АЛГЕБРАИЧЕСКИЕ УРАВНЕНИЯ И ФУНКЦИИ

Алгебраические уравнения

Канонический вид
$F(x)=0$

Полиномы
ax^2+bx+c

Трансцендентные
(ln, exp, sin…)

Этапы нахождения корней

При отыскании корней во всех случаях последовательно решается две задачи:

1. Отделение корней, т.е. определение областей в каждой из которых заключён один и только один корень.
2. Вычисление корней с заданной точностью.

Методы отделения корней

<u>Аналитический.</u> Если на отрезке [a, b] функция непрерывна и монотонна, ее значения на концах отрезка имеют разные знаки, то на рассматриваемом отрезке существует один и только один корень.

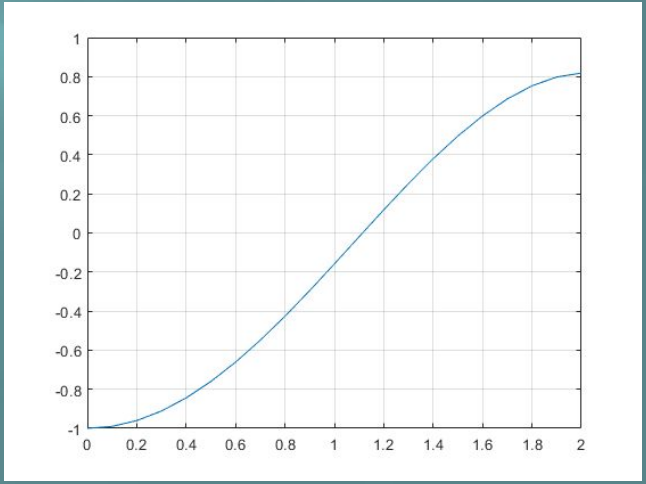
П*од этот критерий не подпадают кратные корни уравнений!*

<u>Табличный.</u> Надежность рассмотренного подхода к отделению корней уравнений зависит как от характера самой функции $f(x)$, так и от выбранной величины шага.

<u>Графический.</u>

x·sin(x)=1

x·sin(x)-1=0



Решение уравнений в пакете Matlab (универсальный метод)

Для решения уравнения произвольного вида используется

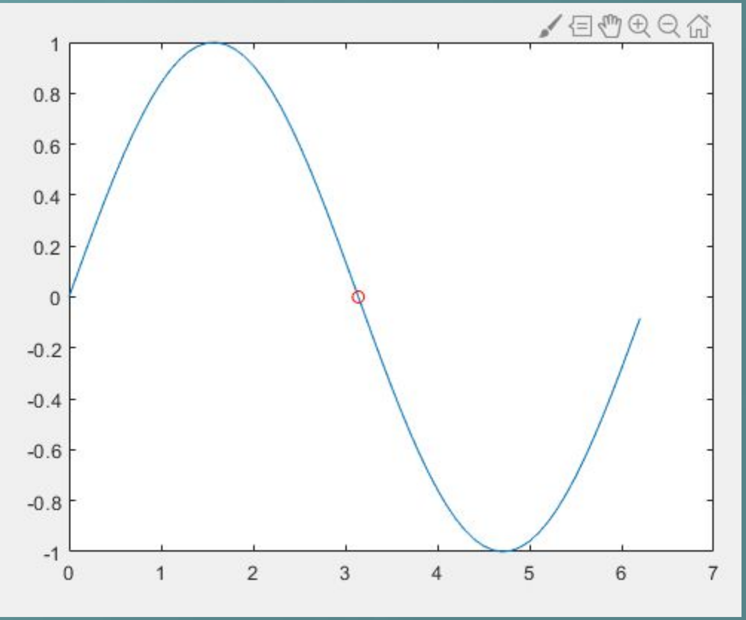
x = **fzero**(fun,x0,*options*)

fun – функция для которой ищется корень
x0 – начальное приближение для корня / начальный интервал
options – опции, задающие модификации параметров решателя – *необязательный параметр*

