Введение в систему инженерных и научных расчётов Scilab

1. **Введение.**
2. **Элементарные действия.**

1) a=sin(%pi/4); b=sqrt(a)

2) 1/0

3) x=0.1; sin(x)/x

4) x=0.0001; sin(x)/x

5) format('v',25); sin(x)/x //возможно ‘e’ или ‘v’

6) %e; %e^300; log(exp(300)) // если не 300, а 800 то Inf

7) help exp; log(1); log(%i);

3. **Векторы и матрицы. Простой вывод графиков одной переменной**

Показать окно обозревателя переменных. home

1) clear

2) X=[0:10]; X’

3) M=[1,2,3; 4,5,6; 7,8,9]; M(1, 1); M(: ,2); det(M) //близок к предельной точности

4) M=M+5; M(: ,1)=M(: ,1)–5;

5) X=[-2*%pi : 0.1 : 2*%pi];

6) plot(sin(X)); // проблемы с аргументом

7) plot(X, sin(X)); plot(X, sin(X), 'rs-'); plot(X, sin(X), 'rs-', X, cos(X), 'b^-');

8) R=rand(10,1); length(R); R=gsort(R); R=gsort(R,'lc','i'); max(R); min(R); mean(R)

4. **Пользовательские функции. Нахождение корней квадратного уравнения**

Текстовый редактор SciNotes

1) Пусть дано уравнение $2x^2+3x=5$

2)

```
function Z=my2roots(a, b, c)
   d=b^2-4*a*c
   Z(1)=(-b+sqrt(d))/2/a;
```

```
   Z(2)=(-b-sqrt(d))/2/a
endfunction
```

3) выполнить exec('…');

4) Y=my2roots(2,3,-5)

Ещё пример с красивым графиком:

```
for n=1:50 u(n)=(-0.8)^n; end

clf; plot(u,"*r")
```

5. Решение системы линейных алгебраических уравнений

Система m уравнений с n неизвестными вида:

$$
\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1,\\
&a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2,\\
&\dots\dots\dots\ \dots\dots\dots\ \dots\dots\dots\ \dots\dots\dots\ \dots\dots\dots\\
&a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m
\end{aligned}
$$

называется системой линейных алгебраических уравнений (СЛАУ), причём x_j – неизвестные, a_{ij} – коэффициенты при неизвестных, b_i – свободные коэффициенты (i=1…m, j=1…n). Система из m линейных уравнений с n неизвестными может быть описана при помощи матриц $\boldsymbol{A}\cdot\vec{x}=\vec{b}$, где $\vec{x}$ – вектор неизвестных, $\boldsymbol{A}$ – матрица коэффициентов при неизвестных или матрица системы, $\vec{b}$ – вектор свободных членов системы или вектор правых частей. Численным решением системы являются такие значения вектора x, что выполняется уравнение $\boldsymbol{A}\cdot\vec{x}-\vec{b}=0$ с заданной точностью.

1) СЛАУ:

   $2x_1+x_2–5x_3+x_4=8$

   $x_1–3x_2–6x_4=9$

   $2x_2–x_3+2x_4=–5$

   $x_1+4x_2–7x_3+6x_4=0$

2) A=[2 1 -5 1; 1 -3 0 -6; 0 2 -1 2; 1 4 -7 6]; b=[8; 9; -5; 0];
3) x = linsolve(A,b); A*x+b // проверка невязок
4) G = [A, –b]; C = rref(G); x=C(: ,5) // метод Гаусса, верхнетреугольная матрица
5) x=inv(A)*b // по определению СЛАУ
6) x=A\b //используя «левое деление»

6. **Построение трёхмерных графиков**

Предварительно рассмотреть векторное и поэлементное умножение.

A=[1 2 3]; B=[4 5 6];

Чем отличаются A*B; A'*B; A*B'; A.*B

1) **Построить график** поверхности $z = \sin(x) \cdot \cos(y)$ на промежутках от 0 до 2 π
2) X=[0 : 0.1 : 2*%pi]; Y=[0 : 0.1 : 2*%pi];
3) Z=sin(X)'*cos(Y) // обратить внимание на знак транспонирования
4) plot3d(X,Y,Z);

Дополнительный пример:

function z=ff(x,y) z=2*x^2+y^2; endfunction – делается только в редакторе

x=linspace(-1,1,100);

y=linspace(-2,2,200);

z=(feval(x,y,f))';

surf(x,y,z)

