

Príklad – na hraniciach možného

Budeme hľadať korene rovnice $f(x) = e^x - e*x - 0.0001$. Vyskúšame najprv orientačne možnosti, ktoré ponúka MatLab:

```
» f=inline('exp(x)-exp(1)*x-0.0001')
```

Tipneme si štartovací bod: `fzero(x, 0)`, `fzero(x, 2)`, `fzero(x, 70)`, ... skúste aj iné hodnoty. Odpoveď MatLabu je vo väčšine prípadov vyhubavá. Má to vôbec nejaký koreň?

A. Lokalizácia možných koreňov a opätovný pokus s fzero:

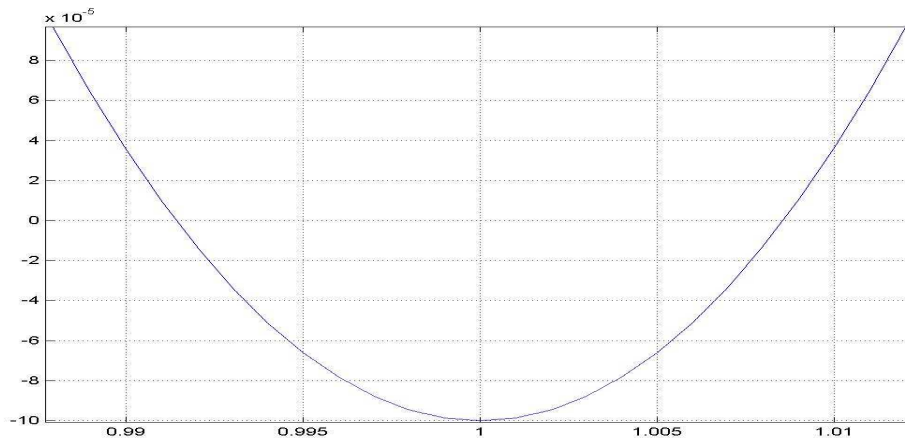
Nepochybne platí, že pre $x \geq 2$ je $\exp(x) - \exp(1)*x - 0.0001 > 0$

a podobne pre $x < 0$ je $\exp(x) - \exp(1)*x - 0.0001 > 0$

Ubezpečte sa o tom... Stačí nám preto nakresliť si graf od 0 do 2:

```
» x=0:0.001:2; plot(x,f(x)), grid on
```

Pekný obrázok, ale to dôležité poriadne nevidíme – musíme si to zväčšiť:



Vidíme, že funkcia má dva korene a sú na intervaloch $[0.99, 0.995]$, $[1.005, 1.01]$. Intervaly sú malé a blízko seba, pri zbežnom pohľade zďaleka to vyzeralo na dvojnásobný koreň. V tom je príčina zlyhania nášho zbežného pokusu s `fzero`. Je to síce silná zbraň, ale je nám nanič, ak nemierime presne. Napravíme to – `fzero` je oveľa šikovnejšie, ak mu miesto odhadu koreňa zadáme interval, v ktorom ten koreň určite je:

```
» fzero(f,[0.99,0.995])
```

Zero found in the interval: [0.99, 0.995]. ans = 9.914100809515262e-001

```
» fzero(f,[1.005, 1.01])
```

Zero found in the interval: [1.005, 1.01]. ans = 1.008565393712301e+000

Úloha I: Vráťte sa k obrázku a napíšte trochu iné hranice intervalov, obsahujúcich korene. Dosad'te do fzero – ako sa zmení výsledok?

Odhad chyby získaných výsledkov.

1. skusmo: Na základe pokusov v úlohe I sa zdá, že vo výsledku $x_1=9.914100809515262e-001$ sa nedá spoľahnúť na posledné tri cifry a pri $x_2=1.008565393712301$ je to podobné. Odhadneme preto $e_1=e_2=1e-14$:

» $e_1=1e-14$; $f(x_1-e_1)$, $f(x_1+e_1)$

ans = 2.110264003471474e-016 ans = -2.330628095029153e-016

» $e_2=1e-14$; $f(x_2-e_2)$, $f(x_2+e_2)$

ans = -2.330628095029153e-016 ans = 2.110264003471474e-016

Odhad je správny. Skusmo nájdite menšie (presnejšie) e_1 , e_2 .

2. Univerzálny odhad chyby:

Derivácia f : $f_1=\text{inline}(\text{'exp}(x)-\text{exp}(1)\text{'})$ je rastúca na celom definičnom obore. Minimum $|f_1|$ na intervale preto nájdeme len porovnaním hodnôt na okraji:

Na intervale $[0.99,0.995]$:

$f_1(0.99) = -2.704735610978304e-002$
 $f_1(0.995) = -1.355748717959315e-002$
 $mx_1 = 1.355748717959315e-002$

Na intervale $[1.005, 1.01]$:

$f_1(1.005) = 1.362544436688129e-002$
 $f_1(1.01) = 2.731918655787080e-002$
 $mx_2 = 1.362544436688129e-002$

Odhady chyby nájdených koreňov sú potom:

$e_1 = f(x_1)/mx_1 = 1.5565e-014$
 $e_2 = f(x_2)/mx_2 = 1.5488e-014$

Naše odhady skusmo boli dokonca lepšie...