Использование fzero в случае задания начального приближения для корня

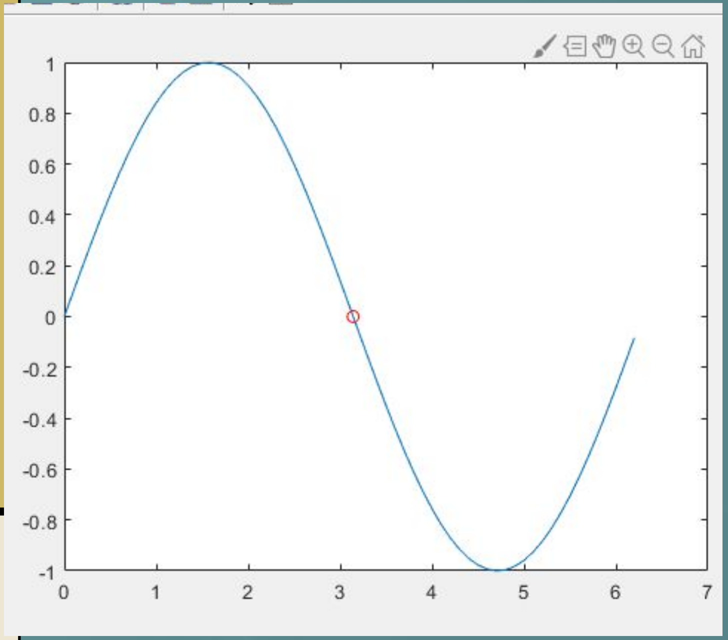
Знак @ должен быть обязательно!!!!!

Ф	fun = **@**sin; % функция **x0 = 3;** % начальная точка res = fzero(fun,x0) x = 0:0.1:2*pi; plot(x,sin(x), res,0,'ro')
КС	x = 3.1416



Использование fzero в случае задания начального интервала

Ф	fun = @sin; % функция **x0 = [1 4];** % начальный интервал res = fzero(fun,x0) x = 0:0.1:2*pi; plot(x,sin(x), res,0,'ro')
КС	x = 3.1416



Опции для функции fzero

Опция	Возможные значения
Display	•'off' ничего не выводится на дисплей. •'iter' выводится информация по каждой итерации. •'final' выводится информация по последней итерации. •'notify' выводится информация в случае, если процесс не сходится (по умолч).
FunValCheck	Проверка правильности значений целевой функции. 'on' отображает ошибку, когда целевая функция возвращает комплексное значение, Inf или NaN. • 'off', проверка не проводится (по умолч).
PlotFcns	Выводит график сходимости во время выполнения алгоритма. Требует дескриптор функции или массива ячеек функций. По умолчанию установлено значение none ([]). @optimplotx отображает текущую точку. @optimplotfval отображает значение функции..
TolX	Изменяет точность вычислений, по умолчанию $2.2204 \cdot 10^{-16}$

Пример использования опций для функции fzero

Установка опций производится с помощью функции

optimset ('параметр', 'значение')

```
fun = @(x) exp(-exp(-x)) - x; % function
 x0 = [0 1]; % initial interval
options = optimset('Display','iter'); % show iterations
 x = fzero(fun,x0,options)
```

Пример использования опций для функции fzero

```
>> fun = @(x) exp(-exp(-x)) - x; % function
 x0 = [0 1]; % initial interval
options = optimset('Display','iter'); % show iterations
 x  = fzero(fun,x0,options)

 Func-count    x          f(x)             Procedure
    2               1       -0.307799        initial
    3        0.544459       0.0153522        interpolation
    4        0.566101      0.00070708        interpolation
    5        0.567143    -1.40255e-08        interpolation
    6        0.567143     1.50013e-12        interpolation
    7        0.567143               0        interpolation

Zero found in the interval [0, 1]

x =

    0.5671
```

Что еще возвращает функция fzero

Функция может возвращать более одного значения – в этом случае возвращаемые значения записываются в квадратных скобках через пробел!

[x fval exitflag output] = **fzero(**fun,x0**,***options***)**

x - решение
fval – значение функции в точке x
exitflag – флаг выхода:
output - структура, поля которой содержат информацию о процессе вычисления корня

Что возвращает функция fzero (пример)

```
x =

    0.5671


fval =

     0


exitflag =

     1


output =

  struct with fields:

    intervaliterations: 0
            iterations: 5
             funcCount: 7
             algorithm: 'bisection, interpolation'
               message: 'Zero found in the interval [0, 1]'
```

Значения Exitflag

Exitflag – флаг выхода
1 - Функция сходится к решению x.
-1 – выполнение алгоритма прервано функцией вывода или графиком.
-3 - При поиске интервала знакопеременности функции было обнаружено значение функции NaN или Inf.
-4 - Комплексное значение функции было обнаружено при поиске интервала знакопеременности.
-5 - алгоритм сходиться к особой точке.
-6 - fzero не обнаружил интервал знакопеременности.

Значения полей структуры Output

Output – структура, содержащая информацию о:

- числе итераций, выполненных для уточнения отрезка, содержащего корень
- числе итераций, выполненных для уточнения корня
- Числе вычислений функции
- Использованном алгоритме
- Текстовое сообщение о значении exitflag

Правила оформления заголовка функции в Matlab

1. На первом месте всегда стоит ключевое слово **function (функция).**
2. На втором месте стоит возвращаемое значение - имя переменной, которой будет присваиваться возвращаемое функцией значение.
3. На третьем месте стоит знак «=».
4. На четвертом месте стоит имя функции
5. На пятом месте в круглых скобках стоят входные аргументы, которые функция принимает – **список формальных аргументов**.

