

🏠 (<http://dkhramov.dp.ua>)
/ [comp](http://dkhramov.dp.ua/category/comp) (<http://dkhramov.dp.ua/category/comp>)
/ Вывод данных из программ на языках C/C++ в gnuplot

Вывод данных из программ на языках C/C++ в gnuplot

(<http://dkhramov.dp.ua/Comp.GnuplotOutFromC>)

Date 📅 Пт 07 Март 2014 Редакция 📅 Пт 26 Август 2016
Категория [comp](http://dkhramov.dp.ua/category/comp) (<http://dkhramov.dp.ua/category/comp>) Теги
[gnuplot](http://dkhramov.dp.ua/tag/gnuplot) (<http://dkhramov.dp.ua/tag/gnuplot>) / [C](http://dkhramov.dp.ua/tag/c) (<http://dkhramov.dp.ua/tag/c>) / [C++](http://dkhramov.dp.ua/tag/cpp) (<http://dkhramov.dp.ua/tag/cpp>) / [linux](http://dkhramov.dp.ua/tag/linux) (<http://dkhramov.dp.ua/tag/linux>) / [графика](http://dkhramov.dp.ua/tag/grafika) (<http://dkhramov.dp.ua/tag/grafika>)

Сохранение данных в файл

Данные из программы, например,

Дмитрий Храмов
Компьютерное моделирование и все, что с ним связано: сбор данных, их анализ, разработка математических моделей, софт для моделирования, визуализации и оформления публикаций. Ну и за жизнь немного.

🏠 Категории

(<http://dkhramov.dp.ua/>)

📁 [comp](http://dkhramov.dp.ua/category/comp) (<http://dkhramov.dp.ua/category/comp>)

📁 [edu](http://dkhramov.dp.ua/category/edu) (<http://dkhramov.dp.ua/category/edu>)

📁 [misc](http://dkhramov.dp.ua/category/misc) (<http://dkhramov.dp.ua/category/misc>)

📁 [sci](http://dkhramov.dp.ua/category/sci) (<http://dkhramov.dp.ua/category/sci>)

```
#include <stdio.h>
#include <math.h>

const double pi = 3.14;

int main()
{
    const double npoints = 200;
    const double step = 4 * pi / npoints;

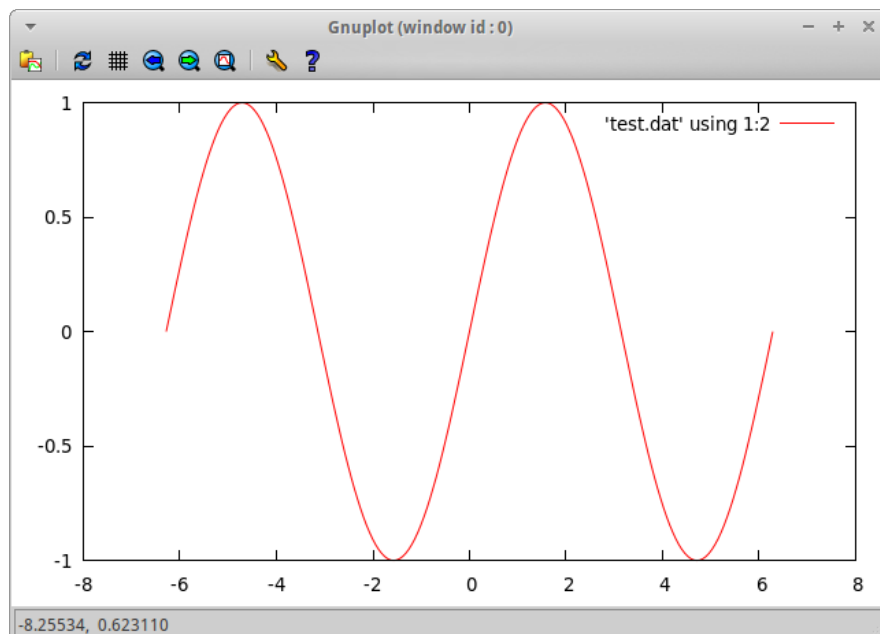
    FILE* fp = fopen("test.dat", "w");

    for (int i = 0; i < npoints+1; i++)
    {
        double x = -2 * pi + i * step;
        double y = sin(x);
        fprintf(fp, "%f\t%f\n", x, y);
    }

    fclose(fp);
}
```

сохраняются в файле (test.dat). Затем в gnuplot по ним строится график

```
gnuplot> plot 'test.dat' using 1:2 with lines
```



Подробнее визуализация данных из файла с помощью gnuplot описана [здесь \(Sci.Gnuplot2DGraphsFromDatafile.html\)](http://Sci.Gnuplot2DGraphsFromDatafile.html).

