НИУ Высшая школа экономики
Факультет физики

Информатика для физиков.

Основы работы с GNUPlot.

Plotting with GNUPlot. Basics of basics.

В.Н.Глазков
В.Ю.Попов

Москва
2017

Оглавление

Задачи документа

Этот документ содержит основные сведения по основам работы в программе построения графиков GNUPlot. В нем содержатся некоторые базовые «рецепты» решения стандартных задач. Этот список заведомо не полон, подробные сведения могут быть найдены в разнообразных он-лайн руководствах и справочных материалах (google gor it!).

Используемое ПО

Файл написан по GNUPlot v.5 patchlevel 7 для Windows.

Программа может быть загружена с сайта http://www.gnuplot.info/

На этом же сайте есть доступное к загрузке подробное описание, на многие вопросы можно легко найти ответ при помощи Google, есть много он-лайн руководств (http://www.gnuplotting.org/, http://gnuplot.sourceforge.net/demo/). Имеется встроенная система подсказок, вызываемая командой help в командной строке.

Элементарные примеры

ввод команд

Команды могут вводиться последовательно в командной строке интерфейса GNUPlot либо быть записаны в текстовый файл (традиционное расширение plt).

Файл может «компилироваться» либо из командной строки ОС, либо загружаться в GNUPlot командой load (в графическом интерфейсе для GNUPlot для Windows эта команда сформируется автоматически по команде меню Open)

Далее приведено несколько примеров наборов команд с краткими комментариями и результатами выполнения.

тест №1: первый график

`set terminal windows`	Вывод в окно
`set xrange [0:12]`	X от 0 до 12
`set yrange [0:3]`	Y от 0 до 3
`plot sin(x), x**2, tanh(x)`	Строить графики sin(x), x^2, tanh(x) при указанном диапазоне X

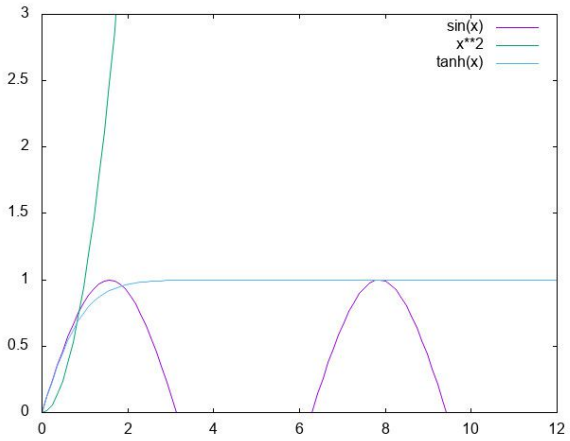

Рисунок 1: Результат выполнения команд теста 1

тест №2: стили линий и символов

`set terminal windows`	
`set xrange [0:12]`	
`set yrange [*:3]`	нижний предел по Y - автоматически
`plot sin(x) with lines lt 4 lw 5, cos(x) with points pt 15 ps 1.5`	опции рисования графиков. sin(x) рисовать линиями типа (lt=linetype) 4, толщиной (lw=linewidth) 5. cos(x) рисовать символами типа (pt=pointtype) 15 размером (ps=pointsize) 1.5 Типы линий и точек можно увидеть командой test в командной строке GNUPlot

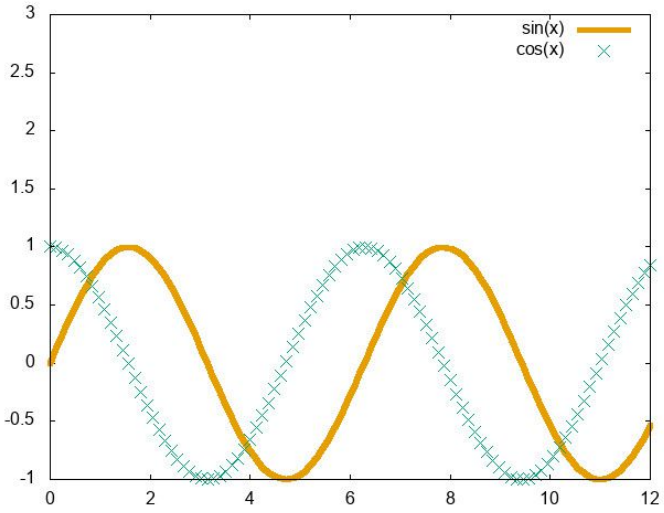

Рисунок 2: Результат выполнения теста №2

тест №3: несколько графиков на одном рисунке

`set terminal windows`	
`set multiplot`	переход в режим рисования нескольких графиков, по окончании работы надо выйти из этого режима
`set size 0.5,0.5`	размер первого графика, относительные единицы
`set origin 0.,0.`	расположение левого нижнего угла первого графика
`plot sin(x)`	
`set size 0.5,0.5`	размер очередного (второго) графика
`set origin 0.5,0.5`	расположение левого нижнего угла второго графика
`plot cos(x)`	
`unset multiplot`	выход из режима рисования нескольких графиков
`set size 1.,1.`	восстановление исходных значений размера и положения начала координат, без этого следующие графики рисовались бы с последним установленным размером и из последнего установленного расположения левого нижнего угла
`set origin 0.,0.`	

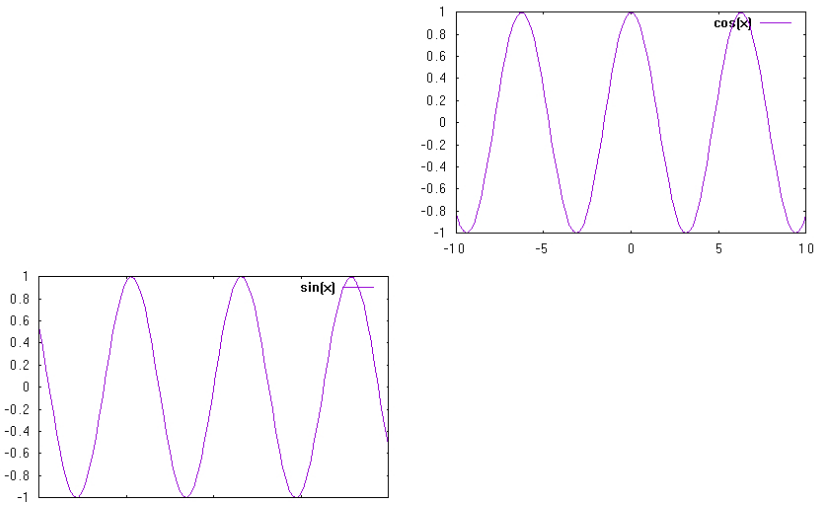

Рисунок 3: Результат выполнения теста №3

тест №4: подписи к осям и к графику

`set terminal windows`	
`set xrange [0:12]`	
`set yrange [*:3]`	
`set title "The test graph #4" offset -10 font "Arial, 20" textcolor "red"`	заголовок графика, параметр offset определяет смещение заголовка вправо или влево
`set xlabel "This is X-axis" font "Helvetica, 14"`	название оси X
`set ylabel "This is Y-axis" font "Helvetica, 14"`	название оси Y
`set xtics 0, pi/2`	задание расположения отсечек по оси X с шагом π/2
`plot sin(x) with lines lt 4 lw 5, cos(x) with points pt 15 ps 1.5`	

