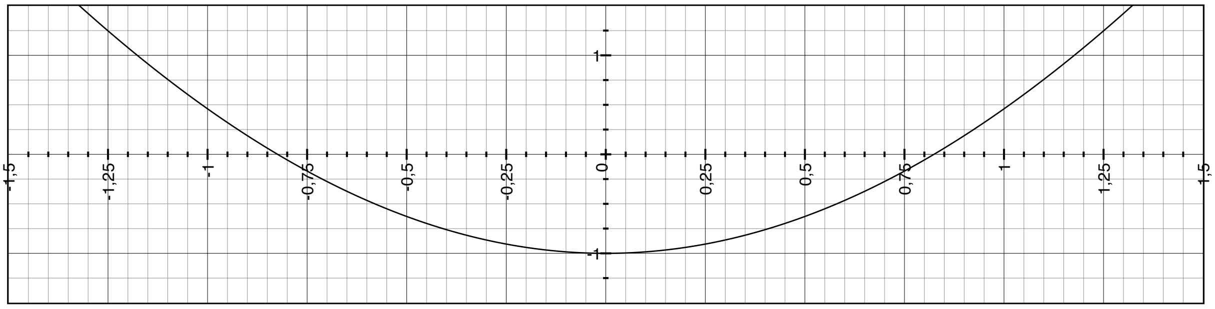
1. Odhadněte graficky polohu kořene dané rovnice a nalezněte tento kořen metodou bisekce s přesností $\epsilon = 10^{-6}$.

Zvoleno zadání a) $x^2 - cos(x) = 0$.



Grafický odhad. Funkce představuje symetrickou parabolu (x^2) posunutou v záporném směru osy y o hodnotu určenou goniometrickou funkcí *cosinus*. Definičním oborem této funkce mohou být všechna reálná čísla. Oborem hodnot funkce je sjednocení všech kladných reálných čísel parabolické složky x^2 se záporným oborem hodnot funkce *cosinus*, která v bodě $x = 0$ dosahuje svého maxima $y = 1$ ($-cos(0) = -1$). $D_f : \mathbb{R}$, $H_f : \langle -1; +\infty)$. Parabola protíná osu x ve dvou bodech ξ_1, ξ_2 zrcadlených podle osy y, funkce má tedy dva reálné kořeny. Pro odhad bude použito okolí kladného kořene, které bylo vzhledem k charakteru růstu jednotlivých složek dané funkce stanoveno na $\langle \frac{\pi}{6}; 1 \rangle$.

Uvažujme následující program v prostředí MATLAB využívající rekurzivní funkci `bisekce`, přijímající jako argumenty funkci $f(x)$, hledanou hodnotu y, počáteční meze intervalu $\langle a; b \rangle$ a požadovanou přesnost ϵ,

```
function m = bisekce(funkce, hledana_hodnota, a_start, b_start, epsilon)

% počáteční odhad
m = (a_start + b_start) / 2.0;

% výpočet funkce v bodě odhadu
x = funkce(m);

if (abs(x - hledana_hodnota) >= epsilon)
    % nastaveních nových mezí pro další krok
    if (x - hledana_hodnota < 0)
        a = m;
        b = b_start;
    else
        a = a_start;
        b = m;
    end
    % rekurze
    m = bisekce(funkce, hledana_hodnota, a, b, epsilon);
end
```

a její následné zavolání s odpovídajícimi parametry řešeného příkladu:

```
>> xi = bisekce(@(x) x^2 - cos(x), 0, pi/6, 1.0, 10e-6)
xi =

    0.8241
```

Při požadované přesnosti $\epsilon = 10^{-6}$ budou pak kořeny funkce nalezené metodou bisekce $\underline{\underline{\xi_{1,2} = \{-0,824135; 0,824135\}}}$.

2. Nalezněte všechny kořeny dané rovnice. Použijte Newtonovu metodu. Nejprve vždy odhadněte polohu kořene a ověřte Fourierovy podmínky konvergence metody. Kořen hledejte s přesností $\epsilon = 10^{-6}$.

Zvoleno zadání b) $\ln x + (x+1)^3 = 0$.

