

[КАК СТАТЬ АВТОРОМ](#)[Иди сюда, первая работа](#)[Masterpiece as a service: итог...](#)**2569.81**

Рейтинг

RUVDS.comVDS/VPS-хостинг. Скидка 15% по коду **HABR15****dlinyj**

2 сен 2020 в 16:12

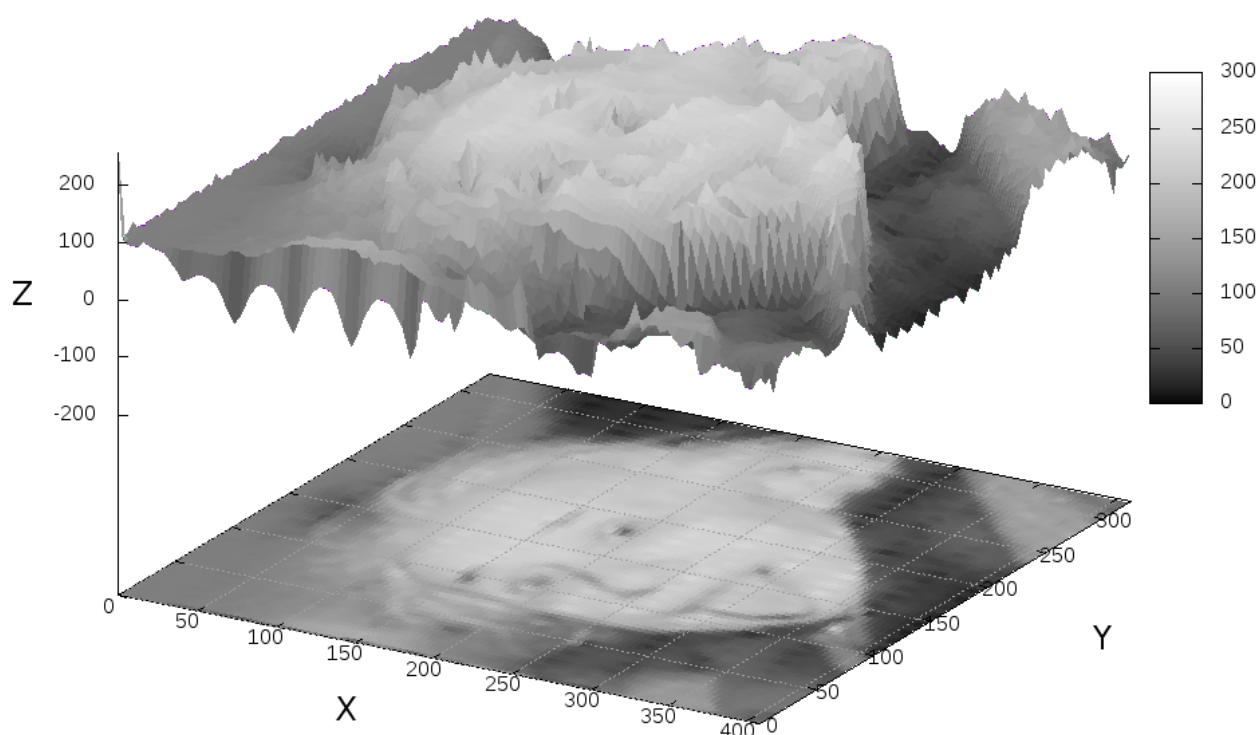
Gnuplot и с чем его едят

🕒 11 мин 👁 40K

Блог компании RUVDS.com, Визуализация данных*, Лайфхаки для гиков

[Тutorial](#)

Albert Einstein



Наверняка многие из вас листая западные научные издания видели красивые и простые графики. Возможно некоторые из вас задумывались в чём же эти учёные мужи визуализируют свою данные. И вот есть шикарный и очень простой инструмент для

построения графиков, который есть практически везде: Windows, linux, android, и прочих, уверен даже есть под ДОС. Он надёжен, прост и позволяет представить в виде красивых графиков любые текстовые-табличные данные.

Почему именно gnuplot?

Если вы уже читали мои статьи «Одновременный speedtest на нескольких LTE-модемах», «Гармонические колебания», «Создаём аппаратный генератор случайных чисел»), то могли обратить внимание на красивые графики.