5)
6) **Построить график** поверхности $z = \sin(x) + \cos(y)$ на промежутках от 0 до 2π. Предыдущий способ не приводит к положительному результату
7) [x y]=meshgrid(-%pi:0.5:%pi, -%pi:0.5:%pi);
8) z=sin(x)+cos(y);
9) mesh(x,y,z); surf(x,y,z) // plot3d в данном случае не работает
10) plot3d2(x,y,z); plot3d3(x,y,z);
11) **Построить график** параметрически заданной функции (сфера)

    x(u, v) = cos(u)⋅cos(v);

    y(u, v) = cos(u)⋅sin(v);

    z(u, v) = sin(u)
12) u = linspace(-%pi/2,%pi/2,60); v = linspace(0,2*%pi,40);
13) x=cos(u)'*cos(v); y=cos(u)'*sin(v); z=sin(u)'*ones(v);
14) plot3d2(x,y,z);
15) **Ракушка**

$$\begin{cases} x = \cos(u) \cdot u \cdot \left(1 + \cos\dfrac{(v)}{2}\right) \,; \\ y = \dfrac{u}{2} \cdot \sin(v) \,; \\ z = (\sin(u) \cdot u) \cdot \left(1 + \cos\dfrac{(v)}{2}\right) \,. \end{cases}$$

16) u = linspace(0,2*%pi,40); v = linspace(0,2*%pi,20);

17) x = (cos(u).*u)'*(1+cos(v)/2);

18) y = (u/2)'*sin(v);

19) z = (sin(u).*u)'*(1+cos(v)/2);

20) plot3d2(x,y,z);

Гистограмма трёхмерная: hist3d(100*rand(10,10),20,10);

Построение нескольких графиков на одном поле

21) mtlb_hold // альтернатива
22) x=[-2*%pi:0.1:2*%pi];
23) subplot(2,2,1) ; plot2d(x,x^2, style=color("red"), axesflag=0);
24) и так далее

7. Аппроксимация экспериментальных данных

Метод наименьших квадратов позволяет по экспериментальным данным подобрать такую аналитическую функцию, которая проходит настолько близко к экспериментальным точкам, насколько это возможно.

Пусть в результате эксперимента были получены некоторые данные, отображенные в виде таблицы (табл. 11.1). Требуется построить аналитическую зависимость, наиболее точно описывающую результаты эксперимента.

Таблица 11.1. Экспериментальные данные

x_i	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$\ldots$	x_n
y_i	y_1	y_2	y_3	y_4	y_5	y_6	y_7	$\ldots$	y_n

Идея метода наименьших квадратов заключается в том, что функцию $Y = f(x, a_0, a_1, \ldots, a_k)$ необходимо подобрать таким образом, чтобы сумма квадратов отклонений измеренных значений y_i от расчетных Y_i была наименьшей:

$$S = \sum_{i=1}^{n} (y_i - f(x_i, a_0, a_1, \ldots, a_k))^2 \to \min \tag{11.1}$$

Задача сводится к определению коэффициентов a_i из условия (11.1). Для реализации этой задачи в Scilab предусмотрена функция

```
[a,S]=datafit(F,z,c)
```

```
X = linspace(0,50,100);
Y = (X.^2-X)./100+7*rand(1,100);

function Y=myapp(c, x)
        Y = c(1)+c(2)*exp(-c(3)*x);
endfunction

function res=resapp(c, Z)
  res=Z(2)-myapp(c, Z(1));
endfunction
```

1) plot(X,Y,'bo'); xgrid // исходный набор точек
2) [A, err]=datafit(resapp,[X; Y],[0; 0; 0]); //вычисление коэффициентов
3) plot(X,Y,'ro', X,myapp(A,X),'b');

В случае полиномиальной аппроксимации необходимую функцию несложно придумать самому, или воспользоваться «атомами» (atoms.scilab.org): **polyfit**

[A,Yd] = polyfit(X,Y,7); plot(X,Y,'bs',X,Yd,'r');

4) привести пример аппроксимации прямой удалением пары коэффициентов

   линейная регрессия:

5) a=regress(X,Y);

6) plot(X,Y,'ro', X,a(1)+a(2)*X,'b');

7) R=sqrt(1-sum((Y-(a(1)+a(2)*X))^2)/sum((Y-mean(Y))^2)) // индекс корреляции

$$R = \sqrt{1 - \frac{\sum_{i=1}^{n}(y_i - Y_i)^2}{\sum_{i=1}^{n}(y_i - M_y)^2}}$$