Заголовок функции никогда не заканчивается «;»!!!

Пример оформления заголовка функции в Matlab

function y = f(x)

1. Ключевое слово **function.**
2. Имя переменной, которой будет присваиваться возвращаемое функцией значение - **y**.
3. На третьем месте стоит знак «=».
4. Имя функции - **f**
5. В круглых скобках стоит аргумент, который функция принимает – формальный аргумент - **x**.

Комментарии к правилам оформления заголовка функции в Matlab

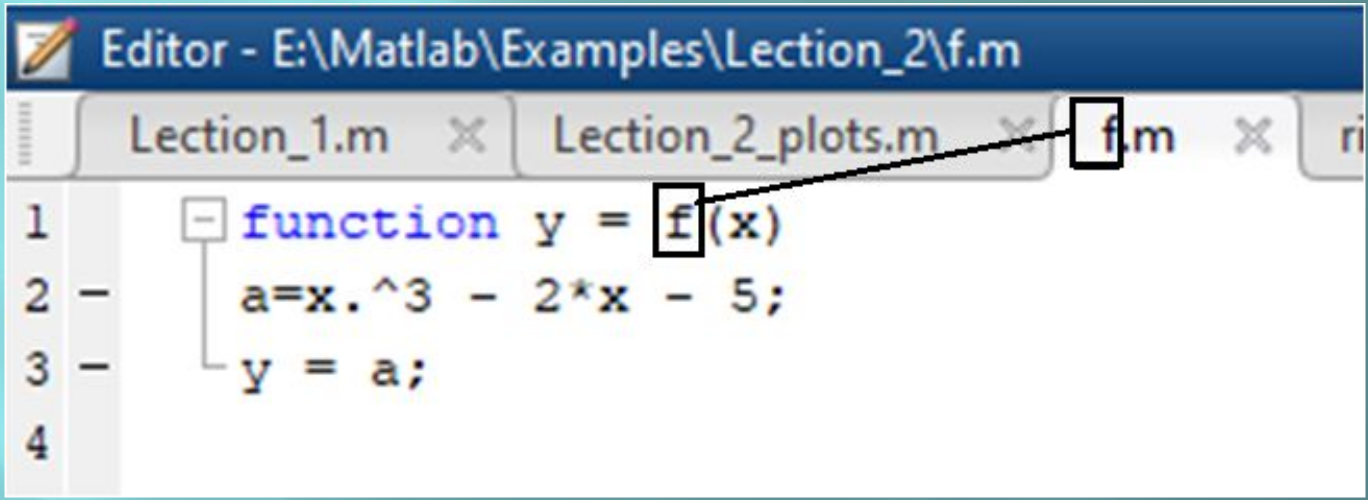
1. Обязательными параметрами являются ключевое слово функция и имя функции.
2. Возвращаемое значение и входные аргументы могут отсутствовать.
3. Рекомендуется всегда возвращать результат работы функции – хотя бы в виде логической переменной, принимаемой значения 0/1 (false/true), для возможности контроля корректности работы функции

Правила оформления файла для функции в Matlab

1. Файл, в котором будет сохраняться функция, должен иметь то же имя, что и функция.
2. Путь к каталогу с файлом должен быть прописан в Set Path.
3. Один файл – одна функция! В одном файле может быть несколько функций, но «видеть» Matlab будет только ту функцию, имя которой совпадает с именем файла.
4. После редактирования файл всегда должен быть сохранен перед использованием, иначе изменения не вступят в силу.

Функция для задания уравнения

Написание функции для нахождения корня алгебраического уравнения



Здесь используется покомпонентное возведение в степень, поскольку вид *x* заранее неизвестен: число, вектор, матрица….