Теги

(<http://dkhramov.dp.ua/>)

агентное моделирование (<http://dkhramov.dp.ua/tag/agentnoe-modelirovanie>)

API (<http://dkhramov.dp.ua/tag/api>)

arch (<http://dkhramov.dp.ua/tag/arch>)

астрономия (<http://dkhramov.dp.ua/tag/astronomiia>)

баллистика (<http://dkhramov.dp.ua/tag/ballistika>)

базы данных (<http://dkhramov.dp.ua/tag/bazy-dannykh>)

beamer (<http://dkhramov.dp.ua/tag/beamer>)

биатлон (<http://dkhramov.dp.ua/tag/biatlon>)

bibtex (<http://dkhramov.dp.ua/tag/bibtex>)

box2d (<http://dkhramov.dp.ua/tag/box2d>)

bullet (<http://dkhramov.dp.ua/tag/bullet>)

C (<http://dkhramov.dp.ua/tag/c>)

caret (<http://dkhramov.dp.ua/tag/caret>)

cas (<http://dkhramov.dp.ua/tag/cas>)

чтение (<http://dkhramov.dp.ua/tag/chtenie>)

CodeBlocks (<http://dkhramov.dp.ua/tag/codeblocks>)

C++ (<http://dkhramov.dp.ua/tag/cpp>)

css (<http://dkhramov.dp.ua/tag/css>)

cubesat (<http://dkhramov.dp.ua/tag/cubesat>)

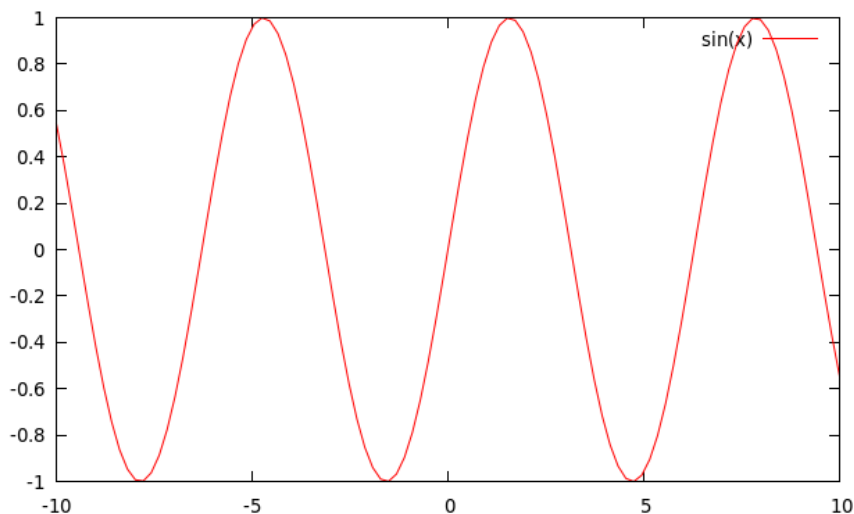
дамы (<http://dkhramov.dp.ua/tag/damy>)

Использование каналов

Каналы в Unix/Linux позволяют перенаправить вывод команд и программ на вход gnuplot. В простейшем случае, с помощью команды `echo` можно передать в gnuplot ее же собственную команду. Например:

```
echo "plot sin(x)" | gnuplot -persist
```

выводит график синусоиды



Опция `persist` нужна, чтобы задержать окно gnuplot на экране после окончания построения графика.

gnuplot позволяет строить графики на основе данных, вводимых в командной строке

```
plot '-' using ... with ...  
x1 y1  
x2 y2  
x3 y3  
...  
e
```

Псевдофайл данных обозначается как `'-'`; символ `e` обозначает конец файла (EOF).

