Metóda pevného bodu

Príklad z prednášky

Úloha: Nájdite metódou pevného bodu korene rovnice $x^2 - 2x - 3 = 0$

```
Solve[x^2 - 2 x - 3 == 0, x]
```

```
{{x → -1}, {x → 3}}
```

Vieme, že korene rovnice sú −1 a 3 . Vďaka tejto informácii budeme môcť sledovať rýchlosť a kvalitu iteračnej schémy

Zostavíme rôzne iteračné schémy:

1. x = Sqrt[2x+3]
2. x = 3 / (x-2)
3. x = (x^2 - 3) / 2 - schema nekonverguje

Okrem konvergencie si všímajte aj rýchlosť konvergencie iteračnej schémy. Aj tá je totiž často určujúcim faktorom vh

Všimnite si, že zastavovacia podmienka v tomto algoritme sa líši od predchádzajúcich podmienok. Musíme porovnáv

1. x == Sqrt[2x+3]

```
Clear[x, g];
g[x_] = Sqrt[2 x + 3];
x[0] = 4;

Do[
  x[n + 1] = g[x[n]] // N;
  Print[n + 1, ". iteracia:  " , "x = ", SetPrecision[x[n + 1], 10]];
  If[Abs[x[n + 1] - x[n]] < 10^(-5), Break[]],
  {n, 0, 15}];
```

```
1. iteracia:  x = 3.316624790
2. iteracia:  x = 3.103747667
3. iteracia:  x = 3.034385495
4. iteracia:  x = 3.011440019
5. iteracia:  x = 3.003810919
6. iteracia:  x = 3.001270038
7. iteracia:  x = 3.000423316
8. iteracia:  x = 3.000141102
9. iteracia:  x = 3.000047034
10. iteracia:  x = 3.000015678
11. iteracia:  x = 3.000005226
12. iteracia:  x = 3.000001742
```

2. $x == \frac{3}{(x-2)}$

```
Clear[x, g];
g[x_] = 3 / (x - 2);
x[0] = 4;

Do[
  x[n + 1] = g[x[n]] // N;
  Print[n + 1, ". iteracia:  " , "x = ", SetPrecision[x[n + 1], 10]];
  If[Abs[x[n + 1] - x[n]] < 10^(-5), Break[]],
  {n, 0, 15}];
```

```
1. iteracia:  x = 1.500000000
2. iteracia:  x = -6.000000000
3. iteracia:  x = -0.3750000000
4. iteracia:  x = -1.263157895
5. iteracia:  x = -0.9193548387
6. iteracia:  x = -1.027624309
7. iteracia:  x = -0.9908759124
8. iteracia:  x = -1.003050641
9. iteracia:  x = -0.9989841528
10. iteracia:  x = -1.000338730
11. iteracia:  x = -0.9998871026
12. iteracia:  x = -1.000037634
13. iteracia:  x = -0.9999874555
14. iteracia:  x = -1.000004182
15. iteracia:  x = -0.9999986062
```

3. $x == \frac{x^2-3}{2}$

```
Clear[x, g];
g[x_] = (x^2 - 3) / 2;
x[0] = 4;

Do[
  x[n + 1] = g[x[n]] // N;
  Print[n + 1, ". iteracia:  ", "x = ", SetPrecision[x[n + 1], 10]];
  If[Abs[x[n + 1] - x[n]] < 10^(-5), Break[]],
  {n, 0, 15}];
```

```
1. iteracia:  x = 6.500000000
2. iteracia:  x = 19.62500000
3. iteracia:  x = 191.0703125
4. iteracia:  x = 18 252.43216
5. iteracia:  x = 1.665756384×10^8
6. iteracia:  x = 1.387372165×10^16
7. iteracia:  x = 9.624007619×10^31
8. iteracia:  x = 4.631076133×10^63
9. iteracia:  x = 1.072343307×10^127
10. iteracia:  x = 5.749600845×10^253
11. iteracia:  x = 1.652895494×10^507
12. iteracia:  x = 1.366031756×10^1014
13. iteracia:  x = 9.330213798×10^2027
14. iteracia:  x = 4.352644476×10^4055
15. iteracia:  x = 9.472756965×10^8110
16. iteracia:  x = 4.486656226×10^16 221
```

Predefinované príkazy systému Mathematica

Výhodou metódy pevného bodu je aj skutočnosť, že metódu nemusíme programovať, ale môžeme použiť zabudovan

Parametre udávame v poradí: štartovacia funkcia (schéma, funkcia $g(x)$), štartovací bod, maximálny počet opakovan

Všimnnite si prosím, že aj keď funkciu zadávame štandardným spôsobom do schémy zapisujeme len jej meno.

