



MATURITNÍ PRÁCE

Geometrická posloupnost a posloupnosti z ní odvozené

David Kolář

vedoucí práce: Mgr. Michaela Petrová

Prohlášení

Prohlašuji, že tato maturitní práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

V Českých Budějovicích dne podpis

David Kolář

Abstrakt

Klíčová slova

Poděkování

Rád bych vyjádřil hlubokou vděčnost vedoucí mojí práce, paní Michaela Petrové, za její neocenitelné vedení, mnoho užitečných rad a hlavně za její trpělivost. Dále bych rád poděkoval svému dobrému příteli Lukáši Knotovi, velkému zdroji inspirace, za poskytnutí nového pohledu na mou práci, konstruktivní kritiku a zpětnou vazbu.

Obsah

| | | |
|-----------|--|-----------|
| I | Teoretická část | 2 |
| 1 | Zavedení pojmu posloupnost | 3 |
| 1.1 | Definice a vysvětlení pojmu funkce | 3 |
| 1.1.1 | Zadání funkce | 3 |
| 1.2 | Vysvětlení pojmu posloupnost | 4 |
| 1.3 | Značení | 5 |
| 1.3.1 | Značení nekonečné posloupnosti | 5 |
| 1.3.2 | Značení konečné posloupnosti | 5 |
| 1.4 | Zadání posloupnosti | 5 |
| 1.4.1 | Vzorec pro n -tý člen | 6 |
| 1.4.2 | Rekurentní vzorec | 7 |
| 1.5 | Vlastnosti posloupností | 7 |
| 1.5.1 | Monotónnost posloupnosti | 8 |
| 1.5.2 | Omezenost posloupnosti | 9 |
| 2 | Řady | 10 |
| II | Praktická část | 11 |
| | Bibliografie | 12 |
| | Přílohy | 15 |
| A | Generátor vzorců v Pythonu | 16 |
| B | Příloha další | 18 |

Úvod

Vždy mě fascinovaly matematické posloupnosti a jejich součty, a proto jsem se rozhodl na toto téma zpracovat svou maturitní práci. V matematice a informatice se při určování časové a prostorové složitosti algoritmu často spoléháme na vyhodnocení právě součtu členů posloupností. Proto může být nalezení uzavřeného tvaru, neboli součtového vzorce posloupnosti, nesmírně užitečné.

Moje maturitní práce je rozdělena na dvě části: teoretickou část a praktickou část. V teoretické části začnu definicí toho, co je to matematická posloupnost a pojednám o jejích různých typech a vlastnostech. Poté se budu věnovat specifickým vlastnostem geometrické a aritmetické posloupnosti, včetně jejich součtových vzorců, a ukážu uplatnění geometrické posloupnosti při určování časové a prostorové složitosti. Dále představím pojem aritmeticko-geometrické posloupnosti a vysvětlím, čím se liší od ostatních typů posloupností. Nakonec ukážu svou vlastní posloupnost, kterou jsem objevil při studiu.

V praktické části své práce uvedu důkaz součtu geometrické, aritmeticko-geometrické a mojí posloupnosti.

Celkově si má maturitní práce klade za cíl poskytnout základní přehled týkající se matematických posloupností, konkrétně geometrické, aritmeticko-geometrické a mojí posloupnosti a představit mé poznatky a myšlenky na toto téma.

Část I

Teoretická část

1 Zavedení pojmu posloupnost

1.1 Definice a vysvětlení pojmu funkce

Pro zavedení pojmu posloupnost je nutné nejprve zavést a vysvětlit obecnější pojem funkce. V matematice označuje pojem funkce vztah mezi množinou vstupů a množinou výstupů s vlastností, že každému vstupu je přiřazen právě s jeden výstup. Vstup funkce se nazývá argument a výstup se nazývá hodnota funkce.

Funkce jsou důležitým nástrojem v matematice a používají se v mnoha různých oblastech, včetně fyziky, inženýrství, ekonomie a dalších. Lze je analyzovat pomocí různých matematických technik jako je infinitezimální počet a algebra. Nejprve zavedeme základní pojmy týkající se funkcí.