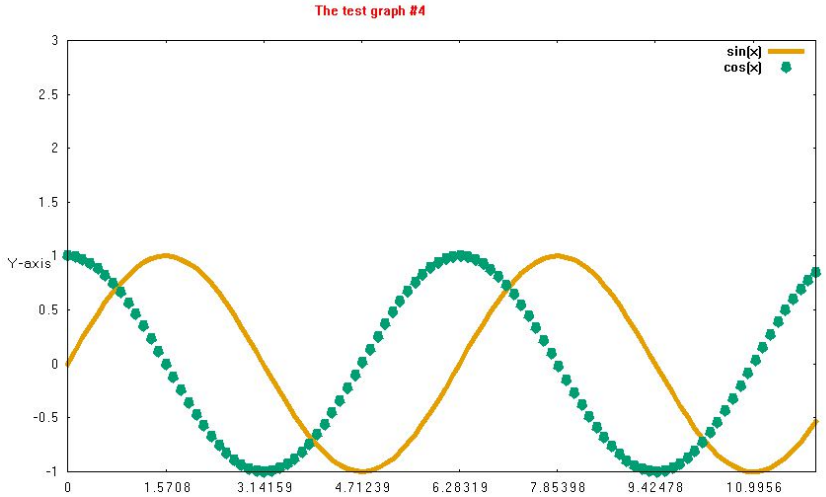

Рисунок 4: Результат выполнения теста №4

тест №5: вывод файла данных

Простейший файл данных состоит из двух колонок — первая колонка X, вторая Y.

set terminal windows	
set xtics autofreq	Автоматический выбор расстановки отсечек по оси X
set xrange [*:*]	Автоматический выбор диапазона значений по осям
set yrange [*:*]	
set title "Plotting sample data"	
set xlabel "Магнитное поле, кЭ"	
set ylabel "Мощность прошедшего СВЧ излучения, отн.ед."	
plot "sampledata.dat" with lines	нарисовать данные из файла, лежащего в текущей директории (в той директории, откуда вызван GNUPlot)

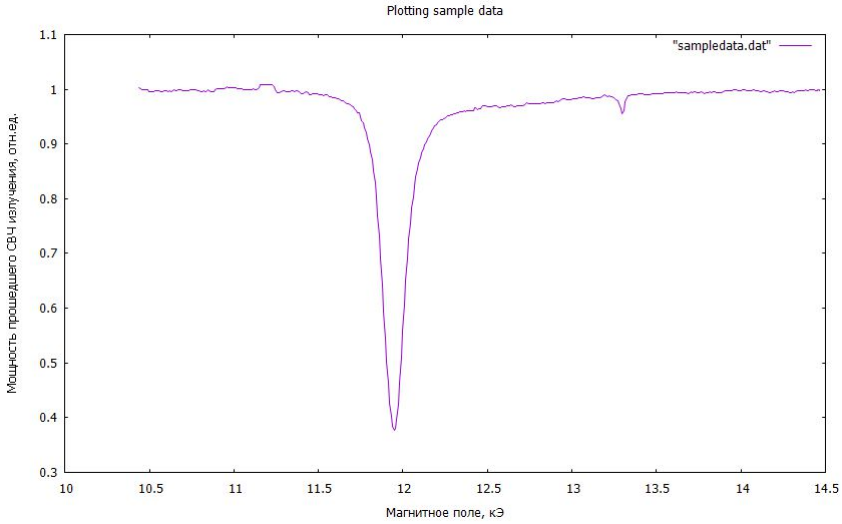

Рисунок 5: Результат выполнения теста №5

тест №6: логарифмический масштаб по осям

`reset`	вернуть все настройки к настройкам «по умолчанию», в частности отменяет определенные ранее размеры графиков, положения графиков и подобные параметры
`set terminal windows`	
`set title "Двойной логарифмический масштаб (log-log plot) превращает степенную зависимость в прямую"`	
`set logscale xy 10`	установить логарифмический масштаб (по основанию 10) по обеим осям
`set xrange [0.01:12]`	
`set yrange [*:150]`	
`plot x**2 with points pt 15`	x**2 обозначает «икс в квадрате»

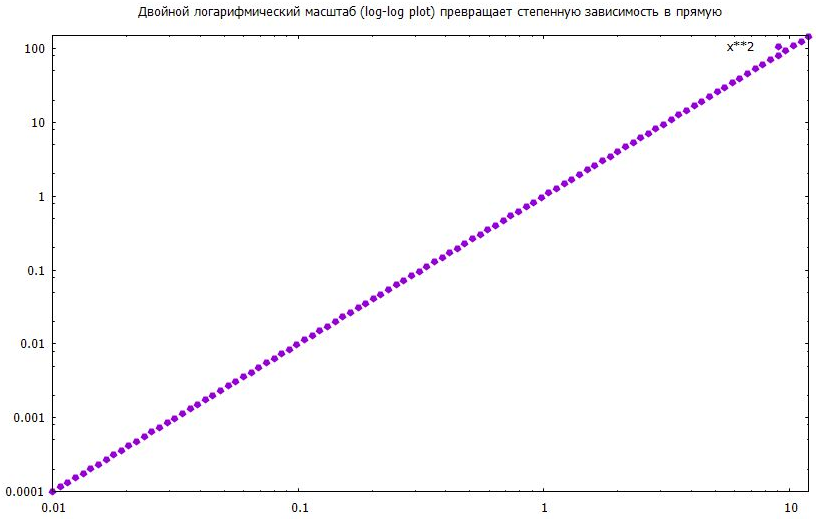

Рисунок 6: Результат выполнения теста №6

тест №7: вывод графика в файл

`reset`	
`#set terminal jpeg`	знак диеза «#» в начале строки оюозначает эту строку, как закомментированную — не подлежащую исполнению; эта пара строк выводит график не в окно на мониторе, а в JPEG-файл output.jpg
`#set output "output.jpg"`	
`#set terminal png`	эта пара строк (если не закомментирована) выводит график в PNG-файл output.png
`#set output "output.png"`	
`set terminal postscript eps enhanced color level1 font "Arial"`	эти строки выводят график в постскрипт файл output.eps и задают информацию, необходимую для корректного отображения русских букв
`set encoding cp1251`	
`set output "output.eps"`	
`set title "Двойной логарифмический масштаб (log-log plot) превращает степенную зависимость в прямую"`	
`set logscale xy 10`	
`set xrange [0.01:12]`	
`set yrange [*:150]`	
`plot x**2 with points pt 15`	

Более сложные примеры отображения данных

Построение графика из таблицы с несколькими колонками

Рассмотрим для примера файл с 3 колонками (файл 3column.dat):

