ПРИМЕРИ НА РЕШАВАНЕ НА НЕЛИНЕЙНИ УРАВНЕНИЯ И СИСТЕМИ

За решаване на нелинейни уравнения най-често се използва вградената функция fzero на Matlab, която се извиква най-често по следния начин

```
x = fzero(@fun,x0,options),
```

където във функцията fun се описва самото уравнение $f(x)=0$. Тя може да бъде файл-функция, инлайн-функция, ако е достатъчно „проста“ от вида fun=@(x) … или символична функция. Тук x0 е начално приближение, което се задава от физични или други съображения. Options не винаги са задължителни, а се използват когато искаме да променим настройките по подразбиране. Вижте help за повече подробности.

Първата стъпка е да локализираме корените.

Най-лесно това може да се направи *графически*:

Изчертаваме графиката за някакъв диапазаон на независимата променлива и „на око“ определяме корена, чиято стойност задаваме като начално приближение x0:

```
X=a:h:b;  Y=fun(X);        %тук а е началото на интервала по х, b края му, h -  стъпката
plot(X,Y); grid on
```

Втората стъпка е уточняването на корена, т.е решаването на самото уравнение с нужната точност чрез функцията fzero(…).

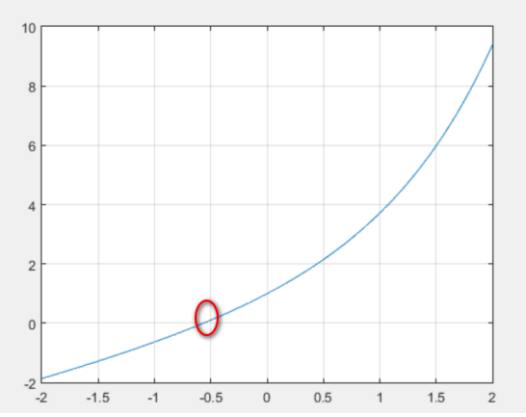
Нека е дадено уравнението $x+e^x=0$**.**

Съставяме файл-функция fun.m

```
function f = fun( x )
%уравнението
f=x+exp(x);
end
```

В CommandWindow създаваме масивте X и У

```
>> a=-2;b=2;
>> a=-2;b=2;h=(b-a)/100;
>> X=a:h:b; Y=fun(X);
>> plot(X,Y);grid on;
```



Виждаме, че корена е близо до -0.5

```
>> x0=-0.5;
>> x=fzero(@fun,x0)

x =

   -0.5671
```

Това се приема за търсеното решение.

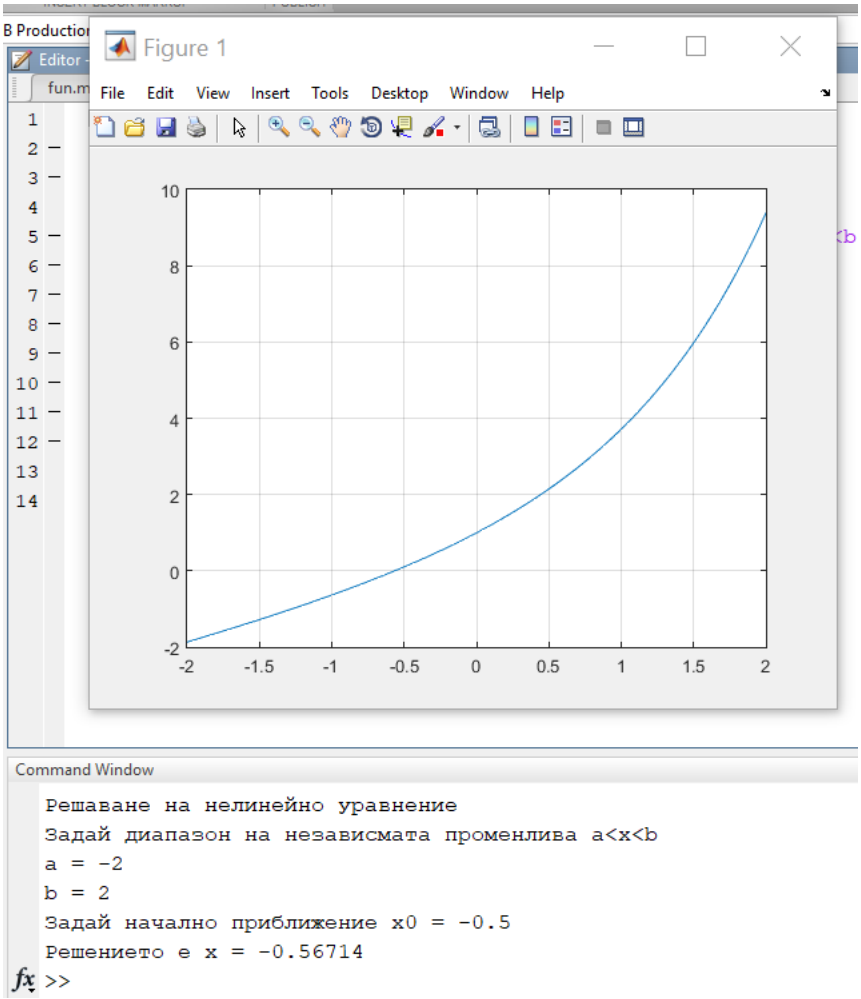
Предлагаме да създадете файл-функция myfun, в която да автоматизирате горните действия.