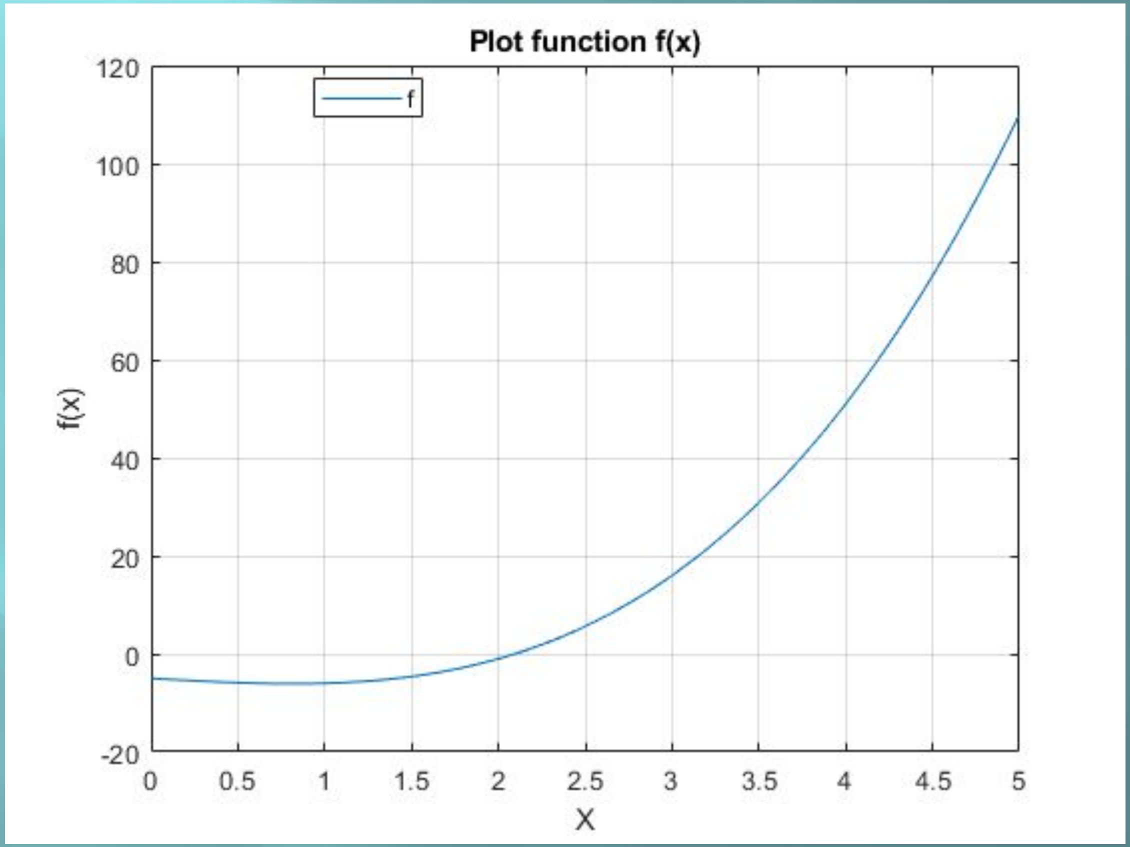
Функция для построения графика функции

Написание функции для построения графика произвольной функции в произвольном диапазоне

```
Editor - E:\Matlab\Examples\Lection_2\risunok.m
Lection_1.m | Lection_2_plots.m | f.m | risunok.m

function risunok(x)
xl=x(1):0.01:x(end);
y=f(xl);
plot(xl,y)
grid on
title('Plot function f(x)');
xlabel('X');
ylabel('f(x)');
legend({'f'},'Location','best');
```

