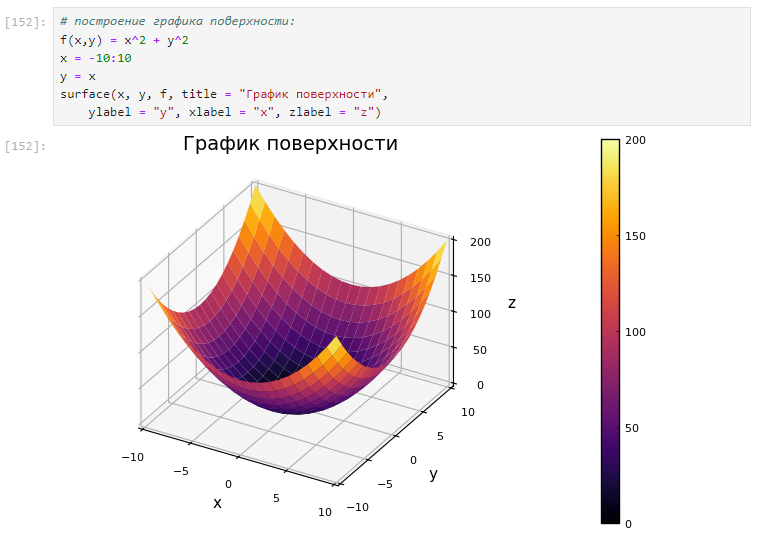


Рис. 13: График поверхности (использована функция surface())

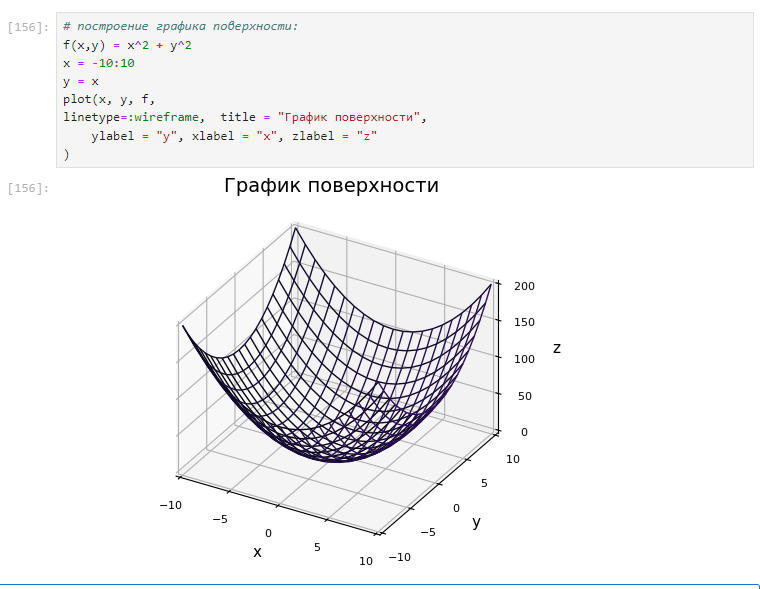


Рис. 14: График поверхности (использована функция plot())

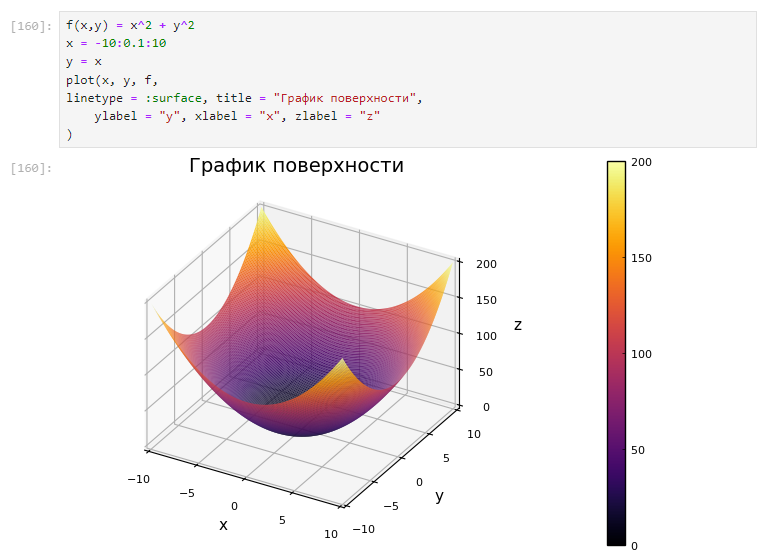


Рис. 15: Сглаженный график поверхности

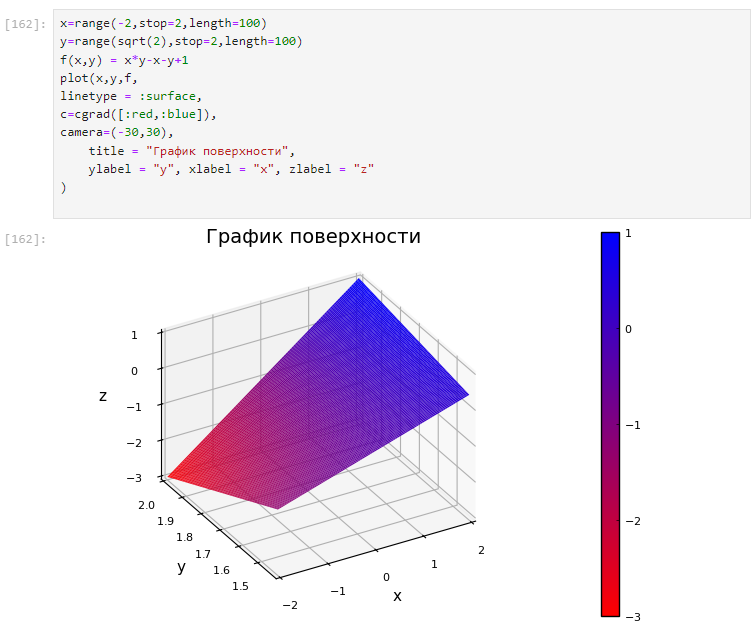


Рис. 16: График поверхности с изменённым углом зрения

**11.** Линией уровня некоторой функции от двух переменных называется множество точек на координатной плоскости, в которых функция принимает одинаковые значения. Линий уровня бесконечно много, и через каждую точку области определения можно провести линию уровня. Рассмотрела поверхность, заданную функцией(рис. 17)

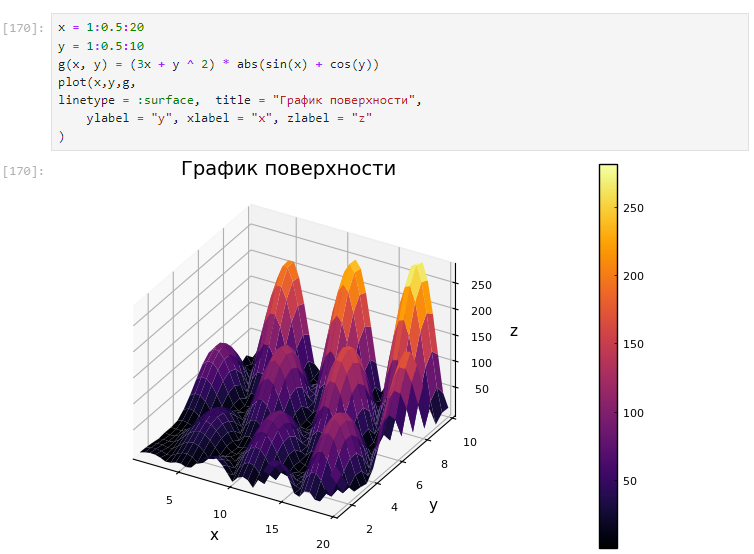


Рис. 17: График поверхности, заданной функцией

Линии уровня можно построить, используя проекцию значений исходной функции на плоскость. Можно дополнительно добавить заливку цветом. (рис. 18)

