К лабораторной работе №1

Нелинейные уравнения и системы

В общем случае аналитическое решение уравнения f(x) = 0 можно найти только для узкого класса функций. Чаще всего приходится решать его численными методами. Численное решение уравнения проводят в два этапа:

- отделяют корни уравнения, то есть находят достаточно тесные промежутки, в которых содержится только один корень, их называют интервалами изоляции корня, и определить их можно, изобразив график функции или любым другим методом, основанным на том, что непрерывная функция f(x) имеет на интервале [a; b] хотя бы один корень; если она поменяла знак $f(a) \cdot f(b) < 0$, a и b называют пределами интервала изоляции;
- на втором этапе проводят уточнение отделенных корней, то есть находят корни с заданной точностью.

Любое уравнение P(x) = 0, где P(x) - это многочлен, отличный от нулевого, называется алгебраическим уравнением (полиномом) относительно переменных x. Всякое алгебраическое уравнение относительно x можно записать в виде:
$a_0x^n + a_1x^{n-1} + \ldots + a_{n-1}x + a_n = 0$,
где $a_0 \neq 0$, $n \geq 1$ и a_i. - коэффициенты алгебраического уравнения n-й степени.

В MATLAB полином задается и хранится в виде вектора, элементами которого являются коэффициенты полинома от a_n до a_0.
Рассмотрим функции, предназначенные для действий над полиномами:

- conv(p1, p2) - формирует вектор, соответствующий коэффициентам полинома степени n+m, полученного в результате умножения вектора p1 степени n на вектор p2 степени m, то есть вычисляет произведение двух полиномов;
- deconv(pl, p2) -осуществляет деление полинома p1 на полином p2, то есть формирует вектор коэффициентов полинома, который является частным от деления p1 на p2; вызов функции в общем виде [q, r]=deconv(pl, p2) приводит к получению двух полиномов: q - частного от деления и r -остатка от деления;
- polyval(p1, x) - вычисляет значение полинома с коэффициентами p1 в точке x;
- polyder (p1 [, p2]) - формирует вектор коэффициентов полинома, являющегося производной от полинома с коэффициентами p1, то есть вычисляет производную от полинома; если в качестве аргументов указаны два вектора polyder(p1, p2), то производная вычисляется от произведения этих векторов, вызов функции в общем виде [q, r]= polyder (p1, p2) приведет к вычислению производной от частного pl/p2 и выдаст результат в виде отношения полиномов q и r.

Решить алгебраическое уравнение в MATLAB можно при помощи встроенной функции roots(p). Она формирует вектор, элементы которого являются корнями полинома с коэффициентами p. Наоборот, построить вектор p по заданному вектору его корней можно с помощью функции poly(x).
Для решения уравнений вида f(x)=0 в MATLAB применяют функцию fzero(name, x0), где:

- name – имя M-функции, вычисляющей левую часть уравнения;
- x0 – начальное приближение к корню или интервал его изоляции [a, b];

Формат вызова функции может выглядеть так: [x, y]=fzero(name, x0), где x – корень уравнения, y – значение функции в точке x.

Пример(использование *conv* и *deconv*):

```
pl=[2 0 1];
p2=[1 0 0 -1];
%Умножение полиномов: (2x^2+1) (x^3-1) = 2x^5+x^3-2x^2-1
p=conv(p2,pl)
```

Результат:

```
p =
2 0 1 -2 0 -1
%Деление полиномов: (2x^5+x^3-2x^2-1) / (x^3-1) = (2x^2+1)
deconv(p,p2)
```

Результат:

```
ans =
2 0 1
```

Пример(использование *polival*):

```
p1=[2 1 0 -1 0 -3];
%Значение полинома (2x^5+x^4-x^2-3) в точке x=0
polyval(p1,0)
```

Результат:

```
ans =
-3
```

Пример(использование *polider*):

```
p1=[2 1 0 -1 0 -3];
polyder(p1); %Производная от p1
p2=[1 0 0 -1];
polyder(p1,p2); %Производная от (p1*p2)
[q, r]=polyder(p1,p2) %Производная от частного и остатка (p1/p2)
```

Результат:

```
q =
4 1 0 -9 -4 9 2 0
r =
1 0 0 -2 0 0 1
```

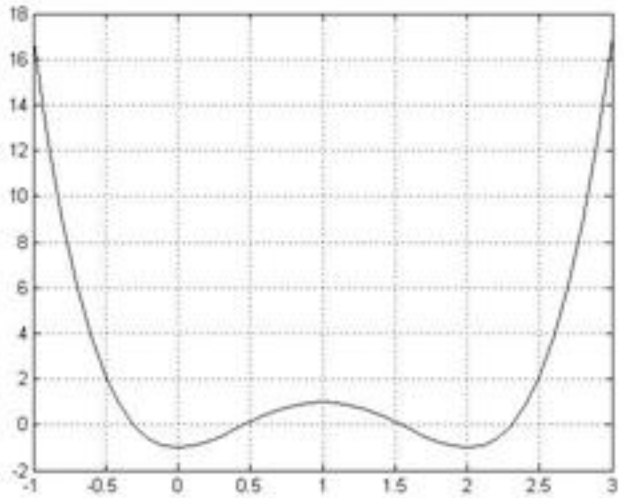
Пример(использование *roots*):