Definice 1 [Funkce]: Funkce na množině $\mathbb{D} \subset \mathbb{R}$ je předpis, který každému číslu z množiny \mathbb{D} přiřazuje právě jedno reálné číslo.

Definice 2 [Definiční obor]: Pokud máme funkci f , pak množině D , na které je funkce f definována, se říká *definiční obor* funkce f a značí se $D(f)$.

Definice 3 [Obor hodnot]: Obor hodnot je množina všech reálných čísel y , která lze dostat jako výstupní hodnotu funkce f , jestliže se za x dosadí všechny přípustné hodnoty z $D(f)$. Obor hodnot funkce f se značí $H(f)$.

1.1.1 Zadání funkce

Funkci lze zadat několika způsoby, ty nejvýznamnější si ukážeme na funkci $\sin(x)$.

Předpisem

- $f : y = \sin(x)$

- $y = \sin(x)$

- $x \rightarrow \sin(x)$

Toto vše jsou validní notace předpisu funkce. Při používání předpisu je nezbytné určit definiční obor.

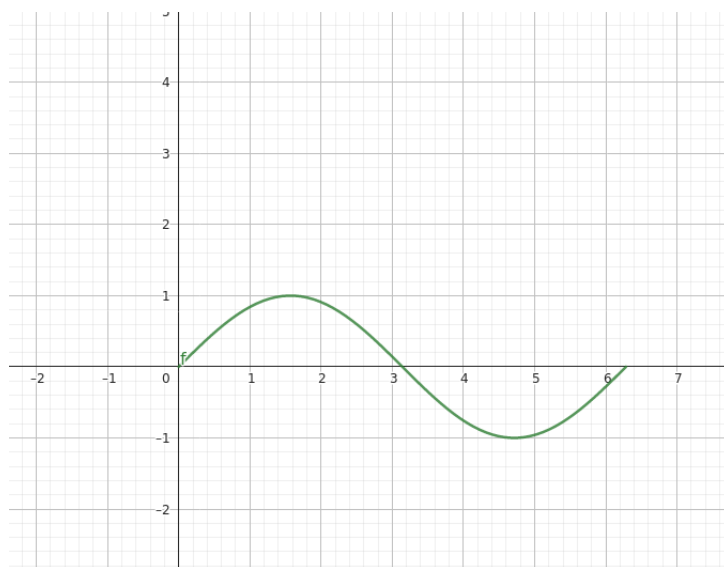
Tabulkou

První řádek představuje vstup funkce x a druhý řádek výstup funkce $\sin(x)$.

| | | | | | |
|-----------|---|-----------------|----------------------|----------------------|-----------------|
| x | 0 | $\frac{\pi}{6}$ | $\frac{\pi}{4}$ | $\frac{\pi}{3}$ | $\frac{\pi}{2}$ |
| $\sin(x)$ | 0 | $\frac{1}{2}$ | $\frac{\sqrt{2}}{2}$ | $\frac{\sqrt{3}}{2}$ | 1 |

Grafem

Graf funkce $\sin(x)$ s oborem definice $D(f) = \langle 0, 2\pi \rangle$. Pro vizualizaci jsem použil program Geogebra.



1.2 Vysvětlení pojmu posloupnost

Posloupnost je funkce definovaná na množině přirozených čísel, což jsou celá kladná čísla (1, 2, 3 atd.). Stejně jako u funkce tedy platí, že se jedná o vztah mezi množinou vstupů a množinou možných výstupů s vlastností, že ke každému vstupu je přiřazený právě jeden výstup. Definiční obor posloupnosti může být buď konečná množina, pak se posloupnost nazývá *konečná*, nebo *nekonečná* množina, pak se posloupnost nazývá *nekonečná*.

1.3 Značení

Prvky z definičního oboru posloupnosti (a_n) budu označovat n a funkční hodnoty posloupnosti (a_n) označovat a_n . Platí tedy, že n -tému prvku je přiřazena hodnota a_n .

