1.6 Úlohy (kapitola 1, str. 40; tu sú zadania, riešenia nasledujú)

1.6.1 Uvažujme fyzikálne kyvadlo so závažím o hmotnosti *m* na závese dĺžky *l*. Chceme získať výchýlku φ ako funkciu času, φ = φ(t). Napíšte diferenciálnu rovnicu, ktorá modeluje kyvadlo, (resp. jeho výchylku), keď neuvažujeme trenie v závese, a pritom

a) odpor prostredia modelujeme (zjednodušene) silou úmernou rýchlosti, F = c.ds/dt, kde c je koeficient odporu. Výchylku φ v čase 0 označme φ_0, teda $\varphi_0 = \varphi(0)$. Nech φ'(0) = 0.
b) Teraz diferenciálnu rovnicu z bodu a) zjednodušme, keď namiesto sin(φ) píšme φ. Rovnica sa stane lineárnou diferenciálnou rovnicou druhého rádu. Viete takúto rovnicu riešiť?
c) Diferenciálnu rovnicu z bodu b) zjednodušme neuvažovaním odporu prostredia. Rovnica má tvar:

$$\varphi'' + (\frac{g}{l})\varphi = 0, \ \ \varphi(0)=\varphi_0,\, \varphi'(0)=0$$

Overte, že riešením je funkcia $\varphi = \varphi_0 \cos(\sqrt{\frac{g}{l}}\, t)$.

1.6.2 Stirlingov vzorec je vzťah, ktorý dáva nasledujúcu aproximáciu funkcie faktoriál:

$$n! \approx \sqrt{2\pi n}\,.\left(\frac{n}{e}\right)^n$$

V MATLABe tabelujte závislosť absolútnej aj relatívnej chyby aproximácie od n, pre n = 2:2:20.

1.6.3 Nech w = $\sqrt{x}yz^2$, kde x, y, z sú neúplné čísla, x = 4.0 ± 0.5, y = 2.0 ± 0.2, z = 1.0 ± 0.1. Určte interval, v ktorom leží neznáma hodnota w, resp. nájdite w* a ε, aby sme mohli písať: w = w* ± ε .

1.6.4 Využitím výpočtových možností ML určte R* a ε(R*), keď R = $(R_1R_2)/(R_1 + R_2)$, pričom R_1 aj R_2 sú dané v tvare neúplných čísel: $R_1 = 10 \pm 0.5$, $R_2 = 20 \pm 0.8$.

1.6.5 Realizujte výpočet aproximácií S_1, S_2 súčtu radu $\sum_{n=1}^{\infty}\frac{1}{2n-1}$ takto: S_1 nech vznikne ako súčet prvých 100 členov radu, kým S_2 nech vznikne postupným načítavaním členov radu dovtedy, kým člen radu je väčší ako 0,005. Porovnajte S_1 a S_2. Ktorá z nich je lepšou aproximáciou súčtu radu? Čo si myslíte o korektnosti tejto úlohy?

1.6.6 Riešte úlohu 1.6.5, ale tentoraz nech ide o konvergentný rad: $\sum_{n=1}^{\infty}\frac{1}{n^2}$

1.6.7 Uvažujme počítačovú množinu čísel F = F(10, 3, -9, 9). Určte:
a) hodnotu najmenšieho kladného čísla množiny F
b) hodnotu najväčšieho čísla množiny F
c) vzdialenosti medzi číslom 1 a jeho susedmi
d) vzdialenosť medzi susedmi čísla 1000
e) najmenšiu vzdialenosť medzi prvkami F
f) najväčšiu vzdialenosť medzi prvkami F

1.6.8 V počítačovej množine čísel systému MATLAB určte vzdialenosti medzi bezprostrednými susedmi čísel: 1, 8, 64, 4096.

1.6.9 Nasledujúce funkcie linearizujte v zadaných bodoch x_0. Jednotlivé lineárne funkcie L(x) berme ako lokálne aproximácie odpovedajúcich funkcií f(x). Vyčíslite chybu | f(u) – L(u) | pre u = x_0 + 0.2 .