Теперь сформируем команду gnuplot, строящую график по данным из псевдофайла, и получим сами данные

Днепропетровск (<http://dkhramov.dp.ua/tag/dnepropetrovsk>)

дом (<http://dkhramov.dp.ua/tag/dm>)

doxygen (<http://dkhramov.dp.ua/tag/doxygen>)

Екатеринослав (<http://dkhramov.dp.ua/tag/ekaterinoslav>)

Excel (<http://dkhramov.dp.ua/tag/excel>)

физические движки (<http://dkhramov.dp.ua/tag/fizicheskie-dvizhki>)

формулы (<http://dkhramov.dp.ua/tag/formuly>)

fortran (<http://dkhramov.dp.ua/tag/fortran>)

foss (<http://dkhramov.dp.ua/tag/foss>)

фото (<http://dkhramov.dp.ua/tag/photo>)

gdb (<http://dkhramov.dp.ua/tag/gdb>)

ГИС (<http://dkhramov.dp.ua/tag/gis>)

GLUT (<http://dkhramov.dp.ua/tag/glut>)

gnuplot (<http://dkhramov.dp.ua/tag/gnuplot>)

Google (<http://dkhramov.dp.ua/tag/google>)

графика (<http://dkhramov.dp.ua/tag/grafika>)

HTML (<http://dkhramov.dp.ua/tag/html>)

InSAR (<http://dkhramov.dp.ua/tag/insar>)

история (<http://dkhramov.dp.ua/tag/istoriya>)

```
#include <stdio.h>
#include <math.h>

const double pi = 3.14;

int main()
{
    const double npoints = 200;
    const double step = 4 * pi / npoints;

    printf("%s\n", "plot '-' using 1:2 with lines"); // Рис
                                                    // из
    for (int i = 0; i < npoints+1; i++)           // Пол
    {
        double x = -2 * pi + i * step;
        double y = sin(x);
        printf("%f\t%f\n", x, y);
    }

    printf("%s\n", "e"); // За
}
```

Направим результат работы программы на вход gnuplot

```
./tttest | gnuplot -persist
```

и получим уже знакомый по первому рисунку результат.

Запуск процесса gnuplot из программы на C/C++

В предыдущем примере мы использовали неименованный канал, теперь создадим канал именованный. Начнем с примера, работающего в Unix/Linux:

юмор (<http://dkhramov.dp.ua/tag/iumor>)

календарь (<http://dkhramov.dp.ua/tag/kalendar>)

картины (<http://dkhramov.dp.ua/tag/kartiny>)

карты (<http://dkhramov.dp.ua/tag/karty>)

Космические тросовые системы (<http://dkhramov.dp.ua/tag/kosmicheskie-trosovye-sistemy>)

космос (<http://dkhramov.dp.ua/tag/kosmos>)

K&R (<http://dkhramov.dp.ua/tag/k&r>)

кухня (<http://dkhramov.dp.ua/tag/kukhnya>)

курсы (<http://dkhramov.dp.ua/tag/kursy>)

LaTeX (<http://dkhramov.dp.ua/tag/latex>)

linux (<http://dkhramov.dp.ua/tag/linux>)

maple (<http://dkhramov.dp.ua/tag/maple>)

Markdown (<http://dkhramov.dp.ua/tag/markdown>)

Matlab (<http://dkhramov.dp.ua/tag/matlab>)

modelica (<http://dkhramov.dp.ua/tag/modelica>)

научный софт (<http://dkhramov.dp.ua/tag/nauchnyi-soft>)

netlogo (<http://dkhramov.dp.ua/tag/netlogo>)

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    FILE *gp = popen("gnuplot -persist", "w"); // gp - дескриптор

    if (gp == NULL)
    {
        printf("Error opening pipe to GNU plot.\n");
        exit(0);
    }

    fprintf(gp, "plot sin(x)\n");
    pclose(gp);

    return 0;
}
```

Любопытно, что используя при компиляции стандарт `-c99`, выдается предупреждение о неявной декларации `popen` и `pclose`.

Процесс в Unix/Linux открывается функцией `popen`. В результате будет создан канал (`gp`), запись в который выполняется при помощи `fprintf`. Закрывается процесс функцией `pclose`.

Рассмотрим кроссплатформенный пример создания процесса `gnuplot`. Для каждой операционной системы используются свои функции открытия/закрытия процесса:

Office (<http://dkhramov.dp.ua/tag/office>)

OpenGL (<http://dkhramov.dp.ua/tag/opengl>)

отчет о НИР (<http://dkhramov.dp.ua/tag/otchet-o-nir>)

открытки (<http://dkhramov.dp.ua/tag/otkrytki>)

открытые данные (<http://dkhramov.dp.ua/tag/otkrytye-dannye>)

pascal (<http://dkhramov.dp.ua/tag/pascal>)

pdf (<http://dkhramov.dp.ua/tag/pdf>)

Pelican (<http://dkhramov.dp.ua/tag/pelican>)

полезности (<http://dkhramov.dp.ua/tag/poleznosti>)

pov-ray (<http://dkhramov.dp.ua/tag/pov-ray>)

программирование (<http://dkhramov.dp.ua/tag/programirovanie>)

python (<http://dkhramov.dp.ua/tag/python>)

Qt (<http://dkhramov.dp.ua/tag/qt>)

R (<http://dkhramov.dp.ua/tag/r>)

рисунки (<http://dkhramov.dp.ua/tag/risunki>)

SAR (<http://dkhramov.dp.ua/tag/sar>)

scilab (<http://dkhramov.dp.ua/tag/scilab>)

selenium (<http://dkhramov.dp.ua/tag/selenium>)

Shiny (<http://dkhramov.dp.ua/tag/shiny>)

школа (<http://dkhramov.dp.ua/tag/shkola>)