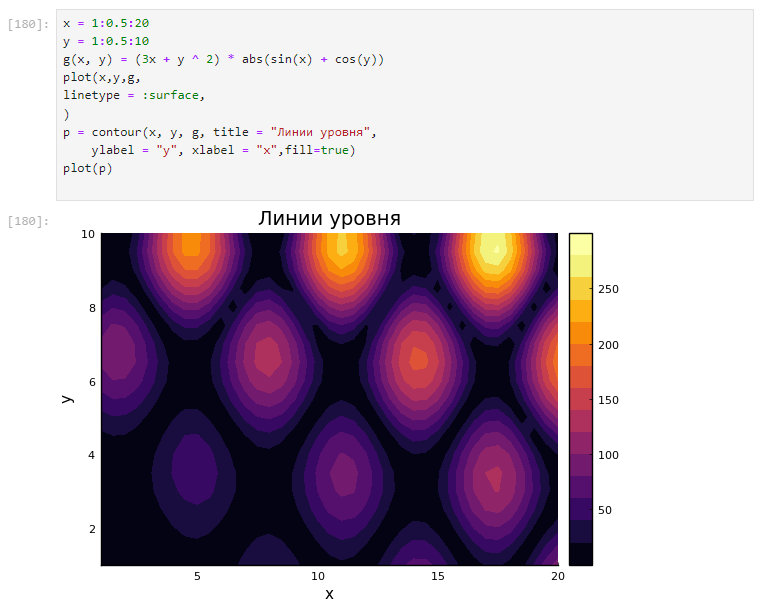


Рис. 18: Линии уровня с заполнением

**12.** Если каждой точке некоторой области пространства поставлен в соответствие вектор с началом в данной точке, то говорят, что в этой области задано векторное поле. (рис. 19) (рис. 20)

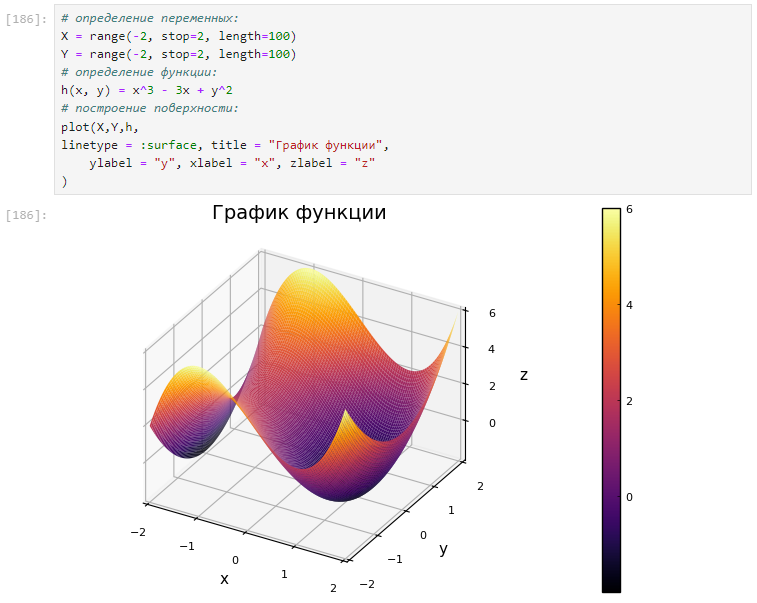


Рис. 19: График функции

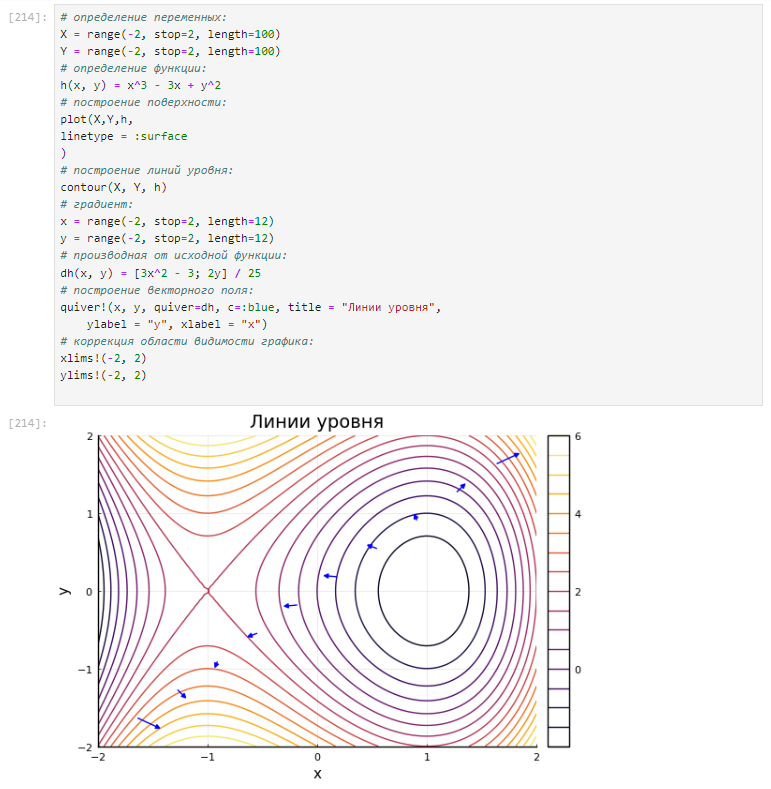


Рис. 20: Векторное поле функции

**13.** В Julia рекомендуется использовать gif-анимацию в pyplot(). (рис. 21) (рис. 22)



Рис. 21: Код для статичного графика и анимации

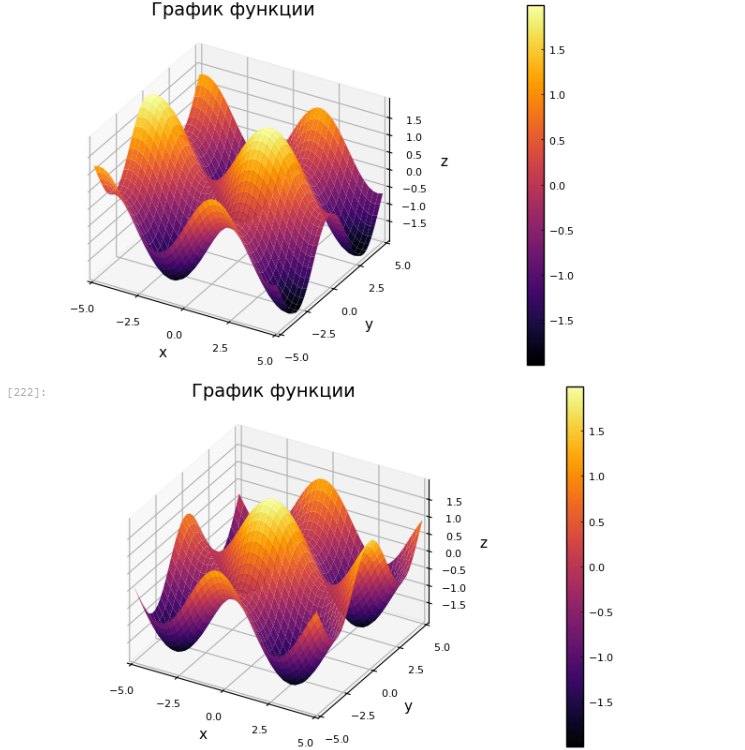


Рис. 22: Анимированный график поверхности

**14.** Гипоциклоида — плоская кривая, образуемая точкой окружности, катящейся по внутренней стороне другой окружности без скольжения. Дано поэтапное создания графика. Приложу конечный вариант. (рис. 23) (рис. 24)