Построение графика с помощью функции risunok

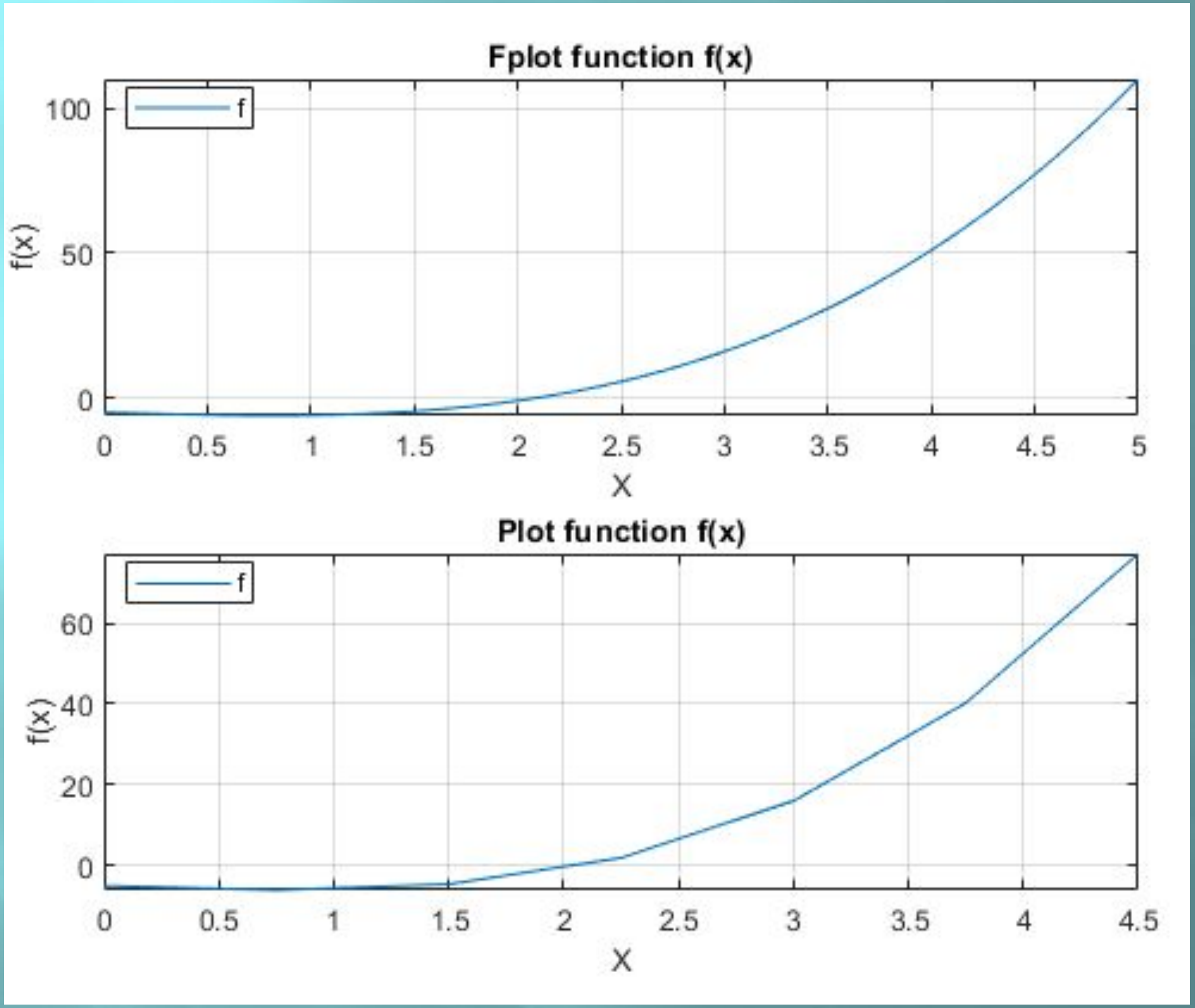
```
>> risunok([0 5])
```



Построение графика с автоматическим выбором шага

```
function risunok_fplot(x)
fplot(@f,[x(1) x(end)])
grid on
title('Plot function f(x) with fplot');
xlabel('X');
ylabel('f(x)');
legend({'f'},'Location','best');
```

Сравнение функций plot и fplot

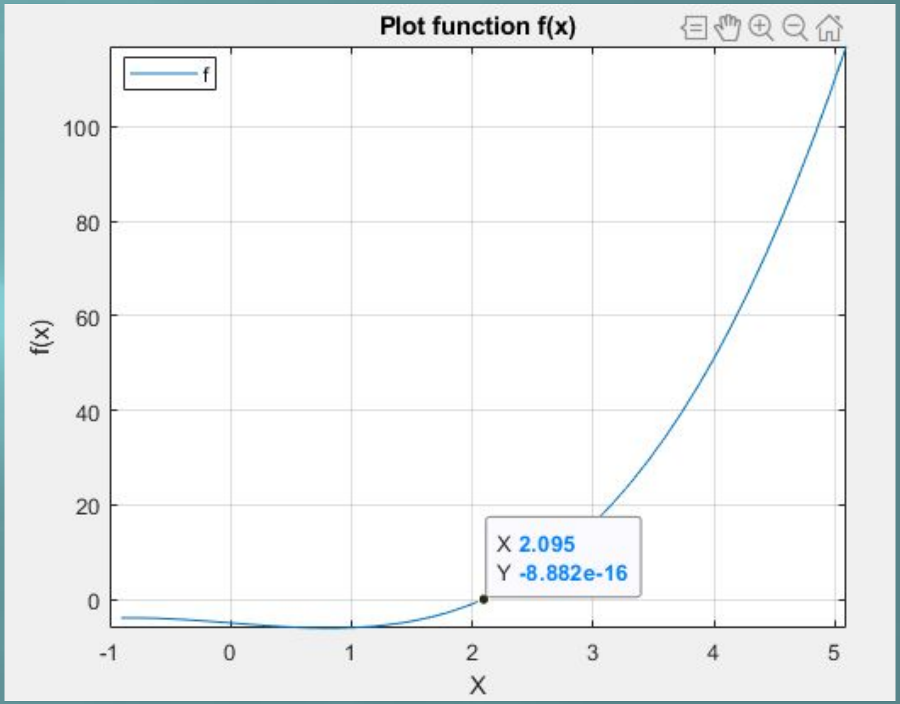


Шаг выбран автоматически

Шаг 0,75

Решение уравнений в случаях, когда функция задана в файле

Ф	fun = @f; % функция x0 = 2; % начальная точка z = fzero(fun,x0) risunok_fplot([z-3 z+3])
КС	z = 2.0946



Функция для решения уравнений

```
function my_zero(f,x)
fun = f; % function
 options = optimset('Display','iter'); % show iterations
 [x fval exitflag output] = fzero(fun,x,options)
 risunok_fplot([x-2 x+2])
 hold on
 plot(x,0,'ro')
```

Функция имеет 2 входных аргумента – ссылку на функцию, для которой будет решаться уравнение и x, который может являться или начальным приближением или отрезком на котором корень единственный.