1. x == Sqrt[2x+3]

```
Clear[x, g];
g[x_] = Sqrt[2 x + 3];
FixedPointList[g, 4.]
FixedPointList[g, 4., 10]
```

```
{4., 3.31662, 3.10375, 3.03439, 3.01144, 3.00381, 3.00127, 3.00042, 3.00014, 3.00005, 3.00002, 3
```

```
{4., 3.31662, 3.10375, 3.03439, 3.01144, 3.00381, 3.00127, 3.00042, 3.00014, 3.00005, 3.00002}
```

Ak potrebujeme vo výpise len výsledný pevný bod, použijeme nasledujúci príkaz.

```
Clear[x, g];
g[x_] = Sqrt[2 x + 3];
FixedPoint[g, 4.]
SetPrecision[FixedPoint[g, 4.], 20]
```

```
3.
```

```
3.0000000000000000000
```

2. $x == \frac{3}{(x-2)}$

```
Clear[x, g];
g[x_] = 3 / (x - 2);
FixedPointList[g, 4.]
FixedPointList[g, 4., 10]
```

```
{4., 1.5, -6., -0.375, -1.26316, -0.919355, -1.02762, -0.990876, -1.00305, -0.998984, -1.0003
```

```
{4., 1.5, -6., -0.375, -1.26316, -0.919355, -1.02762, -0.990876, -1.00305, -0.998984, -1.0003
```

3. $x == \frac{x^2-3}{2}$

Schéma nekonverguje a preto volíme len 10 opakovaní.

```
Clear[x, g];
g[x_] = (x^2 - 3) / 2;
FixedPointList[g, 4., 10]
```

$\{4., 6.5, 19.625, 191.07, 18252.4, 1.66576 \times 10^{8}, 1.38737 \times 10^{16}, 9.62401 \times 10^{31}, 4.63108 \times 10^{63}, 1.0$

Poznámka: Ak nemáme explicitne overenú konvergenciu schémy, volíme najskôr menší, presne definovaný počet op

Úloha

Overte, či nasledujúce schémy sú konvergentné. Testujte všetky schémy a ich súlad s prednáškou.

4. $x == x^2 - x - 3$

```
Clear[x, g];
g[x_] = x^2 - x - 3
FixedPointList[g, 4., 10]
```

$-3 - x + x^2$

$\{4., 9., 69., 4689., 2.1982 \times 10^7, 4.8321 \times 10^{14}, 2.33491 \times 10^{29}, 5.45183 \times 10^{58}, 2.97224 \times 10^{117}, 8.83$

Vidíme, že táto schéma diverguje.

5. $x^3 == x^3 + x^2 - x - 3$, po úprave $x == \sqrt[3]{x^3 - x^2 - 2x - 3}$

```
Clear[x, g];
g[x_] = (x^3 - x^2 - 2 x - 3) ^ (1 / 3)
FixedPointList[g, 4., 10]
```

$\left(-3 - 2x - x^2 + x^3\right)^{1/3}$

$\{4., 3.33222, 2.53196, 1.20671, 0.861348 + 1.4919\, i, 1.42073 - 1.62241\, i, 1.31272 + 2.00561\, i, 1.602$

Vidíme, že táto schéma našla komplexný výsledok, nie je to hľadaný koreň. Schéma nie je taktiež konvergentná.

Úlohy na samostatné počítanie

1. príklad

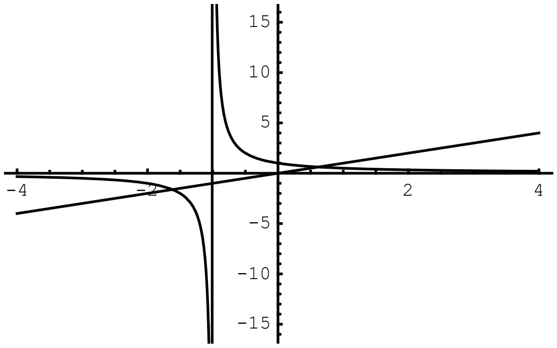
Hľadajte korene kvadratickej rovnice $x^2 + x - 1 = 0$.
Jedným z jej koreňov je $x_1 = -0.6180339$.
Jednou z možných iteračných schém je predpis $x == \frac{1}{(x+1)}$.

