

1. Генерируем небольшие примерные данные.
2. Затем сформируем скрипт для gnuplot, чтобы построить график.

Данные сохраним в файл - data.txt:

```
# X Y
1 2
2 4
3 9
4 16
5 25
6 36
7 49
8 64
9 81
10 100
```

Это квадратичная зависимость ($y = x^2$).

Создадим скрипт для gnuplot, который назовем (plot.gp):

```
set title "Квадратичная зависимость"
set xlabel "X"
set ylabel "Y"
set grid
plot "data.txt" using 1:2 with linespoints title "y = x^2"
```

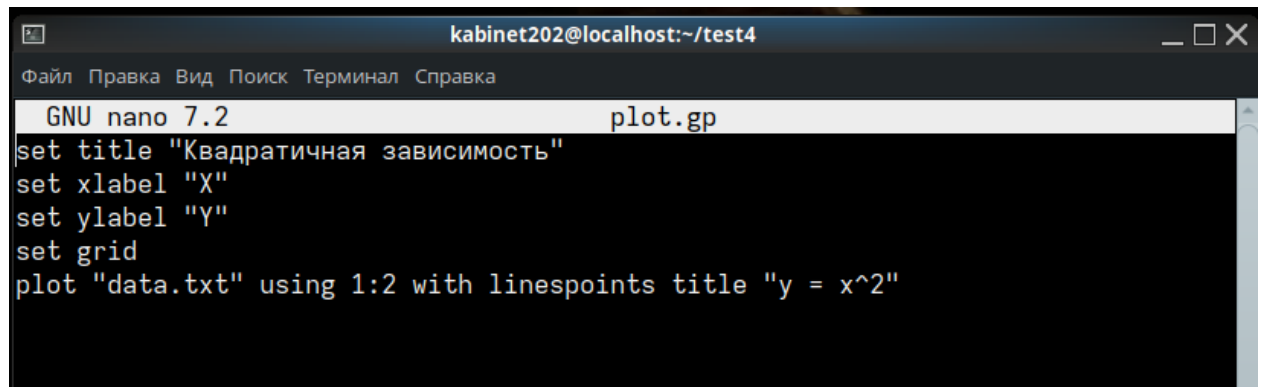
Что нужно сделать:

Сохранить данные в файл data.txt.

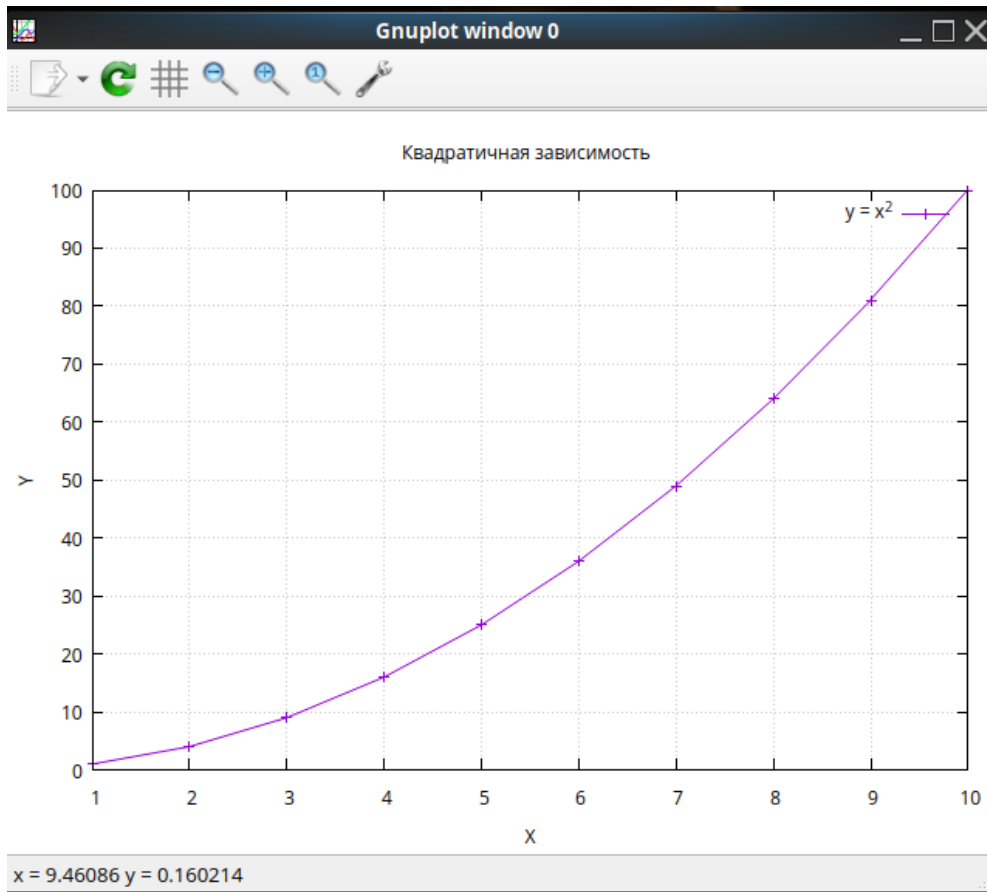
Сохранить скрипт в файл plot.gp.

