

Численные методы

Вычисления без программирования

М.М. Хапаев

Кафедра вычислительных методов

6 декабря 2015 г.

План действий

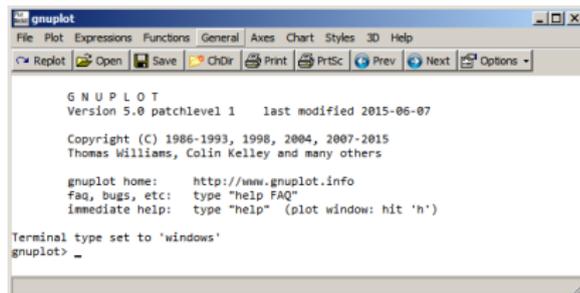
Планируется обсудить

- ▶ Как построить хороший график: **gnuplot**
- ▶ Как выполнить простые и не очень простые вычисления
 - ▶ **MatLab**
 - ▶ **Octave, Scilab**
 - ▶ **Python**
- ▶ Как написать правильно оформленный текст с формулами, графиками, ссылками и списком литературы: **L^AT_EX**

gnuplot

Строит всевозможные научные графики и помогает отлаживать программы

- ▶ Научные графики значительно отличаются от офисных
- ▶ Необходимо соответствовать правилам оформления научных публикаций
- ▶ Нужно строить много графиков
- ▶ Нужно средство отладки с графикой
- ▶ Приятно экономить свое время и силы
- ▶ **gnuplot** содержит много алгоритмов для работы с данными
- ▶ **gnuplot** всегда есть в Linux
- ▶ есть **gnuplot** для Windows:



```
gnuplot
File Plot Expressions Functions General Axes Chart Styles 3D Help
Replot Open Save ChDir Print PrtSc Prev Next Options
GNU PLOT
Version 5.0 patchlevel 1 last modified 2015-06-07

Copyright (C) 1986-1993, 1998, 2004, 2007-2015
Thomas Williams, Colin Kelley and many others

gnuplot home: http://www.gnuplot.info
faq, bugs, etc: type "help FAQ"
immediate help: type "help" (plot window: hit 'h')

Terminal type set to 'windows'
gnuplot> _
```

gnuplot это просто

Примеры

- ▶ 1D: `plot [-5*pi:5*pi] sin(x)/x with lines lt 1 lw 5, 1/x with lines lt 2 lw 1`
- ▶ 3D: `splot (x**2)*(y**2)`
- ▶ Data 1D: `plot "output.dat" using 1:2 with lines`
- ▶ <http://gnuplot.ikir.ru/>
- ▶ <http://folk.uio.no/hpl/scripting/doc/gnuplot/Kawano/index-e.html>
- ▶ Существует книга "Gnuplot in Action"

gnuplot

Еще пример

```
#Gnuplot file for plotting data.dat
reset
set terminal windows font "arial,12"
set xlabel 'r'
set ylabel 'Re[sin(teta(r,z=z_k))]'
set title 'Re[sin(teta(r,z=z_k))].dat'
set grid
plot 'infile.dat' using 1:2 w l title 'z=-0.01*dS)', \
     'infile.dat' using 1:3 w l title 'z=-0.1*dS)', \
     'infile.dat' using 1:4 w l title 'z=-0.2*dS)', \
     'infile.dat' using 1:5 w l title 'z=-0.3*dS)', \
     'infile.dat' using 1:6 w l title 'z=-0.5*dS)', \
     'infile.dat' using 1:7 w l title 'z=-1.0*dS)'
set terminal postscript eps enhanced color
set output 'this_graph.eps'
replot
```

MatLab

И его клоны

- ▶ Мы изучили некоторые численные методы линейной алгебры, математического анализа и дифференциальных уравнений
- ▶ Как минимизировать затраты на программирование наших знаний?
- ▶ Для вычислений без использования FORTRAN, C, C++ придумали интерпретаторы высокого уровня
- ▶ Популярен MatLab
 - ▶ Разработан давно специалистами по численному анализу
 - ▶ Живет и поддерживается
 - ▶ Его язык (совсем без программирования нельзя) стал стандартом
 - ▶ Файлы со скриптами для MatLab имеют расширение **.m**
 - ▶ Имеет недостаток, не бесплатен
- ▶ Имеются свободно доступные аналоги: **Octave** и **SciLab**

- ▶ Решаем систему линейных уравнений:

```
A=[ 2, 1; 1, 2 ];
```

```
b=[ 2; 1 ];
```

```
x = A\b
```

- ▶ Вычисляем интеграл (квадратуры Гаусса)

```
x=0.0
```

```
function y = f (x)
```

```
    y = x .* sin (1.0 ./ x) .* sqrt (abs (1.0 - x));
```

```
endfunction
```

```
s = linspace (0.00001, 3, 200)';
```

```
q=f(s);
```

```
plot (s, q)
```

```
[q, ier, nfun, err] = quad ("f", 0, 3)
```

- ▶ Все величины векторные
- ▶ Покомпонентные операции: `.*` , `./`

Octave

Решаем систему ДУ (многошаговым методом)

```
function xdot = f (x, t)
    r = 0.25;    k = 1.4;    a = 1.5;
    b = 0.16;   c = 0.9;    d = 0.8;
    xdot(1) = r*x(1)*(1 - x(1)/k) - a*x(1)*x(2)/(1 + b*x(1));
    xdot(2) = c*a*x(1)*x(2)/(1 + b*x(1)) - d*x(2);
endfunction
x0 = [1; 2];
t = linspace (0, 50, 200)';
x = lsode ("f", x0, t);
plot (t, x)
```