```

#include <iostream>
#include <cstdio>

// Для Windows нужно указать путь к исполняемому файлу gnup
// например: "c:Program Files (x86)gnuplotbinpgnuplot.exe"
#ifdef WIN32
    #define GNUPLOT_NAME "pgnuplot -persist"
#else
    #define GNUPLOT_NAME "gnuplot -persist"
#endif

int main()
{
    #ifdef WIN32
        FILE *pipe = _popen(GNUPLOT_NAME, "w");
    #else
        FILE *pipe = popen(GNUPLOT_NAME, "w");
    #endif

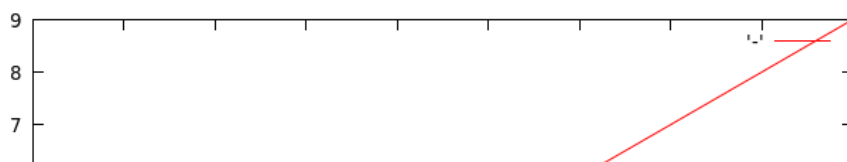
    if (pipe != NULL)
    {
        fprintf(pipe, "plot '-' with lines\n");
        for(int i = 0; i < 10; i++)
            fprintf(pipe, "%d\n", i);
        fprintf(pipe, "%s\n", "e");
        fflush(pipe);

        // ожидание нажатия клавиши
        std::cin.clear();
        std::cin.ignore(std::cin.rdbuf()->in_avail());
        std::cin.get();