Рис. 23: Код малая окружность гипоциклоиды с добавлением радиуса

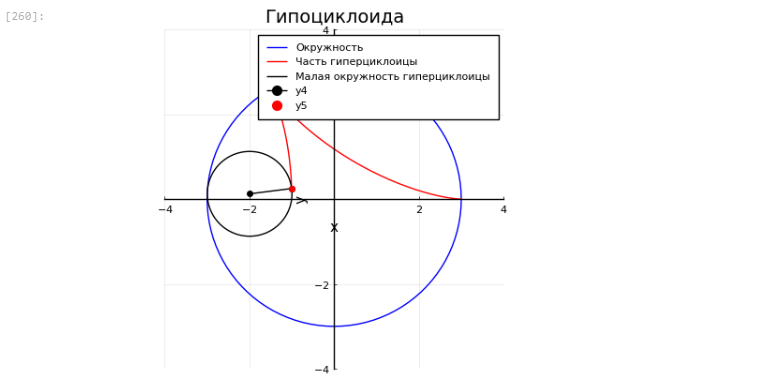


Рис. 24: Малая окружность гипоциклоиды с добавлением радиуса

В конце сделаем анимацию получившегося изображения. (рис. 25)

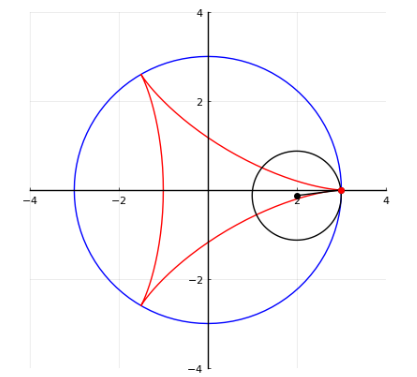


Рис. 25: Анимация гипоциклоиды

**15.** В исследованиях часто требуется изобразить графики погрешностей измерения. Подключила пакет Statistics. (рис. 26) (рис. 27) (рис. 28)