Запустить в терминале:

```
[kabinet202@localhost test4]$ touch plot.gp
[kabinet202@localhost test4]$ nano plot.gp
[kabinet202@localhost test4]$ gnuplot -persist plot.gp
[kabinet202@localhost test4]$ gnuplot -persist plot.gp
```

A screenshot of a terminal window titled "kabinet202@localhost:~/test4". The window shows the GNU nano 7.2 editor editing the file plot.gp. The editor's menu bar includes "Файл", "Правка", "Вид", "Поиск", "Терминал", and "Справка". The content of the file is as follows:

```
set title "Квадратичная зависимость"
set xlabel "X"
set ylabel "Y"
set grid
plot "data.txt" using 1:2 with linespoints title "y = x^2"
```



Генерация другого типа данных с более красивым оформлением графика (цвета, легенды, стили)

Пример: синусоида + шум

(Типичные экспериментальные данные)

Данные (data.txt):

```
# X Y
0 0.1
0.5 0.48
1.0 0.89
1.5 0.99
```

2.0 0.91
2.5 0.59
3.0 0.04
3.5 -0.53
4.0 -0.88
4.5 -0.98
5.0 -0.83
5.5 -0.58
6.0 -0.1
6.5 0.45
7.0 0.84

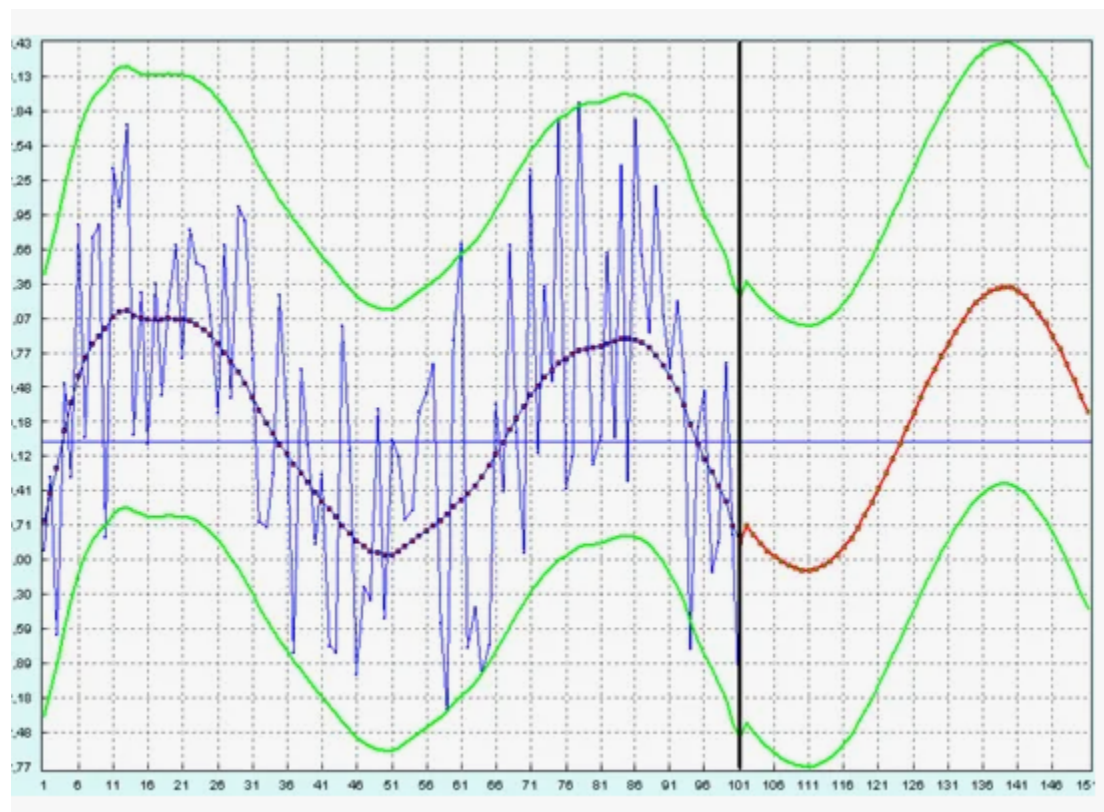
Скрипт для gnuplot (plot.gp):