Octave: сценарий использования

Краевая задача для ДУ второго порядка с разрывами

$$u''(x) - \alpha^2 u(x) = 0, \quad -dS < x < 0 \quad (1)$$

$$k^2 u''(x) - \alpha^2 u(x) = 0, \quad 0 < x < dN \quad (2)$$

$$u'(-dS) = h, \quad k^2 u'(dN) = h, \quad - \text{краевые условия} \quad (3)$$

$$[u] \Big|_{x=0} = u(x) \Big|_{x=+0} - u(x) \Big|_{x=-0} = a, \quad \text{разрывное решение} \quad (4)$$

$$(k^2 - \beta) u'(x) \Big|_{x=+0} - u'(x) \Big|_{x=-0} = 0. \quad \text{разрыв производной} \quad (5)$$

Общее решение:

$$u = C_1 e^{-\alpha x} + C_2 e^{+\alpha x}, \quad x < 0 \quad (6)$$

$$u = C_3 e^{-\alpha x/k} + C_3 e^{+\alpha x/k}, \quad x > 0. \quad (7)$$

Нужно найти неизвестные константы и построить график.

Octave: сценарий использования

Скрипт

```
dS=1.0; dN=1.0;
alf=5; ja=5; bet=0.3; k=1.5; alf2=alf^2; k2=k^2;

a11=-exp(alf*dS); a12=exp(-alf*dS); a13=0.0; a14=0.0;
a21=0.0; a22=0.0; a23=-exp(-alf*dN/k); a24=exp(alf*dN/k);
a31=-1.0; a32=-1.0; a33=1.0; a34=1.0;
a41=alf; a42=-alf; a43=-(k2-bet)*alf/k; a44=(k2-bet)*alf/k;
A=[a11,a12,a13,a14;a21,a22,a23,a24;a31,a32,a33,a34;a41,a42,a43,a44];
b=[h/alf;h*k/alf;ja;0.0];
c = A \ b

N=200
xx=linspace(-dS,dN,N); f=linspace(-dS,dN,N);
for ia = 1:N
    x=xx(ia);
    if (x<0.0)
        f(ia)=c(1)*exp(-alf*x)+c(2)*exp(alf*x);
    else
        f(ia)=c(3)*exp(-alf*x/k)+c(4)*exp(alf*x/k);
    endif
endfor
plot(xx,f);
```

Python

Язык программирования, который нарушает много правил

- ▶ **Python** лучше загрузить сразу со всеми пакетами (anaconda)
- ▶ IPython – An enhanced Interactive Python.
- ▶ **Python** можно выучить за 10 минут:
<http://www.korokithakis.net/tutorials/python/>
- ▶ Существует незначительная разница между версиями
- ▶ **Python** применяется для разбора логов и файлов
- ▶ Существует огромное сообщество любителей **Python**
- ▶ Обычно программирование происходит с помощью готовых фрагментов кода
- ▶ <http://stackoverflow.com>

Пример использования Python

Решение краевой задачи

```
import numpy as np, matplotlib as mpl
import matplotlib.pyplot as plt

dS=1.0; dN=1.0;
h=1.0; alf=1.5;
ju=0.5; jdu=2.0;
bet=0.0; k=2.0;

alf2=alf*alf; k2=k*k;
ak=alf/k; p=ak*k2;

a11=-alf*np.exp(alf*dS)
a12=alf*np.exp(-alf*dS)
a13=a14=a21=a22=0.0
a23=-p*np.exp(-alf*dN/k)
a24=p*np.exp(alf*dN/k)
a31=a32=-1.0; a33=a34=1.0;
a41=alf; a42=-alf;
a43=-p; a44=p;

A=np.array( [ [a11,a12,a13,a14],
              [a21,a22,a23,a24],
              [a31,a32,a33,a34],
              [a41,a42,a43,a44] ] )
b=np.array( [[h], [h], [ju], [jdu]] )
c=np.linalg.solve(A,b)
print c

N=200
xx=np.linspace(-dS,dN,N)
f=np.linspace(-dS,dN,N)
for ia in range(0,N):
    x=xx[ia]
    if x<0.0:
        f[ia]=c[0]*np.exp(-alf*x)+
              c[1]*np.exp(alf*x)
    else:
        f[ia]=c[2]*np.exp(-alf*x/k)+
              c[3]*np.exp(alf*x/k)
plt.plot(xx,f)
plt.show()
```

Символьные вычисления в Python

<http://docs.sympy.org/dev/tutorial/>

```
>>> from sympy import *
>>> x = symbols('x')
>>> a = Integral(cos(x)*exp(x), x)
>>> Eq(a, a.doit())
Integral(exp(x)*cos(x), x) == exp(x)*sin(x)/2 +
                               exp(x)*cos(x)/2
```

- ▶ Установлено, что Word для подготовки научных текстов не годится
- ▶ L^AT_EX существует в Linux (Unix) и скачивается для Windows: www.miktex.org
- ▶ Можно L^AT_EX не устанавливать, а работать в облаке: <https://papeeria.com>, <https://www.overleaf.com/>
- ▶ Существует большое число руководств под условным названием "L^AT_EX для начинающих"
- ▶ L^AT_EX понятен и удобен для всех, владеющих каким-нибудь программированием
- ▶ Этот документ написан в L^AT_EX с использованием специального стиля для подготовки презентаций Beamer
- ▶ Поэтому рассмотрим текст этой презентации в соответствующем редакторе

Что мы не обсудили

- ▶ Электронные таблицы - **Calc**
- ▶ **awk**