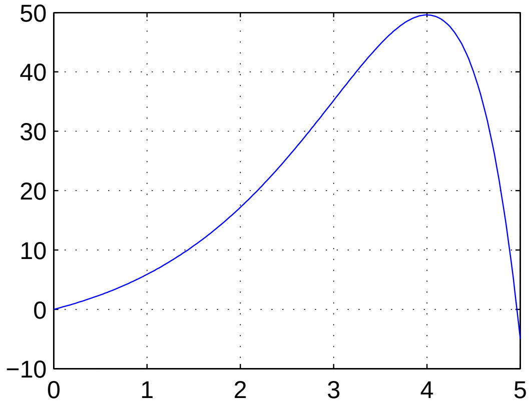
Exempel 3. Gör en **M-File Function** som löser ekvationen $f(x) = 0$ genom intervallhalvering. Funktionen skall som indata ges en funktion som beräknar $f(x)$, ett intervall $[a\ b]$ där f växlar tecken, dvs. där $f(a) \cdot f(b) < 0$, samt den noggrannhet roten skall bestämmas med.

```
function m=bisect(f,intervall,tol)

a=intervall(1);
b=intervall(2);
fa=f(a);
fb=f(b);

if fa*fb>0
   disp('Ingen teckenväxling i intervallet!')
   m=[];
   return
end

while b-a > tol
   m=(a+b)/2;
   fm=f(m);
   if fm==0
      return
   end
   if fa*fm<0
      b=m;
      fb=fm;
   else
      a=m;
      fa=fm;
   end
end

m=(a+b)/2;
```

Exempel 4. Vi skall beräkna minsta positiva roten till $f(x) = e^x(5-x) - 5 = 0$ med intervallhalvering.

```
>> x=linspace(0,5);
>> y=exp(x).*(5-x)-5;
>> plot(x,y)
>> grid on
```



Ser att positiva roten ligger i intervallet $4.5 \leq x \leq 5$.

```
>> a=4.5; b=5;
>> x=bisect(@(x)exp(x)*(5-x)-5,[a b],1e-10)
x =
   4.965114231716143
```

Här använde vi ett s.k. **anonymt funktionshantag**.

Alternativt skapar vi en funktionsfil

```
function y=funex4(x)
y=exp(x).*(5-x)-5;
```

och vidare

```
>> x=linspace(0,5);
>> plot(x,funex4(x))
>> grid on

>> a=4.5; b=5;
>> x=bisect(@funex4,[a b],1e-10)
x =
   4.965114231716143
```

Exempel 5. Med funktionen **fzero** löser vi ekvationen från exempel 4 enligt

```
>> x=fzero(@funex4,[4.5 5])
x =
   4.965114231744276
```

Men det går också med alternativen

```
>> x=fzero('(5-x)*exp(x)-5',[4.5 5])
>> x=fzero(@(x)(5-x)*exp(x)-5,[4.5 5])
```

Givetvis med samma resultat.