Ето примерен код на подобна функция:

```
function  myfun
clc;
display(['Решаване на нелинейно уравнение']);
% Построяваме графика за да локализираме корени
display(['Задай диапазон на независмата променлива a<x<b']);
a=input('a = ');b=input('b = ');h=(b-a)/100;
x=a:h:b; y=fun(x); %линейно множество x, y
plot(x,y);grid on; %визуално определяме корена
x0=input('Задай начално приближение x0 = ');
x1=fzero(@fun,x0);
display(['Решението е x = ',num2str(x1)]);
end
```

Извикване в Command Window и изглед от изпълнението й

>>myfun



Нелинейни уравнения могат да се решават и чрез вградената функция fsolve(…).

СИСТЕМИ ОТ НЕЛИНЕЙНИ УРАВНЕНИЯ

Най-общо можем да ги запишем в следния вид:

$$f_1(x_1, x_2, \cdots, x_n) = 0$$
$$f_2(x_1, x_2, \cdots, x_n) = 0$$
...
$$f_n(x_1, x_2, \cdots, x_n) = 0$$

Системите могат да имат единствено решение, много решения или да нямат решение.

В Matlab най-често използваната функция за решаването им е fsolve(…). Извикването в Command Window или във ваша функция става по следния начин:

[x,fval] = fsolve(@fun,x0, options)

Тук във функцията fun са описани функционалните зависимости $f_1, f_2, \cdots, f_n$ и тя връща вектор стълб f с изчислените стойности за вектора x, а x0 е вектора с начал $3x_1 + x_2 - 1$ ните приближения, зададени по усмотрение на потребителя. Параметър options не е задължителен, но се използва, за отпечатване номера на итерацията, стойността на функцията, промяна на точността и др. Ако не е зададен се използва подразбиращия се.

Стъпка първа: Задават се началните приближения (вектор x0) от физични или инженерни съображения. *Само понякога* при функции на две променливи могат да се изразят графически и проекциите им на точките на пресичане показват приблизителната стойност на корените.

Стъпка втора: Търсят се корените посредством [x,fval] = fsolve(@fun,x0, options), като в x се получава вектора на решенията, а fval е стойността на функцията за тези решения.

Ще покажем горното върху един конкретен пример:

$$x_1^2 + x_2^2 - 16 = 0$$
$$3x_1 + x_2 - 1 = 0$$

Създаваме файл-функция fun.m

```
function f = fun( x )
%f - вектор стълб [f1;f2;...;fn]
f=[x(1).^2+x(2).^2-16;
    3*x(1)+x(2)-1];
End
```

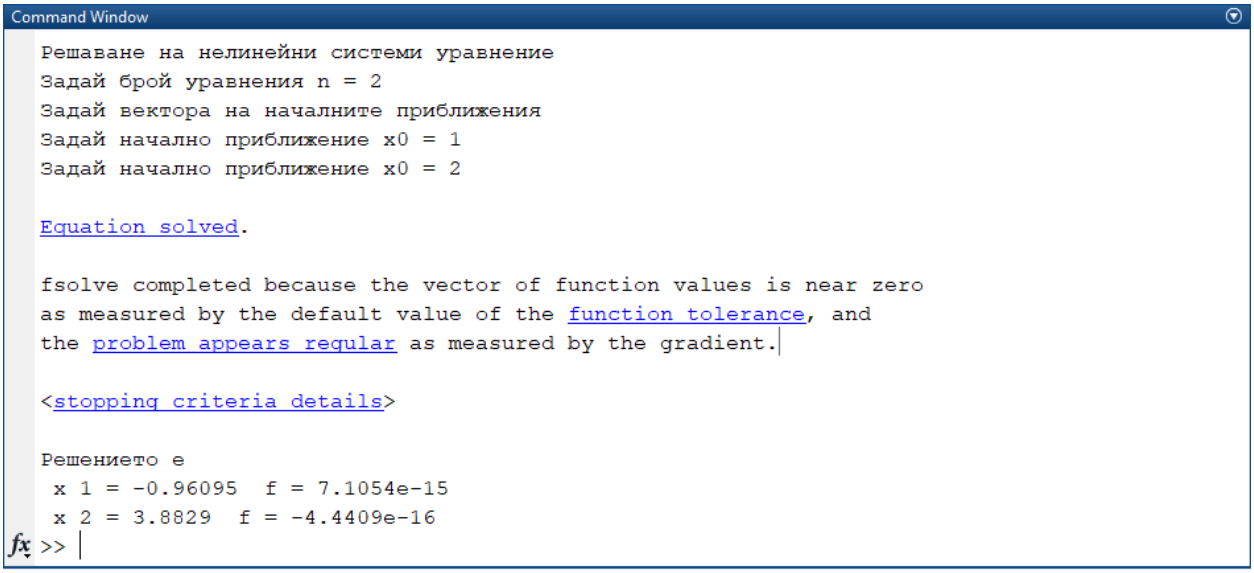
Създаваме файл-функция , в която се задават началните приближения и се отпечатват резултатите.

```
function  myfun
clc;
display(['Решаване на нелинейни системи уравнение']);
% Построяваме графика за да локализираме корени
n=input('Задай брой уравнения n = ');
display(['Задай вектора на началните приближения ']);
for i=1:n
x0(i)=input('Задай начално приближение x0 = ');
end
```

```
[x1,fval]=fsolve(@fun,x0);
display(['Решението е']);
for i=1:n
display([' x ',num2str(i),' = ',num2str(x1(i)),'  f =
',num2str(fval(i))]);
end
end
```

Извикване в Command Window

>>myfun



Задачата има и второ решение. При друго начално приближение x0=[1, - 4] се получава

>>myfun