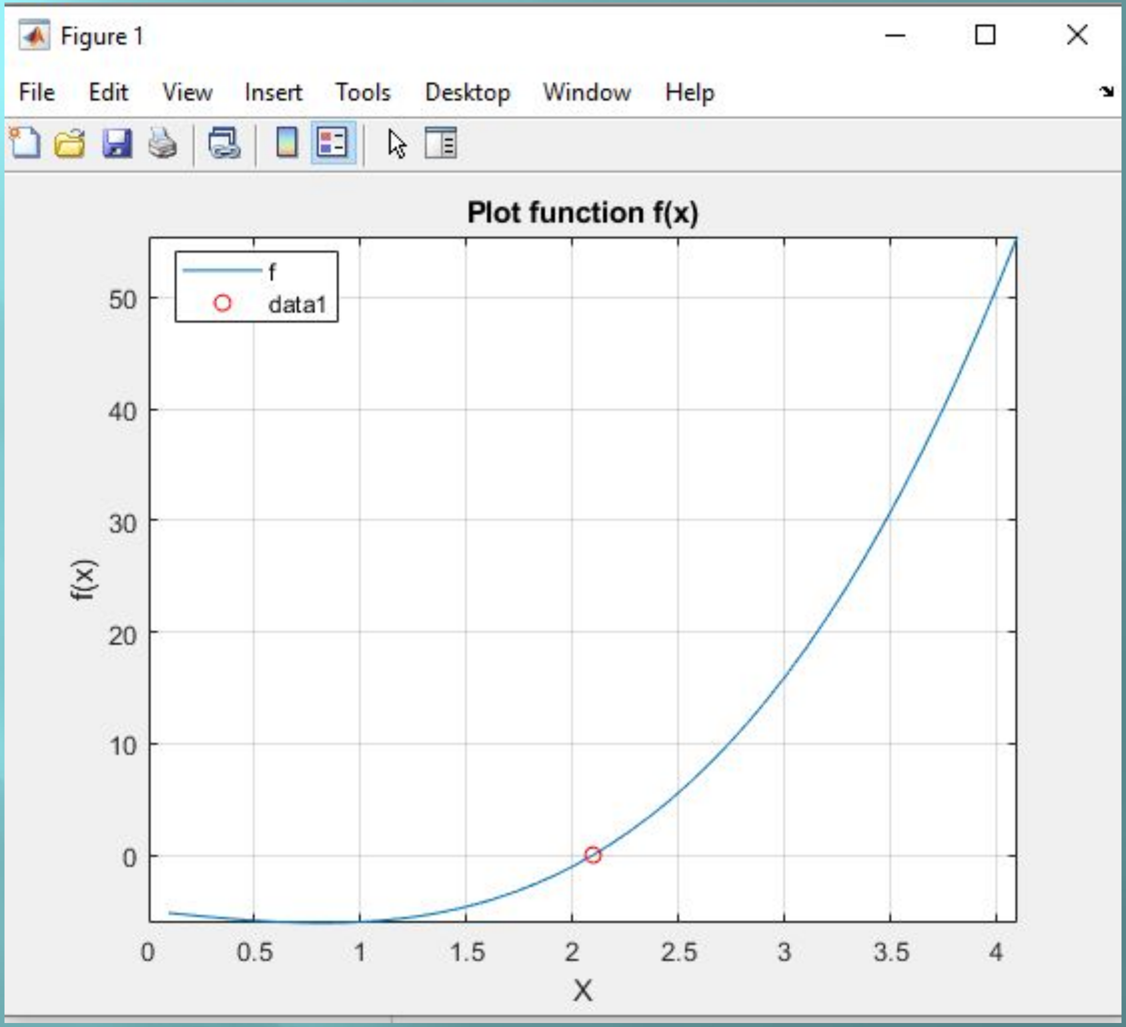
Вызов функции для решения уравнений

\>> my_zero(@f,3)

Функция f заранее написана и сохранена в папке, имеющийся в путях Matlab

```
   21         2.09455   -8.88178e-16        interpolation
   22         2.09455   -8.88178e-16        interpolation

Zero found in the interval [2.04, 3.67882]

x =

    2.0946


fval =

  -8.8818e-16


exitflag =

     1


output =

  struct with fields:

    intervaliterations: 8
            iterations: 6
             funcCount: 22
             algorithm: 'bisection, interpolation'
               message: 'Zero found in the interval [2.04, 3.67882]'
```