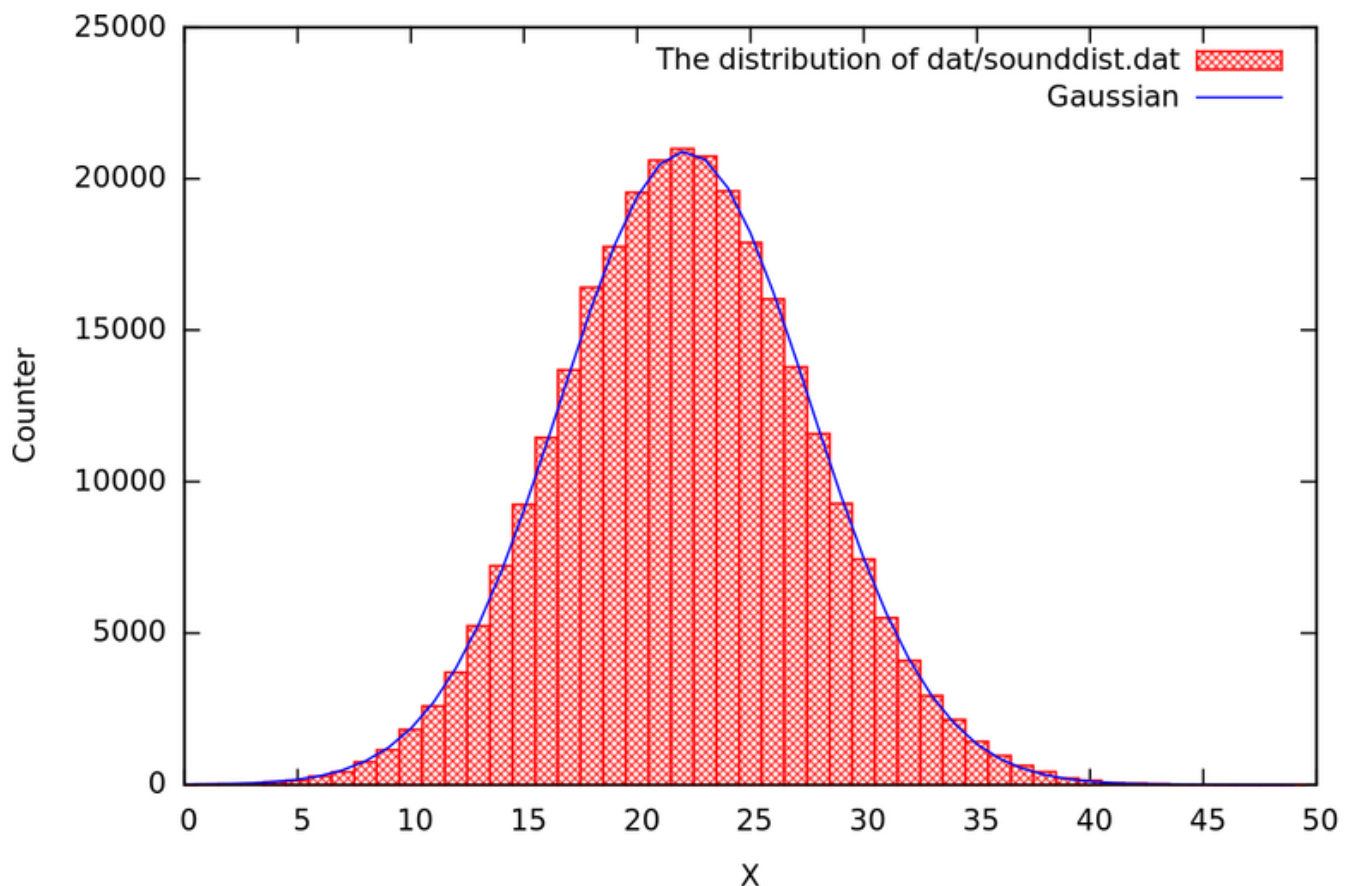
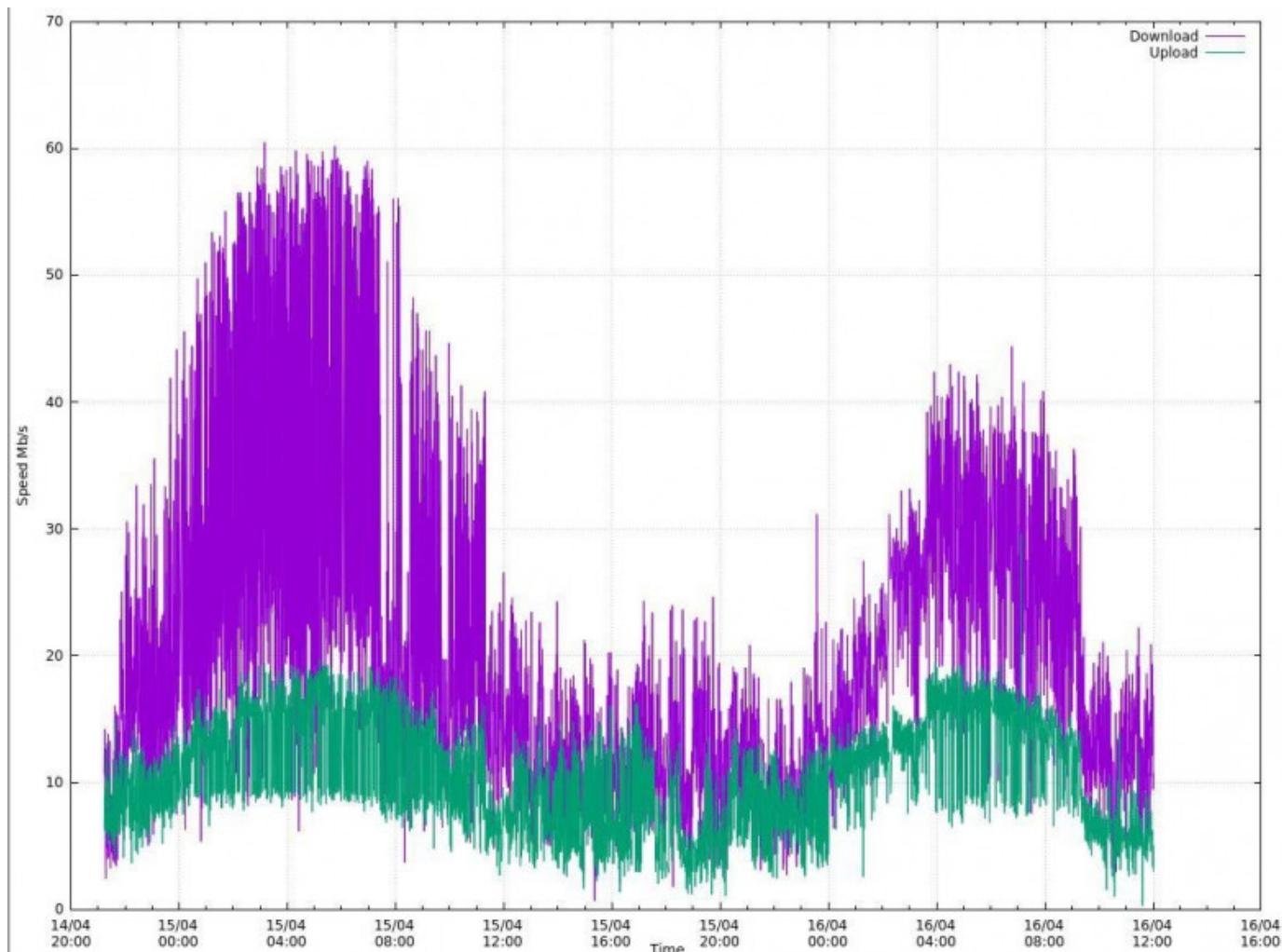


График из поста про генератор случайных чисел



Картинка из поста про speedtest модемов

Графики простые и классные. Наиболее ценное преимущество gnuplot заключается в том, что для их построения вам нужен только текстовый файл с исходными данными, gnuplot на вашей любимой ОС (хоть OpenWRT) и любимый тестовый редактор vim.

На первый взгляд может показаться, что gnuplot сложнее в использовании для построения графиков чем MS Excel. Но это только так кажется, порог вхождения чуть выше (это вам не мышкой наклацать, тут надо документацию читать), но на практике выходит намного проще и удобнее. Один раз написал скрипт и используешь его всю жизнь. Мне реально намного сложнее построить график в Excel, где всё не логично, нежели в gnuplot. А главное преимущество gnuplot, то что его можно встраивать в свои программы и на ходу визуализировать данные. Так же gnuplot без особых проблем строит график с 30-ти гигабайтового файла статистических данных, тогда как Excel просто падал и не мог его открыть.

К плюсам gnuplot можно отнести то, что он легко интегрируется в код на стандартных языках программирования. Есть готовые библиотеки для многих языков, лично я сталкивался с php и python. Таким образом можно генерировать графики прямо из своей программы.

