Bisekcia a Newtonova metóda

<u>Úloha</u>

Sú dané funkcie:

g1 = sin(x^5-x^4+3)
g2 = exp(x^2-a), kde *a* je parameter.

Nájdite parameter *a* tak, aby rovnica g1(x)=g2(x) mala aspoň jedno dvojnásobné riešenie.

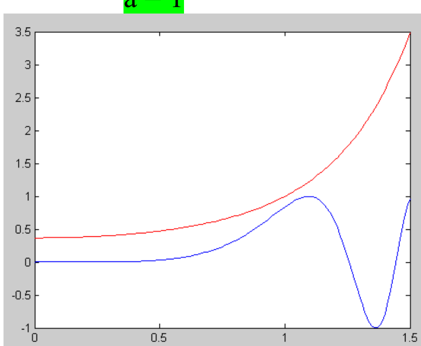
<u>Riešenie</u>

Na začiatok sa zorientujeme v situácii pomocou obrázku. Aby sme nezdržovali, uvádzame dva obrázky, ktoré sú už výsledkom dlhšieho skúšania. Zaujímavé sú a=1 a a=2, pričom sa s x-om obmedzíme na interval 0 až 1.5.

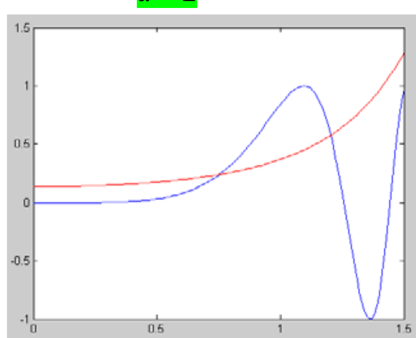
```
>> g1=inline('sin(x.^5)'); g2=inline('exp(x.^2-1)');
>> x=0:0.01:1.5;
>> plot(x, g1(x)), hold on, plot(x, g2(x),'r');

******
>> g2=inline('exp(x.^2-2)');
>> plot(x, g1(x)), hold on, plot(x, g2(x),'r')
```

a = 1



a = 2

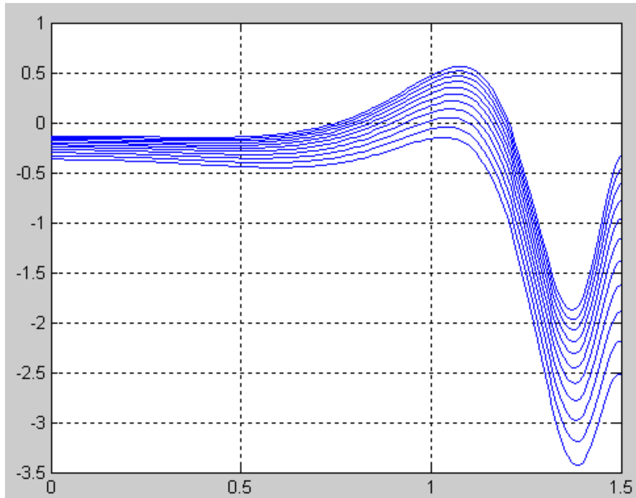


Riešením bude *a* z intervalu [1, 2], pričom hodnotu *x* možno predbežne očakávať v [0.5, 1.5]. Dvojnásobný koreň *x* je číslo, v ktorom funkcia aj jej derivácia majú hodnotu nula. Budeme preto pre rôzne *a* hľadať korene f1 (derivácie f) a overovať funkčnú hodnotu f v nich.

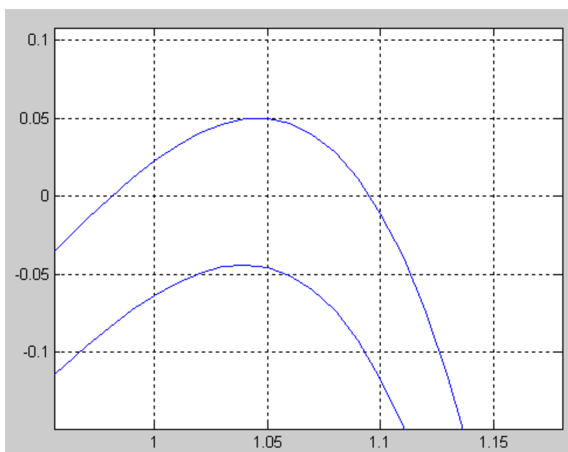
```
» f=inline('sin(x.^5)-exp(x.^2-a)')
» f1=inline('cos(x.^5).*(5*x.^4)-2*exp(x.^2-a).*x')
» f2=inline('-sin(x.^5).*(5*x.^4).^2+cos(x.^5).*(20*x.^3)-4*exp(x.^2-a).*x.^2-2*exp(x.^2-a)')
```

Spresníme situáciu pomocou obrázkov. Začnime grafmi funkcie f pre rôzne a:

```
>> for a=1:0.1:2, y=f(a,x);  plot(x,y), hold on, end, grid on
```