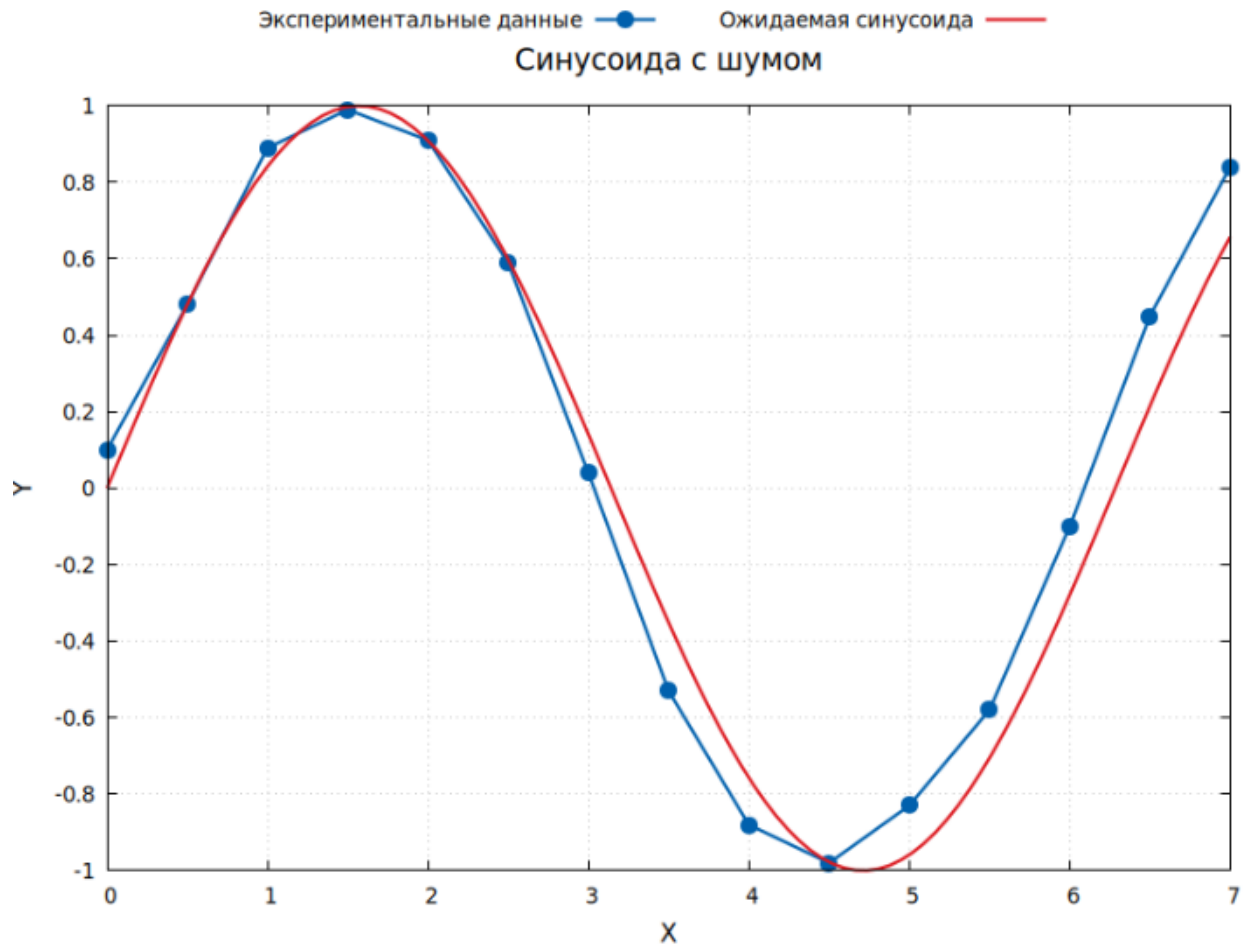
```
set terminal pngcairo size 800,600 enhanced font 'Verdana,10'  
set output 'output.png'  
  
set title "Синусоида с шумом" font ",14"  
set xlabel "X" font ",12"  
set ylabel "Y" font ",12"  
set grid  
set style line 1 lc rgb '#0060ad' lt 1 lw 2 pt 7 ps 1.5 # синий цвет, линии и  
маркеры  
set key outside top center horizontal  
plot "data.txt" using 1:2 with linespoints ls 1 title "Экспериментальные  
данные", \  
sin(x) with lines lw 2 lc rgb '#dd181f' title "Ожидаемая синусоида"
```

Что будет на графике:

- Синие точки с линиями — реальные измерения с шумом.
- Красная кривая — эталонная функция $\sin(x)$.
- Подписи, сетка, красивая легенда сверху.

👉 Запуск: *gnuplot plot.gp*





3. Генерация данных (например, экспоненту + шум)? И подготовка сразу несколько графиков в одном окне 🚀

1. Реалистичные данные: экспонента + шум

Файл данных exp_data.txt:

```
# X Y
0 1.05
0.5 1.65
1.0 2.82
1.5 4.32
2.0 7.43
2.5 11.59
```

3.0 20.16
3.5 32.01
4.0 53.97
4.5 87.12
5.0 146.80

(модель примерно $(y = e^{\{0.9x\}}) + \text{небольшой шум}$)

