Numerické metody

E 3011

Jan Böhm

RECETOX

March 29, 2023

Co nás dnes čeká

1. Babylónská odmocnina
2. Newtonova metoda
3. $\int$
4. Test
5. 🍕

Babylónská odmocnina

Odvození

Máme číslo x a chceme spočítat $\sqrt{S}$. Jak na to?

1. Nejprve si tipneme. Označíme x_0 jako tip.

Babylónská odmocnina

Odvození

Máme číslo x a chceme spočítat $\sqrt{S}$. Jak na to?

1. Nejprve si tipneme. Označíme x_0 jako tip.
2. Asi jsme se netrefili přesně, označíme ϵ jako chybu. Platí

$$S = (x_0 + \epsilon)^2 = x_0^2 + 2x_0\epsilon + \underbrace{\epsilon^2}_{\approx 0} \approx x_0^2 + 2x_0\epsilon$$

Babylónská odmocnina

Odvození

Máme číslo x a chceme spočítat $\sqrt{S}$. Jak na to?

1. Nejprve si tipneme. Označíme x_0 jako tip.
2. Asi jsme se netrefili přesně, označíme ϵ jako chybu. Platí

$$S = (x_0 + \epsilon)^2 = x_0^2 + 2x_0\epsilon + \underbrace{\epsilon^2}_{\approx 0} \approx x_0^2 + 2x_0\epsilon$$

3. Spočítáme, jak velké chyby ϵ jsme se dopustili $\epsilon = \frac{S - x_0^2}{2x_0}$.

Babylónská odmocnina

Odvození

Máme číslo x a chceme spočítat $\sqrt{S}$. Jak na to?

1. Nejprve si tipneme. Označíme x_0 jako tip.
2. Asi jsme se netrefili přesně, označíme ϵ jako chybu. Platí

$$S = (x_0 + \epsilon)^2 = x_0^2 + 2x_0\epsilon + \underbrace{\epsilon^2}_{\approx 0} \approx x_0^2 + 2x_0\epsilon$$

3. Spočítáme, jak velké chyby ϵ jsme se dopustili $\epsilon = \frac{S - x_0^2}{2x_0}$.
4. Opravíme náš původní odhad:

$$x_1 = x_0 + \epsilon = x_0 + \frac{S - x_0^2}{2x_0} = \frac{x_0^2 + S}{2x_0}$$

Babylónská odmocnina

Odvození

Máme číslo x a chceme spočítat $\sqrt{S}$. Jak na to?

1. Nejprve si tipneme. Označíme x_0 jako tip.
2. Asi jsme se netrefili přesně, označíme ϵ jako chybu. Platí
$$S = (x_0 + \epsilon)^2 = x_0^2 + 2x_0\epsilon + \underbrace{\epsilon^2}_{\approx 0} \approx x_0^2 + 2x_0\epsilon$$
3. Spočítáme, jak velké chyby ϵ jsme se dopustili $\epsilon = \frac{S - x_0^2}{2x_0}$.
4. Opravíme náš původní odhad:
$$x_1 = x_0 + \epsilon = x_0 + \frac{S - x_0^2}{2x_0} = \frac{x_0^2 + S}{2x_0}$$
5. Vesele iterujeme.

Babylónská odmocnina

Implementace

Implementujte funkci `sqrt(S, tol = 0.001)`, která spočítá babylónskou odmocninu čísla `S`. Rekurentní formule je:

$$x_n = \begin{cases} \frac{S}{2} & \text{pro } n = 0, \\ \frac{x_{n-1}^2 + S}{2x_{n-1}} & \text{jinak.} \end{cases}$$

Stop kritérium je $|x_n - x_{n-1}| < \texttt{tol}$.

Co nás dnes čeká

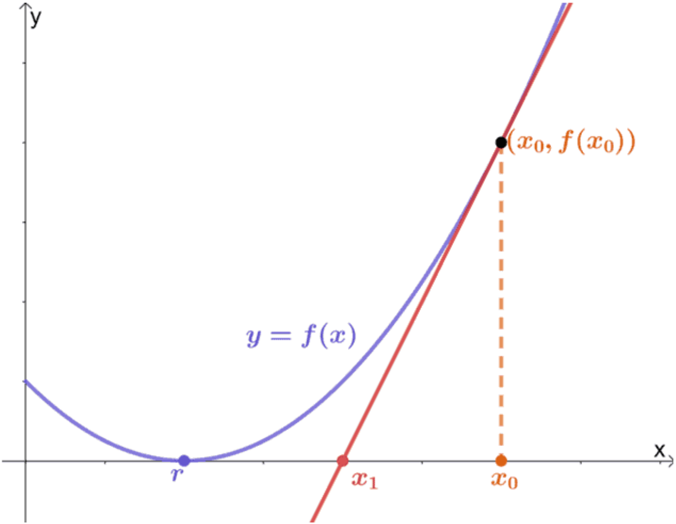
1. Babylónská odmocnina
2. Newtonova metoda
3. $\int$
4. Test
5. 🍕

Newtonova metoda

Odvození

Řešíme (přibližně) rovnici $f(x) = 0$, kde f je *hezká* funkce.

1. Začneme tak, že si řešení tipneme: x_0
2. Lokálně nahradíme funkci $f(x)$ přímkou a najdeme místo, kde tato přímka protíná osu x. Toto místo označíme jako x_1.
3. Iterujeme dle chuti.


y
$(x_0, f(x_0))$
$y = f(x)$
x
r
x_1
x_0

Newtonova metoda - odvození

Matematika

Chceme z n-tého odhadu x_n kořene r rovnice $f(x) = 0$ udělat přesnější, $(n+1)$-odhad x_{n+1}.