#x	y1	y2
1	1	0.1973753202
2	2	0.3799489623
3	3	0.537049567
4	4	0.6640367703
5	5	0.761594156
6	6	0.833654607
7	7	0.8853516482
8	8	0.9216685544
9	9	0.9468060128
10	10	0.9640275801
11	11	0.97574313
12	12	0.9836748577
13	13	0.9890274022
14	14	0.9926315202
15	15	0.9950547537
16	16	0.9966823978
17	17	0.9977749279
18	18	0.9985079423
19	19	0.9989995978
20	20	0.9993292997

Для отображения на графике желаемого столбца в команде plot нужно указать какие столбцы использовать:

`plot '3column.dat' using 1:2`	первый столбец — координата X, второй столбец — координата Y
`plot '3column.dat' using 1:3`	первый столбец — координата X, третий столбец — координата Y
`plot '3column.dat' using 1:2, '3column.dat' using 1:3`	вывод обеих зависимостей одновременно

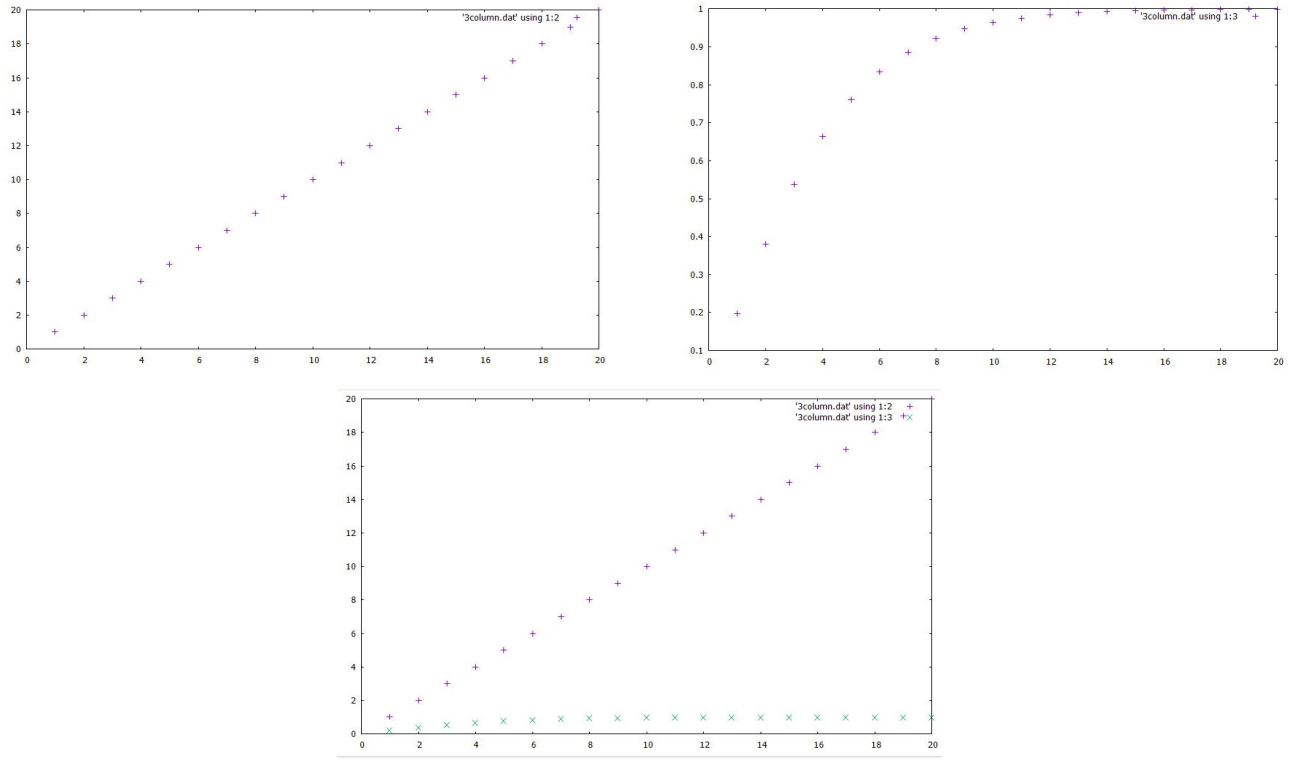

Рисунок 7 Результат построения графиков из файла 3column.dat

Преобразование данных

Иногда необходимо преобразовать данные при построении графика: например, при построении графика зависимости периода маятника T от его длины l ($T=2\pi\sqrt{\frac{l}{g}}$) удобнее строить зависимость $T^2(l)$ или $T(\sqrt{l})$.

В качестве примера возьмём тот же файл 3column.dat, его третья колонка получена по формуле $y2=\tanh(x/5)$, это означает, что в координатах $(\tanh(x/5), y2)$ должна получиться прямая.

`plot '3column.dat' using (tanh($1/5)):3 with points pt 15 ps 2`	знак доллара отсылает к последовательному использованию соответствующей колонки, компактная формула может быть использована непосредственно в команде plot
`F(x)=tanh(x/5)` `plot '3column.dat' using (F($1)):3 with points pt 15 ps 2`	формула преобразования может быть определена отдельной функцией, к которой можно обратиться в команде plot

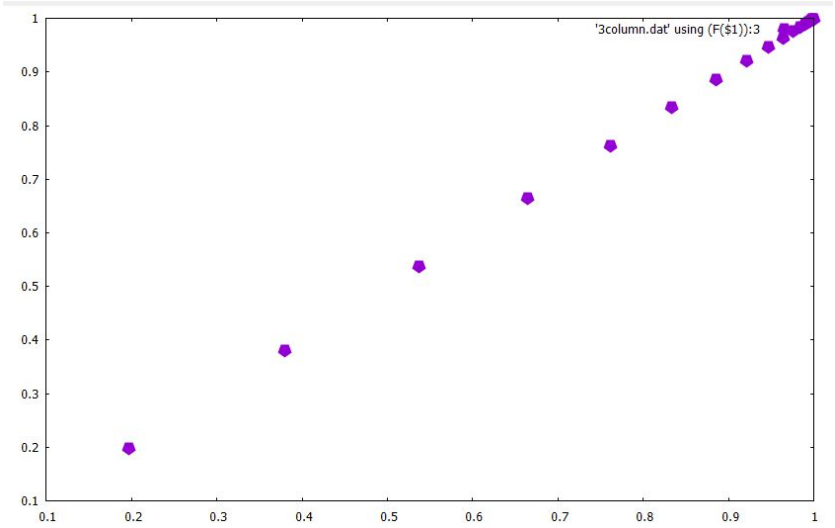

Рисунок 8: Отображение данных третьей колонки файла 3column.dat после преобразования x=tanh(x/5)

Сохранение результатов преобразования в файл-таблицу

После преобразования или вычисления результатов может возникнуть необходимость сохранить полученные данные в виде таблицы — построенные графики содержат только «картинку».