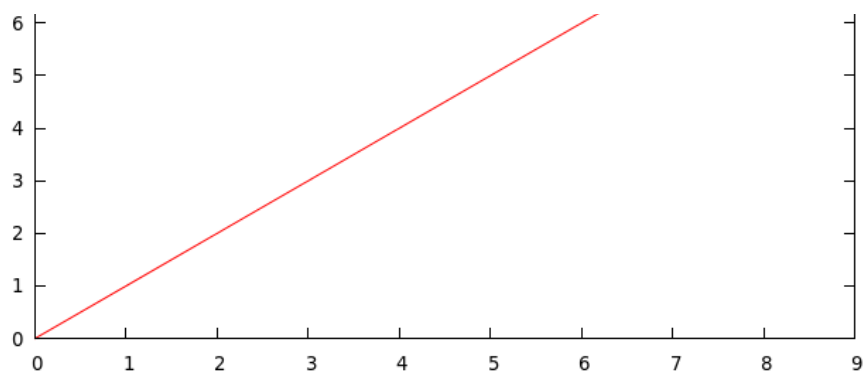
        #ifdef WIN32
            _pclose(pipe);
        #else
            pclose(pipe);
        #endif
    }
    else
        std::cout << "Could not open pipe" << std::endl;
    return 0;
}

```

Результат (наконец-то не синусоида!):



[шрифты \(http://dkhramov.dp.ua/tag/shrifty\)](http://dkhramov.dp.ua/tag/shrifty)
[simulink \(http://dkhramov.dp.ua/tag/simulink\)](http://dkhramov.dp.ua/tag/simulink)
[скрапинг \(http://dkhramov.dp.ua/tag/skrapping\)](http://dkhramov.dp.ua/tag/skrapping)
[skype \(http://dkhramov.dp.ua/tag/skype\)](http://dkhramov.dp.ua/tag/skype)
[SNAP \(http://dkhramov.dp.ua/tag/snap\)](http://dkhramov.dp.ua/tag/snap)
[спорт \(http://dkhramov.dp.ua/tag/sport\)](http://dkhramov.dp.ua/tag/sport)
[ссылки \(http://dkhramov.dp.ua/tag/ssylki\)](http://dkhramov.dp.ua/tag/ssylki)
[стихи \(http://dkhramov.dp.ua/tag/stikhi\)](http://dkhramov.dp.ua/tag/stikhi)
[таблицы \(http://dkhramov.dp.ua/tag/tablitsy\)](http://dkhramov.dp.ua/tag/tablitsy)
[TeX \(http://dkhramov.dp.ua/tag/tex\)](http://dkhramov.dp.ua/tag/tex)
[texmaker \(http://dkhramov.dp.ua/tag/texmaker\)](http://dkhramov.dp.ua/tag/texmaker)
[tikz \(http://dkhramov.dp.ua/tag/tikz\)](http://dkhramov.dp.ua/tag/tikz)
[ТВ \(http://dkhramov.dp.ua/tag/tv\)](http://dkhramov.dp.ua/tag/tv)
[ubuntu \(http://dkhramov.dp.ua/tag/ubuntu\)](http://dkhramov.dp.ua/tag/ubuntu)
[учебник \(http://dkhramov.dp.ua/tag/uchebnik\)](http://dkhramov.dp.ua/tag/uchebnik)
[веб-скрапинг \(http://dkhramov.dp.ua/tag/veb-skrapping\)](http://dkhramov.dp.ua/tag/veb-skrapping)
[верстка \(http://dkhramov.dp.ua/tag/verstka\)](http://dkhramov.dp.ua/tag/verstka)
[Visual Studio \(http://dkhramov.dp.ua/tag/visual-studio\)](http://dkhramov.dp.ua/tag/visual-studio)
[windows \(http://dkhramov.dp.ua/tag/windows\)](http://dkhramov.dp.ua/tag/windows)
[word \(http://dkhramov.dp.ua/tag/word\)](http://dkhramov.dp.ua/tag/word)
[xcos \(http://dkhramov.dp.ua/tag/xcos\)](http://dkhramov.dp.ua/tag/xcos)
[XeTeX \(http://dkhramov.dp.ua/tag/xetex\)](http://dkhramov.dp.ua/tag/xetex)



xpath (<http://dkhramov.dp.ua/tag/xpath>)
здоровье (<http://dkhramov.dp.ua/tag/zdorove>)

Класс-обертка для работы с gnuplot

Поместим подробности работы с gnuplot внутрь класса. Ничего нового мы здесь не делаем, а нюансы отмечены в комментариях.

```
// gnuplot.h
#ifndef _GNUPLOT_H_
#define _GNUPLOT_H_

#include <stdio>
#include <string>
#include <iostream>

#ifdef WIN32
    #define GNUPLOT_NAME "pgnuplot -persist"
#else
    #define GNUPLOT_NAME "gnuplot -persist"
#endif

using std::string;
using std::cerr;

class Gnuplot
{
public:
    Gnuplot() ;
    ~Gnuplot();
    void operator ()(const string & command); // отправить

protected:
    FILE *gnuplotpipe;
};

Gnuplot::Gnuplot()
{
    #ifdef WIN32
        gnuplotpipe = _popen(GNUPLOT_NAME, "w");
    #else
        gnuplotpipe = popen(GNUPLOT_NAME, "w");
    #endif

    if (!gnuplotpipe)
    {
        cerr << ("Gnuplot not found !");
    }
}

Gnuplot::~Gnuplot()
{
    fprintf(gnuplotpipe, "exit\n");

    #ifdef WIN32
        _pclose(gnuplotpipe);
    #else
```



```
        pclose(gnuplotpipe);
    #endif
}
void Gnuplot::operator()(const string & command)
{
    fprintf(gnuplotpipe,"%s\n",command.c_str());
    fflush(gnuplotpipe); //без fflush ничего рисоваться не
};