Для примера скажу, что моя хорошая подруга, когда писала диссертацию освоила gnuplot (с моей подачи). Она ни разу не технарь, но разобралась за один вечер. После чего строила графики только там, и уровень её работы стал выгодно отличаться на фоне коллег, использующих Excel. Лично для меня показателем высокого качества научной работы являются графики, построенные специализированными программами.

Таким образом, gnuplot – это просто, доступно и красиво. Едем дальше.

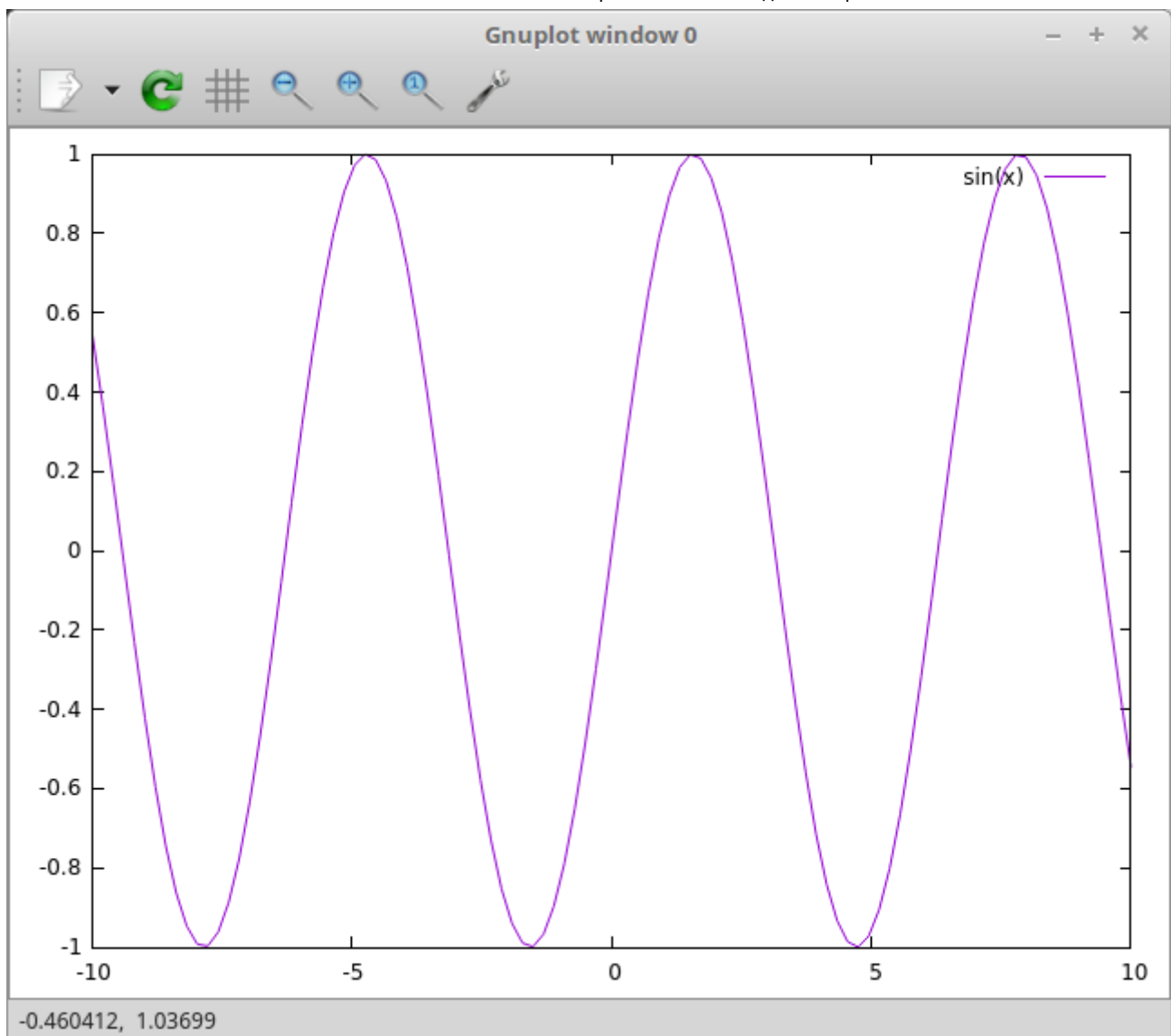
Gnuplot — применение

Работа с gnuplot возможна двумя способами: командный режим и режим выполнения скриптов. Рекомендую сразу использовать второй режим, как наиболее удобный и быстрый. Тем более, что функционал абсолютно тот же.

Но для старта погоняем gnuplot в командном режиме. Запускаем gnuplot из командной строки, либо тем способом который доступен для вашей ОС. Обучение можно начать даже не с чтения этой статьи, а прямо с самой первой команды **help**. Она выведет шикарную справку и дальше по ней можно идти. Но мы пробежимся по основным моментам.

График строится командой **plot**. В качестве параметров команды можно задать функцию, либо имя файла данных. А так же какой столбец данных использовать и чем соединять точки, как их обозначать и т.д. Давайте проиллюстрирую.

```
gnuplot> plot sin(x)
```



После выполнения команды у нас откроется окно, где будет график синуса, с подписями по умолчанию. Давайте улучшим этот график, заодно разберёмся с дополнительными опциями.

Подпишем оси.

```
set xlabel "X"
```

Задаёт подпись для оси абсцисс.

```
set ylabel "Y"
```

Задаёт подпись для оси ординат.

Добавим сетку, чтобы было видно где построен график.

```
set grid
```

Мне не нравится, что по оси ординат синусоида упирается в конец графика, поэтому зададим лимиты значений, которыми будет ограничен график.

```
set yrange [-1.1:1.1]
```

Таким образом график у нас будет отрисован от минимального значения $-1,1$ до максимального $1,1$.

Точно так же, устанавливаю диапазон для оси абсцисс, чтобы был виден только один период синусоиды.

```
set xrange[-pi:pi]
```

Надо бы добавить заголовок к нашему графику, чтобы всё по феншую было.

```
set title "Gnuplot for habr" font "Helvetica Bold, 20"
```

Обратите внимание, что можно задавать шрифты и их размер. Какие шрифты можно использовать, смотрите в документации на gnuplot.

Ну и наконец, давайте кроме синуса на графике ещё нарисуем и косинус, да ещё и зададим тип линии и её цвет. А так же добавим легенды, что же мы чертим.

```
plot sin(x) title "sinux" lc rgb "red", cos(x) title "cosinus" lc rgb "green"
```

Здесь мы рисуем два графика на одном холсте, красным и зелёным цветом. Вообще вариантов линий (пунктир, штрих, сплошная), тощин линий, цветов великое множество. Как и типов точек. Моя цель лишь продемонстрировать спектр возможностей. Сведём все команды в одну кучку и выполним их последовательно. Чтобы сбросить предыдущие настройки, введём `reset`.

```
reset
set xlabel "X"
set ylabel "Y"
set grid
set yrange [-1.1:1.1]
set xrange[-pi:pi]
set title "Gnuplot for habr" font "Helvetica Bold, 20"
plot sin(x) title "sinux" lc rgb "red", cos(x) title "cosinus" lc rgb "green"
```

В результате получаем вот такую красоту.