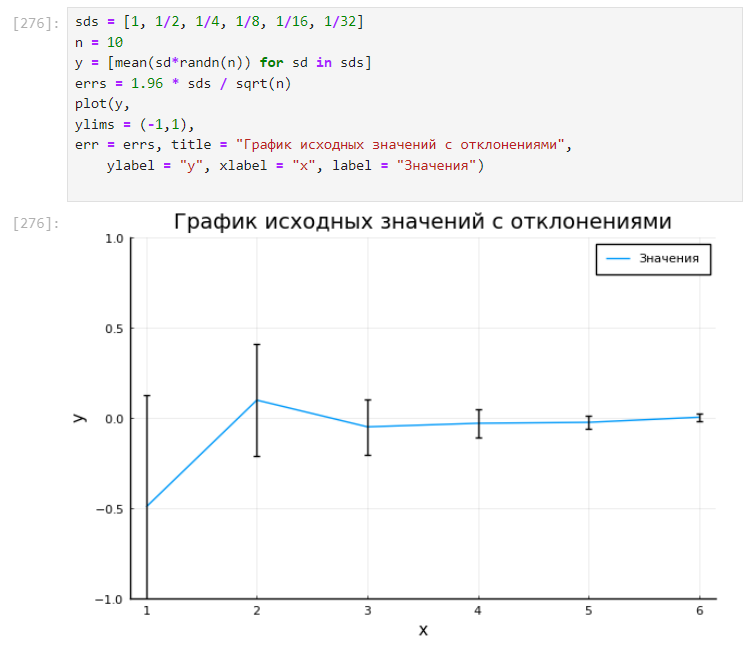


Рис. 26: График исходных значений с отклонениями

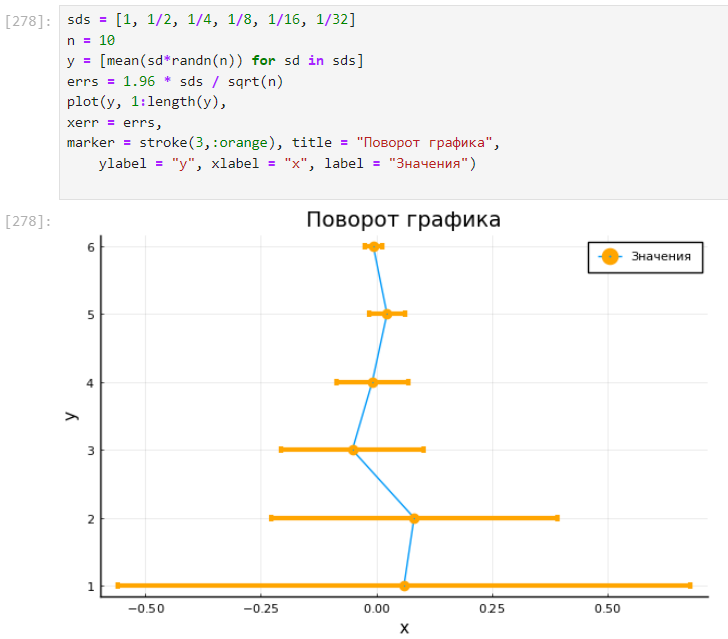


Рис. 27: Поворот графика

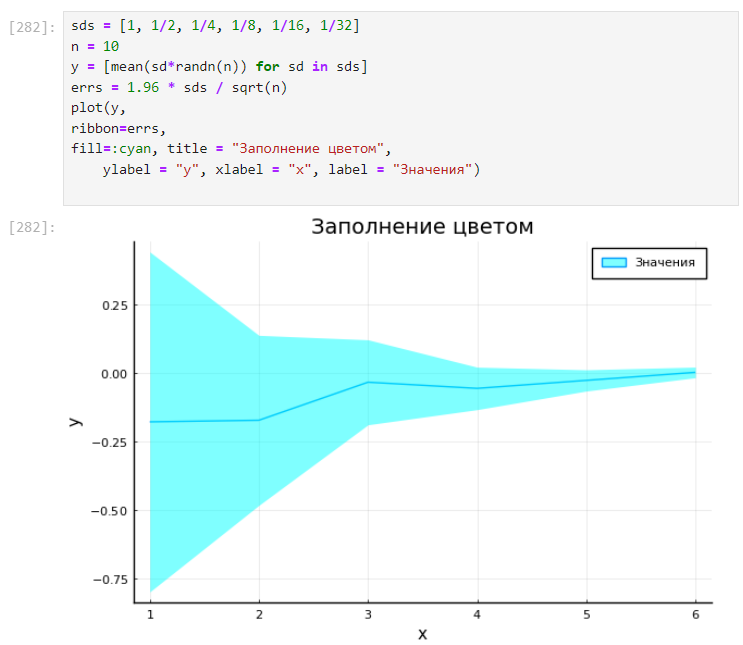


Рис. 28: Заполнение цветом

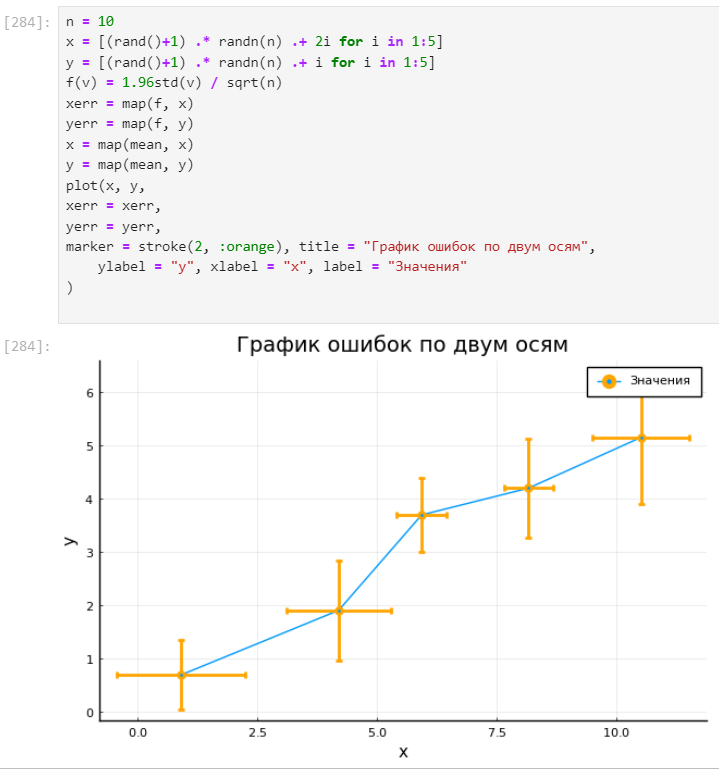


Рис. 29: График ошибок по двум осям

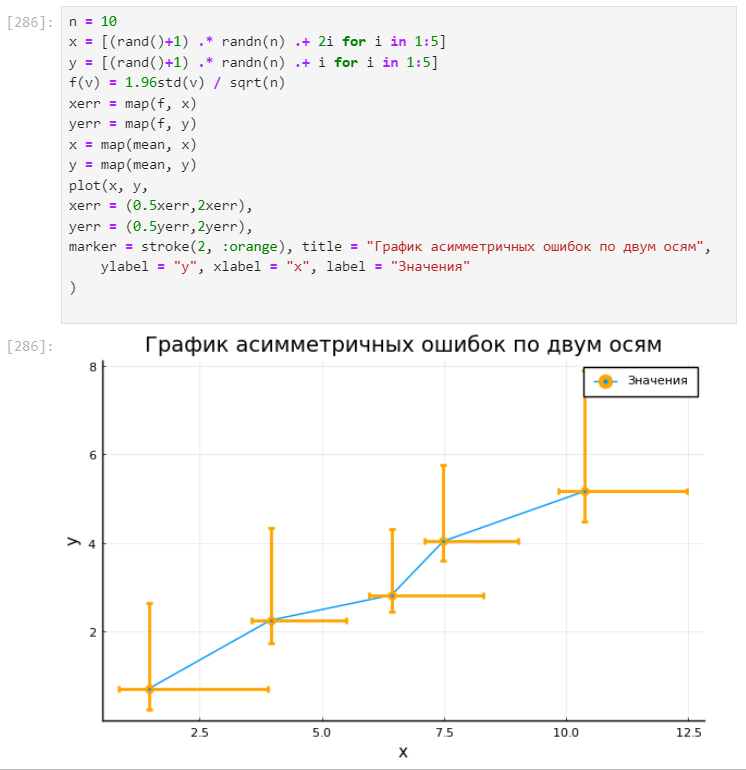


Рис. 30: График асимметричных ошибок по двум осям

**16.** Использование пакета Distributions. (рис. 31) (рис. 32) (рис. 33)

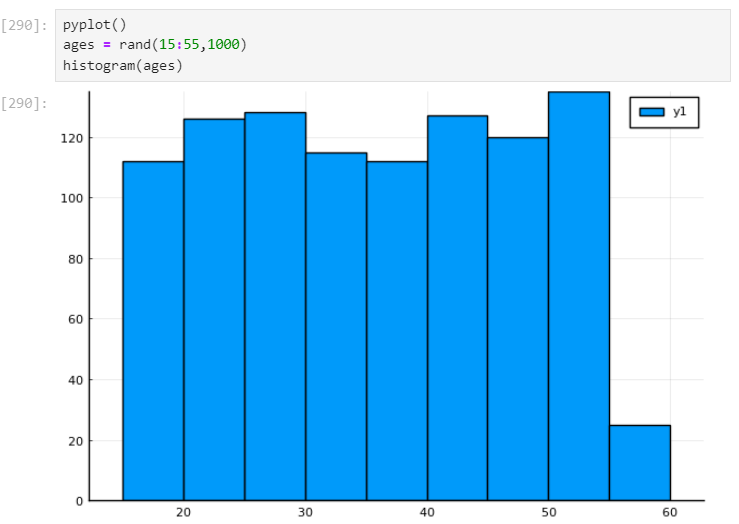