#endif // #ifndef _GNUPLOT_H_
```

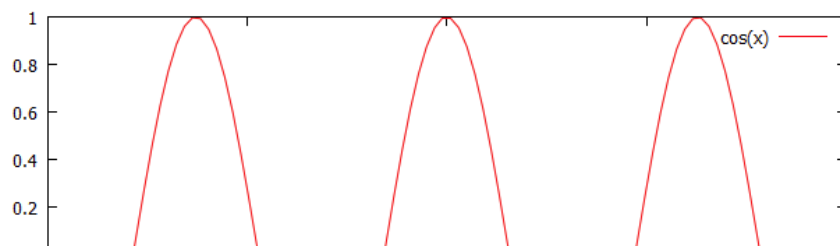
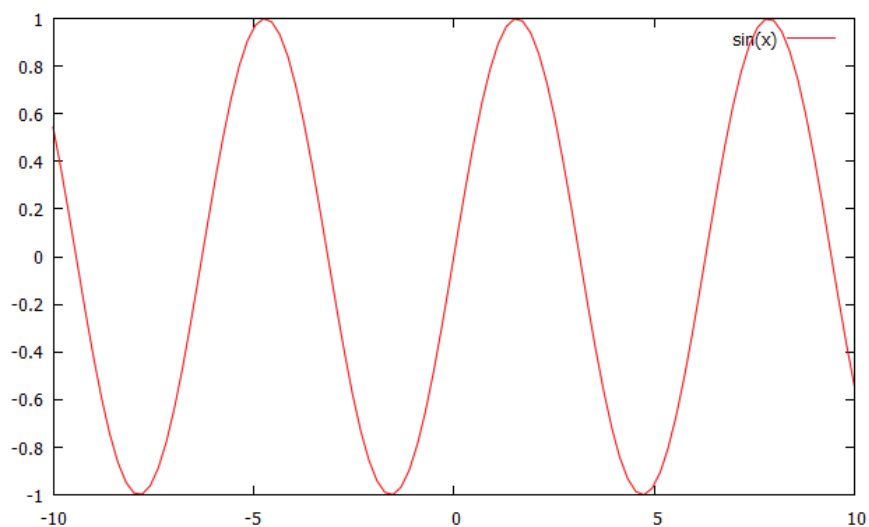
Пример использования созданного класса:

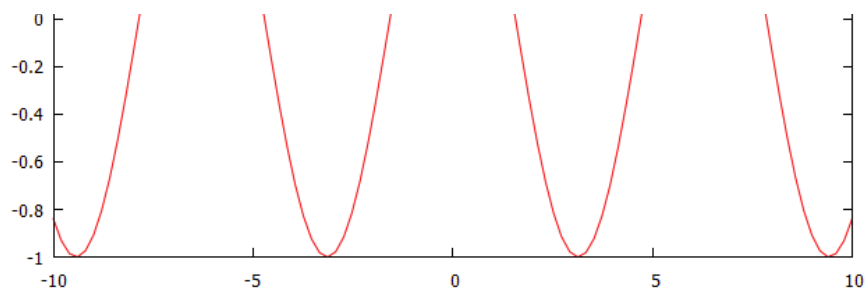
```
#include <iostream>
#include "gnuplot.h"

int main()
{
    Gnuplot plot;

    plot("plot sin(x)");
    std::cin.get();

    plot("plot cos(x)");
    std::cin.get();
}
```





Библиотеки

- [gnuplot interfaces in ANSI C \(http://ndevilla.free.fr/gnuplot/\)](http://ndevilla.free.fr/gnuplot/) — послужила основой для двух следующих библиотек.
- [gnuplot-cpp \(https://code.google.com/p/gnuplot-cpp/\)](https://code.google.com/p/gnuplot-cpp/): C++ интерфейс к gnuplot через POSIX-каналы.
- [C++ to gnuplot library \(http://sourceforge.net/projects/c2gnuplot/\)](http://sourceforge.net/projects/c2gnuplot/).

Читайте также

- Библиотеки для работы с разреженными матрицами (<http://dkhramov.dp.ua/Sci.SparseMatrixLibs>)
- Графика в LaTeX (<http://dkhramov.dp.ua/Comp.LatexGraphics>)
- gnuplot возвращается (<http://dkhramov.dp.ua/Sci.ReturnOfTheGnuplot>)
- gnuplot, который строит графики (<http://dkhramov.dp.ua/Sci.GnuplotPlotData>)
- Вернуться к настройкам по умолчанию (<http://dkhramov.dp.ua/Sci.GnuplotResetting>)

[Vkontakte](#)[Twitter](#)[Facebook](#)[E-mail](#)

Комментарии

0 Комментариев

dkhramov

 Maxim Bondarenko ▾ Рекомендовать Твитнуть Поделиться

Лучшее в начале ▾



Начать обсуждение...

Прокомментируйте первым.

 Подписаться Добавь Disqus на свой сайтДобавить DisqusДобавить

© 2009—2017 Дмитрий Храмов · При поддержке pelican-bootstrap3
(<https://github.com/getpelican/pelican-themes/tree/master/pelican-bootstrap3>),
Pelican (<http://docs.getpelican.com/>), Bootstrap (<http://getbootstrap.com>)

 Наверх