Функция для решения уравнений



Нахождение корней полинома

Для нахождения корней полинома используется функция

x = roots(a)

a – вектор коэффициентов полинома, записанный от старшей степени.

x – вектор корней полинома

Ф	**%polynom $3x^2$-2x-4=0** **p = [3 -2 -4];** **r = roots(p)**
КС	r = 1.5352 -0.8685

Комментарий

Для вычисления значений полинома по известным коэффициентам можно использовать функцию

r=polyval (p,x)

p – коэффициенты полинома, начиная от старшего

x – точка, в которой необходимо вычислить значение полинома

r – значение полинома с коэффициентами p в точке x

Ф	**p = [3 -2 -4];** **r=polyval(p,2)**
КС	r = 4

Решение параметрических уравнений

Найти решения квадратного уравнения $x^2+b\cdot x-8$ в зависимости от параметра b. Привести зависимость величины корней от параметра b. Параметр b изменяется в диапазоне от -1 до 5 с шагом 0.1 (b=-1:0.1:5;)

1. Создать вектор b из 60 элементов
2. для каждого значения вектора b (индекс *i* от 1 до 60)
3. *Начать повторяющееся действие*

- сформировать вектор коэффициентов полинома $x^2+b\cdot x-8$
- найти корни уравнения $x^2+b\cdot x-8$
- записать корни в вектора x1 и x2 на соответствующее место

4. Закончить *повторяющееся действие*

Цикл

Цикл – это разновидность управляющей конструкции, предназначенная для организации многократного исполнения набора инструкций.

Бывают:

- Цикл с предусловием
- Цикл с постусловием
- **Цикл со счетчиком**

Программирование цикла

1. Создать вектор b из 50 элементов
2. для каждого значения вектора b (индекс *i* от 1 до 60)
3. *Начать повторяющееся действие*

- сформировать вектор коэффициентов полинома $x^2+b\cdot x-8$
- найти корни уравнения $x^2+b\cdot x-8$
- записать корни в вектора x1 и x2 на соответствующее место

4. Закончить *повторяющееся действие*

1. b=0:0.1:5;
2. for *i* = 1:length(b)
3.

- p=[1 b(i) -8]
- r=roots(p)
- x1(i)=r(1); x2(i) = r(2);