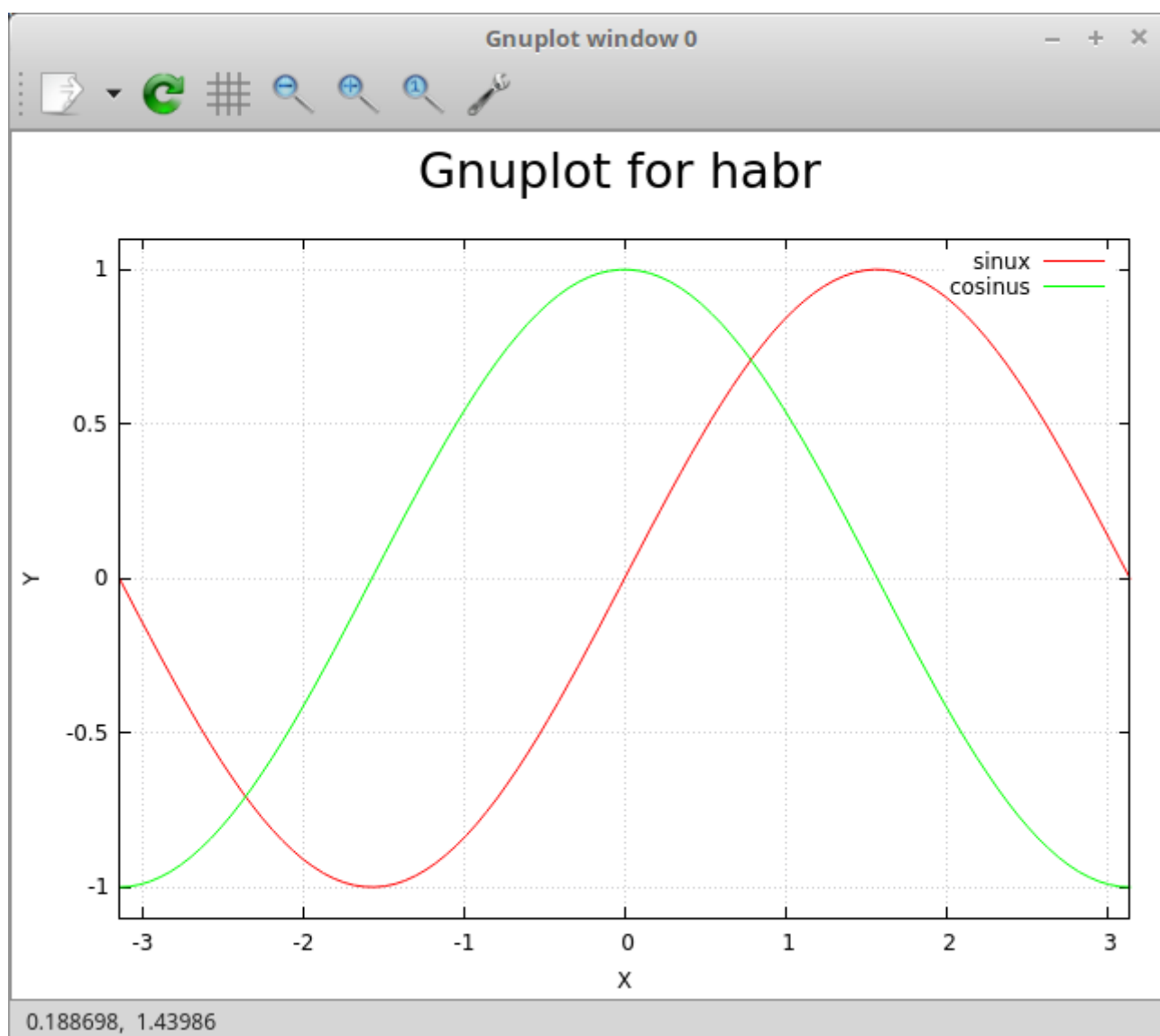


График уже не стыдно публиковать в научном журнале.

Если вы повторяли честно всё это за мной, то могли заметить, что вручную каждый раз вводить это, даже копируя, как-то не комильфо. Но это же готовый скрипт? Так давайте же его и сделаем!

Выходим из командного режима, командой `exit` и создаём файл:

```
vim testsin.gpi
```

```
#!/usr/bin/gnuplot -persist
set xlabel "X"
set ylabel "Y"
set grid
set yrange [-1.1:1.1]
set xrange[-pi:pi]
set title "Gnuplot for habr" font "Helvetica Bold, 20"
plot sin(x) title "sinux" lc rgb "red", cos(x) title "cosinus" lc rgb "green"
```

Делаем его исполняемым и запускаем. ~~Ну не забываем как выйти из vim.~~

```
chmod +x testsin.gpi
./testsin.gpi
```

В результате получаем такое же окно с графиками. Если в заголовке не добавить “-persist”, то окно автоматически закроется после выполнения скрипта.