Это можно сделать следующей последовательностью команд:

`set table "output.dat"`	замена вывода в графический терминал на вывод в таблицу с указанным именем
`replot`	повторяет последнюю команду plot (в последовательности нашего изложения — результат предыдущего раздела), может быть заменена командой plot в явном виде
`unset table`	выход из режима создания таблицы

полученный файл имеет вид:

```
# Curve 0 of 1, 20 points
# Curve title: "'3column.dat' using (F($1)):3"
# x y type
 0.197375  0.197375  i
 0.379949  0.379949  i
 0.53705  0.53705  i
 0.664037  0.664037  i
 0.761594  0.761594  i
 0.833655  0.833655  i
 0.885352  0.885352  i
 0.921669  0.921669  i
 0.946806  0.946806  i
 0.964028  0.964028  i
 0.975743  0.975743  i
 0.983675  0.983675  i
 0.989027  0.989027  i
```

```
 0.992632  0.992632  i
 0.995055  0.995055  i
 0.996682  0.996682  i
 0.997775  0.997775  i
 0.998508  0.998508  i
 0.999  0.999  i
 0.999329  0.999329  i
```

Столюцы с числами — это X и Y координаты точек после преобразования, буква в третьей колонке указывает попала ли эта точка в установленный интервал отображения (i), не попала (o) или точка оказалась неопределённой (u).

Полярные координаты

`set polar`	переход в режим использования полярных координат
`plot cos(2*t) with lines lt 7 dt 3 lw 3 ,sin(4*t) lt 12 lw 3`	изображение двух тригонометрических функций с разными стилями линий, опция dt (dashtype) определяет тип пунктира
`unset polar`	выход из режима использования полярных координат

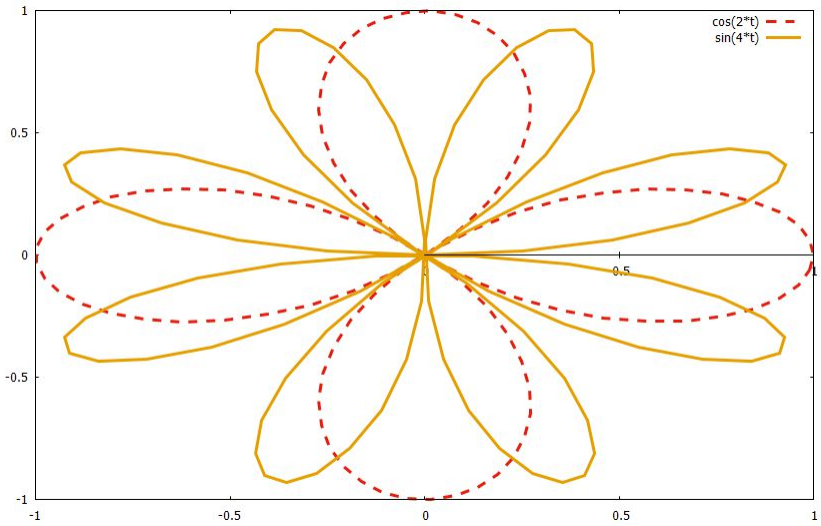

Рисунок 9: Изображение графиков в полярных координатах

Можно дополнительно отметить здесь некоторую «угловатость» графика $\sin(4t)$. Это дефект отображения, связанный с большим шагом при вычислении функции. Шаг можно уменьшить (увеличить число вычисляемых точек) командой:

`set samples 1000` `replot`	установить вычисление 1000 точек и перерисовать график

Параметрические графики

Простейшим примером является серия команд:

`set parametric`	Переход в режим построения параметрической кривой
`set trange [0:2*pi]`	диапазон изменения параметра *t*
`# Parametric functions for a circle`	
`fx(t) = r*cos(t)`	
`fy(t) = r*sin(t)`	
`r=1`	
`plot fx(t),fy(t)`	
`unset parametric`	выход из режима параметрического построения

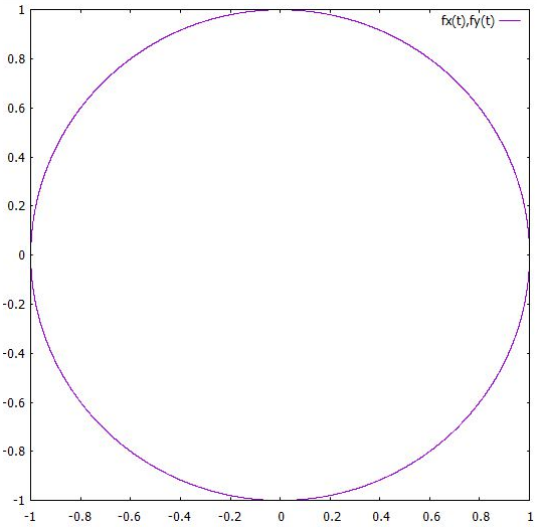

Рисунок 10: Результат построения параметрической кривой

3D-поверхности

Простейшая последовательность команд для построения трёхмерной поверхности:

`set pm3d`	включение стиля рисования трёхмерных поверхностей с цветовой картой
`splot [-1.5:1.5] [-1.5:1.5] [-`	построение графика функции

`1.2:1.2] sin(x**2+y**2)`	$z=\sin\left(x^2+y^2\right)$, интервалы по осям заданы непосредственно в команде splot

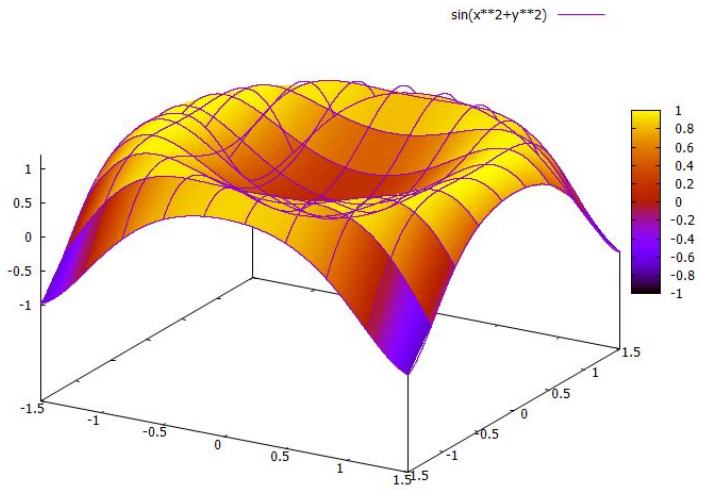

Рисунок 11: Построение трёхмерной поверхности

Построение трёхмерной поверхности по данным из таблицы-файла производится аналогично тому, как строились обычные графики.

Построение гистограмм распределения данных

Иногда оказывается необходимым построить гистограмму распределения данных: посчитать сколько точек попадает в определённый интервал. Для примера возьмём опять третью колонку файла 3column.dat. По графику видно, что число точек в окрестности 1 оказывается большим (функция $y=\tanh\left(x/5\right)$ асимптотически приближается к единице).