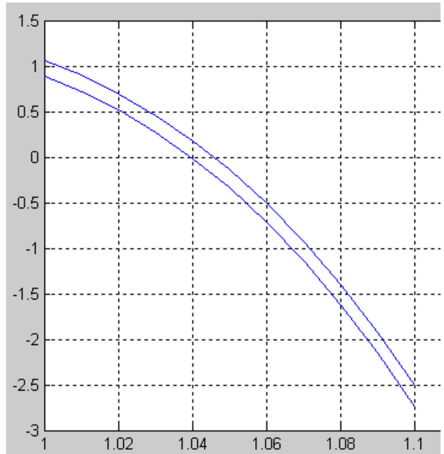
Jednotlivé krivky ***zdola hore*** zodpovedajú hodnotám a=1 až 2. Evidentne nami hľadané *a* sa nachádza medzi hodnotami 1,1 a 1,2. Hľadané dvojnásobné riešenie na základe obrázku odhadujeme v rozmedzí 1 až 1.1 – zväčšime si túto časť obrázku:



Z priebehu funkcie vidíme, že f1 bude kladná vľavo od hľadaného koreňa a záporná vpravo od neho a bude klesať. Funkcia f je konkávna a tak f2 bude na celom úseku záporná. Sme si vedomí, že takáto analýza „od oka“ z obrázku môže byť zradná, ale analytický rozbor prvej a zvlášť druhej derivácie by bol v tomto prípade trochu náročnejší.

Nakreslíme si prvú deriváciu:

```
>> x=1:0.01:1.1;
>> for a=1.1:0.1:1.2, y= f1(a,x); plot(x,y), hold on, end
>> grid on
```



Priebeh prvej derivácie f1 je podľa všetkého klesajúci a konkávny, f2 bude teda záporné a f3 tiež, koreň f1 preto budeme Newtonovou metódou hľadať „sprava“.

Preskúmajme približne f2 kvôli univerzálnemu odhadu chyby. Pre a=1.1 a 1.2 je f2 na pohľad klesajúca, pričom hodnoty k nule sú najbližšie pre x=1,2 a to v x=1. Preto máme m=15.143 (minimum abs. hodnoty f1 na intervale).

Nájdeme predbežne koreň x funkcie f1 pre a=1.1 a 1.2 s presnosťou 1e-14.

```
>> a=1.1; x=1.1;
» while(abs(f1(a,x)/15.143) > 1e-14),x=x-f1(a,x)/f2(a,x);end, x, y=f(a,x)

        x =   1.03955204576188,   y =  -0.04382131725516

» while(abs(f1(a,x)/15.143) > 1e-14), x=x-f1(a,x)/f2(a,x); end, x, y=f(a,x);

        x =   1.04579351529284,  y =   0.05013393956937
```

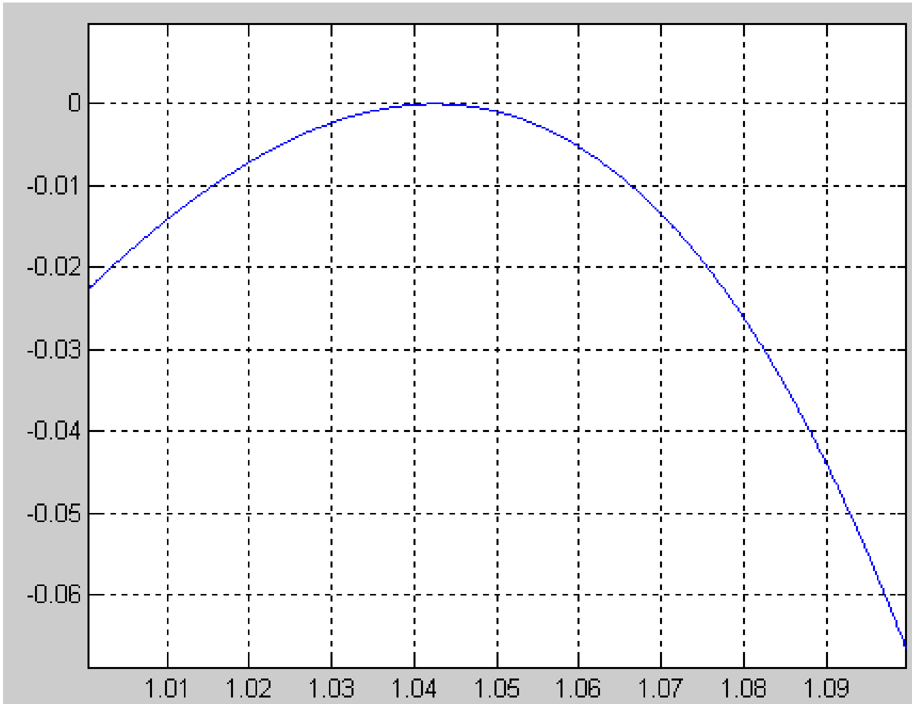
Takže, pre a=1.1 je y záporné, pre a=1.2 kladné. My chceme nulu s presnosťou 1e-14, spustíme preto bisekciu na intervale [1.1, 1.2] a to 44 krokov (overte, či to stačí).

```
>> x=1.1; L=1.1; P=1.2;
» for i=1:44, a=(L+P)/2; while (abs(f1(a,x)/15.143) > 1e-14),
    x =x-f1(a,x)/f2(a,x);end,y=f(a,x);
  if y<0, L=a; else P=a; end, end, x, a

                x =    1.04251269963853
                a =   1.14556095703886
```

Takže pre a =1.14556095703886 má funkcia f dvojnásobný koreň x = 1.04251269963853.
Overíme:

```
>> x=1:0.0001:1.1; a=1.14556095703886; y=f(a,x); plot(x,y), hold on, grid on
```


0
-0.01
-0.02
-0.03
-0.04
-0.05
-0.06
1.01
1.02
1.03
1.04
1.05
1.06
1.07
1.08
1.09