Newtonova metoda - odvození

Matematika

Chceme z n-tého odhadu x_n kořene r rovnice $f(x) = 0$ udělat přesnější, $(n+1)$-odhad x_{n+1}.

1. V okolí x_n nahradíme f přímkou (Taylorův polynom 1. stupně, lol).

Matematika

Chceme z n-tého odhadu x_n kořene r rovnice $f(x) = 0$ udělat přesnější, $(n+1)$-odhad x_{n+1}.

1. V okolí x_n nahradíme f přímkou (Taylorův polynom 1. stupně, lol).
2. Rovnice přímky $(y - y_k) = f'(x_k)(x - x_k)$.

Newtonova metoda - odvození

Matematika

Chceme z n-tého odhadu x_n kořene r rovnice $f(x) = 0$ udělat přesnější, $(n+1)$-odhad x_{n+1}.

1. V okolí x_n nahradíme f přímkou (Taylorův polynom 1. stupně, lol).
2. Rovnice přímky $(y - y_k) = f'(x_k)(x - x_k)$.
3. My chceme takové x_{k+1}, že $y = 0$. Navíc $y_k = f(x_k)$.

Newtonova metoda - odvození

Matematika

Chceme z n-tého odhadu x_n kořene r rovnice $f(x) = 0$ udělat přesnější, $(n+1)$-odhad x_{n+1}.

1. V okolí x_n nahradíme f přímkou (Taylorův polynom 1. stupně, lol).
2. Rovnice přímky $(y - y_k) = f'(x_k)(x - x_k)$.
3. My chceme takové x_{k+1}, že $y = 0$. Navíc $y_k = f(x_k)$.
4. $$-f(x_k) = f'(x_k)(x_{k+1} - x_k) \to x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Newtonova metoda - odvození

Matematika

Chceme z n-tého odhadu x_n kořene r rovnice $f(x) = 0$ udělat přesnější, $(n+1)$-odhad x_{n+1}.

1. V okolí x_n nahradíme f přímkou (Taylorův polynom 1. stupně, lol).
2. Rovnice přímky $(y - y_k) = f'(x_k)(x - x_k)$.
3. My chceme takové x_{k+1}, že $y = 0$. Navíc $y_k = f(x_k)$.
4. $$-f(x_k) = f'(x_k)(x_{k+1} - x_k) \rightarrow x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Stačí zvolit počáteční odhad x_0 a vymyslet, kdy přestat iterovat.
Nějaký problém?

Intermezzo - derivace

Derivace

Ne všechny funkce jdou hezky derivovat. Ale snad by šlo aspoň přibližně odhadnout hodnotu derivace funkce v nějakém bodě $f'(x)$.
Definice derivace $f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$.

`derivative(f, x, tol = 0.001)`

Napište funkci `derivative(f, x, tol = 0.001)`, která spočítá přibližnou hodnotu derivace funkce `f` v bodě `x` s přesností `tol`.
Využijte definice derivace výše. Začněte s nějakou rozumnou hodnotou h a tuto hodnotu zmenšujte, dokud změna v hodnotě odhadu derivace neklesne pod požadovanou `toleranci`.

Newtonova metoda - implementace

newton(f, x0, tol = 0.001)

Napište funkci `newton(f, x0, tol = 0.001)`, která spočítá přibližný kořen rovnice `f(x)=0` v okolí bodu `x(0)` s přesností `tol`.
Pro samotný výpočet použijte iterativní vztah

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)},$$

derivaci již spočítat umíte.
S iterováním přestaňte, jakmile rozdíl mezi předchozím a novým odhadem je menší, než `tol`.

```python
def myFun(x):
    return math.cos(x)-x

print(newton(myFun, 1))
```

Co nás dnes čeká

1. Babylónská odmocnina
2. Newtonova metoda
3. $\int$
4. Test
5. 🍕

`int(f, a, b, dx = 0.001)`

Napište funkci `int(f, a, b, dx = 0.001)`, která počítá přibližně integrál $\int_a^b f(x)\mathrm{d}x$
Vyberte si libovolnou metodu, kterou chcete implementovat – obdélníkovou (levou, pravou, střední, dolní-horní odhad), lichoběžníkovou, nebo klidně nějakou úplně jinou.

Co nás dnes čeká

1. Babylónská odmocnina
2. Newtonova metoda
3. $\int$
4. Test
5. 🍕

Test

S využitím vašich funkcí s přesností na 6 desetinných míst zodpovězte na následující otázky:

1. Vyčíslete $\sqrt{408849}$
2. Jaký je kořen funkce $\sin(\log^2(x))$ okolo bodu $x = 5$?
3. Polynom $4x^5 + 0x^4 + 8x^3 + 8x^2 + 4x + 9$ má jen jeden reálný kořen. Najděte jej.
4. Spočítejte $\int_0^1 2x^3 e^{x^2} dx$.

Co nás dnes čeká

1. Babylónská odmocnina
2. Newtonova metoda
3. $\int$
4. Test
5. 🍕

Takeaways

Po tomto cvičení byste měli mít připravené funkce

- `sqrt(S, tol = 0.001)` pro výpočet odmocniny
- `derivative(f, x, tol = 0.001)`, která spočítá přibližnou hodnotu derivace funkce `f` v bodě `x`
- `newton(f, x0, tol = 0.001)` pro numerický výpočet řešení rovnice `f(x)=0` poblíž `x0`.
- `int(f, a, b, dx = 0.001)`, která numericky integruje funkci `f` na intervalu `(a,b)` s krokem `dx`.

Tyto funkce si přidejte do své "knihovny" `functions.py`.
Po tomto cvičení byste měli umět:

- Implementovat další numerické metody pro řešení nelineárních rovnic jedné proměnné.
- Využívat své funkce pro řešení daných problémů a pokud to problém vyžaduje, funkci upravit.