a) cos(x), x_0 = 3.pi/4 b) $\sin(x^2)$, x_0 = pi/4 c) $f(x) = x/(x^2 + 1)$, x_0 = 1.5

1.6.10 Použitím M-funkcie "conv" a konečnými Taylorovými radmi funkcií exp(-x), cos(x), aproximujte polynómom p(x) funkciu f(x) = exp(-x).cos(x). Tabelujte diferenciu
| exp(-x).cos(x) – p(x) | pre x ∈ [-0.5, 0.5] s krokom 0.1

1.6.11 Zostrojte graf kvantilovej funkcie rozdelenia N(0,1) využitím aproximácie ψ kvantilovej funkcie, ktorá je daná takto:

a0 = 2.515517, a1 = 0.802853, a2 = 0.010328, b1 = 1.432788, b2 = 0.189269, b3 = 0.001308

w = sqrt(-2.ln(y)), keď pre y∈(0, 0.5] platí:

$$\psi(y) = -\ w\ +\ (a0 + a1.w + a2.w^2)/(1 + b1.w + b2.w^2 + b3.w^3),$$

a pre y∈(0.5, 1) platí:

$$\psi(y) = -\psi(1-y)$$

1.6.1 Uvažujme fyzikálne kyvadlo so závažím o hmotnosti m na závese dĺžky l. Chceme získať výchýlku φ ako funkciu času, φ = φ(t). Napíšte diferenciálnu rovnicu, ktorá modeluje kyvadlo (resp. jeho výchylku), keď neuvažujeme trenie v závese, a pritom

a) odpor prostredia modelujeme (zjednodušene) silou úmernou rýchlosti, F = c.ds/dt, kde c je koeficient odporu. Výchylku φ v čase 0 označme φ_0, teda $\varphi_0 = \varphi(0)$. Nech φ'(0) = 0.

b) Teraz diferenciálnu rovnicu z bodu a) zjednodušme, keď namiesto sin(φ) píšme φ. Rovnica sa stane lineárnou diferenciálnou rovnicou druhého rádu. Viete takúto rovnicu riešiť?

c) Diferenciálnu rovnicu z bodu b) zjednodušme neuvažovaním odporu prostredia. Rovnica má tvar:

$$\varphi'' + (\frac{g}{l})\varphi = 0,\ \ \varphi(0)=\varphi_0,\ \varphi'(0)=0$$

Overte, že riešením je funkcia $\varphi = \varphi_0 \cos(\sqrt{\frac{g}{l}}\,t)$.

Riešenie. a) F = $-\,m.g.\sin(\varphi) - c.l.d\varphi/dt$ (uvažovanie gravitačnej sily a odporu prostredia). Newtonov pohybový zákon dáva vzťah:

$$m.a = \mathrm{F},\quad \text{t.j.}\quad m.l.d^2\varphi/dt^2 = \mathrm{F},$$

do ktorého dosaďme za F. Po krátení súčinom $m.l$ dostávame diferenciálnu rovnicu:

$$\frac{d^2\varphi}{dt^2} + \frac{c}{m}\frac{d\varphi}{dt} + \frac{g}{l}\sin(\varphi) = 0$$

Začiatočné podmienky formulujme takto: $\varphi(0) = \varphi_0$, $(d\varphi/dt)(0) = 0$. Túto diferenciálnu rovnicu nevieme analyticky riešiť. Ak ale začiatočná výchylka je malá, môžeme akceptovať náhradu sin(φ) = φ a rovnica sa zmení na lineárnu diferenciálnu rovnicu 2. rádu, ktorú už vieme riešiť (dokonca viacerými postupmi).

Zmysel tejto úlohy je na jednoduchej situácii ukázať modelovanie diferenciálnou rovnicou a jej riešenie vidieť ako niečo, čo má v reálnej situácii zrejmú interpretáciu. Pri tejto príležitosti zopakovať, ČO je riešenie diferenciálnej rovnice (so začiatočnou podmienkou).