Необходимая последовательность команд:

`reset`	
`bin_width = 0.1`	определяет переменную, равную желаемой ширине 'окна', внутри которого точки считаются попадающими близко друг к другу
`set boxwidth 0.9*bin_width absolute`	определяет стиль рисования гистограммы: ширина столбцов 90% от ширины окна
`set style fill solid 1 noborder`	определяет стиль рисования столбцов гистограммы
`bin_number(x) = floor(x/bin_width)`	вычисляет в какое 'окно' должна попасть точка со значением x
`rounded(x) = bin_width * ( bin_number(x) + 0.5 )`	вычисляет координату середины 'окна'
`plot '3column.dat' using (rounded($3)):(1) smooth frequency with boxes`	опция smooth frequency относится к одной из опций «сглаживания» кривых (с другими ключами происходит нормальное

	сглаживание с вычислением среднего значения близких точек) — при каждом появлении точки с координатой x к координате y добавляется указанное число; в данном случае в качестве координаты x используется вычисленная координата середины окна, а добавляемое число равно 1; опция with boxes отображает результат в виде гистограммы

Эта последовательность команд немного некорректно отображает результат (смещает 'окно' на 0.5) для случая единичной ширины окна при изначально целочисленных результатах. Поправка очевидна — убрать прибавление 0.5.

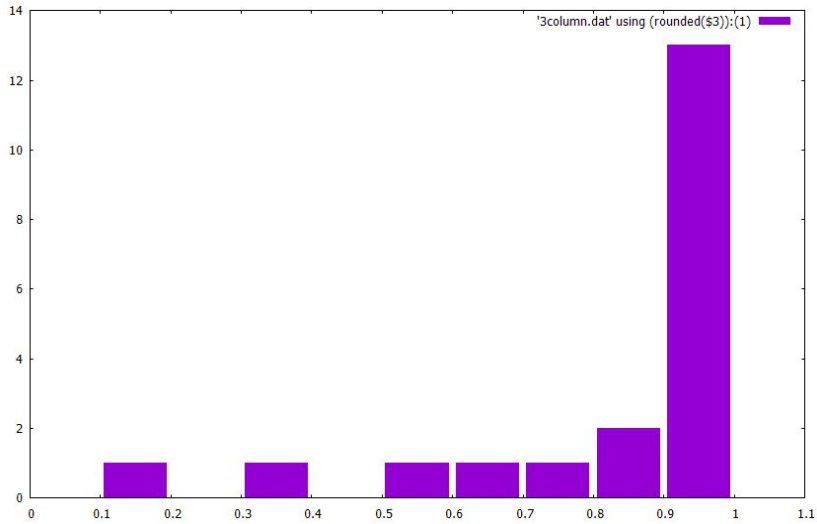

Рисунок 12: Построение гистограммы

Данные с погрешностью и подгонка данных

Понятие погрешности

Реально измеряемые данные никогда не измеряются точно — всегда есть некоторая погрешность измерения. Погрешность показывает, насколько достоверно получены измеряемые данные. Во многих случаях понятие погрешности естественно — например, взвешивающие автомобиль весы на пункте весового контроля измеряют вес автомобиля с точностью порядка десятка или нескольких десятков килограмм, а лабораторные весы могут измерять малые массы с микрограммной точностью.

Погрешность измерения указывают так: $m = (1.07 \pm 0.17)$ мг .

Погрешность не измеряется точно, а *оценивается* по результатам измерения, поэтому её обычно указывают с точностью округления около 10%. «Правилом правой руки», подходящим в большинстве случаев, является следующее:

- если первая ненулевая цифра в погрешности равна 1, то удерживается две цифры, если больше 1 — одна цифра (т.е., нормальной записью будет ± 0.12 ± 1300 ± 6);
- среднее измеренное значение указывается с той же точностью, что и последняя

ненулевая цифра погрешности (т.е. 112400 ± 1300 правильно, а 112436 ± 1300 — неправильно);

- в промежуточных расчётах допустимо для избежания накопления ошибки округления указывать на одну значащую цифру больше.

В рамках курса информатики мы будем рассматривать только случайную погрешность — отличия измеряемого результата от «идеала», связанную с действием неконтролируемых и предположительно случайных факторов. Систематическую погрешность — поддающееся учёту и корректировке отличие измеряемого результата от «идеала» мы не рассматриваем. Примером систематической погрешности является, например, прилипший к чашке лабораторных весов комочек клея, который *систематически* смещает результаты измерения (увеличивая их на свою массу). Примером случайной погрешности является, например, шум при измерении слабых электрических сигналов, влияние на измерения малой массы порывов сквозняка в комнате или *случайный* перекос чашки весов, возникающий при установке на неё груза. Это различие нестрогое, часто подробным анализом и исправлением экспериментальной методики можно уменьшить случайную часть погрешности.

Немного экспериментальной статистики: нормальное распределение

Важным для анализа погрешностей является тот факт, что во многих случаях (но не всегда) случайная погрешность складывается под воздействием большого числа нескоррелированных случайных процессов. Так как каждый случайный процесс случайным образом влияет и в сторону увеличения, и в сторону уменьшения измеряемого значения, возникает вопрос -как суммируются эти отклонения, как в результате окажутся распределены вероятности получить измеренное значение отстоящее от «идеала» на некоторую величину.

Эта задача родственна классической (и анекдотической) задаче «о пьяном моряке»: вышедший из портового кабака моряк делает N шагов в случайном направлении (направо или налево). Как далеко он уйдёт к последнему шагу?

Задачу можно смоделировать при помощи монетки и некоторой усидчивости. Мы воспользуемся простым компьютерным моделированием. Содержательный фрагмент кода на C++ тривиален:

```
for(i=0;i<ntries;i++)
      {
      sum=0;
      for(j=0;j<nsteps;j++)
            {
            sum+=2*(int)(rand()%2)-1;
            }
      printf("%i\n",sum);
      }
```

Повторив моделирование (ntries) 10000 раз для последовательности из 25 шагов (nsteps). Результатом оказывается гладкое распределение возможных исходов, показанное на рисунке 13.

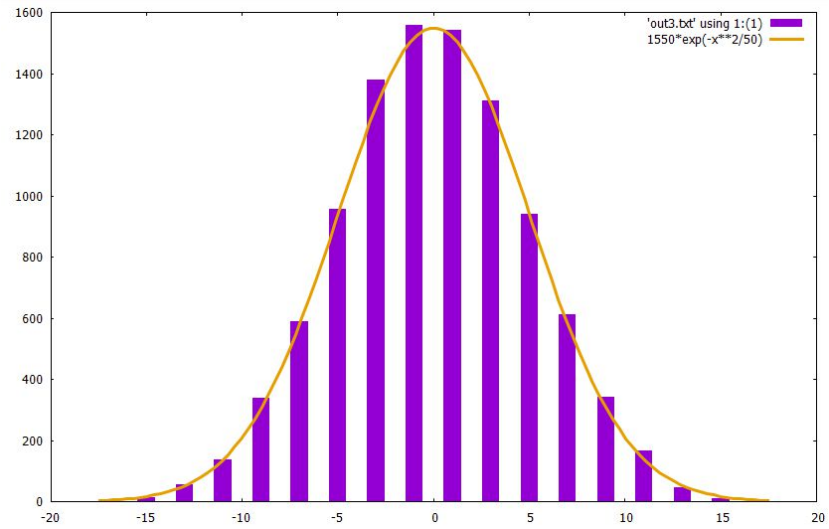

Рисунок 13 Результат моделирования конечного смещения «пьяного моряка» после 25 шагов, смоделировано 10000 опытов. Чётные конечные положения отсутствуют из-за нечётности полного числа шагов.

