Hledání kořenů rovnic jedné reálné proměnné
– metoda sečen –

Michal Čihák

19. října 2011

Opakování – rovnice přímky

Úloha: Určete rovnici přímky procházející body $A[a, f(a)]$ a $B[b, f(b)]$, kde f je funkce spojitá na intervalu $\langle a, b\rangle$. (Přímka je sečnou grafu funkce v zadaných bodech).

Opakování – rovnice přímky

Úloha: Určete rovnici přímky procházející body $A[a, f(a)]$ a $B[b, f(b)]$, kde f je funkce spojitá na intervalu $\langle a, b\rangle$. (Přímka je sečnou grafu funkce v zadaných bodech).
Řešení:

$$y = f(b) + \frac{f(b) - f(a)}{b - a}(x - b)$$

Metoda sečen

- obvykle konverguje rychleji k řešení než metoda půlení intervalů (méně iterací)
- mohou ale nastat případy, kdy tato metoda k řešení nekonverguje (nedosáhneme předepsané přesnosti aproximace kořenu rovnice) – lze ji ale modifikovat tak, aby vždy konvergovala (viz metoda *regula falsi*)
- na začátku musíme zvolit dvě startovní hodnoty p_0 a p_1, které by měly být co nejblíže hledanému kořenu (kořen ale nemusí ležet mezi těmito hodnotami)

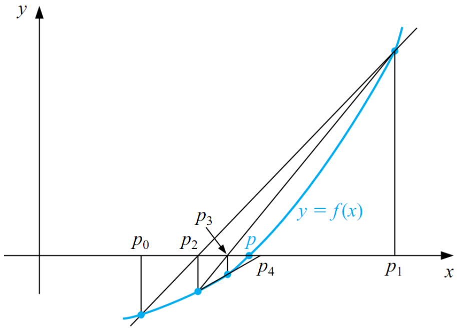
Metoda sečen

- obvykle konverguje rychleji k řešení než metoda půlení intervalů (méně iterací)
- mohou ale nastat případy, kdy tato metoda k řešení nekonverguje (nedosáhneme předepsané přesnosti aproximace kořenu rovnice) – lze ji ale modifikovat tak, aby vždy konvergovala (viz metoda *regula falsi*)
- na začátku musíme zvolit dvě startovní hodnoty p_0 a p_1, které by měly být co nejblíže hledanému kořenu (kořen ale nemusí ležet mezi těmito hodnotami)

Metoda sečen

- obvykle konverguje rychleji k řešení než metoda půlení intervalů (méně iterací)
- mohou ale nastat případy, kdy tato metoda k řešení nekonverguje (nedosáhneme předepsané přesnosti aproximace kořenu rovnice) – lze ji ale modifikovat tak, aby vždy konvergovala (viz metoda *regula falsi*)
- na začátku musíme zvolit dvě startovní hodnoty p_0 a p_1, které by měly být co nejblíže hledanému kořenu (kořen ale nemusí ležet mezi těmito hodnotami)

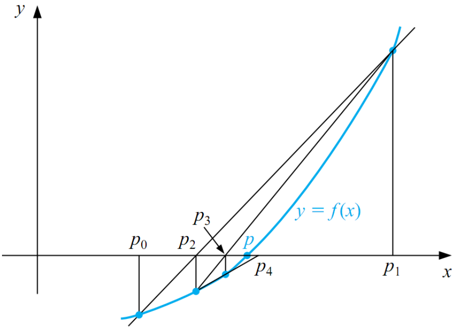
Algoritmus metody sečen



Na začátku jsou dány hodnoty p_0, p_1. Rovnice sečny grafu funkce f v bodech $[p_0, f(p_0)]$ a $[p_1, f(p_1)]$ je

$$y = f(p_1) + \frac{f(p_1) - f(p_0)}{p_1 - p_0}(x - p_1).$$

Algoritmus metody sečen

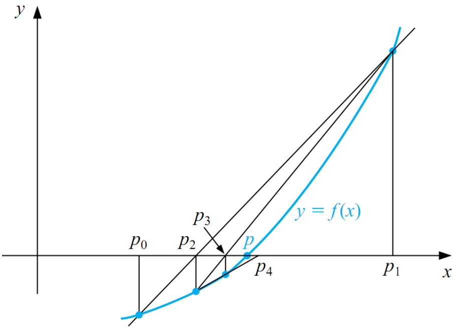


Hodnota p_2 (další iterace) se určí jako průsečík sečny s osou x soustavy souřadnic. Do předchozí rovnice tedy dosadíme $y = 0$

$$0 = f(p_1) + \frac{f(p_1) - f(p_0)}{p_1 - p_0}(x - p_1)$$

a rovnici vyřešíme (vyjádříme neznámou x)

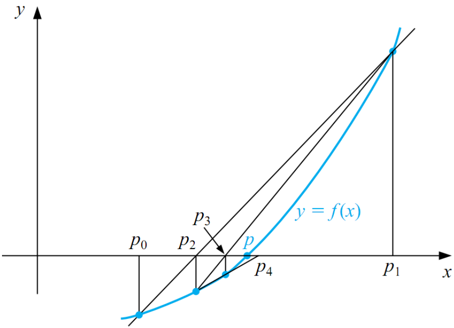
Algoritmus metody sečen



$$x = p_1 - \frac{f(p_1)(p_1 - p_0)}{f(p_1) - f(p_0)}.$$

Získanou hodnotu označíme p_2.