Рис. 31: Гистограмма, построенная по массиву случайных чисел

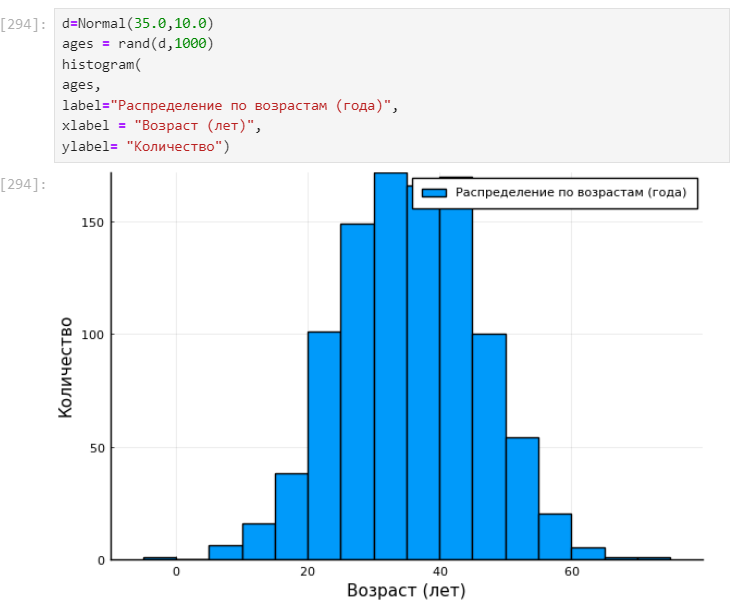


Рис. 32: Гистограмма нормального распределения

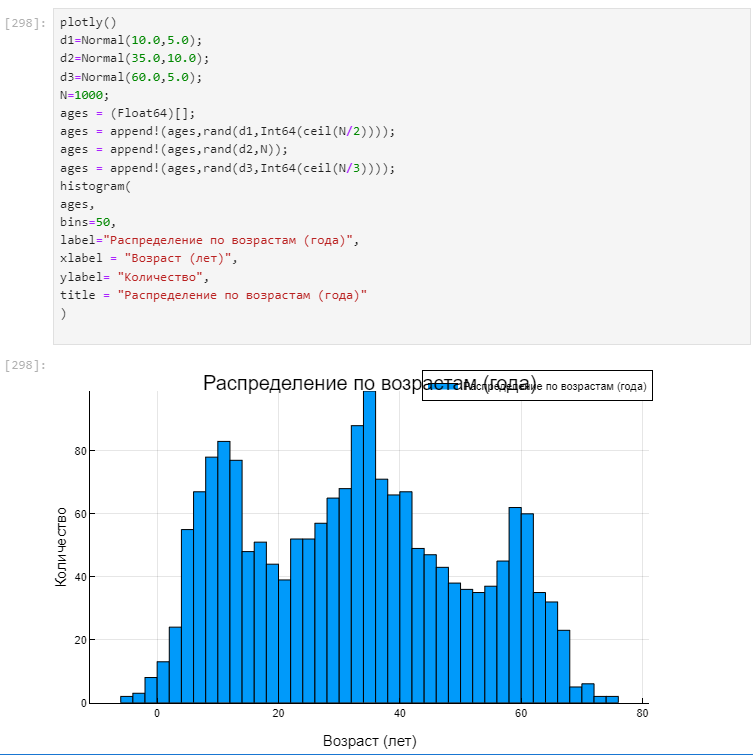


Рис. 33: Гистограмма распределения людей по возрастам

**17.** Определим макет расположения графиков. Команда layout принимает кортеж layout = (N, M), который строит сетку графиков NxM. Например, если задать layout = (4,1) на графике четыре серии, то получим четыре ряда графиков. (рис. 34) (рис. 35) (рис. 36) (рис. 37) (рис. 38)