Obmena úlohy: Napíšte diferenciálnu rovnicu voľného pádu, keď odpor prostredia modelujeme zase len ako c.ds/dt. Poznamenávame, že takéto modelovanie odporu prostredia je vskutku problematické a robíme to takto len preto, aby sme získali relatívne jednoduché modely.

1.6.2 Stirlingov vzorec je vzťah, ktorý dáva nasledujúcu aproximáciu funkcie faktoriál:

$$n! \approx \sqrt{2\pi n}\,.\left(\frac{n}{e}\right)^n$$

V MATLABe tabelujte závislosť absolútnej aj relatívnej chyby aproximácie od n, pre n = 2:2:20.

Riešenie.

Najprv zvládnime tabeláciu: $n \longrightarrow n!$ Funkcia faktoriál je v súčasných verziách MATLABu dostupná pod menom "factorial" . My použijeme funkciu "prod", t.j. postupujeme tak, ako museli postupovať užívatelia starších verzií MATLABu: n! = prod(1:n).

```
n = (2:2:20)'; for i = 1:10, nfact(i) = prod(1:n(i)); end, nfact = nfact';
mat = [n, nfact]; fprintf(' %2d %22.0f \n', mat')
```

```
  2                     2
  4                    24
  6                   720
  8                 40320
 10               3628800
 12             479001600
 14           87178291200
 16        20922789888000
 18      6402373705728000
 20   2432902008176640000
```

Teraz pridáme stĺpec aproximácií faktoriálu. Vektorizovanie je trochu riskantné, ale vyskúšajme, čo robí naše prostredie pri vektorizovaní funkcie x^x

```
x = (1:4)'; [x, (x.^x)]
ans =
     1     1
     2     4
     3    27
     4   256
```

OK, funguje to! Preto:

```
aprox = (sqrt(2*pi*n).*(n./exp(1)).^n);
mat2 = [n, aprox]; fprintf(' %2d  %22.0f \n', (mat2)')
  2                      2
  4                     24
  6                    710
  8                  39902
 10                3598696
 12              475687486
 14            86661001741
 16         20814114415223
 18       6372804626194298
 20    2422786846761129000
```

Tabelujme to, čo zatiaľ máme:

```
mat3 =[n, nfact, aprox]; fprintf(' %2d  %22.0f  %22.0f \n', (mat3)')
  2                      2                       2
  4                     24                      24
  6                    720                     710
  8                  40320                   39902
 10                3628800                 3598696
 12              479001600               475687486
 14            87178291200             86661001741
 16         20922789888000          20814114415223
 18       6402373705728000        6372804626194298
 20    2432902008176640000     2422786846761129000
```

Na vyriešenie úlohy však treba tabelovať rozdiely, pretože máme tabelovať absolútnu chybu aproximácie. Presnejšie, potrebujeme absolútne hodnoty rozdielov a v prípade relatívnej chyby, je treba získať podiely.

```
abschyby = abs(nfact - aprox); relchyby = abschyby./nfact;
mat4 = [n, abschyby, relchyby]; fprintf(' %2d  %22.0f  %22.9f \n', (mat4)')
  2                      0             0.040497824
  4                      0             0.020576036
  6                     10             0.013780299
  8                    418             0.010357256
 10                  30104             0.008295960
 12                3314114             0.006918794
 14              517289459             0.005933696
 16           108675472777             0.005194120
 18         29569079533702             0.004618456
 20      10115161415511040             0.004157653
```

1.6.3 Nech $w = \sqrt{xyz^2}$, kde x, y, z sú neúplné čísla, $x = 4.0 \pm 0.5$, $y = 2.0 \pm 0.2$, $z = 1.0 \pm 0.1$. Určte interval, v ktorom leží neznáma hodnota w, resp. nájdite w^* a ε, aby sme mohli písať: $w = w^* \pm \varepsilon$.

Riešenie.