Algoritmus metody sečen – shrnutí



Pro $n > 1$ se aproximace p_{n+1} hodnoty kořenu rovnice $f(x) = 0$ vypočítá z aproximací p_n a p_{n-1} pomocí vztahu

$$p_{n+1} = p_n - \frac{f(p_n)(p_n - p_{n-1})}{f(p_n) - f(p_{n-1})}.$$

Metoda sečen

Pro ukončení algoritmu metody sečen používáme dvě kritéria:

1. hodnota $|p_n - p_{n-1}|$ klesne pod předem danou toleranci TOL
2. počet iterací algoritmu překročí předem danou mez N_0 (neexistuje způsob jak ji odhadnout – proto volíme hodně vysokou hodnotu jako pojistku pro případ, že by metoda nekonvergovala)

Metoda sečen

Pro ukončení algoritmu metody sečen používáme dvě kritéria:

1. hodnota $|p_n - p_{n-1}|$ klesne pod předem danou toleranci TOL
2. počet iterací algoritmu překročí předem danou mez N_0 (neexistuje způsob jak ji odhadnout – proto volíme hodně vysokou hodnotu jako pojistku pro případ, že by metoda nekonvergovala)

Příklad

Zadání: Najděte kořen rovnice $x^3 + 4x^2 - 10 = 0$ v intervalu $\langle 1, 2 \rangle$ s tolerancí $0{,}0005$.

Příklad

Zadání: Najděte kořen rovnice $x^3 + 4x^2 - 10 = 0$ v intervalu $\langle 1, 2 \rangle$ s tolerancí $0{,}0005$.

Řešení: Položíme $p_0 = 1, p_1 = 2$ a postupně vypočítáme:

n	p_n	$f(p_n)$
2	1.2631578947	−1.6022743840
3	1.3388278388	−0.4303647480
4	1.3666163947	0.0229094308
5	1.3652119026	−0.0002990679
6	1.3652300011	−0.0000002032

Příklad

Zadání: Najděte kořen rovnice $x^3 + 4x^2 - 10 = 0$ v intervalu $\langle 1, 2 \rangle$ s tolerancí $0{,}0005$.
Řešení: Položíme $p_0 = 1, p_1 = 2$ a postupně vypočítáme:

n	p_n	$f(p_n)$
2	1.2631578947	−1.6022743840
3	1.3388278388	−0.4303647480
4	1.3666163947	0.0229094308
5	1.3652119026	−0.0002990679
6	1.3652300011	−0.0000002032

Všimněte si, že $|p_6 - p_5| = 0{,}0000180985$, což je hodnota menší než daná hodnota TOL.

Výhody metody sečen

- rychlá konvergence (většinou malý počet iterací pro dosažení dané přesnosti)
- jednoduchý princip a snadná implementace (naprogramování algoritmu v konkrétním programovacím jazyce)

Výhody metody sečen

- rychlá konvergence (většinou malý počet iterací pro dosažení dané přesnosti)
- jednoduchý princip a snadná implementace (naprogramování algoritmu v konkrétním programovacím jazyce)

Nevýhody metody sečen

- metoda nemusí konvergovat (v konečném počtu kroků metoda nedosáhne dané přesnosti)
- není k dispozici jednoduché kritérium pro stanovení odhadu počtu iterací potřebného pro dosažení dané přesnosti
- výsledkem metody není určení intervalu, ve kterém se kořen skutečně nachází

Nevýhody metody sečen

- metoda nemusí konvergovat (v konečném počtu kroků metoda nedosáhne dané přesnosti)
- není k dispozici jednoduché kritérium pro stanovení odhadu počtu iterací potřebného pro dosažení dané přesnosti
- výsledkem metody není určení intervalu, ve kterém se kořen skutečně nachází

Nevýhody metody sečen

- metoda nemusí konvergovat (v konečném počtu kroků metoda nedosáhne dané přesnosti)
- není k dispozici jednoduché kritérium pro stanovení odhadu počtu iterací potřebného pro dosažení dané přesnosti
- výsledkem metody není určení intervalu, ve kterém se kořen skutečně nachází

Rizika implementace metody na počítači

- při výpočtech musíme vzít v úvahu riziko ztráty přesnosti v důsledku zaokrouhlovacích chyb – například místo vztahu

$$p_{n+1} = p_n - \frac{f(p_n)(p_n - p_{n-1})}{f(p_n) - f(p_{n-1})}$$

nelze použít matematicky ekvivalentní vztah

$$p_{n+1} = \frac{f(p_n)p_{n-1} - f(p_{n-1})p_n}{f(p_n) - f(p_{n-1})}$$

- nesmíme zapomenout stanovit horní hranici počtu iterací algoritmu – mohlo by se stát, že algoritmus nikdy neskončí

Rizika implementace metody na počítači

- při výpočtech musíme vzít v úvahu riziko ztráty přesnosti v důsledku zaokrouhlovacích chyb – například místo vztahu

$$p_{n+1} = p_n - \frac{f(p_n)(p_n - p_{n-1})}{f(p_n) - f(p_{n-1})}$$

  nelze použít matematicky ekvivalentní vztah

$$p_{n+1} = \frac{f(p_n)p_{n-1} - f(p_{n-1})p_n}{f(p_n) - f(p_{n-1})}$$

- nesmíme zapomenout stanovit horní hranici počtu iterací algoritmu – mohlo by se stát, že algoritmus nikdy neskončí