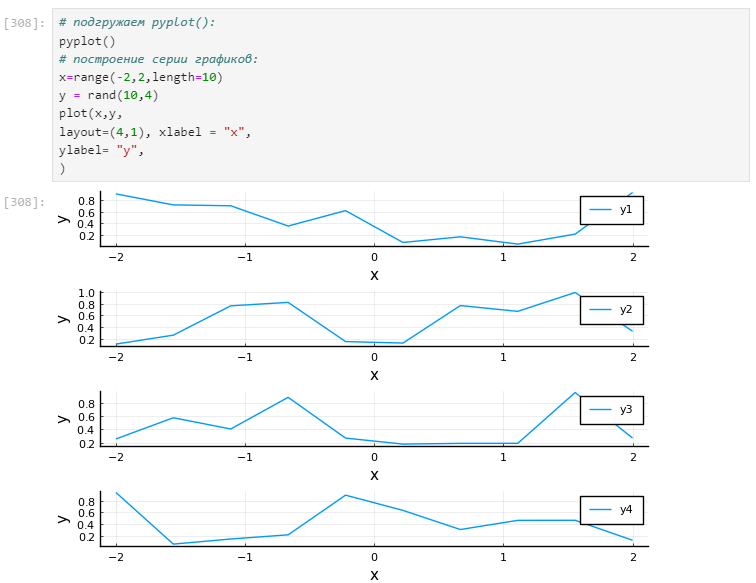


Рис. 34: Серия из 4-х графиков в ряд

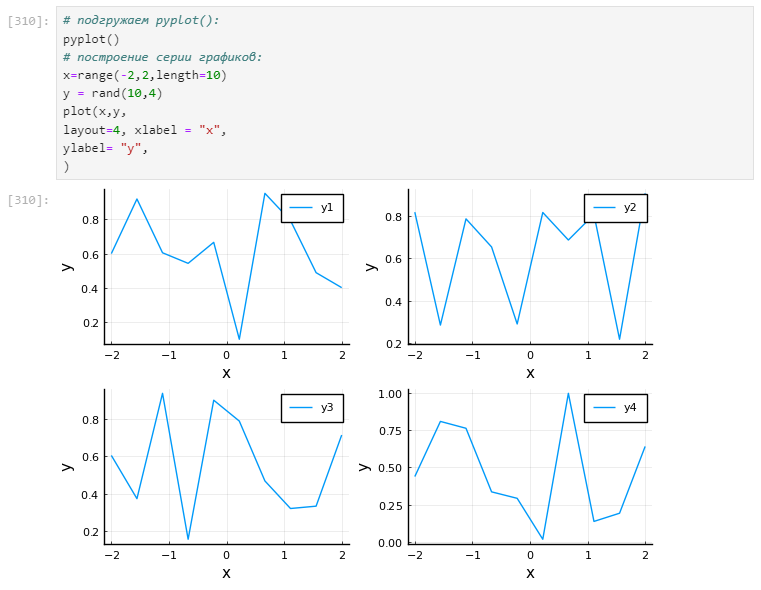


Рис. 35: Серия из 4-х графиков в сетке

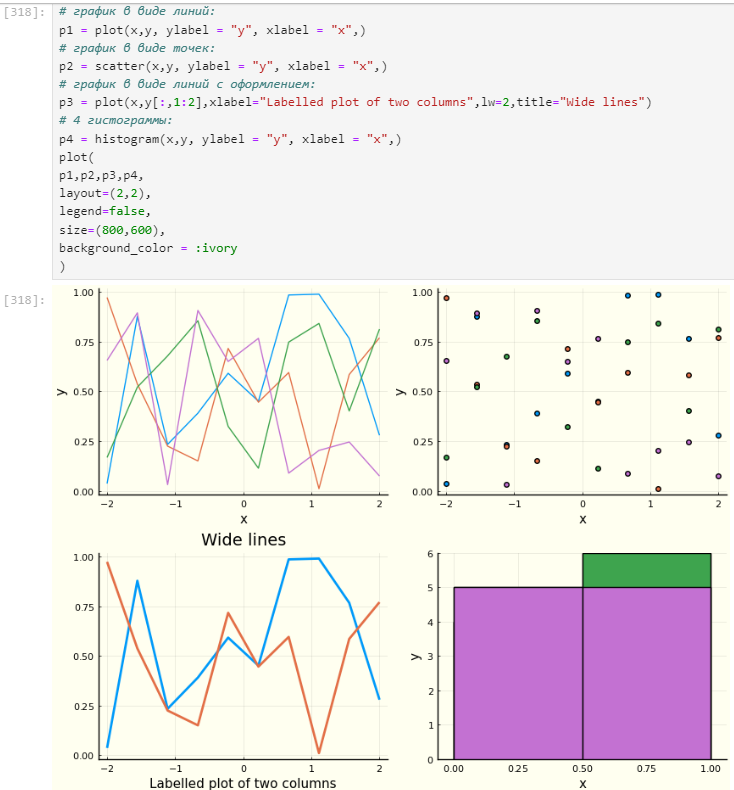


Рис. 36: Объединение нескольких графиков в одной сетке

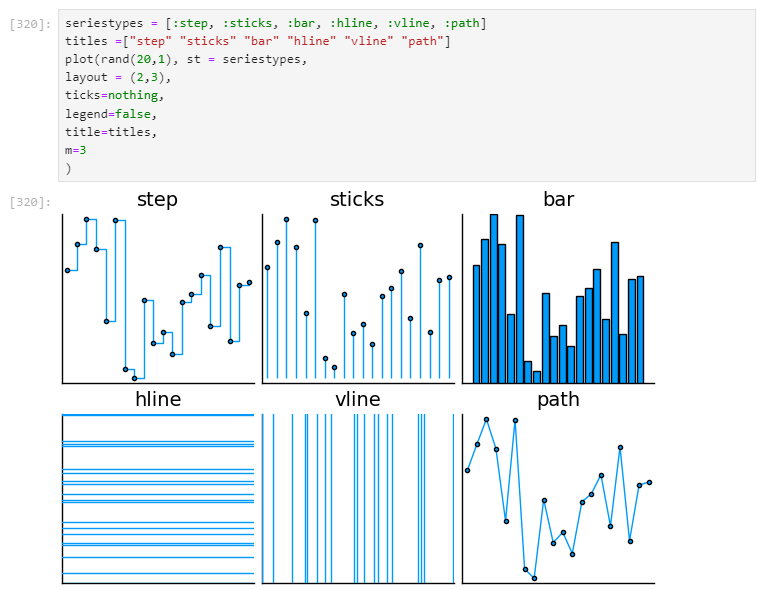


Рис. 37: Разнообразные варианты представления данных

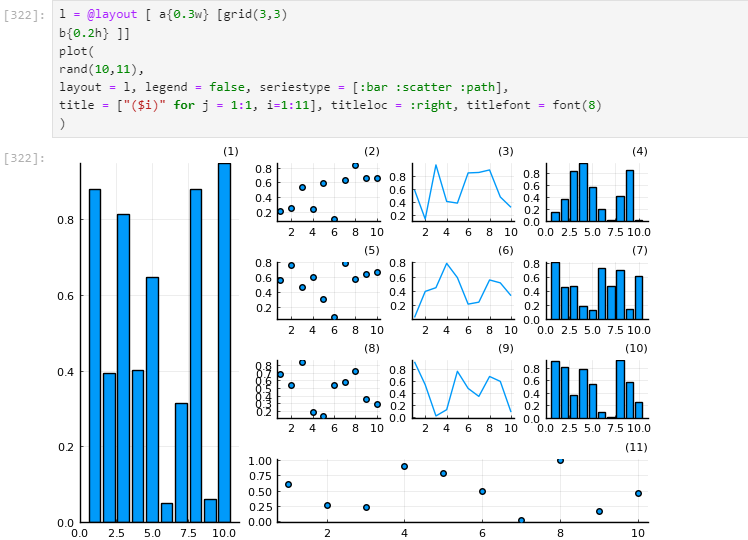


Рис. 38: Демонстрация применения сложного макета для построения графиков

**18.** Перешла к заданиям для самостоятельной работы. Нумерация соответствует.

* Задание 1 (рис. 39)

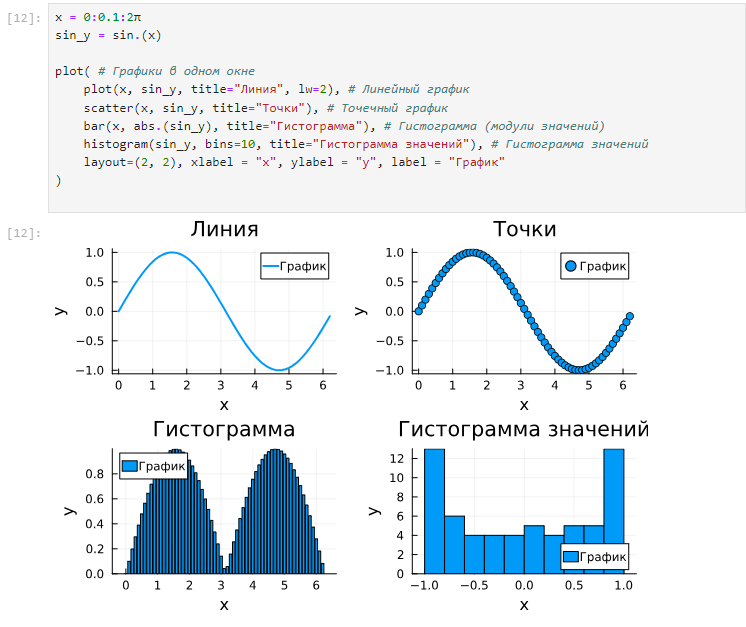


Рис. 39: Задание 1

* Задание 2 (рис. 40)

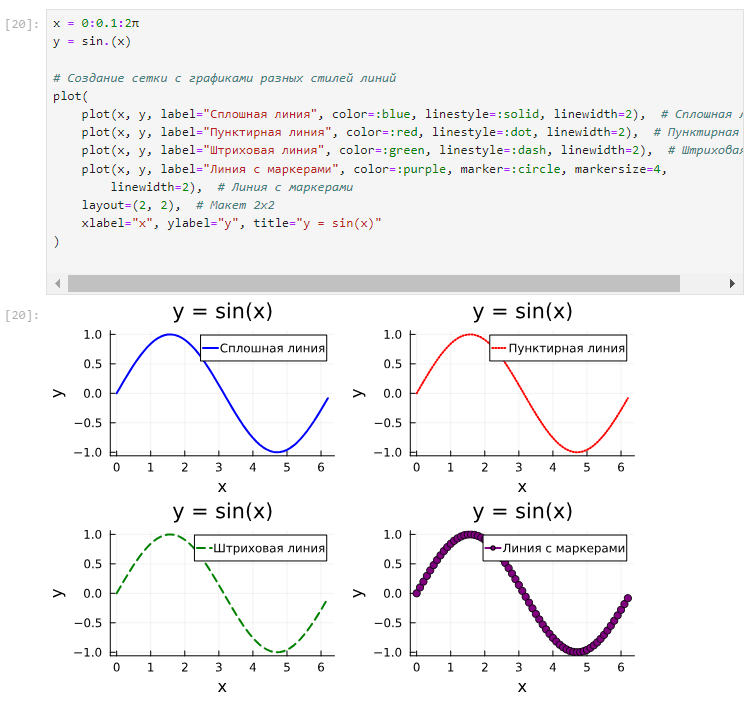


Рис. 40: Задание 2

* Задание 3 (рис. 41)