Это плавное распределение может быть описано *гауссовой функцией* $f \propto \exp\left(-x^2/50\right)$ (сравните гистограмму с жёлтой кривой на рисунке 13), здесь число 50 — это удвоенное число шагов (доказательство за рамками курса). Можно показать, что среднее значение квадрата отклонения от среднего значения $\langle (x-x_0)^2 \rangle$ также окажется равно числу шагов в нашей модели. Это среднее значение квадрата отклонения для гауссова распределения называют квадратом дисперсии $\sigma^2 = \langle (x-x_0)^2 \rangle$, дисперсия σ используется в качестве погрешности. В нашем опыте $\sigma = 5$, при этом примерно 2/3 конечных результатов оказывается в интервале $[x_0-\sigma, x_0+\sigma]$, 95% в интервале $[x_0-2\sigma, x_0+2\sigma]$ и более 99% в интервале $[x_0-3\sigma, x_0+3\sigma]$.

На практике это означает, что так как случайные погрешности измерения формируются как сумма большого количества случайных отклонений, то во многих случаях (но не всегда!) результаты измерения при повторении измерения большое число раз окажутся распределены по Гауссу. Ширина такого распределения характеризуется его дисперсией, некоторая условность такого определения (с вероятностью около 1/3 отклонение превышает дисперсию) обосновывает сделанное выше утверждение о приблизительной оценке погрешности.

Метод наименьших квадратов

Простейшая реализация метода

Часто возникающей задачей является проведение через имеющийся набор точек $\{y_i; x_i\}$ некоторой модельной кривой $y(x)$, в простейшем случае прямой $y = ax + b$. Как выбрать параметры модельной кривой оптимальным образом?

Одним из таких способов является метод наименьших квадратов — параметры модельной кривой выбираются так. чтобы минимизировать сумму квадратов отклонения модельных результатов от реальных (эту сумму называют также невязкой): $R = \sum_{i=1}^{N} \left(y(x_i) - y_i\right)^2$.

Невязка является функцией изменяемых параметров модельной кривой и её минимум может искаться по стандартным правилам поиска минимума функции нескольких переменных. Для простейшего случая линейной модельной зависимости:

$$R(a,b)=\sum_{i=1}^{N}\left(a\,x_i+b-y_i\right)^2=\sum_{i=1}^{N}\left(a^2x_i^2+b^2+y_i^2+2a\,x_i b-2a\,x_i\,y_i-2b\,y_i\right)$$

$$\frac{\partial R}{\partial a}=0=2a\sum_{i=1}^{N}x_i^2+2b\sum_{i=1}^{N}x_i-2\sum_{i=1}^{N}x_i\,y_i$$

$$\frac{\partial R}{\partial b}=0=2bN+2a\sum_{i=1}^{N}x_i-2\sum_{i=1}^{N}y_i$$

Решая получившуюся пару уравнений получим (пределы суммирования от 1 до N убраны для компактности):

$$b=\frac{1}{N}\left(\sum y_i-a\sum x_i\right)$$

$$a=\frac{N\sum x_i\,y_i-\sum x_i\sum y_i}{N\sum x_i^2-\left(\sum x_i\right)^2}$$

В принципе, аналогичным образом можно получить аналитические формулы для некоторых других модельных зависимостей (например, для многочленов n-й степени). Однако в общем случае такая задача может не иметь компактного аналитического решения (и даже скорее не иметь аналитического решения вовсе). Поэтому при компьютерной подгонке данных минимизация невязки проводится численными методами.

Несколько слов о численной минимизации функций

Численные методы поиска минимума функции многих переменных $R(a_1,a_2\ldots a_n)$ подробно разработаны (см., например, классическую книгу Numerical Recipies), мы здесь не будем рассматривать конкретные алгоритмы[1] и примем как данность, что какой-то из таких алгоритмов, обладающий достаточной универсальностью, реализован в GNUPlot или другом используемом для подгонки данных ПО. Отметим несколько общих моментов, свойственных всем численным методам поиска минимума:

1. Численный поиск минимума требует задать начальное приближение параметров минимизации $\left\{a_i^{(0)}\right\}$ (в некоторых алгоритмах требуется несколько начальных точек). Выбор этих точек — ответственность пользователя.
2. Всякий численный метод ищет *локальный* минимум (в каком-то смысле, достаточно «близкий» к исходному приближению). Поэтому при неудачном выборе начального приближения (как правило, слишком «далёкого» от точного значения) алгоритм минимизации может не найти правильное решение.
3. Всякий численный метод вычисляет *приближённый* ответ. Обычно вычисление идёт методом последовательных приближений (итераций). Критерием прекращения

1 Для функции одной переменной поиск минимума на некотором отрезке может выполняться известным методом деления отрезка пополам. Для функции многих переменных в качестве наивной иллюстрации возможного (но не оптимального) метода поиска минимума можно взять алгоритм «спуска по градиенту», когда последовательно делаются «шаги» (изменения параметров) в направлении наибыстрейшего убывания минимизируемой функции.

итерационной процедуры обычно является момент, когда изменение невязки между последовательными итерациями $\delta_p = R^{(p)} - R^{(p-1)}$ стало меньше некоторой наперёд заданной точности.[2]

4. Возможны неприятные случаи, когда алгоритм минимизации работает неустойчиво. Одним из примеров такого типа является случай, когда в пространстве параметров функция $R(a_1 \dots a_n)$ имеет узкий минимум (аналог каньона в терминах земной географии): малые изменения одного из параметров сильно меняют невязку, а большие изменения другого параметра её почти не меняют. Другим примером является случай, когда по каким-то причинам параметры оказались не независимы (тривиальный пример $R(a_1, a_2) = (a_1 - a_2)^2$): тогда минимальное значение достигается на целом множестве параметров. В случаях такого типа численные алгоритмы часто «зацикливаются», для программного ограничения этой проблемы обычно имеется некоторое максимальное число итераций, которое можно допустить. Если процесс подгонки прекратился по достижению предельного числа итераций имеет смысл внимательно проверить, не возникла ли какая-то проблема.

Учёт «веса» точек или их погрешности

В рассмотренном выше применении метода наименьших квадратов все точки считались одинаково надёжными. В реальных измерениях разные данные могут иметь разные погрешности — какие-то измерены точнее, какие-то менее точно. Естественно пытаться учесть это при поиске оптимальной модельной кривой.