Ak výraz $\sqrt{xyz^2}$ chápeme ako funkciu, vidíme, že ide o funkciu, ktorá je rastúcou vzhľadom na každý zo svojich troch argumentov. Preto jednoducho dokážeme nájsť minimálnu a tiež aj maximálnu hodnotu, ktorú uvažovaná funkcia nadobudne na danej oblasti (oblasťou je kváder so stranami 1, 0.4 a 0.2). Ak získame w_{min}, w_{max}, tak za aproximáciu neznámej hodnoty vezmeme stred intervalu $[w_{min}, w_{max}]$ a za ε vezmeme polomer nájdeného intervalu. Výsledok získame nasledovnými príkazmi:

```
wmin = sqrt(3.5)*1.8*(0.9)^2; wmax = sqrt(4.5)*2.2*(1.1)^2;
interval = [wmin, wmax], stred = 0.5*(wmax+wmin), polomer = stred - wmin
interval =
           2.72766823495820   5.64695475455577
stred =                       polomer =
           4.18731149475699            1.45964325979878
```

Naša odpoveď na zadanie: $w = w^* \pm \varepsilon = 4.187 \pm 1.46$

1.6.4 Využitím výpočtových možností MATLABu určte R^* a $\varepsilon(R^*)$, keď $R = (R_1 R_2)/(R_1 + R_2)$, pričom R_1 aj R_2 sú dané v tvare neúplných čísel: $R_1 = 10 \pm 0.5$, $R_2 = 20 \pm 0.8$.

Riešenie. V minulosti, v rámci numerických metód, sme sa zoznamovali s postupmi, ako takúto úlohu riešiť základnými prostriedkami matematickej analýzy. Dnes musíme vystačiť s elementárnejším postupom, pričom máme možnosť využiť MATLAB. Uvážme, že teraz (na rozdiel od predchádzajúcej úlohy) uvažovaná funkcia dvoch premenných nie je (na pohľad) rastúcou funkciou vzhľadom k svojím argumentom. Avšak zaiste platí, že zdola hodnoty R ohraničuje hodnota "dolna" a zhora hodnota "horna", keď ich definujeme takto:

$$\text{dolna} = (R_{1\,min} . R_{2\,min})/(R_{1\,max} + R_{2\,max}), \quad \text{horna} = (R_{1\,max} . R_{2\,max})/(R_{1\,min} + R_{2\,min})$$

Za aproximáciu R^* vezmeme stred intervalu [dolna, horna] a za $\varepsilon(R^*)$ vezmeme polomer získaného intervalu.

```
dolna = (9.5*19.2)/(10.5 + 20.8);   horna = (10.5*20.8)/(9.5 + 19.2);
interval = [dolna, horna],
stred = 0.5*(dolna + horna), polomer = stred - dolna
interval =
          5.82747603833866   7.60975609756098
stred =
          6.71861606794982
polomer =
          0.89114002961116
```

Na základe uvedeného postupu výsledok prezentujeme v tvare $R = 6.72 \pm 0.89$. Môžeme namietať, že uvedený postup dáva len hrubý odhad pre chybu, pretože samotné hranice sme získali postupom, ktorý dal hrubé odhady (zhora, resp. zdola – veď hodnota R nemôže byť v skutočnosti dvojaká – iná v čitateli a iná v menovateli!).

Využime výpočtovú silu prostredia a v cykloch prejdime uvažovanú oblasť (ktorá je v tomto prípade obdĺžnik) a nájdime (hrubou silou) R_{min} a R_{max}. Začneme definovaním pomocných hodnôt R_{min} a R_{max} :

```
Rmin = 1000; Rmax = 0;
for r1 = 9.5:0.1:10.5
    for r2 = 19.2:0.1:20.8
         r = r1*r2/(r1 + r2);
         if  r < Rmin, Rmin = r; end
         if  r > Rmax, Rmax = r; end
    end
end
interval = [Rmin, Rmax]
interval =
           6.35540069686411   6.97763578274760
```

Pre zaujímavosť ukážme, že vektorizovaním výrazu sa zaobídeme bez cyklov "for" (uznávam, žiada to istý čas, vidieť "vektorovo" a hore uvedené cykly sú bezpochyby čitateľnejšie).

```
r1 = (9.5:0.1:10.5)';
r2 = 19.2:0.1:20.8;
Rcit = r1*r2;

R1 = r1*ones(1,length(r2));
R2 = ones(length(r1),1)*r2;
Rmen = R1 + R2;

R = Rcit./Rmen;
int = [min(min(R)), max(max(R))]
int =
     6.3554    6.9776
```