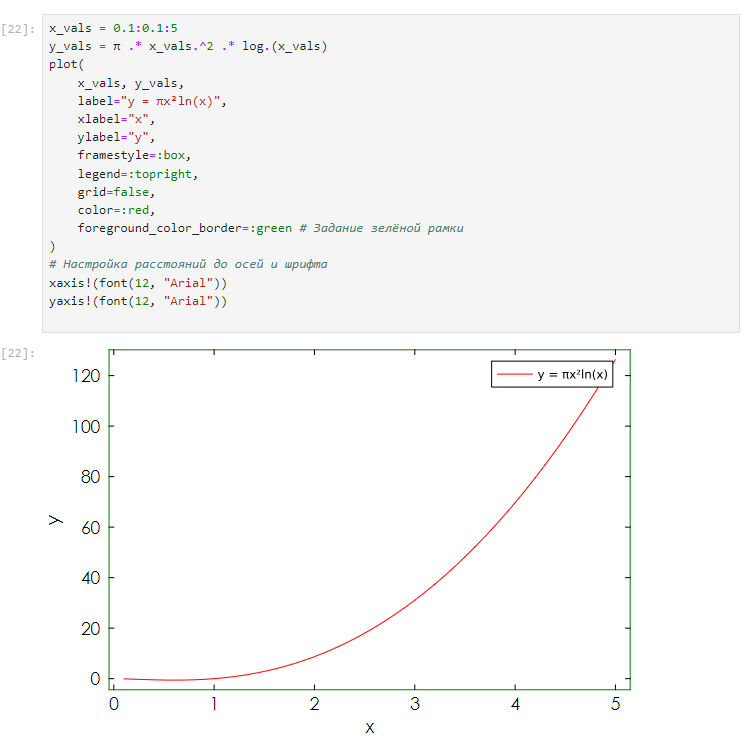


Рис. 41: Задание 3

* Задание 4 (рис. 42)



Рис. 42: Задание 4

* Задание 5 (рис. 43) (рис. 44)

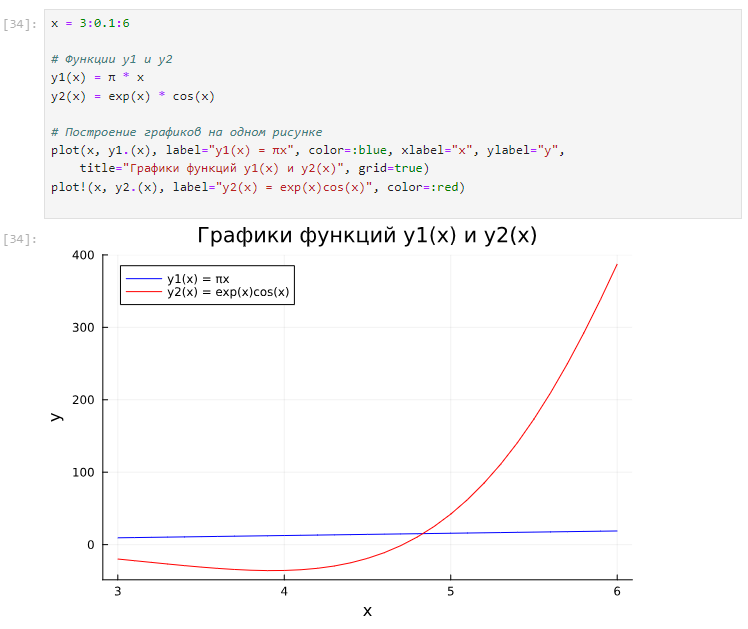


Рис. 43: Задание 5.1

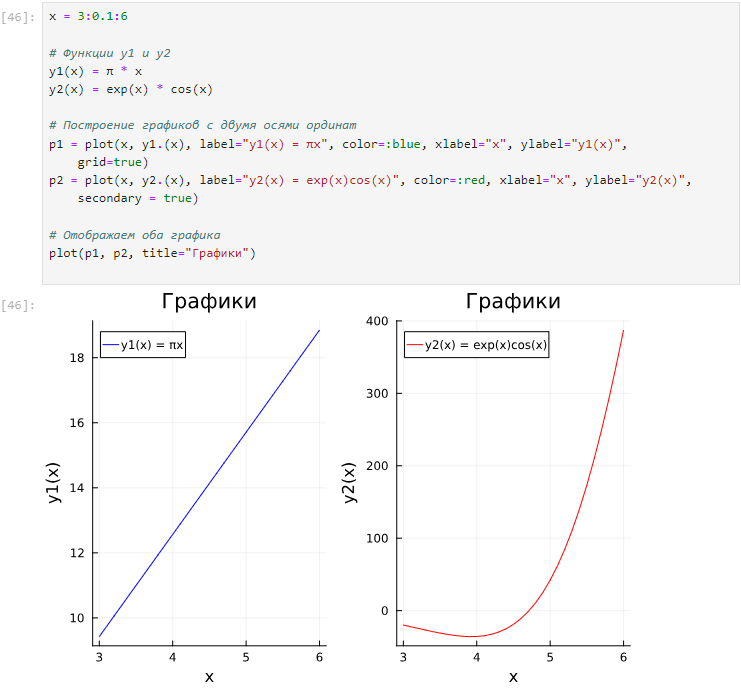


Рис. 44: Задание 5.2

* Задание 6 (рис. 45)

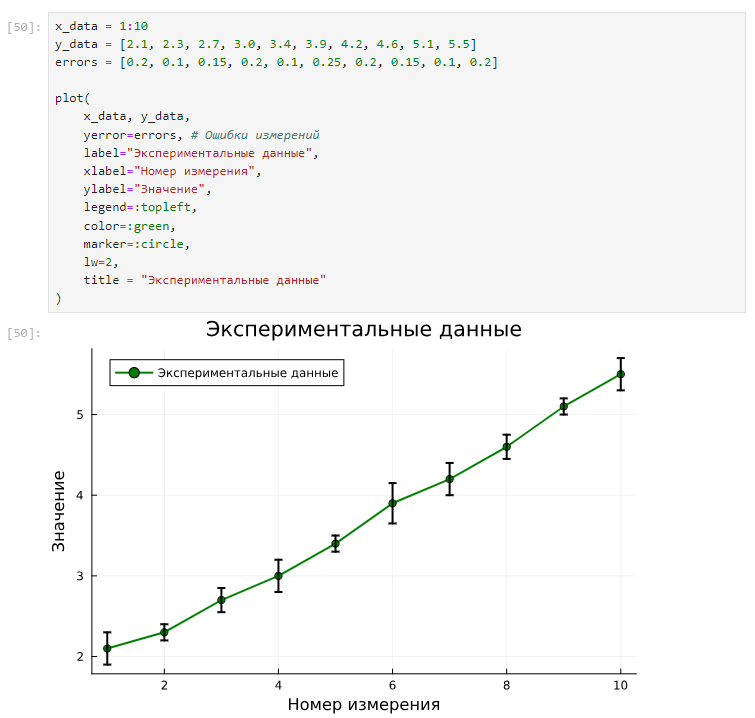


Рис. 45: Задание 6

* Задание 7 (рис. 46)