1.3.1 Značení nekonečné posloupnosti

Nekonečné posloupnosti budu značit $(a_n)_{n=1}^{\infty}$.

1.3.2 Značení konečné posloupnosti

Konečné posloupnosti budu značit jako $(a_n)_{n=1}^k$ kde přirozené číslo k označuje maximální velikost n . Definiční obor posloupnosti (a_n) tedy zahrnuje všechna přirozená čísla od 1 do k .

Zápisem ve tvaru $n \in \{1, \dots, k\}$ vyjadřuji, že n nabývá hodnotu všech přirozených čísel od 1 do k .

1.4 Zadání posloupnosti

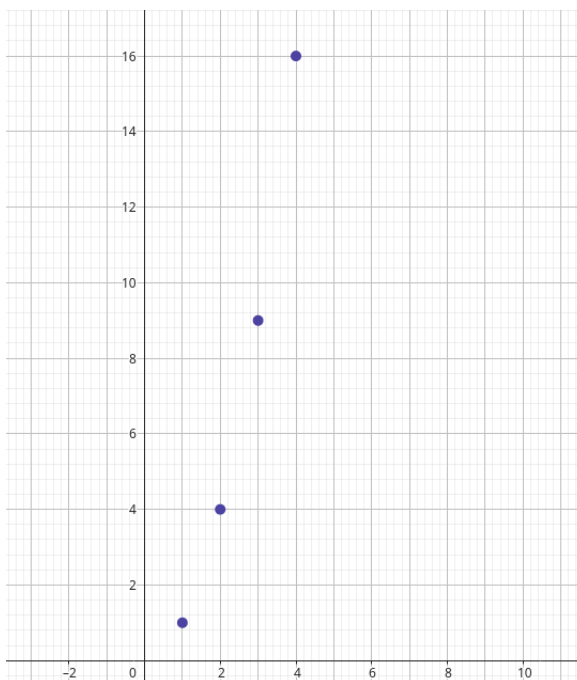
Posloupnost lze zadat různými způsoby v závislosti na povaze posloupnosti, potřebách dané situace a preferencích osoby, která je používá.

Jedním ze způsobů vyjádření posloupnosti je rekurentní vzorec, který definuje každý člen posloupnosti pomocí jeho předchozích členů.

Dalším způsobem vyjádření posloupnosti je pomocí vzorce, který definuje n -tý člen posloupnosti an pomocí proměnné n . Tento způsob může být užitečný třeba pro posloupnosti, které mají jednoduchý vzorec a lze je snadno definovat. To platí například pro aritmetickou nebo geometrickou posloupnost. Konečnou posloupnost lze vyjádřit výčtem hodnot:

$$(a_n)_{n=1}^k = [1, 4, 9, 16, 25]$$

Dále lze zadat konečnou posloupnost graficky. Tato metoda je užitečná pro vizualizaci posloupnosti, neboť s ní lze posloupnost lépe pochopit a interpretovat.



Poslední ze základních metod, jakou lze zadat konečnou posloupnost je s pomocí tabulky:

| | | | | | |
|-------|---|---|---|----|----|
| n | 1 | 2 | 3 | 4 | 5 |
| a_n | 1 | 4 | 9 | 16 | 25 |

1.4.1 Vzorec pro n -tý člen

Vzorcem pro n -tý člen lze zadat posloupnost konečnou i nekonečnou. Vzorec pro n -tý člen vyjadřuje vztah mezi hodnotou n z definičního oboru a a_n z oboru hodnot. Dosazením hodnoty n do tohoto vzorce lze zjistit n -tý člen a_n posloupnosti.