Funkce je složena z přirozeného logaritmu a polynomu třetího stupně. Průnikem definičních oborů těchto dílčích funkcí $D_{f_1} : \{x \in \mathbb{R} : x > 0\}$, $D_{f_2} : \mathbb{R}$ je obor kladných reálných čísel větších než nula, $D_f : \{x \in \mathbb{R} : x > 0\}$. Funkce může nabývat hodnot všech reálných čísel, $H_f : \mathbb{R}$. Z tohoto důvodu lze očekávat pouze jeden reálný kořen, jehož poloha bude ovlivněna především složkou $\ln x$, mající v nevlastním bodě $x = 0$ nevlastní limitu $\lim_{x=0} \ln x = -\infty$ a protínající osu x v bodě $x = 1$, se záporným posunem ve směru osy x způsobeným složkou kubického polynomu. Odhadovaný interval polohy kořene ξ byl tedy zvolen $\xi \in (0; 0,5\rangle$.

První derivace: $f'(x) = \dfrac{1}{x} + 3(x+1)^2$

Druhá derivace: $f''(x) = -\dfrac{1}{x^2} + 6(x+1) = -\dfrac{1}{x^2} + 6x + 6$

Pro konvergenci Newtonovy metody musí být splněna Fourierova podmínka $f(x_0) \cdot f''(x_0) > 0$. Jako počáteční aproximace $x_0 \in (0; 0,5\rangle$ byla zvolena pravá krajní hodnota $x_0 = 0,5$.

$$f(x_0) \cdot f''(x_0) > 0 \tag{1}$$

Dosazení:

$$\left(\ln x_0 + (x_0+1)^3\right) \cdot \left(-\frac{1}{x_0^2} + 6x_0 + 6\right) > 0 \tag{2}$$

$$\left(\ln 0,5 + (0,5+1)^3\right) \cdot \left(-\frac{1}{0,5^2} + 6 \cdot 0,5 + 6\right) > 0 \tag{3}$$

$$\left(\sqrt{e} + 1,5^3\right) \cdot \left(-\frac{1}{0,25} + 3 + 6\right) > 0 \tag{4}$$

$$\frac{-1 + 0,75 + 1,5}{0,25} \cdot \left(\sqrt{e} + 1,5^3\right) > 0 \tag{5}$$

$$\frac{1,25}{0,25} \cdot \left(\sqrt{e} + 1,5^3\right) > 0 \tag{6}$$

$$5 \cdot (\sqrt{e} + 3,375) > 0 \tag{7}$$

Fourierova podmínka konvergence pro počáteční aproximaci $x_0 = 0,5$ je tedy splněna. Pro každý n-tý krok výpočtu Newtonovou iterační metodou pak platí

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{8}$$

Po dosazení tedy

$$x_{n+1} = x_n - \frac{\ln x_n + (x_+1)^3}{\frac{1}{x_n} + 3(x_n+1)^2} \tag{9}$$

Obdobně jako v předchozím příkladě uvažujme následující program v prostředí MATLAB s funkcí `newton`, přijímající jako argumenty původní funkci $f(x)$, její první derivaci $f'(x)$, počáteční aproximaci x_0 a požadovanou přesnost ϵ. Výstupem funkce je odhad ξ při požadované přesnosti ϵ a n označující počet iterací.

```
% Newtonova iterační metoda
function [xn, n] = newton(f, df, x0, epsilon)

n = 0;
xk = x0;
xn = x0 - f(x0)/df(x0);

while (abs(xn - xk) > epsilon & n < 250)
    % počítadlo iterací
    n = n+1;

    % aktualizace hodnoty
    xk = xn;

    % další člen v řadě
    xn = xk - f(xk)/df(xk);
end
```

Funkce bude bude následně zavolána s konkrétními argumenty příkladu:

```
>> [xi, n] = newton(@(x) log(x)+(x+1)^3, @(x) (1/x)+3*(x+1)^2, 0.5, 10e-6)

xi =
    0.1874
n =
    3
```

Při požadované přesnosti $\epsilon = 10^{-6}$ je kořen funkce nalezené Newtonovou iterační metodou $\underline{\underline{\xi = 0,187439}}$.

3. Metodou prosté iterace nalezněte všechny kořeny dané rovnice s přesností $\epsilon = 10^{-6}$. Nejprve odhadněte polohu kořene (kořenů) a rovnici upravte do vhodného tvaru. Ověřte podmínky konvergence.

Zvoleno zadání b) $f : 5x - 8\ln x - 8 = 0$.