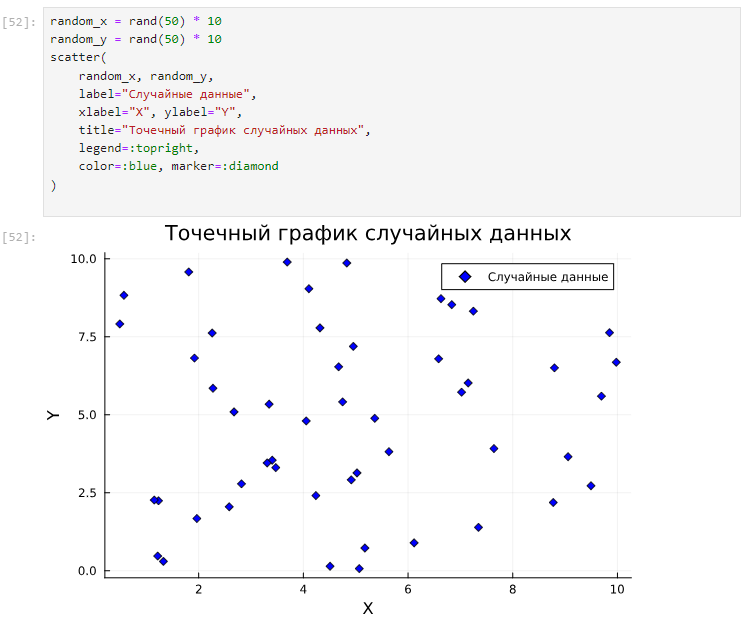


Рис. 46: Задание 7

* Задание 8 (рис. 47)

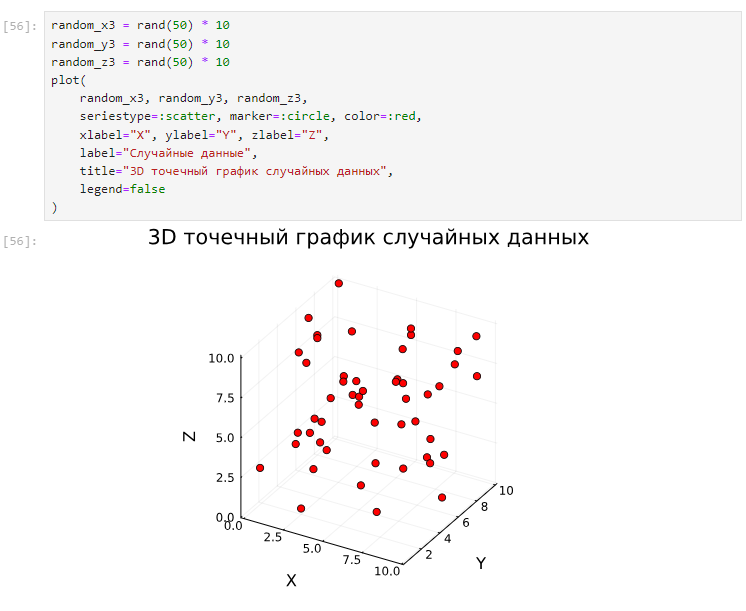


Рис. 47: Задание 8

* Задание 9 (рис. 48)

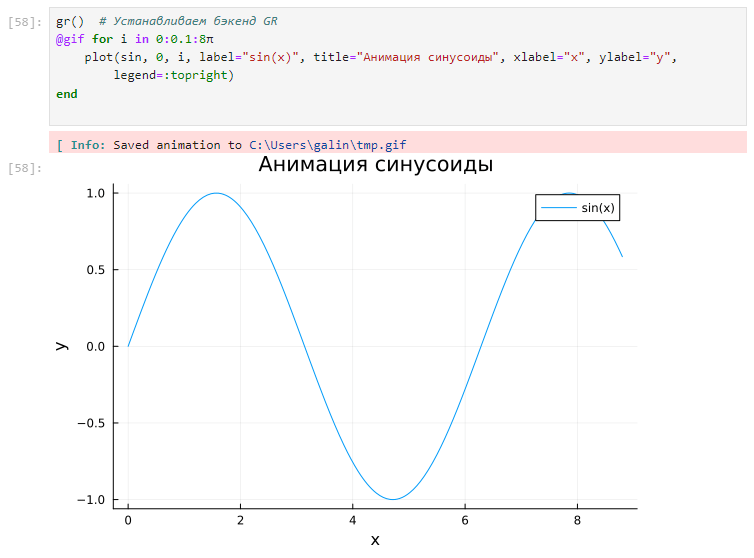


Рис. 48: Задание 9

* Задание 10 (рис. 49)

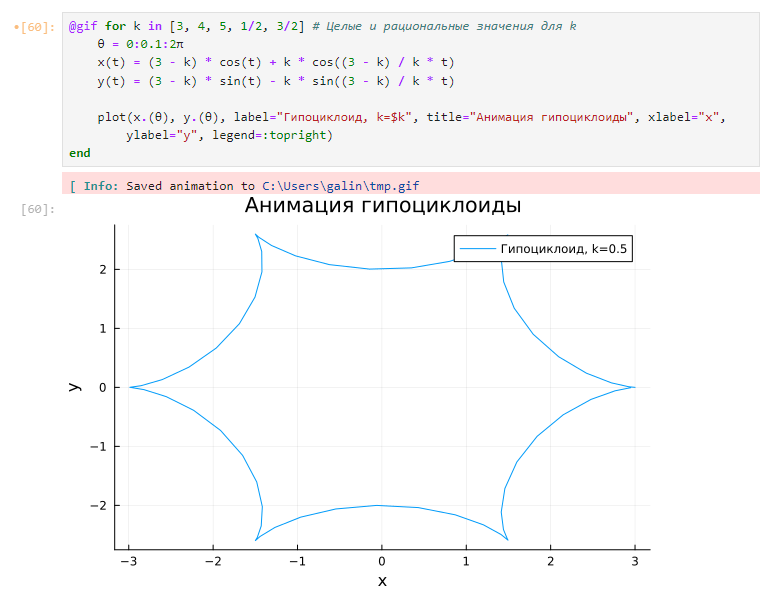


Рис. 49: Задание 10

* Задание 11 (рис. 50)

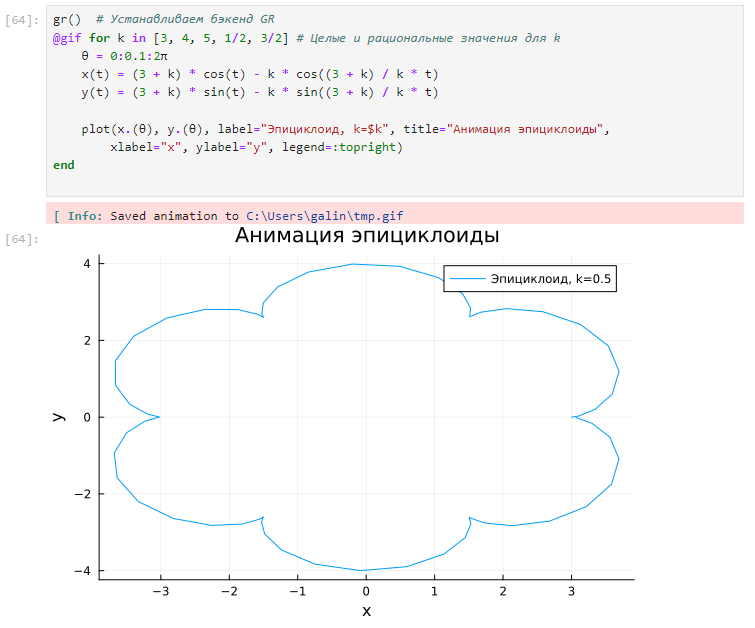


Рис. 50: Задание 11

# 4 Вывод

Освоила синтаксис языка Julia для построения графиков.