Найти корни полинома $2x^4 - 8x^3 + 8x^2 - 1 = 0$.

```
p=[2 -8 8 0 -1];
roots(p)
%Найдем графическое решение
x=-1:0.1:3;
y=polyval(p,x);
plot(x,y,'-k'),grid
```

Результат:

```
ans =
2.3066
1.5412
0.4588
-0.3066
```



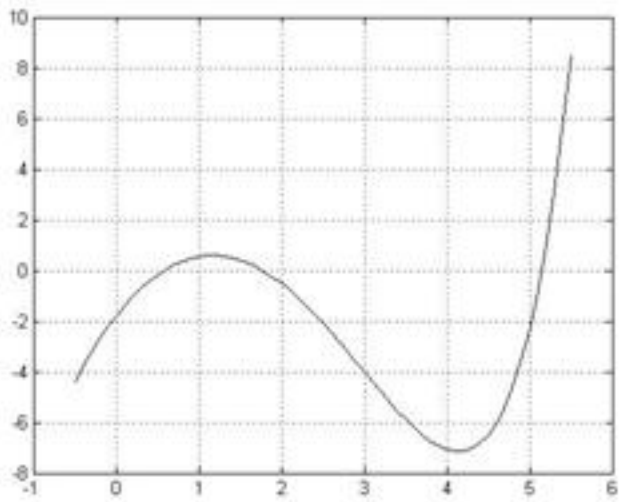
Пример:
Найти корни уравнения $(e^x/5)-2(x-1)^2=0$
В М-файле с именем ff.m пишем:

```
function y=f(x)
y=exp(x)/5-2*(x-1).^2;
end
```

Потом в командном окне пишем:

```
%Построение графика
x=-0.5:0.1:5.5;
y=exp(x)/5-2*(x-1).^2;
plot(x,y,'-k'), grid
[x(1),y(1)]=fzero('ff',[0 1]);
[x(2),y(2)]=fzero('ff',[1 2]);
[x(3),y(3)]=fzero('ff',[5 6]);
x
y
```

Результат:



x=
0.5778 1.7639 5.1477

y=
-0.0000 0.0000 0
Пример:
Решить систему уравнений:

$$\begin{cases} \sin(x+1) - y = 1.2 \\ 2x + \cos(y) = 2 \end{cases}$$

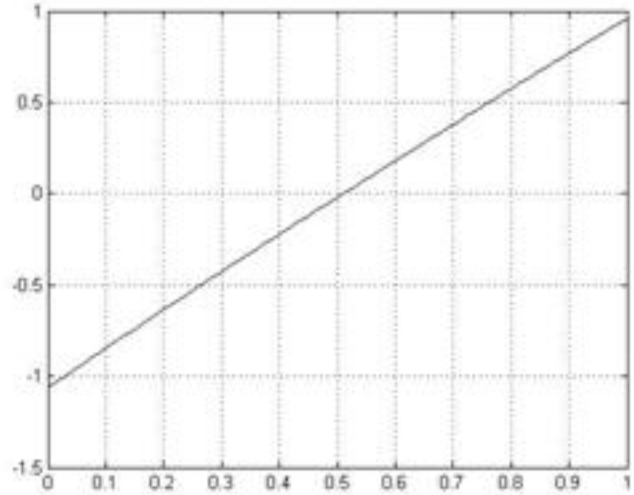
В М-файле с именем f25.m пишем:

```
function y=f25(x)
z=sin(x+1)-1.2;
y=2*x+cos(z)-2;
end
```

Потом в командном окне пишем:

```
x=0:0.01:1;
y=f25(x);
plot(x,y,'-k'), grid
```

Результатом будет график:



Потом находим решение системы(пишем в командном окне):

```
x=fzero('f25',0)
y=sin(x+1)-1.2
```

Результат:
x =
0.5102
y =
-0.2018

При решении задач максимума и минимума функции y = f(x) одной переменной выделяют задачи локального (на каком-либо интервале) и глобального (на всей числовой оси) экстремума. В MATLAB поиск локального минимума осуществляет функция:

[x, y]= fminbnd(name, a, b [, options]), где name - имя М-функции, вычисляющей значение f(x), a, b - границы интервала, на котором осуществляется поиск минимума,

options - параметры, управляющие ходом решения, x, y - координаты точки, в которой достигается минимум функции на заданном интервале.