Funkce je tvořena lineárním členem $5x$, absolutním členem -8 a přirozeným logaritmem $8\ln x$. Díky přítomnosti přirozeného logaritmu je definiční obor funkce $f(x)$ omezen na kladná reálná čísla větší než nula, $D_f : \{x \in \mathbb{R} : x > 0\}$.

Následující tabulka předpokládá existenci dvou kořenů, ležících v intervalech $x_{0_2} \in \langle 0,2; 1,5\rangle$, resp. $x_{0_1} \in \langle 2,5; 4\rangle$.

x	0,1	0,2	1,5	2,5	4,0
y	10,921	5,876	-3,744	-2,830	0,910
$sig f(x)$	$+$	$+$	$-$	$-$	$+$

Pro hledání pevných bodů existují dvě možná ekvivalentní vyjádření funkce $f(x) = 0$ ve tvaru iterační funkce $x = g(x)$:

$$g_1(x) = \frac{8}{5} \cdot (\ln x + 1), \quad g_1'(x) = \frac{8}{5} \cdot \frac{1}{x}, \quad D_{g_1'} : \mathbb{R} \setminus \{0\} \tag{1}$$

$$g_2(x) = \sqrt[8]{e^{5x-8}}, \quad g_2'(x) = \frac{5}{8} \cdot \sqrt[8]{e^{5x-8}}, \quad D_{g_2'} : \mathbb{R} \tag{2}$$

Pro splnění podmínek konvergence prosté iterační metody musí platit, že je daná funkce $g(x)$ na odpovídajícím intervalu I spojitá a zobrazuje tento interval I zpět na tentýž interval I. Ověření spojitosti funkce $g(x)$ lze provést důkazem, že v daném intervalu I existuje její derivace $g'(x)$, tedy $I \subset D_{g'}$. Platí-li zároveň $|g'(x)| < 0, \forall x \in I$, pak v tomto intervalu existuje právě jeden pevný bod.

$I_{1,2}$	$I_2 = \langle 0,2; 1,5\rangle$	$I_1 = \langle 2,5; 4\rangle$
$I_{1,2} \subset D_{g_1'}$	ano	ano
$I_{1,2} \subset D_{g_2'}$	ano	ano
$\forall x \in I_{1,2} : g_1(x) \in I_{1,2}$	ne	ano
$\forall x \in I_{1,2} : g_2(x) \in I_{1,2}$	ano	ne

Z přehledové tabulky podmínek pro intervaly vyplývá, že prostá iterační metoda použitelná pro nalezení odhadů $\xi_{1,2}$ původní funkce $f(x)$ bude konvergovat v intervalu $I_1 = \langle 2,5; 4\rangle$ při použití ekvivalentní funkce $g_1(x)$ a v intervalu $I_2 = \langle 0,2; 1,5\rangle$ při použití ekvivalentní funkce $g_2(x)$.

Uvažujme následující program v prostředí MATLAB s funkcí `iterace` realizující výpočet odhadu kořene prostou iterační metodou. Argumenty této funkce jsou ekvivalentní vyjádření funkce $g(x)$, počáteční bod odhadu x_0 a požadovaná přesnost ϵ. Výstupem funkce je odhad ξ při požadované přesnosti ϵ a n označující počet iterací.

```
% Metoda prosté iterace
function [xn, n] = itrace(g, x0, epsilon)

n = 0;
x = x0;
% x = g(x) - pevný bod
xn = g(x);

while (abs(x - xn) && n < 250)
    n = n + 1;
    % další krok
    x = xn;
    xn = g(x);
end
```

Zavolání programu pro funkci $g_1(x_{0_1})$ v bodě $x_{0_1} = 3,5 \in \langle 2,5;4\rangle$:

```
>> [xi1, n1] = iterace(@(x) (8*log(x)+8)/5, 3.5, 10e-6)

xi1 =
    3.6882

n2 =
    12
```

Zavolání programu pro funkci $g_2(x_{0_2})$ v bodě $x_{0_2} = 0,5 \in \langle 0,2;1,5\rangle$:

```
>> [xi2, n2] = iterace(@(x) exp((5*x-8)/8), 0.5, 10e-6)

xi2 =
    0.5041

n2 =
    5
```

Při požadované přesnosti $\epsilon = 10^{-6}$ jsou kořeny funkce nalezené prostou iterační metodou $\underline{\underline{\xi_{2,1} = \{0,504132; 3,688238\}}}$.