Nájdite aj iné konvergentné iteračné schémy.
Nájdite vhodný štartovací bod tak, aby zvolené schémykonvergovali ku druhému koreňu $x_2 = -1.6180339$.
Koľko iterácií potrebuje na nájdenie tohto koreňa na 5 desatinných miest
a) Newtonova metóda
b) Secant metóda (metóda tetív)
c) Mullerova metóda
d) metóda pevného bodu

```
Solve[x^2 + x - 1 == 0, x] // N
```

{{x → -1.61803}, {x → 0.618034}}

```
h[x_] = 1 / (x + 1);
Plot[{x, h[x]}, {x, -4, 4}]
```



- Graphics -

```
FixedPointList[h, 2., 10]
SetPrecision[FixedPointList[h, 2.], 10]
```

{2., 0.333333, 0.75, 0.571429, 0.636364, 0.611111, 0.62069, 0.617021, 0.618421, 0.617886, 0.618(

{2.000000000, 0.3333333333, 0.7500000000, 0.5714285714, 0.6363636364, 0.6111111111, 0.62068965

2. príklad

Hľadajte koreň rovnice $2x^3 + 4x^2 - 4x - 6 = 0$ v blízkosti bodu 1.3 .
Nájdite aspoň 2 konvergentné iteračné schémy a realizujte pomocou nich výpočet.

3. príklad

Hľadajte koreň rovnice $2x^3 + 4x^2 - 4x - 6 = 0$ v blízkosti bodu −2, 3 a v blízkosti bodu −0, 9.
Konvergujú iteračné schémy, ktoré ste našli v predchádzajúcom príklade aj k týmto koreňom? Ak nie nájdite aspoň je

4. príklad

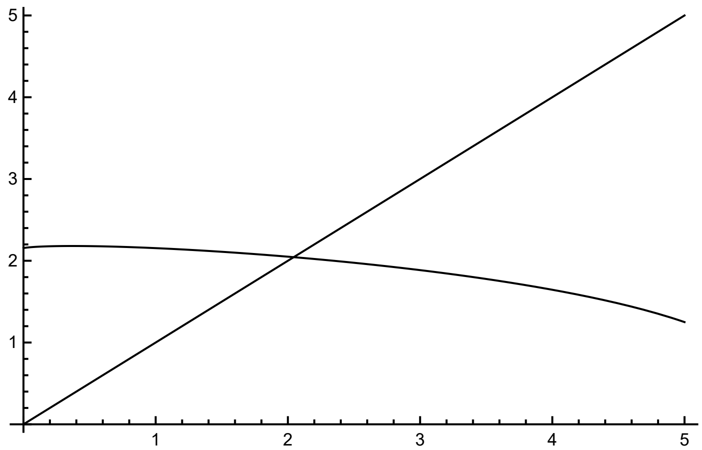
Nájdite korene rovnice $x^3 - \frac{10}{x} + x\,.\,\ln x = 0$. Výpočet realizujte pomocou metódy pevného bodu. Použite schémy $x = \sqrt[3]{}$

Úlohu vypočítajte klasickým spôsobom, metódou pevného bodu, ostatnými metódami
Pre každý postup porovnajte (ak je to možné) 6. iteráciu. Počítajte s toleranciou 10^{-6}.

```
Clear[g, x]
g[x_] = (10 - x * Log[x]) ^ (1 / 3)
```

$(10 - x\,\text{Log}[x])^{1/3}$

```
Plot[{x, g[x]}, {x, 0, 5}]
```



Metóda pevného bodu - len pevný počet iterácií

```
Clear[g, x]
x[0] = 2.5;
g[x_] = (10 - x * Log[x]) ^ (1 / 3);
pocetopakovani = 13;
Do[x[i + 1] = g[x[i]], {i, 0, pocetopakovani}]
```

```
While[x[i + 1] ≠ x[i], x[i + 1] = g[x[i]]]
```

```
Table[x[i], {i, 0, 20}]
```

```
{2.5, 1.97547, 2.05316, 2.04267, 2.0441, 2.04391, 2.04393, 2.04393, 2.04393, 2.04393, 2.04393, 2
```

```
TableForm[Table[{i, NumberForm[x[i], 10], Abs[x[i + 1] - x[i]]}, {i, 0, pocetopakovani}],
     TableHeadings → {None, {"i", "x_i", "|x_{i+1}-x_i|"}},
     TableSpacing → {1, 5}]
```

i	"x_i"	"$\lvert x_{i+1}-x_i \rvert$"
0	2.5	0.524527
1	1.975473212	0.0776912
2	2.053164424	0.0104986
3	2.042665841	0.00143891
4	2.044104752	0.00019684
5	2.043907913	0.0000269343
6	2.043934847	3.68538×10^{-6}
7	2.043931161	5.04267×10^{-7}
8	2.043931666	6.89984×10^{-8}
9	2.043931597	9.44099×10^{-9}
10	2.043931606	1.2918×10^{-9}
11	2.043931605	1.76756×10^{-10}
12	2.043931605	2.41855×10^{-11}
13	2.043931605	3.30935×10^{-12}

5. príklad

Hľadajte koreň rovnice $576\,x^5 + 912\,x^4 + 172\,x^3 - 261\,x^2 - 54\,x + 27 = 0$

Najdite aspoň dve iteračné schémy na $[-1, 1]$ a overte ich koncvergenciu. Aspoň jedna zo schém musí byť konvergent