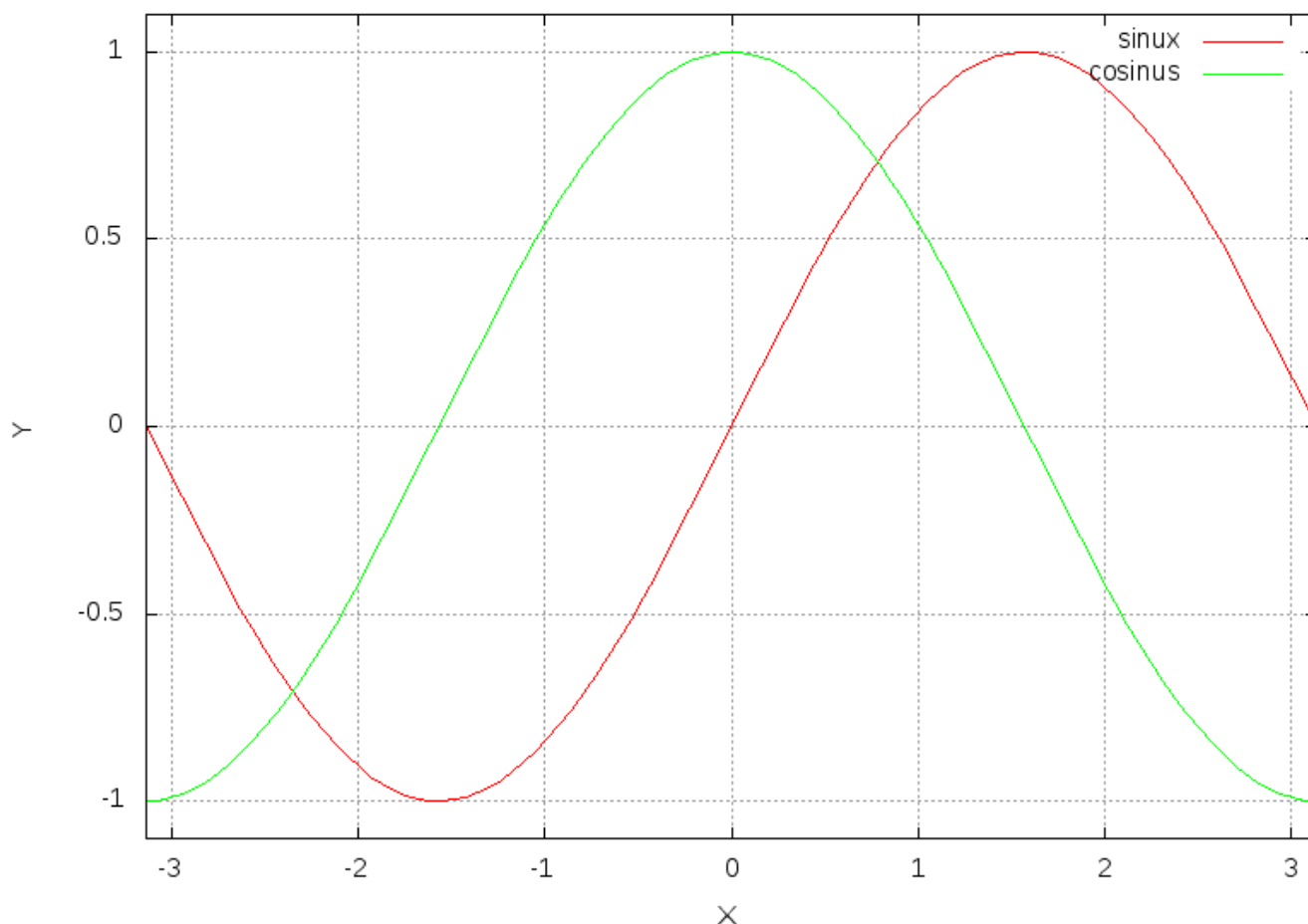
Но окно создавать часто не очень нужно, плюс не всегда удобно им пользоваться, да и вы можете работать в операционной системе без GUI. Значительно чаще нужно получать графические файлы. Лично я предпочитаю векторный формат `postscript`, так как при большом количестве точек можно приближать различные участки графика, без потери качества. И так же просмотрщик в линуксе автоматически обновляет окно с графиком, при изменении файла `postscript`, что тоже весьма удобно.

Для того, чтобы вывести данные в файл, а не на экран, надо переназначить терминал.

```
set terminal png size 800, 600
set output "result.png"
```

Как не трудно догадаться, указываем тип файла, его разрешение, затем указываем имя файла. Добавляем эти две строки в начало нашего скрипта, и получаем эту картинку в текущей папке.

Gnuplot for habr



Для того, чтобы сохранять в postscript нужно использовать следующие команды:

```
set terminal postscript eps enhanced color solid
set output "result.ps"
```

Реальные данные

Синусы, косинусы рисовать конечно прикольно, но всё же реальные данные рисовать намного интереснее! Напомню недавнюю задачу, о которой писал статью – вывод графиков скорости интернета за длительные промежутки времени. Формат данных там был следующий.

```
Operator; #Test; Date; Time; Coordinates; Download Mb/s; Upload Mb/s; ping;
Testserver
Rostelecom;0;21.05.2020;09:56:00;NA,
NA;3.7877656948451692;5.231226008184113;132.227;MaximaTelecom (Moscow) [0.12 km]:
132.227 ms
Rostelecom;1;21.05.2020;10:01:02;NA,
```

```
NA;5.274994541394363;5.1088572634075815;127.52;MaximaTelecom (Moscow) [0.12 km]:  
127.52 ms  
Rostelecom;2;21.05.2020;10:04:35;NA,  
NA;3.61044819424076;4.624132180211938;135.456;MaximaTelecom (Moscow) [0.12 km]:  
135.456 ms
```

Видно, что разделителем у нас служит точка с запятой, нам нужно вывести скорость загрузки, скорость выгрузки в зависимости от времени. При этом, если обратить внимание дата и время находятся в разных столбцах. Сейчас расскажу, как я это обошёл. Сразу приведу скрипт, по которому строился график.

```
#!/usr/bin/gnuplot -persist  
set terminal postscript eps enhanced color solid  
set output "Rostelecom.ps"  
  
#set terminal png size 1024, 768  
#set output "Rostelecom.png"  
  
set datafile separator ';'   
set grid xtics ytics  
set xdata time  
set timefmt '%d.%m.%Y;%H:%M:%S'  
  
set ylabel "Speed Mb/s"  
set xlabel 'Time'  
set title "Rostelecom Speed"  
  
plot "Rostelecom.csv" using 3:6 with lines title "Download", '' using 3:7 with l:  
  
set title "Rostelecom 2 Ping"  
set ylabel "Ping ms"  
plot "Rostelecom.csv" using 3:8 with lines title "Ping"
```

В начале файла мы задаём выходной файл postscript (либо png, если будет нужен).

set datafile separator ';' – мы задаём символ разделитель. По умолчанию столбцы разделяет пробел, но csv-файл предлагает множество вариантов разделителей, и нужно уметь использовать все.

set grid xtics ytics – устанавливаем сетку (можно сетку установить только по одной оси).