Příklady:

Nekonečnou posloupnost lze zapsat několika způsoby:

$$a_n = 2n + 1$$

Tento zápis nápadně připomíná formu zápisu funkce. Druhý způsob už se více blíží zápisu posloupností, jak jsme si jej definovali dříve.

$$(2n + 1)_{n=1}^{\infty}$$

Konečnou posloupnost, kde $n \leq k$ potom lze zapsat kupříkladu takto:

$$(2n + 1)_{n=1}^k$$

$$a_n = 2n + 1 \quad n \leq k \quad n \in \mathbb{N}$$

1.4.2 Rekurentní vzorec

Rekurentní vzorec určuje člen posloupnosti pomocí znalosti jednoho nebo více předcházejících členů. Tento způsob je užitečný zejména u posloupností, které mají složitý předpis a nelze je snadno definovat pomocí jediného vzorce. Známým příkladem je Fibonacciho posloupnost.

Člen posloupnosti a_n můžeme určit jen tehdy, pokud známe členy předcházející, tedy časová složitost určení n -tého prvku je $O(n)$.

Příklad na Fibonacciho posloupnosti:

Fibonacciho posloupnost, poprvé popsána v indické matematice již 200 let př. n. l., je posloupnost čísel, ve které je každé číslo, kromě prvních dvou čísel, 0 a 1, součtem dvou předchozích, tedy 0, 1, 1, 2, 3, 5, 8, 13, 21 atd.

$$a_n = \begin{cases} 0 & n = 1 \\ 1 & n = 2 \\ a_{n-1} + a_{n-2} & n > 2 \end{cases}$$

Tato posloupnost je sice nekonečná, nicméně pomocí rekurentního vzorce lze zadávat i konečné posloupnosti. Ukážeme si to opět na předchozím příkladu:

$$a_n = \begin{cases} 0 & n = 1 \\ 1 & n = 2 \\ a_{n-1} + a_{n-2} & 2 < n \leq 10 \end{cases}$$

1.5 Vlastnosti posloupností

Protože je posloupnost definovaná jako funkce s definičním oborem přirozených čísel, lze mnoho vlastností, které platí pro funkce obecně, určit i pro posloupnosti, například monotónnost nebo omezenost.

1.5.1 Monotónnost posloupnosti

Jednou z vlastností, kterou lze pro posloupnost určit, je monotónnost, která se týká rostoucí nebo klesající povahy jednotlivých prvků.

Definice 4 [Rostoucí posloupnost]: O posloupnosti se říká, že je rostoucí, jestliže je každý člen větší než předchozí.

$$\forall n, m \in D : n < m \Rightarrow a_n < a_m \quad (1.1)$$

Definice 5 [Klesající posloupnost]: Posloupnost je klesající, jestliže je každý člen menší než předchozí člen.

$$\forall n, m \in D : n < m \Rightarrow a_n > a_m \quad (1.2)$$

Definice 6 [Nerostoucí posloupnost]: O posloupnosti se říká, že je nerostoucí, jestliže je každý člen menší nebo roven předchozímu členu.

$$\forall n, m \in D : n < m \Rightarrow a_n \geq a_m \quad (1.3)$$

Definice 7 [Neklesající posloupnost]: O posloupnosti se říká, že je neklesající, jestliže je každý člen větší nebo roven předchozímu členu.

$$\forall n, m \in D : n < m \Rightarrow a_n \leq a_m \quad (1.4)$$

Definice 8 [Nerostoucí posloupnost:] O posloupnosti se říká, že je nerostoucí, jestliže je každý člen menší nebo roven předchozímu členu. Posloupnost (a_n) se nazývá nerostoucí právě tehdy, když:

$$\forall n, m \in D : n < m \Rightarrow a_n \geq a_m. \quad (1.5)$$

Definice 9 [Neklesající posloupnost]: O posloupnosti se říká, že je neklesající, jestliže je každý člen větší nebo roven předchozímu členu. Posloupnost (a_n) se nazývá neklesající právě tehdy, když:

$$\forall n, m \in D : n < m \Rightarrow a_n \leq a_m \quad (1.6)$$

1.5.2 Omezenost posloupnosti

Další vlastností, kterou lze u posloupnosti určit, je omezenost, která se týká maximálních a minimálních hodnot, kterých mohou členy nabývat a neomezená, pokud žádná taková hranice neexistuje. To je třeba opravit.