2. Скрипт для нескольких графиков в одном окне

Файл multi_plot.gp:

gnuplot

Copy code

```
set terminal pngcairo size 1200,800 enhanced font 'Verdana,10'  
set output 'multi_output.png'
```

```
set multiplot layout 2,1 title "Сравнение данных" font ",16"
```

```
# Первый график: Синусоида
```

```
set title "Синусоида с шумом"
```

```
set xlabel "X"
```

```
set ylabel "Y"
```

```
set grid
```

```
set key outside top center horizontal
```

```
set style line 1 lc rgb '#0060ad' lt 1 lw 2 pt 7 ps 1.5
```

```
set style line 2 lc rgb '#dd181f' lt 1 lw 2
```

```
plot "data.txt" using 1:2 with linespoints ls 1 title "Экспериментальные  
данные", \
```

```
sin(x) with lines ls 2 title "Ожидаемая синусоида"
```

```
# Второй график: Экспонента
```

```
set title "Экспонента с шумом"
```

```
set xlabel "X"
set ylabel "Y"
set grid
set style line 3 lc rgb '#228B22' lt 1 lw 2 pt 5 ps 1.5
set style line 4 lc rgb '#8B0000' lt 1 lw 2
plot "exp_data.txt" using 1:2 with linespoints ls 3 title "Экспериментальные
данные", \
    exp(0.9*x) with lines ls 4 title "Модель exp(0.9x)"
```