B. Riešenie Newtonovou metódou

Hoci už výsledok vieme, budeme úlohu riešiť opäť. Cieľom je nielen precvičenie Newtonovej metódy bezpečným spôsobom (=vieme, čo má vyjsť), ale aj ambícia spresniť výsledky. Využijeme už získaný predpis derivácie f_1 a to, že na našich intervaloch nemení znamienko. Podobne druhá derivácia $f_2=\text{exp}(x)$ znamienko nemení, lebo je všade kladná.

x_1 : » $x=0.99$;
 » $x=x-f(x)/f_1(x)$ – tento príkaz opakujte, kým sa výsledky neustália...

9.914095197613563e-001
9.914100809332571e-001
9.914100809515074e-001
9.914100809515356e-001
9.914100809515256e-001
9.914100809515156e-001
9.914100809515247e-001
9.914100809515338e-001
9.914100809515238e-001
9.914100809515329e-001
9.914100809515229e-001
9.914100809515130e-001
9.914100809515412e-001
9.914100809515121e-001
9.914100809515212e-001
9.914100809515112e-001

Toto sa evidentne nechce ustáliť. $x_1=9.914100809515???$ e-001 Podobne pri výpočte x_2 zistíme $x_2=1.008565393712???$ e000

Hoci Newtonova metóda speje rýchlo k presnému výsledku, náš výpočet určitú hranicu neprekročí – narazil na medze presnosti MatLabu. Skúsme aspoň približne pochopiť, čo sa deje. Číslo x v iteráciách Newtonovej metódy sa zobrazí v MatLabe s presnosťou cca. $5e-17$. Po dosadení do f je najcitlivejší člen e^*x , kde sa chyba x prejaví ako $e*5e-17=cca. 1.36e-16$. Po vydelení deriváciou f_1 , ktorej hodnoty sú na našom intervale menšie než $-1.35e-002$, dostávame člen $f(x)/f_1(x)$ s presnosťou najviac $1e-14$. A toto je číslo, ktorým sa má akože spresňovať každá ďalšia iterácia x ?! Samotný vzorec Newtonovej metódy má teda v MatLabe ohraničenú presnosť na cca. $1e-14$, čo znamená, že Newtonovou metódou po ľubovoľnom počte iterácii oveľa vyššiu presnosť nedosiahneme.

C. Bisekcia – pomalá, ale spoľahlivá. Napokon sa ešte zide metóda najpomalšia, ale najmenej obmedzená. Aby sme nemali veľa práce, využijeme všetko to, čo už vieme. Na základe výpisu iterácií Newtonovej metódy vyberieme nádejné hodnoty štartovacieho intervalu

`xx=[9.914100809515e-001, 9.914100809516e-001]`

Overme si, že koreň tu je:

`» f(xx)`
`ans = 2.110264003471474e-016 -1.565330439053103e-015`

Je tam. Spustíme bisekciu. Krokov nebude veľa, stačí ručný postup z promptu. Budeme opakovať nasledujúci postup, v ktorom si necháme vypisovať poradie, hodnotu a presnosť jednotlivých iterácií:

`» i=0;`
`» i=i+1; s=(xx(2)+xx(1))/2; if f(s)>0, xx(1)=s;end, if f(s)<0, xx(2)=s;end, [i,s, (xx(2)-xx(1))]`

```
ans = 1.000000000000000e+000 9.914100809515500e-001 5.007105841059456e-014
2.000000000000000e+000 9.914100809515249e-001 2.498001805406602e-014
3.000000000000000e+000 9.914100809515125e-001 1.254552017826427e-014
4.000000000000000e+000 9.914100809515063e-001 6.217248937900877e-015
5.000000000000000e+000 9.914100809515094e-001 3.108624468950438e-015
6.000000000000000e+000 9.914100809515110e-001 1.554312234475219e-015
7.000000000000000e+000 9.914100809515117e-001 7.771561172376096e-016
8.000000000000000e+000 9.914100809515121e-001 4.440892098500626e-016
9.000000000000000e+000 9.914100809515123e-001 2.220446049250313e-016
1.000000000000000e+001 9.914100809515124e-001 1.110223024625157e-016
1.100000000000000e+001 9.914100809515125e-001 1.110223024625157e-016
1.200000000000000e+001 9.914100809515125e-001 1.110223024625157e-016
```

Po 11 krokoch sme na konci. Ďalšie opakovanie už výsledky nezmení (nové s sa zaokrúhľuje na xx(2)). Interval sme zmenšili na

xx = 9.914100809515124e-001 9.914100809515125e-001
a platí f(xx)= 2.110264003471474e-016 -2.330628095029153e-016

Hľadaný výsledok je kdesi medzi xx(1) a xx(2) – presnejšie to už MatLab neprezradí. Odhadneme výsledok číslom (pre ML už nestráviteľným)

xs=0.99141008095151245
a jeho chybu určíme ako 5e-17. Lepšie to už s ML nejde.

Úloha: Podobným spôsobom zistite čo najpresnejšie druhý koreň funkcie f.
(1.0085653937122942)