Porovnajte tieto hodnoty s tými, ktoré sme získali tým jednoduchším spôsobom (pomocou hraníc "horna" a "dolna"). S využitím týchto nových hodnôt dostávame:

```
stred = 0.5*( 6.3554 + 6.9776), polomer = stred - 6.3554
stred =
        6.66650000000000
polomer =
        0.31110000000000
```

Na základe týchto výsledkov môžeme písať: $R = 6.66 \pm 0.31$ a samozrejme, považujeme ho za lepší ako ten predchádzajúci, keď sme dostali $R = 6.72 \pm 0.89$.

1.6.4 Realizujte výpočet dvoch aproximácií S_1 a S_2 súčtu radu

$$\sum_{n=1}^{\infty} \frac{1}{2n-1}$$

takto: S_1 nech vznikne ako súčet prvých 100 členov radu.

S_2 nech vznikne postupným načítavaním členov radu dovtedy, kým člen radu je väčší ako 0,005.

Porovnajte S_1 a S_2. Ktorá z nich je lepšou aproximáciou súčtu radu? Čo si myslíte o korektnosti tejto úlohy?

Riešenie. $S_1 = 1 + 1/3 + 1/5 + \ldots + 1/199$. Súčet dostaneme napr. cyklom typu "for"

```
S1 = 0; for i = 1:100, S1 = S1 + 1/(2*i - 1); end, S1
S1 =
     3.28434218930163
```

Keďže (berme $n = 100$)

```
1/199
ans =
     0.00502512562814
```

ale (berieme teraz $n = 101$)

```
1/201
ans =
     0.00497512437811
```

tak je jasné, že nie je žiaden rozdiel medzi S1 a S2. Presvedčme sa:

```
S2 = 0; n = 1; clen = 1;
while  clen > 0.005,
       S2 = S2 + clen;
       n = n + 1;
       clen = 1/(2*n - 1);
end; S2
S2 =
     3.28434218930163
```

Keby sme však namiesto 0.005 uvažovali inú hodnotu, napr. 0.00005, tak je prirodzené očakávať, že získaný súčet bude lepšie aproximovať súčet radu. Označme teraz súčet ako S3:

```
S3 = 0; n = 1; clen = 1;
while  clen > 0.00005,
       S3 = S3 + clen;
       n = n + 1;
       clen = 1/(2*n - 1);
end; S3
S3 =
     5.58692519920711
```

Je ľahké uvážiť, že počet členov pre súčet S3 sa rovná 10000 (overte). Máme pocit, že keď počet členov narastá, tak sa blížime k hľadanému súčtu nekonečného radu. Napr. berme n = 10000000 a vráťme sa k cyklu typu "for":

```
S7 = 0; for i = 1:10000000, S7 = S7 + 1/(2*i - 1); end, S7
S7 =
     9.04080283849124
```

Rozdiel medzi S3 a S7 je značný, a preto je celkom prirodzené pýtať sa: Koľko členov radu treba vziať, aby čiastočný súčet aproximoval súčet radu dostatočne presne?

Pravdepodobne väčšina z Vás si už uvedomila, že úloha nie je korektná. Súčet uvažovaného radu nie je konečné číslo. Dá sa ukázať, že postupnosť čiastočných súčtov neobmedzene narastá (to platí, pokiaľ neuvažujeme chyby aritmetických operácií). Úloha nie je korektná.

Ak ale predsa experimentujeme v reálnom výpočtovom prostredí (napr. MATLABe), môžeme pozorovať, že postupnosť čiastočných súčtov nerastie tak, ako by sme očakávali. Vysvetlenie prečo je to tak, však prekračuje rámec tohoto textu.