Definice 10 [Shora omezená]: Posloupnost (a_n) je shora omezená, jestliže existuje $h \in \mathbb{R}$ takový, že pro všechny $n \in D$ platí $a_n \leq h$.

Definice 12 [Zdola omezená]: Posloupnost (a_n) je zdola omezená, jestliže existuje $d \in \mathbb{R}$ takový, že pro všechny $n \in D$ platí $a_n \geq d$.

Definice 12 [Omezená]: Posloupnost (a_n) je omezená, jestliže je shora i zdola omezená, tedy existují $h, d \in \mathbb{R}$ takové, že pro všechny $n \in D$ platí $d \leq a_n \leq h$.

2 Řady

Řada vznikne sečtením prvků posloupnosti. Pokud je posloupnost konečná, vznikne konečná řada, pokud je posloupnost nekonečná, vznikne sečtením jejích členů nekonečná řada. V textu jsem se zatím pojmu řada vyhýbal, i když by mohl být na mnoha místech užitečný, takže ho postupně do textu zařadím.

Část II

Praktická část

Odvození vztahu

Značení: Uvažujme e , kde $e \in \mathbb{Z}_{\geq 0}$. Pak ${}^e a_i$ je člen posloupnosti ve tvaru $i^e q^i$ a ${}^e s_n = \sum_{i=0}^n {}^e a_i$

Věta 1: Necht' $q \in \mathbb{R} \setminus \{1\}$

$${}^1 s_n = \frac{q}{q-1} \left(nq^n + q^n - \frac{q^{n+1} - 1}{q-1} \right)$$

Lemma:

$${}^0 s_n = \sum_{i=0}^n q^i = \frac{q^{n+1} - 1}{q-1}$$

Z definice platí, že:

$${}^0 s_{n+1} = {}^0 s_n + q^{n+1} \quad (2.1)$$

Dále uvedu ještě jeden vztah mezi ${}^0 s_{n+1}$ a ${}^0 s_n$, který již není tolik zřejmý. Vynasobením každého členu ${}^e a_i$ koeficientem q získám člen o stupeň větší, neboť platí:

$${}^0 a_n q = q^i q = q^{i+1} = {}^0 a_{n+1}$$

Tím vytvořím posloupnost ${}^0 a_1 \dots {}^0 a_{n+1}$. Přičtením ${}^0 a_0$ mi tedy vznikne posloupnost ${}^0 a_0 \dots {}^0 a_{n+1}$ a tudíž platí, že:

$${}^0 s_{n+1} = {}^0 s_n q + 1$$

Dosadím vztah (1):

$${}^0 s_n + q^{n+1} = {}^0 s_n q + 1$$

Vyjádřím si ${}^0 s_n$ a získám námi hledaný tvar.

$${}^0 s_n = \frac{q^{n+1} - 1}{q-1}$$

Důkaz:

Budu postupovat podobně jako v lemmatu. Z definice platí:

$${}^1 s_{n+1} = {}^1 s_n + (n+1)q^{n+1} \quad (2.2)$$

Mějme posloupnost ${}^1a_0 \dots {}^1a_n$. Ke každému členu 1a_i přičtu odpovídající 0a_i . Takto získám novou posloupnost, kde každý člen odpovídá tvaru: $(x+1)q^x$. Náhlednu, že vynásobením členu q získám člen ${}^1a_{i+1}$.

$$q({}^1a_i + {}^0a_i) = {}^1a_{i+1}$$

Z čehož plyne:

$${}^1s_{n+1} = q({}^1s_n + {}^0s_n)$$

Dosadím si vztahy (1), (2):

$${}^1s_n + (n+1)q^{n+1} = q({}^1s_n + \frac{q^{n+1}-1}{q-1})$$

Vyjádřím si 1s_n .