unset multiplot

Что произойдет:

- Будет одно окно (или один файл multi_output.png).
- В нём два графика:
 - верхний — синусоида с шумом,
 - нижний — экспоненциальный рост с шумом.
- Оба красиво оформлены, с разными цветами, подписями, легендами.

Запуск: `gnuplot multi_plot.gp`

Разберем каждую строчку скрипта multi_plot.gp.

```
set terminal pngcairo size 1200,800 enhanced font 'Verdana,10'
```



Задаем тип вывода:

- `pngcairo` — красивый вывод в PNG формате.
- `size 1200,800` — размер картинки (ширина x высота в пикселях).
- `enhanced` — разрешает форматирование текста (например, индексы, надстрочные знаки).
- `font 'Verdana,10'` — шрифт Verdana размером 10.

```
set output 'multi_output.png'
```

▶ Указываем имя выходного файла для сохранения результата.

set multiplot layout 2,1 title "Сравнение данных" font ",16"

▶ Активируем режим multiplot (много графиков в одном окне).

- layout 2,1 — два графика по вертикали (2 строки, 1 колонка).
- title — заголовок для всей композиции.
- font ",16" — размер шрифта заголовка.

Первый график: синусоида

set title "Синусоида с шумом"

▶ Заголовок первого графика.

set xlabel "X"

set ylabel "Y"

▶ Подписи осей X и Y.

set grid

▶ Включаем сетку.

set key outside top center horizontal

▶ Легенда (обозначение линий) будет снаружи, сверху, по центру, горизонтальная.

set style line 1 lc rgb '#0060ad' lt 1 lw 2 pt 7 ps 1.5

▶ Определяем стиль линии №1:

- lc rgb '#0060ad' — цвет синий;

- lt 1 — тип линии обычный (сплошная);
- lw 2 — толщина линии 2;
- pt 7 — тип маркера (например, кружок);
- ps 1.5 — размер маркера.

set style line 2 lc rgb '#dd181f' lt 1 lw 2

▶ Определяем стиль линии №2:

- красный цвет,
- сплошная линия,
- толщина 2,
- без маркеров.

*plot "data.txt" using 1:2 with linespoints ls 1 title "Экспериментальные данные", *
sin(x) with lines ls 2 title "Ожидаемая синусоида"

▶ Строим два графика на одном поле:

- данные из data.txt, колонки 1 и 2, стиль №1 (линии + точки), подпись "Экспериментальные данные",
- функцию $\sin(x)$, стиль №2 (линия), подпись "Ожидаемая синусоида".

Второй график: экспонента

set title "Экспонента с шумом"

▶ Новый заголовок для второго графика.

set xlabel "X"
set ylabel "Y"
set grid

▶ Снова подписи осей и сетка (нужно заново для каждого подграфика в `multiplot`).

set style line 3 lc rgb '#228B22' lt 1 lw 2 pt 5 ps 1.5

▶ Стилль линии №3:

- темно-зеленый цвет,
- сплошная линия,
- толщина 2,
- маркеры типа 5 (например, квадратик).

set style line 4 lc rgb '#8B0000' lt 1 lw 2

▶ Стилль линии №4:

- темно-красный цвет,
- сплошная линия,
- без маркеров.