1.6.5 Riešte úlohu 1.6.5, ale tentoraz nech ide o *konvergentný* rad: $\sum_{n=1}^{\infty} \frac{1}{n^2}$

Riešenie.

```
 x = (1:100).^2; S1 = sum( ones(1, 100)./x )
S1 =
      1.63498390018489
S2 = 0; n = 1; clen = 1;
while clen > 0.005,
      S2 = S2 + clen;
      n = n + 1;
      clen = 1/n^2;
end,
n, S2
n =
      15
S2 =
      1.57599583900054
```

Poznamenávame, že súčet radu sa rovná číslu $\pi^2/6$. Lepšou aproximáciou je S1, pretože odpovedá súčtu 100 členov radu, kým S2 odpovedá súčtu len 15-tich členov. Overme:

```
[pi^2/6 - S1, pi^2/6 - S2]
ans =
      0.00995016666333   0.06893822784768
```

Ako vidíme, čísla potvrdzujú správnosť našej úvahy.

1.6.6 Uvažujme počítačovú množinu čísel F = F(10, 3, -9, 9). Určte:

a) hodnotu najmenšieho kladného čísla množiny F
b) hodnotu najväčšieho čísla množiny F
c) vzdialenosti medzi číslom 1 a jeho susedmi
d) vzdialenosť medzi susedmi čísla 1000
e) najmenšiu vzdialenosť medzi prvkami F
f) najväčšiu vzdialenosť medzi prvkami F

Riešenie.

a) Najmenším kladným prvkom F je $x_{\min} = 0.100.10^{-9}$, pretože množinu F tvoria čísla s normalizovanou mantisou, a teda prvá cifra za desatinnou bodkou musí byť nenulová.

b) Najväčšie kladné číslo F je $x_{\max} = 0.999.10^{9}$

c) Označme ľavého suseda čísla 1 symbolom LS a pravého symbolom PS. Zrejme $LS = 0.999.10^0$ a $PS = 0.101.10^1$, pretože číslo 1 má tvar $0.100.10^1$. Vzdialenosť čísla 1 k LS sa rovná 0.001 a vzdialenosť čísla 1 k PS sa rovná 0.01.

d) Teraz $LS = 0.999.10^3$ a $PS = 0.101.10^4$ Vzdialenosť medzi susedmi sa rovná

$$PS - LS = 0.101.10^4 - 0.999.10^3 = (1.01 - 0.999).10^3 = 11$$

čo je asi prekvapujúce.

e) Zrejme je to vzdialenosť medzi $x_{\min}$ a jeho pravým susedom, teda $0.001.10^{-9}$, t.j. 10^{-12}

f) Teraz ide o vzdialenosť medzi $x_{\max}$ a jeho ľavým susedom, teda $0.001.10^{9}$, t.j. 10^{6}, čo je asi opäť nečakané.

1.6.8 V počítačovej množine čísel systému MATLAB určte vzdialenosti medzi bezprostrednými susedmi čísel: 1, 8, 64, 4096.

Riešenie. V prvých verziách systému MATLAB jediným datovým typom bola matica, ktorej prvky boli typu *double*. Aj keď dnes je to už inak, hovorme (pre jednoduchosť) len o type double. MATLAB má pre tento typ vyhradených 64 bitov, 11 na exponent, 52 pre mantisu a jeden bit pre signum (znamienko).

Pretože prvá cifra za binárnou bodkou musí byť 1, nie je treba ju kódovať, a tak MATLAB pracuje s 53 miestnou mantisou. Teraz už ľahko nájdete odpovede na uvedené otázky, napr.

Ľavý sused jednotky je číslo $LS = 0.111\ 111\ 111\ 111\ 111\ ...\ 111\ .2^0$ (mantisa má 53 jednotiek), kým pravý sused jednotky je číslo $PS = 0.100\ 000\ 000\ 000\ 000\ ...\ 001\ .2^1$.