Сохраняя идеологию метода наименьших квадратов, можно изменить функцию невязки так, чтобы она учитывала разные точки с разным «весом»:

$$R(a_1 \dots a_n) = \sum_{i=1}^{N} w_i \left(y(x_i) - y_i \right)^2 \quad ,$$

где w_i — некоторый назначенный пользователем весовой коэффициент, который тем больше, чем выше доверие к i-ой точке.

Если для всех точек определены погрешности σ_i , то удобным является вычисление суммы квадратов нормированных на погрешность отклонений:

$$R(a_1 \dots a_n) = \sum_{i=1}^{N} \left(\frac{y(x_i) - y_i}{\sigma_i} \right)^2 \quad .$$

Естественно, любой из этих способов не идеален, но они разумно работают в большинстве простых случаев.

Точность определения параметров методом наименьших квадратов

Многие компьютерные программы (например, Origin) после подгонки данных выдают не только оптимальные значения параметров, но и их погрешность. Использование этой погрешности требует некоторой осторожности. Её вычисление основано на некоторых результатах статистики и теории вероятности, а также на предположении, что используемая модельная зависимость $y(x)$ — правильная: то есть, что экспериментальные данные

2 Обратите внимание, что так как точное решение для оптимума нам неизвестно, невозможно требовать в качестве критерия остановки отличие текущей невязки от оптимальной.

$\{x_i; y_i\}$ должны были бы подчиняться этой зависимости, но отличаются от неё случайным образом, причём эти отклонения распределены по нормальному (гауссову) распределению.

Чисто графически эта ситуация соответствует тому, что модельная кривая проходит более-менее посередине множества экспериментальных точек и число точек, расположенных выше или ниже модельной кривой примерно равно. Более того *для случая равного веса (равной погрешности) точек и при большом количестве точек*[3] по свойствам нормального распределения невязка для оптимально проведённой кривой $R_0 = \sum \left(y(x_i) - y_i\right)^2 \approx N\,\sigma^2$, где σ — дисперсия (погрешность) экспериментальных точек. В качестве нестрогого аргумента, предположим, что при изменении некоторого параметра модельной кривой все значения $y(x_i)$ увеличились на величину σ. Графически (по свойствам нормального распределения) это соответствует тому, что примерно 2/3 точек оказалось ниже смещённой модельной кривой, а 1/3 выше. Невязка для такой кривой:

$R = \sum \left(y(x_i) + \sigma - y_i\right)^2 \approx N \left\langle \left(y(x_i) + \sigma - y_i\right)^2 \right\rangle = N\left[\left\langle \left(y(x_i) - y_i\right)^2 \right\rangle + \sigma^2 + 2\,\sigma \left\langle y(x_i) - y_i \right\rangle\right]$, здесь угловые скобки $\langle .. \rangle$ обозначают усреднение. В последнем выражении первое слагаемое по определению равно σ^2, а последнее зануляется (среднее смещение данных от идеальной модельной кривой равно нулю). Таким образом $R = 2\sigma^2 = 2\,R_0$. Эти рассуждения, в принципе, позволяют автоматически оценивать точность определения параметров модельной кривой: в качестве погрешности определения параметров модельной кривой можно взять изменение параметров, при которых, например, удваивается невязка.

Однако это автоматическое определение существенно опирается на несколько математических предположений: «правильность» модельной кривой, нормальное распределение отклонений, большое число точек. В реальных случаях это может не всегда выполняться, что требует некоторой осторожности при использовании «выданных компьютером» значений. Иногда для оценки точности определения модельного параметра проще оказывается построить серию модельных кривых со значениями этого параметра, отличающимися от оптимального, и посмотреть, при каком отклонении значения параметра модельная кривая начнёт существенно отличаться от экспериментальных данных.

Сопоставление модельных кривых

Иногда возникает задача, проверить какой из нескольких возможных моделей описываются экспериментальные данные. Надёжного общего правила не существует, математическая статистика имеет инструменты типа критерия χ^2, однако (как уже отмечалось выше) при этом требует дополнительного обоснования соответствие реальных данных математическим предположениям.

Отметим несколько простых соображений:

1. Наличие физической модели, лежащей в основе некоторой модельной кривой, является плюсом такой модели;
2. При эмпирическом проведении модельной кривой (без теоретического обоснования, например при проведении квадратичной параболы вблизи экстремума либо при проведении простых тригонометрических функций через периодические зависимости) более надёжны наиболее простые функциональные зависимости;

3 При большом количестве точек сумма какой-то величины по всем точкам примерно равна среднему значению этой величины, умноженному на число точек.

3. Модельная кривая с меньшим числом подгоночных параметров более надёжна. Во всяком случае, число подгоночных параметров должно быть меньше (в идеале — много меньше), чем число подгоняемых точек.
4. Если параметры модели существенно зависят от положения одной-двух точек, то есть очень большой риск некорректной интерпретации результатов[4].

Отображение данных с погрешностями

Есть три основных варианта при отображении данных $\{x_i; y_i\}$:

- погрешность есть только у координаты X $\{x_i \pm \delta x_i; y_i\}$;
- погрешность есть только у координаты Y $\{x_i; y_i \pm \delta y_i\}$;
- погрешность есть у обеих координат: $\{x_i \pm \delta x_i; y_i \pm \delta y_i\}$.

Для отображения таких данных таблица данных должна содержать дополнительный столбец в первом и втором случае и два дополнительных столбца в третьем случае. Рассмотрим модельную таблицу данных (errortest.dat)

```
0.  0.   0.2  0.2
1.  2.0  0.3  0.1
1.5 1.1  0.2  0.5
2.3 2.1  0.1  0.2
3.1 2.8  0.3  0.4
3.7 4.1  0.2  0.1
4.8 4.5  0.3  0.4
```

Мы будем считать, что первый столбец — это координата X, второй столбец – Y, третий и четвёртый — погрешности по X и Y.