***plot "exp_data.txt" using 1:2 with linespoints ls 3 title "Экспериментальные
данные", ***
exp(0.9*x) with lines ls 4 title "Модель exp(0.9x)"

▶ Строим второй график:

- экспериментальные данные `exp_data.txt` стилем №3,
- теоретическую экспоненту `exp(0.9*x)` стилем №4.

`unset multiplot`

▶ Завершаем режим multiplot (объединение графиков).

Разные масштабы осей, цветовые схемы и полупрозрачные маркеры в gnuplot. 🔥

1. Разные масштабы осей

Если, например, хочется, чтобы ось Y была логарифмической (полезно для экспонент):

```
set logscale y
```

▶ Включает логарифмическую шкалу по оси Y.

Можно отдельно логарифмировать ось X:

```
set logscale x
```

Пример: в нашем втором графике для экспоненты добавить:

```
set logscale y
```

(Это покажет экспоненту как прямую на графике!)

Чтобы выключить логарифм:

```
unset logscale y
```

2. Разные цветовые схемы

gnuplot поддерживает цветовые палитры (особенно для тепловых карт, но можно применять и к линиям).

Стандартная смена палитры:

```
set palette defined ( 0 "blue", 1 "green", 2 "yellow", 3 "red" )
```

▶ Задаем переход от синего → зелёного → желтого → красного.

Но для обычных линий лучше заранее задать цвета в `set style line`, как мы делали:

```
set style line 1 lc rgb '#1f77b4' lt 1 lw 2  
set style line 2 lc rgb '#ff7f0e' lt 1 lw 2
```

Пример красивой цветовой пары:

```
set style line 1 lc rgb '#a6cee3' lt 1 lw 2  
set style line 2 lc rgb '#1f78b4' lt 1 lw 2
```

(оба — разные оттенки синего,)

3. Полупрозрачные маркеры

В gnuplot, начиная с версии 5.2+, можно задавать прозрачность через `fillstyle` для областей.

Для линий и точек прозрачность задается через RGBA-цвет.

Как задать полупрозрачный цвет:

```
set style line 1 lc rgb "#0060ad80" pt 7 ps 1.5
```

Обратите внимание:

- #0060ad — обычный синий,
- 80 — уровень прозрачности в HEX (от 00 — полностью прозрачный до ff — полностью непрозрачный).

Итог — маркеры будут полупрозрачными!

Пример:

```
set style line 2 lc rgb "#ff000080" pt 5 ps 1.8
```

(полупрозрачный красный)

Запуск Gnuplot с данными из файла Excel

Чтобы использовать Gnuplot для визуализации данных, хранящихся в файле Excel, вам нужно сначала экспортировать данные из Excel в формат, который Gnuplot может прочитать, например, в текстовый файл (CSV или TSV). Вот шаги, которые вам нужно выполнить:

1. Экспорт данных из Excel:

Откройте файл Excel.

Выделите данные, которые вы хотите экспортировать.

Перейдите в меню "Файл" и выберите "Сохранить как".

Выберите формат "CSV (разделитель - запятая)" или "Текстовый файл (разделитель - табуляция)" и сохраните файл.

2. Подготовка данных:

Убедитесь, что данные в файле имеют правильный формат.

Например, если вы используете CSV, строки должны быть разделены запятыми, а если TSV — табуляцией.

3. Запуск Gnuplot:

Откройте терминал или командную строку.

Запустите Gnuplot, введя команду:

```
bash
```

gnuplot

4. Загрузка данных в Gnuplot:

В Gnuplot используйте команду `plot` для построения графика. Например, если ваш файл называется `data.csv`, вы можете использовать следующую команду:

```
gnuplot
```

```
plot "data.csv" using 1:2 with lines title 'My Data'
```

Здесь `using 1:2` указывает, что Gnuplot должен использовать первый столбец как ось X и второй столбец как ось Y. Вы можете изменить номера столбцов в зависимости от ваших данных.