Vzdialenosť medzi nimi môžeme určiť ako súčet ich vzdialeností d_1, d_2 k jednotke: $d_1(LS, 1) = 2^{-53}$, kým $d_2(PS, 1) = 2^{-52}$ preto hľadaná vzdialenosť $d(LS, PS) = 2^{-53} + 2^{-52} = 2^{-53}(1 + 2) = 3.2^{-53}$

Pre ďalšie uveďme len odpovede:

Vzdialenosť medzi bezprostrednými susedmi čísla 8 sa rovná 3.2^{-50}

Vzdialenosť medzi bezprostrednými susedmi čísla 64 sa rovná 3.2^{-47}

Vzdialenosť medzi bezprostrednými susedmi čísla 4096 sa rovná 3.2^{-41}

1.6.9 Nasledujúce funkcie linearizujte v zadaných bodoch x_0. Jednotlivé lineárne funkcie $L(x)$ berme ako lokálne aproximácie odpovedajúcich funkcií $f(x)$. Vyčíslite chybu $|f(u) - L(u)|$ pre $u = x_0 + 0.2$.

a) $\cos(x)$, $x_0 = 3.pi/4$ b) $\sin(x^2)$, $x_0 = pi/4$ c) $f(x) = x/(x^2 + 1)$, $x_0 = 1.5$

<u>Riešenie</u>.

a) Lineárnou aproximáciou $f(x) = \cos(x)$ je $L(u) = f(x_0) + f'(x_0).(u - x_0) = \cos(x_0) - \sin(x_0).(u - x_0)$ preto

```
fu = cos(3*pi/4 + 0.2) ; Lu = cos(3*pi/4) - sin(3*pi/4)*0.2;
chyba = abs(fu - Lu)
```

```
chyba =
        0.0150
```

b) Lineárnou aproximáciou $f(x) = \sin(x^2)$ je $L(u) = f(x_0) + f'(x_0).(u - x_0) = \sin(x_0^2) + 2x_0\cos(x_0^2).(u - x_0)$ preto

```
fu = sin((pi/4 + 0.2)^2); Lu = sin((pi/4)^2) + 2*(pi/4)*cos((pi/4)^2)*0.2;
chyba = abs(fu - Lu)
```

```
chyba =
        0.0093
```

c) Lineárnou aproximáciou je teraz funkcia $L(u) = f(x_0) + f'(x_0).(u - x_0)$, kde

$$f(x) = x/(x^2 + 1), \quad f'(x) = (1 - x^2)/(1 + x^2)^2$$

$L(u) = f(x_0) + f'(x_0).(u - x_0) = 0.4615 - 0.1183\ 0.2 = 0.4378$ (pre zaujímavosť, $f(1.7) = 0.4370$, ako vidíme, aproximácia je pomerne presná.).

1.6.9 Použitím M-funkcie "conv" a konečnými Taylorovými radmi funkcií exp(-x), cos(x), aproximujte polynómom p(x) funkciu $f(x) = \exp(-x).\cos(x)$.
Tabelujte diferenciu

$$|\exp(-x).\cos(x) - p(x)| \quad \text{pre } x \in [-0.5, 0.5] \text{ s krokom } 0.1$$

<u>Riešenie</u>.

Zo známeho Taylorovho radu funkcie exp($-x$) vezmime 4 členy: $1 - x + x^2/2 - x^3/6$. Označme tento polynóm ako $e(x)$. V MATLABe je takýto polynóm reprezentovaný vektorom jeho koeficientov. Pretože prvá zložka vektora predstavuje koeficient pri najvyššej mocnine, $e(x)$ bude reprezentovaný vektorom

$$ex = [\ -1/6,\ 1/2,\ -1,\ 1\ ].$$

Z radu funkcie cos(x) vezmime 3 členy: $1 - x^2/2 + x^4/24$ a označme ho ako $c(x)$. Reprezentujúci vektor bude vektor

$$cx = [\ 1/24,\ 0,\ -1/2,\ 0,\ 1],$$

pretože koeficienty pri párnych mocninách $c(x)$ sa rovnajú nule. Polynóm $p(x) = e(x).c(x)$ a jeho koeficienty získame M-funkciou "conv":

```
ex = [-1/6, 1/2, -1, 1]; cx = [1/24, 0, -1/2, 0, 1]; p = conv(ex, cx)
```

```
p =  -0.0069   0.0208   0.0417   -0.2083   0.3333   0.00   -1.0000    1.0000
```