set xdata time – это важный момент, мы говорим о том, что по оси X формат данных

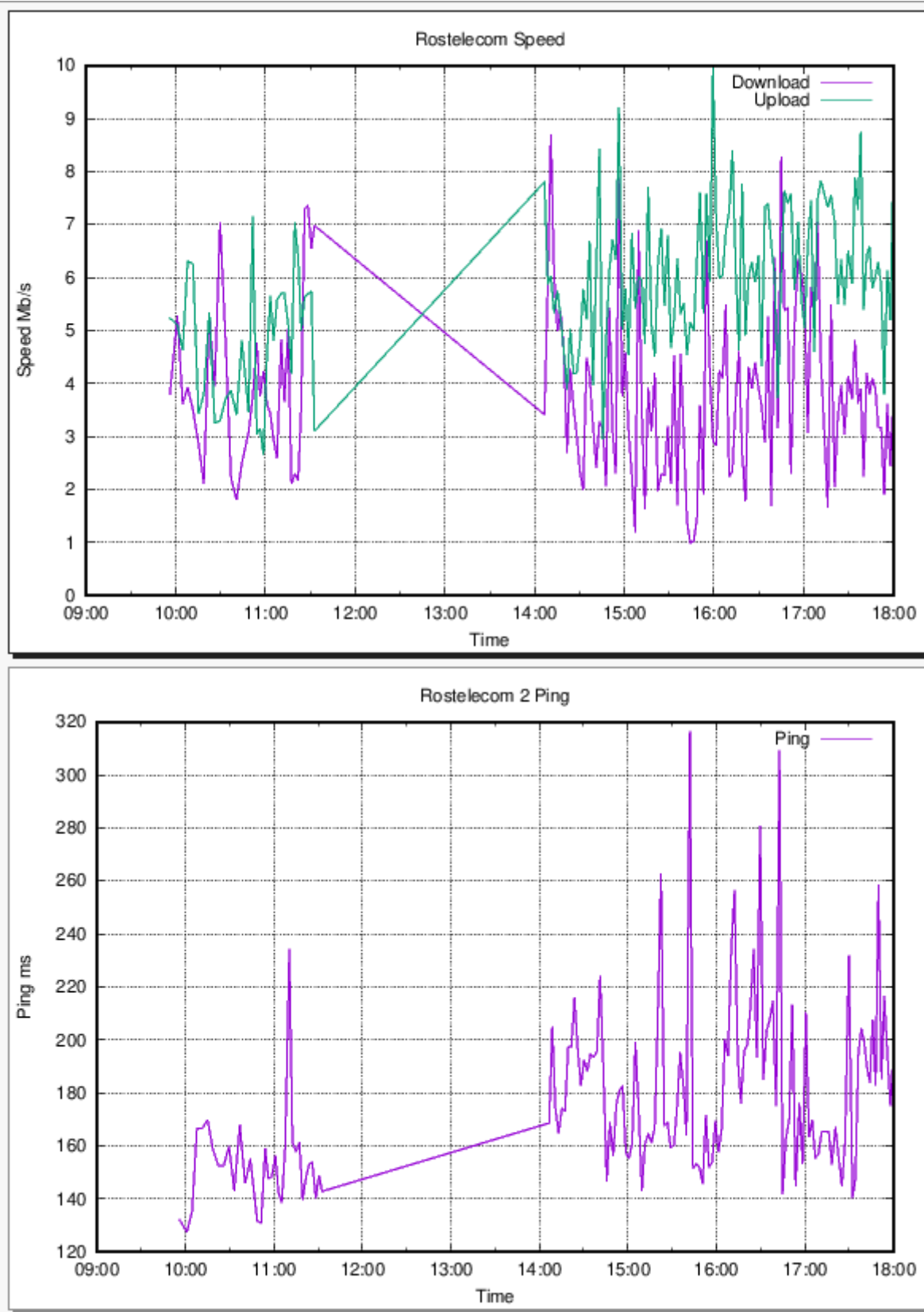
будет время.

`set timefmt '%d.%m.%Y;%H:%M:%S'` – задаём формат данных времени. Обратите внимание, что в формат времени попал символ разделения столбцов(";"), таким образом мы обработаем две колонки как одну.

Задаём подписи осей и графика. После чего строим график.

`plot «Rostelecom.csv» using 3:6 with lines title «Download», '' using 3:7 with lines title «Upload»` – на одном графике мы строим как скорость скачивания, так и скорость отдачи. `Using 3:6` – это номер столбца в нашем файле исходных данных, что от чего строим (X:Y).

Далее так же точно строим график для пинга. Результирующий график будет выглядеть следующим образом.



Это скриншот с postscript. Прямые линии в графике связаны с тем, что там пропуски данных. Вот вполне реальный пример построения графика.

А 3D???

Вы хотите 3D? Их есть у меня!

Долго думал, какой же пример трёхмерного графика привести, и не придумал ничего лучше, чем визуализировать картинку. Ведь по сути картинка – это трёхмерный график, где яркость пикселя – это координата по z . Поэтому давайте немного похулиганим. Возьмём самое знаменитое фото Эйнштейна.



И сделаем из него график. Для этого конвертнём его в формат pgm ASCII и выкинем все пробелы, заменив переводом строки, такой простой командой.

```
convert Einstein.jpg -compress none pgm:- | tr -s '\012\015' ' ' | tr -s ' ' '\n'
```

Кто не понял, что здесь происходит, поясню: мы конвертируем с помощью `imagemagic` картинку в формат `pgm`, а потом с помощью `tr` заменяем перевод каретки с переносом на новую строку на пробел, а потом все пробелы на перенос каретки и сохраняем это всё в `outfile.pgm`. Кому это сложно, могут открыть файл в `gimp` и экспортировать его как `pgm-ASCII`.

После чего открываем получившийся файл нашим любимым редактором ~~vim~~ и удаляем у него заголовок. В моём случае это первые три строки. Из заголовка не забываем узнать разрешение файла, в данном случае было 325x408 пикселей. Всё, мы получили текстовый файл координат Z ! Теперь наша задача добавить координаты X и Y , для этого прогоним всё это через питоновский скрипт.

```
f = open('outfile.pgm')
for x in range(408):
    for y in range(325):
        line = f.readline()
        print('%d %d %s' % (x, y, line)),
f.close()
```

Сохраняем это как `convert.py` и запускаем:

```
python convert.py > res.txt
```

Всё, у нас теперь `res.txt` содержит координаты Эйнштейна... Хм, ну точнее сказать координаты его изображения. Ну в общем, вы поняли :).

```
...
406 317 60
406 318 54
406 319 30
406 320 41
406 321 84
406 322 101
406 323 112
406 324 119
407 0 128
407 1 53
407 2 89
407 3 95
407 4 87
...
```

Пример файла.

Скрипт для построения этой красоты выглядит следующим образом.

```
#!/usr/bin/gnuplot -persist
#set terminal png size 1024, 768
#set output "result.png"
#set grid xtics ytics
```

```
#set terminal postscript eps enhanced color solid
#set output "result.ps"

set title "Albert Einstein"
set palette gray
set hidden3d
set pm3d at bs
set dgrid3d 100,100 qnorm 2
set xlabel "X" font "Helvetica Bold ,18"
set ylabel "Y" rotate by 90 font "Helvetica Bold ,18"
set zlabel "Z" font "Helvetica Bold ,18"

set xrange [0:408]
set yrange [0:325]
set zrange [-256:256]
unset key
splot "./res.txt" with l
```

Прежде чем мы пойдём разбирать скрипт, если вы будете это повторять, то настоятельно рекомендую выполнить строки скрипта в командном режиме, чтобы можно было мышкой вращать график (разумеется не указывая `set terminal`). Это очень круто!