5. Настройка графика:

Вы можете добавлять различные параметры для настройки графика, такие как заголовки, метки осей и стили линий.

Например:

```
gnuplot
```

```
set title "My Graph Title"
```

```
set xlabel "X-axis Label"
```

```
set ylabel "Y-axis Label"
```

6. Сохранение графика:

Чтобы сохранить график в файл, используйте команду `set terminal` и `set output`. Например, для сохранения в PNG:

```
gnuplot
```

```
set terminal png
```

```
set output 'my_graph.png'
```

```
replot
```

```
set output
```

Таким образом можно визуализировать данные из Excel с помощью Gnuplot

Bash-скрипт для построения 3D-графика в Gnuplot с автоматической генерацией данных и настройками визуализации:

```
#!/bin/bash
```

```
# 1. Генерация данных для 3D-графика (пример: гауссов холм)
```

```
cat > 3d_data.dat <<EOF
```

```
# X Y Z
-5 -5 0.1
-5 -4 0.2
... (и так далее) ...
5 5 0.1
EOF
```

Альтернативно: генерируем данные математической функцией

```
for i in $(seq -5 0.5 5); do
  for j in $(seq -5 0.5 5); do
    # Пример: 3D-гауссова поверхность
    z=$(echo "e(-($i^2 + $j^2)/10)" | bc -l)
    echo "$i $j $z"
  done
done > 3d_function.dat
```

2. Скрипт Gnuplot для 3D-визуализации

```
gnuplot <<- EOF
set terminal qt size 1000,800 enhanced font 'Verdana,12'
set title "3D-график: Гауссов холм" font ",16"
set xlabel "Ось X" font ",14"
set ylabel "Ось Y" font ",14"
set zlabel "Ось Z" font ",14" rotate by 90

# Настройки 3D-отображения
set view 60, 45, 1, 1
set pm3d at b # Цветовая карта у основания
set hidden3d # Удаление невидимых линий
set palette rgbformulae 33,13,10 # Цветовая схема

# Построение графика
splot '3d_function.dat' with pm3d title "Поверхность", \
    " with lines lc rgb "black" title "Контур"
```

```
# Сохранение в файл (опционально)
set terminal pngcairo size 1000,800 enhanced
set output "3d_plot.png"
replot
```

```
pause mouse "Нажмите любую клавишу для выхода..."
EOF
```

3. Удаление временных файлов (раскомментировать при необходимости)

```
# rm 3d_data.dat 3d_function.dat
```

Как использовать:

1. Сохраните скрипт как `plot_3d.sh`
2. Дайте права на выполнение:
3. `bash`
4. Copy
5. Download
6. `chmod +x plot_3d.sh`
7. Запустите:
8. `bash`
9. Copy
10. Download
11. `./plot_3d.sh`

Ключевые особенности:

1. **Генерация данных:**
 - Ручное задание точек или математическая функция
 - Пример использует 3D-гауссову поверхность: $z = \exp(-(x^2+y^2)/10)$
2. **Настройки 3D-графика:**
 - `set view` - угол обзора (60° по азимуту, 45° по углу места)
 - `pm3d` - цветовая карта поверхности
 - `hidden3d` - правильное отображение перекрытий
3. **Визуализация:**
 - Цветная поверхность + черные контурные линии

- Интерактивное окно (Qt) + сохранение в PNG

4. Модификации:

- Для данных из файла замените `3d_function.dat` на ваш файл формата:
- Copy
- Download

x1 y1 z1

x2 y2 z2

- ...

Пример сложной поверхности (замените в скрипте):

bash

Copy

Download

```
z=$(echo "sin(sqrt($i^2 + $j^2)) / (sqrt($i^2 + $j^2) + 0.1)" | bc -l)
```

Скрипт автоматически создаст интерактивное 3D-окно с возможностью вращения (в Qt-терминале) и сохранит изображение.