Použime formát "long" a tabelujme diferenciu medzi hodnotami $f(x) = \exp(-x).\cos(x)$ a hodnotami polynómu $p(x)$. Ako vieme, v MATLABe hodnoty polynómu získavame M-funkciou "polyval":

```
x =  (-0.5:.1:0.5)';  [ x, abs(polyval(p,x) - exp(-x).*cos(x))]
ans =
     -0.50000000000000   0.00249884561195
     -0.40000000000000   0.00105816110974
     -0.30000000000000   0.00034141779494
     -0.20000000000000   0.00006793246700
     -0.10000000000000   0.00000422863496
                     0                  0
      0.10000000000000   0.00000406303964
      0.20000000000000   0.00006273623141
      0.30000000000000   0.00030325927635
      0.40000000000000   0.00090502567942
      0.50000000000000   0.00206154618789
```

1.6.11 Zostrojte graf kvantilovej funkcie rozdelenia N(0,1) využitím aproximácie ψ kvantilovej funkcie, ktorá je daná takto:

a0 = 2.515517, a1 = 0.802853, a2 = 0.010328, b1 = 1.432788, b2 = 0.189269, b3 = 0.001308

w = sqrt(-2.ln(y)), keď pre $y \in (0, 0.5]$ platí:

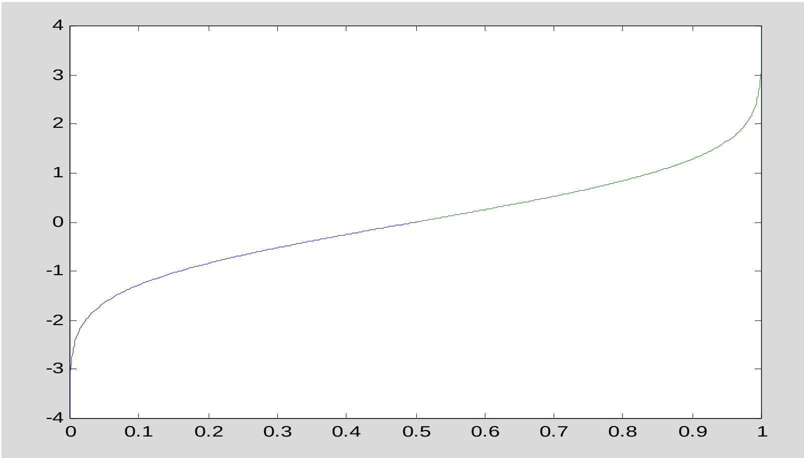
$$\psi(y) = - w + (a0 + a1.w + a2.w^2)/(1 + b1.w + b2.w^2 + b3.w^3),$$

a pre $y \in (0.5, 1)$ platí:

$$\psi(y) = - \psi(1 - y)$$

Riešenie.

```
a0 = 2.515517; a1 = 0.802853; a2 = 0.010328; b1 = 1.432788; b2 = 0.189269;
b3 = 0.001308;
y1 = linspace(eps, 0.5, 501); w = sqrt(-2*log(y1)); psi1 = -w + (a0 + a1*w +
a2*w.^2)./(1 + b1*w +b2*w.^2 + b3*w.^3);
y2 = linspace(0.5+eps, 1-eps, 501); w2= sqrt(-2*log(1-y2)); psi2 = w2 -(a0 +
a1*w2 + a2*w2.^2)./(1 + b1*w2 +b2*w2.^2 + b3*w2.^3);
plot(y1,psi1,y2,psi2), axis([0, 1, -4 4])
```



Kvantilová funkcia rozdelenia N(0, 1)

Pre posúdenie presnosti získanej aproximácie tabelujme horné decily rozdelenia N(0, 1):

```
x = 0.5:0.1:0.90; wd = sqrt(-2*log(1-x)); decily = wd - (a0 + a1*wd +
a2*wd.^2)./(1 + b1*wd +b2*wd.^2 + b3*wd.^3); [x', decily']
ans =
    0.5000   -0.0000
    0.6000    0.2529
    0.7000    0.5240
    0.8000    0.8415
    0.9000    1.2817
```

Ľahko sa možno presvedčiť, že chyba aproximácie nepresahuje 5.10^{-4}.