Вначале мы устанавливаем тип выходных данных, а так же границы данных. Границы выставлены по размерам картинки, и плюс от низа я отступил по оси Z на 256 символов, чтобы была видна проекция картинки. Дальше мы озаглавливаем график, подписываем оси. Командой `unset key` – я отключаю легенды (она не нужна на графике). А вот далее идёт настоящая магия!

set palette gray – мы задаём палитру. Если оставить по умолчанию, то график будет цветным, как на тепловизоре. Чем выше, тем более жёлтое пятно, чем ниже тем темнее красный цвет.

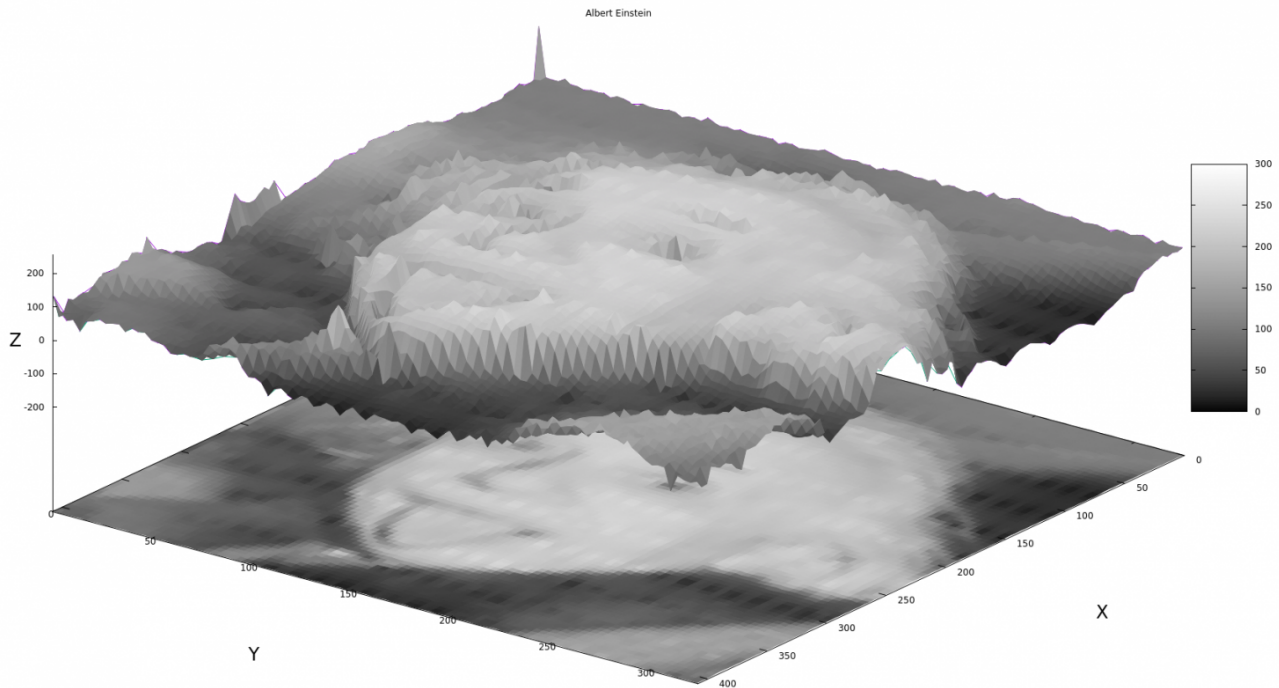
set hidden3d – как бы натягивает изогнутую поверхность (удаляет линии), таким образом формируется красивая выпуклая поверхность.

set pm3d at bs – включаем стиль рисования трёхмерных данных, который рисует данные с координатой сеткой и цветом. Подробнее читайте в документации, более детальное описание выходит за рамки статьи.

set dgrid3d 100,100 qnorm 2 – устанавливаем размер ячеек сетки 100x100, и сглаживание между ячейками. Значение 100x100 и так очень большое, и программа сильно тормозит. `qnorm 2` – это сглаживание (интерполяция данных между ячейками).

`splot "./res.txt" with l` – рисуем получившийся график. «With l» – означает рисовать график линиями. Это мелкий хак, потому что точки видны на графике (можно задать маленькие точки).

После запуска ждём некоторое время, и получаем полигональный «барельеф». Попробуйте поэкспериментировать с настройками, чтобы получить другие варианты визуализации.



Изображение в командном режиме, после того как поворачали.

Тут вспоминается сразу анекдот.

Как найти площадь Ленина?

Только необразованный человек ответит на этот вопрос, что нужно высоту Ленина умножить на ширину Ленина.

Образованный человек знает, что нужно взять интеграл по поверхности.

Применение gnuplot в своих программах

Пример взят со [stackoverflow](#) с моими небольшими доработками.

Код генерирует текстовый файл и постоянно вызывает перестроение графика. Код приложу под спойлер, чтобы не рвать статью.

► [Пример кода на Си использующего gnuplot](#)

Код работает очень просто, мы открываем pipe:

```
gp = popen("gnuplot -", "w");
```

Это аналогично вертикальной черте в bash, когда мы за одной командой пишем другую, только внутри программы. Пишем данные в файл `lorenzgplot.dat`. Один раз вызываем в `gnuplot` команду `splot`:

```
fprintf(gp, "splot './lorenzgplot.dat' with lines \n");
```

И далее при добавлении новой точки, мы перестраиваем график.

```
fprintf(gp, "replot \n");
```

В результате получаем очень красивое медленное построение Аттрактора Лоренца. Ниже видео, снятое почти десять лет назад, на старенький фотоаппарат, поэтому не ругайтесь сильно. Важно в видео другое, что всё это прекрасно работает на таком старом железе, как Nokia N800. Смотреть это желательно без звука.

Gnuplot on Nokia N800, real-time plot



Важно понимать, что команда **replot** очень хорошо кушает память и процессорное время, то есть, вот такое построение графика некило так подтормаживает систему. Так что, при всей любви к `gnuplot` – это не лучший способ его использования. Ещё одна проблема, что данное окно не удастся ни закрыть, ни передвинуть.

Заключение

Напоследок хочу показать видео, в котором я собрал сотни графиков случайного логарифмического распределения регистраций радиоактивных частиц, реальные данные одного исследования. Видео можно и нужно смотреть со звуком.