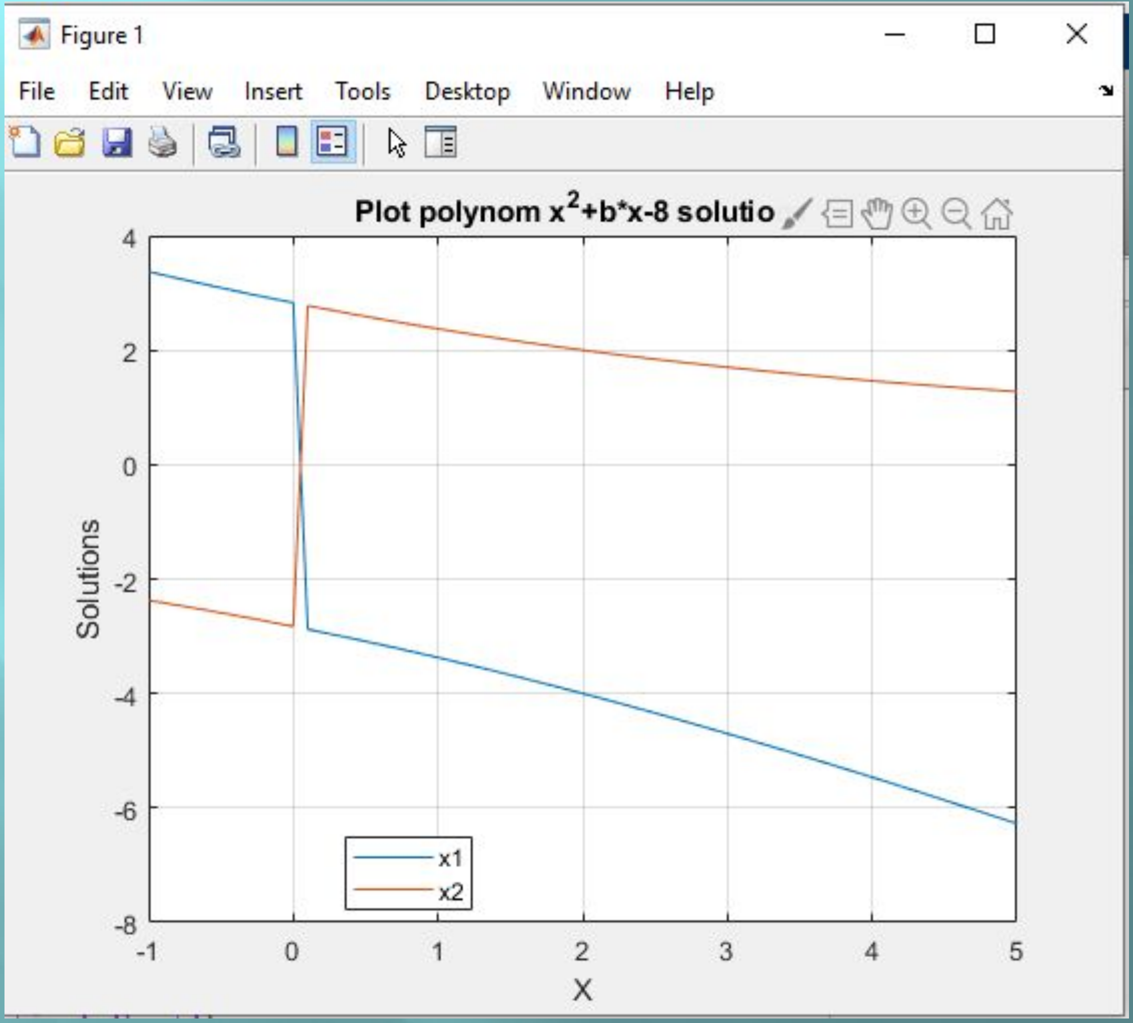
4. end

Решение параметрических уравнений (функция в Matlab)

my_zero.m | risunok_fplot.m | risunok.m | scenario.m | parametric_polynom.m

```
function parametric_polynom(B)
b=B(1):0.1:B(end);
for i  = 1:length(b)
 p=[1 b(i) -8];
 r=roots(p);
 x1(i)=r(1);
 x2(i) = r(2);
end
plot(b,x1,b,x2)
print_text
end
%function print_text may  used only in function parametric_polynom
function print_text
grid on
title('Plot polynom x^2+b*x-8 solutions');
xlabel('X');
ylabel('Solutions');
legend({'x1','x2'},'Location','best');
end
```

Решение параметрических уравнений (график в Matlab)



Решение уравнений в символьном виде

Для получения аналитического решения в Matlab есть функция

r=solve(eqn, var)

eqn – уравнение, записанное в виде RHS == LHS
var – переменная, по которой решается уравнение
r – возвращаемое значение – структура, каждый элемент которой представляет собой аналитическую запись решения

Решение уравнений в символьном виде

<table>
<tr><td>Ф</td><td><pre>syms a b c x %create symbolic parameters
eqn = a*x^2 + b*x + c == 0
S = solve(eqn)</pre></td></tr>
<tr><td>КС</td><td><pre>eqn =

a*x^2 + b*x + c == 0

S =

 -(b + (b^2 - 4*a*c)^(1/2))/(2*a)
 -(b - (b^2 - 4*a*c)^(1/2))/(2*a)</pre></td></tr>
</table>

Численное решение уравнений, записанных в символьном виде

Для получения численного решения в Matlab есть функция

r=vpasolve(eqn, var, init_param)

eqn – уравнение, записанное в виде RHS == LHS

var – переменная, по которой решается уравнение

init_param – начальное приближение для нахождение нужного корня, *необязательный параметр*

r – возвращаемое значение – числовое значение решения

Если уравнение является полиномом – находит все корни!

Численное решение уравнений, записанных в символьном виде

<table>
<tr><td>Ф</td><td>

```
syms x
S = vpasolve(x^6 - x^2 == 3, x)
```

</td></tr>
<tr><td>КС</td><td>

```
S =

                                 -1.2929423350084724369196550436382
                                  1.2929423350084724369196550436382
 - 0.50188125716943915856832436499602 - 1.0429452224956770037495194222175i
 - 0.50188125716943915856832436499602 + 1.0429452224956770037495194222175i
   0.50188125716943915856832436499602 - 1.0429452224956770037495194222175i
   0.50188125716943915856832436499602 + 1.0429452224956770037495194222175i
```

</td></tr>
</table>

Численное решение уравнений, записанных в символьном виде (3 аргумента)

Ф	syms x
	eqnLeft = 200*sin(x);
	eqnRight = x^3 - 1;
	fplot([eqnLeft eqnRight])
	title([texlabel(eqnLeft) ' = ' texlabel(eqnRight)])
	S1 = vpasolve(eqnLeft == eqnRight, x)
	S2 = vpasolve(eqnLeft == eqnRight, x, -3)
	S3 = vpasolve(eqnLeft == eqnRight, x, 4)

КС	S1 = -0.0050000214585835715725440675982988 S2 = -3.0009954677086430679926572924945 S3 = 3.0098746383859522384063444361906

Численное решение уравнений, записанных в символьном виде (график)