$${}^1s_n = \frac{(n+1)q^{n+1} - q(\frac{q^{n+1}-1}{q-1})}{q-1}$$

Výsledný vztah upravím na námi hledaný tvar:

$${}^1s_n = \frac{q}{q-1}(nq^n + q^n - \frac{q^{n+1}-1}{q-1})$$

Věta 2: Necht' $e > 0$, potom:

$${}^e s_n = \frac{(n+1)^e q^{n+1} - q(\sum_{i=1}^e \binom{e}{i} {}^{e-i} s_n)}{q-1}$$

Vyjádřím si rozdíl $\frac{{}^e a_{n+1}}{q}$ a ${}^e a_n$ pomocí binomické věty.

$$\frac{{}^e a_{n+1}}{q} - {}^e a_n = (n+1)^e q^n - n^e q^n = \sum_{i=1}^e \binom{e}{i} n^{e-i} q^n$$

Mějme opět posloupnost ve tvaru $(i^e q^i)_{i=0}^n$. Ke každému členu posloupnosti přičtu tento rozdíl a dostanu posloupnost. $((i+1)^e q^i)_{i=0}^n$ Nyní vynásobím každý její člen q .

Závěr

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Seznam obrázků

Seznam tabulek

Přílohy

A Generátor vzorců v Pythonu

Na závěr implementuji algoritmus, který jsem odvodil v praktické části. Program přijímá jako vstup polynom, základ a maximální hodnotu x . Pro srovnání jsem do programu přidal funkci, která počítá součet hrubou silou, abych demonstroval, že se hodnoty skutečně rovnají a též tak zkontroloval svoji implementaci.

```
1 # importovani modulu
2 import sympy
3 from math import comb
4 import re
5
6 def hrubou_silou(polynom, q, exp):
7     """Reseni hrubou silou pro overeni spravnosti implementace.
8         """
9     soucet = 0
10    for i in range(exp+1):
11        soucet += eval(polynom.replace("x", str(i)))*q**i
12    return soucet
13
14 def spocitej_mocniny(stupen):
15     """Funkce, která urci soucet jednotlivych  $x^i \cdot q^i$ , pro  $0 \leq i \leq$ 
16         maximální stupeň polynomu"""
17     nulty_clen = "(q**(x+1) - 1)/(q - 1)"
18     stupne = [sympy.sympify(nulty_clen)]
19     for exp in range(1, stupen+1):
20         novy_clen = sympy.sympify("0")
21         for k in range(exp):
22             novy_clen += sympy.sympify(comb(exp, k)*stupne[k])
23         novy_clen = sympy.sympify(f"((x+1)**{exp})*q**(x+1)"
24             - sympy.sympify("q")*novy_clen)
25         novy_clen /= sympy.sympify("q - 1")
26         stupne.append(sympy.simplify(novy_clen))
27     return stupne
28
29 def vyres(polynom, q, x):
```

```

27 """Sestavi z jednotlivych vzorcu vzorec pro dany polynom, do
    ktereho dosadi spravna cisla."""
28 vzorec = sympy.simplify(polynom)
29 vzorec = vzorec.expand()
30 koeficienty = sympy.Poly(vzorec).all_coeffs()[::-1]
31 stupne = spocitej_mocniny(len(koeficienty) - 1)
32 soucet = 0
33 for index, k in enumerate(koeficienty):
34     val = str(stupne[index])
35     val = eval(val.replace("q", str("q")).replace("x", str(x
        ))))*k
36     soucet += val
37 return int(soucet)
38
39 # Nacteni vstupu
40 polynom = input()
41 q = input()
42 x = input()
43
44 # Vystup
45 print("Hrubá síla:", hrubou_silou(polynom, q, x))
46 print("Vzorcem:    ", vyres(polynom, q, x))

```

Zdrojový kód A.1: generator.py

Výstup pro $q = 10$, $x = 5$ a polynom roven $100x^5 - 3x + 2587x(x+1)$. Je vidět, že hodnoty se rovnají.

```

Hrubá síla: 40608039310
Vzorcem:    40608039310

```

Výstup našeho programu

B Příloha další