God plays dice



В этой статье не смог рассказать и тысячной доли возможностей данного графопостроителя, разве что немного ознакомил читателя с данной программой. Далее вам следует самостоятельно искать примеры, читать документацию на официальном сайте gnuplot.sourceforge.net либо www.gnuplot.info. Обязательно загляните в примеры, там очень много интересного и полезного.

Для старта так же могу порекомендовать [Краткое введение в gnuplot \(рус\)](#). Искренне удивлён, что такая замечательная программа не изучается во всех технических ВУЗах наравне с Latex. У нас зачем-то учили MS Excel и Word.

Изучить gnuplot не сложно, я потратил буквально несколько дней в попытке разобраться с нуля. Но с данной статьёй, верю, что у вас всё будет быстрее. Теперь я забыл о всяких Excel/Calc в качестве графопостроителей, использую только гнуплот. Тем более, что я даже не знаю и десятой доли всех возможностей построения графиков.

Хочу отметить, что существуют множество других графопостроителей, не хуже, чем гнуплот, тем более, что он достаточно старый. Но для меня gnuplot оказался наиболее простым и исчерпывающим. Плюс он самый распространённый графопостроитель, и в сети громадное количество примеров его использования. Спасибо что дочитали!

Habrahabr10

Промокод для скидки 10% на виртуальные серверы RUVDS


Теги: `gnuplot`, графики, построение графиков, визуализация данных

Хабы: Блог компании RUVDS.com, Визуализация данных, Лайфхаки для гиков

Редакторский дайджест


Присылаем лучшие статьи раз в месяц

Электронпочта

 RUVDS.com

VDS/VPS-хостинг. Скидка 15% по коду **HABR15**

Telegram ВКонтакте Twitter



758 126.6

Карма Рейтинг

Сергей @dlinyj


Автор-фрилансер

Задонатить




 Комментарии 26




Публикации

ЛУЧШИЕ ЗА СУТКИ ПОХОЖИЕ

 **averkij**
18 часов назад

Это не чат, это GigaChat. Русскоязычная ChatGPT от Сбера

 Средний  14 мин  33K

 +145  126  148



maria4kova1

16 часов назад

Как «Сбер» воспользовался студенческим стартапом и даже не сказал его создателям спасибо



Простой



7 мин



23K

Из песочницы



+141



44



61



Antikiller

14 часов назад

Прекратите звать (не) меня на работу



Простой



3 мин



14K

Кейс



+54



22



57



ru_vds

14 часов назад

Новый способ угона автомобилей без ключа: инъецирование через CAN



Средний



19 мин



5.8K

Кейс

Перевод



+36



28



37



alizar

18 часов назад

Математический подход к выбору девушки*. Правило 37%



Простой



7 мин



9.1K



+31



62



34

Показать еще

ИНФОРМАЦИЯ

Сайт

ruvds.com

Дата регистрации	18 марта 2016
Дата основания	27 июля 2015
Численность	11–30 человек
Местоположение	Россия
Представитель	ruvds

ССЫЛКИ

VPS / VDS сервер от 130 рублей в месяц.

ruvds.com

Дата-центры RUVDS в Москве, Санкт-Петербурге, Казани, Екатеринбурге, Новосибирске, Лондоне, Франкфурте, Цюрихе, Амстердаме

ruvds.com

Помощь и вопросы

ruvds.com

Партнерская программа RUVDS

ruvds.com

VPS (CPU 1x2ГГц, RAM 512Mb, SSD 10 Gb) – 190 рублей в месяц

ruvds.com

VPS Windows от 523 рублей в месяц. Бесплатный тестовый период 3 дня.

ruvds.com

VDS в Цюрихе. Дата-центр TIER III – швейцарское качество по низкой цене.

ruvds.com

Антивирусная защита виртуального сервера. Легкий агент для VPS.

ruvds.com

VPS в Лондоне. Дата-центр TIER III – английская точность за рубли.

ruvds.com

VPS с видеокартой на мощных серверах 3,4ГГц

ruvds.com

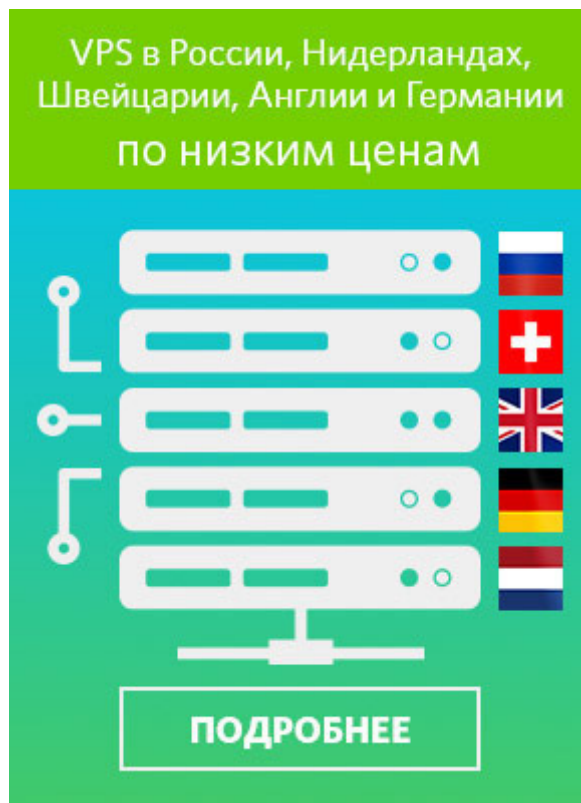
ПРИЛОЖЕНИЯ



RUVDS Client

Приложение для мониторинга и управления виртуальными серверами RUVDS с мобильных устройств.

ВИДЖЕТ



БЛОГ НА ХАБРЕ

14 часов назад

Новый способ угона автомобилей без ключа: инъецирование через CAN

5.7K 37

18 часов назад

Математический подход к выбору девушки*. Правило 37%

9.1K 34

23 апр в 21:00

Halo: Combat Evolved — отец современных шутеров

3.1K 26

23 апр в 17:00

Как передовые технологии порождают религиозные культы и эзотерическое мышление

 6K  15

22 апр в 21:00

Триггер — электронная ячейка памяти

 3.4K  11

Ваш аккаунт

Разделы

Информация

Услуги

Войти

Статьи

Устройство сайта

Корпоративный блог

Регистрация

Новости

Для авторов

Медийная реклама

Хабы

Для компаний

Нативные проекты

Компании

Документы

Образовательные

Авторы

Соглашение

программы

Песочница

Конфиденциальность

Стартапам

Спецпроекты



Настройка языка

Техническая поддержка

Вернуться на старую версию

© 2006–2023, Habr