8. Поиск минимума функции (одной переменной)

```
function f=myf(x)
  f=x^4+3*x^3-13*x^2-6*x+26;
endfunction

function [f, g, ind]=myminf(x, ind)
  f=myf(x);
  g=numdiff(myf, x);
endfunction
```

x=[-5 : 0.5 : 5]

1) plot(x,myf(x)); xgrid // имеются два минимума

2) [fmin, xmin]=optim(myminf, 1)

3) [fmin,xmin]=optim(myminf, -1)

9. Решение нелинейных уравнений

1) deff('[y]=myequ(x)', 'y=exp(x)/5-2*(x-1)^2')

2) x=[-5:0.2:6]; plot(x,myequ);

3) fsolve([0; 2; 5], myequ) // нашли все корни

10. Решение обыкновенных дифференциальных уравнений

Решить задачу Коши

$$x' = \cos(xy)\,,$$
$$y' = \sin(x + ty)\,,$$
$$x(0) = 0, \quad y(0) = 0.$$

на интервале [0; 10].

```
function dy=mydifeq(t, y)
dy=zeros(2,1); // чтобы не было проблем с неизвестно откуда взявшимися цифрами
dy(1)=cos(y(1)*y(2));
dy(2)=sin(y(1)+y(2)*t);
endfunction
```

1) t=[0 : 0.1 : 10];
2) y=ode([0; 0], 0, t, mydifeq); // x(t0), y(t0), t0, диапазон, имя функции
3) plot(t, y);

Примеры решения физических задач

1. Электрическое поле системы неподвижных зарядов.

Скалярный потенциал электрической системы, состоящей из N электрических зарядов, $q_1, q_2 \ldots q_N$, и напряженность электрического поля удовлетворяют принципу суперпозиции:

$$\varphi(\vec{R}) = \frac{1}{4\pi\varepsilon_0} \sum_{i=1}^{N} \frac{q_i}{|\vec{R} - \vec{r}_i|}, \tag{5.3}$$

$$\vec{E}(\vec{R}) = \frac{1}{4\pi\varepsilon_0} \sum_{i=1}^{N} \frac{q_i}{|\vec{R} - \vec{r}_i|^3} (\vec{R} - \vec{r}_i), \tag{5.4}$$

где $\vec{r}_i$ – координата i-го заряда.

Функция, рассчитывающая значения потенциала в узлах координатной сетки

```
function res=potential(q, xq, yq, X, Y)
//q - вектор, содержащий значения электрических зарядов
//xq, yq - координаты соответствующих зарядов
// X,Y - координаты узлов сетки для расчёта потенциала
   eps0=8.85e-12;
   Nq=length(q); // число зарядов
   Nx=length(X);
   Ny=length(Y);
```

```
  for l=1:Ny
    for j=1:Nx
      s=0
      for k=1:Nq
        s=s+q(k)/sqrt((X(j)-xq(k))^2+(Y(l)-yq(k))^2);
      end;
      D(l,j)=s/(4*%pi*eps0);
    end;
  end;
  res=D;
endfunction
```

Вычислим потенциал, создаваемый системой из 40 зарядов, расположенных в линию

```
e=1.6e-19;
r0=1e-6; //элементарное расстояние, м
N=40; // число зарядов
M=80; // число узлов в сетке
q=ones(1,N)*e;
xq=linspace(-5*r0,5*r0,N);  yq=zeros(1,N); //заряды расположены равномерно в линию
X=linspace(-10*r0,10*r0,M); Y=X; // вычисление координат узлов сетки
Psi=potential(q,xq,yq,X,Y);
[X1,Y1]=meshgrid(X,Y);
surf(X1,Y1,Psi); // распределение потенциала системы зарядов
contour(X,Y,Psi,20); //либо так
```

2. Линейный гармонический осциллятор.

$$\ddot{x} + \omega_o^2 x = 0$$

```
function dy=myosc(t, y)

global omega

dy=zeros(2,1);

dy(1)=y(2);

dy(2)=-omega^2*y(1);

endfunction
```

```
global omega
k=9; // коэффициент жёсткости
m=1; // масса осциллятора
T=sqrt(2*%pi*k/m); // период колебаний
omega=2*%pi/T; // циклическая частота
t=linspace(0,5*T,80);
y=ode([0.5; 1], 0, t, myosc);
plot(t,y(1,:),'r'); // координата
mtlb_hold on
plot(t,y(2,:),'b'); // скорость
plot(y(1,:),y(2,:)); // фазовая плоскость
comet(y(1,:),y(2,:)); // визуализация движения по фазовой плоскости
```