Формат команд, необходимых для построения соответствующих графиков (рисунок 14):

`plot 'errortest.dat' using 1:2:3 with xerrorbars pt 5 ps 2 lc rgb 'blue'`	команда using определяет используемые столбцы, команда with xerrorbars определяет, что третий из упомянутых в команде using столбцов обозначает погрешность по оси X, дальнейшие команды определяют стиль рисования точек и их цвет
`plot 'errortest.dat' using 1:2:4 with yerrorbars pt 15 ps 2 lc rgb 'red'`	команда with yerrorbars определяет, что третий из упомянутых в команде using столбцов обозначает погрешность по оси Y

4 Связанная с этим история рассказана в знаменитой книге «Вы конечно шутите, мистер Фейнман!»: *«...существует принцип, что если точка находится на конце диапазона данных,– последняя точка,– то она не слишком хорошая, потому что если бы она была хорошей, то с ее помощью определили бы еще одну точку...»*

`plot 'errortest.dat' using 1:2:3:4 with xyerrorbars pt 8 ps 2 lc rgb 'green', 'errortest.dat' with lines dt 2 lw 2`	команда with xyerrorbars определяет, что третий из упомянутых в команде using столбцов обозначает погрешность по оси X, а четвёртый столбец — погрешность по осиY; повторное обращение к файлу errortest.dat соединяет те же точки пунктирной кривой

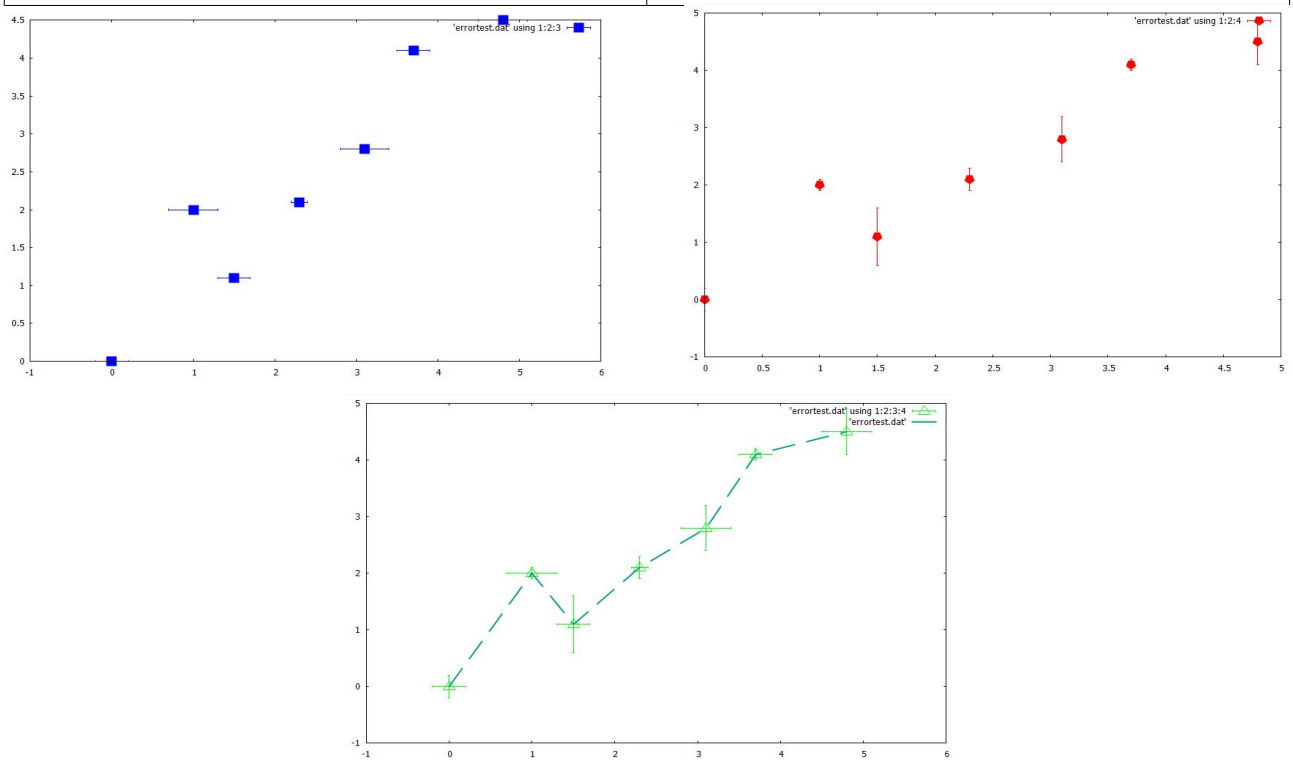

Рисунок 14: Примеры построения данных с погрешностью

Простейшая подгонка данных

В качестве простейшей подгонки построим оптимальную кривую вида $y = a\,x$ через точки из файла errortest.dat. Последовательность команд:

`f(x)=a*x`	определяет функцию для модельной зависимости
`a=0.5`	начальное приближение
`fit f(x) 'errortest.dat' using 1:2 via a`	подогнать при помощи функции f(x) данные из файла errortest.dat, используя первую колонку как X и вторую колонку как Y с варьируемым параметром a
`plot 'errortest.dat' using 1:2:3:4 with xyerrorbars,f(x)`	отобразить на графике исходные данные и подгоночную кривую (оптимальные значения подгоночных параметров остаются в памяти компьютера)

Результаты исполнения команды fit отображаются в окне GNUPlot и автоматически сохраняется в файле fit.log (новые результаты дописываются в конец файла):

```
iter      chisq       delta/lim  lambda   a
   0 1.4310000000e+01  0.00e+00 1.40e+00  5.000000e-01
   1 1.7233432318e+00 -7.30e+05 1.40e-01  9.223533e-01
   2 1.5235554357e+00 -1.31e+04 1.40e-02  9.826034e-01
   3 1.5235550292e+00 -2.67e-02 1.40e-03  9.826895e-01
iter      chisq       delta/lim  lambda   a

After 3 iterations the fit converged.
final sum of squares of residuals : 1.52356
rel. change during last iteration : -2.66855e-007

degrees of freedom    (FIT_NDF)                        : 6
rms of residuals      (FIT_STDFIT) = sqrt(WSSR/ndf)    : 0.503911
variance of residuals (reduced chisquare) = WSSR/ndf   : 0.253926

Final set of parameters            Asymptotic Standard Error
=======================            ==========================
a                  = 0.98269       +/- 0.06802      (6.922%)
```

Результаты показаны на рисунке 15.

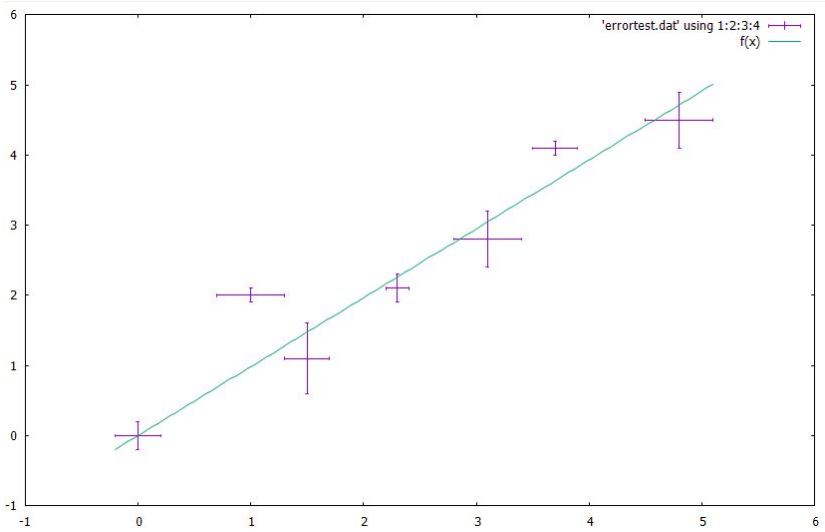

Рисунок 15: Результат подгонки