Функцию fminbnd можно использовать и для вычисления локального максимума. Для этого достаточно взять функцию name с противоположным знаком.

При вычислении экстремума функции создается М-файл, где прописывается ссылка на исследуемую функцию и сама функция. Затем в командной строке прописывается интервал вычисления, на котором необходимо найти экстремум, после чего и сама функция. После этого делаем запись с вызовом исследуемой функции из М-файла со знаком «-», после чего программа выдает координаты экстремума функции.

Пример:
В М-файле с именем mf.m пишем:

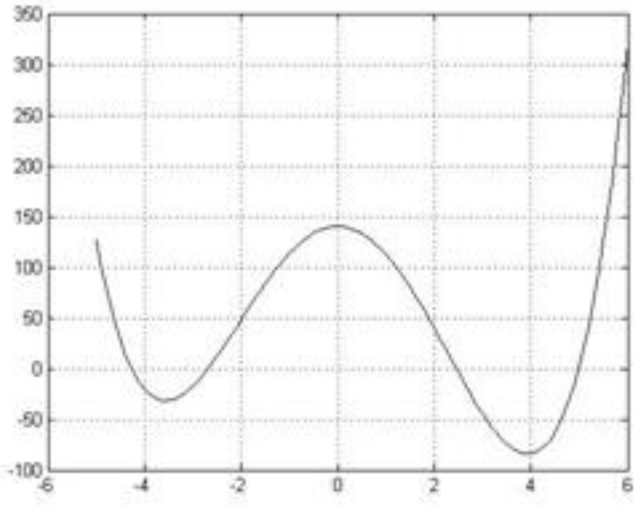
```
function y=mf(x)
y=x.^4-0.5*x.^3-28*x.^2+140;
end
```

Потом в командном окне пишем:

```
x=-5:0.1:6;
y=x.^4-0.5*x.^3-28*x.^2+140;
plot(x,y,'-k'), grid
%Максимум функции на интервале [-2 2]
y=-mf(x);
[x,y]=fminbnd(@mf,-2,2)
```

Результат:

```
x =
3.7224e-008
y =
-140.0000
```



Поиск экстремума функции нескольких переменных.

Вычисление *экстремума функции многих переменных* $z=f(x_1,x_2,\ldots,x_n)$ осуществляет команда:

[x, z] = fminsearch(name, x0 [, options]

где:

- name - имя М-функции, вычисляющей значение $z=f(x_1,x_2,\ldots,x_n)$, зависящей от n переменных;
- x0 – вектор из n элементов, содержащий координаты точки начального приближения;
- options – параметры, управляющие ходом решения;
- x - из n элементов, содержащий координаты точки, в которой достигается минимум функции;
- z – значение функции в точке с координатами x.

Пример:

Найти минимум функции $f(x,y)=\sqrt{x^2+y^2}$

[z,f] = fminsearch(@(x) sqrt(x(1)^2+x(2)^2), [2,2])

%Построение графика

[x y]=meshgrid(-2:0.2:2, -2:0.2:2);

z=sqrt(x.^2+y.^2);

surf(x,y,z);

Результат:

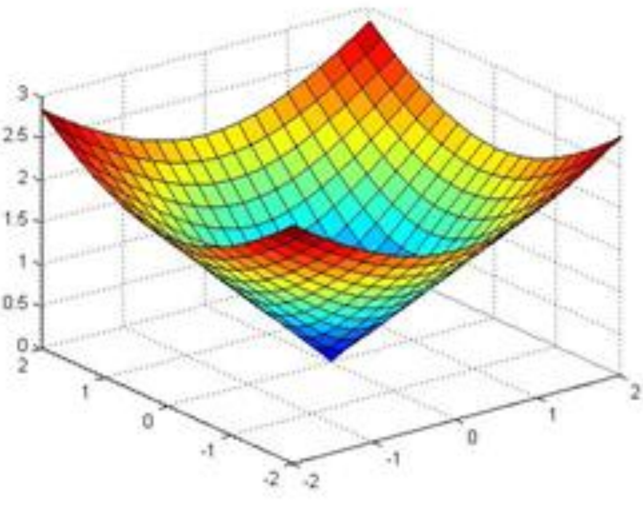
z =

1.0e-004 *

-0.4133 -0.1015

f =

4.2559e-005



Задание к лабораторной работе №1

Найти корни уравнения, а также определить локальные минимумы и максимумы функции на интервале от -10 до 10.
y=3x^6-x^4-5x^3-3x^2+6x-5